University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

January 20, 2014 Notes: Ashia Wilson and Benjamin Recht

Lecture 1

1 What is optimization?

Maximize Goals/ Minimize Costs
s.t. Constraints

Example 1 *Some examples of goals/costs and constraints include*

- **Goals:** *Energy, Cost, Power, Data-fidelity, Log-likelihood*
- **Contraints:** *Natural law, Priors, Resource Budgets*

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in \Omega \end{aligned}$$

Note:

$$\max_{x\in\Omega} f(x) = -\min_{x\in\Omega} -f(x)$$

$$f = \begin{cases} \text{goodness of fit} & \frac{1}{n}\sum_{i=1}^n \ell(mx_i + b, y_i) \\ \text{utility} & \sum_{i=1}^n \mathcal{U}_i x_i \\ \text{power} & IV\cos\phi \end{cases}$$

$$\Omega = \begin{cases} \text{budget} & \sum_i x_i \leq B \\ \text{priors} & x_i \in \mathbb{N} \text{ integer constraints} \\ \text{location} & ||x - \hat{x}|| \leq D \text{ "don't place } x \text{ more than } D \text{ away from } \hat{x}\text{"} \end{cases}$$

Linear programming: The general formulation is a linear function we are trying to optimize and constraints defined by linear equalities and inequalities.

$$\begin{aligned} \min_x \quad & c^T x \\ s.t. \quad & Ax = b \\ & Dx \geq e \end{aligned} \tag{P}$$

Nonlinear programming: Every other problem that can not be reformulated as a linear program falls into this category. Obviously that's too general to be able to say anything reasonable about. This course aims to chart out the sorts of nonlinear problems that can be efficiently solved. We will pay close attention to problems that are scalable to high-dimensional decision problems and where the number of data points is large.

Things we need to be able to do to solve optimization problem:

1. Computing $f(x), \nabla f(x), \nabla^2 f(x)$. We will typically tag such computations as resources, and bookkeep the number of times we need to evaluate functions, Hessians, and gradients.

2. We also will need to be able to test membership in Ω.

1.1 Solutions of optimization problems

Definition 1

- *x^* is a solution of* (P) *if $x^* \in \Omega$ and $f(x^*) \leq f(x)\ \forall x \in \Omega$*
- *x^* is a local solution of* (P) *if there is a neighborhood $\mathcal{N}$ around x such that $x^* \in \Omega$, $f(x^*) \leq f(x)\ \ \forall x \in \mathcal{N} \cap \Omega$.*
  Fact*: it is NP hard to check local optimality, even if f is ∞-differetiable and $\Omega = \mathbb{R}^n$. We will prove this fact in a future lecture.*

For which functions can we check optimality? Convex ones

2 Convexity

Convex functions are functions for which the line between any two points lies above the graph. Formally

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)\ \ \forall x, y \in \Omega,\ t \in [0,1]$$

Proposition 1 *For convex functions, local minima/maxima are global minima/maxima.*

Proof If x_1, x_2 are local minima, $f(x_1) < f(x_2)$, and f is convex, then

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) < f(x_2) \quad \forall t \in [0,1]$$

Therefore x_1 is not a local minimum. ■

Example 2 *A simple example of a convex function is a quadratic: $f(x) = x^T Q x + b^T x + c$, Q has nonnegative eigenvalues.*

3 Topics

1. Accerlation (add memory to the process)
2. Higher order methods (Hessians) and quasi-Newton methods
3. Noise/randomization
4. Non-Smooth functions
5. Projected gradients
6. Constrained optimization (Lagrange Analysis)
7. Eigenvalue optimization and Semi-definite programming

4 Positive definite matrices

Positive definite matrices are central to convex optimization algorithms and their analysis. In this section, we quickly review some of the core properties that we will use throughout the semester.

Definition 2 (Positive definite) *A matrix A is positive definite (pd) if it is symmetric ($A = A^T$) and $x^T Ax > 0, \ \forall x \in \mathbb{R}^n \backslash \{0\}$. We denote this as $A \succ 0$*

Definition 3 (Positive semi-definite) *A matrix A is positive semidefinite (psd) if it is symmetric ($A = A^*$) and $x^T Ax \geq 0, \ \forall x \in \mathbb{R}^n$. We denote this as $A \succeq 0$*

Note that all pd matrices are psd, but not vice versa. Some of the main properties of positive semidefinite matrices include.

Properties 1

1. *If $A \succeq 0$, and $B \succeq 0$, then $A + B \succeq 0$.*
2. *$a \in \mathbb{R}$, $a \geq 0$ implies $aA \succeq 0$.*
3. *For any $F \in \mathbb{R}^{n \times n}, FF^T$ is psd. Conversely, if A is psd there exists an F such that $A = FF^T$.*

We leave the proofs of these properties as exercises. Note that (1) and (2) still hold if "psd" is replaced with "pd." That is, the sum of two pd matrices is pd. And multiplying a pd matrix by a positive scalar preserves positive definiteness.

Definition 4 (Eigenvalues) *λ is a eigenvalue of A if $Ax = \lambda x \ \ \forall x \in \mathbb{R}^n$*

Eigenvalues of psd matrices are all non-negative. Eigenvalues of pd matrices are all positive. This follows by multiplying the equation $Ax = \lambda x$ on the right by x^T.

The set of psd matrices admits a partial ordering, and we will make use of this fact frequently.

Definition 5 (P.S.D Ordering) *We write that $A \succ B$ iff $A - B \succ 0$.*

We will also need the following properties.

Properties 2

1. *For any $m \times n$ matrix F, if $A \succ B \Rightarrow FAF^T \succ FBF^T$*
2. *$A \preceq LI$ if and only if all eigenvalues are less than or equal to L*
3. *$A \succeq \ell I \iff$ all eigenvalues are greater than or equal to ℓ*

The last two properties are mostly useful as notational shorthands. Upper and lower bounding the eigenvalues of psd matrices will be critical in the complexity analysis of optimization algorithms.

5 Convex Sets

Most of this was covered in 227BT, but we'll focus in on a few facts that will be useful throughout the course.

Definition 6 (Convex set) *$\mathcal{C} \subseteq \mathbb{R}^n$ is convex if it contains all line segments between $\forall x \in \mathcal{C}$*

Definition 7 (Line Segment) *For $x, y \in \mathbb{R}^d$, the line segment between them is defined to be $\{tx + (1-t)y : t \in [0,1]\}$*

Definition 8 (Convex Combination) *For a set of points $x_1, \ldots, x_n$, a convex combination of them is defined as*

$$\{z = \sum_{i=1}^{n} p_i x_i : \sum_{i=1}^{n} p_i = 1, p_i \geq 0\}$$

One can readily check that $\mathcal{C}$ is convex if and only if $\mathcal{C}$ contains all convex combinations of its elements. Moreover, by a limiting argument, if p is any probability distribution on $\mathcal{C}$, then $\mathbb{E}_p[X] = \int_X xp(x)dx$ is contained in $\mathcal{C}$.

If we form the set of all convex combinations of some arbitrary set $\mathcal{A}$, then we get a convex set which best approximates $\mathcal{A}$.

Definition 9 (Convex hull) *The convex hull of a set $\mathcal{A} \subset \mathbb{R}^d$ is defined as*

$$\text{conv}(\mathcal{A}) = \left\{ \sum_{i=1}^{n} p_i x_i \ : \ \ p_i \geq 0, \sum_{i=1}^{n} p_i = 1, \ x_i \in \mathcal{A}, n \in \mathbb{N} \right\}$$

$\text{conv}(\mathcal{A})$ is the smallest convex set containing $\mathcal{A}$: if C is some set that contains $\mathcal{A}$, $\mathcal{C}$ contains all convex combinations of points in $\mathcal{A}$, and hence it contains $\text{conv}(\mathcal{A})$.

The following properties of convex hulls are straight forward and perhaps are ones we will use most frequently in this course:

Properties 3 *1. Arbitrary intersection of convex sets are convex:*

$$\mathcal{C}_1, \ \mathcal{C}_2 \ \ \textit{convex} \Rightarrow \mathcal{C}_1 \cap \mathcal{C}_2 \ \textit{is convex}$$

or more generally

$$\textit{If } \{\mathcal{C}_\alpha\}_{\alpha \in \mathcal{I}} \textit{ is a collection of convex sets, then } \bigcap_{\alpha \in \mathcal{I}} \mathcal{C}_\alpha \textit{ is convex}$$

2. *Convexity is invariant under affine transformation. If A is an $m \times d$ matrix and $b \in \mathbb{R}^m$, then $\{Ax + b : x \in \mathcal{C}\}$ is convex.*

3. *Covexity is also preserved under inverse affine transforms. If $\mathcal{C}$ is convex, then $\{x : Ax + b \in \mathbb{C}\}$ is also convex.*

We can build all convex sets from a few elementary building blocks. But note that this is not always (or even not typically) the most parsimonious representation of a given convex set. So note the following are convex:

1. Subspaces. Linearity is a more restrictive condition than convexity.

2. Affine spaces $\{x : Ax = b\}$. This can also be checked by writing out the definition. But also note that affine spaces are defined by the condition that $x, y \in \mathcal{S}$ implies $tx + (1-t)y \in \mathcal{S}$ for any $t \in \mathbb{R}$.

3. Hyperplanes $\{x : \langle a, x\rangle = b\}$. Hyperplanes are just special affine spaces (with co-dimension 1).

4. Half-spaces: $\{x : \langle a, x\rangle \leq b\}$. Half-spaces are perhaps the most important convex sets in all of optimization. Note that any closed convex set is equal to the intersection of all half-spaces containing it. This follows from the separating hyperplane theorem.

5. Cones. $\mathcal{C}$ is a cone if $\forall x, y \in \mathbb{C},\ \ t_1, t_2 \geq 0\ \ t_1 x + t_2 y \in \mathbb{C}$

6 Taylor's theorem

Let's now turn to functions. For $f : \mathbb{R}^d \to \mathbb{R}$, define

$$\text{Gradient} \quad \nabla f(x) = \left[\frac{\partial f}{\partial x_i}\right]_{i=1,\ldots d}$$

$$\text{Hessian} \quad \nabla^2 f(x) = \left[\frac{\partial^2 f}{\partial x_i x_j}\right]_{i,j=1,\ldots d}$$

The most important theorem in all optimization is very old, and you learned it in multivariable calculus:

Theorem 2 (Taylor's Theorem)

1. *If f is continuously differentiable, then*

$$f(x) = f(x_0) + \nabla f(tx + (1-t)x_0)^T (x - x_0) \quad \textit{for some } t \in [0, 1]$$

2. *If f is twice continuously differentiable, then*

$$\nabla f(x) = \nabla f(x_0) + \int_0^1 \nabla^2 f(tx + (1-t)x_0)(x - x_0) dt$$

and

$$f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(tx + (1-t)x_0)^T (x - x_0) \quad \textit{for some } t \in [0, 1]$$

We will use Taylor's theorem countless times in the course.

7 Extended Real-Valued Functions

We will primarily consider optimization of convex functions over convex sets. Thus, it will be important to review the following properties of convex sets.

First, we establish a bit of notation about extended real-valued functions. For convenience we will let functions achieve the value ∞ on certain points. These are essentially the places where f is not defined. This will be useful for simplifying the problem statements and proofs when we study constrained optimization.

First define the notion of the domain and extension of a function f:

Definition 10 *The domain of f is defined as*

$$dom(f) = \{x : f(x) < \infty\}$$

Definition 11 (Extended Real Valued Functions) *$f : D \to \mathbb{R}$ where $D = domf \subseteq \mathbb{R}^d$. Define the extension $\tilde{f}$ of f*

$$\tilde{f}(x) = \begin{cases} f(x) & x \in domf \\ \infty & x \notin domf \end{cases}$$

The addition of the value at ∞ lets us define functions on all of $\mathbb{R}^d$, thus treating constrained problems as unconstrained. The most important extended real-valued function is the indicator function:

Definition 12 *The indicator function of a set $\mathcal{C}$ as*

$$I_{\mathcal{C}}(x) = \begin{cases} 0 & x \in \mathcal{C} \\ \infty & x \notin \mathcal{C} \end{cases}$$

.

Note that in computer science, the indicator function is often defined to be 1 inside $\mathcal{C}$ and 0 outside. Our definition is the negative logarithm of the computer science convention.

8 Convex Functions

We now turn to the most definition of the course:

Definition 13 (Convex function) *$f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is convex if*

$$\forall x, y \in domf, \quad f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Note that in this definition we don't need to require that $\operatorname{dom} f$ is convex because of the definition of extended real valued functions. But you should always keep in mind that the set of points where a convex function is finite is a convex set.

Proposition 3 *If f is differentiable, then f is convex if and only if $\forall x, y, \quad f(x) \geq f(y) + \nabla f(y)^T(x-y)$ (i.e. the first order Taylor approximation is a global lower bound of f)*

Proof We just prove the one dimensional case. A full proof can be found in a variety of places, including Boyd and Vandenberghe. Note that by the definition of convexity

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Rearranging this inequality gives

$$f(y) + \frac{1}{t}\left(f(y + t(x-y)) - f(y)\right) \leq f(x)$$

Taking the limit $t \to 0$ gives our desired result.

The converse can also be found in Boyd and Vandenberghe. ■

The second condition of convex functions is a curvature condition. Smooth f are convex if and only if they have positive curvature everywhere.

Proposition 4 *If f is two times continuously differentiable on* $\mathrm{dom}(f)$, $\nabla^2 f(x) \succeq 0\ \forall x \in domf$ *if and only if f is convex*

Proof Our first order condition from Proposition 3 gives us that for all x and y

$$\begin{aligned} 0 &\leq (\nabla f(x) - \nabla f(y))^T (y - x) \\ &= (y-x)^T \left[\int_0^1 \nabla^2 f(ty + (1-t)xdt\right](y-x) \end{aligned}$$

where the final equality is Taylor's theorem. Letting $y = x + \alpha p$ for some arbitrary vector p and $\alpha > 0$, we have

$$p^T \left[\int_0^1 \nabla^2 f(x + t\alpha p)dt\right] p \geq 0$$

Letting α go to zero proves that the Hessian is psd.

Conversely, if the Hessian is psd, then Taylor's theorem implies that for some $t \in [0,1]$

$$\begin{aligned} f(x+p) &= f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 (x + tp) p \\ &\geq f(x) + \nabla f(x)^T p \end{aligned}$$

and hence f is convex by Proposition 3. ■

The most common convex functions are

1. Affine functions $f(x) = b^T x + c$.
2. Quadratics: $f(x) = \frac{1}{2} x^T A x + b^T x + c$ is convex if and only if $A \succeq 0$.

Note that we can take these primitives and make considerably more complex convex functions by pointwise maximization:

Proposition 5 *If f_α are convex for $\alpha \in \mathcal{I} \Rightarrow$ $\max_{\alpha \in \mathcal{I}} f_\alpha(x)$ is convex*

Proof This is a nearly trivial consequence of the definition of convexity. Let $g_{\mathcal{A}}(x) = \max_\alpha f_\alpha(x)$. Then for $t \in [0,1]$ and arbitrary x and y, we have

$$\begin{aligned} g_{\mathcal{A}}(tx + (1-t)y) &= \max_{\alpha \in \mathcal{A}} \; f_\alpha(tx + (1-t)y) \\ &\leq \max_{\alpha \in \mathcal{A}} \; tf_\alpha(x) + (1-t)f_\alpha(y) \\ &\leq t \max_{\alpha \in \mathcal{I}} \; f_\alpha(x) + (1-t) \max_{\beta \in \mathcal{I}} \; f_\beta(y) \\ &= tg_{\mathcal{A}}(x) + (1-t)g_{\mathcal{A}}(y) \end{aligned}$$

The first inequality is the definition of convexity. The second inequality follows because if you are allowed to choose α and β to be not equal to each other, then you can achieve a larger upper bound. ■

The following examples demonstrate the power of this proposition

1. The maximum eigenvalue is convex on function when defined on the set of symmetric matrices (we can define it to be ∞ if not symmetric). This follows by looking at the variational representation
$$\lambda_{\max}(A) = \max_{\substack{x \in \mathbb{R}^d \\ ||x||=1}} x^T A x$$
2. For any norm x, $||x|| = \max_{\substack{z \in \mathbb{R}^d \\ ||z||_* \leq 1}} \langle x, z \rangle$.

Finally, we define the epigraph of a function. The epigraph transforms statements about convex functions back into assertions about convex sets.

Definition 14 (Epigraph) *We define the epigraph of a function f to be*

$$\text{epi}(f) = \{(x,t) \in \mathbb{R}^{d+1} : t \geq f(x)\}$$

Proposition 6 *f is convex if and only if* $\text{epi}(f)$ *is convex*

Example: A is a positive definite matrix, $x \in \mathbb{R}^d$ $f(x,A) = x^T A^{-1} x$ is convex. This simply follows because

$$\begin{aligned} epi\ f &= \{(x,A,t) : t - x^T A^{-1} x \geq 0\} \\ &= \{(x,A,t) : \begin{bmatrix} t & x^T \\ x & A \end{bmatrix} \succeq 0\} \end{aligned}$$

which follows from the Schur Complement lemma.
Example: Just like every closed convex set is the intersection of half-spaces, every convex function is the point-wise maximum of affine functions. This follows from the epigraph formulation.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

January 22, 2015 Notes: Ashia Wilson and Benjamin Recht

Lecture 2: the gradient method

In this lecture, we take a tour of the gradient method, providing several different perspectives on this fundamental algorithm. The gradient method follows the simple algorithmic procedure:

1. Choose $x_0 \in \mathbb{R}^d$ and set $k = 0$
2. Choose $t_k > 0$ and set $x_{k+1} = x_k - t_k \nabla f(x_k)$ and $k = k + 1$,
3. Repeat 2 until converged.

This simple iterative procedure forms the basis for every algorithm we will study between now and the midterm. So we're going to do a deep dive on its properties for the next lecture or two.

1 Descent directions and optimality conditions

Let's suppose we want to minimize a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. Most of the algorithms we will consider start at some point x_0 and then aim to find a new point x_1 with a lower function value. The simplest way to do so is to find a direction v such that f is decreasing when moving along the direction v. This notion can be formalized by the following definition:

Definition 1 *v is a descent direction for f at x_0 if $f(x_0 + tv) < f(x_0)$ for some $t > 0$.*

A simple characterization of descent directions is given by the following proposition.

Proposition 1 *For a continuous differentiable function f on a neighborhood of x_0, if $v^T \nabla f(x) < 0$ then v is a descent direction.*

Proof By continuity, there exists a T such that $\nabla f(x_0 + tv)^T v < 0$ for all $t \in [0, T]$. By Taylor's theorem, $f(x_0 + tv) = f(x_0) + t\nabla f(x_0 + \tilde{t}v)^T v$ for some $\tilde{t} \in [0, t]$.Therefore $f(x_0 + tv) < f(x_0)$ and v is a descent direction. ■

Note that among all directions with unit norm,

$$\inf_{||v||=1} v^T \nabla f(x) = -||\nabla f||$$

which is achieved when $v = -\frac{\nabla f(x)}{||\nabla f(x)||}$. This means that $-\nabla f(x)$ is the direction of *steepest descent.*

This characterization allows us to provide conditions as to when x minimizes f.

Definition 2 *1. x_* is a* global minimizer *of f if $f(x_*) \leq f(x)\ \forall x \in \mathbb{R}^d$.*

2. x_* is a local minimizer *of f if there is a neighborhood $\mathcal{N}$ around x such that $f(x_*) \leq f(x)$ for all $x \in \mathcal{N}$.*

The first conditions concern local optimality.

Proposition 2 (Optimality Conditions) *1. $x_\star$ is a local minimizer only if $\nabla f(x_\star) = 0$*

2. If $\nabla^2 f$ is continuous and $x_\star$ is a local minimzer, then $\nabla^2 f(x_\star) \succeq 0$

3. If f is twice continuously differentiable, $\nabla f(x_\star) = 0$, $\nabla^2 f(x_\star) \succ 0$ then $x_\star$ is a local minimizer.

Proof

1. Since $-\nabla f(x_\star)$ is always a descent direction, the gradient must vanish.
2. If $x_\star$ is a local minimizer, $f(x_\star + td) \geq f(x_\star)$ for all d and some t sufficiently small. Using part 1 and Taylor's theorem,

$$f(x_\star + td) = f(x_\star) + \frac{1}{2}t^2 d^T \nabla^2 f(x_\star + \hat{t}d)d$$

for some $\hat{t} \in [0, t]$. Therefore $d^T \nabla^2 f(x_\star)d \geq 0$ for all d.

3. There exists an $r > 0$ such that $\nabla^2 f(x) \succ 0$ for all $x \in B(x_\star, r)$. Pick d with $||d|| < r$. Then we have

$$\begin{aligned} f(x_\star + d) &= f(x_\star) + d^T \nabla f(x_\star) + \frac{1}{2} d^T \nabla^2 f(x + td)d \qquad (\text{ for some } t \in [0,1]) \\ &\geq f(x_\star) \end{aligned}$$

proving $x_\star$ is a local minimizer.

■

For convex f, the situation is dramatically simpler. This is part of the reason why convexity is so appealing.

Proposition 3 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable convex function. Then $x_\star$ is a global minimizer of f if and only if $\nabla f(x_\star) = 0$.*

Proof What is particularly remarkable about the proof of this proposition is that it is almost tautological: if f is differentiable, then f is convex if and only if

$$f(x) \geq f(x_\star) + \nabla f(x_\star)^T (x - x_\star)$$

for all x. Using this equivalence, if $\nabla f(x_\star) = 0$, then $f(x) \geq f(x_\star)$ for all x. Conversely, if $f(x) \geq f(x_\star)$ for all x, we also have by the first-order convexity condition that

$$f(x_\star + t\nabla f(x_\star)) \geq f(x_\star) + t\|\nabla f(x_\star)\|^2$$

for all $t > 0$. Subtracting $f(x_\star)$ from both sides shows that $\|\nabla f(x_\star)\|^2 = 0$, and thus $\nabla f(x_\star) = 0$.
■

2 Fixed point iteration

Our first view of the gradient method is as a fixed point iteration. In order to solve for our optimal $x_\star$ it suffices to solve the (typically nonlinear) equation $\nabla f(x_\star) = 0$. A popular method for solving such an equation is by a fixed point iteration. Come up with some mapping $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ such that $x_\star = \Phi(x_\star)$ if and only if $\nabla f(x_\star) = 0$.

A simple candidate is $\Phi(x) = x - \alpha \nabla f(x)$. Let's assume that

1. There exits an $x_\star \in \mathcal{D}$ with $\nabla f(x_\star) = 0$.
2. $\Phi(x) = x - \alpha \nabla f(x)$ is *contractive* on $\mathcal{D}$ for some $\alpha > 0$: i.e., there is a $\beta \in [0, 1)$ such that

$$||\Phi(x) - \Phi(z)|| \leq \beta ||x - z|| \quad \forall x, z \in \mathcal{D}$$

.

Then if we run the gradient method starting at $x_0 \in \mathcal{D}$,

$$\begin{aligned} ||x_{k+1} - x_\star|| &= ||x_k - \alpha \nabla f(x_k) - x_\star|| \\ &= ||\psi(x_k) - \psi(x_\star)|| \\ &\leq \beta ||x_k - x_\star|| \\ &\vdots \\ &\leq \beta^{k+1} ||x_0 - x_\star|| \,. \end{aligned}$$

This derivation reveals that x_k converges *linearly* to $x_\star$. That is, at every iteration, the distance to the optimal solution is decreased by a constant factor.

As an aside, we say that this is linear convergence because

$$||x_{k+1} - x_\star|| \leq \beta ||x_k - x_\star||$$

So the iterates are related by a linear recursion. If we had instead

$$||x_{k+1} - x_\star|| \leq \beta ||x_k - x_\star||^2$$

we'd say that the convergence was quadratic.

OK, back to the gradient method. How many iterations must we run to guarantee that $\|x_k - x_\star\| \leq \epsilon$? A simple calculation reveals that

$$\boxed{k \geq -\frac{\log\left(\frac{||x_0 - x_\star||}{\epsilon}\right)}{\log(\beta)}}$$

suffice.

Quick check:

$$\begin{aligned}
\beta^k||x_0 - x_\star|| &\leq \epsilon \\
k\log\beta &\leq -\log\left(\frac{||x_0 - x_\star||}{\epsilon}\right) \\
-k\log\beta &\geq \log\left(\frac{||x_0 - x_\star||}{\epsilon}\right) \\
-k &\leq \frac{\log\left(\frac{||x_0 - x_\star||}{\epsilon}\right)}{\log\beta} \qquad (\log\beta < 0) \\
k &\geq -\frac{\log\left(\frac{||x_0 - x_\star||}{\epsilon}\right)}{\log\beta}
\end{aligned}$$

We are left with a few questions. First, when can we guarantee that Φ is a contractive map? This can be summarized by the following proposition:

Proposition 4 *If f is twice continuously differentiable and Φ is contractive then f must be convex.*

Proof First, by the definition of contractivity, we have for all $t > 0$ that

$$\frac{1}{t}||\Phi(x + t\Delta x) - \Phi(x)|| \leq \beta||\Delta x||$$

provided $x + t\Delta x \in \mathcal{D}$. Taking the limit as t goes to zero yields

$$\begin{aligned}
\beta||\Delta x|| &\geq \lim_{t\to 0}\frac{1}{t}||\psi(x + t\Delta x) - \psi(x)|| \\
&= \lim_{t\to 0}||\Delta x - \frac{\alpha}{t}(\nabla f(x + t\Delta x) - \nabla f(x))|| \\
&= ||\left[I - \alpha\nabla^2 f(x)\right]\Delta x||\,.
\end{aligned}$$

This inequality means

$$||I - \alpha\nabla^2 f(x)|| \leq \beta\,,$$

and hence we all of the eigenvalues of $I - \alpha\nabla^2 f(x)$ are between $-\beta$ and β. We can write this in terms of the semidefinite ordering as

$$\frac{1-\beta}{\alpha}I \preceq \nabla^2 f(x) \preceq \frac{1+\beta}{\alpha}I\,.$$

The first term in the partial order implies that f is convex. It actually implies that f is *strongly covnex*, a concept we will revisit shortly. The latter partial ordering states that the curvature of f must be globally bounded. In this next section we will discuss this curvature condition and its consequences. We will indeed show that, for differentiable functions, the gradient method converges at a linear rate if and only if f is strongly convex and has bounded curvature. ■

3 Lipschitz Continuity

Let's dive a bit more into the curvature condition that popped out of our analysis of the fixed point iteration. Unconstrained optimization algorithms should be scale invariant. For any $a > 0$ and $b \in \mathbb{R}$, $af(x) + b$ has the same optimal solution as $f(x)$. Our algorithms should respect this symmetry. One convenient way to set a scale is to define the Lipschitz constants associated with f and its gradients. We will see throughout the course that our precision should scale proportional to these constants.

Definition 3 (Lipschitz Continuity) *A mapping $\phi : \mathbb{R}^d \to \mathbb{R}^n$ is Lipschitz continuous on Ω if $\exists L \geq 0$ such that $\forall (x, y) \in \Omega$*

$$||\phi(x) - \phi(y)|| \leq L||x - y||$$

Note that if $\phi(x)$ is L-Lipschitz continuous and $a > 0$ then $a\phi(x)$ is aL-Lipschitz continuous.

For real-valued functions, the Lipschitz constant gives us a scale of how quickly f can vary from point to point. In particular, if the gradient is bounded, then so is the Lipschitz constant.

Proposition 5 *If $||\nabla f||$ is bounded on Ω, $M = \sup_z ||\nabla f(z)||$, then*

$$|f(x) - f(y)| \leq M||x - y||$$

Proof By Taylor's theorem,

$$f(x) - f(y) = \int_0^1 \nabla f(tx + (1-t)y)^T (x - y) dt \,.$$

Therefore,

$$\begin{aligned}
|f(x) - f(y)| &= \left| \int_0^1 \nabla f(tx + (1-t)y)^T (x - y) dt \right| \\
&\leq \int_0^1 \left| \nabla f(tx + (1-t)y)^T (x - y) dt \right| \\
&\leq \int_0^1 \left|\left| \nabla f(tx + (1-t)y)^T \right|\right| ||(x - y)|| \, dt \\
&= \left(\int_0^1 ||\nabla f(tx + (1-t)y)|| dt \right) ||x - y|| \\
&\leq M \, ||x - y||
\end{aligned}$$

Here, the first inequality is the triangle inequality. The second inequality is Cauchy-Schwartz, and the final inequality uses our bound on the gradient. ■

In a very similar fashion, the Lipschitz constant of the gradient controls how quickly the curvature of the function f can change. It is upper bounded by the operator norm of the Hessian:

Proposition 6 *If $||\nabla^2 f||$ is bounded in the operator norm on Ω, $L = \sup_x ||\nabla^2 f(x)||$, then*

$$|\nabla f(x) - \nabla f(y)| \leq L||x - y||$$

Proof The proof is more or less identical to the proof of Proposition 5. By Taylor's theorem.

$$\nabla f(x) - \nabla f(y) = \int_0^1 \nabla^2 f(tx + (1-t)y)^T (x-y) dt$$

And then we have the same chain of inequalities:

$$\begin{aligned} |\nabla f(x) - \nabla f(y)| &= \left| \int_0^1 \nabla^2 f(tx + (1-t)y)^T (x-y) dt \right| \\ &\leq \int_0^1 |\nabla^2 f(tx + (1-t)y)^T (x-y)| dt \\ &\leq \left(\int_0^1 ||\nabla^2 f(tx + (1-t)y)|| dt \right) ||x-y|| \\ &\leq L||x-y||. \end{aligned}$$

As above, the first inequality follows from the triangle inequality and the second from Cauchy-Schwarz. ■

We can derive an even stronger coupling between the Hessian and the Lipschitz constant of the gradient via the following chain of equivalences. Again, almost everything is a simple consequence of Taylor's theorem.

Proposition 7 *Suppose f is twice differentiable on Ω. Then the following are equivalent.*

1. *∇f is Lipschitz with constant L on Ω*
2. *$\forall x, y \in \Omega$, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}||x-y||^2$*
3. *$\forall x, y \in \Omega$, $\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq L||x-y||^2$*
4. *$\langle y - x, \nabla^2 f(x)(y-x) \rangle \leq L||x-y||^2$*

Proof ((1) $\Rightarrow$ (2)): Apply Taylor's theorem and Cauchy-Schwartz,

$$\begin{aligned} f(y) - f(x) - \nabla f(x)^T (y-x) &= \int_0^1 \left(\nabla f(ty + (1-t)x) - \nabla f(x) \right)^T (y-x) dt \\ &\leq ||y-x|| \int_0^1 ||\nabla f(ty + (1-t)x) - \nabla f(x)|| dt \\ &\leq ||y-x|| \int_0^1 Lt||y-x|| dt \\ &= L||y-x||^2 \int_0^1 t dt \\ &= \frac{L}{2}||y-x||^2 \end{aligned}$$

Here, the third inequality follows by applying Proposition 6. Rearranging terms proves the assertion.

((2) ⇒ (3)): Note that switching the roles of x and y, we have two inequalities.

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}||y-x||^2$$
$$f(x) \leq f(y) + \nabla f(y)^T(x-y) + \frac{L}{2}||x-y||^2$$

adding the inequalities proves this implication.
((3) ⇒ (4)): Substituting $x = z + t(u-z)$ and $y = z$ gives

$$\langle \nabla f(x + t(y-x)) - \nabla f(x), t(y-x)\rangle \leq Lt^2||y-x||^2$$

Dividing by t^2 gives

$$\left\langle \frac{\nabla f(x + t(y-x)) - \nabla f(x)}{t}, y-x \right\rangle \leq L||y-x||^2$$

Finally, taking the limit as $t \to 0$ proves the assertion.
((4) ⇒ (1)): Condition 4 is the same as saying that $||\nabla^2 f||$ is bounded in operator norm. We established from the previous proposition that bounds on the operator norm of the Hessian implies the gradients are Lipschitz. ■

4 Line search methods

We now turn to our second interpretation of gradient descent: that it is a line search method. The main idea here is to find a descent direction, and then minimize f—exactly or approximately—along that direction. That is, we will study the procedure

1. Pick v_k such that $\nabla f(x_k)^T v_k < 0$
2. Pick t_k to decrease f in the direction of v_k (1-D search).
3. Repeat 1 and 2 until converged.

There are a variety of ways to choose v_k. The most obvious choice is $-\nabla f(x_k)$. This is the gradient method. As we discussed in Section 1, $-\nabla f(x_k)$ is the direction of *steepest* descent. Though it seems odd that we would pick any direction other than the steepest descent direction, we will find many examples where being a bit less greedy can result in considerably faster convergence.

The line search part is a bit more nebulous. When we analyzed the gradient method as a fixed point iteration, we pulled the α parameter out of a hat. But there are far more systematic choices when we view the gradient method as a descent method. Some examples include

1. *Exact Line Search*: We choose $t_k = \arg\min_{t \geq 0}\{f(x_k + tv_k)\}$. This method relies on being able to solve differentiable, one-dimensional optimization problems quickly. While this is not generically easy, there are many cases where exact line search is straightforward. The most notable example is when f is a multivariate polynomial. In this case, $f(x_k + tv_k$ is a polynomial in t, and the minimum can be found by computing the derivative and root finding.

2. *Constant Step Size*: As we saw in section 2, constant step sizes can yield rapid convergence rates. The main drawback with these methods is one often needs some prior information about f to properly choose the stepsize.

3. *Diminishing stepsize*: Another canonical choice is to pick a stepsize sequence that tends to zero but whose sum diverges. For example $t_k = C/k$ for some C. We will return to this more when we analyze nonsmooth and stochastic optimization.

4. *Goldstein-Armijo Condition*: This condition has two parameters $0 < \alpha < \beta < 1$. We choose t such that

$$f(x_k + tv_k) \leq f(x_k) + \alpha t \nabla f(x_k)^T v$$
$$f(x_k + tv_k) \geq f(x_k) + \beta t \nabla f(x_k)^T v$$

   The idea behind the Armijo condition is displayed in Figure 1. When $\alpha = 1$ and f is convex, the linear approximation lies below the curve. As we shrink α to zero, the approximation becomes more and more of an over approximation of f. The first condition guarantees that we lie *below* the over-approximation and hence are not moving too far. The second condition guarantees that we make some progress on this iteration. The Armijo-Goldstein condtions are the main principle behind back-tracking line search.

5. *Wolfe Conditions* The Wolfe conditions are primarily geared towards quasi-Newton methods, and we will spend more time on these when we revisit that topic. They are

$$f(x_k + tv_k) \leq f(x_k) + \alpha t \nabla f(x_k)^T v$$
$$\nabla f(x_k + tv_k)^T v_k \geq \gamma \nabla f(x_k)^T v$$

   where $\gamma \in (\alpha, 1)$. The first condition is the sufficient decrease condition from the Armijo condtitions. The second states that the derivative of $\phi(t) = f(x_k + tv_k)$ after taking our step is sufficiently larger than at $t = 0$. This makes sense as if the slope is smaller, we should keep moving along the direction v!

Let's now provide a simple analysis of the gradient method as a line search method. Suppose f has L-Lipschitz gradients. Observe that for any t, the Lipschitz continuity of the gradient implies

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - t||\nabla f(x_k)||^2 + \frac{Lt^2}{2}||\nabla f(x_k)||^2 \\ &= f(x_k) - t\left(1 - \frac{tL}{2}\right)||\nabla f(x_k)||^2 . \end{aligned}$$

Note that if $0 < t < \frac{2}{L}$, then $t\left(1 - \frac{tL}{2}\right) > 0$, and the function value is decreasing. In particular, if you choose $t = 1/L$, we have

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}||\nabla f(x_k)||^2 \tag{1}$$

Also note that if you perform exact line search, (1) holds. This is because

$$\min_{t>0} f(x_k - t\nabla f(x_k)) \leq f(x_k - 1/L \nabla f(x_k))$$

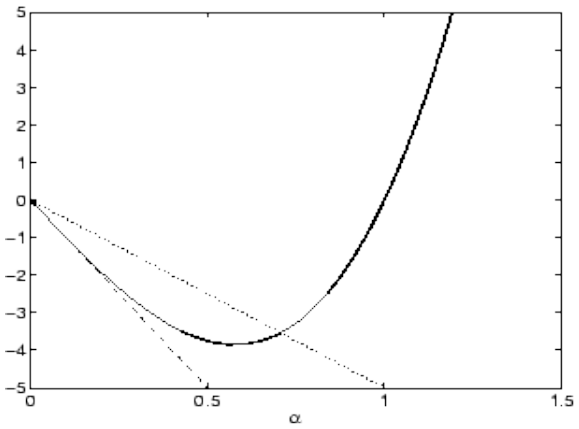

Figure 1: A graphical display of the Armijo condition. The dashed line is the tangent curve defined by the gradient ($\alpha = 1$). As we shrink α, the curve now lies partially above f. Note that if we like under the stronger dashed curve, then we are likely to not overshoot the minimum.

Define the quantity

$$\eta := \begin{cases} 2L & \text{for exact line search} \\ \frac{1}{t\left(1-\frac{TL}{2}\right)} & \text{for constant step size} \end{cases}$$

We can rearrange (1) and sum over the iterates of the algorithm to find that

$$\begin{aligned} \sum_{k=0}^{N} ||\nabla f(x_k)||^2 &\leq \eta \sum_{k=0}^{N} f(x_k) - f(x_{k+1}) \\ &= \eta[f(x_0) - f(x_N)] \\ &\leq \eta[f(x_0) - f(x_\star)]\,. \end{aligned}$$

The second line follows because the sum telescopes.

This implies that $\lim_{N\to\infty} ||\nabla f(x_N)|| = 0$. More concretely

$$\begin{aligned} \min_{0\leq k\leq N} ||\nabla f(x_k)|| &\leq \sqrt{\frac{\eta[f(x_0) - f(x_N)]}{N}} \\ &\leq \sqrt{\frac{\eta[f(x_0) - f(x_\star)]}{N}} \\ &\leq \sqrt{\frac{\eta\frac{L}{2}||x_0 - x_\star||^2}{N}}\,. \end{aligned}$$

For exact line search, this guarantees that we find a point x

$$\|\nabla f(x)\| \leq \frac{L||x_0 - x_\star||}{N^{1/2}}\,.$$

For constant step size we are guaranteed to find a point with

$$\|\nabla f(x)\| \leq \sqrt{\frac{1}{2\beta(1-\frac{\beta}{2})}} \frac{L||x_0 - x_\star||}{N^{1/2}}$$

when our stepsize is $t = \frac{\beta}{L}$.

Note that this convergence rate is very slow, and only tells us that we will find a stationary point. We need more structure about f to guarantee faster convergence and global optimality.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

January 22, 2015 Notes: Ashia Wilson and Benjamin Recht

Lecture 3: the gradient method (loose ends)

Let's wrap up a few loose ends about the gradient method. First, we will discuss convex and strongly convex functions and estimate the convergence of the gradient method when applied to these functions. Then, we'll turn to how to estimate step-sizes without knowledge of the Lipschitz constants or strong convexity parameters.

1 Strong Convexity

A function $f : \mathbb{R}^d \to \mathbb{R}$ is *strongly convex* if there is a scalar $m > 0$ such that

$$f(z) \geq f(x) + \nabla f(x)^T(z - x) + \tfrac{m}{2}\|z - x\|^2 \tag{1}$$

Strong convexity asserts that f can be lower bounded by quadratic functions. These functions change from point to point, but only in the linear term. It also tells us that the curvature of the function is bounded away from zero. Note that if a function is strongly convex *and* has L-Lipschitz gradients, then f is bounded above and below by simple quadratics. This sandwiching will enable us to prove the linear convergence of the gradient method.

The simplest strongly convex function is the squared Euclidean norm $\|x\|^2$. Any convex function can be perturbed to form a strongly convex function by adding any small multiple of the squared Euclidean norm. In fact, if f is any differentiable function with L-Lipschitz gradients, then

$$f_\mu(x) = f(x) + \mu\|x\|^2$$

is strongly convex for μ large enough. Verifying this fact is a fun exercise.

As another canonical example, note that a quadratic function $f(x) = \frac{1}{2}x^TQx$ is strongly convex if and only if the smallest eigenvalue of Q is strictly positive.

Strongly convex functions are in essence the "easiest" functions to optimize by first-order methods. First, the norm of the gradient provides useful information about how far away we are from optimality. Note that if we minimize the right hand side of our strong convexity condition with respect to x, we find that the minimizer is $x - \frac{1}{m}\nabla f(x)$. Plugging that into (1), we find

$$\begin{aligned} f(z) &\geq \min_x f(x) + \nabla f(x)^T(z - x) + \frac{m}{2}||z - x||^2 \\ &\geq f(x) - \nabla f(x)^T\frac{1}{m}\nabla f(x) + \frac{m}{2}\left\|\frac{1}{m}\nabla f(x)\right\|^2 \\ &\geq f(x) - \frac{1}{2m}||\nabla f(x)||^2 \end{aligned} \tag{2}$$

Now if $||\nabla f(x)|| < \delta$ then

$$f(x) - f(x_\star) \leq \frac{||\nabla f(x)||^2}{2m} \leq \frac{\delta^2}{2m}$$

Thus, when the gradient is small, we are close to having found a point with minimal function value. We can even derive a stronger result about the distance of x to the optimal point $x_\star$. Using (1) and Cauchy-Schwartz, we have

$$\begin{aligned} f(x_{opt}) &\geq f(x) + \nabla f(x)^T (x_\star - x) + \frac{m}{2}||x - x_\star||^2 \\ &\geq f(x) - ||\nabla f(x)||\, ||x_\star - x|| + \frac{m}{2}||x - x_\star||^2 \end{aligned}$$

Rearranging terms proves that

$$||x - x_\star|| \leq \frac{2}{m}||\nabla f(x)||\,. \tag{3}$$

This says we can estimate the distance to the optimal value purely in terms of the norm of the gradient.

An immediate consequence of (3) is the following

Corollary 1 *If f is strongly convex then f has a unique optimal solution.*

Essentially, strongly convex functions are nice, wide bowls, and we just need to roll downhill to the bottom.

We close this discussion of strong convexity by proving that for differentiable functions, strong convexity is equivalent to the Hessian having positive eigenvalues. We encountered this fact when we were discussing contraction mappings in the previous lecture.

Proposition 2 *If f is strongly convex and two-times differentiable, then $\nabla^2 f(x) \succeq mI$*

Proof Using the fact that

$$f(x) \leq f(y) + \nabla f(y)^T (x - y) + \frac{m}{2}||x - y||^2$$

and

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}||y - x||^2$$

we can add these two inequalities together and get

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq m||x - y||^2\,.$$

Setting $x = u + \alpha v$ and $y = u$, for u and v in $\mathbb{R}^d$ yields

$$\langle \nabla f(u + \alpha d) - \nabla f(v), \alpha v \rangle \geq m\alpha^2 ||d||^2$$

Dividing through by α^2 and taking the limit as α goes to zero proves

$$d^T \nabla^2 f(x) d \geq m||x - y||^2$$

as desired. ■

2 Three proofs of convergence of the gradient method on strongly convex functions

Strongly convex functions allow us to show off three popular search strategies for proving convergence of optimization algorithms.

2.1 Descent analysis

Recall from last lecture that if we run the gradient method with stepsize $1/L$ (or use exact line search), we have the inequality

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}||\nabla f(x_k)||^2 .$$

Subtracting $f(x_\star)$ from both sides and using (2) gives

$$f(x_{k+1}) - f(x_{opt}) \leq \left[1 - \frac{m}{L}\right] (f(x_k) - f(x_{opt}))$$

Here $\frac{m}{L}$ is an estimate of the worst-case "condition number" of the Hessian. When m is significantly smaller than L, f will have very eccentric level sets. Since the gradient is orthogonal to the contours of f, this will cause the gradient method to oscillate rapidly, and convergence will be quite slow.

2.2 Contraction Mapping

If f is twice continuously differentiable, note that the map

$$\Phi(x) = x - \alpha \nabla f(x)$$

is a contraction mapping for any $0 < \alpha < \frac{2}{L+m}$. To see this, observe

$$\begin{aligned}
\|\Phi(x) - \Phi(y)\| &= \|x - \alpha\nabla f(x) - (y - \alpha \nabla f(y))\| \\
&\leq \left\| \int_0^\infty (I - \alpha \nabla^2 f(x + t(y-x))(y-x) dt \right\| \\
&\leq \sup_z \|I - \alpha \nabla^2 f(z)\| \|y - z\| .
\end{aligned}$$

Note that the minimum eigenvalue of $\nabla^2 f(z)$ is at least m and the maximum eigenvalue is at least L. Therefore the eigenvalues of $I - \alpha \nabla^2 f(z)$ are at most $\max(1 - \alpha L, 1 - \alpha m)$ and at least $\min(1 - \alpha L, 1 - \alpha m)$. Therefore,

$$\|I - \alpha \nabla^2 f(z)\| \leq \max(|1 - \alpha L|, |1 - \alpha \ell|) .$$

The right-hand side is minimized when $\alpha = \frac{2}{L+m}$. Moreover, the quantity is less than 1 and Φ is contractive whenever $0 < \alpha < 2/L$.

Note that with the stepsize $\alpha = \frac{2}{L+m}$, we see that the iterates converge linearly with the rate

$$\frac{L - m}{L + m}$$

which is slightly faster than the rate $(1 - L/m)$ derived above.

2.3 Lyapunov Analysis

A function $V : \mathbb{R}^d \to \mathbb{R}$ is a Lypaunov function for an algorithm if

1. $V(x) \geq 0$
2. $V(x_\star) = 0$
3. $V(x_{k+1}) < V(x_k)$ for all iterates of the algorithm

The existence of a Lyapunov question is clearly sufficient to guarantee asymptotic convergence of a method.

Note that we showed in Section 2.1 that $f(x) - f_\star$ was a Lyapunov function. Here, we can show that $\|x_k - x_\star\|$ is also a Lyapunov function. This follows from the contraction argument, but it's worth mentioning this connection to Lyapunov analysis, as this will be another tool in our belt moving forward.

2.4 From rates to iterations

When we have linear convergence, we are often left with expressions of the form

$$f(x_k) - f(x_\star) \leq (1 - \beta)^k D_0$$

where D_0 defines some initial distance or deviation in the function value. We'd like to turn these expressions into bounds on how many iterations it takes to get $f(x_k) - f(x_\star) \leq \epsilon$.

Let's say we have run for N iterations. Taking logarithms, we have the inequality

$$\log(\epsilon) \geq N \log\left[1 - \beta\right] + \log D_0$$

Rearranging terms, we find that

$$N \geq \frac{-\log(D_0/\epsilon)}{\log(1-\beta)}$$

Using the inequality $1 - x \leq \exp(-x)$ or equivalently $\log(1-x) \leq -x$ we get the sufficient condition that

$$N \geq \beta^{-1} \log(D_0/\epsilon)\,.$$

are required to get $f(x_k) - f(x_\star) \leq \epsilon$.

For the gradient method, this means

$$N \geq \frac{L}{m} \log\left(\frac{f(x_0) - f(x_\star)}{\epsilon}\right)$$

iterations suffice to guarantee convergence to tolerance ϵ. This ratio L/m governs how regular the curvature of f is, and it is akin to the condition number of a matrix. Note that this is not the condition number of the Hessian, per se. We can define 1-dimensional functions where L/m is arbitrarily large, but the Hessian has condition number 1 everywhere.

3 Gradient method with "weak" convexity

The gradient method for functions that are convex but not strongly convex also converges faster than in the nonconvex case we previously studied. We'll refer to such functions as "weakly convex" to contrast them to strongly convex functions. Note that all convex functions are weakly convex, but not all convex functions are strongly convex.

The analysis of this case uses an important property of convex functions. Indeed, the following inequality completely characterizes the set of convex functions with L-Lipschitz gradients.

Lemma 3 *f is convex with L-Lipschitz gradients if and only if*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{L}||\nabla f(x) - \nabla f(y)||^2$$

Proof Define $\varphi_{x_0}(y) = f(y) - \langle \nabla f(x_0), y \rangle$ (i.e. φ_{x_0} is just f perturbed by a linear function). φ_{x_0} has lipschitz gradients. Furthermore

$$\nabla_y \varphi_{x_0}(y) = \nabla f(y) - \nabla f(x_0)$$

implying that $x_0 \in \arg\min \varphi_{x_0}$. Therefore we have

$$\begin{aligned} \varphi(x_0) &= \varphi(y - \frac{1}{L}(\nabla f(y) + \nabla f(x_0)) \\ &\leq \varphi(y) - \frac{1}{2L}||\nabla f(y) - \nabla f(x_0)||^2 \end{aligned}$$

where the last step follows from a standard Lipschitz upper bound $f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}||x - y||^2$

For the converse, note that convexity follows because the quadratic term is nonnegative. The Lipschitz gradients follows because if we swap the role of x and y and add the inequalities, we have

$$|\langle \nabla f(x) - \nabla f(y), x - y \rangle| \geq \frac{1}{2L}||\nabla f(x) - \nabla f(y)||^2$$

Cauchy-Schwartz completes the proof. ■

It is interesting about this lemma is that the set of convex functions with L-Lipschitz gradients is characterized by *one* inequality. Rather than two. This property is called *co-coercivity* and will be used many times in the course.

With the co-coercivity lemma in hand, the following theorem gives our convergence rate for general convex functions.

Lemma 4 *If f is convex with L-Lipschitz gradients, the gradient method with $t = \frac{1}{L}$ satisfies*

$$f(x_k) - f(x^*) \leq \frac{2L||x_0 - x^*||^2}{k + 1}$$

Proof Recall again that we have

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}||\nabla f(x_k)||^2$$

We also have (by first order conditions),

$$f(x_k) - f(x_\star) \leq \langle \nabla f(x_k), x_k - x* \rangle \leq ||\nabla f(x_k)||\, ||x_k - x^*|| \tag{4}$$

Now by the lemma

$$\begin{aligned} ||x_{k+1} - x^*||^2 &= ||x_k - \frac{1}{L}\nabla f(x_k) - x^*||^2 \\ &= ||x_k - x^*||^2 - \frac{2}{L}\langle \nabla f(x_k), x_k - x^* \rangle + \frac{1}{L^2}||\nabla f(x_k)||^2 \end{aligned}$$

By first order conditions

$$\begin{aligned} -\frac{2}{L}\langle \nabla f(x_k), x_k - x^* \rangle &\leq \frac{2}{L}(f(x^*) - f(x_k)) \\ &\leq \frac{2}{L}(f(x_{k+1}) - f(x_k)) \\ &\leq \frac{2}{L}\left(-\frac{1}{2L}||\nabla f(x_k)||^2\right) \\ &\leq -\frac{1}{L^2}||\nabla f(x_k)||^2 \end{aligned}$$

Therefore we get

$$||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 \leq ||x_0 - x^*||^2$$

This means that the iterates live in a bounded set. Using (4),

$$\begin{aligned} f(x_{k+1}) - f(x^*) &\leq f(x_k) - f(x^*) - \frac{1}{2L\|x_k - x_\star\|^2}(f(x_k) - f(x^*))^2 \\ &\leq f(x_k) - f(x^*) - \frac{1}{2L\|x_0 - x_\star\|^2}(f(x_k) - f(x^*))^2 \end{aligned}$$

Defining $D_0 \equiv ||x_0 - x^*||$ we have

$$f(x_{k+1}) - f(x^*) \leq f(x_k) - f(x^*) - \frac{1}{2LD_0^2}(f(x_k) - f(x^*))^2 \tag{5}$$

This recursion is similar to the bound we had for linear convergence, but now we have an additional quadratic term. The rest of the proof is just algebraic manipulation to show that the convergence rate is achieved.

Define $\Delta_k := f(x_k) - f(x_\star)$. Then the recursion (5) becomes

$$\Delta_{k+1} \leq \Delta_k - \frac{1}{2LD_0^2}\Delta_k^2\,.$$

Dividing both sides by $\frac{1}{\Delta_k \Delta_{k+1}}$ we have

$$\frac{1}{\Delta_k} \leq \frac{1}{\Delta_{k+1}} - \frac{1}{2LD_0^2}\frac{\Delta_k}{\Delta_{k+1}}$$

Rearranging the inequality gives

$$\begin{aligned}\frac{1}{\Delta_{k+1}} &\geq \frac{1}{\Delta_k} + \frac{1}{2LD_0^2}\frac{\Delta_k}{\Delta_{k+1}} \\ &\geq \frac{1}{\Delta_k} + \frac{1}{2LD_0^2} \\ &\geq \frac{1}{\Delta_0} + \frac{k+1}{2LD_0^2} \\ &\geq \left(\frac{1}{2} + \frac{k+1}{2}\right)\left(\frac{1}{LD_0^2}\right) \\ &= \frac{k+2}{2LD_0^2}\end{aligned}$$

Where we used $\Delta_0 \leq 2LD_0^2$. Thus,

$$f(x_{k+1}) - f(x^*) \leq \frac{2L||x_0 - x^*||^2}{k+2}$$

■

4 Backtracking Line Search

Finally, let's figure out how to automatically tune the step-size without knowledge of the Lipschitz constant. One of the most popular schemes is called *backtracking line seach*. The idea is simple: we choose a starting step-size, and if the function is not sufficiently decreased, we choose a smaller step. This is summarized as:

1. First we pick a direction $v = -\nabla f(x_k)$.
2. Start with $t = 1$
3. Test the Armijo Condition:

$$f(x_k - t\nabla f(x_k)) \leq f(x_k) - \gamma t||\nabla f(x_k)||^2$$

   (a) If fails $t = \beta t$
   (b) If succeeds, terminate

A decent rule of thumb for this procedure is to choose $\beta = .8$ and $\gamma = \frac{1}{2}$. But your mileage may vary.

Analyzing backtracking line search is not much more difficult than analyzing constant stepsize. Assume $\gamma \leq \frac{1}{2}$. Then at termination,

$$f(x_{k+1}) \leq f(x_k) - \gamma t||\nabla f(x_k)||^2$$

by assumption. Note that we *always* have

$$f(x_{k+1}) \leq f(x_k) - t\left(1 - \frac{tL}{2}\right)||\nabla f(x_k)||^2,$$

for our step t.

There are now two cases to analyze:

1. Case 1: $t_0 = 1$ satisfies the Armijo condition. In this case,
$$\Rightarrow ||\nabla f(x_k)||^2 \leq \gamma^{-1}(f(x_k) - f(x_{k+1}))\,.$$

2. Case 2: We succeed at some t after backtracking. But this means that $\beta^{-1}t$ failed to satisfy the Armijo condition. Writing this out explicitly:
$$\begin{aligned} f(x_k) - \gamma\beta^{-1}t||\nabla f(x_k)||^2 &\leq f(x_k - \beta^{-1}t\nabla f(x_k)) \\ &\leq f(x_k) - \beta^{-1}t\left(1 - \frac{tL}{2}\right)||\nabla f(x_k)||^2 \end{aligned}$$
If we match terms here, we find that
$$-\gamma\beta^{-1}t \leq \beta^{-1}t\left(1 - \frac{\beta^{-1}tL}{2}\right)$$
Or, again rearranging things,
$$t \geq \frac{2(1-\gamma)\beta}{L} \geq \frac{\beta}{L}$$

Therefore we can combine the cases and find
$$\begin{aligned} ||\nabla f(x_k)||^2 &\leq \frac{L}{\beta\gamma}\left[f(x_k) - f(x_{k+1})\right] \\ &\leq \frac{L}{\gamma\min(L,\beta)}\left[f(x_k) - f(x_{k+1})\right] \end{aligned}$$

Now we can apply the previous analyses.

5 Complexity

How many iterations will this take?

$$\begin{aligned} F &: \text{Time to compute the function} && \sim O(d) \\ G &: \text{Time to compute the gradient} && \sim O(d) \\ H &: \text{Time to compute the Hessian} && \sim O(d^2) \end{aligned}$$

Definition 1 (Gaxpy) *A gaxpy is a scale and add of a vector (i.e. our gradient method update* $x_k \leftarrow x_{k+1} + \alpha g$*)*

For Gradient method on convex function, we have
$$(\underbrace{\text{time for gaxpy}}_{O(d)} + \underbrace{\text{Gradient call}}_{O(d)} + \underbrace{\text{backtracking}}_{O(\log(L))F}) \cdot \frac{L||x_0 - x^*||}{N}$$

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

January 29, 2015 Notes: Ashia Wilson and Benjamin Recht

Lecture 4: Momentum

We saw last lecture that the gradient method achieved a linear rate of convergence for strongly convex functions. Namely, if the gradients were L-Lipschitz, and f was strongly convex with parameter m, then

$$O\left(\frac{L}{m}\log(1/\epsilon)\right)$$

iterates sufficed to converge to an accuracy of ϵ. For weakly convex functions,

$$O\left(\frac{L}{\epsilon}\right)$$

iterations suffice. The question remains: is this the best we can do? In this lecture we describe a minor modification to the gradient method that results in a major reduction in iteration complexity.

1 Appealing to differential equations for motivation

We described in the last lecture that the gradient method could be overly greedy, always taking the steepest descent direction. When the contours of f are very narrow and elongated, this strategy results in oscillation.

One possible way to avert this situation is to add *momentum* to the iterate. That is, in the spirit of the Wolfe conditions, if the direction we were moving was a good one, we might consider moving along this direction for more than one step.

The intuition for these momentum methods comes from looking at our algorithm as a differential equation. The gradient method is akin to moving down a potential well where the potential is defined by the gradient of f:

$$\frac{dx}{dt} = -\nabla f(x)$$

This differential equation has fixed points precisely when $\nabla f(x) = 0$.

Of course, there are other differential equations whose stationary points are precisely those where the gradient vanishes. In particular, the second order differential equation:

$$\mu\frac{d^2x}{dt^2} = -\nabla f(x) - b\frac{dx}{dt}$$

also converges to a point where $\nabla f(x) = 0$. This corresponds to the Euler-Lagrange equations associated with an object in a potential well in the presence of friction or viscosity. If $b = 0$, then the system may always gain acceleration. Moreover, the trajectories will tend to continue to move in the direction they were moving before (heavier objects move down hill faster than light objects in the presence of friction).

If we approximate this ODE using simple finite differences, we have

$$\mu \frac{x(t+\Delta t) - 2x(t) + x(t-\Delta t)}{\Delta t^2} \approx -\nabla f(x(t)) - b\frac{x(t) - x(t-\Delta t)}{\Delta t}$$

Rearranging the terms in this expression gives the finite difference equation:

$$x(t+\Delta t) = x(t) - \frac{\Delta t^2}{\mu}\nabla f(x(t)) + \left(1 - \frac{b}{\mu}\Delta t\right)(x(t) - x(t-\Delta t))$$

which is precisely the momentum method we are interested in.

2 Momentum methods

The method we derived in the previous section is known as *the heavy ball method*, as it appeals to physical intuition. Rewriting things in a more pleasant way we have

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1}),$$

where α and β are positive constants to be determined. Define

$$\begin{aligned} y_k &= x_{k+1} - x_k \\ &= -\alpha\nabla f(x_k) + \beta(x_k - x_{k-1}) \\ &= -\alpha\nabla f(x_k) + \beta y_{k-1} \end{aligned}$$

With this identification, we can rewrite the iteration in terms of two sequences:

$$\begin{aligned} x_{k+1} &= x_k + y_k \\ y_k &= -\alpha\nabla f(x_k) + \beta y_{k-1} \end{aligned}$$

A similar algorithm is known as *Nesterov's accelerated* method or *Nesterov's optimal method*:

$$x_{k+1} = x_k - \alpha\nabla f(x_k + \beta(x_k - x_{k-1})) + \beta(x_k - x_{k-1}).$$

This method only differs in their discretization of the ODE. In the two state version, Nesterov's method becomes

$$\begin{aligned} x_{k+1} &= x_k + y_k \\ y_k &= -\alpha\nabla f(x_k + \beta y_{k-1}) + \beta y_{k-1} \end{aligned}$$

Another popular way to write Nesterov's accelerated method is by the iterations

$$\begin{aligned} z_k &= x_k + \beta_k(x_k - x_{k-1}) \\ x_{k+1} &= z_k - \alpha_k\nabla f(z_k). \end{aligned}$$

All three of these formulations are equivalent to one another.

It turns out that Nesterov's method converges more rapidly than the Heavy Ball method for general convex functions. This is peculiar, because as you will see in next week's homework, they achieve the same rate of convergence when f is a quadratic. Nonetheless, we will focus our attention on Nesterov's method, even though the two approaches look very similar to one another.

3 Analysis of the Nesterov's method for convex quadratic functions

Later in the semester, we will analyze Nesterov's method for general convex f. But, unfortunately, the proofs are still a bit unintuitive or require some more advanced concepts that we have not yet covered. So we will focus our attention on quadratics for now. But one must be careful! Just because one proves that an algorithm converges for all strongly convex quadratics, it does not mean that the method necessarily converges for non-quadratic functions. A simple one dimensional example will be studied in the homework.

Consider the inhomogenous convex quadratic objective

$$f(x) = \frac{1}{2}x^T Q x - p^T x + r$$

where $LI \succeq Q \succeq mI$. Strongly convex quadratics can be solved in closed form by direct calculation.

$$x_\star = Q^{-1}p$$

is the minimizer of f and

$$\nabla f(x) = Qx - p = Q(x - x_\star)$$

Plugging in these simplications, our algorithm takes the form

$$\begin{aligned} x_{k+1} - x_\star &= x_k - x_\star - \alpha Q(x_k + \beta(x_k - x_{k-1}) - x_\star) + \beta(x_k - x_{k-1}) \\ &= x_k - x_\star - \alpha Q(x_k + \beta((x_k - x_\star) - (x_{k-1} - x_\star)) - x_\star) + \beta((x_k - x_\star) - (x_{k-1} - x_\star)) \,. \end{aligned}$$

In this second equality, we just added and subtracted $x_\star$ a few times. This trick let's us write the algorithm as the simple matrix equation

$$\begin{bmatrix} x_{k+1} - x_\star \\ x_k - x_\star \end{bmatrix} = T \begin{bmatrix} x_k - x_\star \\ x_{k-1} - x_\star \end{bmatrix} \tag{1}$$

Where T is the matrix

$$T := \begin{bmatrix} (1+\beta)(I - \alpha Q) & -\beta(I - \alpha Q) \\ I & 0 \end{bmatrix} \tag{2}$$

This matrix governs the evolution of the algorithm.

We want to show that all trajectories of the algorithm converge to 0, implying that x_k converges to $x_\star$. The following theorem guarantees such convergence.

Definition 1 *The* spectral radius *of a matrix A is equal to $\rho(A) := \max\{|\lambda| \,:\, \lambda \text{ is an eigenvalue of } A\}$.*

Theorem 1 *Suppose $A \in \mathbb{R}^{d\times d}$. Then $\rho(A) < \rho$ if and only if there exists a $P \succ 0$ satisfying $A^T P A - \rho^2 P \prec 0$.*

Proof If $\rho(A) < \rho$, then the matrix

$$P := \sum_{k=0}^{\infty} \rho^{-2k} (A^k)^T (A^k)$$

is well defined, positive definite because the first term in the sum is a multiple of the identity, and satisfies $A^T P A - \rho^2 P = -\rho^2 I_d \prec 0$. Conversely, assume the LMI has a solution $P \succ 0$ and let λ be an eigenvalue of A with corresponding eigenvector v. Then

$$0 > v^T A^T P A v - \rho^2 v^T P v = (|\lambda|^2 - \rho^2) v^T P v$$

But since $v^T P v > 0$, we must have that $|\lambda| < \rho$. ■

Now suppose we are studying the iteration (1). Then, if there exists a $P \succ 0$ satisfying $T^T P T - \rho^2 P \prec 0$, we have

$$\begin{bmatrix} x_{k+1} - x_\star \\ x_k - x_\star \end{bmatrix}^T P \begin{bmatrix} x_{k+1} - x_\star \\ x_k - x_\star \end{bmatrix} < \rho^2 \begin{bmatrix} x_k - x_\star \\ x_{k-1} - x_\star \end{bmatrix}^T P \begin{bmatrix} x_k - x_\star \\ x_{k-1} - x_\star \end{bmatrix} \tag{3}$$

along all trajectories. If $\rho < 1$, then the sequence $\{x_k\}$ converges linearly to $x_\star$. Iterating (3) down to $k = 0$, we see that

$$\begin{bmatrix} x_k - x_\star \\ x_{k-1} - x_\star \end{bmatrix}^T P \begin{bmatrix} x_k - x_\star \\ x_{k-1} - x_\star \end{bmatrix} < \rho^{2k} \begin{bmatrix} x_0 - x_\star \\ x_{-1} - x_\star \end{bmatrix}^T P \begin{bmatrix} x_0 - x_\star \\ x_{-1} - x_\star \end{bmatrix} .$$

Assuming that $x_0 = x_{-1}$, this implies that

$$\|x_k - x_\star\| < \sqrt{2 \operatorname{cond}(P)}\, \rho^k \|x_0 - x_\star\|$$

where $\operatorname{cond}(P)$ is the condition number of P. The function

$$V(x, z) = \begin{bmatrix} x - x_\star \\ z - x_\star \end{bmatrix}^T P \begin{bmatrix} x - x_\star \\ z - x_\star \end{bmatrix}$$

is a Lyapunov function for the algorithm. This function strictly decreases over all trajectories and hence certifies that the algorithm is *stable*, i.e., converges to nominal values. For quadratic f, we are able to construct a quadratic Lyapunov function by doing an elementary eigenvalue analysis. But this proof doesn't generalize to the non-quadratic case, which is why we reserve the study of Nesterov's method on more general convex functions for later in the semester.

With this theorem in hand, it suffices to find parameters α and β such that T has its eigenvalues to have small modulus as possible.

Proposition 2 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex quadratic function $f(x) = \frac{1}{2} x^T Q x - p^T x + r$ with $mI \preceq Q \preceq LI$. Let $\kappa := L/m$. Then Nesterov's accelerated method with $\alpha = \frac{1}{L}$ and $\beta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ converges and $\rho(T) \leq 1 - \kappa^{-1/2}$.*

Proof We will upper bound the spectral radius of T by explicitly computing all of the eigenvalues and upper bounding their magnitudes.

To proceed, write the eigenvalue decomposition of Q as $Q = U_0 \Lambda U_0^T$, where $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$. By defining the permutation matrix Π as follows:

$$\Pi_{ij} = \begin{cases} 1 & i \text{ odd}, \ j = (i+1)/2 \\ 1 & i \text{ even}, \ j = d + (i/2) \\ 0 & \text{otherwise} \end{cases} .$$

we have by applying a similarity transformation to the matrix T that

$$\begin{aligned}
&\Pi \begin{bmatrix} U_0 & 0 \\ 0 & U_0 \end{bmatrix}^T \begin{bmatrix} (1+\beta)(I-\alpha Q) & -\beta(I-\alpha Q) \\ I & 0 \end{bmatrix} \begin{bmatrix} U_0 & 0 \\ 0 & U_0 \end{bmatrix} \Pi^T \\
&= \Pi \begin{bmatrix} (1+\beta)(I-\alpha\Lambda) & -\beta(I-\alpha\Lambda) \\ I & 0 \end{bmatrix} \Pi^T \\
&= \begin{bmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_d \end{bmatrix},
\end{aligned}$$

where

$$T_i = \begin{bmatrix} (1+\beta)(1-\alpha\lambda_i) & -\beta(1-\alpha\lambda_i) \\ 1 & 0 \end{bmatrix}, \quad i = 1, 2, \ldots, d.$$

The eigenvalues of T are the eigenvalues of T_i, for $i = 1, 2, \ldots, d$. The eigenvalues of T_i are the roots of the following quadratic:

$$\nu_{i,j}{}^2 - (1+\beta)(1-\alpha\lambda_i)\nu_{i,j} + \beta(1-\alpha\lambda_i) = 0, \tag{4}$$

which are given by the formula for the quadratic equation:

$$\begin{aligned}
\nu_{i,1} &= \frac{1}{2}\left[(1+\beta)(1-\alpha\lambda_i) + i\sqrt{4\beta(1-\alpha\lambda) - (1+\beta)^2(1-\alpha\lambda_i)^2}\right], \\
\nu_{i,2} &= \frac{1}{2}\left[(1+\beta)(1-\alpha\lambda_i) - i\sqrt{4\beta(1-\alpha\lambda) - (1+\beta)^2(1-\alpha\lambda_i)^2}\right].
\end{aligned}$$

The two roots are distinct complex numbers when $1-\alpha\lambda_i > 0$ and $(1+\beta)^2(1-\alpha\lambda_i) < 4\beta$, which happens when with our choice of α and β when $m < \lambda < L$. When $\lambda = L$, $\nu_{i,1} = \nu_{i,2} = 0$. When $\lambda = m$, $\nu_{i,1} = \nu_{i,2} = 1 - \sqrt{m/L}$.

It remains to bound the magnitude of the $\nu_{i,j}$ for when $\lambda_i \in (m, L)$.

The magnitude is given by

$$\frac{1}{2}\sqrt{(1+\beta)^2(1-\alpha\lambda_i)^2 + 4\beta(1-\alpha\lambda) - (1+\beta)^2(1-\alpha\lambda_i)^2} = \frac{1}{2}\sqrt{4\beta(1-\alpha\lambda)} = \sqrt{\beta}\sqrt{1-\frac{\lambda_i}{L}}$$

Note that

$$\begin{aligned}
\sqrt{\beta}\sqrt{1-\frac{\lambda_i}{L}} \le \sqrt{\beta}\sqrt{1-\frac{m}{L}} &= \left(\frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}} \cdot \frac{L-m}{L}\right)^{1/2} \\
&= \left(\frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}} \cdot \frac{(\sqrt{L}-\sqrt{m})(\sqrt{L}+\sqrt{m})}{L}\right)^{1/2} \\
&= \frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}} = 1 - \sqrt{m/L}\,.
\end{aligned}$$

which verifies our bound on the spectral radius.

■

Note that if the rate of convergence of the algorithm is given by $(1-\kappa^{-1/2})$, the number of iterations required to reach tolerance ϵ scales as $O(\sqrt{\kappa})$. This is a square root of the number of iterations required by the gradient method.

4 Weakly convex functions

When f is weakly convex, the parameter β is chosen slightly differently. We keep $\alpha = 1/L$ and set

$$\beta_k = \theta_k(\theta_{k-1}^{-1} - 1) \qquad \text{where } \theta_0 = 1,\ \ \theta_k = \frac{1}{2}\left[-\theta_{k-1}^2 + \sqrt{\theta_{k-1}^4 + 4\theta_{k-1}^2}\right].$$

This rather bizarre expression pops out of the proof of Nesterov's method. It is a bit magical, and even Nesterov refers to this as "an algebraic trick" to prove convergence. We will dive into a different derivation after we have discussed Mirror Descent.

In the meantime, let's at least describe the result. The main take away is that Nesterov's method satisfies:

$$f(x_k) - f(x_\star) \leq \tfrac{4L}{(k+2)^2}\|x_0 - x_\star\|^2$$

So, just as was the case for strongly convex f, the number of iterations required to achieve tolerance ϵ scales as $\epsilon^{-1/2}$, or the square root of the number of iterations required for the gradient method.

5 Conjugate Gradient Method

The problem with Nesterov's method as presented is that one is required to know the convexity parameters L and m to compute the appropriate step-sizes. The conjugate gradient method is a simple modification of Nesterov's method that does not require knowledge of these parameters. Consider the momentum iterations

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k y_k \\ y_k &= -\nabla f(x_k) + \beta_{k-1} y_{k-1} \end{aligned}$$

This is equivalent to our previous iteration after a change of variables. To choose α_k, we can pick the stepsize to minimize f along the direction y_k (i.e. $\min_{\alpha>0} f(x_k + \alpha y_k)$). For quadratics, if we take derivatives, this gives us

$$\alpha_k = \frac{y_k^T r_k}{y_k^T Q y_k} \qquad \text{where } r_k = Qx_k - p$$

Definition 2 *We say that two non-zero vectors u and v are conjugate (with respect to Q) if*

$$u^T Q v = 0\,.$$

We now pick β_k so that $\langle y_k, Qy_{k-1}\rangle = 0$ (i.e to make y_k and y_{k-1} conjugate).

$$\begin{aligned} \langle y_k, Qy_{k-1}\rangle &= \langle -r_{k-1} + \beta y_{k-1}, Qy_{k-1}\rangle \\ &= \langle -r_{k-1}, Qy_{k-1}\rangle + \beta_k \langle y_{k-1}, Qy_{k-1}\rangle \end{aligned}$$

$$\beta_k = \frac{\langle r_{k-1}, Qy_{k-1} \rangle}{\langle y_{k-1}, Qy_{k-1} \rangle}$$

Conjugacy guarantees that the directions y_k are orthogonal with respect to the inner product

$$\langle u, v \rangle_Q := u^T Q v \,.$$

Consequently, walking along conjugate directions using exact line search yields convergence to $x_\star$ in n steps.

Unfortunately, the conjugate gradient method doesn't admit particularly rigorous analysis when f is not quadratic.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

Feburary 3, 2015 Notes: Ashia Wilson and Benjamin Recht

Lecture 5: Lower Bounds

Our previous lectures have looked at the iteration complexity of first order methods. We saw that we could efficiently find approximate minimizers of convex functions in polynomial time, and provided guarantees on the rate of convergence in terms of parameters of the function to be minimized. These bounds held *for any* function in the class. So, for example, we discussed how the gradient method could be used to optimize a convex function satisfying

$$mI \preceq \nabla^2 f(x) \preceq LI$$

in a number of iterations proportional to L/m. Nesterov's method scales proportionally to $\sqrt{L/m}$. Is this the best we can do?

In this lecture we discuss lower bounds for a variety of function classes. In some sense, we show that Nesterov is "optimal" for algorithms that only evaluate function values and gradients of convex functions. In sharp contrast, we show that for non-convex functions, even verifying local optimality is intractable. So though we can find points with small gradients relatively quickly, having a small gradient tells us nothing about the local optimality of a nonconvex function. We show that the situation is only worse for Lipschitz continuous functions (i.e., the function is Lipschitz, but the gradient may not be). In this case, even finding a stationary point is intractable.

Before we proceed, I want to emphasize that all lower bounds are suspect. We are establishing the existence of one function on which restricted algorithms must take a long time to optimize. But these counter-examples are very fragile. One must wonder does this show a deficiency in modeling my function class or in my algorithm family? All lower-bounds must be taken with a grain of salt. We will emphasize the short comings of these particular bounds in our discussion.

1 Lower bounds for convex functions

Let's first consider the class of convex functions. We would like to minimize a convex function f. Assume throughout that f has L-Lipschitz gradients and f is strongly convex function with parameter m. We will also examine the case where $m = 0$.

We will restrict our attention to a particular class of algorithms. Our algorithms will be able to evaluate the value of f and its gradient at any point in $\mathbb{R}^d$. We will allow ourselves to initialize our search at any $x^{(0)} \in \mathbb{R}^d$. Note that because the set of convex functions is translation invariant, we can always assume that $x^{(0)} = 0$.

We additionally make the very stringent assumption

- **Restriction** *The iterate $x^{(k)}$ must be a linear combination of*

$$\{x^{(1)}, \ldots, x^{(k-1)}\} \qquad \text{and} \qquad \{\nabla f(x^{(1)}), \ldots, \nabla f(x^{(k-1)})\}\,.$$

That is, the next iterate is in the span of all the previous iterates and all the previous gradients.

Our goal will be to minimize the number of function evaluations and gradient calls to achieve

$$||x^{(N)} - x_\star|| < \epsilon$$

Interestingly, the worst-case instance we will construct for this case is a quadratic function. So even though the class of strongly convex functions is considerably richer than the space of quadratics, our bad case will be a quadratic.

Methods obeying our restriction are often called *Krylov methods* when f is quadratic. So what would our algorithm be able to do when applied to the function

$$f(x) = x^T Ax + b^T x$$

Recall from last lecture: the optimal solution is given by $Ax = b$. Let's expand out the iterates, starting at $x^{(0)} = 0$.

$$\begin{aligned}
x^{(0)} &= 0 \\
x^{(1)} &= x^{(0)} + t_1(Ax^{(0)} - b) = t_1 b \\
x^{(2)} &= t_1 b + t_2(Ax^{(1)} - b) = t_{21} Ab + t_{20} b \text{ (for some } t_{21} \text{ and } t_{20}) \\
x^{(3)} &= t_{32} A^2 b + t_{31} Ab + t_{30} b
\end{aligned}$$

Repeating this calculation we see that

$$x^{(k)} = \sum_{i=0}^{k-1} t_{k-i} A^i b$$

and hence x_k is in the span of $\{b, Ab, A^2 b, \ldots, A^{k-1} b\}$. One can now probably devise how to construct a bad instance. We need to find a positive definite matrix A and a vector b such that the distance of $A^{-1}b$ is as far away from the span of $\{b, Ab, A^2 b, \ldots, A^{k-1} b\}$ as possible.

A particular construction, due to Nesterov is

$$\begin{aligned}
f(x) &= (1/2) x^T Ax - b^T x \\
A &= \alpha A_0 + \beta I \\
b &= \alpha e_1
\end{aligned}$$

where

$$(A_0)_{ij} = \begin{cases} 2 & i = j \\ -1 & |i - j| = 1 \\ 0 & o.w. \end{cases}$$

Thiat is,

$$A_0 = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & 0 & \ldots & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \end{bmatrix}$$

$A_0 \succeq 0$ because it is diagonally dominant (in fact it is p.d., but just barely). We can also see this by noticing it is a sum of squares:

$$f(x) = \tfrac{1}{2}(\alpha+\beta)(x_1^2 + x_d^2) + (\alpha/2)\sum_{k=1}^{d}(x_k - x_{k+1})^2 + \frac{\beta}{2}\sum_{k=2}^{d=1} x_k^2 - \alpha x_1$$

The Lipschitz constant of the quadratic is equal to the maximum eigenvalue (i.e. the operator norm) of A. We begin by looking at A_0:

$$\begin{aligned} x^T A_0 x &= x_1^2 + \sum_{i=1}^{n}(x_i - x_{i+1})^2 + x_n^2 \\ &\leq x_1^2 + \sum_{i=1}^{n}(x_i^2 + x_{i+1}^2) + x_n^2 \\ &\leq 4||x||^2 \end{aligned}$$

Thus $||A_0|| \leq 4$. If we set $\alpha = \frac{L-m}{4}$ and $\beta = m$, we see that f has L-Lipschitz gradients and strong convexity parameter m.

For the remainder of the proof, we make the simplifying assumption that d is infinite. This will let us solve for the optimal solution in closed form. This argument can be truncated to finite d with some messy algebra. We leave this construction to the interested reader.

Writing out $Ax = b$ in coordinates, we see that the optimal x satisfies:

$$(2 + \frac{4}{\kappa - 1})x_1 - x_2 = 1 \tag{1}$$

$$-x^{(j+1)} + 2\left(\frac{\kappa+1}{\kappa-1}\right)x_j - x_{j+1} = 0 \tag{2}$$

Where $\kappa = \frac{L}{m}$ is the condition number L/m. This system has a simple ansatz. Indeed, the optimal x has coordinates $x_k = u^k$,where u^k satisfies

$$\begin{aligned} -u^{k-1} + 2\left(\frac{\kappa+1}{\kappa-1}\right)u^k - u^{k+1} \\ -u^{k-1}\left[1 - \frac{2(\kappa+1)}{\kappa-1}u + u^2\right] = 0 \end{aligned}$$

Solving the quadratic equation yields the solution

$$u = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right).$$

And the optimal solution is the sequence

$$x_\star = [u^k] = \left[\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k\right].$$

Now after k iterations

$$x^{(k)} \in span\Big\{\left[\frac{L-m}{4}A_0 + mI\right]^i e_1\Big\}_{i=0,\ldots,k-1}$$

In particular, we must have that $x_j^{(k)} = 0$ for all $j > k$. Thus, our error can be no better than the norm of the tail. That is, the sum over the remaining coordinates.

$$\begin{aligned}
||x^{(k)} - x_\star|| &\geq \sum_{j=k+1}^{\infty} u^{2j} \\
&= \frac{u^{2(k+1)}}{1-u^2} \\
&= u^{2(k+1)}||x_\star||^2 \\
&= u^{2(k+1)}||x_\star - x^{(0)}||^2
\end{aligned}$$

Here, the first equality sums the geometric series. The second follows by computing the ℓ_2 norm of $x_\star$. The final equality follows because we started at $x^{(0)} = 0$.

We can also bound the function value

$$\begin{aligned}
f(x^{(k)}) - f(x_\star) &\geq \frac{m}{2}||x^{(k)} - x_\star|| \\
&\geq \frac{m}{2}u^{2(k+1)}||x_\star - x^{(0)}||^2 \\
&= \frac{m}{2}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{2k+2}||x_\star - x^{(0)}||^2
\end{aligned}$$

This calculation confirms that no Krylov method can find an ϵ approximate solution in less than $\Theta(\sqrt{\kappa}\log(1/\epsilon))$ iterations.

Thus, we can say that Nesterov's algorithm is an "optimal method." Of course, there are many caveats here. First, note that this only holds for small iteration counts. If we could run for d steps, then conjugate gradient is a Krylov method that solves the problem exactly. Moreover, we can solve the system of equations $Ax = b$ in closed form in $O(d)$ time whenever A is tridiagonal. Even more importantly, it is not at all clear that Nesterov is the best algorithm for *all* strongly convex functions. It is optimal for this particular instance, but other algorithms might be more efficient on other functions—even on other quadratics.

1.1 Weakly convex f

A similar quadratic function shows that $1/k^2$ iterations are optimal for weakly-convex f. In this case,

$$f(x) = x^T\frac{L}{8}A_0x - \frac{L}{4}x^Te_1\,.$$

We can write this out in coordinates

$$\begin{cases}
2x_1 - x_2 = 1 \\
-x_{k-1} + 2_k - x_k = 0 \\
2x_d - x_{d-1} = 0
\end{cases}$$

Here we will run for $d/2$ iterations (which is also not fair... again, if we ran for d iterations, we would have solved the problem exactly with conjugate gradient). By forward substituting, we get

$$x_\star = [1 - \frac{j}{d+1}]$$

This means that

$$\begin{aligned} f(x_\star) &= \frac{L}{8} x_\star^T A x_\star - x_{opt}^{(1)} \\ &= \frac{L}{8}\left[1 - \frac{1}{d+1} - 2\left[1 - \frac{1}{d+1}\right]\right] \\ &= -\frac{L}{8}\left[1 - \frac{1}{d+1}\right] \end{aligned}$$

Moreover,

$$\begin{aligned} ||x_\star||^2 &= \sum_{i=1}^{d} \left(1 - \frac{i}{d+1}\right)^2 \\ &= d - \frac{2}{d+1}\sum_{i=1}^{d} i + \frac{1}{(d+1)^2}\sum_{i=1}^{d} i^2 \\ &= d - \frac{2(d \cdot d + 1)}{(d+1)\cdot 2} + \frac{1}{(d+1)^2} \cdot \frac{d(d+1)(2d+1)}{6} \\ &= d - d + \frac{d(2d+1)}{6(d+1)} \\ &\leq \frac{d}{3} \end{aligned}$$

These calculations let us establish the following

Theorem 1 *For any $1 \leq k \leq \frac{1}{2}(d-1)$, and any $x^{(0)} \in \mathbb{R}^1$, there exists an f with L-Lipschitz gradients such that for any first order method*

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

$$||x^{(k)} - x_\star||^2 \geq \frac{1}{8}||x^{(0)} - x_\star||^2$$

Proof Use

$$f_d := \frac{1}{8} x^T A_0 x - \frac{L}{4} e_1$$

with $d = 2k+1$. Note

$$f_{2k+1}(x_k) = f_k(x_k) \geq f_k(x_{opt}) = \frac{L}{8}\left(-1 + \frac{1}{k+1}\right)$$

Therefore

$$\begin{aligned} f(x_k) - f(x_\star) &\geq \frac{L}{8}\left(-1 + \frac{1}{k+1} + 1 - \frac{1}{2k+2}\right) \\ &= \frac{L}{8}\left(\frac{1}{2k+1}\right) \\ &\geq \frac{3}{8}\frac{1}{(2k+1)(2k+2)}||x_{opt} - x^{(0)}||^2 \\ &\geq \frac{3}{32}\frac{1}{(d+1)^2}||x_{opt} - x^{(0)}||^2 \end{aligned}$$

For the other inequality

$$\begin{aligned}
||x_k - x_\star||^2 &\geq \sum_{i=k+1}^{2k+1} \left(1 - \frac{i}{2k+2}\right)^2 \\
&= k + 1 - \frac{1}{k-1} \sum_{i=k+1}^{2k+1} i + \frac{1}{4(k+1)^2} \sum_{i=k+1}^{2k+1} i^2 \\
&\geq \frac{2k^2 + 7k + 6}{24(k+1)} \cdot \frac{3}{2k+1} ||x_{opt} - x^{(0)}||^2 \\
&\geq \frac{1}{8} ||x_{opt} - x^{(0)}||^2
\end{aligned}$$

■

2 Lower bounds for smooth functions

Let's now restrict our attention to differentiable, rather than convex functions. We know that we can efficiently find a solution with $\|\nabla f(x)\|^2 \leq \epsilon$ in $O(1/\epsilon^2)$ time. But what about a global minimum?

It turns out that even finding a *local* minimum is intractable. Consider the subset of homogenous quartics:

$$f(x) = \sum_{i,j=1}^{d} Q_{ij} x_i^2 x_j^2$$

where Q is some arbitrary symmetric matrix. If Q is positive definite or nonnegative then 0 is obviously a global minimizer. What about for general Q.

First note that

$$\frac{\partial f}{\partial x_i} = 4Q_{ii}x_i^3 + 2\sum_{j \neq i} Q_{ij} x_i x_j^2$$

Thus, the gradient is necessarily 0 at $x = 0$. However,

$$\begin{aligned}
\frac{\partial^2 f}{\partial x_i^2} &= 12Q_{ii}x_i + 2\sum_{j \neq i} Q_{ij} x_j^2 \\
\frac{\partial^2 f}{\partial x_i \partial x_j} &= 4\sum_{j \neq i} Q_{ij} x_i x_j \qquad \text{for } i \neq j
\end{aligned}$$

Meaning the *Hessian* is also zero at zero. So we cannot conclude that x is a local minimum from Hessian information alone.

It turns out that the situation is quite dire, actually:

Theorem 2 (Murty and Kabadi, 1987) *Given an integer square matrix Q, deciding if there exists and u with non-negative entries satisfying $u^T Q u < 0$ is NP complete.*

The proof of this fact follows by a relatively straightforward reduction to the SUBSET-SUM problem. Beginning with the formulation of this problem as an integer program, Murty and Kabadi produce a quadratic function such that 0 is the minimizer over the positive orthant if and only if the subset sum problem is feasible.

Reformulating this problem in our language, note that there is a 1 to 1 correspondence between vectors with nonnegative entries and vectors whose entries are squares. Thus, deciding if there exists and u with non-negative entries satisfying $u^T Q u < 0$ is equivalent to deciding if there exits a vector x such that $f(x) = \sum_{i,j=1}^{d} Q_{ij} x_i^2 x_j^2 < 0$. Note that by homogeneity, x is a local optimum if and only if there does not exist any x with $f(x) < 0$. Indeed, if there was such an x, then $f(tx)$ would be negative for $t \neq 0$. Thus, deciding if 0 is a local minimum of f is *co-NP complete*.

Now there are two caveats here. First, the hardness depends on P not equalling $co-NP$. Most people think this is true, but it remains a conjecture. Second, just because a problem is NP-hard doesn't mean you shouldn't try to solve it! NP-hard means that there are very difficult instances out there, so one must be careful. But many NP-hard problems have very large sets of instances that are efficiently solvable. One just has to make sure that the problem you care about is in the solvable part of the complexity zoo.

3 Lower bounds for continuous functions

Continuous functions are even more difficult than smooth functions. Consider the class of functions f that are L-Lipschitz. That is,

$$|f(x) - f(y)| \leq L|x - y|$$

for all x and y. These functions are necessarily continuous, but not necessarily differentiable. Thus, all we can do is evaluate function values and hope for the best. It turns out that the best is rather bad:

Theorem 3 *Let $\mathcal{A}$ be any algorithm that can access function values of Lipschitz $f : [0,1]^d \to \mathbb{R}$. Then there is no method that can find a solution with $f(x) - f(x_\star) \leq \epsilon$ in less than $(L/2\epsilon)^d$ function calls.*

This theorem says that no matter what you try to do, you can't optimize a Lipschitz continuous function in time less than exponential in the dimension. This is a rather hard limit, it doesn't rest on algorithmic restrictions nor on resolving the $P = NP$ question. As we'll see in the proof, it's just too easy to hide a needle in the haystack of Lipschitz continuous functions.

Proof Consider our algorithm $\mathcal{A}$ applied first to the 0 function. This function is certainly L-Lipschitz. Let's suppose this returns a sequence of points $\mathcal{P} \subset [0,1]^d$ with $|\mathcal{P}| = N$. By a dimension counting argument, there exists a point $\hat{x} \in [0,1]^d$ such that

$$\mathcal{P} \cap \{x \ : \ \hat{x_i} - N^{1/d}/2 \leq x_i \leq \hat{x}_i + N^{1/d}/2\} = \emptyset$$

Let's make the horrible function

$$f(x) = \min\{0, L\|x - \hat{x}\|_\infty - \epsilon\}$$

This function is clearly Lipschitz continous. Moreover, it is zero outside of the box

$$\{x \ : \ \hat{x_i} - \frac{\epsilon}{L} \leq x_i \leq \hat{x}_i + \frac{\epsilon}{L}\}$$

Thus, our algorithm when applied to this function will return a point with $f(x) = 0$, and hence will not find an ϵ optimal solution if $N \leq (\frac{L}{2\epsilon})^d$. ■

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

Feburary 10, 2015 Notes: Ashia Wilson and Benjamin Recht

Lecture 6: Incremental and Stochastic Gradients

The stochastic gradient method is one of the most popular algorithms for contemporary data analysis and machine learning. It has a long history and has been "invented" several times by many different communities (under the names "least mean squares," "back propagation," "online learning," and the "randomized Kaczmarz method"). Most people attribute this algorithm to the initial work of Robbins and Monro from 1950. There, they were interested in efficient algorithms for computing random means.

In this lecture, we explore some of the properties and implementation details of the stochastic gradient method.

As has been the case, our goal is to minimize $f : \mathbb{R}^d \to \mathbb{R}$. Where the stochastic gradient method differs from the previous methods we have studied is the sort of information we can extract from the function f. Assume that rather than accessing $\nabla f(x)$, we can compute or acquire a random function $g(x)$ such that $\mathbb{E}[g(x)] = \nabla f(x)$. We pretend that g is the gradient and form the update

$$x_{k+1} = x_k - \alpha_k g\left(x_k\right)$$

The intuition behind this method should be clear: we are following a descent direction in expectation, so if we wait long enough, we should be able to get close to the optimal solution. However, it's not quite that simple. Note that if $\nabla f(x_\star) = 0$, then we may move away from $x_\star$ depending on the statistics of $g(x)$. The fact that $x_\star$ is no longer a fixed point complicates the analysis.

1 Noisy gradients

The simplest application of SGM is to the case when we observe noisy gradients. Assume at each iteration, we compute the gradient of a convex function f, but our computation is corrupted by random noise. That is

$$g(x) = \nabla f(x) + \omega$$

where ω is some noise process. The stochastic gradient analysis will allow us to devise a step-size protocol to guarantee convergence to the optimum of $f(x)$ in the presence of such noise.

2 The incremental gradient method

The incremental gradient method, also known as the perceptron or back-propagation, is one of the most common applications of the stochastic gradient method. In this case, we assume that f has the form

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)\,.$$

where n is a very large number. Computing a full gradient requires computation that scales as $O(n)$, but we would like an algorithm that is sublinear in n. The incremental gradient method proceeds by selecting an $i_k \in \{1, \ldots, n\}$ and computing

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k) \ .$$

That is, we choose one of the f_i and follow its negative gradient. By cycling through all of the functions, we hope to eventually find a minimizer of f.

This method as presented is not random, but randomness provides extra intuition into why it converges. Note that an unbiased estimate of the gradient of f can be formed by computing

$$g(x) = \nabla f_i(x)$$

where i is selected at random. That is, $\mathbb{E}[g(x)] = \nabla f(x)$. If we select the coordinates in random order, the incremental gradient method becomes a special case of thes stochastic gradient method.

Another reason to use randomness is that the analysis becomes greatly simplified. Though one can show that the incremental gradient method converges, its proof in the randomized case is considerably simpler. Moreover, and perhaps more importantly, the converge guarantees in the non-random case are substantially weaker than the rates in the random case.

2.1 Example: Classification and the Perceptron

As a specific example, let's consider the canonical machine learning problem of classification. We are provided pairs (x_i, y_i), with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ for $i = 1, \ldots n$. The goal is to find a vector $w \in \mathbb{R}^d$ such that

$$w^T x_i > 0 \text{ for } y_i = 1$$
$$w^T x_i < 0 \text{ for } y_i = -1$$

Such a w defines a half-space where we believe all of the positive examples lie on one side and the negative examples on the other.

A popular method was invented in the 50s and uses one example at a time. We initialize our half-space at some w_0. At iteration k, we choose one of our data points, (x_{i_k}, y_{i_k}) and update

$$w_{k+1} = (1 - \gamma) w_k + \eta \begin{cases} y_{i_k} x_{i_k} & y_{i_k} w_k^T x_{i_k} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The idea behind this iteration is that if we get the sign incorrect—that is, x_{i_k} lies in the wrong halfspace—then we adjust w_k in a direction to make $w_k^T x_{i_k}$ closer to the correct sign.

It turns out that this is an instance of stochastic gradient descent. A quick calculation will convince you that this procedure is applying the incremental gradient method to the cost function

$$\frac{1}{n} \sum_{i=1}^{n} \max\left(1 - y_i x_i^T w, 0\right) + \lambda \|w\|_2^2 \ . \tag{2}$$

In the update equation (1), we have chosen where $\gamma = \frac{\eta\lambda}{N}$ and η is the stepsize parameter. In machine learning, the stepsize is often referred to as the *learning rate*. The cost function (2) is often called *the support vector machine*. The perceptron algorithm is thus equivalent to "training" a support vector machine using the stochastic gradient method.

2.2 Empirical Risk Minimiation

In machine learning, the Support Vector Machine is one of many instances of the class of optimization problems called *Empirical Risk Minimization*. Many classification, regression, and decision tasks can be evaluated as expected values of error over the data generating distributions. The most common example is the Bayes risk. Given a data generating distribution $p(x, y)$, and a *loss function* $\ell(u, v)$ we define the Bayes risk as

$$R_B[f] := \mathbb{E}[\ell(f(x), y)] \,.$$

This is the expected loss of the decision rule $f(y)$ with respect to the probability distribution $p(x, y)$ that gives rise to the data. The ℓ function measures how much cost we pay for assigning the value $f(x)$ when the quantity to be estimated is y. The goal of many learning tasks is to choose the function f that minimizes the Bayes risk. For example, the Support Vector Machine uses a *hinge loss* that measures how far our prediction $w^T x$ is in to the correct half-space. In regression y is a target variate, and the loss measures the squared difference between $f(x)$ and y.

Often times, minimizing the Bayes risk is computationally intractable and depends strongly on knowing the likelihood and prior models for the data pairs (x, y). A popular alternative uses samples to provided an estimate for the true risk. The setup is as follows, suppose we have a process that generates independent, identically distributed samples $(x_1, y_1), \ldots, (x_N, y_N)$ from the joint distribution $p(x, y)$. Now, for these data points and a fixed decision rule $\hat{x}(y)$, we'd expect that *the empirical risk*

$$R_{\text{emp}}[f] := \frac{1}{N} \sum_{i=1}^{N} \ell(f(y_i), x_i)$$

is "close" to the true Bayes risk. Indeed, $R_{\text{emp}}[f]$ is a random variable equal to the sample mean of the loss function. It is immediate that if we take the expectation with respect to our samples that

$$\mathbb{E}[R_{\text{emp}}[f]] = R_B \,.$$

Now, given these samples, the empirical risk is no longer a function of the likelihood and prior models. Once we condition on the data, we have a simpler optimization problem: minimizing the empirical risk corresponds to finding the best function f that minimizes the average of the loss over our data.

The stochastic gradient method and ERM are intimately tied together. Rather than forming an empirical risk and minimizing this using a method like gradient descent, the SGM would sample a pair (x, y), and then move down the gradient of the loss with respect to the current estimate of f. At the end of the operation, we will have an approximate minimizer of the Bayes Risk. The perceptron algorithm is just a particular instance of this approach to machine learning.

3 Implementation Details

Before we turn to a rigorous analysis of the stochastic gradient method, it will be useful to get some background and insight into how to choose the stepsize parameters. We'll explore the possibilities through some illustrative examples.

3.1 Example: Computing a mean

Consider applying the stochastic gradient method to the function

$$\frac{1}{2n}\sum_{i=1}^{n}(x-\omega_i)^2$$

where ω_i are n fixed scalars. Note that the gradient of one of the increments is

$$\nabla f_i(x) = x - \omega_i \,.$$

Starting with $x_1 = 0$ and use the stepsize $\alpha_k = 1/k$. We can then observe that

$$\begin{aligned}
x_2 &= x_1 - x_1 + \omega_1 = \omega_1 \\
x_3 &= x_2 - \frac{1}{2}(x_2 - \omega_2) = \frac{1}{2}\omega_1 + \frac{1}{2}\omega_2 \\
x_4 &= x_3 - \frac{1}{3}(x_3 - \omega_3) = \frac{1}{3}\omega_1 + \frac{1}{3}\omega_2 + \frac{1}{3}\omega_3
\end{aligned}$$

Thus, we can quickly conclude by induction that

$$x_{k+1} = \left(\frac{k-1}{k}\right)x_k + \frac{1}{k}\omega_k = \frac{1}{k}\sum_{i=1}^{k}\omega_i \,.$$

The $1/k$ stepsize was the originally proposed stepsize by Robbins and Monro. This simple example justifies why: we can think of the stochastic gradient method as computing a running average. Another motivation for the $1/k$ stepsize is that the steps tend to zero, but the path length is infinite.

Note that our cost after n steps is equal to one half of the variance of the sequence $\{\omega_i\}$. In fact, suppose we run the stochastic gradient method on the function

$$f(x) = \tfrac{1}{2}\mathbb{E}[(x-\omega)^2]$$

where ω is some random variable with mean μ and variance σ^2. If we run for n steps with i.i.d. samples of ω at each iteration, the calculation above reveals that

$$x_k = \frac{1}{n}\sum_{i=1}^{n}\omega_i \,.$$

The associated cost is

$$f(x_k) = \tfrac{1}{2}\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\omega_i - \omega\right)^2\right] = \frac{1}{2n}\sigma^2 + \tfrac{1}{2}\sigma^2$$

Now, suppose we could just compute the minimizer exactly. In this case, expand $f(x)$ to find

$$f(x) = \tfrac{1}{2}\mathbb{E}[x^2 - 2\omega x + \omega^2] = \tfrac{1}{2}x^2 - 2\mu x + \tfrac{1}{2}\sigma^2 + \tfrac{1}{2}\mu^2 \,.$$

The minimizer is $x_\star = \mu$, and the cost is

$$f(x_\star) = \frac{1}{2}\sigma^2$$

So after n iterations, we have

$$f(x) - f(x_\star) = \frac{1}{2n}\sigma^2 \,.$$

This is the best we could have achieved using any estimator for $x_\star$ given the sequence ω. Interestingly, the "one-at-a-time" or recursive method finds as good a solution as one that considers all of the data together. This example, while trivial, also reveals a fundamental limitation of the stotchastic gradient method: we can't expect to generically get fast convergence rates. *Statistics* not computation, stand in the way of any method achieving linear convergence rates.

Now, one drawback of this example is there was no need for randomness at all. We just marched through the increments in order and converged after n steps to the global optimum. Indeed, if we had chosen a random set of increments, there would be no guarantee that we would have even seen all of the ω_i after n iterations. The next example demonstrates that randomization can dramatically speed up convergence on certain instances.

3.2 Example: The Kaczmarz method

So why do we need randomness? This can be seen by expanding a higher dimensional problem.

$$\min \frac{1}{2n} \sum_{i=1}^{n} \left(a_i^T x - b_i\right)^2$$

Assume that there exists an $x_\star$ such that $a_i^T x_\star = b_i$ for all i, and that $\|a_i\| = 1$. Running the stochastic gradient method with stepsize 1, we find the recursion

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - a_i \left(a_i^T x^{(k)} - b_i\right) \\ &= \left(I - a_i a_i^T\right)\left(x^{(k)} - x_\star\right) + x_\star \\ &= \prod_{i=1}^{k} \left(I - a_i a_i^T\right)\left(x^{(0)} - x_\star\right) + x_\star \end{aligned}$$

We see that we are doing a series of projections onto one dimensional subspaces. If two subsequent subspaces are very close to one another, then we don't make much progress in that particular iteration. Indeed, we can design a sequence of a_i such that we make almost no progress whatsoever.

In contrast, symmetrizing over the order of the product is necessary for noncommutative operators. The following example in fact provides deterministic without-replacement orderings that have exponentially larger norm than the with-replacement expectation. Let $\omega_n = \pi/n$. For $n \geq 3$, define the collection of vectors

$$a_{k;n} = \begin{bmatrix} \cos\left(k\omega_n\right) \\ \sin\left(k\omega_n\right) \end{bmatrix} . \tag{3}$$

Note that all of the $a_{k;n}$ have norm 1 and, for $1 \leq k < n$, $\langle a_{k;n}, a_{k+1;n} \rangle = \cos(\omega_n)$. The matrices $A_k := a_{k;n} a_{k;n}^T$ are all positive semidefinite for $1 \leq k \leq n$, and we have the identity

$$\frac{1}{n} \sum_{k=1}^{n} A_k = \tfrac{1}{2} I \,. \tag{4}$$

Any set of unit vectors satisfying (4) is called a *normalized tight frame*, and the vectors (5) form a *harmonic frame* due to their trigonometric origin.

If we used the stochastic gradient method for n steps, picking steps uniformly at random each time, then

$$\begin{aligned} \mathbb{E}\left[\prod_{i=1}^{n} \left(I - a_{k_i} a_{k_i}^T\right)\left(x^{0)} - x_\star\right)\right] &= \left(I - \frac{1}{n}\sum_{i=1}^{n} a_i a_i^T\right)^n \left(x^{(0)} - x_\star\right) \\ &= 2^{-n}\left(x^{(0)} - x_\star\right) . \end{aligned}$$

Thus, the stocastic gradient method converges *linearly* to the optimal solution. What is different in this example? First, the optimal solution $x_\star$ is a fixed point of both a gradient map *and* a stochastic gradient step. So we don't have to average out the noise that tries to move us away from the optimum.

But the stochastic sampling also contributes to the fast rate of convergence. What happens if we use a deterministic order? Define the vectors

$$\hat{a}_{k;n} = \begin{bmatrix} \sin(-k\omega_n) \\ \cos(-k\omega_n) \end{bmatrix} . \tag{5}$$

Note that

$$I - a_{k;n} a_{k;n}^T = \hat{a}_{k;n} \hat{a}_{k;n}^T$$

Hence, the product of the $I - A_i$ is given by

$$\prod_{i=1}^{k} A_i = \hat{a}_{k;n} \hat{a}_{1;n}^T \prod_{j=1}^{k-1} \langle \hat{a}_{j;n}, \hat{a}_{j+1;n} \rangle = \hat{a}_{k;n} \hat{a}_{1;n}^T \cos^{k-1}(\omega_n) \,,$$

and hence

$$\begin{aligned} \left\| \prod_{i=1}^{k} \left(I - a_i a_i^T\right)\left(x^{(0)} - x_\star\right) \right\| &= \cos^{n-1}(\omega_n) \left| \hat{a}_{1;n}^T \left(x^{(0)} - x_\star\right) \right| \\ &\leq \left(1 - \tfrac{1}{n}\right) \left| \hat{a}_{1;n}^T \left(x^{(0)} - x_\star\right) \right| . \end{aligned}$$

If we had selected $x^{(0)} = [0; 1]$ and $x_\star = 0$, we see that we will have made nearly no progress if we marched through our increments in deterministic order!

Therefore, the arithmetic mean is less than the (deterministic) geometric mean for all $n \geq 3$.

This is a case where picking at random can cause a huge improvement in the convergence rate compared to drawing vectors in some predefined order.

We note here that this particular example has a special name, *the randomized Kaczmarz method*. It is a simple case of the stochastic gradient method for solving least-squares. This method was analyzed by signal processing researchers, without knowledge of the long standing work on the stochastic gradient method.

3.3 Epochs

Another central concept in stochastic gradient methods is the notion of *epochs*. In an epoch, some number of iterations are run, and then a choice is made about whether to change the stepsize. A common strategy is to run with a constant step size for some fixed number of iterations T, and then reduce the stepsize by a constant factor γ. Thus, if our initial stepsize is α, on the kth epoch, the stepsize is $\alpha\gamma^{k-1}$. This method is often more robust in practice than the diminishing stepsize rule. For this stepsize rule, a reasonable heuristic is to choose γ between 0.8 and 0.9.

Another rule which we will study is called *epoch doubling*. In epoch doubling, we run for T steps with stepsize α, then run $2T$ steps with stepsize $\alpha/2$, and then $4T$ steps with stepsize $\alpha/4$ and so on. Note that this provides a piecewise constant approximation to the function α/k.

3.4 Momentum

Finally, we note that one can run stochastic gradient descent with momentum. This method would be identical to the momentum methods studied earlier in the course, with the gradient replaced by a stochastic gradient. That is:

$$x_{k+1} = x_k - \alpha_k g\left(x_k\right) + \beta(x_k + x_{k-1})\,.$$

In practice, these methods are very successful. Typical choices for β here are between 0.8 and 0.95. The theoretical guarantees for momentum methods only demonstrate meager gains over the standard SGM. Essentially, we know that the function value will converge at a rate of $1/k$, but the constant in front of the $1/k$ can be made smaller using momentum or acceleration. Regardless of the theoretical guarantees, one should always keep in mind that momentum can provide significant accelerations, and should be considered an option in any implementation of SGM.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

Feburary 12, 2015 Notes: Benjamin Recht

Lecture 7: Analysis of the Stochastic Gradient Method

We now turn to an analysis of the stochastic gradient method. Before we proceed, let's set up some conventions. We will assume that we are trying to minimize a convex function $f : \mathbb{R}^d \to \mathbb{R}$. Let $x_\star$ denote any optimal solution of f. We will assume we gain access at every iteration to a *stochastic function* $g(x;\xi)$ such that

$$\mathbb{E}_\xi[g(x;\xi)] = \nabla f(x)\,.$$

Here ξ is a random variable which determines what our direction looks like. We additionally assume that there exist non-negative constants L_g and B such that

$$\mathbb{E}_\xi\left[\|g(x)\|_2^2\right] \leq L_g^2\|x - x_\star\|^2 + B^2\,.$$

Note that this makes no assumption about the boundedness of g, only that it is bounded in expectation.

We will study the stochastic gradient iteration

$$x_{k+1} = x_k - \alpha_k g\left(x_k;\xi_k\right)$$

for various choices of stepsizes under different assumptions about L_g and B. Throughout, we will assume that the sequence $\{\xi_j\}$ is selected i.i.d. from some fixed distribution. One can weaken the assumptions of both independence and identical distribution, but doing so will complicate and clutter the analysis.

1 The constants L_g and B

Let's briefly discuss how the constants L_g and B manifest themselves in different problem settings.

1.1 Case 1: Bounded gradients ($L_g = 0$ and $B > 0$.)

In this case, we are asserting that the stochastic gradient function is bounded almost surely for all x. Some examples of this would be the support vector machine where

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \max(1 - x^T(z_i y_i), 0)$$

In this case, for the incremental gradient method, ξ is a random index between 1 and n and

$$g(x,i) = \begin{cases} z_i y_i & x^T(z_i y_i) < 1 \\ 0 & \text{otherwise} \end{cases}$$

and hence,

$$B = \sup_i \|z_i\|_2 \,.$$

(note that this function is technically not differentiable. But any smoothing of the hinge around zero, say by convolution with a gaussian, will still result in a problem with bounded gradients)

Note that under this assumption, f cannot be strongly convex. This is because if f is a strongly convex function with parameter m, we have

$$\|\nabla f(x)\| \geq \frac{m}{2}\|x - x_\star\|$$

for all x. But, by Jensen's inequality,

$$\|\nabla f(x)\|^2 \leq \mathbb{E}[\|g(x;\xi)\|^2]$$

which implies that the norm of the stochastic gradients cannot be bounded above.

1.2 Case 2: The randomized Kaczmarz method ($B = 0$, $L_g > 0$)

In the randomized Kaczmarz method, we assume that

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n}(a_i^T x - b_i)^2 \,,$$

and there exists a point $x_\star$ such that $a_i^T x_\star = b_i$ for all i. In this case

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n}(x - x_\star)^T a_i a_i^T (x - x_\star)$$

and

$$g(x;i) = a_i a_i^T (x - x_\star)$$

Computing the expected norm of the stochastic gradient yields

$$\mathbb{E}[\|g(x;i)\|^2] = \mathbb{E}[\|a_i\|^2 |a_i^T(x - x_\star)|^2] \leq \mathbb{E}[\|a_i\|^4]\|x - x_\star\|^2$$

Hence, we can use the bounds

$$L_g \leq \mathbb{E}[\|a_i\|_2^4]^{1/2} \quad \text{and} \quad B = 0 \,.$$

1.3 Case 3: additive Gaussian noise

Suppose that

$$g(x;\omega) = \nabla f(x) + \omega$$

where ω is a Gaussian random vector with mean 0 and covariance $\sigma^2 I$. In this case,

$$\mathbb{E}[g(x;\omega)] = \nabla f(x)$$

and

$$\mathbb{E}[\|g(x;\omega)\|^2] = \|\nabla f(x)\|^2 + \sigma^2 d$$

Thus, L_g is upper bounded by the Lipschitz constant of the gradient of f. We additionally have that $B \leq \sigma\sqrt{d}$.

1.4 Case 4: The incremental gradient method

For the general incremental gradient method, we have

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

and

$$g(x;i) = \nabla f_i(x)\,.$$

Assume that $\nabla f_i(x)$ has Lipschitz constant L_i. Let $x_\star^{(i)}$ denote any point where $\nabla f_i(x_\star^{(i)}) = 0$. Then we can compute

$$\begin{aligned}
\mathbb{E}[\|g(x;i)\|^2] &= \mathbb{E}[\|\nabla f_i(x)\|^2] \\
&\leq \mathbb{E}[L_i^2\|x - x_\star^{(i)}\|^2] \\
&\leq \mathbb{E}\left[2L_i^2\|x - x_\star\|^2 + 2L_i^2\|x_\star^{(i)} - x_\star\|^2\right] \\
&= \left(\frac{2}{n}\sum_{i=1}^{n} L_i^2\right)\|x - x_\star\|^2 + \frac{2}{n}\sum_{i=1}^{n} L_i^2\|x_\star^{(i)} - x_\star\|^2\,.
\end{aligned}$$

Thus, we can identify

$$L_g \leq \left(\frac{2}{n}\sum_{i=1}^{n} L_i^2\right)^{1/2} \quad \text{and} \quad B \leq \left(\frac{2}{n}\sum_{i=1}^{n} L_i^2\|x_\star^{(i)} - x_\star\|^2\right)^{1/2}\,.$$

There is a nice intuition of this parameter B. If all of the minima of f_i coincide with $x_\star$, then B will equal zero. We will see in the next section that this means that the stochastic gradient method would converge at a linear rate. When B is nonzero, we will only be able to prove convergence at a rate inversely proportional to the iteration counter. However, the smaller B, the faster the convergence.

2 Analysis

The analysis for the different settings of L_g and B are different, but they all start from the same point. We begin by expanding the distance to the optimal solution

$$\begin{aligned}
\|x_{k+1} - x_\star\|^2 &= \|x_k - \alpha_k g_k(x_k;\xi_k) - x_\star\|^2 \\
&= \|x_k - x_\star\|^2 - 2\alpha_k\langle g_k(x_k;\xi_k), x_k - x_\star\rangle + \alpha_k^2\|g_k(x_k;\xi_k)\|^2
\end{aligned}$$

We deal with each term in this expansion separately. First note that if we apply the law of iterated expectation

$$\begin{aligned}
\mathbb{E}[\langle g_k(x_k;\xi_k), x_k - x_\star\rangle] &= \mathbb{E}\left[\mathbb{E}_{\xi_k}[\langle g_k(x_k;\xi_k), x_k - x_\star\rangle \mid \xi_0,\ldots,\xi_{k-1}]\right] \\
&= \mathbb{E}\left[\langle \mathbb{E}_{\xi_k}[g_k(x_k;\xi_k) \mid \xi_0,\ldots,\xi_{k-1}], x_k - x_\star\rangle\right] \\
&= \mathbb{E}\left[\langle \nabla f(x_k), x_k - x_\star\rangle\right]\,.
\end{aligned}$$

Here, we are simply using the fact that ξ_k being independent of all of the preceding ξ_i implies that it is independent of x_k. This means that when we iterate the expectation, the stochastic gradient can be replaced by the gradient.

By a similar argument, we can bound the last term as

$$\mathbb{E}[\|g(x_k;\xi_k)\|_2^2] = \mathbb{E}\left[\mathbb{E}_{\xi_k}[\|g(x_k;\xi_k)\|_2^2 \mid \xi_0,\ldots,\xi_{k-1}]\right] \leq \mathbb{E}\left[L_g^2\|x_k - x_\star\|_2^2 + B^2\right]$$

Letting $a_k := \mathbb{E}[\|x_k - x_\star\|^2]$, this gives

$$a_{k+1} \leq (1+\alpha_k^2 L_g^2)a_k - 2\alpha_k \mathbb{E}\left[\langle \nabla f(x_k), x_k - x_\star\rangle\right] + \alpha_k^2 B^2\,. \tag{1}$$

All of our analyses follow from different manipulations of (1). We'll proceed through several cases.

2.1 Case 1: $L_g = 0$.

Let's run the algorithm on a convex f with $L_g = 0$. Let α_k be our stepsize sequence. Define

$$\lambda_k = \sum_{j=0}^{k} \alpha_j\,.$$

This is the sum of all the stepsizes up to iteration k. Also define

$$\bar{x}_k = \lambda_k^{-1} \sum_{j=0}^{k} \alpha_j x_j\,.$$

$\bar{x}_k$ is the average of the iterates weighted by the stepsize. We are going to analyze the deviation of $f(\bar{x}_k)$ from optimality.

Also let $D_0 = \|x_0 - x_\star\|^2$. D_0 is the initial distance to an optimal solution. It is not necessarily a random variable.

To proceed, we just expand the following expression:

$$\begin{aligned}
\mathbb{E}\left[f\left(\bar{x}_T\right) - f(x_\star)\right] &\leq \mathbb{E}\left[\lambda_T^{-1}\sum_{t=0}^{T}\alpha_t(f(x_t) - f(x_\star))\right] && \text{(2a)}\\
&\leq \lambda_T^{-1}\sum_{t=0}^{T}\alpha_t\mathbb{E}[\langle\nabla f(x_t), x_\star - x_t\rangle] && \text{(2b)}\\
&\leq \lambda_T^{-1}\sum_{t=0}^{T}\tfrac{1}{2}(a_t - a_{t+1}) + \tfrac{1}{2}\alpha_t^2 B^2 && \text{(2c)}\\
&= \frac{a_0 - a_{T+1} + B^2\sum_{t=0}^{T}\alpha_t^2}{2\lambda_T} && \text{(2d)}\\
&\leq \frac{D_0^2 + B^2\sum_{t=0}^{T}\alpha_t^2}{2\sum_{t=0}^{T}\alpha_t} && \text{(2e)}
\end{aligned}$$

Here, (2a) follows because f is convex, (2b) is the first order condition for convexity, and (2c) uses (1).

With this in hand, we can now easily prove the following

Proposition 1 (Nemirovski *et al* [?]) *Suppose we run the SGM on a convex f with $L_g = 0$ for T steps with stepsize α. Define*

$$\alpha_{\text{opt}} = \frac{D_0}{B\sqrt{T}}$$

and

$$\theta := \frac{\alpha}{\alpha_{\text{opt}}}$$

Then we have the bound

$$\mathbb{E}[f(\bar{x}) - f_\star] \leq \left(\tfrac{1}{2}\theta + \tfrac{1}{2}\theta^{-1}\right) \frac{BD_0}{\sqrt{T}}\,. \tag{3}$$

This proposition asserts that we pay linearly for errors in selecting the optimal constant stepsize. If we guess a constant stepsize that is two-times or one-half of the optimal choice, then we need to run for twice as many iterations.

We can prove this just by minimizing (2e). Plugging in our stepsize, we see that

$$\mathbb{E}\left[f\left(\bar{x}_T\right) - f(x_\star)\right] \leq \frac{D_0^2 + B^2 T\alpha^2}{2T\alpha} = \left(\tfrac{1}{2}\theta^{-1} + \tfrac{1}{2}\theta\right) \frac{BD_0}{\sqrt{T}}\,.$$

Other stepsizes could also be selected here, including diminishing stepsizes. But the constant stepsize, it turns out, is optimal for this upper bound.

2.2 Case 2: $B = 0$

When $B = 0$, we surprisingly get a *linear rate* of convergence provided that f is strongly convex. Assume f has strong convexity parameter m. Then we have the inequality

$$\langle \nabla f(x), x - x_\star \rangle \geq m\|x - x_\star\|^2\,.$$

Plugging this into (1) with $B = 0$ gives the recursion

$$a_{k+1} \leq (1 - 2m\alpha_k + L_g^2\alpha_k^2)a_k\,. \tag{4}$$

Choosing any constant α in the range $(0, 2m/L_g^2)$ gives a linear rate of convergence. The optimal rate is when $\alpha = \frac{m}{L_g^2}$, in which case we have

$$a_k \leq \left(1 - \frac{m^2}{L_g^2}\right)^k D_0\,,$$

and

$$T = \frac{L_g^2}{m^2}\log\left(\frac{D_0}{\epsilon}\right)$$

suffice to guarantee that $\mathbb{E}[\|x_T - x_\star\|^2] \leq \epsilon$.

2.3 Case 3: B and L_g both nonzero

We finally turn to analyzing the general form of the recursion (1) when f is strongly convex:

$$a_{k+1} \leq (1 - 2m\alpha_k + \alpha_k^2 L_g^2)a_k + \alpha_k^2 B^2$$

Constant stepsize First, consider the case of a constant stepsize. Assuming that $\alpha \in (0, \frac{m}{L_g^2})$, we can roll the recursion out to find

$$a_k \leq (1 - 2m\alpha + \alpha^2 L_g^2)^k D_0 + \frac{\alpha B^2}{2m - \alpha L_g^2}\,.$$

What we see is that no matter how long we run, we can't prove that a_k converges to 0. Indeed, even "at infinity" (taking the limit as k tends to infinity), we see that

$$\lim_{k\to\infty} a_k \leq \frac{\alpha B^2}{2m - \alpha L_g^2}$$

This is actually representative of what happens in practice. The iterates converge to a ball around the optimal solution, but bounce around inside of this ball. To get closer to the optimal solution, we can reduce the stepsize α. But then, the rate at which we converge to the optimal ball is controlled by the quantity $1 - 2m\alpha + \alpha^2 L_g^2$. This number tends to 1 as α tends to zero.

One way to balance these two effects is to use *epochs*. The idea is to run with an agressively large stepsize for T iterations. Then, we halve the stepsize and double the number of iterations. This has the effect of shrinking the radius about the optimal solution, and guarantees we get close to this radius by running for an extended period of time.

Diminishing stepsize Note that by running in epochs, we are making a piecewise approximation to the stepsize C/k. We can use such a stepsize and get direct convergence to the optimum. Suppose we choose the stepsize

$$\alpha_k = \frac{\gamma}{k_0 + k}$$

for some constants γ and k_0. We will now show that for certain choices of these constants, we can guarantee that

$$a_k \leq \frac{Q}{k_0 + k}\,.$$

One can prove by induction the following

Proposition 2 *Suppose f is strongly convex with parameter m. If we run the stochastic gradient method with stepsize*

$$\alpha_k = \frac{1}{2m(L_g^2/2m^2 + k)}$$

then

$$\mathbb{E}[\|x_k - x_\star\|^2] \leq \frac{B^2}{2m(L_g^2/2m^2 + k)}$$

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

February 23, 2016 Notes: Benjamin Recht

Lecture 9: Constrained Optimization and the Projected gradient methods

In constrained optimization, we aim to find a point x which achieves the smallest value of some function f subject to the requirement that x lives in some specified set Ω.

Constrained optimization lets us design considerably more rich and complex optimization problems. The constraints could simply be bounds on the values of the variables, but could model temporal dependencies, resource constraints, or statistical models. In this first introductory lecture, we will focus on case when Ω is a simple convex set. We will move to more complicated scenarios in the coming weeks.

1 The Minimum Principle

What does it mean for $x_\star$ to minimize f over a set Ω? We say $x_\star$ minimizes f over Ω if $x_\star \in \Omega$ and

$$f(x_\star) \leq f(z) \quad \forall\, z \in \Omega\,.$$

That is, $x_\star$ is an element of Ω and attains the lowest function value. We say that $x_\star$ is a *local minimizer* if $x_\star \in \Omega$ and there is a convex neighborhood $\mathcal{N}$ of $x_\star$ such that

$$f(x_\star) \leq f(z) \quad \forall\, z \in \mathcal{N} \cap \Omega\,.$$

These definitions seem reasonable enough, but now how do we check if we have a minimizer or local minimizer? When $\Omega = \mathbb{R}^d$, we could simply check if $\nabla f(x) = -$. But with constraints, this is not the case. For example, suppose we want to minimize x^2 subject to $x \in [1, 2]$. Then clearly the minimum is $x = 1$, but the gradient at 1 is 2. The following proposition provides a solution.

Proposition 1 (The Minimum Principle) *If f is smooth and $x_\star$ locally minimizes f on a closed convex set Ω, then $\langle \nabla f(x_\star), z - x_\star \rangle \geq 0$ for all $z \in \Omega$. If f is convex, then the converse holds.*

Proof $\langle \nabla f(x_\star), z - x_\star \rangle$ is the directional derivative of f in the direction $z - x_\star$. If this directional derivative is negative, then we can move from $x_\star$ to z along the line segment connecting them and decrease the function with a sufficiently small step. That is, there exists a $t > 0$ such that $f((1-t)x_\star + tz) < f(x_\star)$, But since Ω (and also $\mathcal{N} \cap \Omega$) is convex, $(1-t)x_\star + tz \in \Omega$ contradicting the optimality of $x_\star$.

For the converse, assume f is convex and $\langle \nabla f(x_\star), z - x_\star \rangle \geq 0$ for all $z \in \Omega$. Then for $z \in \Omega$

$$f(z) \geq f(x_\star) + \langle \nabla f(x_\star), z - x_\star \rangle \geq f(x_\star)$$

proving $x_\star$ is optimal. ■

Note that this proposition proves that 1 minimizes x^2 over $[1, 2]$ because we have that $z - 1 > 0$ for all $z \in [1, 2]$.

In the case that f is strongly convex, we can use the Minimum Principle to prove that f will have a unique minimizer over Ω.

Proposition 2 *Suppose f is strongly convex. Then f has a unique minimizer over the closed convex set Ω.*

Proof Let x_1 and x_2 minimize f over Ω. Suppose $x_1 \neq x_2$, then we have

$$f(x_2) \geq f(x_1) + \langle \nabla f(x_1), x_2 - x_\star \rangle + \tfrac{m}{2}\|x_1 - x_2\|^2 > f(x_1)$$

which contradicts the optimality of x_2. Thus, x_1 must equal x_2. ■

While the Minimum Principle looks like a difficult condition to check, we will discuss an algorithm below that will find an $x_\star$ approximately satisfying this condition.

2 Euclidean Projection

Let Ω be a closed, convex set. The *Euclidean projection* of a point x onto Ω is the closest point in Ω to x. Denote this point by $\Pi_\Omega(x)$. Note that $\Pi_\Omega(x)$ is the solution of a constrained optimization problem:

$$\Pi_\Omega(x) = \arg\min\{\|z - x\| \ : \ z \in \Omega\}$$

That is, $\Pi_\Omega(x)$ is the solution to the optimization problem

$$\begin{array}{ll} \text{minimize}_z & \frac{1}{2}\|z - x\|^2 \\ \text{subject to} & z \in \Omega \end{array} .$$

Since the cost function of this problem is strongly convex, this proves that $\Pi_\Omega(x)$ is unique for all x.

Using the minimum principle, we can compute a variety of projections onto simple sets.

Example 1: The nonnegative orthant The nonnegative orthant is the set of vectors which are nonnegative in all coordinates.

$$\Omega = \{x \ : \ x_i \geq 0 \ \forall \ i = 1, \ldots, d\}$$

Note that Ω is a closed, convex cone.

Unpacking the condition $\langle \Pi_\Omega(x) - x, z - \Pi_\Omega(x) \rangle \geq 0$, we must have that

$$[\Pi_\Omega(x) - x]_i \geq 0$$

for all coordinates. Note that simply setting

$$[\Pi_\Omega(x)]_i = \begin{cases} x_i & x_i \geq 0 \\ 0 & x_i < 0 \end{cases}$$

satisfies the Minimum Principle.

Example 2: Unit norm ball Let

$$\Omega = \{x \ : \ \|x\| \leq 1\}$$

To compute $\Pi_\Omega(x)$, note that we require $\langle \Pi_\Omega(x) - x, z - \Pi_\Omega(x) \rangle \geq 0$. for all $z \in \Omega$. One can readily check that

$$\Pi_\Omega(x) = \frac{x}{\|x\|}$$

satisfies the Minimum Principle.

One of the most useful properties of the Euclidean projection is the fact that projections are *nonexpansive* in the following sense:

Proposition 3 *Let Ω be a closed convex set. Then*

$$\|\Pi_\Omega(x) - \Pi_\Omega(y)\| \leq \|x - y\|$$

for all $x, y \in \mathbb{R}^d$.

Proof By the minimum principle, $\Pi_\Omega(x)$ satisfies

$$\langle \Pi_\Omega(x) - x, z - \Pi_\Omega(x) \rangle \geq 0$$

for all $z \in \Omega$. Now we can write out

$$\begin{aligned}
\|x - y\|^2 &= \|(x - \Pi_\Omega(x)) - (y - \Pi_\Omega(y)) + \Pi_\Omega(x) - \Pi_\Omega(y)\|^2 \\
&= \|(x - \Pi_\Omega(x)) - (y - \Pi_\Omega(y))\|^2 + \|\Pi_\Omega(x) - \Pi_\Omega(y)\|^2 \\
&\qquad + 2\langle \Pi_\Omega(x) - x, \Pi_\Omega(y) - \Pi_\Omega(x) \rangle + 2\langle \Pi_\Omega(y) - y, \Pi_\Omega(x) - \Pi_\Omega(y) \rangle \\
&\geq \|(x - \Pi_\Omega(x)) - (y - \Pi_\Omega(y))\|^2 + \|\Pi_\Omega(x) - \Pi_\Omega(y)\|^2
\end{aligned}$$

completing the proof. ∎

3 The projected gradient algorithm

The projected gradient algorithm combines a projection step with a gradient step. This lets us solve a variety of constrained optimization problems with simple constraints.

We will aim to solve the constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \end{array} \tag{1}$$

where f is smooth and Ω is convex. Let us assume as usual that ∇f is Lipschitz so that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

Let us define a projected gradient scheme to solve this problem. Let $\alpha_0, \ldots, \alpha_T, \ldots$, be a sequence of positive step sizes. Choose $x_0 \in X$, and iterate

$$x_{k+1} = \Pi_\Omega(x_k - \alpha_k \nabla f(x_k)) \,. \tag{2}$$

The algorithm simply alternates between taking gradient steps and then taking projection steps.

The key idea behind this algorithm is summed up by the following proposition

Proposition 4 *Let f be differentiable and convex and let Ω be convex. $x_\star$ is an optimal solution of (1) if and only if $x_\star = \Pi_\Omega(x_\star - \alpha \nabla f(x_\star))$ for all $\alpha > 0$.*

Proof $x_\star$ is an optimal solution if and only if $\langle \nabla f(x_\star), x - x_\star \rangle \geq 0$ for all $x \in \Omega$. This is equivalent to

$$\langle x_\star - (x_\star - \alpha \nabla f(x_\star)), x - x_\star \rangle \geq 0 \,,$$

which, by the Minimum Principle is equivalent to $x_\star$ being the optimal solution of

$$\begin{array}{ll} \text{minimize}_z & \frac{1}{2}\|z - x_\star\|^2 \\ \text{subject to} & z \in \Omega \end{array}$$

in other words, $x_\star = \Pi_\Omega(x_\star - \alpha \nabla f(x_\star))$. ■

For non-convex f, we see that a fixed point of the projected gradient iteration is a stationary point of h. We first analyze the convergence of this projected gradient method for arbitrary smooth f, and then focus on strongly convex f.

3.1 General Case

Let f_* denote the optimal value of (1). Suppose we set $\alpha_k = 1/M$ for all k with $M \geq L$. Then we have

$$\|x_{k+1} - x_k\| \leq \sqrt{\frac{2(f(x_0) - f_\star)}{M(k+1)}}\,. \tag{3}$$

This expression confirms that we will find a point x where

$$\|\Pi_\Omega(x - \alpha \nabla f(x)) - x\| \leq \epsilon\,.$$

To verify this inequality, note that for any x, y,

$$f(x) = f(x) \leq f(y) + \nabla f(y)^T(x - y) + \frac{M}{2}\|x - y\|^2 =: \ell(x; y)$$

for any $M \geq L$. This is just Taylor's series. Note that the minimizer of $\ell(x; y)$ (with respect to x) over Ω is equal to

$$\Pi_\Omega(y - 1/M \nabla f(y))\,.$$

and also note that $\ell(x; y)$ is strongly convex with parameter M.

Now we have the chain of inequalities

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq f(x_k) - \ell(x_{k+1}; x_k) \\ &= \ell(x_k; x_k) - \ell(x_{k+1}; x_k) \\ &\geq \frac{M}{2}\|x_{k+1} - x_k\|^2 \end{aligned}$$

Summing these inequalities up for $k = 1, \ldots, n$, we have

$$\sum_{k=0}^{n} \|x_{k+1} - x_k\|^2 \leq \frac{2}{M}(f(x_0) - f_\star)$$

and the conclusion follows.

3.2 Strongly Convex Case

Let's now assume that f is strongly convex with strong convexity parameter m:

$$f(z) \geq f(x) + \nabla f(x)^T (z - x) + \frac{m}{2}\|z - x\|^2 . \tag{4}$$

Let $x_\star$ denote the optimal solution of (1). $x_\star$ is unique because of strong convexity. Observe that

$$\begin{aligned} \|x_{k+1} - x_\star\| &= \|\Pi_\Omega(x_k - \alpha_k \nabla f(x_k) - \Pi_\Omega(x_\star - \alpha_k \nabla f(x_\star))\| &(5)\\ &\leq \|x_k - \alpha_k \nabla f(x_k) - x_\star + \alpha_k \nabla f(x_\star)\| &(6)\end{aligned}$$

Here, the first equality follows by the definition of x_{k+1} and because $x_\star$ is optimal (see Proposition 4). (6) follows from Proposition 3.

Since f is strongly convex and has a Lipschitz continuous gradient, it follows that for all vectors x and y and all positive scalars η

$$\|x - \eta \nabla f(x) - (y - \eta \nabla f(y))\| \leq \max\{|1 - \eta L|, |1 - \eta m|\}\|x - y\| . \tag{7}$$

To see this, note that there exists a $\hat{t} \in [0, 1]$ such that

$$\|x - \eta \nabla f(x) - (y - \eta \nabla f(y))\| = \left\| \left(I - \eta \nabla^2 f(x + \hat{t}(y - x))\right)(y - x) dt \right\| . \tag{8}$$

From this, it follows that

$$\|x - \eta \nabla f(x) - (y - \eta \nabla f(y))\| \leq \sup_z \|I - \eta \nabla^2 f(z)\| \|y - z\| . \tag{9}$$

Note that the minimum eigenvalue of $\nabla^2 f(z)$ is at least m and the maximum eigenvalue is at least L. Therefore the eigenvalues of $I - \eta \nabla^2 f(z)$ are at most $\max(1 - \eta L, 1 - \eta m)$ and at least $\min(1 - \eta L, 1 - \eta m)$. Therefore, $\|I - \eta \nabla^2 f(z)\| \leq \max(|1 - \eta L|, |1 - \eta m|)$.

In particular, using this upper bound in (6), we have

$$\|x_{k+1} - x_\star\| \leq \max\{|1 - \alpha_k L|, |1 - \alpha_k m|\}\|x - y\| . \tag{10}$$

Note that $\alpha_k = \frac{2}{L+m}$ minimizes the right hand side for all k. Setting α_k to this value, we find that

$$\|x_{k+1} - x_\star\| \leq \left(\frac{L - m}{L + m}\right)\|x_k - x_\star\| \tag{11}$$

or, denoting $\kappa = \frac{L}{m}$ and $D_0 = \|x_0 - x_\star\|$,

$$\|x_k - x_\star\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k D_0 \tag{12}$$

That is, for strongly convex f and arbitrary Ω, the projected gradient algorithm converges at a linear rate under a constant step-size policy.

3.3 Nesterov Iteration, no proof

We can even define an *accelerated* version of the proximal gradient method. Iterations take the form:

$$\begin{aligned} \xi_{k+1} &= \Pi_\Omega \left(y_k - \alpha \nabla f(y_k) \right) \\ y_k &= \xi_k + \beta(\xi_k - \xi_{k-1}) \end{aligned} \tag{13}$$

Note that when $\Pi_\Omega = I$, we recover the standard Nesterov algorithm. When $\beta = 0$, we recover the proximal gradient method. This method will converge in

$$O\left(\sqrt{\frac{L}{m}} \log(1/\epsilon) \right)$$

iterations for strongly convex functions. We will prove this when we return to Nesterov's method after the midterm.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

Feburary 24, 2015 Notes: Benjamin Recht

Lecture 10: Subgradients

Up until now, we have been assuming that our convex functions are everywhere differentiable. Indeed, we have been a bit cavalier with this fiat. Consider the SVM loss function

$$f(x) = \max(1 - x, 0)$$

This function is differentiable everywhere except at $x = 1$. But what do we do at 1 if we want to compute a derivative. 1 is clearly a minimizer of f, but what general rule can prove such a thing. And if we had the function

$$f(x) = \max(-2x, -x, x - 2)$$

then we have a function that is not differentiable at $x = \{0, 1, 2\}$.

We can obviously create a long list of interesting functions to optimize that are convex but fail to be differentiable. *Every norm* is not differentiable at $x = 0$. More exotically, the maximum eigenvalue of a symmetric matrix is convex, but not differentiable (consider the example where the maximum eigenvalue has multiplicity greater than 1).

In this lecture we turn to analyzing such non-smooth convex functions. In the next lecture we will show that they can be optimized in polynomial time, albeit at a very slow rate of $O(\epsilon^{-2})$. But nonsmooth functions also play a major role in constrained optimization, and we will see that understanding their structure will be critical when we turn to constrained problems.

1 Subgradients

Recall our first order condition for convex, differentiable functions. For all z and x we had the relation

$$f(z) \geq f(x) + \nabla f(x)^T (z - x)\,.$$

This condition asserts that every convex function can be lower bounded at every point by an affine function. It turns out that this notion generalizes to non-smooth functions as well.

We say that g is a *subgradient* of f at x if

$$f(z) \geq f(x) + g^T (z - x)$$

for all z. This is precisely a generalization of the first-order convexity conditions. Under very mild conditions, every convex function has at least one subgradient.

Before turning to the abstract analysis of subgradients, we can already see that this notion buys us something very powerful. Indeed, suppose f has at least one subgradient at every point. Then $x_\star$ is a global minimizer of f if and only if 0 is a subgradient of f at x. The proof of this wholly trivial. 0 is a subgradient if and only if

$$f(z) \geq f(x_\star) + 0^T (y - x)\,.$$

Note that if f is convex and differentiable, then there is unique subgradient at every x and it is equal to $\nabla f(x)$. We can easily verify this in the one dimensional case. Obviously, $\nabla f(x)$ is a subgradient. Conversely, let g be any subgradient. Note that $f(x+h) \geq f(x) + g^T h$. Rearranging both sides and taking the limit as h goes to zero verifies

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \geq g$$

A similar argument shows

$$-f'(x) = \lim_{h \to 0} \frac{f(x-h) - f(x)}{h} \geq -g$$

Together, these show that $f'(x) = g$.

We will prove this fact in the multidimensional case when we turn to directional derivatives.

2 Directional Derivatives

Though convex functions might not be differentiable at every point, they are differentiable *in every direction.* The *directional derivative* of f at x in direction v is defined as

$$f'(x; v) = \lim_{\alpha \downarrow 0} \frac{f(x + \alpha v) - f(x)}{\alpha}$$

Note that if f is differentiable, $f'(x; v) = \nabla f(x)^T v$. Moreover, $f'(x; v) = -f'(x; -v)$ when f is differentiable. This equality need not hold when f is merely convex and not smooth. But we will show that the limit always exists. This is rather surprising, and a deep property of convex functions.

To proceed, let's begin with one dimensional functions. Let $f : I \to \mathbb{R}$ where I is an interval $[a, b]$ with $a = -\infty$ and $b = \infty$ allowed. If $x < y < z$, then we have the relationship

$$\frac{f(y) - f(x)}{y - x} \leq \frac{f(z) - f(x)}{z - x} \leq \frac{f(z) - f(y)}{z - y} \tag{1}$$

This can be verified by drawing a picture. But the proof is also an immediate consequence of the definition of convex functions.

For $x > a$ and $x \leq b - \alpha$, we can define

$$s^+(x, \alpha) := \frac{f(x + \alpha) - f(x)}{\alpha}$$

which provides a measure of the slope of the function as we move in the positive direction away from x. Note that if $0 < \alpha_1 < \alpha_2$,

$$s^+(x, \alpha_1) \leq s^+(x, \alpha_2)\,. \tag{2}$$

This can be verified by using $y = x + \alpha_1$ and $z = x + \alpha_2$ in equation (1).

Now, we can define

$$f^+(x) = \lim_{\alpha \downarrow 0} \frac{f(x + \alpha) - f(x)}{\alpha} = \inf_{\alpha > 0} \frac{f(x + \alpha) - f(x)}{\alpha}\,.$$

By (2), either $f^+(x)$ is finite, or it is equal to $-\infty$. Similarly, we can define

$$s^-(x,\alpha) := \frac{f(x) - f(x-\alpha)}{\alpha}$$

and

$$f^-(x) = \lim_{\alpha \downarrow 0} \frac{f(x) - f(x-\alpha)}{\alpha} = \sup_{\alpha > 0} \frac{f(x) - f(x-\alpha)}{\alpha}\,.$$

$f^-(x)$ is either finite or equal to ∞. By convention, we'll set $f^-(a) = -\infty$ and $f^+(b) = \infty$, as we can't properly define these limits. We can collect some properties of these limits in the following

Proposition 1 *1. $f^-(x) \le f^+(x)$ for all $x \in I$.*

2. $x \in (a,b)$ implies that $f^+(x)$ and $f^-(x)$ are both finite.

3. If $a \le x < z \le b$, then $f^+(x) < f^-(z)$.

4. f^- and f^+ are both nondecreasing

Proof

1. This is clear at the endpoints by definition. For $t > a$,

   $$s^-(t,\alpha) \le s^+(t,\alpha)\,.$$

   This follows by using $x = t - \alpha$, $y = t$, and $z = t + \alpha$ in (1). Taking limits completes the argument.

2. $f^-(x) \ge s^-(x,\alpha)$ for all α, so it is bounded below. Similarly, $f + (x)$ is bounded above. But then part (a) implies that both must be finite.

3. Using (1) with $y = \frac{1}{2}(z + x)$, we have

   $$s^+(x, \tfrac{1}{2}(z+x)) \le s^-(z, \tfrac{1}{2}(z+x))\,.$$

   Using $f^+(x) \le s^+(x, \frac{1}{2}(z+x))$ and $f^-(z) \ge s^-(z, \frac{1}{2}(z+x))$ completes the proof.

4. This follows from (a) and (c).

■

Note that by our definition, $f^+(x) = f'(x;1)$ and $f^-(x) = -f'(x;-1)$ This proposition proves that the directional derivative exists in both directions for one-dimensional convex functions. Moreover, both $f'(x;1)$ and $-f'(x;-1)$ are nondecreasing functions.

For the multidimensional case, choose $v \in \mathbb{R}^d$ and define $F(t) = f(x + tv)$. F is a convex function and

$$f'(x;v) = \lim_{\alpha \downarrow 0} \frac{f(x+\alpha v) - f(x)}{\alpha} = \lim_{\alpha \downarrow 0} \frac{F(\alpha - F(0)}{\alpha} = F^+(0)$$

Hence, the directional derivative exists in all directions. Moreover, $-f'(x;-v) \le f'(x;v)$ from our proposition.

3 The subdifferential

The collection of subgradients of f at x is called the *subdifferential* of f at x and is denoted $\partial f(x)$. We will establish that the subdifferential of convex functions are nonempty, convex, and compact. (**Note for fans of convex analysis:** we will skirt issues of properness and regularity in this section by assuming that f is finite everywhere.)

Proposition 2 *g is a subgradient of f at x if and only if*

$$f'(x;y) \geq \langle g, y \rangle$$

for all y. Moreover,

$$f'(x;y) = \sup_{g \in \partial f(x)} \langle g, y \rangle$$

This proposition asserts that the function mapping y to $f'(x;y)$ is the support function for the subdifferential of f at x.

Proof [Rockafellar] By the definition of the subgradient,

$$\frac{f(x+\alpha y) - f(x)}{\alpha} \geq \langle g, y \rangle$$

for all $\alpha > 0$. Using (2), this proves the first assertion of the theorem. For the second, note that $f'(x;y)$ is a convex function of y. It is also homogeneous in the sense that

$$f'(x;\alpha y) = \alpha f'(x;y)$$

for $\alpha > 0$. Thus,

$$f'(x;y) = \sup_u \ \langle u, y \rangle - \varphi^*(u)$$

where $\varphi^*(u)$ denotes the Fenchel conjugate of $f'(x;y)$ as a function of y. That is, the closure of $f'(x;y)$ is equal to the double conjugate of $f'(x;y)$. Homogeneity of the directional derivative shows that $\varphi^*(u)$ must equal 0 if $\varphi^*(u)$ is defined (and ∞ otherwise). In particular,

$$\varphi^*(u) = \sup_y \langle u, y \rangle - f'(x;y)\,.$$

Hence, $\varphi^*(u) = 0$ if and only if $f'(x;y) \geq \langle u, y \rangle$ for all y if and only if $u \in \partial f(x)$. This completes the proof. ■

(**Another note for the convex analysis junkies:** This assumes that $f'(x;y)$ is closed and proper. But this can be shown because we are assuming $f(x)$ is finite everywhere. See Rockafellar Thm 23.4.)

Proposition 3 *Let f be a convex function. Then $\partial f(x)$ is nonempty, convex, and compact for all x.*

Proof Existence follows from Proposition 2. It is immediate from the definition that the convex combination of two subgradients is a subgradient. Moreover, the subdifferential must be closed as it is the intersection of set of inequalities. This point is a bit subtle, but note that

$$\partial f(x) = \{g \ : \ \langle a, g\rangle \leq b \ \text{ whenever } a = z - x\,,\ b = f(z) - f(x) \ \text{ for some } \ z \in \mathbb{R}^d\}\,.$$

To prove boundedness, note that the directional derivative is finite everywhere. ■

Proposition 2 also lets us show that if f is convex and differentiable, then the $\nabla f(x)$ is the unique subgradient of f at x. To see this, note that

$$f'(x; y) = \nabla f(x)^T y$$

for all y. If there were two subgradients $g_1 \neq g_2$, then there would be a direction y such that

$$\langle g_1, y\rangle > \langle g_2, y\rangle\,.$$

Hence, for this direction, $f'(x; y) \neq -f'(x; -y)$.

3.1 Subdifferential calculus

We can enumerate some properties important of the subdifferential. Essentially, these properties are what lets us form a "subgradient calculus" from which we can compute subgradients from simple primitives.

Proposition 4 *For $\alpha > 0$, $\partial\,(\alpha f)\,(x) = \alpha\partial f(x)$.*

Proof Immediate from the definition. ■

Proposition 5 *If $h(x) = f(Ax + b)$, then $\partial h(x) = A^T\partial f(Ax + b)$.*

Proof [Bertsekas, 4.2.5] From the definition of directional derivatives, it follows that

$$h'(x; y) = f'(Ax + b; Ay)\,.$$

Let $g \in \partial f(Ax + b)$. Then

$$\langle g, z\rangle \leq f'(Ax + b; z)\,.$$

This implies that

$$\langle A^T g, y\rangle = \langle g, Ay\rangle \leq h'(x; y)$$

for all y. Thus $A^T\partial f(Ax + b) \subset \partial h(x)$. Conversely, suppose $v \in \partial h(x)$ but $v \notin A^T\partial f(Ax + b)$. Since the set $\Omega := A^T\partial f(Ax + b)$ is compact, there exists a hyperplane strictly separating g from Ω. That is, there is a vector y and a scalar β such that

$$y^T(A^T g) < \beta < y^T v$$

for all $g \in \partial f(Ax + b)$. This means that

$$\sup_{g\in\partial f(Ax+b)} (Ay)^T g < y^T v\,.$$

Using Proposition 2, this means

$$h'(x,y) = f'(Ax+b;Ay) < y^T v$$

but this contradicts the fact that $v \in \partial h(x)$. ■

A similar argument gives the following:

Proposition 6 $\partial(f_1+f_2)(x) = \partial f_1(x) + \partial f_2(x) = \{g+h \ : \ g \in \partial f_1(x),\ h \in \partial f_2(x)\}$.

Proof Certainly the "$\supseteq$" direction is an immediate consequence of the definition. The other inclusion is pretty much the same separating hyperplane argument we just made. Suppose there exists $g \in \partial(f_1+f_2)(x)$ but $g \neq \partial f_1(x) + \partial f_2(x)$. Then there exists a hyperplane strictly separating g and $\partial f_1(x) + \partial f_2(x)$:

$$y^T(g_1+g_2) < b < y^T g$$

for all $g_1 \in \partial f_1(x)$, $g_2 \in \partial f_2(x)$. Taking suprema, we see

$$(f_1+f_2)'(x;y) = f_1'(x;y) + f_2'(x;y) = \sup_{g_1 \in \partial f_1(x)} \langle y, g_1 \rangle + \sup_{g_2 \in \partial f_2(x)} \langle y, g_2 \rangle < y^T g$$

contradicting the assertion that $g \in \partial(f_1+f_2)(x)$. ■

3.2 The max function

We conclude this section with one of the most important tools in subdifferential calculus: the subgradient of the max-function. Let I be a compact subset of $\mathbb{R}^n$. Let $\varphi : \mathbb{R}^d \times I \to \mathbb{R}$ be continuous and assume $\varphi(\cdot;i)$ is convex for all $i \in I$. Then

$$f(x) = \max_{i \in I} \varphi(x,i)$$

f is clearly convex as it is the pointwise maximum of convex functions.

Let $\varphi'(x,i;y)$ denote the directional derivative of $\varphi(\cdot,i)$ at x in the direction y. For a fixed x, let $I_{\max}(x)$ denote the set of maximizing points:

$$I_{\max}(x) := \{j \ : \ \varphi(x,j) = \max_{i \in I} \varphi(x,i)\}.$$

The following proposition characterizes the subdifferential of this max-function

Theorem 7 (Danskin's Theorem) .

$$f'(x,y) = \max_{i \in I_{\max}(x)} \varphi'(x,i;y).$$

If $\varphi(\cdot,i)$ is a differentiable function of x for all $i \in I$ and $\nabla_x \varphi(x,\cdot)$ is continuous on I for all x then

$$\partial f(x) = \operatorname{conv}\{\nabla_x \varphi(x,i) \ : \ i \in I_{\max}(x)\}$$

Proof [Bertsekas] Using the definition of f, we have that for any $i \in I_{\max}(x)$,

$$\frac{f(x+\alpha y) - f(x)}{\alpha} \geq \frac{\varphi(x+\alpha y, i) - \varphi(x,i)}{\alpha}$$

because $f(x+\alpha y) \geq \varphi(x+\alpha y, i)$ and $f(x) = \varphi(x,i)$. Taking the limit as α tends to zero proves $f'(x;y) \geq \varphi'(x,i;y)$. Since this is true for all $i \in I_{\max}(x)$,

$$f'(x;y) \geq \sup_{i \in I_{\max}(x)} \varphi'(x,i;y)\,.$$

For the reverse inequality, Let α_k be a decreasing sequence on positive scalars whose limit is 0. For each k, let i_k be some element of $I_{\max}(x+\alpha_k y)$. Since I is compact, there is a subsequence of $\{i_k\}$ that converges to $\hat{i} \in I$. Without loss of generality, assume that i_k converges to $\hat{i}$ (you could just subsample the sequence). Then we have

$$\varphi(x+\alpha_k y, i_k) \geq \varphi(x+\alpha_k y, i)$$

for all $i \in I$. Taking the limit of both sides shows $\varphi(x,\hat{i}) \geq \varphi(x,i)$ for all $i \in I$. This means that $\hat{i} \in I_{\max}(x)$, and we have

$$\begin{aligned}
f'(x;y) &\leq \frac{f(x+\alpha_k y) - f(x)}{\alpha_k} \\
&= \frac{\varphi(x+\alpha_k y, i_k) - \varphi(x,\hat{i})}{\alpha_k} \\
&\leq \frac{\varphi(x+\alpha_k y, i_k) - \varphi(x,i_k)}{\alpha_k} \\
&= -\frac{\varphi(x+\alpha_k y + \alpha_k(-y), i_k) - \varphi(x+\alpha_k y, i_k)}{\alpha_k} \\
&\leq -\varphi'(x+\alpha_k y, i_k; -y) \\
&\leq \varphi'(x+\alpha_k y, i_k; y)
\end{aligned}$$

Letting k go to infinity, we have

$$f'(x;y) \leq \lim_{k\to\infty} \varphi'(x+\alpha_k y, i_k; y) \leq \varphi'(x,\hat{i};y)\,.$$

This proves the first part of the theorem. For the second part, note that for all $i \in I_{\max}(x)$,

$$\begin{aligned}
f(z) &= \max_{j \in I} \varphi(z,j) \\
&\geq \varphi(z,i) \\
&\leq \varphi(x,i) + \nabla_x \varphi(x,i)^T (z-x) \\
&= f(x) + \nabla_x \varphi(x,i)^T (z-x)
\end{aligned}$$

proving that $\varphi(x,i) \in \partial f(x)$. This proves the "$\supseteq$" direction. For the reverse direction, we use our now familiar separating hyperplane argument.

Suppose there is a $v \in \partial f(x)$ that is not in $\text{conv}\{\nabla_x \varphi(x,i) \ : \ i \in I_{\max}(x)\}$. Now, $I_{\max}(x)$ is a compact set because $\varphi(x,\cdot)$ is continuous. Moreover, $\text{conv}\{\nabla_x \varphi(x,i) \ : \ i \in I_{\max}(x)\}$ must be compact. Thus, there exists a y and scalar β such that

$$\langle v, y\rangle > \beta > \nabla_x \varphi(x,i)^T y$$

for all $i \in I_{\max}(x)$. But this means that

$$\langle v, y\rangle > \max_{i \in I_{\max}(x)} \nabla_x \varphi(x,i)^T y = f'(x;y)$$

which is a contradiction by Proposition 2. ■

4 Extended Real-Valued Functions

We say that a function $f : \mathbb{R}^n \to \mathbb{R} \cup \infty$ is *an extended real-valued function*. This is a useful construction for constrained optimization, but one needs to be a bit careful when playing with these functions, as certain properties are not inherited from their real counterparts.

The most common extended real valued function is the indicator function, $\mathbb{I}_C$, of a convex set C

$$\mathbb{I}_C(x) = \begin{cases} 0 & x \in C \\ \infty & \text{otherwise} \end{cases} . \tag{3}$$

Note that convex extended real-valued functions form a convex cone. The sum of two indicator functions is simply the indicator of their intersection. We can even define subgradients of extended convex function. Note that for an indictor function, the subdifferential need not be convex. As a simple example, let $[a,b] \subset \mathbb{R}$ be an interval. Then,

$$\partial I_{[a,b]}(a) = (-\infty, 0]$$

This is simple to check. If, $x \in [a,b]$, and $g \geq 0$,

$$I_{[a,b]}(x) = 0 \geq 0 + g(x-a) = I_{[a,b]}(x) + g(x-a)\,.$$

Similarly, if $x \notin [a,b]$,

$$I_{[a,b]}(x) = \infty \geq 0 + g(x-a) = I_{[a,b]}(x) + g(x-a)\,.$$

Many calculations are simple manipulations inequalities using ∞ in this manner. However, some facts are more difficult. For example, $\partial f_1(x) + \partial f_2(x)$ may not equal $\partial(f_1 + f_2)(x)$. Let $C_1 = \{(x,y) \ : \ x^2 + y^2 \leq 1\} \subset \mathbb{R}^2$ and $C_2 = \{(x,y) \ : \ x = 1\} \subset \mathbb{R}^2$. Then $C_1 \cap C_2 = \{(1,0)\} =: C_3$ and

$$I_{C_1} + I_{C_2} = I_{C3}$$

Now, at $(1,0)$,

$$\begin{aligned} \partial I_{C_1}(1,0) &= \{(1,0)\} \\ \partial I_{C_2}(1,0) &= \{(g,0) \ : \ g \in \mathbb{R}\} \\ \partial I_{C_3}(1,0) &= \mathbb{R}^2 \end{aligned}$$

4.1 Constrained Optimization

Let C be a convex set and let $\mathbb{I}_C$ denote its indicator function. What's the subdifferential of $\mathbb{I}_C(x)$ for $x \in C$? By definition $g \in \partial \mathbb{I}_C(x)$ if and only if

$$\mathbb{I}_C(y) \geq \mathbb{I}_C(x) + g^T(y - x) \tag{4}$$

for all y. This is equivalent to

$$\partial \mathbb{I}_C(x) = \{g \ : \ g^T(x - y) \geq 0 \ \forall y \in C\} \tag{5}$$

for $x \in C$. This set is often called the *normal cone* of C at x.

Consider the constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array} \tag{6}$$

for smooth, convex f. Then $x_\star$ is optimal if and only if $-\nabla f(x_\star) \in \partial \mathbb{I}_C(x_\star)$. In other words, $x_\star$ is optimal if and only if the directional derivative $f'(x_\star; y) = \nabla f(x_\star)^T(y - x_\star)$ is nonnegative for all $y \in C$. This is precisely the minimum principle derived last lecture.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

Feburary 26, 2015 Notes: Benjamin Recht

Lecture 11: The Subgradient and Proximal Point Methods

1 Subgradient Descent

Though we have seen examples where certain elements of the subdifferential are *ascent* directions, there aways exists a descent direction in the subdifferential. Indeed, it is the element with smallest norm. Let g_{mn} be the minimum norm element of the subgradient:

$$g_{\mathrm{mn}} := \arg \min_{z \in \partial f(x)} \|z\|_2 \,.$$

g_{mn} exists because $\partial f(x)$ is nonempty and compact.

Proposition 1 *Either $0 \in \partial f(x)$ or $-g_{\mathrm{mn}}$ is a descent direction.*

Proof Note that in order to be the minimum norm subgradient,

$$\langle g_{\mathrm{mn}}, \hat{g} - g_{\mathrm{mn}} \rangle \geq 0 \tag{1}$$

for all $\hat{g} \in \partial f(x)$. To see this, suppose otherwise. Then

$$g_{\mathrm{mn}} + t(\hat{g} - g_{\mathrm{mn}}) \in \partial f(x)$$

for all $t \in [0,1]$, and

$$\frac{d}{dt}\|g_{\mathrm{mn}} + t(\hat{g} - g_{\mathrm{mn}})\|^2 \Big|_0 = 2\langle g_{\mathrm{mn}}, \hat{g} - g_{\mathrm{mn}} \rangle < 0$$

which contradicts that g_{mn} has minimum norm.

Using (1), we have

$$\langle \hat{g}, g_{\mathrm{mn}} \rangle \geq \|g_{\mathrm{mn}}\|_2^2$$

for all $\hat{g} \in \partial f(x)$.

Now we have

$$\begin{aligned} f'(x; -g_{\mathrm{mn}}) &= \sup_{g \in \partial f(x)} \langle -g_{\mathrm{mn}}, g \rangle \\ &= - \inf_{g \in \partial f(x)} \langle g_{\mathrm{mn}}, g \rangle \\ &\leq -\|g_{\mathrm{mn}}\|_2^2 \end{aligned}$$

proving that $-g_{\mathrm{mn}}$ is a descent direction whenever it is nonzero. ■

This suggests a natural algorithm for minimizing convex, nonsmooth functions. Compute the minimum norm element of the subdifferential and follow it down hill. There is only one problem with this approach: computing the minimum norm element might be prohibitively expensive. In the next section, we show that a very naive algorithm that simply follows arbitrary subgradients will actually converge.

2 The subgradient method

The subgradient method is rather simple: at each step k, we choose an element of the subdifferential $g_k \in \partial f(x_k)$ and set

$$x_{k+1} = x_k - \alpha_k g_k \,.$$

Though we have already pointed out that this method may be taking ridiculous steps, it turns out that the average of the iterates

$$\bar{x}_k = \left(\sum_{j=1}^{k} \alpha_k \right)^{-1} \sum_{j=1}^{k} \alpha_k x_k$$

converges to a minimizer of f.

The proof of this method is *nearly identical* to the proof of the converge of the stochastic gradient method for convex functions with bounded stochastic gradients. We simply assume that for all x,

$$\|g\|_2 \leq G$$

for all $g \in \partial f(x)$. Note that this assumption implies that f must be Lipschitz with constant G (why?).

As we did with the stochastic gradient method. α_k be our stepsize sequence. Define

$$\lambda_k = \frac{1}{\sum_{j=0}^{k} \alpha_j} \,.$$

This is the sum of all the stepsizes up to iteration k. Also define

$$\bar{x}_k = \lambda_k^{-1} \sum_{j=0}^{k} \alpha_j x_j \,.$$

$\bar{x}_k$ is the average of the iterates weighted by the stepsize. We are going to analyze the deviation of $f(\bar{x}_k)$ from optimality.

Let $D_0 = \|x_0 - x_\star\|^2$. D_0 is the initial distance to an optimal solution.

To proceed, we just expand the distance to an optimal solution:

$$\begin{aligned} \|x_{k+1} - x_\star\|^2 &= \|x_k - \alpha_k g_k - x_\star\|^2 \\ &= \|x_k - x_\star\|^2 - 2\alpha_k \langle g_k, x_k - x_\star \rangle + \alpha_k^2 \|g_k\|^2 \\ &\leq \|x_k - x_\star\|^2 - 2\alpha_k \langle g_k, x_k - x_\star \rangle + \alpha_k^2 G^2 \,. \end{aligned} \tag{2}$$

Now we have

$$f(\bar{x}_T) - f(x_\star) \le \lambda_T^{-1} \sum_{t=1}^{T} \alpha_t (f(x_t) - f(x_\star)) \tag{3a}$$

$$\le \lambda_T^{-1} \sum_{t=1}^{T} \alpha_t \langle g_t, x_t - x_\star \rangle \tag{3b}$$

$$\le \lambda_T^{-1} \sum_{t=1}^{T} \tfrac{1}{2} \left(\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2 \right) + \tfrac{1}{2} \alpha_t^2 G^2 \tag{3c}$$

$$= \frac{\|x_0 - x_\star\|^2 - \|x_0 - x_T\|^2 + G^2 \sum_{t=0}^{T} \alpha_t^2}{2\lambda_T} \tag{3d}$$

$$\le \frac{D_0^2 + G^2 \sum_{t=0}^{T} \alpha_t^2}{2 \sum_{t=1}^{T} \alpha_t} \tag{3e}$$

Here, (3a) follows because f is convex, (3b) is the definition of a subgradient, and (3c) uses (2).

Note that we also immediately have the bound

$$\min_{t \le T} f(x_t) - f(x_\star) \le \lambda_T^{-1} \sum_{t=1}^{T} \alpha_t (f(x_t) - f(x_\star))$$

so our analysis works for both the average iterate and the *best* iterate.

2.1 Stepsizes

Let's dive into different possibilities for our stepsize.

Constant step size. First, we can just pick $\alpha_t = \alpha$. In this case, we get

$$f(\bar{x}_T) - f(x_\star) \le \frac{D_0^2 + TG^2\alpha^2}{2T\alpha}.$$

The choice of $\alpha = \frac{\theta D_0}{G\sqrt{T}}$ results in the convergence rate

$$f(\bar{x}_T) - f(x_\star) \le \tfrac{1}{2} \left(\theta + \theta^{-1} \right) \frac{D_0 G}{\sqrt{T}}.$$

Constant step length. An alternative choice is to choose $\alpha_k = \frac{\alpha}{\|g_k\|}$. This ensures that the *length* of each step is a constant. In this case, we have the bound

$$f(\bar{x}_T) - f(x_\star) \le \frac{D_0^2 + T\alpha^2}{2T\alpha/G}.$$

With $\alpha = \frac{\theta D_0}{\sqrt{T}}$, we get the (worst-case) convergence rate

$$f(\bar{x}_T) - f(x_\star) \le \tfrac{1}{2} \left(\theta + \theta^{-1} \right) \frac{D_0 G}{\sqrt{T}}$$

This is exactly the same as what we had before! But note that here we only had to estimate the distance to optimality, not the maximal norm of the gradient, to get a decent rate of convergence.

2.2 Diminishing Step Size

Diminishing stepsizes are attractive as one doesn't have to choose a final T, and we don't have to estimate any quantities about f.

Note that for any sequence α_k such that α_k tends to zero, but $\sum_k \alpha_k$ diverges, then

$$\lim_{T\to\infty} f\left(\bar{x}_T\right) = f(x_\star)\,.$$

This is particularly easy to see if

$$\sum_k \alpha_k^2 = M < \infty$$

In this case,

$$f(\bar{x}_T) - f_\star \leq \frac{D_0^2 + G^2 \sum_{t=0}^T \alpha_t^2}{2\sum_{t=1}^T \alpha_t} \leq \frac{D_0^2 + G^2 M}{2\sum_{t=1}^T \alpha_t}$$

and the left-hand side clearly tends to zero.

To see that this works for general diminishing stepsizes, one only needs to prove that

$$\frac{\sum_{k=1}^\infty \alpha_k^2}{\sum_{k=1}^\infty \alpha_k} \to 0$$

whenever the sum of α_k diverges but α_k tends to zero. We close this section by deriving more quantitative bounds for an explicit stepsize choice. Suppose we set $\alpha_k = \frac{\theta}{\sqrt{k}}$. Then we have

$$f(\bar{x}_T) - f_\star \leq \frac{D_0^2 + G^2\theta^2 \sum_{t=0}^T t^{-1}}{2\theta \sum_{t=1}^T t^{-1/2}} \leq \frac{D_0^2 + G^2\theta^2 \log(T)}{2\theta\sqrt{T}}\,.$$

Note that this bound tends to zero at a rate of $\log(T)/\sqrt{T}$. This is slower than the $T^{-1/2}$ rate of a constant stepsize, but we are guaranteed asymptotic convergence to zero.

Note that any other choice of $\alpha_k \propto k^{-p}$ for $p \in (0,1)$ would have yielded a worse convergence rate.

3 The Proximal Point Method

Let P be an extended-real-valued convex function on $\mathbb{R}^n$. Define the operator

$$\mathrm{prox}_P(x) = \arg\min_y \tfrac{1}{2}\|x-y\|_2^2 + P(y) \tag{4}$$

Since the optimized function is strongly convex, it must have a unique optimal solution. Therefore, we can conclude that $\mathrm{prox}_P(x)$ is a well-defined mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. By the first order optimality conditions, we conclude that $\mathrm{prox}_P(x)$ is the unique point satisfying

$$x - \mathrm{prox}_P(x) \in \partial P(\mathrm{prox}_P(x))\,. \tag{5}$$

The definition of prox_P also reveals that it is well-defined for all $x \in \mathbb{R}^n$, and maps onto the set $\mathrm{dom}(P) := \{z \in \mathbb{R}^n \,:\, P(z) < \infty\}$. The mapping prox_P is called the *proximity operator* or *proximal point mapping* associated with P.

Let's look at some examples.

1. If $\mathbb{I}_C$ is an indicator function for a convex set C

$$\mathbb{I}_C(x) = \begin{cases} 0 & x \in C \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

then $\text{prox}_{\mathbb{I}_C}$ is the Euclidean projection onto C. That is, $\text{prox}_{\mathbb{I}_C}(x)$ is the closest point in the set C to x in Euclidean distance.

2. For $\mathbb{I}_{\mathbb{R}_+}$, this proximity mapping takes on the trivial form:

$$\mathbb{I}_{\mathbb{R}_+}(x)_i = \max(x_i, 0) \tag{7}$$

3. For $P(x) = \frac{\mu}{2}\|x\|_2^2$, $\text{prox}_P(x) = \frac{1}{1+\mu}x$. That is, $\text{prox}_P(x)$ is equal to a multiple of x, shrunk towards the origin.

4. For $P(x) = \mu\|x\|_1$,

$$\text{prox}_P(x)_i = \begin{cases} x_i + \mu & x_i < -\mu \\ 0 & -\mu \le x_i \le \mu \\ x_i - \mu & x_i > \mu \end{cases} \tag{8}$$

This function is called the *shrinkage* operator and has many applications in signal processing. To see that this is the correct form, one needs only to analyze the optimality conditions of the one dimensional problem

$$\text{minimize } \tfrac{1}{2}(x-y)^2 + \mu|y| \tag{9}$$

3.1 The Proximal Point Algorithm

Proximity operators have many algorithmic applications. As a warm up, consider the following simple iteration: pick $x_0 \in \mathbb{R}^n$ and $\alpha > 0$ and define the iteration $x_{k+1} = \text{prox}_{\alpha P}(x_k)$. This simple iteration can be shown to converge to a minimizer of the function P. To prove this, we need the following two lemmas. The first is a simple consequence of the convexity of P.

Lemma 2 *Let P be convex on X. Let x, $y \in X$, and let $g_y \in \partial P(y)$ and $g_x \in \partial P(y)$. Then $\langle g_x - g_y, x - y\rangle \ge 0$.*

Proof By the definition of the subdifferential, we have

$$\begin{aligned} P(x) - P(y) &\ge \langle g_y, x - y\rangle \\ P(y) - P(x) &\ge \langle g_x, y - x\rangle \end{aligned} \tag{10}$$

Adding these two equations gives $-\langle g_x - g_y, x - y\rangle \le 0$. ■

The second lemma uses this key inequality to establish several facts about the proximity operator. This lemma is proven in [**?**].

Lemma 3 *Let $Q_\alpha(x) := x - \text{prox}_{\alpha P}(x)$. Then we have*

1. $\alpha^{-1}Q_\alpha(x) \in \partial P(\text{prox}_{\alpha P}(x))$

2. $\langle \operatorname{prox}_{\alpha P}(x) - \operatorname{prox}_{\alpha P}(z), Q_\alpha(x) - Q_\alpha(z)\rangle \geq 0$

3. $\| \operatorname{prox}_{\alpha P}(x) - \operatorname{prox}_{\alpha P}(z)\|^2 + \|Q_\alpha(x) - Q_\alpha(z)\|^2 \leq \|x - z\|^2$

4. $\|x - z\| = \| \operatorname{prox}_{\alpha P}(x) - \operatorname{prox}_{\alpha P}(z)\|$ *if and only if* $x - z = \operatorname{prox}_{\alpha P}(x) - \operatorname{prox} \alpha P(z)$

Proof The first assertion follows from the definitions. The second assertion follows from (i), and Lemma 2. The third assertion follows from (ii) after expanding the identity

$$\|x - z\|^2 = \|[\operatorname{prox}_{\alpha P}(x) - \operatorname{prox}_{\alpha P}(z)] + [Q_\alpha(x) - Q_\alpha(z)]\|^2 .$$

(iv) follows immediately from (iii). ■

By Lemma 3 (3), we have

$$\| \operatorname{prox}_{\alpha P}(x) - \operatorname{prox}_{\alpha P}(z)\|^2 \leq \|x - z\|^2 \tag{11}$$

and we say that the proximity operator is *nonexpansive*. This is the essential property needed to prove the convergence of the proximal point method. That the proximity operator is nonexpansive also plays a role in the projected gradient algorithm, analyzed below. As a particular consequence, we have the following

Corollary 4 *Let C be a closed convex set. Let $\Pi_C(x)$ denote the closest point to x in the set C in the Euclidean distance. Then for all $x, y \in \mathbb{R}^n$, $\|\Pi_C(x) - \Pi_C(y)\| \leq \|x - y\|$.*

Proof Since $\Pi_C = \operatorname{prox}_{\mathbb{I}_C}$, this follows immediately from (11). ■

Using the nonexpansive property of the proximity operator, we can now verify the convergence of the proximal point method. Since $\operatorname{prox}_{\alpha P}$ is non-expansive, $\{z_k\}$ lies in a compact set and must have a limit point $\bar{z}$. Also for any $z_\star$ with $0 \in \partial P(z_\star)$,

$$\|z_{k+1} - z_\star\| = \| \operatorname{prox}_{\alpha P}(z_k) - \operatorname{prox}_{\alpha P}(z_\star)\| \leq \|z_k - z_\star\| \tag{12}$$

which means that the sequence $\|z_k - z_\star\|$ is monotonically non-increasing. Therefore

$$\lim_{k\to\infty} \|z_k - z_\star\| = \|\bar{z} - z_\star\| . \tag{13}$$

where $\bar{z}$ is any limit point of z_k. By continuity we have $\operatorname{prox}_{\alpha P}(\bar{z})$ is also a limit point of z_k. Therefore, we must have

$$\| \operatorname{prox}_{\alpha P}(\bar{z}) - \operatorname{prox}_{\alpha P}(z_\star)\| = \| \operatorname{prox}_{\alpha P}(\bar{z}) - z_\star\| = \|\bar{z} - z_\star\| \tag{14}$$

But this means that $\operatorname{prox}_{\alpha P}(\bar{z}) - \operatorname{prox}_{\alpha P}(z_\star) = \bar{z} - z_\star$, and in turn that $\operatorname{prox}_{\alpha P}(\bar{z}) = \bar{z}$ and $0 \in \partial P(\bar{z})$. Now using $\bar{z}$ for $z_\star$ in (13) shows that

$$\lim_{k\to\infty} \|z_k - \bar{z}\| = 0 \tag{15}$$

In other words, the sequence z_k converges to $\bar{z}$.

4 The proximal gradient algorithm

The projected gradient algorithm combines a proximal step with a gradient step. This lets us solve a variety of constrained optimization problems with simple constraints, and it lets us solve some non-smooth problems at linear rates.

We will aim to analyze a function h which admits a decomposition

$$h(x) = f(x) + P(x) \tag{16}$$

where f is smooth and P is a convex extended real valued function. Let us assume that ∇f is Lipschitz so that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

Let us define a projected gradient scheme to solve this problem. Let $\alpha_0, \ldots, \alpha_T, \ldots$, be a sequence of positive step sizes. Choose $x_0 \in X$, and iterate

$$x_{k+1} = \text{prox}_{\alpha_k P}(x_k - \alpha_k \nabla f(x_k)) \,. \tag{17}$$

The algorithm alternates between taking gradient steps and then taking proximal point steps.

The key idea behind this algorithm is summed up by the following proposition

Proposition 5 *Let f be differentiable and convex and let P be convex. $x_\star$ is an optimal solution of*

$$\text{minimize}_x \; f(x) + P(x) \tag{18}$$

if and only if $x_\star = \text{prox}_{\alpha P}(x_\star - \alpha \nabla f(x_\star))$ for all $\alpha > 0$.

Proof $x_\star$ is an optimal solution if and only if $-\nabla f(x_\star) \in \partial P(x_\star)$. This is equivalent to

$$(x_\star - \alpha \nabla f(x_\star)) - x_\star \in \alpha \partial P(x_\star) \,,$$

which is equivalent to $x_\star = \text{prox}_{\alpha P}(x_\star - \alpha \nabla f(x_\star))$. ■

For non-convex f, we see that a fixed point of the projected gradient iteration is a stationary point of h. We first analyze the convergence of this projected gradient method for arbitrary smooth f, and then focus on strongly convex f.

To summarize, we have proven that proximal steps have all of the properties of projections. Hence, our analyses of projected gradient immediately apply to the proximal point method. Moreover, now that we have the tools of subgradient calculus, we can prove a more precise convergence rate for weakly convex functions.

4.1 New convergence proof for weakly convex functions.

The proof is borrowed from Lieven Vandenberghe's course notes[1]. Define the function:

$$Q_\alpha(x) = \tfrac{1}{\alpha}(x - \text{prox}_{\alpha P}(x))$$

Note that we have $Q_\alpha(x) \in \partial P(\text{prox}_{\alpha P})$.

[1] `http://www.seas.ucla.edu/~vandenbe/236C/lectures/ppm.pdf`

By the first order convexity condition

$$\begin{aligned} P(\text{prox}_{\alpha P}(x)) &\leq P(z) + Q_\alpha(x)^T(\text{prox}_{\alpha P}(x) - z) \\ &= P(z) + Q_\alpha(x)^T(x - z) - \alpha\|Q_\alpha(x)\|^2 \end{aligned} \tag{19}$$

Using (19) with $z = x$, we have

$$P(\text{prox}_{\alpha P}(x)) \leq P(x) - \alpha\|Q_\alpha(x)\|^2$$

implying that the proximity operator reduces the value of the function P. That is, the proximal point method is a descent method and the function values $P(x_k)$ form a decreasing sequence.

Using (19) with $z = x_\star$ proves that

$$\begin{aligned} P(\text{prox}_{\alpha P}(x)) - P_\star &\leq Q_\alpha^T(x - x_\star) - \alpha\|Q_\alpha(x)\|^2 \\ &= \frac{1}{2\alpha}\left(\|x - x_\star\|^2 - \|\,\text{prox}_{\alpha P}(x) - x_\star\|^2\right) - \frac{\alpha}{2}\|Q_\alpha(x)\|^2 \end{aligned}$$

In terms of the iterates of the algorithm, this means

$$P(x_{k+1}) - P_\star \leq \frac{1}{2\alpha}\left(\|x_k - x_\star\|^2 - \|x_{k+1} - x_\star\|^2\right) - \frac{1}{2\alpha}\|x_k - x_{k+1}\|^2 \tag{20}$$

Summing (20) over all iterates yields

$$\sum_{i=1}^{k} P(x_i) - P_\star + \frac{1}{2\alpha}\sum_{i=1}^{k}\|x_i - x_{i+1}\|^2 \leq \frac{1}{2\alpha}\|x_0 - x_\star\|^2$$

Since the function values form a non-increasing sequence this means

$$P(x_k) - P_\star \leq \frac{1}{2\alpha k}\|x_0 - x_\star\|^2 .$$

Note that in the previous argument, we were only able to show the iterates themselves converge. This proves that the function values converge at a rate that is asymptotically faster than the subgradient method. Also note that picking $\alpha = \infty$ would be ideal. But this is obvious! This is equivalent to minimizing P directly.

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

March 3, 2016 Notes: Ben Recht and Ashia Wilson

Lecture 12: Duality Theory

1 Optimality Conditions for Equality Constrained Optimization

Recall that $x_\star$ minimizes a smooth, convex function f over a closed convex set Ω if and only if

$$\langle \nabla f(x_\star), x - x_\star \rangle \geq 0 \quad \forall x \in \Omega \,. \tag{1}$$

Let's specialize this to the special case where Ω is an affine set. Let A be an $n \times d$ matrix with rank n such that $\Omega = \{x \ : \ Ax = b\}$ for some $b \in \mathbb{R}^n$. Note that we can always assume that $\text{rank}(A) = n$ or else we would have redundant constraints. We could also parameterize Ω as $\Omega = \{x_0 + v : Av = 0\}$ for any $x_0 \in \Omega$. Then using (1), we have

$$\langle \nabla f(x_\star), x - x_\star \rangle \geq 0 \ \ \forall x \in \Omega \quad \text{if and only if} \quad \langle \nabla f(x_\star), u \rangle \geq 0 \ \ \forall u \in \ker(A) \,.$$

But since $\ker A$ is a subspace, this can hold if and only if $\langle \nabla f(x_\star), u \rangle = 0$ for all $u \in \ker(A)$. In particular, this means, $\nabla f(x_\star)$ must lie in $\ker(A)^\perp$. Since we have that $\mathbb{R}^d = \ker(A) \oplus \text{Im}(A^T)$, this means that $\nabla f(x_\star) = A^T \lambda$ for some $\lambda \in \mathbb{R}^n$.

To summarize, this means that $x_\star$ is optimal for f over Ω if and only if there $\exists \lambda_\star \in \mathbb{R}^m$ such that

$$\begin{cases} \nabla f(x_\star) + A^T \lambda_\star = 0 \\ Ax_\star = b \end{cases}$$

These optimality conditions are known as the *Karush-Kuhn-Tucker Conditions* or *KKT Conditions*.

As an example, consider the equality constrained quadratic optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} x^T Q x + c^T x \\ \text{subject to} & Ax = b \end{array}$$

The KKT conditions can be expressed in matrix form

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} c \\ b \end{bmatrix} .$$

1.1 Nonlinear constraints

Suppose we want to minimize a smooth function over an intersection of an affine set and a closed convex set

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X \\ & Ax = b \end{array} \tag{2}$$

where A is again a full rank $n \times d$ matrix. In this section, we will generalize (1) to show

Proposition 1. *$x_\star$ is optimal for (2) if and only if there exists a $\lambda_\star$ in $\mathbb{R}^n$ such that*

$$\begin{cases} \langle \nabla f(x_\star) + A^T \lambda_\star, x - x_\star \rangle \geq 0 \ \ \forall x \in \Omega \\ Ax_\star = b \end{cases} .$$

The key to our analysis here will be to rely on convex analytic arguments. Let Ω be a closed convex set. Let's define the *tangent cone* of Ω at x as

$$\mathcal{T}_\Omega(x) = \text{cone}\{z - x : z \in \Omega\}$$

The tangent cone is the set of all directions that can move from x and remain in Ω. We can also define the *normal cone* of Ω at x to be the set

$$\mathcal{N}_\Omega = \mathcal{T}_\Omega(x)^\circ = \{u \ : \ \langle u, v \rangle \leq 0 \ \forall v \in \mathcal{T}_\Omega(x)\} \ .$$

Note that when there is no equality constraint, our constrained optimality condition is completely equivalent to the assertion

$$-\nabla f(x_\star) \in \mathcal{N}_\Omega(x_\star) \, . \tag{3}$$

Thus, to prove Proposition 1, it will suffice for us to understand the normal cone of the set

$$\Omega \cap \{z \ : \ Az = b\}$$

at the point $x_\star$.

To obtain a reasonable characterization, we begin by proving a general fact.

Proposition 2. *Let Ω be a closed, convex set. Let $\mathcal{A}$ denote the affine set $\{x \ : \ Ax = b\}$. Suppose there $\exists x \in \text{ri}(\Omega) \cap \mathcal{A}$. Then,*

$$\mathcal{N}_{\Omega \cap \mathcal{A}}(x) = \mathcal{N}_\Omega(x) + \{A^T \lambda \ : \ \lambda \in \mathbb{R}^n\} \, .$$

Proof. The "$\subseteq$" assertion is straightforward. To see this, suppose $z \in \Omega$ satisfies $z \in \text{null}(A)$. Then if $u \in \mathcal{N}_\Omega(x)$ and $\lambda \in \mathbb{R}^n$, we have

$$\langle z - x, u + A^T \lambda \rangle = \langle z - x, u \rangle \leq 0$$

implying that $u + A^T \lambda \in \mathcal{N}_{\Omega \cap \mathcal{A}}(x)$.

For the reverse inclusion, let $v \in \mathcal{N}_{\Omega \cap \mathcal{A}}(x)$. Then we have

$$v^T (z - x) \leq 0$$

for all $z \in \Omega \cap \mathcal{A}$. Now define the sets

$$C_1 = \left\{ (y, \mu) \in \mathbb{R}^{d+1} \ : \ y = z - x, \ z \in \Omega, \ \mu \leq v^T y \right\}$$
$$C_2 = \left\{ (y, \mu) \in \mathbb{R}^{d+1} \ : \ y \in \ker(A), \ \mu = 0 \right\} .$$

Note that $\text{ri}(C_1) \cap C_2 = \emptyset$ because otherwise there would exist a $(\hat{y}, \hat{\mu})$ such that

$$v^T \hat{y} > \hat{\mu} = 0$$

and $\hat{y} \in \mathcal{T}_{\Omega\cap\mathcal{A}}(x)$. This would contradict our assumption that $v \in \mathcal{N}_{\Omega\cap\mathcal{A}}(x)$. Since their intersection is empty, we can properly separate $\mathrm{ri}(C_1)$ from C_2. Indeed, since C_2 is a subspace and C_1 has nonempty relative interior, there must be a (w, γ) such that

$$\inf_{(y,\mu)\in C_1} \{w^T y + \gamma\mu\} < \sup_{(y,\mu)\in C_1} \{w^T y + \gamma\mu\} \leq 0$$

while

$$w^T u = 0$$

for all $u \in \ker(A)$. In particular, this means that $w = A^T\lambda$ for some $\lambda \in \mathbb{R}^n$.

Now, γ must be nonnegative, as otherwise,

$$\sup_{(y,\mu)\in C_1} \{w^T y + \gamma\mu\} = \infty$$

(which can be seen by letting μ tend to negative infinity). If $\gamma = 0$, then

$$\sup_{y\in C_1} w^T y \leq 0$$

but the set $\{y \ : \ w^T y = 0\}$ does not contain the set $\{z - x \ : \ z \in \Omega\}$ as the infimum is strictly negative. This means that the relative interior of $\Omega - \{x\}$ cannot intersect the kernel of A which contradicts our assumptions. Thus, we can conclude that γ is strictly positive. By homogeneity, we may as well assume that $\gamma = 1$.

To complete the argument, note that we now have

$$(w + v)^T(z - x) \leq 0$$

for all $z \in \Omega$. This means that $v + w \in \mathcal{N}_\Omega(x)$. And we have already shown that $w = A^T\lambda$. Thus,

$$v = (v + w) - w \in \mathcal{N}_\Omega(x) + \mathcal{N}_\mathcal{A}(x)\,.$$

□

Let's now translate the consequence of this proposition for our problem. Using (3) and Proposition 2, we have that $x_\star$ is optimal for

$$\min f(x) \quad s.t \quad x \in \Omega, \quad Ax = b$$

if and only if $Ax_\star = b$ and there exists a $\lambda \in \mathbb{R}^n$ such that

$$\langle \nabla f(x_*) + A^T\lambda, x - x_*\rangle \geq 0 \quad \forall x \in \Omega\,.$$

This reduction is not immediately useful to us, as it doesn't provide an algorithmic approach to solving the constrained optimization problem. However, it will form the basis of our dive into duality.

2 Duality

Duality lets us associate to any constrained optimization problem, a concave maximization problem whose solutions lower bound the optimal value of the original problem. In particular, under mild assumptions, we will show that one can solve the primal problem by first solving the dual problem.

We'll continue to focus on the standard primal problem for an equality constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \\ & Ax = b \end{array} \tag{4}$$

Here, assume that Ω is a closed convex set, f is differentiable, and A is full rank.

The key behind duality (here, Lagrangian duality) is that problem (4) is equivalent to

$$\min_{x \in \Omega} \max_{\lambda \in \mathbb{R}^n} f(x) + \lambda^T (Ax - b)$$

To see this, just note that if $Ax \neq b$, then the max with respect to λ is infinite. On the other hand, if $Ax = b$ is feasible, then the max with respect to λ is equal to $f(x)$.

The *dual problem* associated with (4) is

$$\max_{\lambda \in \mathbb{R}^n} \min_{x \in \Omega} f(x) + \lambda^T (Ax - b)$$

Note that the function

$$q(\lambda) := \min_{x \in \Omega} f(x) + \lambda^T (Ax - b)$$

is always a concave function as it is a minimum of linear functions. Hence the dual problem is a concave maximization problem, regardless of what form f and Ω take. We now show that it always lower bounds the primal problem.

2.1 Weak Duality

Proposition 3. *For any function $\varphi(x, z)$,*

$$\min_x \max_z \varphi(x, z) \geq \max_z \min_x \varphi(x, z)\,.$$

Proof. The proof is essentially tautological. Note that we always have

$$\varphi(x, z) \geq \min_x \varphi(x, z)$$

Taking the maximization with respect to the second argument verifies that

$$\max_z \varphi(x, z) \geq \max_z \min_x \varphi(x, z) \quad \forall x\,.$$

Now, minimizing the left hand side with respect to x proves

$$\min_x \max_z \varphi(x, z) \geq \max_z \min_x \varphi(x, z)$$

which is precisely our assertion. □

2.2 Strong duality

For convex optimization problems, we can prove a considerably stronger result. Namely, that the primal and dual problems attain the same optimal value. And, moreover, that if we know a dual optimal solution, we can extract a primal optimal solution from a simpler optimization problem.

Theorem 4 (Strong Duality)**.**

1. *If* $\exists z \in \text{relint}(\Omega)$ *that also satisfies our equality constraint, and the primal problem has an optimal solution, then the dual has an optimal solution and the primal and dual values are equal*
2. *In order for* $x_\star$ *to be optimal for the primal and* $\lambda_\star$ *optimal for the dual, it is necessary and sufficient that* $Ax_\star = b$ *,* $x_\star \in \Omega$ *and*

$$x_\star \in \arg\min_{x\in\Omega} \ \mathcal{L}(x, \lambda_\star) = f(x) + {\lambda_\star}^T(Ax - b)$$

Proof. For all λ and all feasible x

$$q(\lambda) \leq f(x) + \lambda(Ax - b) = f(x)$$

where the second equality holds because $Ax = b$.

Now by Proposition 1, $x_\star$ is optimal if and only if there exists a $\lambda_\star$ such that

$$\langle \nabla f(x_\star) + A^T\lambda_\star, x - x_\star\rangle \geq 0 \ \ \forall x \in \Omega$$

and $Ax_\star = b$. Note that this condition means that $x_\star$ minimizes $\mathcal{L}(x, \lambda_\star)$ over Ω.

By preceding argument, it now follows that

$$\begin{aligned} q(\lambda_\star) &= \inf_{x\in\Omega} \mathcal{L}(x, \lambda_\star) \\ &= \mathcal{L}(x_\star, \lambda_\star) \\ &= f(x_\star) + {\lambda_\star}^T(Ax_\star - b) = f(x_\star) \end{aligned}$$

which completes the proof. □

University of California, Berkeley Lecturer: Benjamin Recht

CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis

March 8, 2016 Notes: Benjamin Recht

Lecture 13: Dual Decomposition

1 Dual Ascent

Let's again begin with an equality constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \\ & Ax = b \end{array} \tag{1}$$

Here, assume that Ω is a closed convex set, f is convex and differentiable, and A is full rank.

The Lagrangian for this problem is given by the function

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T (Ax - b)\,.$$

As we showed last time, problem (1) is equivalent to the optimization problem

$$\min_{x \in \Omega} \max_{\lambda \in \mathbb{R}^n} f(x) + \lambda^T (Ax - b)\,.$$

The *dual problem* associated with (1) is

$$\max_{\lambda \in \mathbb{R}^n} \min_{x \in \Omega} f(x) + \lambda^T (Ax - b)$$

And the concave function

$$q(\lambda) := \min_{x \in \Omega} f(x) + \lambda^T (Ax - b)$$

is called the *dual function* associated with problem (1).

Recall again from last lecture that if the relative interior of Ω contains a point satisfying the equality constraint then *strong duality* holds. In particular, if we can solve the dual problem, then we can find a primal optimal point by solving the problem

$$\text{minimize}_{x \in \Omega} \max_{\lambda \in \mathbb{R}^n} f(x) + \lambda_\star^T (Ax - b) \tag{2}$$

where $\lambda_\star$ is a dual optimal solution.

Note that even if λ is only approximately dual optimal, solving (2) gives a reasonable approximation to the original optimization problem. This can be seen by the calculation

$$\begin{aligned} f(x_\star) = q(\lambda_\star) &\leq q(\lambda) + \epsilon \\ &= \inf_{x \in \Omega} \mathcal{L}(x, \lambda) + \epsilon \\ &\leq \mathcal{L}(x, \lambda) + \epsilon \\ &= f(x) + \lambda^T (Ax - b) + \epsilon \end{aligned}$$

Hence, if $\|Ax - b\|$ is small and our dual optimal solution is nearly accurate, then we get a reasonable approximation to the optimal function value.

2 Algorithms

So how do we solve the dual problem? It's concave, so we can apply the subgradient method:

$$x^{(k)} \leftarrow \arg\min_{x\in\Omega} \mathcal{L}(x, \lambda^{(k)}$$
$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} + s_k(Ax^{(k)} - b)$$

Using our analysis, we then have

$$\frac{1}{\sum_{k=1}^{T} s_k} \sum_{k=1}^{T} s_k \lambda_k \to \lambda_\star$$

at a rate of $O(T^{-1/2})$.

To get a $1/T$ rate, we can do something a bit more clever. Note that we can apply the proximal-point method to the dual problem. This would result in the iteration

$$\lambda^{(k+1)} \leftarrow \arg\max_{\lambda} \inf_{x\in\Omega} f(x) + \lambda^T(Ax - b) - \frac{1}{2\alpha_k}\|\lambda - \lambda^{(k)}\|^2$$

Now, the objective is convex in x and strongly convex in λ. So we can swap the infimum and supremum by Sion's minimax theorem. This results in the equivalent problem

$$\inf_{x\in\Omega} \left\{ \max_{\lambda} f(x) + \lambda^T(Ax - b) - \frac{1}{2\alpha_k}\|\lambda - \lambda^{(k)}\|^2 \right\}$$

But the internal problem here has a trivial solution with respect to λ:

$$\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k(Ax - b)$$

Plugging this solution back in for λ shows that the optimal x can be found by

$$\inf_{x\in\Omega} f(x) + \lambda^{(k)}(Ax - b) + \frac{\alpha_k}{2}\|Ax - b\|^2 =: \mathcal{L}_{\alpha_k}(x, \lambda^{(k)})\,.$$

That is, we find the next x by minimizing an *Augmented Lagrangian*.

This results in the iteration

$$x^{(k)} \leftarrow \arg\min_{x\in\Omega} \mathcal{L}_{\alpha_k}(x, \lambda^{(k)})$$
$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha_k(Ax^{(k)} - b)$$

Note that this algorithm is *nearly identical* to the subgradient method. The only difference is that we have to solve an augmented Lagrangian for the x-step. This can speed up iterations, but also may add algorithmic difficulty when computing the x-step.

2.1 Practical method of multipliers

Note that the proximal-point method is guaranteed to converge for even a constant step-size α_k. Nonetheless, there are some heuristics that seem to work very well in practice. In particular, the Lancelot code for nonlinear programming makes the following recommendations:

The code has two parameters $\eta \in (0, 1)$ and $\gamma > 1$. We perform the following steps

1. Start with a small α_0.
2. After minimizing your augmented Lagrangian at step k, compute $\delta = \|Ax^{(k)} - b\|^2$.
3. If $\delta < \eta\delta_k$, our iterate became more feasible. In this case, we don't need to increase the penalty in the augmented Lagrangian. Thus, we proceed as
   (a) $\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha_k(Ax^{(k)} - b)$
   (b) $\alpha_{k+1} \leftarrow \alpha_k$
   (c) $\delta_{k+1} \leftarrow \delta$
4. If $\delta \geq \eta\delta_k$, then we didn't improve the feasibility of x. So we increase the penalty parameter and try again:
   (a) $\lambda^{(k+1)} \leftarrow \lambda^{(k)}$
   (b) $\alpha_{k+1} \leftarrow \gamma\alpha_k$
   (c) $\delta_{k+1} \leftarrow \delta_k$

Typical values for η are $1/4$ and for γ is 10.

3 Examples

3.1 Consensus Optimization

Let $G = (V, E)$ be a graph and consider the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{v\in V} f_v(x_v) \\ \text{subject to} & x_u = x_v \qquad (u,v) \in E \end{array} \tag{3}$$

The agumented Lagrangian is

$$\begin{aligned} \mathcal{L}(x,\lambda) &= \textstyle\sum_{v\in V} f_v(x_v) + \sum_{(u,v)\in E} \lambda_{u,v}^T(x_u - x_v) \\ &= \textstyle\sum_{v\in V} \left\{ f_v(x_v) + \left(\sum_{(v,w)\in E} \lambda_{v,w} - \sum_{(u,v)\in E} \lambda_{u,v}\right)^T x_v \right\}. \end{aligned}$$

Note that the x-step is now completely distributed. We can compute the minimizer with respect to x_v solely by knowing the Lagrange multipliers associated with the neighbors of v in G. The λ-step consists of the simple step

$$\lambda_{u,v} = \lambda_{u,v} + s(x_u - x_v)$$

Note that there are many decompositions of this form. Suppose we want to minimize $\sum_{v\in V} f_v(x)$. Here, some of the f_v might even be indicator functions of convex sets. We can rewrite this optimization problem to decouple the constraints in the from (3). This is completely equivalent to the original formulation, no matter what graph we choose, provided the graph is connected.

Note that if we used the augmented Lagrangian, this distributed decoupling would not work, as we cannot split the quadratic term. We'll discuss how to address this in the next lecture.

3.2 Utility Maximization

The general utility maximization problem is stated as

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} U_i(x_i) \\ \text{subject to} & Rx \leq c \end{array}$$

Here, each utility function represents some happiness or well-being for the ith user as a function of the amount of resource x_i. The inequalities are resource constraints, coupling the amount of utility available to each user.

Lets rewrite this in our standard form

$$\begin{array}{ll} \text{minimize} & -\sum_{i=1}^{n} U_i(x_i) \\ \text{subject to} & Rx - c + s = 0 \\ & s \geq 0 \end{array}$$

Then our Lagrangian becomes

$$\mathcal{L}(x, s, \lambda) = \sum_{i=1}^{n} -U_i(x_i) + \lambda^T (Rx - c + s)$$

Minimizing with respect to $s \geq 0$ shows that $\lambda \geq 0$. Otherwise, the Lagrangian is unbounded below. Thus, our dual problem is equivalent to

$$\max_{\lambda \geq 0} \min_{x} \sum_{i=1}^{n} -U_i(x_i) + \lambda^T (Rx - c)$$

The x-step can be done in a distributed fashion with each user maximizing

$$U(x_i) - \left[\sum_{j=1}^{p} R_{ji} \lambda_j \right] x_i$$

Then we can update λ by the rule

$$\lambda \leftarrow [\lambda + \alpha(Rx - c)]_+ \ .$$

The dynamics of this model are very interesting. λ can be interpreted as *prices* for a particular resource. If the prices are large, users get negative cost for acquiring more of their quantity x_i. If the resource constraints are loose, then the prices go down. If they are violated, then the prices go up.

3.3 Linear and Quadratic Programming

Consider the problem

$$\begin{array}{ll} \text{minimize} & c^T x + \frac{1}{2} x^T Q x \\ \text{subject to} & \ell \leq x \leq u \\ & Ax = b \end{array}$$

That is, we aim to solve a box-constrained quadratic program. Note that if $\ell = 0$ and $u = \inf$ and $Q = 0$, this is a standard form linear program.

The augmented Lagrangian for this problem is

$$\mathcal{L}(x, \lambda) = c^T x + \tfrac{1}{2} x^T Q x + \lambda^T (Ax - b) + \tfrac{\alpha_k}{2} \|Ax - b\|^2$$

and the x-step reduces to

$$\text{minimize}_{\ell \leq x \leq u} c^T x + \tfrac{1}{2} x^T Q x + \lambda^T (Ax - b) + \tfrac{\alpha_k}{2} \|Ax - b\|^2$$

which you could solve via the projected gradient method, or something similar.