UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Perc

Implementacija in eksperimentalna analiza tehnike razvrščanja podatkov s konsenzom

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Blaž Zupan

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01706/2010
Datum: 01.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DOMEN PERC**

Naslov: **IMPLEMENTACIJA IN EKSPERIMENTALNA ANALIZA TEHNIKE RAZVRŠČANJA PODATKOV S KONSENZOM**

IMPLEMENTATION AND EXPERIMENTAL ANALYSIS OF CONSENSUS CLUSTERING

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V diplomskem delu preglejte literaturo na področju razvrščanja podatkov s konsenzom in to tehniko implementirajte v okolju za odkrivanje znanj iz podatkov Orange. Implementirano tehniko nato eksperimenentalno primerjajte s klasičnimi metodami razvrščanja. Uporabite podatke, kjer so razredi primerov že znani in opazujte, kako uspešne so tehnike razvrščanja pri odkrivanju teh razredov. Pri eksperimentih ocenite točnost algoritmov in njihovo stabilnost.

Mentor:

prof. dr. Blaž Zupan

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI 2

Dekan:

prof. dr. Nikolaj Zimic

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Domen Perc,

z vpisno številko 63040124,

sem avtor/-ica diplomskega dela z naslovom:

Implementacija in eksperimentalna analiza tehnike razvrščanja podatkov s konsenzom

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom prof. dr. Blaža Zupana;
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela;
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 23.1.2011 Podpis avtorja/-ice:

Zahvala

Zahvaljujem se mentorju prof. dr. Blažu Zupanu za pomoč pri nastajanju diplomskega dela. Prav tako se zahvaljujem vsem, ki so mi kakor koli pripomogli priti do tega cilja. Sami najbolje veste, kdo ste.

Kazalo

Povzetek **1**

Abstract **2**

1 Uvod **3**

2 Tehnike razvrščanja podatkov **5**
2.1 Razvrščanje z voditelji 5
2.2 Hierarhično razvrščanje 8

3 Razvrščanje s konsenzom **12**
3.1 Problem zaupanja v rezultat razvrščanja 12
3.2 Algoritem 12
3.3 Merjenje konsenza 14
3.4 Vizualizacija matrike konsenzov 15
3.5 Urejanje matrike konsenzov 16
3.6 Določanje števila skupin 17
3.6.1 Porazdelitev konsenza 18
3.7 Algoritmi vzorčenja 19

4 Implementacija in uporaba algoritma **21**
4.1 Implementacija 21
4.2 Primer uporabe 23

5 Eksperimentalno vrednotenje **25**
5.1 Tehnika vrednotenja 25
5.1.1 Prilagojeni Rand indeks 26
5.2 Testiranje 27
5.2.1 Način testiranja 27
5.2.2 Rezultati testiranja pri uporabi razvrščanja z voditelji . . 28

5.2.3 Rezultati testiranja hierarhičnega razvrščanja 33
5.2.4 Povzetek testiranj 36
5.2.5 Vpliv števila ponovitev 36
5.2.6 Vpliv velikosti vzorca 37

6 Zaključek **38**

Literatura **39**

A Testni primer 1 **40**

B Testni primer 2 **42**

C Rezultati testiranja **43**

Povzetek

Razvrščanje s konsenzom je tehnika s področja strojnega učenja, ki se uporablja za iskanje in potrjevanje skupin v podatkih. Za razvrščanje podatkov v skupine uporablja večkratno razvrščanje različnih metod nad različnimi vzorci, pridobljenimi z razporejanjem vhodnih podatkov v manjše podmnožice. Pomembni lastnosti te metode sta, da je možno informacije, pridobljene z večkratnim vzorčenjem in razvrščanjem, uporabiti za določanje skupin in tudi za grafičen prikaz stabilnosti in pripadnosti k skupinam. Možnost grafičnega prikaza igra pomembno vlogo pri lažjem razumevanju rezultatov razvrščanja. V pričujočem diplomskem delu smo skušali preveriti uporabnost razvrščanja s konsenzom ter njegove prednosti pred klasičnimi tehnikami nenadzorovanega razvrščanja. Tehniko smo tudi implementirali v programskem jeziku Python s pomočjo odprtokodnega okolja za odkrivanje znanj iz podatkov in strojno učenje Orange. Implementacijo smo ovrednotili s testnimi podatki, dobljenimi iz repozitorija za strojno učenje fakultete Irvine iz Kalifornije. Rezultati eksperimentov so pokazali, da lahko tehnika s konsenzom izboljša rezultate klasičnih metod razvrščanja, predvsem v smislu stabilnosti rezultatov. Do velikih razlik pri rezultatih ni prihajalo, če imajo klasične razvrščevalne tehnike težave pri razvrščanju podatkovne množice, tudi tehnika s konsenzom ne bo razvrščala bistveno bolje.

Ključne besede:

nenadzorovano učenje, razvrščanje s konsenzom, razvrščanje z voditelji, hierarhično razvrščanje, prilagojeni Rand indeks, vzorčenje.

Abstract

Consensus clustering is a machine learning tehnique for class discovery and clustering validation. The method uses various clustering algorithms in conjunction with different resampling tehniques for data clustering. It is based on multiple runs of clustering and sampling algorithm. Data gathered in these runs is used for clustering and for visual representation of clustering. Visual representation helps us to understand clustering results. In this thesis we compare consensus clustering with standard clustering algorithms to find advances of using this tehnique. We have implemented consensus clustering in programming language Python using open-source data mining and machine learning suite Orange. We tested the implementation with data sets from machine learning repository of University of California, Irvine. Experiment results showed some improvements in comparison with standard tehniques, especially in terms of clustering consistency. Changes in overall performance were rather smaller. If standard tehniques have clustering problems on specific data set, also consesus clustering won't be much better.

Key words:

unsupervised learning, consensus clustering, KMeans, hierahical clustering, adjusted Rand index, sampling.

Poglavje 1

Uvod

Raziskovanje novih taksonomij - klasifikacij objektov glede na njihove lastnosti je v zadnjem času deležno precejšnje pozornosti v statistiki in strojnem učenju, zato je vsaka uspešna nova metoda dobrodošla. Nenadzorovano učenje je metoda strojnega učenja, ki temelji na podatkih, kjer so podane samo značilke vzorcev, niso pa podani razredi, katerim pripadajo. Naloga učnega algoritma je določiti te razrede. Gre za iskanje neprekrivajočih podmnožic objektov neke problemske domene. Razvrščanje se rahlo razlikuje od klasifikacije; pri klasifikaciji so podani razredi in iščemo, kateremu pripada določen objekt. Poleg besede razvrščanje se uporabljajo še sinonimi: rojenje, grupiranje ali grozdenje (angl. *clustering*).

Razvrščanje se uporablja pri analizi naravnih in tehnoloških procesov, analizi ekonomskih trendov, preverjanju konsistentnosti in odvisnosti podatkov itd. Nenadzorovano učenje se veliko uporablja tudi v zdravstvu, zlasti je pomembno na področju genetike za zmanjšanje široko definiranih bioloških razredov. Med drugim predstavlja potencial, da veliko doprinese k diagnostiki, prognostiki in zdravljenju raka.

Število iskanih razredov je lahko podano vnaprej kot predznanje ali pa mora primerno število razredov določiti učni algoritem. Naloga učnega algoritma je določiti relativno majhno število koherentnih razredov - skupin primerov, ki imajo čim več skupnih lastnosti. Podobnost med primeri je odvisna od izbrane metrike in je odločilnega pomena za rezultat razvrščanja.

Temeljna vprašanja, s katerimi se srečamo pri razvrščanju podatkov, so:

- kako določiti število skupin,
- kako izmeriti zaupanje v število skupin ter razvrstitev elementov v skupine.

Stabilnost generiranih skupin lahko povečamo z uporabo vzorčenja. Z naključnim izbiranjem vzorcev iz vhodnih podatkov povečamo stabilnost in zanesljivost rezultatov, ker zmanjšamo vpliv motenj in prevelikega prilagajanja določenim oz. izbranim učnim podatkom.

Prvi sklop diplomske naloge obravnava dva osnovna in najbolj razširjena razvrščevalna algoritma, ki sta razvrščanje z voditelji in hierarhično razvrščanje. Spoznali bomo osnovno idejo njunega delovanja ter njune slabosti in prednosti. V nadaljevanju se bomo natančno posvetili razvrščanju s konsenzom. Pogledali si bomo težave, na katere naletimo pri razvrščanju, in kaj je vodilo avtorje k razvoju te tehnike razvrščanja. Podrobno bomo pogledali delovanje ter zmožnosti algoritma. Predstavili bomo možne metode za realizacijo posameznih sklopov. Predstavitvi sledi poglavje o implementaciji. V tem poglavju pojasnjujemo, kako smo implementirali tehniko, in podajamo primer uporabe. V zadnjem delu diplomske naloge navajamo, komentiramo in vrednotimo pridobljene rezultate testiranja in primerjanja s klasičnima tehnikama razvrščanja, ki smo ju predstavili v prvem sklopu.

Poglavje 2

Tehnike razvrščanja podatkov

V diplomski nalogi smo tehniko razvrščanja s konsenzom uporabili z dvema različnima metodama nenadzorovanega učenja, ki ju podrobneje opisujemo v tem razdelku.

2.1 Razvrščanje z voditelji

Razvrščanje z voditelji[1] [2] (angl. *KMeans clustering*) je eden izmed najstarejših in najbolj razširjenih razvrščevalnih algoritmov. Je metoda nenadzorovanega strojnega učenja, ki vhodne podatke razvrsti v k skupin. Vsak element se uvrsti v skupino, katere vrednosti atributov elementov imajo najbližjo vrednost. Obstaja več variacij glede na to, od česa in kako merimo razdaljo skupine. Skupino najpogosteje predstavimo tako, da izračunamo povprečne vrednosti atributov elementov, ki so v njej. V nadaljevanju si bomo podrobneje pogledali, kako poteka razvrščanje s to tehniko.

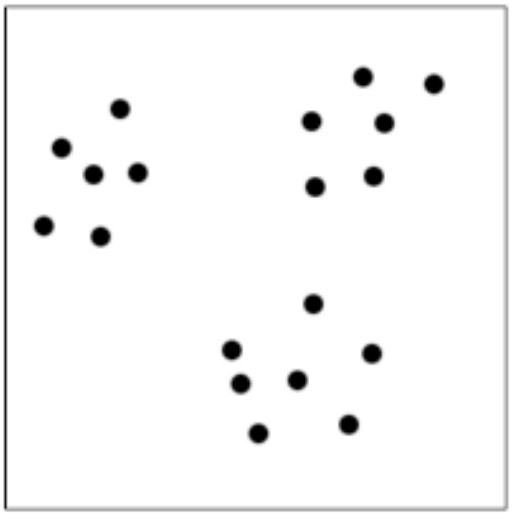
Z x označimo vhodno množico elementov, $x = (x_1, x_2...x_n)$. Naj bodo ti razporejeni, kot je prikazano na sliki (2.1). Želimo jih razvrstiti v množico skupin S, sestavljeno iz k podmnožic, $S = (S_1, S_2...S_k)$. Ko so elementi enkrat porazdeljeni v skupine, moramo oceniti, kako dobro smo jih razvrstili. Mera, ki jo lahko uporabimo, je vsota kvadratne napake (angl. *Sum of Squared Error - SSE*). Vrednost vsote kvadratne napake dobimo tako, da za vsak element izračunamo oddaljenost od najbližjega centroida. Naš cilj je, da zmanjšamo SSE znotraj skupine na minimum. Matematično lahko ta cilj

[1] http://en.wikipedia.org/wiki/K-means_clustering

zapišemo z enačbo:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{2.1}$$

S μ_i smo označili centroid skupine S_i.



Slika 2.1: Prikaz vhodnih podatkov z razsevnim diagramom množice.

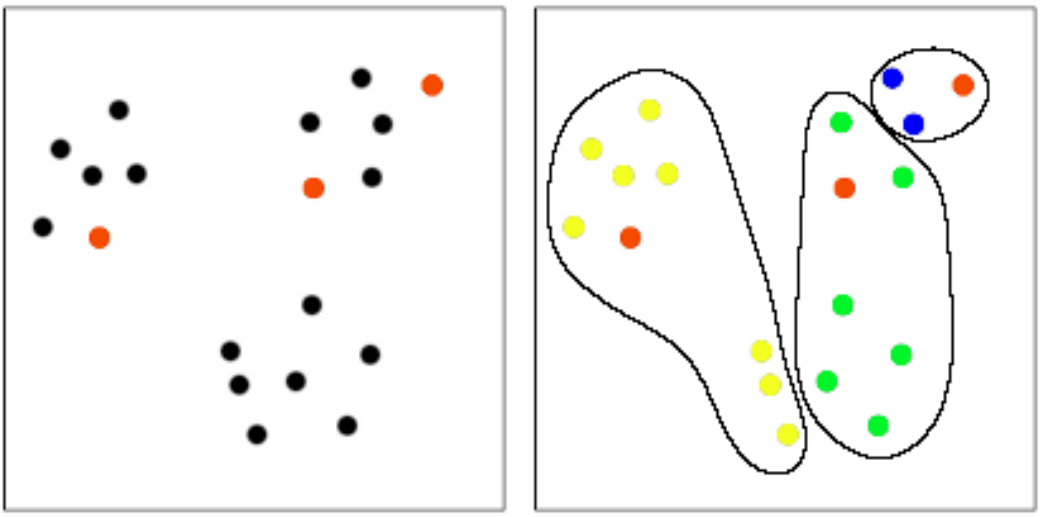
Algoritem razvrščanja z voditelji je iterativen. Razbijemo ga lahko na dva koraka:

1. Najprej naključno izberemo k elementov vhodne množice, ki bodo predstavljali začetne centroide k skupin. Označimo jih z m_k. Vsem preostalim elementom nato izračunamo razdaljo do vseh k izbranih predstavnikov skupin. Uvrstimo jih v skupine, katerih predstavnik jim je najbliže. 1. korak predstavljata enačba (2.2) in slika (2.2):

$$S_i = \{x_j : \|x_j - m_i\| \leq \|x_j - m_{i^*}\| \; za \; vsak \; i^* = 1...k\} \tag{2.2}$$

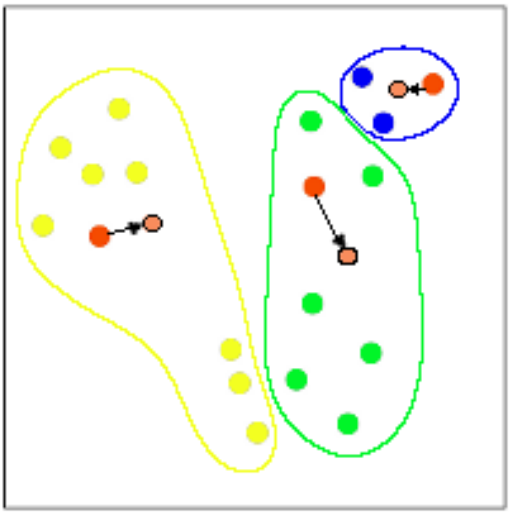
2. Po končanem prvem koraku, ko so elementi dodeljeni v primerne skupine, na novo izračunamo centroide skupin. Izračun novih centroidov izvedemo, kot prikazuje enačba:

$$m_i = \frac{1}{\|S_i\|} \sum_{x_j \in S_i} x_j \tag{2.3}$$



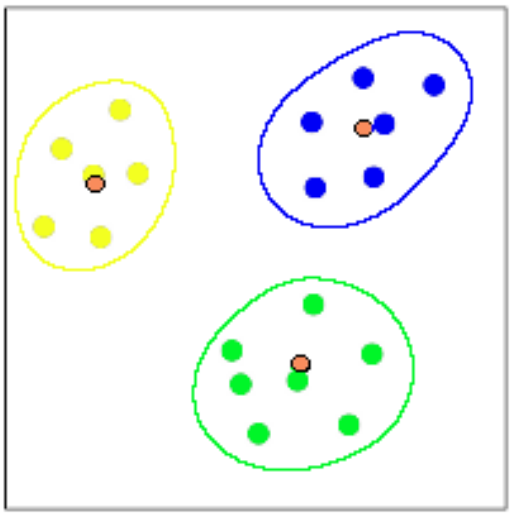
Slika 2.2: Slika na levi prikazuje izbor centroidov, slika na desni pa uvrstitev elementov v primerno skupino.

Vsaki skupini izračunamo novo povprečje. Slika (2.3) prikazuje, kako bi se pri danem primeru v tem koraku pri eni iteraciji prestavili centroidi. Po končanem drugem koraku, ko imamo izračunane nove centroide, se vrnemo k 1. koraku in ponovno generiramo skupine.



Slika 2.3: Premik centroidov v 2. koraku.

Koraka ponavljamo, dokler niso premiki centroidov dovolj ustaljeni. Po nekaj ponovitvah teh dveh korakov dobimo stanje, prikazano na sliki (2.4). Dobili smo tri pričakovane skupine.



Slika 2.4: Prikaz dobljenih skupin po končanem razvrščanju.

Razvrščanje z voditelji spada v skupino iterativnih metod. Prednost iterativnih metod pred hierarhičnimi je, da enolično določijo skupine. Ni nam potrebno določiti števila skupin, ko so podatki že razvrščeni. Obenem je to lahko pomanjkljivost, število skupin moramo izbrati vnaprej. Poleg tega nimajo intuitivnosti in grafične upodobite hierarhičnega razvrščanja. Iterativne metode razvrščanja, ki temeljijo na požrešnem iskanju, pesti še ena pomanjkljivost, ki je lahko zelo pomembna. Občutljive so na, v začetku izbrane vrednosti. Na rešitev lahko namreč vpliva začetna izbira centroidov, ki je po navadi naključna. Posledično nam nič ne zagotavlja, da bo algoritem zares konvergiral h globalnemu optimumu. Zato je rezultatom težko zaupati. Če algoritem večkrat poženemo nad isto množico elementov in z enakimi parametri, lahko dobimo različne, nekonsistente rezultate.

2.2 Hierarhično razvrščanje

Pri hierarhičnem razvrščanju[1] [2] (angl. *hierahical clustering*) se uporabljata dva pristopa, tj. od zgoraj navzdol in od spodaj navzgor. Pogostejši je sle-

[1] http://www.ics.uci.edu/ eppstein/280/tree.html

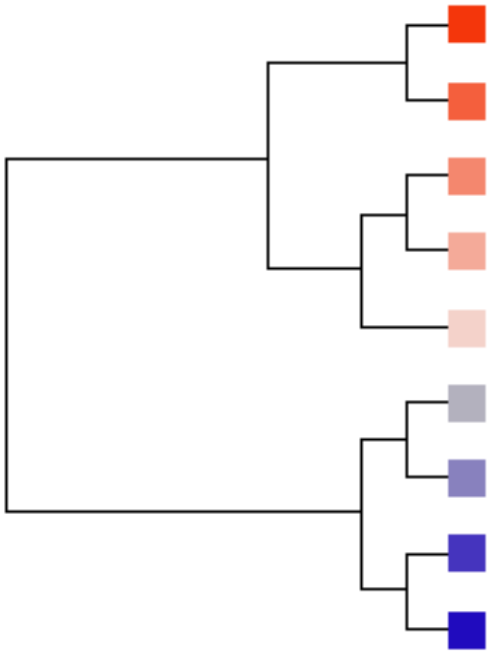
dnji. Na začetku vsak element predstavlja svojo skupino. Algoritem iterativno združuje najbolj podobne skupine med seboj, dokler ne ostaneta le še dve. Pri pristopu od zgoraj navzdol imamo na začetku eno skupino, ki jo nato razcepljamo, dokler ne pridemo do posameznih elementov. Pristop od spodaj navzgor poteka po naslednjih korakih:

- vsak element postavi v svojo skupino,
- za vsak par skupin izračunaj razdaljo med njima,
- sestavi matriko razdalj,
- poišči par skupin z najmanjšo razdaljo,
- par odstrani iz matrike in ju združi,
- ponovno izračunaj razdaljo nove skupine do vseh ostalih in popravi matriko,
- ponavljaj, dokler ne prideš do samo ene skupine.

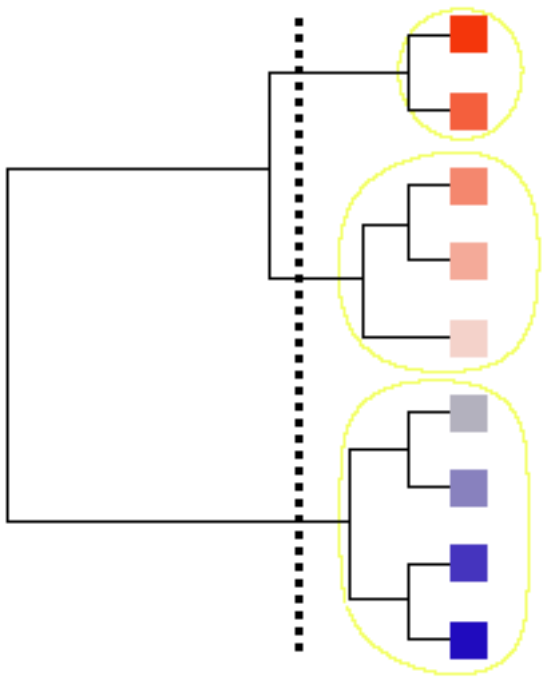
Rezultat hierarhičnega razvrščanja lahko lepo upodobimo kot drevo, imenovano dendrogram[1]. Dendrogram je binarno drevo, pri katerem so vsi listi na istem nivoju. Vse elemente ima shranjene v listih. Višina notranjih vozlišč nam pove, kakšna je razdalja med skupinami, manjša kot je višina, manjša je razdalja. Slika (2.5) prikazuje dendrogram. Slednjega smo dobili z izrisom hierarhičnega razvrščanja vhodne množice elementov, ki so se razlikovali po barvi. So različnih odtenkov rdeče in modre barve. Različni odtenki iste barve so si bolj podobni kot dve povsem različni barvi, zato je višina vej med odtenki rdeče oziroma modre barve veliko nižja kot med modro in rdečo barvo. Število skupin lahko izberemo sami naknadno, ko so podatki že razvrščeni v skupine z rezanjem (angl. *pruning*). Ta lastnost je včasih nezaželena, ker lahko subjektivno vplivamo na razvrščanje. Na sliki (2.6) smo z rumeno barvo označili skupine, ki bi nastale, če bi želeli kot rezultat dobiti tri skupine.

Hierarhično razvrščanje je precej razširjeno zaradi intuitivnega delovanja in možnosti dobrega vizualnega prikaza rezultata. Ker ni usmerjeno v določeno število skupin, je zelo uporabno za raziskovalno analizo podatkov. Poleg problema subjektivnosti pri rezanju ima še eno slabo lastnost. Slaba - nepravilna razvrstitev elementa v začetku razvrščanja lahko močno vpliva na kasnejše razvrščanje, zaradi česar je potrebno dobro pregledati drevo, če je smiselno.

[1] http://www.ics.uci.edu/ eppstein/280/tree.html



Slika 2.5: Dendrogram hierarhičnega razvrščanja elementov glede na njihovo barvo.



Slika 2.6: Črtkana črta prikazuje razred drevesa oziroma vizualizacijo praga razdalje med skupinami, pri katerem smatramo, da so skupine tako različne, da tvorijo nove koncepte. Kot rezultat rezanja bi dobili tri skupine, obkrožene z rumeno.

Poglavje 3

Razvrščanje s konsenzom

3.1 Problem zaupanja v rezultat razvrščanja

Zelo pomemben problem analize razvrščanja je potrditev rezultatov. Problem, s katerim se srečamo, je, kako zagotoviti zaupanje v dobljen rezultat, tako v dobljeno število skupin kot tudi v razvrstitev elementov. Pri razvrščanju namreč nimamo zunanjega pričakovanega rezultata, kot je v primeru klasifikacije znani razred. Obstajajo statistične procedure za oceno in preverjanje smiselnosti rezultata razvrščanja, a so namenjene elementom z malo atributi, niso splošno namenske. Njihovo delovanje nad elementi z večjim številom atributov je zato vprašljivo [1].

Alternativen pristop validacije skupin je vzorčenje. Z različnimi algoritmi vzorčenja lahko simuliramo več različnih vhodnih podatkovnih množic z uporabo ene množice in tako povečamo stabilnost razvrščanja. Predpostavka, na kateri temelji ta pristop, je, da bolj kot so rezultati pri različno premešanih vhodnih podatkih stabilni, zanesljivejši je rezultat razvrščanja. Razvrščanje s konsenzom spada med metode, ki temeljijo na vzorčenju.

3.2 Algoritem

Glavna motivacija avtorjev metode razvrščanja s konsenzom je bila potreba po čim večji stabilnosti dobljenih rezultatov razvrščanja. Želeli so doseči čim večje zaupanje v rezultate. Rezultati morajo biti robustni na spremenljivost vzorcev. Osnovna predpostavka je preprosta: če vhodni podatki predstavljajo podmnožico množice podatkov A, se skupine in število le-teh ne smejo bistveno spremeniti za neko drugo, podobno podmnožico množice podatkov A.

Robustnejše kot so dobljene skupine na različne vzorce vhodnih podatkov, zanesljivejši so dobljeni rezultati.
Algoritem 1 predstavlja psevdo kodo razvrščanja s konsenzom. Na vhodu mu poleg množice podatkov D ki jih želimo razvrstiti, podamo še kateri razvrščevalni in vzorčevalni algoritem želimo uporabiti, število ponovitev vzorčenja in razvrščanja H, ter množico števil $\mathcal{K}$, ki določajo število generiranih skupin. V notranji zanki algoritem H-krat zgradi vzorec iz vhodnih podatkov in nad njim izvede razvrščanje. Kot rezultat razvrščanja dobimo povezovalno matriko. Ker se vse to izvaja v zanki, dobimo množico H povezovalnih matrik. Na podlagi dobljene množice izračunamo matriko konsenzov. S tem postopkom dobimo matriko konsenzov razvrščanja podatkov v neko določeno število skupin. Ker želimo podatke razvrstiti v različna števila skupin, se ta postopek nahaja v zanki, ki nam izračuna matrike konsenzov za vsa željena števila skupin. Ko se izvajanje konča, nam preostane le še, da pregledamo vse dobljene matrike konsenzov in ugotovimo pri katerem številu skupin je le-ta optimalna. Optimalno matriko lahko nato uporabimo kot matriko razdalj pri razvrščanju podatkov v dobljeno optimalno število skupin.

Algoritem 1 Razvrščanje s konsenzom

vhod:
množica elementov $D = \{e_1, e_2 ... e_N\}$
razvrščevalni algoritem *Cluster*
vzorčevalni algoritem *Resample*
število iteracij ponovnega vzorčenja in razvrščanja H
seznam števila skupin, ki jih želimo generirati $\mathcal{K} = \{K_1 ... K_{max}\}$

for K $\in \mathcal{K}$ **do**
  $M \leftarrow 0$ {množica povezovalnih matrik, na začetku prazna}
  for $h = 1, 2 ... H$ **do**
    $D^{(h)} \leftarrow Resample(D)$ {zgradi premešano množico D}
    $M^{(h)} \leftarrow Cluster(D^{(h)}, K)$ {razvrsti $D^{(h)}$ v K skupin}
    $M \leftarrow M \cup M^{(h)}$
  end for{for h}
  $\mathcal{M}^{(K)} \leftarrow$ *izračunaj matriko konsenzov iz* $M = \{M^{(1)} ... M^{(H)}\}$
end for{for K}
$\hat{\text{K}} \leftarrow$ *najboljši* $K \in \mathcal{K}$ *pridobljen s porazdelitvijo konsenza* $\mathcal{M}^{(K)}$
$P \leftarrow$ *razdeli* D *v* $\hat{\text{K}}$ na podlagi $\mathcal{M}^{(\hat{K})}$

3.3 Merjenje konsenza

Recimo, da je algoritem za vzorčenje in razvrščanje izbran. Določiti moramo še metodo za predstavitev in ocenjevanje različnih razvrstitev, izvedenih nad različnimi vzorci vhodne množice podatkov. Za ta namen se uporablja matrika konsenzov. To je matrika velikosti $N \times N$, ki za vsak par elementov hrani razmerje med številom, ko se oba elementa pojavita skupaj v vzorcu in številom, ko sta elementa razvrščena v isto skupino. Dobimo jo tako, da za vsako vrednost izračunamo povprečje povezovalnih matrik istoležnih elementov, ki jih dobimo v vsaki iteraciji razvrščanja vzorca elementov.

Z $D^{(1)}, D^{(2)}...D^{(H)}$ označimo H premešanih množic elementov, pridobljenih z vzorčenjem vhodne množice. $M^{(h)}$ označuje $(N \times N)$ povezovalno matriko nad množico podatkov $D^{(h)}$, katere vrednosti definira enačba:

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{če elementa } i \text{ in } j \text{ pripadata isti skupini} \\ 0 & \text{sicer} \end{cases} \tag{3.1}$$

Na podoben način definiramo (NxN) matriko pojavitev $I^{(h)}$:

$$I^{(h)}(i,j) = \begin{cases} 1 & \text{če sta elementa } i \text{ in } j \text{ prisotna v množici } D^{(h)} \\ 0 & \text{sicer} \end{cases} \tag{3.2}$$

Matriko konsenzov M definiramo kot normalizirano vsoto povezovalnih matrik vseh generiranih množic vzorcev $D^{(h)} : h = 1, 2...H$:

$$\mathcal{M}(i,j) = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)} \tag{3.3}$$

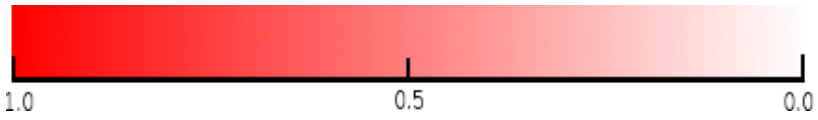
Element matrike (i,j) hrani vrednost, ki nam pove razmerje med številom, ko sta bila oba elementa skupaj v isti skupini, in številom, ko sta bila oba elementa skupaj izbrana iz vhodne množice v vzorec. Matrika je simetrična:

$$M(i,j) = M(j,i). \tag{3.4}$$

Vrednosti so realna števila med 0 in 1. Popolna matrika konsenzov je sestavljena iz samih 0 in 1. Pomembna lastnost matrike konsenzov je, da če so si elementi, ki pripadajo isti skupini, v matriki sosednji, dobimo v idealnem primeru bločno diagonalno matriko (angl. *block-diagonal matrix*) enic, ki jih obkrožajo ničle. Matrika konsenzov ima še eno pomembno lastnost, ki nam bo prav tako prišla prav. Uporabimo jo lahko kot mero podobnosti pri hierarhičnem razvrščanju.

3.4 Vizualizacija matrike konsenzov

Glede na lastnosti matrike konsenzov je povsem samoumevno, da jo uporabimo za vizualizacijo in pomoč pri odločitvi glede razvrščanja. Matriko lahko narišemo v obliki mreže s kvadratnimi prostori, ki jih ustrezno pobarvamo. Vsak prostor pripada enemu elementu v matriki konsenzov. Vrednosti v matriki uporabimo za določitev odtenka barve. Vrednost 1 predstavlja temno rdečo barvo, 0 belo, vmesne vrednosti pa predstavljajo ustrezne vmesne odtenke rdeče barve. Kako se barva spreminja glede na vrednost, prikazuje slika (3.1). S takšnim barvanjem dobimo toplotno sliko (angl. *heat map*). Na ta

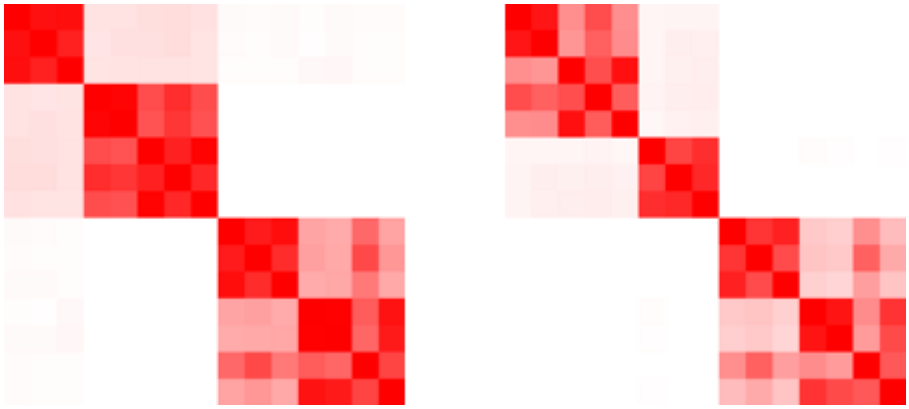


Slika 3.1: Sprememba barve glede na vrednost.

način bi za popolno matriko konsenzov dobili na diagonali temno rdeče kvadrate na belem ozadju.
Sliki (3.2) in (3.3) prikazujeta toplotne slike, dobljene z razvrščanjem testnega primer A.1 v 2, 3, 4 in 5 skupin. Testni primer je naveden v dodatku A na koncu diplomskega dela. S pomočjo toplotnih slik lahko določimo število

Slika 3.2: Toplotni sliki primera A.1 iz dodatka A za 2 in 3 skupine.

skupin v podatkih in njihovo stabilnost. Čistejše kot so barve, bolj temno rdeči kot so kvadrati in bolj kot je belo ozadje za neko število skupin, večja je verjetnost, da je v podatkih toliko skupin. V tem primeru lahko opazimo, da lahko brez težav na levi sliki slike (3.2) zaznamo dva kvadrata, na desni pa



Slika 3.3: Toplotni sliki primera A.1 iz dodatka A za 4 in 5 skupin.

tri. Na sliki (3.3), ki prikazuje toplotni sliki, ko je algoritem razvrščal podatke v 4 in 5 skupin, so meje med kvadrati opazno slabše zaznavne. Posledično gre za 2 ali 3 skupine. V tem primeru bi se zgolj na podlagi toplotne slike težko odločili, ali sta skupini 2 ali so 3. Obe sliki imata nekaj vmesnih odtenkov rdeče barve, kar kaže, da se je algoritem nekajkrat odločil napačno. Razlog za napačno razvrščanje je predvsem to, da je ta primer zelo majhen, vsaki skupini pripada le nekaj elementov, kar razvrščevalni algoritmi težko zaznajo. Opazna je tudi različna velikost kvadratov. V dodatku v tabeli (A.2), lahko vidimo, da ima prva skupina 5 elementov, druga 3 in tretja 7. Zato lahko pričakujemo, da bodo kvadrati različnih velikosti. Ker je bila toplotna slika pred izrisom urejena, si velikosti ne sledijo v istem zaporedju, kot so zapisane v tabeli.

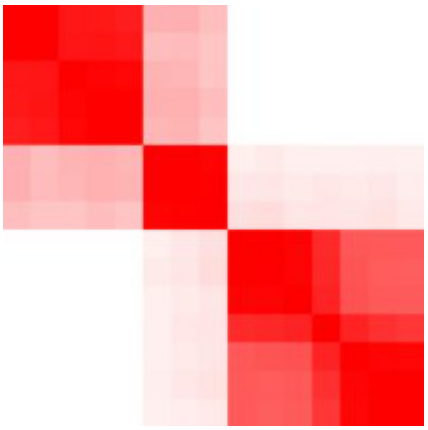
3.5 Urejanje matrike konsenzov

Pri risanju matrike konsenzov želimo dobiti čim lepšo bločno diagonalno matriko. A če nismo pazljivi, se nam lahko zgodi, da dobimo povsem popačeno sliko, kljub temu da so vhodni podatki razvrščeni idealno. Kot smo lahko videli že na sliki (3.2), lahko z nje razberemo tudi velikost posamezne dobljene skupine. A pogoj za to je, da so elementi, ki so skupaj v skupini, skupaj tudi v matriki. Biti morajo sosednji. Če so si sosednji, bomo dobili lepo bločno diagonalno matriko (ob predpostavki, da so skupine dobro zaznane). Če pa so elementi v matriki naključno razvrščeni, bomo dobili mrežo, polno pik, s katere bomo težko razbrali kakšno uporabno informacijo. Na splošno ne vemo, kateri elementi pripadajo istim skupinam, zato ne moremo računati na to, da bodo elementi v matriki urejeni. Razvrščeni so naključno, tako kot so podani

na vhodu v algoritem. Cilj urejenja je, da ustrezno prerazporedi elemente v matriki.

Za urejanje lahko uporabimo kar matriko konsenzov. Uporabimo jo kot mero podobnosti za gradnjo hierarhičnega drevesa. Tako zgrajeno drevo ima liste v dendrogramu razvrščene tako, da so si elementi, ki imajo visoke vrednosti konsenza, sosednji. Če matriko uredimo, kot so urejeni listi v dendrogramu, dobimo želeno bločno diagonalno obliko matrike. Pri tem moramo poudariti, da je vrstni red elementov v vrsticah in stolpcih matrike enak. Pred izrisom toplotnih slik v prejšnjem poglavju matrike ne bi bilo potrebno urediti, ker so podatki že urejeni. Če matrike za razvrščanje v 3 skupine ne bi uredili, bi dobili sliko (3.5). Pri tej sliki lahko vidimo, da si velikosti kvadratov sledijo v enakem zaporedju, kot si velikosti skupin v dodatku v tabeli (A.2). Slika (3.4) prikazuje, kaj se zgodi, če podatke s tabele (A.1) premešamo. Algoritem je bil pognan nad podatki iz dodatka B, podani so v tabeli (B.1). Tabela je enaka tabeli (A.1), le elementi so malo premešani. V tem primeru nam slika ne pove prav dosti. Če podatke pred izrisom uredimo, dobimo desno toplotno sliko na sliki (3.2).

Slika 3.4: Toplotna slika primera B.1.



Slika 3.5: Toplotna slika nesortirane matrike.

3.6 Določanje števila skupin

S pomočjo matrike konsenzov lahko določimo tudi število skupin. Glede na to, da je optimalna matrika sestavljena iz samih 0 in 1, lahko odklon od teh vrednosti interpretiramo kot pomanjkanje stabilnosti. Groba ideja je, da na-

redimo matrike $M^{(K)}$ za vsako vrednost iz množice možnega števila skupin ($K = 2, 3...K_{max}$). Nato primerjamo, pri katerem številu skupin dobimo matriko, ki ima vrednosti najbliže idealnim (same 0 in 1). Mera za ocenjevanje optimalnega števila skupin, ki se uporablja pri razvrščanju s konsenzom, se imenuje porazdelitev konsenza (angl. *consensus distribution*), ker nam pove, kako so elementi matrike porazdeljeni na intervalu 0-1.

3.6.1 Porazdelitev konsenza

Če bi narisali histogram idealne matrike konsenzov, bi dobili dva koša, koncentirana pri vrednostih 0 in 1. Šum v vhodnih podatkih pa povzroči, da vrednosti limitirajo v en koš, ki ima središče pri nekem realnem številu med 0 in 1. Do tega pride, ker imata katera koli dva elementa enako verjetnost, da sta razvrščena v isto skupino. Slika a) na sliki (3.6) prikazuje histogram vrednosti matrike konsenzov za primer (B.1) podan v dodatku B. Na histogramu sta lepo vidna dva koša v okolici vrednosti 0 in 1. Za podan histogram lahko izračunamo in narišemo porazdelitev empirične vsote (CDF):

$$CDF(c) = \frac{\sum_{i<j} I\{M(i,j) < c\}}{N(N-1)/2} \tag{3.5}$$

Pri tem $I\{pogoj\}$ predstavlja funkcijo pokazatelja (angl. *indicator function*), ki je definirana tako, da ima vrednost 1, če je pogoj resničen, sicer je 0.
M(i,j) je element *(i,j)* matrike konsenzov *M*,
N je število vrstic (oziroma stolpcev, števili sta enaki, ker je matrika kvadratna) matrike *M*.
Slika c) na sliki (3.6) prikazuje krivuljo porazdelitve empirične vsote za različno število skupin. Idealna oblika je stopničasta. Razliko med dvema krivuljama lahko delno povzamemo z izračunom površine pod njo:

$$A(K) = \sum_{i=2}^{m} [x_i - x_{i-1}] CDF(x_i) \tag{3.6}$$

Na sliki porazdelitve empirične vsote lahko opazimo, da se površina med krivuljama med dvema in tremi skupinami opazno poveča, medtem ko se razlika med naslednjimi pari vrednosti zmanjšuje. To je posledica tega, da se začne stabilnost skupin zmanjševati, ko enkrat dosežemo pravilno število. Ker postane število skupin preveliko, so le-te vedno bolj nestabilne, zato se vrednosti v matriki konsenzov oddaljijo od idealnih 0 in 1. To obnašanje lahko povzamemo tako, da narišemo krivuljo spremembe površine pod CDF, ko se število

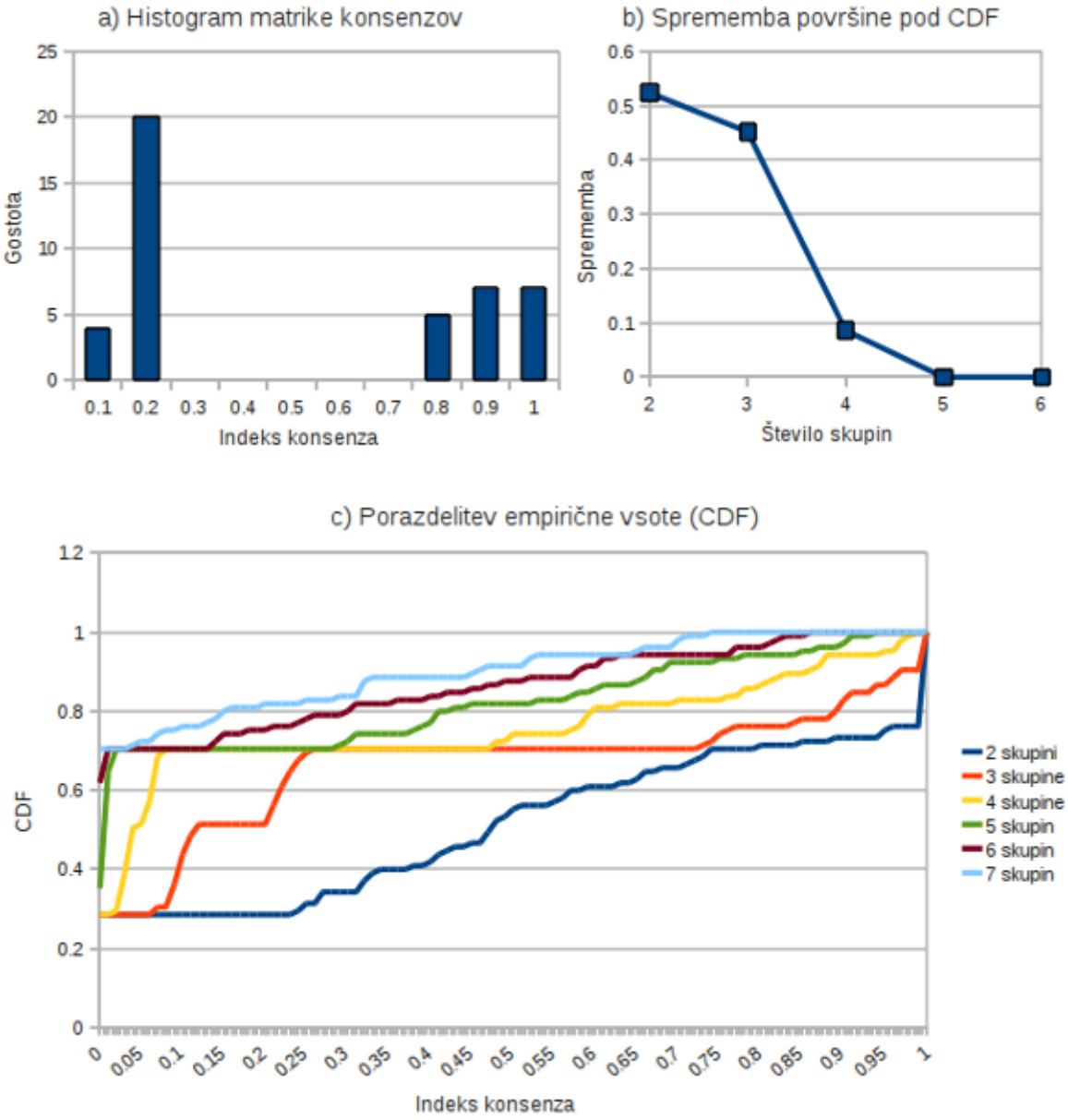
skupin k povečuje. Spremembo površine izračunamo z enačbo:

$$\Delta(K) = \begin{cases} A(K) & \text{če } K = 2 \\ \frac{A(K+1)-A(K)}{A(K)} & \text{če } K > 2 \end{cases} \tag{3.7}$$

Slika b) na sliki (3.6) prikazuje krivuljo spremembe površine CDF za primer (B.1) iz dodatka B. Vidimo lahko, da izstopata vrednosti 2 in 3. Če bi se odločali za število skupin zgolj po maksimalni vrednosti spremembe površine, bi se napačno odločili za 2 skupini. Kadar sta dve vrednosti blizu skupaj, kot v tem primeru 2 in 3, je potrebno pogledati še obliko krivulje CDF. Iščemo, katera ima lepšo stopničasto obliko. Ker je v tem primeru razlika spremembe površine med 2 in 3 skupinami majhna, krivulja za 3 skupine pa opazno bolj stopničasta, lahko pravilno določimo, da gre za 3 skupine.

3.7 Algoritmi vzorčenja

Kot je razvidno s psevdokode (1), se za gradnjo matrike konsenzov pri razvrščanju ne uporablja celotne vhodne množice, ampak se v vsaki iteraciji razvrščanja zgradi podmnožico. Za gradnjo podmnožice sta primerna predvsem dva algoritma, metoda stremena (angl. *bootstrap*) in podvzorčenje (angl. *subsampling*). Metoda stremena zgradi množico tako, da se lahko isti element v njej ponovi večkrat, elemente jemlje iz začetne množice z vračanjem. To pomeni, da lahko z njo zgradimo množico iste ali celo večje velikosti, kot je vhodna množica. Ponavljanje istega elementa lahko negativno vpliva na razvrščanje. Ker se isti element ponovi n-krat, lahko razvrščevalni algoritem naredi skupino velikosti n, v katero razvrsti teh n enakih elementov. Podvzorčenje zgradi množico, ki je manjša od celotne tako, da elemente naključno jemlje iz nje, pri čemer se jih ne vrača, isti element ne more biti izbran večkrat.



Slika 3.6: Grafi za testni primer B.1. Slika a) prikazuje histogram vrednosti v matriki konsenzov, na sliki b) je graf spremembe površine pod krivuljo porazdelitve empirične vsote (CDF), c) je graf porazdelitve empirične vsote za razvrstitev v različno število skupin.

Poglavje 4

Implementacija in uporaba algoritma

4.1 Implementacija

Algoritem je implementiran kot razred `ConensusClustering` v programskem jeziku Python, ker se implementacija in uporaba močno prepletata z okoljem Orange[1]. Izgled konstruktorja razreda prikazuje:

```
def __init__(self, data = None, r = 400, max_clusters = 5,
  min_clusters = 2, sample_size = 0.8, cluster_alg = 'KMeans',
  resample = 'SubSample', initialize_only = False)
```

Pomen atributov in možne vrednosti prikazuje tabela (4.1). Kot je razvidno iz zaloge vrednosti parametra `cluster_alg`, se v notranji zanki algortima za razvrščanje uporablja bodisi hierarhično razvrščanje bodisi razvrščanje z voditelji. Za gradnjo vzorcev lahko uporabimo metodo stremena ali podvzorčenje. Nadgradnja s kakšnim novim algoritmom razvrščanja ali podvzorčenja je preprosta, potrebni posegi v kodo so minimalni.

[1] http://www.ailab.si/orange/

Tabela 4.1: Pomeni in zaloga vrednosti parametrov.

Parameter	**Možne vrednosti**	**Pomen**
`data`	Orangeov objekt, podatki prebrani iz datoteke .tab z metodo orange.ExampleTable ali None	Vhodni podatki
`r`	Celo število od 1 naprej	Število iteracij, kolikokrat naj algoritem zgradi in razvrsti podmožico elementov.
`max_clusters`	Od vključno 3 naprej in večje od `min_clusters`	Zgornja meja števila skupin
`min_clusters`	Od vključno 2 naprej	Spodnja meja števila skupin
`sample_size`	Decimalna vrednost od 0.1 do 1.0	Velikost podmnožice elementov podana kot delež celotne vhodne množice
`cluster_alg`	"KMeans" ali "hierahical"	Algoritem, ki se uporablja za razvrščanje pri gradnji matrike konsenzov.
`resample`	"SubSample" ali "bootstrap"	Algoritem za kreiranje vzorcev
`initialize_only`	True ali False	Ali želimo zgolj kreirati objekt s podanimi podatki ali tudi pognati algoritem.

4.2 Primer uporabe

Najprej moramo uvoziti potrebni knjižnici Orange in razvrščanje s konsenzom ConsensusClustering, kar naredimo z:

```
>>> import orange
>>> import ConsensusClustering
```

Nato lahko preberemo vhodne podatke:

```
>>> data = orange.ExampleTable("/pot/do/datoteke.tab")
```

Kot je razvidno iz konstruktorja, imajo parametri že privzete vrednosti, zato lahko preprosto poženemo razvrščanje s konsenzom tako, da podamo le vhodne podatke:

```
>>> consensusClustering =
    ConsensusClustering.ConsensusClustering(data)
```

S tem smo kreirali nov objekt in hkrati pognali razvrščanje nad podanimi podatki. Če želimo zgolj kreirati nov objekt, spremeniti zgornjo mejo števila skupin in za razvrščanje uporabiti hierarhično razvrščanje, šele nato pa pognati algoritem, to naredimo tako:

```
>>> consensusClustering = ConsensusClustering.ConsensusClustering
    (max_clusters = 8, cluster_alg = "hierarhical",
    initialize_only = True)
>>> consensusClustering(data)
```

Ko se izvajanje konča, se nam izpišeta prvo in drugo optimalno število skupin, dobljena na podlagi spremembe površine pod krivuljo porazdelitve empirične vsote, in njuni vrednosti:

```
Optimal number of clusters: 2 value: 0.523243276026
Second best number of clusters: 3 value: 0.440247975574
```

S klicem metode `getGraphValues()` lahko dobimo podatke za izris grafa porazdelitve empirične vsote (3.5) in krivulje spremembe površine (3.7):

```
>>> cc.getGraphValues()
** Data for proportion change under CDF **
Number of clusters:
  2 3 4 5 6
```

```
Proportion change values:
  0.523243276026 0.440247975574 0.105263275466 0 0

** Data for CDF graph
x axis values:
  0.0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 ...
CDF values:
  0.28571428571 0.28571428571 0.28571428571 0.28571428571 ...
  0.28571428571 0.28571428571 0.28571428571 0.28571428571 ...
  0.28571428571 0.29523809523 0.42857142857 0.50476190476 ...
  0.28571428571 0.38095238095 0.52380952381 0.67619047619 ...
  0.55238095238 0.70476190476 0.70476190476 0.70476190476 ...
```

Prvi dve zaporedji števil predstavljata vrednosti na abscisi in ordinati grafa porazdelitve empirične vsote, tretje zaporedje so vrednosti na abscisi grafa spremembe površine, naslednja zaporedja pa vrednosti na ordinati pri različnem številu skupin. Vsa števila so ločena s tabulatorji, tako da jih lahko preprosto prenesemo v Excel/Calc in izrišemo graf.

Za izris toplotne slike (3.4) uporabimo metodo `drawMatrix(Integer)`, ki ji kot parameter podamo število skupin, za katere želimo dobiti toplotno sliko.

```
>>> consensusClustering.drawMatrix(3)
```

Izpis skupin in njihovih elementov dosežemo s klicem metode `printClusterMembers(Integer)`, ki kot parameter sprejme število skupin, ki jih želimo generirati.

```
>>> consensusClustering.printClusterMembers(3)
0  1  2  3  4  5
6  7  8  9
10  11  12  13  14
```

Vsaka vrstica izpisa predstavlja eno skupino. Vrednosti predstavljajo mesto elementa v vhodni datoteki. V gornjem primeru so bili elementi zapisani po vrsti, kot tudi sodijo v skupine, zato vrednosti naraščajo linearno od 0 do 14, gre za testni primer (A.2). Vidimo lahko, da je algoritem napačno razvrstil elemente 5, 8 in 9.

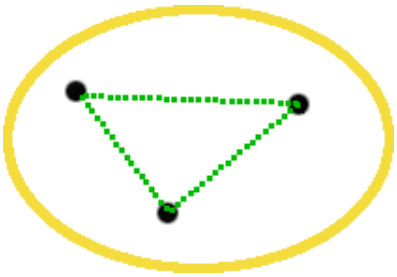
Poglavje 5

Eksperimentalno vrednotenje

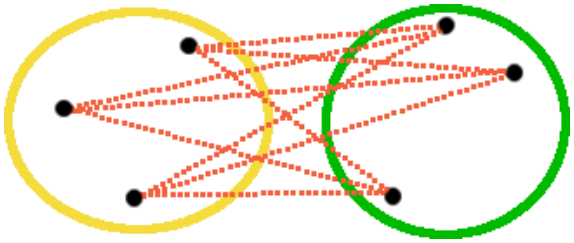
Cilj vrednotenja rezultatov razvrščanja je ugotoviti pravilnost dobljenih skupin. Vsak algoritem za razvrščanje nam namreč lahko razvrsti podatke, vprašanje pa je, kakšne so nastale skupine.

5.1 Tehnika vrednotenja

Tehnike vrednotenja razvrščanja lahko razdelimo na dve skupini: nadzorovane in nenadzorovane. Nenadzorovane se ne ozirajo na zunanje znanje, uporabne so, kadar nimamo informacij o pravilni razvrstitvi. Za oceno se uporabljata dve meri, prva je kompaktnost skupin (5.1) (angl. *cluster cohesion, tightness*), ki nam pove, kako močno so povezani elementi znotraj skupine. Druga mera je izolacija skupine (5.2) (angl. *cluster separation*), ki določa, kako dobro so ločene oziroma kako oddaljene so skupine. Pri nadzorovanem načinu vredno-



Slika 5.1: Kompaktnost skupin.



Slika 5.2: Izolacija skupin.

tenja razvrščanje primerjamo z zunanjim znanjem. Ta način bomo uporabili tudi mi. Uporabili bomo testne podatke, za katere poznamo pravilno razvrstitev, in jih primerjali z razultati razvrščevalnega algoritma. Kadar je število

dobljenih skupin enako pravemu številu, ocena točnosti ni problematična. Ko ugotovimo, katera dobljena skupina se preslika v katero izmed pravilnih, le preverimo, koliko elementov je napačno razvrščenih. Bolj problematično je, ko število skupin ni enako. Kako naj takrat utežimo napačno določeno število skupin in kako napačno razvrščene elemente? Za oceno točnosti smo implementirali mero, t.i. prilagojeni Rand indeks (angl. *adjusted Rand index - ARI*), ki z omenjeno problematiko nima težav.

5.1.1 Prilagojeni Rand indeks

Prilagojeni Rand indeks[1] [3] je mera za merjenje podobnosti med dvema množicama podatkov, ki upošteva tudi različno število skupin. Vrednost, ki jo vrne, se nahaja med 0 in 1. 1 označuje popolno ujemanje, 0 povsem naključno razvrščene podatke.

Recimo, da imamo seznam S z N elementi in dve množici skupin:
$U = \{U_1, U_2, ..., U_R\}$ in $V = \{V_1, V_2, ..., V_R\}$. V eni množici so skupine, pridobljene z razvrščanjem, v drugi so skupine, v katerih so elementi razporejeni pravilno. Množici U in V lahko predstavimo s kontingenčno tabelo, v kateri element n_{ij} predstavlja število ujemajaočih elementov skupin U_i in V_j: $n_{ij} = U_i \cap V_j$.

U\V	$\mathbf{V_1}$	$\mathbf{V_2}$	...	$\mathbf{V_c}$	**Sums**
$\mathbf{U_1}$	n_{11}	n_{12}	...	n_{1C}	a_1
$\mathbf{U_2}$	n_{21}	n_{22}	...	n_{2C}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{U_R}$	n_{R1}	n_{R2}	...	n_{2RC}	a_R
Sums	b_1	b_2	...	b_c	

Vrednost prilagojenega Rand indeksa dobimo z enačbo: $\frac{indeks - pričakovaniIndeks}{maksimalenIndeks - pričakovaniIndeks}$
To lahko podrobneje zapišemo z enačbo:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]/\binom{N}{2}} \tag{5.1}$$

[1] http://en.wikipedia.org/wiki/Adjusted_rand_index

5.2 Testiranje

5.2.1 Način testiranja

Zanima nas predvsem, kako se obnese navadno razvrščanje v primerjavi z razvrščanjem s konsenzom. Primerjanje bo osredotočeno na primerjavo hierarhičnega razvrščanja z razvrščanjem s konsenzom, pri katerem se v notranji zanki uporablja hierarhično razvrščanje. Na enak način bomo primerjali razvrščanje z voditelji. Parametri, s katerimi bo izvedeno testiranje, bodo pripisani k vsakemu testu posebej. Kar je skupno vsem testom, je, da so izvedeni desetkrat, rezultat pa izračunan kot povprečje pridobljenih rezultatov v 10 ponovitvah. Razlog za večkratno ponavljanje je, da rezultati razvrščanja včasih močno nihajo. To velja tako za navadno razvrščanje kot tudi za razvrščanje s konsenzom. Razlog za nihanje ocen pri razvrščanju z voditelji je, kot je napisano že v opisu algoritma (2.1), da se začetne točke izbere naključno, kar lahko vpliva na dobljene rezultate.

Tabela (5.1) prikazuje ocene ARI desetih razvrščanj in povprečni oceni. Razvrščanje je bilo izvedeno nad podatki iris.tab. S tabele je razvidno opazno nihanje ocene navadnega razvrščanja z voditelji, medtem ko je ocena razvrščanja s konsenzom v vseh desetih primerih povsem enaka.

Tabela 5.1: Rezultati razvrščanja.

Razvrščanje s konsenzom	Razvrščanje z voditelji
0.722	0.431
0.722	0.427
0.722	0.737
0.722	0.737
0.722	0.720
0.722	0.737
0.722	0.700
0.722	0.682
0.722	0.737
0.722	0.720
Povprečna vrednost	Povprečna vrednost
0.722	0.662

Testiranje bomo izvedli nad tremi podatkovnimi množicami[1]. Celotne tabele rezultatov testiranja so podane na koncu v dodatku C.
Iris.tab vsebuje 148 elementov s štirimi atributi, število skupin je 3.
Wine.tab vsebuje 178 elementov s 13 atributi, število skupin je 3.
Glass.tab vsebuje 212 elementov z 9 atributi, število skupin je 6.

5.2.2 Rezultati testiranja pri uporabi razvrščanja z voditelji

Za vsako testno množico podatkov bosta navedeni vrednosti ocen ARI in ocenjeno optimalno število skupin. Ocena optimalnega števila skupin pri razvrščanju s konsenzom je pridobljena zgolj z upoštevanjem spremembe površine pod krivuljo porazdelitve empirične vsote, brez upoštevanja oblike krivulje. Za predlagano število skupin vzamemo tisto, pri katerem je sprememba največja. Klasično razvrščanje z voditelji poganjamo s privzetimi lastnostmi. Pri razvrščanju z voditelji je implementirano avtomatsko določanje števila skupin.

Podatkovna množica iris.tab
Razvrščanje s konsenzom je bilo pognano z naslednjimi parametri: število ponovitev 400, minimalno število skupin 2, maksimalno število 6, velikost vzorca 0.8.

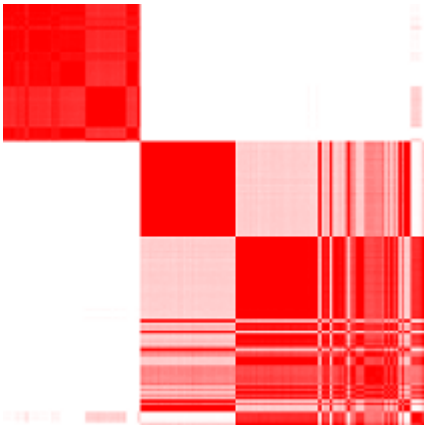
Tabela (5.2) prikazuje rezultate razvrščanja. Števili v stolpcu št. skupin povesta, kolikokrat je kateri algoritem pravilno uganil število skupin. Vidimo lahko, da sta oba algoritma v vseh desetih ponovitvah pravilno ugotovila število skupin. Sodeč po rezultatih, zapisanih v tabeli (5.1), bi glede na odstopanje pri nekaterih ocenah pomislili na to, da je algoritem za navadno razvrščanje z voditelji slabo določil število skupin. Posledično bi lahko bila ocena v primerih, ko bi bilo število skupin napačno določeno, slabša. To ne velja, algoritem je preprosto slabo razvrščal. Razvrščanje s konsenzom se je izkazalo za natančnejše.

Tabela 5.2: Rezultati razvrščanja podatkovne množice iris.tab.

Razvrščanje s konsenzom		Razvrščanje z voditelji	
ARI	Št. skupin	ARI	Št. skupin
0.722	10	0.662	10

[1] http://www.ailab.si/orange/datasets.psp

Slika (5.3) prikazuje dobljeno toplotno sliko za 3 skupine. S slike je razvidno,



Slika 5.3: Toplotna slika, dobljena pri razvrščanju podatkov iris.tab.

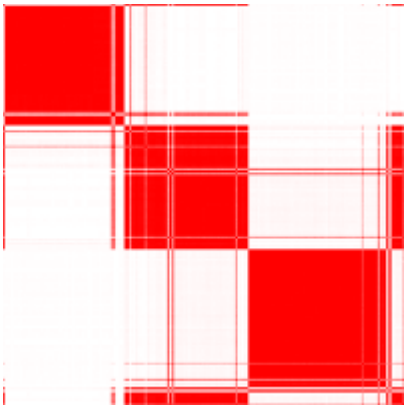
da je pri razvrščanju prišlo do napak, a so še vedno vidne meje med skupinami.

Podatkovna množica wine.tab
Parametri algoritmu so enaki kot pri iris.tab.
Tabela (5.3) prikazuje rezultate razvrščanja. Ponovno sta oba algoritma v vseh desetih ponovitvah pravilno ugotovila število skupin. Tudi pri razvrščanju sta bila zelo uspešna, razvrščanje s konsenzom je bilo ponovno boljše. Glede na dobro oceno ARI lahko pričakujemo lepo toplotno sliko, prikazuje jo slika (5.4). S toplotne slike so lepo razvidne 3 skupine, veliko lepše kot pri testu iris.tab. To smo lahko glede na dobro vrednost ARI tudi pričakovali.

Podatkovna množica glass.tab
Algoritem je bil ponovno pognan s 400 iteracijami notranje zanke, spodnja meja števila skupin je bila ponovno 2, zgornja pa 9.
Tabela (5.4) prikazuje rezultate razvrščanja. Pri teh testnih podatkih so rezultati obeh algoritmov zelo slabi. Glede na to, da razvrščanje s konsenzom

Tabela 5.3: Rezultati razvrščanja podatkovne množice wine.tab.

Razvrščanje s konsenzom		Razvrščanje z voditelji	
ARI	Št. skupin	ARI	Št. skupin
0.915	10	0.850	10

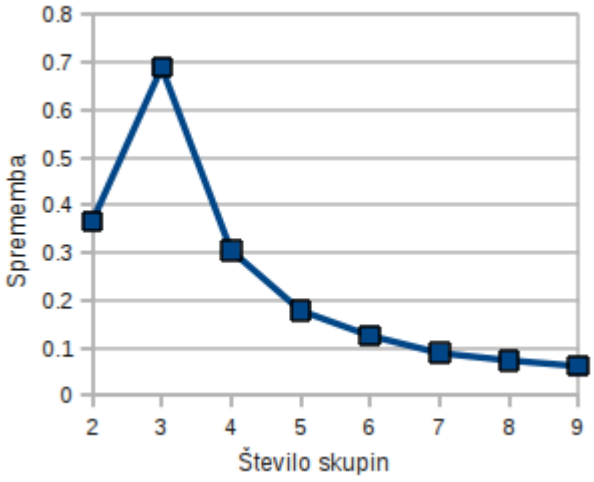


Slika 5.4: Toplotna slika razvrščanja podatkovne množice wine.tab.

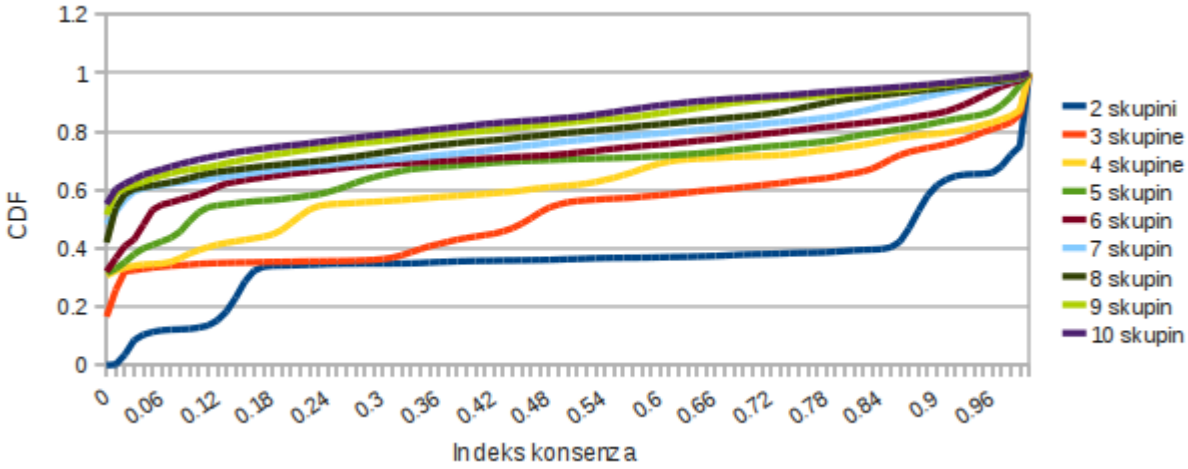
temelji na navadnem razvrščanju z voditelji, ne moremo pričakovati čudeža. Očitno je ta podatkovna množica zelo zahtevna za razvrščanje, oba algoritma sta v vseh desetih primerih določila, da je optimalno število skupin 3 namesto 6, kolikor jih je res. V tem primeru si ne moremo pomagati niti z grafi niti s toplotno sliko. Z grafa slike spremembe površine pod CDF (5.5) je razvidno, da 3 zelo izstopa, medtem ko se 6 povsem porazgubi. Tudi graf porazdelitve empirične vsote (5.6) nam ne pove kaj dosti več. Še najbolj stopničasto izgledata krivulji za dve in za tri skupine, potem pa so krivulje vedno bolj linearno naraščajoče. S toplotnimi slikami si prav tako ne moremo pomagati. Slika (5.7) prikazuje toplotne slike za 2, 3 in 4 skupine, slika (5.8) pa za 5, 6 in 7 skupin. Najlepše zgleda slika za 2 skupini, ostale so vse precej nerazločne.

Tabela 5.4: Rezultati razvrščanja podatkovne množice glass.tab.

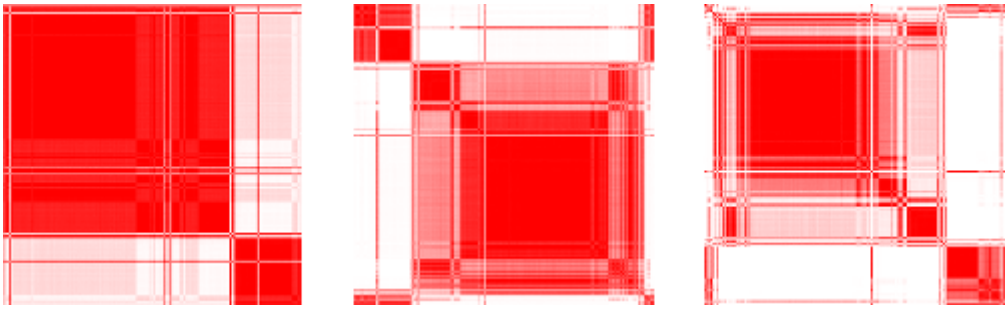
Razvrščanje s konsenzom		KMeans	
ARI	Št. skupin	ARI	Št. skupin
0.137	0	0.150	0



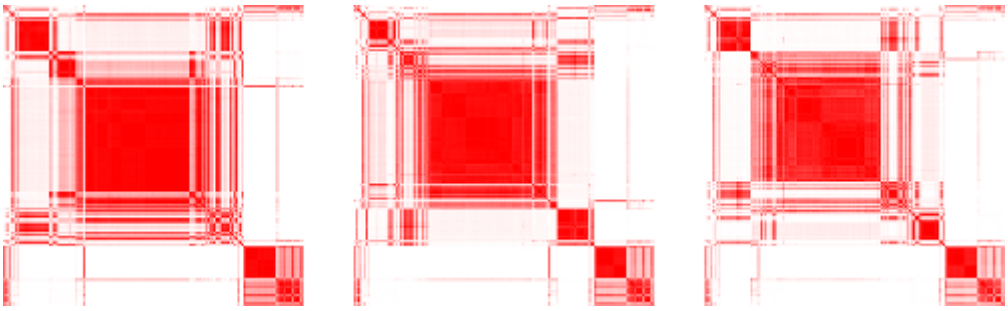
Slika 5.5: Sprememba površine pod CDF pri podatkovni množici glass.tab.



Slika 5.6: Porazdelitev empirične vsote glass.tab.



Slika 5.7: Toplotne slike za 2, 3 in 4 skupine glass.tab.



Slika 5.8: Toplotne slike za 5, 6 in 7 skupin glass.tab.

5.2.3 Rezultati testiranja hierarhičnega razvrščanja

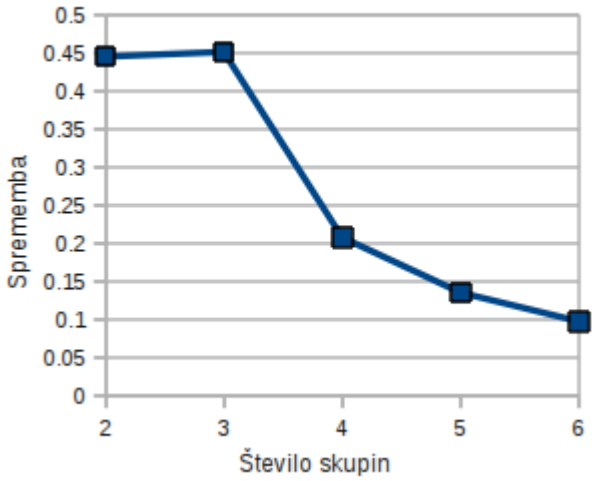
Klasično hierarhično razvrščanje nima implementiranega predlaganja optimalnega števila skupin, zato bomo pri testiranju uporabili število, ki ga predlaga razvrščanje s konsenzom, in znano pravilno število.

Podatkovna množica iris.tab
Algoritem je bil pognan s 400 ponovitvam, minimalno število skupin je bilo 2, maksimalno 6, velikost vzorca 0,8.
Tabela (5.5) prikazuje rezultat razvrščanja. V tabeli C.1 lahko vidimo, da je ocena razvrščanja nihala med dvema vrednostma. Očitno je bil kakšen element močno na meji, v katero skupino naj se uvrsti. Nihala je tudi ocena števila skupin. V petih primerih je algoritem pravilno določil, da so skupine 3, v preostalih petih pa napačno, da so 2. Vrednosti spremembe pod krivuljo med dvema skupinama in tremi je bila zelo majhna, kar je videti z grafa (5.9). Graf prikazuje primer, ko je naklonjenost trem skupinam malo večja kot dvem. V ostalih petih primerih, ko je ocena napačno določila, da sta skupini dve, je bila situacija podobna. Dve skupini sta za malo prevladali. Ker sta vrednosti tako blizu, pogledamo, če nam pri odločitvi lahko pomaga graf porazdelitev empirične vsote (5.10). Najlepše izgleda krivulja pri dveh skupinah, tako da se bi lahko na podlagi teh dveh ocen napačno odločili, da sta skupini dve.
O tem, da se v podatkih skrivata 2 skupini, nas še dodatno prepriča toplotna slika (5.11). Leva slika, ki predstavlja sliko za 2 skupini, je neverjetno čista, skoraj popolna, zelo malo je svetlo rdečih delov, robovi so povsem ravni. To pomeni, da sta bili dobljeni skupini zelo stabilni, elementi, ki so se pojavili skupaj, so bili skupaj v vsaki iteraciji. Desna slika je po drugi strani opazno bolj motna. Očitno je gradnjo drevesa nekaj močno zavedlo.

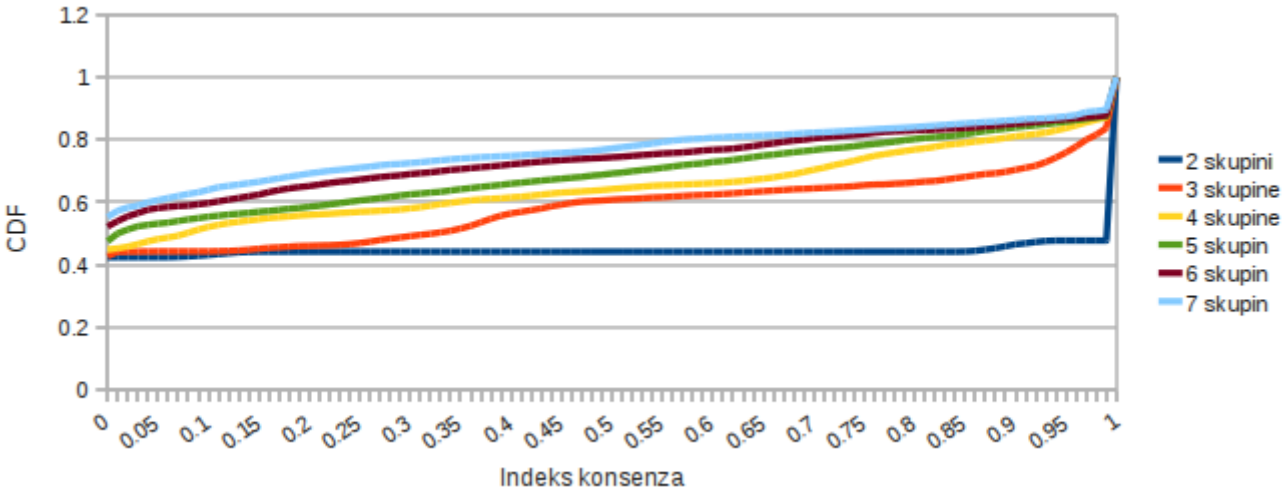
Podatkovna množica wine.tab
Algoritem je bil pognan z enakimi vrednostmi parametrov kot pri iris.tab.
Tabela (5.6) prikazuje rezultat. Rezultati so zelo zanimivi, sploh v primerjavi z rezultati razvrščanja z voditelji, ki se je za razliko od hierarhičnega obneslo

Tabela 5.5: Rezultati razvrščanja podatkovne množice iris.tab.

Razvrščanje s konsenzom			Hierarhično	
ARI 2	ARI 3	Št. skupin	ARI 2	ARI 3
0.561	0.701	5	0.561	0.725

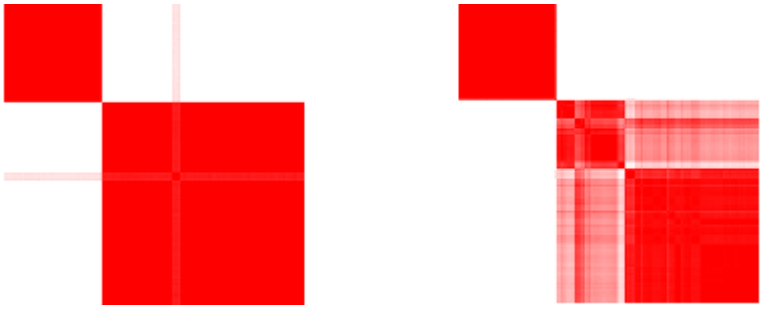


Slika 5.9: Sprememba površine pod krivuljo CDF za podatkovno množico iris.tab.



Slika 5.10: Krivulje porazdelitve empirične vsote podatkovne množice iris.tab.

zelo dobro. Testni primer wine.tab povzroči hierahičnemu razvrščanju hude težave, ARI ocena je 0, kar pomeni, da algoritem ni ugotovil nobenega vzorca za formiranje skupin. Tudi v tem primeru razvrščanje s konsenzom ne naredi čudeža, rezultati so enako zanič kot pri navadnem hierarhičnem razvrščanju.



Slika 5.11: Levo toplotna slika za 2 skupini, desno za 3.

Tabela 5.6: Rezultati razvrščanja množice wine.tab.

Razvrščanje s konsenzom		hierarhično
ARI	Št. skupin	ARI
0.0	10	0.0

Maksimalna sprememba površine pod krivuljo sicer pravilno določi število skupin, a to ne igra bistvene vloge, ker so dobljene skupine povsem napačne.

Podatkovna množica glass.tab

Za testiranje glass.tab smo ponovno nastavili zgornjo mejo števila skupin na 9, ostali parametri so enaki kot za prejšnja dva testa. Glass.tab se je tudi pri hierarhičnem načinu izkazal za problematičnega. Razvrščanje s konsenzom ponovno ne ugotovi pravilnega števila skupin. Če sami podamo pravilno število skupin, je povprečna ocena malo boljša od navadnega razvrščanja, pride pa do enakega pojava kot pri iris.tab. Ocena pri šestih skupinah niha, medtem ko je pri štirih povsem stabilna.

Tabela 5.7: Rezultati razvrščanja glass.tab.

Razvrščanje s konsenzom			Hierarhično	
ARI 4	ARI 6	Št. skupin	ARI 4	ARI 6
0.014	0.206	0	0.014	0.195

5.2.4 Povzetek testiranj

Pri razvrščanju z voditelji lahko vidimo, da razvrščanje s konsenzom popravi rezultate. Predvsem ugodno vpliva na konsistentnost rezultatov. Pri več ponovitvah se ocena rezultata popolnoma nič ne spremeni, medtem ko je pri navadnem razvrščanju opaziti nihanje ocene. Tudi predlagano število skupin je pravilno v primeru, ko je razvrščanje dobro in je takšna tudi ocena.

Razvrščanje s konsenzom v povezavi s hierarhičnim razvrščanjem ne prinese velikih izboljšav. Zanimivo je, kako so vrednosti ocene pri navadnem hierarhičnem razvrščanju ves čas enake. Razlog za konstantnost je, da pri hierarhičnem razvrščanju ni naključnosti kot pri razvrščanju z voditelji, matrika razdalj je vedno enaka, zato so formirane skupine vedno enake in posledično tudi ocena. Z gradnjo dreves iz naključno narejenih vzorcev vnesemo naključnost. Z velikim številom ponovitev sicer to naključnost omilimo, a ima očitno še vedno velik vpliv na končno razvrščanje, zaradi česar pride do nihanja ocen.

Tako za razvrščanje z voditelji kot hierarhično razvrščanje velja, da razvrščanje s konsenzom ne bo čudežno spreobrnilo rezultatov. Če imata klasična načina težave z razvrščanjem podatkov, ju niti razvrščanje s konsenzom ne reši.

5.2.5 Vpliv števila ponovitev

Eden izmed parametrov, ki lahko močno vpliva na rezultate in katerega vpliv na rezultate bi bilo zanimivo opazovati, je število iteracij. Število iteracij določa, kolikokrat se izvedeta gradnja vzorca in razvrščanje elementov v skupine. Tabela (5.8) prikazuje ARI oceno razvrščanja, pri čemer je bilo uporabljeno različno število ponovitev, od 10 do 400, velikost vzorca je bila vedno 0.8. Test je bil za oba načina razvrščanja ponovljen trikrat, da lahko vidimo vpliv velikost pri večkratnem poganjanju. Za test je bila uporabljena podatkovna množica iris.tab. Po pričakovanju se pri razvrščanju z voditelji z večanjem števila ponovitev ocena ustali. S tem ko povečamo število ponovitev, zmanjšamo vpliv naključnosti pri izbiri začetnih elementov. Malo vseeno preseneča, kako hitro se je ocena ustalila. Po drugi strani pri hierarhičnem razvrščanju ni opaziti vzorca kako bi število ponovitev vplivalo na oceno. Ocena niha ne glede na to, koliko ponovitev naredimo.

Tabela 5.8: Vpliv števila iteracij na oceno razvrščanja ARI.

Število ponovitev	KMeans			Hierarhično		
10	0.561	0.706	0.722	0.725	0.665	0.688
50	0.722	0.722	0.701	0.665	0.665	0.711
100	0.722	0.722	0.722	0.665	0.711	0.665
200	0.722	0.722	0.722	0.665	0.725	0.665
400	0.722	0.722	0.722	0.665	0.725	0.725

5.2.6 Vpliv velikosti vzorca

Drug parameter, ki lahko močno vpliva na razvrščanje, je izbira velikosti vzorca. Če vzamemo majhen vzorec, je večja možnost, da zajamemo same oziroma veliko elementov, ki spadajo v isto skupino. Pri razvrščanju jih bomo potem razbili v več skupin, namesto da bi ostali v eni. Če vzamemo velikega, pa zmanjšamo spremenljivost množice. V tabeli (5.9) so navedeni rezultati razvrščanja pri spremenljivi velikosti vzorca. Opazimo lahko, da ocena pri razvrščanju z voditelji bolj niha in dosega slabše rezultate pri manjših vzorcih. Pri hierarhičnem razvrščanju je podobno. Ocena ARI je pri manjših vzorcih slabša. Za razliko od razvrščanja z voditelji se nihanje z večanjem vzorca ne umiri.

Tabela 5.9: Vpliv velikosti vzorca na oceno razvrščanja ARI.

Velikost vzorca	KMeans			Hierarhično		
0.2	0.667	0.708	0.667	0.655	0.655	0.555
0.4	0.722	0.722	0.722	0.555	0.645	0.712
0.6	0.722	0.722	0.722	0.688	0.725	0.724
0.8	0.722	0.722	0.722	0.725	0.725	0.665

Poglavje 6

Zaključek

Razvrščanje s konsenzom je pustilo raznovrstne vtise. Ne moremo reči, da je boljše od klasičnega razvrščanja, niti da je slabše. Tako razvrščanje z voditelji kot tudi hierarhično razvrščanje sta v nekaterih primerih dosegla boljše rezultate brez uporabe razvrščanja s konsenzom kakor z njegovo uporabo. Pri razvrščanju z voditelji je bil v primeru, ko je bil rezultat klasičnega razvrščanja boljši, le-ta minimalno boljši, medtem ko je bil v nasprotnem primeru občutneje slabši. Rezultati pri klasičnem razvrščanju z voditelji opazno nihajo, če algoritem poženemo večkrat. Uporaba konsenza nihanje povsem umiri, rezultat je vedno isti. Uporaba hierarhičnega razvrščanja v razvrščanju s konsenzom v primerjavi s klasičnim hierahičnim razvrščanjem ne prinese bistvenih izboljšav.

Razvrščanje s konsenzom pa ima še nekaj prednosti. Toplotna slika je zagotovo dobrodošla lastnost za lažje razumevanje in vrednotenje rezultatov. Kot smo lahko videli v primeru hierarhičnega razvrščanja s konsenzom nad podatkovno množico iris.tab, je pri uporabi slike potrebna pazljivost. Slika je lahko prav tako zavajujoča kot rezultati razvrščanja. Druga uporabna funkcionalnost je določanje števila skupin. Če razvrščamo podatke z algoritmom, ki tega nima implementiranega si lahko pomagamo z razvrščanjem s konsenzom. Lahko tako da uporabimo katerega od že implementiranih razvrščevalnih algoritmov ali pa implementiramo nov algoritem. Pri določanju števila skupin smo opazili eno pomanjkljivost. Rezultata ne dobimo v obliki ene vrednosti, ampak moramo upoštevati več različnih rezultatov. Pri tem vidimo možnost za izboljšavo; vse parametre, na podlagi katerih se odločimo za število skupin, bi lahko združili v eno vrednost, ki nam bi jo algoritem vrnil.

Literatura

[1] S. Monti, P. Tamayo, J. Mesirov, T. Golub, "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data," *Machine Learning 52*, 91-118, 2003.

[2] P.-N. Tan, M. Steinbach, V.Kumar, "Introduction to Data Mining," *Data Mining*, Boston: Pearson Addison Wesley, str. 487-643, 2006.

[3] K. Y. Yeung, W. L. Ruzzo, Details of the Adjusted Rand index and Clustering algorithms, Supplement to the paper "An Empirical Study on Principal Component Analysis for Clustering Gene Expression Data", *Bioinformatics*, 17, 763-774, May 2001.

Dodatek A

Testni primer 1

Tabela (A.1) vsebuje podatke množice testnih primerov, nad katerimi lahko izvedemo razvrščanje. Množica vsebuje 15 testnih primerov, ki imajo po 2 atributa, poimenovana kar Atribut 1 in Atribut 2. Naloga razvrščevalnega algoritma je, da teh 15 primerov razporedi v manjše število skupin. Skupine morajo biti logično oblikovane glede na lastnosti elementov.
Če pogledamo elemente v tabeli, lahko opazimo, da 1. atribut ne določa no-

Tabela A.1: Nerazvrščeni elementi.

Atribut 1	Atribut 2	Skupina
0	1	?
1	1	?
2	1	?
3	1	?
4	1	?
5	2	?
6	2	?
7	2	?
8	3	?
9	3	?
10	3	?
11	3	?
12	3	?
13	3	?
14	3	?

benih opaznih skupin. Vrednosti linearno naraščajo od 0 do 14. Mogoče bi lahko elemente razporedili na sode in lihe, na tiste z vrednostjo pod 5 in tiste z večjo od pet ipd. Drugi atribut se zdi obetavnejši, nakazuje nam tri skupine elementov, tiste z vrednostjo 1, 2 in 3. Tudi pogled na oba atributa skupaj ne razkriva kakšne nove zveze med elementi, tako da lahko od razvrščevalnega algortima pričakujemo, da bo izoblikoval skupine glede na vrednosti atributa 2. Razvrščanje po Atributu 2 nam da rezultat, prikazan v tabeli (A.2).

Tabela A.2: Razvrščeni elementi.

Atribut 1	Atribut 2	Skupina
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
5	2	1
6	2	1
7	2	1
8	3	2
9	3	2
10	3	2
11	3	2
12	3	2
13	3	2
14	3	2

Dodatek B

Testni primer 2

Tabela (B.1) vsebuje podatke iste množice testnih primerov kot tabela (A.1), razlika je le v vrstnem redu, elementi so premešani.

Tabela B.1: Premešani nerazvrščeni elementi.

Atribut 1	Atribut 2	Skupina
12	3	?
13	3	?
0	1	?
1	1	?
7	2	?
8	3	?
2	1	?
6	2	?
3	1	?
4	1	?
9	3	?
10	3	?
5	2	?
11	3	?
14	3	?

Dodatek C

Rezultati testiranja

Tabela C.1: Rezultati razvrščanja iris.tab.

KMeans		Hierarhično			
S konsenzom	Klasično	S konsenzom		Klasično	
		2	3	2	3
0.722	0.561	0.561	0.725	0.561	0.725
0.722	0.427	0.561	0.665	0.561	0.725
0.722	0.737	0.561	0.725	0.561	0.725
0.722	0.737	0.561	0.725	0.561	0.725
0.722	0.720	0.561	0.725	0.561	0.725
0.722	0.737	0.561	0.665	0.561	0.725
0.722	0.700	0.561	0.725	0.561	0.725
0.722	0.682	0.561	0.725	0.561	0.725
0.722	0.737	0.561	0.665	0.561	0.725
0.722	0.720	0.561	0.725	0.561	0.725
Povprečni vrednosti		Povprečne vrednosti			
0.722	0.662	0.561	0.701	0.561	0.725

Tabela C.2: Rezultati razvršanja wine.tab.

KMeans		Hierarhično	
S konsenzom	Klasično	S konsenzom	Klasično
0.915	0.865	0.0	0.0
0.915	0.831	0.0	0.0
0.915	0.915	0.0	0.0
0.915	0.837	0.0	0.0
0.915	0.832	0.0	0.0
0.915	0.847	0.0	0.0
0.915	0.847	0.0	0.0
0.915	0.838	0.0	0.0
0.915	0.837	0.0	0.0
0.915	0.847	0.0	0.0
Povprečni vrednosti		Povprečni vrednosti	
0.915	0.850	0.0	0.0

Tabela C.3: Rezultati razvrščanja glass.tab.

KMeans		Hierarhično			
s konsenzom	klasično	s konsenzom		klasično	
		4	6	4	6
0.137	0.120	0.014	0.200	0.014	0.195
0.137	0.237	0.014	0.225	0.014	0.195
0.137	0.127	0.014	0.225	0.014	0.195
0.137	0.137	0.014	0.195	0.014	0.195
0.137	0.203	0.014	0.202	0.014	0.195
0.137	0.127	0.014	0.200	0.014	0.195
0.137	0.137	0.014	0.202	0.014	0.195
0.137	0.134	0.014	0.197	0.014	0.195
0.137	0.135	0.014	0.218	0.014	0.195
0.137	0.132	0.014	0.200	0.014	0.195
Povprečni vrednosti		Povprečne vrednosti			
0.137	0.150	0.014	0.206	0.014	0.195