Istraživanje podataka u bioinformatici

Istraživanje podataka - pregled

Matematički fakultet, Univerzitet u Beogradu

June 1, 2026

Agenda

1. Istraživanje podataka - pregled
2. Podaci
3. Klasifikacija
4. Klasterovanje

Istraživanje podataka - pregled

Istraživanje podataka - definicija

Hand

Proces sekundarne analize baza podataka sa ciljem pronalaženja neočekivanih odnosa koji su od interesa ili vrednosti za vlasnika baze podataka

Simoudis

Proces izdvajanja tačnih, ranije nepoznatih, razumljivih i primenljivih informacija iz velikih baza podataka i njihove upotrebe za donošenje ključnih poslovnih odluka

Istraživanje podataka - definicija

Fayyad et al.

KDD (eng. Knowledge discovery in databases) je proces korišćenja baze podataka, zajedno sa potrebnom selekcijom, preprocesiranjem, uzorkovanjem i transformacijama; zatim primene metoda (algoritama) istraživanja podataka radi izdvajanja obrazaca iz baze; kao i procene rezultata istraživanja podataka u cilju identifikovanja podskupa izdvojenih obrazaca koji se smatraju znanjem.

Karakteristike istraživanja podataka

- Algoritamska zasnovanost
- Svaki algoritam pokušava da ukalupi podatke u neki model
- Bira se model koji je najbliži karakteristikama podataka
- Podaci su u bazama podataka
- Potrebna su znanja iz oblasti:
  - baza podataka,
  - verovatnoće i statistike,
  - veštačke inteligencije,
  - programiranja.

Glavne tehnike

- klasterovanje
- klasifikacija
- regresija
- pravila pridruživanja
- analiza elemenata van granica

Koraci u procesu istraživanja podataka

- razumevanje problema
- pronalaženje i/ili prikupljanje podataka koji mogu da se koriste za rešavanje problema
- analiza podataka
- priprema podataka za pravljenje modela
- pravljenje modela primenom tehnika istraživanja podataka
- evaluacija dobijenog modela
- primena modela

Koraci u procesu istraživanja podataka

Podaci

Podaci

- Najčešći oblik podataka je višedimenzioni
- Podaci su sastavljeni od više atributa (promenljivih, polja, karakteristika)
- Atributi zajedno opisuju svaki objekat u skupu podataka
- Objekat se naziva i slog, instanca, entitet, primer

Primer višedimenzionih bioinformatičkih podataka

Za više pacijenata se meri ekspresija nekoliko gena, prisustvo mutacija i klinička klasa bolesti.

Uzorak	BRCA1	TP53	EGFR	Mutacija TP53	Starost	Tip tumora
S1	12.4	5.1	20.3	1	45	A
S2	8.7	7.9	15.2	0	52	B
S3	14.1	4.8	22.6	1	39	A
S4	6.3	9.5	11.4	0	61	B
S5	10.8	6.2	18.7	1	48	A

Napomena: Mutacija TP53 je kodirana kao 1 = prisutna, 0 = odsutna.

Tipovi atributa

- Prema vrednosti, atributi mogu biti:
  - Kvantitativni
    - primer: starost u godinama
  - Kategorijski
    - primer: boja

Tipovi atributa

Prema broju vrednosti, atributi mogu biti:

- Diskretni
  - konačan broj vrednosti
  - primeri: etnička pripadnost, ime, binarni atributi (npr. 1/0, da/ne, ...)
- Kontinuirani (neprekidni)
  - primeri: temperatura, težina

Tipovi atributa prema operacijama

- Diskretni
  - Imenski
    - dozvoljene operacije: $=$, $\neq$
    - primeri: naziv ulice, boja očiju, pol
  - Redni
    - dozvoljene operacije: $=$, $\neq$, $<$, $\leq$, $>$, $\geq$
    - primeri: ocena kvaliteta (loše, srednje, dobro, odlično), mesto na takmičenju (1,2,3)

Tipovi atributa prema operacijama

- Neprekidni
  - Intervalni
    - dozvoljene operacije: $=$, $\neq$, $<$, $\leq$, $>$, $\geq$, $+$, $-$
    - ne postoji apsolutna nula koja predstavlja odsustvo nečega
    - primeri: kalendarska godina, temperatura u stepenima Celzijusa
  - Razmerni
    - dozvoljene operacije: $=$, $\neq$, $<$, $\leq$, $>$, $\geq$, $+$, $-$, $*$, $/$
    - primeri: težina, dužina, cena

Asimetrični atributi - bitno je samo prisustvo ne-nula vrednosti

Uzorak	BRCA1 mutacija	TP53 mutacija	EGFR mutacija	HER2 amplifikacija
Pacijent 1	1	0	0	1
Pacijent 2	0	0	1	0
Pacijent 3	1	1	0	0
Pacijent 4	0	0	0	0
Pacijent 5	0	1	1	0

Table: Primer asimetričnih binarnih podataka u bioinformatici

Tipovi podataka

Tipovi podataka se mogu podeliti u dve osnovne grupe:

- Podaci bez zavisnosti
  - objekti i atributi se posmatraju nezavisno
  - primer:
    - tabela pacijenata sa atributima: starost, pol, težina
- Podaci sa zavisnostima
  - postoje veze između objekata ili atributa

Podaci sa zavisnostima – primeri

Promena ekspresije gena tokom infekcije

- podaci su prikupljani tokom vremena,
- vrednosti u jednom trenutku zavise od prethodnih trenutaka,
- rast virusnog opterećenja utiče na ekspresiju gena i temperaturu.

Vreme (h)	Ekspresija IL6	Ekspresija TNFα	Temperatura	Virusno opterećenje
0	5	3	36.8	120
6	18	14	37.5	450
12	35	28	38.2	920
24	42	31	39.0	1500
48	20	15	37.4	700
72	8	5	36.9	180

Table: Primer bioinformatičkih podataka sa vremenskom zavisnošću. IL6 i TNFα su citokini - mali signalni protein koji ćelije koriste za međusobnu komunikaciju, naročito u imunskom sistemu.

Podaci sa zavisnostima – primeri

- Vremenske serije
  - podaci prikupljeni tokom vremena
  - vrednosti zavise od prethodnih trenutaka
  - primer: promena ekspresije gena tokom infekcije
- Genske regulatorne mreže
  - geni međusobno regulišu aktivnost drugih gena
  - veze predstavljaju aktivaciju ili inhibiciju
  - primer: gen TP53 reguliše ćelijski ciklus i apoptozu (proces kojim organizam uklanja oštećene/stare ćelije)

Podaci sa zavisnostima – primeri

- Proteinske interakcije
  - proteini međusobno stupaju u fizičke ili funkcionalne interakcije
  - interakcije omogućavaju biološke procese u ćeliji
- Filogenetska stabla
  - predstavljaju evolutivne odnose između organizama ili sekvenci
  - sličnije sekvence imaju bližeg zajedničkog pretka
  - primer: evolutivni odnosi između različitih koronavirusa

Podaci sa zavisnostima – primeri

Biološke sekvence - nizovi nukleotida ili aminokiselina

Priprema podataka

- Primena i uspeh istraživanja podataka u velikoj meri zavise od tehnika pripreme podataka
- Skup atributa podataka ima veći uticaj na kvalitet rezultata nego odabir algoritma
- Priprema podataka zahteva:
  - ljudsku intuiciju,
  - stručna znanja iz domena
- Cilj je da podaci budu pripremljeni na pravi način za analizu i modelovanje

Priprema podataka

- Koraci u pripremi podataka:
  - Izdvajanje karakteristika od interesa
  - Čišćenje podataka
    - obrada nedostajućih vrednosti,
    - obrada pogrešnih vrednosti – uklanjanje ili procena vrednosti
  - Transformacija vrednosti atributa
  - Smanjenje podataka

Rad sa nedostajućim vrednostima

- Eliminacija instanci koje sadrže nedostajuće vrednosti
- Procena nedostajućih vrednosti (imputacija)
- Jednostavna procena:
  - srednja vrednost ili medijana za numeričke vrednosti
  - najčešća vrednost za kategorijske vrednosti

Rad sa nedostajućim vrednostima

- Hot-deck metoda
  - nedostajuća vrednost zamenjuje se vrednošću koju ima slična instanca
  - primer: najbliži sused
- Modelske metode
  - kNN
    - vrednost se procenjuje kao prosek K najbližih suseda
  - Regresija
    - nedostajuća vrednost procenjuje se pomoću regresionog modela

Rad sa netačnim i nedoslednim vrednostima

- Nekad je potrebno primeniti domensko znanje
- Potrebno je znati:
  - očekivane intervale ili moguće vrednosti atributa
  - pravila koja definišu odnose između različitih atributa
- Primer:
  - ako je visina osobe $185\,cm$, a težina $15\,kg$,
  - verovatno postoji greška u podacima
- Domensko znanje omogućava:
  - otkrivanje anomalija,
  - korekciju podataka,
  - pouzdaniju analizu

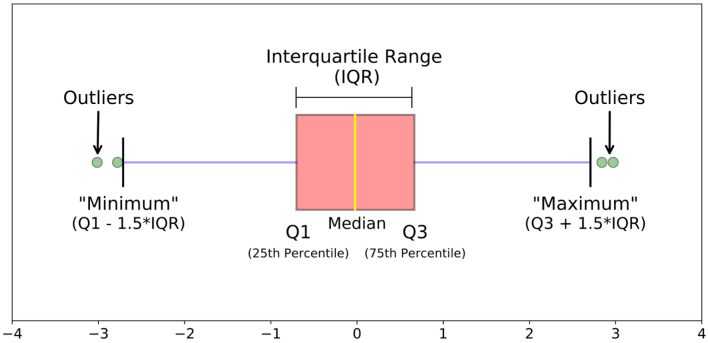
Rad sa nedostajućim ili netačnim vrednostima

- Greške nastale u procesu imputacije mogu uticati na rezultate algoritama istraživanja podataka
- Pogrešno procenjene vrednosti mogu:
  - smanjiti tačnost modela,
  - promeniti raspodelu podataka,
  - dovesti do pogrešnih zaključaka
- Mogu se birati algoritmi istraživanja podataka koji mogu da rade sa nedostajućim vrednostima
  - neki algoritmi mogu direktno obrađivati nepotpune podatke

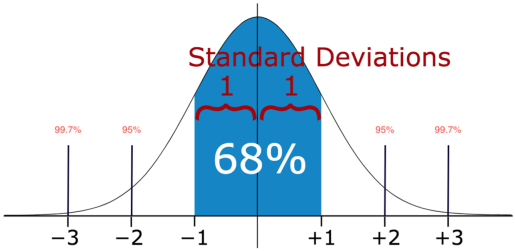
Rad sa netačnim i nedoslednim vrednostima

- Metodi zasnovani na podacima
  - koristi se statističko ponašanje podataka kako bi se otkrili odstupajući unosi
- Odstupajući unosi mogu biti:
  - rezultat grešaka pri prikupljanju podataka,
  - retke, ali validne pojave,
  - indikatori interesantnog ili važnog ponašanja sistema
- Odstupajući unos mora biti ručno proveren pre nego što se odbaci

Određivanje elemenata van granica korišćenjem percentila



Određivanje elemenata van granica metodom standardne devijacije



Transformacija vrednosti atributa

- Podaci često sadrže različite tipove atributa
- Primer:
  - demografski skup podataka može sadržati:
    - starost,
    - pol,
    - prihod,
    - mesto stanovanja
- Različiti tipovi atributa predstavljaju problem pri izboru algoritma istraživanja podataka
- Neki algoritmi:
  - rade samo sa numeričkim atributima,
  - zahtevaju podatke iste skale ili formata

Diskretizacija

- Diskretizacija
  - pretvaranje numeričkog atributa u kategorijski tip
- Koraci:
  1. podela opsega numeričkih vrednosti na N intervala
  2. intervalima se dodeljuju kategorijske vrednosti
- Primer:
  - atribut starost deli se na intervale:

$$[0, 10], [11, 20], [21, 30], \ldots, [100, 110]$$

  - intervalima se dodeljuju vrednosti: “1″, “2″, “3″, . . .
- Problem:
  - način određivanja intervala utiče na kvalitet transformacije

Intervali jednake širine

- Svaki interval $[a, b]$ bira se tako da je $b - a$ isto za sve intervale
- Postupak:
  - određuju se minimalna i maksimalna vrednost atributa $[min, max]$
  - opseg se deli na N intervala jednake dužine
- Primeri:
  - starost,
  - visina
- Nedostatak:
  - ne radi dobro za neujednačeno raspoređene podatke
  - primer:
    - atribut plata

Intervali jednake dubine

- Intervali se biraju tako da svaki interval sadrži približno jednak broj instanci
- Procedura:
  1. vrednosti atributa se urede
  2. određuju se granične tačke tako da svaki interval sadrži približno jednak broj elemenata
- Primeri:
  - plate,
  - cene

Intervali jednake logaritamske širine

- Svaki interval $[a, b]$ bira se tako da je $\log(b) - \log(a)$ isto za sve intervale
- Ovakav izbor intervala dovodi do geometrijskog povećavanja intervala
- Primer intervala:

$$[a, a \cdot \alpha],\ [a \cdot \alpha, a \cdot \alpha^2],\ \ldots$$

gde je:

$$\alpha > 1$$

- Ovakvi intervali su korisni kada atribut ima eksponencijalnu raspodelu
- Primer: veličina fajla

Binarizacija

- Binarizacija– pretvaranje kategorijskih podataka u binarni oblik
- Ako kategorijski atribut ima N različitih vrednosti pravi se N binarnih atributa
- Svaki binarni atribut odgovara jednoj mogućoj vrednosti kategorijskog atributa
- Za svaku instancu:
  - tačno jedan od N binarnih atributa dobija vrednost 1
  - svi ostali binarni atributi imaju vrednost 0

Primer binarizacije u bioinformatici

Originalni kategorijski atribut:

Uzorak	Tip virusa
U1	Alpha
U2	Delta
U3	Omicron
U4	Delta

Nakon binarizacije:

Uzorak	Alpha	Delta	Omicron
U1	1	0	0
U2	0	1	0
U3	0	0	1
U4	0	1	0

- Svaka moguća vrednost atributa „Tip virusa“ postaje poseban binarni atribut

Mere bliskosti

- Neki algoritmi istraživanja podataka kao ulaz koriste samo matricu bliskosti instanci
- Blizina (eng. proximity) označava:
  - sličnost,
  - različitost
- Sličnost
  - numerička mera koliko su dva objekta slična
  - što dva objekta više liče jedan na drugi, sličnost im je veća
  - često se meri vrednostima iz intervala: $[0, 1]$

Mere bliskosti

- Različitost
  - numerička mera koliko su dva objekta različita
  - što dva objekta više liče jedan na drugi, različitost im je manja
  - najmanja različitost je često 0, gornja granica može varirati u zavisnosti od mere
- Kao sinonim često se koristi termin rastojanje

Različitost između objekata podataka

- Rastojanje Minkovskog:

$$dist(p,q) = \left(\sum_{k=1}^{n} |p_k - q_k|^r \right)^{\frac{1}{r}}$$

- Gde je:
  - r – parametar,
  - n – broj dimenzija (atributa),
  - p_k i q_k – vrednosti k-tog atributa objekata p i q
- Specijalni slučajevi:
  - $r = 1$
    - Menhetn rastojanje
    - L_1 norma
  - $r = 2$
    - Euklidsko rastojanje

Skaliranje podataka

- Različite karakteristike mogu biti na različitim skalama, pa nisu direktno uporedive
- Primer: plata i starost
- Standardizacija

$$\frac{x - \mu}{\sigma}$$

  - μ – srednja vrednost
  - σ – standardna devijacija
  - većina standardizovanih vrednosti biće u opsegu $[-3, 3]$
  - važi pod pretpostavkom normalne raspodele
- Normalizacija

$$\frac{x - min}{max - min}$$

  - vrednosti se preslikavaju u opseg: $[0, 1]$
  - nije efikasna kada atribut ima ekstremne vrednosti
  - većina vrednosti može biti „sabijena“ u mali opseg

Mere sličnosti

- Kosinusna sličnost

$$\cos(p, q) = \frac{p \cdot q}{\|p\| \, \|q\|}$$

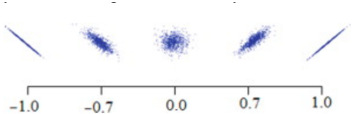
- Gde je:
  - p i q – dva vektora
  - $p \cdot q$ – skalarni proizvod vektora
  - $\|p\|$ i $\|q\|$ – dužine vektora
- Karakteristike:
  - često se koristi za asimetrične podatke
  - jedna od najčešćih mera sličnosti dokumenata

Mere sličnosti – korelacija

- Korelacija

$$r = \frac{\text{kovarijansa}(x, y)}{\text{standardna\_devijacija}(x) \cdot \text{standardna\_devijacija}(y)}$$

- Gde su x i y dva vektora
- Karakteristike:
  - korelacija meri linearni odnos između atributa
  - može se koristiti za binarne i neprekidne atribute
- Tumačenje:
  - $r \approx 1$ – jaka pozitivna povezanost
  - $r \approx -1$ – jaka negativna povezanost
  - $r \approx 0$ – nema linearne povezanosti



Mere sličnosti – korelacija

$$r = \frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum(x_i - \bar{X})^2 \sum(y_i - \bar{Y})^2}}$$

$$\text{kovarijansa}(x, y) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y})$$

$$\text{standardna devijacija}(z) = s_z = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})^2}$$

$$\text{srednja vrednost od } z = \bar{z} = \frac{1}{n} \sum_{k=1}^{n} z_k$$

Mere sličnosti za kategoričke podatke

- Najjednostavniji slučaj:

$$S(x_i, y_i) = \begin{cases} 1 & \text{ako } x_i = y_i \\ 0 & \text{inače} \end{cases}$$

- Ne uzima se u obzir relativna frekvencija atributa
- Koristi se agregiranje statističkih osobina
- Manje frekventne vrednosti koje su uparene imaju veću težinu
- Primer: poklapanje retke mutacije gena značajnije je od poklapanja česte vrednosti

Podaci sa kvantitativnim i kategoričkim atributima

Sličnost dva sloga

$$\overline{X} = (\overline{X}_n, \overline{X}_c) \quad \text{i} \quad \overline{Y} = (\overline{Y}_n, \overline{Y}_c)$$

sa „mešanim“ (kvantitativnim i kategoričkim) atributima:

$$Sim(\overline{X}, \overline{Y}) = \lambda \times NumSim(\overline{X}_n, \overline{Y}_n) + (1 - \lambda) \times CatSim(\overline{X}_c, \overline{Y}_c)$$

gde λ određuje relativnu važnost kategoričkih i numeričkih atributa

Edit rastojanje – rastojanje za transformacije

$$\overline{X} = (x_1, x_2, \ldots, x_m) \quad \text{u} \quad \overline{Y} = (y_1, y_2, \ldots, y_n)$$

Za prvih i simbola iz $\overline{X}$ i prvih j simbola iz $\overline{Y}$, cena transformacije je:

$$Edit(i,j) = \min \begin{cases} Edit(i-1,j) + \text{cena brisanja} \\ Edit(i,j-1) + \text{cena umetanja} \\ Edit(i-1,j-1) + I_{ij} \times \text{cena zamene} \end{cases}$$

gde je I_{ij} indikator jednakosti i-tog simbola iz $\overline{X}$ i j-tog simbola iz $\overline{Y}$

Koristi se u:

- poravnanju sekvenci,
- poređenju DNK i RNK sekvenci,
- filogenetskim analizama

Primer edit rastojanja u bioinformatici

Poređenje DNK sekvenci $S_1 = ACTGACCTGA$ i $S_2 = ATTGTCGA$

Jedna moguća transformacija:

$$A\mathbf{C}TGACCTGA \to A\mathbf{T}TGACCTGA \qquad (C \to T)$$
$$ATTG\mathbf{A}CCTGA \to ATTG\mathbf{T}CCTGA \qquad (A \to T)$$
$$ATTGTC\mathbf{CT}GA \to ATTGTCGA \qquad (\text{brisanje } CT)$$

Ukupno:

- 2 zamene
- 2 brisanja

$$Edit(S_1, S_2) = 4$$

- Veće edit rastojanje znači veću evolutivnu udaljenost
- Razlike mogu predstavljati: mutacije, insercije, delecije

Sličnost na osnovu najduže zajedničke podniske

Za prvih i simbola iz $\overline{X} = (x_1, x_2, \ldots, x_m)$ i prvih j simbola iz $\overline{Y} = (y_1, y_2, \ldots, y_n)$ u oznaci $\overline{X}_i$ i $\overline{Y}_j$, najduža zajednička podniska (eng. *Longest Common SubSequence, LCSS*)

$$LCSS(i,j) = \max \begin{cases} LCSS(i-1, j-1) + 1 & \text{ako } x_i = y_j \\ LCSS(i-1, j) & x_i \text{ nije uparen} \\ LCSS(i, j-1) & y_j \text{ nije uparen} \end{cases}$$

- Veća vrednost označava veću sličnost
- Koristi se za:
  - poređenje DNK i proteinskih sekvenci,
  - pronalaženje konzerviranih regiona,
  - filogenetske analize,
  - detekciju sličnih bioloških obrazaca

Primena LCSS u bioinformatici

Poređenje DNK sekvenci $S_1 = ACGTACCGTA$ i $S_2 = ACTACGTA$. Najduža zajednička podniska: ACTACGTA ili jedna od mogućih zajedničkih podniski.

Primer uparivanja:

$$\begin{array}{cccccccccc} A & C & G & T & A & C & C & G & T & A \\ | & | & & | & | & | & & | & | & | \\ A & C & & T & A & C & & G & T & A \end{array}$$

- LCSS pronalazi najduži zajednički redosled simbola
- Nije neophodno da simboli budu susedni

Redukcija i transformacija podataka

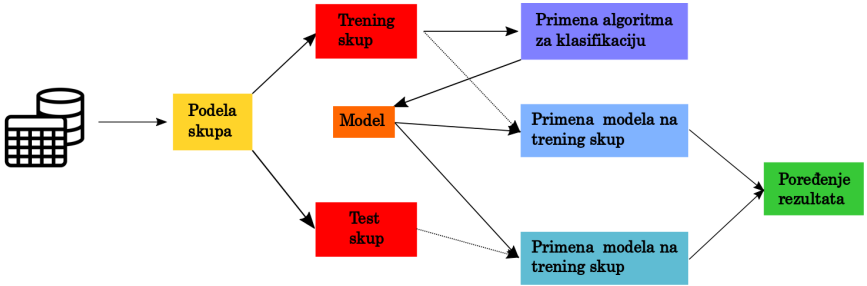
- Manja količina podataka omogućava efikasniju primenu algoritama
  1. Agregacija
  2. Uzimanje uzoraka
  3. Izbor karakteristika
  4. Redukcija pomoću rotacije osa
  5. Ostale metode dimenzione redukcije

Klasifikacija

Klasifikacija

- Ulazni podaci: svaki slog (instanca) je oblika (x, y) gde je x skup (ulaznih) atributa, a y je ciljni atribut (klasa).
- Cilj klasifikacije: pronaći funkciju f (model klasifikacije) koja preslikava skup atributa x u jednu od predefinisanih oznaka klasa y.
- Podela skupa na trening i test skup.

Klasifikacija



Merenje performansi izračunavanja

		Predviđena klasa	
		Klasa=Da	**Klasa=Ne**
Stvarna klasa	**Klasa=Da**	a TP	b FN
	Klasa=Ne	c FP	d TN

Najčešće korišćena metrika

$$\text{Tačnost} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+FN+FP+TN}$$

- Problem preprilagođavanja (eng. overfitting) - model se previše prilagodi specifičnostima trening podataka, zbog čega postiže visoku tačnost na trening skupu, ali daje loše rezultate na novim, neviđenim podacima.

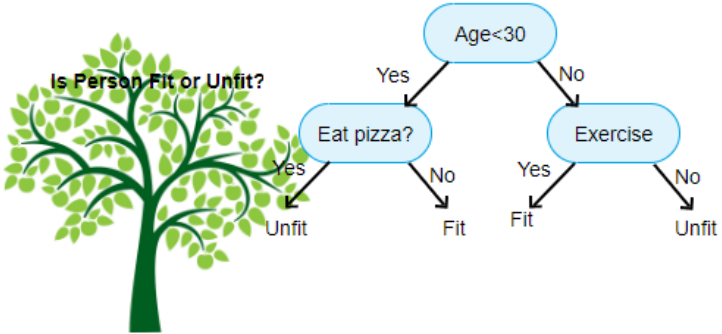
Tehnike klasifikacije

- Osnovne tehnike/metode klasifikacije
  - Drveta odlučivanja
  - Pravila
  - Neuronske mreže
  - Statistički zasnovane metode
  - Mašine sa potpornim vektorima
  - Određivanje najbližeg suseda
  - ...
- Metode ansambla
  - Pojačavanje (eng. *Boosting*)
  - Pakovanje (eng. *Bagging*)
  - Nasumična šuma (eng. *Random Forest*)
  - ...

Drveta odlučivanja

- Model klasifikacije se predstavlja kao drvo odlučivanja koje ima
  - unutrašnje čvorove. Svaki unutrašnji čvor sadrži uslov nad test atributom koji služi za podelu slogova koji imaju različite karakteristike tako da se dobiju *čistije* grupe slogova. Grane koje izlaze iz unutrašnjeg čvora odgovaraju mogućim vrednostima test atributa.
  - listove. Svakom listu je dodeljena jedna klasa.

Primer drveta odlučivanja



Izazovi

- Kako odabrati atribut(e) po kome se vrši podela?
- Kako formirati upit za različite tipove atributa?
- Kako odrediti najbolju podelu?
- Na koji način primeniti prethodne kriterijume na drvo po dubini?
- Kada stati sa konstrukcijom drveta?
- Način rada sa nedostajućim vrednostima?
- Cena i performanse modela?
- ...

Algoritmi

- ID3 (*Iterative Dichotomiser 3*)
- C4.5
- C5.0
- CART (*Classification And Regression Trees*)
- CHAID (*CHi-squared Automatic Interaction Detection*)
- Exhaustive CHAID
- QUEST (*Quick, Unbiased, Efficient Statistical Trees*)
- SLIQ (*Supervised Learning In Quest*)
- SPRINT (*Scalable PaRallelizable INduction and decision Trees*)
- ...

Mere nečistoće

$p(j|t)$ je relativna frekvencija klase j u čvoru t

Ginijev indeks

$Gini(t) = 1 - \sum_j [p(j|t)]^2$

Entropija

$Entropy(t) = -\sum_j p(j|t) * \log_2 p(j|t)$

Greška klasifikacije

$Error(t) = 1 - \max_j p(j|t)$

Drveta odlučivanja - podela prema tipu atributa

- Imenski atributi: binarna ili višestruka podela
- Redni atributi: binarna ili višestruka podela vodeći računa o uređenju
- Neprekidni atributi: potrebno je pronaći najbolju tačku/tačke prekida za binarna ili višestruku podelu

Karakteristike drveta odlučivanja

- Jeftina za konstruisanje
- Jako brza u klasifikaciji nepoznatog materijala
- Laka za interpretaciju za drveta male veličine
- Preciznost je uporediva sa ostalim tehnikama klasifikacije za jednostavne tipove podataka
- Izbor mere nečistoće nema veliki uticaj na performanse

Karakteristike drveta odlučivanja

- Primenljivost
  - na sve tipove podataka
- Izražajnost
  - mogu da predstave svaku funkciju diskretnih atributa
- Efikasnost izračunavanja
- Rad sa nedostajućim vrednostima
- Rad sa irelevantnim atributima i redundantnim atributima

Karakteristike drveta odlučivanja

- Slaba mogućnost rada sa povezanim atributima
  - spajanje dva atributa daje značajnu informaciju
- Kriterijum podele samo jedan atribut $\rightarrow$ interval je pravougaonog oblika
- Izbor načina potkresivanja značajno utiče na rezultate
- Problem preprilagođavanja i potprilagođavanja

- Uslovna verovatnoća

$$P(C|A) = \frac{P(A,C)}{P(A)}$$

- Verovatnoću da se zajedno dese događaj A i događaj C možemo računati sa

$$P(A,C) = P(C|A) * P(A)$$

kao i sa

$$P(A,C) = P(A|C) * P(C)$$

Naivni Bajesovski klasifikatori

- Bajesova teorema

$$P(C|A) = \frac{P(A|C) * P(C)}{P(A)}$$

- Pretpostavka o nezavisnosti između atributa

$$P(C|A_1, A_2, ..., A_n) = \frac{\prod_{i=1}^{n} P(A_i|C) * P(C)}{P(A)}$$

- Određivanje klase

$$\widehat{C} = \arg\max_{C} \prod_{i=1}^{n} P(A_i|C) * P(C)$$

Naivni Bajesovski klasifikatori – rezime

- Robusni su u odnosu na izolovani šum
- Barataju nedostajućim vrednostima ignorišući instancu pri izračunavanju procene verovatnoće
- Robusni su u odnosu na irelevantne atribute
- Pretpostavka nezavisnosti ne mora da važi za sve atribute
  - kada postoje zavisni atributi ili atributi u korelaciji, dobijene rezultate treba uzeti sa oprezom

K najbližih suseda

- K najbližih suseda - KNN
- Klasifikacija instance je zasnovana na sličnosti sa drugim instancama
- Test instanca se klasifikuje tako što se u trening skupu pronalazi k najbližih instanci (suseda) i test instanci se dodeljuje klasa koja je najzastupljenija među k najbližih suseda.

K najbližih suseda

Osnovni parametri algoritma KNN su:

- k - broj suseda
- *dist* - funkcija za računanje rastojanja između instanci

K najbližih suseda

- U najjednostavnijem obliku algoritma KNN, svaki sused ima istu težinu, te klasa koja je najzastupljenija među susedima test instance se dodeljuje test instanci.
- Nekada je bolje susedima dodeliti težine tako da bliži susedi imaju veći uticaj pri određivanju klase test instance. Tada su težine suseda proporcionalne vrednosti $\frac{1}{d}$, gde je d rastojanje suseda od test instance.

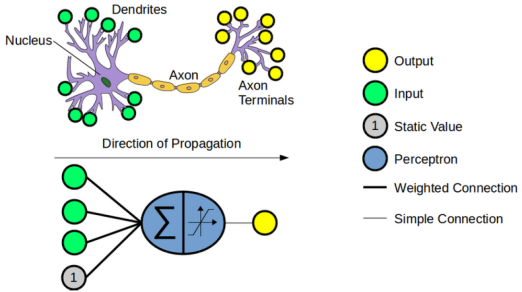
K najbližih suseda

Za računanje rastojanja između instanci najčešće se koriste Euklidsko i Menhetn rastojanje, zbog čega je u okviru preprocesiranja podataka potrebno izvršiti:

- pretvaranje kategoričkih atributa u numeričke
- transformisati numeričke atribute tako da imaju isti ili sličan opseg vrednosti

Veštačke neuronske mreže (VNM)

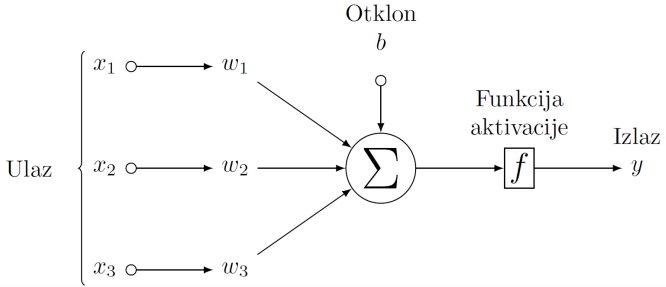
- Ideja: simulacija rada bioloških nervnih sistema
- Analogno strukturi u mozgu, VNM čine:
  - čvorovi,
  - veze između njih
- U VNM čvorovi predstavljaju:
  - neurone ili jedinice



Perceptron

- Najjednostavnija verzija VNM
- Modelira pojedinačnu ćeliju
- Dva tipa: ulazni i izlazni
- Čvorovi su povezani vezom sa težinama
- Težine simuliraju jačinu sinaptičke veze u biološkim neuronima

- Treniranje (obučavanje) perceptrona uključuje promenu vrednosti težina
- Treniranje traje dok se ne sinhronizuju ulazno/izlazne zavisnosti podataka



Perceptron

- Računa izlaznu vrednost $\overline{y}$ kao težinsku sumu ulaznih vrednosti uz oduzimanje otklona (eng. *bias*) uz proveru znaka rezultata
- Prethodni perceptron za vrednosti težina 0.3 i otklona 0.4 predstavlja model za izračunavanje:

$$\overline{y} = \begin{cases} 1, & \text{ako } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1, & \text{ako } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0 \end{cases}$$

Matematički, izlazni model perceptrona je jednak:

$$\overline{y} = sign(w_d x_d + w_{d-1} x_{d-1} + \cdots + w_2 x_2 + w_1 x_1 - b)$$

$$= sign(\mathbf{w} \cdot \mathbf{x})$$

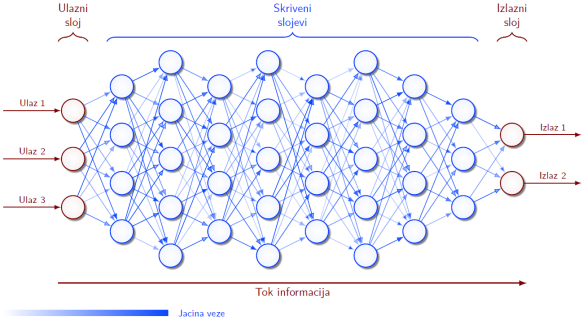
Učenje modela

$$w_j^{(k+1)} = w_j^{(k)} + \eta(y_i - \hat{y}_i^{(k)})x_{ij}$$

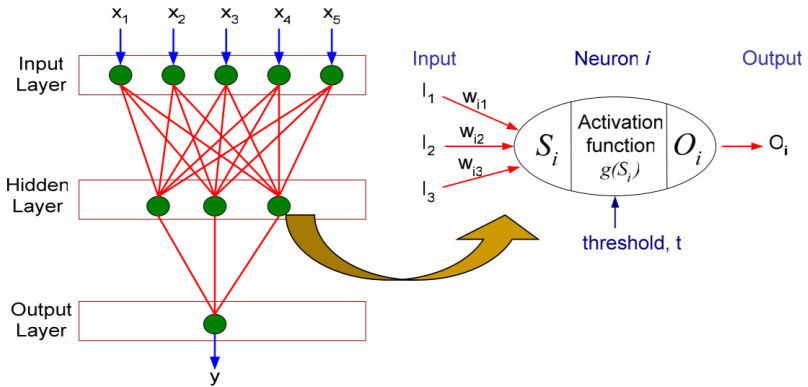
Za

- $y = +1$ i $\hat{y} = -1$ važi $(y - \hat{y}) = 2$
- $y = -1$ i $\hat{y} = +1$ važi $(y - \hat{y}) = -2$
- za linearno razdvojive probleme klasifikacije

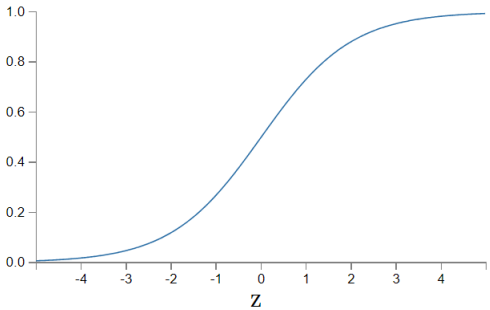
Neuronska mreža sa skrivenim slojevima



Neuronska mreža sa skrivenim slojevima



Koriste se i druge aktivcione funkcije (osim $sign$). Npr. pozadinska (eng. logistic/sigmoid) funkcija $\frac{1}{1+e^{-z}}$ kod koje mala promena u težinama dovodi do male promene u izlazu (rezultatu aktivacione funkcije).



Gradijentni spust je algoritam za nalaženje lokalnog minimuma funkcije.

Cilj - minimizacija zbira kvadrata greške

$$E(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Učenje modela

$$w_j = w_j - \eta * \frac{\partial E(w)}{\partial w_j}$$

Pristup gradijentni spust

- Kod višeslojnih mreža, problem je što stvarn izlazi skrivenih čvorova nisu poznati, jer ne postoje trening oznake za njihove izlaze.
- Potrebna neka vrsta „povratne informacije“ iz kasnijih slojeva ka ranijim slojevima o očekivanim izlazima i greškama radi ažuriranja težina tih čvorova.

Pristup gradijentni spust

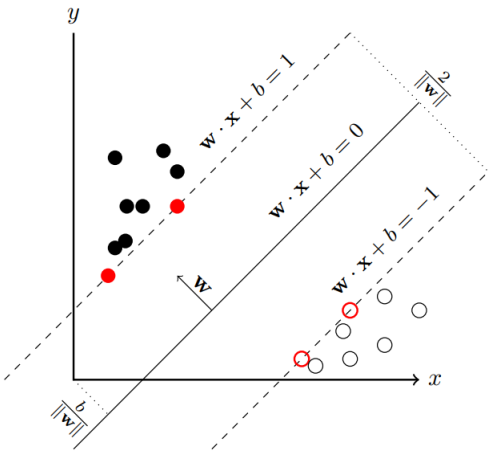
Algoritam propagacije unatrag (eng. backpropagation).
Faze:

- *unapred* - težine iz prethodne iteracije se koriste za računanje izlazne vrednosti svakog neurona. Konačni izlaz se poredi sa klasnom oznakom kako bi se utvrdilo da li je došlo do greške.
- *unazad* - formula za ažuriranje težina se primenjuje u obrnutom smeru

Ciljni atribut sa više klasa

- jedan izlazni neuron za svaku klasu, bira se klasa čiji neuron da najveći izlaz
- funkcija aktivacije izlaznih neurona - funkcija mekog maksimuma (eng. softmax)
  $softmax(x) = (\frac{e^{x_1}}{\sum_{i=1}^{C} e^{x_i}}, ..., \frac{e^{x_C}}{\sum_{i=1}^{C} e^{x_i}})$
  gde je x_i izlaz neurona vezanog za klasu i.
- funkcija gubitka - unakrsna entropija

Metod podržavajućih (potpornih) vektora - linearno separabilan skup



Metod podržavajućih (potpornih) vektora - linearno separabilan skup

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

$$w * x + b = 0$$

- w je normalni pravac hiper-ravni, a b je skalar, tj. pomak

Metod podržavajućih (potpornih) vektora - linearno separabilan skup

- Najbliže instance trening skupa iz različitih klasa nazivaju se podržavajući (potporni) vektori
- Maksimalna margina se određuje rešavanjem optimizacionog problema nelinearnog programiranja koji maksimizuje marginu tako što se ona izražava kao funkcija koeficijenata razdvajajuće hiper-ravni

Metod podržavajućih (potpornih) vektora - linearno separabilan skup

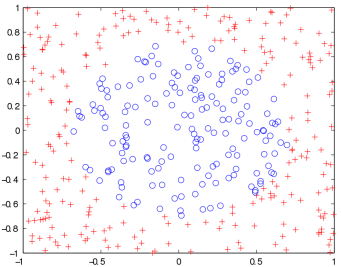
- Ograničenja margina

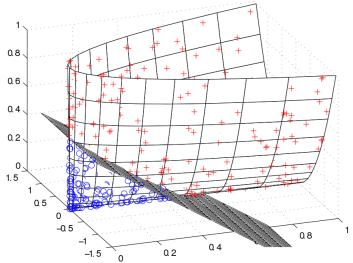
$$w * x + b = 1$$

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

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

Nelinearni metod podržavajućih (potpornih) vektora





Ideja: odrediti funkciju $\Phi : \mathbb{R}^m \to \mathbb{R}^n$ tako da hiperravan $w \cdot \Phi(x) + b = 0$ razdvaja transformisane podatke.

Nelinearni metod podržavajućih (potpornih) vektora

- Moguća mera sličnosti u transformisanom prostoru:

$$\Phi(x_i) \cdot \Phi(x_j)$$

- Kernel trik – računanje sličnosti u transformisanom prostoru korišćenjem originalnog skupa atributa.
- Prednost: ne tretira se eksplicitno n-dimenzioni prostor.
- Kernel funkcija – funkcija sličnosti K koja se računa pomoću originalnog skupa atributa.

Neke kernel funkcije:

- Gausov radijalni bazni kernel (RBF):

$$K(X_i, X_j) = e^{-\frac{\|X_i - X_j\|^2}{2\sigma^2}}$$

- Polinomijalni kernel:

$$K(X_i, X_j) = (X_i \cdot X_j + c)^h$$

- Sigmoidni kernel:

$$K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$$

Klasifikacija - mere za ocenu modela

Najčešće korišćene mere za ocenu modela:

- preciznost $= \frac{\text{Broj slogova čija klasa je dobro predviđena modelom}}{\text{Ukupan broj slogova}}$ (eng. accuracy)
- stopa greške $= \frac{\text{Broj slogova čija klasa nije dobro predviđena modelom}}{\text{Ukupan broj slogova}}$ (eng. error rate)

Nisu dovoljne za skupove podataka sa neuravnoteženim klasama.

Klasifikacija - mere za ocenu modela

Za bolji uvid kako se model ponaša za svaku klasu koristi se matrica konfuzije.

Table: Matrica konfuzije za 4 klase

		Dodeljena klasa			
		C_1	C_2	C_3	C_4
Stvarna klasa	C_1				
	C_2		x		
	C_3				
	C_4	y			

Klasifikacija - mere za ocenu modela

- Postoje mere koje daju bolji uvid kako se model ponaša za svaku klasu.
- Pri binarnoj klasifikaciji, retka klasa se označava kao pozitivna a većinska klasa kao negativna

Klasifikacija - matrica konfuzije

		Dodeljena klasa	
		+	-
Stvarna klasa	+	TP	FN
	-	FP	TN

- preciznost (eng. precision) $p = \frac{TP}{TP+FP}$
  Za skup sa više klasa preciznost se računa za svaku klasu C_i sa

$$p = \frac{\text{broj instanci klase } C_i \text{ kojima model dodeljuje klasu } C_i}{\text{broj instanci kojima model dodeljuje klasu } C_i}$$

- odziv (eng. recall) $r = \frac{TP}{TP+FN}$
  Za skup sa više klasa odziv se računa za svaku klasu C_i sa

$$r = \frac{\text{broj instanci klase } C_i \text{ kojima model dodeljuje klasu } C_i}{\text{broj instanci klase } C_i}$$

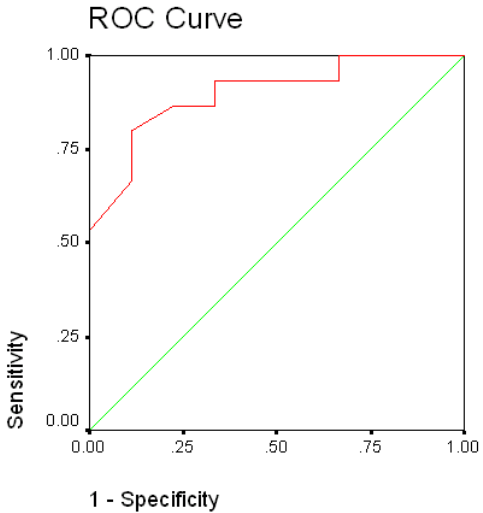
Klasifikacija - mere za ocenu modela

- F_1 uzima u obzir preciznost i odziv $F_1 = \frac{2rp}{r+p} = \frac{2}{\frac{1}{r}+\frac{1}{p}}$
  - harnomijska sredina preciznosti i odziva
  - bliža je manjoj vrednosti

Klasifikacija - mere za ocenu modela

- ROC kriva (receiver operating characteristic curve)
  - grafički prikaz kompromisa između TPR i FPR
  - x osa - FPR ili 1-specifičnost
  - y osa - TPR ili osetljivost

Klasifikacija - mere za ocenu modela



Klasifikacija - ROC kriva

- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0)

Klasifikacija - ROC kriva

- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0) - model svakoj instanci dodeljuje negativnu klasu

Klasifikacija - ROC kriva

- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0) - model svakoj instanci dodeljuje negativnu klasu
  - (TPR=1 i FPR=1)

Klasifikacija - ROC kriva

- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0) - model svakoj instanci dodeljuje negativnu klasu
  - (TPR=1 i FPR=1) - model svakoj instanci dodeljuje pozitivnu klasu

Klasifikacija - ROC kriva

- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0) - model svakoj instanci dodeljuje negativnu klasu
  - (TPR=1 i FPR=1) - model svakoj instanci dodeljuje pozitivnu klasu
  - (TPR=1 i FPR=0)

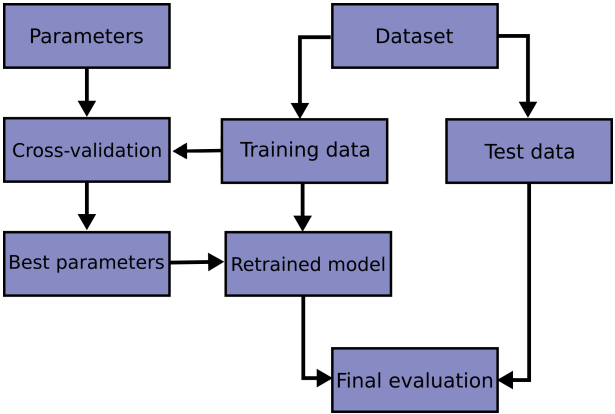
Klasifikacija - ROC kriva

- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0) - model svakoj instanci dodeljuje negativnu klasu
  - (TPR=1 i FPR=1) - model svakoj instanci dodeljuje pozitivnu klasu
  - (TPR=1 i FPR=0) - idealan model

Klasifikacija - ROC kriva

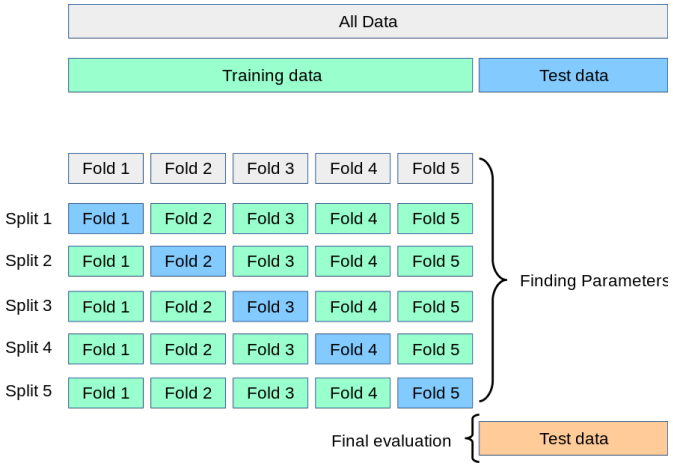
- Interpretacija određenih tačaka
  - (TPR=0 i FPR=0) - model svakoj instanci dodeljuje negativnu klasu
  - (TPR=1 i FPR=1) - model svakoj instanci dodeljuje pozitivnu klasu
  - (TPR=1 i FPR=0) - idealan model
- *AUC* (eng. area under the ROC curve) - površina ispod ROC krive

Unakrsna validacija



- Podešavanje hiper-parametara procenjivača

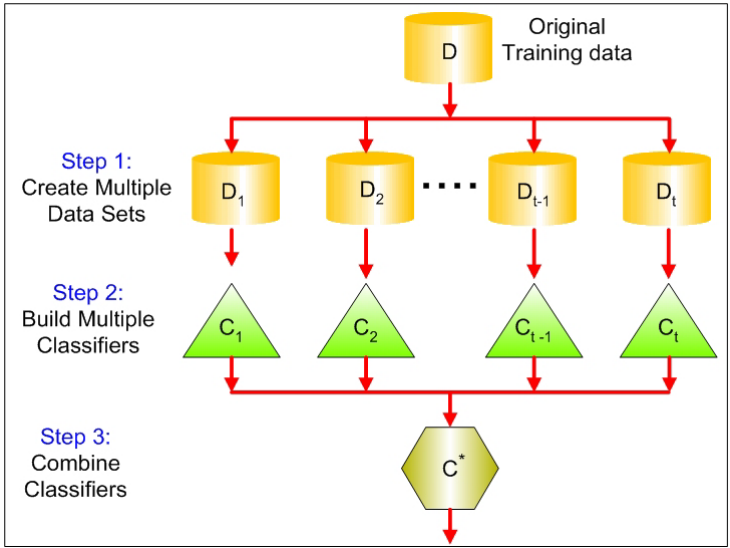
Unakrsna validacija



Ansambl metode

- Različiti klasifikatori mogu davati različite predikcije za iste test instance.
- Ansambl metod je pristup kojim se povećava tačnost predikcije kombinovanjem rezultata više klasifikatora.
- Osnovna ideja je da se rezultati različitih klasifikatora se kombinuju u jednu predikciju.
- Klasifikatori mogu biti razvijeni korišćenjem različitih algoritama ili istog algoritma na različitim trening skupovima

Ansambl metode



Ansambl metode

Ulaz: Trening skup D, bazni algoritmi $A_1, \ldots, A_r$, test instance $\mathcal{T}$

1. $j \leftarrow 1$
2. **ponavljaj**
   - izaberi algoritam $Q_j \in \{A_1, \ldots, A_r\}$
   - konstruši skup $f_j(D)$
   - nauči model M_j primenom Q_j nad $f_j(D)$
   - $j \leftarrow j + 1$
3. **do** ispunjenja uslova zaustavljanja
4. za svaku $T \in \mathcal{T}$ vrati klasu kombinovanjem predikcija svih modela M_j

Vrste ansambl metoda

- **Ansambl metode usmerene na podatke** - koristi se jedan algoritam učenja (npr. SVM ili stablo odlučivanja), a glavna varijacija se odnosi na način na koji se konstruiše izvedeni skup podataka. Može se koristiti uzorkovanje podataka, fokusiranje na pogrešno klasifikovane delove trening skupa u prethodnim komponentama, manipulacija atributa ili manipulacija klasnim oznakama.
- **Ansambl metode usmerene na modele** - koriste se različiti algoritmi, a skup za svaku komponentu ansambla je originalni skup.

Uzroci grešaka klasifikatora

- **Pristrasnost** (eng. bias) - svaki klasifikator pravi određene pretpostavke o prirodi granice odlučivanja između klasa.
- **Varijansa** - slučajne varijacije u izboru trening podataka dovode do različitih modela.

Različiti pristupi ansambla povećavaju tačnost smanjenjem uticaja pristrasnosti, varijanse ili kombinacije oba.

Bagging

- **Pakovanje** (Bagging skraćeno od bootstrapped aggregating) je pristup koji pokušava da smanji varijansu predikcije.
- Instance se uzorkuju uniformno iz originalnog skupa sa vraćanjem (eng. bootstrap).
- Izvlači se uzorak iste veličine kao originalni skup.
- Taj uzorak može sadržati duplikate i tipično sadrži približno 63% početnog skupa jer se svaki uzorak bira sa verovatnoćom $1 - (1 - 1/N)^N$
- Ako je N dovoljno veliko verovatnoća konvergira ka $1 - 1/e \approx 0,632$
- Za datu test instancu, ansambl vraća klasu za koju je glasala većina različitih klasifikatora.

Boosting

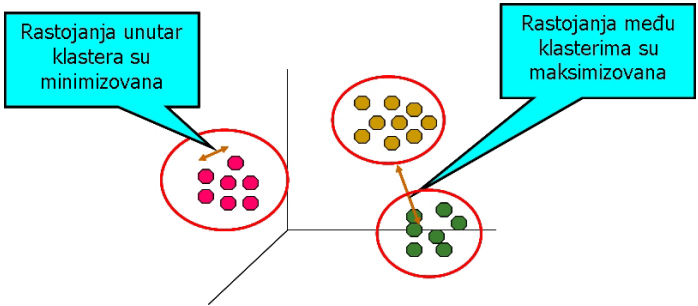
- **Pojačavanje** (eng. boosting)
- Svakom trening primeru dodeljuje se težina, i klasifikatori se treniraju uz korišćenje tih težina.
- Težine se iterativno menjaju na osnovu uspeha klasifikatora čime budući modeli zavise od prethodnih.
- Ideja je da se u narednim iteracijama fokus prebaci na pogrešno klasifikovane instance povećanjem njihove težine.
- Pretpostavka je da su greške posledica pristrasnosti.

Slaganje modela

- Slaganje modela (eng. stacking)
- Opšti pristup sa dva nivoa klasifikacije.
- Trening podaci se dele na A i B.
  1. Na skupu A se trenira k klasifikatora.
  2. Na skupu B se izračunaju izlazi svih k klasifikatora. Formira se novi skup od k atributa gde je svaki atribut izlaz jednog klasifikatora, sa klasom. Zatim se trenira klasifikator nad tom novom reprezentacijom.
- Za test instancu, modeli prvog nivoa generišu k izlaza, a klasifikator drugog nivoa na osnovu njih daje finalnu predikciju.
- Može smanjiti i bias i varijansu

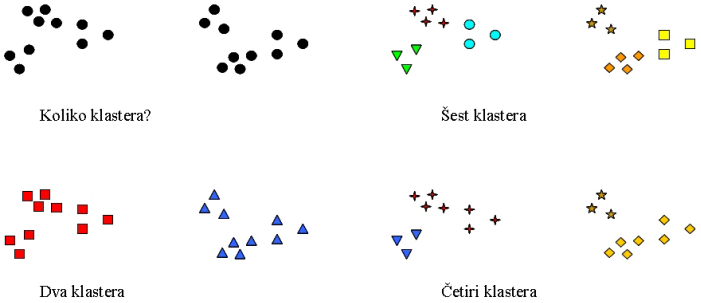
Klasterovanje

Pronalaženje grupa objekata takvih da su objekti u grupi međusobno slični (ili povezani), i da su objekti u različitim grupama međusobno različiti (ili nepovezani).



Dvosmislenost pojma klastera

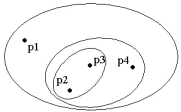
Broj klastera zavisi od posmatranog kriterijuma.



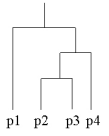
Tipovi klasterovanja

U slučaju da klasteri mogu da sadrže (ugneždene) klastere, tada jedan element može da pripada više klastera na različitim nivoima hijerarhije – **hijerarhijsko klasterovanje**.

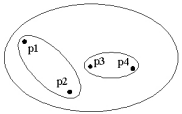
Prikaz hijerarhije klastera se često naziva **dendogram**, i dosta često je u upotrebi u prirodnim naukama, pogotovu u biologiji.



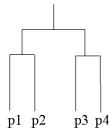
Tradicionalno hijerarhijsko klasterovanje



Tradicionalni dendogram



Netradicionalno hijerarhijsko klasterovanje



Netradicionalni dendogram

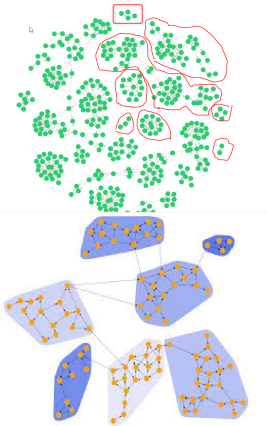
Tipovi klastera

Klasteri zasnovani na centru (eng. *center-based*, *prototype-based*

Klaster je skup objekata takvih da je bilo koji objekat u klasteru bliži (ili više sličan) prototipu (*centru*) klastera u odnosu na prototipove (centre) ostalih klastera. Centar klastera je često centroid (prosek svih tačaka u klasteru) ili medoid (najreprezentativnija tačka u klasteru).

Tipovi klastera

Klasteri zasnovani na grafovima (eng. *graph based*)



Ako su elementi predstavljeni kao čvorovi povezanog grafa, tada klasteri mogu da budu skupovi objekata koji su međusobno povezani, ali nisu povezani sa objektima van grupe, odnosno koji pripadaju izolovanom podgrafu.

Neke definicije dopuštaju da između klastera (podgrafova) postoje veze, ali u mnogo manjem broju (ili sa mnogo većim rastojanjem) nego između elemenata podgrafa.

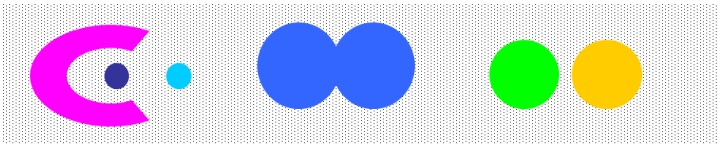
Tipovi klastera

Klasteri zasnovani na susedstvu (eng. *contiguous based clusters*)

Klasteri zasnovani na susedstvu predstavljaju vrstu klastera zasnovanih na grafovima kod kojih **dva elementa pripadaju istom klasteru ako su na rastojanju koje je manje od unapred definisanog praga**. Posledica ovakvog uslova je da za svaki element koji pripada ovom tipu klastera postoji element iz istog klastera kome je on bliži nego bilo kom elementu koji pripada drugom klasteru.

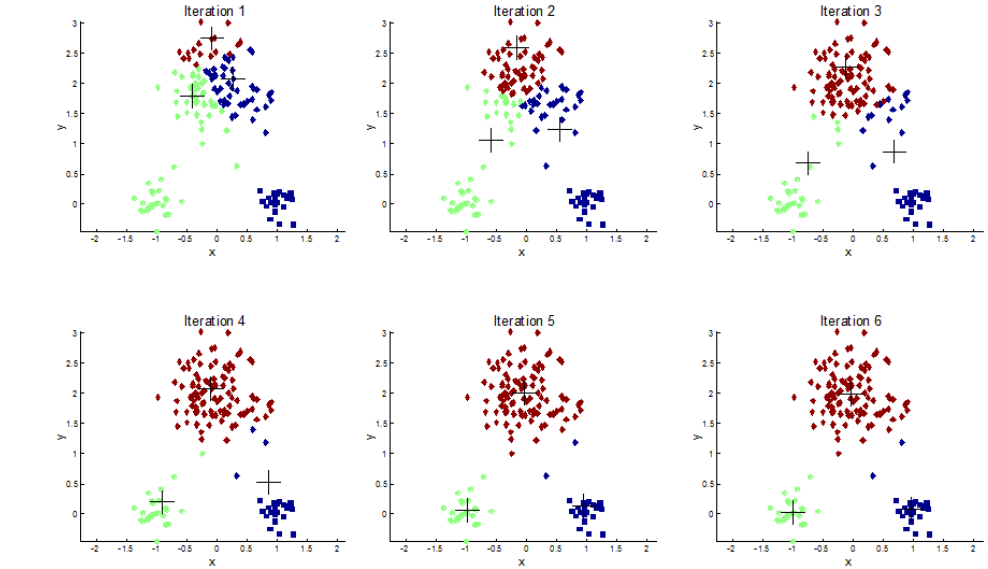
Tipovi klastera

Klasteri zasnovani na gustini (eng. *density-based*)



Klasteri su oblasti sa velikom gustinom tačaka koje su razdvojene oblastima sa malom gustinom tačaka. Ova karakteristika klastera se koristi kada su klasteri nepravilni ili isprepleteni, i kada su prisutni šum ili elementi van granica.

Algoritam k-sredina: primer

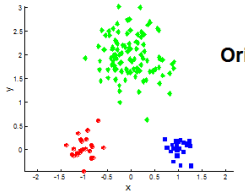


Algoritam k-sredina

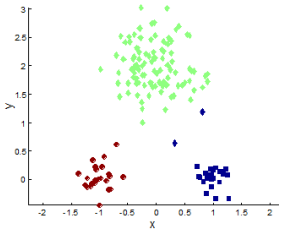
- Za određivanje rastojanja mogu da se koriste različite mere. Algoritam konvergira, pri čemu se najveći deo konvergencije dešava u prvih nekoliko iteracija.
- Kao uslov zaustavljanja algoritma se zadaje broj tačaka koje promene klaster u određenoj iteraciji. Ako je broj tačaka koje promene klaster manji od zadatog praga, algoritam se zaustavlja.
- Za proveru rezultata se u Euklidskom prostoru najčešće kao mera koristi zbir kvadrata grešaka (eng. *sum of squared errors*, SSE).

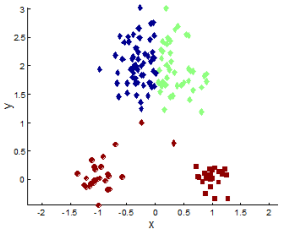
Optimalno i suboptimalno klasterovanje

Izbor početnog centroida je jako važan jer može da dovede do nekorektnih rezultata. U grupu nekorektnih rezultata spada i tzv. sub-optimalno klasterovanje u kom slučaju je izvršeno klasterovanje materijala, ali nije dobijena globalno već samo lokalno najmanja vrednost SSE.



Original Points





Izbor početnih centroida

Različite tehnike mogu da se primene radi poboljšanja dobijenih rezultata ili povećanja šansi za dobijanje kvalitetnijih rezultata. Jedan deo tehnika se odnosi na **izbor početnih centroida**, dok je drugi **orijentisan na dodatnu obradu dobijenih rezultata**. Moguće tehnike su:

- Uzastopna izvršavanja algoritma
  - Svako izvršavanje sa npr. slučajno izabranim centroidima.
  - Između njih se izabere klaster sa najmanjim SSE.
- Nad uzorcima se primeni hijerarhijsko klasterovanje i izaberu početni centroidi.
- Izabere se m ($m > k$) početnih centroida i biraju se "dobri" centroidi između njih.
  - Da bi ovaj način bio uspešan potrebno je da izabrani kandidati za centroide pokrivaju što širi prostor.
- Primeniti pristup K-sredine++.
- Primeniti metodu bisekcije K-sredina.
- Izvršiti postprocesiranje dobijenih rezultata.

Rad sa praznim klasterima

Strategije za eliminaciju praznih klastera uključuju zamenu centroida na neki od sledećih načina:

- Izabrati tačku koja najviše učestvuje u SSE.
- Izabrati tačku koja je najdalje od tekućih centroida.
- Izabrati tačku iz klastera sa najvećim SSE. Ovaj način obično dovodi do deobe klastera.
- Ako ima više praznih klastera ponoviti postupak.

Nedostaci algoritma k-sredina

Nedostaci i ograničenja algoritma k-sredina su:

- ne funkcioniše za klastere proizvoljnog oblika;
- ne funkcioniše za klastere različitih gustina;
- osetljiv je na elemente van granica koji mogu da dovedu do jediničnih ili praznih klastera;
- problem predstavlja određivanje reprezentativnih predstavnika i broja klastera k.

Dobre strane algoritma k-sredina

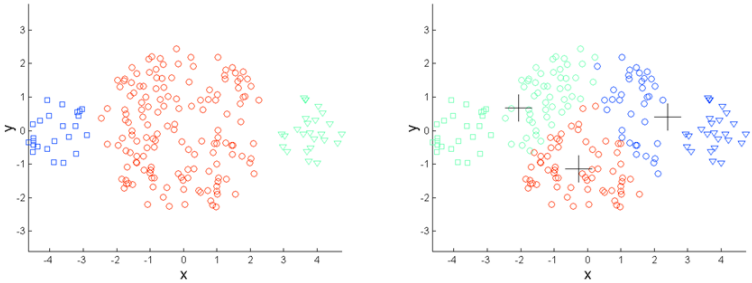
Dobre strane algoritma k-sredina su:

- jednostavnost implementacije i primene;
- najbolje radi sa globularnim podacima;
- ako se kao mera rastojanja koristi Mahalanobisovo rastojanje, algoritam k-sredina prepoznaje klastere različitih gustina.

Neki nedostaci i ograničenja algoritma k-sredina su ilustrovani na narednim slajdovima. U sva tri slučaja prikazana ograničenja mogu da se prevaziđu povećanjem broja klastera k i nalaženjem klastera koji su podklasteri prirodnih klastera.

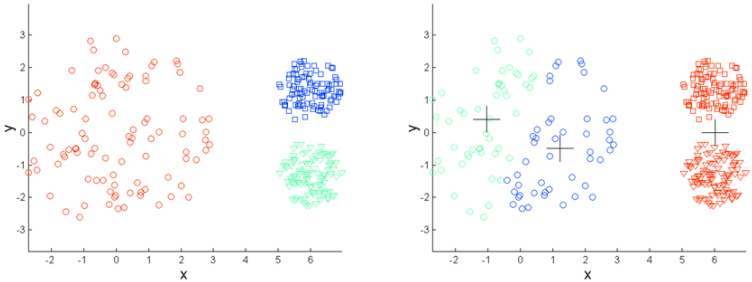
Ograničenja algoritma k-sredina

Primena algoritma k-sredina na klastere različitih veličina.



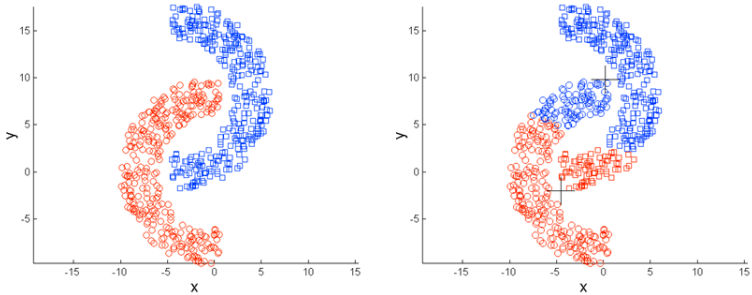
Ograničenja algoritma k-sredina

Primena algoritma k-sredina na klastere različitih gustina.



Ograničenja algoritma k-sredina

Primena algoritma k-sredina na ne-globularne klastere.

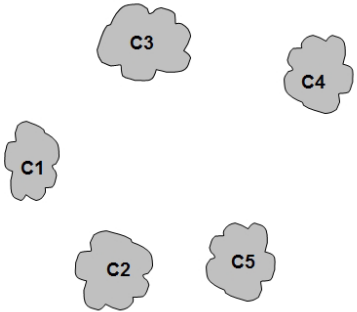


Algoritmi hijerarhijskog klasterovanja

- Algoritmi sakupljajućeg klasterovanja (eng. *agglomerative clustering*);
- Algoritmi razdvajajućeg klasterovanja (eng. *divisive clustering*);
- Karakteristika ove grupe algoritama je da se inicijalno ne navodi broj klastera koji će se formirati.

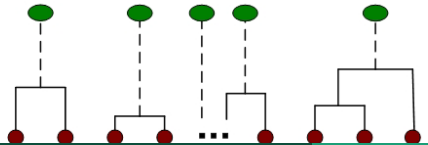
Algoritmi sakupljajućeg klasterovanja

- Problem – čuvanje matrice rastojanja.
- Formiranje novog klastera – matrica se modifikuje (ili se pravi nova).



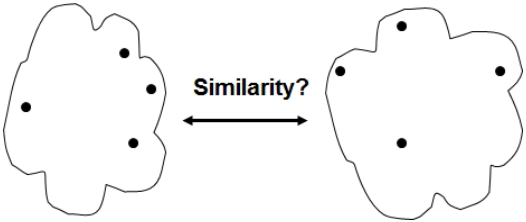
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



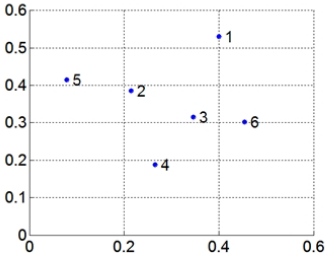
Sličnost klastera

U skladu sa opštim principom za izvršavanje algoritama (sakupljajućeg) hijerarhijskog klasterovanja koji podrazumeva spajanje dva klastera koja su najsličnija (na najmanjem rastojanju), postavlja se pitanje kako definisati funkcije i metode za izračunavanje sličnosti (rastojanja) dva klastera P i Q sa m i n elemenata.



Sličnost klastera

Na osnovu matrice sličnosti tekućeg skupa klastera, svaka od funkcija koja se koristi izračunava sličnost između para klastera i dobijenu vrednost upisuje u matricu sličnosti.



Distance Matrix:

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Metode za izračunavanje sličnosti klastera

1. najbolja, najkraća, pojedinačna veza;
2. najgora, najduža, kompletna veza;
3. prosečno rastojanje parova tačaka u klasterima;
4. rastojanje između centroida klastera;
5. novi klaster se formira od dva klastera čijim spajanjem se minimizuje promena varijanse unutar novodobijenog klastera;

Procena klasterovanja

- U mnogim slučajevima kriterijum za validaciju kvaliteta algoritma predstavlja ciljna funkcija koju optimizuje određeni model klasterovanja.
- Kada se kao mera bliskosti koristi rastojanje u Euklidskom prostoru, za evaluaciju klasterovanja algoritmom K-sredina često se koristi mera: Suma kvadrata greške (SSE – Sum of Squared Error)

Suma kvadrata greške (SSE)

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

gde je:

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

- Cilj je da SSE bude što manja.
- Problem ovih kriterijuma nastaje prilikom poređenja algoritama sa različitim metodologijama.

Odnos intraklasterskog i interklasterskog rastojanja

- Uzorkuje se r parova instanci iz osnovnog skupa podataka.
- P je skup parova koji pripadaju istom klasteru prema klasterovanju koje je pronašao algoritam.
- Preostali parovi pripadaju skupu Q.

Prosečna intraklasterska i interklasterska rastojanja definišu se kao:

$$Intra = \frac{1}{|P|} \sum_{(x_i,x_j)\in P} dist(x_i, x_j)$$

$$Inter = \frac{1}{|Q|} \sum_{(x_i,x_j)\in Q} dist(x_i, x_j)$$

$$\frac{Intra}{Inter}$$

Male vrednosti ukazuju na bolje ponašanje klasterovanja.

Silueta koeficijent

- Silueta koeficijent (eng. *Silhouette coefficient*) predstavlja meru koliko su instance grupisane sa instancama koje su slične njima samima.
- Silueta koeficijent računa se za svaku instancu formulom:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

gde je:

- $a(i)$ – prosečno rastojanje između instance i ostalih instanci u istom klasteru
- $b(i)$ – prosečno rastojanje između instance i svih instanci iz najbližeg susednog klastera

Tumačenje silueta koeficijenta

- Silueta koeficijent za ceo skup predstavlja prosečnu vrednost koeficijenta za pojedinačne instance.
- Vrednost silueta koeficijenta je u intervalu:

$$[-1, 1]$$

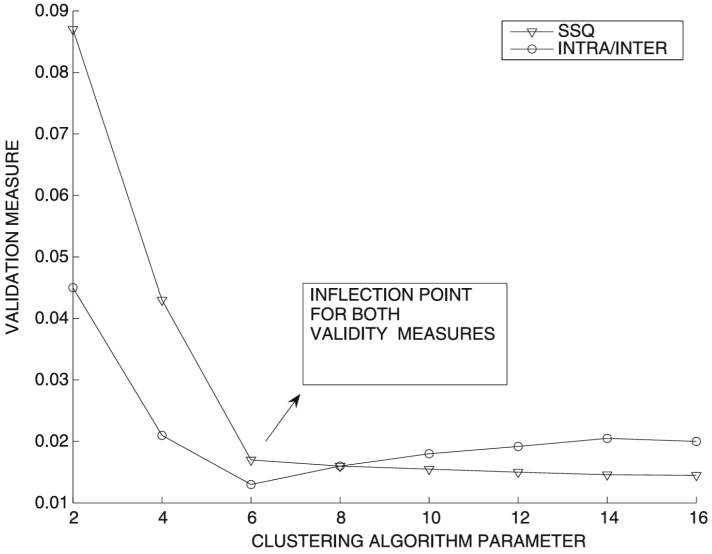
- -1 – neispravno grupisanje
- $+1$ – gusto grupisanje

- Veće vrednosti ukazuju na guste i dobro razdvojene klastere.

Podešavanje parametara klasterovanja

- Mere za procenu klasterovanja pogodne su za:
  - poređenje sličnih algoritama,
  - poređenje različitih pokretanja istog algoritma,
  - podešavanje parametara algoritma.
- Varijacija mere validacije može pokazati tačku prevoja ili tzv. *lakat* (eng. *elbow*).
  - Kod algoritma K-sredina može se koristiti za izbor broja klastera k.
  - SSE mera se smanjuje sa povećanjem broja klastera, ali sporije nakon tačke prevoja.
- Odnos intraklasterskog i interklasterskog rastojanja opada do tačke prevoja, a zatim može blago porasti.

Procena klasterovanja



Davies–Bouldin indeks

- Davies–Bouldin indeks predstavlja internu meru kvaliteta klasterovanja.
- Zasniva se na:
  - kompaktnosti klastera,
  - međusobnoj razdvojenosti klastera.

Davies–Bouldin indeks – formula

Davies–Bouldin indeks definiše se formulom:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{M_{ij}} \right)$$

gde je:

- k – broj klastera
- S_i – prosečno rastojanje instanci klastera i od centroida klastera
- M_{ij} – rastojanje između centroida klastera i i j

- Veće vrednosti S_i povećavaju indeks.
- Veće rastojanje između klastera smanjuje indeks.
- Manje vrednosti Davies–Bouldin indeksa ukazuju na bolje klasterovanje.