Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Rok Gomišček

Prikaz in tolmačenje modelov nenegativne matrične faktorizacije

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: doc. dr. Tomaž Curk

Ljubljana, 2015

Izjava o avtorstvu magistrskega dela

Spodaj podpisani Rok Gomišček sem avtor magistrskega dela z naslovom:

Prikaz in tolmačenje modelov nenegativne matrične faktorizacije

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom doc. dr. Tomaž Curk
- so elektronska oblika magistrskega dela, naslov (slovenski, angleški), povzetek (slovenski, angleški) ter ključne besede (slovenske, angleške) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 15. septembra 2015 Podpis avtorja:

Kazalo

Povzetek

Atributi, s katerimi opisujemo primere v bazah podatkov, so pogosto zelo številni. Določanje resnično pomembnih atributov za klasifikacijo ter njihovih medsebojnih odvisnosti zato predstavlja velik izziv. Eden od načinov, kako zmanjšati dimenzionalnost prostora in določiti pomembne atribute in primere, je z uporabo nenegativne matrične faktorizacije. V magistrski nalogi smo najprej preučili osnove nenegativne matrične faktorizacije in nekaj načinov prikaza podatkov in faktorskih modelov v matrikah. Predlagamo nekaj načinov, kako prikazati in razumeti modele, pridobljene s faktorizacijo. Uspeh metod smo ovrednotili na nekaj podatkovnih zbirkah in ugotovili, da nam vsaka metoda razkrije uporabne informacije o modelu. Z gručenjem faktoriziranih matrik lahko dobimo čistejše gruče kot z gručenjem izvornih podatkov. S projekcijo primerov v prostor faktorjev lahko ugotovimo, kateri faktorji vplivajo na določene razrede. Če pa tej projekciji dodamo še atribute, lahko sklepamo še o povezavi med primeri in atributi izvornega prostora.

Ključne besede

nenegativna matrična faktorizacija, faktorski model, vizualizacija podatkov

Abstract

Attributes that describe data in the databases present themselves in large numbers. For this reason defining truly important attributes for classification and establishing their mutual dependence poses a significant challenge. One way of reducing the dimensionality of the space and defining important attributes and examples is by using non-negative matrix factorization. In this master thesis we first examined the basics of non-negative matrix factorization and a few ways of visualizing the data and factor models in matrices. We propose a few ways of presenting and understanding the models acquired with factorization. We evaluated the effectiveness of the methods on several databases and learnt that each method reveals useful information about a model. Clustering of the factorized matrices can produce purer clusters than clustering of the source data. By projecting examples to the factor space we can see which factors affect certain classes. Adding attributes to this projection makes it possible to deduce the link between the examples and the attributes of the source space.

Keywords

non-negative matrix factorization, factor model, data visualization

Poglavje 1

Uvod

Razvoj tehnologije je prinesel zbiranje in hranjenje podatkov različnih vrst in oblik. Vse, kar lahko merimo, se da shraniti v takšni ali drugačni obliki. Pogosta oblika shranjevanja podatkov so tabele, kjer ena os predstavlja primere, druga os pa njihove atribute. Če se izrazimo matematično, lahko te tabele poimenujemo matrike. Podatki so pogosto sestavljeni iz obilice atributov, kar otežuje ugotavljanje, kateri so zares pomembni in kako so povezani med seboj ter z razredom. Zato si za lažje razumevanje želimo imeti manj atributov, kar lahko dosežemo na različne načine. Eden od načinov je nenegativna matrična faktorizacija, saj z njo podatke skrčimo in izrazimo kot produkt dveh manjših matrik. Prva matrika, ki opisuje meta atribute, združi atribute in pove, kako se izražajo v primerih. Druga pa pove, koliko je nek atribut prisoten v danem faktorju. V tako zmanjšani dimenzionalnosti prostora lahko odkrivamo vzorce v podatkih ali napovedujemo vrednosti novih primerov.

Tovrstni prijemi so se večkrat pokazali za uspešne, vendar pa tolmačenje modelov še vedno predstavlja izziv. Iz povezav v nižje-dimenzionalnem prostoru faktorjev ne moremo enostavno razbrati povezav v izvornem prostoru. Eden od možnih načinov razumevanja nižje-dimenzionalnega prostora je iskanje takšnih projekcij podatkov, kjer so ločnice med primeri jasno začrtane. Želimo si namreč projekcije, kjer so gruče čim bolj ločene druga od druge in

hkrati čim bolj čiste.

V magistrski nalogi smo definirali in ovrednotili nekaj načinov za tolmačenje in prikazovanje modelov, pridobljenih s faktorizacijo nenegativnih matrik. Uporabili smo metodo VizRank v nižje-dimenzionalnem prostoru faktorjev. Z njo želimo iz množice možnih projekcij dobiti in oceniti najboljše in tako lažje ugotoviti, katere podmnožice faktorjev dobro razmejijo razrede v dani vizualizacijski metodi. Predlagamo način, kako na isti sliki poleg primerov pokazati še pomembne atribute, ki primere uspešno ločijo v izvornem prostoru.

V nalogi najprej na kratko predstavimo nenegativno matrično faktorizacijo in dva načina, kako jo izvesti. Nato opišemo nekaj načinov prikazovanja matrik in metodo za iskanje najboljših projekcij podatkov VizRank. V nadaljevanju predstavimo nove metode za tolmačenje modelov, ki smo jih razvili v sklopu magistrske naloge. V poglavju 4 poročamo o rezultatih, ki smo jih dobili tako z obstoječimi kot tudi z novimi metodami prikazovanja faktorskih modelov ter z njimi poskusili tolmačiti modele. Nalogo zaključimo s kratkim povzetkom ugotovitev.

1.1 Cilji in glavni prispevki

Glavni cilj naloge je določiti najustreznejše prikaze faktoriziranih modelov, ki bi uporabnikom omogočili boljše razumevanje modelov. Glavni prispevki dela so:

- Vrednotenje stabilnosti modelov in projekcij v odvisnosti od transformacije podatkov in ranga faktorizacije.
- Razvoj metode za vrednotenje projekcij modelov, pridobljenih z dvofaktorizacijo nenegativnih matrik.
- Razvoj načina za hkratno prikazovanje primerov in atributov v prostoru faktorjev.

- Razvoj metode za razpihovanje podatkov glede na dobljene faktorje in vrednotenje projekcij, dobljenih z njimi.

1.2 Metodologija

V magistrski nalogi smo v programskem jeziku `Python` [11] združili metode za matrično faktorizacijo, ki jih nudi knjižnica `Nimfa` [14] in metodo za ocenjevanje kakovosti atributov za vizualizacije `VizRank` [7], ki je del paketa `Orange` [3]. Za obdelavo matrik smo uporabili tudi knjižnico `numpy` [12]. Za izris grafov smo uporabili knjižnico `matplotlib` [4].

Razvite metode smo preizkusili na osmih podatkovnih zbirkah. Na izvornih podatkih smo merili oceno projekcij. Po faktorizaciji smo merili napako aproksimacije in oceno projekcij aproksimiranih podatkov. Faktorizirane matrike smo uporabili za hierarhično gručenje in iskanje zanimivih projekcij.

Poglavje 2

Metode, orodja in podatki

V magistrski nalogi združujemo nenegativno matrično faktorizacijo z vizualizacijo podatkov, zato v tem poglavju najprej opišemo osnove nenegativne matrične faktorizacije. Nato naredimo kratek pregled različnih načinov, kako lahko vizualiziramo podatke v matrikah. Predstavimo tudi dve orodji, ki smo ju uporabljali v nalogi in sicer eno za nenegativno matrično faktorizacijo in eno za ocenjevanje kakovosti vizualizacij. Na koncu še opišemo podatke, ki smo jih uporabili v raziskavi.

2.1 Matrična faktorizacija

Lee in Seung [9] sta primerjala delovanje nenegativne matrične faktorizacije (NMF) z analizo glavnih komponent (PCA) in vektorsko kvantizacijo (VQ) pri učenju delov objektov. Kar NMF ločuje od drugih dveh metod, je njena nenegativna omejitev. Zaradi tega se lahko NMF nauči predstavitve po delih, saj omogoča le seštevanje delov, ne pa tudi odštevanja. Ugotovila sta, da se vse tri metode naučijo predstaviti obraz kot linearno kombinacijo osnovnih slik, a z zelo različnimi rezultati. VQ ima za osnovo prototipe obrazov, osnova pri PCA so "eigenobrazi", nekateri so popačene verzije celotnih obrazov. NMF pa ima za osnovo lokalizirane lastnosti, ki jih lahko prepoznamo kot dele obraza.

2.1.1 Teoretična osnova

Pri dani nenegativni matriki $\boldsymbol{V}$ iščemo taki nenegativni matriki $\boldsymbol{W}$ in $\boldsymbol{H}$, da velja $\boldsymbol{V} \approx \boldsymbol{W} * \boldsymbol{H}$. Dano $n * m$ matriko $\boldsymbol{V}$, kjer je n število vrstic, m pa število stolpcev, faktoriziramo na $n * r$ matriko $\boldsymbol{W}$ in $r * m$ matriko $\boldsymbol{H}$. Rang faktorizacije r je praviloma manjši od n ali m, tako da sta $\boldsymbol{W}$ in $\boldsymbol{H}$ manjši od $\boldsymbol{V}$ in dobimo stisnjeno obliko izvorne matrike.

Kadar iščemo faktorizacijo, se moramo najprej odločiti za cenilno funkcijo, ki bo ocenila kvaliteto aproksimacije. Lee in Seung [8] predlagata kvadrat evklidske razdalje med matrikama $\boldsymbol{A}$ in $\boldsymbol{B}$

$$\| \boldsymbol{A} - \boldsymbol{B} \|^2 = \sum_{ij} (\boldsymbol{A}_{ij} - \boldsymbol{B}_{ij})^2 \tag{2.1}$$

in divergenco

$$D(\boldsymbol{A} \parallel \boldsymbol{B}) = \sum_{ij} (\boldsymbol{A}_{ij} \log \frac{\boldsymbol{A}_{ij}}{\boldsymbol{B}_{ij}} - \boldsymbol{A}_{ij} + \boldsymbol{B}_{ij}), \tag{2.2}$$

ki pa ni razdalja, saj ni simetrična. Divergenca se reducira v Kullback-Leiblerjevo divergenco, kadar velja $\sum \boldsymbol{A} = \sum \boldsymbol{B} = 1$, torej sta $\boldsymbol{A}$ in $\boldsymbol{B}$ normalizirani verjetnostni porazdelitvi. Obe funkciji imata spodnjo mejo pri 0, ki jo dosežeta natanko takrat, kadar velja $\boldsymbol{A} = \boldsymbol{B}$.

Problem faktorizacije lahko opišemo kot optimizacijski problem, kjer minimiziramo

$$\| \boldsymbol{V} - \boldsymbol{W}\boldsymbol{H} \|^2 \tag{2.3}$$

ali

$$D(\boldsymbol{V} \parallel \boldsymbol{W}\boldsymbol{H}) \tag{2.4}$$

z ozirom na $\boldsymbol{W}$ in $\boldsymbol{H}$.

Čeprav sta obe funkciji konveksni samo v $\boldsymbol{W}$ ali samo v $\boldsymbol{H}$, nista konveksni v obeh spremenljivkah in zato ne moremo pričakovati, da bomo našli globalni minimum, lahko pa najdemo lokalni minimum.

V originalnem članku avtorja predlagata posodobitvena pravila in dokaz o konvergenci, a so izven namena te naloge, zato tiste, ki bi jih zanimalo več o tem, vabimo k branju originalnega članka [8].

Brunet et al. [1] so uporabili NMF na podatkih o ekspresiji genov pri različnih oblikah raka. Njihov namen je bil odkriti metagene, ki jih definirajo pozitivne linearne kombinacije genov. Uporabili so matrike velikosti $n * m$ z n geni in m vzorci. V tem primeru matrika $\boldsymbol{W}$ predstavlja metagene, celica w_{ij} pa koeficient gena i v metagenu j. Podobno matrika $\boldsymbol{H}$ predstavlja ekspresijo metagenov v vzorcih, celica h_{ij} pa predstavlja raven ekspresije metagena i v vzorcu j. Z uporabo faktorizacije so uporabili matriko $\boldsymbol{H}$ za iskanje k gruč. Vzorec so dodelili gruči, ki ustreza najbolj izraženemu metagenu vzorca. Ugotovili so, da s svojo metodo gručenja po matriki $\boldsymbol{H}$ lahko uspešno identificirajo različne razrede v podatkih.

2.1.2 Natančnost aproksimacije

S povečevanjem ranga faktorizacije dobimo večji matriki $\boldsymbol{W}$ in $\boldsymbol{H}$, kar nam omogoča natančnejšo rekonstrukcijo podatkov. Kot smo že omenili, lahko razliko med aproksimiranimi in izvornimi podatki merimo na več načinov (dva možna načina sta bila predstavljena v enačbah 2.1 in 2.2, obstajajo pa še drugi). Pomembno je, da dobimo 0, kadar sta obe matriki enaki. Rang faktorizacije nam lahko predstavlja število skupin, ki jih iščemo v podatkih, poleg tega nizko število faktorjev poenostavi iskanje povezav, zato želimo rang obdržati na dovolj nizki vrednosti.

2.1.3 Orodje Nimfa

Nimfa [14] je odprtokodna knjižnica za Python, ki je namenjena nenegativnemu matričnemu faktoriziranju, ki sta jo razvila Žitnik in Zupan. Poleg optimizacij, ki sta ju predlagala Lee in Seung (enačbi 2.3 in 2.4), ima Nimfa implementirane še druge optimizacijske algoritme. Deluje tudi na redkih matrikah.

Vse vključene optimizacije so inkrementalne in začnejo z inicializacijo $\boldsymbol{W}$ in $\boldsymbol{H}$ matrik. Ker ima inicializacija teh dveh matrik pomembno vlogo pri kvaliteti faktorizacije in pohitri konvergenco, Nimfa vsebuje več različnih

inicializacijskih metod. Poleg tega lahko uporabnik sam določi vrednosti v začetnih $\boldsymbol{W}$ in $\boldsymbol{H}$ matrikah ali pa se odloči za popolnoma naključne vrednosti.

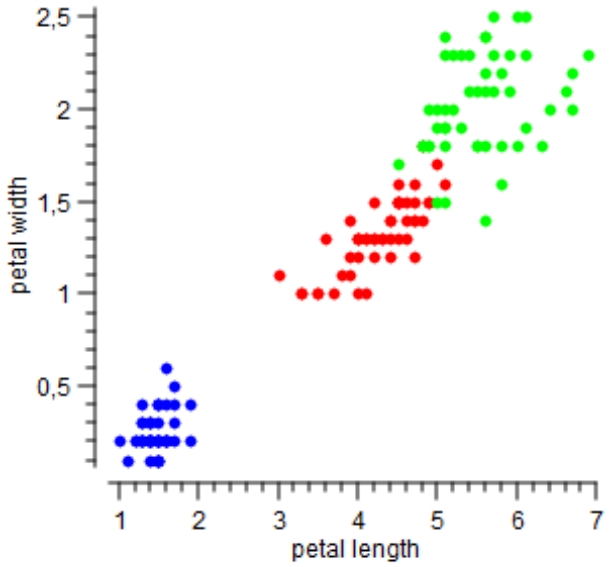
2.2 Osnovne vizualizacije matrik in faktoriziranih modelov

Podatke, shranjene v matrikah, lahko prikažemo na različne načine. V magistrski nalogi smo uporabljali prikaz matrik s toplotnimi kartami (angl. *heatmap*), kjer so celice pobarvane glede na svojo vrednost. Vrstice in stolpce v matrikah lahko premikamo, da postavimo podobne vrednosti oziroma barve skupaj. Poleg tega smo za projekcije primerov v dvodimenzionalnem prostoru uporabili še razsevni diagram in metodo ***radviz***.

2.2.1 Prikaz z gručenimi toplotnimi kartami

Toplotne karte prikažejo vsebino matrike z barvami. Glede na barvo celice lahko hitro ocenimo, ali je njena vrednost visoka ali nizka. Uporabljajo se že več kot stoletje, saj jih najdemo v statističnih zapisih iz druge polovice 19. stoletja. Uporabljali so barvno lestvico, na primer od bele (nizka vrednost) prek modre in rumene do rdeče (visoka). Vrstice in stolpce matrike lahko premikamo, da bi prikazali dodatne informacije. Tudi takšno sortiranje matrik ima dolgo zgodovino. Na začetku so ročno postavljali visoke vrednosti na diagonalo, danes pa z računalniško pomočjo izvajamo hierarhično gručenje po obeh oseh [13].

Z izvajanjem dvojnega hierarhičnega gručenja, torej tako po primerih kot po atributih, želimo dobiti takšno toplotno karto, kjer so podobne barve skupaj in tako izpostavili vzorce v podatkih. Za boljšo predstavljivost podobnosti med atributi oziroma primeri k izrisu dodamo še dendrograma, ki prikazujeta, koliko so primeri (oziroma atributi) oddaljeni med seboj.

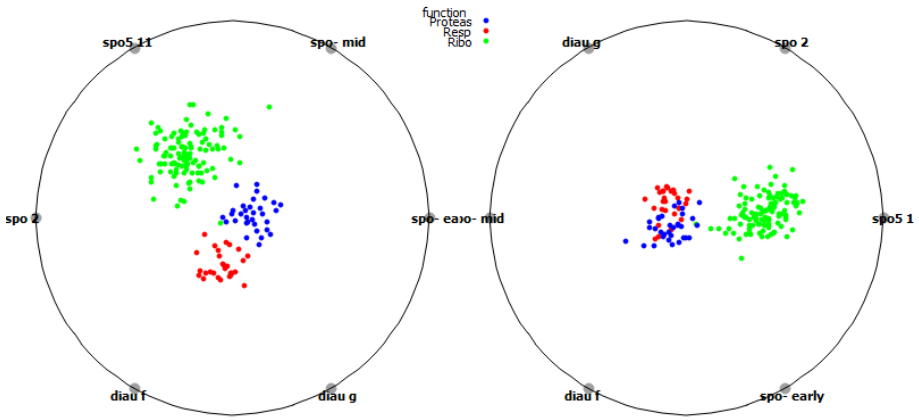


Slika 2.1: Prikaz primerov iz podatkovne zbirke `iris` z razsevnim diagramom.

2.2.2 Prikaz z razsevnim diagramom

Na razsevnih diagramih prikažemo primere kot točke v dvodimenzionalnem prostoru, za osi izberemo atribute. Razpon vrednosti različnih atributov bo lahko zelo različen, zato se moramo odločiti, kako ravnati v teh primerih. Ena rešitev je, da sta osi v različnih merilih, druga rešitev pa je skaliranje podatkov, s čimer pa izgubimo informacijo o dejanski vrednosti. Primer razsevnega diagrama vidimo na sliki 2.1, kjer so prikazani primeri za podatkovno zbirko `iris`. V tem primeru lahko vidimo, da sta osi v različnih merilih.

Koren et al. [5] so predlagali hkraten prikaz uporabnikov in filmov (torej primerov in atributov) na istem razsevnem diagramu z uporabo faktorjev. Svoj primer so ilustrirali s faktorjem, ki prepozna, komu je film namenjen glede na spol, in s faktorjem, ki opisuje resnost filma. Na tak način naj bi ugotovili, kaj je uporabnikom všeč in kateri filmi bi zadovoljili njihove želje. Niso pa ponudili odgovora na vprašanje, kako ravnati v primeru, ko ima isti faktor v različnih matrikah različne razpone vrednosti.

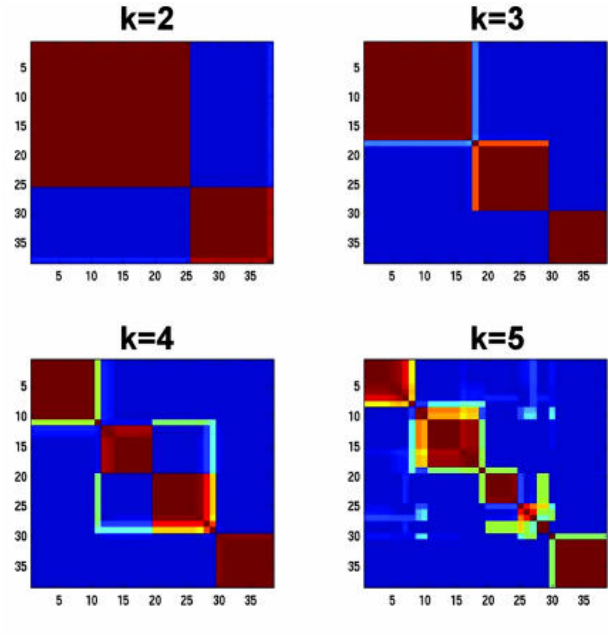


Slika 2.2: Prikaz primerov iz podatkovne zbirke `Brown selected` z uporabo ***radviza***. Oba primera uporabljata iste atribute za prikaz, a jih imata razporejene po krožnici v različnem vrstnem redu, kar vpliva na kakovost prikaza.

2.2.3 Radviz

Z razsevnim diagramom lahko prikažemo primere v dvodimenzionalnem prostoru v odvisnosti od dveh atributov. Pri zahtevnejših podatkih, kjer obstajajo zahtevnejše povezave med atributi, pa to ni dovolj. Zato se moramo poslužiti načina za dvodimenzionalno projekcijo z uporabo več atributov. En tak način je ***radviz***, oziroma radial coordinate visualization. Tukaj so atributi predstavljeni kot enakomerno oddaljene točke na krožnici, primeri pa kot točke znotraj krožnice. Pozicija primerov je odvisna od vrednosti njihovih atributov, pri čemer višja vrednost atributa močneje privlači primer k krožnici. [2]

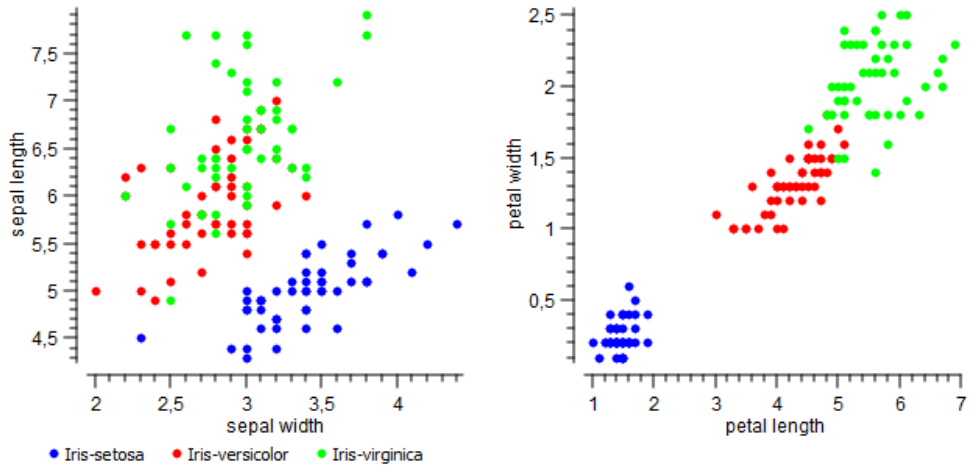
Na sliki 2.2 vidimo dva primera prikaza podatkov iz zbirke `Brown selected` z uporabo metode ***radviz***. Oba prikaza uporabljata iste atribute, le njihov razpored po krožnici je drugačen. V levem primeru vidimo, da so vsi trije razredi ločeni, v desnem primeru pa se dva razreda prepletata. To kaže na pomembnost vrstnega reda atributov pri tej metodi.



Slika 2.3: Matrika konsenza, ki povpreči 50 matrik povezljivosti primerov pri različnih rangih faktorizacije (k) za podatkovno zbirko `leukemia`. Vir slike: [1].

2.2.4 Prikaz in gručenje faktorjev

Brunet et al. [1] so v svojem eksperimentu gručenje izvajali na faktorizirani matriki $\boldsymbol{H}$. To so naredili tako, da so za vsak primer pogledali, kateri metagen ima najvišjo vrednost in ga razporedili v odgovarjajočo gručo. Gruče primerov so nato predstavili z matriko povezljivosti (angl. *connectivity matrix*) $\boldsymbol{C}$, dimenzije $m * m$, kjer je vrednost celice c_{ij} 1, če spadata primera i in j v isto gručo, v nasprotnem primeru pa je ta vrednost enaka 0. Po več izvajanjih faktorizacije so izračunali še matriko konsenza (angl. *consensus matrix*) $\bar{\boldsymbol{C}}$, ki je povprečje več matrik $\boldsymbol{C}$. Vrednosti te matrike so med 0 in 1 in odražajo verjetnost, da sta dva primera v isti gruči. Matrike so nato še preuredili, da so dobili bolj informativne prikaze. Rezultat tega postopka je prikazan na sliki 2.3.



Slika 2.4: Prikaz primerov iz podatkovne zbirke `iris` z uporabo razsevnega diagrama. Levo projekcijo je VizRank ocenil slabše kot desno, kjer so razredi bolje ločeni.

2.3 Metoda VizRank

Leban [6] je v svoji doktorski disertaciji opisal metodo VizRank, ki so jo razvili z namenom identifikacije in ocenjevanja zanimivosti različnih projekcij klasificiranih podatkov. Deluje na različnih vizualizacijski metodah, kot na primer razsevni diagram in ***radviz***. Zanimivost prikaza ocenjuje na podlagi ločenosti primerov v prikazu glede na razred. Primeri istega razreda morajo biti čim bolj skupaj in hkrati čim bolj oddaljeni od primerov drugih razredov.

Na sliki 2.4 vidimo dva prikaza podatkovne zbirke `iris` z razsevnim diagramom. Na levi strani slike je primer slabo ocenjene projekcije, na desni pa najbolje ocenjene. Vizrank nam pomaga, da izmed vseh možnih kombinacij atributov, ki jih lahko uporabimo pri projekciji, izberemo najboljšo.

2.3.1 Vrednotenje projekcij

VizRank [7] podatke najprej pretvori v dvodimenzionalne koordinate in ohrani vrednosti njihovega razreda. Na tako spremenjenih podatkih oceni napovedno točnost z izbranim učnim algoritmom in jo uporabi kot oceno zanimivo-

sti prikaza. To je smiselno, ker visoka točnost pomeni, da so razredi dobro ločeni med seboj, kar pripomore k boljšemu prikazu. VizRank kot rezultat vrne rangiran seznam projekcij, tako da lahko uporabnik sam izbere med najboljšimi.

Učni algoritem, ki ga VizRank uporablja za ocenjevanje zanimivosti projekcije, je metoda k-najbližjih sosedov (k-NN). Ta algoritem vrne verjetnostno porazdelitev razredov, ki jo izračuna glede na razrede k primerov, ki so novemu najbližji, pri čemer imajo primeri različne uteži, odvisno od razdalje do novega primera. Nov primer klasificira v razred z največjo verjetnostjo.

Da bi lahko govorili o najbližih sosedih, so morali najprej določiti način za merjenje razdalj. Zaradi številnih želenih matematičnih lastnosti in dobre aproksimacije človeškega ocenjevanja oddaljenosti so izbrali evklidsko metriko.

Morali so definirati tudi način, kako določiti parameter k. Eksperimentalno so ugotovili, da VizRank vrača boljše rezultate, če uporabijo višjo vrednost od klasične $k = \sqrt{N}$, kjer je N število primerov, in uporabijo na primer $k = N/c$, kjer je c število razredov. Za pohitritev iskanja k-najbližjih sosedov so uporabili k-dimenzionalna (k-D) drevesa.

Za ocenjevanje uspešnosti klasifikatorja so se odločili za povprečno verjetnost $\bar{P}$, ker upošteva napovedane verjetnosti in ne samo napovedane vrednosti razreda. Uporabo te mere priporočajo bolj kot uporabo Brierjeve ocene, saj jo je lažje interpretirati. Klasifikacijske točnosti pa ne priporočajo, saj gleda samo na pravilnost klasifikacije, ne upošteva pa verjetnosti, ki jo klasifikator pripiše posameznim razredom.

2.3.2 Preverjanje uspešnosti modelov

VizRank izračuna oceno klasifikatorja po metodi prečnega preverjanja. Ta metoda razdeli primere na učno in testno množico. Klasifikator na učni množici zgradi model, nato napove vrednosti testne množice. Postopek se ponovi tolikokrat, da so vsi primeri enkrat v testni množici. Ko imajo vsi primeri poleg prave vrednosti razreda tudi napovedane verjetnosti razredov,

Podatki	št. primerov	št. atributov	št. razredov
`Iris`	150	4	3
`Brown selected`	186	79	3
`DLBCL`	77	7070	2
`Leukemia`	72	5147	2
`Lung`	203	12600	5
`MLL`	72	12533	3
`Prostata`	102	12533	2
`SRBCT`	83	2308	4

Tabela 2.1: Število primerov, atributov in razredov v podatkih, ki smo jih uporabili za raziskavo.

se lahko izračuna povprečna verjetnost $\bar{P}$ oziroma ocena projekcije.

2.4 Uporabljeni podatki

V namene testiranja pravilnega delovanja razvitih metod smo najprej uporabili manjšo in znano zbirko podatkov `iris`. Ta ima samo štiri atribute, ki primere lepo ločijo na tri razrede. Naslednja, nekoliko večja uporabljena zbirka podatkov je `Brown selected`. Zaradi manjkajočih vrednosti smo nad njimi morali pred uporabo izvesti še imputacijo. Obe zbirki podatkov smo dobili v paketu Orange [3].

Da bi čim bolje ocenili uspešnost uporabe VizRanka na faktoriziranih matrikah, smo se odločili uporabiti še podatke, na katerih so Mramor et al. [10] že uspešno ovrednoti VizRank. To je šest podatkovnih zbirk o izražanju genov pri različnih oblikah raka. Podatki so na voljo na spletni strani, ki služi kot dodatek njihovemu članku, kjer so tudi že pripravljeni za delo z Orangeom. V tabeli 2.1 so predstavljene uporabljene podatkovne zbirke s številom primerov, atributov in razredov.

Poglavje 3

Prikazovanje faktoriziranih modelov

V sklopu magistrske naloge smo razvili nekaj metod za pomoč pri tolmačenju in vizualizaciji faktoriziranih modelov. V tem poglavju najprej opišemo način, kako ovrednotiti stabilnost modelov. Nato opišemo način, kako ugotoviti, ali so lahko projekcije v nižje-dimenzionalnem prostoru bolj informativne od projekcij v izvornem prostoru podatkov. Predstavimo način, kako te projekcije lahko nadgradimo še s prikazom atributov, ki so se izkazali za pomembne pri klasifikaciji izvornih podatkov. Nazadnje opišemo še način, kako razcepiti atribute po faktorjih ter kako raziskati, ali lahko na tak način dobimo dobre projekcije primerov. Opišemo tudi kako tako spremenjene podatke uporabiti pri razumevanju pomena posameznih faktorjev in opisovanju prispevkov posameznih atributov.

3.1 Določanje stabilnosti faktoriziranih modelov in njihovih napovedi

Nenegativna matrična faktorizacija nam ne zagotavlja, da bomo vedno dobili isti faktorizirani matriki, zato smo stabilnost definirali glede na rezultate, ki jih dajejo modeli. Po eni strani smo gledali napako aproksimacije

pri večkratni faktorizaciji pri istem rangu ter kako se spreminja z višanjem ranga. Zanimalo nas je tudi, koliko so stabilne ocene projekcij, ki jih da VizRank pri faktoriziranju pri nekem rangu. Zato smo matrike večkrat faktorizirali in vsakokrat poiskali najboljšo projekcijo, ki jo VizRank najde na aproksimiranih podatkih.

3.1.1 Vpliv skaliranja

Kot smo že omenili, pri nenegativni matrični faktorizaciji moramo zadostiti nenegativnemu pogoju, ki pa ne velja za vse podatke. Zato smo iskali najboljši način, kako spremeniti podatke, da bi zadostil temu pogoju in ohranili najbolje ocenjene projekcije podatkov.

Psevdokoda 1 prikazuje postopek, s katerim smo transformirali podatke pred faktorizacijo. Najprej smo poskusili z istim premikom vseh vrednosti in sicer z odštevanjem matričnega minimuma. To je morda smiselno pri podatkih z atributi v isti metriki, na primer pri ekspresiji genov, ker tako ohranimo tudi relacije med atributi. Ta način pa ni smiseln, kadar atributi predstavljajo neprimerljive lastnosti, na primer težo in dolžino. V teh primerih je bolj smiselno dati vsak stolpec na isto lestvico. To smo naredili v dveh korakih. Najprej smo od vsakega stolpca odšteli njegov minimum. Tudi tako spremenjene podatke smo uporabili v raziskavi.

Skaliranje atributov smo nato dokončali z deljenjem stolpcev z njihovo novo najvišjo vrednostjo. Tako smo dosegli to, da so imeli vsi atributi isti razpon vrednosti, med 0 in 1.

Algorithm 1 Psevdokoda za transformacije podatkov

1: $X \leftarrow$ izvorni podatki
2: $X_{odmik} \leftarrow X - X.min$
3: $col\_min \leftarrow$ seznam najmanjših vrednosti stolpcev X
4: $X_{stolpci} \leftarrow X - col\_min$
5: $col\_max \leftarrow$ seznam najvišjih vrednosti stolpcev $X_{stolpci}$
6: $X_{skalirano} \leftarrow X_{stolpci}/col\_max$

3.1.2 Vpliv večkratnega izvajanja faktorizacije

Vrednosti v faktoriziranih matrikah $\boldsymbol{W}$ in $\boldsymbol{H}$ ter posledično v aproksimirani matriki so odvisne od inicializacije faktorizacije. Ker smo izvajali faktorizacijo z naključno inicializacijo, nas je zanimalo, koliko se med seboj razlikujejo dobljeni modeli. Stabilnost modelov smo ocenjevali na dva načina: z napako aproksimacije ter oceno projekcij.

Napako aproksimacije smo merili s kvadratom evklidske razdalje (enačba 2.1). Za to mero sem se odločil, ker je simetrična, poleg tega pa lahko divergenca vrne tudi neskončnost. Pri danem rangu smo večkrat ponovili faktorizacijo, vsakokrat zračunali aproksimirane podatke in jih primerjali z izvornimi. Postopek smo ponovili pri večih rangih, tako da smo dobili vpogled tudi v to, kako se napaka aproksimacije spreminja glede na rang faktorizacije.

Ker nas je zanimalo tudi, kako se spreminjajo projekcije, ki jih dobimo po faktorizaciji, smo pri izbranem rangu večkrat ponovili faktorizacijo in iskanje najboljše projekcije z VizRankom. Pri vseh podatkovnih zbirkah, z izjemo `iris`, smo za projekcije uporabili metodo ***radviz***, saj imajo veliko število atributov. Vse projekcije smo lahko primerjali s projekcijami, dobljenimi nad izvornimi podatki.

3.2 Prikazovanje pomembnih faktorjev za klasifikacijo

Zaradi dobrih rezultatov, ki so jih drugi raziskovalci dobili pri gručenju z uporabo faktoriziranih matrik, smo želeli ugotoviti, če lahko dobimo dobre ocene projekcij primerov z uporabo faktorjev v matriki $\boldsymbol{W}$. Ker v naših primerih sestavljajo matriko $\boldsymbol{W}$ meta atributi in imamo podane razrede, ki jim primeri pripadajo, smo lahko uporabili VizRank. Za način prikazovanja smo uporabili tako ***radviz*** kot tudi razsevni diagram, saj ***radviz*** deluje le, kadar prikazujemo vsaj tri atribute.

Za prikaz smo uporabili podatke, ki so bili pred faktorizacijo skalirani po

stolpcih. Za projekcijo smo uporabili matriko $\boldsymbol{W}$, kot jo vrne faktorizacija. Poleg tega smo tudi poskusili narediti projekcije s spremenjenimi vrednostmi meta atributov matrike $\boldsymbol{W}$ tako, da smo za vsak primer nastavili na 1 meta atribut z najvišjo vrednostjo, ostale pa na 0. Na tak način izvedemo še hitro gručenje, kjer faktorji predstavljajo gruče. Tako lahko ocenimo povezavo med posameznimi faktorji in razredi. Pri vizualizaciji tako spremenjenih podatkov nam razsevni diagram praktično služi za prikaz treh faktorjev, od katerih sta dva na konceh osi, tretji pa v izhodišču.

3.3 Hkratna vizualizacija primerov in atributov v prostoru faktorjev

V prejšnjem podpoglavju smo spoznali dva možna načina, kako prikazati primere v nižje-dimenzionalnem prostoru faktorjev. Zdaj pa želimo te prikaze nadgraditi še s prikazom, na katerem se v istem prostoru nahajajo atributi. Ker je atributov lahko zelo veliko in bi prikaz vseh preveč napolnil projekcijo, s čimer bi izgubili na njeni informativnosti, sem se odločil za prikaz samo tistih atributov, ki jih je VizRank označil kot pomembne pri najbolje ocenjeni projekciji izvornih podatkov.

Tudi tukaj smo uporabili transformacijo in dobili dve obliki podatkov. Prva oblika predstavlja takšne vrednosti matrike $\boldsymbol{H}$, kot jih je dala faktorizacija. Drugo obliko pa smo dobili s skaliranjem atributov na razpon med 0 in 1. Skupaj s prejšnjo ločitvijo smo za vsako podatkovno zbirko dobili štiri projekcije na vizualizacijsko metodo. S temi projekcijami smo želeli ugotoviti, koliko in kateri faktorji vplivajo na atribute, ki so pomembni za razločevanje med razredi ter ali so v tem prostoru vidne povezave med atributi in primeri.

3.4 Razcep podatkov na faktorje

Po faktorizaciji smo dobili matriki $\boldsymbol{W}$ in $\boldsymbol{H}$, ki zmnoženi dasta aproksimirano matriko $\boldsymbol{X}_a$. Če pa zmnožimo i-ti stolpec matrike $\boldsymbol{W}$ z i-to vrstico matrike

$\boldsymbol{H}$, dobimo del aproksimirane matrike, ki ustreza i-temu faktorju. Če vse te dele seštejemo dobimo aproksimirano matriko. Da dobimo razcep podatkov po faktorjih, pa moramo vse te dele zlepiti skupaj vodoravno. Tako dobimo matriko, ki ima $n * r$ stolpcev, kjer je n število stolpcev v izvorni matriki in r rang faktorizacije. Tako transformirana matrika torej ohrani število primerov in dobi r-krat več atributov. Pri tem smo pretvorjene atribute poimenovali po principu $atrfx$, kjer x označuje vrstni red faktorja, atr pa je izvorno ime atributa. Tako poimenovanje smo uporabili zato, da bi lažje primerjali pretvorjene atribute z izvirnimi.

S takšno transformacijo podatkov smo želeli doseči to, da dobimo celice z visoko vrednostjo, kjer sta delež atributa v meta atributu in višina meta atributa v primeru visoka, kjer pa je vsaj eden nizek, nizko vrednost. Cilja take transformacije sta bila dva. Prvi cilj je bil s tako spremenjenimi podatki iskati zanimive projekcije z VizRankom. Drugi cilj pa je bil ugotoviti, ali lahko na tak način lažje razložimo pomen posameznih faktorjev pri opisovanju posameznih atributov.

Poglavje 4

Rezultati

V tem poglavju podamo primere vizualizacij faktoriziranih modelov z gručenimi toplotnimi kartami, poročamo o stabilnosti modelov, prikažemo vizualizacije primerov v prostoru faktorjev, tem vizualizacijam dodamo še prikaz atributov in jih uporabimo za tolmačenje modelov.

4.1 Osnovne vizualizacije

Kot smo videli v podpoglavju 2.2.1, lahko podatke v matrikah gručimo po obeh oseh in glede na dobljene gruče premaknemo vrstice in stolpce, da bi dobili lepši, bolj informativen prikaz. V sklopu magistrske naloge sem se odločil izvesti hierarhično gručenje tako na obeh oseh izvornih podatkov, kot na obeh oseh faktoriziranih matrik $\boldsymbol{W}$ in $\boldsymbol{H}$ ter prikazati gručene toplotne karte za izvorno matriko $\boldsymbol{X}$, aproksimirano $\boldsymbol{X}_a$ ter faktorizirani matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Da bi prikazali razliko, smo želeli preurediti izvorne podatke, enkrat z gručami, dobljenimi na izvornih podatkih, enkrat pa z gručami, dobljenimi na matrikah $\boldsymbol{W}$ in $\boldsymbol{H}$. Aproksimirano matriko $\boldsymbol{X}_a$ smo želeli preurediti samo z gručami iz faktoriziranih matrik. A izkazalo se je, da je pri vseh podatkovnih zbirkah, razen pri dveh najmanjših, število atributov preveliko za prikaz vseh stolpcev. Zato smo matrike v teh primerih stisnili po stolpcih. To smo naredili tako, da smo najprej izračunali dendrogram in ga nato uporabili za

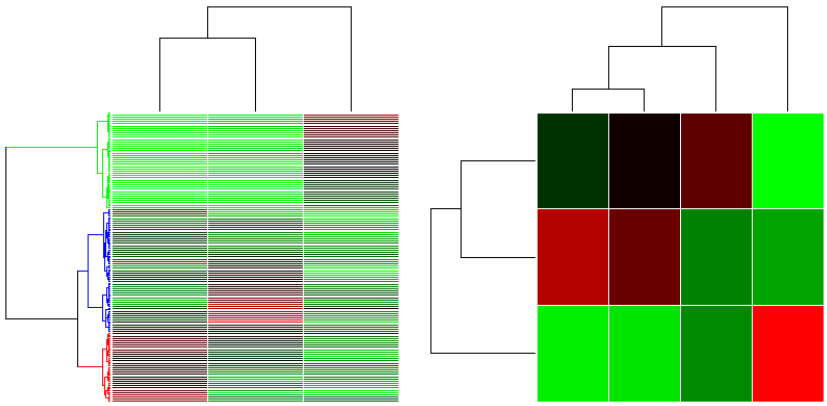
	$\boldsymbol{X}$	$\boldsymbol{W}$
iris	**88,67**	82,67
Brown selected	**95,16**	**95,16**
DLBCL	**77,92**	76,62
leukemia	65,28	**76,39**
lung	**77,34**	**77,34**
MLL	76,39	**81,94**
prostata	58,82	**60,78**
SRBCT	46,99	**62,65**

Tabela 4.1: Primerjava čistosti gruč, ki smo jih dobili s hierarhičnim gručenjem na izvornih podatkih in na matriki $\boldsymbol{W}$.

izračun povprečja podobnih stolpcev.

Pri hierarhičnem gručenju igra pomembno vlogo uporabljena razdalja in način povezovanja. V vseh naslednjih primerih smo uporabili evklidsko razdaljo. Način povezovanja smo prilagajali podatkovnim zbirkam. Vsako podatkovno zbirko smo hierarhično gručili po primerih s štirimi načini povezovanja in izračunali čistost gruč. Ko smo izbrali način povezovanja, ki nam da najčistejše gruče, smo vsako podatkovno zbirko hierarhično gručili še po atributih in izrisali dvojno gručene toplotne karte. Najboljši način povezovanja za hierarhično gručenje atributov smo izbrali glede na izrisano toplotno karto. Isti postopek smo uporabili tudi za gručenje matrik $\boldsymbol{W}$ in $\boldsymbol{H}$. V tabeli 4.1 so prikazane čistosti gruč v izvorni matriki $\boldsymbol{X}$ in faktorizirani matriki $\boldsymbol{W}$. Vidimo lahko, da smo v večini primerov dobili čistejše gruče v faktoriziranih matrikah $\boldsymbol{W}$.

Začeli bomo z manjšimi podatki, kjer lahko prikažemo celotne matrike. Na sliki 4.1 sta prikazani faktorizirani matriki podatkov `iris`, na levi strani $\boldsymbol{W}$, na desni $\boldsymbol{H}$. Na matriki $\boldsymbol{W}$ je vidna podobnost med dvema faktorjema ter razlika med njima in tretjim faktorjem. Na sliki 4.2 so še vedno prikazane gručene toplotne karte za `iris`, tokrat za celotne podatke. Na levi strani

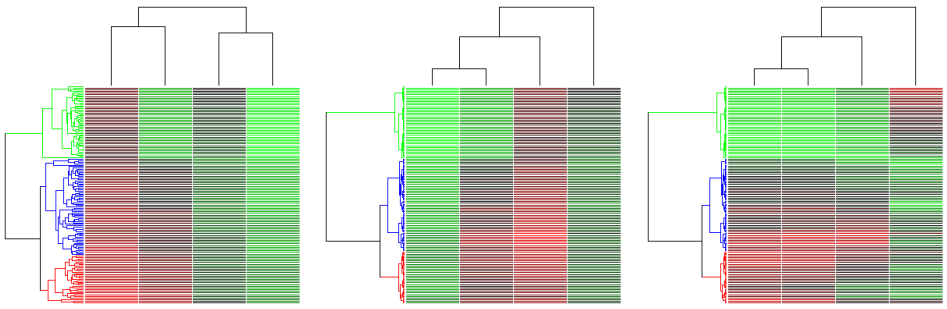


Slika 4.1: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `iris`. Levo je matrika $\boldsymbol{W}$, desno pa matrika $\boldsymbol{H}$.

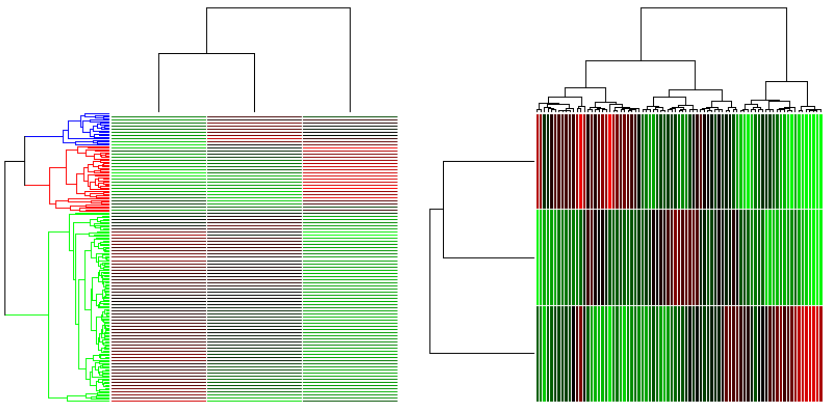
so hierarhično gručeni izvorni podatki. Na sredini so izvorni podatki, ki so hierarhično gručeni po primerih glede na matriko $\boldsymbol{W}$ in po atributih glede na matriko $\boldsymbol{H}$. Na desni pa so aproksimirani podatki, ki so hierarhično gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Gručenje po atributih je tu precej drugačno, po primerih pa precej podobno. S pomočjo srednje in desne slike lahko vidimo razliko med izvornimi in aproksimiranimi podatki.

Tudi podatkovna zbirka `Brown selected` je dovolj majhna, da še lahko prikažemo celotne matrike. Najprej sta na sliki 4.3 prikazani faktorizirani matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Hierarhično gručenje po primerih nam je dalo tri gruče, ki imajo dobro vidne razlike med faktorji. Pri hierarhičnem gručenju po atributih vidimo dve večji gruči. Na sliki 4.4 so prikazani celotni podatki. Na levi toplotni karti vidimo, da smo tudi na izvornih podatkih dobili dve gruči atributov. Iz slike je razvidno, da je gruča, ki je prikazana na desni strani toplotne karte, pomembna za ločevanje primerov v gruče. S pomočjo sredinske in desne toplotne karte se že na prvi pogled opazi razliko med izvornimi in aproksimiranimi podatki, saj je videti, da aproksimirani podatki bolj poudarijo nizke ter visoke vrednosti in s tem barve prikaza.

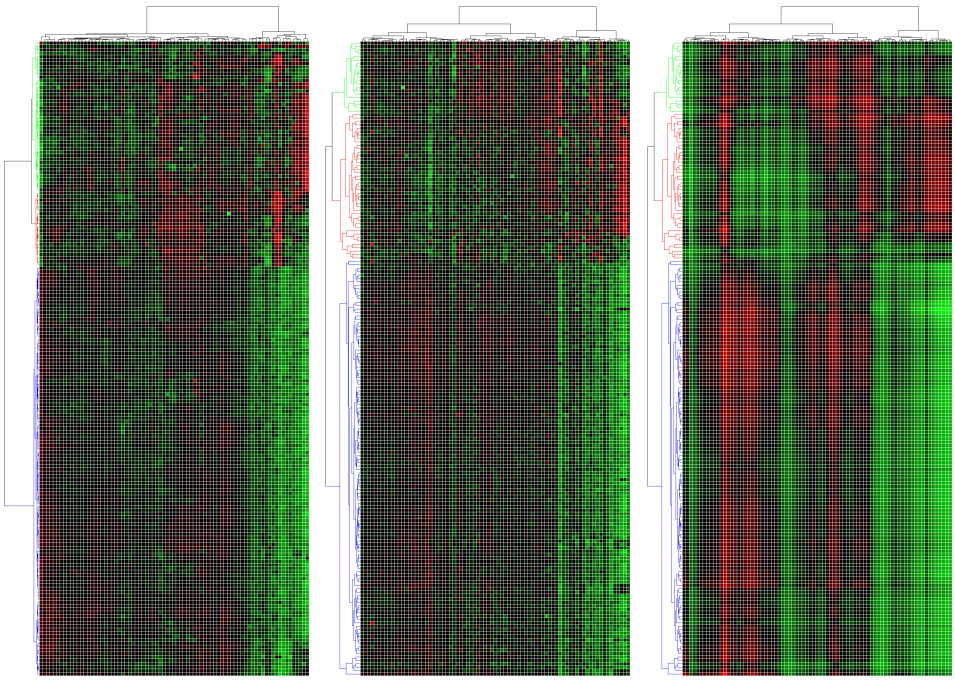
Pri ostalih podatkovnih zbirkah je bilo število atributov preveliko, da bi lahko dobro prikazali tako celotne podatke kot matriko $\boldsymbol{H}$. Matrika $\boldsymbol{W}$ ima



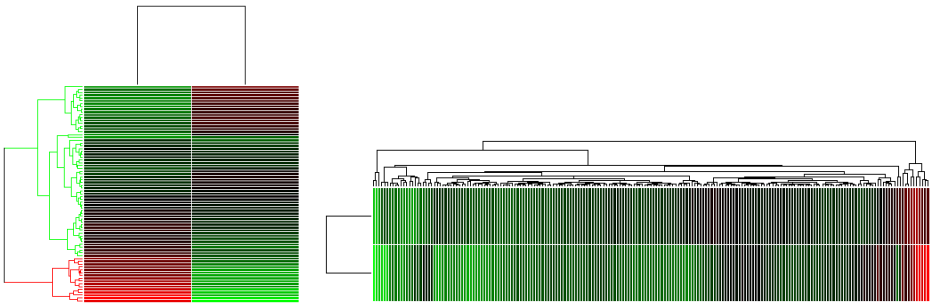
Slika 4.2: Levo so izvorni gručeni podatki `iris`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Desno pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.



Slika 4.3: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `Brown selected`. Levo je matrika $\boldsymbol{W}$, desno pa matrika $\boldsymbol{H}$.



Slika 4.4: Levo so izvorni gručeni podatki `Brown selected`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Desno pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.



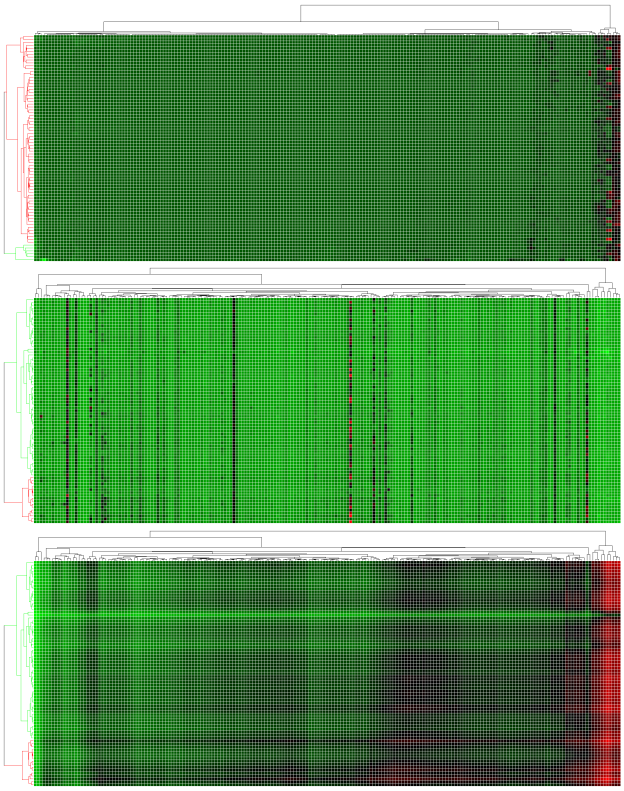
Slika 4.5: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `DLBCL`. Levo je matrika $\boldsymbol{W}$, desno pa matrika $\boldsymbol{H}$.

zelo zmanjšano dimenzionalnost atributov in je tako še vedno primerna za prikaz. Stolpce matrik $\boldsymbol{X}$, $\boldsymbol{X}_a$ in $\boldsymbol{H}$ smo stisnili in jim s tem zmanjšali dimenzionalnost, kar nam je omogočilo, da jih prikažemo.

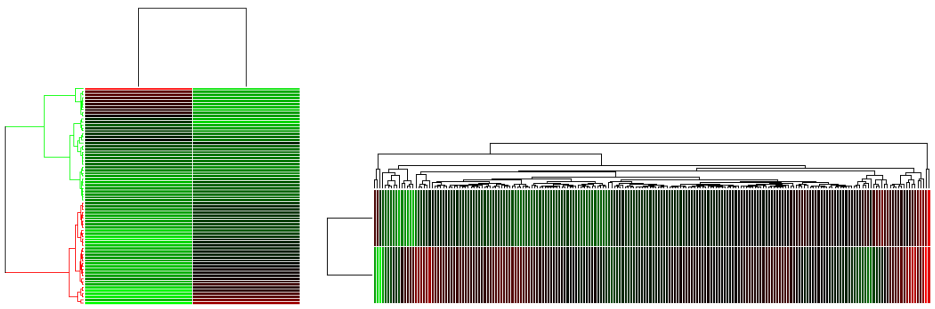
Hierarhično gručenje primerov na matriki $\boldsymbol{W}$ podatkovne zbirke `DLBCL` (slika 4.5) nam da eno manjšo in eno večjo gručo. Ker imamo samo dva faktorja, ne moremo govoriti o gručenju faktorjev, lahko pa lažje opazujemo razlike med njima. Te so predvsem očitne na zgornjem in spodnjem delu toplotne karte. Tam lahko opazimo, da je pri nekaterih primerih zelo izrazita razlika vrednosti med faktorjema. Na desni strani toplotne karte matrike $\boldsymbol{H}$ vidimo manjšo gručo atributov. Na sliki 4.6 je prikazana primerjava med izvornimi in aproksimiranimi podatki. Vidimo lahko, da stisnjeni izvorni podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$, izgubijo na informativnosti prikaza.

Na sliki 4.7 smo prikazali faktorizirani matriki za podatkovno zbirko `leukemia`. Iz toplotne karte matrike $\boldsymbol{W}$ se da razbrati, kateri faktor je pomembnejši za katero gručo. Na sliki 4.8 vidimo, da nam je hierarhično gručenje po primerih matrike $\boldsymbol{X}$ dalo drugačne gruče, kot gručenje po primerih matrike $\boldsymbol{W}$.

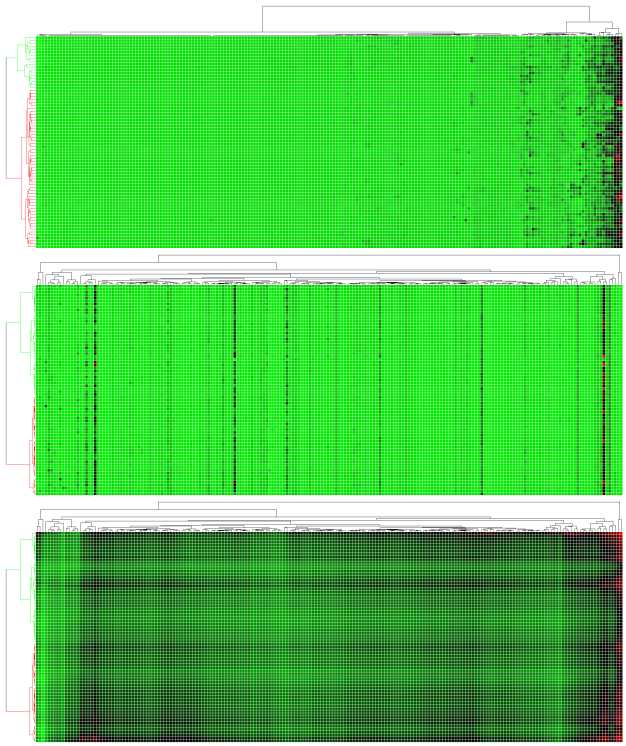
Podatkovna zbirka `lung` ima pet razredov, zato smo izvorno matriko faktorizirali z rangom faktorizacije pet. Dobljeni matriki $\boldsymbol{W}$ in $\boldsymbol{H}$ sta prikazani na sliki 4.9. Ker imamo več faktorjev, lahko opazujemo hierarhično gručenje na njih in opazimo, da bi jih lahko razdelili na dve gruči. Na sliki 4.10 vidimo



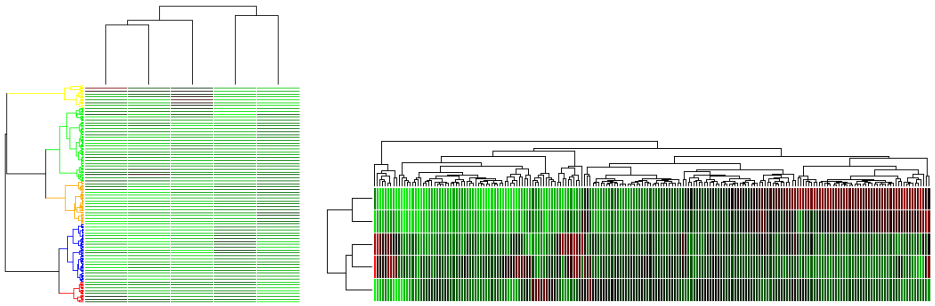
Slika 4.6: Zgoraj so izvorni gručeni podatki `DLBCL`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Spodaj pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.



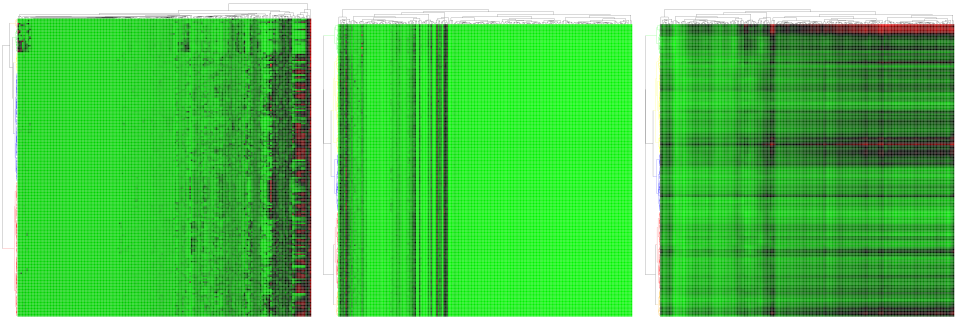
Slika 4.7: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `leukemia`. Levo je matrika $\boldsymbol{W}$, desno pa $\boldsymbol{H}$.



Slika 4.8: Zgoraj so izvorni gručeni podatki `leukemia`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Spodaj pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.



Slika 4.9: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `lung`. Levo je matrika $\boldsymbol{W}$, desno pa $\boldsymbol{H}$.

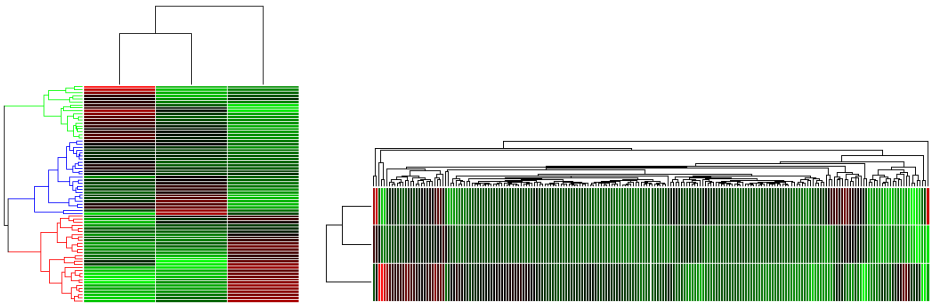


Slika 4.10: Levo so izvorni gručeni podatki `lung`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Desno pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.

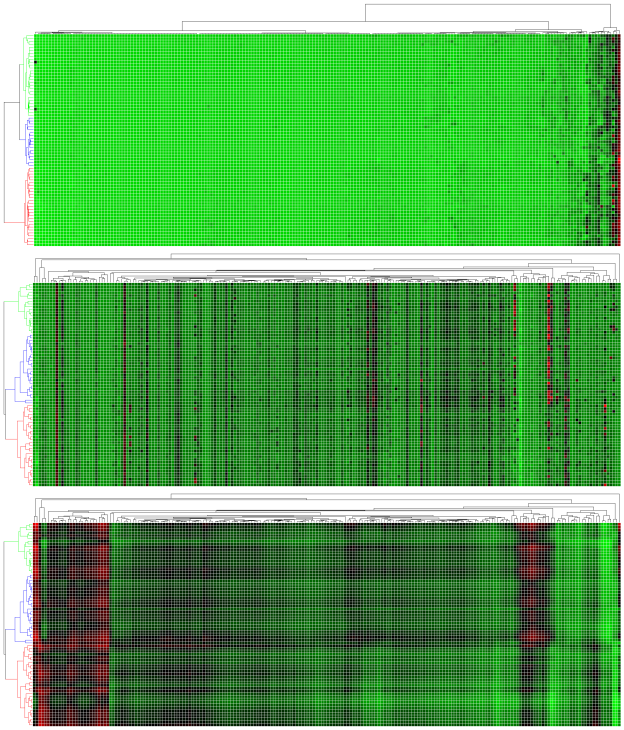
primerjavo med izvornimi in aproksimiranimi podatki. Iz gručene toplotne karte izvornih podatkov lahko razberemo, kateri atributi so pomembni za klasifikacijo, iz aproksimiranih podatkov pa ne.

Iz faktorizirane matrike $\boldsymbol{W}$ podatkovne zbirke `MLL` (slika 4.11) lahko razberemo, kateri faktor je najpomembnejši za posamezno gručo. Predvsem je lepo vidna meja med drugo in tretjo gručo. Na sliki 4.12 vidimo, da imajo aproksimirani podatki več atributov, ki so pomembni za klasifikacijo.

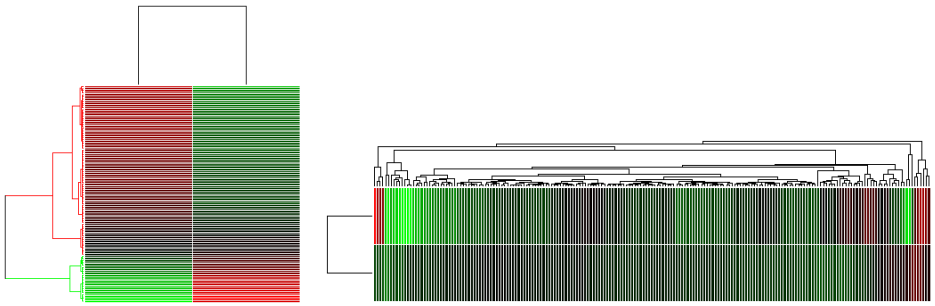
Pri podatkovni zbirki `prostata` dobimo po faktorizaciji dve lepo ločeni gruči primerov. Na sliki 4.13 vidimo, da bi bila tretja gruča na območju, kjer sta oba faktorja primerljiva po vrednosti. V tej matriki pride zelo do izraza



Slika 4.11: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `MLL`. Levo je matrika $\boldsymbol{W}$, desno pa matrika $\boldsymbol{H}$.



Slika 4.12: Zgoraj so izvorni gručeni podatki `MLL`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Spodaj pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.

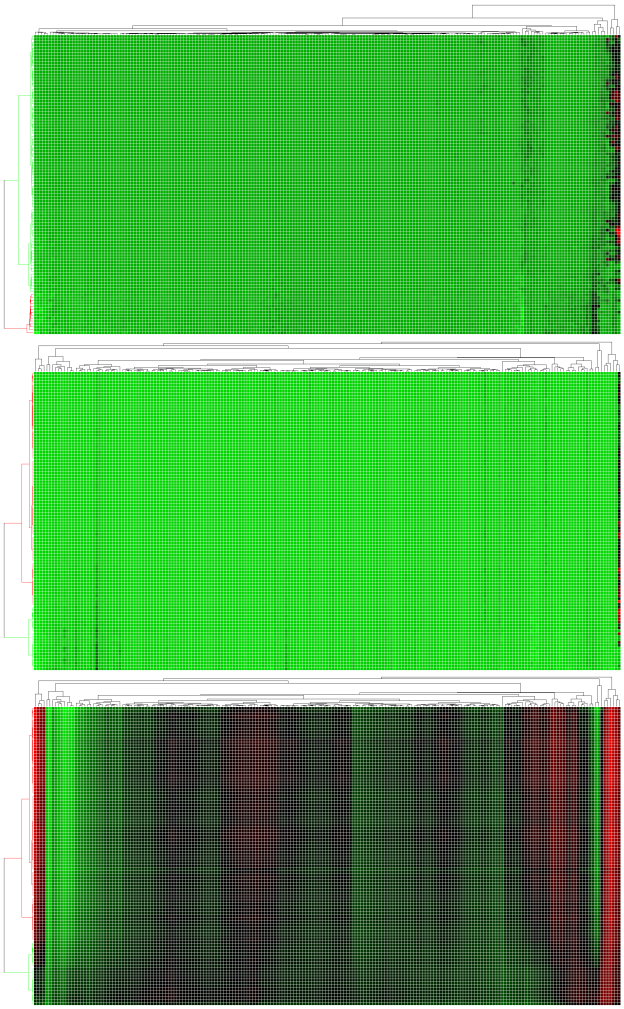


Slika 4.13: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `prostata`. Levo je matrika $\boldsymbol{W}$, desno pa matrika $\boldsymbol{H}$.

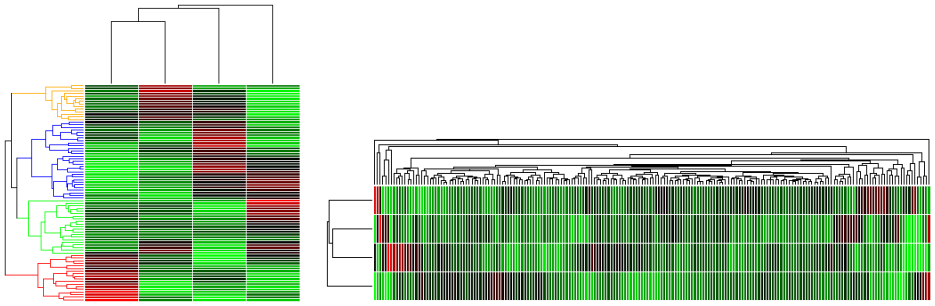
razlika med faktorjema. Na sredinski toplotni karti na sliki 4.14 vidimo, da se informacija o pomembnih atributih izgubi.

V gručeni toplotni karti faktorizirane matrike $\boldsymbol{W}$ podatkovne zbirke `SRBCT` na sliki 4.15 so se lepo izrazile štiri gruče primerov in opazimo lahko njihove povezave s faktorji. Po drugi strani pa so se faktorji gručili tako, da so se dodajali eni gruči. Iz toplotnih kart izvornih podatkov na sliki 4.16 ne izstopa nobena posebna oblika. Je pa toplotna karta aproksimiranih podatkov bolj informativna.

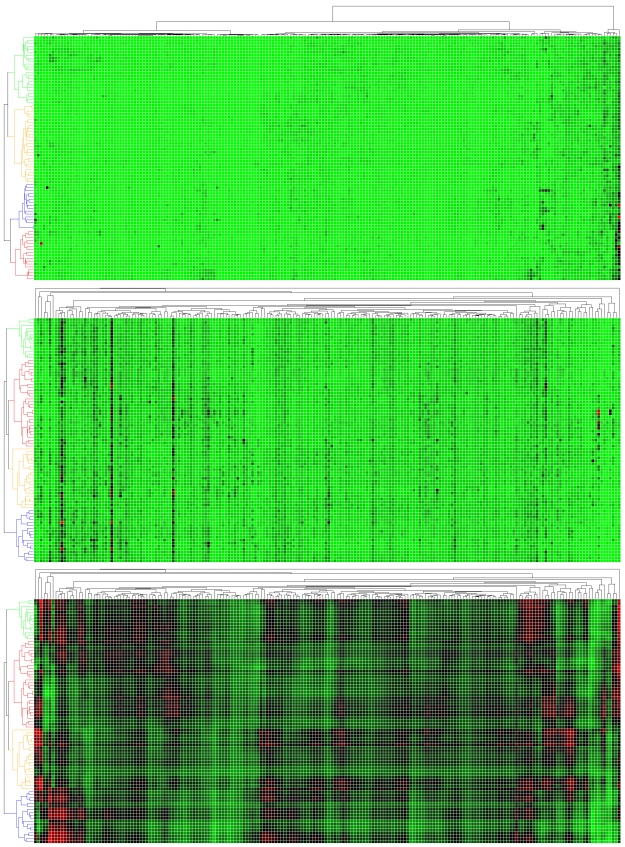
Ti primeri so pokazali, da lahko s hierarhičnim gručenjem v faktoriziranem prostoru dobimo čistejše gruče kot v izvornem prostoru. Pri stiskanju prevelikih matrik se nekatere podrobnosti izgubijo in je lahko zato prikaz izvornih in aproksimiranih podatkov manj zanimiv.



Slika 4.14: Zgoraj so izvorni gručeni podatki `prostata`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Spodaj pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.



Slika 4.15: Gručeni toplotni karti za faktorizirani matriki podatkovne zbirke `SRBCT`. Levo je matrika $\boldsymbol{W}$, desno pa matrika $\boldsymbol{H}$.



Slika 4.16: Zgoraj so izvorni gručeni podatki `SRBCT`. Na sredini so isti podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$. Spodaj pa so aproksimirani podatki, gručeni glede na matriki $\boldsymbol{W}$ in $\boldsymbol{H}$.

	izvorni	odmaknjeni	odmaknjeni po stolpcih	skalirani po stolpcih
iris	**94,00**	**94,00**	**94,00**	**94,00**
Brown selected	**98,87**	**98,87**	**98,87**	**98,87**
DLBCL	**93,34**	**93,34**	**93,34**	**93,34**
leukemia	**95,53**	**95,53**	**95,53**	**95,53**
lung	**83,01**	**83,01**	**83,01**	**83,01**
MLL	**95,92**	**93,91**	**93,91**	**93,91**
prostata	**91,86**	**91,86**	**91,86**	**91,86**
SRBCT	**97,02**	**97,02**	**97,02**	**97,02**

Tabela 4.2: Primerjava ocen projekcij primerov glede na začetno transformacijo podatkov.

4.2 Stabilnosti faktoriziranih modelov in napovedi

Ker vsebujejo nekateri izvorni podatki tudi negativne vrednosti, smo jih morali najprej transformirati. Zato smo najprej želeli ugotoviti, kakšen vpliv ima na faktorizacijo sprememba izvornih podatkov. V ta namen smo za vsako podatkovno zbirko štirikrat pognali VizRank: prvič nad izvornimi podatki, drugič nad podatki, ki smo jim odšteli minimum matrike, nad podatki, kjer smo stolpcem odšteli njihove minimume in na koncu še nad podatki s skaliranimi stolpci. Ti rezultati nam bodo kasneje služili kot osnovna ocena pri primerjanju ocen, dobljenih s faktorizacijo. Ker VizRank sam opravi transformacijo nad podatki, nismo pričakovali večjih razlik med različnimi transformacijami, kar se je izkazalo za pravilno. Tam, kjer je prišlo do razlik med ocenami, so te razlike bile majhne, saj so najboljše projekcije izbrale iste atribute. Rezultati so podani v tabeli 4.2.

V tem delu nas je zanimalo samo, kako predhodno spreminjanje podatkov vpliva na faktorizacijo, zato smo vse podatkovne zbirke faktorizirali pri

	izvorni	odmaknjeni	odmaknjeni po stolpcih	skalirani po stolpcih
`iris`	**94,81**	94,56	94,18	94,18
`Brown selected`	/	97,41	**98,75**	98,64
`DLBCL`	/	84,66	**89,28**	85,47
`leukemia`	/	93,94	**96,02**	95,10
`lung`	/	68,41	**90,34**	85,22
`MLL`	/	86,29	**95,53**	94,58
`prostata`	/	74,31	84,77	**85,08**
`SRBCT`	**98,33**	98,06	98,29	97,23

Tabela 4.3: Primerjava ocen projekcij primerov na aproksimiranih podatkih po faktorizaciji glede na začetno transformacijo podatkov.

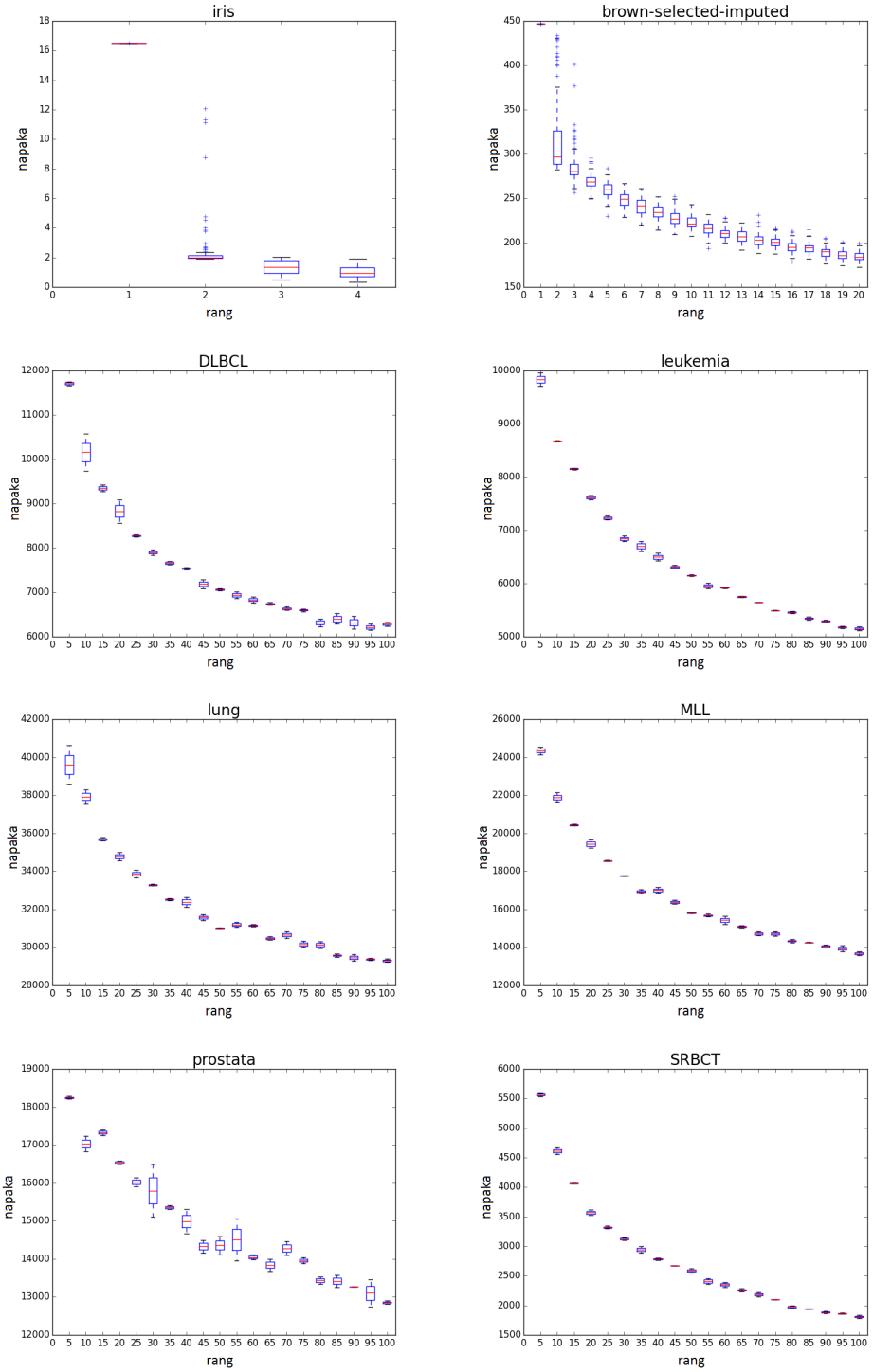
istem rangu, zračunali aproksimirano matriko in jo uporabili z VizRankom. Pri vseh podatkovnih zbirkah sem uporabil odmik, odmik po stolpcih in skaliranje po stolpcih. Kjer je bilo možno (ni negativnih vrednosti v matriki), pa smo uporabili tudi izvorne podatke. V tabeli 4.3 so prikazane najboljše ocene projekcij, ki smo jih dobili z VizRankom pri faktoriziranju pri rangu 200 z različnimi transformacijami podatkov. Kot lahko vidimo, so si tokrat ocene precej bolj različne.

Faktorizacija ne zagotavlja, da bomo dobili vedno enak rezultat. Zanimalo nas je, kako se spreminja faktorizacija ob večkratnem poganjanju z nespremenjenimi parametri. To smo ugotavljali na dva načina. Najprej smo pri različnih rangih opravili več faktorizacij in opazovali napako aproksimacije (kvadrat evklidske razdalje med izvornimi in aproksimiranimi podatki, enačba 2.1). Podatke smo pred faktorizacijo skalirali po stolpcih. Na sliki 4.17 vidimo, da napaka aproksimacije po pričakovanjih pada z rangom faktorizacije. Za podatkovni zbirki `iris` in `Brown selected` sem se odločil prikazati manjši razpon rangov faktorizacije, saj se je izkazalo, da pri obeh napaka aproksimacije zelo hitro pade in ostane skoraj nespremenjena. Pri

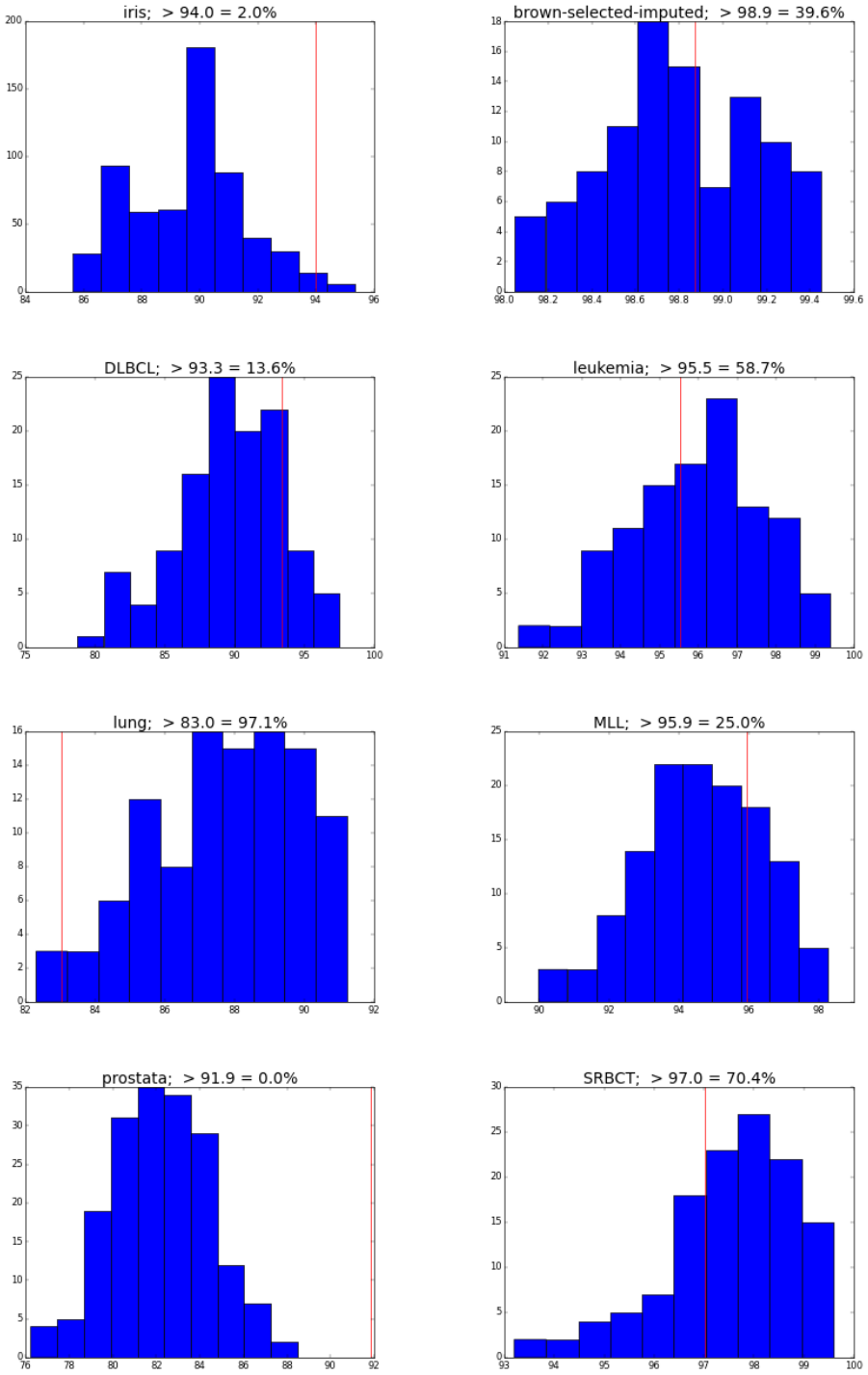
obeh lahko tudi vidimo, da pri rangu faktorizacije ena dobimo praktično vedno isto napako aproksimacije.

Nadalje nas je zanimal bolj podroben pogled na razporeditve ocen najboljših projekcij podatkov. Za to smo pri vsaki podatkovni zbirki vzeli predhodno pripravo podatkov in rang faktorizacije, kjer smo za dano zbirko podatkov predhodno našli najboljšo oceno in smo nato večkrat ponovili faktorizacijo pri istih pogojih. Na sliki 4.18 so prikazani histogrami razporeditev ocen za vse zbirke podatkov. Na slikah je z rdečo navpično črto prikazana ocena projekcije, ki jo je VizRank našel na izvornih podatkih. Razen v enem primeru je VizRank na aproksimiranih podatkih našel bolje ocenjene projekcije kot na izvornih podatkih. Pri vseh primerih vidimo porazdelitev, ki spominja na normalno.

To sta dva načina, kako lahko ocenimo stabilnost modela pri danih parametrih, oziroma koliko lahko variira njegova uspešnost. Ocenjevanje delovanja modela je vsekakor bolj informativno, a je lahko tudi časovno zelo potratno; odvisno od problema, ki ga rešujemo. Računanje napake aproksimacije je hitrejši način, vendar nam ne pove vsega.



Slika 4.17: Napaka aproksimacije v odvisnosti od ranga, vse podatkovne zbirke.



Slika 4.18: Porazdelitev ocen projekcij na aproksimirani matriki dobljeni po faktorizaciji. Rdeča navpična črta predstavlja oceno najboljše projekcije na izvornih podatkih. Nad posamezno sliko je zapisano ime podatkovne zbirke in delež projekcij, kjer je na aproksimiranih podatkih bil dosežen boljši rezultat kot na izvornih.

4.3 Primerjava pomembnosti atributov in faktorjev za klasifikacijo

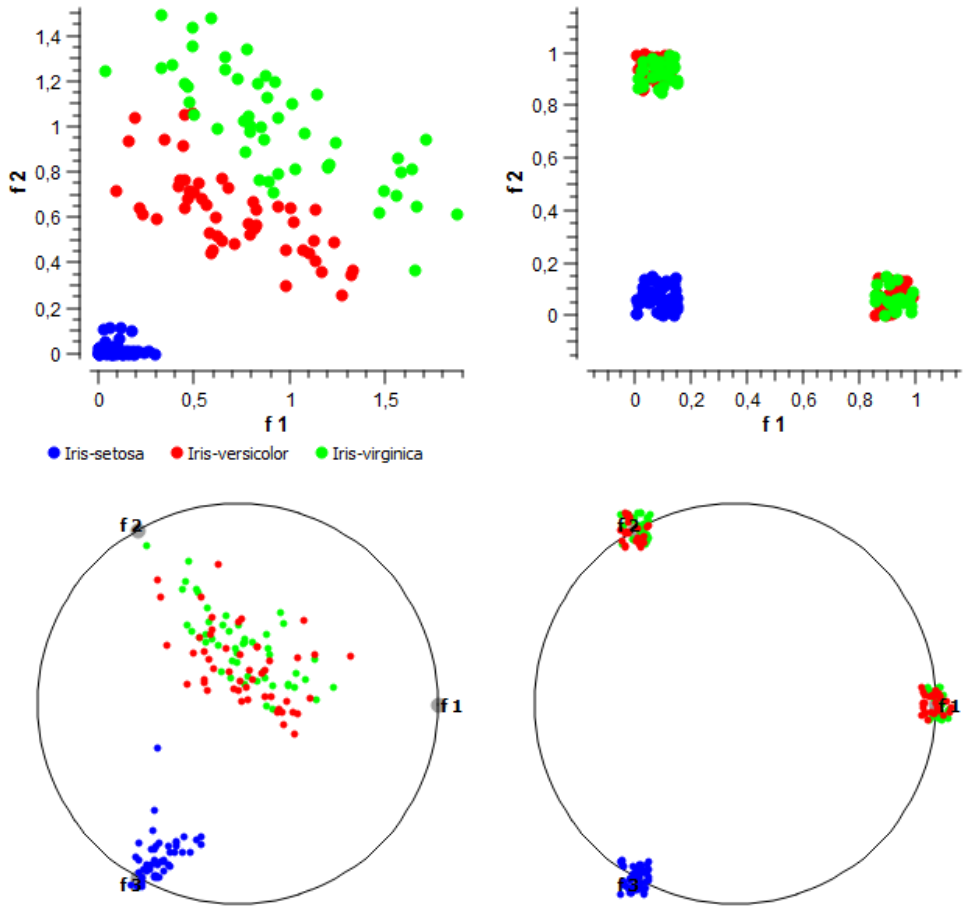
Pri ugotavljanju, kako dobro faktorji razdelijo primere glede na razrede pri projekcijah podatkov, je bilo seveda najprej potrebno faktorizirati matrike. Odločil sem se nastaviti rang faktorizacije r na število razredov v podatkovni zbirki. Motivacija za to odločitev je bila v tem, da bi idealno vsak razred imel posebno močno izraženega enega izmed faktorjev. Pri podatkih, kjer sta samo dva razreda in potemtakem dva ranga, smo rezultate prikazali samo z razsevnim diagramom, pri ostalih pa tudi z metodo ***radviz***. Psevdokoda 2 prikazuje, kako smo gručili podatke iz matrike $\boldsymbol{W}$. Primere smo gručili tako, da smo jih dali v gručo, ki odgovarja najmočnejšemu faktorju. V projekcijah smo uporabili tako izvorne kot gručene podatke. V nadaljevanju si bomo podrobneje pogledali projekcije posameznih podatkov.

Algorithm 2 Psevdokoda za gručenje W

1: $W, H \leftarrow faktorizacija(X, rang)$
2: $W_{gruceni} \leftarrow W$
3: **for** i od 0 do št. vrstic v W **do**
4: $\quad W_{gruceni}[i,:].max \leftarrow 1$
5: $\quad$ ostale vrednosti $W_{gruceni}[i,:] \leftarrow 0$
6: **end for**

Na sliki 4.19 so prikazane vse štiri projekcije za podatkovno zbirko `iris`. Zgornja leva projekcija je precej podobna projekciji na sliki 2.1, le da je meja med *iris-setosa* in *iris-versicolor* še močneje izražena. V ***radviz*** projekciji vidimo, da je tretji faktor močno povezan z razredom *iris-setosa*, ostala dva faktorja pa podobno vplivata na oba razreda, kar poslabša projekcijo. To se pokaže tudi pri obeh projekcijah z gručenimi primeri, kjer je ena gruča čista, drugi dve pa mešani. Med tema dvema razredoma lahko mejo povlečemo samo pri razsevnem diagramu s faktoriziranimi podatki, pri ostalih pa so primeri preveč prepleteni.

Tudi pri podatkovni zbirki `Brown selected` imamo tri razrede in tukaj



Slika 4.19: Projekcija primerov iz podatkovne zbirke `iris`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.

je najboljša projekcija z izvornimi podatki, saj pri ostalih dobimo mešane gruče (slika 4.20). ***Radviz*** projekcija je v tem primeru boljša od razsevnega diagrama, ker so primeri bolj ločeni glede na razred. Iz projekcij gručenih podatkov lahko vidimo, da je tretji faktor močno povezan z razredom *Ribo*, razred *Proteas* pa se nahaja večinoma pri drugem faktorju.

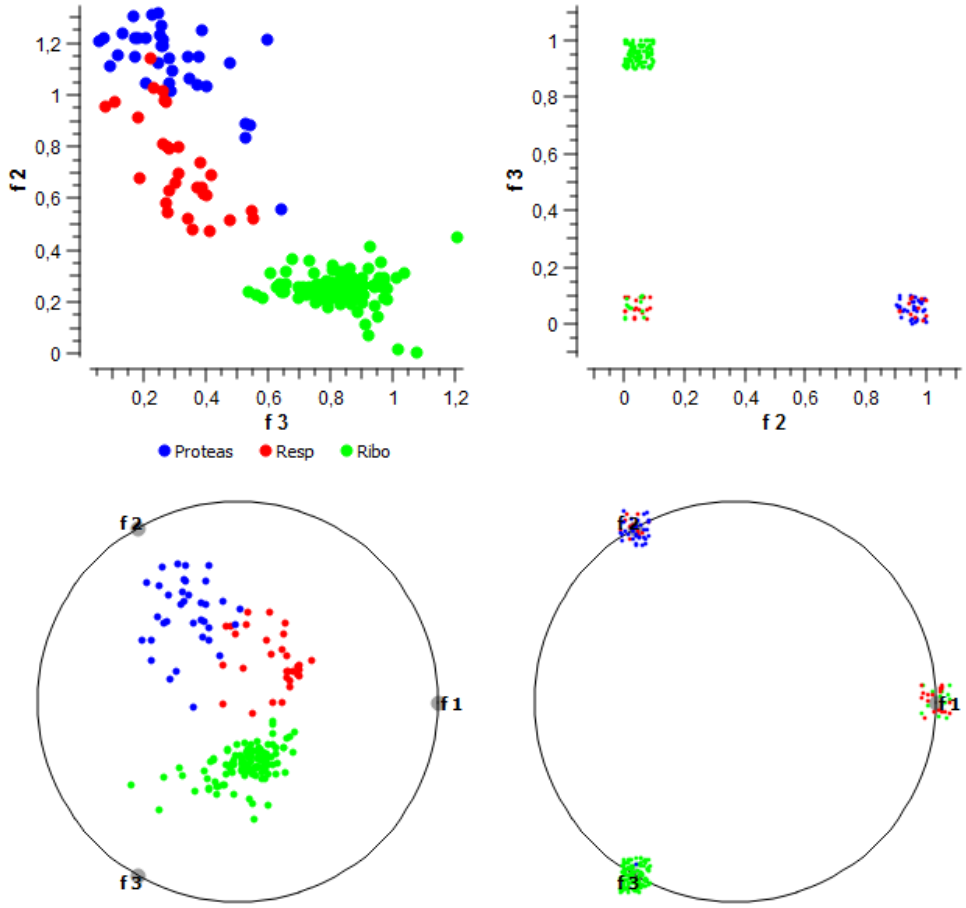
Pri podatkovni zbirki `DLBCL` imamo samo dva razreda, zato je na sliki 4.21 uporabljen samo razsevni diagram. Iz leve projekcije lahko slutimo, da razred *DLBCL* predstavlja nekakšno korelacijo med faktorjema, saj izgleda, da vrednost drugega faktorja pada, če narašča vrednost prvega faktorja. Drugi razred je po vrednostih na obeh faktorjih precej bolj omejen, a kljub temu po gručenju ne dobimo čistih gruč.

Na sliki 4.22 vidimo še eno podatkovno zbirko z dvema razredoma. V tem primeru pa nam gručeni podatki dajo eno zelo čisto gručo, saj je v njej samo en primer z drugačnim razredom. Druga gruča pa je precej bolj mešana. V takšnem slučaju bi dodaten faktor lahko pomagal razbiti mešano gručo na dve čistejši.

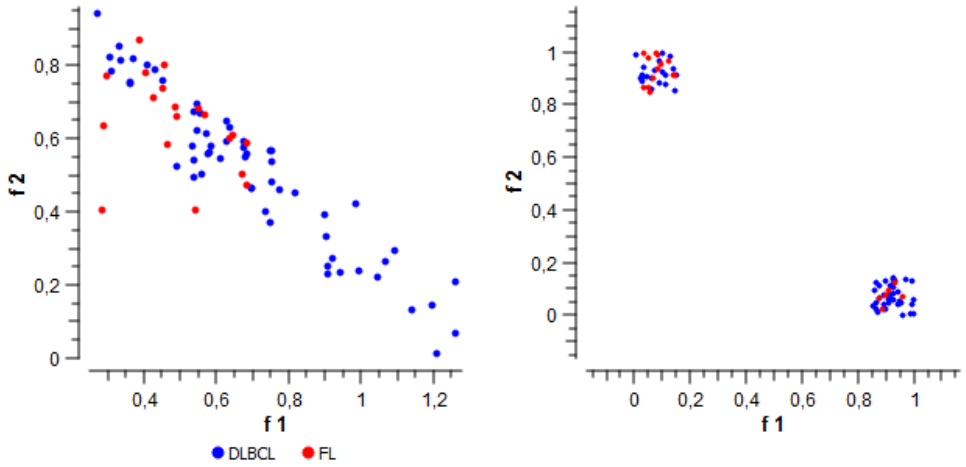
Pri podatkovni zbirki `lung`, ki je prikazana na sliki 4.23, imamo pet razredov. Iz zgornje leve projekcije bi lahko izpeljali nekaj ugotovitev, kot na primer, da so primeri, ki spadajo v razred *AD* v spodnji polovici slike, primeri iz *COID* v zgornji, ostali trije razredi pa večinoma levo spodaj. Ta ugotovitev se preslika tudi v projekcijo gručenih primerov z razsevnim diagramom, a so tri dobljene gruče še vedno zelo mešane. V spodnji desni sliki lahko opazimo, da je peti faktor najmočnejši samo pri razredu *AD*, da imajo primeri iz *NL* imajo večinoma najmočnejši četrti faktor, primeri iz *COID* pa imajo večinoma najmočnejši drugi faktor.

Ta primer je pokazal, da projekcija gručenih primerov z razsevnim diagramom ni primerna, kadar imamo več kot tri gruče. Kljub večjemu številu faktorjev pa nam projekcija izvornih podatkov matrike $\boldsymbol{W}$ z razsevnim diagramom lahko razkrije kakšno podrobnost o odnosu med dvema faktorjema.

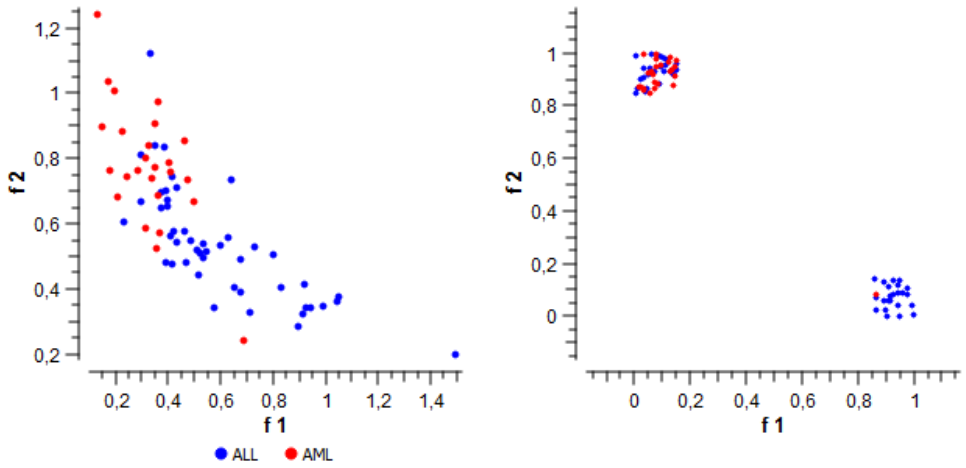
Pri projekcijah za podatkovno zbirko `MLL` na sliki 4.24 vidimo močno povezavo med tretjim faktorjem in razredom *AML*. To povezavo lahko zasle-



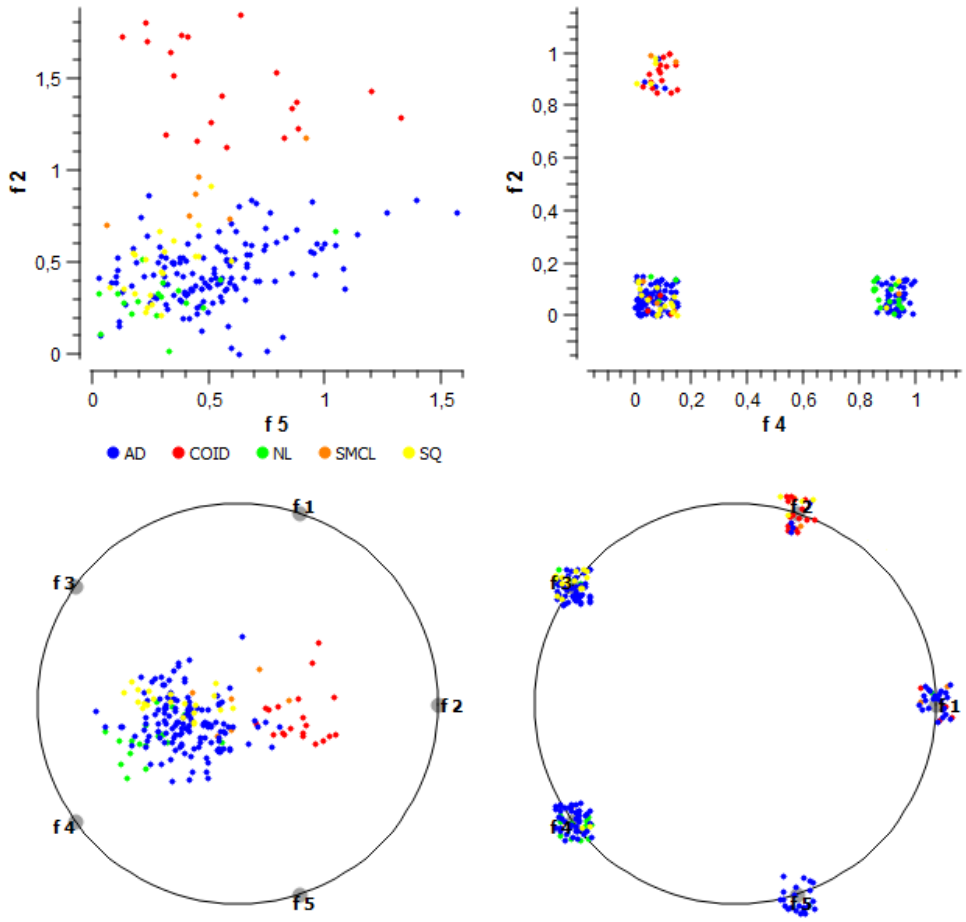
Slika 4.20: Projekcija primerov iz podatkovne zbirke `Brown selected`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.



Slika 4.21: Projekcija primerov iz podatkovne zbirke `DLBCL`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.



Slika 4.22: Projekcija primerov iz podatkovne zbirke `leukemia`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.

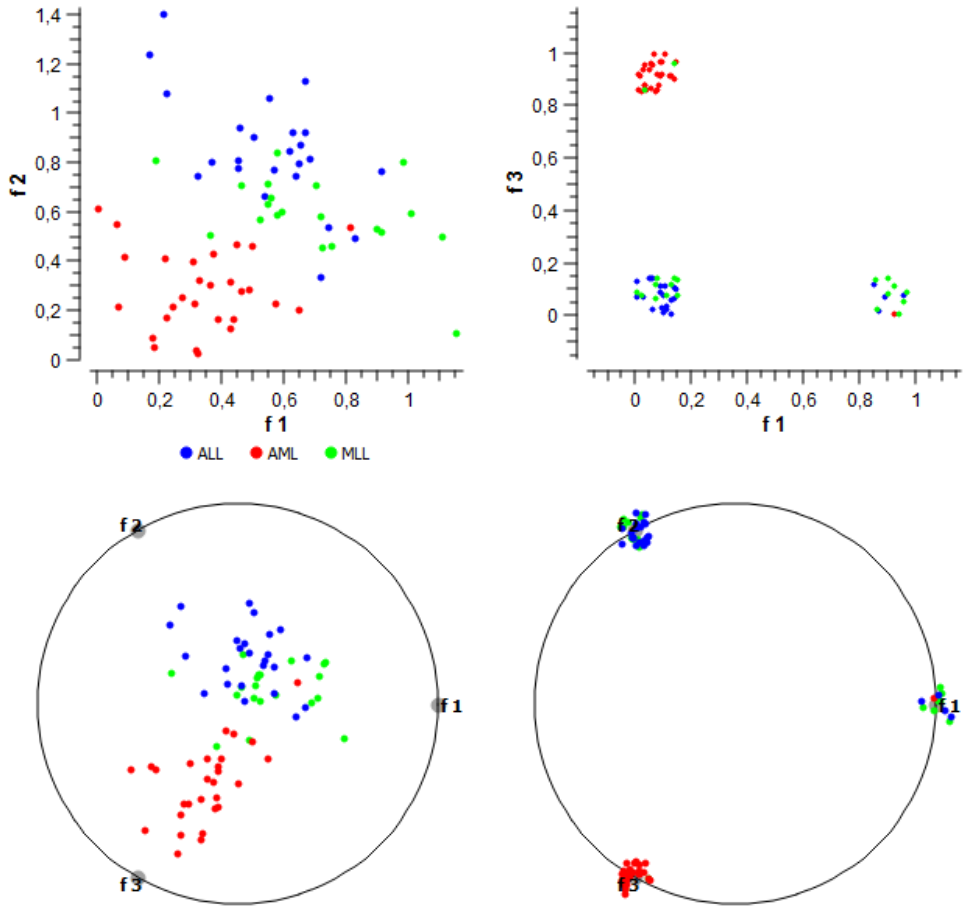


Slika 4.23: Projekcija primerov iz podatkovne zbirke `lung`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.

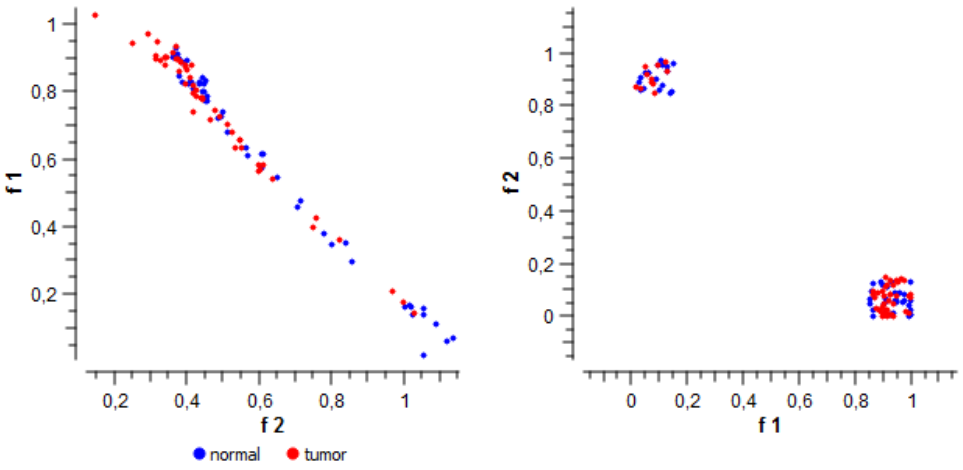
dimo pri vseh štirih načinih prikaza. Druga dva razreda pa sta v tem prostoru preveč prepletena, da bi se dalo dobro razlikovati med njima v katerem koli od naših prikazov.

Pri faktorizaciji podatkovne zbirke `prostata` smo dobili dva faktorja, ki imata zelo podoben vpliv na oba razreda. Na levi strani slike 4.25 vidimo, da sta faktorja korelirana, toda za oba razreda podobno, zato ne moremo ločiti primerov na razrede. Tudi na desni strani vidimo, da smo dobili dve zelo mešani gruči.

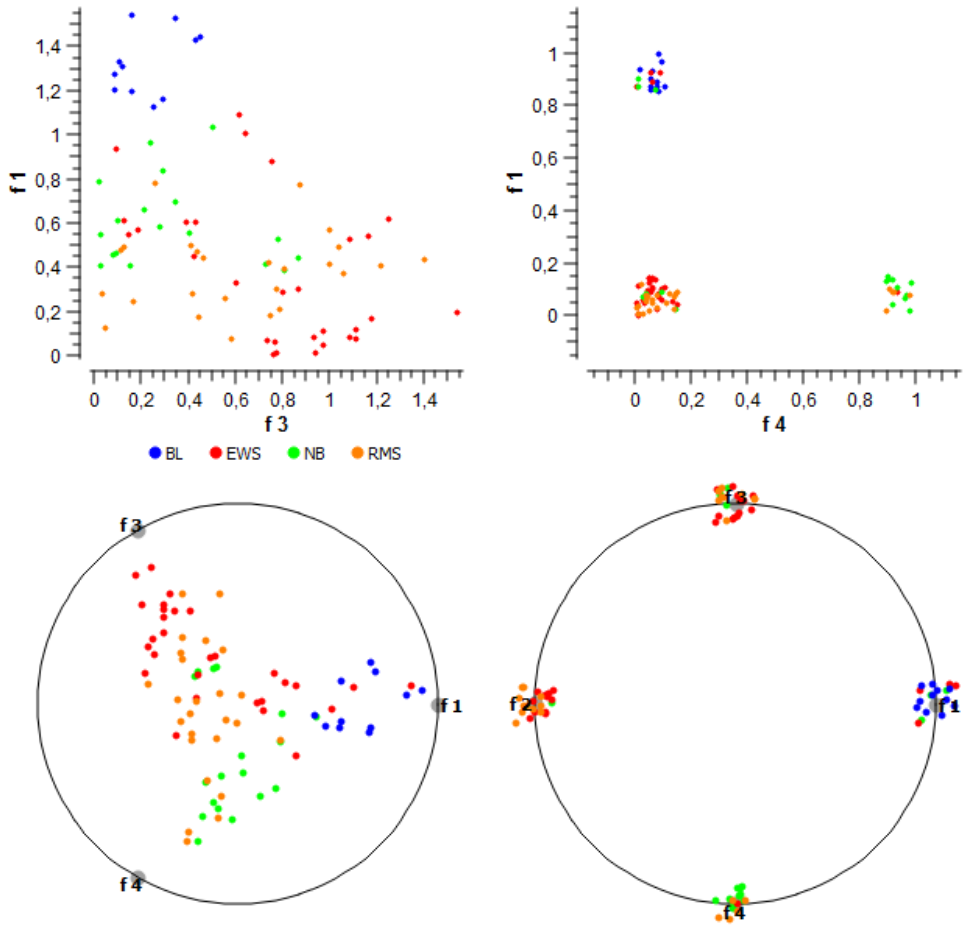
Na sliki 4.26 vidimo projekcije za podatkovno zbirko `SRBCT`. Opazimo, da so primeri iz razreda *BL* najbolj ločeni od ostalih, kar se najbolje vidi na zgornjem levem delu slike. Ponovno imamo več kot tri faktorje, zato je zgornja leva slika neuporabna. Pri ostalih projekcijah vidimo, da so v tem prostoru primeri zelo prepleteni.



Slika 4.24: Projekcija primerov iz podatkovne zbirke `MLL`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.



Slika 4.25: Projekcija primerov iz podatkovne zbirke `prostata`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.



Slika 4.26: Projekcija primerov iz podatkovne zbirke `SRBCT`. Levo so izvorni podatki matrike $\boldsymbol{W}$, desno po gručenju.

4.4 Hkratna vizualizacija primerov in atributov

Vizualizacijam primerov v prostoru faktorjev smo želeli dodati še prikaz atributov. Ker je v nekaterih podatkovnih zbirkah več tisoč atributov, smo se odločili za prikaz samo tistih atributov, ki so pomembni za klasifikacijo. Kateri so pomembni za klasifikacijo je določil VizRank.

Psevdokoda 3 prikazuje postopek priprave podatkov za hkraten prikaz primerov in atributov. Po faktoriziranju smo dobili matriko $\boldsymbol{H}$, iz katere smo najprej izbrali tiste stolpce, ki ustrezajo pomembnim atributom in matriko transponirali. Za prikaz v prostoru faktorjev smo uporabili tako izvorne podatke iz matrike $\boldsymbol{H}$, kot tudi skalirane.

Matriki $\boldsymbol{W}$ in $\boldsymbol{H}$ smo lahko navpično združili in v kombinaciji s prejšnjo metodo dobili štiri vrste združenih podatkov $\boldsymbol{WH}$: gručene-skalirane, gručene-neskalirane, negručene-skalirane in negručene-neskalirane. V takšnem vrstnem redu smo projekcije tudi prikazali na slikah, ki sledijo. Kot pri prejšnjih prikazih, smo tudi tokrat uporabili tako razsevni diagram kot metodo ***radviz***.

Algorithm 3 Psevdokoda za skaliranje H in združitev z W

1: $W, H \leftarrow faktorizacija(X, rang)$
2: $H \leftarrow$ pomembni atributi za klasifikacijo
3: $H \leftarrow H^T$
4: $col\_min \leftarrow$ seznam najmanjših vrednosti vrstic H
5: $H_{skalirano} \leftarrow H - col\_min$
6: $col\_max \leftarrow$ seznam najvišjih vrednosti vrstic $H_{skalirano}$
7: $H_{skalirano} \leftarrow H_{skalirano}/col\_max$
8: $W_{gruceni} \leftarrow$ psevdokoda 2
9: $WH_{gruceni,skalirani} \leftarrow$ navpično združi $W_{gruceni}$ in $H_{skalirano}$
10: $WH_{gruceni,izvorni} \leftarrow$ navpično združi $W_{gruceni}$ in H
11: $WH_{izvorni,skalirani} \leftarrow$ navpično združi W in $H_{skalirano}$
12: $WH_{izvorni,izvorni} \leftarrow$ navpično združi W in H

S tovrstnimi prikazi smo želeli ugotoviti, kateri atributi so še posebej

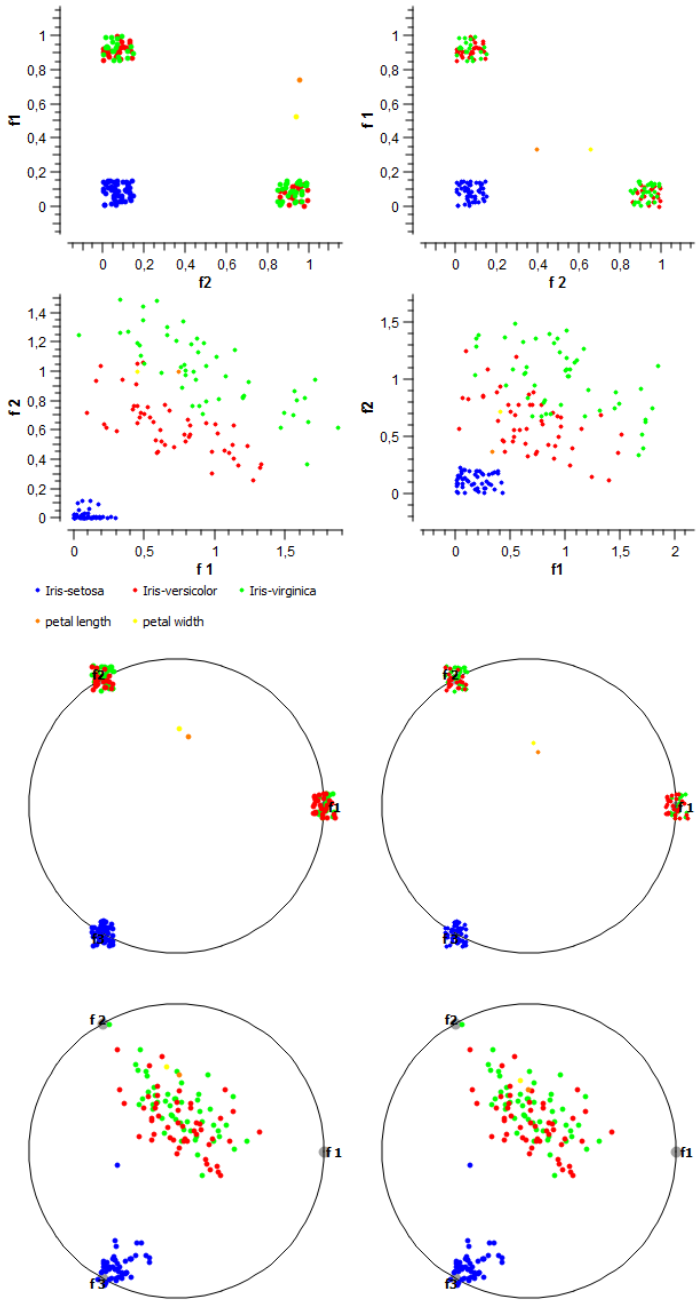
pomembni za razločevanje med razredi. Pričakujemo, da bodo takšni atributi bližje razredom, na katere močneje vplivajo.

Na sliki 4.27 vidimo prikaz podatkovne zbirke `iris`. Relativna pozicija atributov v faktoriziranem prostoru nam pove o vlogi atributov v faktorju. Na projekciji levo v prvi vrstici vidimo, da ima *petal width* nižjo vrednost po $f1$ kot *petal length*. Atributa lahko ločimo po vertikali, na pa po horizontali. Na zgornjem desnem delu slike, kjer so primeri gručeni in atributa neskalirana, pa vidimo, da imata atributa večjo razliko po $f2$ kot pa po $f1$. V projekcijah z metodo ***radviz***, kjer so primeri gručeni, vidimo, da ležita atributa bližje delu, kjer imamo mešani gruči. Sklepamo lahko, da imata pomembno vlogo pri razlikovanju med razredom *Iris-setosa* in ostalima dvema razredoma. V projekciji negručenih primerov z metodo ***radviz*** imamo sicer mešano gručo, a lahko vidimo, da atributa ležita sredi tega oblaka.

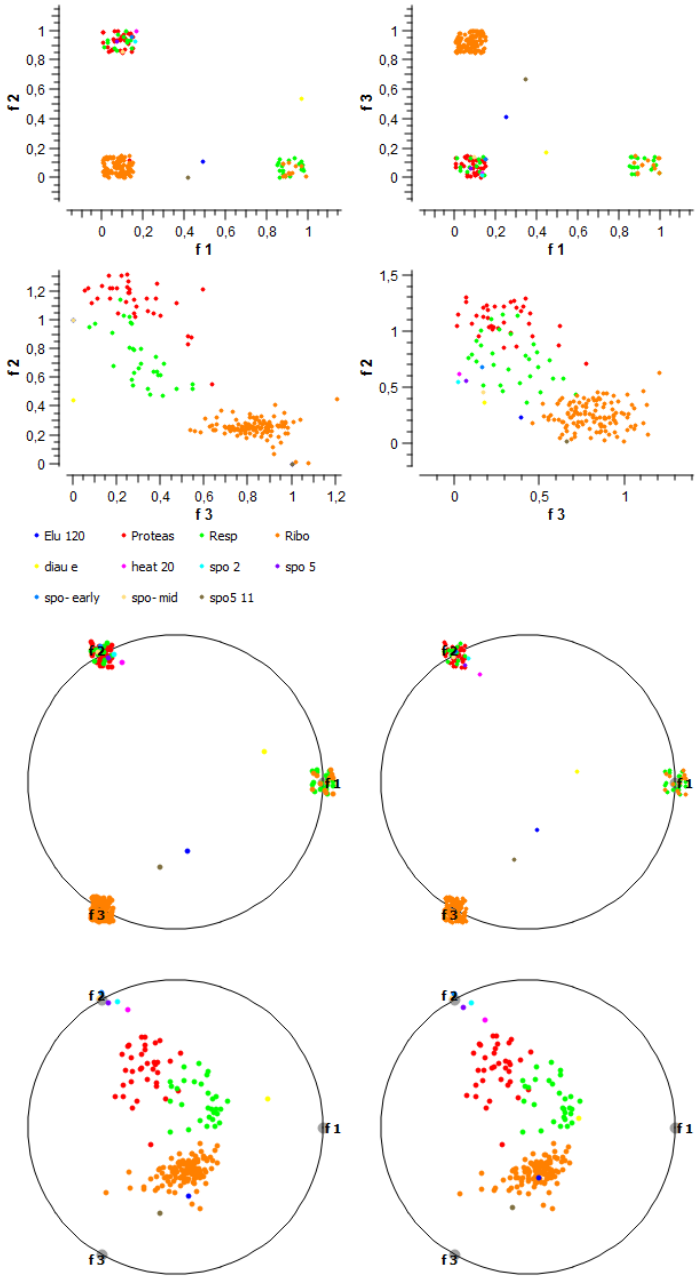
Pri podatkovni zbirki `Brown selected`, ki je prikazana na sliki 4.28, imamo v prikazu več atributov. V zgornjem levem delu slike, kjer so atributi skalirani, lahko najprej opazimo, da se večina atributov razporedi v tri gruče primerov. To se zgodi zaradi tega, ker imajo eno vrednost visoko, drugi dve pa nižji. Iz vizualizacij s skaliranimi atributi bi lahko sklepali, da sta atributa *Elu 120* in *spo5 11* še posebej pomembna za ločevanje razreda *Ribo* od ostalih. V prikazu z metodo ***radviz*** vidimo, da manj atributov pripada gručam, kot je bilo videti z razsevnim diagramom. Še posebej pri spodnjih dveh projekcijah lahko sklepamo, kateri atributi so pomembni za posamezen razred.

Pri podatkovni zbirki `DLBCL` smo že prej videli, da se v prostoru faktorjev primeri ne ločijo na lepe gruče. Na sliki 4.29 so primerom dodani še atributi. Kadar so atributi skalirani, se skoraj enakomerno razdelijo v obe gruči, izvorni pa ležijo nekje vstran, proč od primerov. Iz tega je razvidno, da je pomembno imeti že v izhodišču lepo projekcijo primerov, da lahko s hkratno vizualizacijo primerov in atributov dobimo informativen prikaz.

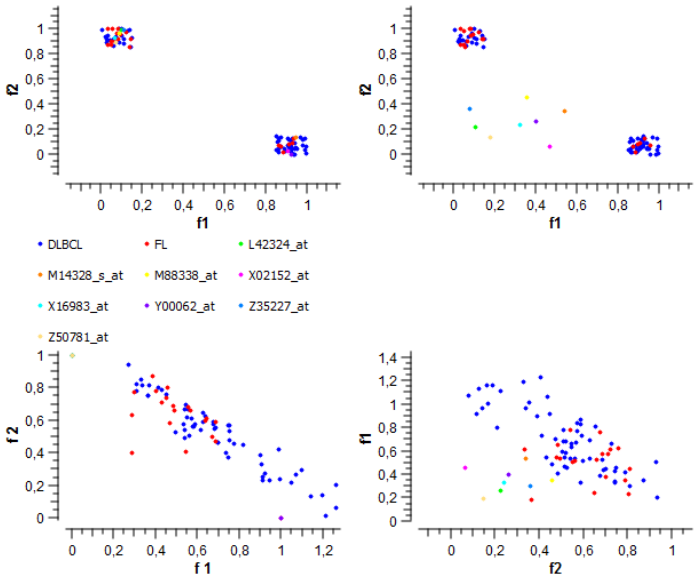
Pri podatkovni zbirki `leukemia` smo dobili eno precej čisto gručo in eno mešano. Na projekcijah na sliki 4.30, izgleda, kot da je za ločenje te gruče



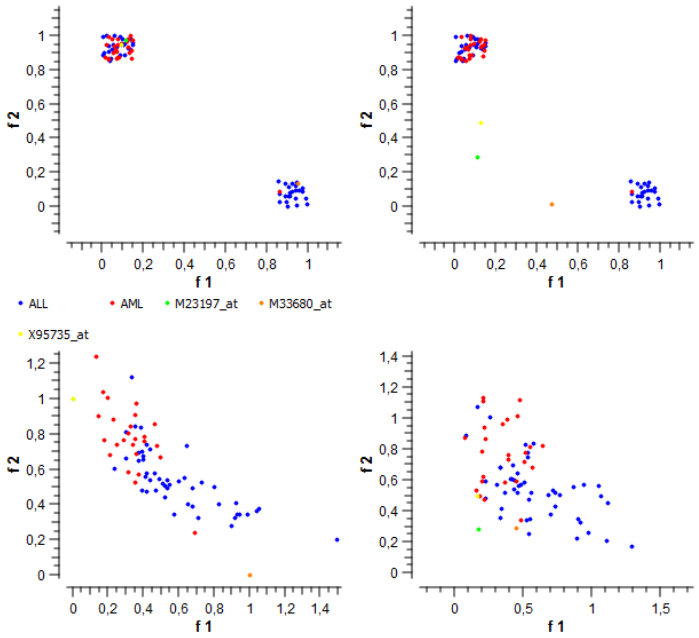
Slika 4.27: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `iris`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.



Slika 4.28: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `Brown selected`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.



Slika 4.29: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `DLBCL`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.

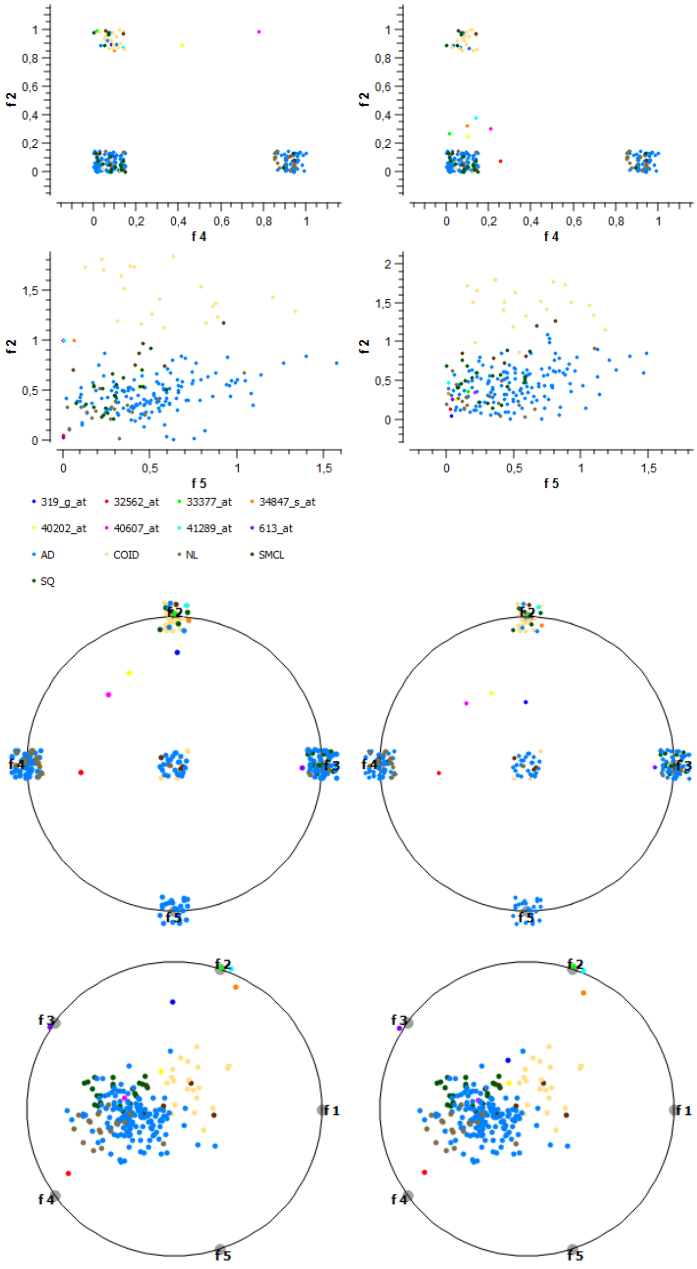


Slika 4.30: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `leukemia`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.

od druge še posebej pomemben atribut *M33680_at*. Razen v prikazu z negručenimi primeri in neskaliranimi atributi je ta namreč bližje primerom iz razreda *ALL*.

Pri projekcijah podatkovne zbirke `lung`, prikazanimi na sliki 4.31, je še posebej zanimiv prikaz z gručenimi primeri in neskaliranimi atributi v razsevnem diagramu, kjer 6 atributov obkroža eno izmed gruč. A temu ne gre pripisati večje pomembnosti, saj je to gruča v izhodišču, ki je zaradi večjega števila faktorjev sestavljena iz ostalih gruč. V tem primeru je vsekakor bolj priporočljiv prikaz z metodo ***radviz***. Tu vidimo, da se atributi prerazporedijo po prostoru, a ker nimamo jasnih mej med razredi, težko opazimo kakšno povezavo med atributi in razredi.

Ob prikazovanju podatkovne zbirke `MLL` z razsevnim diagramom, ki je na



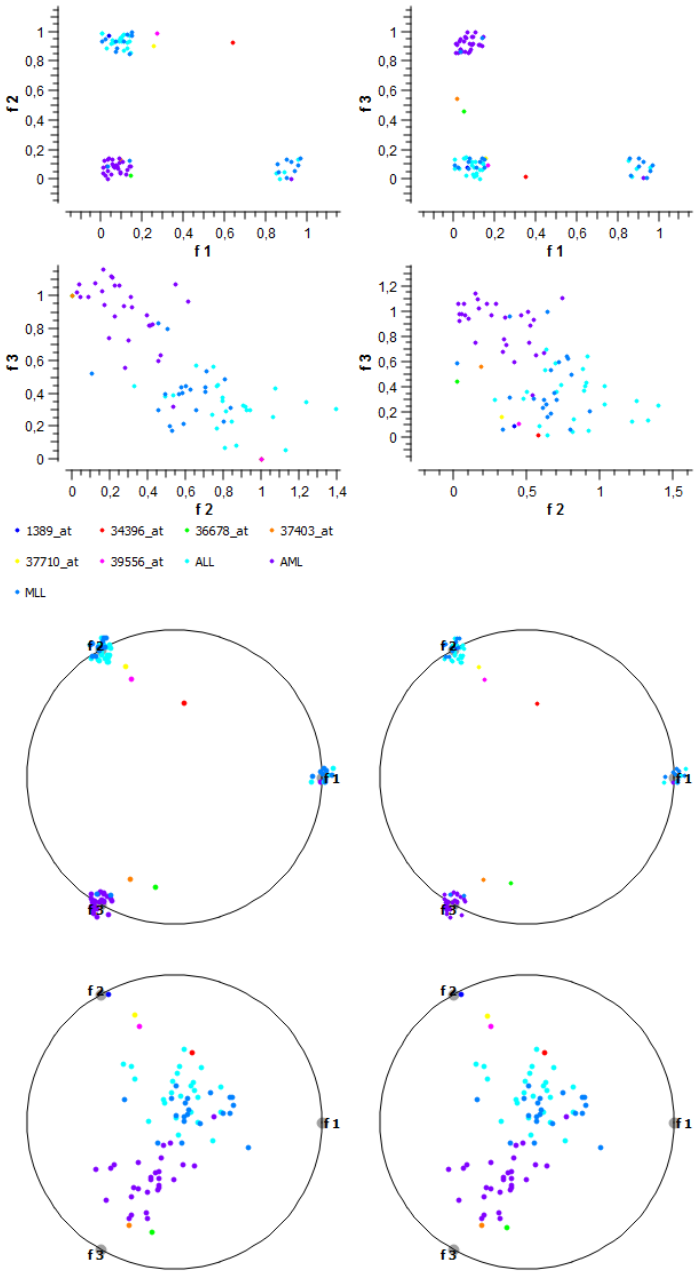
Slika 4.31: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `lung`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.

sliki 4.32, vidimo, da na čistejšo gručo verjetno delujeta dva atributa. To se najlepše vidi na zgornji desni sliki, kjer sta atributa med dvema gručama, in tudi na obeh slikah v drugi vrstici, kjer sta ista atributa v kotu, v katerem je več primerov iz razreda *AML*, ki sestavljajo prej omenjeno gručo. Na spodnji polovici slike imamo prikaze z metodo ***radviz*** in ponovno lahko vidimo, da sta ista atributa bližje tretjemu faktorju, kjer so tudi primeri iz razreda *AML*.

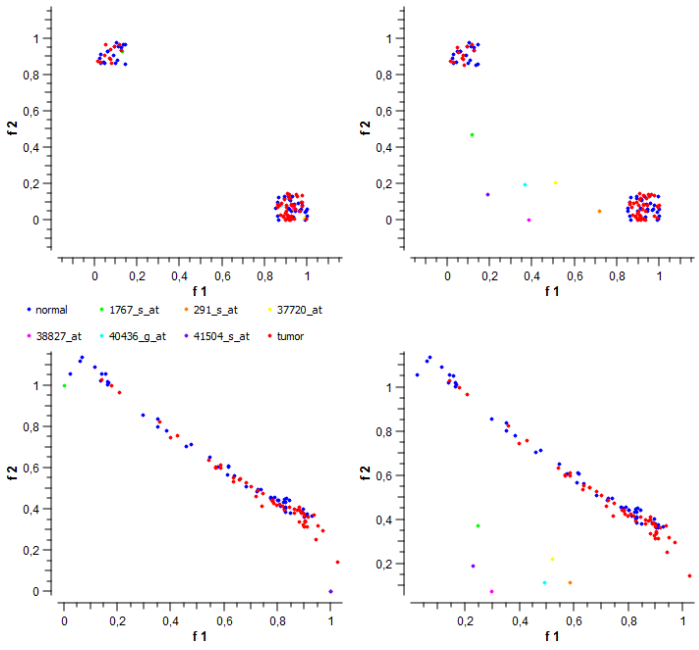
Kot smo že prej videli, se primeri iz podatkovne zbirke `prostata` gručijo v dve mešani gruči. Ker imamo samo dva faktorja, se pri skaliranju atributi praktično postavijo v gruči in na sliki 4.33 vidimo, da so vsi razen enega v isti gruči. Prikaz z negručenimi primeri in neskaliranimi atributi tudi pokaže, da so atributi in primeri v tem prostoru ločeni med seboj.

Na sliki 4.34 vidimo prikaze podatkovne zbirke `SRBCT`. Pri prikazu skupaj z gručenimi primeri v razsevnem diagramu sicer lahko vidimo, katerim gručam se atributi približajo, a je gruča pri izhodišču pravzaprav sestavljena iz dveh gruč, zato tak prikaz ni najbolj primeren. V prikazu z metodo ***radviz*** je nekoliko bolj razvidno, kateri atributi sodijo h katerim gručam, oziroma kako se razporedijo po prostoru.

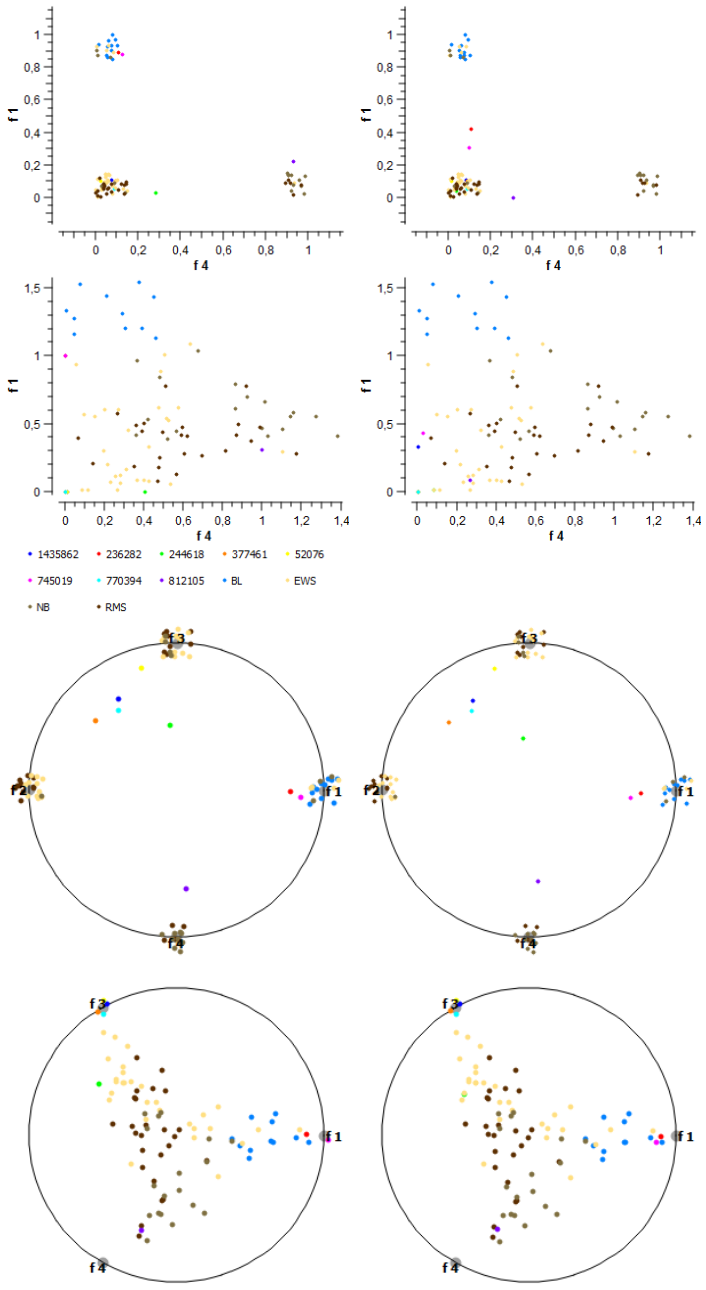
Kot smo lahko videli na primerih z našimi podatkovnimi zbirkami, je kakovost hkratnega prikaza primerov in atributov zelo odvisna že od prikaza primerov v prostoru faktorjev. Če bomo že za projekcijo primerov dobili čiste, ločene skupine, bomo s hkratnim prikazom lažje ugotovili, v kakšni povezavi so primeri in atributi. V nekaterih primerih smo tako lahko sklepali o povezavi med določenim atributom in razredom, oziroma o tem, da atributi pomagajo ločiti en razred primerov od ostalih. Kadar pa dobimo mešane gruče že pri projekciji primerov, nam pozicija atributov ne bo nič povedala, oziroma v nekaterih primerih smo videli, da so atributi ležali povsem ločeno od primerov in tako nismo mogli govoriti o povezavah.



Slika 4.32: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `MLL`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.



Slika 4.33: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `prostata`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.



Slika 4.34: Hkratna projekcija primerov in pomembnih atributov podatkovne zbirke `SRBCT`. Projekcije prikazujejo kombinacije podatkov iz $\boldsymbol{W}$ in $\boldsymbol{H}$. Zgoraj so gručeni, spodaj so izvorni primeri iz $\boldsymbol{W}$. Levo so skalirani, desno izvorni atributi iz $\boldsymbol{H}$.

4.5 Vpliv razcepljanja podatkov po faktorjih na uspešnost napovednih modelov

Poskusili smo ugotoviti, če lahko dobimo boljše projekcije z uporabo dela atributa, ki ga da določen faktor. Psevdokoda 4 prikazuje postopek, kako smo dobili novo matriko. Zmnožili smo i-ti stolpec matrike $\boldsymbol{W}$ z i-to vrstico matrike $\boldsymbol{H}$ in dobili matriko. To smo ponovili za vsak faktor, dobljene matrike pa vodoravno združili. Želeli smo ugotoviti, ali lahko z razcepom podatkov po atributih najdemo boljše projekcije primerov kot z izvornimi podatki. Želeli smo tudi primerjati atribute, ki jih VizRank na taki matriki označi za pomembne, z atributi, ki jih VizRank izbere na izvornih podatkih. Pri vsakem rangu dobimo veliko število atributov, kar je predstavljalo časovno zelo potraten problem pri že nekoliko višjih rangih, saj se je VizRank na primer pri rangu faktorizacije 100 pri podatkovni zbirki `SRBCT` izvajal skoraj dva dneva. Ocene projekcij, dobljenih s to metodo, so prikazane v tabeli 4.4. Zaradi časovne potratnosti te metode nismo uporabili na vseh podatkovnih zbirkah ter pri vseh rangih in zato nekatere vrednosti niso prikazane.

Algorithm 4 Psevdokoda za razcepljanje podatkov po faktorjih

1: $W, H \leftarrow faktorizacija(X, rang)$
2: $M \leftarrow$ prazna matrika
3: **for** i od 0 do rang - 1 **do**
4: $\quad M_i \leftarrow W[:, i] * H[i, :]$
5: $\quad M \leftarrow$ vodoravno združi M in M_i
6: **end for**

Razcep podatkov po faktorjih (pri nižjih rangih, do 10) ni prinesel želenih rezultatov, saj so bile ocene projekcij precej nižje od projekcij na izvornih podatkih. Dobljene ocene so ob večkratni faktorizaciji pri nekaterih podatkih zasedale zelo širok razpon. Na primer pri transformiranih podatkih `iris` pri rangu dve je imela najslabše ocenjena projekcija oceno 49,45, najboljša pa 89,77. Tudi z višanjem ranga faktorizacije, kolikor smo lahko, nismo dobili bistveno boljših rezultatov. Na primer pri transformiranih podatkih `SRBCT`

	2	5	10	50
iris	70,56	71,16	71,27	71,02
Brown selected	53,57	52,92	53,97	53,80
DLBCL	74,63	75,16	74,09	
leukemia	68,37	71,44	68,17	
MLL	48,33	50,00		
prostata	61,42	60,34		
SRBCT	41,40	45,62	41,04	40,07

Tabela 4.4: Primerjava ocen projekcij primerov na podatkih, razcepljenih po faktorjih glede na rang faktorizacije.

z rangom dve so bile ocene okoli 40, pri transformiranih podatkih SRBCT z rangom 50 so bile ocene še vedno okoli 40.

Poglavje 5

Zaključek

Nenegativna matrična faktorizacija je večkrat uporabljena metoda na različnih podatkovnih zbirkah za zmanjšanje dimenzionalnosti prostora in iskanje novih povezav med atributi in primeri. Čeprav se z dobljenimi modeli dosega dobre rezultate, ni vedno enostavno razumeti, zakaj model dobro deluje. Zaradi tega smo v magistrski nalogi poskušali najti načine, kako z vizualizacijo modelov izboljšati njihovo tolmačenje.

Glavni cilj magistrske naloge je bil prikaz in razumevanje modelov, dobljenih z nenegativno matrično faktorizacijo. Iskanje najboljših modelov ni bil eden izmed ciljev naloge. Kljub temu smo želeli poiskati dobre modele, na katerih smo nato preizkusili metode, ki smo jih razvili za prikaz in tolmačenje modelov.

Prvo vprašanje, ki se nam pojavi pred faktorizacijo, je, kako zagotoviti nenegativnost matrike. Preučili smo tri načine, kako spremeniti podatke, da zadostimo pogoju nenegativnosti. Videli smo, da način transformiranja podatkov pred faktorizacijo vpliva na rezultat. A vpliv ni tako velik, da ne bi mogli izbrati načina, ki nam najbolj ustreza. V nalogi smo atribute pred faktorizacijo skalirali, saj smo jih tako postavili v isti razpon in smo dobljene faktorje lažje primerjali.

Na naslednji dve vprašanji naletimo, kadar želimo izvesti hierarhično gručenje nad podatki. Ti dve vprašanji se nanašata na splošno na hierarhično

gručenje in nista vezani izključno na faktorizirane modele. Sprašujemo se namreč, kako računati razdaljo med primeri in kako združevati primere v gruče. V nalogi smo izbirali način združevanja v gruče glede na čistost dobljenih gruč in glede na dobljen izris. Pri prikazovanju gručenih toplotnih kart pa smo naleteli na težavo pri prikazovanju matrik, pri katerih je ena dimenzija precej večja od druge. Ta problem smo poskusili rešiti s stiskanjem matrik, tako da smo združevali stolpce. Videli smo, da lahko s hierarhičnim gručenjem podatkov v faktoriziranih matrikah dobimo zanimive gruče.

Nadalje smo si pogledali, v kolikšni meri smo lahko prepričani, da je dobljeni model najboljši za dani rang faktorizacije. Videli smo, da so dobljeni modeli lahko zelo različni in je potrebno faktorizacijo večkrat ponoviti, da se izognemo modelom, dobljenih v lokalnih minimumih.

Preučili smo en možen način, kako prikazati primere v prostoru faktorjev. Rang faktorizacije smo nastavili na isto vrednost, kot je število razredov v podatkih. Ker smo atribute skalirali že pred faktorizacijo, so bile vrednosti faktorjev primerljive in smo podatke iz faktorizirane matrike $\boldsymbol{W}$ lahko uporabili v projekcijah. Za prikaz smo uporabili tudi gručene primere, ki smo jih gručili tako, da smo jih dodelili gruči, ki ustreza faktorju z najvišjim koeficientom. V razsevnem diagramu lahko prikažemo do tri takšne gruče, za več pa je boljša metoda ***radviz***. Kakšen način prikaza izbrati in kako transformirati faktorizirane podatke se spreminja od primera do primera. Zato priporočamo, da se preizkusi vse kombinacije in izbere najprimernejšo.

V prikaze smo vključili še nekatere atribute. Atributov nismo gručili, temveč smo jih skalirali. Pri nekaterih podatkih se atributi postavijo zelo blizu primerov istega razreda, po čemer smo lahko sklepali, da so tisti atributi še posebej pomembni za taisti razred. Spet drugod so bili atributi oddaljeni od čiste gruče in bližje mešanim. V takšnih primerih lahko sklepamo na to, da so takšni atributi pomembni za ločevanje enega razreda od ostalih. Pri teh prikazih je najbolje izhajati iz prejšnjih, kjer smo prikazali samo primere, in nato poskusiti različne kombinacije metod prikaza in transformacij podatkov.

Glavni prispevek te naloge je predvsem razvoj treh načinov za vizualiza-

cijo faktoriziranega prostora, kar nam pomaga pri tolmačenju modelov. Ti načini so prikaz gručenih toplotnih kart faktoriziranih matrik, prikaz primerov v prostoru faktorjev ter hkratna vizualizacija primerov in atributov v prostoru faktorjev. Vsaka od teh vizualizacij nam lahko pove nekaj uporabnega o modelu, a za to moramo pri vsaki vizualizaciji najprej izbrati nekaj parametrov, da dobimo čim boljšo predstavitev. Pri gručenih toplotnih kartah moramo izbrati način merjenja razdalje in kako povezovati skupine med seboj. Pri vizualizaciji primerov in atributov v prostoru faktorjev se moramo odločiti za način prikaza (v nalogi smo predstavili razsevni diagram in metodo ***radviz***, možni pa so še drugi načini) ter kako transformirati faktorizirane podatke pred prikazom.

Literatura

[1] Brunet, J.-P., Tamayo, P., Golub, T. R., Mesirov, J. P., 2004. Metagenes and molecular pattern discovery using matrix factorization.

[2] Brunsdon, C., Fotheringham, A., Charlton, M., 1998. An investigation of methods for visualising highly multivariate datasets. Case Studies of visualization in the Social Sciences (September 2015), 55–80.
URL `http://www.agocg.ac.uk/sosci/casestudies/brunsdon/brunsdon.pdf`

[3] Demšar, J., Curk, T., Erjavec, A., Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., Zupan, B., 2013. Orange: Data mining toolbox in python. Journal of Machine Learning Research 14, 2349–2353.
URL `http://jmlr.org/papers/v14/demsar13a.html`

[4] Hunter, J. D., 2007. Matplotlib: A 2d graphics environment. Computing In Science & Engineering 9 (3), 90–95.

[5] Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems, 42–49.

[6] Leban, G., 2007. Vizualizacija podatkov s strojnim učenjem. Ph.D. thesis, Univerza v Ljubljani.

[7] Leban, G., Bratko, I., Petrovic, U., Curk, T., Zupan, B., Feb. 2005. VizRank: finding informative data projections in functional genomics

by machine learning. Bioinformatics (Oxford, England) 21 (3), 413–4.
URL http://www.ncbi.nlm.nih.gov/pubmed/15358614

[8] Lee, D., Seung, H., 2001. Algorithms for non-negative matrix factorization. Advances in neural information processing ... (1).
URL http://papers.nips.cc/paper/1861-alg

[9] Lee, D. D., Seung, H. S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788–791.

[10] Mramor, M., Leban, G., Demsar, J., Zupan, B., Aug. 2007. Visualization-based cancer microarray data classification analysis. Bioinformatics (Oxford, England) 23 (16), 2147–54.
URL http://www.ncbi.nlm.nih.gov/pubmed/17586552

[11] Python Software Foundation, 2015. Python Language Reference, version 2.7.6.
URL http://www.python.org/

[12] van der Walt, S., Colbert, S., Varoquaux, G., March 2011. The numpy array: A structure for efficient numerical computation. Computing in Science Engineering 13 (2), 22–30.

[13] Wilkinson, L., Friendly, M., 2009. The history of the cluster heat map. The American Statistician 63 (2), 179–184.
URL http://econpapers.repec.org/article/besamstat/v_3a63_3ai_3a2_3ay_3a2009_3ap_3a179-184.htm

[14] Zitnik, M., Zupan, B., 2012. Nimfa: A python library for nonnegative matrix factorization. Journal of Machine Learning Research 13, 849–853.