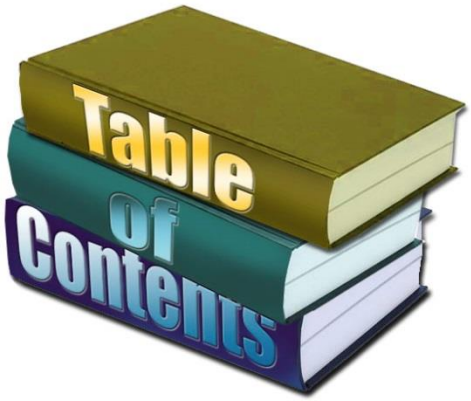
OSNOVE UMETNE INTELIGENCE
2018/19

nenadzorovano učenje
preiskovanje

del gradiva povzet po: https://lagunita.stanford.edu/courses

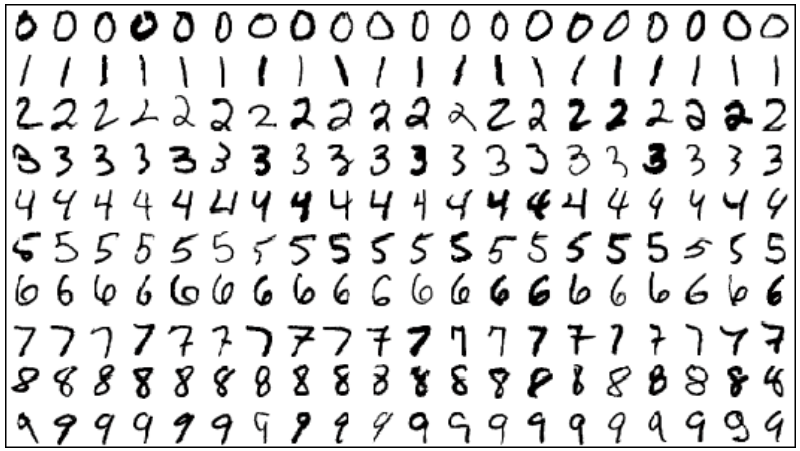
Pregled

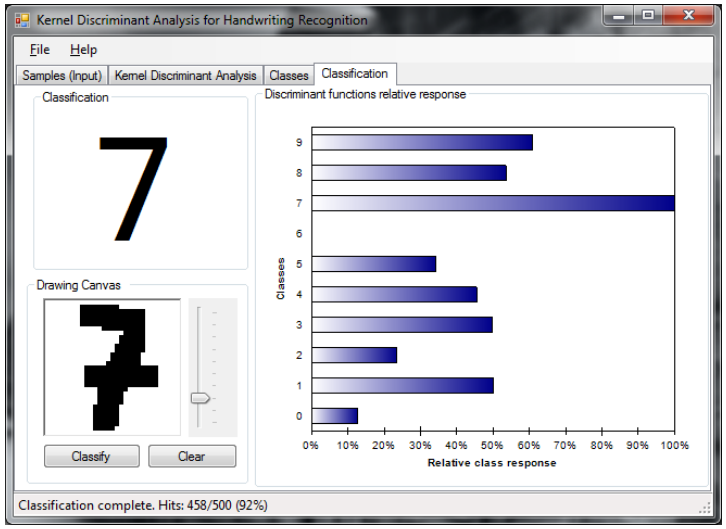


- strojno učenje
  - primer aplikacije nadzorovanega učenja
  - nenadzorovano učenje
    - hierarhično gručenje
    - metoda k voditeljev

Primer aplikacije nadzorovanega učenja

- razpoznavanje zapisanih števk (angl. *handwritten digit recognition*)
  - aktualen problem (sortiranje pošte, branje finančnih/davčnih dokumentov, branje obrazcev)
  - NIST (US National Institute of Science and Technology) je oblikoval klasificirano učno množico 60,000 števk – podatki so predstavljeni kot 8-bitne slike (sivinske) dimenzij 20×20=400 pikslov



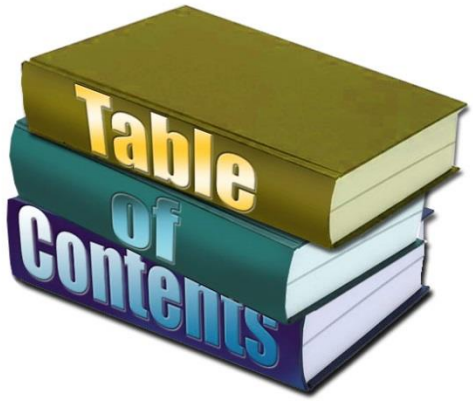


Primer aplikacije nadzorovanega učenja

- razpoznavanje zapisanih števk
  - človeška napaka: 0,2 – 2,5%

	3-NN	NN	LeNet	Lenet + boosting	SVM1	SVM2	prilaganje oblike
napaka [%]	2,4	1,6	0,9	0,7	1,1	0,56	0,63
čas [ms/števko]	1000	10	30	50	2000	200	?
spomin [MB]	12	0,49	0,012	0,21	11	?	?
čas učenja [dnevi]	0	7	14	30	10	?	?

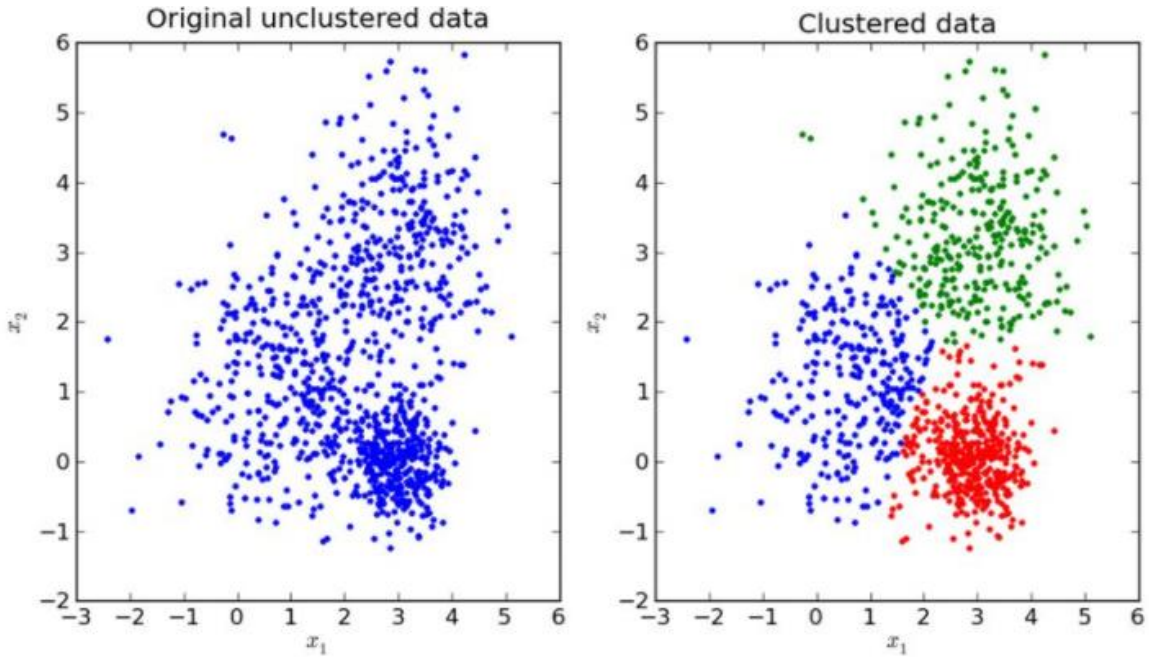
Pregled



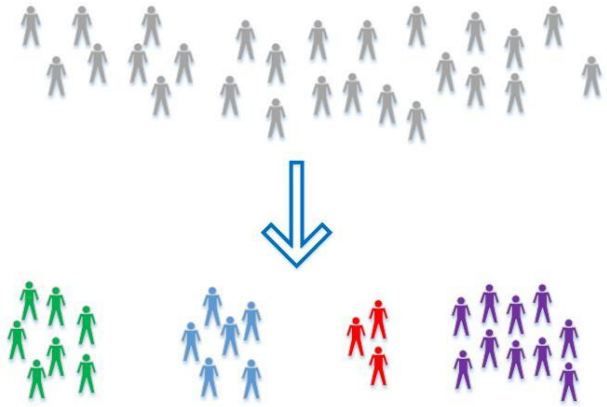
- strojno učenje
  - primer aplikacije nadzorovanega učenja
  - nenadzorovano učenje
    - hierarhično gručenje
    - metoda k voditeljev

Nenadzorovano učenje

- drugačni scenarij in cilji učenja kot pri nadzorovanem učenju:
  - nimamo ciljne (odvisne) spremenljivke, zato nas ne zanima napoved primera
  - podani so samo atributi primerov
- cilj: odkrivanje zakonitosti glede učnih primerov. Vprašanja:
  - ali lahko primere razdelimo v smiselne skupine?
  - ali obstaja priročen način za vizualizacijo podatkov?



Nenadzorovano učenje

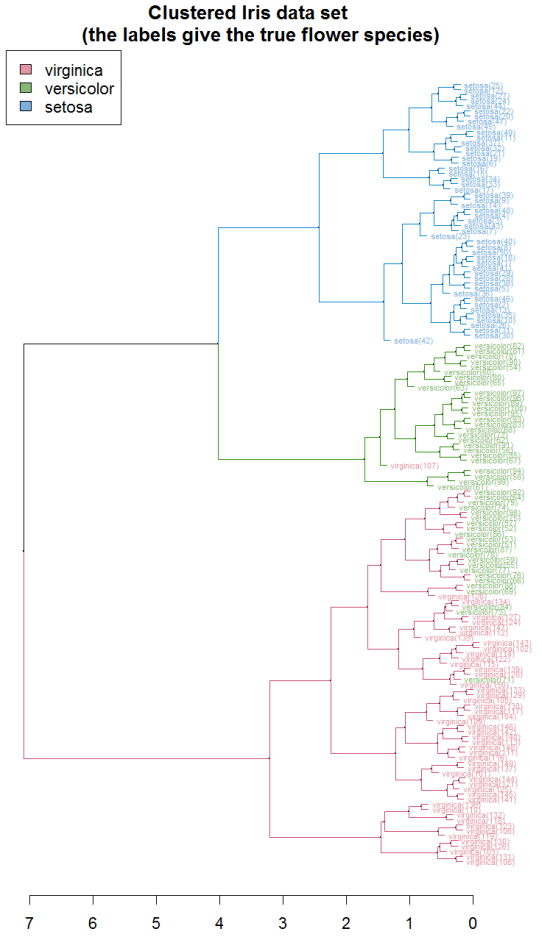


- **lastnosti**:
  - nenadzorovano učenje je **bolj subjektivno** kot nadzorovano učenje, ker nima enoznačnega formalnega cilja kot je "napovedovanje vrednosti odvisne spremenljivke" pri nadzorovanem učenju
  - velikokrat lažje (cenejše) pridobimo neoznačene podatke (podatke brez odvisne spremenljivke): drage meritve, ekspertno mnenje, globalna ocena (npr. filma)?

- **primeri uporabe**:
  - odkrivanje skupin rakavih bolnikov, grupiranih po različnih rezultatov meritev izraženosti genov,
  - odkrivanje skupin kupcev, grupiranih po njihovi zgodovini brskanja in nakupovanja
  - odkrivanje skupin filmov, grupiranih glede na ocene, podane s strani gledalcev

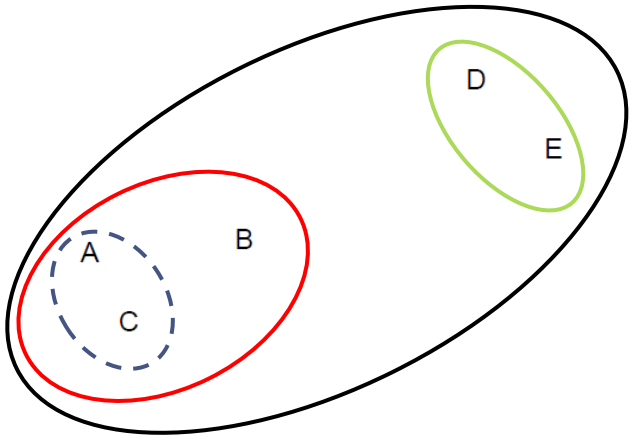
Gručenje

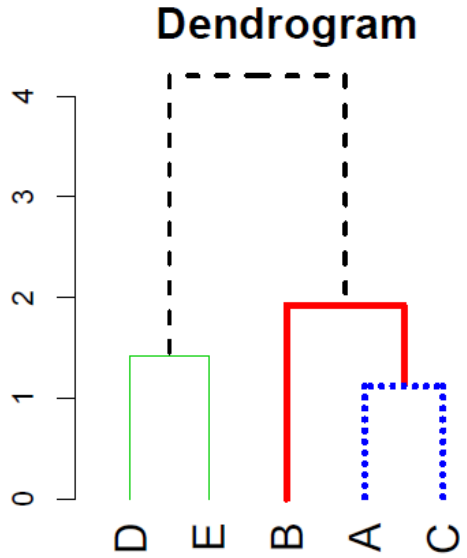
- **gručenje (angl. clustering)** je najbolj uporabljana metoda nenadzorovanega učenja
- **cilj**: iskanje homogenih podskupin v učnih podatkih
- metode:
  - **hierarhično gručenje**: iščemo vnaprej neznano število gruč. Rezultat gručenja je vizualna reprezentacija skupin, imenovana dendrogram, ki nam nudi vpogled v oblikovanje različnega števila gruč
  - **metoda k-means**: gručimo v vnaprej podano število *k* gruč



Hierarhično gručenje

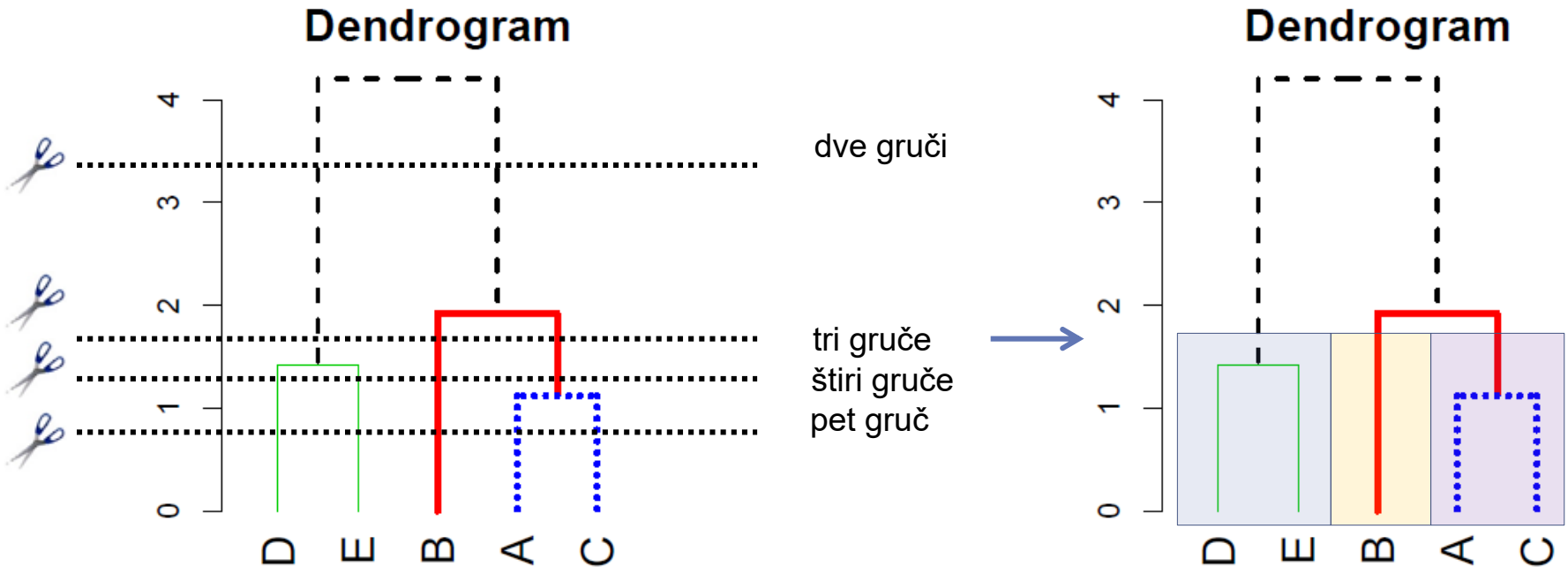
- dva pristopa:
  - **združevalni** (angl. *agglomerative*): gradnja dendrograma začenši od listov proti korenu s postopkom **združevanja** glede na razdaljo
  - **delilni** (angl. *divisive*): gradnja dendrograma od korena proti listom, na vsakem koraku **delimo** gručo na podgruče
- primer združevalnega pristopa:
  - začni z vsako točko v svoji gruči
  - najdi dve najbližji gruči in ju združi
  - ponavljaj, dokler ne združiš vseh gruč





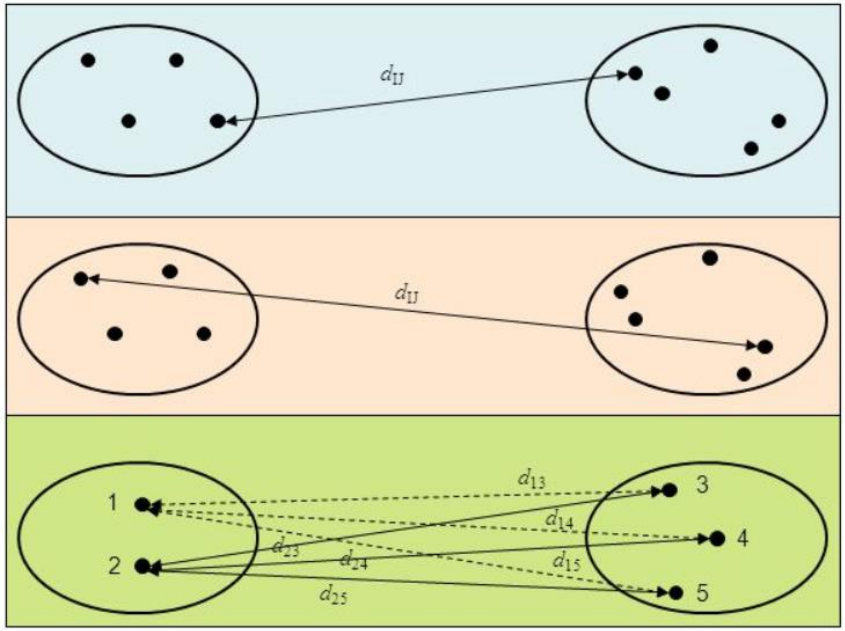
Interpretacija dendrograma

- rezanje dendrograma določi mejo, pri kateri prenehamo z združevanjem gruč
- z rezanjem dendrograma na različnih višinah torej določamo število ciljnih gruč



Merjenje razdalj

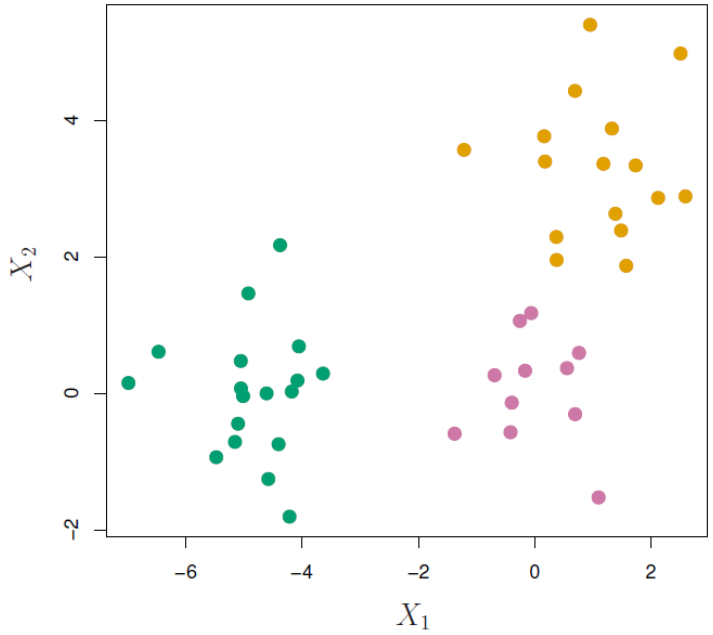
- med učnimi primeri uporabljamo že znane mere za merjenje razdalj (evklidska razdalja, manhattanska razdalja, korelacija med vrednostmi atributov ...)
- posebno obravnavo moramo posvetiti merjenju razdalj:
  - med učnim primerom in gručo
  - med dvema gručama
- kot razdaljo v teh primerih lahko upoštevamo:



- razdaljo med **najbližjima** primeroma (enojna povezanost, angl. *single linkage*)
  $d(C_1, C_2) = \min_{i,j}\{d_{ij} | i \in C_1, j \in C_2\}$
- razdaljo med **najbolj oddaljenima** primeroma (popolna povezanost, angl. *complete linkage*)
  $d(C_1, C_2) = \max_{i,j}\{d_{ij} | i \in C_1, j \in C_2\}$
- **povprečno razdaljo** med vsemi primeri (povprečna povezanost, angl. *average linkage*)
  $d(C_1, C_2) = \sum_{i \in C_1, j \in C_2} \frac{d_{ij}}{|C_1||C_2|}$

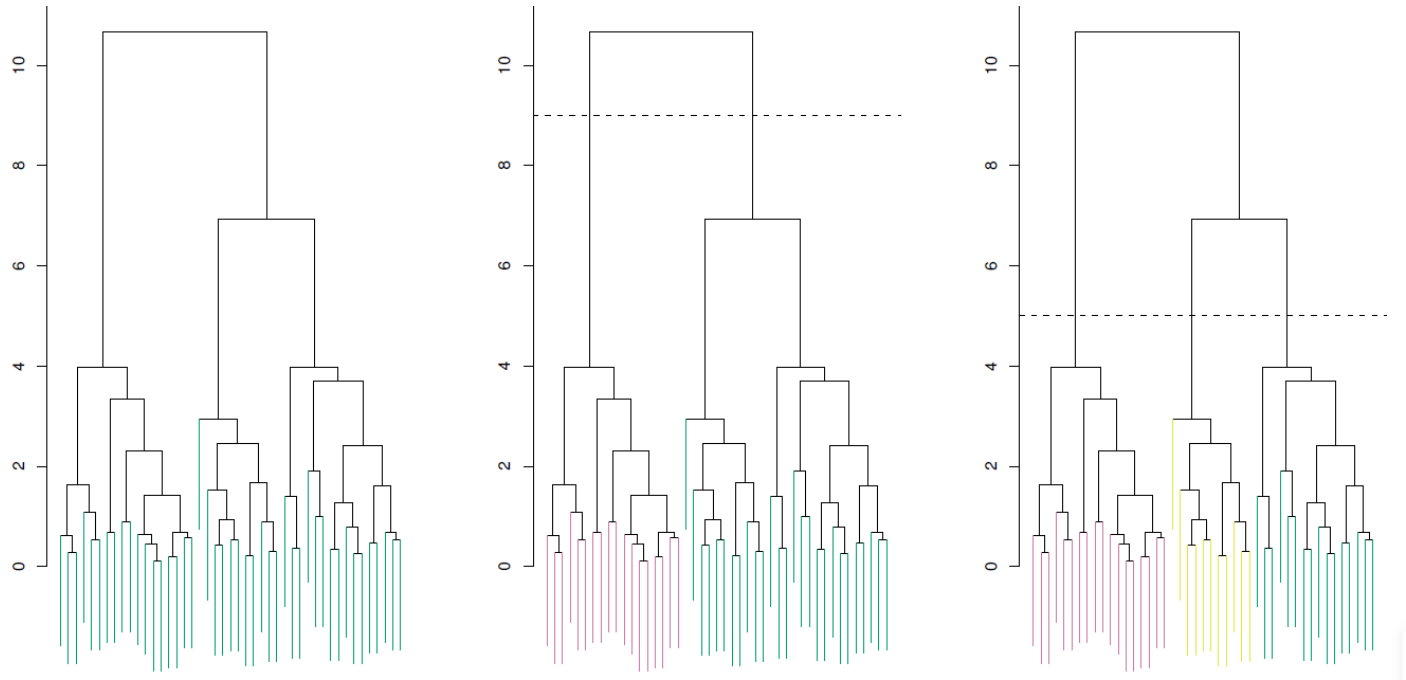
Primer

- 45 primerov v dvorazsežnem prostoru, ki v realnosti pripadajo trem različnim razredom (razredi so prikazani z barvo). Dejanske razrede lahko skrijemo pred algoritmom za gručenje in poskusimo z gručenjem odkriti gruče, ki ustrezajo razredom.



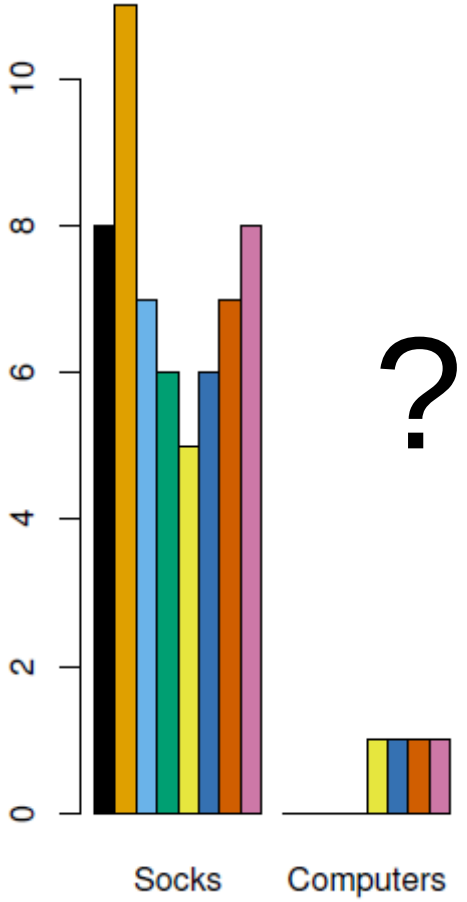
Primer

- dendrogram, pridobljen z evklidsko razdaljo in pristopom merjenja razdalje med gručami s popolno poveznostjo (complete linkage),
- prikaz rezanja dedrograma na različnih višinah



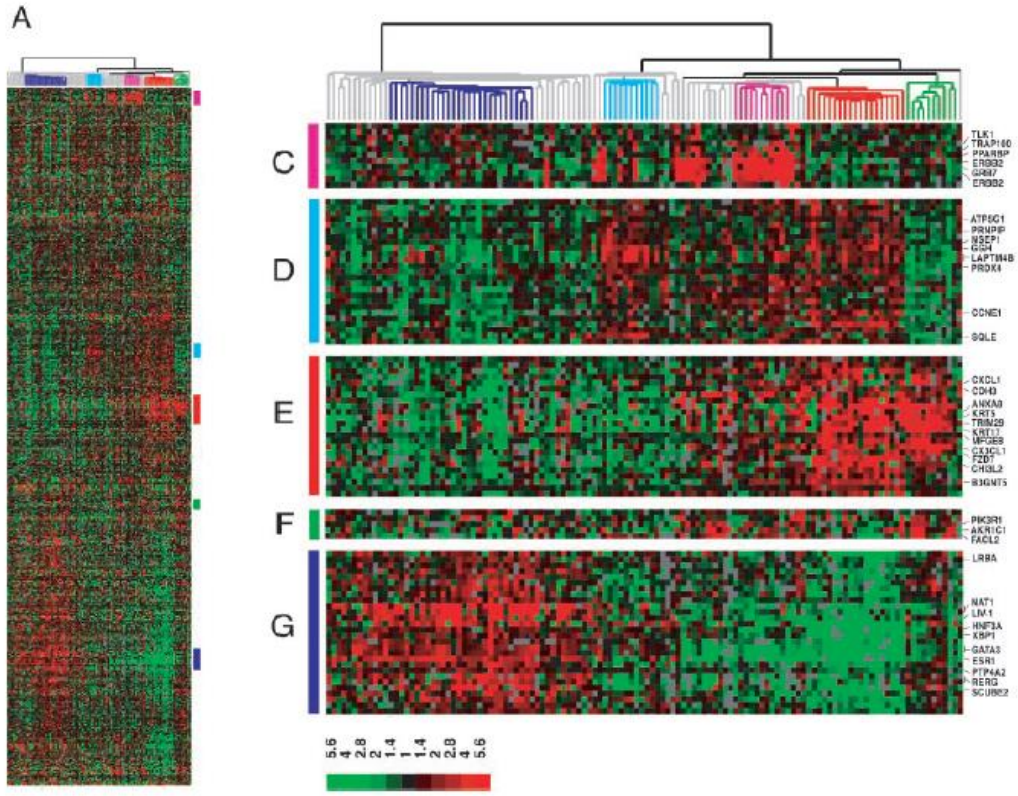
Opombe

- časovna zahtevnost:
  - združevalni pristop: $O(n^2 logn)$:
    n^2 časa za izračun matrike razdalj, $logn$ za urejanje razdalj
  - delilni pristop: $O(2^n)$:
    za iskanje optimalne delitve na dve podgruči
- dileme
  - katero mero razdalje izbrati?
  - kateri pristop merjenja razdalj med gručami izbrati?
  - kolikšno naj bo ciljno število gruč?
- normalizacija atributov?

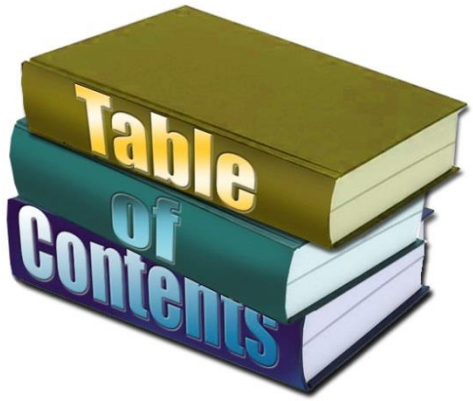


Primer uporabe

- analiza različnih vrst tumorja na prsih glede na izraženost genov
  - Sørlie, Therese, et al. "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications." Proceedings of the National Academy of Sciences 98.19 (2001): 10869-10874.
- mera razdalje: korelacija
- razdalja med gručami: povprečna razdalja med primeri
- atributi: izraženost 500 genov
- rezultati:
  - identifikacija sorodnih skupin pacientov
  - identifikacija izraženih genov v skupinah pacientov



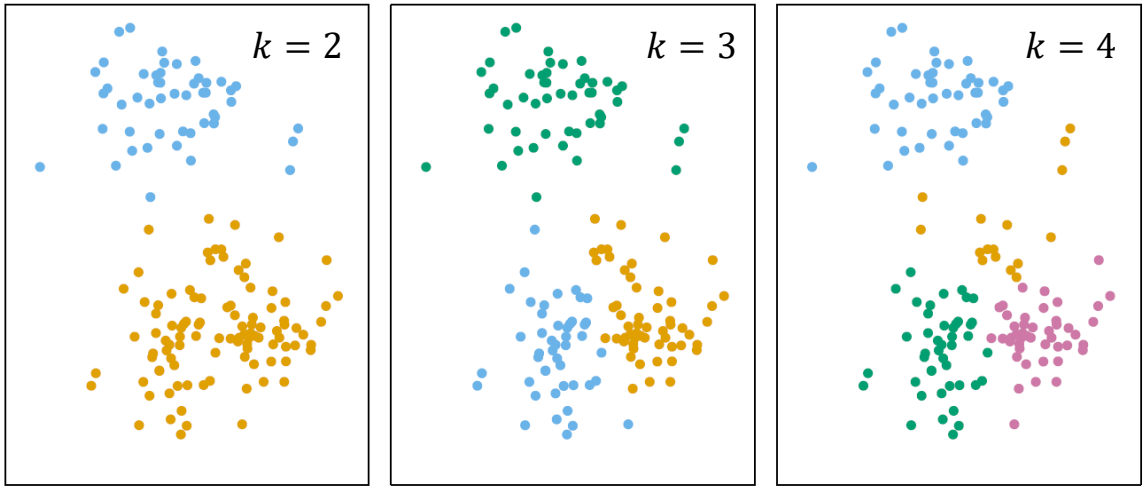
Pregled



- strojno učenje
  - primer aplikacije nadzorovanega učenja
  - nenadzorovano učenje
    - hierarhično gručenje
    - metoda k voditeljev

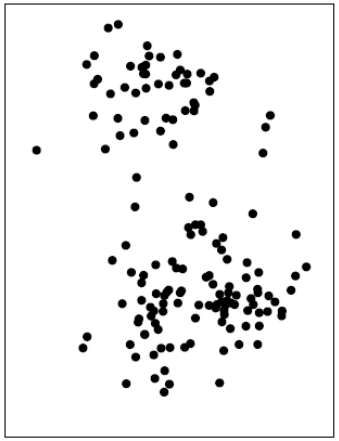
Metoda voditeljev

- angl. *k-means clustering*
- postopek:
  - izberi začetno število gruč k in naključno priredi vsak učni primer eni od gruč
  - ponavljaj dokler ni sprememb:
    - za vsako izmed k gruč izračunaj **centroid** (točka, določena s srednjo vrednostjo atributov vseh primerov, ki pripadajo gruči)
    - spremeni pripadnost vsakega primera tisti gruči, katere centroid je najbližji (glede na izbrano mero razdalje)
- primer: različne rešitve za $k = 2, k = 3$ in $k = 4$

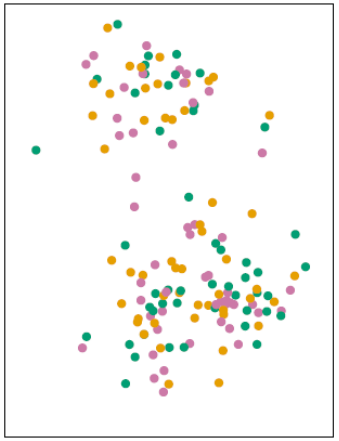


Primer izvajanja

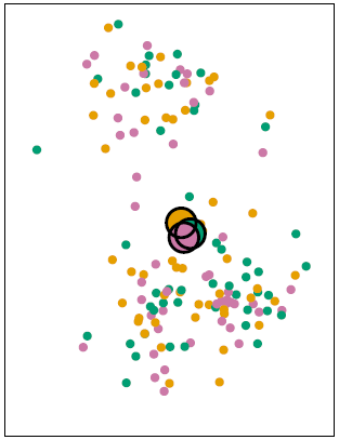
učna množica



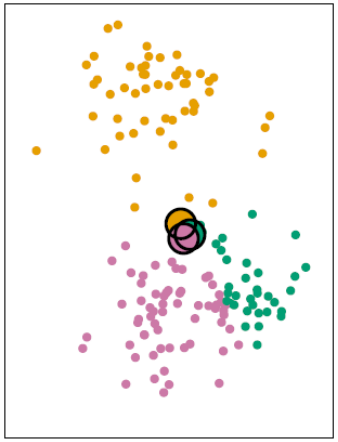
naključna določitev gručam



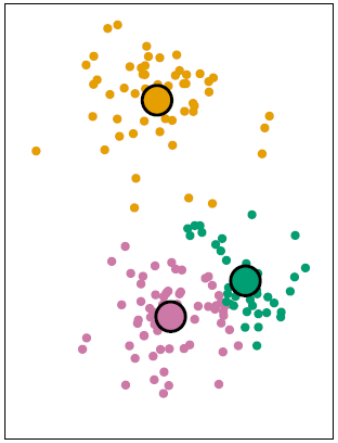
izračun začetnih centroidov gruč



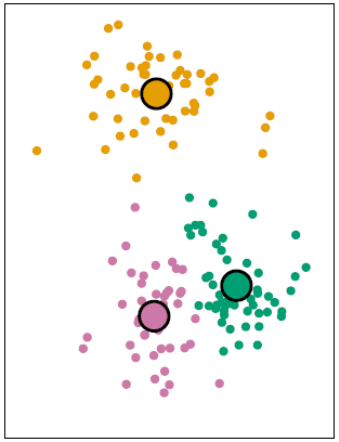
določitev pripadnosti primerov najbližjemu centroidu



ponovni izračun centroidov gruč

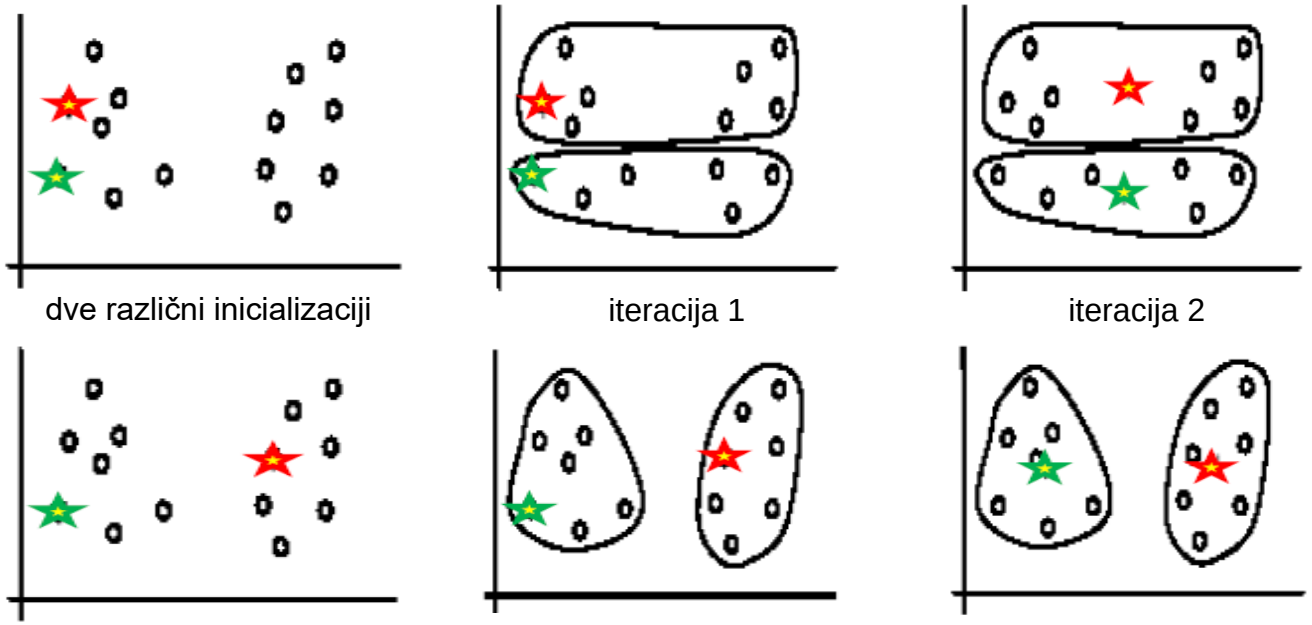


rezultat po 10 iteracijah



Lastnosti algoritma

- metoda na vsakem koraku znižuje varianco znotraj gruč (s prerazporejanjem pripadnosti primerov najbližji gruči)
- algoritem ne najde globalnega optimuma (glede na cilj minimizacije variance znotraj gruč), rešitev je odvisna od začetne inicializacije
- izračun centroidov je potrebno prilagoditi, če so atributi diskretni
- algoritem je občutljiv na šum (angl. *outliers*)
- težka objektivna evalvacija, vendar uporabno v praksi

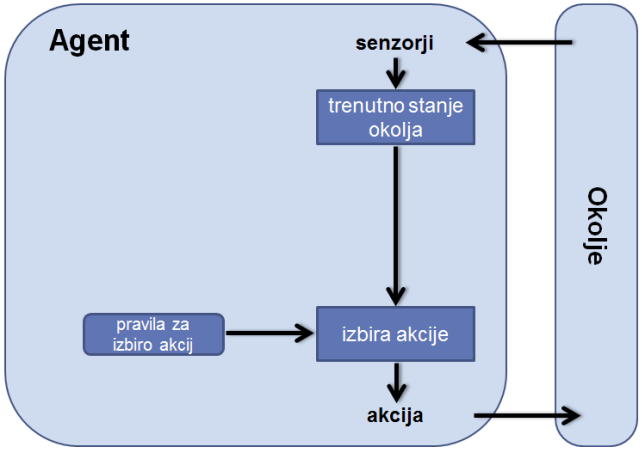


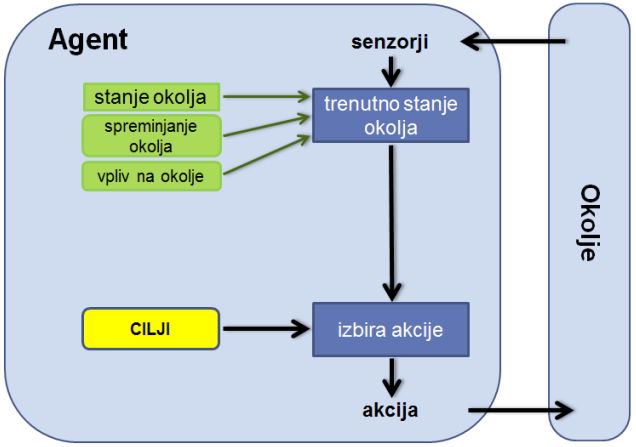
OSNOVE UMETNE INTELIGENCE

preiskovanje prostora stanj

Uvod

- *refleksni* agenti : *ciljno-usmerjeni* agenti
  - ciljno-usmerjeni: upoštevajo zaželenost bodočih akcij za doseganje cilja

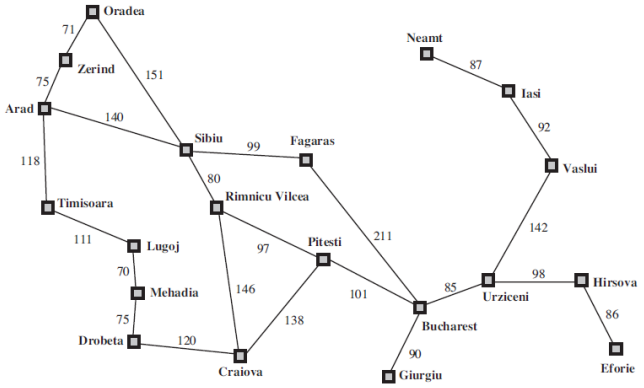


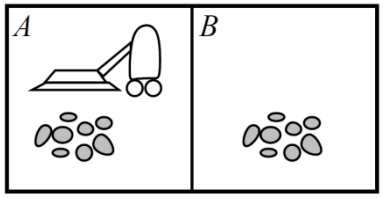


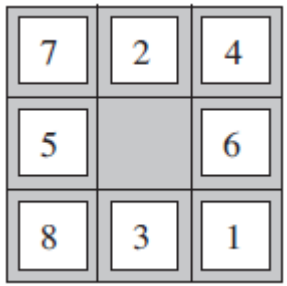
- problem:
  - abstrakcija problema
  - simbolična predstavitev (graf, drevo)
  - uporaba algoritma na simbolični predstavitvi za iskanje poti do cilja

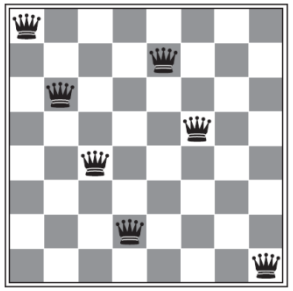
Primeri problemov

- avtomatski sesalec
- iskanje najkrajše poti
- igra 8 ploščic
- 8 kraljic na šahovnici
- planiranje v svetu kock
- manevriranje robotov
- trgovski potnik









Definicija problema

Za formalni opis problema potrebujemo:

- **začetno stanje**
- opis vseh možnih **akcij**, ki so na razpolago v posameznih stanjih
- **prehodno funkcijo**, ki definira naslednika stanja

$$rezultat(trenutno\_stanje, akcija) = novo\_stanje$$

- **ciljni** predikat:
$cilj(stanje) = \{true, false\}$
- **cenilna funkcija**, ki vsaki poti določi ceno (numerično vrednost)

$$c(stanje, akcija, stanje') \geq 0$$

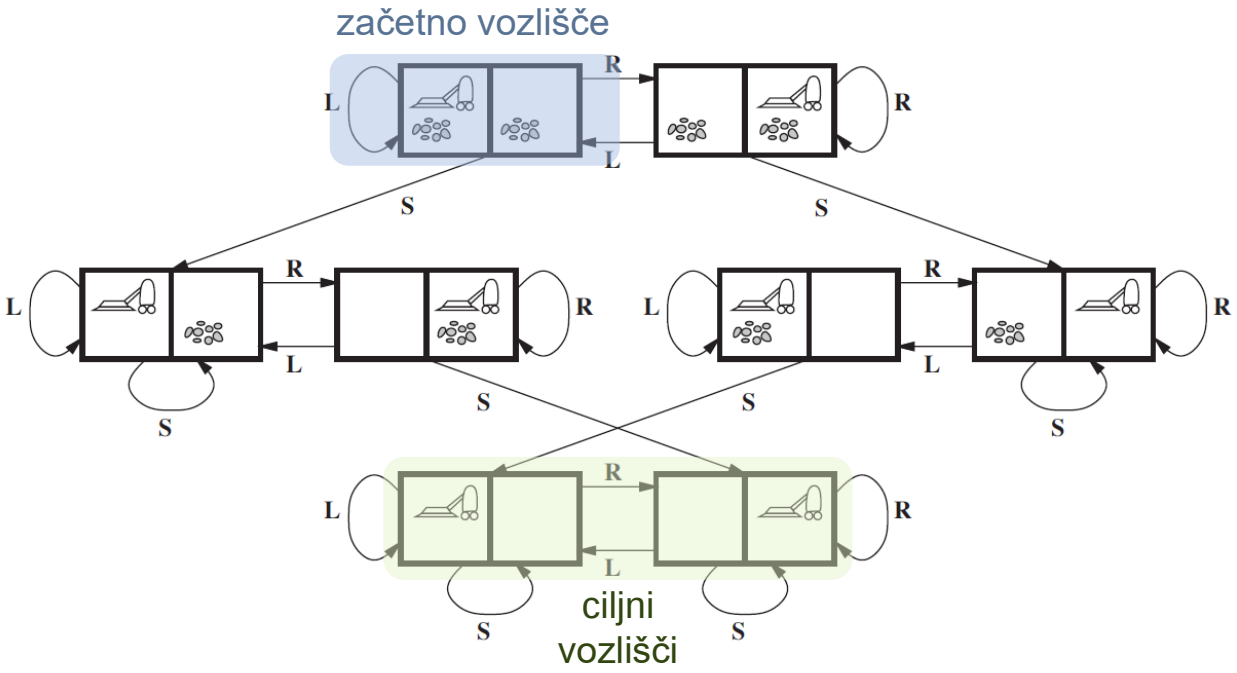
Definicija problema

- opisani način podajanja problema omogoča, da problem predstavimo z usmerjenim grafom, ki ponazarja ***prostor stanj***:
  - **vozlišča**: stanja (problemske situacije)
  - **povezave**: akcije (dovoljene poteze)
  - **pot** v grafu: zaporedje stanj, ki ga povezuje zaporedje akcij
  - eno **začetno** vozlišče
  - eno ali več **ciljnih** vozlišč

- reševanje problema – **iskanje poti v grafu s preiskovanjem**

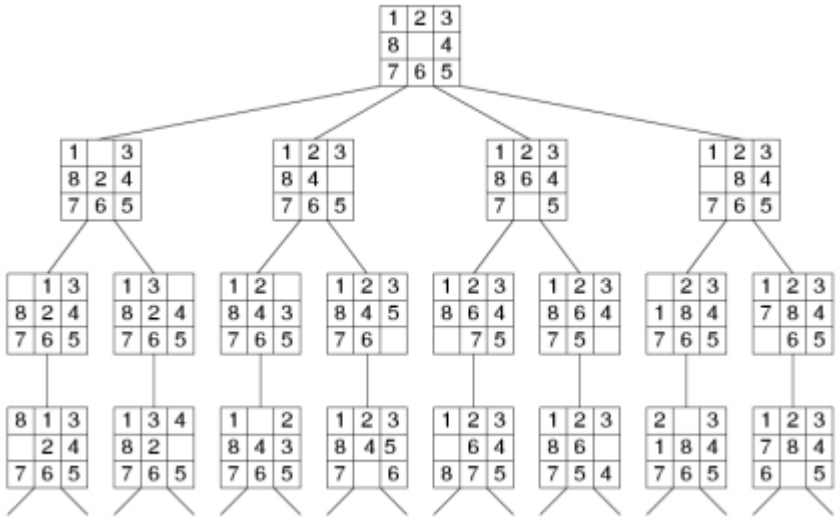
- rešitev problema – zaporedje akcij (pot), ki vodi od začetnega do ciljnega vozlišča
  - optimalna rešitev problema izmed vseh možnih ima najnižjo ceno poti

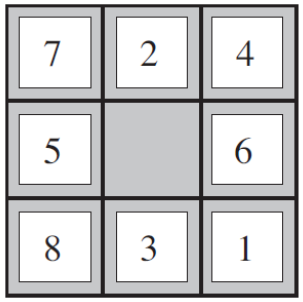
Primer: sesalec



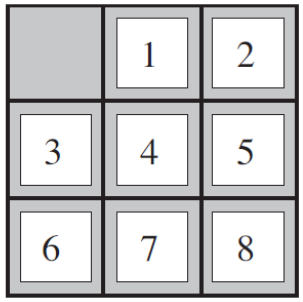
- stanja – kombinacija lokacije robota in umazanije v prostorih A in B ($2 \times 2^2 = 8$ stanj)
- akcije – levo, desno, sesaj (in počakaj)
- cilj – oba prostora čista
- cena akcije – npr. 1 za vsako potezo

Primer: igra 8 ploščic





začetno stanje

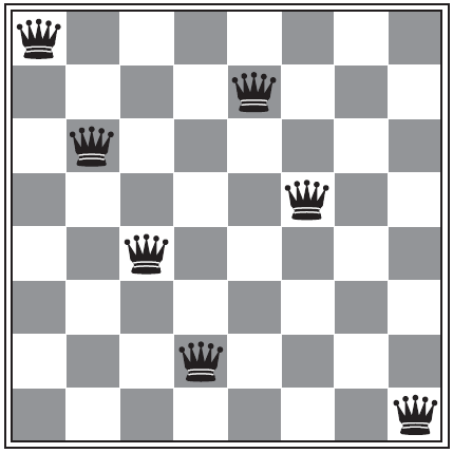


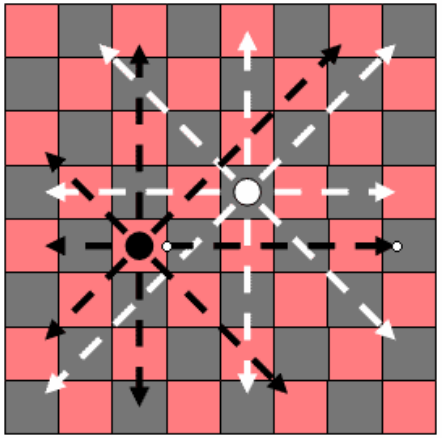
končno stanje

- stanja – lokacije ploščic
- akcije – premiki "prazne" ploščice levo, desno, gor, dol
- cilj – podano ciljno stanje
- cena akcije – 1 za vsako potezo

- problem je NP-poln
- igra 3x3 ima 9!/2 =181.440 dosegljivih stanj
- igra 4x4 ima okoli $1{,}3 \times 10^{12}$ dosegljivih stanj
- igra 5x5 ima okoli 10^{25} dosegljivih stanj

Primer: problem 8 kraljic





- stanja – postavitve 0-8 kraljic na plošči ($64 \times 63 \times \cdots \times 57 = 1.8 \times 10^{14}$) stanj
- začetno stanje – prazna igralna plošča
- akcije – postavitve nove kraljice na prazno polje
- cilj – 8 kraljic na plošči, nobeni dve se ne napadata

bolj optimalna formulacija problema:

- stanja – postavitve 0-8 kraljic od leve proti desni, v vsak stolpec ena (2057 stanj)
- akcije – postavitve nove kraljice na prazno polje v prvi skrajno levi prosti stolpec

Primer: Knuthov neskončni prostor

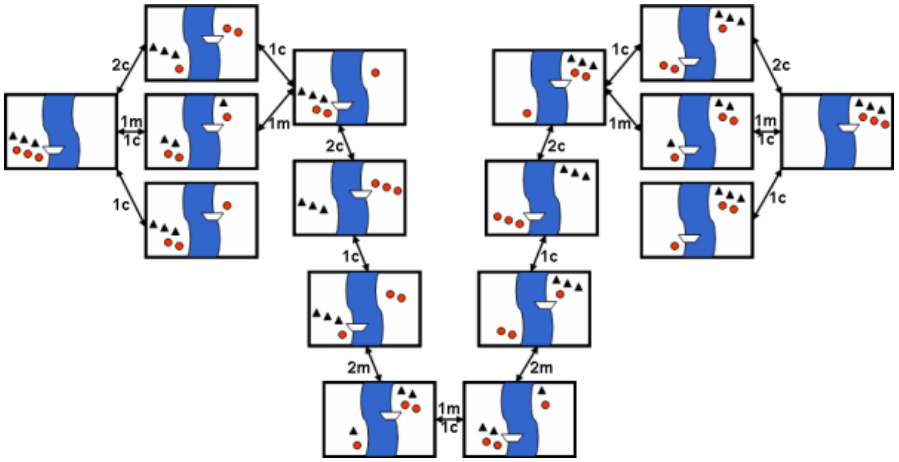
- Knuthova predpostavka: začenši s številom 4 in zaporedjem treh operacij (faktoriela, kvadratni koren, zaokroževanje navzdol), lahko dosežemo poljubno pozitivno celo število. Npr. število 5 lahko zapišemo kot:

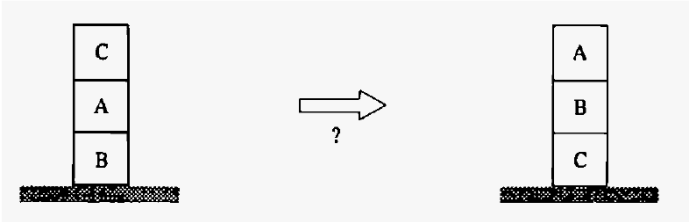
$$5 = \left\lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right\rfloor$$

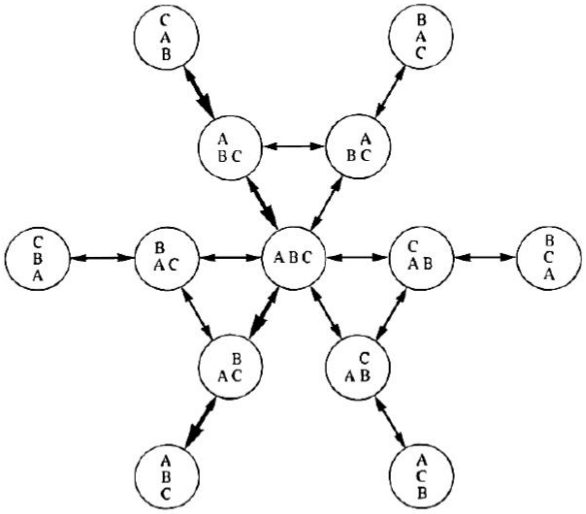
- začetno stanje – število 4
- akcije – uporaba ene izmed operacij *{faktoriela, kvadratni koren, zaokroževanje navzdol}*
- stanja – pozitivna števila
- ciljno stanje – želeno pozitivno število

Primeri drugih umetnih problemov

- misionarji in ljudožerci
- preurejanje postavitve kock

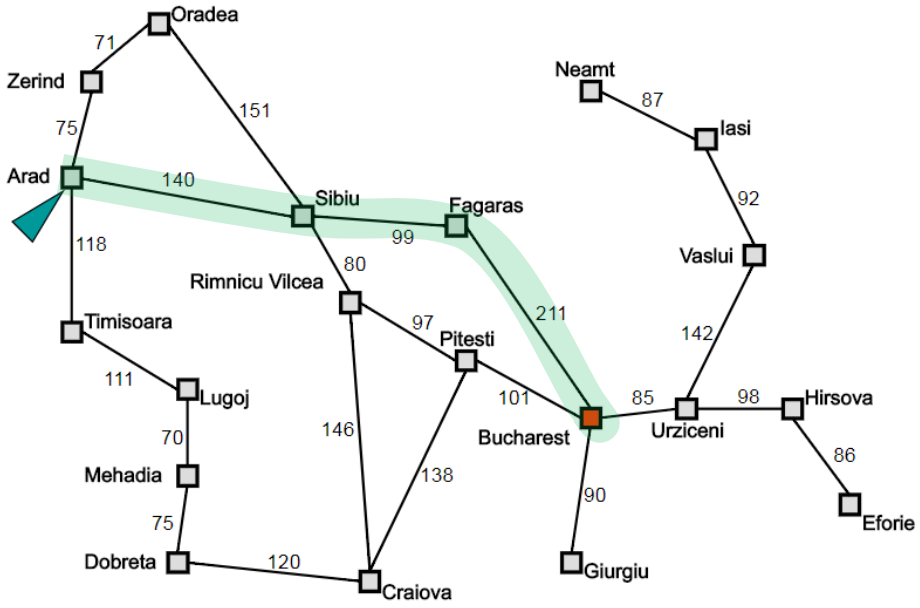






Primeri realnih problemov

- iskanje najkrajše poti
- primer obhoda ("obišči vsa mesta vsaj enkrat, začenši v mestu X")
- problem potujočega trgovca - TSP ("najdi najkrajšo pot za obhod vseh mest natanko enkrat, začenši v mestu X")
- postavitev komponent na vezju (minimizacija površine, dolžine povezav itd.)
- navigacija robotov v prostoru
- sestavljanje izdelkov na proizvodni liniji



Preiskovanje

- problem: velika kombinatorična eksplozija možnih stanj
- preiskovalni algoritmi:
  - **neinformirani**: razpolagajo samo z definicijo problema
    - iskanje v širino (*breadth-first search*)
    - iskanje v globino (*depth-first search*)
    - iterativno poglabljanje (*iterative deepening*)
  - **informirani**: razpolagajo tudi z dodatno informacijo (domensko znanje, hevristične ocene), kako bolj učinkovito najti rešitev
    - algoritem A*
    - algoritem IDA*
    - prioritetno preiskovanje (*best-first search*)
    - algoritem RBFS (*recursive best-first search*)
    - plezanje na hrib (*hill climbing*)
    - iskanje v snopu (*beam search*)
    - …


Neinformirani algoritmi