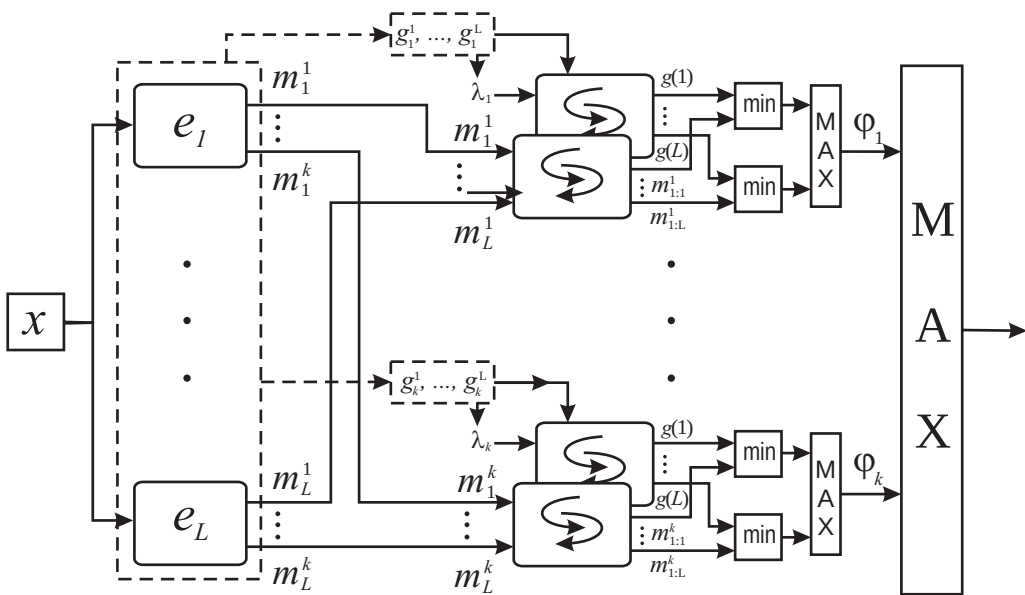


Дејан Ѓорѓевиќ
Душан Чакмаков

Комбинирана на класификатори за препознавање на примероци



Скопје, 2006

Дејан Ѓорѓевиќ
Душан Чакмаков

**Комбинирање на
класификатори за
препознавање
примероци**

Скопје, 2006

Печатено во Македонија, 2006
Машински факултет - Скопје

Автори

Д-р Дејан Ѓорѓевиќ, Доцент, Институт за компјутерска техника и информатика, Електротехнички факултет, Универзитет „Св. Кирил и Методиј“ - Скопје
dejan@etf.ukim.edu.mk

Д-р Душан Чакмаков, Професор, Оддел за математика и информатика, Машински факултет, Универзитет Св. Кирил и Методиј“ - Скопје
dusan@mf.ukim.edu.mk

CIP - Каталогизација во публикација
Национална и универзитетска библиотека „Св. Климент Охридски“,
Скопје

004.93'1

ЃОРЃЕВИЌ, Дејан

Комбинирање на класификатори за препознавање примероци / Дејан Ѓорѓевиќ, Душан Чакмаков. - Скопје : Машински факултет, 2006. – 202 стр. : илустр. ; 24 см

Библиографија: стр. [141]-150

Содржи и:
Додаток

ISBN 9989-2131-8-6

1. Чакмаков, Душан

а) Апликативни компјутерски техники – Препознавање облици
COBISS.MK-ID 66077962

Печати: ЈОФИ СКЕН

Воведен збор

Комбинирањето на класификатори веќе поголем број години е активно поле на истражување во светот, но и техника применета во пракса во многу системи за препознавање примероци. И покрај тоа, денеска во светот практично и да нема книга комплетно посветена на комбинирање на класификатори. Најчест е случајот на оваа тема да ѝ се посвети една глава во книгите за препознавање примероци.

Авторите во оваа книга детално ги дискутираат проблемите на комбинирање класификатори, пред сè методите за статистичка фузија на нивните одлуки. Други видови на фузија на одлуките, како што се правилата или каскадните методи се дискутирани во некои од трудовите на авторите на крајот од книгата. Изложени се голем број методи за комбинирање на класификатори и нивни варијации, а понатаму се дадени заклучоците за тоа како методите се однесуваат во различни ситуации и кои методи треба да се преферираат во кои ситуации. Додатокот е посебна целина каде што авторите ги имаат дадено нивните резултати во областа на комбинирањето на класификатори во последните неколку години. Неколку резултати на авторите заслужуваат внимание. Интересен е тростепениот систем за препознавање на цифри испишани со рака базиран на 2 невронски мрежи еден SVM класификатор изложен како 3-ти труд во додатокот. Исто така би го истакнал големиот број презентирани шеми на комбинирање и нивните варијации изложени во втората глава. На крај би ја нагласил спремноста на авторите да го отворат суштинското прашање, дали и колку е корисно комбинирањето на класификатори во апликациите за препознавање примероци. Се разбира, дефинитивен одговор во овој момент не може да се даде, но дискусиите на авторите во третата глава заслужуваат внимание.

Оваа книга е пионерски потфат во областа на комбинирањето на класификатори во нашата земја. Убеден сум дека таа ќе им биде од полза на студентите, универзитетските професори, истражувачите, и на сите други што се занимаваат со машинското учење и имаат потреба од користење на техниките на комбинирање на класификатори за препознавање примероци.

Проф. Д-р Данчо Давчев

Институт за компјутерска техника и информатика,
Електротехнички факултет,
Универзитет Св. Кирил и Методиј,
Скопје, Македонија

Предговор

Способноста за препознавање е секако една од клучните особини на интелигентното однесување. Препознавањето примероци како можност на техничките системи е еден од главните предизвици на машинската интелигенција. Неговата задача е да реагира целисходно на одредени сигнали кои доаѓаат од околината. Препознавањето примероци т.е. нивната класификација претставува пресликување од множеството примероци во множеството класи. Обично, секој примерок е претставен со множество обележја коишто го карактеризираат, т.е. овозможуваат да се одреди на која класа тој и припаѓа. Задача на класификаторот е по неговата обука со веќе класифицирани примероци да може правилно да ги класифицира (препознае) примероците што не ги „видел“ (да генерализира). Резултатите на препознавањето може да се подобрат со подобрување и зголемување на бројот на обележјата кои се користат или со подобрување на самите класификатори. Еден од начините што може да се искористи за подобрување на резултатите на препознавањето е да се комбинираат одлуките на повеќе класификатори наместо да се користи само еден класификатор. Податоците од повеќе извори кои се обработуваат одвоено често пати може ефективно да се искombинираат со цел да се добие поконцизна, покомплетна и попрецизна претстава за ситуацијата.

Оваа книга е посветена на комбинирањето на класификатори и е пред сè мотивирана од фактот што и покрај релативно големиот број предложени методи за комбинирање на класификатори релативно мал број публикувани трудови се занимаваат со споредба на методите и сугерираат која од нив да се избере во конкретна ситуација. Книгата произлезе како резултат на научноистражувачкиот проект „Комбинирање и оптимизирање на класификатори за препознавање облици“ спонзориран од Министерството за образование и наука на Република Македонија.

Книгата содржи вовед, три глави, заклучок и прилог.

Во **првата глава** е дефиниран проблемот на препознавање примероци и практичните проблеми кои се наметнуваат при креирањето на класификатори.

Втората глава е посветена на принципите на комбинирање на класификатори, како и методите за добивање на класификатори погодни за комбинирање. Претставени се голем број методи за комбинирање на класификатори, теориските основи на кои почиваат и начините за нивна реализација. Изложени се и неколку нови, како и модификации и проширувања на некои од постојните методи за комбинирање на класификатори.

Последната, **трета глава** ги сумира резултатите и ги обединува заклучоците за обработените проблеми и ги дава можните насоки за натамошни истражувања. Таа може да се гледа и како упатство за користење на техниките за комбинирање на класификатори во апликациите на препознавање примероци.

Во **прилогот** во целост се дадени 5 трудови на авторите за комбинирање на класификатори, публикувани во последните 4 години. Два труда се презентирани на светски конференции за препознавање облици, еден на меѓународна конференција во нашата земја, еден во меѓународно и еден во домашно списание.

Книгата е наменета на истражувачите од областа на комбинирање на класификатори, како и за сите оние што имаат потреба или сакаат да применат метода на комбинирање класификатори во практична апликација на препознавање примероци.

Авторите му изразуваат посебна благодарност на Министерството за образование и наука на Република Македонија за обезбедените сретства за реализација на проектот од кој произлезе и оваа книга.

Авторите,

Окопје, 10 јули 2006

Содржина

ВОВЕД.....	1
1 ПРЕПОЗНАВАЊЕ ПРИМЕРОЦИ	3
1.1 Теоретски аспект на препознавањето примероци	3
1.2 Практичен аспект на препознавањето примероци.....	5
1.3 Класификатори	6
1.4 Невронска мрежа како класификатор	26
1.5 Машини со носечки вектори (SVM)	27
1.6 Решавање на повеќекласни класификациски проблеми со SVM класификатори	29
1.7 Конверзија на излезите од SVM класификаторите во веројатности.....	33
2 КОМБИНИРАЊЕ НА КЛАСИФИКАТОРИ.....	35
2.1 Традиционален пристап за препознавање примероци	36
2.2 Системи за препознавање примероци со повеќе класификатори.....	37
2.3 Ансамбл од класификатори	38
2.4 Декомпозиција на грешката.....	41
2.5 Методи за креирање на класификатори за комбинирање	43
2.6 Различни пристапи при комбинирање на класификатори	47
2.7 Комбинирање по пат на гласање	52
2.8 „Наивно“ баесово комбинирање	54
2.9 Комбинирање со познавање на поведението на класификаторите – BKS (Behavior Knowledge Space).....	58
2.10 Методи за комбинирање со прeredување на ранг листата на класи	62
2.11 Комбинирање по пат на усреднување.....	66
2.12 Комбинирање со подредување	69
2.13 Демпстер-Шефер комбинирање	71
2.14 Веројатносен производ.....	74
2.15 Комбинирање со неспецифициран интеграл.....	75
2.16 Комбинирање со тежинско усреднување	82
2.17 Тежинско комбинирање со динамичко одредување на тежините.....	90
2.18 Комбинирање со шаблони за одлучување.....	91
2.19 Комбинирање со проширени шаблони за одлучување.....	99
2.20 Комбинирање со повеќекратна линеарна регресија (MLR)	108
2.21 Комбинирање како класификациска задача	111
2.22 Комбинирање на еден-против-еден класификатори.....	114
3 НАШИ ИСКУСТВА И ЗАКЛУЧОЦИ	117
3.1 Комбинирање на класификатори со различни методи.....	117
3.2 Обучени или фиксни методи на комбинирање	130

3.3	Комбинирање на класификатори со различни перформанси	131
3.4	Комплексност на методите за комбинирање	133
3.5	Зависноста на перформансите на обучените методи за комбинирање на класификатори од бројот на примероци за нивна обука	134
3.6	Комбинирање на еден-против-еден класификатори	135
ЗАКЛУЧОК.....		137
ЛИТЕРАТУРА		141
ДОДАТОК		151
ОДБРАНИ ПУБЛИКАЦИИ НА АВТОРИТЕ.....		151
	Combining SVM Classifiers for Handwritten Digit Recognition	153
	Decision Fusion and Reliability Control in Handwritten Digit Recognition System	161
	Partitioning of the Feature Set for Classifier Cooperation	177
	An Efficient Three-Stage Classifier for Handwritten Digit Recognition.....	187
	Handwritten Digit Recognition Using Classifier Cooperation Schemes	195

Вовед

Препознавањето примероци е еден од главните предизвици на машинската интелигенција. Менталните процеси со кои се постигнува препознавањето кај човекот се сложени и сè уште недоволно познати за да бидат директно применети во светот на машините. Задачата на машинското препознавање примероци е да ги процени и интерпретира мострите во сигналите кои претставуваат репрезентација на дел од реалниот свет, а се измерени со помош на одредени инструменти или апаратура.

Апликации кои во себе вклучуваат решавање на проблеми од областа на препознавање примероци се: препознавање на текст, верификација на потписи, препознавање објекти на слики и видео-секвенци, роботска визија, предвидување на временски настани како што се оптовареност на телефонски, електрични и компјутерски мрежи, движење на цени на берзата, автоматска дијагностика и автоматизација на лабораториски испитувања (на пр. броење на крвните зрнца) и др. Во индустријата проверката на исправност и контролата на квалитет, како и автоматизираното составување на механички делови генерално бараат препознавање примероци. Слични барања има и во автоматската анализа на сателитски слики, радарски сигнали, подводни сонарни снимања, автоматското лоцирање и одредување на мета во воената индустрија и др.

Водена од барањата на многу практични и комерцијални апликации, минатата деценија донесе значаен напредок во областа на истражувањето на методите за препознавање примероци, како и презентирање на потполно нови типови на класификатори. И покрај нивната успешна примена во голем број комерцијални апликации, обработката на примероци со големи варијации, на примероци со шум и/или со голем број класи сè уште претставува проблем за класичните методи на препознавање примероци.

Карактеристиката на класификаторите правилно да ги класификуваат и примероците за кои не биле обучени и кои прв пат ги гледаат, односно да генерализираат, е најверојатно нивната најважна особина според која тие се вреднуваат. Различните архитектури на класификатори или нивното обучување под различни услови резултира во класификатори со различни перформанси. Затоа најчесто во пракса при решава-

њето на одреден класификациски проблем се продуцираат, обучуваат и проценуваат повеќе класификатори, за да се обезбеди правилен избор на моделот и параметрите на класификаторот. Меѓутоа, изборот на „најдобриот“ меѓу нив не претставува неопходно и идеален избор, бидејќи со тоа би се отфрлиле значителни информации од резултатите кои би ги обезбедиле глобално помалку успешните класификатори. Исто така, проценката на перформансите на класификаторите користејќи ограничен број на познати примероци, не мора секогаш да соодветствува со нивните реални перформанси.

Идејата за комбинирање на повеќе класификатори со цел да се компензираат нивните индивидуални слабости, а да се зачуваат нивните индивидуални квалитети, сè пошироко се користи во модерните апликации за препознавање примероци. Податоците од различни извори, одделно обработени честопати можат да бидат корисно искombинирани, што придонесува кон покомплетна и попрецизна претстава на реалната ситуација. Комбинирањето на класификатори за препознавање примероци е интензивно поле на истражување во последниве години. На овој проблем му се посветени поголем број посебно организирани научни собири и посебни изданија на најпознатите стручни и научни списанија. Сепак, основното прашање: „Дали, и во кои случаи и под кои услови продуцирањето на повеќе класификатори и нивно комбинирање е целисходно и води кон подобрување на перформансите?“ е сè уште без јасен генерален одговор. Во последниве години се предложени мноштво методи за комбинирање на класификатори, на пр.: различни шеми за гласање, обично и тежинско усреднување, линеарно и нелинеарно комбинирање и повеќефазни стратегии за комбинирање во кои излезите од индивидуалните класификатори се нови обележја кои се користат како влез за други класификатори. Иако најголем дел од истражувањата се фокусирани кон изнаоѓање на методи за комбинирање кои ги подобруваат перформансите на препознавањето, конструирањето на самите класификатори кои ќе бидат комбинирани, е исто така многу важно, бидејќи доколку тие постојано се сложуваат во индивидуалните одлуки, не може многу да се добие со нивно комбинирање. Од ова произлегува дека изборот и обучувањето на класификаторите наменети за комбинирање е исто така важно како и самиот избор на шемата за нивно комбинирање, бидејќи перформансите на индивидуалните класификатори и нивната независност директно влијаат на перформансите на комбинацијата.

I

Препознавање примероци

Во оваа глава се претставени основните концепти на машинското учење, особено од аспект на препознавањето примероци. Дадени се основните теоретски концепти на проблемот на класификација, како и практичните проблеми кои се наметнуваат при креирањето на класификатори. Изложена е и поделба на класификаторите од неколку аспекти.

Разгледани се два типа на класификатори: невронски мрежи и машини со носечки вектори (Support Vector Machine – SVM). Невронските мрежи веќе подолго време доста често се употребуваат во практика за решавање на класификациски проблеми. Машините со носечки вектори се релативно нов и многу ветувачки концепт во областа на препознавањето примероци.

Наведени се и неколку начини за решавање на повеќекласен проблем со интеграција на неколку SVM класификатори кои се во основа бинарни (двокласни) класификатори. Изложени се и неколку начини за конверзија на излезите на SVM класификаторот во постериорни веројатности кои се неопходни за некои од методите за комбинирање на класификатори.

1.1 Теоретски аспект на препознавањето примероци

Препознавањето е едно од основните својства на човекот. Додека човекот сосема лесно црните и белите полиња на хартијата ги интер-

претира како букви, линии, форми, за компјутерот тие и понатаму остануваат само низи од единици и нули во неговата меморија. Екстрахирањето на основните информации од низата на единици и нули во компјутерската меморија и нивното претставување и интерпретација во форма соодветна за понатамошна обработка со компјутер се сведува на проблемот на препознавање облици. Облиците кои треба да се препознаат можат да бидат печатени букви, ракопис, технички цртежи, мапи итн.

Зборот *pattern* (примерок, модел, шаблон, калап), потекнува од зборот *патрон* и во оригиналната употреба означува нешто што претставува совршен пример за имитирање. Препознавањето примероци (*pattern recognition*) го има значењето на идентификација на идеалот според кој даден објект е направен. Во психологијата препознавањето примероци е дефинирано како процес во кој „екстерните“ сигнали кои доаѓаат во сетилата се претвораат во препознатливи перцептуални искуства.

Едно од клучните прашања кое се наметнува е како се создава концептот за идеалното. Тој може да се формира преку дедуктивен или индуктивен процес. Во првиот случај се смета дека концептот за идеалното е вроден кај набљудувачот, додека кај индуктивното создавање на концептот, набљудувачот го создава концептот за идеалното преку набљудување на многу неидеални примероци. Овој процес може да се нарече и учење, и тоа учење со учител доколку примероците се означени како претставници на еден или повеќе идеали, или учење без учител доколку примероците не се означени.

Учењето како да се препознае објект, односно како да се одреди на која класа од претходно специфицирано множество на класи и припаѓа, е една од клучните особини на интелигенцијата која е опсежно истражувана и кај човекот (природна интелигенција) и кај компјутерите (вештачка интелигенција). Класификацијата и придружниот проблем на учење на описите на класите овозможуваат да се издвои што е суштинско за да се биде припадник на одредена класа. Издвојувањето на суштинските обележја е од огромно значење бидејќи овозможува да се насочи вниманието на интелигентниот агент (било тоа да се човек или машина). На пример, луѓето генерално знаат дека доколку сакаат да дознаат дали животното зад дрвото е леопард, треба да се фокусираат да гледаат дали има дамки на крзното а не која му е бојата на очите. Способноста да се разликуваат леопардите од не-леопардите и способноста да се научи ваквото диференцирање е од егзистенционално значење особено во временски критични ситуации. Затоа и не е за чудење што човекот релативно лесно успева да изведе препознавање на комплексни примероци, но, има тешкотии со брзото множење на повеќецифрени

броеви. Фактот што човековиот нервен систем не е прилагоден за брзо изведување на нумерички операции нè изненадува, имајќи го предвид фактот што способноста да се изведуваат вакви операции и немало некое особено значење за опстанокот на единката (во биолошка смисла на зборот).

Друг клучен придонес при учењето е генерализацијата, односно издвојувањето на суштинските карактеристики кои означуваат дека одреден објект е припадник на одредена класа, што овозможува да се класификуваат нови (дотогаш невидени) објекти во некоја од познатите класи.

Менталните процеси со кои се постигнува препознавањето се очигледно сложени и сè уште недоволно познати. Препознавањето примероци не му е својствено само на човекот, тоа е присутно и кај животните и имало значајна улога во еволуциска смисла. Притоа, генерализацијата овозможила организмот успешно да се снајде во нови ситуации, и преземајќи слични акции или однесување, на некои кои се покажале успешни во претходното искуство со ситуации од ист или сличен тип, успешно да излезе од нив. Природно, нервниот систем на човекот е многу поефикасен во справувањето со проблемите кои се од животно значење за човекот (препознавање на опасни ситуации) отколку со оние кои не се.

1.2 Практичен аспект на препознавањето примероци

Појавата на дигиталните компјутери стави на располагање машини кои далеку ги надминуваат човековите способности за комплексни нумерички операции. Се постави и прашањето дали овие машини ќе се покажат подеднакво успешни и во ненумеричките операции, особено во менталните функции како препознавањето примероци. Проблемите од оваа категорија генерално се адресирани како вештачка интелигенција или машинска интелигенција. Се покажа дека раниот оптимизам во поглед на способноста на компјутерите да решаваат проблеми од класата на машинската интелигенција не беше оправдан. Човековиот мозок има многу повисок степен на комплексна организација и од најнапредните компјутери.

Сепак со развојот на науката границите на она што компјутерот не може да го направи полека но, сигурно се поместуваат. Постојат многу работи во чија автоматизација е потребно препознавање примероци. Сортирањето на поштата и внесување на податоци во компјутер бараат препознавање на печатени или ракописни букви. Броењето на крвните зрнца, проверката на исправност и контролата на квалитетот,

како и автоматизираното составување на механички делови генерално бараат препознавање облици. Слични барања има и во автоматската анализа на сателитски слики, радарски сигнали, подводни сонарни снимача и др.

Според повеќето енциклопедии препознавањето примероци се дефинира како дисциплина која ја изучува теоријата и методологијата на дизајнирање на машини кои се во состојба да препознаваат примероци од податоци загадени со шум. Иако ова е само една од можните дефиниции за препознавањето примероци, во неа јасно се истакнува „инженерската“ природа на оваа научна дисциплина во која се вели дека крајната цел на оваа дисциплина е дизајнирањето на „машини“. Од оваа инженерска перспектива јасно е дека теоријата на препознавањето примероци има мултидисциплинарни корени, како што впрочем и повеќето инженерски дисциплини настојуваат да го пребродат јазот помеѓу реалните технички апликации на теоријата и т.н. чисти теориски дисциплини како математиката, статистиката, физиката и др.

1.3 Класификатори

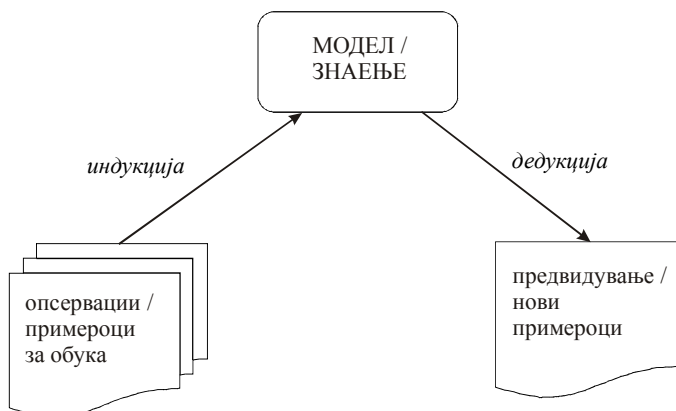
Во практика под препознавање примероци се подразбира автоматска детекција и класификација на објекти или настани. Мерките и податоците кои се користат за класифицирање на објектите се нарекуваат особини или обележја (features), а прототиповите или категориите во кои се распоредуваат се нарекуваат класи. Постојат техники (како кластерирањето на пример) што овозможуваат ненадгледувано учење (учење без учител) односно автоматско откривање на класите, кои се обидуваат да го поделат множеството податоци во природно појавни групи без предетерминирана структура на класите. Во практиката најчесто постои множество класирани податоци кое може да се употреби при дизајнирањето на автоматскиот систем. Индивидуалните единици, објекти или настани кои треба да се класираат се нарекуваат примероци или мостри. Множеството класирани примероци кое се употребува во дизајнот на системот се нарекува *множесѝво за обука* (training set), додека множеството исто така класирани примероци кое се употребува за тестирање, евалуација и нагудување на некои дополнителни параметри на системот се вика *множесѝво за тѝсѝтирање* (test set).

1.3.1 Конструирање (обучување) на класификатори

Неформално зборувајќи една од задачите на машинското учење е да генерира (произведе) модел од зададено множество на обележани примероци. Ваквиот модел потоа може да се употреби за класификаци-

ја на необележани, претходно невидени примероци. Ваквото својство на обучениот модел правилно да препознава и нови примероци што не биле презентирани за време на обуката е познат како способност за генерализација. Задачата на конструирање на класификатор за препознавање примероци е позната како надгледувано учење или учење со учител (supervised learning) за разлика од задачата на ненадгледуваното учење (unsupervised learning) или кластерирање (clustering) каде ознаките (класите) дедуктивно се изведуваат од самите неозначени примероци во процесот на учење. Генерираниот модел може да се евалуира од повеќе аспекти како: точност, разбирливост, компактност, брзина на обучување и друго.

Концептот за учење (слика 1.1) претставува конструкција на модел за даден број на опсервации. Во процесот на обука се акумулира знаење од индивидуалните опсервации со помош на индукција, односно се воопштува на основа на повеќе примероци за обука. За да се конструира моделот мора да се примени индукција, односно да се генерализира (воопшти) на основа на индивидуалните опсервации и да се изгради модел кој ќе важи и надвор од специфичните услови на индивидуалните опсервации. Откако моделот е конструиран, со истиот може по пат на дедукција да се изведат нови примери кои не мора да биле меѓу опсервациите.



Слика 1.1. Концепт на учење

1.3.2 Дефиниции

Во областа на машинското учење се среќаваат следните поими:

Примерок или *моџира* претставува подредено множество од променливи. Секоја променлива исто така означена и како *ајтрибути*, *особина* или *обележје* прима вредности од некое предефинирано мно-

жество на можни вредности што се зависни од проблемот. Ова предефинирано множество на вредности се нарекува *домен* на обележјето. Во основа, обележјата по својот вид од аспект на доменот може да се поделат во три основни групи: бинарни обележја, номинални обележја и континуални обележја.

Бинарните обележја како што зборува и самото име можат да примат една од само две можни вредности (на пример: има/нема, да/не, 1/0). Номиналните обележја можат да примат вредност од преброиво множество на вредности (на пример боја: црвена, жолта, зелена, или високо, средно, ниско). Во множеството на номинални вредности може, но не мора да постои релација на подреденост. Номиналните обележја од аспект на подреденоста на доменот од кој примаат вредности може да се поделат на две подгрупи: номинални обележја со дефинирана подреденост и номинални обележја со недефинирана подреденост. Континуалните обележја примаат произволна вредност најчесто репрентирана како реален број евентуално ограничена во даден опсег, но со непреброиво многу можни вредности.

Една од променливите на примерокот е специјален атрибут означен како *целно обележје*. Ова обележје го опишува феноменот кој е од интерес, односно феноменот во однос на кој би сакале да предвидиме одредено однесување. Примерокот за кој целното обележје е познато се нарекува *означен примерок* (labelled example). Примерокот за кој целното обележје не е познато се нарекува *неозначен примерок* или примерок за препознавање (query example).

Кога целното обележје е континуална променлива проблемот на надгледувано учење е познат како *регресија* [42]. Регресијата претставува математички метод со помош на кој се изведува емпириска функционална зависност на независните променливи со зависната променлива (целното обележје) на основа на множество од експериментални податоци. Кога пак целното обележје е номинална или бинарна променлива проблемот на надгледувано учење е познат како *класификација*. Целното обележје со номинална вредност се нарекува уште и *класа*, асоцирајќи на ознаката на класата од преброивото множество на класи на која припаѓа примерокот.

Во процесот на машинското учење се употребуваат множества на примероци (datasets) кои во основа се делат на множество за обука и множество за тестирање односно проценка на перформансите на моделот. Множеството за обука претставува множество на означени примероци на основа на кое се гради моделот. Множеството за тестирање претставува множество на неозначени примероци со кои се тестира моделот. Кај некои видови на алгоритми за градење на модели или оптимизирање на некои од параметрите на моделот се употребува и т.н.

множесѝиво за валидација (validation set). Во текот на изградбата на моделот, со примероците од множеството за валидација се проценуваат перформансите на моделот користејќи примероци што не ги видел (не учествуваат во обучувањето) и на основа на добиените резултати се нагдуваат некои од параметрите на моделот со цел да се подобрат перформансите на моделот.

Модел (регресиски модел во случај на регресион проблем или класификатор во случај на класификациски проблем) претставува пресликување од множеството на неозначени примероци во множеството од доменот на целното обележје. Моделот се користи за предвидување на вредноста на целното обележје. Вообичаено целното обележје се нарекува и *зависна менлива*, а останатите обележја *независни менливи*. Кај проблемите на класификација целта на моделот е непознатиот примерок да го распореди во точно една од конечниот број класи.

Формално, задачата на обучувањето е да конструира „обучен“ систем (модел) користејќи го множеството за обука D со цел да го класификува (или предвиди) y како излез од вака обучениот систем $g(x)$, за непознат влез во системот x . Множеството за обука D содржи обележја $\{(x_i, y_i), i = 1, \dots, N\}$, додека множеството за проверка V содржи обележја $\{(x_i, y_i), i = 1, \dots, M\}$. Овде x_i претставува m -димензионален вектор кој е влез за системот, додека y_i е посакуваниот излез од системот. По конструирањето на моделот неговите перформанси се проценуваат на множеството V .

Задачата на обучениот систем е да го класификува или да го предвиди посакуваниот излез што е можно поточно. Кога посакуваниот излез е од континуална природа ($y_i \subseteq \mathbb{R}$) се работи за регресија. Ако пак излезот од системот е со дискретна вредност $y_i \in C = \{c_1, c_2, \dots, c_k\}$ каде C е множество на класи а c_i е ознака на класата, станува збор за класификација. Класификаторот всушност претставува функција $g(x)$ која врши пресликување од просторот на обележја во просторот на класи, $g(x) : F \rightarrow \{c_1, \dots, c_k\}$.

Нека веројатноста за појава на класата c_i ја означиме како $P(C = c_i)$. Оваа веројатност може да се изрази како односот на бројот на примероци кои се припадници на класата c_i и вкупниот број на примероци во дадената популација и се нарекува априорна веројатност и се означува како $P(c_i)$. Веројатноста пак дека се појавила класата c_i под услов да се појавил примерокот $X = x$ се нарекува постериорна веројатност и се означува како $P(C = c_i | X = x)$ или само $P(c_i | x)$ и е многу важна за класификацијата. Користејќи го баесовото правило можеме да ја изразиме постериорната веројатност преку функцијата на распределба на условната веројатност $p(x|c_i)$:

$$P(c_i | x) = \frac{p(x | c_i)P(c_i)}{p(x)}, \quad (1.1)$$

каде $p(x)$ ја означува функцијата на распределба на веројатност на случајната променлива X (примероците) и е дадена со:

$$p(x) = \sum_{i=1}^L p(x | c_i)P(c_i). \quad (1.2)$$

Тренер или обучувач е процедура која се користи за да се изгради моделот. Во контекстот на овој труд под тренер ќе се подразбира компјутерска програма која го гради моделот на основа на примероците од множеството за обука без интеракција со корисникот или какви било други предзнаења за природата на проблемот. Во поширока смисла, секој метод кој ги користи информациите од означени примероци за обука при конструкцијата на класификатор, вклучува учење.

Претпоставката на процедурата за обука е дека постои непозната функција која го изведува пресликувањето од независните променливи во зависната променлива. Генерално не е можно да се открие точната функција користејќи конечно множество на примероци. Во реалните проблеми, исто така, расположивите податоци од кои се изведува множество за обука содржат и шум и тоа како кај атрибутите така и кај целните обележја (кога некои од примероците се погрешно обележани).

Кај надгледуваното учење (или учење со учител), „учителот“ ја дава ознаката на класата или посакуваниот излез од системот и се обидува во процесот на учење параметрите на системот да ги нагоди така да грешката во класификувањето или отстапувањето помеѓу предвидената и посакуваната вредност биде што помало. Кај надгледуваното учење (или самоучење), не постои експлицитен учител и системот настојува самиот да ги открие „природните“ групи на влезните податоци и истите да ги групира во овие групи. Овој проблем е познат и како проблем на кластеризација. „Природното“ групирање е секогаш експлицитно или имплицитно дефинирано на некој начин во системот за кластеризација.

Со цел да се подобри точноста, ефикасноста или скалабилноста на класификацијата или предикцијата, може да се применат некои од следниве чекори на претпроцесирање на податоците кои ќе учествуваат во обучувањето.

„Чистење“ на податоците (data cleaning). Овој чекор се однесува на отстранување или намалување на шумови и третманот на обележјата кои недостигаат во некои примероци. Иако повеќето алгоритми за класификација и самите поседуваат некаков механизам за справу-

вање со недостасувачки податоци или податоци загадени со шум, овој чекор може дополнително да помогне во намалувањето на грешките при обучувањето.

Анализа на релевантност (relevance analysis). Многу од атрибутите во примероците можат да бидат *ирелевантни* за дадениот проблем или некои од нив дури и да бидат *редундантни*. Со соодветна анализа на релевантност може да се откријат и отстранат од примероците за обука оние атрибути кои се ирелевантни или редундантни, со цел да се намали димензионалноста на проблемот, да се упрости моделот и да се подобрат неговите перформанси. Во машинското учење овој процес е познат и како *селекција на обележја* (feature selection).

Трансформација на обележјата (Data transformation). Трансформацијата на обележјата опфаќа промена на скалата во која се репрезентираат атрибутите, што може да опфати и нивна дискретизација или аналогизација. Дискретизацијата подразбира замена на континуална величина со дискретна при што истата се дискретизира во определен број однапред определени вредности (на пример високо, средно, ниско). Аналогизацијата пак подразбира замена на дискретните обележја со соодветна аналогна вредност која соодветно ги репрезентира. Во трансформацијата на обележјата спаѓа и *нормализацијата*, која е особено корисна кога класификаторот е невронска мрежа, или друг класификатор кој вклучува мерење на растојанија (во одреден метрички простор) во процесот на обуката на класификаторот. Нормализацијата опфаќа промена на скалата во која се репрезентирани атрибутите, односно скалирање на сите вредности за даден атрибут така да нивните вредности се ограничат на одреден мал регион како на пример $[0, 1]$ или $[-1, 1]$. Нормализацијата, исто така, може да се однесува и на трансформација на обележјата на начин кој би обезбедил нивните вредности да задоволуваат определена распределба, како на пример средна вредност нула и/или единична варијанса.

1.3.3 Карактеристики на обучен систем

Прекомплексен систем може да биде обучен перфектно да ги препознава примероците од множеството за обука, но истиот може да покаже значително послаби перформанси на множеството за тестирање, односно на примероци кои не ги видел за време на обуката. Ваквата ситуација е позната како „*преобучување*“ (overfitting). Еден од проблемите при конструкцијата на моделот претставува и одредувањето на неговиот *капацитет*, кој треба да биде избран соодветно – да не биде предноставен за да не може да се разграничат класите, но не и прекомплексен со што би давал слаби резултати во препознавањето на нови примероци.

Од чисто практични причини често е потребно да се фаворизираат класификатори кои користат помал број на обележја, бидејќи истите обично водат кон поедноставни граници меѓу регионите и поедноставно обучување на класификаторите. Некогаш пак е клучно обележјата да бидат што поробусни и поотпорни на шум. Во практичните апликации исто така, може да биде потребно класификаторот да реагира бргу, да користи малку мемориски или процесорски ресурси или да може лесно да се имплементира во хардвер.

Некои од карактеристиките според кои може да се споредуваат и евалуираат класификациските и регресионите методи за предикција се:

Точноста ја изразува способноста на моделот правилно да го класификува новиот, претходно не виден примерок.

Брзината се однесува на комплексноста на пресметките (или бројот на чекорите) кои треба да бидат изведени при генерирањето или експлоатацијата на моделот.

Робусноста ја изразува способноста на моделот правилно да ги класификува изобличените примероци и примероците загадени со шум.

Скалабилноста ја претставува способноста ефикасно да се конструира модел за голем број на податоци (типично голем број на примероци во множеството за обука и/или голем број на обележја).

Интерпретабилноста се однесува на можноста за интерпретација на моделот, и обично значи репрезентација на знаењето и правилата за одлучување содржани во моделот во форма разбирлива за човекот.

1.3.4 Евалуација на обучен систем

Аналитичка евалуација на даден алгоритам за учење е многу тешко да се изведе. Соодветно, и анализа на обучен класификатор кој бил произведен од даден алгоритам за учење и дадено множество на обележани примероци, многу е тешко да се изведе аналитички. Затоа во практика најчесто се применува емпириска евалуација. Вообичаен метод за евалуација на класификатор е преку експерименти [107], со цел да се потенцираат важните релации меѓу неговото однесување и условите што влијаат на него. Притоа анализата може да биде од различни аспекти како: рата на грешка, интерпретабилност, компактноста, брзина на учење, брзина на класификација, скалабилност и др.

Најчесто, најрелевантна е *ратата на грешка* кој претставува проценка на разликата помеѓу бараниот излез од класификаторот (вистинската класа) и актуелниот излез за даден примерок. Ратата на грешка на класификаторот се пресметува како однос меѓу бројот на погрешно класификувани и вкупниот број на примероци. Аналогно, *точноста* на класификаторот се дефинира како однос меѓу бројот на точно класификувани и вкупниот број на примероци.

Важно е да се направи разлика помеѓу двете репрезентации на ратата на грешка. Првата репрезентација, означена како *грешка на редистрибуција*, ја проценува грешката над множеството за обука. Скоро сите алгоритми за учење во процесот на учење настојуваат да ја минимизираат оваа грешка. Втората репрезентација означена како *грешка на генерализација*, ја проценува грешката над независно множеството за тестирање составено од примероци кои не биле во множеството за обука. Во првиот случај се проценува колку добро во процесот на учење класификаторот го совладал презентираниот материјал. Во вториот случај се проценува колку добро го „разбира“ наученото, односно колку успешно ги препознава нововидените примероци. Карактеристиката на класификаторот која што вообичаено сакаме да ја знаеме е токму неговата грешка на генерализација, бидејќи тоа е грешката која можеме да ја очекуваме кога истиот ќе биде применет за класификација на нови примероци.

Интерпретируемост на даден модел ја означува можноста за интерпретација на знаењето добиено за време на обуката во форма прифатлива за човекот. Кај класификаторите, ова знаење најчесто се репрезентира како множество на некои „правила“ на основа на кои се врши класификувањето. Ова е особено важно доколку примената на класификаторот бара јасна репрезентација и „документирање“ на неговите акции во форма разбирлива за човекот (на пример во медицинската дијагностика).

Брзината на обука се однесува на времето потребно да се обучи даден класификатор. Често од особен интерес, не е само вкупното време потребно да се обучи даден класификатор, туку скалабилноста на ова време, односно зависноста на брзината на обуката од дадени параметри на моделот како бројот на примероци во множеството за обука, димензионалноста (бројот на обележја) на примероците, бројот на класи, комплексноста на проблемот што треба да се реши, и др.

Брзината на класификација го изразува просечното време потребно да се класификува одреден непознат примерок, како и варијациите во ова време, односно дали е исто за секој примерок или може да биде и различно за различни примероци и во кои граници варира.

Скалабилноста на класификаторот пред сè се однесува на промената на неговата комплексност во зависност од бројот на класи, бројот на обележја со кои се репрезентира секој примерок, бројот на примероци кои се употребуваат при обуката и др.

1.3.5 Проценка на перформансите на обучен класификатор

Проценка на перформансите на обучен класификатор се изведува со негово тестирање. Во фазата на тестирање, на обучениот класифика-

тор му се презентираат неозначени примероци и се бара од него да ги препознае, односно да ги распореди во некоја од класите. Резултатот на класификацијата се споредува со нивните вистински класи и се проценува успешноста на класификаторот. Ако со N_t се означи бројот на примероци кои класификаторот точно ги препознал, а со N_f бројот на примероци кои биле погрешно препознаени, тогаш проценетата точност на класификаторот е дадена како:

$$Rr = \frac{N_t}{N} \quad (1.3)$$

каде $N = N_t + N_f$ е вкупниот број на примероци презентирани на класификаторот во фазата на тестирање. Проценетата точност на даден класификатор е еквивалентна на веројатноста дека класификаторот правилно ќе класификува случајно одбран непознат примерок.

За реална проценка на перформансите на класификаторот, примероците кои му се презентираат во фазата на тестирање треба да бидат од независно множество, односно да не биле користени во фазата на обучување на класификаторот. Теоретски, за да се обучи и непристрасно да се процени одреден класификатор потребно е да се дефинираат две дисјунктни множества: множество од примероци за обука и множество од примероци за тестирање чии примероци би биле случајно избрани од големо, независно репрезентативно множество на примероци од иста популација. Потоа, класификаторот обучен на множеството за обука се употребува за класификување на секој од примероците од множеството за тестирање. Доколку бројот на примероци е доволно голем, односот на погрешно препознаените примероци во однос на сите препознавани примероци дава непристрасна проценка на ратата на грешка за дадениот класификатор.

Во практиката скоро секогаш е тешко да се обезбедат доволен број на означени примероци за обука и тестирање. Најчесто, во практиката располагаме само со ограничен број на означени примероци кои треба што подобро да се искористат и за обука и за тестирање на перформансите на обучениот класификатор. Доколку истите примероци се искористат и за обука и за тестирање на класификаторот, ќе се добие проценка на грешката на ресубституција која е премногу пристрасна проценка за перформансите на класификаторот и води кон нереален заклучок за можностите за генерализација на класификаторот. Голем број модерни класификатори, а особено ако не вклучуваат регуларизација или кастрење (grunning), евалуирани на множеството за обука даваат премногу оптимистичка проценка на ратата на грешка многу блиска до 0, скоро независно од комплексноста на множеството за обука.

За да се избегене нереалната оптимистичка проценка на грешката која се добива кога тестирањето на класификаторот се изведува со истите примероци со кои и е обучуван, во практика проценката се изведува со некои од методите за вкрстена валидација (cross-validation). Сите методи за вкрстена валидација се засноваат на основната идеја: множеството за обука и множеството за тестирање да немаат заеднички примероци. Затоа, во практиката методите за евалуација на перформансите на класификаторите, скоро секогаш вкупниот број на означени примероци со кои се располага го делат на две дисјунктни множества: множество за обука и множество за тестирање. Бидејќи во практиката најчесто се располага со единствено ограничено множество од означени примероци, се наметнува прашањето како да се изведе оваа поделба, со цел да се постигне успешна обука а потоа и непристрасна проценка на вака обучениот класификатор.

Наједноставниот начин за да се постигне ова е множеството на означени примероци уште на почетокот да се подели на две дисјунктни и независни множества (најчесто во однос 2:1 до 3:1). Поголемото од нив означено како множество за обука притоа ќе се употребува за обучување на класификаторот, а со другото означено како множество за тестирање ќе се проценат перформансите на изградениот модел. Овој метод е наречен „метод на задржување“ (holdout method). Ваквиот пристап трајно исклучува дел од примероците од фазата на обука, односно класификаторот се обучува со нешто помал број на примероци од вкупниот број на означени примероци со кои располагаме. Ваквиот пристап е применлив во практика во случаите кога имаме доволен број на означени примероци, и можеме да дозволиме дел од нив да се издвои само за тестирање.

Честопати во практиката располагаме со доста ограничен број на означени примероци, од кои ако се одвои дел за тестирање, бројот на преостанатите примероци за обука може значително да влијае на успешноста на обуката на класификаторот. Затоа кога се располага со релативно мал број на означени примероци или значително нерамномерна застапеност на класите, во практиката често се употребува методот на повеќекратната вкрстена валидација (n -fold cross-validation). Кај овој метод множеството со M означени примероци се дели на n еднакви дисјунктни подмножества со $N = M / n$ примероци во секое. Потоа, за да се процени генерализацијата на моделот во секоја од n -те итерации се исклучува различно подмножество од множеството и тоа се чува за проценка на перформансите на класификаторот кој се обучува на останатиот дел од примероците. Перформансите на системот се добиваат како средна вредност од измерените перформанси за секоја од поделбите.

Како специјален случај, ако бројот на групи n стане еднаков на бројот примероците во множеството M , односно се исклучува по само еден примерок при обуката на класификаторот и така обучениот класификатор потоа се тестира на единствениот исклучен примерок, станува збор за исклучи-еден (leave-one-out) тестирање. Ваквиот пристап дава и најпрецизна проценка на бајасот¹ и варијансата на моделот (види поглавје 2.4), но, често пресметковно е премногу комплексен за практично да се употреби.

Друг метод за проценка на перформансите кој креира нови множества е bootstrap методот [13][105]. Според овој метод се генерираат одреден број (обично помеѓу 50 и 200 [135]) множества за обука и тестирање со случајно одбирање со повторување од оригиналното множество означени примероци. Со овие множества се обучува и се тестира класификаторот повеќе пати и добиените резултати се усреднуваат. Покажано е дека проценката на перформансите добиени на овој начин е статистички еквивалентна на исклучи-еден методот [45]. Во **Табела 1.1** се дадени главните карактеристики на набројаните методи за проценка на перформанси на обучен класификатор.

Табела 1.1: Методи за проценка на перформанси на класификатор за множество со N означени примероци

Метод	Примероци за обука	Примероци за тестирање	Број на обучувања/ тестирања
Ресубституција	N	N	1
Задржување	$\frac{2}{3}N - \frac{3}{4}N$	$\frac{1}{4}N - \frac{1}{3}N$	1
k -крајина вкрстена	$N - N/k$	N/k	k
Исклучи-еден	$N - 1$	1	N
Bootstrap	$m < N$	N	$> 50, < 200$

1.3.6 Баесова грешка

Кога класификаторот носи одлука можеме да сметаме дека тоа повлекува одредена *загуба* (во точноста) која е нула во случај на правилна одлука и позитивна во случај на грешка во класификацијата. Не-

¹ статистичка грешка која се прави со систематско фаворизирање на одредени примероци над останатите

ка функцијата на загуба L_{kj} ја дава загубата кога примерок кој припаѓа на класата c_k е класификуван како да припаѓа на класата c_j . Генерално, грешката е матрица која ги содржи загубите за сите можни комбинации на вистински и доделени класи.

$$L_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases} \quad (1.4)$$

Изразот (1.4) е вообичаената дефиниција за функција на загуба кога не е можно прецизно да се процени загубата на различните можни грешки. Иако во изразот не е експлицитно наведено (поради едноставност) функцијата на загуба L_{ij} е функција од влезниот примерок X бидејќи нему доделената класа c_j е функција од X .

Ако сите примероци се од дадена класа c_k тогаш очекуваната вредност на функцијата на загуба е:

$$E[L_{kj} | C = c_k] = \sum_{j=1}^L L_{kj} \int_{R_j} p(x | c_k) dx, \quad (1.5)$$

бидејќи доделената класа c_j е функција од X . Со R_j е означен регионот од просторот на влезови во кој X припаѓа на класата c_j , а со $E[\cdot]$ е означена очекуваната вредност.

Очекуваната вредност на загубата ако примерокот е од која било класа, е наречен *ризик* и е изразен како:

$$\begin{aligned} R &= E[L_{kj}] \\ &= \sum_{k=1}^L E[L_{kj} | C = c_k] P(c_k) \\ &= \sum_{k=1}^L \sum_{j=1}^L L_{kj} \int_{R_j} p(x | c_k) dx P(c_k) \\ &= \sum_{j=1}^L \int_{R_j} \sum_{k=1}^L L_{kj} p(x | c_k) P(c_k) dx \end{aligned} \quad (1.6)$$

За да се минимизира ризикот, подинтегралната големина во изразот (1.5) треба да се минимизира за секоја вредност на x . Соодветно примерокот x треба да се распореди во класата c_j кога:

$$\sum_{k=1}^L L_{kj} p(x | c_k) P(c_k) < \sum_{k=1}^L L_{ki} p(x | c_k) P(c_k), \quad \forall i \neq j. \quad (1.7)$$

Ако за функцијата на загуба се земе т.н. еден-нула функција на загуба (1.4), условот (1.7) може да се напише како:

$$p(x|c_j)P(c_j) > p(x|c_i)P(c_i), \quad \forall i \neq j. \quad (1.8)$$

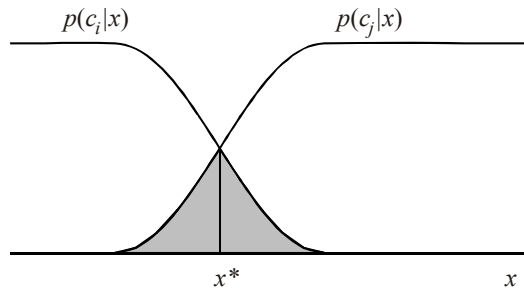
Со замена на дефиницијата за постериорна веројатност, овој израз може да се напише како:

$$p(c_j|x) > p(c_i|x), \quad \forall i \neq j. \quad (1.9)$$

Класификаторот кој го минимизира ризикот, притоа користејќи еден-нула функција на загуба (1.4), е наречен Баесов класификатор [121] и се сведува на:

$$g(X) = j \quad \text{ако} \quad j = \arg \max_{i=1,\dots,L} p(c_i|x). \quad (1.10)$$

Баесовиот класификатор е оптимален класификатор во смисла на минимизација на грешката и грешката што тој ја прави се вика Баесова грешка. На слика 1.2 сивиот регион ја претставува Баесовата грешка за еднодимензионален проблем. Баесовата граница x^* , која е оптимална граница помеѓу двете класи ја минимизира грешката при класификацијата.



Слика 1.2. Графички приказ на Баесовата грешка

Целото резонирање околу Баесовата грешка може да се повтори и за класификатор со опција за отфрлање (reject option). На класификаторите со опција за отфрлување им е дозволено да отфрлат (воопшто не класификуваат) одреден број на примероци кои според нив веројатно погрешно би ги класификувале [46][47]. Кај класификувањето со опција за отфрлање проблемот на класификација на N примероци се трансформира во проблем на класификација на $M < N$ примероци и отфрлање на оние кои се блиску до границите меѓу класите, односно оние примероци за кои постои голема веројатност при нивната класификација да се направи грешка. Од друг агол, класификувањето со опција за отфрлање може да се разгледува и како трансформација на оригиналниот

проблем на распоредување на непознатите примероци во k класи, во проблем на класификување во $k+1$ класа, каде во $k+1$ -вата класа се сместуваат примероците кои се отфрлаат, односно кои не може доволно сигурно да се класификуваат во некоја од k -те класи.

1.3.7 Димензионалност на класификатор

Основната цел на секој класификатор е да овозможи што поточно препознавање на непознатите примероци користејќи го моделот изграден на основа на конечен број на примероци за обука. Според ова, основната цел при обуката (градењето на моделот) на секој класификатор не е да ја научи егзактната репрезентација на примероците за обука, во кои сигурно постои и одреден шум, туку да изгради што поопшт статистички модел **на процесот** кој ги генерира податоците (примероците), односно да може успешно да генерализира.

При препознавањето примероци најчесто од примерокот се екстрахираат одреден број на обележја на основа на кои се гради моделот. Прашањето што се наметнува е колкав број на обележја да се употребат при градењето на моделот. Интуитивно е јасно дека со употреба на повеќе обележја би се постигнала попрецизна репрезентација на моделот. Меѓутоа, во практика е забележано дека со додавање на обележја над некој одреден број не само што повеќе не се подобруваат перформансите на класификаторот туку почнуваат и да опаѓаат. Вклучувањето на секое дополнително обележје, ја зголемува димензионалноста на проблемот за 1, додека комплексноста на моделот расте експоненцијално во однос на димензионалноста на обележјата. Ова повлекува бројот на примероци потребни за успешна обука на моделот исто така да расте експоненцијално. Ваквиот феномен е наречен „проклетство на димензионалноста“. Бидејќи во практика најчесто располагаме со ограничен број на примероци за обука, со зголемување на димензионалноста на репрезентациониот простор на обележјата, лесно може да се надмине граничната точка над која примероците ќе бидат премногу расеани во просторот за на основа на истите да може да се изгради конзистентна репрезентација на моделот. Различните видови на класификатори се различно осетливи на проблемот на димензионалноста на примероците, од линеарната регресија и фишеровиот класификатор како најосетливи до SVM како скоро потполно неосетливи.

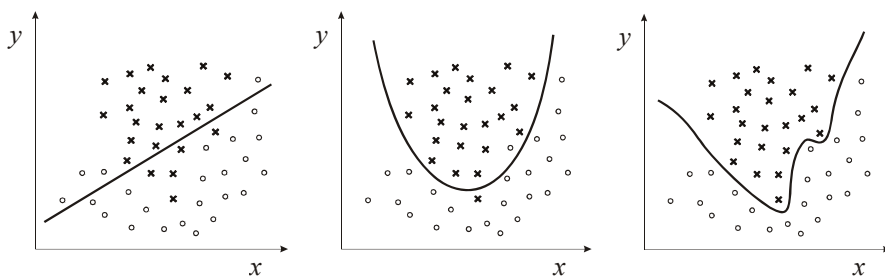
1.3.8 Капацитет на класификатор

И самите класификатори имаат своја скриена димензионалност, односно одреден број на слободни параметри (кои се нагудуваат при нивното обучување) на моделот кој го репрезентираат. Од нив директ-

но зависи и комплексноста на моделот кој може да го репрезентираат, односно комплексноста на границата меѓу регионите на класите која ќе може да ја репрезентираат. Кај некои класификатори како регресионите модели и еднослојните невронски мрежи, комплексноста на моделот е директно зависна од димензионалноста на обележјата, додека кај други се поставува при изборот на моделот најчесто како фиксна. На пример, кај повеќеслојниот перцептрон (Multilayer Perceptron – MLP) таа освен од димензионалноста на обележјата зависи и од бројот на неврони во скриените нивоа. Кај SVM, комплексноста воопшто не зависи од димензионалноста на обележјата, туку пред сè од употребениот кернел.

1.3.9 Оптимизирање на класификатори

За успешна генерализација неопходно е комплексноста на системот за класификација да биде избрана соодветно на комплексноста на проблемот кој треба да го решава. Изборот на преедноставен или прекомплексен модел доведува до намалување на перформансите на системот. Преедноставниот систем нема да може доволно добро да го репрезентира проблемот, додека прекомплексен систем ќе успее да ги научи и најмалите непожелни девијации како резултат на присутниот шум во примероците за обука што не се дел од моделот кој се репрезентира (слика 1.3).



Слика 1.3. Шематски пример за примероци во две димензии (x, y) кои припаѓаат на две класи (x, o). Полната линија ја прикажува границата меѓу класите изградена од класификатори со различен капацитет: а) линеарен, б) квадратен, в) од повисок ред

Од ова произлегува дека изборот на комплексноста на моделот е значајна за неговите генерализациони перформанси. За објаснувањето на овој феномен погодно е грешката при генерализацијата да се разложи на две компоненти: бајас (*bias*) или наклонетост и варијанса (*variance*) (види поглавје 2.4). Моделот кој е преедноставен односно недоволно флексибилен ќе има голем бајас, додека модел кој е прекомплексен за дадениот проблем ќе има голема варијанса. Бајасот и варијансата се комплементарни, односно со зголемувањето на едното се намалува

другото и спротивно. Најдобра генерализација се постигнува при најдобриот компромис помеѓу конфликтните барања за мал бајас и мала варијанса.

За да се пронајде оптималниот баланс помеѓу бајасот и варијансата мора да се располага со можноста да се контролира комплексноста на моделот. Комплексноста на моделот може да се нагодува на различни начини и зависи од самиот модел и алгоритмот за обучување. Кај невронските мрежи на пример, комплексноста може да се менува со менување на бројот на параметри на мрежата, што е уште наречено и *структурна стабилизација*. Еден начин да се постигне ова е да се испробаат повеќе модели со различен број на неврони во скриеното ниво. Друг приод е да се почне со модел со голем број на неврони, а потоа да се отстранат помалку значајните врски во мрежата или цели неврони. Постојат и приоди кои почнуваат со мала мрежа и во процесот на учењето додаваат неврони со цел да се постигне оптимална структура.

Вториот пристап за оптимизација на генерализационите перформанси на моделот е преку контрола на неговата *ефективна комплексност* со т.н. *регуларизација*. Ова може да се постигне со употреба на модел со поголем број на параметри кои може да се нагодат, и со промена на процедурата за обука со додавање на дополнителен член во функцијата на грешка со кој се изразува „казната“ при избор на прекомплексен модел. Ваквиот пристап не дозволува шумот во примероците за обука премногу да влијае на класификаторот, односно да се добие „помазна“ и „порегуларна“ одлучувачка функција.

Алтернатива на регуларизацијата како начин за контролирање на ефективната комплексност на моделот е процедурата за „рано прекинување“ на обуката. Обуката на голем број видови на класификатори се одвива во итеративна редукција на функцијата на грешка дефинирана над примероците за обука. За време на обуката оваа грешка генерално опаѓа како што бројот на итерации при обуката расте. Сепак, грешката проценета на независно множество на примероци за валидација, на почетокот опаѓа, но подоцна почнува да расте како што класификаторот станува пренагоден за конкретните примероци за обука. Од овој аспект, за да се постигнат подобри перформанси, обуката треба да престане во моментот кога се добиваат најдобри резултати мерено на независното валидациско множество, бидејќи од таквиот модел се очекува да има најдобра генерализација на непознати примероци.

Во практика, оптимизацијата на класификатор се изведува најчесто во неколку чекори кои се повторуваат:

1. Обука на класификаторот;
2. Испитување на перформансите на обучениот класификатор на независно множество примероци;

3. Нагодување на интерната структура или некои од параметрите на класификаторот и повторна обука (чекор 1) сè додека не се постигнат задоволувачки перформанси;

Притоа во фазата на обука на класификаторот, моделот се гради за дадено множество на примероци за обука. Меѓутоа, проценката на перформансите на даден модел, а особено на нелинеарните модели со значителен бајас, мора де се изведе на независно множество, односно на примероци кои не биле употребени во обуката. Бидејќи во практика најчесто се располага со ограничено множество на обележани примероци за обука, од истото множество треба да се земат и примероци за обука и примероци за проценка при оптимизацијата на класификаторот, се наметнува прашањето како да се поделат примероците. Слично како и кај поделбата на примероците за обука и тестирање (во поглавјето 1.3.5) и при оптимизацијата на моделот може да се употреби методот на повеќекратна вкрстена валидација, кој е пресметковно покомплексен. Наједноставниот начин е множеството на примероци наменети за обука уште на почетокот да се подели на две дисјунктни множества (најчесто во однос 2:1 до 3:1). Поголемото множество означено како множество за обука притоа ќе се употребува за обучување на класификаторот, а со помалото означено како множество за валидација ќе се проценуваат перформансите на изградениот модел и ќе се врши негова оптимизација. По определувањето на оптималните параметри на системот, истиот може конечно да се обучи користејќи ги сите примероци (вклучувајќи и ги и оние кои се користеле за валидација при оптимизацијата на моделот).

1.3.10 Поделба на класификаторите

Класификаторите според своите карактеристики како архитектура, дизајн, можности, начин на обука, интерна работа и друго, може да се поделат на повеќе групи од различни аспекти [165]. Во основа, задачата на секој класификатор е за даден непознат примерок претставен со своите обележја најчесто во вид на вектор од вредности x на неговиот влез, да го класификува или да го предвиди посакуваниот излез y што е можно поточно.

Во случајот на класификација бараниот излез е дискретна вредност $y \in C = \{c_1, c_2, \dots, c_k\}$ каде C е множество на класи, а c_i е ознака на класата во која класификаторот го класификувал непознатиот примерок. Според видот на вредностите со кои се претставени обележјата во векторот x , класификаторите можат да се поделат на класификатори кои на влезот примаат вектор од реални вредности (какви што се, или на што може да се сведат најголемиот број на класификатори кои се ко-

ристат во практиката) и на класификатори кои на влезот примаат бинарни вредности. Најголем број на класификатори на својот излез даваат повторно вектор од вредности на основа на кој се одредува дискретната вредност на класата во која се распоредува примерокот.

Според бројот на излезни вредности од класификаторот, класификаторите може да се поделат во две главни групи: бинарни (двокласни) класификатори и класификатори за повеќе класи. Кај бинарните класификатори постои само еден излез и истите непознатиот примерок може да го распоредат во точно една од двете можни класи. Кај класификаторите за повеќе класи најчесто постојат повеќе излези кои репрезентираат припадност на примерокот кон некоја (или повеќе) од n -те класи.

Од аспект на видот на излезите од класификаторот тие најчесто се со реални вредности, но кај некои класификатори можат да бидат и бинарни. Според ова бинарен класификатор со бинарен излез може да го класификува непознатиот примерок во една од двете класи, односно како припадник или не, на дадена класа без дополнителна информација за сигурноста на ваквата одлука. За разлика од ова, кај бинарниот класификатор со реален излез, вредноста на излезот може да се интерпретира како мерка на припадност кон класата. Пример за бинарен класификатор со бинарни влезови се адаптивните логички мрежи [2][3]. Бинарни класификатори со реални влезови се на пример машините со носечки вектори (SVM). Невронските мрежи се класификатори кои обично имаат повеќе реални излези, од кои секој обично претставува мерка на припадност на примерокот кон една од класите. Класификаторите по принцип на најблизок сосед (nearest neighbor classifiers), како излез го даваат растојанието до најблискиот сосед (или неколку најблиски соседи) и нивните класи, по што на примерокот му се придружува класата на неговиот најблизок сосед или онаа која најчесто се јавува меѓу неговите соседи.

Од аспект на проблемот кој треба да го решат, посакуваниот конечен излез од класификаторот е само класата на препознаениот примерок. Сепак голем број на постоечките алгоритми за класификација се во можност да снабдат и дополнителни информации околу класификацијата на примерокот, како на пример постериорната веројатност за негова припадност кон секоја од можните класи. Ваквата информација може корисно да се употреби особено при комбинирањето на повеќе класификатори или воведувањето на критериум за отфрлање на несигурно препознаените примероци.

Од гледна точка на комбинирањето на класификаторите од особен интерес е нивната поделба според бројот и видот на излези на класификаторот, односно информацијата што ја нудат на излезот при кла-

сификација на непознат примерок. Од аспект на нивото на информацијата која се добива од класификаторот според [165] класификаторите може да се поделат во три групи:

1. Апстрактно ниво – класификаторот ја дава само ознаката на класата во која го распоредил непознатиот примерок.

2. Подредено ниво - класификаторот дава редослед на сите (или дел) ознаки на класите кои може да ги препознава со ознаката на класата која е прв избор на врвот на подредена листа.

3. Квантитативно ниво - класификаторот за секоја класа од класите кои ги препознава дава квантитативна информација во вид на мера која го репрезентира степенот на припадност на примерокот на дадената класа.

Се разбира класификаторите од третата група со едноставно подредување на класите според степенот на припадност на примерокот кон секоја од нив може секогаш да се сведе на класификатор од втората група. Класификаторите пак од втората група со едноставен избор само на првиот кандидат се сведуваат на класификатори од првата група.

Зависно од начинот на кој се интерпретира излезната вредност кај класификаторите со квантитативен излез, класификаторите според [9] можат да се поделат во три основни групи:

1. Строги (сурови) класификатори (crisp) кај кои излезот има само две можни вредности и изразува припадност или неприпадност кон класата.

2. Непрецизирани класификатори (fuzzy, probabilistic) кај кои излезната вредност репрезентира непрецизирна припадност на примерокот кон определена класа најчесто претставена како реален број помеѓу 0 и 1.

3. Веројатносни класификатори (possibilistic) кај кои излезната вредност ја репрезентира веројатноста за припадност на примерокот кон определена класа.

Од аспект на проблемот кој го решаваат класификаторите постојат два основни вида на класификациски проблеми: дво-класен класификациски проблем и повеќе-класен класификациски проблем. Задачата на првиот е да определи дали даден објект е или не е припадник на одредена класа. На пример, задачата може да биде да се одреди дали објектот е припадник на класата леопарди. Кај дво-класната класификација втората класа автоматски произлегува од првата класа, односно за конкретниот пример таа би претставувала сè што не е леопард. Повеќе-класниот класификациски проблем претставува генерализација на дво-класниот класификациски проблем. Кај повеќе-класниот класификациски проблем на класификаторот му се дава непознатиот објект и множество на можни класи а задачата е да се распореди непознатиот објект

во една од дадените класи. Притоа се смета дека класите се дисјунктни, па ниту еден објект не може да биде припадник на повеќе од една од дадените класи.

1.3.11 Комбинирање на класификатори

Познато е дека резултатите добиени од повеќе независни класификатори можат ефикасно да се искомбинираат со цел да се добие поконцизна, покомплетна и поточна претстава за проблемот. Исто така, понекогаш класата на објектот не може да биде одредена по некој апсолутен критериум, туку зависи од мислењата на повеќе експерти. Во вакви случаи податоците треба да бидат независно класифицирани од страна на неколку експерти, а потоа нивните резултати да бидат некако комбинирани. Дури и ако некој од експертите понекогаш греша во класирањето, мнозинското гласање или некој друг систем на комбинирање на резултатите во групата веројатно ќе доведе до точен резултат. Комбинирањето на повеќе експерти во одлучувањето исто така носи и дополнителна информација за доверливоста (level of confidence) на одлуката донесена од групата. Ако сите експерти се сложиле во заклучокот, доверливоста на ваквата одлука е голема, ако пак мислењата биле поделени, доверливоста ќе зависи од распределбата на гласовите односно дисјунктноста на нивните мислења.

Често пати во практиката не е можно да се конструира единствен класификатор кој безгрешно (или со задоволителни перформанси) ќе ги препознава сите нови примероци. Исто така, било забележано дека различни класификатори кои имплементираат различни методологии, архитектури, пристапи или обележја, иако можеби со слични вкупни перформанси, можат да се однесуваат независно и да носат различни одлуки за ист примерок, односно да се надополнуваат во препознавањето. Токму оваа независност во одлуките на класификаторите навела на идејата дека со паралелна употребата на повеќе класификатори може да се подобрат перформансите при препознавањето. Комбинирањето на класификаторите исто така е и начин за намалување на варијансата на системот за класификација.

Поради интерните ограничувања на некои видови класификатори некои од нив не се во состојба самостојно да решат одреден вид на проблеми. На пример бинарните класификатори не можат самостојно да решат повеќе класен проблем. Сепак, повеќе бинарни класификатори може да се интегрираат во единствен систем кој ќе биде во состојба да решава и повеќе класни проблеми. При ова како посебен проблем се наметнува начинот на организација на еден ваков систем: како да бидат конструирани и обучени бинарните класификатори и како да бидат комбинирани нивните излези за решавање на повеќе класен проблем.

1.4 Невронска мрежа како класификатор

Вештачки невронски мрежи (*Artificial Neural Networks – NN*) или поедноставно невронски мрежи, кои уште се познати и под името поврзувачки модел (*connectionist model*) или паралелен дистрибуиран процесирачки модел, е најуспешната компјутерска парадигма која досега била предложена како алтернатива на класичната Фон Нојманова компјутерска архитектура. Инспирирана од структурата на биолошкиот нервен систем, една невронска мрежа се состои од множество на нелинеарни процесирачки единици – неврони или јазли кои оперираат паралелно. Поради добрите страни на ваквиот модел, како што се масивна паралелна структура, толеранција на шум и робусност, можноста за учење од примери, можноста на моделирање на нелинеарности и др., невронските мрежи имаат широка примена во разни области како што се класификација, оптимизација, пресликување и апроксимирање на функции [10].

Една невронска мрежа е специфицирана со својата мрежна топологија, карактеристиките на невроните и правилата за учење. Зависно од овие карактеристики постојат поголем број на разни варијанти на невронски мрежи. Обично за одредена проблематика, една варијанта на невронски мрежи може да постигне подобри карактеристики од другите, затоа треба внимателно да се избере најсоодветниот тип на невронска мрежа за дадена област.

Вештачките невронски мрежи претставуваат поврзана целина од едноставни процесирачки елементи (единици или јазли), чија функционалност е заснована на биолошкиот неврон. Способноста за процесирање на мрежата е содржана во интензитетот на меѓусебната поврзаност, т.е. во вредноста на тежините на тие врски, остварени преку процесот на адаптација или учење од множество на обучувачки примероци.

Секој единица извршува релативно едноставна работа, таа ги прима влезните сигнали од нејзините соседи или надворешните извори и ги користи да го пресмета излезниот сигнал кој е проследен до наредните единици. Друга задача на процесирачките единици е нагудување на тежинските фактори. Системот е паралелен и во тој случај повеќе процесирачки единици ги вршат пресметките во исто време.

Кај невронските мрежи може да се разгледуваат три вида на процесирачки единици: *влезни единици* кои ги примаат податоците надвор од невронската мрежа, *излезни единици* кои ги праќаат излезните податоци надвор од невронската мрежа, и *скриени единици* каде влезните и излезните сигнали остануваат внатре во мрежата.

За проблемите поврзани со препознавање примероци најчесто се користат невронски мрежи од типот на т.н. повеќеслоен перцептрон

(MLP). MLP се состои од повеќе нивоа (слоеви) од неврони од кои првиот (влезен) слој е составен од влезни единици, а последниот (излезен) слој е составен од излезни единици. MLP може произволно прецизно да апроксимира која било функција или да генерира каква и да е одлучувачка површина.

Во процесот на обука се изведува нагодување на параметрите, пред сè тежините на врските меѓу невроните во невронската мрежа. Ова најчесто подразбира постапка во која на невронската мрежа и се презентираат познати примероци и на основа на нејзиното однесување се врши нагодување на тежините. Најчесто користено правило за обука е правилото со повратна пропација на грешка (back-propagation) според кое грешките регистрирани на излезното ниво се пропацираат наназад низ мрежата и служат за нагодување на тежините на останатите нивоа. Главното ограничување на невронските мрежи обучени според правилото со повратна пропација на грешка е проблемот на локален минимум, според кој поради градиентното опаѓање на грешката постои реална можност алгоритмот да заглави во локален минимум.

1.5 Машини со носечки вектори (SVM)

Во изминативе неколку години машините со носечки вектори (Support Vector Machines – SVM или SV машини) станаа важна техника во машинското учење. За препознавање примероци, SV машините беа употребени за препознавање на изолирани ракописни цифри [30][127][128], препознавање на објекти [12], идентификација на говорник [126], детекција на лице на слика [111], категоризација на текст [82] и др. SV машините се успешно употребени и во бројни регресиони проблеми како апроксиматори на функции, предвидувачи на временски серии и др. Во повеќето примери способностите за генерализација на SV машините ги достигнуваат или забележително ги надминуваат можностите на споредбените методи.

Основните идеи кои го иницирале развојот на SV машините може да се набљудува од неколку аспекти како: бајас-варијанса компромисот, контролата на капацитет и пресовпаѓањето (overfitting), но основната идеја е истата како за повеќето проблеми на машинското учење: за дадена ограничена количина на податоци за обука да се направи „машина“ која ќе има најдобри можности за генерализација. Таа може да се постигне само ако се оствари вистински баланс помеѓу точноста на машината на даденото множество за обука и капацитетот на машината, односно способноста на машината да научи правилно да ги распознава сите примероци од множеството за обука без грешка. Машината со преголем капацитет е како ботаничар со фотографска меморија на кој кога

ќе му се покаже ново дрво ќе заклучи дека тоа не е дрво затоа што нема ист број на листови како примероците на дрва кои ги видел дотогаш. Машината со премал капацитет пак е како мрзливиот брат на ботаничарот кој сè што е зелено го прогласува за дрво. Очигледно ниту едниот од нив не генерализира на задоволителен начин. Истражувањата и формализирањето на овој концепт резултира со едно од најсјајните достигнувања во теоријата на статистичкото машинско учење во делата на Владимир Вапник [147][148].

Објаснување на теоријата на која почиваат SV машините бара познавање на исклучително широк спектар на области како: дуална репрезентација, простори на обележја, теорија на учење, теорија на оптимизација, алгоритми, минимизација на структурен ризик (structural risk minimization), Vapnik-Chervonenkis димензија, разбивачки функции (shattering functions), репродуцирачки кернел простори (reproducing kernel spaces) и др. Иако во повеќето од овие области сè уште постојат многу активни истражувачки активности, сепак постојат стабилни основи кои ја формираат базата на концептот на SV машините.

SV машините претставуваат обучен систем кој користи простор на хипотези со линеарни функции во многудимензионален простор на обележја, обучен со алгоритам за обука според теоријата на оптимизација, користејќи принципи изведени од теоријата за статистичко учење. Послободно зборувајќи, SV машините за класификација се обидуваат да најдат хипер-рамнина која го максимизира растојанието помеѓу позитивните и негативните примероци за класификацискиот проблем, истовремено минимизирајќи ги грешките над множеството за обука. Одличен вовед во принципите на SV машините е изложен во [22][32]. Решенијата засновани на концептот на SV машините, по нивното воведување во 1995 година, за само неколку години успеаја да ги надминат повеќето системи во најразлични апликации. Иако не постои теоретски доказ дека SV методите се подобри од останатите и лесно може да се конструира пример на кој SV машините би биле полоши од другите методи, во практика тие одлично функционираат и најчесто ги надминуваат останатите пристапи. Сепак, комплексноста на SV машините и при обуката и при препознавањето често е поголема од онаа на другите методи.

Класификаторите засновани на SV машини се во основа бинарни класификатори, и имаат единствен излез врз основа на кој се одредува дали примерокот е или не е припадник на одредена класа. Нивната обука се изведува со множество примероци од кои секој е обележан или како позитивен или како негативен примерок за класата за која се гради SV машината.

За решавање на линеано неодвоиви проблеми се применува пресликување во високодимензионален просотор во кој проблемот би бил линеарно одвоив со помош на т.н. кернел функција. Главната предност на SV машините е гарантираното пронаоѓање на глобалниот минимум (оптималната одвоива површина) во процесот на обуката.

Едно од најголемите ограничувања на SVM класификаторите е изборот на кернелот. Иако има одредени истражувања во правецот на користење на претходното знаење за проблемот во изборот на кернелот [129], сепак одбирањето на најдобриот кернел за даден проблем како и неговите параметри е сè уште поле на истражување. Еден оригинален алгоритам за одредување на оптималните параметри при обуката на машините со носечки вектори кој не зависи од иницијалните параметри и не бара нивно ограничување во даден опсег како и предложена постапка за забрзување на верификацијата (проценка на перформансите) на машините со носечки вектори во процесот на барање на оптимални параметри е дадена во [56]. Со нејзина примена, без забележлива загуба, процесот на изнаоѓање на оптималните параметри при обуката на SVM класификаторите може да се забрза за неколку пати.

Друго ограничување на SVM класификаторите е нивната спорост и при обучувањето и при препознавањето. Иако со декомпозиционите алгоритми за обука ова е делумно надминато, обучувањето со многу голем број на примероци е сè уште релативно бавно. Предложени се и различни пристапи за забрзување на препознавањето [21].

Дополнително ограничување на SVM класификаторите е тоа што тие природно се бинарни класификатори, односно решаваат двокласен проблем. Нивното проширување за решавање на повеќекласен проблем не е директно и едноставно како кај некои други видови на класификатори. Иако се направени обиди да се конструираат SVM класификатори кои одеднаш би се обучувале за повеќекласен проблем, во практика сè уште најчесто се употребуваат повеќе SVM бинарни класификатори за решавање на повеќекласен проблем.

1.6 Решавање на повеќекласни класификациски проблеми со SVM класификатори

SVM класификаторите се по природа бинарни класификатори, односно решаваат дали даден примерок е позитивен или негативен пример за даден концепт или едноставно дали примерокот е припадник или не на дадена класа. Но, многу практични проблеми се повеќекласни, односно постојат повеќе од 2 класи и задачата е да се одреди единствената класа на која припаѓа примерокот. Постојат два правци за решавање на повеќекласниот проблем со примена на SVM класификатори. Први-

от е да се прошират концептуално самите класификатори за да можат да решаваат и повеќекласни проблеми. Вториот правец е да се употребат повеќе бинарни SVM класификатори за да се реши единствен повеќекласен проблем. Додека првиот пристап е специфичен за SVM класификаторите и вклучува промена и проширување на теоретските претпоставки околу кои е развиен концептот на SV машините, вториот пристап е многу подиректен, поопшт и генерално важи за било кој тип на бинарен класификатор а не ексклузивно за SVM.

1.6.1 Повеќекласна класификација со единствена SV машина

Употребата на единствена SV машина за решавање на повеќекласен проблем е предложена истовремено независно од Вапник [148] и Weston и Watkins [156]. Според овој пристап, симултано се конструираат k (каде k е бројот на класи) делбени површини како единствен проблем за минимизација со нешто изменет начин на пресметување на казна за грешките што резултира во оптимизациски проблем од k пати повисок ред. Декомпозиција за нумеричко решавање на ваквиот проблем не е понудена. Lee et al. [100] предлагаат поинаков пристап заснован на асимптотското однесување на SV машините. Поинаков пристап предлагаат Bredensteiner и Bennett [17]. Пристапот на Crammer и Singer [31] е сличен на оригиналниот предложен од Вапник, но со одредени поедноставувања со цел да не се зголеми редот на оптимизацискиот проблем, при што е понуден и алгоритам за декомпозиција на ваквиот проблем.

1.6.2 Повеќекласна класификација со употреба на повеќе бинарни SV машини во еден-против-сите шема

Веројатно наједноставниот и најинтуитивниот начин за да се реши повеќекласен проблем со помош на бинарни класификатори е да се обучат k индивидуални бинарни класификатори да ги разликуваат примероците од една класа од примероците од сите останати класи. При препознавањето, непознатиот примерок се презентира на секој од k -те индивидуални бинарни класификатори и примерокот се распоредува во онаа класа чиј класификатор ќе даде најголем излез, односно ќе покаже најголема убеденост дека се работи за примерок од неговата класа. Ваквата шема е позната како еден-против-сите (one-vs-all) или еден-против-останатите шема. Иако е концептуално едноставна, еден-против-сите шемата е исклучително моќна, најчесто подобра и ретко незначително полоша од останатите шеми [120]. Нејзината главна предност е што k -те индивидуални бинарни класификатори може да се набљудуваат и како единствен повеќекласен класификатор со k излези, какви што

се повеќето други повеќекласни класификатори (невронските мрежи на пример).

Една од причините за преферирањето на еден-против-сите шемата е нејзината едноставна имплементација и релативна брзина при обуката и препознавањето. Обучувањето на еден-против-сите шемата подразбира креирање на k бинарни класификатори секој обучен со сите примероци до множеството за обука. Притоа секој од индивидуалните бинарни класификатори е одговорен за една од класите и се обучува со примероците од таа класа како позитивни примери и сите останати примероци како негативни примери. Според ова, обуката на ваквиот систем ќе трае околу k пати подолго од обуката на еден бинарен класификатор.

1.6.3 Повеќекласна класификација со употреба на повеќе бинарни SV машини во еден-против-еден шема

Друг метод за решавање на проблемот на класификација во повеќе од 2 класи со употреба на бинарни класификатори е т.н. еден-против-еден (one-against-one) или секој-против-секого (all-vs-all) пристап. Оваа стратегија за првпат е употребена со SVM класификатори во [44][93]. Според овој метод се конструираат $k(k-1)/2$ класификатори e_{ij} (по еден за секој пар различни класи) од кои секој се обучува на примероците од само две класи. При ова секој од класификаторите e_{ij} се обучува да разликува примероци од само две класи, односно да го решава бинарниот проблем на одвојување на примероците од класата i (позитивните примери) од примероците од класата j (негативните примери). Бидејќи проблемот на разграничување на класата i од j е симетричен, класификаторите e_{ij} и e_{ji} се комплементарни, односно нема потреба од креирање два одвоени класификатори. Ваквите класификатори при обуката ја одредуваат оптималната делбена хипер-рамнина со максимална маргина помеѓу примероците од двете класи, не земајќи ги притоа предвид останатите класи. Од вака обучените класификатори се очекува при препознавањето за примероците од едната класа (онаа која била сметана за позитивен пример при обуката) да даваат позитивни вредности, а за другата класа негативни. Примероците од останатите класи на кои не биле обучувани ќе ги класификуваат во една од овие две класи.

По конструирањето на класификаторите, постојат повеќе различни методи за начинот на нивната примена при препознавањето примероци. Веројатно наједноставниот, предложен во [44], е принципот на гласање, според кој примерокот кој се препознава се проследува до сите $k(k-1)/2$ класификатори. Доколку класификаторот e_{ij} даде позитивен излез се додава еден глас за класата i , а ако излезот е негативен гласот

се доделува на класата j . Потоа се одлучува за класата која собрала најголем број на гласови. Оваа стратегија е наречена и *сирашегија на најголем број на победи* (Max Wins). Во ваквиот пристап се користат само цврстите конечни одлуки на секој од бинарните класификатори, односно секој од одговорите се интерпретира како „примерокот припаѓа на класата i “ или „примерокот припаѓа на класата j “. Доколку класификаторите даваат и проценка на постериорната веројатност на класата, оваа информација би можела да се искористи при одредувањето на конечната класа на примерокот.

Hastie и Tibshirani [69] предлагаат постапка за одредување на постериорната веројатност на секоја од класите наречена *pairwise coupling*, на основа на добиените проценки на постериорните веројатности за бинарните класификациски проблеми од сите класификатори за паровите класи. Проблемот се сведува на следниов: за дадено множество на настани $A_1, A_2, \dots, A_M$, $M = k(k-1)/2$ експерти генерираат веројатности за настаните $r_{ij} = P(A_i | A_i \vee A_j)$. Кое е множеството на веројатности $p_i = P(A_i)$ кои се компатибилни со r_{ij} ? Во егзактна смисла проблемот нема решение. Бидејќи $P(A_i | A_i \vee A_j) = p_i / (p_i + p_j)$ и $\sum p_i = 1$, имаме предетерминиран проблем во кој $k-1$ слободни променливи треба да задоволат $k(k-1)/2$ ограничувања, што генерално нема решение. Hastie и Tibshirani предлагаат наоѓање на најдобрата апроксимација за вредностите $\hat{r}_{ij} = \hat{p}_i / (\hat{p}_i + \hat{p}_j)$ за даденото множество на r_{ij} минимизирајќи го Kullback-Leibler растојанието, во итеративна постапка и тоа за секој од примероците кој се препознава одделно [69].

Поинаков пристап за одредување на класата на непознат примерок употребувајќи $k(k-1)/2$ класификатори предлага Platt [114]. Според овој пристап бинарните SVM класификатори се распоредени како јазли на насочен ацикличен граф (Directed Acyclic Graph – DAG) со $k(k-1)/2$ внатрешни јазли и k листови. Непознатиот примерок се проследува прво на коренот на графот и во зависност од излезот од бинарниот класификатор во овој јазел примерокот се проследува на следниот класификатор по левата или десната врска. Истото се повторува со сите следни јазли на патеката сè додека не се стигне до лист кој ја одредува класата на примерокот. Според оваа постапка примерокот се проследува секвенцијално на точно $k-1$ класификатори наместо на сите $k(k-1)/2$ со што препознавањето е доста побрзо. Сепак и кај вака организираниот класификатор се користат само цврстите одлуки на бинарните класификатори, а и конечната одлука е само цврста одлука за припадност на примерокот кон една од класите. За разлика од Max Wins стратегијата која исто така ги користи цврстите одлуки на бинарните класификатори, но дава број на освоени гласови за секоја од класите од кои може

да се одреди втората кандидат класа, DAG пристапот на излезот дава единствена класа. Обучувањето на класификаторите и кај DAG пристапот е идентично како и кај Max Wins стратегијата.

Kindermann et al. [88] предлагаат примена на теоријата за корекција на грешки за повеќекласна класификација со SVM бинарни класификатори. Според предложениот пристап за повеќекласниот проблем се обучуваат $N > k$ класификатори и за секој од нив се одредува дали примероците од i -тата класа ќе бидат позитивни или негативни примери за него на основа на дадена $N \times k$ матрица која треба да поседува добри својства за корекција на грешки од аспект на теоријата на информации. При препознавањето примерокот се проследува на сите N класификатори и во зависност од излезите добиени од нив примерокот се распоредува во класата до која има најмало хамингово растојание. Од аспект на овој пристап и еден-против-сите и еден-против-еден стратегиите може да се разгледуваат како негови специјални случаи.

1.7 Конверзија на излезите од SVM класификаторите во веројатности

За голем број примени потребно е излезот од класификаторот при препознавање на непознат примерок да биде изразен како калибрирана постериорна веројатност, особено кога истиот треба дополнително да се обработува, интегрира со други податоци или кога чини само дел од целосната одлука како при комбинирањето на класификатори. Дури и во случајот на класификатор за повеќе класи, изборот на класата со најголема постериорна веројатност претставува баесово оптимално одлучување.

Излезот од SVM класификаторите е некалибрирана вредност која не претставува веројатност. Еден начин за добивање на веројатносни излези од SVM класификаторите е предложен од Wahba [151] кој користи логистичка функција:

$$P(y = 1 | x) = p(x) = \frac{1}{1 + e^{-f(x)}}, \quad (1.11)$$

каде $f(x)$ е некалибрираниот излез од SVM класификаторот. Друг начин е да се прилагодат излезите од SVM класификаторот да ја следат гаусовата распределба на условните распределби на веројатност $p(f | y = 1)$ и $p(f | y = 0)$. Вакво решение предлагаат Hastie и Tibshirani [69] при што користат единствена варијанса за обете распределби што може да го деградира пресликувањето. Пофлексибилен е пристапот во кој одвоено би се одредиле параметрите на двете гаусови распределби $p(f | y = \pm 1)$,

проценети на множеството за обука или уште подобро на независно множество за валидација. Потоа баесовото правило може да се искористи за добивање на постериорната веројатност:

$$P(y = 1 | f) = \frac{p(f | y = 1)P(y = 1)}{\sum_{i=+1,-1} p(f | y = i)P(y = i)}, \quad (1.12)$$

каде $P(y = i)$ се априорните веројатности кои може да се проценат од множеството за обука. Според ваквата формулација постериорните веројатности претставуваат аналитичка функција од f која има форма:

$$P(y = 1 | f) = \frac{1}{1 + \exp(af^2 + bf + c)}. \quad (1.13)$$

Проблемот со вака добиената функција е што таа не е строго монотона, а исто така и претпоставката дека распределбите на густините на веројатност на излезите од SVM класификаторите може да се моделираат како гаусови не мора секогаш да е точна.

Platt [113] предлага употреба на параметарски модел кој директно ја моделира постериорната веројатност $P(f | y = 1)$. Параметрите на моделот се адаптираат со цел да даваат најдобра проценка на постериорните веројатности. Функцијата за пресликување од SVM излезите во постериорни веројатности која ја предлага Platt е параметарска форма на сигмоида:

$$P(y = 1 | f) = \frac{1}{1 + \exp(Af + B)}. \quad (1.14)$$

Параметрите A и B се одредуваат во процес на минимизација на отстапувањата со употреба на Levenberg-Marquardt model-trust алгоритмот за оптимизација и користење на примероците од множеството за валидација [113].



Комбинирање на класификатори

Суштинската цел при дизајнирањето на секој систем за препознавање примероци е да се постигнат што подобри перформанси при класификацијата. За постигнувањето на оваа цел, во практиката често пати се развиваат повеќе различни пристапи за решавање на конкретниот проблем. Потоа, на основа на проценетите перформанси од експериментите спроведени над различните дизајни, се избира еден од нив како конечно решение за проблемот. При ваквите испитувања на различни дизајни забележано било дека иако еден од дизајните има најдобри перформанси, множествата на примероци што се неправилно класификувани од различните класификатори не секогаш потполно се преклопуваат, односно различните класификатори прават грешки на различни примероци. Ова навело за заклучокот дека различните класификатори потенцијално нудат комплементарни информации за примероците кои треба да се класификуваат, и истите би можеле ефективно да се употребат за подобрување на перформансите при класификувањето.

Ваквите опсервации го мотивираа интересот на повеќе истражувачи кон истражувањето на комбинирањето на класификатори во последниве десетина години. Главната идеја во комбинирањето на класификатори е при донесувањето на конечната одлука да не се потпремена единствена шема за одлучување, туку да се искористат сите дизајни или дел од нив со комбинирање на нивните индивидуални мислења. Идејата за комбинирање на повеќе класификатори со цел да се компензираат индивидуалните слабости а да се задржат индивидуалните пре-

дности на секој од нив, последниве десетина години опсежно е истражувана во областа на препознавањето примероци. Показано е дека со соодветно комбинирање на резултатите добиени од различни експерти според одредено правило, може да се добијат перформанси подобри од перформансите на кој било од индивидуалните експерти. По охрабрувачките прелиминарни резултати, истражувањата биле сконцентрирани кон испитување на комплементарната природа на експертите кои треба да се употребат, нивниот оптимален број и оптималниот начин на комбинирање [165][76][91]. И покрај тоа што во литературата последниве години беа предложени голем број на различни пристапи и шеми за комбинирање на класификатори, сè уште не е потполно јасно зошто некои од нив се подобри од други и во кои услови.

Во ова поглавје се разгледани принципите на комбинирање на класификатори, како и методите за добивање на класификатори погодни за комбинирање. Претставени се повеќе методи за комбинирање на класификатори, теориските основи на кои почиваат и начините за нивна реализација.

2.1 Традиционален пристап за препознавање примероци

За жал досега не е пронајдена доминантна архитектура која би ги покрила сите можни распределби на примероци, а распределбата за проблемот кој треба да се реши во практика најчесто не е позната (теорема „нема бесплатен ручек“ – no free lunch) [158][159][160]. Традиционалниот пристап при практичното решавање на проблеми од областа на препознавање примероци се базира на евалуација и избор на класификатор: *евалуација* на множество на различни класификациски алгоритми или нивни верзии на множество од репрезентативни примероци и *избор* на најдобриот. Според ваквиот традиционален пристап:

1. Се дизајнираат N класификатори: $C_1, C_2, \dots, C_N$;
2. Се евалуираат грешките $E_1 < E_2 < \dots < E_N$ на класификаторите употребувајќи множество на примероци за валидација;
3. Се избира класификаторот C_k кој прави најмалку грешки и се смета за „оптимален“ (во баесова смисла на пример);

Традиционалниот пристап релативно добро функционира кога на располагање имаме големо и репрезентативно множество на примероци, така што проценетите грешки овозможуваат избор на најдобриот класификатор. Во практика, поради ограничениот број на примероци за валидација и нивната поинаква распределба проценетата грешка на класификаторите се разликува од вистинската. Ова може да го направи из-

борот на оптималниот класификатор невозможно, а во најлош случај постои можност и да се избере најлошиот класификатор.

2.2 Системи за препознавање примероци со повеќе класификатори

Во минативе десетина години предложени се повеќе методи за комбинирање на повеќе класификатори при дизајнирањето на системи за препознавање примероци со високи перформанси [91][165]. Основата на растечкиот интерес за истражување во областа на системите со повеќе класификатори (multiple classifier systems - MCS) е што класичниот пристап за дизајнирање на систем за препознавање примероци кој се концентрира на изнаоѓање на најдобар можен единствен класификатор има сериозни недостатоци [90]. Главниот недостаток е што најдобриот индивидуален класификатор за даден проблем е многу тешко да се пронајде, освен ако не располагаме со големо предзнаење за самиот проблем [41] што за жал е најчестиот случај во практика.

Во литературата комбинирањето на класификатори се среќава под различни имиња:

- комбинирање на повеќе класификатори (combination of multiple classifiers) [92], [98], [122], [162], [165]
- фузија на класификатори (classifier fusion) [14], [29], [46], [65], [84]
- мешавина од експерти (mixture of experts) [79], [80], [83], [108]
- комитети од невронски мрежи (committees of neural networks) [10], [38]
- консензусна агрегација (consensus aggregation) [7], [8], [109]
- гласачко тело од класификатори (voting pool of classifiers) [4]
- динамички избор на класификатор (dynamic classifier selection) [162]
- композитен систем од класификатори (composite classifier system) [33]
- ансамбли од класификатори (classifier ensembles) [38], [43]
- подели-и-владеј класификатори (divide-and-conquer classifiers) [27] и други.

Грубо зборувајќи, систем со повеќе класификатори вклучува ансамбл од повеќе различни класификатори и функција на одлучување која ги комбинира (обединува) излезите од сите класификатори. Соодветно, и дизајнирањето на систем за комбинирање на класификатори природно се состои од две главни фази: дизајнирањето на ансамблот од класификатори и дизајнирање на функцијата за комбинирање. Иако ваквата дефиниција на систем со повеќе класификатори наведува на

заклучокот дека ефективниот дизајн треба да ги опфати паралелно обе-те фази, поголемиот број методи за дизајнирање на систем со повеќе класификатори објавени во литературата се фокусираат само на едната од нив. Методите за креирање на ансамбли од класификатори најчесто претпоставуваат едноставна фиксирана функција за комбинирање и се трудат да генерираат множество на меѓусебно независни и комплементарни класификатори кој би постигнале оптимални перформанси за даденото правило за комбинирање [91][132]. За да се постигне ова најчесто се користат техники за „одбирање“ [18] кои обезбедуваат секој од различните класификатори да биде обучуван на различно подмножество од примероците за обука. Од друга страна, методите кои се фокусираат на дизајнот на функцијата за комбинирање подразбираат расположливост на множество од внимателно дизајнирани класификатори и се концентрираат на изнаоѓање на оптимална функција за комбинирање на одлуките (излезите) од овие класификатори.

Некои истражувачи [50][54] го предлагаат т.н. „направи многу па избори“ (overproduce and choose) метод за дизајнирање на системи со повеќе класификатори кој се сведува на „процени и избори“ принципот [133]. Основната идеја на овој пристап е да се произведе иницијално големо множество на „кандидат“ ансамбли од класификатори, а потоа да се избере оној со чие комбинирање на класификаторите се добиваат оптимални перформанси [123].

Иако некои од предложените методи за дизајн на системи со повеќе класификатори се покажаа како доста ефективни и по истражувањата на компаративните предности на различните методи [72], сè уште не постојат јасни правци (препораки) за избор на најдобриот метод за даден проблем, со што дизајнирањето на системи со повеќе класификатори е сè уште отворено прашање.

2.3 Ансамбл од класификатори

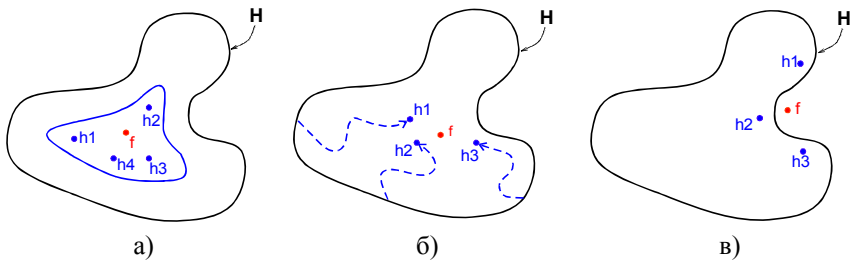
Ансамбл од класификатори претставува множество од класификатори чии индивидуални одлуки се комбинираат на некој начин (се обединуваат во единствена одлука) со цел подобро да се класификуваат непознатите примероци. Главното откритие во областа на комбинирањето на класификатори е што комбинацијата (ансамблот) е најчесто многу подобра од индивидуалните класификатори од кои е составена.

Потребен и доволен услов за ансамблот од класификатори да биде подобар од кој било од индивидуалните членови е индивидуалните класификатори кои го сочинуваат да бидат *прецизни* и *различни* [67] [1]. Прецизен класификатор е класификатор чија рата на грешка е помала од случајниот избор на класата на непознатите примероци. Два кла-

сификатора се различни доколку прават различни грешки за непознатите примероци. Доколку индивидуалните класификатори прават некорелирани грешки со рати помали од 0.5, тогаш и нивната наједноставна комбинација (мнозинско гласање) не е полоша од најлошиот од нив. Одовде клучот за успешно комбинирање на класификаторите е да се конструираат индивидуални класификатори со рата на грешки помала од 0.5 чии грешки ќе бидат барем делумно некорелирани. Покрај оваа формална дефиниција на проблемот се наметнува прашањето дали вакво нешто е изводливо во практика. За среќа, најчесто конструирањето на успешни ансамбли од класификатори е изводливо, поради најмалку три причини [35].

Првата причина е статистичка. Најголемиот број на алгоритми за обука може да се набљудуваат како пребарување на просторот на хипотези со цел да се пронајде најдобрата хипотеза во дадениот простор. Статистичкиот проблем настапува кога имаме премалку податоци за обука во однос на големината на просторот на хипотези. Без доволно примероци за обука алгоритмот за обука може да пронајде повеќе различни хипотези кои имаат исти перформанси над множеството за обука. Со креирањето на ансамбл од овие класификатори со иста прецизност, алгоритмот за комбинирање може да ги упросечи нивните одговори и да го намали ризикот од избор на погрешниот класификатор. На слика 3.1а графички е прикажана оваа ситуација. Надворешната крива го претставува просторот на хипотези, додека внатрешната крива го претставува множеството на хипотези кои имаат иста точност над множеството за обука. Ако со точката f е означена вистинската хипотеза, може да се види дека со упросечување на неколку прецизни хипотези, може да се пронајде добра апроксимација на f .

Втората причина е од пресметковна природа. Многу од алгоритмите за учење работат на принципот на некоја форма на локално пребарување кое многу лесно може да заглави во некој од локалните оптимуми. На пример, невронските мрежи употребуваат техника на намалување на градиентот (gradient descent) при минимизирање на функцијата на грешка при обучувањето над множеството од примероци за обука, додека дрвата за одлучување користат „лакоми“ (greedy) алгоритми при градењето на стеблото. Дури и во случаите кога располагаме со доволно примероци за обука (со што статистичкиот проблем би бил надминат) сè уште може да биде пресметковно многу тешко да се изнајде најдобрата хипотеза (обуката на невронските мрежи и дрвата за одлучување се NP тешки проблеми [15]). Ансамблот конструиран од повеќе класификатори кои се добиени со пребарување од различни почетни точки можат да дадат подобра апроксимација на непознатата целна функција одколку кој било од индивидуалните класификатори (слика 2.1б).



Слика 2.1. Причини зошто ансамблот од класификатори може да биде подобар од единствен класификатор. а) Статистичка причина; б) Пресметковна причина; в) Репрезентатиска причина.

Третата причина е од репрезентатиска природа. Во повеќето апликации на машинското учење вистинската целна функција не е можно прецизно да се репрезентира со која било од хипотезите од H . Но, со комбинирање на повеќе хипотези од H , можно е да се прошири просторот на репрезентатиски функции (слика 2.1в). Репрезентатиското ограничување е доста суптилно, бидејќи за голем број алгоритми просторот на хипотези ги опфаќа сите можни класификатори. Ваквите алгоритми теоретски при неограничено множество на примероци за обука го пребаруваат целиот простор на хипотези во потрага за најдобрата хипотеза. Но, во практика кога се располага само со ограничено множество на примероци за обука ваквите алгоритми го прекинуваат пребарувањето во моментот кога ќе најдат хипотеза која одговара на податоците за обука. Соодветно на слика 2.1в е претставен случај кога просторот на хипотези е само ефективен простор изграден на основа на ограниченото множество на примероци. Овие три причини се фундаменталните причини поради кои денешните алгоритми за учење „паѓаат“. Комбинирањето на класификатори ветува намалување (ако не и елиминирање) на ефектите на овие три клучни недостатоци на стандардните алгоритми за учење.

Инхерентната редувантност присутна кај ансамблите од класификатори може да се разгледува и како еден вид заштита од индивидуалните грешки на поединечните класификатори. Комбинирањето на несовршени класификатори, исто така, може да се разгледува и како начин за надминување на нивните индивидуални ограничувања и слабости. Иако е познато дека индивидуалните класификатори прават грешки, фактот дека примероците кои погрешно ги класификуваат не секогаш се истите [92], наведува на заклучокот дека употребата на повеќе класификатори може да го подобри препознавањето на примероците. Комбинирањето на класификаторите на начин кој би го минимизирал ефектот на поединечните грешки на индивидуалните класификатори може да се покаже како корисно. Покажано е [143] дека грешката

на ансамбл од класификатори (E_{ens}) опаѓа со зголемувањето на бројот на класификаторите (K). Исто така грешката на ансамблот опаѓа и со намалувањето на корелираноста меѓу индивидуалните класификатори (ρ) како и со грешката на индивидуалните класификатори (E_{err}):

$$E_{ens} = \frac{1 + \rho(K-1)}{K} E_{err} + E_{bay}, \quad (2.1)$$

каде E_{bay} е баесовата грешка (дефинирана во поглавјето 1.3.6). Одовде се гледа дека доколку класификаторите се исти, односно носат исти одлуки за исти примероци што значи дека се крајно корелирани ($\rho = 1$), грешката на ансамблот е иста со грешката на индивидуалните класификатори. Ако класификаторите се потполно некорелирани ($\rho = 0$), грешката опаѓа со зголемувањето на бројот на класификатори вклучени во ансамблот.

Ефектот на комбинирање на редундантни класификатори може да се анализира и од аспект на статистичките карактеристики на класификаторите: бајас и варијанса. Имено, грешката која ја прави секој класификатор може да се изрази како сума на две големини: квадрат од бајасот и варијанса [10], наречено уште и декомпозиција на грешка.

2.4 Декомпозиција на грешката

При градењето на класификатор (или претскажувач во општа смисла), врз основа на множеството примероци за обука $(x_1, y_1), \dots, (x_m, y_m)$, претскажувачот се обучува да репрезентира функција $f(x)$ која ќе ја апроксимира вредноста на y за потполно нови опсервации на x . Ако наместо со $f(x)$ функцијата која ја репрезентира претскажувачот ја означиме со $f(x; D)$ со цел да се изрази нејзината зависност од множеството примероци за обука D [119], тогаш средната квадратна грешка на функцијата која ја репрезентира претскажувачот f може да се напише како:

$$E_D[(f(X; D) - E[y | x])^2], \quad (2.2)$$

каде E_D е операторот за очекуваната вредност за даденото множество на примероци за обука D , а $E[y | x]$ е целната функција. Истиот израз по одредени математички модификации може да се напише во форма:

$$E_D[(f(X; D) - E[y | x])^2] = \underbrace{(E_D[f(X; D) - E[y | x]])^2}_{bias} + \overbrace{E_D[(f(X; D) - E[f(x; D)])^2]}^{variance} \quad (2.3)$$

разложен на збир од две вредности: бајас² и варијанса. Бајасот ја мери разликата меѓу употребениот и најдобриот претскажувач, додека варијансата ја изразува вредноста за која проценките се разликуваат од просекот. Бајасот на даден претскажувач може да се интерпретира како мерка на неговата способност правилно да генерализира над одредено множество на примероци за обука, откако ќе биде обучен. Варијансата пак може да се интерпретира и како мерка за осетливоста на неговите излези во зависност од множество на примероци со кои е обучен, односно во која мера се разликуваат резултатите при препознавањето во зависност од изборот на различни примероци за обука.

При обуката на класификаторите секогаш мора да постои компромис помеѓу овие две конфликтни вредности кои ја сочинуваат вкупната грешка, бидејќи со намалувањето на едната компонента, другата расте и обратно. Подобрувањето на перформансите со комбинирањето на класификатори обично повеќе се должи на намалувањето на варијансата отколку на бајасот. Ова се должи пред сè на фактот што ефектот на упросечување во ансамблиите е да се намали варијансата, а притоа да не се промени бајасот. Соодветно, за ефективен пристап при комбинирањето на класификатори се смета обучувањето на класификаторите да се изведе со минимизирање на бајасот, додека евентуално зголемената варијанса на секој од нив ќе може да се отстрани при комбинирањето. Од овој аспект, комбинирањето на класификатори може да се гледа како метод со кој може да се заобиколи или барем да се намали потребата од компромис меѓу бајасот и варијансата при креирањето на системи за класификација.

Ансамблиите од класификатори кои имаат поголема варијанса се очекува да имаат и помала корелација на грешките. Една од главните детерминанти на ефективност на комбинирањето на класификатори е мерката за независност на грешките на индивидуалните класификатори, односно во која мера тие прават различни грешки [92]. Главниот проблем при комбинирање на класификатори е што во практика нивните излези најчесто се доста корелирани и зависни [79]. Во идеални услови, ансамблот е множество од класификатори чии грешки не коинцидират, односно, секој од класификаторите има добра генерализација (низок бајас), а кога греши, грешките не ги прави истовремено со некој од другите класификатори.

Имајќи ги предвид овие факти, се наметнуваат две главни прашања при градењето на системи кои комбинираат класификатори: како да се направат индивидуалните класификатори и како да се изведе нивно-то комбинирање. Во продолжение ќе бидат дадени некои најчесто употребувани пристапи што нудат одговор на овие прашања.

2.5 Методи за креирање на класификатори за комбинирање

Бидејќи главната причина за комбинирање на класификатори во ансамбл е да се зголемат вкупните перформанси, јасно е дека нема да има никакво подобрување во перформансите доколку ансамблот е составен од идентични класификатори, односно класификатори кои генерализираат на потполно ист начин. Значи, целта е да се добијат класификатори кои ќе генерализираат различно, односно ќе бидат комплементарни. Постојат различни параметри на самите модели на класификаторите или во алгоритмите и начините на нивна обука со кои може да се манипулира со цел да се добијат класификатори кои различно ќе генерализираат. Некои од методите се независни од употребениот вид на класификатори, додека некои зависат од самите класификатори или се тесно поврзани со начинот на нивното градење и обука. Во литературата се предложени повеќе начини за добивање на комплементарни класификатори кои главно почиваат на:

- употреба на дизајнерското познавање за проблемот или моделот;
- вметнување (додавање) на случајност;
- варијации во типот на класификаторот, неговата архитектура или параметри;
- манипулирање со примероците за обука;
- манипулирање со обележјата;
- манипулирање со излезите од класификаторот.

Во продолжение ќе бидат разгледани некои од начините кои најчесто се користат во практика.

2.5.1 Употреба на дизајнерското предзнаење за проблемот или моделот

Во случаите кога постои одредено познавање за природата на проблемот кој се решава, може да се дизајнираат комплементарни модели или алгоритми користејќи го ова предзнаење. Вакви се случаите во апликациите кои природно вклучуваат повеќе сензори или пак во апликациите каде природно постојат различни репрезентации на примероците (на пример статистичка и структурна репрезентација). Исто така, ако постои соодветно предзнаење за проблемот, истото може да се искористи за промена на типот на класификаторот, архитектурата или параметрите со цел да се добијат комплементарни класификатори. Овие пристапи се хеуристички и не гарантираат добивање на комплементарни класификатори, односно резултатот може да биде онолку добар колку што е добро и предзнаењето за проблемот.

2.5.2 Вметнување на случајност

Наједноставните методи за дизајнирање на комплементарни класификатори се потпираат на вметнување на случајност во процесот на класификација или обука на класификаторите. Оваа техника е применлива само кај нестабилните класификатори кај кои и мала промена во почетната состојба или параметрите на класификаторот може да доведат до потполно поинаква генерализација на добиениот модел. Кај невронските мрежи на пример, алгоритмот за обука поаѓа од различни (случајно избрани) појдовни точки (почетни тежини) со што се генерираат различни модели. Кај дрвата за одлучување пак, правилото за одлучување во секој од внатрешните јазли може да биде случајно избрано од неколкуте најдобри правила. И пристапите базирани на вметнување случајност во моделите, се хеуристички и не гарантираат дека добиените модели ќе бидат комплементарни, иако во практика најчесто функционираат.

2.5.3 Варијации во типот на класификаторот, неговата архитектура или параметри

Со промената на типот на класификаторот (невронска мрежа, стебло за одлучување, најблизок сосед, SVM, ...) користејќи ги истите примероци за обука се добива и различна генерализација. Промената на интерната архитектура на класификаторот (на пример бројот на неврони во скриеното ниво кај невронските мрежи) генерира во основа класификатори со различна генерализација. Исто така, промената на параметрите на моделот (како активациските функции во невронските мрежи) или алгоритмот со кој е извршено обучувањето се добиваат модели со поинаква генерализација.

2.5.4 Манипулирање со обележјата

Кај овој метод различните класификатори користат различни обележја екстрахирани од примероците. На пример, подмножества обележја добиени од различни сензори или обележја добиени со употреба на различни алгоритми, како и методи за мануелна или автоматска екстракција и селекција на обележја може да се употребат за генерирање на комплементарни класификатори. Се очекува дека набљудувајќи го проблемот од различни агли (претставен со различни обележја) ќе се добијат независни проценки за него, односно класификаторите обучени со ваквите обележја ќе бидат независни. Со овој метод ефективно се намалува бројот на обележја, со што може да се намали и можноста за успешно класификување на примероците опишани со помал број на обележја. Од друга страна класификаторите со помал број на влезни

обележја обично имаат и помал број на слободни параметри (како невронските мрежи на пример) и може да се очекува дека полесно ќе се обучат и со помал број на примероци. Овој метод е независен од класификаторите и дава приближно подеднакво комплементарни класификатори независно дали се употребуваат зависни или независни класификатори. Авторот Но [71] предлага техника за случајно избирање на подмножества обележја наречена *random subspace method* според која од просторот на обележја се генерираат различни множества за обука со „проектирање“ на обележјата.

2.5.5 Манипулирање со излезите од класификаторот

Под манипулирање со излезите од класификаторите се подразбира градење на комплементарни класификатори со поделба на множеството на класи на различни начини. При овој пристап секој од компонентните класификатори се обучува да решава под-проблем на класификување во $M < N$ класи. Се разбира, потребен е и соодветен метод за комбинирање на ваквите класификатори за да се реконструира оригиналниот проблем на класификација во N класи. Овој проблем автоматски се наметнува кога N -класен проблем треба да се реши со употреба на бинарни класификатори (како SVM).

2.5.6 Манипулирање со примероците за обука

Едни од најчесто користените методи за добивање на комплементарни класификатори се методите кои манипулираат или ги менуваат примероците за обука. Овие методи се базирани на обучување на N класификатори со N различни множества за обука. Овој метод има најголем број на варијации кои вклучуваат: избирање од податоците, користење на дисјунктни множества за обука, нагласување (*boosting*), адаптивно пребирање (*adaptive re-sampling*), претпроцесирање на податоците, и комбинации од нив.

Наједноставен од нив е методот за поделба на примероците за обука. Кај овој метод множеството на примероци за обука случајно се дели на N дисјунктни подмножества и потоа со секое од нив се обучува еден од класификаторите. Нема никакви ограничувања кон класификаторите и начинот на нивна обука. Бидејќи секој од класификаторите е обучен на потполно различно множество од примероци се очекува тие да бидат независни. Недостаток на овој метод е што се потребни релативно голем број на примероци за обука за да може класификаторите без значително намалување на перформансите успешно да се обучат и само со дел од нив, што го прави неупотреблив кога располагаме со мало множество за обука.

Сличен на овој е методот на обучување со вкрстена валидација или $N - 1$ од N обучување. Всушност не станува збор за никаква валидација при обуката, туку само се користи начин на поделба на податоците за обука сличен на поделбата која се користи за проверка на перформансите на обучен класификатор. Кај овој метод множеството на примероци за обука повторно случајно се дели на N дисјунктни подмножества, но секој од класификаторите се обучува користејќи $N - 1$ од овие подмножества, односно се исклучува различно подмножество од множеството на примероци за обука на секој од класификаторите, со што секој од нив е обучен со делумно различни примероци. Со ова потребата од големо множество за обука како главен недостаток на претходниот метод е надминат, но класификаторите ќе бидат и значително покорелирани бидејќи се обучувани со голем дел заеднички примероци за обука, и затоа овој метод е практичен само кога станува збор за нестабилни класификатори.

Друг метод кој се базира на добивање различни множества за обука со иста големина како и оригиналното со употреба на статистичката техника *bootstrap* е предложена од Brieman [18] и е позната како *bagging* (**B**ottstrap **A**ggregating). Множествата за обука добиени на овој начин обично малку се разликуваат од оригиналот, и примената на овој метод повторно е ограничена само за нестабилни класификатори.

Освен наведените методи кои се наменети за генерирање независни класификатори кои може да се комбинираат со произволно правило за комбинирање, предложени се голем број методи и нивни варијации за изградба на ансамбли кои го вклучуваат и правилото за комбинирање и каде секој следен класификатор во ансамблот се обучува со примероци посебно избрани на основа на перформансите на претходните веќе конструирани класификатори. Таков е методот наречен *boosting* [40] и голем број негови варијации (*Arcing*, *Adaboost*) кој при обучувањето на секој нареден класификатор придава поголемо внимание на примероците кои биле погрешно препознавани од претходно обучените класификатори, а комбинирањето се остварува со тежинска сума. И овој метод дава подобри резултати со нестабилни класификатори.

Schapire [125] демонстрирал дека група од „слаби ученици“ (*weak learners*) (во контекстот на алгоритми за учење) можат да се претворат во „јаки ученици“ како резултат на обучувањето на членовите на ансамблот со примероци кои претходно биле филтрирани од претходните веќе обучени членови на ансамблот. Голем број на емпириски студии [38][49] ја потврдија ефикасноста на *boosting* методот, иако негов главен недостаток е што тој бара голем број на примероци. Предложената варијација [43] т.н. *Adaboost* алгоритам, значително го надминува овој недостаток. Во овој пристап примероците за обука адаптивно се преби-

раат според методот Arcing [19][20] (adaptive reweighting and combining), така што тежините на оние примероци кои најчесто погрешно се препознаваат се зголемуваат, со што се зголемува веројатноста за нивен избор и соодветно нивното влијание во обуката на новите класификатори.

2.6 Различни пристапи при комбинирање на класификатори

Начините за комбинирање на класификатори може да се поделат според повеќе критериуми. Еден од критериумите е дали при комбинирањето се врши избор на само еден од нив или се фузионираат излезите на сите класификатори. Според друг критериум комбинирањето на класификаторите може да се подели на статичко и динамичко. Според трет критериум, комбинирањето би можело да се подели на фиксно или обучено. Според четврт критериум би можело да се подели на паралелно и сериско (секвенционално) во зависност од тоа дали сите класификатори работат паралелно (примерокот се проследува веднаш на сите класификатори) или постепено се вклучуваат нови во зависност од резултатот на претходните класификатори. Различни автори нудат различна поделба на начините за комбинирање на класификатори [132].

Според еден од критериумите комбинирањето на класификатори се разликуваат два главни пристапа: *фузија* (или интеграција) на класификатори и *избор* на класификатор. Кај фузијата на класификатори, непознатиот примерок паралелно се проследува на сите индивидуални класификатори, а потоа излезите од сите класификатори на некој начин се комбинираат со цел да се постигне „консензусен“ одговор од групата. Кај изборот на класификатор се прави обид да се избере класификаторот кој има најголеми изгледи точно да го препознае дадениот примерок и истиот се проследува само на избраниот класификатор чија одлука е единствена и конечна.

Од друг аспект методите за комбинирање би можеле да се поделат на *статички* и *динамички*. Кај статичките методи правилото за фузија е однапред одредено и е исто за сите примероци. Динамичките методи подразбираат динамичка промена на правилото за фузија во зависност од примерокот кој се класификува. Динамичката промена на правилото може да вклучува промена на параметрите на функцијата за комбинирање или избор на дел од класификаторите кои ќе учествуваат во донесувањето на конечната одлука. Критериумот според кој се изведува промената може да биде на основа на некои од особините на примерокот уште пред неговото проследување до класификаторите или на основа на излезите добиени од класификаторите откако примерокот ќе биде проследен до сите класификатори.

Изборот на класификатор е секогаш динамички метод и најчесто се среќава под името динамички избор на класификатор. Динамичноста подразбира одредување на класификаторот кој ќе ја донесе конечната одлука динамички за секој примерок одделно. Доколку изборот на класификатор би бил статички метод, тогаш би се свел на употреба на само еден и секогаш ист класификатор. Кај динамичкиот избор на класификатор се разликуваат два главни пристапа во зависност од начинот на кој се проценува на кој од класификаторите од комитетот ќе му се додели најголема (или целата) одговорност за класификување на непознатиот примерок. Првиот пристап се темели на динамички избор на класификатор врз основа на одредени предуслови (*preconditioned selection*). Изборот на класификаторот на кој ќе се проследи примерокот се одредува уште пред тој да се проследи на било кој од класификаторите, само на основа на неговите обележја и евентуалните статистички познавања за доменот во кој дадените класификатори се експерти. Кај вториот пристап за динамички избор на класификатор, примерокот се проследува на сите класификатори, а потоа врз основа на нивните одговори и нивната доверливост се избира класификаторот чиј одговор ќе се усвои (*postconditioned selection*). Помеѓу овие два пристапа не постои остра граница. Постојат и шеми за комбинирање кои се наоѓаат некаде во средина и кај кои иако примерокот се проследува на сите класификатори при изведувањето на конечната одлука нивните одговори не се вреднуваат подеднакво, туку поголема тежина (предност за усвојување на нивната одлука) се дава на класификаторите за кои според некои критериуми е проценето дека имаат поголеми изгледи да донесат правилна одлука.

При динамичкиот избор на класификатор [51][52][53][55], најчесто се употребува некој метод за групирање (кластерирање, партиционирање) на примероците. На пример, групите може да бидат дефинирани според множествата одлуки на индивидуалните класификатори [77], во кои класификаторите се согласуваат меѓу себе [73], или дури и според обележјата на примероците. Во некои пристапи уште пред креирањето на индивидуалните класификатори кои ќе учествуваат во динамичкиот избор, примероците за обука се групираат во одвоени групи според некои критериуми и секој од индивидуалните класификатори се обучува на само една од групите со што истиот станува експерт само за одредена група примероци [38][36][37].

Методите за фузијата на паралелни класификатори можат да се поделат од два главни аспекта. Еден од аспектите е нивната динамичност, односно дали правилата за комбинирање се менуваат од примерок до примерок, според кој може да се поделат на динамички и статички.

Вториот аспект според кој може да се поделат е на фиксни и обучени методи за комбинирање.

Кај фиксните методи правилата за комбинирање се фиксни. Тие имаат ригидни дефиниции кои не оставаат можност некои од нивните параметри или карактеристики да се променат на основа на дополнителни информации издвоени од примероците од множеството за обука или валидација. Обучените методи за комбинирање подразбираат нагдување на одредени параметри во функцијата за комбинирање на класификаторите. Овие параметри се изведуваат од некои статистички особини на примероците во множествата за обука и за валидација или од однесувањето на класификаторите на множеството за валидација, односно проценката на нивните карактеристики (пред сè точноста и меѓусебната зависност).

Кај динамичките методи, правилата за комбинирање вклучуваат динамичка самоадаптација на нивните параметри од примерок до примерок, во зависност од излезите добиени од класификаторите. И фиксните и обучените методи можат да бидат статички или динамички во зависност од тоа дали правилото за комбинирање ја зема предвид и рас пределбата на излезите од класификаторите за дадениот примерок.

Двете главни причини за комбинирање на класификатори се зголемување на ефикасноста и зголемување на точноста. За зголемување на ефикасноста (брзината) предложени се повеќе правила за комбинирање во повеќе степени (фази) каде во првиот степен примероците се класификуваат користејќи обележја кои лесно се екстрахираат од примерокот од страна на едноставен но брз класификатор со задолжителна опција на отфрлање на примероците за кои не е задоволен степенот на сигурност со кој е донесена одлуката. За препознавање на потешките примероци кои ќе бидат отфрлени од првиот класификатор потоа се употребуваат покомплексни процедури употребувајќи евентуално други обележја во секвенцијална шема [130].

Некои други пристапи применуваат постепено редуцирање на множеството на можни класи [130] во хиерархиска шема. Комбинациите на ансамбли од невронски мрежи (обучувани со различна иницијализација) се доста изучувани во подрачјето на невронските мрежи [146]. Други пристапи пак предлагаат конкатенирање на класификаторите односно поврзување на излезите од класификаторите кои се комбинираат како влезови на друг, каде проблемот на комбинирање се сведува на проблем на класификација на излезите добиени од различни класификатори. Овој пристап во литературата се среќава и под името на класификација на мета-ниво (meta level classification) [141] или (stacked generalization) [161][167]. Во литературата се предложени и голем број на пристапи и архитектури за комбинирање кои генерираат нови класи-

фикатори и кои во себе вклучуваат и специфични начини на избирање на примероците за обучување на новогенерираните класификатори.

Методите за комбинирање на класификатори можат да се поделат и според видот на базичните класификатори кои ќе се комбинираат. Од овој аспект (според [165]) комбинирањето на класификатори може да се подели на три типа:

Тип 1: Комбинацијата се прави на основа на информациите од апстрактно ниво. За L дадени индивидуални класификатори e_i , $i = 1, \dots, L$ од кои секој му доделува ознака за класа j_i на примерокот x , односно го произведува настанот $e_i(x) = j_i$. Проблемот се сведува на обединување на овие настани за да се изгради интегрален класификатор E кој на примерокот x ќе додели една дефинитивна ознака за класа j , односно $E(x) = j, j \in C = \{c_1, c_2, \dots, c_k\}$.

Тип 2: Комбинирањето се изведува на база на информациите на подредено ниво. За секој примерок x , секој од класификаторите e_i произведува подмножество $S_i \subseteq C$ со сите или дел од ознаките на класите рангирани во подредена листа. Проблемот се сведува на употреба на настаните $e_i(x) = S_i$, $i = 1, \dots, L$ за да се изведе единствен $E(x) = j, j \in C$.

Тип 3: Комбинирањето се изведува со излезите од класификаторите на квантитативно ниво. За секој примерок x , секој од класификаторите e_i произведува вектор од реални вредности $M_e(i) = [m_i^1, \dots, m_i^k]^T$ каде со m_i^t е означен степенот на верување дека примерокот x е од класата t според класификаторот e_i . Проблемот е како да се искористат овие настани $e_i(x) = M_e(i)$, $i = 1, \dots, L$ за да се изведе единствен $E(x) = j, j \in C$.

Наведените три типа на комбинирање на класификатори покриваат различни области на примена. Така на пример со комбинирањето од типот 1 може да се комбинираат индивидуални класификатори кои можат да бидат и доста различни по природа. Имено, секој класификатор, без оглед на теоријата или методологијата на која почива мора на својот излез да даде информација барем на апстрактно ниво, односно да ја понуди барем класата на непознатиот примерок. Комбинирањето од типот 3 пак бара секој од индивидуалните класификатори да може на својот излез да даде информација на квантитативно ниво. Уште повеќе, доколку излезите се со различна метричка природа мора да постои начин тие да се трансформираат во вредности со иста мерна скала. На пример, доколку некои од индивидуалните класификатори даваат проценка на постериорната веројатност, додека други растојание од прототипот, нивното комбинирање ќе има смисла само ако претходно се трансформираат во компатибилни вредности со иста мерна скала.

Се разбира, секогаш класификатор кој дава квантитативна информација може да се сведе на класификатор со излез на подредено ниво, со едноставно подредување на класите во листа според степенот на

припадност на примерокот кон секоја од нив. Ваквиот класификатор секогаш може едноставно да се сведе на класификатор со излез на апстрактно ниво со едноставно издвојување на првата класа од листата. Секој класификатор што дава квантитативна информација секогаш може да се сведе на класификатор со излез на апстрактно ниво, со едноставен избор на класата за која квантитативната информација нуди најголема поддршка. Ова наједноставно се изведува според правилото за „максимална припадност“:

$$e(x) = j \Leftrightarrow m^j = \max_{i=1, \dots, k} \{m^i\}. \quad (2.4)$$

Слично, и класификатор од типот 1 може да се претстави како класификатор од типот 3 (барем што се однесува до бројот на излезите од класификаторот $M_e = [m^1, \dots, m^k]^T$) но со ограничувања на вредностите кои може да ги прими секој од излезите:

$$m^i \in \{0, 1\}, \quad \sum_{i=1}^k m^i = 1, \quad (2.5)$$

односно, секогаш точно еден од излезите има вредност 1 а сите останати имаат вредност 0.

Од изложеново произлегува дека методите за комбинирање на класификатори од типот 1, се најгенерални бидејќи се применливи за секаков вид на индивидуални класификатори. Сепак, секое сведување на класификатори со квантитативни излези на класификатори со подредени или апстрактни излези претставува и губење на одредена информација која би можела корисно да се искористи при нивното комбинирање. Генерално, методите за комбинирање на класификатори од типот 3, имаат поголем потенцијал за подобрување на перформансите при препознавањето, иако ова не е правило и зависи од индивидуалното однесување на класификаторите.

Во литературата се предложени голем број на различни шеми за комбинирање на класификатори [80][165]. Мнозинското гласање е веројатно најупотребуваната техника [4][81][98][103][165] за комбинирање на класификатори пред сè поради нејзината едноставност и генерална употребливост на сите видови класификатори. Методот за пребројување според Борда (Borda count) [73] претставува генерализација на мнозинското гласање. Други често употребувани методи за комбинирање се минимум, максимум, медијана [97], аритметичка средина [139] и геометриска средина [140]. Тежинското усреднување [68][80][104] се обидува да ги одреди оптималните тежини за различните класификатори. Behaviour-Knowledge Space (BKS) методот се обидува да го избере

најдобриот класификатор за даден регион од просторот на влезови, а одлуката ја изведува на основа на излезите од класификаторите [162]. Останати методи за комбинирање на класификатори се Баесовиот метод [98][165], Демпстер-Шефер (Dempster-Shafer) теоријата [39][110][122][149][164][165], неспецифицираниот интеграл (fuzzy integral) [29][48][87][138], шаблоните за одлучување [96], различни веројатносни шеми [89][92], комбинирање со помош на невронска мрежа [26] и др.

2.7 Комбинирање по пат на гласање

Комбинирањето по пат на гласање е далеку наједноставниот метод за имплементација. Не бара никакво претходно познавање за однесувањето на индивидуалните класификатори, нивните перформанси и меѓузависности. Почива на добро познатата теорија за комбинирање на мислења, долго применувана во социјалната сфера. Широко е применувано и во други области каде е потребно постигнување на одредена доверливост на хардверот или софтверот. Многу лесно се имплементира во хардвер и е применливо на секаков вид класификатори. Особено е лесно применливо на готови класификатори чии карактеристики и однесување не се познати. Класификаторите ги третира како црни кутии од кои единствен и конечен излез е класата на примерокот кој се препознава. Модуларно е и многу лесно се проширува на друг број на класификатори или класи. Во самата дефиниција на гласањето природно е содржана опцијата за отфрлување на примероците за кои не постои поширока согласност меѓу гласачите. За повеќето проблеми со повеќе можни решенија, гласањето може да се употреби за да се подобри доверливоста или точноста на системот. Доверливоста на системот лесно може да се нагоди со поставување на минималниот број на потребни гласови што дадено решение треба да ги добие за да биде усвоено [101]. Токму од овој минимален број гласови произлегуваат неколку познати шеми за гласање како: консензус, квалификувано мнозинство, просто мнозинство и плурално гласање [5].

Нека имаме L независни класификатори и нека за секој непознат примерок секој од класификаторите дава единствена одлука на која од класите припаѓа примерокот. При комбинирањето на одлуките на L експерти, примерокот се распоредува во класата за која постои консензус или кога за неа ќе се добие поддршка од најмалку k од L -те индивидуални класификатори. Ако секој од класификаторите има иста веројатност p да е во право, тогаш веројатноста дека и мнозински донесената одлука $P_C(L)$ е точна, ја следи биномната распределба и може да се изрази како:

$$P_C(L) = \sum_{m=k}^L \binom{L}{m} p^m (1-p)^{L-m}, \quad (2.6)$$

каде за мнозинското гласање, k е дефинирано како:

$$k = \begin{cases} \frac{L}{2} - 1 & \text{за } L \text{ парно} \\ \frac{L+1}{2} & \text{за } L \text{ непарно} \end{cases} \quad (2.7)$$

CJT теоремата (Condorcet Jury Theorem) експлицитно ја докажува валидноста на тврдењето дека одлуката донесена од група е супериорна во однос на индивидуалните одлуки, доколку индивидуите се доволно компетентни за самостојно да донесуваат правилни одлуки со некоја разумно висока веројатност p . Тогаш за непарно $L \geq 3$ според CJT теоремата важат следниве тврдења:

Ако $p > 0.5$, тогаш $P_C(L)$ монотонно расте со растењето на L и $P_C(L) \rightarrow 1$ кога $L \rightarrow \infty$;

Ако $p < 0.5$, тогаш $P_C(L)$ монотонно опаѓа со растењето на L и $P_C(L) \rightarrow 0$ кога $L \rightarrow \infty$;

Ако $p = 0.5$, тогаш $P_C(L) = 0.5$ за сите L ;

Кај гласањето примерокот се проследува на сите L класификатори и секој од нив го класификува во точно една од можните M класи (или може дури и да го отфрли не распоредувајќи го во ниту една од класите, „воздржувајќи“ се од гласањето). Се усвојува онаа класа за која што се одлучиле, односно „гласале“ $n \geq k$ класификатори. Зависно од избраната вредност на k се добиваат различните шеми за гласање кои ги одредуваат особините на системот.

Гласањето со консензус е најрестриктивно и бара сите класификатори да се сложат околу класата на која припаѓа примерокот. Доколку и само еден од нив изрази друго мислење, заедничка одлука не може да се донесе и примерокот се отфрла. Но, кога системот ќе донесе заедничка одлука, таа е многу надежна. Гласањето со консензус одговара на вредноста на $k = n$.

Кај **квалификуваното или „ k -од- L “ мнозинство**, за да се донесе одлука потребно е барем $k > L/2$ од L -те гласачи да се согласат околу припадноста на примерокот кон една заедничка класа. Со избирање на соодветна вредноста за параметарот k може фино да се нагоди надежноста на системот. Системот со поголеми вредности на k ќе биде понадежен односно ќе има ниска рата на грешка, но ќе отфрла поголем број на примероците. За пониски вредности на k ќе се добие систем кој ќе отфрла помалку од примероците, но за сметка на тоа ќе има повисока рата на грешка.

Обичното или „просто“ **мнозинско гласање** е најмногу користено во практика и кај него за конечната одлука се бара да гласале повеќе од половината од гласачите.

Кај **плуралното гласање** нема ограничувања во минималниот број на гласови кои треба да ги освои „победникот“, односно се усвојува онаа класа која едноставно собрала најголем број на гласови. Кај него нема отфрлање на примерокот поради недоволен број на собрани гласови. За разлика од другите шеми за гласање кај **плуралното гласање** не е гарантирана единственоста на победникот, односно може да се случи и неколку кандидати да соберат подеднаков број на гласови. Истото може да се случи и кај другите шеми доколку се работи со парен број на гласачи. Во ваквите случаи или примерокот се отфрла или случајно се избира една од класите со максимален број гласови.

2.8 „Наивно“ баесово комбинирање

Наивното баесово (Bayes) комбинирање [165] претпоставува дека класификаторите кои се комбинираат се меѓусебно независни (затоа е називот „наивен“ во името). Баесовото комбинирање е обучен метод за комбинирање на класификатори од типот 1, кое почива на баесовиот формализам. Според овој пристап примерокот x се класификува во класата c_i ако за нејзината постериорна веројатност важи:

$$P(c_i | S^{(1)}, S^{(2)}, \dots, S^{(L)}) > P(c_j | S^{(1)}, S^{(2)}, \dots, S^{(L)}) \quad (2.8)$$

$$\forall i \neq j, S^{(i)} \in \{c_1, c_2, \dots, c_k\}$$

каде со $S^{(i)}$ се означени одлуките на секој од индивидуалните класификатори, при што правилата базирани на баесовата формула се обидуваат да ги проценат овие постериорни веројатности на независно множество за валидација.

Грешките што ги прави секој класификатор e_m вообичаено се изразуваат преку неговата конфузиона матрица CM_m дефинирана како:

$$CM_m = \begin{bmatrix} n_{11}^{(m)} & n_{12}^{(m)} & \dots & n_{1k}^{(m)} \\ n_{21}^{(m)} & n_{22}^{(m)} & \dots & n_{2k}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ n_{k1}^{(m)} & n_{k2}^{(m)} & \dots & n_{kk}^{(m)} \end{bmatrix}, \quad m = 1, 2, \dots, L, \quad (2.9)$$

во која секоја редица i одговара на класата c_{i_u} , а секоја колона на наставот $e_k(x) = j$. Секој од елементите $n_{ij}^{(m)}$ од матрицата го претставува бројот на примероци од класата c_i кои се класификувани во класата c_j од

страна на класификаторот e_m . Конфузионата матрица CM_m за обучениот класификатор e_m може да се добие ако класификаторот се употреби за класификување на множеството обележани примероци за валидација. Од (2.9) произлегува дека вкупниот број на примероци во множеството за валидација е:

$$N^{(m)} = \sum_{i=1}^k \sum_{j=1}^k n_{ij}^{(m)}, \quad (2.10)$$

во кои вкупниот број на примероци од класата c_i е:

$$n_{i.}^{(m)} = \sum_{j=1}^k n_{ij}^{(m)}, i = 1, \dots, k, \quad (2.11)$$

а бројот примероци кои се распоредени во класата j од страна на класификаторот e_m е:

$$n_{.j}^{(m)} = \sum_{i=1}^k n_{ij}^{(m)}, j = 1, \dots, k. \quad (2.12)$$

Проценетата точност на класификаторот, според вредностите во конфузионата матрица може да се изрази како:

$$Rr^{(m)} = \frac{\sum_{i=1}^k n_{ii}^{(m)}}{\sum_{i=1}^k \sum_{j=1}^k n_{ij}^{(m)}}. \quad (2.13)$$

За неидеалниот класификатор e_m кој понекогаш и греша, настанот $e_m(x) = j$ со кој класификаторот го изразува своето верување дека примерокот x е од класата j содржи и извесна несигурност. Со познавањето на неговата конфузиона матрица, оваа несигурност може да се опише преку условната веројатност дека претпоставката $x \in c_i$, $i = 1, \dots, k$ е точна под услов да се појавил настанот $e_m(x) = j$, односно:

$$P(x \in c_i | e_m(x) = j) = \frac{n_{ij}^{(m)}}{n_{.j}^{(m)}} = \frac{n_{ij}^{(m)}}{\sum_{i=1}^k n_{ij}^{(m)}}, i = 1, \dots, k. \quad (2.14)$$

На конфузионата матрица CM_m може да се гледа и како на претходно знаење на експерт кој по појавата на настанот $e_m(x) = j$, ги изразу-

ва своите несигурни верувања за секој од меѓусебно ексклузивните настани $x \in c_i$, $i = 1, \dots, k$ како реален број $bel(\cdot)$. Колку е повисока $bel(\cdot)$ вредноста за дадена претпоставка, толку е поверојатно дека таа е точна. Според ова, познавајќи ја конфузионата матрица за даден класификатор, „експертот“ го изразува своето верување преку $bel(\cdot)$ вредноста за секоја од претпоставките $x \in c_i$ во форма на условна веројатност изразена како:

$$bel(x \in c_i | e_m(x), EN) = P(x \in c_i | e_m(x) = j_m), i = 1, \dots, k, \quad (2.15)$$

каде со EN е означено опкружувањето кое ги опфаќа настаните независни од $e_m(x) = j_m$, $m = 1, \dots, L$, како на пример настанот на појава на специфичен примерок x на влезот на класификаторот.

На овој начин практично од класификаторот од типот 1 (кои дава само ознака на класа) се добива класификатор од типот 3 (кој дава квантитативни излези за секоја од класите како степен на верување дека примерокот е токму од неа).

При интегрирањето на мислењата според баесовиот критериум, од L класификатори $e_1, \dots, e_L$ секој од нив ќе има и сопствена конфузионата матрица CM . При препознавањето, истиот примерок x проследен на сите класификатори ќе произведе L настани $e_m(x) = j_m$, $m = 1, \dots, L$. Секој од овие настани во комбинација со соодветната конфузиона матрица за тој класификатор може да даде множество од верувања $bel(x \in c_i | e_m(x), EN)$, $i = 1, \dots, k$, од кои секое поддржува една од m -те претпоставки. Прашањето кое се наметнува е како да се обединат индивидуалните поддршки за да се добијат комбинирани $bel(\cdot)$ вредности:

$$\begin{aligned} bel(i) &= bel[x \in c_i | e_1(x), \dots, e_L(x), EN] \\ &= P[x \in c_i | e_1(x) = j_1, \dots, e_L(x) = j_L, EN], i = 1, \dots, k \end{aligned} \quad (2.16)$$

Од (2.15) и (2.16) според баесовото правило (1.1) се добива:

$$\begin{aligned} bel(i) &= P[x \in c_i | e_1(x) = j_1, \dots, e_L(x) = j_L] \\ &= \frac{P(e_1(x) = j_1, \dots, e_L(x) = j_L | x \in c_i, EN) P(x \in c_i | EN)}{P(e_1(x) = j_1, \dots, e_L(x) = j_L | EN)} \end{aligned} \quad (2.17)$$

Доколку класификаторите $e_1, \dots, e_L$ се независни тогаш и настаните $e_1(x) = j_1, \dots, e_L(x) = j_L$ ќе бидат независни еден од друг под услов $x \in c_i$ и EN , и ќе имаме:

$$\frac{P(e_1(x) = j_1, \dots, e_L(x) = j_L | x \in c_i, EN)}{P(e_1(x) = j_1, \dots, e_L(x) = j_L | EN)} =$$

$$\begin{aligned}
& \prod_{m=1}^L P(e_m(x) = j_m \mid x \in c_i, EN) \\
&= \frac{\prod_{m=1}^L P(e_m(x) = j_m \mid EN)}{\prod_{m=1}^L P(x \in c_i \mid e_m(x) = j_m)} \\
&= \frac{\prod_{m=1}^L P(x \in c_i \mid EN)}{\prod_{m=1}^L P(x \in c_i \mid EN)}
\end{aligned} \tag{2.18}$$

Бидејќи:

$$\frac{P(e_m(x) = j_m \mid x \in c_i, EN)}{P(e_m(x) = j_m \mid EN)} = \frac{P(x \in c_i \mid e_m(x) = j_m)}{P(x \in c_i \mid EN)}, \tag{2.19}$$

со замена на (2.19) во (2.17) се добива:

$$bel(i) = P(x \in c_i \mid EN) \frac{\prod_{m=1}^L P(x \in c_i \mid e_m(x) = j_m)}{\prod_{m=1}^L P(x \in c_i \mid EN)}, \tag{2.20}$$

каде $P(x \in c_i \mid e_m(x) = j_m)$ може да се процени според (2.14) со замена на j со j_m . $P(x \in c_i \mid EN)$ ја претставува веројатноста дека претпоставката $x \in c_i$ е точна под услов да се појавил x на влезот во даденото опкружување EN . Во практика, наместо изразот (2.20) се користи изразот:

$$bel(i) = \eta \prod_{m=1}^L P(x \in c_i \mid e_m(x) = j_m), \tag{2.21}$$

каде η е константа која обезбедува $\sum_{i=1}^k bel(i) = 1$ и се добива како:

$$\frac{1}{\eta} = \sum_{i=1}^k \prod_{m=1}^L P(x \in c_i \mid e_m(x) = j_m). \tag{2.22}$$

Конечно, примерокот x се класификува во класата j според правилото:

$$E(x) = j \mid bel(j) = \max_{1 \leq i \leq k} bel(i). \tag{2.23}$$

2.9 Комбинирање со познавање на поведението на класификаторите – BKS (Behavior Knowledge Space)

BKS е обучен метод за комбинирање на класификатори од типот 1. За да се избегне претпоставката дека индивидуалните класификатори кои се комбинираат се независни и се со иста точност, BKS методот предлага изведување на конечната одлука на основа на познавањето на поведението на класификаторите кое може симултано да ги регистрира одлуките на сите класификатори на секој од примероците за обука.

Просторот на познавање на поведението (BKS) е L -димензионален простор каде секоја димензија кореспондира на одлуките на еден од класификаторите. Секој од класификаторите избира една од k -те можни одлуки. Пресекот на одлуките од секој од индивидуалните класификатори адресира една ќелија во BKS. Секоја ваква ќелија во процесот на обука го акумулира бројот на примероци од секоја од класите. При класификација на непознати примероци, индивидуалните одлуки на класификаторите адресираат една ќелија во BKS и примерокот се класификува во класата која има најголем број на примероци во таа ќелија [77].

Нека $c_1, \dots, c_L$ се класите во кои класификаторите $e_1, \dots, e_L$ го распоредуваат примерокот x соодветно. Секоја можна комбинација на класите $(c_1, \dots, c_L) \in \{1, \dots, c\}^L$ претставува индекс на една ќелија во L -димензионална BKS табелата. Табелата се пополнува користејќи ги примероците од множеството за обука на комбинирањето, при што класите во кои го распределе секој од означените примероци индивидуалните класификатори претставуваат адреса на ќелијата за која дадениот примерок претставува „влезен“ примерок. Секоја ќелија содржи: (1) бројачи на влезните примероци од секоја класа, (2) вкупен број на сите влезни примероци и (3) најдобрата репрезентативна класа за дадената ќелија. Ако со $BKS(e_{(1)}, \dots, e_{(L)})$ ја означиме ќелијата во BKS табелата за која првиот класификатор го распоредил примерокот во класата $e_{(1)}$, и L -тиот класификатор во класата $e_{(L)}$, а со $n_{e_{(1)}, \dots, e_{(L)}}(m)$ вкупниот број на влезни примероци од класата m во ќелијата $BKS(e_{(1)}, \dots, e_{(L)})$, тогаш вкупниот број на влезни примероци во ќелијата $BKS(e_{(1)}, \dots, e_{(L)})$ е

$$T_{e_{(1)}, \dots, e_{(L)}} = \sum_{m=1}^k n_{e_{(1)}, \dots, e_{(L)}}(m), \quad (2.24)$$

а најдобрата репрезентативна класа за дадената ќелија е:

$$R_{e_{(1)}, \dots, e_{(L)}} = \{j \mid n_{e_{(1)}, \dots, e_{(L)}}(j) = \max_{1 \leq m \leq k} n_{e_{(1)}, \dots, e_{(L)}}(m)\}. \quad (2.25)$$

Главниот недостаток на BKS методот е што е потребно релативно големо множество на примероци за пополнување на BKS табелата во која бројот на ќелиите расте експоненцијално со бројот на класификатори кои се комбинираат. Од друга страна комплексноста на BKS табелата зависи и од бројот на класите. Според оригинално предложениот BKS метод [77], во случаите кога ќелијата е празна ($T_{e(1), \dots, e(L)} = 0$) или за која постојат барем две класи i и j со ист број на акумулирани влезни примероци ($n_{e(1), \dots, e(L)}(i) = n_{e(1), \dots, e(L)}(j)$, $i \neq j$), примерокот или се отфрла, или ако отфрлањето не е опција случајно се класификува во некоја од класите.

Со цел да се надминат овие ограничувања предложени се различни модификации на BKS методот, како вметнување на нови деформирани примероци во множеството за обука [124], методи за надминување на премногу оптимистичките проценки за точноста на класификаторите [118], методи за надминување на неодлучноста при одредувањето на репрезентативната класа за ќелиите за кои две или повеќе класи се приближно подеднакво веројатни [117] и методи за апроксимирање на дистрибуциите од висок степен со производ од дистрибуции од понизок ред [76][85].

2.9.1 Модификација на BKS методот со дополнително гласање за неопределените случаи

Видовме дека големината на BKS табелата директно зависи од бројот на класификатори кои се комбинираат и бројот на класи во кои се класификуваат примероците. Со порастот на некој (или обата) од овие фактори димензионалноста на BKS табелата експоненцијално расте, со што бројот на примероци во валидациското множество потребни за нејзина популација станува многу голем. Ова особено доаѓа до израз ако индивидуалните класификатори кои се комбинираат се со релативно добри перформанси или се значително корелирани, што често се случува во практика, и при пополнувањето на BKS табелата најголемиот број на примероци се групираат во релативно мал број на ќелии. Во ваквиот случај најчесто голем број на примероците од валидациското множество е концентриран во и (на мало хамингово растојание) околу ќелијата која ја репрезентира идеалната одлука на класификаторите односно $BKS(c, c, \dots, c) \approx N_c$, $c = 1, \dots, L$ каде со N_c е означен бројот на примероци од валидациското множество кои се од класата c . При ова голем број на ќелиите во BKS табелата остануваат без влезни примероци, односно се празни: $n_{e(1), \dots, e(L)}(m) = 0$, $\forall m$. Ова е особено изразено во практика кога валидациското множество има слични распределби како и множеството за обука.

Во процесот на препознавање, кога BKS табелата се користи за донесување на најдобрата одлука на основа на индивидуалните одлуки на класификаторите, во случаите кога распределбата на индивидуалните одлуки на класификаторите се така распоредени што адресираат „празна“ ќелија во BKS табелата, според оригиналниот BKS метод, доколку неговото отфрлање не е опција, примерокот мора случајно да се распореди во некоја од класите. При ова случајно избирање на некоја од класите во која ќе биде класификуван примерокот за кој во BKS табелата немаме никакви податоци кои би ни помогнале да одлучиме во која класа да го распоредиме, се прави просечна грешка $(k-1)/k$ каде k е бројот на класи. Оваа грешка би можела да се избегне доколку за ваквите примероци за кои не може ништо дополнително да се заклучи од BKS табелата, истата воопшто не ја користиме, односно за класификацијата на ваквите примероци одлучуваме само на основа на податоците добиени од класификаторите. Во ваквите случаи, кога располагаме со одлуките на класификаторите само на апстрактно ниво и никакви дополнителни информации за индивидуалните перформанси и меѓузависности на класификаторите, најлогичен начин за изведувањето на заедничката одлука е некоја од шемите за гласање. Бидејќи целата шема се спроведува под претпоставка дека отфрлањето на примерокот не е опција, соодветно мора да се примени и некоја од гласачките шемии кои сигурно мораат да донесат одлука, а тоа е плуралното гласање. И кај плуралното гласање, во случаите кога две или повеќе класи ќе соберат подеднаков број на гласови, мора случајно да се одлучува помеѓу нив. Под претпоставка дека вистинската класа на примерокот ќе се најде меѓу класите со најголем број на гласови, просечната грешка која се прави при ова ќе биде k_v/k каде k_v е бројот на класи кои добиле еднаков број на гласови, а k е вкупниот број на класи. Под претпоставка на релативно точни индивидуални класификатори, може да се очекува $k_v \ll k$ и вистинската класа најчесто да се најде меѓу класите кои добиле најголем број на гласови, и соодветно грешката која ќе се прави при ова ќе биде помала од грешката која би се правела доколку примерокот би се распоредувал во случајно избрана класа.

Предложената модификација на BKS методот се однесува само на примероците за кои распределбата на индивидуалните одлуки на класификаторите се такви што адресираат „празна“ ќелија во BKS табелата. Ако се споредат просечните грешки кои се прават при случајниот избор на класа за ваквите примероци кај оригиналниот BKS метод и предложената модификација:

$$\frac{k-1}{k} \quad \text{и} \quad \frac{k_v}{k}, \quad (2.26)$$

се гледа дека кај предложениот метод со дополнително гласање, грешката нема да биде поголема доколку $k_v \leq k - 1$. Ова е автоматски исполнето ако бројот на класификатори кои се комбинираат е различен од целоброен производ од бројот на класи $L \neq n \cdot k$. Општо, грешката за ваквите примероци ќе биде еднаква на очекуваната грешка за плуралното гласање. Оваа грешка според СЈТ теоремата (поглавје 2.7) е помала од грешката на индивидуалните класификатори, доколку тие се доволно компетентни за самостојно да прават грешки со рата помала од ратата на грешка на случаен избор. Гласањето нуди подобрување само ако бројот на класификатори $L > 2$.

Според ова, за примероците за кои оригиналниот BKS метод случајно избира една од можните k класи, предложената модификација со дополнително гласање сигурно нема да прави поголема грешка при нивното класификување, вклучувајќи ја и грешката при случајниот избор на класата за оние примероци за кои нема да има јасен победник при гласањето.

Во практика, примероците кои во BKS табелата ќе адресираат „празна“ ќелија и за кои ќе се примени плуралното гласање, најчесто произведуваат $k_v = 1$ и скоро секогаш $k_v \ll k$ што значи дека и кога ќе мора да се избира една од класите кои добиле еднаква поддршка од гласачите, ќе се прави случаен избор од k_v а не од сите k можни класи како во оригиналниот метод. Под услов вистинската класа на примерокот да се нашла меѓу класите со најголем број гласови (во практика најчесто се случува), со ваквиот случаен избор ќе се прави помала грешка.

Имплементацијата на дополнителното гласање за примероците кои ќе „паднат“ во празна ќелија на BKS табелата, е крајно едноставно и е тривијално во однос на комплексноста на реализацијата на BKS табелата. Гласањето може да биде дополнително изведено за примероците кои нема да може да се класификуваат според вредностите во BKS табелата. Доколку BKS табелата целосно се имплементира уште поефикасно е „празните“ ќелии во BKS табелата, односно оние за кои важи:

$$n_{e(1), \dots, e(L)}(m) = 0, \forall m, m = 1, \dots, k, \quad (2.27)$$

уште при нејзината конструкција над валидациското множество да се пополнат со класата која најчесто се појавува во нејзиниот индекс, односно:

$$R_{e(1), \dots, e(L)} = \text{mode}\{e(1), \dots, e(L)\}, \quad (2.28)$$

со што сите примероци кои можат да бидат еднозначно класификувани, ќе може да се класификуваат со директно читање на соодветната вредност од BKS табелата без дополнително броење на гласовите. Примеро-

ците кои ќе адресираат ќелија за која не може да се одреди најдобра репрезентативна класа според (2.25) или (2.28) ќе бидат отфрлени или доколку отфрлувањето не е опција ќе се класификуваат во случајно избрана класа. Сепак целосната имплементација на BKS табелата најчесто не се практикува поради нејзината голема димензионалност и огромните мемориски ресурси потребни за нејзина потполна реализација особено кога се работи за поголем број на класификатори или класи.

Предложениот метод се однесува исто како и оригиналниот за примероците кои при препознавањето ќе адресираат „пополнета“ ќелија во BKS табелата. Однесувањето е исто и за примероците кои според индивидуалните одлуки на класификаторите ќе адресираат ќелија за која не може да се одреди единствен максимум според (2.25) или пак се „празни“, а најчесто појавената класа не е единствена според (2.28). Ваквите примероци или ќе бидат отфрлувани или доколку отфрлувањето на примерокот не е опција ќе се класификуваат во една случајно избрана од k -те класи. Разликата се однесува само на примероците за кои соодветната ќелија во BKS табелата е празна, за кои едноставното броење на гласовите од класификаторите има поголеми изгледи правилно да ги класификува отколку случајниот избор на класа според оригиналниот BKS метод.

Едноставната имплементација и фактот дека предложената модификација на BKS методот со дополнително гласање има најмалку еднакви перформанси на препознавање со оригиналниот метод, го прави особено добар кандидат за негова замена. Подобрените перформанси при препознавањето особено доаѓаат до израз при комбинирањето на поголем број на класификатори. Ова се должи пред сè на фактот што со зголемувањето на димензионалноста на BKS табелата, а притоа користејќи ист број на примероци за валидација за нејзина конструкција, таа станува сè „поретка“, односно процентуално поголем број на ќелии се „празни“. Ова практично значи дека при препознавањето, поголем број на примероци ќе адресираат „празни“ ќелии и наместо да бидат случајно класификувани во некоја од класите, за нив ќе биде применето плурално гласање со просто мнозинство кое има поголеми изгледи да го распореди примерокот во вистинската класа.

2.10 Методи за комбинирање со прередување на ранг листата на класи

Целта на методите за комбинирање со прередување на класите во ранг листата (class set reordering) е да ја промовираат вистинската класа и да ја постават што повисоко на ранг листата на можни класи за дадениот примерок. Овие методи комбинираат класификатори кои даваат

информација на подредено ниво (класификатори од типот 2). Основниот критериум за проценка на нивната успешност е позицијата на вистинската класа во пререцената ранг листа да биде повисока во однос на нејзината позиција пред пререкувањето во ранг листите на индивидуалните класификатори.

2.10.1 Комбинирање со избор на највисок ранг

Една од наједноставните методи за комбинирање на повеќе рангирања за пререкување на ранг листата е методот на највисок ранг. Според овој метод секој примерок се проследува на L класификатори кои произведуваат ранг листа на класите за дадениот примерок, со што секоја класа е потенцијално различно рангирана од секој од L -те класификатори. Минимумот од овие рангирања (највисокиот ранг) се усвојува како конечна оценка за дадената класа. Потоа класите се пререкуваат според добиените оценки со што се добива конечната пререцената листа на класи за дадениот примерок. Пример за пресметување на оценките на класите според методот на највисок ранг е илустриран на слика 2.2. Во случаите кога повеќе класи се подеднакво рангирани во финалниот распоред, случајно се избира една од нив. Овој начин на комбинирање е употреблив само кога се работи за многу голем број на класи и релативно мал број на класификатори. Слабата страна на овој метод е што комбинираната ранг листа може да има повеќе класи со иста оценка, што ја прави доста непогодна за донесување на конечната одлука. Затоа овој метод не е погоден за комбинирање на класификатори при изведување на конечната одлука за припадноста на примерокот. Истиот може да биде доста користен за рангирањето на класите, односно редуцирање на бројот на можни класи за даден примерок кој ќе биде дополнително третиран имајќи ги предвид само највисоко рангираните класи кај повеќестепените системи за препознавање примероци.

2.10.2 Комбинирање според методот за пребројување на Борда

Комбинирањето според методот за пребројување на Борда (Borda count) претставува генерализација на методот за мнозинско гласање. За секоја класа c_j Борда вредноста ја претставува сумата на бројот на класите кои се рангирани под класата c_j од секој од класификаторите [11]. Формално, за секоја класа c од множеството класи S , нека $B_j(c)$ е бројот на класи во S кои се рангирани пониско од класата c според класификаторот C_j ($j = 1, \dots, L$). Тогаш Борда вредност за класата c е

$$B(c) = \sum_{j=1}^L B_j(c).$$

Пример за пресметување на Борда вредноста е претставен на слика 2.2.

Интуитивно, ако класата е рангирана блиску до врвот од страна на повеќе класификатори, нејзината Борда вредност тежи да биде поголема. Големината на Борда вредноста за дадена класа ја изразува согласноста од страна на ансамблот од класификатори дека примерокот е член на дадената класа. Резултантното рангирање на класите се прави според големината на вака дефинираната согласност. Конечно, примерокот се класификува во класата за која Борда вредноста е најголема, односно прворангираниот според Борда вредноста. За проблем со само две класи, комбинирањето по методот на Борда е еквивалентно на мнозинското гласање.

Ранг	класа	Ранг	класа	Ранг	класа	Ранг	класа
1		1		1		1	
2		2		2	y	2	
3		3		3		3	y
4	y	4	x	4		4	
5		5		5		5	x
6		6		6		6	
7		7	y	7		7	
8	x	8		8	x	8	
9		9		9		9	
10		10		10		10	
тежина:		0.23		0.16		0.41	0.35

Метод на највисок ранг:

$$H(x) = \min(8, 4, 8, 5) = 4$$

$$H(y) = \min(4, 7, 2, 3) = 2$$

Метод на Борда:

$$\begin{aligned} B(x) &= (10 - 8) + (10 - 4) + (10 - 8) + (10 - 5) \\ &= 2 + 6 + 2 + 5 \\ &= 15 \end{aligned}$$

$$\begin{aligned} B(y) &= (10 - 4) + (10 - 7) + (10 - 2) + (10 - 3) \\ &= 6 + 3 + 8 + 7 \\ &= 24 \end{aligned}$$

Тежински метод на Борда:

$$\begin{aligned} L(x) &= 0.23 \cdot (10 - 8) + 0.16 \cdot (10 - 4) + 0.41 \cdot (10 - 8) + 0.35 \cdot (10 - 5) \\ &= 0.23 \cdot 2 + 0.16 \cdot 6 + 0.41 \cdot 2 + 0.35 \cdot 5 \\ &= 3.99 \end{aligned}$$

$$\begin{aligned} L(y) &= 0.23 \cdot (10 - 4) + 0.16 \cdot (10 - 7) + 0.41 \cdot (10 - 2) + 0.35 \cdot (10 - 3) \\ &= 0.23 \cdot 6 + 0.16 \cdot 3 + 0.41 \cdot 8 + 0.35 \cdot 7 \\ &= 7.59 \end{aligned}$$

Слика 2.2. Пресметување на највисокиот ранг и Борда вредноста за две (x и y) од десет можни класи според рангирањата на четири класификатори

Методот на Борда може да се разгледува како генерализација на либерално гласање каде секој од гласачите не гласа за само еден кандидат, туку ги рангира сите понудени кандидати, односно дава различен степен на поддршка за секој од нив. За проблем на избор од k кандидати, ова може да се интерпретира како давање на различен број на гласови за секој од кандидатите од страна на секој од гласачите, при што секој од гласачите располага со точно k пакети со по 1, 2, ..., k гласови кои ги распоредува на сите кандидати. Секој од гласачите го доделува пакетот со k гласови на својот прв избор, пакетот со $k-1$ гласови на вториот избор, итн.

Кај Борда методот може да се случи две или повеќе класи да добијат иста поддршка, односно да имаат исти Борда вредности, во кои случаи може едноставно случајно да се избере една од еднакво оценетите класи. Постојат различни варијации на Борда методот [11], вклучувајќи и такви кои посебно го третираат проблемот на нерешен резултат при рангирањето.

2.10.3 Тежински Борда метод и логистичка регресија

Борда методот за комбинирање ги третира индивидуалните класификатори како потполно независни и рамноправни, што може да биде непогодно доколку однапред знаеме дека одредени класификатори имаат повеќе шанси да бидат во право при одлучувањето во однос на останатите. Во ваквите случаи може да се примени тежинското Борда пресметување, како обучен метод за комбинирање на класификатори кои даваат барем ранг листа на класите за секој примерок.

Тежинското Борда комбинирање претставува модификација на Борда методот со придружување на тежини со кои се множат рангирањата од секој од класификаторите со што методот се адаптира за комбинирање на класификатори со различни точности. Овие тежини ја одразуваат релативната важност на секој од индивидуалните класификатори во ансамблот. Пример за пресметување на тежинската Борда вредност е претставен на слика 2.2. Тежините на секој од индивидуалните класификатори во ансамблот може да се одредат според методот на логистичка регресија [70].

Методот за комбинирање со логистичка регресија е најмоќен од трите разгледани методи за прередување на ранг листата на класи, и тој е обучен метод. За негова имплементација е потребна проценка на тежините асоцирани кон секој од индивидуалните класификатори кои мора да се изведат на релативно големо независно множество на примероци за валидација.

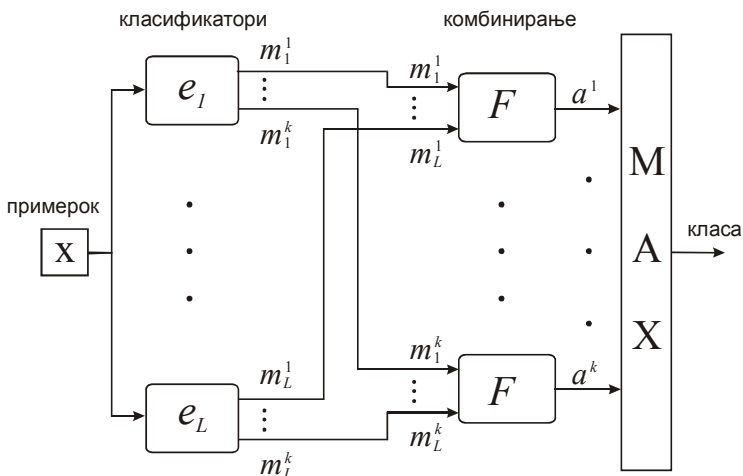
2.11 Комбинирање по пат на усреднување

Методите за комбинирање со усреднување се едни од најчесто применуваните во практика. Усреднувањето е директно применливо само при комбинирањето на класификатори од типот 3, односно комбинирањето се изведува на квантитативните излези од класификаторите. Методите за комбинирање по пат на усреднување почиваат на претпоставката дека класификаторите се независни меѓусебе и дека се со слични перформанси. При усреднувањето интуитивно се очекува средната вредност од повеќе несигурни, но релативно точни мерења на иста величина да дадат подобра проценка за мерената величина.

Формално, ако за секој примерок x , секој од класификаторите e_i , ($i = 1, \dots, L$), произведува вектор од реални вредности $M_e(i) = [m_i^1, \dots, m_i^k]^T$ каде со m_i^t е означен степенот на верување дека примерокот x е од класата t според класификаторот e_i , усреднувањето претставува функција со која од излезите од сите L класификатори за класата t ($t = 1, \dots, k$) се произведува единствена вредност a^t како степен на верување дека примерокот е од дадената класа:

$$a^t = F(m_1^t, \dots, m_L^t), \forall t = 1, \dots, k, \quad (2.29)$$

каде F е функцијата за усреднување (слика 2.3).



Слика 2.3. Процес на комбинирање на класификатори

Финалната одлука се донесува според правилото за максимална припадност (2.4) односно:

$$Q(x) = \arg \max_{j=1, \dots, k} F(m_1^j, \dots, m_L^j). \quad (2.30)$$

Функцијата за усреднување секогаш ги усреднува излезите за иста класа од сите класификатори, притоа еднакво третирајќи ги сите класификатори и сите класи. Во зависност од избраната функција за усреднување се добиваат различни методи за комбинирање.

2.11.1 Геометриска средина

Ако за функција за усреднување се земе геометриската средина:

$$F_G(m_1, \dots, m_L) = \sqrt[L]{\prod_{i=1}^L m_i}, \quad (2.31)$$

се добива т.н. комбинирањето со производ, геометриско усреднување или „правило на производ“ (product rule). При имплементацијата во практика најчесто се избегнува коренувањето на производот на излезите од класификаторите, бидејќи од него не зависи изборот на класата според правилото за максимална припадност:

$$Q(x) = \arg \max_{j=1, \dots, k} \sqrt[L]{\prod_{i=1}^L m_i^j} = \arg \max_{j=1, \dots, k} \prod_{i=1}^L m_i^j. \quad (2.32)$$

Правилото за комбинирање по пат на производ произлегува директно од баесовото правило, под претпоставките дека излезите од класификаторите може да се интерпретираат како постериорни веројатности, класификаторите да се меѓусебно статистички независни и априорните веројатности на класите да се меѓусебно еднакви, односно сите класи да се подеднакво веројатни [92].

2.11.2 Аритметичка средина

Ако за функција за усреднување се земе аритметичката средина:

$$F_A(m_1, \dots, m_L) = \frac{1}{L} \sum_{i=1}^L m_i, \quad (2.33)$$

се добива комбинирањето со просек или „правило на сума“ (sum rule). При имплементацијата во практика најчесто се избегнува делењето на сумата на излезите од класификаторите со нивниот број, бидејќи од оваа операција не зависи изборот на класата според правилото за максимална припадност:

$$Q(x) = \arg \max_{j=1, \dots, k} \frac{1}{L} \sum_{i=1}^L m_i^j = \arg \max_{j=1, \dots, k} \sum_{i=1}^L m_i^j. \quad (2.34)$$

И правилото за комбинирање по пат на сумирање може да се изведе од баесовото правило под претпоставка дека излезите од класификаторите може да се интерпретираат како постериорни веројатности и дека вака добиените постериорни веројатности не се разликуваат драстично од априорните веројатности [92]. Иако втората претпоставка е доста слободно усвоена и честопати не може да се обезбеди во практика, сепак комбинирањето на класификатори по пат на аритметичка средина од нивните излези е најшироко употребуваниот метод за комбинирање кој во практика редовно дава еден од најдобрите резултат од неочуваните методи за комбинирање.

Главната разлика при комбинирањето со примена на геометриската и аритметичката средина на излезите е во нивната осетливост на грешка, односно колку грешките на индивидуалните класификатори се одразуваат на комбинираниот излез. Грешката на некој од индивидуалните класификатори многу повеќе влијае при комбинирањето користејќи геометриска средина отколку при користењето аритметичка средина. Така на пример доколку само еден од индивидуалните класификатори не го препознае примерокот и произведе $m_i^j \approx 0$ тогаш и за целиот производ ќе важи $\prod_{i=1}^L m_i^j \approx 0$, независно од тоа како го препознале примерокот останатите класификатори во ансамблот. Овој ефект кај хармониското комбинирање е уште подрастично изразен. Комбинирањето со аритметичка средина е најмалку осетливо на грешките на индивидуалните класификатори.

2.11.3 Хармониска средина

Ако за функција за усреднување се земе хармониската средина:

$$F_H(m_1, \dots, m_L) = \frac{L}{\sum_{i=1}^L \frac{1}{m_i}}, \quad (2.35)$$

се добива хармониско комбинирање. И тука при практичната имплементацијата може да се изостави факторот L во броителот од кој не зависи изборот на класата според правилото за максимална припадност кое може да се трансформира во:

$$Q(x) = \arg \max_{j=1, \dots, k} \frac{L}{\sum_{i=1}^L \frac{1}{m_i^j}} = \arg \max_{j=1, \dots, k} \frac{1}{\sum_{i=1}^L \frac{1}{m_i^j}} = \arg \min_{j=1, \dots, k} \sum_{i=1}^L \frac{1}{m_i^j} \quad (2.36)$$

2.11.4 Комплементарен производ

Кога вредностите кои се усреднуваат може да се интерпретираат како веројатности, може да се примени и усреднување на комплементарниот производ дефинирано како:

$$F_C(m_1, \dots, m_L) = 1 - \sqrt[L]{\prod_{i=1}^L (1 - m_i^j)} . \quad (2.37)$$

И кај комбинирањето со комплементарен производ при практичната имплементацијата може да се изостави коренувањето на производот и може да се бара минимумот од производите на комплементите наместо максимум од комплементите на ваквите производи:

$$Q(x) = \arg \max_{j=1, \dots, k} \left(1 - \sqrt[L]{\prod_{i=1}^L (1 - m_i^j)} \right) = \arg \min_{j=1, \dots, k} \prod_{i=1}^L (1 - m_i^j) . \quad (2.38)$$

Доколку излезите од класификаторите $m_i^j = P(x \in c_j | e_i(x))$ се интерпретираат како постериорна веројатност дека примерокот x е од класата c_i , тогаш може да се смета дека веројатноста на спротивниот настан претставува комплемент на оваа веројатност:

$$P(x \notin c_j | e_i(x)) = \neg P(x \in c_j | e_i(x)) = \neg m_i^j = 1 - m_i^j \quad (2.39)$$

Тогаш, комбинирањето на излезите од повеќе класификатори може да се изведе со комбинирање на веројатностите на комплементарните (спротивните) настани и комплементирање на вака добиената вредност. Токму овој пристап го следи и комбинирањето со комплементарен производ.

2.12 Комбинирање со подредување

Комбинирањето со подредување (order statistics) подразбира подредување на излезите за одредена класа од сите класификатори во неопаѓачки редослед според нивната вредност:

$$m_{1:L}^t \leq m_{2:L}^t \leq \dots \leq m_{L:L}^t . \quad (2.40)$$

По вака дефинираното подредување може да се дефинираат три нови методи за комбинирање на класификатори: правило на максимум, правило на минимум и правило на медијана [143][145].

2.12.1 Правило на максимум

Според правилото на максимум од излезите од сите класификатори за одредена класа t се избира оној со максимална вредност $m_{\max}^t = m_{L:L}^t$. Од вака избраните излези за секоја од класите потоа се избира класата во која конечно ќе се распореди примерокот според правилото за максимална припадност (2.30), односно функцијата за „усреднување“ во овој случај е функција за избор на максимум:

$$Q(x) = \arg \max_{j=1, \dots, k} \max(m_1^j, \dots, m_L^j). \quad (2.41)$$

Овој метод за комбинирање е познат и под името макс-макс критериум за избор бидејќи прво за секоја класа се избира максималната поддршка од сите класификатори, а потоа се избира класата за која поддршката е максимална.

2.12.2 Правило на минимум

Правилото на минимум, од излезите на сите класификатори за одредена класа t го избира оној со минимална вредност $m_{\min}^t = m_{1:L}^t$. Од вака избраните минимуми за секоја од класите потоа се избира класата во која конечно ќе се распореди примерокот според правилото за максимална припадност (2.30), односно во овој случај функцијата за „усреднување“ е функција за избор на минимум:

$$Q(x) = \arg \max_{j=1, \dots, k} \min(m_1^j, \dots, m_L^j). \quad (2.42)$$

Овој метод за комбинирање е позната и под името мин-макс критериум за избор бидејќи прво за секоја класа се избира минималната поддршка од сите класификатори, а потоа се избира класата за која поддршката е максимална.

2.12.3 Правило на медијана

Според правилото на медијана, се избира излезот кој би се нашол во средината на подредената листа од сите излези од класификаторите за одредена класа t , или попрецизно:

$$m_{med}^t = \frac{\frac{m_{\frac{L}{2}:L}^t + m_{\frac{L+1}{2}:L}^t}{2}}{2}, \text{ ако } L \text{ е парно; } \quad m_{\frac{L+1}{2}:L}^t, \text{ ако } L \text{ е непарно} \quad (2.43)$$

Со изборот на средишната (медијална) вредност од дадена дистрибуција се обезбедува подеднаков број од примероците од дистрибуцијата да бидат со поголема и со помала вредност од медијалната. Во овој случај функцијата за „усреднување“ е функција за медијана, по што примерокот се распоредува во класата според правилото за максимална припадност:

$$Q(x) = \arg \max_{j=1, \dots, k} \text{med}(m_1^j, \dots, m_L^j). \quad (2.44)$$

Комбинирањето по правилото на максимум практично значи дека за секоја класа се избира излезот од само еден од класификаторите и тоа од оној кој е „најубеден“ дека примерокот е од дадената класа. Недостатокот на овој вид на комбинирање е што единствен класификатор кој постојано дава високи (премногу самоуверени) вредности, може лесно да ги наруши перформансите на комбинацијата. При комбинирањето по правилото на минимум за секоја класа се избира излезот од оној класификатор кој е „најмалку убеден“ дека примерокот е од дадената класа. Недостатокот на овој вид на комбинирање е сличен како за правилото на максимум, односно перформансите на комбинацијата може лесно да ги наруши единствен класификатор кој постојано дава ниски (неуверливи) вредности. Комбинирањето по правило на медијана пак, го избира „најтипичниот“ излез од класификаторите за секоја класа, со што е отпорен на недостатоците на мин-макс и макс-макс критериумите. Доколку системот по комбинирањето превзема и отфрлање на несигурните примероци, мин-макс критериумот е попожелен бидејќи овозможува со едноставно правило за отфрлање да се постигне поголема доверливост на системот, односно овозможува сомнителните примероци полесно да бидат разобличени.

2.13 Демпстер-Шефер комбинирање

Ова комбинирање почива на примената на теоријата за докази (Dempster-Shafer theory of evidence) [34][131]. Демпстер-Шефер (D-S) теоријата претставува моќно оружје за претставување на несигурно знаење. D-S теоријата наместо да ја пресметува веројатноста на дадена претпоставка се концентрира кон наоѓањето на веројатноста дека доказите (сведоштвата) ја поддржуваат претпоставката. Според ова, недостатокот на докази не мора неизбежно да повлекува и дека претпоставката е погрешна. Особено е погодна за користење кај непотполни модели кај кои не може да се обезбедат доволно податоци кои би можеле прецизно да ги проценат априорните и условните веројатности на настаните. D-S теоријата овозможува претстава на непрецизноста на под-

дршката која се дава на основа на доказите, поради нивната несигурност или нецелосност, односно додава скептичност во резонирањето.

Довербата (belief) кон некој настан ја претставува веројатноста дека настанот е поддржан од страна на расположливите докази. Уверливоста (plausability) ја изразува веројатноста дека настанот е компатибилен со расположливите докази (доказите не го побиваат настанот иако можеби директно не го поддржуваат).

D-S теоријата почива на одредувањето на доверливоста за некој настан која се пресметува преку ВРА (basic probability assignment) функција. Нека $\Theta = \{\theta_1, \dots, \theta_M\}$ е множество на конечен број хипотези. Подмножеството $\{\theta_{i_1}, \dots, \theta_{i_q}\} \subset \Theta$ ја претставува хипотезата која ја претставува унијата $\theta_{i_1} \cup \dots \cup \theta_{i_q}$. Сите можни подмножества на множеството Θ го формираат множеството 2^Θ . Според D-S теоријата ВРА функцијата m доделува вредност во опсегот $[0, 1]$ на секое подмножество A на Θ и кои ја задоволуваат следната релација:

$$m(\emptyset) = 0, \quad \sum_{A \subset \Theta} m(A) = 1. \quad (2.45)$$

Во стандардната теорија на веројатност мерката *веројатносии* се придружува кон атомична хипотеза θ_u , додека $m(A)$ е дел од довербата која ја поддржува хипотезата A , но не поддржува ништо поспецифично, како одредени подмножества од A . Доколку A не е едноелементно подмножество ($A \neq \theta_i$), $m(A)$ не може понатаму да се подели на помали подмножества. Традиционалната теорија на веројатност условува довербата и недовербата искажани кон некој настан да бидат комплементарни $P(a) + P(\neg a) = 1$, додека D-S теоријата оди чекор подалеку и дозволува релативната доверливост или недоверливост да се остават неспецифицирани и според неа $m(A) + m(\neg A) \leq 1$.

Степенот на доверливост $\text{bel}(\cdot)$ е вредност од опсегот $[0, 1]$ која се доделува на секое непразно подмножество B од Θ и се изразува преку ВРА функцијата како:

$$\text{bel}(B) = \sum_{A \subseteq B} m(A). \quad (2.46)$$

Одовде доверливоста претставува мерка која ги зема предвид сите докази кои ја поддржуваат хипотезата A .

Степенот на уверливост (plausability) $\text{pla}(\cdot)$ се дефинира како:

$$\text{pla}(B) = 1 - \sum_{A \cap B = \emptyset} m(A) \quad (2.47)$$

и претставува мерка која ги зема предвид сите докази кои не ја побиваат хипотезата A . Одовде $\text{bel}(\cdot)$ и $\text{pla}(\cdot)$ вредностите претставуваат долна и горна граница соодветно на интервалот на доверливост за дадена хипотеза или настан. Од друга страна ВРА функцијата може да се смета како генерализација на функцијата на распределба на веројатност каде степенот на доверливост би бил генерализација на функцијата на веројатност.

Кога постојат две или повеќе сведоштва, ќе постојат и две или повеќе множества од ВРА функции и доверливости за подмножества од истото множество Θ . Под претпоставка дека овие сведоштва се независни меѓу себе, може да се примени D-S правилото за комбинирање со кое се добиваат нови ВРА и доверливости кои го изразуваат комбинираното влијание на повеќето сведоштва. Според D-S правилото за комбинирање две ВРА функции m_1 и m_2 кои ги поддржуваат хипотезите A и B соодветно, во единствена m која ја поддржува хипотезата $C = A \cap B \neq \emptyset$ е дефинирано како:

$$m(C) = m_1 \oplus m_2(C) = \frac{\sum_{A \cap B = C} m_1(A)m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A)m_2(B)}, \quad C \neq \emptyset. \quad (2.48)$$

При комбинирањето според D-S правилото $\oplus$ операторот е комутативен и асоцијативен, па комбинирањето на повеќе сведоштва директно произлегува од (2.48). Комбинирањето на класификатори според D-S правилото директно зависи од сведоштвата обезбедени од класификаторите. Ху [165] комбинира класификатори кои како излез даваат само класа на примерокот и вклучуваат отфрлање според D-S правилото, при што ратите на препознавање, супституција и отфрлање се употребени како мерка за ВРА и доверливоста m . Rogova [122] користи неколку мерки за сличност помеѓу референтниот излезен вектор и актуелните излези од класификаторите кои ги трансформира во сведоштва за да го примени D-S правилото за комбинирање. Употребените референтните вектори се составени од средните вредности μ_n^k на излезите на секој од класификаторите c^n за секоја од класите k . Сличноста на актуелните излези од секој класификатор y^n со референтните вредности изразена како d_n^k се трансформира во ВРА вредности како:

$$\begin{aligned} m_k(\theta_k) &= d_k^n, & m_k(\Theta) &= 1 - d_k^n \\ m_{\bar{k}}(\bar{\theta}_k) &= 1 - \prod_{l \neq k} (1 - d_l^n), & m_{\bar{k}}(\Theta) &= \prod_{l \neq k} (1 - d_l^n) \end{aligned} \quad (2.49)$$

Сведоштвото од класификаторот c^n за класата k се добива со комбинирањето на $m_k \oplus m_{\bar{k}}$. Конечно, D-S правилото се употребува за да се комбинираат сведоштвата од сите класификатори и да се добие мерка на доверливост за секоја од класите. Варијации на D-S правилото за комбинирање вклучуваат различни мерки за сличност [122], поинакви фактори за нормализација [1] и тежинско комбинирање на сведоштвата [163][164]. Со примена на различни мерки за сличност како опишаните мерки за сличност и растојанија применети при комбинирањето со шаблони за одлучување (види поглавја 2.18 и 2.19) се добиваат различни варијанти на D-S правилото за комбинирање.

2.14 Веројатносен производ

Веројатносниот производ [94][94][140] е метод за комбинирање на класификатори кој слично како и комбинирањето со правилото на производ (геометриска средина) тргнува од баесовото правило и под претпоставка дека излезите од класификаторите може да се интерпретираат како постериорни веројатности и дека класификаторите се меѓусебно статистички независни се добива:

$$Q(x) = \arg \max_{j=1, \dots, k} \frac{\prod_{i=1}^L m_i^j}{P(x \in c_j)^{L-1}}, \quad (2.50)$$

каде со $P(x \in c_j)$ е означена априорната веројатност за појава на класата c_j . Априорните веројатности за секоја од класите се проценуваат од множеството за обука на класификаторот како:

$$P(x \in c_j) = \frac{N_j}{N}, \quad (2.51)$$

каде N_j е бројот на примероци од класата c_j а N е вкупниот број на примероци за обука.

Комбинирањето со веројатносен производ се разликува од правилото на производ само по вклучувањето на проценетите вредности на априорните веројатности за секоја од класите, со што донекаде ги амортизира грешките на класификаторите. За класификациските проблеми во кои сите класи се приближно подеднакво застапени ова правило за комбинирање се сведува на правилото на производ.

2.15 Комбинирање со неспецифициран интеграл

Неспецифицираниот интеграл (fuzzy integral) е нелинеарен функционал кој е дефиниран во однос на неспецифицирана мера, особено g_λ неспецифицираната мера предложена од Sugeno [137] и е користен од повеќе автори за комбинирање на резултатите од повеќе извори на информации [28][29][138][150][155][166].

Неспецифицираните мери претставуваат генерализација на класичните мери. Пресликувањето $g : 2^Y \rightarrow [0, 1]$, каде Y е конечно множество, е наречено неспецифицирана мера ако:

1. $g(\emptyset) = 0, g(Y) = 1$,
2. $g(A) \leq g(B)$ ако $A \subset B$,
3. ако $\{A_i\}_{i=1}^\infty$ е растечка секвенца на мерливи множества, тогаш:

$$\lim_{i \rightarrow \infty} g(A_i) = g(\lim_{i \rightarrow \infty} A_i). \quad (2.52)$$

Вака дефинираното пресликување g не мора да биде адитивно и кај него својството на монотоност го заменува својството на адитивност. Од оваа дефиниција Sugeno ја вовел g_λ неспецифицираната мера која го задоволува и следниот дополнителен услов:

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B) \quad (2.53)$$

за сите $A, B \subset Y$ и $A \cap B = \emptyset$ и за некое $\lambda > -1$. Таа овозможува мерата на унијата на две дисјунктни множества да може да се пресмета директно од мерите на компонентите (множествата).

Ако Y е конечно множество и $h : Y \rightarrow [0, 1]$ е неспецифицирано подмножество на Y , тогаш неспецифицираниот интеграл над $A \subset Y$ на функцијата h во однос на неспецифицираната мера g е дефиниран како:

$$\begin{aligned} \int_A h(y) \circ g(\cdot) &= \sup_{E \subseteq Y} \left[\min \left(\min_{y \in E} h(y), g(A \cap E) \right) \right] \\ &= \sup_{\alpha \in [0,1]} [\min(\alpha, g(A \cap h_\alpha))] \end{aligned} \quad (2.54)$$

каде h_α е т.н. α -ниво подмножество од h :

$$h_\alpha = \{y \mid h(y) \geq \alpha\}. \quad (2.55)$$

Интерпретацијата на употребените ознаките е следнава: $h(y)$ го изразува степенот со кој концептот h е задоволен од y . Изразот $\min_{y \in E} h(y)$ го изразува степенот со кој концептот h е задоволен од сите елементи во E . Вредноста $g(E)$ го изразува степенот со кој подмножест-

вото E го задоволува концептот кој го мери g . Неспецифицираниот интеграл може да се интерпретира како пронаоѓање на максимален степен на согласност меѓу објективните докази и очекувањето.

Пресметувањето на неспецифицираниот интеграл е дефинирано на следниот начин: Ако $Y = \{y_1, y_2, \dots, y_n\}$ е конечно множество и $h : Y \rightarrow [0, 1]$ е функција, нека елементите во Y ги подредиме така да биде задоволено:

$$h(y_1) \geq h(y_2) \geq \dots \geq h(y_n). \quad (2.56)$$

Тогаш неспецифицираниот интеграл e , во однос на неспецифицираната мера g дефинирана над Y може да се пресмета како:

$$\varphi = \max_{i=1}^n [\min(h(y_i), g(A_i))], \quad (2.57)$$

каде $A_u = \{y_1, y_2, \dots, y_u\}$. Кога g е g_λ неспецифицираната мера, вредностите на $g(A_i)$ може да се одредат рекурзивно како:

$$g(A_1) = g(\{y_1\}) = g^1, \quad (2.58)$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), 1 < i \leq n. \quad (2.59)$$

каде λ се добива со решавање на равенката:

$$\lambda + 1 = \prod_{i=1}^n (1 + \lambda g^i), \quad (2.60)$$

каде $\lambda \in (-1, +\infty)$ и $\lambda \neq 0$. Според ова за пресметување на неспецифицираниот интеграл во однос на g_λ неспецифицираната мера потребно е само познавање на функцијата на густина на распределба, каде u -тата густина g^u се интерпретира како степенот на важност на изворот y_u за конечниот исход.

При комбинирањето на класификатори со неспецифицираниот интеграл класификаторите се третираат како извори на информации и нивните излези се фузионираат користејќи го неспецифицираниот интеграл за да се донесе единствена одлука. Нека $C = \{c_1, c_2, \dots, c_k\}$ е множеството на класи во класификацискиот проблем и нека $Y = \{y_1, y_2, \dots, y_L\}$ е множеството на класификатори кои се комбинираат. Ако x е објектот кој треба да се препознае, нека функцијата $h_j : Y \rightarrow [0, 1]$ претставува делумна проценка дека објектот x е од класата c_j , односно $h_j(y_i)$ претставува индикација за тоа колку сме сигурни дека класификацијата на објектот x во класата c_j од страна на класификаторот y_i е точна, каде 1 означува апсолутна сигурност, додека 0 означува апсолутна сигурност дека тоа е погрешно односно дека објектот x не е од класата c_j .

За секој класификатор c_i треба да се придружи соодветна неспецифицирана густина g^i која го изразува степенот на важност на соодветниот класификатор во ансамблот. Овие густини можат да бидат субјективно доделени од страна на експерт или пак да бидат проценети на основа на примероци од множеството за валидација. На основа на доделените неспецифицирани густини, од (2.60) се пресметува λ , и со (2.58) и (2.59) се конструира неспецифицираната мера g_λ . Конечно со (2.57) се пресметува неспецифицираниот интеграл, а потоа класата c_j со најголема вредност на интегралот се избира како конечна класа во која се распоредува примерокот.

Неспецифицираните густини g^i што се придружуваат на секој од класификаторите треба да ја презентираат важноста на класификаторот. Најсоодветно е за ова да се земе проценетата точност на класификаторите при класификацијата на примероци од множеството за валидација. Wang et al. [155] предлагаат за g^i да се земе половина од проценетата веројатност за точна класификација од страна на класификаторот e_i :

$$g^i = \frac{\hat{P}_i}{2}, \quad (2.61)$$

додека Cho et al. [29] користат $g^i = \hat{P}_i$.

При комбинирањето на класификатори за препознавање примероци, λ се пресметува само еднаш на основа на проценетите точности на класификаторите од ансамблот. Неспецифицираната мера g_λ пак мора да се одредува за секој примерок поодделно, бидејќи поради (2.56) и мерите g^i мора да се подредат соодветно на актуелните излези од секој од класификаторите.

Според ова комбинирањето на класификатори со употреба на неспецифициран интеграл се одвива во следниве чекори:

1. Фиксирај ги неспецифицираните густини $g^1, \dots, g^L$ според важноста на секој од класификаторите.
2. Пресметај го λ како единствениот реален корен поголем од -1 на равенката (2.60).
3. За секој примерок x :
4. За секоја од можните класи j ($j = 1, \dots, k$):
5. Подреди ги излезите од класификаторите $e_i(x) = m_i^j$, $i = 1, \dots, L$ кои ја изразуваат поддршката на хипотезата дека примерокот x е од класата j во опаѓачки редослед така да: $m_{i1}^j \geq m_{i2}^j \geq \dots \geq m_{iL}^j$;
6. Подреди ги соодветно и густините $g^{i1}, \dots, g^{iL}$;
7. Пресметај ги рекурзивно $g(A_i)$ според (2.58) и (2.59);

8. Пресметај го неспецифицираниот интеграл според (2.57) каде $h(y_i)$ се заменува со излезот од класификаторот m_i^j и n со бројот на класификатори L . Добиената вредност φ_j се интерпретира како степен на фузионирана поддршка за класата j (односно дека примерокот x е до класата j);
9. Распреди го примерокот во класата за која вредноста на интегралот е најголема, односно одреди ја конечната класа според правилото за максимална припадност $Q(x) = \arg \max_{j=1, \dots, k} (\varphi_j)$.

Клучна за комбинирањето со неспецифициран интеграл е вредноста на λ која се добива од неспецифицираните густини кои се одредуваат на основа на проценетата важност на индивидуалните класификатори. Важноста на секој од класификаторите може да се одреди според проценетата точност на секој од нив според (1.3), користејќи го множеството за валидација. Точноста на класификаторот може да се пресмета и од вредностите во конфузионата матрица (2.9) за дадениот класификатор, според (2.13). При ваквото одредување на неспецифицираните густини, точноста се одредува интегрално за целиот класификатор, независно од класата на примероците. На овој начин се проценува веројатноста дека произволен примерок ќе биде правилно класификуван независно од неговата вистинска класа. При комбинирањето со неспецифициран интеграл тоа значи дека класификаторите се третираат како симетрични класификатори за кои се одредува единствена густина на распределба за сите класи, односно се одредува единствено λ за ансамлот. Со вака одреденото единствено λ потоа се одредуваат поддршките за секоја од можните класи за секој примерок кој се препознава.

2.15.1 Модификација на комбинирањето со неспецифициран интеграл

Во практика класификаторите не мора да имаат симетрични перформанси во однос на сите класи, односно често се случува класификаторот подобро да ги препознава примероците од една, а послабо од некоја друга класа. Тоа значи дека во основа перформансите на класификаторот зависат од класата на примерокот кој се препознава, односно за секоја класа класификаторот ќе има различна точност. Некој друг од класификаторите во ансамлот може да има спротивно поведение, односно подобро да ги препознава примероците од класата чии примероци послабо ги препознава првиот класификатор, иако можеби генерално е послаб од него. Ова често се среќава во практика кога класификаторите имаат слични генерални перформанси. При комбинирањето на вакви класификатори логично би било, при одредувањето дали непоз-

натиот примерок е припадник на одредена класа да се даде поголема предност на оние класификатори кои подобро ја препознаваат токму таа класа, а не на генерално подобрите класификатори во ансамблот.

Ако се земе ова предвид, за даден класификатор може да се процени точноста при препознавање на примероците од секоја од класите, односно за секој класификатор да се одредат k точности Rr_i за секоја од класите $i = 1, \dots, k$. За проценка на точноста на класификаторот за одредена класа повторно може да се искористи конфузионата матрица. Точноста на класификаторот за класата i може да се изрази и преку бројот на погрешно препознаени примероци како:

$$Rr_i = 1 - \frac{N_{err}^i}{N}, \quad (2.62)$$

каде N_{err}^i е бројот на погрешно препознаени примероци (од аспект на класата i), а N е вкупниот број на примероци во множеството за валидација. Погрешно препознаените примероци ги опфаќаат оние примероци од другите класи кои биле препознаени како припадници на класата i (N_{fp}^i) и примероците од класата i кои не биле правилно препознаени (N_{fn}^i), односно биле распоредени во некоја од другите класи. Во конфузионата матрица (2.63) погрешно препознаените примероци за класата i се регистрираат во елементите од i -тата редица и i -тата колона без дијагоналниот елемент.

$$CM = \begin{bmatrix} n_{11} & \cdots & n_{1i} & \cdots & n_{1k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ n_{i1} & \cdots & n_{ii} & \cdots & n_{ik} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ n_{k1} & \cdots & n_{ki} & \cdots & n_{kk} \end{bmatrix} \quad (2.63)$$

Притоа, примероците од другите класи погрешно препознаени како припадници на класата i се регистрирани во i -тата колона:

$$N_{fp}^i = \sum_{j=1, j \neq i}^k n_{ji}, \quad (2.64)$$

додека примероците од класата i погрешно препознаени како припадници на другите класи се регистрирани во i -тата редица:

$$N_{fn}^i = \sum_{j=1, j \neq i}^k n_{ij}. \quad (2.65)$$

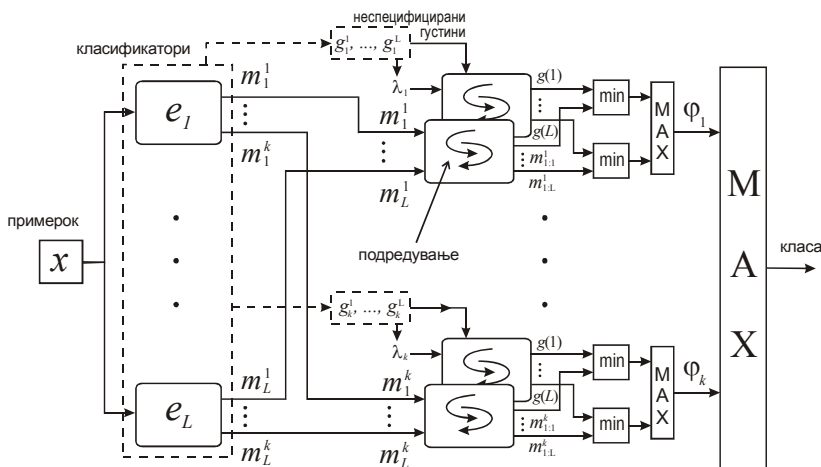
Конечно точноста на m -тиот класификатор од аспект на класата i може да се одреди како:

$$\begin{aligned}
 Rr_i^{(m)} &= 1 - \frac{\sum_{j=1, j \neq i}^k n_{ij}^{(m)} + \sum_{j=1, j \neq i}^k n_{ji}^{(m)}}{\sum_{i=1}^k \sum_{j=1}^k n_{ij}^{(m)}} \\
 &= \frac{\sum_{i=1}^k \sum_{j=1}^k n_{ij}^{(m)} - \sum_{j=1}^k (n_{ij}^{(m)} + n_{ji}^{(m)}) + 2 \cdot n_{ii}^{(m)}}{\sum_{i=1}^k \sum_{j=1}^k n_{ij}^{(m)}}.
 \end{aligned} \tag{2.66}$$

Ако веќе можеме да одредиме (потенцијално) различни точности на секој од класификаторите за секоја од класите, тогаш и проценетата важност на секој од индивидуалните класификатори ќе биде различна зависно од хипотезата која се тестира: дали непознатиот примерок е припадник на дадена класа. Според вака проценетите важности на класификаторите може да се одредат различни неспецифицирани густини за секоја од можните класи, односно да се проценат $k \cdot L$ густини, за секој класификатор и за секоја од класите: g_j^i , $i = 1, \dots, L$, $j = 1, \dots, k$. Од овие густини може да се добијат k различни λ_j , $j = 1, \dots, k$, односно различно λ_j за секоја од можните класи. Постапката за комбинирање на класификатори со неспецифициран интеграл би се променила во:

1. За секоја од можните класи j ($j = 1, \dots, k$):
2. Фиксирај ги неспецифицираните густини $g_j^1, \dots, g_j^L$ според важноста на секој од класификаторите при препознавањето на примероци од класата j ;
3. Пресметај го λ_j како единствениот реален корен поголем од -1 на равенката (2.60) во која g^i се заменува со g_j^i и λ со λ_j ;
4. За секој примерок x :
5. За секоја од можните класи j ($j = 1, \dots, k$):
6. Подреди ги излезите од класификаторите $e_i(x) = m_i^j$, $i = 1, \dots, L$ кои ја изразуваат поддршката на хипотезата дека примерокот x е од класата j во опаѓачки редослед така да: $m_{i1}^j \geq m_{i2}^j \geq \dots \geq m_{iL}^j$;
7. Подреди ги соодветно и густините $g_j^{i1}, \dots, g_j^{iL}$;
8. Пресметај ги рекурзивно $g_j(A_i)$ според (2.58) и (2.59) во кои g се заменува со g_j ;

9. Пресметај го неспецифицираниот интеграл φ_j според (2.57) каде $h(y_i)$ се заменува со излезот од класификаторот m_i^j и n со бројот на класификатори L .
10. Распореди го примерокот во класата за која вредноста на интегралот е најголема, односно одреди ја конечната класа според правилото за максимална припадност $Q(x) = \arg \max_{j=1, \dots, k} (\varphi_j)$.



Слика 2.4. Комбинирање со неспецифициран интеграл

Во практика разликата во резултатите при комбинирањето со неспецифициран интеграл според првиот и вториот пристап најчесто е минимална. Вториот пристап за пресметување на неспецифицираните густини има позначителна предност само кога се комбинираат класификатори со слични перформанси кај кои е поверојатно да се случи едниот од нив да е подобар за едни а другиот за други класи. Кога се комбинираат класификатори со значително поинакви перформанси најчесто подобрите класификатори се подобри во препознавањето на секоја од класите од послабите класификатори, така што одредувањето на густините зависно од класите не дава значителни подобрувања.

Изборот на неспецифицираните густини интерпретирани како важност на секој од класификаторите е клучна за комбинирањето. Доколку за ова се користат проценетите рати на препознавање на класификатори со високи перформанси, во минимумот од (2.57) излезот од класификаторот секогаш ќе биде помал од неспецифицираните мери, па комбинирањето се сведува на комбинирање со избор на максимум. Затоа, се чини дека е посоодветно за неспецифицираните густини да се земат пропорционални вредности помали од ратите на препознавање на индивидуалните класификатори, на пример половина од нив (2.61).

2.16 Комбинирање со тежинско усреднување

Најчесто употребуваниот метод за комбинирање на класификатори во практика е земањето на аритметичката средина од излезите од сите класификатори. Усреднувањето со аритметичка средина е специјален случај на *линеарно комбинирање* кое е синоним за тежинското усреднување. Линеарното комбинирање е доста погодно за формална анализа, а се чини дека и во практика доста често добро работи.

Tumer [145] покажал дека додадената грешка при аритметичко усреднување на L класификатори е:

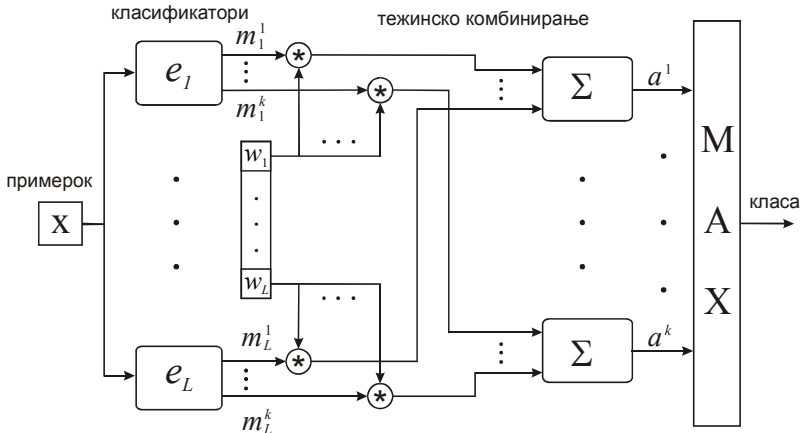
$$E_{ave}^{add} = \frac{E_{add}}{L}, \quad (2.67)$$

каде E_{add} е додадената грешка од секој од класификаторите, односно вредноста за која излезот од класификаторот отстапува од посакуваната вредност. Од (2.67) се гледа дека со едноставно аритметичко усреднување на L класификатори додадената грешка се намалува L пати. Bishop [10] покажал дека со едноставното аритметичко усреднување и квадратната грешка се намалува L пати во однос на збирот од квадратните грешки на индивидуалните класификатори. Но, обете тврдења важат само под претпоставка дека додадените грешки имаат нулова средна вредност и дека грешките на индивидуалните класификатори се меѓусебно некорелирани. Во практика, намалувањето на грешката обично е многу помало од оваа теоретски добиена вредност. Ова главно се должи на фактот што класификаторите, колку и да се меѓусебно различни по конструкција или обучувани на различни начини или со различни примероци за обука, сепак се доста корелирани. Може лесно да се докаже дека аритметичкото усреднување не може да предизвика зголемување на додадената грешка [10].

При комбинирањето со аритметичка средина, сите класификатори се третираат исто и имаат подеднаков удел во добиениот комбиниран излез. Доколку имаме познавање дека некој од класификаторите е подобар, односно дека можеме повеќе да веруваме на неговите одлуки, логично би било да му дадеме предност при комбинирањето. Ова неједноставно се прави со множење на излезите од класификаторите со тежински фактор која ја изразува доверливоста кон дадениот класификатор (слика 2.5). Функцијата за тежинско комбинирање ја има формата:

$$F_{WA}(m_1, \dots, m_L) = \sum_{i=1}^L w_i m_i, \quad (2.68)$$

каде w_i е тежината придружена на i -тиот класификатор.



Слика 2.5. Тежинско комбинирање

Логично е да се очекува дека тежинското комбинирање со соодветно избрани тежини кои ќе ги фаворизираат подобрите класификатори уште повеќе ќе ја намали грешката при комбинирањето. Тежините може да бидат доделени како субјективна оценка на надворешен експерт кој ги оценува перформансите на индивидуалните класификатори, но најчесто се одредуваат на основа на проценетите перформанси. Оценувањето на перформансите на класификаторите и одредувањето на тежинските фактори може да се изведе по различни методи.

2.16.1 Тежинско комбинирање со обопштен комитет

Тежинското комбинирање со обопштен комитет се изведува според (2.94) при што тежините се оптимизираат така да вкупната грешка при ваквото комбинирање биде минимална, користејќи го притоа множеството за валидација за минимизација на грешката. Познавајќи ги излезите на сите класификатори за секој од примероците од множеството за валидација, истите можеме да ги претставиме како збир од точната (посакуваната) вредност на излезот и грешката (отстапувањето):

$$m_i(x) = h(x) + \delta_i(x). \quad (2.69)$$

Ако (2.69) се стави во (2.68), за излезот од комбинацијата имаме:

$$a(x) = \sum_{i=1}^L w_i (h(x) + \delta_i(x)) = h(x) + \sum_{i=1}^L w_i \delta_i(x), \quad (2.70)$$

каде $\sum_{i=1}^L w_i \delta_i(x)$ го означува отстапувањето од посакуваниот излез при комбинирањето. Ако со C се означи корелационата матрица на грешки

$C_{ij} = E[\delta_i(x)\delta_j(x)]$, тогаш грешката на обопштениот комитет може да се напише како:

$$\begin{aligned} E_{GC} &= E\left[m_{GC}(x) - h(x)\right]^2 = \\ &= E\left[\left(\sum_{i=1}^L w_i \delta_i\right)\left(\sum_{j=1}^L w_j \delta_j\right)\right] = \sum_{i=1}^L \sum_{j=1}^L \delta_i \delta_j C_{ij} \end{aligned} \quad (2.71)$$

од каде оптималните вредности на тежините w_i може да се одредат со минимизирање на грешката (2.71) [10]. За решавање на овој проблем, за да се избегне тривијалното решение $w_i = 0, \forall i = 1, \dots, L$ неопходно е да се воведо дополнително ограничување на тежините, на пример:

$$\sum_{i=1}^L w_i = 1. \quad (2.72)$$

Со примена на лагранжовиот множител λ [10] за да се наметнат ограничувањата (2.72), се добива дека минимумот се јавува за:

$$2 \sum_{j=1}^L w_j C_{ij} + \lambda = 0, \quad (2.73)$$

од каде решението е:

$$w_i = -\frac{\lambda}{2} \sum_{j=1}^L (C^{-1})_{ij}. \quad (2.74)$$

По наоѓањето на λ со замена на (2.74) во (2.72), конечното решение за тежините е:

$$w_i = \frac{\sum_{j=1}^L (C^{-1})_{ij}}{\sum_{k=1}^L \sum_{j=1}^L (C^{-1})_{kj}}. \quad (2.75)$$

Корелационата матрица C се пресметува како апроксимација на вистинската корелациона матрица користејќи конечен број познати примероци како:

$$C_{ij} \equiv \frac{1}{N} \sum_{n=1}^N \left(m_i(x^n) - t^n\right) \left(m_j(x^n) - t^n\right), \quad (2.76)$$

каде t^n е целната вредност на дадениот излез за примерокот x^n .

Според ова правилото за комбинирање на класификатори според методот на обопштен комитет е:

$$Q(x) = \arg \max_{j=1, \dots, k} \sum_{i=1}^L w_i m_i^j. \quad (2.77)$$

Hu et al. [75] за одредување на тежините го предлагаат изразот:

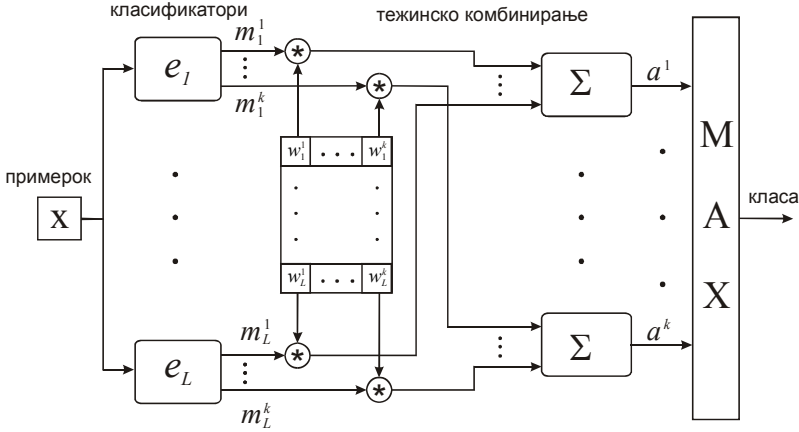
$$w_i = \frac{1/\sigma_i^2}{\sum_{j=1}^L (1/\sigma_j^2)}, \quad (2.78)$$

каде σ_i е стандардната девијација на излезот од i -тиот класификатор, со што се дава поголема тежина на оние класификатори кои имаат помало расејување на излезот. Изразот (2.78) е специјален случај на (2.75), во кој се зема дека класификаторите се потполно независни, па корелационата матрица се сведува на дијагонална со $C_{ii} = \sigma_i^2$ и $C_{ij} = 0$, $i \neq j$, односно не се земаат предвид меѓузависностите на класификаторите.

2.16.2 Тежинско комбинирање со модифициран обопштен комитет

Обопштениот комитет предложен од Bishop [10] и одредувањето на тежините на горенаведениот начин не ги зема предвид различните излези од индивидуалните класификатори и е доследен само кога се комбинираат класификатори со еден излез. При комбинирањето на класификатори со повеќе излези, според предложената шема за секој од класификаторите се одредува единствена тежина со која се множат сите излези од дадениот класификатор. За одредувањето на овие тежини при одредувањето на корелационата матрица во изразот (2.76) се работи со векторот составен од сите излези од класификаторот и соодветниот вектор од посакувани излезни вредности. При ваквата постапка доколку за одреден класификатор биде проценето дека е подобар од останатите и соодветно му биде доделена поголема тежина, ќе се смета дека тој е подобар од останатите во препознавањето примероци од која било класа. Меѓутоа, во практика често се случува некој од класификаторите добро да ја препознава класата c_i но послабо класата c_j , додека некој од другите класификатори да има спротивно однесување, односно подобро да ја препознава класата c_j , а послабо класата c_i . Бидејќи според шемата за тежинско комбинирање на класификатори сите излези од секој од класификаторите независно се третираат, можно е и секој од нив индивидуално да се модифицира множејќи го со специфичен тежински коефициент соодветен за дадениот излез од дадениот класификатор. Во ваквиот пристап повеќе не би се работело со вектор од онолку тежински

коэффициенти колку што има класификатори во ансамблот, туку со матрица од $L \times k$ коэффициенти со соодветен коэффициент за секој од излезите на секој од класификаторите (слика 2.6). За секој од излезите на секој од класификаторите би бил придружен соодветен тежински коэффициент кој би ја одразувал доверливоста на класификаторот при препознавањето примероци од дадената класа.



Слика 2.6. Комбинирање со модифициран обопштен комитет

Одредувањето на тежините во овој случај е аналогно на претходно опишаната постапката само што секој од излезите на секој од класификаторите се набљудува како индивидуален класификатор за една класа. Соодветно, тежините се пресметуваат независно за секоја од колоните на матрицата со тежини, односно во однос на j -тите излези од сите L класификатори:

$$w_i^j = \frac{\sum_{r=1}^L (C_j^{-1})_{ir}}{\sum_{s=1}^L \sum_{r=1}^L (C_j^{-1})_{sr}}, \quad j = 1, \dots, k. \quad (2.79)$$

C_j е корелационата матрица на j -тиот излез од сите класификатори, односно корелациона матрица на ансамблот од класификатори за класата j , чии елементи се добиваат како:

$$(C_j)_{sr} \equiv \frac{1}{N} \sum_{n=1}^N (m_s^j(x^n) - \text{Ind}(x^n, j)) (m_r^j(x^n) - \text{Ind}(x^n, j)). \quad (2.80)$$

Со ова, правилото за комбинирање на класификатори според методот на модифициран обопштен комитет станува:

$$Q(x) = \arg \max_{j=1, \dots, k} \sum_{i=1}^L w_i^j m_i^j \quad (2.81)$$

На овој начин овозможено е преку тежините да се моделираат индивидуалните предности на некои од класификаторите во препознавањето на примероците од одредени класи.

Главниот проблем при имплементацијата на комбинирањето со обопштен комитет е одредувањето на корелационата матрица располагајќи притоа со ограничено множество за валидација. Доколку бројот на примероци во множеството за валидација е премал, или кога класификаторите кои се комбинираат се исклучително корелирани, лесно може да се случи да се добие корелациона матрица која е блиску до сингуларна со што нејзината инверзија ќе биде нестабилна и ќе повлече непрецизна процена на тежините. Еден начин да се надмине ова е при пресметувањето на тежините да се земат предвид само дијагоналните елементи на корелационата матрица [116], но со ова се занемаруваат меѓузависностите на класификаторите. Исто така, колку што класификаторите кои се комбинираат се со подобри перформанси толку е поголема веројатноста корелационата матрица на нивните излези да биде блиска до сингуларна [96]. Ова значи дека со употребата на подобри класификатори во ансамблот, ефектот од нивното тежинско комбинирање опаѓа главно поради тоа што станува сè потешко да се одредат нивните индивидуални предности и недостатоци на ограничен примерок. За одредување на корелационата матрица од особен интерес се примероците од множеството за валидација за кои класификаторите имаат различни одлуки, односно се некорелирани. Примероците кои сите класификатори правилно ги препознаваат или за кои сите грешат, не придонесуваат многу кон одлуката на кој од класификаторите да му се даде предност при комбинирањето. Со подобрувањето на точноста на индивидуалните класификатори, се намалува бројот на примероци за кои класификаторот греша, а со тоа се намалува бројот на случаи во кои класификаторите нема да се сложуваат во одлуките, што директно влијае на прецизноста на одредувањето на корелационата матрица.

Во случаите кога корелационата матрица е блиска до сингуларна, често се случува при одредувањето на тежините според (2.75) или (2.79) да се добијат негативни тежини за некои од класификаторите. Проблемот е што критериумот (2.72) не ги спречува тежините да станат многу мали (па и негативни) или многу големи, со што се губи на ефикасноста на комбинирањето [10]. Затоа во практика се наметнува потребата од воведување дополнителни ограничувања за тежините на пример $w_i > 0$. Но, во ваквиот случај одредувањето на оптималните те-

жини за вака зададени ограничувања е доста покомплициран проблем кој може да се реши со употреба на линеарно програмирање.

Ако попрецизно се анализираат изразите (2.76) и (2.80) се гледа дека за одредување при корелационите матрици се земаат отстапувањата на класификаторите од посакуваната вредност за дадениот примерок, со што секој од нив се разгледува индивидуално. При ова еднакво се вреднуваат отстапувањата и во позитивна и во негативна насока, што е можеби соодветно за регресијата, но не секогаш најсоодветно кај класификацијата. Со ова практично се стреми со одредување на соодветни тежини да се оптимизираат комбинираниот излез од тежинските суми a^i да бидат што поблиску кон нивните оптимални посакувани вредности пред да влезат во максимум селекторот (слика 2.6). Меѓутоа, од аспект на финалниот резултат на комбинирањето на класификаторите важна е конечната одлука како излез од максимум селекторот. Според тоа, за правилна одлука, односно одредување на вистинската класа на непознатиот примерок, сосема доволно е тежинската комбинација за соодветната класа (a^i) да биде поголема од сите останати ($a^i > a^j, \forall i \neq j$). Од овој аспект за грешка се сметаат само отстапувањата на крајниот резултат, односно кога примерокот не е правилно класификуван, што е познато како 0-1 функција на загуба.

Од математички аспект, максимум селекторот е нелинеарна недиференцијабилна функција, и токму поради ваквата негова природа, аналитичкото моделирање на вака поставениот проблем на оптимизација на тежините е невозможно. Сепак оптималните тежини би можеле да се одредат применувајќи некоја од техниките за нелинеарна оптимизација, програмирање со ограничувања или генетски алгоритми.

2.16.3 Оптимизирани тежини за комбинирање со обопштен комитет

Оптимизираниот тежини за комбинирање со обопштен комитет всушност е начин на одредување на тежините со оптимизациски техники, не користејќи ја притоа корелационата матрица и земајќи ја предвид крајната одлука на класификаторот од максимум селекторот. Позната оптимизациска техника за наоѓање на минимум на мултиваријантна функција која не мора да биде ниту континуална ниту диференцијабилна е алгоритмот на Хук-Џивс (Hooke-Jeeves) [74]. Со адаптација на алгоритмот предложен од Кауре [86], вклучувајќи ги и подобрувањата предложени од Bell и Pike [6] и од Tomlin и Smith [142], и додавајќи дополнителни услови кои го држат оптимумот по кој се трага во претходно зададени граници, добиен е алгоритам за одредување на оптималните тежини. Тој е применлив и за одредување на тежините на обопштениот комитет и за тежините на модифицираниот обопштен комитет.

Во првиот случај се оптимизира функцијата (2.77) барајќи оптимални вредности на тежините w_i за кои се прават најмалку грешки, односно критериумот за минимизација е бројот на грешки при комбинирањето на класификаторите за препознавање примероците од множеството за валидација. Функцијата (2.77) има L слободни променливи кои ги оптимизираме, но со вклучувањето на ограничувањето (2.72) произлегува дека тие се зависни и дека се работи за ефективни $L-1$ степени на слобода. Слично, и при одредувањето на тежините на модифицираниот обопштен комитет според (2.81) која има Lk тежини, поради ограничувањата се работи за $(L-1)k$ степени на слобода.

Одредувањето на тежините користејќи оптимизациски методи како Хук-Џивс немаат некои од слабостите на оригиналниот пристап за одредувањето на тежините, пред сè бидејќи не ја користат корелационата матрица и нејзината инверзија. Неможноста доволно прецизно да се одреди корелационата матрица поради ограничениот број на примероци со помош на кои се врши нејзина приближна проценка, е главниот извор на грешка во одредувањето на тежинските фактори. Недостатоците на оптимизациските методи се што работат побавно поради итеративната постапка за одредување на тежините. Исто така, тие се осетливи на почетните услови и чекорот на итерациите, а не гарантираат дека ќе го пронајдат глобалниот минимум. Генетските алгоритми би биле нешто подобри во овој поглед, но тие се покомплицирани за имплементација и доста побавни.

2.16.4 Тежинско комбинирање според проценети перформанси на индивидуалните класификатори

Веројатно наједноставен начин за поставување на тежинските коефициенти кај тежинското комбинирање е тие да се одредат од конфузионите матрици на секој од класификаторите според (2.66) со нивна дополнителна нормализација за да се задоволи условот (2.72). Со ова се одредуваат онолку тежински фактори колку што вкупно има излези од сите класификатори во ансамблот како и кај модифицираниот обопштен комитет. Притоа поголеми тежини добиваат излезите од оние класификатори кои биле поуспешни во одредувањето на вистинската класа на примероците од множеството за валидација. Ваквиот начин на одредување на тежините е многу помалку осетлив на ограничениот број на примероци во множеството за валидација и често во практика се покажува како подобар од обопштениот комитет, но сепак нешто послаб од модифицираниот обопштен комитет кој ги зема предвид и меѓузависностите на излезите на класификаторите.

2.17 Тежинско комбинирање со динамичко одредување на тежините

Кај претходно изложените тежински комбинирања, тежинските фактори се одредуваат на основа на снименото однесување на класификаторите на множеството за валидација и истите еднаш фиксирани тежини се користат при комбинирањето на класификаторите при препознавањето на непознатите примероци. Динамичкото одредување на тежините пак, подразбира одредување на тежинските фактори за секој примерок одделно. За разлика од тежинските комбинирања со фиксирани тежини кои се обучени методи за комбинирање, кај динамичкото тежинско комбинирање не е потребана обука на класификаторот. Кај него, тежините се одредуваат динамички за секој случај одделно на основа на излезите на класификаторите. Динамичкото одредување на тежините почива на т.н. *сигурносии* на излезот од класификаторот.

Ако излезот од класификаторот $y = f(x)$ може да се интерпретира како веројатност дека примерокот е припадник на дадена класа, тогаш може да се каже дека како што y се приближува кон 1, сигурноста дека примерокот е навистина од дадената класата е сè поголема. Како пак излезот од класификаторот се приближува кон 0, може да се смета дека сигурноста дека примерокот **не** е од дадената класата е сè поголема. Според овој пристап се смета дека сигурноста на излезот од класификаторот е поголема колку што неговата одлука е појасна било во поддршката било во отфрлувањето на претпоставката дека примерокот е припадник на класата. Сигурноста на излезот е помала кога и самиот класификатор е неодлучен. Ќе ја квантифицираме оваа нотација и воведеме мерка на сигурност на излезот на класификаторот $c(y)$ како:

$$c(y) = \begin{cases} y & \text{ако } y \geq 1/2 \\ 1 - y & \text{инаку} \end{cases} \quad (2.82)$$

Сигурноста расте за вредности на излезот $y < 0.5$ како што y опаѓа, и за вредностите на $y \geq 0.5$ како што y расте. Според ова можеме да кажеме дека излезот од еден класификатор y_1 е помалку сигурен од излезот од друг класификатор y_2 , ако $c(y_1) < c(y_2)$. Вака дефинираната сигурност се однесува симетрично во однос на позитивните и негативните одлуки, односно сигурноста на излез со вредност 0.1 и излез со вредност 0.9 се исти, но нивните одлуки за кои се одредува сигурноста се различни. Ако наместо фиксирани тежини, дозволиме тежините динамички да се одредуваат пропорционално на сигурноста на секој од излезите за актуелниот примерок кој се препознава, ќе добиеме комбинирање со динамички одредени тежини. Комбинирањето се изведува според изразот (2.81) при што тежините се пресметуваат како:

$$w_i^j = \frac{c(m_i^j(x))}{\sum_{k=1}^L c(m_k^j(x))}. \quad (2.83)$$

Вака дефинираните тежини го задоволуваат условот (2.72), со што повторно имаме тежинско комбинирање на еден вид обопштен комитет. Но, во овој случај, матрицата со тежински фактори се пресметува за секој примерок одделно во обид да се изнајде најдобрата одлука за секој поодделен примерок кој се препознава, наместо статички одредени тежини кои даваат оптимални резултати при комбинирањето во однос на множеството за валидација. На овој начин влијанието на одлуката од секој од класификаторите е пропорционална на неговата сигурност во истата одлука. Така на пример, вредност блиска до 0.5 би влијала многу малку во сумата од (2.83), додека сигурните вредности како 0.99 (или 0.01) би доминирале во сумата покрај другите помалку сигурни вредности. Изложениов метод за комбинирање е инспириран од идејата за употреба на степенот на согласност на група на класификатори при одредување на сигурноста на нивната комбинирана одлука [144] само што во овој пристап се користи степенот на сигурност на секој индивидуален класификатор, поточно секој од неговите излези поодделно при добивањето на финалната одлука.

Предноста на тежинското комбинирање со динамичкото одредување на тежините е што за негова имплементација не е потребна обука, ниту множество за валидација и може да се адаптира на секој поодделен случај. Сепак ваквата комбинациона шема е „без меморија“ и не може да ги искористи меѓузависностите на излезите од класификаторите кои релативно поуспешно се моделираат при одредувањето на тежините на основа на снименото однесување на класификаторите на множеството за валидација. Тежинското комбинирање со динамичко одредување на тежините има предност при комбинирањето на класификатори кои се слабо корелирани. Како што класификаторите кои се комбинираат стануваат покорелирани, предноста преминува на страната на обучените шеми за тежинското комбинирање кои се во состојба нешто подобро да ги искористат меѓузависностите на класификаторите.

2.18 Комбинирање со шаблони за одлучување

Шаблоните за одлучување (Decision Templates – DT) претставуваат робусна шема за комбинирање на класификатори предложени од Kuncheva et al [96]. Комбинирањето со шаблони за одлучување се изведува преку споредување на излезите од класификаторите со карактеристични шаблони за секоја од класите. При комбинирањето шаблони-

те за одлучување ги користат **сите** излези од **сите** класификатори за одредувањето на комбинираната поддршка за секоја од класите. Овој пристап е драстично различен од повеќето други пристапи за комбинирање на класификатори кои ги користат само излезите за одредената класа од сите класификатори при одредувањето на комбинираната поддршка за дадената класа.

Нека $\{e_1, \dots, e_L\}$ е множество од L класификатори. Ако со $e_i(x) = [m_{i,1}(x), \dots, m_{i,k}(x)]^T$ ги означиме излезите од i -тиот класификатор, каде со m_i^t е означен степенот на верување дека примерокот x е од класата t според класификаторот e_i , излезите од сите класификатори можеме да ги организираме во матрица т.н. одлучувачки профил:

$$DP(x) = \begin{bmatrix} m_{1,1}(x) & \dots & m_{1,j}(x) & \dots & m_{1,k}(x) \\ \vdots & & \vdots & & \vdots \\ m_{i,1}(x) & \dots & m_{i,j}(x) & \dots & m_{i,k}(x) \\ \vdots & & \vdots & & \vdots \\ m_{L,1}(x) & \dots & m_{L,j}(x) & \dots & m_{L,k}(x) \end{bmatrix}. \quad (2.84)$$

Во оваа матрица во i -тата редица се наоѓаат поддршките кои i -тиот класификатор ги дава на хипотезите дека примерокот е од класата $j, j = 1, \dots, k$. Во j -тата колона пак се наоѓаат поддршките кои ги добила хипотезата дека примерокот е од класата j од секој од класификаторите $e_i, i = 1, \dots, L$.

Шаблоните за одлучување се добиваат испитувајќи го однесувањето на обучените класификатори на множеството за валидација $Z = \{z_1, \dots, z_N\}$. Секој од шаблоните за одлучување DT_i за класата i претставува $L \times k$ матрица:

$$DT_i = \begin{bmatrix} d_{1,1}^i & \dots & d_{1,j}^i & \dots & d_{1,k}^i \\ \vdots & & \vdots & & \vdots \\ d_{q,1}^i & \dots & d_{q,j}^i & \dots & d_{q,k}^i \\ \vdots & & \vdots & & \vdots \\ d_{L,1}^i & \dots & d_{L,j}^i & \dots & d_{L,k}^i \end{bmatrix} \quad (2.85)$$

во која секој од елементите се добива како:

$$d_{q,j}^i = \frac{\sum_{s=1}^N Ind(z_s, i) \cdot m_{q,j}(z_s)}{\sum_{s=1}^N Ind(z_s, i)}, \quad q = 1, \dots, L; j = 1, \dots, k, \quad (2.86)$$

каде:

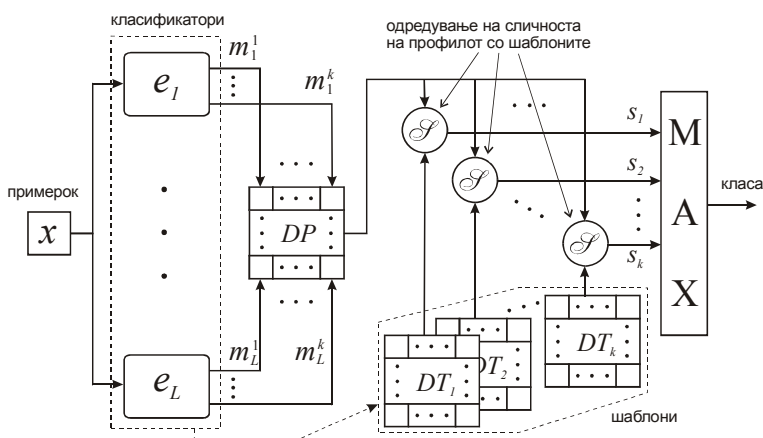
$$Ind(z_s, i) = \begin{cases} 1, & \text{ако } z_s \in c_i \\ 0, & \text{инаку} \end{cases} \quad (2.87)$$

е индикатор функција која има вредност 1 ако примерокот z_s е од класата i и 0 инаку. Секој од елементите на матрицата DT_i може да се опише и како очекувана вредност за дадениот излез од дадениот класификатор кога на влезот на класификаторите се појавува примерок од класата i . Шаблонот за одлучување за класата i е всушност првиот момент на профилите за примероците од подмножеството на множеството за валидација составено само од примероците од класата i .

Секој непознат примерок x се проследува до секој од класификаторите и добиениот профил од излезите од класификаторите $DP(x)$ се споредува со секој од шаблоните за одлучување DT_i со што се добива поддршката за секоја од класите:

$$\mu^i(x) = S(DT_i, DP(x)), i = 1, \dots, k \quad (2.88)$$

каде S се интерпретира како *мерка за сличност*. Според ова, колку што е поголема сличноста на профилот за дадениот примерок $x - DP(x)$ со шаблонот за класата $i - DT_i$, толку е поголема и поддршката за таа класа $\mu^i(x)$. Мерката за сличност може да се разбере во пошироката смисла на зборот како која било мера, степен на совпаѓање или каква било постапка со која може да се „спореди“ профилот (матрицата $DP(x)$) со секој од шаблоните (матриците $DT_i, i = 1, \dots, k$). На слика 2.7 е даден шематски приказ на функционирањето на комбинирање на класификатори со шаблони за одлучување. Шаблоните се пресметуваат однапред според (2.86) употребувајќи ги притоа примероците од множеството за валидација Z .



Слика 2.7. Комбинирање со шаблони за одлучување

Во зависност од употребената мерка за сличност и резултатите при комбинирањето може да бидат различни. Kuncheva et al. [96] предлагаат 11 различни мерки за сличност од кои 4 „вистински“ мери за сличност, 5 индекси на вклучување (содржење), 1 мерка за конзистентност и мерка базирана на евклидовото растојание меѓу матриците на профилот и шаблонот.

2.18.1 Неспецифицирани мерки за сличност

Ако аргументите на $\mathcal{S}$ од (2.88) ги третираме како неспецифицирани множества над некое универзално множество со $L \cdot k$ елементи, тогаш како мерка за сличност на профилот и шаблоните може да се употребат различни познати мерки за сличност [41]. Ако A и B се неспецифицирани множества над множеството $U = \{u_1, \dots, u_n\}$, тогаш меѓу нив може да се дефинираат следниве „вистински“ мерки за сличност:

$$S_1(A, B) = \frac{\|A \cap B\|}{\|A \cup B\|}, \quad (2.89)$$

каде $\|\zeta\|$ е релативната кардиналност на неспецифицираното множество ζ над U :

$$\|\zeta\| = \frac{1}{n} \sum_{i=1}^n \mu_{\zeta}(u_i). \quad (2.90)$$

$$S_2(A, B) = 1 - \|A \nabla B\|, \quad (2.91)$$

каде $A \nabla B$ е симетричната разлика дефинирана преку хаминговото растојание:

$$\mu_{A \nabla B}(u) = |\mu_A(u) - \mu_B(u)|, u \in U. \quad (2.92)$$

$$S_3(A, B) = 1 - \|A \Delta B\|, \quad (2.93)$$

каде:

$$\mu_{A \Delta B}(u) = \max\{\mu_{A \cap \bar{B}}(u), \mu_{\bar{A} \cap B}(u)\}, u \in U. \quad (2.94)$$

$$S_4(A, B) = 1 - \sup\{\mu_{A \nabla B}(u)\}. \quad (2.95)$$

Како мерка за сличност може да се употребат и следниве индекси на вклучување (содржење) на A во B :

$$I_1(A, B) = \frac{\|A \cap B\|}{\|A\|}. \quad (2.96)$$

$$I_2(A, B) = 1 - \|A - B\|, \quad (2.97)$$

каде $\| - \|$ ја претставува ограничената разлика:

$$\mu_{A \setminus B}(u) = \max\{0, \mu_A(u) - \mu_B(u)\}, u \in U. \quad (2.98)$$

$$I_3(A, B) = \|\bar{A} \cup B\|. \quad (2.99)$$

$$I_4(A, B) = \inf_{u \in U} \{\mu_{\overline{A \setminus B}}(u)\}. \quad (2.100)$$

$$I_5(A, B) = \inf_{u \in U} \{\mu_{\bar{A} \cup B}(u)\}. \quad (2.101)$$

Индексот на конзистентност пак е дефиниран како:

$$C(A, B) = \sup_{u \in U} \{\mu_{A \cap B}(u)\}. \quad (2.102)$$

За пресек и унија се користени минимумот и максимумот соодветно, а комплементот е сметан како $\mu_{\bar{A}}(u) = 1 - \mu_A(u)$.

Ако вредностите во матриците на профилот и шаблоните се интерпретираат како координати на точка во $L \cdot k$ димензионален метрички простор, тогаш нивната сличност може да се процени и со употреба на некоја од многуте мерки за растојание во метрички простори. При ова треба да се има предвид дека измереното растојание може да се интерпретира како мерка за „различност“. Доколку вредностите во матриците може да се интерпретираат како веројатности, односно се во опсегот $[0, 1]$, како мерка за сличност соодветна на растојанието најчесто се зема комплементот на растојанието. Без ограничување на опсегот на вредностите во матриците пак, сличноста може да се одреди и како реципрочна вредност на растојанието или најчесто како:

$$S(A, B) = \frac{1}{1 + D(A, B)} \quad (2.103)$$

за да се избегне делењето со нула.

Користејќи ги дефинициите на неколку мерки за растојание може да се изведат соодветни сличности кои може да се употребат при комбинирањето на класификатори со шаблони за одлучување. При пресметувањето на растојанијата, елементите од $L \times k$ матриците DP и DT_i се третираат како $L \cdot k$ димензионални вектори. Ако имаме $L \times k$ матрица со елементи m_{ij} , $i = 1, \dots, L$, $j = 1, \dots, k$, истата можеме да ја претставиме како вектор $\mathbf{x}$ со елементи x_i , $i = 1, \dots, n$; $n = L \cdot k$, и да ги дефинираме растојанијата дадени во следните поглавја.

2.18.2 Евклидово растојание

$$d_{euclid}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.104)$$

Евклидовото растојание го претставува најкусото праволиниско растојание помеѓу две точки во простор. Евклидовото растојание претставува наједноставен начин за груба проценка на „различноста“ на два примерока претставени како повеќедимензионални вектори.

2.18.3 Менхетен растојание

$$d_{manhattan}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i| \quad (2.105)$$

Менхетен (Manhattan) растојанието е еднакво на патот што треба да се измине за да се стигне од едната до другата точка притоа движејќи се само паралелно со координатните оски. Името го добило според растојанието кое треба да го измине пешак од едно до друго место во Менхетен – Њу Јорк, движејќи се по улиците кои се ориентирани од исток кон запад и од север кон југ.

2.18.4 Канбера растојание

$$d_{canberra}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \frac{|x_i - y_i|}{(x_i + y_i)} \quad (2.106)$$

Канбера растојанието е изведено од Канбера (Canberra) метриката според Lance & Williams (1967) [99].

2.18.5 Аглово одвојување

$$d_{angle}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \frac{x_i y_i}{\sqrt{\sum_{j=1}^n x_j^2 \sum_{j=1}^n y_j^2}} = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad (2.107)$$

Мерка за сличност може да биде и аголот помеѓу двата вектора. Изразот (2.107) го дава практично косинусот на аголот помеѓу двата вектора и претставува мерка за сличност а не растојание.

2.18.6 Чебишев растојание

$$d_{chebychev}(\mathbf{x}, \mathbf{y}) = \max_{i=1, \dots, n} |x_i - y_i| \quad (2.108)$$

Растојанието според Чебишев познато и како максимално растојание, е дефинирано како максималното растојание помеѓу кој било пар координати на точките.

2.18.7 Минковски растојание

$$d_{minkowsky}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^q \right)^{\frac{1}{q}} \quad (2.109)$$

Растојанието според Минковски (Minkowsky) е генерализирано растојание од кое за $q = 1$ се добива Менхетен растојанието, за $q = 2$ – Евклидовото растојание, а за $q = \infty$ – Чебишев растојанието.

2.18.8 χ^2 растојание

$$d_{\chi^2}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n \frac{1}{\sum_{k=1}^R x_i} \left(\frac{x_i}{\sum_{j=1}^n x_j} - \frac{y_i}{\sum_{j=1}^n y_j} \right)^2} \quad (2.110)$$

χ^2 растојанието е изведено од χ^2 метриката [106][157] и покрај вредностите на самите вектори ги користи и просечните вредности на секоја од компонентите (координатите) на векторите над целата популација.

2.18.9 Квадратно χ^2 растојание

$$d_{\chi^2 sq}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{|x_i + y_i|} \quad (2.111)$$

Квадратното χ^2 растојание често се користи како растојание при кластеризација [16][101].

2.18.10 Корд растојание

$$d_{chord}(\mathbf{x}, \mathbf{y}) = \sqrt{2 \left(1 - \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}} \right)} \quad (2.112)$$

Корд (Chord) растојанието е предложено од Orloci [110]. Тоа е евклидово растојание меѓу две точки по скалирање на векторите на единична должина, односно по делење на секој од нив со нормата или должината на векторот.

2.18.11 Квадратно Корд растојание

$$d_{chord^2}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (\sqrt{x_i} - \sqrt{y_i})^2} \quad (2.113)$$

Квадратното Корд растојание [102] често се користи при кластеризација во екологијата и во био-информатиката. Квадратното Корд растојание е применливо кога координатите се интерпретираат како веројатности или пропорционални удели, односно се ограничени во опсегот [0,1] и притоа растојанието прима вредности помеѓу 0 и 2, каде 0 означува објекти со идентични особини.

2.18.12 Хелинцер растојание

$$d_{hellinger}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n \left[\sqrt{\frac{x_i}{\sum_{j=1}^n x_j}} - \sqrt{\frac{y_i}{\sum_{j=1}^n y_j}} \right]^2} \quad (2.114)$$

Хелинцер (Hellinger) растојанието е предложено како алтернатива при кореспондентната анализа од страна на Rao [112][115] и претставува еден вид нормализирано Корд растојание.

2.18.13 Брај-Куртис растојание

$$d_{bray-curtis}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n (x_i + y_i)} \quad (2.115)$$

Брај-Куртис (Bray-Curtis) растојанието [168] често се користи во екологијата и претставува комплемент на сличноста по Мотука за квантитативни податоци и Sørensen индексот за бинарни податоци.

2.18.14 Јакард растојание II

$$d_{jaccard}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i - 2 \sum_{i=1}^n \min(x_i, y_i)}{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i - \sum_{i=1}^n \min(x_i, y_i)} \quad (2.116)$$

Јакард растојанието (Jaccard II) е комплемент на Јакардовиот коефициент на сличност за бинарни податоци и Ellenberg индексот на сличност за квантитативни податоци.

2.18.15 Процент на сличност

$$p_{sim}(\mathbf{x}, \mathbf{y}) = 200 \frac{\sum_{i=1}^n \min(x_i, y_i)}{\sum_{i=1}^n (x_i + y_i)} \quad (2.117)$$

Процентот на сличност ја изразува сличноста помеѓу два вектора при што 100% означува идентични вектори.

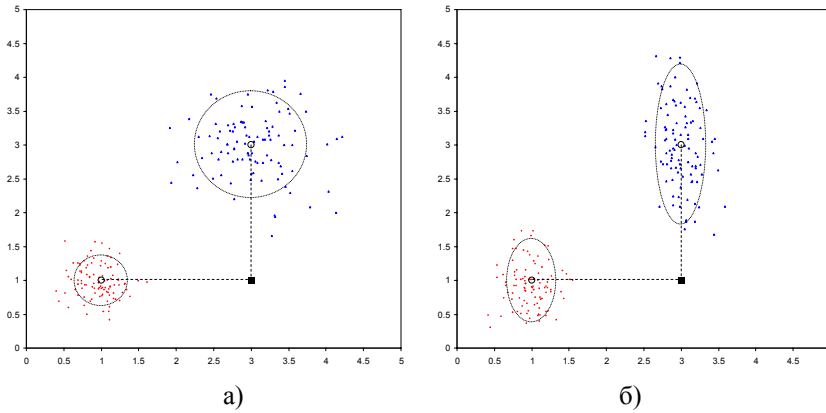
2.19 Комбинирање со проширени шаблони за одлучување

Досега изложените растојанија и мерки за сличност (освен донекаде χ^2 растојанието), споредуваат две изолирани точки, вектори или единки независно од популацијата на која и припаѓаат. Од статистиката (особено онаа која се применува во областите на биометријата) се познати повеќе методи за одредување на сличноста на две или повеќе популации, нивното раздвојување, како и сличноста односно релативната припадност на единка кон некоја од популациите [78][136]. Статистички гледано, при комбинирањето на класификатори според методот на шаблони за одлучување, споредувањето на профилот со шаблоните за секоја од класите всушност претставува одредување на припадност на единката – непознатиот примерок претставен преку профилот на излезите од класификаторите кон секоја од популациите – класите претставени со соодветен шаблон. Одовде се гледа дека во статистичка смисла шаблонот за дадена класа всушност ја репрезентира популацијата на сите примероци од дадена класа. Шаблонот се гради на основа на излезите од класификаторите кои се добиваат кога на нивниот влез ќе се по-

јави примерок од дадената класа, односно на јасно одвоена популација од примероци со позната класа (за ова се користи множеството за валидација). Одовде се гледа дека шаблонот претставува репрезент на популацијата. Според предложениот модел од Kuncheva популациите во шаблоните се претставени само со својот прв момент, односно средните вредности на секој од излезите од класификаторите за примероците од таа класа. Со ова практично целата популација (класа) се претставува со **еден единствен** „синтетски“ примерок како најдобар репрезент на дадената популација. Примерокот е синтетски бидејќи е изведен како средна вредност од сите единки во дадената популација и истиот во крајна линија воопшто не мора да постои ниту во обучувачкото ниту во множеството за валидација. Со ваквата претстава, припадноста на непознатиот примерок кон некоја од популациите се одредува со едноставно одредување на сличноста на неговиот профил (вектор во n -димензионален простор) со шаблонот на секоја од класите (повторно вектор во n -димензионален простор), односно како растојание или сличност меѓу два n -димензионални вектори.

Меѓутоа, во практика популациите кои претставуваат множество на повеќе единки, можат да имаат различна распределба во просторот. Растојанието на дадена единка до дадена популација не е самото растојание до нејзиниот центар (првиот момент) туку зависи и од самата распределба на популацијата во просторот. Во идеален случај ова растојание би требало да се изведе како некаква средина од растојанијата меѓу единката и сите членови на популацијата, но во практика ова е најчесто неизводливо. Затоа во практика се настојува популацијата да се претстави како некоја од познатите распределби и да се претстави преку параметрите на позната и однапред избрана распределба. За ова најчесто се употребува нормалната (гаусова) распределба.

Кога популациите се претставени како нормални распределби околу своите центри со униформно расејување по сите димензии, во одредувањето на припадноста на дадена единка кон некоја популација удел ќе има расејувањето. Така на пример, за примерокот кој се наоѓа на подеднакво растојание од центрите на две распределби, може да се смета дека е поблиску, односно дека е поверојатен припадник на популацијата која има поголемо расејување (слика 2.8a). Уште повеќе, расејувањето може да биде различно по различните оски од просторот, што повторно влијае на „бликоста“ на одредена единка до дадена популација (слика 2.8б). Според ова, ако се смета дека расејувањето е според законот на нормалната распределба, секоја популација ќе биде опишана со центарот и со стандардната девијација, односно шаблоните ќе треба да се прошират и со стандардните девијации.



Слика 2.8. Растојание на единка до две независни нормално распределени популации со различни стандардни девијации: а) еднакви по сите оски, б) различни по различните оски од просторот

Во ваков случај, за да се одреди растојанието на примерок од дадена популација посоодветно е да се употреби стандардизираното евклидово растојание кое го зема предвид и расејувањето на популацијата.

2.19.1 Стандардизирано евклидово растојание

Стандардизираното евклидово растојание го зема предвид и расејувањето, односно стандардната девијација на секоја од популациите при одредувањето на растојанието на единката до дадена популација.

$$d_{seuclid}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n \left(\frac{x_i - y_i}{\sigma_i} \right)^2} \quad (2.118)$$

каде $\mathbf{x}$ е примерокот односно во нашиот случај профилот, $\mathbf{y}$ е центарот на популацијата – во нашиот случај еден од шаблоните, а σ_i е стандардната девијација по i -тата димензија на популацијата, односно проширениот шаблон на расејувања за класата. Секој од проширените шаблони ST_i за класата i претставува $L \times k$ матрица од стандардни девијации:

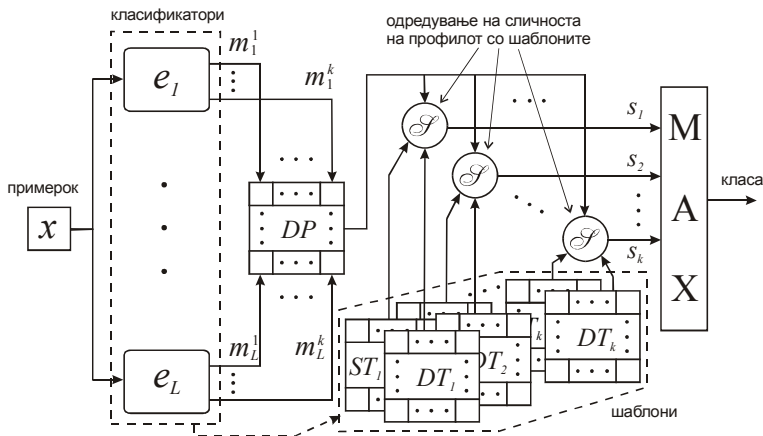
$$ST_i = \begin{bmatrix} \sigma_{1,1}^i & \cdots & \sigma_{1,j}^i & \cdots & \sigma_{1,k}^i \\ \vdots & & \vdots & & \vdots \\ \sigma_{q,1}^i & \cdots & \sigma_{q,j}^i & \cdots & \sigma_{q,k}^i \\ \vdots & & \vdots & & \vdots \\ \sigma_{L,1}^i & \cdots & \sigma_{L,j}^i & \cdots & \sigma_{L,k}^i \end{bmatrix} \quad (2.119)$$

во која секој од елементите се добива како:

$$\sigma_{q,j}^i = \sqrt{\frac{\sum_{s=1}^N \text{Ind}(z_s, i) \cdot (m_{q,j}(z_s))^2}{\sum_{s=1}^N \text{Ind}(z_s, i)} - (d_{q,j}^i)^2}, \quad (2.120)$$

каде $d_{q,j}^i$ е дадено со (2.86). Секој од елементите на матрицата ST_i ја претставува стандардната девијација на излезите од класификаторите за примероците од класата i . Проширениот шаблон за одлучување за секоја од класите е всушност вториот момент на профилите за примероците од подмножеството на множеството за валидација составено само од примероците од класата i .

За имплементација на стандардизираното евклидово растојание за комбинирање на класификатори по методот на шаблони за одлучување, шаблоните мора да се прошират со дополнителни шаблони за расејување за секоја од класите (2.119). На слика 2.9 е даден шематски приказ на функционирањето на комбинирањето на класификатори со проширени шаблони за одлучување.



Слика 2.9. Комбинирање со проширени шаблони за одлучување

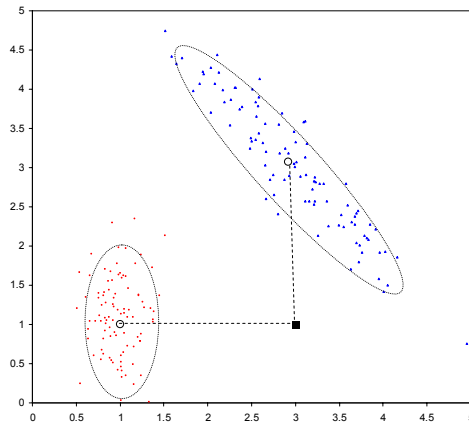
Шаблоните на расејување се пресметуваат однапред според (2.120) употребувајќи ги притоа примероците од множеството за валидација Z . Во овој случај влез во функцијата за одредување на сличноста $\mathcal{S}$ освен профилот и шаблонот DT_i е и матрицата со стандардни девијации ST_i .

2.19.2 Фрешеово F-растојание

Фрешеовото (Fréchet) F-растојание е слично на стандардното евклидово растојание и исто така го зема во предвид расејувањето на популацијата [78]:

$$d_{fréchet}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \frac{|x_i - y_i|}{\sigma_i} \quad (2.121)$$

Предноста на Фрешеовото и стандардизираното евклидово растојание во однос на другите претходно изложени растојанија и мерки за сличност е користењето на информацијата за расејувањето на популациите изразено преку нивните стандардни девијации, меѓутоа како и останатите растојанија и тие подразбираат меѓусебна независност на компонентите (обележјата) во векторите. Во случај кога постои корелираност меѓу компонентите на векторите кои ги сочинуваат популациите, растојанието на дадена единка до дадена популација ќе зависи и од оваа корелираност (слика 2.10). Ниту Фрешеовото ниту стандардизираното евклидово растојание не ја моделира ваквата зависност.



Слика 2.10. Растојание на единка до две зависни нормално распределени популации со различни дисперзии во различни правци

За проблемот на комбинирање на класификатори според методот на шаблони за одлучување, векторите кои се споредуваат се всушност профилите, односно нивните обележја (координати) се излезите од класификаторите кои се комбинираат. Меѓутоа, излезите од разумно ефикасни класификатори се далеку од независни. Имено, кога на влезот на сите класификатори ќе се појави примерок од класата c_i , логично е да се очекува во најголемиот број на случаи излезите за дадената класа од секој од класификаторите да имаат високи вредности (како потврда дека

примерокот е од таа класа), додека останатите излези да имаат ниски вредности. Според ова, може да се очекува значителна зависност на излезите од класификаторите, односно позитивна корелираност на излезите за дадена класа од сите класификатори m_i^c , $i = 1, \dots, L$ за сите $c = 1, \dots, k$ и негативна корелираност на овие излези со останатите излези од сите класификатори m_i^c , $j \neq c$. Меѓутоа, самиот степен на корелираноста меѓу излезите од класификаторите нуди дополнителна информација која може корисно да се искористи при нивното комбинирање за донесување на конечната одлука. Методот кој овозможува користење и на овие информации е растојанието според Махаланобис.

2.19.3 Махаланобис растојание

Оваа мерка за растојание е воведена од Махаланобис (Mahalanobis) во 1936 година и се базира на зависностите меѓу променливите со кои различните примероци може да се идентификуваат и анализираат во однос на референтните точки. Растојанието на Махаланобис ја придружува и зависноста меѓу параметрите во анализата и го стандардизира секој параметар на нулта средна вредност и единична варијанса со употреба на коваријансната (дисперзиона) матрица во пресметувањето на растојанието. Изразот по кој се пресметува растојанието на Махаланобис е:

$$d_{mahalanobis}(\mathbf{x}, \mathbf{y}, \Sigma) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})}. \quad (2.122)$$

$\mathbf{x}$ и $\mathbf{y}$ се векторите меѓу кои се бара растојанието, а Σ^{-1} е инвертираната дисперзиона матрица:

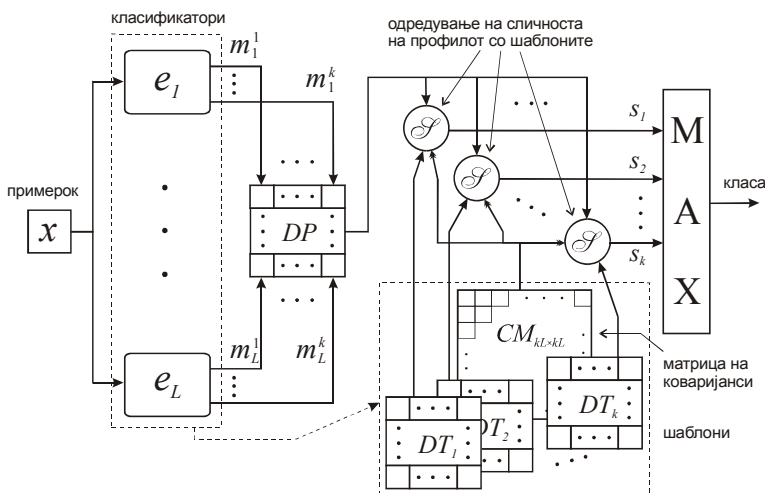
$$\Sigma = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{n,1} & \cdots & \cdots & \sigma_{n,n} \end{bmatrix}, \quad (2.123)$$

чији елементи се пресметуваат како:

$$\sigma_{i,j}(\mathbf{Z}) = \frac{1}{N} \sum_{l=1}^N (\mathbf{Z}_{l,i} - \bar{\mathbf{z}}_i)(\mathbf{Z}_{l,j} - \bar{\mathbf{z}}_j). \quad (2.124)$$

$\bar{\mathbf{z}}_i$ е средната вредност на i -тата компонента на векторите, односно средната вредност на излезот од класификаторот за сите примероци. Растојанието на Махаланобис со различна брзина се менува во различни правци во просторот зависно како од расејувањето по пооделните оски, така и од меѓузависноста на компонентите на векторите.

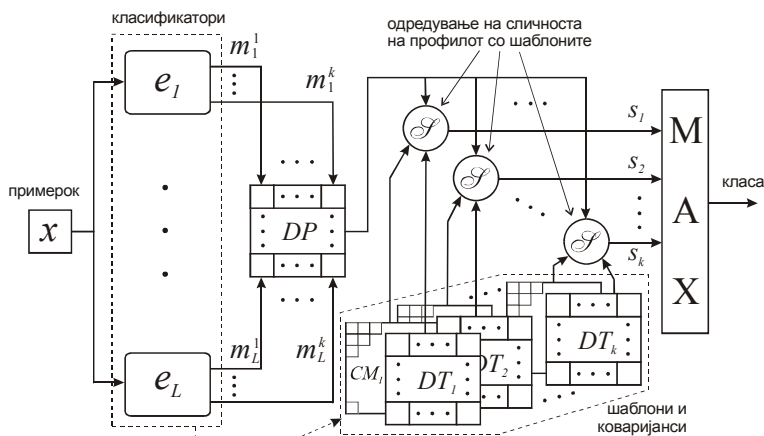
За примена на растојанието на Махаланобис при комбинирањето на класификатори според методот на шаблони за одлучување, векторите x и y од (2.122) се добиваат со подредување на редиците од матрицата на профилот односно шаблонот во еднодимензионален вектор кој ќе има $L \cdot k$ елементи. Соодветно, дисперзионата матрица Σ ќе има $Lk \times Lk$ елементи и ќе ги содржи коваријансите на секој од излезите од класификаторите со сите останати излези од сите класификатори. Шематскиот приказ на функционирањето на комбинирање на класификатори со шаблони за одлучување и растојание на Махаланобис како функција за одредување на сличноста на профилот со шаблоните е прикажан на слика 2.11. Притоа, дисперзионата матрица може да се пресмета според (2.124) од примероците од множеството за валидација земено како единствена популација во рамките на која се бараат меѓузависностите на излезите од класификаторите. Шаблоните за одлучување се пресметуваат исто како и за претходните шеми.



Слика 2.11. Комбинирање со шаблони за одлучување и растојание на Махаланобис како мерка на сличност со заедничка дисперзиона матрица

Пресметувањето на единствена дисперзиона матрица според (2.124) од сите примероци од множеството за валидација земено како единствена популација ги дава генералните меѓузависности на излезите за сите примероци. И без користење на ознаките на примероците од множеството за валидација може да се пресмета дисперзиона матрица која сè уште добро ги одразува меѓузависностите на излезите од класификаторите. Но, за пресметување на шаблоните (кои се различни за секоја класа) неопходно е да се познаваат класите на примероците кои ќе се користат за нивно одредување.

Бидејќи примероците во множеството за валидација се веќе обележани, тие природно можат да се поделат на онолку популации колку што класи има класификацискиот проблем. Користењето на вака изолираните популации за одредување на онолку различни коваријансни матрици колку што има класи, даваат уште подобра слика за меѓузависностите на излезите од класификаторите и тоа за секоја класа посебно. При ваквиот пристап на секој шаблон за одлучување се придружува посебна дисперзиона матрица специфична за дадената класа. Оваа дисперзиона матрица се добива само од примероците од дадената класа од валидациското множество. Користејќи го ваквиот пристап, шемата за комбинирање користејќи го растојанието на Махаланобис може да се модифицира во шема во која функциите за одредување на сличноста на профилот со секој од шаблоните ќе користат различни коваријансни матрици специфични за секоја класа како на слика 2.12.



Слика 2.12. Комбинирање со шаблони за одлучување и растојание на Махаланобис како мерка на сличност, со различни дисперзиони матрици специфични за секоја класа

2.19.4 Махаланобис растојание 2

Ова растојание се добива исто како и оригиналното Махаланобис растојание само што при неговата примена за комбинирање на класификатори за одредувањето на растојанието на профилот до секој од шаблоните се користи и соодветната дисперзиона матрица придружена на шаблонот (слика 2.12). За разлика од шаблоните кои се матрици со димензии $L \times k$, дисперзионата матрица има димензии $kL \times kL$, каде k е бројот на класи, а L е бројот на класификатори. При ваквиот пристап функцијата за одредување на растојанието меѓу профилот и шаблонот

за i -тата класа, земајќи ја предвид и дисперзионата матрица на излезите од класификаторите за класата i преминува во:

$$d_{mahalanobis2}^i(\mathbf{x}, \mathbf{t}_i, \Sigma_i) = \sqrt{(\mathbf{x} - \mathbf{t})^T \Sigma_i^{-1} (\mathbf{x} - \mathbf{t})}, \quad (2.125)$$

каде $\mathbf{x}$ е вектор добиен со нанижување на редиците од матрицата на профилот (2.84):

$$\mathbf{x} = [m_{1,1}(x) \ m_{1,2}(x) \dots m_{1,k}(x) \ m_{2,1}(x) \dots m_{L,1}(x) \dots m_{L,k}(x)]. \quad (2.126)$$

$\mathbf{t}$ е соодветно вектор добиен со нанижување на редиците од матрицата на шаблонот (2.85) за i -тата класа:

$$t_i = [d_{1,1}^i \ d_{1,2}^i \dots d_{1,k}^i \ d_{2,1}^i \dots d_{L,1}^i \dots d_{L,k}^i]. \quad (2.127)$$

Елементите од дисперзионата матрица $\Sigma_u = [\sigma_{q1,j1/q2,j2}]$ ја изразуваат коваријансата меѓу излезот за класата $j1$ од класификаторот $q1$ (m_{j1}^{q1}) и излезот за класата $j2$ од класификаторот $q2$ (m_{j2}^{q2}) и се добиваат од множеството за валидација како:

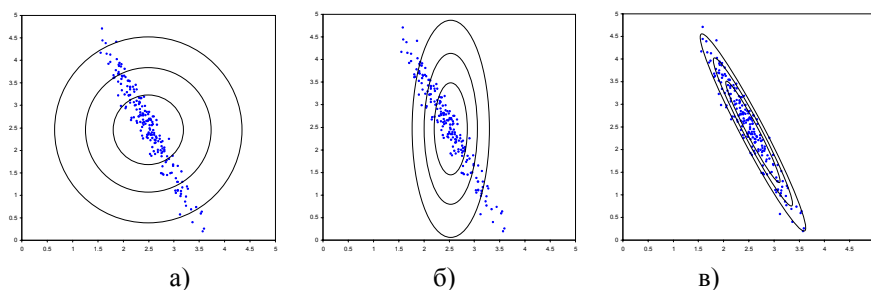
$$\sigma_{q1,j1/q2,j2}^i = \frac{1}{\sum_{s=1}^N Ind(z_s, i)} \sum_{s=1}^N Ind(z_s, i) (m_{j1}^{q1}(z_s) - {}^i\overline{m}_{j1}^{q1}) (m_{j2}^{q2}(z_s) - {}^i\overline{m}_{j2}^{q2}). \quad (2.128)$$

${}^i\overline{m}_j^q$ е средната вредност на излезот за класата q од класификаторот j за примероците од класата i :

$${}^i\overline{m}_j^q = \frac{\sum_{s=1}^N Ind(z_s, i) \cdot m_j^q(z_s)}{\sum_{s=1}^N Ind(z_s, i)}, \quad (2.129)$$

каде $Ind(z_s, i)$ е индикаторската функција дефинирана со (2.87), а N е бројот на примероци во множеството за валидација.

Растојанието на Махаланобис се сведува на стандардизираното евклидово растојание доколку сите променливи во векторите се независни, односно сите елементи на дисперзионата матрица освен дијагоналните се нула. Ако пак сите променливи имаат единечна варијанса, растојанието се сведува на обичното евклидово растојание. Сливовит приказ на еквидистантните криви за евклидовото, стандардизираното евклидово растојание и растојанието на Махаланобис за една зависна нормална дистрибуција се прикажани на слика 3.14.



Слика 2.13. Сликвит приказ на а) евклидовото растојание; б) стандардизираниот евклидово растојание; и в) растојанието на Махаланобис за една зависна нормална дистрибуција

Практичните проблеми кои се наметнуваат при реализацијата на комбинирањето на класификатори со примена на шаблоните за одлучување и растојанието на Махаланобис се однесуваат најмногу на бројот на расположливи примероци во множеството за валидација. За одредување на секој од шаблоните од кои секој има Lk елементи се употребуваат во просек N/k примероци од множеството за валидација. Одредување пак на дисперзионата матрица која има $(Lk)^2$ елементи треба да се изведе со истите N/k примероци за дадената класа, чиј број може да биде недоволен за прецизно одредување на дисперзионата матрица и добивање на нејзина стабилна инверзија. Ова е особено изразено ако се располага со ограничен број на примероци во множеството за валидација или некои од класите се значително помалку застапени. Во ваквите случаи е подобро да се примени шемата со единствена дисперзиона матрица која ќе може попрецизно да се одреди бидејќи за тоа ќе се користат сите примероци (од сите класи) од множеството за валидација.

2.20 Комбинирање со повеќекратна линеарна регресија (MLR)

Линеарната регресија е позната техника за откривање на линеарната зависност меѓу независните и зависната променлива [66]. Задачата на регресионата анализа е од одреден број на набљудувања (мерења) на независните и зависната променлива да се изведе општиот закон на зависност на зависната променлива од независните имајќи притоа предвид дека мерењата се со одредени отстапувања (грешки) чие влијание при градењето на моделот треба да се минимизира. Со вака изградениот модел потоа ќе може да се предвиди вредноста на зависната променлива на основа на измерените вредности на независните променливи.

Кога претпоставената зависност е линеарна, се работи за линеарна регресија:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \delta_i. \quad (2.130)$$

Ако зависната променлива зависи од повеќе независни променливи истовремено се работи за повеќекратна линеарна регресија:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \delta_i, \quad (2.131)$$

каде β_j се коефициентите на регресијата, δ_i се отстапувањата а p е димензионалноста на проблемот. Задачата на регресионата анализа е да се одредат коефициентите b_j на линеарната зависност:

$$\hat{Y}_i = b_0 + b_1 X_{i1} + b_2 X_{i2} + \dots + b_p X_{ip}, \quad (2.132)$$

така да проценетите вредности на зависната променлива $\hat{Y}_i$ отстапуваат што помалку од нивните вистински вредности Y_i .

Нека Y_i се зависните променливи, а $\hat{Y}_i$ нивните проценети вредности според регресиониот модел. Ако располагаме со n одмерози, истите би можеле да се напишат и како $n \times 1$ матрици $\mathbf{Y}$ и $\hat{\mathbf{Y}}$ соодветно. Нека претпоставиме постоење на една фиктивна независна променлива X_{0i} секогаш еднаква на 1, за да се моделира слободниот коефициент b_0 во (2.132), која ќе претставува еден вид хомогената координата. Тогаш множеството на сите независни променливи X_{ij} , $i = 1, \dots, n, j = 0, 1, \dots, p$, каде p е димензионалноста на проблемот, може да се претстави како матрица $\mathbf{X}$ со димензии $n \times (p+1)$. Исто така, коефициентите β_j и проценетите регресиони коефициенти b_j можат да се напишат во матрична форма како $(p+1) \times 1$ матрици $\boldsymbol{\beta}$ и $\mathbf{b}$, а отстапувањата како $n \times 1$ матрица $\boldsymbol{\delta}$. Тогаш (2.131) и (2.132) можат да се напишат во матрична форма како:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\delta} \quad (2.133)$$

и

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{b} \quad (2.134)$$

соодветно.

Доколку треба да моделираме повеќе зависни променливи на основа на исто множество независни променливи станува збор за повеќекратна линеарна регресија или линеарна регресија со повеќе одзиви (multi response linear regression – MLR). Во ваков случај за секоја од k -те зависни променливи би имале по еден израз (2.131) и еден (2.132). Доколку сите зависни променливи ги организираме во единствана $n \times k$

матрица $\mathbf{Y}$, изразите во матрична форма (2.133) и (2.134) и понатаму ќе важат, само сега $\mathbf{b}$ ќе биде матрица со димензии $(p+1) \times k$.

Коефициентите на линеарната регресија се одредуваат по методот на најмали квадрати [66] од изразите (2.131) и (2.132) односно решението на матрицата $\mathbf{b}$ напишано во матрична форма е:

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (2.135)$$

Со вака добиените коефициенти $\mathbf{b}$, изразот (2.134) може да се употреби за проценување на вредностите на зависните променливи за произволни вредности на независните променливи сместени во матрица $\mathbf{X}$ со димензии $1 \times (p+1)$.

Во проблемот на комбиниране на класификатори, ако излезите од класификаторите се третираат како независни случајни променливи, а посакуваните излези од комбинаторот (поддршките за секоја од класите) како зависни променливи, комбинирањето се сведува на проблем на регресија. Под претпоставка на нивна линеарна зависност, комбинирањето ќе претставува примена на моделот на повеќекратна линеарна регресија. Ваквиот регресионен модел ќе има онолку влезови колкав што е бројот на излезите од сите класификатори заедно, и онолку излези колку што има класи проблемот. За одредување на коефициентите на линеарната регресија како независни променливи се употребуваат вредностите на излезите од класификаторите за примероците од множеството за валидација, а за вредности на зависните променливи се земаат посакуваните вредности на излезите од комбинаторот, односно влезовите во максимум селекторот. Вака изградениот модел потоа се користи за од излезите на класификаторите за непознат примерок да се добијат соодветни влезови во максимум селекторот.

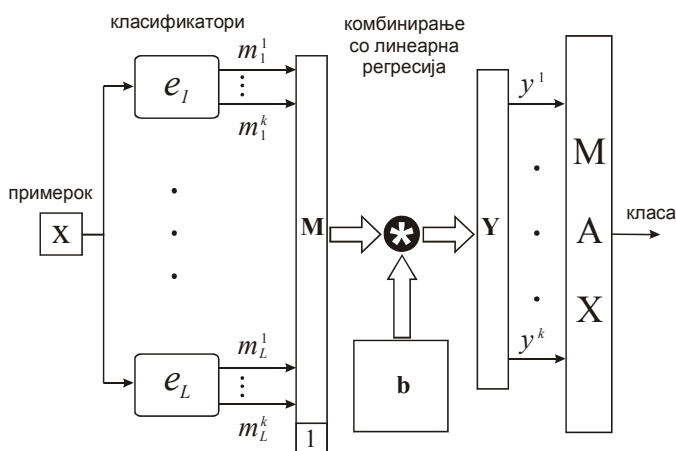
Шемата за комбиниране на класификатори е прикажана на слика 2.14. Правилото за комбиниране во овој случај е:

$$Q(x) = \arg \max_{i=1, \dots, k} (\mathbf{M}(x) \cdot \mathbf{b})_i, \quad (2.136)$$

каде $\mathbf{M}(x)$ е матрицата со $1 \times (Lk+1)$ елементи, добиена од сите излези од класификаторите за дадениот примерок проширен со хомогената координата, а $\mathbf{b}$ е матрицата со регресионите коефициенти (со димензии $(Lk+1) \times k$).

Методот за комбиниране со повеќекратна линеарна регресија е донекаде сличен на тежинското комбиниране, бидејќи кај обете методи излезите од комбинирањето претставуваат тежинска комбинација од излезите од индивидуалните класификатори. Разликата е во тоа кои од излезите од индивидуалните класификатори учествуваат во комбинирањето за добивање на поддршката за одредена класа. Кај тежинското

комбинирање i -тиот влез во максимум селекторот претставува линеарна комбинација само од i -тите излези од секој од индивидуални класификатори, односно поддршката за дадена класа се добива со комбинирање само на поддршките кои за таа класа ја дал секој од индивидуалните класификатори. Кај линеарната регресија пак, поддршката за дадена класа се добива како линеарна комбинација од излезите за **сите** класи од сите класификатори. Со ова практично при одредувањето на поддршката која ќе биде дадена за i -тата класа учествуваат и поддршките кои индивидуалните класификатори ги дале и за другите, а не само за i -тата класа.



Слика 2.14. Комбинирање со повеќекратна линеарна регресија

При практичната реализација, линеарната регресија е релативно едноставана за имплементација бидејќи се изведува со едноставни матрични операции и при одредувањето на коефициентите (обуката) и при комбинирањето. Сепак, со порастот на бројот на класификатори и бројот на класи, димензиите на матриците брзо растат, а со тоа и нејзината комплексност. Ефикасноста на линеарниот регресионен модел за комбинирање на класификатори директно зависи од тоа колку зависноста помеѓу излезите од класификаторите и влезовите во максимум селекторот, на основа на кои се донесува одлуката, е навистина линеарна.

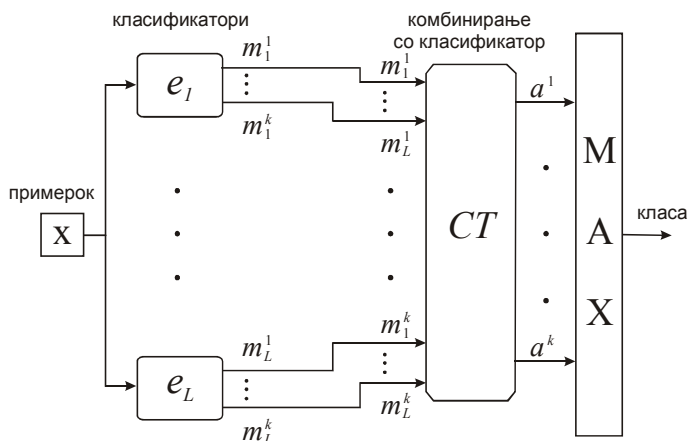
2.21 Комбинирање како класификациска задача

Целта на секое комбинирање на класификатори е што поефикасно да ги искористи одлуките на индивидуалните класификатори при донесувањето на конечната одлука, што најефикасно може да се постигне доколку паралелно се набљудуваат сите индивидуални класификатори со цел да се разбере нивното глобално однесување. За да се постигне

ова потребно е учење и адаптација, кои се едни од основните карактеристики на класификаторите.

Ова води кон концептот дека комбинирањето на одлуките на повеќе класификатори кои работат паралелно, било истите да се или не се независни е повторно класификациски проблем. Следејќи ја оваа идеја шемата за комбинирање на класификатори ја добива формата прикажана на слика 2.15.

Примерокот се проследува паралелно до сите индивидуални класификатори како и кај другите шеми за комбинирање. Кај овој пристап излезите од класификаторите претставуваат влезови во класификатор кој го „изведува“ комбинирањето на овие излези и на излезот дава соодветна поддршка за секоја од класите. На крајот максимум селекторот ја одредува класата на примерокот.



Слика 2.15. Комбинирање на класификатори како класификациска задача

Класификаторот за комбинирање има онолку влезови колку што вкупно имаат излези сите класификатори заедно – $k \cdot L$, и k излези, каде k е бројот на класи во проблемот, а L е бројот на класификатори кои се комбинираат. Класификаторот за комбинирање (СТ) се обучува со излезите кои индивидуалните класификатори ги даваат за примероците од множеството за валидација кое всушност претставува множество за обука за овој класификатор. Со ова, правилото за комбинирање на класификатори со обучен класификатор станува:

$$Q(x) = \arg \max_{i=1, \dots, k} \mathcal{F}_i(m_1^1, \dots, m_1^k, m_2^1, \dots, m_L^k), \quad (2.137)$$

каде со $\mathcal{F}_i$ е означен i -тиот излез од обучениот класификатор кој го изведува комбинирањето. При ова класификаторот СТ се нарекува и кла-

сификатор на мета-ниво бидејќи влезови за него претставуваат излезите од класификаторите од првото или „влезно“ ниво.

Пристапи за комбинирањето на класификатори со помош на друг класификатор многу слични или идентични на изложениот во литературата се сретнуваат под различни имиња како: комбинирање на класификатори како класификациска задача [130], фузија на одлуки (decision fusion) [154], агрегација на одлуки [152], класификација на мета-ниво (meta level classification) [141], конкатенација на класификатори [130], напластена генерализација (stacked generalization) [161][167], адаптивна агрегација на класификатори [153] и сл.

При изборот на видот на класификатор кој ќе се употреби за комбинирање треба да се има предвид дека распределбата на вредностите кои ќе бидат проследени на неговите влезови ќе биде далеку од нормалната или униформната распределба, па за оваа задача треба да се избегнуваат класификатори кои подразбираат некоја од овие распределби. Со доволно моќен нелинеарен класификатор за комбинирање и доволно примероци за негова обука, ваквото комбинирање е супериорно во однос на сите други шеми за комбинирање и далеку ги надминува во перформансите. Исто така, со доволно примероци за обука на класификаторот за комбинирање, резултатите на ваквото комбинирање најмалку зависат од квалитетот на индивидуалните класификатори или нивната корелираност.

Од аспект на класификаторот кој го изведува комбинирањето, класификаторите од првото ниво (класификаторите кои се комбинираат) делуваат и како екстрактори на обележја, кои ги користи овој класификатор за донесување на конечната одлука. Класификатори како невронските мрежи или машините со носечки вектори се одлично квалификувани за ваквата задача [76], пред сè поради фактите како:

- во состојба се да ги согледаат невидливите, скриени релации од податоците;
- знаат да генерализираат, односно да одговараат точно на примероци кои се само слични на примероците со кои се обучувани;
- нелинеарни се, што значи дека можат поуспешно да решат некои проблеми од линеарните техники.

Сепак, обуката на класификаторот за комбинирање редовно е далеку покомплексна и временски подолготрајна задача од обучувањето на некоја од другите методи за обучено комбинирање. Исто така, зависно од видот на избраниот класификатор и препознавањето, односно комбинирањето на излезите од индивидуалните класификатори може да биде побавно од останатите шеми.

2.22 Комбинирање на еден-против-еден класификатори

Во поглавјето 1.6.3 беше изложен концептот како со бинарни класификатори кои се обучени со еден-против-еден стратегија може да се реши повеќекласен проблем. Комбинирањето на повеќе еден-против-еден класификатори претставува нужно комбинирање кое е неопходно за да се реши повеќекласниот проблем. Во ова поглавје ќе биде разгледан проблемот на комбинирање на неколку потполни множества од еден-против-еден класификатори, секое од кои самостојно го решава повеќекласниот проблем, со цел постигнување на подобри перформанси. Во овој случај располагаме со неколку еквивалентни условно независни еден-против-еден бинарни класификатори обучени за истиот пар класи, но под различни услови (користејќи различни обележја или множества за обука).

Како што беше изложено, за решавање на повеќекласен проблем потребни се $k(k-1)/2$ еден-против-еден обучени класификатори при што нивните излези се комбинираат за да се донесе одлуката за припадност на примерокот кон некоја од класите. Ако имаме L групи од $k(k-1)/2$ еден-против-еден класификатори кои се условно независни, нивните одлуки може да се комбинираат со цел да се подобри точноста на класификувањето. Притоа ова „незadolжително“ комбинирање може да се изведе на три основни начини:

Прво од секоја група од $k(k-1)/2$ еден-против-еден класификатори се изведува здружената поддршка за секоја од k -те класи според некоја од методите Max Wins или pairwise coupling, а потоа вака добиените L поддршки се комбинираат според некоја од опишаните методи за комбинирање на еден-против-сите класификатори;

Поединечно комбинирање – прво се комбинираат L -те соодветни еден-против-еден класификатори од сите групи со што се добиваат $k(k-1)/2$ комбинирани излези, а потоа од нив се изведува здружената поддршка за секоја од k -те класи и конечно се избира класата со максимална поддршка;

Интегрално комбинирање – истовремено комбинирање на сите $Lk(k-1)/2$ еден-против-еден класификатори од сите групи за добивање на поддршките за секоја од k -те класи.

Првиот начин практично се сведува на обединување на излезите од сите еден-против-еден класификатори од секоја од групите поодделно со што од секоја група се добиваат k излези како поддршка за секоја од класите. На овој начин секоја од групите прво се сведува на класификатор со k излези а потоа овие излези се комбинираат со некоја од

фиксните или обучени методи како комбинирање на класификатори со повеќе излези.

Кај поединечниот начин се комбинираат еден-против-еден класификаторите според некоја од шемите за комбинирање на класификатори а потоа од комбинираните излези се изведува поддршката за секоја од класите. Комбинирањето на еден-против-еден класификаторите со фиксните методи за комбинирање има директна теориска интерпретација како еден вид усреднување на мислењата на повеќе „парцијални“ експерти пред обединувањето на овие парцијални мислења во поддршка за класите. Комбинирањето со обучени методи подразбира обука на методот за комбинирање во кој процес е потребно да се достават и целните вредности на излезите. Бидејќи излезите од еден-против-еден класификаторите немаат директна интерпретација за сите примероци туку само за оние две класи за кои се обучени, директна примена на обучените методи за комбинирање во овој случај не е можна.

Интегралниот начин на комбинирање може да се изведе само со некоја од методите кај која излезите кои се комбинираат не зависат од класите, односно се набљудуваат како информација на мета ниво, како шаблоните за одлучување, повеќекратната регресија, комбинирањето со помош на класификатор. Ваквите начини на комбинирање вообичаено имаат и најдобри перформанси бидејќи одеднаш обединуваат најголем број на информации од индивидуалните еден-против-еден класификатори и во состојба се да ги искористат зависностите меѓу сите нив. Сепак сите вакви методи се обучени методи за чија обука се потребни значителен број на дополнителни обележани примероци особено поради фактот што димензионалноста во овој случај е многу поголема. Така на пример, при комбинирањето со проширени шаблони за одлучување, дисперзионите матрици ќе бидат со димензии $Lk(k-1)/2 \times Lk(k-1)/2$ за чија проценка ќе бидат потребни доста поголем број на примероци отколку кога се комбинираат еден-против-сите класификатори.

Кај поединечното комбинирање, при одредувањето на поддршката за одредена класа предвид се земаат само излезите од класификаторите кои биле обучувани на примероци од таа класа, а не и од останатите двокласни класификатори. За разлика од ова, кај интегралното комбинирање во одредувањето на поддршката за одредена класа учествуваат сите класификатори, вклучувајќи ги и оние кои не се обучувани за препознавање на таа класа. Ова е од особена важност, бидејќи некои еден-против-еден класификатори се значително корелирани, односно може корисно да се искористат како дополнителна информација при одредувањето на поддршката за дадена класа иако тие воопшто не биле обучувани за препознавање на таа класа.

III

Наши искуства и заклучоци

Мислењата (или одлуките) од повеќе независни експерти (класификатори) кои истиот проблем го гледаат од различни агли можат ефикасно да се искомбинираат со цел да се добие пореална претстава за проблемот и да се донесе поквалитетна одлука. Комбинирањето на вака добиените класификатори претставува *фузија на одлуки*. Од друга страна, различните групи обележја можат да се соберат во единствено множество од обележја за кое може да се конструира единствен класификатор кој проблемот го гледал од сите аспекти истовремено. Ова е природот на т.н. *фузија на податоци*.

Се наметнува прашањето кој од овие два приода е подобар. Одговорот на ова прашање не е едноставно да се даде, а пред сè зависи од класификаторите и од начинот на нивно комбинирање. Како дополнителен фактор се наметнува и бројот и квалитетот на означените примероци кои се неопходни за обучување на класификаторите и некои од методите за комбинирање.

3.1 Комбинирање на класификатори со различни методи

Испитувани се комбинирања на класификатори за препознавање со рака испишани цифри со 77 различни методи и нивни варијации [56][24][62][63]. Комбинирани се различен број класификатори. Сите комбинирања се изведени за сите верзии на класификаторите обучени на 1000, 2000, 5000, 10000, 30000 и сите 53449 примероци. Истите експерименти се спроведени за SVM класификатори со гаусов кернел во

еден-против-сите шема, SVM класификатори со линеарен кернел во еден-против-сите шема, невронски мрежи и SVM класификатори со гаусов кернел во еден-против-еден шема.

За испитуваните методи за комбинирање на класификатори користени се следниве кратенки:

vote	Плурално мнозинско гласање (објаснето во поглавјето 2.7). За конечна класа се избира онаа која ќе добие најголем број на гласови. Отфрлувањето на примероците за кои нема мнозинство не е разгледувано како опција и во ваквите случаи случајно се одбира една од класите кои добиле ист број на гласови. Гласањето како метод за комбинирање на класификатори има оправдување само кога се комбинираат три или повеќе класификатори, но поради комплетност дадено е и во табелите за два класификатора.
borda	Борда пребројување. Излезот од секој класификатор е ранг листа на класи која се добива подредувајќи ги класите според соодветниот излез од класификаторот во опаѓачки редослед. Конечната класа се одредува според методот за пребројување според Борда (објаснето во поглавјето 2.10.2).
bks	BKS комбинирање (објаснето во поглавјето 2.9). На основа на BKS табелата со снименото поведение на класификаторите се одредува најверојатната класа за дадената комбинација на одлуки на класификаторите. И при одредувањето на поведението на класификаторите и при комбинирањето се користат конечните одлуки за припадност на примерокот кон одредена класа на секој од класификаторите. Во случаите каде комбинацијата на одлуки на класификаторите е таква што не може да се одреди конечната класа на основа на снименото поведение (BKS табелата за таа комбинација е празна) случајно се избира една од класите.
bksv	Модифицирано BKS комбинирање со дополнително гласање (објаснето во поглавјето 2.9.1). Слично е на BKS комбинирањето со таа разлика што за случаите каде што не може да се одреди конечната класа на основа на BKS табелата, примерокот се распоредува во онаа класа која добила најмногу гласови. Доколку нема доминантна класа и ни според гласањето не може да се донесе квалификувана одлука, примерокот случајно се распоредува во некоја од класите кои освоиле ист број гласови.
avg	Средна вредност (аритметичка средина) или правило на сума (објаснето во поглавјето 2.11.2).
prod	Геометриска средина (правило на производ) (објаснето во поглавјето 2.11.1).
harm	Хармониска средина (објаснето во поглавјето 2.11.3).
cprod	Комплементарен производ (објаснето во поглавјето 2.11.4).
maxmax	Максимум од максимумите (објаснето во поглавјето 2.12.1).
minmax	Минимум од максимумите (објаснето во поглавјето 2.12.2).
med	Комбинирање со избор на медијана (објаснето во поглавјето 2.12.3).

davg	Тежинско комбинирање со динамичко одредување на тежините (објаснето во поглавјето 2.17).
pprod	Веројатносен производ (објаснето во поглавјето 2.14).
bayes	Наивно баесово комбинирање (објаснето во поглавјето 2.8).
fi	Комбинирање со неспецифициран интеграл (објаснето во поглавјето 2.15). За неспецифицираните густини е земена половина од проценетата точност на класификаторот на множеството за валидација.
fic	Модификација на комбинирањето со неспецифициран интеграл каде неспецифицираните густини се одредуваат одвоено за секоја класа и секој класификатор (објаснето во поглавјето 2.15).

Демпстер-Шефер метод за комбинирање на сведоштва (објаснето во поглавјето 2.13) во 27 варијации. Варијациите се состојат во различните мерки за сличност помеѓу референтниот излезен вектор и актуелните излези од класификаторите од кои се трансформираат во сведоштва за да се примени D-S правилото за комбинирање. Истите мерки за сличност се употребуваат и кај шаблоните за одлучување кои се објаснети во поглавјето 2.18.

demp	Демпстер-Шефер метод со евклидово растојание според (2.104).
dempp1	Демпстер-Шефер метод со неспецифицирана мерка за сличност S_1 според (2.89).
dempp2	Демпстер-Шефер метод со неспецифицирана мерка за сличност S_2 според (2.91).
dempp3	Демпстер-Шефер метод со неспецифицирана мерка за сличност S_3 според (2.93).
dempp4	Демпстер-Шефер метод со неспецифицирана мерка за сличност S_4 според (2.95).
dempi1	Демпстер-Шефер метод со индекс на вклучување I_1 според (2.96).
dempi2	Демпстер-Шефер метод со индекс на вклучување I_2 според (2.97).
dempi3	Демпстер-Шефер метод со индекс на вклучување I_3 според (2.99).
dempi4	Демпстер-Шефер метод со индекс на вклучување I_4 според (2.100).
dempi5	Демпстер-Шефер метод со индекс на вклучување I_5 според (2.101).
dempc	Демпстер-Шефер метод со индекс на конзистентност од (2.102).
dempmnk	Демпстер-Шефер метод со Минковски растојание според (2.109). Резултатите се за степен $q=1$, односно за Менхетен растојание.
dempch	Демпстер-Шефер метод со Чебишево растојание според (2.108).
dempcan	Демпстер-Шефер метод со Канбера растојание според (2.106).
dempas	Демпстер-Шефер метод со аглово одвојување според (2.107).
dempchi	Демпстер-Шефер метод со χ^2 растојание според (2.110).
dempchi2	Демпстер-Шефер метод со квадратно χ^2 растојание според (2.111).
dempbc	Демпстер-Шефер метод со Брај-Куртис растојание според (2.115).
dempchl	Демпстер-Шефер метод со Хелинцер растојание според (2.114).
dempchr	Демпстер-Шефер метод со Корд растојание според (2.112).

dempchr2	Демпстер-Шефер метод со квадратно Корд растојание според (2.113).
dempjac	Демпстер-Шефер метод со Jaccard II растојание според (2.116).
demppeer	Демпстер-Шефер метод со процент на сличност според (2.117).
dempse	Демпстер-Шефер метод со стандардизирано евклидово растојание според (2.118).
dempfr	Демпстер-Шефер метод со Фрешеовото растојание според (2.121).
dempm	Демпстер-Шефер метод со Махаланобис растојание според (2.122).
dempmc	Демпстер-Шефер метод со Махаланобис растојание и одвоени дисперзионии матрици за секоја класа според (2.125).

Комбинирање со шаблони за одлучување (објаснето во поглавјето 2.18) во 27 варијации зависно од употребената мерка за сличност меѓу профилот и шаблоните.

dtp1	Комбинирање со шаблони за одлучување и неспецифицирана мерка за сличност S_1 според (2.89).
dtp2	Комбинирање со шаблони за одлучување и неспецифицирана мерка за сличност S_2 според (2.91).
dtp3	Комбинирање со шаблони за одлучување и неспецифицирана мерка за сличност S_3 според (2.93).
dtp4	Комбинирање со шаблони за одлучување и неспецифицирана мерка за сличност S_4 според (2.95).
dti1	Комбинирање со шаблони за одлучување и индекс на вклучување I_1 според (2.96).
dti2	Комбинирање со шаблони за одлучување и индекс на вклучување I_2 според (2.97).
dti3	Комбинирање со шаблони за одлучување и индекс на вклучување I_3 според (2.99).
dti4	Комбинирање со шаблони за одлучување и индекс на вклучување I_4 според (2.100).
dti5	Комбинирање со шаблони за одлучување и индекс на вклучување I_5 според (2.101).
dtc	Комбинирање со шаблони за одлучување и индекс на конзистентност според (2.102).
dte	Комбинирање со шаблони за одлучување и евклидово растојание според (2.104).
dtmnk	Комбинирање со шаблони за одлучување и Минковски растојание според (2.109). Резултатите се дадени за степен $q=1$, односно за Менхетен растојание.
dtch	Комбинирање со шаблони за одлучување и Чебишев растојание според (2.108).
dtcan	Комбинирање со шаблони за одлучување и Канбера растојание според (2.106).

dtas	Комбинирање со шаблони за одлучување и аглово одвојување според (2.107).
dtchi	Комбинирање со шаблони за одлучување и χ^2 растојание според (2.110).
dtchi2	Комбинирање со шаблони за одлучување и квадратно χ^2 растојание според (2.111).
dtbc	Комбинирање со шаблони за одлучување и Брај-Куртис растојание според (2.115).
dthl	Комбинирање со шаблони за одлучување и Хелинцер растојание според (2.114).
dtchr	Комбинирање со шаблони за одлучување и Корд растојание според (2.112).
dtchr2	Комбинирање со шаблони за одлучување и квадратно Корд растојание според (2.113).
dtjac	Комбинирање со шаблони за одлучување и Jaccard II растојание според (2.116).
dtper	Комбинирање со шаблони за одлучување и процент на сличност според (2.117).
dtse	Комбинирање со проширени шаблони за одлучување и стандардизирани евклидово растојание според (2.118).
dtfr	Комбинирање со проширени шаблони за одлучување и Фрешеовото растојание според (2.121).
dtm	Комбинирање со проширени шаблони за одлучување и Махаланобис растојание според (2.122).
dtmc	Комбинирање со проширени шаблони за одлучување и Махаланобис растојание и одвоени дисперзионии матрици за секоја класа според (2.125).
epw	Тежинско комбинирање според проценетите перформанси на индивидуалните класификатори (објаснето во поглавјето 2.16.4).
gc	Тежинско комбинирање со обопштен комитет (објаснето во поглавјето 2.16.1).
mgc	Тежинско комбинирање со модифициран обопштен комитет (објаснето во поглавјето 2.16.2).
ogc	Тежинско комбинирање со обопштен комитет со тежини одредени според Хук-Џивс методот за оптимизација (објаснето во поглавјето 2.16.3).
omgc	Тежинско комбинирање со модифициран обопштен комитет со тежини одредени според Хук-Џивс методот за оптимизација (објаснето во поглавјето 2.16.3).
mlr	Комбинирање со повеќекратна линеарна регресија (објаснето во поглавјето 2.20).
svmcm	Комбинирање како класификациска задача со употреба на SVM класификатор со гаусов кернел за комбинирање на излезите од индивидуалните класификатори. Примероците кои се користат за обука на комбинаторот се различни од примероците кои биле користени за обука на индивидуалните класификатори.

- mlpcmb** Комбинирање како класификациска задача со употреба на невронска мрежа за комбинирање на излезите од индивидуалните класификатори. Невронската мрежа која го изведува комбинирањето е со онолку влезови колку што имаат вкупно излези сите класификатори кои се комбинираат (10 пати по бројот на класификатори), 30 неврони во скриеното ниво и 10 излези.
- joinedin** Ова не е метод за комбинирање на класификатори, туку перформансите на единствен класификатор кој како обележја ги користи сите обележја кои ги користат индивидуалните класификатори како единствено множество обележја, односно реализира фузија на податоци. Даден е за споредба со останатите методи за фузија на одлуки.
- oracle** Хипотетички метод за комбинирање наречен „пророк“ (oracle) која знае секогаш да ја избере вистинската класа на примерокот доколку барем еден од индивидуалните класификатори се одлучил за таа класа. Даден е само за споредба.
- cdc** Коефициент на зависност на класификаторите (classifier dependancy coefficient) [95][134]. Ова не е метод за комбинирање на класификатори, туку мерка за изразување на зависноста меѓу класификаторите. Даден е за споредба бидејќи од него најмногу зависат перформансите на комбинирањето.

Во следната табела (**Табела 3.1**) се дадени **типичните** резултати на комбинирање на класификатори за 4 множества обележја за препознавање со рака испишани цифри. Во првите четири редови (после насловниот) се дадени перформансите на индивидуалните класификатори кои учествуваат во комбинирањето (FS1, FS2, FS3, FS4). Во следните редови се дадени перформансите на комбинацијата од класификатори, комбинирани според секоја од различните методи за комбинирање. Перформансите се дадени како рата на препознавање односно постигнатата точност на множеството за тестирање изразено во проценти, без опција за отфрлување. Во претпоследниот ред за споредба се дадени перформансите на единствен класификатор кој ги користи сите обележја од индивидуалните класификатори како единствено множество (joinedin). Во последниот ред се дадени перформансите на хипотетичкото „пророк“ комбинирање (oracle).

Во следните колони се дадени перформансите на комбинирањето на класификаторите обучени користејќи 1000, 2000, 5000, 10000, 30000 и сите примероци за обука и рангот (R) на методот. Рангот го означува редниот број во подредената листа на методи според постигнатите резултати за дадената колона (1 го означува методот со најдобри перформанси). Во табелата во која се прикажани резултатите при комбинирањето на SVM класификаторите со гаусов кернел дадена е и колона за класификаторите кои се обучени без автоматско одредување на параметрите на SVM машините. Кај методите за комбинирање кои ги комби-

нираат цврстите одлуки на индивидуалните класификатори (излези од тип 1), за одлука на секој класификатор одредена е класата за која излезот на класификаторот е максимален. Кај методите за кои е потребен веројатносен излез од класификаторите, излезите од SVM класификаторите во веројатности се трансформираат според изразот (1.14).

Табела 3.1: Резултати на комбинирањето (во %) на SVM класификатори за четири множества обележја во апликацијата за препознавање цифри испишани со рака; R го дава рангот на методот.

#	метод	1000	R	2000	R	5000	R	10000	R	30000	R	сите	R
	FS1	84.7969		86.1576		88.5107		89.7998		91.9824		92.7719	
	FS2	87.3853		89.9857		91.5476		93.1931		94.8829		95.3398	
	FS3	86.5123		89.6003		90.3557		92.4087		94.7840		95.1454	
	FS4	89.9226		91.5203		92.4240		94.1002		95.7133		96.1003	
1	vote	90.2704	67	92.2552	68	93.0055	71	94.3730	68	96.0509	67	96.3220	64
2	borda	90.8894	54	92.7310	54	93.5597	60	94.8470	58	96.2811	49	96.6221	40
3	bks	91.3208	36	92.5417	61	93.2203	68	94.0371	72	95.3961	75	95.7150	75
4	bksv	92.1444	7	93.5170	12	94.1803	6	95.0142	31	96.2691	52	96.5181	53
5	avg	91.3106	38	93.0123	34	93.7302	36	95.0278	26	96.3800	34	96.6630	31
6	prod	90.9354	51	92.6116	59	93.7677	25	95.0892	21	96.5420	22	96.7091	26
7	harm	90.8041	56	92.3609	67	93.5648	58	94.9903	43	96.4397	31	96.6085	45
8	cprod	90.0249	69	92.3797	65	93.6449	54	94.8931	56	96.4465	30	96.6869	28
9	maxmax	89.2627	76	91.5783	75	93.1129	70	94.2485	71	96.0799	65	96.2657	65
10	minmax	90.4086	63	91.8085	70	93.2442	67	94.6561	67	96.2095	63	96.3561	63
11	med	91.0173	48	92.8708	46	93.6108	56	95.0244	27	96.4107	32	96.6698	29
12	davg	91.3327	35	92.9322	41	93.7268	37	94.9954	42	96.3101	41	96.6204	41
13	demp	91.3447	34	92.8469	47	93.6569	52	94.8897	57	96.2436	58	96.6102	44
14	dempp1	91.1349	45	93.0328	33	93.7711	24	95.1148	17	96.5573	15	96.7329	22
15	dempp2	90.9985	50	92.9680	39	93.7592	29	95.1011	20	96.5573	15	96.7381	20
16	dempp3	90.3148	65	92.6559	55	93.7234	41	95.0483	24	96.5727	13	96.7415	18
17	dempp4	90.9167	53	92.9356	40	93.9109	17	95.1318	15	96.6187	8	96.8097	10
18	dempi1	91.3089	39	93.1931	19	93.8547	20	95.2409	9	96.5829	11	96.7687	14
19	dempi2	90.7308	59	92.7497	52	93.6381	55	94.9596	50	96.4840	29	96.6375	34
20	dempi3	88.0742	77	91.8034	71	93.4147	63	94.8436	59	96.5232	26	96.6511	33
21	dempi4	90.2994	66	92.3746	66	93.5835	57	94.8334	60	96.3578	36	96.6136	43
22	dempi5	89.7930	72	92.1376	69	93.4181	62	94.7396	64	96.3442	37	96.5931	48
23	dempc	90.3455	64	92.7514	51	93.7592	29	95.1199	16	96.5710	14	96.7312	23
24	dempmnk	91.0173	48	92.9288	43	93.7472	33	95.0568	23	96.5062	27	96.7227	24
25	dempch	90.7752	57	92.7770	50	93.7626	28	95.0397	25	96.5471	20	96.7585	15

Табела 3.1 (продолжува):

#	метод	1000	R	2000	R	5000	R	10000	R	30000	R	сите	R
26	dempcan	91.7062	23	92.8060	48	93.0004	72	94.2844	69	94.8846	76	94.7720	77
27	dempas	92.1222	10	93.5563	10	93.9365	14	95.2222	10	96.5403	23	96.9427	5
28	dempchi	91.2219	43	93.0976	25	93.8751	19	95.1847	13	96.5368	24	96.7466	16
29	dempchi2	91.2407	42	93.1351	23	93.9382	13	95.1983	12	96.5965	9	96.7858	13
30	dempbc	91.0736	47	93.0055	35	93.7558	31	95.1079	18	96.5488	18	96.7210	25
31	dempchl	91.4487	32	93.2476	15	93.9587	10	95.2631	7	96.5761	12	96.7892	12
32	dempchr	92.1734	6	93.6841	7	93.9706	9	95.2205	11	96.5300	25	96.9273	7
33	dempchr2	91.4692	31	93.2289	18	93.9894	7	95.2529	8	96.5948	10	96.7960	11
34	dempjac	90.9337	52	92.8998	45	93.7199	43	95.0602	22	96.4874	28	96.7074	27
35	demppe	91.1622	44	93.0549	30	93.7847	23	95.1642	14	96.5539	17	96.7398	19
36	dempse	90.7445	58	93.5545	11	93.9024	18	94.9800	45	96.2947	48	96.5539	49
37	dempfr	92.1359	9	93.6910	6	93.9365	14	94.9477	52	96.2436	58	96.4874	55
38	dempm	91.2441	41	92.6440	57	93.4778	61	94.6834	66	96.3834	33	96.6187	42
39	dempmc	92.3746	5	93.9194	5	93.9894	7	94.9425	53	96.0645	66	96.2419	66
40	pprod	91.0753	46	92.7395	53	93.8035	21	95.1045	19	96.5437	21	96.7346	21
41	bayes	91.7795	19	93.2374	17	93.9126	16	95.0142	31	96.3220	38	96.6630	31
42	fi	90.6763	60	92.6201	58	93.4011	64	94.7840	63	96.2708	50	96.5454	50
43	fic	90.5722	62	92.4684	64	93.2868	66	94.7379	65	96.2981	47	96.5300	51
44	dtp1	91.8102	15	93.0601	26	93.6841	47	95.0005	37	96.2674	53	96.4755	56
45	dtp2	91.8630	13	93.1811	20	93.7438	34	94.9664	47	96.3067	45	96.6051	46
46	dtp3	91.6550	25	93.2613	14	93.8035	21	95.0176	30	96.3220	38	96.6340	35
47	dtp4	89.5679	74	91.1332	76	92.6576	76	93.8973	76	95.5018	73	95.8514	73
48	dti1	91.9671	11	93.3414	13	93.7523	32	95.0227	28	96.3101	41	96.6340	35
49	dti2	91.3157	37	92.4888	63	93.3465	65	94.9135	54	96.0816	64	96.2026	67
50	dti3	90.6763	60	92.6508	56	93.6569	52	94.9545	51	96.3663	35	96.6238	39
51	dti4	89.6088	73	91.7931	72	93.1845	69	94.2502	70	95.9724	69	96.1702	68
52	dti5	89.8612	71	91.6891	73	92.8503	73	93.9962	74	95.7542	71	96.1106	71
53	dte	89.9055	70	91.6891	73	92.8282	74	94.0081	73	95.7627	70	96.1191	70
54	dte	91.6107	27	93.0549	30	93.7251	40	95.0056	34	96.3152	40	96.6647	30
55	dtmnc	91.8630	13	93.1811	20	93.7438	34	94.9664	47	96.3067	45	96.6051	46
56	dtch	89.5679	74	91.1332	76	92.6576	76	93.8973	76	95.5018	73	95.8514	73
57	dtcan	91.3805	33	92.5025	62	92.7617	75	93.9859	75	94.7703	77	94.8726	76
58	dtas	91.5476	29	92.9782	37	93.7660	26	95.0039	35	96.3084	43	96.6323	37
59	dtchi	91.7266	22	93.0498	32	93.7012	45	95.0227	28	96.2657	57	96.4976	54
60	dtchi2	91.7386	21	93.1061	24	93.7268	37	95.0124	33	96.2708	50	96.5198	52
61	dtbc	91.8102	15	93.0601	26	93.6841	47	95.0005	37	96.2674	53	96.4755	56
62	dthl	91.6090	28	92.9305	42	93.7097	44	94.9886	44	96.2316	60	96.4482	62

Табела 3.1 (продолжува):

#	метод	1000	R	2000	R	5000	R	10000	R	30000	R	сите	R
63	dtchr	91.5476	29	92.9782	37	93.7660	26	95.0039	35	96.3084	43	96.6323	37
64	dtchr2	91.6533	26	92.9816	36	93.7268	37	94.9783	46	96.2248	62	96.4755	56
65	dtjac	91.8102	15	93.0601	26	93.6841	47	95.0005	37	96.2674	53	96.4755	56
66	dtper	91.8102	15	93.0601	26	93.6841	47	95.0005	37	96.2674	53	96.4755	56
67	dtse	91.7744	20	93.1709	22	93.6927	46	94.7925	62	96.0304	68	96.1600	69
68	dtfr	92.1427	8	93.6790	8	93.9587	10	94.9067	55	96.2265	61	96.4755	56
69	dtm	93.8325	2	94.4770	2	94.6612	3	95.7661	2	96.7432	4	96.9324	6
70	dtmc	93.5256	4	94.2332	4	94.4907	4	94.9630	49	95.5359	72	95.9571	72
71	epw	90.1767	68	92.5894	60	93.6807	51	94.9971	41	96.5488	18	96.7432	17
72	gc	90.8706	55	92.9185	44	93.7234	41	95.2836	6	96.6255	7	96.8557	9
73	mgc	91.6721	24	93.2442	16	93.9501	12	95.4370	5	96.6375	6	96.9069	8
74	ogc	91.9159	12	93.5614	9	94.1906	5	95.5393	4	96.8216	2	97.0433	2
75	omgc	91.2986	40	92.8009	49	93.5648	58	94.8181	61	96.6954	5	97.0058	3
76	mlr	93.7711	3	94.4395	3	94.6697	2	95.7457	3	96.7585	3	96.9461	4
77	svmcmb	97.4099	1	97.3638	1	97.3690	1	97.4764	1	97.7799	1	97.8225	1
	joinedin	92.8554		94.1019		94.3611		95.8446		97.1473		97.2735	
	oracle	96.2384		96.8779		97.1439		97.8788		98.4722		98.6990	
	cdc	0.668		0.6283		0.6768		0.7011		0.7361		0.767	

Нашите истражувања покажуваат дека комбинирањето со употреба на класификатор (77) е неприкосновено и постига најдобри перформанси во сите случаи независно од бројот, карактеристиките и зависноста на класификаторите кои се комбинираат. Но, сепак ваквото комбинирање со дополнителен класификатор е најкомплексно и за негова обука е потребно незанемарливо време и дополнителни обележани примероци кои не учествувале во обуката на основните класификатори.

Со зголемувањето на бројот на примероци кои се употребуваат при обуката на индивидуалните класификатори, се разбира нивните перформанси растат, а со тоа и перформансите на нивната комбинација. Но, со порастот на перформансите на индивидуалните класификатори расте и нивната зависност, а со тоа се стеснува просторот за подобрување на резултатите при комбинирањето. Кај различните методи овој пораст не е со исто темпо, што може да се забележи преку промената на рангот на методот во зависност од употребените примероци за обука. Кај најголемиот број методи за комбинирање се забележува дека со зголемувањето на бројот на примероци употребени за обука на индивидуалните класификатори, нивниот ранг генерално или расте или опаѓа (многу ретко осцилира). Според ова повеќето од методите можат да се

карактеризираат како подобри за комбинирање на „слаби“ или на „јаки“ класификатори. Исто така осетливоста на комбинацијата со еден „слаб“ класификатор во друштвото на неколку „јаки“ класификатори е различно за ралзичните методи за комбинирање.

Техниките за комбинирање базирани на гласање (1-4) се меѓу нај-слабите во спроведените експерименти. Ова пред сè се должи на фактот што истите за комбинирање ги користат конечните „тврди“ одлуки на индивидуалните класификатори. При ова од комбинирањето се отфрла значителен дел од корисната информација, како кој е вториот избор, која е доверливоста на одлуката, каква е распределбата на поддршките за другите класи, кои голем број на класификатори ги генерираат. Ефикасноста на гласањето пред сè зависи од бројот на индивидуални класификатори кои гласаат. Од спроведените експерименти на комбинирање на 2, 3 и 4 класификатори, евидентно е дека со зголемувањето на бројот на индивидуални класификатори кои гласаат, се зголемува и ефикасноста на овој метод. Исто така, кога се комбинираат релативно мал број на класификатори (како во нашите експерименти), во голем број случаи доаѓа до поделба на гласовите и ниту една класа не се издвојува како јасен победник, со што можноста да се згреши или дури да мора случајно да се избира некоја од класите (ако се исклучено отфрлањето на примероци) е многу зачестена. Според ова, гласањето како метод за комбинирање иако доста често користена во пракса, има оправдување само доколку се комбинираат класификатори со излез на апстрактно ниво, односно кои даваат само конечна класа за примерокот кој го препознаваат без дополнителни информации и ако се комбинираат релативно поголем број (>10) класификатори. Од друга страна, доколку е потребно отфрлање на несигурните примероци поради постигање пови-сока доверливост на системот, кај гласањето тоа наједноставно се изведува бидејќи е практично вклучено во самиот метод (гласање со консензус, квалификувано мнозинство, просто мнозинство). Сепак при ва-ка изведеното отфрлање, не може фино да се нагодува бројот на при-мероци кои ќе бидат отфрлени за да се постигне одредена доверливост, односно прецизноста на ваквото нагодување директно зависи од бројот на употребени класификатори.

Методот за комбинирање според пребројувањето на Борда (2) да-ва секогаш подобри резултати од гласањето. Затоа, иако нешто поком-плексна за изведба, препорачливо е да се применува секогаш кога кла-сификаторите кои се комбинираат се од типот 2 (како излез даваат ранг листа на класи). За спроведените експерименти, оваа листа се добива со едноставно подредување на излезите од класификаторот за класите во опаѓачки редослед.

BKS методот (3) е единствениот обучен метод за комбинирање на класификатори од типот 1. Резултатите кои се добиваат со него се секогаш подобри од гласањето, но тој е сепак доста комплексен (особено мемориски). Предложената модификација на BKS методот (4) со дополнително гласање отстранува еден битен недостаток на овој метод, а тоа е што да се прави со примероците за кои распределбата на поддршките од класификаторите не е регистрирана во процесот на обука. Од табелите се гледа дека во сите случаи модифицираниот BKS метод со дополнително гласање дава подобри резултати од оригиналниот.

Кај необучените методи (5-12) за комбинирање (avg, prod, harm, med, sprod, minmax, maxmax, davg) забележлива е предноста при комбинирање на појаки класификатори. Генерално рангот на необучените методи се подобрува кога се комбинираат класификатори обучени со поголем број на примероци, иако има одредени осцилации. Необучените методи се инфериорни во однос на обучените особено кога се комбинираат слаби класификатори и класификатори со значително различни перформанси. При комбинирањето на јаки класификатори со слични перформанси обучените методи ја губат оваа предност па некои од необучените методи во ваквите случаи успеваат да се вбројат меѓу неколкуте најдобри методи. Така аритметичката и геометриската средина, (5, 6) медијаната (11) и тежинското комбинирање со динамичко одредување на тежините (12), се меѓу методите кои даваат најдобри резултати кога се комбинираат јаки класификатори со слични перформанси. Хармониската средина (7) и комплементарниот производ (8) се нешто полош избор, но секогаш подобри од изборот на максимум или минимум (9,10).

Методот за комбинирање со **веројатносен производ** (40) не се разликува многу од геометриската средина (6) пред сè затоа што во конкретниот проблем на препознавање на цифри класите се прилично рамномерно застапени во множествата за обука, валидација и тестирање. Карактеристично за комбинирањето според баесовиот метод (41) е тоа што тој е подобар од повеќето останати методи во случаите кога се комбинираат слаби класификатори со значително различни перформанси и кога се комбинираат повеќе класификатори.

Методот на **неспецифицираниот интеграл** (42) не се истакнува како особено добар но карактеристичен е по тоа што е најмалку зависна од „јачината“ на класификаторите. Модификација на комбинирањето со неспецифициран интеграл (43) каде неспецифицираните густини се одредуваат одвоено за секоја класа и секој класификатор има предност над стандардниот метод само кога се комбинираат класификаторот со слични перформанси каде е поверојатно да се случи некој од нив да е подобар за едни а другиот за други класи.

Комбинирањето на класификатори со **шаблони за одлучување** (44-70) и според **Демпстер-Шефер** теоријата (13-39), (испитани се по 27 варијации), зависат доста од применетите постапки за одредување на сличноста на актуелните излези од класификаторите со нивните просечни (очекувани) вредности за секоја од класите. Генерално, комбинирањето според Демпстер-Шефер теоријата дава подобри резултати од шаблоните за одлучување кога се комбинираат појаки класификатори, додека при комбинирањето на слаби класификатори предноста е на страната на шаблоните за одлучување. Од аспект на применетите мерки за сличност од кои произлегуваат варијациите и кај шаблоните за одлучување и кај Демпстер-Шефер методот, мерките базирани на неспецифицираните припадности може да се поделат на две групи: интегрални мерки ($S_1, S_2, S_3, I_1, I_2, I_3$) и локални (pointwise) мерки (S_4, I_4, I_5, C) на припадност. Првите се засновани на кардиналноста, додека вторите мерат единствен степен на припадност и се поподложни на влијание на грешките на индивидуалните класификатори. Од останатите мерки за сличност, кај Демпстер-Шефер методот како најдобра при комбинирање на слаби класификатори се истакнува Махаланобис растојанието со одвоени дисперзионии матрици за секоја класа, следено од Фрешеовото растојание. При комбинирањето на јаки класификатори предноста ја преземаат неспецифицираните мерки за сличност, пред сè S_3, S_4 и C . Агловото одвојување и Корд растојанието се подеднакво успешни и кај слабите и кај јаките класификатори. Кај шаблоните за одлучување, најдобри перформанси се добиваат со проширените шаблони и користењето на мерка за сличност изведена од растојанието на Махаланобис. Притоа, варијантата која користи различни дисперзионии матрици за секоја класа има предност само кога се комбинираат слаби класификатори. Интересно за овој метод е што тој одлично се снаоѓа во комбинациите каде некој од класификаторите е послаб од останатите. При комбинирањето на слаби класификатори со шаблоните за одлучување, меѓу мерките за сличност со подобри резултати се издвојуваат неспецифицираните мерки за сличност I_1 и S_2 , χ^2 и Менхетен растојанијата.

Тежинските методи за комбинирање (71-75) ги постигаат најдобрите перформанси кога се комбинираат јаки класификатори. Тежинските комбинирања за кои тежините се одредени според Хук-Џивс алгоритмот за оптимизација, се скоро секогаш подобри од генерализираниот комитет кој тежините ги одредува преку инверзијата на корелационата матрица. Модифицираниот обопштен комитет е во блага предност пред класичниот, додека кај оптимизираниите верзии подобри резултати дава класичниот пристап со единствени тежини. Ова најверојатно се должи на комплексноста на оптимизацискиот проблем кој се јавува при одредувањето на тежините на модифицираниот обопштен

комитет кој треба да оптимизира k пати повеќе тежини (каде k е бројот на класите) и поголемата веројатност да се заглави во локален минимум на многудимензионалната функција.

Како една од најдобрите методи за комбинирање се покажа и MLR методот (76) на **повеќекратна линеарна регресија**. Особено се истакнува при комбинирањето на класификатори со различни перформанси и на поголем број класификатори.

Неколку од методите за комбинирање кои покажаа најдобри резултати, како проширените шаблони за одлучување со Махаланобис растојание (69), Демпстер-Шефер методот со Махаланобис растојание (38) и генерализираниот комитет (72-75) почиваат на дисперзионите матрици на зависноста на излезите од класификаторите. Доколку оваа зависност се набљудува интегрално за сите случаи (кога на влезот на класификаторите се појавува примерок од било која класа) се добива единствена дисперзиона матрица. Ако пак зависноста се разгледува одделно за случаите кога на влезот на класификаторите се појавува примерок од одредена класа, се добиваат онолку дисперзни матрици колку што има различни класи во класификацискиот проблем (10 за препознавање на цифри). Дисперзионите матрици на одреден начин ја даваат зависноста меѓу класификаторите, а индиректно и зависноста меѓу множествата обележја над кои класификаторите се изградени.

Многу слично поведение има и комбинирањето на SVM класификатори со линеарен кернел. SVM класификаторите со линеарен кернел доста побргу се обучуваат, а и оптимизацијата на нивните параметри е побрза бидејќи има само еден слободен параметар кој треба да се оптимизира. Сепак тие имаат послаби перформанси и во нашите експерименти и индивидуалните класификатори и комбинациите покажаа околу 3% пониски рати на препознавање.

Забелешките за предноста на одредени методи за комбинирање генерално важат и при комбинирањето на SVM класификаторите со линеарен кернел, со тоа што зависноста од бројот на примероци со кој се обучени класификаторите е многу помалку изразена. Поизразена предност имаат проширените шаблони за одлучување, особено со примена на растојанието на Махаланобис, за сметка на генерализираниот комитет кој дава послаби резултати. Помеѓу методите кои даваат подобри резултати во комбинирањето се истакнуваат и шаблоните за одлучување и комбинирањето според Демпстер-Шефер теоријата кога го користат Фрешеовото, квадратното Корд и χ^2 растојанието.

Слично е и поведението при комбинирањето на невронските мрежи. Невронските мрежи исто така релативно побргу се обучуваат, а се значително побрзи при одлучувањето од SVM класификаторите. Нивните перформанси се околу 2% послаби од оние кај SVM класификато-

рите со гаусов кернел, а и релативните придобивки од комбинирањето се нешто поголеми отколку кај SVM-ата со линеарен кернел.

Од ова може да се изведе заклучок дека SVM класификаторите со линеарен кернел не се особено погодни за користење во ансамбл за комбинирање. Во случаите каде што е потребна што поголема точност по секоја цена, SVM класификаторите со гаусов кернел нудат многу подобри перформанси. Од друга страна пак, иако со нешто послаби перформанси, невронските мрежи побргу се обучуваат а при класификацијата се многу побрзи од SVM класификаторите.

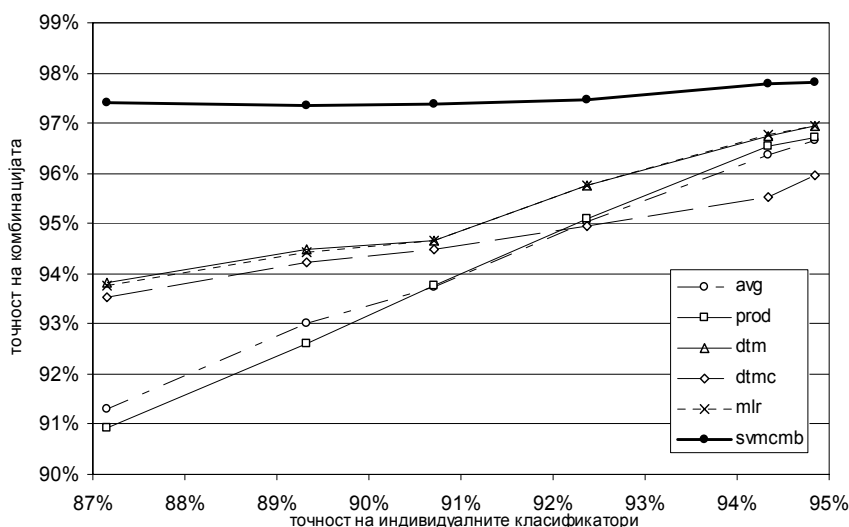
3.2 Обучени или фиксни методи на комбинирање

Од нашите резултати може да се заклучи дека предноста на обучените методи (3,4,13-77), за комбинирање на класификатори пред фиксните методи (1,2,5-12) е неспорна. Сепак може да се издвојат специјални случаи во кои примената на фиксен метод за комбинирање на класификатори е посоодветна. Примероците за обука на методот за класификација се доста важни за перформансите на комбинацијата и тоа пред сè со својот квалитет. Примероците кои биле употребени за обука на индивидуалните класификатори не можат да се искористат и за обука на методот за комбинирање на ваквите класификатори. За овие примероци сите класификатори оптимално ќе одлучуваат и од ваквото нивно однесување нема да може да се согледаат нивните индивидуални предности и слабости и нивните меѓузависности кои обучените методи се обидуваат да ги искористат при нивното комбинирање. Затоа, доколку не се располага со дополнителни примероци за обука на комбинаторот подобро е да се користи фиксен метод за комбинирање.

Перформансите на фиксните методи за комбинирање директно зависат од перформансите на индивидуалните класификатори и од нивниот број. Колку што се побројни индивидуалните класификатори и колку што се нивните перформанси подобри и меѓусебно слични по вредност, толку се подобри и резултатите од фиксните комбинирања. Кај обучените методи овие зависности не се толку изразени.

Кај обучените методи за комбинирање со зголемувањето на бројот на индивидуалните класификатори кои се комбинираат, се зголемува и бројот на независни примероци потребни за нивна успешна обука. Затоа од практични причини бројот на индивидуалните класификатори кај обучените методи за комбинирање обично е релативно помал. Исто така, сè додека располагаме со доволно примероци за обука на комбинаторот, перформансите на индивидуалните класификатори се од многу помало значење. Резултатите при употребата на методите за комбинирање кои екстензивно ги користат меѓузависностите на класификато-

рите (како комбинирањето со класификатор (77), Махаланобис растојанието кај шаблоните за одлучување (69.70) и повеќекратната линеарна регресија (76)) се далеку помалку зависни од перформансите на индивидуалните класификатори за разлика од фиксните методи кај кои оваа зависност е скоро линеарна (слика 3.1).



Слика 3.1. Зависност на перформансите на комбинацијата од просечните перформанси на индивидуалните класификатори

На слика 3.1 е дадена зависноста на резултатите добиени при комбинирање на четирите класификатори со некои од најдобрите фиксни и обучени методи во однос на просечните перформанси на индивидуалните класификатори кои се комбинираат. Од сликата се гледа дека комбинирањето со помош на класификатор скоро воопшто не зависи од перформансите на индивидуалните класификатори. Кај другите обучени методи оваа зависност е помалку изразена, додека кај фиксните методи таа е скоро линеарна.

3.3 Комбинирање на класификатори со различни перформанси

Нашите искуства покажуваат дека најдобро е да се комбинираат класификатори со слични перформанси. Повеќето од методите за комбинирање постигаат подобри резултати кога од комбинацијата ќе се отстрани класификаторот со битно различни рати на препознавање кој ги „збунува“ останатите. Овој резултат интуитивно постојано се јавува во реалниот живот. На пример, во размена на мислења меѓу луѓе во ед-

на група, интуитивно најмногу може да се постигне ако тие се приближно со исти знаења и интуиција.

Ова логична идеја, потврдена во нашите експерименти наметнува потреба од посебни процедури за селекција на обележја што на оптимален начин ќе го разбиваат множеството обележја за системите што користат комбинирање на класификатори.

Да претпоставиме дека имаме множество обележја S коешто сакаме да го разбиеме на k дисјунктни подмножества $S_1, S_2, \dots, S_k$ со приближно еднакви перформанси $J(S_1) \approx J(S_2) \approx \dots \approx J(S_k)$. Под перформанси на множеството обележја $J(S)$ стандардно ја подразбираме ратата на препознавање на класификаторот дизајниран над S . Условот подмножествата S_i да се со слични перформанси може да се претстави во следнава форма:

$$\max_{i,j} |J(S_i) - J(S_j)| \leq \delta \quad \text{или} \quad \max_i J(S_i) - \min_j J(S_j) \leq \delta$$

каде што δ е саканата максимална разлика во перформансите на подмножествата обележја. На почеток, ако не располагаме со некоја додатна информација, можеме на секое подмножество S_i да му доделиме $|S|/k$ или $|S|/k+1$ обележја. Така, на почеток бројот на обележја меѓу кои било две подмножества се разликува најмногу за еден.

Следниот алгоритам е еден можеен пристап кон решавање на проблемот како на прифатлив начин да се разбие почетното множество обележја на дисјунктни подмножества над кои се дизајнираат класификатори чишто одлуки потоа се комбинираат.

```

m ← mod(|S|, k)
for i ← 1 to m do { Si ← random избрани  $\frac{|S|}{k} + 1$  обележја од S
                   S ← S - Si
for i ← m + 1 to k do { Si ← random избрани  $\frac{|S|}{k}$  обележја од S
                       S ← S - Si
Smax ← подмножество Si со најголемо J(Si)
Smin ← подмножество Si со најмало J(Si)
while J(Smax) - J(Smin) > δ
do { // се зема едно обележје од Smax и се става во Smin
    f ← random избрано обележје од Smax
    Smax ← Smax - {f}
    Smin ← Smin ∪ {f}
    Smax ← подмножество Si со најголемо J(Si)
    Smin ← подмножество Si со најмало J(Si)

```

Со „разумно“ избрана вредност на δ , овој алгоритам секогаш ќе заврши со урамнотежено разбивање на множеството обележја на k дисјунктни подмножества $S_1, S_2, \dots, S_k$ чиишто перформанси се разликуваат за помалку од δ .

Како што веќе напомнавме, под $J(S)$ стандардно ја подразбираме ратата на препознавање на класификаторот дизајниран над S . Во одредени случаи перформансите на множествата S_i , можат да се проценат преку некоја друга мера за квалитет на едно множество обележја, како што тоа се прави во методите за селекција на обележја [23]. Така на пример, $J(S)$ може да ги рефлектира внатрешните особини на множеството обележја: нивната зависност, корелираност или конзистентност. Исто така перформансите на едно множество обележја може да се проценуваат преку неговата корелираност со другите множества. Сепак останува фактот дека најсигурен начин за проценка на перформансите на едно множество обележја е ратата на препознавање на класификаторот дизајниран над нив.

3.4 Комплексност на методите за комбинирање

Пресметковната комплексност на различните методи за комбинирање на класификатори, може да се разгледува од два аспекта: комплексност на обуката на методите за комбинирање и комплексност при нивната примена, односно при самото комбинирање. Комплексноста на обуката пред сè зависи од бројот и квалитетот на расположивите примероци за обука. Најкомплексно за конструкција е комбинирањето со помош на класификатор. Доста комплексно е и одредувањето на тежинските фактори со Хук-Џивс оптимизацијата. Следни по комплексност се методите кои вклучуваат матрични операции вклучувајќи и инверзија на релативно големи матрици чии димензии зависат од бројот на класификаторите и бројот на класите. Во оваа група спаѓаат повеќекратната линеарната регресија, како и методите кои вклучуваат одредување на дисперзиона односно корелациона матрица на грешки и нивна инверзија, како генерализираниот комитет, и методите кои користат растојание на Махаланобис. Помала комплексност имаат методите кои вклучуваат одредување на конфузиона матрица (Баес, Демпстер-Шефер и неспецифицираниот интеграл).

Комплексноста на методите при нивната примена не зависи директно од комплексноста при нивната обука. Комплексноста на комбинирањето со помош на класификатор зависи од карактеристиките на применетиот класификатор. Така на пример, ако како класификатор се употреби невронска мрежа, комплексноста е незначителна и споредлива со некои од другите методи. Примената на SVM класификатор е

многу покомплексна. Сите методи базирани на генерализиран комитет имаат иста комплексност при нивната употреба независно на кој начин се одредени тежините и се релативно едноставни. Освен неспецифицираниот интеграл кој е покомплексен, бидејќи вклучува сортирање на поддршките во зависност од неспецифицираните густини за секој примерок кој се препознава поодделно, останатите методи имаат слична комплексност. Единствено оние методи кои вклучуваат одредување на растојанието на Махаланобис се нешто покомплексни, поради множењето со инверзната дисперзиона матрица. БКС методот е посебен случај и тој не е особено пресметковно комплексен, но може да биде значително мемориски комплексен и при обуката и при користењето.

Од друг аспект методите за комбинирање можат да се разгледуваат како статички и динамички. Кај статичките методи, со истите излези од индивидуалните класификатори се преземаат секогаш исти операции со однапред одредени коефициенти, сè до максимум селекторот каде што конечно се одредува класата на примерокот. Динамичките методи вклучуваат дополнително прередување и оперирање со различни коефициенти кои се одредуваат за секој примерок одделно (тежинско комбинирање со динамичко одредување на тежините и неспецифициран интеграл). Од овој аспект, статичките методи се многу поедноставни за хардверска имплементација.

3.5 Зависноста на перформансите на обучените методи за комбинирање на класификатори од бројот на примероци за нивна обука

Претходно наведените заклучоци беа изведени на основа на снимените перформанси на комбинирање на класификатори со различни распознавачки способности моделирани преку бројот на примероци употребени при нивната обука. За да се испита влијанието на бројот на примероци употребени за обука на обучените методи на нивните перформанси спроведен беше и следниот експеримент: Комбинирањето на излезите од сите четири индивидуални класификатори беше изведено со обучените методи за комбинирање при што истите беа обучени употребувајќи по 1000, 2000, 5000, 10000, 30000 и сите примероци и тоа за два случаи: кога индивидуалните класификатори се обучени со само 1000 примероци и кога истите се обучени со сите примероци. Со ова сакавме да го испитаме влијанието на бројот на примероци употребени за обука на обучените методи во случај на комбинирање на слаби и во случај на за комбинирање на јаки индивидуални класификатори.

Од добиените резултати може да се заклучи дека бројот на примероци со кои се обучуваат методите за комбинирање кај најголем број од нив не игра преголема улога кога се комбинираат слаби класификатори. Најдобар начин за комбинирање е повторно употребата на класификатор за комбинирање (77). Помеѓу методите кои постигнуваат добри резултати во комбинирањето на слаби класификатори независно од расположливите примероци за нивна обука се вбројуваат проширените шаблони за одлучување со Махаланобис растојание (69,70) и повеќекратната линеарна регресија (76). Поевидентна зависност од бројот на примероци се забележува кај комбинирањето според Демпстер-Шефер теоријата и кај проширените шаблони за одлучување кога кај истите се користи Махаланобис растојанието со одвоени корелациони матрици за секоја класа. Ова најверојатно се должи на одвоеноста на корелационите матрици за чија проценка со приближно иста прецизност како и единствената се потребни околу 10 пати повеќе примероци. Одредена зависност од бројот на примероци за обука на методите за комбинирање се забележува и кај Фрешеовото растојание (37,68), како и кај модифицираниот VKS метод (4).

При комбинирањето на јаки класификатори диверзитетот на нивните одлуки е многу помал, па станува сè потешко обучените методи за комбинирање да ги воочат нивните зависности и ефикасно да се прилагодат со цел да постигнат што подобри резултати. Во ваквиот случај, комбинирањето со употреба на класификатор го губи приматот кога има ограничен број на примероци за негова обука. Кога бројот на примероци е поголем (>10000 во нашите експерименти) комбинирањето со употреба на класификатор (77) ги дава повторно најдобрите резултати. При комбинирањето на јаки класификатори, комбинирањето според Демпстер-Шефер теоријата со употреба на Корд растојание и аглово одвојување (32,33,27), како и проширените шаблони за одлучување со Махаланобис растојание (69,70) и повеќекратната линеарна регресија (76) се меѓу најдобрите методи независно од бројот примероци за нивна обука. Методите кај кои е најмногу изразена зависноста од бројот на примероци за нивна обука се комбинирањата базирани на генерализиран комитет (72-75).

3.6 Комбинирање на еден-против-еден класификатори

Комбинирањето на еден-против-еден класификаторите може да се изведе во основа на три различни начини (поглавје 2.22), при што првиот начин е еквивалентен на комбинирањето на еден-против-сите класификаторите по обединувањето на одлуките од сите еден-против-

еден класификатори за истото множество на обележја. Перформансите на индивидуалните SVM класификатори во еден-против-еден шемата се слични на SVM класификаторите во еден-против-сите шемата за истото множество на обележја но скоро секогаш малку послаби од нив.

Комбинирањето на еден-против-еден класификаторите со методите на: шаблони за одлучување, Демпстер-Шефер, Баес, генерализиран комитет, повеќекратната линеарна регресија и комбинирање со помош на класификатор, беа изведени со интегрално комбинирање на сите двокласни класификатори (третиот начин за комбинирање). Останатите методи за комбинирање беа спроведени индивидуално над L -торките соодветни двокласни класификатори, а потоа беа изведени поддршките за секоја од класите според Max Wins стратегијата. Слично како и при комбинирањето на еден-против-сите класификаторите убедливо најдобри резултати се добиваат при комбинирањето со класификатор. Од останатите методи за комбинирање како поефикасни повторно се издвојуваат шаблоните за одлучување со Махаланобис растојание, повеќекратната линеарна регресија и генерализираниот комитет. За разлика од комбинирањето на еден-против-сите класификатори, при комбинирањето на еден-против-еден класификаторите доста повисоко беше рангиран методот за баесово комбинирање. Во споредба со перформансите на комбинирање на еквивалентно обучените еден-против-сите класификатори, резултатите при комбинирањето на еден-против-еден класификаторите со фиксните методи за комбинирање се околу 0.5% послаби. Перформансите на еден-против-еден класификаторите комбинирани со обучените методи за комбинирање се подобри од истите на еден-против-сите класификаторите кога се комбинираат „слаби“ индивидуални класификатори (обучени со помалку од 10000 примероци). Кога се комбинираат „јаки“ класификатори, предноста е повторно на страната на комбинирањето на еден-против-сите класификаторите за околу 0.5%. Некои од неспецифицираните мерки на сличност кои се користат кај шаблоните за одлучување и кај Демпстер-Шефер комбинирањето даваат доста лоши резултати најмногу поради нивната неможност да се прилагодат на диверзитетот на одлуките на некои од класификаторите при препознавањето на примероци од класи за кои не биле обучувани, односно се некомпетентни за таа класа.

Заклучок

Изложените методи за комбинирање на класификатори можат да се поделат во по две групи и тоа од два независни аспекта. Според првиот аспект, поделбата е на обучени и фиксни методи за комбинирање на класификатори во однос на тоа дали им е потребана претходна обука, односно одредување на некои параметри на основа на однесувањето на класификаторите измерено на множеството за валидација или не. Втората поделба е од аспект на тоа дали методот се однесува идентично во сите случаи, односно извршува идентични операции над секој од излезите од индивидуалните класификатори за секој примерок кој се препознава или пак тие се менуваат од случај до случај. Според оваа поделба методите можат да се поделат на статички и динамички. Кај динамичките методи одредени параметри на комбинирањето се одредуваат одделно за секој примерок кој се препознава. Кај обучените статички методи, откако еднаш ќе се одредат параметрите на основа на проценетото однесување на класификаторите при препознавањето на сите примероци од множеството за валидација, вака фиксираните параметри потоа се користат за комбинирање при препознавањето на нови примероци. Логично, статичките методи се многу поедноставни за изведба од динамичките. Во следната табела е дадена поделбата на изложените методи за комбинирање од овие два аспекта.

Методи за комбинирање	Фиксни	Обучени
Статички	- гласање - сите усреднувања (аритметичка, геометриска, хармониска средина, комплементарен производ) - сите подредувања (минимум, максимум, медијана)	- BKS - веројатносен производ - Баесово правило - обопштен комитет - шаблони за одлучување - MLR
Динамички	- пребројување според Борда - тежинско усреднување со динамички тежини	- Демпстер-Шефер комбинирање - неспецифициран интеграл - комбинирање со класификатор

Комбинирањето со класификатор е обучен метод, но дали ќе биде статички или динамички зависи од типот на користениот класификатор.

Бидејќи резултатите добиени со комбинирање на класификатори директно зависат и од карактеристиките на самите класификаторите, а со цел подобро да се процени универзалноста или специфичноста на одредени методи за комбинирање на класификатори, во експериментите се разгледувани два вида класификатори: невронски мрежи и машини со носечки вектори со линеарен и гаусов кернел. Експериментите се изведени со класификатори за препознавање на броеви напишани со рака, но голем број од изведените заклучоци не зависат од доменот на примерна на класификаторите и важат генерално за комбинирање на класификатори кои решаваат произволен класификациски проблем.

Според резултатите на спроведените експерименти комбинирањето со употреба на класификатор е неприкосновен шампион во перформансите, но воедно и најкомплексно за реализација. При комбинирањето на слаби (и/или хетерогени) класификатори предност имаат обучените методи за комбинирање меѓу кои особено се истакнуваат: шаблоните за одлучување со Махаланобис растојание, повеќекратната линеарна регресија и Демпстер-Шефер комбинирањето со Махаланобис растојание. Кај комбинирањето на јаки класификатори обучените методи немаат толку изразена предност пред фиксните меѓу кои најдобри се аритметичката и геометриската средина и медијаната.

Генералниот заклучок е дека обучените методи за комбинирање постигнуваат видливо подобри перформанси, но се покомплексни за реализација и бараат дополнителни примероци за обука на комбинаторот. Од обучените методи најдобри перформанси во препознавањето се постигаат при користењето на обучен класификатор за комбинирање на излезите од индивидуалните класификатори. Овој метод е и најкомплексен и најбавен (и при обуката и при препознавањето).

Нашите истражувања јасно ја покажуваат предноста на SVM класификаторите пред невронските мрежи од аспект на резултатите на препознавањето. Но, SVM класификаторите се исто така и многу покомплексни и побавни особено при препознавањето на примероците. Со цел да се искористат предностите на SVM класификаторот без жртвување на брзинските перформанси на систем за препознавање на броеви напишани со рака, ние имаме предложено ефикасен тростепен класификатор составен од две невронски мрежи во првите два степен и SVM класификатор во третиот степен. Спроведените тестовите за препознавање на изолирани цифри од MNIST (<http://yann.lecun.com/exdb/mnist>) базата со предложениот систем, покажаа забрзување во препознавањето од скоро 28 пати и мало подобрување на точноста во од-

нос на индивидуалниот SVM класификатор [61] (4-ти труд во додатокот). Добиените резултати се меѓу најдобрите резултати за MNIST базата на ракописни цифри.

Во следните точки се дадени некои поважни прашања во врска со комбинирањето на класификатори:

- Колку навистина се корисни техниките за комбинирање на класификатори базирани на правила (rules). Нашите искуства покажуваат дека ваквите шеми се доста „крути“ и инфериорни во однос на статистичките [57][58][59];
- Колку може да бидат корисни секвенцијалните методи на комбинирање на класификатори каде што секој следен класификатор работи со отфрлените (неселектирани) примероци [25] од претходниот. Нашите резултати покажуваат дека ваквите пристапи може да водат до многу ефикасни (временски) системи за препознавање примероци [61];
- Какво ќе биде однесувањето на методите за комбинирање кога бројот на класите значително ќе се зголеми (на пр. 100 и повеќе);
- Дали групирањето на обележјата според начинот на нивна екстракција од сликата на знакот е најсоодветно за добивање на некорелирани класификатори, односно доколку сите обележја се разгледуваат како единствено множество на кој начин да се разделат на неколку групи од кои би произлегле независни класификатори погодни за комбинирање [60][64];
- Кога се располага со ограничен број на означени примероци кои треба да се искористат за решавање на класификациски проблем дали вреди дел од нив да се одвои за неопходната обука на некоја од обучените методи за комбинирање или можеби е подобро тие да се искористат во обуката на индивидуалните класификатори кои потоа би се комбинирале со некоја од фиксните методи;
- Комбинирањето на класификатори со стратегии еден-против-еден или еден-против-неколку наспроти стандардната стратегија кај овој тип класификатори еден-против-сите во нашите почетни испитувања се покажаа инфериорни. Сепак, би требало да се испитаат различни методи на комбинирање со овие стратегии за да се даде конечен суд колку тие може да бидат корисни.

На крај, да се навратиме на суштинското прашањето: Дали комбинирањето на класификатори креирани над различни фамилии обележја е навистина корисно? Се разбира конечен одговор не може да се даде. Нашите истражувања искристализираа некои битни заклучоци. Како прво, многу е тешко со комбинирање на класификатори дизајни-

рани над дисјунктни групи обележја да се достигне (и надмине) ратата на препознавање на еден добро обучен класификатор дизајниран над целото множество обележја. Особено тоа е тешко да се направи кај SVM-те. Зошто да правиме систем за препознавање облици со комбинирање на класификатори ако истиот класификациски проблем може подобро да се реши со еден добро дизајнирана SVM класификатор [90]. Сепак, како второ би заклучиле дека комбинирањето на класификатори преку фузија на одлуките ја намалува комплексноста на целиот класификатор, бара помалку примероци и време за обука и нуди повеќе можности за користење на најразлични критериуми за отфрлање на несигурните одлуки.

Литература

- [1] Ahmed Al-Ani, Mohamed Deriche, A New Technique for Combining Multiple Classifiers Using The Dempster-Shafer Theory of Evidence, *Journal of Artificial Intelligence Research*, Vol. 17, pp. 333-361, 2002.
- [2] William W. Armstrong, Jan Gecsei, Adaptation Algorithms for Binary Tree Networks, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, No. 5, May 1979.
- [3] William W. Armstrong, Andrew Dwelly, Jiandong Liang, Dekang Lin, Scott Reynolds, Some Results Concerning Adaptive Logic Networks, *TR, University of Alberta, Edmonton, Alberta, Canada T6G 2H1*, September 1991.
- [4] Robert R. Bailey, Mandyam Srinath, Ortogonal Moment Features for Use With Parametric and Non Parametric Classifiers, *IEEE Transactions on PAMI*, Vol. 18, No. 4, pp. 389-399, 1996.
- [5] R. Battiti, A.M. Colla, Democracy in Neural Nets: Voting Schemes for Classification, *Neural Networks*, Vol. 7, pp. 691-707, 1994.
- [6] M. Bell, M. C. Pike, Remark on Algorithm 178 [E4] Direct Search, *Communications of the ACM*, Vol. 9, No. 9, pp. 684-685, 1966.
- [7] J.A. Benediktsson, P.H. Swain, Consensus Theoretic Classification Methods, *IEEE Tran. on Systems, Man and Cybernetics*, Vol. 22, pp. 688-704, 1992.
- [8] J.A. Benediktsson, J.R. Sveinsson, J.I. Ingimundarson, H. Sigurdsson, O.K. Ersoy, Multistage Classifiers Optimized by Neural Networks and Genetic Algorithms, *Nonlinear Analysis Theory Methods Applications*, Vol. 30, No. 3, pp. 1323-1334, 1997.
- [9] J. C. Bezdek, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic, Boston, 1999.
- [10] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- [11] D. Black, *The Theory of Committees and Elections*, Cambridge University Press, London, 1958, reprinted 1963.
- [12] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, T. Vetter, Comparison of View-based Object Recognition Algorithms Using Realistic 3D Models, in C, *Artificial Neural Networks – ICANN'96*, pp. 251-256, Springer Lecture Notes in Computer Science, Vol. 1112, Berlin, 1996.
- [13] F. Blayo, Y. Cheneval, A. Guérin-Dugué, C. Chentouf, C. Aviles-Cruz, J. Madrenas, M. Moreno, and J.L. Voz, Deliverable R3-B4-P – Task B4: Benchmarks. *Technical Report 6891*, ESPRIT Research Project, June 1995.
- [14] I. Bloch, Information Combination Operators for Data Fusion: A Comparative Review with Classification, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems & Humans*, Vol. 26, pp. 52-67, 1996.
- [15] Avrim L. Blum, Ronald L. Rivest, Training a 3-node Neural Network is (NP)-Complete, in D. Touretzky, editor, *Machine Learning: From Theory to Applications*, pp. 9-28, 1993.

- [16] Kristin Branson, Sameer Agarwal, Structured Principal Component Analysis, UCSD Technical Report, October, 2003.
- [17] Erin J. Bredensteiner, Kristin P. Bennett, Multicategory Classification by Support Vector Machines, *Computational Optimizations and Applications*, Vol. 12, pp 53-79, 1999.
- [18] Leo Breiman, Bagging Predictors, *Machine Learning*, Vol. 24, pp. 123-140, 1996.
- [19] Leo Breiman, Bias, Variance, and Arcing Classifiers, Technical Report 460, Statistics Department, University of California, 1996.
- [20] Leo Breiman, Arcing classifiers, *The Annals of Statistics*, Vol. 26, No. 3, pp. 801-849, 1998.
- [21] Christopher J.C. Burges, B. Schölkopf, Improving the Accuracy and Speed of Support Vector Learning Machines, *Advances in Neural Information Processing Systems 9*, pp. 375-381, MIT Press, Cambridge, MA, 1997.
- [22] Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, Vol. 2, No.2, pp. 955-974, 1998.
- [23] Dusan Cakmakov, Younès Bennani, *Feature Selection for Pattern Recognition*, Informa, Skopje, 2002.
- [24] Dusan Cakmakov, Dejan Gorgevik, Handwritten Digit Recognition Using Classifier Cooperation Schemes, *Proceedings of the 2nd Balkan Conference in Informatics, BCI 2005*, pp. 23-30, Ohrid, November 17-19, 2005.
- [25] Dusan Cakmakov, Vladimir Radevski, Younes Bennani, Dejan Gorgevik, Decision Fusion and Reliability Control in Handwritten Digit Recognition System, *Journal of Computing and Information Technology - CIT*, Vol. 10, No. 4, pp. 283-293, 2002.
- [26] M. Ceccarelli, A. Petrosino, Multifeature Adaptive Classifiers for Sar Image Segmentation, *Neurocomputing*, Vol. 14, pp. 345-363, 1997.
- [27] C.C. Chiang, H.C. Fu, A Divide-and-conquer Methodology for Modular Supervised Neural Network Design, *Proceedings of IEEE International Conference on Neural Networks*, Orlando, FL, pp. 119-124, 1994.
- [28] Sung-Bae Cho, Jin H. Kim, Multiple Network Fusion Using Fuzzy Logic, *IEEE Transactions on Neural Networks*, Vol. 6, No. 2, pp. 497-501, 1995.
- [29] Sung-Bae Cho, Jin H. Kim, Combining Multiple Neural Networks by fuzzy Integral and Robust Classification, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, pp. 380-384, 1995.
- [30] C. Cortes, Vladimir Vapnik, Support Vector Networks, *Machine Learning*, Vol. 20, pp. 273-297, 1995.
- [31] Koby Crammer, Yoram Singer, On the Algorithmic Implementation of Multi-class Kernel-based Vector Machines, *Journal of Machine Learning Research*, Vol 2, pp. 265-292, 2001.
- [32] Nelo Cristianini, John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, United Kingdom, 2000.
- [33] B.V. Dasarathy, B.V. Sheela, A Composite Classifier System Design: Concepts and Methodology, *Proceedings of IEEE*, Vol. 67, pp. 708-713, 1978.
- [34] Arthur P. Dempster, A Generalization of Bayesian Inference, *Journal of the Royal Statistical Society, Series B*, Vol. 30, No. 2, pp. 205-247, 1968.

- [35] T. G. Dietterich, Ensemble Methods in Machine Learning. in J. Kittler and F. Roli (Ed.) *First Int. Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science, Vol. 1857, pp. 1-15, New York Springer Verlag, 2000.
- [36] Jian-xiong Dong, Adam Krzyzak, Ching Y. Suen, A Local Learning Framework for Pattern Classification, *Proceedings of the 14th International Conference on Vision Interface*, pp. 220-227, Ottawa, Canada, June 2001.
- [37] Jian-xiong Dong, Adam Krzyzak, Ching Y. Suen, Local Learning Framework for Recognition of Lowercase Handwritten Characters, *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 226-238, Leipzig, Germany, July 2001.
- [38] Jian-xiong Dong, Adam Krzyzak, Ching Y. Suen, Local Learning Framework for Handwritten Character Recognition, *Engineering Applications of Artificial Intelligence*, Vol. 15, No. 2, pp. 151-159, 2002.
- [39] T. Denouex, A K-nearest Neighbor Classification Rule Based on Dempster-Shafer Theory, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 5, pp. 804-813, 1995.
- [40] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, V. Vapnik, Boosting and Other Ensemble Methods, *Neural Computation*, Vol. 6, No. 6, pp. 1289-1301, 1994.
- [41] D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, NY, 1980.
- [42] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, ISBN: 0-471-05669-3, John Wiley & Sons, 2000.
- [43] Yoav Freund, Robert E. Schapire, Experiments with a New Boosting Algorithm, *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148-156, 1996.
- [44] Jerome H. Friedman, Another Approach to Polychotomous Classification, Technical report, Department of Statistics, Stanford University, 1997.
- [45] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Academic Press, Inc., New York, 1990.
- [46] G. Fumera, F. Roli and G. Giacinto, Reject Option with Multiple Thresholds, *Pattern Recognition*, Vol. 33, No. 12, pp. 165-167, 2000.
- [47] G. Fumera, F. Roli, G. Vernazza, A Method for Error Rejection in Multiple Classifier Systems, *Proc. 11th Int. Conf. on Image Analysis and Processing ICIAP 2001*, Palermo, Italy, IEEE Computer Society Ed., pp. 454-458, 2001.
- [48] P.D. Gader, M.A. Mohamed, J.M. Keller, Fusion of Handwritten Word Classifiers, *Pattern Recognition Letters*, Vol. 17, pp. 577-584, 1996.
- [49] Stuart Geman, E. Bienenstock, R. Doursat, Neural Networks and the Bias/Variance Dilemma, *Neural Computation*, Vol. 4, pp. 1-58, 1991.
- [50] Giorgio Giacinto, Fabio Roli, Methods for Dynamic Classifier Selection, *Proceedings of 10th International Conference on Image Analysis and Processing, ICIAP '99*, Venice, Italy, pp. 659-664, 1999.
- [51] Giorgio Giacinto, Fabio Roli, A Theoretical Framework for Dynamic Classifier Selection, *Proceedings of 15th International Conference on Pattern Recognition ICPR 2000*, Barcelona, Spain, Sept. 3 - 8, Vol. 2, pp. 8-11, 2000.
- [52] Giorgio Giacinto, Fabio Roli and G. Fumera, Selection of Classifiers Based on Multiple Classifier Behavior, *Proc. of the Joint IAPR Int. Workshops on Syntactical, Structural and Statistical Pattern Recognition*, Alicante, Spain, Lecture Notes in Computer Science 1876, Springer, pp. 87-93, 2000.

- [53] Giorgio Giacinto, Fabio Roli, Dynamic Classifier Selection, *Proceeding of the First Int. Workshop on Multiple Classifier Systems*, Cagliari, Italy, Lecture Notes in Computer Science 1857, Springer, Berlin, pp. 177-189, 2000.
- [54] Giorgio Giacinto, Fabio Roli, An Approach to the Automatic Design of Multiple Classifier Systems, *Pattern Recognition Letters*, Vol. 22, No. 1, pp. 25-33, 2001.
- [55] Giorgio Giacinto, Fabio Roli, Dynamic Classifier Selection Based on Multiple Classifier Behavior, *Pattern Recognition*, Vol. 34, No. 9, pp. 179-181, 2001.
- [56] Дејан Горгевиќ, Комбинирање на класификатори за препознавање на броеви напишани со рака, докторска дисертација, Електротехнички факултет – Скопје, 2004.
- [57] Dejan Gorgevik, Dusan Cakmakov, Vladimir Radevski, Handwritten Digit Recognition by Combining Support Vector Machines Using Rule-Based Reasoning, *Proceedings of the 23th Int. Conference on Information Technology Interfaces ITI 2001*, pp. 139-144, June 19-22, Pula, Croatia, 2001.
- [58] Dejan Gorgevik, Dusan Cakmakov, Vladimir Radevski, Handwritten Digit Recognition Using Statistical and Rule Based Decision Fusion, *Proceedings of 11th Mediterranean Electrotechnical Conference - MELECON '2002*, pp. 131-135, May 7-9, Cairo, Egypt, 2002.
- [59] Dejan Gorgevik, Dusan Cakmakov, Combining SVM Classifiers for Handwritten Digit Recognition, *Proceedings of 16th International Conference on Pattern Recognition ICPR2002*, Vol. 3, SII.8p, Quebec City, Canada, 2002.
- [60] Dejan Gorgevik, Dusan Cakmakov, Cooperation of Support Vector Machines for Handwritten Digit Recognition Trough Partitioning of the Feature Set, in Mile J. Stankovski (eds.), *Proceedings of the Sixth National Conference with International Participation ETAI'2003*, pp. 1102-1108, Ohrid, Macedonia, 17–20 September 2003.
- [61] Dejan Gorgevik, Dusan Cakmakov, An Efficient Three-Stage Classifier for Handwritten Digit Recognition, *Proceedings of 17th Int. Conference on Pattern Recognition, ICPR2004*, Vol. 4, pp. 507-510, IEEE Computer Society, Cambridge, UK, 23-26 August 2004.
- [62] Dejan Gorgevik, Dusan Cakmakov, Methods of Combining SVMs Applied to Handwritten Digit Recognition, in Mile J. Stankovski (eds.), *Proceedings of 7th National Conference with International Participation, ETAI 2005*, pp. 151-155, Ohrid, Macedonia, 21-24 September 2005.
- [63] Dejan Gorgevik, Dusan Cakmakov, Handwritten Digit Recognition by Combining SVM Classifiers, *Proceedings of the Conference Computer as a Tool, EUROCON 2005*, pp. 1393-1396, Belgrade SCG, November 21-24, 2005.
- [64] Dejan Gorgevik, Dusan Cakmakov, Partitioning of the Feature Set for Classifier Cooperation, *Journal of Engineering Creation and Technology*, UDC 62, MK, pp. 27-35, 2006.
- [65] M. Grabisch, F. Dispot, A Comparison of some Fuzzy Classification on Real Data, *Proceedings of the Second International Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, pp. 659-662, 1992.
- [66] Stevan Hadživuković, Rudolf Zegnal, Katarina Čobanović, *Regresiona Analiza*, Privredni pregled, Beograd, 1982.
- [67] Lars K. Hansen, Peter Salamon, Neural Network Ensembles, *IEEE Transactions on PAMI*, Vol. 12, No. 10, pp. 93-1001, 1990.

- [68] Sherif Hashem, Algorithms for Optimal Linear Combinations of Neural Networks, *Proceedings of the International Conference on Neural Networks*, Vol. 1, pp. 242–247, Houston, 1997.
- [69] Trevor Hastie, Robert Tibshirani, Classification by Pairwise Coupling, *Advances in Neural Information Processing Systems*, Vol. 10, M. I. Jordan, M. J. Kearns, S. A. Solla, eds., MIT Press, 1998.
- [70] Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari, A Regression Approach to Combination of Decisions by Multiple Character Recognition Algorithms, *SPIE Proceedings Vol. 1661, Machine Vision Applications in Character Recognition and Industrial Inspection*, San Jose, February 10-12, 1992.
- [71] Tin Kam Ho, The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on PAMI*, Vol. 20, No. 8, 832-844, 1998.
- [72] Tin Kam Ho, Complexity of Classification Problems and Comparative Advantages of Combined Classifiers, *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, in J. Kittler and F. Roli, Eds., Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 1857, pp. 97-106, Springer-Verlag, 2000.
- [73] Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari, Decision Combination in Multiple Classifier Systems, *IEEE Transactions on PAMI*, Vol. 16, No. 1, pp. 66-75, 1994.
- [74] Robert Hooke, T. A. Jeeves, Direct Search Solution of Numerical and Statistical Problems, *Journal of the ACM*, Vol. 8, No. 2, pp. 212-229, April 1961.
- [75] Yu Hen Hu, Jong-Min Park, T. Knoblock, Committee Pattern Classifiers, *Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Proc.*, ICASSP-97, Vol. 4, pp. 3389-3392, Munich, Germany, 1997.
- [76] Y. S. Huang, Ching Y. Suen, A Method of Combining Multiple Classifiers - A Neural Network Approach, *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, Vol. 2, pp. 473-475, 1994.
- [77] Y. S. Huang, Ching Y. Suen, A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals, *IEEE Transactions on PAMI*, Vol. 17, No. 1, pp. 90-94, 1995.
- [78] Branislav Ivanović, *Diskriminaciona analiza sa primenom u ekonomskim istraživanjima*, Naučna Knjiga, Beograd, 1963.
- [79] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive Mixtures of Local Experts, *Neural Computation*, Vol. 3, pp. 79-87, 1991.
- [80] R.A. Jacobs, Methods of Combining Experts' Probability Assessments, *Neural Computation*, Vol. 7, pp. 867-888, 1995.
- [81] Chuanyi Ji, Sheng Ma, Combinations of Weak Classifiers, *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, pp. 32-42, 1997.
- [82] Thorsten Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of the European Conference on Machine Learning (ECML)*, Springer, 1998.
- [83] M.I. Jordan, L. Xu, Convergence Results for the EM Approach to Mixtures of Experts Architectures, *Neural Networks*, Vol. 8, pp. 1409-1431, 1995.
- [84] H.J. Kang, S.W. Lee, A Dependency-based Framework of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals, *Proceedings of the 1999 Int. Conference on Computer Vision and Pattern Recognition – CVPR '99*, pp. 124-129, Fort Collins, Colorado, USA, June 1999.

- [85] H.J. Kang, S.W. Lee, Combining Classifiers Based on Minimization of a Bayes Error Rate, *Proceedings of the 5th Int. Conference on Document Analysis and Recognition ICDAR'99*, pp. 398-401, Bangalore, India, 1999.
- [86] Arthur F. Kaupe Jr., Algorithm 178: Direct Search, *Communications of the ACM*, Vol 6., No. 6, pp. 313-314, June 1963.
- [87] J.M. Keller, P. Gader, H. Tahani, J.-H. Chiang, M. Mohamed, Advances in Fuzzy Integration for Pattern Recognition, *Fuzzy Sets and Systems*, Vol. 65, pp. 273-283, 1994.
- [88] Jörg Kindermann, Edda Leopold, Gerhard Paass, Multi-class Classification with Error Correcting Codes, in E. Leopold, M. Kirsten, editors, Treffen der GI-Fachgruppe 1.1.3, Maschinelles Lernen, GMD Report 114, 2000.
- [89] Josef Kittler, A.Hojjatoleslami, T. Windeatt, Strategies for Combining Classifiers Employing Shared and Distinct Pattern Representations, *Pattern Recognition Letters*, Vol. 18, pp. 1373-1377, 1997.
- [90] Josef Kittler, A Framework for Classifier Fusion - Is It Still Needed, in F. J. Ferri, J. M. Inesta, A. Amin and P. Pudil, Eds., *Advances in Pattern Recognition*, Lecture Notes in Computer Science, Vol. 1876, pp. 45-56, Springer-Verlag, 2000.
- [91] Josef Kittler and Fabio Roli (Eds.), *Multiple Classifier Systems*, Lecture Notes in Computer Science, Vol. 1857, 2000.
- [92] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, Jiri Matas, On Combining Classifiers, *IEEE Transactions on PAMI*, Vol. 20, No. 3, pp. 226-239, 1998.
- [93] Ulrich Kreßel, Pairwise Classification and Support Vector Machines, in B. Schölkopf, C. J. C. Burges, A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pp. 255-268, MIT Press, Cambridge, MA, 1999.
- [94] Ludmila I. Kuncheva, On Combining Multiple Classifiers, *Proceedings of the 7th International Conference on Information Processing and Management of Uncertainty (IPMU'98)*, pp. 1890-1891, Paris, France, 1998.
- [95] Ludmila I., Kuncheva, C. J. Whitaker, C. A. Shipp, R. Duin, Is Independence Good for Combining Classifiers?, *Proceedings of the 15th International Conference on Pattern Recognition*, pp. 168-171, Barcelona, Spain, 2000.
- [96] Ludmila I. Kuncheva, James C. Bezdek, Robert P.W. Duin, Decision Templates for Multiple Classifier Fusion: An Experimental Comparison, *Pattern Recognition*, Vol. 34, pp. 299-314, 2001.
- [97] Ludmila I. Kuncheva, Switching Between Selection and Fusion in Combining Classifiers: An Experiment, *IEEE Transactions on Systems, Man, and Cybernetics Part B*, Vol. 32, No. 2, pp. 146-156, 2002.
- [98] Louisa Lam, Ching Y. Suen, Optimal Combination of Pattern Classifiers, *Pattern Recognition Letters*, Vol. 16, pp. 945-954, 1995.
- [99] G.N. Lance, W.T. Williams, Mixed-data Classificatory Programs, I. Agglomerative Systems, *Australian Computer Journal*, Vol. 1, pp. 15-20, 1967.
- [100] Yoonkyung Lee, Yi Lin, Grace Wahba, Multicategory Support Vector Machines, *Tech. Report 1043*, Dept. of Statistics, University of Wisconsin, 2001.
- [101] Paul R. Lorczak, Alper K. Caglayan, Dave E. Eckhardt, A Theoretical Investigation of Generalized Voters for Redundant Systems, *Proceedings of the Fault Tolerant Computing Systems*, pp. 444-451, Chicago, 1989.
- [102] Brian T. Luke, Distances in Clustering, [<http://fconyx.ncifcrf.gov/~lukeb/clusdis.html>]

- [103] Ofer Matan, On Voting Ensembles of Classifiers, *Proceedings of the National Conference on Artificial Intelligence*, (workshop on Integrating Multiple Learned Models), pp. 84-88, Portland, Oregon, August 1996.
- [104] Christopher J. Merz, M. Pazzani, Combining Neural Network Regression Estimates with Regularized Linear Weights, in M. Mozer, M. Jordan, and T. Petsche, Eds., *Neural Information Processing Systems*, Vol. 9, pp. 564-570, MIT Press, Cambridge, 1997.
- [105] Christopher J. Merz, *Classification and Regression by Combining Models*, PhD thesis, University of California, Irvine, USA, 1998.
- [106] Ryszard S. Michalski, Robert E. Stepp, Edwin Diday, A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts, Invited chapter in the book *Progress in Pattern Recognition*, L. Kanal and A. Rosenfeld (Eds.), Vol. 1, pp. 33-55, North-Holland, 1981.
- [107] D. Michie, D.J. Spiegelhalter, C. Taylor, *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, 1994.
- [108] S.J. Nowlan, G.E. Hinton, Evaluation of Adaptive Mixtures of Competing Experts, in: R.P. Lippmann, J.E. Moody, D.S. Touretzk, Eds., *Advances in Neural Information Processing Systems*, Vol. 3, pp. 774-780, 1991.
- [109] K.C. Ng, B. Abramson, Consensus Diagnosis: a Simulation Study, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, pp. 916-928, 1992.
- [110] L. Orloci, An Agglomerative Method for Classification of Plant Communities, *Journal of Ecology*, No. 55, pp. 193-206, 1967.
- [111] Edgar Osuna, Robert Freund, Federico Girosi, An Improved Training Algorithm for Support Vector Machines, in J. Principe, L. Gile, N. Morgan, and E. Wilson, eds, *Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing VII*, pp. 276-285, New York, 1997.
- [112] Theo Pavlidis, A Review of Algorithms for Shape Analysis, *Computer Graphics and Image Processing*, Vol. 7, pp. 243-258, Academic Press, Inc., 1978.
- [113] John C. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, in *Advances in Kernel Methods: Support Vector Learning*, pp. 61-74, MIT Press, Cambridge, MA, 1999.
- [114] John C. Platt, Nello Cristianini, John Shawe-Taylor, Large Margin DAGs for Multiclass Classification, *Advances in Neural Information Processing Systems*, S.A. Solla, T.K. Leen and K.R. Müller (eds.), Vol. 12, pp. 547-553, MIT Press, 2000.
- [115] E. Rao, A Review of Canonical Coordinates and an Alternative to Correspondence Analysis Using Hellinger Distance, *Qüestiió (Quaderns d'Estadística i Investivació Operativa)*, No. 19, pp. 23-63, 1995.
- [116] Viswanath Ramamurti, Joydeep Ghosh, Structurally Adaptive Modular Networks for Non-Stationary Environments, *IEEE Transactions on Neural Networks*, Vol. 10, No. 1, pp. 152-60, 1999.
- [117] Šarūnas Raudys, Fabio Roli, The Behavior Knowledge Space Fusion Method: Analysis of Generalization Error and Strategies for Performance Improvement, *Proceedings of the 4th Int. Workshop on Multiple Classifier Systems (MCS 2003)*, LNCS 2709, pp. 55-64, Guildford, UK, June 11-13, 2003.
- [118] Šarūnas Raudys, Experts' Boasting in Trainable Fusion Rules, *IEEE Transactions on PAMI*, Vol. 25, No. 9, 1178-1182, 2003.

- [119] Y. Raviv, N. Intrator, Bootstrapping with Noise: An Effective Regularization Technique, *Connection Science, Special issue on Combining Estimators*, Vol. 8, pp. 356-372, 1996.
- [120] Ryan Rifkin, Aldebaro Klautau, In Defense of One-vs-All Classification, Vol. 5, pp. 101-141, January 2004.
- [121] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
- [122] G. Rogova, Combining the Results of Several Neural Network Classifiers, *Neural Networks*, Vol. 7, pp. 777-781, 1994.
- [123] Fabio Roli, Giorgio Giacinto, Design of Multiple Classifier Systems, in H. Bunke and A. Kandel (Eds.) *Hybrid Methods in Pattern Recognition*, World Scientific Publishing, 2002.
- [124] Fabio Roli, Šarūnas Raudys, Gian Luca Marcialis, An Experimental Comparison of Fixed and Trained Fusion Rules for Crisp Classifiers Outputs, *Proc. of the 3rd Int. Workshop on Multiple Classifier Systems (MCS 2002)*, F.Roli and J.Kittler Eds., LNCS 2364, pp. 232-241, Cagliari, Italy, June 2002.
- [125] Robert E. Schapire, The Strength of Weak Learnability, *Machine Learning*, Vol. 5, No. 2, pp. 97-127, 1990.
- [126] M. Schmidt, H. Gish, Speaker Identification Via Support Vector Classifiers, *Proceedings of the International Conference on Acoustics, Speech and Signal Processing '96*, pp. 105-108, IEEE, Atlanta, GA, 1996.
- [127] Bernhard Schölkopf, Christopher J.C. Burges, Vladimir Vapnik, Extracting Support Data for a Given Task, *Proceedings of the First Int. Conference on Knowledge Discovery & Data Mining*, AAAI Press, Menlo Park, CA, 1995.
- [128] Bernhard Schölkopf, Christopher J.C. Burges, Vladimir Vapnik, Incorporating Invariances in Support Vector Learning Machines, in C. *Artificial Neural Networks - ICANN'96*, pp. 47-52, Berlin, 1996.
- [129] Bernhard Schölkopf, A.J. Smola, K.-R. Müller, C. Burges, V. Vapnik, Support Vector Methods in Learning and Feature Extraction, *Ninth Australian Conference on Neural Networks*, pp. 72-78, Brisbane, Australia, 1998.
- [130] Jürgen Schürmann, *Pattern Classification: A Unified View of Statistical and Neural Approaches*, John Wiley & Sons, Inc., 1996.
- [131] Glenn Shafer, *A Mathematical Theory of Evidence*, Princeton University Press Princeton, N.J., 1976.
- [132] Amanda J. C. Sharkey, Multi-Net Systems, *Combining Artificial Neural Nets Ensemble and Modular Multi-Net Systems*, 1-27, Springer-Verlag, 1999.
- [133] Amanda J. C. Sharkey, N. E. Sharkey, U. Gerecke, G. O. Chandroth, The 'test and select' Approach to Ensemble Combination, in Kittler and F. Roli, Eds., *Multiple Classifier Systems*, Lecture Notes in Computer Science, Vol. 1857, 30-44, Springer-Verlag, 2000.
- [134] Catherine A. Shipp, Ludmila I. Kuncheva, Relationships Between Combination Methods and Measures of Diversity in Combining Classifiers, *Information Fusion*, Vol. 3, No. 2, pp. 135-148, 2002.
- [135] Patrick van der Smagt, The bootstrap, [<http://www.robotic.dir.de/Smagt/research/feed-forward/bootstrap/>]
- [136] Robert R. Sokal, F. James Rohlf, *Biometry*, second edition, W. H. Freeman and Company, New York, 1981.

- [137] M. Sugeno, Fuzzy Measures and Fuzzy Integrals: A Survey, *Fuzzy Automata and Decision Processes*, pp. 89-102, North Holland, Amsterdam, 1977.
- [138] Hossein Tahani, James M. Keller, Information Fusion in Computer Vision Using the Fuzzy Integral, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 3, pp. 733-741, 1990.
- [139] Michiaki Taniguchi, Volker Tresp, Averaging Regularized Estimators, *Neural Computation*, Vol. 9, pp. 1163-1178, 1997.
- [140] D.M.J. Tax, R.P.W. Duin, M. van Breukelen, Comparison Between Product and Mean Classifier Combination Rules, in: P. Pudil, J. Novovicova, J. Grim (eds.), *Proceedings of the 1st International Workshop on Statistical Techniques in Pattern Recognition*, pp. 165-170, Czech Republic, Prague, 1997.
- [141] Ljupco Todorovski, Sašo Džeroski, Combining Classifiers with Meta Decision Trees, *Machine Learning Journal*, Vol. 50, pp. 223-249, 2003.
- [142] F. K. Tomlin, L. B. Smith, Remark on Algorithm 178: Direct Search, *Communications of the ACM*, Vol. 12, No. 11, pp. 637-638, 1969.
- [143] Kagan Tumer, Joydeep Ghosh, Order Statistics Combiners for Neural Classifiers, *Proceedings of the World Congress on Neural Networks*, pp. I:31-34, Washington, DC, July 1995.
- [144] Kagan Tumer, Joydeep Ghosh, Classifier Combining: Analytical Results and Implications, *Proc. of the National Conference on Artificial Intelligence, Workshop on Integrating Multiple Learned Models*, Portland, August 1996.
- [145] Kagan Tumer, Joydeep Ghosh, Linear and Order Statistics Combiners for Pattern Classification, A. Sharkey, ed., *Combining Artificial Neural Nets*, pp. 127-162, Springer-Verlag, 1999.
- [146] Naonori Ueda, Optimal Linear Combination of Neural Networks for Improving Classification Performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 2, pp. 207-215, 2000.
- [147] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.
- [148] Vladimir N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1996.
- [149] P. Vasseur, C. Pegard, E. Mouaddib, L. Delahoche, Perceptual Organization Approach Based on Dempster-Shafer Theory, *Pattern Recognition*, Vol. 32, pp. 1449-1462, 1999.
- [150] Antanas Verikas, Arunas Lipnickas, Kerstin Malmqvist, Fuzzy Measures in Neural Networks Fusion, *Proceedings of the 7th Int. Conference on Neural Information Processing, ICONIP-2000*, pp. 1152 -1157, Taejon, Korea, 2000.
- [151] Grace Wahba, Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV, in B. Schölkopf, C. J. C. Burges, A. J. Smola, eds., *Advances in Kernel Methods: Support Vector Learning*, pp. 69-88, MIT Press, Cambridge, MA, 1999.
- [152] Nayer Wanas, Mohamed S. Kamel, Gasser Auda, Fakhreddine Karray, Feature-based Decision Aggregation in Modular Neural Network Classifiers, *Pattern Recognition Letters*, Vol. 20, No. 11-13, pp. 1353-1359, November 1999.
- [153] Nayer Wanas, Mahmoud El-Sakka, Mohamed S. Kamel, Adaptive Aggregation of Multiple Classifiers for Character Recognition, *Cognitive and Neural Models for Word Recognition and Document Processing*, Nabil Murshed Ed., World Scientific Press, 2000.

- [154] Nayer Wanas, Mohamed S. Kamel, Decision Fusion in Neural Network Ensembles, *Proceedings of the Int. Joint Conference on Neural Networks (IJCNN'01)*, Vol 4, pp. 2952-2957, Washington D.C., USA, Jul 15-19, 2001.
- [155] Dayou Wang, James M. Keller, C.A. Carson, Kelly K. McAdoo-Edwards, Craig W. Bailey, Use of Fuzzy-Logic-Inspired Features to Improve Bacterial Recognition Through Classifier Fusion, *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, pp. 583-591, Vol. 28, No. 4, August 1998.
- [156] J. Weston, C. Watkins, Multi-class Support Vector Machines, *Technical Report CSD-TR-98-04*, Royal Holloway, University of London, UK, 1998.
- [157] D. Randall Wilson, Tony R. Martinez, Improved Heterogeneous Distance Functions, *Journal of Artificial Intelligence Research*, Vol. 6, pp. 1-34, 1997.
- [158] David H. Wolpert, William G. Macready, No Free Lunch Theorems for Search, *Tech. Rep. No. SFI-TR-95-02-010*, Santa Fe, NM, 1995.
- [159] David H. Wolpert, William G. Macready, The Mathematics of Search, *Tech. Rep. No. SFI-TR-95-02-010*, Santa Fe, NM, 1995.
- [160] David H. Wolpert, William G. Macready, No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67-82, 1997.
- [161] David H. Wolpert, Stacked Generalization, *Neural Networks*, Vol. 5, No. 2, pp. 241-260, 1992.
- [162] Kevin Woods, W. Philip Kegelmeyer Jr., Kevin Bowyer, Combination of Multiple Classifiers Using Local Accuracy Estimates, *IEEE Transactions on PAMI*, Vol. 19, No. 4, pp. 405-410, 1997.
- [163] Huadong Wu, Mel Siegel, Rainer Stiefelhagen, JieYang, Sensor Fusion Using Dempster-Shafer Theory, *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, Anchorage, AK, USA, 21-23 May 2002.
- [164] Huadong Wu, Mel Siegel, Sevim Ablay, Sensor Fusion Using Dempster-Shafer Theory II: Static Weighting and Kalman Filter-like Dynamic Weighting, *Proceedings of the IEEE annual Instrumentation and Measurement Technology Conference - IMTC*, Vail, CO USA, May 20-22, 2003.
- [165] Lei Xu, Adam Krzyzak, Ching Y. Suen, Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 3, pp. 418-435, 1992.
- [166] Ronald R. Yager, Element Selection from a Fuzzy Subset Using the Fuzzy Integral, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 2, pp. 467-477, 1993.
- [167] Bernard Ženko, Sašo Džeroski, Stacking with an Extended Set of Meta-Level Attributes and MLR, *Proceedings of the 13th European Conference on Machine Learning*, pp. 493-504, Springer, Berlin, Germany, 2002.
- [168] ____, VegAna, Vegetation Edition and Analysis Tools, *Data Operations*, [<http://biodiver.bio.ub.es/vegana/resources/help/ginkgo/DataOperations.htm>].

Додаток

Додаток

Одбрани публикации на авторите

Во оваа глава се дадени 5 публикации на авторите од областа на комбинирање на класификатори. Трудовите се презентирани на домашни и меѓународни конференции и печатени во домашни и меѓународни списанија во последните 4 години.

Сите трудови ги разгледуваат проблемите на комбинирање на класификатори во апликацијата за препознавање на цифри испишани со рака. Класификаторите што се комбинираат се или невронски мрежи или SVM класификатори со гаусов кернел.

Веројатно најважното, би кажале суштинското прашање што се проткајува низ сите изложени трудови наједноставно би го формулирале вака: „Дали и колку се корисни комбинирањата на класификатори во апликациите за препознавање примероци?“. Нашите истражувања покажуваат дека е тешко да се направи систем за препознавање примероци базиран на комбинација на класификатори што ќе има повисоки рати на препознавање од еден добро дизајниран SVM класификатор. Според наше мислење, предностите на комбинирањето на класификатори треба да бараат на друга страна и тоа во намалувањето на комплексноста на класификаторот, редукацијата на бројот на примероци потребни за обука на индивидуалните класификатори (земени заедно и со примероците евентуално потребни за обука на комбинаторот), намалување на потребното време за обука, поголеми можности за воведување најразлични методи за отфрлување на несигурните класификации и по потреба фино подесување на трговијата меѓу брзината на класификаторот и ратите на препознавање. Се разбира, комбинирањето на класификатори е неопходно кога сакаме да обединиме одлуки на веќе обучени класификатори (можно е користење само на фиксен метод), а немаме примероци или не сакаме да обучуваме нов класификатор.

Првиот труд „Combining SVM Classifiers for Handwritten Digit Recognition“ е изложен на 16-тата меѓународна конференција за препознавање примероци (ICPR 2002) што се одржа во Квебек, Канада (референца 59 на страна 144).

Трудот е посветен на комбинирањето класификатори со статистички шеми и шеми базирани на правила, користејќи ги одлуките на индивидуалните класификатори. Суштинскиот придонес на трудот во областа на комбинирањето на класификатори е во споредбата на статистичкото коминирање и комбинирањето со правила, како и споредбата на двата пристапи на комбинирање со „обичниот“ класификатор што го користи целото множество обележја како влез. Множеството обележја е поделено на статистичко и структурно во зависност од тоа како обележјата се екстрахирани. За комбинирањето се користени SVM-и. Добиените резултати покажуваат дека е тешко да се надминат ратата на препознавање на еден добро подесен SVM класификатор креиран над комплетното множество обележја. Комбинирањата на класификаторите базирани на правила се лесни и ефикасни за имплементација при што на многу едноставен начин се додаваат критериуми за отфрлување на несигурните одлуки (rejection criteria). Од друга страна, статистичките комбинирања даваат подобри рати на препознавање и овозможуваат фино подесување на трговијата меѓу ратите на препознавање и доверливоста на системот.

Combining SVM Classifiers for Handwritten Digit Recognition

Abstract

In this paper, we investigate the advantages and weaknesses of various decision fusion schemes using statistical and rule-based reasoning. The cooperation schemes are applied on two SVM (Support Vector Machine) classifiers performing classification task on two feature families referenced as structural and statistical features. The obtained results show that it is difficult to exceed the recognition rate of a single classifier applied straightforwardly on both feature families as one set. The rule based cooperation schemes enable an easy and efficient implementation of various rejection criteria. On the other hand, the statistical cooperation schemes provide higher recognition rates and offer possibility for fine-tuning of the recognition versus the reliability tradeoff.

1. Introduction

Combining features of different nature and the corresponding classifiers has been shown to be a promising approach in many pattern recognition applications. Data from more than one source that are processed separately can often be profitably re-combined to produce more concise, more complete and/or more accurate situation description. In this paper, we discuss classification systems for handwritten digit recognition using two different feature families and SVM classifiers [1]. Our feature families are referenced as structural and statistical feature sets [2], and they differ (especially structural features) from the feature sets with the same reference used in other systems for handwritten character recognition [3][4]. We started with a SVM classifier applied on both feature families as one set. Further, we examined two SVM classifiers that worked separately on the different feature families for the same digit image. We analyzed the possibility of their cooperation using various statistical and rule-based cooperation schemes. In order to improve the system reliability, we introduced rejection criteria as a part of the classifier decision fusion. Our aim was not to compete with the recognition rates of the other handwritten digit recognition systems [5], but to compare the qualities of different feature families, corresponding SVM classifiers and their combination based on different statistical and rule-based decision fusion.

2. The system architecture

The recognition system is constructed around a modular architecture of feature extraction and digit classification units. Preprocessed image is an input for the feature extraction module, which transfers the extracted features toward SVM classifiers (Figure 1). From the digit images with resolution of 128×128 pixels, we have obtained 16×16 binary images on which the smoothing and centralizing preprocessing techniques have been applied. We have extracted 116 features that are classified as 54 structural and 62 statistical. The both feature families as one set are forwarded to the SVM classifier and obtained results are basis for future comparisons.

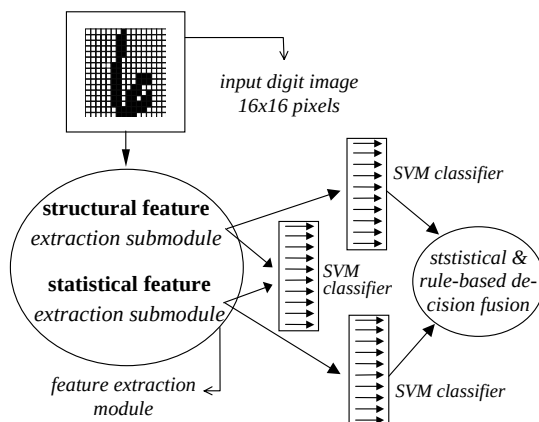


Figure 1. The system architecture

The structural and statistical feature sets are also forwarded to the separate SVM classifiers, and obtained results are combined using statistical and rule-based reasoning. On this level, rejection criteria are introduced and the corresponding system reliabilities are calculated.

3. The data base and feature extraction

The database for our experiments is an extraction of the NIST (National Institute of Standards and Technology) segmented handwritten digit database. The digit images are in 128×128 gray level pixels presented with real numbers in $[-1, 1]$ interval. The total number of 23898 digit images is divided into two groups, 17952 images for the training phase and 5946 images for the test phase. The digits from the original database are rearranged in order that digits in the test set belong to different writers from those in the learning set.

To create the structural feature set we have defined a set of elementary shape primitives for digit constructions. We have proposed 27 elementary primitives showed on Figure 2. The digit image is searched for these primitives twice: firstly on the original digit image orientation, and secondly on the rotated digit image for 90° . So, the total number of primitives is 54, and that is the number of the elements in the structural feature set.

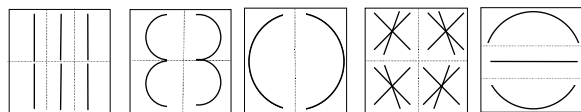


Figure 2. Image sub-regions and elementary primitives

The existing shape in each of those sub-regions is compared with the referent, idealized primitives in the same sub-regions whose existence is expected. The similarity measure between the found shape and the primitive is based on differ-

ences of changes of angles along both shapes, normalized to take values between 0 and 1. This similarity measure is a simplified variation of the curve matching technique described in [6].

The statistical feature set is composed of 62 features that give the pixel-based information in the terms of density of the lit pixels in various digit image regions. The first 54 statistical features are obtained from the projection histograms issued from the vertical (16), horizontal (16) and two diagonal (22) projections (with 5 pixels left and right around the main diagonals). The last 8 features are obtained from the zone-pattern regions showed in Figure 3.

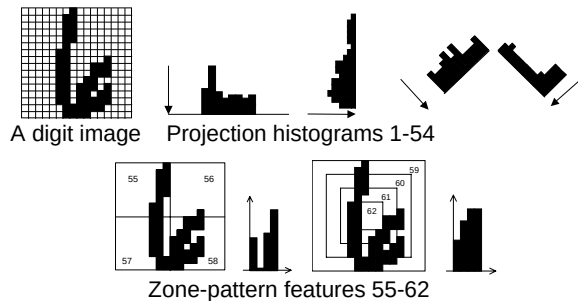


Figure 3. Projection histograms and zone-pattern features

Each of the numerical values of the 62 statistical features represents the filled up percentage of the projection histograms. So, the statistical features have values between 0 and 1.

4. The Recognition Result

We have used a SVM classifier with Gaussian kernel. The outputs of this classifier applied on our samples fall in $[-8, 5]$ interval. Let us denote the classifier outputs in descending order by $O_1, O_2, \dots, O_{10}$ ($O_1 \geq O_2 \geq \dots \geq O_{10}$). The rejection criterion is based on the “top two” classifier outputs. All the samples with highest value of classifier outputs that is smaller than a certain threshold T_1 ($O_1 < T_1$) or for which the difference between the “top two” classifier outputs is smaller than a certain threshold T_2 ($O_1 - O_2 < T_2$) are rejected. Varying these thresholds to obtain reliability of at least 99% we have obtained the results shown in Table 1.

Table 1. Recognition rates on the structural, statistical and both feature families

SVM (Gaussian kernel)	Recog.	T_1, T_2	RRecog.	Miscl.	Rejec.	Reliab.
Statistical features	96.80% (5756)	0.1 0.96	92.23% (5484)	0.92% (55)	6.48% (407)	99.01%
Structural features	94.92% (5644)	0.4 1.5	81.48% (4845)	0.81% (48)	17.71% (1053)	99.02%
Structural + Statistical features	97.53% (5799)	-0.4 0.72	94.80% (5637)	0.96% (57)	4.24% (252)	99.00%

Recog. is the classifier recognition rate. RRecog., Miscl. and Rejec. denote the recognition, misclassification and rejection rates for reliability of at least 99%. Reliab. denotes the reliability that is calculated as $\text{Reliability} = \text{Recognition}/(100\% - \text{Rejection})$. The results show that the statistical feature set has stronger discrimination power and provides better recognition rates. However, the recognition rate on the statistical feature set is more than 0.7 percent lower than the recognition rate of the classifier applied to the complete feature set.

4.1. The Statistical Decision Fusion

The statistical decision fusion is built around two SVM classifiers performing classification separately on the structural and statistical feature sets. In Table 2, the recognition rates using various statistical cooperation schemes are presented. We used the same rejection criterion as in Table 1, and suitable values for T_1 and T_2 were chosen in order to achieve reliability of at least 99%.

The decision fusion methods: Product, Dempster Rule, Fuzzy Integral, and Decision Templates require possibilistic outputs. To map the original output values to $[0, 1]$ interval we used the mapping $1/(1+e^{-x})$.

Table 2. Statistical cooperation schemes and the corresponding recognition rates

#	Cooperation schemes	Recog.	T_1, T_2	RRecog.	Miscl.	Rejec.	Reliab.
1.	Average	97.71%	-0.05, 0.65	94.87%	0.96%	4.17%	99.00%
2.	Product	97.70%	0.15, 0.13	95.16%	0.96%	3.88%	99.00%
3.	Max-Max	97.07%	0.4, 0.87	93.41%	0.94%	5.65%	99.00%
4.	Min-Max	97.29%	-0.3, 0.09	93.47%	0.94%	5.58%	99.00%
5.	Dempster	97.73%	-0.257, 0.025	94.95%	0.96%	4.09%	99.00%
6.	Naive Bayes	96.92%	0.8, 0.865	93.74%	0.94%	5.31%	99.01%
7.	Borda count	96.80%	18, 2	93.12%	0.79%	6.09%	99.16%
8.	Fuzzy Integral	97.07%	0.585, 0.2	93.58%	0.94%	5.48%	99.00%
9.	Decision templates	97.70%	0.88, 0.034	94.69%	0.96%	4.36%	99.00%
10.	Generalized Committee	97.78%	0.514, 0.05	95.34%	0.96%	3.70%	99.00%

In order to make the final decision, first four cooperation schemes use the maximum of the sum, the maximum of the product, the maximum of the maximum and the maximum of the minimum of the corresponding pairs of the classifier outputs [7]. The Dempster rule considers the fuzziness of the classifier votes by giving less confidence to less certain votes [8]. The naive Bayes cooperation scheme uses the confusion matrices of member classifiers to estimate the certainty of the classifier decisions [8]. The Borda count cooperation method is a generalization of the majority vote [9]. The fuzzy integration is based on searching for the maximal grade of agreement between the objective evidence (provided by the sorted classifier out-

puts for class i) and the expectation (the fuzzy measure values of both classifiers) [10]. We have also used one of the decision templates approaches described elsewhere [11]. The generalized committee prediction is based on a weighted combination of the predictions of the member classifiers [12].

A few results in Table 2 deserve attention. The best recognition rates ($>97.7\%$) are obtained by five of the cooperation schemes. Let us note that these results are about 0.2% higher than the recognition rate of the SVM that uses both feature families as one feature set (Table 1). The best recognition rates with reliability of 99% are provided by the schemes 10 (Generalized Committee) and 2 (Product). These results are also noticeably better than the corresponding results shown in Table 1. Generally speaking, the statistical cooperation schemes slightly improved recognition rates and reliabilities in comparison to the classifier that utilizes simple integration of the both feature families in one feature set.

4.2. The Rule-Based Decision Fusion

Let us denote by $a1$, $a2$ and $a3$ the first, the second and the third choice of the structural feature classifier, and by $b1$, $b2$ and $b3$ the first, the second and the third choice of the statistical feature classifier for a given pattern. Our experiments showed that the inclusion of additional choices (after the third) provides insignificant recognition rate improvement. The results of classifier outputs based on various rule-based cooperation schemes are evaluated and given in **Error! Reference source not found.**

Table 3. Various rule-based cooperation schemes and the corresponding recognition rates

#	Rule-based cooperation schemes	Recog.	Miscl.	Rejec.	Reliab.
1.	if $a1=b1$ then $c=a1$ else REJECT	93.12%	0.79%	6.09%	99.16%
2.	if $a1=b1$ or $a1=b2$ then $c=a1$ else REJECT	94.48%	2.56%	2.96%	97.37%
3.	if $b1=a1$ or $b1=a2$ then $c=b1$ else REJECT	95.80%	1.55%	2.66%	98.41%
4.	if $a1=b1$ or $a1=b2$ then $c=a1$ elseif $b1=a2$ then $c=b1$ else REJECT	95.90%	2.79%	1.31%	97.17%
5.	if $a1=b1$ or $a1=b2$ or $a1=b3$ then $c=a1$ else REJECT	94.77%	3.50%	1.73%	96.44%
6.	if $b1=a1$ or $b1=a2$ or $b1=a3$ then $c=b1$ else REJECT	96.45%	2.05%	1.50%	97.92%
7.	if $a1=b1$ or $a1=b2$ or $a1=b3$ then $c=a1$ elseif $b1=a2$ or $b1=a3$ then $c=b1$ else REJECT	95.88%	3.73%	0.39%	96.25%
8.	if $b1=a1$ or $b1=a2$ or $b1=a3$ then $c=b1$ elseif $a1=b2$ or $a1=b3$ then $c=a1$ else REJECT	97.24%	2.37%	0.39%	97.62%

Four results in Table 3 deserve attention. Best reliability is obtained by rule 1 (consensus) but the recognition rate is relatively weak. A good compromise is provided by rules 3 and 6, where we choose the first decision $b1$ of the statistical feature classifier as a final decision c , if it is among the “top two” decisions ($a1$, $a2$) in the rule 3 and among the “top three” decisions ($a1$, $a2$, $a3$) in the rule 6 of the structural feature classifier. It seems that in this case the structural feature classifier gives a safety rule for the right decision. The reliabilities of 98.41% and 97.92% by recognition rates of 95.80% and 96.45% are noticeable results, better than some previous attempts using the same feature sets [2].

On the other hand, best recognition rate is provided by the relatively complex rule 8. Unfortunately, this rule produces high misclassification rate that results in lower reliability. Let us notice that the recognition rates achieved by rule-based cooperation schemes is still about 0.3% lower than the recognition rate of the SVM that uses both feature families as one feature set (Table 1). They are also noticeably lower than the recognition rates of the statistical cooperation schemes (Table 2).

5. Conclusion

In this paper, we addressed some issues in designing high reliability system for hand-written digit recognition using SVM classifiers. We used two different feature families referenced as structural and statistical features. Decision level fusion is performed using statistical and rule-based cooperation schemes. To improve the system reliability, we introduced rejection criteria in our decision fusion schemes.

The presented results show that it is difficult to achieve the recognition rate of a single classifier applied on the feature set that includes both feature families by combining the individual classifier decisions using statistical or rule based decision fusion. However, the statistical cooperation schemes slightly improve recognition rates and offer possibility for fine-tuning of the recognition versus the reliability tradeoff. The rule-based cooperation schemes enable an easy implementation of rejection criteria. Additionally, the cooperation of separate classifiers designed for separate feature families reduces classifier complexity and offers better possibilities to understand the role of the features in the recognition process.

References

- [1] Burges C., A Tutorial on Support Vector Machines for Pattern Recognition, *Knowledge Discovery and Data Mining*, Vol. 2, 1998, pp. 1-47.
- [2] Radevski V. and Bennani Y., Reliability Control in Committee Classifier Environment, *Int. Joint Conference on Neural Networks, IJCNN 11*, Como, Italy, Vol. III, 2000, pp. 561-565.
- [3] Duerr B., Haettich W., Tropf H., and Winkler G., A Combination of Statistical and Syntactical Pattern Recognition Applied to Classification of Unconstrained Handwritten Numerals, *Pattern Recognition*, Vol. 12, 1980, pp. 189-199.
- [4] Heutte L., Moreau J. V., Paquet T., Lecourtier Y., and Olivier C., Combining Structural and Statistical Features for the Recognition of Handwritten Characters, *Proc. of the 13th Int. Conf. on Pattern Recognition*, 1996, pp. B74.4.

- [5] LeCun Y., Jackel L. D., Bottou L., Brunot A., Cortes C., Denker J. S., Drucker H., Guyon I., Muller U. A., Sackinger E., Simard P., and Vapnik V., Comparison of Learning Algorithms for Handwritten Digit Recognition, In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, Paris, 1995, pp. 53-60.
- [6] Cakmakov D., Curve Matching Using Turning Functions, *Proc. of the Int. Conf. on Signal and Image Processing SIP'98*, Las Vegas, USA, 1998, pp. 588-592.
- [7] Kittler J., Hatef M., Duin, R.P.W., and Matas J., On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, March 1998, pp. 226-239.
- [8] Xu, L., Krzyzak, A., Suen, C.Y., Methods of Combining Multiple Classifiers and Their Application to Handwritten Recognition, *IEEE Transactions on System, Man and Cybernetics*, Vol. 22, 1992, pp. 418-435.
- [9] Ho, T.K., Hull, J.J., Srihari, S.N., Decision Combination in Multiple Classifier Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1, January 1994, pp. 66-75.
- [10] Cho, S.B., Kim, J.H., Combining Multiple Neural Networks by Fuzzy Integral and Robust Classification, *IEEE Transactions on System, Man and Cybernetics*, Vol. 20, No. 3, 1995, pp. 380-384.
- [11] Kuncheva, L.I., Bezdek, J.C., Duin, P.W., Decision Templates for Multiple Classifier Fusion: an Experimental Comparison, *Pattern Recognition*, Vol. 34, No. 2, 2001, pp. 299-314.
- [12] Bishop C.M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995, pp. 364-369.

Следниот труд со наслов „Decision Fusion and Reliability Control in Handwritten Digit Recognition System“ е печатен во меѓународното научно списание за компјутерска и информациона технологија (International Journal of Computer and Information Technology), во Загреб, Хрватска (референца 25 на страна 142).

Трудот е посветен на постапна изградба на комбинатор на класификатори со висока доверливост кој препознава броеви испишани со рака. Од повеќето шеми на комбинирање што беа испитани, како најдобра се покажа обопштениот комитет (Generalized Committee) со софистициран критериум за отфрлање на несигурните одлуки. Доверливоста на системот е зголемена со користење на двонивоска класификација, користејќи најпрво комбинирање со правила со строг критериум за отфрлање на несигурните одлуки. Потоа, отфрлените примероци се проследуваат до статистички класификатор кој е способен да ги класифицира овие примероци. Добиените резултати покажуваат дека предложениот систем за препознавање на броеви испишани со рака прави успешна контрола на доверливоста на системот при што истовремено нуди високи рати на препознавање. Тоа покажува дека комбинацијата на правила со статистички методи на комбинирање на класификатори е ветувачки пристап за креирање на системи за препознавање примероци што истовремено ги нудат двете клучни работи, висока доверливост и рата на препознавање.

Decision Fusion and Reliability Control in Handwritten Digit Recognition System

Abstract

In this paper, the cooperation of two feature families for handwritten digit recognition using a committee of neural network (NN) classifiers will be examined. Various cooperation schemes will be investigated and corresponding results will be presented. To improve the system reliability, we will upgrade the committee scheme using multistage classification based on rule-based and statistical cooperation. The rule-based cooperation enables an easy and efficient implementation of various rejection criteria while the statistical cooperation offers better possibility for fine-tuning of the recognition versus the reliability tradeoff. The final system has been implemented using rule-based reasoning with rejection criteria for classifier decision fusion and the generalized committee cooperation scheme for classification of the rejected digit patterns. The presented results show that we propose a successful approach for reliability control in committee classifier environment and indicate that a suitable cooperation of statistical and rule-based decision fusion is a promising approach in handwritten recognition systems.

1. Introduction

The idea of combining classifiers in order to compensate their individual weakness and to preserve their individual strength has been widely used in recent pattern recognition applications (Xu et al. 1992), (Ho et al. 1994), (Kittler, 1998), (Duin and Tax 2000), (Roli et al. 2001). Data from more than one source that are separately processed, can often be profitably re-combined to produce more concise, more complete and/or more accurate situation description. A theoretical and mathematical framework that explains the reasons for expecting the improvement of the performances in cases of combining classifier outputs can be found in (Kittler 1988), (Tumer and Ghosh 1999).

The classical paradigm for character recognition is concentrated around two steps, feature extraction, where an appropriate representation of the pattern is developed, and classification, where decision rules for separating pattern classes are defined. Combining features of different nature and the corresponding classifiers has been shown to be a promising approach in handwritten recognition systems (Kimura and Shridar 1991), (Huang and Suen 1993), (Suzuki et al. 1995), (Yamaguchi et al. 1997), (Dahmen et al. 2001).

In this paper, we present a two-stage classification system for handwritten digit recognition using a committee of NN classifiers. We start with two NN classifiers which work on two different feature families for the same digit image. Our feature families are referenced as structural and statistical feature sets (Radevski and Bennani 2000), and they differ (especially structural features) from the feature sets with the same reference used in other handwritten recognition systems (Duer et al. 1980), (Heute et al. 1996), (Lou et al. 1999).

In order to improve the system reliability, we introduced two-stage classification based on rule-based and statistical cooperation. The rule-based cooperation enabled an easy and efficient implementation of rejection criteria. The statistical cooperation offers better possibility for fine-tuning of the recognition versus the reliability tradeoff. The final system has been implemented using rule-based reasoning with rejection criteria for the classifier cooperation and the generalized committee cooperation scheme for classification of the rejected digit patterns. Our goals in this paper are to examine usefulness of our feature extraction and selection technique, to show an approach for reliability control in committee classifier environment and to present a two stage classifier cooperation using rule-based and statistical decision fusion rather than to compete with the recognition rates of other handwritten digit recognition systems (LeCun et al. 1995).

2. The System Architecture

The recognition system is constructed around a modular architecture of feature extraction and digit classification units. The preprocessed image is input for the feature extraction module, which transfers the extracted features toward NN classifiers (Fig. 1).

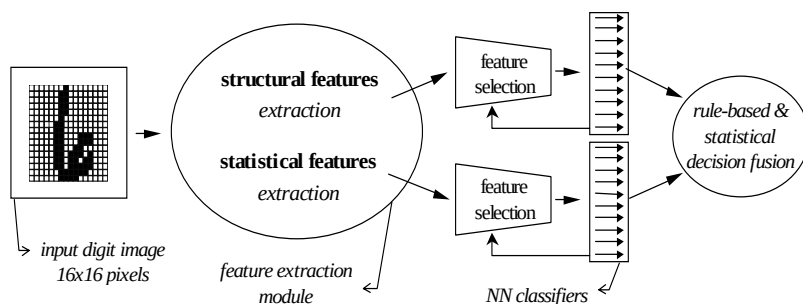


Figure 1. The system architecture

From the digit images with resolution of 128×128 pixels, we obtained 16×16 binary images on which the smoothing and centralizing preprocessing techniques have been applied. We have extracted 54 structural and 62 statistical features. The structural and statistical feature sets are forwarded to two separate NN classifiers where the feature selection procedure is performed. NN classifier outputs are combined using rule-based and statistical cooperation. On this level, rejection criteria are introduced and the corresponding system reliabilities are calculated.

2.1. The Structural Features

The structural feature set is a domain dependent set. Its nature and the techniques implemented for detection and extraction are strongly dependent of the nature of the objects to be recognized.

The first step in the creating of the structural feature set is defining a reasonable set of elementary shape primitives for digit constructions. We have proposed 27 elementary primitives showed in Fig. 3. The digit image is searched for these primitives twice: firstly on the original digit image orientation, and secondly on the rotated digit image for 90°. So, the total number of primitives is 54, and that is the number of the elements in the structural feature set.

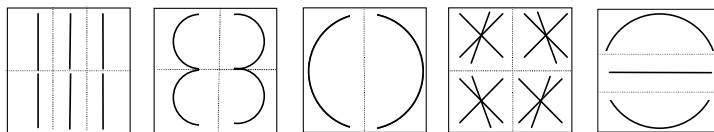


Figure 3. Image sub-regions and the elementary primitives

The detection and the extraction of the structural features is performed by dividing the image binary matrix into two, three, four and six sub-regions. The existing shape in each of those sub-regions is compared with the proposed primitives in the same sub-regions whose existence is expected.

We have made the search for the primitives with parallel lines from up, down, left or right side, depending on their position. In this way, we have obtained control points for the shape description (Fig. 4a). The shape is represented by the line segments lengths s_i (between the control points), and the corresponding angles between the line segments and the x -axis (Fig. 4b).



Figure 4. a) Line control points; b) Line representation: segments and angles

The similarity measure between the found shape and the corresponding predefined primitive is based on the differences between the changes of angles along the shapes, normalized to take value between 0 and 1. This similarity measure is a simplified variation of the curve matching technique described in (Cakmakov 1998).

Thus, the structural feature is composed of 54 values of the calculated similarities between the found shapes in the corresponding sub-regions and the corresponding elementary primitives.

2.2. The Statistical Features

The statistical feature set is composed of 62 features that give the pixel-based information presented by the densities of the lit pixels in various regions of the digit image. The first 54 statistical features are obtained from the projection histograms

obtained by the vertical (16), horizontal (16) and two diagonal (22) projections (5 pixels left and right around the main diagonals). The last 8 features are obtained from the zone-pattern regions showed in Fig. 5.

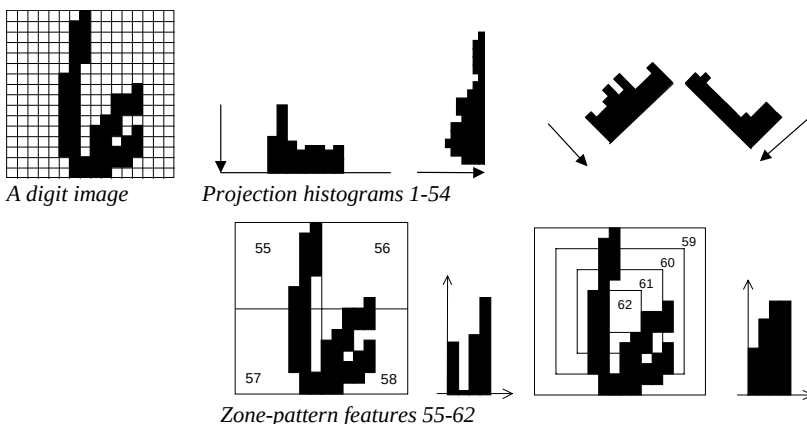


Figure 5. Projection histograms and zone-pattern features

Each of the numerical values of the 62 statistical features represents the filled up percentage of the projection histograms. So, the statistical features have values between 0 and 1. This kind of features in different forms has been exploited in many pattern recognition systems. Burel and colleagues called them oriented profiles (the projection histograms in our case), and statistical features (the zone-pattern features in our case) (Burel et al. 1992).

2.3. Feature Selection

In order to optimize member classifiers we included a feature selection phase. Various feature selection techniques have been proposed (Liu and Motoda 1998). We have implemented a supervised feature selection based on the Optimal Cell Damage technique (Cibas et al. 1994). The features are selected one by one, according to the error rate of the classifier. The selection phase is deduced to deleting of the input NN nodes, after establishing a threshold value for their saliency. Thus, the feature selection phase prunes the input NN nodes with lower importance. This procedure is performed separately on the member classifiers. This reduces the system complexity at least 15 to 20%, keeping the same recognition rate.

2.4. The Handwritten Digit Data Base

The database for our experiments is an extraction of the NIST (National Institute of Standards and Technology) handwritten digit database. The digit images are consisted of gray level pixels presented with real numbers in $[-1, 1]$ interval. The total number of 23898 digit images is divided into two groups, 17952 images for the training phase and 5946 images for the test phase.

The digits from the original database are rearranged in order that digits in the test set belong to different writers from those in the learning set.

In Fig. 5, a fragment and the composition of the digit database are presented.

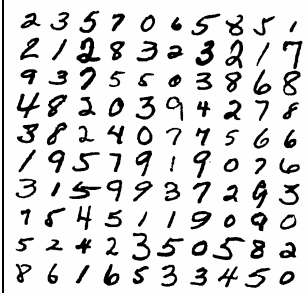
	Class	Learning Set : 17952 samples	Test Set : 5946 samples
	0	1860 (10.36 %)	606 (10.19 %)
	1	2026 (11.29 %)	670 (11.23 %)
	2	1750 (9.75 %)	594 (9.99 %)
	3	1895 (10.56 %)	622 (10.46 %)
	4	1714 (9.55 %)	556 (9.35 %)
	5	1535 (8.55 %)	515 (8.66 %)
	6	1726 (9.61 %)	591 (9.94 %)
	7	1878 (10.46 %)	613 (10.31 %)
	8	1783 (9.93 %)	589 (9.91 %)
	9	1785 (9.94 %)	590 (9.92 %)

Figure 5. A fragment and the composition of the digit database

The class distribution of the samples in the learning and the test set is nearly uniform (Fig. 5).

3. Committee cooperation schemes

The cooperation schemes are designed around two full connected Multi-layer Perceptron (MLP) NNs with one hidden layer. They perform the classification task on the sets of structural and statistical features respectively. The size of their input layers corresponds to the cardinality of the corresponding feature set, and the number of hidden layer nodes is determined experimentally (Fig. 6). Each output node corresponds to one digit. The combined classification unit is based on different cooperation schemes.

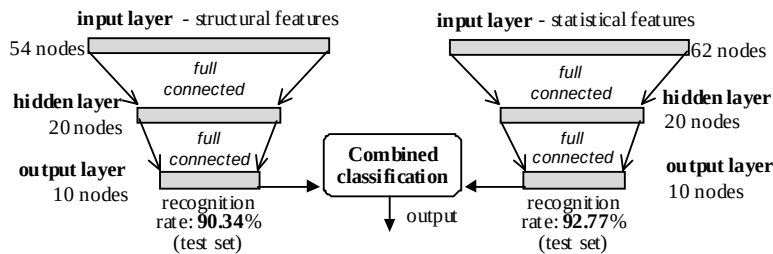


Figure 6. The classification module

Let us $y_i(x)$, $i=1,2$ be the output for input feature vector x of two MLPs for structural and statistical features respectively. Using the individual classifier information, we have examined a few cooperation schemes (Barabas 1983), (Xu et al. 1992), (Ho et al. 1994), (Bishop 1995), (Kittler et al. 1998):

– Simple average

This simplest cooperation is based on the average of the rescaled outputs from the individual classifiers. In our case, we have

$$y_{ave}(\mathbf{x}) = \frac{1}{2} \sum_{k=1}^2 y_k(\mathbf{x}) .$$

This approach does not take into consideration the measure of the belief of the each of the member decisions.

– Classification task

In this cooperation, we consider the outputs of the individual classifiers as a data to be classified by another MLP NN, full connected with one hidden layer. The input layer contains 20 units, 10 from the output of each individual classifier. The hidden layer contains 15 units and the output layers 10 units.

– Dempster's rule

Here, the probability of the favoring of the class C_i is computed as a product of the individual classifier outputs $y_k(\mathbf{x} \in C_i)$

$$y_{demp}(\mathbf{x} \in C_i) = \frac{1}{\eta} \prod_{k=1}^2 y_k(\mathbf{x} \in C_i)$$

where normalization η is calculated by the formula

$$\eta = \sum_{i=1}^{10} \prod_{k=1}^2 y_k(\mathbf{x} \in C_i) .$$

This cooperation takes into account the fuzziness of the classifier votes, by giving less confidence to less certain votes.

– Bayesian formalism

Bayesian Formalism uses the error of each classifier, represented by the corresponding confusion matrices $[n_{i,j}^1]$ and $[n_{i,j}^2]$. The element $n_{i,j}^k$ means that the elements of the class C_i have been assigned to the class C_j by the classifier k . If we consider the confusion matrices as a sources of prior knowledge, they can be used for estimation of the certainty for each classifier. With this knowledge, the conditional probability that the proposition $\mathbf{x} \in C_j$, $i=1,10$ are true under the decision $D_k(\mathbf{x})=C_j$, $k=1,2$ made by each of the classifiers is calculated by

$$P(\mathbf{x} \in C_i / D_k(\mathbf{x}) = C_j) = \frac{n_{i,j}^k}{\sum_{i=1}^{10} n_{i,j}^k} .$$

So, the final classification is made by the formula

$$P_{bayes}(\mathbf{x} \in C_i) = \frac{1}{\eta} \prod_{k=1}^2 P(\mathbf{x} \in C_i / D_k(\mathbf{x}) = C_j) ,$$

where normalization η is calculated by

$$\eta = \sum_{i=1}^{10} \prod_{k=1}^2 P(\mathbf{x} \in C_i / D_k(\mathbf{x}) = C_j) .$$

– Generalized committee cooperation scheme

Generalized committee prediction is given by a weighted combination of the predictions of the member classifiers

$$y_{gen}(\mathbf{x}) = \sum_{i=1}^2 \gamma_i y_i(\mathbf{x}) .$$

For the finite-sample approximation of the correlation matrix M , we can obtain the solutions for the parameters γ_i

$$M_{i,j} \approx \frac{1}{N} \sum_{n=1}^N (y_i(\mathbf{x}_n) - t_n)(y_j(\mathbf{x}_n) - t_n) \text{ and } \gamma_i = \frac{\sum_{j=1}^2 (M^{-1})_{i,j}}{\sum_{k=1}^2 \sum_{j=1}^2 (M^{-1})_{k,j}} ,$$

where t_n is the target value corresponding to the input vector $\mathbf{x}_n$, and N is the number of examples in the training set. Following this procedure we have obtained $\gamma_1 = 0.37$ and $\gamma_2 = 0.63$. So, the committee act is performed according to

$$y_{gen}(\mathbf{x}) = 0.37y_1(\mathbf{x}) + 0.63y_2(\mathbf{x}) .$$

The recognition results of the cooperation of two classifiers according to the described cooperation schemes are given in Table 1.

Committee decision rule	Recognition rate Test set (%)
Simple average	94.97
Classification task	95.22
Dempster's rule	94.87
Bayesian formalism	95.04
Generalized committee	95.09

Table 1. The results of the classifier cooperation

The cooperation results presented in Table 1 show that the non-linear combination of the classifier outputs given by the classification task and the generalized committee provide better recognition results. However, we find that the misclassification rate of approximately 5% (for recognition rate of 95%) is not enough satisfactory.

To design a recognition system with higher reliability, we will study the possibilities of introducing more sophisticated multistage cooperation including the implementation of rejection criteria (Radevski and Bennani 2000).

4. On the individual classification units

In order to establish more effective cooperation scheme, we have studied the properties of the individual classifiers.

4.1. Analysis of classifier outputs

The recognition rates of the NN classifiers and the recognition rates of the “top two” cases (the correct decision is the first or the second classifier choice) are evaluated and given in Table 2.

Classification module	Recognition rate Test set (%)	“top two” recognition rate Test set (%)
structural features	90.34	95.52
statistical features	92.77	97.00

Table 2. Recognition rates of the individual classifiers

The high recognition rates in the “top two” cases show that the number of right answers among the first two choices of the classifiers is significant, and an effort to use these answers more appropriately is a reasonable one.

Let us denote by $a1$ and $a2$ the top two choices (the first and the second respectively) of the structural feature classifier, and by $b1$ and $b2$ the top two choices of the statistical feature classifier. Some relations between the recognition outputs of the classifiers are given in Table 6.

EVENT	Recognition rate Test set (%)
$a1$ and $b1$	86.26
$a1$ or $b1$	96.86
$a1$ or $a2$ or $b1$ or $b2$	98.77

Table 6. Some relations between classifier decisions

These observations confirm that the “top two” classifier outputs offer significant number of right answers, and an appropriate cooperation could lead to improvement of the recognition and reliability rates of the system.

4.2. Rejection criteria

To improve reliability of the system we will use rejection criteria as a part of the cooperation scheme. Let us denote by A_1, A_2 the best two outputs of the first and by B_1, B_2 the best two outputs of the second classifier (let us note that A_1, A_2, B_1, B_2 are classifier outputs, not class labels). Then, simplest rejection criteria for the individual classifiers can be defined by

$$O1(x) = \begin{cases} a1 = c(A_1), & \text{if } |A_1 - A_2| \geq \alpha \\ M + 1, & \text{otherwise} \end{cases},$$

$$O2(x) = \begin{cases} b1 = c(B_1), & \text{if } |B_1 - B_2| \geq \alpha \\ M + 1, & \text{otherwise} \end{cases}$$

where $c()$ is the function which gives the corresponding class for the classifier outputs, M is the number of classes and $M+1$ stays for the additional, rejection class. The parameter α , $0 \leq \alpha \leq 1$, is a threshold parameter. It controls the rejection rate according to the information for the certainty of the choice of the classes $a1$ and $b1$. The results of classifier outputs (Recognition, Misclassification, Rejection and Reliability = Recognition/(100%–Rejection)) using these rejection criteria are shown in Table 7.

α	Structural features				Statistical features			
	Recog. %	Miscl. %	Rejec. %	Reliab. %	Recog. %	Miscl. %	Rejec. %	Reliab. %
0	90.34	9.66	0	90.34	92.77	7.23	0	92.77
2×10^{-5}	88.43	7.00	4.57	92.66	91.49	5.45	3.06	94.37
4×10^{-5}	86.24	5.25	8.51	94.26	89.77	4.31	5.92	95.42
6×10^{-5}	83.88	3.85	11.08	94.33	87.47	3.53	9.00	96.12
8×10^{-5}	80.86	2.99	16.15	96.43	79.77	4.09	16.14	95.12

Table 7. The classifier decisions using rejection criteria

Imposing stronger rejection criteria by increasing the value of the parameter α , the misclassification rate decreases while the rejection rate increases. The reliability of the system also increases until some specific value of the parameter α .

However, the results show that such rejection criterion is not a promising way to place a part of misclassified into the set of rejected patterns. Imposing a threshold level for accepting/refusing a given decision will not only discard a lot of

misclassified digits, but also will discard a significant part of the well-recognized digits. These observations suggest that to achieve improved rejection criteria it should be more useful to consider the activity values of all NN output nodes, or alternatively, to consider an integration of the activity values of two NN output nodes that “won the competition”. Considering the good recognition rates of the classifiers in cases of the “top two” classifier outputs, we implemented a combination of both approaches.

We have investigated the values of the NN output nodes that give the first and the second decision label in each of the classifiers. In Table 5, we show the means $\mu_{1,k}$, $\mu_{2,k}$, $k = 1, 2$ and the standard deviations $\sigma_{1,k}$, $\sigma_{2,k}$, $k = 1, 2$ for the differences between the values of the NN output nodes that gave the first and the second decision label ($A_1 - A_2$, $B_1 - B_2$) in cases of specific events.

Classification module	Event	Mean ($\mu_{i,k}$)	St. deviation ($\sigma_{i,k}$)
Structural Features	1 st decision is the right one	$\mu_{1,1} = 1.4908$	$\sigma_{1,1} = 0.5237$
	2 nd decision is the right one	$\mu_{1,2} = 0.0249$	$\sigma_{1,2} = 0.1455$
Statistical Features	1 st decision is the right one	$\mu_{2,1} = 1.6179$	$\sigma_{2,1} = 0.4669$
	2 nd decision is the right one	$\mu_{2,2} = 0.0258$	$\sigma_{2,2} = 0.1551$

Table 5. Some statistic values of classifier outputs

This information, obtained on the trained classifiers, will be used in our rule-based cooperation schemes and in the corresponding rejection criteria incorporated in these schemes.

5. Multistage classification and final result

The main aim of the first decision fusion phase is to classify as much as possible input patterns, keeping a low misclassification rate. So, the result of this phase will be a set of correctly classified patterns and a set of “hard” patterns, rejected from this phase of classification. To complete this task we will use the information provided by the member classifiers in the following form:

- The “top two” class labels of each of the classifiers: a_1 and a_2 for the structural, and b_1 and b_2 for the statistical feature classifier;
- The differences between the “top two” classifier outputs $d_1 = A_1 - A_2$, $d_2 = B_1 - B_2$ and the intervals $[\mu_{i,k} - s \cdot \sigma_{i,k}, \mu_{i,k} + s \cdot \sigma_{i,k}]$, $i, k = 1, 2$, where these values belong. The parameter s is a suitable constant.

These values allow introduction of an additional certainty measure for the reliability of the classifier outputs. Thus, along with the first and the second decisions of the classifiers, we will consider the intervals where the differences between the corresponding classifier outputs belong (see Table 6).

Classification module	Notation
Structural Features	$f1$ if $d_1 \in [\mu_{1,1} - s \cdot \sigma_{1,1}, \mu_{1,1} + s \cdot \sigma_{1,1}]$
	$f2$ if $d_1 \in [\mu_{1,2} - s \cdot \sigma_{1,2}, \mu_{1,2} + s \cdot \sigma_{1,2}]$
Statistical Features	$g1$ if $d_2 \in [\mu_{2,1} - s \cdot \sigma_{2,1}, \mu_{2,1} + s \cdot \sigma_{2,1}]$
	$g2$ if $d_2 \in [\mu_{2,2} - s \cdot \sigma_{2,2}, \mu_{2,2} + s \cdot \sigma_{2,2}]$

Table 6. Notation of the characteristic events

The events $f1$, $f2$, $g1$ and $g2$ are occurred, when the difference between the “top two” classifier outputs falls near to the averages of the same differences for the corresponding classifier decisions $a1$, $a2$, $b1$ and $b2$. These averages are given in Table 5 and they are obtained on the trained classifiers. Thus, we have additional certainty that the corresponding classifier decision $a1$, $a2$, $b1$ or $b2$ is correct.

Using the above information, we introduced the rule-based decision procedure for this phase of the classification process. We investigated some rule-based decision schemes in order to obtain a low misclassification rate with as high as possible recognition rate. In Table 7, some of the examined rule-based scheme with low misclassification rates and the corresponding results are presented.

#	Rule	s	Recog. %	Misc. %	Rejec. %	Reliab. %
1.	if $a1 = b1$ then $c \leftarrow b1$ elseif ($g2$ and $f1$) and $b2 = a1$ then $c \leftarrow b2$ elseif ($f2$ and $g1$) and $a2 = b1$ then $c \leftarrow a2$ else REJECT	$s=0.2$	86.33	1.5	12.18	98.30
2.	if $g1$ or $a1 = b1$ then $c \leftarrow b1$ elseif $f1$ then $c \leftarrow a1$ elseif ($g2$ or $f1$) and $b2 = a1$ then $c \leftarrow b2$ elseif ($f2$ or $g1$) and $a2 = b1$ then $c \leftarrow a2$ else REJECT	$s=2$ $s=0.2$	94.13 88.26	4.76 2.05	1.11 9.69	95.19 97.73
3.	if $g1$ and $a1 = b1$ then $c \leftarrow b1$ elseif $f1$ then $c \leftarrow a1$ elseif $g2$ and $b2 = a1$ then $c \leftarrow b2$ elseif $f2$ and $a2 = b1$ then $c \leftarrow a2$ else REJECT	$s=2$ $s=1$	91.19 85.97	5.21 2.24	3.60 11.79	97.46 94.78
4.	if $g1$ or $a1 = b1$ then $c \leftarrow b1$ elseif $f1$ then $c \leftarrow a1$ elseif $g2$ or $b2 = a1$ then $c \leftarrow b2$ elseif $f2$ or $a2 = b1$ then $c \leftarrow a2$ else REJECT	$s=2$ $s=1$	94.30 93.90	5.70 5.01	0.0 1.09	94.30 94.93

Table 7. Various rule-based strategies and the corresponding recognition results

There is no a general guideline how to chose the value of the parameter s , as well as, how to design the “best” rule-based decision scheme. These choices of the pa-

parameter s and the rule-based decision scheme follow the idea of minimizing the classifier misclassification rate.

Our experiments show that the lowest misclassification rate is provided by the rule 1 in Table 7. According to this rule, if the member classifiers give the same class as the first choice, we take that decision as final. Otherwise, we take the second choice label of better individual classifier (in our case the statistical feature classifier) $b2$ only if it gives the same decision label as the first choice of the structural feature classifier ($b2 = a1$), and both classifiers are enough sure in their decisions ($g2$ and $f1$). Finally, we take the second choice $a2$ of the structural feature classifier if it gives the same decision label as the first choice of the statistical feature classifier ($a2 = b1$), and both classifiers are enough sure in their decisions ($f2$ and $g1$).

Using a strength rule-based cooperation in the first phase of the classification, we have minimized the misclassification rate of our cooperation scheme. In the second phase, we have used a statistical cooperation scheme on the reduced set of “hard” patterns that have been rejected during the rule-based classification. For this phase of classification we have examined some statistical cooperation schemes. Best results were provided by the generalized committee cooperation scheme (Bishop 1995), described in section 3.

Thus, in the second decision fusion phase, we used the generalized committee cooperation scheme to classify rejected digit patterns. This combination of two different decision fusion schemes resulted in a high reliability recognition system.

In Table 8, the recognition results of the final classification are presented.

System	Recog. %	Misc. %	Rejec. %	Reliab. %
Rule 1 (Table 7) + generalized committee	95.10	1.64	3.26	98.30

Table 8. *The final classification*

The results of the final classification show that the primary goal of the research is achieved. We have obtained a high reliability rate for the recognition system that also keeps a high recognition rate.

6. Conclusion

In this paper, we discussed the high reliability system for hand-written digit recognition. We have shown a possibility for cooperation of classifiers based on different feature families using committee classifiers. For each of the feature families a simple MLP NN has been designed. We examined different cooperation schemes. The results show that the non-linear cooperation schemes give better recognition results.

To design high reliability recognition system we introduced a multistage cooperation scheme with a rejection criterion. In the first decision fusion phase, we used rule-based cooperation scheme in order to provide as high as possible recognition rate, keeping as low as possible misclassification rate. In the second decision fusion

phase, we used the generalized committee cooperation scheme only on the set of rejected pattern. The results of the final classification show that we proposed a successful approach for reliability control in committee classifier environment. Additionally, the results indicate that a suitable cooperation of statistical and rule-based decision schemes is a promising approach in handwritten recognition systems.

7. References

- [1] Y. L. Barabas, *Kollektivnie statisticheskie resenia pri raspoznavanii*, Moskva, Radio i svjaz, (*Collective statistical solutions for recognition*, in Russian), 1983.
- [2] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.
- [3] G. Burel, I. Pottier, J. Y. Catros, Recognition of Handwritten Digits by Image Processing and Neural Network, *Proc. of the Int. Joint Conference on Neural Networks, IJCNN 3*, (1992) pp. 666-671, Beijing.
- [4] D. Cakmakov, Curve Matching Using Turning Functions, *Proc. of the Int. Conf. on Signal and Image Processing, SIP'98*, (1998) pp. 588-592, Las Vegas, USA.
- [5] T. Cibas, F. F. Soulie, P. Gallinari, S. Raudys, Variable Selection with Optimal Cell Damage, *Proc. of the Int. Conference on Artificial Neural Networks, ICANN'94*, (M. Marinaro and P. G. Morasso Eds.), Vol. 1, (1994) pp. 727-730, Springer-Verlag.
- [6] J. Dahmen, D. Keysers, H. Ney, Combined Classification of Handwritten Digits Using the 'Virtual Test Sample Method', *MCS 2001, LNCS 2096*, (J. Kittler and F. Roli Eds.), (2001) pp. 109-118, Springer-Verlag.
- [7] B. Duerr, W. Haettich, H. Tropf, G. Winkler, A Combination of Statistical and Syntactical Pattern Recognition Applied to Classification of Unconstrained Handwritten Numerals, *Pattern Recognition*, 12 (1980), 189-199.
- [8] R. P. W. Duin, D. M. J. Tax, Experiments with Classifier Combining Rules, *MCS 2000, LNCS 1857*, (J. Kittler and F. Roli Eds.), (2000) pp. 16-29, Springer-Verlag.
- [9] L. Heutte, J. V. Moreau, T. Paquet, Y. Lecourtier, C. Olivier, Combining Structural and Statistical Features for the Recognition of Handwritten Characters, *Proc. of the 13th Int. Conf. on Pattern Recognition*, (1996) B74.4, Vienna, Austria.
- [10] T. K. Ho, J. J. Hull, S. N. Srihari, Decision Combination in Multiple Classifier Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1) (1994), 66-77.
- [11] Y. S. Huang, C. Y. Suen, An Optimal Method of Combining Multiple Classifiers for Unconstrained Hand-written Numeral Recognition, *Proc. of IWFHR-3*, (1993) 11-20, Buffalo NY.
- [12] F. Kimura, M. Shridar, Handwritten Numerical Recognition Based on Multiple Algorithms. *Pattern Recognition*, 24(10) (1991), 969-983.
- [13] J. Kittler, Combining Classifiers: A Theoretical Framework, *Pattern Analysis and Applications*, 1(1) (1988), 18-27.
- [14] J. Kittler, M. Hatef, R. P. W. Duin, J. Matas, On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (3) (1998), 226-239.
- [15] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, V. Vapnik, Comparison of Learning Algorithms for Handwritten Digit Recognition, *International Conference on Artificial Neural Networks*, (F. Fogelman and P. Gallinari Eds), (1995) pp. 53-60, Paris.
- [16] H. Liu, H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publisher, Boston, 1998.

- [17] Z. Lou, K. Liu, J. Y. Yang, C. Y. Suen, Rejection Criteria and Pairwise Discrimination of Handwritten Numerals Based on Structural Features, *Pattern Analysis and Applications*, 2(3), (1999), 228-238.
- [18] V. RADEVSKI, Y. BENNANI, Reliability Control in Committee Classifier Environment, *Int. Joint Conf. on Neural Networks, IJCNN 11*, Vol. III, (2000) pp. 561-565. Como, Italy.
- [19] Roli F., Giacinto G., Vernazza G., Methods for Designing Multiple Classifier Systems, *MCS 2001, LNCS 2096*, (Kittler and F. Roli Eds.), (2001) pp. 78-87, Springer-Verlag.
- [20] T. Suzuki, H. Nishida, Y. Nakajima, H. Yamagata, M. Tachikawa, G. Sato, A Handwritten Character Recognition System by Efficient Combination of Multiple Classifiers. *Int. Association for Pattern Recognition Workshop on Document Analysis Systems*, (1995), pp. 169-187, World Scientific, Singapore.
- [21] K. Tumer, J. Ghosh, Linear and Order Statistics Combiners for Pattern Classification, *A. Combining Artificial Neural Nets*, (Sharkey Ed.), (1999) pp 127-162, Springer-Verlag.
- [22] L. Xu, A. Krzyzak, C. Y. Suen, Methods of Combining Multiple Classifiers and Their Applications to Handwritten Recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3), (1992), 418-435.
- [23] S. Yamaguchi, T. Tsutsumida, F. Kimura, A. Iwata, Study on Multi-Expert Systems for Handprinted Numeral Recognition, *Progress in Handwriting Recognition*, (A.C. Downton - S. Impedovo Eds.) (1997), pp. 285-292, World Scientific Publ., Singapore.

Следниот труд со наслов „Partitioning of the Feature Set for Classifier Cooperation“ е печатен во списанието за инженерско творештво и технологија (Journal of Engineering Creation and Technology, UDC 62) во Скопје, Македонија (референца 64 страна 144).

Во овој труд беа испитувани различни разбивања на множеството обележја како и предностите и слабостите на различните комбинирања на класификаторите креирани врз одделните множества обележја. Основниот придонес на овој труд е што авторите покажаа дека разбивањето на множеството обележја според природата на обележјата (структурни и статистички) не е секогаш најдобриот пат за креирање на систем базиран на комбинирање на класификатори. Овој проблем е генерален и може да се изрази на следниов начин: Дали групирањето на обележјата според начинот на нивна екстракција од сликата на знакот е најсоодветно за добивање на некорелирани класификатори, односно доколку сите обележја се разгледуваат како единствено множество на кој начин да се разделат на неколку групи од кои би произлегле независни класификатори погодни за комбинирање.

Резултатите од овој труд ја покажуваат потребата од развивање на посебни процедури за селекција на обележја што на оптимален начин ќе го разбиваат множеството обележја за дизајнирање на индивидуални класификатори што потоа ќе се комбинираат.

Partitioning of the Feature Set for Classifier Cooperation

Abstract

In this paper, various cooperation schemes of SVM (Support Vector Machine) classifiers applied on two feature sets for handwritten digit recognition are examined. We start with a feature set composed of structural and statistical features and corresponding SVM classifier applied on the complete feature set. Later, we investigate the various partitions of the feature set as well as the advantages and weaknesses of various decision fusion schemes applied on SVM classifiers designed for partitioned feature sets. The obtained results show that it is difficult to exceed the recognition rate of a single SVM classifier applied straightforwardly on complete feature set. Additionally, we show that the partitioning of the feature set according to feature nature (structural and statistical features) is not always the best way for designing classifier cooperation schemes. These results impose need of special feature selection procedures for partitioning of feature set for classifier cooperation schemes.

1. INTRODUCTION

The classical paradigm for character recognition is concentrated around two steps, feature extraction, where an appropriate representation of the pattern is developed, and classification, where decision rules for separating pattern classes are defined. Combining features of different nature and the corresponding classifiers has been shown to be a promising approach in handwritten recognition systems [1], [2], [3], [4], [5]. Data from more than one source that are processed separately can often be profitably re-combined to produce more concise, more complete and/or more accurate situation description.

In this paper, classification systems for handwritten digit recognition using two different feature families and SVM classifiers [6] are examined. Following widely used terminology, our feature families are referenced as structural and statistical feature sets [7]. We start with a SVM classifier applied on both feature families as one feature set. These results serve as a basis for future investigations. Further, we used two SVM classifiers that work on the structural and statistical feature families and examined their cooperation using statistical decision fusion. In our terminology, we partition the complete feature set according to the feature nature into statistical and structural feature sets. Different statistical cooperation schemes were examined and corresponding recognition results are presented. In order to improve the system reliability, we introduced rejection criteria as a part of the classifier cooperation schemes.

The presented results show that it is difficult to achieve the recognition rate of a single classifier applied on the feature set that includes both feature families by combining the individual classifier decisions. However, the classifier cooperation schemes offer better possibility for fine tuning of the recognition versus the reliability tradeoff and reduce both, the classifier complexity and the need for samples.

Additionally, we tackle the problem of usefulness of portioning of the feature set according to the natures of the obtaining features. The researchers in area of combining classifiers tacitly agree that portioning of the features according to their nature is the acceptable approach for designing pattern recognition system based on classifier cooperation. The basic supposition is that different features (by their origin) can be considered as reasonably independent. They "see" the same pattern from different points of view and consequently, corresponding individual classifier decisions can be profitably re-combined to produce more accurate recognition. Our results show that statistical and structural features, widely used for designing pattern recognition systems based on classifier cooperation schemes is not always the best way to partition the feature set. This result imposes need of special feature selection procedures for optimal portioning of the feature set in cases of classifier decision fusion applied on different feature families.

Our goals in this paper are to examine usefulness of our feature extraction and selection technique, to study different classifier cooperation schemes and to investigate usefulness of partitioning of the feature set according to the nature of the features rather than to compete with the recognition rates of other handwritten digit recognition systems [8].

2. THE SYSTEM ARCHITECTURE

The recognition system is constructed around a modular architecture of feature extraction and digit classification units. Preprocessed image is an input for the feature extraction module, which transfers the extracted features toward SVM classifiers (see Fig. 1).

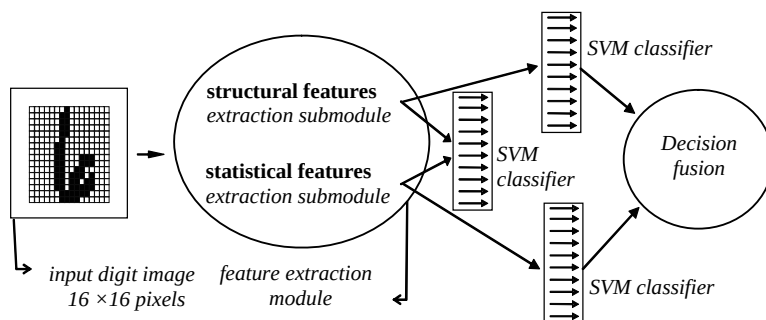


Fig. 1. The system architecture

From the digit images with resolution of 128x128 pixels, we have obtained 16x16 binary images on which the smoothing and centralizing preprocessing techniques have been applied. We have extracted 116 features that are classified as 54 structural and 62 statistical. The both feature families as one set are forwarded to the SVM classifier and obtained results are basis for future comparisons.

The structural and statistical feature sets are also forwarded to the separate SVM classifiers, and obtained results are combined using statistical cooperation schemes. On this level, rejection criteria are introduced and the corresponding system reliabilities are calculated.

3. THE FEATURE EXTRACTION MODULE

The structural feature set is a domain dependent set. Its nature and the techniques implemented for detection and extraction are strongly dependent of the nature of the objects to be recognized.

The first step in creating of the structural feature set is defining a reasonable set of elementary shape primitives for digit constructions. We have proposed 27 elementary primitives showed in Fig. 2. The digit image is searched for these primitives twice: firstly on the original digit image orientation, and secondly on the rotated digit image for 90°. So, the total number of primitives is 54, and that is the number of the elements in the structural feature set.

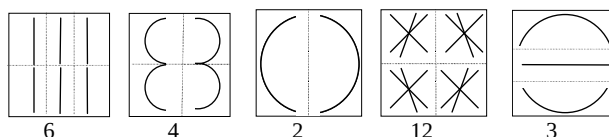


Fig. 2. Image sub-regions and the elementary primitives

The detection and the extraction of the structural features are performed by dividing the image binary matrix into two, three, four and six sub-regions. The existing shape in each of those sub-regions is compared with the proposed primitives in the same sub-regions whose existence is expected.

Thus, the structural feature is composed of 54 values of the calculated similarities [9] between the found shapes in the corresponding sub-regions and the corresponding elementary primitives.

The statistical feature set is composed of 62 features that give the pixel-based information presented by the densities of the lit pixels in various regions of the digit image. The first 54 statistical features are obtained from the projection histograms obtained by the vertical (16), horizontal (16) and two diagonal (22) projections (5 pixels left and right around the main diagonals). The last 8 features are obtained from the zone-pattern regions. This kind of features in different forms has been exploited in many pattern recognition systems [e.g. 10].

4. THE RECOGNITION RESULTS

The database for our experiments is an extraction of the NIST (National Institute of Standards and Technology) handwritten digit database. The total number of 23898 digit images is divided into two groups, 17952 images for the training

phase and 5946 images for the test phase. The digits from the original database are rearranged so that digits in the test set belong to different writers from those in the learning set.

We have used a SVM classifier with Gaussian kernel. Because of the large number of samples, a more robust variation of SVM training software (SVM Torch) has been used [11].

The outputs of this classifier applied on our samples fall in $[-8, 5]$ interval. Let us denote the classifier outputs in descending order by $O_1, O_2, \dots, O_{10}$ ($O_1 \geq O_2 \geq \dots \geq O_{10}$). We have used a rejection criterion based on the top two classifier outputs. All the samples with highest value of classifier outputs that is smaller than a certain threshold T_1 ($O_1 < T_1$) or for which the difference between the top two classifier outputs is smaller than a certain threshold T_2 ($O_1 - O_2 < T_2$) are rejected. Varying these thresholds to obtain reliability of at least 99% we have obtained the results shown in Table 1.

Table 1. Recognition rates on the structural, statistical and both feature families

SVM (Gaussian kernel)	Recog. (%)	T_1, T_2	RRecog. (%)	Miscl. (%)	Reject. (%)	Reliab. (%)
Statistical features	97.01	-0.533, 0.9988	92.40	0.92	6.68	99.01
Structural features	94.92	0.382, 1.4626	81.97	0.81	17.22	99.02
Structural + Statistical features	97.73	0.0056, 0.577	94.76	0.94	4.29	99.02

Recog. is the classifier recognition rate. RRecog., Miscl. and Reject. denote the recognition, misclassification and rejection rates for reliability of at least 99% provided by the rejection criterion using the corresponding values of T_1 and T_2 . Reliab. denotes the reliability that is calculated as $\text{Reliab.} = \text{RRecog.}/(100\% - \text{Reject.})$. These results show that the statistical feature set has stronger discrimination power and provides better recognition rate. However, the recognition rate of the statistical feature set is more than 0.7 percent lower than the recognition rate of the classifier applied to the complete feature set.

4.1. Decision Fusion on Statistical and Structural Feature Sets

The classifier cooperation schemes are built around two SVM classifiers performing classification separately on structural and statistical feature families. In Table 4, the recognition rates using various statistical cooperation schemes are presented. We have used the same rejection criterion as in Table 1, and suitable values for T_1 and T_2 were chosen in order to achieve reliability of at least 99%.

The decision fusion methods: Product, Dempster Rule, Fuzzy Integral, and Decision Templates require possibilistic outputs. To map the original output values to $[0, 1]$ interval we used the mapping $1/(1+e^{-x})$.

Table 2. Statistical cooperation schemes and corresponding recognition rates

#	Cooperation schemes	Recog. (%)	T_1, T_2	RRecog. (%)	Miscl. (%)	Reject. (%)	Reliab. (%)
1.	Average	97.70	0.453, 0.140	95.34	0.96	3.70	99.00
2.	Product	97.73	0.376, 0.039	95.39	0.96	3.65	99.01
3.	Max-Max	97.09	0.590, 0.197	93.51	0.92	5.57	99.02
4.	Min-Max	97.23	0.322, 0.122	94.80	0.92	4.27	99.03
5.	Borda count	97.70	0.907, 0.281	95.34	0.96	3.70	99.00
6.	Naive Bayes	97.24	0.919, 0.860	93.98	0.92	5.10	99.03
7.	Dempster	97.78	0.217, 0.081	94.82	0.94	4.24	99.02
8.	Fuzzy Integral	97.09	0.590, 0.197	93.51	0.92	5.57	99.02
9.	Decision Templates P1	97.66	0.518, 0.097	94.57	0.94	4.49	99.01
10.	Decision Templates P2	97.68	0.857, 0.033	94.90	0.92	4.17	99.03
11.	Decision Templates P3	97.61	0.805, 0.030	95.06	0.94	4.00	99.02
12.	Decision Templates P4	97.56	0.497, 0.107	94.82	0.92	4.25	99.03
13.	Decision Templates I1	97.66	0.645, 0.097	95.19	0.96	3.85	99.00
14.	Decision Templates I2	97.60	0.932, 0.016	94.38	0.92	4.69	99.03
15.	Decision Templates I3	97.56	0.827, 0.015	94.94	0.94	4.12	99.02
16.	Decision Templates I4	96.99	0.489, 0.183	93.53	0.92	5.55	99.02
17.	Decision Templates I5	97.09	0.352, 0.216	93.02	0.91	6.07	99.03
18.	Decision Templates C	97.12	0.577, 0.195	93.58	0.92	5.50	99.02
19.	Decision Templates E	97.71	0.967, 0.022	95.16	0.94	3.90	99.02
20.	Dynamic average	97.70	0.204, 0.083	95.24	0.94	3.82	99.02
21.	Generalized Committee	97.73	0.221, 0.073	95.41	0.94	3.65	99.02
22.	Modified Generalized Committee	97.76	0.216, 0.074	95.27	0.94	3.78	99.02

In order to make the final decision, first four cooperation schemes use the maximum of the sum, the maximum of the product, the maximum of the maximum and the maximum of the minimum of the corresponding pairs of the classifier outputs [12]. The Dempster rule considers the fuzziness of the classifier votes by giving less confidence to less certain votes [13]. The naive Bayes cooperation scheme uses the confusion matrices of member classifiers to estimate the certainty of the classifier decisions [13]. The Borda count cooperation method is a generalization of the majority vote [14]. The fuzzy integration is based on searching for the maximal grade of agreement between the objective evidence (provided by the sorted classifier outputs for class i) and the expectation (the fuzzy measure values of both classifiers) [15]. We have also used decision template approaches described elsewhere [16]. The dynamic average approach uses dynamic weights that are adjusted to be proportional to the certainties of the respective classifier outputs rather than globally chosen weights as in generalized committee [17]. The generalized committee prediction is based on a weighted combination of the predictions of the member classifiers [18]. The modified generalized committee differs from the generalized committee only by the chosen weights that are different for every class output. In other word we are using a vector of adjusted weights rather than one common weight per classifier.

A few results in Table 2 deserve attention. The best recognition rates (Recog.>97.70%) are obtained by five of the cooperation schemes. Let us note that

these results are almost identical to the recognition rate of the SVM that uses both feature families as one feature set (see Table 1).

On the other hand, the best recognition rates with reliability of 99% are provided by the schemes 21 (Generalized Committee) and 2 (Product). These results are noticeably better than the corresponding results shown in (Table 2). Generally speaking, the classifier cooperation schemes with rejection criteria offer improved recognition rates in comparison to the classifier that utilizes the both feature families in one feature set.

4.2. Decision Fusion on Random Partitions of the Feature Set

To examine usefulness of grouping features according to their "nature" we performed a few random partitioning of the complete feature set in different relations between structural and statistical features.

Table 3. Average recognition rates of cooperation schemes applied on 10 random partitioning of the feature set

#	Cooperation schemes	Recog. (%)	RRecog. (%)	Miscl. (%)	Reject. (%)	Reliab. (%)
1.	Average	97.67 ($\sigma=0.10$)	95.32 ($\sigma=0.37$)	0.94	3.74	99.00
2.	Product	97.69 ($\sigma=0.10$)	95.31 ($\sigma=0.35$)	0.95	3.75	99.01
3.	Max-Max	97.38 ($\sigma=0.09$)	94.51 ($\sigma=0.62$)	0.93	4.56	99.02
4.	Min-Max	97.36 ($\sigma=0.14$)	94.45 ($\sigma=0.42$)	0.94	4.61	99.03
5.	Borda count	97.67 ($\sigma=0.10$)	95.32 ($\sigma=0.37$)	0.94	3.74	99.00
6.	Naive Bayes	97.24 ($\sigma=0.12$)	74.60 ($\sigma=15.1$)	0.65	24.75	99.03
7.	Dempster	97.63 ($\sigma=0.10$)	95.10 ($\sigma=0.45$)	0.94	3.96	99.02
8.	Fuzzy Integral	97.38 ($\sigma=0.09$)	94.51 ($\sigma=0.62$)	0.93	4.56	99.02
9.	Decision Templates P1	97.61 ($\sigma=0.08$)	95.10 ($\sigma=0.35$)	0.94	3.96	99.01
10.	Decision Templates P2	97.63 ($\sigma=0.07$)	95.09 ($\sigma=0.34$)	0.94	3.97	99.03
11.	Decision Templates P3	97.66 ($\sigma=0.08$)	95.24 ($\sigma=0.41$)	0.94	3.83	99.02
12.	Decision Templates P4	97.43 ($\sigma=0.10$)	94.58 ($\sigma=0.54$)	0.94	4.49	99.03
13.	Decision Templates I1	97.60 ($\sigma=0.08$)	95.07 ($\sigma=0.32$)	0.93	4.00	99.00
14.	Decision Templates I2	97.57 ($\sigma=0.09$)	94.96 ($\sigma=0.45$)	0.94	4.10	99.03
15.	Decision Templates I3	97.64 ($\sigma=0.11$)	95.21 ($\sigma=0.40$)	0.94	3.85	99.02
16.	Decision Templates I4	97.13 ($\sigma=0.12$)	94.20 ($\sigma=0.60$)	0.93	4.88	99.02
17.	Decision Templates I5	97.38 ($\sigma=0.09$)	94.45 ($\sigma=0.62$)	0.93	4.63	99.03
18.	Decision Templates C	97.37 ($\sigma=0.10$)	94.44 ($\sigma=0.62$)	0.93	4.63	99.02
19.	Decision Templates E	97.63 ($\sigma=0.10$)	95.21 ($\sigma=0.31$)	0.94	3.86	99.02
20.	Dynamic average	97.65 ($\sigma=0.10$)	95.30 ($\sigma=0.35$)	0.95	3.75	99.02
21.	Generalized Committee	97.75 ($\sigma=0.07$)	95.34 ($\sigma=0.31$)	0.94	3.72	99.02
22.	Modified Generalized Committee	97.82 ($\sigma=0.12$)	95.34 ($\sigma=0.38$)	0.94	3.72	99.02

Ten pairs of randomly partitioned feature subsets were created by combining 25 randomly chosen features from the structural feature set and 29 randomly chosen features from the statistical feature set to form a mixed feature set containing 54 fea-

tures. The remaining 29 features from the structural feature set and the remaining 33 features from the statistical feature set were combined in another feature set containing 62 features. By such partitioning and joining the parts of the structural and statistical feature sets, we have obtained two feature sets with the same number of features as the original structural and statistical feature sets. The new feature subsets contain features from both feature families in relations that correspond to structural and statistical feature families $25:29 \approx 29:33 \approx 54:62$. By this partitioning we were able to use the same system to conduct the experiments, i.e. to avoid most of disturbances that could be implied by the altered system.

For each pair of obtained feature subsets a pair of SVM classifiers was trained to perform classification over a single set (separately). The outputs of each pair of classifiers were combined using the same classifier cooperation schemes given in Table 2. In Table 3, average recognition rates over the ten random partitioning of the feature set in described way are presented.

The obtained results show that by mixing the features from the “weaker” structural and “stronger” statistical feature set, the performances of the classifier trained on these mixed feature sets perform more comparably with 0.65% average difference in recognition performance and 2.69% when applying rejection criteria for reliability of 99%. The slight advantage of the feature set that contains 62 features is expected, because of the larger number of features used to represent the sample.

The recognition results obtained by random partitioning of the feature set (Table 3) are not worse, but also not better than the recognition results obtained by partitioning of the feature set according to the feature nature (Table 2). Let us note that in Table 3 five recognition rates exceeds 97.70%, while in Table 4 there are only two. On the other hand, the best recognition rate comes from the Modified Generalized Committee in Table 4 (97.82%). By applying rejection criteria in the cooperation schemes, the situation is quite similar in opposite way. Now, in Table 3 we have five recognition rates over 95.30%, while there are four in Table 3. The best result comes from the Generalized Committee in Table 3 (93.41%).

5. CONCLUSION

In this paper, we address some issues in designing high reliability system for hand-written digit recognition using SVM classifiers. We used two different feature families referenced as structural and statistical features. Decision level fusion is performed using statistical cooperation schemes. To improve the system reliability, we introduced rejection criteria in decision fusion schemes.

The presented results show that it is difficult to achieve the recognition rate of the single classifier applied on the complete feature set by statistical decision fusion applied on the individual classifier outputs. However, classifier cooperation schemes reduce classifier complexity, slightly improve recognition rates and enable fine tuning of the recognition versus the reliability tradeoff. Additionally, we compare decision fusions of the classifiers designed for different partitioning of the feature set including random partitioning and partitioning by feature nature and show that “natural” partitioning of the feature set according to feature nature is not the best way to apply classifier decision fusion.

REFERENCES

- [1] F. Kimura, M. Shridar: Handwritten Numerical Recognition Based on Multiple Algorithms, *Pattern Recognition*, Vol. 24, No. 10, 1991, pp. 969-983.
- [2] Y. S. Huang, C. Y. Suen: A Method of Combining Multiple Classifiers – A Neural Network Approach. *Proc. 12th Int. Conf. Pattern Recognition and Computer Vision*, Jerusalem, 1994, pp. 473-475.
- [3] T. Suzuki, H. Nishida, Y. Nakajima, H. Yamagata, M. Tachikawa, G. Sato: A Handwritten Character Recognition System by Efficient Combination of Multiple Classifiers, *Int. Association for Pattern Recognition Workshop on Document Analysis Systems*, World Scientific, Singapore, 1995, pp. 169-187.
- [4] S. Yamaguchi, T. Tsutsumida, F. Kimura, A. Iwata: Study on Multi-Expert Systems for Handprinted Numeral Recognition, In: Downton, A.C., Impedovo, S. (eds.): *Progress in Handwriting Recognition*. World Scientific, Singapore, 1997, pp. 285-292.
- [5] J. Dahmen, D. Keysers, H. Ney: Combined Classification of Handwritten Digits Using the 'Virtual Test Sample Method'. In: Kittler, J., Roli, F. (eds.): *Multiple Classifier Systems*, 2nd Int. Workshop, MCS 2001 Cambridge, UK, *Lecture Notes in Computer Science* Vol. 2096, Springer-Verlag, 2001, pp. 109-118.
- [6] C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, *Knowledge Discovery and Data Mining*, Vol. 2, 1998, pp. 1-47.
- [7] V. Radevski, Y. Bennani: Reliability Control in Committee Classifier Environment, *Int. Joint Conf. on Neural Networks*, IJCNN 11, Como, Italy, Vol. III., 2000, pp. 561-565.
- [8] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, V. Vapnik: Comparison of Learning Algorithms for Handwritten Digit Recognition. In: Fogelman, F., Gallinari, P. (eds.): *Int. Conf. Artificial Neural Networks*, Paris, 1995, pp. 53-60.
- [9] D. Cakmakov: Curve Matching Using Turning Functions. *Proc. of the Int. Conf. on Signal and Image Processing SIP'98*, Las Vegas, 1998, pp. 588-592.
- [10] G. Burel, I. Pottier, J. Y. Catros: Recognition of Handwritten Digits by Image Processing and Neural Network, *Proc. Of the Int. Joint Conf. on Neural Networks*, IJCNN 3, 1992, pp. 666-671.
- [11] R. Collobert, S. Bengio: SVM Torch: Support Vector Machines for Large-Scale Regression Problems, *Journal of Machine Learning Research*, Vol. 1, 2001, pp. 143-160.
- [12] J. Kittler, M. Hatef, R. P. W. Duin, J. Matas: On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, 1998, pp. 226-239.
- [13] L. Xu, A. Krzyzak, C. Y. Suen: Methods of Combining Multiple Classifiers and Their Application to Handwritten Recognition, *IEEE Transactions on System, Man and Cybernetics*, Vol. 22, 1992, pp. 418-435.
- [14] T. K. Ho, J. J. Hull, S. N. Srihari: Decision Combination in Multiple Classifier Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1, 1994, pp. 66-75.
- [15] S. B. Cho, J. H. Kim: Combining Multiple Neural Networks by Fuzzy Integral and Robust Classification, *IEEE Transactions on System, Man and Cybernetics*, Vol. 20, No. 3, 1995, pp. 380-384.
- [16] L. I. Kuncheva, J. C. Bezdek, P. W. Duin: Decision Templates for Multiple Classifier Fusion, *Pattern Recognition*, Vol. 34, No. 2, 2001, pp. 299-314.

-
- [17] A. D. Jimenez, N. Walsh: Dynamically Weighted Ensemble Neural Networks for Classification, *Proceedings of the 1998 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK 1998.
 - [18] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995, pp. 364-369.

Трудот со наслов „An Efficient Three-Stage Classifier for Handwritten Digit Recognition“ е изложен на 17-тата меѓународна конференција за препознавање примероци (ICPR 2004) што се одржа во Кембриџ, Англија (референца 61 на страна 144).

Имајќи ја предвид супериорноста на SVM класификаторите во поглед на генерализацијата, но и нивната пресметковна комплексност којашто е значително поголема од, на пример невронските мрежи, овој труд дава решение како да се искористат добрите особини на различните типови класификатори. Главната цел беше да се задржи точноста на SVM класификаторите кои користат голем број на софистицирани обележја, а при тоа да се намали пресметковната комплексност на системот и да се забрза препознавањето. Сето тоа е постигнато со тростепен класификатор за препознавање на броеви напишани со рака. Системот користи две невронски мрежи во првите два степенa и SVM класификатор на последниот степен, при што секој степен работи со отфрлените примероци од претходниот.

Спроведените тестовите за препознавање на изолирани цифри од MNIST базата, покажаа дека предложениот систем постигна забрзување во препознавањето од скоро 28 пати и мало подобрување на точноста во однос на соодветниот индивидуален SVM класификатор. Добиените резултати се меѓу најдобрите резултати за MNIST базата на ракописни цифри.

An Efficient Three-Stage Classifier for Handwritten Digit Recognition

Abstract

This paper proposes an efficient three-stage classifier for handwritten digit recognition based on NN (Neural Network) and SVM (Support Vector Machine) classifiers. The classification is performed by 2 NNs and one SVM. The first NN is designed to provide a low misclassification rate using a strong rejection criterion. It is applied on a small set of easy to extract features. Rejected patterns are forwarded to the second NN that uses additional, more complex features, and utilizes a well-balanced rejection criterion. Finally, rejected patterns from the second NN are forwarded to an optimized SVM that considers only the “top k ” classes as ranked by the NN. This way a very fast SVM classification is obtained without sacrificing the classifier accuracy. The obtained recognition rate is among the best on the MNIST database and the classification time is much better compared to the single SVM applied on the same feature set.

1. Introduction

The accuracy of an overall recognition system mainly depends on the discriminative capability of the extracted features and the generalization performance of the designed classifier.

Over the last two decades, NNs have been widely used to solve complex classification problems [1]. They are very fast classifiers with relatively fast training time. However, a single NN often exhibits the overfitting behavior which results in a weak generalization performance when trained on a limited set of training data. On the other hand, there is a consensus in machine learning community that SVMs are most promising classifiers due to their excellent generalization performance [2]. Curse of dimensionality and overfitting in NNs, seldom occur in SVMs. However, SVMs for multi-classes classification problems are relatively slow and their training on a large data set is still a bottle-neck.

In this paper, we present an efficient system for handwritten digit recognition attempting to utilize the advantages of both, NN and SVM classifiers. Our system is three-staged where firstly two NNs with rejection criteria are used and then, a reduced one-against-the-rest SVM classification is exploited for the rejected patterns.

In the first stage, our goal was to perform fast classification and to achieve low misclassification rate using a small set of 40 features and NN with strong rejection criterion.

In the second stage, rejected patterns from the first stage are represented by additional 252 features and forwarded to the second NN with a well-balanced rejection criterion.

In the third stage, the rejected patterns from the NN of the second stage together with the class rankings are forwarded to a SVM. Thus, we have obtained a SVM with reduced complexity aimed to classify small number of “hard to classify” patterns considering only the k top-ranked classes of the rejected patterns.

Carefully optimized classifiers and choice of the features have led to fast and accurate recognition system. Among the many handwritten digit recognition systems with high recognition rates e.g. [3], [4], [5], [6], the recognition rate of our system on MNIST database is among the best.

2. The System Architecture

Pre-processed isolated digit images are input for the feature extraction modules, that transfer the extracted features toward NN and SVM classifiers (Figure 4).

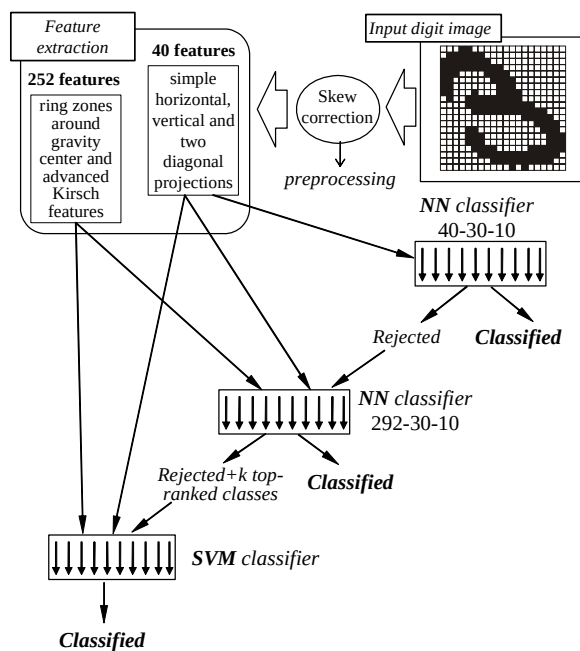


Figure 4. The system architecture

Every image is centered in a square bounding box, and then slant correction is performed. The slant angle is estimated as the inclination of the line connecting the gravity centers of the top 25% part and the bottom 25% part of the image. Then a sub-pixel precision shear transformation is performed in order to remove the estimated inclination. This approach provides more reliable slant correction than the “standard” approaches e.g. [7].

The first set of features contains 40 easy to extract projection-based features that are input for the first NN classifier (40–30–10). From the patterns that are rejected from this network, additional 252 more robust features are extracted. All 292 features are used in the second and the third classification stage where “hard to classify” patterns are firstly forwarded to the second NN (292-30-10). The patterns rejected from the second NN are finally classified by the SVM.

This way, we have obtained a low time consuming classifier with slightly better recognition rate compared to the single SVM applied on the same feature set.

3. Feature extraction

Features that offer better discriminative power are usually more complicated and harder to extract regarding processing time that can not be neglected when building a fast recognition system.

The first 40 features used in the first classification stage are simple horizontal, vertical and diagonal projections. Since the character images are not of the same size, the projection vectors are linearly rescaled in order to obtain 10 features from the horizontal projections, 8 features from the vertical projections, and 11 features from each of the two diagonal projections (see Figure 5).

The second 44 features are used in the second and third stages of classification. They are extracted as pixel counts in rings zones around the gravity center of the image (see Figure 5). We use three rings each divided in different number of equal zones. The outermost ring has a radius r equal to the distance from the gravity center to the furthest black pixel of the image. The first ring with radius $0.2 \cdot r$ provides 4 features and the second ring with radius $0.5 \cdot r$ provides 24 features. The last 16 features of this group are provided from the outermost ring.

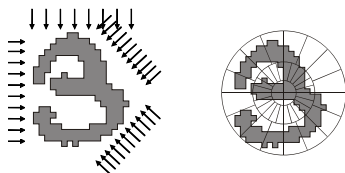


Figure 5. Projection and ring-zone features

The last 208 features are also used in the second and third stages of classification. These features use Kirsch operator to detect local directional information of the input pattern edges [8]. Compared with chain code which also describes the edge direction, Kirsch edge detection is more robust even under noisy conditions.

The image is scanned from left to right, top to bottom, right to left and bottom to top, respectively. The first black pixel which the scanning line intersects forms the first outermost periphery. The second black pixel which is the starting point of the second run forms the second outermost periphery (see Figure 6). When the image is scanned in horizontal direction, the vertical and both diagonal Kirsch features are extracted at the outermost periphery. When the image is scanned in vertical direction, the horizontal and both diagonal Kirsch features are extracted at the outermost and second outermost periphery. This way, 3 Kirsch directional features plus one feature representing the position of the periphery measured as the distance from the edge of the bounding box of the image are provided for each periphery pixel. The feature vectors are again linearly rescaled to 40 features coming from the left and right periphery each, 32 features coming from the first outermost top and bottom periphery each, and 32 features coming from the second outermost top and bottom peripheries.

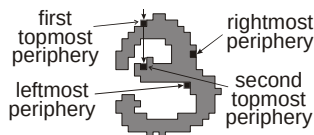


Figure 6. Kirsch features

The image processing is performed on the grayscale digit images using sub-pixel precision. All parameters including the number of features by projection, the radiuses of rings for zone-pattern regions and the number of features coming from the outermost peripheries for Kirsch features are carefully chosen after several iterations using observations about their discriminative power. The features are pre-processed for zero mean and unit variance.

4. Classification

Classification is performed in three stages where the first and second stage classifiers send rejected patterns to the next classifier.

In the first stage, we have used a multilayer perceptron NN with one hidden layer and architecture 40-30-10. In this stage, our goal was to perform fast classification of “easy to classify” patterns keeping low misclassification rate using a strong rejection criterion. So, the input feature set was small (40 features) containing features extracted from pattern projections. The rejection criterion is based on the “top 2” NN outputs. Each sample for which the highest NN output O_1 is smaller than a certain threshold T_1 ($O_1 < T_1$) or for which the difference between the “top 2” classifier outputs is smaller than a certain threshold T_2 ($O_1 - O_2 < T_2$) are rejected. Varying these thresholds to obtain low misclassification rate we have found suitable values $T_1 = 0.994$ and $T_2 = 5.5$ that are used to obtain presented recognition results.

In the second stage, we have used a multilayer perceptron NN with one hidden layer and architecture 292-30-10. This NN uses 252 additional features extracted from the digit image. The rejection criterion was the same as in stage two with parameters $T_1 = 0.985$ and $T_2 = 4.0$.

In the third stage, we have used a SVM with Gaussian kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \sigma^2)$. In this stage, for the remained “hard to classify” patterns, the complete feature set of 292 features is used again. To reduce SVM complexity and speed up the classification process, the SVM examines only the k top-ranked classes obtained by the second NN.

Since SVMs are binary classifiers, the one-against-the-rest method is used to construct ten-class classifier. That is, each classifier is constructed by separating one class from the rest. The usual classification approach is to present the pattern sequentially to all ten one-against-the-rest SVMs and then to make the decision by choosing the class with the largest classifier output value. However, if strong evidence is provided in advance that certain pattern is not member of some classes, one can decide not to present the pattern to the SVMs for the corresponding classes. Since the SVMs are considered in a sequentially manner, discarding some classes from the consideration saves significant amount of time/processing.

5. The Recognition Result

Our experiments were performed on the well-known MNIST database (<http://yann.lecun.com/exdb/mnist>) of handwritten digits. MNIST database consists of 60000 training samples and 10000 test samples. All digits have been size-normalized and centered in a 28×28 box.

The NN and SVM classifiers were implemented using the TORCH [9] library while the feature extraction was performed using proprietary C++ code. The C++ programs were compiled by Microsoft Visual C++.NET 7.1. All tests were performed on 2.4GHz P4 processor under Windows XP.

The NNs were trained on the whole training set using stochastic gradient descent. The SVM for every class was carefully optimized for the parameters σ and C using an automated parameter search procedure. For the parameter optimization 40000 samples were used for training and 20000 for validation. After finding the optimal σ and C , the SVM was finally trained on the whole set of 60000 samples.

In Table 4, the individual error rates and the recognition times of the two NNs and the SVM are presented. It is obvious that the SVM has a superior recognition rate but it is a level of magnitude slower than the NNs. The CPU times are given for the recognition of the whole MNIST test set containing 10000 samples (excluding pre-processing and feature extraction).

Table 4. Individual classifier performance

Classifier	Error rate (%)	CPU (s)
NN 40-30-10	3.31	0.129
NN 292-30-10	1.31	0.330
SVM	0.85	196.031

Table 5, the number of recognized, misclassified and rejected patterns in every stage of the proposed recognition system together with the rejection thresholds and the CPU times are given. To obtain the complete recognition time, the time spent for pre-processing (PP), extraction of the first 40 features (FE40) and extraction of the additional 252 features (FE252) are also given.

Table 5. Characteristics and efficiency of the proposed 3-stage classifier

Action	#Rec.	#Mis.	#Rej.	T_1, T_2	CPU (s)
PP + FE40	on 10000 patterns			–	1.252
1 st stage NN	6551	6	3443	0.994, 5.5	0.129
FE252	on 3443 patterns			–	2.492
2 nd stage NN	3053	23	367	0.985, 4.0	0.114
SVM top 4	311	54	–	–	3.031
Total	9917	83	–	–	7.018

The lowest error rate of 0.83% (83/10000) was obtained when 4 top-ranked classes from the second NN were considered by the SVM. Of course, faster recognition times (5.656 and 6.354 seconds) could be achieved by considering only 2 or 3 top-ranked classes by the SVM, but the obtained error rates (0.96% and 0.84%) were

higher. Increasing the number of top-ranked classes that are considered by the SVM above $k = 4$ has not decreased the error rate. This means that the NN from the second stage manages to keep the right class among the 4 top-ranked classes for the rejected patterns that are going to be recognized with high accuracy by the SVM.

The lowest obtained error rate of 0.83% is slightly better than the error rate of the best individual classifier (SVM) given in Table 4. That is because some of the patterns that would be misclassified by the SVM are correctly recognized by some of the NN classifiers.

In Figure 7, a comparison of the performances of different algorithms tested on the MNIST database is given. The proposed method (3-Stage NN-NN-SVM) was used on the original MNIST database and provided error rate of 0.83% that is better than error rates of most of the classifiers.

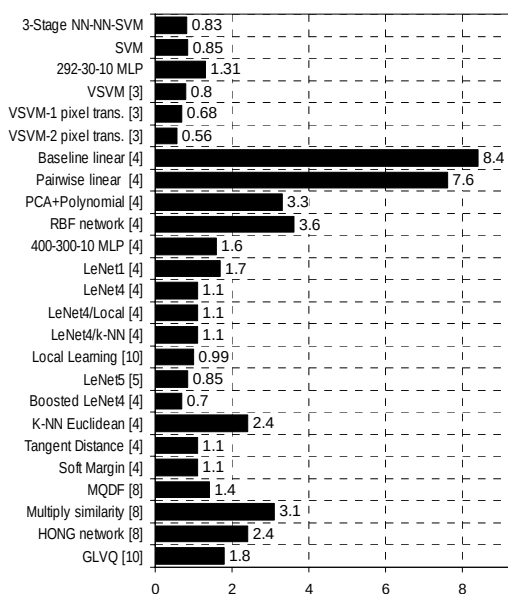


Figure 7. Error rates of different methods on the test set of MNIST database

The best results, provided by Boosted LeNet4 [4] and Virtual SVMs [3] are the state-of-art results, but corresponding classifiers were trained on a perturbed MNIST database, where the training set was augmented with artificially altered versions of the original training samples.

6. Conclusion

This paper proposes an efficient three-stage classification of handwritten digits based on two NNs and one SVM.

In the first stage, a NN is designed to provide a low misclassification rate using strong rejection criterion. In order to be fast it is applied on a small set of easy to extract features.

In the second stage, additional more complex features are extracted from the rejected patterns and forwarded to the second NN. A well-balanced rejection criterion is also applied in order to provide a low misclassification rate.

Rejected patterns from the second stage, together with the class ranking obtained by the NN are forwarded to a SVM. Thus, we have obtained a SVM with reduced complexity because instead of considering all possible classes for every pattern it considers only the k top-ranked classes (best recognition rate was for $k = 4$).

To achieve so high recognition rates, the feature set is carefully chosen as a combination of simple projections, zone-pattern regions around gravity center and powerful Kirsch features extracted from the image peripheries.

Obtained classification time and recognition rates are among the best on MNIST database. They are also better than the recognition time and the recognition rate of single SVM applied on the same feature set.

References

- [1] Bishop C.M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995, pp. 364-369.
- [2] Burges C., A Tutorial on Support Vector Machines for Pattern Recognition, *Knowledge Discovery and Data Mining*, Vol. 2, 1998, pp. 1-47.
- [3] DeCoste D. and Scholkopf B., Training Invariant Support Vector Machines, *Machine Learning*, Vol. 46, No. 1-3, pp. 161-190, 2002.
- [4] LeCun Y., Jackel L. D., Bottou L., Brunot A., Cortes C., Denker J. S., Drucker H., Guyon I., Muller U. A., Sackinger E., Simard P., and Vapnik V., Comparison of Learning Algorithms for Handwritten Digit Recognition, In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, Paris, 1995, pp. 53-60.
- [5] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, 86 (11), 1998, pp. 2278-2324.
- [6] Dong J.X., Comparison of Algorithms for Handwritten Numeral Recognition, *Technical Report*, CENPARMI, Concordia University, Nov. 1999.
- [7] Grother P. J., Karhunen Loève Feature Extraction for Neural Handwritten Character Recognition, *Proceedings of Applications of Artificial Neural Networks III*, SPIE, Orlando, Florida, 1992, pp. 155-166.
- [8] Lee S.W., Multilayer Cluster Neural Networks for Totally Unconstrained Handwritten Numeral Recognition, *Neural Networks*, Vol. 8, No. 5, 1995, pp. 783-792.
- [9] Collobert R., Bengio S., and Mariéthoz J., Torch: a Modular Machine Learning Software Library, *Technical Report IDIAP-RR 02-46*, IDIAP, 2002. (<http://www.torch.ch>)
- [10] Dong J.X., Krzyzak A., Suen C.Y., Local Learning Framework for Handwritten Character Recognition, *Engineering Applications of Artificial Intelligence*, Vol. 15, 2002, pp. 151-159.

Последниот труд што тука ќе го дадеме е со наслов „Handwritten Digit Recognition Using Classifier Cooperation Schemes“. Тој е изложен на 2-та балканска конференција за информатика (BCI 2005) што се одржа во Охрид, Македонија (референца 24 на страна 142).

Во овој труд се разгледани кооперации на три SVM класификатори за препознавање на цифри испишани со рака каде што секој класификатор користи различно множество обележја. Според нашите сознанија ваков голем број шеми на комбинирање до сега не се презентирани во научната литература. Одреден број од разгледуваните комбинирања се оригинални подобрувања предложени од страна на авторите на веќе познати комбинаторни шеми. Резултатите од овој труд, како и од други трудови на авторите, покажуваат дека е тешко да се надмине ратата на препознавање на еден добро дизајниран SVM класификатор што го користи целото множество обележја со комбинирање на индивидуални класификатори дизајни посебно за дисјунктни групи обележја. Сепак, главните придобивки на методите за комбинирање на класификатори се во намалувањето на комплексноста на класификаторот, редукцијата на бројот на примероци потребни за обука на индивидуалните класификатори (земени заедно и со примероците евентуално потребни за обука на комбинаторот) и намалување на потребното време за обука.

Handwritten Digit Recognition Using Classifier Cooperation Schemes

Abstract

Recent results in pattern recognition applications have shown that SVMs (Support Vector Machines) have superior recognition rates in comparison to other classification methods. In this paper, the cooperation of three SVM classifiers for handwritten digit recognition, each using different feature family is examined. We investigate the advantages and weaknesses of various cooperation schemes based on classifier decision fusion using statistical reasoning. Although most of the used schemes are variations and adaptations of existing ones, such an extensive number of cooperation schemes have not been presented in the literature until now. The obtained results show that it is difficult to exceed the recognition rate of a single, well-tuned SVM classifier applied straightforwardly on all feature families as a single set. However, the classifier cooperation reduces the classifier complexity and need for samples, decreases classifier training time and sometimes improves the classifier performance.

1. Introduction

Combining features of different nature and the corresponding classifiers has been shown to be a promising approach in many pattern recognition applications. Data from more than one source that are processed separately can often be profitably re-combined to produce more concise, more complete and/or more accurate situation description. In this paper, we discuss classification systems for handwritten digit recognition using three different feature families and SVM classifiers [1]. We start with a SVM classifier applied on all feature families as one set. Further, we used three SVM classifiers that work on the different feature families for the same digit image. As the feature sets “see” the same digit image from different points of view, we examined the possibility of decision fusion using statistical cooperation schemes. An extensive number of cooperation schemes were examined and corresponding recognition results are presented. Our aim was not to compete with the recognition rates of the other handwritten digit recognition systems e.g. [2], [3], but to compare the qualities of different feature families, corresponding SVM classifiers and their combination based on different classifier decision fusion.

The presented results show that it is difficult to achieve the recognition rate of a single optimized SVM classifier applied on the feature set that includes all feature families. However, the cooperation of individual classifiers designed for separate feature families reduce the classifier complexity and need for training samples, offering better opportunity to understand the role of the features in the recognition process.

2. The System Architecture

The recognition system is constructed around a modular architecture of feature extraction and digit classification units. The preprocessed isolated digit images

are input for the feature extraction module, that transfers the extracted features toward SVM classifiers (see Fig. 1).

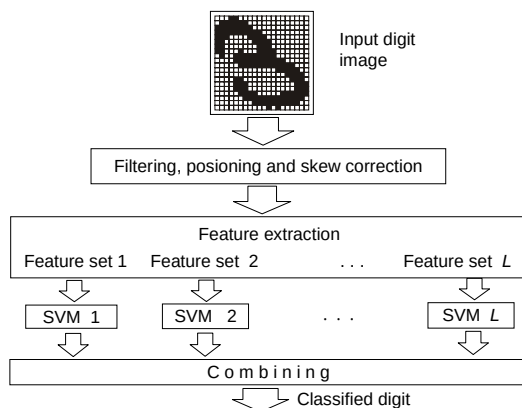


Fig. 8. The system architecture

Each image is centered in a square bounding box, and then slant correction is performed. The slant angle is estimated as the inclination of the line connecting the gravity centers of the top 25% part and the bottom 25% part of the image. Then a sub-pixel precision shear transformation is performed in order to remove the estimated inclination.

3. Feature Extraction

Three feature families were extracted from each digit image:

- contour profiles,
- ring-zones and
- Kirsch features.

Feature extraction was performed on the original unscaled images, after the slant correction.

The first feature family (FS1) is composed of 30 contour profile features (see Fig. 2). The image is scanned from left to right, top to bottom, right to left and bottom to top, respectively. The distance from the corresponding edge of the image to the first black pixel which the scanning line intersects, represent the contour profile feature on the first level. The distance to the first black pixel in the second black pixel run represent the contour profile features on the second level. Since not all of the character images were of the same size, the profile vectors were linearly rescaled in order to obtain 6 features from the left and right contour profiles and 5 features from the upper and lower profiles on the first level of the digit image. Finally, 4 features were extracted from the upper and the lower contour profiles of the second level.

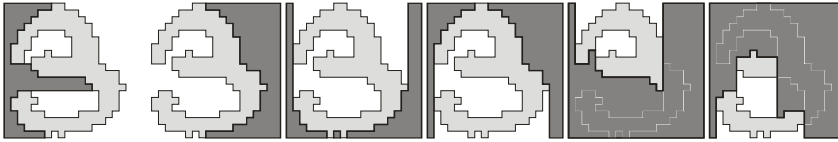


Fig. 2. Contour profiles of first and second level

The second group of 44 features (FS2) are extracted as pixel counts in rings zones around the gravity center of the image (see Fig. 3). We have used three rings, each divided in different number of equal zones. The outermost ring has a radius r equal to the distance from the gravity center to the furthest black pixel of the image. The first ring with radius $0.2 \cdot r$ provides 4 features and the second ring with radius $0.5 \cdot r$ provides 24 features. The last 16 features are provided from the outermost ring.

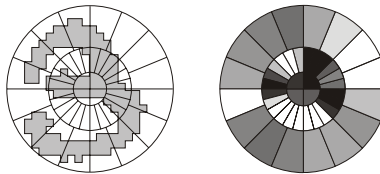


Fig. 3. Ring-zone features

The last group of 72 features (FS3) use Kirsch operator [4] to detect local directional information of the edges of the input pattern. Compared with chain code which also describes the edge direction, Kirsch edge detection is more robust even under noisy conditions.

The first black pixel which the scanning line intersects forms the first outermost periphery. The second black pixel which is the starting point of the second black pixel run forms the second outermost periphery (see Fig. 4). When the image is scanned in horizontal direction, the vertical and both diagonal Kirsch features are extracted at the outermost periphery. When the image is scanned in vertical direction, the horizontal and both diagonal Kirsch features are extracted at the outermost and second outermost periphery. This way, 3 Kirsch directional features are provided for each periphery pixel. The feature vectors are again linearly rescaled to 15 features coming from the left and right periphery each, 12 features coming from the first outermost top and bottom periphery each, and 9 features coming from the second outermost top and bottom peripheries.



Fig. 4. Kirsch features

Kirsch feature extraction is performed on the grayscale digit images using sub-pixel precision. All parameters including the number of features by projection, the radiuses of rings for zone-pattern regions and the number of features coming from the outermost peripheries for Kirsch features are carefully chosen after several iterations using observations about their discriminative power. The features were preprocessed for zero mean and unit variance.

4. The Recognition Results

Our experiments were performed on an extract of the well-known NIST (National Institute of Standards and Technology) handwritten digit database. This database consists of 7 partitions denoted as: hsf_0, ..., hsf_4, hsf_6 and hsf_7. Digit images from the hsf_0 partition were used for classifier training while the tuning of classifier parameters (kernel width σ and penalty C) was performed using the hsf_1 partition for validation. The final recognition rates were estimated on most difficult partition hsf_4. So, the samples in the test set belong to different writers from those in the learning set. In Fig. 5 a fragment of the NIST database is given.



Fig. 5. A fragment from the NIST database

We used SVMs with Gaussian kernel because it provided better recognition rates than linear, polynomial or sigmoidal kernel. Because of the large number of samples we have used SVMTorch that is a more robust variation of SVM training software library [5].

The decision fusion methods: Product, Dempster Rule, Fuzzy Integral, and Decision Templates require possibilistic outputs. To map the original outputs to $[0, 1]$ interval we used the mapping $1/(1+e^{-x})$.

The recognition rates of different classifier cooperation schemes applied on the above described 3 feature sets: FS1, FS2 and FS3 are given in Table 1. In the second column the corresponding cooperation scheme is given, followed by the rec-

ognition rate and the rank of the cooperation scheme when combining classifiers are trained using 1000, 2000, 5000, 10000, 30000 and all 53449 available samples.

The first 3 rows show recognition rates of each feature set individually. The row a) gives the recognition results of a single optimized SVM classifier applied on the three feature sets as a whole. The next row b) gives the recognition rate of a hypothetical cooperation scheme that knows to choose the right class if it is predicted by at least one of the member classifiers. This is the theoretical upper bound of the recognition rates achievable by classifier decision fusion.

The cooperation schemes 1-4 are voting schemes including variations of the Borda count that is a generalization of the majority vote [6]. The 5-12 cooperation schemes use various averages, the maximum of the sum, product, maximum and the minimum of the corresponding pairs of the classifier outputs respectively to make the final decision [7]. The Dempster Rule [8] and many variations [9] are given in rows 13-35. The naive Bayes cooperation scheme given in rows 36-37 uses the confusion matrices of member classifiers to estimate the certainty of the classifier decisions [10]. The fuzzy integration 38-39 is based on searching for the maximal grade of agreement between the objective evidence (provided by the sorted classifier outputs for class i) and the expectation (the fuzzy measure values of both classifiers) [11]. We have also used a variety of decision templates schemes 40-62 described elsewhere [12], [9]. The generalized committee prediction and its variations 63-67 are based on a weighted combination of the predictions of the member classifiers [13]. Cooperation scheme 76 uses linear regression to make decision fusion. In the cooperation scheme 69 the 4 individual SVM outputs (40 features) are input to another SVM classifier. This kind of cooperation is also known as classification task [8].

Table 1 shows that the cooperation 69 (svmcmb) has unbeatable recognition rate in all cases. However, this method is most complex because it needs additional classifier and additional samples for its training. Because of additional number of samples used in the training, this method sometimes outperforms even the "oracle method".

Increasing the number of training samples, indeed increase recognition rates of individual classifiers and their cooperation. On the other hand, increasing recognition rates of individual classifiers increase their correlation that reduces the possibility for improvement of cooperation recognition rates.

Voting cooperation schemes (1-4) are among worst because they use most limited information of member classifiers, ignoring useful information about second choices, reliability of the choice, distribution of the choices for different classes, etc.

The simplest cooperation schemes (5-12) as we expected, have average recognition rates and should be used in not demanding applications.

It is interesting that Dempster Rule and its variations (13-35) have in average better recognition rates than decision templates schemes (40-62).

The naive Bayes cooperation schemes (36-37) are relatively good choice while the fuzzy integration (38-39) shows weak results.

The generalized committee prediction and its variations (63-67), together with linear regression (68) are among the best methods and should be considered as serious candidates for implementation in any pattern recognition application based on classifier cooperation.

Table 1. Recognition rates (%) of combining SVM classifiers for 3 feature families (FS1, FS2, FS3) and different sizes of learning set (1000, 2000, 5000, 10000, 30000 and all 53449 samples). R stands for rank

	cooperation	1000	R	2000	R	5000	R	10000	R	30000	R	all	R
	FS1	87.3853		89.9857		91.5476		93.1931		94.8829		95.3398	
	FS2	86.5123		89.6003		90.3557		92.4087		94.7840		95.1454	
	FS3	89.9226		91.5203		92.4240		94.1002		95.7133		96.1003	
a)	Single svm	92.7514		93.9911		94.2758		95.9076		97.1899		97.3843	
b)	oracle	95.4387		96.3629		96.6664		97.5105		98.2505		98.4909	
1	vote	90.6012	64	92.4974	66	93.2493	67	94.7925	66	96.3084	60	96.6698	51
2	borda	91.0957	54	92.9850	55	93.6944	59	95.2017	49	96.5573	44	96.9597	6
3	bks	91.9500	27	93.1692	44	94.1445	8	94.9698	62	96.1532	65	96.4038	65
4	bksv	92.1120	14	93.3329	20	94.2861	5	95.1915	50	96.2964	61	96.5727	59
5	avg	91.5646	35	93.3585	18	94.0320	11	95.3944	9	96.6681	23	96.9461	8
6	prod	91.1264	53	92.9646	57	93.8581	43	95.3296	19	96.7670	7	96.9086	15
7	harm	90.9218	59	92.6645	65	93.6910	60	95.2171	46	96.6562	25	96.8182	40
8	cprod	90.5296	66	92.7889	62	93.7643	56	95.1386	55	96.6289	34	96.8898	24
9	maxmax	89.9567	68	92.3678	67	93.3226	65	94.6151	67	96.2402	62	96.5334	62
10	minmax	90.5143	67	92.1768	68	93.3209	66	94.9101	64	96.4942	49	96.5880	54
11	med	91.0804	55	93.0805	51	93.6910	60	95.1318	56	96.5692	41	96.8864	26
12	davg	91.5578	36	93.3636	17	94.0166	13	95.4183	7	96.6494	28	96.9222	12
13	demp	91.9278	28	93.1453	48	93.8853	40	95.1761	53	96.5795	38	96.8455	32
14	dempp1	91.3191	45	93.2050	40	93.8785	41	95.2921	29	96.6426	29	96.8352	36
15	dempp2	91.2441	49	93.2272	37	93.9075	35	95.2733	33	96.6698	22	96.8932	22
16	dempp3	90.6029	63	92.9066	60	93.8103	53	95.2648	34	96.6920	19	96.9512	7
17	dempp4	91.3344	44	93.2374	35	94.0013	16	95.3245	20	96.7500	9	96.9086	15
18	dempi1	91.3651	42	93.2561	34	93.9399	31	95.3552	15	96.7040	17	96.8949	21
19	dempi2	91.1793	51	93.1658	45	93.8188	52	95.2034	48	96.5914	37	96.7619	45
20	dempc	90.7581	60	93.0362	54	93.8103	53	95.2921	29	96.7398	11	96.9324	9
21	dempmk	91.2219	50	93.1914	42	93.8905	38	95.2409	44	96.6409	30	96.8728	27
22	dempch	91.1673	52	93.1334	49	93.9570	28	95.2648	34	96.7346	12	96.9239	11
23	dempas	92.1785	11	93.3005	26	93.9007	36	95.3688	12	96.7739	4	97.0143	4
24	dempchi	91.3992	41	93.2647	30	93.8888	39	95.3006	26	96.6716	21	96.8438	33
25	dempchi2	91.4777	40	93.3482	19	93.9979	18	95.3620	14	96.7142	14	96.8983	20
26	dempbc	91.2833	46	93.2050	40	93.8734	42	95.3023	25	96.6528	27	96.8335	38
27	dempbl	91.6175	33	93.4062	16	93.9859	22	95.3910	10	96.6835	20	96.8932	22
28	dempchr	92.2075	8	93.4505	11	93.9467	29	95.3688	12	96.7517	8	97.0313	3
29	dempchr2	91.6448	32	93.4386	12	94.0814	10	95.4115	8	96.7193	13	96.9069	17
30	dempjac	91.2458	48	93.1538	47	93.8257	51	95.2478	39	96.6340	31	96.8216	39
31	demppe	91.3617	43	93.2306	36	93.8922	37	95.3211	21	96.6630	24	96.8352	36
32	dempse	89.8339	69	93.5869	8	93.9996	17	95.1830	51	96.4038	56	96.6903	49
33	dempfr	92.0148	20	93.7080	7	93.9945	20	95.1778	52	96.3783	58	96.5692	60
34	dempm	91.6005	34	93.1555	46	93.8581	43	94.9937	61	96.5744	40	96.7466	46
35	dempmc	92.4019	5	93.936	5	94.1002	9	95.2085	47	96.2350	63	96.4107	64

Table 1. (continues)

	cooperation	1000	R	2000	R	5000	R	10000	R	30000	R	all	R
36	pprod	91.2594	47	93.0498	53	93.9126	34	95.3705	11	96.7432	10	96.9188	13
37	bayes	91.5322	38	93.1231	50	93.7387	58	95.0789	59	96.3646	59	96.7892	41
38	fi	90.6797	61	92.9134	59	93.5187	62	94.9647	63	96.4226	51	96.6698	51
39	fic	90.5688	65	92.8145	61	93.4488	63	94.9084	65	96.4840	50	96.7142	47
40	dtp1	92.1001	15	93.3141	21	93.8478	45	95.2426	40	96.4158	52	96.5863	55
41	dtp2	92.1580	12	93.4096	14	93.9314	32	95.2614	36	96.5215	46	96.7841	42
42	dtp3	91.7215	30	93.4215	13	94.0303	12	95.3501	18	96.5761	39	96.8728	27
43	dti1	92.0745	19	93.4829	10	93.9894	21	95.2563	38	96.5965	35	96.8694	29
44	dti2	91.5442	37	92.7293	63	93.4369	64	95.1165	57	96.0986	67	95.9008	68
45	dti3	90.9252	58	93.0771	52	93.8427	49	95.2972	27	96.5931	36	96.8540	31
46	dte	91.9892	23	93.2834	28	93.9723	24	95.2938	28	96.6545	26	96.9069	17
47	dtmnk	92.1580	12	93.4096	14	93.9314	32	95.2614	36	96.5215	46	96.7841	42
48	dtch	90.9713	57	92.0813	69	92.9646	68	94.3440	69	96.1072	66	96.4414	63
49	dtcan	92.0012	21	92.7241	64	92.8861	69	94.3645	68	94.6237	69	94.0337	69
50	dtas	91.9585	25	93.2613	32	93.9689	25	95.3518	16	96.6323	32	96.8438	33
51	dtchi	92.2314	6	93.2988	27	93.9962	19	95.3091	23	96.5658	42	96.7057	48
52	dtchi2	92.2160	7	93.3039	25	94.0047	15	95.3108	22	96.5624	43	96.7636	44
53	dtbc	92.1001	15	93.3141	21	93.8478	45	95.2426	40	96.4158	52	96.5863	55
54	dthl	92.1819	10	93.1879	43	93.9450	30	95.2836	31	96.5334	45	96.6392	53
55	dtchr	91.9585	25	93.2613	32	93.9689	25	95.3518	16	96.6323	32	96.8438	33
56	dtchr2	92.1904	9	93.2186	39	93.9621	27	95.2836	31	96.5215	46	96.6869	50
57	dtjac	92.1001	15	93.3141	21	93.8478	45	95.2426	40	96.4158	52	96.5863	55
58	dtper	92.1001	15	93.3141	21	93.8478	45	95.2426	40	96.4158	52	96.5863	55
59	dtse	91.7164	31	93.2630	31	93.7933	55	95.0022	60	96.2350	63	96.2896	66
60	dtfr	92.0012	21	93.7319	6	94.0166	13	95.1540	54	96.3919	57	96.5624	61
61	dtm	93.0686	3	94.1786	3	94.3594	3	95.6451	2	96.7687	6	96.8898	24
62	dtmc	93.3636	2	94.2724	2	94.4208	2	95.1165	57	95.9520	68	96.1839	67
63	epw	90.6353	62	92.9424	58	93.8376	50	95.2324	45	96.7091	15	96.9171	14
64	gc	91.0378	56	92.9663	56	93.7626	57	95.3074	24	96.6988	18	96.8574	30
65	mgc	91.7352	29	93.2766	29	93.9859	22	95.4609	6	96.7091	15	96.9307	10
66	ogc	91.9602	24	93.5784	9	94.1599	7	95.5342	4	96.8301	2	97.0723	2
67	omgc	91.4964	39	93.2203	38	94.2622	6	95.4728	5	96.7773	3	97.0075	5
68	mlr	92.9458	4	94.1258	4	94.3389	4	95.6280	3	96.7739	4	96.9052	19
69	svmcmb	97.1814	1	97.2888	1	97.2990	1	97.3792	1	97.7765	1	97.8788	1

5. Conclusion

In this paper, the cooperation of three feature families for handwritten digit recognition using SVM classifiers is examined. We investigate an extensive number of cooperation schemes based on classifier decision fusion.

The presented results show that it is difficult to achieve the recognition rate of a single SVM applied on the feature set that includes all feature families by

combining the individual SVM decisions. In our experiments only one of the cooperation schemes exceeded the recognition rate of a single SVM classifier. These results impose the crucial question: whether the methods for classifier cooperation are still needed [14] or pattern recognition tasks could be better solved by a single, well-optimized SVM classifier. However, the classifier cooperation reduces the classifier complexity, need for samples, and sometimes can increase the classifier performance.

References

1. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition, *Knowledge Discovery and Data Mining*, Vol. 2 (1998) 1–47
2. LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Denker, J. S., Drucker, I. Guyon, H., Muller, U. A., Sackinger, E., Simard, P., Vapnik, V.: Comparison of learning algorithms for handwritten digit recognition, In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, Paris (1995) 53–60
3. Gorgevik, D., Cakmakov, D.: An Efficient Three-Stage Classifier for Handwritten Digit Recognition, *Proc. of 16th Int. Conference on Pattern Recognition*, Vol. 4, IEEE Computer Society, Cambridge, UK (2004) 507–510
4. Pratt, W. K.: *Digital Image Processing: PIKS Inside*, Third Ed., John Wiley & Sons (2001)
5. Collobert, R., Bengio, S., Mariéthoz, J.: Torch: a modular machine learning software library, *Technical Report IDIAP-RR 02-46*, Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP), CH-1920 Martigny, Switzerland (2002) (www.torch.ch)
6. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision Combination in Multiple Classifier Systems, *IEEE Tran. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1 (1994) 66–75
7. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3 (1998) 226–239
8. Schürmann, J.: *Pattern Classification: A Unified View of Statistical and Neural Approaches*, John Wiley & Sons, Inc. (1996)
9. Gorgevik, D.: Classifier Combining for Handwritten Digit Recognition, *Ph.D. dissertation*, Faculty of Electrical Engineering, Skopje, Macedonia (2004).
10. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their application to handwritten recognition, *IEEE Transactions on System, Man and Cybernetics*, Vol. 22 (1992) 418–435
11. Cho, S.B., Kim, J.H.: Combining multiple neural networks by fuzzy integral and robust classification, *IEEE Tran. on System, Man and Cyber.*, Vol. 20, No. 3 (1995) 380–384.
12. Kuncheva, L.I., Bezdek, J.C., Duin, P.W.: Decision templates for multiple classifier fusion: an experimental comparison, *Pattern Recognition*, Vol. 34, No. 2 (2001) 299–314.
13. Bishop, C.M.: *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford (1995).
14. Kittler, J.: A Framework for Classifier Fusion - Is It Still Needed, in F. J. Ferri, J. M. Inesta, A. Amin and P. Pudil, Eds., *Advances in Pattern Recognition, Lecture Notes in Computer Science*, Vol. 1876, Springer-Verlag (2000) 45–56