Universitatea „Lucian Blaga” din Sibiu
Facultatea de inginerie “Hermann Oberth”
Catedra de Calculatoare şi automatizări

Dezvoltarea unei ontologii de domeniu

(Support Vector Machine versus Bayes Naive)

Referat de doctorat nr. 2

Autor:
mat. Radu CREȚULESCU

Coordonator:
Prof. univ. dr. Ing. Lucian N. VINȚAN

SIBIU, 2009

CUPRINS

1 Clasificarea documentelor text 3

1.1 Seturile de date utilizate în experimente 4

1.2 Alegerea documentelor pentru antrenare - testare 5

1.3 Tipuri de reprezentare a datelor 6

2 Evaluarea clasificatorilor de tip SVM 10

2.1 Problema limitării metaclasificatorului cu clasificatori de tip SVM 11

2.2 O primă tatonare a problemei 12

2.3 Soluții pentru îmbunătățirea metaclasificatorului folosind clasificatoare de tip SVM 14

2.3.1 Soluția 1 – introducerea unor noi clasificatori SVM 14

2.3.2 Soluția 2 14

3 Clasificatorul Naïve Bayes 15

3.1 Clasificarea Bayes 15

3.1.1 Antrenarea clasificatorului Bayes 17

3.1.2 Testarea clasificatorului 19

3.2 Rezultate obținute cu clasificatorului Bayes 20

3.3 Adaptarea clasificatorului Bayes pentru utilizarea în metaclasificator 22

4 Compararea clasificatorului Bayes adaptat (BNA) cu clasificatorii de tip SVM 25

4.1 Antrenarea clasificatorilor pe setul A1 şi testarea pe setul T1 25

4.2 Antrenarea pe setul A1 şi testarea pe setul T2 27

4.3 Antrenarea şi testarea pe setul T2 28

5 Metaclasificatori 29

5.1 Selecția bazată pe vot majoritar 30

5.2 Selecția pe baza distanței euclidiene (SBED) 30

5.3 Selectarea bazată pe cosinus (SBCOS) 32

5.4 Rezultate obținute modificând alegerea clasei 33

6 Concluzii 36

7 Bibliografie 38

1 Clasificarea documentelor text

Cele mai multe colecții de date din lumea reală sunt în format text. Datele astfel memorate sunt considerate ca fiind semistructurate sau nestructurate deoarece, comparativ cu cele din baze de date nu au o structură completă. Tehnici de recunoaştere a informațiilor, cum ar fi metodele de indexare a textului, au fost dezvoltate pentru a manevra documente nestructurate.

Tehnicile tradiționale de recunoaştere a informațiilor, folosite în cazul datelor structurate cum ar fi bazele de date, devin inadecvate pentru căutarea în aceste tipuri de date. De obicei, doar o mică parte din documentele disponibile vor fi relevante pentru utilizator. Fără a ştii ce este în document este dificil să formulezi interogări pentru analiza şi extragerea informațiilor interesante. Utilizatorul are nevoie de componente pentru compararea diferitelor documente, pentru măsurarea importanței şi relevanței documentelor sau pentru extragerea şabloanelor şi a ideilor din mai multe documente.

Recunoaşterea informațiilor (IR) este un domeniu care a fost dezvoltat în paralel cu sistemele de regăsire a informațiilor în bazele de date. O problemă tipică de recunoaştere a informației este gruparea documentelor relevante pe baza intrări furnizate de utilizator, cum ar fi cuvintele cheie sau documentele exemplu. De obicei sistemele de recunoaştere a informației includ sistemele de cataloage din librăriile on-line şi sistemele de management a documentelor on-line [Mann08].

Exista câțiva indicatori pentru măsurarea eficienței algoritmilor de recunoaştere a informației. Notăm cu [Relevant] documentele relevante dintr-o mulțime de documente şi [Regăsit] documentele regăsite din acea mulțime. Mulțimea de documente care cuprinde elementele comune ambelor mulțimi este notată cu $[\mathrm{Re}\,levant] \cap [\mathrm{Re}\,gasit]$. Există doi indicatori de bază pentru aprecierea calității textului regăsit:

- **Precizie regăsite** (*precision*) este procentajul din documentele regăsite care sunt într-adevăr relevante: $precision = \frac{\|[\mathrm{Re}\,levant] \cap [\mathrm{Re}\,gasit]\|}{\|[\mathrm{Re}\,gasit]\|}$
- **Precizie relevante** (*recall*) este procentajul de documente care sunt relevante pentru interogare şi care de fapt sunt şi recunoscute. $recall = \frac{\|[\mathrm{Re}\,levant] \cap [\mathrm{Re}\,gasit]\|}{\|[\mathrm{Re}\,levant]\|}$

Una dintre cele mai utilizate metode de recunoaştere a informației foloseşte cuvinte cheie de bază şi /sau similarități de bază. În metodele de recunoaştere a informațiilor pe baza cuvintelor cheie documentul este reprezentat printr-un şir cuvinte (bag-of-words) considerate relevante pentru acel document. Utilizatorul furnizează un cuvânt sau o expresie formată dintr-un set de cuvinte cum ar fi „car and repair shop”. Un sistem de IR trebuie să găsească acele documente care sunt relevante pentru cuvântul sau cuvintele furnizate de utilizator. Ieşirea sistemului trebuie

să furnizeze şi un grad de relevanţă pentru fiecare document propus ca rezultat, care se bazează şi pe ordinea cuvintelor cheie. În multe cazuri este dificil să furnizezi o măsură precisă a gradului de relevanţă între mulţimi de cuvinte (documente).

Pornind cu un set de *d* documente şi *t* termeni (cuvinte), putem modela fiecare document ca un vector *v* într-un spaţiu *t* dimensional $\mathbb{R}^t$ Componenta *j* a lui **v** reprezintă ponderea termenului de la poziţia *j* pentru documentul dat care este de obicei 0 dacă documentul nu conţine termenul respectiv şi diferit de zero în rest. De exemplu *v[j]* poate reprezenta frecvenţa (numărul de apariţii a) termenului în document.

Sistemele de recunoaştere oferă asocieri între liste de cuvinte de legătură şi mulţimi de documente. Listele de cuvinte de legătură sunt mulţimi de cuvinte care sunt considerate nerelevante. (*a, the, of, for*) pentru acel set de documente. De asemenea un grup de cuvinte diferite împart aceeaşi rădăcină de cuvânt. Pentru reducerea numărului de cuvinte sistemele de recunoaştere a textului trebuie să identifice grupuri de cuvinte care au variaţii semantice mici şi să colecteze doar rădăcini de cuvinte comune pe grup.

Sistemul de recunoaştere a informaţiei se bazează pe ideea că documentele similare au frecvenţă similară a termenilor. Putem măsura similaritatea prin compararea frecvenţelor cuvintelor de bază folosind de exemplu calculul cosinusului unghiului între cei doi vectori de documente.

$$sim(v_1, v_2) = \frac{v_1 * v_2}{|v_1||v_2|}$$

1.1 Seturile de date utilizate în experimente

Experimentele prezentate sunt efectuate folosind colecţia de date Reuters-2000 [Reut00], care conţine 984Mbytes de articole de tip ştiri prezentată într-un format comprimat. Această colecţie este de obicei utilizată în cercetare pentru clasificarea automată a documentelor. Colecţia include un total de 806.791 documente, articole de ştiri publicate de agenţia de presă Reuters în perioada 20 august 1996 – 19 august 1997. Analizate, articolele conţin 9.822.391 paragrafe, 11.522.847 propoziţii şi 310.033 rădăcini de cuvinte distincte rămase după eliminarea cuvintelor de legătură (stopword). Documentele sunt preclasificate de Reuters din punct de vedere a trei categorii distincte. După ramura industrială la care se referă articolul, existând 870 categorii. După regiunea geografică la care se referă articolul existând 366 categorii şi, după anumite categorii propuse de Reuters, în funcţie de conţinut, existând 126 categorii distincte. Dintre acestea din urmă, 23 nu conţin nici un articol. În experimente s-au luat în considerare categoriile în funcţie de conţinut, propuse de Reuters.

În partea de extragere a cuvintelor din documente pentru eliminarea cuvintelor de legătură s-a utilizat o listă generală de cuvinte considerate de legătură pentru limba engleză pusă la dispoziție de Universitatea din Texas. Această listă cuprinde un număr de 509 cuvinte generale, nu neapărat legate de contextul Reuters.

Pentru fiecare cuvânt rămas după procesul de eliminare a cuvintelor de legătură s-a extras rădăcina cuvântului şi s-a contorizat numărul de apariții al acestuia în document. Astfel s-a creat un vector de frecvente de cuvinte pentru fiecare document din Reuters, aceste cuvinte le vom numi în continuare trăsături caracteristice - *features*. Acest vector îl vom considera ca fiind reprezentarea vectorială a documentului în spațiul trăsăturilor caracteristice. Deoarece nu toate cuvintele apar în fiecare document, a mai fost creat un vector care conține toate cuvintele ce apar în toate documentele din setul de date. Acest vector caracterizează întreg setul de documente iar dimensiunea lui reprezintă dimensiunea spațiului de reprezentare a tuturor documentelor sin setul repectiv.

Fiecare vector inițial creat pentru un document a fost modificat astfel încât să devină de lungimea vectorului care conține toate cuvintele, specificându-se valoarea 0 pe pozițiile cuvintelor ce nu apar în documentul respectiv, pe celelalte poziții specificându-se frecvența termenilor.

După acest pas, toți vectorii de reprezentare a documentelor din setul de date au devenit de aceeaşi dimensiune şi putem considera că fiecare vector reprezintă semnătura unui document în spațiul de repezentare a setul de date.

Deoarece memorarea necesară pentru a stoca aceşti vectori este destul de mare, s-a ales varianta de a memora doar valorile din vector pentru care numărul de apariții al cuvintelor este diferit de zero. Astfel, s-au creat perechi de forma *atribut – valoare* care reprezintă trăsătura şi numărul de apariții ale acesteia în documentul curent. Pentru fiecare vector în pate la sfârşit se păstrează categoriile (clasele) propuse de Reuters pentru documentul respectiv.

1.2 Alegerea documentelor pentru antrenare - testare

Datorită dimensiunii mari a bazei de date voi prezenta rezultatele obținute utilizând o submulțime a acesteia. Din toate cele 806.791 documente, s-au selectat acelea care sunt grupate de Reuters în categoria „System Software” după din punct de vedere al codul industrial. După această selecție, s-a obținut un număr de 7.083 documente, care sunt reprezentate utilizând un număr de 19.038 trăsături. În setul rezultat se găsesc 68 clase diferite din punct de vedere al grupării după conținut făcută de Reuters. Dintre aceste clase s-au eliminat acelea care apar în mai

puțin de 1% din toate documentele (slab reprezentate). De asemenea, s-au eliminat clasele care apar în mai mult de 99% dintre documente (excesiv reprezentate).

După aceste eliminări au rămas doar 24 clase distincte şi un număr de 7.053 documente. Pentru a reduce numărului de trăsături de la 19.038 s-a folosit o metodă de selecție a trăsăturilor caracteristice numită „Information Gain" - câştigul informațional, prezentată în secțiunea din capitolul anterior. Astfel s-a calculat pentru fiecare atribut (trăsătură) valoarea care reprezintă câştigul obținut în clasificare dacă păstrăm acel atribut. Valorile din vectorii de reprezentare a documentelor au fost normalizate folosind reprezentarea binară prezentată în secțiunea 1.3. Valoarea maximă optenabilă pentru câştigul informațional este 1. Pentru selectarea doar a atributelor considerate relevante din punct de vedere al câştigului informațional am impus un prag de 0.01. Astfel au fost selectate un număr de 1309 trăsături din cele 19038 existente, selecție realizată pe baza valorii descrescătoare a câştigului informațional.

Cele 7.053 de documente rezultate în urma modificărilor prezentate anterior au fost împărțite aleator într-o mulțime de antrenare de 4702 documente (notată în continuare A1) şi respectiv o mulțime de testare de 2351 documente (notată în continuare T1).

1.3 Tipuri de reprezentare a datelor

Există mai multe posibilități de reprezentare a ponderilor atributelor din vectori. În funcție de reprezentarea aleasă, anumiți algoritmi de clasificare funcționează mai bine sau mai prost. În aplicație s-au folosit trei reprezentări diferite a datelor de intrare astfel:

Reprezentarea Binară – în vectorul de intrare sunt memorate valorile „0" dacă cuvântul respectiv nu apare în document, şi „1" dacă cuvântul respectiv apare în document, fără a se mai memora şi numărul de apariții al acelui cuvânt în document.

Reprezentare Nominală – valorile vectorului de intrare sunt normalizate astfel încât toate valorile să fie cuprinse între 0 şi 1 utilizând următoarea formulă:

$$TF(d,t) = \frac{n(d,t)}{\max_{\tau} n(d,\tau)} \tag{1.1}$$

unde *n(d,t)* reprezintă numărul de apariții al termenului *t* în documentul *d* şi numărătorul reprezintă valoarea termenului care apare de cele mai multe ori în documentul *d*.

Reprezentarea Cornell SMART – în vectorul de intrare valoarea ponderilor este calculată utilizând formula:

$$TF(d,t)=\begin{cases}0 & dacă\ n(d,t)=0\\ 1+\log(1+\log(n(d,t))) & altfel\end{cases} \quad (1.2)$$

unde *n(d,t)* reprezintă numărul de apariţii ale termenului *t* în documentul *d*. În acest caz ponderea poate lua valoarea 0, dacă cuvântul nu apare în document sau o valoare situată în mod practic între 1 şi 2, dacă apare, în funcţie de numărul de apariţii. Valoarea maximă este mărginită superior la 2 pentru un număr de apariţii mai mic decât 10^9, deoarece logaritmul este în baza 10.

Pentru antrenarea şi testarea clasificatorilor implementaţi, am utilizat următoarele seturi de date:

A1 - Acest set de date conţine 4.702 exemple, 1.309 de atribute şi 24 de clase (topic-uri) şi este de forma:

```
#Samples 4702
#Attributes 1309
#Topics 24

@attribute 1.0
@attribute 2.0
@attribute 3.0
@attribute 4.0
@attribute 5.0
@attribute 6.0
@attribute 7.0
...
@attribute 1309.0

@topic c18  747
@topic c181  722
@topic c15  3645
@topic c152  2096
@topic c11  626
@topic c14  179
@topic c22  580
@topic gcat  451
@topic c33  529
@topic c31  456
@topic c13  275
@topic c17  448
@topic c171  385
@topic c12  249
@topic gcrim  258
@topic c21  270
@topic c23  152
@topic c41  395
@topic c411  369
@topic ecat  107
@topic m11  131
@topic mcat  137
@topic c151  1630
@topic c1511  410

@data
0:1 1:8 6:1 8:5 10:1 11:1 13:1 16:2 30:3 35:19 40:1 42:1 57:5 62:2 63:11 64:1
68:3 71:1 77:3 86:1 95:1 111:2 117:1 118:3 120:1 129:1 136:2 147:1 152:1 159:1 168:1
174:1 177:1 181:1 190:1 191:2 194:1 203:2 220:1 226:1 227:1 232:2 234:1 284:1 288:1
293:1 313:1 317:1 332:1 337:4 340:1 342:1 351:1 352:1 353:1 363:1 375:1 442:1 475:1
476:2 481:1 486:1 488:1 492:2 537:2 541:7 555:1 568:1 570:1 641:2 706:1 725:1 743:1
745:1 872:1 877:1 912:1 949:1 979:3 1029:1 1033:1 1051:1 1150:1 1160:1 # c31
0:1 1:1 8:5 16:1 18:2 23:1 39:1 41:1 43:1 46:9 48:1 62:1 71:1 73:1 79:1 82:1
93:1 95:1 114:2 122:1 136:1 150:1 154:1 160:1 175:1 184:1 201:1 213:1 217:1 232:1
```

```
240:1 242:1 251:1 256:2 266:1 284:2 285:1 338:4 351:1 355:4 359:2 372:2 374:5 382:1
386:1 387:2 424:2 450:2 461:1 467:1 469:1 478:1 481:1 511:3 521:3 532:1 552:1 554:1
574:1 579:4 609:4 612:1 619:1 626:1 655:1 663:1 674:1 689:1 701:1 702:2 705:1 718:1
725:1 748:1 776:1 796:1 879:1 896:2 924:1 979:1 1074:1 1131:2 1162:1 1202:1 1227:1
1263:6 # c14 c17 c171
      0:1 2:1 19:1 35:2 43:2 44:1 63:1 75:1 94:1 115:1 126:1 127:1 128:1 130:1 134:1
135:2 136:1 258:1 300:1 464:1 485:1 671:3 1051:1 1052:1 1121:1 # c15
      ...
```

Setul A1 este folosit în deosebi pentru antrenarea clasificatorilor. Pentru testarea acestora am utilizat următoarele seturi de date:

T1 - Setul acesta de date conține 2.351 de exemple cu 1.309 atribute şi 24 de clase. Este folosit pentru evaluarea (testarea) după antrenare a clasificatorilor.

```
#Samples 2351
#Attributes 1309
#Topics 24

@attribute 1.0
@attribute 2.0

...

@attribute 1309.0
@topic c18  747
@topic c181  722
@topic c15  3645
@topic c152  2096
@topic c11  626
@topic c14  179
@topic c22  580
@topic gcat  451
@topic c33  529
@topic c31  456
@topic c13  275
@topic c17  448
@topic c171  385
@topic c12  249
@topic gcrim  258
@topic c21  270
@topic c23  152
@topic c41  395
@topic c411  369
@topic ecat  107
@topic m11  131
@topic mcat  137
@topic c151  1630
@topic c1511  410

@data

      0:1 1:15 2:1 3:12 4:3 5:2 6:2 7:3 8:9 9:2 10:2 11:2 12:3 13:1 14:2 15:3 16:2
17:2 18:1 19:4 20:1 21:2 22:1 23:1 24:2 25:1 26:1 27:2 28:1 29:1 30:1 31:4 32:1 33:1
34:2 35:15 36:1 37:1 38:1 39:1 40:1 41:1 42:1 43:1 44:1 45:2 46:1 47:1 48:2 49:1 50:1
51:2 52:1 53:1 54:1 55:1 56:2 57:1 58:2 59:4 60:2 61:2 62:1 63:4 64:1 65:3 66:2 67:3
68:1 69:1 70:1 71:2 72:1 73:1 74:1 75:1 76:4 77:1 78:1 79:1 80:1 81:1 82:1 83:1 84:1
85:1 86:2 87:1 88:1 89:1 90:1 91:1 92:1 93:1 94:1 95:1 96:1 97:1 98:1 99:1 100:1 101:1
102:2 103:1 104:1 105:1 106:1 107:1 108:1 109:1 110:1 111:1 112:1 113:1 114:1 115:1
116:1 117:1 118:1 119:1 120:1 121:1 122:1 123:1 124:1 125:1 126:1 127:1 # c18 c181
      0:1 2:1 18:1 31:1 115:1 128:1 129:2 130:1 131:1 132:1 133:1 134:1 135:1 136:1
137:1 138:1 139:1 # c15 c152

      ...
```

T2 - Setul acesta de date conține 136 de exemple cu 1.309 atribute şi 24 de clase. Acest set conține acele documentele din setul T1 care nu au putut fi clasificate corect de nici un clasificator selectat în metaclasificatorul prezentat în [Mor07].

```
#Samples 136
#Attributes 1309
#Topics 24

@attribute 1.0
@attribute 2.0

...

@attribute 1309.0
@topic c18  747
@topic c181  722
@topic c15  3645
@topic c152  2096
@topic c11  626
@topic c14  179
@topic c22  580
@topic gcat  451
@topic c33  529
@topic c31  456
@topic c13  275
@topic c17  448
@topic c171  385
@topic c12  249
@topic gcrim  258
@topic c21  270
@topic c23  152
@topic c41  395
@topic c411  369
@topic ecat  107
@topic m11  131
@topic mcat  137
@topic c151  1630
@topic c1511  410

@data

      0:1 1:15 2:1 3:12 4:3 5:2 6:2 7:3 8:9 9:2 10:2 11:2 12:3 13:1 14:2 15:3 16:2
17:2 18:1 19:4 20:1 21:2 22:1 23:1 24:2 25:1 26:1 27:2 28:1 29:1 30:1 31:4 32:1 33:1
34:2 35:15 36:1 37:1 38:1 39:1 40:1 41:1 42:1 43:1 44:1 45:2 46:1 47:1 48:2 49:1 50:1
51:2 52:1 53:1 54:1 55:1 56:2 57:1 58:2 59:4 60:2 61:2 62:1 63:4 64:1 65:3 66:2 67:3
68:1 69:1 70:1 71:2 72:1 73:1 74:1 75:1 76:4 77:1 78:1 79:1 80:1 81:1 82:1 83:1 84:1
85:1 86:2 87:1 88:1 89:1 90:1 91:1 92:1 93:1 94:1 95:1 96:1 97:1 98:1 99:1 100:1 101:1
102:2 103:1 104:1 105:1 106:1 107:1 108:1 109:1 110:1 111:1 112:1 113:1 114:1 115:1
116:1 117:1 118:1 119:1 120:1 121:1 122:1 123:1 124:1 125:1 126:1 127:1 # c18 c181
      0:1 2:1 18:1 31:1 115:1 128:1 129:2 130:1 131:1 132:1 133:1 134:1 135:1 136:1
137:1 138:1 139:1 # c15 c152

...
```

2 Evaluarea clasificatorilor de tip SVM

În [Mor07] este prezentat un metaclasificator bazat pe 8 clasificatoare de tip SVM care era folosit pentru îmbunătățirea acurateței de clasificare a documentelor de tip text. Maximul acurateței de clasificare obținut de către un singur clasificatorul de tip SVM este 87.11% şi a fost obținut de clasificatorul SVM de tip polinomial de grad 2 cu reprezentare Cornell Smart. În [Mor07] sunt prezentați şi testați mai mulți clasificatori de tip SVM bazați atât pe nucleul polinomial cât şi pe cel Gaussian cu diferite forme de reprezentare. Dintre toți clasificatorii testați şi prezentați, s-au inclus în metaclasificator 8 clasificatori SVM distincți. Alegerea celor 8 clasificatori s-a făcut pe baza acurateței de clasificare obținută de aceştia. Pentru metaclasificatorul din [Mor07] s-au ales clasificatorii de tip SVM cu cea mai bună acuratețe de clasificare astfel:

Nr. crt.	Tipul nucleului	Grad	Reprezentarea datelor
1	Polinomial	1	Nominal
2	Polinomial	2	Binar
3	Polinomial	2	Cornell Smart
4	Polinomial	3	Cornell Smart
5	Gaussian	1.8	Cornell Smart
6	Gaussian	2.1	Cornell Smart
7	Gaussian	2.8	Cornell Smart
8	Gaussian	3.0	Cornell Smart

Tabel 2.1 - Clasificatorii de tip SVM aleşi în metaclasificatorul din [Mor07]

Utilizând aceşti clasificatori s-a ajuns la o acuratețe maximă de clasificare de 92,04% în cazul metaclasificatorului bazat pe distanța euclidiană şi după un număr de 14 paşi de învățare. Tot în [Mor07] s-a prezentat şi o analiză în care se calcula şi limita maximă la care ar putea să ajungă metaclasificatorul astfel creat (cu cei 8 clasificatori selectați). Limita calculată era de 94.21%. Această limită a clasificării s-a obținut, deoarece din 2351 de documente de test, 136 de documente nu au putut fi clasificate corect de nici un clasificator selectat în cadrul metaclasificatorului.

În prima fază am încercat găsirea unui nou clasificator care să reuşească să clasifice corect documentele, ce s-au dovedit imposibil de clasificat de către toți clasificatorii selectați în metaclasificator din [Mor07].

2.1 Problema limitării metaclasificatorului cu clasificatori de tip SVM

O parte din documentele de testare nu pot fi clasificate corect de nici un clasificator selectat în cadrul metaclasificatorului, fapt ce duce la o acuratețe a clasificării - maximum optenabilă - de 94,21%.

Pentru a verifica clasificatorii în condițiile prezentate în [Mor07], am antrenat clasificatorii SVM pe setul de date A1 (4.702 exemple şi 1.309 atribute) şi am utilizat ca date de testare setul de date T2 (136 de exemple - cele cu probleme - şi 1.309 atribute). Având în vedere faptul că documentele din setul T2 nu au putut fi clasificate corect în urma efectuării testelor după cum ne aşteptam, rezultatul clasificării corecte este aproape de 0%. Totuşi există clasificatori SVM, care nu au fost selectați în cadrul metaclasificatorului, dar care reuşesc să clasifice corect o parte din cele 136 documente. Aceşti clasificatori nu au fost selectați deoarece au avut o acuratețe de clasificare pe setul T1 mai slabă dar se pare că pe setul T2 dau rezultate mai bune.

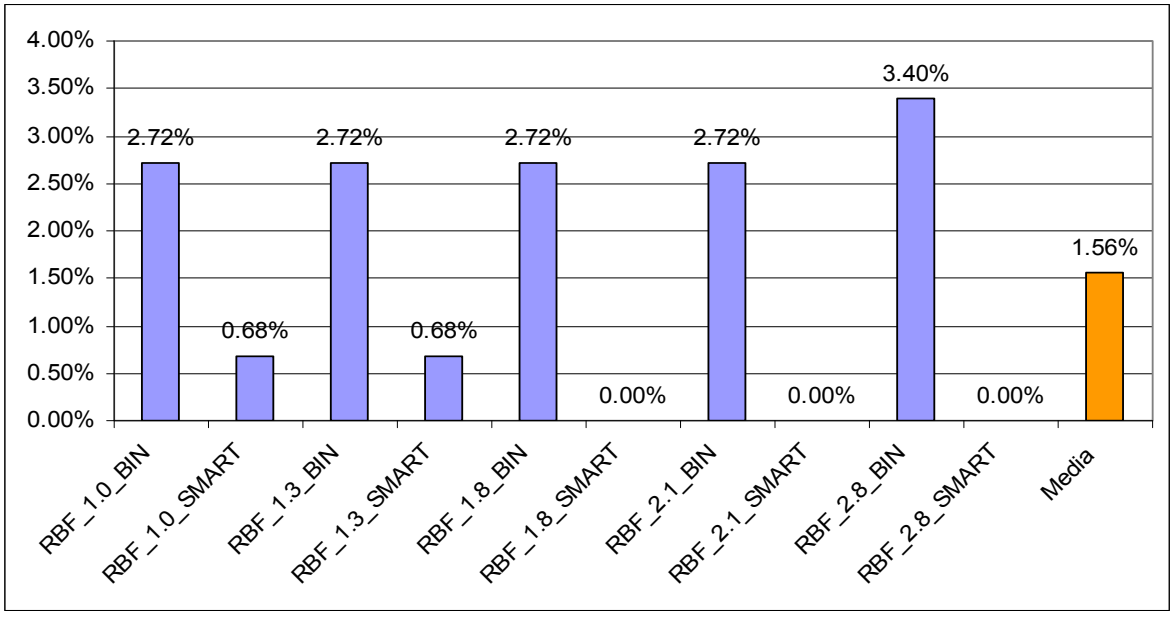

Fig. 2.1 Rezultate obținute de clasificatorii SVM pe setul de date de antrenament A1 şi testat pe T2 - nucleu gaussian

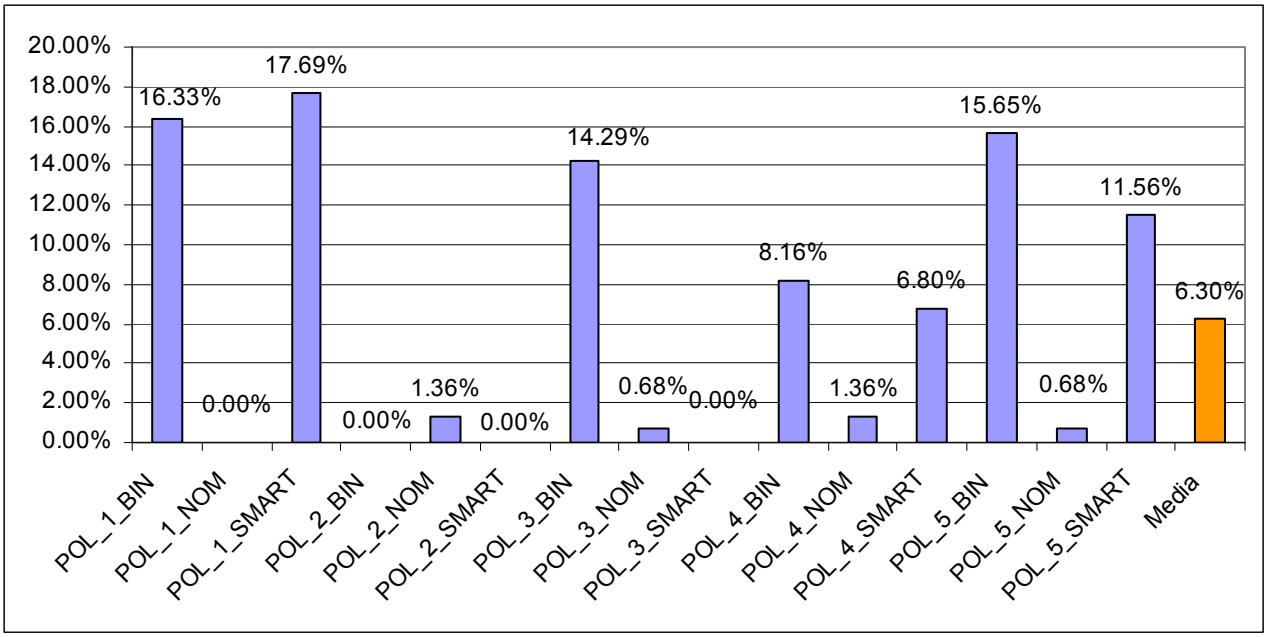

Fig. 2.2 Rezultate obținute de clasificatorii SVM pe setul de date de antrenament A1 şi testat pe T2 - nucleu polinomial

Doar clasificatorul cu nucleu polinomial de grad 1 şi reprezentare Cornell Smart a reuşit clasificarea corectă a 24 de documente din cele 136. Avem două explicații:

a. documentele care nu pot fi clasificate apar în prea puține clase sau sunt cazuri particulare ale unei clase
b. au fost greşit clasificate în baza de date Reuters

2.2 O primă tatonare a problemei

Într-o prima etapă, pentru a testa dacă documentele ar putea fi clasificate corect am ales ca set de antrenament pentru clasificatorii de tip SVM (selectați în metaclasificatorul prezentat) setul de date T1, care au fost utilizate în [Mor07] ca şi set de test. În cadrul acestui set de date se regăsesc şi cele 136 de exemple care nu au putut fi clasificate corect de către metaclasificator. Cu alte cuvinte, clasificatorii de tip SVM au fost antrenați pe acel set de date care conține şi documentele considerate cu probleme. (Precizez aici că antrenând SVM-urile doar pe documentele cu probleme – cele 136 documente (setul T2) -, majoritatea, după antrenare, au reuşit să clasifice corect toate documentele cu probleme.) În graficele următoare sunt prezentate rezultatele clasificării obținute pe un setul de testare T2 (doar 136 documente.) dar antrenate pe setul T1

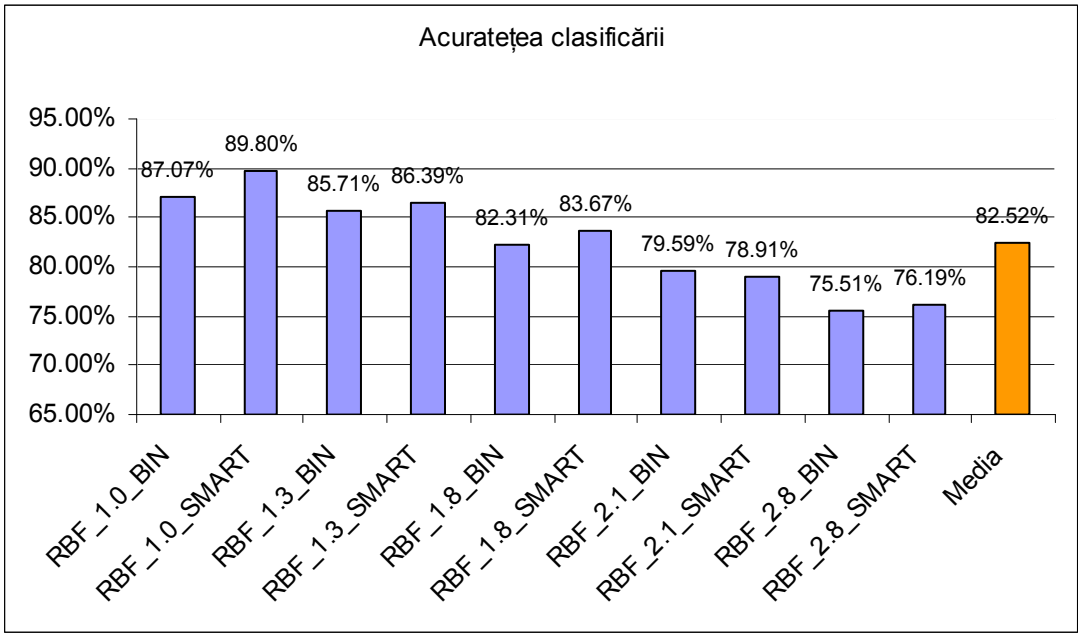

Fig. 2.3 Rezultate obținute de clasificatorii SVM utilizând diferite tipuri de reprezentare a datelor (binar, nominal şi Cornell-Smart) cu nucleu de tip gaussian

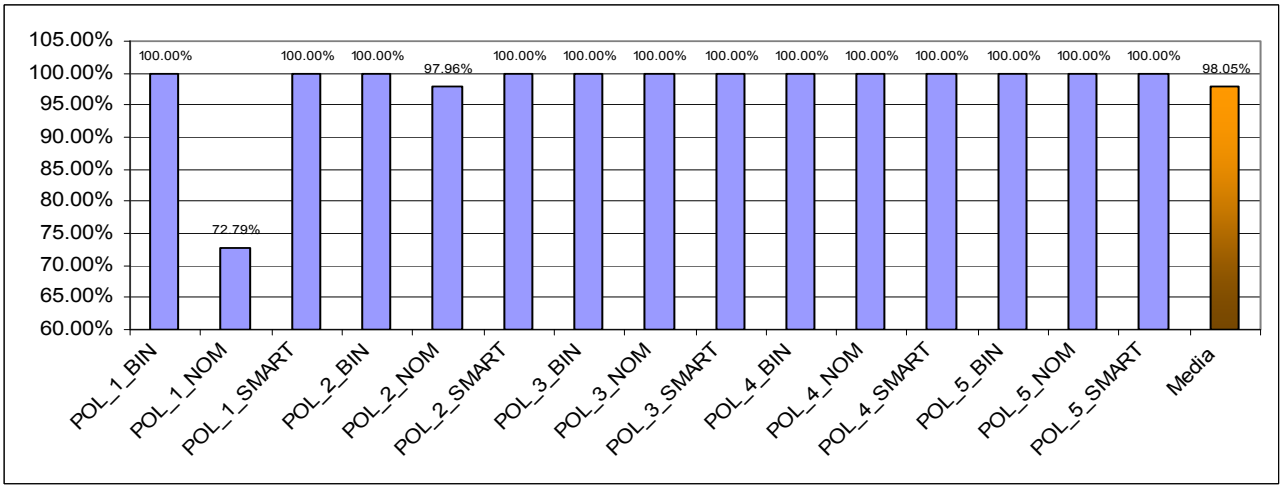

Fig. 2.4 Rezultate obținute aplicând clasificatori svm utilizând diferite tipuri de reprezentare a datelor (binar, nominal şi Cornell-Smart) cu nucleu de tip polinomial

Observăm că rezultatele obținute cu clasificatori SVM, care utilizează nucleul polinomial, au o acuratețe a clasificării mai mare (media este de 98,05%) decât cei care utilizează nucleul Gaussian (media este de 82,52%). Remarcăm faptul că din 15 clasificatori cu nucleu polinomial, doar doi clasificatori - polinomial de grad 1 şi 2 cu reprezentarea datelor de tip nominal - nu au reuşit o acuratețe a clasificării de 100% a documentelor.

Rezultatele de mai sus pot fi explicate prin faptul, că utilizarea la clasificatorii de tip SVM a unui nucleu polinomial (liniar) duce la rezultate mai bune, deoarece s-a constatat că documentele-text în reprezentarea vectorilor de termeni sunt liniar separabile. Transformarea acestor vectori într-un nou spațiu folosind funcții neliniare (nucleu Gaussian) îngreunează găsirea unui hiperplan optim de separare fapt confirmat şi în [Gab04].

2.3 Soluții pentru îmbunătățirea metaclasificatorului folosind clasificatoare de tip SVM

2.3.1 Soluția 1 – introducerea unor noi clasificatori SVM

O soluție ar fi introducerea dinamică de noi clasificatori în metaclasificator, în cazul în care în faza de antrenare/testare ar apărea un anumit număr (de ex. 50) de documente care nu pot fi clasificate corect de nici unul dintre clasificatorii deja existenți. Astfel se va introduce un nou clasificator SVM de tip polinomial, antrenat special pe acele (50) documente.

2.3.2 Soluția 2

O altă soluție ar consta în alegerea unei alte categorii pentru un document dificil clasificabil, pentru care, de asemenea, clasificatorul întoarce un răspuns pozitiv mare. Dacă nici un clasificator nu va fi selectat pentru clasificarea unui document conform regulilor prezentate in [Mor07] atunci se va alege clasificatorul cu cea mai mare probabilitate de reuşită (distanța între documentul curent şi toate documentele din coada clasificatorului este maximă, chiar dacă este mai mică decât pragul stabilit). De la clasificatorul astfel selectat nu se va mai alege prima clasă propusă ci următoarea dacă ea este suficient de aproape față de prima clasă. Această soluție se va prezenta mai bine pe un exemplu.

Exemplu:

Presupunem că în metaclasificator avem 4 clasificatoare diferite, fiecare având în coada de erori un număr de documente. Când avem un nou document d_n care trebuie clasificat, se ia pe rând fiecare clasificator şi se calculează distanta între d_n şi fiecare document din coada de erori a clasificatorului respectiv. Dacă cel puțin o distanță calculată este mai mică decât pragul stabilit, metaclasificatorul nu va folosi acel clasificator pentru a clasifica documentul d_n. În cazul în care se rejectează astfel toți clasificatorii, metaclasificatorul totuşi va alege pe cel care are distanța cea mai mare obținută (chiar dacă este mai mică decât pragul stabilit). Actualmente, metaclasificatorul va prezice clasa specificată de acest clasificator, chiar dacă se ştie cu o probabilitate mare că acesta va clasifica prost documentul d_n. Modificarea ar fi ca, în acest caz, clasificatorul ales să nu mai selecteze clasa pentru care se obține valoarea cea mai mare (pentru că oricum va da greş deoarece clasifică prost tipul respectiv de documente), ci să aleagă clasa imediat următoare din lista de clase pe care le prezice. Se va alege următoarea clasă prezisă doar dacă valoarea pentru aceasta este suficient de apropiată de valoarea maximă obținută de clasificator (cu un ε). În acest caz, clasificatorul ar specifica o altă clasă pentru documentul curent d_n.

Rezultatele obținute utilizând această ipoteză le vom prezenta în secțiunea 5.4.

3 Clasificatorul Naïve Bayes

O altă soluție pentru îmbunătățirea metaclasificatorului ar fi găsirea unui alt clasificator, nu neapărat de tip SVM, care să reuşească să clasifice documentele cu problemă (cele 136) fără să fie antrenat pe acele documente. Clasificatorul de tip Naive Bayes s-a dovedit a fi de succes în cazul utilizării sale în clasificare documentelor de tip text [Lewis98], [McCall98],[Domin97]. Pentru aceasta am făcut câteva experimente cu un clasificator de tip Bayes Naive. În acest caz, am folosit clasificatorul Bayes din pachetul IR pus la dispoziție de Universitatea din Texas [WEB09]. Într-o primă etapă, l-am folosit aşa cum este implementat în [WEB09], apoi i-am adus anumite modificări pentru a putea fi integrat ulterior în metaclasificator [Mor07]. Modificările aduse se referă la modul de funcționare în cazul clasificării în mai multe clase, la comportamentul acestuia când există documente clasificate inițial în mai multe clase şi la modul de selecție al datelor de antrenare şi testare.

Clasificatorul Bayes Naive nemodificat (BNN) testat primeşte în cazul de față ca date de intrare toate fişierele care urmează a fi clasificate urmând ca alegerea datelor de antrenament şi a datelor de test să se facă după metoda "*n-fold crossvalidation*". Ideea acestei metode este de a împărți un set de date în *n* subseturi, urmând ca *n-1* de subseturi să se folosească la antrenare iar testarea să se facă pe subsetul nefolosit la antrenament, adică antrenarea şi testarea să se execute pe seturi de date disjuncte. Algoritmul se va executa de n ori astfel încât fiecare subset de date va fi o singură dată un subset de test pentru verificarea antrenării. Din păcate în acest caz nu mai pot fi selectate exact seturile prezentate la început fapt ce a dus la modificarea modului de lucru al clasificatorului BNA (Bayes Naive adaptat)

3.1 Clasificarea Bayes

Fie Y o variabilă pentru o clasă (categorie) care poate lua valorile $\{y_1, y_2, ..., y_m\}$.

Fie X o instanță a unui vector cu n atribute $<X_1, X_2, ..., X_n>$ şi x_k o valoare posibilă pentru X şi x_{ki} o valoare posibilă pentru x_k. Pentru clasificarea de tip Bayes calculăm probabilitățile $P(Y = y_i | X = x_k)$, *pentru* $i = \overline{1,m}$. Asta ar însemna calcularea tuturor probabilităților pentru fiecare categorie pentru fiecare instanță posibilă din spațiul de instanțe – ceea ce este foarte greu de calculat pentru un set rezonabil de date.

Practic pentru a determina categoria lui x_k, trebuie să determinăm pentru fiecare y_i probabilitatea [Duda73]:

$$P(Y = y_i \mid X = x_k) = \frac{P(Y = y_i)P(X = x_k \mid Y = y_i)}{P(X = x_k)} \quad (3.1)$$

Probabilitatea $P(X = x_k)$ poate fi determinată deoarece categoriile sunt complete şi disjuncte. Rezultă imediat relația de echilibru de mai jos:

$$\sum_{i=1}^{m} P(Y = y_i \mid X = x_k) = \sum_{i=1}^{m} \frac{P(Y = y_i)P(X = x_k \mid Y = y_i)}{P(X = x_k)} = 1 \quad (3.2)$$

Aşadar:

$$P(X = x_k) = \sum_{i=1}^{m} P(Y = y_i)P(X = x_k \mid Y = y_i) \quad (3.3)$$

Probabilitatea $P(Y = y_i)$ poate fi uşor aproximată având în vedere faptul că dacă n_i exemple din D se regăsesc în y_i atunci $P(Y = y_i) = \frac{n_i}{|D|}$, unde D reprezintă mulțimea documentelor din setul de antrenament.

Probabilitatea $P(X = x_k \mid Y = y_i)$ trebuie estimată (deoarece există 2^n posibile instanțe pentru a calcula probabilitatea). De aceea, dacă presupunem că atributele unei instanțe sunt independente (condițional independente), atunci:

$$P(X \mid Y) = P(X_1, X_2, \cdots X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y) \quad (3.4)$$

Astfel trebuie să calculăm doar $P(X_i \mid Y)$ pentru fiecare posibilă pereche "valoare atribut" - "categorie"

Dacă Y şi toate X_i sunt binare, atunci trebuie să calculăm doar 2n valori

$P(X_i = true \mid Y = true)$ şi $P(X_i = true \mid Y = false)$ pentru fiecare X_i

$P(X_i = false) \mid Y) = 1 - P(X_i = true \mid Y)$

față de 2^n valori, dacă nu am presupune independența atributelor.

Practic, dacă setul de date D conține n_k exemple din categoria y_k şi n_{ij_k} din aceste n_k exemple au a j-a valoare pentru atributul X_i pe x_{ij} atunci estimăm că:

$$P(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk}}{n_k} \quad (3.5)$$

Această estimare poate genera erori la seturi foarte mici de date deoarece un atribut rar într-un set de antrenament face ca X_i să fie fals în setul de antrenament $\forall y_k \; P(X_i = true \mid Y = y_k) = 0$.

Dacă X_i=*true* într-un exemplu de test atunci $\forall y_k \;\; P(X \mid Y = y_k) = 0$ şi $\forall y_k \;\; P(Y = y_k \mid X) = 0$

Pentru a evita acest lucru se utilizează uniformizarea (normalizarea) lui Laplace. Această normalizare pleacă de la premisa că fiecare atribut are o probabilitate *p* observată într-un exemplu virtual de dimensiune *m*.

Astfel

$$P(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk} + mp}{n_k + m} \quad (3.6)$$

unde *p* este o constantă. De exemplu pentru atribute binare p=0,5

Pentru clasificarea de text, clasificatorul Bayes generează pentru un document dintr-o anumită categorie un "bagaj de cuvinte" dintr-un vocabular V = {w_1, w_2,…w_m} calculând probabilitatea $P(w_j|c_i)$. Pentru normalizarea Laplace se presupune existența unei distribuții uniforme a tuturor cuvintelor (adică ar fi echivalentul unui exemplu virtual în care fiecare cuvânt apare doar o singură dată).

$p = \frac{1}{|V|}$ şi m = |V|

3.1.1 Antrenarea clasificatorului Bayes

Probabilitatea ca un document *Y* să aparțină clasei *Xi* se calculează după cum urmează:

$$P(X_i \mid Y) \simeq P(X_i) \prod_{j=1}^{n} P(y_j \mid X_i) \quad (3.7)$$

unde $P(y_j \mid X_i)$ este probabilitatea condițională ca termenul y_j să apară într-un document al clasei X_i. Interpretăm $P(y_j \mid X_i)$ ca fiind o măsură a contribuției lui y_j în stabilirea faptului că X_i este clasa corectă.

$P(X_i)$ este probabilitatea apariției unui document în clasa X_i.

$\langle y_1, y_2, ..., y_j \rangle$ sunt termeni din documentul *Y* şi sunt submulțime a vocabularului utilizat pentru clasificare, iar *n* reprezintă numărul termenilor.

În clasificarea documentelor-text, scopul nostru este de a găsi cea mai bună clasă pentru respectivul document. În clasificarea Naive Bayes cea mai bună clasă se stabileşte după metoda maximului a posteriori (MAP) şi o notăm cu c_{map} :

$$c_{map} = \arg\max_{1<i<m} \overline{P}(X_i \mid Y) = \arg\max_{1<i<m} \overline{P}(X_i) \prod_{j=1}^{n} \overline{P}(y_j \mid X_i) \qquad (3.8)$$

Am utilizat notarea $\overline{P}$ pentru *P* deoarece nu cunoaştem exact valorile parametrilor $\overline{P}(X_i)$ şi $\overline{P}(y_j \mid X_i)$ dar care pot fi estimați pe baza setului de antrenament. Există mai multe modele a clasificatorului Naive Bayes incluzând modelul bazat pe reprezentarea binară, modelul multinominal, modelul Poisson [Eyher03]. S-a demonstrat că utilizarea reprezentării multinominale este de obicei cea mai bună alegere în clasificarea documentelor text [Eyher03], [McCall98].

La noi estimarea parametrilor $\overline{P}(X_i)$ şi $\overline{P}(y_j \mid X_i)$ se face după cum este descris în continuare. Pentru antrenarea clasificatorului fie *V* vocabularul de cuvinte din documentele conținute în *D* şi pentru $\forall$ o categorie $X_i \in X$ fie D_i un subset de documente din *D* din categoria X_i, atunci:

$$\overline{P}(X_i) = \frac{|D_i|}{|D|} \qquad (3.9)$$

Fie Y_i concatenarea tuturor documentelor din D_i şi n_i numărul aparițiilor tuturor cuvintelor din Y_i atunci pentru fiecare cuvânt $y_j \in V$ fie n_{ij} numărul aparițiilor cuvântului y_j în Y_i
atunci

$$\overline{P}(y_j \mid X_i) = \frac{(n_{ij} + 1)}{(n_i + |V|)} \qquad (3.10)$$

Parametrii luații în considerare pentru cazul nostru sunt:
Numărul total de atribute *n = 1309*, numărul total de documente din setul de antrenare *D= 4702*, numărul total de clase, în cazul nostru *m = 16* şi de asemenea numărul total de documente existente în fiecare clasă $D_1, \ldots D_{16}$.
De exemplu din setul de date de antrenament luăm clasa C18 (X_1=C18) şi ea conține 328 de documente. Algoritmul utilizează uniformizarea Laplace şi probabilitatea clasei C18 se calculează astfel:

$$\overline{P}(X_1) = \frac{Nr.\ documente + 1}{Nr.\ totaldocumente + Nr.\ clase} = \frac{328 + 1}{4702 + 16} = 0,069732938 \qquad (3.11)$$

Pentru fiecare din cele 1309 atribute calculăm probabilitățile în raport cu fiecare clasă în parte.

Atribute	X_1 (C18)		X_2 (C15)		...		X_{16} (m11)	
	Nr. apariții	$\overline{P}(y_1 \mid X_1)$	Nr. apariții	$\overline{P}(y_1 \mid X_2)$			Nr. apariții	$\overline{P}(y_1 \mid X_{16})$
y_1	50		12				1	
y_2	12							
...								
y_{1309}	0							

Tabel 3.1 Calcularea probabilităților pentru fiecare trăsătură (1309)

Probabilitatea condițională ca un atribut să fie într-o anumită clasă se calculează astfel:

$$\overline{P}(y_1 \mid X_1) = \frac{Nr.apariții\ y_1 + 1}{Nr.total\ \ cuv.din X_1 + Nr.cuv.din Y} = \frac{50+1}{2570+1309} = 0,013147718$$

ş.a.m.d

3.1.2 Testarea clasificatorului

Fie D_1 un document de test care conține n_{D_1} termeni. În cazul nostru, pentru un document care trebuie clasificat, extragem toate atributele şi extragem din tabelul prezentat mai sus probabilitățile condiționale.

În ecuația (3.8) din 3.1.1 trebuiesc calculate multe probabilități condiționale şi, datorită faptului că unele pot fi foarte mici, prin înmulțire se poate ajunge la *floating point underflow*. Pentru a evita acest lucru, ne folosim de binecunoscuta proprietate a logaritmului conform căreia

$$\log(xy) = \log x + \log y \qquad (3.12)$$

Deoarece funcția logaritmică este monotonă, logaritmarea ecuației (3.8) nu va modifica rezultatul alegerii clasei

Astfel:

$$c_{map} = \arg\max_{1<i<m} \left[\overline{P}(X_i) + \sum_{j=1}^{n} \log \overline{P}(y_j \mid X_i) \right] \qquad (3.13)$$

Ecuația (3.13) are o interpretare simplă: fiecare parametru condițional $\log \overline{P}(y_j \mid X_i)$ este o pondere care arată cât de bun este un "indicator" y_j pentru X_i. Similar probabilitatea $\log \overline{P}(X_i)$ este o pondere care indică frecvența relativă a clasei *c*. Suma acestora este o măsură a evidenței ca un document să aparțină unei clase. Ecuația (3.13) alege cea mai semnificativă clasă.

Astfel vom obține pentru fiecare clasă o valoare (care va fi negativă, deoarece logaritmăm o valoare subunitară) şi alegem clasa cu valoarea cea mai mare. De exemplu pentru fişierul nostru de test obținem următoarele valori pentru fiecare clasă în parte:

```
Document: TestFile1578
Results:      c18(-366.59583445019484)
              c15(-277.3757195772894)
              c11(-393.7555314343488)
              c14(-376.69712554934097)
              c22(-405.2708070760941)
              gcat(-390.2472558128614)
              c33(-393.06805295995537)
              c31(-379.5501501924242)
              c13(-397.21992866728175)
              c17(-371.92813590774523)
              c12(-398.6571293708641)
              c21(-387.3768662210122)
              c23(-403.69793168505237)
              c41(-409.92000232701463)
              ecat(-390.18163176978925)
              m11(-395.70584000569795)
Correct class: 1 (c15), Predicted class: 1 (c15)
```

3.2 Rezultate obținute cu clasificatorului Bayes

Am testat clasificatorul Bayes (BNN) pe un sistem cu procesor AMD X2 Turion la 1,7GHz şi 3 GB RAM. Pentru validarea datelor de test am utilizat metoda "*n-Fold Crossvalidation*". Am ales n=10 ceea ce înseamnă că setul de date existent se va împărți în 10 submulțimi disjuncte, fiind utilizate 9 submulțimi pentru antrenament şi şi a 10-a submulțime neutilizată la antrenare să fie folosită la testare. Această operație se execută de 10 ori, astfel încât toate submulțimile alese vor fi utilizate o singură dată în testarea clasificării. De asemenea, pentru a urmări şi acuratețea învățării, am ales procente diferite din datele de intrare care vor fi folosite pentru antrenarea clasificatorului astfel: 0%, 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% şi 100%.

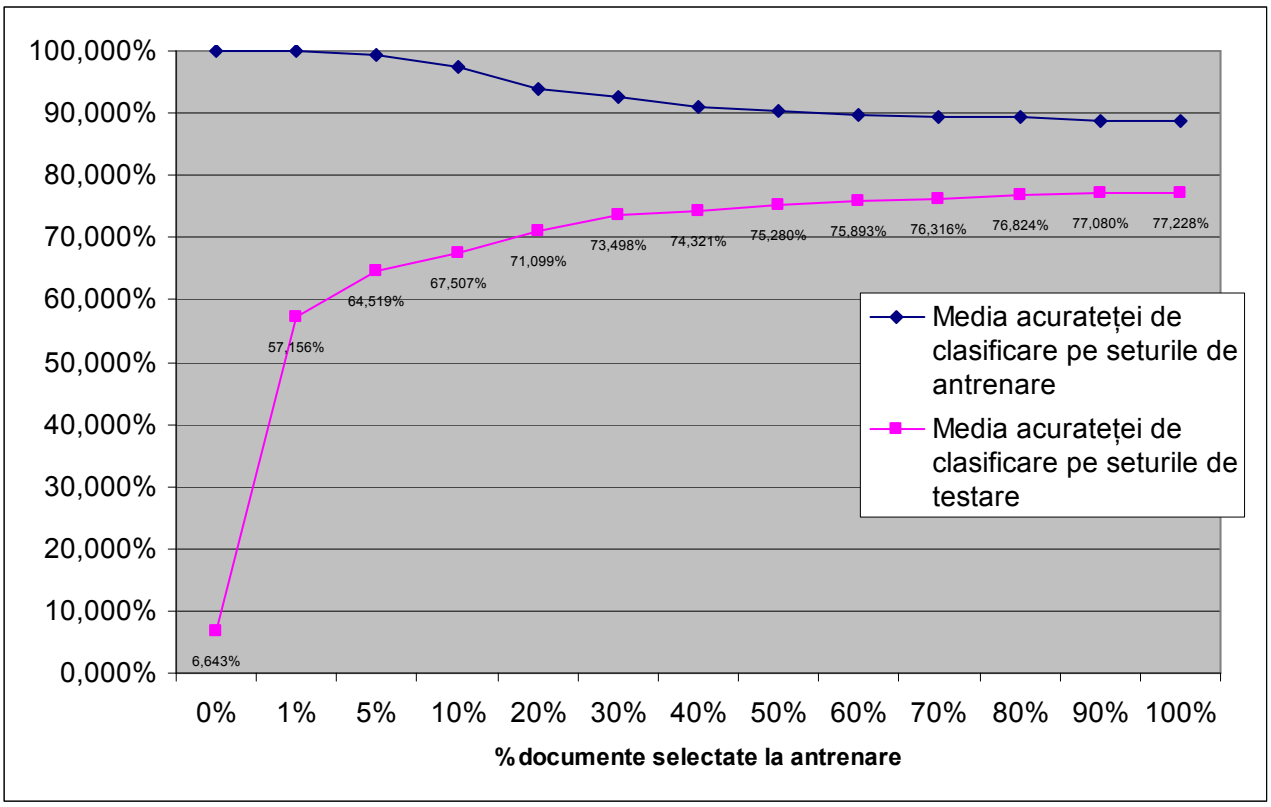

Fig. 3.1 Acuratețea clasificării şi curba de învățare a clasificatorului Bayes

În graficul prezentat în Fig. 3.1 pe axa x sunt reprezentate numărul de documente utilizate succesiv pentru antrenare. Valorile corespund procentajelor: 0%, 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% şi 100%. Pe axa y sunt reprezentate valorile acurateței de testare şi respectiv de antrenare.

În acest experiment s-au utilizat setul de antrenament A1 şi setul de testare T1(7053 de documente din baza de date Reuters) împreună urmând ca pentru împărțire în date de antrenare şi testare să se utilizeze metoda „*n-Fold Crossvalidation*". Ne propunem să utilizăm acest clasificator în clasificarea documentelor din setul de testare T2, adică testăm dacă poate să clasifice corect documente care nu au putut fi clasificate corect de clasificatorii SVM.

Astfel, se va utiliza setul de antrenare A1 şi pentru testare se va utiliza setul T2 care este format din 136 documente pe care Bayes le împarte în 10 subseturi. În medie, rezultatele clasificării sunt mai bune decât SVM (care a obținut maxim 18% pe când Bayes a obținut un maxim de 33.56%). Totuşi, deocamdată, nu putem specifica în cazul clasificatorului Bazes numărul de atribute alese din cele 1306 prezentate la intrare, şi nici metoda de selecție folosită.

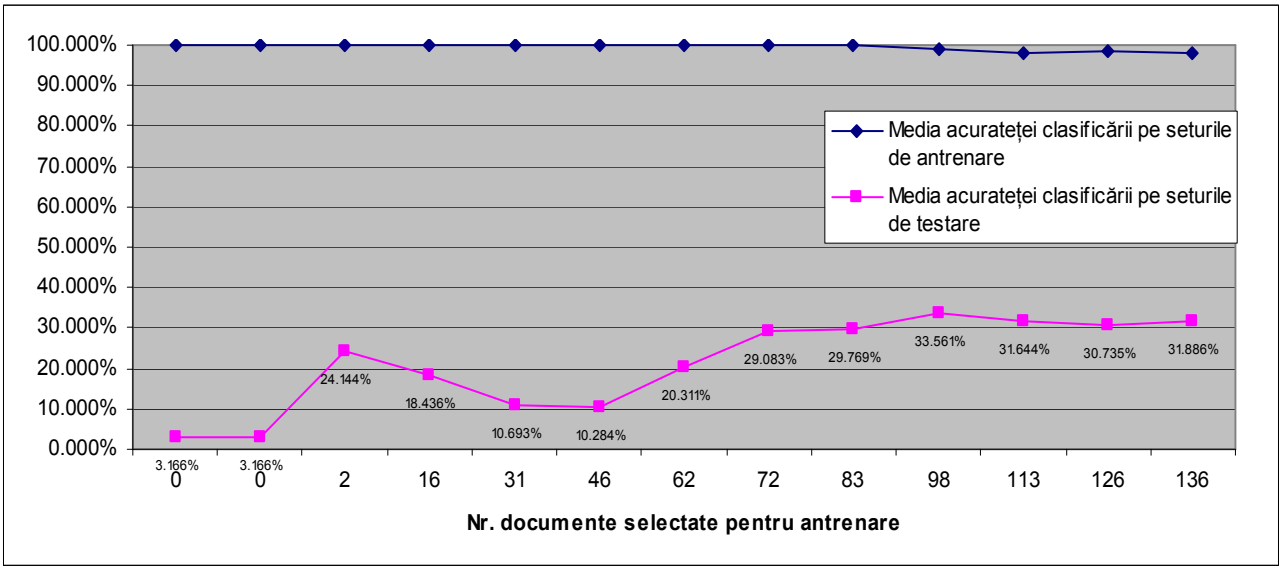

Fig. 3.2 Utilizarea clasificatorului Bayes în clasificarea unor documente care nu au fost clasificate corect de clasificatorul SVM (T2)

În Fig. 3.2 sunt prezentate rezultatele de clasificare ca şi medie pentru cele 10 treceri prin clasificator. Observăm că în Fig. 2.2 cel mai bun rezultat obținut de clasificatorul de tip SVM a fost o clasificare corectă de 17,69%. În cazul clasificatorului de tip Bayes (vezi Fig. 3.2) am obținut un maxim de 33,56%.

3.3 Adaptarea clasificatorului Bayes pentru utilizarea în metaclasificator

Pentru a putea integra şi clasificatorul Bayes în metaclasificatorul prezentat în [Mor07] au trebuit făcute câteva modificări. Clasificatorului Bayes primeşte la intrare un set de date din care el alege aleator subseturi de antrenament şi de test, după metoda propusă în [WEB09]. În prima fază, am modificat modul de alegere a setului de antrenament şi a celui de testare astfel încât acestea să fie identice cu seturile de antrenare şi testare folosite pentru celelalte clasificatoare din metaclasificator (SVM polinomial, SVM gaussian).

În cazul clasificatorului Bayes acesta se va antrena întotdeauna pe acelaşi set de antrenament (A1) şi se va testa pe alt set de date (T1). Astfel, datele de antrenare şi de testare devin identice pentru toate clasificatoarele din metaclasificator.

Pentru verificarea funcționării corecte a clasificatorului Bayes modificat (BNM), în contextul bazei de date Reuters, am ales în prima fază din setul A1 toți vectorii de reprezentare a documentelor, dar pentru fiecare vector s-au ales doar primele 100 de atribute (în ordine descrescătoare a câştigului informațional obținut de fiecare atribut în parte), reducând astfel dimensiunea acestora. În urma antrenării şi testării pe aceste seturi de date, acuratețea de

clasificare obținută a fost de 33,18%. Această clasificare slabă s-a obținut deoarece în baza de date Reuters majoritatea vectorilor sunt clasificați în mai mult de o categorie.

În prima fază am încercat o clasificare la mai multe clase de genul „*one class versus the rest*". În cazul în care un document aparținea mai multor clase (ceea ce este obişnuit la baza de date Reuters), acel document a fost trecut în setul de antrenare de mai multe ori, în funcție de numărul de clase specificate de Reuters pentru acel document. Aceasta face ca multe clase să fie suprapuse (există multe clase care sunt subcategorii ale unei clase de bază – atunci toate documentele dintr-o subclasă pot să fie şi într-o altă clasă de bază).

Am luat în considerare prima dată această opțiune, deoarece în clasificatorii de tip SVM această metodă („one class versus the rest") este folosită pentru antrenarea la mai mult de două clase.

Clasificatorul BNN din [WEB09] cu un procentaj de 60% din documente alese pentru antrenare obține o acuratețe de 75,893% (Fig. 3.1). Această valoare reprezintă o diferență majoră față de 33,176% obținut acum de BNA. Diferența apare deoarece în prima fază clasificatorul BNA a fost antrenat pe un set de date din Reuters în care am considerat că fiecare document aparține doar unei singure categorii (clase), indiferent de câte categorii erau specificate de Reuters pentru acel document, la fel ca în [WEB09].

Din acest motiv, am modificat antrenarea BNA pe cele două seturi fixe (A1 şi T1), eliminând clasele în cazul în care un document aparținea mai multor clase. De exemplu, dacă fişierul *file134.xml* era clasificat în Reuters ca aparținând clasei *C15* şi clasei *C151,* am eliminat clasa *C151*. Astfel un document se consideră că aparține doar primei categorii specificate de Reuters, celelalte categorii fiind ignorate. Rezultă astfel doar 16 categorii distincte care vor fi folosite pentru clasificatorul Bayes (cel modificat).

O altă modificare adusă clasificatorului BNA, pentru a funcționa pe aceleaşi set de date şi a furniza rezultatul în acelaşi mod ca şi clasificatorii SVM a fost schimbarea modului în care se validează rezultatul clasificării. Deoarece în baza de date Reuters documentele erau clasificate în două sau mai multe categorii, unele din acestea fiind însă subcategorii ale unei categorii de bază, am validat rezultatul ca fiind corect, dacă clasificatorul Bayes specifică corect una din clasele precizate de Reuters. De exemplu, dacă clasificatorul stabileşte pentru un document clasa *C151,* iar în Reuters el apare în *C15* şi apoi în *C151,* am considerat rezultatul clasificării ca fiind corect.

Clasificarea cu BNA s-a îmbunătățit considerabil ajungând la o acuratețe de 72,14% aproape identică cu cea atinsă de clasificatorul BNN din [WEB09]. Diferența apare deoarece în primul caz clasificatorul primeşte un set de date pe care îl împarte el automat în subseturi de antrenare şi testare prezentând la sfârşit o medie a rezultatelor obținute pe aceste subseturi, iar în

al doilea caz clasificatorul primeşte un set fix de antrenare şi un set fix de testare, iar rezultatul de 72,14% este cel obținut pe aceste seturi fixe.

Deşi prezentarea medie acurateței de clasificare este o măsură mai bună a performanțelor clasificatorului, deoarece acesta este testat şi antrenat pe mai multe submulțimi diferite disjuncte, în cazul metaclasificatorului acest lucru nu se pretează, deoarece seturile de date ar trebui schimbate la nivel de metaclasificator nu la nivel de clasificator. Acesta este motivul pentru care clasificatorii se antrenează şi se testează pe seturi prestabilite.

În acest moment, putem afirma că cele 3 categorii de clasificatoare - SVM cu nucleu polinomial, SVM cu nucleu gaussian şi Bayes (adaptat - BNA) rulează în acelaşi context. Singura diferență între clasificatorii de tip SVM şi cel de tip Bayes este că pentru SVM se vor folosi 24 de clase iar pentru Bayes doar 16 clase.

4 Compararea clasificatorului Bayes adaptat (BNA) cu clasificatorii de tip SVM

În această secțiune prezentăm rezultate comparative între clasificatorii de tip SVM şi clasificatorul Bayes adaptat. Ideea este de a putea vedea dacă sunt şanse de îmbunătățire a acurateței de clasificare a metaclasificatorului prezentat în [Mor07], în cazul introducerii în metaclasificator şi a unui clasificator de tip Bayes.

4.1 Antrenarea clasificatorilor pe setul A1 şi testarea pe setul T1

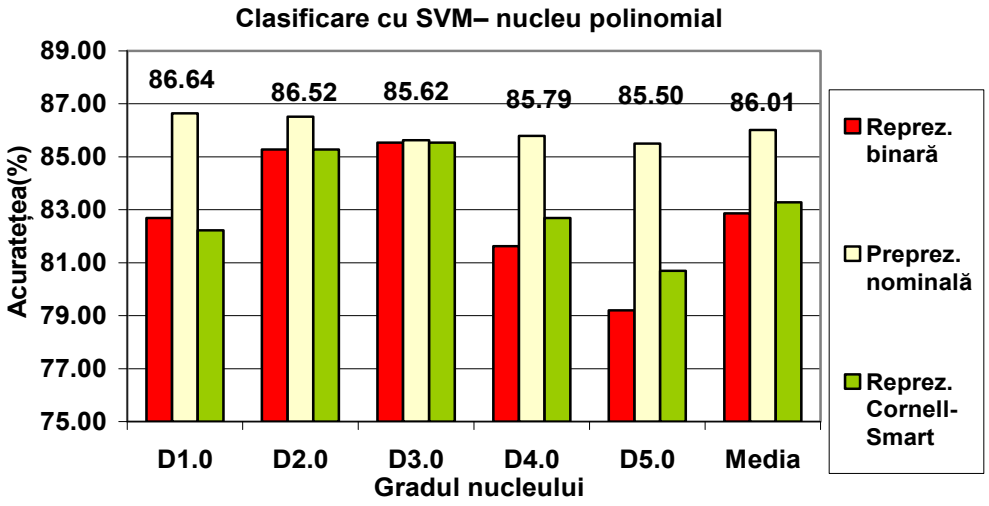

Fig. 4.1Rezultatele clasificării cu SVM nucleu polinomial (preluat din[Mor07])

Rezultatele prezentate s-au obținut de clasificatorii SVM cu nucleu polinomial, pentru diferite grade ale nucleului şi pentru diferite reprezentări ale vectorilor de documente, antrenați pe setul A1 şi testați pe setul T1. Graficul din fig. 4.1 indică faptul că rezultatele cele mai bune (86,64%) s-au obținut utilizând clasificatorul SVM cu nucleu polinomial de grad 1 şi reprezentare nominală. Ca şi medie a acurateței de clasificare, acest tip de clasificator a obținut cel mai bun rezultat pentru reprezentarea nominală (86,01%).

Singurul parametru care ne permite reglarea acurateței de clasificare la clasificatorul Bayes este procentul de documente de antrenament (după cum se vede în Fig 3.1). În cazul acesta, setul de antrenare şi de testare este prestabilit. Din acest motiv, avem doar un singur rezultat pentru Bayes, rezultat pe care îl comparăm atât cu cel mai bun rezultat obținut de SVM polinomial, cât şi cu media obținută de acesta. După cum se poate observa în figura următoare cu clasificatorul Bayes antrenat pe setul A1 şi testat pe setul T1, fiecare având 1.309 atribute, s-a

obținut o acurateţe a clasificării de 81,32%. Trebuie specificat că clasificatorul Bayes foloseşte doar metoda nominală de reprezentare a vectorului de documente.

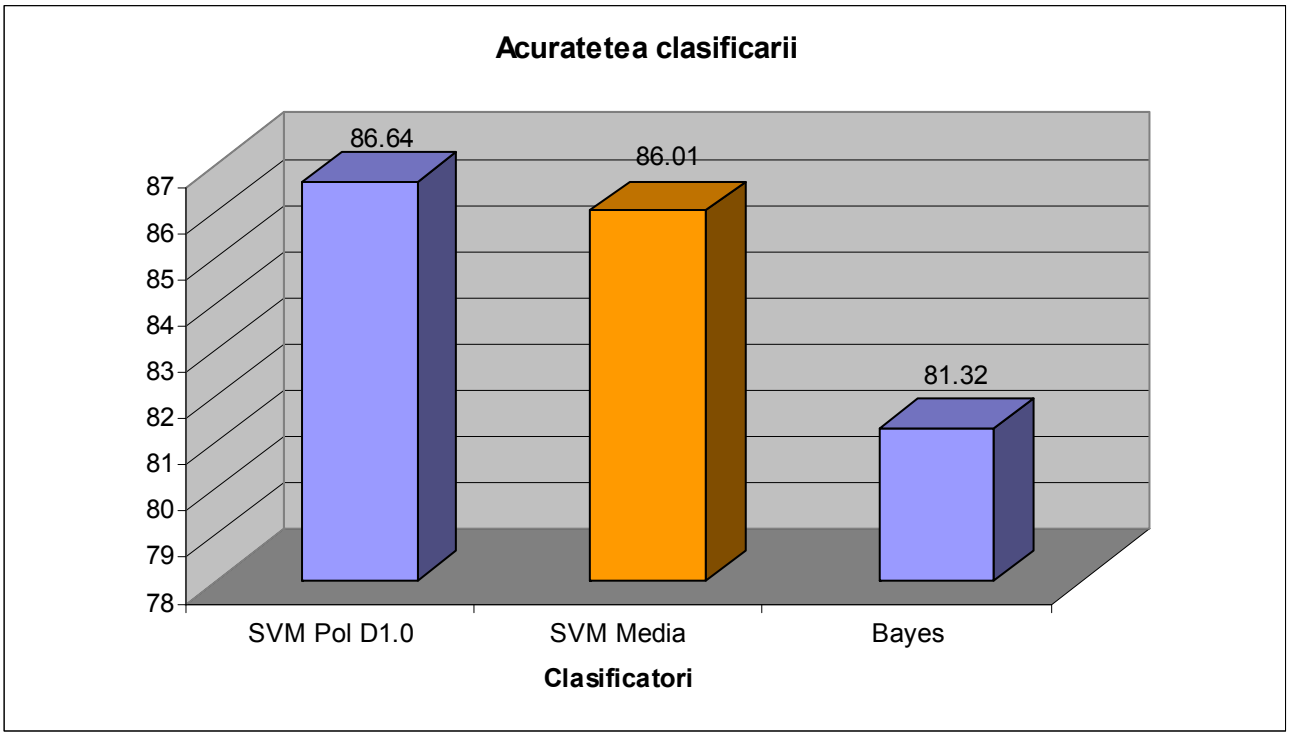

Fig. 4.2 Compararea rezultatelor obținute cu clasificatori SVM şi Bayes

În graficul următor se prezintă timpii de antrenare obținuți de fiecare clasificator în parte. Timpii de antrenare sunt obținuți pe un calculator P IV la 3.2Ghz cu 1Gb memorie.

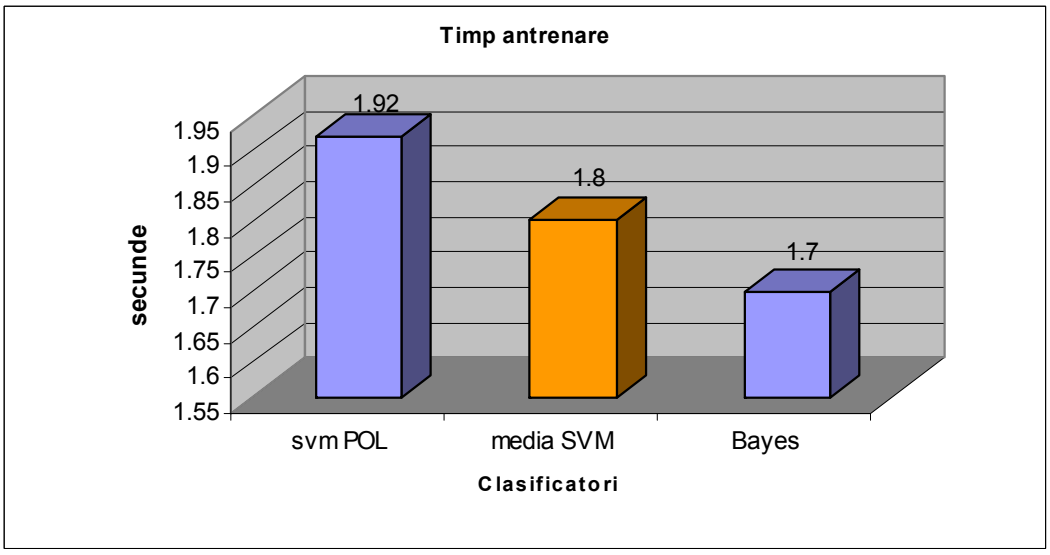

Fig. 4.3 Compararea timpilor de antrenare obținuți cu clasificatori SVM şi Bayes

Ca şi timpi de antrenare, clasificatorul Bayes oferă timpi mai mici de antrenare deoarece calculează doar rapoarte între atribute şi clase.

4.2 Antrenarea pe setul A1 şi testarea pe setul T2

Amintim că în setul de test T2 avem doar acele documente care nu au putut fi clasificate corect de nici un clasificator SVM din cadrul metaclasificatorului prezentat in [Mor07]. Antrenarea pentru fiecare clasificator s-a făcut pe setul A1.

Amintim faptul că clasificatorii de tip SVM nu au dat rezultate satisfăcătoare pe acest set de documente. După cum s-a observat din figurile 2.2 (SVM polinomial) şi 2.1 (SVM Gaussian) clasificatorii de tip SVM obţin rezultate diferite de 0 pe acest set (T2), dar cu alţi parametri de intrare decât cei folosiţi în metaclasificator. În graficul următor prezentăm cele mai bune rezultate obţinute de SVM polinomial şi SVM Gaussian pe acest set precum şi mediile obţinute pentru toate testele (cele din figurile 2.1 şi 2.2). Pe ultima bară se prezintă rezultatul obţinut de clasificatorul Bayes pe acest set de test. În [Mor07], „regula" de selecţie a clasificatorilor care care au fost introduşi în metaclasificator a fost să obţină cele mai bune rezultate pe setul T1 (cu 2.351 vectori de documente). După cum s-a observat şi din figurile 2.1 şi 2.2, există clasificatori SVM pentru care documentele din setul T2 (cel cu 136 vectori de documente) se clasifică ceva mai corect (oricum, nemulţumitor), dar aceştia au obţinut rezultate nesatisfăcătoare pe setul mare (T1)(de exemplu: *SVM polinomial, grad 1, reprezentarea Cornell SMART* -80,99%, *SVM polinomial, grad 1, reprezentarea Binară* -81,45%, *SVM polinomial, grad 3, reprezentarea BIN* -85,79% - din Tabel 6.1 [Mor07]) şi de aceea nu au fost selectaţi în metaclasificator. Introducerea în metaclasificator a unui alt clasificator SVM care clasifică ceva mai bine setul T2 (de exemplu cel polinomial de grad 1 cu reprezentare Cornell Smart) alături de cele selectate în [Mor07], care obţin 0% pe acest set, ar modifica limita maximă la care poate ajunge metaclasificatorul (în sus sau în jos).

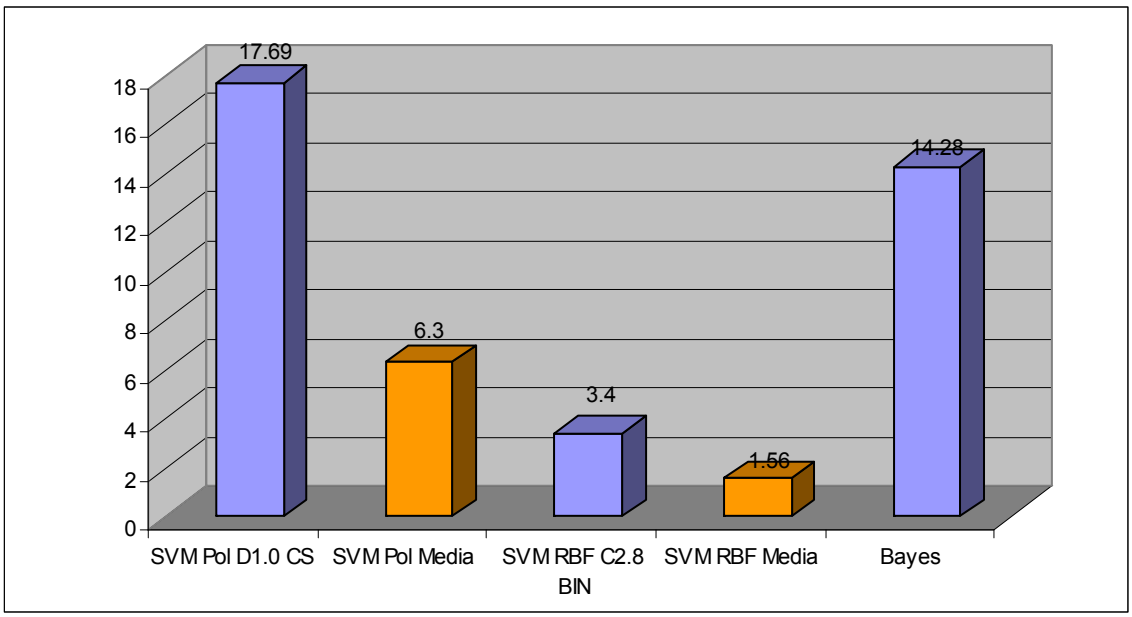

Fig. 4.4 Rezultate obținute de clasificatorii SVM şi Bayes în clasificarea setului T2 şi antrenat pe A1

În urma testelor efectuate, am observat că, deşi acuratețea totală a clasificatorului Bayes (BNA) generează rezultate mai slabe decât SVM el totuşi clasifică corect 104 documente din cele 136 documente cu probleme din setul T2, chiar dacă este antrenat pe setul A1.

4.3 Antrenarea şi testarea pe setul T2

În figurile 2.3 şi 2.4 am prezentat rezultatele obținute de clasificatorii de tip SVM de tip gaussian şi SVM de tip polinomial în cazul în care s-au antrenat pe setul T2 (cu doar 136 documente) şi s-au testat pe acelaşi set. Clasificatorii SVM de tip polinomial au reuşit o acuratețe a clasificării şi de 100%, în medie ei obținând o valoare de 98.05%. Clasificatorul Bayes a obținut o acuratețe a clasificării de 97.95%, puțin mai mică față de SVM. Ceea ce ne interesează pe noi este dacă acest clasificator va reuşi în metaclasificator să clasifice corect documentele din setul T2.

5 Metaclasificatori

În urma rezultatelor favorabile din punct de vedere al setului T2 (sec. 4.2) am decis includerea clasificatorul de tip Bayes în metaclasificatorul prezentat în [Mor07]. Astfel pe lângă cei 8 clasificatori selectați, am mai introdus unul (Bayes), metaclasificatorul având acum nouă clasificatori. Am refăcut testele pentru toate cele 3 modele de metaclasificatori prezentate: vot majoritar (MV), selectare pe baza distanței euclidiene (SBED) şi selectare pe bază de cosinus (SBCOS).

Acum metaclasificatorul conține 8 clasificatori SVM şi un clasificator Bayes, astfel:

Nr. Crt.	Tip clasificator	Paremetrul nucleului	Reprezentarea datelor de intrare	Acuratețea obținută(%)
1	SVM-Polinomial	1	Nominal	86,69
2	SVM-Polinomial	2	Binary	86,64
3	SVM-Polinomial	2	Cornell Smart	87,11
4	SVM-Polinomial	3	Cornell Smart	86,51
5	SVM-Gaussian	1.8	Cornell Smart	84,30
6	SVM-Gaussian	2.1	Cornell Smart	83,83
7	SVM-Gaussian	2.8	Cornell Smart	83,66
8	SVM-Gaussian	3.0	Cornell Smart	83,41
9	Bayes	-	Nominal	81,32

Tabel 5.1 Clasificatorii incluşi în metaclasificator

De asemenea, am calculat limita maximă la care ar putea ajunge noul metaclasificator. Astfel în urma introducerii clasificatorului Bayes limita maximă a metaclasificatorului creşte la **98,63%** (față de 94,21% cât era fără Bayes), ceea ce oferă o posibilitatea obținerii unei acurateţe a clasificării mult mai bune.

5.1 Selecția bazată pe vot majoritar

Ideea acestei metode este de a utiliza toți clasificatorii din metaclasificator pentru a clasifica documentul curent. Fiecare clasificator votează pentru o anumită categorie pentru documentul curent. Metaclasificatorul va păstra pentru fiecare categorie votată un contor, incrementând contorul categoriei când un clasificator votează acea categorie. Metaclasificatorul va selecta categoria cu cel mai mare contor. Dacă se obțin două sau mai multe categorii cu aceeaşi valoare a contorului se va considera documentul curent clasificat în toate categoriile propuse de către metaclasificator. Marele dezavantaj al acestui metaclasificator este că nu îşi modifică evoluția o dată cu datele de intrare în scopul îmbunătățirii acurateței clasificării, fiind deci un model neadaptiv.

Folosind această metodă acuratețea clasificării obținute este de 86,09%. În cazul introducerii unui nou clasificator în metaclasificator, acuratețea clasificării a scăzut față de valoarea obținută cu 8 clasificatori (86,38%), deci având o scădere de 0,29%. Aceasta poate apărea deoarece pe tot setul de test clasificatorul Bayes obține o acurateţe de doar 81,32%, clasificând destul de multe documente (439) incorect, ceea ce se pare că „ajută”, în metaclasificator, la selectarea în 7 cazuri a unor categorii greşite deoarece Bayes a întărit votul greşit.

5.2 Selecția pe baza distanței euclidiene (SBED)

Deoarece metoda prezentată anterior nu obține rezultate satisfăcătoare, în [Mor07] a fost dezvoltat un metaclasificator care îşi modifică comportamentul în funcție de datele de intrare. Pentru a face aceasta, clasificatorul va fi selectat în funcție de eşantionul curent de intrare. Astfel, putem afirma că metaclasificatorul învață datele de intrare după o metodă euristică. Metaclasificatorul va învăța doar eşantioanele incorect clasificate, deoarece, ne aşteptăm ca numărul de eşantioane corect clasificate să fie mai mare decât numărul de eşantioane incorect clasificate. În parte, fiecare clasificator va învăța eşantioanele care sunt incorect clasificate de către el. Metaclasificatorul va conține pentru fiecare clasificator o coadă proprie în care se vor memora documentele incorect clasificate de acel clasificator. Astfel, metaclasificatorul va conține 9 cozi ataşate celor 9 clasificatori componenți. În continuare vom exemplifica funcționarea acestui metaclasificator printr-un exemplu.

Fie un document de intrare (eşantion curent) care trebuie să fie clasificat. Se alege aleator un clasificator din cei 9 disponibili. Calculăm distanța euclidiană (ecuația 5.1) dintre eşantionul curent şi toate eşantioanele care se găsesc în coada clasificatorului selectat. Dacă obținem cel puțin o distanță mai mică decât un prag prestabilit atunci vom renunța la clasificatorul selectat şi vom selecta un alt clasificator dintre cei rămaşi. Dacă nu, îl vom folosi pentru clasificarea

eşantionului curent. În cazul în care toți clasificatorii sunt eliminați, îl vom alege pe acela pentru care am obținut distanța euclidiană cea mai mare chiar dacă ea este mai mică decât pragul stabilit..

$$Eucl(x,x') = \sqrt{\sum_{i=1}^{n}([x]_i - [x']_i)^2} \qquad (5.1)$$

unde $[x]_i$ reprezintă valoarea pentru intrarea i a vectorului x, iar x şi x' reprezintă vectorii de intrare, unul fiind eşantionul curent iar celălalt fiind vectorul din coada clasificatorului.

După selectarea clasificatorului, acesta va fi utilizat pentru a clasifica eşantionul curent. Dacă clasificatorul selectat reuşeşte să clasifice corect eşantionul curent, nu se acționează asupra metaclasificatorului. În caz contrar, eşantionul curent este pus în coada de documente a clasificatorului selectat. Se face aceasta deoarece se doreşte ulterior evitarea utilizării acestui clasificator pentru a clasifica documente asemănătoare cu acest document.

Acest proces are doi paşi. Toate acțiunile prezentate până acum se realizează în primul pas numit şi pasul de învățare. În acest pas, metaclasificatorul analizează setul de antrenament şi, de fiecare dată când un document este clasificat greşit, este pus în coada clasificatorului selectat la acel moment. În cel de-al doilea pas, numit pasul de testare, se verifică acuratețea procesului de clasificare. În acest pas, caracteristicile metaclasificatorului rămân neschimbate. Deoarece după fiecare pas de antrenare caracteristicile metaclasificatorului vor fi schimbate, am repetat aceşti doi paşi de mai multe ori.

Prezentăm rezultatele obținute de noul metaclasificator cu 9 clasificatori comparativ cu metaclasificatorul din [Mor07]. Se prezintă doar primii 14 paşi deoarece după acest număr de paşi am observat că acuratețea clasificării nu se mai modifică substanțial. La fel ca în [Mor07] pragul pentru primi 7 paşi s-a ales egal cu 2,5 şi pragul pentru ultimii 7 paşi s-a ales egal cu 1,5. De asemenea, prezentăm în acest grafic şi rezultatele obținute cu metoda vot majoritar atât pentru metaclasificatorul cu 8 clasificatoare cât şi pentru cel nou.

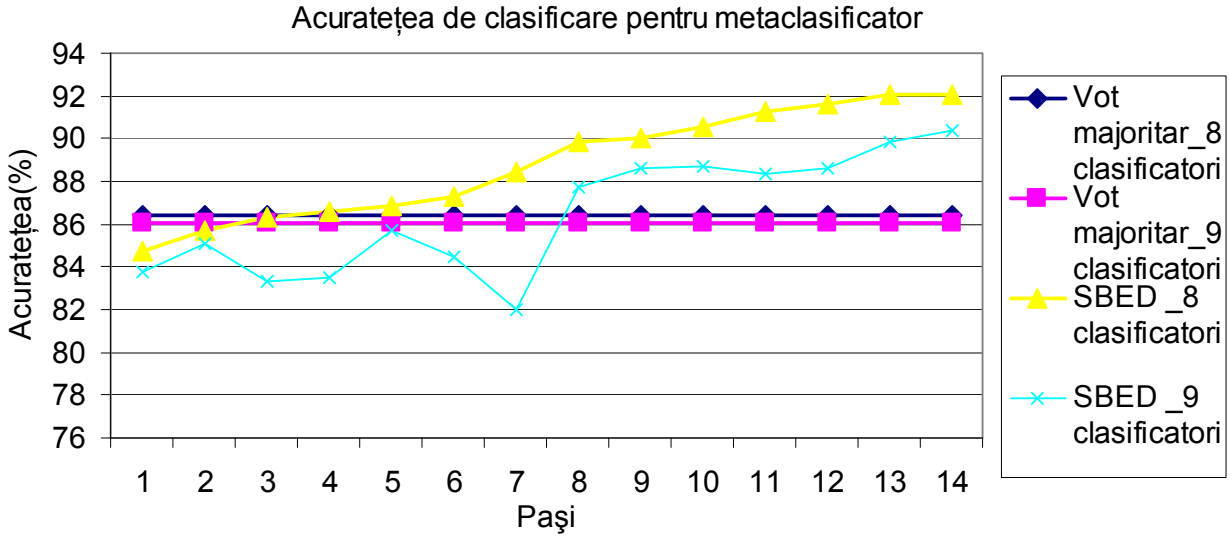

Fig. 5.1 Acuratețea clasificării - vot majoritar şi distanță euclidiană (metaclasificator cu 8 şi 9 clasificatori)

În cazul metaclasificatorului cu 9 clasificatoare rezultatele obținute sunt mai slabe decât cele realizate de metaclasificatorul cu 8 clasificatoare. Se pare că acuratețea proastă a clasificatorul Bayes (81.32%) comparativ cu cel SVM reduce, în acest caz, acuratețea globală de clasificare a metaclasificatorului.

Observăm că acuratețea clasificării în cazul metaclasificatorul cu 9 clasificatoare are şi tendințe descrescătoare. Aceasta se poate datora faptului că un clasificator poate clasifica corect un document d_1 şi clasifica incorect un alt document d_2 apropiat ca distanță de documentul d_1. Din acest motiv, la o nouă parcurgere a setului de test, clasificatorul respectiv nu a mai fost selectat pentru clasificarea lui d_1 (deoarece a dat rezultate proaste pentru d_2) şi atunci căutându-se un alt clasificator (care poate, la rândul său să clasifice greşit).

5.3 Selectarea bazată pe cosinus (SBCOS)

Cosinusul este o altă posibilitate de calculare a similarității între documente. Această metrică este des utilizată în literatură când se lucrează cu vectori care caracterizează documentele şi se bazează pe calcularea produsului scalar dintre doi vectori. Formula utilizată pentru calculul cosinusului unghiului θ dintre doi vectori de intrare x şi x' este:

$$\cos\theta = \frac{\langle x, x' \rangle}{\|x\| \cdot \|x'\|} = \frac{\sum_{i=1}^{n} [x]_i [x']_i}{\sqrt{\sum_{i=1}^{n} [x]_i^2} \cdot \sqrt{\sum_{i=1}^{n} [x']_i^2}} \qquad (5.2)$$

unde x şi x' sunt vectori de intrare (documentele) şi $[x]_i$ reprezintă componenta i a vectorului x.

Această metodă, se aseamănă cu metoda SBED singura modificare apărând în calculul similarității între documente. În această metodă consider că clasificatorul curent selectat este acceptat dacă toate cosinusurile calculate între eşantionul curent şi toate eşantioanele care se găsesc în coadă sunt mai mici decât un prag prestabilit. Clasificatorul va fi respins dacă cel puțin un cosinus este mai mare decât pragul stabilit.

Prezentăm rezultatele obținute de noul metaclasificator cu 9 clasificatori comparativ cu metaclasificatorul cu 8 clasificatori. Se prezintă doar primii 14 paşi, deoarece după acest număr de paşi acuratețea clasificării nu se mai modifică substanțial. La fel ca în [Mor07], pragul pentru primi 7 paşi s-au ales egali cu 0,8 şi pragul pentru ultimii 7 paşi s-a ales egal cu 0,9. De asemenea în acest grafic prezentăm şi limita maximă optenabilă a noului metaclasificator.

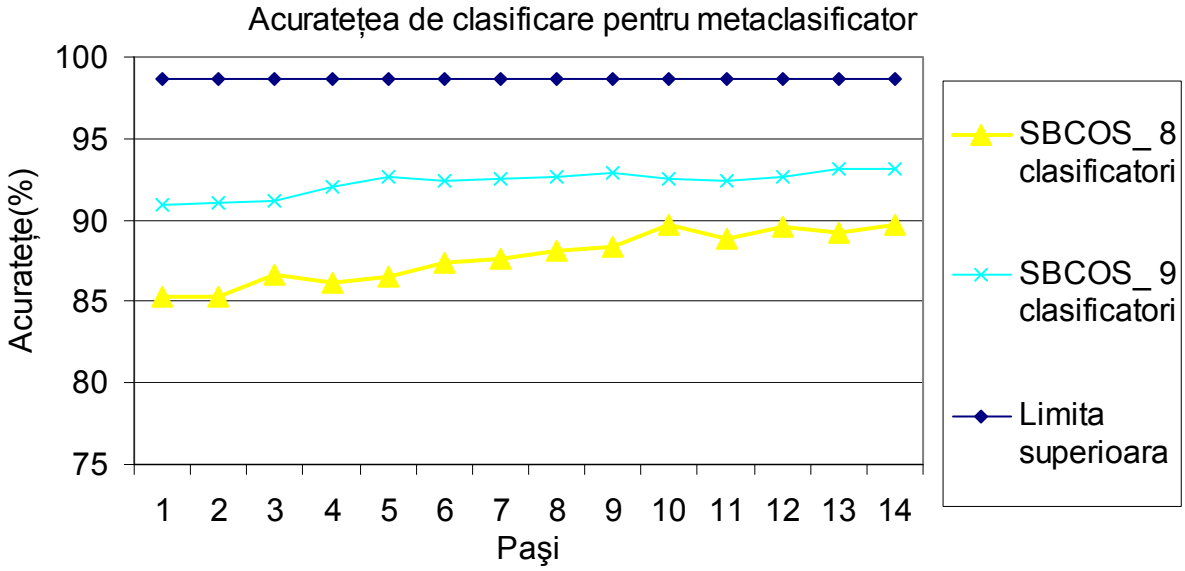

Fig. 5.2 Rezultatele obținute cu metaclasificatorul cu 8 şi cu 9 clasificatori utilizând cosinusul

În cazul acestui metaclasificator, rezultatele obținute arată că acuratețea de clasificare s-a îmbunătățit de la 89,74% la **93,10%** prin adăugarea clasificatorului Bayes (BNA).

5.4 Rezultate obținute modificând alegerea clasei

În secțiunea 2.3 am propus două posibile soluții pentru îmbunătățirea acurateței de clasificare a metaclasificatorului. În secțiunea 2.3.2 am afirmat că, în cazul unui nou document d_n care trebuie clasificat, se ia pe rând fiecare clasificator şi se calculează distanta între d_n şi fiecare document din coada de erori a clasificatorului respectiv. Dacă cel puțin o distanță calculată este mai mică decât pragul stabilit, metaclasificatorul nu va folosi acel clasificator pentru a clasifica documentul d_n. În cazul în care se rejectează astfel toți clasificatorii, metaclasificatorul totuşi va alege pe cel care are distanța cea mai mare obținută (chiar dacă este mai mică decât pragul stabilit). Actualmente, metaclasificatorul va prezice clasa specificată de

acest clasificator. Modificarea adusă în acest caz metaclasificatorului ar fi că, în acest caz, clasificatorul ales să nu mai selecteze clasa cu valoarea cea mai mare (pentru că oricum va da greş deoarece este predispus să clasifice eronat tipul respectiv de documente), ci să aleagă clasa imediat următoare din lista de clase pe care le prezice. Se va alege următoarea clasă prezisă doar dacă valoarea pentru aceasta este suficient de apropiată de valoarea maxim obținută de clasificator (cu un ε=0,5 ales în experimentele efectuate). În acest caz, clasificatorul ar specifica o altă clasă pentru documentul curent d_n. Efectuând aceste modificări, rezultatele metaclasificatorului cu 9 clasificatori s-au îmbunătățit. În continuare vom numi acest metaclasificator: metaclasificator cu 9 clasificatori modificat.

Prezentăm comparativ rezultatele obținute de metaclasificatorul cu 9 clasificatori modificat în cazul selecției bazate pe distanța euclidiană şi selecției bazate pe cosinus. Pentru selecția bazată pe votul majoritar, nefiind vorba de învățare, modificările aduse metaclasificatorul nu au nici o influență asupra rezultatului final. De asemenea am prezentat în fiecare grafic şi limita maximă optenabilă a metaclasificatorului cu 9 clasificatoare.

Rezultatul obținut în cazul selecție bazate pe distanța euclidiană:

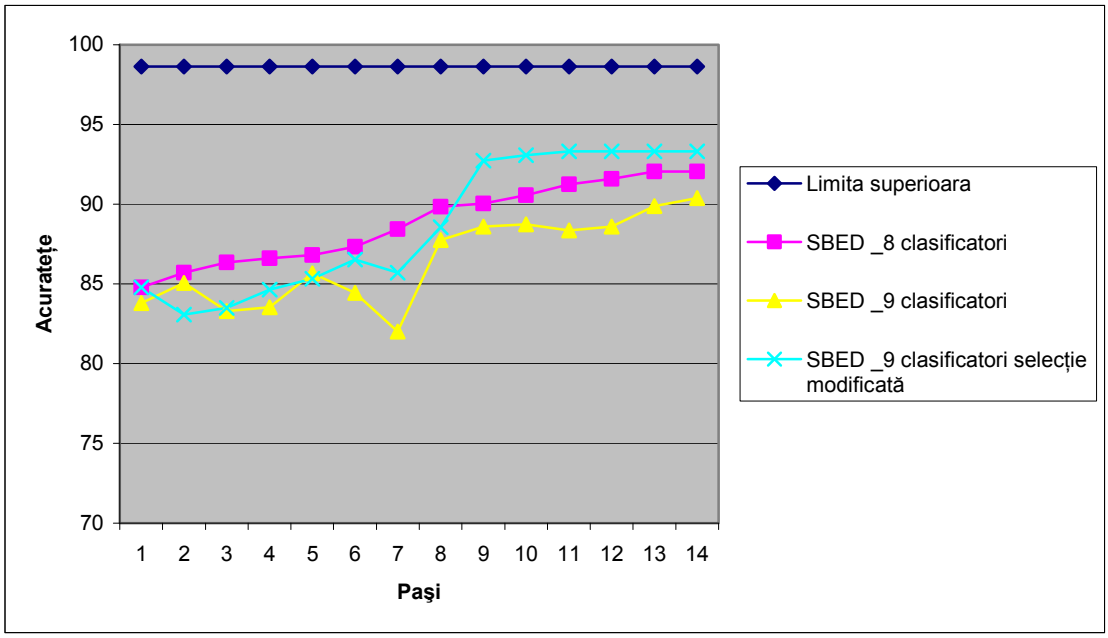

Fig. 5.3 Acuratețea de clasificare obținută de metaclasificatorul cu 9 clasificatori modificat bazat pe distanța euclidiană

Rezultatele obținute de către metaclasificatorul cu 9 clasificatori modificat care se bazează pe distanța euclidiană şi-a îmbunătățit clasificarea obținând o acuratețe a clasificării de **93,32%** față de cel cu 9 clasificatori nemodificat care a obținut doar 90,38%. Reamintim faptul că în aceleaşi condiții metaclasificatorul cu 8 clasificatori de tip SVM din [Mor07] a obținut o acuratețe de clasificare de 92,04%, maximul obținut pe 8 clasificatoare. În primii 7 paşi acuratețea clasificării pentru metaclasificatorul cu 9 clasificatoare modificat este aproape identică

cu cea a metaclasificatorului cu 9 clasificatoare nemodificat deoarece în primii paşi nu apare nici o dată cazul în care trebuie aleasă ce-a de-a doua clasă. În primii paşi rezultatele între cele două metaclasificatoare cu 9 clasificatori cel modificat şi nemodificat sunt puțin diferite deoarece întotdeauna se alege aleator un clasificator din cei 9 existenți.

Rezultatul obținut în cazul selecției bazate pe cosinus.

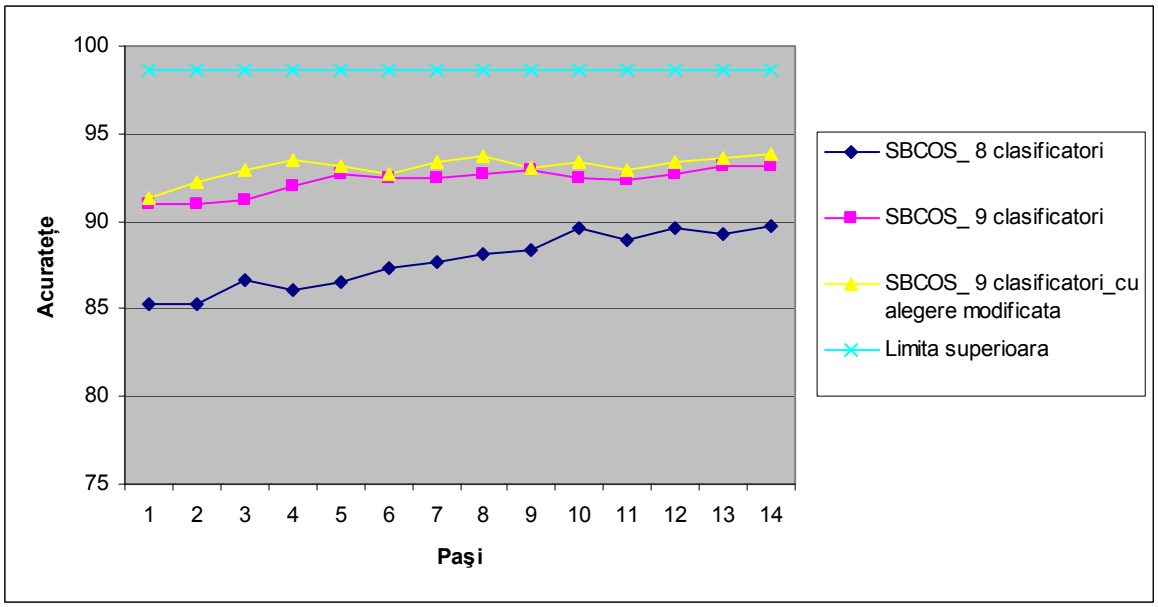

Fig. 5.4 Acuratețea de clasificare a metaclasificatorului cu 9 clasificatori modificat bazat pe cosinus

În acest caz acuratețea de clasificare a metaclasificatorului s-a îmbunătățit de la 93,10% la **93,87%.** Reamintim că metaclasificatorul cu 8 clasificatori de tip SVM în aceleaşi condiții a obținut o acuratețe de clasificare de 89,74%.

6 Concluzii

În această lucrare am încercat să analizăm clasificarea documentelor text cu ajutorul clasificatorilor de tip SVM şi posibilități de îmbunătățire a acurateței de clasificare realizată de metaclasificatorul prezentat în [Mor07]. În prima parte a lucrări am prezentat o parte din experimentele efectuate pentru a analiza dacă este sau nu fezabilă introducerea unui clasificator de alt tip în cadrul metaclasificatorului. Am verificat pe baza datelor de test prezentate în 1.1 rezultatele obținute de clasificatorii de tip SVM prezentați în [Mor07] şi am testat dacă un clasificator de tip Bayes ar putea aduce îmbunătățiri asupra metaclasificatorului în cazul clasificării de documente text. Problemele care apar la clasificarea documentelor text sunt în primul rând dimensiunea foarte mare a vectorilor de reprezentare a documentelor şi în al doilea rând problema existenței claselor suprapuse. Dimensiunea foarte mare a vectorilor de reprezentare face ca nu foarte mulți algoritmi de învățare automată să se preteze la asemenea probleme datorită complexității calculelor care apar şi a timpilor de învățare foarte mari. În cazul bazei de date Reuters 2000 pe care am făcut experimente în acest raport documentele sunt clasificate în mai multe clase ceea ce face posibilă existența de clase suprapuse chiar şi în totalitate făcând imposibilă învățarea pentru majoritatea algoritmilor de clasificare automată.

În capitolul 2 din această lucrare am prezentat problemele care au apărut în cazul metaclasificatorului prezentat în [Mor07] precum şi faptul că selectarea clasificatorilor pe baza celui mai bun rezultat întors poate introduce anumite limitări. Aşa cum s-a prezentat încă din acea lucrare, metaclasificatorul avea o limită maximă la care putea ajunge de 94.02% având un număr de 136 documente care nu puteau fi clasificate corect de nici unul din clasificatoarele selectate. Analizând doar cele 136 documente, în capitolul 2, am observat că există clasificatori SVM prezentați în lucrarea amintită care ar fi reuşit să clasifice corect acele documente dar nu au fost incluşi în metaclasificator. O modificare „statică” a metaclasificatorului nu ne garantează că nu ar apărea un al set de documente imposibil de clasificat. O posibilă rezolvare a acestei probleme ar fi introducerea „dinamică” de noi clasificatori în cadrul metaclasificatorului care ar învăța doar aceste documente ceea ce nu ni se pare o soluție fezabilă.

În capitolul 3 am prezentat partea matematică pentru un clasificator care foloseşte teoria lui Bayes. De asemenea am prezentat câteva experimente realizate cu acest clasificator pe baza de date Reuters 2000 pentru a arăta modificarea acestui tip de clasificator în sensul de a utiliza uniformizarea (normalizarea) lui Laplace îl face fezabil să fie utilizat la clasificarea de vectori de dimensiune foarte mare. Din păcate nu am reuşit să în facem să lucreze cu documente suprapune. În cazul claselor suprapuse am obținut o acuratețe a clasificării de doar 33.17% pentru un

procent de 60% din documente ales pentru antrenare. Dacă eliminăm posibilitatea existenței claselor suprapuse acuratețea de clasificare a ajuns la 75.89%.

În capitolul 4 am prezentat câteva experimente comparative între clasificatorii de tip SVM şi cel de tip Bayes realizate pe toate seturile de date prezentate. Rezultatele prezentate în acest capitol ne-au dat speranța ca introducând în cadrul metaclasificatorului şi un clasificator de tip Bayes să obținem o acuratețe mai bună. Deşi ca şi acuratețe a clasificării pe setul de documente T1 clasificatorul Bayes obține rezultate mai slabe decât SVM (SVM cu nucleu polinomial obține o medie a acurateței de clasificare de 86.01% iar clasificatorul Bayes obține o medie de 81,32%). Ca şi timp de antrenare şi testare clasificatorul Bayes obține cel mai bun timp de 1.7s comparativ cu SVM care în medie obține un timp de 1.8s.

În cazul testării clasificatorul Bayes pe setul T2 (cel care conține doar 136 documente) am obținut rezultate încurajatoare. În urma testelor efectuate, am observat că, deşi acuratețea totală a clasificatorului Bayes generează rezultate mai slabe decât SVM el totuşi a reuşit să clasifice corect 104 documente din cele 136 chiar dacă a fost antrenat pe setul A1.

În capitolul 5 am introdus şi noul clasificator de tip Bayes în metaclasificatorul din [Mor07] obținând o îmbunătățire semnificativă a limitei superioare la care poate ajunge metaclasificatorul. Astfel aceasta a crescut de la 94.21% în cazul folosirii a 8 clasificatoare SVM la 98,63% în cazul folosirii celor 8 clasificatoare SVM plus clasificatorul Bayes.

În acest capitol am prezentat rezultatele obținute pentru toate cele trei modele de metaclasificatori: vot majoritar (MV), selecție pe baza distanței euclidiene (SBED) şi selecție pe bază de cosinus (SBCOS). În cazul MV am obținut o acuratețe a clasificării de doar 86.09%, cu 0.29% mai mică decât în cazul folosirii doar a 8 clasificatori.

În cazul SBED prin modificările aduse noul metaclasificator față de metaclasificatorul cu 8 clasificatori de tip SVM din [Mor07] am obținut rezultate chiar mai slabe scăzând în medie de la 92.04% la 90.38%. În cazul SBCOS acuratețea de clasificare a metaclasificatorului cu 9 clase a crescut la 93.10% de la 89.74% cât era la cel cu 8 clase.

O altă modificare adusă clasificatorului ar fi ca în cazul în care există suspiciunea că clasa care va fi prezisă nu va fi cea corectă acesta să prezică următoarea clasă din lista de clase dacă aceasta este suficient de apropiată de prima clasă prezisă. Aceste modificări au dus la o îmbunătățire substanțială a rezultatelor metaclasificatorului cu 9 clasificatori. Am făcut experimente doar cu acesta deoarece doar in acest caz puteam ajunge la o acuratețe maximă de 98.63%. În cazul SBED am obținut o acuratețe a clasificării de 93.32% iar în cazul SBCOS am obținut o îmbunătățire de la 93.10% la 93.87%.

Ca şi experimente ulterioare ne-am propus realizarea unor noi metaclasificatori neadaptivi precum şi a unui adaptiv bazat pe o rețea neuronală feed-forward de tip backpropagation.

7 Bibliografie

[Gab04] Gabrilovich, E., Markovitch S., *Text Categorization with Many Redundant Features Using Aggressive Feature Selection to Make SVM Competitive with C4.5*, Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.

[Domin97] Domingos, P. and Pazzani, M. *Beyond independence: Conditions for the optimality of the simple bayesian classifier*. Machine Learning, 29, 1997

[Duda73] Duda, R and Hart, P., *Pattern Classification and Scene Analysis*. Wiley, New York. 1973

[Eyher03], Eyheramendy, S., Lewis, D. and Madigan, D. *On the naive bayes model for text categorization*. In: Proceedings Artificial Intelligence & Statistics 2003.

[Lewis98] D. Lewis, *Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval*. In Proceedings of the 10th European Conference on Machine Learning, 1998

[Mann08] Manning, D. C., Raghavan, P. and Schütze, H., *Introduction to Information Retrival,* Cambridge University Press, 2008

[Mor06] Morariu, D., Vintan, L., Tresp, V., *Feature Selection Method for an Improved SVM Classifier*, Proceedings of the 3rd International Conference of Intelligent Systems (ICIS'06), Prague, August, 2006.

[McCall98] McCallum, A and Nigam, K., *A comparison of event models for naive bayes text classification*. In: Proceedings of AAAI-98 Workshop on "Learning for Text Categorization., 1998.

[Mor07] Morariu, Daniel - Contributions to Automatic Knowledge Extraction from Unstructured Data, PhD Thesis, Sibiu, 2007

[Reut00] Misha Wolf and Charles Wicksteed - Reuters Corpus: http://www.reuters.com/researchandstandards/corpus/ lansat în noiembrie 2000, accesat în septembrie 2009

[WEB09] http://www.cs.utexas.edu/~mooney/ir-course/, accesat în ianuarie 2009