SIMPLEKS METOD ZA TRAŽENJE TEŽINSKOG VEKTORA

Prije nego što se predloži postupak za traženje težinskog vektora $\vec{w}$, razmotrit ćemo dataset sastavljen od 30 instanci, koji služi da se validira logistička regresija u kontekstu predviđanja da li će određeni pacijent sa nekim procentom bubrežnog oboljenja preživjeti ili ne. Dakle, dataset *D* ovako izgleda

$$D = \{(\vec{x}_i, y_i) : y_i \in \{0,1\}, i = \overline{1,30}\},$$

pri čemu $\vec{x}_i = \begin{pmatrix} x_1^i & x_2^i & x_3^i \end{pmatrix}$. Oznaka $y_i = 0$ znači pacijent je preživio, dok oznaka $y_i = 1$ implicira da pacijent nije preživio. Atributi vektora $\vec{x}_i$ imaju sljedeća značenja:

- atribut x_1^i označava godine i-tog pacijenta;
- atribut x_2^i označava spol i-tog pacijenta, pri čemu -1 označava muški spol, dok +1 ženski spol;
- atribut x_3^i označava stepen bubrežnog oboljenja i-tog pacijenta.

Na primjer, instanca $\vec{x}_1 = (48 \quad 1 \quad 4.40)$ sa oznakom $y_1 = 0$ predstavlja ženskog pacijenta odnosno ženu koja je preživjela zbog ne velikog stepena bubrežnog oboljenja (4.40), što će kasnije biti pokazano kada se bude radila predikcija. S druge strane, instanca odnosno primjer $\vec{x}_2 = (60 \quad -1 \quad 7.89)$ sa oznakom $y_2 = 1$, predstavlja muškog pacijenta odnosno muškarca koji nažalost zbog svog stepena bubrežnog oboljenja od 7.89 nije uspio preživjeti. Od ranije znamo da se linearna regresija u oznaci $h_{LinR}(\vec{x}; \vec{w})$ ovako zadaje

$$h_{LinR}(\vec{x}; \vec{w}) = \vec{w}^T \vec{x}.$$

Važno je prisjetiti se da bi se omogućilo skalarno množenje vektora $\vec{x} = [x_1 \quad x_2 \quad x_3]$ i $\vec{w} = [w_0 \quad w_1 \quad w_2 \quad w_3]$, neophodno je vektor $\vec{x}$ proširiti sa još jednim atributom (obično je to **lažna varijabla** (eng. *dummy variable*), koja se obično postavlja na vrijednost jedan, pa se u skladu sa tim vektor $\vec{x}$ zapisuje u ovoj formi $\vec{x} = [1 \quad x_1 \quad x_2 \quad x_3]$.

Također, znamo da se logistička regresija gradi pomoću linearne regresije djelovanjem sigmoidne funkcije δ na nju, tj. logistička regresija u oznaci $h_{LogR}(\vec{x}; \vec{w})$ ovako se zadaje

$$h_{LogR}(\vec{x}; \vec{w}) = \delta(\vec{w}^T \cdot \vec{x}) = \frac{1}{1+e^{-\vec{w}^T \cdot \vec{x}}}.$$

Da bi se radila klasifikacija logističkom regresijom obično se instanca $\vec{x}$ za koju vrijedi $h_{LogR}(\vec{x}; \vec{w}) \geq 0.5$ klasificira u pozitivnu klasu, tj. klasu sa oznakom $y = 1$, dok ukoliko vrijedi $h_{LogR}(\vec{x}; \vec{w}) < 0.5$, onda se instanca klasificira u negativnu klasu, tj. klasu sa oznakom $y = 0$ ili $y = -1$ u zavisnosti od toga koja oznaka više odgovara.

Primjer 1. Neka je za pacijenta $\overrightarrow{x_1} = [1 \quad 50 \quad -1 \quad 6]$ pronađen ovakav težinski vektor $\vec{w} = [w_0 \quad w_1 \quad w_2 \quad w_4] = [-7.5 \quad 0.11 \quad -0.22 \quad 0.33]$. Odredite da li će pacijent preživjeti ili ne.

Rješenje. Kako za pacijenta $\overrightarrow{x_1}$ važi $h_{LinR}(\overrightarrow{x_1}; \vec{w}) = \vec{w}^T \cdot \vec{x}_1 = -7.5 + 0.11 \cdot 50 + 0.22 + 6 \cdot 0.33 = 0.20$, to je onda vrijednost logističke regresije jednaka

$$h_{LogR}(\overrightarrow{x_1}; \vec{w}) = \frac{1}{1 + e^{-h_{LinR}(\overrightarrow{x_1}; \vec{w})}} = \frac{1}{1 + e^{-0.20}} = 0.5498.$$

Kako je $h_{LogR}(\overrightarrow{x_1}; \vec{w}) \geq 0.5$, slijedi da pacijent $\overrightarrow{x_1}$ nažalost neće preživjeti. Ovaj rezulatat se sa vjerovatnosnog aspekta može ovako intrepretirati: *Vjerovatnoća da će pacijent* $\overrightarrow{x_1}$ *završiti u pozitivnoj klasi* $y = 1$ *jednaka je 0.5498, dok je vjerovatnoća da ista ta instanca bude klasificirana u klasi sa oznakom* $y = 0$ *jednaka je 1-0.5498=0.4502*. Drugim riječima, ako se ovo izrazi u procentima, slijedi da mogućnost da pacijent $\overrightarrow{x_1}$ završi u klasi $y = 1$ iznosi 54.98%, a u klasi $y = 0$ samo 45.02%.

Prije nego što se predloži algoritam baziran na Simpleks metodi za traženje težinskog vektora $\vec{w}$, kod logističke regresije u najvećem broju slučajeva radi se Gausova normalizacija podataka (u slobodno vrijeme pročitajte nešto o Carl Friedrich Gaussu, ostavio je dubok trag u nauci).

Gausova normalizacija podataka

Neka je zadat dataset D ovako $D = \{(\vec{x}_i, y_i) : i = \overline{1,p}\}$, pri čemu je p ukupan broj instanci tog dataseta. Normalizacija ovog dataseta obavlja se tako što se svaka instanca $\vec{x}_i$ normalizira preko standardne Gausove distribucije. Da bi se instanca normalizirala, neophodno je da se svaka kolona x_i^j $(j = 1,2,\cdots,n)$ te instance normalizira na sljedeći način

$$x_i^j = \left(x_i^j - m_j\right)/SD_j,$$

pri čemu n ukupan broj kolona instance $\vec{x}_i$, m_j je srednja vrijednost kolone x_i^j cijelog dataseta D, a SD_j je standardna devijacija te kolone u odnosu na D. Ona se ovako računa

$$SD_j = \sqrt{\frac{1}{p-1}\sum_{i=1}^{p}\left(x_i^j - m_j\right)^2} \quad (\forall j).$$

Srednja vrijednost m_j kolone x_i^j računa se kao

$$m_j = \frac{1}{p}\sum_{i=1}^{p} x_i^j \quad (\forall j).$$

Primjer 2. Neka je dat dataset $D = \{(\vec{x}_i, y_i) : i = \overline{1,4}\}$, tako da su starosne godine za pacijente $\overrightarrow{x_1}$, $\overrightarrow{x_2}$, $\overrightarrow{x_3}$ i $\overrightarrow{x_4}$ redom date sa 25, 36, 40 i 23. Pomoću standardne Gausove distribucije, normalizirajte samo prvu kolonu pacijenta $\overrightarrow{x_1}$.

Rješenje. Prvo se izračuna srednja vrijednost m_1, a potom standardna devijacija SD_1:

$$m_1 = \frac{1}{4}(25 + 36 + 40 + 23) = 31.0,$$

$$SD_1 = \sqrt{[(25-31)^2 + (36-31)^2 + (40-31)^2 + (23-31)^2]/3} = 8.29.$$

Dakle, prva kolona pacijenta $\overrightarrow{x_1}$ nakon normalizacije jednaka je $x_1^1 = \frac{25-31}{8.29} = -0.72.$

Simpleks optimizacijja

U ovoj se sekciji daje pseudokod za traženje težinskog vektora $\vec{w}$ kod logističke regresije koji je baziran na paradigmi Simpleksa.

Pseudokod se sastoji od tri koraka:

Korak 1. Sasvim slučajno kreirati tri rješenja R1, R2 i R3.

Korak 2. Iterirati određen broj iteracija ili dok neka unaprijed zadata vrijednost ϵ nije postignuta.

A) Sortirati rješenja R1, R2 i R3 u odnosu na vrijednost funkcije cilja h_{LogR}, čime se dobijaju sljedeća rješenja: NAJGORE RJEŠENJE (**WS**), NAJBOLJE RJEŠENJE (**BS**), DRUGO RJEŠENJE (**OS**).

B) Pronaći *PROŠIRENO RJEŠENJE* **(ES**). Ukoliko je ES bolje od WS u odnosu na vrijednost funkcije cilja $h_{LogR}(ES) < h_{LogR}(WS)$, tada zamijeniti WS sa ES, i vratiti se na korak 2.

C) Pronaći *REFLEKTOVANO RJEŠENJE* (**RS**). Ukoliko je RS bolje od WS, tada zamijeniti WS sa RS, tj. WS postaviti na RS i vratiti se na korak 2.

D) Pronaći *KONTRAKTOVANO RJEŠENJE* (**CS**). Ako je CS bolje od WS, tada WS postaviti na CS i vratiti se na korak 2.

E) Pronaći *RANDOM RJEŠENJE* (**RR**). Ako je RR bolje od WS, tada WS postaviti na RR i vratiti se na korak 2.

F) Obaviti operaciju "SHRINKING", kod koje se rješenja **WS** i **OS** ažuriraju na ovaj način:

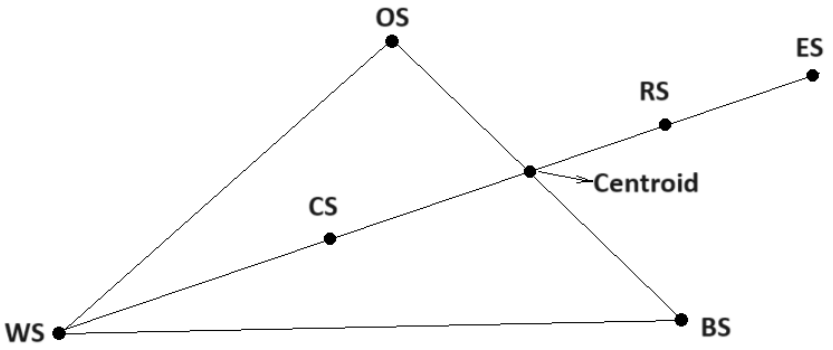
WS=0.5(WS+BS); OS=0.5(OS+BS).

Potom se vratiti se na korak 2.

Korak 3. Odštampati težinski vektor $\vec{w}$.

<u>Opis nalaženja rješenja CS, RS i ES</u>

Rješenja CS, RS i ES se nalaze na osnovu trougla, čija su tjemena zapravo rješenja OS, WS i BS, koja se u xy ravni sasvim slučajno inicijalno kreiraju. Jedan primjer zadavanja ovih rješenja grafički je ispod prikazan.



Sa gornje slike, nije teško uočiti da rješenje Centroid je zapravo središte duži $\overline{OSBS}$, pa se nalazi ovako

$$Centroid = 0.5(OS + BS).$$

Također, **rješenje CS** se može odrediti da bude središte duži $\overline{WSCentroid}$, ali gotovo uvijek u praksi ovakav izbor rješenja neće voditi ka najboljem odabiru težinskog vektora $\vec{w}$. Prema tome, budući da rješenje CS generalno ne mora biti središte spomenute duži, ono se gotovo uvijek nalazi na općenitiji način, kao što je ispod urađeno

$$CS = Centroid - \beta(Centroid - WS),$$

pri čemu je β hiperparametar iz intervala $(0,1)$. Ukoliko je $\beta = 0$ ili $\beta = 1$, tada CS=Centroid ili CS=WS, što nema smisla, pa se ove kombinacije za parameter β zabranjuju. Za $\beta = 0.5$, slijedi da je CS središte duži $\overline{WSCentroid}$.

Rješenje RS se nalazi ovako

$$RS = Centroid + \alpha(Centroid - WS),$$

pri čemu se hiperparametar α uzima iz intervala $(0,1)$.

Rješenje ES može se ovako odrediti

$$ES = Centroid + \gamma(Centroid - WS),$$

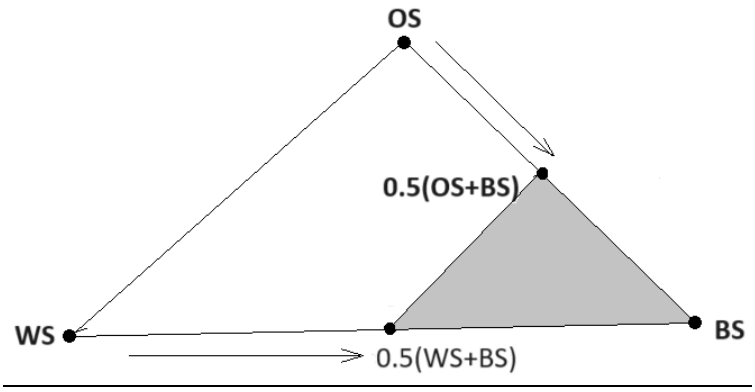
pri čemu se hiperparametar γ uzima iz intervala $(0,3)$.

Iako su jednačine za traženje rješenja RS i ES veoma slične (razlika je u parametrima β i γ), obično se vrijednosti za ove parametre u implementaciji uzimaju da budu različite, npr. $\beta = 0.5$ i $\gamma = 2$, ili ako se radi tuniranje ovih parametara preko neke metode (npr. metoda mreže), tada ta metoda pronađe vrijednosti za koje vrijedi $\beta < \gamma$.

Važno je uočiti da rješenja CS, RS i ES zapravo rade "lokalnu pretragu", dok rješenje RR radi "globalnu pretragu" prostora pretrage. Prema tome, itekako je važno što postoji korak D), jer on u najvećem broju slučajeva ne dopušta da se algoritam "zaglavi" u nekom od lokalnih optimuma.

Operacija SHRINKING

Nije teško primijetiti na osnovu drugog koraka gornjeg pseudokoda (pod koraci A), B), C) i D)) da ukoliko nijedno od rješenja ES, RS, CS i RR nije bolje od rješenja WS, tada početni trougao (vidi gore sliku) $\Delta OSWSBS$ treba skalirati odnosno treba se istom smanjiti površina, jer se ne radi uspješna pretraga 2D prostora. Drugim riječima, treba se pojačati lokalna pretraga, pa se zbog toga rješenja WS i OS trebaju ažurirati. Njihovo ažuriranje se treba obavljajti u pravcu najboljeg rješenja BS, tj. stara rješenja WS i OS redom se mijenjaju sa novim rješenjima 0.5(WS+BS) i 0.5(OS+BS), što je urađeno na slici ispod.



Napomena: Do boljeg razumijevanja izložene Simpleks metode za traženje težinskog vektora $\vec{w}$ kod logističke regresije možete doći tako što ćete analizirati implementaciju ove metode izvedenu u programskom jeziku C#, koju je autor ovih materijala detaljno opisao i veoma pažljivo izabrao strukture podataka, vodeći pri tome računa o optimalnom korištenju resursa, kao što je memorija računara.