Notes on statistical machine learning

February 17, 2014

1 From linear to logistical regression and SVMs: a natural progression

This section describes the progressive development of the underlying link leading from linear regression to logistic regression and SVMs with linear kernels. We show the logical underlying flow that naturally leads to this progression of concepts.

1.1 Linear Regression

In linear regression, our hypothesis is that of a linear function (input/output relationship) which maps the feature vectors to the observations. Our goal is to find a $\theta \in \mathbb{R}^N$ under the hypothesis

$$y_j = \sum_{i=0}^{N} \theta_i x_j^i, \qquad \forall j \in \{1, 2, \dots M\}. \tag{1}$$

Here

$$\begin{aligned} N &= \text{ number of features,} \\ M &= \text{ number of samples,} \\ y_j &= \text{ output scalar of the j-th observation,} \\ x_j &= \text{ feature vector for the j-th sample.} \end{aligned} \tag{2}$$

We add a dimension to the feature vector to account for the constant term as follows.

$$x_j^0 = 1, \forall j.$$

We may formulate this regression problem as an optimization problem via the cost function

$$J(\theta) = \frac{1}{2M} \sum_{j=1}^{M} (y_j - \sum_{i=0}^{N} \theta_i x^i)^2 \tag{3}$$

As this cost function is convex, it can be optimized using standard techniques such as gradient descent or other approaches which use gradient information as well. Here the gradient can be analytically computed to be:

$$\frac{\partial J}{\partial \theta_i} = \frac{1}{M}\sum_{j=1}^{M}\Big(y_j - h_\theta(x_j)\Big)x_j^i, \tag{4}$$

$$\text{where} \quad h_\theta(x) := \theta^T x = \sum_{i=1}^{N}\theta_i x^i. \tag{5}$$

We note that we may solve for the optimal θ using using linear algebra (via the construction of a matrix inverse of a matrix built up using the factor matrix). However the optimization method enables us to obtain the solution for large scale feature vectors (for which the matrix inversion methods would not fit in RAM).

1.2 Linear regression with regularization

If high variance is an issue (ref. Sec.4) we may introduce a regularization term in the cost function as follows

$$J(\theta) = \frac{1}{2M}\sum_{j=1}^{M}(y_j - \sum_{i=0}^{N}\theta_i x^i)^2 + \frac{1}{2}\lambda\sum_{i=1}^{M}{\theta_i}^2. \tag{6}$$

The use of the l_2 norm is termed ridge regression. Note that alternative approaches to regularization cast the ridge regression as

$$J(\theta) = \frac{1}{2M}\sum_{j=1}^{M}(y_j - \Sigma_{i=0}^{N}\theta_i x^i)^2 + \frac{1}{2*M}\lambda\sum_{i=1}^{M}{\theta_i}^2. \tag{7}$$

In this case the gradient takes the form (for the cost function in (6))

$$\frac{\partial J}{\partial \theta_i} = \frac{1}{M}\sum_{j=1}^{M}\Big(y_j - h_\theta(x_j)\Big)x_j^i + \lambda\theta_i. \tag{8}$$

2 Logistic regression via linear regression

Logistic regression is a classification technique which can be used to train a data set to separate two classes - in effect it yields the probability of belonging to one of the two classes $\{1, 2\}$.

We denote

$$\begin{aligned} p &:= \text{probability of belonging to class 1,} \\ 1-p &:= \text{probability of belonging to class 2.} \end{aligned} \tag{9}$$

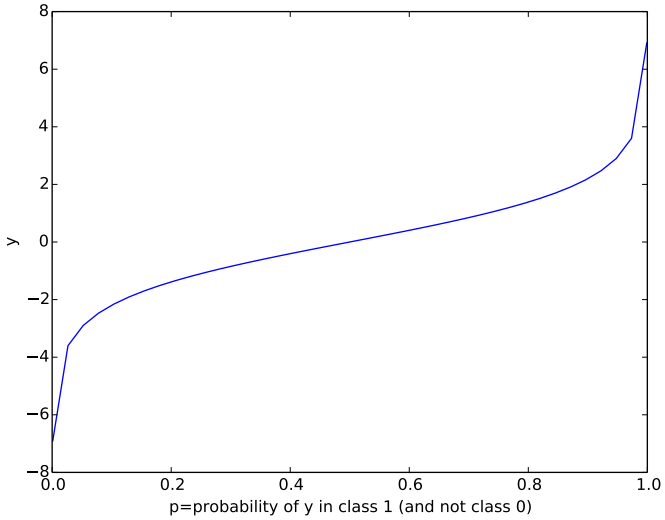

Figure 1: mapping between probability value and y for logistic regression

For each feature vector we must obtain an output of 0 (class 1) or 1 (class 2). Noting that linear regression requires an output which can reside in $\mathbb{R}$ we note that this can be obtained via the map

$$y_j = \log\left(\frac{p_j}{1-p_j}\right) \quad \in (-\infty, +\infty), \tag{10}$$

where p_j is the probability for the feature vector x_j ($\forall j \in \{1, 2, \ldots, M\}$) to belong to class 1 (Ref. Fig. 2).

Thus we can construct a new linear regression problem where the hypothesis is

$$y_j \sim \sum_{i=0}^{N} x_j^i \theta_i. \tag{11}$$

This leads to

$$p(x_j) = \frac{1}{1+\exp\{-\theta^T x_j\}} \quad =: h_\theta(x_j). \tag{12}$$

Thus we formulate an optimization problem using the cost function which, when solved, yields the logistic regression. One approach is to formulate the loglikelyhood function

$$\begin{aligned} L(\theta) &= \sum_{j=1}^{M} \Big[y_j \log(p(x_j)) + (1-y_j)\log(1-p(x_j)) \Big], \\ &= \sum_{j=1}^{M} \Big[y_j \log(h_\theta(x_j)) + (1-y_j)\log(1-h_\theta(x_j)) \Big]. \end{aligned} \tag{13}$$

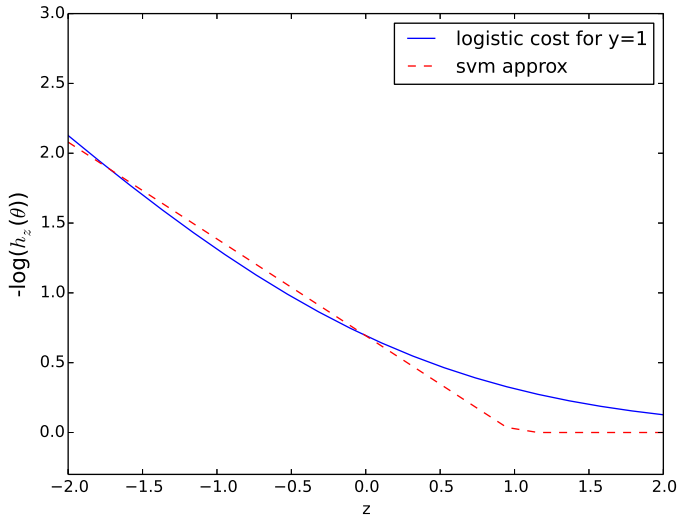

Figure 2: cost if $y_j = 1$ for various values of $z = \theta^T x_j$

Hence the corresponding optimization (minimization problem) takes the form

$$J(\theta) = -\frac{1}{M}\sum_{j=1}^{M}\Big[y_j \log(h_\theta(x_j)) + (1-y_j)\log(1-h_\theta(x_j))\Big]. \tag{14}$$

$$\begin{aligned} &= \frac{1}{M} \times \begin{cases} \displaystyle\sum_{i \text{ where } y_j=1} \log\left(1+\exp(-\theta^T x_j)\right) \\ \quad + \\ \displaystyle\sum_{i \text{ where } y_j=0} \log \frac{(1+\exp(-\theta^T x_j))}{\exp(-\theta^T x_j)} \end{cases} \\ &= \frac{1}{M} \times \begin{cases} \displaystyle\sum_{i \text{ where } y_j=1} \log\left(1+\exp(-\theta^T x_j)\right) & (\text{make } \theta^T x_j \text{ large}) \\ \quad + \\ \displaystyle\sum_{i \text{ where } y_j=0} \log\left(1+\exp(\theta^T x_j)\right) & (\text{make } \theta^T x_j \text{ small / [large in abs val and -ve]}) \end{cases} \end{aligned} \tag{15}$$

The cost function plots for the logistic and an SVM approximation (Ref. Sec.3) are as indicated in Fig. 2.

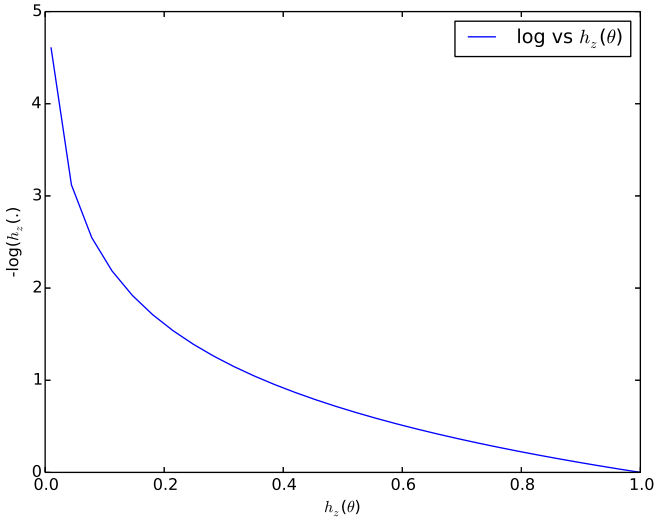

Figure 3: cost if $y_j = 1$ for various values of $p = Pr(y_j = 1) = h_\theta(x_j)$

2.1 Logistic regression with regularization

For the case of logistic cost function with regression the optimal cost takes the form

$$J(\theta) = -\frac{1}{M}\sum_{j=1}^{M}\Big[y_j \log(h_\theta(x_j)) + (1-y_j)\log(1-h_\theta(x_j))\Big] + \lambda\sum_{i=1}^{N}\theta_i^2. \tag{16}$$

3 SVM

From the figure 2 we note that a piecewise constant function can be used to approximate the cost function. This leads to the following form of the cost function for a 2 class classification problem

$$J(\theta) = -\frac{1}{M}\sum_{j=1}^{M}\Big[y_j cost_1(\theta^T x_j) + (1-y_j)cost_0(\theta^T x_j)\Big] + \lambda\sum_{i=1}^{N}\theta_i^2. \tag{17}$$

This form of the cost function inspires the formulation of a generalized cost function with generalized feature vectors as follows. Given M observations we generate M dimensional feature vectors (explained further below). The resulting cost function would take the form

$$J(\theta) = -\frac{1}{M}\sum_{j=1}^{M}\Big[y_j cost_1(\theta^T f_j) + (1-y_j)cost_0(\theta^T f_j)\Big] + \lambda\sum_{i=1}^{N}\theta_i^2. \tag{18}$$

where the feature vectors f_j are constructed as follows. Given a kernel function $\phi(\alpha, \beta)$ and observations $x_j, j \in \{1, 2, \dots M\}$ we define

$$\begin{aligned} & f_j = f_j^i, i \in \{1, 2, \dots M\}, \\ \text{where} \qquad & f_j^i := \phi(x_i, x_j). \end{aligned} \tag{19}$$

By dividing the cost function (16) by λ we can rewrite it as

$$J(\theta) = C \sum_{j=1}^{M} \Big[y_j cost_1(\theta^T f_j) + (1 - y_j) cost_0(\theta^T f_j) \Big] + \sum_{i=1}^{M} \theta_i^2. \tag{20}$$

An alternate approach to the optimization problem is to reformulate the cost to be minimized as follows

$$J(\theta) = \sum_{i=1}^{M} \theta_i^2, \tag{21}$$

$$\begin{aligned} \text{subject to} \quad & \theta^T f_j \geq 1, \quad y_j = 1, \\ & \theta^T f_j \leq -1, \quad y_j = 0. \end{aligned} \tag{22}$$

This assumes that the parts of the cost functions apart from the regularization are zero.

4 Bias-Variance studies and Learning Curves

Understanding if the hypothesis needs to be modified is an important step in the analysis which can be performed in order to determine if:

1. a larger training data set is required
2. the regularization penalty is to be increased (high variance case) or decreased (high bias case)
3. a higher order hypothesis is required

Typically if the hypothesis is extremely general then the training errors will be small but the generalization power of the trained model will be low owing to overfitting. This is termed a case of *high variance*. In contrast to this, a hypothesis which is extremely specific will also not generalize well. In addition, in this case the training errors would also likely be larger than in the case of a high variance model. The training and cross validation cost functions can be plotted to determine

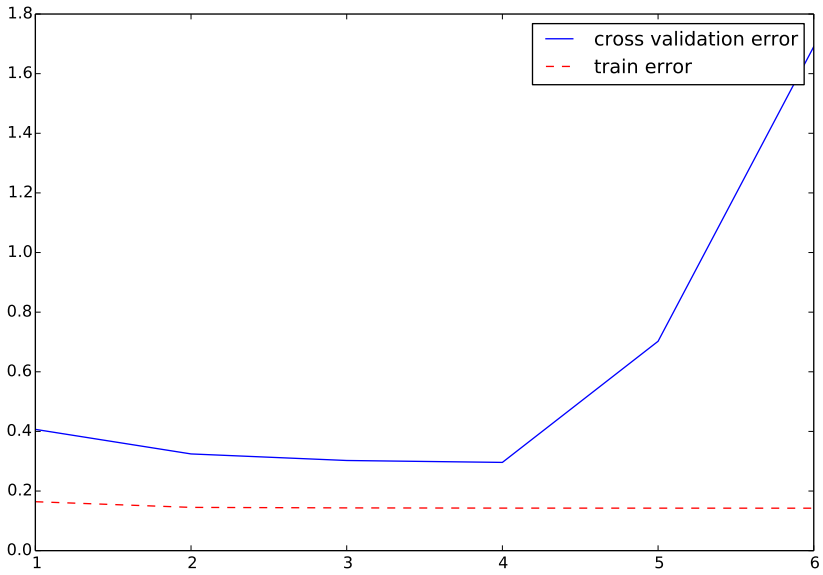

Figure 4: Bias variance curves

if there is a bias or variance. In addition the errors can be plotted against the number of data points used in training in order to understand if more data will help in improving the accuracy of the algorithm.

4.1 Bias-Variance curves

In this section we generate data using the true model

$$y = 4\log(x) + \epsilon, \qquad \epsilon \sim N(0, 1). \tag{23}$$

We increase the order of the polynomial in the hypothesis from linear (order = 1) to order = 6. The training and cross validation(in this case == test) errors (cost functions) are plotted as shown in Fig. 4. At lower order hypothesis the cross validation error and the training error are large (high variance case). At higher order hypothesis the cross validation error is large but the training error is small (high bias case).

4.2 Learning curves

In order to determine if a larger training data set will improve the accuracy of the algorithm we plot the training and cross validation errors as a function of the size of the training set used. These are termed *learning curves*. We demonstrate the two cases of these curves as follows. The data is generated from the true model:

$$y = x^3 + 2x^2 + 8x + 5 + \epsilon, \qquad \epsilon \sim N(0, 1), \tag{24}$$

and the training is carried out by shuffling the training data and then using an increasing number of training data points in order to fit the hypothesis. In the case of high bias we use a first order model to fit the model in (24). This yields the result in Fig. 5. Note that in this case the training and cross validation errors are typically large and do not reduce with an increase in the number of training data points. Thus an increase in the number of data points used in training is not anticipated to yield improvements in the generalization accuracy of the hypothesis. The hypothesis in this case can be improved by

- choosing a more general model
- using a larger number of features
- reducing the regularization penalty

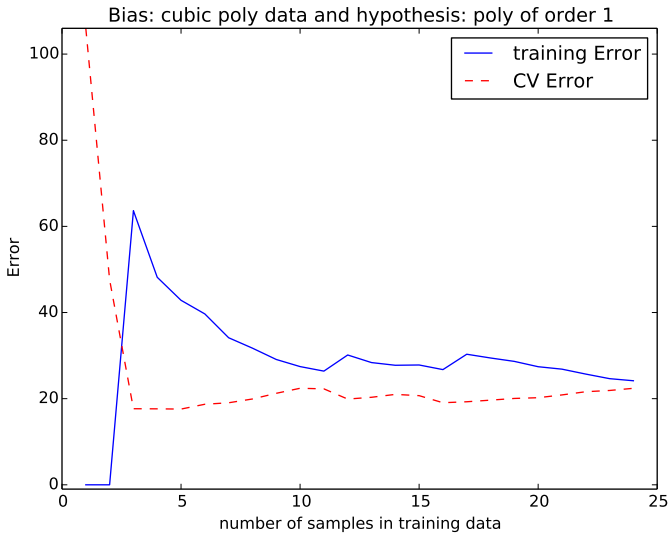

Figure 5: Learning curve. Bias case: 3rd order model and first order hypothesis

In the case of high variance we use a fifth order model to fit the model in (24). This yields the result in Fig. 6. Note that in this case the training errors are low and the cross validation errors converge to the training errors as the number of samples is increased. In this case it might be worthwhile collecting a larger amount of data if we wish to reduce the generalization (cross validation) errors. **However, once the training and cross validation errors are close, further data collection is not of much value**. The hypothesis in this case can be improved by

- choosing a more general model
- reducing the number of features used
- gathering more training data
- increasing the regularization penalty

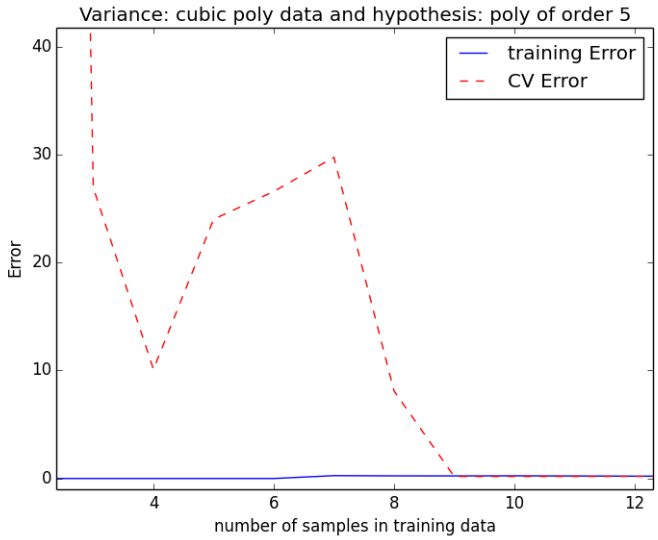

Figure 6: Learning curve. Variance case: 3rd order model and fifth order hypothesis