



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Řešení optimalizační úlohy LASSO pomocí proximálních algoritmů

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie
Autor práce: **Václav Langr**
Vedoucí práce: doc. Ing. Zbyněk Koldovský, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Solution to LASSO Using Proximal Algorithms

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology

Author: **Václav Langr**
Supervisor: doc. Ing. Zbyněk Koldovský, Ph.D.



Tento list nahrad'te
originálem zadání.

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Chtěl bych tímto poděkovat všem, kteří mne podporovali. Především mé rodině za velikou podporu, díky které mi bylo umožněno studovat.

Zároveň bych chtěl poděkovat také vedoucímu této práce doc. Ing. Zbyňku Koldovskému, Ph.D. za velice přínosné konzultace, rady a trpělivost.

Abstrakt

Tato bakalářská práce je zaměřena na rekonstrukci řídkého vektoru z jeho komprimovaného pozorování. Pro rekonstrukci se využívá optimalizačního problému LASSO a jeho řešení pomocí proximálních algoritmů. Po vytvoření takového algoritmu, který je schopen původní signál rekonstruovat, se využívá metody Monte Carlo pro simulaci algoritmu na větším souboru dat. Pro takto získaný výpočet je zjištěna kvadratická chyba řešení LASSO od původního vektoru dat, která je následně porovnávána s teoretickou chybou.

Vypracování této bakalářské práce bylo rozděleno do několika navazujících částí. Prvním a také nejdůležitějším krokem bylo nastudování vlastností proximálních algoritmů a výpočet proximálního operátora při různých vstupních funkcích. Po takto provedené rešerši proximálních algoritmů proběhla také rešerše vlastností optimalizační úlohy LASSO. Následně bylo možné přistoupit k implementaci algoritmu v programovacím jazyce a vývojovém prostředí MATLAB. Při postupné implementaci byl algoritmus upraven tak, aby vždy dokonvergoval ke správnému nebo alespoň přibližnému řešení optimalizačního problému. Z tohoto důvodu byl algoritmus rozšířen o podmínky optimality, jež ukončují výpočet při dosažení poměrně přesné aproximace. Dále byl rozšířen o výpočet dynamické velikosti kroku. S takto připraveným algoritmem mohla být vytvořena metoda Monte Carlo, která generuje nekomprimovaný řídký vektor dat, měřící matice, jejichž prvky mají Gaussovo rozložení, a parametr λ v zadaném rozsahu s logaritmickým rozdělením.

Výsledek této práce může být využit pro další zpracování. Jedním z možných použití je např. pro vytvoření nového datového formátu, ve kterém by byl uložen jen komprimovaný vektor dat případně i měřící matice, pokud by nebyla shodná pro všechny komprimace. Na straně klienta by byl tedy pouze zrekonstruován původní nekomprimovaný signál.

Klíčová slova: MATLAB, proximální algoritmus, proximální operátor, LASSO, Monte Carlo

Abstract

This bachelor thesis is focused on reconstruction of sparse vector from his compressed observation. For the reconstruction is used the LASSO problem and its solution using proximal algorithms. After creation of an algorithm that is able to restore the original signal is used Monte Carlo method for simulation of the algorithm with bigger set of data. Then is calculated the squared error for the found solution and the original data that is compared with the theoretical error.

Realization of this bachelor thesis was divided into several parts. The very first and the most important step was studying the properties of proximal algorithms and evaluation of proximal operator for different functions. After the research on proximal algorithms there was also research on the properties of the LASSO. After that is was possible to implement algorithm using MATLAB language and development environment. The algorithm was modified during the implementation so it always converges into correct or at least approximate solution of LASSO. Due to this reason optimality conditions were added that terminates solving if the approximation is very accurate and a computation of dynamical step size. This prepared algorithm could be used for creation of Monte Carlo method that generates uncompressed sparse vector of data, random measurement matrix with Gaussian distribution and a lambda parameter within an interval with logarithmic distribution.

The result of this thesis can be used for another usage. One of the possible usage is for example creation of a new data format. The compressed data would be saved in this format and only if the measurement matrix is not the same for all compressions it would be also saved. On the client side the original uncompressed signal would be recovered.

Keywords: MATLAB, Proximal Algorithm, Proximal Operator, LASSO, Monte Carlo

Obsah

1	Úvod	11
2	Optimalizační problémy	13
2.1	Optimalizační problém LASSO	13
3	Proximální algoritmy	15
3.1	Proximální operátor	15
4	Dopředno-zpětný algoritmus	16
4.1	Varianta s pevnou délkou kroku	17
4.2	Dynamická délka kroku	17
4.3	Podmínky optimality	17
4.4	Konečná verze algoritmu	17
5	Monte Carlo simulace	19
6	Závěr	20

Seznam obrázků

2.1	Graf $ x $	14
3.1	Průběh zvoleného proximálního operátoru pro $\lambda = 0.5$	15
4.1	Schéma dopředno-zpětného algoritmu	16
4.2	Ukázka průběhu algoritmu při různé volbě délky kroku	17
4.3	Ukázka průběhu algoritmu s dynamickou délkou kroku	18

Seznam tabulek

1 Úvod

V dnešním světě, kdy získáváme stále více důležitých a velkých dat je stále důležitější nějakým způsobem získaná data komprimovat. Z tohoto důvodu se tato práce zabývá komprimací dat a následnou rekonstrukcí u koncového uživatele. Bude pro to použito nového přístupu pomocí proximálních algoritmů, které jsou rychlé a jednoduché na implementaci.

Základní měření odpovídá funkci, viz Vzorec 1, kde jsou původní data $x_0 \in \mathbb{C}^n$ transformovány měřicí maticí $A \in \mathbb{C}^{m \times n}$, kde m je počet řádků a n počet sloupců matice a v poslední řadě je nutné uvažovat i náhodná data $z \in \mathbb{C}^m$, v našem případě toto reprezentuje náhodný šum. Takto jsou získaná nová data $y \in \mathbb{C}^m$. V obecných případech platí, že $m \geq n$, ale takto nedochází k žádné komprimaci dat případně by i vzrostl počet dat. Následná rekonstrukce by se tak redukovala pouze na spočtení inverzní, v případě obdélníkové matice pseudoinverzní, matice A^{-1} a spočtení původních dat x_0 . Nicméně, tato bakalářská práce se zabývá případy, kdy $m < n$ a nelze tedy použít pseudoinverzní matici.

$$y = A \times x_0 + z \quad (\text{Vzorec 1})$$

Dle pravidel lineární algebry tak existuje nekonečně mnoho řešení a je tak nemožné v tomto případě zrekonstruovat původní data. Nicméně ve skutečnosti lze data zrekonstruovat a to za podmínky, že původní vektor dat x_0 je řídký, tedy většina prvků z x_0 je rovno 0. Právě z tohoto důvodu jsou kompresní algoritmy, například pro kompresi formátu JPEG kdy se ukládají pouze největší koeficienty diskrétní kosinové transformace, tak efektivní. Toto ale není jedinná podmínka, nutná k úspěšné re-

konstrukci. Další podmínkou je správná volba měřicí matice A . Například, při volbě diagonální jednotkové matice, by komprimovaná data y byla z velké části nulová. Z tohoto důvodu by pak bylo nemožné rekonstruovat původní data.

2 Optimalizační problémy

Veškeré další řešení je založeno na tom, že je lze zapsat v neomezeném tvaru. V tomto tvaru se minimalizuje obecný vstup x přes součet m konvexních funkcí $f_0 \dots f_m$, které náležejí množině \mathbb{R}^n . Takto zapsaný problém by byl definován jako:

$$\min_{x \in \mathbb{R}^n} f_0(x) + \dots + f_m(x) \quad (\text{Vzorec 2})$$

Nicméně, jelikož lze uvažovat jakoukoliv výše zmíněnou funkci, naskytá se tu typický problém. Některé funkce nejsou diferenciovatelné. Z tohoto důvodu se dále budou tyto funkce rozdělovat a řešit jednotlivě.

2.1 Optimalizační problém LASSO

Problém rekonstrukce původních dat se vyskytuje v nejrůznějších oborech. Mezi tyto obory patří strojové učení, zpracování signálů a další. Jeden z možných přístupů je využít minimalizace a to ve tvaru, který popisuje Vzorec 3. Funkce $f(x)$ ve vzorci je libovolná a volí se podle toho, jaké vlastnosti mají vstupní data.

$$\operatorname{argmin}_x \{ \|y - A \times x\|_2^2 + \lambda \times f(x) \} \quad (\text{Vzorec 3})$$

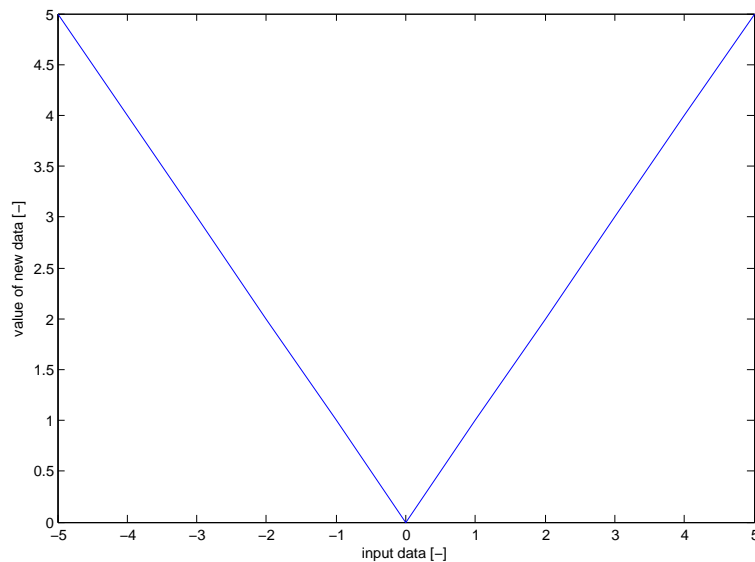
V případě řídkého vektoru se volí funkce $\|x\|_1$ a vyniká tak vztah popsany níže. Tento nový vztah zároveň řeší dva problémy a to rekonstrukci původních dat

x_0 a jejich odšumnění.

$$\underset{x}{\operatorname{argmin}} \left\{ \|y - A \times x\|_2^2 + \lambda \times \|x\|_1 \right\} \quad (\text{Vzorec 4})$$

Protože hodnota pro $l1$ normu je suma absolutních hodnot prvků, definice pro obecnou p normu je popsána Vzorec 5. Pro lepší ilustraci je tato funkce zobrazena v 2.1. Z tohoto zobrazení je patrné, že tato funkce má nekonečně mnoho derivací v bodě $[0, 0]$ a nelze tak spočítat její derivaci.

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (\text{Vzorec 5})$$



Obrázek 2.1: Graf $|x|$

3 Proximální algoritmy

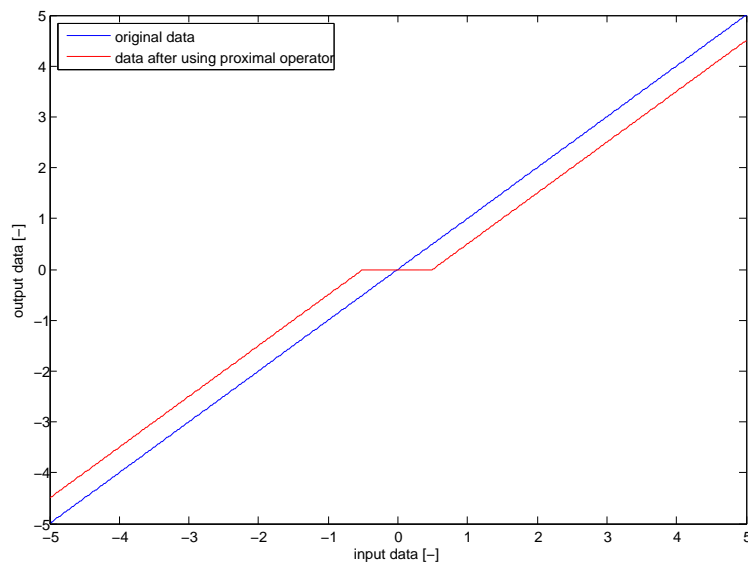
Jednou z možností, jak vyřešit problém zmíněný v předchozí kapitole, je využít proximální algoritmy a proximální operátory příslušející zadaným funkcím. Proximální algoritmus je:

$$x_{n+1} = \text{prox}_{\lambda \times f}(x_n) \quad (\text{Vzorec 6})$$

V tomto vzorci je f uzavřená konvexní funkce, která splňuje $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$. Pokud f má minimum, pak x v n -té iteraci konverguje ke správnému řešení.

3.1 Proximální operátor

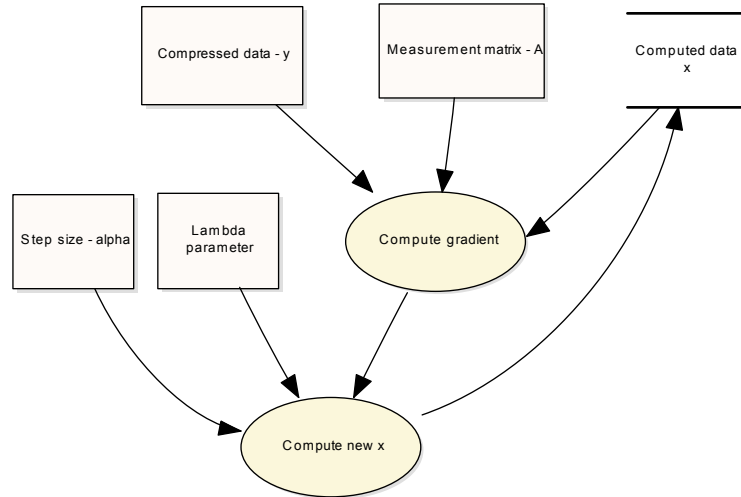
Právě v těchto algoritmech se lze velice často setkat s tzv. proximálním operátorem.



Obrázek 3.1: Průběh zvoleného proximálního operátoru pro $\lambda = 0.5$

4 Dopředno-zpětný algoritmus

Jako hlavní algoritmus, na který se tato práce zaměřuje, je dopředo-zpětný algoritmus. Jeho základní varianta, popsaná obrázkem 4.1, používá pevnou délku kroku. Je to iterační algoritmus, který získá na vstupu komprimovaná data y , měřicí matici A , velikost kroku α a parametr λ , který si volí uživatel.



Obrázek 4.1: Schéma dopředno-zpětného algoritmu

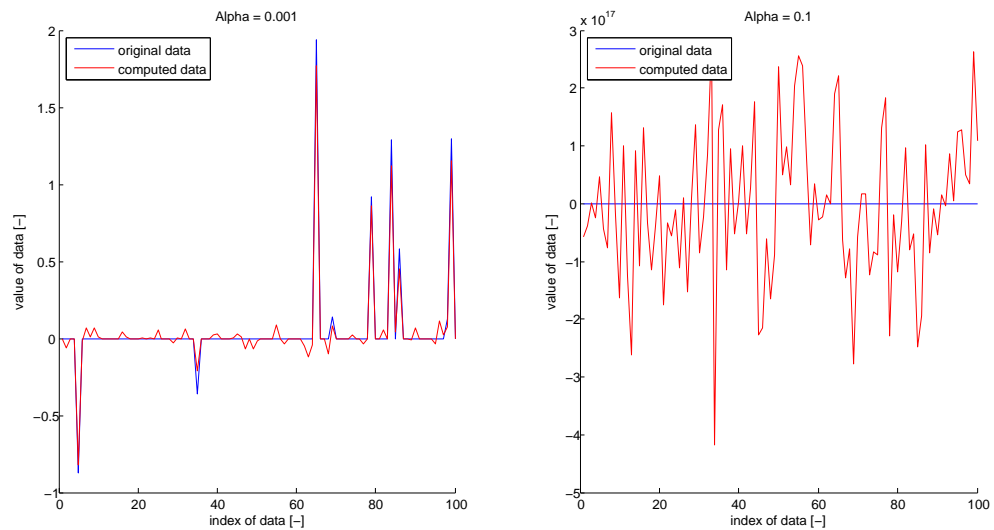
Za pomoci těchto údajů se v každém kroku počítá gradient diferenciovatelné části optimalizační úlohy LASSO, jehož předpis funkce je Vzorec 7.

$$\partial f(x) = -2 \times A^T \times (y - A \times x) \quad (\text{Vzorec 7})$$

Poté se již spočítá nový vektor dat x za pomoci proximálního operátoru, jehož průběh je znázorněn v 3.1, dle vzorce viz Vzorec 8.

$$x_{n+1} = \text{prox}_{\lambda \times \alpha_n}(x_n - \alpha_n \times \partial f(x_n)) \quad (\text{Vzorec 8})$$

4.1 Varianta s pevnou délkou kroku

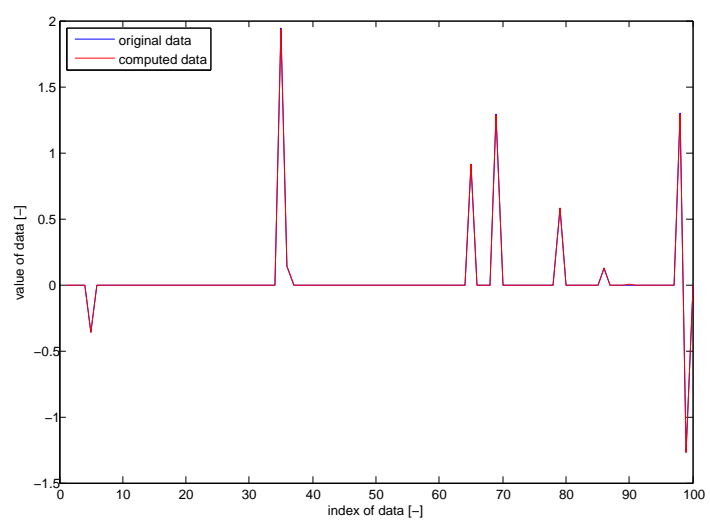


Obrázek 4.2: Ukázka průběhu algoritmu při různé volbě délky kroku

4.2 Dynamická délka kroku

4.3 Podmínky optimality

4.4 Konečná verze algoritmu



Obrázek 4.3: Ukázka průběhu algoritmu s dynamickou délkou kroku

5 Monte Carlo simulace

6 Závěr

Seznam použité literatury

- [1] BOYD, Stephen P a Lieven VANDENBERGHE. Convex optimization. Seventh printing with corrections. New York: Cambridge University Press, 2004. ISBN 05-218-3378-7.