

Poročilo za 1. domačo nalogo

Numerična Linearna Algebra

Lenart Treven
Fakulteta za matematiko in fiziko
Oddelek za matematiko

26. april 2018

1 Uvod

V poročilu najprej pokažemo veljavnost dveh formul, ki jih potem uporabimo pri praktičnem delu, pri iskanju maksimalne ukrivljenosti L -krivulje. Nato pokažemo rezultate, ki smo jih dobili s praktičnim delom ter podamo okviren pregled, o organizaciji matlabovih datotek. Na koncu podamo komentar.

2 Izpeljava $\eta(\alpha)$ in $\rho(\alpha)$

Na predavanjih smo pokazali, da je

$$\min_x \|Ax - b\|_2^2 + \alpha^2 \|x\|_2^2$$

dosežen pri

$$x = \sum_{i=1}^n \frac{\sigma_i}{\sigma_i^2 + \alpha^2} u_i^T b v_i, \quad (1)$$

kjer je

$$A = \begin{bmatrix} u_1 & \dots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

singularni razcep matrike $A \in \mathbb{R}^{m \times n}$. Velja:

$$\begin{aligned}\eta(\alpha) &= \|x(\alpha)\|_2^2 = x^T x \\ &= \sum_{i=1}^n \frac{\sigma_i}{\sigma_i^2 + \alpha^2} u_i^T b v_i^T \sum_{j=1}^n \frac{\sigma_j}{\sigma_j^2 + \alpha^2} u_j^T b v_j \\ &= \sum_{i=1}^n \sum_{j=1}^n \frac{\sigma_i}{\sigma_i^2 + \alpha^2} u_i^T b \frac{\sigma_j}{\sigma_j^2 + \alpha^2} u_j^T b v_i^T v_j\end{aligned}$$

Sedaj upoštevamo, da je

$$v_i^T v_j = \begin{cases} 1; & i = j, \\ 0; & \text{sicer.} \end{cases}$$

Dobimo:

$$\eta(\alpha) = \sum_{i=1}^n \frac{\sigma_i^2 \beta_i^2}{\sigma_i^2 + \alpha^2},$$

kjer je $\beta_i = u_i^T b$.

Pokažimo še enakost za $\rho(\alpha)$. Po eni strani velja:

$$\begin{aligned}\rho(\alpha) &= \|Ax - b\|_2^2 = (Ax - b)^T (Ax - b) \\ &= x^T A^T A x - 2x^T A^T b + b^T b\end{aligned}$$

Velja tudi:

$$\|r_\perp\|_2^2 + \sum_i \frac{\alpha^4 \beta_i^2}{(\alpha^2 + \sigma_i^2)^2} = b^T b - \sum_i \beta_i^2 + \sum_i \frac{\alpha^4 \beta_i^2}{(\alpha^2 + \sigma_i^2)^2}.$$

Želimo pokazati da velja:

$$x^T A^T A x - 2x^T A^T b + b^T b = b^T b - \sum_i \beta_i^2 + \sum_i \frac{\alpha^4 \beta_i^2}{(\alpha^2 + \sigma_i^2)^2}.$$

To velja natanko tedaj, ko velja:

$$x^T A^T A x - 2x^T A^T b = - \sum_i \beta_i^2 + \sum_i \frac{\alpha^4 \beta_i^2}{(\alpha^2 + \sigma_i^2)^2}.$$

Sedaj vstavimo x iz (1) v zgornjo enačbo in dobimo:

$$\sum_i \frac{\sigma_i^4 \beta_i^2}{(\sigma_i^2 + \alpha^2)^2} - 2 \sum_i \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \beta_i^2 = - \sum_i \beta_i^2 + \sum_i \frac{\alpha^4 \beta_i^2}{(\alpha^2 + \sigma_i^2)^2}.$$

Enačbo pogledamo pri i -tem indeksu:

$$\begin{aligned}\frac{\sigma_i^4 \beta_i^2}{(\sigma_i^2 + \alpha^2)^2} - 2 \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \beta_i^2 &= -\beta_i^2 + \frac{\alpha^4 \beta_i^2}{(\alpha^2 + \sigma_i^2)^2} \cdot (\alpha^2 + \sigma_i^2)^2 \\ \sigma_i^4 \beta_i^2 - 2 \sigma_i^2 \beta_i^2 (\sigma_i^2 + \alpha^2) &= -\beta_i^2 (\sigma_i^2 + \alpha^2)^2 + \alpha^4 \beta_i^2,\end{aligned}$$

kar pa vidimo da drži. S tem smo pokazali formuli za $\eta(\alpha)$ in $\rho(\alpha)$. Sedaj izračunajmo še odvode, ki smo jih uporabili pri funkciji *kappa.m*. Velja:

$$\begin{aligned}\eta(\alpha)' &= \sum_i \frac{-4\alpha\sigma_i^2\beta_i^2}{(\alpha^2 + \sigma_i^2)^3} \\ \eta(\alpha)'' &= \sum_i \frac{-4\beta_i^2\sigma_i^2(\sigma_i^2 - 5\alpha^2)}{(\alpha^2 + \sigma_i^2)^4} \\ \rho(\alpha)' &= \sum_i \frac{4\alpha^3\beta_i^2\sigma_i^2}{(\alpha^2 + \sigma_i^2)^3} \\ \rho(\alpha)'' &= \sum_i \frac{12\beta_i^2\sigma_i^2(\alpha^2\sigma_i^2 - \alpha^4)}{(\alpha^2 + \sigma_i^2)^4}.\end{aligned}$$

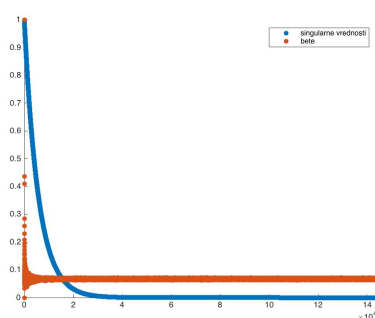
Za izračun ukrivljenosti uporabimo zvezi:

$$\begin{aligned}\hat{\rho}' &= \frac{\rho'}{\rho}, \hat{\eta}' = \frac{\eta'}{\eta}, \\ \hat{\rho}' &= \frac{\rho'' - \rho'^2}{\rho^2}, \hat{\eta}' = \frac{\eta'' - \eta'^2}{\eta^2}.\end{aligned}$$

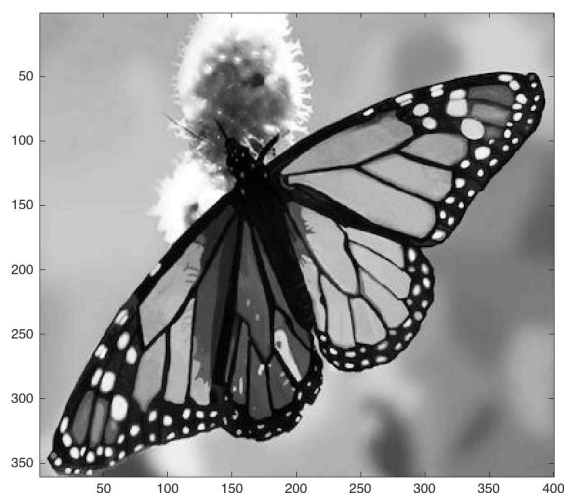
3 Obdelava podatkov

Funkcija, ki poišče najboljši α je v datoteki *najboljsiAlpha.m*. Funkcija, ki izračuna približen α je v datoteki *priblizniAlpha.m*. Algoritem za izračun mesta odreza je v datoteki *izracunajMestoOdreza.m*. Idejo za algoritem dobimo na podlagi slike 1. Algoritem deluje tako, da izberemo mesto odseka tam, kjer postanejo β_i približno velikosti $\frac{1}{4}\sigma_i$, kjer smo bete normalizirali. Ideja je, da takrat postane šum že dovolj velik v primerjavi z singularnimi vrednostmi in preveč moti člene v vsoti, zato od tam naprej postavimo faktorje filtra na 0.

Sedaj bomo prikazali prvotno sliko, 4 različno zmotene slike ter njihove popravke s tremi različnimi algoritmi. Podali bomo tudi rezultate ostrin, katerih delovanje bomo razložili v naslednjem poglavju. V vseh naslednjih poglavjih bo tam, kjer so slike metulja, veljalo, da je v zgornjem levem kotu zmotena slika \hat{B} , zgoraj desno slika $X1$, ki jo dobimo z regularizacijo Tihonovega, kjer α dobimo z iskanjem minimuma s pomočjo matlabove vgrajene



Slika 1: Singularne vrednosti in bete



Slika 2: Prvotna slika

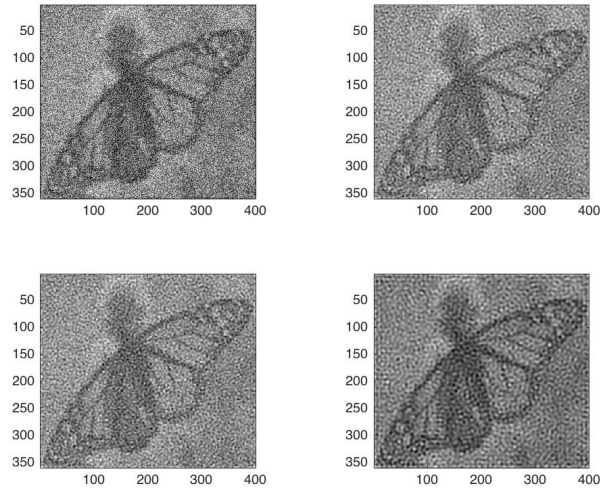
knjižnjice. Spodaj levo je slika X_2 , ki jo dobimo z regularizacijo Tihonovega, kjer α dobimo s približnim izračunom, po algoritmu polinomskih zlepkov. Spodaj desno pa je slika X_3 , ki jo dobimo s pomočjo odrezanega singularnega razcepa, kjer mesto odreza izberemo z algoritmom, opisanim zgoraj. GRAT in LAPD prikazujeta dve različni oceni ostrin slik, ki smo ju uporabili, podrobnejša razlaga, je v naslednjem poglavju. Med sabo nista primerljivi. Velja omeniti, da pri obeh metodah manjša številka pomeni manjšo prisotnost šuma. Za prvotno sliko velja: GRAT = 0.0597 in LAPD = 0.1478. V primerih je s faktor zamegljevanja, ki ga potrebuje knjižnjica HNO, e pa je faktor šuma, s katerim pomnožimo šum $rand(m, n)$.

3.1 Parametri $s = 2, e = 2$

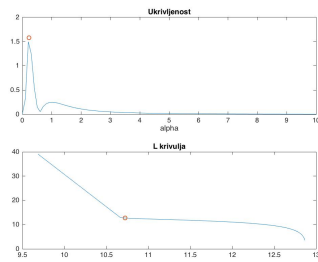
Optimalni $\alpha = 0.230406149323062$. Približno izračunani $\alpha = 0.227031955843888$.

Tabela 1: Ostrina slik

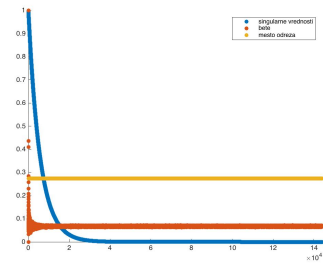
	\hat{B}	$X1$	$X2$	$X3$
GRAT	0.9214	0.2512	0.2559	0.1737
LAPD	3.9822	0.6447	0.6586	0.3381



Slika 3: Parametri: $s = 2, e = 2$



(a) Parametri: $s = 2, e = 2$



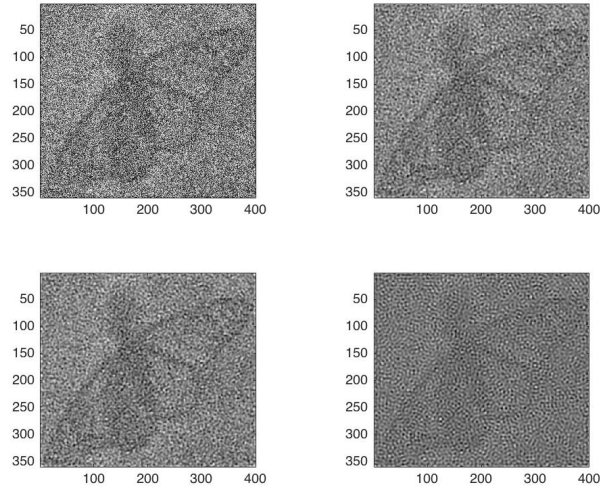
(b) Parametri: $s = 2, e = 2$

3.2 Parametri $s = 2, e = 5$

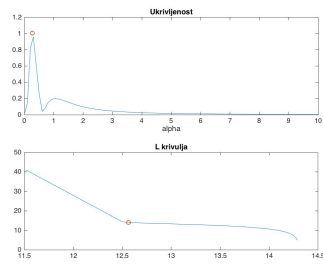
Optimalni $\alpha = 0.267700394904301$. Približno izračunani $\alpha = 0.279591217686607$.

Tabela 2: Ostrina slik

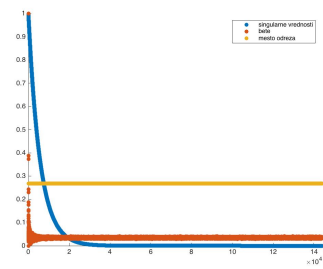
	\hat{B}	$X1$	$X2$	$X3$
GRAT	2.3002	0.5079	0.4798	0.9672
LAPD	9.9548	1.2828	1.2022	2.3325



Slika 5: Parametri: $s = 2, e = 5$



(a) Parametri: $s = 2, e = 5$



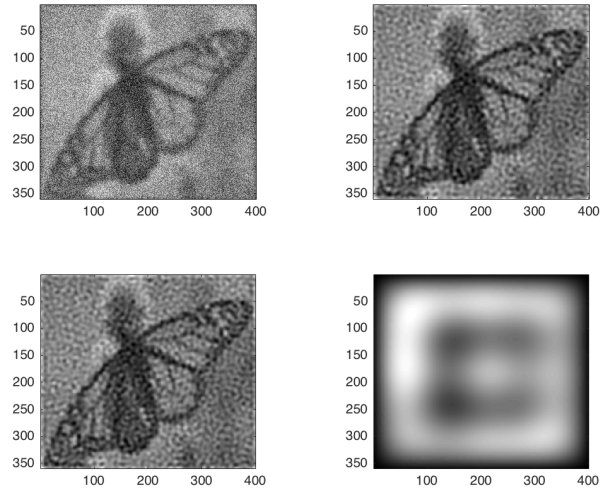
(b) Parametri: $s = 2, e = 5$

3.3 Parametri $s = 4, e = 1$

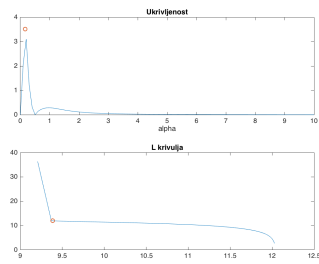
Optimalni $\alpha = 0.162161879812697$. Približno izračunani $\alpha = 0.148268295601612$.

Tabela 3: Ostrina slik

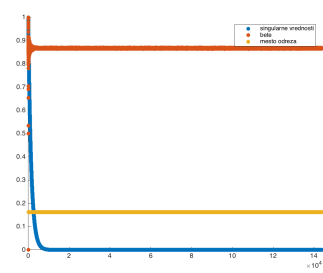
	\hat{B}	$X1$	$X2$	$X3$
GRAT	0.4614	0.0614	0.0671	0.0150
LAPD	1.9910	0.0798	0.0893	0.0021



Slika 7: Parametri: $s = 4, e = 1$



(a) Parametri: $s = 4, e = 1$



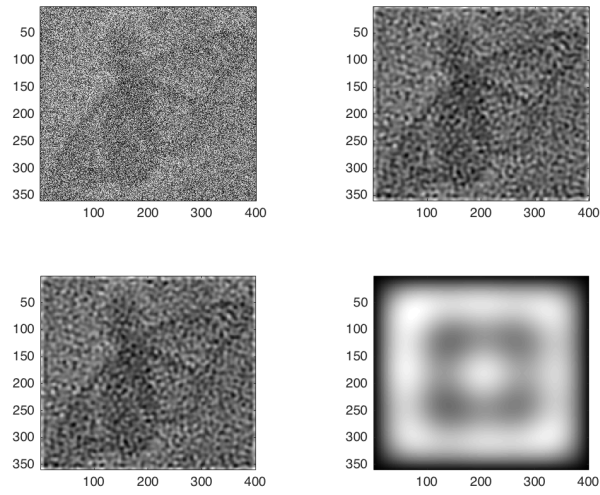
(b) Parametri: $s = 4, e = 1$

3.4 Parametri $s = 4, e = 5$

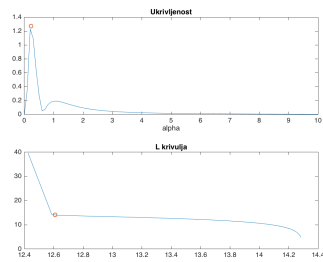
Optimalni $\alpha = 0.230958185004528$. Približno izračunani $\alpha = 0.230009259071327$.

Tabela 4: Ostrina slik

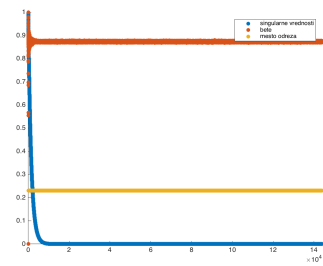
	\hat{B}	$X1$	$X2$	$X3$
GRAT	2.2998	0.1774	0.1783	0.0416
LAPD	9.9547	0.2283	0.2296	0.0058



Slika 9: Parametri: $s = 4, e = 5$



(a) Parametri: $s = 4, e = 5$



(b) Parametri: $s = 4, e = 5$

4 Ostrina

Za ostrino slike sem uporabil že napisano funkcijo, ki sem jo našel na [1]. Uporabil sem dve metodi, GRAT temelji na gradientu slike, ki izračuna odvod slike. Ideja je, da ima ostra slika bolj ostre robove. Druga metoda, ki sem jo uporabil pa je LAPD, ki poleg prvih odvodov opazuje tudi druge odvode. Več se lahko prebere v [2]. Ideja, da gledamo, samo drugo normo razlike med slikama, ni dobra, saj lahko zelo pokvarimo majhno število elementov v matriki, tako da se slika praktično ne spremeni, sprememba v drugi normi, pa je precej velika. Po drugi strani lahko elemente za malo spremenimo tako, da ima zmotena matrika podobno normo, vendar je slika popačena.

Definicija 1 Lokalni linearni operator *je definiran kot konvolucija slike I v $p = (x, y)$ s filtrom W :*

$$J(p) = I * W(p) = \frac{1}{S} \sum_{i=-k}^k \sum_{j=-k}^k w_{i,j} I(x+i, y+j),$$

kjer so $w_{i,j}$ uteži in $S > 0$ faktor skaliranja.

Odvod slike dobimo tako, da se z ustreznim filtrom zapeljemo čez sliko, ter tako dobimo novo tabelo števil. Primer matrike, s katero izvedemo filtriranje je npr.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

S tako matriko potem množimo podmatrike začetne slike. Temu postopku pravimo filtriranje. Na podoben način deluje algoritem LAPD. Algoritem GRAT pa deluje na osnovi razlik med posamezni piksli.

5 Komentar

Opazimo, da regularizacija Tihonovega v večini primerov da na oko boljše rezultate. Edini primer, ko vrne primerljive rezultate je pri paramterih $s = 2$ in $e = 2$. Pri ostalih pa je regularizacija Tihonovega precej boljša, vsaj v primerjavi z mojim algoritmom. Tudi če iščemo faktor odreza na roke, se odrezani singularni razcep na tej sliki ne more primerjati z regularizacijo Tihonova.

Iskanje optimalnega α smo z metodo polinomskih zlepkov precej pohitrili, hkrati pa so slike praktično iste v primerjavi s tistimi, kjer vzamemo α , ki ga izračuna matlabova vgrajena funkcija.

Za iskanje mesta odreza nam algoritem deluje dobro, če sliko malo popačimo. Z večanjem šuma pa postane algoritem občutno slabši.

Literatura

- [1] <https://www.mathworks.com/matlabcentral/fileexchange/27314-focus-measure>
- [2] <https://sites.google.com/view/cvia/focus-measure>