

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



BAKALÁRSKA PRÁCA

Bratislava 2011

Roman Kukumberg

Proximal-gradient, metóda konvexného programovania

BAKALÁRSKA PRÁCA

Roman Kukumberg

**UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ MATEMATIKY A ŠTATISTIKY**

9.1.9 aplikovaná matematika
ekonomická a finančná matematika

Vedúci bakalárskej práce:
RNDr. Mária Trnovská, PhD.

Kód práce:
6a1b8de2-2295-47e0-8b36-0b2b45bd12eb

BRATISLAVA 2011



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Roman Kukumberg
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský

Názov : Proximal-gradient metóda konvexného programovania
Cieľ : Teoretické a praktické spracovanie proximal-gradient metódy.
Literatúra : [1] L. Vandenberghe - Convex Optimization - course - lecture notes.
Anotácia : Praca má obsahovať teoretický popis metódy, analýzu konvergenzie, otestovanie metódy na príkladoch .

Kľúčové slová : gradientné metódy, konvexné programovanie

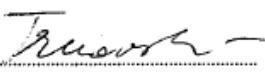
Vedúci : RNDr. Mária Trnovská, PhD.

Dátum zadania: 05.11.2010

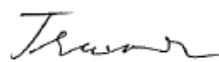
Dátum schválenia: 08.11.2010

.....
doc. RNDr. Margaréta Halická, CSc.
garant študijného programu


.....
študent


.....
vedúci práce

Dátum potvrdenia finálnej verzie práce, súhlas s jej odovzdaním (vrátane spôsobu prístupnenia)

2.6.2011 
.....
vedúci práce

Čestné prehlásenie

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s využitím teoretických vedomostí a s použitím uvedenej odbornej literatúry.

Bratislava, 2. jún 2011

.....

Roman Kukumberg

Pod'akovanie

Aj touto cestou by som sa chcel poďakovať svojej vedúcej RNDr. Márii Trnovskej, PhD. za cenné rady a pripomienky pri písaní tejto bakalárskej práce.

Abstrakt

KUKUMBERG, Roman: Proximal-gradient, metóda konvexného programovania [Bakalárska práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; vedúci práce: RNDr. Mária Trnovská, PhD., Bratislava, 2011

Bakalárska práca sa zaoberá tzv. proximal gradient metódou konvexného programovania, ktorá slúži na optimalizáciu konvexných nediferencovateľných funkcií ako neohraničeného problému (čiže bez väzieb). Základným prvkom metódy je tzv. proximal operátor, podľa ktorého je metóda pomenovaná. Zaoberáme sa s rôznymi modifikáciami metódy ako je zrýchlená proximal gradient metóda a spádová proximal gradient metóda. Práca obsahuje taktiež analýzu konvergenzie pôvodnej aj zrýchlenej metódy, kde sa ukáže, že obe konvergujú rýchlejšie ako klasické subgradientné metódy.

Kľúčové slová: gradientné metódy, konvexné programovanie, konvexné nediferencovateľné funkcie, subgradients, proximal operátor, proximal gradient metóda

Abstrakt

KUKUMBERG, Roman: Proximal-gradient method of convex optimization [Bachelor's thesis], COMENIUS UNIVERSITY; Faculty of Mathematics, Physics and Informatics; Department of Applied Mathematics and Statistics ;
thesis supervisor: RNDr. Mária Trnovská, PhD., Bratislava, 2011

Bachelor's thesis deals with the so-called proximal gradient method of convex optimization, which is used for optimization of convex nondifferentiable functions as unconstrained problem (without bonds). An essential element of the method is called proximal operator, after which is the method called. We deal with various modifications of the method such as accelerated proximal gradient method and descent proximal gradient method. The thesis also includes convergence analysis of the classic and accelerated method, which shows that both of them converge faster than subgradient methods.

Key words: gradient methods, convex optimization, convex nondifferentiable functions, subgradients, proximal operator, proximal gradient method

Obsah

1	Úvod	1
2	Konvexné programovanie	4
2.1	Konvexné funkcie	4
2.2	Úloha konvexného programovania	7
2.3	Subgradienty	8
2.4	Subgradientné metódy konvexného programovania	13
3	Proximal-gradient metóda	15
3.1	Proximal operátor	15
3.2	Algoritmus proximal-gradient metódy	22
3.3	Konvergencia proximal-gradient metódy	24
4	Zrýchlená proximal gradient metóda	29
4.1	Konvergencia zrýchlenej proximal gradient metódy	30
4.2	Descent verzia zrýchlenej proximal gradient metódy	34
4.3	Nová modifikácia proximal gradient metódy	34
5	Numerické experimenty	35
5.1	Vplyv počtu nenulových prvkov v optimálnom riešení na rýchlosť dosiahnutia riešenia	37
5.2	Vplyv zmeny požadovanej presnosti na rýchlosť dosiahnutia riešenia	39
5.3	Vplyv rozmerov úlohy na rýchlosť dosiahnutia riešenia	40
5.4	Vplyv vzdialenosti štartovacieho bodu od optima na rýchlosť dosiahnutia riešenia	41
5.5	Porovnanie metód	42
6	Záver	48
7	Literatúra	49
8	Príloha - zdrojové kódy programov	50

1 Úvod

V optimalizácii neexistujú optimálne metódy na riešenie úloh. Často sa stretávame s potrebou riešiť úlohy s nediferencovateľnými konvexnými funkciami. Jednou z možností ako dané úlohy riešiť sú tzv. subgradientné metódy konvexného programovania, ktoré ako smer optimalizácie využívajú subgradients. Subgradientné metódy majú však mnoho nevýhod, sú veľmi pomalé a nemajú dobré ukončovacie kritériá. Tieto metódy nie sú spádovými metódami, nerastúcosť účelovej funkcie nie je zaručená. Väčšina spôsobov voľby kroku taktiež nezaručuje konvergenicu týchto metód.

Vzhľadom na nevýhody subgradientných metód boli vytvorené iné metódy na riešenie úloh konvexného programovania s nediferencovateľnými funkciami. Medzi tieto metódy patrí aj proximal gradient metóda. Proximal gradient metóda rieši úlohy, ktoré majú špeciálnu štruktúru $\min_x f(x) = g(x) + h(x)$, kde sa účelová funkcia dá rozdeliť na konvexnú diferencovateľnú časť $g(x)$ a konvexnú nediferencovateľnú časť $h(x)$. Proximal gradient je spádová metóda, hodnota účelovej funkcie po každom kroku klesá alebo ostáva rovnaká.

Klasické subgradientné metódy majú zložitosť $O(1/\epsilon^2)$, čo znamená, že potrebujeme $O(1/\epsilon^2)$ iterácii aby sa našiel ϵ -suboptimálny bod spĺňajúci $f(x^{(k)}) - \hat{f} \leq \epsilon$. Proximal gradient metóda má v oboch prípadoch voľby kroku lepšiu zložitosť $O(1/\epsilon)$, stačí nám menej iterácii na dosiahnutie tej istej presnosti ϵ . Avšak proximal gradient metóda sa dá vylepšiť, touto úpravou dosahuje ešte lepšiu zložitosť $O(1/\sqrt{\epsilon})$. Upravený algoritmus sa nazýva zrýchlená proximal gradient metóda alebo proximal gradient metóda s extrapoláciou.

Zrýchlená metóda však nemá vlastnosť nerastúcosť účelovej funkcie $f(x^{(k)}) \leq f(x^{(k-1)})$ tak ako pôvodná verzia algoritmu. Z uvedeného dôvodu existuje ďalšia modifikácia proximal gradient metódy, ktorá túto vlastnosť spĺňa. Nazýva sa spádová (descent) verzia zrýchlenej proximal gradient metódy. Vlastnosť nerastúcosť účelovej funkcie spĺňa taktiež nová modifikácia proximal gradient metódy, ktorú navrhne s účelom ďalšieho vylepšenia metódy.

V reálnom svete nachádza proximal gradient metóda využitie na odšumovanie signálu, kompresiu a zaostrovanie (deblur) obrázkov a fotiek. Proximal gradient metóda a jej

modifikácie majú dostatok presnosti a rýchlosti pre riešenie veľkorozmerných problémov.

Práca obsahuje teoretickú analýzu proximal gradient metódy, pri spracovaní sme čerpali najmä z [1]. Práca pozostáva zo štyroch hlavných častí.

V prvej časti práce zhrnieme základné vedomosti o konvexných funkciách a konvexnom programovaní, následne zdefinujeme subgradients, ktoré sú analógiou ku gradientom pri nediferencovateľných funkciách. Spomenieme ich vlastnosti a podmienky optimality pre nediferencovateľné funkcie, ktoré sú pomocou nich definované.

V ďalšej časti sa oboznámime s proximálnym operátorom, základným prvkom proximal gradient metódy, s jeho vlastnosťami a taktiež so samotnou proximal gradient metódou. Práca takisto obsahuje analýzu konvergenzie, pri spracovaní sme využili hlavne zdroj [1].

V tretej časti sa budeme venovať rôznym modifikáciám metódy, predstavíme si zrýchlenú metódu a vyšetříme jej konvergenciu. Taktiež navrhne novú modifikáciu metódy. V ďalšej kapitole sa venujeme numerickým experimentom s proximal gradient metódou a jej upravenými verziami. Skúmame napríklad vplyv počtu nenulových prvkov v optimálnom riešení na rýchlosť riešenia úloh pomocou danej metódy. Metódy otestujeme aj na veľkorozmerných úlohách.

Notácia:

I_C	-	indikátorová funkcia $I_C(x) = \begin{cases} 0 & x \in C \\ \infty & \text{inak} \end{cases}$
P_C	-	projekcia na množinu C , $P_C(x) = \underset{u \in C}{\operatorname{argmin}} \ u - x\ _2^2$
$D(f)$	-	definičný obor funkcie f
$\nabla f(x)$	-	gradient funkcie f v bode x
$\partial f(x)$	-	subdiferenciál funkcie f v bode x
$\nabla^2 f(x)$	-	Hessova matica funkcie f
$\operatorname{epi}(f)$	-	epigraf funkcie f
$\ x\ _1$	=	$\sum_{i=1}^n x_i $
$\ x\ _2$	-	Euklidovská norma, $\ x\ _2 = \sqrt{\sum_{i=1}^n x_i^2}$
\hat{x}	-	bod optima, optimálne riešenie
\hat{f}	-	optimálna hodnota funkcie f v bode \hat{x}
L	-	Lipschitzovská konštanta
ϵ	-	presnosť
x^+	-	bod nasledujúcej iterácie
x^0	-	počiatočný/štartovací bod
t, t_k	-	dĺžka kroku fixná / v k -tej iterácii
argmin	-	argument minima
$\operatorname{prox}_h(x)$	-	proximálny operátor funkcie h v bode x

2 Konvexné programovanie

V nasledujúcej kapitole zhrnieme základné poznatky o konvexných množinách a funkciách, spomenieme hlavne vlastnosti, ktoré budeme v ďalšom postupe využívať. Taktiež sformulujeme základnú úlohu konvexného programovania. Kapitola bola spracovaná na základe poznatkov získaných zo zdrojov [1], [2], [4] a [5].

2.1 Konvexné funkcie

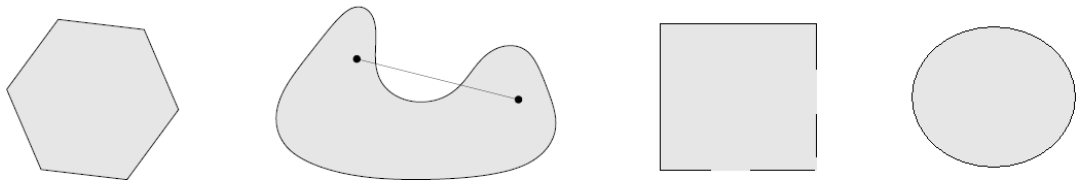
Definícia 2.1. Množina $C \subseteq R^m$ sa nazýva konvexná, ak pre každé dva body x, y patriace množine C obsahuje aj celú úsečku $[x, y]$ medzi nimi.

Poznámka 2.1. Ekvivalentne: Množina C sa nazýva konvexná, ak pre každé dva body $x, y \in C$ obsahuje aj ich každú konvexnú kombináciu.

Teda $\lambda x + (1 - \lambda)y \in C$ pre $0 \leq \lambda \leq 1$.

Vlastnosti konvexných množín:

- Prienik konvexných množín je konvexná množina.
- Lineárna transformácia konvexnej množiny je konvexná množina.

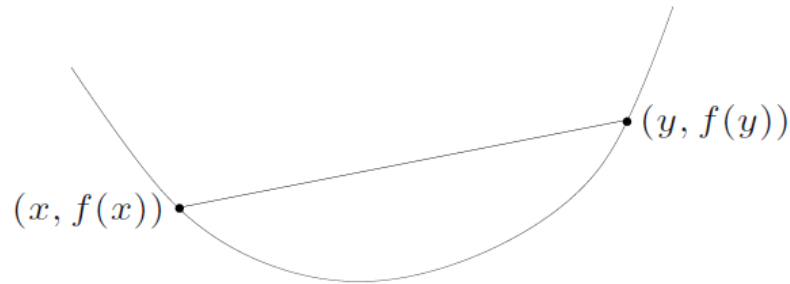


Obr. 1: Definícia 1: konvexná množina, nekonvexná množina, nekonvexná množina, konvexná množina

Definícia 2.2. Funkciu n -premenných $f(x)$ definovanú na konvexnej množine $C \subseteq R^n$ nazývame konvexnou funkciou, ak pre každú dvojicu rôznych bodov $x, y \in C$ a ľubovoľné číslo $0 \leq \lambda \leq 1$ platí:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

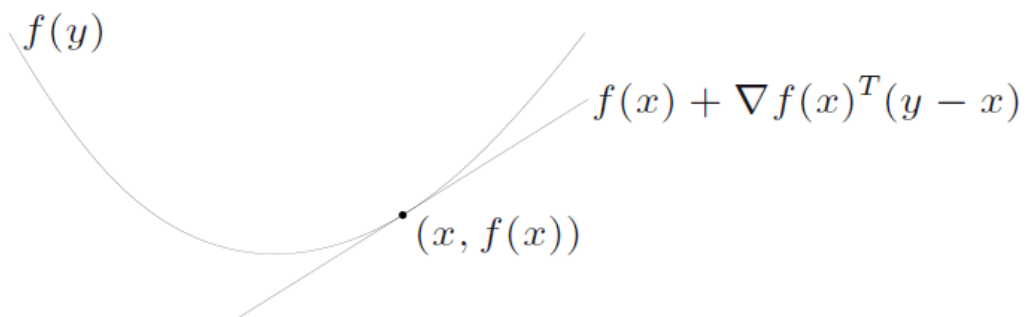
Príklad 2.1. Indikátorová funkcia konvexnej množiny C definovaná ako $I_C = \begin{cases} 0 & x \in C \\ \infty & \text{inak} \end{cases}$ je konvexná funkcia.



Obr. 2: Definícia 2

Veta 2.1. (podmienky 1.rádu): Nech $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ má spojité prvé parciálne derivácie. Potom funkcia $f(x)$ je konvexná práve vtedy, keď platí:

$$\forall y \neq x : f(y) - f(x) \geq \nabla f(x)^T (y - x) \quad (1)$$



Obr. 3: Veta (podmienky 1.rádu)

Veta 2.2. (podmienky 2.rádu): Nech $f(x) : R^n \rightarrow R$ má spojité druhé parciálne derivácie. Potom funkcia $f(x)$ je konvexná práve vtedy, keď platí $\forall x : \nabla^2 f(x)$ je kladne semidefinitná matica.

Definícia 2.3. Funkcia $f(x) : R^n \rightarrow R$ sa nazýva silnokonvexná, ak existuje $\beta > 0$ také, že pre všetky $x \neq y \in R^n$, $0 \leq \lambda \leq 1$ platí:

$$f(\lambda x + (1 - \lambda)y) + \beta \lambda(1 - \lambda)\|x - y\|^2 \leq \lambda f(x) + (1 - \lambda)f(y)$$

Veta 2.3. Nech $f(x) : R^n \rightarrow R$ má spojité druhé parciálne derivácie. Potom funkcia $f(x)$ je silnokonvexná práve vtedy, ak existuje $\beta > 0$ tak, že pre všetky $x, s \in R^n$ platí:

$$s^T \nabla^2 f(x) s > \beta s^T s$$

Vlastnosti silnokonvexných funkcií:

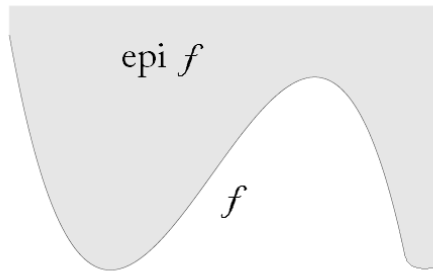
- Ak $f(x)$ je konvexná funkcia a $h(x) = \alpha \|x\|_2^2$, tak ich súčet $F(x) = f(x) + h(x)$ je silnokonvexnou funkciou.
- Množina $S_b = \{x \in R^n | f(x) \leq b\}$ je konvexná, uzavretá a ohraničená.
- Silnokonvexná funkcia má vždy práve jedno minimum.

Definícia 2.4. Epigraf funkcie $f(x) : R^n \rightarrow R$ definujeme ako:

$$epi(f) = \{(x, t) | x \in D(f), f(x) \leq t\},$$

čo je podmnožina R^{n+1} . (Epi znamená "nad", a preto epigraf znamená "nad grafom".)

Definícia 2.5. Konvexná funkcia sa nazýva uzavretá, ak má uzavretý konvexný epigraf.



Obr. 4: Definícia 4 - Epigraf funkcie f

2.2 Úloha konvexného programovania

Za základnú úlohu konvexného programovania možno považovať úlohu tvaru:

$$\begin{aligned} \min_x \quad & f_0(x) \\ & f_i(x) \leq 0 \quad i = 1, \dots, m \end{aligned}$$

kde funkcie $f_i(x) : \mathbb{R}^n \rightarrow \mathbb{R} \quad i = 0, \dots, m$ sú konvexné, spĺňajú Definíciu 2.2.

Z toho vyplýva, že oblasť, na ktorej hľadáme minimum (optimalizujeme), je konvexná. Úloha konvexného programovania sa preto môže charakterizovať ako hľadanie minima konvexných funkcií na konvexných množinách.

Výhodou konvexného programovania je fakt, že každé lokálne minimum účelovej funkcie $f_0(x)$ je zároveň globálnym minimom a množina všetkých miním je konvexná. Ďalšou výhodou je fakt, že optimalizujeme na konvexnej množine, čo umožňuje prechod z ľubovoľného bodu x množiny do akéhokoľvek iného jej bodu po úsečke.

2.3 Subgradienty

Ako vieme z predchádzajúcej kapitoly, úlohou konvexného programovania je nájsť minimum konvexných funkcií. Pre diferencovateľné konvexné funkcie poznáme podmienky optimality a aj mnohé metódy konvexného programovania na ich optimalizáciu. Často však treba optimalizovať nediferencovateľné funkcie. Pri nediferencovateľných funkciách existuje analógia k gradientom zvaná subgradienty, ktorým sa budeme v tejto kapitole venovať. Spomenieme vlastnosti subgradientov, pomocou nich zadefinujeme podmienky optimality nediferencovateľných funkcií a taktiež si ukážeme subgradientné metódy konvexného programovania. Pri písaní tejto kapitoly sme využili informácie z literatúry [1],[3] a [6].

Definícia 2.6. Nech je $f(x) : R^n \rightarrow R$ konvexná funkcia. Vektor $g \in R^n$ nazývame subgradient funkcie f v bode $x \in D(f)$ ak platí:

$$f(z) \geq f(x) + g^T(z - x), \quad \forall z \in D(f) \quad (2)$$

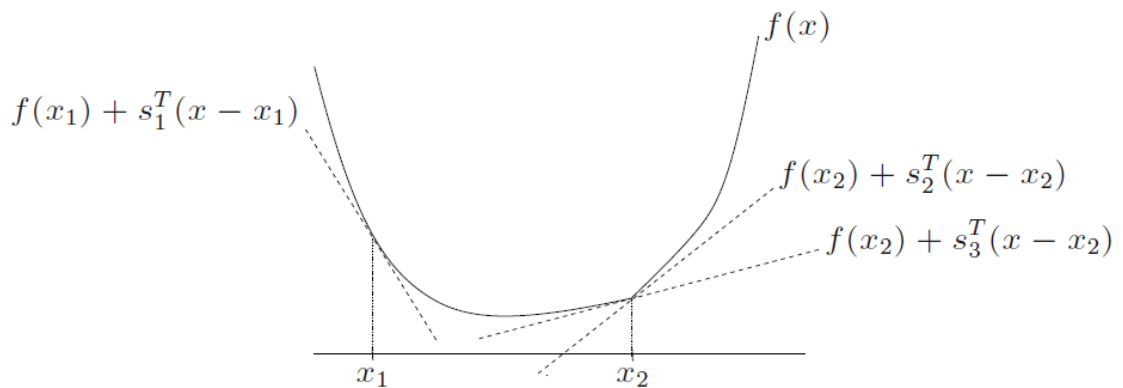
Poznámka 2.2. Pre konvexnú diferencovateľnú funkciu f je subgradient rovný gradientu funkcie pre každé x . Keďže f je konvexná a diferencovateľná funkcia, platí pre ňu podľa vety 2.1.(podmienky 1.rádu) vzťah: $\forall z : f(z) - f(x) \geq \nabla f(x)^T(z - x)$, kde $\nabla f(x)$ je gradient funkcie f v bode x . Z tohto vzťahu vidíme, že vektor $\nabla f(x)$ spĺňa definíciu subgradientu.

Poznámka 2.3. Subgradient existuje aj v bodoch x , v ktorých funkcia nie je diferencovateľná. Avšak v týchto bodoch nemusí byť subgradient jednoznačne definovaný, t.j. existuje množina vektorov, ktoré sú subgradientami danej funkcie v bode x . Táto konvexná množina sa nazýva subdiferenciál v bode x a značí sa $\partial f(x)$.

Definícia 2.7. Funkcia f sa nazýva subdiferencovateľná v x , ak existuje aspoň jeden subgradient v x . Funkcia f sa nazýva subdiferencovateľná, ak existuje subgradient v každom bode jej definičného oboru.

Príklad 2.2. Subgradient absolútnej hodnoty $|x|$

Pre hodnoty $x \neq 0$ je subdiferenciál absolútnej hodnoty jednoprvkovou množinou, čiže



Obr. 5: Definícia 5 - Subgradienty funkcie f

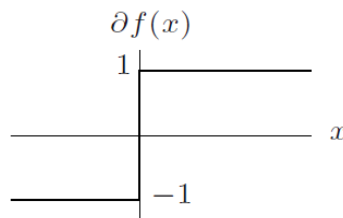
subgradient je v týchto bodoch jednoznačne definovaný. Ak je bod x z intervalu $(-\infty, 0)$ tak je subdiferenciál rovný $\partial f(x) = -1$. Pre body x z intervalu $(0, \infty)$ naopak platí $\partial f(x) = 1$.

V oboch predošlých prípadoch je subgradient v týchto bodoch rovný gradientu. Preto nás najviac zaujíma bod nediferencovateľnosti funkcie.

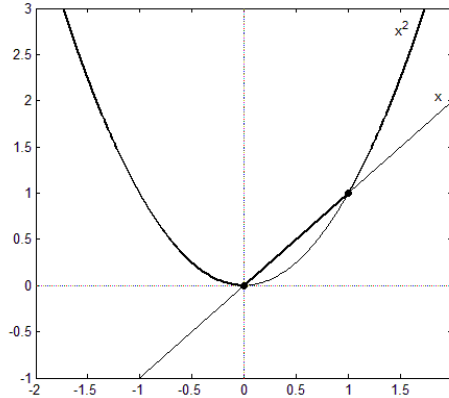
Ak $x = 0$ vieme, že pre subgradient platí nerovnosť (2) z Definície 2.6.:

$$f(z) \geq f(x) + g^T(z - x), \quad \forall z \in D(f).$$

V našom prípade sa táto nerovnosť zjednoduší na tvar $|z| \geq gz, \quad \forall z \in D(f)$. Z toho dostávame, že subdiferenciál funkcie v bode $x = 0$ je množina $\partial f(0) = \langle -1, 1 \rangle$ a subgradient je ľubovoľným bodom tohto intervalu.



Obr. 6: Subdiferenciál absolútnej hodnoty



Obr. 7: $f = \max(x, x^2)$

Príklad 2.3. Subgradient funkcie $f = \max(x, x^2) = \begin{cases} x & x \in \langle 0, 1 \rangle \\ x^2 & x \notin \langle 0, 1 \rangle \end{cases}$

Najzaujímavejšie body x funkcie f budú tie, v ktorých sa pretínajú funkcie x a x^2 . Sú to body $x_1 = 0$ a $x_2 = 1$. V ostatných bodoch sa subgradient rovná gradientu funkcie, ktorá má na danom intervale väčšiu funkčnú hodnotu.

Subgradient v bode $x_1 = 0$: Pre bod x_1 sa nerovnosť (2) z definície subgradientu zredukuje na tvar: $\forall z \quad f(z) \geq 0 + g(z - 0)$.

Ak $z \in (-\infty, 0)$:

$$z^2 \geq gz$$

$$z \leq g$$

Aby subgradient g spĺňal nerovnosť pre všetky z z intervalu musí byť $g \geq 0$.

Ak $z \in \langle 0, \infty \rangle$:

$$z \geq gz$$

$$1 \geq g$$

Spojením predošlých dvoch podmienok pre g spĺňame definičnú nerovnosť pre všetky z , a preto subgradient v bode x_1 je ľubovoľný bod z intervalu $\langle 0, 1 \rangle$. Teda subdiferenciál v tomto bode je: $\partial f(0) = \langle 0, 1 \rangle = \langle (x^2)'(0), x'(0) \rangle$.

Subgradient v bode $x_2 = 1$: Pre bod x_2 sa nerovnosť (2) z definície subgradientu zjednoduší na tvar: $\forall z \quad f(z) \geq 1 + g(z - 1)$.

Ak $z \in (-\infty, 1)$:

$$z \geq 1 + g(z - 1)$$

$$z - 1 \geq g(z - 1)$$

$$1 \leq g$$

Ak $z \in \langle 1, \infty \rangle$:

$$z^2 \geq 1 + g(z - 1)$$

$$z^2 - 1 \geq g(z - 1)$$

$$z + 1 \geq g$$

Aby subgradient g spĺňal nerovnosť pre všetky z z daného intervalu musí byť $g \leq 2$.

Spojením podmienok pre g spĺňame nerovnosť z definície v bode x_2 pre všetky z , a preto subdiferenciál v bode x_1 je: $\partial f(1) = \langle 1, 2 \rangle = \langle x'(1), (x^2)'(1) \rangle$.

Vidíme, že v oboch prípadoch je subgradient v bode "zlomu" x_i ľubovoľný bod z intervalu $\langle \nabla f_1(x_i), \nabla f_2(x_i) \rangle$. Tento jav platí aj všeobecne pri funkciách typu

$f(x) = \max\{f_1(x), f_2(x)\}$, kde funkcie $f_1(x), f_2(x)$ sú konvexné a diferencovateľné.

Príklad 2.4. Subdiferenciál Euklidovskej normy $\|x\|_2$ v bode 0

Zredukujeme nerovnosť z Definície 2.6. a následne použijeme Cauchy - Schwarzovú nerovnosť.

$$\|z\|_2 \geq \|g\|_2 \|z\|_2 \geq g^T z$$

$$1 \geq \|g\|_2$$

$$\partial f(0) = \{g \mid \|g\|_2 \leq 1\}$$

Subdiferenciál Euklidovskej normy v bode 0 obsahuje všetky vektory, ktoré majú dĺžku menšiu alebo rovnú 1.

Aplikácia subgradientov

- Pomocou subgradientov sa definujú podmienky optimality pre nediferencovateľné funkcie.
- Subgradients sú využívané v algoritmoch pre optimalizáciu nediferencovateľných konvexných funkcií ako optimálne smery a ukončovacie kritériá.

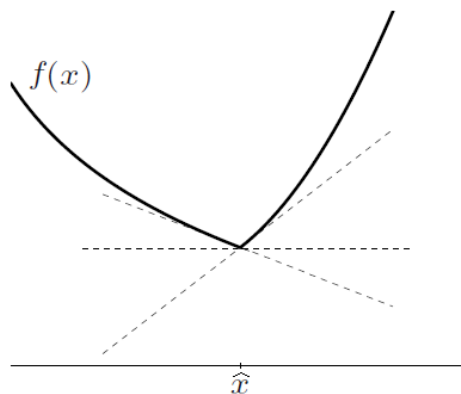
Minimum nediferencovateľnej funkcie

Veta 2.4. *Nech f je konvexná a subdiferencovateľná v \hat{x} . Funkcia nadobúda v bode \hat{x} minimum práve vtedy, keď $0 \in \partial f(\hat{x})$.*

Dôkaz: Pri dokazovaní použijeme definíciu subgradientu 2.6..

” \Leftarrow ” Z faktu $0 \in \partial f(\hat{x})$ vyplýva, že v definícii subgradientu môžeme položiť subgradient g rovný nule. Preto platí vzťah $f(y) \geq f(\hat{x}) + 0^T(y - \hat{x})$, $\forall y$, ktorý je zároveň definíciou minima \hat{x} funkcie $f(x)$.

” \Rightarrow ” Ak bod \hat{x} je minimom funkcie $f(x)$ potom platí $f(y) \geq f(\hat{x})$, $\forall y$. Po pripočítaní nuly k pravej strane nerovnice získame: $f(y) \geq f(\hat{x}) + 0^T(y - \hat{x})$, $\forall y$. Vidíme, že bod 0 spĺňa definíciu subgradientu v bode \hat{x} , preto patrí do množiny subdiferenciálu $0 \in \partial f(\hat{x})$. □



Obr. 8: Minimum nediferencovateľnej funkcie

Vlastnosti subgradientov

Veta 2.5. *Subgradient funkcie $f(x)$ má nasledujúce vlastnosti:*

- Ak α je kladný skalár, potom platí $\partial(\alpha f)(x) = \alpha \partial f(x)$.
- Ak $f(x) = f_1(x) + \dots + f_n(x)$, tak pre subgradient platí $\partial f(x) = \partial f_1(x) + \dots + \partial f_n(x)$.
- Pre funkciu tvaru $h(x) = f(Ax + b)$ sa subgradient rovná $\partial h(x) = A^T \partial f(Ax + b)$.

Veta 2.6. (*Monotónnosť*) Ak $s \in \partial f(x)$ a $\hat{s} \in \partial f(\hat{x})$ potom platí $(s - \hat{s})^T(x - \hat{x}) \geq 0$.

Dôkaz: Postupujeme z definície subgradientu 2.6.. Keďže s je subgradientom v bode x a \hat{s} je subgradientom v bode \hat{x} dostávame dve nerovnice:

$$f(x) \geq f(\hat{x}) + \hat{s}^T(x - \hat{x})$$

$$f(\hat{x}) \geq f(x) + s^T(\hat{x} - x)$$

Sčítaním a následne preusporiadaním týchto dvoch nerovnic dostaneme požadovaný tvar.

$$0 \geq \hat{s}^T(x - \hat{x}) + s^T(\hat{x} - x)$$

$$(s - \hat{s})^T(x - \hat{x}) \geq 0$$

□

2.4 Subgradientné metódy konvexného programovania

Subgradientné metódy sú využívané na optimalizáciu nediferencovateľných, konvexných funkcií. V týchto metódach sa ako optimálny smer minimalizácie používa subgradient.

Algoritmus: Zvolíme začiatočný krok x_0 a opakujeme hlavnú iteráciu, ktorá má nasledovný tvar:

$$x_k = x_{k-1} - t_k g_{k-1}, \quad k = 1, 2, \dots, \quad ,$$

kde g_{k-1} je akýkoľvek subgradient účelovej funkcie f v bode x_{k-1} z predošlej iterácie. Premenná t_k predstavuje dĺžku kroku, býva často konštantou alebo sa volí tak, aby vzdialenosť dvoch za sebou idúcich bodov x_k bola konštantná ($\|x_k - x_{k-1}\|_2 = t_k \|g_{k-1}\|_2 = const.$). Tretí spôsob voľby kroku t_k je tzv. zmenšujúci sa krok, v ktorom $t_k \rightarrow 0$, $\sum_{k=1}^{\infty} t_k = \infty$.

Konvergencia metódy

Predpoklady:

- f je konvexná funkcia s $D(f) = R^n$.
- f má konečnú optimálnu hodnotu \hat{f} a bod optima \hat{x} .

- f je Lipschitzovsky spojitá s konštantou $L > 0$:

$$|f(x) - f(y)| \leq L\|x - y\|_2 \quad \forall x, y$$

Poznamenajme, že konvergencia metódy vyžaduje práve zmenšujúcu sa (dimishing) dĺžku kroku t_k , ostatné spôsoby voľby kroku konvergenciu nezaručujú. Subgradientná metóda nie je spádovou (nerastúcou) metódou, platnosť vzťahu $f^{(k)} \leq f^{(k-1)}$ nie je zaručená. Z uvedeného dôvodu označíme $f_{best}^{(k)}$ hodnotu účelovej funkcie, ktorá je najbližšia optimálnej hodnote účelovej funkcie zo všetkých hodnôt $f^{(i)}$ až po iteráciu k . Dá sa ukázať, že počet operácií potrebných pre dosiahnutie ϵ -suboptimálneho riešenia $f_{best}^{(k)} - \hat{f} \leq \epsilon$ je $O(1/\epsilon^2)$. Subgradientné metódy majú však nevýhody. Sú veľmi pomalé a zložitosť $O(1/\epsilon^2)$ nie je dobrá, pre úlohy rôznych tvarov sa dajú nájsť rýchlejšie metódy s lepšou zložitosťou. Subgradientné metódy taktiež nemajú dobré ukončovacie kritériá.

3 Proximal-gradient metóda

Vzhľadom na nevýhody subgradientných metód boli vymyslené iné metódy na riešenie úloh rôzneho tvaru s nediferencovateľnými konvexnými funkciami. Špeciálne úlohy typu $\min f(x) = g(x) + h(x)$, v ktorých sa dá účelová funkcia rozdeliť na diferencovateľnú a nediferencovateľnú časť, vedú k zavedeniu tzv. proximal operátora a metódy proximal gradient. Tejto metóde konvexného programovania a jej rôznym modifikáciám sa budeme venovať v nasledujúcich kapitolách. Kapitoly 3 a 4 boli spracované na základe [1] a [5].

Proximal gradient metóda sa používa na optimalizáciu neohraničeného problému (čiže bez väzieb), pričom účelová funkcia $f(x)$ sa dá napísať ako súčet dvoch funkcií.

$$\min_x f(x) = g(x) + h(x) \quad (3)$$

kde:

- Funkcia $g(x) : R^n \rightarrow R$ je diferencovateľná, konvexná.
- Funkcia $h(x)$ je uzavretá konvexná. Môže byť nediferencovateľná, najlepšie taká, pri ktorej je proximal operátor nenáročný, jednoduchý.

3.1 Proximal operátor

Základným prvkom algoritmu proximal gradient metódy je tzv. proximal operátor, ktorému sa budeme v nasledujúcej podkapitole venovať.

Definícia 3.1. Proximal operátor konvexnej funkcie h je definovaný ako:

$$\text{prox}_h(x) = \underset{u}{\operatorname{argmin}} \left(h(u) + \frac{1}{2} \|u - x\|_2^2 \right) \quad (4)$$

Veta 3.1. Proximal operátor $\text{prox}_h(x)$ pre konvexú funkciu h existuje a je jednoznačne určený pre každé x .

Dôkaz: Ukážeme, že funkcia $h(u) + \frac{1}{2} \|u - x\|_2^2$ je silnokonvexná vzhľadom na premennú u . Ak je funkcia silnokonvexná tak má práve jeden bod minima, z čoho vyplýva existencia a jednoznačnosť.

Keďže funkcia $h(u)$ je konvexná, budeme sa ďalej sústrediť na vyšetrovanie konvexnosti prídavku $\frac{1}{2}\|u - x\|_2^2 = \frac{1}{2}(u - x)^T(u - x) = f(u)$. Po dvojnásobnom derivovaní funkcie $f(u)$ podľa u zistujeme, že jej Hessova matica je rovná jednotkovej matici. Z toho podľa Vety 2.3. vyplýva, že funkcia $f(u)$ je silnokonvexnou funkciou v premennej u .

Ak h je konvexná a f je silnokonvexná, potom funkcia $F(u) = h(u) + f(u)$ je tiež silnokonvexná (vlastnosť silnokonvexných funkcií). Teda funkcia $h(u) + \frac{1}{2}\|u - x\|_2^2$ je silnokonvexná, čo sme chceli ukázať. \square

Definícia pomocou subgradientu

Veta 3.2. *Proximal operátor $prox_h(x)$ funkcie h možno charakterizovať ako:*

$$\hat{u} = prox_h(x) \Leftrightarrow x - \hat{u} \in \partial h(\hat{u}) \quad (5)$$

Dôkaz: Veta 2.4. o minime nediferencovateľnej funkcie nám hovorí o podmienkach optimality pre minimalizáciu nediferencovateľnej funkcie. Keďže funkcia $h(u)$ môže byť nediferencovateľná použijeme pri odvodzovaní vzťahu medzi subgradientom funkcie h v bode optima \hat{u} a proximal operátorom funkcie h práve túto vetu.

$$\hat{u} = prox_h(x) = \underset{u}{\operatorname{argmin}} \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

Funkcie $h(u)$ a $\frac{1}{2}\|u - x\|_2^2$ sú konvexné, ich súčet bude tiež konvexná funkcia. Hľadáme bod minima \hat{u} funkcie $F(u) = h(u) + \frac{1}{2}\|u - x\|_2^2$. Z podmienok optimality nediferencovateľných funkcií dostávame: $0 \in \partial F(\hat{u})$

$$\begin{aligned} 0 &\in \partial \left(h(\hat{u}) + \frac{1}{2}(\hat{u} - x)^T(\hat{u} - x) \right) \\ 0 &\in \partial h(\hat{u}) + (\hat{u} - x) \\ x - \hat{u} &\in \partial h(\hat{u}). \end{aligned}$$

Medzi proximal operátorom a subgradientom funkcie h v optime \hat{u} teda platí vzťah:

$$\hat{u} = prox_h(x) \Leftrightarrow x - \hat{u} \in \partial h(\hat{u})$$

\square

Vlastnosti proximal operátora

Tvrdenie 3.1. Ak je $h : R^{n_1} \times R^{n_2} \rightarrow R$ separovateľná funkcia $h(u_1, u_2) = h_1(u_1) + h_2(u_2)$, potom platí $\text{prox}_h(x_1, x_2) = (\text{prox}_{h_1}(x_1), \text{prox}_{h_2}(x_2))$.

Dôkaz:

$$\begin{aligned} \text{prox}_h(x_1, x_2) &= \underset{(u_1, u_2)}{\operatorname{argmin}} \left(h(u_1, u_2) + \frac{1}{2} \|u - x\|_2^2 \right) = \\ &= \underset{(u_1, u_2)}{\operatorname{argmin}} \left(h(u_1) + \frac{1}{2} \|u_1 - x_1\|_2^2 + h(u_2) + \frac{1}{2} \|u_2 - x_2\|_2^2 \right) = \end{aligned}$$

Z dôvodu, že hľadanie \hat{u}_1 nezávisí od funkcií s premennou u_2 (berú sa ako konštanta) a opačne, môžeme namiesto hľadania celého vektora \hat{u} hľadať jeho 2 časti a potom ich spojiť do jedného vektora.

$$\begin{aligned} &= \left(\underset{u_1}{\operatorname{argmin}} \left(h(u_1) + \frac{1}{2} \|u_1 - x_1\|_2^2 \right), \underset{u_2}{\operatorname{argmin}} \left(h(u_2) + \frac{1}{2} \|u_2 - x_2\|_2^2 \right) \right) = \\ &= (\text{prox}_{h_1}(x_1), \text{prox}_{h_2}(x_2)). \end{aligned}$$

□

Tvrdenie 3.2. (Škálovanie a posúvanie argumentu): Ak je funkcia $h(u) = f(tu + a)$, $t \neq 0$ získaná škálovaním a posúvaním argumentu funkcie f , potom sa proximálny operátor funkcie h dá prepísať ako:

$$\text{prox}_h(x) = \frac{1}{t} [\text{prox}_{t^2 f}(tx + a) - a]$$

Dôkaz:

$$\begin{aligned} \text{prox}_h(x) &= \underset{u}{\operatorname{argmin}} \left(f(tu + a) + \frac{1}{2} \|u - x\|_2^2 \right) = |u' = tu + a| = \\ &= \frac{1}{t} \left[\underset{u'}{\operatorname{argmin}} \left(f(u') + \frac{1}{2} \left\| \frac{u' - a}{t} - x \right\|_2^2 \right) - a \right] \\ &= \frac{1}{t} \left[\underset{u'}{\operatorname{argmin}} \left(t^2 f(u') + \frac{1}{2} \|u' - a - tx\|_2^2 \right) - a \right] \\ &= \frac{1}{t} [\text{prox}_{t^2 f}(tx + a) - a] \end{aligned}$$

□

Tvrdenie 3.3. (*Nerozpínavosť*): Ak $u = \text{prox}_h(x)$, $\hat{u} = \text{prox}_h(\hat{x})$ potom platí $\|u - \hat{u}\|_2 \leq \|x - \hat{x}\|_2$

Dôkaz: Využijeme definíciu proximal operátora pomocou subgradientu (vzťah (5)) a následne Vetu 2.6. o monotónnosti subgradientu.

$$x - u \in \partial h(u), \quad \hat{x} - \hat{u} \in \partial h(\hat{u}) \quad \Rightarrow \quad (x - u - \hat{x} + \hat{u})^T (u - \hat{u}) \geq 0$$

$$0 \leq ((x - \hat{x}) - (u - \hat{u}))^T (u - \hat{u}) = (x - \hat{x})^T (u - \hat{u}) - (u - \hat{u})^T (u - \hat{u})$$

$$\|u - \hat{u}\|_2^2 \leq (x - \hat{x})^T (u - \hat{u}) \stackrel{C-S}{\leq} \|x - \hat{x}\|_2 \|u - \hat{u}\|_2$$

$$\|u - \hat{u}\|_2 \leq \|x - \hat{x}\|_2$$

prox_h je nerozpínavý alebo Lipschitzovsky spojitá funkcia s konštantou $L = 1$. □

Príklady

V nasledujúcich príkladoch si precvičíme počítanie proximálneho operátora pre rôzne tvary funkcií $h(u)$. Často budeme odvádzať proximal operátor $\text{prox}_{th}(x)$ funkcie $th(u)$ z dôvodu štruktúry algoritmu proximal gradient metódy, kde skalár t predstavuje dĺžku kroku.

Príklad 3.1. $h(u) = 0$: $\text{prox}_h(x) = \underset{u}{\operatorname{argmin}} \left(\frac{1}{2}(u - x)^T(u - x) \right) = x$

Príklad 3.2. Kvadratická funkcia: $h(u) = \frac{1}{2}u^T A u + b^T u + c$

$$\begin{aligned} \text{prox}_{th}(x) &= \underset{u}{\operatorname{argmin}} \left(th(u) + \frac{1}{2}\|u - x\|_2^2 \right) = \\ &= \underset{u}{\operatorname{argmin}} \left(t\left(\frac{1}{2}u^T A u + b^T u + c\right) + \frac{1}{2}(u - x)^T(u - x) \right) = \\ &= \underset{u}{\operatorname{argmin}} \left(t\left(\frac{1}{2}u^T A u + b^T u\right) + \frac{1}{2}(u^T u - 2x^T u) + tc + \frac{1}{2}x^T x \right) = \\ &= \underset{u}{\operatorname{argmin}} \left(\frac{1}{2}u^T (tA + I)u + u^T (tb - x) \right) \end{aligned}$$

Zderivovaním funkcie $F = \frac{1}{2}u^T (tA + I)u + u^T (tb - x)$ podľa u dostaneme:

$$\frac{\partial F}{\partial u} = (tA + I)u + (tb - x) = 0 \quad \Leftrightarrow \quad \hat{u} = (tA + I)^{-1}(x - tb), \text{ a preto}$$

$$\text{prox}_{th}(x) = (tA + I)^{-1}(x - tb)$$

Príklad 3.3. Logaritmická bariéra: $h(u) = -\sum_{i=1}^n \log u_i$

Keďže logaritmická bariéra je separovateľnou funkciou stačí vyjadriť proximal operátor pre jednu zložku.

$$\text{prox}_{th}(x)_i = \underset{u_i}{\operatorname{argmin}} \left(th(u_i) + \frac{1}{2} \|u_i - x_i\|_2^2 \right) = \underset{u_i}{\operatorname{argmin}} \left(-t \log u_i + \frac{1}{2} (u_i - x_i)^2 \right)$$

Zderivovaním funkcie $F = -t \log u_i + \frac{1}{2} (u_i - x_i)^2$ podľa u_i dostaneme: $\frac{\partial F}{\partial u_i} = -\frac{t}{u_i} + u_i - x_i$.

Položíme deriváciu rovnú nule: $-\frac{t}{\hat{u}_i} + \hat{u}_i - x_i = 0$

Prenásobením rovnice premennou \hat{u}_i dostaneme kvadratickú rovnicu, ktorá má 2 korene: $\hat{u}_i = \frac{x_i \pm \sqrt{x_i^2 + 4t}}{2}$. Avšak premenná u_i musí byť vďaka definičnému oboru logaritmu kladnou, a preto koreň s mínusovým znamienkom nebudeme ďalej uvažovať.

Dostávame

$$\text{prox}_{th}(x)_i = \frac{x_i + \sqrt{x_i^2 + 4t}}{2}, \quad i = 1, \dots, n.$$

Príklad 3.4. Jednotková norma: $h(u) = \|u\|_1 = \sum_{i=1}^n |u_i|$

Vďaka separovateľnosti jednotkovej normy stačí vyjadriť proximal operátor pre jednu zložku.

$$\text{prox}_{th}(x)_i = \underset{u_i}{\operatorname{argmin}} \left(th(u_i) + \frac{1}{2} \|u_i - x_i\|_2^2 \right) = \underset{u_i}{\operatorname{argmin}} \left(t|u_i| + \frac{1}{2} (u_i - x_i)^2 \right)$$

Pre $u_i \geq 0$,

$$\text{prox}_{th}(x)_i = \underset{u_i}{\operatorname{argmin}} \left(tu_i + \frac{1}{2} (u_i - x_i)^2 \right) = \underset{u_i}{\operatorname{argmin}} \left(\frac{1}{2} u_i^2 + u_i(t - x_i) + \frac{1}{2} x_i^2 \right)$$

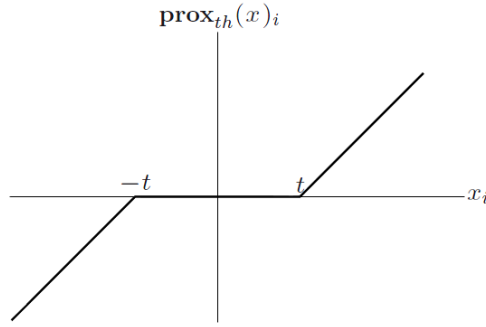
Dostali sme kvadratickú funkciu jednej premennej (u_i), ktorá nadobúda minimum v bode $\hat{u}_i = -\frac{b}{2a} = x_i - t$. Z nezápornosti \hat{u}_i získame podmienku $x_i \geq t$.

Pre $u_i \leq 0$ analogickým postupom dostaneme $\hat{u}_i = x_i + t$, keď $x_i \leq -t$.

Na zvyšnej oblasti, kde $|x_i| \leq t$ je $\hat{u}_i = 0$.

Spojením predchádzajúcich zistení získavame proximal operátor funkcie $th(x) = t\|x\|_1$, ktorý má tvar:

$$\text{prox}_{th}(x)_i = \begin{cases} x_i - t & x_i \geq t \\ x_i + t & x_i \leq -t \\ 0 & |x_i| \leq t \end{cases} \quad (6)$$



Obr. 9: Proximálny operátor jednotkovej normy

Príklad 3.5. Euklidovská norma: $h(u) = \|u\|_2$

$$\text{prox}_{th}(x) = \underset{u}{\operatorname{argmin}} \left(th(u) + \frac{1}{2} \|u - x\|_2^2 \right) = \underset{u}{\operatorname{argmin}} \left(t\|u\|_2 + \frac{1}{2} \|u - x\|_2^2 \right)$$

Bod u , v ktorom funkcia $F(u) = (t\|u\|_2 + \frac{1}{2}\|u - x\|_2^2)$ nadobúda minimum, nájdeme podobne ako v predošlých prípadoch zderivovaním funkcie podľa premennej u a položením derivácie rovnej nule.

$$\begin{aligned} \frac{\partial F(u)}{\partial u} &= \frac{\partial (t\|u\|_2 + \frac{1}{2}\|u - x\|_2^2)}{\partial u} = \\ &= t \frac{u}{\|u\|_2} + u - x = 0 \end{aligned}$$

Po derivácii funkcie dostávame výraz, z ktorého na prvý pohľad nevieme vyjadriť premennú \hat{u} predstavujúcu proximalny operátor. Preto si z výrazu najprv vyjadríme čomu je rovná $\|u\|_2$, ktorú spätne dosadíme do pôvodného výrazu. Poznamenajme, že v bode $u = 0$ nie je funkcia diferencovateľná. Pre u nenulové platí:

$$\begin{aligned} u \left(\frac{t}{\|u\|_2} + 1 \right) &= x \\ \|u \left(\frac{t}{\|u\|_2} + 1 \right)\|_2 &= \|x\|_2 \\ \left(\frac{t}{\|u\|_2} + 1 \right) \|u\|_2 &= \|x\|_2 \\ \|u\|_2 &= \|x\|_2 - t \end{aligned}$$

Pri upravovaní sme využili fakt, že $\left(\frac{t}{\|u\|_2} + 1 \right)$ je konštanta, a preto sme ju mohli vytiahnuť pred normu. Z dôvodu vlastností nezápornosti normy dostávame podmienku

$\|x\|_2 \geq t$. Nahradíme normu $\|u\|_2$ a následne vyjadríme, čomu sa rovná u :

$$\begin{aligned} u \left(\frac{t}{\|u\|_2} + 1 \right) &= x \\ u \left(\frac{\|x\|_2}{\|x\|_2 - t} \right) &= x \\ u &= x \left(1 - \frac{t}{\|x\|_2} \right) \end{aligned}$$

Ak nie je podmienka $\|x\|_2 \geq t$ splnená bodom minima je práve bod nediferencovateľnosti funkcie $u = 0$. Proximálny operátor Euklidovskej normy je teda rovný:

$$prox_{th}(x) = \begin{cases} x \left(1 - \frac{t}{\|x\|_2} \right) & \|x\|_2 \geq t \\ 0 & \text{inak} \end{cases}$$

Príklad 3.6. Indikátorová funkcia: $h(u) = I_C(u) = \begin{cases} 0 & u \in C \\ \infty & \text{inak} \end{cases}$

$$prox_h(x) = \underset{u}{\operatorname{argmin}} \left(I_C(u) + \frac{1}{2} \|u - x\|_2^2 \right) =$$

Zaujíma nás bod minima \hat{u} danej funkcie, ktorá obsahuje indikátorovú funkčnú časť. Vďaka tejto časti sme "penalizovaní" hodnotou $+\infty$ vtedy, keď sa pohybujeme mimo množiny C . Preto hľadanie bodu minima zúžime práve na množinu C , v ktorej bodoch má indikátorová funkcia funkčnú hodnotu rovnú 0.

$$= \underset{u \in C}{\operatorname{argmin}} \frac{1}{2} \|u - x\|_2^2 = \underset{u \in C}{\operatorname{argmin}} \|u - x\|_2^2 = P_C(x).$$

Proximal operátor indikátorovej funkcie I_C je Euklidovská projekcia na množinu C .

$$prox_{I_C}(x) = P_C(x)$$

3.2 Algoritmus proximal-gradient metódy

Algoritmus proximal-gradient metódy na riešenie úlohy (3) pozostáva z opakovania hlavnej iterácie až po dosiahnutie požadovanej presnosti. Proximal-gradient iterácia vyzerá nasledovne:

$$x^{(k)} = \text{prox}_{t_k h}(x^{(k-1)} - t_k \nabla g(x^{(k-1)}))$$

Tento vzťah sa dá prepísať ako

$$x^{(k)} = x^{(k-1)} - t_k G_{t_k}(x^{(k-1)}),$$

pričom $G_t(x) = \frac{1}{t}(x - \text{prox}_{th}(x - t\nabla g(x)))$ je smer, ktorý môžeme chápať ako subgradient funkcie f , o čom hovorí aj Veta 3.3. a Poznámka 3.1..

Premenná $t_k > 0$ je dĺžka kroku, ktorú volíme buď konštantnú alebo ju budeme určovať v každej iterácii pomocou *line search*.

Príklad 3.7. $h(u) = 0$

Minimalizačná úloha sa zjednoduší na tvar $\min_x g(x)$ a iterácia je nasledovného tvaru:

$$x^{(k)} = x^{(k-1)} - t_k \nabla g(x^{(k-1)})$$

Dostaneme teda klasickú gradientnú metódu.

Príklad 3.8. $h(u) = I_C(u)$

Minimalizačná úloha sa dá chápať ako $\min_{x \in C} g(x)$. Iterácia sa zmení na tvar gradientnej projekčnej metódy, pretože proximal operátor sa správa ako projekcia.

$$x^{(k)} = P_C(x^{(k-1)} - t_k \nabla g(x^{(k-1)}))$$

Veta 3.3. : Pre smer $G_t(x)$ definovaný ako $G_t(x) = \frac{1}{t}(x - \text{prox}_{th}(x - t\nabla g(x)))$ platí:

$$G_t(x) \in \nabla g(x) + \partial h(x - tG_t(x)). \quad (7)$$

Dôkaz: Vzťah definujúci smer $G_t(x)$ prepíšeme na nasledovný tvar:

$$x - tG_t(x) = \text{prox}_{th}(x - t\nabla g(x))$$

V ďalšom kroku si označíme $u' = x - tG_t(x)$, $x' = x - t\nabla g(x)$, rovnica sa nám teda zjednoduší na tvar $u' = \text{prox}_{th}(x')$. Potom z Vety 3.2.(definícia pomocou subgradientu) vyplýva, že $x' - u' \in t\partial h(u')$. Vrátením sa na pôvodné značenie získame vzťah, ktorý jednoduchými úpravami dostaneme do požadovaného tvaru.

$$\begin{aligned} t(-\nabla g(x) + G_t(x)) &\in t\partial h(x - tG_t(x)) \\ G_t(x) &\in \nabla g(x) + \partial h(x - tG_t(x)) \end{aligned}$$

□

Poznámka 3.1. Veta nám vlastne hovorí o tom, že smer $G_t(x)$ patrí do subdiferenciálu funkcie h v bode x^k novej iterácie posunutom o gradient funkcie g v bode predchádzajúcej iterácie x^{k-1} . Preto sa $G_t(x)$ dá chápať ako "subgradient" účelovej funkcie f .

Dôsledok 3.1. : *Bod x je minimom funkcie $f(x) = g(x) + h(x)$ práve vtedy, keď $G_t(x) = 0$. Táto skutočnosť vyplýva priamo z podmienok optimality diferencovateľných a nediferencovateľných funkcií.*

Interpretácia iterácie

Využijeme definíciu proximal operátora.

$$\begin{aligned} x^+ &= \text{prox}_{th}(x - t\nabla g(x)) = \underset{u}{\text{argmin}} \left(th(u) + \frac{1}{2}\|u - x + t\nabla g(x)\|_2^2 \right) \\ &= \underset{u}{\text{argmin}} \left(h(u) + \frac{1}{2t}\|u - x\|_2^2 + \nabla g(x)^T(u - x) + \frac{t}{2}\|\nabla g(x)\|_2^2 \right) \\ &= \underset{u}{\text{argmin}} \left(h(u) + g(x) + \nabla g(x)^T(u - x) + \frac{1}{2t}\|u - x\|_2^2 \right) \end{aligned}$$

Výsledok nám hovorí o tom, že v každej iterácii nový bod $x^{(k)}$ minimalizuje funkciu $h(u)$ plus jednoduchý lokálny kvadratický model funkcie $g(u)$ okolo bodu $x^{(k-1)}$ z predchádzajúcej iterácie.

Line search

Cieľom line search algoritmu je určiť optimálnu dĺžku kroku t_k v časti metódy:

$$x^{(k)} = \text{prox}_{t_k h}(x^{(k-1)} - t_k \nabla g(x^{(k-1)})) = x^{(k-1)} - t_k G_{t_k}(x^{(k-1)})$$

Algoritmus:

Začínáme s krokom $t = t_{k-1}$ z predošlej iterácie a opakujeme $t = \beta t$ ($0 < \beta < 1$) až

pokým náš krok splní line search nerovnosť:

$$g(x^{(k)}) = g(x^{(k-1)} - tG_t(x^{(k-1)})) \leq g(x^{(k-1)}) - t\nabla g(x^{(k-1)})^T G_t(x^{(k-1)}) + \frac{t}{2} \|G_t(x^{(k-1)})\|_2^2$$

Ako počiatočný krok t_0 volíme ľubovoľnú kladnú hodnotu. Algoritmus vyžaduje výpočet jedného proximal operátora za line search iteráciu. Line search nerovnosť je inšpirovaná analýzou konvergence, čo ukážeme v ďalšej podkapitole.

3.3 Konvergencia proximal-gradient metódy

Predpoklady:

- Gradient funkcie ∇g je Lipschitzovsky spojitá funkcia s konštantou $L > 0$

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in R^n.$$

- Optimálna funkčná hodnota \hat{f} je konečná a funkcia ju nadobúda v bode \hat{x} , ktorý nemusí byť jednoznačný.

Konvergenciu metódy budeme vyšetrovať pre fixný krok t ako aj pre krok t_k , ktorý určujeme pri každej iterácii pomocou line search.

Dôsledky Lipschitzovského predpokladu

Ako prvé odvodíme horné ohraničenie pre funkciu g z Lipschitzovej vlastnosti, ktoré budeme v ďalšom postupe potrebovať.

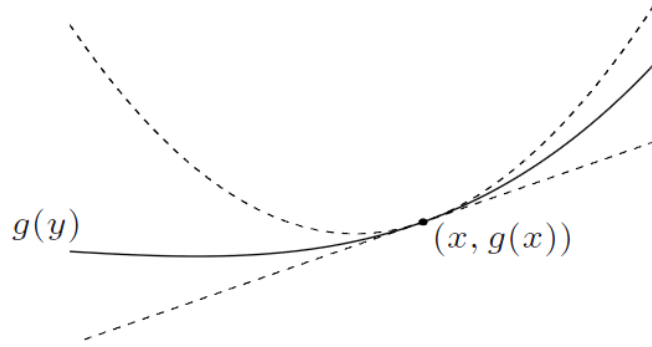
Veta 3.4. *Konvexná funkcia g s Lipschitzovsky spojitým gradientom s konštantou L má horné ohraničenie*

$$g(y) \leq g(x) + \nabla g(x)^T(y - x) + \frac{L}{2}\|y - x\|_2^2 \quad \forall x, y \quad (8)$$

Dôkaz: Označme $v = y - x$. Vieme, že pre $g(y)$ platí nasledujúca rovnosť, z ktorej postupnými úpravami získame horné ohraničenie.

$$\begin{aligned} g(y) &= g(x) + \nabla g(x)^T v + \int_0^1 (\nabla g(x + tv) - \nabla g(x))^T v dt \leq \\ &\stackrel{(1)}{\leq} g(x) + \nabla g(x)^T v + \int_0^1 \|\nabla g(x + tv) - \nabla g(x)\|_2 \|v\|_2 dt \leq \\ &\stackrel{(2)}{\leq} g(x) + \nabla g(x)^T v + \int_0^1 Lt \|v\|_2^2 dt = \\ &= g(x) + \nabla g(x)^T v + \frac{L}{2} \|v\|_2^2 \end{aligned}$$

Úprava (1) vyplýva z Cauchy-Schwarzovej nerovnosti $x^T y \leq \|x\|_2 \|y\|_2$. V druhej úprave (2) sa využíva Lipschitzovská spojitosť gradientu ∇g , ktorá bola zahrnutá v predpokladoch konvergenencie. \square



Obr. 10: Horné a dolné ohrančenie konvexnej funkcie $g(x)$

Dôsledok 3.2. (*Line search nerovnosť*) Položením $y = x - tG_t(x)$ do horného ohraničenia (8) dostaneme:

$$g(x - tG_t(x)) \leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t^2 L}{2} \|G_t(x)\|_2^2$$

Na základe toho je line search nerovnosť v tvare:

$$g(x - tG_t(x)) \leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2} \|G_t(x)\|_2^2 \quad (9)$$

Nerovnosť je splnená pre dĺžku kroku $0 \leq t \leq 1/L$. Pre krajné hodnoty t sa uvedené vzťahy rovnajú. Mimo intervalu je prvý vzťah slabšou podmienkou, a preto nie je platnosť druhého vzťahu zaručená.

Veta 3.5. (*globálna nerovnosť*): Nech funkcia f je tvaru $f(x) = g(x) + h(x)$, kde funkcia $g(x)$ je konvexná diferencovateľná a $h(x)$ je konvexná nediferencovateľná funkcia. Ak platí line search nerovnosť (9) pre funkciu g , potom pre každé z platí vzťah:

$$f(x - tG_t(x)) \leq f(z) + G_t(x)^T (x - z) - \frac{t}{2} \|G_t(x)\|_2^2 \quad (10)$$

Dôkaz:

$$\begin{aligned}
f(x - tG_t(x)) &= g(x - tG_t(x)) + h(x - tG_t(x)) \\
&\leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2} \|G_t(x)\|_2^2 + h(x - tG_t(x)) \\
&\leq g(z) + \nabla g(x)^T (x - z) - t\nabla g(x)^T G_t(x) + \frac{t}{2} \|G_t(x)\|_2^2 \\
&\quad + h(z) + (G_t(x) - \nabla g(x))^T (x - z - tG_t(x)) \\
&= f(z) + G_t(x)^T (x - z) - \frac{t}{2} \|G_t(x)\|_2^2
\end{aligned}$$

V druhom riadku je využitá line search nerovnosť (9) pre funkciu g . V nasledujúcom riadku sa použila konvexnosť oboch funkcií: pre funkciu g bola použitá Veta 2.1. (podmienky 1. rádu), pre funkciu h bola využitá Definícia 2.6. (subgradient) a fakt, že $(G_t(x) - \nabla g(x)) \in \partial h(x - tG_t(x))$, čo sme si ukázali po definovaní smeru $G_t(x)$ vo Vete 3.3.. □

Veta 3.6. *Proximal gradient je spádová metóda, platí $f(x^+) \leq f(x) - \frac{t}{2} \|G_t(x)\|_2^2$.*

Dôkaz: Označme bod nasledujúcej iterácie $x^+ = x - tG_t(x)$. Ak v globálnej nerovnosti (10) položíme $z=x$ dostaneme požadovaný tvar. □

Poznámka 3.2. Tento vzťah nám hovorí o tom, že proximal gradient je nerastúca (klesajúca) metóda. Po každej novej iterácii sa funkčná hodnota účelovej funkcie zmenší alebo ostane rovnaká (ak sme v optime). Funkčná hodnota sa za každú iteráciu zlepšuje prínajmenšom o hodnotu $\frac{t}{2} \|G_t(x)\|_2^2$, tento jav je dostatočný na to, aby garantoval konvergenciu metódy.

Ak do globálnej nerovnosti (10) dosadíme $z = \hat{x}$, dostaneme:

$$\begin{aligned}
0 \leq f(x^+) - \hat{f} &\leq G_t(x)^T (x - \hat{x}) - \frac{t}{2} \|G_t(x)\|_2^2 \\
&= \frac{1}{t} (x - x^+)^T (x - \hat{x}) - \frac{1}{2t} \|x - x^+\|_2^2 - \frac{1}{2t} \|x - \hat{x}\|_2^2 + \frac{1}{2t} \|x - \hat{x}\|_2^2 \\
&= \frac{1}{2t} (\|x - \hat{x}\|_2^2 - \|x^+ - \hat{x}\|_2^2)
\end{aligned} \tag{11}$$

V druhom riadku sme využili rovnosť $G_t(x) = \frac{x-x^+}{t}$ a pripočítali sme nulu. Pokračovali sme s úpravou na štvorec podľa vzorca $(A-B)^2 = A^2 - 2AB + B^2$, kde $A = (x - \hat{x})$ a $B =$

$(x - x^+)$. Z predchádzajúcej nerovnice taktiež pozorujeme, že vzdialenosť priebežného riešenia od optimálneho sa znižuje každou iteráciou, pretože platí:

$$\|x^+ - \hat{x}\|_2^2 \leq \|x - \hat{x}\|_2^2 \quad (12)$$

Analýza konvergenzie PGM

Konštantný krok $t = 1/L$

Veta 3.7. *Proximal gradient metóda s konštantným krokom $t = 1/L$ dosahuje ϵ presné riešenie po $O(1/\epsilon)$ krokoch - platí $f(x^{(k)}) - \hat{f} \leq \frac{1}{2kt} \|x^{(0)} - \hat{x}\|_2^2$.*

Dôkaz: V nerovnosti (11) položíme $x = x^{(i-1)}$, $x^+ = x^{(i)}$:

$$f(x^{(i)}) - \hat{f} \leq \frac{1}{2t} (\|x^{(i-1)} - \hat{x}\|_2^2 - \|x^{(i)} - \hat{x}\|_2^2) \quad (13)$$

Zosumujeme všetky nerovnosti (13) od prvej iterácie až po poslednú k -tú iteráciu.

$$\begin{aligned} \sum_{i=1}^k (f(x^{(i)}) - \hat{f}) &\leq \frac{1}{2t} \sum_{i=1}^k (\|x^{(i-1)} - \hat{x}\|_2^2 - \|x^{(i)} - \hat{x}\|_2^2) \\ &= \frac{1}{2t} (\|x^{(0)} - \hat{x}\|_2^2 - \|x^{(1)} - \hat{x}\|_2^2 + \|x^{(1)} - \hat{x}\|_2^2 - \dots \\ &\quad \dots - \|x^{(k-1)} - \hat{x}\|_2^2 + \|x^{(k-1)} - \hat{x}\|_2^2 - \|x^{(k)} - \hat{x}\|_2^2) \\ &= \frac{1}{2t} (\|x^{(0)} - \hat{x}\|_2^2 - \|x^{(k)} - \hat{x}\|_2^2) \\ &\leq \frac{1}{2t} \|x^{(0)} - \hat{x}\|_2^2 \end{aligned}$$

Nerovnosť predelíme počtom iterácii k a z poznatku, že $f(x^{(i)})$ je nerastúca dostávame:

$$f(x^{(k)}) - \hat{f} = \frac{1}{k} \sum_{i=1}^k (f(x^{(k)}) - \hat{f}) \leq \frac{1}{k} \sum_{i=1}^k (f(x^{(i)}) - \hat{f}) \leq \frac{1}{2kt} \|x^{(0)} - \hat{x}\|_2^2$$

□

Krok určovaný pomocou line search v každej iterácii $t = t_i \geq t_{min}$

Veta 3.8. *Proximal gradient metóda s line search krokom t_k nájde ϵ presné riešenie po $O(1/\epsilon)$ iteráciách - platí $f(x^{(k)}) - \hat{f} \leq \frac{1}{2kt_{min}} \|x^{(0)} - \hat{x}\|_2^2$.*

Dôkaz: Budeme postupovať analogicky ako v predošlom prípade, jediným rozdielom bude závislosť kroku t od iterácie i . Zosumovaním všetkých nerovností (13) od prvej

iterácie až po k -tú iteráciu dostávame:

$$\begin{aligned}
 \sum_{i=1}^k (f(x^{(i)}) - \hat{f}) &\leq \sum_{i=1}^k \frac{1}{2t_i} (\|x^{(i-1)} - \hat{x}\|_2^2 - \|x^{(i)} - \hat{x}\|_2^2) \\
 &\leq \frac{1}{2t_{\min}} \sum_{i=1}^k (\|x^{(i-1)} - \hat{x}\|_2^2 - \|x^{(i)} - \hat{x}\|_2^2) \\
 &= \frac{1}{2t_{\min}} (\|x^{(0)} - \hat{x}\|_2^2 - \|x^{(k)} - \hat{x}\|_2^2) \\
 &\leq \frac{1}{2t_{\min}} \|x^{(0)} - \hat{x}\|_2^2
 \end{aligned}$$

Nerovnosť predelíme počtom iterácii k a z poznatku, že $f(x^{(i)})$ je nerastúca dostaneme:

$$f(x^{(k)}) - \hat{f} \leq \frac{1}{2kt_{\min}} \|x^{(0)} - \hat{x}\|_2^2$$

□

Ukázali sme, že pri oboch spôsoboch voľby kroku sa rozdiel $f(x^{(k)}) - \hat{f}$ znižuje najmenej rýchlosťou $1/k$. Na nájdenie ϵ -suboptimálneho bodu spĺňajúceho $f(x^{(k)}) - \hat{f} \leq \epsilon$ pomocou klasickej proximal gradient metódy teda potrebujeme $O(1/\epsilon)$ iterácii. Táto zložitosť sa však dá vylepšiť vhodnou úpravou metódy na hodnotu $O(1/\sqrt{\epsilon})$. Upravený algoritmus s lepšou zložitosťou sa nazýva zrýchlená proximal gradient metóda.

4 Zrýchlená proximal gradient metóda

Zrýchlená metóda má približne rovnakú náročnosť za iteráciu ako základná proximal gradient metóda. Od klasického algoritmu sa líši v tom, že do proximálneho operátora v každej iterácii vstupujú namiesto hodnôt $x^{(k-1)}$ hodnoty $y^{(k-1)}$, ktoré sú získané extrapoláciou z predošlých hodnôt $x^{(k-1)}$ a $x^{(k)}$. Preto je zrýchlená metóda nazývaná tiež ako proximal gradient metóda s extrapoláciou.

Algoritmus:

Zvolíme štartovací bod $x^{(0)} \in D(h)$ a položíme $y^{(0)} = x^{(0)}$, potom pre $k \geq 1$:

$$\begin{aligned}x^{(k)} &= \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) \\y^{(k)} &= x^{(k)} + \frac{k-1}{k+2} (x^{(k)} - x^{(k-1)})\end{aligned}$$

Krok t_k býva konštantný alebo určovaný pomocou line search.

Všimnime si, že prvá iterácia zrýchlenej metódy je zhodná s prvou iteráciou pôvodnej metódy.

Line search

Cieľom line search algoritmu je určiť optimálnu dĺžku kroku t_k v časti metódy:

$$x^{(k)} = \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) = y^{(k-1)} - t_k G_{t_k}(y^{(k-1)})$$

Algoritmus line search:

Začínáme s krokom $t = t_{k-1}$ z predošlej iterácie a opakujeme $t = \beta t$ ($0 < \beta < 1$) až pokým náš krok splní nerovnosť:

$$g(x^{(k)}) = g(y^{(k-1)} - t G_t(y^{(k-1)})) \leq g(y^{(k-1)}) - t \nabla g(y^{(k-1)})^T G_t(y^{(k-1)}) + \frac{t}{2} \|G_t(y^{(k-1)})\|_2^2$$

Ako počiatočný krok t_0 volíme ľubovoľnú kladnú hodnotu. Poznamenajme, že takto definovaný line search algoritmus implikuje nerastúcosť kroku $t_k \leq t_{k-1}$.

4.1 Konvergencia zrýchlenej proximal gradient metódy

Predpoklady:

- Gradient funkcie ∇g je Lipschitzovsky spojitá funkcia s konštantou $L > 0$

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y.$$

- Optimálna funkčná hodnota \hat{f} je konečná a funkcia ju nadobúda v bode \hat{x} , ktorý nemusí byť jednoznačný.

Rovnako ako v predošlej kapitole budeme analyzovať konvergenciu zrýchlenej metódy pre oba spôsoby voľby kroku t - pre fixný krok aj pre krok pomocou line search.

Postup je analogický k predchádzajúcemu vyšetrovaniu konvergenzie pre klasickú metódu, a preto využijeme niektoré vzťahy, ktoré boli odvodené a zdôvodnené tam.

Line search nerovnosť je tvaru:

$$g(y - tG_t(y)) \leq g(y) - t\nabla g(y)^T G_t(y) + \frac{t}{2}\|G_t(y)\|_2^2,$$

v ktorej je jediným rozdielom to, že namiesto pôvodných hodnôt x používame extrapolované hodnoty y . Nerovnosť je splnená pre dĺžku kroku $0 \leq t \leq 1/L$. Takisto platí aj globálna nerovnosť pre všetky z , keď t spĺňa line search nerovnosť.

$$f(y - tG_t(y)) \leq f(z) + G_t(y)^T(y - z) - \frac{t}{2}\|G_t(y)\|_2^2 \quad (14)$$

Notácia:

Označme $\theta_k = \frac{2}{k+1}$, definujeme $v^{(0)} = x^{(0)}$ a pre $k \geq 1$:

$$v^{(k)} = x^{(k-1)} + \frac{1}{\theta_k}(x^{(k)} - x^{(k-1)})$$

Pri novom označovaní sa dá extrapolované $y^{(k)}$ prepísať nasledovne:

$$\begin{aligned} y^{(k)} &= x^{(k)} + \frac{k-1}{k+2}(x^{(k)} - x^{(k-1)}) \\ &= x^{(k)} + \frac{k+1}{k+2}(x^{(k)} - x^{(k-1)}) - \frac{2}{k+2}(x^{(k)} - x^{(k-1)}) \\ &= x^{(k)} + \frac{\theta_{k+1}}{\theta_k}(x^{(k)} - x^{(k-1)}) - \theta_{k+1}(x^{(k)} - x^{(k-1)}) \\ &= (1 - \theta_{k+1})x^{(k)} + \theta_{k+1}(x^{(k-1)} + \frac{1}{\theta_k}(x^{(k)} - x^{(k-1)})) \\ &= (1 - \theta_{k+1})x^{(k)} + \theta_{k+1}v^{(k)} \end{aligned} \quad (15)$$

Pomocou nového $y^{(k)}$ prepíšeme aj $v^{(k)}$:

$$\begin{aligned}
v^{(k)} &= x^{(k-1)} + \frac{1}{\theta_k}(x^{(k)} - x^{(k-1)}) \\
&= x^{(k-1)} + \frac{1}{\theta_k}(y^{(k-1)} - t_k G_{t_k}(y^{(k-1)}) - x^{(k-1)}) \\
&= x^{(k-1)} + \frac{1}{\theta_k}((1 - \theta_k)x^{(k-1)} + \theta_k v^{(k-1)} - t_k G_{t_k}(y^{(k-1)}) - x^{(k-1)}) \\
&= v^{(k-1)} - \frac{t_k}{\theta_k} G_{t_k}(y^{(k-1)})
\end{aligned} \tag{16}$$

Druhý riadok vychádza z rovnosti $x^{(k)} = y^{(k-1)} - t_k G_{t_k}(y^{(k-1)})$. V treťom riadku sme $y^{(k-1)}$ rozpísali pomocou predchádzajúceho vzťahu (15).

V ďalšom postupe využijeme aj fakt, že θ_k spĺňa nerovnosť:

$$\frac{1 - \theta_k}{\theta_k^2} = \frac{k^2 - 1}{4} \leq \frac{k^2}{4} = \frac{1}{\theta_{k-1}^2} \tag{17}$$

Progres v jednej iterácii

Veta 4.1. *Zrýchlená proximal gradient metóda dosahuje za jednu iteráciu progres*

$$\frac{1}{\theta_i^2}(f(x^{(i)}) - \hat{f}) + \frac{1}{2t_i}\|v^{(i)} - \hat{x}\|_2^2 \leq \frac{1 - \theta_i}{\theta_i^2}(f(x^{(i-1)}) - \hat{f}) + \frac{1}{2t_i}\|v^{(i-1)} - \hat{x}\|_2^2 \tag{18}$$

Dôkaz: Použijeme globálnu nerovnosť(14), ktorá platí pre všetky z . V našom prípade položíme z rovné konvexnej kombinácii $z = (1 - \theta_i)x^{(i-1)} + \theta_i\hat{x}$

$$\begin{aligned}
f(x^{(i)}) &\leq f((1 - \theta_i)x^{(i-1)} + \theta_i\hat{x}) + G_t(y^{(i-1)})^T(y^{(i-1)} - (1 - \theta_i)x^{(i-1)} - \theta_i\hat{x}) - \frac{t}{2}\|G_t(y^{(i-1)})\|_2^2 \\
&\leq (1 - \theta_i)f(x^{(i-1)}) + \theta_i\hat{f} + G_t(y^{(i-1)})^T(\theta_i v^{(i-1)} - \theta_i\hat{x}) - \frac{t}{2}\|G_t(y^{(i-1)})\|_2^2 \\
&= (1 - \theta_i)f(x^{(i-1)}) + \theta_i\hat{f} + \frac{\theta_i^2}{2t} \left(\frac{2t}{\theta_i} G_t(y^{(i-1)})^T(v^{(i-1)} - \hat{x}) - \frac{t^2}{\theta_i^2} \|G_t(y^{(i-1)})\|_2^2 \right) \\
&= (1 - \theta_i)f(x^{(i-1)}) + \theta_i\hat{f} + \frac{\theta_i^2}{2t} \left(\|v^{(i-1)} - \hat{x}\|_2^2 - \|v^{(i-1)} - \frac{t}{\theta_i} G_t(y^{(i-1)}) - \hat{x}\|_2^2 \right) \\
&= (1 - \theta_i)f(x^{(i-1)}) + \theta_i\hat{f} + \frac{\theta_i^2}{2t} (\|v^{(i-1)} - \hat{x}\|_2^2 - \|v^{(i)} - \hat{x}\|_2^2)
\end{aligned}$$

Pri úpravách na druhý riadok bola využitá lambda definícia konvexnosti funkcie f (Definícia 2.2.), takisto sa použil upravený vzťah(15) pre $y^{(i-1)}$. V štvrtom riadku sme predchádzajúci výraz upravili na štvorce pomocou vzorca $(A - B)^2 = A^2 - 2AB + B^2$, kde $A = (v^{(i-1)} - \hat{x})$, $B = \frac{t}{\theta_i} G_t(y^{(i-1)})$. Posledný riadok používa rovnosť(16) pre premennú $v^{(i)}$.

Odpočítaním od oboch strán \hat{f} , predelením nerovnice θ_i^2 a preusporiadaním, aby na ľavej strane boli indexy novej iterácie i a na pravej indexy predchádzajúcej iterácie $i - 1$, dostaneme "progres" za jednu iteráciu, ktorý použijeme na odvodenie konvergen-
cie zrýchlenej metódy.

$$\frac{1}{\theta_i^2}(f(x^{(i)}) - \hat{f}) + \frac{1}{2t_i}\|v^{(i)} - \hat{x}\|_2^2 \leq \frac{1 - \theta_i}{\theta_i^2}(f(x^{(i-1)}) - \hat{f}) + \frac{1}{2t_i}\|v^{(i-1)} - \hat{x}\|_2^2$$

□

Analýza konvergenzie zrýchlenej PGM

Konštantný krok $t_i = t = 1/L$

Veta 4.2. *Zrýchlená proximal gradient metóda s konštantným krokom potrebuje na dosiahnutie ϵ -presného riešenia $f(x^{(k)}) - \hat{f} \leq \epsilon$ počet $O(1/\sqrt{\epsilon})$ iterácií, platí $f(x^{(k)}) - \hat{f} \leq \frac{2}{(k+1)^2t}\|x^{(0)} - \hat{x}\|_2^2$*

Dôkaz: Budeme opakovane využívať vlastnosť θ_k (17) a nerovnicu progresu za jednu iteráciu (18).

$$\begin{aligned} \frac{1}{\theta_k^2}(f(x^{(k)}) - \hat{f}) &\leq \frac{1}{\theta_k^2}(f(x^{(k)}) - \hat{f}) + \frac{1}{2t}\|v^{(k)} - \hat{x}\|_2^2 \leq \\ &\leq \frac{1 - \theta_k}{\theta_k^2}(f(x^{(k-1)}) - \hat{f}) + \frac{1}{2t}\|v^{(k-1)} - \hat{x}\|_2^2 \leq \\ &\leq \frac{1}{\theta_{k-1}^2}(f(x^{(k-1)}) - \hat{f}) + \frac{1}{2t}\|v^{(k-1)} - \hat{x}\|_2^2 \leq \dots \\ &\dots \leq \frac{1 - \theta_1}{\theta_1^2}(f(x^{(0)}) - \hat{f}) + \frac{1}{2t}\|v^{(0)} - \hat{x}\|_2^2 = \\ &= \frac{1}{2t}\|x^{(0)} - \hat{x}\|_2^2 \end{aligned}$$

Preto platí:

$$f(x^{(k)}) - \hat{f} \leq \frac{\theta_k^2}{2t}\|x^{(0)} - \hat{x}\|_2^2 = \frac{2}{(k+1)^2t}\|x^{(0)} - \hat{x}\|_2^2$$

□

Krok určovaný pomocou line search v každej iterácii $t_{i-1} \geq t_i \geq t_{min}$

Veta 4.3. *Zrýchlená proximal gradient metóda s line search krokom nájde ϵ -suboptimálny bod $f(x^{(k)}) - \hat{f} \leq \epsilon$ za počet $O(1/\sqrt{\epsilon})$ iterácií, platí $f(x^{(k)}) - \hat{f} \leq \frac{2}{(k+1)^2 t_{min}} \|x^{(0)} - \hat{x}\|_2^2$*

Dôkaz: Budeme postupovať analogickým spôsobom ako v predošlom prípade, jediným rozdielom bude závislosť kroku t od iterácie i .

$$\begin{aligned}
 \frac{t_{min}}{\theta_k^2} (f(x^{(k)}) - \hat{f}) &\leq \frac{t_k}{\theta_k^2} (f(x^{(k)}) - \hat{f}) + \frac{1}{2} \|v^{(k)} - \hat{x}\|_2^2 \leq \\
 &\leq \frac{t_k(1 - \theta_k)}{\theta_k^2} (f(x^{(k-1)}) - \hat{f}) + \frac{1}{2} \|v^{(k-1)} - \hat{x}\|_2^2 \leq \\
 &\leq \frac{t_{k-1}}{\theta_{k-1}^2} (f(x^{(k-1)}) - \hat{f}) + \frac{1}{2} \|v^{(k-1)} - \hat{x}\|_2^2 \leq \dots \\
 \dots &\leq \frac{t_1(1 - \theta_1)}{\theta_1^2} (f(x^{(0)}) - \hat{f}) + \frac{1}{2} \|v^{(0)} - \hat{x}\|_2^2 = \\
 &= \frac{1}{2} \|x^{(0)} - \hat{x}\|_2^2
 \end{aligned}$$

Preto platí:

$$f(x^{(k)}) - \hat{f} \leq \frac{\theta_k^2}{2t_{min}} \|x^{(0)} - \hat{x}\|_2^2 = \frac{2}{(k+1)^2 t_{min}} \|x^{(0)} - \hat{x}\|_2^2$$

□

Ukázali sme si, že rozdiel $f(x^{(k)}) - \hat{f}$ sa znižuje prinajmenšom rýchlosťou $1/k^2$. Pri oboch spôsoboch voľby kroku dosahuje algoritmus proximal gradient metódy s extrapoláciou zložitosť $O(1/\sqrt{\epsilon})$. Zrýchlená metóda však nezaručuje nerastúcosť účelovej funkcie $f(x^{(k)}) \leq f(x^{(k-1)})$ tak ako pôvodná verzia. Táto podmienka všeobecne nie je platná, metóda sa dostáva k optimálnej hodnote často skokovito. Z uvedeného dôvodu existuje ďalšia modifikácia proximal gradient metódy, ktorá nerastúcosť spĺňa. Nazýva sa spádová (descent) verzia zrýchlenej proximal gradient metódy.

4.2 Descent verzia zrýchlenej proximal gradient metódy

Nasledujúca modifikácia algoritmu zaručuje nerastúcosť účelovej funkcie.

$$\begin{aligned} z^{(k)} &= \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) \\ x^{(k)} &= \begin{cases} z^{(k)} & f(z^{(k)}) \leq f(x^{(k-1)}) \\ x^{(k-1)} & \text{inak} \end{cases} \\ v^{(k)} &= x^{(k-1)} + \frac{1}{\theta_k} (z^{(k)} - x^{(k-1)}) \\ y^{(k)} &= (1 - \theta_{k+1})x^{(k)} + \theta_{k+1}v^{(k)} \end{aligned}$$

Premenná θ_k ostáva definovaná ako v predošlej kapitole ako $\theta_k = \frac{2}{k+1}$. Všimnime si, že ak nie je splnená podmienka, premenná x_k ostáva konštantná až pokým sa nenájde lepšie suboptimálne riešenie.

4.3 Nová modifikácia proximal gradient metódy

Nasledujúcu úpravu sme vytvorili na základe poznatkov z predošlých kapitol, využili sme predovšetkým Vetu 3.6. (proximal gradient je spádová metóda) a vzťah (12). Modifikácia je podobná descent verzii, taktiež má vlastnosť nerastúcosť účelovej funkcie. Najväčší rozdiel je ten, že premenná x neostáva konštantnou, ale v každej iterácii sa mení, približuje sa optimálnemu riešeniu.

$$\begin{aligned} z^{(k)} &= \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) \\ x^{(k)} &= \begin{cases} z^{(k)} & f(z^{(k)}) \leq f(x^{(k-1)}) \\ \text{prox}_{t_k h} (x^{(k-1)} - t_k \nabla g(x^{(k-1)})) & \text{inak} \end{cases} \\ y^{(k)} &= x^{(k)} + \frac{k-1}{k+2} (x^{(k)} - x^{(k-1)}) \end{aligned}$$

Pokiaľ je podmienka nerastúcosť splnená, používa sa klasická zrýchlená metóda, kvôli jej rýchlosti konverencie. Ak daná podmienka neplatí, využívame pôvodnú proximal gradient metódu na predchádzajúce $x^{(k-1)}$, ktorá je vždy spádová. Táto úprava je založená na tom, že v klasickom proximal gradient algoritme je účelová funkcia nerastúca a vzdialenosť priebežného od optimálneho riešenia sa znižuje každou iteráciou, čo sme si ukázali vo Vete 3.6. a vzťahu (12). Vďaka tomu naše priebežné riešenie $x^{(k)}$ nebude nikdy stagnovať, od čoho očakávame rýchlejšie dosiahnutie suboptimálneho riešenia.

5 Numerické experimenty

V tejto kapitole otestujeme proximal gradient metódu a jej modifikácie na príkladoch. Experimenty sú vykonané v programe Matlab, zdrojové kódy sa nachádzajú v poslednej časti práce. Všetky pokusy prebiehali na zostave PC so štvorjadrovým procesorom Intel Core i7 2.66 GHz a 6GB RAM. Úlohy, na ktorých prebiehajú numerické experimenty nám generuje generátor úloh, ktorý popíšeme v ďalšej časti.

Všetky testy a pozorovania spracujeme na úlohe, ktorú si zvolíme nasledovného tvaru:

$$\text{Min}\left\{\frac{1}{2}\|Ax - b\|_2^2 + \|x\|_1, \quad x \in R^n\right\},$$

kde sa účelová funkcia dá rozložiť na diferencovateľnú funkciu $g(x) = \frac{1}{2}\|Ax - b\|_2^2$ a nediferencovateľnú časť $h(x)$, ktorou je v našom experimente jednotková norma $\|x\|_1$. Vzhľadom na štruktúru algoritmu proximal gradient metódy využijeme proximálny operátor funkcie $th(u) = t\|u\|_1$, ktorý sme odvodili v 3. kapitole.

$$\text{prox}_{th}(x)_i = \begin{cases} x_i - t & x_i \geq t \\ x_i + t & x_i \leq -t \\ 0 & |x_i| \leq t \end{cases}$$

Určovanie kroku t_k :

Metódy otestujeme na oboch spôsoboch voľby kroku t_k . V prípade kroku line search použijeme algoritmus z kapitoly 3.2. resp. v prípade zrýchlenej metódy z kapitoly 4. Konštantný krok volíme veľkosti $t_k = t = 1/L$, kde L je Lipschitzovská konštanta gradientu funkcie $g(x)$, čo je taktiež inšpirované line search nerovnosťou. Ukážeme si, že v prípade našej funkcie $g(x) = \frac{1}{2}\|Ax - b\|_2^2$ je táto konštanta rovná najväčšiemu vlastnému číslu matice $A^T A$, a preto volíme v algoritme metódy s fixným krokom krok dĺžky $t = 1/\lambda_{max}(A^T A)$.

Hľadáme teda konštantu L z Lipschitzovskej vlastnosti gradientu $\nabla g(x)$.

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in R^n$$

Po zderivovaní funkcie $g(x)$ podľa x zistíme, že gradient je tvaru:

$$\nabla g(x) = A^T(Ax - b)$$

Gradient následne dosadíme do vzťahu Lipschitzovskej vlastnosti a jednoduchými úpravami nájdeme L .

$$\|A^T Ax - A^T b - (A^T Ay - A^T b)\|_2 = \|A^T A(x - y)\|_2 \leq \lambda_{\max}(A^T A)\|x - y\|_2$$

Preto $L = \lambda_{\max}(A^T A)$.

Generátor úloh:

Nami vytvorený generátor úloh tvorí úlohy tak, že poznáme ich optimálne riešenie. Funguje nasledovne:

- 1.) Zvolíme rozmery m, n matice A , l - počet nenulových prvkov bodu optima, v - vzdialenosť počiatočného bodu $x^{(0)}$ od optima.
- 2.) Náhodné generovanie matice A , bodu optima $\hat{x} \in R^n$.
- 3.) Dopočítanie vektora $b \in R^m$ tak, aby optimálnym riešením úlohy bolo \hat{x} .
- 4.) Dopočítanie štartovacieho bodu $x^{(0)}$ podľa vzdialenosti od optimálneho riešenia.

Funkcie generátora "*počet nenulových prvkov*" a "*vzdialenosť od bodu optima*" sa dajú vypnúť hodnotou vstupu -1 , vtedy sa údaje tvoria úplne náhodne.

Keďže generujeme úlohy, u ktorých poznáme riešenie, použijeme ako stopovacie kritérium pri dosiahnutí ϵ presnosti $\|x^{(k)} - \hat{x}\|_2 < \epsilon$. Vo výsledkoch budeme uvádzať priemerný počet iterácií a priemernú časovú náročnosť v sekundách.

5.1 Vplyv počtu nenulových prvkov v optimálnom riešení na rýchlosť dosiahnutia riešenia

Tabuľky sú spracované zo vzorky 500 úloh rozmerov 500x200 na každý riadok.

Všimnime si, že line search metódy dosahujú suboptimálne riešenie za menej iterácií ako metódy s konštantným krokom, ale sú časovo náročnejšie, preto je výhodnejšie používať fixný krok.

Výsledok experimentu: Metódy potrebujú viac iterácií a času na nájdenie riešenia, ak je viac nenulových prvkov v optimálnom riešení.

m = 500		$\epsilon = 0.0001$		# pokusov = 500		voľba kroku: konštantný = 1/L			
n = 200		$\ x^0 - \hat{x}\ = 5$		max # it. = 500					
metóda:	Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG		
l	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií	
0	0.0446	67.028	0.0305	45.674	0.0337	44.772	0.0278	35.596	
40	0.0518	77.724	0.0396	59.68	0.0442	58.904	0.0325	41.132	
80	0.0578	87.95	0.0459	70.46	0.052	70.272	0.0348	44.946	
100	0.0616	94.386	0.0498	76.666	0.0569	76.5	0.0368	46.786	
120	0.0679	101.53	0.0562	83.62	0.0643	83.916	0.0381	47.44	
160	0.0781	120.1	0.0647	100.04	0.0735	100.96	0.04	52.664	
200	0.0967	146.5	0.0802	120.986	0.0927	123.494	0.0441	56.212	

Obr. 11: Metódy s konštantným krokom

m = 500		$\epsilon = 0.0001$		# pokusov = 500		voľba kroku: LINESEARCH			
n = 200		$\ x^0 - \hat{x}\ = 5$		max # it. = 500					
metóda:	Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG		
l	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií	
0	0.0538	46.688	0.0512	41.906	0.054	40.996	0.0479	32.244	
40	0.0596	54.296	0.0607	54.938	0.0652	54.67	0.0523	37.072	
80	0.0642	61.21	0.0683	65.546	0.0729	65.19	0.0555	40.936	
100	0.0676	65.2	0.0733	71.158	0.0788	71.248	0.0568	43.326	
120	0.0734	70.414	0.0798	77.972	0.0864	78.156	0.0599	44.684	
160	0.0792	83.526	0.0878	93.648	0.0965	94.634	0.0617	49.374	
200	0.0977	104.688	0.1056	113.972	0.1198	118.646	0.0659	52.6	

Obr. 12: Metódy s line search krokom

5.2 Vplyv zmeny požadovanej presnosti na rýchlosť dosiahnutia riešenia

Tabuľky sú spracované zo vzorky 1000 úloh rozmerov 400x100 na každý riadok.

Výsledok experimentu: Zvyšovanie presnosti spôsobuje zvyšovanie počtu iterácií a časovej náročnosti.

metóda:		Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG	
ϵ	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií	
0.1	0.0075	22.549	0.0049	14.748	0.0057	14.826	0.0053	13.495	
0.01	0.0128	37.856	0.0092	27.028	0.0107	27.181	0.0086	21.127	
0.005	0.0138	42.313	0.0108	33.051	0.0127	33.367	0.0092	23.275	
0.001	0.0167	52.744	0.0145	45.328	0.0168	45.376	0.011	27.929	
0.0005	0.0184	56.783	0.0167	50.847	0.0194	51.386	0.0117	29.204	
0.0001	0.0224	67.741	0.0219	65.849	0.0256	66.542	0.0134	32.321	
0.00005	0.0237	71.995	0.0237	72.03	0.0278	72.744	0.0142	34.225	

Obr. 13: Metódy s konštantným krokom

metóda:		Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG	
ϵ	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií	
0.1	0.0155	16.304	0.0144	12.751	0.015	12.632	0.0147	11.874	
0.01	0.0201	27.338	0.0196	24.993	0.0211	25.422	0.0192	20.485	
0.005	0.0206	30.274	0.021	30.452	0.0222	29.915	0.0197	22.949	
0.001	0.0232	37.684	0.0255	42.555	0.0275	42.883	0.0215	27.467	
0.0005	0.0245	40.645	0.0278	48.243	0.03	48.236	0.0225	29.208	
0.0001	0.028	48.431	0.0342	63.206	0.0372	63.346	0.0249	32.637	
0.00005	0.0291	51.796	0.0367	69.46	0.0403	69.841	0.0258	34.376	

Obr. 14: Metódy s line search krokom

5.3 Vplyv rozmerov úlohy na rýchlosť dosiahnutia riešenia

Tabuľky sú spracované zo vzorky 500 úloh na každý riadok.

Všimnime si, že pre veľké rozmery úloh metódy s line search krokom nedokázali nájsť optimum v limite 500 iterácií, čo je pravdepodobne spôsobené zaokrúhlovacími chybami a tým, že line search krok sa už veľmi zmenšil a metódy sa tak nedokážu pohnúť k optimálnej hodnote.

Výsledok experimentu: Zvyšovanie rozmerov úloh spôsobuje zvyšovanie počtu iterácií a časovej náročnosti.

metóda:		Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG	
m	n	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií
10	4	0.0023	62.668	0.0018	45.034	0.0022	40.706	0.0012	19.39
50	20	0.0096	124.674	0.0077	98.122	0.01	103.634	0.0041	38.336
100	40	0.022	142.294	0.0186	119.1	0.0227	125.804	0.0088	45.562
200	80	0.0439	157.458	0.0375	140.108	0.0478	145.256	0.0186	52.754
500	200	0.1129	173.074	0.1071	162.696	0.1252	165.448	0.0511	63.092
1000	400	0.2482	182.402	0.2361	176.876	0.275	177.682	0.1216	75.598
2000	800	0.5033	190.994	0.4926	189.704	0.9016	187.698	0.4256	86.376
3000	1200	0.8742	196.06	0.8815	196.774	1.7191	193.604	0.8926	98.23

Obr. 15: Metódy s konštantným krokom

metóda:		Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG	
m	n	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií
10	4	0.0045	61.244	0.0037	46.044	0.004	41.358	0.0024	19.55
50	20	0.0124	103.19	0.0111	89.082	0.0132	93.998	0.0066	35.808
100	40	0.0248	115.998	0.0223	104.696	0.0247	110.312	0.0116	41.352
200	80	0.0504	128.878	0.0485	121.672	0.0551	125.912	0.0269	47.416
500	200	0.1093	120.588	0.1342	154.37	0.149	156.884	0.0722	59.564
1000	400	0.2839	154.164	0.376	214.642	0.4771	243.218	0.1679	72.02
2000	800	2.4567	449.924	2.6791	486.916	3.6203	493.894	0.9425	117.504
3000	1200	5.0364	498.264	5.0671	499	7.2229	499	2.9538	194.88

Obr. 16: Metódy s line search krokom

5.4 Vplyv vzdialenosti štartovacieho bodu od optima na rýchlosť dosiahnutia riešenia

Tabuľky sú spracované zo vzorky 500 úloh rozmerov 500x200 na každý riadok.

Výsledok experimentu: Zvyšovanie vzdialenosti štartovacieho bodu od optima spôsobuje zvyšovanie počtu iterácií a časovej náročnosti.

m = 500		ε = 0.0001		# pokusov = 500		voľba kroku: konštantný = 1/L			
n = 200				max # it. = 500					
metóda:	Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG		
$\ x^0 - \hat{x}\ $	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií	
0.5	0.0626	95.162	0.0451	68.596	0.0516	69.196	0.0346	44.704	
1	0.0689	106.092	0.0527	80.406	0.0601	80.906	0.0371	48.308	
2	0.0764	116.56	0.0603	92.086	0.0695	93.138	0.0399	51.114	
5	0.0867	132.56	0.0726	110.562	0.0827	112.07	0.042	54.206	
10	0.0961	143.762	0.0826	124.076	0.0949	126.202	0.045	57.45	
20	0.103	154.858	0.0926	139.056	0.107	141.19	0.047	58.908	
50	0.1141	169.724	0.1065	158.536	0.1227	161.288	0.0504	62.262	
100	0.1213	182.418	0.1161	175.042	0.1337	178.618	0.0511	64.51	
200	0.1303	194.094	0.1297	192.12	0.1492	195.406	0.0563	68.516	
1000	0.1477	222.404	0.1552	233.078	0.1778	236.968	0.0591	73.01	

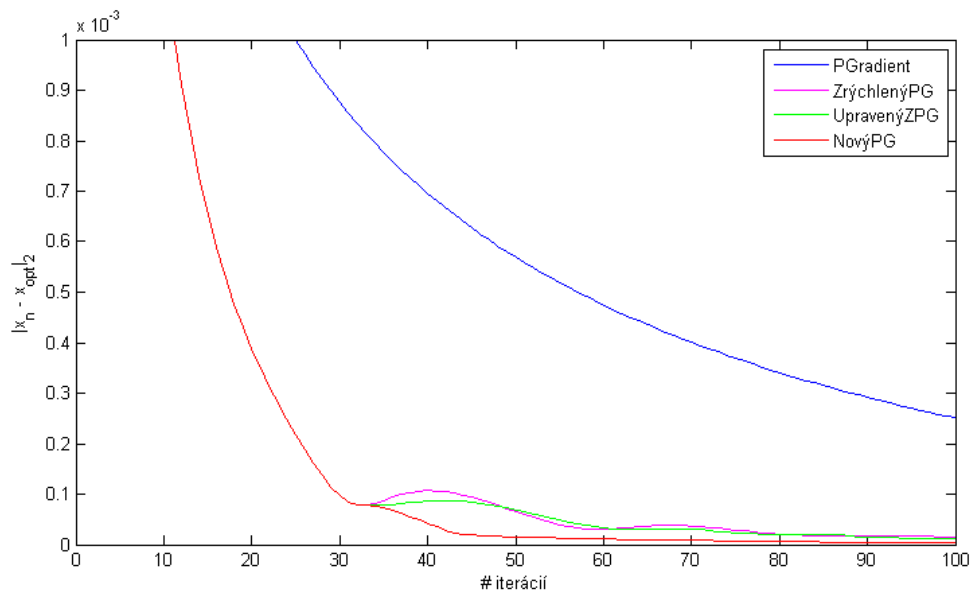
Obr. 17: Metódy s konštantným krokom

m = 500		ε = 0.0001		# pokusov = 500		voľba kroku: LINESEARCH			
n = 200				max # it. = 500					
metóda:	Proximal Gradient		Zrýchlený PG		Spádový PG		Nová verzia PG		
$\ x^0 - \hat{x}\ $	čas	# iterácií	čas	# iterácií	čas	# iterácií	čas	# iterácií	
0.5	0.0685	66.398	0.0669	63.168	0.0725	63.468	0.055	41.212	
1	0.0728	73.668	0.0741	74.398	0.0811	75.116	0.0568	43.908	
2	0.0793	80.904	0.0853	87.762	0.0921	87.02	0.0616	47.97	
5	0.0875	92.576	0.0964	103.822	0.1071	105.174	0.063	50.53	
10	0.0942	100.514	0.1081	117.032	0.1185	118.95	0.0668	54.172	
20	0.1005	108.79	0.1183	131.352	0.1319	133.456	0.0688	55.504	
50	0.11	118.922	0.1344	150.058	0.1508	152.908	0.0729	58.836	
100	0.115	128.936	0.1436	166.632	0.161	169.712	0.0739	61.002	
200	0.1247	136.632	0.1603	183.19	0.1781	186.122	0.0791	64.76	
1000	0.1363	154.79	0.189	223.304	0.2106	226.542	0.0832	69.062	

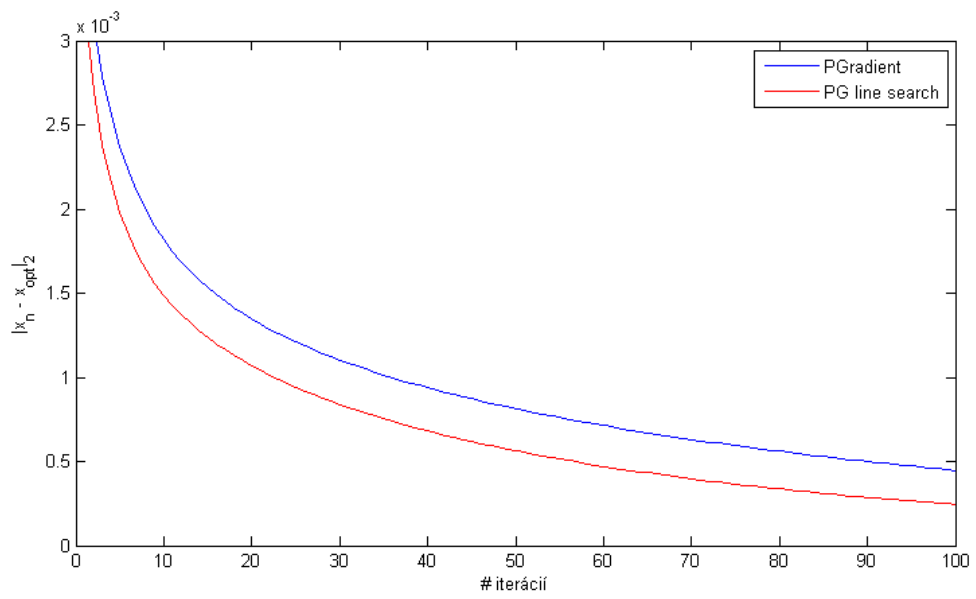
Obr. 18: Metódy s line search krokom

5.5 Porovnanie metód

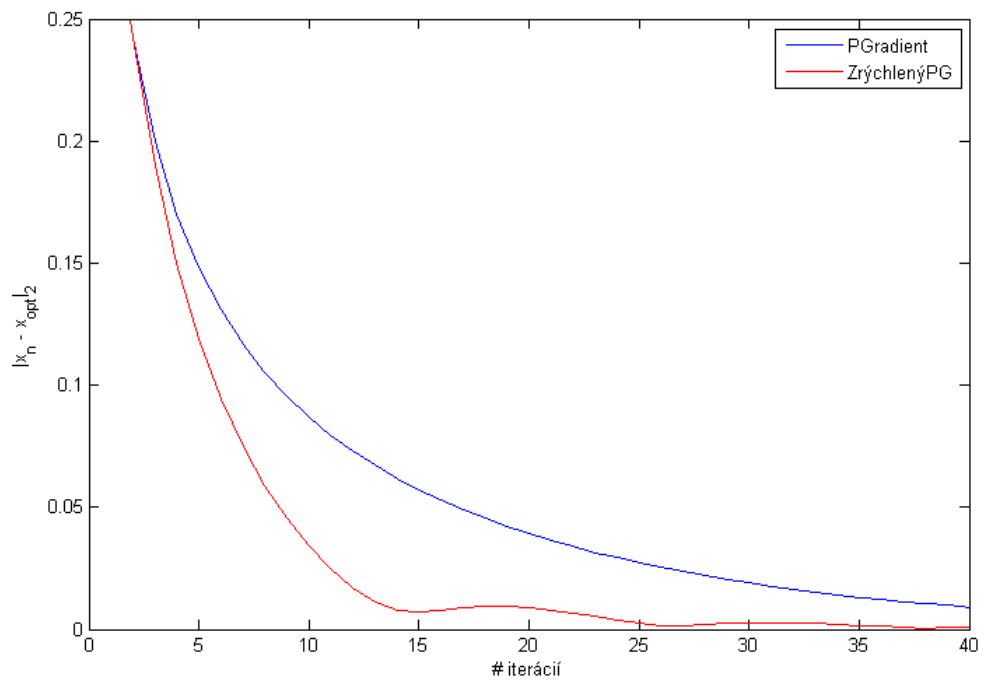
Na nasledujúcich obrázkoch si ukážeme vývoj vzdialenosti priebežného riešenia $x^{(k)}$ od bodu optima \hat{x} , čiže vzťah $\|x^{(k)} - \hat{x}\|_2$ v závislosti od iterácie k . Pomocou tohto vývoja porovnáme rôzne modifikácie proximal gradient metódy, prípadne poukážeme na špeciálne správanie metód.



Obr. 19: Porovnanie metód s konštantným krokom



Obr. 20: Porovnanie metódy s konštantným krokom a line search krokom



Obr. 21: Ukážka "skokov" zrýchlenej metódy

6 Záver

V bakalárskej práci sme sa venovali proximal gradient metóde konvexného programovania, ktorá sa používa na minimalizáciu neohraničeného problému s nediferencovateľnou účelovou funkciou špeciálnej štruktúry.

V jednotlivých častiach sme sa oboznámili s rôznymi poznatkami z oblasti konvexných funkcií a konvexného programovania, zadefinovali sme subgradienty a objasnili sme podmienky optimality nediferencovateľných funkcií. Teoreticky sme spracovali proximal gradient metódu, proximálny operátor aj s jeho vlastnosťami. Predstavili sme si rôzne verzie proximal gradient metódy - základná verzia, zrýchlená verzia a späťdová verzia. Práca obsahuje analýzu konvergenzie prvých dvoch pre oba spôsoby voľby kroku. Navrhli sme novú modifikáciu s účelom ďalšieho vylepšenia metódy, ktorú sme podrobili testom. V časti numerické experimenty sme testovali metódy na úlohách, ktoré generoval nami navrhnutý generátor úloh, vďaka ktorému sme optimálne riešenia úloh poznali. Testovali sme vplyv nenulových prvkov v optime na počet iterácii a časovú náročnosť, výsledok hovorí o tom, že čím je viac nenulových prvkov tým metóda pomalšie nachádza optimum. Otestovali sme taktiež účinnosť metód na vysokorozmerné úlohy rozmerov až do 3000x1200 a zistili sme, že sú efektívne. Metódy poskytovali v experimentoch veľmi dobré výsledky.

7 Literatúra

Literatúra

- [1] L. Vandenberghe: Algorithms for large-scale convex optimization - lecture notes, course at Technical University of Denmark, August 23–27, 2010

<http://www.ee.ucla.edu/~vandenbe/shortcourses.html>

- [2] M. Hamala: Konspekty k prednáškam z nelineárneho programovania

- [3] S. Boyd and L. Vandenberghe, Subgradients, Notes for EE364b, Stanford University, 2007

- [4] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge university press, 2004

www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

- [5] D. P. Bertsekas, Lecture slides on convex analysis and optimization based on 6.253 class lectures at the Mass. Institute of technology Cambridge, spring 2010

web.mit.edu/dimitrib/www/Convex_Slides_2010.pdf

- [6] L. Vandenberghe: First-order methods - lecture notes, OPTEC/K.U.Leuven, July 6–7, 2009

<http://www.ee.ucla.edu/~vandenbe/shortcourses.html>

8 Príloha - zdrojové kódy programov

```
%%%%%%%%%%%%%%
%GENERATOR ULOH%
%%%%%%%%%%%%%%
function [G,h,A,b,x_opt,x0]=generatorm1(m,n,l,v)
%vstupy rozmery m,n; pocet nenulovych prvkov l; vzdialenost od optima

%tvorba optimalneho riesenia
if(l~-1) %hodnota l=-1 vypina volbu pocetu nenulovych prvkov
    xnen=(2*round(rand(1,1))-1).*round(4*rand(1,1)+1);
    %tvorba nenulovych z (-5,-1) a (1,5)
    x_opt=[xnen;zeros(n-l,1)]; %pridanie nul
    x_opt=x_opt(randperm(n)); %premiesanie
else
    x_opt=round(10*(rand(n,1)-0.5));
end

%tvorba A
A=round(10*(rand(m,n)-0.5));
G=A'*A;

%dopocitanie vektora b
h=zeros(n,1);
for i=1:n
    if(x_opt(i)~=0)
        h(i)=G(i,:)*x_opt+x_opt(i)/abs(x_opt(i));
        %dotvorenie h aby bolo optimum x_opt
    else
        h(i)=G(i,:)*x_opt;
    end
end
b=A'\h; %riesenim sustavy dostaneme b

%vzdialenost od optima
if(v~-1) %hodnota v=-1 vypne funkciu vzdialenost od optima
    dd=zeros(n,1);
    sumdd=0;
    for i=1:n-1
        dd(i)=(2*rand()-1)*v/sqrt(n);
        sumdd=sumdd+dd(i)^2;
    end
    dd(n)=sqrt(v^2-sumdd);
    dd=dd(randperm(n));
    x0=x_opt+dd;
else
```

```

    x0=round(10*(rand(n,1)-0.5));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%proximalny operator funkcie t*(jednotkova norma)%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [x]=proxnorm1(x,t)
m=size(x);
for i=1:m
    if(x(i)>=t)
        x(i)=x(i)-t;
    end
    if(abs(x(i))<=t)
        x(i)=0;
    end
    if(x(i)<=-t)
        x(i)=x(i)+t;
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 8 VERZII PROXIMAL GRADIENT METODY %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [d1,d2,d3,d4,d5,d6,d7,d8]=norm1(m,n,l,v,beta,ls,mmax)
%vstup rozmery m,n; pocet nenulovych prvkov l; vzdialenost od optima;
%beta v line search; pouzit/nepouzit line search; max pocet iteracii
clc;
[G,h,A,b,x_opt,x0]=generatornorm1(m,n,l,v);
e=0.0001;

%KONSTANTNY KROK
t=1/(max(eig(G)));

t1 = tic;
%Proximal Gradient
x=x0;
delta=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    gg=G*x-h;
    [x]=proxnorm1(x-t*gg,t);
    delta(k)=norm(x-x_opt);
    k=k+1;
end

```

```

end
d1=delta(k-1);
Cas1=toc(t1)

t1 = tic;
%Zrychleny PG
x=x0;
y=x0;
delta2=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    xs=x;
    gg=G*y-h;
    [x]=proxnorm1(y-t*gg,t);
    y=x+((k-1)/(k+2))*(x-xs);
    delta2(k)=norm(x-x_opt);
    k=k+1;
end

d2=delta2(k-1);
Cas2=toc(t1)

t1 = tic;
%DESCENT PG
x=x0;
y=x0;
delta3=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    xs=x;
    gg=G*y-h;
    [z]=proxnorm1(y-t*gg,t);
    fs=(1/2)*norm(A*xs-b)^2+sum(abs(xs));
    fn=(1/2)*norm(A*z-b)^2+sum(abs(z));
    if(fn<=fs)
        x=z;
    end
    v=xs+((k+1)/2)*(z-xs);
    y=(k/(k+2))*x+(2/(k+2))*v;
    delta3(k)=norm(x-x_opt);
    k=k+1;
end

d3=delta3(k-1);
Cas3=toc(t1)

```

```

t1 = tic;
%NOVA MODIFIKACIA PG
x=x0;
y=x0;
delta4=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    xs=x;
    gg=G*y-h;
    [z]=proxnorm1(y-t*gg,t);
    fs=(1/2)*norm(A*xs-b)^2+sum(abs(xs));
    fn=(1/2)*norm(A*z-b)^2+sum(abs(z));
    if(fn<=fs)
        x=z;
    else
        gg=G*xs-h;
        x=proxnorm1(xs-t*gg,t);
    end
    y=x+((k-1)/(k+2))*(x-xs);
    delta4(k)=norm(x-x_opt);
    k=k+1;
end

d4=delta4(k-1);
Cas4=toc(t1)

%LINE SEARCH KROK
d5=0; d6=0; d7=0; d8=0;
if(ls==1)
t1 = tic;
%Proximal Gradient
tk=2;
x=x0;
delta5=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    gg=G*x-h;
    [tk,x]=lsearch(A,b,gg,tk,x,beta);
    delta5(k)=norm(x-x_opt);
    k=k+1;
end

d5=delta5(k-1);
Cas5=toc(t1)

```



```

t1 = tic;
%Zrychleny PG
tk=2;
x=x0;
y=x0;
delta6=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    xs=x;
    gg=G*y-h;
    [tk,x]=lsearch(A,b,gg,tk,y,beta);
    y=x+((k-1)/(k+2))*(x-xs);
    delta6(k)=norm(x-x_opt);
    k=k+1;
end

d6=delta6(k-1);
Cas6=toc(t1)

```

```

t1 = tic;
%DESCENT PG
tk=2;
x=x0;
y=x0;
delta7=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    xs=x;
    gg=G*y-h;
    [tk,z]=lsearch(A,b,gg,tk,y,beta);
    fs=(1/2)*norm(A*xs-b)^2+sum(abs(xs));
    fn=(1/2)*norm(A*z-b)^2+sum(abs(z));
    if(fn<=fs)
        x=z;
    end
    v=xs+((k+1)/2)*(z-xs);
    y=(k/(k+2))*x+(2/(k+2))*v;
    delta7(k)=norm(x-x_opt);
    k=k+1;
end

d7=delta7(k-1);
Cas7=toc(t1)

```

```

t1 = tic;
%NOVA MODIFIKACIA PG
tk=2;
x=x0;
y=x0;
delta8=ones(mmax,1);
k=1;
while((norm(x-x_opt) > e) && (k < mmax))
    xs=x;
    gg=G*y-h;
    [tk,z]=lsearch(A,b,gg,tk,y,beta);
    fs=(1/2)*norm(A*xs-b)^2+sum(abs(xs));
    fn=(1/2)*norm(A*z-b)^2+sum(abs(z));
    if(fn<=fs)
        x=z;
    else
        gg=G*xs-h;
        [tk,x]=lsearch(A,b,gg,tk,xs,beta);
    end
    y=x+((k-1)/(k+2))*(x-xs);
    delta8(k)=norm(x-x_opt);
    k=k+1;
end

d8=delta8(k-1);
Cas8=toc(t1)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LINE SEARCH spojeny s proximal iteraciou%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [tk,x]=lsearch(A,b,gg,tk,xs,beta)
prox=proxnorm1(xs-tk*gg,tk);
Gt=1/tk*(xs-prox);
g=1/2*norm(A*xs-b)^2;
gprox=1/2*norm(A*prox-b)^2;
while(gprox>g-tk*gg'*Gt+tk/2*norm(Gt)^2)
    tk=beta*tk;
    prox=proxnorm1(xs-tk*gg,tk);
    gprox=1/2*norm(A*prox-b)^2;
    Gt=1/tk*(xs-prox);
end
x=prox;

```