Introduction: Why Optimization?

Lecturer: Aarti Singh
Co-instructor: Pradeep Ravikumar

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:

- Instructors: Pradeep Ravikumar, Aarti Singh
- Teaching assistants: Hao Gu, Devendra Sachan, Yifeng Tao, Yichong Xu, Hongyang Zhang
- Course website:
  `http://www.cs.cmu.edu/~aarti/Class/10725_Fall17/`
  `http://www.cs.cmu.edu/~pradeepr/convexopt/`
- We will use Piazza for announcements and discussions

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, Python, ...)
- 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 - 45%
- 2 little in-class tests (Oct 16, Dec 6) - 25%
- 1 project (poster presentation Dec 13 1:30-4:30 pm) - 25%
- Many easy quizzes - 5%

Project: something useful/interesting with optimization. Groups of 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 ...

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

Conceptual idea

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

Optimization problem

Examples of this? Examples of the contrary?

This course:

- how to solve P, and also why this is important

Let the solution to P be

$$f^* = \min_{x \in D} f(x)$$

This course:

- how close is the solution obtained by different optimization algorithms to f^*?

Not this course:

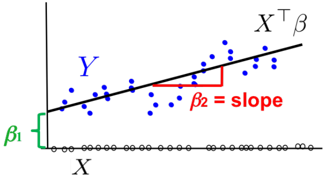
- Not focus on generalization: A good solution to P only implies good generalization error if optimization problem is a good empirical surrogate for true error (10-702 Statistical Machine Learning)

Example Optimization Problems in ML and Stat

Example Optimization Problems in ML and Stat

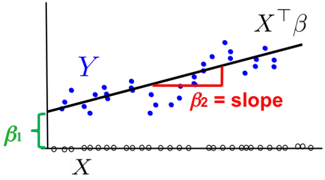
Parametric (e.g. linear) Regression:

- Least squares
  $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2$
- Least absolute deviation
  $\min_\beta \sum_{i=1}^n |y_i - x_i^\top \beta|$



Example Optimization Problems in ML and Stat

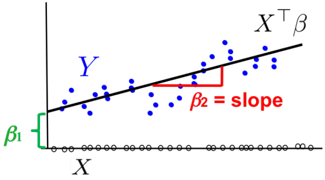
Parametric (e.g. linear) Regression:



- Least squares
  $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2$
- Least absolute deviation
  $\min_\beta \sum_{i=1}^n |y_i - x_i^\top \beta|$
- Regularized least squares
  Ridge $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda\|\beta\|_2^2$
  Lasso $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda\|\beta\|_1$

Example Optimization Problems in ML and Stat

Parametric (e.g. linear) Regression:



- Least squares
  $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2$
- Least absolute deviation
  $\min_\beta \sum_{i=1}^n |y_i - x_i^\top \beta|$
- Regularized least squares

  Ridge $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_2^2$

  Lasso $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_1$

De-noising:

- Least squares $\min_\theta \sum_{i=1}^n (y_i - \theta_i)^2$
- Regularized least squares

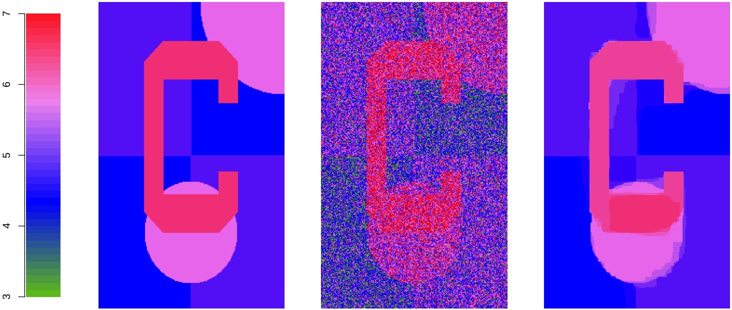
  Fused lasso $\min_\theta \; \frac{1}{2} \sum_{i=1}^n (y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$

Example: 2d fused lasso for denoising

The 2d fused lasso or 2d total variation denoising problem is:

$$\min_{\theta} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$

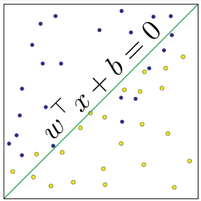
This fits a piecewise constant function over an image, given data y_i, $i = 1, \ldots, n$ at pixels



True image Data Solution

Example Optimization Problems in ML and Stat

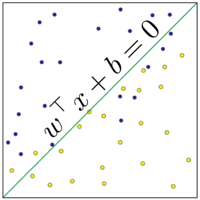
Classification:



Example Optimization Problems in ML and Stat

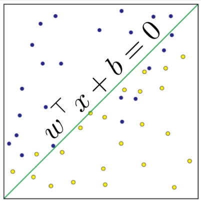
Classification:

- Support vector machines (SVM)
  - regularized hinge loss
  $\min_{w,b} \|w\|^2 + C \sum_{i=1}^{n} (1 - (w^\top x_i + b) y_i)_+$



Example Optimization Problems in ML and Stat

Classification:

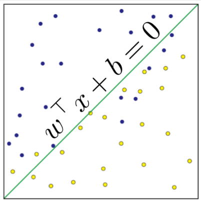


- Support vector machines (SVM)
  - regularized hinge loss
  $\min_{w,b} \|w\|^2 + C \sum_{i=1}^n (1 - (w^\top x_i + b) y_i)_+$
- Logistic regression
  - max conditional likelihood
  $\max_{w,b} \sum_{i=1}^n P(y_i | x_i, w, b)$ where $P(y = 0 | x) = \frac{1}{1+\exp(w^\top x + b)}$

Max Likelihood Estimation

Example Optimization Problems in ML and Stat

Classification:



- Support vector machines (SVM)
  - regularized hinge loss
  $\min_{w,b} \|w\|^2 + C \sum_{i=1}^{n} (1 - (w^\top x_i + b) y_i)_+$
- Logistic regression
  - max conditional likelihood
  $\max_{w,b} \sum_{i=1}^{n} P(y_i | x_i, w, b)$ where $P(y = 0|x) = \frac{1}{1+\exp(w^\top x + b)}$

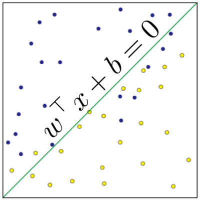
Max Likelihood Estimation

Matrix completion/Factorization/Principal Component Analysis:

- PCA $\min_A \|X - A\|_F^2$ s.t. rank$(A) = k$

Example Optimization Problems in ML and Stat

Classification:



- Support vector machines (SVM)
  - regularized hinge loss
  $\min_{w,b} \|w\|^2 + C \sum_{i=1}^{n} (1 - (w^\top x_i + b) y_i)_+$
- Logistic regression
  - max conditional likelihood
  $\max_{w,b} \sum_{i=1}^{n} P(y_i | x_i, w, b)$ where $P(y = 0 | x) = \frac{1}{1+\exp(w^\top x + b)}$

Max Likelihood Estimation

Matrix completion/Factorization/Principal Component Analysis:

- PCA $\min_A \|X - A\|_F^2$ s.t. $\text{rank}(A) = k$

What ML and Stats problems are not optimization?

Presumably, other people have already figured out how to solve

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

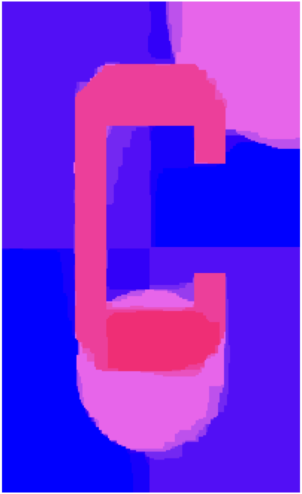
So why bother?

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

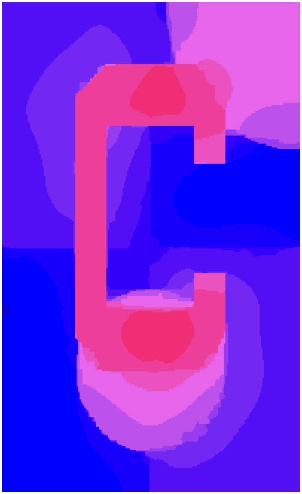
2d denoising problem: $$\min_{\theta} \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta_i)^2+\lambda\sum_{(i,j)\in E}|\theta_i-\theta_j|$$



Algorithms:

Specialized ADMM, 20 iterations

2d denoising problem: $$\min_{\theta} \; \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$

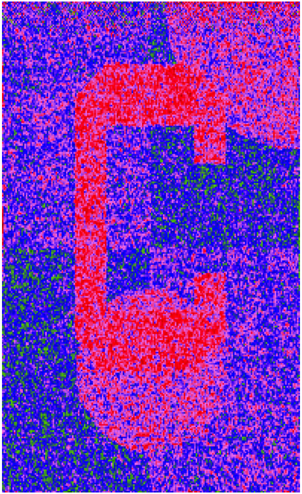


Algorithms:

Specialized ADMM, 20 iterations

Proximal gradient descent, 1000 iterations

2d denoising problem: $$\min_\theta \ \frac{1}{2}\sum_{i=1}^n (y_i-\theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$



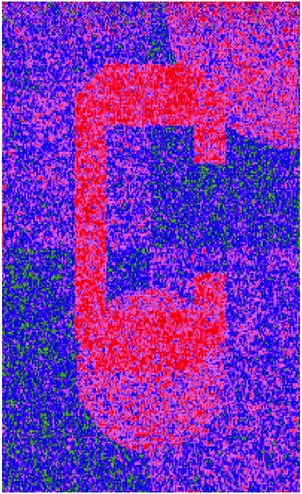
Algorithms:

Specialized ADMM, 20 iterations

Proximal gradient descent, 1000 iterations

Coordinate descent, 10K cycles

2d denoising problem: $$\min_{\theta} \; \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j|$$



Algorithms:

Specialized ADMM, 20 iterations

Proximal gradient descent, 1000 iterations

Coordinate descent, 10K cycles

What's the message here?

So what's the right conclusion here?

Is the alternating direction method of multipliers (ADMM) 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:

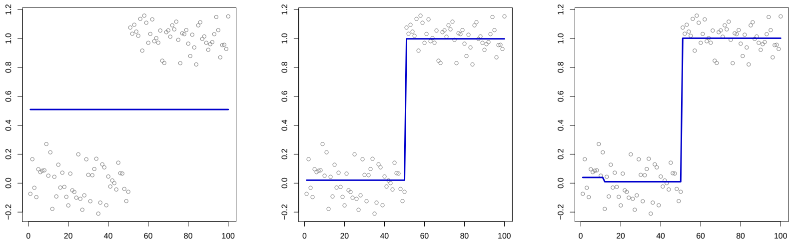
- Specialized ADMM: fast (structured subproblems)
- Proximal gradient: slow (poor conditioning)
- Coordinate descent: slow (large active set)

Example: testing changepoints from the 1d fused lasso

In the 1d fused lasso or 1d total variation denoising problem

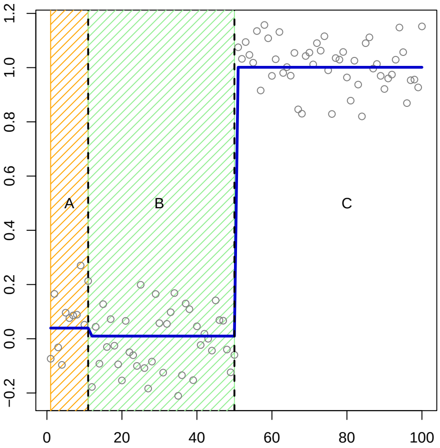
$$\min_{\theta} \ \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}|$$

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



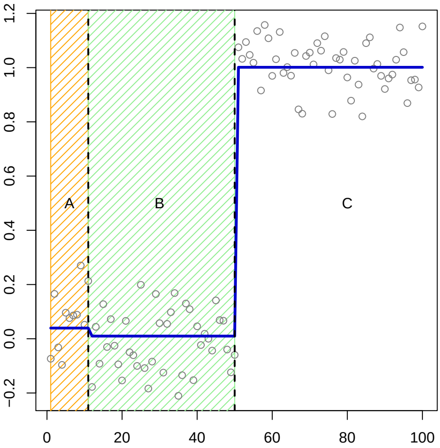
$\lambda = 25$ $\quad$ $\lambda = 0.62$ $\quad$ $\lambda = 0.41$

Let's look at the solution at $\lambda = 0.41$ a little more closely



How can we test the significance of detected changepoints? Say, at location 11?

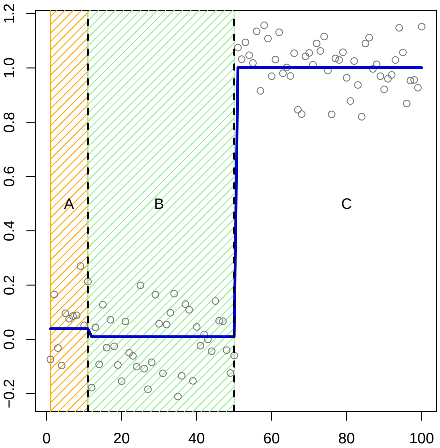
Let's look at the solution at $\lambda = 0.41$ a little more closely



How can we test the significance of detected changepoints? Say, at location 11?

Classically: take the average of data points in region A minus the average in B, compare this to what we expect if the signal was flat

Let's look at the solution at $\lambda = 0.41$ a little more closely

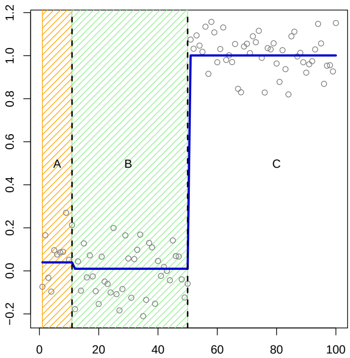


How can we test the significance of detected changepoints? Say, at location 11?

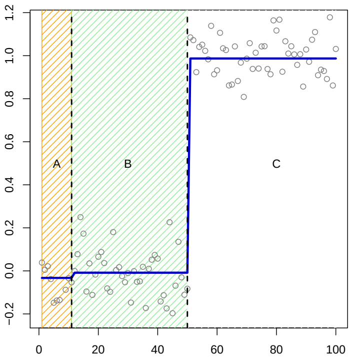
Classically: take the average of data points in region A minus the average in B, compare this to what we expect if the signal was flat

But this is incorrect, because location 11 was selected based on the data, so of course the difference in averages looks high!

What we want to do: compare our observed difference to that in reference (null) data, in which the signal was flat and we happen to select the same location 11 (and 50)



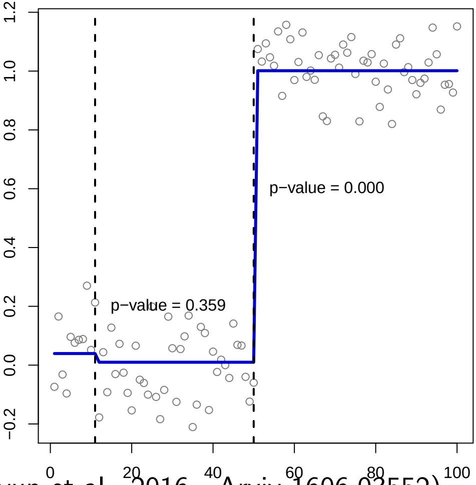
Observed data
Abs. difference ≈ 0.088



Reference data
Abs. difference ≈ 0.072

But it took 1222 simulated data sets to get one reference data set!

The role of optimization: if we understand the 1d fused lasso, i.e., the way it selects changepoints (stems from KKT conditions), then we can come up with a reference distribution without simulating



Accordingly, we can efficiently conduct rigorous significance tests

(Hyun et al., 2016 - Arxiv 1606.03552)

Example: Sparsity of Lasso solution

Lasso: $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_1$

Example: Sparsity of Lasso solution

Lasso: $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_1$

Surrogate for: $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_0$

Desire solution β to be sparse aka with small $\|\beta\|_0$ i.e. few non-zero coefficients.

Why? Only few features are relevant, require correspondingly few data points, ...

Example: Sparsity of Lasso solution

Lasso: $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_1$

Surrogate for: $\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|_0$

Desire solution β to be sparse aka with small $\|\beta\|_0$ i.e. few non-zero coefficients.

Why? Only few features are relevant, require correspondingly few data points, ...

But is the lasso solution sparse?

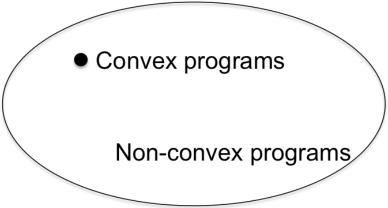
Analysis of KKT (Karush-Kuhn-Tucker) conditions (satisfied by solution to optimization problem) helps us understand when the lasso solution is sparse.

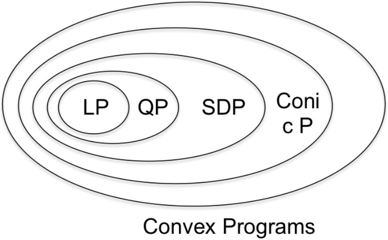
(Wainwright, 2006 - Arxiv: math/0605740)

Central concept: convexity

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





Books

Your supplementary textbooks for the course:

BV: Convex Optimization, Stephen Boyd and Lieven Vandenberghe, (available online for free).

DB: Nonlinear Programming, Dimitri P. Bertsekas.

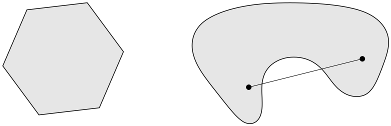
NW: Numerical Optimization, Jorge Nocedal and Stephen Wright.

YN: Introductory lectures on convex optimization: a basic course, Yurii Nesterov.

Convex sets and functions

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

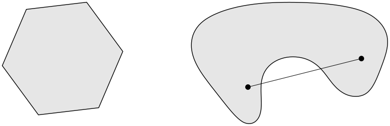
$$x, y \in C \implies tx + (1-t)y \in C \text{ for all } 0 \le t \le 1$$



Convex sets and functions

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

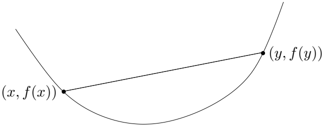
$$x, y \in C \implies tx + (1-t)y \in C \text{ for all } 0 \le t \le 1$$



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

Local minimum: If x is feasible ($x \in D$, and satisfies all constraints) and minimizes f in a local neighborhood,

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

Local minima are global minima

For convex optimization problems, local minima are global minima

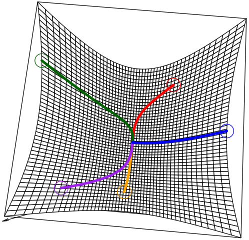
Local minimum: If x is feasible ($x \in D$, and satisfies all constraints) and minimizes f in a local neighborhood,

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

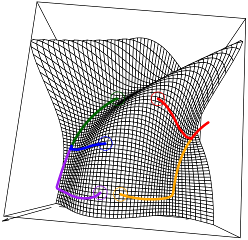
For convex problems, x is also a global minimum

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

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



Convex



Nonconvex