Introduction: Why Optimization?

Ryan Tibshirani
Convex Optimization 10-725/36-725

Course setup

Welcome to the course on Convex Optimization, with a focus on its ties to Statistics and Machine Learning!

Basic adminstrative details:

- Instructor: Ryan Tibshirani
- Teaching assistants: Dallas Card, Hanzhang Hu, Shashank Srivastava, Matt Wytock
- Course website: `http://www.stat.cmu.edu/~ryantibs/convexopt/`
- We will use Piazza for announcements and discussions
- We will Blackboard just as a gradebook

Prerequisites: no formal ones, but class will be fairly fast paced

Assume working knowledge of/proficiency with:

- Real analysis, calculus, linear algebra
- Core problems in Stats/ML
- Programming (Matlab or R)
- Data structures, computational complexity
- Formal mathematical thinking

If you fall short on any one of these things, it's certainly possible to catch up; but don't hesitate to talk to us

Evaluation:

- 5 homeworks
- 1 midterm
- 1 little test
- 1 project (can enroll for 9 units with no project)
- Many easy quizzes

Project: something useful/interesting with optimization. Groups of 2 or 3, milestones throughout the semester, details to come

Quizzes: due at midnight the day of each lecture. Should be very short, very easy if you've attended lecture ...

Scribing: sign up to scribe one lecture per semester, on the course website (multiple scribes per lecture). Can bump up your grade in boundary cases

Lecture videos: see links on the course website. Supposed to be helpful supplements, not replacements for the lectures! Attending lectures is still best

Auditors: welcome, please audit rather than just sitting in

Most important: work hard and have fun!

Optimization problems are ubiquitous in Statistics and Machine Learning

Optimization problems underlie most everything we do in Statistics and Machine Learning. In many courses, you learn how to:

translate  into $P : \min_{x \in D} f(x)$

Conceptual idea *Optimization problem*

Examples of this? Examples of the contrary?

This course: how to solve P, and also why this is important

Presumably, other people have already figured out how to solve

$$P \ : \ \min_{x \in D} \ f(x)$$

So why bother?

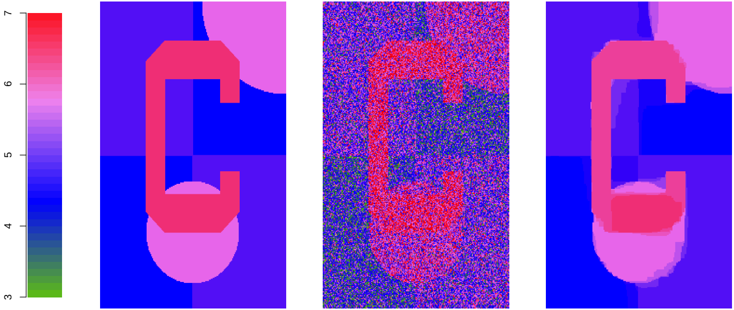
Many reasons. Here's two:

1. Different algorithms can perform better or worse for different problems P (sometimes drastically so)
2. Studying P can actually give you a deeper understanding of the statistical procedure in question

Optimization is a very current field. It can move quickly, but there is still much room for progress, especially at the intersection with Statistics and ML

Example: 2d fused lasso

Given observations $y_i \in \mathbb{R}$, $i = 1, \ldots n$ over the pixels of an image

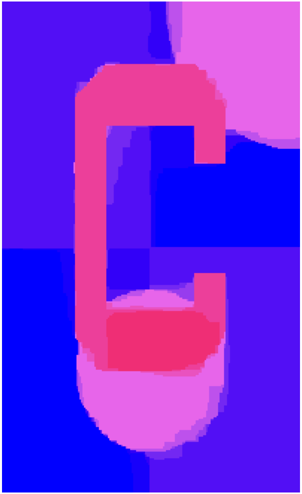


True image Data Solution

The 2d fused lasso or 2d total variation denoising fits a piecewise constant function over the image. How? By solving

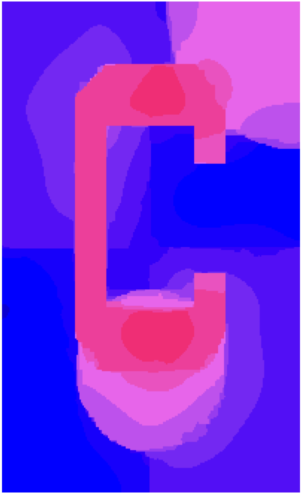
$$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{(i,j) \in E} |\beta_i - \beta_j|$$

Problem: $$\min_{\beta \in \mathbb{R}^n} \frac{1}{2}\sum_{i=1}^n (y_i - \beta_i)^2 + \lambda \sum_{(i,j)\in E} |\beta_i - \beta_j|$$



Interior point method,
10 iterations

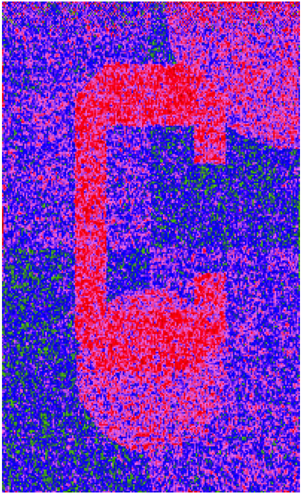
Problem: $$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{(i,j) \in E} |\beta_i - \beta_j|$$



Interior point method, 10 iterations

Proximal gradient descent, 1000 iterations

Problem: $$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{(i,j) \in E} |\beta_i - \beta_j|$$

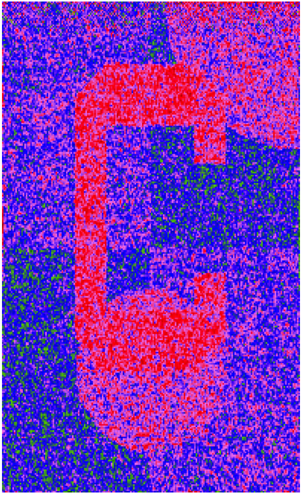


Interior point method, 10 iterations

Proximal gradient descent, 1000 iterations

Coordinate descent, 10K cycles

Problem: $$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{(i,j) \in E} |\beta_i - \beta_j|$$



Interior point method, 10 iterations

Proximal gradient descent, 1000 iterations

Coordinate descent, 10K cycles

(all from the dual)

What's the message here?

So what's the right conclusion here?

Is primal-dual interior point method simply a better method than proximal gradient descent, coordinate descent? ... No

In fact, different algorithms will work better in different situations. We'll learn details throughout the course

In the 2d fused lasso problem:

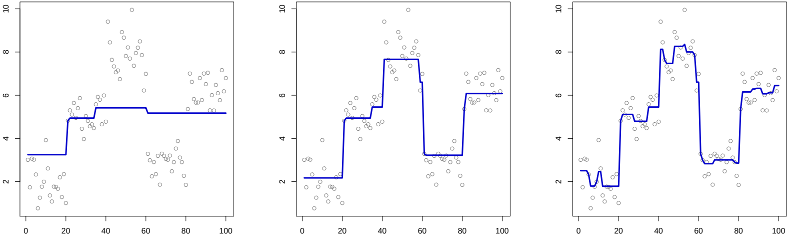
- Primal-dual: fast (structured linear systems)
- Proximal gradient: slow (conditioning)
- Coordinate descent: slow (large active set)

Example: 1d fused lasso changepoints

In the 1d fused lasso problem

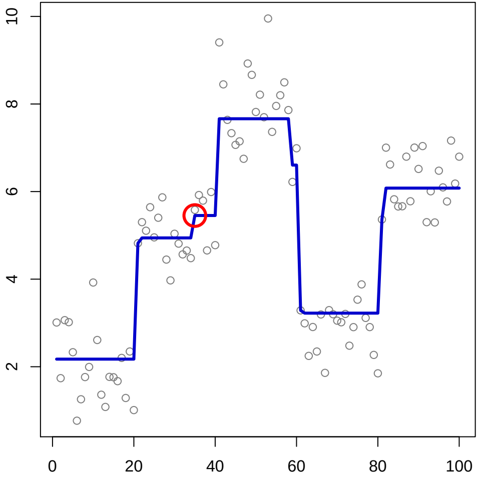
$$\min_{\beta \in \mathbb{R}^n} \; \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{i=1}^{n-1} |\beta_i - \beta_{i+1}|$$

the parameter $\lambda \geq 0$ is called a tuning parameter. As λ decreases, we see more changepoints in the solution $\hat{\beta}$



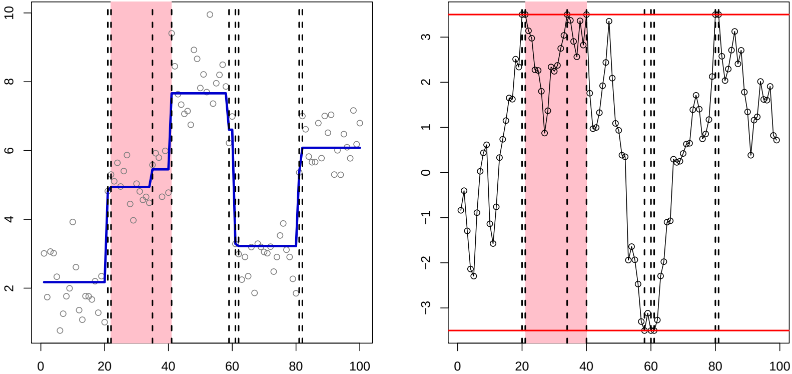
$\lambda = 25$ $\lambda = 3.5$ $\lambda = 0.8$

Let's look at the solution at $\lambda = 3.5$ a little more closely



Why does the solution exhibit such a "spurious" changepoint?

The Karush-Kuhn-Tucker (KKT) optimality conditions for the 1d fused lasso reveal a surprising property: basically: such spurious jumps will always happen, when there are staircases in the signal!



Changepoint locations in the fused lasso can be seen as the hitting times of a random walk with drift (e.g., Rojas and Wahlberg, 2014)

Central concept: convexity

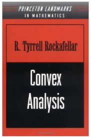
Historically, linear programs were the focus in optimization

Initially, it was thought that the important distinction was between linear and nonlinear optimization problems. But some nonlinear problems turned out to be much harder than others ...

Now it is widely recognized that the right distinction is between convex and nonconvex problems

Your supplementary textbooks for the course:

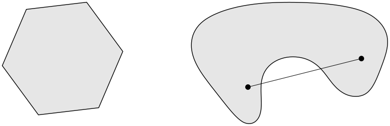
Boyd and Vandenberghe (2004) 

, Rockafellar (1970) 

Convex sets and functions

Convex set: $C \subseteq \mathbb{R}^n$ such that

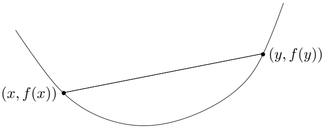
$$x, y \in C \implies tx + (1-t)y \in C \text{ for all } 0 \le t \le 1$$



Convex function: $f : \mathbb{R}^n \to \mathbb{R}$ such that $\mathrm{dom}(f) \subseteq \mathbb{R}^n$ convex, and

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y) \quad \text{for } 0 \le t \le 1$$

and all $x, y \in \mathrm{dom}(f)$



Convex optimization problems

Optimization problem:

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

Here $D = \mathrm{dom}(f) \cap \bigcap_{i=1}^m \mathrm{dom}(g_i) \cap \bigcap_{j=1}^p \mathrm{dom}(h_j)$, common domain of all the functions

This is a convex optimization problem provided the functions f and $g_i, i = 1, \ldots m$ are convex, and $h_j, j = 1, \ldots p$ are affine:

$$h_j(x) = a_j^T x + b_j, \quad j = 1, \ldots p$$

Local minima are global minima

For convex optimization problems, local minima are global minima

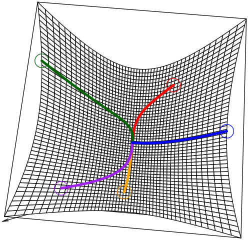
Formally, if x is feasible—$x \in D$, and satisfies all constraints—and minimizes f in a local neighborhood,

$$f(x) \leq f(y) \text{ for all feasible } y, \ \|x - y\|_2 \leq \rho,$$

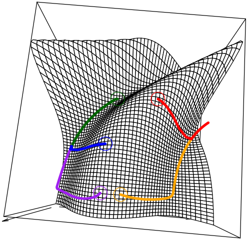
then

$$f(x) \leq f(y) \text{ for all feasible } y$$

This is a very useful fact and will save us a lot of trouble!



Convex



Nonconvex