Осма национална конференција со меѓународно учество - ЕТАИ 2007
Eighth National Conference with International Participation - ETAI 2007
Охрид, Република Македонија - Ohrid, Republic of Macedonia, 19 - 21. IX 2007

I6-3

СТЕБЛА ОД МАШИНИ СО НОСЕЧКИ ВЕКТОРИ ЗА ПРЕПОЗНАВАЊЕ НА ПРИМЕРОЦИ

Ѓорѓи Маџаров, Дејан Ѓорѓевиќ, Елена Икономовска

Факултет за Електротехничка и информациски технологии – Скопје, Карпош II бб,
П. Фах 574, 1000 Скопје, madzarovgore@yahoo.com, {dejan, elenai}@feit.ukim.edu.mk

***Апстракт* – Во овој труд е претставена архитектура на класификатори наречена Бинарно Дрво од Машини со Носечки Вектори (БД - МНВ), или BTS со која се овозможува решавање на повеќекласни проблеми со помош на бинарни класификатори распоредени во бинарно дрво. За тестирање на ефикасноста на оваа архитектура, извршена е серија од споредби со SVM класификатори обучени со стратегиите еден-против-сите и еден-против-еден (со мнозинско гласање и DAG методот). Перформансите на класификационите шеми се споредувани при решавање на проблемот на препознавање на ракописни знаци. Експериментите беа извршени со примероци од MNIST (Mini-National Institute for Standards and Technology) базата на сегментирани знаци и Statlog базата.**

***Клучни зборови* – Бинарно Дрво на машините со носечки вектори (BTS), машини со носечки вектори (SVM), повеќекласна класификација.**

1. ВОВЕД

SV машините претставуваат обучен систем кој користи простор на хипотези со линеарни функции во многудимензионален простор на обележја. Системот е обучен со алгоритам за обука според теоријата на оптимизација, користејќи принципи изведени од теоријата за статистичко учење[3]. 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.

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 повеќекласни проблеми со SVM постојат два пристапа: повеќекласен SVM кој целиот проблем на класификација го сведува на еден голем оптимизациски проблем [11] и комбинирање на повеќе бинарни класификатори. Постојат неколку пристапи за комбинирање на повеќе бинарни класификатори за решавање на повеќекласен проблем:

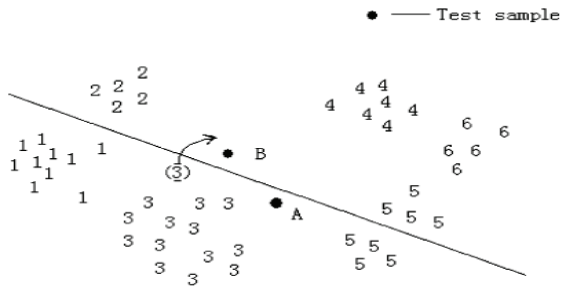
1. Еден-против-сите: Секоја класа се обучува во однос на преостанатите N-1 класи. На тој начин се добиваат N бинарни класификатори кои внесуваат голема нелинеарност во оригиналниот проблем. При препознавањето примерокот се проследува до сите N класификатори а за припадноста на примерокот се одлучува според класификаторот со максимален излез.
2. Еден-против-еден: На овој начин се врши обучување на N(N-1)/2 бинарни класификатори и секој класификатор разделува по еден пар од класи. Примерокот се проследува до сите N(N-1)/2 класификатори, а конечната одлука за неговата припадност, се носи по пат на мнозинско гласање [6].
3. Directed acyclic graph SVM (DAGSVM). Во овој случај се обучуваат вкупно N(N-1)/2 класификатори. Одлуката директно зависи од поставеноста на ацикличниот граф. Кога примерокот ќе дојде до листот од графот се донесува одлука за неговата припадност. Бројот на проверки за секој примерок е N-1 [9].
4. Error correcting output codes (ECOC) machine. За да се донесе одлука за припадноста на примерокот, во овој случај потребни се најмалку N односно најмногу N(N-1)/2 тестирања. Кај ECOC машините бројот на тестови е во зависност од кодирачката шема[4].

Архитектурата на дрво е често споменувана тема во областа на теорија на одлучување. Машините со носечки вектори (SVM) во комбинација со бинарна архитектура на дрво (Binary Tree Architecture) или SVM-BTA [1] се воведуваат со цел да се намали бројот на бинарните класификатори и да допринесе за побрза одлука. BTS архитектурата целосно е базирана на бинарната класификација.

2. АЛГОРИТАМОТ НА BTS

2.1. BTS

На Сл. 1 е претставен повеќекласен проблем.



Сл. 1 – Повеќекласен проблем со шест класи

Ако за обучување на првиот класификатор се користат сите примероци од класата 1 и класата 2 ја добиваме делбената рамнина претставена на сликата (или уште наречена бинарна одлучувачка функција *f(x)* , или бинарен класификатор). За примероците на класата 1 вредностите на одлучувачката функција се -1 (sgn(*f(x)*)=-1), а за примероците на класата 2 се 1 (sgn(*f(x)*)=1). За дадениот пример оваа делбена рамнина истовремено ги дели примероците на класата 3 од оние на класите 4 и 6 (се разбира оваа поделба не е оптимизирана). Класата 5 е распоредена од двете страни на делбената рамнина. Во фазата на тестирање, кога влезниот примерок се наоѓа на позиција A (sgn(*f(x)*)=-1), се забележува дека најверојатно примерокот не припаѓа на класите 4 и 6. Со други зборови кажано не се јавува потреба за обучување на класификатори за класите 3 и 4 или класите 3 и 6. Тоа го скратува времето за обучување. За таа цел се применува архитектурата наречена бинарно дрво на машините со носечки вектори (BTS) [5]. Исто така на Сл. 1 може да се види дека класата 3 е потполно придружена на подјазелот 0, иако постојат примероци кои се наоѓаат премногу блиску до делбената рамнина (т.е. нивната постериорна веројатност е помала од предвиденот праг). Во ваков случај овие примероци се придружуваат на подјазелот 1, со што класата 3 добива шанса да учествува во обучувањето на класификаторите на подјазелот 1 со останатите класи. Класите 4 и 6 се наоѓаат далеку од делбената рамнина па поради тоа не е потребно нивно прегрупирање.

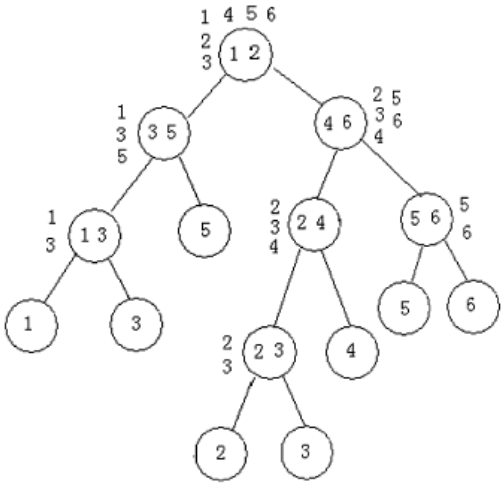
Доколку горенаведеното се примени, кога тестирачкиот примерок се наоѓа на позиција B од коренот, ќе биде класифициран и придружен на подјазелот 1. Со прегрупирањето тој би можел да биде класифициран во класата 3. Во спротивно уште кај коренот би ја изгубил можноста да биде правилно класифициран.

На почетокот на алгоритамот сите примероци се носат на коренот на дрвото, од каде случајно избираме еден пар од класи (класа *i* и класа *j*), со кој ќе обучуваме. Доколку, претходно немаме обучено бинарен класификатор за овој пар од класи, ги земаме сите примероци кои припаѓаат на овие класи и започнуваме со обучувањето. Вака обучениот класификатор го зачувуваме во листа наречена Trained_list која ги содржи сите обучени бинарни класификатори до тој момент. Со обучувањето ја добиваме делбената рамнина помеѓу класите *i* и *j*. Во случај претходно да имаме обучено бинарен класификатор за овој пар од класи тој веќе постои во Trained_list. Потоа креираме два необучени подјазли (подјазел 0 и подјазел 1) на кои соодветно се придружуваат класните примероци на овие две класи. Сите примероци од останатите класи ги носиме на тековниот јазел за да се изврши нивна поделба врз основа на претходно добиената делбена рамнина. При тестирањето на примероците, доколку излезот на функцијата sgn($f_K(x)$) е еднаков со оној на класата *i*, примерокот го придружуваме на подјазелот 0, додека ако е еднаков со излезот на класата *j* го придружуваме на подјазелот 1. Доколку сите примероци од конкретна класа во тековниот јазел (освен примероците од двете класи кои беа искористени во процесот на обучување) целосно се придружени на дете јазелот 0 (дете јазелот 1), но притоа некои од нив не го задоволуваат условот растојанието помеѓу нив и делбената рамнина да е поголем од δ (праг со кој се означува растојанието меѓу примероците и делбената рамнина на јазелот), правиме нивно прегрупирање така што ќе ги придружиме на дете јазелот 1 (дете јазелот 0). Целта на овој чекор е да се овозможи на оваа класа да учествува во процесот на обучување на бинарните класификатори и во другото подстебло. Доколку сите примероци го задоволуваат условот не се прават прегрупирања. Во случај сите преостанати класи од тековниот јазел да се поделени на двата негови подјазли се бришат подјазлите и се селектира нов пар на класи во тековниот јазел. Доколку успешно најдеме две класи кои не сме ги евалуирале, тогаш ја повторуваме претходно опишаната постапка. Доколку пак ги имаме евалуирано сите парови од класи, ја имаме најлошата ситуација. Во ваков случај ги бришеме креираните подјазли, го зачувуваме моделот на овој пар од класи за соодветниот јазел и овде застануваме. Овој процес се нарекува процес на градење на BTS.

Процесот на класификација во BTS започнува од самиот корен. Тестирачките примероци се

класифицираат во подјазел 0 или подјазел 1 од страна на јазелот родител (класификатор добиен од соодветниот пар од класи за тој јазел), се додека не се дојде до терминален јазел во оваа архитектура на дрво. Доколку терминалниот јазел содржи повеќе од две класи (најлошата ситуација) конечната одлука се донесува со мнозинско гласање како во еден-против-еден алгоритамот.

По завршување на градењето на BTS, секој терминален јазел содржи примероци кои припаѓаат на една или повеќе од две класи, додека една класа може да се најде во повеќе терминални јазли. Сл. 2 е една од можните варијанти на BTS на шесткласниот проблем од Сл. 1.



Сл. 1 – Илустрација на BTS со примероци за обучување дадени на слика 1.

Круговите на Сл. 2 претставуваат јазли на BTS додека парот на класи во јазелот го претставува прифатениот пар, кој учествува во процесот на обучување на бинарниот класификатор. Се забележува дека класата 3 и класата 5 се поделени во два јазли во првиот чекор. Класата 3 е поделена како резултат на прегрупирањето додека класата 5 како резултат на одлучувачката функција на класите 1 и 2.

2.2. Веројатносниот модел и δ

Прагот δ чија вредност ја избираме е глобална за сите класификатори. Со зголемување на прагот се зголемува и прецизноста на алгоритамот, но како последица на тоа се јавува зголемување на бројот на бинарните класификатори, како и на времето на обучување.

3. АНАЛИЗА НА ПЕРФОРМАНСИТЕ

Во овој дел ќе биде извршена куса анализа на перформансите на BTS. Анализата е направена за случаите кои покажуваат најдобри и најлоши перформанси.

Во најдобра ситуација, потребни ни се N-1 обучувања на бинарни класификатори, затоа што примероците не се прегрупирани, односно нема класи кои се поделени во различни подјазли. Тоа значи дека во секој јазел постои совршена класификација и ниту една од класите при распределувањето во јазлите не се појавува во обата креирани подјазли.

Во најлоша ситуација имаме обучување на N(N─1)/2 бинарни класификатори. Оваа ситуација ја имаме кога класификаторите добиени од секој пар на класи ги распределуваат останатите класи на тој начин што секоја од класите се појавува во двата подјазли на соодветниот јазел каде е извршена класификацијата. Претпоставувајќи дека еден од јазлите содржи N_0 класи, неговите подјазли по извршената класификација ќе содржат N_0-1 класи. Ваквиот модел кој сме го добиле со обучување со еден пар од класи го зачувуваме во листа (Trained_list) така што доколку се појави повторна потреба од користење на истиот модел не го обучуваме повторно. Доколку бројот на нивоа во BTS е N бројот на терминални јазли изнесува 2^{N-1}, затоа што секој терминален јазел содржи една единствена класа. Од ова следува дека бројот на јазлите експоненцијално се зголемува и ваквата ситуација е наречена „најлоша“. Во ваков случај класификацијата може да биде направена по донесување на N-1 бинарни одлуки.

За да се надмине овој проблем мора да се усвојат некои техники на разрешување на оваа ситуација. Кога јазелот за одреден пар ќе дојде во најлоша ситуација го зачувуваме обучениот пар и класификаторот за овој пар (во Trained_list). Потоа ги бришеме двата добиени подјазли и земаме друг пар на класи за обучување. Доколку повторно наидеме на најлошата ситуација тогаш овој чекор го повторуваме се додека:

1) не најдеме пар кој не нѐ води до најлоша ситуација

2) не ги евалуираме сите парови од класи.

Во вториот случај овде завршуваме и понатаму не креираме други подјазли. Во фазата на класифицирање кога примерокот кој се тестира ќе дојде до терминален јазел последната негова класификација се прави по методот на мнозинско гласање. Како резултат на ова се добива неекспоненцијално растење на бројот на јазлите кај BTS.

4. ЕКСТРАКЦИЈА НА ОБЕЛЕЖЈА

Експериментите со класификаторите (еден-против-сите, еден-против-еден, DAG и BTS) беа спроведени на MNIST базата од ракописни цифри и Statlog базата [10] од ракописни букви. Примероците од Statlog базата се дадени со веќе екстрахирани обележја и како такви беа користени во експериментите. За разлика од примероците од Statlog базата, примероците од MNIST базата се дадени како слики. По

првичната обработка за отстранување на шумот и наоѓање на рамката околу знакот на овие слики, се врши проценка на аголот на закосеност на знакот и истиот се исправа [8]. Потоа од знакот се екстрахираат неколку видови на обележја. За потребите на експериментите беа екстрахирани три фамилии на обележја [7]. Од секоја фамилија се формира по едно множество обележја кое ја опишува цифрата која треба да се препознае. Трите фамилии на обележја кои се екстрахираат од секоја слика се:

1. Проекциони хистограми
2. Прстенести зони
3. Kirsch обележја

4.1. Проекциони хистограми

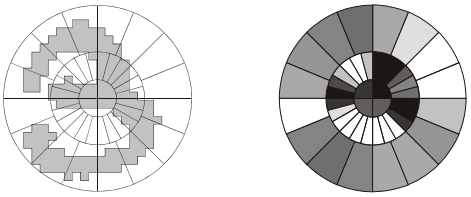
Проекционите хистограми претставуваат пресликување од слика (дводимензионална функција) во хистограм (еднодимензионална дискретна функција) чии вредности претставуваат сума на вредностите на пикселите долж одреден правец. Хистограмот може да се претстави како вектор од реални вредности. Во зависност од избраниот правец се дефинираат:

- хоризонтален проекцион хистограм
- вертикален проекцион хистограм
- кос проекцион хистограм

Проекцион хистограм се добива со сумирање на вредностите на пикселите во еден правец од матрицата со сликата на знакот.

4.2. Прстенести зони

Централните прстенести зони претставуваат зонски обележја. Кај зонските обележја сликата на знакот се дели на одреден број зони и во секоја од нив се испитуваат одредени особини на сликата. Типично за секоја зона се генерира едно или неколку обележја. Кај предложените централни прстенести проекции обележјето кое се генерира за секоја од зоните е релативната пополнетост на подрачјето на зоната со „мастило“, односно бројот на црни пиксели. Зоните се дефинирани како концентрични прстени околу тежиштето на сликата на знакот, и секој прстен е поделен на одреден број еднакви исечоци (Сл. 3).



Сл. 2 – Централни прстенести зони: лево – дефинирање на зоните; десно – пополнетост на секоја од зоните како обележје

4.3. Kirsch обележја

Kirsch обележјата се дирекциони обележја. Тие се екстрахираат од сликата на цифрата со помош на Kirsch операторот и претставуваат локални детектори на правецот во секој пиксел од работ на сликата.

5. ЕКСПЕРИМЕНТИ

Во овој дел се прикажани резултатите од експериментите кои беа извршени за повеќекласните проблеми на препознавање на ракописни знаци. За класификација беа користени програми за SVM класификатори имплементирани во C++ користејќи ја TORCH библиотеката [2]. Сите тестови беа изведени на персонален компјутер со Intel Core 2 Duo процесор на 1.86 GHz под оперативниот систем Windows XP. За опис на цифрите од MNIST базата во првиот случај беа користени 40 обележја додека во вториот случај беа издвоени 292 обележја што резултираше во помал процент на погрешно препознаени примероци. Четириесетте обележја кои беа издвоени во првиот случај се состоеја од 10 хоризонтални проекции, 8 вертикални проекции, 11 проекции во правец на главната дијагонала и уште 11 проекции во правец на споредната дијагонала. SVM класификаторите беа обучени користејќи го целото множество за обука (60000 примероци), додека тестирањето беше направено на 10000 примероци. Во табела 1 се прикажани резултатите од спроведените експерименти.

Табела 1. Перформанси и ефикасност на класификаторите за препознавање на цифри со издвоени 40 обележја

Класи-фикатор	σ, C	Гре-шка (%)	Време (s)	
			тести рање	обучу-вање
OvA	1.4, 100	1.90	19.78	411.28
OvO	0.8, 100	2.24	34.16	98.25
DAG	0.8, 100	2.27	6.46	98.25
BTS	0.8, 100	2.24	34.79	96.33

Првата колона го опишува методот на комбинирање на SVM класификаторите: еден-против-сите (OvA), еден-против-еден (OvO), DAG и BTS. Во втората колона се дадени параметрите на Гаусовиот кернел (σ) и казната (C) користени при обуката на SVM класификаторите. Последните три колони ги прикажуваат ратите на грешка, времето на обучување и времето на тестирање кај секоја од методите.

Грешката и времето на тестирање се однесуваат за препознавање на сите 10000 примероци од множеството за тестирање не сметајќи го времето потребно за претпроцесирање (проценка и отстранување на закосеноста) и екстракција на обележјата.

Од резултатите се гледа дека најмала грешка се прави при комбинирањето на класификаторите во еден-против-сите шема. Шемите кои користат еден-против-еден обучени класификатори (Ovo, DAG и BTS) имаат приближно иста точност, но секоја од нив е околу четири пати побрза при обуката од обуката на десетте еден-против-сите класификатори иако при ова се обучуваат и до 45 класификатори. DAG методата постигна најмала точност но најголема брзина при препознавањето.

Во вториот случај експериментите кои беа направени се однесуваат на истите множества од примероци со таа разлика што бројот на обележја кои ги издвоивме го зголемивме на 292. Овие 292 обележја се состоеја од 40 проекциони хистограми, 44 обележја од прстенести зони и 208 Kircsh обележја. Ратите на грешка, времето на обучување и времето на тестирање кај секоја од методите се прикажани во табела 2.

Табела 2. Перформанси и ефикасност на класификаторите за препознавање на цифри со издвоени 292 обележја

Класи-фикатор	σ, C	Гре-шка (%)	Време (s)	
			тести-рање	обучу-вање
OvA	3.0, 100	0.82	96.47	385.38
OvO	3.0, 100	0.85	132.05	171.73
DAG	3.0, 100	0.88	26.34	171.73
BTS	3.0, 100	0.85	132.78	170.57

Од табелата може да се забележи дека методите во поглед на грешката даваат резултати кои се приближно еднакви (±3 погрешно препознаени примероци од 10000). Најмала грешка покажува методата еден-против-сите 0.82%, додека DAG методата е најбрза при тестирањето но прави и најмногу грешки. Времето за препознавање со DAG методата е 4 до 5 пати покусо од времињата на останатите методи. Времето потребно за обука на класификаторите кај методите кои користат еден-против-еден класификатори е околу два пати покусо од обуката кај еден-против-сите методата.

Во обата случаја на препознавање на цифри BTS методата покажа перформанси многу слични на еден-против-еден пристапот со гласање. При ова дојде до креирање на релативно големо BTS стебло со голем број на несовршени поделби каде мораше да се користи гласање на еден-против-еден класификаторите.

Примероците кои ги користевме во експериментите за препознавање на ракописни букви се дел од Statlog базата[10]. Притоа множеството на примероци со кои беа обучувани класификаторите се состоеше од 15000 примероци, додека множество за тестирање се броеше 5000 примероци. За опис на овие примероци се користени 16 веќе екстрахирани обележја од сликите на буквите кои беа нормализирани на нулова средна вредност и единична варијанса. Во табелата 3 се прикажани резултатите од спроведените експерименти на овој повеќекласен проблем.

Табела 3. Перформанси и ефикасност на класификаторите за буквите од англиската азбука

Класи-фикатор	σ, C	Гре-шка (%)	Време (s)	
			тести-рање	обучу-вање
OvA	1.1, 100	3.20	119.52	554.22
OvO	1.1, 100	4.72	160.58	80.97
DAG	1.1, 100	4.74	12.55	80.97
BTS	1.1, 100	4.70	39.76	67.11

И во овие експерименти најмала грешка се постигнува со методата еден-против-сите 3,20%. Останатите три методи постигнуваат многу слична точност при препознавањето која е околу 1.5% послаба од еден-против-сите методата. Повторно најбрза при препознавањето е DAG методата која е скоро 10 пати побрза од еден-против-сите методата. Времето потребно за обука на 26-те еден-против-сите класификатори беше скоро 7 пати подолго од обуката на сите 325 еден-против-еден класификатори. BTS методата постигна најмала грешка од методите кои користат еден-против-еден класификатори со време за препознавање кое е три пати подолго од DAG методата но се уште 3 пати покусо од еден-против-сите методата. Најбрза беше обуката на BTS класификаторот кој при градењето на стеблото обучи 287 (од можните 325) класификатори. Од добиените резултати е гледа дека предностите на BTS методата стануваат се поевидентни кога таа се применува на проблеми со поголем број на класи.

6. ЗАКЛУЧОК

Во овој труд е детално претставена BTS методата за употреба на SVM класификатори за решавање на повеќекласен класификациски проблем. Истата е споредена со други пошироко користени шеми за решавање на повеќекласни проблеми со помош на SVM класификатори: еден-против-сите и еден-против-еден со гласање и во DAG структура. Споредувањето на перформансите беше извршено при решавање на проблемот на препознавање на ракописни знаци (цифри и букви) со примероци од MNIST и statlog базите.

Добиените резултати покажаа дека кај проблемите со помал број класи, BTS методата има слични перформанси (точност на препознавање, брзина на обучување и брзина на препознавање) како и еден-против-еден методата со гласање. Кај проблемите со помал број на класи најдобра точност постигнува еден-против-сите методата која е и најбавна, додека најбрза е DAG методата.

Кај проблемите со поголем број на класи повторно најдобра точност постигна еден-против-

сите методата, но времето за препознавање а особено времето за нејзина обука е неколкукратно подолго од BTS и DAG методите. Кај проблемите со поголем број на класи BTS методата постигна најдобри перформанси во брзината на обучувањето притоа задржувајќи ја или надминувајќи ја точноста на другите еден-против-еден класификатори. Генерално, BTS методата е особено применлива при решавањето на повеќекласни проблеми со релативно голем број на класи за кои обуката на SVM класификаторите според некоја од другите методи може да стане недозволиво долго.

7. ЛИТЕРАТУРА

[1] S. Cheong, S. H. Oh, S.-Y. Lee: Support vector machines with binary tree architecture for multi-class classification, *Neural Info. Process. Lett. Rev.*, vol. 2, no. 3, Mar. 2004.

[2] Ronan Collobert, Samy Bengio, Johnny Mariéthoz: Torch: a modular machine learning software library, Technical Report IDIAP-RR 02-46, IDIAP, 2002.

[3] C. Cortes, Vladimir Vapnik: Support vector networks, *Machine Learning*, Vol. 20, pp. 273-297, 1995.

[4] T. G. Dietterich, G. Bakiri: Solving multiclass learning problems via error-correcting output codes, *J. Artific. Intell. Res.*, vol. 2, pp. 263–286,1995.

[5] Ben Fei, Jinbai Liu: Binary Tree of SVM: A New Fast Multiclass Training and Classification Algorithm, Trans. on Neural Networks, Vol. 17, No. 3, pp. 696-703, 2006.

[6] Jerome H. Friedman: Another approach to polychotomous classification, Technical report, Department of Statistics, Stanford University, 1997.

[7] Dejan Gorgevik, Dusan Cakmakov: An Efficient Three-Stage Classifier for Handwritten Digit Recognition, *Proc. of 17th Int. Conference on Pattern Recognition*, ICPR2004, Vol. 4, pp. 507-510, IEEE Computer Society, Cambridge, UK, 23-26 August 2004.

[8] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Inc., New Jersey, 1989.

[9] J. Platt, N. Cristianini, J. Shawe-Taylor: Large margin DAGSVM's for multiclass classification, *Advances in Neural Information Processing System*, vol. 12, pp. 547–553, 2000.

[10] Statlog Data Set, ftp://ftp.ncc.up.pt/pub/statlog/ [Online]

[11] J. Weston, C. Watkins: Multi-class support vector machines, *Proc. ESANN99*, M. Verleysen, Ed.. Brussels, Belgium, 1999.

- -

Summary

TREE OF SVM FOR PATTERN RECOGNITION

Gjorgji Madzarov, Dejan Gjorgjevik, Elena Ikonomovska

Faculty of Electrical Engineering and Information Technologies – Skopje, Karpoš II bb, PO Box 574, 1000 Skopje, madzarovgore@yahoo.com, {dejan, elenai}@feit.ukim.edu.mk

***Abstract* – This paper presents architecture of classifiers named Binary Tree of Support Vector Machines (BTS) for solving multi-class classification problems. In order to test the efficiency of this architecture, a series of comparisons with SVM classifiers trained with different strategies (one-against-all, one-against-one using voting and DAG) has been performed. The performance of BTS was measured on the problem of recognition of handwritten digits and letters. The experiments were conducted with samples from MNIST and Statlog base of segmented digits and letters.**

***Keywords* - Binary tree of support vector machine (BTS), multi-class classification, support vector machine (SVM).**