Nenegativna faktorizacija matrice u modeliranju tema i grupiranju teksta

Gregor Boris Banušić
PMF - Matematika, Zagreb, Hrvatska

Domagoj-Jure Galić
PMF - Matematika, Zagreb, Hrvatska

Domagoj Lacmanović
PMF - Matematika, Zagreb, Hrvatska

Filip Milinković
PMF - Matematika, Zagreb, Hrvatska

1. Sažetak

Nenegativna matrična faktorizacija - NMF (engl. *Nonnegative matrix factorization*) je aproksimacija nenegativne matrice sa produktom dvije matrice nižeg ranga. Kako rješenje faktorizacije možemo lako interpretirati u grupiranju podataka (engl. *data clustering*), NMF je široko korištena metoda za grupiranje tekstualnih dokumenata te modeliranje tema.

U dokumentu ćemo opisati nekoliko osnovnih činjenica NMF-a te optimizacijski alat blok koordinatni spust(engl. *block coordinate descent*). U kontekstu grupiranja, naš alat pruža jednostavan način za nadogranju NMF-a, kao što je rijedak NMF (engl. *sparse NMF*) i slabo-nadziran NMF (engl. *weakly-supervised NMF*).

U ovom dokumentu ćemo prezentirati eksperimentalne rezultate koji pokazuju superiornost alata što se tiče konvergencije te zadovoljavajuću točnost grupiranja.

2. Uvod

Nenegativna faktorizacija matrice je metoda smanjivanja dimenzije. Mnoge tehnike smanjivanja dimenzije su usko vezane uz nisko rangirane aproksimacije nekih matrica, a NMF je specijalan po tome što su njegove matrice umnoška niskog ranga nenegativne. NMF su prvi objavili Paatero i Trapper[5] kao pozitivnu faktorizaciju matrica. Zadnjih dva desetljeća, NMF je dobio veliku pozornost te se počeo primjenjivati u mnogim važnim problemima kao što su rudarenje podataka (engl. *text mining*), računalni vid(engl. *computer vision*), bioinformatika i mnoge druge.

Neka je dana nenegativna matrica $A \in \mathbb{R}^{m \times n}$. Ako imamo zadanu dimenziju $k \in \mathbb{N}$, cilj NMF-a je pronaći dvije nenegativne matrice $W \in \mathbb{R}^{m \times k}$ i $H \in \mathbb{R}^{k \times n}$ tako da

$$A \approx WH \tag{1}$$

Prema (1), svaki vektor reprezentiran kao stupac od A, može biti aproksimiran sa kombinacijom nenegativne vektora koji čine bazu, a koji su reprezentirani sa stupcima matrice W. Kako je cilj smanjenja dimenzije $k << \min\{n, m\}$.

Matrice W i H ćemo računati rješavajući optimizacijski problem definiran Forbeniusovom normom (mjera udaljenosti dvije matrice), čija formulacija glasi:

$$W, H \leftarrow \operatorname{argmin}_{W \geq 0, H \geq 0} f(W, H) \tag{2}$$

gdje je f funkcija definirana sa:

$$f(W, H) = ||A - WH||_F^2 \tag{3}$$

gdje se vidi da matrice W i H moju biti nenegativne. NMF sa formulacijom (2) se pokazala doista uspješnom u grupiranju dokumenata i modeliranju tema.

Grupiranje dokumenata je veoma važno u rudarenju podataka sa ciljem razvrstavanja velike kolekcije u nekoliko semantičkih grupa. Pomoću takve grupacije korisnici lakše pretražuju dokumente.

Modeliranje tema je povezano sa mekim grupiranjem (engl. *soft clustering*) gdje su dokumenti prezentirani kao kombinacija težina za pojedinu temu, u smislu bliskosti dokumenta sa temom. Dodatno, zbog mekog grupiranja, modeliranje tema se odnosi i na semantičko značenje pojedine grupe/teme. Modelira se kao kombinacija težina za svaku riječ nekog teksta.

Cilj ovog članka je dati pregled kako se NMF koristi u grupiranju i modeliranju tema za tekstualne dokumente.

3. Redukcija dimenzije i grupiranje

Redukcija dimenzije i grupiranje su blisko povezani pojmovi. Kako bismo objasnili povezanost, razmotrimo aproksimaciju niskog ranga iz (1) gdje $A \in \mathbb{R}_+^{m \times n}$, $W \in \mathbb{R}_+^{m \times k}$, $H \in \mathbb{R}_+^{k \times n}$ te $k << min\{n, m\}$ prethodno navedeni niži rang. Stupci od A predstavljaju n podataka u m-dimenzionalnom prostoru. Svaki stupac od H je k-dimenzionalna reprezentacija podataka. Ako H shvatimo kao pridruživanje n podataka u k grupa, grupiranje možemo shvatiti kao specijalnu redukciju dimenzija.

Primjer redukcije dimenzije je algoritam K-sredina dan formulom:

$$\min \sum_{i=1}^{n} ||a_i - w_{g_j}||_2^2 \tag{4}$$

gdje su $a_1, ..., a_n$ stupci od A, $w_1, ..., w_k$ je k centroid te $g_i = j$ kada je i-ti podatak pridružen j-toj grupi ($1 \leq j \leq k$). Algoritam K-sredina formuliran kao problem redukcije dimenzije prikazan je sa:

$$\min_{H \in \{0,1\}^{k \times n}, H^T \mathbf{1}_k = \mathbf{1}_n} ||A - WH||_F^2 \tag{5}$$

gjde su oznake $\mathbf{1}_k \in \mathbb{R}^{k \times 1}$ i $\mathbf{1}_n \in \mathbb{R}^{n \times 1}$ vektori stupci čiji su elementi svi jednaki jedan.

U formulciji K-sredina pomoću formule (5) stupci matrice W su centroide grupa. Za svaki stupac matrice H postoji jedan element različit od 0 i jednak jedan, koji reprezentira dodijeljenu grupu.

Još jedan primjer redukcije dimenzije je NMF:

$$\min_{W \geq 0, H \geq 0} ||A - WH||_F^2. \tag{6}$$

Stupci matrice W predstavljaju k-dimenzionalni prostor, dok stupci matrice H predstavljaju reprezentaciju od $a_1, ..., a_n$ u tom k-dimenzionalnom prostoru. Zbog nenegativnosti matrice H, ova formulacija može biti interpretirana kao rezultat grupiranja: stupci od W su reprezentirani kao k grupa, a svaki element i-tog stupca matrice H predstavlja težinu pridruživanja podatka u svaku od k grupa. Vektori stupci u W reprezentiraju k tema, a koeficijenti u i-tom stupcu od H su koeficijenti pripadnosti od a_i pojedinoj temi. Za egzaktno grupiranje možemo uzeti temu čiji je koeficijent uz a_i najveći te tako dobiti za svaki podatak jednu grupu kojoj pripada.

Uspješnost NMF-a u grupiranju ovisi uvelike o podacima na kojima vršimo grupiranje te se najbolje pokazalo u grupiranju dokumenata. Podatci u dokumentima su često prikazivani kao vektori u linearnom prostoru. Za riječ-dokument matricu(engl. *term-document matrix*) A, vektor baze w_j je interpretiran kao distribucija riječi u temi. Kada je tih k distribucija linearno nezavisno, što je najčešće slučaj, NMF može dobro grupirati podatke.

4. Optimizacijski algoritmi za NMF

Iako je NMF np-težak problem[3], nadamo se da možemo naći lokalni minimum kao aproksimaciju rješenja NMF-a. U ovom odjeljku opisati ćemo algoritam koji optimizira ciljnu funkciju, zvan kvadratni koodrinatni spust (engl. *block coordinate descent, BCD*). Multiplikativno ažuriranje je također jedan od algoritama za rješavanje NMF-a no njime se nećemo baviti zbog spore konvergencije i slabih rješenja[1].

4.1. Blok koordinatni spust (BCD)

Algoritam *blok koordinatni spust* je široko primjenjiva strategija na nelinearnim optimizacijskim problemima. Algoritam dijeli varijable na nekoliko disjunktnih podgrupa te iterativno minimizira ciljnu funkciju u odnosu na dobivene varijable svake podgrupe.

U formulaciji (6) NMF-a, A je dana ulazna matrica a W i H su izlazne varijable koje se trebaju pronaći. Rješavamo manje probleme respektivno:

$$W \leftarrow \arg\min_{W \geq 0} f(W, H), \tag{7}$$

$$H \leftarrow \arg\min_{H \geq 0} f(W, H). \tag{8}$$

Probleme (7) i (8) možemo napisati kao:

$$\min_{W \geq 0} ||H^T W^T - A^T||_F^2, \tag{9}$$

$$\min_{H \geq 0} ||WH - A||_F^2. \tag{10}$$

U nastavku dajemo pseudokod BCD algoritma.

Algorithm 1 BCD algoritam za rješavanje NMF-a

1: Ulaz: matrica $A \in \mathbb{R}^{m \times n}$, parametar tolerancije $0 < \epsilon << 1$, maksimalan broj iteracija T
2: Inicijaliziraj H
3: **ponavljaj**
4: izračunaj optimalno rješenje problema (9)
5: izračunaj optimalno rješenje problema (10)
6: **dok** neki od zaustavnih kriterija baziran na W, H ili ϵ nije zadovoljen ili dok nije broj maksimalnih iteracija izvršen
7: Izlaz: W, H

Problemi (9) i (10) su problemi nenegativnih ograničenih najmanjih kvadrata (engl. *nonnegativity constrained least squares (NLS)*), dok je BCD poznat kao algoritam alternirajućih nenegativnih najmanjih kvadrata (engl. *alternating nonnegative least squares (ANLS)*). Primijetimo da moramo inicijalizirati matricu H i rješavati potprobleme (9) i (10) iterativno kao što je prikazano u algoritmu 1. Alternativno možemo inicijalizirati matricu W te rješavati navedene probleme obrnutim redoslijedom. Različite inicijalizacije rezultiraju različitim rješenjima NMF-a. Dobra praksa je algoritam pokrenuti sa različitim inicijalizacijama te uzeti rješenje sa najboljom aproksimacijom ciljne funkcije.

5. Konvergencija

Ciljna funkcija NMF-a je polinom četvrtog stupnja s obzirom na W i H, iz čega slijedi da je nekonveksna. Za optimizaciju nekonveksnog optimizacijskog problema većina algoritama garantira stacionarnost limesa, nije nužno da je dani limes točka minimuma. Sljedeći teorem se odnosi na konvergenciju BCD algoritma:

Teorem 1. Ako je za svaku iteraciju u potproblemima (9) i (10) dostignut minimum, tada svaka rubna točka niza $\{(W, H)^{(i)}\}$ generirana BCD algoritmom je stacionarna točka od (2).

BCD algoritam zahtjeva optimalna rješenja za svaki NLS potproblem. Minimum pojedinog potproblema ne mora biti jedinstven za održivost rezultat konvergencije zbog postojanja dva bloka[4]. Za stacionarnu točku u rješenju vrijede Karush-Kuhn-Tucher-ovi (KKT) uvjeti:

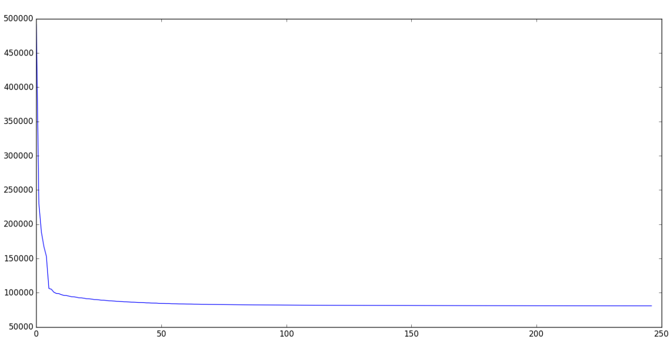
$$W \geq 0, \quad h \geq 0, \tag{11}$$

$$\begin{aligned} \nabla f_W &= 2WHH^T - 2AH^T \geq 0, \\ \nabla f_H &= 2W^TWH - 2W^TA \geq 0, \end{aligned} \tag{12}$$

$$W. * \nabla f_W = 0, \quad H. * \nabla f_H = 0, \tag{13}$$

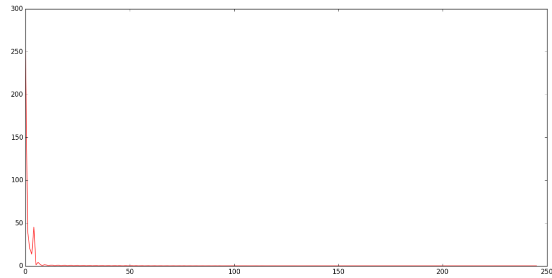
5.1. Rezultati konvergencije

U praksi nam se pokazalo da algoritam relativno brzo konvergira, osim kriterija zaustavljanja postavili smo maksimalan broj iteracija koji smo postavili na 500, no rijetko se desilo da je stvarno toliko iteracija potrebno. U prosjeku je bilo dovoljno pola od toga, odnosno oko 250. No neovisno o broju iteracija, grafički prikaz funkcije troška u ovisnsoti o broju iteracija, u pravilu je uvijek izgledao isto. Strmi pad do neke točke, te od tada relativno mala razlika do kraja. Kada smo samo naznačili iteraciju na kojoj bi algoritam stao te ga pustili da dođe do maksimalnog broja iteracija te usporedili razliku funkcije troška na naznačenom mjestu izlaska i na kraju zadnje iteracije, nerijetko smo dobili da razlika ispada ispod 1%. Kada smo nekoliko puta povečali maksimalan broj iteracija, greška se i dalje nije značajno povečala. Ovdje se postavlja pitanje izbora kriterija zaustavljanja i pogađanja optimalnog trenutka za prekid kako bi dobili najbolji mogući odnos točnosti i brzine. Prilažemo slike za jedan prosječan slučaj konvergencije i jedan kada smo pustili algoritam da dođe do maksimalnog broja iteracija. Prvo stavljamo prosječan slučaj.

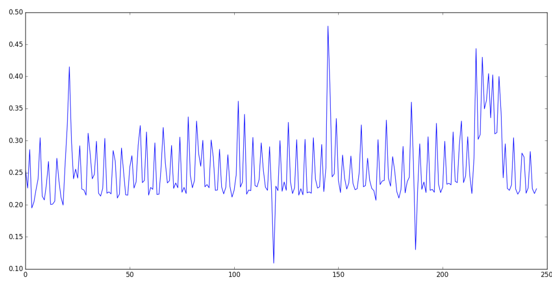


Slika 1. Funkcija troška u ovisnosti o broju iteracija za prosječan slučaj

Kao što smo i najavili, grafovi u pravilu izgledaju jednako, dakle ranijim izbacivanjem iz algoritma ne kompromitiramo točnost, što ide u prilog tezi da je kriterij zaustavljanja dobro izabran.



Slika 2. Relativna razlika dviju sustavnih aproksimacija u ovisnosti o broju iteracija za prosječan slučaj



Slika 3. Trajanje pojedine iteracije u sekundama za prosječan slučaj

6. Kriterij zaustavljanja

Iterativne metode zahtijevaju kriterij zaustavljanja kada algoritam dođe dovoljno blizu rješenja. U ovom članku koristili smo naivni pristup za zaustavljanje iteracija dan sljedećom formulom:

$$|f(W^{i-1}, H^{i-1}) - f(W^i, H^i)| \leq \epsilon \tag{14}$$

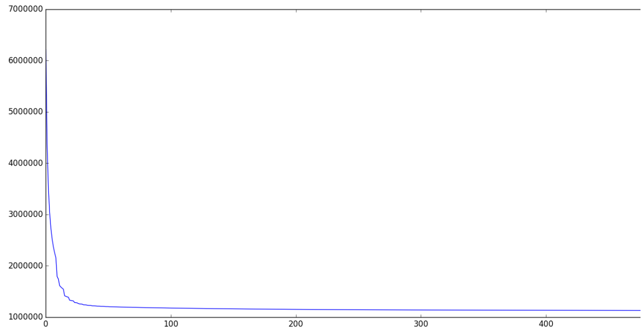
za neki prethodno definirani $\epsilon > 0$.

7. Rijetka NMF

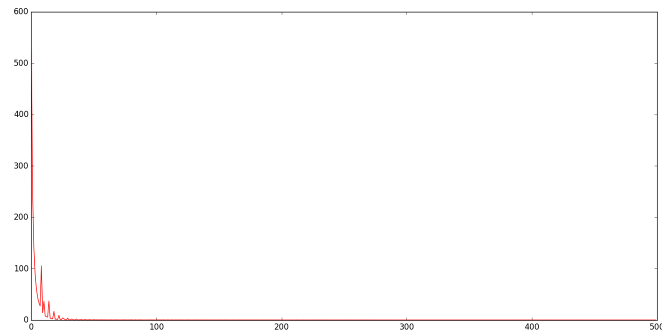
Iz 3. odjeljka znamo da matrica H iz NMF-a sadrži težine pripadnosti podatka temama. Na tragu ideje "manje je više" želimo odrediti matrice W i H sa što manje ne-nul elemenata. Osim veće interpretabilnosti time dobijemo lakše grupiranje. Idealno bi htjeli da imamo jedan velik element u svakom stupcu matrice H, te veliku razliku između najvećeg i ostalih elemenata. U suprotnom male razlike u veličini elementa, mogu drastično utjecati na krajnji izlaz algoritma. Drugim rječima, rješenje nije stabilno.
Taj rezultat postižemo modifikacijom funkcije troška. U najopćnitijem obliku funkcija troška dana je sa:

$$f(W, H) = ||A - WH||_F^2 + \phi(W) + \psi(H) \tag{15}$$

gdje su ϕ i ψ regularizacijski izrazi koji najčešće uključuju matričnu ili vektorsku normu. Mi regularizacijske izraze definiramo sa:



Slika 4. Funkcija troška u ovisnosti o broju iteracija bez kriterija zaustavljanja



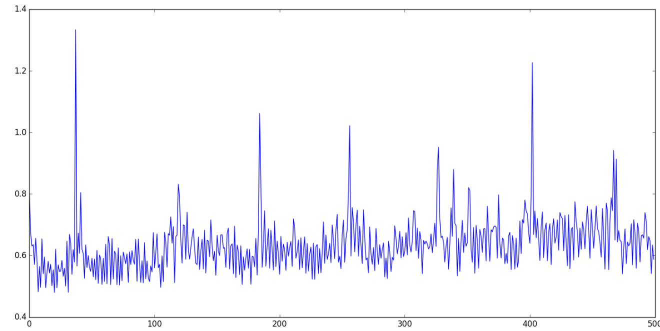
Slika 5. Relativna razlika dviju sustavnih aproksimacija u ovisnoti o broju iteracija bez kriterija zaustavljanja

$$\begin{aligned} \phi\left(W\right) &= \alpha\left|\left|W\right|\right|_F^2 \\ \psi\left(H\right) &= \beta\sum_{i=1}^{n}\left|\left|H(:,i)\right|\right|_1^2, \end{aligned} \tag{16}$$

gdje je $H(:,i)$ i-ti stupac matrice H. Ovdje smo izraz ψ definirali pomoću L^1 norme kako bi postigli rijetkost u stupcima matrice H, a izraz ϕ pomoću norme L^2 kako bi spriječili da elementi od W previše narastu. Rijetka NMF se može lako izračunati pomoću BCD-a. Kada u (9) i (10) uvrstimo izraze (16) i (17) dobijemo:

$$\begin{aligned} &\min_{W\geq 0}\left|\left|\begin{pmatrix} H^T \\ \sqrt{\alpha}I_k \end{pmatrix}W^T - \begin{pmatrix} A^T \\ 0_{k\times m} \end{pmatrix}\right|\right|_F^2 \\ &\min_{H\geq 0}\left|\left|\begin{pmatrix} W \\ \sqrt{\beta}\mathbf{1}_k^T \end{pmatrix}H - \begin{pmatrix} A \\ \mathbf{0}_n^T \end{pmatrix}\right|\right|_F^2 \end{aligned} \tag{17}$$

Ovdje su $\mathbf{1}_k \in \mathbb{R}^{k\times 1}$, $\mathbf{0}_n \in \mathbb{R}^{n\times 1}$ vektori stupci čiji su elementi 1 i 0, a I_k je $k \times k$ jedinična matrica. Nadalje, problemi (17) i (18) za rijetku NMF se mogu riješiti slično Algoritmu 1 za originalni NMF.



Slika 6. Trajanje pojedine iteracije u sekundama bez kriterija zaustavljanja

8. Slabo nadzirana NMF

Osim rijetke NMF promotriti ćemo još jednu bitnu varijantu koja se zove slabo-nadzirana NMF (WS-NMF). Slabo-nadzirana NMF može pripojiti razne korisničke ulazne podatke tako da se rezultati grupiranja i modeliranja tema mogu poboljšati. Korisnički ulazni podaci se manifestiraju u obliku "pomoćnih" matrica za W i H te je njihova uloga napraviti W i H sličnim njima. Neka su pomoćne matrice $W_r \in \mathbb{R}_+^{m\times k}$ za W, $H_r \in \mathbb{R}_+^{k\times n}$ za H te $M_W \in \mathbb{R}_+^{k\times k}$ i $M_H \in \mathbb{R}_+^{n\times n}$ dijagonalne težinske matrice

$$\begin{aligned} f(W,H,D_H) = \min_{W,H,D_H}\Big(&||A-WH||_F^2 + \\ ||(W-W_r)M_W||_F^2 + &||(H-H_rD_H)M_H||_F^2\Big) \end{aligned} \tag{18}$$

za $W \in \mathbb{R}_+^{m\times k}$, $H \in \mathbb{R}_+^{k\times n}$ i dijagonalnu $D_H \in \mathbb{R}_+^{n\times n}$.

Prije nego što krenemo o pojedinostima, dajmo prvo malo intuicije o WS-NMF - u. Poanta ovog pristupa je ugrađivanje nekakvog znanja o problemu otprije. Ako je na primjer poznat način na koji su podaci generirani i otprilike se zna kako bi podaci trebali biti grupirani, ili se zna samo za dio. Način na koji se to radi je penaliziranje "pogrešnog" oblika matrica W i H. Zato dodajemo matrice W_r i H_r. Osim toga možemo povećati težinu nekog stupca razlike matrica W i W_r odnosno H i H_r. To radimo matricama $M_W i M_H$ Ako nam je bitno da jedan stupac bude jako sličan, a drugi ne. Na taj način postižemo uključivanje prijašnjeg znanja. Osim toga imamo još i dijagonalnu matricu D_H, njezina uloga je čisto tehničke prirode i ona se u svakoj iteraciji računa kako naše penaliziranje isključuje slučaj kolinearnosti. Odnosno ako imamo dva vektora koji su kolinearni, njih želimo smatrati jednakima u ovom kontekstu, zbog mehanizma grupiranja koji mi koristimo, nije nam bitno s kojim faktorom stupci matrice H dolaze. Primjetimo da ne postoji matrica D_W, no to nije problem, jer za dijagonalnu matricu D_H vrijedi $WH = \left(WD_H^{-1}\right)\left(D_HH\right)$ odnosno skaliranjem matrice H indirektno smo skalirali i matricu W.

Nadalje, u formuli (19) vrijedi $\alpha||(W-W_r)M_W||_F^2$ je ekvivalentno $\alpha||(W-W_r)M_W^{new}||_F^2$ kad vrijedi $M_w^{new} =$

αM_W.
Isti argument vrijedi za M_H. Optimizacija iste formule slijedi BCD tako što iterativno računa W, H i D_H. Kad imamo početne uvjete za ove varijable, W se ažurira na sljedeći način

$$W \leftarrow \arg\min_{W \geq 0} \left|\left| \begin{bmatrix} H^T \\ M_W \end{bmatrix} W^T - \begin{bmatrix} A^T \\ M_W W_r^T \end{bmatrix} \right|\right|_F^2 \tag{19}$$

. Nadalje, svaki stupac od H se ažurira jedan po jedan rješavajući:

$$H(:,i) \leftarrow \arg\min_{H(:,i) \geq 0} \left|\left| \begin{bmatrix} W \\ M_H(i) I_k \end{bmatrix} H(:,i) - \begin{bmatrix} A(:,i) \\ M_H(i) D_H(i) H_r(:,i) \end{bmatrix} \right|\right|_F^2 \tag{20}$$

gdje $(: i)$ označava i ti stupac matrice. Konačno, i-ta dijagonalna komponenta $D_H(i)$ se dobije kao:

$$D_H(i) \leftarrow \begin{cases} \dfrac{H_r(:,i)^T H(:,i)}{||H_r(:,i)||_2^2} & \text{ako} \quad M_H(i) \neq 0 \\ 0 & \text{inače} \end{cases} \tag{21}$$

9. Set podataka

Skup podataka korištenu istraživanju se nalazi na Reuters-ovoj stranici [2] te sadrži 21 578 članaka. Članci se nalaze u više datoteka u .sgm formatu te sadrže xml strukturu. Radi bolje preciznosti i lakše interpretacije koristili smo samo članke čija je tematika jedinstvena sudeći po ¡TOPICS¿ tagu u svakom članku (za dodatna uputstava pročitati reutersov README.TXT koji se nalazi u zipu sa člancima). Zbog velikog broja podataka te puno različitih tema odabrali smo 20 tema s najvećim brojem članaka, u svrhu testiranja. Izbacili smo dokumente sa manje od 10 riječi jer smatramo da nam neće doprinijeti boljem grupiranju.

9.1. Težinsko označavanje tf-idf

Riječ-dokument matricu smo konstruirali koristeći tf-idf algoritam, gdje svaki redak pripada jednoj riječi a svaki stupac jednom dokumentu.

Dokument ili članak koji spominje riječ iz upita, što je u našem slučaju samo tema, više puta ima više veze sa tim upitom te bi stoga trebao dobiti veću pažnju. Kako bismo to postigli, svakoj riječi dokumenta pridodajemo neku težinu koja ovisi o broju pojavljivanja iste u tom dokumentu. Želimo izračunati korelaciju između upita, riječi t i dokumenta d, baziranom na težini riječi t u dokumentu d. Pridruživanje težine bazirane na broju pojavljivanja unutar dokumenta nazivamo frekvencijom riječi (engl. *term frequency*) i označavamo sa $tf_{t,d}$.

Problem sa gornjim pristupom je što su sve riječi jednako važne kada je riječ o procjeni relevantnosti za upit, tj. temu. Također neke riječi nemaju nikakvog učinka u određivanju relevantnosti. Radi toga ćemo smanjit težine riječi pomoću frekvencije dokumenata (engl. *document frequency*), definirane kao broj dokumenata u kojima se pojavljuje riječ t i označava se sa df_t.

Težinu riječi t skalirati ćemo pomoću formule za inverznu frekvenciju dokumenta (engl. *inverse document frequency*) u oznaci idf:

$$idf_t = log \frac{N}{df_t}$$

Gdje je N broj dokumenata u kolekciji. Iz formule vidimo da je idf od manje pojavljivanje riječi visok, dok je idf više frekventne riječi niži.

Algoritam tf-idf[6] kombinacijom frekvencije pojavljivanja i inverzne frekvencije dokumenta daje težinu svakoj riječi u dokumentu. Shema tf-idf za dodjeljivanje težina riječima dokumenata ide po formuli:

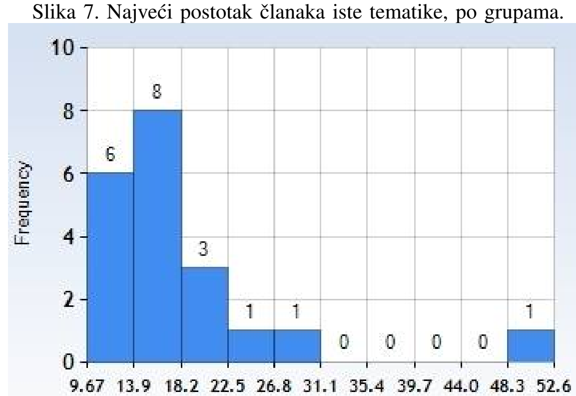
$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

Vrijednost je najveća kada t se pojavi mnogo puta u malom broju dokumenata, manja kada se riječ pojavi manje puta u dokumentu ili se pojavi u puno dokumenata, a najmanja kada se pojavljuje u gotovo svim dokumentima.

10. Rezultati grupiranja

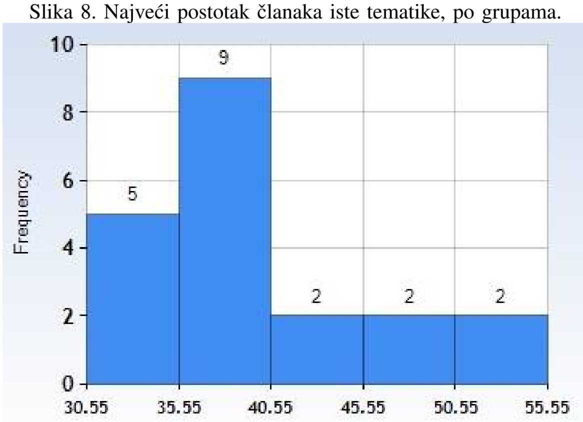
Nakon testiranja NMF-a na podatcima od 400 dokumenata, iz svake od 20 tema po 20 dokumenata, dobili smo sljedeće rezultate:

Slika 7. Najveći postotak članaka iste tematike, po grupama.



Vidimo da je samo jedna grupa uspjela imati 50% članaka iste tematike, dok su ostali poprilično raznovrsno raspoređeni. Ako gledamo više od 4000 dokumenata koji se nalaze u podacima, za koje samo znamo da pripadaju jednoj od 20 grupa, dobijemo sljedeće rezultate

Slika 8. Najveći postotak članaka iste tematike, po grupama.



Ovi naizgled dobri rezultati zapravo daju lažnu sliku grupiranja zbog velikog nesrazmjer broja članaka pojedine tematike. Slika 2 prikazuje 20 grupa od kojih je u gotovo svakoj najviše dokumenata jedne, iste klase. Pretpostavljamo da su problemi, zbog kojih smo dobili ove rezultate, relativno mali članci sa malo riječi te usko specijalizirana područja po kojima smo htjeli grupirati.

Pokušali smo sa matricama A različitih dimenzija, no unatoč tome rezultati su bili slični u odnosu na $slika8$.

11. Alternirajući nenegativni najmanji kvadrati

U strukturi rješenja problema nenegativne matrične faktorizacije kao potproblem se javlja rješavanje problema alternirajućih nenegativnih najmanjih kvadrata. Mi smo se odlučili na aproksimaciju rješenja metodom gradijentnog spuštanja (eng. gradient descent), točnije metodom projiciranog gradijenta za optimizaciju sa uvjetima i to iz dva razloga. Sa metodom gradijentnog spuštanja imamo puno finiju kontrolu nad brojem iteracija, tojest možemo birati koja nam je točnost prihvatljiva i time potencijano dodatno ubrzati algoritan. Drugi razlog je mogučnost paralelizacije algoritma, kojim se još dodatno dobiva na brzini algoritma, što je pogotovo značajno za velike količine podataka (eng. big data) iako se mi time nismo bavili, ali ostavljamo mogućnost dorade.

11.1. Formulacija problema

Zapravo razmatramo standardnu zadaću optimizacije sa uvjetima:

$$\begin{aligned} &\min_{X\in\mathbb{R}^{m\times n}} f(X) \\ &\text{uz uvjet} \quad 0 \leq X_{ij}, \quad i=1,\dots,m,\ j=1,\dots,n \end{aligned} \tag{22}$$

Gdje je $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ neprekidno diferencijabilna funkcija. Neka je k indeks iteracije, metodu projeciranog gradijenta realiziramo na način da trenutno rješenje X^k nadogradimo do X^{k+1}, sljedećim pravilom:

$$X^{k+1} = P[X^k - \alpha^k \nabla f(X^k)]$$

gdje je

$$P[x] = \begin{cases} x, & \text{za } x > 0 \\ 0, & \text{inaće} \end{cases}$$

Funkcija koju mi minimiziramo je kvadrat Frobenijusove matrične norme, to jest naš problem svodi se na rješavanje problema:

$$\min_{H\geq 0} f(H) = \min_{H\geq 0} \frac{1}{2} \left|\left|A - WH\right|\right|_F^2$$

Gdje su A i W konstantne matrice. Gradijent funkcije f dan je sa:

$$\nabla f(H) = W^T (WH - A)\,.$$

Drugi problem rješavamo slično

$$\min_{W\geq 0} \bar{f}(W) = \min_{W\geq 0} \frac{1}{2} \left|\left|A^T - H^T W^T\right|\right|_F^2$$

Gdje su sada A i H konstantne matrice. Gradijent funkcije f dan je sa:

$$\nabla \bar{f}(H) = (WH - A)\, H^T.$$

Postupak je isti za f i $\bar{f}$.

12. Zaključak

U ovom radu promatrali smo nenegativnu matričnu faktorizaciju i njenu primjenu u modeliranju i grupiranju teksta. Naglasak ovog rada bio je više na matematičkom, odnosno algoritamskom dijelu te smo se gdje je bilo moguće dati prednost intuiciji i interpretaciji nad formalizmom, ali ne kompromitirajući preciznost izlaganja. Izložili smo osnovnu formulaciju problema nenegativne matrične faktorizaciji diskutirali njenu primjenu u grupiranju dokumenata. Nakon toga smo uveli algoritam blok koordinatnog spuštanja (BCD) diskutirali njegovu implementaciju pomoću alternirajućih nenegativnih najmanjih kvadrata. Također toga smo diskutirali konvergenciju algoritma te kriterije zaustavljanja. Te smo uveli i dvije varijacije NMF-a, rijetki NMF i WS-NMF te objasnili zašto se koriste. Diskutirali smo i neke probleme s kojima smo se susreli tijekom pisanja ovog rada i predložili neka rješenja. Također smo iznjeli i neke ideje koje nismo stigli implementirati na primjer paralelizacija algoritma, te eventualno testiranje nekih drugih kriterija zaustavljanja.

Literatura

[1] Da Kuang, Jaegul Choo, and Haesun Park (2015) Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering, Partitional Clustering Algorithms, pp. 215-243, Springer, 2015 (Book chapter)

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3] Berman A, Plemmons RJ (1994) Nonnegative matrices in the mathematical sciences. SIAM, Philadelphia

[4] Grippo L, Sciandrone M (2000) On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Oper Res Lett 26:127–136

[5] Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5:111–126

[6] Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge

[7] Chih-Jen Lin (2007) Projected Gradient Methods for Non-negative Matrix Factorization. Neural Computing