

Pronalaženje skrivenog znanja

Elektrotehnički fakultet Univerziteta u Beogradu

Master akademske studije

modul Računarska tehnika i informatika

2017/2018

Linearna regresija

Vuk Batanović, Elektrotehnički fakultet Univerziteta u Beogradu

Linearna regresija

- ▶ Osnovni tip regresije - predviđanja kontinualnih numeričkih vrednosti
- ▶ Prvi oblik linearne regresije predstavljen 1805. godine
- ▶ Prosta/jednostruka linearna regresija - postoji samo jedna ulazna promenljiva/odlika x
- ▶ Višestruka linearna regresija - postoji više (n) ulaznih promenljivih/odlika x_1, x_2, \dots, x_n
- ▶ Univarijantna linearna regresija - samo jedna izlazna promenljiva y
- ▶ Multivarijantna linearna regresija - više izlaznih promenljivih

Linearna regresija

- ▶ Linearna regresija pretpostavlja da zavisnost y od x ima oblik linearne kombinacije odlika (i njihovih težina)
- ▶ Dve glavne vrste upotrebe linearne regresije su:
 - ▶ Predviđanje vrednosti izlaza y za nove podatke na osnovu poznatih vrednosti tog izlaza na starim podacima
 - ▶ Analiza stepena povezanosti između vrednosti izlaza y i vrednosti promenljivih x_1, x_2, \dots, x_n
- ▶ Široko korišćena i u prirodnim i u društvenim naukama

Jednostruka univarijantna linearna regresija

- ▶ Hipoteza jednostruke univarijantne linearne regresije:

$$h(x) = w_0 + w_1x$$

- ▶ w_0, w_1 su težine - parametri modela (negde se označavaju sa θ_0, θ_1)
- ▶ Obučavanje modela predstavlja odabir optimalnih vrednosti parametara modela tako da $h(x)$ bude blisko stvarnom y za sve ulazne parove (x, y)
- ▶ Da bi se model obučio neophodno je odabrati funkciju cene/greške koja govori koliko hipoteza (sa trenutnim vrednostima parametara) odskake od stvarnih vrednosti y

Funkcija gubitka i funkcija greške

- ▶ Funkcija gubitka $L(h(x), y)$ (engl. *loss function*) definiše meru odstupanja vrednosti hipoteze $h(x)$ (za neko x , sa trenutnim vrednostima parametara w) od tačne vrednosti y na pojedinačnom podatku
- ▶ Funkcija greške/cene $J(w)$ (engl. *error/cost function*) je prosek vrednosti funkcije gubitka na svim podacima iz posmatranog skupa:

$$J(w) = \frac{1}{m} \sum_{i=1}^m L(h(x^{(i)}), y^{(i)})$$

- ▶ gde je m broj podataka u skupu
- ▶ Obučavanje modela se svodi na odabir optimalnih vrednosti parametara modela w_0, \dots, w_n tako da funkcija greške $J(w)$ bude minimalna za parove (x, y) iz skupa za obučavanje

Funkcija greške u linearnoj regresiji

- ▶ U linearnoj regresiji se najčešće kao funkcija greške koristi prosek kvadrata odstupanja $h(x)$ od y (engl. *Mean Squared Error - MSE*):

$$J(w) = \frac{1}{2m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2$$

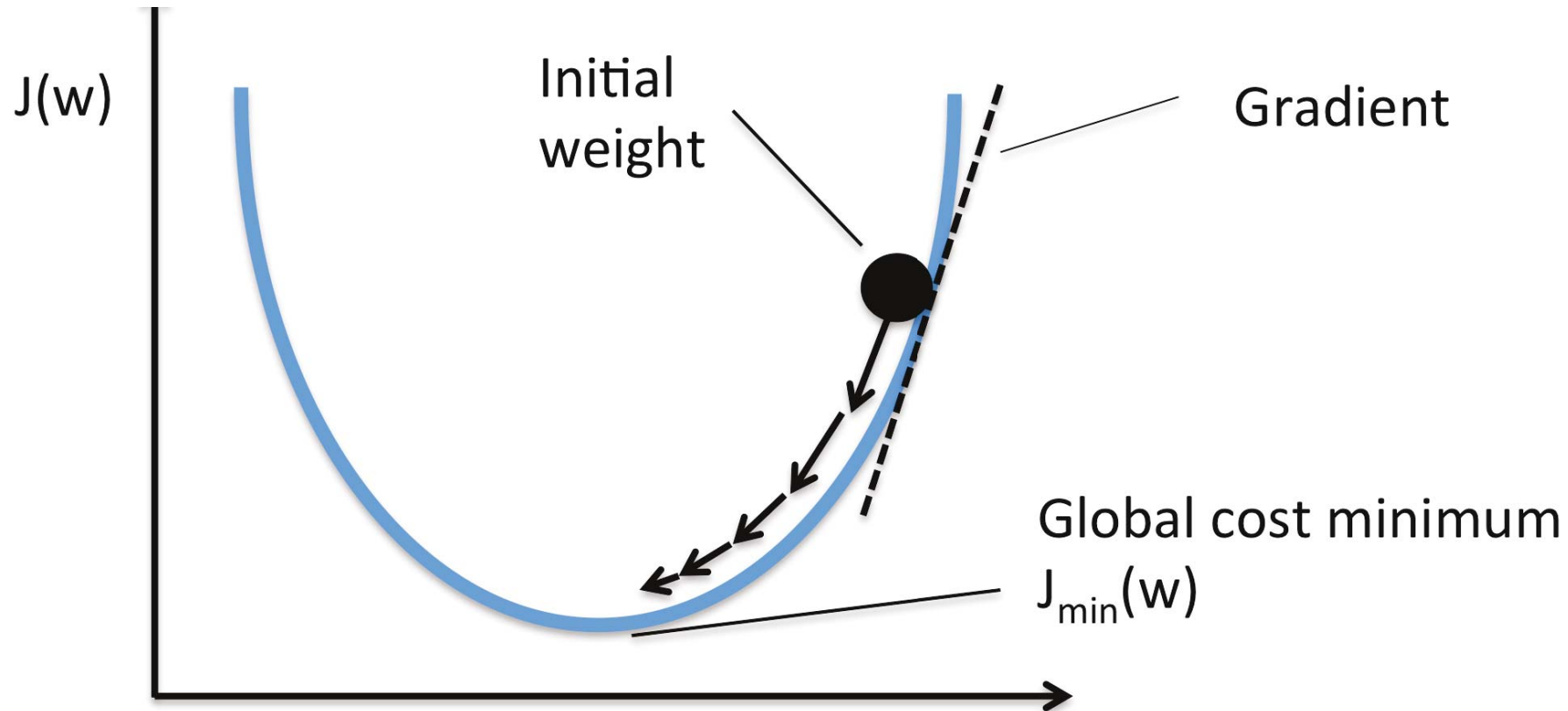
- ▶ Moguće je pronaći minimum analitički - podrazumeva računanje inverznih matrica
 - ▶ Može biti preskupa operacija kada je broj korišćenih odlika veliki

Obučavanje u linearnoj regresiji

- ▶ U praksi se najčešće za obučavanje modela koristi metoda gradijentnog spusta
- ▶ Gradijentni spust je iterativni metod traženja minimuma funkcije
- ▶ Zasniva se na korišćenju gradijenta - generalizacije izvoda na veći broj promenljivih
- ▶ Gradijent je vektor čiji su elementi parcijalni izvodi funkcije
- ▶ Gradijent funkcije greške:

$$\nabla J(w) = \left(\frac{\partial J(w)}{\partial w_0}, \frac{\partial J(w)}{\partial w_1}, \dots, \frac{\partial J(w)}{\partial w_n} \right)$$

- ▶ Parametre w_0, \dots, w_n treba menjati u smeru opadanja funkcije greške



Ilustracija gradijentnog spusta

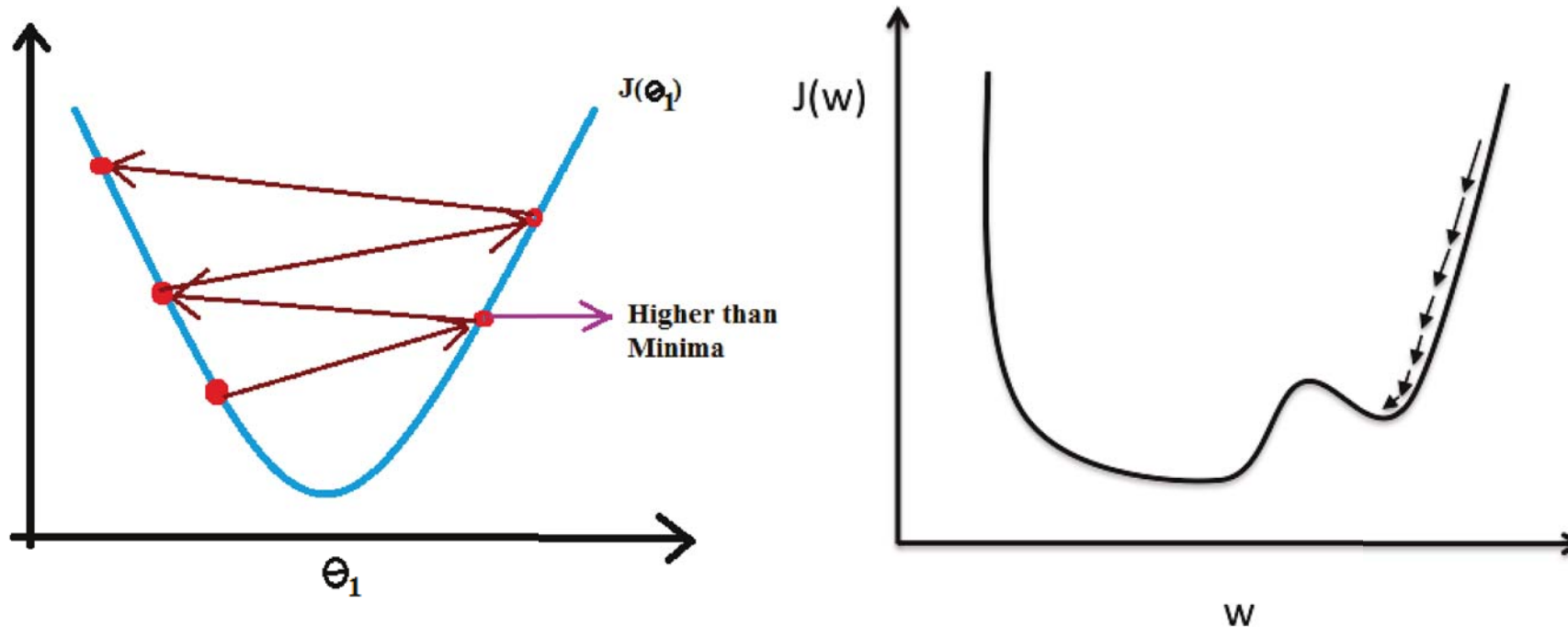
Slika preuzeta sa: <http://sebastianraschka.com/faq/docs/closed-form-vs-gd.html>

Gradijentni spust

- ▶ Izraz za ažuriranje težinskih vrednosti w :

$$w_i := w_i - \alpha \frac{\partial J(w)}{\partial w_i}$$

- ▶ α je pozitivan broj koji predstavlja brzinu učenja (engl. *learning rate*)
- ona određuje koliko će se veliki skokovi niz gradijent praviti pri obučavanju
- ▶ Brzina učenja mora da bude pažljivo odabrana
 - ▶ Kada je α premalo učenje je veoma sporo tj. ima veliki broj koraka. U zavisnosti od oblika funkcije greške, moguće je i zaglavljanje u lokalnim minimumima
 - ▶ Kada je α preveliko optimizacija može da preskoči minimum ili čak i da divergira



Ilustracija uticaja prevelike i premale brzine učenja na obučavanje modela

Slike preuzete sa:

<http://wingshore.wordpress.com/2014/11/19/linear-regression-in-one-variable-gradient-descent-contd/>

http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

Ažuriranje vrednosti parametara (primer jednostruke regresije)

- ▶ Obavezno treba ažurirati sve težinske vrednosti odjednom, na primer:

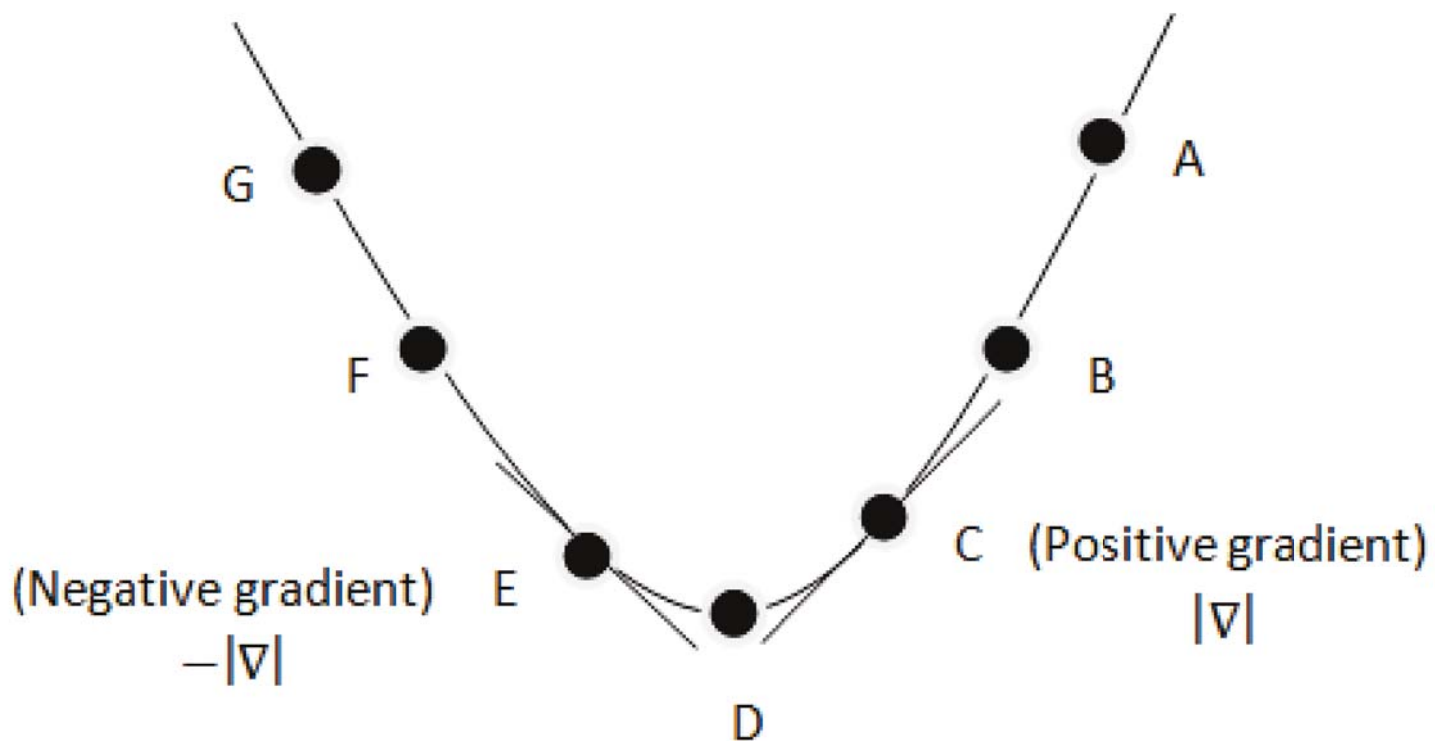
$$temp_0 := w_0 - \alpha \frac{\partial J(w)}{\partial w_0}$$

$$temp_1 := w_1 - \alpha \frac{\partial J(w)}{\partial w_1}$$

$$w_0 := temp_0$$

$$w_1 := temp_1$$

- ▶ U suprotnom, promena jednog parametra utiče na parcijalne izvode po drugim parametrima - više se ne vrši spust u smeru gradijenta



Ilustracija smeru spusta u zavisnosti od trenutnog položaja

Slika preuzeta sa: <http://stackoverflow.com/questions/21064030/gradient-descent-in-linear-regression>

Konvergencija

- ▶ Gradijentni spust može da konvergira i sa fiksnom vrednošću brzine učenja α

$$w_i := w_i - \alpha \frac{\partial J(w)}{\partial w_i}$$

- ▶ Kako se spust približava minimumu, vrednost parcijalnih izvoda se sama smanjuje, tako da automatski dolazi do konvergencije
- ▶ Vrednost izvoda u minimumu je jednaka nuli, tako da se parametri w više ne menjaju kad dostignu optimalne vrednosti
- ▶ Obučavanje se nekad obustavlja kada promena vrednosti parametara u jednom koraku postane manja od neke predefinisane vrednosti

Konvergencija

- ▶ Ukoliko je brzina učenja dobro odabrana, gradijentni spust će uvek konvergirati ka globalnom minimumu
 - ▶ Funkcija greške $J(w)$ je kvadratna funkcija što znači da je konveksna tj. ima samo jedan minimum koji je globalan
- ▶ Broj koraka potrebnih za konvergenciju zavisi od konkretnog zadatka koji se rešava i može da znatno varira
- ▶ Automatska provera konvergencije - da li se greška smanjuje u svakom koraku?
 - ▶ Za dovoljno malo α greška treba da se smanjuje u svakom koraku

Obučavanje u jednostrukoj linearnoj regresiji pomoću gradijentnog spusta

$$J(w) = \frac{1}{2m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2 = \frac{1}{2m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)})^2$$

$$\frac{\partial J(w)}{\partial w_0} = \frac{1}{m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)}) = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J(w)}{\partial w_1} = \frac{1}{m} \sum_{i=1}^m (w_0 + w_1 x^{(i)} - y^{(i)}) x^{(i)} = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x^{(i)}$$

Obučavanje u jednostrukoj linearnoj regresiji pomoću gradijentnog spusta

- ▶ Algoritam spusta - ponavljati do konvergencije:

$$w_0 := w_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})$$

$$w_1 := w_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})x^{(i)}$$

- ▶ Ovakav režim obučavanja se naziva grupni gradijentni spust (engl. *batch gradient descent*), jer se u svakom koraku spusta koristi svih m podataka iz skupa za obučavanje

Stohastički gradijentni spust

- ▶ Alternativan pristup je da se podaci iz skupa za obučavanje obrađuju jedan po jedan, tj. da se parametri w ažuriraju do konvergencije kao:

randomly shuffle the data

for $i = 1$ to m :

$$w_0 := w_0 - \alpha \frac{1}{m} (h(x^{(i)}) - y^{(i)})$$

$$w_1 := w_1 - \alpha \frac{1}{m} (h(x^{(i)}) - y^{(i)})x^{(i)}$$

- ▶ Ovakav režim obučavanja se naziva stohastički gradijentni spust (engl. *stochastic gradient descent* - SGD)

Grupni vs stohastički gradijentni spust

- ▶ Grupni gradijentni spust mora da prođe kroz ceo skup za obučavanje da bi napravio jedan korak
 - ▶ Ovo može da bude skupa operacija ako je broj podataka m veliki
- ▶ Stohastički gradijentni spust može da krene sa minimizacijom funkcije greške na osnovu samo jednog podatka
- ▶ Stohastički spust se često približi minimumu znatno brže nego grupni
- ▶ Stohastički spust ima veću varijansu nego grupni jer je svaki gradijent nešto drugačiji
 - ▶ Veća varijansa usporava konvergenciju
 - ▶ Zahteva da brzina učenja α vremenom opada

Grupni vs stohastički gradijentni spust

- ▶ Može se desiti da stohastički spust nikad ne konvergira sasvim, već da vrednosti w blago osciluju oko minimuma
 - ▶ Uglavnom nije problem jer su vrednosti oko minimuma obično dovoljno dobre
- ▶ Zbog efikasnosti se kod velikih skupova podataka preferira stohastički spust
- ▶ Stohastički spust je pogodan za *online* obučavanje - situacije kada novi podaci pristižu sekvencijalno
- ▶ Kao kompromis se često koristi obučavanje na manjim podskupovima podataka (engl. *mini-batch gradient descent*)

Višestruka univarijantna linearna regresija

- ▶ Hipoteza višestruke univarijantne linearne regresije:

$$h(x) = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

- ▶ w_0, w_1, \dots, w_n su težine - parametri modela
- ▶ x_0, x_1, \dots, x_n su odlike koje se koriste u modelu
- ▶ Funkcija greške sada ima oblik:

$$J(w) = \frac{1}{2m} \sum_{i=1}^m \left(w_0 + w_1x_1^{(i)} + w_2x_2^{(i)} + \dots + w_nx_n^{(i)} - y^{(i)} \right)^2$$

Obučavanje u višestrukoj linearnoj regresiji pomoću grupnog gradijentnog spusta

$$\frac{\partial J(w)}{\partial w_0} = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J(w)}{\partial w_1} = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\frac{\partial J(w)}{\partial w_2} = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

$$\vdots$$

Obučavanje u višestrukoj linearnoj regresiji pomoću grupnog gradijentnog spusta

- ▶ Algoritam spusta - ponavljati do konvergencije:

$$w_0 := w_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})$$

$$w_1 := w_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$w_2 := w_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

⋮

Obučavanje u višestrukoj linearnoj regresiji pomoću stohastičkog gradijentnog spusta

Algoritam spusta - ponavljati do konvergencije:

randomly shuffle the data

for $i = 1$ to m :

$$w_0 := w_0 - \alpha \frac{1}{m} (h(x^{(i)}) - y^{(i)})$$

$$w_1 := w_1 - \alpha \frac{1}{m} (h(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$w_2 := w_2 - \alpha \frac{1}{m} (h(x^{(i)}) - y^{(i)})x_2^{(i)}$$

⋮

Skaliranje vrednosti odlika

- ▶ U grupnom gradijentnom spustu težinske vrednosti se ažuriraju kao:

$$w_r := w_r - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_r^{(i)}$$

- ▶ a u stohastičkom kao:

$$w_r := w_r - \alpha \frac{1}{m} (h(x^{(i)}) - y^{(i)}) x_r^{(i)}$$

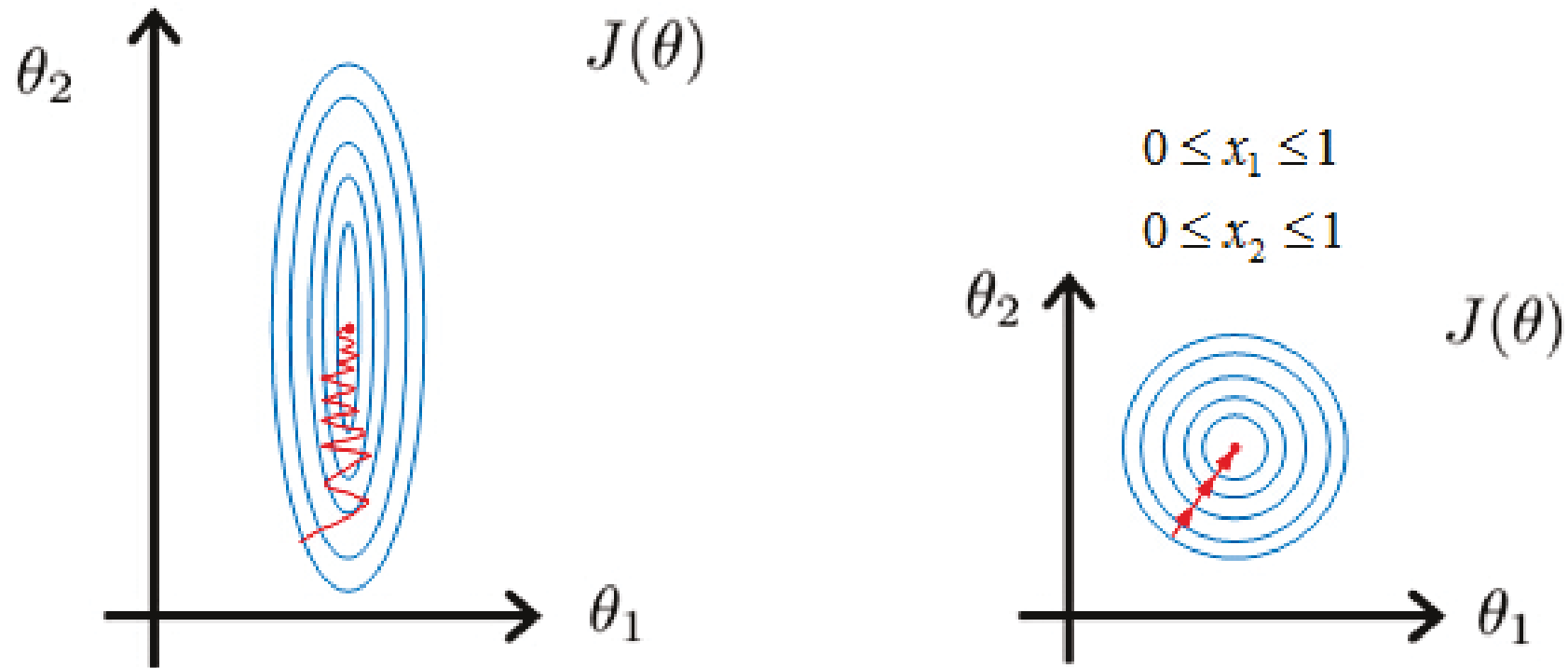
- ▶ Brzina učenja α je ista za sve težinske parametre, te stoga mora da se bira prema onom parametru gde je magnituda vrednosti odlike najveća (da bi se sprečila divergencija)
 - ▶ Veliki broj koraka u gradijentnom spustu i puno cik-cak kretanja, pošto neki težinski parametri konvergiraju brže od drugih

Skaliranje vrednosti odlika

- ▶ Gradijentni spust brže konvergira kada se vrednosti svih odlika nalaze u sličnim opsezima
 - ▶ Put do minimuma je brži i direktniji tj. ima manje cik-cak kretanja
- ▶ Skaliranje vrednosti odlika se obično radi tako da vrednosti budu u opsegu:

$$-1 \leq x \leq 1$$

- ▶ Moguće je i da se skaliranje vrši tako da srednja vrednost odlike bude jednaka nuli
- ▶ Nije neophodno da vrednosti svih odlika budu u apsolutno identičnim opsezima - manja odstupanja neće znatno uticati na brzinu spusta



Ilustracija efekta skaliranja vrednosti odlika

Slika preuzeta sa: <http://m.blog.csdn.net/article/details?id=50670674>

Polinomijalna regresija

- ▶ Moguće je stepenovati vrednosti odlika - time se dobija polinomijalna regresija
- ▶ Jednostruka polinomijalna regresija:

$$h(x) = w_0 + w_1x + w_2x^2$$

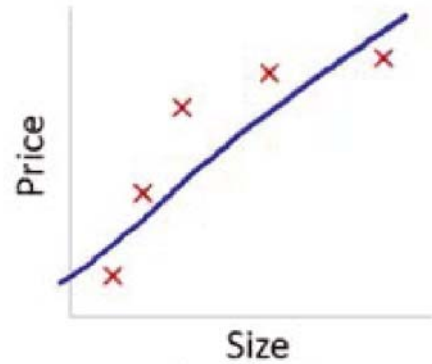
- ▶ Takođe je moguće kombinovati odlike, čime se dobija višestruka polinomijalna regresija:

$$h(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$$

- ▶ Skaliranje vrednosti odlika je naročito važno kod višestruke polinomijalne regresije

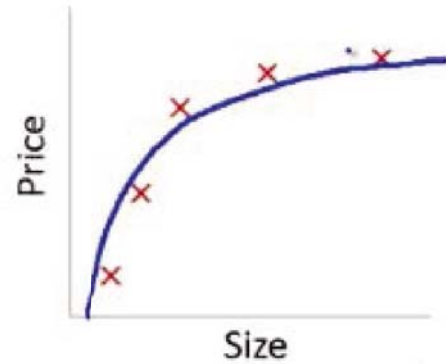
Polinomijalna regresija

- ▶ Polinomijalni modeli višeg stepena mogu zbog veće složenosti lako postati preterano prilagođeni podacima nad kojima se obučavaju
- ▶ Polinomijalni modeli su uglavnom dosta loši izvan opsega podataka korišćenih pri obučavanju - polinomijalne funkcije imaju „repove“
 - ▶ To ih čini veoma lošim za ekstrapolaciju
- ▶ Iako je regresija sada nelinearna, model je i dalje linearan jer predstavlja linearnu kombinaciju parametara



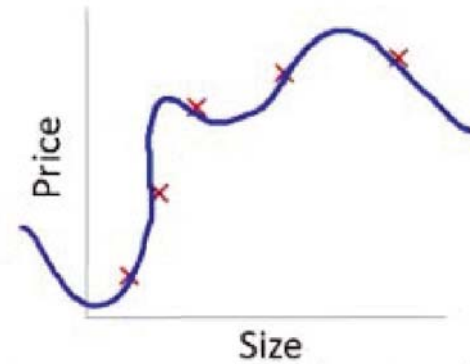
Size
 $\theta_0 + \theta_1 x$

High bias
(underfit)



Size
 $\theta_0 + \theta_1 x + \theta_2 x^2$

“Just right”



Size
 $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

Ilustracija ponašanja jednostruke linearne i polinomijalne regresije

Slika preuzeta sa: Andrew Ng, Machine Learning, Coursera

Regularizacija

- ▶ Regularizacija je način sprečavanja preterane prilagođenosti modela podacima (engl. *overfitting-a*) penalizovanjem izuzetno velikih magnituda nekih parametara w
 - ▶ Jako velike magnitude nekih parametara w nastaju kao posledica pokušaja modela da prilagodi oblik funkcije izlaza šumu i izuzecima u podacima za obučavanje (radi smanjenja funkcije greške)
 - ▶ Takvi modeli se u određenim situacijama (tj. za određene vrednosti ulaznog podatka) efektivno rukovode samo malim brojem odlika (ili njihovih kombinacija) koje imaju velike magnitude w
- ▶ Šire govoreći, pod regularizacijom se podrazumeva bilo koji postupak koji za cilj ima smanjenje greške generalizacije (tj. greške nad novim podacima/skupom za testiranje) a da se pritom ne poveća greška nad skupom za obučavanje

Regularizacija

- ▶ Primer - predikcija cene akcija na berzi
 - ▶ Preterano prilagođen model će možda zaključiti da su, uprkos opštim trendovima kretanja cena akcija, akcije *Apple*-a skočile u određenom trenutku zato što akcije firmi koje imaju kapitalizaciju veću od 103.17 milijardi dolara i prethodnu cenu akcija manju od \$146.23 rastu 165. dana u godini
 - ▶ Model verovatno ne može da zbog ovog neuobičajenog primera menja vrednosti težinskih parametara za odlike/kombinacije odlika koje realno jesu prediktivne, jer će time povećati grešku na svim ostalim podacima
 - ▶ Ta veoma specifična kombinacija odlika najverovatnije nema uticaja ni na jedan drugi podatak u skupu za obučavanje, a omogućava modelu da tačno pogodi cenu akcija *Apple*-a u posmatranom trenutku
 - ▶ Stoga će model toj kombinaciji odlika dati veliku težinu

Nastavite sekvencu brojeva: 1, 3, 5, 7, ?

Primer javljanja velikih magnituda težinskih parametara kod preterano prilagođenih modela

Nastavite sekvencu brojeva:
1, 3, 5, 7, ?

Tačan odgovor:
1, 3, 5, 7, 217341

Primer javljanja velikih magnituda težinskih parametara kod preterano prilagođenih modela

$$f(x) := \frac{18111}{2} x^4 - 90555 x^3 + \frac{633885}{2} x^2 - 452773 x + 217331$$

$$f(1) = 1$$

$$f(2) = 3$$

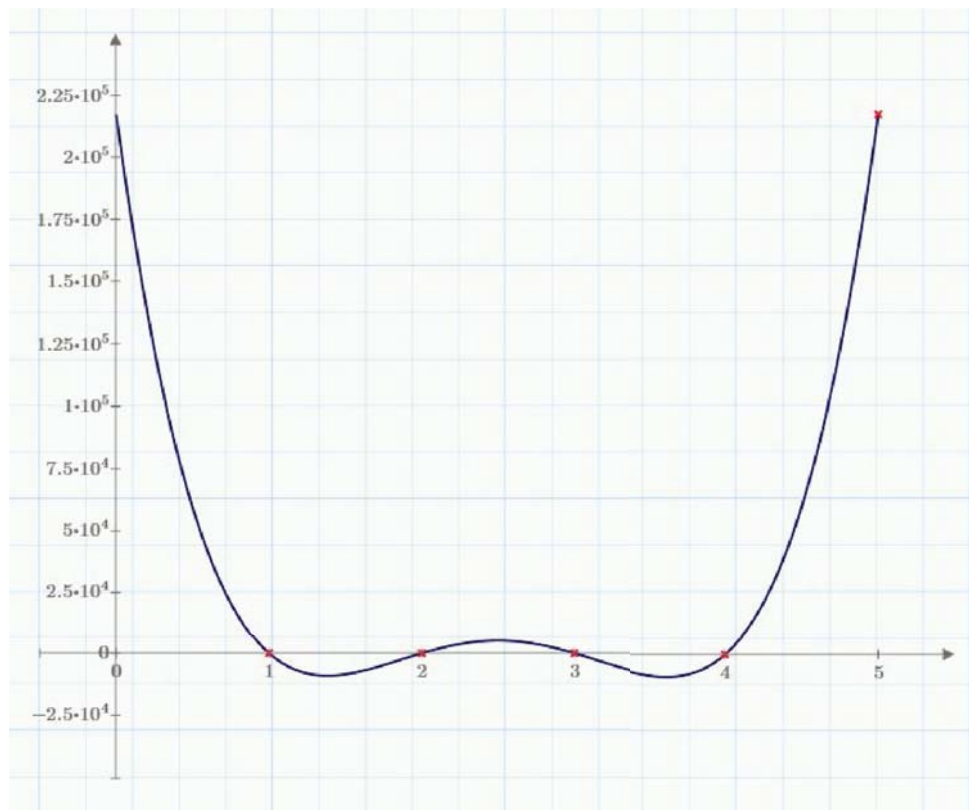
$$f(3) = 5$$

$$f(4) = 7$$

$$f(5) = 217341$$

Primer javljanja velikih magnituda težinskih parametara kod preterano prilagođenih modela

Slike preuzete sa: <http://blogs.ptc.com/2015/01/22/i-bet-you-cant-guess-the-next-value-in-the-sequence/>



Primer javljanja velikih magnituda težinskih parametara kod preterano prilagođenih modela

Slika preuzeta sa: <http://blogs.ptc.com/2015/01/22/i-bet-you-cant-guess-the-next-value-in-the-sequence/>

Regularizacija

- ▶ Kod preterano prilagođenih modela funkcija izlaza je veoma vijugava (i samim tim kompleksna) iz nastojanja da se greška na skupu za obučavanje smanji koliko god to fleksibilnost modela dozvoljava
 - ▶ To je posledica postojanja šuma u podacima za obučavanje i/ili nedovoljno podataka
- ▶ Regularizacija je dobila ime po tome što joj je cilj da funkciju izlaza učini regularnijom tj. pravilnijom, čime se sprečava preterana prilagođenost modela
- ▶ Model se regularizuje tako što se u funkciju greške ubacuje regularizacioni izraz $\lambda R(w)$, čime se forsiraju manje magnitude w
 - ▶ λ - hiperparametar modela koji određuje jačinu regularizacije
 - ▶ $R(w)$ - regularizaciona funkcija

Regularizacija

- ▶ Regularizacija takođe smanjuje nelinearnost modela (ako je ima)
- ▶ Vrednost hiperparametra λ se obično određuje eksperimentalno, korišćenjem unakrsne validacije
 - ▶ Ukoliko se unakrsna validacija koristi i za evaluaciju modela, neophodno je primeniti ugnežđenu unakrsnu validaciju za izbor hiperparametra λ
 - ▶ Previše mala vrednost λ - regularizacija neće imati (dovoljnog) efekta
 - ▶ Previše velika vrednost λ
 - ▶ Dovodi do nedovoljne prilagođenosti modela podacima (engl. *underfitting*)
 - ▶ Gradijentni spust neće uspeti da konvergira
- ▶ Pošto vrednosti w zavise i od vrednosti odlika, obično se pre regularizacije sprovodi skaliranje vrednosti odlika

Regularizacija

- ▶ Regularizaciona funkcija $R(w)$ je p -norma vektora parametara w (pri čemu se samostalni član w_0 obično izuzima):

$$\|w\|_p = \sqrt[p]{\sum_{j=1}^n |w_j|^p}$$

- ▶ U čestoj upotrebi su

- ▶ L_2 / Tihonovljeva regularizacija:

$$R(w) = \|w\|_2^2 = \sum_{j=1}^n |w_j|^2 = \sum_{j=1}^n w_j^2$$

- ▶ L_1 regularizacija:

$$R(w) = \|w\|_1 = \sum_{j=1}^n |w_j|$$

Grebena regresija

- ▶ Grebena (engl. *ridge*) regresija je termin koji se koristi za linearnu regresiju sa L_2 regularizacijom
- ▶ Funkcija greške kod grebene regresije je:

$$J(w) = \frac{1}{2m} \left(\sum_{i=1}^m \left(w_0 + w_1 x_1^{(i)} + \dots + w_n x_n^{(i)} - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n w_j^2 \right)$$

- ▶ Parcijalni izvodi kod grebene regresije imaju oblik:

$$\frac{\partial J(w)}{\partial w_r} = \frac{1}{m} \left(\sum_{i=1}^m \left(h(x^{(i)}) - y^{(i)} \right) x_r^{(i)} + \lambda w_r \right)$$

Grebenasta regresija

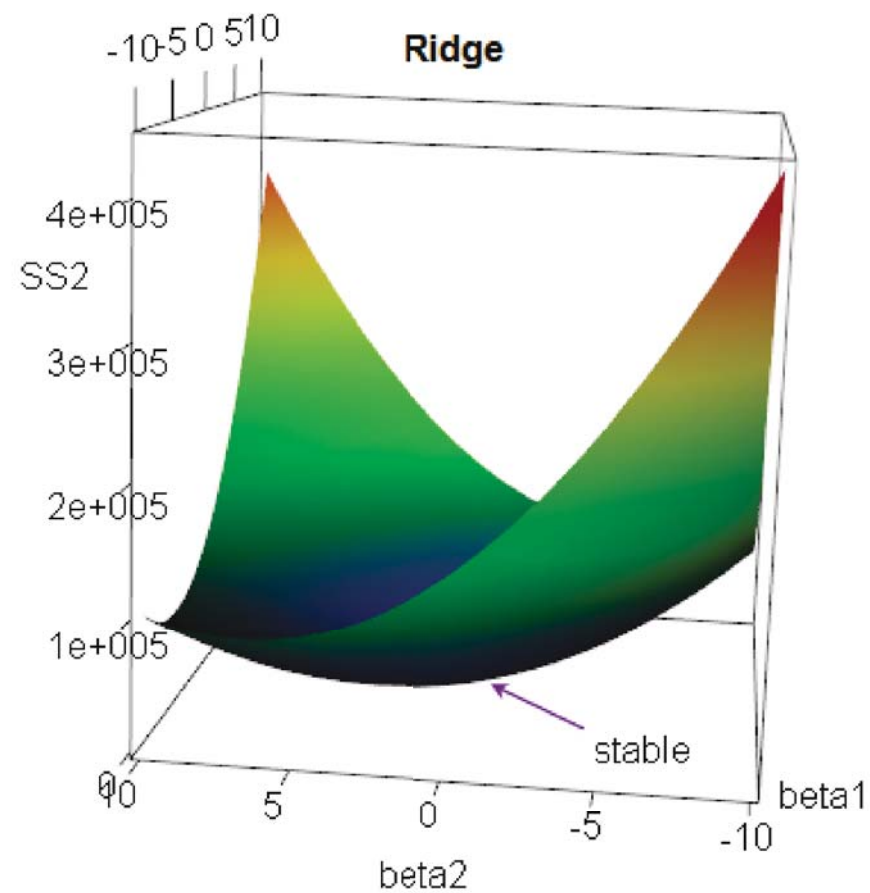
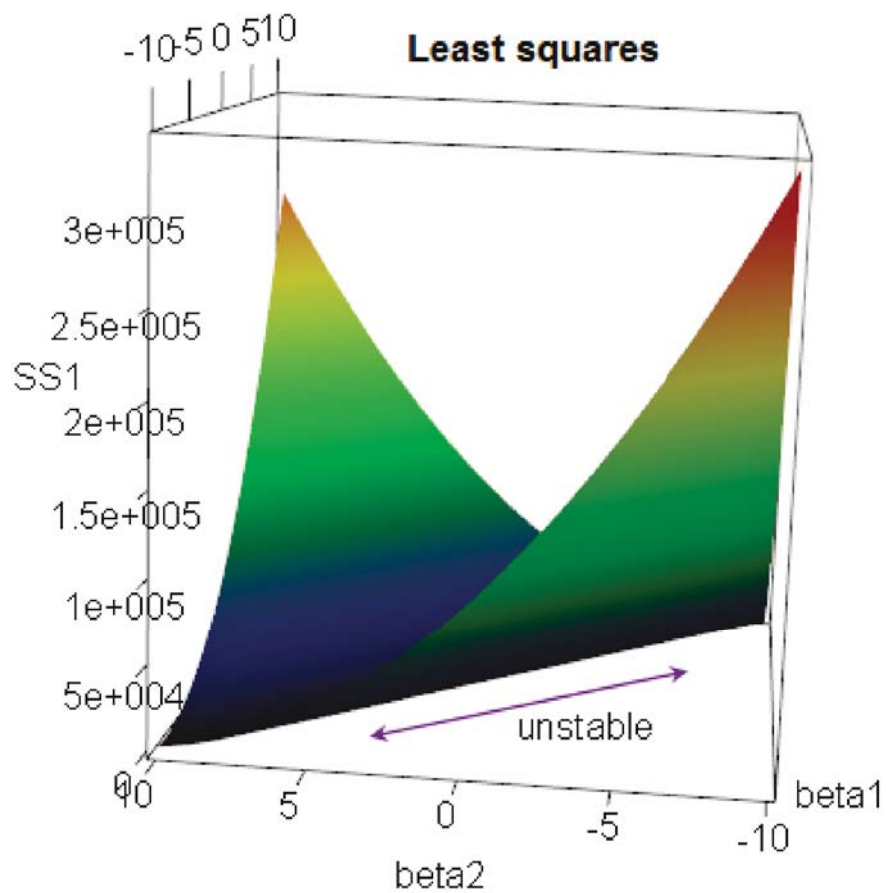
- ▶ Parametri w se u grebenastoj regresiji ažuriraju na sledeći način:

$$\begin{aligned}w_r &:= w_r - \alpha \frac{1}{m} \left(\sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_r^{(i)} + \lambda w_r \right) \\ &= w_r \left(1 - \alpha \frac{\lambda}{m} \right) - \alpha \frac{1}{m} \left(\sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_r^{(i)} \right)\end{aligned}$$

- ▶ Pošto regularizacija obično ne obuhvata w_0 , pravilo ažuriranja za w_0 ostaje isto kao u neregularizovanoj regresiji

Grebenarna regresija

- ▶ Ako postoji visoka korelisanost između odlika, neregularizovana linearna regresija nema jedinstven minimum, već „greben“ minimalnih vrednosti
 - ▶ To čini model nestabilnim - male promene u ulaznim podacima dovode do velikih promena u vrednostima težinskih parametara
 - ▶ Ova nestabilnost otežava interpretaciju modela
- ▶ L_2 regularizacija zamenjuje greben jedinstvenim minimumom



Ilustracija efekta uvođenja L_2 regularizacije

Slika preuzeta sa: <http://stats.stackexchange.com/questions/151304/why-is-ridge-regression-called-ridge-why-is-it-needed-and-what-happens-when>

LASSO regresija

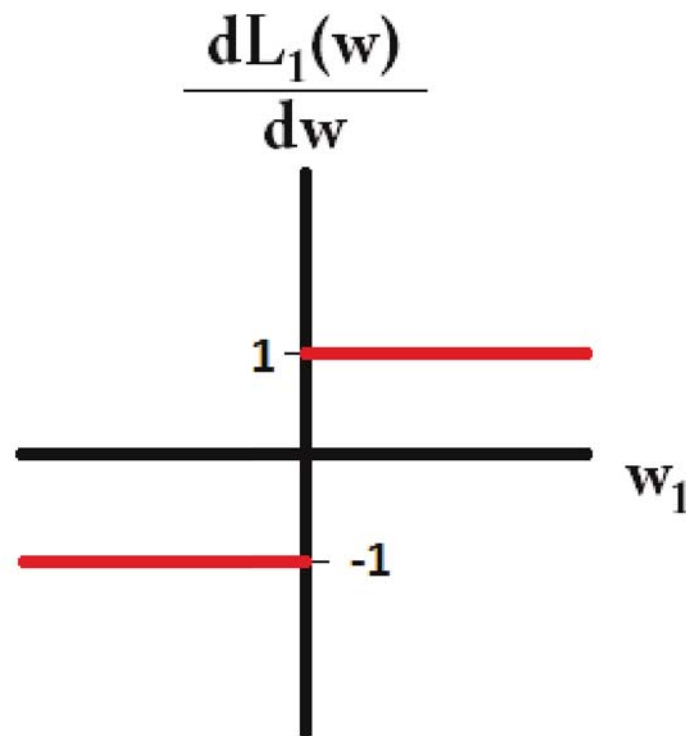
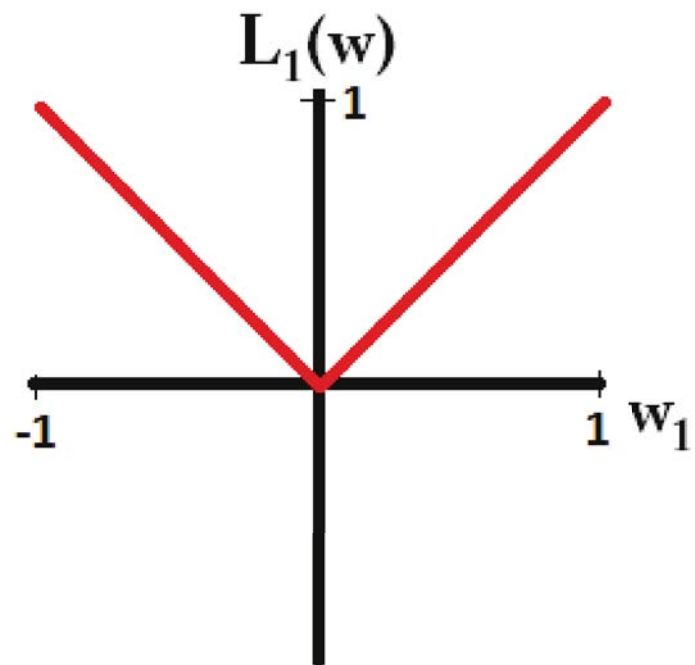
- ▶ LASSO (engl. *Least Absolute Shrinkage and Selection Operator*) regresija je termin koji se koristi za linearnu regresiju sa L_1 regularizacijom
- ▶ Funkcija greške kod LASSO regresije je:

$$J(w) = \frac{1}{2m} \left(\sum_{i=1}^m \left(w_0 + w_1 x_1^{(i)} + \dots + w_n x_n^{(i)} - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n |w_j| \right)$$

- ▶ L_1 regularizator nije diferencijabilan zbog apsolutne vrednosti
- ▶ Za ažuriranje vrednosti w koriste se napredniji mehanizmi optimizacije, kao što su subgradijentne metode
 - ▶ Koordinatni spust
 - ▶ Proksimalne metode

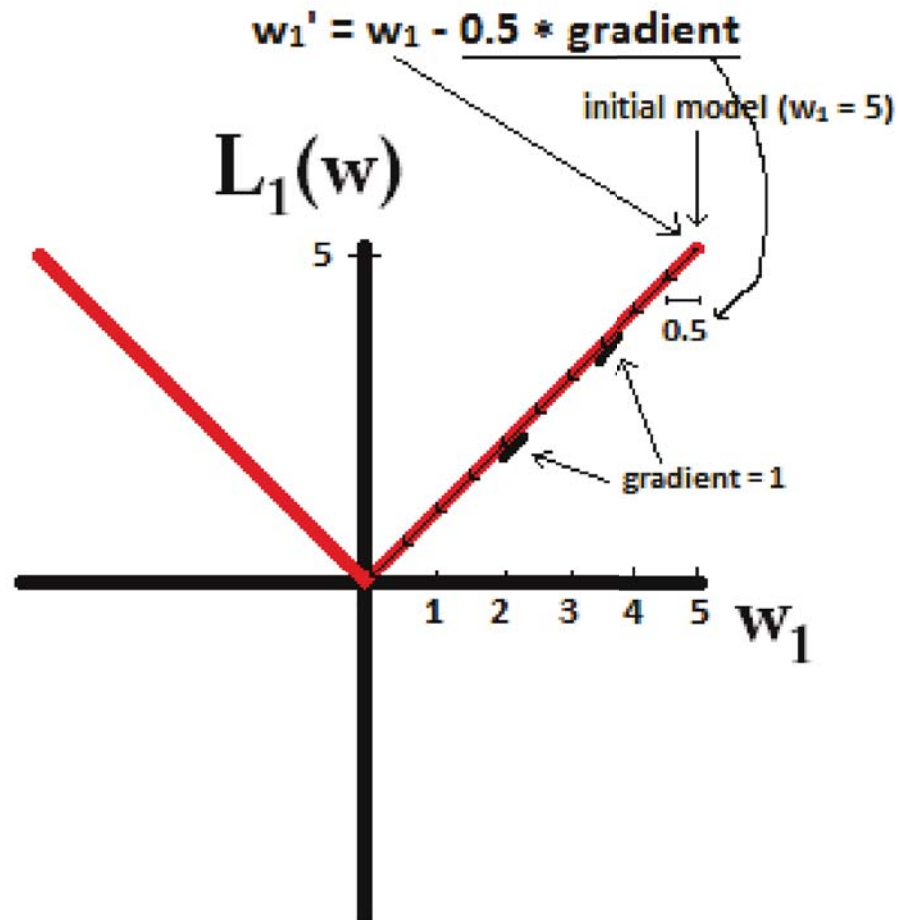
LASSO regresija

- ▶ L_1 regularizacija uz dovoljno veliku vrednost λ forsira određene težinske vrednosti w da postanu jednake nuli
 - ▶ Time se stvaraju proređeni modeli (engl. *sparse models*) zato što se odlike čije su težine jednake nuli efektivno ignorišu
 - ▶ Kako se vrednost λ povećava, raste i broj težinskih vrednosti postavljenih na nulu
- ▶ Ovo je korisno kada je potrebno istovremeno sprovesti i regularizaciju i selekciju odlika
 - ▶ Može biti dobro kada u modelu postoji ogroman broj odlika



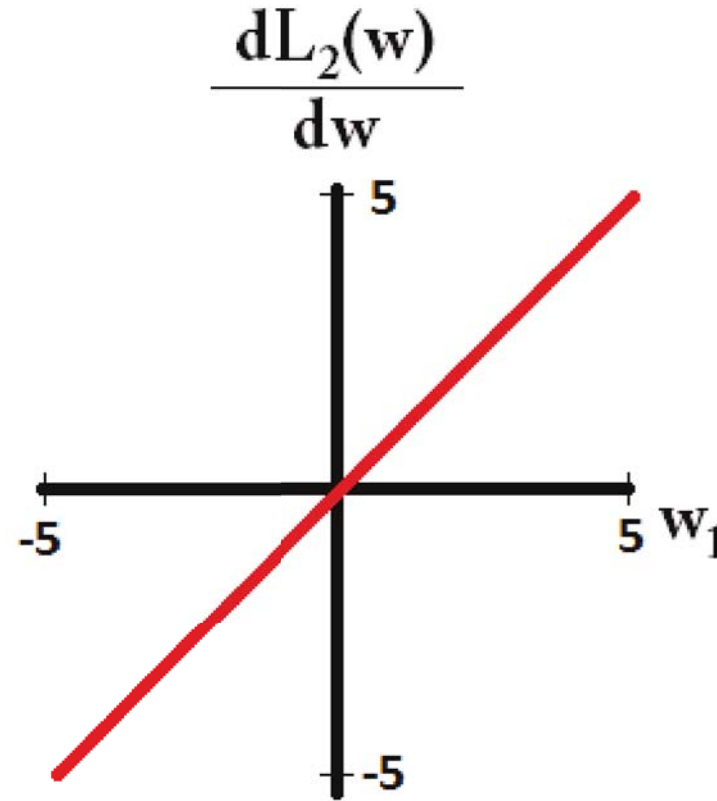
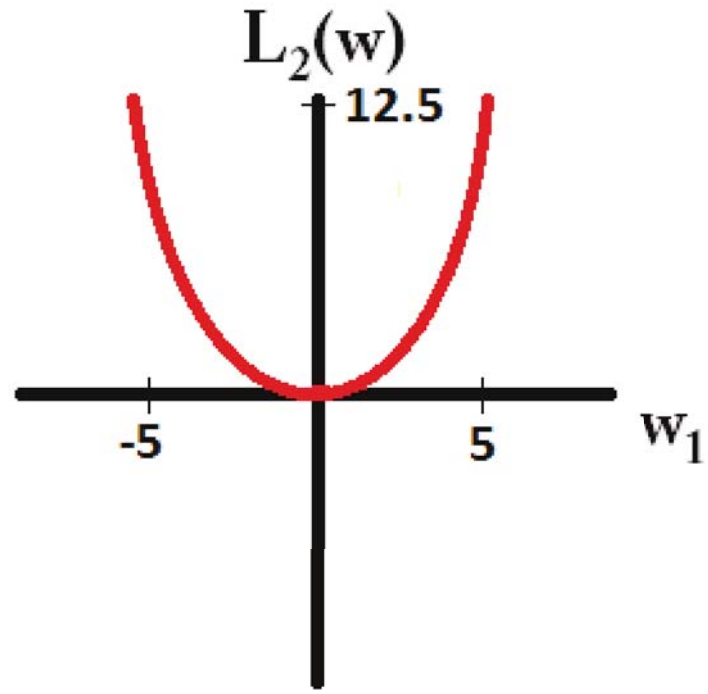
Oblik funkcije L_1 regularizacije i njenog izvoda

Slika preuzeta sa: <http://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models/>



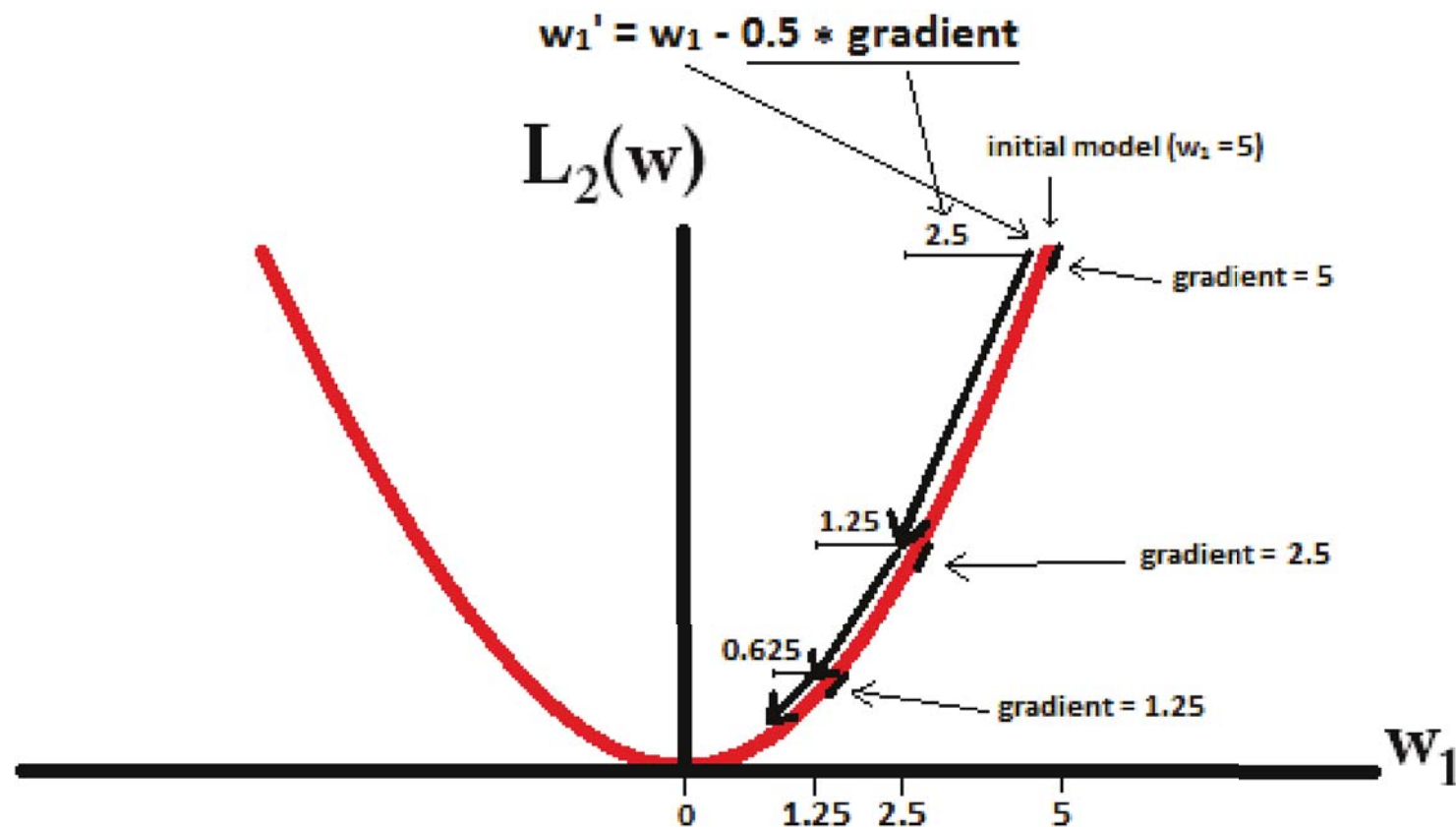
Promena vrednosti w pri gradijentnom spustu kod L_1 regularizacije

Slika preuzeta sa: <http://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models/>



Oblik funkcije L_2 regularizacije i njenog izvoda

Slika preuzeta sa: <http://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models/>



Promena vrednosti w pri gradijentnom spustu kod L_2 regularizacije

Slika preuzeta sa: <http://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models/>

L_1 vs L_2 regularizacija

- ▶ Minimizacija vrednosti w
 - ▶ L_1 regularizacija dovodi do toga da neki parametri postanu jednaki nuli
 - ▶ Vršiti selekciju odlika
 - ▶ Povećava interpretabilnost modela
 - ▶ L_2 regularizacija u principu nema ovu osobinu, te stoga ne može da se koristi za selekciju odlika
 - ▶ Teorijski je moguće da neki parametar postane jednak nuli za određenu vrednost brzine učenja α ili kao posledica smanjenja funkcije greške
- ▶ Broj odlika koje utiču na izlaz
 - ▶ Mali broj odlika koje imaju veliki uticaj - L_1 regularizacija
 - ▶ Veliki broj odlika koje imaju međusobno uravnotežen uticaj - L_2 regularizacija

L_1 vs L_2 regularizacija

- ▶ L_1 regularizacija je manje stabilna od L_2 regularizacije
 - ▶ Male promene u podacima za obučavanje mogu dovesti do velikih promena modela kada se koristi L_1 regularizacija
 - ▶ L_2 regularizovani modeli su otporniji na ove efekte
- ▶ L_2 regularizacija može da bude brža za izračunavanje
- ▶ Izbor napraviti pomoću unakrsne validacije na konkretnom skupu podataka

Prednosti i mane linearne regresije

▶ Prednosti

- ▶ Jednostavnost
- ▶ Interpretabilnost - veća težina dodeljena odlici signalizira njenu veću važnost (ako su vrednosti odlika skalirane)
 - ▶ Treba biti oprezan - interpretabilnost je znatno otežana ako postoji međusobna zavisnost između odlika
- ▶ Često ostvaruje dosta dobru predikciju budućih podataka

▶ Mane

- ▶ Realne regresione funkcije skoro nikada nisu linearne
- ▶ Osetljivost na *outlier*-e