10-725/36-725: Convex Optimization **Fall 2015**

Lecture 5: September 12

Lecturer: Lecturer: Ryan Tibshirani *Scribes: Scribes: Barun Patra and Tyler Vuong*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

5.1 Canonical Convex Programs Revisited

5.1.1 Conic Programs

Conic Programs are defined as

$$\begin{aligned} &\min_x c^T x \\ \text{subject to } &Ax = b \\ &D(x) + d \in \mathcal{K} \end{aligned} \tag{5.1}$$

Where

- $c, x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times x}$, $b \in \mathbb{R}^m$
- $D : \mathbb{R}^n \leftarrow Y$ is a linear map, $d \in Y$, for Euclidean space Y
- $K \subset Y$ is a closed convex cone.

5.1.2 Second Order Cone program

Second Order Cone program (SOCP) is defined as:

$$\begin{aligned} &\min_x c^T x \\ \text{subject to } &||D_i x + d_i||_2 \leq e_i^T x + f_i, i = 1 \ldots p \\ &Ax = b \end{aligned} \tag{5.2}$$

Theorem 5.1 *SOCP's are Conic Programs*

Proof: Consider the second order cone:

$$Q = \{(x, t) : ||x||_2 \leq t\} \tag{5.3}$$

Consider the constraint

$$\begin{aligned} &||D_1 x + d_1||_2 \leq e_1^T x + f_1 \\ \implies &(D_1 x + d_1, e_1^T x + f_1) \in Q_1 \end{aligned} \tag{5.4}$$

Similarly, $(D_2x + d_2, e_2^Tx + f_2) \in Q_2$. To generalize, we have

$$||D_ix + d_i||_2 \leq e_i^Tx + f_i \iff (D_ix + d_i, e_i^Tx + f_i) \in Q_i, \forall i \in \{1, 2, \dots p\} \tag{5.5}$$

Now, considering the cone taking the cartesian product of all the previous cones gives us the conic program, i.e $K = Q_i \times \dots Q_p$ ■

Theorem 5.2 *Every SOCP is an Semi Definite Program (SDP)*

Proof: First note that:

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \iff A - BC^{-1}B^T \succeq 0 \tag{5.6}$$

For symmetric A, C and $C \succeq 0$

Now we prove the following

$$||x||_2 \leq t \iff \begin{bmatrix} tI & x \\ x^T & t \end{bmatrix} \succeq 0 \tag{5.7}$$

Proof:

$$\begin{aligned} \begin{bmatrix} tI & x \\ x^T & t \end{bmatrix} \succeq 0 &\iff t^2I - xx^T \succeq 0 \\ \iff a^T \left[t^2I - xx^T\right] a \geq 0 \ \forall a > 0 \\ \text{In particular, for a = x, we have} \\ &\iff t^2||x||^2 - ||x||^4 \geq 0 \\ &\iff ||x||^2 \leq t^2 \\ &\iff ||x|| \leq t \end{aligned} \tag{5.8}$$

■ ■

Finally, we have the following:

Theorem 5.3 *Every QP is a SOCP*

Proof: Consider the QP

$$\begin{aligned} &\min_{x,t} c^x + t \\ &\text{subject to} Dx \leq d \\ &\frac{1}{2}x^TQx \leq t \\ &Ax = b \end{aligned} \tag{5.9}$$

In particular, consider the constraint $\frac{1}{2}x^TQx \leq t$. We claim the following

$$\frac{1}{2}x^TQx \leq t \iff ||(\frac{1}{\sqrt{2}}Q^{\frac{1}{2}}x, \frac{1}{2}(1-t))||_2 \leq \frac{1}{2}(1+t) \tag{5.10}$$

Proof:

$$\begin{aligned} ||(\frac{1}{\sqrt{2}}Q^{\frac{1}{2}}x, \frac{1}{2}(1-t))||_2 &\leq \frac{1}{2}(1+t) \\ ||(\frac{1}{\sqrt{2}}Q^{\frac{1}{2}}x, \frac{1}{2}(1-t))||_2^2 &\leq \frac{1}{4}(1+t)^2 \\ \frac{1}{2}x^TQx + \frac{1}{4}(1-t)^2 &\leq \frac{1}{4}(1+t)^2 \\ \frac{1}{2}x^TQx &\leq t \end{aligned} \tag{5.11}$$

■ Thus, this establishes the fact that QPs $\subseteq$ SOCPs ■

Thus, we end up with the following hierarchy:

$$\text{LPs} \subseteq \text{QPs} \subseteq \text{SOCPs} \subseteq \text{SDPs} \subseteq \text{Conic programs} \tag{5.12}$$

1. LPs and QPs are "easy" algorithmically
2. SDPs in generally are "hard" (don't generalize to a large number of variables. Only very special SDPs are easy (like graphical lasso, tree norm imputation).
3. Conic problems are usually "hard".

5.2 Gradient Descent

Consider the unconstrained, smooth convex optimization:

$$\min_x f(x) \tag{5.13}$$

- f is convex and differentiable
- $\text{dom}(f) = \mathbb{R}^n$
- Optimal criterion value $f^* = \min_x f(x)$ and solution at x^*

Define Gradient Descent algorithm as:

Choose initial point $x^0 \in \mathbb{R}^n$
Repeat:

$$x^k = x^{k-1} - t_k \nabla f(x^{k-1}), \text{k = 1,2,3} \tag{5.14}$$

5.2.1 Interpretation

The Taylor expansion of f gives us:

$$f(y) \approx f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)\nabla^2 f(x)(y-x) \tag{5.15}$$

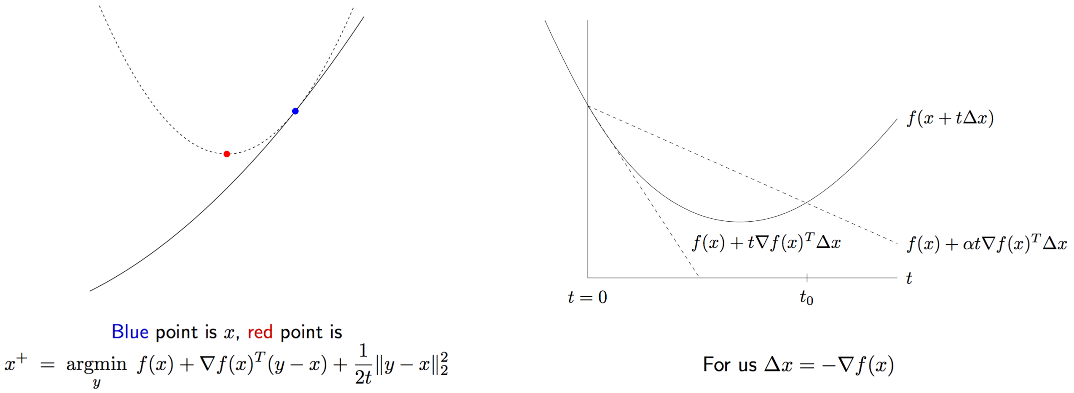
Consider the Quadratic approximation of f, replacing $\nabla^2 f(x)$ by $\frac{1}{t}I$, we have

$$f(y) \approx f(x) + \nabla f(x)^T (y-x) + \frac{1}{2t}||y-x||_2^2 \tag{5.16}$$

Now, minimizing wrt y, we get:

$$\begin{aligned} &\frac{\partial f(y)}{\partial y} \approx \nabla f(x) + \frac{1}{2}(y-x) = 0 \\ &\implies y = x - t\nabla f(x) \end{aligned} \tag{5.17}$$

Which gives us the gradient descent update rule. Figure 5.1a shows pictorially the interpretation. The dotted function shows the quadratic approximation, and the red dot shows the minima of the quadratic approximation.


Blue point is x, red point is
$x^+ = \underset{y}{\text{argmin}}\ f(x) + \nabla f(x)^T(y-x) + \frac{1}{2t}\|y-x\|_2^2$
$f(x+t\Delta x)$
$f(x)+t\nabla f(x)^T\Delta x$
$f(x)+\alpha t\nabla f(x)^T\Delta x$
t
$t=0$
t_0
For us $\Delta x = -\nabla f(x)$

(a) Gradient Descent

(b) Backtracking Line Search

5.2.2 Backtracking Line Search

If the step size is too large, gradient descent can diverge. Analogously, if the step-size is too small, gradient descent can converge too slowly.

One way to avoid this is to adaptively choose the step size using backtracking line search:

1. Fix parameters $0 < \beta < 1$ and $0 < \alpha \leq \frac{1}{2}$
2. $t = t_{\text{init}}$.
   While $f(x - t\nabla f(x)) > f(x) - \alpha t||f(x)||_2^2$
   - $t := \beta t$
3. $x^+ := x - t\nabla f(x)$

5.2.3 Backtracking Intuition

Consider the Figure 5.1b. In the figure, fixing x, $\delta x = -\nabla f(x)$ and α, as a function of t, we can see that by shrinking t, we move from right to left on the graph; as long as the line is under the function, and stop when it exceeds it.

5.2.4 Direct minimization For Line Search

One could also think of doing an exact line search:

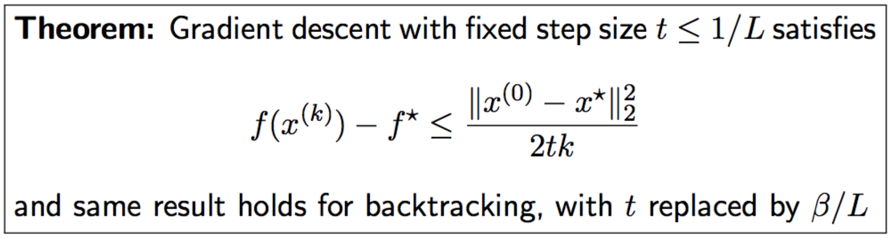
$$t = \text{argmin}_{s\geq 0} f(x - s\nabla f(x)) \tag{5.18}$$

But that is usually **not** possible to do directly; and approximations to the same are not as efficient as general backtracking. So is **not** worth the effort.

5.3 Convergence Analysis

5.3.1 Gradient Descent Convergence

- Convergence result for GD used for bases in all future lectures for different algorithms comparisons
- Rate is how quickly an algorithm converges
- If f is convex and differentiable
- $\text{dom}(f) = \mathbb{R}^n$
- ∇f is Lipschitz continuous with constant $L > 0$
  - if f has two derivatives, same thing as saying the largest eigenvalue of the Hessian of f is at most L for all x - assumes twice differentiable
  - second derivative some notion of curviture and this is the bound on that curvature



Theorem: Gradient descent with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)}) - f^\star \leq \frac{\|x^{(0)} - x^\star\|_2^2}{2tk}$$

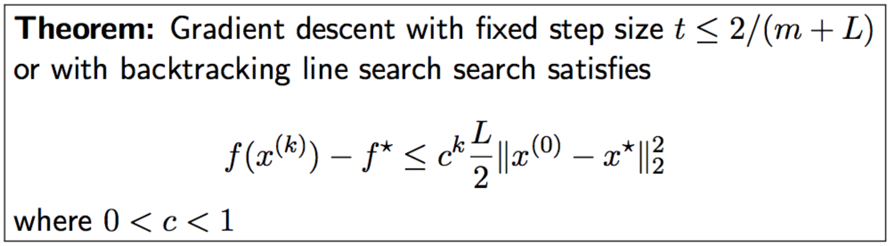
and same result holds for backtracking, with t replaced by β/L

(a) Gradient Descent Convergence

- criterian value at step k minus the optimal value
- t is the step size and is at most 1/L
- t equal to 1/L is biggest step sized allowed
- β is less than 1 and makes the bound a little worse when using backtracking
- proof is straight forward and uses properties about Lipschitz and can upperbound function by quadratic
- gradient descent has convergence rate $O(\frac{1}{k})$
- We read this by saying after k iterations, the gap between the criterion and where we are goes down by 1/k
- finds ϵ-suboptimal point in $O(\frac{1}{\epsilon})$ iterations
- gradient descent with fixed step size

5.3.2 Gradient Descent Convergence Strong Convexity

- what happens if you know more?
- strong convexity of f means f(x) - $\frac{m}{2}||x||_2^2$ is convex for some m > 0
- If we assume ∇f is Lipschitz continuous with constant $L > 0$
- can lower bound the function by a quadratic - Strong convexity
- Lipschitz and strong convexity together means you can sandwitch the function by two quadratics



Theorem: Gradient descent with fixed step size $t \leq 2/(m+L)$ or with backtracking line search search satisfies

$$f(x^{(k)}) - f^\star \leq c^k \frac{L}{2} \|x^{(0)} - x^\star\|_2^2$$

where $0 < c < 1$

(a) Gradient Descent Convergence under Strong Convexity

- goes to 0 a lot faster
- rate under strong convexity is $O(c^k)$
- Exponentially fast!
- finds ϵ-suboptimal point in $O(\log(\frac{1}{\epsilon}))$ iterations
- if you want converge to a small ϵ guarantee, this is a lot faster since log
- Called linear convergence since it looks linear on a semi-log plot
- contraction factor c in rate depends adversely on condition number L/m: higher condition number means slower rate
- contractor factor c approaches 1 as L/m gets larger
- poorly conditioned will make gradient descent do very poorly
- later will see how second order methods will help
- depends on ratio of largest eigenvalue and smallest eigenvalue of Hessian which is upperbounded by L/m under strong convexity
- affects not only our upper bound and is very apparent in practice too

5.3.3 Necessary Conditions

Conditions for a simple problem, $f(\beta) = \frac{1}{2}||y - X\beta||_2^2$
The gradient of that function is $X^T(X\beta - Y)$ and the Hessian is $X^T X$

- Lipschitz contiunity of ∇f means
  - $\nabla^2 f(x) \preceq LI$
  - As $\nabla^2 f(\beta) = X^T X, L = \sigma^2_{max}(X)$
  - if X is big, better idea to do backtracking
- Strong convexity of f means
  - $\nabla^2 f(x) \succeq mI$
  - As $\nabla^2 f(\beta) = X^T X, L = \sigma^2_{min}(X)$
  - If X is wide (more columns than rows), then $\sigma_{min}(X) = 0$ and f can't be strongly convex
  - Even if $\sigma_{min}(X) > 0$, can have a very large condition number $L/m = \sigma^2_{max}(X)/\sigma^2_{min}(X)$
  - condition number could be very large even if it's strongly convex
  - non convex proof is simplier than convex

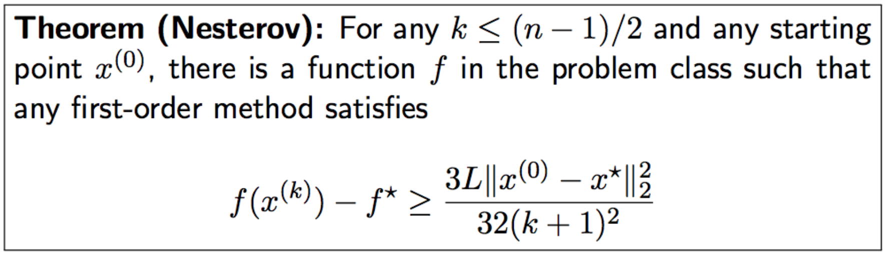
5.3.4 Practicalities

- stopping rule we use is when $||\nabla f(x)||_2$ is small
- gradient is small we stop
  - remember that $\nabla f(x^*) = 0$ at the solution x^*
- if f is strongly convex with parameter m, then $||\nabla f(x)||_2 \leq \sqrt{2m\epsilon} \rightarrow f(x) - f^* \leq \epsilon$
- this implies we are ϵ sub-optimal
- Pros and Cons of Gradient Descent
  - Pro: Simple idea and each iteration is usually cheap
  - Pro: fast for well-conditioned, strongly convex problems
  - Con: can often be slow since many problems aren't strongly convex or well conditioned
  - Con: can't handle nondifferentiable functions
  - everything we do depends on the gradient

5.3.5 Can we do better?

Gradient Descent has $O(1/\epsilon)$ convergence rate over problem class of convex, differentiable functions with Lipschitz gradients
Is gradient descent the best we can do if we only knew the gradient? We have first order methods which are itereative methods that update $x^{(k)}$

Can attain rate $O(1/k^2)$ or $O(1/\sqrt{\epsilon})$ There is method that can do faster using acceleration



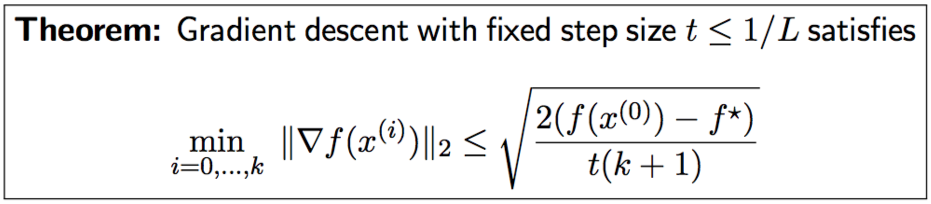
Theorem (Nesterov): For any $k \leq (n-1)/2$ and any starting point $x^{(0)}$, there is a function f in the problem class such that any first-order method satisfies

$$f(x^{(k)}) - f^\star \geq \frac{3L\|x^{(0)} - x^\star\|_2^2}{32(k+1)^2}$$

(a) Nesterov Convergence Theorem

5.3.6 What about nonconvex functions??

- Assume f is differentiable with Lipschitz gradient but now f is nonconvex
- Rather than optimality, find x such that $||\nabla f(x)||_2 \leq \epsilon$
- This is called $\epsilon-$stationary
- smallest gradient at step k is bounded by the below

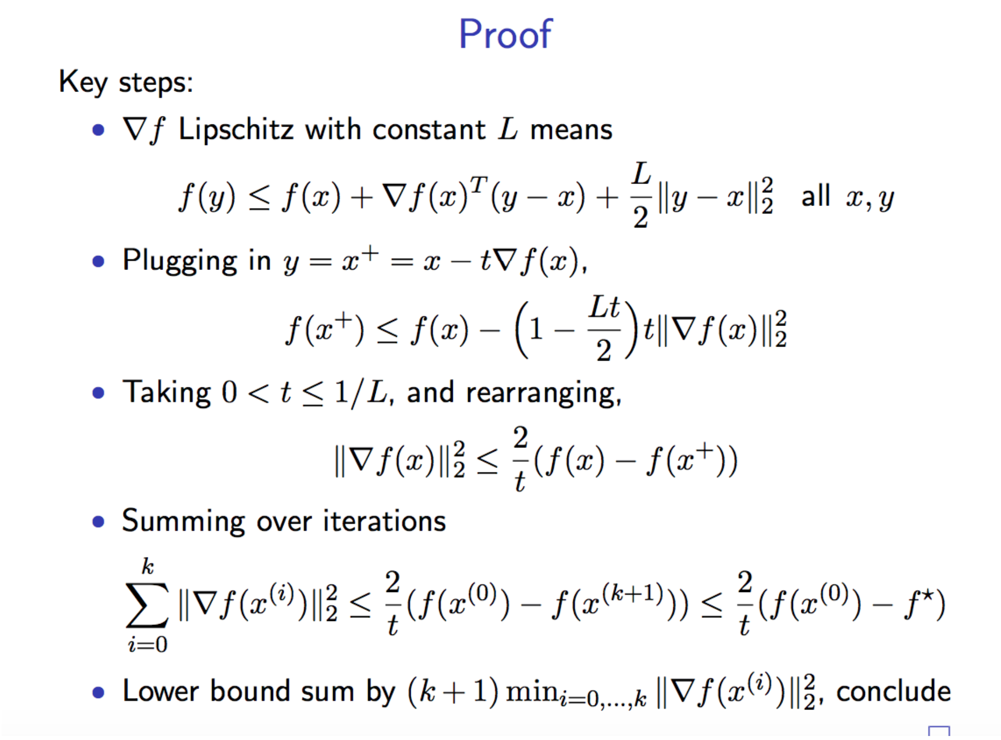


Theorem: Gradient descent with fixed step size $t \leq 1/L$ satisfies

$$\min_{i=0,\ldots,k} \|\nabla f(x^{(i)})\|_2 \leq \sqrt{\frac{2(f(x^{(0)}) - f^\star)}{t(k+1)}}$$

(a) Nonconvex Convergence

- Gradient descent has rate $O(1/\sqrt{k})$ or $O(1/\epsilon^2)$ even in the nonconvex case for finding stationary points
- Can't be improved (over class of differentiable functions with Lipschitz gradients) by any deterministic algorithm



Proof

Key steps:

- ∇f Lipschitz with constant L means

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|y-x\|_2^2 \quad \text{all } x, y$$

- Plugging in $y = x^+ = x - t\nabla f(x)$,

$$f(x^+) \leq f(x) - \Big(1 - \frac{Lt}{2}\Big)t\|\nabla f(x)\|_2^2$$

- Taking $0 < t \leq 1/L$, and rearranging,

$$\|\nabla f(x)\|_2^2 \leq \frac{2}{t}(f(x) - f(x^+))$$

- Summing over iterations

$$\sum_{i=0}^{k}\|\nabla f(x^{(i)})\|_2^2 \leq \frac{2}{t}(f(x^{(0)}) - f(x^{(k+1)})) \leq \frac{2}{t}(f(x^{(0)}) - f^\star)$$

- Lower bound sum by $(k+1)\min_{i=0,\ldots,k}\|\nabla f(x^{(i)})\|_2^2$, conclude

□

(a) Proof Nonconvex Convergence