Mathematical Optimization
IMA Summer Course: First-Order Algorithms

Shuzhong Zhang

Department of Industrial & Systems Engineering
University of Minnesota

zhangs@umn.edu

August 1, 2016

Convex Optimization

A very well developed tool in optimization: *convex optimization*.

Essentially, it deals with a model like

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X}, \end{array}$$

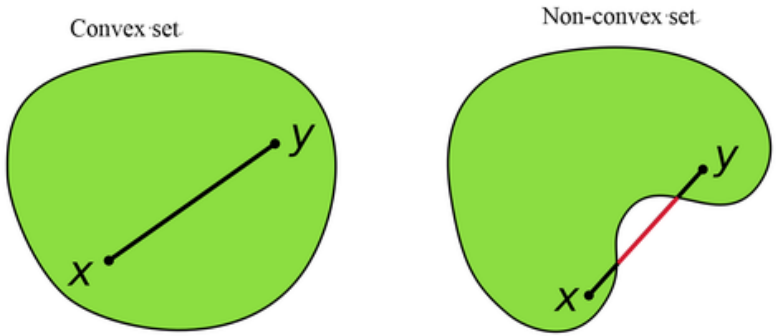
where $\mathcal{X}$ is a convex set:

$$\forall x, y \in \mathcal{X},\ \forall t \in [0,1] \implies tx + (1-t)y \in \mathcal{X},$$

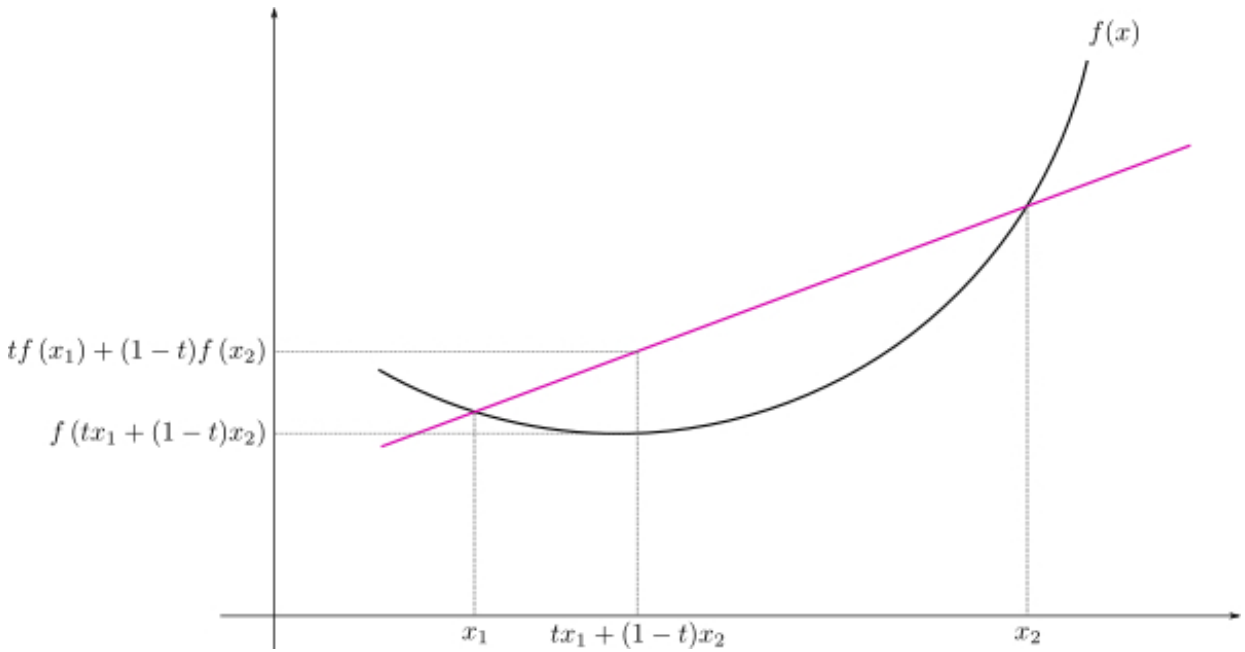
and f is a convex function:

$$\forall x, y \in \mathcal{X},\ \forall t \in [0,1] \implies f(tx + (1-t)y) \leqslant tf(x) + (1-t)f(y).$$

Convex set



Convex function



Least Squares

The famous least squares problem can be posed as:

$$\begin{array}{ll} \min & \|A^{\mathsf{T}}x - b\|_2^2 \\ \text{s.t.} & x \in \mathbf{R}^n. \end{array}$$

By setting the gradient of $\|A^{\mathsf{T}}x - b\|_2^2$ to zero, we have

$$\nabla \left(\|A^{\mathsf{T}}x - b\|_2^2 \right) = 2A(A^{\mathsf{T}}x - b) = 0.$$

If $\text{rank}(A) = n$ then the solution can be found explicitly

$$x^* = (AA^{\mathsf{T}})^{-1}Ab.$$

The Ridge Regression

In some cases, rank$(A) < n$, and the solution set may be unbounded.
In other cases, we may not want the solution to be "too large" anyway.

The concept of *least square with a regulatory term* becomes important.
The ridge regression is one such example:

$$\min \quad \|A^{\mathsf{T}}x - b\|_2^2 + \gamma\|x\|_2^2.$$

The solution is

$$x^* = (AA^{\mathsf{T}} + \gamma I)^{-1}Ab.$$

The model can also be equivalently posed as

$$\begin{array}{ll} \min & \|A^{\mathsf{T}}x - b\|_2 \\ \text{s.t.} & \|x\|_2 \leqslant \alpha \end{array}$$

Matlab and CVX

Michael Grant and Stephen Boyd's code CVX:

`http://cvxr.com/cvx/`

Example:

$$\begin{array}{ll} \min & \|Ax - b\|_2 \\ \text{s.t.} & Cx = d \\ & \|x\|_\infty \leqslant e. \end{array}$$

CVX syntax:

```
>> cvx_begin
>>       variable x(n)
>>       minimize( norm( A * x - b, 2 ) )
>>       subject to
>>         C * x == d
>>         norm( x, Inf ) <= e
>> cvx_end
```

What if the object is known to be 'sparse'?

In that case, the optimization model should be something like

$$\begin{array}{ll} \min & \|A^{\mathsf{T}}x - b\|_2^2 \\ \text{s.t.} & \|x\|_0 \leqslant k, \end{array}$$

or,

$$\min \quad \|A^{\mathsf{T}}x - b\|_2^2 + \gamma\|x\|_0.$$

The difficulty with this model is that $\|x\|_0$ is not really a norm: it is not even *convex*.

The LASSO Regression

The so-called LASSO (Least Absolute Shrinkage and Selection Operator) regulatory term – the L_1 norm – is its closest convex approximation

$$\min \quad \|A^{\mathsf{T}}x - b\|_2^2 + \gamma\|x\|_1.$$

The model can also be equivalently posed as

$$\begin{array}{ll} \min & \|A^{\mathsf{T}}x - b\|_2^2 \\ \text{s.t.} & \|x\|_1 \leqslant \alpha. \end{array}$$

Unlike the ridge regression case, no closed form solution exists for the LASSO model.

A possible CVX code for solving the LASSO problem

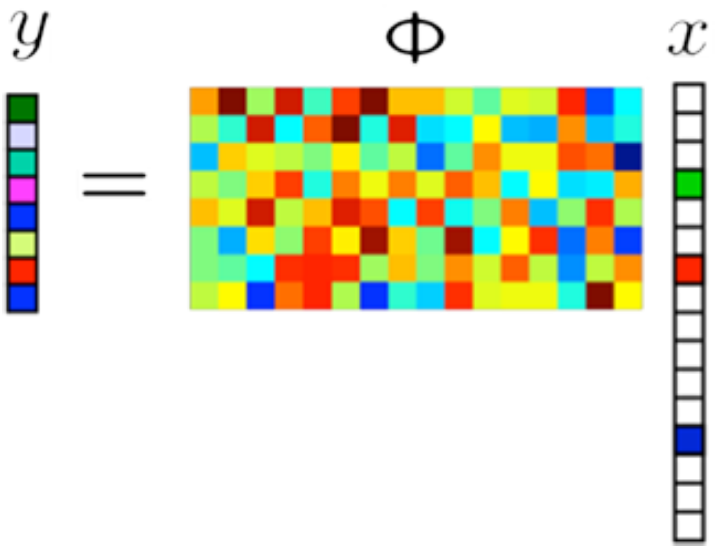
```
>> cvx_begin
>>       variable x(n)
>>       minimize( norm( A' * x - b, 2 ) )
>>       subject to
>>         norm( x, 1 ) <= alpha
>> cvx_end
```

Compressive Sensing

In image processing, often a picture is expressed by its wavelet coefficients

$$y = \sum_{i=1}^{n} x_i \Phi_i.$$

Though y is not necessarily a sparse vector, but its wavelet coefficients x_i's may be very sparse. The situation is depicted as follows:



The 'Cameraman'

Sparse approximation of a natural image:

The left picture is the original image;

The right picture is an approximation of image obtained by keeping only the largest 10% of the wavelet coefficients.

What if our sensing is *incomplete*? That is, we only have a small part of y, and the rest of the vector is simply missing.

So the problem really is to extract a solution x from an under-determined system $A^{\mathsf{T}}x = b$.

Though there are infinitely many solutions, we DO know that our true solution is very compact, or *sparse*. Therefore, the problem can be formulated as

$$\begin{array}{ll} \min & \|x\|_0 \\ \text{s.t.} & A^{\mathsf{T}}x = b. \end{array}$$

Convexification:

$$\begin{array}{ll} \min & \|x\|_1 \\ \text{s.t.} & A^{\mathsf{T}}x = b. \end{array}$$

Unlike the Least Squares fitting case, $A^{\mathsf{T}}x = b$ is usually under-determined in this context.

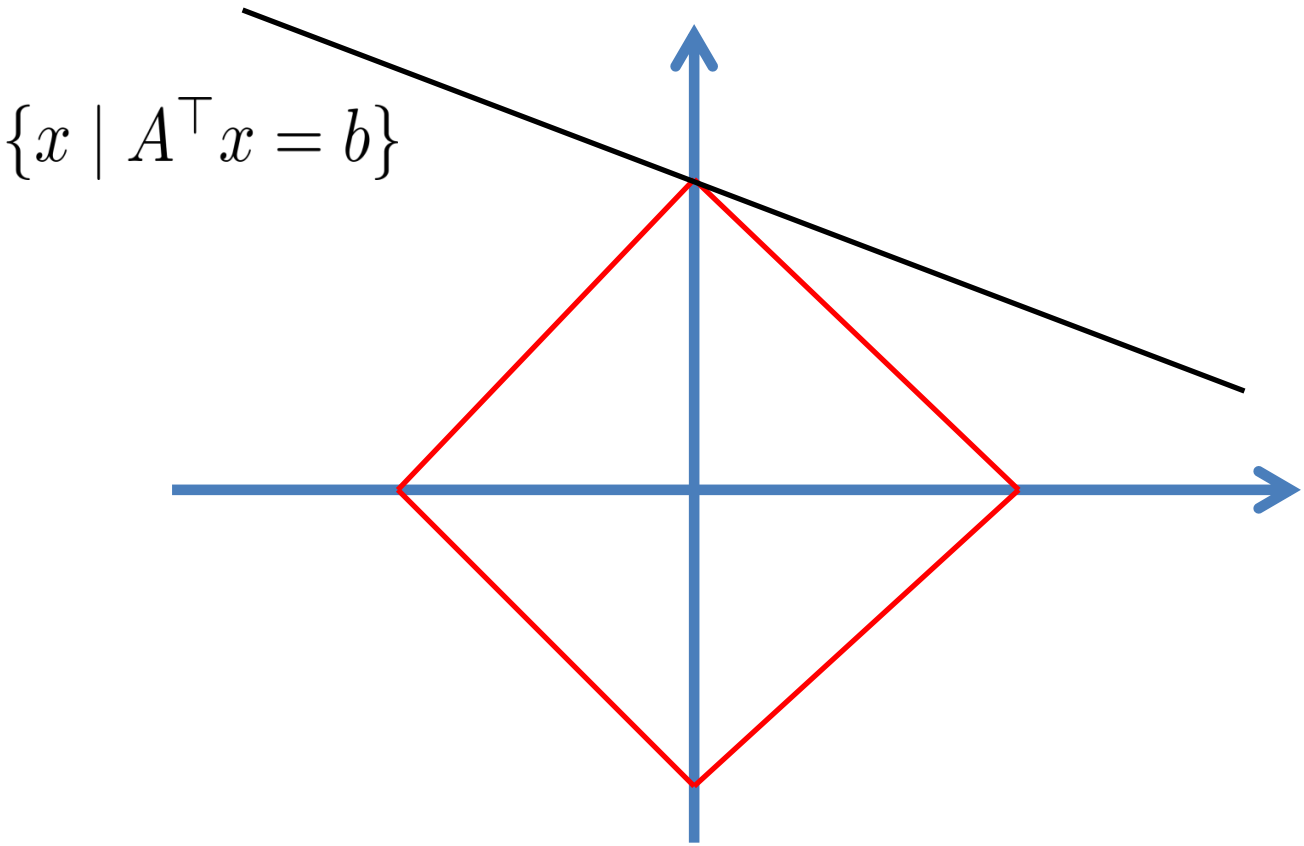
The above approach is known as *sparse optimization*, sometimes also called compressive sensing, or *the L_1-regularization method*.

A penalized version is

$$\min \quad \|x\|_1 + \gamma \|A^{\mathsf{T}}x - b\|_2^2,$$

which is essentially a LASSO model again.

Why LASSO regression makes sense



Convex Optimization Models

Unconstrained optimization

$$(M_1) \quad \min \quad f(x)$$

where f is a convex function.

Example:

$$\min \quad \|A^{\mathsf{T}}x - b\|_2^2 + \gamma\|x\|_2^2.$$

Constrained optimization

$$(M_2) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$

where $\mathcal{X} \subseteq \mathbf{R}^n$ is a convex set.

Example:

$$\begin{array}{ll} \min & \|A^{\mathsf{T}} x - b\|_2^2 \\ \text{s.t.} & \|x\|_2 \leqslant \alpha. \end{array}$$

Regulated convex optimization

$$(M_3) \quad \begin{array}{ll} \min & f(x) + h(x) \\ \text{s.t.} & x \in \mathcal{X} \end{array}$$

where f, h are both convex functions.

Example:

$$\begin{array}{ll} \min & \|A^{\mathsf{T}} x - b\|_2^2 + \gamma \|x\|_1 \\ \text{s.t.} & \|x\|_2 \leqslant \alpha. \end{array}$$

Structured convex optimization with linear constraints

$$(M_4) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X} \\ & Ax = b \end{array}$$

where $\mathcal{X} \subseteq \mathbf{R}^n$ is a convex set.

Example:

$$\begin{array}{ll} \min & \|x\|_1 \\ \text{s.t.} & \|x\|_2 \leqslant \alpha \\ & A^\mathsf{T} x = b. \end{array}$$

Separable structured convex optimization with coupling linear constraints

$$
\begin{array}{lll}
(M_5) & \min & f(x)+h(y) \\
 & \text{s.t.} & x\in\mathcal{X},\ y\in\mathcal{Y} \\
 & & Ax+By=b
\end{array}
$$

where $\mathcal{X}\subseteq\mathbf{R}^n$, $\mathcal{Y}\subseteq\mathbf{R}^m$ are convex sets, and f and h are convex functions.

Example:

$$
\begin{array}{ll}
\min & \|A^{\mathsf{T}}x-b\|_2^2+\gamma\|y\|_1 \\
\text{s.t.} & \|x\|_2\leqslant\alpha \\
 & x-y=0.
\end{array}
$$

Convergence Analysis

An iterative algorithm for solving an optimization model, producing iterates $\{x^0, x^1, ...\}$.

Possible error measurements:

- $e(x^k) = \|x^k - x^*\|$;
- $e(x^k) = f(x^k) - f(x^*)$.

Convergence: $\lim_{k\to\infty} e(x^k) = 0$.

Rate of Convergence:

- *Linear convergence*: there is $0 < a < 1$ such that $e(x^k) \leqslant Ca^k$;
- *Sub-linear convergence*: $e(x^k) \leqslant C/k^p$, where $p > 0$.

Basic Ideas Behind Many Optimization Algorithms

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\top (x - x^k)$$

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k) \quad + \tfrac{1}{2}(x - x^k)^\mathsf{T}\nabla^2 f(x^k)(x - x^k)$$

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k) \quad + \tfrac{1}{2}L\|x - x^k\|^2$$

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$$

$$\min f(x) \leftrightsquigarrow \min f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$$

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$$

$$\min f(x) \rightsquigarrow \min f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k) \text{ s.t. } \|x - x^k\| \leqslant \delta$$

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$

min $f(x) \rightsquigarrow$ min $f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$ s.t. $\|x - x^k\| \leqslant \delta$

$x^{k+1} = x^k - t\nabla f(x^k) \Longrightarrow$ Gradient (Sub-gradient) method

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$

min $f(x) \rightsquigarrow$ min $f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$ s.t. $\|x - x^k\| \leqslant \delta$

$x^{k+1} = x^k - t\nabla f(x^k) \Longrightarrow$ Gradient (Sub-gradient) method

min $f(x) \rightsquigarrow$

$$\begin{array}{ll} \min & f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k) + \frac{1}{2}(x - x^k)^\mathsf{T}\nabla^2 f(x^k)(x - x^k) \\ \text{s.t.} & \left|\frac{1}{2}(x - x^k)^\mathsf{T}\nabla^2 f(x^k)(x - x^k)\right| \leqslant \delta \end{array}$$

Basic Ideas Behind Many Optimization Algorithms

Approximation: replace unmanageable with manageable
Adaptation: apply approximation with caution!

$f(x) \rightsquigarrow f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$

min $f(x) \leftrightsquigarrow$ min $f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$ s.t. $\|x - x^k\| \leqslant \delta$

$x^{k+1} = x^k - t\nabla f(x^k) \Longrightarrow$ Gradient (Sub-gradient) method

min $f(x) \leftrightsquigarrow$

$$\begin{array}{ll} \min & f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k) + \frac{1}{2}(x - x^k)^\mathsf{T}\nabla^2 f(x^k)(x - x^k) \\ \text{s.t.} & \left|\frac{1}{2}(x - x^k)^\mathsf{T}\nabla^2 f(x^k)(x - x^k)\right| \leqslant \delta \end{array}$$

$x^{k+1} = x^k - t(\nabla^2 f(x^k))^{-1}\nabla f(x^k) \Longrightarrow$ Newton type method

(Sub)-Gradient Algorithm with Line-Search

Consider (M_1).

Let $\{t_k > 0 \mid k = 0, 1, 2, ...\}$ be a sequence of step-sizes.

The (Sub)-Gradient Algorithm

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

Take $d^k \in \partial f(x^k)$;

Let $x^{k+1} := x^k - t_k d^k$.

end for

(Sub)-Gradient Algorithm with Line-Search

Consider (M_1).

Let $\{t_k > 0 \mid k = 0, 1, 2, ...\}$ be a sequence of step-sizes.

The (Sub)-Gradient Algorithm

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

Take $d^k \in \partial f(x^k)$;

Let $x^{k+1} := x^k - t_k d^k$.

end for

How to choose step-sizes t_k's?

Convergence of the Gradient Method

Theorem 1

If there are $0 < m \leqslant M$ *such that* $0 \prec mI \preceq \nabla^2 f(x) \preceq MI$ *(i.e.,* f *is strongly convex), and line-search is performed*

$$t_k := \arg\min_t f\left(x^k - t\nabla f(x^k)\right),$$

then

$$f(x^{k+1}) - f(x^*) \leqslant \left(1 - \frac{m}{M}\right)(f(x^k) - f(x^*)).$$

Convergence of the Gradient Method

Theorem 1

If there are $0 < m \leqslant M$ *such that* $0 < mI \preceq \nabla^2 f(x) \preceq MI$ *(i.e.,* f *is strongly convex), and line-search is performed*

$$t_k := \arg\min_t f\left(x^k - t\nabla f(x^k)\right),$$

then

$$f(x^{k+1}) - f(x^*) \leqslant \left(1 - \frac{m}{M}\right)(f(x^k) - f(x^*)).$$

What if there is no strong convexity?

Convergence of the Gradient Method

Theorem 1

If there are $0 < m \leqslant M$ *such that* $0 < mI \preceq \nabla^2 f(x) \preceq MI$ *(i.e.,* f *is strongly convex), and line-search is performed*

$$t_k := \arg\min_t f\left(x^k - t\nabla f(x^k)\right),$$

then

$$f(x^{k+1}) - f(x^*) \leqslant \left(1 - \frac{m}{M}\right)(f(x^k) - f(x^*)).$$

What if there is no strong convexity?

Then there is no linear convergence!

Convergence of the Gradient Method

Theorem 1

If there are $0 < m \leqslant M$ *such that* $0 < mI \preceq \nabla^2 f(x) \preceq MI$ *(i.e.,* f *is strongly convex), and line-search is performed*

$$t_k := \arg\min_t f\left(x^k - t\nabla f(x^k)\right),$$

then

$$f(x^{k+1}) - f(x^*) \leqslant \left(1 - \frac{m}{M}\right)(f(x^k) - f(x^*)).$$

What if there is no strong convexity?

Then there is no linear convergence!

But there is still sub-linear convergence.

Worst case bounds for (sub)-gradient type algorithms

Suppose we generate a sequence of iterates $\{x^k \mid k = 0, 1, 2, ...\}$, in such a way that x^k is in the affine space spanned by $x^0, \nabla f(x^0), ..., \nabla f(x^{k-1})$.

Suppose f is Lipschitz continuous and no other information is known. Then, one can construct an example such that

$$\min_{x \in \mathcal{L}\{x^0, \nabla f(x^0), ..., \nabla f(x^{k-1})\}} f(x) - f(x^*) \geqslant O(1/\sqrt{k}), \ \forall k = 1, 2, ..., \lfloor n/2 \rfloor.$$

If additionally, we know f is differentiable and ∇f is Lipschitz continuous, then one can construct an example such that

$$\min_{x \in \mathcal{L}\{x^0, \nabla f(x^0), ..., \nabla f(x^{k-1})\}} f(x) - f(x^*) \geqslant O(1/k^2), \ \forall k = 1, 2, ..., \lfloor n/2 \rfloor.$$

Conditional Gradient Method

Consider (M_2).

Let $\{0 < t_k \leqslant 1 \mid k = 0, 1, 2, ...\}$ be a sequence of step-sizes.

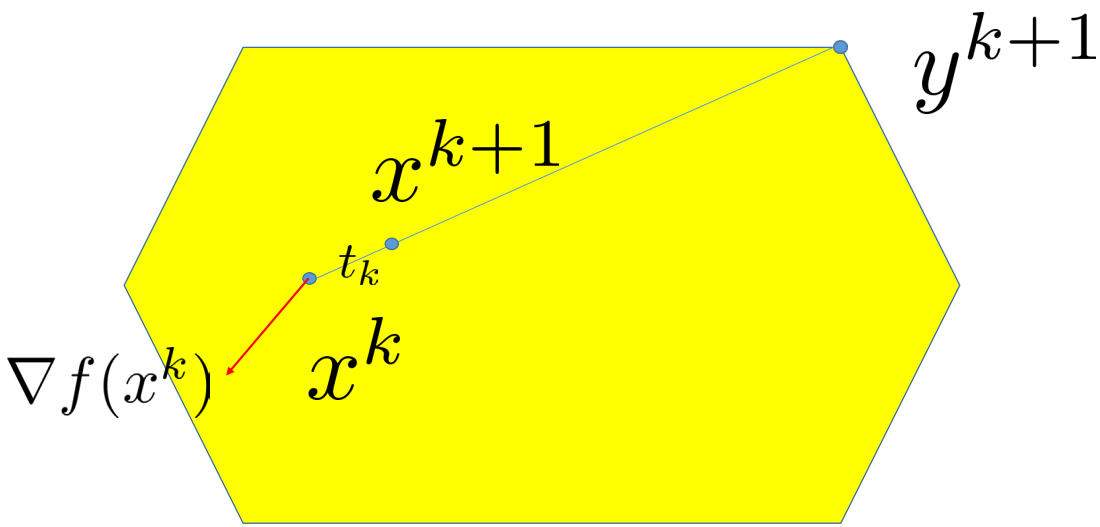
The Conditional Gradient Algorithm

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

$$y^{k+1} := \arg\min_{y \in \mathcal{X}} (\nabla f(x^k))^\top (y - x^k);$$
$$x^{k+1} := x^k + t_k(y^{k+1} - x^k).$$

end for



Theorem 2

Suppose that ∇f is Lipschitz continuous with constant L, and $\mathcal{X}$ is contained in a D-ball. Suppose the constants are chosen such that $f(x^2) - f(x^*) \leqslant LD^2$. If we choose $t_k = 2/k$ for $k \geqslant 2$, then $f(x^k) - f(x^*) \leqslant 2LD^2/k$ for all $k \geqslant 2$.

Proximal Point Algorithm

Let $f_k(x)$ be a certain convex approximation of $f(x)$.

Let $\{t_k > 0 \mid k = 0, 1, ...\}$ be a sequence of parameters.

The Proximal Point Algorithm

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

$$x^{k+1} := \arg\min_{x \in \mathcal{X}} f_k(x) + \frac{1}{2t_k}\|x - x^k\|^2.$$

end for

Theorem 3

Suppose that ∇f is Lipschitz continuous with constant L, and $t_k = \frac{1}{L}$ for all k and $f_k(x) := f(x^k) + \nabla f(x^k)^\mathsf{T}(x - x^k)$. Then, the proximal point algorithm has the convergence rate:

$$f(x^k) - f(x^*) \leqslant \frac{L\|x^0 - x^*\|^2}{2k}.$$

Gradient Projection

Consider (M_2).

Let $\{t_k > 0 \mid k = 0, 1, 2, ...\}$ be a sequence of step-sizes.

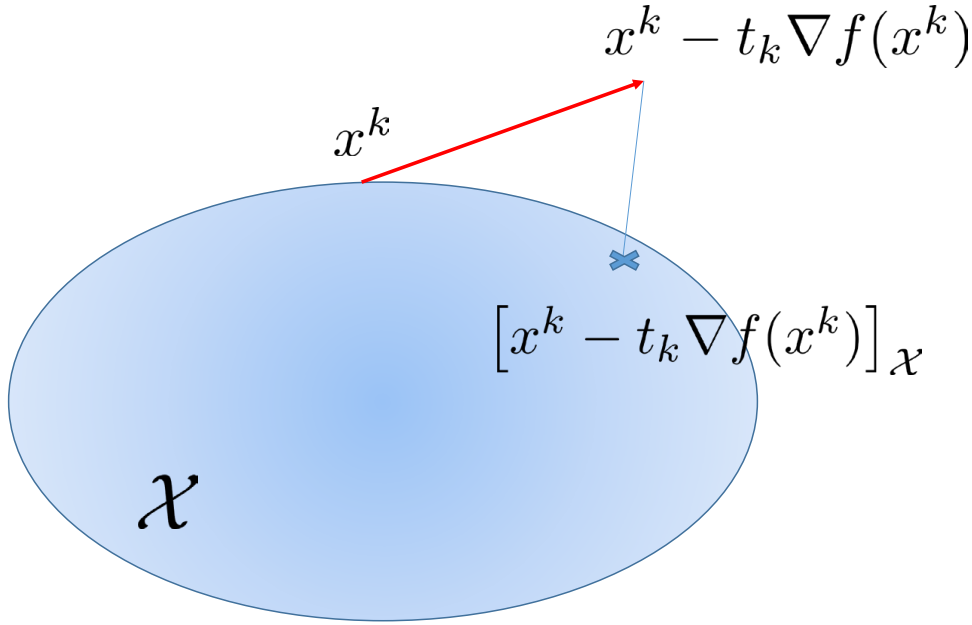
The Gradient Projection Algorithm

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

$$x^{k+1} := [x^k - t_k \nabla f(x^k)]_{\mathcal{X}}.$$

end for



The Proximal Point Interpretation

Observe that

$$[x^\ell - t_\ell d^\ell]_{\mathcal{X}} = \arg\min_{x\in\mathcal{X}} \ (d^\ell)^\mathsf{T}(x - x^\ell) + \frac{1}{2t_\ell}\|x - x^\ell\|^2.$$

This is true because

$$(d^\ell)^\mathsf{T}(x - x^\ell) + \frac{1}{2t_\ell}\|x - x^\ell\|^2 = \frac{1}{2t_\ell}\|x - (x^\ell - t_\ell d^\ell)\|^2 - \frac{t_\ell}{2}\|d^\ell\|^2.$$

Theorem 4

Suppose that ∇f is Lipschitz continuous with constant L, and $t_k = \frac{1}{L}$ for all k. Then, the gradient projection algorithm has the convergence rate:

$$f(x^k) - f(x^*) \leqslant \frac{L\|x^0 - x^*\|^2}{2k}.$$

Iterative Shrinkage-Thresholding Algorithm

Next we consider (M_3).

We introduce an algorithm known as *Iterative Shrinkage-Thresholding Algorithm* (ISTA). The method is also known as the *Proximal Gradient* algorithm. It is applicable when f is smooth and convex, and h is a structured simple convex function but may not be differentiable (e.g. $h(x) = \|x\|_1$).

ISTA

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

$$x^{k+1} := \arg\min_{x \in \mathcal{X}} \nabla f(x^k)^{\mathsf{T}}(x - x^k) + \frac{L}{2}\|x - x^k\|^2 + h(x).$$

end for

Theorem 5

Suppose that ∇f is Lipschitz continuous with constant L. Then, the following convergence rate holds for the ISTA

$$[f(x^k) + h(x^k)] - [f(x^*) + h(x^*)] \leqslant \frac{L\|x^0 - x^*\|^2}{2k}.$$

The Shrinkage Operator

The term ISTA is relevant when we apply it to solve the LASSO problem

$$\min \quad \frac{1}{2}\|Ax-b\|^2+\gamma\|x\|_1.$$

In that case, the key step is to implement

$$x^{k+1} := \arg\min_x \left\{ \frac{1}{2t_k}\|x-v^k\|^2+\gamma\|x\|_1 \right\}$$

which can be explicitly given by the so-called *shrinkage* operator

$$\arg\min_x \left\{ \frac{1}{2t_k}\|x-v^k\|^2+\gamma\|x\|_1 \right\} =: \mathcal{T}_{\gamma t_k}(v^k)$$

with

$$\mathcal{T}_\alpha(x)_i := (|x_i|-\alpha)_+ \cdot \mathrm{sign}(x_i), \ \ i=1,2,...,n.$$

The Mirror Descent Algorithm

The choice of Euclidean distance in the proximal point algorithm may appear to be quite arbitrary.

Let $\Phi(x)$ be a smooth and strongly convex function defined in the whole of $\mathbf{R}^n$. The Bregman distance is

$$B(y,x) := \Phi(y) - \Phi(x) - \nabla\Phi(x)^{\mathsf{T}}(y-x).$$

Consider (M_2).

Mirror Descent

Initialize $x^0 \in \mathcal{X}$

for $k = 0, 1, \cdots,$ **do**

Take $d^k \in \partial f(x^k)$;

$x^{k+1} := \arg\min_{x \in \mathcal{X}} (d^k)^\mathsf{T}(x - x^k) + \frac{1}{t_k} B(x, x^k)$.

end for

Theorem 6

If f is Lipschitz continuous, then $\min_{1 \leqslant \ell \leqslant k} \left(f(x^\ell) - f(x^*)\right) \leqslant \frac{B(x^*, x^0) + \frac{L^2}{4\sigma}}{\sqrt{k}}$.

Theorem 7

If ∇f is Lipschitz continuous, then $\min_{1 \leqslant \ell \leqslant k} \left(f(x^\ell) - f(x^*)\right) \leqslant \frac{B(x^*, x^0) \frac{M}{\sigma}}{k}$.

Nesterov's Acceleration

Nesterov's Acceleration

If the objective is only known to be convex and Lipschitz continuous, then we can achieve a convergence rate of $O(1/\sqrt{k})$ after k iterations. This bound is tight.

Nesterov's Acceleration

If the objective is only known to be convex and Lipschitz continuous, then we can achieve a convergence rate of $O(1/\sqrt{k})$ after k iterations. This bound is tight.

If the objective function is convex and differentiable, and moreover its gradient is Lipschitz continuous, then we have achieved the convergence rate of $O(1/k)$.

Nesterov's Acceleration

If the objective is only known to be convex and Lipschitz continuous, then we can achieve a convergence rate of $O(1/\sqrt{k})$ after k iterations. This bound is tight.

If the objective function is convex and differentiable, and moreover its gradient is Lipschitz continuous, then we have achieved the convergence rate of $O(1/k)$.

We also have examples to show that under the scheme $x^{\ell+1} \in \mathcal{L}(x^0, \nabla f(x^1), ..., \nabla f(x^\ell))$ for all iterations, then the convergence rate cannot be faster than $O(1/k^2)$.

Nesterov's Acceleration

If the objective is only known to be convex and Lipschitz continuous, then we can achieve a convergence rate of $O(1/\sqrt{k})$ after k iterations. This bound is tight.

If the objective function is convex and differentiable, and moreover its gradient is Lipschitz continuous, then we have achieved the convergence rate of $O(1/k)$.

We also have examples to show that under the scheme $x^{\ell+1} \in \mathcal{L}(x^0, \nabla f(x^1), ..., \nabla f(x^\ell))$ for all iterations, then the convergence rate cannot be faster than $O(1/k^2)$.

This raises a question: *Is the $O(1/k^2)$ rate achievable?*

Let us consider the gradient projection algorithm (or equivalently, the proximal point algorithm) for (M_1).

Nesterov (1983) proposed to modify the proximal point algorithm slightly: the point being linearized is not x^k but a combination of x^k and x^{k-1}.

Nesterov's Accelerated Algorithm

Initialize $x^0 = y^1 \in \mathcal{X}$

for $k = 1, 2, \cdots,$ **do**

$$x^k := \arg\min_{x \in \mathcal{X}} f(y^k) + \nabla f(y^k)^\mathsf{T}(x - y^k) + \frac{L}{2}\|x - y^k\|^2;$$

$$y^{k+1} := x^k + \frac{t_k - 1}{t_{k+1}}\left(x^k - x^{k-1}\right).$$

end for

The sequence $\{t_k \mid k = 1, 2, ...\}$ is generated recursively:

$t_1 = 1$, $t_{k+1}^2 - t_{k+1} = t_k$, $k = 1, 2, ...,$. More explicitly, $t_{k+1} = \frac{1 + \sqrt{4t_k^2 + 1}}{2}$ for $k = 1, 2, ...$ One easily verifies that $t_k \geqslant \frac{k+1}{2}$ for $k = 1, 2, ...$

Theorem 8

Nesterov's algorithm has a convergence rate of

$$\begin{aligned} f(x^k) - f(x^*) &\leqslant \frac{f(x^1) - f(x^*) + \frac{L\|x^1 - x^*\|^2}{2}}{t_k^2} \\ &\leqslant \frac{4(f(x^1) - f(x^*)) + 2L\|x^1 - x^*\|^2}{(k+1)^2}. \end{aligned}$$

FISTA

Nesterov's algorithm is *optimal* since it achieves the best possible rate, in the order of magnitude: $O(1/k^2)$. How about (M_2)?

Recall that we discussed the so-called ISTA for (M_2), which is also known as proximal gradient method. Can we improve the rate of convergence? The answer is: *yes*. The result is the so-called FISTA ('Fast ISTA').

FISTA

Initialize $x^0 = y^1 \in \mathcal{X}$

for $k = 1, 2, \cdots,$ **do**

$$x^k := \arg\min_{x \in \mathcal{X}} f(y^k) + \nabla f(y^k)^\mathsf{T}(x - y^k) + \frac{L}{2}\|x - y^k\|^2 + h(x);$$

$$y^{k+1} := x^k + \frac{t_k - 1}{t_{k+1}}\left(x^k - x^{k-1}\right).$$

end for

The sequence $\{t_k \mid k = 1, 2, ...\}$ is the same as in Nesterov's scheme.

Theorem 9

The FISTA has a convergence rate of

$$\begin{aligned} & [f(x^k) + h(x^k)] - [f(x^*) + h(x^*)] \\ \leqslant \quad & \frac{4[(f(x^1) + h(x^1)) - (f(x^*) + h(x^*))] + 2L\|x^1 - x^*\|^2}{(k+1)^2}. \end{aligned}$$

Method of Multipliers

Let us now consider (M_4):

$$\begin{array}{lll} (M_4) & \min & f(x) \\ & \text{s.t.} & x \in \mathcal{X} \\ & & Ax = b \end{array}$$

Let λ be the Lagrangian multiplier associated with the constraint $Ax = b$.

The augmented Lagrangian function is

$$\mathcal{L}_\gamma(x;\lambda) := f(x) - \lambda^\mathsf{T}(Ax - b) + \frac{\gamma}{2}\|Ax - b\|^2.$$

Method of Multipliers

Initialize $\lambda^0 \in \mathbf{R}^m$

for $k = 0, 1, \cdots,$ **do**

$$x^{k+1} := \arg\min_{x \in \mathcal{X}} \mathcal{L}_\gamma(x; \lambda^k);$$
$$\lambda^{k+1} := \lambda^k - \gamma(Ax^{k+1} - b).$$

end for

Theorem 10

Let $\bar{x}^k = \frac{1}{k}\sum_{\ell=1}^k x^\ell$. *Then, for any fixed* $\rho > 0$ *we have*

$$f(\bar{x}^k) - f(x^*) + \rho\|A\bar{x}^k - b\| \leqslant \frac{\rho^2/\gamma + \|\lambda^0\|^2/\gamma}{k}.$$

The ADMM

Let us now introduce a very popular method for solving

$$(M_5) \quad \begin{array}{ll} \min & f(x) + h(y) \\ \text{s.t.} & x \in \mathcal{X},\ y \in \mathcal{Y} \\ & Ax + By = b. \end{array}$$

The method is known as *Alternating Direction Method of Multipliers*.
The augmented Lagrangian function

$$\mathcal{L}_\gamma(x, y; \lambda) := f(x) + h(y) - \lambda^\mathsf{T}(Ax + By - b) + \frac{\gamma}{2}\|Ax + By - b\|^2.$$

ADMM

Initialize $\lambda^0 \in \mathbf{R}^m$, $x^0 \in \mathcal{X}$, $y^0 \in \mathcal{Y}$.

for $k = 0, 1, \cdots,$ **do**

$$\begin{aligned} x^{k+1} &:= \arg\min_{x \in \mathcal{X}} \mathcal{L}_\gamma(x, y^k, \lambda^k) \\ y^{k+1} &:= \arg\min_{y \in \mathcal{Y}} \mathcal{L}_\gamma(x^{k+1}, y, \lambda^k) \\ \lambda^{k+1} &:= \lambda^k - \gamma(Ax^{k+1} + By^{k+1} - b). \end{aligned}$$

end for

The analysis for the ADMM was only completed recently. It is more involved than that for the method of multipliers. We shall skip the details and only present the main result below.

Let

$$\bar{x}^k = \frac{1}{k}\sum_{\ell=1}^{k} x^\ell \text{ and } \bar{y}^k = \frac{1}{k}\sum_{\ell=1}^{k} y^\ell.$$

Theorem 11

Consider the ADMM. For any fixed $\lambda \in \mathbf{R}^m$, the following estimation holds

$$\begin{aligned} & [f(\bar{x}^k) + h(\bar{y}^k)] - [f(x^*) + h(x^*)] - \lambda^{\mathsf{T}} \left(A\bar{x}^k + B\bar{y}^k - b\right) \\ \leqslant & \frac{\|\lambda - \lambda^0\|^2/\gamma + \|Ax^* + By^0 - b\|^2\gamma}{2k}. \end{aligned}$$

References

- Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- Dimitri Bertsekas, *Convex Optimization Algorithms*, Athena Scientific, 2015.
- Sébastien Bubeck, *The Complexities of Optimization*, Lecture Notes, 2013.
- Lieven Vandenberghe's lecture notes: `http://www.seas.ucla.edu/~vandenbe/ee236c.html`