Lecture 3

DJM

9 October 2018

Why learn convex optimization?

- Many software packages have built-in optimizers
- This is fine in certain cases (I have a function! Throw it in there.)
- Will work "fine" some of the time.
- Immensely suboptimal most of the time

Questions to answer

- Is it convex?
- Can you take analytic derivatives?
- Can you solve them for zero?
- Is the problem small, low dimensional?
- Do you need to solve it only once?
- Can you find the analytic Hessian?

Built-in optimizers are good if most of these answers are yes, or if they are all no.

In other cases, it's better to write custom code.

Knowing a little about optimization can reveal properties of the solutions.

Most packages do many things

```
head(optim)
```

```
##
## 1 function (par, fn, gr = NULL, ..., method = c("Nelder-Mead",
## 2     "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"), lower = -Inf,
## 3     upper = Inf, control = list(), hessian = FALSE)
## 4 {
## 5     fn1 <- function(par) fn(par, ...)
## 6     gr1 <- if (!is.null(gr))
```

Classes of algorithms

- General purpose (Simulated Annealing, Genetic Algorithms)
- First order
- Second order
- Constrained/unconstrained?
- Linear/non-linear?
- Convex/Non-convex?

We're going to talk about first-order methods for convex problems.

Convex sets and functions

Definitions

Thanks: Much of this material is borrowed/copied from Ryan Tibshirani.

Set C is *convex* iff $\forall x, c \in C, \forall t \in [0;1] \;\; tx + (1-t)y \in C$.

So C is convex iff for any two points in C their segment is also entirely in C.

Convex combination of set of points $x_1, \ldots, x_k \in \mathbb{R}^n$ is

$$\left\{ \sum_{i=1}^{k} \Theta_i x_i : \sum_{i=1}^{k} \Theta_i = 1, \; \forall i \; \Theta_i \in [0;1] \right\}.$$

Convex hull of any $C \in \mathbb{R}^n$, denoted $conv(C)$ is a union of all convex combinations of different elements of C.

Some examples

- Empty set, point, line, segment.
- Norm ball: $\{x : \|x\| < r\}$.
- Hyperplane $\left\{x : a^\top x = b\right\}$, Affine space $\{x : Ax = b\}$.
- Hyperspace: $\left\{x : a^\top x \leq b\right\}$, Polyhedron $\{x : Ax \leq b\}$.
- Cone such that if $x_1, x_2 \in C$ then $t_1 x_1 + t_2 x_2 \in C \; \forall t_1, t_2 \geq 0$.

Cones

Set C is a *cone* iff $\forall t \geq 0, \; x \in C \implies t^\top x \in C$.

Type of cones:

- Norm cone: $\{(x,t) : \|x\| \leq t\}$.
- Normal cone for some C and $x \in C$: $N_C(x) = \left\{g : g^\top x \geq g^\top y \; \forall y \in C\right\}$.
- Positive semidefinite cone $S_+^n = \{x \in S^n : x \succeq 0\}$, S^n is Hilbert space.

Key properties of convex sets

- Separating hyperplane. A, B are convex, nonempty, disjoint. Then $\exists a, b : \; A \subseteq \left\{x : a^\top x \leq b\right\}, B \subseteq \left\{x : a^\top \geq b\right\}$.
- Supporting hyperplane. C nonempty, convex, $x_0 \in boundary(C)$. Then $\exists a : \; C \subseteq \left\{x : a^\top x \leq a^\top x_0\right\}$.

Operations preserving convexity

- Intersection.
- Scaling, translation. C is convex $\implies aC + b$ is convex.
- Affine image and preimage. $f(x) = Ax + b$, C is convex $\implies f(C), f^{-1}(C)$ are convex.
- Lots more (See (Boyd and Vandenberghe 2004), chapter 2).

$A_1, \ldots, A_k, B \in \mathbb{S}^n$ – symmetric matrices. Then $C = \left\{ x \in \mathbb{R}^k : \sum_{i=1}^{k} x_i A_i \preceq B \right\}$.

$f : \mathbb{R}^k \to \mathbb{S}^n$, $f(x) = B - \sum_{i=1}^{k} x_i A_i$. $C = f^{-1}(S_+^n)$ – affine preimage of convex cone.

Convex functions

Function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* iff $\operatorname{dom}(f) \subseteq \mathbb{R}^n$ is convex and

$$\forall x, y \in \operatorname{dom}(f), t \in [0;1] \quad f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Other definitions:

- f is *concave* iff $-f$ is convex.
- f is *strictly convex* iff $\forall t \in (0;1)$ the inequality in definition is strict.
- f is *strongly convex* with parameter τ iff $f(x) - \frac{\tau}{2}\|x\|_2^2$ is convex.

Examples

- $f(x) = \frac{1}{x}$ is strictly convex, but not strongly.
- Univariate functions:
  - e^{ax} is convex $\forall a \in \mathbb{R}$ over $\mathbb{R}$.
  - x^a convex given $a \geq 1$ or $a \leq 0$ over $\mathbb{R}_+$.
  - $\log x$ is concave over $\mathbb{R}_+$.
- Affine $a^\top x + b$ is both convex and concave.
- Quadratic $\frac{1}{2}x^\top Q x + b^\top x + c$ is convex if $Q \succeq 0$.
- $\|u - Ax\|_2^2$ convex since $A^\top A \succeq 0$.
- Norms: all vector norms and most matrix norms are convex.
- Indicator function is convex. C is a convex set, then $I_C(x) = \begin{cases} 0, & x \in C \\ \infty, otherwise \end{cases}$.
- Support function is convex $\forall C$. $I_C^*(x) = \max_{y \in C} x^\top y$.

Key properties

- f is convex iff its epigraph is convex, where $epi(f) = \{(x,t) \in \operatorname{dom}(f) \times \mathbb{R} : f(x) \leq t\}$.
- f is convex $\implies$ all its sublevel sets are convex. $C_t = \{x \in \operatorname{dom}(f) : f(x) \leq t\}$. The converse is false.
- Assume f is differentiable. Then f is convex iff $\operatorname{dom}(f)$ is convex and $\forall x, y \in \operatorname{dom}(f) \quad f(y) \geq f(x) + \nabla f(x)^\top (y - x)$. Essentially, it means that f's graph is above any tangent plain.
- Assume f is twice differentiable. f is convex iff $\operatorname{dom}(f)$ is convex and $\forall x \in \operatorname{dom}(f) \quad \nabla^2 f(x) \succeq 0$.

Operations preserving function convexity

- Nonnegative linear combination.
- Pointwise maximum. $\forall s \in S \;\; f_s$ is convex $\implies f(x) = \max\limits_{S} f_s(x)$ is also convex.
- Partial minimum. $g(x, y)$ convex over variables x, y; C convex. Then $f(x) = \min\limits_{y \in C} g(x, y)$ is also convex. E.g., $f(x) = \max\limits_{y \in C} \|x - y\|$ or $f(x) = \min\limits_{y \in C} \|x - y\|$.

Terminology

Optimization problem

A convex optimization problem (program)

$$\begin{array}{rl} \min\limits_{x \in D} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad \forall i \in [1:m] \\ & Ax = b \end{array} \tag{1}$$

where f, g_i are convex and $D = \operatorname{dom}(f) \cap \operatorname{dom}(g_i)$.

Terms

$$\begin{array}{rl} \min\limits_{x \in D} & f(x) \\ \text{subject to} & g_i(x) \leq 0 \quad \forall i \in [1:m] \\ & Ax = b \end{array} \tag{2}$$

- f – criteria or objective function.
- g_i – inequality constraints.
- x is a *feasible point* if it satisfies the conditions, namely $x \in D$, $g_i(x) \leq 0$, and $Ax = b$.
- $\min f$ over feasible points points – *optimal value* f^*.
- If x is feasible and $f(x) = f^*$ then x is an *optimum* (solution, minimizer).
- Feasible x is a *local optimum* if $\exists R > 0$ such that $\forall y \in B_R(x) \;\; f(x) \leq f(y)$.
- If x is feasible and $f(x) \leq f^* + \varepsilon$ then x is *ε-suboptimal*.
- If x is feasible and $g_k(x) = 0$ then g_k is *active* at x (otherwise inactive).

Properties

- Solution set X_{opt} is convex.
- If f is strictly convex then the solution is unique.
- For convex optimization problems all local optima are global.
- The set of feasible points is convex.

Example: Lasso.

$\min\limits_{\beta} \|y - X\beta\|_2^2$ subject to $\|\beta\|_1 \leq s$.

- $g_1(\beta) = \|\beta\|_1 - s$ – convex, no equality constraints.
- X is $n \times p$ matrix

– If $n \geq p$ and X is full rank then $\nabla^2 f(\cdot) = 2X^\mathsf{T}X$ is positive definite matrix. The function is strictly convex, therefore the solution is unique.

– If $p > n$ then $\exists \beta \neq 0$ such that $X\beta = 0 \implies$ multiple solutions.

First Order Condition

- convex problem with differentiable f
- a feasible x is optimal iff $\nabla f(x)^T(x-y) \geq 0$, $\forall$ feasible y
- if unconstrained, the condition reduces to $\nabla f(x) = 0$

$$\min_x \frac{1}{2}x^T Q x + b^T x + c, \qquad Q \succeq 0$$

- FOC: $\nabla f(x) = Q^T x + b = 0$
- if $Q \succ 0 \rightarrow x^* = -Q^{-1}b$
- if Q singular, $b \notin Col[Q] \rightarrow$ no solution
- if Q singular, $b \in Col[Q] \rightarrow x^* = -Q^* b + z$ with $z \in null[Q]$

Useful operations

Partial optimization

Recall: $h(x) = \min_{y \in C} f(x,y)$ is convex if f is convex, and C is convex.

$$\begin{array}{lll} \min\limits_{x_1,x_2} & f(x_1,x_2) & \min\limits_{x_1} \quad \tilde{f}(x_1) \\ \text{s.t.} & g_1(x_1) \leq 0 & \Longleftrightarrow \text{s.t.} \quad g_1(x_1) \leq 0 \\ & g_2(x_2) \leq 0 & \end{array} \tag{3}$$

where $\tilde{f}(x_1) = \min\{f(x_1,x_2) : g_2(x_2) \leq 0\}$.

- The right problem is convex if the left is (and vice versa)

Transformations

- We can use a monotone increasing transformation $h : \mathbb{R} \rightarrow \mathbb{R}$:

$$\min_{x \in C} f(x) \Rightarrow \min_{x \in C} h(f(x))$$

- We can use a change of variable transformation $\phi : \mathbb{R}^n \Rightarrow \mathbb{R}^m$:

$$\min_{x \in C} f(x) \Leftrightarrow \min_{\phi(y) \in C} f(\phi(y))$$

Example: Geometric Program

$$\min_{x \in C} f(x) = \sum_{k=1}^{p} \gamma_k x_1^{a_{k_1}} x_2^{a_{k_2}} .. x_n^{a_{k_n}} \qquad \text{(posynomial)}$$

- C : involves (convex) inequalities in some form and equalities are affine.
- We can change above non-convex problem to the following convex problem by letting $y_i = \log x_i$

Eliminate equality constraints

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \\ & Ax = b \end{aligned} \tag{4}$$

- x feasible means $\exists M : col[M] = null[A]$ and $x_0 \quad \text{s.t.} \quad Ax_0 = b$
- Let $x = My + x_0$

Then the following is an equivalent problem:

$$\begin{aligned} \min_{y} \quad & f(My + x_0) \\ \text{s.t.} \quad & g_i(My + x_0) \leq 0 \end{aligned} \tag{5}$$

Introduce slack variables

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \\ & Ax = b \end{aligned} \tag{6}$$

- Can add equality constraints:

$$\begin{aligned} \min_{x,s} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) + s_i = 0 \\ & s_i \geq 0 \\ & Ax = b \end{aligned} \tag{7}$$

- But this is nonconvex unless g_i are affine

Relaxation

We can relax nonaffine constraints

$$\min_{x \in C} f(x) \Rightarrow \min_{x \in \tilde{C}} f(x)$$

with $C \subset \tilde{C}$

- In this case optimum of new problem is smaller or equal to the optimum of the original problem.

Standard problems

LP (Linear Programs)

$$\min_{x} c^T x$$

with affine constraints

- Basis Pursuit (not an LP) $\min_{\beta} \|\beta\|_0$ s.t. $X\beta = y$
- Above problem can be relaxed to :
  $\min_{\beta} \|\beta\|_1$ s.t. $X\beta = y$.
- This relaxation can be reformulated to a LP problem:
  $\min_{\beta,z} 1^T z$ s.t. $z \geq \beta, z \geq -\beta, X\beta = y$
- Dantzig selector
  $\min_{\beta} \|\beta\|_1$ s.t. $\|X^\top(y - X\beta)\|_\infty \leq \lambda$

QP (Quadratic program)

Lasso, ridge regression, OLS, Portfolio Optimization

SDP (Semi-Definite program)

$$\begin{aligned} \min_{X \in S_n} \quad & \text{tr}(()C^T X) \\ \text{s.t.} \quad & \text{tr}(A_i^\top X) = b_i \\ & X \succeq 0 \end{aligned} \tag{8}$$

Conic program

$$\begin{aligned} \min_{x} \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & D(x) + d \in K \end{aligned} \tag{9}$$

D a linear mapping, K a closed convex cone.

- Very similar to an LP

Relations

$LP \subset QP \subset SOCP \subset SDP \subset CP(ConicProgramming)$

Duality

Introduction

- Suppose we want to *Lower bound* our convex program
- Find $B \leq \min_x f(x), \quad x \in C$.

Example:

$$\begin{aligned} \min_{x,y} \quad & x + y \\ \text{s.t.} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned} \tag{10}$$

- What is B?

Harder

Example:

$$\begin{aligned} \min_{x,y} \quad & x + 3y \\ \text{s.t.} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned} \tag{11}$$

- What is B?

Why?

Example:

$$\begin{aligned} \min_{x,y} \quad & x + 3y \\ \text{s.t.} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned} \iff \begin{aligned} \min_{x,y} \quad & (x + y) + 2y \\ \text{s.t.} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned} \tag{12}$$

- What is B?

Harderer

Example:

$$\begin{aligned} \min_{x,y} \quad & px + qy \\ \text{s.t.} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned} \tag{13}$$

- What is B?

Solution

$$\begin{aligned} \min_{x,y} \quad & px + qy \\ \text{s.t.} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned} \iff \begin{aligned} \min_{x,y} \quad & px + qy \\ \text{s.t.} \quad & ax + ay \geq 2a \\ & bx, cy \geq 0 \\ & a, b, c \geq 0 \end{aligned} \tag{14}$$

- Adding implies
$$(a + b)x + (a + c)y \geq 2a$$
- Set $p = (a + b)$ and $q = (a + c)$ we get that $B = 2a$

Better

- We can make this lower bound bigger by maximizing:

$$\begin{aligned} \max_{a,b,c} \quad & 2a \\ \text{s.t.} \quad & a+b=p \\ & a+c=q \\ & a,b,c \geq 0 \end{aligned} \tag{15}$$

- This is the **Dual** of the **Primal** LP

$$\begin{aligned} \min_{x,y} \quad & px+qy \\ \text{s.t.} \quad & x+y \geq 2 \\ & x,y \geq 0 \end{aligned} \tag{16}$$

- Note that the number of Dual variables (3) is the number of Primal constraints

General LP

$$\begin{array}{lll} & \text{(P)} & \text{(D)} \\ \min\limits_{x} & c^\top x & \max\limits_{u,v} \quad -b^\top u - h^\top v \\ \text{s.t} & Ax=b & \text{s.t} \quad -A^\top u - G^\top v = c \\ & Gx \leq h & v \geq 0 \end{array} \tag{17}$$

Exercise

Alternate derivation

$$\begin{aligned} \min_{x} \quad & c^\top x \\ \text{s.t} \quad & Ax=b \\ & Gx \leq h \end{aligned} \tag{18}$$

- Suppose that some x is feasible.
- Then, for that x,
$$c^\top x \geq c^\top x + u^\top(Ax-b) + v^\top(Gx-h) =: L(x,u,v).$$
as long as $v \geq 0$ and u is anything.
- We call $L(x,u,v)$ the **Lagrangian**.

Now, suppose C is the feasible set, and f^* is the optimum

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

- We call $g(u,v)$ the **Lagrange Dual Function**
- Maximizing $g(u,v)$ is the Dual problem.

Weak duality

Consider the generic (primal) convex program

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t} \quad & l_i(x) = 0 \\ & h_i(x) \leq 0 \end{aligned} \tag{19}$$

- For feasible x, $v \geq 0$

$$f(x) \geq f(x) + u^\top h(x) + v^\top l(x) \geq \min_x L(x, u, v) = g(u, v).$$

- Therefore,

$$f^* \geq \max_{\forall u, v \geq 0} g(u, v) = g^*.$$

- This is **weak duality**.
- Note that the Dual Program is always convex even if P is not (pointwise max of linear functions)

Strong duality

$$f^* = g^*$$

- Sufficient conditions for strong duality: **Slater's conditions**
- If P is convex and there exists x such that for all i, $h_i(x) < 0$ (strictly feasible), then we have strong duality. (Extension: strict inequalities for i when h_i not affine.)
- Sufficient conditions for strong duality of an LP: strong duality if either P or D is feasible. (Dual of D = P)

Example

Dual for Support Vector Machine

$$\begin{array}{lll} & \text{(P)} & \text{(D)} \\ \min\limits_{\xi, \beta, \beta_0} & \frac{1}{2}\|\beta\|_2^2 + C\mathbb{1}^\top \xi & \max\limits_w \quad -\frac{1}{2} w^\top \tilde{X}^\top \tilde{X} w + \mathbb{1}^\top w \\ \text{s.t} & \xi_i \geq 0 & \text{s.t} \quad 0 \leq w \leq C\mathbb{1} \\ & y_i(x_i^\top \beta + \beta_0) \geq 1 - \xi_i & \quad w^\top y = 0 \end{array} \tag{20}$$

where $\tilde{X} = \text{diag}(y)X$.

- $w = 0$ is Dual feasible. (Don't need strict, because affine)
- We have strong duality by Slater's conditions.

KKT conditions

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t} \quad & l_i(x) = 0 \\ & h_i(x) \leq 0 \end{aligned} \tag{21}$$

1. Stationarity: $0 \in \partial(f(x) + v^\top h(x) + u^\top l(x))$: For some pair (u, v), x minimizes the Lagrangian.
2. Complementary slackness: $v_i h_i(x) = 0 \ \ , \forall i$

3. Primal feasibility: $h_i(x) \leq 0$, $l_i(x) = 0$
4. Dual feasibility: $v \geq 0$

Theorem: Solutions x^* and (u^*, v^*) Primal-Dual optimal and $f^* = g^*$, then they satisfy the KKT conditions.

Theorem: Solutions x^* and (u^*, v^*) that satisfy the KKT conditions are Primal-Dual optimal.

Example (SVM cont.) (eliminated the v from Dual problem)

1. Stationarity: $w^\top y = 0$, $\beta = w^\top \tilde{X}$, $w = C1 - v$
2. CS: $v_i \zeta_i = 0$, $w_i(1 - \zeta_i - y_i(x_i^\top \beta + \beta_0)) = 0$
3. Clear.
4. Clear.

Constraints and Lagrangians

When are the two following forms equivalent?

constrained form (C):

$$\begin{aligned} &\min f(x) \\ &s.t.\ h(x) \leq t \end{aligned}$$

Lagrangian form (L):

$$\min f(x) + \lambda h(x)$$

When C is strictly feasible, strong duality holds. So there exists λ such that for each x that solves C those x minimize L.

Now, if x^* solves L, then KKT condition for C hold by taking $t = h(x^*)$ and so x^* is a solution of C.

Algorithms

First order methods

For simplicity, consider unconstrained optimization

$$\min_x f(x) \tag{22}$$

assume f is convex and differentiable

Gradient descent

- Choose $x^{(0)}$
- Iterate $x^{(k)} = x^{(k-1)} - t_k \nabla f(x^{(k-1)})$
- Stop sometime

Why?

- Taylor expansion

$$f(y) \approx f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top H(y - x) \tag{23}$$

- replace H with $1/t$
- Choose $y = x^+$ to minimize the quadratic approximation

What t?

What to use for t_k?

Fixed

- Only works if t is exactly right
- Usually does not work

Sequence

$$t_k \quad s.t. \sum_{k=1}^{\infty} t_k = \infty, \quad \sum_{k=1}^{\infty} t_k^2 < \infty \tag{24}$$

Backtracking line search

At each iteration choose best t

1. Set $0 < \beta < 1, 0 < \alpha < \frac{1}{2}$
2. At each k, while
$$f\left(x^{(k)} - t\nabla f(x^{(k)})\right) > f(x^{(k)}) - \alpha t \left\| f(x^{(k)}) \right\|_2^2 \tag{25}$$
set $t = \beta t$ (shrink t)
3. $x^{t+1} = x - t\nabla f(x_t)$

Exact line search

- Backtracking approximates this
- At each k, solve
$$t = \arg\min_{s>=0} f(x^{(k)} - sf(x^{(k-1)}))$$
- Usually can't solve this.

Convergence

If ∇f is Lipschitz, use fixed t

1. GD converges at rate $O(1/k)$
2. ϵ-optimal in $O(1/\epsilon)$ iterations

If f is strongly convex as well

1. GD converges at rate $O(c^k)$ $(0 < c < 1)$.
2. ϵ-optimal in $O(\log(1/\epsilon))$ iterations

We call this second case "linear convergence" because it's linear on the log scale.

Stochastic Gradient Descent

- Workhorse in Neural Networks
- Suppose
$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)$$
- GD would use
$$x^{(k)} = x^{(k-1)} - \frac{t}{m} \sum_{i=1}^{m} \nabla f_i(x^{(k-1)})$$

- SGD uses
$$x^{(k)} = x^{(k-1)} - t\nabla f_{i_k}(x^{(k-1)})$$
$i_k \in \{1, \ldots, m\}$ is an index chosen at time k.

Standard choices for i_k

1. Draw $i_k \sim \text{Unif}(\{1, \ldots, m\})$.
2. Set $i_k = 1, \ldots, m, 1, \ldots, m, 1, \ldots, m, \ldots$

Most frequently update "mini-batches". Why?

1. Convergence rates are slower for SGD than GD.
2. Batches improves this.
3. SGD is computationally more efficient in per-iteration cost, memory
4. Efficient when far from the optimum, less good close to the optimum (but statistical properties don't care when you're close)

Subgradient descent

What happens when f isn't differentiable?

A **subgradient** of convex f at a point x is any g such that
$$f(y) \geq f(x) + g^\top (y - x) \qquad \forall y$$

- Always exists
- f differentiable $\Rightarrow g = \nabla f(x)$ (unique)
- Just plug in the subgradient in GD
- f Lipschitz gives $O(1/\sqrt{k})$ convergence

Example (LASSO)

$$\min_\beta \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|$$

- Subdifferential (the set of subgradients)
$$\begin{aligned} g(\beta) &= -X^T(y - X\beta) + \lambda \partial \|\beta\|_1 \\ &= -X^\top (y - X\beta) + \lambda v \end{aligned}$$
$$v_i = \begin{cases} \{1\} & \text{if } \beta_i > 0 \\ \{1\} & \text{if } \beta_i < 0 \\ [-1, 1] & \text{if } \beta_i = 0 \end{cases}$$

So $sign(\beta) \in \partial \|\beta\|_1$

- From KKT (stationarity)
$$X^T(y - X\beta) = \lambda v$$
$$X_i^T(y - X\beta) = \lambda v_i$$
- Therefore
$$|X_i^T(y - X\beta)| = \lambda \Rightarrow \beta_i \neq 0$$
$$|X_i^T(y - X\beta)| < \lambda \Rightarrow \beta_i = 0$$
- Checking the KKT conditions underlies the LARS algorithm

Proximal gradient descent

Suppose f is **decomposable**

$$f(x) = g(x) + h(x)$$

- We assume g is convex and differentiable, but h is convex only
- Approximate g only (via Taylor with approximate Hessian)

$$\begin{aligned} x^+ &= \arg\min_z g(x) + \nabla g(x)^\top (z - x) + \frac{1}{2t} \|z - x\|_2^2 + h(z) \\ &= \arg\min_z \frac{1}{2t} \|z - (x - t\nabla g(x))\|_2^2 + h(z) \end{aligned}$$

- The first part keeps us close to the gradient update for g, but still minimize over h.
- Define

$$\text{prox}_t(x) := \arg\min_z \frac{1}{2t} \|x - z\|_2^2 + h(z)$$

- The prox operator depends only on h, not g
- Easily solvable for many h
- Update becomes

$$x^{(k)} = \text{prox}_{t_k}\left(x^{(k-1)} - t_k \nabla g(x^{(k-1)})\right)$$

Example (LASSO)

$$\min_\beta \underbrace{\frac{1}{2} \|y - X\beta\|_2^2}_{g} + \underbrace{\lambda \|\beta\|_1}_{h}$$

$$\begin{aligned} \text{prox}_t(\beta) &= \arg\min_z \|z - \beta\|_2^2 + \lambda \|z\|_1 \\ &=: S_{\lambda t}(\beta) \\ S_\tau(\beta) &= \text{sign}(\beta)(|\beta| - \tau)_+ \end{aligned}$$

So the update becomes:

1. $\beta \leftarrow \beta + tX^\top(y - X\beta)$
2. $\beta \leftarrow S_{\lambda t}(\beta)$

This is called "Iterative soft thresholding" (ISTA)

Projected gradient descent

$$\min_{x \in C} f(x)$$

- Incorporate the constraint into the objective using the indicator function:

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

- Call the indicator h, and use proximal GD
- Now $\text{prox}_t(x) = \arg\min_{z \in C} \|z - x\|_2^2$
- This is just the Euclidean projection onto C

So the update becomes:

1. Use GD on f
2. Project onto C

Dual methods

Dual (sub)gradient ascent/descent

Primal problem:

$$\min_x f(x) \quad \text{s.t.} \quad Ax = b.$$

Dual is

$$\max_u -f^*(-A^\top u) + b^\top u,$$

where f^* is the **conjugate** of f.

- Let $g(u) := f^*(-A^\top u) - b^\top u$.
- Our goal is to minimize $g(u)$.
- The subdifferential is given by
$$\partial g(u) = A\partial f^*(-A^\top u) - b^\top u,$$
but if $x \in \arg\min_z L(z, u)$ then $\partial g(u) = Ax - b$.
- We may solve this as follows:
  1. guess initial $u^{(0)}$
  2. choose $x^{(k)} \in \arg\min_x f(x) + (u^{(k-1)})^\top Ax$
  3. Update $u^{(k)} = u^{(k-1)} + t_k \left(Ax^{(k)} - b\right)$

Formally: if f is strictly convex then

1. conjugate f^* is differentiable
2. procedure is dual gradient ascent
3. $x^{(k)}$ is the unique minimizer

We can choose t_k as before and apply proximal methods (or acceleration).

Decomposable dual

$$\min_x \sum_{i=1}^m f_i(x_i) \quad \text{s.t.} \quad Ax = b$$

standard minimization decomposes into x^+, which is equivalent to solving separately for each x_i:

$$x_i^+ \in \arg\min_{x_i} f_i(x_i) + u^\top A_i x_i.$$

So we can iterate:

$$x_i^{(k)} \in \arg\min_{x_i} f(x_i) + (u^{(k-1)})^T A_i x_i$$

$$u^{(k)} = u^{(k-1)} + t_k \left(\sum_{i=1}^m A_i x_i^{(k)} - b\right)$$

- Strong duality holds in this particular example since we have no inequality constraints.
- If the constraints are inequalities, i.e. $Ax \leq b$, we make a slight modification to $u^{(k)}$:
$$u^{(k)} = \left(u^{(k-1)} + t_k \left(\sum_{i=1}^m A_i x_i^{(k)} - b\right)\right)_+$$
- Updates can be parallelized.

Augmented Lagrangian

- For for dual ascent to work ($\to g^*$), we need f "nice"
- To achieve strong duality (primal optimality) the program must also satisfy one of the conditions mentioned earlier (e.g. Slater's condition).

We can alter the problem to enforce strong convexity

- Use $\min_x f(x) + \frac{\rho}{2}||Ax - b||_2^2 \quad \text{s.t.} \quad Ax = b$.
- The objective is strongly convex if A has full column rank.
- Dual gradient ascent then becomes

$$x^{(k)} = \arg\min_x f(x) + (u^{(k-1)})^\top Ax + \frac{\rho}{2}\|Ax - b\|_2^2$$
$$u^{(k)} = u^{(k-1)} + \rho(Ax^{(k-1)} - b)$$

- Replacing the step size t_k with ρ gives better convergence properties than the original DGA.
- But introducing the norm kills decomposability (if we had it), no parallelization
- ρ balances primal feasibility with a small objective
- larger ρ deemphasizes objective but forces $x^{(k)}$ toward primal feasiblity

Alternating Direction Method of Multipliers (ADMM)

$$\min_{x,z} f(x) + g(z) \quad \text{s.t.} \quad Ax + Bz = c$$

Add $\frac{\rho}{2}\|Ax + Bz - c\|_2^2$ to the objective, penalizing infeasibility:

$$L_\rho(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2$$

Iteratively update:

$$x^{(k)} = \arg\min_x L_\rho(x, z^{(k-1)}, u^{(k-1)})$$
$$z^{(k)} = \arg\min_z L_\rho(x^{(k-1)}, z, u^{(k-1)})$$
$$u^{(k)} = u^{(k-1)} + \rho(Ax^{(k-1)} + Bx^{(k-1)} - b)$$

Properties of ADMM (some of which do not require A and B to be full rank):

1. $Ax^{(k)} + Bz^{(k)} - c \to 0$ as $k \to \infty$ (primal feasibility)
2. $f^{(k)} + g^{(k)} \to f^* + g^*$ (primal optimality)
3. $u^{(k)} \to u^*$ (dual solution)
4. doesn't necessarily give $x^{(k)} \to x^*$ and $z^{(k)} \to z^*$

- The exact convergence rate is unknown, but empirically seems close to $O(1/\epsilon)$.

Example (LASSO)

$$\min_\beta \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\alpha\|_1 \quad \text{s.t.} \quad \alpha = \beta$$

ADMM update (compare with ridge regression):

$$\begin{aligned}\beta^{(k)} &= \left(X^T X + \rho I\right)^{-1}\left(X^T y + \rho(\alpha^{(k-1)} - w^{(k-1)})\right)\\ \alpha^{(k)} &= S_{\lambda/\rho}(\beta^{(k)} + w^{(k-1)})\\ w^{(k)} &= w^{(k-1)} + \beta^{(k)} - \alpha^{(k)}\end{aligned}$$

- There's an "equivalence" between β and α. You use α as the solution.

Issues with ADMM:

- How to choose ρ.
- Different ADMM formulations of the problem may have different convergence properties.

Coordinate descent

- Works well with LASSO.
- If $f(x) = g(x) + \sum_{i=1}^{n} h_i(x_i)$ where g is convex and differentiable, h merely convex
- Then:

1. Guess $x^{(0)}$.

2. Update according to:

$$\begin{aligned}x_1^{(k)} &\in \arg\min_{x_1} f(x_1, x_2^{(k-1)}, ..., x_n^{(k-1)})\\ x_2^{(k)} &\in \arg\min_{x_2} f(x_1^{(k)}, x_2, ..., x_n^{(k-1)}) \quad \text{(minimize over whole vector or block)}\\ &\vdots\end{aligned}$$

Example (LASSO)

This is what `GLMNET` does

$$\min_{\beta} \frac{1}{2}\left\|y - X\beta\right\|_2^2 + \lambda\left\|\beta\right\|_1$$

- $\left\|\beta\right\|_1 = \sum_{i=1}^{p} |\beta_i|$
- Iterate over p

$$\beta_i \leftarrow S_{\lambda/\|X_i\|_2^2}\left(\frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^\top X_i}\right)$$

- Comes from taking derivative of objective wrt β_i

References

Boyd, S.P., and L. Vandenberghe. 2004. *Convex Optimization.* Cambridge, UK: Cambridge Univ Press.