Tehnike učenja za klasifikaciju bioloških nizova

Pezić, Marija

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Marija Pezić

TEHNIKE UČENJA ZA KLASIFIKACIJU BIOLOŠKIH NIZOVA

Diplomski rad

Voditelj rada:
doc.dr.sc. Pavle Goldstein

Zagreb, veljača, 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

Biološki nizovi su nizovi znakova iz odgovarajućeg biološkog alfabeta bez separatora. Primjer bioloških nizova su proteini, kemijske tvari koje se nalaze u svim stanicama živih bića i izravno sudjeluju u svim procesima unutar stanice, što ih čini osnovom života na Zemlji. Izgrađeni su od aminokiselina međusobno povezanih u lance. Svaka aminokiselina predstavljena je jednim od 20 slova engleskog alfabeta što čini biološki alfabet za analiziranja proteina. Po evolucijskom podrijetlu proteini se grupiraju u proteinske familije. Proteini unutar iste familije imaju sličnu strukturu i sadrže slične nizove aminokiselina što je jasan pokazatelj podrijetla nekog proteina. Postupak identificiranja familije kojoj neki protein pripada naziva se klasifikacija.

Cilj ovog rada je prepoznati karakteristične motive proteinske familije u svrhu što točnije klasifikacije proteina. Identificiranje karakterističnih motiva, odnosno podnizova aminokiselina koji se često pojavljuju, zanimljivije je pitanje od klasifikacije jer ne postoji eksplicitno objašnjenje kako pronaći značajke koje pridonose točnosti klasifikacije. Osim pronalaženja značajki koje identificiraju familije potrebno je takve značajke matematički reprezentirati kako bi se poznate metode klasifikacije mogle uspješno primijeniti. Stoga ćemo u ovom radu prikazati jedan način određivanja karakterističnih motiva proteinske familije te kako se takvi motivi mogu prikazati u obliku vektora. Za klasifikaciju ćemo koristiti metode linearne klasifikacije, točnije klasifikaciju max-normom te stroj potpornih vektora (eng. *Support Vector Machine, SVM*) algoritam za učenje koji se najčešće primjenjuje kod problema klasifikacije.

Navedeni postupci bit će opisani u trećem poglavlju, dok ćemo rezultate ovakvog pristupa prikazati u četvrtom poglavlju. Kao uvod u matematičke pojmove i okvire problema linearne klasifikacije poslužit će nam prva dva poglavlja.

Poglavlje 1

Matematički pojmovi

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.

Definicija 1.1.2. *Neka je V neprazan skup na kojem su zadane binarna operacija zbrajanja $+ : V \times V \to V$ i operacija množenja sklarima 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.

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

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

pri čemu su $a_1, a_2, ..., a_k \in V$, $\alpha_1, \alpha_2, ..., \alpha_k \in \mathbb{F}$ 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 poljem $\mathbb{F}$ i*

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

konačan skup vektora iz V. Kažemo da je skup S linearno nezavisan ako vrijedi

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

U suprotnom kažemo da je skup S linearno zavisan.

Definicija 1.1.5. *Neka je V vektorski prostor nad poljem $\mathbb{F}$ i $S \subseteq V$, $S \neq \emptyset$. Linearnu ljusku skupa S označavamo $[S]$ i definiramo kao*

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

Linearna ljuska praznog skupa definira se kao $[\emptyset] = \{0\}$.

Definicija 1.1.6. *Neka je V vektorski prostor i $S \subseteq V$. Kažemo da je skup S sustav izvodnica za 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.*

Definicija 1.1.8. *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. $\langle x, y \rangle = \overline{\langle y, x \rangle}, \ \forall x, y \in V$.

Definicija 1.1.9. *Vektorski prostor na kojem je definiran skalarni produkt zove se unitaran prostor.*

Definicija 1.1.10. *Neka je V unitaran prostor. Kažemo da su vektori $x, y \in V$ međusobno okomiti ili ortogonalni ako je $\langle x, y \rangle = 0$.*

Definicija 1.1.11. *Neka je V unitaran prostor. Norma na V je funkcija*

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

definirana s

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

Propozicija 1.1.12. *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 1.1.13. *Svaka funkcija* $\|\cdot\| : V \to \mathbb{R}$ *na vektorskom prostoru V sa svojstvima iz 1.1.12 naziva se norma. Tada* $(V, \|\cdot\|)$ *nazivamo normirani prostor.*

Neka je dan vektor $x = (x_1, ..., x_n) \in V$ nad poljem $\mathbb{R}^n$. Euklidska ili 2-norma dana je sa $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$. Max-norma dana je sa $\|x\|_{max} = \max\{x_1, ..., x_n\}$.

Definicija 1.1.14. *Neka je V unitaran prostor. Na V definiramo preslikavanje*

$$d : V \times V \to \mathbb{R}$$

formulom

$$d(x, y) = \|x - y\|$$

te ga nazivamo metrika ili udaljenost vektora x od vektora y.

Propozicija 1.1.15. *Metrika na unitarnom prostoru V ima sljedeća svojstva:*

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

Definicija 1.1.16. *Neka je* $X \neq \emptyset$. *Svaka funkcija* $d : X \times X \to \mathbb{R}$ *sa svojstvima iz 1.1.15 naziva se metrika ili udaljenost. Tada* (X, d) *nazivamo metrički prostor.*

Definicija 1.1.17. *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 1.1.18. *Neka je* $A \in M_{mn}(\mathbb{R})$. *Transponirana matrica* A^T *matrice* $A = [a_{ij}]$ *definirana je sa* $A^T = [a_{ji}]$.

Definicija 1.1.19. *Neka je A kvadratna matrica. Kažemo da je A simetrična ako je* $A^T = A$.

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*

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

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

ekvivalentan je problemu

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

Dokaz. Neka je $g(x) := (g_1(x), g_2(x), ..., g_m(x))$. Tada za fiksni $x \in \mathbb{R}^n$ vrijedi:

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

Za $g(x) \leq 0$ maksimum funkcije L po varijabli $\alpha \geq 0$ postiže se za $\alpha = 0$. S druge strane, ako je $g_i(x) > 0$ za neki $i \in \{1, 2, ..., m\}$, povećanjem vrijednosti komponenata vektora $\alpha \in \mathbb{R}^m_+$ možemo proizvoljno povećati funkciju $L(x, \alpha)$. Minimizacijom po $x \in \mathbb{R}^n$ uočavamo da se minimum funkcije $\max L(x, \alpha)$ postiže za $g(x) \leq 0$, te je on jednak minimumu funkcije f na dopustivom skupu $U = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$. Stoga su navedeni problemi ekvivalentni. □

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

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

1.3 Vjerojatnost i statistika

Definicija 1.3.1. *Slučajni pokus je pokus čiji ishod nije jednoznačno određen uvjetima pod kojima se pokus vrši.*

Definicija 1.3.2. *Prostor elementarnih događaja nekog slučajnog pokusa jest neprazan skup Ω sa svojstvom da svakom ishodu pokusa odgovara točno jedan element skupa. Elemente skupa Ω nazivamo elementarni događaji.*

Definicija 1.3.3. *Neka je $\omega \in \Omega$ događaj vezan uz neki slučajni pokus. Pretpostavimo da smo pokus ponovili n puta te da se događaj ω dogodio točno n_ω puta. Tada broj n_ω zovemo frekvencija događaja ω, a broj $\frac{n_\omega}{n}$ zovemo relativna frekvencija događaja ω.*

Definicija 1.3.4. *Familija $\mathcal{F}$ podskupova od Ω ($\mathcal{F} \subseteq \mathcal{P}(\Omega)$) jest σ-algebra skupova na Ω ako vrijedi:*

(F_1) $\emptyset \in \mathcal{F}$

(F_2) $A \in \mathcal{F} \Rightarrow A^C \in \mathcal{F}$

(F_3) $A_i \in \mathcal{F} \Rightarrow \cup_{i=1}^{\infty} A_i \in \mathcal{F}$

Definicija 1.3.5. *Neka je $\mathcal{F}$ σ-algebra na Ω. Uređeni par $(\Omega, \mathcal{F})$ zove se izmjeriv prostor.*

Definicija 1.3.6. *Neka je $(\Omega, \mathcal{F})$ izmjeriv prostor. Funkcija $\mathbb{P} : \mathcal{F} \to \mathbb{R}$ jest vjerojatnost ako vrijedi:*

(P_1) $\mathbb{P}(A) \geq 0, \ \forall A \in \mathcal{F}$ (*nenegativnost*)

(P_2) $\mathbb{P}(\Omega) = 1$ (*normiranost*)

(P_3) $A_i \in \mathcal{F}, i \in \mathbb{N}$ *te* $A_i \cap A_j = \emptyset$ *za* $i \neq j \Rightarrow \mathbb{P}(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$ $(\sigma - aditivnost)$

Uređena trojka $(\Omega, \mathcal{F}, \mathbb{P})$ *gdje je* $\mathcal{F}$ σ*-algrebra na* Ω *i* $\mathbb{P}$ *vjerojatnost na* $\mathcal{F}$ *zove se vjerojatnosni prostor. Za događaj* $A \in \mathcal{F}$ *broj* $\mathbb{P}(A)$ *nazivamo vjerojatnost događaja A.*

Definicija 1.3.7. *Neka je* $(\Omega, \mathcal{F}, \mathbb{P})$ *vjerojatnostni prostor i* $A \in \mathcal{F}$ *takav da je* $\mathbb{P}(A) > 0$*. Definiramo funkciju* $\mathbb{P}_A : \mathcal{F} \to [0, 1]$ *kao:*

$$\mathbb{P}_A = \mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}, \quad B \in \mathcal{F}.$$

$\mathbb{P}_A$ *je vjerojatnost na* $\mathcal{F}$ *koju zovemo uvjetna vjerojatnost uz uvjet A. Broj* $\mathbb{P}(B|A)$ *zovemo vjerojatnost od B uz uvjet da se A dogodio.*

Definicija 1.3.8. *Neka je* $\mathcal{B}(\mathbb{R}^m)$ σ*-algebra generirana familijom svih otvorenih skupova na* $\mathbb{R}^m$*.* $\mathcal{B}(\mathbb{R}^m)$ *zovemo Borelova* σ*-algebra na* $\mathbb{R}^m$*, a njezine elemente Borelovi skupovi.*

Neka su dana dva izmjeriva prostora $(\Omega, \mathcal{F})$ i $(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m))$.

Definicija 1.3.9. *Slučajna varijabla* $(m = 1)$ *ili slučajni vektor* $(m > 1)$ *je izmjerivo preslikavanje* $X : (\Omega, \mathcal{F}) \to (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m))$*, odnosno preslikavanje za koje vrijedi*

$$X^{-1}(\langle -\infty, x]) \in \mathcal{F}, \quad \forall x \in \mathbb{R}^m.$$

Pritom je za $m \geq 2$, $x = (x_1, x_2, ..., x_m)$, $\langle -\infty, x] := \langle -\infty, x_1] \times ... \times \langle -\infty, x_m]$.

Definicija 1.3.10. *Neka su* $X_1, X_2, ..., X_n$ *slučajne varijable na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$*. Kažemo da su* $X_1, X_2, ..., X_n$ *nezavisne slučajne varijable ako za proizvoljne* $B_i \in \mathcal{B}(\mathbb{R}), i = 1, 2, ..., n$ *vrijedi:*

$$\mathbb{P}\Big(\cap_{i=1}^{n} (X_i \in B_i)\Big) = \sum_{i=1}^{n} \mathbb{P}(X_i \in B_i)$$

Definicija 1.3.11. *Neka je X slučajna varijabla na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$*. Funkcija distribucije od X je funkcija* $F_X = F : \mathbb{R} \to [0, 1]$ *definirana sa:*

$$F(x) = \mathbb{P}\{X \leq x\} = \mathbb{P}\{\omega \ : \ X(\omega) \leq x\}, \ x \in \mathbb{R}.$$

Definicija 1.3.12. *Neka je X slučajna varijabla na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$ *i neka je* F_X *pripadna funkcija distribucije. Kažemo da je X apsloutno neprekidna ili, kraće, neprekidna slučajna varijabla ako postoji nenegativna realna Borelova funkcija f na* $\mathbb{R}$ *takva da je:*

$$F_X(x) = \int_{-\infty}^{x} f(t) d\lambda(t), \quad x \in \mathbb{R}.$$

Funkciju f nazivamo funkcija gustoće vjerojatnosi od X, odnosno njezine funkcije distribucije F_X *ili, kraće, gustoća od X.*

Neka je X slučajna varijabla na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$. Tada vrijedi

$$X = X^+ - X^-,$$

pri čemu je $X^+ = \max\{X, 0\}$ i $X^- = \max\{-X, 0\}$.

Definicija 1.3.13. *Kažemo da slučajna varijabla X ima matematičko očekivanje ukoliko je barem jedan od integrala*

$$\mathbb{E}X^+ := \int_\Omega X^+ d\mathbb{P}, \quad \mathbb{E}X^- := \int_\Omega X^- d\mathbb{P}$$

konačan. Tada je matematičko olekivanje, u oznaci $\mathbb{E}X$, jednako

$$\mathbb{E}X := \mathbb{E}X^+ - \mathbb{E}X^- \in \overline{\mathbb{R}}$$

Definicija 1.3.14. *Pretpostavimo da za slučajnu varijablu X postoji $\mathbb{E}X$. Tada $\mathbb{E}[(X - \mathbb{E}X)^r]$ zovemo r-ti centralni moment od X.*

Definicija 1.3.15. *Drugi centralni moment od X zovemo varijanca od X i označavamo sa VarX ili σ_X^2. Pozitivan drugi korijen iz varijance zovemo standardna devijacija od X i označavamo sa σ_X.*

Primjeri neprekidnih slučajnih varijabli

Neka je $\alpha > 0$, $\beta > 0$ i $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$, $x > 0$ gama funkcija. Neprekidna slučajna varijabla ima **gamma distribuciju** s parametrima α i β ako joj je funkcija gustoće dana sa:

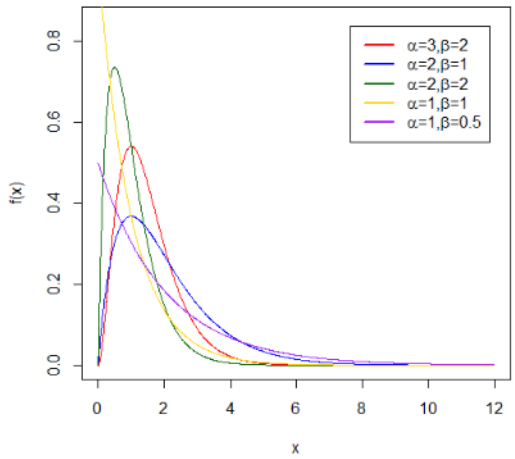
$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{\frac{-x}{\beta}} & , x > 0 \\ 0 & , x \leq 0 \end{cases} \tag{1.1}$$

Ako je $\alpha = 1$ i $\beta = \frac{1}{\lambda}$, kažemo da X ima **eksponencijalnu distribuciju** s parametrom λ. Funkcija gustoće eksponencijalne distribucije s parametrom λ dana je sa:

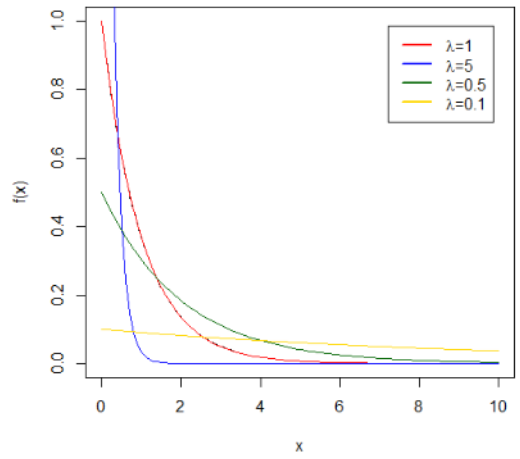
$$f(x) = \begin{cases} \lambda e^{-\lambda x} & , x > 0 \\ 0 & , x \leq 0 \end{cases} \tag{1.2}$$

Neka su $\mu, \beta \in \mathbb{R}, \beta > 0$. Neprekidna slučajna varijabla X ima **logističku distribuciju** s parametrima μ, β ako joj je funkcija gustoće dana sa:

$$f(x) = \frac{1}{\beta} \frac{e^{-\frac{x-\mu}{\beta}}}{\left(1 + e^{-\frac{x-\mu}{\beta}}\right)^2}, \ x \in \mathbb{R} \tag{1.3}$$



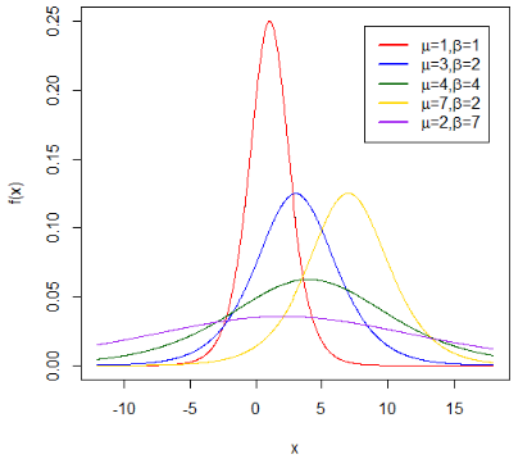
Slika 1.1: Funkcije gustoće gama distribucije za razne vrijednosti α i β



Slika 1.2: Funkcije gustoće eksponencijalne distribucije za razne vrijednosti λ

Neka su $p, q > 0$. Slučajna varijabla X ima **generaliziranu logističku distribuciju** ako joj je funkcija gustoće dana sa:

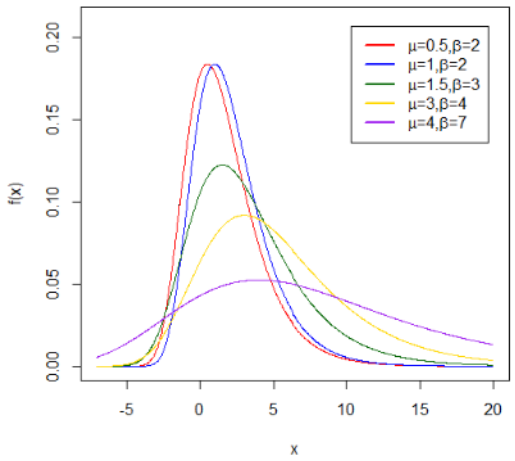
$$f(y) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} \frac{e^{-qy}}{(1+e^{-y})^{p+q}}, \; y \in \mathbb{R} \tag{1.4}$$



Slika 1.3: Funkcije gustoće logističke distribucije za razne vrijednosti μ i β

Neka su $\mu \in \mathbb{R}$ i $\beta > 0$. Neprekidna slučajna varijabla ima **Gumbelovu distribuciju** sa parametrima μ i β ako joj je funkcija gustoće dana sa:

$$f(x) = \frac{1}{\beta} e^{-\frac{x-\mu}{\beta} - e^{-\frac{x-\mu}{\beta}}}, \ x \in \mathbb{R} \tag{1.5}$$



Slika 1.4: Funkcije gustoće Gumbel distribucije za razne vrijednosti λ

Neka je $p > 0$. Slučajna varijabla ima **generaliziranu Gumbelovu distribuciju** s parametrom p ako joj je funkcija gustoće dana sa:

$$f(y) = \frac{1}{\Gamma(p)} e^{-py} e^{e^{-py}}, \ y \in \mathbb{R} \tag{1.6}$$

Korolar 1.3.16. *Neka su X_1 i X_2 nezavisne generalizirane Gumbel distribuirane slučajne varijable s parametrima p i q, respektivno. Tada slučajna varijabla $Y = X_1 - X_2$ ima generaliziranu logističku distribuciju s parametrima p i q.*

Slike 1.1, 1.2, 1.3 i 1.4 preuzete su iz [6].

Poglavlje 2

Stroj potpornih vektora

Stroj potpornih vektora (SVM) je algoritam za nadzirano učenje (eng. *supervised learning*) koji se može koristiti za probleme klasifikacije i regresije, no uglavnom se koristi za probleme klasifikacije. Nadzirano učenje je tehnika strojnog učenja koja iz skupa podataka za treniranje izračunava funkciju koja će u slučaju SVM algoritma generirati model za određivanje klase kojoj pripada primjer.

2.1 Linearna klasifikacija

Pretpostavimo da svaki primjer koji želimo klasificirati opisujemo sa n značajki. Tada se svaki primjer interpretira kao *vektor značajki* $x = (x_1, \ldots, x_n)^T$ kojemu je vrijednost svake koordinate određena vrijednošću pojedine značajke. Skup svih primjera čini n-dimenzionalni *ulazni prostor* ili *prostor primjera* $X \subseteq \mathbb{R}^n$. Pretpostavka je da su svi primjeri iz skupa X međsobno nezavisni i iz iste zajedničke distribucije. S obzirom da je SVM algoritam nadziranog učenja, to znači da je za svaki primjer poznata oznaka klase $y_i \in Y$ kojoj primjer x_i pripada. U slučaju binarne klasifikacije *prostor rezultata* obično se označava sa $Y = \{-1, 1\}$, a u slučaju m-klasne klasifikacije sa $Y = \{1, 2, \ldots, m\}$.

Skup od N primjera za učenje, koji kraće zovemo *skup za učenje* (eng. *training set*), možemo označiti na sljedeći način:

$$D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subseteq (X \times Y)^k.$$

Ukoliko svi primjeri imaju istu pridruženu oznaku kažemo da je skup D *trivijalan*. Kažemo da su podaci *linearno odvojivi* ako postoji hiperravnina koja točno razdvaja primjere za učenje, a u suprotnom kažemo da nisu odvojivi.

Hipoteza je funkcija $h : X \to Y$ koja klasificira primjere za učenje. Kažemo da je hipoteza h *konzistenta* s primjerom za učenje (x, y) ako i samo ako je $h(x) = y$. *Model* ili

prostor hipoteza je skup mogućih hipoteza $h \in H$. Dakle, učenje se svodi na pretraživanje prostora hipoteza i pronalaženje najbolje hipoteze $h \in H$ tj. one hipoteze koja najtočnije klasificira primjere za učenje.

Općeniti linearni model binarne klasifikacije zadan je kao $h(x) = w^T\phi(x) + b$ pri čemu je ϕ preslikavanje vektora značajki, *w vektor težina* modela i *b pomak*. Skup za učenje D sadržava N vektora $x^{(i)} \in \mathbb{R}^n$ i N pripadajućih oznaka $y^{(i)} \in Y = \{-1, 1\}$. Primjer x klasificirat ćemo u klasu C_1 ako je $h(x) \geq 0$, odnosno u klasu C_{-1} ako je $h(x) < 0$.

Ako pretpostavimo da je skup primjera X linearno odvojiv, tada postoji barem jedna funkcija odluke $h(x)$ koja savršeno klasificira skup za učenje (tj. postoji hipoteza koja je konzistentna sa skupom za učenje). U slučaju linearno odvojivih klasa postoji više konzistentnih hipoteza. Cilj je koristiti onu hipotezu koja najbolje klasificira neviđene primjere odnosno hipotezu sa najmanjom pogreškom generalizacije.

2.2 Margina linearnog klasifikatora

Kako bi se riješio problem odabira konzistentne hipoteze koja najbolje generalizira, model stroja potpornih vektora uvodi koncept *margine*. Margina se definira kao udaljenost između funkcije odluke i njoj najbližeg primjera. Pretpostavka stroja potpornih vektora je da će hipoteza sa najmanjom pogreškom generalizacije biti upravo ona s maksimalnom marginom.

Granica između klasa definirana je hiperravninom $h(x) = 0$. Konzistentna hipoteza zadovoljava sljedeće

$$\begin{cases} h(x^{(i)}) \geq 0, & \text{za sve } y^{(i)} = 1 \\ h(x^{(i)}) < 0, & \text{za sve } y^{(i)} = -1 \end{cases} \tag{2.1}$$

što možemo ekvivalentno zapisati kao

$$y^{(i)}h(x^{(i)}) \geq 0, \quad \forall\, i \in \{1, 2..., N\} \tag{2.2}$$

Udaljenost $d^{(i)}$ primjera $x^{(i)}$ od granice odluke računa se na sljedeći način:

$$d^{(i)} = \frac{|h(x^{(i)})|}{||w||} = \frac{|w^T\phi(x^{(i)}) + b|}{||w||} = \frac{y^{(i)}(w^T\phi(x^{(i)}) + b)}{||w||}. \tag{2.3}$$

Ukoliko parametre w i b skaliramo koeficijentom $k > 0$ udaljenost se neće promijeniti pa možemo pisati:

$$d^{(i)} = \frac{y^{(i)} \cdot (kw^T\phi(x^{(i)}) + kb)}{||kw||} = \frac{k \cdot y^{(i)} \cdot (w^T\phi(x^{(i)}) + b)}{|k| \cdot ||w||} = \frac{y^{(i)} \cdot (w^T\phi(x^{(i)}) + b)}{||w||}. \tag{2.4}$$

Dakle, možemo namjestiti w i b tako da za primjer koji je najbliži granici odluke vrijedi

$$y^{(i)} \cdot (w^T\phi(x^{(i)}) + b) = 1. \tag{2.5}$$

Tada će općenito vrijediti

$$y^{(i)} \cdot h(x^{(i)}) \geq 1, \quad \forall\, i \in \{1, 2..., N\}. \tag{2.6}$$

Margina je jednaka udaljenosti do najbližeg primjera, odnosno

$$min\{d^{(i)}\} = min\Big\{\frac{y^{(i)} \cdot (w^T \phi(x^{(i)}) + b)}{||w||}\Big\} = \frac{1}{||w||} min\Big\{y^{(i)} \cdot (w^T \phi(x^{(i)}) + b)\Big\}. \tag{2.7}$$

S obzirom da za najbliži primjer vrijedi $y^{(i)} \cdot \Big(w^T \phi(x^{(i)}) + b\Big) = 1$, slijedi da je margina $min\{d^{(i)}\} = \frac{1}{||w||}$. Prema pretpostavci, hipoteza koja najbolje generalizira ima maksimalnu marginu pa je potrebno maksimizirati sljedeći izraz

$$arg \max_{w,b} \{min\{d^{(i)}\} = arg \max_{w,b} \frac{1}{||w||}, \tag{2.8}$$

a da pritom vrijede sljedeća ograničenja

$$y^{(i)} \cdot (w^T \phi(x^{(i)}) + b) \geq 1, \quad \forall\, i \in \{1, 2..., N\}. \tag{2.9}$$

Radi lakšeg rješavanja problem maksimizacije zapisujemo kao problem minimizacije uz iste uvjete

$$arg \max_{w,b} \frac{1}{||w||} \iff arg \min_{w,b} \frac{1}{2}||w||^2. \tag{2.10}$$

Za rješavanje problema optimizacije uz ograničenja koristimo metodu Lagrangeovih multiplikatora. Lagrangeova funkcija tada glasi:

$$L(w, b; \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i \cdot \Big[y^{(i)} \cdot (w^T \phi(x^{(i)}) + b) - 1\Big] \tag{2.11}$$

gdje je $\alpha = (\alpha_1, ..., \alpha_N)$ vektor Lagrangeovih multiplikatora. Lagrangeovu funkciju minimiziramo uz sljedeće uvjete, za svaki $i \in \{1, .., N\}$:

$$\begin{cases} \alpha_i \geq 0 \\ \alpha_i \cdot [y^{(i)} \cdot (w^T \phi(x^{(i)}) + b) - 1] = 0 \end{cases} \tag{2.12}$$

Nakon što funkciju $L(w, b; \alpha)$ deriviramo po parametrima w i b te izjednačimo s 0 dobivamo sljedeće uvjete:

$$\begin{cases} w = \sum_{i=1}^{N} \alpha_i \cdot y^{(i)} \cdot \phi(x^{(i)}) \\ \sum_{i=1}^{N} \alpha_i \cdot y^{(i)} = 0 \end{cases} \tag{2.13}$$

Uvrštavanjem gornjih uvjeta u funkciju $L(w, b; \alpha)$ dobivamo dualnu Lagrangeovu funkciju

$$\widetilde{L}(w, b; \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(x^{(i)}) \phi(x^{(j)}) \tag{2.14}$$

uz uvjete

$$\begin{cases} \alpha_i \geq 0, & \forall\ i \in \{1, ..., N\} \\ \sum_{i=1}^{N} \alpha_i \cdot y^{(i)} = 0 \end{cases} \tag{2.15}$$

Dualna formulacija omogućava traženje rješenja polaznog problema izračunom skalarnog produkta što je jednostavan postupak. Također, iz dualnog problema možemo izračunati Lagrangeove multiplikatore α_i. Nakon što odredimo Lagrangeove multiplikatore možemo odrediti vektor težine w pa uvrštavanjem u funkciju hipoteze slijedi

$$h(x) = \sum_{i=1}^{N} \alpha_i y^{(i)} \phi(x^{(i)})^T \phi(x^{(i)}) + b. \tag{2.16}$$

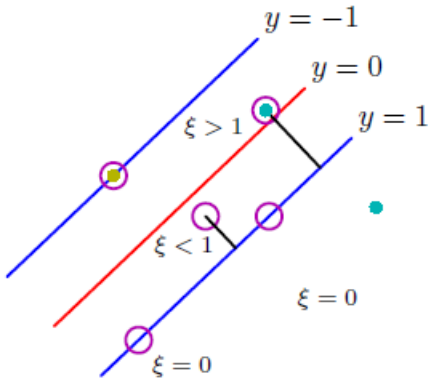
Uočimo, i dalje vrijedi uvjet $\alpha_i \cdot [y^{(i)} h(x^{(i)}) - 1] = 0$. Dakle, možemo zaključiti da vrijedi ili $\alpha_i = 0$ ili $y^{(i)} h(x^{(i)}) - 1 = 0$. Bilo koji vektor $x^{(i)}$ za koji vrijedi $\alpha_i = 0$ ne sudjeluje u sumi, pa neće biti relevantan za klasifikaciju budućih primjera x. Preostali vektori nazivaju se *potpornim vektorima*, a pošto zadovoljavaju uvjet $y^{(i)} h(x^{(i)}) = 1$ zaključujemo da se radi o vektorima koji leže na margini oko funkcije odluke.

2.3 Meka margina

Ranije smo pretpostavljali da je skup za učenje linearno odvojiv jer je u tom slučaju moguća točna klasifikacija svih primjera iz skupa za učenje. Prethodno definirani problem podrazumijeva pogrešku 0 za ispravno klasificirane primjere, dok za neispravno klasificirane primjere poprima grešku koja teži u beskonačnost. Način za rješavanje ovakvog problema je dozvoljavanje da neki primjeri budu krivo klasificirani, ali uz penalizaciju proporcionalnu udaljenosti od margine. Za svaki primjer za učenje uvodimo novu varijablu $\xi_i \geq 0$ koja će govoriti koliko je pojedini primjer prešao na krivu stranu. Za primjere koji se nalaze na margini ili s prave strane margine vrijedi $\xi_i = 0$, a za ostale primjere je $\xi_i = |y^{(i)} - h(x^{(i)})|$. Dakle, ukoliko je $\xi_i > 1$ prisutna je pogrešna klasifikacija.

Ograničenje sada izgleda ovako

$$y^{(i)} h(x^{(i)}) \geq 1 - \xi_i\,, \quad \xi_i \geq 0, \quad \forall\ i \in \{1, 2..., N\}. \tag{2.17}$$



Slika 2.1: Stroj potpornih vektora - meka margina, preuzeto iz [9]

S obzirom da želimo penalizirati ulaz u marginu, minimiziramo sljedeći izraz

$$C \sum_{i=1}^{N} \xi_i + \frac{1}{2}\|w\|^2, \tag{2.18}$$

pri čemu je $C > 0$ parametar koji kontrolira omjer između penalizacije ulaza u marginu i problema minimizacije margine. Male vrijednosti parametra C podrazumijevaju malu penalizaciju za ulaz u marginu tj. jednostavan model, dok za $C \to \infty$ imamo SVM model za odvojive klase (tvrda margina).
Primarna Lagrangeova funkcija je sljedećeg oblika

$$L(w, b; \alpha, \beta, \xi) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left[y^{(i)}(w^T \phi(x^{(i)}) + b) - 1 + \xi_i \right] - \sum_{i=1}^{N} \beta_i \xi_i, \tag{2.19}$$

gdje su $\alpha = (\alpha_1, ..., \alpha_N)$ i $\beta = (\beta_1, ..., \beta_N)$ vektori Lagrangeovih multiplikatora. Lagrangeovu primarnu funkciju deriviramo po w, b i ξ_i te izjednačimo s 0 i dobivene uvjete uvrstimo u $L(w, b; \alpha, \beta, \xi)$. Time dobivamo dualnu Lagrangeovu funkciju sljedećeg oblika

$$\widetilde{L}(w, b; \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(x^{(i)}) \phi(x^{(j)}) \tag{2.20}$$

uz uvjete

$$\begin{cases} 0 \le \alpha_i \le C & , \forall\, i \in \{1, ..., N\} \\ \sum_{i=1}^{N} \alpha_i \cdot y^{(i)} = 0 \end{cases} \tag{2.21}$$

Poglavlje 3

Izrada modela za klasifikaciju

Zadatak klasifikacije je pridružiti dokument jednoj od postojećih klasa. U ovom radu ćemo klasificirati biološke nizove, točnije proteine. Proteini su nizovi aminokiselina bez separatora pri čemu je svaka aminokiselina predstavljena jednim od sljedećih slova A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V. Po evolucijskom podrijetlu proteini se grupiraju u proteinske familije. Znamo da su proteini unutar iste familije slične strukture i sadrže slične nizove aminokiselina. Promatrat ćemo proteine iz 7 familija: *UcrQ, Upf2, Glicerol 3-fosfat, Kromatin, XendoU, Plasmid, Peptidase S8*. Uočimo, u terminima klasifikacije dokumenata, svaki protein je jedan dokument dok familija proteina čini klasu.

Skup za učenje sadržavat će po 300 proteina iz svake klase uz odgovarajuću oznaku iz skupa $\{1, 2, ..., 7\}$, ovisno o tome iz koje familije dolaze. Za testiranje ćemo koristiti skup sačinjen od po 100 proteina iz svake od 7 familija.

Proteine koji dolaze iz iste proteinske familije, a sadržani su u skupu za učenje nazivat ćemo *kolekcijom*. Svaki protein unutar kolekcije zapisan je u jednom retku. Uočimo, proteini unutar iste kolekcije ne moraju biti jednake duljine.

Semantičko indeksiranje je postupak opisan u radu [3], a temelji se na indeksiranju teksta dodjeljivanjem odgovarajućih težina pojedinim pojmovima. Pokazano je da takav jednostavan pristup dovoljno dobar. Generalizirano semantičko indeksiranje je postupak kojim ćemo određivati značajke svake klase, a usko slijedi izvor [4].

3.1 Pronalaženje motiva

Ključno je identificirati značajne podnizove aminokiselina, točnije, one podnizove koji se često pojavljuju unutar proteina, ali uz dozvoljene varijacije. Dozvoljavat ćemo varijacije koje se evolucijski mogu povezati s nekom aminokiselinom. Primjerice, ukoliko dođe do promjene aminokiseline *Alanin (A)*, najčešće varijacije su *Valin (V)* ili *Fenilanin (F)*. Ako je dan značajan podniz aminokiselina $AGPPAAKHYM$, njegova moguća varijacija je

podniz *AGPPAVKHYM*. Značajan podniz aminokiselina sa svim njegovim varijacijama odnosno specifičnim supstitucijama nazivat ćemo *motiv*.

Pretpostavimo da je zadan podniz aminokiselina $u = (u_1, ..., u_n)$ kojeg ćemo nazivati *upit*. Želimo pronaći njemu najsličniji podniz aminokiselina iste duljine n u svakom proteinu tj. u svakom retku dane kolekcije. Neka je dan protein duljine m, pri čemu je $n << m$. Opisat ćemo postupak traženja podniza duljine n koji je sličan početnom upitu u. Kako bismo početni upit usporedili sa svakim podnizom duljine n unutar zadanog proteina koristimo metodu klizećeg prozora (eng. *sliding window*). Početni upit u uspoređujemo sa svakim odsječkom proteina duljine n počevši od mjesta 1 pa do mjesta $(m-n+1)$ u proteinu.

Postupak uspoređivajanja ilustriramo primjerom kada je zadan upit $u = MNQVAKDALE$ duljine 10 i dio proteina duljine m iz neke kolekcije.

```
MNQVLKDALEDN......ISSTNTDVWFL
MNQVAKDALE
MNQVLKDALEDN......ISSTNTDVWFL
 MNQVAKDALE
MNQVLKDALEDN......ISSTNTDVWFL
  MNQVAKDALE
        ......
MNQVLKDALEDN......ISSTNTDVWFL
                  MNQVAKDALE
MNQVLKDALEDN......ISSTNTDVWFL
                   MNQVAKDALE
```

Opisani postupak analogno provodimo u svakom retku kolekcije. Prilikom prolaska upitom kroz kolekciju računamo i ocjenu sličnosti svakog podniza i upita. Što je ocjena sličnosti veća to je podniz sličniji upitu. Kako bismo definirali ocjenu sličnosti potrebno je uvesti još nekoliko pojmova.

Pretpostavljamo da se svaka aminokiselina u proteinu pojavljuje neovisno o prethodnoj. Računanjem relativnih frekvencija aminokiselina u proteinima iz više različitih organizama dobiven je vektor vjerojatnosti pojavljivanja svake aminokiseline

q=(0.078, 0.051, 0.043, 0.053, 0.019, 0.043, 0.063, 0.072, 0.023, 0.053, 0.091, 0.059, 0.022, 0.039, 0.052, 0.068, 0.059, 0.014, 0.032, 0.066).

Neka je zadan upit u duljine n i neka je P profil pridružen potencijalnom motivu, konstruiran analognim postupkom kao u 3.3. Neka je $y^{(k)}$ podniz duljine n na k-toj poziciji u proteinu duljine m, za $k \in \{1, 2, ..., m - n + 1\}$. Ocjenu sličnosti ili *score* definiramo sljedećom formulom

$$s_k = \sum_{i=1}^{n} log \frac{\mathbb{P}(y^{(k)}|P)}{\mathbb{P}(y^{(k)}|q)}. \tag{3.1}$$

Vrijednost s_k računamo na svakoj mogućoj poziciji unutar podniza te uzimamo maksimum. Kako bismo znali koji su podnizovi dovoljno slični početnom upitu, definirat ćemo prag te ćemo dovoljno sličnima smatrati one podnizove čija je ocjena sličnosti veća ili jednaka od praga. S obzirom da su maksimalne ocjene sličnosti svakog retka logistički distribuirane, prag definiramo na sljedeći način

$$prag = \mu + skala \cdot \beta,$$

pri čemu je μ očekivanje neprekidne slučajne varijable s logističkom distribucijom, β parametar logističke distribucije, a *skala* proizvoljan pozitivan broj koji određuje razinu sličnosti odabranih podnizova. Preniska *skala* dala bi velik broj podnizova s malom međusobnom sličnosti, stoga u ovom radu koristimo $skala = 4$. Razlog zbog kojeg su parametri za izračunavanje praga upravo iz logističke distribucije objasit ćemo na kraju odjeljka.

Opisanim postupkom za svaki upit dobit ćemo skup koji će se sastojati od podnizova koji su dovoljno slični polaznom upitu, odnosno kojima je ocjena sličnosti veća ili jednaka od zadanog praga. Dobiveni skup podnizova sličnih polaznom upitu nazivat ćemo *odgovor*. U radu koristimo upite duljine 10, a generiramo ih iz proteina unutar kolekcije uzimajući redom odsječke zadane duljine. Grafički to izgleda ovako

$$\underbrace{MGGASAKTFM}_{\text{upit1}}GWWGSIGSPKQKGV\ldots$$
$$M\underbrace{GGASAKTFMG}_{\text{upit2}}WWGSIGSPKQKGV\ldots$$

Ponavljamo opisani postupak dok ne potrošimo sve upite unutar kolekcije. Time dobivamo sve moguće odgovore koji se odnose na zadanu kolekciju.

Potencijalno značajnim odgovorima smatrat ćemo one koji sadrže više od 15 podnizova. Takve odgovore u daljnjem tekstu nazivat ćemo *motivi*. Navodimo jedan motiv sa njegovim specifičnim supstitucijama.

```
MGGASAKTFM
AGPPAAKHYM
MGSPSAKTYL
GGAAGGKTYL
GGASGPKTYM
MGGAAAKTYM
MGGAGGHGYM
MSPPGAKAYM
AGAPHPHTYM
GGPTGGKAYM
GGAAGGKTYM
MGGPHAKTYM
```

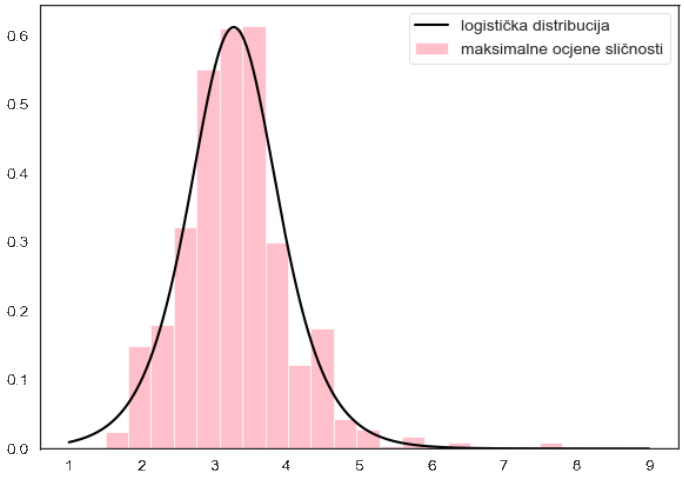
```
GGAPHPKAYM
MGGAKGKSYM
MAGAGPKAYM
MGGPTAKTFL
MGGPHAKTFM
GGAPSGKTYL
MGGPSAKTYM
```

U gornjem primjeru možemo uočiti kako se na prvom mjestu pojavljuju varijacije *Alanin (A), Glicin (G) i Metionin (M)* dok je drugo mjesto očuvano jer se gotovo uvijek pojavljuje *Glicin (G)*.

Primjenom opisane metode pronađeno je u prosjeku 26706 potencijalnih značajki tj. motiva po klasi.

Određivanje praga sličnosti

Prisjetimo se, score-ove računamo prema formuli 3.1. Po nizu proteina scoreovi su distribuirani kao degenerirani χ^2 što je distribucija sa eksponencijalnim repom. Zanimaju nas maksimalni scoreovi unutar proteina, a oni se nalaze u repu distribucije. Poznato je da maksimumi nezavisnih jednako distribuiranih eksponencijalnih slučajnih varijabli imaju Gumbel distribuciju. Proteini nisu jednakih duljina pa ne možemo pretpostaviti da maksimalne ocjene sličnosti prate Gumbelovu distribuciju. Histogrami duljina proteina pokazuju da duljine proteina prate Gumbelovu distribuciju.



Slika 3.1: Histogram maksimalnih ocjena sličnosti i funkcija gustoće logističke distribucije

Prema korolaru 1.3.16 razlika dviju slučajnih varijabli s Gumbelovom distribucijom je slučajna varijabla s logističkom distribucijom. Ako promotrimo sliku 3.1, uočavamo da maksimalne ocjene sličnosti prate logističku distribuciju. Stoga za računanje praga u 3.1 uzimamo očekivanje neprekidne slučajne varijable s logističkom distribucijom μ te parametar logističke distribucije β. Više o distribuciji maksimalnih ocjena sličnosti u [5] i [6].

3.2 Produživanje motiva

Ako dva motiva imaju iste specifične supstitucije, možemo reći da se radi o istom motivu. Dakle, opravdano je više motiva koji sadrže neke zajedničke podnizove spojiti u jedan veći motiv prepisujući zajedničke podnizove samo jednom. Posljedica ovakvog pristupa bit će smanjenje broja motiva unutar pojedine klase.

Mi ćemo spajati one motive kojima je prvi podniz jednak. Uočimo, postoje i bolji načini za odabir motiva koji ulaze u spajanje, no radi jednostavnosti i brzine algoritma odlučili smo se za ovakav pristup. Primjerice, neka su dana sljedeća dva motiva

```
MGGASAKTFM      MGGASAKTFM
AGPPAAKHYM      AGPPAAKHYM
MGSPSAKTYL      MGSPSAKTYL
GGAAGGKTYL      MGGAAAKTYM
GGASGPKTYM      SEMPNGKHYM
MGGAAAKTYM      MSPPGAKAYM
MGGAGGHGYM      SGMPTGKSYL
MSPPGAKAYM      SGMPTGKKYM
AGAPHPHTYM      GGAAGGKTYM
GGPTGGKAYM      SGMPTGKKYM
GGAAGGKTYM      MGGPHAKTYM
MGGPHAKTYM      GGAPHPKAYM
GGAPHPKAYM      MGGAKGKSYM
MGGAKGKSYM      GGMPTGKSYM
MAGAGPKAYM      MGGPTAKTFL
MGGPTAKTFL      MGGPHAKTFM
MGGPHAKTFM      SGMPTGNKYM
GGAPSGKTYL      SGMPSGKTYM
MGGPSAKTYM      GGAPSGKTYL
MGGPEAKTYM      MGGPSAKTYM
                MGGPEAKTYM
```

S obzirom da se prvi podniz podudara u oba motiva spojit ćemo ih u jedan motiv, zapisujući na kraj prvog motiva samo one podnizove iz drugog motiva koji se ranije nisu pojavili.

```
MGGASAKTFM
AGPPAAKHYM
MGSPSAKTYL
GGAAGGKTYL
GGASGPKTYM
MGGAAAKTYM
MGGAGGHGYM
MSPPGAKAYM
AGAPHPHTYM
GGPTGGKAYM
GGAAGGKTYM
MGGPHAKTYM
GGAPHPKAYM
MGGAKGKSYM
MAGAGPKAYM
MGGPTAKTFL
MGGPHAKTFM
GGAPSGKTYL
MGGPSAKTYM
MGGPEAKTYM
SEMPNGKHYM
SGMPTGKSYL
SGMPTGKKYM
GGMPTGKSYM
SGMPTGKKYM
SGMPSGKTYM
```

Primjenom opisane metode ostaje u prosjeku 2760 potencijalnih značajki tj. motiva po klasi, što je približno 10% od inicijalno pronađenog prosjeka.

3.3 Izrada profila

Cilj nam je značajke svake klase predstaviti vektorima kako bismo mogli primijeniti algoritme linearne klasifikacije. Prvi korak je motive opisati pomoću niza vjerojatnosnih distribucija koji ćemo nazivati *profil*. Vjerojatnosnu distribuciju ćemo računati za svaki stupac unutar motiva, pa kako smo motive gradili od podnizova duljine 10 to znači da ćemo svaki motiv opisati sa 10 vjerojatnosnih vektora.

Općenito, neka je dan motiv koji je sastavljen od m podnizova duljine l. Želimo izračunati relativnu frekvenciju svih aminokiselina unutar stupca. Prisjetimo se, amino-

kiseline su označene sa 20 slova. Stoga ćemo profil definirati pomoću l vektora, po jedan za svaki stupac unutar motiva, na sljedeći način

$$f_i = (f_{i1}, f_{i2}, ..., f_{i20}), \quad i = 1, 2, ..., l,$$

gdje je f_{ij} vjerojatnost da se u i-tom stupcu zadanog motiva pojavi j-to slovo. Moguće je da vjerojatnost pojave određenog slova u nekom stupcu bude 0. Kako bismo to izbjegli, napravimo sljdeće

$$\widetilde{f_{ij}} = \frac{f_{ij} + \varepsilon}{1 + 20\varepsilon} \tag{3.2}$$

pri čemu je ε pozitivna vrijednost bliska nuli.

Konačno, $\{\widetilde{f_{ij}} : i = 1, ..l, j = 1, ..., 20\}$ čini profil zadanog motiva.

3.4 Proširivanje motiva

Prethodno smo promotrili kako su aminokiseline distribuirane po stupcima unutar motiva. Ako promotrimo raspored aminokiselina po recima unutar motiva, možemo uočiti da postoje motivi čiji se podnizovi međusobno nastavljaju. Ova pojava je direktna posljedica načina na koji smo odabirali podnizove koji sačinjavaju motiv (*klizeći prozor*).
Primjerice, sljedeća dva podniza imaju preklapanje duljine 9.

GGASAKTFMG
GASAKTFMGW

U slučaju preklapanja duljine k ideja je *proširiti* prvi podniz za $n - k$ mjesta sa aminokiselinama koje nisu sadržane u preklapanju. Gornji primjer bi nakon proširivanja izgledao ovako

GGASAKTFMGW.

Kada motivi nose istu informaciju, željeli bismo preklapanje zapisati samo jednom. Primjerice, uočimo da se prvi podnizovi sljedeća dva motiva preklapaju.

```
GGASAKTFMG      GASAKTFMGW
GSPSAKTYLG      SPSAKTYLGW
GAAGGKTYLG      AAGGKTYLGW
GASGPKTYMG      ASGPKTYMGW
GGAAAKTYMG      APSGKTYLGW
GAPHPHTYMG      GAAAKTYMGW
GMPTGKKYMG      PPGAKAYMGW
   .....           .....
```

Upravo takve motive u kojima se prvi podnizovi preklapaju predstavit ćemo pomoću jednog profila i u budućoj analizi smatrati jednim motivom. Uočimo, kod preklapanja duljine k profili će sadržavati $2n-k$ distribucija. Važno je odrediti kada ovakav pristup ima smisla, odnosno kada su motivi koje želimo interpretirati kao jedan usitinu dovoljno slični. Koristit ćemo relativnu entropiju odnosno mjeru sličnosti distribucija stupaca kod kojih su uočena preklapanja. Prisjetimo se, distribucija aminokiselina unutar stupaca motiva zapisana je u profilima kako je opisano u 3.3.

Neka su p i q distribucije stupaca u kojima je uočeno preklapanje. Relativna entropija distribucije p u odnosu na distribuciju q je

$$H(p||q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}. \tag{3.3}$$

Radi simetričnosti, prilikom računanja sličnosti distribucija stupaca, koristit ćemo sljedeću formulu

$$Re(p,q) = H(p||q) + H(q||p). \tag{3.4}$$

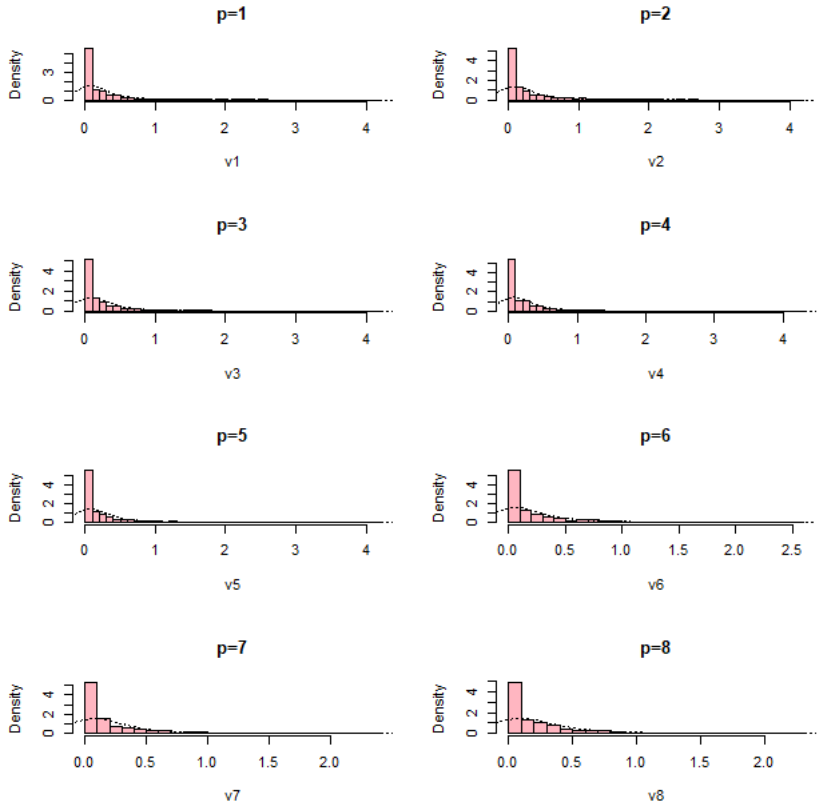
Ukupna relativna entropija dvaju motiva je suma relativnih entropija svih pozicija na kojima je uočeno preklapanje.

Željeli bismo da je ukupna relativna entropija čim bliže nuli jer su tada uspoređivani motivi sličniji. Kako bismo znali odrediti granicu sličnosti promotit ćemo histograme relativnih entropija za sve dozvoljene pomake. U našem slučaju polazna duljina podnizova koji grade motive bila je 10, pa ćemo dozvoliti maksimalno 8 pomaka. Napomenimo, proširivanje za k mjesta dozvoljeno je samo ako su prethodno zadovoljeni uvjeti za $k-1$ proširivanje, pri čemu je $k \in \{1, 2, ..., 8\}$.

Na slici 3.2 uočavamo da bi približno četvrtina motiva imala veliku grešku pri ovakvom spajanju. Kako bismo izbjegli spajanja s velikim greškama, za granicu relativne entropije uzet ćemo 75%-kvantil.

Konačno, ako za neka dva motiva postoji preklapanje duljine k te je suma relativne entropije među njima manja od dozvoljene granice, navedene motive predstavljat ćemo jednim profilom koji će sadržavati $2n-k$ distribucija. U našem slučaju to znači da će broj distribucija u novim profilima biti varijabilan, $l \in \{10, 11, ..., 18\}$.

Opisat ćemo na koji način nastaju novi profili. Neka su dana dva motiva sa profilima $P = (p_1, ..., p_{10})$ i $Q = (q_1, ..., q_{10})$ te neka je preklapanje među njima duljine $k = 5$ takvo da je zadovoljena granica relativne entropije. Preciznije, distribucije $p_i, i = 6, 7, 8, 9, 10$ i $q_j, j = 1, 2, 3, 4, 5$ imaju sumu relativnih entropija manju od zadane granice. Tada ćemo ta dva motiva reprezentirati novim profilom PQ i to na načina da u profil P prepišemo $n-k = 5$ zadnjih distribucija iz profila Q. Novi profil je tada $PQ = (p_1, ..., p_{10}, q_6, ..., q_{10})$ duljine $2n-k = 15$.



Slika 3.2: Sume relativnih entropija za sve dozvoljene pomake

Primjenom opisane metode ostaje u prosjeku 1329 potencijalnih značajki tj. motiva po klasi, što je približno 4.98% od inicijalno pronađenih motiva. Tako dobivene motive iz svake klase smatrat ćemo njezinim značajkama.

3.5 Izrada modela

Kako bismo mogli primijeniti metode linearne klasifikacije potrebno je značajke svake klase opisati pomoću vektora. Stoga ćemo izgraditi model koji će za zadani protein vektorski opisati značajke zadanih klasa. Važno je uočiti da prethodno opisanim postupcima traženja motiva ne dobivamo jednak broj motiva unutar svake klase. Kako naš model ne bi preferirao klasu za koju je pronađeno najviše motiva, uzet ćemo jednak broj motiva za opisivanje svake klase unutar modela. Postavlja se pitanje kako odabrati N motiva koji

najbolje opisuju klasu iz koje dolaze. Za odabir takvih motiva koristit ćemo entropiju. Entropija je mjera neuređenosti sustava odnosno mjera raspršenosti. Formula za računanje entropije sustava kojemu je distribucija p glasi:

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i. \tag{3.5}$$

Ako je $p = (0, ..., 0, 1, 0, ..., 0)$, uvijek znamo ishod sustava, odnosno možemo reći da je sustav uređen. Uočimo, u navedenom slučaju entropija je 0. Stoga ćemo birati one motive čije distribucije imaju najmanju entropiju. Nakon što odredimo entropiju svakog motiva odnosno distribucije koja ga predstavlja, odabrat ćemo po 100 motiva sa najmanjom entropijom iz svake klase.

Prisjetimo se, profil koji su opisani u 3.4 sadržavaju vjerojatnosne vektore

$$\widetilde{f_i} = (\widetilde{f_{i1}}, ..., \widetilde{f_{i20}}), \quad i = 1, 2, ..l, \quad l \in \{10, ..., 18\}$$

što nam govori koja je vjerojatnost da se $j-to$ slovo tj. aminokiselina pojavi u $i-tom$ stupcu odgovarajućeg motiva. Ranije smo definirali vektor q koji sadrži vjerojatnosti pojavljivanja svake od mogućih 20 aminokiselina u cijeloj kolekciji (vidi 3.1). Zanima nas omjer tih dviju vjerojatnosti. Istovremeno želimo minimizirati pogreške prve i druge vrste pa ćemo računati *log-likelihood ratio* navedenih vrijednosti.

$$h_{ij} = \log \frac{\widetilde{f_{ij}}}{q_j}, \quad i = 1, ..., l, \quad j = 1, ..., 20 \tag{3.6}$$

Vektori $\{h_i = (h_{i1}, ..., h_{i20};\ i = 1, ..., l\}$ govore koliko je prisutnost ili odsutnost slova u odabranom motivu povezana sa prisutnošću ili odsutnošću slova u cijeloj kolekciji. Model izrađujemo prema 100 odabranih motiva iz kolekcije, pa je model neke familije tada

$$M = \{h^{(k)},\ k = 1, ..., 100\}$$

pri čemu je $h^{(k)} = \{h_i;\ i = 1, .., l_k\}$, za $l_k \in \{10, ..., 18\}$.
U našem slučaju imat ćemo 7 modela izrađenih na opisani način, svaki izrađen pomoću motiva iz jedne od zadanih proteinskih familija.

Sada za svaki protein možemo po modelu izraditi vektor duljine 100 koji će sadržavati informacije o odnosu zadanog proteina i familije koju model opisuje. Koristeći svih 7 modela, možemo opisati odnos svakog zadanog proteina sa familijama koje proučavamo, a sve pomoću jednog vektora duljine 700.

Opisani postupak ponovit ćemo tako da za svaku kolekciju odaberemo 150 motiva sa najmanjom entropijom, a sve kako bismo mogli vidjeti koliko broj motiva koji ulaze u model mijenja rezultate klasifikacije u konačnici. Uočimo, tada će odnos svakog zadanog proteina i familija koje proučavamo biti opisan vektorom duljine 1050.

Poglavlje 4

Rezultati

Za testiranje dobivenih modela koristit ćemo po 100 proteina iz svake proteinske familije, ukupno njih 700. Usporedit ćemo rezultate klasifikacije max-normom sa rezultatima klasifikacije dobivene pomoću algoritma stroja potpornih vektora (SVM). Razmotrit ćemo kakav utjecaj na rezultate klasifikacije ima broj motiva koje koristimo za izradu modela. Prisjetimo se, za izradu modela koristimo 100 odnosno 150 motiva iz svake od 7 proteinskih familija, što je ukupno 700 odnosno 1050 motiva. Također, usporedit ćemo rezultate pristupa opisanog u ovom radu sa rezultatima pristupa iz rada [7]. Navedeni pristupi razlikuju se u pripremi podataka koji se koriste u izradi modela za klasifikaciju. Dok mi za izradu modela koristimo motive koji sadrže podnizove koji se često pojavljuju unutar proteina sa dozvoljenim varijacijama, u radu [7] opisana je izrada modela pomoću rječnika sastavljenog od svih mogućih 5-grama, odnosno svih nizova od 5 uzastopnih aminokiselina, koji se pojavljuju u zadanim proteinskim familijama.

4.1 Mjere uspješnosti

Za mjerenje uspješnosti klasifikacije koristit ćemo sljedeće mjere:

- osjetljivost (eng. recall, sensitivity) - sposobnost klasifikatora da pronađe pozitivne primjere, odnosno da za neki primjer odredi klasu kojoj zbilja pripada

$$osjetljivost = \frac{broj\ stvarno\ pozitivnih}{broj\ stvarno\ pozitivnih + broj\ lazno\ negativnih}$$

- pozitivna prediktivna vrijednost (eng. precision) - sposobnost klasifikatora da ne označava pozitivnima one primjere koji to nisu, odnosno da u neku klasu ne svrstava primjere koji joj ne pripadaju

$$pozitivna\ prediktivna\ vrijednost = \frac{broj\ stvarno\ pozitivnih}{broj\ stvarno\ pozitivnih + broj\ lazno\ pozitivnih}$$

4.2 Rezultati klasifikacije

Prisjetimo se, opisali smo način na koji izrađujemo model koji opisuje odnos određene familije i nekog zadanog proteina. Pri izradi modela koristili smo 100, odnosno 150, motiva koji najbolje opisuju familiju iz koje dolaze. Dakle, za svaki zadani protein iz skupa za testiranje možemo po modelu dobiti vektor duljine 100, odnosno 150, koji sadržava informacije o odnosu zadanog proteina i familije koju odabrani model opisuje.

Max-norma

Klasifikacija max-normom podrazumijeva da za svaki vektor dobiven pomoću ranije opisanih modela odredimo max-normu, a zatim primjer svrstamo u onu klasu koju opisuje model pomoću kojeg je dobiven vektor s najvećom max-normom. Primjerice, ako je vektor max-normi po modelima za neki protein iz skupa za testiranje [30.07, 0.99, 1.11, 0.47, 0.0, 0.0, 3.63], tada se zadani primjer klasificira u familiju koju opisuje prvi model.

Promotrimo rezultate iz tablice 4.1 za familiju UcrQ u slučaju kada smo za izradu modela koristili 100 motiva. Osjetljivost iznosi 0.99 što znači da je 99 od 100 testnih primjera iz familije UcrQ ispravno klasificirano. Pozitivna prediktivna vrijednost iznosi 1.00 što znači da ni jedan primjer iz preostalih familija nije klasificiran u familiju UcrQ.

	familija	osjetljivost	PPV	točnost
100 motiva	UcrQ	0.99	1.00	**97.14%**
	Upf2	0.93	0.98	
	Glicerol 3-fosfat	1.00	0.97	
	Kromatin	0.94	0.95	
	XendoU	0.96	1.00	
	Plasmid	0.99	0.93	
	Peptidase S8	0.99	0.98	
150 motiva	UcrQ	0.99	1.00	**97.57%**
	Upf2	0.92	1.00	
	Glicerol 3-fosfat	1.00	0.99	
	Kromatin	0.95	0.93	
	XendoU	0.98	1.00	
	Plasmid	1.00	0.93	
	Peptidase S8	0.99	0.99	

Tablica 4.1: Rezultati klasifikacije max-normom

Linearni SVM algoritam

Prisjetimo se, kada je skup za učenje linearno odvojiv moguća je točna klasifikacija svih primjera. Kako to često nije slučaj, dozvoljavamo da neki primjeri budu krivo klasificirani, ali uz penalizaciju proporcionalnu udaljenosti margine, kao što je objašnjeno u 2.3. Omjer između penalizacije ulaza u marginu i problema minimizacije margine kontroliramo parametrom $C > 0$. Parametrom C određujemo koliko je algoritmu stroja potpornih vektora dozvoljeno da krivo klasificira primjere iz skupa za učenje. Mala vrijednost parametra dopušta klasifikatoru da krivo klasificira neke primjere, dok velika vrijednost parametra ne dopušta krivu klasifikaciju. Odabirom prevelike vrijednosti parametra C možemo dobiti previše precizan model, što može rezultirati lošom klasifikacijom nepoznatih tj. testnih primjera.

Kako je vrijednost parametra C jedino što možemo mijenjati, promotrit ćemo rezultate točnosti klasifikacije algoritmom stroja potpornih vektora za različite vrijednosti navedenog parametra (vrijednosti su prikazane u tablici 4.2).

Uočimo, za vrijednost parametra $C = 0.01$ dopuštamo algoritmu stroja potpornih vektora da neke primjere krivo klasificira. Drugim riječima, dozvoljavamo da margina između odvajajuće hiperravnine i potpornog vektora bude veća, pritom zanemarujući neke primjere iz skupa za učenje koji se nalaze unutar margine ili sa druge strane hiperravnine. Promotrimo li tablicu 4.2, uočavamo da ovakva modifikacija rezultira sa približno 98% točno klasificiranih primjera iz skupa za testiranje. Povećanjem parametra C ne postižemo bolje rezulate u smislu točnosti klasifikacije primjera iz skupa za testiranje.

Promotrimo tablicu 4.3 i rezultate za familiju UcrQ u slučaju kada smo za izradu modela koristili 100 motiva. Osjetljivost iznosi 0.98 što znači da je 98 od 100 testnih primjera iz familije UcrQ ispravno klasificirano. Pozitivna prediktivna vrijednost iznosi 1.00 što znači da ni jedan primjer iz preostalih familija nije klasificiran u familiju UcrQ.

	100 motiva	150 motiva
C = 0.01	**97.86%**	**98.71%**
C = 0.1	97.29%	98.43%
C = 1	97.00%	98.00%
C = 10	96.29%	97.71%
C = 100	96.57%	97.85%
C = 1000	96.57%	97.85%

Tablica 4.2: Točnost linearnog SVM algoritma za različite vrijednosti parametra C

	familija	osjetljivost	PPV	točnost
100 motiva	UcrQ	0.98	1.00	**97.86%**
	Upf2	0.97	1.00	
	Glicerol 3-fosfat	0.99	1.00	
	Kromatin	0.96	0.99	
	XendoU	0.97	1.00	
	Plasmid	0.99	0.95	
	Peptidase S8	0.99	0.92	
150 motiva	UcrQ	0.98	1.00	**98.71%**
	Upf2	0.99	0.99	
	Glicerol 3-fosfat	1.00	1.00	
	Kromatin	0.96	0.99	
	XendoU	0.99	1.00	
	Plasmid	0.99	0.98	
	Peptidase S8	1.00	0.95	

Tablica 4.3: Rezultati klasifikacije linearnim SVM algoritmom za $C = 0.01$

4.3 Usporedba rezultata

Cilj je usporediti dvije metode linearne klasifikacije, klasifikaciju max-normom i algoritmom stroja potpornih vektora. Također, zanima nas kolika je važnost broja motiva koje uzimamo za izradu modela koji opisuje odnos nekog proteina sa proteinskim familijama u koje ga želimo klasificirati. Promotrimo, stoga, sve dobivene rezultate u tablici 4.4.

Uočavamo da je klasifikacija algoritmom stroja potpornih vektora kod oba modela bolja, no neznatno. Klasifikacija max-normom je jednostavnija metoda koja daje gotovo jednako dobre rezultate, pa se kod klasifikacije sa pripremom podataka opisanom u ovom radu ona može koristiti kao mjerodavna. Kada govorimo o broju motiva pomoću kojih gradimo model, prema očekivanju veći broj motiva bolje opisuje proteinske familije, pa su i rezultati klasifikacije nešto bolji. Iako postoji poboljšanje, možemo zaključiti da i manji broj motiva gotovo jednako dobro opisuje proteinske familije.

	100 motiva	150 motiva
max-norma	97.14%	97.57%
linearni SVM	97.86%	98.71%

Tablica 4.4: Točnost linearne klasifikacije za različite pristupe

Kako bismo opravdali postupak pripreme podataka na način opisan u poglavlju 3, usporedit ćemo rezultate klasifikacije dobivene algoritmom stroja potpornih vektora sa rezultatima iz rada [7]. Navedeni rad donosi rezultate klasifikacije gotovo istog skupa podataka pri čemu se za izradu modela koristi nešto jednostavniji pristup, točnije rječnik sastavljen od svih mogućih 5-grama koji se nalaze u odabranim proteinskim familijama. Osim rječnika korištena je matrica frekvencija za svaki 5-gram. Drugim riječima, izbrojano je koliko se puta svaki 5-gram pojavljuje unutar pojedine proteinske familije. Opisani pristup rezultirao je rječnikom sačinjenim od 603531 riječi pomoću kojeg je izrađen model. Dakle, za zadani protein model daje rezultat u obliku vektora duljine 603531 koji opisuje odnos danog proteina i proteinskih familija koje su sudjelovale u izradi rječnika. Iako dobiveni rezultati klasifikacije imaju zadovoljavajuću točnost, cilj je smanjiti dimenziju prostora u kojemu se opisani vektori nalaze.

Prisjetimo se, u ovom radu koristili smo modele čiji su rezultati za zadani protein bili vektori duljine 700, odnosno 1050, što je značajno smanjenje dimenzije prostora u kojem algoritam stroja potpornih vektora određuje razdvajajuću hiperravninu. Smanjenje dimenzije rezultira znatno kraćim vremenom izvršavanja algoritma.

Promotrimo rezultate različitih pristupa u tablici 4.5. Uočavamo da se korištenjem motiva pri izradi modela povećava točnost linearne klasifikacije u odnosu na izradu modela pomoću rječnika. No, prava vrijednost pristupa pripremi podataka na način opisan u ovom radu upravo je u smanjenju dimenzije vektorskog prostora u koji zapisujemo primjere koje želimo klasificirati. Zaključujemo da je višestruko isplativo uložiti vrijeme u nešto kompliciraniju metodu otkrivanja značajki pojedinih klasa.

	100 motiva	150 motiva	rječnik
C = 0.01	**97.86%**	**98.71%**	**93.80%**
C = 0.1	97.29%	98.43%	93.80%
C = 1	97.00%	98.00%	93.80%
C = 10	96.29%	97.71%	93.80%
C = 100	96.57%	97.85%	93.80%
C = 1000	96.57%	97.85%	93.80%

Tablica 4.5: Točnost linearne klasifikacije SVM algoritmom za različite pristupe

Bibliografija

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

[2] N.Sarapa *Teorija vjerojatnosti*, Školska knjiga, Zagreb, 2002.

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

[4] M. Mirković *Preciznost i klasifikacija*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odjsek, 2020.

[5] M. Cigula *Iterativna optimizacija modela i pretraživanje proteoma*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odjsek, 2016.

[6] M.Kobovac *Neki aspekti iterativnog pretraživanja proteoma*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odjsek, 2017.

[7] F. Janjić *Semantičko indeksiranje i klasifikacija dokumenata*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odjsek, 2019.

[8] A. Kokor *Pretraživanje, usporedba i klasifikacija*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odjsek, 2019.

[9] C.M.Bishop *Pattern Recognition and Machine Learning*, New York, 2016.

[10] S. Ray, *Understanding Support Vector Machine algorithm from examples* `https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/`, 10. studeni, 2019.

Sažetak

Cilj rada je analiza linearne klasifikacije proteina pomoću max-norme i algoritma stroja potpornih vektora. Na početku rada dani su osnovni matematički pojmovi iz linearne algebre, optimizacije, vjerojatnosti i statistike potrebni za razumijevanje rada. Također, opisan je rad algoritma stroja potpornih vektora. Za analizu se koristi sedam proteinskih familija. Skup za učenje sačinjen je od po 300 proteina iz svake proteinske familije, dok je skup za testiranje sačinjen od po 100 proteina iz svake proteinske familije. Objašnjen je postupak određivanja značajki svake familije te način izrade modela koji opisuju odnos svakog proteina sa zadanim proteinskim familijama. U konačnici su provedeni testovi linearne klasifikacije pomoću max-norme odnosno algoritma stroja potpornih vektora za različite parametre, te je napravljena usporedba dobivenih rezultata. Pokazalo se da su rezultati klasifikacije za različite metode gotovo isti i jako dobri.

Summary

The aim of this work is an analysis of protein classification using the max-norm and support vector machine algorithm. First, we present some basic concepts from linear algebra, optimization, probability, and statistics that are needed for understanding later. Also, we explained mathematical background of the support vector machine algorithm. There are 7 protein families used for analysis. The training set consists of 300 proteins from each family, and the test set consists of 100 proteins from each family. We described the process of determining features for each family and how to build a model that describes the relation between given protein and protein families. Finally, we implemented the linear classification using the max-norm and support vector machine algorithm for different parameter values. It is shown that results are almost the same for both algorithms, and classification is very accurate.

Životopis

Rođena sam 14.09.1994. godine u Puli gdje sam i odrasla. Pohađala sam Osnovnu školu Kaštanjer te Osnovnu glazbenu školu Ivan Matetić Ronjgov. Obrazovanje nastavljam u Tehničkoj školi Pula gdje sam 2013. godine stekla zvanje arhitektonski tehničar. Zatim upisujem Preddiplomski studij matematike na Prirodoslovno-matematičkom fakultetu u Zagrebu koji završavam 2017. godine. Po završetku preddiplomskog studija, na istom fakultetu upisujem Diplomski sveučilišni studij Matematička statistika.