VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MATEMATINĖS STATISTIKOS KATEDRA

Laura Šlepikaitė

STATISTINĖ MEDICININIŲ DUOMENŲ ANALIZĖ, TAIKANT DIMENSIJOS MAŽINIMO METODUS

Magistro baigiamasis darbas

Vilnius, *2009*

Darbo vadovas :

Doc., matematikos mokslų dr., Pranas Vaitkus

Recenzentas:

Doc., matematikos mokslų dr., Julius Jonas Kruopis

Registracijos Nr.: ....

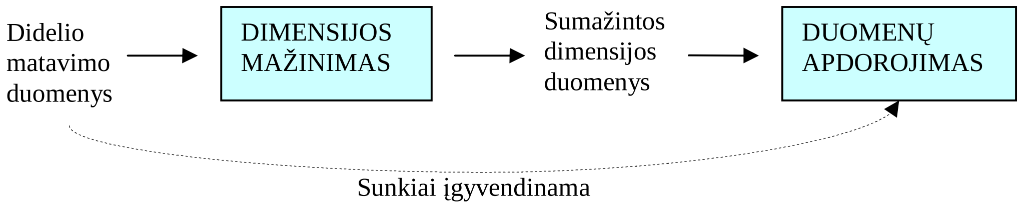
Darbo gynimo data: ....

T U R I N Y S

Įvadas

Darbo problema ir aktualumas. Statistikoje dimensijos mažinimas yra procesas, kada sumažinamas tiriamųjų kintamųjų skaičius taip, kad būtų išlaikyta duomenų struktūra. Dimensijos mažinimo – dideles dimensijos duomenų struktūros atvaizdavimo mažesnės dimensijos erdvėje – problema dažniausiai iškyla bioinformatikoje, vaizdų atpažynimo uždaviniuose, mašininiame mokyme, dirbant su atsakymais į anketų klausimus ir kt. tada, kai turimi dumenys yra didesnės dimensijos negu toleruotina matematiniu požiūriu. Pavyzdžiui, vaizdų atpažynimo ar klasifikavimo sistemose, kada yra tiriami $m \times n$ matavimo nespalvoti vaizdai, kurie gali būti transformuoti į mn matavimo vektorius. Praktikoje dažniausiai pasitaiko 256×256 matavimo vaizdai, todėl vektorius atitinkantis tokį paveikslėli jau bus 65536 matavimo ir, daugeliu atvejų, dirbti su tokiais vektoriais tampa labai sudėtinga. Daugiamtėje statistikoje, kada yra keletas matavimų, tikslas paprastai yra surasti klasterius arba kitokią duomenų struktūrą ir interpretuoti kintamuosius [6] Tam yra patogu duomenis pavaizduoti vizualiai, tačiau tai tampa praktiškai neįmanoma, kai matavimų yra daugiau negu trys. Pavyzdžiui, jeigu yra tiriama 60 žmonių atsakymai į LHFQ (*ang. Living With Heart Failure Quastionnaire*) klausimus ir norima nustayti, ar skirtingi širdies ligų gydymo metodai įtakoja žmogaus gyvenimo kokybę, yra dirbama su 21×60 matavimo duomenų rinkiniais. Grafiškai pavaizduoti tokių duomenų struktūrą yra neįmanoma, todėl paprastai atsakymai į klausimus yra sumuojami ir nagrinėjamos gautos sumos [23] Tačiau tokia duomenų analizė irgi nėra efektyvi, nes žinant, kad kiekvienas žmogus yra skirtingas, atsakymai į klausimus gali būti labai nelygiaverčiai, t.y., kas svarbu vienam žmogui, nebūtinai bus svarbu kitam ir atvirkščiai, todėl yra gana apgaulinga spręsti apie žmogaus gyvenimo kokybę pagal atsakymų į klausimus sumą. Taigi ir čia yra tikslinga sumažinti duomenų rinkinio dimensiją taip, kad būtų prarasta kuo mažiau informacijos. Pavyzdžiui, sumažinę matavimų dimensiją nuo 21 iki 2 dirbame jau nebe su 21 – mačiais, bet su 2 – mačiais vektoriais, kurie taip pat teikia maksimalų informacijos kiekį apie originalius duomenis.

Šiuolaikinė technika leidžia konstruoti gana tiksius modelius didelio matavimo duomenų rinkinių apdorojimui. Tačiau tokie metodai reikalauja nemažai skaičiavimui skiriamo laiko. Todėl labai dažnai prieš duomenų analizę yra siūloma sumažinti duomenų dimensiją [6] Taigi dimensijos mažinimas gali būti naudojamas kaip duomenų analizės dalis, supaprastinanti duomenų modelį. Tai paprastai liečia tam tikrą didelio matavimo duomenų rinkinio efektyvų atvaizdavimą mažos dimensijos erdvėje. Pavyzdžiui, dirbant su sumažintos dimensijos duomenimis, klasifikavimo uždavinių sprendimas gali duoti labai tikslius ir lengvai interpretuojamus rezultatus, o tuo pačiu yra supaprastinamas ir pats skaičiavimas.



1 pav. Dimensijos mažinimo uždavinys.

Nors didelės dimensijos duomenų rinkiniai sukelia nemažai sunkumų matematiniu atžvilgiu, tačiau tuo pačiu atsiranda ir galimybė ieškoti naujų ir efektyvių tokių duomenų rinkinių apdorojimo būdų. Viena iš pagrindinių didelio matavimo duomenų rinkinių problemų yra, kad norint nusakyti duomenų struktūrą, daugeliu atvejų ne visi kintamieji yra reikalingi. Kitas svarbus atvejis, kada reikalingas dimensijos mažinimas yra tada, kai matavimų skaičius viršija (kartais netgi labai stipriai) stebėjimų skaičių. Iš statistinės pusės yra pageidaujama, kad tiriamame duomenų rinkinyje stebėjimų skaičius reikšmingai viršytų matavimų skaičių. Teoriškai, jeigu yra daromos išvados apie duomenis, stebėjimų skaičius turi eksponentiškai augti matavimų skaičiaus atžvilgiu [7].

Egzistuoja daugybė diemensijos mažinimo metodų, kurie gali būti skirstomi į tiesinius ir netiesinius. Netiesiniai metodai yra naudojami tada, kai dirbama, pavyzdžiui, su 3-mačiais vaizdais, kurie priklauso nuo keleto parametrų, tuo tarpu tiesiniai metodai gali būti taikomi bet kokiems duomenims, kuriuos galima išdėstyti plokštumoje. Keletas tiesinių dimensijos mažinimo metodų:

- Pagrindinių komponenčių analizė (PCA), kai yra maksimizuojama projektuotų duomenų dispersija;
- Kanoninė koreliacijos analizė (CCA), kai yra maksimizuojama koreliacija tarp projektuotų duomenų.
- Tiesinė diskriminantinė analizė (LDA).
- Neneigima matricos faktorizacija (NMF), kai neneigiamų duomenų matrica N

yra išskaidoma į dvi taip pat neneigiamas matricas, vadinamas transformacijų (M) ir iššifravimo (L) matricomis, taip, kad yra minimizuojama tikslo funkcija $L(M; L) = \|N - ML\|_F^2$.

- Nepriklausomų komponenčių analizė (ICA).
- Ypatingųjų reikšmių dekompozicija (SVD).

Šiame darbe yra trumpai aprošomi PCA ir SVD metodai ir jų tarpusavio ryšys, daugiausia dėmesio skirta yra neneigiamai matricos faktorizacijai ir algoritmų igyvendinimui.

Atliekant PCA kai kurie pagrindinių komponenčių elementai gaunami neigiami. Toks rezultatas labai apsunkina duomenų interpretaciją, ypač jeigu duomenys yra teigiami. Todėl, kaip naujausia PCA alternatyva, duomenų analizei buvo panaudota NMF. Pagrindinė NMF idėja yra ta, kad tiek gaunama transformacijų matrica, tiek iššifravimo matrica yra neneigiamos.

Pirmieji darbai NMF stityje apie 1990 m. buvo atlikti suomių mokslininkų, o pats metodas vadinosi teigiama matricos faktorizacija (*ang. Positive Matrix factorization*). Metodas tapo žinomas kaip neneigiama matricos faktorizacija po to kai D. Lee ir S. Seung [3] NMF idėjos įgyvendinimui sukūrė keletą paprastų algoritmų, atliekamų naudojant atnaujinimo dauginant (*ang. Multiplicative Update*) metodą. Vėliau R. Plemmons [20] pasiūlė gradientinį NMF metodą, o P. Paatero alternuojantį mažiausių kvadratų metodą. 2004 m. P. O. Hoyer [21] parodė, kad naudojant NMF yra naudinga įvesti išretinimo apribojimus, o M. I. Jordan [15] 2008 m. pasiūlė naujus NMF metodus, kada duomenų matricai nėrai taikomi neneigiamumo apribojimai.

Šiame darbe taip pat yra aprašomas ir įgyvendinamas kitas algoritmas, leidžiantis NMF metodu gautą transformacijų matricą naudoti naujų duomenų analizei.

Darbo tikslas ir uždaviniai. *Darbo tikslas* buvo rasti efektyvią alternatyvą pacientų sergančiųjų širdies nepakankumu gyvenimo kokybės įvertinimui sumuojant atsakymus į LHFQ anketos klausimus, bei išanalizuoti, pritaikyti ir įvykdyti dimensijos mažinimo algoritmus taip, kad jie duotų efektyvius rezultatus dirbant su įvairiais medicininiais duomenimis.

Darbo uždaviniai:

- Aprašyti dimensijos mažinimo metodus ir jų panaudojimo galimybes;
- Pasiūlyti ir įgyvendinti efektyvius ir lengvai interpretuojamus dimensijos mažinimo algoritmus medicininių duomenų analizei;
- Pasinaudojus šiais algoritmais, pasiūlyti naują algoritmą, kurio pagalba būtų galima analizuoti papildomai prie duomenų rinkinio prijungiamus stebėjimus.
- Naudojantis SAS/IML® statistiniu paketu sukurti modulius (funkcijas), kuriuos būtu nesunkiai galima pritaikyti įvairių duomenų analizei;

- Palyginti gautus NMF ir PCA rezultatus, bei parodyti, kad naujų duomenų analizei pasiūlytas algoritmas yra gana efektyvus, rezultatus palyginant su kitais panašiais algoritmais.

Pasiekti rezultatai. Buvo atlikta atsakymų į LHFQ klausimyno klausimus analizė naudojant PCA ir NMF metodus. Buvo tiriamos 5 grupės ligonių, sergančių širdies nepakankumu ir kuriems buvo atlikti pakartotini keli matavimai po gydymo ar operacijos. Buvo atlikti NMF ir NMF/ALS algoritmai ir parodytas jų efektyvumas. Tuo remiantis, gautoji transformacijų matrica buvo panaudota naujų duomenų analizei. Tačiau gauti neigiami elementai iššifravimo matricoje, davė pagrindą ieškoti kitokio būdo naujų duomenų analizei, kai transformacijų matrica yra fiksuota. Todėl buvo pritaikytas naujas algoritmas, atliekamas naudojant fiksuotą transformacijų matricą ir atsitiktinį iššifravimo matricos inicializavimą. Pasidomėjus tokio metodo taikymo galimybėmis, taip pat buvo panaudotas iššifravimo matricos radimo algoritmas taikant kitokį būdą inicializavimui ir naudojant duomenų normalizavimą.

Visi algoritmai buvo sukurti naudojantis SAS/IML® statistiniu paketu. Kadangi SAS® paketas yra skirtas statistinei duomenų analizei, grafiniam jų vaizdavimui ir turi plačias panaudojimo galimybes, o taip pat leidžia atlikti ir įvairius veiksmus su matricomis, algoritmai buvo kuriami naudojantis būtent šiuo paketu. Be to SAS® pagalba nesunkiai gali būti atlita PCA. Taigi dar viena priežastis, kodėl buvo pasirinktas šis paketas - paprastesnis PCA ir NMF rezultatų palyginimas. Sukurti algoritmai yra išsaugoti kaip moduliai (funkcijos) ir, prieš tai nusistačius tam tikrus parametrus, gali būti naudojami bet kokiems duomenims. Realizuoti šie algoritmai:

- NMF – neneigiamos matricos faktorizacijos, naudojant matricų atnaujinimą dauginant, algoritmas;
- NMF/ALS – alternuojantis neneigiamos matricos faktorizacijos, naudojant mažiausių kvardratų metoda;
- NMF algoritmas, kai transformacijų matrica yra fiksuota ir naudojamas atsitiktinis iššifravimo matricos inicializavimas.
- NMF algoritmas, kai transformacijų matrica yra fiksuota ir naudojamas O. Okun [14] siūlomas iššifravimo matricos inicializavimas.

1. Pagrindinių komponenčių analizė

Pagrindinių komponenčių analizė (PCA) (*ang. Principal Component Aalysis*) dėl savo paprastumo ir yra dažniausiai praktikoje naudojamas dimensijos mažinimo būdas[6].

Pagrindnių komponenčių analizės (PCA) (*ang. Principal Components Analysis*) idėja yra sumažinti duomenų rinkinio, susidedančio iš didelio kiekio koreliuojačių kintamųjų, dimensiją taip, kad būtų išsaugota kiek galima daugiau informacijos apie duomenis. Šis rezultatas yra pasiekiamas duomenų rinkinį transformuojant į mažesnės dimensijos erdvę, susidedančią iš nekoreliuotų kintamųjų – *pagrindinių komponenčių*, kurios yra išdėstytos tokia tvarka, kad kelios pirmosios teikia didžiąją dalį informacijos apie pradinius duomenis.

1.1 Apibrėžimas ir pagrindinių komponenčių radimas

Tarkime, kad turime n-matį vektorių $X \in R^n$. Naginėjant duomenų rinkinio struktūrą ne visada yra tikslinga imti visas n komponentes, nors jos visos kartu ir teikia 100% informacijos apie duomenų rinkinį. Tikslas yra rasti keletą kintamųjų, atspindinčių daugiausia informacijos, išreikštos žinoma dispersija, koreliacijomis ir kovariacijomis [2].

Nagrinėkime atsitiktinį vektorių

$$X: X = (X_1, X_2, \ldots, X_n)^T \tag{2.1.1}$$

ir tiesines kombinacijas

$$\begin{aligned} Y_1 &= a_1^T X = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n = \sum_{i=1}^{n} a_{1i}X_i \\ Y_2 &= a_2^T X = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n = \sum_{i=1}^{n} a_{2i}X_i \\ &\vdots \\ Y_n &= a_n^T X = a_{n1}X_1 + a_{n2}X_2 + \cdots + a_{nn}X_n = \sum_{i=1}^{n} a_{ni}X_i \end{aligned} \tag{2.1.2}$$

kur i-toji tiesinė kombinacija yra vadinama i-tąja pagrindine komponente. Tegu $cov(X, X) = \Sigma = [\sigma_{ij}]_{n \times n}$ yra a.v. (2.1.1) kovariacijų matrica. Tada atsitiktinio dydžio $Y_i = a_i^T X$ dispersija yra $DY_i = D\left(a_i^T X\right) = cov\left(a_i^T X, a_i^T X\right) = a_i^T cov(X, X)\, a_i = a_i^T \Sigma\, a_i$, kurią galima padaryti kiek norint didele didinant vektoriaus a_i ilgį. Todėl natūralu nagrinėti tik tokias transformacijas, kai vektorius a_i yra normuotas, t.y. tenkina tokią sąlygą

$$a_i^T a_i = 1 \tag{2.1.3}$$

Teorema 1.1

Pagrindinės komponentės $Y_1 = a_1^T X, Y_2 = a_2^T X, \ldots, Y_n = a_n^T X$ *gaunamos tada, kai* $a_1, \ldots, a_n$ *yra tikriniai vektoriai*

$$\Sigma a_i = \lambda_i a_i, \quad i = 1, \dots, n \tag{2.1.4}$$

atitinkantys charakteringosios lygties

$$|\Sigma - \lambda I| = 0 \tag{2.1.5}$$

šaknis $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. *Pagrindinės komponentės yra nekoreliuotos ir išreikštos taip, kad jų dispersijos*

$$DY_i = a_i^T \Sigma a_i = \lambda_i, \quad i = 1, \dots, n$$

tenkina sąlygą

$$DY_1 \geq DY_2 \geq \dots \geq DY_n$$

Perėjimas nuo a.v. $X^T = (X_1, X_2, \dots, X_n)$ *prie a.v.* $Y^T = (Y_1, Y_2, \dots, Y_n)$ *nepakeičia apibendrintosios dispersijos ir koordinačių dispersijų sumos:*

$$|DY| = |DX| = |\Sigma|, \quad \sum_{i=1}^{n} DY_i = \sum_{i=1}^{n} DX_i \tag{2.1.6}$$

Įrodymas:

Norint rasti pirmąją pagrindinę komponentę, vektorių a_1 reikia pasirinkti taip, kad DY_1 būtų maksimali, su sąlyga (2.1.3). Tam patogu būtų pasinaudoti Lagranžo daugiklių metodu. Taigi sudarykime Langranžo funkciją pirmąjai didžiausia dispersijai rasti:

$$L(a_1) = a_1^T \Sigma a_1 - \lambda\left(a_1^T a - 1\right) \tag{2.1.7}$$

Randame jos išvestinę ir prilyginame 0,

$$\frac{\partial L(a_1)}{\partial a_1} = a\Sigma a_1 - 2\lambda_1 a_1 = 0 \tag{2.1.8}$$

Taigi gauname, kad

$$\Sigma a_1 - \lambda_1 a_1 = 0 \tag{2.1.9}$$

arba

$$(\Sigma - \lambda_1 I)\, a_1 = 0,$$

kur I yra $n \times n$ matavimo vienetinė matrica. Lygybės (2.1.9) abi puses dauginant iš a_1^T, gauname

$$a_1^T \Sigma a_1 - a_1^T \lambda_1 a_1 = 0, \tag{2.1.10}$$

iš to seka, kad $\lambda_1 = a_1^T \Sigma a_1 = DY_1$. Taigi galima sakyti, kad λ_1 yra didžiausia tikrinė kovariacinės matricos Σ reikšmė ir yra lygi didžiausiai dispersijai DY_1, o a_1 yra atitinkamas tikrinis vektorius. Analogiškai galima sudaryti Lagranžo funkciją dispersijai $DY_2 \leq DY_1$ rasti, tirint omenyje, kad pirma ir antra pagrindinės komponentės nekoreliuoja, t.y., kad $\operatorname{cov}\left(a_1^T X, a_2^T X\right) = 0$ ir $a_2^T a_2 = 1$. Jau buvo minėta, kad $\operatorname{cov}\left(a_i^T X, a_j^T X\right) = a_i^T \Sigma a_j$. Taigi

$$\begin{aligned} \operatorname{cov}\left(a_1^T X; a_2^T X\right) &= a_1^T \Sigma a_2 = a_2^T \Sigma a_1 = a_2^T a_1^T a_1 \Sigma a_1 \\ &= a_2^T a_1^T \Sigma a_1 a_1^T = a_2^T \lambda_1 a_1^T = \lambda_1 a_2^T a_1 = 0 \end{aligned}$$

t.y. $a_2^T a_1 = 0$. Sudarome Lagranžo funkciją

$$L(a_2) = a_2^T \Sigma a_2 - \lambda_1 a_2^T a_1 - \lambda_2 \left(a_2^T a_2 - 1\right) \tag{2.1.11}$$

Randame jos išvestinę ir prilyginame 0,

$$\frac{\partial L(a_2)}{\partial a_2} = a\Sigma a_2 - \lambda_1 a_1 - 2\lambda_2 a_2 = 0 \tag{2.1.12}$$

Abi lygybės (2.1.12) puses padauginę iš a_2^T gauname

$$a_2^T \Sigma a_2 - a_2^T \lambda_2 a_2 - a_2^T \frac{\lambda_1}{2} a_1 = 0$$

Iš čia, atsižvelgiant į tai, kad $a_2^T a_1 = 0$, seka

$$a_2^T \Sigma_2 a_2 - a_2^T \lambda_2 a_2 = 0 \tag{2.1.13}$$

arba

$$(\Sigma - \lambda_2 I)\, a_2 = 0,$$

kur I vėl gi $n \times n$ matavimo vienetinė matrica. Iš (2.1.13) gauname, kad $\lambda_2 = a_2^T \Sigma a_2 = DY_2$. Todėl galima sakyti, kad λ_2 yra antra pagal didumą tikrinė kovariacinės matricos Σ reikšmė ir lygi dispersijai DY_2, o a_2 yra atitinkamas tikrinis vektorius. Analogiškai randama ir $\lambda_3; \lambda_4; \ldots; \lambda_n$. Imkime k-ąją pagrindinę komponentę $Y_k = a_k^T X$ ir sudarykime Lagranžo funkciją dispersijai $DY_k \le DY_{k-1} \le \ldots \le DY_1$ rasti, turint omenyje, kad $\operatorname{cov}\left(a_k^T X; a_l^T X\right) = 0;\ \text{kai}\ k \ne l$ ir $a_k^T a_k = 1$. Užrašome Lagranžo funkciją

$$L(a_k) = a_k^T \Sigma a_k - \lambda_k \left(a_k^T a_k - 1\right) - \sum_{i=1}^{k-1} \lambda_i a_k^T a_i \tag{2.1.14}$$

Randame išvestinę ir prilyginame 0,

$$\frac{\partial L(a_k)}{\partial a_k} = 2\Sigma a_k - 2\lambda_k a_k - \sum_{i=1}^{k-1} \lambda_i a_i = 0, \tag{2.1.15}$$

abi lygybės (2.1.15) puses dauginant iš a_k^T gauname

$$a_k^T \Sigma a_k = a_k^T \lambda_k a_k - a_k^T \sum_{i=1}^{k-1} \frac{\lambda_i}{2} a_i = 0. \tag{2.1.16}$$

Atsižvelgiant į tai, kad $a_k^T a_l = 0;\ \text{kai}\ k \ne l$, gauname

$$\Sigma a_k - \lambda_k a_k = 0 \tag{2.1.17}$$

kas yra tas pats, kas

$$(\Sigma - \lambda_k I)\, a_k = 0.$$

Iš (2.1.17) gauname, kad $\lambda_k = a_k^T \Sigma a_k = DY_k$. Taigi $D\left(a_k^T X\right) = \lambda_k;\ k = 1, \dots, n$, kur a_k yra k-tasis tikrinis kovariacinės matricos Σ vektorius, o λ_k - jį atitinkanti tikrinė reikšmė, lygi k-tosios komponentės dispersijai DY_k.

Pagrindinių komponenčių vektorius $Y = (Y_1, Y_2, \dots, Y_n)$ yra gaunamas atliekant transformaciją $Y = A^T X$, kur A - matrica, kurios stulpelius sudaro tikriniai vektoriai $a_1, a_2, \dots, a_n$. Remiantis tuo, kad $\lambda_k = a_k^T \Sigma a_k = DY_k$ ir $\mathrm{cov}(Y_k, Y_l) = a_k^T \Sigma a_l = 0, \quad \text{kai } k \neq l$ vektoriaus Y kovariacinė matrica yra diagonali su diagonaliaisiais elementais $\lambda_1, \lambda_2, \dots, \lambda_n$. Gauname, kad

$$|DY| = |A^T \Sigma A| = |A^T||\Sigma||A| = |\Sigma||A^T A| = |\Sigma|.$$

Pažymėkime $\sum_{i=1}^{n} \sigma_{ii} = tr(\Sigma)$. Kadangi $\lambda_1, \lambda_2, \dots, \lambda_n$ yra kovariacijų matricos Σ tikrinės reikšmės su atitinkamais tikriniais vektoriais $a_1, a_2, \dots, a_n$, tai iš tikrinių reikšmių apibrėžimo[1] seka, kad $\Sigma = A \Lambda A^T$, kur Λ yra tikrinių reikšmių diagonalinė matrica, o $A = (a_1, a_2, \dots, a_n)$ ir $AA^T = I$, tada

$$\begin{aligned}\sum_{i=1}^{n} DX_i &= \sum_{i=1}^{n} \sigma_{ii} = tr(\Sigma) = tr\left(A \Lambda A^T\right) = tr\left(\Lambda A A^T\right) \\ &= tr(\Lambda) = \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} DY_i\end{aligned}$$

$\square$

1.2 Ypatingųjų reikšmių dekompozicija (SVD)

1.2.1 SVD apibrėžimas ir ypatingųjų reikšmių radimas

Ypatingųjų reikšmių dekompozicija (*ang. Singular Value Decomposition*) (SVD) – tai metodas, kurio pagalba duomenų matrica N yra suskaidoma į matricas $M;\ L;\ \text{ir}\ S$ taip, kad

$$N = MSL^T \tag{2.2.1}$$

kur $N \in R^{m \times n};\ M \in R^{m \times m};\ L \in R^{n \times n}$, o $S \in R^{m \times n}$ - diagonalinė matrica su teigiamais elementais pagrindinėje diagonalėje[2] ir nuliais visur kitur.

SVD yra glaudžiai susietas su PCA ir tikrinių reikšmių skaičiavimu. Pagal tikrinių reikšmių apibrėžimą, tikrinės reikšmės ir tikriniai vektoriai kvadratinei matricai A yra atitinkamai skaliarai λ ir nenuliniai vektoriai X tokie, kad

---

[1] Tegu A yra kvadratinė matrica. Nenulinis vektorius C yra vadinamas tikriniu matricos A vektoriumi, tada ir tik tada, jeigu egzistuoja toks λ, kad $AC = \lambda C$. Jei toks λ egzistuoja, tai jis yra vadinamas tikrine matricos A reikšme.

[2] Tiesinėje algebroje matricos A pagrindine diagonale yra vadinamas elementų A_{ij} rinkinys, kur $i = j$

$$AX = \lambda X. \tag{2.2.2}$$

Taigi tam, kad būtų galima suskaičiuoti tikrines reikšmes λ, matrica A turi būti kvadratinė. SVD metodas neturi tokio apribojimo ir gali būti pritaikytas bet kuriai $m \times n$ matavimo matricai. Ypatingosios matricos A reikšmės σ (*ang. singular values*) yra randamos sprendžiant lygtis

$$Al = \sigma m \tag{2.2.3}$$

ir

$$A^T m = \sigma l, \tag{2.2.4}$$

kur vektoriai m ir l yra nenuliniai ir yra vadinami atitinkamai dešiniuoju ir kairiuoju ypatingaisiais vektoriais (*ang. singular vectors*). Iš (2.2.2) seka, kad

$$(A - \lambda I)X = 0 \tag{2.2.5}$$

kas reiškia, kad

$$|A - \lambda I| = 0. \tag{2.2.6}$$

Tarkime, kad $\lambda_1, \lambda_2, \dots, \lambda_n$ yra matricos A tikrinės reikšmės, o $X_1, X_2, \dots, X_n$ - atitinkami tikriniai vektoriai. Tegu $\Lambda \in R^{n \times n}$ yra diagonalinė matrica su elementais λ_j diagonalėje, o matrica $X \in R^{n \times n}$ tokia, kad jos $j-$tasis stulpelis yra X_j ir $j = 1, 2, \dots, n$. Tada (2.2.2) galima užrašyti taip:

$$AX = X\Lambda \tag{2.2.7}$$

Tariame, kad matricai X egzistuoja atvirkštinė matrica X^{-1} ir iš (2.2.7) gauname, kad

$$A = X\Lambda X^{-1} \tag{2.2.8}$$

Išraiška (2.2.8) yra vadinama tikrinių reikšmių dekompozicija (*ang. eigenvalue deomposition*). Analogiškai (2.2.3) ir (2.2.4) galima užrašyti matricų išraiška. Tegu matricos M ir L yra tokios, kad jų stulpeliai yra normuoti vektoriai atitinkamai m_j ir l_j, t.y. jų normos $\|m_j\| = \|l_j\| = 1$ ir tegu matrica $S \in R^{m \times n}$. Vadinasi $M^T M = 1$, $L^T L = 1$ ir (2.2.3) ir (2.2.4) galima užrašyti atitinkamai kaip

$$AL = MS \tag{2.2.9}$$

ir

$$A^T M = SL = LS^T \tag{2.2.10}$$

Tada (2.2.9) dauginant iš L^T, o (2.2.10) iš M^T gauname, kad

$$A = MSL^T \tag{2.2.11}$$

Išraiška (2.2.11) yra vadinama matricos A ypatingųjų reikšmių dekompozicija (*ang. Singular Value Decomposition*) (SVD).

Aprašytas metodas, kai matrica $S \in R^{m \times n}$ dar yra vadinamas pilnąja SVD (FSVD) (*ang. Full Singular Value Decomposition*). Tačiau atliekant SVD kai $m > n$, matrica M tampa didelė $m \times m$ matavimo matrica. Turint omenyje, kad visi matricos S elementai, išskyrus elementus pagrindinėje diagonalėje, yra nuliai (t.y. paskutinėse $m - n$ matricos S eilutėse elementai yra nuliai) ir dauginant matricas M ir S, gauname, kad visi elementai iš matricos $M \in R^{m \times r}$ paskutinių $m - n$ stulpelių yra dauginami iš 0. Taigi galima sakyti, kad norint atstatyti duomenų matricą N, jie yra nereikalingi. Tokiu atveju yra naudinga naudoti kita SVD metodą, vadinamą ekonomiška SVD (ESVD) (*ang. Economy Singular Value Decomposition*), kada į skaičiavimą yra įtraukiami tik pirmieji n matricos M stulpeliai ir pirmosios n matricos S eilutės. Tuo tarpu abiem atvejais matrica L išlieka tokio paties $n \times n$ matavimo. Taigi atliekant ESVD matrica N suskaidoma į matricas $M;\ L;\ S$ taip, kad

$$N = MSL^T \tag{2.2.12}$$

kur $N \in R^{m \times n};\ M \in R^{m \times n};\ L \in R^{n \times n};\ S \in R^{n \times n}$. Matricos S diagonalėje esančios reikšmės, vadinamos ypatingosiomis reikšmėmis, yra teigiamos ir išrykiuotos mažėjimo tvarka [10].

1.2.2 SVD ir PCA ryšys

Egzistuoja tiesioginis ryšys tarp SVD ir PCA tuo atveju, kada pagrindinės komponentės yra skaičiuojamos naudojantis kovariacijų matrica. Galima parodyti, kad SVD gali būti panaudota PCA atlikimui [9]. Naudojant SVD, matrica N gali būti užrašyta (2.2.12) išraiška. Jei tartume, kad $\mu = E(N) = 0$, tai gautume, kad $\mathrm{cov}(N;N) = E(N - E(N))(N - E(N))^T = E\left[NN^T\right] = \frac{1}{n}NN^T$. Tada matricos N kovariacijų matrica

$$\Sigma = \frac{1}{n}NN^T = \frac{1}{n}MSL^TLSM^T = \frac{1}{n}MS^2M^T \tag{2.2.13}$$

Pasinaudojant tuo, kad SVD metu ypatingosios reikšmės matricos S pagrindinėje diagonalėje yra išrikiuojamos mažejimo tvarka, išrūšiuotas kovariacijų matricos Σ tikrines reikšmes atitinka elementai matricos M stulpeliuose, kai $m < n$ ir elementai pirmuosiuose n matricos M stulpeliuose, kai $m \geq n$. Tada galima užrašyti tiesines kombinacijas

$$Y = \tilde{M}^TN = \tilde{M}^TSL^T \tag{2.2.14}$$

kur matrica $\tilde{M}^TM \in R^{m \times n}$ yra diagonalinė matrica su vienetais pagrindinėje diagonalėje ir nuliais visur kitur.

1.3 PCA privalumai ir trūkumai

1.2.2 skyrelyje buvo parodyta, kad PCA gali būti atlikta pasinaudojus kitu dimensijos mažinimo metodu, vadinamu SVD. Kaip jau buvo minėta, kiekviena matrica N gali būti užrašyta kaip (2.2.12), kur M yra pagrindinių komponenčių matrica, o matricos L stulpelių elementai yra pagrindines komponentes atitinkantys koeficientai. Paprastai yra pasirenkamos pirmosios p $(p \ll n)$ pagrindinės komponentės ir to pakanka, kad būtų galima nusakyti duomenų struktūrą.

PCA turi keletą pagrindinių savybių, kurios suteikia galimybę efektyviai naudoti šį metodą:

1. PCA yra tiesinis būdas dimensijos mažinimo procedūrai atlikti.
2. PCA metu maksimalios pagrindinių komponenčių dispersijos yra išrykiuojamos nuosekliai mažėjimo tvarka, kas garantuoja minimalų informacijos praradimą.
3. Pagrindinės komponentės yra nekoreliuotos, todėl dirbant su viena pagrindine komponente nereikia atsižvelgti į kitas.

Tačiau PCA turi ir vieną gana ryškų trūkumą, t.y., kad kiekviena pagrindinė komponentė yra visų n kintamųjų tiesinė kombinacija, kur kiekvienas koeficientas paprastai yra nelygus 0, o tai dažnai sukelia sunkumų interpretuojant gautąsias pagrindines komponentes. Šiai problemai spręsti dažnai yra naudojami pasukimo metodai, kuriuos 1995 m. aprašė I. T. Jolliffe, o 2000 m. Vines pasiūlė būdą nagrinėti pagrindines komponentes, kai koeficientai yra maži sveiki skaičiai: 0, 1 ir -1 [11].

2. Neneigiama matricos faktorizacija (NMF)

2.1 NMF apibrėžimas

Neneigiama matricos faktorizacija (NMF) yra bene naujausia PCA alternatyva, kada duomenys ir gaunamos komponentės yra neneigiami. Tarkime, kad turime matricą $N \in \mathbb{R}^{m \times n}$, kur $N_{ij} \geq 0;\ i = 1;\dots;m;\ j = 1;\dots;n$ ir $r \leq \max(m;n)$. Tada NMF pagalba yra randamos matricos $M \in \mathbb{R}^{m \times r}$ ir $L \in \mathbb{R}^{r \times n}$ tokios, kad [5]:

$$N \approx ML \tag{3.1.1}$$

Tarkime, kad $M_{ia};\ L_{bj} \geq 0;\ i = 1;\dots;m;\ j = 1;\dots;n;\ a;b = 1;\dots;r$. Matricos N j-tasis stulpelis gali būti užrašytas kaip matricos M ir j-tojo matricos L stulpelio sandauga, t.y. $N_j \approx ML_j;\ \ j = 1;\dots;n$. Kitaip tariant kiekvienas duomenų vektorius $N_j;\ \ j = 1;\dots;n$ gali būti išreikštas matricos M elementų ir matricos L j-tojo stulpelio elementų sandaugos sumomis.

Matricos M ir L yra randamos maksimizuojant tikslo funkcijas,

$$L(M;L) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(N_{ij} - (ML)_{ij}\right)^2; \qquad (3.1.2)$$
$$M_{ia};\quad L_{bj} \geq 0;\quad i = ;\ldots;n;\quad a;b = 1;\ldots;r$$

(3.1.2) gali būti perrašyta taip,

$$L(M;L) = \frac{1}{2}\|N - ML\|_F^2\ , \qquad (3.1.3)$$

kur $\|\cdot\|_F$ yra Frobenijaus norma [3]. Taigi uždavinys gali būti suformuluotas taip:

Tarkime, kad yra duota neneigiama matrica $N \in R^{m\times n}$ *ir* $r > \min\{n;m\}:\ r > 0;\ r \in Z:$ *Rekia rasti tokias neneigiamas matricas* $M \in R^{n\times r}$ *ir* $L \in R^{r\times m}$ *, kad būtų minimizuojama tikslo funkcija* $L(M;L) = \frac{1}{2}\|N - ML\|_F^2$.

Kitas matricų M ir L radimo būdas yra minimizavimas tikslo funkcijos:

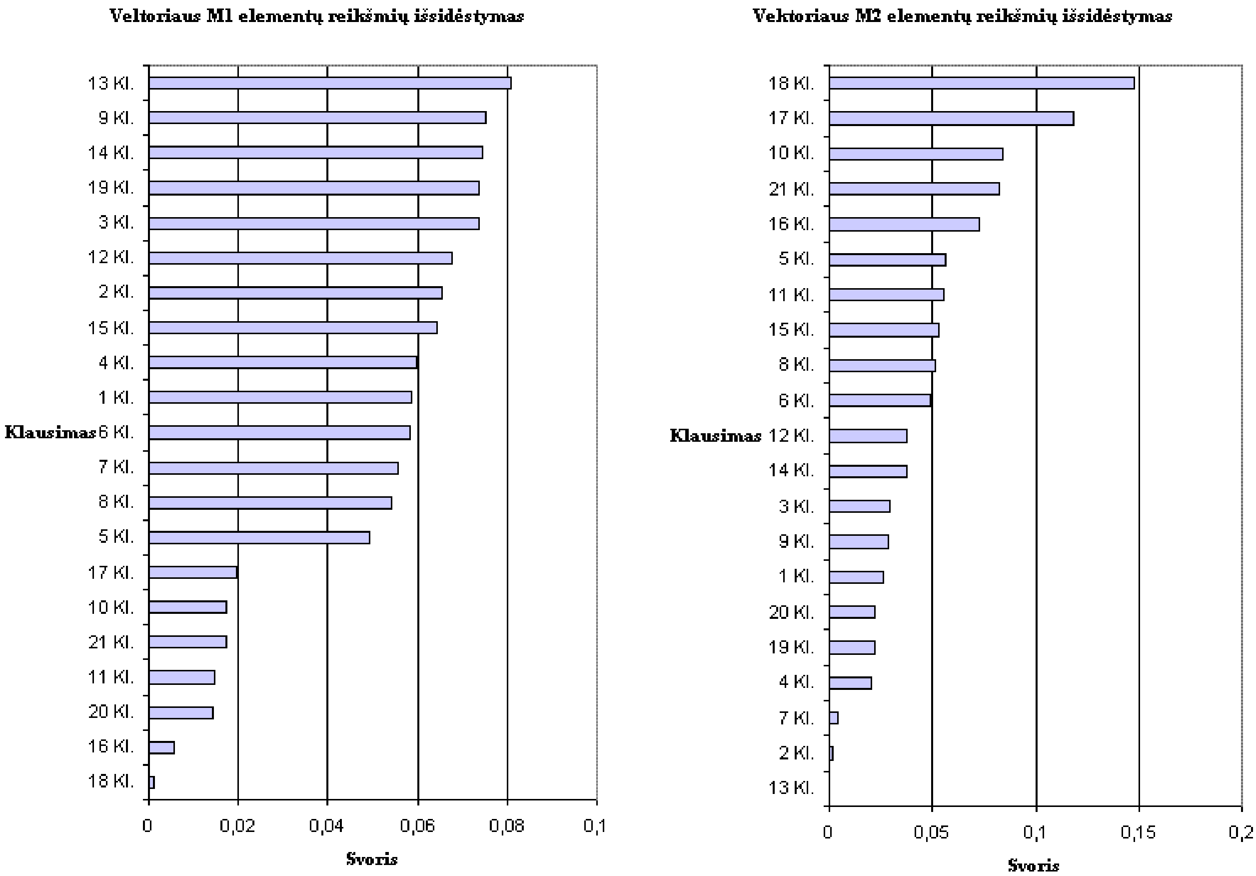
$$L(M;L) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(N_{ij}\log\frac{N_{ij}}{(ML)_{ij}} - N_{ij} + (ML)_{ij}\right); \qquad (3.1.4)$$
$$M_{ia};\ L_{bj} \geq 0;\ a;b = 1;\ldots;r$$

2.2 NMF interpretacija

Neneigiamą matricą N NMF pagalba galima išskaidyti taip kaip buvo parodyta 2.1 skyrelyje. Tokiu būdu matricą M galima pavadinti transformacijų matrica, kuri yra naudojama tiesinei matricos N aproksimacijai. Tuo tarpu matrica L gali būti vadinama iššifravimo matrica. Taip pat, galima pastebėti, kad vienas iš NMF privalumų, lyginant su kitais faktorizacijų metodais, yra tas, kad gaunama transformacijų matrica M yra neneigiama, ko negauname naudodami, pavyzdžiui, PCA metodą. Šis NMF privalumas labai palengvina transformacijų matricos elementų interpretaciją. Kaip pavyzdį panagrinėkime atsakymus į LHFQ klausimyno klausimus, kurie buvo užduoti pacientams prieš atliekant vienokį ar kitokį gydymą. Tokiu atveju turime $m \times n$ (m ir n - atitinkamai klausimų ir pacientų skaičius) duomenų matricą N, kuriai galime atlikti faktorizaciją. Šiuo atveju pasirenkamas faktorizacijs rangas r gali būti laikomas „nematomų“ klausimų grupių skaičiumi. Atlikus NMF, gaunama transformacijų matrica $M \in R^{m\times r}$ tampa klausimų – grupių matrica, kurios stulpeliai yra baziniai NMF vektoriai. Nenuliniai matricos M stulpeliai, kurie tuo pačiu yra ir neneigiami, atitinka atitinkamus klausimyno klausimus. Nagrinėjant didžiausius svorius šiame vektoriuje turinčius klausimus (anketa priede Priedas Nr. 1) galima pastebėti, kad susidaro maždaug dvi klausimų grupės,

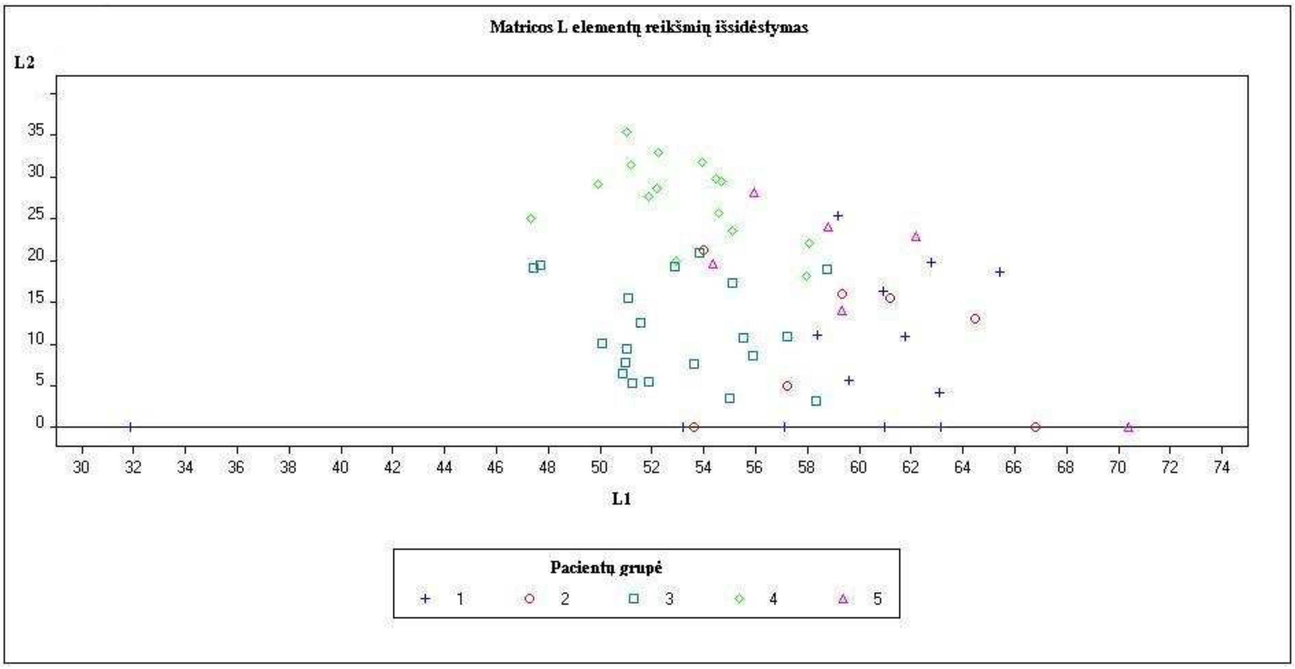
[3] Frobenijaus norma – tai matricos $A_{p\times s}$ norma, apibrėžiama kaip kvadratinė šaknis iš matricos A elementų absoliutinių dydžių kvadratų sumos, t.y. $\|A\|_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{s} a_{ij}^2}$.

kurios klausimus suskirsto į labiau atspindinčius paciento emocinę būsena (vektorius M2) ir labiau orientuotus į paciento fizinę būklę (M1). Bazinius vektorius, gautus atlikus NMF duomenų matricai, kurios elementai yra pacientų atsakymai į LHFQ klausimyno klausimus, galima pavaizduoti grafiškai (1 pav.) Ši gaunamos transformacijų matricos savybė sudaro galimybę vartotojui interpretuoti gaunamus bazinius vektorius.



1 pav. NMF transformacijų matricos interpretacija, faktorizacijai panaudojus pacientų atsakymus į LHFQ klausimyno klausimus. Pacientų duomenys prieš atliekant gydymą. n=61, m=21, r=2.

Analogiškai interpretuoti būtų galima ir iššifravimo matricą L. Aprašomu atveju atlikus matricos N faktorizaciją, gaunama matrica $L \in R^{r \times n}$ tampa grupių – pacientų matrica, kurios elementai taip pat yra neneigiami. Galima sakyti, kad pirmojo matricos L stulpelio j-taisis elementas parodo pirmojo paciento, šiuo atveju, savo fizinės ir emocinės būsenos vertinimą. Kadangi pasirenkamas faktorizacijos rangas $r = 2$, tai kiekvieną pacientą galime pavaizduoti plokštumoje. Pavaizduokime matricos L elemenus grafiškai (2 pav.) Iš grafiko matome, kad visų grupių pacientai pasiskirsto maždaug vienodai, nes, kaip jau buvo minėta, testas buvo atliktas dar prieš gydymą, todėl pacientai su vienoda ligos diagnoze (visi tiriami pacientai serga širdies nepakankamumu) savo savijautą vertina panašiai. Veliau (3.2 skyrelis) bus pastebėta, kad panaudojus vienokį ar kitokį gydymą atsiskiria tam tikros pacientų grupės.



2 pav. NMF iššifravimo matricos interpretacija, faktorizacijai panaudojus pacientų atsakymus į LHFQ klausimyno klausimus. Pacientų duomenys prieš atliekant gydymą. n=61, m=21, r=2.

2.3 Tikslo funkcijos (3.1.3) minimizavimo algoritmai

2.3.1 Atnaujinimo dauginant algoritmas

Tikslo funkcijos (3.1.3) minimizavimui dažniausiai naudojamas būdas yra Lee ir Seung (2001) pasiūlytas atnaujinimo dauginant algoritmas [3]

Algoritmas 1

Atnaujinimo dauginant algoritmas uždaviniui (3.1.2) spręsti

---

1. Apsibrėžiam r ir atsitiktinai generuojame matricas M ir L.
2. Kiekvienam $k = 1, 2 \ldots$

$$L_{bj}^{(k+1)} \leftarrow L_{bj}^{(k)} \frac{\left(\left(M^{(k)}\right)^T N\right)_{bj}}{\left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)}\right)_{bj}}; \quad b = 1, \ldots, r; \; j = 1, \ldots, n. \qquad (3.3.1)$$

$$M_{ia}^{(k+1)} \leftarrow M_{ia}^{(k)} \frac{\left(N \left(L^{(k+1)}\right)^T\right)_{ia}}{\left(M^{(k)} L^{(k+1)} \left(L^{(k+1)}\right)^T\right)_{ia}}; \quad i = 1, \ldots, m; \; a = 1, \ldots, r. \qquad (3.3.2)$$

Algoritmas gerai funkcionuoja tik tada, kai lygčių (3.3.1) ir (3.3.2) vardikliai nėra lygūs 0. Šiai problemai spręsti buvo pasiūlyta [5] naudotis teorema:

Teorema 2.1

Jeigu visi matricos N stulpeliai ir visos eilutės yra nenuliniai ir $M_{ia}^{(1)} > 0; \; L_{bj}^{(1)} > 0$, kur $i = 1, \ldots, m; \quad j = 1, \ldots, n; \quad a, b = 1, \ldots, r$, tada

$$M_{ia}^{(k)} > 0 \text{ ir } L_{bj}^{(k)} > 0; \text{ visiems } i; \; a; \; b; \; j; \text{ ir visiems } k \geq 1. \qquad (3.3.3)$$

Įrodymas

Kai $k = 1$; tai $M_{ia}^{(1)} > 0$ ir $L_{bj}^{(1)} > 0$ pagal teoremą. Naudojame indukciją ir tariame, kad (3.3.3) yra teisinga k-tąjame žingsnyje. Tada $\left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)}\right)_{bj} > 0$. Taip pat iš teoremos žinome, kad visi stulpeliai ir visos eilutės iš matricos N yra nenuliniai. Taigi galime sakyti, kad $\left(\left(M^{(k)}\right)^T N\right)_{bj} > 0$. Vadinasi $L_{bj}^{(k+1)} > 0$. Iš to seka, kad $\left(M^{(k)} L^{(k+1)} \left(L^{(k+1)}\right)^T\right)_{ia} > 0$ ir $\left(N \left(L^{(k+1)}\right)^T\right)_{ia} > 0$, kas reiškia, kad $M_{ia}^{(k+1)} > 0$. Taigi padarę prielaidą, kad (3.3.3) tvirtinimas teisingas k-tąjame žingsnyje, gauname, kad jis teisingas ir $k+1$-ąjame žingsnyje. Vadinasi, kai išpildytos teoremos sąlygos, tai $M_{ia}^{(k)} > 0$ ir $L_{bj}^{(k)} > 0$, su visais i, a, b, j ir su visais $k \geq 0$.

$\square$

2.4 Modifikuotas atnaujinimo dauginant algoritmas (3.1.2) uždaviniui

Diferencijuokime tikslo funkciją (3.1.3):

$$\begin{aligned}\frac{\partial L(M,L)}{\partial M} &= \frac{1}{2}\frac{\partial tr(N - ML)^2}{\partial M} = (ML - N)\frac{\partial tr(ML)}{\partial M}\\ &= (ML - N)L^T\end{aligned}$$

$$\begin{aligned}\frac{\partial L(M,L)}{\partial L} &= \frac{1}{2}\frac{\partial tr(N - ML)^2}{\partial L} = (ML - N)\frac{\partial tr(ML)}{\partial L}\\ &= M^T(ML - N)\end{aligned}$$

Taigi galime užrašyti, kad

$$\begin{aligned}\nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj} &= \left[\left(M^{(k)}\right)^T \left(M^{(k)}L^{(k)} - N\right)\right]_{bj}\\ &= \left(\left(M^{(k)}\right)^T M^{(k)}L^{(k)}\right)_{bj} - \left(\left(M^{(k)}\right)^T N\right)_{bj}\end{aligned} \tag{3.4.1}$$

$$\begin{aligned}\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} &= \left[\left(M^{(k)}L^{(k+1)} - N\right)\left(L^{(k+1)}\right)^T\right]_{ia}\\ &= \left(M^{(k)}L^{(k+1)}\left(L^{(k+1)}\right)^T\right)_{ia} - \left(N\left(L^{(k+1)}\right)^T\right)_{ia}\end{aligned} \tag{3.4.2}$$

Pasinaudoję gautomis išraiškomis (3.4.1) ir (3.4.2) galime užrašyti lygybes:

$$\begin{aligned}L_{bj}^{(k+1)} &= L_{bj}^{(k)}\frac{\left(\left(M^{(k)}\right)^T N\right)_{bj}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}} = L_{bj}^{(k)}\left(1 - \frac{\left(\left(M^{(k)}\right)^T M^{(k)}L^{(k)}\right)_{bj} - \left(\left(M^{(k)}\right)^T N^{(k)}\right)_{bj}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}}\right) =\\ &= L_{bj}^{(k)}\left(1 - \frac{\nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}}\right) = L_{bj}^{(k)} - \frac{L_{bj}^{(k)}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}}\nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj}\end{aligned}$$

$$M_{ia}^{(k+1)} = M_{ia}^{(k)} \frac{\left(N\left(L^{(k+1)}\right)^T\right)_{ia}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}} = M_{ia}^{(k)} \left(1 - \frac{\left(M^{(k)}L^{(k+1)}\left(L^{(k+1)}\right)^T - N\left(L^{(k+1)}\right)^T\right)_{ia}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}}\right) =$$

$$= M_{ia}^{(k)} \left(1 - \frac{\left(\nabla_M f\left(M^{(k)};L^{(k+1)}\right)\right)_{ia}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}}\right) = M_{ia}^{(k)} - \frac{M_{ia}^{(k)}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}} \left(\nabla_M f\left(M^{(k)};L^{(k+1)}\right)\right)_{ia}$$

Taigi gauname, kad (3.3.1) ir (3.3.2) galima užrašyti atitinkamai:

$$L_{bj}^{(k+1)} \leftarrow L_{bj}^{(k)} \frac{\left(\left(M^{(k)}\right)^T N\right)_{bj}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}} = L_{bj}^{(k)} - \frac{L_{bj}^{(k)}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}} \left(\nabla_L f\left(M^{(k)};L^{(k)}\right)\right)_{bj} \tag{3.4.3}$$

$$M_{ia}^{(k+1)} \leftarrow M_{ia} \frac{\left(N\left(L^{(k+1)}\right)^T\right)_{ia}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}} = M_{ia}^{(k)} - \frac{M_{ia}^{(k)}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}} \left(\nabla_M f\left(M^{(k)};L^{(k+1)}\right)\right)_{ia} \tag{3.4.4}$$

$\frac{L_{bj}^{(k)}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}}$ ir $\frac{M_{ia}^{(k)}}{\left(M^{(k)}L^{(k+1)}\left(L^{(k+1)}\right)^T\right)_{ia}}$ nusako žingsnio dydį atnaujinant atitinkamai $L_{bj}^{(k)}$ ir $M_{ia}^{(k)}$.

Vykdant modifikuotą atnaujinimo dauginant algoritmą galimos dvi problemos:

1. $\left(\left(M^{(k)}\right)^T L^{(k)}\right)_{bj} = 0$ arba $\left(M_{ia}^{(k)} L^{(k+1)}\left(L^{(k+1)}\right)^T\right)_{ia} = 0$
2. Jeigu $L_{bj}^{(k)} = 0$ $\left(M_{ia}^{(k)} = 0\right)$ ir $\left(\nabla_L f\left(M^{(k)};L^{(k)}\right)\right)_{bj} > 0$ $\left(\left(\nabla_M f\left(M^{(k)}L^{(k+1)}\right)\right)_{ia} > 0\right)$, tai $L_{bj}^{(k)}$ $\left(M_{ia}^{(k)}\right)$ nepasikeičia.

Tam, kad būtų galima išvengti paminėtų problemų buvo pasiūlyta [5] žingsnio dydį $\frac{L_{bj}^{(k)}}{\left((M^{(k)})^T M^{(k)}L^{(k)}\right)_{bj}}$ modifikuoti į

$$\left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)}\bar{L}^{(k)}\right)_{bj}+\delta}\right); \text{ kur}$$

$$\bar{L}^{(k)}_{bj} = \begin{cases} L^{(k)}_{bj}, & \text{jei } \nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj} \geq 0 \\ \max\left(L^{(k)}_{bj}, \varepsilon\right), & \text{jei } \nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj} < 0 \end{cases} \tag{3.4.5}$$

Analogiškai žingsnio dydį $\left(\frac{M^{(k)}_{ia}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}}\right)$ modifikuojam į

$$\left(\frac{\bar{M}^{(k)}_{ia}}{\left(\bar{M}^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}}\right); \text{ kur}$$

$$\bar{M}^{(k)}_{ia} = \begin{cases} M^{(k)}_{ia}, & \text{jei } \nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \geq 0 \\ \max\left(M^{(k)}_{ia}, \varepsilon\right), & \text{jei } \nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} < 0 \end{cases} \tag{3.4.6}$$

Maži skaičiai $\varepsilon, \delta > 0$ yra apibrėžiami prieš algoritmo vykdymą.

Algoritmas 2

Modifikuotas atnaujinimo dauginant algoritmas (3.1.2) uždaviniui spręsti

---

1. Apibrėžiam $\varepsilon > 0$ ir $\delta > 0$. Generuojame $M^{(1)}_{ia} \geq 0$, $L^{(1)}_{bj} \geq 0$, su visais i, a, b, j
2. Kiekvienam $k = 1, 2, \ldots$
   - Jei $\left(M^{(k)}, L^{(k)}\right)$ stacionarus, tai algoritmą stabdom;
   - Kitu atveju su visais a, b, i, j skaičiuojam:

$$L^{(k+1)}_{bj} = L^{(k)}_{bj} - \left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)}\bar{L}^{(k)}\right)_{bj}+\delta}\right)\nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj} \tag{3.4.7}$$

$$M^{(k)}_{ia} = M^{(k)}_{ia} - \left(\frac{\bar{M}^{(k)}_{ia}}{\left(\bar{M}^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia}+\delta}\right)\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \tag{3.4.8}$$

Kadangi prieš algoritmo vykdyma apibrėžiam, kad $\delta > 0$, tai tam, kad lygčių (3.4.7) ir (3.4.8) vardikliai nebūtų lygūs 0, reikia pareikalauti tik, kad sugeneruotų matricų elementai būtų neneigiami, t.y. $M^{(1)}_{ia} \geq 0$ ir $L^{(1)}_{ia} \geq 0$, su visais i, a, b, j. Ši sąlyga gali būti įrodyta kaip teorema,

Teorema 2.2

Jei $M^{(1)}_{ia} \geq 0$ ir $L^{(1)}_{ia} \geq 0$, su visais i, a, b, j, tai tada

$$M^{(k)}_{ia} \geq 0 \text{ ir } L^{(k)}_{bj} \geq 0, \text{ su visais } i, a, b, j \text{ ir su visais } k \geq 1. \tag{3.4.9}$$

Įrodymas

Kai $k = 1$, tai $M^{(1)}_{ia} \geq 0$, $L^{(1)}_{bj} \geq 0$, su visais i, a, b, j pagal teoremą. Naudojame indukciją ir tariame, kad (3.4.9) yra teisinga k-tajame žingsnyje. Įrodysime, kad (3.4.9) tesinga ir $(k+1)$-ajame žingsnyje. $(k+1)$-ajame žingsnyje gauname, kad

$$\left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)} \bar{L}^{(k)}\right)_{bj} + \epsilon}\right) \geq 0 \qquad (3.4.10)$$

Nagrinėjame dvi situacijas:

1. Jei $\nabla_L f\left[M^{(k)}, L^{(k)}\right]_{bj} < 0$, tada pasinaudoję (3.4.10) gauname, kad

$$L^{(k+1)}_{bj} = L^{k}_{bj} - \left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T \bar{L}^{(k)}\right)_{bj} + \epsilon}\right) \nabla_L f\left[M^{(k)}, L^{(k)}\right]_{bj} \geq L^{(k)}_{bj} \geq 0$$

2. Jei $\nabla_L f\left[M^{(k)}, L^{(k)}\right]_{bj} \geq 0$, tada atsižvelgdami į (3.4.5) gauname, kad

$$\bar{L}^{(k)}_{bj} = L^{(k)}_{bj} \qquad (3.4.11)$$

Iš to, kad (3.4.5) sąlyga parodo, kad $\bar{L}^{(k)}$ komponentės yra nemažesnės už $L^{(k)}$ komponentes ir iš prielaidos, kad $\left[M^{(k)}, L^{(k)}\right]$ susideda iš neneigiamų elementų, gauname, kad

$$\left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)} \bar{L}^{(k)}\right)_{bj} + \epsilon}\right) \geq \left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj} + \epsilon}\right) \qquad (3.4.12)$$

Pasinaudoję (3.4.11) ir (3.4.12) galime užrašyti:

$$L^{(k+1)}_{bj} = L^{(k)}_{bj} - \left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)} \bar{L}^{(k)}\right)_{bj} + \varepsilon}\right) \nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj}$$

$$\ge L^{(k)}_{bj} - \left(\frac{\bar{L}^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon}\right) \nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj}$$

$$= L^{(k)}_{bj} - \left(\frac{L^{(k)}_{bj}}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon}\right) \nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj}$$

$$= L^{(k)}_{bj} \frac{\left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon - \nabla_L f\left(M^{(k)}, L^{(k)}\right)_{bj}}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon}$$

$$= L^{(k)}_{bj} \frac{\left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon - \left(\left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)}\right)_{bj} + \left(\left(M^{(k)}\right)^T N\right)_{bj}\right)}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon}$$

$$= L^{(k)}_{bj} \frac{\left(\left(M^{(k)}\right)^T N\right)_{bj} + \varepsilon}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj} + \varepsilon} \ge 0$$

Analogiškai galima parodyti, kad $M^{(k)}_{ia} \ge 0$ su visais i, a, ir su visais $k \ge 1$. Įrodymo pradžioje buvo padaryta prielaida, kad (3.4.9) teisinga k-tąjame žingsnyje, tada $(k+1)$-ąjame žingsnyje gauname, kad

$$\left(\frac{\bar{M}^{(k)}_{ia}}{\left(\bar{M}^{(k)} L^{(k+1)} (L^{(k+1)})^T\right)_{ia}}\right) \ge 0 \qquad (3.4.13)$$

Nagrinėjame dvi situacijas:

1. Jei $\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} < 0$, tada pasinaudoję (3.4.13) gauname, kad

$$M^{(k+1)}_{ia} = M^{(k)}_{ia} - \left(\frac{\bar{M}^{(k)}_{ia}}{\left(\bar{M}^{(k)} L^{(k+1)} (L^{(k+1)})^T\right)_{ia} + \varepsilon}\right) \nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \ge M^{(k)}_{ia} \ge 0$$

2. Jei $\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \ge 0$, tada atsižvelgdami į (3.4.6) gauname, kad

$$\bar{M}^{(k)}_{ia} = M^{(k)}_{ia} \qquad (3.4.14)$$

Jau buvo įrodyta, kad $L^{(k)}_{bj} \ge 0$ su visais b, j. Pasinaudojame (3.4.6) ir gauname, kad

$$\left(\frac{\bar{M}^{(k)}_{ia}}{\left(\bar{M}^{(k)} L^{(k+1)} (L^{(k+1)})^T\right)_{ia} + \varepsilon}\right) \le \left(\frac{\bar{M}^{(k)}_{ia}}{M^{(k)} L^{(k+1)} (L^{(k+1)})^T + \varepsilon}\right) \qquad (3.4.15)$$

Pasinaudojame (3.4.14) ir (3.4.15) ir gauname, kad

$$
\begin{aligned}
M_{ia}^{(k+1)} &= M_{ia}^{(k)} - \left(\frac{\overline{M}_{ia}^{(k)}}{\left(\overline{M}^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia} + \delta}\right)\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \\
&\geq M_{ia}^{(k)} - \left(\frac{\overline{M}_{ia}^{(k)}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia} + \delta}\right)\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \\
&= M_{ia}^{(k)} - \left(\frac{M_{ia}^{(k)}}{\left(M^{(k)}L^{(k+1)}(L^{(k+1)})^T\right)_{ia} + \delta}\right)\nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia} \\
&= M_{ia}^{(k)}\frac{\left(M^{(k)}L^{(k+1)}\left(L^{(k+1)}\right)^T\right)_{ia} + \delta - \nabla_M f\left(M^{(k)}, L^{(k+1)}\right)_{ia}}{M^{(k)}L^{(k+1)}(L^{(k+1)})^T + \delta} \\
&= M_{ia}^{(k)}\frac{\left(M^{(k)}L^{(k+1)}\left(L^{(k+1)}\right)^T\right)_{ia} + \delta - \left(M^{(k)}L^{(k+1)}\left(L^{(k+1)}\right)^T\right)_{ia} + \left(N\left(L^{(k+1)}\right)^T\right)_{ia}}{M^{(k)}L^{(k+1)}(L^{(k+1)})^T + \delta} \\
&= M_{ia}^{(k)}\frac{\left(N\left(L^{(k+1)}\right)^T\right)_{ia} + \delta}{M^{(k)}L^{(k)}(L^{(k+1)})^T + \delta} \geq 0
\end{aligned}
$$

Taigi padarę prielaidą, kad teoremos tvirtinimas (3.4.9) patenkintas k-tąjame žingsnyje gavome, kad (3.4.9) patenkintas ir $(k+1)$-ąjame žingsnyje. Vadinasi $M_{ia}^k \geq 0$ ir $L_{bj}^k \geq 0$ su visais i, a, b, j ir su visais $k \geq 1$.

$\square$

2.5 Alternuojantis NMF algoritmas, naudojant mažiausių kvadratų metodą

Viena iš NMF algoritmų klasių yra alternuojančių NMF algoritmų, naudojant mažiausių kvadratų metodą, klasė (*ang. Alternating Least Squares Algorithms*) (NMF/ALS). Šio algoritmo idėja yra ta, kad turint vieną matricą, kita matrica gali būti rasta mažiausių kvadratų metodu. Jei turime matricą $N \in \mathbb{R}^{m\times n}$, tai nustačius faktorizacijos rangą $r \leq \max(m, n)$, NMF/ALS algoritmas pradedamas inicializuojant neneigiamą matricą $L \in \mathbb{R}^{r\times n}$. Tada kol pasiekiamas konvergavimas yra atliekamos iteracijos, kurių metu sprendžiami du minimizavimo uždaviniai:

$$\min_M \|L^T M^T - N^T\|_F^2, \qquad (3.5.1)$$

kur matrica $L \geq 0$ fiksuota, o uždavinys sprendžiamas M atžvilgiu. Fiksuojame M ir sprendžiame

$$\min_L \|ML - N\|_F^2, \qquad (3.5.2)$$

kur matrica $M \geq 0$, o uždavinys sprendžiamas L atžvilgiu. Analogiškai algoritmą vykdyti galima inicializuojant $M \in \mathbb{R}^{m\times r}$ ir iteracijas pradedant vykdyti nuo (3.5.2). Ch. J. Lin [4] įrodė, kad bet kuris sekos $\{M, L\}$ taškas, rastas atliekant NMF/ALS algoritmą yra stacionarus lygties (3.1.3) taškas. Minimizuojame (3.5.1):

$$\frac{\partial tr\left(L^T M^T - N^T\right)^2}{\partial M} = \left(L^T M^T - N^T\right)\frac{\partial tr L^T M^T}{\partial M}$$
$$= \left(L^T M^T - N^T\right) L^T = 0$$

Iš čia gauname, kad $L^T M^T = N^T$. Abi lygybės puses iš kairės padauginame iš $\left(LL^T\right)^{-1} L$ ir gauname, kad

$$M = \left(\left(LL^T\right)^{-1} L N^T\right)^T. \qquad (3.5.3)$$

Minimizuojame (3.5.2):

$$\frac{\partial tr\,(ML - N)^2}{\partial L} = (ML - N)\frac{\partial ML}{\partial L}$$
$$= (ML - N)M^T = 0$$

Iš čia seka, kad $M^T M L = M^T N$. Abi lygybės puses iš kairės padauginame iš $\left(M^T M\right)^{-1}$ ir gauname, kad

$$L = \left(M\left(M^T M\right)^{-1}\right)^T N. \qquad (3.5.4)$$

Algoritmas 3

NMF/ALS algoritmas

---

1. Turime neneigiamą duomenų matricą $N \in R^{m \times n}$;
2. Nustatom faktorizacijos rangą $r < \max(m, n)$ ir atsitiktinai generuojame matricas $L \in R^{r \times n}$, $M \in R^{m \times r}$;
3. Vykdom iteracijas:
   - Fiksavę M, sprendžiame (3.5.4). Visus gautus neigiamus matricos L elementus prilyginame 0. Fiksuojame L;
   - Sprendžiame (3.5.3). Visus gautus neigiamus matricos M elementus prilyginame 0. Fiksuojame M;
4. Jeigu $\sum_{i=1}^{m}\sum_{j=1}^{n}\left(N_{ij} - (ML)_{ij}\right)^2 \leq \varepsilon$ (ε - labai mažas teigiamas skaičius) algoritmą stabdom, priešingu atveju grįžtam prie 3 žingsnio.

Pateiktamame algoritme naudojamas paprasčiausias neneigiamumo užtikrinimo būdas – neigiami elementai prilyginami 0. Toks metodas sąlygoja išretinimą.

2.6 LASSO atnaujinimas dauginant

Kaip jau buvo minėta, atnaujinimo dauginant metodas yra naudojamas NMF uždaviniams spręsti [3]. Patobulinus šį metodą, jis buvo pritaikytas pusiau neneigiamai matricos faktorizacijai atlikti [15]. M. Mørup ir L. H. Clemmenson [18] parodė, kad dauginamojo atnaujinimo (*ang. Multiplicative Update*) (MU) metodą galima pritaikyti sprendžiant ir LASSO (*ang. Least Absolute Shrinkage and Selection Operator*) uždavinius. Iš 2.1 skyrelio žinome, kad NMF tikslo funkcija

$$L(M;L) = \frac{1}{2}\|N - ML\|_F^2.$$

LASSO yra tiesinės regresijos metodas, kuris minimizuoja paklaidų kvadratų sumą su tam tikrais apribojimais koeficientų absoliutinių didumų sumai, dar vaidinamai L_1 - normai [19], t.y.

$$\beta = \arg\min \frac{1}{2}\|Y - \beta X\|_F^2;\ kur \sum_i |\beta_i| \le t, \tag{3.6.1}$$

arba ekvivalentiškai

$$\beta = \arg\min \frac{1}{2}\|Y - \beta X\|_F^2 + \lambda \sum_i |\beta_i|. \tag{3.6.2}$$

Šiems ekvivalentiškiems minimizavimo uždaviniams spręsti yra siūloma [18] įversti neneigiamus kintamuosius β^+ ir β^-, tokius, kad $\beta_m = \beta^+ - \beta^-$, tada uždavinys gali būti sprendžiamas pagal $\tilde{X} = \begin{pmatrix} X \\ -X \end{pmatrix}$ ir $\tilde{\beta} = [\beta^+; \beta^-]$, atsižvelgiant į apribojimą $\tilde{\beta} \ge 0$. M.Mørup ir L.H.Clemmenson [18] parodė, kad LASSO uždavinius galima spręsti naudojant atnaujinimo dauginant metodus (2.3.1 skyrelis)

Pasinaudokime (3.4.3) ir žingsnio dydį pažymėkime $\eta_{bj}^{(k)}$. Tada (3.4.3) galima užrašyti taip:

$$L_{bj}^{(k+1)} = L_{bj}^{(k)} + \eta_{bj}^{(k)} \left[\left(\left(M^{(k)}\right)^T N \right)_{bj} - \left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)} \right)_{bj} \right] \tag{3.6.3}$$

Tarkime, kad $L_{bj}^{(k)} - \eta_{bj}^{(k)} \left(\left(M^{(k)}\right)^T M^{(k)} L^{(k)} \right)_{bj} = 0$. Iš čia gauname, kad

$$\eta_{bj}^{(k)} = \frac{L_{bj}^{(k)}}{\left((M^{(k)})^T M^{(k)} L^{(k)} \right)_{bj}} \tag{3.6.4}$$

Tada (3.6.3) galima užrašyti taip:

$$
\begin{aligned}
L_{bj}^{(k+1)} &= \left(\frac{L_{bj}^{(k)}}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj}}\right)\left(\left[M^{(k)}\right]^T N\right)_{bj} \\
&= L_{bj}^{(k)} \left(\frac{\left(\left[M^{(k)}\right]^T N\right)_{bj}}{\left((M^{(k)})^T M^{(k)} L^{(k)}\right)_{bj}}\right)
\end{aligned}
\tag{3.6.5}
$$

Pasinaudokime (3.4.1) ir pažymėkime:

$$
\left(\left[M^{(k)}\right]^T M^{(k)} L^{(k)}\right)_{bj} = \frac{\partial L(M;L)^+}{\partial L_{bj}}
$$

$$
\left(\left[M^{(k)}\right]^T N\right)_{bj} = \frac{\partial L(M;L)^-}{\partial L_{bj}}
$$

Taigi (3.6.5) galima užrašyti taip:

$$
L_{bj}^{(k+1)} = L_{bj}^{(k)} \left(\frac{\frac{\partial L(M;L)^-}{\partial L_{bj}}}{\frac{\partial L(M;L)^+}{\partial L_{bj}}}\right). \tag{3.6.6}
$$

Analogiškai tą patį galima padaryti pasinaudojant (3.4.4) ir žingsnio dydį pažymint $\mu_{ia}^{(k)}$. Tada (3.4.4) galima užrašyti, kaip:

$$
M_{ia}^{(k+1)} = M_{ia}^{(k)} + \mu_{ia}^{(k)} \left(\left(N\left[L^{(k+1)}\right]^T\right)_{ia} - \left(M^{(k)} L^{(k+1)} \left[L^{(k+1)}\right]^T\right)_{ia}\right) \tag{3.6.7}
$$

Tariame, kad $M_{ia}^{(k)} - \mu_{ia}^{(k)} \left(M^{(k)} L^{(k+1)} \left[L^{(k+1)}\right]^T\right)_{ia} = 0$ ir gauname, kad

$$
\mu_{ia}^{(k)} = \left(\frac{M_{ia}^{(k)}}{\left(M^{(k)} L^{(k+1)} (L^{(k+1)})^T\right)_{ia}}\right). \tag{3.6.8}
$$

Tada (3.6.7) galima užrašyti taip:

$$
\begin{aligned}
M_{ia}^{(k+1)} &= \left(\frac{M_{ia}^{(k)}}{\left(M^{(k)} L^{(k+1)} (L^{(k+1)})^T\right)_{ia}}\right)\left(N\left[L^{(k+1)}\right]^T\right)_{ia} \\
&= M_{ia}^{(k)} \left(\frac{\left(N\left[L^{(k+1)}\right]^T\right)_{ia}}{\left(M^{(k)} L^{(k+1)} (L^{(k+1)})^T\right)_{ia}}\right)
\end{aligned}
\tag{3.6.9}
$$

Pasinaudokime (3.4.2) ir pažymėkime:

$$
\left(N\left[L^{(k+1)}\right]^T\right)_{ia} = \frac{\partial L(M;L)^-}{\partial M_{ia}}
$$

$$
\left(M^{(k)} L^{(k+1)} \left[L^{(k+1)}\right]^T\right)_{ia} = \frac{\partial L(M;L)^+}{\partial M_{ia}}
$$

Vadinasi (3.6.9) galima užrašyti taip:

$$M_{ia}^{(k+1)} = M_{ia}^{(k)} \left(\frac{\frac{\partial L(M;L)^-}{\partial M_{ia}}}{\frac{\partial L(M;L)^+}{\partial M_{ia}}} \right) \qquad (3.6.10)$$

M. Mørup ir L. H. Clemmenson [18] siūlo įvesti parametrą α:

$$M_{ia}^{(k+1)} = M_{ia}^{(k)} \left(\frac{\frac{\partial L(M;L)^-}{\partial M_{ia}}}{\frac{\partial L(M;L)^+}{\partial M_{ia}}} \right)^{\alpha} \qquad (3.6.11)$$

Jeigu $\frac{\partial(M;L)}{\partial M_{ia}} > 0$, t.y. jeigu $\frac{\partial(M;L)^+}{\partial M_{ia}} > \frac{\partial(M;L)^-}{\partial M_{ia}}$, tai $M_{ia}^{(k+1)}$ mažės ir atvirkščiai, jeigu $\frac{\partial(M;L)}{\partial M_{ia}} < 0$. α vadinamas „žingsnio dydžio“ parametru, kuris gali būti keičiamas. Kai $\alpha \to 0$, prie sprendinio artėjama labai mažais žingsniais. Naudojant NMF imama, kad $\alpha = 1$ [3], tuo tarpu Semi – NMF atveju imama $\alpha = 0{,}5$ [15].

(3.1.1) galima užrašyti kaip

$$N \approx \phi L,$$

kur $N \in R^{m \times n}$; $\phi \in R^{m \times r}$; $L \in R^{r \times n}$. M. Mørup ir L. H. Clemmenson [18] ima $m = 1$ ir tikslo funkciją užrašo taip:

$$C_{LASSO} = \frac{1}{2}\|N - \tilde{\phi}\tilde{L}\|_F^2 + \lambda \sum_i \tilde{\phi}_i, \qquad (3.6.12)$$

kur $\tilde{L} = \begin{bmatrix} L \\ -L \end{bmatrix}$ ir $\tilde{\phi} = \left[\phi^+;\ \phi^-\right]$. Tada tikslo funkcijos (3.6.12) išvetinė

$$\frac{\partial C_{LASSO}}{\partial \tilde{\phi}} = -\left(N - \tilde{\phi}\tilde{L}\right)\tilde{L}^T + \lambda \cdot 1 \qquad (3.6.13)$$

čia

$$N - \tilde{\phi}\tilde{L} = N - \phi^+ L + \phi^- L = N - \left(\phi^+ - \phi^-\right) L = N - \phi L \qquad (3.6.14)$$

$$N\tilde{L}^T = N\left[L^T;\ -L^T\right] = \left[NL^T;\ -NL^T\right] \qquad (3.6.15)$$

$$\tilde{L}\tilde{L}^T = \begin{bmatrix} L \\ -L \end{bmatrix} \left[L;\ -L\right]^T = \begin{bmatrix} LL^T & -LL^T \\ -LL^T & LL^T \end{bmatrix} \qquad (3.6.16)$$

Imam $\alpha = 0{,}5$ ir pasinaudojame (3.6.11) [18]:

$$\phi_j^+ \leftarrow \phi_j^+ \sqrt{\frac{\left[NL^T\right]_+ + \phi^- \left[LL^T\right]_- + \phi^- \left[LL^T\right]_+}{\left[NL^T\right]_- + \phi^+ \left[LL^T\right]_+ + \phi^- \left[LL^T\right]_-}} \qquad (3.6.17)$$

$$\AE_{ij} \leftarrow \AE_{ij}\sqrt{\frac{\left[NL^T\right]^- + \AE\left[LL^T\right]^+ + \AE\left[LL^T\right]^-}{\left[NL^T\right]^+ + \AE\left[LL^T\right]^- + \AE\left[LL^T\right]^+}} \qquad (3.6.18)$$

kur $[K]^+$ ir $[K]^-$ reiškia teigiamas ir neigiamas matricos K dalis.

2.7 Kiti matricų faktorizacijos metodai

2.7.1 Pusiau neneigiama matricos faktorizacija

Ankstesniuose skyreliuose buvo aprašytos matricų faktorizacijos, kai naudojamas neneigiamumo apribojimas, t.y. duomenų matrica N ir faktorizacijos pagalba gaunamos matricos M ir L turėjo būti neneigiamos. M. I. Jordan [15] pusiau neneigiamą matricos faktorizaciją (*ang. Semi – Nonnegative Matrix factorization*) (Semi - NMF) apibrėžia, kaip metodą, kur duomenų matricos elementai gali būti tiek neigiami, tiek teigiami, iššifravimo matrica L turi būti neneigiama, kai tuo tarpu transformacijų matricai M netaikomi jokie apribojimai ženklams. Lygybė (3.7.1) vadinama pusiau neneigiama matricos faktorizacija

$$N \approx ML, \qquad (3.7.1)$$

kur $L \geq 0$, o matricų N, M elementams jokie apribojimai netaikomi.

Semi – NFM gali būti būti nadojamas klasterizavimo uždaviniuose. Pavyzdžiui, K-vidurkių klasterizavimo uždaviniuose. Tegu $M = (m_1; \dots; m_r)$ yra klasterių centrai, o L yra klasterių indikatoriai, t.y. $l_{rj} = 1$, jei N_j priklauso klasteriu c_r ir $l_{rj} = 0$, kitu atveju. K-vidurkių klasterizavimo tikslo funkcija galima užrašyti taip [15]

$$J_{K-means} = \sum_{j=1}^{n}\sum_{r=1}^{K} \|N_j - m_r\|^2 l_{rj} = \|N - ML\|^2$$

Taigi K – vidurkių tikslo funkciją galima užrašyti kaip ir NMF tikslo funkciją.

Matricos L inicializavimui M. I. Jordan [15] siūlo atlikti K-vidurkių klasterizavmą. Kaip jau buvo minėta, $l_{rj} = 1$, jei $N_j \in c_r$ ir $l_{rj} = 0$, jei $N_j \notin c_r$. Lygybė (3.5.3) gali būi užrašyta taip:

$$M = NL^T\left(LL^T\right)^{-1}, \qquad (3.7.2)$$

kur $L \geq 0$, todėl ir $r \times r$ matrica $LL^T \geq 0$. Jeigu matrica LL^T nėra singuliari, t.y. jeigu $|LL^T| \neq 0$, tai ji yra apverčiama ir galime rasti jos atvirkštinę $\left(LL^T\right)^{-1}$, jei $|LL^T| = 0$, randame jos pseudoatvirkštinę. Fiksuojame M ir atnaujiname L [15]:

$$L_{bj}^{k+1} = L_{bj}^{k}\sqrt{\frac{(M^T N)^{+}_{bj} + \left[(M^T M)^{+} L\right]_{bj}}{(M^T N)^{+}_{bj} + \left[(M^T M)^{+} L\right]_{bj}}}, \qquad (3.7.3)$$

čia $j = 1, \ldots, n;\ b = 1, \ldots, r$ ir $A^{+}_{bj} = (|A_{bj}| + A_{bj})/2;\ A^{+}_{bj} = (|A_{bj}| - A_{bj})/2$ yra atitinkamai teigiama ir neigiama matricos A dalys. M. I. Jordan [15] įrodė, kad fiksavus M išreikštą kaip (3.7.2), tikslo funkcija (3.1.3) monotoniškai mažėja (t.y. ji yra nedidėjanti), o fiksavę L, iš (3.7.2) gauname, kad M yra optimalus sprendinys minimizuojantis tikslo funkciją (3.1.3).

2.7.2 Išgaubta neneigiama matricos faktorizacija

Jei Semi – NMF atveju netaikomi jokie apribojimai transformacijų matricai M, tai išgaubtos neneigiamos matricos faktorizacijos atveju (*ang. Convex Nonnegative Matrix Factorization*) (Convex - NMF) matricos M r - tąjį stulpelis apibrėžiamas taip:

$$M_r = w_{1r} N_1 + \cdots + w_{nr} N_n = N w_r \qquad (3.7.4)$$

arba

$$M = NW, \text{ kur } w_{jr} \geq 0 \qquad (3.7.5)$$

Tokia matricos M išraiška leidžia stulpelius M_r interpretuoti kaip duomenų matricos stulpelių N_j svertines sumas. Convex – NMF naudoja tiek neneigiamas, tiek neapribotų ženklų matricas

$$N \approx NWL, \qquad (3.7.6)$$

kur $W, L \geq 0$, o matricos N elementų ženklams apribojimai nėra taikomi. Uždavinys yra minimizuoti tikslo funkciją

$$L(W, L) = \frac{1}{2}\|N - NWL\|^2. \qquad (3.7.7)$$

Matricų W ir L inicializavimą galima atlikti dviem būdais:

1. Atliekant K – vidurkių klasterizavimą;
2. Tariant, kad jau turime NMF arba Semi – NMF sprendinius M ir L.

Pirmuoju atveju tariame, kad turime klasterių indikatorius $H = (H_1, \ldots, H_r)$, kur $H_{jr} = \{0, 1\};\ j = 1, \ldots, n$ ir apibrėžiame $L^{(0)} = H^T + 0.2 \cdot 1$, kur 1 yra vienetų matrica. Klasterių centrai gali būti rasti iš $M_r = NH_r/n_r$ arba $M = NHD_n^{-1}$, kur $D_n = \mathrm{diag}(n_1, \ldots, n_r)$. Tada iš (3.7.5) matome, kad $W = HD_n^{-1}$, prie kiekvieno H elemento pridėdame 0.2 ir gauname

$$W = (H + 0.2 \cdot 1) D_n^{-1} \qquad (3.7.8)$$

Antruoju atveju tariame, kad matrica L yra žinoma, todėl galima užrašyti, kad $L^{(0)} = L + 0{,}2 \cdot 1$. Tam, kad būtų rasta W, reikia išspręsti (3.7.6), iš kur gauname, kad $W = L^T \left(L L^T\right)^{-1}$. Kadangi W turi būti neneigiama, pažymėkime, kad $W^{(0)} = W^+ + 0{,}2 \cdot 1\,[W^+]$, kur $[A] = \sum_{ij} |A_{ij}| / \|A\|_{(0)}$, o $\|A\|_{(0)}$ yra neneigiamų elementų skaičius matricoje A. Matrica L yra randama iš [15]:

$$L_{bj}^{k+1} = L_{bj}^{k} \sqrt{\frac{\left[W^T (N^T N)^+\right]_{bj} + \left[W^T (N^T N)^- W L\right]_{bj}}{\left[W^T (N^T N)^-\right]_{bj} + \left[W^T (N^T N)^+ W L\right]_{bj}}} \qquad (3.7.9)$$

Ši lygybė gali būti gaunama iš (3.7.3) M pakeičiant NW. Matricos W atnaujinimui naudojame

$$W_{jb}^{(k+1)} = W_{jb}^{(k)} \sqrt{\frac{\left[(N^T N)^+ L^T\right]_{jb} + \left[L^T L (N^T N)^- W\right]_{jb}}{\left[(N^T N)^- L^T\right]_{jb} + \left[L^T L (N^T N)^+ W\right]_{jb}}} \qquad (3.7.10)$$

Galima įrodyti [15], kad fiksavus L ir sprendžiant (3.7.10) tikslo funkcija (3.7.7) yra nedidėjanti, t.y. monotoniškai mažėja.

2.7.3 Neneigiama matricos tri - faktorizacija

Kai duomenų matricos N eilutės ir stulpeliai klasterizuojami vienu metu, yra atliekama neneigiama matricos N tri – faktorizacija [16] (*ang. Nonnegative Matrix Tri - Factorization*) (NMTF). NMTF yra apibrėžiama taip

$$N \approx M S L, \qquad (3.7.11)$$

kur matricos $M \in R^{m \times r}$; $S \in R^{r \times r}$; $L \in R^{r \times n}$ yra neneigiamos [17]. Tokiu atveju yra sprendžiamas minimizavimo uždavinys

$$\min_{M \geq 0;\ L \geq 0;\ S \geq 0} \|N - M S L\|^2; \qquad (3.7.12)$$

kur $M^T M = I$; $L^T L = I$. Tikslo funkcija galima užrašyti taip

$$L(M; L; S) = \sum_{i=1}^{r} \sum_{j=1}^{r} M_{ia} S_{ab} L_{bj} \qquad (3.7.13)$$

Matricų L; M; S atnaujinimo taisyklės [16]:

$$L_{bj}^{(k+1)} = L_{bj}^{(k)} \sqrt{\frac{(S^T M^T N)_{bj}}{(S^T M^T N L^T L)_{bj}}} \qquad (3.7.14)$$

$$M_{ia}^{(k+1)} = M_{ia}^{(k)} \sqrt{\frac{(N L^T S^T)_{ia}}{(M M^T N L^T S^T)_{ia}}} \qquad (3.7.15)$$

$$S_{ba}^{(k+1)} = S_{ba}^{(k)} \sqrt{\frac{(M^T N L^T)_{ba}}{(M^T M S L L^T)_{ba}}} \tag{3.7.16}$$

2.7.4 Branduolių neneigiama matricos faktorizacija

Tarkime, kad $N_j \to \phi(N_j)$, arba $N = \phi(N) = (\phi(N_1), \ldots, \phi(N_n))$. Tokiu atveju yra sudėtinga atlikti skaičiavimus paprastu NMF arba Semi – NMF metodais, nes $\phi(N) = ML$ ir matricos M ir L priklauso nuo funkcijos $\phi(\cdot)$. Tačiau panaudojant Convex – NMF (3.7.6) [15]galime užrašyti

$$\phi(N) \approx \phi(N) M L. \tag{3.7.17}$$

Sprendžiam minimizavimo uždavinį

$$\min_{W \geq 0;\ L \geq 0} \|\phi(N) - \phi(N) M L\|^2. \tag{3.7.18}$$

Nesunku patikrinti [15], kad tikslo funkcija

$$\|\phi(N) - \phi(N) M L\|^2 = \mathrm{Tr}\left(\phi(N)^T \phi(N) - 2L\phi^T(N) W + W^T \phi^T(N)\phi(N) W L L^T\right)$$

priklauso tik nuo branduolio $K = \phi^T(N)\phi(N)$.

2.8 NMF panaudojimas duomenų stuktūros nustatymui, kai transformacijų matrica M fiksuota

NMF metodu neneigiama $m \times n$ matavimo matrica N gali būti suskaidyta į dvi taip pat neneigiamas matricas, kaip parodyta (3.1.1). Matricą M galima vadinti transformacijų matrica, kuri originalius duomenis transformuoja į mažesnės (iš anksto apibrėžtos) dimensijos erdvę, o matricą L - iššifravimo matrica, kurios kiekvienas stulpelis atitinka originalių duomenų matricos atitinkamą stulpelį, tik informacija apie duomenis yra pateikiama nebe m-matėje, bet r-matėje erdvėje ($r < m$). Iš (3.1.1) nesunkiai galime išsireikšti L:

$$L = \left(M^T M\right)^{-1} M^T N \tag{3.8.1}$$

Čia matrica $M^T M$ yra $m \times m$ matavimo kvadratinė matrica, todėl ji yra apverčiama, t.y. galima rasti jai atvirkštinę matricą. Taigi galime sakyti, kad fiksavus transformacijų matricą M, išraišką (3.8.1) galima pritaikyti naujiems duomenims, aišku, jeigu sutampa matavimai. Pavyzdžiui, turime n pacientų, kuriems buvo atlikta m medicininių matavimų, pasirinkę faktorizacijos rangą r ir pritaikę NMF algoritmą, gauname matricą M, kuri visus n stebėjimus atvaizduoja r-matėje erdvėje. Prijunkime papildomus k pacientus, kuriems buvo atlikti analogiški matavimai. Naujų duomenų matrica yra $m \times k$

matavimo matrica $N_{(k)}$. Tikslas yra sumažinti $N_{(k)}$ dimensiją taip, kad būtų galima nustatyti duomenų struktūrą. Tą padaryti galima vienu iš metodų:

1. Atsitiktinai generuojant L ir vykdant iteracijas, kaip aprašyta 2.3 skyrelyje. Tik čia matrica M yra fiksuota.
2. Taikant (3.8.1) ir randant $L^{(k)} = \left(M^T M\right)^{-1} M^T N^{(k)}$, kur $N^{(k)}$ yra naujų duomenų matrica, o matrica M yra fiksuota.

Pirmasis metodas yra vadinamas *iteraciniu* metodu, tuo tarpu antrasis – *tiesioginiu*, nes sprendinys gaunamas tiesiogiai, t.y. nevykdant iteracijų. Tačiau naudojant antrąjį metodą dėl atvirkštinės matricos panaudojimo formulėje (3.8.1) galimos neigiamos L reikšmės, todėl naujų duomenų nebegalime taip pat interpretuoti kaip pradinių, nors ir atitinkamoms grupėms jie priskiriami pakankamai tiksliai. Šiai problemai spręsti buvo panaudotas pirmasis - iteracinis metodas, aprašomas kitame skyrelyje.

2.8.1 Dimensijos mažinimo algoritmas, kai matrica M fiksuota. Atsitiktinės inicializacijos taikymas

Tarkime, kad turime $m \times n$ matavimo matricą N, kur kiekvienas stulpelis $N_j;\ j = 1, \ldots, n$ yra atskiras stebėjimas, o m yra – matavimai. Matricą N_{tr} pavadinkime *apmokymų matrica*. Taip pat tarkime, kad turime testinių duomenų $m \times k$ matavimo matricą N_{te}, susidedančią iš k stebėjimų ir m matavimų. Tikslas yra duomenis iš matricos N_{tr} panaudoti algoritmo apmokymui ir, fiksavus transformacijų matricą M, rasti testinių (naujų) duomenų struktūrą. Pirmiausia nusistatom faktorizacijos rangą $r < \max(m, n)$ ir atliekam NMF algoritmą, kaip aprašytą 2.3 skyrelyje, fiksuojame transformacijų matricą M. Matrica M originalius duomenis iš m-matės erdvės transformuoja į r-matę, išlaikydama jų struktūrą. Todėl ją pritaikysime ir naujų duomenų atvaizdavimui r-matėje erdvėje. Tokiu būdu inicializuoti reikės tik iššifravimo matricą L_{te}. Naujų duomenų iššifravimo matrica L_{te}^{new} randama kaip parodyta (3.3.1).

Algoritmas 4

Diemensijos mažinimas taikant atsitiktinę matricos L_{te} inicializaciją

1. Turime apmokymų matricą $N_{tr} \in R^{m \times n}$ ir testinių duomenų matricą $N_{te} \in R^{m \times k}$
2. Nustatom faktorizacijos rangą $r < \max(m; n)$, atliekame NMF algoritmą su matrica N_{tr} ir fiksuojame M
3. Atsitiktinai generuojame L_{te}
4. Pritaikę fiksuotą M ir sugeneruotą L_{te}, iteruojame (3.3.1) tol, kol bus pasiektas konvergavimas. Gautoji matrica L_{te}^{new} yra testinių duomenų atitiktis r-matėje erdvėje.

2.8.2 O. Okun ir H. Priisalu siūlomas dimensijos mažinimo algoritmas, kai matrica M yra fiksuota

O. Okun ir H. Priisalu [14] naudoja kitokį metodą matricos L_{te}^{new} radimui. Siūlomame algoritme jie apjungia tiesioginį ir iteracinį metodus, tiesioginį metodą naudodami L_{te} inicializavimui. Taip pat yra siūloma normalizuoti matricos N_{tr} elementus. Tegu N_j yra m-matis vektorius, $j = 1; \ldots; n$, tada jo komponentės N_{ij} yra normalizuojamos kiekvieną jų dalinant iš atitinkamo vektoriaus didžiausio elemento, t.y. N_{ij} / N_{pj}, kur $p = \arg\max_l N_{lj}$. Tokiu būdu matricos N_{tr} elementų reikšmės pasiskirsto intervale $[0; 1]$ ir, anot autorių, greičiau yra pasiekiamas konvergavimas. Algoritmo vykdymo spartumui taip pat yra svarbus tinkamas matricų M ir L inicializavimas. Autoriai siūlo L_{te} inicializavimui rinktis ne atsitiktinį generavimą, bet prieš algoritmo vykdyma L_{te} apibrėžti kaip parodyta (3.8.1), t.y. $L_{te} = \left(M^T M\right)^{-1} M^T N_{te}$. Tada iteracijų vykdymo pradinis taškas yra arčiau siekiamo rezultato ir algoritmo konvergavimui pasiekti reikia atlikti mažiau iteracijų.

Algoritmas 5

Dimensijos mažinimo algoritmas L_{te} inicializavimui naudojant tiesiogini metodą

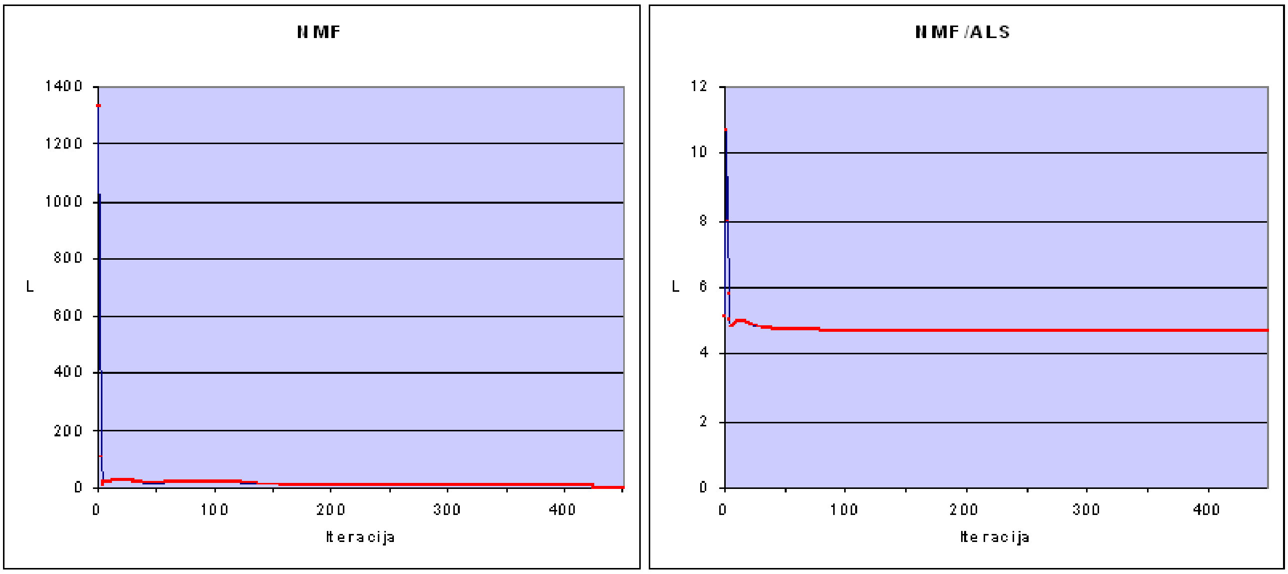
1. Normalizuojame apmokymų matricos $N_{tr} \in R^{m\times n}$ ir testinių duomenų matricos $N_{te} \in R^{m\times k}$ elementus N_{ij}, $i = 1,\ldots,m;\ j = 1,\ldots,n$ ir $N_{ia(k)}$ $i = 1,\ldots,m;\ a = 1,\ldots,k$;
2. Nustatom faktorizacijos rangą $r < \max(m,n)$, atliekame NMF algoritmą su matrica N_{tr} ir fiksuojame M;
3. Naudojant tiesiogini metodą inicializuojame L_{te}, kaip parodyta (3.8.1);
4. Pritaikę fiksuotą M ir sugeneruotą L^{te}, iteruojame (3.3.1) tol, kol bus pasiektas konvergavimas. Gautoji matrica $L^{new}_{(iter)}$ yra testinių duomenų atitiktis r-matėje erdvėje.

Taigi pagrindinė šio algoritmo idėja yra tiesioginį metodą panaudoti iniacializavimui, o po to vykdytį iteracinį metodą ir, tokiu būdu, apjungti abu metodus, vietoj to, kad būtų naudojamas tik vienas iš jų.

3. Modulių, realizuojančių NMF ir NMF/ALS algoritmus, aprašymas

Naudojantis SAS/IML® buvo sukurti atskiri moduliai „nmf“ ir „nmf_ls“, realizuojantys atititinkamai NMF ir NMF/ALS algoritmus. Modulius nėra sudėtinga naudoti bet kokių duomenų matricų faktorizacijos atlikimui. Prieš pradedant vykdyti algoritmus reikia į vartotojo susikurtą biblioteką importuoti duomenų failą, tokį, kad imtys būtų eilutėse ir apsibrėžti keletą parametrų. Tiek „nmf“, tiek „nmf_ls“ modulių vykdymui reikia nusistatyti faktorizacijos rangą ***r*** ir maksimalų iteracijų skaičių ***niter.*** „nmf“ moduliui papildomai reikia apibrėžti parametrą ***stopconv***, kuris sustabdo algoritmą, jei skaičiuojama jungiamoji matrica nepasikeičia *stopconv* kartų, ir mažą teigiamą skaičių ***e,*** kuris bus naudojamas pakeisti labai mažoms matricų M ir L reikšmėms. Be to prieš vykdant algoritmus reikia pasiruošti duomenų rinkinius taip, kad juos sudarytų tiktai tyrimui reikalingi stebėjimai ir matavimai. Modulių iškvietimui naudojamas **RUN** sakinys.

Atliekant eksperimentus pastebėta, kad abu algoritmai duoda gana panašius rezultatus, tik tikslo funkcija (3.1.3) (L) greičiau minimizuojama naudojant NMF/ALS algoritmą, nes vykdant pastarąjį algoritmą, matricų atnaujinimui yra naudojamos ne dauginamojo atnaujinimo taisyklės (3.3.1) ir (3.3.2),o matricų M ir L išraiškos, gautos minimizuojant atitinkamai (3.5.1) ir (3.5.4). Tokiu būdu iteracijos pradedamos vykdyti esant daug arčiau sprendinio, minimizuojančio tikslo funkciją.



3 pav. Minimizuojamos tikslo funkcijos (3.1.3) reikšmių kitimas naudojant NMF ir NMF/ALS algoritmus. m=21, n =46, r=2

3.1 Tyrimui panaudotų duomenų aprašymas

Tyrimui buvo panaudoti ligonių, kuriems atlikta širdies persodinimo operacijos nuo 1996 iki 2007 m., ir kuriems prijungti dirbtiniai širdies skilveliai nuo 1999 iki 2007 m. Vilniaus Universiteto ligoninės Santariškių klinikų Širdies chirurgijos centro duomenys. Iš viso buvo tiriama 61 ligonis sergantis širdies nepakankamumu. Ligonių amžiaus vidurkis 60;9 § 10;8 metų. 40 ligonių pagal pasirinktą tolimesnę gydymo taktiką, buvo suskirstyti į 1 - 3 grupes ir 21 ligonis po širdies persodinimo operacijos ir po dirbtinių širdies skilvelių prijungimo – atitinkamai į 4 ir 5 grupes:

1 grupė. 13 ligonių, kuriems atlikta aorto koronarinio šuntavimo opetacija.

2 grupė. 7 ligoniai, kuriems atlikta priekinės tarpskilvelinės šakos atkimšimas ir angioplastika.

3 grupė. 20 ligonių, kurie buvo gydyti konservatyviai.

4 grupė. 15 ligonių, kuriems buvo atlikta širdies persodinimo operacija.

5 grupė. 6 ligoniai, kuriems prijungti dirbtiniai širdies skilveliai.

1 – 3 grupių ligoniams prieš tyrimą ir 12, 24, 48 savaitėmis po pradėto tyrimo ar atliktų procedūrų buvo atlikti šie tyrimai: Minesotos Universiteto anketa „Gyvenimas sergant širdies nepakankamumu“ (*ang. Living With Heart Failure Quastionare*) (LHFQ) (Priedas Nr. 1), šešių minučių trukmės ėjimo testas, širdies echoskopija, veloergometrija, natriuretinio peptido tyrimas (NT – pro BNP). 4 grupės ligoniams šie tyrimai atlikti ne anksčiau kaip 48 savaitės po širdies persodinimo operacijos, o 5 grupės ligoniams – ne anksčiau kaip 12 savaičių po dirbtinio širdies skilvelio prijungimo.

Skaičiavimams atlikti buvo panaudoti ligonių atsakymai į LHFQ klausimyno klausimus.

3.1.1 Trumpai apie LHFQ klausimyną

Minesotos Universiteto anketa „Gyvenimas serant širdies nepakankamumu“ (LHFQ) buvo sukurta 1984 metais, siekiant įvertint širdies ligų ir jų gydymo metodų daromą įtaką paciento gyvenimo kokybei. Klausimynas susideda iš 21 klausimo, kurie leidžia nustatyti, kaip pacientas pergyvena širdies ligos sukeltus nepatogumus. Pacientai kiekvieną pablogėjimą, atsižvelgiant į tai, kiek jis trukdė jų gyvenimo kokybei, turi pažymėti įvertindami nuo 0 iki 5. Kadangi yra nagrinėjamos bendros visų klausimų sumos, tai galutinis įvertinimai gali būti nuo 0 (nėra neįgalumo) iki 105 (labai stiprus neįgalumas).

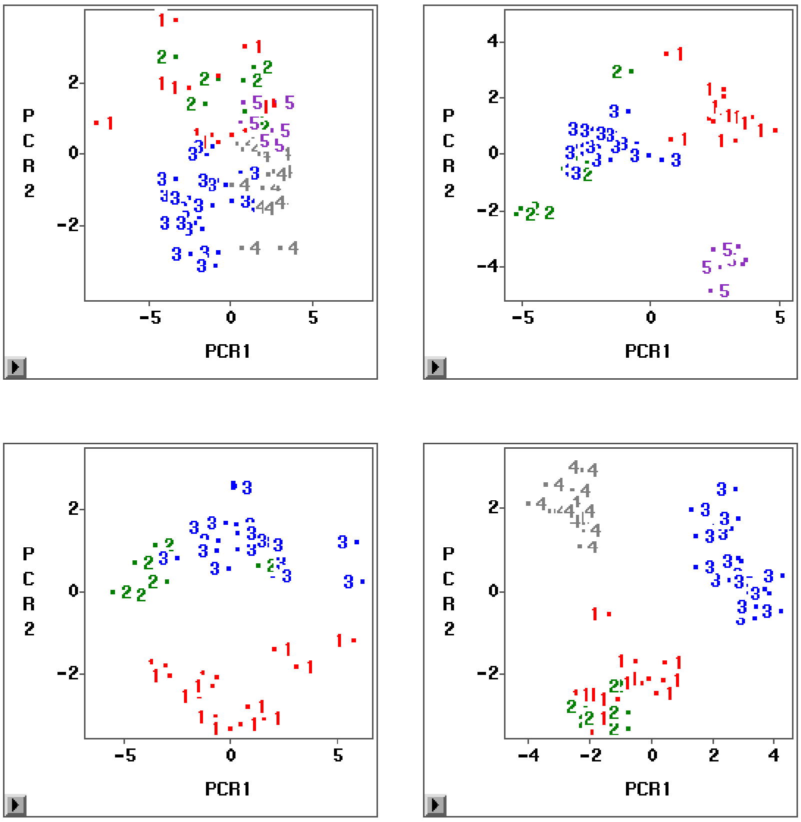
Faktorinės analizės pagalba buvo nustatyta [23], kad atsisiria du klausimų pogrūpiai – vienas, nusakantis paciento fizinius gebėjimus, susideda iš 8 labiausiai koreliuojačių klausimų, kitas, nusakantis paciento emocinę būseną – iš 5 klausimų. Atsakymų į šių pogrūpių klausimus sumos gali būti naudojamos pacientų fizinei ir emocinei būsenai vertinti. Tačiau yra teigiama [23], kad toks būdas neužtikrina efektyvaus širdies ligos ar jos gydymo poveikio paciento gyvenimo kokybei vertinimo.

Šiame darbe yra siekiama pristatyti kitokį būdą paciento gyvenimo kokybės, naudojant LHFQ klausimyną, įvertinimui.

3.2 Atlikti eksperimentai

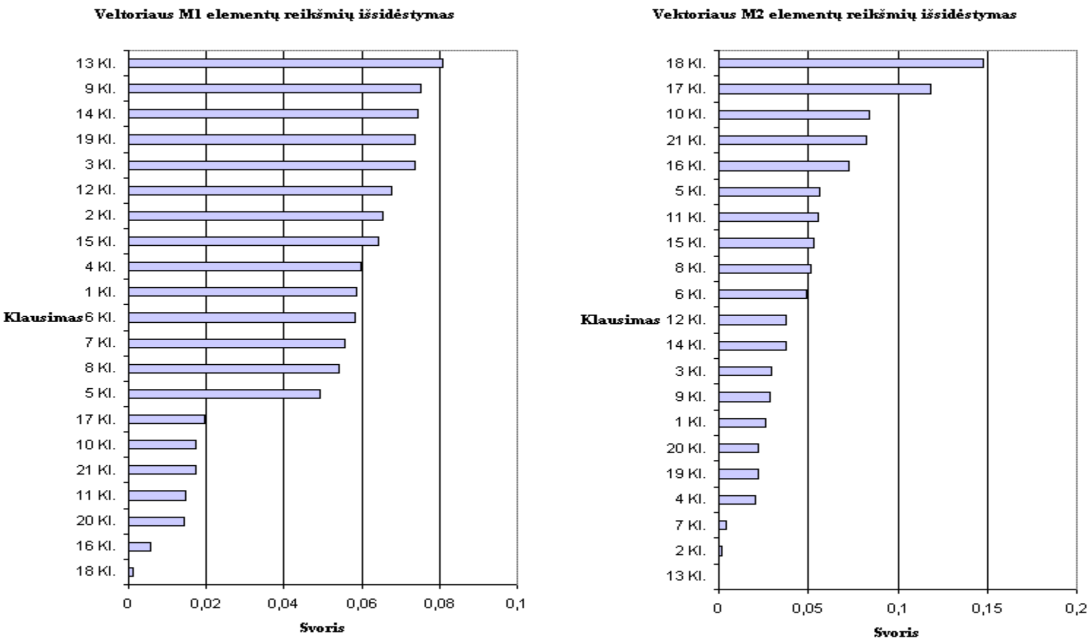
NMF ir NMF/ALS procedūros buvo prititaikytos 3.1 skyriuje aprašytiems duomenims. Naudojama duomenų matrica buvo tokia, kad kiekvienas stulpelis yra atskiras stebėjimas. Kadangi sebėjimai buvo daryti dar ir atskirais gydymo laikotarpiais (prieš gydymą, 12, 24 ir 48 savaitės po gydymo), tai eksperimentai buvo atlikti su 4 skirtingais duomenų rinkiniais.

Pagrindinių komponenčių analizė duomenims buvo atlikta naudojant SAS/INSIGHT®. Tikslas buvo parodyti, kad kai kurios gaunamos pagrindinės komponentės yra neigiamos. Kaip matome iš žemiau pateikiamo paveikslėlio (4 pav.), kai kurios gaunamos pagrindinės komponentės turi neigiamą ženklą, kas ir apsunkina jų interpretaciją.

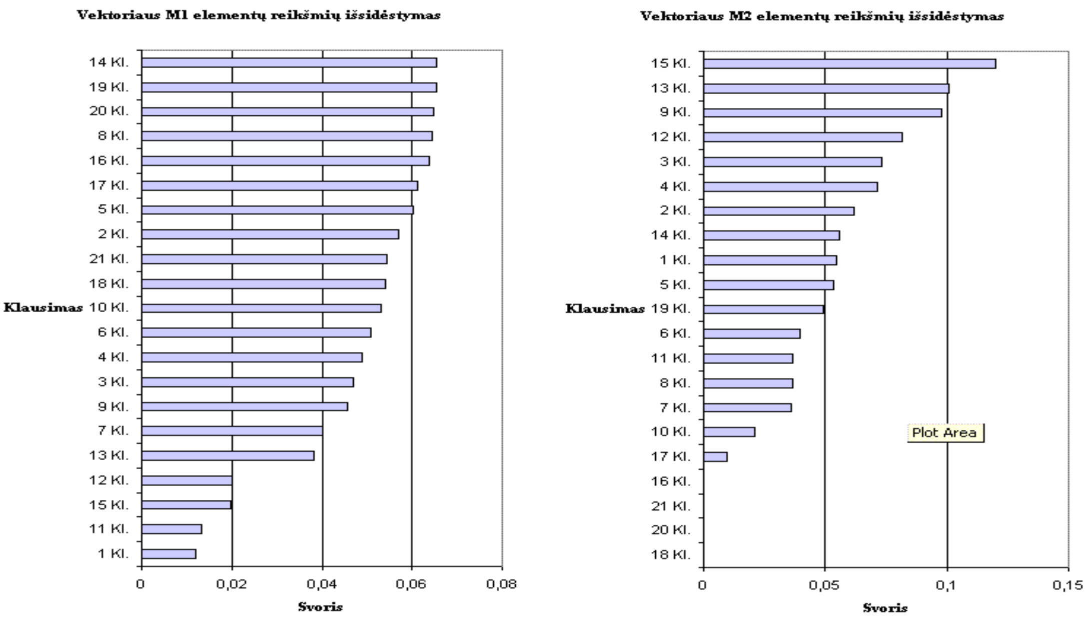


4 pav. Pirmos ir antros pagrindinių komponenčių sklaidos grafikai. Pacientų duomenys (iš kairės į dešinę) prieš gydymą(m=21, n=61), 12 (m=21, n=46), 24 (m=21, n=40) ir 48 (m=21, n=55) sav. po gydymo.

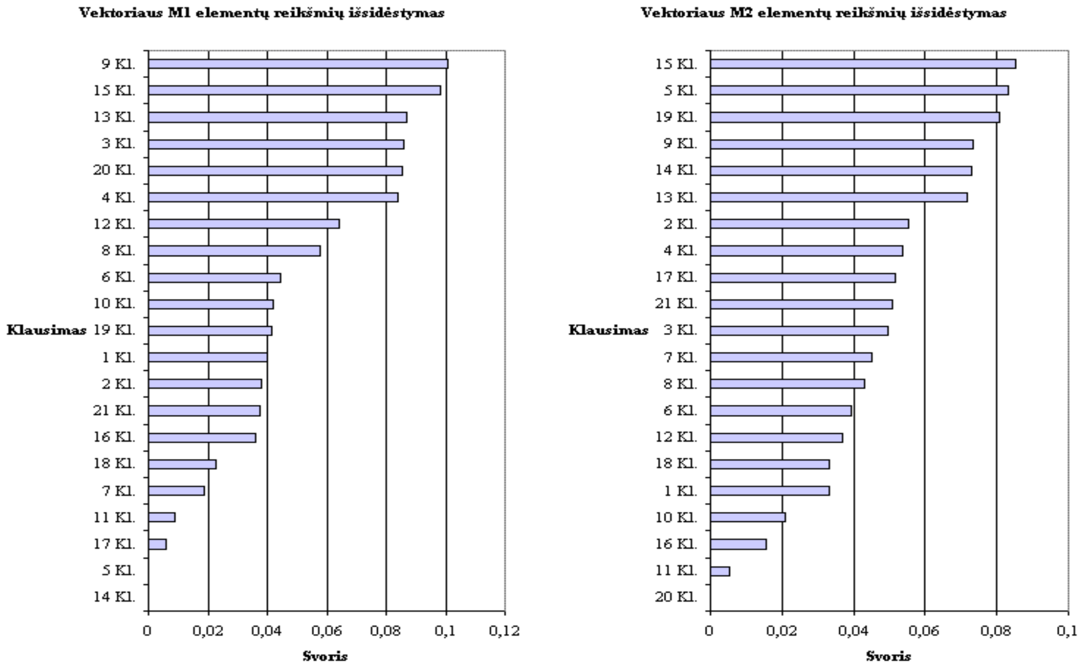
Siekiant atskirti pacientų grupes ir tuo pačiu gauti neneigiamas komponentes buvo atlikta NMF procedūra. Gauti rezultatai pasirodė gana informatyvūs ir lengvai interpretuojami. Žemiau pateikiamuose paveikslėliuose (5 pav. - 8 pav.) yra grafiškai pavaizduota kaip keičiasi transformacijų matricos M reikšmės. Pavyzdžiui, jei panagrinėtume 5 pav., pastebėtume, kad didžiausias reikšmes (0.148 ir 0.119) matricos M antrame stulpelyje (vektoriuje M_2) įgyja atitinkamai 18 ir 19 klausimai. Iš 1 priede (**Priedas Nr.** 1) pateikiamo LHFQ klausimyno matome, kad tie klausimai atspindi paciento emocinę būsena, kai tuo tarpu pirmame transformacijų matricos M stulpelyje didžiausias reikšmes įgyja 13, 9, 14, 19, 3, 12, 2, 15, 4, 1, 6, 7, 8 ir 5 klausimai, kurie visi, išskyrus 19 klausimą, atspindi paciento fizinę būseną. Paveikslėliuose (5 pav. - 8 pav.) pavaizduotas matricos M M1 ir M2 stulpeliuose esančių elementų reikšmių išsidėstymas:



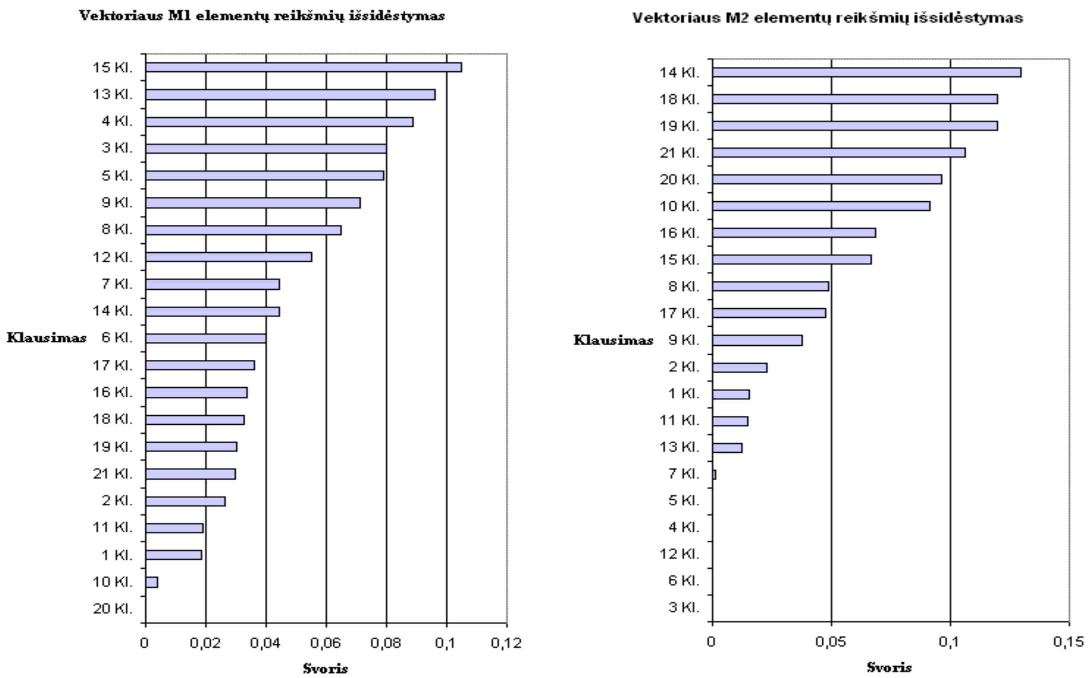
5 pav. Transformacijų matricos M reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais prieš atliekant gydymą. m=21, n=61, r=2.



6 pav. Transformacijų matricos M reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais 12 sav, po atlikto gydymo. m=21, n=46, r=2.



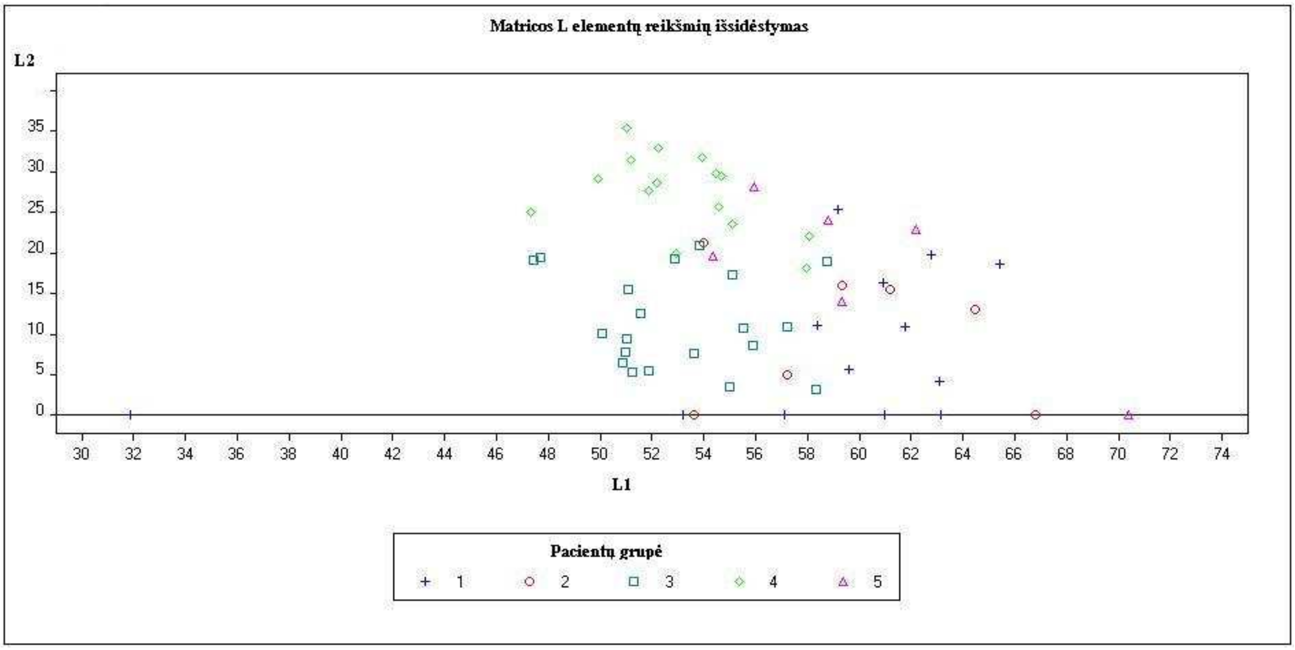
7 pav. Transformacijų matricos M reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais 24 sav. po atlikto gydymo. m=21, n=40, r=2.



8 pav. Tranformacijų matricos M reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais 48 sav. po atlikto gydymo. m=21, n=55, r=2.

Kaip matome, pacientams atlikus tą patį testą praėjus 12 savaičių laikotarpiui po atlikto vienokio ar kitokio gydymo, koeficientai prie klausimų pasikeičia, pavyzdžiui, didesnius koeficientus įgyja klausimai tokie kaip „Ar buvote priverstas gulėti ligoninėje?“ (14 klausimas), „Ar turėjote išlaidų gydymui“ (15 klausimas). Taigi labai logiška, kad šie klausimai pacientams laikotarpiu po operacijos ar kitokio gydymo buvo svarbiausi. Todėl galima manyti, kad matricos M elementus išsidesčiusius 1 ir 2 stulpeliuose, galima interpretuoti kaip atitinkamus koeficientus prie klausimų, ko negalėjome padaryti atlikę PCA, nes kai kurie gauti koeficientai turėjo neigiamą ženklą.

Analogiškai galime nagrinėti matricų L eilutes ir pastebėti kaip atsiskiria atskiros pacientų grupės. Ivykdžius NMF su matricomis, kurias sudaro duomenys, gauti prieš atliekant gydymą, 12, 24 ir 48 savaitėmis po gydymo, gautos skirtingos transformacijų matricos L. Paveikslėliuose (9 pav. - 12 pav.) pavazduota kaip atskirais gydymo laikotarpiais atsiskiria pacientų grupės, matricų L eilutese esančius elementus atidėjus plokštumoje:



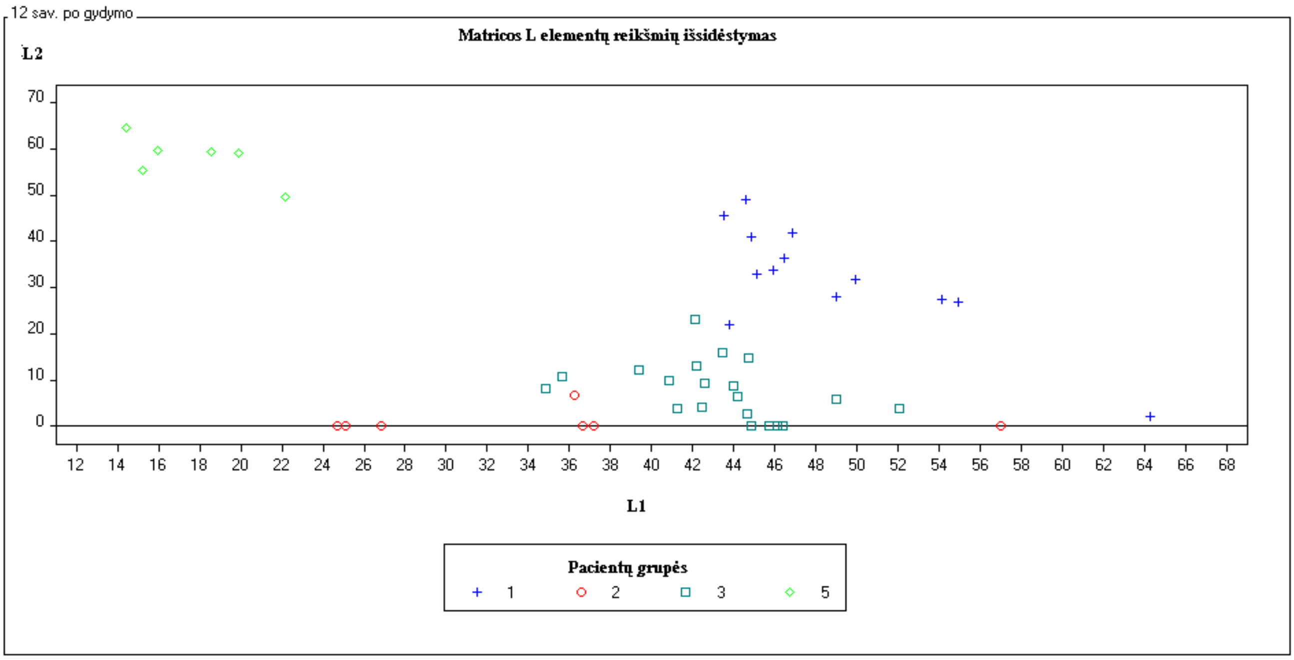
9 pav. Maricos L elementų reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais prieš atliekant tyrimą. m=21, n=61, r=2.

Iš grafiko matome, kad ryškiausiai nuo visų grupių atsiskiria 4 ir 5 pacientų grupės, o 1, 2 ir 3 grupių pacientai susimaišo. Tačiau tie skitumai nėra labia ryškūs. Tačiau taip ir turi būti, nes procedūrai panaudoti pacientų duomeys, prieš atliekant gydymą, kada visi pacientai jaučiasi daugmaž panašiai, o iš to, kad 4 ir 5 grupių pacientams buvo atliekama operacija, galima spręsti, kad jų būklė prieš gydymą buvo blogesnė negu kitu trijų grupių pacientų, todėl gali būti, kad dėl to 4 ir 5 grupės labiau ir atsiskiria. Taip pat pasirinkus 5% reikšmingumo lygmenį ir pagal Mann – Withney testą patikrinus, ar grupės statistiškai reikšmingai skiriasi buvo gauti tokie rezultatai:

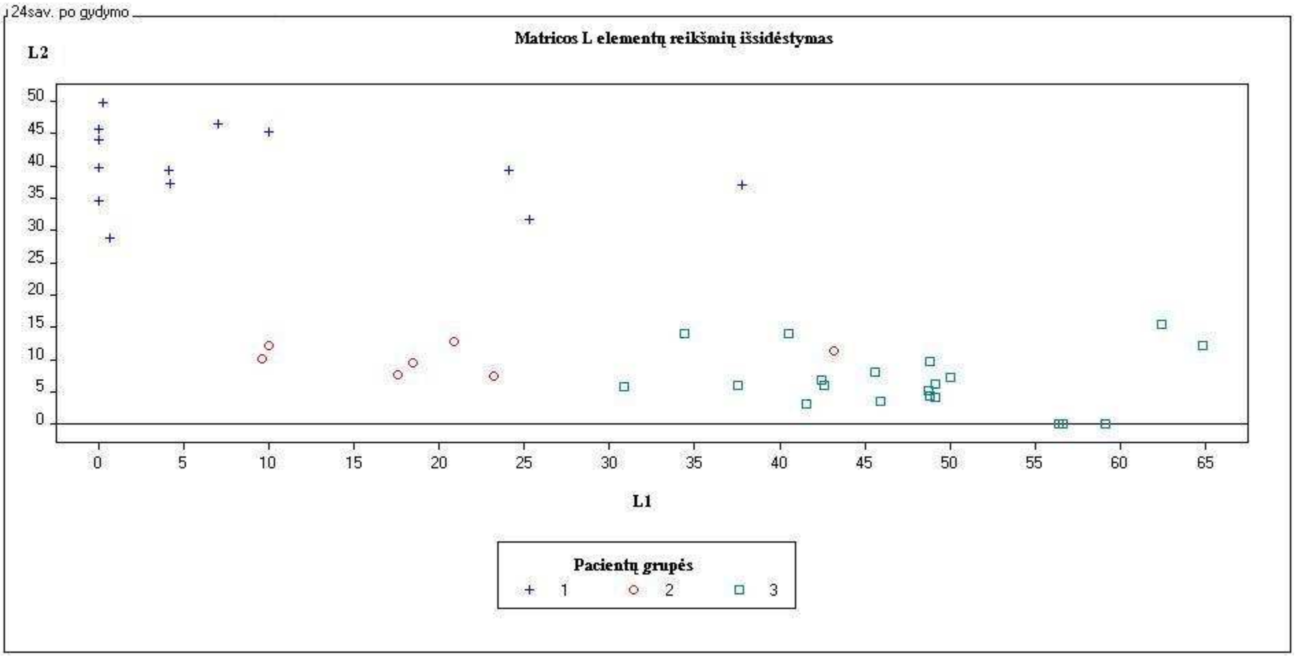
Lentelė. p-reikšmės gautos lyginant pacientų grupes atskirais gydymo laikotarpiais.

	Gydymo laikotarpis (sav.)			
Lyginamos grupės	0	12	24	48
1-2	0.751	0.000	0.006	0.152
1-3	0.428	0.000	0.268	0.000
1-4	0.006	-	-	0.000
1-5	0.008	0.001	-	-
2-3	0.203	0.293	0.007	0.046
2-4	0.003	-	-	0.000
2-5	0.174	0.003	-	-
3-4	0.001	-	-	0.000
3-5	0.001	0.000	-	-
4-5	0.008	-	-	-

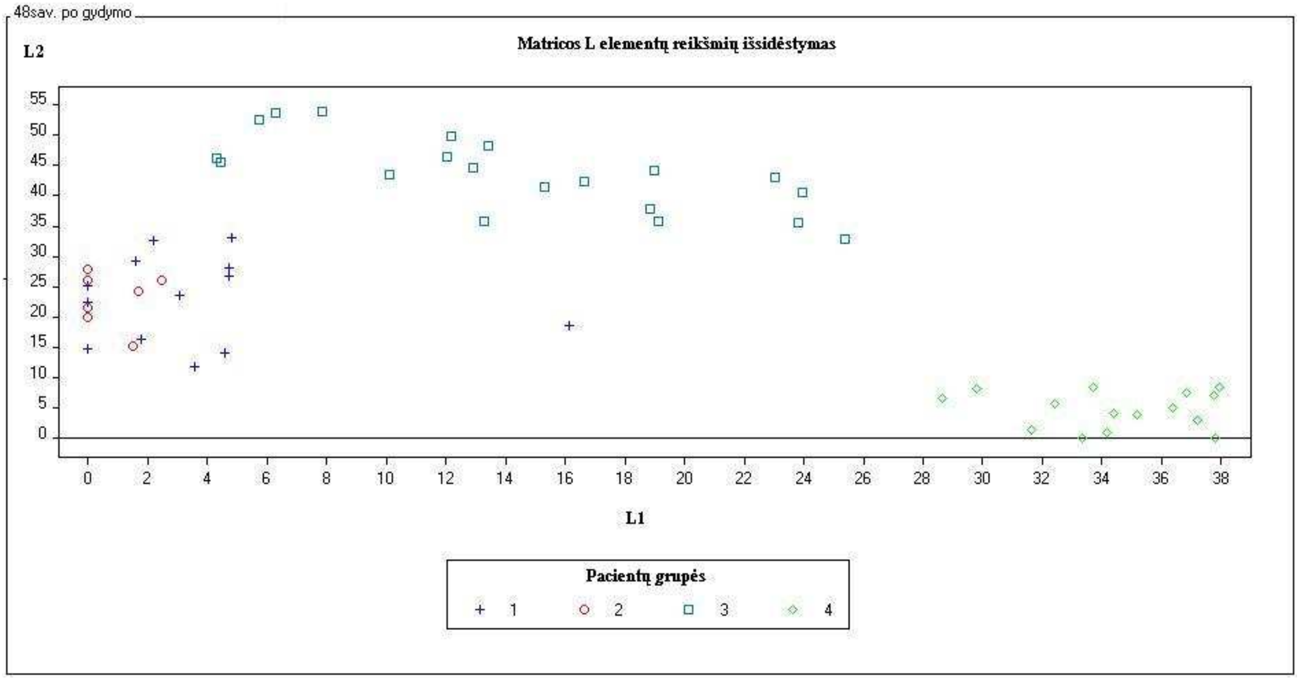
Pastebėkime, kad toliau pateikiamuose paveikslėliuose (9 pav. - 12 pav.) pacientų grupės atsiskiria gana aiškiai, kada jiem yra atlikta gydymo procedūra, kas suteikia priežastį teigti, kad gydymas įtakojo paciento gyvenimo kokybę.



10 pav. Matricos L elementų reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais 12 sav. po atlikto gydymo. m=21, n=46, r=2.



11 pav. Matricos L elementų reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais 24 sav. po atlikto gydymo. m=21, n=40, r=2.



12 pav. Matricos L elementų reikšmių išsidėstymas, atlikus NMF su pacientų duomenimis, gautais 48 sav. po atlikto gydymo. m=21, n=55, r=2.

Taigi grafiškai pavaizdavus matricų L elementų, išsidėsčiusių eilutėsi reikšmes, galima pastebėti, kad jau po gydymo (12 – 48 sav.) atsiskiria atskiros pacientų grupės. Tačiau taip pat reikia pastebėti, kad tam tikrais laikotarpiais tam tikros grupės skiriasi arba labai nežymiai, arba jų elementai susitelkia į vieną grupę. Pavyzdžiui 12 savaitę po gydymo (10 pav.) sunkiai atsiskiria 2 ir 3 grupės, o 48 savaitę (12 pav.) 1 ir 2 grupės visiskai nesiskiria. 1 ir 2 grupių neatsiskyrmui įtaką gali daryti atlikta panaši gydymo procedūrą, tačiau tokiam rezultui būtų naudinga ir medicininė interpretacija. Taip pat matome, kad tiek 12, tiek 48 savaitėmis(10 pav.,12 pav.) labai aiškiai atsiskiria 5 grupė, t.y. ligoniai, kuriems buvo prijungti dirbtiniai širdies skilveliai. Be to svarbu būtų paminėti ir tai, kad kiekvieno taško koordinatės parodo, kaip stipriai kiekvieno paciento gyvenimo kokybė yra įtakojama vienos ar kios klausimų grupių. Pavyzdžiui, jei sakytume, kad matricos $\mathbf{L}$ pirmoji eilutė L_1 parodo kaip stipriai paciento gyvenimo kokybė yra įtakojama jo fizinės būsenos, o L_2 - emocinės, tai iš 10 pav. pastebėtume, kad po operacijos 5 grupės ligoniai savo fizinę būsena vertino daug prasčiau negu emocinę. Kadangi šios grupės pacientams buvo atlikta operacija, kurios metu prijungti dirbtiniai širdies skilveliai, tai labai realu, kad dar kurį laiką po tokio gydymo paciento fizinė būklė negerėja. Taip pat galima pastebėti, kad 12 savatę po gydymo prasčiausiai savo gyvenimo kokybė vertino 2 grupės pacientai. Panagrinėję 12 pav. pastebimė, kad labai aiškiai atsiskiria 4 pacientų grupė, t.y. tie pacientai, kuriems buvo atlikta širdies persodinimo operacija. Toks rezultatas galėtų buti dėl labai skirtingų 1-3 ir 4 grupių gydymo būdų.

3.2.1 Algoritmų, aprašytų 2.8.1 ir 2.8.2 skyreliuose panaudojimas ir gauti rezultatai

Siekiant patikrinti, ar NMF metu gautą transformacijų matricą M galima efektyviai panaudoti kaip klasterizavimo metodą buvo atlikti ekperimentai naudojant 2.8.1 ir 2.8.2 skyreliuose aprašytus algoritmus.

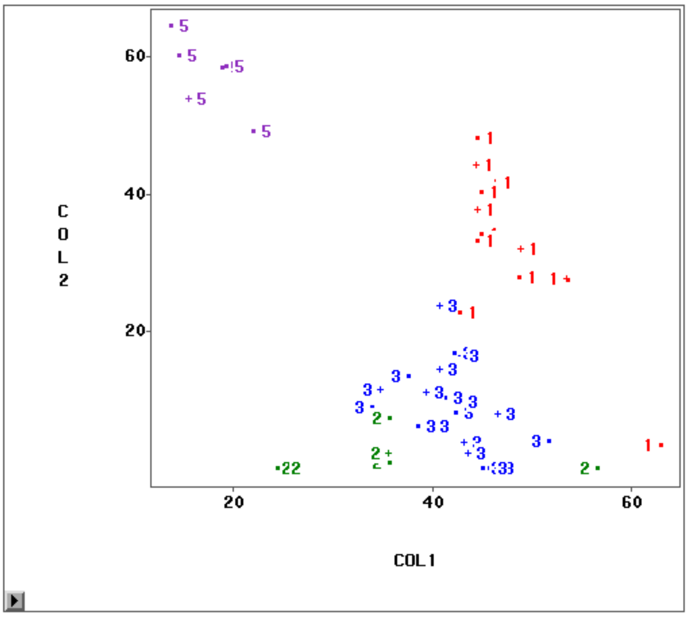
Kiekvienas duomenų rinkinys buvo išskaidytas į dvi dalis: apmokymui naudojamą duomenų rinkinį ir testinių duomenų rinkinį. Apmokymui naudoti duomenys iš kiekvienos grupės buvo atrinkti atsitiktinai. Atrenkamų stebėjimų skaičius buvo pasirinktas pagal taisyklę [24]:

$$k_i = \left\lceil \frac{n_i m_i}{n_i + m_i} \right\rceil ,$$

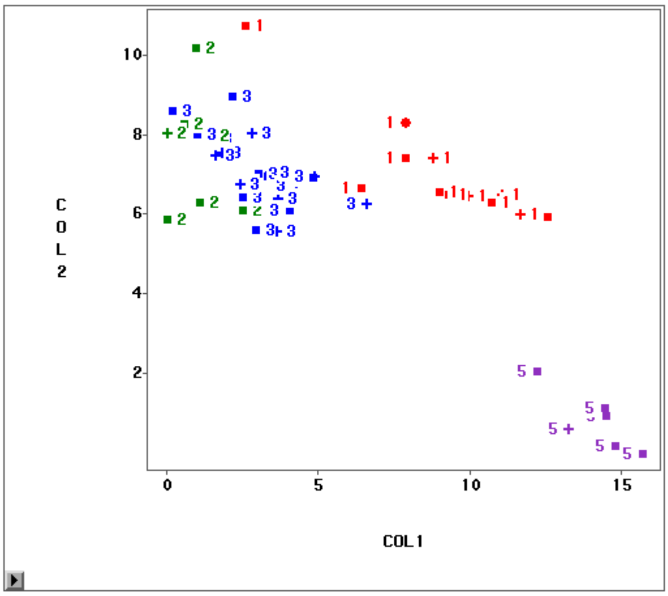
kur n_i yra stebėjimų skaičius i-tojoje grupėje, m_i - kiekvienam stebėjimui iš i-tosios grupės atliktų matavimų skaičius, o k_i - algoritmo apmokymui iš i-tosios grupės atrenkamų stebėjimų skaičius. Iš 1 – 5 grupių buvo atrinkta atitinkamai po 8, 5, 10, 9 ir 5 pacientus. Astsitiktinei atrankai atlikti naudota SAS® procedūra **surveyselect:**

```
proc surveyselect data = Class.duom out = class.tr method = srs
sampsize = (8, 5, 10, 9, 5);
strata f1;
run;
```

Gauti rezultatai:



13 pav. Algoritmo 4 rezultatas. + pažymėti ligoniai iš testinių grupių, . Apmokymų matrica $\mathrm{N}_{tr} \in \mathbb{R}^{21\times28}$, testinių duomenų matrica $\mathrm{N}_{te} \in \mathbb{R}^{21\times18}$, r=2. Pacientų duomenys 12 sav. po gydymo.



14 pav. Algoritmo 5 rezultatas. + pažymėti ligoniai iš testinių grupių, . Apmokymų matrica $\mathrm{N}_{tr} \in \mathbb{R}^{21\times28}$, testinių duomenų matrica $\mathrm{N}_{te} \in \mathbb{R}^{21\times18}$, r=2. Pacientų duomenys 12 sav. po gydymo.

Iš grafikų matome, kad tiek vienas, tiek kitas metodas testinius duomenis grupėms priskiria gana tiksliai, tik 2 ir 3 grupių pacientai šiek tiek susimaišo, tačiau toks pats rezultatas buvo gautas ir atliekant NMF su visais duomenimis (10 pav.). Taip pat galima pastebėti, kad abiem skirtingais atvejais matricos $\mathbf{L}$ eilučių elementai įgyja skirtingas reikšmes. Taip yra dėl to, kad 5 algoritmo atveju, prieš atliekant NMF matricos N elementai yra normalizuojami, kaip parodyta 2.8.2. Priede Nr. 4. yra pateikiamos lentelės su transformacijų matricos M elementų reikšmėmis, kai NMF atliekama nenormalizavus duomenų matricos N elementų ir, kai NMF atliekama, kai duomenų matricos N elementai yra normalizuoti.

Išvados

Šiame darbe buvo išnagrinėti skirtingi dimensijos mažinimo metodai. Panaudotas klasikinis dimensijos mažinimo būdas – pagrindinių komponenčių analizė ir viena naujausių jos alternatyvų – neneigiama matricos faktorizacija.

Eksperimentai buvo atliekami su pacientų, sergančių širdies nepakankamumu atsakymais į LHFQ klausimyno, naudojamo ligonių gyvenimo kokybei nustatyti, klausimus. Nagrinėtos 5 atskiros ligonių grupės 4 skirtingais gydymo laikotarpiais. Parodyta, kad PCA metodai nors ir duoda pakankamai realius rezultatus, tačiau dėl gaunamų neigiamų pagrindinių komponenčių, tie rezultatai nėra lengvai interpretuojami. Todėl kaip PCA alternatyva buvo pasirinktas vienas naujesnių dimensijos mažinimo metodų – neneigiama matricos faktorizacija. Gauti rezultatai parodė ne tik, kaip kiekvienu gydymo laikotarpiu atsiskiria atskiros pacientų grupės, tačiau buvo galima stebėti, kaip gydymo laikotarpyje keičiasi pacientų atsakymai į klausimus, t.y., kad klausimai, parodantys, kurie veiksniai, labiau įtakoja pacientų gyvenimo kokybę vienu gydymo laikotarpiu, kitame gydymo laikotarpyje tampa nebe tokie svarbūs, o išryškėja kiti – labiausiai pacientų gyvenimo kokybę atspindintys klausimai.

Dimensijos mažinimui buvo pritaikyti dviejų tipų NMF algoritmai: paprastas matricų atnaujinimo dauginant algoritmas ir alternuojantis mažiausių kvardratų NMF algoritmas (NMF/ALS). Pastarasis, įvykdymo konvergavimo atžvilgiu pasirodė geresnis, nes transformacijos ir iššifravimo matricų inicializavimui buvo naudojamas ne atsitiktinis generavimas, bet šių matricų išraiškos, minimizuojančios atitinkamą tikslo funkciją. Turimų duomenų rinkinių dimensijos mažinimui tiek NMF, tiek ir NMF/ALS algoritmai pasirodė gana efektyvūs, tačiau žinant plačias NMF taikymo galimybes įvairiose srityse, spartesnis algoritmas, gali būti efektyviau naudojamas su didelio matavimo duomenų rinkiniais, pavyzdžiui, nuotraukomis ar magnetinio rezonanso vaizdais, kada vienu metu yra dirbama su keliais vaizdais, kurių elementai yra pikseliai.

Pasinaudojus NMF algoritmo efektyvumu ir informatyvumu, buvo atlikti ekperimentai iš pradžių apmokant algoritmą ir, tokiu būdu, gaunant tranformacijų matricą, kurią fiksavus ir atlikus iteracijas tik iššifravimo matricai, buvo pastebėta, kad testiniai duomenys gana tiksliai yra priskiriami toms grupėms, iš kurių jie ir buvo paimti. Ieškant efektyvesnio tokio algoritmo pritaikymo, ekperimentas buvo atliktas, taikant kitokį iššifravimo matricos inicializavimo būdą – turimą tikslo funkciją minimizuojančią matricos L išraišką.

Gauti rezultatai rodo, kad dimensijos sumažinimas gali labai palengvinti duomenų interpretaciją ir padėti atskirti duomenų grupes, kurių pilname duomenų rinkinyje yra neįmanoma pastebėti, o tam panaudojamas NMF metodas turi labai plačias efektyvaus taikymo panašiuose uždaviniuose galimybes.

Summary

This work deals with several dimensionality reduction techniques and their implementations in real medical problems. For this reason, firstly, one speaks about classical dimension reduction methods called principal component analysis and singular value decomposition. After these methods are introduced, non – negative matrix factorization (NMF) based on non – negativity constraints in data and basis matrices are presented. Also algorithms for its implementation are introduced. Furthermore, some extended versions of NMF such as Semi - Nonnegative Matrix Factorization, Convex - Nonnegative Matrix factorization and Nonnegative Matrix Tri – Factorization are shortly introduced. Moreover, two ways for implementation of dimensionality reduction via NMF are presented when applied for feature extraction, followed by pattern recognition. These algorithms were executed using real data sets containing patients answers to LHFQ. When experiments were performed , it was noticed that NMF could give quite easy interpretable results. Therefore, it was shown that dimensionality reduction could be effective tool for multidimensional data analysis and classification problems.

Sutrumpinimai ir žymėjimai

ang. -	angliškai
PCA –	pagrindinių komponenčių analizė
SVD –	ypatingųjų reikšmių dekompozicija
NMF -	neneigiama matricos faktorizacija
NMF/ALS –	neneigiama matricos faktorizacija mažiausių kvadratų metodu
N –	duomenų matrica
$X = (X_1; X_2; :::; X_N)$ -	duomenų vektorius
Y_i -	i – toji pagrindinė komponentė
$ø_i$ -	i –toji tikslinė reikšmė
a_i -	i – tasis tikrinis vektorius
M –	transformacijų matrica
L –	iššifravimo matrica
N_{te} -	testinių duomenų matrica
N_{tr} -	apmokymų matrica
I -	vienetinė matrica
§ -	kovariacijų matrica
A_j -	j – tasis matricos A stulpelis
r -	faktorizacijos rangas
LHFQ -	Minesotos Universiteto anketa „Gyvenimas sergant širdies nepakankamumu“
$L(:; :)$ -	tikslo funkcija
$L(:)$ -	Lagranžo funkcija
$k:k_F^2$	Frobenijaus norma
A^T -	transponuota matrica A
$A^{\Box 1}$ -	matricos A atvirkštinė matrica
Z -	sveikų skaičių aibė

Literatūra

[1] V. Bagdonavičius, J. Kruopis. Matematinė statistika, III dalis. Rankraštis.

[2] R. A. Johnson, D. W. Wichern. Applied Multivariate Statistical Analysis. 8 skyrius.

[3] Daniel D. Lee, H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. [tinkle] http://hebb.mit.edu/people/seung/papers/nmfconverge.pdf

[4] Chih – Jen Lin. Projected Gradient Methods for Non-negative Matrix Factorization. *Neural Computation*, 19(2007), 2756-2779.

[5] Chih – Jen Lin. On the Convergence of Multiplicative Update Algorithms for Non-negalive Matrix Factorization. *IEEE Transactions on Neural Networks*, 18(2007), 1589-1596.

[6] M. Á. Carreira-Perpiñán. A Review of Dimension Reduction Techniques. 1997. [tinkle] http://faculty.ucmerced.edu/mcarreira-perpinan/papers/cs-96-09.pdf

[7] P. Cunnigham. Dimension Reduction. Technical Report UCD-CSI-2007-07 [tinkle] http://www.csi.ucd.ie/files/UCD-CSI-2007-7.pdf

[8] I. T. Jolliffe. Principal Component Analysis. Edition 2. Springer, 2002

[9] R. E. Madson, L. K. Hansen ir O. Winther. Singular Value Decomposition and Principal Component Analysis. 2004

[10] K. Baker. Singular Value Decomposition Tutorial, 2005

[11] H. Zou, T. Hastie, R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics.* 2004

[12] T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Edition: 2. Springer, 2008

[13] H. Liu, H. Motoda. Computational Methods of Feature Selection. CRC Press, 2007

[14] O. Okun, H.Priisalu. Nonnegative Matrix Factorization for Pattern Recognition. Proceedings of the Fifth IASTED International Conference VIZUALIZATION, IMAGING, AND IMAGE PROCESSING. September 7 – 9, 2005, Benidorm, Spain

[15] Ch. Ding, T. Li, M.I. Jordan. Convex and Semi – Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,2008

[16] T. Li, Ch. Ding. The Relationship Among Various Nonnegative Matrix Factorization Methods for Clustering

[17] J. Yoo, S. Choi. Probabilistic Matrix Tri – Factorization. Department of Computer Science, POSTECH, Korea, 2008.

[18] M. Mørup, L. H. Clemmenson. Multiplicative Updates for the LASSO. Technical University of Denmark, Informatics and Mathematics Modelling, Denmark.

[19] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of Royal Statistical Society*. Series B (Methodological), Volume 58, Issue 1 (1996), 276 – 288.

[20] M. Chu, F. Diele, R. Plemmons, S. Ragni. Optimality, Computation, and Interpretation of Nonnegative Matrix Factorization. *Unpublished Report*. 2004

[21] P. O. Hoyer Non – negative Matrix Factorization with Sparseness Constaints. *Journal of Machine Learning Research* 5(2004), 1457-1469.

[22] SAS/IML® 9.1 User's Guide. SAS Publishing.

[23] Thomas S., Rector, Ph.D. Overview of The Minnesota Living With Heart Falure® Questionnaire. 2005.

[24] E. Benetos, C. Kotropoulos, Th. Lidy, A.Rauber. Testing Supervided Clasiffiers Based on Non – negative Matrix Factorization to Musical Instrument Classification.

Priedai

Priedas Nr. 1

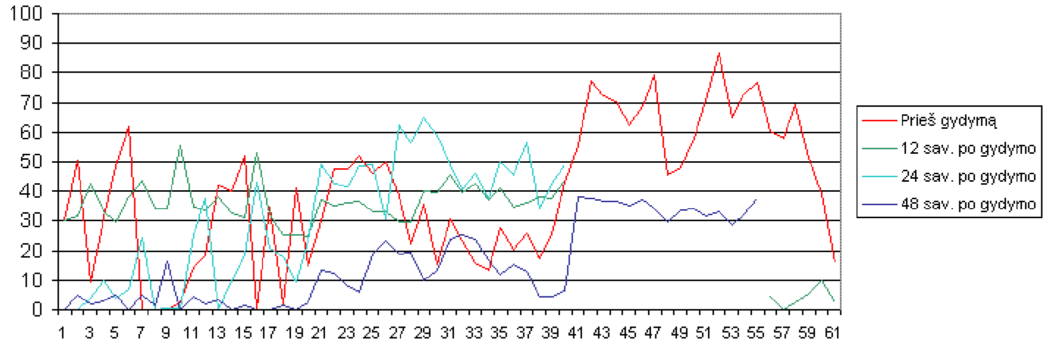
MINESOTOS UNIVERSITETO KLAUSIMYNAS "GYVENIMAS, SERGANT ŠIRDIES NEPAKANKAMUMU"

Ar širdies nepakankamumas jums trukdė įprastai gyventi per praėjusį mėnesį (4 savaites):	Ne	Labai mažai				Labai daug
1. Ar tino kulkšnys ir kojos?	0	1	2	3	4	5
2. Ar privertė jus sėdėti ar gulėti dienos metu?	0	1	2	3	4	5
3. Ar sunku buvo vaikščioti ar lipti laiptais?	0	1	2	3	4	5
4. Ar sunku darbuotis namuose ar kieme?	0	1	2	3	4	5
5. Ar sunku buvo keliauti toliau nuo namų?	0	1	2	3	4	5
6. Ar sunku buvo miegoti naktį?	0	1	2	3	4	5
7. Ar buvo sunku bendrauti ar ką nors veikti su draugais ar namiškiais?	0	1	2	3	4	5
8. Ar sunku buvo atlikti darbą, kuris jums būtinas pragyvenimui?	0	1	2	3	4	5
9. Ar sunku buvo naudotis laisvalaikiu, sportuoti ar užsiimti kitais mėgstamais užsiėmimais?	0	1	2	3	4	5
10. Ar apsunkino jūsų seksualinį aktyvumą?	0	1	2	3	4	5
11. Ar trukdė jums daugiau suvalgyti mėgstamo maisto?	0	1	2	3	4	5
12. Ar sunku buvo kvėpuoti (trūko oro)?	0	1	2	3	4	5
13. Ar jautėtės pavargęs, buvo silpna, trūko energijos?	0	1	2	3	4	5
14. Ar buvote priverstas gulėti ligoninėje?	0	1	2	3	4	5
15. Ar turėjote išlaidų gydymui?	0	1	2	3	4	5
16. Ar kentėjote nuo vaistų šalutinių poveikių?	0	1	2	3	4	5
17. Ar jautėtės našta savo šeimai ir draugams?	0	1	2	3	4	5
18. Ar jautėte, kad prarandate savitvardą gyvenime?	0	1	2	3	4	5
19. Ar vertė jus išgyventi, nerimauti?	0	1	2	3	4	5
20. Ar sunku buvo susikaupti ir atsiminti dalykus?	0	1	2	3	4	5
21. Ar jautėte depresiją?	0	1	2	3	4	5

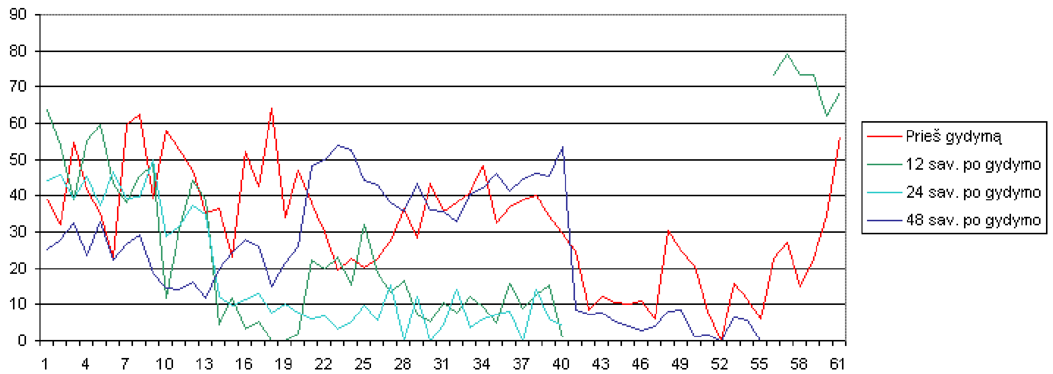
Priedas Nr. 2

Atlikus NMF su ligonių duomenimis gautais prieš gydymą ir kitais gydymo laikotarpiais, gautos iššifravimo matricos **L**, jų pirmos ir antros eilučių (L_1, L_2) elementų reikšmės pavaizduotos grafiškai:

L1



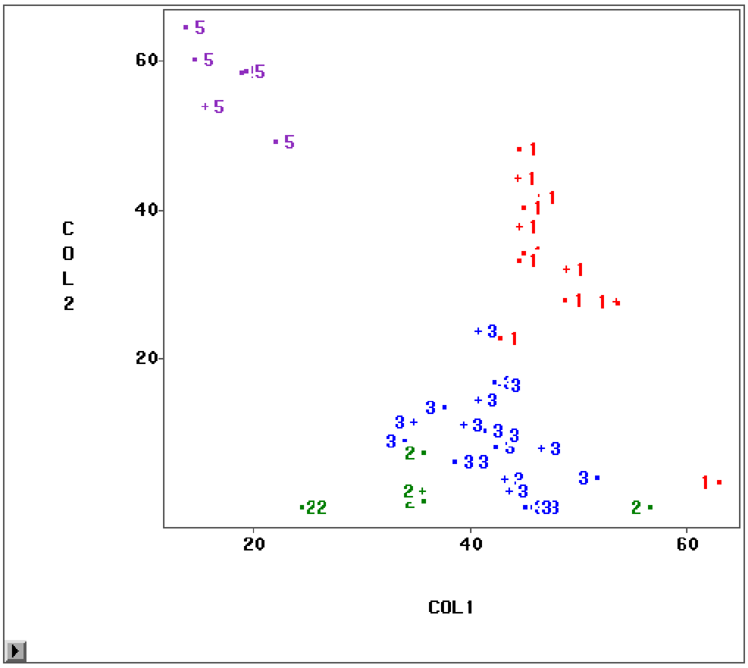
L2



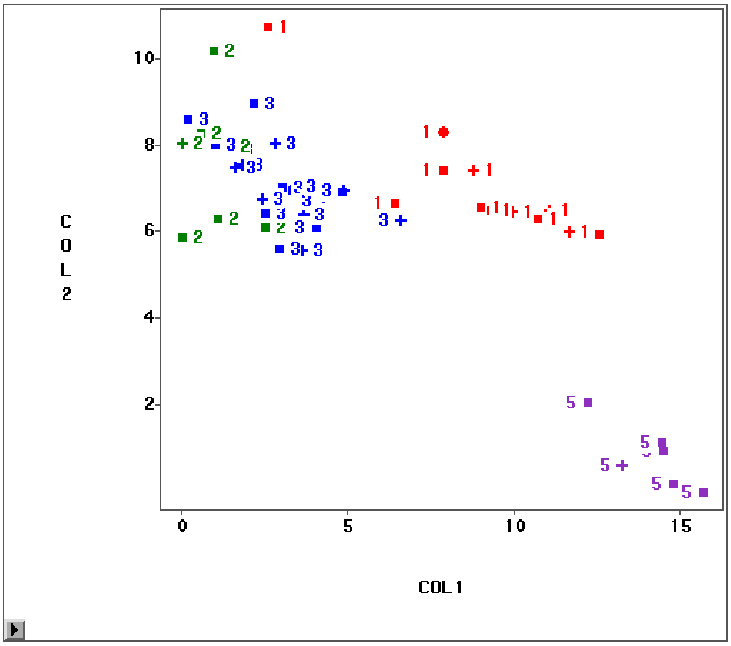
15 pav. Iššifravimo matricos L eilučių elementų reikšmės.

Priedas Nr. 3

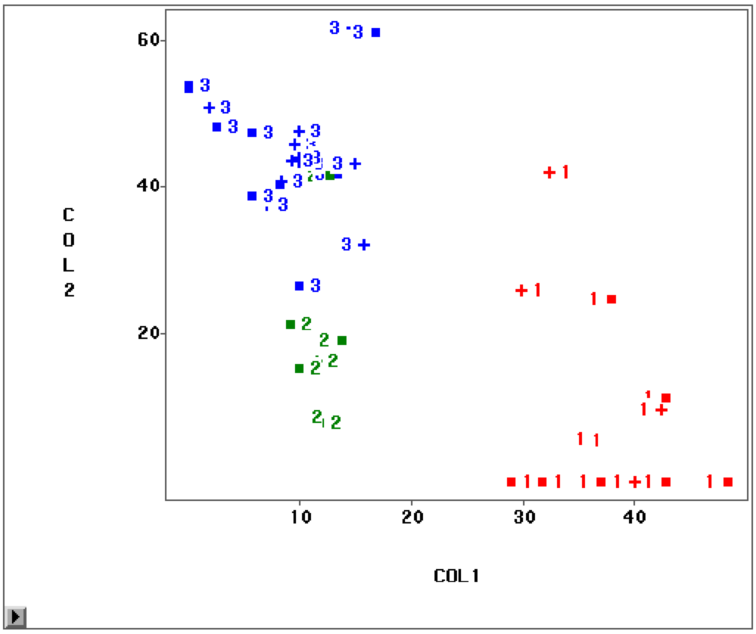
Algoritmų, aprašytų 2.8.1 ir 2.8.2 skyreliuose, gautų rezultatų grafinis pavaizdavimas:



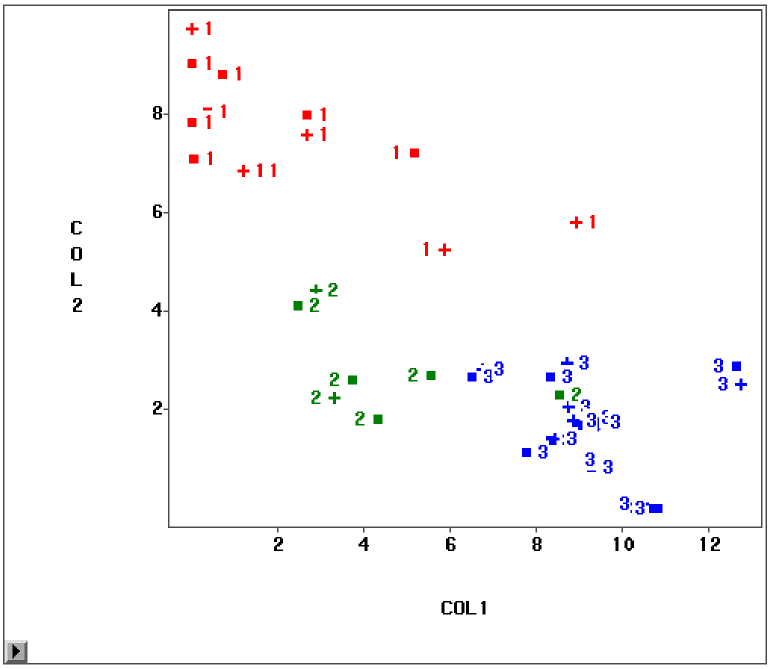
16 pav. Algoritmo 4 (2.8.1 skyrelis) rezultatas. + pažymėti ligoniai iš testinių grupių . Apmokymų matrica $N_{tr} \in \mathbb{R}^{21\times 28}$, testinių duomenų matrica $N_{te} \in \mathbb{R}^{21\times 18}$, r=2. Pacientų duomenys 12 sav. po gydymo.



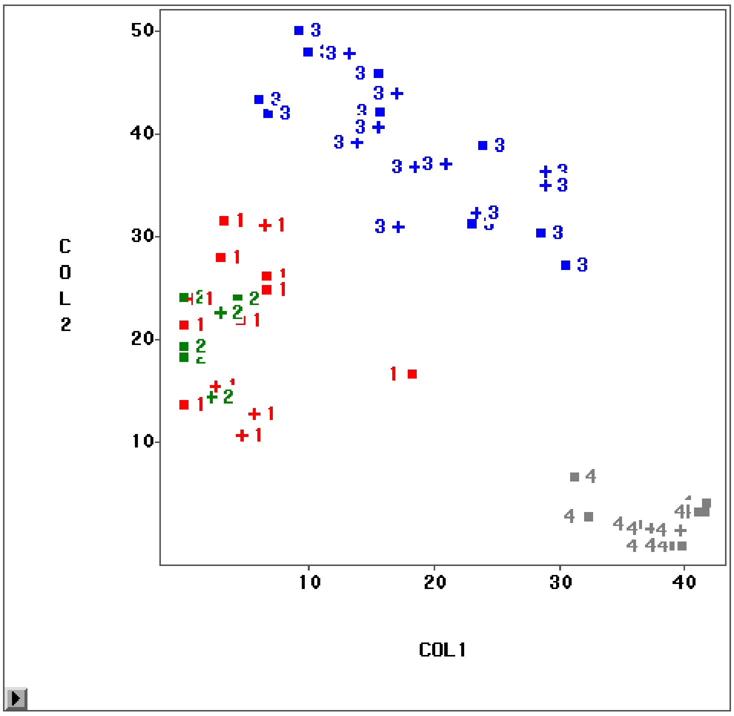
17 pav. Algoritmo 5 (2.8.2 skyrelis) rezultatas. + pažymėti ligoniai iš testinių grupių,. Apmokymų matrica $N_{tr} \in \mathbb{R}^{21\times 28}$, testinių duomenų matrica $N_{te} \in \mathbb{R}^{21\times 18}$, r=2. Pacientų duomenys 12 sav. po gydymo.



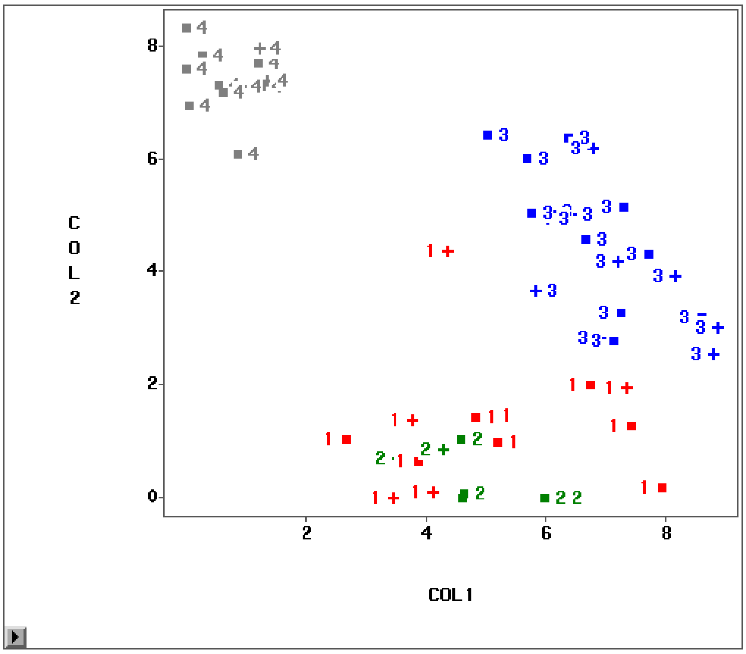
18 pav. Algoritmo 4 (2.8.1 skyrelis) rezultatas. + pažymėti ligoniai iš testinių grupių . Apmokymų matrica $\mathrm{N}_{tr} \in \mathbb{R}^{21\times 23}$, testinių duomenų matrica $\mathrm{N}_{te} \in \mathbb{R}^{21\times 27}$, r=2. Pacientų duomenys 24 sav. po gydymo.



19 pav. Algoritmo 5 (2.8.2 skyrelis) rezultatas. + pažymėti ligoniai iš testinių grupių . Apmokymų matrica $\mathrm{N}_{tr} \in \mathbb{R}^{21\times 23}$, testinių duomenų matrica $\mathrm{N}_{te} \in \mathbb{R}^{21\times 27}$, r=2. Pacientų duomenys 24 sav. po gydymo.



20 pav. Algoritmo 4 (2.8.1 skyrelis) rezultatas. + pažymėti ligoniai iš testinių grupių . Apmokymų matrica $\mathrm{N}_{tr} \in \mathbb{R}^{21\times 32}$, testinių duomenų matrica $\mathrm{N}_{te} \in \mathbb{R}^{21\times 23}$, r=2. Pacientų duomenys 48 sav. po gydymo.



21 pav. Algoritmo 5 (2.8.2 skyrelis) rezultatas. + pažymėti ligoniai iš testinių grupių . Apmokymų matrica $\mathrm{N}_{tr} \in \mathbb{R}^{21\times 23}$, testinių duomenų matrica $\mathrm{N}_{te} \in \mathbb{R}^{21\times 27}$, r=2. Pacientų duomenys 48 sav. po gydymo.

Priedas Nr. 4

1 Lentelė. **Transformacijų matricos M stulepelių M1 ir M2 elementų reikšmės kai NMF atliekama nenormalizuotų duomenų matricai N (a1) ir atlikus matricos N elementų normalizavimą(a2). m=21, n=46, r=2. 12 sav.**

a1	12 savaitė		a2	12 savaitė	
	M1	M2		M1	M2
1 Kl.	0,0072714	0,0523262	1 Kl.	0,045358	0,0221522
2 Kl.	0,0605865	0,0586831	2 Kl.	0,0838217	0,0197966
3 Kl.	0,0394371	0,0809328	3 Kl.	0,0907953	0,020519
4 Kl.	0,0514016	0,0717702	4 Kl.	0,0706784	0,0523171
5 Kl.	0,0604365	0,055422	5 Kl.	0,0530597	0,0536785
6 Kl.	0,0490486	0,0418506	6 Kl.	0,0420078	0,0502847
7 Kl.	0,0376265	0,0331939	7 Kl.	0,0310156	0,0445806
8 Kl.	0,0625136	0,0465486	8 Kl.	0,0442005	0,0642027
9 Kl.	0,0471063	0,0969579	9 Kl.	0,1113521	0,0074969
10 Kl.	0,055039	0,0172885	10 Kl.	0,0146968	0,0663417
11 Kl.	0,0092829	0,0436559	11 Kl.	0,0059472	0,0571316
12 Kl.	0,0148473	0,0756451	12 Kl.	0,0624661	0,0324023
13 Kl.	0,0371176	0,0999117	13 Kl.	0,1062652	0,0177666
14 Kl.	0,0682696	0,0475071	14 Kl.	0,008105	0,1393162
15 Kl.	0,017714	0,1116245	15 Kl.	0,1150952	1,654E-07
16 Kl.	0,0640159	0,0080733	16 Kl.	8,67E-05	0,0946671
17 Kl.	0,0652584	0,0077227	17 Kl.	0,0213637	0,063362
18 Kl.	0,0594876	2,22E-16	18 Kl.	0,0106582	0,0252921
19 Kl.	0,0686826	0,0452437	19 Kl.	0,062441	0,0567575
20 Kl.	0,0693569	2,646E-06	20 Kl.	0,0008535	0,092143
21 Kl.	0,0554002	0,0055396	21 Kl.	0,0196323	0,0196917

2 Lenetelė. **Transformacijų matricos M stulepelių M1 ir M2 elementų reikšmės kai NMF atliekama nenormalizuotų duomenų matricai N (b1) ir atlikus matricos N elementų normalizavimą (b2). m=21, n=40, r=2. 24 sav.**

b1)	24 savaitė		b2)	24 savaitė	
	M1	M2		M1	M2
1 Kl.	0,0384061	0,0305693	1 Kl.	0,0373098	0,0303271
2 Kl.	0,0294631	0,0524382	2 Kl.	0,0328991	0,0513523
3 Kl.	0,0822593	0,0530416	3 Kl.	0,0839565	0,0533348
4 Kl.	0,082284	0,0581261	4 Kl.	0,086283	0,0587907
5 Kl.	1,772E-05	0,0883804	5 Kl.	8,189E-08	0,0876034
6 Kl.	0,0427702	0,0386503	6 Kl.	0,042908	0,0379417
7 Kl.	0,0137711	0,0473472	7 Kl.	0,0122686	0,0462618
8 Kl.	0,0558024	0,0430279	8 Kl.	0,0579091	0,0425527
9 Kl.	0,1057281	0,0762807	9 Kl.	0,1128477	0,0772567
10 Kl.	0,0464993	0,0246335	10 Kl.	0,0436018	0,0256595
11 Kl.	0,0083752	0,0020037	11 Kl.	0,0075867	0,0020098
12 Kl.	0,0572977	0,0302346	12 Kl.	0,0592255	0,0301904
13 Kl.	0,0864009	0,0720462	13 Kl.	0,0925337	0,0712336
14 Kl.	2,22E-16	0,0763254	14 Kl.	2,22E-16	0,0734598
15 Kl.	0,1067606	0,0905136	15 Kl.	0,11303	0,0913847
16 Kl.	0,0321476	0,0090838	16 Kl.	0,0300035	0,0091828
17 Kl.	0,0026753	0,0423623	17 Kl.	3,235E-10	0,0407696

18 Kl.	0,0306677	0,0311995	18 Kl.	0,025629	0,0336297
19 Kl.	0,0551394	0,0855637	19 Kl.	0,0457623	0,0865473
20 Kl.	0,0781059	2,22E-16	20 Kl.	0,0797565	2,22E-16
21 Kl.	0,0453286	0,0480721	21 Kl.	0,0363888	0,0504114

3 *Lenetelė.* Transformacijų matricos M stulepelių M1 ir M2 elementų reikšmės kai NMF atliekama nenormalizuotų duomenų matricai N (c1) ir atlikus matricos N elementų normalizavimą (c2). m=21, n=55, r=2. 48 sav.

c1)	48 savaitė		c2)	48 savaitė	
	M1	M2		M1	M2
1 Kl.	0,0151101	0,0184899	1 Kl.	0,0211583	0,0137915
2 Kl.	0,0202902	0,0278278	2 Kl.	0,0309064	0,0179985
3 Kl.	7,824E-15	0,0853246	3 Kl.	0,0891363	2,22E-16
4 Kl.	0,0037877	0,0919719	4 Kl.	0,0938953	0,0034117
5 Kl.	0,0041433	0,0801644	5 Kl.	0,0774717	0,0062469
6 Kl.	0,0022175	0,0448625	6 Kl.	0,0462586	0,0020413
7 Kl.	0,0032615	0,0472258	7 Kl.	0,0460252	0,0045333
8 Kl.	0,057128	0,0620888	8 Kl.	0,0614063	0,0579092
9 Kl.	0,0342146	0,078669	9 Kl.	0,0838066	0,0315939
10 Kl.	0,0957751	6,538E-12	10 Kl.	1,992E-08	0,0944983
11 Kl.	0,0107161	0,0171259	11 Kl.	0,0167507	0,0110301
12 Kl.	4,324E-07	0,0536668	12 Kl.	0,0543844	6,765E-12
13 Kl.	0,0156353	0,0956428	13 Kl.	0,0996075	0,0136601
14 Kl.	0,131123	0,0435762	14 Kl.	0,035689	0,1347826
15 Kl.	0,0684617	0,1172459	15 Kl.	0,1248303	0,0638857
16 Kl.	0,0635423	0,0296331	16 Kl.	0,0263715	0,0654149
17 Kl.	0,0378248	0,0260761	17 Kl.	0,0236347	0,0394414
18 Kl.	0,1131352	0,0229763	18 Kl.	0,0181078	0,1152009
19 Kl.	0,1256921	0,0286703	19 Kl.	0,0251322	0,1267574
20 Kl.	0,1041249	2,22E-16	20 Kl.	2,22E-16	0,1025908
21 Kl.	0,0937162	0,0286619	21 Kl.	0,0253272	0,0951114