Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.036 Introduction to Machine Learning

**Midterm exam (March 18, 2014)**

---

**Your name & ID:** _____

- This is a closed book exam
- You do not need nor are permitted to use calculators
- The value of each question – number of points awarded for full credit – is shown in parenthesis
- The problems are not necessarily in any order of difficulty. We recommend that you read through all the problems first, then do the problems in whatever order suits you best.
- Record all your answers in the places provided

| Problem 1.1 | Problem 2.1 | Problem 3.1 | Problem 3.2 | Problem 4.1 | Total |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |

**(1.1)** The passive-aggressive algorithm is an on-line algorithm that updates its parameters in response to each training example and label. If $x \in \mathcal{R}^2$ is the example, $y$ the corresponding label, and $\theta^{(k)}$ the parameter vector prior to seeing $(x, y)$, then the algorithm finds $\theta^{(k+1)}$ by minimizing

$$\frac{1}{2}\|\theta - \theta^{(k)}\|^2 \text{ subject to } \text{Loss}(y\,\theta \cdot x) = 0 \tag{1}$$

with respect to $\theta$. Here $\text{Loss}(z) = \max\{0, 1 - z\}$ is the Hinge loss. The loss is zero when $z \geq 1$. We have seen in lecture that the algorithm updates the parameters similarly to the perceptron algorithm. In other words,

$$\theta^{(k+1)} = \theta^{(k)} + \eta_k\, yx \tag{2}$$

where $\eta_k = 1$ for the perceptron algorithm but this is not typically the case for the passive aggressive algorithm. Note that $x$ is a two dimensional vector for us here.

(a) **(3 points)** If $\theta^{(0)} = [0, 0]^T$, what is the value of $\theta^{(1)}$ in response to the first labeled training example $(x, y)$? Write down an expression for $\theta^{(1)}$ as a function of $x$ and $y$.

Since $\theta^{(0)} = [0, 0]^T$, $\theta^{(1)} = \eta_0 yx$. We set $\eta_0$ such that $y(\theta^{(1)} \cdot x) = 1$ or $y(\eta_0 yx \cdot x) = 1$. $y^2 = 1$ so we get $\eta_0 = 1/\|x\|^2$.

(b) **(3 points)** The update changes based on where we start. Suppose $\theta^{(0)} = [-1, 0]^T$ and $x = [1, 1]^T$, $y = -1$. What is the numerical value of $\eta_0$ in the update if we receive this particular $(x, y)$ as the first example?

In this case $y\theta^{(0)} \cdot x = 1$ so the loss is already zero. As a result, $\eta_0 = 0$.

(c) **(6 points)** Derive the value of $\eta_k$ for the passive aggressive algorithm. Assume that we start with $\theta^{(k)}$ and the example in question is $(x, y)$. Note that the loss is zero if $y\theta \cdot x \geq 1$. You could divide your answer into cases based on whether $\eta_k$ is zero or not. Show your derivation.

Similarly to part (b), $\eta_k = 0$ if $y\theta^{(k)} \cdot x \geq 1$. If not, we set $\eta_k$ such that $y\theta^{(k+1)} \cdot x = 1$ or

$$y(\theta^{(k)} + \eta_k yx) \cdot x = y(\theta^{(k)} \cdot x) + \eta_k \|x\|^2 = 1$$

which gives us

$$\eta_k = \frac{1 - y(\theta^{(k)} \cdot x)}{\|x\|^2} = \frac{\text{Loss}(y\,\theta^{(k)} \cdot x)}{\|x\|^2}$$

This answer also works when $y\theta^{(k)} \cdot x \geq 1$ giving $\eta_k = 0$.

**(2.1)** We have discussed the k-means clustering algorithm. Here you will simulate how the algorithm functions. Figure 1 shows an initialization of three means for the k-means algorithm (i.e., $k = 3$).

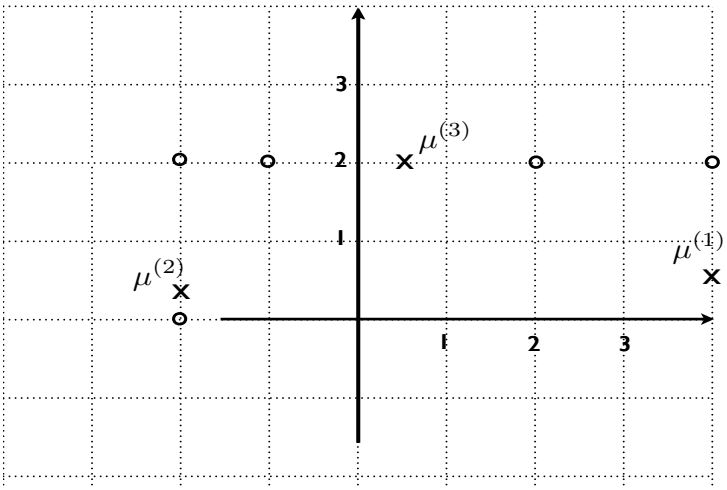



Figure 1: K-means clustering example. Each 'o' refers to a data point and 'x' corresponds to a centroid location.

(a) **(4 points)** What are the numerical values of the means $\mu^{(1)}$, $\mu^{(2)}$, $\mu^{(3)}$ after one iteration of the k-means algorithm (one update of the means) using the initialization provided in the Figure?

$$\mu^{(1)} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \quad \mu^{(2)} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \quad \mu^{(3)} = \begin{bmatrix} 0.5 \\ 2 \end{bmatrix}$$

(b) **(3 points)** How many k-means iterations (updates to the cluster means) will be required until the means reach their final values? Assume that we start with the initialization as in the figure. Please select one.

( ) 1 iteration

( x ) 2 iterations

( ) More than 2 iterations

(c) **(3 points)** We can modify the k-means algorithm slightly by biasing the selection of the cluster means. In other words, we can regularize how the cluster means are set. In particular, consider a cluster $C$ (a set of indices of training points). We will obtain the new mean (representative) for this cluster by minimizing

$$\left(\sum_{i \in C} \|x^{(i)} - \mu\|^2\right) + \|\mu\|^2 \tag{3}$$

with respect to vector $\mu$. What is the resulting $\mu$?

$$\frac{d}{d\mu}\left[\left(\sum_{i \in C} \|x^{(i)} - \mu\|^2\right) + \|\mu\|^2\right] = \sum_{i \in C}(-2)(x^{(i)} - \mu) + 2\mu = 0$$

$$\Rightarrow \quad \mu = \frac{\sum_{i \in C} x^{(i)}}{|C| + 1}$$

(d) **(3 points)** We can understand the new setting of the cluster mean in part (c) as a standard k-means update with additional point(s) in the cluster $C$. What is/are these point(s) that we should add to the cluster $C$ so as to obtain the same value for the mean vector $\mu$?

If we add point $x^{(0)} = [0, 0]^T$ to the cluster and perform the standard k-means update (average of points), we get

$$\mu = \frac{x^{(0)} + \sum_{i \in C} x^{(i)}}{|C| + 1} = \frac{\sum_{i \in C} x^{(i)}}{|C| + 1}$$

**(3.1)** We can obtain a non-linear classifier by mapping each example $x$ to a non-linear feature vector $\phi(x)$. Consider a simple classification problem in two dimensions shown in Figure 2. The points $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$ are labeled $y^{(1)} = 1$, $y^{(2)} = 1$, and $y^{(3)} = -1$.

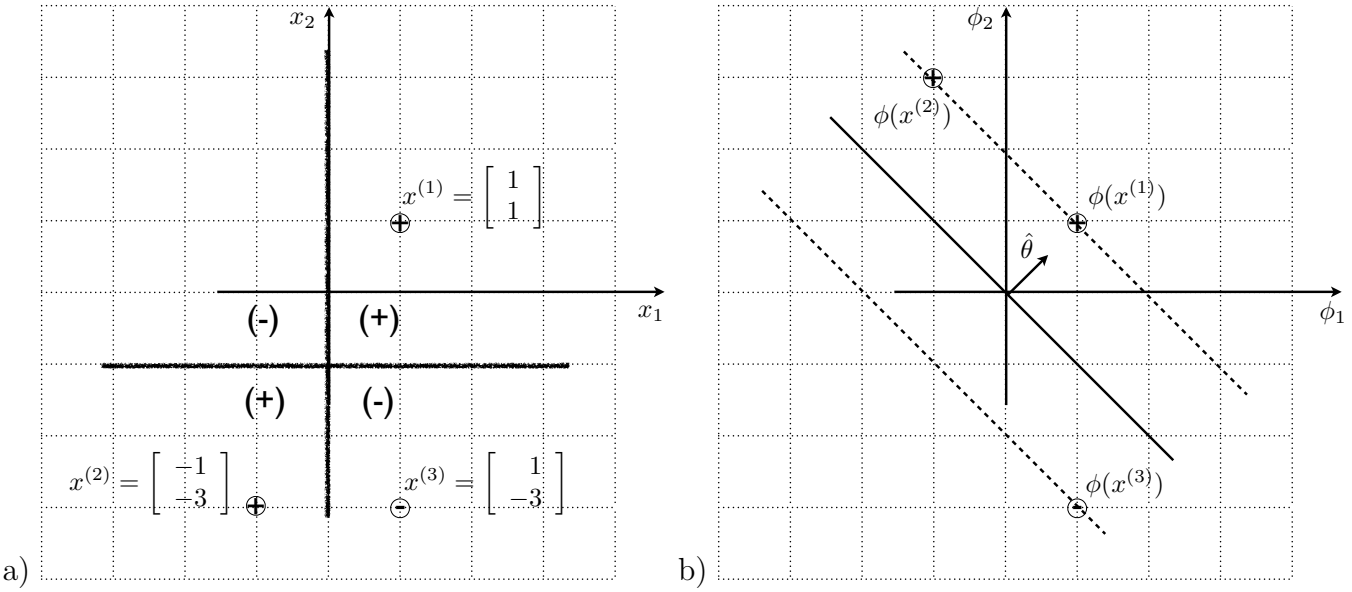



Figure 2: a) A labeled training set with three points. The labels are shown inside the circles. b) Examples in feature coordinates (to be mapped).

(a) **(2 points)** Are the points in Figure 2 linearly separable (Y/N) **( Y )**

(b) **(2 points)** Are the points in Figure 2 linearly separable through origin (Y/N) **( N )**

(c) **(2 points)** Consider a two dimensional feature mapping $\phi(x) = [x_1, x_2 x_1]^T$ where $x_1$ and $x_2$ are the coordinates of $x$. Map the three training examples in Figure 2a) to their feature coordinates in Figure 2b).

(d) We will use the support vector machine to solve the classification problem in the feature space. In other words, we will find $\theta = [\theta_1, \theta_2]^T$ that minimizes

$$\frac{1}{2}\|\theta\|^2 \text{ subject to } y^{(i)}\theta \cdot \phi(x^{(i)}) \geq 1, \quad i = 1, 2, 3 \tag{4}$$

We denote the solution by $\hat{\theta}$.

(i) **(6 points)** Draw the resulting $\hat{\theta}$ (orientation is fine), the corresponding decision boundary, and the margin boundaries in the feature coordinates in Figure 2b). Circle the support vectors.

(ii) **(2 points)** What is the value of the resulting margin? **( $\sqrt{2}$ )**

(iii) **(3 points)** What is the value of $\|\hat{\theta}\|$? **( $1/\sqrt{2}$ )**

(e) **(4 points)** Draw the resulting decision boundary $\{x : \hat{\theta} \cdot \phi(x) = 0\}$ in the original x-coordinates in Figure 2a). The boundary divides the space in two or more areas. Mark each area based on how the points there would be classified. Show any calculations below.

$\hat{\theta} = \frac{1}{2}[1, 1]^T$ so that

$$\hat{\theta} \cdot \phi(x) = \frac{1}{2}(x_1 + x_1 x_2) = 0$$

The decision boundary is therefore all $x$ such that either $x_1 = 0$ or $x_2 = -1$.

**(3.2)** Perhaps we can solve the classification problem in Figure 2a) a bit easier using kernel perceptron. Recall that in this case we would predict the label for each point $x$ according to

$$\sum_{j=1}^{3} \alpha_j y^{(j)} K(x^{(j)}, x) \tag{5}$$

The algorithm requires that we specify a kernel function $K(x, x')$. But, for training, it suffices to just have the kernel function evaluated on the training examples, i.e., have the 3x3 matrix $K_{ij} = K(x^{(i)}, x^{(j)})$, $i, j = 1, 2, 3$, known as the Gram matrix. The matrix $K$ in our case was

$$K = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \tag{6}$$

(a) **(6 points)** Based on what you know about kernels, which of the following matrices could **not** be Gram matrices. Check all that apply.

$$(\quad) \begin{bmatrix} 2 & -1 & 0 \\ -1 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{ x }) \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}, \quad (\text{ x }) \begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix} \tag{7}$$

(b) **(4 points)** Now, let's apply the kernel perceptron using the kernel $K$ given in Eq (6). We cycle through the three training points in the order $i = 1, 2, 3$. What is the **the 2nd** misclassified point? **( 3 )**

**(4.1)** We have discussed another way to build non-linear classifiers – ensembles. An ensemble of decision stumps is given by

$$h_m(x) = \sum_{j=1}^{m} \alpha_j h(x; \theta_j) \tag{8}$$

where $\alpha_j \geq 0$ specifies the (real valued) "votes" assigned to the corresponding decision stump $h(x; \theta_j)$. The stump parameters $\theta_j$ specify the coordinate to rely on, where to cut the coordinate (threshold), and which side of the cut is positive or negative. In other words, it is a linear classifier where the decision boundary is constrained to be axis-parallel.

The ensembles are quite powerful if we can include many stumps. A small number of stumps may already suffice to classify the examples correctly.

(a) **(6 points)** Solve the classification problem in Figure 3 with an ensemble. Assume a unit vote per stump. For each stump, draw its decision boundary and mark the positive/negative side.

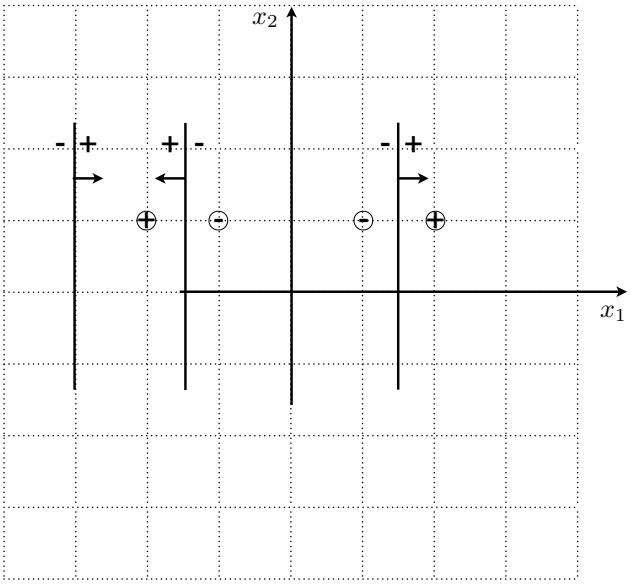
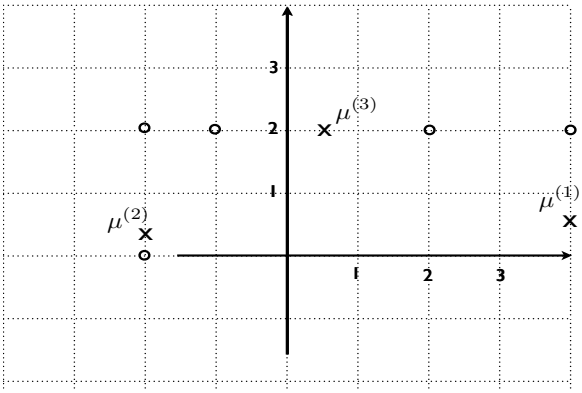
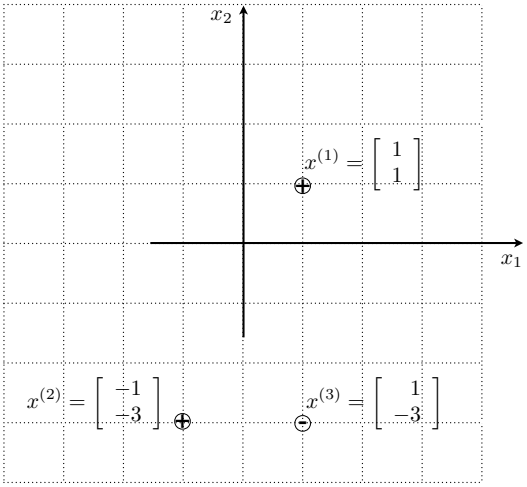



Figure 3: The training examples for the ensemble question.

## Additional set of figures for reference




a)




b)