Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Klančar

Vzorec diplomskega dela

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: Naslov diplomskega dela v angleščini

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: Vzorec diplomskega dela

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

Poglavje 2

Pregled sorodnih del

[1] Opisuje tehnike, ki jih je uporabila zmagovalna ekipa Netflixove nagrade. Poglavje 3.1 opisuje uporabo k-NN metode ter izračun uteži glede na podatke o uporabnikih in filmih.

Poglavje 3.2 opisuje modele latentnih faktorjev. Metoda uporablja regulitacijo s členom $\lambda(||p_u||^2 + ||q_i||^2)$, kjer so ugotovili, da je najboljše določiti $\lambda = 0.05$. Algoritem omogoča iskanje medsebojnih vplivov uporabnikov in filmov zaradi določenih lastnosti.

[3] Opisuje uporabo matričnih napolnitev. Poglavje 1.1.1 opisuje katere matrike lahko napolnimo. Poglavja 1.1.3, 1.3 opisujejo katere algoritme je vredno obravnavati, poglavje 1.4 pa kako lahko to dosežemo z uporabo semidefinitnega programiranja.

Poglavje 2 opisuje inkoherenco, vrednost, ki nam pomaga določiti verjetnost, da bomo matriko lahko obnovili.

Poglavje 7 definira preizkus uporabe algoritmov matričnih napolnitev, ter predstavlja rezultate kdaj je bilo matrike možno obnoviti in kdaj ne.

[6] Kratek video o matričnih napolnitvah, ki vsebuje preprost a počasen algoritem za reševanje problema.

2.1 Singularni razcep

Pri iskanju singularnega razcepa (SVD) matrike $A \in \mathbb{R}^{n_1 \times n_2}$ iščemo tri matrike, tako da velja

$$A = U \Sigma V^T$$

kjer velja, da je matrika $U \in \mathbb{R}^{n_1 \times n_1}$ ortogonalna in sestavljena iz levih singularnih vektorjev. Matrika $\Sigma \in \mathbb{R}^{n_1 \times n_2}$ je diagonalna, elementi na diagonali pa so po velikosti urejene singularne vrednosti matrike A. Matrika $V \in \mathbb{R}^{n_2 \times n_2}$ je ortogonalna, vendar sestavljena iz desnih singularnih vektorjev.

Razcep SVD je bližnje povezan z dekompozicijo lastnih vrednosti simetričnih matrik $A^T A$, AA^T ter $\left[\begin{smallmatrix} 0 & A^T \\ A & 0 \end{smallmatrix}\right]$. Algoritmi nato reducirajo matriko A na $A = U_1 B V_1^T$, kjer sta U_1 ter V_1^T ortogonalna, B pa ima elemente le na glavni diagonali ter na diagonali nad njo. Nato poiščemo SVD razcep matrike $B = U_2 \Sigma V_2^T$, ter razcep vstavimo v prejšnjo enačbo. Tako dobimo $A = U_1 U_2 \Sigma (V_1 V_2)^T$. Sedaj lahko vidimo da velja $U = U_1 U_2$ ter $V = V_1 V_2$.[4]

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. $$[\mathcal{P}_\Omega(A)]_{i,j} = \begin{cases} a_{ij} & (i,j) \in \Omega \\ 0 & \text{drugače} \end{cases}$$

3. Operator $\mathcal{D}_\tau$ kot

$$\mathcal{D}_\tau(A) := U\mathcal{D}_\tau(\Sigma)V^T, \quad \mathcal{D}_\tau(\Sigma) = diag(max(\sigma_i - \tau, 0))$$

[2]

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

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

.

Minimizacijo nuklearne norme je možno pretvoriti v semidefinitni problem, ki ga lahko rešujemo z različnimi pripomočki, na primer SeDuMi [5].

Po [3] 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

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

kjer $\tau > 0$ predstavlja prag, δ_k predstavlja k-ti premik, $X \in \mathbb{R}^{n_1 \times n_2}$ ter $Y^0 = 0 \in \mathbb{R}^{n_1 \times n_2}$. [2]

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

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

[2]

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

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

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

Preveri kako se to citira

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. 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 $Y - X' \in \tau\partial||X'||_*$.

je to res pravilen razlog?

Prav tako lahko sedaj zapišemo algoritem kot

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

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

[2].

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 [2].

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.

Literatura

[1] Robert M. Bell in Yehuda Koren. “Lessons from the Netflix Prize Challenge”. V: *SIGKDD Explor. Newsl.* 9.2 (dec. 2007), str. 75–79. ISSN: 1931-0145. DOI: 10.1145/1345448.1345465. URL: https://doi.org/10.1145/1345448.1345465.

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

[3] 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.

[4] James W. Demmel. *Applied Numerical Linear Algebra.* SIAM, jan. 1997. DOI: 10.1137/1.9781611971446.

[5] 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.

[6] Barry Van Veen. *Matrix Completion.* 2020. URL: https://www.youtube.com/watch?v=lSOFVKJ4Xfg (pridobljeno 12. 3. 2023).