

# Inteligentni sistemi

Mašinsko učenje

Linearna regresija

# Podela

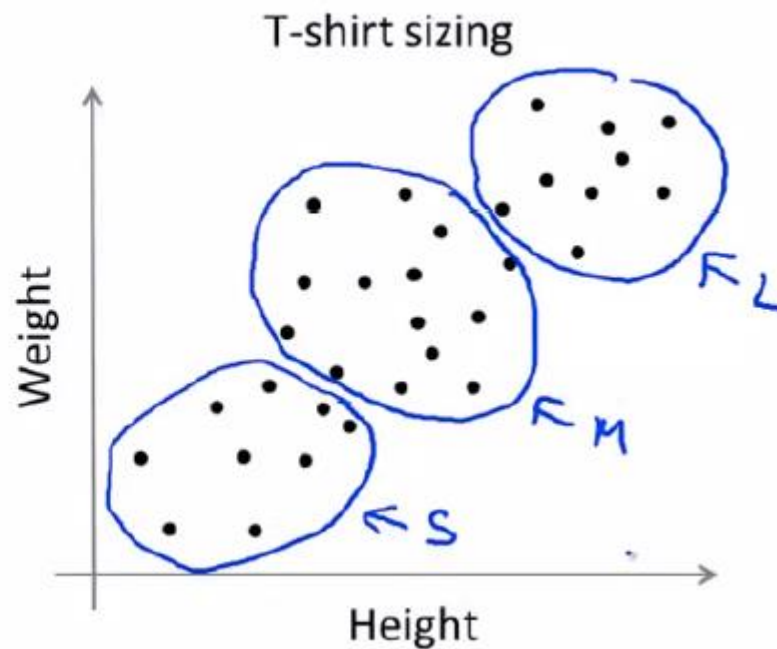
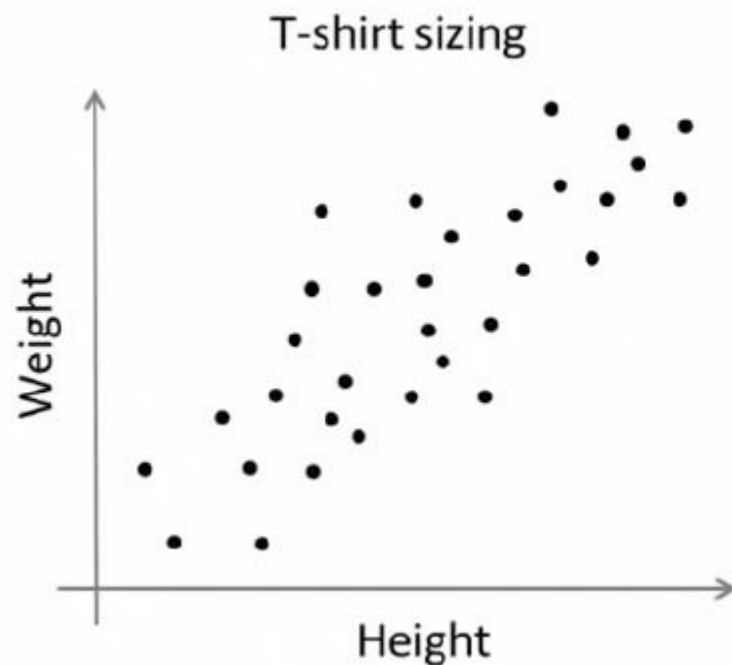
- Postoji nekoliko različitih tipova algoritama u mašinskom učenju
- Supervised learning (nadgledano učenje)
- Unsupervised learning
- Polunadgledano učenje
- Učenje sa podrškom (reinforcement learning)

# Nenadgledano učenje

- Nemamo informaciju o željenoj izlaznoj vrednosti
- Sistem dobija samo skup ulaznih podataka
- Zadatak – otkriti skrivene strukture/zakovitosti
  - Grupisanje ( klasterovanje)
  - Smanjenje dimenzionalnosti
  - Asocijativna pravila
- Algoritmi:
  - K-means
  - PCA (Principal Component Analysis)
  - Učenje na bazi asocijativnih pravil

# Nenadgledano učenje primer - grupisanje

- određivanje konfekcijskih veličina na osnovu visine i težine ljudi



# Nenadgledano učenje

- Neobebežen podatak iz skupa podataka za treniranje sastoji se samo iz ulaznih podataka, tipično predstavljenih nizom realnih brojeva.
- Zadatak algoritama je da otkriju zakonitosti u podacima
- Nakon što je zakonitost u podacima pronađena - da za novi skup podataka (takođe neobebežen) izračunaju neki zaključak koji zavisi od samog algoritma

# Polunadgledano učenje

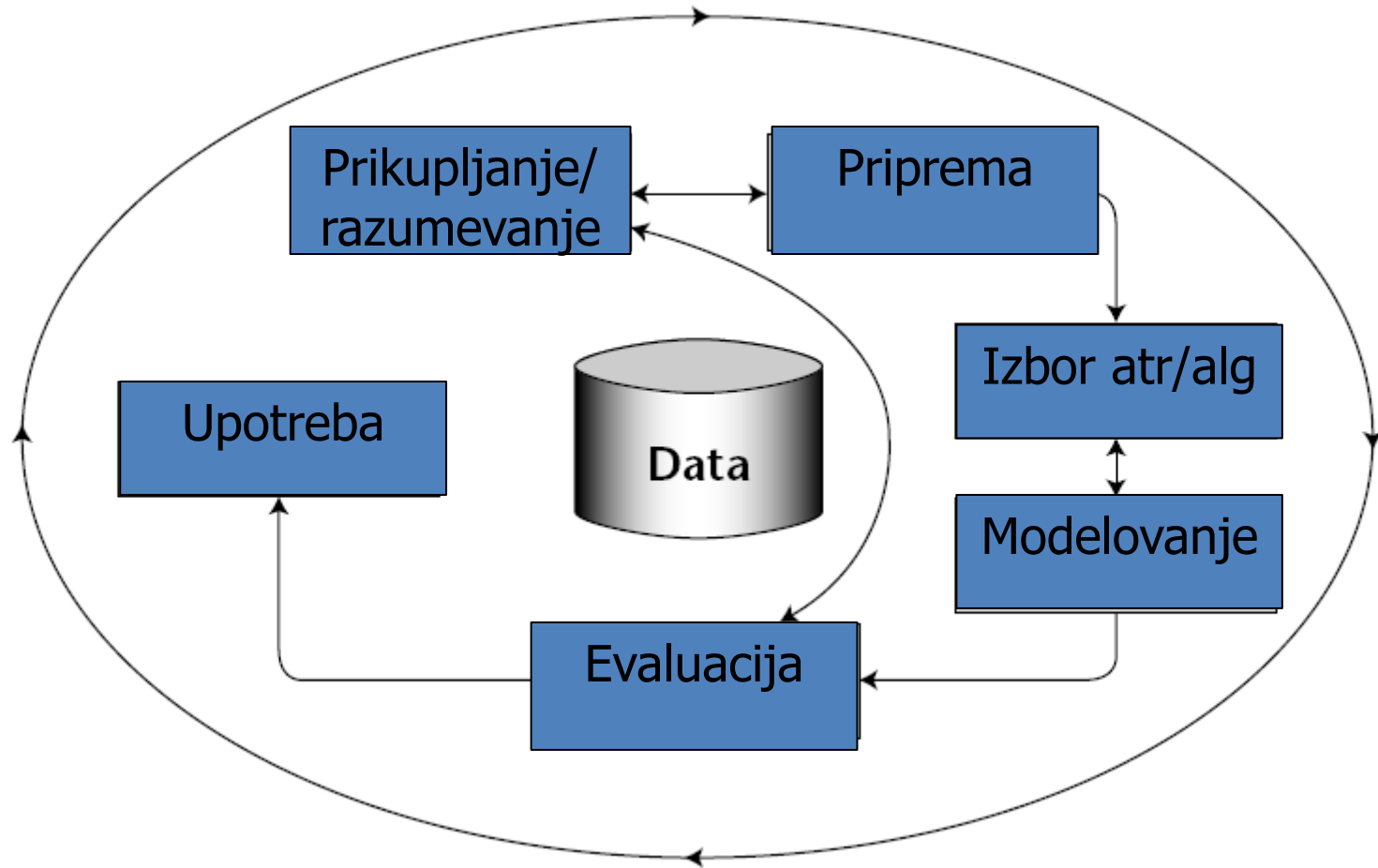
- Kombinacija nadgledanog i nenadgledanog učenja
- Manji deo je obeležen, veći deo nije
- Anotacija i označavanje je teška i skupa, a moguće su i greške
- Prepoznavanje ručno ispisanih cifara

4 4 4 4  
7 7 2 2

# Učenje sa podrškom

- Obuka softverskih agenata
- Moguće akcije i pozitivne i negativne nagrade
- Agent koristi signal za korekciju svog ponašanja
- AlphaGo

# Proces mašinskog učenja





# Podaci

- Potrebni su za trening, validaciju i testiranje modela
- Tipična podela na 60% za trening, 20% za validaciju i 20% za testiranje
- Izbor uzoraka treba uraditi na slučajan način (random selection)
- U podacima za obučavanje, dešava da se nađe određeni broj podataka koji značajno odstupaju od zakonitosti koja važi za ostale podatke - outliers (odudarajući)
- Odstranjuju se ili prilagoditi algoritam

# Atributi (*features*)

- Model treba da “verno” opisuje pojave/entitete
- Zato prepoznajemo osobine i odnose u datom domenu i predstavljamo ih atributima
- Izazov je odabrati prave attribute

# Atributi - primeri

- Za kreditne zahteve: vrednost imovine podnosioca, primanja, zaposlenje, bračno stanje, itd.
- Za identifikaciju nepoželjne elektronske pošte (spam): naslov, prisustvo tipičnih reči (buy, visit,...), dužina email-a, broj primalaca, itd.

# Testiranje

- Procena uspešnosti modela
- Koriste se podaci kojima model nije imao pristup u fazi učenja (20-30% ukupnih podataka)
- Uspešnost se utvrđuje različitim metrikama: tačnost, preciznost, odziv, ...

# TRAIN/VALIDATE/TEST

- Pored treniranja i testiranja modela, najčešće se radi i validacija modela kako bi se:
  - a) izabrao najbolji model između više kandidata
  - b) odredila optimalna konfiguracija parametara modela
  - c) izbegli problemi *over/under-fitting-a*
- Podaci za validaciju koriste se za poređenje performansi
  - različitih modela (a);
  - izabranog modela sa različitim vrednostima parametara (b)

# Analiza greške

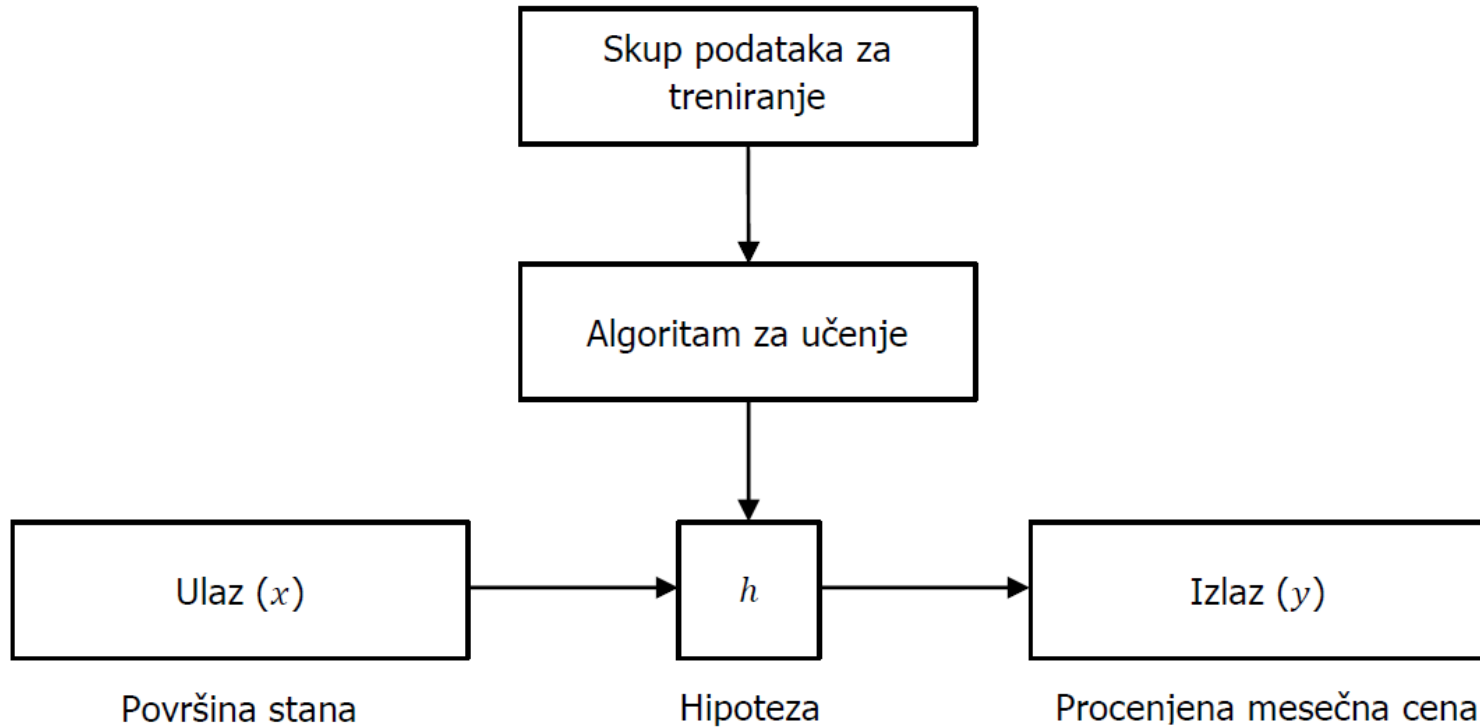
- “ručno” pregledanje primera na kojima je model pogrešio
- Pomaže da se stekne osećaj zbog čega model greši i šta bi se moglo uraditi da se greške otklone; npr:
  - identifikovati suvišne attribute
  - identifikovati attribute koji nedostaju
  - drugačije podesiti parametre modela

# Linear regression

- Uvodi se notacija za određena svojstva skupa podataka za treniranje.
- Definisani su sledeći simboli:
  - $m$  – broj primera u skupu podataka za treniranje
  - $x$  – ulazni podatak
  - $y$  – izlazni podatak

	Površina stana m <sup>2</sup> ( $x$ )	Mesečna cena € ( $y$ )
$i = 1$	35	215
$i = 2$	45	250
	35	250
	45	280
	40	300
$i = m$	...	...

# Linear regression



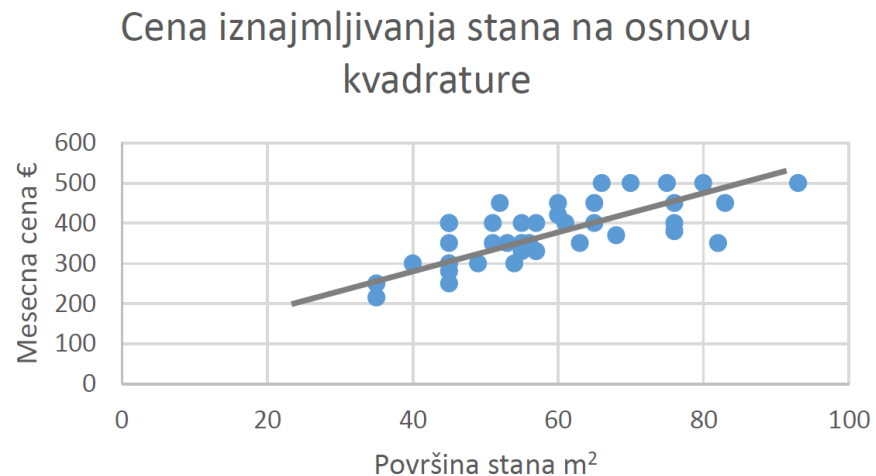


# Linear regression

- $h$  je simbol koji predstavlja hipotezu definisanu funkcijom koja vrši mapiranje iz ulaznog podatka  $x$  u izlazni  $y$ .
- Pri dizajniranju algoritma za učenje bitno je odlučiti kako definisati funkciju koja predstavlja hipotezu.
- Radi jednostavnosti, biće usvojeno da je hipoteza definisana na sledeći način:
- $h_{\theta}(x) = \theta_0 + \theta_1 x$
- Simbol  $\theta$  predstavlja skup parametara ove funkcije.

# Linear regression

- Ovako definisana hipoteza predviđa da je rešenje predstavljano linearnom funkcijom od  $x$ .
- Primer sa slike ne mora uvek da bude rešenje
- U ovom slučaju radi se o modelu sa jednom promenljivom, ali hipoteza može da bude dosta složenija



# Linear regression

- Odabrana funkcija za hipotezu je  $h_{\theta}(x)$  i pitanje koje se postavlja jeste kako odabrati parametre  $\theta_0$  i  $\theta_1$ .
- Za različite parametre dobijaju se različite funkcije
- Parametri koji najviše odgovaraju podacima su oni za koje je  $h(x)$  najbliže  $y$  za svaki primer  $(x, y)$

$$\underset{\theta_0, \theta_1}{\text{minimum}} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

# Linear regression

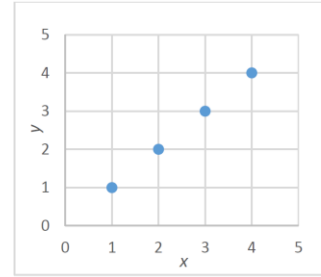
- Problem se dakle predstavlja kao potraga za  $\theta_0, \theta_1$  takvim da prosečna kvadratna greška između procenjene vrednosti  $h_{\theta}(x^{(i)})$  i prave izlazne vrednosti  $y^{(i)}$  bude minimalna.
- Po konvenciji, definiše se optimizaciona funkcija kao:

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

- Ovakav oblik optimizacione funkcije, menja prethodni cilj u:

$$\underset{\theta_0, \theta_1}{\text{minimum}} J(\theta)$$

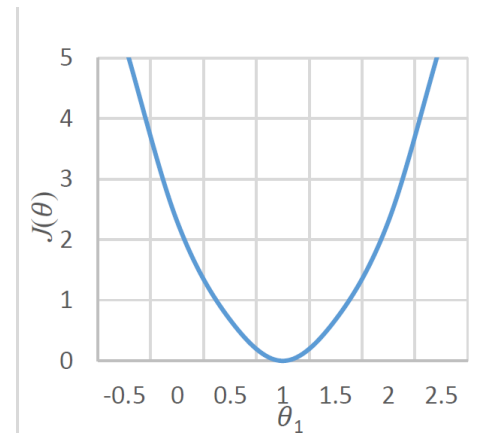
# Linear regression



- Optimizaciona funkcija se može definisati i na drugačije načine, ali se prosečna kvadratna greška pokazala kao jedna od najboljih i najčešće korišćenih funkcija kada se radi o regresionim problemima
- Kvadrat će velike razlike koje odgovaraju odudarajućim podacima učiniti još većim
- Proces optimizacije neproporcionalno fokusirati na smanjivanje grešaka modela na tim podacima

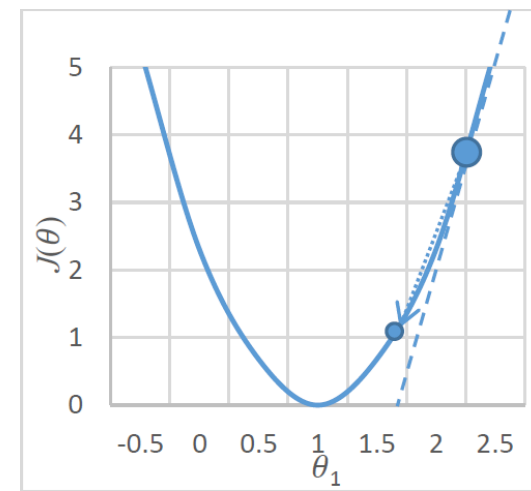
# Linear regression

- Radi jednostavnosti recimo da je parametar  $\theta_0$  pronađen i fiksiran na 0, kako bi još jednostavnije prikazali optimizacionu funkciju, za parametar  $\theta_1$ .
- Za različite vrednosti  $\theta_1$  dobijaju se različite vrednosti  $J(\theta)$ .
- Treba pronaći  $\theta_1$  za koje  $J(\theta)$  ima najmanju vrednost, odnosno treba pronaći parametar  $\theta_1$  za koji je  $h(x)$  najbliže  $y$  za svaki primer  $(x, y)$ .
- U datom primeru za  $\theta_0 = 0$ ,  $\theta_1 = 1$ ,  $J(\theta)$  ima najmanju vrednost što znači da hipoteza  $h(x) = x$  najbolje odgovara datom skupu podataka.



# Linear regression

- U prethodnom primeru, definisana je funkcija  $J(\theta)$  i pokazano je da minimizacija te funkcije daje potreban skup parametara  $\theta$  tj. parametre  $\theta_0$  i  $\theta_1$ .
- Moguće je koristiti i odgovarajući gradijent koji se koristi u raznim algoritmima mašinskog učenja.
- Koristi se za optimizaciju funkcije, u ovom slučaju  $J(\theta)$  - traženje parametara dok se ne nađu oni za koje data funkcija ima najmanju vrednost
  - Započeti sa nekim skupom parametara  $\theta$
  - Menjati parametre  $\theta$ , u cilju smanjivanja  $J(\theta)$ , dok se ne nađe minimum funkcije
- Neka je početna vrednost parametra  $\theta_1 = 2.25$ , i radi jednostavnosti prikaza neka je  $\theta_0 = 0$ .



# Linear regression

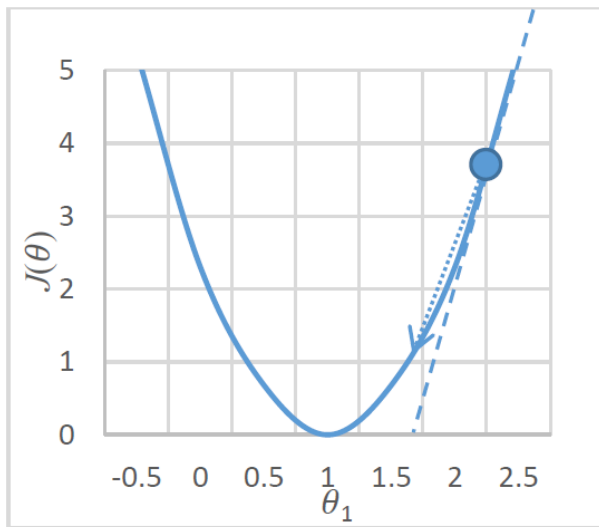
- Potrebno je odrediti nagib funkcije za određenu tačku koji se može izračunati parcijalnim izvodom funkcije  $J(\theta)$ , po nekom parametru  $\theta_j$ , gde je  $j$  definisano kao indeks parametra (za dati primer  $j = 0$  i  $j = 1$ ).
- Veličina koraka se može kontrolisati novim parametrom  $\alpha$ , koji se naziva stopa učenja (Learning rate)
- Ovaj parametar se koristi u većini algoritama mašinskog učenja, kao pozitivan realan broj veći od nula.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$



# Linear regression

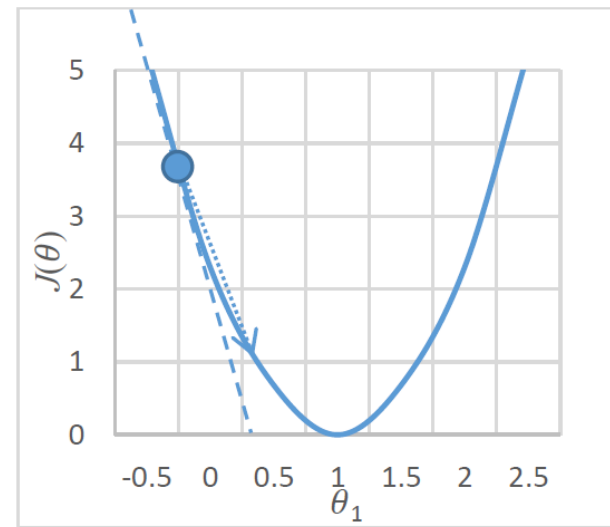
- Promena parametra u zavisnosti od nagiba optimizacione funkcije



$$\frac{\partial}{\partial \theta_1} J(\theta) > 0$$

$$\theta_1 := \theta_1 - \alpha * \text{pozitivan broj}$$

$\theta_1$  se smanjuje



$$\frac{\partial}{\partial \theta_1} J(\theta) < 0$$

$$\theta_1 := \theta_1 - \alpha * \text{negativan broj}$$

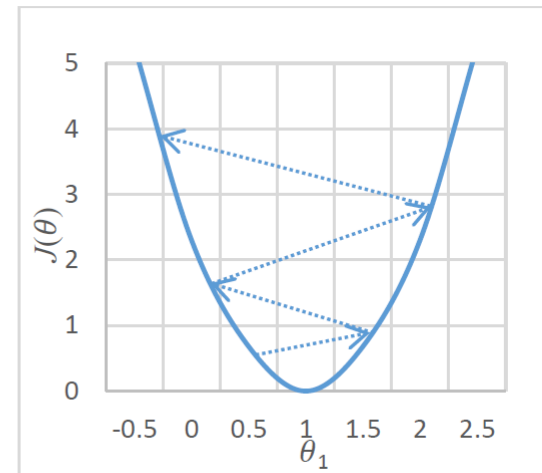
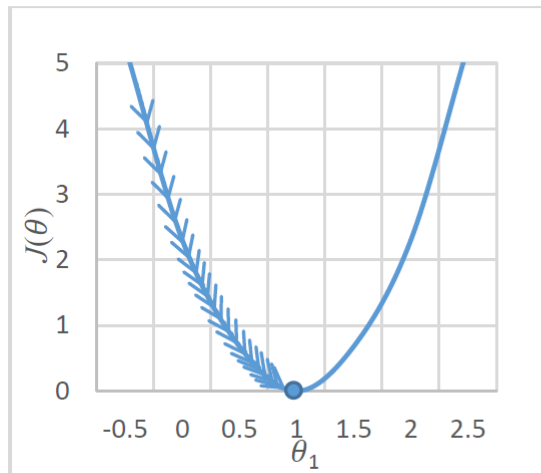
$\theta_1$  se povećava

# Linear regression

- Približavanje minimumu smanjuje faktor, pa će gradijent automatski praviti manje korake - stopu učenja nije potrebno vremenom smanjivati kako bi se ostvarila konvergencija ka minimumu funkcije.
- Međutim od stope učenja veoma zavisi ponašanje algoritma kao i uopšte pronalazak svih parametara  $\theta$ , za koje  $J(\theta)$  ima minimalnu vrednost.
- Ukoliko je stopa učenja premala, algoritam će biti veoma spor
- Ukoliko je stopa učenja prevelika moguće je da se minimum funkcije nikada neće pronaći, što znači da neće doći do konvergencije, a može se desiti da dođe i do divergencije.

# Linear regression

- Primer premale i prevelike stope učenja



# Linear regression

*Ponavljati do konvergencije {*

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \text{ (istovremeno promeniti } \theta_j \text{ za } j = 0 \text{ i } j = 1)$$

*}*

Istovremena promena parametara je veoma bitna jer jedino tako će  $J(\theta)$ , vraćati tačne vrednosti iz prethodne iteracije petlje. Ispravna implementacija za dati primer je data u nastavku.

$$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta)$$

$$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta)$$

$$\theta_0 := temp0$$

$$\theta_1 := temp1$$

# Linear regression

Cilj je naravno pronalazak skupa parametara za koje optimizaciona funkcija  $J(\theta)$  ima najmanju vrednost. Gradient descent ne koristi funkciju  $J(\theta)$  u njenom originalnom obliku već kroz parcijalni izvod, pa je potrebno izračunati šta taj faktor predstavlja.

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

Posmatrajući dobijeno opšte rešenje, mogu se izračunati parcijalni izvodi u zavisnosti o indeksa parametra  $j$ .

$$j = 0: \frac{\partial}{\partial \theta_0} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$

$$j = 1: \frac{\partial}{\partial \theta_1} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)}$$

# Linear regression

*Ponavljaj do konvergencije {*

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)}$$

*}*

- Prikazana implementacija algoritma u svakoj iteraciji koristi sve primere iz skupa podataka za treniranje.
- Ovakav pristup se naziva Batch Gradient descent

# Mane

- Jedna od slabosti linearnih modela je nezavisnost dejstva atributa na ciljnu promenljivu
- Jedinična promena svakog atributa doprinosi promeni ciljne promenljive proporcionalno koeficijentu koji odgovara tom atributu, nezavisno od vrednosti drugih atributa.
- Neadekvatno u mnogim problemima
- U genetici - ponašanje nekih gena može zavisiti od varijacije nekog drugog gena.

# Proizvod atributa

- Uključivanje interakcija – proizvoda atributa, kao novih atributa u model:

$$f_w(x) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{1 \leq i < j \leq n} w_{ij} x_i x_j$$

- Vrednost jedne promenljive kontroliše ispoljavanje druge promenljive
- Interakcije značajno povećavaju izražajnost linearnih modela - kod praktičnih primena