I007 Osnove umjetne inteligencije

Tema: Strojno učenje: naivan Bayesov klasifikator i perceptron.

19. 5. 2021.

predavač: Darija Marković

asistent: Rebeka Čorić

Strojno učenje: kako dobiti model iz podataka / iskustva

- učenje parametara
- učenje strukture
- učenje skrivenih koncepata

Klasifikacija: za dani ulaz x, treba predvidjeti klasu (oznaku) y

Primjeri

- detekcija neželjenih poruka e-pošte
- optičko prepoznavanje znakova
- medicinska dijagonostika (ulaz: simptomi, izlaz: dijagnoza)
- automatizirano ocjenjivanje eseja
- klasifikacija novinskih dokumenata u rubrike
- predviđanje kretanja dionica
- otkrivanje prevara
- predviđanje ishoda nogometnih utakmica

Klasifikacija: za dani ulaz x, treba predvidjeti klasu (oznaku) y

Primjeri

- detekcija neželjenih poruka e-pošte
- optičko prepoznavanje znakova
- medicinska dijagonostika (ulaz: simptomi, izlaz: dijagnoza)
- automatizirano ocjenjivanje eseja
- klasifikacija novinskih dokumenata u rubrike
- predviđanje kretanja dionica
- otkrivanje prevara
- predviđanje ishoda nogometnih utakmica

Klasifikacija temeljena na modelu

- izgraditi model (npr. Bayesovu mrežu) gdje su i značajke i klase slučajne varijable
- zabilježiti sve uočene značajke
- odrediti distribuciju klasa uvjetovanu značajkama

Pitanja:

- kakvu strukturu bi Bayesova mreža trebala imati?
- kako naučiti njene parametre?

Klasifikacija temeljena na modelu

- izgraditi model (npr. Bayesovu mrežu) gdje su i značajke i klase slučajne varijable
- zabilježiti sve uočene značajke
- odrediti distribuciju klasa uvjetovanu značajkama

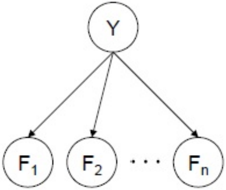
Pitanja:

- kakvu strukturu bi Bayesova mreža trebala imati?
- kako naučiti njene parametre?

1. Naivan Bayesov klasifikator

2. Perceptron

Naivan bayesov klasifikator: pretpostavlja da su sve značajke nezavisni efekti klase



$$P(Y, F_1, F_2, \ldots, F_n) = P(Y) \prod_i P(F_i|Y)$$

- potrebno je samo odrediti kako svaka značajka ovisi o klasi
- ukupni broj parametara je linearan u n
- model je jako pojednostavljen, no unatoč tome često dobro radi

Zaključivanje s Naivnim Bayesovim klasifikatorom

Cilj: izračunati posterior vjerojatnost varijable klase Y $(P(Y|f_1, \dots, f_n))$

- korak 1: odrediti zajedničku vjerojatnost klase i svih dokaza (značajki) za svaku klasu

$$P(Y, f_1, \dots, f_n) = \begin{bmatrix} P(y_1, f_1, \dots, f_n) \\ P(y_2, f_1, \dots, f_n) \\ \vdots \\ P(y_k, f_1, \dots, f_n) \end{bmatrix} = \begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}$$

- korak 2: zbrojiti kako bi se dobila vjerojatnost dokaza

$$P(f_1, \dots, f_n) = \sum_j P(y_j) \prod_i P(f_i|y_j)$$

- korak 3: normalizirati tako da $P(Y, f_1, \dots, f_n)$ podijelimo s $P(f_1, \dots, f_n)$

Kako bi mogli koristiti naivan Bayesov klasifikator trebamo:

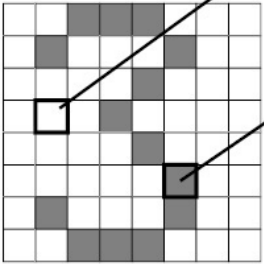
- metodu zaključivanja (upravo opisano)
- procjenu uvjetnih vjerojatnosti:
  - $P(Y)$ za klase
  - $P(F_i|Y)$ za svaku značajku
  - ove vjerojatnosti se zajedno zovu parametri modela i označavaju s θ

Kako bi mogli koristiti naivan Bayesov klasifikator trebamo:

- metodu zaključivanja (upravo opisano)
- procjenu uvjetnih vjerojatnosti:
  - $P(Y)$ za klase
  - $P(F_i|Y)$ za svaku značajku
  - ove vjerojatnosti se zajedno zovu parametri modela i označavaju s θ

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Učenje i testiranje

- podaci: označeni primjeri
  - skup za učenje
  - skup za provjeru
  - skup za testiranje
- značajke: parovi atribut-vrijednost koji opisuje svaki primjer
- ciklus eksperimentiranja
  - naučiti parametre na skupu za učenje
  - podesiti hiperparametre na skupu za provjeru
  - izračunati točnost na skupu za testiranje
  - važno: nikada ne "viriti" u skup za testiranje
- evaluacija
  - točnost: postotak dobro označenih primjera
- prenaučenost i generalizacija
  - želimo klasifikator koji dobro radi na skupu za testiranje
  - prenaučenost: opisuje skup za učenje jako dobro, ali ne radi dobro na novim primjerima
  - kako bi bolje generalizirali trebamo koristiti zaglađivanje ili neku drugu vrstu poboljšanja procjenitelja

Učenje i testiranje

- podaci: označeni primjeri
  - skup za učenje
  - skup za provjeru
  - skup za testiranje
- značajke: parovi atribut-vrijednost koji opisuje svaki primjer
- ciklus eksperimentiranja
  - naučiti parametre na skupu za učenje
  - podesiti hiperparametre na skupu za provjeru
  - izračunati točnost na skupu za testiranje
  - važno: nikada ne "viriti" u skup za testiranje
- evaluacija
  - točnost: postotak dobro označenih primjera
- prenaučenost i generalizacija
  - želimo klasifikator koji dobro radi na skupu za testiranje
  - prenaučenost: opisuje skup za učenje jako dobro, ali ne radi dobro na novim primjerima
  - kako bi bolje generalizirali trebamo koristiti zaglađivanje ili neku drugu vrstu poboljšanja procjenitelja

Učenje i testiranje

- podaci: označeni primjeri
    - skup za učenje
    - skup za provjeru
    - skup za testiranje
- značajke: parovi atribut-vrijednost koji opisuje svaki primjer
- ciklus eksperimentiranja
    - naučiti parametre na skupu za učenje
    - podesiti hiperparametre na skupu za provjeru
    - izračunati točnost na skupu za testiranje
    - važno: nikada ne "viriti" u skup za testiranje
- evaluacija
    - točnost: postotak dobro označenih primjera
- prenaučenost i generalizacija
    - želimo klasifikator koji dobro radi na skupu za testiranje
    - prenaučenost: opisuje skup za učenje jako dobro, ali ne radi dobro na novim primjerima
    - kako bi bolje generalizirali trebamo koristiti zaglađivanje ili neku drugu vrstu poboljšanja procjenitelja

Učenje i testiranje

- podaci: označeni primjeri
  - skup za učenje
  - skup za provjeru
  - skup za testiranje
- značajke: parovi atribut-vrijednost koji opisuje svaki primjer
- ciklus eksperimentiranja
  - naučiti parametre na skupu za učenje
  - podesiti hiperparametre na skupu za provjeru
  - izračunati točnost na skupu za testiranje
  - važno: nikada ne “viriti” u skup za testiranje
- evaluacija
  - točnost: postotak dobro označenih primjera
- prenaučenost i generalizacija
  - želimo klasifikator koji dobro radi na skupu za testiranje
  - prenaučenost: opisuje skup za učenje jako dobro, ali ne radi dobro na novim primjerima
  - kako bi bolje generalizirali trebamo koristiti zaglađivanje ili neku drugu vrstu poboljšanja procjenitelja

Učenje i testiranje

- podaci: označeni primjeri
  - skup za učenje
  - skup za provjeru
  - skup za testiranje
- značajke: parovi atribut-vrijednost koji opisuje svaki primjer
- ciklus eksperimentiranja
  - naučiti parametre na skupu za učenje
  - podesiti hiperparametre na skupu za provjeru
  - izračunati točnost na skupu za testiranje
  - važno: nikada ne "viriti" u skup za testiranje
- evaluacija
  - točnost: postotak dobro označenih primjera
- prenaučenost i generalizacija
  - želimo klasifikator koji dobro radi na skupu za testiranje
  - prenaučenost: opisuje skup za učenje jako dobro, ali ne radi dobro na novim primjerima
  - kako bi bolje generalizirali trebamo koristiti zaglađivanje ili neku drugu vrstu poboljšanja procjenitelja

Procjena parametara

- procjena distribucije slučajne varijable
- empirijsko: koristi se skup za učenje
  - npr. relativne frekvencije su procjenitelji maksimalne vjerodostojnosti

$$P_{ML}(x) = \frac{\text{broj}(x)}{\text{ukupni broj primjera}}$$

  - na ovaj način može doći do prenaučenosti
- Laplaceovo zaglađivanje: pretpostavimo da se svaki ishod pojavio jedan puta više nego što zaista je

$$P_{LAP}(x) = \frac{\text{broj}(x) + 1}{\text{ukupni broj primjera} + |X|}$$

Procjena parametara

- procjena distribucije slučajne varijable
- empirijsko: koristi se skup za učenje
  - npr. relativne frekvencije su procjenitelji maksimalne vjerodostojnosti

$$P_{ML}(x) = \frac{\text{broj}(x)}{\text{ukupni broj primjera}}$$

  - na ovaj način može doći do prenaučenosti
- Laplaceovo zaglađivanje: pretpostavimo da se svaki ishod pojavio jedan puta više nego što zaista je

$$P_{LAP}(x) = \frac{\text{broj}(x) + 1}{\text{ukupni broj primjera} + |X|}$$

Procjena parametara

- poopćeno Laplaceovo zaglađivanje: pretpostavimo da se svaki ishod pojavio k puta više nego što zaista je

$$P_{LAP,k}(x) = \frac{\text{broj}(x) + k}{\text{ukupni broj primjera} + k|X|}$$

- poopćeno Laplaceovo zaglađivanje za uvjetovanu vjerojatnost

$$P_{LAP,k}(x|y) = \frac{\text{broj}(x,y) + k}{\text{broj}(y) + k|X|}$$

- u praksi Laplaceovo zaglađivanje ne radi najbolje za $P(X|Y)$
- druga mogućnost je linearna interpolacija
  - iz primjera se empirijski procjeni $\hat{P}(X)$ i $\hat{P}(X|Y)$, a novi procjenitelj se definira kao

$$P_{LIN}(x|y) = \alpha\hat{P}(x|y) + (1-\alpha)\hat{P}(x)$$

Procjena parametara

- poopćeno Laplaceovo zaglađivanje: pretpostavimo da se svaki ishod pojavio k puta više nego što zaista je

$$P_{LAP,k}(x) = \frac{\text{broj}(x) + k}{\text{ukupni broj primjera} + k|X|}$$

- poopćeno Laplaceovo zaglađivanje za uvjetovanu vjerojatnost

$$P_{LAP,k}(x|y) = \frac{\text{broj}(x,y) + k}{\text{broj}(y) + k|X|}$$

- u praksi Laplaceovo zaglađivanje ne radi najbolje za $P(X|Y)$
- druga mogućnost je linearna interpolacija
  - iz primjera se empirijski procjeni $\hat{P}(X)$ i $\hat{P}(X|Y)$, a novi procjenitelj se definira kao

$$P_{LIN}(x|y) = \alpha\hat{P}(x|y) + (1-\alpha)\hat{P}(x)$$

Podešavanje na skupu za provjeru

- imamo dvije vrste nepoznanica
  - parametre: vjerojatnosti $P(X|Y)$ i $P(Y)$
  - hiperparametre: količina/tip zaglađivanja, k, α
- odakle što naučiti?
  - parametre učimo iz skupa za učenje
  - hiperparametre podešavamo na skupu za provjeru
  - za svaku vrijednost hiperparametara naučimo i testiramo
  - odaberemo najbolju vrijednost i napravimo završni test na skupu za testiranje

Podešavanje na skupu za provjeru

- imamo dvije vrste nepoznanica
  - parametre: vjerojatnosti $P(X|Y)$ i $P(Y)$
  - hiperparametre: količina/tip zaglađivanja, k, α
- odakle što naučiti?
  - parametre učimo iz skupa za učenje
  - hiperparametre podešavamo na skupu za provjeru
  - za svaku vrijednost hiperparametara naučimo i testiramo
  - odaberemo najbolju vrijednost i napravimo završni test na skupu za testiranje

Podešavanje na skupu za provjeru

- imamo dvije vrste nepoznanica
  - parametre: vjerojatnosti $P(X|Y)$ i $P(Y)$
  - hiperparametre: količina/tip zaglađivanja, k, α
- odakle što naučiti?
  - parametre učimo iz skupa za učenje
  - hiperparametre podešavamo na skupu za provjeru
  - za svaku vrijednost hiperparametara naučimo i testiramo
  - odaberemo najbolju vrijednost i napravimo završni test na skupu za testiranje

1. Naivan Bayesov klasifikator

2. Perceptron

Linearni klasifikatori

- zadani su nam vektori značajki: svakom primjeru x pridružujemo vektor $f(x)$, na osnovu kojeg trebamo odrediti klasu kojoj primjer pripada
- svaka značajka ima određenu težinu (ne nužno pozitivnu)
- računamo aktivaciju kao težinsku sumu značajki:

$$\text{aktivacija}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- ako je aktivacija:
  - nenegativna, izlaz je +1
  - negativna, izlaz je -1

Linearni klasifikatori

- zadani su nam vektori značajki: svakom primjeru x pridružujemo vektor $f(x)$, na osnovu kojeg trebamo odrediti klasu kojoj primjer pripada
- svaka značajka ima određenu težinu (ne nužno pozitivnu)
- računamo aktivaciju kao težinsku sumu značajki:

$$\text{aktivacija}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- ako je aktivacija:
  - nenegativna, izlaz je +1
  - negativna, izlaz je -1

Linearni klasifikatori

- zadani su nam vektori značajki: svakom primjeru x pridružujemo vektor $f(x)$, na osnovu kojeg trebamo odrediti klasu kojoj primjer pripada
- svaka značajka ima određenu težinu (ne nužno pozitivnu)
- računamo aktivaciju kao težinsku sumu značajki:

$$\text{aktivacija}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- ako je aktivacija:
  - nenegativna, izlaz je +1
  - negativna, izlaz je -1

Linearni klasifikatori

- zadani su nam vektori značajki: svakom primjeru x pridružujemo vektor $f(x)$, na osnovu kojeg trebamo odrediti klasu kojoj primjer pripada
- svaka značajka ima određenu težinu (ne nužno pozitivnu)
- računamo aktivaciju kao težinsku sumu značajki:

$$\text{aktivacija}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- ako je aktivacija:
  - nenegativna, izlaz je +1
  - negativna, izlaz je -1

- u binarnom slučaju: uspoređuju se značajke s vektorom težina
- učenje: odrediti vektor težina s obzirom na primjere

Pravilo odluke

- u prostoru vektora značajki:
  - primjeri su točke
  - svaki vektor težina je hiperravnina
  - jedna strana odgovara vrijednosti $Y = +1$
  - druga strana odgovara vrijednosti $Y = -1$

- u binarnom slučaju: uspoređuju se značajke s vektorom težina
- učenje: odrediti vektor težina s obzirom na primjere

Pravilo odluke

- u prostoru vektora značajki:
  - primjeri su točke
  - svaki vektor težina je hiperravnina
  - jedna strana odgovara vrijednosti $Y = +1$
  - druga strana odgovara vrijednosti $Y = -1$

Korigiranje težina: binarni perceptron

- inicijaliziraj sve težine na 0
- za svaki primjer iz skupa za učenje:
  - klasificiraj s obzirom na trenutne težine

$$y = \begin{cases} +1, & \text{ako je } w \cdot f(x) \geq 0 \\ -1, & \text{ako je } w \cdot f(x) < 0 \end{cases}$$

  - ako je primjer dobro svrstan (tj. $y = y^*$), nema promjene
  - ako je primjer krivo svrstan, popravi vektor težina

$$w = w + y^* \cdot f(x)$$

Višeklasno pravilo odluke

- ako imamo više klasa tada:
  - svakoj klasi y pridružujemo vektor težina w_y
  - izračunamo aktivaciju klase y : $w_y \cdot f(x)$
  - klasificiramo u klasu s najvećom aktivacijom

$$y = \arg\max_{y} w_y \cdot f(x)$$

Učenje: višeklasni perceptron

- inicijaliziraj sve težine na 0
- za svaki primjer iz skupa za učenje:
  - klasificiraj s obzirom na trenutne težine

$$y = \arg\max_y w_y \cdot f(x)$$

  - ako je primjer dobro svrstan (tj. $y = y^*$), nema promjene
  - ako je primjer krivo svrstan, popravi vektore težina

$$w_y = w_y - f(x)$$

$$w_{y^*} = w_{y^*} + f(x)$$

- odvojivost / separabilnost: ukoliko postoje parametri koji cijeli skup za učenje pravilno klasificiraju
- konvergencija: ukoliko je skup za učenje separabilan, perceptron konvergira u binarnom slučaju
- u slučaju neodvojivih primjera, moguće je uzeti srednju vrijednost kao separator

Poboljšavanje perceptrona

- minimalno ispravljajuće ažuriranje:

$$w_y = w'_y - \tau f(x)$$

$$w_{y^*} = w'_{y^*} + \tau f(x)$$

$$\min_w \frac{1}{2} \sum_y \|w_y - w'_y\|^2$$

uz uvjet $w_{y^*} \cdot f \geq w_y \cdot f + 1$

$$\tau = \frac{(w'_y - w'_{y^*}) \cdot f + 1}{2 f \cdot f}$$

Poboljšavanje perceptrona

- minimalno ispravljajuće ažuriranje:

$$w_y = w'_y - \tau f(x)$$

$$w_{y^*} = w'_{y^*} + \tau f(x)$$

$$\min_w \frac{1}{2} \sum_y \|w_y - w'_y\|^2$$

uz uvjet $w_{y^*} \cdot f \geq w_y \cdot f + 1$

$$\tau = \frac{(w'_y - w'_{y^*}) \cdot f + 1}{2 f \cdot f}$$

Poboljšavanje perceptrona

- ponekad je ova vrijednost prevelika, pa se može uvesti maksimalna veličina koraka C i τ tada definiramo kao

$$\tau = \min \left\{ \frac{(w'_y - w'_{y*}) \cdot f + 1}{2f \cdot f}, C \right\}$$

- metoda potpornih vektor: maksimizirati margine (ima smisla intuitivno, teoretski i praktično)
- u obzir se uzimaju samo potporni vektori (primjeri najblilži hiperravnini), dok se ostali primjeri ignoriraju
- kao rezultat se dobije separator s maksimalnim marginama

$$\min_w \frac{1}{2} \sum_y \|w\|^2$$

uz uvjet $\forall i, y \;\; w_{y^*} \cdot f(x_i) \geq w_y \cdot f(x_i) + 1$

Poboljšavanje perceptrona

- ponekad je ova vrijednost prevelika, pa se može uvesti maksimalna veličina koraka C i τ tada definiramo kao

$$\tau = \min\left\{\frac{(w'_y - w'_{y^*}) \cdot f + 1}{2f \cdot f}, C\right\}$$

- metoda potpornih vektor: maksimizirati margine (ima smisla intuitivno, teoretski i praktično)
- u obzir se uzimaju samo potporni vektori (primjeri najblilži hiperravnini), dok se ostali primjeri ignoriraju
- kao rezultat se dobije separator s maksimalnim marginama

$$\min_w \frac{1}{2} \sum_y \|w\|^2$$

uz uvjet $\forall i, y \;\; w_{y^*} \cdot f(x_i) \geq w_y \cdot f(x_i) + 1$

Poboljšavanje perceptrona

- ponekad je ova vrijednost prevelika, pa se može uvesti maksimalna veličina koraka C i τ tada definiramo kao

$$\tau = \min \left\{ \frac{(w'_y - w'_{y^*}) \cdot f + 1}{2f \cdot f}, C \right\}$$

- metoda potpornih vektor: maksimizirati margine (ima smisla intuitivno, teoretski i praktično)
- u obzir se uzimaju samo potporni vektori (primjeri najblilži hiperravnini), dok se ostali primjeri ignoriraju
- kao rezultat se dobije separator s maksimalnim marginama

$$\min_w \frac{1}{2} \sum_y \|w\|^2$$

uz uvjet $\forall i, y \;\; w_{y^*} \cdot f(x_i) \geq w_y \cdot f(x_i) + 1$