ECE 598: Interplay between Control and Machine Learning **Spring 2019**

Lecture 4

Empirical Risk Minimization: A General Paradigm for Machine Learning

Lecturer: Bin Hu, Date:01/29/2019

Empirical risk minimization (ERM) is a key paradigm in machine learning. Today we will briefly talk about ERM. In the next few lectures, we will cover control perspectives of stochastic optimization methods for ERM.

4.1 Introduction to ERM

Many supervised learning tasks, including ridge regression, logistic regression, and support vector machines, can be formulated as the following empirical risk minimization problem

$$\underset{x\in\mathbb{R}^p}{\text{minimize}}\, f(x) := \frac{1}{n}\sum_{i=1}^{n} l_i(x) + \lambda\Omega(x)$$

Here one wants to fit some prediction/classification model parameterized by x using the training data. The training set consists of n data points. The task is to fit a model x that works well for all the "unseen" data.

Interpretations for $l_i(x)$. The loss function $l_i(x)$ measures how well x performs on the i-th data point in the training set. If $l_i(x)$ is large, it means that the "loss" on the i-th data point is big and the model x works poorly on this data point. If $l_i(x)$ is small, it means the "loss" on the i-th data point is small and the model x works well on this data point. By minimizing the empirical risk $\frac{1}{n}\sum_{i=1}^{n} l_i(x)$, one expects the model x to work reasonably well for training data. This prevents underfitting.

Importance of $\Omega(x)$. If we allow the model to be arbitrarily complicated, we can obtain zero loss on training data but the model will work poorly on the data that have not been seen. This is called over-fitting. Roughly speaking, the difference between the model performance on the training data and the unseen data is called generalization error. One way to prevent overfitting and induce generalization is to add a regularizer $\Omega(x)$ that measures the model complexity. By adding such a term in the cost function, one expects the complexity of the resultant x is somehow controlled and hence the model x should "generalize" to the unseen data.[1] For example, one can choose $\Omega(x) = \|x\|^2$, and there exists some learning theory (e.g. stability theory) that can be used to explain how such ℓ_2-regularization induces

[1] What we mean is that the model x should work similarly on the training data and the unseen data.

generalization. Confining the search of x on small norm models can help generalization in many situations. Sometime $\Omega(x)$ is used to induce other desired structures. For example, the ℓ_1-regularization is typically used to induce sparsity.

What is λ? In ERM, λ is a hyperparameter which is tuned to trade off training performance and generalization. For the purpose of this course, let's say λ is a fixed positive number. In practice, λ is typically set as a small number between 10^{-8} and 0.1.

4.2 Examples

4.2.1 Ridge regression

The ridge regression is formulated as an ERM problem with the following objective function

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}(a_i^\mathsf{T} x - b_i)^2 + \frac{\lambda}{2}\|x\|^2 \tag{4.1}$$

where $a_i \in \mathbb{R}^p$ and $b_i \in R$ are data points used to fit the linear model x.

- What is this problem about? The purpose of this problem is to fit a linear relationship between a and b. One wants to predict b from a as $b = a^\mathsf{T} x$. The ridge regression gives a way to find such x based on the observed pairs of (a_i, b_i).
- Why is there a term $\frac{\lambda}{2}\|x\|^2$? Again, the term $\frac{\lambda}{2}\|x\|^2$ is just the ℓ_2-regularizer. It confines the complexity of the linear predictors you want to use. The high-level idea is that you want x to work for all (a, b), not just the observed pairs (a_i, b_i). Again, this is called generalization in machine learning. So adding such a term can induce the so-called stability and helps the predictor x to "generalize" for the data you have not seen. You need to take a machine learning course if you want to learn about generalization.
- What is λ? λ is a hyperparameter which is tuned to trade off training performance and generalization. Again, λ is typically set as a small number between 10^{-8} and 0.1.

It is worth mentioning that f is L-smooth and m-strongly convex in this case. It is straightforward to verify that

$$f(x) = \frac{1}{2}x^\mathsf{T}\left(\frac{2}{n}\sum_{i=1}^{n} a_i a_i^\mathsf{T} + \lambda I\right)x - \left(\frac{2}{n}\sum_{i=1}^{n} b_i a_i\right)^\mathsf{T} x + \frac{1}{n}\sum_{i=1}^{n} b_i^2$$

which is a special case of the positive definite quadratic minimization problem. Notice $\frac{2}{n}\sum_{i=1}^{n} a_i a_i^\mathsf{T} + \lambda I > 0$ and hence f is m-strongly convex and L-smooth (why?). Therefore, we can apply gradient method to ridge regression, and obtain a convergence rate $\rho = 1 - \frac{1}{\kappa}$ where κ is the condition number of the positive definite matrix $\frac{2}{n}\sum_{i=1}^{n} a_i a_i^\mathsf{T} + \lambda I$.

4.2.2 ℓ_2-Regularized Logistic regression

The ℓ_2-regularized logistic regression is formulated as an ERM problem with the following objective function

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-b_i a_i^\mathsf{T} x}) + \frac{\lambda}{2}\|x\|^2 \tag{4.2}$$

where $a_i \in \mathbb{R}^p$ and $b_i \in \{-1, 1\}$ are data points used to fit the linear model x.

- What is this problem about? The purpose of this problem is to fit a linear "**classifier**" between a and b. Let's say you have collected a lot of images of cats and dogs. You augment the pixels of any such image into a vector a and wants to predict whether the image is a cat or a dog. Let's say $b = 1$ if the image is a cat, and $b = -1$ if the image is a dog. So you want to predict b based on a. You want to find x such that $b = 1$ when $a^\mathsf{T} x \geq 0$, and $b = -1$ when $a^\mathsf{T} x < 0$. The logistic regression gives a way to find such x based on the observed feature/label pairs of (a_i, b_i). You may want to take a statistics course or a machine learning course if you want to learn more about logistic regression.
- Why is there a term $\frac{\lambda}{2}\|x\|^2$? Again, the term $\frac{\lambda}{2}\|x\|^2$ is the ℓ_2-regularizer. It is used to induce generalization and help x work on all the (a, b) not just the observed data points (a_i, b_i).

The function (4.2) is also L-smooth and m-strongly convex.

4.2.3 Other examples

There are many other convex examples including multi-class logistic regression, support vector machines, and elastic nets. The ERM problems in deep learning involve non-convex loss functions. The optimization of deep learning has not been fully understood and it is an important research topic.

4.3 Finite-sum Structure of ERM

The ERM problem can be rewritten as

$$\min_{x\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{4.3}$$

where $f_i = l_i + \lambda\Omega$. The objective function has a finite sum structure, i.e. $f = \frac{1}{n}\sum_{i=1}^{n} f_i$.

If we apply the gradient method or Nesterov's method, we need to evaluate $\nabla f = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i$ for each iteration. In other words, we need to evaluate the gradient on all the

data points. The computation cost for each iteration scales with $O(n)$. For big data applications, n is typically very large. The per iteration cost of the gradient method and Nesterov's method is high. This motivates the use of stochastic optimization methods that sample one or a small batch of data points for gradient estimate at every iteration. In stochastic optimization, the computation cost for each iteration does not depend on n and scales with $O(1)$. The hope is that there will be a lot of redundancy between data points and these stochastic methods will work well in some average sense in the long run. We will talk about various stochastic optimization methods and represent them as feedback interconnections in the next lecture.