Short Version of Convex Optimization

Virtually all content taken from Ryan Tibshirani's excellent 10-725 slides

Dan Schwartz

10-701 Recitation: 2017-02-09

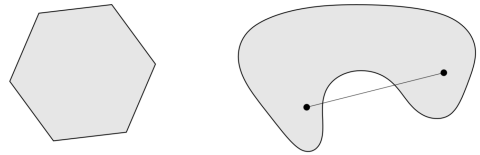
Outline

Motivation

Convex optimization problems can be understood and solved using generic algorithms, while non-convex problems are (frequently) more difficult to understand and solve.

Outline

Convex Sets



Definition

A set $C \subseteq \mathbb{R}^n$ is convex if for any $x, y \in C$ and $t \in [0, 1]$, we have:

$$tx + (1 - t)y \in C$$

Examples

- empty set
- lines, line segments
- affine spaces $(\{Ax + b : x \in \mathbb{R}^n\})$
- hyperplane $(\{x : a^\top x = b\})$, halfspace $(\{x : a^\top x \leq b\})$
- norm ball: $(\{x : \|x\| \leq t\})$ for some norm $\|\cdot\|$

Convex Sets

Additional Definitions

A convex combination of a set of points $x_1, ..., x_k \in \mathbb{R}^n$ takes the form:

$$\theta_1 x_1 + \theta_2 x_2 + ... + \theta_k x_k, \text{ where } \theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$$

A convex hull of a set C is the set of all convex combinations in C:

$$\text{conv}(C) = \left\{ \sum_{i=1}^{k} \theta_i x_i : x_i \in C, \theta_i \geq 0 \text{ for } i = 1, ..., k, \text{ and } \sum_{i=1}^{k} \theta_i = 1 \right\}$$

- simplex is a convex hull of points that are affinely independent ($x_1 - x_0, ..., x_k - x_0$ are linearly independent)
- probability simplex is a convex hull of standard basis vectors in $\mathbb{R}^n$ which can be written as:

$$\{\theta : \theta \geq 0, \mathbf{1}^\top \theta = 1\}$$

Convex Sets

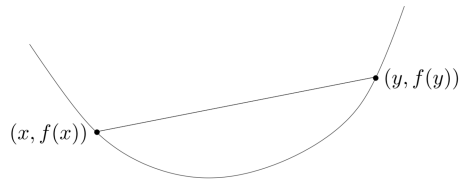
Operations Preserving Convexity

While a set can be verified as convex from the definition, it is sometimes easier to recognize that a set is convex because it can be formed from known convex sets using operations that preserve convexity.

- The intersection of any number of convex sets is convex
- affine images and affine preimages of convex sets are convex: if $f : \mathbb{R}^n \to \mathbb{R}^m$ can be written $f(x) = Ax + b$ and $S \subseteq \mathbb{R}^n$, $T \subseteq \mathbb{R}^m$ then: $f(S) = \{f(x) : x \in S\}$ is convex and $f^{-1}(T) = \{x : f(x) \in T\}$ is convex. Scaling and translation are special cases.

Outline

Convex Functions



Definition

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain $\mathrm{dom}(f)$ is convex, and for any $x, y \in \mathrm{dom}(f)$ and $t \in [0, 1]$,

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

I.e. the function lies below the line segment joining its evaluations at x, y

Convex Functions

Additional Definitions

- A function is strictly convex if this inequality is strict for $x \neq y$ and $t \in (0, 1)$.
- A function f is concave or strictly concave if $-f$ is convex or strictly convex respectively
- Affine functions, $f(x) = a^\top x + b$ are both convex and concave (and conversely, if a function is both convex and concave, it is affine).
- A function f is strongly convex with parameter m (m-strongly convex) if

$$f(x) - \frac{m}{2}\|x\|_2^2$$

  is convex. I.e. the function is more convex than a quadratic
- strong convexity $\implies$ strict convexity $\implies$ convexity

Convex Functions

Examples and Important Convex Functions

- Any norm is convex (can show from the triangle inequality)
- Indicator function $f(x) = I_C(x)$ of a convex set $C \subseteq \mathbb{R}^n$:

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

- quadratic function $f(x) = \frac{1}{2}x^\top Qx + c^\top x + b, Q \succeq 0$
  Note: $Q \succ 0 \iff$ strict convexity.
  Note: When $p > n, A^\top A \not\succ 0 \implies$ not strictly convex.
- least squares: $f(x) = \|Ax - b\|_2^2 = x^\top A^\top Ax - 2b^\top Ax + b^\top b$ is a special case of quadratic, hence convex
- $f(x) = \max\{x_1, ...x_n\}$

Convex Functions

Properties

- Continuous on relative interior of domain
- A function is convex iff its restriction to a line is convex:
  $f : \mathbb{R}^n \to \mathbb{R}$ is convex $\iff$ $g(t) = f(x + tv)$ is convex in t,
  where $\{t \in \mathbb{R} : x + tv \in \mathrm{dom}(f)\}, v \in \mathbb{R}^n$

First Order Characterization

If f is differentiable and ∇f is the gradient of f, then:
f is convex $\iff$ $\mathrm{dom}(f)$ is convex, and for all $x, y \in \mathrm{dom}(f)$:

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x)$$

I.e. the function is always above the first order linear Taylor approximation. For strict convexity the inequality is made strict for all $x \neq y$.

What happens if $\nabla f(x) = 0$? We get a minimizer! If f is strictly convex, the minimizer is unique!

From the first order convexity characterization, we can immediately write the first order optimality condition:

First Order Optimality

For convex, differentiable f, a feasible point $x \in C$ (with C the set of feasible points (coming later)) is optimal if and only if

$$\nabla f(x)^\top (y - x) \geq 0, \quad \forall y \in C$$

If $C = \mathbb{R}^n$ (i.e., the problem is unconstrained), this reduces to $\nabla f(x) = 0$.

Second Order Characterization

If f is twice differentiable and $\nabla^2 f$ is the Hessian of f, then:
f is convex $\iff$ $\mathrm{dom}(f)$ is convex, and for all $x \in \mathrm{dom}(f)$:

$$\nabla^2 f(x) \succeq 0$$

Note that $\nabla^2 f(x) \succ 0$ for all $x \in \mathrm{dom}(f)$ implies strict convexity, but the converse is not true.

Second Order Optimality

If f (convex or not) is twice differentiable at a feasible point x and $\nabla^2 f$ is the Hessian of f, then a necessary (but not sufficient) condition for x to be optimal is:

$$\nabla^2 f(x) \succeq 0$$

.

This is the analog of the second derivative being positive for a univariate function. It tells you that the function is curving up.

Strong Convexity Characterizations

If f is differentiable then m-strong convexity is equivalent to $\mathrm{dom}(f)$ is convex and:

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{m}{2}\|y - x\|_2^2 \ \forall x, y \in \mathrm{dom}(f)$$

I.e. f is lower bounded by its second order Taylor approximation.
If f is twice differentiable, then m-strong convexity is equivalent to $\mathrm{dom}(f)$ being convex and $\nabla^2 f(x) \succeq mI \ \forall x \in \mathrm{dom}(f)$

Convex Functions

Operations That Preserve Convexity

Often it is easiest to show that a function is convex by building it up from other functions using operations that preserve convexity.

- non-negative linear combinations: $a_1 f_1 + ... + a_n f_n, a_i \geq 0, f_i$ convex
- affine composition: $g(x) = f(Ax + b), f$ convex
- pointwise maximum: $f(x) = \max\{f_1(x), ...f_n(x)\}, f_i convex$ (also true for (uncountably) infinitely many functions)

Convex Functions

Operations That Preserve Convexity (Continued)

- partial minimization: If $f(x, y)$ is convex in x, y and C is a convex, non-empty set, then:

$$g(x) = \min_{y \in C} f(x, y)$$

is convex in x provided $g(x) > -\infty$ for all x in its domain,

$$\mathrm{dom}(g) = \{x : (x, y) \in \mathrm{dom}(f) \text{ for some } y \in C\}$$

- composition: This is a little tricky. Suppose $f = h \circ g, g : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R} \to \mathbb{R}$:
  - f is convex if h is convex and nondecreasing, g is convex
  - f is convex if h is convex and nonincreasing, g is concave
  - f is convex if h is concave and nondecreasing, g is concave
  - f is convex if h is concave and nonincreasing, g is convex

Convex Functions

Showing Convexity: Maximum distance to arbitrary set C

The maximum distance to an arbitrary set C is convex:

$$f(x) = \max_{y \in C} \|x - y\|$$

Why?

$\|x - y\|$ is convex in x for any fixed y, so we can view $f(x)$ as the pointwise maximum of an infinite number of functions (1 for each y), each of which is convex.

Convex Functions

Showing Convexity: Weighted least squares

Weighted least squares is convex as a function of the weights

$$f(w) = \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} w_i (y_i - x_i^\top \beta)^2$$

is concave with domain
$\mathrm{dom}(f) = \left\{ w : \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} w_i (y_i - x_i^\top \beta)^2 > -\infty \right\}$.

Why?

$\sum_{i=1}^{n} w_i (y_i - x_i^\top \beta)^2$ for fixed β is affine, and therefore concave, in w, so pointwise minimum f is also concave

Outline

Optimization Problem

Takes the form:

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \ldots m \\ & l_j(x) = 0, \quad j = 1, \ldots r \end{aligned}$$

$D = \mathrm{dom}(f) \cap \bigcap_{i=1}^{m} \mathrm{dom}(h_i) \cap \bigcap_{j=1}^{r} \mathrm{dom}(l_j)$, the common domain of all functions. f is the criterion or objective. A feasible point, $x \in D$ is a point such that all inequality (h_i) and equality constraints (l_j) are met. A solution or minimizer x^* is a feasible point that achieves the minimum criterion value. The minimum criterion value is often denoted f^*

Optimization

Convex Optimization Problem

A convex optimization problem is an optimization problem in which all functions, f, h_i are convex and all functions l_j are affine. Thus it can be written:

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \ldots m \\ & Ax = b \end{aligned}$$

Why affine equality constraints?

We could rewrite these by separating them into two constraints $(l_j(x) \leq 0, -l_j(x) \leq 0)$ that must both hold. Therefore $l_j(x)$ must be both convex and concave to make both of these conditions convex, which forces $l_j(x)$ to be affine.

Note that this can equivalently be written as a concave maximization:

$$\begin{aligned} \max_{x \in D} \quad & -f(x) \\ \text{subject to} \quad & -h_i(x) \geq 0, \quad i = 1, \ldots m \\ & Ax = b \end{aligned}$$

Convex Optimization

Example: Regularized Logistic Regression

For $y_i \in \{0, 1\}, x_i \in \mathbb{R}^p, i = 1, ..., n$, the ℓ_1 regularized logistic regression problem is:

$$\begin{aligned} \max_{\beta \in \mathbb{R}^p} \quad & \prod_{i=1}^{n} \left(\frac{\exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)} \right)^{y_i} \cdot \left(\frac{1}{1 + \exp(x_i^\top \beta)} \right)^{1 - y_i} \\ \text{subject to} \quad & \|\beta\|_1 \leq t \end{aligned}$$

Taking the log and flipping the sign gives us:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^p} \quad & \sum_{i=1}^{n} \left(-y_i (x_i^\top \beta) + \log \left(1 + \exp(x_i^\top \beta) \right) \right) \\ \text{subject to} \quad & \|\beta\|_1 \leq t \end{aligned}$$

Convex Optimization

Key Properties

- Any local minimizer is a global minimizer (caution: this does not mean the minimum is unique, consider $f(x) = c, c$ some constant)
- The set of solutions forms a convex set. Why? Suppose we have two solutions: x, z. Then because the constraints are convex $tx + (1-t)z$ is also feasible for $t \in [0,1]$. By convexity of the criterion, $f(tx + (1-t)z) \leq tf(x) + (1-t)f(z) = f^*$, which means $tx + (1-t)z$ is also a solution.
- A strictly convex criterion f does have a unique solution. Assuming two solutions, this follows the same argument as above, but now we have $f(tx + (1-t)z) < tf(x) + (1-t)f(z) = f^*$, which is a contradiction
- Sometimes nonconvex problems can be reduced to convex problems

Outline

Gradients

Intuition

Gradients provide the following information for each coordinate of our space:
If our input moves in the positive direction of that coordinate, how much will the value of the function change?

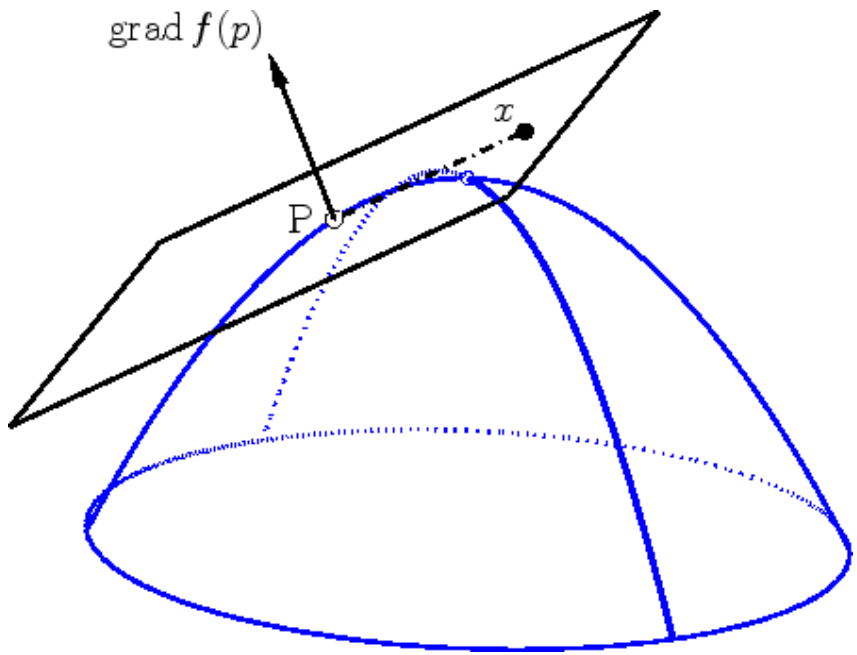
$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n} \right)^\top$$

If our input moves just in a coordinate where the magnitude of the gradient is small, the value of the function will change less than if our input moves just in a coordinate where the magnitude of the gradient is large.

Gradients

Intuition

The gradient ∇f is orthogonal to a hyperplane that is tangent to the function at the point x. Think of how circles on this tangent hyperplane centered at x will look if we project them onto the input hyperplane (i.e. our input space) – where the contours are close together, the function is changing quickly. In which direction will these contours be closest?

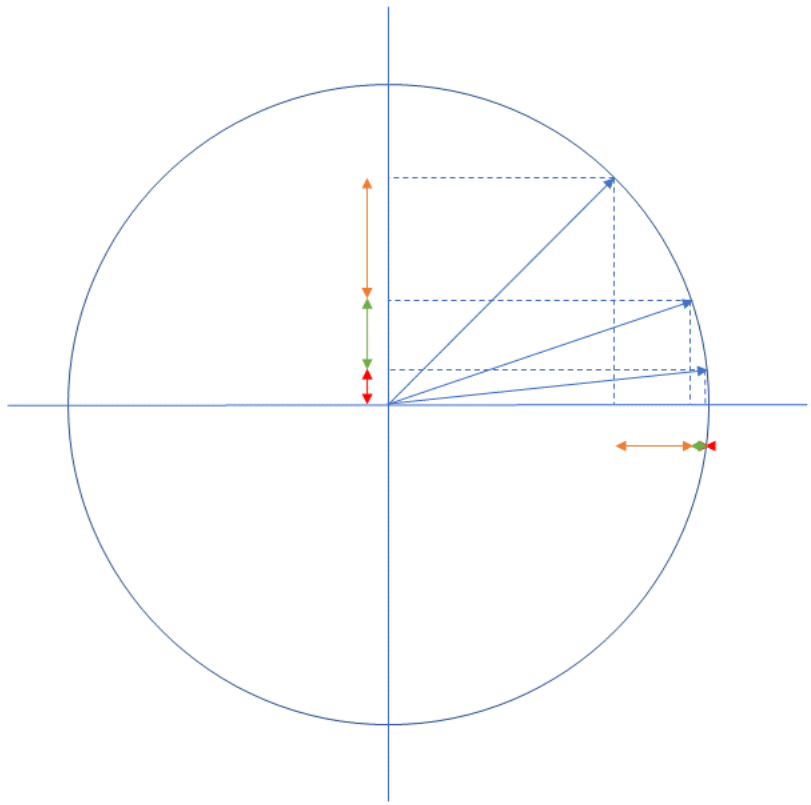


Gradients

Intuition

If we have a budget for how much distance our input can move, which direction will increase the function value most?

- What if we just move along the one component of maximum change?
- We can get more increase by moving in multiple components. Why? We can trade a little movement in one component for a lot of movement in a different component
- The ideal direction is exactly the direction of the gradient



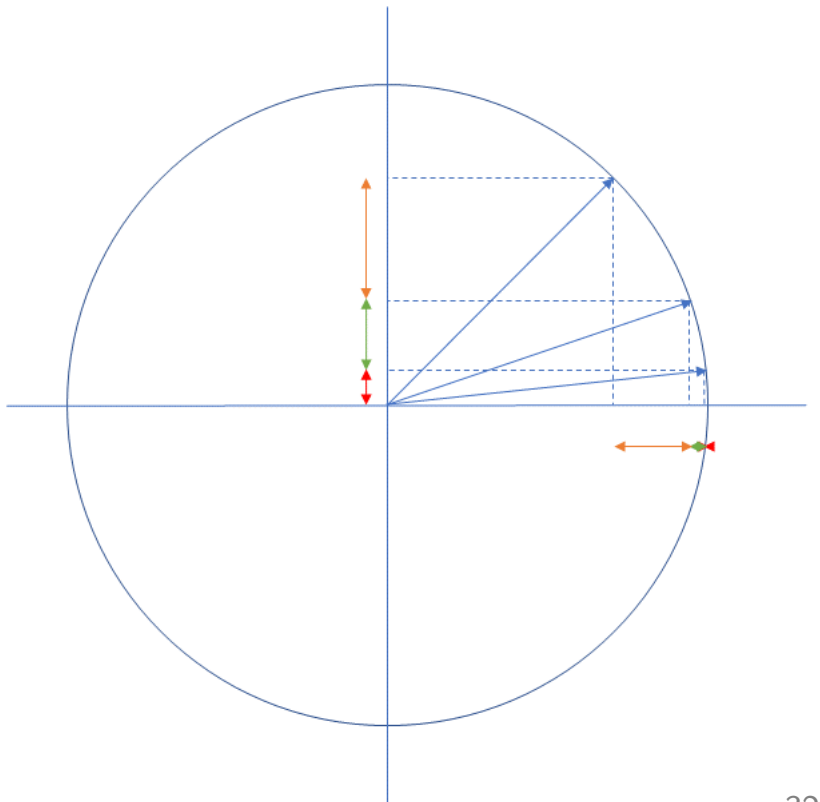
Gradients

Intuition

We can show this algebraically. Letting $u \in \mathbb{R}^n, \|u\|_2 = 1$ be a unit vector, we can ask how much the function changes along this unit vector by projecting it onto the gradient $\nabla f(x)$ since the gradient gives us exactly the rate of change in each component. We then want to maximize this change:

$$\max_{\|u\|_2=1} \frac{\langle \nabla f(x), u \rangle}{\|\nabla f(x)\|_2} = \max_{\|u\|_2=1} u \cos(\theta)$$

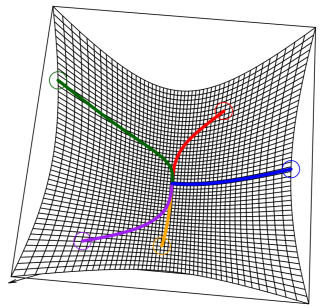
This is maximized when the angle θ between u and $\nabla f(x)$ is 0 since that gives $\cos(\theta) = 1$



Outline

Gradient Descent

If we can't solve for a minimum analytically (or if the analytical solution is computationally too expensive) and f is differentiable over the subspace of interest, a reasonable strategy is to iteratively move in the direction in which the function is decreasing the fastest, i.e. the negative gradient.



- Start at some initial point $x^{(0)}$
- At iteration k, take a step of step size t_k in the direction of the negative gradient:

$$x^{(k)} \leftarrow x^{(k-1)} - t_k \cdot \nabla f\left(x^{(k-1)}\right)$$

- Stop at some point (e.g. $\|\nabla f(x)\|_2^2 < \epsilon$, approximately stationary)

Gradient Descent

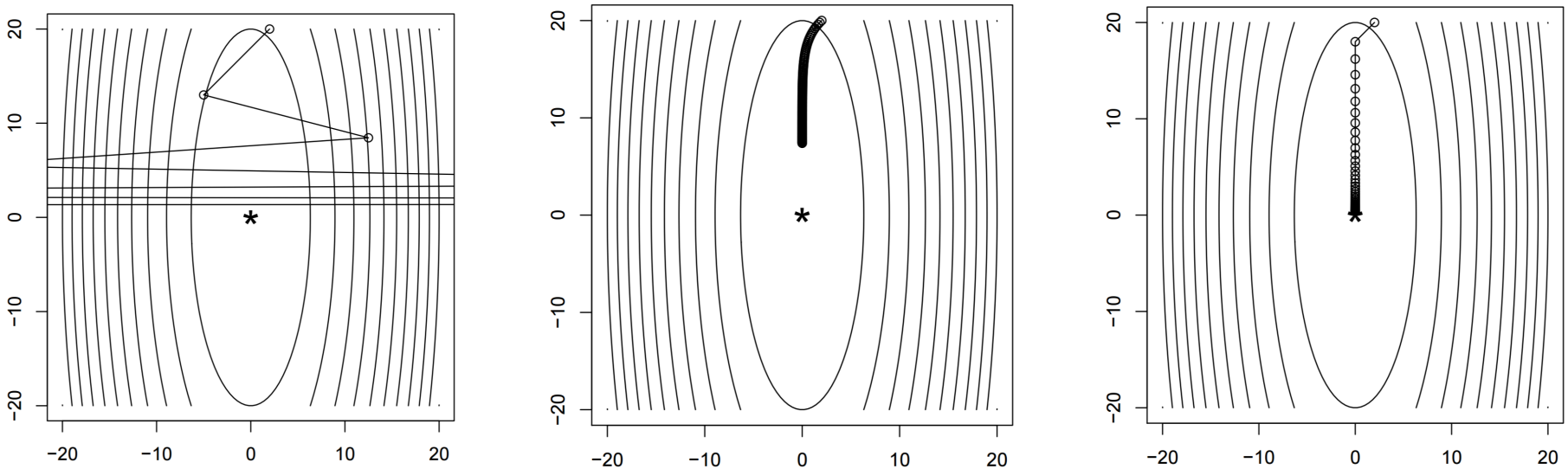
Stationary Point

Is a stationary point optimal? Is it unique?

- For nonconvex functions?
- For convex functions?
- For strictly convex functions?

Gradient Descent

Step Size



Step size is key to making gradient descent work. Too large and the method can diverge, too small and it might never converge. Several strategies:

- fixed step size
- exact line search (usually not worth expense)
- backtracking line search

Outline

Backtracking Line Search

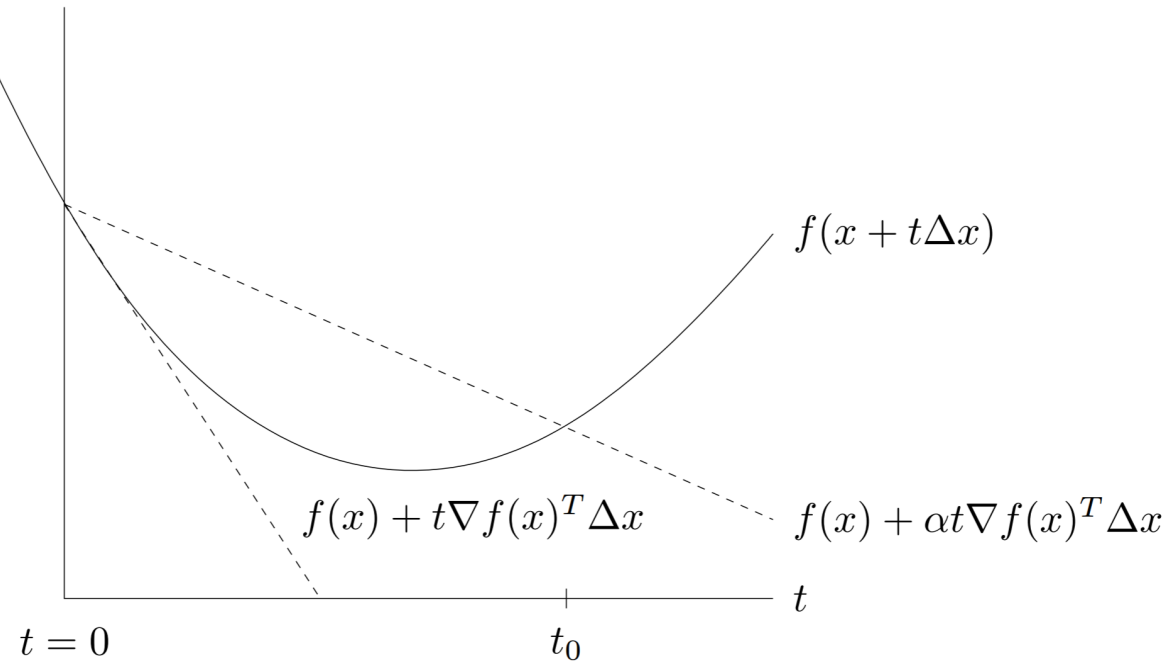
Algorithm

- Fix parameters $0 < \beta < 1, 0 < \alpha \leq \frac{1}{2}$
- At each iteration k, set $t \leftarrow t_{init}$
- Repeat until we set $x^{(k)}$:
  - Set $x^{(candidate)} \leftarrow x^{(k-1)} - t\nabla f(x^{(k-1)})$
  - If $f(x^{(candidate)}) > f(x^{(k-1)}) - \alpha t \left\| \nabla f(x^{(k-1)}) \right\|_2^2$:
    Shrink $t \leftarrow \beta t$
    Else
    Set $x^{(k)} \leftarrow x^{(candidate)}$

Often people just set $\alpha = \beta = \frac{1}{2}$

Backtracking Line Search

Intuition



This says that our step size should be such that the new value is under the reduced tangent line. It prevents step sizes from being too large. We take the biggest step we can that is under this line.

Outline

Stochastic Gradient Descent

Algorithm

Consider a problem of the form $\min_x \sum_{i=1}^n f_i(x)$. For example least squares $\min_\beta \sum_{i=1}^n (y_i - z_i^\top \beta)^2$ falls into this category. At each iteration, normal gradient descent would do:

$$x^{(k)} \leftarrow x^{(k-1)} - t_k \cdot \nabla \sum_{i=1}^n f_i(x^{(k-1)})$$

$$= x^{(k-1)} - t_k \sum_{i=1}^n \nabla f_i(x^{(k-1)})$$

Stochastic gradient descent instead uses a noisy estimate of the gradient:

$$x^{(k)} \leftarrow x^{(k-1)} - t_k \cdot \nabla f_{i_k}(x^{(k-1)}), \quad i_k \in \{1, ..., n\}$$

Stochastic Gradient Descent

Properties

- Only part of the data set needs to be in memory
- n iterations is approximately equal (computationally) to 1 iteration of gradient descent.
- Tends to make great progress far from the optimum, struggles to make progress near the optimum
- Good if you only need a noisy estimate of the optimum
- 'Minibatches' trade off between standard gradient descent and stochastic gradient descent. A random subset of the data is used at each iteration.

Outline

Subgradients

Definition

Recall the first order characterization of convexity for a differentiable $f : \mathbb{R}^n \to \mathbb{R}$:

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x)$$

This says that a linear approximation always underestimates f. A subgradient of any function f (differentiable or not, convex or not) at x is defined as any $g \in \mathbb{R}^n$ such that:

$$f(y) \geq f(x) + g^\top (y - x)$$

for all $y \in \mathrm{dom}(f)$

Subgradients

Properties

- Subgradients always exist for convex functions
- If f is convex and differentiable at x, then $g = \nabla f(x)$ is the unique subgradient at x
- For nonconvex functions f, the subgradient need not exist even when f is differentiable
- The set of all subgradients of f at x is called its subdifferential denoted

$$\partial f(x) = \{g : g \text{ is a subgradient of } f \text{ at } x\}$$

- The subdifferential is closed and convex. For a convex f with x in the relative interior of $\mathrm{dom}(f)$, the set $\partial f(x)$ is non-empty. If f is convex and differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$. Converesely, if f is convex and $\partial f(x) = \{g\}$, then f is differentiable at x and $\nabla f(x) = g$

Subgradients

Examples

- $f(x) = |x|$. When $x \neq 0$, $\partial f(x) = \{\text{sign}(x)\}$. When $x = 0, \partial f(x) = [-1, 1]$
- $f(x) = \|x\|_2$. When $x \neq 0$, $\partial f(x) = \{\frac{x}{\|x\|_2}\}$. When $x = 0, \partial f(x) = \{z : \|z\|_2 \leq 1\}$
- $f(x) = \|x\|_1$. When $x_i \neq 0$, a subgradient g has unique *ith* component $g_i = \text{sign}(x_i)$. When $x_i = 0$, $g_i \in [-1, 1]$

Subgradient Descent

Just like gradient descent, but replace gradient with any subgradient.

- Start at some initial point $x^{(0)}$
- At iteration k, take a step of step size t_k in the negative direction of any subgradient:

$$x^{(k)} \leftarrow x^{(k-1)} - t_k \cdot g^{(k-1)}, \quad g^{(k-1)} \in \partial f(x^{(k-1)})$$

Not necessarily a descent method, save the best answer:

$$f(x_{best}^{(k)}) = \min_{0,\ldots k} f(x^{(k)})$$

Converges for diminishing step sizes

Gradient Descent

Additional Variations

- Acceleration: Use some 'momentum' from steps at previous iterations to move in a better direction
- Proximal gradient: Handle constrained optimization more generally
- ...

Newton's Method

Quick Idea

Based on the same algorithm as Newton's root finding algorithm, but now we are finding the roots of the gradient. Root finding algorithm:

$$x^{(k)} \leftarrow x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}$$

Optimization method:

$$x^{(k)} \leftarrow x^{(k-1)} - \left(\nabla^2 f(x^{(k-1)})\right)^{-1} \nabla f(x^{(k-1)})$$

- Requires doubly smooth f
- Convergence rate (in iterations) is much faster than first order methods: $O(\log(\log \frac{1}{\epsilon}))$
- Can be expensive to compute $\nabla^2 f$
- Quasi-Newton methods approximate the Hessian

Optimization Techniques

Take 10-725!

	Gradient Desc.	**Subgrad. Desc.**	**Proximal Gradient**	**Newton**	**Quasi-Newton**
Criterion	smooth f	any f	smooth + simple $f = g + h$	doubly smooth f	doubly smooth f
Constraints	projection onto constr. set	projection onto constr. set	prox operator	equality constraints	unconstr.
(Opti) Parameters	fixed step/ line search	diminishing step	fixed step/ line search	pure step/ line search	line search
Iteration Cost	cheap	cheap	moderately cheap	moderate to expensive	moderately cheap
Rate	$O(\frac{1}{\epsilon})$, $O(\frac{1}{\sqrt{\epsilon}})$ with acceleration, $O(\log(\frac{1}{\epsilon}))$ strong conv.	$O(\frac{1}{\epsilon^2})$	$O(\frac{1}{\epsilon})$, $O(\frac{1}{\sqrt{\epsilon}})$ with acceleration, $O(\log(\frac{1}{\epsilon}))$ strong conv.	$O(\log(\log\frac{1}{\epsilon}))$	superlinear

Optimization Techniques

Take 10-725!

	Barrier Method	**Primal-dual IPM**	**ADMM**	**Coordinate Desc.**
Criterion	doubly smooth f	doubly smooth f	block seperable $f(x,z) = g(x) + h(z)$	smooth + coordinate-wise separable
Constraints	doubly smooth ineq. constr.	doubly smooth ineq. constr.	eq. constr. (always), ineq. constr. (sometimes)	coordinate-wise separable constr.
(Opti) Parameters	fixed step/ line search, diverging barrier param.	line search, diverging barrier param.	fixed augmented Lagrang. param/ varied by iter.	none!
Iteration Cost	exp. to very exp.	mod. to exp.	cheap to exp.	cheap to exp.
Rate	$O(\log(\frac{1}{\epsilon}))$	$O(\log(\frac{1}{\epsilon}))$	unknown. like 1st order methods in practice	unknown. (freq) faster than 1st order methods in practice

Outline

Duality

Linear Version

Suppose we just want a lower bound on the optimal value f^*. I.e., we want to find $B \leq f^*$. For example:

$$\begin{aligned} \min_{x,y} \quad & x + y \\ \text{subject to} \quad & x + y \geq 2 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

$B = 2$ will work. Is this just because the criterion matches the constraint?

Duality

Linear Version

We can transform the constraints to look like the criterion. In the example below, multiply both sides of $y \geq 0$ by a positive number so we don't reverse the inequality, then add it to the first constraint.

$$\begin{aligned} \min_{x,y} \quad & x + 3y \\ \text{subject to} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned}$$

$$\begin{array}{rr} & x + \ y \geq 2 \\ + & 2y \geq 0 \\ \hline & x \ + 3y \geq 2 \end{array}$$

Now we have a new constraint that looks like our criterion, and we can read off the lower bound again: $B = 2$

Duality

Linear Version

We can always multiply the constraints by some scalars and add them together to get the criterion.

$$
\begin{array}{rl}
\min\limits_{x,y} & px + qy \\
\text{subject to} & x + y \geq 2 \\
 & x, y \geq 0
\end{array}
\qquad
\begin{array}{rrr}
 & ax & +ay \geq 2a \\
+ & & cy \geq 0 \\
+ & bx & \geq 0 \\
\hline
 & (a+b)x & +(a+c)y \geq 2a
\end{array}
$$

Now for $a + b = p$, $a + c = q$, we have that the lower bound is $B = 2a$

Duality

Linear Version

What's the best possible lower bound?

Primal

$$\begin{aligned} \min_{x,y} \quad & px + qy \\ \text{subject to} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} \max_{a,b,c} \quad & 2a \\ \text{subject to} \quad & a + b = p \\ & a + c = q \\ & a, b, c \geq 0 \end{aligned}$$

Duality

Linear Version

General form

Primal

$$\begin{aligned} \min_{x} \quad & c^\top x \\ \text{subject to} \quad & Ax = b \\ & Gx \leq h \end{aligned}$$

Dual

$$\begin{aligned} \max_{u,v} \quad & -b^\top u - h^\top v \\ \text{subject to} \quad & -A^\top u - G^\top v = c \\ & v \geq 0 \end{aligned}$$

Why?

$$\begin{aligned} u^\top(Ax - b) + v^\top(Gx - h) \leq & 0 \\ -(u^\top A + v^\top G)x \geq & -u^\top b - v^\top h \\ (-A^\top u - G^\top v)^\top x \geq & -b^\top u - h^\top v \end{aligned}$$

Outline

Duality

Lagrangian

The Lagrangian gives us a way to find a lower bound in general (not just for linear programs). For the problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, ..., m \\ & \ell_j(x) = 0, \quad j = 1, ..., r \end{aligned}$$

The Lagrangian is defined as:

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x), \quad u \geq 0$$

Note: We understand implicitly that $L(x, u, v) = -\infty$ if $u_i < 0$ for any u_i

Duality

Lagrangian

The Lagrangian is a lower bound for feasible x and dual feasible u, v (i.e. $u \geq 0$). Why? For feasible x, by definition we know

$$h_i(x) \leq 0, \quad \ell_j(x) = 0$$

Then, for any $u \geq 0, v$ we have:

$$u_i h_i(x) \leq 0, \qquad v_j \ell_j(x) = 0$$

$$\sum_{i=1}^{m} u_i h_i(x) \leq 0, \qquad \sum_{j=1}^{r} v_j \ell_j(x) = 0$$

Which implies:

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x) \leq f(x)$$

Duality

Lagrangian

If we let $C = \{x : h_i(x) \leq 0, i = 1, ..., m, \ell_j(x) = 0, j = 1, ..., r\}$ be the set of primal feasible points x, then we note that:

$$f^* \geq \min_{x \in C} L(x, u, v) \geq \min_x L(x, u, v)$$

Because if we minimize over all x instead of some x the value can only get lower. Therefore we define the Lagrange dual function

$$g(u, v) = \min_x L(x, u, v)$$

Which gives us a lower bound on f^* for any dual feasible u, v

Duality

Lagrangian

We now have a way to describe a dual problem for any primal problem (convex or not).

Primal

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, ..., m \\ & \ell_j(x) = 0, \quad j = 1, ..., r \end{aligned}$$

Dual

$$\begin{aligned} \max_{u,v} \quad & g(u, v) \\ \text{subject to} \quad & u \geq 0 \end{aligned}$$

Where

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x)$$

$$g(u, v) = \min_{x} L(x, u, v)$$

Duality

Properties

- $f^* \geq g^*$ always. This is called weak duality
- g is always concave even if f is not convex - so the dual problem is always a convex optimization problem.
- if $f^* = g^*$, then we have strong duality. This often holds for problems of interest

Deriving The Dual

SVM

Let's work an example. For SVM, the primal problem can be written as:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{subject to} \quad & (w^\top x_i + b) y_i \geq 1 \quad \forall i \in \{1, \ldots, n\} \end{aligned}$$

Then we can write our inequality constraints in standard form as:

$$h_i(w, b) = 1 - (w^\top x_i + b) y_i \leq 0$$

We don't have any equality constraints.

Deriving The Dual

SVM

Recall that the Lagrangian is simply adding our primal criterion and our inequality and equality constraints with the constraints multiplied by Lagrange multipliers. In general, for primal criterion f, primal variables β, inequality constraints h_i, inequality Lagrange multipliers u_i, equality constraints ℓ_j, and equality Lagrange multipliers v_j, the Lagrangian is:

$$L(\beta, u, v) = f(\beta) + \sum_{i=1}^{m} u_i h_i(\beta) + \sum_{j=1}^{r} v_j \ell_j(\beta), \quad u_i \geq 0$$

In our case, $\beta = (w, b)$ and the last term is 0. This gives us:

$$L(w, b, u) = \|w\|_2^2 + \sum_{i=1}^{n} u_i \left[1 - (w^\top x_i + b) y_i\right], \quad u_i \geq 0$$

Deriving The Dual

SVM

Now to get our dual criterion, we need to minimize over the primal variables:

$$g(u) = \min_{w,b} L(w, b, u) = \min_{w,b} \left\{ \|w\|_2^2 + \sum_{i=1}^{n} u_i \left[1 - (w^\top x_i + b) y_i \right] \right\}$$

How do we minimize? Take the derivatives, set them to 0!

$$0 = \frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} u_i y_i x_i$$

$$\implies w^* = \sum_{i=1}^{n} u_i y_i x_i$$

$$0 = \frac{\partial L}{\partial b} = -\sum_{i=1}^{n} u_i y_i$$

$$\implies \sum_{i=1}^{n} u_i y_i = 0$$

required for b optimal

Deriving The Dual

SVM

Now use what we know from the minimization. Plug in:

$$L(w^*, b^*, u) = \frac{1}{2}\|w^*\|_2^2 + \sum_{i=1}^{n} u_i \left[1 - ((w^*)^\top x_i + b^*) y_i\right]$$

$$\begin{aligned}
\frac{1}{2}\|w^*\|_2^2 &= \frac{1}{2}(w^*)^\top (w^*) \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} u_i y_i x_i\right)^\top \left(\sum_{j=1}^{n} u_j y_j x_j\right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j y_i y_j x_i^\top x_j
\end{aligned}$$

Deriving The Dual

SVM

Now use what we know from the minimization. Plug in:

$$L(w^*, b^*, u) = \frac{1}{2}\|w^*\|_2^2 + \sum_{i=1}^{n} u_i \left[1 - ((w^*)^\top x_i + b^*)y_i\right]$$

$$\sum_{i=1}^{n} u_i \left[1 - ((w^*)^\top x_i + b^*)y_i\right] = \sum_{i=1}^{n} \left[u_i - u_i y_i (w^*)^\top x_i + u_i y_i b^*\right]$$

$$= \sum_{i=1}^{n} u_i - \sum_{i=1}^{n} u_i y_i (w^*)^\top x_i + b^* \sum_{i=1}^{n} u_i y_i$$

$$= \sum_{i=1}^{n} u_i - \sum_{i=1}^{n} u_i y_i \left(\sum_{j=1}^{n} u_j y_j x_j\right)^\top x_i + 0$$

$$= \sum_{i=1}^{n} u_i - \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j y_i y_j x_i^\top x_j$$

Deriving The Dual

SVM

Putting terms together we have:

$$
\begin{aligned}
g(u) =& L(w^*, b^*, u) = \frac{1}{2}\|w^*\|_2^2 + \sum_{i=1}^{n} u_i \left[1 - ((w^*)^\top x_i + b^*) y_i\right] \\
=& \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j y_i y_j x_i^\top x_j - \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j y_i y_j x_i^\top x_j + \sum_{i=1}^{n} u_i \\
=& -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j y_i y_j x_i^\top x_j + \sum_{i=1}^{n} u_i
\end{aligned}
$$

Deriving The Dual

SVM

Then the dual problem is. Don't forget the constraints. u_i must be non-negative from the Lagrangian, and we derived the other:

$$\max_u \quad -\frac{1}{2}\sum_{i,j=1}^{n} u_i u_j y_i y_j x_i^\top x_j + \sum_{i=1}^{n} u_i$$

$$\text{subject to} \quad u_i \geq 0 \quad \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} u_i y_i = 0$$

$$\Longleftrightarrow$$

$$\min_u \quad \frac{1}{2}\sum_{i,j=1}^{n} u_i u_j y_i y_j x_i^\top x_j - \sum_{i=1}^{n} u_i$$

$$\text{subject to} \quad u_i \geq 0 \quad \forall i \in \{1, \ldots, n\}$$

$$\sum_{i=1}^{n} u_i y_i = 0$$