Varijante semantičkog indeksiranja i klasifikacija dokumenata

Rumec, Tea

Varijante semantičkog indeksiranja i klasifikacija dokumenata

Rumec, Tea

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Tea Rumec

VARIJANTE SEMANTIČKOG INDEKSIRANJA I KLASIFIKACIJA DOKUMENATA

Diplomski rad

Voditelj rada:
doc. dr. sc. Pavle Goldstein

Zagreb, rujan 2020.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Nadzirano učenje (eng. *supervised learning*) je vrsta strojnog učenja koja prima skup podataka u obliku parova ulaznih (nezavisnih) varijabli, čije vrijednosti zovemo značajkama (eng. *features*) i izlaznu (zavisnu) varijablu, čiju vrijednost zovemo oznaka (eng. *label*). Algoritam nadziranog učenja analizira takav skup podataka i uči generalizirati. Drugim riječima, za podatke oblika *(ulaz, izlaz) = (x,y)* algoritam pronalazi preslikavanje $\hat{y} = f(x)$ na temelju kojeg će moći predvidjeti buduće izlazne vrijednosti novih podataka. Jedan od primjera nadziranog učenja je klasifikacija dokumenata.

Klasifikacija dokumenata je problem koji se javlja u raznim područjima poput detekcije neželjenih poruka elektronske pošte (eng. *spam detection*), raspoznavanja govora, rukopisa, internet pretraživanja ili razvrstavanja članaka u određene rubrike. Zadatak klasifikacije je pridružiti dokument jednoj ili više postojećih kategorija koje nazivamo klasama, pri čemu klasom smatramo skup unaprijed definiranih objekata koji bi trebali imati slične karakteristike. Dokumenti koje klasificiramo mogu biti tekstualni, slikovni i dr., a mogu se klasificirati po raznim karakteristikama: autoru, godini izdavanja, vrsti dokumenta, temi itd. U ovom radu bavit ćemo se klasifikacijom tekstualnih dokumenata po sadržaju.

Cilj ovog rada je proučiti i usporediti kako detekcija ključnih riječi i korištenje različitih mjera koje daju težine pojmovima u tekstu utječu na rezultate klasifikacije. Također, usporedit ćemo rezultate dobivene različitim metodama linearne klasifikacije, točnije klasifikaciju normama (max-norma, 1-norma i 2-norma) te stroj potpornih vektora (eng. *Support Vector Machine - SVM*).

Ovaj rad se sastoji od 4 poglavlja. Prvo poglavlje nam služi kao uvod u matematičke pojmove iz linearne algebre i optimizacije, koje ćemo koristiti u nastavku rada. U drugom poglavlju ćemo opisati skup podataka kojim se bavimo te napraviti pripremu i reprezentaciju teksta kako bi se metode klasifikacije mogle uspješno primijeniti. Upravo te razne metode klasifikacije ćemo predstaviti i objasniti u trećem poglavlju, dok ćemo rezultate svih metoda prikazati u zadnjem, četvrtom poglavlju.

Poglavlje 1

Matematički pojmovi

Na početku ovoga rada, moramo uvesti neke osnovne matematičke pojmove iz linearne algebre i optimizacije, koji će nam pomoći u razumijevanju idućih poglavlja. Definicije i tvrdnje iz linearne algebre preuzete su iz [1], a optimizacija je preuzeta iz [2].

1.1 Linearna algebra

Definicija 1.1.1. *Neka je $\mathbb{F}$ neki skup na kojem su zadane binarne operacije zbrajanja*

$$+ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

i množenja

$$\cdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$$

koje imaju sljedeća svojstva:

1. $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma, \ \forall \alpha, \beta, \gamma \in \mathbb{F}$;
2. $\exists\, 0 \in \mathbb{F}$ *sa svojstvom* $\alpha + 0 = 0 + \alpha = \alpha, \ \forall \alpha \in \mathbb{F}$;
3. $\forall\, \alpha \in \mathbb{F}, \ \exists - \alpha \in \mathbb{F}$ *tako da je* $\alpha + (-\alpha) = (-\alpha) + \alpha = 0$;
4. $\alpha + \beta = \beta + \alpha, \ \forall \alpha, \beta \in \mathbb{F}$;
5. $\alpha(\beta\gamma) = (\alpha\beta)\gamma, \ \forall \alpha, \beta, \gamma \in \mathbb{F}$;
6. $\exists\, 1 \in \mathbb{F} \setminus \{0\}$ *sa svojstvom* $1 \cdot \alpha = \alpha \cdot 1 = \alpha, \ \forall \alpha \in \mathbb{F}$;
7. $\forall \alpha \in \mathbb{F}, \alpha \neq 0, \exists\, \alpha^{-1} \in \mathbb{F}$ *tako da je* $\alpha\alpha^{-1} = \alpha^{-1}\alpha = 1$;

8. $\alpha\beta = \beta\alpha, \forall\alpha, \beta \in \mathbb{F}$;

9. $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma,\ \forall\alpha, \beta, \gamma \in \mathbb{F}$.

Tada kažemo da je $\mathbb{F}$ polje. Elemente polja $\mathbb{F}$ nazivamo skalarima.

Od sada nadalje, oznaka $\mathbb{F}$ će označavati $\mathbb{R}$ ili $\mathbb{C}$.

Definicija 1.1.2. *Neka je V neprazan skup na kojem su zadane binarna operacija zbrajanja $+ : V \times V \to V$ i operacija množenja skalarima iz polja $\mathbb{F}$, $\cdot : \mathbb{F} \times V \to V$. Kažemo da je uređena trojka $(V, +, \cdot)$ vektorski prostor nad poljem $\mathbb{F}$ ako vrijedi:*

1. $a + (b + c) = (a + b) + c,\ \forall a, b, c \in V$;

2. $\exists\, 0 \in V$ *sa svojstvom* $a + 0 = 0 + a = a,\ \forall a \in V$;

3. $\forall a \in V,\ \exists - a \in V$ *tako da je* $a + (-a) = (-a) + a = 0$;

4. $a + b = b + a,\ \forall a, b \in V$;

5. $\alpha(\beta a) = (\alpha\beta)a,\ \forall\alpha, \beta \in \mathbb{F},\ \forall a \in V$;

6. $(\alpha + \beta)a = \alpha a + \beta a,\ \forall\alpha, \beta \in \mathbb{F},\ \forall a \in V$;

7. $\alpha(a + b) = \alpha a + \alpha b,\ \forall\alpha \in \mathbb{F},\ \forall a, b, \in V$;

8. $1 \cdot a = a \cdot 1,\ \forall a \in V$.

Elemente vektorskog prostora nazivamo vektori.

Vektorski prostori nad poljem $\mathbb{R}$ nazivaju se realni vektorski prostori, a za one nad poljem $\mathbb{C}$ kažemo da su kompleksni.
Neka je $n \in \mathbb{N}$, te neka $\mathbb{R}^n$ označava skup svih uređenih n-torki realnih brojeva (drugim riječima, $\mathbb{R}^n$ je Kartezijev produkt od n kopija skupa $\mathbb{R}$). Definirajmo

$$(a_1, a_2, ..., a_n) + (b_1, b_2, ..., b_n) = (a_1 + b_1, a_2 + b_2, ..., a_n + b_n)$$

i za $\alpha \in \mathbb{R}$

$$\alpha(a_1, a_2, ..., a_n) = (\alpha a_1, \alpha a_2, ..., \alpha a_n)$$

Jasno je da je uz ovako definirane operacije $\mathbb{R}^n$ realan vektorski prostor.

Definicija 1.1.3. *Neka je V vektorski prostor nad $\mathbb{F}$. Izraz oblika*

$$\alpha_1 a_1 + \alpha_2 a_2 + ... + \alpha_k a_k$$

pri čemu je $a_1, a_2, ..., a_k \in V$, $\alpha_1, \alpha_2, ..., \alpha_k \in \mathbb{F}$ *i* $k \in \mathbb{N}$, *naziva se linearna kombinacija vektora* $a_1, a_2, ..., a_k$ *s koeficijentima* $\alpha_1, \alpha_2, ..., \alpha_k$.

Definicija 1.1.4. *Neka je V vektorski prostor nad* $\mathbb{F}$ *i*

$$S = \{a_1, a_2, ..., a_k\},\ k \in \mathbb{N}$$

konačan skup vektora iz V. Ako vrijedi

$$\forall \alpha_1, \alpha_2, ..., \alpha_k \in \mathbb{F},\ \ \textstyle\sum_{i=1}^{k} \alpha_i a_i = 0 \ \Rightarrow \alpha_1 = \alpha_2 = ... = \alpha_k = 0.$$

kažemo da je skup S linearno nezavisan. U suprotnom kažemo da je skup S linearno zavisan.

Definicija 1.1.5. *Neka je V vektorski prostor nad poljem* $\mathbb{F}$ *i* $S \subseteq V$, $S \neq \emptyset$. *Linearna ljuska skupa S označava se simbolom* $[S]$ *i definira kao*

$$[S] = \{\textstyle\sum_{i=1}^{k} \alpha_i a_i :\ \alpha_i \in \mathbb{F}, \alpha_i \in S,\ k \in \mathbb{N}\}.$$

Dodatno, definira se $[\emptyset] = \{0\}$.

Linearna ljuska nepraznog skupa S je, dakle, skup svih linearnih kombinacija elemenata skupa S.

Definicija 1.1.6. *Neka je V vektorski prostor i* $S \subseteq V$. *Kaže se da je S sustav izvodnica za V (ili da S generira V) ako vrijedi* $[S] = V$.

Definicija 1.1.7. *Konačan skup* $B = \{b_1, b_2, ..., b_n\}$, $n \in \mathbb{N}$, *u vektorskom prostoru V, naziva se baza za V ako je B linearno nezavisan sustav izvodnica za V.*

Sljedeći teorem je fundamentalan rezultat linearne algebre. Smisao je u tome da svaki vektor danog prostora možemo na jedinstven način predočiti kao linearnu kombinaciju vektora baze. Na ovaj se način svaki problem i svaki račun u tom prostoru može svesti na operiranje s konačno mnogo vektora.

Teorem 1.1.8. *Neka je V vektorski prostor nad poljem* $\mathbb{F}$, *te neka je* $B = \{b_1, b_2, ..., b_n\}$ *baza za V. Tada za svaki vektor* $v \in V$ *postoje jedinstveno određeni skalari* $\alpha_1, ..., \alpha_n \in \mathbb{F}$ *takvi da vrijedi* $v = \sum_{i=1}^{n} \alpha_i b_i$.

Definicija 1.1.9. *Neka je V vektorski prostor nad poljem* $\mathbb{F}$. *Skalarni produkt na V je preslikavanje*

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$$

sa sljedećim svojstvima:

1. $\langle x, x\rangle \geq 0,\ \forall x \in V$;
2. $\langle x, x\rangle = 0 \iff x = 0$;
3. $\langle x_1 + x_2, y\rangle = \langle x_1, y\rangle + \langle x_2, y\rangle,\ \forall x_1, x_2, y \in V$;
4. $\langle \alpha x, y\rangle = \alpha\langle x, y\rangle,\ \forall \alpha \in \mathbb{F},\ \forall x, y \in V$;
5. $\langle x, y\rangle = \overline{\langle y, x\rangle},\ \forall x, y \in V$.

Treba primijetiti da skalarni produkt poprima vrijednosti u polju nad kojim je dani vektorski prostor izgrađen; ako je, dakle, prostor kompleksan, zadnje svojstvo kaže da su skalarni umnošci $\langle x, y\rangle$ i $\langle y, x\rangle$ međusobno konjugirano kompleksni brojevi. Ako je pak prostor realan, skalarni umnožak bilo koja dva vektora je realan broj pa kompleksno konjugiranje nema efekta i ovo svojstvo u realnim prostorima glasi:

$$\langle x, y\rangle = \langle y, x\rangle.$$

Stoga se u realnim prostorima svojstvo (5) naziva simetričnost, a u kompleksnim prostorima hermitska simetričnost. Nas će u ovom radu zanimati realan prostor.

Definicija 1.1.10. *Vektorski prostor na kojem je definiran skalarni produkt zove se unitaran prostor.*

Neka je $\langle (x_1, x_2, ..., x_n), (y_1, y_2, ..., y_n)\rangle = \sum_{i=1}^{n} x_i y_i$. Ovako definirano preslikavanje je skalarni produkt u $\mathbb{R}^n$.

Definicija 1.1.11. *Neka je V unitaran prostor. Kažemo da su vektori $x, y \in V$ međusobno okomiti ili ortogonalni (oznaka: $x \perp y$) ako je $\langle x, y\rangle = 0$.*

Definicija 1.1.12. *Neka je V unitaran prostor. Norma na V je funkcija*

$$\|\cdot\| : V \to \mathbb{R}$$

definirana s

$$\|x\| = \sqrt{\langle x, x\rangle}.$$

$(V, \|\cdot\|)$ *nazivamo normirani prostor.*

Propozicija 1.1.13. *Norma na unitarnom prostoru V ima sljedeća svojstva:*

1. $\|x\| \geq 0,\ \forall x \in V$;
2. $\|x\| = 0 \iff x = 0$;

3. $||\alpha x|| = |\alpha|\,||x||,\ \forall \alpha \in \mathbb{F},\ \forall x \in V$;

4. $||x + y|| \leq ||x|| + ||y||,\ \forall x, y \in V$.

Definicija 1.1.14. *Neka je V unitaran prostor. Kaže se da je vektor $x \in V$ normiran ako je* $||x|| = 1$.

Neka je dan vektor $x = (x_1, ..., x_n) \in V$ nad poljem $\mathbb{R}^n$.

- 1-norma dana je sa $||x||_1 = \sum_{i=1}^{n} |x_i|$
- Euklidska ili 2-norma dana je sa $||x||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$.
- Max-norma dana je sa $||x||_{max} = \max\{|x_1|, ..., |x_n|\}$.

Propozicija 1.1.15. *Metrika ili udaljenost je svaka funkcija $d : V \times V \to \mathbb{R}$ sa sljedećim svojstvima:*

1. $d(x, y) \geq 0,\ \forall x, y \in V$;

2. $d(x, y) = 0 \iff x = y$;

3. $d(x, y) = d(y, x),\ \forall x, y, \in V$;

4. $d(x, y) \leq d(x, z) + d(z, y),\ \forall x, y, z \in V$.

(V, d) nazivamo metrički prostor.

Definicija 1.1.16. *Neka je V vektorski prostor nad $\mathbb{F}$ i $M \subseteq V$, $M \neq \emptyset$. Ako je i $(M, +, \cdot)$ vektorski prostor nad $\mathbb{F}$ uz iste operacije iz V, kažemo da je M potprostor od V.*

Ovakvu situaciju možemo zamišljati kao jedan vektorski prostor smješten u drugome. Kasnije u radu bit će situacija kada ćemo vektore iz vektorskog prostora preslikavati u njegove potprostore. U tu svrhu uvodimo sljedeći pojam.

Definicija 1.1.17. *Neka su V i W vektorski prostori nad istim poljem $\mathbb{F}$. Preslikavanje $A : V \to W$ zove se linearan operator ako vrijedi*

$$A(\alpha x + \beta y) = \alpha A x + \beta A y, \quad \forall \alpha, \beta \in \mathbb{F},\ x, y \in V$$

Vektorski prostor linearnih operatora sa skupa V u V označavat ćemo s L(V).

Definicija 1.1.18. *Neka je V konačnodimenzionalan unitaran prostor i A linearan operator na V. Operator A^* sa svojstvom*

$$\langle Ax, y\rangle = \langle x, A^* y\rangle, \quad \forall x, y \in V$$

zove se hermitski adjungiran operator operatoru A.

Definicija 1.1.19. *Neka je V konačnodimenzionalan vektorski prostor i $P \in L(V)$. Operator P zovemo projektor, ako je $P^2 = P$. Ako vrijedi $P^2 = P = P^*$ operator P je ortogonalni projektor.*

Po definiciji, projektor P je idempotentni operator.
Neka je W konačnodimenzionalan vektorski prostor i P projektor na W. Neka su potprostori U i V redom slika i jezgra operatora P. Tada P ima sljedeća svojstva:

1. P je identiteta na U: $\quad \forall x \in U : Px = x$.
2. Vrijedi direktna suma $W = U \bigoplus V$. Svaki vektor $x \in W$ se može napisati kao jedinstvena dekompozicija $x = u + v$, pri čemu su $u = Px$ i $v = x - Px = (I - P)x$, za $u \in U, v \in V$.

Definicija 1.1.20. *Za prirodne brojeve m i n, preslikavanje*

$$A : \{1, 2, ..., m\} \times \{1, 2, ..., n\} \to \mathbb{F}$$

naziva se matrica tipa (m,n) s koeficijentima iz polja $\mathbb{F}$. Takve funkcije pišemo tablično, u m redaka i n stupaca, gdje u i-tom retku i j-tom stupcu piše vrijednost $A(i, j)$ koju ćemo kraće označavati kao a_{ij}. Tada ćemo matricu A sa elementima a_{ij} označavati sa $A = [a_{ij}]$. Skup svih matrica tipa (m,n) označavamo $M_{mn}(\mathbb{F})$. Ako je $m = n$ pišemo kraće $M_n(\mathbb{F})$, a elemente tog skupa nazivamo kvadratnim matricama reda n.

Definicija 1.1.21. *Neka je $A \in M_{mn}(\mathbb{R})$. Transponirana matrica A^T matrice $A = [a_{ij}]$ definirana je sa $A^T = [a_{ji}]$.*

1.2 Optimizacija

Problem određivanja ekstrema neke funkcije, uz zadane uvjete, naziva se problem matematičkog programiranja. Funkciju čiji je minimum ili maksimum potrebno odrediti nazivamo funkcija cilja. Kada je funkcija cilja linearna, te ako su uvjeti izraženi u obliku linearnih jednadžbi i/ili nejednadžbi govorimo o problemu linearnog programiranja. Slično, kada je funkcija cilja kvadratna, govorimo o problemu kvadratnog programiranja.

Definicija 1.2.1. *Neka je $\Omega \subseteq \mathbb{R}^n$ otvoren skup. Kažemo da funkcija $f : \Omega \to \mathbb{R}$ ima lokalni minimum u točki $P_0 \in \Omega$ ako postoji okolina $K(P_0, r) \subseteq \Omega$ takva da*

$$(\forall P \in \{K(P_0, r) \backslash P_0\}) \ \ (f(P) \geq f(P_0)),$$

odnosno funkcija f u $P_0 \in \Omega$ ima lokalni maksimum ako vrijedi

$$(\forall P \in \{K(P_0, r)\backslash P_0\}) \ \ (f(P) \leq f(P_0)).$$

Vrijednosti $f(P_0)$ zovemo minimumom, odnosno maksimumom funkcije f na skupu Ω. Ako vrijede stroge nejednakosti, govorimo o strogom lokalnom minimumu, odnosno strogom lokalnom maksimumu. Ako nejednakosti vrijede za svaku točku $P \in \Omega$, tada funkcija f u točki P_0 ima globalni minimum, odnosno globalni maksimum.

Definicija 1.2.2. *Neka je $\Omega \subseteq \mathbb{R}^n$ otvoren skup i neka je $f : \Omega \to \mathbb{R}^n$ diferencijabilna funkcija. Za točku $P_0 \in \Omega$ kažemo da je stacionarna točka funkcije f ako vrijedi:*

$$\partial_i f(P_0) = 0, \ \ i = 1, 2, ..., n.$$

Teorem 1.2.3. *(Nužan uvjet za postojanje lokalnog ekstrema) Ako je $P_0 \in \Omega \subseteq \mathbb{R}^n$ točka lokalnog ekstrema diferencijabilne funkcije $f : \Omega \to \mathbb{R}$, onda je P_0 stacionarna točka funkcije f, tj. vrijedi:*

$$\partial_i f(P_0) = 0, \ \ i = 1, 2, ..., n.$$

Neka su zadane funkcije $f, g_i : \mathbb{R}^n \to \mathbb{R}, \ \ i = 1, 2, ..., m$. Promatramo sljedeći optimizacijski problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$g_i(x) \leq 0, \ \ i = 1, 2, ..., m.$$

Skup $U = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \ \ i = 1, 2, ..., m\}$ zovemo *dopustivo područje*, a svaki $x \in U$ zovemo *dopustivo rješenje*. Dopustivo rješenje x^* za koje vrijedi $f(x^*) \leq f(x)$ zovemo *optimalno dopustivo rješenje*.

Gornjem problemu možemo pridružiti funkciju $L : \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$ zadanu formulom

$$L(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x).$$

Funkciju L zovemo *Lagrangeova funkcija* koja je pridružena problemu.

Teorem 1.2.4. *Problem*

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$g_i(x) \leq 0, \ \ i = 1, 2, ..., m.$$

ekvivalentan je problemu

$$\min_{x\in\mathbb{R}^n}\max_{\alpha\in\mathbb{R}^m_+} L(x,\alpha).$$

Problem iz teorema zovemo *primarni problem*, a jer je rješenje primarnog problema ujedno rješenje originalnog optimizacijskog problema, njega također zovemo primarni problem. Možemo promatrati sljedeći optimizacijski problem

$$\max_{\alpha\in\mathbb{R}^m_+}\min_{x\in\mathbb{R}^n} L(x,\alpha),$$

kojeg zovemo *dualni problem*. Pretpostavimo da je zadan problem linearnog programiranja

$$\begin{cases} f(x) = c^T x \to \min_x \\ Ax \geq b \end{cases}$$

pri čemu su $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$. Lagrangeovu funkciju $L : \mathbb{R}^n \times \mathbb{R}^m_+ \to \mathbb{R}$ definiramo formulom

$$L(x,\alpha) = c^T x + \alpha^T(b - Ax).$$

Odgovarajući primarni i dualni problem tada su

$$\min_{x\in\mathbb{R}^n}\max_{\alpha\in\mathbb{R}^m_+} L(x,\alpha),$$

$$\max_{\alpha\in\mathbb{R}^m_+}\min_{x\in\mathbb{R}^n} L(x,\alpha).$$

Poglavlje 2

Reprezentacija teksta

2.1 Priprema teksta

Zadatak klasifikacije je pridružiti dokument jednoj od postojećih klasa. U ovom radu ćemo klasificirati tekstualne dokumente. Konkretnije, koristit ćemo 3 kolekcije sažetaka članaka na engleskom jeziku: CRAN, MED i CISI kolekciju. CRAN kolekcija se sastoji od 1400 članaka o aeronautici prikupljenih na sveučilištu u Cranfieldu, MED kolekcija se sastoji od 1033 članka medicinske tematike iz Medline časopisa, a CISI kolekcija se sastoji od 1460 članaka tematski vezanih uz informacijske znanosti, koje je sakupio Centar za izume i znanstvene informacije (eng. ***C**enter for **I**nventions and **S**cientific **I**nformation*). Sve kolekcije se mogu preuzeti sa servera odsjeka računalnih znanosti sveučilišta u Glasgowu [4].

U nastavku rada, članke ćemo nazivati *dokumentima*, a kolekcije *klasama*. Primjer jednog dokumenta iz klase CISI možemo vidjeti na slici 2.1. Za analizu nas zanima tekst sažetka pa ćemo u dokumentu izdvojiti paragraf *"The present study is a history...."*. Analogno za ostale dokumente. Kako bi datoteke mogli koristiti za klasifikaciju, tekst datoteke moramo svesti na smislenu formu.

Za početak znamo da programski jezici velika i mala slova prepoznaju kao različite znakove. U korištenim datotekama poneke riječi su pisane početnim velikim slovom ili kombinacijom velikih i malih slova pa trebamo proći kroz sve datoteke i svako veliko slovo na koje naiđemo pretvoriti u malo. Nadalje, za analizu teksta nam nisu potrebni interpunkcijski znakovi i oznake za novi redak pa ćemo ih ukloniti, skupa sa brojevima. Sljedeće, primijetimo da u primjeru dokumenta postoji mnogo riječi koje ne pridaju značenju teksta kao što su veznici, prilozi, prijedlozi i sl. Takve riječi se u engleskom jeziku nazivaju *stop words*. Alat za obradu prirodnog jezika NLTK (eng. *Natural language toolkit*) sadrži popis stop words u engleskom jeziku pa koristeći NLTK biblioteke u programskom jeziku Python jednostavno uklonimo sve takve riječi iz svih dokumenata. Za kraj pripreme teksta, NLTK u svojoj biblioteci ima koristan algoritam pod nazivom *Porter Stemmer*.

```
.I 1
.T
18 Editions of the Dewey Decimal Classifications
.A
Comaromi, J.P.
.W
   The present study is a history of the DEWEY Decimal
Classification.  The first edition of the DDC was published
in 1876, the eighteenth edition in 1971, and future editions
will continue to appear as needed.  In spite of the DDC's
long and healthy life, however, its full story has never
been told.  There have been biographies of Dewey
that briefly describe his system, but this is the first
attempt to provide a detailed history of the work that
more than any other has spurred the growth of
librarianship in this country and abroad.
```

Slika 2.1: Primjer dokumenta prije čišćenja teksta

Stemming - korjenovanje

Porter Stemmer algoritam je stemming algoritam koji skraćuje riječi na korijen (eng. *stem*) tako da obriše završetak riječi, obično sufiks. Stemming možemo nazvati korjenovanjem. U engleskom jeziku najčešći sufiksi koje ćemo ukloniti su *"-ed, -ing, -ion, -ions"*. Za razliku od lematizacije, dobiveni korijen ne mora biti isti kao morfološki korijen riječi; dovoljno je da se značenjem povezane riječi skrate na isti korijen, bez obzira je li taj korijen valjana riječ. Na primjer, Porter Stemmer će riječi *"trouble, troubling i troubled"* skratiti na korijen *"troubl"*.

Korjenovanje nam je korisno jer ćemo značenjem slične riječi svesti na jednu zajedničku riječ te ćemo time smanjiti dimenziju prostora. Algoritam sadrži popis sufiksa i sa svakim sufiksom kriterij po kojem on može biti odvojen od riječi kako bi dobili valjani korijen. S obzirom na složenost engleskog jezika i gramatike, može se dogoditi da algoritam skrati riječi različitog značenja na isti korijen. Na primjer, algoritam riječi *"experience" i "experiment"* skrati na riječ *"experi"*, iako se te riječi po značenju ne bi trebale spojiti u jedan korijen. Sa dodatnim uvjetima u algoritmu taj problem bi se možda mogao riješiti za svaki od takvih slučajeva posebno, ali s obzirom na mali broj loših primjera skraćivanja, možemo prihvatiti malu mogućnost pogreške. Najveća prednost Porter Stemmera je njegova brzina i jednostavnost. Detaljan prikaz algoritma i pravila po kojima Porter Stemmer skraćuje riječi nalazi se u [5].

Konačno, dokument sa početka poglavlja prelazi u 2.2. Dokument se sada sastoji od niza ključnih riječi koje će ukazivati na sadržaj dokumenta.

```
present studi histori dewey decim classif first edit ddc
publish eighteenth edit futur edit continu appear need spite
ddc long healthi life howev full stori never told biographi
dewey briefli describ system first attempt provid detail
histori work spur growth librarianship countri abroad
```

Slika 2.2: Primjer dokumenta nakon čišćenja teksta

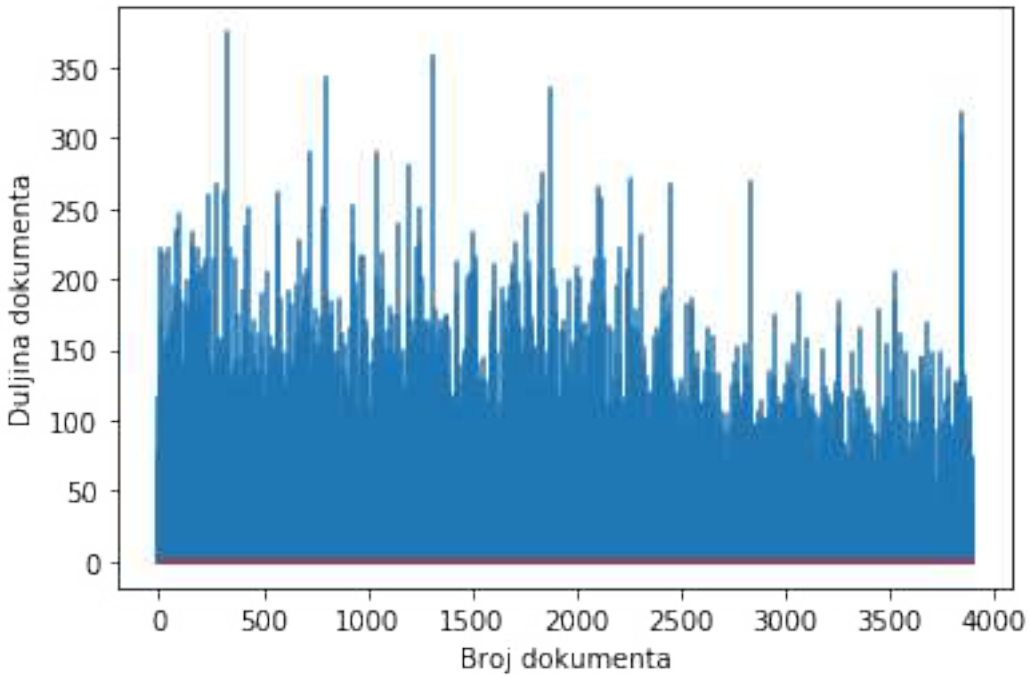
Također treba napomenuti da se u procesu pripreme teksta primijetilo da klasa CRAN sadrži dva prazna dokumenta pa smo ih izbacili iz klase. Novi ukupni broj dokumenata u klasi CRAN je 1398.

2.2 Vektorska reprezentacija teksta

Kao što smo naveli u prošlom odjeljku, imamo 1398 dokumenata iz CRAN klase, 1033 dokumenta iz MED klase i 1460 dokumenata iz CISI klase, što u sumi daje 3891 dokument. Dobro je napomenuti da nisu svi dokumenti jednake duljine, pri čemu pod duljinom smatramo broj riječi u dokumentu nakon čišćenja teksta. Na slici 2.3 možemo vidjeti varijabilnost duljine pojedinih dokumenata.

Kako bi dokumente mogli koristiti za daljnju analizu, ideja nam je prikazati ih u obliku vektora značajki. Prvo što moramo napraviti je formirati bazu ključnih riječi koja će predstavljati bazu vektorskog prostora. S obzirom da smo svaki dokument nakon čišćenja teksta prikazali kao niz ključnih riječi, potrebno je proći svakim dokumentom i izdvojiti svaku od tih riječi (bez ponavljanja), pri čemu nam nije bitan poredak pojavljivanja riječi. Ovakav pristup se naziva *bag-of-words* model (vreća riječi), često korišten u obradi prirodnog jezika i dohvatu informacija (eng. *Information retrieval - IR*). Ukupan broj riječi dobivenih ovim postupkom je 13112.

Formiramo matricu u kojoj svaki redak predstavlja jedan dokument, a svaki stupac predstavlja jedan pojam iz baze. Takva matrica se naziva *document-term matrica*. Dakle, document-term matrica je dimenzija 3891×13112 te se na (i, j)-tom mjestu nalazi frekvencija pojavljivanja j-tog pojma u i-tom dokumentu. Samim time nam i-ti redak predstavlja $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,13112})$ vektor značajki za i-ti dokument, $i = 1, ..., 3891$. Dokumenti iz različitih klasa su poredani tako da se u prvih 1398 redaka nalaze dokumenti iz CRAN klase, u idućih 1033 redaka dokumenti iz MED klase te u zadnjih 1460 redaka dokumenti iz CISI klase. Također, napravili smo i vektor oznaka, gdje na i-tom mjestu stoji oznaka



Slika 2.3: Duljine dokumenata

kojoj klasi *i*-ti dokument pripada: oznaka 1 ako pripada klasi CRAN, oznaka 2 ako pripada klasi MED i oznaka 3 ako pripada klasi CISI.

2.3 Pristupi računanju težina pridruženih pojmovima u dokumentu

Document-term matricu smo formirali tako da elemente matrice čine frekvencije pojavljivanja pojmova u dokumentu. No same frekvencije pojavljivanja pojmova ne mogu uvijek osigurati dobre informacije o važnosti tog pojma u dokumentima. Usko prateći izvor [6] uvest ćemo nove faktore, kako bi provjerili hoće li klasifikacija biti bolja sa novim vrijednostima elemenata matrice.

Prvi problem se javlja ako pojmovi sa velikom frekvencijom nisu koncentrirani u par dokumenata nego su prisutni u cijeloj kolekciji. Tada ćemo dobiti informaciju da su svi dokumenti bitni i relevantni za taj pojam, što nam ne pomaže u klasifikaciji. Zbog toga se uvodi novi faktor pod nazivom inverzna frekvencija dokumenta (eng. *inverse document frequency*) koji će favorizirati pojmove koji se nalaze u samo par dokumenata u kolekciji.

Inverzna frekvencija dokumenta se najčešće računa kao logaritam omjera ukupnog broja dokumenata u kolekciji N i broja dokumenata u kojima se pojavljuje određeni pojam n:

$$log(N/n) \tag{2.1}$$

Drugi problem se javlja zbog nejednake duljine dokumenata u kolekciji, što je prikazano na slici 2.3 u prošlom odjeljku. Veći dokumenti imaju više riječi pa su im i frekvencije veće pa će dulji dokumenti imati veću vjerojatnost da budu izabrani. Naravno, mi bi htjeli da klasifikacija ne ovisi o duljini dokumenta te da svi dokumenti budu ravnopravni. Zato uvodimo novi faktor, faktor normalizacije (eng. *normalization factor*) koji će izjednačiti duljine svih dokumenata. Faktor normalizacije normira svaki element document-term matrice, odnosno element podijeli sa normom retka (dokumenta) u kojem se element nalazi:

$$\frac{x_{i,j}}{||x_i||}, i = 1, ..., 3891, j = 1, ..., 13112 \tag{2.2}$$

Množenjem triju faktora, frekvencije pojma u dokumentu, inverzne frekvencije dokumenta i faktora normalizacije, dobit ćemo nove vrijednosti document-term matrice. Zanima nas utjecaj različitih vrijednosti na klasifikaciju dokumenata. Uspoređivat ćemo 5 varijanti.

Prva varijanta je kada promatramo samo standardne frekvencije tj. broj puta koliko se pojam pojavio u zadanom dokumentu. Oznaka je txx. To je naša početna document-term matrica.

Druga varijanta je kada standardne frekvencije normaliziramo, odnosno kada standardne frekvencije txx pomnožimo sa faktorom normalizacije. Oznaka je txc, a nove vrijednosti su u (2.2).

Množenjem standardnih frekvencija txx sa inverznom frekvencijom dokumenta iz (2.1) dobivamo treću varijantu, u oznaci tfx.

Ako vrijednost tfx iz treće varijante još normiramo tj. pomnožimo faktorom normalizacije iz (2.2) dobivamo četvrtu varijantu, u oznaci tfc.

Postoji mogućnost da uopće ne koristimo standardne frekvencije txx nego umjesto toga frekvenciju pojma promatramo binarno: ako se pojam pojavio u dokumentu označimo ga s 1, inače sa 0. Takav binaran zapis frekvencije sada pomnožimo sa inverznom frekvencijom dokumenta iz (2.1) i dobivamo petu varijantu, u oznaci bfx.

Poglavlje 3

Klasifikacija dokumenata

3.1 Projekcije

Skup podataka potrebno je podijeliti na trening i test skup. Trening skup nam služi za analizu i izgradnju rješenja, a na test skupu testiramo rješenje i procjenjujemo koliko su točni rezultati. Iako je uobičajeno da veći dio podataka bude u trening skupu, a manji u test skupu, u našem slučaju podatke ćemo podijeliti u omjeru 50:50. U ovom poglavlju opisat ćemo postupak reduciranja baze tražeći karakteristične riječi za svaku klasu. Postupak ćemo pokazati prvo za dvije, a zatim i za tri klase. Preslikavanje vektora baze iz prostora veće dimenzije u prostor manje dimenzije naziva se *projekcija*. Klasifikacija dokumenata se zatim provodi tako da se dokumenti iz test seta prikažu kao vektori u novoj reduciranoj bazi, a zatim na svakom dijelu karakterističnom za pojedinu klasu izračunamo normu te klasificiramo dokument tamo gdje se postigne najveća norma. U zadnjem poglavlju 4 vidjet ćemo kako odabir norme (max, l_1 i l_2 norma) utječe na klasifikaciju dokumenata.

CRAN i MED klase

Na početku se bavimo problemom klasifikacije dvije klase, CRAN i MED. U poglavlju 2.2 dokumente iz sve tri klase smo prikazali u document-term matrici, u kojoj su u retcima redom poredani dokumenti iz CRAN, MED i CISI klase, a u stupcima pojmovi iz baze. Budući da u ovom dijelu rada nećemo raditi sa klasom CISI, uzimamo samo prvih 2431 redaka matrice: 1398 predstavlja dokumente iz klase CRAN, 1033 iz klase MED. Skup podataka dijelimo na trening i test skup pa dobivamo 1215 dokumenata u trening, a 1216 dokumenata u test skupu. Baza vektorskog prostora sada sadrži 10635 elemenata.

Sada ćemo dokumente iz trening skupa razdvojiti u dvije matrice: u matrici M_1 će se nalaziti frekvencije pojavljivanja vektora baze u klasi CRAN, a u matrici M_2 frekvencije pojavljivanja vektora baze u klasi MED. Zanima nas koje su to karakteristične riječi koje

se češće pojavljuju u jednoj klasi nego u drugoj. Tada možemo pretpostaviti da takve riječi dobro opisuju klasu u kojoj se nalaze. Zatim ćemo reducirati bazu tako da nju čine upravo te karakteristične riječi.

Prvo želimo vidjeti koliko puta se koji pojam iz baze pojavljuje u cijeloj klasi, ali posebno za CRAN i posebno za MED klasu. To ćemo dobiti tako što ćemo zbrojiti sve frekvencije pojavljivanja određenog pojma u svim dokumentima, a to zapravo znači da je potrebno zbrojiti retke prethodno definiranih matrica M_1 i M_2. Ovim postupkom dobivamo dva vektora, po jedan za svaku klasu, gdje se na i-tom mjestu nalazi frekvencija pojavljivanja i-tog pojma iz baze u toj klasi. Vektor dobiven iz matrice M_1 označavamo sa v_1, a vektor dobiven iz matrice M_2 sa v_2.

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,10635} \\ \vdots & x_{i,j} & \vdots \\ x_{704,1} & \cdots & x_{704,10635} \end{bmatrix}$$

$$v_1 = [\ \underset{\text{investig}}{\ldots}\quad \underset{\text{revert}}{x_j}\quad \underset{\text{root}}{\ldots}\ ]$$

Sada želimo usporediti vektore v_1 i v_2, odnosno želimo promotriti razlike u frekvencijama pojavljivanja pojmova u klasama. Prvo ćemo izračunati vektor $v_1 - v_2$ te ćemo uzeti 1000 najvećih elemenata istovremeno pamteći koji su to elementi baze. Time smo izdvojili 1000 riječi koje se češće pojavljuju u klasi CRAN nego u klasi MED. Dobili smo novu, manju bazu karakterističnih riječi za klasu CRAN, koju ćemo označiti sa B_1. Sada analogno računamo vektor $v_2 - v_1$, uzmemo 1000 pozicija sa najvećim elementima i pamtimo koji se elementi baze nalaze na tim pozicijama. Dobivamo novu, manju bazu karakterističnih riječi za klasu MED, koju označavamo sa B_2. Kada povežemo baze B_1 i B_2 dobivamo novu bazu vektorskog prostora koja se sastoji od 2000 elemenata, za razliku od 10635 elemenata u početnoj bazi.

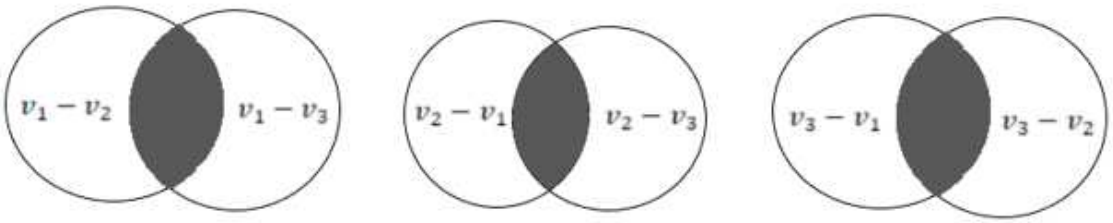
CRAN, MED i CISI klase

Opet ćemo provesti sličan postupak, ali ovoga puta koristeći 3 klase: CRAN, MED i CISI. Document-term matrica iz 2.2 sadrži 4891 redaka i 13112 stupaca. Prvo podijelimo skup dokumenata napola te time dobivamo 2445 dokumenata u trening skupu i 2446 dokumenata u test skupu. Razdvojimo sada dokumente iz trening skupa u tri matrice: u matrici M_1 i M_2 će se kao i u prethodnom slučaju nalaziti frekvencije pojavljivanja vektora baze

u klasama CRAN i MED, a u matrici M_3 u klasi CISI. Zbrajanjem svih redaka, ali svake matrice zasebno dobivamo vektore v_1, v_2 i v_3 u kojima se nalaze frekvencije pojavljivanja svake od 13112 riječi iz baze u određenoj klasi. Sada dolazimo do dijela postupka koji se razlikuje od slučaja sa dvije klase.

Sljedeće što želimo napraviti je pronaći najčešće riječi za svaku klasu u trening skupu. Situacija se ovdje malo komplicira budući da sada imamo 3 klase za usporedbu i želimo pronaći riječi u čijim se pojavljivanjima jedna klasa ističe u odnosu na druge dvije. Na primjer, ako tražimo reduciranu bazu riječi koje su najčešće u prvoj klasi, to znači da se te riječi češće pojavljuju u prvoj klasi nego u drugoj pa provodimo postupak kao u prvom dijelu. Isto tako, vrijedi da se te riječi češće pojavljuju u prvoj klasi nego u trećoj klasi pa i ovdje provedemo isti postupak. Zatim napravimo presjek ta dva skupa kako bi dobili bazu karakterističnih riječi za prvu klasu. Analogno napravimo i za druge dvije klase i dobijemo njihove reducirane baze.

Objasnimo detaljnije postupak za prvu klasu, a onda se isto primjenjuje i za druge



dvije klase. Slično kao u prvom dijelu, promatramo razlike u frekvencijama pojavljivanja pojmova u klasama. Gledamo vektor $v_1 - v_2$, pronalazimo 1000 najvećih elemenata i pamtimo koje riječi iz baze se nalaze na tim pozicijama. Tih 1000 riječi predstavljaju one riječi koje se više pojavljuju u klasi CRAN nego u klasi MED. Taj skup označavamo sa B_{11}. Slično, promatramo vektor $v_1 - v_3$, izvlačimo 1000 najčešćih riječi i skup nazivamo B_{12}. U B_{12} se nalaze riječi koje se više pojavljuju u klasi CRAN nego u CISI. Konačno, kako bi došli do reducirane baze B_1 za prvu klasu napravimo presjek ova dva dobivena skupa: $B_1 = B_{11} \cap B_{12}$. Istim postupkom dobivamo baze B_2 i B_3 za klase MED i CISI. Sa po 1000 elemenata iz baza B_1, B_2 i B_3 nova baza sastoji se od 3000 elemenata, što je veliko smanjenje u odnosu na 13112 elemenata u početnoj bazi.

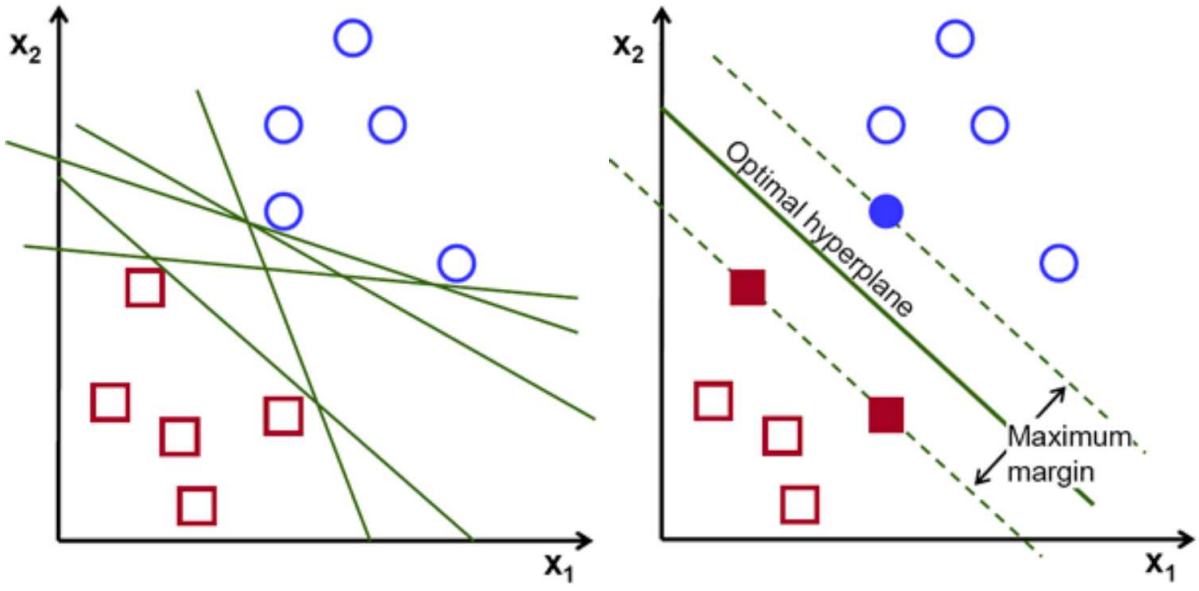
3.2 Stroj potpornih vektora

Stroj potpornih vektora (SVM) je algoritam nadziranog strojnog učenja koji se koristi za probleme klasifikacije i regresije. SVM algoritam na temelju trening skupa podataka generira model koji će za buduće primjere odrediti kojoj klasi pripadaju.

Neka je $X \subseteq \mathbb{R}^n$ prostor ulaznih podataka, a Y prostor rezultata. Skup za učenje (eng. *training set*) označavamo kao:

$$S = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq (X \times Y)^k$$

pri čemu je k broj primjera za učenje. Vektor x_i zovemo vektor značajki, a y_i je njemu pridružena oznaka. Ukoliko se radi o binarnoj klasifikaciji, prostor rezultata je $Y = \{-1, 1\}$, dok je za m-klasnu klasifikaciju $Y = \{1, 2, ..., m\}$. Za skup S kažemo da je trivijalan ako svi primjeri imaju istu pridruženu oznaku. Kažemo da su primjeri *linearno odvojivi* ukoliko postoji hiperravnina koja pravilno razdvaja primjere za učenje, inače kažemo da nisu odvojivi. Takva hiperravnina nije jedinstvena, stoga tražimo optimalnu hiperravninu, a kasnije ćemo pokazati da je to zapravo hiperravnina sa maksimalnom marginom, kao što je prikazano na slici 3.1.Pronalazak optimalne hiperravnine objasnit ćemo na primjeru binarne klasifikacije.



Slika 3.1: Primjer mogućih hiperravnina i optimalna hiperravnina, slika preuzeta sa [7]

Linearno odvojivi primjeri

Neka je funkcija $f : X \to Y$ zadana sa

$$f(x) = \langle w, x \rangle + b = \sum_{i=1}^{k} w_i x_i + b \tag{3.1}$$

funkcija koja klasificira primjere za učenje, određena parametrima $w \in \mathbb{R}^n$ i $b \in \mathbb{R}$. Svaka hiperravnina se može zapisati kao skup točaka koje zadovoljavaju: $\langle w, x \rangle + b = 0$, gdje je w normala hiperravnine, a b predstavlja pomak. Primjer x pridružit ćemo klasi C_1 ako je $f(x) \geq 0$, a klasi C_{-1} ako je $f(x) < 0$.

Kako bi se riješio problem pronalaska optimalne hiperravnine, uvest ćemo pojam *margine*. Margina se definira kao udaljenost primjera za učenje od hiperravnine. Margina skupa za učenje S se definira kao najmanja takva udaljenost. Općenito, što je veća margina, to je manja pogreška generalizacije klasifikacije te zbog toga tražimo hiperravninu koja ima najveću udaljenost od skupa za učenje i takvu udaljenost nazivamo *maksimalna margina*.

Funkcija koja klasificira primjere za učenje mora zadovoljavati:

$$\begin{cases} f(x_i) \geq 0, & \text{za sve } y_i = 1 \\ f(x_i) < 0, & \text{za sve } y_i = -1 \end{cases} \tag{3.2}$$

što možemo ekvivalentno zapisati kao

$$y_i f(x_i) \geq 0, \quad \forall\, i \in \{1, 2..., k\} \tag{3.3}$$

Udaljenost primjera za učenje x_i od hiperravnine računa se kao:

$$d_i = \frac{|f(x_i)|}{||w||} = \frac{|\langle w, x_i \rangle + b|}{||w||} = \frac{y_i(\langle w, x_i \rangle + b)}{||w||} \tag{3.4}$$

Budući da skaliranjem hiperravnine (w, b) skalarom $\lambda \in \mathbb{R}^+$ dobivamo hiperravninu $(\lambda w, \lambda b)$, možemo namjestiti w i b tako da za primjere koji se nalaze najbliže hiperravnini vrijedi:

$$y_i(\langle w, x_i \rangle + b) = 1 \tag{3.5}$$

Tada će općenito vrijediti:

$$y_i f(x_i) \geq 1, \quad \forall\, i \in \{1, 2..., k\} \tag{3.6}$$

Margina skupa za učenje je minimalna udaljenost d_i pa zbog 3.5 vrijedi:

$$min(d_i) = min\left(\frac{y_i(\langle w, x_i \rangle + b)}{||w||}\right) = \frac{1}{||w||} min(y_i(\langle w, x_i \rangle + b) = \frac{1}{||w||}. \tag{3.7}$$

S obzirom da nam je cilj maksimizirati udaljenost, potrebno je minimizirati $\|w\|$, a da pritom ne zaboravimo na ograničenja 3.6.

Ovaj optimizacijski problem možemo zapisati kao:

$$\begin{cases} \|w\| \rightarrow min_{w,b} \\ y_i(\langle w, x_i \rangle + b) \geq 1, \quad \forall\, i \in \{1, 2..., k\} \end{cases} \tag{3.8}$$

Problem optimizacije s ograničenjima rješava se uz pomoć Lagrangeovih multiplikatora. Time će problem 3.8 biti zapisan u obliku Lagrangeove funkcije:

$$L(w, b; \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{k} \alpha_i \cdot \Big[y_i(\langle w, x \rangle + b) - 1 \Big] \tag{3.9}$$

pri čemu su $\alpha_i \geq 0$ Lagrangeovi multiplikatori. Prvi dio predstavlja funkciju koju želimo minimizirati, a drugi dio uvjet koji mora biti zadovoljen. Problem smo sada sveli na traženje minimuma Lagrangeove funkcije i zovemo ga primarni problem. Deriviranjem Lagrangeove funkcije po parametrima w i b i izjednačavanjem parcijalnih derivacija s nulom, dobivamo nužan uvjet za ekstrem funkcije, u našem slučaju minimum:

$$\begin{gathered} \frac{\partial L(w, b; \alpha)}{\partial w} = w - \sum_{i=1}^{k} y_i \alpha_i x_i = 0 \Rightarrow w = \sum_{i=1}^{k} y_i \alpha_i x_i \\ \frac{\partial L(w, b; \alpha)}{\partial b} = \sum_{i=1}^{k} y_i \alpha_i = 0 \end{gathered} \tag{3.10}$$

Uvrstimo gornje uvjete u funkciju 3.9 i dobivamo dualni problem:

$$\begin{aligned} L(w, b; \alpha) &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{k} \alpha_i \cdot \Big[y_i(\langle w, x \rangle + b) - 1 \Big] \\ &= \frac{1}{2} \sum_{i,j=1}^{k} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i,j=1}^{k} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + \sum_{i=1}^{k} \alpha_i \\ &= \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{k} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \end{aligned} \tag{3.11}$$

uz ograničenja:

$$\begin{cases} \alpha_i \geq 0, \forall i = 1, 2, ..., k \\ \sum_{i=1}^{k} y_i \alpha_i = 0 \end{cases} \tag{3.12}$$

Dualni problem nam pojednostavljuje problem maksimizacije koji sada samo ovisi o izračunu skalarnog produkta. Za svaki primjer x_i postoji Lagrangeov multiplikator α_i, a one primjere

za koje vrijedi $\alpha_i > 0$ (što povlači 3.5) zovemo *potporni vektori*. To su primjeri koji se nalaze najbliže optimalnoj hiperravnini i najviše utječu na položaj i orijentaciju optimalne hiperravnine.

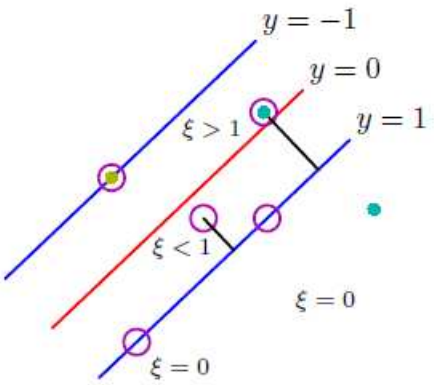
Meka margina

U slučaju kada su primjeri linearno odvojivi margina definirana u prethodnom dijelu zove se *tvrda margina*. No, primjeri nisu uvijek linearno odvojivi. Zato želimo dopustiti ulazak primjera u marginu i pogrešnu klasifikaciju pa uvodimo pojam *meka margina*.

Ograničenje 3.6 prelazi u:

$$y_i f(x_i) \geq 1 - \xi_i, \quad \xi_i \geq 0, \ \ \forall\, i \in \{1, 2..., k\} \tag{3.13}$$

gdje parametri ξ_i označavaju koliko je primjer ušao u marginu. Za primjere koji se nalaze na margini ili sa prave strane margine vrijedi $\xi_i = 0$.



Slika 3.2: SVM - meka margina, slika preuzeta sa [9]

Uvodimo parametar $C > 0$ koji će kontrolirati omjer penalizacije ulaza u marginu i problema minimizacije margine. Mali C dopušta krivo klasificiranje primjera, dok veliki C ne dopušta te se tada problem svodi na SVM model za linearno odvojive primjere. Želimo minimizirati:

$$C \sum_{i=1}^{k} \xi_i + \frac{1}{2}\|w\|^2$$

Primarna Lagrangeova funkcija iz 3.9 prelazi u:

$$L(w,b;\alpha,\beta,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{k}\xi_i - \sum_{i=1}^{k}\alpha_i \cdot \Big[y_i(\langle w,x\rangle + b) - 1 + \xi_i\Big] - \sum_{i=1}^{k}\beta_i\xi_i \tag{3.14}$$

gdje su $\alpha_i \geq 0$ i $\beta_i \geq 0$ Lagrangeovi multiplikatori. Kako bi dobili dualan problem, Lagrangeovu funkciju deriviramo po w, b i ξ, parcijalne derivacije izjednačimo s 0 i uvrstimo u 3.14. Tako dobivamo dualnu Lagrangeovu funkciju:

$$L(w,b;\alpha) = \sum_{i=1}^{k}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{k} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \tag{3.15}$$

uz ograničenja:

$$\begin{cases} 0 \leq \alpha_i \leq C, \forall i = 1,2,...,k \\ \sum_{i=1}^{k} y_i\alpha_i = 0 \end{cases} \tag{3.16}$$

Poglavlje 4

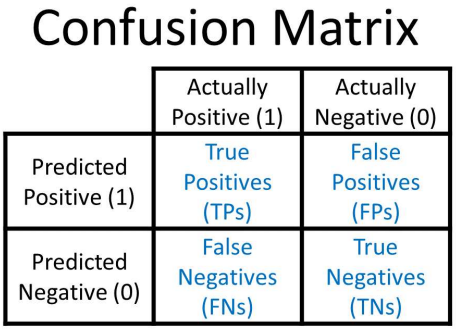
Rezultati

Usporedit ćemo rezultate klasifikacije normama (max, l_1 i l_2 norma) sa rezultatima klasifikacije dobivene pomoću algoritma stroja potpornih vektora. Također ćemo promotriti kakav utjecaj ima odabir težina koje pridajemo pojmovima na klasifikaciju. Rezultate ćemo prikazati prvo samo za dvije kolekcije, a zatim i za sve tri. Prije samih rezultata opisat ćemo koje mjere uspješnosti gledamo kako bi procijenili rezultate klasifikacije.

4.1 Mjere uspješnosti

Jedan od vizualnih alata koji opisuje izvedbu algoritma klasifikacije je matrica zabune (eng. *confusion matrix*). U njoj svaki redak predstavlja predviđenu klasu, a svaki stupac stvarnu klasu. Matrica zabune je uvijek simetrična tj. ima jednak broj stupaca i redaka i taj broj je jednak broju klasa. Matrica zabune prikazana je na slici 4.1. Ako pozitivnu klasu označimo s 1, a negativnu klasu s 0, oznake u matrici zabune su sljedeće: *true positives* (TP) su stvarno pozitivni, odnosno broj dokumenata iz klase 1 koji su klasificirani u klasu 1, *false positives* (FP) su lažno pozitivni, odnosno broj dokumenata iz klase 0 koji su kvalificirani u klasu 1, *true negatives* (TN) su stvarno negativni, odnosno broj dokumenata iz klase 0 koji su kvalificirani u klasu 0 i *false negatives* (FN) su lažno negativni, odnosno broj dokumenata iz klase 1 koji su kvalificirani u klasu 0.

Za procjenu uspješnosti klasifikacije, obično se koriste dvije mjere uspješnosti: odziv (eng. *recall*) i preciznost (eng. *precision*). Odziv računa omjer između dokumenata točno klasificiranih u klasu i ukupnog broja dokumenata koji su trebali biti klasificirani u tu klasu, dok preciznost mjeri omjer između dokumenata koji su točno klasificirani u neku klasu i ukupnog broja dokumenata klasificiranih u tu klasu. U principu, želimo da algoritam klasifikacije ima visok odziv kako bi točno klasificirao što više dokumenata i istovremeno visoku preciznost kako bi krivo klasificirao što manji broj dokumenata. Mjera uspješnosti koja spaja prve dvije navedene mjere naziva se točnost (eng. *accuracy*). U terminima



Slika 4.1: Matrica zabune

oznaka iz matrice zabune, ove tri mjere možemo definirati kao:

$$odziv = \frac{TP}{TP + FN}$$
$$preciznost = \frac{TP}{TP + FP}$$
$$točnost = \frac{TP + TN}{TP + TN + FP + FN}$$

Točnost se najjednostavnije definira kao postotak točno klasificiranih dokumenata, dakle omjer broja točno klasificiranih dokumenata i ukupnog broja dokumenata.

4.2 Klasifikacija 2 kolekcije

Prvo smo proveli klasifikaciju na 2 kolekcije: CRAN i MED. Podsjetimo se, ukupno imamo 2431 dokument, od kojih je 1215 u trening skupu, a 1216 u test skupu. Bez obzira što smo dosta strogo podijelili skup podataka, očekujemo da će klasifikacija dati dobre rezultate jer su korištene tematski dosta različite klase.

U poglavlju 3.1 pokazali smo postupak reduciranja baze sa 10635 elemenata na 2000, od čega 1000 elemenata predstavlja najčešće pojmove korištene iz dokumenata CRAN kolekcije, a drugih 1000 MED kolekcije. Sada dokumente iz test skupa prikažemo kao vektore u novoj, reduciranoj bazi te izračunamo norme posebno na prvih 1000 elemenata koji se odnose na klasu CRAN i na drugih 1000 elemenata koji se odnose na klasu MED te klasificiramo dokument tamo gdje je veća norma. Za klasifikaciju smo koristili 3 norme: max, l_1 i euklidsku normu. Također, primijenili smo razne načine prikaza frekvencija iz 2.3 kako bi vidjeli kako utječu na klasifikaciju te ističe li se koji od načina.

Na slici 4.2 vidimo tablicu sa rezultatima točnosti klasifikacije za sve tri norme i sve metode računanja frekvencija. Kao što smo i očekivali, rezultati su odlični jer svi pokazuju točnost preko 90%. Kod svih triju normi možemo uočiti da ne preferiramo obične frekvencije *txx* s obzirom da nam daju nešto slabije rezultate od ostalih frekvencija. Frekvencije *tfx* poznate pod nazivom TF-IDF se u sva tri slučaja pokazuju kao odlična opcija, iako ipak najveću točnost pokazuje *tfc* kod euklidske norme, jedini koji je dosegao čak 99%, točnije 99.01%.

	max-norma	**1-norma**	**2-norma**
txx	93.91%	95.64%	94.98%
txc	94.24%	96.05%	95.48%
tfx	97.53%	98.52%	98.52%
bfx	97.70%	98.44%	98.36%
tfc	90.30%	98.36%	**99.01%**

Slika 4.2: Točnost klasifikacije normama sa 2 kolekcije

Pogledajmo sada rezultate dobivene algoritmom stroja potpornih vektora, također za svih 5 prikaza frekvencija. Zanima nas hoće li SVM još malo poboljšati već ionako odlične rezultate. Na slici 4.3 možemo vidjeti tablicu sa rezultatima SVM-a koji su stvarno još bolji, sada je svugdje točnost iznad 96%. Zanimljivo je za primijetiti da sada *tfc* daje najlošije rezultate (iako i dalje vrlo dobre), dok obične frekvencije *txx* daju čak drugi najbolji rezultat. U svakom slučaju SVM je dao najbolji rezultat od svih, a to je točnost od 99.34% za metodu bilježenja frekvencija *txc*.

	SVM
txx	98.60%
txc	**99.34%**
tfx	96.63%
bfx	98.60%
tfc	97.86%

Slika 4.3: Točnost klasifikacije SVM-a sa 2 kolekcije

4.3 Klasifikacija 3 kolekcije

Iduće klasificiramo dokumente iz 3 kolekcije: CRAN, MED i CISI. Podsjetimo se, imamo ukupno 3891 dokument, od kojih je 1945 u trening skupu, a 1946 u test skupu. Reduciranje baze, računanje normi i razne načine zapisa frekvencija primjenjujemo jednako kao u klasifikaciji dvije kolekcije.

Zanima nas hoće li sada biti teže klasificirati s obzirom na jednu klasu više ili će algoritmi i dalje davati odlične rezultate. Na slici 4.4 možemo vidjeti tablicu sa rezultatima točnosti klasifikacije normama. Ovdje možemo uočiti da su se za max-normu rezultati pokazali lošiji, većina slučajeva klasificira ispod 80% točnosti. Što se tiče 1 i 2-norme, rezultati su prilično dobri, uglavnom malo lošiji u odnosu na klasifikaciju normama sa 2 kolekcije, ali ništa značajno. U globalu, najbolji rezultat klasifikacije normama ponovno daje *tfc* kod euklidske norme sa 98.41% točnosti, što je malo lošije od najboljeg rezultata za 2 kolekcije, ali i dalje vrlo dobro.

	max-norma	1-norma	2-norma
txx	83.09%	97.43%	96.71%
txc	85.92%	96.56%	96.81%
tfx	93.27%	98.25%	98.25%
bfx	87.87%	97.48%	97.89%
tfc	93.63%	98.10%	**98.41%**

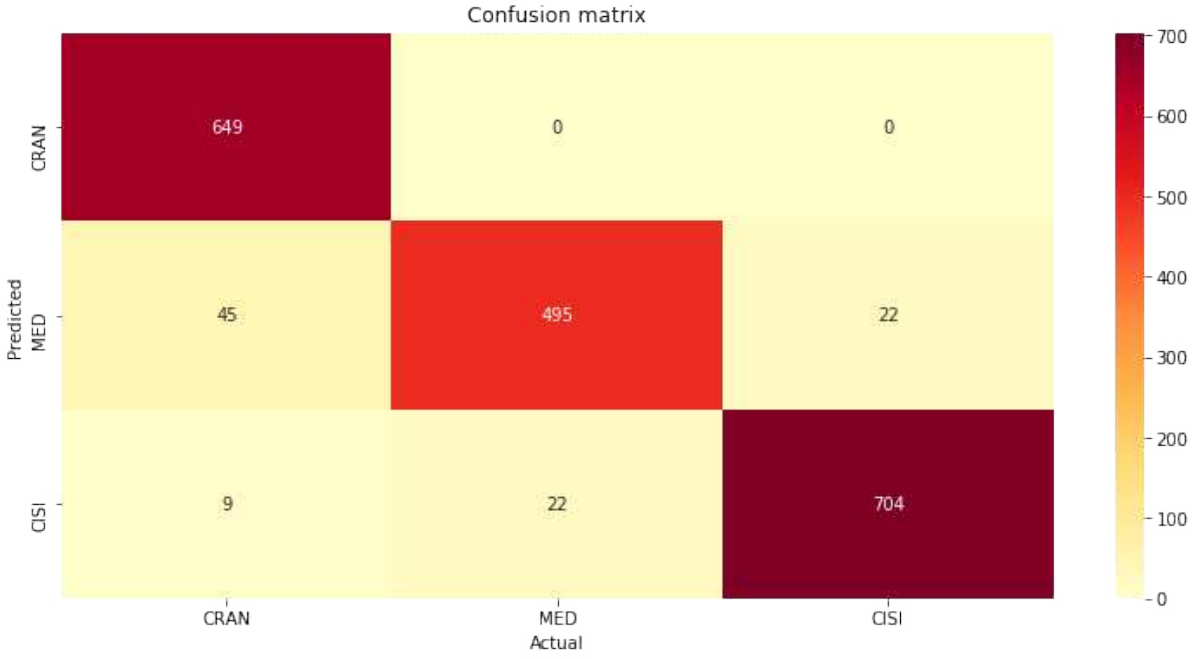
Slika 4.4: Točnost klasifikacije normama sa 3 kolekcije

Pogledajmo sada rezultate algoritma potpornih vektora za 3 kolekcije. Na slici 4.5 možemo vidjeti tablicu sa rezultatima za svih 5 načina zapisa frekvencija, koji i nisu nešto puno bolji od rezultata sa 1 i 2 normom, ali su to i dalje odlični rezultati. Uočavamo da je i dalje najbolji rezultat preko 99% za *txc*, točnije 99.02%.

Primijetimo da se u tablici uz mjeru točnosti pojavljuju i mjere odziva i preciznosti. U podcjelini 4.1 opisali smo što te mjere predstavljaju te kako se računaju uz pomoć oznaka iz matrice zabune. Pokažimo taj postupak i objašnjenje na primjeru rezultata iz zadnje tablice i pomoću matrice zabune sa slike 4.6. U slučaju klase CRAN preciznost iznosi 1.00 (za sve načine prikaza frekvencija) što znači da nijedan primjer iz preostalih klasa nije klasificiran u klasu CRAN. U matrici zabune vidimo da je to stvarno istina jer je broj lažno pozitivnih za klasu CRAN 0, odnosno algoritam nije ni u jednom slučaju dokumente iz klasa MED i CISI klasificirao u klasu CRAN. S druge strane preciznost za klasu MED kod metode *tfx*

	precision			recall			accuracy
	CRAN	MED	CISI	CRAN	MED	CISI	
txx	1	0.95	0.97	0.97	0.97	0.98	97.53%
txc	1	0.99	0.98	0.99	0.99	0.99	**99.02%**
tfx	1	0.88	0.96	0.92	0.96	0.97	94.96%
bfx	1	0.91	0.98	0.96	0.98	0.96	96.61%
tfc	1	0.91	0.98	0.95	0.98	0.96	96.61%

Slika 4.5: Točnost klasifikacije SVM-a sa 3 kolekcije



Slika 4.6: Matrica zabune za slučaj *tfx*

iznosi 0.88, što možemo vidjeti i u matrici zabune jer je algoritam 45 dokumenata iz klase CRAN i 22 dokumenta iz klase CISI pogrešno klasificirao u klasu MED pa preciznost 0.88 dobivamo računom:

$$\frac{TP}{TP + FP} = \frac{495}{495 + 45 + 22} = 0.88$$

Mjeru odziva možemo objasniti na primjeru za klasu CISI kod metode *tfx* gdje ona iznosi 0.97. To znači da je 97% dokumenata iz CISI klase točno klasificirano, točnije 704 dokumenta od ukupnih 704+22=726 dokumenata je točno klasificirano u klasu CISI.

4.4 Usporedba rezultata

Cilj nam je usporediti dvije metode linearne klasifikacije, klasifikaciju normama i algoritmom stroja potpornih vektora. Na slici 4.7 vidimo tablicu sa najboljim rezultatima iz klasifikacije normama (u oba slučaja je to euklidska norma) i rezultatima SVM-a, prvo za klasifikaciju provedenu na dvjema kolekcijama, a zatim na trima. Možemo uočiti da algoritam stroja potpornih vektora u oba slučaja ostvaruje bolje rezultate, ali neznatno. Također primijećujemo da su rezultati klasifikacije bolji kada se klasifikacija provodi na dvije kolekcije, nego na tri. To je i za očekivati s obzirom da se početne dvije kolekcije dosta tematski razlikuju, ali što se više kolekcija dodaje to se otežava klasifikacija i teže je sa stopostotnom preciznošću prepoznati iz koje klase dokument dolazi. No, na kraju možemo reći da je klasifikacija uspjela te da obje metode daju zadovoljavajuće rezultate. Klasifikacija normama ima prednost što reducira bazu jer je uvijek korisno smanjiti dimenzionalnost prostora, dok je algoritam stroja potpornih vektora već gotova metoda i jednostavnije je i brže provesti klasifikaciju s obzirom da vrlo dobro barata sa visokodimenzionalnim prostorima.

	2 kolekcije	3 kolekcije
norme	99.01%	98.41%
SVM	99.34%	99.02%

Slika 4.7: Usporedba rezultata

Bibliografija

[1] D. Bakić, *Linearna algebra*, Sveučilište u Zagrebu, Matematički odjel PMF-a, 2008.

[2] M. Pezić, *Tehnike učenja za klasifikaciju bioloških nizova*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odsjek, 2020.

[3] S. Mavrek, *Iterativno traženje fraza i statistika semantičkog indeksiranja*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odsjek, 2019.

[4] Glasgow IDOM, dostupno na `http://ir.dcs.gla.ac.uk/resources/test_collections/`

[5] Snowball, dostupno na `http://snowball.tartarus.org/algorithms/porter/stemmer.html`

[6] G. Salton, C. Buckley *Term-weighting approaches in automatic text retrieval*, In Information Processing and Management, Volume 24, Issue 5, 1988.

[7] R. Gandhi, *Support Vector Machine - Introduction to Machine Learning Algorithms*, dostupno na `https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444`

[8] S. Bhattacharyya, *Support Vector Machine: Complete theory*, dostupno na `https://towardsdatascience.com/understanding-support-vector-machine-part-1-lagrange-multipliers-5c24a52ff`

[9] C.M. Bishop, *Patter Recognition and Machine Learning*, Springer, New York, 2006.

[10] C.J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, 1998. dostupno na `https://www.di.ens.fr/~mallat/papiers/svmtutorial.pdf`

Sažetak

Cilj rada je analiza linearne klasifikacije tekstualnih dokumenata pomoću normi i algoritma stroja potpornih vektora. Na početku rada definirani su osnovni pojmovi iz linearne algebre i optimizacije važni za daljnje razumijevanje rada. Pojašnjen je postupak sređivanja tekstualnih dokumenata, između ostalog stemming algoritam, kako bi dokumente prikazali u obliku pogodnom za analizu. Opisani su razni načini zapisivanja frekvencija pojavljivanja pojmova u tekstu, kako bi proučili kako utječu na rezultate klasifikacije. Za analizu smo koristili dvije odnosno tri kolekcije tekstualnih dokumenata, a skup podataka smo podijelili na trening i test skup u omjeru 50:50. Opisali smo reduciranje baze i algoritam stroja potpornih vektora. Na kraju prikazujemo rezultate klasifikacije dvije odnosno tri kolekcije te uspoređujemo dobivene rezultate. Pokazalo se da su rezultati klasifikacije za sve metode slični i iznimno dobri.

Summary

The aim of this work is analysis of text document classification using norm and support vector machine algorithm. Initially, we present basic concepts from linear algebra and optimization. We explain text preparation and stemming algorithm, used to convert documents in a suitable form. Different term-weighting approaches are described to see how they affect classification. We explain process of dimension reduction and mathematical background of the support vector machine algorithm. At the end, we present and compare classification results for two and three collections. It is shown that results are similar for all methods and classification is very accurate.

Životopis

Rođena sam 17.08.1994. godine u Zagrebu. Pohađala sam Osnovnu školu Julija Klovića u Zagrebu. Obrazovanje nastavljam 2009. godine upisom u prirodoslovno-matematičku V. gimnaziju. Godine 2013. završavam srednjoškolsko obrazovanje i upisujem preddiplomski studij Matematika na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu, kojeg završavam 2017. godine. Po završetku preddiplomskog studija upisujem diplomski studij Matematička statistika na istom fakultetu, kojeg upravo završavam. Trenutno sam zaposlena u Privatnoj gimnaziji i ekonomsko-informatičkoj školi Futura kao nastavnica matematike.