Introduction: Why optimization?

Barnabas Poczos & Ryan Tibshirani
Convex Optimization 10-725/36-725

Administrative stuff

Instructors:

- Barnabas Poczos
- Ryan Tibshirani

TAs:

- Adona Iosif
- Yifei Ma
- Aaditya Ramdas
- Sashank Reddi

Course website:
`http://www.stat.cmu.edu/~ryantibs/convexopt/`

We will also use blackboard for a lot of things

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

Assume working knowledge of/proficiency with:

- Linear algebra, calculus
- 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 final project (can enroll for 9 units with no final project)

Final project is basically about using optimization to do something useful/interesting. Groups of 2 or 3, milestones throughout the semester, details to come

Scribing: also required once per semester, multiple scribes per lecture, sign up on course website

Recitations: Weds 4:30-6pm in Gates 4307, no recitation this week

Office hours: every day, see website

Discussion board: through blackboard

Anonymous comments: through blackboard

Videos: lectures will be videotaped, put on YouTube

Auditors: welcome, please audit rather than just sitting in

Work hard and have fun!

Optimization problems are ubiquitous

I was going to go this route, but I thought it might sound too cheesy/preachy

Optimization problems are ubiquitous in Stats/ML

More to the point, optimization problems underlie most everything we do in Statistics and Machine Learning

In many Stats/ML/Engineering/etc. courses, you learn how to:

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

Conceptual problem *Optimization problem*

Examples of this? Examples of the contrary?

In this course, you'll learn that translation is not the end of the story. I.e., we'll teach you 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?

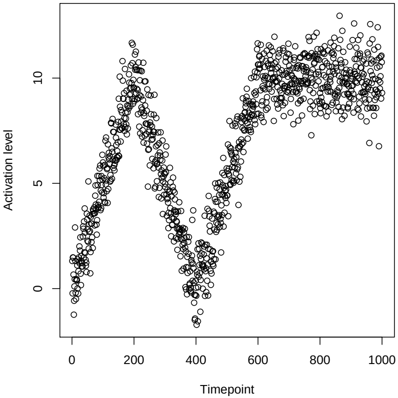
Many reasons. Here's two:

- Different algorithms can perform better/worse for different problems P (sometimes drastically so)
- Studying P can actually give you a deeper understanding of the original problem you're interested in

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

Example: linear trend filtering

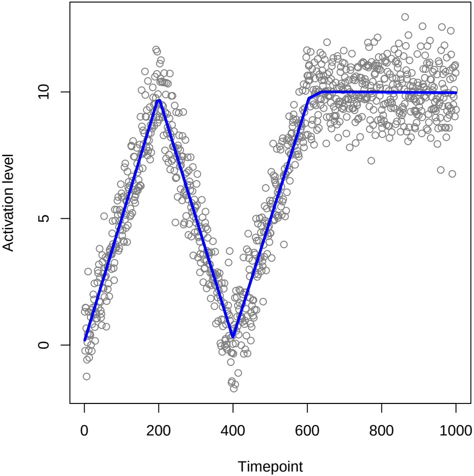
Given observations $y_1, y_2, \ldots y_n \in \mathbb{R}$ corresponding to underlying positions $1, 2, \ldots n$



Linear trend filtering fits a piecewise linear function, with adaptively chosen knots (Kim et al., 2009)

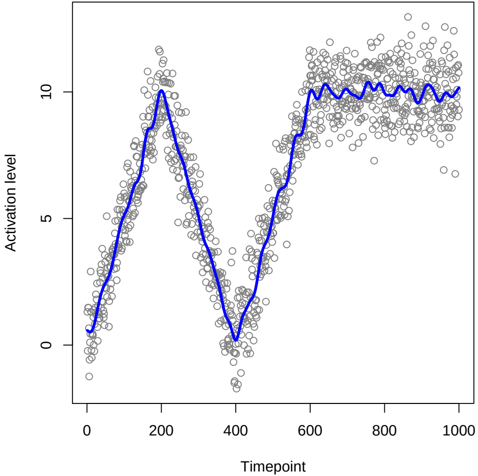
How? By solving $\displaystyle \min_{\beta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{i=1}^{n-2} |\beta_i - 2\beta_{i+1} + \beta_{i+2}|$

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



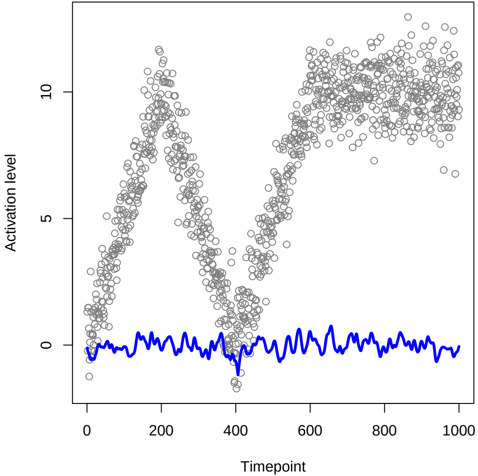
Primal-dual interior point method, 30 iterations

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



Proximal gradient descent, 10K iterations

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



Coordinate descent, 10K iterations

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 linear trend filtering problem:

- Primal-dual: fast (structured linear systems)
- Proximal gradient: slow (conditioning)
- Coordinate descent: doesn't converge (separability)

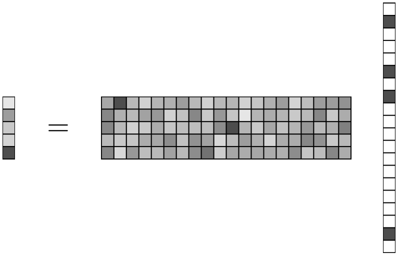
Example: sparse undetermined linear systems

Given $y \in \mathbb{R}^n$ and a matrix $X \in \mathbb{R}^{n \times p}$, with $p \gg n$. Suppose that we know that

$$y = X\beta^*$$

for some unknown vector $\beta^* \in \mathbb{R}^p$. Can we generically solve for β^*? ... No!

But if β^* is known to be sparse (i.e., have many zero entries), then it's a whole new story



There are different approaches to estimating β^*, but one popular way is to solve the problem

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_1 \quad \text{subject to} \quad X\beta = y$$

This is called basis pursuit (Chen et al., 1998). Recall that the ℓ_1 norm is $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$

There are many algorithms for computing a solution to the basis pursuit problem (in fact, it can be cast as a linear program!)

We'll focus on the AMP algorithm, which is designed for somewhat special situations (special matrices X), but has pretty remarkable properties

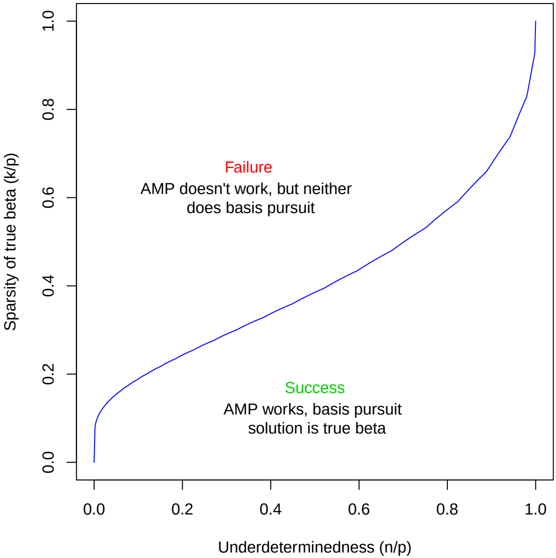
The AMP algorithm (Donoho et al., 2009) is an iterative algorithm that starts with $\beta^{(0)} = 0$, $r^{(0)} = y$, and repeats for $t = 1, 2, 3, \ldots$

$$\beta^{(t)} = S_{\lambda_t}(\beta^{(t-1)} + X^T r^{(t-1)})$$
$$r^{(t)} = y - X\beta^{(t)} + \frac{1^T \partial \|\beta^{(t)}\|_1}{\delta} r^{(t-1)}$$

Here S_λ is the soft-thresholding function at level λ (and λ_t, δ are tuning parameters)

Loosely speaking, amazing properties of AMP (for special X):

- If AMP converges, then it computes a basis pursuit solution, and this very likely recovers unknown solution β^* that we were looking for
- If AMP doesn't converge, then that's OK, because very likely no basis pursuit solution would have recovered the unknown β^* anyway



AMP traces out a phase transition for the basis pursuit problem

Convexity

Historically, linear programs were the focus in optimization

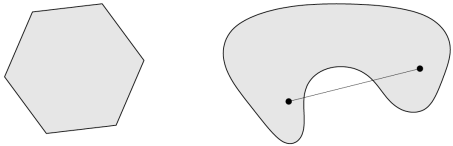
Initially, it was thought that the important distinction was between linear and nonlinear optimization problems. But some nonlinear problems turned out to be much harder than others ...

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

(Boyd and Vandenberghe (2004) sell this idea strongly; see also Rockafellar (1993))

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

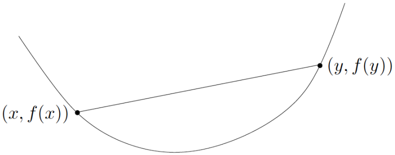
$$x, y \in C \implies tx + (1-t)y \in C \text{ for all } 0 \le t \le 1$$



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

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

for all $x, y \in \mathrm{dom}(f)$ and $0 \le t \le 1$



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 (i.e., $h_j(x) = a_j^T x + b_j$)

Local minima are global minima

For convex optimization problems, local minima are global minima

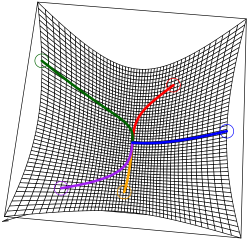
Formally, if x is feasible ($x \in D$, and satisfies all constraints) and minimizes f in a neighborhood of itself, i.e.,

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

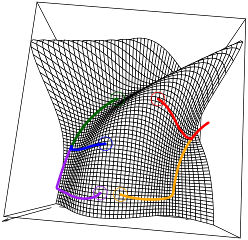
then

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

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



Convex



Nonconvex

References

- S. Boyd and L. Vandenberghe (2004), "Convex optimization"
- S. Chen, D. Donoho, and M. Saunders (1998), "Atomic decomposition by basis pursuit"
- D. Donoho, A. Maleki, and A. Montanari (2009), "Message-passing algorithms for compressed sensing"
- R. T. Rockafellar (1993), "Lagrange multipliers and optimality"