STAT 538 Lecture 2
January 5, 2012
Optimization for Statistical Learning
©Marina Meilă
mmp@stat.washington.edu

1 Why optimization?

Machine learning is both a part of statistics and a part of computer science, and of its subfield **artificial intelligence**. The name machine learning was coined by computer scientists, and emphasizes the view that learning, an attribute of human intelligence, can be emulated by a machine. This was a very bold and exciting idea a few decades ago! With time, it became increasingly clear that one cannot do good machine learning while ignoring statistics. In the same time, groups of statisticians realized the challenge of large, multidimensional data, produced by the pervasive use of computers. They also realized they have a lot to contribute to the field of machine learning. The name statistical learning reflects this new state of understanding, that statistics is indispensable to automated "learning". As a paranthesis, even before the appreciation for statistics was wide-spread, it was realized that optimization is important for learning. Thus, the two areas that we will study are not randomly paired, but organically related.

In fact, some people like to define statistical learning as "computationally minded statistics". Computation - as the theory and methods for efficiently computing - is a fundamental part of statistical learning. Much of this computation is related to optimization.

Optimization studies finding minima or maxima of functions. We encounter this task in most cases when a problem has been "solved on paper" or just formulated, but we need to follow with computing the value of its solution. Most practically interesting problems and many theoretical ones do not have

analytical solutions. Therefore, optimization in the technical sense deals with the algorithms for numerically finding extrema of functions and with finding conditions when these extrema exist.

We will look at applications in statistics (e.g estimating parameters by maximizing likelihood) and in related machine learning methods (support vector machines, boosting). **Convexity** is intimately related to a class of statistical models called **exponential family models** and to the information theoretic concepts of **entropy and Kullback-Liebler divergence**, so we will study the latter too, mainly from an algorithmic perspective.

2 Optimization problems in statistics and machine learning

Here are some situations that call for optimization:

- Estimating parameters in the Maximum Likelihood framework. Given: (1) parametric model $P_\theta(x)$ with $\theta \in \Theta \subseteq \mathbb{R}^d$ a parameter vector; (2) data $\mathcal{D}_N = \{x^1, \dots x^N\} \subseteq X$ sampled iid. Wanted parameter values
$$\theta^{ML} = \operatorname*{argmax}_{\theta \in \Theta} P_\theta(\mathcal{D}_N) \tag{1}$$
- Estimating the most probable outcome for given model P_θ
$$x^* \;=\; \operatorname*{argmax}_x P_\theta(x) \tag{2}$$
- other parameter estimation paradigms: **Least squares estimation**, **MAP estimation**, **Minimax estimation**
- Non-parametric estimationby: e.g **shape constrained estimation**, the **Maximum Entropy** framework (more about this later),
- [**Model selection**]
- **Clustering**
  - "K-means" or least squares clustering
    Given data $\mathcal{D}_N = \{x^1, \dots x^N\}$ find assignments $a(i) \in \{1, \dots K\}$ such that
    $$\text{minimize}: \sum_{k=1}^{K} \sum_{a(i)=k} ||x_i - \mu_k||^2 \tag{3}$$
    for $\mu_k = \frac{\sum_{a(i)=k} x_i}{\sum_{a(i)=k} 1}$

– Minimum diameter clustering
Given data $\mathcal{D}_N = \{x^1, \dots x^N\}$ find assignments $a(i) \in \{1, \dots K\}$ such that

$$\text{minimize}: \quad \max_{a(i)=a(j)} ||x_i - x_j|| \tag{4}$$

– Minimum Normalized Cut ("spectral clustering")
Given data $S_{ij} = S_{ji} \geq 0$, for $i, j = 1, \dots n$ find assignment $a(i) \in \{1, \dots K\}$ such that

$$\text{minimize} \quad \sum_{k=1}^{K} \sum_{a(i)=k} \frac{\sum_{j:a(j)\neq k} S_{ij}}{\sum_{j=1}^{N} S_{ij}} \tag{5}$$

2.1 Prediction problems

- **Linear regression with Least Squares cost**

$$\text{minimize} \quad \sum_{i=1}^{N} ||y^i - \beta^T x^i||^2 \tag{6}$$

As we already know (Lecture 1), this problem has a closed form solution given

$$\hat{\beta} \;=\; (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{7}$$

with $\mathbf{X}, \mathbf{Y}$ representing respectively the matrix with the inputs x^i as rows, and the vector of corresponding outputs.

- **Ridge regression**

$$\text{minimize} \quad \sum_{i=1}^{N} ||y^i - \beta^T x^i||^2 + \lambda||\beta||^2 \tag{8}$$

This is a regularized regression, where a large β parameters are penalized. The problem also has a closed form solution

$$\hat{\beta} \;=\; (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{Y}. \tag{9}$$

- **Lasso**

$$\text{minimize} \quad \sum_{i=1}^{N} ||y^i - \beta^T x^i||^2 + \lambda||\beta||_1 \tag{10}$$

Here, the penalty on β is proportional to $||\beta||_1 = \sum_{j=1}^{N} |\beta_j|$, the 1-norm of β. We shall see later that this is a *sparsity inducing* penalty. The Lasso estimator does not have a closed form expression.

- **Classification** Given data $\mathcal{D}_N = \{(x^1, y^1), \ldots (x^N, y^N)\}$ sampled iid, find classifier function $f_\theta(x)$ such that (for instance)

$$\text{minimize} \underbrace{\sum_{i=1}^{N} L(y^i - f_\theta(x^i))}_{\text{error term}} + \underbrace{\lambda J(f_\theta)}_{\text{regularization term}} \tag{11}$$

  For L_{01} the misclassification error cost, as well as for the weighted misclassificatioh cost this problem does not have a closed form solution. In addition, especially when $\lambda = 0$, there may be multiple solutions.

3 ML estimation problems

3.1 Multinomial model

$X = \{1, 2, \ldots K\}$, $\theta = \{P(1)\, P(2)\, \ldots P(K)\} \equiv \{\theta_1\, \ldots\, \theta_K\}$.

$$P_\theta(\mathcal{D}_N) \;=\; \prod_{k=1}^{K} \theta_k^{n_k} \tag{12}$$

with $n_k = \#(x_i = k)$ in $\mathcal{D}_N$. Max likelihood parameter estimate θ^{ML} is obtained from

$$\underset{\theta \in \Theta}{\operatorname{argmax}} \sum_k N_k \ln \theta_k \tag{13}$$

Constraints: $\sum_k \theta_k \;=\; 1;\; \theta_k \geq 0,\; k = 1 : K$

This is an **exponential family** model:

$$\ln P_\theta(x) \;=\; \sum_k \delta_{x,k} \ln \theta_k - A(\theta) \tag{14}$$

3.2 Normal distribution

$X = \mathbb{R}^p$, $\theta = \{\mu \in \mathbb{R}^p, \Sigma \in \mathbb{R}^{p \times p}\, \text{symmetric}, \geq 0\}$

$$\ln P_\theta(\mathcal{D}_N) \;=\; -\frac{N}{2} \ln |\Sigma| - \frac{Np}{2} \ln(2\pi) - \frac{1}{2} \sum_{i=1}^{N} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \tag{15}$$

For $p = 1$, $\theta = [\mu\ \sigma^2]$ and the log-likelihood is

$$\ln P_\theta(\mathcal{D}_N) \ = \ -\frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 \tag{16}$$

Constraints: $\sigma^2 > 0$

This is an exponential family model:

$$\ln P_\theta(x) \ = \ (-\frac{1}{2\sigma^2})x^2 + \frac{\mu}{\sigma^2}x - A(\mu,\sigma^2) \tag{17}$$

3.3 Logistic density

$X = \mathbb{R}$, $\theta = [a\,b]^T,\ a > 0$

$$\ln P_\theta(\mathcal{D}_N) \ = \ \sum_{i=1}^{N}\log\frac{ae^{-ax_i-b}}{(1+e^{-ax_i-b})^2} \tag{18}$$

The name comes from the logistic CDF given by

$$F(x;a,b) \ = \ \frac{1}{1+e^{-ax-b}} \tag{19}$$

[Exercise: Logistic regression is closely related to logistic density estimation. Formulate the logistic regression problem as an optimization problem.]

3.4 (Finite) Mixture of Gaussians

$X = \mathbb{R}^p$, $\theta = \{\pi_1,\ldots\pi_K \in \mathbb{R},\ \mu_1,\ldots\mu_K \in \mathbb{R}^p,\ \Sigma_1,\ldots\Sigma_K \in \mathbb{R}^{p\times p}\,\text{symmetric}, \geq 0\}$

$$\ln P_\theta(\mathcal{D}_N) \ = \ \sum_{i=1}^{N}\ln\left[\sum_{k=1}^{K}\pi_k\frac{1}{|\Sigma_k|^{1/2}(2\pi)^{p/2}}e^{-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)}\right] \tag{20}$$

Note that for each of the above problems, one can either: (1) estimate the parameters by optimizing over θ with the $\mathcal{D}$ fixed, or (2) estimate the most likely x (the **mode** of the distribution) by optimizing over x with fixed θ. The latter is less often done in statistics, being replaced by expectation.

[To think of: integration vs maximization in statistics]

3.5 Non-parametric shape constrained estimation: Convex regression

This area of statistics deals with estimating the best function that fits the data in a certain class. For example, in **convex Least Squares regression** we want to find the convex curve that best fits a given data set. Let the data be $\mathcal{D}_N = \{(x^1, y^1), \dots (x^N, y^N)\}$ with $x^i, y^i \in \mathbb{R}$. We want to find the *convex* function $f(x)$ that minimizes

$$\sum_{i=1}^{N} (y^i - f(x^i))^2$$

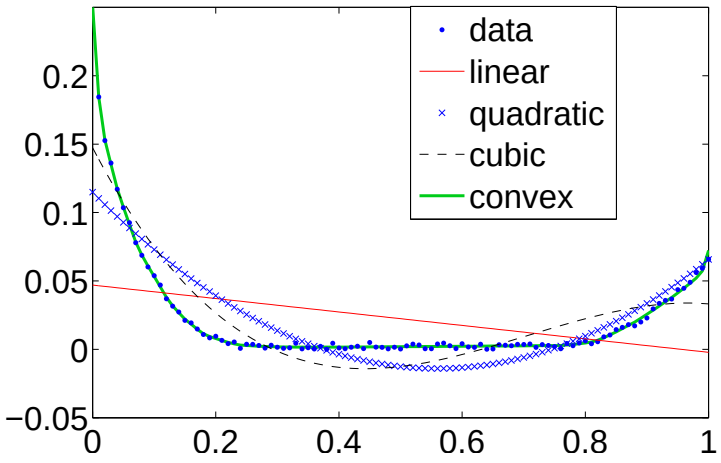
This model/problem is *non-parametric* because rather than searching for the solution in a finite-dimensional parameter space, we search in an infinite dimensional function space (the space of all convex functions). However, one can show (by convex analysis techniques that we will learn later) that the solution will always be a piecewise linear function with break points in a subset of $\{x^1, \dots, x^n\}$. This solution can be obtained by solving for the a_i coefficients:

$$\min_{a_{-1:n+1}} \quad \sum_{i=1}^{N} \left[y^i - \sum_{j=-1}^{N+1} a_j g_j(x^i) \right]^2 \tag{21}$$

$$\text{s.t.} \quad a_j \geq 0 \quad \text{for all } j \tag{22}$$

where $g_j(x)$ are predetermined piecewise linear functions.

The figure below depicts the result of convex regression on a toy data set, highlighting the differences between the convex regressor and the parametric regressors.



4 Optimization in this course

In this course, we will focus on the following kinds of optimization problems.

- **Continuous** optimization
- **Multivariate** optimization. As a consequence, we care about the complexity w.r.t the **number of parameters** to optimize over, in addition to the complexity w.r.t, for instance, the **number of data points (sample size)**. The number of parameters will be denoted by n from now on.
- We assume that in general an *analytical solution is not known*, or doesn't exist. Thus, we will discuss **numerical techniques** for optimization.
  Sometimes the optimum of a problem is not attained for any finite value; but a finite supremum may exist. In other cases, there is an infinite continuous set of possible points where the optimum is attained. If such a case appears in an estimation problem, we say that the respective model is **not identifyable**. One needs to recognize such problems before trying to solve them, and recast them in a way that removes such indeterminacies.
- **Constrained** vs **unconstrained** optimization. We will start with unconstrained problems then will continue with constrained (convex) ones. Most problems in statistics are constrained.
- **Global** vs **local** optima. Finding a global optimum is easy when the problem is convex. Otherwise it is typically hard (NP-hard search problem). We will focus less on search (discrete problem) and more on finding local optima (continuous problem).
- We will distinguish between "easy and nice" cases of numerical optimization which are the **convex** optimization problems, vs the others. We will give special attention to the former – how the recognize them, properties of these problems, algorithms.
- We will also learn to recognize situations when an analytical solution exists to a statistical problem – typically these will be within the **exponential family** class of models. This family has deep connections to convexity, so it will be natural to include its study in the course.

Problem	**Continuous**	**Constrained**	**Analytic**	Unique opt
Lin., ridge regr.	yes	no	yes	yes
Lasso	yes	no	no	yes
(Lin.) Classif, L_{01}	yes	no	no	no
Multinomial	yes	$\sum_k \theta_k = 1,\ \theta_k \geq 0$	yes	yes
Normal	yes	$\Sigma \geq 0$	yes	yes
Logistic	yes	$a > 0$	no	yes
Mixture	yes	$\Sigma_k \geq 0$	no	no
Convex regr.	yes	$a_j \geq 0$	no	yes
Clustering	no	–	no	no
Classification	yes		usually	sometimes
in general			no	yes

5 The big picture

The three columns below list respectively topics in Statistical Learning, ideas and concepts that are useful in Learning, and topics in Optimization.

Statistical Learning	**Concepts**	**Optimization**
Classification	Parametric/Non-parametric	line minimization
Regression	model	unconstrained optimization
Ranking	Exponential Family models	*Convex Analysis*
Parameter estimation	Entropy and information	convex constrained optimization
(Clustering 535)	Regularization methods (e.g.	(non-convex optimization)
(Density estimation)	Support Vector Machines (SVM), Compressed Sensing)	

I regard as central to this course the concepts in the middle column. Many advances in modern machine learning have come from developing the ideas above, with the novel development then applying to several of the problem classes in the left column. For example, SVM started as a classification method, but now it applies to clustering, regression, and many other problems in supervised learning. More recently, Compressed Sensing started as a regression problem, but it now applies to parameter estimation, forms of "density estimation" and so on.

With respect to the topics in the **optimization** column, the course will proceed sequentially. The concepts in the central column will be introduced when the necessary level of background knowledge is attained. These con-

cepts will be exemplified with topics from the left column, most often with classification or regression. Binary classification being the basic example of a discrete decision problem. For example, after the necessary background in convex analysis, we can study exponential family models, which are intimately related to convexity. Then, we will be ready to tackle information theoretical concepts like entropy and divergence, and their connection to the estimation of exponential family models. These will be then applied to classification, regression and parameter estimation problems.

What I hope you will learn, from the optimization point of view

- How to formulate your problem as an optimization problem. Sometimes the same problem can be set up in several different forms, which all have the same solution but which may differ very much in terms of their difficulty! Sometimes by changing the problem a little one can reduce the difficulty of finding the solution by a lot.
- How to recognize what makes an optimization problem hard. How to choose the best optimization algorithm for your problem.
- How to solve an optimization problem. How and when to write your own code and how and when to choose available optimization code.
- The important and growing role of optimization in Machine Learning and Statistics. Results and algorithms in machine learning and data analysis that are based on optimization (e.g boosting, support vector machines). What is different about optimization for data analysis w.r.t generic optimization.