Istraživanje podataka

Vežbe 9

9. maj 2021

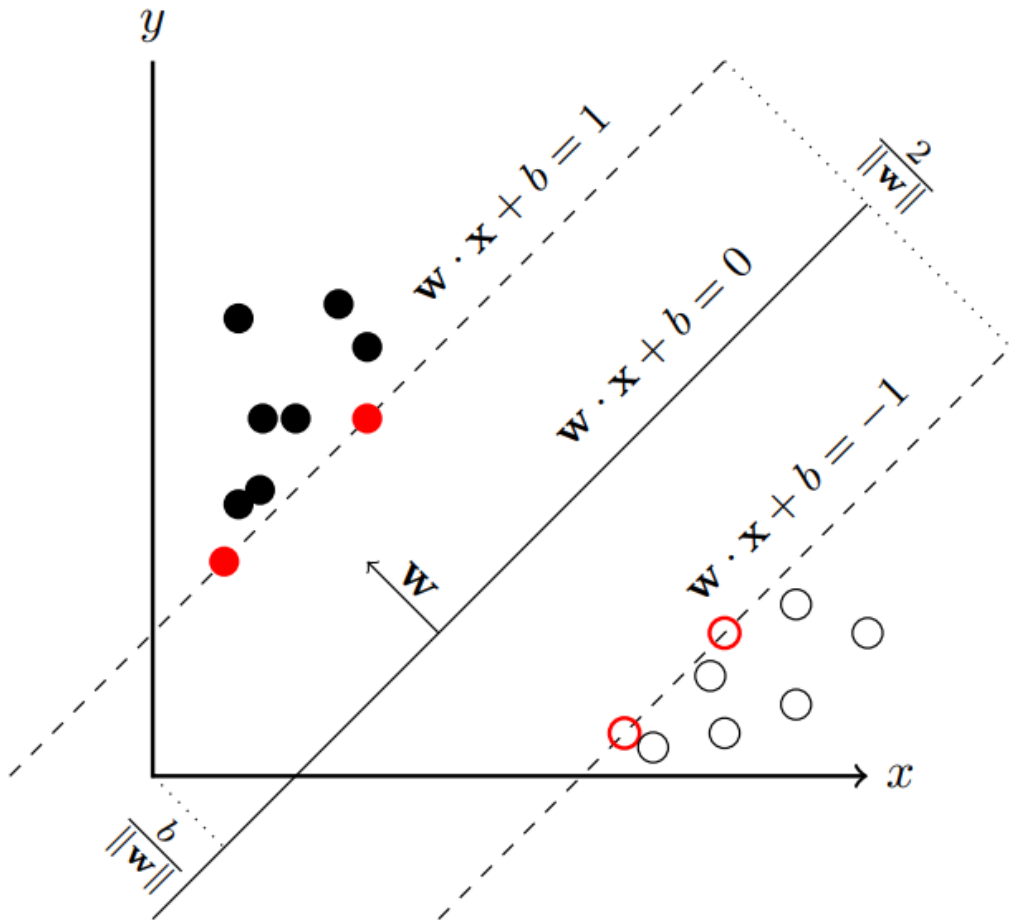
Outline

1. SVM - Metod podržavajućih (potpornih) vektora
2. Zadatak za samostalan rad
3. Klasterovanje, algoritam K-sredina

Outline

1. SVM - Metod podržavajućih (potpornih) vektora
2. Zadatak za samostalan rad
3. Klasterovanje, algoritam K-sredina

Metod podržavajućih (potpornih) vektora - linearno separabilan skup



Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- Pretpostavka: podaci su linearno razdvojivi
- Klase: 1 i -1
- Cilj: naći optimalnu hiper-ravan, tj. hiper-ravan sa maksimalnom marginom koja razdvaja instance klase 1 i instance klase -1

$$w * x + b = 0$$

- Instance na margini - podržavajući (potporni) vektori

Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- Ograničenja margina

$$w * x + b = 1$$

$$w * x + b = -1$$

- Ograničenja za instance
  - za $y_i = 1$ $w * x_i + b \geq 1$
  - za $y_i = -1$ $w * x_i + b \leq -1$
  - ili $\forall i$ iz skupa $y_i * (w * x_i + b) \geq 1$

Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- Rastojanje od hiper-ravni: $\frac{|w*x+b|}{||w||}$
- Rastojanje od hiper-ravni do podržavajućih vektora: $\frac{1}{||w||}$
- Margina: $\frac{2}{||w||}$
- Zahtev

$$\max_w \frac{2}{||w||}$$

uz ograničenja $y_i * (w * x_i + b) \geq 1, i = 1, 2, ..., N$

- Ili

$$\min_w \frac{||w||^2}{2}$$

uz ograničenja $y_i * (w * x_i + b) \geq 1, i = 1, 2, ..., N$

- Klasifikacija instance z: $sign(w * z + b)$

Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- Optimizacioni problem može da se reši korišćenjem metoda Lagranžovi množioci
- $L_p = \frac{||w||^2}{2} - \sum_{i=1}^{N} \lambda_i * [y_i * (w * x_i) + b - 1]$
- nenegativni Lagranžovi množioci λ_i se povezuju sa ograničenjima

Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- Važi

$$w = \sum_{i=1}^{l} y_i \alpha_i * (x_i)$$

$$\sum_{i=1}^{l} y_i \alpha_i = 0$$

- Važi $\lambda_i = 0$ osim $y_i * (w * x_i) + b = 1$
- Ako je $\lambda_i > 0$, x_i je potporni vektor

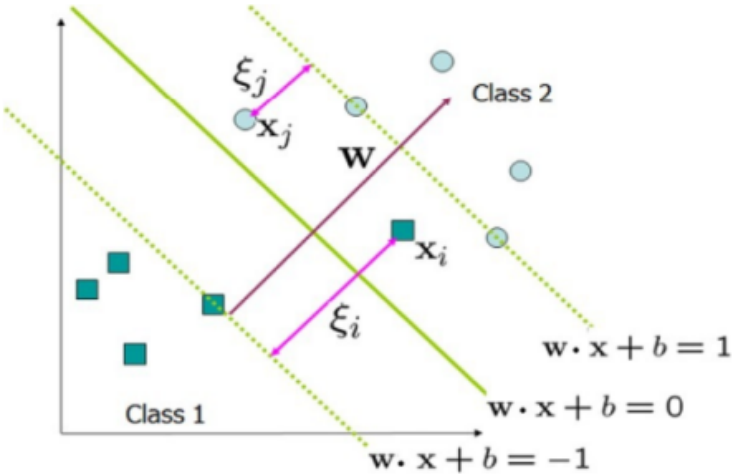
Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- dualni problem (transformacija Lagranžijana u funkciju od Lagranžijanovih množioca)

$$\max_{\lambda_i} L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} * \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i * \lambda_j * y_i * y_j * x_i * x_j$$

- Klasifikacija instance z: $sign(\sum_{i=1}^{N} \lambda_i * y_i * x_i * z + b)$

Metod podržavajućih (potpornih) vektora - meka margina



Metod podržavajućih (potpornih) vektora - meka margina

- Ako instance skupa nisu linearno razdvojive, uvode se promenljive u ograničenja za svaku instancu kako bi se olabavila
- Ograničenja za instance
  - za $y_i = 1$ $w * x_i + b \geq 1 - \xi_i$
  - za $y_i = -1$ $w * x_i + b \leq -1 + \xi_i$

Metod podržavajućih (potpornih) vektora - meka margina

- Ciljna funkcija

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

- C i k (1 ili 2) su konstante
- $\sum_{i=1}^{N} \xi_i^k$ - gubitak
- dualni problem $L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} * \sum_{i,j} \lambda_i * \lambda_j * y_i * y_j * x_i * x_j$

Nelinearni metod podržavajućih (potpornih) vektora

- Za skupove koji nisu linearno razdvojivi, potrebno je odrediti funkciju Φ koja će skup transformisati u prostor sa linearno razdvojivim instancama
- Traži se $w * \Phi(x) + b = 0$
- Klasifikacija instance z: $sign(\sum_{i=1}^{N} \lambda_i * y_i * \Phi(x_i) * \Phi(z) + b)$
- Kernel trik: računanje $\Phi(x_i) * \Phi(z)$ u transformisanom prostoru koristeći originalne atribute

Nelinearni metod podržavajućih (potpornih) vektora

- Zahtev

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

- Ograničenja za instance

$$y_i * (w * \Phi(x_i) + b) \geq 1 - \xi_i, i = 1, 2, ..., N$$

$$\xi_i \geq 0$$

Metod podržavajućih (potpornih) vektora u biblioteci scikit-learn

- *sklearn.svm.SVC*
- parametri
  - C - kazna za grešku: default 1
  - kernel funkcija
    - *linear*: $\langle x, x' \rangle$
    - *poly*: $(\gamma \langle x, x' \rangle + r)^d$, γ se zadaje parametrom *gamma*: d parametrom *degree*, a r sa *coef0*
    - *rbf*: $exp(-\gamma ||x - x'||^2)$ γ se zadaje parametrom *gamma* i mora biti pozitivna
    - *sigmoid*: $tanh((\gamma \langle x, x' \rangle + r))$

Metod podržavajućih (potpornih) vektora u biblioteci scikit-learn

- *degree* - stepen za kernel funkciju *poly*, default=3
- *gamma* - koeficijent za kernel funkcije: *rbf, poly, sigmoid*, default='auto', tj. 1/broj_atributa
- *coef0* - nezavisan član u kernel funkcijama: *poly, sigmoid*,

Metod podržavajućih (potpornih) vektora u biblioteci scikit-learn

- atributi
  - *support_* - indeksi podržavajućih vektora
  - *support_vectors_* - podržavajući vektori
  - *n_support_* - broj podržavajućih vektora za svaku klasu
  - *dual_coef_* - koeficijenti podržavajućih vektora $y_i\alpha_i$. Ukoliko postoji više klasa, postoje koeficijenti za sve *1-vs-1* klasifikatore.
  - *intercept_* - konstane u funckiji odlučivanja

Neuronska mreža sa skrivenim slojevima u biblioteci scikit-learn

- metode
  - *fit(X, y)* - za treniranje modela
  - *predict(X)* - za predviđanje klasa

Outline

1 SVM - Metod podržavajućih (potpornih) vektora

2 Zadatak za samostalan rad

3 Klasterovanje, algoritam K-sredina

Zadatak za samostalan rad

Preuzeti skup podataka *car.csv* o klasama automobila. Koristeći IBM SPSS Modeler i SVM izvršiti klasifikaciju nad datim skupom. Atribut *class* sadrži informaciju kojoj klasi pripada automobil. Primeniti PCA na skup i izvršiti klasifikaciju nad transformisanim skupom. Koji broj atributa ste izabrali i zašto? Diskutovati dobijen model.

Zadatak za samostalan rad

Skup *car* sadrži podatke o automobilima. Atributi skupa su:

- class - klasa automobila
- cylinders - broj cilindara
- displacement - zapremina motora
- horsepower - konjska snaga
- weight - težina
- acceleration - ubrzanje

Outline

1 SVM - Metod podržavajućih (potpornih) vektora

2 Zadatak za samostalan rad

3 Klasterovanje, algoritam K-sredina

Klasterovanje

Primenom klasterovanja nad skupom podataka vrši se grupisanje instanci sa ciljem da instance jedne grupe budu što sličnije, a što udaljenije od instanci iz drugih grupa. Jedna grupa instanci dobijena klasterovanjem naziva se klaster.

Algoritam: K-sredina

Pronalazak klastera u algoritmu K-sredina je iterativni proces računanja centroida za svaki klaster i dodeljivanja instance klasteru.

Algoritam

1. Određivanje inicijalnih centroida za k klastera.
2. Svaku instancu dodeliti najbližem klasteru korišćenjem mere bliskosti.
3. Za svaki klaster ažurirati centroid na osnovu dodeljenih instanci tom klasteru.
4. Ponavljati korake 2 i 3 dok se ne ispuni uslov: nijedan centroid se nije promenio u odnosu na prethodnu iteraciju.

Algoritam: K-sredina

Za algoritam K-sredina potrebno je definisati

- parametar **K** - broj željenih klastera
- meru bliskosti (mera sličnosti ili različitosti) koja se koristi za računanje bliskosti između instance skupa i centroida klastera.
- inicijalne centroide.

Zadatak 1

Algoritmom K-sredina identifikovati 3 klastera u sledećim podacima. Pri tom, koristiti euklidsko rastojanje. Za polazne centroide uzeti prve tri instance.

X	Y	Z
1	0	2
2	0	0
-3	-1	1
-4	-2	2
0	4	9
1	5	9

Zadatak 1

Značenje oznaka koje se koriste u rešenju:
c_i - centroid klastera i
C_i - instance u klasteru i

Zadatak 1

Iteracija I

U tabeli je prikazana matrica rastojanja između instanci i inicijalnih centroida. Za svaku instancu je podebljano rastojanje do najbližeg centroida i njegovom klasteru se instanca dodeljuje.

centroid	i_1	i_2	i_3	i_4	i_5	i_6
c_1	0			5,4	**8,1**	**8,6**
c_2		0		6,6	10	10,3
c_3			0	**1,7**	9,9	10,8

Tabela: Matrica rastojanja između instanci i centroida za iteraciju 1

Zadatak 1

Nakon I iteracije podela instanci po klasterima je:

- $C_1 : i_1, i_5, i_6$
- $C_2 : i_2$
- $C_3 : i_3, i_4$

Novi centroidi su:

- $c_1 = \frac{i_1+i_5+i_6}{3} = (0,67; 3; 6,67)$
- $c_2 = i_2 = (2; 0; 0)$
- $c_3 = \frac{i_3+i_4}{2} = (-3,5; -1,5; 1,5)$

Zadatak 1

Iteracija II

Tabela: Matrica rastojanja između instanci i centroida za iteraciju II

centroid	i_1	i_2	i_3	i_4	i_5	i_6
c_1	5,6	7,4	7,8	8,3	**2,6**	**3,1**
c_2	**2,2**	**0**	5,2	6,6	10	10,3
c_3	4,8	5,9	**0,9**	**0,9**	9,9	10,9

Zadatak 1

Nakon II iteracije podela instanci po klasterima je:

- $C_1 : i_5, i_6$
- $C_2 : i_1, i_2$
- $C_3 : i_3, i_4$

Novi centroidi su:

- $c_1 = \frac{i_5 + i_6}{2} = (0,5; 4,5; 9)$
- $c_2 = \frac{i_1 + i_2}{2} = (1,5; 0; 1)$
- $c_3 = \frac{i_3 + i_4}{2} = (-3,5; -1,5; 1,5)$

Zadatak 1

Iteracija III

Tabela: Matrica rastojanja između instanci i centroida za iteraciju III

centroid	i_1	i_2	i_3	i_4	i_5	i_6
c_1	8,3	10,2	10,3	10,6	**0,7**	**0,7**
c_2	**1,1**	**1,1**	4,6	5,9	9,1	9,4
c_3	4,8	5,9	**0,9**	**0,9**	9,9	10,9

Zadatak 1

Nakon III iteracije podela instanci po klasterima je:

- $C_1 : i_5, i_6$
- $C_2 : i_1, i_2$
- $C_3 : i_3, i_4$

Primetiti da je podala instanci po klasterima ista u II i III iteraciji, zbog čega neće doći do promene u vrednostima centroidima, i time je klasterovanje završeno.

Algoritam: K-sredina

Kada se kao mera bliskosti koristi rastojanje u Euklidskom prostoru, za evaluaciju klasterovanja algortimom K-sredina često se koristi mera suma kvadrata greške (SSE)

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(c_i, x)^2$$

gde je

- x instanca skupa
- C_i klaster
- c_i centroid klastera C_i

Cilj je da SSE bude što manja.