Semantičko indeksiranje i klasifikacija dokumenata

Janjić, Filip

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Filip Janjić

SEMANTIČKO INDEKSIRANJE I KLASIFIKACIJA DOKUMENATA

Diplomski rad

Voditelj rada:
doc.dr.sc. Pavle Goldstein

Zagreb, veljača, 2019.

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

Konstrukcija strojeva sposobnih za učenje iz danih primjera bila je dugo vremena tema filozofskih i tehničkih rasprava. Međutim, tehnološki napredak je pokazao da strojevi imaju značajnu razinu upravo takvih sposobnosti. Postoje mnoge vrste raznih algoritama za učenje a jedan od njih je i stroj s potpornim vektorima (eng. *Support Vector Machine*), koji se ponajviše koristi za probleme klasifikacije.

Veoma ubrzanim rastom broja dostupnih podataka i informacija, klasifikacija teksta je postala jedna od ključnih tehnika za organizaciju i upravljanje tekstualnim podacima. Pri tome je glavni zadatak upravo klasificiranje tekstualnih dokumenata u unaprijed definirani broj kategorija temeljen na zajedničkom kontekstu ili drugim atributima. Problem klasifikacije javlja se u raznim područjima poput filtriranja elektronske pošte, internet pretraživanjima ili razvrstavanju dokumenata po određenoj temi. Zanimljiv pristup klasifikaciji teksta je pristup temeljen na n-gramima, koji nije osjetljiv na gramatiku i pravopis. Strukturu pogodnu za takav pristup imaju biološki nizovi kao što su DNA i proteini.

Proteini su nizovi aminokiselina, zadani u odgovarajućem alfabetu koji koristi 20 engleskih slova. S obzirom na to da postoji jako veliki broj različitih proteina, jedan od ključnih problema u bioinformatici je uočiti strukturne i funkcionalne karakteristike proteina temeljene na nizovima aminokiselina. Kako se proteini grupiraju u proteinske familije, jedan od načina za to je povezivanje novih proteinskih nizova sa proteinima čija su svojstva već poznata. Upravo tim problemom ćemo se baviti, odnosno, pokušat ćemo klasificirati nove proteine u specifičnu proteinsku familiju kojima pripadaju. Pri tome ćemo se služiti strojem s potpornim vektorima te n-gramskim rječnikom.

U ovom radu je sadržano 5 poglavlja. U prvom poglavlju ćemo spomenuti neke pojmove iz vektorskih i euklidskih prostora te optimizacije, koje ćemo koristiti u nastavku rada. Drugo poglavlje će nam pružiti uvid u matematičku pozadinu stroja s potpornim vektorima, gdje će poseban značaj imati jezgrene funkcije, čiji je odabir ključan za uspješnu klasifikaciju dokumenata. Kroz treće poglavlje ćemo detaljno opisati skup podataka sa kojim radimo, vidjeti kako taj skup prilagoditi algoritmu te detaljno opisati način na koji provodimo testove. Četvrto i peto poglavlje će sadržavati rezultate različitih testova s obzirom na jezgrene funkcije te ćemo pokušati iznijeti zaključak koji se odnosi na naš slučaj.

Poglavlje 1

Pojmovi

1.1 Vektorski prostor

Definicija 1.1.1. *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) postoji $0 \in V$ *sa svojstvom* $a + 0 = 0 + a = a$, $\forall a \in V$,
(3) $\forall a \in V$ *postoji* $-a \in V$ *tako da je* $a + (-a) = -a + a = 0$,
(4) $a + b = b + a$, *za svaki* $a, b \in V$,
(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) $1 \cdot a = a$, $\forall a \in V$.

Elementi vektorskog prostora se nazivaju *vektorima*, koje ćemo označavati malim latinskim slovima. Elemente polja $\mathbb{F}$ označavat ćemo malim grčkim slovima i zvati *skalarima*.

Definicija 1.1.2. *Neka je V vektorski prostor nad $\mathbb{F}$. Izraz oblika $\alpha_1 a_1 + ... + \alpha_k a_k$, pri čemu su $a_1, ..., a_k \in V$, $\alpha_1, ..., \alpha_k \in \mathbb{F}$ i $k \in \mathbb{N}$, naziva se linearna kombinacija vektora $a_1, ..., a_k$ s koeficijentima $\alpha_1, ..., \alpha_k$.*

Također, neka je V vektorski prostor nad $\mathbb{F}$ i $S = \{a_1, ..., a_k\}$, $k \in \mathbb{N}$, konačan skup vektora iz V. Kažemo da je skup S linearno nezavisan ako vrijedi

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

U suprotnom kažemo da je skup S linearno zavisan.

Za prirodne brojeve m i n, preslikavanje $A : \{1, ...m\} \times \{1, ...n\} \to \mathbb{F}$ naziva se matrica tipa (m, n) s koeficijentima iz polja $\mathbb{F}$. Djelovanje svake takve funkcije A pišemo tablično, u m redaka i n stupaca, pišući u i-ti redak i j-ti stupac funkciju vrijednost $A(i, j)$ koju jednostavno označavamo sa a_{ij}. Također, matricu A sa elementima a_{ij} ćemo označavati sa $A = [a_{ij}]$. Skup svih matrica s m redaka i n stupaca s koeficijentima iz polja $\mathbb{F}$ označavamo s $M_{mn}(\mathbb{F})$. Ako je $m = n$ pišemo kraće $M_n(\mathbb{F})$, a elemente tog skupa zovemo kvadratnim matricama reda n. Ako nam nije važno radimo li sa realnim ili kompleksnim matricama, pisat ćemo kratko M_n. Dodatno, ako je $\mathbb{F} = \mathbb{R}$, *transponirana matrica* A^T matrice $A = [a_{ij}]$ definirana je sa $A^T = [a_{ji}]$. Također, Neka je A kvadratna matrica. Kažemo da je A *simetrična* ako je $A^T = A$.

Definicija 1.1.3. *Neka je V vektorski prostor nad poljem $\mathbb{F}$. Skalarni produkt na V je preslikavanje $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$ koje ima sljedeća svojstva:*
(1) $\langle x, x \rangle \geq 0, \quad \forall x \in V,$
(2) $\langle x, x \rangle = 0 \iff x = 0,$
(3) $\langle x_1 + x_2, y \rangle = \langle x_1, y \rangle + \langle x_2, y \rangle, \quad \forall x_1, x_2, y \in V,$
(4) $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle, \quad \forall \alpha \in \mathbb{F}, \forall x, y \in V,$
(5) $\langle x, y \rangle = \overline{\langle y, x \rangle}, \quad \forall x, y \in V.$

Uočimo da skalarni produkt poprima vrijednosti u polju nad kojim je dani vektorski prostor izgrađen, odnosno, ako je prostor kompleksan, iz zadnjeg svojstva vidimo da su skalarni umnošci $\langle x, y \rangle = \langle y, x \rangle$ međusobno konjugirani kompleksni brojevi. Ako je prostor realan, skalarni umnožak bilo koja dva vektora je realan broj pa kompleksno konjugiranje nema efekta. Stoga se u realnim prostorima ovo svojstvo naziva simetričnost, a u kompleksnim prostorima hermitska simetričnost. Nas će u ovom radu zanimati realan prostor.

Definicija 1.1.4. *Vektorski prostor na kojem je definiran skalarni produkt zove se unitaran prostor.*

U $\mathbb{R}^n$ skalarni produkt je definiran s $\langle (x_1, ..., x_n), (y_1, ..., y_n) \rangle = \sum_{i=1}^n x_i y_i$, dok je u $\mathbb{C}^n$ skalarni produkt definiran sa $\langle (x_1, ..., x_n), (y_1, ..., y_n) = \sum_{i=1}^n x_i \overline{y_i}$. Također, neka je V unitaran prostor. Kaže se da su vektori x, y iz V međusobno okomiti ili ortorognalni ako je $\langle x, y \rangle = 0$.

Definicija 1.1.5. *Gramova matrica skupa vektora $X = \{x_1, ..., x_n\}$ definirana je sa*

$$G = \begin{bmatrix} \langle x_1, x_1 \rangle & \dots & \dots & \langle x_1, x_n \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle x_n, x_1 \rangle & \dots & \dots & \langle x_n, x_n \rangle \end{bmatrix} = [\langle x_i, x_j \rangle]_{i,j=1}^n.$$

Teorem 1.1.6. *(Cauchy - Schwarz - Bunjakovski nejednakost) Neka je U unitaran prostor uz skalarni produkt $\langle \cdot, \cdot \rangle$. Tada je*

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle$$

za sve x, y iz U. Jednakost vrijedi ako i samo ako su vektori x i y linearno zavisni.

CSB nejednakost je korisna tehnička pomoć pri računanju u unutarnim prostorima te nas navodi na ideju da, poopćujući ovu formulu, i u apstraktnom vektorskom prostoru uvedemo koncept modula, tj. "duljine" vektora.

Definicija 1.1.7. *Neka je V unitaran prostor. Norma na V je funkcija $\| \cdot \| : V \to \mathbb{R}$ definirana s $\|x\| = \sqrt{\langle x, x \rangle}$.*

Propozicija 1.1.8. *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) $\|x + y\| \leq \|x\| + \|y\|, \ \forall x, y \in V.$

Dodatno, neka je V unitaran prostor. Kažemo da je vektor $x \in V$ *normiran* ako je $\|x\| = 1$. Također, neka je dan vektor $x \in V$ nad poljem $\mathbb{R}^n$ i neka je x_i i-ta komponenta vektora x, $i = 1, ..., n$. Tada je *euklidska* ili *2-norma* dana sa $\|x\|_2 = (\sum_{i=1}^{n} |x_i|^2)^{\frac{1}{2}}$.

Definicija 1.1.9. *Kvadratna simetrična matrica A dimenzije $n \times n$ je pozitivno definitna ako je A realna i $x^T A x > 0$ za sve $x \in \mathbb{R}^n, x \neq 0$. Kažemo da je A pozitivno semidefinitna ako je realna i $x^T A x \geq 0$ za sve $x \in \mathbb{R}^n$.*

Primijetimo da se Gramova matrica može napisati kao $G = X^T X$, pri čemu je $X = (x_1, ..., x_n)$. Lako se vidi da je svaka Gramova matrica simetrična, pozitivno semidefinitna matrica. Naime, simetričnost slijedi direktno iz simetričnosti skalarnog produkta, a ako uzmemo proizvoljan $y \in \mathbb{R}^n$, imamo $y^T G y = y^T X^T X y = (Xy)^T (Xy) = \|Xy\|^2 \geq 0$, odakle slijedi i pozitivna semidefinitnost.

1.2 Euklidski prostor

Definicija 1.2.1. *Neka je $\mathcal{A}$ neprazan skup, V vektorski prostor na poljem $\mathbb{F}$, te neka je definirano preslikavanje $v : \mathcal{A} \times \mathcal{A} \to V$. Uređena trojka $(\mathcal{A}, V, v)$ naziva se afinim prostorom nad V, a elementi skupa $\mathcal{A}$ točkama tog prostora, ako vrijedi:*
(1) Za svaki $A \in \mathcal{A}$ i svaki $x \in V$ postoji jednoznačno određena točka $B \in \mathcal{A}$ tako da je $v(A, B) = x$,
(2) Za sve $A, B, C \in \mathcal{A}$ vrijedi $v(A, B) + v(B, C) = v(A, C)$.

U n-dimenzionalnom afinom prostoru $(n-1)$-ravnina naziva se *hiperravninom*, a 1-ravnina se naziva *pravcem*. Također, svaka hiperravnina u n-dimenzionalnom afinom prostoru $\mathcal{A}^n$ se može analitički predstaviti jednom linearnom jednadžbom oblika

$$\alpha_1 x_1 + ... + \alpha_n x_n + \alpha_0 = 0,$$

pri čemu nisu svi α_i jednaki nuli. Takva jednadžba je *opći oblik jednadžbe* hiperravnine.

Definicija 1.2.2. *Euklidski prostor je uređena trojka $(\mathcal{E}^n, U^n, v)$ sastavljena od skupa točaka $\mathcal{E}^n$, realnog unitarnog prostora U^n i preslikavanja $v : \mathcal{E}^n \times \mathcal{E}^n \to U^n$ koje ima svojstva:*
(1) Za svaku točku $A \in \mathcal{E}^n$ i vektor $x \in U^n$ postoji jedna i samo jedna točka $B \in \mathcal{E}^n$ takva da je $v(A, B) = \overrightarrow{AB} = x$,
(2) Za sve $A, B, C \in \mathcal{E}^n$ vrijedi $v(A, B) + v(B, C) = v(A, C)$.

Da bismo uveli pojam udaljenosti neke točke od ravnine, što će nam biti važno u nastavku, moramo prvo vidjeti kako je definirana ortogonalna projekcija točke na ravninu.

Definicija 1.2.3. *Neka je $W^k \subseteq U^n$. Ortogonalni komplement potprostora W^k je skup $\{x \in U^n \mid \langle x, w\rangle = 0,\ \forall w \in W^k\}$, a označavamo ga sa $(W^k)^\perp$.*

Ortogonalni komplement hiperravnine π^{n-1} točkom A je upravo okomica iz te točke na hiperravninu. Štoviše, *ortogonalna projekcija* točke A na bilo koju ravninu π je nožište okomice iz A na π koju označavamo sa A_π.

Definicija 1.2.4. *Preslikavanje $p_\pi : \mathcal{E}^n \to \pi$ koje točki $A \in \mathcal{E}^n$ pridružuje točku A_π naziva se ortogonalno projiciranje točke A na ravninu π.*

Neka je $A(x_1^0, ...x_n^0)$ točka prostora $\mathcal{E}^n$ a π hiperravnina određena jednađžbom $\alpha_1 x_1 + ...\alpha_n x_n + \alpha_0 = 0$, pri čemu je $x = (x_1, ..., x_n)$, a $n = (\alpha_1, ...\alpha_n)$ normala te hipperravnine. Tada se hiperravnina može vektorski zapisati kao

$$\langle x, n\rangle + \alpha_0 = 0.$$

Definicija 1.2.5. *Udaljenost $d(A, A_\pi)$ točke A od njezine ortogonalne projekcije na hiperravninu π, naziva se udaljenost točke A od hiperravnine π i označava sa $d(A, \pi)$.*

Vektor n je okomit na hiperravninu π, pa jednadžba pravca $\pi_A^\perp$ glasi $r = r_A + \tau n$, gdje je r_A radij vektor točke A, a vrijednost parametra τ za točku A_π jednaka $-\frac{\langle n, r_A\rangle + \alpha_0}{\|n\|^2}$. Imamo

$$r_{A_\pi} = r_A - \frac{\langle n, r_A\rangle + \alpha_0}{\|n\|^2} n.$$

Udaljenost točke A od hiperravnine π dana je izrazom

$$d(A, \pi) = \|r_{A_\pi} - r_A\| = \frac{|\langle n, r_A\rangle + \alpha_0|}{\|n\|}.$$

1.3 Optimizacija

Problem koji se sastoji od određivanja ekstrema neke funkcije, uz zadane uvjete, naziva se problemom matematičkog programiranja. Specijalno, ako je funkcija cilja za koju treba odrediti maksimum ili minimum linearna, te ako su uvjeti izraženi u obliku linearnih jednadžbi i/ili nejednadžbi, onda je to problem linearnog programiranja. Slično tome, ako je funkcija cilja kvadratna, imamo problem kvadratnog programiranja.

Definicija 1.3.1. *Neka je $\Omega \subseteq \mathbb{R}^n$ otvoren skup. Kažemo da funkcija $f : \Omega \to \mathbb{R}^n$ ima lokalni minimum u točki $P_0 \in \Omega$ ako postoji okolina $K(P_0, r) \subseteq \Omega$ takva da*

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

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

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

Vrijednosti $f(P_0)$ nazivamo minimumom, odnosno maksimum funkcije f na skupu Ω.

Ako u prethodnoj definiciji vrijede stroge nejednakosti, radi se o strogom lokalnom minimumu, odnosno o strogom lokalnom maksimumu. Također, ako vrijede nejednakosti za svaku točku $P \in \Omega$, tada funkcija f u točki P_0 ima globalni minimum, odnosno maksimum.

Definicija 1.3.2. *Neka je $\Omega \subseteq \mathbb{R}^n$ otvoren skup i neka je $f : \Omega \to \mathbb{R}$ diferencijabilna funkcija. Za točku $P_0 \in \Omega$ kažemo da je stacionarna točka funkcije ako vrijedi:*

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

Teorem 1.3.3. *(Nužan uvjet za postojanje lokalnog ekstrema) Ako je $P_0 \in \Omega \subseteq \mathbb{R}^n$ točka lokalnog ekstrema diferencijaliblne funkcije $f : \Omega \to \mathbb{R}$, onda je P_0 stacionarna točka fukcije f, tj. vrijedi:*

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

Pretpostavimo da su zadane funkcije $f, g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ...m$, te promatrajmo sljedeći optimizacijski problem

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

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

Pri tome skup $U = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, \; i = 1, ...m\}$ zovemo *dopustivo područje*, a svaki $x \in U$ zovemo *dopustivo rješenje*. Dopustivo rješenje x^* sa svojstvom $f(x^*) \leq f(x)$ 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 1.3.4. *Problem*

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$g_i(x) \leq 0, \;\; i = 1, ...m.$$

ekvivalentan je problemu

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

Dokaz. Označimo sa $g(x) := (g_1(x), ..., g_m(x))$. Uočimo da za fiksni $x \in \mathbb{R}^n$ vrijedi:

$$\max_{\alpha \in \mathbb{R}^m_+} L(x, \alpha) = \begin{cases} \text{f(x)}, & \text{g(x)} \leq 0 \\ \infty, & \textit{inače}. \end{cases}$$

Naime, za $g(x) \leq 0$ maksimum funkcije L po varijabli $\alpha \geq 0$ se postiže za $\alpha = 0$. S druge strane, ako je $g_i(x) > 0$ za neki $i \in \{1, ..., m\}$ povećanjem vrijednosti komponenata vektora $\alpha \in \mathbb{R}^m_+$ funkciju $L(x, \alpha)$ možemo proizvoljno povećati. Minimizacijom po $x \in \mathbb{R}^n$ vidimo da se minimum od $max\, L(x, \alpha)$ postiže za $g(x) \leq 0$ i da je on jednak minimumu funkcije f na dopustivom skupu $U = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$, te su prema tome navedeni problemi uistinu ekvivalentni. □

Problem iz teorema zovemo *primarni problem*, a budući da je rješenje primarnog problema ujedno rješenje originalnog optimizacijskog problema, njega također zovemo *primarni problem*. Nadalje, možemo promatrati sljedeći optimizacijski problem

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

kojeg zovemo *dualni problem*. Nadalje, pretpostavimo da je zadan problem linearnog programiranja u sljedećem obliku

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

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

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

Odgovarajući primarni i dualni problem tada glase

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

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

Poglavlje 2

Stroj s potpornim vektorima

2.1 Linearna klasifikacija

Binarna klasifikacija najčešće se izvodi pomoću realne funkcije realne varijable $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ na idući način: vektor $x = (x_1, ..., x_n)^T$ je dodijeljen pozitivnoj klasi ako je $f(x) \geq 0$, odnosno negativnoj klasi ako je $f(x) < 0$. Promatramo slučaj gdje je $f(x)$ linearna funkcija za $x \in X$ te je možemo zapisati kao:

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

pri čemu su $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ parametri koji određuju funkciju. Geometrijska interpretacija ovakve funkcije je ta da je prostor X podijeljen na dva poluprostora hiperravninom koja je određena jednadžbom $\langle w, x \rangle + b = 0$. Pri tome je vektor w normala te hiperravnine a varijacijom s parametrom b je paralelno pomičemo. Parametre w i b ćemo nazivati *vektor težine* (eng. *weight vector*) i *pomak* (eng. *bias*).

Definicija 2.1.1. *Neka je $X \subseteq \mathbb{R}^n$ prostor ulaznih podataka te Y prostor rezultata. Skup za učenje (eng. training set) je kolekcija primjera za učenje (eng. traning examples ili training data) koju označavamo kao*

$$S = ((x_1, y_1), ..., (x_l, y_l)) \subseteq (X \times Y)^l$$

pri čemu je l broj primjera za učenje. Nadalje, x_i zovemo vektor značajki, a y_i njemu pridruženu oznaku (eng. label).

Uočimo, za binarnu klasifikaciju je $Y = \{-1, 1\}$, dok je za m-klasnu klasifikaciju $Y = \{1, 2, ..., m\}$. Također, kažemo da je skup za učenje S trivijalan ako svi primjeri imaju iste pridružene oznake. Ukoliko postoji hiperravnina koja pravilno klasificira primjere za učenje, reći ćemo da su podaci *linearno odvojivi*. U suprotnom kažemo da nisu odvojivi.

2.2 Margina linearnog klasifikatora

Stroj s potpornim vektorima je skup metoda nadziranog učenja (eng. *supervised learning*) koje se koriste za klasifikaciju podataka, pri čemu je nadzirano učenje tehnika strojnog učenja kojom izračunavamo funkciju iz skupa podataka za treniranje. Za dani skup podataka za treniranje SVM algoritam za treniranje generira model na temelju kojega se određuje kojoj kategoriji pripada testni podatak.

Klasifikator maksimalnih margina

Stroj s potpornim vektorima se temlji na klasifikatoru maksimalnih margina (eng. *the maximal margin classifier*) koji se koristi za one podatke koji su linearno odvojivi. Kada gledamo hiperravninu koja dijeli grupe u odnosu na skup za učenje, klasifikator maksimalnih margina optimizira, odnosno, maksimizira udaljenost primjera za učenje određene grupe od hiperravnine koja ih dijeli.

Definicija 2.2.1. *Funkcijska margina hiperravnine (w,b) s obzirom na primjer za učenje* (x_i, y_i) *dana je sa*

$$\gamma_i = y_i(\langle w, x_i \rangle + b).$$

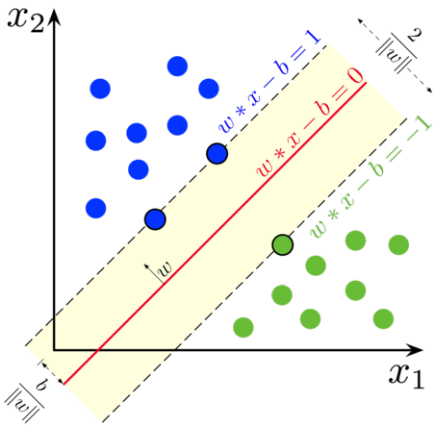
Uočimo da $\gamma_i > 0$ implicira ispravnu klasifikaciju primjera (x_i, y_i).

Definicija 2.2.2. *Funkcijska margina hiperravnine (w,b) s obzirom na skup za učenje S dana je sa* $\gamma = \min_i \gamma_i$.

Poželjno je imati što veću funkcijsku marginu. Uočimo da skaliranjem hiperravnine (w, b) za željeni skalar $\lambda \in \mathbb{R}^+$ dobivamo hiperravninu $(\lambda w, \lambda b)$. Ukoliko želimo izmjeriti euklidsku udaljenost primjera za učenje od linearnog klasifikatora u skupu za učenje, koristiti ćemo *geometrijsku marginu*, koju dobivamo kao funkcijsku marginu normirane hiperravnine, odnosno, kao funkcijsku marginu hiperravnine koju određuju parametri $(\frac{1}{\|w\|}w, \frac{1}{\|w\|}b)$. Konačno, *margina skupa za učenje S* je maksimalna geometrijska margina svih mogućih razdvajajućih hiperravnina. Nadalje, geometrijsku marginu možemo optimizirati tako da namjestimo funkcionalnu marginu da bude jednaka $\gamma = 1$ i minimiziramo normu vektora težine w. Ovaj optimizacijski problem možemo matematički zapisati kao:

$$\begin{cases} \langle w, w \rangle \rightarrow min_{w,b} \\ \mathrm{y}_i(\langle w, x_i \rangle + b) \geq 1, \\ \mathrm{i} = 1,...,\mathrm{l} \end{cases}$$

pri čemu tu zadaću nazivamo primarnom formom.



Slika 2.1: Funkcijska margina; $\gamma = 1$, slika preuzeta iz [8]

Ukoliko je w vektor težine takav da imamo funkcijsku marginu vrijednosti $\gamma = 1$ za pozitivan primjer x^+ i negativan primjer x^-, takva funkcijska margina implicira

$$\langle w, x^+ \rangle + b = +1$$

$$\langle w, x^- \rangle + b = -1$$

dok za geometrijsku marginu moramo normirati vektor w. Geometrijska margina γ je tada:

$$\gamma = \frac{1}{2}\left(\left\langle \frac{w}{\|w\|_2}, x^+ \right\rangle - \left\langle \frac{w}{\|w\|_2}, x^- \right\rangle\right) = \frac{1}{2\|w\|_2}\left(\left\langle w, x^+ \right\rangle - \left\langle w, x^- \right\rangle\right) = \frac{1}{\|w\|_2}.$$

Time smo pokazali sljedeću propoziciju.

Propozicija 2.2.3. *Neka je dan linearno odvojiv skup podataka za trening*

$$S = ((x_1, y_1), ..., (x_l, y_l)).$$

Hiperravnina (w, b) *koja rješava optimizacijski problem*

$$\begin{cases} \langle w, w \rangle \rightarrow min_{w,b} \\ y_i(\langle w, x_i \rangle + b) \geq 1, \\ i = 1,...,l \end{cases}$$

ima maksimalnu marginu hiperravnine kao geometrijsku marginu $\gamma = \frac{1}{\|w\|_2}$.

Podaci za trening x_i za koje je funkcijska margina $\gamma = 1$ su upravo oni koji su najbliže hiperravnini te ih iz tog razloga nazivamo *potporni vektori*.

Za prikaz dualne forme optimizacijskog problema koristiti ćemo Lagrangeovu funkciju. Time će prethodno definiran problem biti prikazan u idućem obliku:

$$L(w, b, \alpha) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \alpha_i[y_i(\langle w, x_i\rangle + b) - 1],$$

pri čemu su $\alpha_i \geq 0$ Lagrangeovi multiplikatori. Prvi dio izraza je funkcija koju želimo minimizirati dok je drugi dio uvjet. Nužni uvjet za ekstrem funkcije, u ovom slučaju minimum, je izjednačavanje parcijalnih derivacija s nulom.

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{l} y_i\alpha_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{l} y_i\alpha_i x_i,$$
$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{l} y_i\alpha_i = 0.$$

Imamo:

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \alpha_i[y_i(\langle w, x_i\rangle + b) - 1] \\ &= \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j\alpha_i\alpha_j\langle x_i, x_j\rangle - \sum_{i,j=1}^{l} y_i y_j\alpha_i\alpha_j\langle x_i, x_j\rangle + \sum_{i=1}^{l} \alpha_i \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j\alpha_i\alpha_j\langle x_i, x_j\rangle. \end{aligned}$$

Time smo pokazali sljedeću propoziciju.

Propozicija 2.2.4. *Neka je dan linearno odvojiv skup podataka za trening*

$$S = ((x_1, y_1), ..., (x_l, y_l)),$$

i pretpostavimo da parametri α^* rješavaju idući optimizacijski problem:

$$\begin{cases} \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j\alpha_i\alpha_j\langle x_i, x_j\rangle \to max, \\ \sum_{i=1}^{l} y_i\alpha_i = 0, \;\; \alpha_i \geq 0, \;\; i = 1, ..., l. \end{cases}$$

Tada vektor težine $w^* = \sum_{i=1}^{l} y_i\alpha_i^* x_i$ određuje maksimalnu marginu hiperravnine čija je geometrijska margina $\gamma = \frac{1}{\|w\|_2}$.

2.3 Jezgra

Generalno gledajući, primjeri koje bismo željeli klasificirati nisu uvijek linearno odvojivi. Naime, linearan klasifikator ima ograničenu primjenu jer je primjenjiv samo na klasifikaciju linearno odvojivih grupa primjera za učenje. Ukoliko želimo klasificirati linearno neodvojive grupe primjera, služimo se *jezgrom*.

Pojednostavljeni prikaz podataka u drugom prostoru može uvelike olakšati posao. Jedna od strategija u strojnom učenju uključuje promjenu reprezentacije podataka na idući način: Neka je dan vektor $x \in \mathbb{R}^n$ i neka je dana funkcija $\phi : \mathbb{R}^n \to \mathbb{R}^m$, pri čemu je $n < m$. Imamo:

$$x = (x_1, ..., x_n) \longmapsto \phi(x) = (\phi_1(x), ..., \phi_m(x)).$$

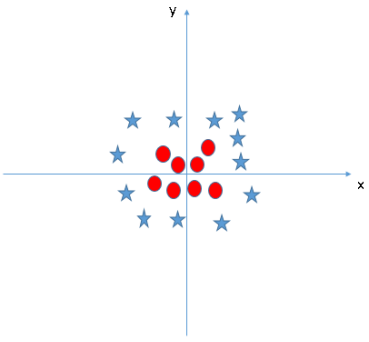
S ovim korakom možemo prikazati podatke iz prostora ulaznih podataka X u novom prostoru značajki $F = \{\phi(x) \mid x \in X\}$.

Implicitni prikaz u prostoru značajki

Da bismo uspijeli riješiti nelinearne probleme pomoću linearne klasifikacije, potrebno je odabrati skup nelinearnih vektora značajki x i prikazati ih u prostoru značajki na kojemu možemo primijeniti linearnu klasifikaciju. Razmatrat ćemo funkciju tipa

$$f(x) = \sum_{i=1}^{m} w_i \phi_i(x) + b,$$

pri čemu je $\phi : X \to F$ nelinearno preslikavanje sa prostora ulaznih podataka u prostor značajki F. Za ilustraciju ćemo se poslužiti idućim primjerom: neka je X dvodimenzionalni prostor ulaznih podataka u kojem su zadani primjeri za trening (Slika 2.2):

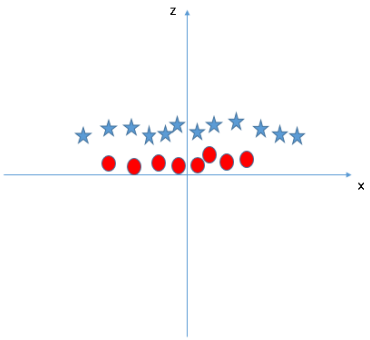


Slika 2.2: Prikaz primjera za trening u prostoru X, slika preuzeta iz [6]

S obzirom na to da primjeri nisu linearno odvojivi u prostoru X, želimo ih prikazati u prostoru značajki F i to tako da možemo primjeniti linearnu klasifikaciju. Jedan od načina je preslikavanje

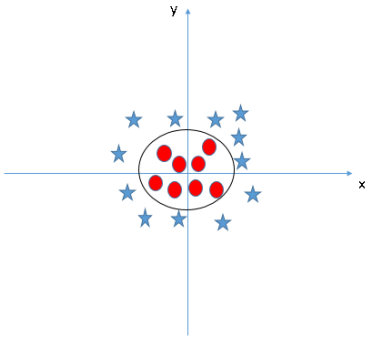
$$(x_1, x_2) \longmapsto \phi(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$$

.



Slika 2.3: Prikaz podataka na x-z ravnini u prostoru značajki F, slika preuzeta iz [6]

Uočimo, prostor značajki F je trodimenzionalan prostor pri čemu su sve vrijednosti na z-osi nenegativne (Slika 2.3). Štoviše, ulazni podaci u prostoru X koji su bliže ishodištu će imati manje vrijednosti na z-osi u prostoru značajki F dok će oni koji su dalje od ishodišta imati veće vrijednosti na z-osi u prostoru značajki F. Nije teško vidjeti da su tako preslikani ulazni podaci linearno odvojivi u F. Također, sada vidimo i kako su oni odvojeni u prostoru X (Slika 2.4).



Slika 2.4: Prikaz klasificiranih vektora značajki u prostoru X, slika preuzeta iz [6]

Kao što smo vidjeli u prethodnom poglavlju, bitno svojstvo stroja za linearno učenje je mogućnost prikaza u dualnoj reprezentaciji. To znači da funkcija koju tražimo može biti

prikazana kao linearna kombinacija primjera za učenje, odnosno, da je možemo odabrati koristeći skalarni produkt primjera za učenje i testnog primjera:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b.$$

Način za izravni izračun skalarnog produkta $\langle \phi(x_i), \phi(x) \rangle$ u prostoru značajki F kao funkciju početnih podataka zovemo *jezgra*.

Definicija 2.3.1. *Kažemo da je funckija $K : X \to \mathbb{R}$ jezgrena funkcija ako je za svaki $x, z \in X$*

$$K(x, z) = \langle \phi(x), \phi(z) \rangle$$

pri čemu je ϕ preslikavanje sa X na prostor značajki F.

Pomoću jezgre možemo implicitno preslikati podatke u prostor značajki i trenirati linearni stroj u takvom prostoru. Također, vrlo često je moguće izračunati $K(x, z)$ puno efikasnije nego da se eksplicitno računa $\langle \phi(x), \phi(z) \rangle$. Ako bismo željeli koristiti ovaj pristup, prvo moramo konstruirati prostor značajki F, doznati koji bi skalarni produkt u tom prostoru bio te konačno naći metodu koja direktno računa tu vrijednost u terminima originalno unesenih vrijednosti. No, da bismo slijedili ove upute, moramo odrediti neka svojstva koja bi jezgrena funkcija trebala zadovoljavati. Očito je da želimo simetričnu funkciju:

$$K(x, z) = \langle \phi(x), \phi(z) \rangle = \langle \phi(z), \phi(x) \rangle = K(z, x)$$

te da bi morala zadovoljavati Cauchy-Schwartz-Bunjakovski nejednakost:

$$K(x, z)^2 = \langle \phi(x), \phi(z) \rangle^2 \leq \|\phi(x)\|^2 \|\phi(z)\|^2 = \langle \phi(x), \phi(x) \rangle \langle \phi(z), \phi(z) \rangle = K(x, x)K(z, z)$$

No, prethodni uvjeti nisu dovoljni da bi nam osigurali postojanje takvog prostora značajki. Mercerov teorem nam daje nužan i dovoljan uvjet da bi $K(x, z)$ bila ispravna jezgra. Dokazi za idući teorem, propoziciju i korolar se može naći u [1].

Teorem 2.3.2. *(Mercerov teorem) Neka je dan proizvoljan skup točaka $\{x_1, ..., x_m\}$. Jezgra $K(x, z)$ je ispravna (Mercerova) ako i samo ako za pripadajuću matricu $K \in \mathbb{R}^{m \times m}$, $K_{ij} = K(x_i, x_j)$ vrijedi da je pozitivno semidefinitna matrica.*

Imajući u vidu ovaj teorem, iskoristit ćemo ga za konstrukciju novih jezgri.

Propozicija 2.3.3. *Neka su K_1 i K_2 jezgre na $X \times X$, $X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^+$, $f(\cdot)$ realna funkcija na X te $\phi : X \to \mathbb{R}^m$. Tada su iduće funkcije jezgre:*
1. $K(x, z) = K_1(x, z) + K_2(x, z)$,
2. $K(x, z) = aK_1(x, z)$,
3. $K(x, z) = K_1(x, z)K_2(x, z)$,
4. $K(x, z) = f(x)f(z)$.

Pomoću prethodne propozicije, lako se dokaže idući korolar.

Korolar 2.3.4. *Neka je $K_1(x, z)$ jezgra na $X \times X$, $x, z \in X$, $\sigma \in \mathbb{R}\backslash\{0\}$ i $p(x)$ polinom sa pozitivnim koeficijentima. Iduće funkcije su jezgre:*
1. $K(x, z) = p(K_1(x, z))$,
2. $K(x, z) = exp(K(x, z))$,
3. $K(x, z) = exp(-\|x - z\|^2/\sigma^2)$.

U ovom radu ćemo koristit 3 osnovne jezgre, koje imaju primjenu u mnogim problemima koji zahtjevaju klasifikaciju:
1. Linearna: $K(x, z) = x^T z$,
2. Polinomijalna: $K(x, z) = (\gamma x^T z + b)^d$; $\gamma > 0$,
3. RBF (eng. *Radial basis function*): $K(x, z) = exp(-\gamma(\|x - z\|^2))$; $\gamma > 0$,
pri čemu su γ, d i b promijenjivi parametri.

Poglavlje 3

Opis i obrada proteina

3.1 Klasifikacija i set podataka

Posljednjih godina, kategorizacija tekstualnih dokumenata je postala jedna od ključnih tehnika za organizaciju informacija. Zadatak je klasificirati određeni tekst ili dokument, na način da ga pridružimo jednoj ili više postojećih, odnosno, određenih klasa. Pri tome pod klasom smatramo skup objekata koji imaju iste karakteristike. Problem klasifikacije teksta ima široku primjenu, poput filtriranja e-mailova, online pretraživanja ili razvrstavanja dokumenata po temi. U ovom radu ćemo analizirati biološke nizove, odnosno, proteine.

Proteini ili bjelančevine su jedan od glavnih čimbenika u rastu i razvoju tkiva te se nalaze u svakoj stanici. Nastaju formiranjem lanaca u čiji sastav ulazi 20 aminokiselina. Također, nalaze se u svoj hrani osim u mastima i rafiniranim šećerima tj. ugljikohidratima. S obzirom na to da su proteini grupirani u familije, odnosno, klase, ponajviše po evolucijskom porijeklu i sličnosti u strukturi, promatrat ćemo 10 klasa proteina na kojima radimo analizu: glicerol 3-fosfat, kromatin, c-ski protein, plasmid, cell-cycle protein, proteasome, UcrQ, Upf2, XendoU i Peptidase S8. Skup za učenje ćemo napraviti tako da iz svake od navedenih klasa uzmemo po 300 različitih proteina, te svakom pridružimo oznaku iz skupa $\{1, 2, ..., 10\}$, ovisno u kojoj familiji se nalaze. Dakle, sveukupno imamo 3000 proteina podijeljenih u 10 klasa.

3.2 Reprezentacija teksta

Obrađivati tekstualni dokument obično podrazumijeva znanje o jeziku na kojem je tekst pisan. Primjerice, metode koje se često koriste prije same obrade teksta su uklanjanje interpunkcijskih znakova iz teksta ili transformiranje riječi u njihove korjene, što očito ne može biti izvršeno bez znanja o jeziku. Dakako, razvoj identifikacije samog jezika i obrade teksta tijekom godina je imao različite pristupe. Jedan od njih je uključivao pravljenje liste

sa svojstvenim riječima i znakovima određenog jezika. Ovaj pristup je imao mnogo mana, poput pretrage za svojstvenim riječima kroz veliku količinu teksta. Također, metoda je jako osjetljiva na prisutnost stranih ili krivo napisanih riječi. Upravo zbog navedenih razloga je metoda bila spora i neprecizna. S druge strane, glavna ideja novijih metoda je bila izrada raspodjele specifičnih "elemenata" za više različitih jezika te potom usporedba s njihovom raspodjelom unutar danog teksta. Takvu analogiju možemo primijeniti i na različite vrste teksta, ne samo jezike. Naime, ukoliko uspoređujemo medicinsku i tehničku literaturu, očito je da postoje riječi specifične za svaku od njih, odnosno, u medicinskom tekstu će se više pojavljivati riječi poput "virus", "igla" i "operacija", dok će tehnički sadržavati "električni", "sila" i "moment". Poprilično je jasno da ni ovo nije najbolji pristup iz brojnih razloga, kao što su nemogućnost raspoznavanja tekstova različitih tema u kojima se pojavljuju iste riječi ili dokumenata iste tematike sa različitim svojstvenim riječima. Također, mnoge riječi koje se smatraju specifičnima za tekst određene teme mogu ostati neiskorištene.

U ovom radu ćemo koristiti metodu koja obuhvaća dio navedenih pristupa. Riječ je o klasifikaciji temeljenoj na *n-gramima*, odnosno, odsječcima veličine n neke riječi veličine m, pri čemu je $n \leq m$. Pri tome koristimo *n-gramski* model, u kojemu su zanemareni gramatika i poredak.

Proteini su nizovi bez separatora, zadani u odgovarajućem alfabetu koji koristi 20 engleskih slova. Pri tome je protein izgrađen od prosječno 466 aminokiselina, a s obzirom na to da svako slovo navedenog alfabeta predstavlja jednu aminokiselinu, možemo reći da je prosječni protein duljine 466 slova bez separatora. Također, nedostatak separatora znači da nemamo rječnik, odnosno, skup riječi koje čine tekst. Vlastiti *n-rječnik* ćemo napraviti pomoću 5-grama, tj. *petorki*. Radi jednostavnosti, u nastavku rada ćemo n-rječnik jednostavno zvati rječnik.

Neka je dan dio proteina iz prve klase

$$MNQVLKDALEDNPIIVAIKDDAGLQ...$$

Redom prolazimo po proteinu te uzimamo niz svakih 5 slova zaredom, što će nam u rječniku biti jedna riječ.

$$\underline{MNQVL}KDALEDNPIIVAIKDDAGLQ... \rightarrow MNQVL,$$

$$M\underline{NQVLK}DALEDNPIIVAIKDDAGLQ... \rightarrow NQVLK,$$

$$MN\underline{QVLKD}ALEDNPIIVAIKDDAGLQ... \rightarrow QVLKD...$$

Na taj način iz prvog proteina dobivamo 184 petorke, odnosno, 184 riječi. Postupak ponovimo za drugi protein gdje dobivamo još 187 riječi. Nakon što smo na isti način prošli cijeli dokument sa 3000 proteina, dobivamo rječnik od ukupno različitih 603531 riječi. Uočimo da će svaka riječ koju smo dobili na ovaj način, te time napravili rječnik, biti iskorištena

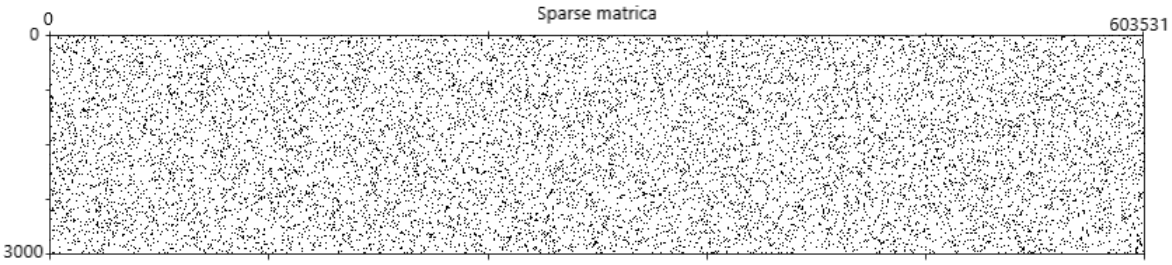
barem jednom. Dodatno, korisno je bilo napraviti i tablicu frenkvencija po klasama proteina:

$$MNQVL \quad 1\,0\,0\,0\,0\,0\,0\,1\,0\,0,$$
$$NQVLK \quad 1\,0\,0\,1\,0\,0\,0\,0\,0\,0,$$
$$QVLKD \quad 1\,0\,0\,0\,0\,0\,0\,0\,0\,0.$$

Iz tablice je vidljivo da se prva riječ MNQVL pojavljuje jednom u prvoj klasi i jednom u osmoj klasi. Riječ će se pojavljivati barem jednom u onoj klasi iz kojega je uzet protein da bi se dobila riječ. Analogno, vidimo da se riječ NQVLK pojavljuje u prvoj i četvrtoj klasi dok se riječ QVLKD pojavljuje samo u prvoj. Također, na ovaj način smo stekli uvid u kojim klasama je riječ najprisutnija,

3.3 Document-term matrica

Nakon što smo generirali rječnik pomoću 5-grama, odnosno petorki, potrebno je prikazati svaki pojedini protein kao vektor značajki pomoću riječi iz rječnika te ćemo tako dobiti skup za učenje kao kolekciju primjera, jer znamo kojoj klasi proteina takav vektor značajki pripada. Jedan od načina za to je *document-term matrica*. Radi se o matrici koja opisuje dokumente pomoću frekvencije pojmova, tj. riječi. Pri tome i-ti red označava i-ti dokument u kolekciji dok se u j-tom stupcu nalazi j-ti pojam. Na (i, j)-tom mjestu se tada nalazi broj koji označava koliko se puta j-ti pojam pojavio u i-tom dokumentu. S obzirom na to da naš rječnik ima 603531 pojmova (riječi) te 3000 dokumenata (proteina), document-term matrica je tipa (3000×603531) pri čemu se na (i, j)-tom mjestu nalazi broj koji označava koliko puta se j-ta riječ pojavila u i-tom proteinu. Samim time nam i-ti redak predstavlja $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,603531})^T$ vektor značajki za i-ti protein, $i = 1, ..., 1800$. Uočimo da će tako generirana matrica imati jako veliki broj nula. Naime, ukoliko gledamo prvu riječ MNQVL, koja se pojavljuje samo jednom u prvom proteinu, tada će na mjestu $(i, 1)$ biti 1, no ako je se ona ne pojavljuje niti u jednom drugom proteinu, na mjestu $(i, 1)$, $i = 2, ..., 300$ će biti broj 0. Slično tome, ukoliko se u prvom proteinu jednom pojavljuju samo one riječi koje smo dobili kao petorke njega samoga, na mjestu $(1, j)$, $j = 1, ..., 184$ će stajati 1, dok će na mjestu $(1, j)$, $j = 185, ..., 603531$, biti 0. Imajući to u vidu, grafički prikaz naše matrice izgleda slično kao na slici 3.1:



Slika 3.1: Primjer rijetke matrice

Takva vrsta matrice naziva se rijetka matrica (eng. sparse matrix) te je sa njom vrlo nepraktično raditi, s obzirom na to da svaka 0, kojih je očito jako puno, zauzima memoriju. Jedan od načina da ubrzamo proces i oslobodimo radnu memoriju je zapis te matrice samo pomoću onih elemenata koji su različiti od 0. To nam omogućuje paket *SciPy* u Python-u, sa naredbom *sparse*:

```
from scipy import sparse
from scipy.sparse import csr_matrix.
```

Poglavlje 4

Rezultati

Skup za učenje dobili smo tako što smo svaki protein prikazali kao vektor značajki pomoću riječi našeg rječnika te im pridružili odgovarajuće oznake klase proteina. Za taj skup ćemo pomoću algoritma potpornih vektora generirati model na temelju kojega se određuje kojoj familiji, odnosno klasi pripada testni podatak. Skup podataka za testiranje generirati ćemo iz istih familija proteina iz kojih smo generirali skup za učenje. S obzirom na to da smo za podatke za učenje uzeli po 300 proteina iz svake familije, za testne podatke ćemo uzeti još 100 različitih proteina iz svake familije. Imajući u vidu da su proteini grupirani u klase po evolucijskom porijeklu i sličnosti strukture, želimo provjeriti koliko dobro algoritam potpornih vektora može klasificirati nove proteine. Ako znamo iz koje su familije testni podaci, lako možemo pratiti koliko ih je dobro algoritam klasificirao.

Za primjer uzmimo 100 proteina za testiranje iz prve grupe. Da bismo ih prikazali pomoću rječnika koji imamo, moramo generirati document-term matricu. Na (i, j)-tom mjestu će se nalaziti broj koji označava koliko se puta j-ta riječ iz našeg rječnika pojavila u i-tom proteinu ($i = 1, ..., 100,\ j = 1, ..., 603531$). Odmah uočimo nekoliko stvari: za očekivati je da će ova matrica imati više brojeva u, otprilike, prvoj desetini broja stupaca iz razloga što su te riječi generirane iz proteina prve familije, koja ima zajednička svojstva sa testnim proteinima iste familije. Nadalje, vidimo da će tako generirana matrica tipa (100×603531) također biti rijetka matrica. Štoviše, s obzirom da rječnik nije generiran iz testnih primjera već iz primjera za učenje, broj nula će biti još veći. Također, za razliku od primjene rječnika na skup za učenje, na testnom skupu ne možemo očekivati da će svaka riječ biti "iskorištena". Na isti način generiramo matricu za ostale grupe proteina za testiranje. Time je svaki protein za testiranje prikazan kao testni vektor u prostoru podataka.

Testiranje ćemo provesti na način da ćemo uvrstiti testne proteine iz određene grupe u algoritam potpornih vektora, te ćemo, s obzirom da znamo iz koje grupe su testni proteini, gledati koliki broj njih je algoritam klasificirao u pravu grupu. Pri tome ćemo *postotak*

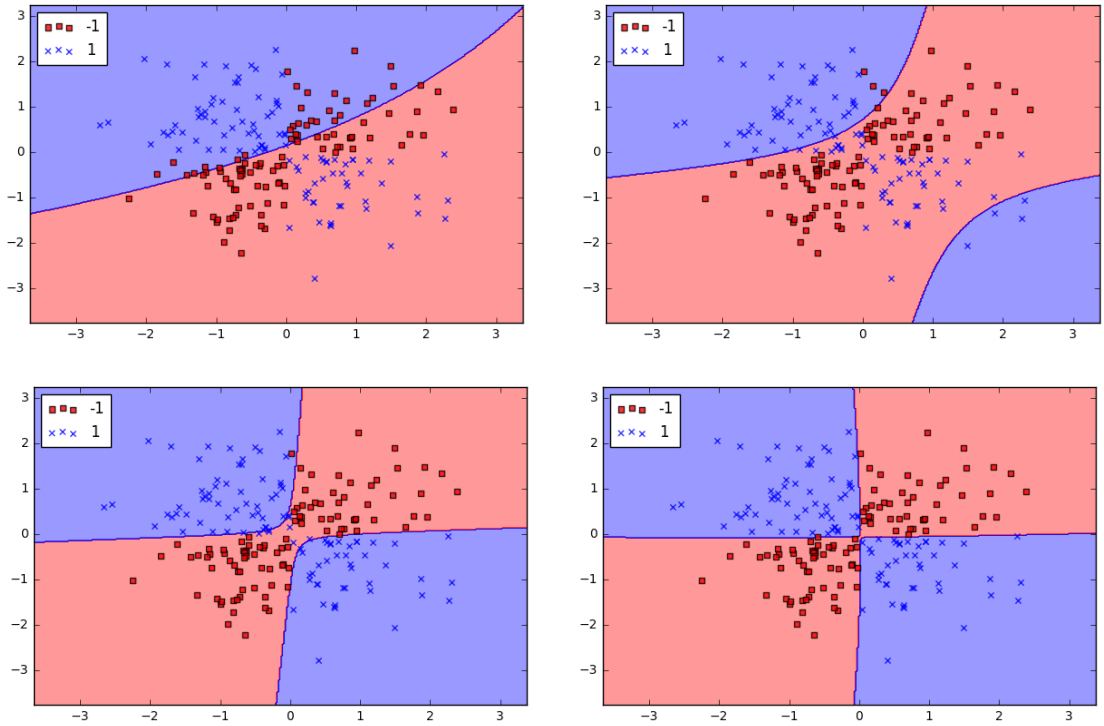
dobro klasificiranih testnih proteina izračunati nakon što smo uvrstili svaki testni protein, dakle njih 1000, a ne samo određenu grupu.

U slučaju kada su podaci linearno odvojivi, moguće ih je klasificirati odgovarajućom hiperravninom. Međutim, mi ne znamo jesu li podaci linearno odvojivi te je li ih moguće odvojiti na taj način. Stoga ćemo koristiti tri vrste jezgrenih funkcija: linearnu, polinomijalnu i RBF te vidjeti koja od njih daje najbolju klasifikaciju i uz koje parametre. Također, u idealnom slučaju, kada se podaci klasificiraju, algoritmom potpornih vektora ćemo dobiti hiperravninu koja potpuno odvaja grupe koje se ne preklapaju. Međutim, potpuno odvajanje grupa često nije moguće. Kako bi se dopustilo određeno odstupanje prilikom klasifikacije, algoritam potpornih vektora sadrži parametar C.

Da bismo objasnili parametar C, moramo uvesti nenegativne dopunske (eng. *slack*) varijable ξ_i, $i = 1, ..., l$ koje predstavljaju mjeru udaljenosti pojedinog uzorka od margine razdvajanja na kojoj leže potporni vektori. Pri tome je $\xi_i = 0$ ako je i-ti uzorak potporni vektor, $0 < \xi_i < 1$ ako se i-ti uzorak nalazi unutar margine, ali s ispravne strane hiperravnine, $\xi_i = 1$ ako i-ti uzorak leži na odvajajućoj hiperravnini, $\xi_i > 1$ ako se i-ti uzorak nalazi s pogrešne strane hiperravnine. Koristeći ξ_i, $i = 1, ..., l$, ukupnu pogrešku klasifikacije možemo prikazati kao sumu svih mjera udaljenosti podataka za trening koji se nalaze unutar ili s pogrešne strane margine razdvajanja. Za takav slučaj, gdje također ne želimo da svaki ξ_i poprimi previše veliku vrijednost, dobivamo sljedeći optimizacijski problem:

$$\begin{aligned}
&\langle w, w\rangle + C\sum_{i=1}^{l}\xi_i^2 \rightarrow min_{\xi,w,b}\\
&y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \quad i = 1, ..., l\\
&\xi_i \geq 0, \quad i = 1, ..., l.
\end{aligned}$$

U ovom optimizacijskom problemu, za parametar C možemo reći da predstavlja "ravnotežu" između veličine margine i ukupne pogreške prethodno opisanih podataka. Štoviše, njime se određuje koliko dopuštamo algoritmu potpornih vektora da krivo klasificira podatke za trening. Kada odredimo malu vrijednost parametra C, klasifikatoru dopuštamo da krivo klasificira neke od podataka za trening. S druge strane, odabirom velike vrijednosti C, algoritam će striktno klasificirati podatke no time se postiže jako, možda i previše precizan model, što u konačnici može rezultirati lošom klasifikacijom nepoznatih, odnosno testnih podataka. Uzmimo za primjer nasumične podatke te RBF jezgru na kojima ćemo pokazati kako parametar C utječe na klasifikaciju podataka.



Slika 4.1: Klasifikacija podataka za $C = 10$; $C = 1000$; $C = 10000$; $C = 100000$; slike preuzete iz [7]

U nastavku ćemo priložiti i komentirati rezultate za tri tipa jezgri te vidjeti za koju jezgrenu funkciju, u ovisnosti o različitim parametrima, algoritam najbolje klasificira podatk.

Linearna jezgra; $K(x, z) = x^T z$

S obzirom na to da je parametar C jedini kojega možemo mijenjati, testiranje za klasifikaciju pomoću linearne jezgre ćemo provesti za $C = 0.01$, $C = 0.1$, $C = 1$, $C = 10$, $C = 100$, $C = 1000$ i $C = 10000$ te ćemo rezultate prikazati pomoću tablice. Primjerice, ako testiramo koliko je ispravno klasificiranih proteina iz prve familije, kao rezultat ćemo dobiti idući vektor:
[1117111111111117111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111]
Broj ispravno klasificiranih proteina 1. familije: 98,
što znači da je od 100 testnih proteina koje smo odabrali, algoritam klasificirao njih 98 u 1. familiju, dok je četvrti i šesnaesti protein klasificirao u 7. familiju. Tablicom ćemo poka-

zati broj ispravno klasificiranih podataka za svaku familiju s obzirom na različiti parametar C te kao konačni rezultat, postotak svih ispravno klasificiranih testnih proteina (Tablica 4.1).

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	Ukupno
$C = 0.01$	98	93	98	97	90	81	100	91	91	91	93.0%
$C = 0.1$	99	93	98	98	90	82	100	92	93	93	**93.8%**
$C = 1$	99	93	98	98	90	82	100	92	93	93	**93.8%**
$C = 10$	99	93	98	98	90	82	100	92	93	93	**93.8%**
$C = 100$	99	93	98	98	90	82	100	92	93	93	**93.8%**
$C = 1000$	99	93	98	98	90	82	100	92	93	93	**93.8%**
$C = 10000$	99	93	98	98	90	82	100	92	93	93	**93.8%**

Tablica 4.1: Ispravno klasificirani testni proteini - linearna jezgra

Uočimo, stavljajući $C = 0.01$, dopustili smo algoritmu da krivo klasificira neke podatke za trening, odnosno, da margina između odvajajuće hiperravnine i potpornog vektora bude veća, zanemarujući pri tome određeni broj podataka za trening koji se nalazi unutar margine ili sa druge strane hiperravnine. To je rezultiralo sa 93% točnosti klasifikacije testnih podataka. Ipak, povećavajući parametar, odnosno, za $C = 0.1$, tu smo marginu smanjili i zahtjevali manju pogrešku u smislu krivo klasificiranih podataka za trening. Time je točnost klasifikacije testnih podataka dostigla 93.8%. Zanimljivo je vidjeti da isti postotak točnosti klasifikaicije dobivamo i za $C = 1$, $C = 10$, $C = 100$, $C = 1000$ te $C = 10000$, iako smo postupno zahtjevali što manje krivo klasificiranih podataka za trening. Štoviše, postotak je isti i na razini familije testnih proteina. Iz navedenog se može naslutiti da je odvajajuća hiperravnina jako dobra te da je teško možemo poboljšati u smislu promjene nagiba ili pomaka, jer i kada dopuštamo određeni broj pogrešaka podataka za trening dobivamo istu točnost testnih podataka kao kada želimo striktnu klasifikaciju podataka za trening.

Polinomijalna jezgra; $K(x, z) = (\gamma x^T z + b)^d$; $\gamma > 0$

Kao što možemo primjetiti, ukoliko u funkciju polinomijalne jezgre uvrstimo $\gamma = 1$, $b = 0$ i $d = 1$, dobivamo upravo linearnu jezgru. Stoga nas zanima možemo li ikako poboljšati klasifikaciju varijacijom tih parametara. Radi jednostavnosti, tablicom ćemo prikazati postotak svih ispravno klasificiranih testnih proteina za one parametre γ, b i d koji su nam potreni za analizu klasifikacije pomoću polinomijalne jezgrene funkcije primjenjene na našim podacima (Tablica 4.2).

C	γ	d	Ukupno
0.01	0.001	1	24.9%
0.01	0.1	1	88.0%
0.01	1	1	93.0%
0.01	10	1	**93.8%**
0.01	100	1	**93.8%**
1	100	1	**93.8%**
100	100	1	**93.8%**
0.01	0.001	2	12.6%
0.01	1	2	59.4%
0.01	100	2	59.4%
1	100	2	59.4%
100	100	2	59.4%
1000	10	2	59.4%
0.01	1	3	43.2%
1	100	3	43.2%
100	10	3	43.2%
1000	100	3	43.2%

Tablica 4.2: Ispravno klasificirani testni proteini - polinomijalna jezgra

Odmah primjećujemo da klasifikacija testnih proteina ovisi o stupnju polinomijalne jezgrene funkcije. Naime, što je veći stupanj biti će i veća zakrivljenost razdvajajuće funkcije. S obzirom na to da je većim stupnjem klasifikacija lošija, možemo naslutiti da su podaci za trening u prostoru podataka raspoređeni na način da kada ih želimo klasificirati odvojivom funkcijom, dobivamo previše precizan model. Najveću klasifikaciju testnih proteina dobivamo upravo za stupanj $d = 1$. Primijetimo također da se varijacijom sa parametrom C ne utječe na klasifikaciju. Parametar koji utječe na klasifikaciju u ovom slučaju je γ, i to posebno za slučaj $0 < \gamma < 1$. Ukoliko obratimo pažnju na izraz $\gamma x^T z$, vidimo da ako je $\gamma \geq 1$, vrijednost funkcije će se proporcionalno povećati za svaka dva podatka za trening, što bi trebalo dodatno olakšati klasifikaciju. Suprotno tome, skalarni produkt dvaju vektora koji može davati veliku vrijednost, pomnožen sa $\gamma < 1$ daje manju vrijednost te to, primjenjeno na svaki podatak za trening, uvelike može utjecati na kompliciranost modela te samim time težu klasifikaciju testnih podataka. Također, u tablici smo izostavili slobodni parametar b jer on ne utječe na model.

RBF jezgra; $K(x, z) = exp(-\gamma||x - z||^2)$; $\gamma > 0$

U strojnom učenju, RBF jezgra je popularna jezgrena funkcija koja se koristi za mnoge vrste problema, posebno za klasifikaciju pomoću algoritma potpornih vektora, koristeći euklidsku normu dva vektora. Parametar kojim možemo varirati u svrhu poboljšanja klasifikacije je γ. Dodatno, ukoliko fiksiramo parametar $\gamma > 0$, jezgrena funkcija će poprimati veće vrijednosti za one vektore čija je euklidska norma manja. Slično kao i u prethodnom slučaju polinomijalne jezgre, tablicom ćemo prikazati postotak svih ispravno klasificiranih testnih proteina koji su nam potrebni za analizu klasifikacije pomoću RBF jezgrene funkcije.

C	γ	Ukupno
0.01	0.01	31.0%
0.01	0.1	13.4%
0.01	1	17.7%
0.01	10	10.9%
0.01	100	10.7%
1	0.0001	61.9%
1	0.01	57.1%
1	0.1	16.9%
1	1	11.6%
1	10	11.3%
1	100	11.1%
2	0.01	61.3%
4	0.01	62.8%
8	0.01	63.0%
100	0.01	63.0%
100	0.0001	**93.4%**
100	100	11.1%
1000	0.0001	**93.4%**

Tablica 4.3: Ispravno klasificirani testni proteini - RBF jezgra

Kao u slučaju polinomijalne jezgrene funkcije, primjećujemo da se varijacijom sa parametrom C ne utječe previše na točnost klasifikacije testnih proteina. Naime, kada fiksiramo $C = 0.01$, time smo smanjili marginu i dopustili pogreške pri klasifikaciji podataka za trening. Pri tome smo dobivali bolju klasifikaciju testnih podataka smanjivanjem parametra γ, posebno za $\gamma < 1$. Ukoliko je $C = 1$, točnost klasifikacije također ovisi o parametru γ, te slično vidimo za $C = 100$ i $C = 1000$. Stoga možemo reći da parametar γ ima veći utjecaj

pri korištenju RBF jezgrene funkcije. Uočimo da smo bolju točnost klasifikacije testnih proteina imali za $\gamma = 0.0001$ i $\gamma = 0.01$ nego za $\gamma \geq 1$. Naime, za $0 < \gamma < 1$ smo povećali vrijednost jezgrene funkcije te smo time olakšali klasifikaciju. Suprotno tome, smanjivanjem vrijednosti jezgrene funkcije za svaka dva podatka za trening dobivamo model koji je previše precizan i samim time loše klasificira testne podatke. Najbolju klasifikaciju smo postigli za $C = 100$ i $\gamma = 0.0001$ te je algoritam u tom slučaju pravilno klasificirao 93.4% testnih proteina.

Poglavlje 5

Zaključak

Podatke za vježbu u visokodimenzionalnom prostoru nije moguće prikazati vizualno. Štoviše, nećemo ih moći vizualno prikazati niti u prostoru značajki, s obzirom na to da je dimenzija prostora značajki veća nego dimenzija prostora ulaznih podataka. Stoga u većini slučajeva ne možemo niti naslutiti kako su podaci raspoređeni te jesu li linearno odvojivi ili ne. Upravo zbog toga ne možemo sa sigurnošću tvrditi koja je prava odvajajuća funkcija koja bi ih najbolje klasificirala. Često je jedini način za pronalazak najbolje odvajajuće funkcije upravo isprobavanje različitih modela te odabir onoga koji nam daje najbolje rezultate.

Vidjeli smo da je točnost klasifikacije linearne jezgre za naše podatke bila 93.8%. Štoviše, točnost je bila ista i kada smo dopuštali krivo klasificirane podatke za vježbu i kada smo zahtijevali strogu klasifikaciju svih podataka. To znači da nismo napravili previše precizan model koji nam generalno nije poželjan. Štoviše, moguće je da su podaci raspoređeni tako da ih linearni klasifikator odvaja u potpunosti, ali zbog strukture proteina i manjka podataka za trening na kojem treniramo algoritam, testni protein iz jedne skupine ima više zajedničkih atributa s proteinima za vježbu iz neke druge skupine.

Iako je pojednostavljena polinomijalna jezgrena funkcija u biti linearna, za očekivati je bilo da ćemo kompleksnijim modelom postići bolju klasifikaciju. Međutim, kako smo povećavali stupanj polinomijalne funkcije, klasifikacija je postajala lošija. Štoviše, najbolju klasifikaciju točnosti testnih podataka od 93.8% smo postigli upravo kada smo pomoću parametara od polinomijalne napravili linearnu funkciju. Također, ukoliko smo tu namještenu linearnu jezgrenu funkciju uvećavali skalarom γ, također nismo mogli postići bolju klasifikaciju, već smo dobili istu, iako smo time proporcionalno povećali vrijednost funkcije za svaka dva podatka za vježbu.

RBF jezgrena funkcija uspješno rješava mnoge probleme klasifikacije za mnoge primjere koji koriste strojno učenje, ne samo klasifikacija teksta. Stoga smo htjeli vidjeti možemo li primjenom upravo te funkcije postići bolju klasifikaciju. Iako su rezultati klasifikacije testnih proteina bili uglavnom od 11% - 63%, što možemo smatrati lošim rezulta-

tima s obzirom da smo u prethodna dva slučaja imali 93.8% točnosti, RBF funkcijom smo uspjeli postići 93.4%. Problem je u tome što smo taj rezultat postigli nakon što smo za parametre funkcije stavljali skoro pa ekstremne vrijednosti te time gotovo sigurno napravili preopterećen i previše precizan model.

Možemo reći da je za naše podatke linearna jezgrena funkcija klasificirala podatke bolje od druge dvije jezgre. Također, kada bismo imali veći skup podataka za trening, za očekivati je da bi i u tom slučaju linearna jezgrena funkcija davala najbolje rezultate klasifikacije. Bitno je napomenuti da navedeni rezultati vrijede isključivo za naš skup podataka koji smo koristili no promjenom skupa bismo najvjerojatnije dobili slične rezultate.

Bibliografija

[1] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge university press, 2000.

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

[3] D. Bakić, *Linearna algebra*, https://web.math.pmf.unizg.hr/b̃akic/la/linearna_algebra_sk_7.pdf, Sveučilište u Zagrebu, Matematički odjel PMF-a, Zagreb 2008.

[4] M. Polonijo, D. Crnković, M. Bombardelli, T. Ban Kirigin, Z. Franušić, R. Sušanj, Z. Iljazović, *Euklidski prostori*, https://web.math.pmf.unizg.hr/nastava/eukl/EP.pdf, Sveučilište u Zagrebu, Matematički odjel PMF-a.

[5] D. Bertsimas, J. N. Tsitsiklis, *Introduction to linear optimization*, MIT, Belmont, Massachusetts, 1997.

[6] S. Ray, *Understanding Support Vector Machine algorithm from examples*, [online] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/, 6. listopad, 2015.

[7] C. Albon, *SVC Parameters When Using RBF Kernel*, [online] https://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/, 20. prosinac, 2017.

[8] S. Sayad, *Support Vector Machine - Classification (SVM)*, [online] https://www.saedsayad.com/support_vector_machine.htm

Sažetak

Cilj ovog rada je klasifikacija proteina pomoću algoritma potpornih vektora i n-gramskog rječnika. Prvo su dani osnovni pojmovi iz linearne algebre, euklidskih prostora i teorije optimizacije, potrebni za lakše shvaćanje tema obrađenih u radu. Objašnjena je matematička pozadina načina rada algoritma, s posebnim naglaskom na jezgrene funkcije koje se koriste. Za analizu se koristi deset proteinskih familija, pri čemu je iz svake familije uzeto 300 proteina za trening i 100 proteina za testiranje. Nakon što je napravljen n-gramski rječnik, provedeni su testovi klasifikacije pomoću algoritma potpornih vektora te su uspoređivani rezultati za različite jezgrene funkcije u ovisnosti o parametrima.

Summary

The aim of this work is to classify proteins using the support vector machine and the n-gram dictionary. First, we present some basic concepts from linear algebra, euclidean spaces, and optimization theory that are needed later on. A mathematical background of the algorithm has been explained in detail, with a special emphasis on the kernel functions. Ten protein families were used for the analysis, using 300 protein for each family for training and 100 test proteins. After the n-gram vocabulary was made, classification tests were performed using the support vector machine, and comparisons were made for different kernel functions depending on the parameters.

Životopis

Rođen sam 16.04.1993. godine u Zagrebu. Osnovnu školu Rapska upisujem 2000. godine koju završavam 2008. godine. Nakon toga, od 2008. do 2012. godine pohađam Graditeljsku tehničku školu u Zagrebu. Odmah po završetku srednje škole, upisujem Preddiplomski studij matematike, nastavnički smjer, na Prirodoslovno-matematičkom fakultetu u Zagrebu, kojeg završavam 2016. godine. Iste godine upisujem Diplomski sveučilišni studij Matematička statistika na istom fakultetu.