A Gentle Introduction to *Primal-Dual Method for Convex Optimization*

Convex Optimization and Applications

Dr. Ir. Valentin Leplat and Prof. Anh Huy Phan
Skoltech - CAIT

Outline

Introduction

Supporting materials

- The content of this presentation is based on the following work(s):

1. S. Wang. *A Tutorial on Primal-Dual Algorithm*, University of Waterloo, March 2016.
2. ADMM Lectures previously given in the frame of *Convex Optimization and Applications* Skoltech course.
3. MIT online course. *Convex Analysis and Optimization: Lecture 8*, Spring 2010.
4. N. Parikh and S. Boyd. *Proximal Algorithms.* Foundations and Trends in Optimization, Vol. 1, No. 3, Stanford 2013.
5. A. Beck and M. Teboulle. *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM J. Imaging Sciences, Vol. 2, No. 1, pp. 183–202, 2013.
6. Y.Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer Academic Publishers, 2004.
7. J. Zhu, S. Rosset, T. Hastie and R. Tibshirani. *1-norm Support Vector Machines.* Neurips 2003.
8. A. Chambolle and T. Pock. *A first-order primal-dual algorithm for convex problems with applications to imaging.* Journal of Mathematical Imaging and Vision 40.1: 120-145, 2011.

Road map and milestones

- The first part is dedicated to recall basic assumptions on the class of Problems we deal with, and key elements for using Primal-Dual methods such as the *Convex Conjugate* of a function, and the *Proximal Operator* of a function.
- Next, we quickly introduce the family of Proximal Gradient Descent Methods. These methods are purely *Primal* methods, however important insights and limitations from them will be used for motivating the use of *Primal-Dual* Methods.
- Then, we formally present the class of Problems well-suited for Primal-Dual Methods, and we present the State-of-the-Art Primal-Dual algorithms for solving such Problems.
- Finally, we showcase the presented methods on problems of interests in
    1. signal processing: denoising with Total Variation, binary classification and linear programming.
    2. machine learning: the so-called *Personalized Federated Learning.*

Generic Problem Formulation

Recall the following generic convex optimization problem:

$$\arg\min_{x \in \mathcal{X}} f(x) \tag{1}$$

with **Assumptions A.**:

1. real Hilbert spaces: $\mathcal{X}$ (here, finite-dimension Hilbert space)
2. $f : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ is *an extended real valued*, *proper*, *closed*, and *convex* function.

Recall of useful materials for us

Semi-continuity

A property of extended real-valued functions that is weaker than continuity.

- A function $f(x)$ is called lower(upper) semi-continuous at point x_0 if function values for arguments near x_0 are either close to $f(x_0)$ or greater than (less than) $f(x_0)$:

$$f(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \quad ? \qquad\qquad g(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad ?$$

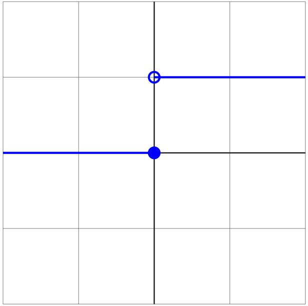
Recall of useful materials for us

Semi-continuity

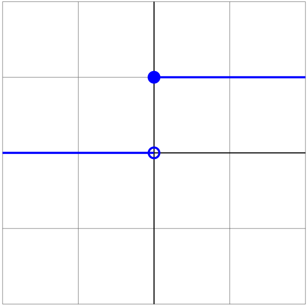
A property of extended real-valued functions that is weaker than continuity.

- A function $f(x)$ is called lower(upper) semi-continuous at point x_0 if function values for arguments near x_0 are either close to $f(x_0)$ or greater than (less than) $f(x_0)$:

$$f(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \quad ?$$



$$g(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad ?$$



Recall of useful materials for us

Semi-continuity

A property of extended real-valued functions that is weaker than continuity.

- A function $f(x)$ is called lower(upper) semi-continuous at point x_0 if function values for arguments near x_0 are either close to $f(x_0)$ or greater than (less than) $f(x_0)$:

Recall of useful materials for us

Semi-continuity

A property of extended real-valued functions that is weaker than continuity.

- A function $f(x)$ is called lower(upper) semi-continuous at point x_0 if function values for arguments near x_0 are either close to $f(x_0)$ or greater than (less than) $f(x_0)$:
- Floor function $\lfloor x \rfloor$ is upper semi-continuous, $\lceil x \rceil$ is lower semi-continuous.

Recall of useful materials for us

Semi-continuity

A property of extended real-valued functions that is weaker than continuity.

- A function $f(x)$ is called lower(upper) semi-continuous at point x_0 if function values for arguments near x_0 are either close to $f(x_0)$ or greater than (less than) $f(x_0)$:
- Floor function $\lfloor x \rfloor$ is upper semi-continuous, $\lceil x \rceil$ is lower semi-continuous.
- The indicator function of any open set is upper semicontinuous. The indicator function of a closed set is lower semicontinuous.
  Attention to the definition of the *indicator function*: for a set $Q \subseteq \mathcal{X}$, we define the indicator function as follows:

$$\mathcal{I}_Q(x) := \begin{cases} 0 & \text{if } x \in Q \\ +\infty & \text{otherwise} \end{cases}$$

Recall of useful materials for us

Semi-continuity

A property of extended real-valued functions that is weaker than continuity.

- A function $f(x)$ is called lower(upper) semi-continuous at point x_0 if function values for arguments near x_0 are either close to $f(x_0)$ or greater than (less than) $f(x_0)$:
- Floor function $\lfloor x \rfloor$ is upper semi-continuous, $\lceil x \rceil$ is lower semi-continuous.
- The indicator function of any open set is upper semicontinuous. The indicator function of a closed set is lower semicontinuous.
  Attention to the definition of the *indicator function*: for a set $Q \subseteq \mathcal{X}$, we define the indicator function as follows:

$$\mathcal{I}_Q(x) := \begin{cases} 0 & \text{if } x \in Q \\ +\infty & \text{otherwise} \end{cases}$$

- Used to convert inequality constraints to objective function.

Recall of useful materials for us

Lipschitz continuity

- A function $f(x)$ is called L-Lipschitz continuous on $\mathcal{X}$ with constant if :

$$|f(x) - f(y)| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{X}$$

  where constant L is an upper bound to the maximum steepness of $f(x)$.
- Stronger than continuous, weaker than continuously differentiable.
- **Example**: $f = |x|$ is Lipschitz continuous but not continuously differentiable (in 0).

Recall of useful materials for us

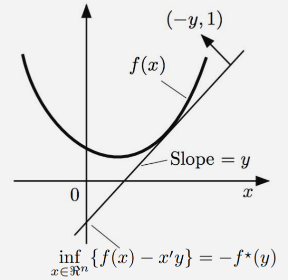
Convex conjugate

- The convex conjugate $f^*(y)$ of a function $f(x)$ is defined as:

$$f^*(y) := \sup_{x \in \mathcal{X}} \langle y, x \rangle - f(x)$$

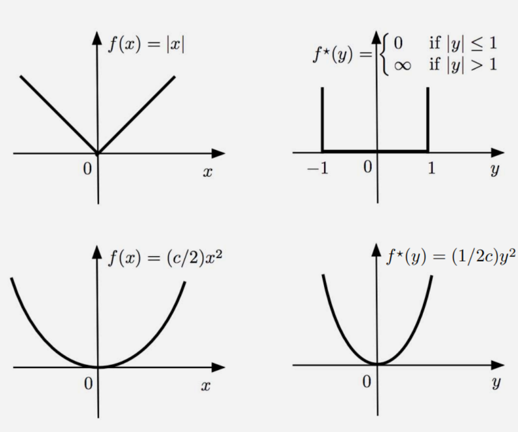
- Graphical Intuition: consider a function f and its epigraph:

1. Nonvertical hyperplanes supporting $\text{epi}(f) = \{(x,t) \in \mathcal{X} \times \mathbb{R} | f(x) \le t\}$
2. Crossing points of vertical axis !



Recall of useful materials for us

Convex conjugate - Examples



▸ Visualizing Convex Conjugates

Recall of useful materials for us

Proximal operator

- The **proximal** operator (or proximal mapping) of a convex function Φ with parameter $\gamma > 0$ is:

$$\mathbf{prox}_{\gamma,\Phi} : v \in \mathcal{X} \to \arg\min_{x \in \mathcal{X}} \{\Phi(x) + \frac{1}{2\gamma}\|x - v\|^2\}$$

- Φ can be nonsmooth, have embedded constraints, ...
- evaluating $\mathbf{prox}_{\gamma,\Phi}$ involves solving a convex optimization problem,
- but often has analytic solution: ℓ_1*-norm, log-barrier function, Quadratic function,* ℓ_0*-norm.*
- or polynomial time algorithm : $\Phi(X) := \|X\|_*$ *(the nuclear norm)* $\to$ *use SVD,* $O(n^3)$.
- or simple linear-time algorithm $O(n)$: *1-D Total Variation Operator, proj. onto the unit simplex*

Proximal operator: generalization of projection operator

$$\mathbf{prox}_{\gamma,\Phi} : v \in \mathcal{X} \rightarrow \arg\min_{x \in \mathcal{X}} \{\Phi(x) + \frac{1}{2\gamma} \|x - v\|^2\}$$

Proximal operator: generalization of projection operator

$$\mathbf{prox}_{\gamma,\Phi} : v \in \mathcal{X} \to \arg\min_{x\in\mathcal{X}}\{\Phi(x) + \frac{1}{2\gamma}\|x - v\|^2\}$$

- Φ is the indicator function:

$$I_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if} \quad x \in \mathcal{C}, \\ +\infty & \text{if} \quad x \notin \mathcal{C}, \end{cases}$$

where $\mathcal{C} \subset \mathcal{X}$ is a closed nonempty convex set.

Proximal operator: generalization of projection operator

$$\mathbf{prox}_{\gamma,\Phi} : v \in \mathcal{X} \to \arg\min_{x\in\mathcal{X}}\{\Phi(x) + \frac{1}{2\gamma}\|x-v\|^2\}$$

- Φ is the indicator function:

$$I_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if} \quad x \in \mathcal{C}, \\ +\infty & \text{if} \quad x \notin \mathcal{C}, \end{cases}$$

  where $\mathcal{C} \subset \mathcal{X}$ is a closed nonempty convex set. The proximal operator becomes:

$$\begin{aligned} \mathbf{prox}_{\gamma,\Phi}(v) &:= \underset{x\in\mathcal{X}}{\operatorname{argmin}} \left(\Phi(x) + \frac{1}{2\gamma}\|x-v\|_2^2\right) \\ \iff &:= \underset{x\in\mathcal{C}}{\operatorname{argmin}} \left(\gamma\underbrace{\Phi(x)}_{=0} + \frac{1}{2}\|x-v\|_2^2\right) \\ \iff &= \underset{x\in\mathcal{C}}{\operatorname{argmin}} \left(\|x-v\|_2^2\right) = \Pi_{\mathcal{C}}(v) \end{aligned}$$

Proximal operator: generalization of projection operator

$$\mathbf{prox}_{\gamma,\Phi} : v \in \mathcal{X} \rightarrow \arg\min_{x \in \mathcal{X}} \{\Phi(x) + \frac{1}{2\gamma}\|x - v\|^2\}$$

- Φ is the indicator function:

$$I_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if} \quad x \in \mathcal{C}, \\ +\infty & \text{if} \quad x \notin \mathcal{C}, \end{cases}$$

where $\mathcal{C} \subset \mathcal{X}$ is a closed nonempty convex set. The proximal operator becomes:

$$\begin{aligned} \mathbf{prox}_{\gamma,\Phi}(v) &:= \underset{x \in \mathcal{X}}{\operatorname{argmin}} \left(\Phi(x) + \frac{1}{2\gamma}\|x - v\|_2^2 \right) \\ \iff &:= \underset{x \in \mathcal{C}}{\operatorname{argmin}} \left(\gamma \underbrace{\Phi(x)}_{=0} + \frac{1}{2}\|x - v\|_2^2 \right) \\ \iff &= \underset{x \in \mathcal{C}}{\operatorname{argmin}} \left(\|x - v\|_2^2 \right) = \Pi_{\mathcal{C}}(v) \end{aligned}$$

Hence the proximal operator of Φ reduces to Euclidean projection onto $\mathcal{C}$.

...Proximal operators can thus be viewed as generalized projections !

Proximal operator: some properties

- if Φ is closed and convex, then $\mathbf{prox}_{\gamma,\Phi}$ exists and is unique for all $x \in \mathcal{X}$.

Proximal operator: some properties

- if Φ is closed and convex, then $\mathbf{prox}_{\gamma,\Phi}$ exists and is unique for all $x \in \mathcal{X}$.
- **conjugate** (Moreau identity):

$$\mathbf{prox}_{\gamma,\Phi^*}(v) = v - \gamma \mathbf{prox}_{\frac{1}{\gamma},\Phi}(\frac{v}{\gamma})$$

where $\Phi^*(u) := \sup_{x \in \mathbf{dom}\Phi}\{\langle u, x\rangle - \Phi(x)\}$ (the conjugate of Φ), with $\langle ., .\rangle$ the inner product associated to Hilbert Space

Proximal operator: some properties

- if Φ is closed and convex, then $\mathbf{prox}_{\gamma,\Phi}$ exists and is unique for all $x \in \mathcal{X}$.
- **conjugate** (Moreau identity):

$$\mathbf{prox}_{\gamma,\Phi^*}(v) = v - \gamma \mathbf{prox}_{\frac{1}{\gamma},\Phi}(\frac{v}{\gamma})$$

  where $\Phi^*(u) := \sup_{x \in \mathbf{dom}\Phi}\{\langle u, x\rangle - \Phi(x)\}$ (the conjugate of Φ), with $\langle ., .\rangle$ the inner product associated to Hilbert Space
- **seperable sum**: $\Phi(x) := \sum_i^N \Phi_i(x_i)$, then:

$$(\mathbf{prox}_{\gamma,\Phi}(v))_i := \mathbf{prox}_{\gamma,\Phi_i}(v_i)$$

Proximal operator: some properties

- if Φ is closed and convex, then $\mathbf{prox}_{\gamma,\Phi}$ exists and is unique for all $x \in \mathcal{X}$.
- **conjugate** (Moreau identity):

$$\mathbf{prox}_{\gamma,\Phi^*}(v) = v - \gamma \mathbf{prox}_{\frac{1}{\gamma},\Phi}(\frac{v}{\gamma})$$

  where $\Phi^*(u) := \sup_{x \in \mathbf{dom}\Phi}\{\langle u, x\rangle - \Phi(x)\}$ (the conjugate of Φ), with $\langle ., .\rangle$ the inner product associated to Hilbert Space
- **seperable sum**: $\Phi(x) := \sum_i^N \Phi_i(x_i)$, then:

$$(\mathbf{prox}_{\gamma,\Phi}(v))_i := \mathbf{prox}_{\gamma,\Phi_i}(v_i)$$

- **Fixed point**: the point $x^\star$ minimizes Φ if and only if $x^\star$ is a fixed point:

$$x^\star = \mathbf{prox}_{\Phi}(x^\star)$$

Proximal gradient method

Proximal gradient method

We consider the following convex optimization problem:

$$\min_{x \in \mathcal{X}} f(x) + g(x)$$

- f is a convex L_f-smooth function, that is the gradient ∇f is Lipschitz continuous with constant L_f
- g is proper closed convex, possibly nondifferentiable; *proximal operator* of g tractable and efficiently computable.
- rules out many methods, e.g. conjugate gradient
- **Example:** lasso problem:

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1$$

Proximal gradient method

Two terms: *Proximal* and *Gradient Descent*

Proximal gradient method

Two terms: *Proximal* and *Gradient Descent*

1. *Gradient Descent*: say we want to solve:

$$\min_{x \in Q \subseteq \mathcal{X}} f(x)$$

where Q is a simple convex subset of $\mathcal{X}$. Equivalently:

$$\min_{x \in \mathcal{X}} f(x) + \mathcal{I}_Q(x)$$

Classically: perform a gradient descent step and a *Correction/Projection* step:

$$x^{k+1} := \Pi_Q(x^k - \gamma_k \nabla f(x^k))$$

Proximal gradient method

Two terms: *Proximal* and *Gradient Descent*

1. *Gradient Descent*: say we want to solve:

$$\min_{x \in Q \subseteq \mathcal{X}} f(x)$$

   where Q is a simple convex subset of $\mathcal{X}$. Equivalently:

$$\min_{x \in \mathcal{X}} f(x) + \mathcal{I}_Q(x)$$

   Classically: perform a gradient descent step and a *Correction/Projection* step:

$$x^{k+1} := \Pi_Q(x^k - \gamma_k \nabla f(x^k))$$

2. *Proximal* as a Generalization of *Projection*: replace $\mathcal{I}_Q(x)$ by a general proper closed convex function, potentially non-smooth:

$$\min_{x \in \mathcal{X}} f(x) + \textcolor{red}{g(x)}$$

Proximal gradient method

Two terms: *Proximal* and *Gradient Descent*

1. *Gradient Descent*: say we want to solve:

$$\min_{x \in Q \subseteq \mathcal{X}} f(x)$$

   where Q is a simple convex subset of $\mathcal{X}$. Equivalently:

$$\min_{x \in \mathcal{X}} f(x) + \mathcal{I}_Q(x)$$

   Classically: perform a gradient descent step and a *Correction/Projection* step:

$$x^{k+1} := \Pi_Q(x^k - \gamma_k \nabla f(x^k))$$

2. *Proximal* as a Generalization of *Projection*: replace $\mathcal{I}_Q(x)$ by a general proper closed convex function, potentially non-smooth:

$$\min_{x \in \mathcal{X}} f(x) + g(x)$$

   Then: $x^{k+1} := \mathbf{prox}_{\gamma_k, g}(x^k - \gamma_k \nabla f(x^k))$

Proximal gradient method

Additional insights: the updates for Proximal gradient method

$$x^{k+1} := \mathbf{prox}_{\gamma_k, g}(x^k - \gamma_k \nabla f(x^k))$$

are equivalent to:

$$x^{k+1} := \arg\min_{x \in \mathcal{X}} \{g(x) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2\gamma_k} \|x - x_k\|_2^2\}$$

where $f(x)$ is replaced by a *model*, that is its first-order Taylor approximation built at the current iterate x^k augmented with a quadratic term.
Why is it equivalent ?

Proximal gradient method

General class of Problems to solve:

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

Proximal gradient method

General class of Problems to solve:

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

Assumptions:

- $f(.)$ is convex, differentiable and $\nabla f(.)$ is Lipschitz continuous with constant L_f
- $g(.)$ proper closed convex, possibly non-smooth; $\mathbf{prox}_{\lambda,g}$ is inexpensive.

Proximal gradient method

General class of Problems to solve:

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

Assumptions:

- $f(.)$ is convex, differentiable and $\nabla f(.)$ is Lipschitz continuous with constant L_f
- $g(.)$ proper closed convex, possibly non-smooth; $\mathbf{prox}_{\lambda,g}$ is inexpensive.

Proximal gradient algorithm:

$$x^{k+1} := \mathbf{prox}_{\gamma_k,g}(x^k - \gamma_k \nabla f(x^k))$$

Proximal gradient method

General class of Problems to solve:

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

Assumptions:

- $f(.)$ is convex, differentiable and $\nabla f(.)$ is Lipschitz continuous with constant L_f
- $g(.)$ proper closed convex, possibly non-smooth; $\mathbf{prox}_{\lambda,g}$ is inexpensive.

Proximal gradient algorithm:

$$x^{k+1} := \mathbf{prox}_{\gamma_k,g}(x^k - \gamma_k \nabla f(x^k))$$

Some convergence results:

- $O(1/k)$ convergence rate
- i.e. to get $(\Phi(x^k) - \Phi(x^\star)) \leq \epsilon$, need $O(1/\epsilon)$ iterations

Accelerated Proximal gradient method

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

[1] Nesterov (2004), Beck and Teboulle (2009)

Accelerated Proximal gradient method

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

- $f(.)$ is convex, differentiable and $\nabla f(.)$ is Lipschitz continuous with constant L_f
- $g(.)$ proper closed convex, possibly non-smooth; $\mathbf{prox}_{\gamma,g}$ is inexpensive.

[1] Nesterov (2004), Beck and Teboulle (2009)

Accelerated Proximal gradient method

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

- $f(.)$ is convex, differentiable and $\nabla f(.)$ is Lipschitz continuous with constant L_f
- $g(.)$ proper closed convex, possibly non-smooth; $\mathbf{prox}_{\gamma,g}$ is inexpensive.

Accelerated Proximal gradient algorithm [1]:

$$x^{k+1} := \mathbf{prox}_{\gamma_k,g}(y^k - \gamma_k \nabla f(y^k))$$

$$y^{k+1} := x^{k+1} + \beta_k(x^{k+1} - x^k)$$

[1] Nesterov (2004), Beck and Teboulle (2009)

Accelerated Proximal gradient method

$$\min_{x \in \mathcal{X}} \Phi(x) := f(x) + g(x)$$

- $f(.)$ is convex, differentiable and $\nabla f(.)$ is Lipschitz continuous with constant L_f
- $g(.)$ proper closed convex, possibly non-smooth; $\mathbf{prox}_{\gamma,g}$ is inexpensive.

Accelerated Proximal gradient algorithm [1]:

$$x^{k+1} := \mathbf{prox}_{\gamma_k,g}(y^k - \gamma_k \nabla f(y^k))$$

$$y^{k+1} := x^{k+1} + \beta_k (x^{k+1} - x^k)$$

- $O(1/k^2)$ convergence rate
- i.e. to get $(\Phi(x^k) - \Phi(x^\star)) \leq \epsilon$, need $O(1/\sqrt{\epsilon})$ iterations

[1] Nesterov (2004), Beck and Teboulle (2009)

Proximal gradient method - Numerical tests

Lasso Regression ▸ Introductory Video in ML course

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

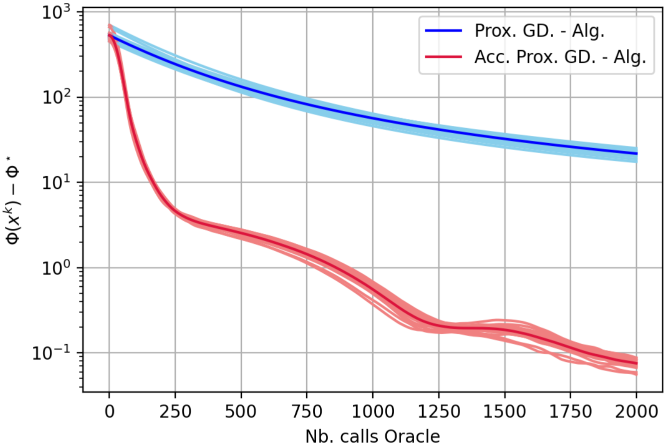
with:

- $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $\lambda > 0$.
- 20 instances created: $m = 300$ and $n = 500$, with A and b generated using random i.i.d. normal distribution.
- $\beta_k = \frac{k-1}{k+2}$ with k the iteration counter, and $\lambda = \frac{1}{\sqrt{m}}$.

Remark: Accelerated Proximal gradient algorithm for solving Lasso Regression has a name: FISTA, introduced by (Beck and Teboulle, 2013).

Proximal gradient method - Numerical tests

Lasso Regression - Results ‣ Colab File



Proximal gradient method - Numerical tests

Homework: Consider Lasso Logistic Regression

You have to solve:

$$\min_{W,b} \frac{1}{n} \sum_{i=1}^{n} \left(\log \left(\sum_{j=1}^{10} e^{[Wx_i+b]^{(j)}} \right) - \sum_{j=1}^{10} y_i^{(j)} \left[Wx_i + b \right]^{(j)} \right) + g(W)$$

where:

- $x_i \in \mathbb{R}^{784}$ is a vectorized gray image of a digit between 0 and 9 from (classes from 1 to 10) ▸ MNSIT database, and $y_i \in \{0,1\}^{10}$ is the binary vector corresponding to the class it belongs to, where $1 \leq i \leq 60000$.
- $g(W)$ is a regularization function, here $g(W) = \lambda \|W\|_1$.

Algorithms: implement both Proximal Gradient Descent and Accelerated Proximal Gradient Descent

Primal-Dual Methods

Primal Problem Formulation

We consider the following convex optimization problem:

$$\arg\min_{x \in \mathcal{X}} f(x) + g(Kx) \tag{2}$$

with **Assumptions A.**:

1. K is a nonzero linear operator: $K : \mathcal{X} \to \mathcal{U}$
2. real Hilbert spaces: $\mathcal{X}, \mathcal{U}$ (here, finite-dimension Hilbert spaces)
3. f can be convex L_f-smooth function (case 1) **or** proper closed convex (case 2)
4. $g : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$, proper closed convex function
5. *proximal operator* of g is tractable and efficiently computable.

Primal Problem Formulation

We consider the following convex optimization problem:

$$\arg\min_{x\in\mathcal{X}} f(x) + g(Kx) \tag{2}$$

with **Assumptions A.**:

1. K is a nonzero linear operator: $K : \mathcal{X} \to \mathcal{U}$
2. real Hilbert spaces: $\mathcal{X}, \mathcal{U}$ (here, finite-dimension Hilbert spaces)
3. f can be convex L_f-smooth function (case 1) **or** proper closed convex (case 2)
4. $g : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$, proper closed convex function
5. *proximal operator* of g is tractable and efficiently computable.

A little snag...: for general Problem (2) with $K \neq Id \to$ the proximal operator of $g \circ K$ is intractable in most cases !

Primal Problem Formulation

Examples:

- Generalized Lasso Problem:

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|Dx\|_1$$

- Image denoising:

$$\min_{x \in \mathcal{X}} \|x - y\|_2^2 + \lambda \|\nabla x\|_1$$

  Here: $g(x) = \lambda \sum_i |[\nabla x]_i|$, where $[\nabla x]_i$ is a two-dimensional intensity gradient vector at image pixel i.

- ℓ_1-norm SVM:

$$\min_{w,b} \sum_{i}^{n} \max\left(0, 1 - y_i\left(\langle w, x_i \rangle + b\right)\right) + \lambda \|w\|_1$$

- Linear Programming: $\min_x \langle c, x \rangle$ s.t. $Ax = b$, $x \geq 0$

Primal-Dual formulation

- **Primal**:
$$\min_{x \in \mathcal{X}} f(x) + g(Kx)$$
- Recall the convex conjugate: $g^*(u) := \max_{x \in \mathcal{X}} \{\langle u, Kx\rangle - g(Kx)\}$, hence: $g(Kx) \geq \max_{u \in \mathcal{U}} \langle u, Kx\rangle - g^*(u)$.
- Now we formulate the **Primal-Dual**:
$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} f(x) + \langle u, Kx\rangle - g^*(u)$$
- Using max–min inequality and the definition of the conjugate of $f(x)$, we derive the **Dual**:
$$\max_{u \in \mathcal{U}} -\left(f^*(-K^*u) + g^*(u)\right) \tag{3}$$
where $K^* : \mathcal{U} \to \mathcal{X}$ is the adjoint operator of K.
- Primal-dual gap: $f(x) + g(Kx) + g^*(u) + f^*(-K^*u) \to 0$ at optimality (for cvx. fun.)

Primal-Dual formulation

We focus today on the **Primal-Dual** formulation:

$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} f(x) + \langle u, Kx \rangle - g^*(u)$$

Why ?

1. As mentioned earlier: proximal operator for $g(Kx)$ is not trivial.
2. but we can get proximal operator g^* "more" easily...

Primal-Dual formulation

We focus today on the **Primal-Dual** formulation:

$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} f(x) + \langle u, Kx \rangle - g^*(u)$$

Why ?

1. As mentioned earlier: proximal operator for $g(Kx)$ is not trivial.
2. but we can get proximal operator g^* "more" easily...

A saddle point $(x^\star, u^\star) \in \mathcal{X} \times \mathcal{U}$ of the min-max function should satisfy the (first-order) optimality conditions (case 1):

$$\begin{cases} 0 \in \nabla f(x^\star) + K^* u^\star \\ 0 \in Kx^\star - \partial g^*(u^\star) \end{cases}$$

where $\partial(.)$ denotes the subdifferential. For case 2, replace $\nabla f(x^\star)$ by $\partial f(x^\star)$.
We iterate according to these conditions !

Primal-dual algorithm

- There exist several (deterministic) algorithms for solving Problem (2).
- Here we recall Chambolle and Pock (2011) (in case 2):
  1. Choose step sizes $\gamma, \tau > 0$, so that $\gamma\tau L^2 < 1$, with $L = \|K\|$ and $\beta \in [0,1]$.
  2. Choose initialization $(x^0, y^0) \in \mathcal{X} \times \mathcal{U}$.
  3. For each iteration:
     3.1 Proximal ascent step on the dual:
     $$u^{k+1} \leftarrow \mathbf{prox}_{\tau,g^*}(u^k + \tau K\hat{x}^k)$$
     3.2 Proximal descent step on the primal variable:
     $$x^{k+1} \leftarrow \mathbf{prox}_{\gamma,f}(x^k - \gamma K^* u^{k+1})$$
     3.3 Extrapolation step :
     $$\hat{x}^{k+1} \leftarrow x^{k+1} + \beta(x^{k+1} - x^k)$$

  $\rightarrow$ essentially alternately do proximal gradient descent and ascent for x and u.

Primal-dual algorithm

- There exist several (deterministic) algorithms for solving Problem (2).
- Here we recall Chambolle and Pock (2011) (in case 2):
  1. Choose step sizes $\gamma, \tau > 0$, so that $\gamma\tau L^2 < 1$, with $L = \|K\|$ and $\beta \in [0, 1]$.
  2. Choose initialization $(x^0, y^0) \in \mathcal{X} \times \mathcal{U}$.
  3. For each iteration:
     3.1 Proximal ascent step on the dual:
     $$u^{k+1} \leftarrow \mathbf{prox}_{\tau, g^*}(u^k + \tau K \hat{x}^k)$$
     3.2 Proximal descent step on the primal variable:
     $$x^{k+1} \leftarrow \mathbf{prox}_{\gamma, f}(x^k - \gamma K^* u^{k+1})$$
     3.3 Extrapolation step :
     $$\hat{x}^{k+1} \leftarrow x^{k+1} + \beta(x^{k+1} - x^k)$$

  $\rightarrow$ essentially alternately do proximal gradient descent and ascent for x and u.
- Note: Algo can rewritten with $\hat{u}^{k+1} \leftarrow u^{k+1} + \beta_k(u^{k+1} - u^k)$ instead of $\hat{x}^{k+1}$ and by exchanging the updates for u^{k+1} and x^{k+1}.

Primal-dual algorithm

- Here we slightly adapt Chambolle and Pock (2011) for case 1:
  1. Choose step sizes $\gamma, \tau > 0$, so that $\gamma\tau L^2 < 1$, with $L = \|K\|$ and $\beta \in [0,1]$.
  2. Choose initialization $(x^0, y^0) \in \mathcal{X} \times \mathcal{U}$.
  3. For each iteration:
     3.1 Proximal ascent step on the dual:
     $$u^{k+1} \leftarrow \mathbf{prox}_{\tau, g^*}(u^k + \tau K \hat{x}^k)$$
     3.2 Descent step on the primal variable:
     $$x^{k+1} \leftarrow x^k - \gamma \nabla f(x^k) - \gamma K^* u^{k+1}$$
     3.3 Extrapolation step :
     $$\hat{x}^{k+1} \leftarrow x^{k+1} + \beta(x^{k+1} - x^k)$$

  $\rightarrow$ essentially alternately do proximal gradient descent and ascent for x and u.
- Example: choose $\gamma \in (0, \frac{2}{L_f})$.

Discussion: Convergence

The algorithm's convergence rate depending on different types of the problem [2]

- Completely non-smooth problem: $O(1/k)$ for the duality gap
- The primal (f) or the dual (g^*) objective is uniformly convex[3]: $O(1/k^2)$ for $\|x^k - x^\star\|^2$
- Both f and g^* are uniformly convex: linear rate of convergence, that is $O(\rho^k)$ with $\rho < 1$ for $\|x^k - x^\star\|^2$

[2] see Chambolle and Pock (2011) for a detailed proof

[3] generalization of *strong*-convexity

Discussion: Update of γ, τ and β

- In the case one term (either f or g^*) is μ-strongly convex, Chambolle and Pock (2011) proposes the following variant of their algorithm:
  1. Choose step sizes $\gamma_0, \tau_0 > 0$, so that $\gamma_0 \tau_0 L^2 < 1$, with $L = \|K\|$ and $\beta \in [0, 1]$.
  2. Choose initialization $(x^0, y^0) \in \mathcal{X} \times \mathcal{U}$.
  3. For each iteration:
     3.1 Proximal ascent step on the dual:
     $$u^{k+1} \leftarrow \mathbf{prox}_{\tau_k, g^*}(u^k + \tau K \hat{x}^k)$$
     3.2 Descent step on the primal variable:
     $$x^{k+1} \leftarrow \mathbf{prox}_{\gamma, f}(x^k - \gamma K^* u^{k+1})$$
     3.3 Update of γ_k, τ_k and β_k:
     $$\beta_k \leftarrow \frac{1}{\sqrt{1 + 2\mu\gamma_k}}, \gamma_{k+1} \leftarrow \beta_k \gamma_k, \tau_{k+1} \leftarrow \frac{\tau_k}{\beta_k}$$
     3.4 Extrapolation step :
     $$\hat{x}^{k+1} \leftarrow x^{k+1} + \beta_k (x^{k+1} - x^k)$$

Discussion: Parallel implementation

$$\begin{cases} u^{k+1} \leftarrow \mathbf{prox}_{\tau, g^*}(u^k + \tau K \hat{x}^k) \quad \text{(dual proximal)} \\ x^{k+1} \leftarrow \mathbf{prox}_{\gamma, f}(x^k - \gamma K^* u^{k+1}) \quad \text{(primal proximal)} \\ \hat{x}^{k+1} \leftarrow x^{k+1} + \beta(x^{k+1} - x^k) \quad \text{(extrapolation)} \end{cases}$$

For Problems in computer vision:

- x and u are defined on a regular grid
- f and g are usually in a **separable sum** format.
- Small number of variables involved gradient part Kx
- perfect for GPU parallel computing!

Discussion: Arrow-Hurwicz method ($\beta = 0$)

$$\begin{cases} u^{k+1} \leftarrow \mathbf{prox}_{\tau, g^*}(u^k + \tau K x^k) \quad \text{(dual proximal)} \\ x^{k+1} \leftarrow \mathbf{prox}_{\gamma, f}(x^k - \gamma K^* u^{k+1}) \quad \text{(primal proximal)} \end{cases}$$

- Also tackles primal-dual method
- Without the ‘momentum’ step.
- Global $O(1/\sqrt{k})$ convergence of the gap, that is worst case rate of black box oriented subgradient methods.
- Potentially faster convergence rate, like $O(1/k)$ convergence guarantee (people haven’t proved it yet).
- In practice, for some problems it is still fast.

Discussion: ADMM

ADMM form (Primal):

$$\min_{x\in\mathcal{X},z\in\mathcal{U}} f(x) + g(z) \quad \text{s.t. } Kx - z = 0$$

Discussion: ADMM

ADMM form (Primal):

$$\min_{x \in \mathcal{X}, z \in \mathcal{U}} f(x) + g(z) \quad \text{s.t. } Kx - z = 0$$

Build Augmented Lagrangian:

$$L_\rho(x, z, u) := f(x) + g(z) + \langle u, Kx - z \rangle + \frac{\rho}{2} \|Kx - z\|_2^2$$

Discussion: ADMM

ADMM form (Primal):

$$\min_{x \in \mathcal{X}, z \in \mathcal{U}} f(x) + g(z) \quad \text{s.t. } Kx - z = 0$$

Build Augmented Lagrangian:

$$L_\rho(x, z, u) := f(x) + g(z) + \langle u, Kx - z \rangle + \frac{\rho}{2}\|Kx - z\|_2^2$$

ADMM Steps: given current iterates (x^k, z^k, u^k):

$$\begin{cases} z^{k+1} \leftarrow \arg\min_{z \in \mathcal{U}} \{g(z) - \langle u^k, z \rangle + \frac{\rho}{2}\|Kx^k - z\|_2^2\} & \text{(z-min., primal)} \\ x^{k+1} \leftarrow \arg\min_{x \in \mathcal{X}} \{f(x) + \langle u^k, Kx \rangle + \frac{\rho}{2}\|Kx - z^{k+1}\|_2^2\} & \text{(x-min., primal)} \\ u^{k+1} \leftarrow u^k + \rho(Kx^{k+1} - z^{k+1}) & \text{(dual update)} \end{cases}$$

Discussion: ADMM

ADMM form (Primal):

$$\min_{x\in\mathcal{X},z\in\mathcal{U}} f(x)+g(z) \quad \text{s.t. } Kx-z=0$$

Build Augmented Lagrangian:

$$L_\rho(x,z,u) := f(x)+g(z)+\langle u, Kx-z\rangle + \frac{\rho}{2}\|Kx-z\|_2^2$$

ADMM Steps: given current iterates (x^k, z^k, u^k):

$$\begin{cases} z^{k+1} \leftarrow \arg\min_{z\in\mathcal{U}}\{g(z)-\langle u^k, z\rangle + \frac{\rho}{2}\|Kx^k - z\|_2^2\} & \text{(z-min., primal)} \\ x^{k+1} \leftarrow \arg\min_{x\in\mathcal{X}}\{f(x)+\langle u^k, Kx\rangle + \frac{\rho}{2}\|Kx - z^{k+1}\|_2^2\} & \text{(x-min., primal)} \\ u^{k+1} \leftarrow u^k + \rho(Kx^{k+1} - z^{k+1}) & \text{(dual update)} \end{cases}$$

If $K = Id$, and $\gamma = \tau = \rho$, then Primal-dual method is very close to ADMM ! (some people claim that P-D is "faster", but we will try...)

Discussion: randomization

Quick remark about the power of randomness in classical finite sum setting:

$$f(x) = \sum_{i}^{n} f_i(x)$$

using only ∇f_i (every f_i is L-smooth and μ-strongly convex), lower bounds in (Woodworth and Srebro, 2016) to get ϵ-accuracy optimal solution:

1. deterministic algorithms: $O(n\sqrt{\frac{L}{\mu}}\log \epsilon^{-1})$
2. randomized algorithms: $O((n + \sqrt{\frac{nL}{\mu}}) \log \epsilon^{-1})$

Discussion: randomization

Quick remark about the power of randomness in classical finite sum setting:

$$f(x) = \sum_{i}^{n} f_i(x)$$

using only ∇f_i (every f_i is L-smooth and μ-strongly convex), lower bounds in (Woodworth and Srebro, 2016) to get ϵ-accuracy optimal solution:

1. deterministic algorithms: $O(n\sqrt{\frac{L}{\mu}}\log\epsilon^{-1})$
2. randomized algorithms: $O((n+\sqrt{\frac{nL}{\mu}})\log\epsilon^{-1})$

Potential directions for us

- randomize $\nabla f \rightarrow$ SGD-type algorithms.
- $\mathbf{prox}_{g^*}$ can be costly, randomize this step ?

Discussion: randomization

Quick remark about the power of randomness in classical finite sum setting:

$$f(x) = \sum_{i}^{n} f_i(x)$$

using only ∇f_i (every f_i is L-smooth and μ-strongly convex), lower bounds in (Woodworth and Srebro, 2016) to get ϵ-accuracy optimal solution:

1. deterministic algorithms: $O(n\sqrt{\frac{L}{\mu}} \log \epsilon^{-1})$
2. randomized algorithms: $O((n + \sqrt{\frac{nL}{\mu}}) \log \epsilon^{-1})$

Potential directions for us

- randomize $\nabla f \rightarrow$ SGD-type algorithms.
- $\mathbf{prox}_{g^*}$ can be costly, randomize this step ?
  $\rightarrow$ *Discussed in advanced part of these lectures dedicated to Primal-Dual Methods :)*

Showcases

Image denoising with Total Variation (TV)

We want to solve the following problem:

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$

Image denoising with Total Variation (TV)

We want to solve the following problem:

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|x - y\|_2^2 + \lambda \|\nabla x\|_1$$

Here:

- x is the denoised vectorized (or column stacked) image to compute, and y is the input noisy vectorized image.

Image denoising with Total Variation (TV)

We want to solve the following problem:

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|x - y\|_2^2 + \lambda\|\nabla x\|_1$$

Here:

- x is the denoised vectorized (or column stacked) image to compute, and y is the input noisy vectorized image.
- $g(x) = \lambda \sum_i |[\nabla x]_i|$, where $[\nabla x]_i$ is a two-dimensional (Isotropic) intensity gradient vector at image pixel i. ▸ Link.

Image denoising with Total Variation (TV)

We want to solve the following problem:

$$\min_{x \in \mathcal{X}} \frac{1}{2}\|x - y\|_2^2 + \lambda\|\nabla x\|_1$$

Here:

- x is the denoised vectorized (or column stacked) image to compute, and y is the input noisy vectorized image.
- $g(x) = \lambda \sum_i |[\nabla x]_i|$, where $[\nabla x]_i$ is a two-dimensional (Isotropic) intensity gradient vector at image pixel i. Link.
  Example: given an image $X \in \mathbb{R}^{2\times 2}$, this chosen TV operator is defined as:
  $TV(X) := \sum_{i=1}^{1}\sum_{j=1}^{2} |X_{i,j} - X_{i+1,j}| + \sum_{i=1}^{2}\sum_{j=1}^{1} |X_{i,j} - X_{i,j+1}|$
  Given $x = \vec{X}$, then by defining $\nabla = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$, we have $\|\nabla x\|_1 = TV(X)$

Image denoising with Total Variation (TV)

We solve the following equivalent problem:

$$\min_{x \in \mathcal{X}} \frac{\tilde{\lambda}}{2} \|x - y\|_2^2 + \|\nabla x\|_1$$

with $\lambda^{-1} = \tilde{\lambda}$.

[4] the dual of ℓ_1-norm is the ℓ_∞-norm

Image denoising with Total Variation (TV)

We solve the following equivalent problem:

$$\min_{x \in \mathcal{X}} \frac{\tilde{\lambda}}{2} \|x - y\|_2^2 + \|\nabla x\|_1$$

with $\lambda^{-1} = \tilde{\lambda}$.

- $f(x) := \frac{\tilde{\lambda}}{2}\|x - y\|_2^2$ is $\tilde{\lambda}$-strongly convex.

[4] the dual of ℓ_1-norm is the ℓ_∞-norm

Image denoising with Total Variation (TV)

We solve the following equivalent problem:

$$\min_{x \in \mathcal{X}} \frac{\tilde{\lambda}}{2} \|x - y\|_2^2 + \|\nabla x\|_1$$

with $\lambda^{-1} = \tilde{\lambda}$.

- $f(x) := \frac{\tilde{\lambda}}{2}\|x - y\|_2^2$ is $\tilde{\lambda}$-strongly convex.
- Expression for $g^*(u)$:
  1. given $g(y) = \|y\|_1 := \sup_{\|p\|_\infty \leq 1} \langle p, y \rangle$, with $\|p\|_\infty = \max_i |p_i|$ [4]
  2. $$g^*(u) = \sup_{y \in \mathcal{X}} \langle u, y \rangle - \|y\|_1$$
     $$= \sup_{y \in \mathcal{X}} \langle u, y \rangle - \sup_{\|p\|_\infty \leq 1} \langle p, y \rangle \underset{\text{Sion's theo.}}{=} \inf_{\|p\|_\infty \leq 1} \sup_{y \in \mathcal{X}} \langle y, u - p \rangle$$
     $$= \inf_{\|p\|_\infty \leq 1} \{0 \text{ if } u = p, \infty \text{ otherwise}\} = \{0 \text{ if } \|u\|_\infty \leq 1, \infty \text{ otherwise}\}$$

$\rightarrow g^*(u)$ is the indicator function for the unit ball w.r.t. ℓ_∞-norm, denoted $\mathcal{B}_\infty(0, 1)$

[4] the dual of ℓ_1-norm is the ℓ_∞-norm

Image denoising with Total Variation (TV)

- Dual Update: proximal operator for convex-set indicator function is just euclidean projecting onto the feasible closed set.
  Denoting $v := u^k + \tau_k K \hat{x}^k$, we have:

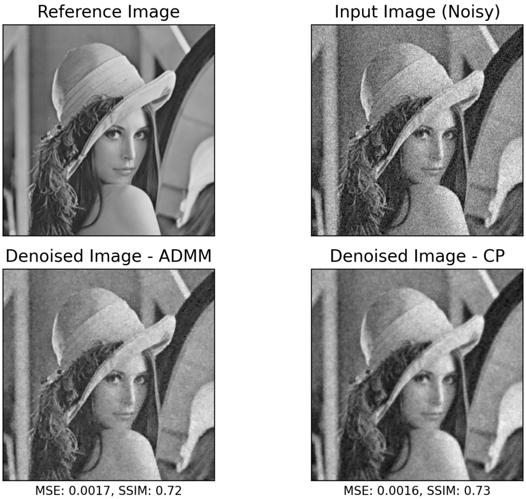
$$u^{k+1} := \Pi_{\mathcal{B}_\infty(0,1)}(v) = \frac{v}{\max(\|v\|_\infty, 1)}$$

- Primal Update: denoting $v := x^k - \gamma_k K^* u^{k+1}$, we have:

$$\begin{aligned} x^{k+1} &:= \mathbf{prox}_{\gamma_k, f}(v) \\ &:= \arg\min_{x \in \mathcal{X}} \{ \frac{\tilde{\lambda}}{2} \|x - y\|_2^2 + \frac{1}{2\gamma_k} \|x - v\|_2^2 \} \\ &:= \frac{v + \gamma_k \tilde{\lambda} y}{1 + \gamma_k \tilde{\lambda}} \end{aligned}$$

Image denoising with Total Variation (TV)

Image denoising - Results ▸ Colab File



Linear Programming

We are interested to solve:

$$\min_x \langle c, x \rangle \text{ s.t. } Ax = b, x \geq 0$$

[5] Proposition 4.4.2 from Bertsekas: Lagrangian function is such that: $L(., u)$ is convex, $L(x, .)$ is linear and $L(x, y) \to \infty$ if $\|x\| \to \infty$

Linear Programming

We are interested to solve:

$$\min_x \langle c, x\rangle \text{ s.t. } Ax = b, x \geq 0$$

Introducing u the Lagrangian multipliers associated to the equality constraints and using strong duality [5], we can write:

$$\min_x \max_u \langle c, x\rangle + \langle Ax - b, u\rangle, \quad x \geq 0$$

[5] Proposition 4.4.2 from Bertsekas: Lagrangian function is such that: $L(., u)$ is convex, $L(x, .)$ is linear and $L(x, y) \to \infty$ if $\|x\| \to \infty$

Linear Programming

We are interested to solve:

$$\min_x \langle c, x\rangle \text{ s.t. } Ax = b, x \geq 0$$

Introducing u the Lagrangian multipliers associated to the equality constraints and using strong duality [5], we can write:

$$\min_x \max_u \langle c, x\rangle + \langle Ax - b, u\rangle, \quad x \geq 0$$

Applying primal-dual algorithm:

- *Ascent step on the dual*: $u^{k+1} \leftarrow u^k + \tau(A\hat{x}^k - b)$
- *Descent step on the primal*: $x^{k+1} \leftarrow \max(0, x^k - \gamma(c + A^* u^{k+1}))$
- *Extrapolation*: $\hat{x}^{k+1} \leftarrow x^{k+1} + \beta(x^{k+1} - x^k)$

[5] Proposition 4.4.2 from Bertsekas: Lagrangian function is such that: $L(., u)$ is convex, $L(x, .)$ is linear and $L(x, y) \to \infty$ if $\|x\| \to \infty$

Conclusions

Summary

- First-order primal-dual algorithm for a class of structured convex optimization problems
- Objective function can be non-differentiable
- Easy to implement (we "just" need to derive the proximal operators)
- Optimal convergence rate on multiple sub-classes

Goodbye, So Soon

THANKS FOR THE ATTENTION

- v.leplat@skoltech.ru
- sites.google.com/view/valentinleplat/