Large Scale Optimization for Machine Learning: Lecture 11

Lecturer: Meisam Razaviyayn Scribe: Jiachuan Chen

Sep 28, 2017

1 Linear Regression and Logistic Regression

First, let's look at the problems of linear regression and logistic regression which have been discussed in first class. Suppose that we have a set of features which impact house prices. We want to estimate the price of a specific house, whose feature is known to us. One way to implement this is to do linear regression.

1.1 linear regression

Assume that we use a linear model to predict the value of y, then the predicted model is $y = \mathbf{w}^T\mathbf{x} + \epsilon$, where noise ϵ has the distribution of $\mathcal{N}(0, \sigma^2)$. Thus, for a given set of features $\mathbf{x}$, we have $y|\mathbf{x} \sim \mathcal{N}(\mu_{\mathbf{x}}, \sigma^2)$ with $\mu_{\mathbf{x}} = \mathbf{w}^T\mathbf{x}$.

By using Maximum Likelihood Estimation, we get following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - y_i)^2 \\ \text{s.t.} \quad & \mathbf{w} \in \mathbb{R}^d \end{aligned}$$

where $\mathbf{x}_i$ represents feature values observation i, and y_i is target variable value for observation i.

In a way, the above problem tries to find the optimal parameter $\mathbf{w}$ such that

it can best predict the observations, i.e., it is to solve for

$$\arg\max_{\mathbf{w}} P(observations|\mathbf{w})$$

After training the model and get a $\mathbf{w}$, we can get predicted price: $\hat{y}_i = \mathbf{w}^{\star T}\mathbf{x}_i$.

Question: what is the loss function in this problem?
Answer: It is given by $\ell(y, \hat{y}) = (y - \hat{y})^2$, or equivalently, we can define $\ell((\mathbf{x}, y), \mathbf{w}) = (\mathbf{w}^T\mathbf{x} - y)^2$

1.2 Logistic regression

Different from linear regression, in logistic regression, the labels we want to predict are discrete, meaning that we want to do classifications.

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are features, where $\mathbf{x}_i \in \mathbb{R}^d$, and $y_1, y_2, \ldots, y_n$ are labels, where $y_i \in \{0, 1\}$. Since we are dealing with discrete case, suppose the logit function is a generalized linear model, i.e.,

$$\log \frac{\mathbb{P}(y = 1|\mathbf{X} = \mathbf{x})}{P(y = 0|\mathbf{X} = \mathbf{x})} = \mathbf{w}^T\mathbf{x}.$$

Thus, $\mathbb{P}(y = 1|\mathbf{X} = \mathbf{x}) = e^{\mathbf{w}^T\mathbf{x}}(1 - P(y = 1|\mathbf{X} = \mathbf{x}))$, meaning that $\mathbb{P}(y = 1|\mathbf{X} = \mathbf{x}) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}}$, and $P(y = 0|\mathbf{X} = \mathbf{x}) = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}}}$.
So by deriving the maximum likelihood estimation of $\mathbf{w}$, we get following optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} \log(1 + e^{\mathbf{w}^T\mathbf{x}_i}) - \sum_{i:y_i=1} \mathbf{w}^T\mathbf{x}_i,$$

or equivalently,

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} \log(1 + e^{\mathbf{w}^T\mathbf{x}_i}) - \sum_{i=1}^{n} y_i\mathbf{w}^T\mathbf{x}_i.$$

Question: What is loss function here?
Answer: It is given by $\log(1 + e^{\mathbf{w}^T\mathbf{x}_i}) - y_i\mathbf{w}^T\mathbf{x}_i$. The reason is that when $y_i = 1$, the function is $\log(1 + e^{\mathbf{w}^T\mathbf{x}_i}) - y_i\mathbf{w}^T\mathbf{x}_i = \log(1 + e^{-\mathbf{w}^T\mathbf{x}_i}) - \mathbf{w}^T\mathbf{x}_i$, you want $\mathbf{w}^T\mathbf{x}_i$ go to ∞ to help you minimize the loss. And when $y_i = 0$,

the function is $\log(1 + e^{\mathbf{w}^T\mathbf{x}_i}) - y_i\mathbf{w}^T\mathbf{x}_i = \log(1 + e^{\mathbf{w}^T\mathbf{x}_i})$, you want $\mathbf{w}^T\mathbf{x}_i$ go to $-\infty$ to minimize the loss.

2 Support Vector Machines

Different from logistic regression, another way of doing classification is support vector machines. Suppose that we have the data of a group of patients, including the features like blood pressure, ages, and so on. And we've also got the data of whether a patient is diabetic or not. We're interested in determining (or predicting) whether a new patient is diabetic or not if we're given his/her feature data. So the task here is to find a mapping h, which maps from features to labels.

If we visualize the process we're doing this task, then we are trying to find a hyperplane which separates the diabetic patients from non-diabetic patients in the graph of their features, and use that hyperplane to predict the new patient's label.

But there're two things which concerned us in finding the hyperplane, the first one is there might be more than one hyperplane which can separate the diabetic and non-diabetic patients. The way we solve this problem is to find a hyperplane with largest margin.

Margin here means the minimum distance from points on both sides to the hyperplane, thus, the optimization problem is formulated as:

$$\begin{aligned} \max_{\mathbf{w},v} \quad & \frac{2}{||\mathbf{w}||} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + v) \geq 1 \quad \forall i = 1, \ldots, n \end{aligned}$$

And it is equivalent to

$$\begin{aligned} \min_{\mathbf{w},v} \quad & ||\mathbf{w}||^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + v) \geq 1 \quad \forall i = 1, \ldots, n \end{aligned}$$

Another question we're concerned is the above optimization problem might be infeasible. In order to deal with this issue, we introduce a soft margin classification SVM. Thus, the soft-margin SVM problem is formulated as

following:

$$\min_{\mathbf{w},v} \quad \lambda||\mathbf{w}||^2 + \frac{1}{n}\sum_{i=1}^{n}\max\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + v)\}$$

Note that the idea behind this is that if $y_i(\mathbf{w}^T\mathbf{x}_i + v) < 1$, which is to say, $(\mathbf{x}_i, y_i)$ is either misclassified or within the margin, $\max\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + v)\}$ is positive, objective function is punished, and if $y_i(\mathbf{w}^T\mathbf{x}_i + v) \geq 1$, which is to say, $(\mathbf{x}_i, y_i)$ is correctly classified and lie outside the margin, then $\max\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + v)\}$ is zero and objective function is not punished on this term.

3 Neural Networks

Neural Networks is another way of doing classifications. In the example of digit classifications, people try to finding a way which make computer recognizes hand-writing digits. Neural networks can help achieve this goal.

In an example of a neural network, there're several layers, including input layer, which takes pixel vector (in the example of digit classification) as input, hidden layers, which may be more than one layer, and output layer, which gives a set of probabilities $(P(y = 0|\mathbf{x}), \ldots, P(y = 9|\mathbf{x}))$ as output.

One thing to mention here is that for each single layer, the mapping(or function) can be simple or complex, and the next layer will take the weighted combination of output from previous layer as input. The optimization problem for neural networks can be written as

$$\min_{h} \sum_{i=1}^{n} l(\mathbf{p}_i(h))$$

where h is hypothesis, or how the neural networks have been established and modeled.

SVM, regression, logistic regression can be viewed as neural networks with one layer.

4 Empirical Risk Minimization Framework

Previous models (logistic regression, linear regression, SVM, neural networks) are all special classes of models in predicting output given input. And if we generalize this process, and set logistic regression as an example in explaining this, we can see that our goal is to predict an output $y \in \mathcal{Y}$ given an input $\mathbf{x} \in \mathcal{X}$, ($\mathcal{X} = \mathbb{R}^d$, $y \in \{0, 1\}$ for logistic regression).

And the set of hypothesis is $\mathcal{H}$ with $h \in \mathcal{H}$ mapping $\mathcal{X}$ to y. In logistic regression, $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, and $\mathbb{P}(y = 1|\mathbf{X} = \mathbf{x}) = \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}}$. If probability is greater than 0.5, the prediction is $y = 1$, and otherwise prediction is 0.

Loss function is $l : (\mathcal{X} \times \mathcal{Y}) \times \mathcal{H} \to \mathbb{R}$.

Data generating distribution is $\mathbb{P}^\star$ with $(\mathbf{x}, y) \sim \mathbb{P}^\star$.

Since the real distribution is $\mathbb{P}^\star$, the expected test error is defined by it: $L(h) \triangleq \mathbb{E}_{\mathbb{P}^\star}(l((\mathbf{x}, y), h))$, and the best hypothesis we hope for is $h^\star \in \arg\min_{h\in\mathcal{H}} L(h)$.

However, in reality, we don't know the real world distribution $\mathbb{P}^\star$. What we can do is use empirical method.

Suppose we have a set of training data, $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, then the empirical risk/training error is $\hat{L}(h) \triangleq \frac{1}{n}\sum_{i=1}^n l((\mathbf{x}_i, y_i), h)$. And the Empirical Risk Minimizer $\hat{h} \in \arg\min_{h\in\mathcal{H}} \hat{L}(h)$.

And thus the expected risk of ERM is $L(\hat{h})$, which denotes the expected test error if we take ERM as the hypothesis in predicting.

4.1 What Set of Hypotheses to Consider?

To solve for ERM $\hat{h}$, we can notice that if the hypothesis become more complex, the empirical risk/training error will decrease. One example is that the linear model can't fit the scatter points very well, and a second order model can fit the points relatively well, and an even higher order model can fit the points exactly and quite well.

However, when the model(hypothesis) exceeds some complexity and becomes more and more complex, the expected test error for $\hat{h}$ will not decrease, on the other hand, it will increase. That is the issue of overfitting.

In the previous example, linear model is somewhat under-fitting with large $L(\hat{h})$, second order model is OK with small $L(\hat{h})$, and 10th order(high order) model is over-fitting with large $L(\hat{h})$.

This is a trade-off between "complexity of hypothesis class" and "expected risk of ERM". Trade-off between $L(\hat{h})$ and $\hat{L}(\hat{h})$.

4.2 Simple Case(Gap between $L(\hat{h})$ and $L(h^\star)$)

Assume that

- $L(h^\star) = \mathbb{E}_{\mathbb{P}^\star}(l((\mathbf{x}, y), h^\star)) = 0$
- $|\mathcal{H}| < \infty$
- Loss function is zero-one loss: $l((x, y), h) = \mathbb{I}(y \neq h(x))$. (Notice that here we assume it is for discrete case and $\mathbb{I}$ is indicator function)

Then, with probability at least $1 - \delta$, the gap $L(\hat{h}) - L(h^\star) \leq \frac{\log|\mathcal{H}| + \log(\frac{1}{\delta})}{n}$.

Remark: above statement means that as long as you have enough number of samples, you can have very small losses.

Proof: Since we want to show

$$P(L(\hat{h}) \leq \frac{\log|\mathcal{H}| + \log(\frac{1}{\delta})}{n}) \geq 1 - \delta$$

and it is equivalent to

$$P(L(\hat{h}) > \frac{\log|\mathcal{H}| + \log(\frac{1}{\delta})}{n}) < \delta$$

So it is enough to show the latter inequality. If we denote ε as a threshold of expected risk error, then

$$P(L(\hat{h}) > \varepsilon) = P(\hat{h} \in \mathcal{B}) \leq P(\exists h \in \mathcal{B} \quad \text{s.t.} \quad \hat{L}(h) = 0) \tag{1}$$

Where $\mathcal{B} = \{h \in \mathcal{H} | L(h) > \varepsilon\}$ is denoted as the set of bad hypothesis (bad hypothesis means the hypothesis h makes the expected test error greater than ε). And the reason for the second inequality to hold is $0 = \hat{L}(h^\star) \geq \hat{L}(\hat{h})(\hat{h} \in \arg\min_{h \in \mathcal{H}} \hat{L}(h))$ and by the fact of $L(h^\star) = 0$, which implies $\hat{L}(\hat{h}) = 0$. Thus, $P(\hat{h} \in \mathcal{B}) \leq P(\exists h \in \mathcal{B} \quad \text{s.t.} \quad \hat{L}(h) = 0)$.
For any fixed $h \in \mathcal{B}$,

$$P(\hat{L}(h) = 0) = (1 - L(h))^n \leq (1 - \varepsilon)^n \leq e^{-\varepsilon n} \tag{2}$$

where the last inequality comes from Taylor expansion: $1 - x \leq e^{-x}$. Thus,

$$\begin{aligned} P(\exists h \in \mathcal{B} \quad \text{s.t.} \quad \hat{L}(h) = 0) \leq & \sum_{h \in \mathcal{B}} P(\hat{L}(h) = 0) \\ \leq & \sum_{h \in \mathcal{B}} e^{-\varepsilon n} \\ = & |\mathcal{B}| e^{-\varepsilon n} \\ \leq & |\mathcal{H}| e^{-\varepsilon n} \end{aligned} \tag{3}$$

where the first inequality is due to union bound and the second inequality is due to (2). Combining (1) and (3), we obtain

$$P(L(h) > \varepsilon) \leq |\mathcal{H}| e^{-\varepsilon n}$$

Setting $\varepsilon = \frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{n}$ will complete the proof.