UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
Pedagoška matematika – 2. stopnja

Erik Novak

Nenegativna matrična faktorizacija

Magistrsko delo

Mentor: prof. dr. Bor Plestenjak
Somentor: asist. dr. Andrej Muhič

Ljubljana, 2016

Podpisan Erik Novak izjavljam:

- da sem magistrsko delo z naslovom *Nenegativna matrična faktorizacija* izdelal samostojno pod mentorstvom prof. dr. Bora Plestenjaka in somentorstvom asist. dr. Andreja Muhiča in

- da Fakulteti za matematiko in fiziko Univerze v Ljubljani dovoljujem objavo elektronske oblike svojega dela na spletnih straneh.

Ljubljana, 2016 Podpis: ....

Zahvala

Zahvaljujem se prof. dr. Boru Plestenjaku in asist. dr. Andreju Muhiču za nasvete in popravke pri pisanju magistrske naloge.

Hvala Janu Rupniku, Blažu Fortuni in Luki Stopar z Inštituta Jožefa Stefana za pomoč, razlago in svetovanje pri pisanju programske kode.

Hvala tudi vsem, ki so mi v času študija stali ob strani, še posebej očetu Andreju, materi Andreji in sestri Mateji.

Kazalo

Seznam algoritmov

Slike

Tabele

Program magistrskega dela

V magistrskem delu obravnavajte nenegativno matrično faktorizacijo. V številnih primerih, v katerih nastopajo nenegativne matrike, je nenegativna faktorizacija primernejša od drugih standardnih razcepov, kot sta npr. singularni razcep ali QR razcep, saj jo je lažje interpretirati. Pri nenegativni faktorizaciji nenegativno matriko zapišemo kot produkt dveh nenegativnih matrik oziroma jo aproksimiramo s produktom nenegativnih matrik manjšega ranga.

Za razliko od standardnih razcepov pri nenegativni faktorizaciji rešujemo optimizacijski problem. Predstavite numerične metode za nenegativno faktorizacijo, ki temeljijo na metodi gradientnega spusta. Metode implementirajte in jih preizkusite na testnih primerih iz področja tekstovnega rudarjenja.

Kot osnovno literaturo uporabite:

- Ngoc-Diep Ho, *Nonnegative Matrix Factorization. Algorithms and Applications*, doktorsko delo, Université Catholique de Louvain, Louvain, 2008.
- Joel E. Cohen, Uriel G. Rothblum, Nonnegative ranks, decompositions, and factorizations of nonnegative matrices, Linear Algebra Appl. 190 (1993) 149-168.

prof. dr. Bor Plestenjak

asist. dr. Andrej Muhič

Povzetek

Nenegativna matrična faktorizacija je optimizacijski problem, kjer nenegativno matriko aproksimiramo s produktom dveh nenegativnih matrik manjšega ranga. Zaradi nekonveksnosti energijske funkcije je težko najti optimalno rešitev. Zato se omejimo na iskanje stacionarnih točk. Te najdemo z uporabo optimizacijskih algoritmov, ki temeljijo na metodi gradientnega spusta. Izbrati moramo ustrezno inicializacijsko metodo, saj ta zelo vpliva na konvergenco algoritma. Prav tako moramo paziti na izbiro zaustavitvenega pogoja. Na koncu predstavimo dva primera uporabe nenegativne matrične faktorizacije. Prvi primer je iz področja tekstovnega rudarjenja, drugi pa iz priporočilnih sistemov.

Abstract

Non-negative matrix factorization is an optimization problem, where we approximate a non-negative matrix with a product of two non-negative matrices with a lesser rank. Due to the non-convexity of the energy function, it is hard to find the optimal solution. That is why we limit our search to the stationary points. We find them using optimization algorithms that are based on the gradient descent method. By choosing an appropriate initialization method, we can influence the convergence of the algorithm. We must also carefully choose the stopping condition. At the end of the paper we present two applications of the non-negative matrix factorization. The first one is from the field text mining and the second is an example of the recommender system.

Math. Subj. Class. (2010): 15A23 [Factorization of matrices], 15B48 [Positive matrices and their generalizations], 90C30 [Nonlinear programming]

Ključne besede: Nenegativna matrična faktorizacija, nenegativna matrika, optimizacijski problem, metoda gradientnega spusta

Keywords: Non-negative matrix factorization, non-negative matrix, optimization problem, gradient descent method

Uvod

V modernem času se vsak dan proizvede ogromno podatkov. Ti podatki so lahko produkt znanstvenih raziskav, eksperimentov, novinarstva, pisanja blogov, senzorskih merjenj itd. Zaradi njihove količine podatki sami po sebi niso zelo informativni. Zato jih moramo procesirati. S tem odkrijemo kakovostne informacije, ki nam pomagajo pri naši raziskavi in pri iskanju resnice.

Procesiranje podatkov je zahtevno delo. Za pridobivanje kakovostnih informacij moramo uporabiti ustrezen matematični model. Ker v večini primerov ne poznamo narave podatkov, je izbira pravega modela težko opravilo. Moramo se tudi zavedati, da niso vsi podatki v numerični obliki. Veliko jih je je v tekstovni ali slikovni obliki. Takrat moramo tudi izbrati način predstavitve tovrstnih podatkov v numerični obliki. Velikokrat se odločimo za predstavitev, kjer so predstavniki nenegativna števila. Zato želimo imeti matematični model, ki ohranja nenegativnost podatkov in vrne rezultat, ki je enostavno razumljiv. Tak model je *nenegativna matrična faktorizacija*.

V magistrski nalogi bomo spoznali problem nenegativne matrične faktorizacije, kjer želimo nenegativno matriko aproksimirati s produktom dveh nenegativnih matrik manjšega ranga. Soroden je problemu aproksimacije matrike z matriko manjšega ranga, ki ga lahko rešimo s singularnim razcepom. Na začetku bomo problem formulirali in ga poskusili rešiti. Nato si bomo ogledali nekaj popularnih algoritmov za reševanje problema. Te bomo med seboj tudi primerjali. Na koncu bomo spoznali dva načina uporabe nenegativne matrične faktorizacije. Prvi bo s področja tekstovnega rudarjenja, drugi pa iz priporočilnih sistemov.

Poglavje 1

Predhodno znanje

Preden se lotimo problema nenegativne matrične faktorizacije si bomo ogledali nekaj stvari. Najprej bomo ponovili in dodali nekaj pojmov iz linearne algebre. Ponovili bomo kaj je singularni razcep in kje se ga uporabi. Na koncu se bomo spoznali z optimizacijskimi problemi, kjer bomo tudi definirali nekaj osnovnih pojmov in dokazali nekaj trditev. To bo osnovno orodje pri reševanju našega problema.

1.1 Linearna algebra

V tem razdelku bomo ponovili in dodali nekaj pojmov iz linearne algebre.

Definicija 1.1.1 (Matrična vektorizacija)**.** Naj bo $A \in \mathbb{R}^{m \times n}$ matrika z m vrsticami in n stolpci. *Matrična vektorizacija* je predstavitev matrike v vektorski obliki, tj.

$$vec(A) = \begin{bmatrix} A_{:1} \\ A_{:2} \\ \vdots \\ A_{:n} \end{bmatrix} \in \mathbb{R}^{nm},$$

kjer smo z $A_{:i}$ označili i-ti stolpec matrike A.

Definicija 1.1.2 (Skalarni produkt matrik)**.** Naj bosta $A, B \in \mathbb{R}^{m \times n}$ matriki. *Skalarni produkt dveh matrik* je definiran kot

$$\langle A, B \rangle = vec(A)^T vec(B) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij} = \text{sled}(A^T B),$$

kjer je $\text{sled}(A)$ vsota vseh diagonalnih elementov matrike A.

S pomočjo skalarnega produkta definiramo *Frobeniusovo normo*

$$\|A\| = \sqrt{\langle A, A \rangle} = \|A\|_F$$

in metrični prostor, definiran z množico vseh matrik in Frobeniusovo razdaljo

$$d(A,B) = \|A - B\|_F. \tag{1.1}$$

Definicija 1.1.3 (Permutacijska matrika)**.** Naj bo $S \in \mathbb{R}^{n\times n}$ kvadratna matrika. Matrika S pravimo *permutacijska matrika*, če je v vsaki vrstici in vsakem stolpcu natanko ena vrednost enaka 1, ostale pa enake 0.

Posplošitev permutacijske matrike je monomialna matrika.

Definicija 1.1.4 (Monomialna matrika)**.** Naj bo $M \in \mathbb{R}^{n\times n}$ kvadratna matrika. Matriki M pravimo *monomialna matrika*, če je v vsaki vrstici in vsakem stolpcu natanko ena neničelna vrednost.

Definicija 1.1.5 (Hadamardov produkt)**.** Naj bodo $A, B, C \in \mathbb{R}^{m\times n}$ matrike. *Hadamardov ali Schurov produkt* dveh enako velikih matrik označimo s $C = A \circ B$, kjer je $C_{ij} = A_{ij}B_{ij}$.

Hadamardov produkt ima naslednje lastnosti.

Lema 1.1.6. *Naj bodo $A, B, C, D \in \mathbb{R}^{m\times n}$ matrike. Potem velja*

1. $A \circ B = B \circ A$,
2. $A^T \circ B^T = (A \circ B)^T$,
3. $(A \circ B)(C \circ D)^T = (AC^T) \circ (BD^T) = (AD^T) \circ (BC^T)$.

Dokaz. Naj bodo $A, B, C, D \in \mathbb{R}^{m\times n}$ matrike.

1. Preverimo, če so elementi matrike $(A \circ B)$ enaki istoležnim elementom matrike $(B \circ A)$.
$$(A \circ B)_{ij} = A_{ij}B_{ij} = B_{ij}A_{ij} = (B \circ A)_{ij}.$$

2. Preverimo, če so elementi matrike $(A^T \circ B^T)$ enaki istoležnim elementom matrike $(A \circ B)^T$.
$$(A^T \circ B^T)_{ij} = A^T_{ij}B^T_{ij} = A_{ji}B_{ji} = (A \circ B)_{ji} = (A \circ B)^T_{ij}.$$

3. Preverimo, če veljajo enakosti iz točke 3.

$$
\begin{aligned}
((A \circ B)(C \circ D)^T)_{ij} &= \sum_{k=1}^{n} (A_{ik} B_{ik})(C^T_{kj} D^T_{kj}) \\
&= \sum_{k=1}^{n} (A_{ik} C^T_{kj})(B_{ik} D^T_{kj}) \; (= ((AC^T) \circ (BD^T))_{ij}) \\
&= \sum_{k=1}^{n} (A_{ik} D^T_{kj})(B_{ik} C^T_{kj}) \; (= ((AD^T) \circ (BC^T))_{ij}).
\end{aligned}
$$

□

S Hadamardovim produktom lahko definiramo naslednje operacije.

Definicija 1.1.7 (Hadamardova potenca)**.** Naj bo $A \in \mathbb{R}^{m \times n}$ matrika in $r \in \mathbb{R}$. *Hadamardova potenca* je definirana kot

$$
(A^{\circ r})_{ij} = A^r_{ij}.
$$

Definicija 1.1.8 (Hadamardovo deljenje)**.** Naj bodo $A, B, C \in \mathbb{R}^{m \times n}$ matrike. *Hadamardovo deljenje* je definirano kot

$$
C = \frac{[A]}{[B]} = A \circ B^{\circ(-1)}.
$$

1.1.1 Nenegativne matrike

Definicija 1.1.9 (Nenegativna matrika)**.** Matrika A je *nenegativna*, če so vse njene vrednosti nenegativne. Nenegativnost matrike označimo z $A \geq 0$. Prostor vseh nenegativnih matrik z m vrsticami in n stolpci označimo z $\mathbb{R}^{m \times n}_+$. Prostor vseh nenegativnih vektorjev dimenzije n označimo z $\mathbb{R}^n_+$.

Preden nadaljujemo, se spomnimo kaj sta spekter in spektralni radij matrike.

Definicija 1.1.10 (Spekter in spektralni radij)**.** Naj bo $A \in \mathbb{R}^{n \times n}$ matrika. *Spekter* matrike A je množica vseh lastnih vrednosti matrike A, tj.

$$
\sigma(A) = \{\lambda \in \mathbb{C} \colon Ax = \lambda x,\ x \in \mathbb{C}^n \setminus \{0\}\}.
$$

Spektralni radij matrike A je največja absolutna vrednost lastnih vrednosti matrike A, tj.

$$
\rho(A) = \max\{|\lambda| \colon \lambda \in \sigma(A)\}.
$$

Definicija 1.1.11 (Razcepna matrika)**.** Naj bo $A \in \mathbb{R}^{n\times n}$ matrika. Matrika A je *razcepna*, če obstaja taka nesingularna matrika S, da velja

$$SAS^{-1} = \begin{bmatrix} E & F \\ 0 & G \end{bmatrix},$$

kjer sta E in G kvadratni matriki. Matrika je *nerazcepna*, če ni razcepna.

Eno izmed najpomembnejših lastnosti nenegativne matrike opisuje *Perron-Frobeniusov izrek*. Dokaz izreka najdemo v [2].

Izrek 1.1.12 (Perron-Frobeniusov izrek)**.** *Naj bo $A \in \mathbb{R}_+^{n\times n}$ nerazcepna nenegativna kvadratna matrika in $\rho(A)$ spektralni radij matrike A. Potem veljajo naslednje trditve:*

1. *$\rho(A)$ je algebraično enostavna lastna vrednost matrike A in njen normiran pozitiven lastni vektor v je enolično določen. Če matrika A ni ničelna, potem je $\rho(A) > 0$.*

2. *Vsak nenegativen lastni vektor matrike A je večkratnik vektorja v.*

3. *Če ima matrika A natanko q lastnih vrednosti λ, za katere velja $|\lambda| = \rho(A)$, potem so te lastne vrednosti podane z enačbo $\rho(A)e^{\frac{2\pi ik}{q}}$ za $k = 0, 1, \ldots, q-1$.*

Opomba 1.1.13. Lastni vrednosti $\rho(A)$ iz zgornjega izreka pravimo *Perronova lastna vrednost*, pripadajoč normiran pozitiven lastni vektor v je *Perronov lastni vektor*.

Naj bo $V \subset \mathbb{R}^{m\times n}$ zaprta množica matrik in $A \in \mathbb{R}^{m\times n}$ matrika. Najbližjemu elementu matrike A v množici V pravimo *projekcija matrike A na V*. Projekcijo označimo s $P_V(A)$. Ko je V množica vseh nenegativnih matrik in bližino matrik merimo s Frobeniusovo razdaljo (1.1), potem projekcijo matrike A označimo s $P_{\mathbb{R}_+^{m\times n}}(A) = [A]_+$. Projekcija je definirana z

$$\left([A]_+\right)_{ij} = \begin{cases} A_{ij}, & \text{če je } A_{ij} > 0, \\ 0, & \text{če je } A_{ij} \leq 0, \end{cases} = \max(0, A_{ij}).$$

1.2 Singularni razcep

Izrek 1.2.1 (Singularni razcep). *Za vsako matriko $A \in \mathbb{R}^{m\times n}$ obstajata ortogonalni matriki $U \in \mathbb{R}^{m\times m}$ in $V \in \mathbb{R}^{n\times n}$ ter matrika $\Sigma \in \mathbb{R}^{m\times n}$, za katere velja*

$$
\begin{aligned}
A &= U\Sigma V^T, \qquad (1.2)\\
\Sigma &= \left[\begin{array}{ccc|c}
\sigma_1 & & 0 & \\
 & \ddots & & O_{r\times(n-r)} \\
0 & & \sigma_r & \\
\hline
 & O_{(m-r)\times r} & & O_{(m-r)\times(n-r)}
\end{array}\right],
\end{aligned}
$$

kjer so singularne vrednosti σ_i pozitivne in v nenaraščajočem vrstnem redu,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$

Dokaz izreka se nahaja v [21, izrek 5.3]. Razcepu (1.2) pravimo *singularni razcep*. Iz singularnega razcepa sledi $\operatorname{rang}(A) = r$ in

$$A = \sum_{i=1}^{r} \sigma_i U_{:i} V_{i:},$$

kjer z $U_{:i}$ označimo i-ti stolpec matrike U in z $V_{i:}$ označimo i-to vrstico matrike V.

Z razcepom lahko rešimo naslednji problem.

Problem 1.2.2 (Aproksimacija matrike z matriko manjšega ranga). *Naj bo $A \in \mathbb{R}^{m\times n}$ matrika in $r \leq \min\{m,n\}$. Reši problem*

$$\min_{B\in\mathbb{R}^{m\times n},\, \operatorname{rang}(B)\leq r} \frac{1}{2}\|A-B\|_F^2. \qquad (1.3)$$

Naslednji izrek nam vrne rešitev zgornjega problema.

Izrek 1.2.3 (Eckart-Young-Mirsky). *Naj bo $A \in \mathbb{R}^{m\times n}$ matrika, kjer je $m \geq n$. Naj ima A singularni razcep*

$$A = U\Sigma V^T,\ \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix},$$

kjer so $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ *singularne vrednosti, matriki* $U \in \mathbb{R}^{m\times m}$ *in* $V \in \mathbb{R}^{n\times n}$ *pa ortogonalni. Naj bo* $r \in \mathbb{N}$*,* $1 \leq r \leq n$*. Potem je matrika*

$$A_r = U\Sigma_r V^T,\ \Sigma_r = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_r & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix},$$

globalni minimum problema 1.2.2. Vrednost (1.3) je tedaj enaka

$$\frac{1}{2}\|A - A_r\|_F^2 = \frac{1}{2}\sum_{i=r+1}^{n} \sigma_i^2.$$

Če velja še $\sigma_r > \sigma_{r+1}$*, potem je* A_r *enolično določen globalni minimum.*

Dokaz izreka je v [16]. Oglejmo si problem, ki je podoben problemu nenegativne matrične faktorizacije.

Problem 1.2.4 (Matrična faktorizacija)**.** *Naj bo* $A \in \mathbb{R}^{m\times n}$ *matrika. Poišči rešitev problema*

$$\min_{X\in\mathbb{R}^{m\times r},\, Y\in\mathbb{R}^{n\times r}} \frac{1}{2}\|A - XY^T\|_F^2.$$

Zaradi dimenzij matrik X in Y velja $\operatorname{rang}(XY^T) \leq r$. Velja tudi, da vsako matriko ranga kvečjemu r lahko zapišemo kot produkt XY^T, kjer sta $X \in \mathbb{R}^{m\times r}$ in $Y \in \mathbb{R}^{n\times r}$. Torej sta si problema 1.2.2 in 1.2.4 ekvivalentna.

Produkt $A_r = XY^T$ ni enoličen, saj vsak par $(XR, Y(R^{-1})^T)$, kjer je $R \in \mathbb{R}^{r\times r}$ kvadratna obrnljiva matrika, vrne enak produkt XY^T. Da se temu izognemo, izberemo matriki

$$X = U\Sigma^{\frac{1}{2}},\ Y = V\Sigma^{\frac{1}{2}},$$

kjer sta matriki U in V ortogonalni, matrika Σ pa je diagonalna in nenegativna. To je ekvivalentno računanju singularnega razcepa za matriko $A_r = XY^T = U\Sigma V^T$. Navedimo naslednji izrek, ki ga najdemo v [10].

Izrek 1.2.5. *Naj bo* $A \in \mathbb{R}^{m\times n}$ *matrika, kjer je* $m > n$ *in* $rang(A) = t$*. Če je par matrik* (X, Y) *stacionarna točka problema 1.2.4, potem sta* $X = U\widehat{\Sigma}_r^{\frac{1}{2}}$ *in* $Y = V\widehat{\Sigma}_r^{\frac{1}{2}}$*, kjer sta* $U \in \mathbb{R}^{m\times m}$ *in* $V \in \mathbb{R}^{n\times n}$ *ortogonalni matriki, za kateri velja*

$$A = U\widehat{\Sigma}V^T,\ A_r = U\widehat{\Sigma}_r V^T,$$

kjer sta

$$\widehat{\Sigma} = \begin{bmatrix} \widehat{\sigma}_1 & 0 & \cdots & 0 \\ 0 & \widehat{\sigma}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widehat{\sigma}_n \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \widehat{\Sigma}_r = \begin{bmatrix} \widehat{\sigma}_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \widehat{\sigma}_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & \widehat{\sigma}_r & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

in $\widehat{\sigma}_i$ so singularne vrednosti matrike A, ki niso nujno urejene po velikosti. Vrednost enačbe 1.2.4 je takrat enaka

$$\frac{1}{2}\|A - XY^T\|_F^2 = \frac{1}{2}\|A - A_r\|_F^2 = \frac{1}{2}\sum_{i=r+1}^{t} \widehat{\sigma}_i^2.$$

Če so singularne vrednosti paroma različne, potem je $\frac{n!}{r!(n-r)!}$ možnih stacionarnih točk A_r. Naslednji izrek nam pa pove, katere točke so lokalni minimumi.

Izrek 1.2.6. *Minimumi problema 1.2.4 so podani z izrekom 1.2.3 in so globalni minimumi. Ostale stacionarne točke so sedla.*

Dokaz. Predpostavimo, da je A_r stacionarna točka iz izreka 1.2.5 in ne iz izreka 1.2.3. Potem obstaja permutacija stolpcev matrik U in V ter vseh diagonalnih elementov matrik $\widehat{\Sigma}$ in $\widehat{\Sigma}_r$, da velja $\widehat{\sigma}_{r+1} > \widehat{\sigma}_r$. Tedaj lahko konstruiramo dve točki znotraj ϵ-okolice matrike A_r, kjer je vrednost funkcije $f(B) = \frac{1}{2}\|A - B\|_F^2$ v eni točki večja, v drugi pa je manjša. Prva točka je

$$\overline{\Sigma}_r(\epsilon) = \begin{bmatrix} \widehat{\sigma}_1 + \epsilon & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \widehat{\sigma}_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \cdots & \widehat{\sigma}_r & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}, \quad \overline{A}_r(\epsilon) = U\overline{\Sigma}_r(\epsilon)V^T,$$

druga pa

$$\underline{\Sigma}_r(\epsilon) = \begin{bmatrix} \widehat{\sigma}_1 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & \widehat{\sigma}_r & \epsilon\sqrt{\widehat{\sigma}_r} & \vdots & 0 \\ 0 & \cdots & \epsilon\sqrt{\widehat{\sigma}_r} & \epsilon^2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}, \quad \underline{A}_r(\epsilon) = U\underline{\Sigma}_r(\epsilon)V^T.$$

Matriki $\overline{A}_r(\epsilon)$ in $\underline{A}_r(\epsilon)$ sta ranga r. Vrednost funkcije f v točki $\underline{A}_r(\epsilon)$ je

$$\begin{aligned} \|A - \underline{A}_r(\epsilon)\|_F^2 &= \|U(\widehat{\Sigma} - \underline{\Sigma}_r(\epsilon))V^T\|_F^2 \\ &= \|\widehat{\Sigma} - \underline{\Sigma}_r(\epsilon)\|_F^2 \\ &= 2\widehat{\sigma}_r\epsilon^2 + (\widehat{\sigma}_{r+1} - \epsilon^2)^2 + \sum_{i=r+2}^{t} \widehat{\sigma}_i^2 \\ &= 2\widehat{\sigma}_r\epsilon^2 + \widehat{\sigma}_{r+1}^2 - 2\widehat{\sigma}_{r+1}\epsilon^2 + \epsilon^4 + \sum_{i=r+2}^{t} \widehat{\sigma}_i^2 \\ &= \epsilon^2(\epsilon^2 - 2(\widehat{\sigma}_{r+1} - \widehat{\sigma}_r)) + \sum_{i=r+1}^{t} \widehat{\sigma}_i^2 \\ &< \sum_{i=r+1}^{t} \widehat{\sigma}_i^2 = \|A - A_r\|_F^2 \end{aligned}$$

za vsak $\epsilon \in \left(0, \sqrt{2(\widehat{\sigma}_{r+1} - \widehat{\sigma}_r)}\right)$. Vrednost funkcije v točki $\overline{A}_r(\epsilon)$ je

$$\begin{aligned} \|A - \overline{A}_r(\epsilon)\|_F^2 &= \|U(\widehat{\Sigma} - \overline{\Sigma}_r(\epsilon))V^T\|_F^2 \\ &= \|\widehat{\Sigma} - \overline{\Sigma}_r(\epsilon)\|_F^2 \\ &= \epsilon^2 + \sum_{i=r+1}^{t} \widehat{\sigma}_i^2 \\ &> \sum_{i=r+1}^{t} \widehat{\sigma}_i^2 = \|A - A_r\|_F^2 \end{aligned}$$

za vsak $\epsilon > 0$. Pri izračunu smo upoštevali ortogonalnost matrik U in V. Za poljubno majhen $\epsilon \in \left(0, \sqrt{2(\widehat{\sigma}_{r+1} - \widehat{\sigma}_r)}\right)$ velja

$$\|A - \underline{A}_r(\epsilon)\|_F^2 < \|A - A_r\|_F^2 < \|A - \overline{A}_r(\epsilon)\|_F^2,$$

kar pomeni, da je v točki A_r sedlo. □

1.3 Optimizacija

Problem nenegativne matrične faktorizacije je optimizacijski problem. V tem razdelku bomo spoznali osnove optimizacije. Na koncu si bomo ogledali algoritem koordinatnega spusta. Snov je povzeta po [3].

Matematični model pri optimizaciji je običajno predstavljen z *omejitveno množico* $\Omega \subseteq \mathbb{R}^n$ in *energijsko funkcijo* $f\colon \mathbb{R}^n \to \mathbb{R}$. Želimo najti optimalno rešitev $x^* \in \Omega$, ki izpolnjuje nek kriterij funkcije f. Običajno je to minimalna

$$f(x^*) = \min_{x \in \Omega} f(x), \tag{1.4}$$

ali maksimalna vrednost

$$f(x^*) = \max_{x \in \Omega} f(x),$$

funkcije f. V magistrski nalogi se bomo osredotočili na iskanje optimalne rešitve (1.4).

Definicija 1.3.1 (Lokalni minimum)**.** Vektorju $x^* \in \Omega$ pravimo *lokalni minimum* funkcije f nad Ω, če obstaja tak $\epsilon > 0$, da velja $f(x^*) \leq f(x)$ za vsak $x \in \Omega$, $\|x - x^*\| < \epsilon$. Vektorju x^* pravimo *strogi lokalni minimum*, če v zgornji enačbi velja stroga neenakost za $x^* \neq x$.

Definicija 1.3.2 (Globalni minimum)**.** Vektorju $x^* \in \Omega$ pravimo *globalni minimum* funkcije f nad Ω, če velja $f(x^*) \leq f(x)$ za vsak $x \in \Omega$. Vektorju x^* pravimo *strogi globalni minimum*, če v zgornji enačbi velja stroga neenakost za $x^* \neq x$.

1.3.1 Prost optimizacijski problem

Radi bi imeli pogoj, ki nam pove, kdaj je vektor $x^* \in \Omega$ optimalna rešitev energijske funkcije f. Za začetek obravnavajmo *prost optimizacijski problem*, tj. ko je $\Omega = \mathbb{R}^n$.

Trditev 1.3.3. *Naj bo vektor $x^* \in \mathbb{R}^n$ lokalni minimum funkcije $f\colon \mathbb{R}^n \to \mathbb{R}$ in naj bo f zvezno diferenciabilna v odprti okolici $S = \{x \in \mathbb{R}^n : \|x - x^*\| < \epsilon\}$ vektorja x^* za nek $\epsilon > 0$. Potem velja*

$$\nabla f(x^*) = 0. \tag{1.5}$$

Če je funkcija f dvakrat zvezno diferenciabilna znotraj okolice S, potem je

$$\nabla^2 f(x^*) \text{ pozitivno semidefinitna matrika.} \tag{1.6}$$

Opomba 1.3.4. Točki $x \in \mathbb{R}^n$ pravimo *stacionarna točka*, če zadostuje pogoju (1.5).

Dokaz. Najprej dokažimo (1.5). Naj bo funkcija f diferenciabilna. Naj bo $d \in \mathbb{R}^n$ poljuben vektor in $\alpha \in \mathbb{R}$ poljuben pozitiven skalar. Ker je funkcija f diferenciabilna, velja

$$f(x^* + \alpha d) = f(x^*) + \alpha \nabla f(x^*)^T d + o(\alpha).$$

Enačbi odštejemo $f(x^*)$ in jo delimo z α. Dobimo

$$0 \leq \frac{f(x^* + \alpha d) - f(x^*)}{\alpha} = \nabla f(x^*)^T d + \frac{o(\alpha)}{\alpha},$$

kjer neenakost sledi iz minimalnosti vektorja x^*. Ko pošljemo $\alpha \to 0$ in upoštevamo $\lim_{\alpha \to 0} \frac{o(\alpha)}{\alpha} = 0$, dobimo $0 \leq \nabla f(x^*)^T d$. Ker je vektor d poljuben, ga lahko zamenjamo z vektorjem $-d$ in dobimo $0 \leq -\nabla f(x^*)^T d$ oziroma $0 \geq \nabla f(x^*)^T d$. Torej velja $\nabla f(x^*)^T d = 0$ za vsak $d \in \mathbb{R}^n$, iz česar sledi $\nabla f(x^*) = 0$.

Dokaz (1.6) poteka na podoben način. Naj bo funkcija f dvakrat zvezno diferenciabilna, vektor $d \in \mathbb{R}^n$ poljuben in $\alpha \in \mathbb{R}$ poljuben pozitiven skalar. Funkcijo f razvijemo v Taylorjevo vrsto okoli točke x^* in dobimo

$$f(x^* + \alpha d) - f(x^*) = \alpha \nabla f(x^*)^T d + \frac{\alpha^2}{2} d^T \nabla^2 f(x^*) d + o(\alpha^2).$$

Upoštevamo minimalnost vektorja x^* in pogoj $\nabla f(x^*) = 0$. Potem obstaja tako število $\epsilon > 0$, da za vsak skalar $\alpha \in (0, \epsilon)$ velja

$$0 \leq \frac{f(x^* + \alpha d) - f(x^*)}{\alpha^2} = \frac{1}{2} d^T \nabla^2 f(x^*) d + \frac{o(\alpha^2)}{\alpha^2}.$$

Ko pošljemo $\alpha \downarrow 0$ in upoštevamo $\lim_{\alpha \downarrow 0} \frac{o(\alpha^2)}{\alpha^2} = 0$, dobimo $d^T \nabla^2 f(x^*) d \geq 0$. Ker to velja za poljuben vektor $d \in \mathbb{R}^n$, je matrika $\nabla^2 f(x^*)$ pozitivno semidefinitna. □

Globalni minimum funkcije f iščemo na naslednji način:

1. Poiščemo vse vektorje $M = \{x_1, x_2, \ldots, x_k\}$, ki zadoščajo pogojem (1.5) in (1.6).

2. Globalni minimum funkcije f je $x^* = \operatorname{argmin}_{x \in M} f(x)$.

1.3.2 Optimizacijski problem z omejitvami

Običajno iščemo optimalno rešitev znotraj množice $\Omega \subset \mathbb{R}^n$, ki jo definirajo enačbe oblike $h(x) = 0$ ali $g(x) \leq 0$. Takim optimizacijskim problemom pravimo *optimizacijski problemi z omejitvami*. V tem razdelku bomo obravnavali lastnosti množice Ω in funkcije f, ki lahko olajšajo iskanje optimalne rešitve.

Definicija 1.3.5 (Konveksna množica)**.** Naj bo Ω podmnožica množice $\mathbb{R}^n$. Množica Ω je *konveksna*, če velja

$$\alpha x + (1-\alpha) y \in \Omega$$

za vsaka $x, y \in \Omega$ in vsak $\alpha \in [0,1]$.

Primer 1.3.6. Pokažimo, da je množica vseh nenegativnih matrik z m vrsticami in n stolpci konveksna množica. Izberimo poljubni nenegativni matriki $A, B \in \mathbb{R}_+^{m\times n}$ in poljuben skalar $\alpha \in [0,1]$. Potem je $\alpha A + (1-\alpha)B \in \mathbb{R}_+^{m\times n}$. Torej je množica vseh nenegativnih matrik res konveksna množica.

Definicija 1.3.7 (Konveksna funkcija)**.** Naj bo Ω konveksna podmnožica množice $\mathbb{R}^n$. Naj bo $f\colon \mathbb{R}^n \to \mathbb{R}$ skalarna funkcija. Funkcija f je *konveksna*, če velja

$$f(\alpha x + (1-\alpha) y) \le \alpha f(x) + (1-\alpha) f(y) \tag{1.7}$$

za vsaka $x, y \in \Omega$ in vsak $\alpha \in [0,1]$. Če v (1.7) velja stroga neenakost za vse $x, y \in \Omega$, $x \neq y$ in $\alpha \in (0,1)$, potem pravimo, da je funkcija f *strogo konveksna*. Če je konveksna funkcija f definirana na konveksni množici Ω, pravimo, da je *funkcija f konveksna nad* Ω.

Lema 1.3.8. *Vsaka vektorska norma in linearna funkcija je konveksna funkcija.*

Dokaz. Naj bo $f\colon \mathbb{R}^n \to \mathbb{R}$ linearna funkcija, tj. velja $f(x+y) = f(x) + f(y)$ in $f(\lambda x) = \lambda f(x)$ za vsak $x, y \in \mathbb{R}^n$ in $\lambda \in \mathbb{R}$. Naj bosta $x, y \in \mathbb{R}^n$ vektorja in $\alpha \in [0,1]$ poljuben skalar. Potem iz lastnosti linearne funkcije sledi

$$\begin{aligned} f(\alpha x + (1-\alpha) y) &= f(\alpha x) + f((1-\alpha) y) \\ &= \alpha f(x) + (1-\alpha) f(y). \end{aligned}$$

Torej je linearna funkcija konveksna.

Naj bo zdaj $g\colon \mathbb{R}^n \to \mathbb{R}$ poljubna vektorska norma, tj. $g(x) = \|x\|$. Naj bosta $x, y \in \mathbb{R}^n$ vektorja in $\alpha \in [0,1]$ poljuben skalar. Dokažimo, da je funkcija g konveksna.

$$\begin{aligned} g(\alpha x + (1-\alpha) y) &= \|\alpha x + (1-\alpha) y\| \\ &\le \|\alpha x\| + \|(1-\alpha) y\| \\ &= \alpha \|x\| + (1-\alpha) \|y\| \\ &= \alpha g(x) + (1-\alpha) g(y), \end{aligned}$$

kjer smo upoštevali trikotniško neenakost norme. Dokazali smo konveksnost vektorske norme. □

Dokažimo naslednjo trditev, ki jo bomo uporabili pri določevanju optimizacijskih pogojev za optimizacijski problem z omejitvami.

Trditev 1.3.9. *Naj bo $\Omega \subset \mathbb{R}^n$ konveksna množica in naj bo funkcija $f\colon \mathbb{R}^n \to \mathbb{R}$ diferenciabilna nad Ω. Potem velja naslednje: funkcija f je konveksna nad Ω natanko tedaj, ko velja*

$$f(y) \geq f(x) + (y-x)^T \nabla f(x) \tag{1.8}$$

za vsaka $x, y \in \Omega$.

Dokaz. Dokažimo najprej implikacijo v desno. Naj bo funkcija f konveksna in diferenciabilna nad konveksno množico Ω. Naj bosta $x, y \in \Omega$ poljubna vektorja. Iz konveksnosti Ω sledi $x + \alpha(y-x) = \alpha y + (1-\alpha)x \in \Omega$ za vsak $\alpha \in [0,1]$. Zaradi diferenciabilnosti f velja

$$\lim_{\alpha \downarrow 0} \frac{f(x+\alpha(y-x)) - f(x)}{\alpha} = (y-x)^T \nabla f(x). \tag{1.9}$$

Zaradi konveksnosti funkcije f sledi

$$f(x+\alpha(y-x)) \leq \alpha f(y) + (1-\alpha) f(x),$$

iz česar sledi

$$\frac{f(x+\alpha(y-x)) - f(x)}{\alpha} \leq f(y) - f(x),$$

za vsak $\alpha \in [0,1]$. Ko pošljemo $\alpha \downarrow 0$ in upoštevamo (1.9), dobimo

$$\begin{aligned} (y-x)^T \nabla f(x) &\leq f(y) - f(x), \\ f(x) + (y-x)^T \nabla f(x) &\leq f(y). \end{aligned}$$

Dokažimo implikacijo v levo. Predpostavimo, da velja (1.8). Poljubno izberemo vektorja $x, y \in \Omega$ in skalar $\alpha \in [0,1]$. Naj bo $z = \alpha x + (1-\alpha) y$. Dvakrat uporabimo (1.8) na vektorjih x, y, z in dobimo

$$\begin{aligned} f(x) &\geq f(z) + (x-z)^T \nabla f(z), \\ f(y) &\geq f(z) + (y-z)^T \nabla f(z). \end{aligned}$$

Prvo neenakost množimo z α, drugo pa množimo z $(1-\alpha)$ in ju seštejemo. Dobimo

$$\begin{aligned} \alpha f(x) + (1-\alpha) f(y) &\geq f(z) + (\alpha x + (1-\alpha) y - z)^T \nabla f(z) \\ &= f(z) \\ &= f(\alpha x + (1-\alpha) y), \end{aligned}$$

s čimer smo dokazali konveksnost funkcije f. □

Definicija 1.3.10 (Dopustna smer)**.** Naj bo Ω podmnožica množice $\mathbb{R}^n$. Vektorju $d \in \Omega$ pravimo, da je *dopustna smer* za vektor $x^* \in \Omega$, če obstaja tako število $\epsilon > 0$, da je $x^* + \alpha d \in \Omega$ za vsak $\alpha \in [0, \epsilon]$.

Za optimizacijski problem z omejitvami želimo imeti podane optimizacijske pogoje. Iz obravnave prostega optimizacijskega problema vemo, da je $x^* \in \Omega$ lokalni minimum funkcije f, če velja $\nabla f(x^*)^T d \geq 0$ za vsak vektor $d \in \mathbb{R}^n$. Pri optimizacijskem problemu z omejitvami pa se omejimo samo na dopustne smeri. Če je Ω konveksna, potem je vsaka dopustna smer oblike $d = x - x^*$, kjer je $x \in \Omega$. Res, saj za poljuben $\alpha \in [0,1]$ velja

$$x^* + \alpha d = x^* + \alpha(x - x^*) = \alpha x + (1-\alpha)x^* \in \Omega.$$

Torej se optimizacijski pogoj $\nabla f(x^*)^T d \geq 0$ prepiše v

$$\nabla f(x^*)^T (x - x^*) \geq 0,$$

za vsak $x \in \Omega$. Dokažimo zdaj naslednjo trditev.

Trditev 1.3.11. *Naj bo Ω konveksna podmnožica množice $\mathbb{R}^n$. Naj bo $f\colon \Omega \to \mathbb{R}$ zvezno diferenciabilna funkcija in $x^* \in \Omega$ lokalni minimum funkcije f. Potem velja*

$$\nabla f(x^*)^T (x - x^*) \geq 0 \tag{1.10}$$

za vsak $x \in \Omega$. Če je funkcija f konveksna nad množico Ω, potem je vsak lokalni minimum funkcije f tudi globalni minimum.

Dokaz. Dokažimo najprej prvo trditev. Naj bo Ω konveksna podmnožica množice $\mathbb{R}^n$. Naj bo $f\colon \Omega \to \mathbb{R}$ zvezno diferenciabilna funkcija in $x^* \in \Omega$ lokalni minimum funkcije f. Dokazovali bomo s protislovjem. Recimo da velja $\nabla f(x^*)^T (x - x^*) < 0$ za nek $x \in \Omega$. Definirajmo funkcijo $g(\epsilon) = f(x^* + \epsilon(x - x^*))$. Uporabimo Langrangeov izrek [5, izrek 39] na funkciji g. Ta pravi, da za vsak $\epsilon > 0$ obstaja tak $s \in [0,1]$, da velja

$$\begin{aligned} g(\epsilon) - g(0) &= \epsilon g'(s\epsilon), \\ f(x^* + \epsilon(x - x^*)) - f(x^*) &= \epsilon \nabla f(x^* + s\epsilon(x - x^*))^T (x - x^*). \end{aligned}$$

Iz zveznosti funkcije ∇f sledi, da za vse dovolj majhne $\epsilon > 0$ velja $\nabla f(x^* + s\epsilon(x - x^*))^T (x - x^*) < 0$. Sledi neenakost

$$f(x^* + \epsilon(x - x^*)) < f(x^*).$$

Prišli smo do protislovja z lokalno minimalnostjo vektorja x^*, saj je $x^* + \epsilon(x - x^*) \in \Omega$ za vsak $\epsilon \in [0,1]$. S tem smo dokazali prvo trditev.

Dokažimo še drugo trditev. Predpostavimo, da je f konveksna nad množico Ω. Potem po trditvi 1.3.9 velja

$$f(x) \geq f(x^*) + \nabla f(x^*)^T (x - x^*)$$

za vsak vektor $x \in \Omega$. Če velja (1.10) za vse $x \in \Omega$, potem velja $f(x) \geq f(x^*)$. Vektor x^* je globalni minimum funkcije f na množici Ω. □

1.3.3 Lagrangeovi množitelji

Obravnavajmo primer, ko je množica Ω definirana z enačbami oblike $h(x) = 0$. Naj bodo $f, h_i \colon \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$, zvezne diferenciabilne funkcije. Iščemo optimalno rešitev problema

$$\min_{x \in \Omega} f(x),$$

kjer je $\Omega \subset \mathbb{R}^n$, $\Omega = \{x \in \mathbb{R}^n \colon h(x) = 0\}$ omejena množica in $h \colon \mathbb{R}^n \to \mathbb{R}^m$, $h = \begin{bmatrix} h_1 & h_2 & \cdots & h_m \end{bmatrix}^T$ omejitvena funkcija. Zapišimo zvezo med lokalnimi minimumi optimizacijskega problema in Lagrangeovo funkcijo.

Izrek 1.3.12 (Izrek o Lagrangeovih množiteljih)**.** *Naj bo $f \colon \mathbb{R}^n \to \mathbb{R}$ zvezna diferenciabilna funkcija. Naj bo x^* lokalni minimum funkcije f, ki izpolnjuje pogoj $h(x) = 0$. Predpostavimo, da so gradienti $\nabla h_1(x^*), \dots, \nabla h_m(x^*)$ linearno neodvisni. Potem obstajajo koeficienti $\lambda_1^*, \dots, \lambda_m^*$, ki rešijo enačbo*

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x^*) = 0. \tag{1.11}$$

Opomba 1.3.13. Koeficientom $\lambda_1^*, \dots, \lambda_m^*$ iz zgornjega izreka pravimo *Lagrangeovi množitelji.*

Dokaz. Pomagali si bomo z ugotovitvami iz analize prostega optimizacijskega problema. Za $k = 1, 2, \dots$ definirajmo funkcije

$$F_k(x) = f(x) + \frac{k}{2}\|h(x)\|^2 + \frac{\alpha}{2}\|x - x^*\|^2,$$

kjer je α poljuben pozitiven skalar in x^* lokalni minimum funkcije f, ki izpolnjuje pogoj $h(x) = 0$. Izraz $\frac{k}{2}\|h(x)\|^2$ v enačbi predstavlja kazen, če x ne izpolnjuje pogoja $h(x) = 0$. Izraz $\frac{\alpha}{2}\|x - x^*\|^2$ v enačbi zagotavlja, da je x^* strogi lokalni minimum funkcije $f(x) + \frac{\alpha}{2}\|x - x^*\|^2$, ki zadošča $h(x) = 0$. Te funkcije bomo najprej analizirali, nato jih bomo uporabili za dokaz izreka o Lagrangeovih množiteljih.

Ker je x^* lokalni minimum funkcije f, lahko izberemo tak $\epsilon > 0$, da velja $f(x^*) \leq f(x)$ za vse dopustne smeri x znotraj zaprte sfere

$$S = \{x \in \mathbb{R}^n \colon \|x - x^*\| \leq \epsilon\}.$$

Naj bo x_k optimalna rešitev problema

$$\min_{x \in S} F_k(x). \tag{1.12}$$

Rešitev obstaja, saj je funkcija F_k zvezna na zaprti množici S in po Weierstrassovem izreku [5, izrek 30] doseže svoj globalni minimum. Pokazali bomo, da zaporedje $\{x_k\}_{k=1}^{\infty}$ konvergira k vektorju x^*.

Za vsak k velja

$$F_k(x_k) = f(x_k) + \frac{k}{2}\|h(x_k)\|^2 + \frac{\alpha}{2}\|x_k - x^*\|^2 \leq F_k(x^*) = f(x^*). \tag{1.13}$$

Ker je f omejena na množici S, mora veljati

$$\lim_{k\to\infty} \|h(x_k)\| = 0,$$

sicer bi bila leva stran neenačbe (1.13) neomejena, ko gre k proti neskončnosti. Torej vsaka limita $\overline{x}$ zaporedja $\{x_k\}_{k=1}^{\infty}$ reši enačbo $h(\overline{x}) = 0$. Iz (1.13) dobimo tudi neenačbo $f(x_k) + \frac{\alpha}{2}\|x_k - x^*\|^2 \leq f(x^*)$, ki velja za vsak k. Ko pošljemo k proti neskončnosti dobimo

$$f(\overline{x}) + \frac{\alpha}{2}\|\overline{x} - x^*\|^2 \leq f(x^*).$$

Ker je $\overline{x} \in S$ in je dopustna smer za x^*, iz $f(x^*) \leq f(\overline{x})$ v kombinaciji s prejšnjo neenakostjo sledi

$$f(\overline{x}) + \frac{\alpha}{2}\|\overline{x} - x^*\|^2 \leq f(\overline{x}),$$

iz česar sledi $\|\overline{x} - x^*\| = 0$, kar pomeni $\overline{x} = x^*$. Dokazali smo, da zaporedje $\{x_k\}_{k=1}^{\infty}$ konvergira k vektorju x^*. Iz tega sledi, da so vsi x_k, od nekega dovolj poznega člena dalje, v notranjosti množice S. Vektor x_k je lokalni minimum prostega optimizacijskega problema

$$\min_{x\in S} F_k(x).$$

Dokažimo izrek o Lagrangeovih množiteljih. Uporabimo pogoj (1.5) iz trditve 1.3.3 na funkciji F_k. Za dovolj velike k dobimo

$$0 = \nabla F_k(x_k) = \nabla f(x_k) + k\nabla h(x_k)h(x_k) + \alpha(x_k - x^*). \tag{1.14}$$

Matrika $\nabla h(x^*)$ je ranga m, saj so vektorji $\nabla h_1(x^*), \ldots, \nabla h_m(x^*)$ linearno neodvisni. Zaradi zveznosti so matrike $\nabla h(x_k)$ ranga m za dovolj velike k. Pri takih k je matrika $\nabla h(x_k)^T \nabla h(x_k) \in \mathbb{R}^{m\times m}$ obrnljiva. Enačbo (1.14) z leve strani množimo z matriko $(\nabla h(x_k)^T \nabla h(x_k))^{-1} \nabla h(x_k)^T$ in dobimo

$$kh(x_k) = -(\nabla h(x_k)^T \nabla h(x_k))^{-1} \nabla h(x_k)^T (\nabla f(x_k) + \alpha(x_k - x^*)).$$

Ko pošljemo k proti neskončnosti in $x_k \to x^*$, zaporedje $\{kh(x_k)\}_{k=1}^{\infty}$ konvergira k vektorju

$$\lambda^* = -(\nabla h(x^*)^T \nabla h(x^*))^{-1} \nabla h(x^*)^T \nabla f(x^*).$$

Ko v enačbi (1.14) pošljemo $k \to \infty$, dobimo

$$\nabla f(x^*) + \nabla h(x^*)\lambda^* = 0,$$

s čimer smo dokazali izrek o Langrangeovih množiteljih. □

Sedaj obravnavajmo primer, ko je množica $\Omega \subset \mathbb{R}^n$ definirana tudi z neenačbami oblike $g(x) \leq 0$. Naj bodo f, h_i, $g_j \colon \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$, $j = 1, \dots, r$, zvezne diferenciabilne funkcije. Iščemo optimalno rešitev problema

$$\min_{x \in \Omega} f(x),$$

kjer je $\Omega \subset \mathbb{R}^n$, $\Omega = \{x \in \mathbb{R}^n \colon h(x) = 0, g(x) \leq 0\}$ omejena množica, funkciji $h \colon \mathbb{R}^n \to \mathbb{R}^m$, $h = [\, h_1 \quad h_2 \quad \cdots \quad h_m \,]^T$ in $g \colon \mathbb{R}^n \to \mathbb{R}^r$, $g = [\, g_1 \quad g_2 \quad \cdots \quad g_r \,]^T$ pa omejitveni diferenciabilni funkciji. Naslednji izrek nam pove, kakšna je povezava med Lagrangeovo funkcijo in zgornjim optimizacijskim problemom.

Izrek 1.3.14 (Karush-Kuhn-Tuckerjevi optimizacijski pogoji)**.** *Naj bo $f \colon \mathbb{R}^n \to \mathbb{R}$ zvezna diferenciabilna funkcija. Naj bo x^* lokalni minimum problema*

$$\min_{x \in \Omega} f(x),$$

kjer je $\Omega = \{x \in \mathbb{R}^n \colon h(x) = 0, g(x) \leq 0\}$ podmnožica množice $\mathbb{R}^n$. Naj bodo vrstice matrik ∇h in ∇g linearno neodvisne. Potem obstajajo Lagrangeovi množitelji $\lambda_1^*, \dots, \lambda_m^*$ in $\mu_1^*, \dots, \mu_r^*$, ki rešijo

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x^*) + \sum_{j=1}^{r} \mu_j^* \nabla g_j(x^*) = 0, \tag{1.15}$$

$$\begin{aligned} \mu_j^* &\geq 0, \quad j = 1, \dots, r, \\ \mu_j^* &= 0, \quad j \notin A(x^*), \end{aligned}$$

kjer je $A(x) = \{j \colon g_j(x) = 0\}$ množica aktivnih omejitev vektorja x.

Opomba 1.3.15. V zgornjem izreku lahko lastnost $\mu_j^* = 0$, $j \notin A(x^*)$ zapišemo tudi kot $\mu_j^* g_j(x^*) = 0$ za $j = 1, \dots, r$, saj iz $g_j(x^*) < 0$ sledi $\mu_j^* = 0$.

Dokaz. Ideja dokaza. Dokaz poteka na podoben način kot za izrek 1.3.12. Definirajmo najprej funkcije

$$g_j^+(x) = \max\{0, g_j(x)\}, \quad j = 1, \dots, r.$$

Funkcije g_j^+ predstavljajo kazen, če pogoj $g_j(x) \leq 0$ ni izpolnjen. Podobno kot pri dokazu izreka 1.3.12, za $k = 1, 2, \dots$ definiramo funkcije

$$F_k(x) = f(x) + \frac{k}{2}\|h(x)\|^2 + \frac{k}{2}\sum_{j=1}^{r} \left(g_j^+(x)\right)^2 + \frac{\alpha}{2}\|x - x^*\|^2,$$

kjer je $\alpha \in \mathbb{R}$ pozitiven skalar. Naj bo $S = \{x \in \mathbb{R}^n \colon \|x - x^*\| \leq \epsilon\}$ zaprta množica in $\epsilon > 0$ tak, da velja $f(x^*) \leq f(x)$ za vse dopustne smeri $x \in S$. Opazimo, da je funkcija $(g_j^+)^2$ zvezno diferenciabilna z gradientom $2g_j^+(x)\nabla g_j(x)$. Če je x_k lokalni minimum funkcije F_k znotraj množice S, potem lahko na podoben način kot pri izreku 1.3.12 dokažemo, da so Lagrangeovi množitelji λ_i^* in μ_j^* podani z

$$\begin{aligned} \lambda_i^* &= \lim_{k\to\infty} kh_i(x_k), \quad i = 1, \ldots, m, \\ \mu_j^* &= \lim_{k\to\infty} kg_j^+(x_k) \quad j = 1, \ldots, r. \end{aligned}$$

Ker je $g_j^+(x_k) \geq 0$, potem so tudi $\mu_j^* \geq 0$ za vse j. $\square$

Opomba 1.3.16. Pogoj iz izrekov 1.3.12 in 1.3.14 lahko opišemo tudi z iskanjem lokalnega ekstrema *Lagrangeove funkcije*

$$L(x, \lambda_1, \lambda_2, \ldots, \lambda_m, \mu_1, \mu_2, \ldots, \mu_r) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{r} \mu_j g_j(x).$$

Lokalni ekstrem funkcije L nam vrne lokalni ekstrem x^* funkcije f in Lagrangeove množitelje $\lambda_1^*, \ldots, \lambda_m^*, \mu_1^*, \ldots, \mu_r^*$, ki zadoščajo pogojem iz izrekov 1.3.12 in 1.3.14.

1.3.4 Algoritem koordinatnega spusta

Obstaja več metod za iskanje optimalnih rešitev optimizacijskih problemov. V tem razdelku bomo opisali *algoritem koordinatnega spusta*.

Denimo, da želimo poiskati optimalno rešitev problema

$$\min_{x\in\Omega} f(x),$$

kjer je $\Omega \subset \mathbb{R}^n$ konveksna množica. Predpostavimo, da je Ω kartezični produkt konveksnih množic $\Omega_1, \ldots, \Omega_m$, kjer je $\Omega_i \subset \mathbb{R}^{n_i}$ za $i = 1, \ldots, m$ in $\sum_{i=1}^m n_i = n$. Vektor x ustrezno bločno razdelimo

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix},$$

kjer je $x_i \in \Omega_i$ za $i = 1, \ldots, m$.

Algoritem koordinatnega spusta minimizira energijsko funkcijo f za vsako koordinato x_i, $i = 1, \ldots, m$, posebej. Postopek je opisan v algoritmu 1.

Če v koordinatnem spustu korak 4 izračunamo točno in je minimum enolično določen, potem se postopek ustavi v stacionarni točki funkcije f. To opisuje naslednji izrek, ki je dokazan v [3].

Algoritem 1: Koordinatni spust

Podatki: Funkcija $f\colon \mathbb{R}^n \to \mathbb{R}$

Rezultat: Vektor x, ki minimizira funkcijo f na konveksni množici Ω

Inicializiraj začetni vektor $x = (x_1, x_2, \dots, x_m)^T$;
ponavljaj
    za $i = 1, \dots, m$ **naredi**
        Reši $x_i = \operatorname{argmin}_{\zeta \in \Omega_i} f(x_1, \dots, x_{i-1}, \zeta, x_{i+1}, \dots, x_m)$;
dokler *zaustavitveni pogoj*;

Izrek 1.3.17. *Naj bo f zvezno diferenciabilna funkcija nad zgoraj opisano množico Ω. Predpostavimo tudi, da je za vsak $i \in \{1, \dots, m\}$ in vsak vektor $x \in \Omega$ rešitev problema*

$$\operatorname*{argmin}_{\zeta \in \Omega_i} f(x_1, \dots, x_{i-1}, \zeta, x_{i+1}, \dots, x_m)$$

enolično določena. Potem zaporedje $\{x^k\}_{k=1}^{\infty}$, ki ga generira algoritem koordinatnega spusta, konvergira k stacionarni točki.

Algoritem bomo uporabili v razdelku 3, ko bomo spoznali različne algoritme za reševanje problema nenegativne matrične faktorizacije.

Poglavje 2

Nenegativna matrična faktorizacija

V tem poglavju bomo spoznali problem nenegativne matrične faktorizacije. Najprej bomo definirali problem in analizirali njegove lastnosti. Nato bomo poiskali lokalne minimume, kjer bomo ugotovili, da je to težak problem. Na koncu poglavja se bomo soočili z iskanjem točne nenegativne matrične faktorizacije. Poglavje je povzeto po [10].

2.1 Opis problema

Za dano nenegativno matriko $A \in \mathbb{R}_+^{m \times n}$ in naravno število $r < \min\{m, n\}$ želimo poiskati taki nenegativni matriki $U \in \mathbb{R}_+^{m \times r}$ in $V \in \mathbb{R}_+^{n \times r}$, ki dobro aproksimirata matriko A, tj.

$$A \approx UV^T.$$

Opazimo, da lahko vsak stolpec matrike A aproksimiramo z linearno kombinacijo stolpcev matrike U,

$$A_{:i} \approx \sum_{j=1}^{r} V_{ij} U_{:j}.$$

Problem lahko opišemo na drug način: za dane vektorje $a_1, \ldots, a_n \in \mathbb{R}_+^m$ in naravno število $r < \min\{m, n\}$ poišči take nenegativne vektorje $u_1, \ldots, u_r \in \mathbb{R}_+^m$, ki jih dobro aproksimirajo, tj.

$$a_i \approx \sum_{j=1}^{r} v_{ij} u_j,$$

kjer so v_{ij} nenegativni skalarji. Podobnost med matriko A in UV^T se da opisati na več načinov. Najpogostejša mera podobnosti je definirana s Frobeniusovo normo.

$$F(A, UV^T) = \frac{1}{2} \|A - UV^T\|_F^2 = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(A_{ij} - (UV^T)_{ij} \right)^2$$

Pri našem raziskovanju problema bomo uporabili zgornjo mero podobnosti. Formulirajmo osrednji problem.

Problem 2.1.1 (Nenegativna matrična faktorizacija - NMF). *Naj bo $A \in \mathbb{R}_+^{m\times n}$ nenegativna matrika in naj bo $r < \min\{m,n\}$ naravno število. Poišči optimalno rešitev*

$$\min_{U\in\mathbb{R}_+^{m\times r},\, V\in\mathbb{R}_+^{n\times r}} \frac{1}{2}\|A-UV^T\|_F^2. \tag{2.1}$$

Opomba 2.1.2. Številu r iz problema 2.1.1 pravimo *reduciran rang*.

Energijska funkcija, podana v problemu NMF, je enaka

$$F(U,V) = \frac{1}{2}\|A-UV^T\|_F^2.$$

Če fiksiramo matriko U, je funkcija F kompozitum Frobeniusove norme in linearne transformacije matrike V. Torej je funkcija F konveksna nad $\mathbb{R}_+^{n\times r}$. Prav tako, če fiksiramo V, je funkcija F konveksna nad $\mathbb{R}_+^{m\times r}$. V splošnem pa funkcija F ni konveksna.

Opazimo, da je NMF optimizacijski problem z omejitvami, kjer so iskane spremenljivke elementi matrike U in V. Enačbo (2.1) lahko prepišemo v standardno obliko optimizacijskega problema z omejitvami

$$\min_{-U\le 0,\, -V\le 0} \frac{1}{2}\|A-UV^T\|_F^2.$$

Rešitev iščemo v prostoru, ki je definiran s funkcijama $g_1(U) = U \ge 0$ in $g_2(V) = V \ge 0$, oziroma $-U \le 0$ in $-V \le 0$. Pripadajoča Lagrangeova funkcija je enaka

$$L(U,V,\mu,\nu) = \frac{1}{2}\|A-UV^T\|_F^2 - \langle \mu, U\rangle - \langle \nu, V\rangle,$$

kjer matrika $\mu \in \mathbb{R}^{m\times r}$ vsebuje Lagrangeove množitelje asociirane z nenegativnostnimi omejitvami $U_{ij} \ge 0$, matrika $\nu \in \mathbb{R}^{n\times r}$ pa vsebuje Lagrangeove množitelje asociirane z nenegativnostnimi omejitvami $V_{ij} \ge 0$.

Recimo, da je par (U, V) lokalni minimum funkcje F. Potem po Karush-Kuhn-Tuckerjevih optimizacijskih pogojih (izrek 1.3.14) obstajajo taki Lagrangeovi množitelji $\mu_{ij} \ge 0$ in $\nu_{ij} \ge 0$, da velja

$$U \ge 0, \qquad V \ge 0, \tag{2.2}$$
$$\nabla L_U = 0, \qquad \nabla L_V = 0, \tag{2.3}$$
$$\mu \circ U = 0, \qquad \nu \circ V = 0. \tag{2.4}$$

Najprej si oglejmo, kaj nam pove pogoj (2.3). Razpišimo funkcijo L.

$$\begin{aligned} L(U,V,\mu,\nu) &= \frac{1}{2}\|A-UV^T\|_F^2 - \langle \mu, U\rangle - \langle \nu, V\rangle \\ &= \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^n (A_{ij}-(UV^T)_{ij})^2 - \sum_{i=1}^m\sum_{j=1}^r \mu_{ij}U_{ij} - \sum_{i=1}^n\sum_{j=1}^r \nu_{ij}V_{ij} \\ &= \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^n (A_{ij}-\sum_{k=1}^r U_{ik}V_{kj}^T)^2 - \sum_{i=1}^m\sum_{j=1}^r \mu_{ij}U_{ij} - \sum_{i=1}^n\sum_{j=1}^r \nu_{ij}V_{ij}. \end{aligned}$$

Gradient funkcije L po matriki U je enak

$$\nabla L_U = \begin{bmatrix} \frac{\partial L}{\partial U_{11}} & \frac{\partial L}{\partial U_{12}} & \cdots & \frac{\partial L}{\partial U_{1r}} \\ \frac{\partial L}{\partial U_{21}} & \frac{\partial L}{\partial U_{22}} & \cdots & \frac{\partial L}{\partial U_{2r}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial U_{m1}} & \frac{\partial L}{\partial U_{m2}} & \cdots & \frac{\partial L}{\partial U_{mr}} \end{bmatrix}.$$

Izračunamo parcialne odvode

$$\begin{aligned} \frac{\partial L}{\partial U_{t\ell}}(U,V,\mu,\nu) &= \frac{1}{2}\sum_{j=1}^n 2\Big((A_{tj}-\sum_{k=1}^r U_{tk}V_{kj}^T)(-V_{\ell j}^T)\Big) - \mu_{t\ell} \\ &= \sum_{j=1}^n \big(-A_{tj}V_{\ell j}^T + \sum_{k=1}^r (U_{tk}V_{kj}^T)V_{\ell j}^T\big) - \mu_{t\ell} \\ &= \sum_{j=1}^n(-A_{tj}V_{j\ell}) + \sum_{j=1}^n \big((UV^T)_{tj}V_{j\ell}\big) - \mu_{t\ell} \\ &= (-AV)_{t\ell} + (UV^TV)_{t\ell} - \mu_{t\ell}, \end{aligned}$$

in jih vstavimo v matriko ∇L_U. Dobimo

$$\nabla L_U = UV^TV - AV - \mu.$$

Izračunamo še gradient funkcije L po matriki V.

$$\nabla L_V = \begin{bmatrix} \frac{\partial L}{\partial V_{11}} & \frac{\partial L}{\partial V_{12}} & \cdots & \frac{\partial L}{\partial V_{1r}} \\ \frac{\partial L}{\partial V_{21}} & \frac{\partial L}{\partial V_{22}} & \cdots & \frac{\partial L}{\partial V_{2r}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial V_{n1}} & \frac{\partial L}{\partial V_{n2}} & \cdots & \frac{\partial L}{\partial V_{nr}} \end{bmatrix}$$

Parcialne odvode

$$
\begin{aligned}
\frac{\partial L}{\partial V_{t\ell}}(U,V,\mu,\nu) &= \frac{1}{2}\sum_{i=1}^{m} 2(A_{it} - \sum_{k=1}^{r} U_{ik}V_{kt}^T)(-U_{i\ell}) - \nu_{t\ell} \\
&= \sum_{i=1}^{m}\big(-A_{it}U_{i\ell} + \sum_{k=1}^{r}(V_{tk}U_{ki}^T)U_{i\ell}\big) - \nu_{t\ell} \\
&= \sum_{i=1}^{m}(-A_{ti}^T U_{i\ell}) + \sum_{i=1}^{m}\big((VU^T)_{ti}U_{i\ell}\big) - \nu_{t\ell} \\
&= (-A^T U)_{t\ell} + (VU^T U)_{t\ell} - \nu_{t\ell}
\end{aligned}
$$

vstavimo v matriko ∇L_V in dobimo

$$\nabla L_V = VU^T U - A^T U - \nu.$$

Iz pogoja (2.3) dobimo

$$UV^T V - AV - \mu = 0, \quad VU^T U - A^T U - \nu = 0,$$

oziroma

$$\mu = UV^T V - AV, \quad \nu = VU^T U - A^T U. \tag{2.5}$$

Enačbi (2.5) v kombinaciji s pogoji $\mu_{ij} \geq 0$, $\nu_{ij} \geq 0$ in (2.4) vrneta naslednje optimizacijske pogoje:

$$
\begin{aligned}
U \geq 0, &\qquad V \geq 0, && (2.6)\\
UV^T V - AV \geq 0, &\qquad VU^T U - A^T U \geq 0, && (2.7)\\
(UV^T V - AV) \circ U = 0, &\qquad (VU^T U - A^T U) \circ V = 0. && (2.8)
\end{aligned}
$$

Opomba 2.1.3. Matriki Lagrangeovih množiteljev μ in ν sta gradienta ∇F_U in ∇F_V, tj. $\mu = \nabla F_U = UV^T V - AV$ in $\nu = \nabla F_V = VU^T U - A^T U$.

Ker funkcija F ni konveksna, pari matrik (U, V), ki zadostujejo zgornjim pogojem, niso nujno lokalni minimumi. Lahko so tudi sedla ali pa lokalni maksimumi. Na tem mestu definiramo stacionarno točko.

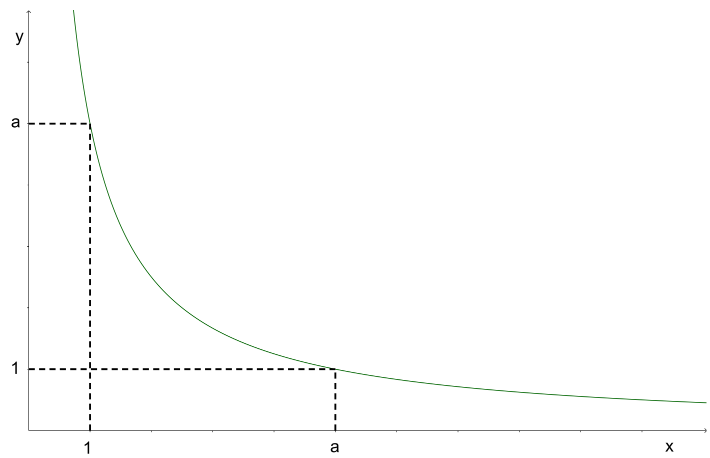
Definicija 2.1.4 (Stacionarna točka NMF)**.** Paru matrik (U, V) pravimo *stacionarna točka NMF* natanko tedaj, ko matriki U in V zadostujeta pogojem (2.6), (2.7) in (2.8).

V splošnem je lahko več rešitev NMF. To bomo pokazali v naslednjih primerih. Oglejmo si najprej preprost primer.

Primer 2.1.5. Imamo skalar $a \in \mathbb{R}_+$. Iščemo dve nenegativni števili $u, v \in \mathbb{R}_+$, ki rešita problem

$$\min_{u\in\mathbb{R}_+, v\in\mathbb{R}_+} |a - uv|^2 = (a - uv)^2.$$

Rešitev problema je točka (u, v), ki leži na grafu $uv = a$ (slika 2.1). Sledi, da je rešitev neskončno. Če zahtevamo, da je skalar u unitaren, tj. $\|u\|_2 = 1$, potem dobimo eno samo rešitev $u = 1$ in $v = a$.



Slika 2.1: Graf $a = uv$

Oglejmo si še geometrijsko interpretacijo NMF.

Primer 2.1.6. Naj bo $A \in \mathbb{R}_+^{m\times n}$ nenegativna matrika, za katero smo našli taki nenegativni matriki $U \in \mathbb{R}_+^{m\times r}$ in $V \in \mathbb{R}_+^{n\times r}$, da velja $A = UV^T$. Na začetku poglavja smo ugotovili, da velja

$$A_{:i} = \sum_{j=1}^{r} V_{ij} U_{:j}. \tag{2.9}$$

Denimo, da imamo množico nenegativnih vektorjev $\Phi = \{\varphi_1, \ldots, \varphi_k\}$. Definirajmo *simplicialen stožec* kot množico

$$\Gamma_\Phi = \{x \in \mathbb{R}^m \colon x = \sum_{i=1}^{k} \alpha_i \varphi_i,\ \alpha_i \geq 0\}.$$

Iz (2.9) sledi, da so stolpci matrike A vsebovani v simplicialnem stožcu generiranem s stolpci matrike U, tj.

$$A_{:i} \in \Gamma_U = \{x \in \mathbb{R}^m \colon x = \sum_{j=1}^{r} \alpha_j U_{:j},\ \alpha_j \geq 0\} \text{ za } i = 1, \ldots, m.$$

Potem vsak simplicialen stožec $\Gamma_{\widehat{U}}$, za katerega velja $\Gamma_U \subset \Gamma_{\widehat{U}}$, vsebuje stolpce matrike A. Torej obstaja nova faktorizacija matrike $A = \widehat{U}\widehat{V}^T$.

V nadaljevanju bomo iskali take rešitve NMF, kjer ima matrika U normirane stolpce, tj. $\|U_{:i}\|_2 = 1$ za $i = 1, \ldots, r$. To v splošnem ne zagotavlja enoličnosti aproksimacije. Prav tako ne vemo, kdaj in kako lahko dosežemo enoličnost rešitve NMF.

Definicija 2.1.7 (Ekvivalenca dveh rešitev NMF)**.** Dve rešitvi NMF (U_1, V_1) in (U_2, V_2) sta *ekvivalentni* natanko tedaj, ko vrneta enak produkt, tj. $U_1 V_1^T = U_2 V_2^T$.

Recimo, da imamo stacionarno točko (U, V) in tako obrnljivo matriko S, da sta matriki $\widehat{U} = US$ in $\widehat{V} = V(S^{-1})^T$ nenegativni. Ali je par $(\widehat{U}, \widehat{V})$ ekvivalenten paru (U, V)? Najprej moramo preveriti, če je par $(\widehat{U}, \widehat{V})$ stacionarna točka NMF. Zadostovati mora pogojem (2.6), (2.7) in (2.8).

Pogoj (2.6) je izpolnjen zaradi izbire matrike S. Izpolnjena morata biti še pogoj (2.7),

$$
\begin{aligned}
\widehat{U}\widehat{V}^T\widehat{V} - A\widehat{V} &= US(V(S^{-1})^T)^T V(S^{-1})^T - AV(S^{-1})^T \\
&= USS^{-1}V^T V(S^{-1})^T - AV(S^{-1})^T \\
&= (UV^T V - AV)(S^{-1})^T \\
&\geq 0, \\
\widehat{V}\widehat{U}^T\widehat{U} - A^T\widehat{U} &= V(S^{-1})^T (US)^T US - A^T US \\
&= V(S^T)^{-1} S^T U^T US - A^T US \\
&= (VU^T U - A^T U)S \\
&\geq 0,
\end{aligned}
$$

in pogoj (2.8),

$$
\begin{aligned}
(\widehat{U}\widehat{V}^T\widehat{V} - A\widehat{V}) \circ \widehat{U} &= (US(V(S^{-1})^T)^T V(S^{-1})^T - AV(S^{-1})^T) \circ US \\
&= \big((UV^T V - AV)(S^{-1})^T\big) \circ US \\
&= 0, \\
(\widehat{V}\widehat{U}^T\widehat{U} - A^T\widehat{U}) \circ \widehat{V} &= (V(S^{-1})^T (US)^T US - A^T US) \circ V(S^{-1})^T \\
&= \big((VU^T U - A^T U)S\big) \circ V(S^{-1})^T \\
&= 0.
\end{aligned}
$$

Če je S permutacijska matrika, potem so stolpci matrike $\widehat{U}$ ravno permutirani stolpci matrike U in stolpci matrike $\widehat{V}$ so permutirani stolpci matrike V. Torej je par $(\widehat{U}, \widehat{V})$ praktično enak paru (U, V). Za permutacijsko matriko S so zgornji pogoji izpolnjeni. Ker velja

$$\widehat{U}\widehat{V}^T = US\big(V(S^{-1})^T\big)^T = USS^{-1}V^T = UV^T,$$

je par $(\widehat{U}, \widehat{V})$ ekvivalenten paru (U, V). Opazimo tudi, da matrika S ne more biti nenegativna monomialna matrika, saj smo določili, da morata matriki $\widehat{U}$ in U imeti normirane stolpce. Oglejmo si naslednji primer.

Primer 2.1.8. Naj bo par (U, V) stacionarna točka problema NMF za matriko $A \in \mathbb{R}^{3\times 3}$ in reduciran rang $r = 2$ in naj ima matrika U normirane stolpce. Naj bo

$$M = \begin{bmatrix} 0 & 3 \\ 2 & 0 \end{bmatrix}$$

monomialna matrika. Potem matrika $\widehat{U} = UM$ nima normiranih stolpcev, saj je $\|\widehat{U}_{:1}\|_2 = \|2U_{:2}\|_2 = 2$ in $\|\widehat{U}_{:2}\|_2 = \|3U_{:1}\|_2 = 3$.

Preverjanje pogojev (2.6), (2.7) in (2.8) za splošno matriko S ni enostavno.

2.2 Rešitev NMF

Globalno rešitev NMF poznamo za dva reducirana ranga. Ko je reduciran rang $r = 1$, je par dominantnih singularnih vektorjev matrike A globalna rešitev NMF. Primer $r = \min\{m, n\}$ ločimo na dve možnosti.

1. Če je $r = m$, potem je par matrik $U = I$, $V = A^T$ globalni minimum, kjer je $I \in \mathbb{R}^{m\times m}$ matrika identitete.

2. Če je $r = n$, potem je par matrik

$$U = \left[\frac{A_{:1}}{\|A_{:1}\|_2} \cdots \frac{A_{:n}}{\|A_{:n}\|_2}\right], \quad V = \begin{bmatrix} \|A_{:1}\|_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \|A_{:n}\|_2 \end{bmatrix}$$

   globalni minimum problema NMF.

Ker je večina znanih algoritmov NMF neka oblika koordinatnega spusta, se bomo osredotočili na iskanje lokalnih minimumov.

Najprej bomo poiskali rešitev za primer, ko je reduciran rang $r = 1$. Takrat je problem NMF oblike

$$\min_{u\in\mathbb{R}^m_+,\, v\in\mathbb{R}^n_+} \frac{1}{2}\|A - uv^T\|_F^2. \tag{2.10}$$

Da se dokazati, da vsak par nenegativnih Perronovih lastnih vektorjev matrik AA^T in A^TA porodi optimalno rešitev problema (2.10). Lahko tudi dokažemo, da je vsaka stacionarna točka problema porojena s Perronovimi lastnimi vektorji. Naslednji izrek ne obravnava primera, ko je $u = 0$ ali $v = 0$.

Izrek 2.2.1. *Par (u, v) je lokalni minimum problema (2.10) natanko tedaj, ko je u nenegativen lastni vektor matrike AA^T in v nenegativen lastni vektor matrike $A^T A$ za lastno vrednost $\sigma = \|u\|_2^2 \|v\|_2^2$.*

Dokaz. Dokažimo implikacijo v levo. Naj bo u nenegativen lastni vektor matrike AA^T in v nenegativen lastni vektor matrike $A^T A$. Lastna vektorja sta singularna vektorja matrike A. Iz izrekov 1.2.5 in 1.2.6 sledi, da je par (u, v) lokalni minimum enačbe (2.10).

Dokažimo še implikacijo v desno. Brez izgube za splošnost lahko permutiramo vrstice in stolpce matrike A, da bosta vektorja u in v oblike

$$u = \begin{bmatrix} u_+ \\ 0 \end{bmatrix}, \quad v = \begin{bmatrix} v_+ \\ 0 \end{bmatrix},$$

kjer sta u_+ in v_+ pozitivna vektorja. Matriko A ustrezno razbijemo na particije

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

Iz optimizacijskega pogoja (2.7) dobimo

$$\begin{aligned} uv^T v - Av &= \begin{bmatrix} u_+ v_+^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_+ \\ 0 \end{bmatrix} - \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} v_+ \\ 0 \end{bmatrix} \geq 0, \\ vu^T u - A^T u &= \begin{bmatrix} v_+ u_+^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_+ \\ 0 \end{bmatrix} - \begin{bmatrix} A_{11}^T & A_{12}^T \\ A_{21}^T & A_{22}^T \end{bmatrix} \begin{bmatrix} u_+ \\ 0 \end{bmatrix} \geq 0. \end{aligned}$$

Opazimo, da mora veljati $A_{21} v_+ \leq 0$ in $A_{12}^T u_+ \leq 0$. Ker sta A_{21}, A_{12} nenegativna bloka in sta vektorja u_+, v_+ pozitivna, sledi $A_{12} = 0$ in $A_{21} = 0$. Iz optimizacijskega pogoja (2.8) dobimo

$$\begin{aligned} u \circ (uv^T v - Av) &= \begin{bmatrix} u_+ \\ 0 \end{bmatrix} \circ \left(\|v_+\|_2^2 \begin{bmatrix} u_+ \\ 0 \end{bmatrix} - \begin{bmatrix} A_{11} v_+ \\ 0 \end{bmatrix} \right) = 0, \\ v \circ (vu^T u - A^T u) &= \begin{bmatrix} v_+ \\ 0 \end{bmatrix} \circ \left(\|u_+\|_2^2 \begin{bmatrix} v_+ \\ 0 \end{bmatrix} - \begin{bmatrix} A_{11}^T u_+ \\ 0 \end{bmatrix} \right) = 0. \end{aligned}$$

Izpišemo samo neničelne koordinate in dobimo

$$\begin{aligned} u_+ \circ \left(\|v_+\|_2^2 u_+ - A_{11} v_+ \right) &= 0, \\ v_+ \circ \left(\|u_+\|_2^2 v_+ - A_{11}^T u_+ \right) &= 0. \end{aligned}$$

Ker sta u_+ in v_+ pozitivna, sledi

$$\|v_+\|_2^2 u_+ = A_{11} v_+, \qquad \|u_+\|_2^2 v_+ = A_{11}^T u_+,$$

oziroma

$$\|u_+\|_2^2 \|v_+\|_2^2 u_+ = A_{11} A_{11}^T u_+, \qquad \|u_+\|_2^2 \|v_+\|_2^2 v_+ = A_{11}^T A_{11} v_+.$$

Ko označimo $\lambda = \|u_+\|_2^2 \|v_+\|_2^2$ in upoštevamo bločno diagonalno strukturo matrike A, dobimo

$$AA^T u = \lambda u, \qquad A^T A v = \lambda v,$$

kar smo dokazovali. □

Izrek 2.2.1 nam zagotavlja, da so vse stacionarne točke problema (2.10) nenegativni singularni vektorji neke podmatrike matrike A. To namiguje, da lahko izračunamo točno nenegativno faktorizacijo, če vzamemo največjo singularno vrednost matrike A in njene pripadajoče singularne vektorje.

Za reduciran rang različen od 1 ali $\min\{m, n\}$ ni trivialnih stacionarnih točk. Zato si oglejmo nekaj preprostih lastnosti lokalnih minimumov NMF.

2.2.1 Lastnosti lokalnih minimumov

Iz pogoja (2.8) bomo izpeljali lastnosti stacionarnih točk problema NMF. Če seštejemo vse elemente prve enačbe (2.8), dobimo

$$\begin{aligned} 0 &= \sum_{i=1}^{m} \sum_{j=1}^{r} \Big(U \circ (UV^T V - AV) \Big)_{ij} \\ &= \langle U, UV^T V - AV \rangle \\ &= \langle UV^T, UV^T - A \rangle. \end{aligned} \tag{2.11}$$

S to enačbo lahko pokažemo nekaj preprostih lastnosti.

Trditev 2.2.2. *Naj bo par nenegativnih matrik (U, V) stacionarna točka problema NMF. Potem je $UV^T \in \mathcal{B}\Big(\frac{A}{2}, \frac{1}{2}\|A\|_F\Big)$, tj. matrika UV^T je vsebovana v krogli s središčem v $\frac{A}{2}$ in radijem $\frac{1}{2}\|A\|_F$.*

Dokaz. Predpostavimo, da je par (U, V) stacionarna točka problema NMF za matriko A. Potem je

$$\begin{aligned}
\left\langle \frac{A}{2} - UV^T, \frac{A}{2} - UV^T \right\rangle &= \left\langle \frac{A}{2}, \frac{A}{2} \right\rangle - 2\left\langle \frac{A}{2}, UV^T \right\rangle + \left\langle UV^T, UV^T \right\rangle \\
&= \left\langle \frac{A}{2}, \frac{A}{2} \right\rangle - \left\langle A, UV^T \right\rangle + \left\langle UV^T, UV^T \right\rangle \\
&= \left\langle \frac{A}{2}, \frac{A}{2} \right\rangle + \left\langle UV^T - A, UV^T \right\rangle \\
&\overset{(2.11)}{=} \left\langle \frac{A}{2}, \frac{A}{2} \right\rangle,
\end{aligned}$$

iz česar sledi $UV^T \in \mathcal{B}\left(\frac{A}{2}, \frac{1}{2}\|A\|_F\right)$. □

Trditev 2.2.3. *Naj bo par nenegativnih matrik (U, V) stacionarna točka problema NMF. Potem velja*

$$\frac{1}{2}\|A - UV^T\|_F^2 = \frac{1}{2}(\|A\|_F^2 - \|UV^T\|_F^2).$$

Dokaz. Naj bo par (U, V) stacionarna točka NMF za matriko A. Iz enačbe (2.11) sledi

$$\langle UV^T, A \rangle = \langle UV^T, UV^T \rangle. \tag{2.12}$$

$$\begin{aligned}
\frac{1}{2}\|A - UV^T\|_F^2 &= \frac{1}{2}\langle A - UV^T, A - UV^T \rangle \\
&= \frac{1}{2}\Big(\langle A, A \rangle - 2\langle UV^T, A \rangle + \langle UV^T, UV^T \rangle\Big) \\
&\overset{(2.12)}{=} \frac{1}{2}\Big(\langle A, A \rangle - \langle UV^T, UV^T \rangle\Big) \\
&= \frac{1}{2}\Big(\|A\|_F^2 - \|UV^T\|_F^2\Big).
\end{aligned}$$

□

V razdelku 1.2 smo pokazali ekvivalenco med aproksimacijo matrike z matriko manjšega ranga (problem 1.2.2) in matrično faktorizacijo (problem 1.2.4). Pokazali smo če je matrika A_r rešitev problema 1.2.2 in $A_r = U\Sigma V^T$ njen singularni razcep, potem je par $(X = U\Sigma^{\frac{1}{2}}, Y = V\Sigma^{\frac{1}{2}})$ rešitev problema 1.2.4.

Ugotovili smo tudi, da so globalni minimumi edine stabilne stacionarne točke problema 1.2.2. Če se stacionarne točke problema 1.2.4 nahajajo v prostoru nenegativnih matrik, potem je mogoče doseči globalni minimum problema NMF.

Iz trditve 2.2.3 sledi, da za stacionarno točko (U, V) velja $\|A\|_F^2 \geq \|UV^T\|_F^2$. Enakost je dosežena v primeru, ko je $A = UV^T$. O tem bomo govorili v naslednjem razdelku.

Naj bo A_r optimalna rešitev problema 1.2.2 za nenegativno matriko A. Z $[A_r]_+$ označimo projekcijo matrike A_r na prostor nenegativnih matrik. Dokažimo naslednji izrek, ki govori o omejitvi napake $\|A - [A_r]_+\|_F$.

Trditev 2.2.4. *Naj bo A_r najboljša aproksimacija nenegativne matrike A v prostoru matrik ranga kvečjemu r. Naj bo $[A_r]_+$ projekcija matrike A_r na prostor nenegativnih matrik. Potem velja*

$$\|A - [A_r]_+\|_F \leq \|A - A_r\|_F.$$

Dokaz. Naj bo A_r najboljša aproksimacija nenegativne matrike A v prostoru matrik ranga kvečjemu r in $[A_r]_+$ projekcija matrike A_r na prostor nenegativnih matrik. Naj bo $[A_r]_-$ projekcija matrike A_r na prostor nepozitivnih matrik. Zanj velja

$$([A]_-)_{ij} = \begin{cases} -A_{ij}, & \text{če velja } A_{ij} < 0, \\ 0, & \text{sicer} \end{cases} = -\min\{0, A_{ij}\}.$$

Matriko A_r lahko zapišemo kot $A_r = [A_r]_+ - [A_r]_-$.

$$\begin{aligned}
\|A - A_r\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - (A_r)_{ij})^2 \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - ([A_r]_+ - [A_r]_-)_{ij})^2 \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(\mathbb{1}_{([A_r]_+)_{ij}>0}(A_{ij} - ([A_r]_+)_{ij}) + \mathbb{1}_{([A_r]_+)_{ij}=0}(A_{ij} + ([A_r]_-)_{ij})\Big)^2 \\
&\geq \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(\mathbb{1}_{([A_r]_+)_{ij}>0}(A_{ij} - ([A_r]_+)_{ij}) + \mathbb{1}_{([A_r]_+)_{ij}=0}A_{ij}\Big)^2 \\
&= \|A - [A_r]_+\|_F^2,
\end{aligned}$$

kjer je

$$\mathbb{1}_x = \begin{cases} 1, & \text{če velja } x, \\ 0, & \text{sicer.} \end{cases}$$

□

Zgornja neenakost ne velja samo za matriko A_r, ampak velja za splošno matriko B.

Trditev 2.2.5. *Naj bo $A \in \mathbb{R}_+^{m\times n}$ nenegativna matrika in $B \in \mathbb{R}^{m\times n}$ poljubna matrika. Naj bo $[B]_+$ projekcija matrike B na prostor nenegativnih matrik. Tedaj velja*

$$\|A - [B]_+\|_F \leq \|A - B\|_F.$$

Dokaz. Dokaz poteka na podoben način kot za trditev 2.2.4. □

Naj bo $U_* V_*^T$ optimalna rešitev problema NMF za nenegativno matriko A in reduciran rang r. Naj bo UV^T stacionarna točka problema NMF ranga r. Potem velja

$$\|A - [A_r]_+\|_F^2 \leq \|A - A_r\|_F^2 = \sum_{i=r+1}^{n} \sigma_i^2 \leq \|A - U_* V_*^T\|_F^2 \leq \|A - UV^T\|_F^2.$$

2.3 Točna faktorizacija

V tem razdelku bomo iskali točno nenegativno matrično faktorizacijo, tj. za nenegativno matriko $A \in \mathbb{R}_+^{m \times n}$ bomo iskali taki nenegativni matriki U in V, da velja $A = UV^T$. Podrobna analiza je podana v [6].

Definicija 2.3.1 (Nenegativen rang)**.** Naj bo $A \in \mathbb{R}_+^{m \times n}$. Številu $r \in \mathbb{N}$ pravimo *nenegativen rang* matrike A, če je najmanjše tako število, za katero obstaja točen razcep $A = UV^T$, kjer sta $U \in \mathbb{R}_+^{m \times r}$ in $V \in \mathbb{R}_+^{n \times r}$. Nenegativen rang bomo označili z $\mathrm{rang}_{UV^T}^+(A)$.

Obstoj točne faktorizacije za reduciran rang r je ekvivalenten določanju števila $\mathrm{rang}_{UV^T}^+(A)$. Za vsako število $r > \mathrm{rang}_{UV^T}^+(A)$ lahko konstruiramo točno nenegativno faktorizacijo iz faktorizacije UV^T tako, da matrikama U in V dodamo ustrezno število ničelnih stolpcev.

Lema 2.3.2 (Omejitev nenegativnega ranga)**.** *Naj bo $A \in \mathbb{R}_+^{m \times n}$. Potem velja*

$$\mathrm{rang}(A) \leq \mathrm{rang}_{UV^T}^+(A) \leq \min\{m, n\}.$$

Dokaz. Ker ne moremo konstruirati iste matrike z matrikami manjšega ranga, sledi prva neenakost. Ker lahko matriko A razcepimo na produkt AI_n ali pa $I_m A$, kjer je I_k matrika identitete dimenzije $k \times k$, sledi druga neenakost. □

V nekaterih primerih velja $\mathrm{rang}(A) = \mathrm{rang}_{UV^T}^+(A)$. Če je $\mathrm{rang}(A) = 1$, potem je matrika $A = uv^T$, kjer sta u in v nenegativna vektorja. To je tudi nenegativna matrična faktorizacija matrike A. Potem velja $\mathrm{rang}(A) = \mathrm{rang}_{UV^T}^+(A) = 1$. Če je $\mathrm{rang}(A) = 2$, enakost še vedno velja.

Lema 2.3.3. *Naj bo $A \in \mathbb{R}_+^{m \times n}$ nenegativna matrika ranga 2. Potem velja*

$$\mathrm{rang}_{UV^T}^+(A) = 2.$$

Dokaz se nahaja v [6]. Naj bo A nenegativna matrika ranga 2. Matriko U generiramo tako, da vzamemo par stolpcev matrike A, med katerim je največji kot, tj.

$$\begin{aligned} (u_1, u_2) &= \underset{A_{:i}, A_{:j}}{\operatorname{argmin}} \frac{A_{:i}^T A_{:j}}{\|A_{:i}\| \|A_{:j}\|}, \\ U &= [u_1, u_2] . \end{aligned}$$

Matrika V je rešitev enačbe $UV^T = A$ po metodi najmanjših kvadratov.

Pokazali smo, kako se izračuna točno nenegativno matrično faktorizacijo matrike A, za katero velja $\operatorname{rang}(A) \in \{1, 2, \min\{m, n\}\}$. Za ostale range pa je izračun nenegativnega ranga težak problem. V [23] so dokazali, da je točna nenegativna matrična faktorizacija *NP-težak* problem. Za te probleme se predvideva, da se jih ne da rešiti v polinomskem času. V [6] so dokazali obstoj algoritma za določevanje nenegativnega ranga, ki ga izračuna v eksponentnem času.

Zaradi težavnosti iskanja točne nenegativne matrične faktorizacije, vsi NMF algoritmi iščejo taki matriki U in V, da dobro aproksimirata matriko A. Matriki se izračuna preko standardnih nelinearnih optimizacijskih metod, ki zagotavljajo le konvergenco k stacionarni točki, ne pa nujno k optimalni rešitvi.

Poglavje 3

Algoritmi NMF

Algoritme za nenegativno matrično faktorizacijo ločimo v dve kategoriji.

- *Iskanje po celem prostoru* poskuša posodobiti matriki U in V istočasno. To zahteva iskanje v $(m+n)r$-dimenzionalnem prostoru. V razdelku 2.1 smo že ugotovili, da energijska funkcija $F(U,V) = \frac{1}{2}\|A - UV^T\|_F^2$ ni konveksna v celem prostoru. Zato lahko algoritem konvergira k točki, ki ni optimalna rešitev. Se pa lažje dokaže konvergenco.

- *Bločno iskanje* poskuša posodobiti vsak blok koordinat posebej. To zagotavlja manjšanje vrednosti energijske funkcije. Običajno se izbere konveksne podprostore, na katerih se uporabi učinkovite algoritme. Ta način posodabljanja lahko povzroči izgubo nekaterih lastnosti konvergence.

  Večina algoritmov uporablja stolpično particijo,

  $$\frac{1}{2}\|A - UV^T\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\|A_{:i} - UV_{:i}^T\|_2^2.$$

  Problem se razdeli na več manjših konveksnih problemov, saj lahko iščemo optimalno rešitev za vsak stolpec matrike V^T posebej. Potem moramo poiskati rešitev problema

  $$\min_{v\in\mathbb{R}_+^r} \frac{1}{2}\|a - Uv\|_2^2, \tag{3.1}$$

  kjer je $a \in \mathbb{R}_+^m$ stolpec matrike A. Problem imenujemo *problem nenegativnih najmanjših kvadratov*. Ker velja $\|A - UV^T\|_F^2 = \|A^T - VU^T\|_F^2$, lahko matriko U posodobimo z istim postopkom, ki smo ga uporabili za V.

V tem poglavju bomo spoznali različne algoritme za reševanje problema nenegative matrične faktorizacije. Nato se bomo soočili s težavo inicializacije algoritmov in določevanjem zaustavitvenega kriterija. Na koncu poglavja bomo algoritme med seboj primerjali. Poglavje je povzeto po [10].

3.1 Algoritem multiplikativnega pravila

Najpopularnejši algoritem za reševanje NMF problema je *algoritem multiplikativnega pravila*, ki sta ga Lee in Seung predstavila v [14]. Algoritem je primer bločnega iskanja optimalne rešitve. Preden ga spoznamo, bomo definirali naslednjo oznako.

Naj bo $x \in \mathbb{R}^n$ poljuben vektor. Diagonalna matrika $D_x \in \mathbb{R}^{n\times n}$ je generirana z vektorjem x. Zanjo velja

$$(D_x)_{ii} = x_i.$$

Oglejmo si, kako so razvili algoritem. Denimo, da imamo nenegativno matriko $A \in \mathbb{R}_+^{m\times n}$, reduciran rang $r < \min\{m, n\}$ in da sta matriki U in V pozitivni. Ideja algoritma je, da fiksiramo eno izmed matrik U ali V in rešimo problem nenegativnih najmanjših kvadratov za drugo matriko. Za začetek fiksirajmo matriko U. Najti moramo optimalno rešitev problema

$$\min_{V\in\mathbb{R}_+^{n\times r}} \|A - UV^T\|_F^2.$$

Vemo, da velja $\frac{1}{2}\|A - UV^T\|_F^2 = \frac{1}{2}\sum_{i=1}^n \|A_{:i} - UV_{:i}^T\|_2^2$. Zato se lahko osredotočimo na iskanje posameznega stolpca matrike V^T. Potrebno je poiskati optimalno rešitev problema nenegativnih najmanjših kvadratov (problem 3.1), kjer je $a \in \mathbb{R}_+^m$ stolpec matrike A in $v \in \mathbb{R}_+^r$ stolpec matrike V^T.

Denimo, da smo rešitev problema (3.1) aproksimirali s pozitivnim vektorjem $\widehat{v}$. Formulirajmo problem

$$\min_{v\in\mathbb{R}_+^r} \widehat{F}(v) = \min_{v\in\mathbb{R}_+^r} \left(\|a - Uv\|_2^2 + (v-\widehat{v})^T H_{\widehat{v}}(v-\widehat{v})\right), \tag{3.2}$$

kjer je $H_{\widehat{v}} = D_x - U^TU$ in $x = \frac{[U^TU\widehat{v}]}{[\widehat{v}]}$.

Pokažimo pozitivno semidefinitnost matrike $H_{\widehat{v}}$. Za vsak vektor $z \in \mathbb{R}^r$ mora veljati $z^T H_{\widehat{v}} z \geq 0$.

$$\begin{aligned}
z^T H_{\widehat{v}} z &= z^T D_x z - z^T U^T U z \\
&= \sum_{i=1}^r \frac{(U^TU\widehat{v})_i}{\widehat{v}_i} z_i^2 - \|Uz\|_2^2 \\
&= \sum_{i=1}^r \frac{U_{:i}^T U\widehat{v}}{\widehat{v}_i} z_i^2 - \sum_{i=1}^r\sum_{j=1}^r (U_{:i}^T U_{:j}) z_i z_j \\
&= \sum_{i=1}^r\sum_{j=1}^r \frac{\widehat{v}_j}{\widehat{v}_i}(U_{:i}^T U_{:j}) z_i^2 - \sum_{i=1}^r\sum_{j=1}^r (U_{:i}^T U_{:j}) z_i z_j. \qquad (3.3)
\end{aligned}$$

Seštejemo i,j-ti in j,i-ti člen zgornje vsote. Dobimo

$$\begin{aligned}\frac{\widehat{v}_j}{\widehat{v}_i}(U_{:i}^T U_{:j})z_i^2 + \frac{\widehat{v}_i}{\widehat{v}_j}(U_{:j}^T U_{:i})z_j^2 - 2(U_{:i}^T U_{:j})z_i z_j &= \left(z_i\sqrt{\frac{\widehat{v}_j}{\widehat{v}_i}U_{:i}^T U_{:j}} - z_j\sqrt{\frac{\widehat{v}_i}{\widehat{v}_j}U_{:i}^T U_{:j}}\right)^2 \\ &\geq 0.\end{aligned}$$

V primeru $i = j$ je vsota členov enaka

$$\frac{\widehat{v}_i}{\widehat{v}_i}(U_{:i}^T U_{:i})z_i^2 - (U_{:i}^T U_{:i})z_i z_i = 0.$$

Vsota (3.3) je večja ali enaka 0. S tem smo pokazali pozitivno semidefinitnost matrike $H_{\widehat{v}}$.

Ker je matrika $H_{\widehat{v}}$ pozitivno semidefinitna, je $\widehat{F}(v) \geq F(v)$ za vse $v \in \mathbb{R}_+^r$ in $\widehat{F}(\widehat{v}) = F(\widehat{v})$, kjer je F energijska funkcija problema (3.1). Izračunajmo gradient funkcije $\widehat{F}$.

$$\nabla\widehat{F} = \begin{bmatrix} \frac{\partial\widehat{F}}{\partial v_1} \\ \vdots \\ \frac{\partial\widehat{F}}{\partial v_r} \end{bmatrix}$$

Razpišemo funkcijo $\widehat{F}$.

$$\begin{aligned}\widehat{F}(v) &= \frac{1}{2}\big(\|a - U\widehat{v}\|_2^2 + (v-\widehat{v})^T H_{\widehat{v}}(v-\widehat{v})\big) \\ &= \frac{1}{2}\big(\|a - U\widehat{v}\|_2^2 + (v-\widehat{v})^T D_x(v-\widehat{v}) - (v-\widehat{v})^T U^T U(v-\widehat{v})\big) \\ &= \frac{1}{2}\sum_{k=1}^m(a_k - \sum_{j=1}^r U_{kj}v_j)^2 + \frac{1}{2}\sum_{i=1}^r\sum_{k=1}^r \frac{(v_i-\widehat{v}_i)}{\widehat{v}_i}(U^TU)_{ik}\widehat{v}_k \\ &\quad -\frac{1}{2}\sum_{i=1}^r\sum_{k=1}^r(v_i-\widehat{v}_i)(U^TU)_{ik}(v_k-\widehat{v}_k).\end{aligned}$$

Parcialne odvode

$$\begin{aligned}\frac{\partial\widehat{F}}{\partial v_t}(v) &= \sum_{k=1}^m(a_k - \sum_{j=1}^r U_{kj}v_j)(-U_{kt}) \\ &\quad + \frac{(v_t-\widehat{v}_t)}{\widehat{v}_t}\sum_{k=1}^r(U^TU)_{tk}\widehat{v}_k - \sum_{k=1}^r(U^TU)_{tk}(v_k-\widehat{v}_k) \\ &= \sum_{k=1}^m\Big((U^T)_{tk}(Uv)_k - (U^T)_{tk}a_k\Big) + \frac{(U^TU\widehat{v})_t}{\widehat{v}_t}(v_t-\widehat{v}_t) - \big(U^TU(v-\widehat{v})\big)_t \\ &= (U^TUv)_t - (U^Ta)_t + (D_t(v-\widehat{v}))_t - (U^TU(v-\widehat{v}))_t \\ &= (U^TUv)_t - (U^Ta)_t + (H_{\widehat{v}}(v-\widehat{v}))_t\end{aligned}$$

vstavimo v $\nabla\widehat{F}$ in ga enačimo z 0. Dobimo

$$\nabla\widehat{F} = U^T U v - U^T a + H_{\widehat{v}}(v - \widehat{v}) = 0.$$

Naj bo v^* rešitev zgornje enačbe. Zanj velja

$$(U^T U + H_{\widehat{v}})v^* = U^T a + H_{\widehat{v}}\widehat{v}. \tag{3.4}$$

Ker veljata enakosti

1. $U^T U + H_{\widehat{v}} = U^T U + D_x - U^T U = D_x = D_{U^T U\widehat{v}} D_{\widehat{v}}^{-1}$,
2. $H_{\widehat{v}} = D_x\widehat{v} - U^T U\widehat{v} = 0$,

lahko enačbo (3.4) prepišemo v

$$D_{U^T U\widehat{v}} D_{\widehat{v}}^{-1} v^* = U^T a.$$

Izrazimo i-to koordinato rešitve v^* in dobimo

$$\begin{aligned} \frac{(U^T U\widehat{v})_i}{\widehat{v}_i} v_i^* &= (U^T a)_i, \\ v_i^* &= \widehat{v}_i \frac{(U^T a)_i}{(U^T U\widehat{v})_i}. \end{aligned}$$

Rešitev v^* dobimo z enačbo

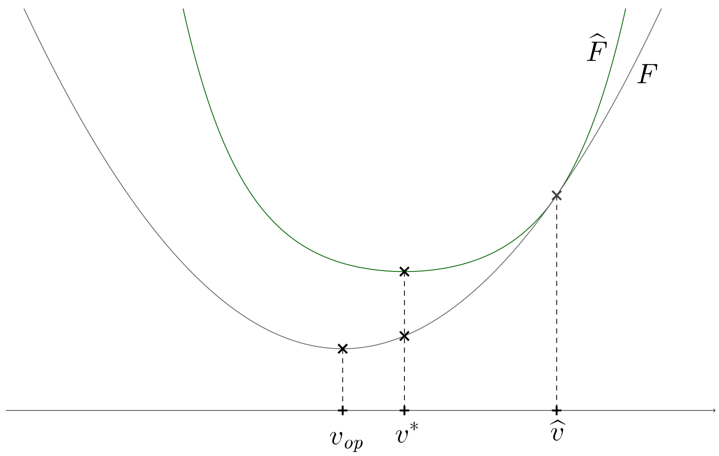
$$v^* = \widehat{v} \circ \frac{[U^T a]}{[U^T U\widehat{v}]}. \tag{3.5}$$

Potek uporabimo na vseh stolpcih matrike in s tem posodobimo matriko V^T. Na podoben način posodobimo matriko U. Postopek je zapisan v algoritmu 2.

Algoritem 2: Multiplikativno pravilo (Mult)

Podatki: Matrika $A \in \mathbb{R}_+^{m\times n}$ in naravno število $r < \min\{m, n\}$
Rezultat: Nenegativni matriki $U \in \mathbb{R}_+^{m\times r}$ in $V \in \mathbb{R}_+^{n\times r}$

1 Inicializiraj matriki U_0 in V_0 in postavi $k = 0$;
2 **ponavljaj**
3 $\quad U_{k+1} = U_k \circ \frac{[AV_k]}{[U_k V_k^T V_k]}$;
4 $\quad V_{k+1} = V_k \circ \frac{[A^T U_{k+1}]}{[V_k U_{k+1}^T U_{k+1}]}$;
5 $\quad k = k + 1$;
6 **dokler** *zaustavitveni pogoj*;



Slika 3.1: Prikaz nenaraščanja energijske funkcije s posodobitvami metode multiplikativnega pravila. Vektor $\widehat{v}$ je trenutna aproksimacija rešitve problema (3.1), vektor v^* je optimalna rešitev problema (3.2) in vektor v_{op} je optimalna rešitev problema (3.1).

Ker je v^* globalni minimum funkcije $\widehat{F}$, velja $\widehat{F}(v^*) \leq \widehat{F}(\widehat{v})$. Spomnimo se, da je funkcija $\widehat{F}$ izbrana tako, da velja $\widehat{F}(v) \geq F(v)$ za vsak vektor $v \in \mathbb{R}^n_+$. Sledi $F(v^*) \leq \widehat{F}(v^*) \leq \widehat{F}(\widehat{v}) = F(\widehat{v})$. Dobimo zaporedje, ki manjša vrednost energijske funkcije F (slika 3.1). Ugotovitev je zapisana v naslednjem izreku. Dokaz se nahaja v [13].

Izrek 3.1.1. *Za matriki U_k in V_k iz algoritma 2 velja $\|A - U_{k+1}V_{k+1}^T\|_F^2 \leq \|A - U_kV_k^T\|_F^2$.*

Algoritem multiplikativnega pravila v splošnem ne zagotavlja konvergence k stacionarni točki NMF. Če posodobitve matrik U in V vračajo pozitivne matrike, potem algoritem konvergira k stacionarni točki. Opazimo tudi, če je $(U_k)_{ij} = 0$, potem je $(U_\ell)_{ij} = 0$ za vse $\ell \geq k$. Enako velja za matrike V_k. Zato moramo matrike U_k in V_k ohraniti pozitivne za vse k. Naslednji izrek pove, kdaj se pozitivnost matrik U_k in V_k ohranja. Najdemo ga v [15].

Izrek 3.1.2. *Če sta matriki U_0 in V_0 pozitivni ter matrika A nima ničelnih vrstic in stolpcev, potem so matrike U_k in V_k, dobljene z algoritmom 2, pozitivne za vse $k \geq 0$.*

Dokaz. Dokazujemo z indukcijo. Naj bo $k = 0$. Matriki U_0 in V_0 sta po predpostavki pozitivni. Predpostavimo, da velja za število k in dokazujemo, da velja za $k + 1$. Ker so elementi U_k in V_k pozitivni ter matrika A nima ničelnih stolpcev ali vrstic, sta matriki AV_k in $U_kV_k^TV_k$ pozitivni, iz česar sledi pozitivnost matrike U_{k+1}. Podobno sledi pozitivnost matrik A^TU_{k+1}, $V_kU_{k+1}^TU_{k+1}$ in V_{k+1}. $\square$

Izrek ne zagotavlja odsotnosti težav pri posodobitvah matrik U_k in V_k. Če matrika AV_k ali $A^T U_{k+1}$ vsebuje majhne vrednosti, se lahko nekatere vrednosti matrik U_{k+1} ali V_{k+1} zaokrožijo v 0. Da to preprečimo, se ničelne vrednosti poveča na ustrezno veliko število ϵ, ki je večje od najmanjšega predstavljivega števila.

Obstaja tudi možnost, da je $(V_k U_k^T U_k)_{ij} = 0$ za neka indeksa i, j. Takrat pri posodobitvah delimo s številom nič. Oglejmo si dva primera:

1. Če je $(V_k)_{ij} > 0$, potem iz $(V_k U_k^T U_k)_{ij} = 0$ sledi

$$\begin{aligned} 0 &= \sum_{\ell=1}^{m} \sum_{t=1}^{r} (V_k)_{it} (U_k)_{\ell t} (U_k)_{\ell j} \\ &\geq \sum_{\ell=1}^{m} (V_k)_{ij} (U_k)_{\ell j} (U_k)_{\ell j} \\ &= (V_k)_{ij} \sum_{\ell=1}^{m} (U_k)_{\ell j} (U_k)_{\ell j}. \end{aligned}$$

   To se zgodi, ko je $(U_k)_{:j} = 0$. Takrat matrika U_k ni polnega ranga. V tem primeru j-ti stolpec matrike U_k zamenjamo s takim nenegativnim vektorjem, da je $\operatorname{rang}(U_k) = r$. Ena izbira je zamenjava z naključno generiranim nenegativnim vektorjem. Za tak vektor je malo verjetno, da je linearna kombinacija preostalih $r-1$ stolpcev matrike U_k.

2. Če je $(V_k)_{ij} = 0$, potem smo v primeru $\frac{0}{0}$ in velja

$$\nabla F_{V_{ij}} = -(AU_k)_{ij} \leq 0.$$

   Takrat se nam ne splača zamenjati vrednosti $(V_k U_k^T U_k)_{ij}$ z nekim pozitivnim številom ϵ, saj to ohranja $(V_{k+1})_{ij} = 0$. Boljša izbira je, da postavimo $(V_{k+1})_{ij} = \epsilon$, kjer je ϵ večji od najmanjšega predstavljivega števila.

Podobno analizo lahko naredimo za primer, ko je $(U_k V_k^T V_k)_{i,j} = 0$ za neka indeksa i, j.

Izračunajmo časovno zahtevnost algoritma multiplikativnega pravila. Za posodobitev matrike U (korak 3) in posodobitev matrike V (korak 4) moramo opraviti enako število operacij, ki je $2(mnr + mr^2 + nr^2) - r^2$. Za eno iteracijo algoritma je potrebno opraviti $4(mnr + mr^2 + nr^2) - 2r^2$ operacij. Ker je običajno $r \ll \min\{m, n\}$, je časovna zahtevnost algoritma enaka $\mathcal{O}(tmnr)$, kjer je t število iteracij.

V [8] so razvili algoritem multiplikativnega pravila, ki ima hitrejši spust energijske funkcije, niso pa rešili problema konvergence.

3.2 Metoda alternirajočih najmanjših kvadratov

V razdelku 2.1 smo pokazali, da je energijska funkcija konveksna, če fiksiramo eno izmed matrik U ali V. S fiksiranjem matrike se problem NMF preoblikuje v *problem nenegativnih najmanjših kvadratov*.

Algoritem 3 opisuje splošen potek reševanja z *metodo alternirajočih najmanjših kvadratov*.

Algoritem 3: Alternirajoči najmanjši kvadrati (ALS)

Podatki: Matrika $A \in \mathbb{R}_+^{m\times n}$ in naravno število $r < \min\{m, n\}$
Rezultat: Nenegativni matriki $U \in \mathbb{R}_+^{m\times r}$ in $V \in \mathbb{R}_+^{n\times r}$

1 Inicializiraj matriki U_0 in V_0 ter postavi $k = 0$;
2 **ponavljaj**
3   Reši: $V_{k+1} = \operatorname{argmin}_{V\in\mathbb{R}_+^{n\times r}} \frac{1}{2}\|A - U_k V^T\|_F^2$;
4   Reši: $U_{k+1} = \operatorname{argmin}_{U\in\mathbb{R}_+^{m\times r}} \frac{1}{2}\|A^T - V_{k+1} U^T\|_F^2$;
5   k = k + 1;
6 **dokler** *zaustavitveni pogoj*;

Če matriki V_{k+1} in U_{k+1} izračunamo točno in enolično, potem iz izreka 1.3.17 sledi, da je vsaka limitna točka dobljena z algoritmom alternirajočih najmanjših kvadratov stacionarna točka problema NMF.

Recimo, da želimo rešiti problem, ki je opisan v koraku 3. Ker velja

$$\frac{1}{2}\|A - UV^T\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\|A_{:i} - UV_{:i}^T\|_2^2,$$

se lahko osredotočimo na reševanje problema nenegativnih najmanjših kvadratov (3.1). Problem lahko rešimo z *metodo aktivnih množic*, ki je opisana v [12]. Alternativna metoda za reševanje nenegativnega problema najmanjših kvadratov je *projektivna gradientna metoda*. Ta uporabi algoritem gradientnega spusta, ki ga bomo opisali naslednjem razdelku. Več je opisano v [15].

Reševanje nenegativnega problema najmanjših kvadratov je počasno. Za izračun rešitve z uporabo metode aktivnih množic moramo opravili $\mathcal{O}(2^r tmnr)$ operacij, kjer je t število iteracij. Zato so predstavili postopek, ki rešitev nenegativnega problema najmanjših kvadratov nadomesti s projekcijo rešitve klasičnega problema najmanjših kvadratov na prostor nenegativnih matrik. Postopek je opisan v algoritmu 4.

Pri algoritmu netočnih alternirajočih najmanjših kvadratov porabimo največ časa za reševanje problema v korakih 3 in 5. Rešimo ju lahko s konstrukcijo normalnega sistema, ki ga nato rešimo z uporabo razcepa Choleskega [21]. Zanj je

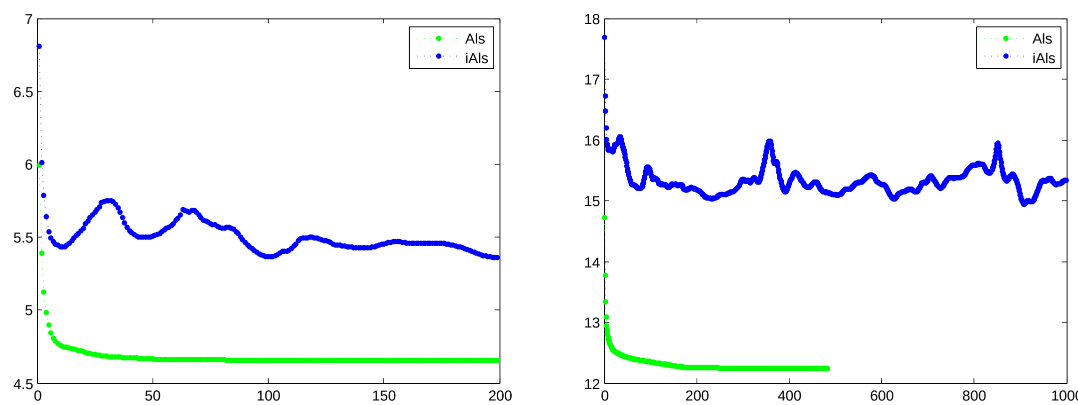
Algoritem 4: Netočni alternirajoči najmanjši kvadrati (iALS)

Podatki: Matrika $A \in \mathbb{R}_+^{m \times n}$ in naravno število $r < \min\{m, n\}$
Rezultat: Nenegativni matriki $U \in \mathbb{R}_+^{m \times r}$ in $V \in \mathbb{R}_+^{n \times r}$

Inicializiraj matriki U_0 in V_0 ter število $k = 0$;
ponavljaj
  Reši: $U_{k+1}^* = \operatorname{argmin}_{U \in \mathbb{R}^{m \times r}} \|A - UV_k^T\|_F$;
  $U_{k+1} = [U_{k+1}^*]_+$;
  Reši: $V_{k+1}^* = \operatorname{argmin}_{V \in \mathbb{R}^{n \times r}} \|A^T - VU_{k+1}^T\|_F$;
  $V_{k+1} = [V_{k+1}^*]_+$;
dokler *zaustavitevni pogoj*;

potrebno opraviti $mn^2 + \frac{1}{3}n^3 + \mathcal{O}(n^2)$ operacij. Ta način reševanja ni stabilen, zato se običajno uporabi stabilnejše metode, kot je reševanje z uporabo QR razcepa. Razcep lahko izračunamo z uporabo Gram-Schmidtove ortogonalizacije, ki zahteva približno $2mn^2$ operacij, kar je približno dvakrat toliko kot pri reševanju normalnega sistema. Ostale stabilne metode so reševanje z uporabo Givensovih rotacij, Householderjevih zrcaljenj in s singularnim razcepom. Več je napisano v [21, poglavje 5].

Algoritem iALS je hitrejši od algoritma ALS, vendar ne konvergira k stacionarni točki (slika 3.2).



(a) $m = 50, n = 20, r = 10$ (b) $m = 100, n = 50, r = 20$

Slika 3.2: Grafa prikazujeta vrednost energijske funkcije NMF problema izračunanega z uporabo algoritma ALS in iALS. Vhodna podatka sta naključna matrika $A \in \mathbb{R}_+^{m \times n}$ in reduciran rang r. Zaustavitveni pogoji pri izračunu grafa (3.2a) so bili $\epsilon = 10^{-6}$, $i = 200$ in $t = 60$, kjer je i omejitev števila iteracij in t časovna omejitev v sekundah. Zaustavitveni pogoji pri izračunu grafa (3.2b) so bili $\epsilon = 10^{-6}$, $i = 1000$ in $t = 100$.

Z grafa (3.2a) se vidi, da zaporedje algoritma ALS konvergira k stacionarni točki problema NMF, v kateri je vrednost energijske funkcije manjša kot v točki, kjer konča algoritem iALS. Z istega grafa se opazi naraščanje in padanje vrednosti funkcije algoritma iALS. To je posledica projeciranja rešitve klasičnega problema najmanjših kvadratov na prostor nenegativnih matrik.

Z grafa (3.2b) se vidi počasnost algoritma ALS. Medtem ko je algoritem iALS opravil 1000 iteracij v omejenem času, je algoritem ALS v 100 sekundah opravil le 484 iteracij.

V praksi se algoritma ALS ne uporablja zaradi prevelike časovne zahtevnosti. Algoritem iALS uporabljajo v hibridnih algoritmih, kjer ga uporabijo za izračun začetnih aproksimacij matrik U in V.

3.3 Projektivni gradientni spust

Problem nenegativne matrične faktorizacije si lahko predstavljamo kot nelinearni problem na konveksni množici nenegativnih matrik. Projekcija matrike na prostor nenegativnih matrik je preprosta. Pri projekciji vsako negativno vrednost spremenimo v 0. V tem razdelku bomo predstavili *projektivni gradientni spust*. Ta je sestavljen iz treh korakov:

- Izračun gradienta $\nabla F(x_k)$,
- izbor velikosti koraka α_k in
- projekcija posodobitve na prostor nenegativnih vektorjev $\mathbb{R}^n_+$,

$$x_{k+1} = [x_k - \alpha_k \nabla F(x_k)]_+,$$

kjer je x_k trenutni približek iskanega vektorja in $[x]_+$ projekcija vektorja x na prostor nenegativnih vektorjev. Gradient $\nabla F(x_k)$ predstavlja smer, v kateri vrednost funkcije F najhitreje narašča oz. vektor $-\nabla F(x_k)$ predstavlja smer, v kateri vrednost funkcije F najhitreje pada iz točke x_k. S korakom α_k določimo velikost premika iz točke x_k v smeri vektorja $-\nabla F(x_k)$. Premik ne zagotavlja, da bo izračunani vektor ostal v istem vektorskem prostoru. Zato ga moramo projecirati nazaj na prostor nenegativnih vektorjev.

V tem razdelku bomo opisali dva algoritma za reševanje problema NMF. Oba temeljita na metodi projektivnega gradientnega spusta. Razlikujeta se v načinu določevanja velikosti koraka α_k. Več o metodi gradientnega spusta v prostoru nenegativnih vektorjev najdemo v [3].

3.3.1 Gradientni spust z uporabo Armijovega pravila

Pri gradientnemu spustu želimo določiti tak korak α_k, da velja $F(x_{k+1}) \leq F(x_k)$. Pri tem želimo tudi, da je x_{k+1} čim bližje optimalni rešitvi. Določevanje takega koraka opisuje algoritem 5.

Algoritem 5: Gradientni spust z uporabo Armijovega pravila (Line)

Podatki: Funkcija $F\colon \mathbb{R}^n \to \mathbb{R}$
Rezultat: Vektor $x^* \in \mathbb{R}^n_+$, ki minimizira funkcijo F

Inicializiraj vektor x_0 in skalarje $\alpha_0 = 1$, $k = 0$ ter $\beta, \sigma \in (0, 1)$;
ponavljaj
    $y = [x_k - \alpha_k \nabla F(x_k)]_+$;
    če velja $F(y) - F(x_k) > \sigma\langle \nabla F(x_k), y - x_k\rangle$ **potem**
        ponavljaj
            $\alpha_k = \alpha_k \cdot \beta$;
            $y = [x_k - \alpha_k \nabla F(x_k)]_+$;
        dokler $F(y) - F(x_k) \leq \sigma\langle \nabla F(x_k), y - x_k\rangle$;
    sicer
        ponavljaj
            $lasty = y$;
            $\alpha_k = \frac{\alpha_k}{\beta}$;
            $y = [x_k - \alpha_k \nabla F(x_k)]_+$;
        dokler $F(y) - F(x_k) > \sigma\langle \nabla F(x_k), y - x_k\rangle$;
        $y = lasty$;
    $x_{k+1} = y$;
    $\alpha_{k+1} = \alpha_k$;
    $k = k + 1$;
dokler *zaustavitveni pogoj*;

V algoritmu imamo pogoj

$$F(y) - F(x_k) \leq \sigma\langle \nabla F(x_k), y - x_k\rangle,$$

ki mu pravimo *Armijevo pravilo*. Izpolnitev pogoja zagotavlja zadosten spust funkcije F. Vsebuje parameter σ, ki ga določimo na začetku algoritma. S parametrom β povečujemo ali zmanjšujemo korak v smeri gradienta. Izkaže se, da parametra β in σ vplivata na konvergenco. V [15] so pokazali, da je metoda pri ustrezno izbranih parametrih ($\sigma = 0.01$ in $\beta = 0.1$) hitrejša kot metoda multiplikativnega pravila (algoritem 2).

Algoritem lahko direktno uporabimo za reševanje problema NMF, če izračunamo trenutni približek

$$x_k = \begin{bmatrix} U_k \\ V_k \end{bmatrix} \in \mathbb{R}^{(m+n)\times r}$$

in njegov gradient

$$\nabla F = \begin{bmatrix} \nabla F_U(U_k, V_k) \\ \nabla F_V(U_k, V_k) \end{bmatrix} = \begin{bmatrix} U_k V_k^T V_k - AV_k \\ V_k U_k^T U_k - A^T U_k \end{bmatrix}.$$

Ker velja

$$\begin{aligned} [x_k - \alpha_k \nabla F(x_k)]_+ &= \left[\begin{bmatrix} U_k \\ V_k \end{bmatrix} - \alpha_k \begin{bmatrix} \nabla F_U(U_k, V_k) \\ \nabla F_V(U_k, V_k) \end{bmatrix} \right]_+ \\ &= \begin{bmatrix} [U_k - \alpha_k \nabla F_U(U_k, V_k)]_+ \\ [V_k - \alpha_k \nabla F_V(U_k, V_k)]_+ \end{bmatrix}, \end{aligned}$$

lahko matriki U_k in V_k posodabljamo ločeno. O načinu posodabljanja bomo govorili v razdelku 3.3.3.

3.3.2 Gradientni spust z uporabo aproksimacije prvega reda

Pri iskanju rešitve problema

$$x^* = \underset{x}{\operatorname{argmin}}\, F(x)$$

lahko funkcijo F pri vsaki iteraciji aproksimiramo s funkcijo

$$\widehat{F}_k(x) = F(x_k) + \langle \nabla F(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|_2^2,$$

kjer je L najmanjše tako število, da velja $F(x) \leq \widehat{F}_k(x)$ za vsak vektor x. Tako število obstaja, ker je funkcija F zvezno odvedljiva in gradient ∇F je Lipschitzov [18, lema 3.3.1].

Za vsako točko

$$x_{k+1} = \underset{x}{\operatorname{argmin}}\, \widehat{F}_k(x)$$

velja $F(x_{k+1}) \leq \widehat{F}_k(x_{k+1}) \leq \widehat{F}_k(x_k) = F(x_k)$. To pomeni, da je vrednost funkcije F v točki x_{k+1} manjša kot v točki x_k. Ker vrednosti L ne poznamo, jo moramo pri vsaki iteraciji izračunati. Postopek je opisan v algoritmu 6.

Glavna razlika med algoritmom 5 in algoritmom 6 je v zaustavitvenem kriteriju notranje zanke. Podobno kot pri algoritmu 5 lahko matriki U_k in V_k posodabljamo ločeno. Več bomo obravnavali v naslednjem razdelku.

Algoritem 6: Gradientni spust z uporabo aproksimacije prvega reda (FO)

Podatki: Funkcija $F\colon \mathbb{R}^n \to \mathbb{R}$
Rezultat: Vektor $x^* \in \mathbb{R}^n_+$, ki minimizira funkcijo F

Inicializiraj x_0, L_0, $\beta > 1$ in $k = 0$;
ponavljaj
    $y = [x_k - \frac{1}{L_k}\nabla F(x_k)]_+$;
    dokler velja $F(y) - F(x_k) > \langle \nabla F(x_k), y - x_k \rangle + \frac{L_k}{2}\|y - x_k\|_2^2$ **naredi**
        $L_k = L_k \cdot \beta$;
        $y = [x_k - \frac{1}{L_k}\nabla F(x_k)]_+$;
    $x_{k+1} = y$;
    $L_{k+1} = \frac{L_k}{\beta}$;
    $k = k + 1$;
dokler *zaustavitveni pogoj*;

3.3.3 Implementacija gradientnega spusta

Najzahtevnejši korak pri opisanih algoritmih gradientnega spusta je iskanje ustreznega koraka α_k. V algoritmu 6 vrednost α_k nadomesti vrednost $\frac{1}{L_k}$. Kot smo že prej omenili, ločimo dva načina posodabljanja matrik:

- Pri *iskanje po celotnem prostoru* matriki U in V posodabljamo hkrati. Za točen izračun energijske funkcije $F(x_k) = F(U_k, V_k) = \|A - U_k V_k^T\|_F^2$ in $F(y)$ moramo opraviti $\mathcal{O}(mnr)$ operacij. Torej moramo opraviti $\mathcal{O}(tmnr)$ operacij za izračun koraka α_k, kjer je t število notranjih iteracij.

- Pri *koordinatnem iskanju* matriki U in V posodabljamo ločeno. Ko je matrika V fiksirana, je energijska funkcija F kvadratna funkcija v spremenljivki U:

$$\begin{aligned} F(U) = \|A - UV^T\|_F^2 &= \langle A, A \rangle - 2\langle UV^T, A \rangle + \langle UV^T, UV^T \rangle \\ &= \|A\|_F^2 - 2\langle U, AV \rangle + \langle U, U(V^T V) \rangle. \end{aligned}$$

  Najzahtevnejši korak je izračun produkta AV, za katerega moramo opraviti $\mathcal{O}(mnr)$ operacij. Ker je matrika V fiksirana, ga lahko izračunamo pred notranjo zanko. Za izračun $\langle U, AV \rangle$ moramo opraviti $\mathcal{O}(mr)$ operacij. Izračunati moramo še vrednost $\langle U, U(V^T V) \rangle$. Ker je matrika V fiksirana, lahko matriko $V^T V$ izračunamo pred notranjo zanko. Zanj moramo opraviti $\mathcal{O}(nr^2)$ operacij. Potem moramo za izračun $\langle U, U(V^T V) \rangle$ opraviti $\mathcal{O}(mr^2 + mr)$ operacij. Skupaj opravimo $\mathcal{O}(t_1 mr^2)$ operacij za izračun ustreznega koraka α_{U_k}, kjer je t_1 število iteracij v notranji zanki. Ko fiksiramo matriko U, moramo za izračun koraka α_{V_k} opraviti $\mathcal{O}(t_2 nr^2)$ operacij, kjer je

t_2 število iteracij v notranji zanki. Za izračun korakov moramo opraviti $\mathcal{O}(t_1 mr^2 + t_2 nr^2)$ operacij. To je manj kot pri iskanju po celotnem prostoru, saj za reduciran rang r običajno izberemo $r \ll \min\{m, n\}$.

3.4 Zaustavitveni pogoj

V tem razdelku se bomo posvetili zaustavitvenim pogojem, ki jih lahko uporabimo v zgoraj omenjenih algoritmih.

Prvi, ki ga bomo omenili, je *merjenje spusta energijske funkcije*. Pri tem zaustavitvenim pogojem preverjamo, ali se je vrednost energijske funkcije s posodobitvijo matrik U in V zadosti zmanjšala. To lahko preverimo z enačbo

$$|F(U_{k+1}, V_{k+1}) - F(U_k, V_k)| < \epsilon$$

ali z enačbo

$$\frac{|F(U_{k+1}, V_{k+1}) - F(U_k, V_k)|}{F(U_k, V_k)} < \epsilon,$$

kjer sta U_k in V_k matriki trenutnega stanja, matriki U_{k+1} in V_{k+1} pa posodobljenega stanja. Pogoj ni zanesljiv, saj lahko algoritem prekine še preden konvergira k stacionarni točki.

Boljši pogoj je *pogoj z normo projektivnega gradienta*. Predstavili so ga v [15]. Naj bosta $\nabla^P F_U$ in $\nabla^P F_V$ matriki, za kateri velja

$$(\nabla^P F_X)_{ij} = \begin{cases} (\nabla F_X)_{ij}, & \text{če velja } X_{ij} > 0, \\ \min\{0, (\nabla F_X)_{ij}\}, & \text{če velja } X_{ij} = 0, \end{cases}$$

kjer je X bodisi matrika U bodisi matrika V. Pogoj z normo projektivenga gradienta se glasi

$$\left\| \begin{bmatrix} \nabla^P F_{U_k} \\ \nabla^P F_{V_k} \end{bmatrix} \right\|_F \leq \epsilon \left\| \begin{bmatrix} \nabla F_{U_0} \\ \nabla F_{V_0} \end{bmatrix} \right\|_F . \tag{3.6}$$

Konvergenco k stacionarni točki lahko preverimo tako, da izračunamo gradient v izračunani točki. Če je gradient blizu nič, potem je izračunana točka blizu stacionarne točke in algoritem je konvergiral. Z enačbo (3.6) preverjamo, ali je gradient v točki (U_k, V_k) blizu nič.

Pogoj je občutljiv na skaliranje matrik U in V. Naj bosta $\widehat{U} = \gamma U$ in $\widehat{V} = \frac{1}{\gamma} V$ matriki, kjer je $\gamma > 0$ poljuben skalar. Potem velja $\widehat{U}\widehat{V}^T = UV^T$. Množenje s številom γ pa spremeni vrednost projektivnega gradienta.

$$\begin{aligned}
\left\| \begin{bmatrix} \nabla^P F_{\widehat{U}} \\ \nabla^P F_{\widehat{V}} \end{bmatrix} \right\|_F^2 &= \|\nabla^P F_{\widehat{U}}\|_F^2 + \|\nabla^P F_{\widehat{V}}\|_F^2 \\
&= \frac{1}{\gamma^2}\|\nabla^P F_U\|_F^2 + \gamma^2\|\nabla^P F_V\|_F^2 \qquad (3.7) \\
&\neq \left\| \begin{bmatrix} \nabla^P F_U \\ \nabla^P F_V \end{bmatrix} \right\|_F^2
\end{aligned}$$

Za ekvivalentni rešitvi NMF (U, V) in $(\widehat{U}, \widehat{V})$ pričakujemo, da bosta imeli enako vrednost projektivnega gradienta.

Izberemo vrednost $\gamma^2 = \frac{\|\nabla^P F_U\|_F}{\|\nabla^P F_V\|_F}$, ki minimizira enačbo (3.7). Pri tej vrednosti γ velja $\|\nabla^P F_{\widehat{U}}\|_F = \|\nabla^P F_{\widehat{V}}\|_F$. S tem zagotovimo, da se vrednosti projektivnih gradientov ne razlikujejo. Vendar to ni dobra izbira, ko je ena izmed vrednosti $\|\nabla^P F_{\widehat{U}}\|_F$ ali $\|\nabla^P F_{\widehat{V}}\|_F$ blizu nič.

Ker je gradient ∇F občutljiv na skaliranje matrik U in V, je vsak zaustavitveni pogoj, ki vsebuje gradient, občutljiv na skaliranje. Občutljivost zmanjšamo tako, da pri vsaki iteraciji skaliramo matriki U_k in V_k,

$$U_k = U_k D_k, \qquad V_k = V_k D_k^{-1},$$

kjer je D_k pozitivna diagonalna matrika z diagonalnimi elementi

$$(D_k)_{ii} = \sqrt{\frac{\|(V_k)_{:i}\|_2}{\|(U_k)_{:i}\|_2}}.$$

S tem zmanjšamo razliko med $\|\nabla^P F_U\|_F$ in $\|\nabla^P F_V\|_F$. Izognemo se tudi morebitni numerični nestabilnosti.

Algoritem lahko ustavimo tudi po vnaprej določenem številu iteracij ali pa določimo časovno omejitev. S tem zagotovimo, da se algoritem ustavi. Zaželjeno je, da se take omejitve določi pri vseh algoritmih, še posebej pri tistih, ki imajo visoko časovno zahtevnost.

3.5 Inicializacija

Pri vseh omenjenih algoritmih moramo na prvem koraku inicializirati matriki U_0 in V_0. Izkaže se, da je inicializacija pomemben korak, saj ta določa hitrost konvergence in h kateri stacionarni točki bo algoritem konvergiral. V tem razdelku bomo opisali nekaj inicializacijskih metod.

3.5.1 Inicializacija z naključno matriko

Najpogostejša inicializacijska metoda je *inicializacija z naključno matriko*. Pri tej metodi se matriki U_0 in V_0 naključno generira. Metoda je preprosta in hitro izvedljiva. Slabost te metode je, da lahko algoritem vsakič vrne drug rezultat. Zato je algoritem potrebno večkrat izvesti in izbrati najboljšo rešitev.

Zaradi naivnosti metode se matriki U_0 in V_0 pred uporabo malo modificira. Množi se ju s skalarjem $\sqrt{\alpha}$,

$$\alpha = \frac{\langle A, U_0 V_0^T \rangle}{\langle U_0 V_0^T, U_0 V_0^T \rangle}, \qquad U_0 = \sqrt{\alpha} U_0, \qquad V_0 = \sqrt{\alpha} V_0,$$

kjer je α optimalna rešitev problema

$$\min_{\alpha} \| A - \alpha U_0 V_0^T \|_F^2.$$

S tem začetno aproksimacijo $U_0 V_0^T$ približamo končni vrednosti.

3.5.2 Razvrščanje z voditelji

Razvrščanje z voditelji (angl. K-means clustering) je metoda za klasificiranje vektorjev v skupine. Naj bo $X = \{x_1, \ldots, x_n\}$ množica vektorjev, kjer je $x_i \in \mathbb{R}^m$, in $r \leq n$ naravno število. Metoda vrne množico centroidov $\{u_i\}_{i=1}^r$, $u_i \in \mathbb{R}^m$, ki so predstavniki disjunktnih množic $\{\pi_i\}_{i=1}^r$, za katere velja

$$\bigcup_{i=1}^{r} \pi_i = X.$$

Centroid u_i je aritmetična sredina vektorjev znotraj množice π_i, tj.

$$u_i = \frac{1}{|\pi_i|} \sum_{x \in \pi_i} x.$$

Postopek razvrščanja z voditelji običajno poteka v treh korakih:

- Določimo metriko in razdaljo, ki jo bomo uporabili pri izračunih,
- Vektor x_i dodelimo množici π_j, če je vektor u_j najbližji centroid vektorju x_i pri izbrani razdalji,
- vektor u_j nadomestimo z aritmetično sredino vektorjev $x \in \pi_j$.

Če imamo matriko $X = [\; x_1 \quad x_2 \quad \cdots \quad x_n \;]$, potem lahko sestavimo matriki $U = [\; u_1 \quad u_2 \quad \cdots \quad u_r \;]$ in V, za katero velja

$$V_{ij} = \begin{cases} 1, & \text{če velja } x_i \in \pi_j, \\ 0, & \text{sicer.} \end{cases}$$

Ko je matrika X nenegativna, je tudi matrika U nenegativna. To sledi iz nenegativnosti vektorjev x_i in iz konstrukcije vektorjev u_j. Za inicializacijo algoritmov NMF lahko vzamemo matriki U in V generirani z razvrščanjem z voditelji, kjer je število centroidov enako reduciranemu rangu r.

Inicializacijo lahko izboljšamo, če poiščemo rešitev problema

$$\min_{Z \in \mathbb{R}_+^n} \|X - UZ^T\|_F^2.$$

Težave te metode nastopajo pri inicializaciji centroidov. Inicializiramo jih lahko na več načinov:

- Začetne centroide naključno generiramo. Z uporabo te metode lahko algoritem vsakič vrne različne centroide. Lahko se tudi zgodi, da so nekateri centroidi prazni.

- Centroide inicializiramo z naključnim izborom r stolpcev matrike X. To je boljša inicializacija, vendar ima enake težave kot pri naključnem generiranju centroidov.

Podobna inicializacijska metoda je z uporabo *sferičnega razvrščanja z voditelji*. Ta se razlikuje od navadnega razvrščanja v tem, da so centroidi normalizirani in da se razdaljo dveh vektorjev meri s kosinusno razdaljo, $\cos(x, y) = \frac{x^T y}{\|x\|_2 \|y\|_2}$, kjer sta $x, y \in \mathbb{R}^m$. Več je napisano v [24].

3.5.3 Singularni razcep

V [4] so predstavili algoritem, ki matriki U_0 in V_0 inicializira z uporabo singularnega razcepa. Ta je sestavljen iz treh korakov:

1. Izračun singularnega razcepa $A = Q\Sigma R^T$.

2. Sestavljanje matrik $\{C_i\}_{i=1}^r$, kjer je r reduciran rang in $C_i = Q_{:i} R_{i:}$ matrika sestavljena iz singularnih vektorjev, ki pripadajo i-ti singularni vrednosti.

3. Izražava singularnih vektorjev in nenegativnih delov matrik $\{C_i\}_{i=1}^r$ ter njihova uporaba v konstrukciji matrik U_0 in V_0.

Algoritem 7: Inicializacija matrik U_0 in V_0 s singularnim razcepom

Podatki: Matrika $A \in \mathbb{R}_+^{m\times n}$ in naravno število $r < \min\{m, n\}$
Rezultat: Nenegativni matriki $U_0 \in \mathbb{R}_+^{m\times r}$ in $V_0 \in \mathbb{R}_+^{n\times r}$

Izračunaj singularni razcep matrike $A = Q\Sigma R^T$;
Postavi $U_{:1} = \sqrt{\Sigma_{11}}Q_{:1}$ in $V_{:1} = \sqrt{\Sigma_{11}}R_{:1}$;
za i = 2, . . . , r **naredi**
    Vzemi nova stolpca $x = Q_{:i}$ in $y = R_{:i}$;
    Vsak stolpec projeciraj na prostor nepozitivnih in nenegativnih vektorjev: $x_P = [x]_+$, $x_N = [x]_-$, $y_P = [y]_+$, $y_N = [y]_-$;
    Izračunaj produkt norm pozitivnih projekcij: $\mathrm{Norm}_P = \|x_P\|_2\|y_P\|_2$;
    Izračunaj produkt norm negativnih projekcij: $\mathrm{Norm}_N = \|x_N\|_2\|y_N\|_2$;
    če velja *poz* > *neg* **potem**
        Definiraj $u = \frac{x_P}{\|x_P\|_2}$, $v = \frac{y_P}{\|y_P\|_2}$ in $\sigma = \mathrm{Norm}_P$;
    sicer
        Definiraj $u = \frac{x_N}{\|x_N\|_2}$, $v = \frac{y_N}{\|y_N\|_2}$ in $\sigma = \mathrm{Norm}_N$;
    Postavi $U_{:j} = \sqrt{\sigma\Sigma_{jj}} \cdot u$ in $V_{:j} = \sqrt{\sigma\Sigma_{jj}} \cdot v$;

Spomnimo se, da smo z $[x]_+$ označili projekcijo vektorja x na prostor nenegativnih vektorjev in z $[x]_-$ projekcijo na prostor nepozitivnih vektorjev. Potek inicializacije je opisan v algoritmu 7.

Inicializacijsko metodo se lahko uporabi v vseh algoritmih za reševanje problema NMF. Ne uporablja nobenih naključno izbranih vrednosti, zato algoritmi vedno konvergirajo k isti točki.

Algoritem ima veliko pozitivnih lastnosti, vendar je počasen. Za izračun singularnega razcepa moramo za matriko $A \in \mathbb{R}^{m\times n}$ opraviti $O(mn^2)$ operacij. Algoritem lahko pospešimo, saj potrebujemo samo prvih r singularnih vrednosti in vektorjev.

Implementacija algoritma je zahtevna, zato se algoritma običajno ne uporablja za inicializacijo matrik.

3.6 Primerjava algoritmov

V tem razdelku bomo naštete algoritme primerjali. Za boljšo preglednost bomo vsak algoritem označili s kratico:

- multiplikativno pravilo (**Mult**),
- alternirajoči najmanjši kvadrati, ki uporabijo Matlabovo funkcijo *lsqnonneg* (**ALS**),

- gradientni spust z uporabo Armijovega pravila in iskanjem po celotnem prostoru (**FLine**),
- gradientni spust z uporabo Armijovega pravila in koordinatnem iskanju (**CLine**),
- gradientni spust z uporabo aproksimacije prvga reda in iskanjem po celotnem prostoru (**FFO**),
- gradientni spust z uporabo aproksimacije prvega reda in koordinatnem iskanju (**CFO**).

Netočne metode alternirajočih najmanjših kvadratov ne bomo vključili v analizo.

Za različne parametre m, n in r bomo generirali 100 naključnih nenegativnih matrik, kjer so elementi matrik enakomerno porazdeljeni na intervalu $(0, 1)$. Na matrikah bomo uporabili zgoraj naštete algoritme in merili število uspelih konvergenc.

Vse metode bodo uporabile enaki začetni matriki U_0 in V_0, ki bosta naključno generirani. Matriki skaliramo na način, ki smo ga opisali v razdelku 3.5.1:

$$U_0 = U_0 D\sqrt{\alpha}, \qquad V_0 = V_0 D^{-1}\sqrt{\alpha},$$

kjer sta

$$\alpha = \frac{\langle A, UV^T\rangle}{\langle UV^T, UV^T\rangle}, \quad D_{ij} = \begin{cases} \sqrt{\frac{\|(V_0)_{:i}\|_2}{\|(U_0)_{:i}\|_2}}, & i = j, \\ 0, & i \neq j. \end{cases}$$

Algoritem ustavimo, ko je izpolnjen pogoj norme projektivnega gradienta (3.6) ali ko preteče 60 sekund. Števila iteracij ne bomo omejili. Konvergenca je uspešna, če je pogoj norme projektivnega gradienta izpolnjen znotraj omejenega časa. Sicer pravimo, da konvergenca ni uspela. Zaustavitveni pogoj bomo merili pri relativni natančnosti ϵ: 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} in 10^{-6}.

Tabela 3.1 vsebuje merjenja pri različnih parametrih m, n, r in ϵ. Iz tabele opazimo naslednje:

Najpogosteje uporabljen algoritem **Mult** že pri manjših matrikah ne konvergira znotraj omejenega časa. To je lahko posledica zaokrožitvenih napak (razdelek 3.1).

Algoritem **ALS** je počasen. Zaradi tega algoritem pri večjih matrikah potrebuje več časa da pride do stacionarne točke (razdelek 3.2).

Najboljše rezultate vrača algoritem **CFO**. V splošnem imata algoritma gradientnega spusta, ki uporabita koordinatno iskanje, boljše rezultate kot tista, ki iščeta po celotnem prostoru. To smo tudi pokazali, saj je v splošnem iskanje po celotnem prostoru bolj časovno zahtevno kot koordinatno iskanje (razdelek 3.3.3).

Tabela 3.1: Primerjava algoritmov NMF: Časovna omejitev je 60 sekund. Tabele prikazujejo povprečen čas uspešnih konvergenc na 100 naključno generiranih matrikah. Zapis 1.812 (67) pomeni, da je algoritem v 67 (od 100) primerih uspešno konvergiral znotraj časovne omejitve pri relativni natančnosti ϵ. Povprečen čas uspešnih konvergenc je 1.812 sekund. Zapis 60* (0) pomeni, da algoritem za nobeno matriko ni konvergiral znotraj časovne omejitve.

	Mult	**ALS**	**FLine**	**CLine**	**FFO**	**CFO**
$(m = 10, n = 5, r = 2)$						
$\epsilon = 10^{-2}$	0.006 (81)	0.026	0.004	0.005	0.003	0.002
$\epsilon = 10^{-3}$	3.112 (58)	0.057	0.008	0.009	0.004	0.005
$\epsilon = 10^{-4}$	1.812 (67)	0.085	0.011	0.015	0.006	0.008
$\epsilon = 10^{-5}$	0.038 (60)	0.117	0.016	0.019	0.009	0.010
$\epsilon = 10^{-6}$	2.038 (60)	0.129	0.018	0.022	0.010	0.013
$(m = 20, n = 10, r = 3)$						
$\epsilon = 10^{-2}$	10.94 (11)	0.076	0.004	0.006	0.005	0.006
$\epsilon = 10^{-3}$	0.109 (3)	0.292	0.014	0.028	0.022	0.018
$\epsilon = 10^{-4}$	30.20 (4)	0.615	0.031	0.068	0.047	0.038
$\epsilon = 10^{-5}$	36.03 (5)	1.055	0.042	0.111	0.066	0.060
$\epsilon = 10^{-6}$	60* (0)	1.263	0.067	0.178	0.103	0.087

Tabela 3.1 (nadalj.): Primerjava algoritmov

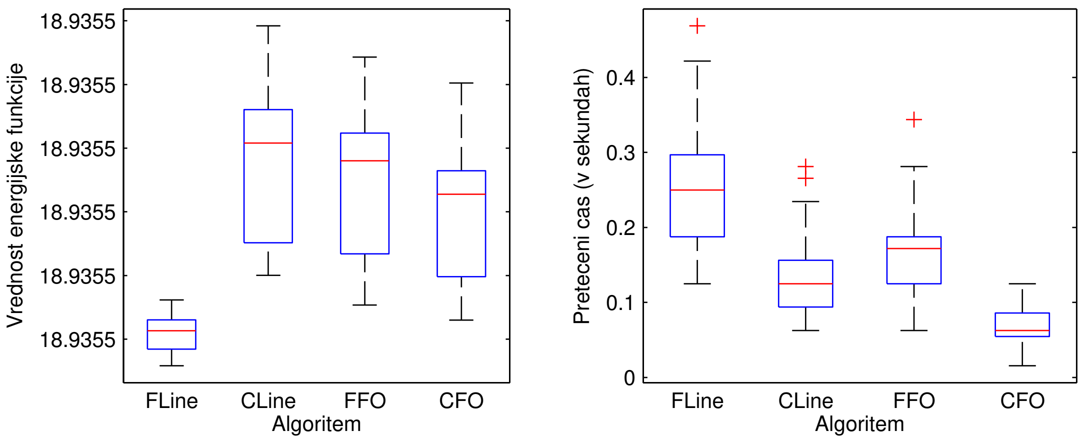
	Mult	**ALS**	**FLine**	**CLine**	**FFO**	**CFO**
$(m = 50, n = 20, r = 5)$						
$\epsilon = 10^{-2}$	59.98 (1)	0.507	0.089	0.030	0.029	0.017
$\epsilon = 10^{-3}$	60* (0)	1.746	0.460	0.193	0.154	0.114
$\epsilon = 10^{-4}$	59.98 (1)	3.309	0.775	0.414	0.352	0.249
$\epsilon = 10^{-5}$	60* (0)	4.816	1.003	0.609	0.473	0.380
$\epsilon = 10^{-6}$	60* (0)	7.766	1.285	0.750	0.644	0.517
$(m = 100, n = 50, r = 10)$						
$\epsilon = 10^{-2}$	60* (0)	2.668	0.219	0.252	0.149	0.083
$\epsilon = 10^{-3}$	60* (0)	18.81	1.641	1.724	1.245	0.616
$\epsilon = 10^{-4}$	60* (0)	40.90 (68)	4.976	4.910	4.121	2.160
$\epsilon = 10^{-5}$	60* (0)	48.95 (24)	8.481	7.548	6.980	3.459
$\epsilon = 10^{-6}$	60* (0)	51.59 (9)	12.27	11.13	9.661	4.951

Tabela 3.1 (nadalj.): Primerjava algoritmov

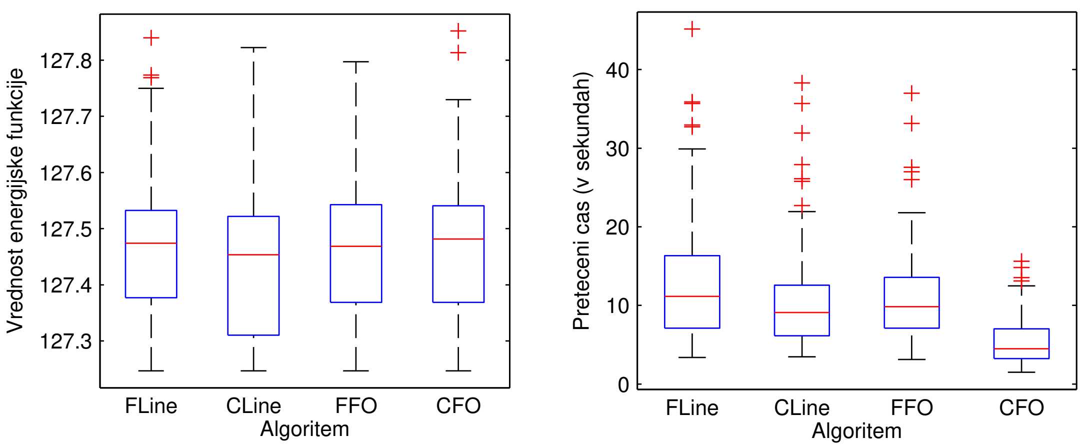
	Mult	**ALS**	**FLine**	**CLine**	**FFO**	**CFO**
($m = 200, n = 100, r = 20$)						
$\epsilon = 10^{-2}$	60* (0)	12.00	0.567	0.479	1.547	0.370
$\epsilon = 10^{-3}$	60* (0)	60* (0)	6.105	4.709	20.19	4.066
$\epsilon = 10^{-4}$	60* (0)	60* (0)	25.19	18.96	49.91 (27)	17.09
$\epsilon = 10^{-5}$	60* (0)	60* (0)	37.53 (77)	30.29 (89)	53.68 (4)	28.15 (90)
$\epsilon = 10^{-6}$	60* (0)	60* (0)	42.35 (52)	38.22 (73)	59.97 (1)	37.01 (74)
($m = 200, n = 100, r = 30$)						
$\epsilon = 10^{-2}$	60* (0)	28.09	1.409	1.239	0.607	0.865
$\epsilon = 10^{-3}$	60* (0)	60* (0)	15.41	13.38	25.06 (99)	10.05
$\epsilon = 10^{-4}$	60* (0)	60* (0)	41.47 (71)	40.05 (80)	47.01 (27)	29.45 (93)
$\epsilon = 10^{-5}$	60* (0)	60* (0)	48.46 (20)	44.07 (34)	52.53 (1)	40.33 (53)
$\epsilon = 10^{-6}$	60* (0)	60* (0)	50.16 (15)	50.80 (17)	60* (0)	46.05 (31)

Algoritme **FLine**, **CLine**, **FFO** in **CFO** si bomo podrobneje ogledali. Za različne parametre m, n in r bomo fiksirali matriko $A \in \mathbb{R}^{m \times n}$ in generirali 100 naključnih začetnih točk (U_0, V_0). Vsak algoritem bomo stokrat zagnali na matriki A, kjer bomo vsakič podali drugo začetno točko.

Merili bomo vrednost energijske funkcije ob konvergenci in pretečen čas v sekundah. Za prikaz rezultatov bomo uporabili prikaz *škatlo z brki*, ki prikazuje razpršenost podatkov. [1] Grafi 3.3, 3.4, 3.5 in 3.6 prikazujejo rezultate meritev.

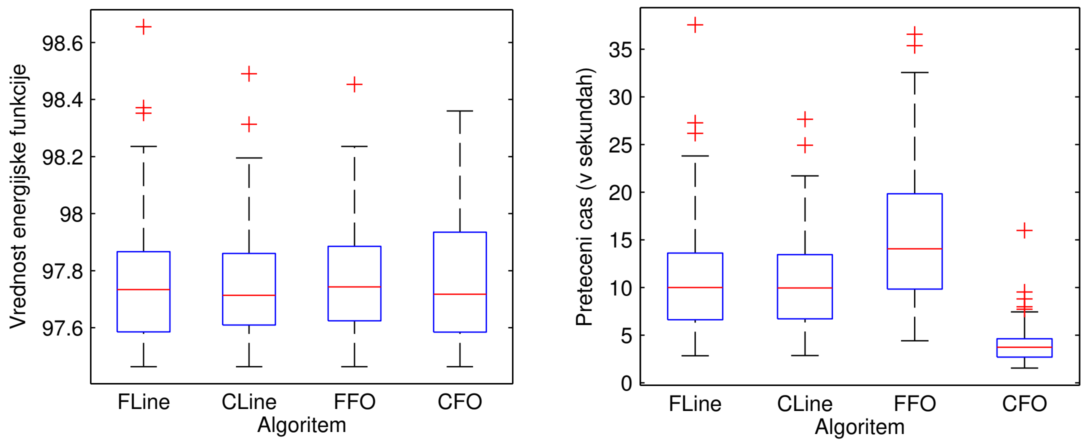


Slika 3.3: Primerjava algoritmov **FLine**, **CLine**, **FFO** in **CFO** pri parametrih $m = 30$, $n = 20$, $r = 2$ in $\epsilon = 10^{-6}$.

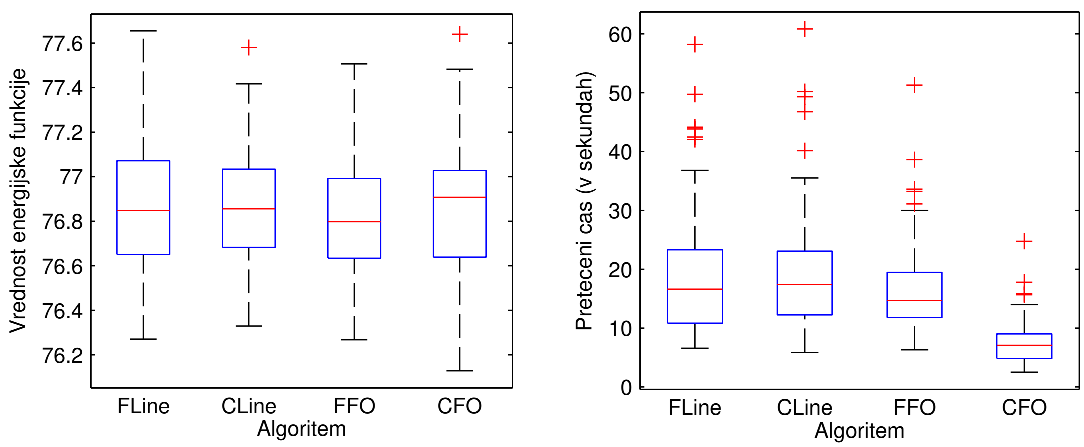


Slika 3.4: Primerjava algoritmov **FLine**, **CLine**, **FFO** in **CFO** pri parametrih $m = 100$, $n = 50$, $r = 10$ in $\epsilon = 10^{-6}$.

[1] Branje prikaza škatle z brki: rdeča črta predstavlja mediano. Nad njo se nahaja zgornjih 50%, pod njo pa spodnjih 50% meritev. Spodnji brki predstavljajo najmanjših 25% meritev, medtem ko zgornji brki predstavljajo največjih 25% meritev. Škatla predstavlja meritve med spodnjim in zgornjim kvartilom.



Slika 3.5: Primerjava algoritmov **FLine**, **CLine**, **FFO** in **CFO** pri parametrih $m = 100$, $n = 50$, $r = 15$ in $\epsilon = 10^{-4}$.



Slika 3.6: Primerjava algoritmov **FLine**, **CLine**, **FFO** in **CFO** pri parametrih $m = 100$, $n = 50$, $r = 20$ in $\epsilon = 10^{-4}$.

Iz grafov opazimo, da se vrednosti energijskih funkcij med algoritmi ne razlikujejo veliko. Vidimo tudi, da algoritem **CFO** najhitreje konvergira, za njim je najhitrejši algoritem **CLine**. Spet smo pokazali, da sta algoritma gradientnega spusta, ki uporabita koordinatno iskanje, hitrejša od tistih, ki iščeta po celotnem prostoru.

Algoritmi so napisani v programu Matlab R2014a. Primerjave so se izvajale na računalniku z Intel Core i7-5600 CPU 2.6 GHz procesorjem in 12GB RAM.

Poglavje 4

Primer uporabe

Nenegativna matrična faktorizacija ima veliko aplikacij na področju strojnega učenja. Uporabljajo jo pri tekstovnem rudarjenju, prepoznavanju značilnosti obraza (angl. face pattern recognition), priporočevalnih sistemih itd. V naštetih aplikacijah so vrednosti, s katerimi ravnamo, običajno nenegativne (frekvenca besede v dokumentu, odtenek sivine na sliki, ocena predmeta, ki mu jo je dal uporabnik). Pri vsaki aplikaciji ima NMF tudi svojo interpretacijo, ki je običajno enostavno razumljiva.

V praksi se za enake izračune uporabljajo tudi druge metode, na primer singularni razcep in razvrščanje z voditelji. Pri teh se lahko zgodi, da rezultat vsebuje negativne vrednosti, ki jih je težko interpretirati.

V tem razdelku si bomo ogledali dva primera uporabe NMF. V prvem primeru bomo črpali teme iz nabora tekstovnih dokumentov. V drugem bomo napovedovali ocene filmov, ki jih uporabniki še niso dali.

4.1 Tekstovno rudarjenje

Tekstovno rudarjenje (angl. text mining) je proces pridobivanja kakovostnih informacij iz besedila. Informacije običajno pridobimo z odkrivanjem vzorcev v besedilih. Te odkrijemo z uporabo različnih statističnih metod.

Proces se ponavadi začne s strukturiranjem vhodnih podatkov iz danega besedila. Vhodni podatki so običajno v vektorski ali matrični obliki. Temu sledi obdelava podatkov z uporabo različnih metod, ki se konča z neko interpretacijo izhodnih podatkov.

V tem razdelku bomo opisali, kako uporabimo NMF za črpanje tem iz besedil. Denimo, da imamo n tekstovnih dokumentov $D_1, D_2, \ldots, D_n$. Iz njih želimo dobiti r glavnih tem, ki nastopajo v dokumentih. Pri tem je $r < n$.

Dokumente moramo najprej pretvoriti v vektorsko ali matrično obliko. To naredimo z modelom *vreča besed* [1]. V tem modelu dokument D_i predstavimo kot večkratno množico svojih besed, kjer ne upoštevamo pravopisja in vrstnega reda besed. Iz večkratnih množic nato generiramo vektorje $\{d_i\}_{i=1}^n$, ki predstavljajo tekstovne dokumente.

Primer 4.1.1. Denimo, da imamo dva tekstovna dokumenta. Dokument D_1 vsebuje besedilo "Lucija je v gledališki skupini. Domen je tudi v gledališki skupini." Dokument D_2 vsebuje besedilo "Domen rad je bonbone." Skupaj vsebujeta 9 različnih besed.

Uporabimo model vreča besed na posameznem dokumentu. Besedam iz dokumentov določimo vrstni red, ki ga fiksiramo. Potem generiramo vektorja d_1 in d_2, ki predstavljata tekstovna dokumenta:

$$B = \begin{bmatrix} \text{Lucija} \\ \text{je} \\ \text{v} \\ \text{gledališki} \\ \text{skupini} \\ \text{Domen} \\ \text{tudi} \\ \text{rad} \\ \text{bonbone} \end{bmatrix}, \quad d_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad d_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

kjer je $(d_i)_k$ število pojavitev besede B_k v dokumentu D_i.

V zgornjem primeru smo opisali preprost način predstavitve dokumentov v vektorski obliki. Lahko ga tudi izboljšamo. Najprej iz vreče besed izločimo besede, ki nimajo pomena, npr. mašila in veznike. Vrednost $(d_i)_k$ nato izračunamo v dveh korakih.

Najprej izračunamo *frekvenco besede* TF_{ki} tj. število pojavitev besede B_k v dokumentu D_i. To smo naredili že v zgornjem primeru.

Nato izračunamo *invertirano frekvenco dokumentov* IDF_k. Ta nam pove, v kolikih dokumentih se beseda B_k pojavi in ji dodeli ustrezno utež. V večih dokumentih je vsebovana, manjša je njena utež. Uteži se najpogosteje izračuna z enačbo $\mathrm{IDF}_k = \log\left(\frac{n}{\mathrm{df}_k}\right)$, kjer je df_k število dokumentov, ki vsebuje besedo B_k.

Posamezno vrednost vektorja d_i se izračuna z enačbo $(d_i)_k = \mathrm{TF}_{ki} \cdot \mathrm{IDF}_k$. S tem izračunamo, koliko pomembna je beseda B_k za dokument D_i.

Dokumente lahko predstavimo z matriko $D = [\ d_1 \quad d_2 \quad \cdots \quad d_n\ ]$. Na matriki D uporabimo nenegativno matrično faktorizacijo z reduciranim rangom r in dobimo matriki $U \in \mathbb{R}_+^{m \times r}$ in $V \in R_+^{n \times r}$, za kateri velja

$$d_i \approx \sum_{j=1}^{r} V_{ij} U_{:j}.$$

Matriko U lahko interpretiramo kot matriko, ki predstavlja množico tem $\{T_i\}_{i=1}^r$. Pri tem je $U_{:j}$ vektor, ki predstavlja temo T_j. Vrednost $(U_{:j})_k$ predstavlja število pojavitev besede B_k v temi T_j. Vrednost V_{ij} je utež, ki meri prisotnost vektorja $U_{:j}$ v vektorju d_i.

4.2 Priporočilni sistemi

Priporočilni sistemi (angl. recommender systems) so podskupina informacijskih sistemov filtriranja, pri čemer poskušajo napovedati oceno (angl. rating), ki jo uporabnik da nekemu predmetu. Sisteme uporabljajo za priporočanje ogleda video posnetkov na internetni strani Youtube [25], priporočanje predmetov na spletnih trgovinah itd.

Denimo, da imamo n uporabnikov $u_1, u_2, \ldots, u_n$, ki so ocenjevali filme f_1, f_2, $\ldots$, f_m. Ponavadi uporabniki ne ocenijo vseh filmov, ker jih še niso gledali ali pa so pozabili podati oceno. Ocena je ponavadi nenegativno število, npr. portala IMDb [11] in Netflix [19] za oceno filma uporabljata število zvezdic. Sestavimo matriko A, kjer vrednost A_{ij} predstavlja oceno, ki jo je uporabnik u_j dal filmu f_i. Matrika je običajno velika in vsebuje malo znanih vrednosti. Naša naloga je napovedati neznane vrednosti matrike A.

Obstaja več pristopov k reševanju tega problema. En pristop je s *filtriranjem, ki temelji na vsebini*. Tukaj uporabimo lastnosti filmov in seznam preferenc uporabnika. Uporabniku priporočamo film, če se lastnosti filma ujemajo s preferencami uporabnika.

Mi bomo uporabili pristop *sodelovalnega filtriranja* (angl. collaborative filtering). Tukaj nabiramo in analiziramo podatke, ki so jih uporabniki dali filmom. Film priporočamo uporabniku u_j, če so uporabniki, ki imajo podobne preference kot uporabnik u_j, dali filmu visoko oceno.

Vrnimo se k problemu. Napovedati želimo neznano vrednost A_{ij}, tj. oceno, ki bi jo uporabnik u_j dal filmu f_i. Vrednost ocenimo z enačbo $A_{ij} = x_i^T \theta_j$, kjer vektor $x_i \in \mathbb{R}_+^r$ predstavlja značilnosti filma f_i in vektor $\theta_j \in \mathbb{R}_+^r$ predstavlja preference uporabnika u_j.

Primer 4.2.1. Denimo, da poznamo lastnosti filma f_i in preference uporabnika u_j.

$$G = \begin{bmatrix} \text{Akcija} \\ \text{Triler} \\ \text{Komedija} \\ \text{Drama} \end{bmatrix}, \quad x_i = \begin{bmatrix} 0.82 \\ 0.10 \\ 0.08 \\ 0.00 \end{bmatrix}, \quad \theta_j = \begin{bmatrix} 4.3 \\ 3.2 \\ 4.5 \\ 2 \end{bmatrix},$$

kjer vrednost $(x_i)_k$ predstavlja prisotnost žanra G_k v filmu f_i in vrednost $(\theta_j)_k$ predstavlja, koliko uporabnik u_j rad gleda filme žanra G_k. Potem lahko napovemo oceno, ki jo uporabnik u_j da filmu f_i.

$$A_{ij} = x_i^T \theta_j = 4.206$$

Vektorjev x_i, θ_j, $i = 1, \ldots, m$, $j = 1, \ldots, n$, ponavadi ne poznamo. Te lahko izračunamo s pomočjo NMF. Naj bo

$$r(i,j) = \begin{cases} 1, & \text{uporabnik } u_j \text{ je ocenil film } f_i, \\ 0, & \text{sicer.} \end{cases}$$

Rešimo problem *obtežene nenegativne matrične faktorizacije*

$$\min_{X \in \mathbb{R}_+^{m \times r}, \theta \in \mathbb{R}_+^{n \times r}} \frac{1}{2} \sum_{(i,j):\, r(i,j)=1} \big(A_{ij} - (X\theta^T)_{ij}\big)^2, \tag{4.1}$$

kjer sta

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1^T \\ \vdots \\ \theta_n^T \end{bmatrix}.$$

Rešitev problema nam izračuna vektorje x_i in θ_j, ki jih lahko uporabimo za napovedovanje vrednosti A_{ij}. Vrednost $r(i,j)$ nam zagotovi, da pri izračunu upoštevamo samo znane vrednosti A_{ij}. Problem 4.1 lahko zapišemo v obliki

$$\min_{X \in \mathbb{R}_+^{m \times r}\, \theta \in \mathbb{R}_+^{n \times r}} F(X,\theta), \quad F(X,\theta) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij} \big(A_{ij} - (X\theta^T)_{ij}\big)^2, \tag{4.2}$$

kjer je W matrika uteži, za katero velja $W_{ij} = r(i,j)$.

Rešimo ga lahko z eno izmed metod projektivnega gradientnega spusta, kjer je gradient enak

$$\nabla F = \begin{bmatrix} \nabla F_X \\ \nabla F_\theta \end{bmatrix} = \begin{bmatrix} (W \circ X\theta^T)\theta - (W \circ A)\theta \\ (W^T \circ \theta X^T)X - (W^T \circ A^T)X \end{bmatrix}.$$

Več o uporabi NMF za priporočilne sisteme je v [7].

Poglavje 5

Zaključek

V delu smo obravnavali problem nenegativne matrične faktorizacije. Zanj smo ugotovili, da nima enolične rešitve. Iskanje optimalne rešitve je težak problem, zato smo običajno zadovoljni, če najdemo lokalni minimimum. Te smo iskali z algoritmi, ki uporabijo neko obliko metode gradientnega spusta. Algoritme smo med seboj primerjali in ugotovili, da je algoritem gradientnega spusta z uporabo aproksimacije prvega reda in koordinatnim iskanjem vračal najboljše rezultate.

Metoda NMF je uporabna zaradi lahke interpetacije izhodnih podatkov. Predstavili smo dva primera uporabe metode, črpanje tem iz besedil in napovedovanje ocen s sodelovalnim filtriranjem.

Trenutno ima metoda nekaj pomankljivosti:

- Algoritmi nimajo podrobne obravnave konvergence. Problemi nastopijo, ko je kakšna vrednost matrik U ali V enaka 0. Takrat lahko algoritem konvergira k točki, ki ni stacionarna.

- Rezultat algoritmov je odvisen od začetnih inicializacij matrik U_0 in V_0. Predstavili smo tri pristope inicializacij matrik. Inicializacija s singularnim razcepom porabi preveč časa, inicializaciji z naključno matriko in z razvrščanjem z voditelji pa generirata naključni matriki U in V, zaradi česar moramo algoritem večkrat zagnati in izbrati tisto rešitev, ki vrne najboljši rezultat. Potrebno je razviti boljšo inicializacijsko metodo.

Problem lahko razširimo tako, da na matriki U in V dodamo dodatne omejitve. V [10] so ločeno obravnavali problem NMF s fiksno vsoto vrstic in stolpcev, dodanimi utežmi na elementih matrik A in UV^T ter obravnavali primer, ko je vhodna matrika A simetrična.

Iščejo se alternativne metode za reševanje problema NMF v polinomskem času. V [10] so razvili algoritem, ki ga imenujejo *alternirajoči NMF*. Ta posodablja stolpce matrik $U = (u_1, \ldots, u_r)$ in $V = (v_1, \ldots, v_r)$ z reševanjem problemov

$$\min_{u \in \mathbb{R}_+^m} \frac{1}{2} \|R_t - u v_t^T\|_F^2, \quad \min_{v \in \mathbb{R}_+^n} \frac{1}{2} \|R_t - u_t v^T\|_F^2,$$

kjer je $R_t = A - \sum_{i \neq t} u_i v_i^T$. Problem je različen od problema NMF, saj matrika R_t ni nujno nenegativna.

Algoritem gradientnega spusta z uporabo aproksimacije prvega reda in koordinatnega iskanja (razdelek 3.3.2) smo implementirali v platformo QMiner [22], ki ga razvijajo na Inštitutu Jožef Stefan. Implementirali smo tudi algoritem za reševanje problema obtežene nenegativne matrične faktorizacije. Uporabili smo ga za napovedovanje ocen, ki bi jih uporabniki dali filmom. Uporabili smo MovieLens bazo podatkov [9], ki vsebuje podatke o 943 uporabnikih in 1682 filmih. Vsak uporabnik je podal vsaj 20 ocen, skupno 100000 ocen. Za izračun učinkovitosti algoritma smo uporabili *normalizirano povprečje absolutnih napak* (NMAE) [17]. Za različne reducirane range smo izračunali vrednost NMAE. Vrednosti so prikazane v tabeli 5. Manjša kot je vrednost NMAE, boljša je napoved ocen. Rezultate smo primerjali s tistimi, ki ga vrne hibridni algoritem iz [7]. Ugotovili smo, da naš algoritem slabše napove ocene kot hibridni algoritem, za katerega je NMAE enak 0.1634, vendar so rezultati zadovoljivi. Primer uporabe NMF za napovedovanje ocen se nahaja v [20].

reduciran rang	**NMAE**
1	0.4285
2	0.3480
5	0.2787
10	0.2531
20	0.2334
50	0.2199
100	0.2146
200	0.2122

Tabela 5.1: Izračun NMAE vrednosti v odvisnosti od reduciranega ranga.

Literatura

[1] *Bag-of-words model*, Wikipedia, [ogled 04.05.2016], dostopno na `https://en.wikipedia.org/wiki/Bag-of-words_model`.

[2] R. B. Bapat, T. E. S. Raghavan, *Nonnegative Matrices and Applications*, Cambridge University Press, New York, 1997.

[3] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, 1999.

[4] C. Boutsidis, E. Gallopoulos, *SVD based initialization: A head start for nonnegative matrix factorization*, Pattern Recogn. **41.4** (2008) 1350–1362.

[5] M. Brojan, J. Globevnik, *Analiza I*, DMFA - založništvo, Ljubljana, 2008.

[6] J. Cohen, U. Rothblum, *Nonnegative ranks, decompositions, and factorizations of nonnegative matrices*, Linear Algebra Appl. **190** (1993) 149–168.

[7] J. Ford, F. Makedon, W. Wang, S. Zhang, *Learning from incomplete ratings using non-negative matrix factorization*, Proceedings of the Sixth SIAM Conference on Data Mining (SDM **6**) (2006) 548–552.

[8] E. F. Gonzalez, Y. Zhang, *Accelerating the Lee-Seung algorithm for non-negative matrix factorization*, Dept. Comput. & Appl. Math., Rice University, 2005.

[9] F. M. Harper, J. A. Konstan, *The MovieLens Datasets: History and Context*, ACM Trans. Interact. Intell. Syst. **5** (2015) Article 19, 19 pages.

[10] N. Ho, *Nonnegative matrix factorization - algorithms and applications*, doktorsko delo, Université Catholique de Louvain, Louvain, 2008.

[11] *IMDb*, [ogled 26.05.2016], dostopno na `http://www.imdb.com/`.

[12] H. Kim, H. Park, *Non-negative matrix factorization based on altering non-negativity constrained least squares and active set method*, SIAM J. Math. Anal. **30.2** (2008) 713-730.

[13] D. D. Lee, H. S. Seung, *Algorithms for non-negative matrix factorization*, Adv. Neural Inf. Process Syst. (2001) 556–562.

[14] D. D. Lee, H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature **401.6755** (1999) 788–791.

[15] C. Lin, *Projected gradient methods for non-negative matrix factorization*, Neural Comput. **19.10** (2007) 2756–2779.

[16] I. Markovsky, *Low-rank Approximation: Algorithms, Implementation, Applications*, Springer Science & Business Media, Springer, London, 2011.

[17] *Mean absolute error*, Wikipedia, [ogled 23.05.2016], dostopno na `https://en.wikipedia.org/wiki/Mean_absolute_error`.

[18] A. Nemirovski, *Optimization II: Numerical methods for nonlinear continuous optimization*, [ogled 04.05.2016], dostopno na `http://www2.isye.gatech.edu/~nemirovs/Lect_OptII.pdf`.

[19] *Netflix*, [ogled 26.05.2016], dostopno na `https://www.netflix.com/`.

[20] E. Novak, *Tonic: Recommender system*, [ogled 04.05.2016], dostopno na `https://tonicdev.com/eriknovak/recommendation-system`.

[21] B. Plestenjak, *Razširjen uvod v numerične metode*, DMFA - založništvo, Ljubljana, 2015.

[22] *QMiner*, [ogled 04.05.2016] dostopno na `http://qminer.ijs.si/`.

[23] S. A. Vavasis, *On the complexity of nonnegative matrix factorization*, SIAM J. Optimiz. **20.3** (2009) 1364–1377.

[24] S. Wild, *Seeding non-negative matrix factorizations with the spherical k-means clustering*, magistrsko delo, University of Colorado, Boulder, 2003.

[25] *YouTube*, [ogled 04.05.2016], dostopno na `https://www.youtube.com/`.

Stvarno kazalo