6.867 Machine learning

Mid-term exam

October 8, 2003

(2 points) Your name and MIT ID:

Problem 1

We are interested here in a particular 1-dimensional linear regression problem. The dataset corresponding to this problem has n examples $(x_1, y_1), \ldots, (x_n, y_n)$, where x_i and y_i are real numbers for all i. Part of the difficulty here is that we don't have access to the inputs or outputs directly. We don't even know the number of examples in the dataset. We are, however, able to get a few numbers computed from the data.

Let $\mathbf{w}^* = [w_0^*, w_1^*]^T$ be the least squares solution we are after. In other words, $\mathbf{w}^*$ minimizes

$$J(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - w_0 - w_1 x_i)^2$$

You can assume for our purposes here that the solution is unique.

1. **(4 points)** Check each statement that must be true if $\mathbf{w}^* = [w_0^*, w_1^*]^T$ is indeed the least squares solution

   ( ) $(1/n) \sum_{i=1}^{n} (y_i - w_0^* - w_1^* x_i) y_i = 0$
   ( ) $(1/n) \sum_{i=1}^{n} (y_i - w_0^* - w_1^* x_i)(y_i - \bar{y}) = 0$
   ( ) $(1/n) \sum_{i=1}^{n} (y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0$
   ( ) $(1/n) \sum_{i=1}^{n} (y_i - w_0^* - w_1^* x_i)(w_0^* + w_1^* x_i) = 0$

   where $\bar{x}$ and $\bar{y}$ are the sample means based on the same dataset.

2. **(4 points)** There are several numbers (statistics) computed from the data that we can use to infer $\mathbf{w}^*$. These are

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \;\; \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \;\; C_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$
$$C_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}), \;\; C_{yy} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

   Suppose we only care about the value of w_1^*. We'd like to determine w_1^* on the basis of only two numbers (statistics) listed above. Which two numbers do we need for this?

3. Here we change the rules governing our access to the data. Instead of simply getting the statistics we want, we have to reconstruct these from examples that we query. There are two types of queries we can make. We can either request additional randomly chosen examples from the training set, or we can query the output corresponding to a specific input that we specify. (We assume that the dataset is large enough that there is always an example whose input x is close enough to our query).

   The active learning scenario here is somewhat different from the typical one. Normally we would assume that the data is governed by a linear model and choose the input points so as to best recover this assumed model. Here the task is to recover the best fitting linear model to the data but we make no assumptions about whether the linear model is appropriate in the first place.

   (2 points) Suppose in our case the input points are constrained to lie in the interval $[0, 1]$. If we followed the typical active learning approach, where we assume that the true model is linear, what are the input points we would query?

   (3 points) In the new setting, where we try to recover the best fitting linear model or parameters $\mathbf{w}^*$, we should (choose only one):

   ( ) Query inputs as you have answered above

   ( ) Draw inputs and corresponding outputs at random from the dataset

   ( ) Use another strategy since neither of the above choices would yield satisfactory results

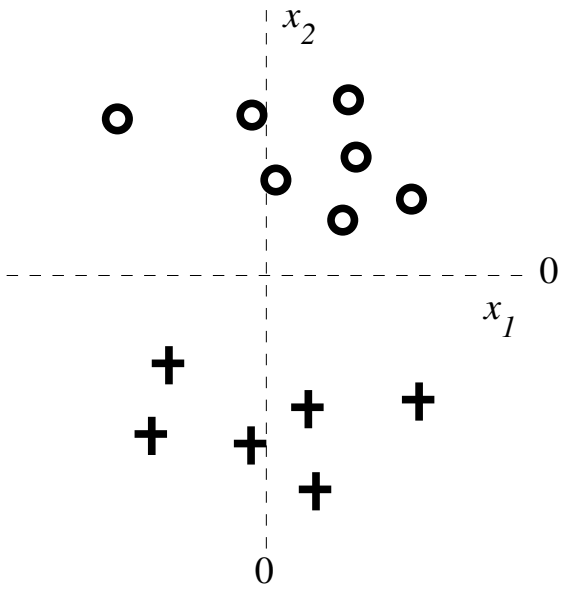
(4 points) Briefly justify your answer to the previous question

Problem 2

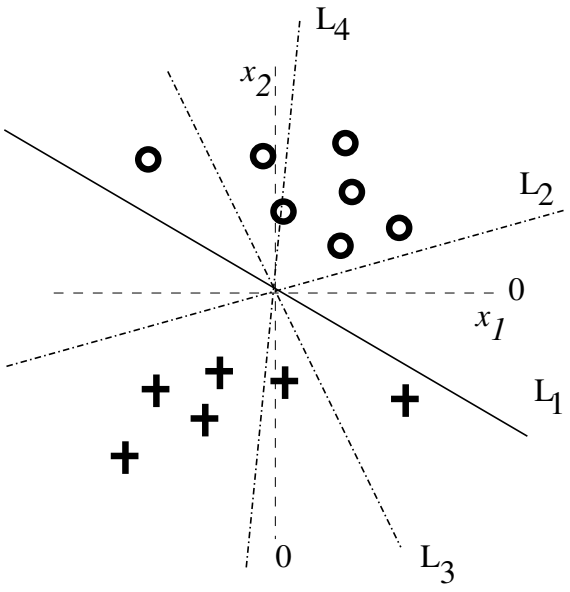
In this problem we will refer to the binary classification task depicted in Figure 1(a), which we attempt to solve with the simple linear logistic regression model

$$\hat{P}\left(y=1|\mathbf{x},w_1,w_2\right)=g(w_1x_1+w_2x_2)=\frac{1}{1+\exp(-w_1x_1-w_2x_2)}$$

(for simplicity we do not use the bias parameter w_0). The training data can be separated with zero training error - see line L_1 in Figure 1(b) for instance.



(a) The 2-dimensional data set used in Problem 1



(b) The points can be separated by L_1 (solid line). Possible other decision boundaries are shown by L_2, L_3, L_4.

1. **(6 points)** Consider a regularization approach where we try to maximize

$$\sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, w_1, w_2) - \frac{C}{2} w_2^2$$

for large C. Note that **only** w_2 is penalized. We'd like to know which of the four lines in Figure 1(b) could arise as a result of such regularization. For each potential line L_2, L_3 or L_4 determine whether it can result from regularizing w_2. If not, explain very briefly why not.

- $L2$

- $L3$

- $L4$

2. **(4 points)**If we change the form of regularization to one-norm (absolute value) and also regularize w_1 we get the following penalized log-likelihood

$$\sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, w_1, w_2) - \frac{C}{2} \left(|w_1| + |w_2| \right).$$

Consider again the problem in Figure 1(a) and the same linear logistic regression model $\hat{P}(y = 1|\mathbf{x}, w_1, w_2) = g(w_1 x_1 + w_2 x_2)$. As we increase the regularization parameter C which of the following scenarios do you expect to observe (choose only one):

( ) First w_1 will become 0, then w_2.

( ) w_1 and w_2 will become zero simultaneously

( ) First w_2 will become 0, then w_1.

( ) None of the weights will become exactly zero, only smaller as C increases

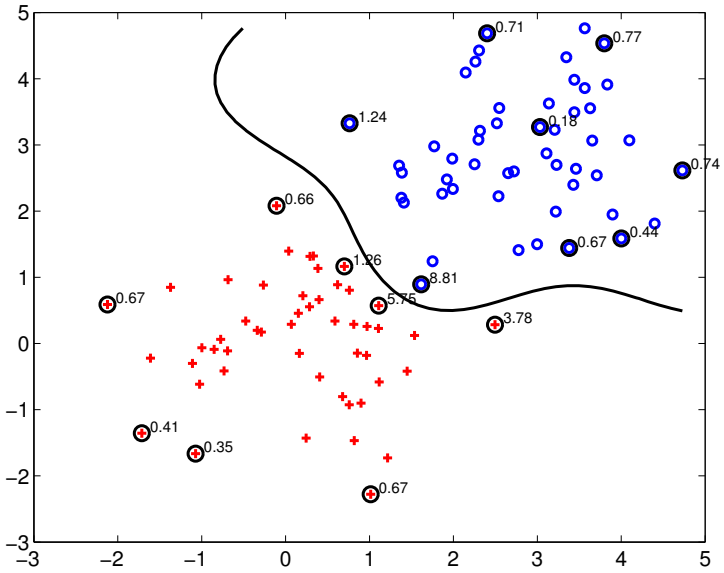


Figure 1: A 2-dim classification problem, the resulting SVM decision boundary with a radial basis kernel, as well as the support vectors (indicated by larger circles around them). The numbers next to the support vectors are the corresponding coefficients $\hat{\alpha}$.

Problem 3

Figure 1 illustrates a binary classification problem along with our solution using support vector machines (SVMs). We have used a radial basis kernel function given by

$$K(\mathbf{x}, \mathbf{x}') = \exp\{ -\|\mathbf{x} - \mathbf{x}'\|^2/2 \}$$

where $\|\cdot\|$ is a Euclidean distance and $\mathbf{x} = [x_1, x_2]^T$. The classification decision for any $\mathbf{x}$ is made on the basis of the sign of

$$\hat{\mathbf{w}}^T \phi(\mathbf{x}) + \hat{w}_0 = \sum_{j \in \mathrm{SV}} y_j \hat{\alpha}_j K(\mathbf{x}_j, \mathbf{x}) + \hat{w}_0 = f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$$

where $\hat{\mathbf{w}}$, $\hat{w}_0$, $\hat{\alpha}_i$ are all coefficients estimated from the available data displayed in the figure and SV is the set of support vectors. $\phi(\mathbf{x})$ is the feature vector derived from $\mathbf{x}$ corresponding to the radial basis kernel. In other words, $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$. While technically $\phi(\mathbf{x})$ is an infinite dimensional vector in this case, this fact plays no role in the questions below. You can assume and treat it as a finite dimensional vector if you like.

The support vectors we obtain for this classification problem (indicated with larger circles in the figure) seem a bit curious. Some of the support vectors appear to be far away from the decision boundary and yet be support vectors. Some of our questions below try to resolve this issue.

1. **(3 points)** What happens to our SVM predictions $f(\mathbf{x}; \hat{\alpha}, \hat{w}_0)$ with the radial basis kernel if we choose a test point $\mathbf{x}_{far}$ far away from any of the training points $\mathbf{x}_j$ (distances here measured in the space of the original points)?

2. **(3 points)** Let's assume for simplicity that $\hat{w}_0 = 0$. What equation do all the training points $\mathbf{x}_j$ have to satisfy? Would $\mathbf{x}_{far}$ satisfy the same equation?

3. **(4 points)** If we included $\mathbf{x}_{far}$ in the training set, would it become a support vector? Briefly justify your answer.

4. **(T/F – 2 points)** Leave-one-out cross-validation error is always small for support vector machines. ☐

5. **(T/F – 2 points)** The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers ☐

6. **(T/F – 2 points)** Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could in principle be reproduced with an SVM and a polynomial kernel of degree less than or equal to three ☐.

7. **(T/F – 2 points)** The decision boundary implied by a generative model (with parameterized class-conditional densities) can be optimal only if the assumed class-conditional densities are correct for the problem at hand ☐

Problem 4

Consider the following set of 3-dimensional points, sampled from two classes:

	x_1	x_2	x_3
labeled '1':	1,	1,	-1
	0,	2,	-2
	0,	-1,	1
	0,	-2,	2

	x_1	x_2	x_3
labeled '0':	1,	1,	2
	0,	2,	1
	1,	-1,	-1
	1,	-2,	-2

We have included 2-dimensional plots of pairs of features in the "Additional set of figures" section (figure 3).

1. **(4 points)** Explain briefly why features with higher mutual information with the label are likely to be more useful for classification task (in general, not necessarily in the given example).

2. **(3 points)** In the example above, which feature (x_1, x_2 or x_3) has the highest mutual information with the class label, based on the training set?

3. **(4 points)** Assume that the learning is done with quadratic logistic regression, where

   $$P(y = 1|\mathbf{x}, \mathbf{w}) = g(w_0 + w_1 x_i + w_2 x_j + w_3 x_i x_j + w_4 x_i^2 + w_5 x_j^2)$$

   for some pair of features (x_i, x_j). Based on the training set given above, which pair of features would result in the lowest training error for the logistic regression model?

4. **(T/F – 2 points)** From the point of view of classification it is always beneficial to remove features that have very high variance in the data

5. **(T/F – 2 points)** A feature which has zero mutual information with the class label might be selected by a greedy selection method, if it happens to improve classifier's performance on the training set

Problem 5

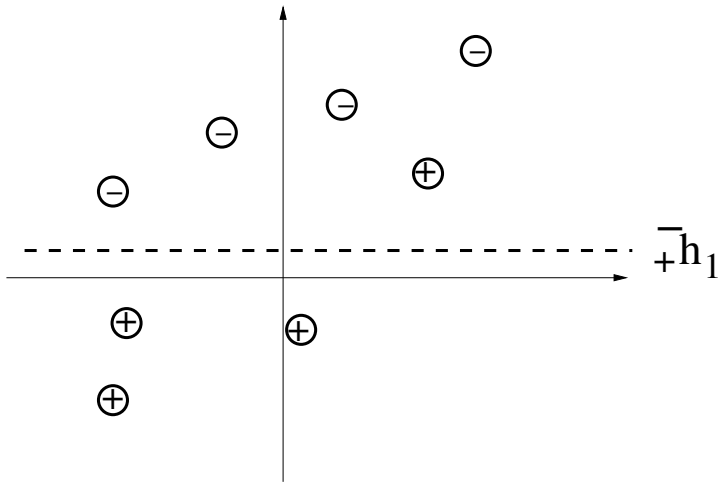


Figure 2: h_1 is chosen at the first iteration of boosting; what is the weight α_1 assigned to it?

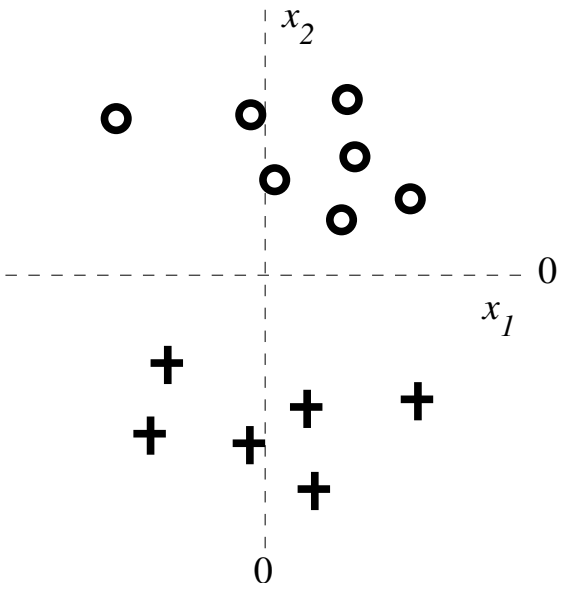
1. **(3 points)** Figure 2 shows a dataset of 8 points, equally divided among the two classes (positive and negative). The figure also shows a particular choice of decision stump h_1 picked by AdaBoost in the first iteration. What is the weight α_1 that will be assigned to h_1 by AdaBoost? (Initial weights of all the data points are equal, or 1/8.)

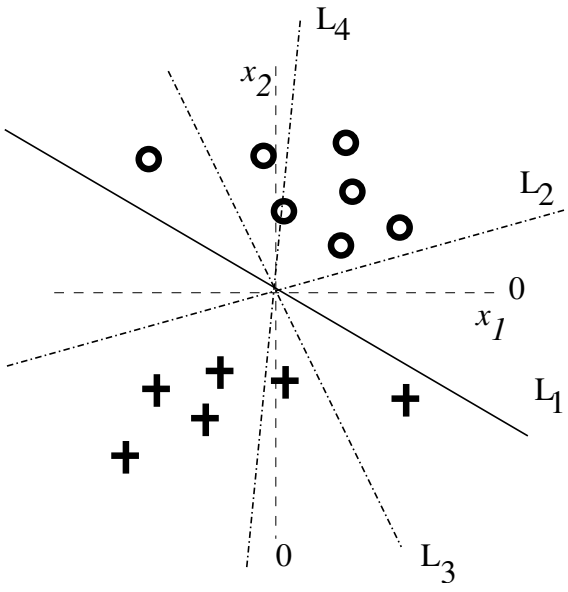
2. **(T/F – 2 points)** AdaBoost will eventually reach zero training error, regardless of the type of weak classifier it uses, provided enough weak classifiers have been combined.

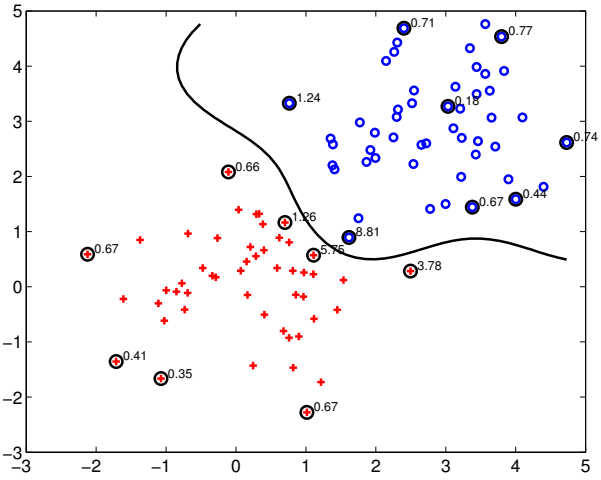
3. **(T/F – 2 points)** The votes α_i assigned to the weak classifiers in boosting generally go down as the algorithm proceeds, because the weighted training error of the weak classifiers tends to go up

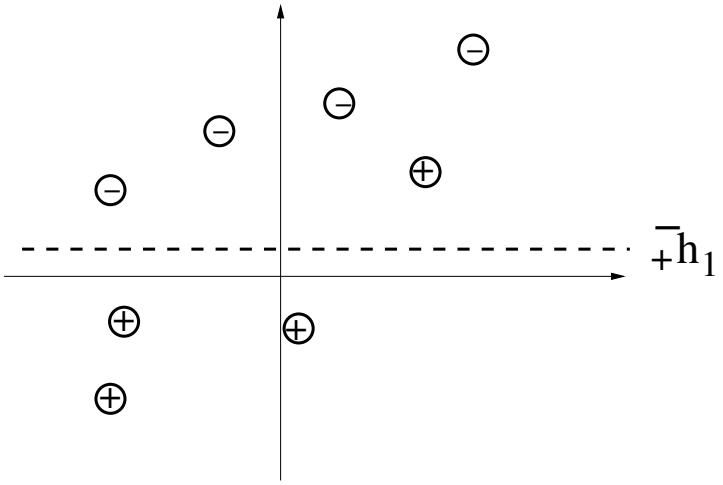
4. **(T/F – 2 points)** The votes α assigned to the classifiers assembled by AdaBoost are always non-negative

Additional set of figures









there's more ...

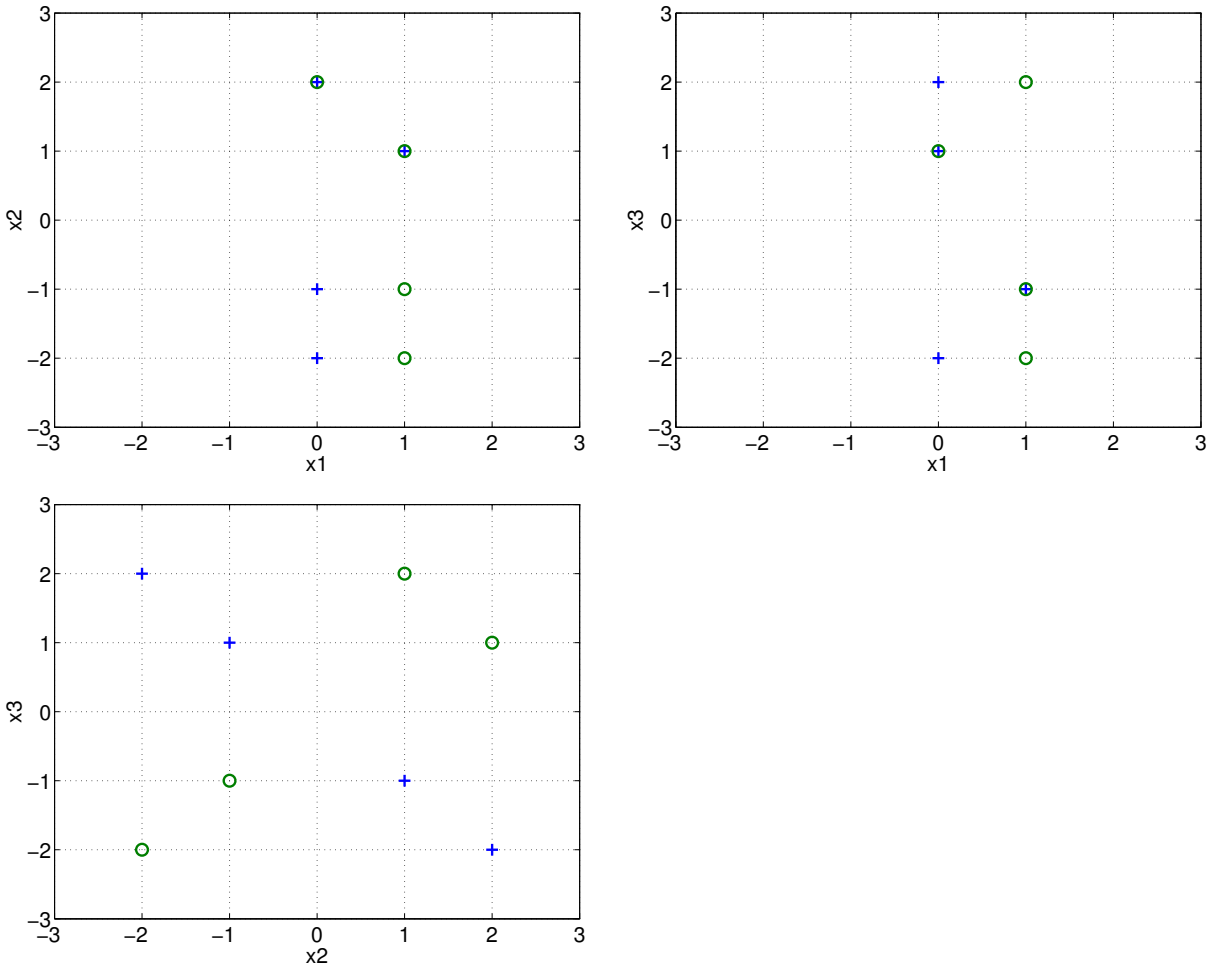


Figure 3: 2-dimensional plots of pairs of features for problem 4. Here '+' corresponds to class label '1' and 'o' to class label '0'.

.