Uvod u nadgledano mašinsko učenje

Mladen Nikolić

Matematički fakultet
Univerzitet u Beogradu

O predavanju

- Glavni cilj predavanja je upoznavanje sa mašinskim učenjem, ali ne na naivnom nivou

O predavanju

- Glavni cilj predavanja je upoznavanje sa mašinskim učenjem, ali ne na naivnom nivou
- Dodatni cilj je sakupiti na jedno mesto neka znanja koja su obično rasuta po literaturi

O predavanju

- Glavni cilj predavanja je upoznavanje sa mašinskim učenjem, ali ne na naivnom nivou
- Dodatni cilj je sakupiti na jedno mesto neka znanja koja su obično rasuta po literaturi
- Bonus je pokazati da je matematika koja se uči na fakultetu izuzetno korisna!

O predavanju

- Glavni cilj predavanja je upoznavanje sa mašinskim učenjem, ali ne na naivnom nivou
- Dodatni cilj je sakupiti na jedno mesto neka znanja koja su obično rasuta po literaturi
- Bonus je pokazati da je matematika koja se uči na fakultetu izuzetno korisna!
- Biće teško

Zahvalnost

- Hvala kolegama Nemanji Mićoviću i Urošu Stegiću na višestrukoj proveri slajdova i implementaciji odabranih algoritama

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Mini anketa

- Ko ima elementarno razumevanje makar jednog algoritma mašinskog učenja?

Mini anketa

- Ko ima elementarno razumevanje makar jednog algoritma mašinskog učenja?
- Ko je primenio neki algoritam mašinskog učenja?

Mini anketa

- Ko ima elementarno razumevanje makar jednog algoritma mašinskog učenja?
- Ko je primenio neki algoritam mašinskog učenja?
- Ko je prilagodio neki algoritam mašinskog učenja ili razvio svoj?

Šta je mašinsko učenje?

- Sa praktičnog stanovišta: bavi se izgradnjom prilagodljivih računarskih sistema koji su sposobni da poboljšavaju svoje performanse koristeći informacije iz iskustva
- Sa teorijskog stanovišta: bavi se proučavanjem generalizacije i konstrukcijom i analizom algoritama koji generalizuju

Mesto u veštačkoj inteligenciji

- Ima istu ulogu u odnosu na induktivno zaključivanje koju logika i automatsko rezonovanje imaju u odnosu na deduktivno zaključivanje
- Pristupi veštačkoj inteligenciji zasnovani na logici zahtevaju definisanje problema pomoću formalno izraženih pravila
- Često se odnose na probleme koji su ljudima intelektualno teški
- Šta sa problemima koji su nam laki, a koje ne znamo da opišemo, poput prepoznavanja lica?
- Binarne kategorije tačnog i netačnog nisu uvek dovoljno dobre zbog neizvesnosti
- Verovatnoća je okvir rezonovanja koji prirodno uključuje neizvesnost
- Mašinsko učenje pruža odgovor na oba problema

Zašto je zanimljivo?

- Duboka teorija indukcije, odnosno generalizacije
- Brojne primene, a postignuti rezultati pomeraju granice tehnologije i mašte
- Spoj statistike, optimizacije i računarstva, ali se oslanja i na linearnu algebru, teoriju grafova, funkcionalnu analizu i na druge matematičke oblasti
- Sve je popularnije i u Srbiji
- Zbog toga je idealna oblast za studente Matematičkog fakulteta

Kako se razlikuje od istraživanja podataka?

- Istraživanje podataka (eng. data mining) se bavi pronalaženjem zakonitosti u podacima i uključuje *primenu* algoritama mašinskog učenja kao jedan od čestih koraka
- Takođe, istraživanje podataka uključuje poslove poput preprocesiranja podataka, eksplorativne analize, vizualizacije...
- Mašinsko učenje se bavi dizajnom algoritama koji generalizuju i stoga mogu da vrše predviđanje, njihovim svojstvima i time kako ta svojstva zavise od elemanta dizajna algoritma

Kako se razlikuje od statistike?

- Ciljevi su vrlo bliski i ukoliko se oblast definiše prema cilju, mnogi se slažu da značajne razlike nema
- Razlikuju se poreklo, terminologija, ali i fokusi, što na strani mašinskog učenja uključuje:
  - Često odustajanje od interpretabilnosti zarad boljeg kvaliteta predviđanja, što vodi većem bogatstvu i složenosti modela nego u klasičnoj statistici
  - Jači akcenat na optimizaciji i algoritmici, uslovljen većom složenošću modela
  - Neretko zanemarivanje analize svojstava za koja statističari mare, poput npr. nepristrasnosti, mimimalnosti varijanse i drugih i fokusiranje na kvalitet predviđanja
  - Teoriju generalizacije[1]
- Zbrka bi bila izbegnuta ako bismo prestali da se trudimo da povučemo granice između oblasti

[1] Ova tema se podjednako uklapa i u statistiku (i zove se statistička teorija učenja), ali se tradicionalno smatra podoblašću mašinskog učenja

Kratka istorija (1)

- 1943 – Mekaloh i Pits formulišu threshold logic, preteču neuronskih mreža
- 1950 – Alen Tjuring razmišlja o mašinama koje uče
- 1952 – Artur Semjuel je napisao prvi program koji uči da igra dame
- 1957 – Frenk Rozenblat je napravio (u hardveru) perceptron
- 1963 – Vapnik i Červonenkis predlažu prvu varijantu metoda potpornih vektora
- 1967 – Kaver i Hart predlažu algoritam k najbližih suseda sa primenom u problemu trgovačkog putnika

Kratka istorija (2)

- 1969 – Marvin Minski i Sejmur Papert kritikuju perceptron, što usporava razvoj neuronskih mreža
- 1975 – Verbos formuliše algoritam propagacije unazad (eng. backpropagation) za izračunavanje gradijenta neuronske mreže
- 1981 – Dedžong uvodi učenje zasnovano na objašnjavanju kojim se omogućuje izvođenje pravila iz podataka
- 1985 – Sejnovski i Rozenberg prave sistem koji uči da izgovara engleske reči
- 1992 – Bozer, Gijon i Vapnik predlažu upotrebu kernela sa metodom potpornih vektora što čini da ovaj metod dominira oblašću tokom devedesetih

Kratka istorija (3)

- 1992 – Tezauro pravi TD-Gammon, sistem koji igra igru tavla (eng. backgammon)
- 2006 – Hinton uvodi izraz *duboko učenje* (eng. deep learning) za algoritme za trening višeslojnih neuronskih mreža koje nadalje dominiraju oblašću
- 2011 – IBM-ov sistem Watson pobeđuje ranije prvake u kvizu Jeopardy!
- 2012 – Google X razvija sistem koji je u stanju da sam pregleda video zapise na YouTube-u i prepoznaje mačke!!
- 2016 – Guglov sistem Alfa Go pobeđuje svetskog prvaka u igri go rezultatom 4:1

Koliko je popularno?

- Jedna od glavnih i trenutno najpopularnija oblast veštačke inteligencije
- Trenutno među najpopularnijim oblastima računarstva i primenjene matematike
- Intenzivno se proučava u akademskim krugovima i primenjuje u industriji

Koji univerziteti imaju ovakav kurs?

- Stanford (1)
- Harvard (2)
- Princeton (3)
- Berkeley (2)
- Carnegie Mellon (master program)
- Washington (3)
- Illinois Urbana-Champaigne (1)
- Cornell (3)
- ...
- Columbia (master program)
- Oxford (2)
- Cambridge (master program)
- EPFL (4)
- ETH Zurich (1)
- Edinburgh (2)
- Imperial College London (1)
- TU Munich (2)

Koje svetske firme primenjuju mašinsko učenje?

- Google
- Google Deep Mind
- Yahoo
- Microsoft
- IBM
- Facebook
- Twitter
- Apple
- Samsung
- HP
- Oracle
- Fujitsu
- Hittachi
- NEC
- Ericsson
- Siemens
- SAP
- Locheed Martin
- Huawei
- DELL
- Bell Labs
- CISCO
- Nokia
- ...

Koliki je obim istraživanja?

- Broj radova na konferencijama u 2015.

Oblast	Konferencija	Primljeno	Objavljeno
MU	NIPS	1838	403
MU	ICML	1037	270
O1	K1	85	36
O1	K2	70	30
O1	K3	54	27
O2	K1	241	42
O2	K2	170	36

Gde se primenjuje?

- Bioinformatika
- Interfejsi mozga sa mašinama
- Hemijska informatika
- Računarsko opažanje
- Detekcija prevara sa kreditnim karticama
- Računarske igre
- Pretraživanje informacija
- Marketing
- Medicinska dijagnostika
- Obrada prirodnog jezika
- Onlajn reklamiranje
- Sistemi za preporučivanje
- Upravljanje robotima
- Prepoznavanje govora
- Prepoznavanje rukom pisanog teksta
- Praćenje zdravstvenog stanja
- Ekonomija
- Analiza društvenih mreža

Šta je ključ uspeha?

- Spoj ozbiljne teorije i važnih praktičnih problema
- Vandredno jaka povratna sprega između nauke i industrije

Analiza slika i videa

- Prepoznavanje lica
- Prepoznavanje objekata na slikama i u videu
- Rekonstrukcija trodimenzionalne informacije iz videa

Autonomna vožnja

- ALVINN je vozio 140km na autoputu krajem osamdesetih bez ljudske pomoći
- Google X je razvio nov sistem namenjen gradskoj vožnji
- Google X je razvio sistem za upravljanje kvadrotorima

Igranje igara

- Devedesetih godina je napravljen sistem koji igra Backgammon u rangu svetskih šampiona
- Alfa Go je 2016. pobedio svetskog šampiona u igri go 4 naprema 1
- Neuronska mreža uspešno igra igre sa Atarija
- Neuronska mreža igra Doom bolje od čoveka

Obrada prirodnog jezika i govora

- Optičko prepoznavanje karaktera i prepoznavanje rukom pisanog teksta
- Sistemi za razgovor i preporuke korisnicima
- Analiza osećanja izraženih u tekstu
- Parsiranje rečenica
- Klasifikacija teksta
- Mašinsko prevođenje
- Automatsko opisivanje slika

Primene u medicini

- Prepoznavanje tumora na snimcima različitih skenera
- Predviđanje budućeg stanja pacijenata
- Određivanje terapije za sepsu

Analiza društvenih mreža

- Otkrivanje povezanosti u terorističkim i kriminalnim mrežama
- Ciljano reklamiranje

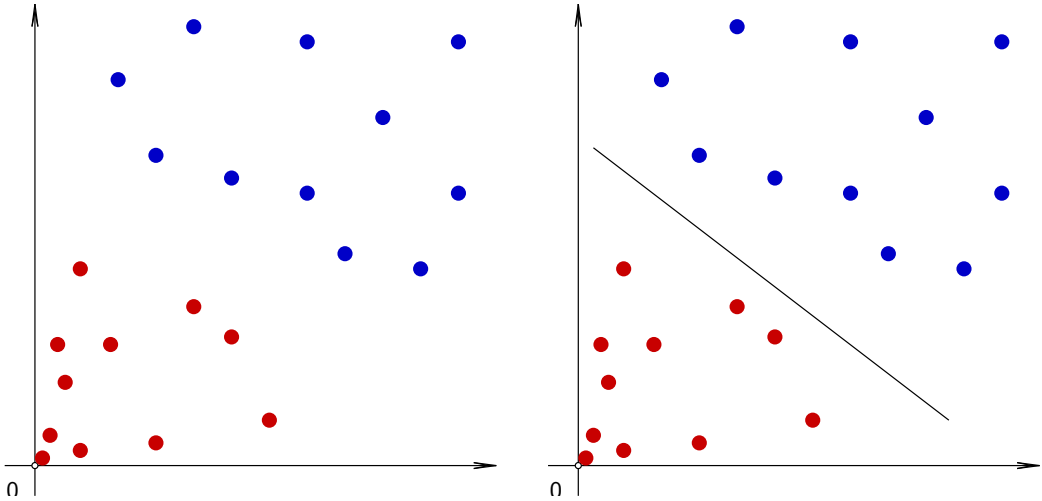
Prepoznavanje računarskih članaka

- Potrebno je napraviti sistem koji automatski prepoznaje računarske članke kako bi ih preporučio klijentima
- Kako ih prepoznati?

Prepoznavanje računarskih članaka

- Potrebno je napraviti sistem koji automatski prepoznaje računarske članke kako bi ih preporučio klijentima
- Kako ih prepoznati?
- Recimo po stručnim terminima
- Na primer ,,računar“ i ,,datoteka“
- Svaki članak se predstavlja frekvencijama ovih reči
- Radi se o tačkama u ravni
- Potrebno je naći način da se u tom prostoru identifikuje granica između računarskih članaka i ostalih

Prepoznavanje računarskih članaka



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Vrste mašinskog učenja

- Nadgledano učenje (eng. supervised learning)
- Nenadgledano učenje (eng. unsupervised learning)
- Učenje uslovljavanjem (eng. reinforcement learning)

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Ishodi i događaji

- Eksperimenti imaju ishode
  - Ishodi bacanja kockice su brojevi od 1 do 6
- Događaji su skupovi ishoda
  - Događaj može biti da je dobijen broj veći od 3, što je skup ishoda $\{4, 5, 6\}$
- Kažemo da se neki događaj desio ako se desio neki ishod iz tog događaja

Verovatnoća događaja

- Verovatnoće su dugoročne frekvencije događaja
- Postoje drugačije interpretacije pojma verovatnoće (npr. Bajesova)
- Formalno, govorimo o verovatnosnoj meri P koja mora da zadovolji sledeće aksiome:
  - $P(\Omega) = 1$, gde je Ω skup svih ishoda
  - $P(A) \geq 0$, za svaki događaj $A \subseteq \Omega$
  - $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$ ako su A_i disjunktni događaji

Uslovna verovatnoća

- $P(A|B)$ je verovatnoća događaja A pri uslovu B i definiše se kao

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- Ako znamo da je broj koji je kockica dala neparan, koja je verovantoća da je to baš 3?

$$P(\{3\}|\{1,3,5\}) = \frac{P(\{3\})}{P(\{1,3,5\})} = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3}$$

Nezavisnost događaja

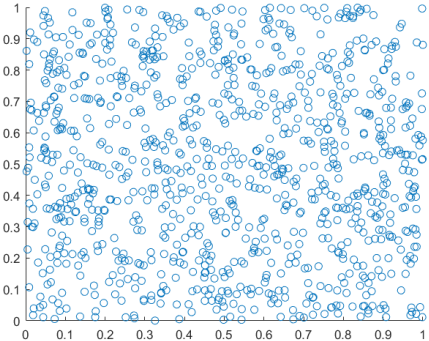
- Događaji A i B su nezavisni ako važi

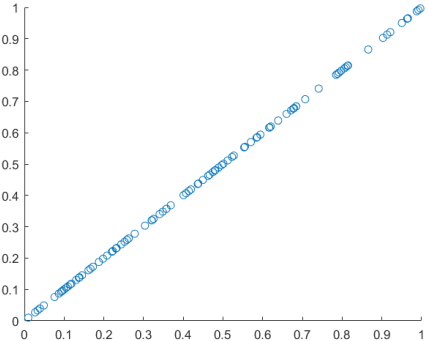
$$P(A|B) = P(A)$$

odnosno

$$P(A \cap B) = P(A)P(B)$$

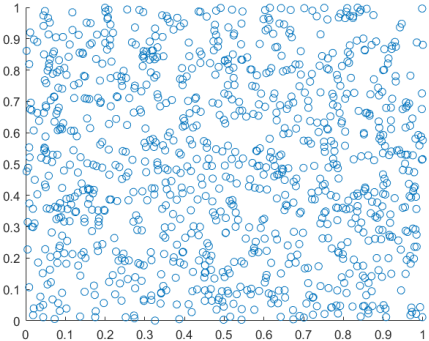
Nezavisnost događaja

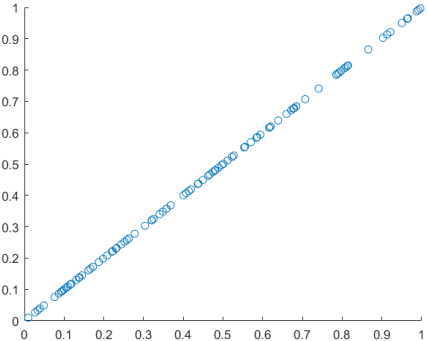




- Posmatrajmo događaje $x > 0.5$, $y > 0.5$ i $x > 0.5 \wedge y > 0.5$
- Koji slučaj odražava zavisne, a koji nezavisne događaje?

Nezavisnost događaja





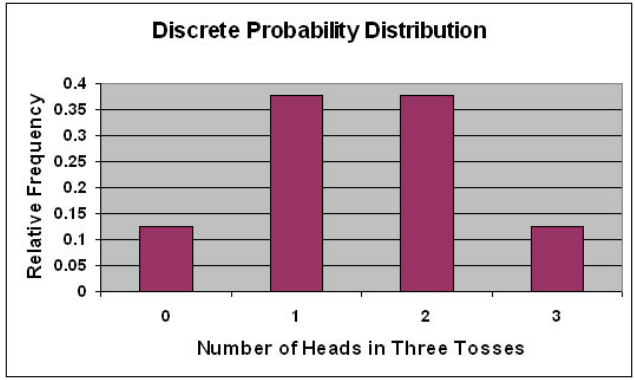
- Posmatrajmo događaje $x > 0.5$, $y > 0.5$ i $x > 0.5 \wedge y > 0.5$
- Koji slučaj odražava zavisne, a koji nezavisne događaje?
- U prvom se vidi nezavisnost, a verovatnoće su 0.5, 0.5 i 0.25
- U drugom se vidi zavisnost, a verovatnoće su 0.5, 0.5 i 0.5

Slučajna promenljiva

- Funkcije koje preslikavaju ishode u skup realnih brojeva nazivaju se slučajnim veličinama
- Na primer visina slučajno izabrane osobe
- Opisujemo ih pomoću pojma raspodele verovatnoće

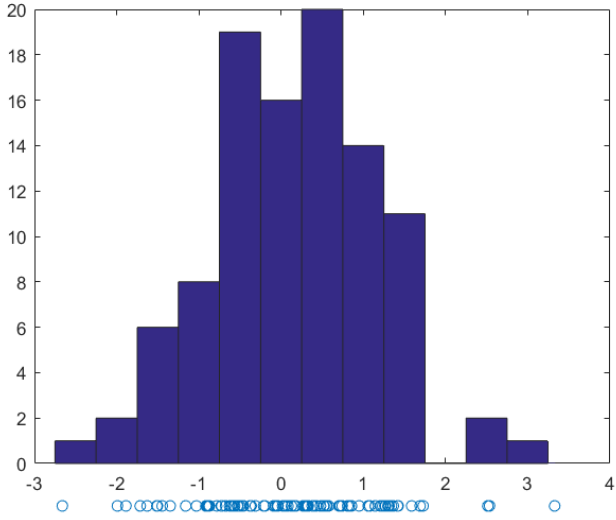
Diskretna raspodela verovatnoće

- Pridružuje verovatnoću svakom od prebrojivo mnogo vrednosti slučajne promenljive

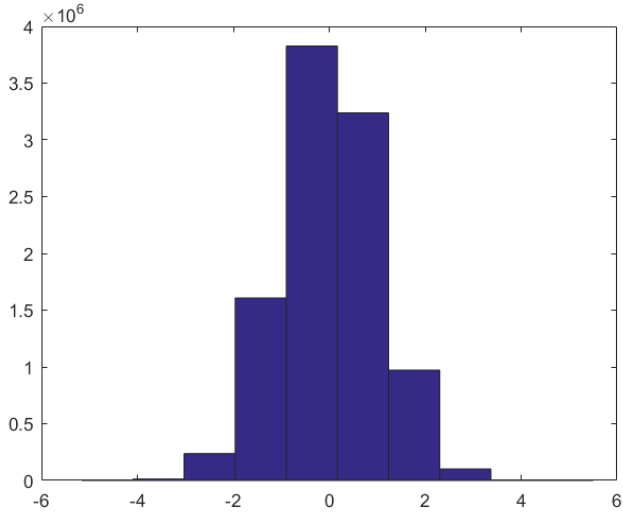


Slika: http://www2.cedarcrest.edu/academic/bio/hale/biostat/session10links/probgraph.html

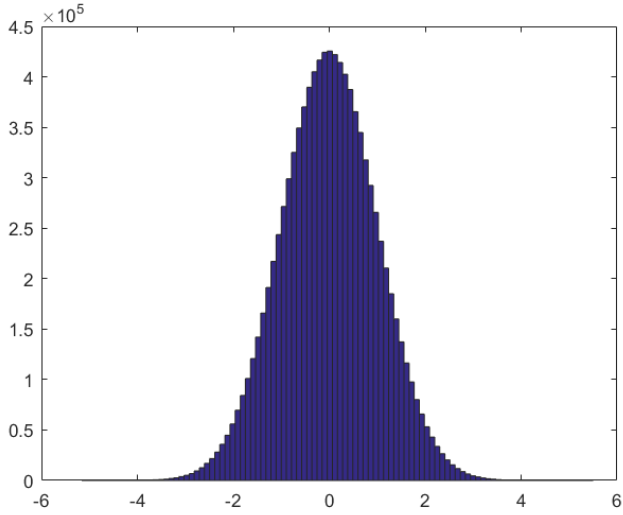
Histogram i neprekidna raspodela verovatnoće (1)



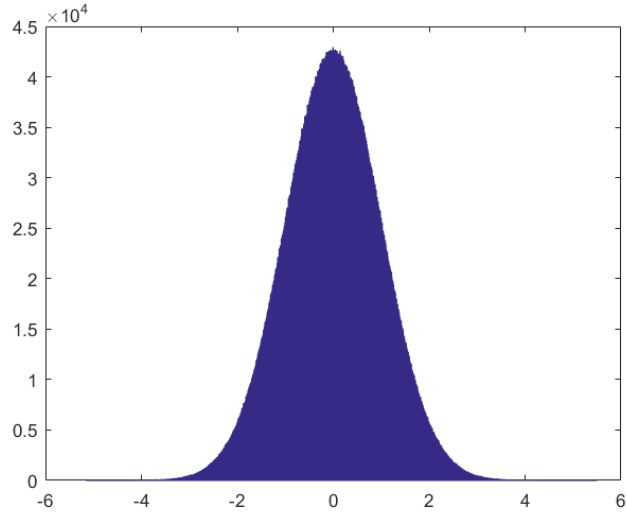
Histogram i neprekidna raspodela verovatnoće (2)



Histogram i neprekidna raspodela verovatnoće (3)



Histogram i neprekidna raspodela verovatnoće (4)



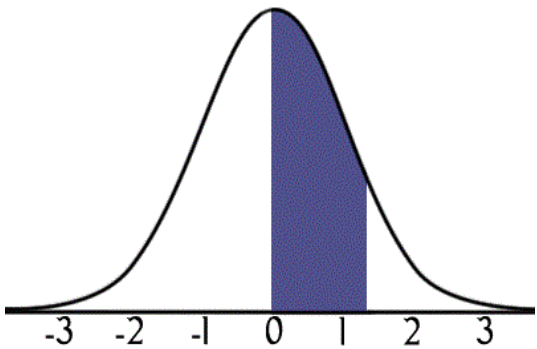
Gustina raspodele

- Histogram oslikava *gustinu raspodele*
- Mora da važi

$$\int p(x)dx = 1$$

Gustina raspodele

$$P([a,b]) = \int_a^b p(x)dx$$

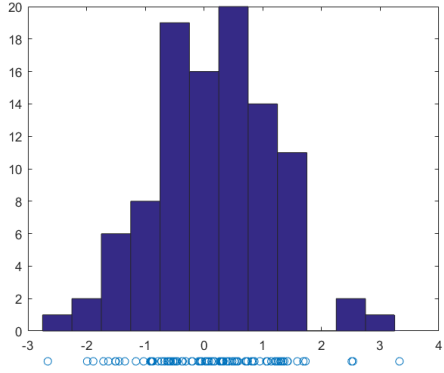


Slika: http://www.disfrutalasmatematicas.com/datos/distribucion-normal-estandar.html

Matematičko očekivanje

- Intuitivno, predstavlja srednju vrednost neke slučajne veličine
- Očekivanje slučajne veličine X se označava $\mathbb{E}[X]$
- Postoje i drugi načini da se definiše srednja vrednost (npr. medijana) koji mogu biti pogodniji u nekim kontekstima

Matematičko očekivanje



$$\mathbb{E}[X] \approx \sum_i x_i \frac{n_i}{N} = -2.5\frac{1}{100} - 2\frac{2}{100} - 1.5\frac{6}{100} - 1\frac{8}{100} - 0.5\frac{19}{100} + 0\frac{16}{100} + 0.5\frac{20}{100} + 1\frac{14}{100} + 1.5\frac{11}{100} + 2\frac{0}{100} + 2.5\frac{2}{100} + 3\frac{1}{100}$$

Matematičko očekivanje

- Za široku klasu raspodela važi (u praksi, praktično uvek):

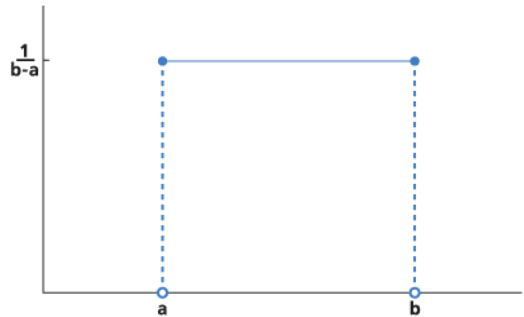
$$\mathbb{E}[f(X)] = \int f(x)p(x)dx$$

- Važi

$$\mathbb{E}[\alpha X + \beta Y] = \alpha\mathbb{E}[X] + \beta\mathbb{E}[Y]$$

Uniformna raspodela

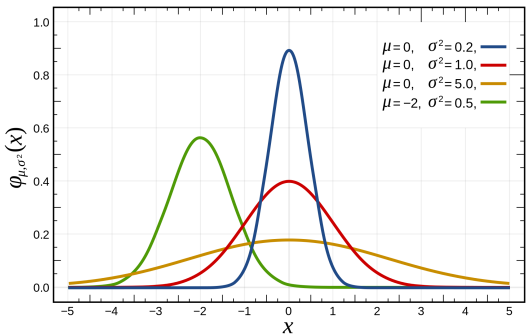
$$X \sim \mathcal{U}(a, b) \qquad p(x) = \frac{1}{b-a} \qquad \mathbb{E}[X] = \frac{a+b}{2} \qquad Var(X) = \frac{1}{12}(b-a)^2$$



Slika: http://pro.arcgis.com/en/pro-app/tool-reference/data-management/distributions-for-assigning-random-values.htm

Normalna raspodela

$$X \sim \mathcal{N}(\mu, \sigma) \qquad p(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \mathbb{E}[X] = \mu \qquad Var(X) = \sigma^2$$



Slika: https://en.wikipedia.org/wiki/Normal_distribution

Varijansa (disperzija)

- *Varijansa* je mera raspršenosti slučajne promenljive i definiše se kao

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

- Predstavlja očekivanje odstupanja slučajne promenljive od njenog očekivanja
- Koren varijanse se naziva *standardna devijacija*

Statistika

- Donosi zaključke o fenomenima na osnovu uzoraka iz iskustva
- Osnovni zadatak matematičke statistike je izabrati, iz skupa dopustivih verovatnosnih mera, jednu koja najbolje odgovara uzorku podataka
- Često se ovaj, a i drugi problemi statistike, svode na ocenu nekih nepoznatih parametara verovatnosne mere, na osnovu podataka
- Funkcija koja slika uzorak u realan broj i *ima jos neka lepa svojstva*, naziva se *statistika*

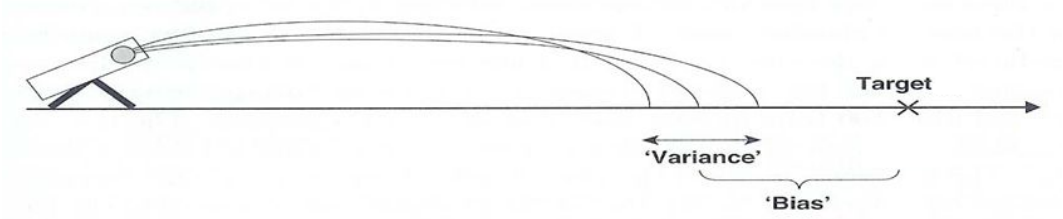
Poželjna svojstva statistika

- Neka je $\hat{\theta}$ ocena parametra θ
- Ocena $\hat{\theta}$ je *nepristrasna* (*centrirana*) ako važi

$$\mathbb{E}[\hat{\theta}] = \theta$$

- $\mathbb{E}[\hat{\theta}] - \theta$ je *sistematsko odstupanje* ocene $\hat{\theta}$
- Nepristrasna ocena koja ima manju varijansu je *bolja* od nepristrasne ocene koja ima veću varijansu

Ilustracija sistematske ocene i varijanse statistike



Slika: P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining. Modifikovano.

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Šta je nadgledano učenje?

- Potrebno je ustanoviti odnos između *atributa* x i *ciljne promenljive* y
- Problemi koje danas razmatramo su često previše kompleksni
- Stoga se pomenuti odnos često aproksimira na osnovu uzorka
- Zavisnosti između promenljivih se modeluju funkcijom koja se naziva *model*
- Model treba da *generalizuje*

Standardni problemi nadgledanog učenja

- Klasifikacija – ciljna promenljiva je kategorička
- Regresija – ciljna promenljiva je neprekidna

Osnovna teorijska postavka nadgledanog učenja

- Odnos između x i y je određen probabilističkim zakonom $p(x, y)$
- Potrebno je odrediti ,,najbolju“ funkciju f takvu da važi $y \approx f(x)$
- *Funkcija greške* (eng. loss) L kvantifikuje odstupanje između y i $f(x)$
- *Funkcional rizika* R formalizuje pojam ,,najboljeg“:

$$R(f) = \mathbb{E}[L(y, f(x))] = \int L(y, f(x))p(x, y)dxdy$$

- Potrebno je odrediti funkciju f za koju je vrednost $R(f)$ najmanja

Ali u praksi...

- Razmatranje svih mogućih modela nije izvodljivo, tako da se pretpostavlja *reprezentacija modela*
- Modeli $f_w(x)$ za neku reprezentaciju zaivise od vektora relanih vrednosti w, koje nazivamo *parametrima modela*
- Funkcional rizika se onda može predstaviti kao funkcija parametara w: $R(w) = R(f_w(x))$
- Problem je što raspodela $p(x, y)$ nije poznata, ali se pretpostavlja da postoji uzorak $\mathcal{D} = \{(x_i, y_i) \mid i = 1, \ldots, N\}$ takav da važi $(x_i, y_i) \sim p(x, y)$
- Potreban je praktičan princip indukcije

Minimizacija empirijskog rizika

- Empirijski rizik:

$$E(w, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_w(x_i))$$

- Kada nije naveden, argument $\mathcal{D}$ se podrazumeva
- Princip minimizacije empirijskog rizika (ERM):
  - funkciju koja minimizuje $E(w, \mathcal{D})$ uzeti za aproksimaciju funkcije koja minimizuje $R(w)$
- Umesto empirijski rizik, govorićemo prosečna greška ili samo greška

Pitanja vezana za ERM

- Da li bi trebalo da radi?

Pitanja vezana za ERM

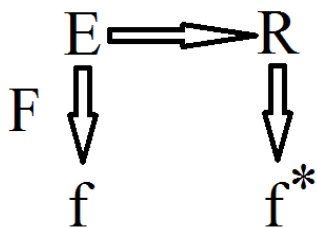
- Da li bi trebalo da radi?
- Valjda bi trebalo, pošto proseci konvergiraju očekivanjima

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Valjda bi trebalo, pošto proseci konvergiraju očekivanjima
- Ali ne aproksimiramo parametar raspodele na osnovu uzorka!

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Valjda bi trebalo, pošto proseci konvergiraju očekivanjima
- Ali ne aproksimiramo parametar raspodele na osnovu uzorka!
- Vršimo aproksimaciju funkcionala, a funkcije se dobijaju minimizacijom po beskonacnoj familiji funkcija!



Pitanja vezana za ERM

- Da li bi trebalo da radi?

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Odgovor bi trebalo da zavisi od svojstava skupa modela $\{f_w(x)\}$

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Odgovor bi trebalo da zavisi od svojstava skupa modela $\{f_w(x)\}$
- Koja je greška aproksimacije?

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Odgovor bi trebalo da zavisi od svojstava skupa modela $\{f_w(x)\}$
- Koja je greška aproksimacije?
- Nešto kasnije...

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Odgovor bi trebalo da zavisi od svojstava skupa modela $\{f_w(x)\}$
- Koja je greška aproksimacije?
- Nešto kasnije...
- Da li je konvergencija aproksimacije uniformna i nezavisna od aproksimirane funkcije?

Pitanja vezana za ERM

- Da li bi trebalo da radi?
- Odgovor bi trebalo da zavisi od svojstava skupa modela $\{f_w(x)\}$
- Koja je greška aproksimacije?
- Nešto kasnije...
- Da li je konvergencija aproksimacije uniformna i nezavisna od aproksimirane funkcije?
- Može biti, ali preskačemo...

ERM za klasifikaciju

- Šta treba da minimizujemo?

ERM za klasifikaciju

- Šta treba da minimizujemo?
- Broj grešaka na trening skupu

ERM za klasifikaciju

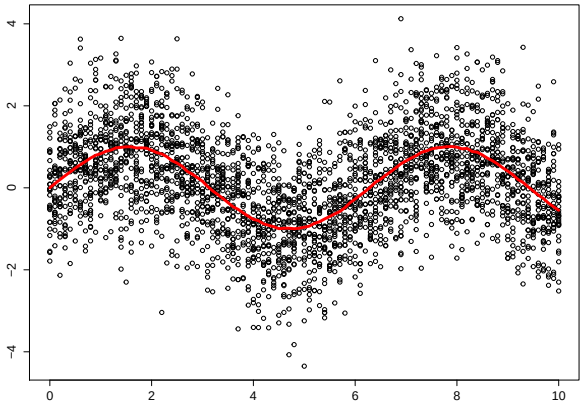
- Šta treba da minimizujemo?
- Broj grešaka na trening skupu
- Indikatorska funkcija:

$$I(F) = \begin{cases} 1 & \text{ako važi } F \\ 0 & \text{ako važi } \neg F \end{cases}$$

- Funkcija greške: $L(u, v) = I(u \neq v)$
- Optimizacioni problem:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq f_w(x_i))$$

Regresija



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Regresija

- Regresiona funkcija: $r(x) = \mathbb{E}(y|x) = \int y \; p(y|x)dy$
- Ako važi $r(x) = f_w(x)$ za neko w^*, tada se minimum rizika dostiže baš za w^*:

$$R(w) = \mathbb{E}[(y - f_w(x))^2]$$

- U opštem slučaju, minimum $R(w)$ se dostiže za funkciju najbližu[2] funkciji $r(x)$

[2] U odnosu na ℓ_2 normu

ERM za regresiju

- Funkcija greške:

$$L(u, v) = (u - v)^2$$

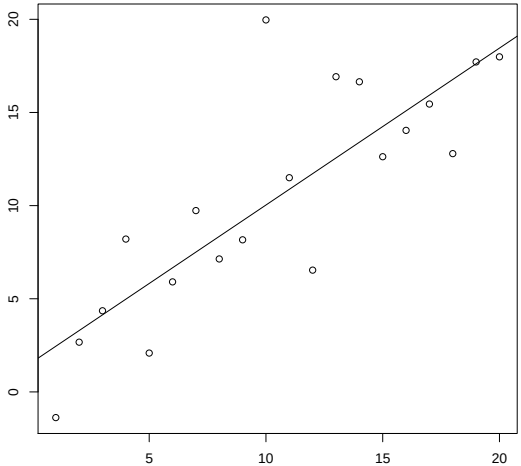
- Optimizacioni problem:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} (y_i - f_w(x_i))^2$$

Koliko dobro se model može prilagoditi podacima?

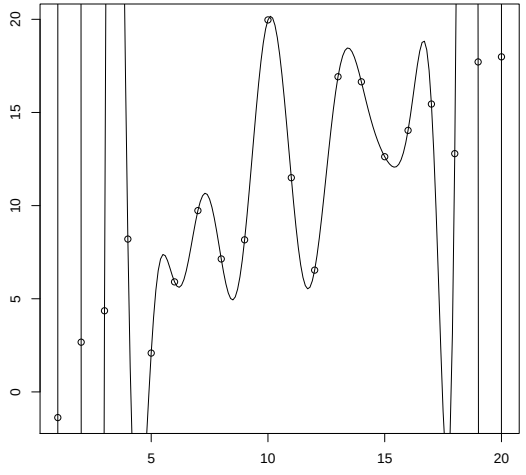
- Imajmo u vidu problem regresije
- Jednostavna linearna regresija: $f_w(x) = w_0 + w_1 x$
- Polinomijalna linearna regresija: $f_w(x) = \sum_{i=0}^{n} w_i x^i$

Jednostavna linearna regresija



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Polinomijalna linearna regresija



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Preprilagođavanje (eng. overfitting)

- Dobra prilagođenost modela trening podacima ne obezbeđuje dobru generalizaciju
- Uporediti sa učenjem napamet
- Uzrok problema je prevelika prilagodljivost modela
- Upravljanje prilagodljivošću modela je od ključnog značaja za dobru generalizaciju!
- Ovo je glavni problem mašinskog učenja i izvor njegove najdublje teorije

Kako učiniti modele manje prilagodljivim?

- Izabrati neprilagodljivu reprezentaciju (npr. linearni modeli)?

Kako učiniti modele manje prilagodljivim?

- Izabrati neprilagodljivu reprezentaciju (npr. linearni modeli)?
- Moguće, ali takav pristup je previše krut i nije podložan finom podešavanju

Kako učiniti modele manje prilagodljivim?

- Izabrati neprilagodljivu reprezentaciju (npr. linearni modeli)?
- Moguće, ali takav pristup je previše krut i nije podložan finom podešavanju
- Da li je moguće fino podešavati prilagodljivost modela nezavisno od izabrane reprezentacije?

Regularizacija (1)

- Minimizacija regularizovane greške:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_w(x_i)) + \lambda \Omega(w)$$

- Čest izbor *regularizacionog izraza* Ω je kvadrat ℓ_2 norme

$$\Omega(w) = \|w\|_2^2 = \sum_{i=1}^{n} w_i^2$$

Regularizacija (1)

- Minimizacija regularizovane greške:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_w(x_i)) + \lambda \Omega(w)$$

- Čest izbor *regularizacionog izraza* Ω je kvadrat ℓ_2 norme

$$\Omega(w) = \|w\|_2^2 = \sum_{i=1}^{n} w_i^2$$

- Regularizacioni izraz kažnjava visoke apsolutne vrednosti parametara, čineći model manje prilagodljivim
- *Regularizacioni parametar* λ služi za fino podešavanje prilagodljivosti modela

Regularizacija (2)

- U slučaju linearnih modela $f_w(x) = \sum_{i=1}^{n} w_i x_i$ važi

$$w = \nabla_x f_w(x)$$

- Koji je efekat regularizacije?

Regularizacija (2)

- U slučaju linearnih modela $f_w(x) = \sum_{i=1}^{n} w_i x_i$ važi

$$w = \nabla_x f_w(x)$$

- Koji je efekat regularizacije?
- Ograničavanje gradijenta ograničava brzinu promene funkcije
- U opštijem smislu, regularizacijom se naziva bilo koja modifikacija optimizacionog problema koja ograničava prilagodljivost modela i čini ga manje podložnim preprilagođavanju
- U još opštijem smislu, regularizacija je bilo kakva modifikacija matematičkog problema koja ga čini bolje uslovljenim

Primer regularizacije – klasifikacioni model

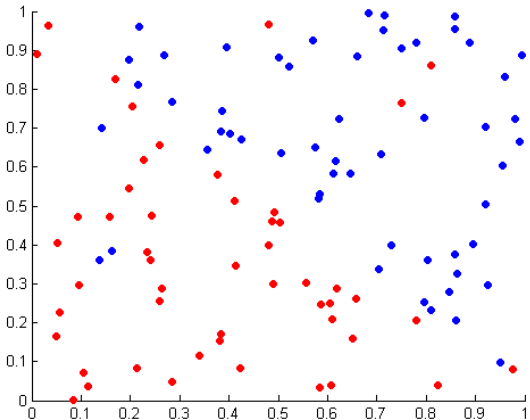
- Linearni klasifikacioni model:

$$f_w(x) = w_0 + w_1 x_1 + w_2 x_2$$

- Polinomijalni klasifikacioni model:

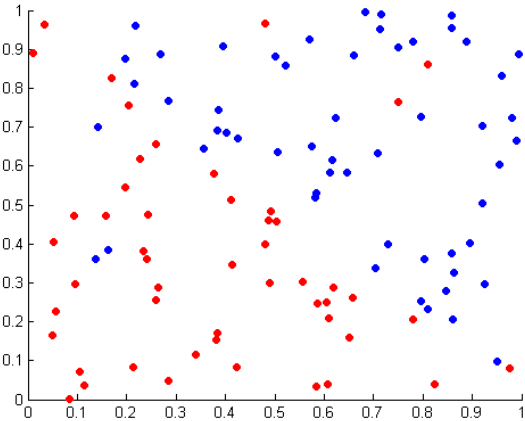
$$f_w(x) = \sum_{i=0}^{n} \sum_{j=0}^{i} w_{ij} x_1^j x_2^{i-j}$$

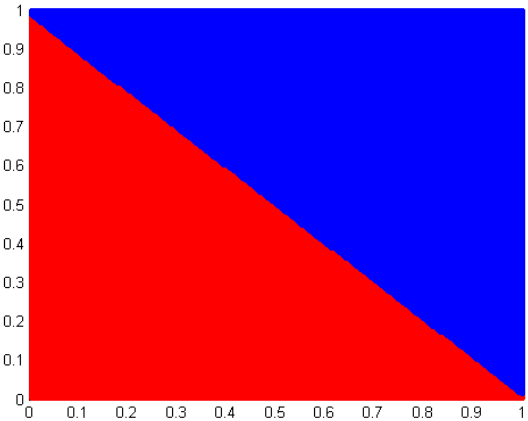
Primer regularizacije – podaci



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

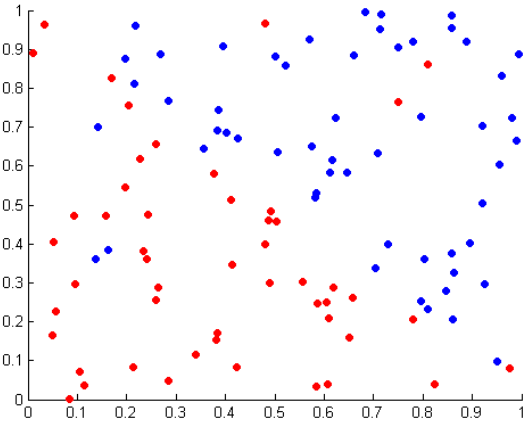
Primer regularizacije – linearni klasifikator

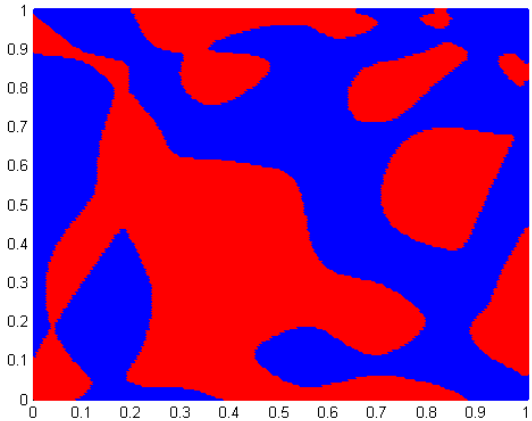




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

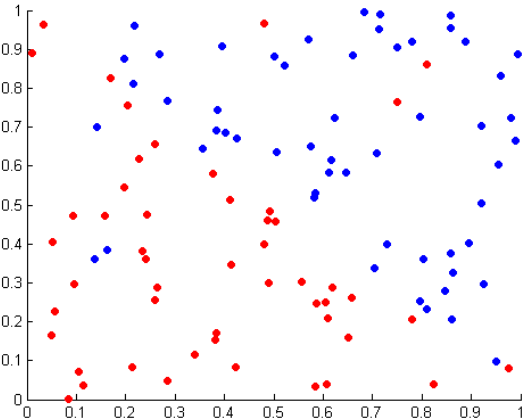
Primer regularizacije – polinomijalni klasifikator

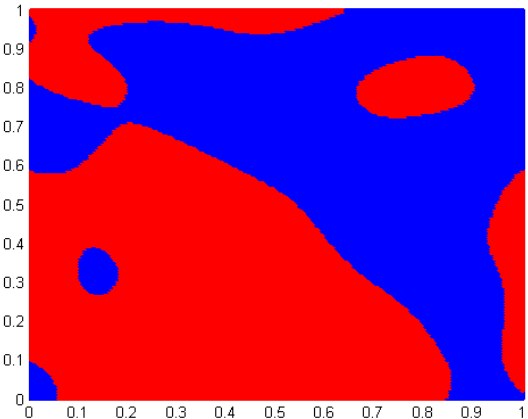




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

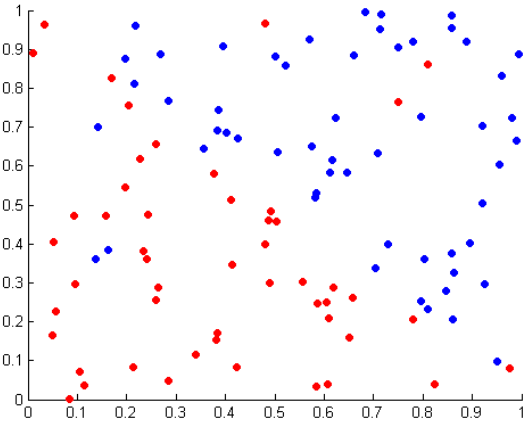
Primer regularizacije – regularizovani polinomijalni klasifikator

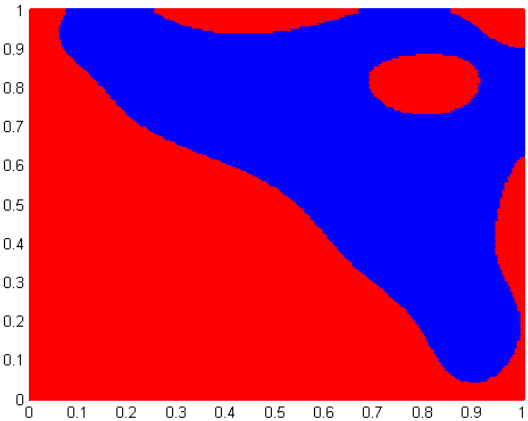




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

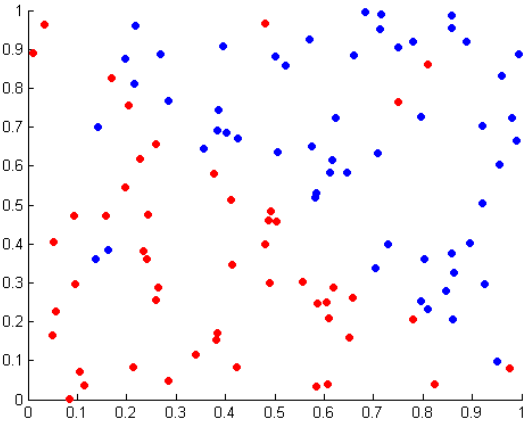
Primer regularizacije – regularizovani polinomijalni klasifikator

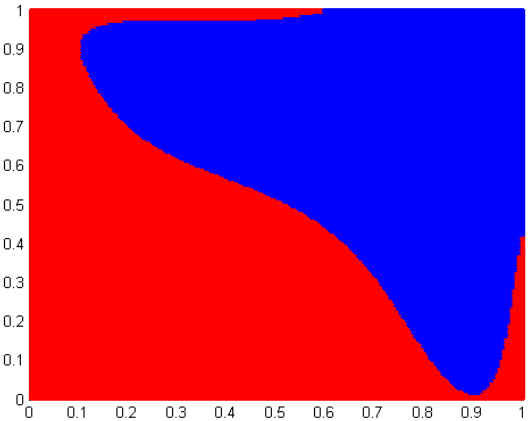




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

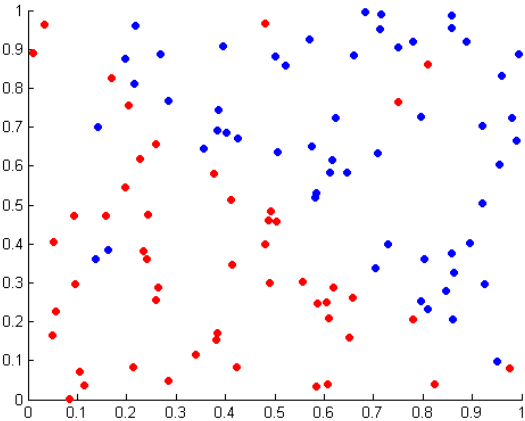
Primer regularizacije – regularizovani polinomijalni klasifikator

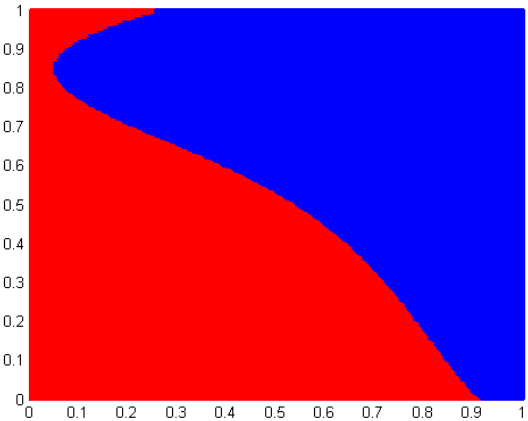




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

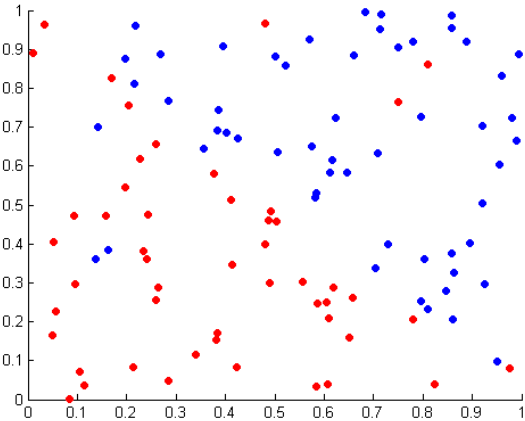
Primer regularizacije – regularizovani polinomijalni klasifikator

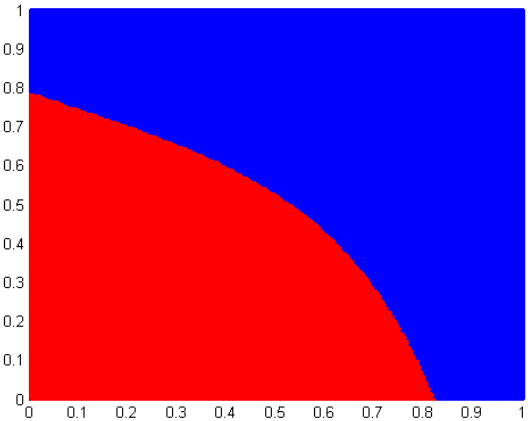




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

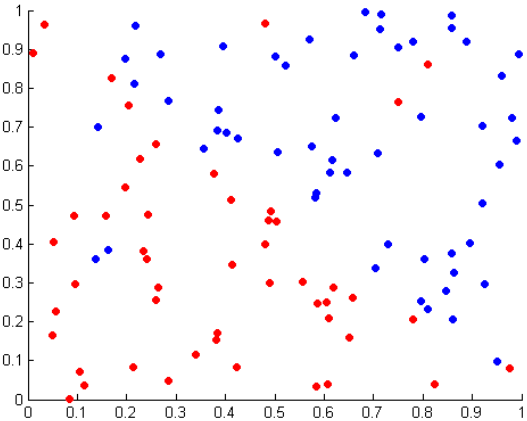
Primer regularizacije – regularizovani polinomijalni klasifikator

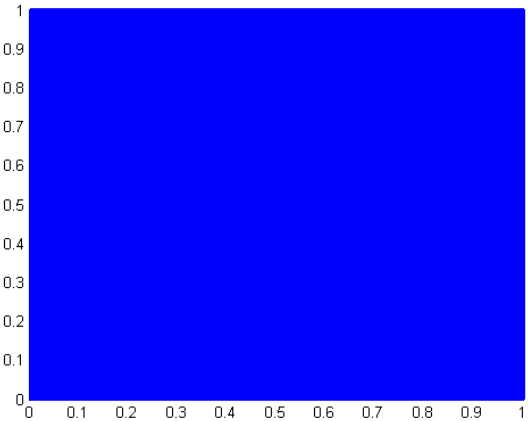




Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

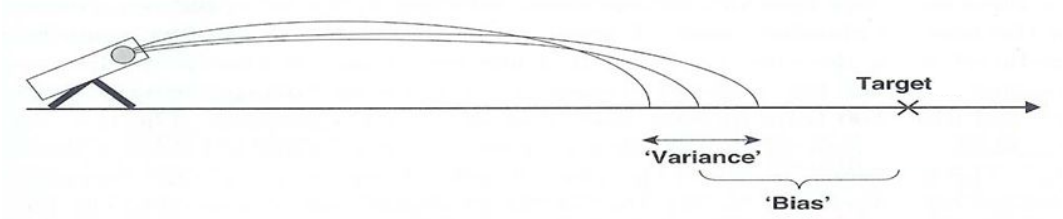
Primer regularizacije – regularizovani polinomijalni klasifikator





Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Sistematsko odstupanje i varijansa



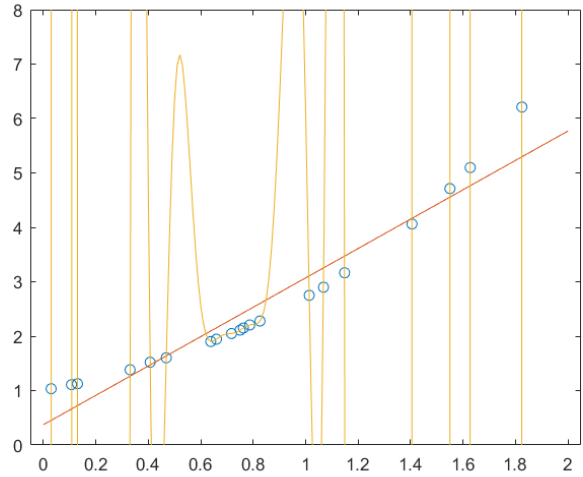
Slika: P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining. Modified.

Nagodba između sistematskog odstupanja i varijanse

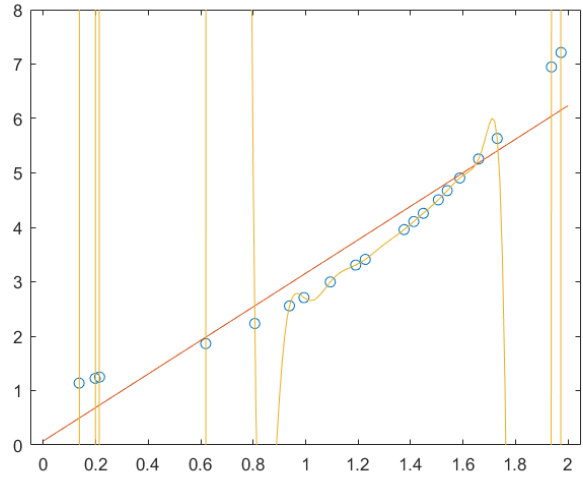
$$\begin{aligned}\mathbb{E}[(f_w(x) - r(x))^2] &= \mathbb{E}[(f_w(x) - \mathbb{E}[f_w(x)] + \mathbb{E}[f_w(x)] - r(x))^2] \\ &= \underbrace{(\mathbb{E}[f_w(x)] - r(x))^2}_{\textit{sistematsko odstupanje}^2} + \underbrace{\mathbb{E}[(f_w(x) - \mathbb{E}[f_w(x)])^2]}_{\textit{varijansa}}\end{aligned}$$

- gde je očekivanje po mogućim uzorcima podataka

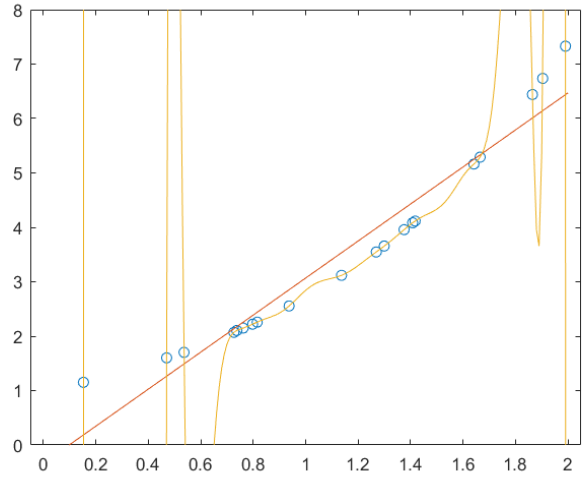
Sistematsko odstupanje i varijansa



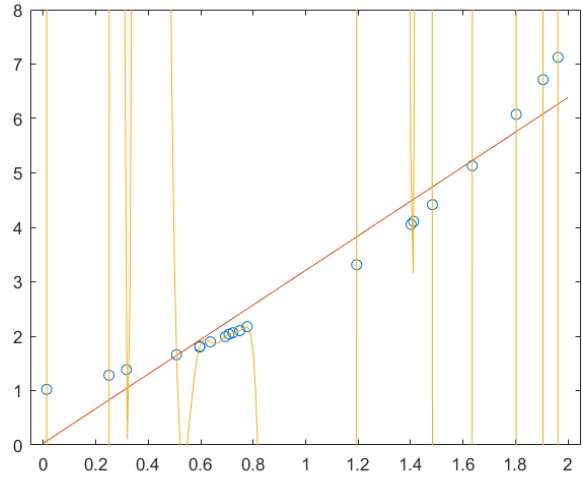
Sistematsko odstupanje i varijansa



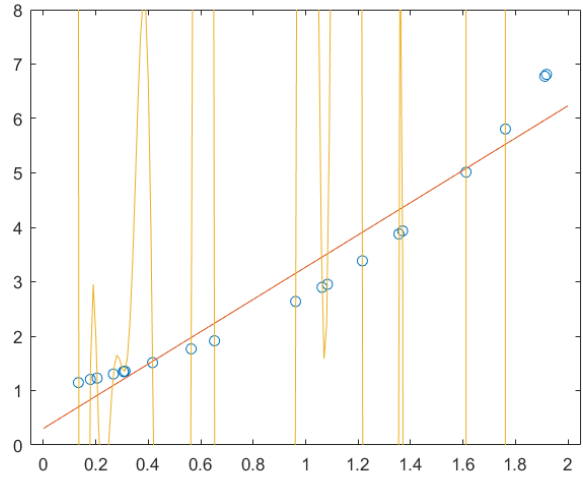
Sistematsko odstupanje i varijansa



Sistematsko odstupanje i varijansa



Sistematsko odstupanje i varijansa



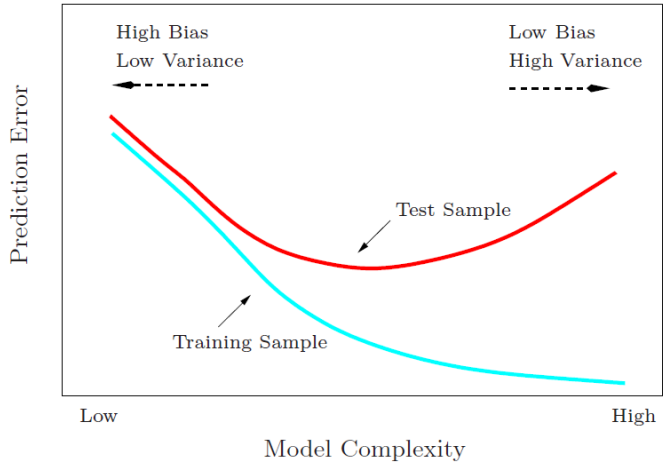
Nagodba između sistematskog odstupanja i varijanse i uslovljenost

- U slučaju fleksibilnih modela, naučeni model dramatično varira u zavisnosti od nebitnih promena u podacima
- To znači visoku varijansu predviđanja, kao i da je problem učenja loše uslovljen

Nagodba između sistematskog odstupanja i varijanse i regularizacija

- Regularizacijom se menja optimizacioni problem, tako što se smanjuje fleksibilnost modela
- Ovim se unosi sistematsko odstupanje u rešenje, ali se varijansa smanjuje više nego što se sistematsko odstupanje povećava!
- To objašnjava smanjenje greške u slučaju regularizovanih modela

Nagodba između sistematskog odstupanja i varijanse



Slika: T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning, 2001.

Prilagodljivost modela

- Razmatramo samo problem binarne klasifikacije
- U strogom smislu, ne analiziraju se pojedinačni modeli, već skup mogućih modela
- Kada kažemo da je reprezentacija modela složena ili prilagodljiva ili da je skup svih modela bogat?

Prilagodljivost modela

- Razmatramo samo problem binarne klasifikacije
- U strogom smislu, ne analiziraju se pojedinačni modeli, već skup mogućih modela
- Kada kažemo da je reprezentacija modela složena ili prilagodljiva ili da je skup svih modela bogat?
- Ako može da razlikuje sva različita obeležavanja podataka

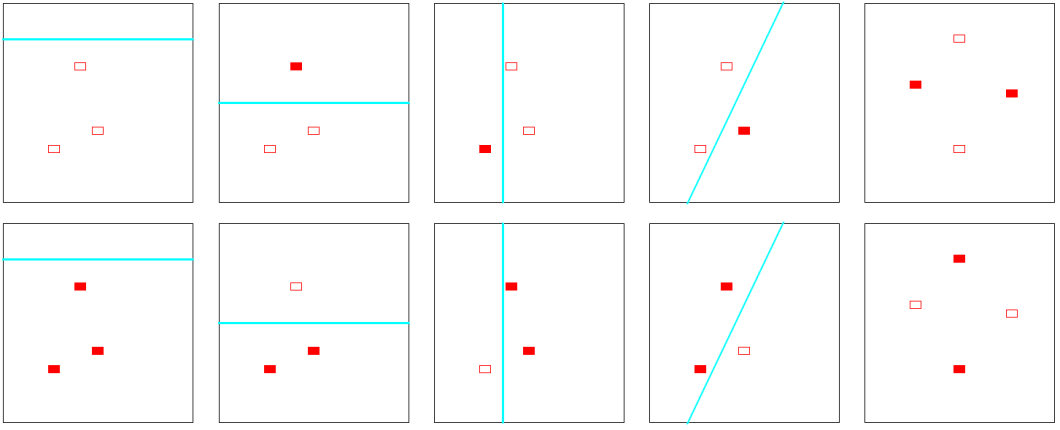
Prilagodljivost modela

- Razmatramo samo problem binarne klasifikacije
- U strogom smislu, ne analiziraju se pojedinačni modeli, već skup mogućih modela
- Kada kažemo da je reprezentacija modela složena ili prilagodljiva ili da je skup svih modela bogat?
- Ako može da razlikuje sva različita obeležavanja podataka
- Na koliko tačaka?

Prilagodljivost modela

- Razmatramo samo problem binarne klasifikacije
- U strogom smislu, ne analiziraju se pojedinačni modeli, već skup mogućih modela
- Kada kažemo da je reprezentacija modela složena ili prilagodljiva ili da je skup svih modela bogat?
- Ako može da razlikuje sva različita obeležavanja podataka
- Na koliko tačaka?
- Što je veći broj tačaka, veća je prilagodljivost

Prave razlikuju različita obeležavanja tačaka



Slika: T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning, 2001. Modifikovano.

VC dimenzija za indikatorske funkcije

- $\mathcal{F} = \{f \mid f : \mathbb{R}^n \to \{0, 1\}\}$
- $\mathcal{X} \subseteq \mathbb{R}^n$ označava skup tačaka
- $\mathcal{F}_{|\mathcal{X}}$ je skup restrikcija funkcija iz skupa $\mathcal{F}$ na domen $\mathcal{X}$
- $\mathcal{F}$ razbija $\mathcal{X}$ ukoliko važi $|\mathcal{F}_{|\mathcal{X}}| = 2^{|\mathcal{X}|}$
- Vapnik Červonenkisova (VC) dimenzija skupa $\mathcal{F}$ je najveći broj N takav da postoji skup $\mathcal{X}$ kardinalnosti N koji $\mathcal{F}$ razbija
- Ako takav broj ne postoji, onda kažemo da je VC dimenzija beskonačna

VC dimenzija – primer

- Kolika je VC dimenzija skupa svih linearnih modela sa n parametara

VC dimenzija – primer

- Kolika je VC dimenzija skupa svih linearnih modela sa n parametara
- Mogu se posmatrati kao hiperravni u $n-1$ dimenzionalnom prostoru

VC dimenzija – primer

- Kolika je VC dimenzija skupa svih linearnih modela sa n parametara
- Mogu se posmatrati kao hiperravni u $n-1$ dimenzionalnom prostoru
- Stoga je VC dimenzija n

VC dimenzija – primer

- Kolika je VC dimenzija skupa svih linearnih modela sa n parametara
- Mogu se posmatrati kao hiperravni u $n-1$ dimenzionalnom prostoru
- Stoga je VC dimenzija n
- Da li ovo važi za bilo koju vrstu modela?

VC dimenzija – primer

- Kolika je VC dimenzija skupa svih linearnih modela sa n parametara
- Mogu se posmatrati kao hiperravni u $n-1$ dimenzionalnom prostoru
- Stoga je VC dimenzija n
- Da li ovo važi za bilo koju vrstu modela?
- Kolika je VC dimenzija skupa $\{I(sin(wx) > 0)\}$?

VC dimenzija – primer

- Kolika je VC dimenzija skupa svih linearnih modela sa n parametara
- Mogu se posmatrati kao hiperravni u $n-1$ dimenzionalnom prostoru
- Stoga je VC dimenzija n
- Da li ovo važi za bilo koju vrstu modela?
- Kolika je VC dimenzija skupa $\{I(sin(wx) > 0)\}$?
- Beskonačna je!
- Za bilo koji broj N, mogu se izabrati tačke $x_i = 10^{-i}$, $i = 1, 2, \ldots, N$
- Za bilo koje oznake $y_1, y_2, \ldots, y_N \in \{-1, 1\}$, dovoljno je izabrati

$$w = \pi \left(1 + \sum_{i=1}^{N} \frac{(1 - y_i)10^i}{2} \right)$$

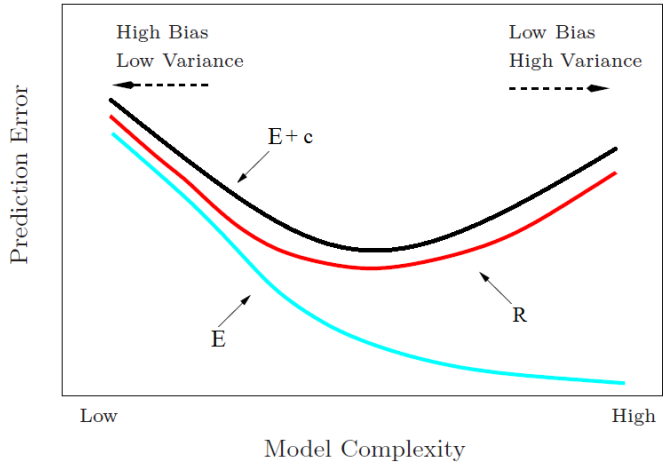
Granica rizika

- h je VC dimenzija skupa modela $\mathcal{F}$
- Sa verovatnoćom bar μ, važi

$$R(w) \leq E(w, \mathcal{D}) + c(h, |\mathcal{D}|, \mu)$$

  za sve modele iz skupa $\mathcal{F}$
- c opada kako $|\mathcal{D}|$ raste, a raste kako rastu h i μ

Granica rizika



Slika: T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning, 2001. Modifikovano.

Značaj VC dimenzije

- Kvantifikuje bogatstvo skupa modela i njegov potencijal da generalizuje!
- Nudi dubok uvid u to šta čini generalizaciju mogućom
- Ima duboke veze sa Poperovom filozofijom nauke

Značaj VC dimenzije

- Kvantifikuje bogatstvo skupa modela i njegov potencijal da generalizuje!
- Nudi dubok uvid u to šta čini generalizaciju mogućom
- Ima duboke veze sa Poperovom filozofijom nauke
- ...ali, u opštem slučaju, VC dimenziju nije lako izračunati

Treniranje i testiranje

- Kako se rizik ne može lako teorijski oceniti, obično se kvalitet naučenog modela proverava na podacima koji nisu učestvovali u izboru modela
- Proces izbora modela se naziva treningom, a skup podataka na osnovu kojih se taj izbor vrši, trening skupom
- Skup podataka na kojima se model evaluira se naziva test skupom
- Najjednostavniji pristup je odvojiti oko 70% svih podataka za treniranje, a 30% za testiranje
- Postoje i komplikovaniji (i pouzdaniji) pristupi evaluaciji modela

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

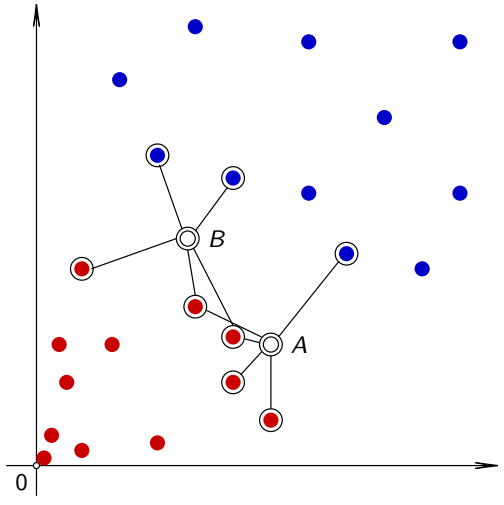
Procena kvaliteta i izbor modela

Finalni saveti

k najbližih suseda

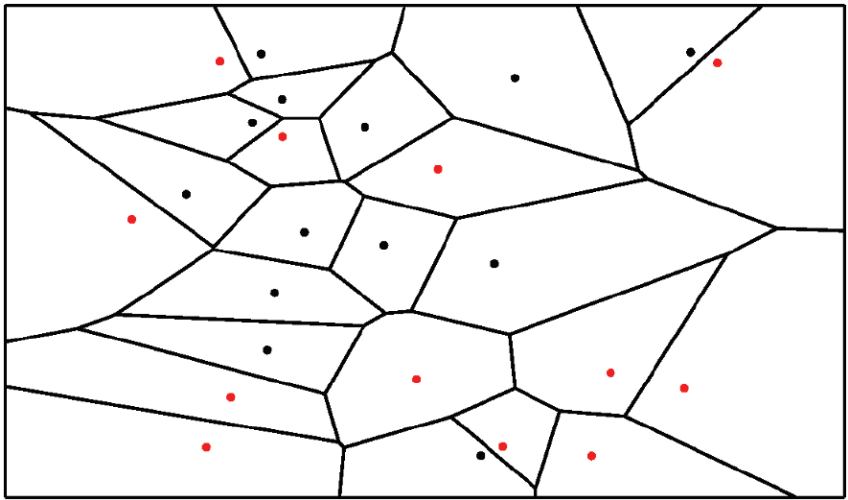
- Za instancu koju je potrebno klasifikovati, pronaći k najbližih instanci u trening skupu
- Instancu klasifikovati u najfrekventniju klasu među tim susedima
- Zahteva metriku (najčešće se koristi euklidska)

Stabilnost



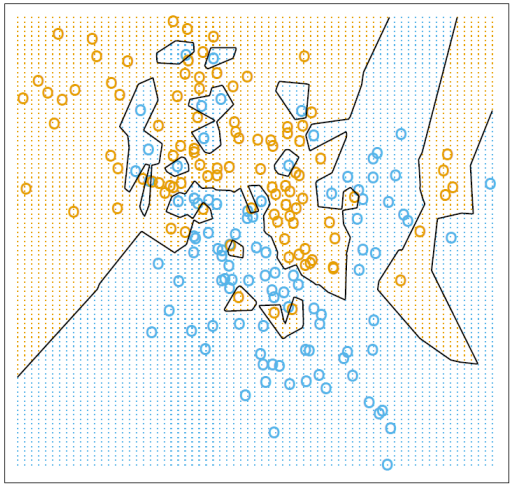
Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

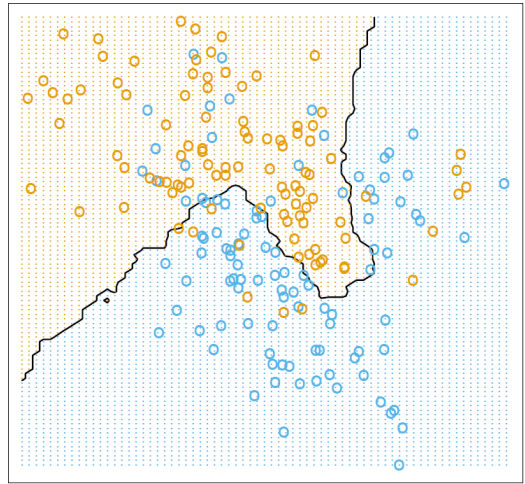
Granice između klasa



Slika: K. Murphy, Machine Learning, A Probabilistic Perspective, 2012.

Granice između klasa





Slika: T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning, 2001.

Primer

Hladnoća	Curenje iz nosa	Glavobolja	Groznica	Grip
Da	Ne	Blaga	Da	Ne
Da	Da	Ne	Ne	Da
Da	Ne	Jaka	Da	Da
Ne	Da	Blaga	Da	Da
Ne	Ne	Ne	Ne	Ne
Ne	Da	Jaka	Da	Da
Ne	Da	Jaka	Ne	Ne
Da	Da	Blaga	Da	Da

- Metrika:

$$d(u, v) = \sum_{i=1}^{n} I(u_i \neq v_i)$$

- Koja je klasa instance ($Da, Ne, Blaga, Ne$) ako se uzima u obzir 1 najbliži sused?

Primer

Hladnoća	Curenje iz nosa	Glavobolja	Groznica	Grip
Da	Ne	Blaga	Da	Ne
Da	Da	Ne	Ne	Da
Da	Ne	Jaka	Da	Da
Ne	Da	Blaga	Da	Da
Ne	Ne	Ne	Ne	Ne
Ne	Da	Jaka	Da	Da
Ne	Da	Jaka	Ne	Ne
Da	Da	Blaga	Da	Da

- Metrika:

$$d(u, v) = \sum_{i=1}^{n} I(u_i \neq v_i)$$

- Koja je klasa instance (Da, Ne, $Blaga$, Ne) ako se uzima u obzir 1 najbliži sused?
- Klasa je ,,Ne“, pošto je najbliži prvi primer iz tabele

VC dimenzija

- Koja je VC dimenzija algoritma jednog najbližeg suseda?

VC dimenzija

- Koja je VC dimenzija algoritma jednog najbližeg suseda?
- Beskonačna je!
- Svaki podatak iz trening skupa je svoj najbliži sused, pa ne postoji mogućnost greške

Kakav je kvalitet predviđanja?

- Neka je R stvarni rizik algoritma jednog najbližeg suseda
- Neka je R^* najmanji rizik koji se može postići u slučaju poznavanja tačnih raspodela M klasa
- Tada važi:

$$R^* \leq R \leq R^* \left(2 - \frac{M}{M-1} R^*\right)$$

- Zvuči ohrabrujuće, ali...

Kakav je kvalitet predviđanja?

- Neka je R stvarni rizik algoritma jednog najbližeg suseda
- Neka je R^* najmanji rizik koji se može postići u slučaju poznavanja tačnih raspodela M klasa
- Tada važi:

$$R^* \leq R \leq R^* \left(2 - \frac{M}{M-1} R^*\right)$$

- Zvuči ohrabrujuće, ali...
- ...ako imamo velike količine podataka

Neintuitivnost visokodimenzionalnih prostora

- Posmatrajmo n dimenzionalne kocke sa stranicom duzine 1 i stranicom dužine 0.99

$$\lim_{n\to\infty} \frac{V_{0.99}^n}{V_1^n} = \lim_{n\to\infty} \frac{0.99^n}{1^n} = 0$$

- Slično vazi i za druge oblike (npr. lopte)
- U visokodimenzionalnim prostorima praktično sve tačke lopte su vrlo daleko od centra
- U visokodimenzionalnim prostorima se distance ponašaju nešto drugačije nego što očekujemo
- Nekoliko radova ukazuje na malu varijaciju distanci u visokodimenzionalnim prostorima, što je nepovoljno za algoritam k najbližih suseda

Potrebna veličina trening skupa

- Posmatrajmo tačke raspoređene u okviru jedinične kocke
- Neka svaka tačka ima suseda na rastojanju, recimo 0.1
- Broj tačaka koji nam treba da bismo popunili n dimenzionalnu kocku je 11^d
- Sa porastom dimenzionalnosti, broj potrebnih tačaka eksponencijalno raste
- k najbližih suseda počiva na pretpostavci da će biti bliskih suseda
- Fenomen da je za adekvatno učenje potrebno eksponencijalno mnogo podataka u odnosu na dimenziju prostora se naziva *prokletstvom dimenzionalnosti*

Primena – algoritamski portfolio (1)

- Za rešavanje instance nekog problema, često je moguće upotrebiti različite algoritme
- U slučaju kombinatornih problema, često se dešava da su za različite instance problema pogodni različiti algoritmi
- Algoritamski portfolio se sastoji od skupa algoritama i mehanizma izbora jednog od algoritama za datu instancu problema (postoje i drugačije definicije)
- Jedan takav problem je problem zadovoljivosti iskaznih formula (SAT), koji ima brojne primene
- Implementacije algoritama koji ga rešavaju se nazivaju SAT rešavačima
- Kako izabrati SAT rešavač za datu ulaznu instancu?
- Neki od najuspešnijih pristupa izgradnji algoritamskog portfolija za SAT zasnivaju se na algoritmu k najbližih suseda

Primena – algoritamski portfolio (2)

- Na raspolaganju su vektori atributa i vremena rešavanja velikog broja instanci koje čine trening skup
- Za ulaznu instancu se računa vektor atributa i pronalazi se k najbližih suseda u trening skupu
- Pronalazi se SAT rešavač koji najefikasnije rešava tih k susednih instanci
- Taj rešavač se primenjuje na ulaznu instancu
- Značajno ubrzanje i pojednostavljenje u odnosu na dotadašnje pristupe

Prednosti i mane

- Prednosti:
  - Jednostavnost
  - Lokalnost – nije potrebno graditi model za ceo skup podataka u kojem možda ne važi isti trend
  - Proizvoljni oblici granica između klasa
- Mane:
  - Standardizacija je neophodna
  - Neotporan na ponovljene atribute
  - Prokletstvo dimenzionalnosti
  - Nedostatak interpretabilnosti

Naivni Bajesov klasifikator (1)

- Naivni Bajesov klasifikator se zasniva na primeni Bajesove formule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

- Za dato x, od svih ishoda y bira se onaj sa maksimalnom verovatnoćom $p(y|x)$
- Problem?

Naivni Bajesov klasifikator (1)

- Naivni Bajesov klasifikator se zasniva na primeni Bajesove formule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

- Za dato x, od svih ishoda y bira se onaj sa maksimalnom verovatnoćom $p(y|x)$
- Problem?
- Za ocenu verovatnoća $p(x|y)$ i $p(x)$, potrebno je eksponencijalno mnogo podataka u odnosu na broj atributa

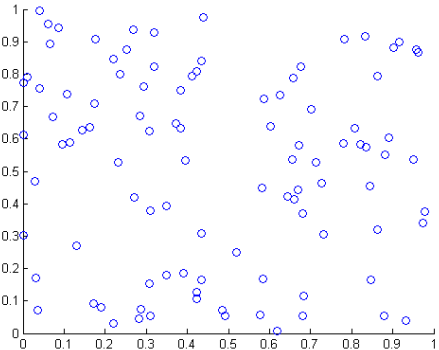
Naivni Bajesov klasifikator (2)

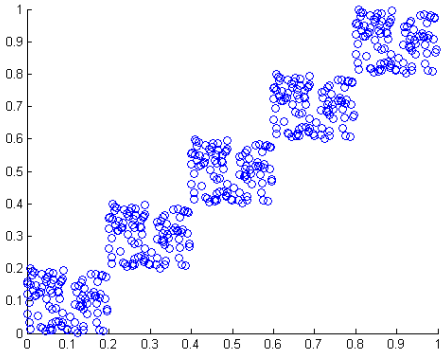
- ,,Naivna“ pretpostavka podrazumeva uslovnu nezavisnost atributa:

$$p(x|y) \approx \prod_{i=1}^{n} p(x_i|y)$$

- To je slabija pretpostavka od nezavisnosti atributa
- $p(x_i|y)$ i $p(y)$ se mogu lako oceniti iz podataka pošto su raspodele jedne promenljive
- $p(x)$ ne zavisi od y, pa se ni ne modeluje

Uslovna nezavisnost atributa – primer





Prednosti i mane naivnog Bajesovog klasifikatora

- Prednosti:
  - Efikasan trening i predviđanje
  - Nije osetljiv na prisustvo nebitnih atributa (imaju iste raspodele u svim klasama)
  - Ne zavisi od vrste atributa (kategorički ili kontinualni)
  - Lako se ažurira kako pristižu podaci
- Mane:
  - Pretpostavlja uslovnu nezavisnost atributa
  - Ako se neke vrednosti atributa ne pojavljuju u trening skupu, dodeljuje im verovatnoću 0, što nije realistično

Naivni Bajesov klasifikator – primer

Hladnoća	Curenje iz nosa	Glavobolja	Groznica	Grip
Da	Ne	Blaga	Da	Ne
Da	Da	Ne	Ne	Da
Da	Ne	Jaka	Da	Da
Ne	Da	Blaga	Da	Da
Ne	Ne	Ne	Ne	Ne
Ne	Da	Jaka	Da	Da
Ne	Da	Jaka	Ne	Ne
Da	Da	Blaga	Da	Da

$$p(Da|Da, Ne, Blaga, Ne) \sim p(Hla = Da|Da)p(Cur = Ne|Da)p(Gla = Blaga|Da)p(Gro = Ne|Da)p(Grip = Da)$$

$$= \frac{3}{5}\frac{1}{5}\frac{2}{5}\frac{1}{5}\frac{5}{8} = \frac{3}{500}$$

$$p(Ne|Da, Ne, Blaga, Ne) \sim p(Hla = Da|Ne)p(Cur = Ne|Ne)p(Gla = Blaga|Ne)p(Gro = Ne|Ne)p(Grip = Ne)$$

$$= \frac{1}{3}\frac{2}{3}\frac{1}{3}\frac{2}{3}\frac{3}{8} = \frac{1}{54}$$

Primena – razrešavanje višeznačnosti reči

- Neke reči se mogu sa jednog jezika prevesti na drugi na više načina (npr. *like*)
- Višeznačnost se često može razrešiti na osnovu konteksta
- Ako su u i v susedne reči, pravilo izbora prevoda je:

$$\arg\max_x p(x|u, v)$$

- Takođe, primenjivan u klasifikaciji teksta, detekciji neželjene pošte, prepoznavanju karaktera, prepoznavanju govora, itd.

Logistička regresija

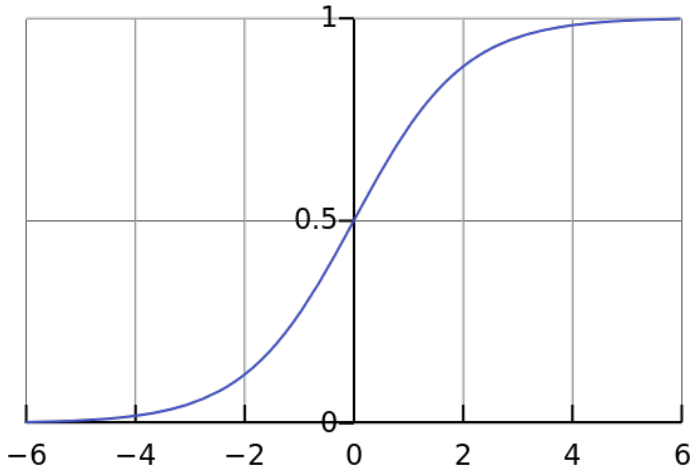
- Neka važi $y \in \{0, 1\}$
- Želimo da modelujemo verovatnoću $p(y|x)$
- Razmotrimo linearnu formu modela:

$$f_w(x) = w_0 + \sum_{i=1}^{n} w_i x_i$$

- Što je tačka dalje od razdvajajuće hiperravni, to je verovatnoća pripadanja odgovarajućoj klasi veća
- Međutim, $f_w(x)$ nije u intervalu $[0, 1]$
- Zbog toga se koristi sigmoidna funkcija:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Sigmoidna funkcija



Slika: https://en.wikipedia.org/wiki/Sigmoid_function

Logistička regresija

- Forma modela:
$$f_w(x) = \sigma(w \cdot x)$$
- Željenu verovatnoću modelujemo kao $p_w(y|x) = f_w(x)^y(1 - f_w(x))^{1-y}$
- Kako izabrati parametre w?

Logistička regresija

- Uslovna verovatnoća opažanja iz skupa podataka je

$$\prod_{i=1}^{N} p_w(y_i|x_i)$$

- Ova vrednost se naziva verodostojnošću parametara
- Potrebno je naći vrednosti parametara w čija je verodostojnost najveća
- Kako su sume numerički i analitički pogodnije od proizvoda, razmatra se logaritam ovog proizvoda
- Obično se umesto maksimizacije koristi minimizacija, pa se razmatra negativna vrednost logaritma
- Negativna vrednost logaritma verodostojnosti:

$$\mathcal{L}(w) = -\sum_{i=1}^{N} \log p_w(y_i|x_i) = -\sum_{i=1}^{N} [y_i \log f_w(x_i) + (1 - y_i) \log(1 - f_w(x_i))]$$

Logistička regresija

- Optimizacioni problem

$$\min_{w} -\sum_{i=1}^{N} [y_i \log f_w(x_i) + (1 - y_i) \log(1 - f_w(x_i))]$$

- Ova funkcija je konveksna, pa ima jedinstven globalni minimum
- Za optimizaciju se obično koristi Njutnova metoda

Primena – ...svuda

- Predviđanje 30-dnevnog mortaliteta od infarkta na osnovu zdravstvenih podataka (pol, visina, težina, pušenje, krvni pritisak, itd.)
- Posebna popularnost u medicniskim primenama
- Standardni model za poređenje prilikom klasifikacije

Metoda potpornih vektora (SVM)

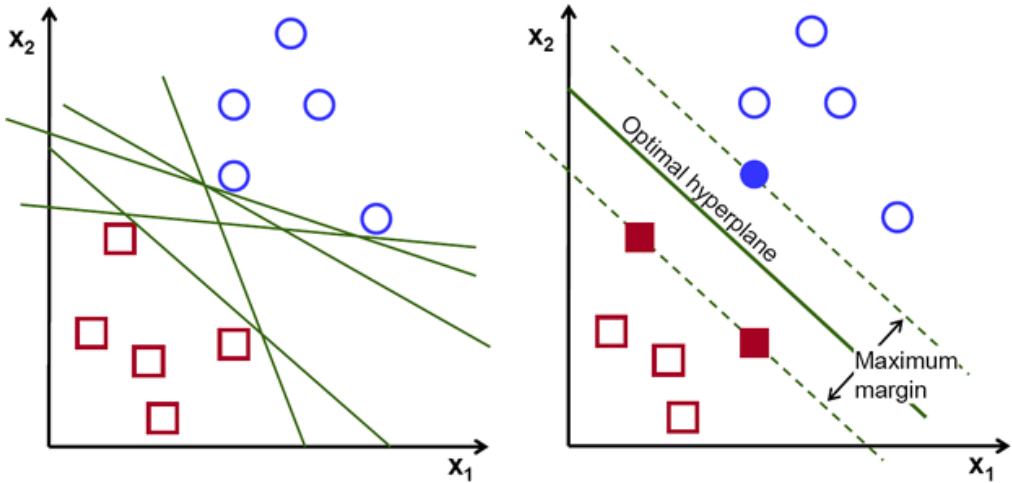
- Neka važi $y \in \{-1, 1\}$
- Model:

$$f_{w,w_0}(x) = w \cdot x + w_0$$

- *Margina razdvajajuće hiperravni* je minimum rastojanja od te hiperravni do neke od tačaka iz skupa podataka
- Među svim razdvajajućim hiperavnima, potrebno je naći optimalnu – onu sa najvećom marginom
- Za svaku od tačaka x, rastojanje do hiperravni je

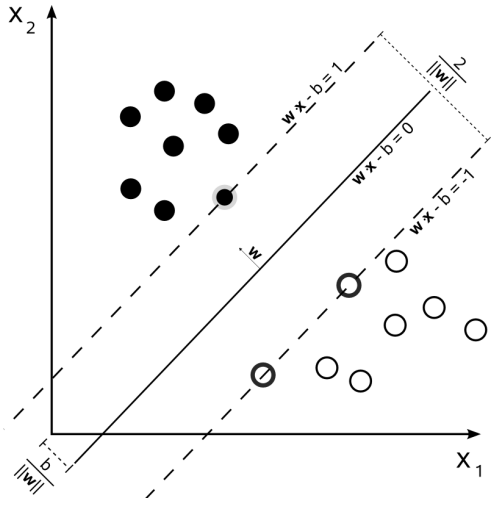
$$\frac{|w \cdot x + w_0|}{\|w\|}$$

Ilustracija razdvajajućih hiperravni



Slika: OpenCV, Introduction to Support Vector Machines.

Ilustracija optimalne razdvajajuće hiperravni



Slika: https://en.wikipedia.org/wiki/Support_vector_machine

Osnovna formulacija

- Optimizacioni problem:

$$\min_{w,w_0} \frac{\|w\|}{2}$$

$$y_i \left(w \cdot x_i + w_0\right) \geq 1 \qquad i = 1, \ldots, N$$

Linearno neseparabilan slučaj

- U slučaju da podaci nisu linearno separabilni, potrebno je dozvoliti greške, ali uz težnju da se njihov broj i intenzitet minimizuju:

$$\min_{w,w_0} \frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)$$

$$y_i\left(w \cdot x_i + w_0\right) \geq 1 - \xi_i, \qquad i = 1, \ldots, N$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, N$$

Rešenje optimizacionog problema

- Problem kvadratnog programrianja, a postoje specijalizovani metodi za njegovo rešavanje
- Rešenje je oblika:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

gde su α_i Lagranžovi množioci za koje važi $0 \leq \alpha_i \leq C$

- Podaci x_i za koje je $\alpha_i > 0$ su *potporni vektori*
- Model je onda oblika

$$f_{w,w_0}(x) = \sum_{i=1}^{N} \alpha_i y_i \; x_i \cdot x + w_0$$

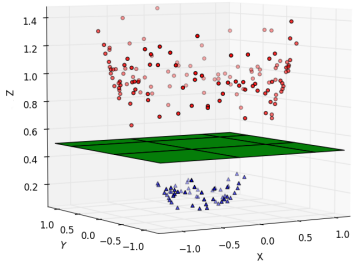
- Klasa se određuje funkcijom $\text{sgn}(f_{w,w_0}(x))$

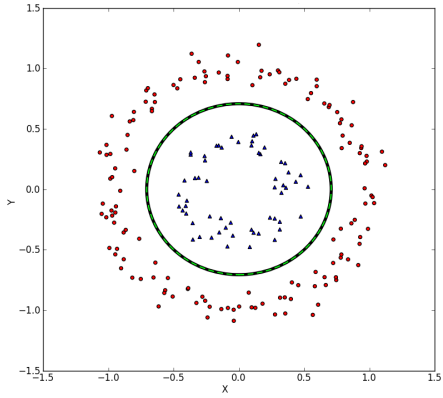
Šta ako granica između klasa nije hiperravan?

- Hiperravan nije uvek pogodna granica između klasa
- Rešenje je se podaci preslikaju u neki visokodimenzionalni prostor

Preslikavanje u visokodimenzionalni prostor

$$(x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$$





Slika: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Kernel

- Matrica K je pozitivno semidefinitna ukoliko za svako x važi $x^T Kx \geq 0$
- Neka je $\mathcal{X}$ neprazan skup
- Neka je k funkcija $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- Ako je za svako $n \in \mathbb{N}$ i svako $x_1, \ldots, x_n \in \mathcal{X}$ matrica dimenzija $n \times n$ sa elementima $k(x_i, x_j)$ pozitivno semidefinitna, funkcija k je *(pozitivno semidefinitan) kernel*
- Za svaki kernel k postoji preslikavanje Φ iz $\mathcal{X}$ u neki prostor $\mathcal{H}$ sa skalarnim proizvodom tako da važi $k(u, v) = \Phi(u) \cdot \Phi(v)$

Primeri kernela

- Polinomijalni kernel:

$$k(u, v) = (u \cdot v + 1)^d$$

- Gausov kernel:

$$k(u, v) = e^{-\gamma \|u - v\|^2}$$

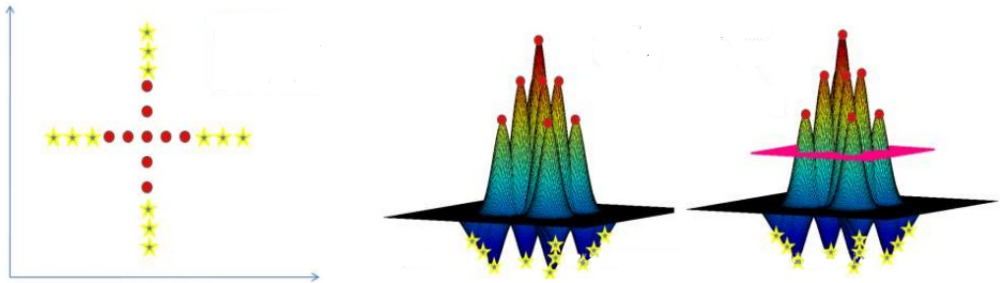
Kernel trik

- Ako se skalarni proizvod u $\mathbb{R}^n$ zameni kernelom k, model ima oblik:

$$f_{w,w_0}(x) = \sum_{i=1}^{N} \alpha_i y_i \; k(x_i, x) + w_0$$

- Ništa drugo se ne menja u metodu, pošto kernel jeste skalarni proizvod u nekom prostoru!
- Reprezentacije podataka u tom prostoru se ne konstruišu eksplicitno
- Postoje kerneli koji odgovaraju skalarnim proizvodima u beskonačnodimenzionalnim prostorima, a izračunavaju se efikasno

Dejstvo Gausovog kernela



Slika: A. Sharma, Support Vector Machine Without Tears. Modifikovano.

VC dimenzija

- Važi:
$$h \leq \min([R^2\|w\|^2], N) + 1$$
- Optimizacioni problem izražava minimizaciju gornje granice VC dimenzije!
- Gornja granica ne zavisi od dimenzionalnosti prostora!

VC dimenzija

- Kolika je VC dimenzija skupa svih modela SVM sa gausovim kernelom sa proizvoljnim vrednostima paramera γ?

VC dimenzija

- Kolika je VC dimenzija skupa svih modela SVM sa gausovim kernelom sa proizvoljnim vrednostima paramera γ?
- Beskonačna je!
- Kakav god da je raspored podataka i kakvo god da je obeležavanje, postoji dovoljno velika vrednost γ (što čini Gausovo zvono dovoljno uskim), tako da uz dovoljno potpornih vektora greška bude 0

Primene

- Kategorizacija teksta
- Prepoznavanje objekata na slikama
- Prepoznavanje rukom pisanih cifara
- Dijagnoza raka dojke
- Među algoritmima sa najboljim kvalitetom predviđanja u najrazličitijim domenima

Linearna regresija

- Forma modela:

$$f_w(x) = w_0 + \sum_{i=1}^{n} w_i x_i$$

- Minimizacioni problem

$$\min_w \sum_{i=1}^{N} (y_i - f_w(x_i))^2$$

- Matrična formulacija

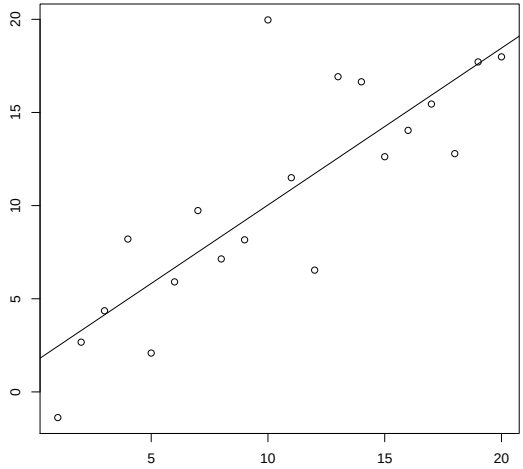
$$\min_w \|y - Xw\|$$

- Rešenje

$$w = (X^T X)^{-1} X^T Y$$

- Ako su matrice prevelike za čuvanje i inverziju, koriste se gradijentne metode optimizacije

Linearna regresija – primer



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

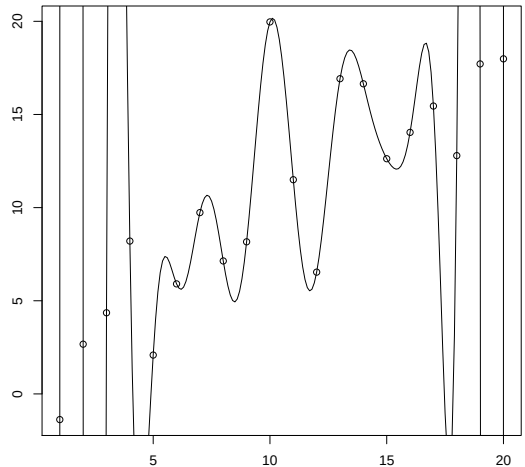
Linearna regresija

- Linearnost označava linearnost po parametrima
- Sledeći model je linearan model:

$$f_w(x) = w_0 + \sum_{i=1}^{n} w_i x^i$$

- Iako grafik ovakvog modela deluje nelinearno u koordinatnom sistemu sa jednom koordinatom x, on ipak predstavlja hiperravan u koordinatnom sistemu $(x, x^2, \ldots, x^n)$

Linearna regresija – primer



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Grebena (eng. ridge) linearna regresija

- Ako su kolone matrice X linearno zavisne, matrica $X^T X$ nije invertibilna
- Zato se razmatra regularizovani problem:

$$\min_w \sum_{i=1}^{N} (y_i - f_w(x_i))^2 + \lambda \|w\|_2^2$$

- Rešenje

$$w = (X^T X + \lambda I)^{-1} X^T Y$$

- Regularizacija čini da matrica koja se invertuje ima pun rang

Primene

- Ustanovljavanje veze između pušenja i rizika od bolesti
- Prvi algoritam koji vredi isprobati u svakom regresionom problemu

Neuronske mreže

- Trenutno najpopularnija familija modela mašinskog učenja
- Od početka inspirisan strukturom mozga
- Osnove postavljene pedesetih (perceptron)
- Osamdesetih je predložen algoritam propagacije unazad (eng. backpropagation)
- Početkom ovog veka su postali dostupni adekvatni računski resursi i otkriveni algoritmi za efikasan trening dubokih neuronskih mreža
- Vrlo širok spektar primena
- Univerzalni aproksimator neprekidnih funkcija
- Puno varijanti specijalizovanih za različite probleme (propagacija unapred, rekurentne, konvolutivne)

Neuronska mreža sa propagacijom unapred

- Neuronska mreža se sastoji od slojeva koji se sastoje od jedinica (neurona)
- Svi slojevi osim poslednjeg se nazivaju skriveni
- Svaka jedinica kao ulaze uzima izlaze jedinica iz prethodnog sloja
- Svaka jedinica računa linearnu kombinaciju svojih ulaza, a potom vrši nelinearnu transformaciju nad rezultatom
- Koriste se različite nelinearne transformacije, odnosno *aktivacione funkcije*

Matematička formulacija modela

- h_i je vektor izlaza i-tog sloja mreže
- a_i je vektor vrednosti linearnih kombinacija koje računaju jedinice i-tog sloja pre primene aktivacione funkcije
- Model je definisan sledećim jednakostima:

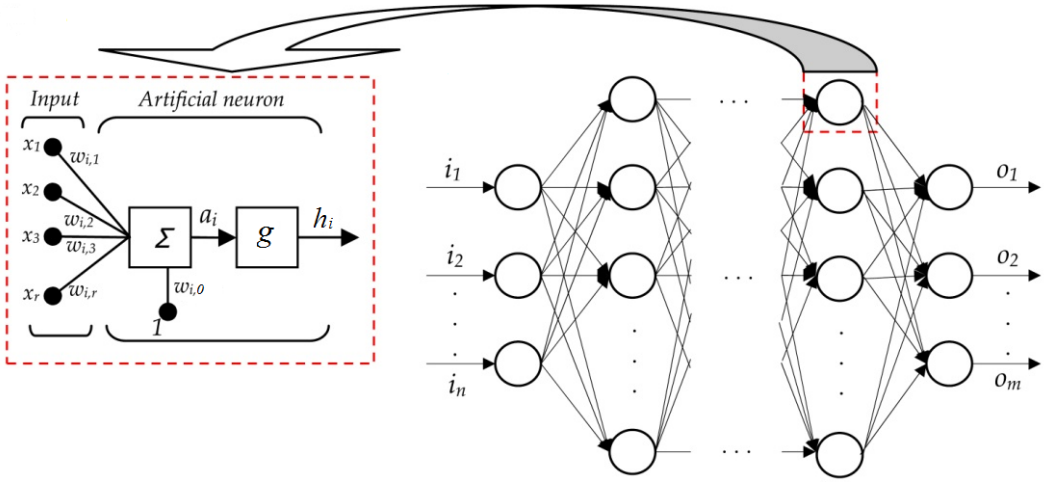
$$h_0 = x$$

$$a_i = W_i h_{i-1} + w_{i0}$$

$$h_i = g(a_i)$$

  gde su W_i matrice koeficijenata, w_{i0} vektori koeficijenata, a g aktivaciona funkcija
- Vektor svih koeficijenata modela biće obeležavan sa w
- Važi $f_w(x) = h_k$, gde je k indeks poslednjeg sloja

Ilustracija



Slika: D. Tanikić, V. Despotović, Artificial Intelligence Techniques for Modelling of Temperature in the Metal Cutting Process. Modifikovano.

Izbor aktivacione funkcije

- Sigmoidna funkcija
- Tangens hiperbolički
- ReLU: $\max(0, x)$

Regresija

- Prilikom primene neuronskih mreža na problem regresije, za poslednji nivo se obično ne koristi aktivaciona funkcija
- Greška se definiše kao srednje kvadratna greška

Klasifikacija

- U slučaju klasifikacije, obično se za poslednji nivo ne koristi aktivaciona funkcija g, već vektorska funkcija *softmax*

$$\text{softmax}(a_k) = \left(\frac{\exp(a_{k1})}{\sum_i \exp(a_{ki})}, \ldots, \frac{\exp(a_{kl})}{\sum_i \exp(a_{ki})} \right)$$

- Softmax funkcija transformiše izlaze tako da su nenegativni i sumiraju se na 1 i stoga se mogu interpretirati kao raspodela verovatnoće po mogućim klasama, pri čemu za svaku klasu postoji tačno jedan izlaz u poslednjim sloju
- Greška se definiše kao negativna vrednost logaritma verodostojnosti parametara

Kako se vrši trening?

- Parametri bi se mogli varirati na razne načine kako bi se videlo u kom pravcu se može izvršiti pomak kako bise dobila manja vrednost greške
- Takav postupak je računski skup
- Gradijent daje pravac pomeranja u kojem se greška *lokalno* najbrže povećava
- Kako izračunati gradijent za grešku neuronske mreže?

Propagacija unazad

- Algoritam koji je omogućio efikasan trening neuronskih mreža korišćenjem gradijenta
- Zasnovan na pravilu za izvod kompozicije funkcija
- Neka su funkcije $g : \mathbb{R}^m \to \mathbb{R}^n$ i $f : \mathbb{R}^n \to \mathbb{R}$

$$\partial_i(f \circ g) = \sum_{j=1}^{n} (\partial_j f \circ g)\partial_i g_j$$

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

Propagacija unazad

- Greška za mrežu sa k slojeva na jednoj instanci (h_k je funkcija parametara w):

$$E(w) = L(y, h_k) + \lambda\Omega(w)$$

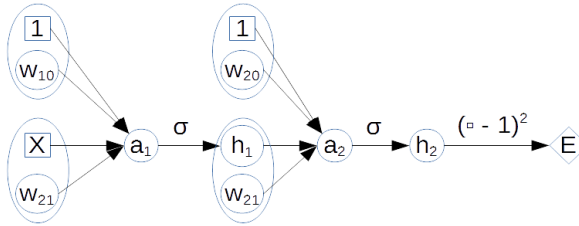
- Gradijent na celom skupu podataka se dobija sabiranjem gradijenata na pojedinačnim instancama
- Pretpostavlja se da je urađeno izračunavanje unapred, tako da su h_i poznati

$d = \nabla_{h_k} L(y, h_k)$
repeat
  $d = d \odot g'(a_k)$
  $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
  $\nabla_{W_k} E(w) = d{h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
  $d = {W_k}^T d$
  $k = k - 1$
until $k = 0$;

Propagacija unazad - primer

$E(w, \mathcal{D}) = (h_2 - 1)^2$

$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$



$d = \nabla_{h_k} L(y, h_k)$

repeat

$d = d \odot g'(a_k)$

$\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$

$\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$

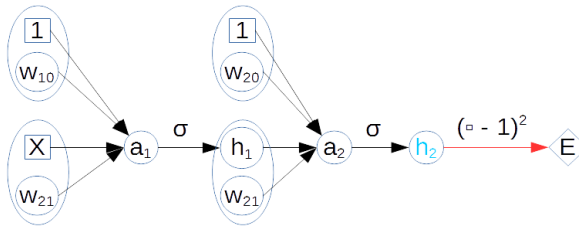
$d = {W_k}^T d$

$k = k - 1$

until $k = 0$;

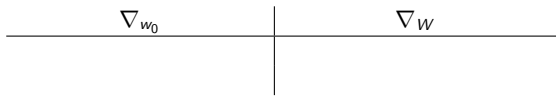
Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$
$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$



$d = \nabla_{h_k} L(y, h_k)$
repeat
  $d = d \odot g'(a_k)$
  $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
  $\nabla_{W_k} E(w) = d h_{k-1}{}^T + \lambda \nabla_{W_k} \Omega(w)$
  $d = W_k{}^T d$
  $k = k - 1$
until $k = 0$;

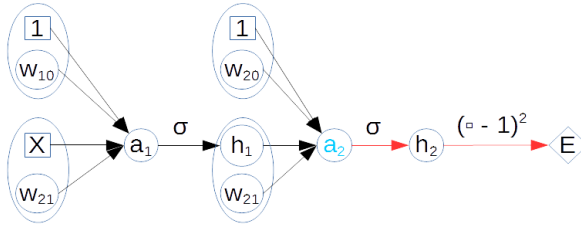
$$d = 2(h_2 - 1)$$



Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(\underbrace{w_{20} + w_{21}\sigma(w_{10} + w_{11}x)}_{a_2})$$



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

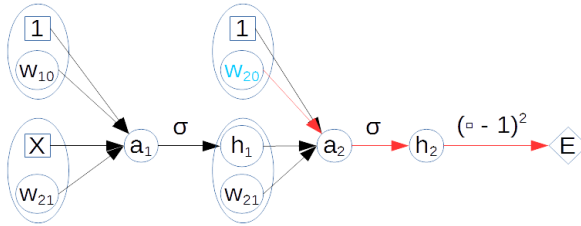
$$d = 2(h_2 - 1)\sigma'(a_2)$$

∇_{w_0}	∇_W

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(\underbrace{w_{20} + w_{21}\sigma(w_{10} + w_{11}x)}_{a_2})$$



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2(h_2 - 1)\sigma'(a_2)$$

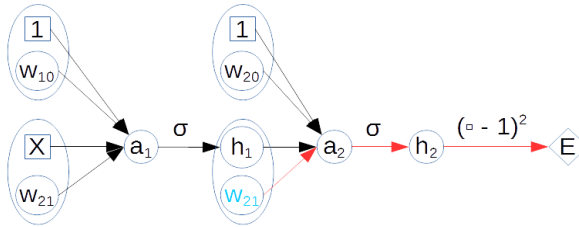
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2(h_2 - 1)\sigma'(a_2)$$

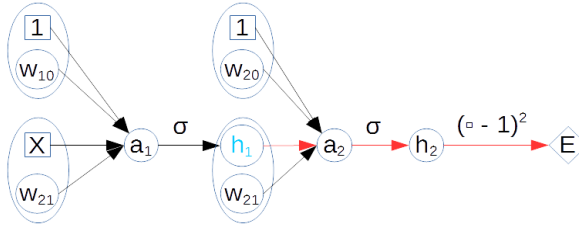
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)$$

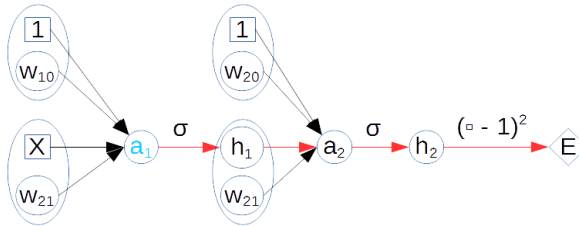
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

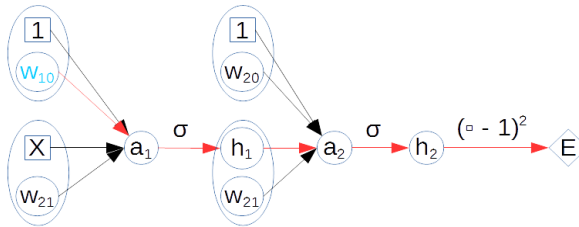
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

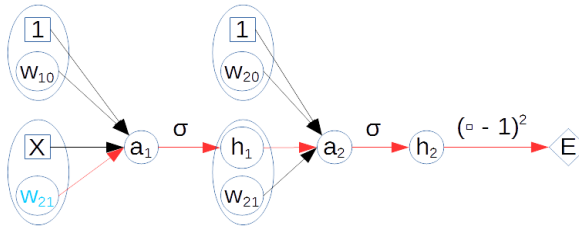
∇_{w_0}	∇_W
$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$	
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

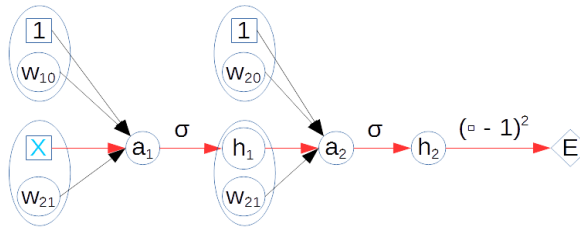
∇_{w_0}	∇_W
$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$	$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)x$
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

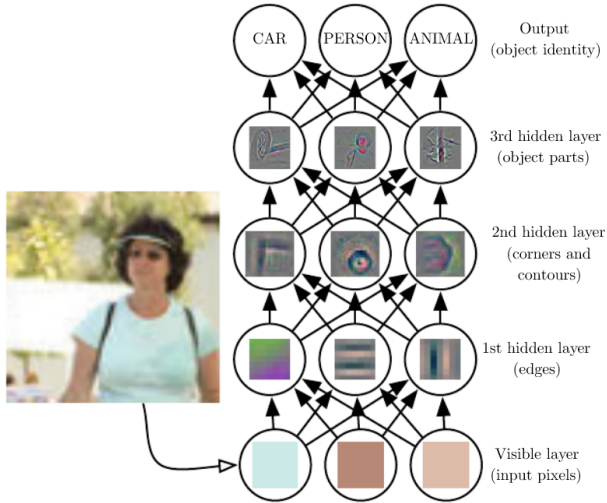
$$d = 2w_{11}w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

∇_{w_0}	∇_W
$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$	$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)x$
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Ekstrakcija atributa

- Izlazi jedinica na nižim slojevima se mogu smatrati atributima dobijenim na osnovu vrednosti prethodnih slojeva
- Svaki sloj vrši *ekstrakciju atributa*, pri čemu je smisao atributa na višim nivoima kompleksniji od smisla onih na nižim
- Omogućava primenu neuronskih mreža na sirove podatke, bez prethodnog definisanja atributa od strane eksperata
- Vrlo korisno za obradu slike, videa i zvuka

Ekstrakcija atributa



Slika: I. Goodfellow, Y. Bengio, A. Courville, Deep Learning

Konvolutivne neuronske mreže

- Posebno popularna klasa neuronskih mreža
- Koriste se za obradu signala (zvuk, slike...)
- Zasnivaju se upravo na prethodno pomenutoj ideji ekstrakcije atributa

Arhitektura konvolutivnih mreža

- Konvolutivna mreža se sastoji od konvolutivnih slojeva, slojeva agregacije (eng. pooling) i standardne neuronske mreže
- Konvolutivni slojevi i slojevi agregacije se smenjuju jedan za drugim, pri čemu su im dimenzije sve manje
- Standardna neuronska mreža je povezana na izlaze poslednjeg sloja agregacije

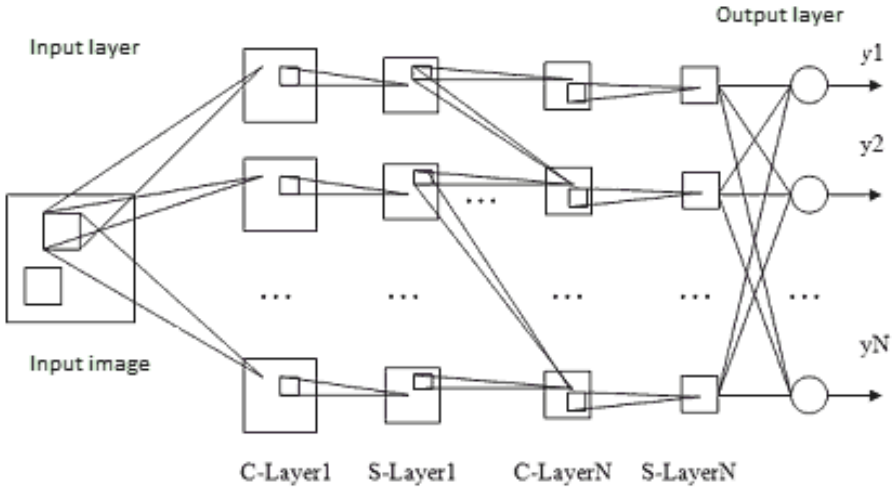
Konvolutivni sloj

- Jedan konvolutivni sloj se sastoji od neurona koji dele vrednosti parametara i stoga ih aktivira ista vrsta ulaza
- Svaka jedinica konvolutivnog sloja je dodeljena jednom delu ulaza (npr. slike)
- Tako konvolutivni sloj detektuje gde se u ulazu nalazi neka zakonitost
- Deljenje parametara omogućava treniranje mreža sa ogromnim brojem jedinica

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja
- Agregacija se najčešće vrši primenom maksimuma
- Time se smanjuje dimenzionalnost podataka, a čuva se informacija da li je negde u ulazu pronađena zakonitost koju konvolutivni sloj pronalazi

Shema konvolutivne mreže



Slika: http://masters.donntu.org/2012/fknt/umiarov/diss/indexe.htm

Mane neuronskih mreža

- Nije ih lako trenirati
- Visoka računska zahtevnost
- Visoka konfigurabilnost, ali bez jasnih smernica kako izabrati konfiguraciju
- Visok potencijal za preprilagođavanje
- Traže velike količine podataka

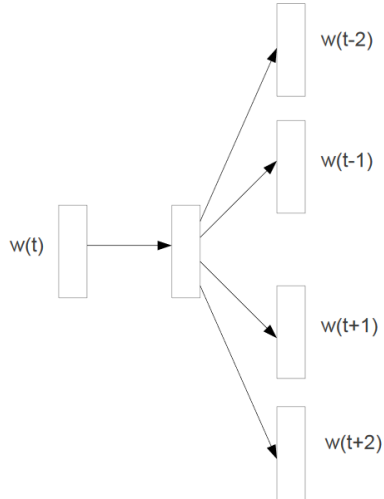
Primeri primena

- Prpoznavanje objekata na slikama
- Prepoznavanje lica
- Autonomna vožnja
- Igranje igara (TD-Gammon, Alfa Go, igre na Atariju, Doom)
- Obrada prirodnog jezika
- Sinteza algoritama iz primera (neuronske Tjuringove mašine)

Primer primene - modelovanje semantike reči

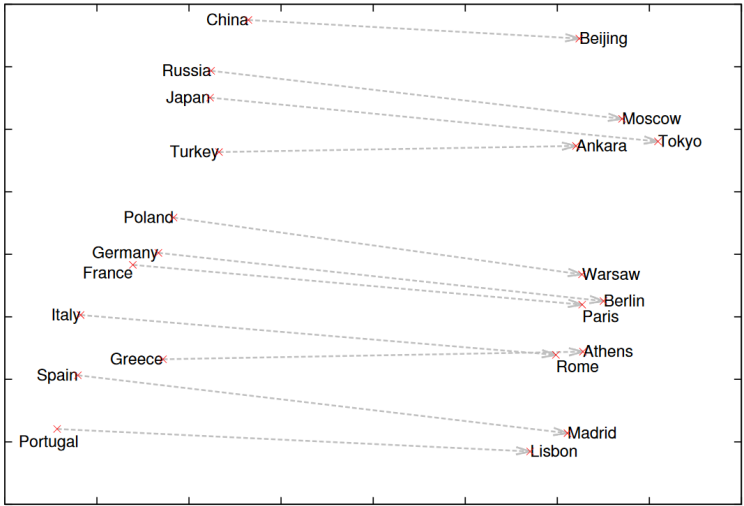
- Rečnik se sastoji od n reči
- i-ta reč se predstavlja vektorom koji ima jedinicu na i-tom mestu
- Mreža ima jedan skriveni sloj i jedan izlazni
- Ulaz neuronske mreže je vektor koji predstavlja reč
- Mreža predviđa koje su okolne reči za datu ulaznu reč
- Za reprezentacije i-te reči se uzimaju koeficijenti skrivenog sloja koji odgovaraju i-tom ulazu

Primer primene - modelovanje semantike reči



Slika: T. Mikolov, I. Sutskever, K Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality

Primer primene - modelovanje semantike reči



Slika: T. Mikolov, I. Sutskever, K Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality

Primer primene - mašinsko prevođenje

- Pristup prevođenju od kojeg se polazi je razumeti misao koju rečenica izražava, a potom tu misao izgovoriti na drugom jeziku
- Rekurentna neuronska mreža se trenira tako da uzima na ulazu rečenice jednog jezika, kroz nekoliko skrivenih slojeva (koji čine enkoder) ih kodira u novu reprezentaciju iz koje se kroz još nekoliko skrivenih slojeva (koji čine dekoder) ponovo rekonstruiše ista rečenica, ali na drugom jeziku
- Vektor koji daje dekoder predstavlja ,,misao“ koju rečenica izražava
- Po sličnom principu moguće je ,,prevoditi“ iz slika na neki prirodni jezik

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Opšta shema dizajna

- Mnogi algoritmi nadlgedanog učenja predstavljaju instance jedne opšte sheme dizajna
- Omogućava povezivanje svojstava algoritama, sa specifičnim aspektima njihovog dizajna
- Olakšava dizajn novih algoritama
- Olakšava razumevanje postojećih algoritama

Dimenzije algoritama nadgledanog učenja

- Sveobuhvatnost modela (generativni, diskriminativni, neprobabilistički diskriminativni)
- Reprezentacija modela (linearna, nelinearna, zasnovana na instancama, ...)
- Fukcija greške (kvadrat razlike, apsolutna vrednost razlike, ...)
- Regularizacija (grebena, laso, elastična mreža, grupni laso, hijerarhijski laso, ...)
- Optimizacioni metod (prvog ili drugog reda, sa ili bez ograničenja, konveksan ili nekonveksan, ...)

Sveobuhvatnost modela

- Koliko informacija o podacima model treba da obuhvati?
- Puna informacija o podacima je sadržana u raspodeli podataka $p(x, y)$
- Ne mora nam biti neophodna
- U poretku opadajuće količine informacija i zahteva, razlikujemo:
  - generativne modele (probabilističke)
  - diskriminativne modele (probabilističke) i
  - neprobabilističke diskriminativne modele

Generativni modeli

- Generativni modeli modeluju zajedničku raspodelu $p(x, y)$
- Ili, alternativno, raspodelu $p(x|y)$, pošto važi $p(x, y) = p(x|y)p(y)$
- U potpunosti opisuju podatke
- Stoga su u stanju da generišu podatke (sa istim svojstvima poput podataka na kojima je trenirano)
- Potrebno je napraviti pretposavke o raspodeli verovatnoće (a one mogu biti pogrešne)
- Podložne su prokletstvu dimenzionalnosti, tako da zahtevaju puno podataka

Generativni modeli

- Generativni modeli modeluju zajedničku raspodelu $p(x, y)$
- Ili, alternativno, raspodelu $p(x|y)$, pošto važi $p(x, y) = p(x|y)p(y)$
- U potpunosti opisuju podatke
- Stoga su u stanju da generišu podatke (sa istim svojstvima poput podataka na kojima je trenirano)
- Potrebno je napraviti pretposavke o raspodeli verovatnoće (a one mogu biti pogrešne)
- Podložne su prokletstvu dimenzionalnosti, tako da zahtevaju puno podataka
- Da li je potrebno poznavati zavisnosti između atributa kako bi se predviđala ciljna promenljiva?

Primer – naivni Bajesov klasifikator

- Naivni Bajesov klasifikator se zasniva na Bajesovoj formuli:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

- ,,Naivna“ pretpostavka:

$$p(x|y) \approx \prod_{i=1}^{n} p(x_i|y)$$

- $p(x_i|y)$ i $p(y)$ se ocenjuju iz podataka
- Činjenica da ocenjuje $p(x|y)$ ga čini generativnim modelom

Diskriminativni modeli

- Diskriminativni modeli modeluju uslovnu raspodelu $p(y|x)$
- Dovoljno da omoguće predviđanje i pruže informaciju o pouzdanosti
- Nisu podložni prokletstvu dimenzionalnosti
- Potrebno je napraviti pretposavke o raspodeli verovatnoće
- Ne mogu generisati podatke

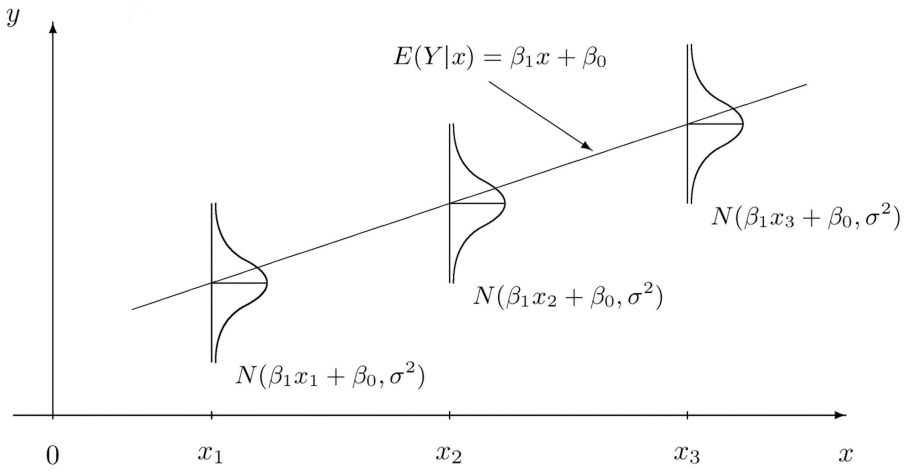
Primer – linearna regresija

- Model:

$$y|x \sim \mathcal{N}\left(w \cdot x, \sigma^2\right)$$

- σ^2 se može oceniti iz podataka

Primer – linearna regresija



Slika: D. Shafer, Z. Zhang, Introductory Statistics, 2012.

Primer – logistička regresija

- Bernulijeva raspodela: $y \sim Ber(\mu)$ means

$$p(y) = \begin{cases} \mu & \text{ako je } y = 1 \\ 1 - \mu & \text{ako je } y = 0 \end{cases}$$

- Model:

$$y|x \sim Ber(\sigma(w \cdot x))$$

Neprobabilistički diskriminativni modeli

- Neprobabilistički diskriminativni modeli modeluju funkciju $f : \mathbb{R}^n \to \mathbb{R}$ koja preslikava ulaze u vrednosti ciljne promenljive
- Omogućavaju predviđanje
- Ne prave pretpostavke o raspodeli podataka
- Ne podležu prokletstvu dimenzionalnosti
- Ne pružaju procenu pouzdanosti predviđanja
- Ne mogu generisati podatke

Primer – SVM

- Model:

$$w \cdot x + w_0$$

- Optimizacioni problem:

$$\min_w \frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)$$

$$y_i\left(w \cdot x_i + w_0\right) \geq 1 - \xi_i, \qquad i = 1, \ldots, N$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, N$$

Reprezentacija modela

- Opisuje odnose između atributa i ciljne promenljive
- Može biti izabrana ili dizajnirana u skladu sa specifičnostima domena
- Primeri:
  - Linearni i njima srodni modeli (linearna i logistička regresija, uopšteni linearni modeli, SVM, ...)
  - Nelinearni (neuronske mreže, ...)
  - Zasnovani na pravilima (stabla odlučivanja, ...)
  - Zasnovani na instancama (k najbližih suseda, SVM, funkcije radijalne baze (RBF), kriging)

Linearni i njima srodni modeli

- Model:

$$f_w(x) = g(w_0 + \sum_{i=1}^{n} w_i x_i)$$

  za neku funkciju g
- Ako je g identitet, ovakav model izražava:
  - Nezavisne uticaje atributa na cilju promenljivu
  - Uticaj na ciljnu promenljivu je proporcionalan promeni vrednosti atributa

Linearni modeli – prednosti i mane

- Prednosti
  - Manje podložni preprilagođavanju
  - Računski često manje zahtevni
  - Interpretabilni
- Mane:
  - Forma moze biti neadekvadna
  - Mogu biti nedovoljno prilagodljivi

Interakcije atributa

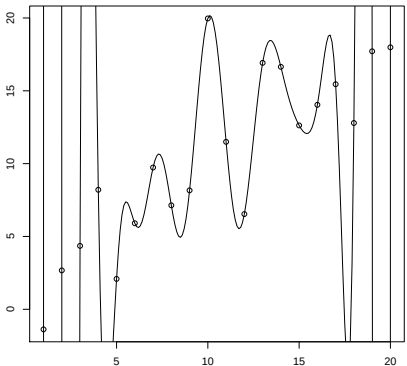
- Moguće je linearnim modelom izraziti i interakcije atributa:

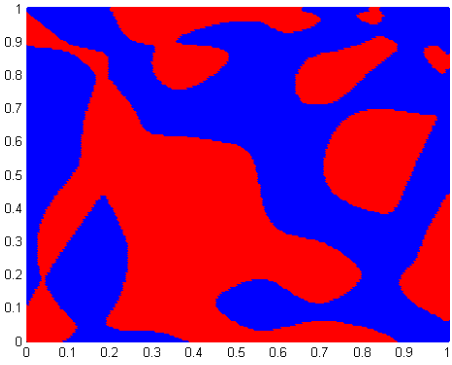
$$f(x, w, \beta) = g(w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i \leq j} \beta_{ij} x_i x_j)$$

- Uticaj promene vrednosti atributa x_i na ciljnu promenljivu zavisi od vrednosti atributa x_j

Interakcije atributa

- Interakcije mogu poslužiti za definisanje modela koji su nelinearni u odnosu na atribute (ali i dalje linearni u odnosu na parametre)





Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Funkcija greške

- Meri odstupanje predviđanja od prave vrednosti ciljne promenljive
- Zavisi od problema
- Može se definisati za specifičan problem sa konkretnim svojstvima na umu

Negativna vrednost logaritma verodostojnosti (1)

- Modeluje se verovatnoća $p_w(y|x)$ (ili $p_w(x, y)$ u generativnom slučaju)
- Negativna vrednost logaritma verodostojnosti (NLL)

$$\min_w \sum_{i=1}^{N} -\log p_w(y_i|x_i)$$

- Stoga, $-\log p_w(y|x)$ predstavlja funkciju greške
- Što je verovatnoća vrendosti y za dato x manja, greška je veća, što je prirodno

NLL za logističku regresiju

- NLL:

$$\min_w -\sum_{i=1}^{N} [y_i \log f_w(x_i) + (1 - y_i) \log(1 - f_w(x_i))]$$

- Stoga $-u \log v - (1 - u) \log(1 - v)$ predstavlja funkciju greške
- Ako u i v saglasni, greška je mala, a u suprotnom je velika

NLL za linearnu regresiju

- Model:

$$y|x \sim \mathcal{N}\left(w \cdot x, \sigma^2\right)$$

ili ekvivalentno:

$$p_w(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - w \cdot x)^2}{2\sigma^2}\right)$$

- NLL:

$$\mathcal{L}(w) = \frac{N}{2} \log 2\pi + N \log \sigma + \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - w \cdot x_i)^2$$

Kvadratna greška

- Funkcija greške:
$$L(u, v) = (u - v)^2$$
- Pruža regresionu funkciju kao rešenje
- U slučaju linearnog regresionog modela, ima isti efekat kao ocena maksimalne verodostojnosti normalne raspodele
- Diferencijabilna
- Osetljiva na izuzetke u podacima (eng. outliers)

Apsolutna greška

- Funkcija greške:
$$L(u, v) = |u - v|$$
- U slučaju linearnog regresionog modela, ima isti efekat kao ocena maksimalne verodostojnosti Laplasove raspodele
- Nije osetljiva na izizetke u podacima
- Nije diferencijabilna

SVM i greška u obliku šarke

- Minimizacioni problem:

$$\min_{w} \frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)$$

$$y_i\left(w \cdot x_i + w_0\right) \geq 1 - \xi_i, \qquad i = 1, \ldots, N$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, N$$

SVM i greška u obliku šarke

- Minimizacioni problem:

$$\min_{w} \frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)$$

$$y_i\left(w \cdot x_i + w_0\right) \geq 1 - \xi_i, \qquad i = 1, \ldots, N$$

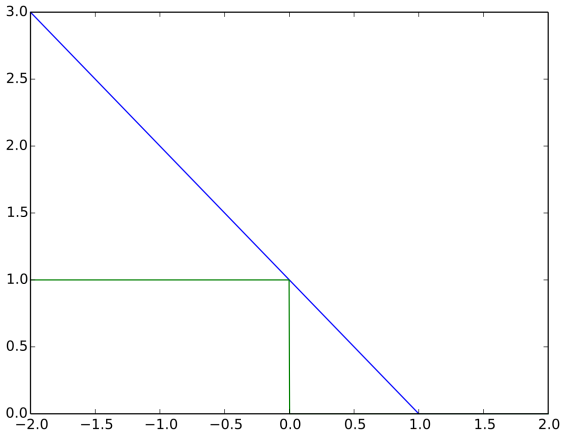
$$\xi_i \geq 0, \qquad i = 1, \ldots, N$$

- Reformulisani minimizacioni problem:

$$\left[\sum_{i=1}^{N} \max(0, 1 - y_i(w \cdot x + w_0))\right] + \lambda \frac{\|w\|^2}{2}$$

- Greška u obliku šarke (eng. hinge loss) $L(u, v) = \max(0, 1 - uv)$
- Konveksna aproksimacija funkcije $I(uv < 0)$

Greška u obliku šarke



Slika: Vikipedijin članak o grešci u obliku šarke

Regularizacija

- Podešavanje prilagodljivosti modela
- Nametanje specifične strukture modela
- Uključivanje domenskog znanja u model

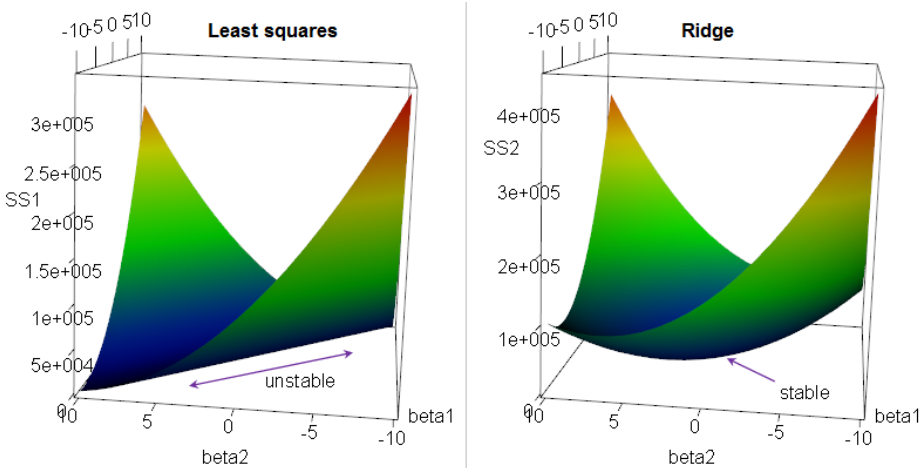
ℓ_2 regularizacija (grebena)

- Koristi regularizacioni izraz

$$\Omega(w) = \|w\|_2^2 = \sum_{i=1}^{n} w_i^2$$

- U kontekstu linearne regresije, u slučaju linearne zavisnosti kolona, funkcija cilja nema jedinstven minimum, već greben
- Stoga je rešenje nestabilno
- Zove se grebena regresija zbog toga što greben zamenjuje minimumom

ℓ_2 regularizacija (grebena)



Slika: Korisnik Glen_b na stats.stackexchange.com

ℓ_1 regularizacija (laso)

- Koristi regularizacioni izraz

$$\Omega(w) = \|w\|_1 = \sum_{i=1}^{n} |w_i|$$

- Pruža retke (eng. sparse) modele i na taj način vrši izbor atributa!
- Ovo je vrlo važno za interpretabilnost modela

ℓ_1 regularizacija (laso)

- Koristi regularizacioni izraz

$$\Omega(w) = \|w\|_1 = \sum_{i=1}^{n} |w_i|$$

- Pruža retke (eng. sparse) modele i na taj način vrši izbor atributa!
- Ovo je vrlo važno za interpretabilnost modela
- Problemi?

ℓ_1 regularizacija (laso)

- Koristi regularizacioni izraz

$$\Omega(w) = \|w\|_1 = \sum_{i=1}^{n} |w_i|$$

- Pruža retke (eng. sparse) modele i na taj način vrši izbor atributa!
- Ovo je vrlo važno za interpretabilnost modela
- Problemi?
- Nije diferencijabilna!
- Preciznost predviđanja je nešto manja nego kod grebene regresije
- Nestabilna u pogledu skupa izabranih atributa
- Elastična mreža (eng. elastic net) kao rešenje

ℓ_1 regularizacija (laso) – dve formulacije

- Formulacija bez ograničenja:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_w(x_i)) + \lambda \|w\|_1$$

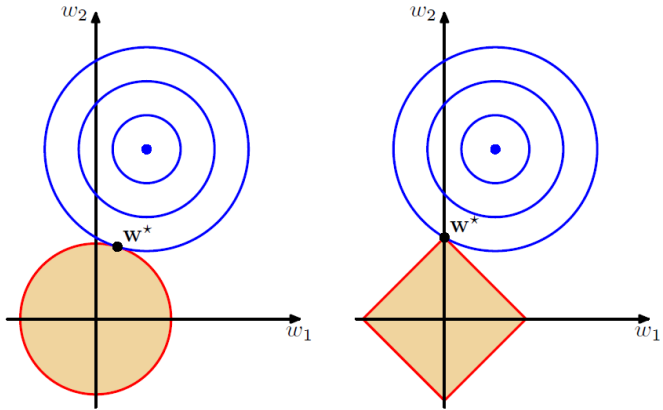
- Formulacija zasnovana na ograničenjima:

$$\min_w \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_w(x_i))$$

$$s.t. \quad \|w\|_1 \leq t$$

Laso regularizacija i retki modeli

- Ključna je razlika u oblicima ℓ_1 i ℓ_2 lopti



Slika: C. Bishop, Pattern Recognition and Machine Learning, 2006.

Grupna laso i retka grupna laso regularizacija

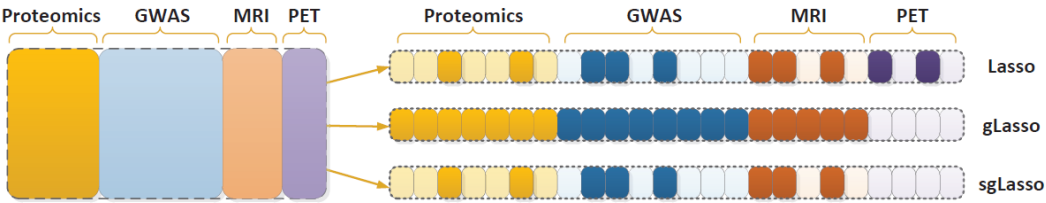
- Atributi su particionisani u disjunktne grupe $\{G_1, \ldots, G_k\}$ (npr. u biomedicinskim podacima)
- w_{G_i} su koeficijenti koji odgovaraju atributima iz grupe G_i
- Groupna laso regularizacija:

$$\Omega(w) = \sum_{i=1}^{k} \|w_{G_i}\|_2$$

- Retka grupna laso regularizacija:

$$\Omega(w) = \mu\|w\|_1 + (1-\mu)\sum_{i=1}^{k} \|w_{G_i}\|_2$$

Grupna laso i retka grupna laso regularizacija

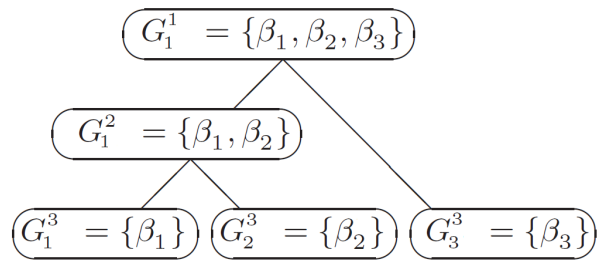


Slika: J. Ye, J. Liu, Sparse Methods for Biomedical Data, 2012.

Hijerarhijska laso regularizacija

- Atributi su organizovani u hijerarhiju u vidu stabla (npr. bolesti ili genetski podaci)
- G_j^i je skup atributa u podstablu čiji je koren čvor j na nivou i
- Hijerarhijska laso regularizacija:

$$\Omega(w) = \sum_{i,j} \|w_{G_j^i}\|_2$$



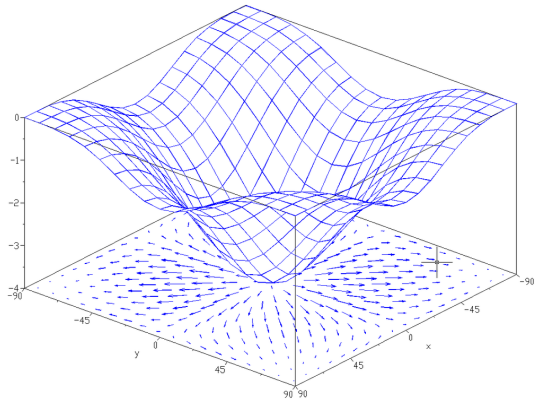
Slika: S. Kim, E. Xing, Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity, 2010. Modifikovano.

Optimizacioni metod

- Mogući izbori zavise od svojstava optimizacionog problema:
  - Diferencijabilnost
  - Dvostruka diferencijabilnost
  - Konveksnost
  - Prisustvo ograničenja
- Često se vrše aproksimacije problema nekim drugim problemom sa poželjnijim svojstvima

Diferencijabilan slučaj

- Diferencijabilna ciljna funkcija dopušta upotrebu gradijenata (pravac najstrmijeg uspona)



Slika: math.wikia.com/wiki/Gradient

Gradijentni spust

- Ponavljati dok postupak ne iskonvergira:

$$w_{k+1} = w_k - \mu_k \nabla E(w_k)$$

- Kako izabrati veličinu koraka μ_k?
- Fiksirana veličina koraka
- Robins-Monroovi uslovi za veličinu koraka dovoljni za konvergenciju

$$\sum_{k=1}^{\infty} \mu_k = \infty \quad \sum_{k=1}^{\infty} \mu_k^2 < \infty$$

- Armiho-Goldštajnova linijska pretraga

Konvergencija gradijentnog spusta (1)

- Funkcija koja slika $X \subseteq \mathbb{R}^n$ u $\mathbb{R}^m$ je Lipšic neprekidna ukoliko postoji konstanta C, takva da za sve $u, v \in X$ važi:

$$\|f(u) - f(v)\| \leq C\|u - v\|$$

- Diferencijabilna funkcija je jako konveksna ukoliko postoji konstanta $m > 0$, takva da u svakoj tački važi:

$$f(u) \geq f(v) + \nabla f(v)^T(u - v) + \frac{m}{2}\|u - v\|^2$$

- Dva puta diferencijabilna funkcija je jako konveksna ako je

$$\nabla^2 f(v) - mI$$

pozitivno semidefinitna matrica za svako v

Konvergencija gradijentnog spusta (2)

- Za konveksne funkcije sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{k}\right)$
- Za jako konveksne funkcije sa Lipšic neprekidnim gradijentom, greška je reda $O\left(c^k\right)$ za neko $0 < c < 1$
- Da li je jaka konveksnost realističan uslov u praksi?

Primer jako konveksne funkcije

- Greška $E(w)$ grebene regresije je:

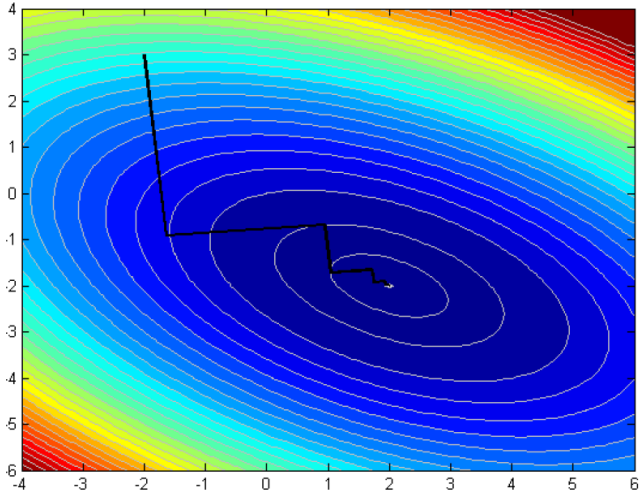
$$\begin{aligned}\|Xw - y\|_2^2 + \lambda\|w\|_2^2 &= (Xw - y)^T(Xw - y) + \lambda w^T w \\ &= w^T X^T Xw - w^T X^T y - y^T Xw + y^T y + w^T w\end{aligned}$$

- Hesijan ove funkcije je:

$$H = X^T X + \lambda I$$

- Pošto je $X^T X$ pozitivno semidefinitna matrica, i $H - \lambda I$ je pozitivno semidefinitna matrica
- Stoga je $E(w)$ jako konveksna funkcija po w
- Ne samo što regularizacija može da ubalaži problem preprilagođavanja, već može i da ubrza konvergenciju optimizacionih metoda!

Ilustracija gradijentnog spusta



Slika: Y. Li, Course materials.

Primer – kvadratna greška za logističku regresiju

- Pretpostavimo $y \in \{0, 1\}$
- Greška:

$$E(w) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \sigma(w \cdot x_i))^2$$

- Gradijent:

$$\frac{\partial}{\partial w_j} E(w) = -\frac{2}{N} \sum_{i=1}^{N} (y_i - \sigma(w \cdot x_i)) \sigma(w \cdot x_i)(1 - \sigma(w \cdot x_i)) x_{ij}$$

- Kolike su vrednosti parcijalnih izvoda ukoliko važi $|y_i - \sigma(w \cdot x_i)| \approx 1$?

Primer – kvadratna greška za logističku regresiju

- Pretpostavimo $y \in \{0, 1\}$
- Greška:

$$E(w) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \sigma(w \cdot x_i))^2$$

- Gradijent:

$$\frac{\partial}{\partial w_j} E(w) = -\frac{2}{N} \sum_{i=1}^{N} (y_i - \sigma(w \cdot x_i)) \sigma(w \cdot x_i)(1 - \sigma(w \cdot x_i)) x_{ij}$$

- Kolike su vrednosti parcijalnih izvoda ukoliko važi $|y_i - \sigma(w \cdot x_i)| \approx 1$?
- Približne nuli, iako je greška velika!

Primer – kvadratna greška za logističku regresiju

- Pretpostavimo $y \in \{0, 1\}$
- Greška:

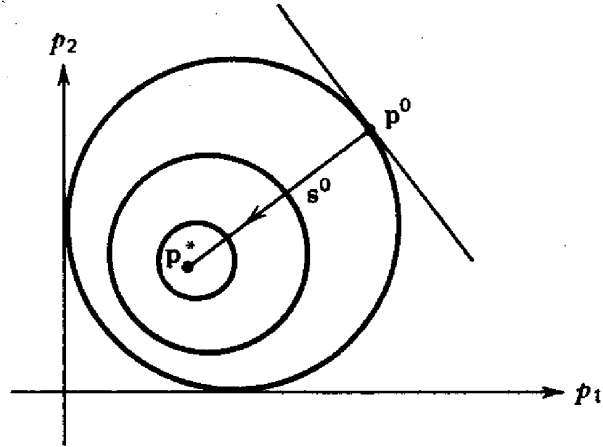
$$E(w) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \sigma(w \cdot x_i))^2$$

- Gradijent:

$$\frac{\partial}{\partial w_j} E(w) = -\frac{2}{N} \sum_{i=1}^{N} (y_i - \sigma(w \cdot x_i)) \sigma(w \cdot x_i)(1 - \sigma(w \cdot x_i)) x_{ij}$$

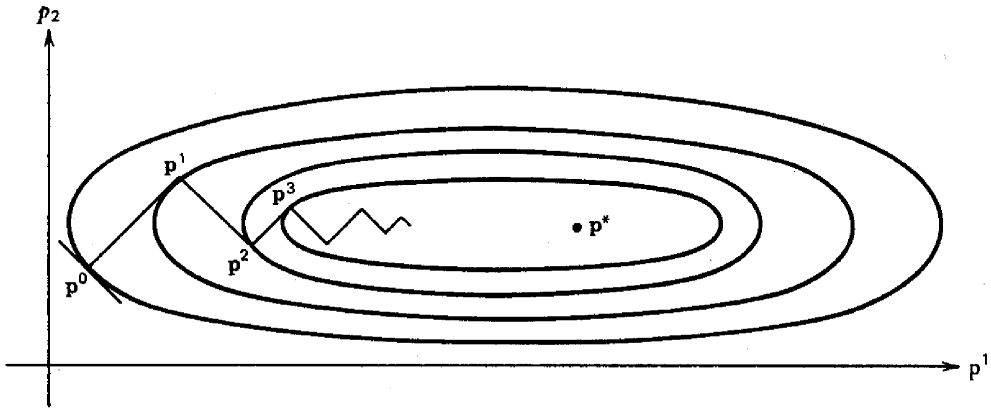
- Kolike su vrednosti parcijalnih izvoda ukoliko važi $|y_i - \sigma(w \cdot x_i)| \approx 1$?
- Približne nuli, iako je greška velika!
- Zato se koristi NLL formulacija

Gradijentni spust pri kružnim konturama funkcije cilja



Slika: K. Yuk, J. Xue, Optimization Techniques for Circuit Design, 2003.

Gradijentni spust pri izduženim konturama funkcije cilja



Slika: K. Yuk, J. Xue, Optimization Techniques for Circuit Design, 2003.

Nedostaci gradijentnog spusta

- Spora konvergencija
- Pravac definisan gradijentom je samo lokalno optimalan
- Cik-cak putanje spusta (problem ublažen konjugovanim gradijentima)
- Za velike količine podatka, puno vremena se troši da bi se izračunao pravac koji nije optimalan

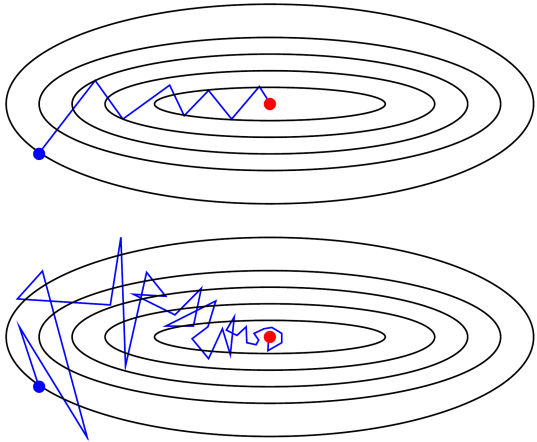
Stohastički gradijentni spust

- Neka korak bude slučajni vektor takav da je njegovo očekivanje kolinearno sa gradijentom funkcije $E(w)$
- Jedna varijanta je ponavljati za sve $(x_i, y_i) \in \mathcal{D}$, dok postupak ne iskonvergira:

$$w_{k+1} = w_k - \mu_k \nabla E(w_k, \{(x_i, y_i)\})$$

- Jeftino aproksimira gradijent
- Greška aproksimacije čak može služiti kao vid regularizacije
- Potencijal za bekstvo iz lokalnih minimuma
- Manje podložan problemima redundantnosti podataka
- Aproksimacija može biti neprecizna
- Trening pomoću podskupova (eng. minibatch) je čest kompromis
- Najpopularniji algoritam za treniranje neuronskih mreža

Gradijentni spust naspram stohastičkog gradijentnog spusta



Slika: F. Bach, Stochastic optimization: Beyond stochastic gradients and convexity.

Brzina konvergencije stohastičkog gradijentnog spusta

- Za konveksne funkcije sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{\sqrt{k}}\right)$
- Za jako konveksne funkcije sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{k}\right)$
- Asimptotski je sporiji od gradijentnog spusta, ali zbog mnogo manje cene jedne iteracije, stohastički gradijentni spust se često koristi u mnogim praktičnim kontekstima, poput treninga neuronskih mreža

Metod inercije (eng. momentum)

- Ideja je zadržati uticaj prethodnih gradijenata, kako bi promena pravca bila teža:

$$d_{k+1} = \beta_k d_k - \mu_k \nabla E(w_k)$$

$$w_{k+1} = w_k + d_{k+1}$$

- Njihov uticaj ipak eksponencijalno opada
- Ublažava problem cik-cak kretanja jer kretanje ne sledi oštre promene pravca gradijenta
- Vrlo popularan za treniranje neuronskih mreža, posebno u kombinaciji sa stohastičkim gradijentnim spustom

Nesterovljev ubrzani gradijentni spust

- Modifikacija metoda inercije:

$$d_{k+1} = \beta_k d_k - \mu_k \nabla E(w_k + \beta_k d_k)$$

$$w_{k+1} = w_k + d_{k+1}$$

- Postoji specifičan izbor vrednosti β_k i μ_k
- Geometrijska interpretacija nije sasvim jasna
- Za konveksne funkcije sa Lipšic neprekidnim gradijentom greška je reda $O\left(\frac{1}{k^2}\right)$
- Asimptotski optimalan optimizacioni metod prvog reda za ovakve funkcije
- Pogodan za probleme visoke dimenzionalnosti

ADAM

- Pristrasne ocene prvog i drugog momenta gradijenta:

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla E(w_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla E(w_k) \odot \nabla E(w_k)$$

- Korekcija pristrasnosti:

$$\widehat{m}_{k+1} = m_{k+1}/(1 - \beta_1^{k+1})$$

$$\widehat{v}_{k+1} = v_{k+1}/(1 - \beta_2^{k+1})$$

- Ažuriranje parametara:

$$w_{k+1} = w_k - \mu_{k+1}\frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Stohastičko uprosečavanje

- Da li je moguće imati prednosti stohastičkog gradijentnog spusta, ali sa brzinom konvergencije standardnog gradijentnog spusta?
- Da, u slučaju funkcija koje su konačne sume/proseci drugih funkcaja iste forme, što je najčešće slučaj u mašinskom učenju
- U slučaju gradijentnog spusta važi:

$$\nabla E(w, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \nabla E(w, \{(x_i, y_i)\})$$

- Osnovna ideja je upotrebiti proseke gradijenata na pojedinačnim instancama, ali u svakom koraku ažurirati samo jedan pojedinačni gradijent

Stohastički gradijentni spust zasnovan na uprosečavanju (SAG)

- Neka je i_k niz slučajnih promenljivih sa uniformnom raspodelom nad skupom vrednosti $\{1, \ldots, N\}$
- Korak SAG algoritma:

$$g_i^{k+1} = \begin{cases} \nabla E(w_k, \{(x_i, y_i)\}) & i = i_{k+1} \\ g_i^{k-1} & \text{inače} \end{cases}$$

$$w_{k+1} = w_k - \frac{\mu_k}{N} \sum_{i=1}^{N} g_i^{k+1}$$

- Zahteva čuvanje N gradijenata

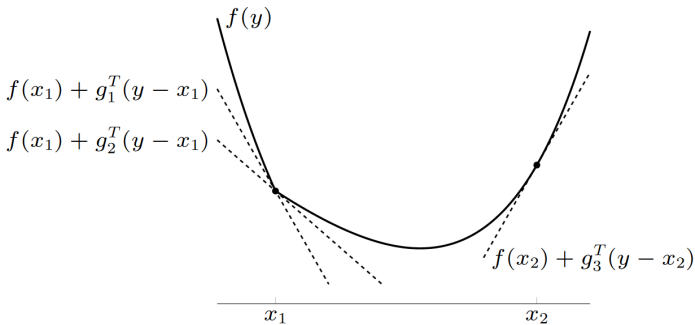
Alternative

- SAGA, SVRG, SDCA,...
- SVRG ne zahteva dodatni prostor, ali zahteva dva izračunavanja gradijenta po koraku
- Razlikuju se u odnsu na robusnost u odnosu na uslovljenost, korišćenje epoha, primenljivost na nediferencijabilne probleme, itd.

Nediferencijabilan slučaj (1)

- Vektor g je podgradijent (eng. subgradient) funkcije f u tački x, ako važi

$$f(x) \geq f(x_0) + g^T(x - x_0)$$



Slika: Lieven Vandenberghe, materijali za kurs

Nediferencijabilan slučaj (2)

- Skup svih podgradijenata funkcije f u tački x se naziva poddiferencijal i označava $\partial f(x)$
- Ako funkcija ima lokalni optimum u tački x, važi $0 \in \partial f(x)$

Primer – podgradijentni spust za ℓ_1 regularizaciju

- Regularizovana greška:
$$\mathcal{L}(w) = E(w) + \lambda \|w\|_1$$
- U minimumu mora važiti $0 \in \partial \mathcal{L}(w)$, odnosno:
$$\nabla_i E(w) + \lambda \text{sgn}(w) = 0, \;\; w_i \neq 0$$
$$|\nabla_i E(w)| \leq \lambda, \;\; w_i = 0$$

Primer – podgradijentni spust za ℓ_1 regularizaciju

- Postoji više podgradijenata, a bira se onaj koji daje pravac nabržeg spusta:

$$\nabla_i \mathcal{L}(w) \triangleq \begin{cases} \nabla_i E(w) + \lambda \text{sgn}(w_i) & w_i \neq 0 \\ \nabla_i E(w) + \lambda & w_i = 0,\ \nabla_i E(w) < -\lambda \\ \nabla_i E(w) - \lambda & w_i = 0,\ \nabla_i E(w) > \lambda \\ 0 & w_i = 0,\ |\nabla_i E(w)| \leq \lambda \end{cases}$$

Primer – podgradijentni spust za ℓ_1 regularizaciju

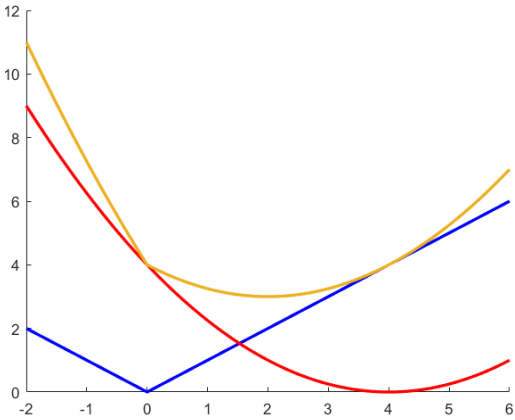
- Slučaj
$$\nabla_i E(w) + \lambda \text{sgn}(w_i) \text{ za } w_i \neq 0$$
sledi iz diferencijabilnosti za $w_i \neq 0$
- Slučaj
$$0 \text{ za } w_i = 0,\ |\nabla_i E(w)| \leq \lambda$$
sledi iz zadovoljenosti uslova optimalnosti:
$$|\nabla_i E(w)| \leq \lambda,\ w_i = 0$$

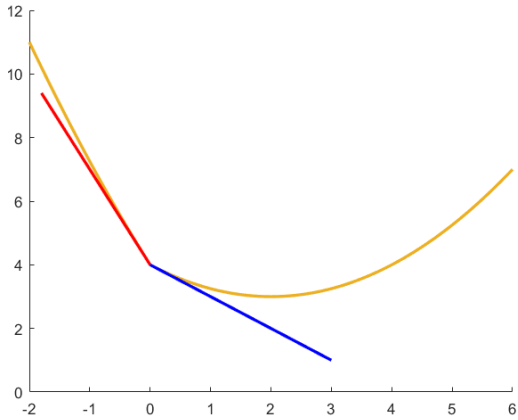
Primer – podgradijentni spust za ℓ_1 regularizaciju

- Slučaj

$$\nabla_i E(w) + \lambda \text{ za } w_i = 0,\ \nabla_i E(w) < -\lambda$$

proizilazi iz izbora smera optimizacije





Diferencijabilnost drugog reda

- Hesijan $\nabla^2 E(w)$ je matrica parcijalnih izvoda drugog reda
- Pruža informaciju o lokalnoj zakrivljenosti
- Na osnovu njega je moguće izabrati kraću putanju nego na osnovu gradijenta

Njutnov metod

- Njutnov metod za funkcije jedne promenljive:

$$w_{k+1} = w_k - \frac{f'(w_k)}{f''(w_k)}$$

- Njutnov metod za funkcije više promenljivih:

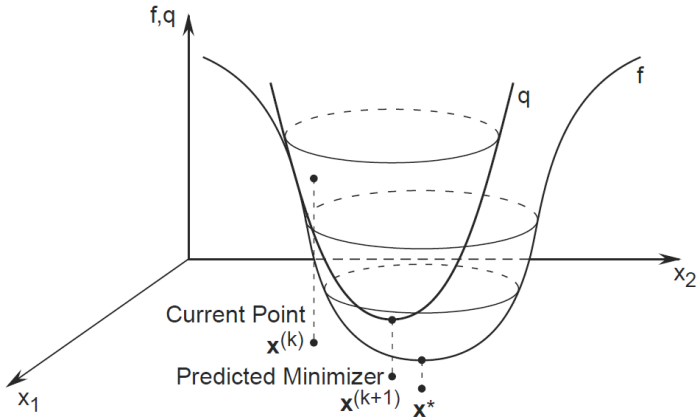
$$w_{k+1} = w_k - \nabla^2 E(w_k)^{-1} \nabla E(w_k)$$

- Svaki korak minimizuje kvadratnu aproksimaciju funkcije
- Za jako konveksne fukcije sa Lipšic neprekidnim Hesijanom, greška je reda $O\left(c^{2^k}\right)$ za neko $0 < c < 1$

Njutnov metod za kvadratne funkcije

- Kvadratna greška:
$$E(w) = c + b^T w + \frac{1}{2} w^T A w$$
- Gradijent je $\nabla E(w) = b + Aw$, a hesijan je $\nabla^2 E(w) = A$
- Korak Njutnovog metoda daje:
$$w_{k+1} = w_k - A^{-1}(b + Aw_k) = -A^{-1}b$$
- Provera optimalnosti na osnovu gradijenta:
$$\nabla E(-A^{-1}b) = b + A(-A^{-1}b) = 0$$

Njutnov metod i kvadratna aproksimacija



Slika: Nepoznat autor

Problemi vezani za Njutnov metod

- Njutnov metod:
$$w_{k+1} = w_k - \frac{f'(w_k)}{f''(w_k)}$$
- Traži nulu gradijenta, tako da maksimumi i sedlaste tačke predstavljaju problem
- Stoga zahteva strogu konveksnost (ne mešati sa jakom)
- Inverzija matrice
- Nije pogodan za probleme visoke dimenzionalnosti

Kvazi-Njutnovi metodi

- Hesijan može biti nedostupan ili previše veliki za čuvanje i inverziju
- Ideja je aproksimirati $\nabla^2 E(w_k)^{-1}$ na osnovu gradijenata, tako da se inverzija ne vrši
- Aproksimacija se efikasno popravlja u svakom koraku
- Najpoznatiji predstavnik je metod BFGS

BFGS (1)

- Kvazi-Njutnov metod:
$$w_{k+1} = w_k - H_k^{-1} \nabla E(w_k)$$
- Može se razumeti kao inkrementalno popravljanje kvadratnih modela funkcije koja se minimizuje
- Aproksimacija H_k^{-1} mora biti simetrična

BFGS (2)

- Kvadratna aproksimacija greške u okolini w_k

$$\bar{E}(w) = E(w_k) + \nabla E(w_k)^T (w - w_k) + \frac{1}{2}(w - w_k)^T H_k (w - w_k)$$

- Gradijent:

$$\nabla \bar{E}(w) = \nabla E(w_k) + H_k (w - w_k)$$

- Gradijenti funkcija $\bar{E}(w)$ i $E(w)$ se slažu u tački w_k
- Zahtevamo da se slažu i u tački w_{k-1}:

$$\nabla E(w_k) + H_k (w_{k-1} - w_k) = \nabla E(w_{k-1})$$

$$H_k{}^{-1} (\nabla E(w_k) - \nabla E(w_{k-1})) = w_k - w_{k-1}$$

BFGS (3)

- Zahtevajmo da je promena aproksimacije u odnosu na H_{k-1}^{-1} minimalna

$$\min_{H^{-1}} \|H^{-1} - H_{k-1}^{-1}\|_2^2$$

$$s.t.\ H^{-1}(\nabla E(w_k) - \nabla E(w_{k-1})) = w_k - w_{k-1}$$

$$H^{-1\,T} = H^{-1}$$

- Postoji analitičko rešenje ovog minimizacionog problema
- Za jako konveksne funkcije sa Lipšic neprekidnim Hesijanom greška opada brže od $O(c^k)$, ali sporije od $O(c^{2^k})$ za neko $0 < c < 1$

L-BFGS

- Skladištenje hesijana nije rešeno metodom BFGS
- L-BFGS ne čuva aproksimaciju eksplicitno, već čuva ograničenu istoriju razlika gradijenata i razlika parametara u uzastopnim koracima
- Vrlo se često koristi u mašinskom učenju

Konveksnost

- Konveksnost dopušta efikasnije optimizacione algoritme
- Garantuje jedinstven globalni minimum
- Vrlo poželjna, ali u praksi nije uvek moguća, niti su uvek moguće ili poznate dobre konveksne aproksimacije

Nekonveksnost

- Nema garancija za jedinstven optimum
- Čak i ako postoji, optimizacija je komplikovanija (recimo, Njutnov algoritam ne mora naći ni lokalni minimum)
- Sekvencijalno kvadratno programiranje (SQP) rešava niz kvadratnih aproksimacija (dopušta ograničenja)
- Konveksno-konkavna procedura za sumu konveksnih i konkavnih funkcija rešava niz aproksimacija u kojima linearizuje konkavni deo (dopušta ograničenja)
- Računski zahtevnija

Problem lokalnih minimuma

- Neuronske mreže su vrlo nekonveksne, ali ipak daju sjajne rezultate u mnogim primenama
- Zar ne bi lokalni minimumi trebalo da predstavljaju veliki problem za njihovu optimizaciju?
- Manji problem u mašinskom učenju, nego što se do skora pretpostavljalo

Kakve su šanse da tačka bude lokalni minimum?

- Da bi stacionarna tačka bila lokalni minimum, sve sopstvene vrednosti Hesijana u toj tački moraju biti pozitivne
- Neka su pozitivne i negativne vrednosti jednako verovatne
- Verovatnoća da tačka bude minimum se eksponencijalno smanjuje sa dimenzijom Hesijana (brojem atributa)
- Današnji praktični problemi su obično visokodimenzionalni
- Ne očekuje se mnogo lokalnih minimuma u slučaju funkcija nad visokodimenzionalnim prostorima!
- Naravno, ipak se javljaju

Koliko su vrednosti lokalnih minimuma loše?

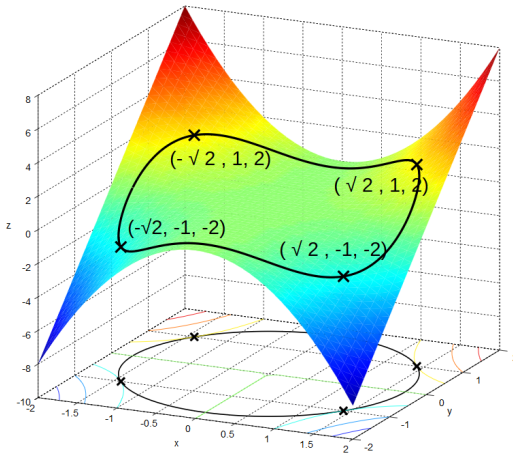
- Za neuronske mreže, pod blago pojednostavljenim uslovima, dokazano je da što je veća vrednost funkcije u stacionarnoj tački, manja je verovatnoća da je ta tačka minimum funkcije
- Većina lokalnih minimuma nisu mnogo lošiji od globalnog minimuma!

Sedlene tačke

- U svetlu prethodne diskusije, sedlene tačke mogu predstavljati veći problem za optimizaciju od lokalnih minimuma
- Metode prvog reda mogu biti sporije u njihovoj okolini
- Metode poput Njutnove traže nulu gradijenta, tako da ih privlače sedlene tačke
- Trenutno se radi na metodama nalik Njutnovoj koje će biti otporne na prisustvo sedlenih tačaka

Ograničenja

- Ograničenja sužavaju skup dopustivih rešenja
- U prisustvu ograničenja, korak gradijentnog spusta može voditi van skupa dopustivih rešenja
- Konveksnost i diferencijabilnost su važne i u kontekstu ograničenja



Slika: Vikipedijin članak o Lagranžovim množiocima

Projektovani gradijentni spust

- $\mathcal{C}$ je konveksan skup dopustivih rešenja
- Ponavljati dok postupak ne iskonvergira:

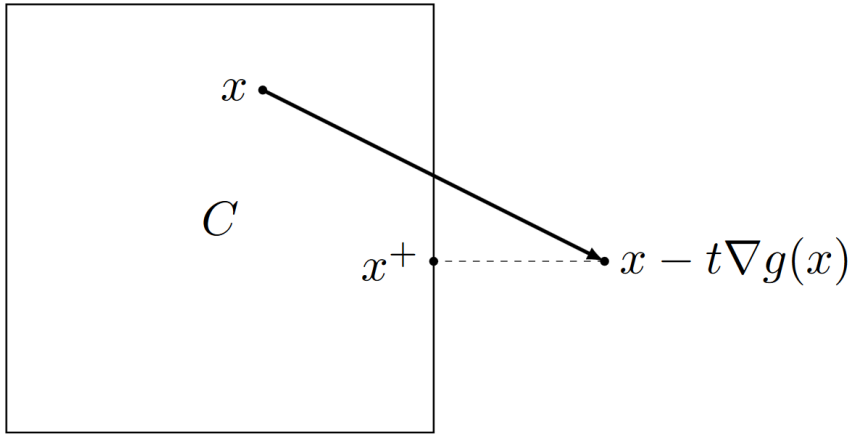
$$w_{k+1} = \mathcal{P}_{\mathcal{C}}(w_k - \mu_k \nabla E(w_k))$$

- $\mathcal{P}_{\mathcal{C}}(v)$ označava operator euklidske projekcije na skup dopustivih rešenja $\mathcal{C}$:

$$\mathcal{P}_{\mathcal{C}}(u) = \text{argmin}_{v \in \mathcal{C}} \|u - v\|_2$$

- Upotrebljiv kada se projekcija može efikasno izračunati (npr. u linearnom vremenu u slučaju ℓ_1 regularizacije)

Projektovani gradijentni spust



Slika: Lieven Vandenberghe, materijal za kurs

Metod unutrašnje tačke

- Koristi kaznene funkcije (npr. logaritamsku barijeru) kako bi kaznio rešenja koja prilaze granici dopustivog skupa rešenja
- U toku optimizacije, aproksimacija skupa dopustivih rešenja postaje sve preciznija
- U slučaju nekonveksnih problema, kombinuje se sa sekvencijalnim kvadratnim programiranjem (LOQO)
- Postoje efikasni algoritmi (ali ne uporedivi sa algoritmima za konveksne probleme bez ograničenja)
- Moguće najbolji izbor u slučaju problema sa ograničenjima u slučaju nekonveksnosti ili kad se projekcija ne može efikasno izračunati

Logaritamska barijera

- Problem sa ograničenjima:

$$\min_w E(w)$$

$$s.t.\ g_i(w) \geq 0, \quad i = 1, ..., m$$

- Problem bez ograničenja sa logaritamskom barijerom:

$$\min_w E(w) + \frac{1}{\mu} \sum_{i=1}^{m} -\log(g_i(w)))$$

- Problem se rešava iznova i iznova za sve veće vrednosti parametra μ dok postupak ne iskonvergira

Aproksimacije

- Konveksne aproksimacije nekonveksnih problema:
  - Funkcija greške u obliku šarke umesto greške klasifikacije
  - ℓ_1 umesto broja nenula koeficijenata
  - ...
- Diferencijabilne aproksimacije nediferencijabilnih problema (npr. glatki maksimum, glatka ℓ_1 norma)
- Relaksacije problema zarad efikasnosti (npr. linearno programiranje umesto celobrojnog linearnog programiranja)

Primer – glatka ℓ_1 norma

$$\max_\mu(x_1, \ldots, x_m) = \log_\mu(\mu^{x_1} + \ldots + \mu^{x_m})$$

$$|x| \approx |x|_\mu = \max_\mu(x, -x) = \log_\mu(\mu^x + \mu^{-x})$$

$$\|w\|_\mu = \sum_{i=1}^{n} |w_i|_\mu$$

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Procena kvaliteta i izbor modela

- Procena kvaliteta modela se bavi ocenom greške predviđanja modela
- Izbor modela se bavi izborom jednog od više mogućih modela
- Izbor modela se zasniva na proceni kvaliteta modela
- Ipak, veza ne mora biti trivijalna

Na čemu se zasnivaju procena kvaliteta i izbor modela?

- Mere kvaliteta (npr. preciznost)
- Tehnike evaluacije i izbora (npr. unakrsna validacija)

Mere kvaliteta

- Zavise od problema
- Za klasifikaciju: preciznost, F_1 skor, AUC
- Za regresiju: srednjekvadratna greška i koeficijent determinacije (R^2)

Tačna i netačna predviđanja

	Pred. pozitivni	Pred. negativni
Pozitivni	Stvarno pozitivni	Lažno negativni
Negativni	Lažno pozitivni	Stvarno negativni

Preciznost klasifikacije

- $ACC = \frac{SP+SN}{SP+SN+LP+LN}$
- Varljiva u slučaju neizbalansiranih klasa

F_1 skor

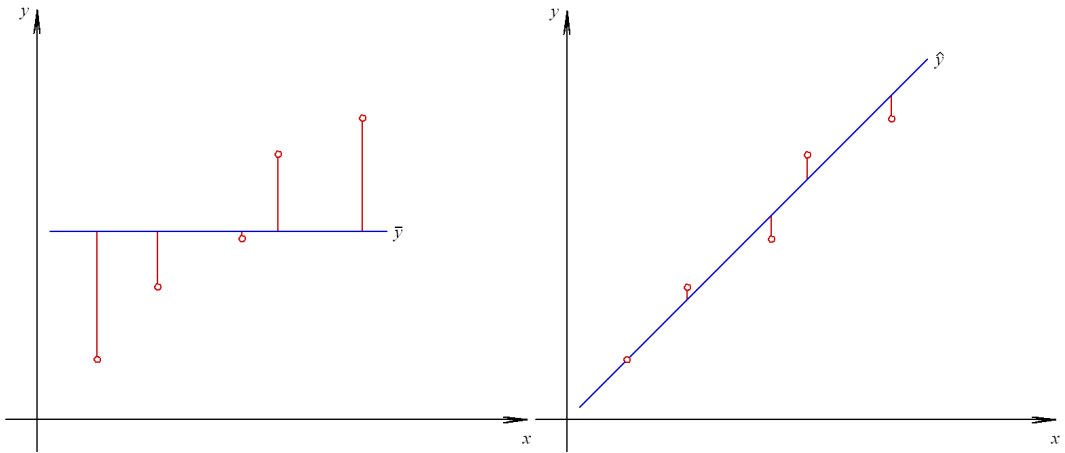
- Preciznost $P = \frac{SP}{SP+LP}$ (nije isto što i prethodno definisana preciznost)
- Odziv $R = \frac{SP}{SP+LN}$
- $F_1 = \frac{2PR}{P+R}$
- Kombinuje preciznost i odziv, ali je bliže manjoj od te dve mere
- Nije osetljiva na neizbalansiranost klasa

Koren srednjekvadratne greške

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_w(x_i) - y_i)^2}$$

- Poput standardne devijacije, ali ne u odnosu na prosek, već u odnosu na model
- Izražava se u istim jedinicama kao i ciljna promenljiva
- Koristi se da kvantifikuje veličinu greške
- Posebno korisna ukoliko znamo prihvatljivu veličinu greške u razmatranoj primeni

Koren srednjekvadratne greške



Slika: P. Janičić, M. Nikolić, Veštačka inteligencija, u pripremi.

Koeficijent determinacije R^2

$$R^2 = 1 - \frac{MSE}{Var} = 1 - \frac{\sum_{i=1}^{N}(f_w(x_i) - y_i)^2}{\sum_{i=1}^{N}(\bar{y} - y_i)^2}$$

- Meri udeo varijanse ciljne promenljive koji je objašnjen modelom
- U rasponu $(-\infty, 1]$
- Koristi se kao mera kvaliteta učenja
- Pogodnija za poređenja nego kao apsolutna mera

Tehnike evaluacije

- Variraju po složenosti zavisno od:
  - Konfigurabilnosti algoritma
  - Željenog kvaliteta ocene

Glavno načelo procene kvaliteta modela

- *Podaci korišćeni za procenu kvaliteta modela ni na koji način ne smeju biti upotrebljeni prilikom treninga*
- Deluje jednostavno, ali se u praksi ispostavlja kao vrlo pipavo

Nekonfigurabilan slučaj

- Pretpostavlja se da algoritam nije konfigurabilan ili da je konfiguracija fiksirana
- To nije realističan scenario
- Ako se naučeni model pokaže loše, u iskušenju smo da vršimo neke izmene i u tom slučaju naredni metodi procene kvaliteta nisu validni!!

Izbor modela za nekonfigurabilan slučaj

- Pošto nema različitih konfiguracija, nema ni većeg broja modela iz kojeg se može birati, pa je izbor praktično trivijalan
- Ipak, postavlja se pitanje na kojim podacima treba trenirati?

Izbor modela za nekonfigurabilan slučaj

- Pošto nema različitih konfiguracija, nema ni većeg broja modela iz kojeg se može birati, pa je izbor praktično trivijalan
- Ipak, postavlja se pitanje na kojim podacima treba trenirati?
- Model M, za buduću upotrebu, se trenira na celom skupu podataka

Procena kvaliteta pomoću trening i test skupa

- Podaci se dele na dva skupa
- Jedan skup se koristi za treniranje modela M' koji služi kao aproksimacija modela M
- Drugi se koristi za procenu greške koju model M' pravi prilikom predviđanja, a smatra se da je ta greška dobra aproksimacija greške modela M

x_1	x_2	x_3	y
1	9	0	8
0	6	2	1
1	3	1	5
4	9	7	6
1	1	6	7
7	2	3	4
2	9	9	9
3	3	4	6
7	2	1	7
6	5	1	5

Problemi vezani za procenu pomoću trening i test skupa

- Sve je u redu ukoliko je skup podataka vrlo veliki i reprezentativan, ali u suprotnom...
- Kako izvršiti podelu?
- Šta ako su raspodele trening i test skupa različite?
- Velika varijansa ocene greške

Procena kvaliteta K-strukom unakrsnom validacijom

- Podaci se dele na K slojeva (tj. delova)
- Za svaki sloj
  - Trenira se model na preostalih $K-1$ slojeva
  - Vrše se predviđanja dobijenim modelom na izabranom sloju
- Računa se ocena greške

x_1	x_2	x_3	y
1	9	0	8
0	6	2	1
1	3	1	5
4	9	7	6
1	1	6	7
7	2	3	4
2	9	9	9
3	3	4	6
7	2	1	7
6	5	1	5

Problemi vezani za procenu unakrsnom validacijom

- Računska zahtevnost
- Kako izabrati broj slojeva?
  - Koristiti 5 ili 10 slojeva
  - Ne koristiti jednočlane slojeve (eng. leave one out), pošto je takva ocena greške optimistična
- Ne računati ocene greške za svaki sloj, pa ih uprosečavati (ne radi za nelinearne mere poput R^2)
- Sve instance se koriste u proceni kvaliteta, pa je pouzdanija, ali i dalje jedan sloj ne mora imati istu raspodelu kao preostali

Još jedno načelo procene kvaliteta modela

- Trening i test skup treba da imaju istu raspodelu kao i buduća opažanja
- Deluje pipavo i jeste pipavo
- Kako ublažiti ovaj problem?

Još jedno načelo procene kvaliteta modela

- Trening i test skup treba da imaju istu raspodelu kao i buduća opažanja
- Deluje pipavo i jeste pipavo
- Kako ublažiti ovaj problem?
  - Koristiti velike količine podatka
  - Koristiti napredne tehnike uzorkovanja
  - Stratifikacija

Stratifikacija

- Prilikom deljenja podataka, obezbediti da delovi imaju istu raspodelu kao i ceo skup podataka
- Teško za male skupove podataka
- Pojednostavljena varijanta: očuvati raspodelu ciljne promenljive

Stratifikacija u odnosu na ciljnu promenljivu

- Sortirati podatke u odnosu na ciljnu promenljivu
- Ako je K broj delova, neka instance sa indeksima $i + j * K$ čine deo $\mathcal{P}_i$ za $i = 1, \ldots, K$ i $j = 0, 1, \ldots$

x_1	x_2	x_3	y
0	6	2	1
7	2	3	4
1	3	1	5
6	5	1	5
4	9	7	6
3	3	4	6
1	1	6	7
7	2	1	7
1	9	0	8
2	9	9	9

Da li je algoritam konfigurabilan?

- Algoritmi mašinskog učenja se obično navode vrednostima metaparametara
  - Linearna i logistička regresija: regularizacioni parametar
  - SVM: cena grešaka, parametri kernela, ...
  - Neuronske mreže: regularizacioni parametar, parametar vezan za metodu inercije ...
  - ...
- Pre učenja, moguće je izabrati podskup atributa
- Neuronske mreže mogu imati različite arhitekture
- Metodi zasnovani na kernelima mogu koristiti različite kernele
- ...
- Uzimamo u obzir algoritamske konfiguracije (vrednosti metaparametara, atributi, arhitekture, kerneli, ...)

Šta ako je konfigurabilan?

- Različite konfiguracije daju različite modele
- Kako izabrati adekvatnu konfiguraciju, a time i model?

Šta ako je konfigurabilan?

- Različite konfiguracije daju različite modele
- Kako izabrati adekvatnu konfiguraciju, a time i model?
- Jednostavno: izvršiti evaluaciju modela dobijenih za različite konfiguracije i izabrati najbolji
- Koja je ocena greške predviđanja tog modela?

Šta ako je konfigurabilan?

- Različite konfiguracije daju različite modele
- Kako izabrati adekvatnu konfiguraciju, a time i model?
- Jednostavno: izvršiti evaluaciju modela dobijenih za različite konfiguracije i izabrati najbolji
- Koja je ocena greške predviđanja tog modela?
- Nije jednostavno!

Procena kvaliteta i izbor modela na pogrešan način

- Evaluira se svaka konfiguracija unakrsnom validacijom
- Bira se najbolja konfiguracija i prijavljuje se upravo dobijena procena kvaliteta modela
- Trenira se finalni model pomoću najbolje konfiguracije na celom skupu podataka

U čemu je greška?

- Uobičajeno opravdanje datog postupka bi bilo: pošto se koristi unakrsna validacija, nikad se ne vrši trening na instancama koji se koriste za testiranje

U čemu je greška?

- Uobičajeno opravdanje datog postupka bi bilo: pošto se koristi unakrsna validacija, nikad se ne vrši trening na instancama koji se koriste za testiranje
- Ali da li je tako?

U čemu je greška?

- Uobičajeno opravdanje datog postupka bi bilo: pošto se koristi unakrsna validacija, nikad se ne vrši trening na instancama koji se koriste za testiranje
- Ali da li je tako?
- Prilikom izbora najbolje konfiguracije, oslonili smo se na informaciju dobijenu korišćenjem celog skupa podataka, a izbor najbolje konfiguracije je deo treninga, pošto se direktno odražava na rezultujući model!

Izbor modela pomoću validacionog skupa

- Podaci se dele na trening i validacioni skup
- Za svaku konfiguraciju
  - Trenira se model za tu konfiguraciju na trening skupu
  - Vrši se ocena greške predviđanja modela na validacionom skupu
- Bira se konfiguracija koja daje najmanju grešku na validacionom skupu
- Trenira se finalni model pomoću najbolje konfiguracije na celom skupu podataka

Ocena greške pomoću validacionog i test skupa

- Podaci se dele na trening i test skup
- Na trening skupu se izvrši izbor modela i pridružene konfiguracije pomoću validacionog skupa
- Za tu konfiguraciju se trenira model na celom trening skupu
- Vrši se ocena greške tog modela na test skupu i ona se prijavljuje kao ocena kvaliteta modela

x_1	x_2	x_3	y
1	9	0	8
0	6	2	1
1	3	1	5
4	9	7	6
1	1	6	7
7	2	3	4
2	9	9	9
3	3	4	6
7	2	1	7
6	5	1	5

Izbora modela pomoću unakrsne validacije

- Za svaku konfiguraciju
  - Vrši se ocena greške predviđanja modela unakrsnom validacijom
- Bira se konfiguracija koja daje najmanju grešku pri unakrsnoj validaciji
- Trenira se finalni model pomoću najbolje konfiguracije na celom skupu podataka

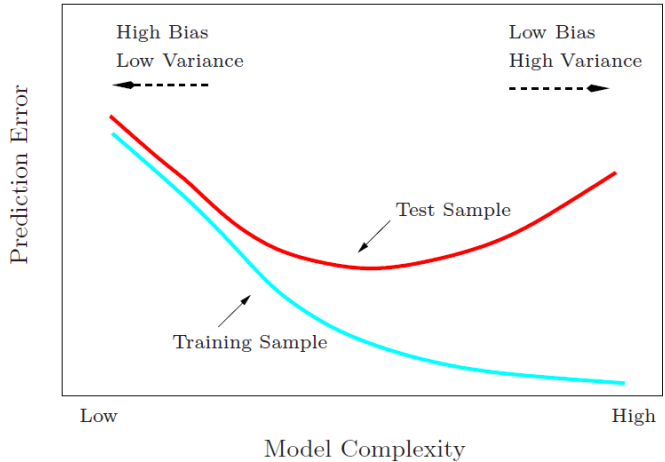
Procenu kvaliteta modela ugnežđenom K-strukom unakrsnom validacijom

- Dele se podaci na K slojeva
- Za svaki sloj
  - Vrši se ocena greške svih konfiguracija na preostalih $K-1$ slojeva K-strukom unakrsnom validacijom
  - Bira se konfiguracija sa najmanjom greškom
  - Trenira se model pomoću te konfiguracije na preostalih $K-1$ slojeva
  - Vrši se predviđanje dobijenim modelom na izabranom sloju
- Računa se ocena greške

Šta ako kvalitet modela nije dobar?

- Nedovoljna prilagođenost modela (visoko sistematsko odstupanje)?
- Preprilagođenost modela (visoka varijansa)?

Nagodba između sistematskog odstupanja i varijanse



Slika: T. Hastie, R. Tibshirani, J. Friedman, Elements of Statistical Learning, 2001.

Šta ako model ima visoko sistematsko odstupanje

- Koristiti prilagodljivije modele
- Koristiti niže vrednosti regularizacionog parametra
- Konstruisati nove atribute

Šta ako je varijansa modela visoka?

- Koristiti manje prilagodljive modele
- Koristiti tehnike za izbor atributa
- Koristiti više vrednosti regularizacionog parametra
- Koristiti više podataka

Nedostatak informativnih atributa

- Ukoliko atributi nisu dovoljno informativni, nijedan algoritam učenja ne može dati rezultate
- Proveriti koje klase se međusobno mešaju i proveriti da li se može očekivati da postojeći atributi diskriminišu između njih
- Proveriti da li su atributi korelirani sa ciljnom promenljivom pomoću koeficijenta korelacije i grafika vrednosti ciljne promenljive naspram vrednosti atributa

Pregled

Uopšteno o mašinskom učenju

Neformalan podsetnik verovatnoće i statistike

Teorijske osnove nadgledanog učenja

Popularni modeli i algoritmi nadgledanog učenja

Dizajn algoritama nadgledanog učenja

Procena kvaliteta i izbor modela

Finalni saveti

Finalni saveti

- Proučiti postojeće algoritme
- Ustanoviti zašto ne daju dobre rezultate
- Proveriti da li je forma modela adekvatna
- Proveriti da li je funkcija greške adekvatna
- Proveriti da li se regularizacija može izmeniti kako bi nametnula adekvatnu strukturu modela
- Proveriti da li se optimizacioni metod može zameniti bržim
- Proveriti da li se optimizacioni problem može aproksimirati
- Koristiti ugnežđenu unakrsnu validaciju za procenu kvaliteta modela ukoliko je količina podataka mala ili trening, validaciju i testiranje ukoliko je na raspolaganju puno podataka
- Analizirati dobijeni model

Knjige

- C. Bishop, Pattern Recognition and Machine Learning
- T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning
- K. Murphy, Machine Learning, A Probabilistic Perspective
- M. Magdon-Ismail, Y. Abu-Mostafa, Learning from Data: A Short course
- S. Shalev-Schwartz, S. Ben-David, Understanding Machine Learning, From Theory to Algorithms
- V. Vapnik, Statistical Learning Theory

Knjige

- I. Goodfellow, Y. Bengio, A. Courville, Deep Learning
- B. Schölkopf, A. Smola, Learning With Kernels, Support Vector Machines, Regularization, Optimization, and Beyond
- R. Sutton, A. Barto, Reinforcement Learning: An Introduction
- S. Boyd, L. Vandenberghe, Convex Optimization
- S. Sra, S. Nowozin, S. Wright, Optimization for Machine Learning
- A. Nemirovski, Efficient Methods in Convex Programming

Radovi

- A. Turing, Computing Machinery and Intelligence, 1950
- P. Domingos, A Few Useful Things to Know about Machine Learning, 2012
- K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is "Nearest Neighbor" Meaningful, 1998
- C. Aggarwal, A. Hinnenburg, D. Keim, On the Surprising Behavior of Distance Metrics in High Dimensional Space, 2001
- T. Mikolov, I. Sutskever, K Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality, 2013
- A. Graves, G. Wayne, I. Danihelka, Neural Turing Machines, 2014
- R. Tibshirani, Regression Shrinkage and Selection via the Lasso, 1996

Radovi

- M. Schmidt, G. Fung, R. Rosales, Fast Optimization Methods for L_1 Regularization: A Comparative Study and Two New Approaches, 2007
- J. Ye, J. Liu, Sparse Methods for Biomedical Data, 2012
- Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Gangauli, Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization
- M. Nikolić, F. Marić, P. Janičić, Simple Algorithm Portfolio for SAT
- V. Alabau, J. Andrés, F. Casacuberta, J. Civera, J. García-Hernández, A. Giménez, A. Juan, A. Sanchis, E. Vidal, The naive Bayes model, generalisations and applications

Linkovi

- http://archive.ics.uci.edu/ml/
- https://github.com/fchollet/keras
- http://www.yelab.net/software/SLEP/
- http://statweb.stanford.edu/~tibs/ElemStatLearn/
- http://statweb.stanford.edu/~tibs/lasso.html
- http://www.seas.ucla.edu/~vandenbe/ee236c.html
- http://www.stat.cmu.edu/~ryantibs/convexopt/
- http://stanford.edu/~boyd/cvxbook/