COS 424: Interacting with Data

Lecturer: Rob Schapire
Scribe: Haipeng Zheng

Lecture #15
April 5, 2007

Recapitulation of Last Lecture

In linear regression, we need to avoid adding too much richness to the model. Therefore we need *feature selection*, or *regularization* to make our fitting curve smoother. Qualitatively, the original linear regression model is an optimization problem of the form

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (\boldsymbol{w} \cdot \boldsymbol{x}_i - y_i)^2$$

And the corresponding regularized version of the same problem is

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (\boldsymbol{w} \cdot \boldsymbol{x}_i - y_i)^2 + \lambda \|\boldsymbol{w}\|_2^2,$$

which is also called *Ridge Regression*. The larger λ is, the smoother, or "simpler" the fitting curve is.

1 Probabilistic View of Regression

A Little Digression: An Introduction to MAP Estimation

Maximum a posteriori (MAP) estimation is a method to estimate parameters by maximizing the posterior probability of the observation.

A brief review of Maximum Likelihood Estimation (MLE) facilitates our derivation of MAP. First, define

- $\boldsymbol{\theta}$ a set of parameters (unknown)
- $\boldsymbol{D}$ data (observed)
- $\Pr[\boldsymbol{D}|\boldsymbol{\theta}]$ the likelihood function, determined by the model of regression, i.e. how the data are generated, given the parameters.

Then the MLE of the parameters $\hat{\boldsymbol{\theta}}$ is

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Pr[\boldsymbol{D}|\boldsymbol{\theta}].$$

If we meditate over the MLE of the parameters, it seems to be paradoxically "backwards". Intuitively, we are given the parameters (or, nature predefines the parameters), and then the observations are generated accordingly. However, in MLE, we are trying to maximize the probability of the *parameters*, which are given and fixed, instead of maximizing the probability of the *observation*! If we change our philosophy and do the opposite thing, i.e. maximize the probability of observation, we are on the track to MAP.

Mathematically speaking, the MAP estimation is given by

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \Pr\left[\boldsymbol{\theta}|\boldsymbol{D}\right]$$

And Bayes Formula says

$$\Pr\left[\boldsymbol{\theta}|\boldsymbol{D}\right] = \frac{\Pr\left[\boldsymbol{D}|\boldsymbol{\theta}\right]Pr[\boldsymbol{\theta}]}{\Pr\left[\boldsymbol{D}\right]}$$

where $\Pr\left[\boldsymbol{\theta}|\boldsymbol{D}\right]$ is called *posterior probability* (or, interchangeably, *a posteriori probability*), $\Pr\left[\boldsymbol{D}|\boldsymbol{\theta}\right]$ is the *likelihood*, $\Pr\left[\boldsymbol{\theta}\right]$ is called *prior probability* (or, interchangeably, *a priori probability*) and $\Pr\left[\boldsymbol{D}\right]$ stands for the probability of data, which is a constant irrelevant to $\boldsymbol{\theta}$.

It is weird to think of parameters $\boldsymbol{\theta}$ as random variables, because usually we believe that parameters are fixed before observation. However, we can also interpret probability as degree of belief or uncertainty. Actually, this is the watershed that separates the Bayesian philosophy from the classical ones.

The basic idea of MAP is simple: maximize the posterior probability of the observation. In order to do this, we introduce the concept of a priori probability. MAP estimation is advantageous when the data set is small and the result of estimation heavily depends on our prior knowledge of the parameters. Yet when the data set is large enough, the effects of prior probability are washed out by the data, hence MAP does not make much difference from MLE.

Regularization in Perspective of MAP

Ridge regression can be interpreted as a form of MAP estimation. First, assume the distribution of Y given $\boldsymbol{X}$ and $\boldsymbol{w}$ is

$$Y|\boldsymbol{X}, \boldsymbol{w} \sim N(\boldsymbol{w} \cdot \boldsymbol{x}, \sigma^2)$$

where σ^2 is a fixed value of variance.

Apply MAP to estimate the parameters, then we need $\boldsymbol{x}$, y and the prior of $\boldsymbol{w}$. We assume

$$w_j \sim N(0, \sigma_0^2), \quad (j = 1, \ldots, n)$$

and $w_j, \quad (j = 1, \ldots, n)$ are generated independently. Or, more concisely

$$\boldsymbol{w} \sim N(\boldsymbol{0}, \sigma_0^2 \boldsymbol{I}).$$

Then the probability density function of $\boldsymbol{w}$, given $(\boldsymbol{X}, Y)$ is

$$\begin{aligned} & p(\boldsymbol{w}|(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)) \\ = \; & \frac{p(y_1, \ldots, y_m|\boldsymbol{w}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)p(\boldsymbol{w}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)}{p(y_1, \ldots, y_m|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)} \\ \propto \; & \prod_{i=1}^{m} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\boldsymbol{w}_i \cdot \boldsymbol{x}_i - y_i)^2}{2\sigma^2}\right)\right] \cdot \frac{1}{(\sqrt{2\pi}\sigma_0)^n} \exp\left(-\frac{\|\boldsymbol{w}\|_2^2}{2\sigma_0^2}\right). \end{aligned}$$

Hence the log likelihood function is given by

$$\begin{aligned} & \log\left(p(\boldsymbol{w}|(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))\right) \\ = \; & C - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (\boldsymbol{w}_i \cdot \boldsymbol{x}_i - y_i)^2 - \frac{1}{2\sigma_0^2} \|\boldsymbol{w}\|_2^2 \end{aligned}$$

where C is a constant. Therefore, the MAP of the parameter is

$$\begin{aligned}\hat{\boldsymbol{w}} &= \arg\max_{\boldsymbol{w}} \quad \log\left(f_{\boldsymbol{w}|(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_m,y_m)}\right) \\ &= \arg\min_{\boldsymbol{w}} \quad \frac{1}{2\sigma^2}\sum_{i=1}^{m}(\boldsymbol{w}_i\cdot\boldsymbol{x}_i-y_i)^2+\frac{1}{2\sigma_0^2}\|\boldsymbol{w}\|_2^2 \\ &= \arg\min_{\boldsymbol{w}} \quad \sum_{i=1}^{m}(\boldsymbol{w}_i\cdot\boldsymbol{x}_i-y_i)^2+\lambda\|\boldsymbol{w}\|_2^2\end{aligned}$$

where $\lambda = \sigma^2/\sigma_0^2$.

Conclusion: Regularization can be interpreted as a form of MAP estimation. For different a priori distribution, we have different forms of regularization terms. Note that regularization is *not* scale invariant.

2 Other Methods of Regularization

Although independent Gaussian prior is often used, we have some other methods of regularization, which can be quite useful under different situations. One of the most commonly used methods is called L_1-*Regularization*, or *Lasso Regularization*, specified by the optimization problem

$$\min_{\boldsymbol{w}}\sum_{i=1}^{m}(\boldsymbol{w}\cdot\boldsymbol{x}_i-y_i)^2+\lambda\|\boldsymbol{w}\|_1.$$

This type of regularization corresponds to a different prior distribution — *Laplacian distribution*, specified by $\exp\left(-C\|\boldsymbol{w}\|_1\right)$, where $\|\boldsymbol{w}\|_1 = \sum_{j=1}^{n}|w_j|$.

Remarks on Lasso Regularization:

- Advantages: Lasso Regularization, with properly selected λ, usually leads to very sparse estimated vector $\hat{\boldsymbol{w}}$, i.e. a large number of components of vector $\hat{\boldsymbol{w}}$ turn out to be zeros. When the dimension of the problem n is very large, Lasso Regularization is preferable because it can effectively reduce the number of parameters from numerous available features.
- Disadvantages: Lasso Regularization can be cumbersome in theoretical analysis, owing to the awkward indifferentiablility of the absolute value function. Practically, it is also somewhat more challenging to implement.

3 Interpretation and Choice of λ

The parameter λ is one of the most significant parameters in regularization. It controls the trade-off of many aspects of the problem.

In classification problems, we are always facing a trade-off between *minimizing the training error* and *generating a simpler classifier*. Here in regression, λ controls a similar trade-off between the *fit to the data of the fitting curve* and the *smoothness of the fitting curve*. This fact is demonstrated in Figure 1.

Observations on Figure 1: When λ is large, we have very bad fit and good smoothness, i.e. the overall fit is very poor, but the performance of any individual curve is similar to

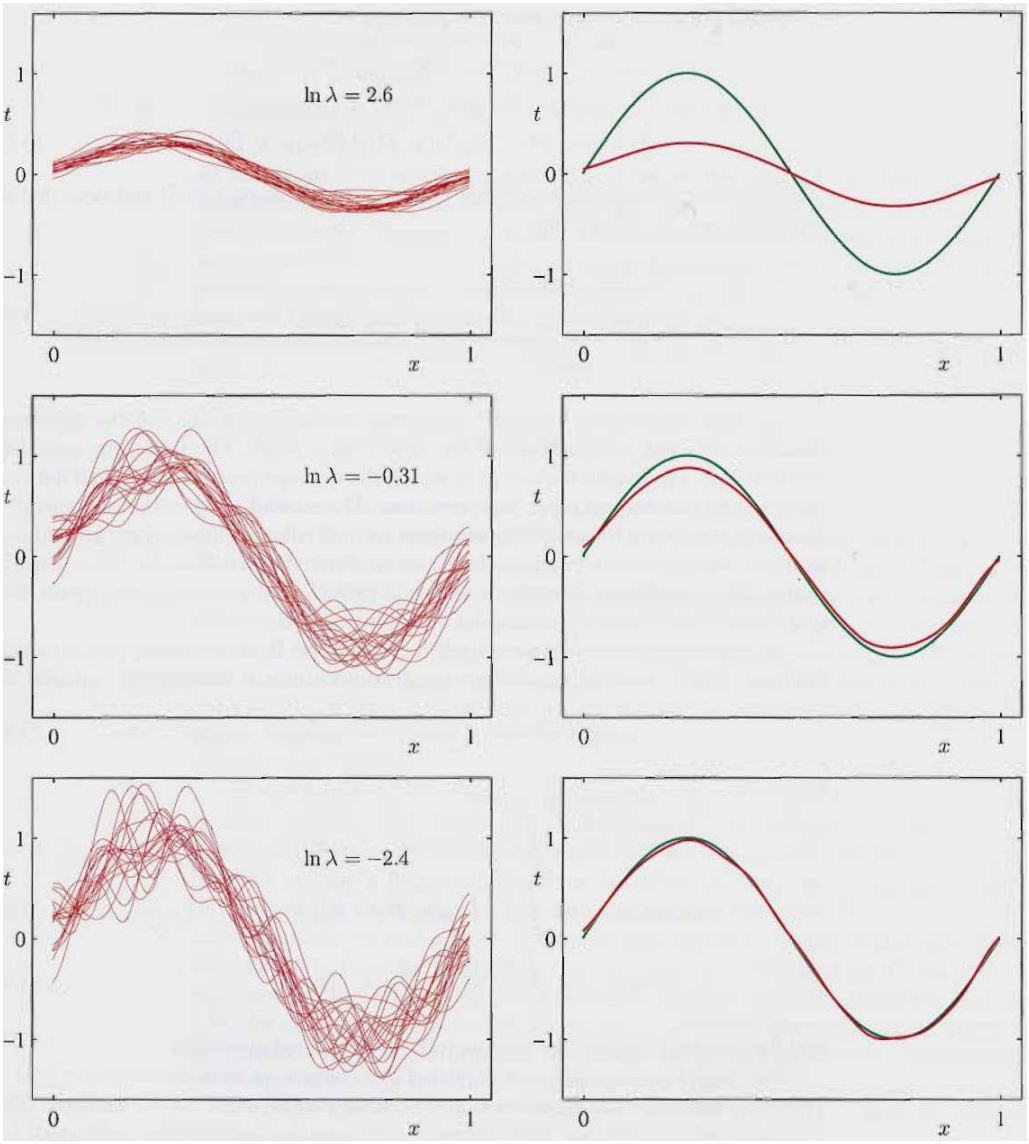

Figure 1: Illustration of the dependence of bias and variance on model complexity, governed by a regularization parameter λ, using a sinusoidal data set. There are $L = 100$ data sets, each having $N = 25$ data points, and there are 24 Gaussian basis functions in the model so that the total number of parameters is $M = 25$ including the bias parameter. the left column shows the result of fitting the model to the data sets for various values of $\ln \lambda$ (for clarity, only 20 of the 100 fits are shown). The right column shows the corresponding average of the 100 fits (red) along with the sinusoidal function from which the data sets were generated (green).

one another; on the contrary, when λ is small, we have very good fit yet bad smoothness, i.e. the overall fit is good yet the performance of any individual curve fluctuates wildly.

Actually, we can quantitatively describe the observation above, and find out the mathematical expression for the "fit to the data" and "smoothness".

Define $\hat{f}_S(x)$ as an estimation to $f(x) = E[Y|X]$, and $\hat{f}_S$ depends on the sample S, i.e. $(x_1, y_1), \ldots, (x_m, y_m)$. Then define $g(x)$ as the average of all the fitting curves (expectation over the samples)

$$g(x) = E_S[\hat{f}_S(x)]$$

Then we can use this concept to decompose the estimation error further. In the expression

$$E[(\hat{f}(X) - Y)^2] = E[(Y - f(X))^2] + E[(\hat{f}(X) - f(X))^2]$$

the term $E[(Y - f(X))^2]$ stands for the *intrinsic noise* and the other term $E[(\hat{f}(X) - f(X))^2]$ can be further decomposed. The overall performance

$$E_{X,Y,S}[(\hat{f}_S(X) - Y)^2] = E_{X,Y}[(Y - f(X))^2] + E_X[(g(X) - f(X))^2] + E_{X,S}[(\hat{f}_S(X) - g(X))^2]$$

where $E_{X,Y}[(Y - f(X))^2]$ stands for the intrinsic noise (irrelevant to sample S), $E_X[(g(X) - f(X))^2]$ stands for how good the average of all the estimator is (this term is called $(bias)^2$), and $E_{X,S}[(\hat{f}_S(X) - g(X))^2]$ stands for the *variance* of the estimator. In Figure 1, the $(bias)^2$ can be interpreted as the difference between the green curve (real model) and the red curve (average performance of all the estimators), and the variance can be interpreted as the average difference between an individual red curve and the average of all the red curves.

The typical relationships between parameter λ and $(bias)^2$, variance, noise and error are illustrated in Figure 2.

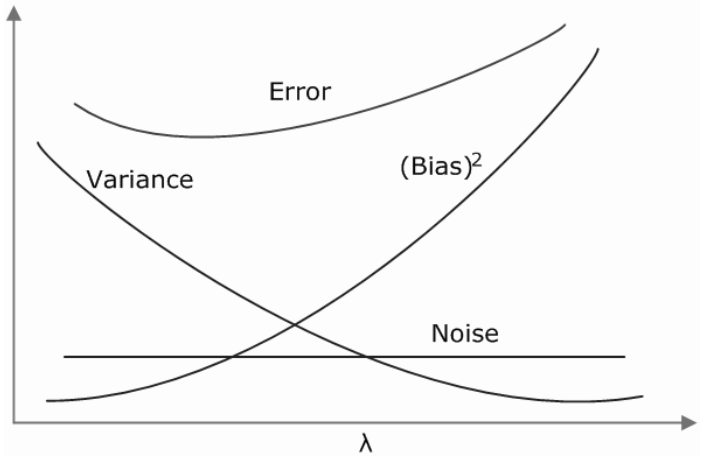

Figure 2: The relationship between λ and $(bias)^2$, variance, noise, and error. The variance is monotonously decreasing with respect to λ, and the bias is increasing. The noise (intrinsic)is irrelevant to λ. And the error is the sum of bias, variance and noise, hence it has a "U" shape with respect to λ. In practice, we are seeking the lowest point of the error curve, which corresponds to the optimal value of λ.

Owing to the "U" shape of the error curve with respect to λ, we are facing a problem of how to find the optimal value of λ that minimizes the error. Here we introduce a method called *10-fold Cross Validation*. This algorithm can be described as follows:

1. Divide the data evenly into 10 random, disjoint blocks.
2. For $j = 1, 2, \ldots, 10$, hold out the jth block as the test set, using the rest as the training set, and calculate the jth empirical error.
3. Calculate the average of the 10 individual errors and get the empirical error.
4. Find the value of λ that minimizes the empirical error.

4 Logistic Regression

Formulate the Problem

Actually, a number of classification algorithms can be generalized to regression, including *Decision Tree Algorithm*, *K-nearest Neighbors Algorithm*, etc. We stop our discussion of linear regression and switch to another type of regression.

In classification problems, we simply want the test error to be small. However, in some cases, we need to estimate the *probability* of some events, say, the presence of precipitation tomorrow, or the existence of a disease. In other words, instead of simply classifying the data points, we intend to estimate the probability of the data points being in each class.

Here is our basic model:Given data $\boldsymbol{X}$, Y, i.e. $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$,where $\boldsymbol{X} \in \mathbb{R}^n$ and $Y \in \{0, 1\}$ or $\{-1, 1\}$. Then we want to estimate $\Pr[Y|\boldsymbol{X}]$.

As in the case of linear regression, the problem can be simplified into the optimization problem

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (\boldsymbol{w} \cdot \boldsymbol{x}_i - y_i)^2$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$.This approach makes sense because

$$\begin{aligned} E[(\boldsymbol{w} \cdot \boldsymbol{X} - Y)^2] &= E[(\boldsymbol{w} \cdot \boldsymbol{X} - f(\boldsymbol{X}))^2] \\ &= E[(\boldsymbol{w} \cdot \boldsymbol{X} - E[Y|\boldsymbol{X}])^2] \\ &= E[(\boldsymbol{w} \cdot \boldsymbol{X} - \Pr[Y = 1|\boldsymbol{X}])^2]. \end{aligned}$$

Therefore the above approach is tantamount to assuming

$$\Pr[Y = 1|\boldsymbol{x}] \simeq \boldsymbol{w} \cdot \boldsymbol{x}.$$

It turns out that we impose a restriction on $\boldsymbol{w} \cdot \boldsymbol{x}$, namely, $\boldsymbol{w} \cdot \boldsymbol{x} \in [0, 1]$, which may not always be true. Moreover, since $Y \in \{0, 1\}$, the noise is far from being Gaussian, as required by our linear model.

It is true that this linear model is still used in many occasions, due to many preferable properties of linearity. However, we can conceive another method by squashing all the real numbers into the range of $[0, 1]$, hence remove the restriction on $\boldsymbol{w} \cdot \boldsymbol{x}$, because in this case

$$\Pr[Y = 1|\boldsymbol{x}] \simeq \sigma(\boldsymbol{w} \cdot \boldsymbol{x}).$$

So we need to find a function that maps all the real numbers into $[0, 1]$. The function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

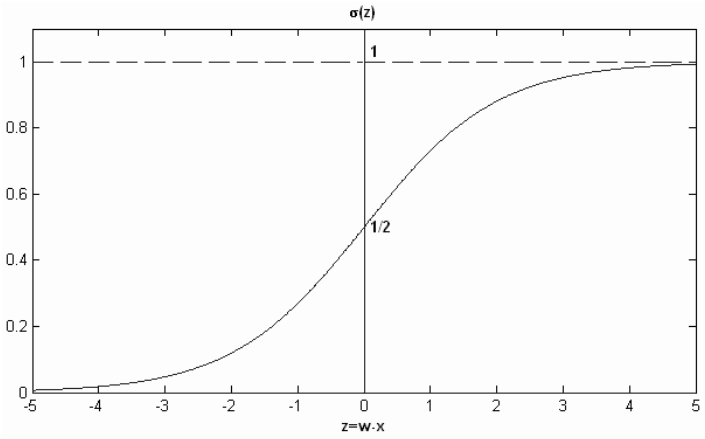

Figure 3: The sigmoidal shaped function $1/(1+e^{-z})$

is our favorite choice, as shown in Figure 3.

The function $\sigma(z)$ has many desirable properties. One of the most useful one is

$$\sigma(-z) = 1 - \sigma(z).$$

Now we switch to the $Y \in \{-1, 1\}$ system. Then

$$\Pr[Y = -1|\boldsymbol{x}] = 1 - \sigma(\boldsymbol{w} \cdot \boldsymbol{x}) = \sigma(-\boldsymbol{w} \cdot \boldsymbol{x}).$$

Therefore

$$\Pr[Y = y|\boldsymbol{x}] = \sigma(y\ \boldsymbol{w} \cdot \boldsymbol{x}).$$

Solve the Problem: Estimate $\boldsymbol{w}$

Now we are facing an optimization problem

$$\min_{\boldsymbol{w}} \quad \sum_{i=1}^{m} (\sigma(\boldsymbol{w} \cdot \boldsymbol{x}_i) - y_i)^2.$$

Unfortunately the objective function is not convex and lots of local minima exist. Moreover, the assumption that $Y|\boldsymbol{X}$ is Gaussian is inappropriate here. All these lead to a highly undesirable problem to solve. Notwithstanding these facts, people still use it a lot.

However, instead of solving the above problem, we try to solve

$$\begin{aligned}
\hat{\boldsymbol{w}} &= \arg\max_{\boldsymbol{w}} \quad \prod_{i=1}^{m} \Pr[y_i|\boldsymbol{x}_i, \boldsymbol{w}] \\
&= \arg\max_{\boldsymbol{w}} \quad \sum_{i=1}^{m} \log\left(\Pr[y_i|\boldsymbol{x}_i, \boldsymbol{w}]\right) \\
&= \arg\min_{\boldsymbol{w}} \quad -\sum_{i=1}^{m} \log\left(\sigma(y_i\ \boldsymbol{w} \cdot \boldsymbol{x}_i)\right) \\
&= \arg\min_{\boldsymbol{w}} \quad \sum_{i=1}^{m} \log\left(1 + \exp\left(y_i\ \boldsymbol{w} \cdot \boldsymbol{x}_i\right)\right).
\end{aligned}$$

Finally, we come up with an elegant optimization problem:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \quad \sum_{i=1}^{m} \log\left(1 + \exp\left(y_i \ \boldsymbol{w} \cdot \boldsymbol{x}_i\right)\right)$$

and the estimation of the probabilities are given by

$$\begin{aligned} \Pr[Y = 1] &= \sigma(\boldsymbol{w} \cdot \boldsymbol{x}) \\ \Pr[Y = -1] &= \sigma(-\boldsymbol{w} \cdot \boldsymbol{x}). \end{aligned}$$

This is called *Logistic Regression*. It possesses many desirable properties, as listed below:

1. The objective function is convex in variable $\boldsymbol{w}$, so there are no local minima.
2. Notice the term $y_i \ \boldsymbol{w} \cdot \boldsymbol{x}_i$, which represents the *margin*. This means that Logistic Regression is also a margin based algorithm, like SVM.
3. Since Logistic Regression is based on a linear combination of the inputs, it can be combined with a number of methods, including regularization, the kernel trick, basis expansions, etc.

Owing to the properties above,Logistic Regression is widely used and turns out to be very effective.

Moreover, Logistic Regression can also be used as an classification method. The algorithm is described as follows:

> Given $\boldsymbol{x}$, predict $y = +1$ if $\sigma(\boldsymbol{w} \cdot \boldsymbol{x}) > 1/2$, or, equivalently, $\boldsymbol{w} \cdot \boldsymbol{x} > 0$. Otherwise, predict $y = -1$.

From the algorithm above, we conclude that Logistic Regression builds up a linear classifier, like SVM.

The connection between Naive Bayes and Logistic Regression is to be explored in the next lecture.