6.867 Machine learning

Mid-term exam

October 18, 2006

(2 points) Your name and MIT ID:

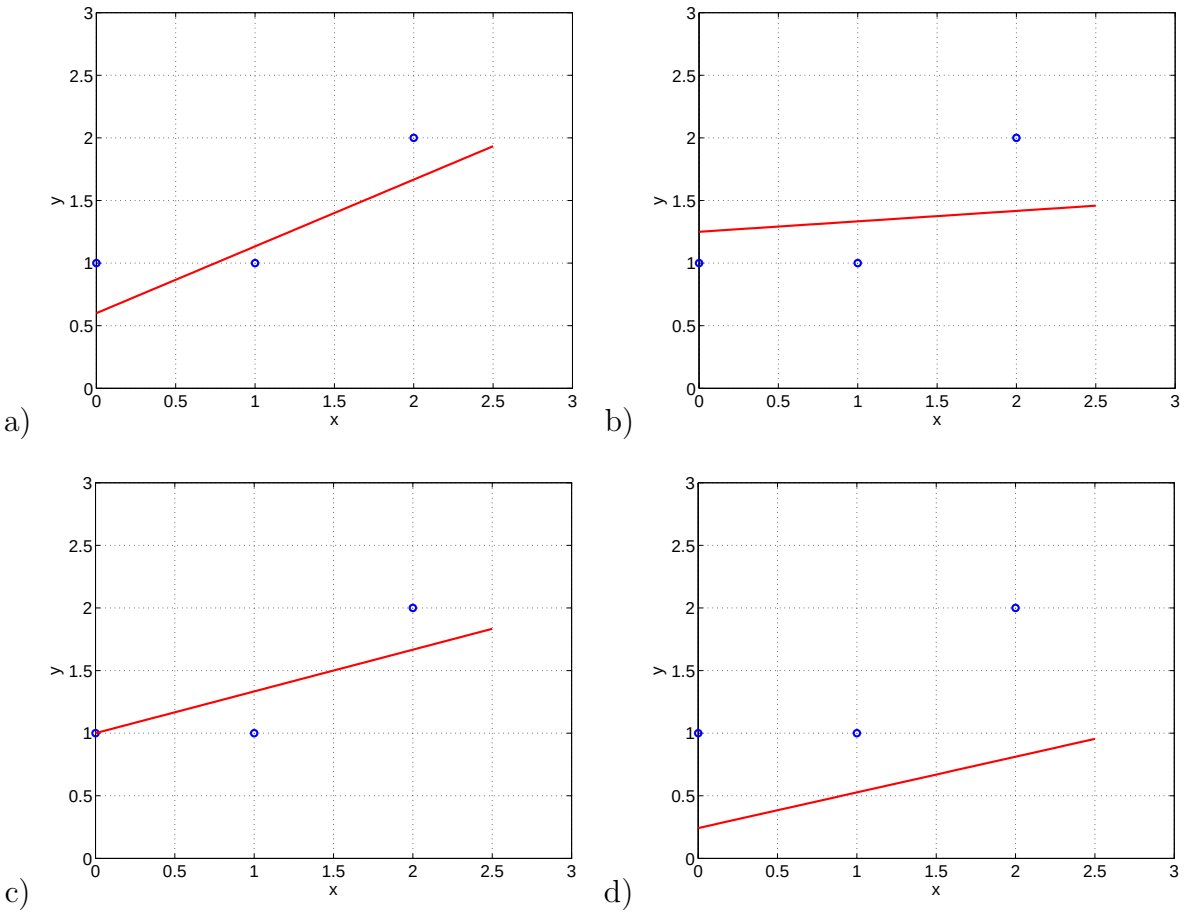

Figure 1: Plots of linear regression results with different types of regularization

Problem 1

Figure 1 plots linear regression results on the basis of only three data points. We used various types of regularization to obtain the plots (see below) but got confused about which plot corresponds to which regularization method. Please assign each plot to one (and only one) of the following regularization method.

1.1 **(2 points)** $\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2$ where $\lambda = 1$ | c

1.2 **(4 points)** $\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2$ where $\lambda = 10$ | b

Briefly explain why

The slope is strongly regularized making the regression function flat. Since we don't regularize the offset parameter it is still possible to lift the flat function in the middle of the responses.

Note: since θ_0 is not regularized, the sum of positive and negative errors will be exactly zero at the optimal setting of the parameters

$$\sum_{t=1}^{3}(y_t - \hat{\theta} x_t - \hat{\theta}_0) = 0$$

1.3 **(2 points)** $\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2)$ where $\lambda = 1$ | a

1.4 **(2 points)** $\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2)$ where $\lambda = 10$ | d

Problem 2

We are trying to solve a regression problem with kernel linear regression models using different degree polynomial kernels. Our regression problem is a little unusual in the sense that the training input points are 1-dimensional and fixed, $x_1, \ldots, x_n$ (all distinct). Our task is to find the underlying function values at the same points and specifically at x_1. The underlying function is $f^*(x) = E\{y|x\}$, where the expectation is over the underlying distribution (pdf) $p(y|x)$ governing how y depends probabilistically on x. We have no knowledge of $f^*(x)$ or $p(y|x)$ beyond real valued training responses $y_1, \ldots, y_n$, sampled from $p(y|x)$ at the training inputs.

Let's assume that our linear regression model (not in the kernel form) is given by

$$f(x; \theta, \theta_0) = \theta^T \phi(x) + \theta_0$$

where $\phi(x)$ is the feature vector corresponding to our choice of the kernel function. We will estimate the parameter θ and θ_0 (or α and θ_0 in a kernel form) by minimizing the mean

squared prediction error without regularization:

$$\frac{1}{n}\sum_{i=1}^{n}\Big(y_i - f(x_i;\theta,\theta_0)\Big)^2$$

We will then use $f(x_1;\hat{\theta},\hat{\theta}_0)$ as an estimator of $f^*(x_1)$. In other words, all we care about is the prediction at x_1. **Assume that $\mathbf{n = 3}$.**

2.1 **(3 points)** Write down an expression for the bias of this estimator. Your expression should involve just $f(x_1;\hat{\theta},\hat{\theta}_0)$, $E\{\cdot\}$, and $f^*(x_1)$, as well as an explanation for what the expectation is over.

$$\textit{Bias at } x_1 = E\{\, f(x_1;\hat{\theta},\hat{\theta}_0)\,\} - f^*(x_1)$$

where the expectation is over the responses y_1, y_2, and y_3 corresponding to the three fixed training points x_1, x_2, and x_3. Each response is sampled from $p(y_i|x_i)$, independently from the others.

2.2 **(2 points)** Which degree polynomial kernel would we need to get zero training error, i.e., fit the three training responses perfectly?

2

You need a quadratic feature vector to perfectly fit three points.

2.3 **(2 points)** Would we get an unbiased estimator at x_1 if we achieve zero training error **(Y/N)**?

Y

Since we are fitting the responses perfectly, our estimator $f(x_1;\hat{\theta},\hat{\theta}_0)$ simply returns y_1, the training response. The expected value of this response is by definition $f^*(x_1)$. The estimator is therefore unbiased.

2.4 **(3 points)** Suppose the noise variance at x_1 is $E\{(y_1 - f^*(x_1))^2\} = \sigma^2$. What is the variance of our "zero training error estimator", again at x_1?

Since the estimator returns the observed training response y_1 at x_1, $f(x_1;\hat{\theta},\hat{\theta}_0) = y_1$, and is unbiased so that $E\{f(x_1;\hat{\theta},\hat{\theta}_0)\} = f^*(x_1)$, we have

$$E\left\{\Big(f(x_1;\hat{\theta},\hat{\theta}_0) - E\{f(x_1;\hat{\theta},\hat{\theta}_0)\}\Big)^2\right\} = E\{(y_1 - f^*(x_1))^2\} = \sigma^2$$

which is just the noise variance. So our estimator is as noisy as the responses (seriously overfitting).

Problem 3

We are trying to solve a classification problem with support vector machines. In our problem there are only a few positive training examples and we are certain that they are classified correctly. We also have a large number of negative training examples, some of which may be misclassified. We'd like to modify the basic dual form of the SVM optimization problem,

$$(1) \qquad \text{maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$(2) \qquad \text{subject to} \quad \alpha_i \geq 0, \ \sum_{i=1}^{n} \alpha_i y_i = 0$$

to better solve this type of problem. We would like to ensure that we *won't misclassify any of the positive examples* but *could misclassify some of the negative examples.* We believe you have to introduce additional parameter(s) (or constants for the purpose of solving the quadratic programming problem) in order to achieve this.

In your solution, please use I^+ to index positively labeled examples ($y_i = +1$) and I^- for negative examples ($y_i = -1$). In other words, $i \in I^+$ means that $y_i = +1$, and $|I^+|$ is the number of positive examples.

3.1 **(6 points)** Your solution must be in the dual form. You can refer to (1) and (2) above.

Maximize

(1), as above

subject to

(2) and $\alpha_i \leq C^-$ for $i \in I^-$ (negative examples).

In other words, we limit how strongly the margin constraints are enforced for the negative examples. Positive examples have no such limit and the classifier will have to satisfy the margin constraints exactly for the positive examples.

3.2 **(6 points)** Check (Y/N) which of the following alternative criteria would work for optimizing your new parameters. We have underlined any differences between the alternatives.

( ) We train your SVM algorithm $|I^+|$ times, each time leaving out one of the positive examples, and testing the classifier on the left out example. The parameter(s) are set to minimize the resulting number of misclassified examples.

Since we are only focusing on how well the positive examples are classified, setting $C^- = 0$ would be optimal. As a result, we wouldn't enforce any classification constraints on the negative examples.

(X) We train your SVM algorithm $|I^-|$ times, each time leaving out one of the negative examples, and testing the classifier on the left out example. The parameter(s) are set to minimize the resulting number of misclassified examples.

Briefly explain why this would or would not work:

The optimization problem described above strictly enforces the classification constraints for the positive examples. Thus no matter how we set C^- it won't be possible to misclassify any of them on the training set. However, focusing solely on the negative examples will not try to gauge how well we generalize in terms of classifying positive examples. We are simply trying to generalize well in terms of correctly classifying negative examples (by optimizing this CV error) with the constraint that we still have have to classify all the positive training examples correctly.

(X) We train your SVM algorithm n times, each time leaving out one of the examples, positive or negative, and testing the classifier on the left out example. The constant is set to minimize the resulting number of misclassified examples.

This is the standard CV error and would work here as well.

Problem 4

A student in a machine learning course claimed that the points in Figure 2a can be separated with "almost a linear kernel". Hard to believe, we responded, since the points are clearly not linearly separable. But the student insisted. The "almost a linear kernel" they had in mind was the following normalized kernel:

$$K_{norm}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|}$$

4.1 **(2 points)** What are the feature vectors corresponding to this kernel?

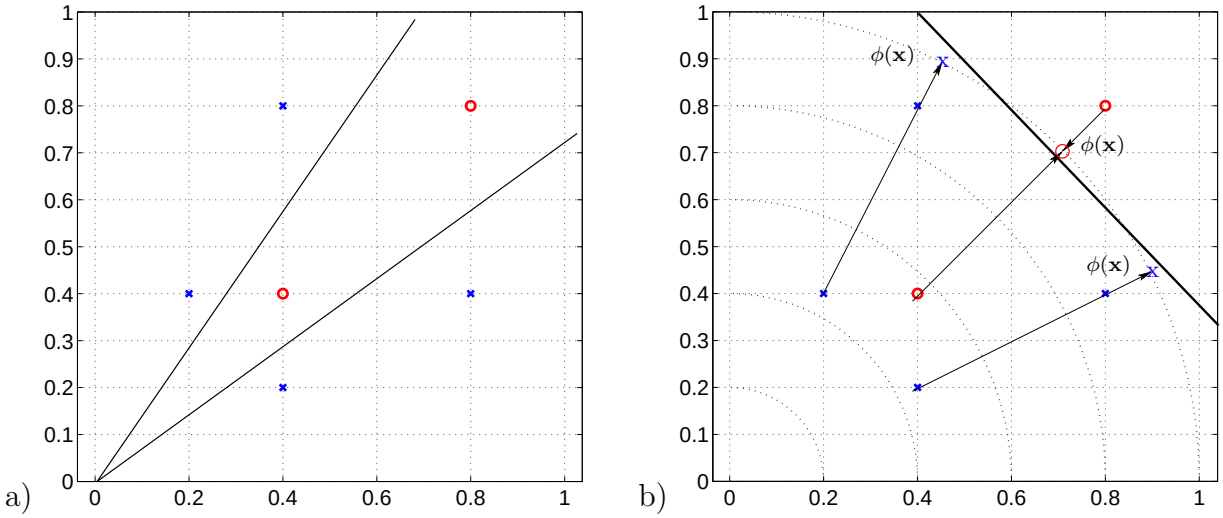

Figure 2: a) Points that should be separable with a normalized linear kernel. b) feature space with the original points overlaid with their original coordinate values.

The feature vectors are just $\phi(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|$. These are two dimensional vectors (although they only vary along the unit circle).

4.2 **(4 points)** Using Figure 2b (right), graphically map the points to their new feature representation using the figure as the feature space.

The points are mapped radially to the unit circle (the largest dotted circle in the figure)

4.3 **(4 points)** Draw the resulting maximum margin decision boundary in the feature space. Use the same Figure 2b (right). The student was right, the points are separable!

See the figure.

4.4 **(2 points)** Does the value of the discriminant function corresponding to your solution change if we scale any point, i.e., evaluate it at $s\,\mathbf{x}$ instead of $\mathbf{x}$ for some $s > 0$? **(Y/N)**

N

4.5 **(4 points)** Draw the decision boundary in the original input space resulting from the normalized linear kernel. Use Figure 2a (left).

The maximum margin boundary in the feature space crosses the unit circle in two places. These are the feature vectors right on the boundary. Since points that are already normalized map onto themselves in the feature space, these are also points right on the boundary in the original space. We know that scaling doesn't affect the discriminant function and thus you can simply draw lines from these points on the unit circle to/from the origin to get the decision boundary in the original space.

Problem 5

There are many criteria for active learning. In particular, in the context of linear regression, we derived such criteria by assuming that the underlying model was also linear (in the feature space). One of the resulting criteria was based on finding points where our predictions varied the most (relative to resampled training sets from an assumed underlying model).

We will focus here on simple active learning methods for classification tasks with the perceptron algorithm. We assume that you can only ask labels for the training examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (those we don't already have labels for). The labels are fixed once revealed so there's no reason to query the same point multiple times. The perceptron algorithm, in response to mistakes, updates its parameters according to

$$\theta \leftarrow \theta + y_t \mathbf{x}_t \quad \text{iff } y_t \theta^T \mathbf{x}_t \leq 0$$

5.1 **(2 points)** In our setting, would it be useful to get a label for a point that we can classify correctly? **(Y/N)**

N

There are two plausible ways of applying the perceptron algorithm in this context. You either run the algorithm until it converges with the labels you already have, or consider each example only once in the order in which they were asked to be labeled. In either case, a point that does not result in an update, i.e., is not misclassified, won't have any immediate effect on θ and therefore not on the next point to be labeled.

5.2 **(6 points)** Given the current θ we have to select which example $\mathbf{x}_1, \ldots, \mathbf{x}_n$ would be the most useful to label. Check *all* of the following criteria you believe would work as a selection criterion. We could select the point $\mathbf{x}_t$ with

( ) the largest norm $\|\mathbf{x}_t\|$

( ) the largest value of $|\theta^T \mathbf{x}_t|$

(X) the smallest value of $|\theta^T \mathbf{x}_t|$

Briefly explain why your chosen criterion (criteria) would work in our active learning setting:

We are looking for points that are potentially misclassified. These are the ones that are close to the boundary. We don't know whether they are misclassified before seeing the label but we can measure how close they are to the boundary, which is what $|\theta^T \mathbf{x}|$ *tries to do (you could normalize this by* $\|\theta\|$ *so as to get an actual distance to the current boundary).*