10-725/36-725: Convex Optimization **Fall 2016**

Lecture 1: August 29

Lecturer: Ryan Tibshirani *Scribes: Lanxiao Xu, Yuxing Zhang, Tianshu Ren*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

1.1 Course Setup

1.1.1 Basic Administrative Details

- **Instructors:** Javier Peña, Ryan Tibshirani
- **Teaching assistants:** Alnur Ali, Christoph Dann, Sangwon Hyun, Mariya Toneva, Han Zhao
- **Course website:** `http://www.stat.cmu.edu/~ryantibs/convexopt/`

1.1.2 Assumed Knowledge

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

1.1.3 Evaluation

- 5 homeworks
- 2 little tests
- 1 project (can enroll for 9 units with no project)
- Many easy quizzes
- Scribing: To bump up grade.

1.2 Overview of Optimization Problems

1.2.1 Optimization Problems are Ubiquitous

Optimization problems underlie most everything we do in Statistics and Machine Learning. Examples of problems we deal with in Statistics and Machine Learning and its relationship with optimization are listed below:

1. Optimization problems
   (a) Regression
      i. Least Squares: $\min_\beta \sum_i (y_i - x_i^T \beta)^2$
      ii. Least Absolute Deviations: $\min_\beta \sum_i |y_i - x_i^T \beta|$
   (b) Regularized Regression
      i. Lasso: $\min_\beta \sum_i (y_i - x_i^T \beta)^2 \quad s.t. \quad \sum_j |\beta_j| \leq t$
   (c) Denoising
      i. Total-Variation denoising / Fused Lasso
   (d) Classification
      i. Logistic regression
      ii. 0-1 loss
      iii. Hinge loss / SVM
   (e) Other problems
      i. Travelling-salesman problem (TSP)
      ii. Planning / Discrete optimization
      iii. Maximum-likelihood estimation
2. Non-optimization problems
   (a) Hypothesis testing / p-values
   (b) Boosting
   (c) Random Forests
   (d) Cross-validation / bootstrap

In general, non-optimization problems are harder to understand because it is tied to a procedure. And to understand those problems, one need to understand all the details in a procedure. On the other hand, an optimization problem is usually clearly defined mathematically as an objective function and a set of constraints, which we have standard tools to analyze.

In a lot of cases, we need to translate a conceptual idea into an optimization problem, which has the form

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

In this course, we study how to solve P and why this is important. The main reasons for studying this is

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

1.2.2 Example: Algorithms for the 2d Fused Lasso

2d Fused Lasso problem can be used to describe the image denoising problem where the original image is composed of several color blocks and a noisy observation is given as input. For instance, Figure 1.1 shows an example of such image denoising problem.

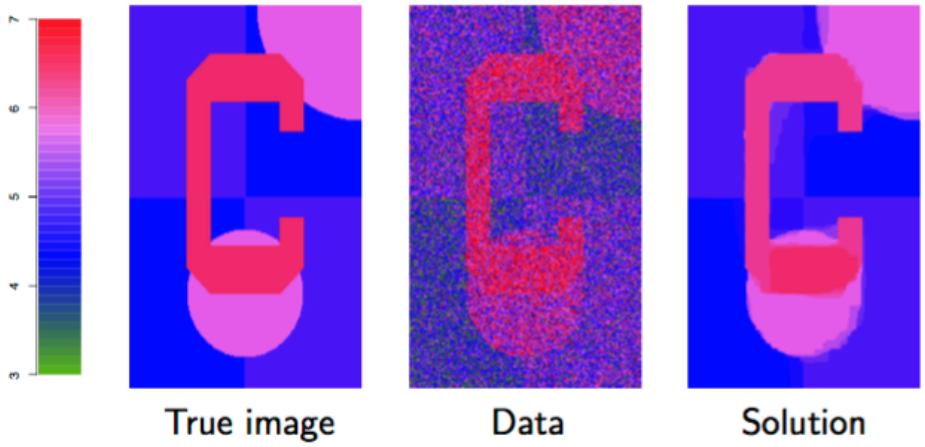
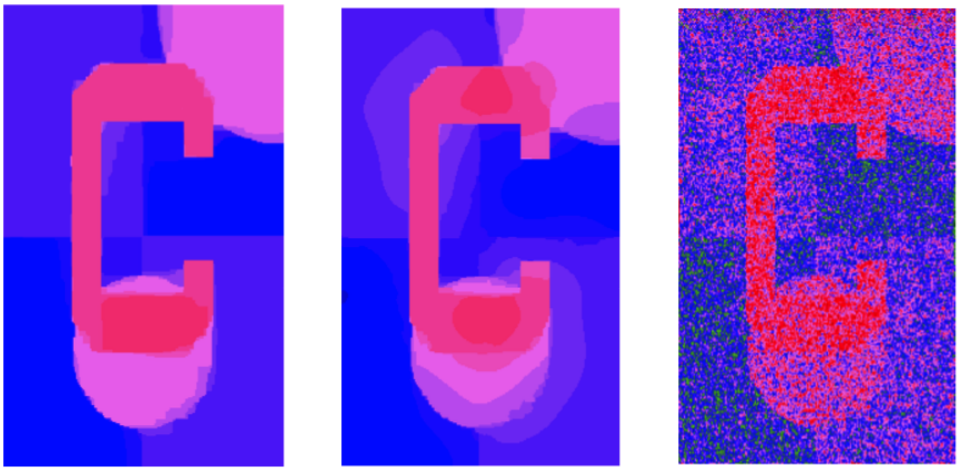


Figure 1.1: An example of image denoising that can be formulated as 2d Fused Lasso

Let y_i be the pixel information for the observed data and θ be our estimate for the original image, the 2d Fused Lasso can be formulated as

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

where $(i, j) \in E$ if and only if the pixels represented by y_i and y_j are adjacent in the original image.



(a) Specialized ADMM (b) Proximal Gradient Descent (c) Coordinate Descent

Figure 1.2: Recovered solution from different algorithms

Figure 1.2 shows the result of running different algorithms to solve the optimization problem on the example data in Figure 1.1.

Specifically, the result was obtained after 20 iterations for ADMM, 1000 iterations for proximal gradient descent and 10000 cycles for coordinate descent. The key message here is NOT that ADMM is a better method than proximal gradient descent. Rather, different algorithms will work better in different situations.

1.3 Main Definitions

1.3.1 Convex Sets and Functions

Definition 1.1 (Convex set) *$C \subseteq \mathcal{R}^n$ is a convex set if $x, y \in C \Rightarrow tx + (1-t)y \in C$, for all $0 \leq t \leq 1$.*

Definition 1.2 (Convex function) *$f : \mathbb{R}^n \to \mathbb{R}$ is a convex function if $dom(f) \subseteq \mathbb{R}^n$ is convex, and $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$, for all $0 \leq t \leq 1$ and all $x, y \in dom(f)$.*

1.3.2 Convex Optimization Problems

Problem formulation (optimization problems):

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

where $D = \text{dom}(f) \cap \bigcap_{i=1}^m \text{dom}(g_i) \cap \bigcap_{j=1}^p \text{dom}(h_j)$ is the intersection of all the domains.

Definition 1.3 (Convex optimization problem) *The optimization problem above is a convex optimization problem if:*

1. *f and $g_i, i = 1, \ldots, m$ are convex*
2. *$h_j, j = 1, \ldots, p$ are affine, meaning that $h_j(x) = a_j^T x + b_j, j = 1, \ldots, p$*

Theorem 1.4 (Local minima are global minima) *For a convex optimization problem, if x is feasible and minimizes f in a local neighborhood,*

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

then $f(x) \leq f(y)$ for all feasible y.

Proof: Suppose $\exists z \in D$ and is feasible such that $f(z) < f(x)$. According to the definition of local minima, we have $\|z - x\|_2 > \rho$.

We let $y = tx + (1-t)z$, where $0 \leq t \leq 1$. Because D is a convex set, according to its definition we also have $y \in D$.

Then for each $i = 1, ..., m$,

$$g_i(tx + (1-t)z) \leq tg_i(x) + (1-t)g_i(z) \leq 0. \tag{1.1}$$

For each $j = 1, ..., r$,

$$h_i(tx + (1-t)z) = 0 \tag{1.2}$$

From (1.1) and (1.2), we conclude y is also feasible.

If we let t large enough (close to 1 but less than 1) such that $\|x - y\|_2 \leq \rho$, we obtain

$f(y) = f(tx + (1-t)z) \leq tf(x) + (1-t)f(z) < f(x)$, which is contradictory to the local minima definition. So by proof of contradiction, we conclude the local minima are also global minima. ■