Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Klančar

Algoritmi za reševanje problema matričnih napolnitev

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Aljaž Zalar

Ljubljana, 2023

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.

Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani `http://www.gnu.org/licenses/`.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Kandidat: Matej Klančar
Naslov: Naslov diplomskega dela
Vrsta naloge: Diplomska naloga na univerzitetnem programu prve stopnje Računalništvo in informatika
Mentor: doc. dr. Aljaž Zalar

Opis:
Besedilo teme diplomskega dela študent prepiše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode naj uporabi, morda bo zapisal tudi ključno literaturo.

Title: Algorithms for solving matrix completion problem

Description:
opis diplome v angleščini

Na tem mestu zapišite, komu se zahvaljujete za pomoč pri izdelavi diplomske naloge oziroma pri vašem študiju nasploh. Pazite, da ne boste koga pozabili. Utegnil vam bo zameriti. Temu se da izogniti tako, da celotno zahvalo izpustite.

Svoji dragi Alenčici.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
DBMS	database management system	sistem za upravljanje podatkovnih baz
SVM	support vector machine	metoda podpornih vektorjev

Povzetek

Naslov: Algoritmi za reševanje problema matričnih napolnitev

Avtor: Matej Klančar

V vzorcu je predstavljen postopek priprave diplomskega dela z uporabo okolja LaTeX. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek. Dober povzetek vključuje: (1) kratek opis obravnavanega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek diplomske naloge.

Ključne besede: računalnik, računalnik, računalnik.

Abstract

Title: Diploma thesis template

Author: Matej Klančar

This sample document presents an approach to typesetting your BSc thesis using LaTeX. A proper abstract should contain around 100 words which makes this one way too short.

Keywords: computer, computer, computer.

Poglavje 1

Uvod

Problem matričnih napolnitev sprejme matriko, največkat označeno z M, pri kateri so nekateri elementi označeni kot neznani. Problem nato sprašuje po vrednostih, ki jih lahko vstavimo v neznane vrednosti, tako da bo rang matrike najmanjši možen. Gre za NP-poln problem, zato ga poskušamo poenostaviti, ter reševati lažje probleme, ki vrnejo dovolj dobre, a ne optimalne rešitve.

Problem je v zadnjih letih zelo popularen, z njim pa se ukvarjajo tako številni matematiki kot računalničarji. Njegova splošnost naredi reševanje problema na številnih področjih, sam pa se v diplomski nalogi osredotočim na razreševanje neznanih pikslov v slikah. Prav tako omenjam in preizkusim algoritem na priporočilnih sistemih. Rezultate teh predstavim v poglavju X.

V tej diplomski nalogi bom predstavil par algoritmov, ki rešujejo omenjen problem. Algoritmi so bili izbrani glede na njihovo popularnost in priznanost v literaturi. Prav tako sem poskrbel, da so algoritmi primerno različni in temeljijo na drugačnih principih. Algoritme sem tudi implementiral, nato pa še napravil analizo ter opisal ugotovitve v poglavju X.

Poglavje 2

Pregled področja

Področje je trenutno zelo aktivno, z številnimi raziskovalci, ki se specializirajo na dano področje. Eden vodilnih raziskovalcev je Emmanuel J. Candès, na čigar dela se tekom diplomske naloge večkrat sklicujem.

Vodilna literatura tekom pisanja diplomske naloge je bil članek [**survey**], ki opisuje problem, ter mnoge algoritme. Medtem ko opisi pogosto niso bili dovolj podrobni, da bi lahko začel algoritme implementirati, je članek ponujal dobro razumljive opise algoritmov, kot tudi navedel vire, ki so pomagali pri implementaciji. V kasnejših poglavjih se tam, kjer sem članke potreboval, nanje sklicujem.

Poglavje 3

Algoritmi

3.1 Pomembe definicije

Nekatere definicije so uporabljene čez več algoritmov. Z namenom preglednosti, te opisujem v tem poglavju

1. Ω je definirana množica znanih vrednosti
2. $$[\mathcal{P}_\Omega]_{i,j} = \begin{cases} a_{ij} & (i,j) \in \Omega \\ 0 & \text{drugače} \end{cases}$$
3. Operator $\mathcal{D}_\tau$ kot
$$\mathcal{D}_\tau(A) := U\mathcal{D}_\tau(\Sigma)V^T, \quad \mathcal{D}_\tau(\Sigma) = diag(max(\sigma_i - \tau, 0))$$
[1]
4. Z oznako $M \in \mathbb{R}^{n_1 \times n_2}$ označujemo vhodno matriko, torej tisto, ki ima nekatere podatke neznane.

V programu nato uporabljamo oznako M za bitno matriko, vendar te v samih dokazih ne potrebujemo, je potem tukaj oznaka M vredu?

3.2 Minimizacija nuklearne norme

Minimizacija nuklearne norme (Nuclear Norm Minimization oziroma NNM) se zanaša na dejstvo, da je rang matrike povezan z nuklearno normo matrike.

Ta je definirana kot

$$||A||_* = \sum_{i=0}^{n} \sigma_i(A)$$

.

Minimizacijo nuklearne norme je možno pretvoriti v semidefinitni problem, ki ga lahko rešujemo z različnimi pripomočki, na primer SeDuMi [4].

Po [2] lahko problem definiramo kot

$$\begin{aligned} \min \quad & tr(Y) \\ \text{tako da} \quad & (Y, A_k) = b_k, k = 1, \cdots, |\Omega| \\ & Y \succcurlyeq 0 \end{aligned}$$

kjer

$$Y = \begin{bmatrix} W_1 & X \\ X^T & W_2 \end{bmatrix}$$

tak problem pa lahko že rešujemo z semidefinitnimi programi.

3.3 Prag singularnih vrednosti

Algoritem praga singularnih vrednosti, oziroma v nadaljevanju SVT (Singular Value Thresholding) sloni na dejstvu, da imamo pri matrikah z majhnim rangom nekaj velikih singularnih vrednosti, ostale pa blizu ničli. Za svoje delovanje uvede dva nova pomembna koncepta, prvi je premik, drugi pa prag, potreben za uporabo operatorja $\mathcal{D}_\tau$. Algoritem lahko na kratko povzamemo z zapisom

$$\begin{cases} X^k = \mathcal{D}_\tau(Y^{k-1}) \\ Y^k = Y^{k-1} + \delta_k \mathcal{P}_\Omega(M - X^k) \end{cases}$$

kjer $\tau > 0$ predstavlja prag, δ_k predstavlja k-ti premik, $X \in \mathbb{R}^{n_1 \times n_2}$ ter $Y^0 = 0 \in \mathbb{R}^{n_1 \times n_2}$. [1]

Smiselnost algoritma lahko pokažemo s pomočjo Lagrangeovega multiplikatorja. Ponovno rešujemo minimizacijski problem, le da dodamo dodatne parametre in s tem minimizacijo omilimo.

Uvedimo funkcijo $f_\tau(X) = \tau||X||_* + \frac{1}{2}||X||_F^2$. Problem lahko tako zapišemo kot

$$\begin{aligned} \min \quad & f_\tau(X) \\ \text{tako da} \quad & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M) \end{aligned}$$

Lahko je videti, da za velike vrednosti τ velja $f_\tau(X) \approx \tau||X||_*$, kar pokaže, da bo s primerno velikim τ, algoritem res skoraj minimiziral nuklearno normo.

Lagrangeov multiplikator je definiran kot $\mathcal{L}(x, \lambda) = f(x) + \lambda \cdot g(x)$, kjer je $f(x)$ funkcija, ki jo minimiziramo, pod pogojem da velja $g(x) = 0$. Naš problem tako prevedemo v

$$\mathcal{L}(X, Y) = f_\tau(X) + \langle Y, \mathcal{P}_\Omega(M - X)\rangle$$

S pomočjo tako imenovanega Uzawoega algoritma, pa lahko problem pretvorimo v iterativni algoritem

$$\begin{cases} \mathcal{L}(X^k, Y^{k-1}) = \arg\min_X \mathcal{L}(X^k, Y^{k-1}) \\ Y^k = Y^{k-1} + \delta_k \mathcal{P}_\Omega(M - X^k) \end{cases}$$

[1]

Za reševanje minimizacijskega problema dokažimo teorem

$$\mathcal{D}_\tau(Y) = \arg\min_X \{\frac{1}{2}||X - Y||_F^2 + \tau||X||_*\}$$

Ker je $h(X) := \frac{1}{2}||X - Y||_F^2 + \tau||X||_*$ strogo konveksna funkcija, lahko za subgradient Z v točki X_0 rečemo, da velja $\forall X : f(X) \geq f(X_0) + \langle Z, X - X_0\rangle$. Ali drugače povedano, vse točke na vseh tangentah na funckijo $h(X)$ bodo pod ali na funkciji $h(X)$. To velja po sami definiciji, saj je to zahtevan pogoj subgradienta v neki točki.

Pri iskanju minimuma torej iščemo tako točko X', da bo subgradient v točki X' enak 0. Problem sedaj zapišemo kot $0 \in X' - Y + \tau\partial||X'||_*$. Izkaže

se, da je množica subgradientov nuklearne norme definirana kot

$$\partial||X||_* = \{UV^* + W : W \in \mathbb{R}^{n_1 \times n_2}, U^*W = 0, WV = 0, ||W||_2 \leq 1\}.$$

Preveri kako se to citira

kjer $U\Sigma V^T$ predstavlja SVD razcep matrike X.

Cilj dokaza je pokazati, da velja $X' = \mathcal{D}_\tau(Y)$ Najprej razčlenimo SVD razcep matrike Y kot

$$Y = U_0\Sigma_0 V_0^T + U_1\Sigma_1 V_1^T$$

, kjer U_0, Σ_0 in V_0 predstavljajo lastne vrednosti ter njihove lastne vektorje večje od τ, U_1, Σ_1 in V_1 pa tiste manjše od τ. Ustrezno je torej pokazati, da velja

$$X' = U_0(\Sigma_0 - \tau I)V_0^T$$

Gre preprosto za drugačen zapis operatorja $\mathcal{D}_\tau(Y)$. Če zapis vstavimo v prejšnji podan pogoj dobimo

$$\begin{aligned} 0 &= X' - Y + \tau\partial||X||_* \\ Y - X' &= \tau(U_0V_0^T + W) \end{aligned}$$

primerna izbira za $W = \tau^{-1}U_1\Sigma_1V_1^T$, saj

$$\begin{aligned} Y - X' &= U_0\Sigma_0V_0^T + U_1\Sigma_1V_1^T - U_0(\Sigma_0 - \tau I)V_0^T = \\ &= U_0V_0^T(\Sigma_0 - \Sigma_0 + \tau I) + U_1\Sigma_1V_1^T = \\ &= \tau U_0V_0^T + U_1\Sigma_1V_1^T \end{aligned}$$

in

$$\begin{aligned} \tau(U_0V_0^T + W) &= \tau(U_0V_0^T + \tau^{-1}U_1\Sigma_1V_1^T) \\ &= \tau U_0V_0^T + U_1\Sigma_1V_1^T \end{aligned}$$

Sedaj je zgolj potrebno pokazati, da veljajo potrebne lastnosti matrike W.

je to res pravilen razlog?

Po sami definiciji SVD vemo, da so vsi stolpci matrik U in V ortogonalni. Torej velja $U_0^TW = 0$ in $WV_0 = 0$. Ker pa ima matrika Σ_1 vse elemente manjše od τ velja tudi $||W||_2 \leq 1$. S tem smo pokazali, da $Y - X' \in \tau\partial||X'||_*$.

zakaj lahko zapisemo $\mathcal{L}(X^k, Y^{k-1}) = X^k$?

Prav tako lahko sedaj zapišemo algoritem kot

$$\begin{cases} X^k = \mathcal{D}_\tau(Y^{k-1}) \\ Y^k = Y^{k-1} + \delta_k \mathcal{P}_\Omega(M - X^k) \end{cases}$$

[1].

3.3.1 Nastavljanje parametrov

Medtem, ko so koraki v samem algoritmu definirani kot množica korakov, sem sam za premik uporabljal konstanto, ter korak nastavil na

$$\delta = 1.2 \frac{n_1 n_2}{m}$$

po priporočilih [1].

Prav tako članek navaja, da je za matrike velikosti $\mathbb{R}^{n \times n}$ smiselno nastaviti $\tau = 5n$, vendar sem v moji implementaciji zaradi posploševanja na nekvadratne matrike, za matrike velikosti $\mathbb{R}^{n_1 \times n_2}$ parameter nastavil na

$$\tau = 5 \frac{n_1 + n_2}{2}$$

Medtem ko so taki parametri dobri za večje matrike, jih ne smemo uporabljati kot definitivno najboljše vrednosti. Med mojimi testiranji sem ugotovil, da je vrednost premika velikokrat treba zmanjšati, posebno za manjše matrike z več neznanimi vrednostmi, saj drugače program ni konvergiral. Prav tako, se je večkrat zgodilo, da je bil pridobljen rezultat še vedno zelo zašumljen. Takrat je bilo smiselno prag τ povečati. To je sicer upočasnilo program, vendar izboljšalo rezultat.

3.4 Minimizacija skrajšane nuklearne norme

Že samo ime nam pove, da bo algoritem minimizacije skrajšane nuklearne norme, oziroma TNNM (Truncated Nuclear Norm Minimization) podoben algoritmu NNM. Vendar tu privzamemo, da imamo o samem algoritmu še neko

dodatno informacijo $r \in \mathbb{N}$, ki je povezana z rangom originalne, nezašumljene matrike.

Sam algoritem uvede tako imenovano skrajšano nuklearno normo, ki je za matriko $X \in \mathbb{R}^{n_1 \times n_2}$ definirana kot

$$||X||_r = \sum_{i=r+1}^{\min(n_1,n_2)} \delta_i(X)$$

torej vsoto $\min(n_1, n_2) - r$ najmanjših singularnih vrednosti, ter z njeno pomočjo definira problem

$$\begin{aligned} \min_X \quad & ||X||_r \\ \text{tako da} \quad & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M) \end{aligned}$$

[3].

Cilj algoritma je torej čim bolj zmanjšati najmanjše singularne vrednosti, medtem ko velikih ne omejujemo. S tem problem minimizacije omilimo.

Za reševanje minimizacije bomo uporabljali algoritem ADMM, vendar moramo prej sam problem nekoliko spremeniti. Razpišimo problem kot

$$\begin{aligned} \min_X \quad & ||X||_* - \sum_{i=1}^{r} \sigma_i \\ \text{tako da} \quad & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M) \end{aligned}$$

.

Za nadaljne korake bomo potrebovali teorem

$$Tr(AXB^T) \leq \sum_{i=1}^{r}$$

kjer velja $X \in \mathbb{R}^{n_1 \times n_2}$, $A \in \mathbb{R}^{r \times n_1}$, $B \in \mathbb{R}^{r \times n_2}$ in $r \in \mathbb{N}, r \leq min(n_1, n_2)$, kot tudi $AA^T = I_r, BB^T = I_r$.

Za dokaz uporabljamo Von Neumannovo neenakost sledi, s katero lahko zapišemo

$$Tr(AXB^T) = Tr(XB^TA) \leq \sum_{i=1}^{\min(n_1,n_2)} \sigma_i(X)\sigma_i(B^TA)$$

Enakost $Tr(AXB^T) = Tr(XB^TA)$ sledi iz dejstva, da je sled produkta matrik invariantna pod cikličnimi permutacijami.

Po definiciji lahko singularne vrednosti matrike Y najdemo tako, da najdemo korene nenegativnih lastnih vrednosti matrike Y^TY. Tako lahko povemo, da so singularne vrednosti matrike B^TA enake lastnim vrednostim matrike A^TBB^TA. Izraz razpišemo v

$$A^TBB^TA = A^TI_rA = A^TA$$

Ker pa velja, da imata matriki XY in YX enake neničelne lastne vrednosti, ter vemo da $AA^T = I_r$, lahko povemo, da ima produkt B^TA r singularnih vrednosti enakih 1, saj ima I_n n lastnih vrednosti enakih 1.

ali je to treba dokazati?

Tako lahko sedaj razpišemo izraz

$$\sum_{i=1}^{\min(n_1,n_2)} \sigma_i(X)\sigma_i(B^TA) = \sum_{i=1}^{r} \sigma_i(X)$$

Ugotovili smo, da velja

$$Tr(AXB^T) \leq \sum_{i=1}^{r} \sigma_i(Xs)$$

Članek navaja uvedbo s, ker $rank(B^TA) \leq r$, vendar bi lahko tako pokazali da je kar enak r?

Če definiramo SVD razcep $X = U\Sigma V^T$, ter matriki $A = (u_1, \ldots, u_r)^T$ in $B = (v_1, \ldots, v_r)^T$, kjer je u_i i-ti stolpec matrike U ter v_i i-ti stolpec matrike V, lahko pokažemo da velja $Tr(AXB^T) = \sum_{i=1}^{r} \sigma_i(X)$.

Zakaj je pomembno pokazati in ¡= in =

$$\begin{aligned}
Tr(AXB^T) &= Tr((u_1, \ldots, u_r)^T X (u_1, \ldots, u_r)^T) = \\
&= Tr((u_1, \ldots, u_r)^T U\Sigma V^T (u_1, \ldots, u_r)^T) = \\
&= \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} = \\
&= Tr(\begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}) = \sum_{i=1}^{r} \sigma_i(X)
\end{aligned}$$

Tako smo sedaj pokazali, da velja

$$\max_{AA^T=I,BB^T=I} Tr(AXB^T) = \sum_{i=1}^{r} \sigma_i(X)$$

Torej je problem, ki ga minimizarmo lahko podan kot

$$\begin{aligned} \min_X \quad & ||X||_* - \max_{AA^T=I,BB^T=I} Tr(AXB^T) \\ \text{tako da} \quad & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M) \end{aligned}$$

Glede na vse ugotovitve, nato nastavimo iterativni algoritem, tako da, izračunamo $X^0 = \mathcal{P}_\Omega(M)$. V i-ti iteraciji izračunamo A^i in B^i, tako da izračunamo SVD razcep $X^i = U\Sigma V^T$, ter A nastavimo kot prvih r stolpcev matrike U, B pa kot prvih r stolpcev matrike V. X^{i+1} lahko sedaj izračunamo kot

$$\begin{aligned} \min_X \quad & ||X||_* - Tr(A^i X (B^i)^T) \\ \text{tako da} \quad & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M) \end{aligned}$$

[3]

To minimizacijo pa lahko rešujemo z uporabo algoritma ADMM.

ADMM vprasanja: kako smo nastavili lagrange-ovo funkcijo, iz kje pride izracun Y^{i+1}, zakaj τ postane $\frac{1}{\beta}$

Poglavje 4

Rezultati

V tem poglavju bom opisal rezultate, ki sem jih pridobil. Kot sem že omenil, bo večji del preizkušanja programa opravljen na slikah, kjer bodo nekateri piksli manjkali. Gre za problem, ki ga je moč lepo vizualizirati, saj pogosto pri surovih podatkih ni lahko definirati njihovo uporabnost, brez številnih metod in računalniških operacij.

Prav tako bom opisal točnost rezulatov različnih metod kot tudi čas izvajanja posameznih metod. Probleme bom zagnal tudi na različnih vrst podatkov, npr. podatkih ki so generirani normalno kot tudi enakomerno porazdeljeno.

Poglavje bom začel pisati, ko končam z vsemi implementacijami, pri delu pa si bom pomagal z orodjem Microsoft Excel, kjer bom lahko napake in čas izvajanja tudi grafično vizualiziral.

Poglavje 5

Zaključek

Ko bodo vsa prejšnja poglavja končana, se bom lahko lotil pisanja zaključka. V njem bom opisal kaj vse sem tekom pisanja diplomske naloge ugotovil ter opravil, ter povedal če sem z rezultati zadovoljen. Omenil bom tudi kaj bi lahko spremenil in kako bi algoritme nadgradil, ter omenil par idej, na katerih raziskovalci trenutno delajo, ter opisal kako se področje razvija.

Literatura

[1] Jian-Feng Cai, Emmanuel J. Candes in Zuowei Shen. *A Singular Value Thresholding Algorithm for Matrix Completion*. 2008. arXiv: 0810.3286 [math.OC].

[2] Emmanuel J. Candès in Benjamin Recht. "Exact Matrix Completion via Convex Optimization". V: *CoRR* abs/0805.4471 (2008). arXiv: 0805.4471. URL: http://arxiv.org/abs/0805.4471.

[3] Yao Hu in sod. "Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization". V: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.9 (2013), str. 2117–2130. DOI: 10.1109/TPAMI.2012.271.

[4] Jos F. Sturm. "Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones". V: *Optimization Methods and Software* 11.1-4 (1999), str. 625–653. DOI: 10.1080/10556789908805766. URL: https://doi.org/10.1080/10556789908805766.