Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.036 Introduction to Machine Learning

Midterm exam (March 19, 2015)

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	Problem 2	Problem 3	Problem 4	Problem 5	Total

Problem 1 The simple perceptron algorithm for estimating a linear classifier cycles through training examples $(x^{(i)}, y^{(i)})$, $i = 1, \ldots, n$, and performs an update of the parameters in response to each mistake. If we omit the offset parameter, then $\theta \leftarrow \theta + y^{(i)}x^{(i)}$ whenever $(x^{(i)}, y^{(i)})$ is misclassified with the current setting of the parameters.

(1.1) **(4 points) (T/F)** Please mark the statements as **T** or **F**.

( ) Suppose we normalize each training example such that $\|x^{(i)}\| = 1$. If there exists any θ^* such that $y^{(i)}(\theta^* \cdot x^{(i)}) \geq 1$, $i = 1, \ldots, n$, then the perceptron algorithm converges after only a single mistake.

( ) Whenever the perceptron algorithm converges, the parameters of the averaged perceptron algorithm also linearly separate the training examples.

We can turn the linear perceptron into a non-linear classifier by mapping each example to a feature vector $\phi(x)$. We can also rewrite the algorithm such that it uses only inner products between examples (feature vectors). In other words, only values of the kernel function $K(x, x') = \phi(x) \cdot \phi(x')$ are needed. The kernel perceptron algorithm cycles through the training examples as before, updating mistake counts α_i whenever a mistake occurs.

(1.2) **(2 points) (T/F) ( )**: The kernel perceptron without offset classifies any new example x according to the sign of $\sum_{j=1}^{n} \alpha_j y^{(j)} K(x^{(j)}, x)$ where $\sum_{j=1}^{n} \alpha_j \leq n$.

Here we use a kernel perceptron to classify nodes in an undirected graph. You can think of the graph as a social network representing friendship relations. Each person is a node in the graph and there's an edge between two nodes whenever people are friends. Our goal is to learn to predict a positive/negative label for each node. We were thinking of using a particular type of kernel function based on how many neighbors any two nodes have in common. In other words, if $N(i)$ is the set of neighbors of node i, then

$$K(i, j) = |N(i) \cap N(j)| \quad (\# \text{ of common neighbors}) \tag{1}$$

(1.3) **(4 points)** Which of the following properties of $K(i, j)$ are *also* properties of any valid kernel function over the nodes. Check all that apply.

( ) $K(i, j) = K(j, i)$ for all i, j

( ) $K(i, j) \geq 0$ for all i, j

( ) $K(i,i) \geq 0$ for all i

( ) $K(i,j) \leq K(i,i)$ for all i,j

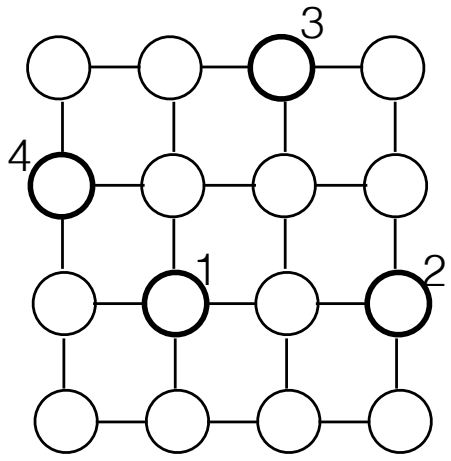
(1.4) **(4 points)** Write down the feature representation $\phi(i)$ corresponding to our chosen kernel $K(i,j) = \phi(i) \cdot \phi(j)$. Specifically, define the k^{th} coordinate of $\phi(i)$ as

$$\phi(i)_k = \begin{cases} 1, & \text{if } \underline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

(1.5) **(6 points)** Consider the following simple grid graph with four nodes numbered and highlighted. These are the training nodes (training examples) we have labels for. Once labels are made explicit, we can run the kernel perceptron algorithm with our kernel on these nodes, in the order given, i.e., cycling through 1,2,3, and 4.

(a) Is it possible to label the highlighted nodes such that our kernel perceptron algorithm would make only 1 mistake in total. Please answer **Y** or **N** ( )

(b) Please label each of the highlighted nodes in the graph with a "+" or "-" such that the kernel perceptron algorithm, if run with these labels, would make a mistake only on nodes 1 and 3, converging after the first round.



Problem 2 The passive-aggressive algorithm differs from the perceptron algorithm in that it updates its parameters in response to each training example (x, y) by minimizing

$$\frac{\lambda}{2}\|\theta - \theta^{(k)}\|^2 + \text{Loss}(y\,\theta \cdot x) \tag{3}$$

with respect to θ where $\theta^{(k)}$ is the current setting of the parameters. Here $\lambda > 0$ is a parameter we have to set and Loss(z) defines how we measure errors. We will adopt a squared Hinge loss throughout this problem:

$$\text{Loss}(z) = \begin{cases} (1-z)^2/2, & \text{if } z \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

This loss, like Hinge loss, is zero when $z \geq 1$ but increases as $(1-z)^2/2$ when z drops below 1. We have seen in lecture that passive-aggressive updates have the form

$$\theta^{(k+1)} = \theta^{(k)} + \eta_k \, y \, x \tag{5}$$

where η_k depends on λ, the loss function, as well as the example.

(2.1) **(2 points)** Suppose we have parameters $\theta^{(k)}$ and see a training example (x, y). If $y\theta^{(k)} \cdot x < 1$ before the update, is it possible that $y\theta^{(k+1)} \cdot x > 1$ after the update? Please answer **Y** or **N** **( )**

(2.2) **(2 points)** With our chosen loss function, if $y\theta^{(k)} \cdot x < 1$ before the update, we can always just solve for $\theta^{(k+1)}$ by minimizing

$$\frac{\lambda}{2}\|\theta - \theta^{(k)}\|^2 + (1 - y\theta \cdot x)^2/2 \tag{6}$$

Please answer **Y** or **N** **( )**

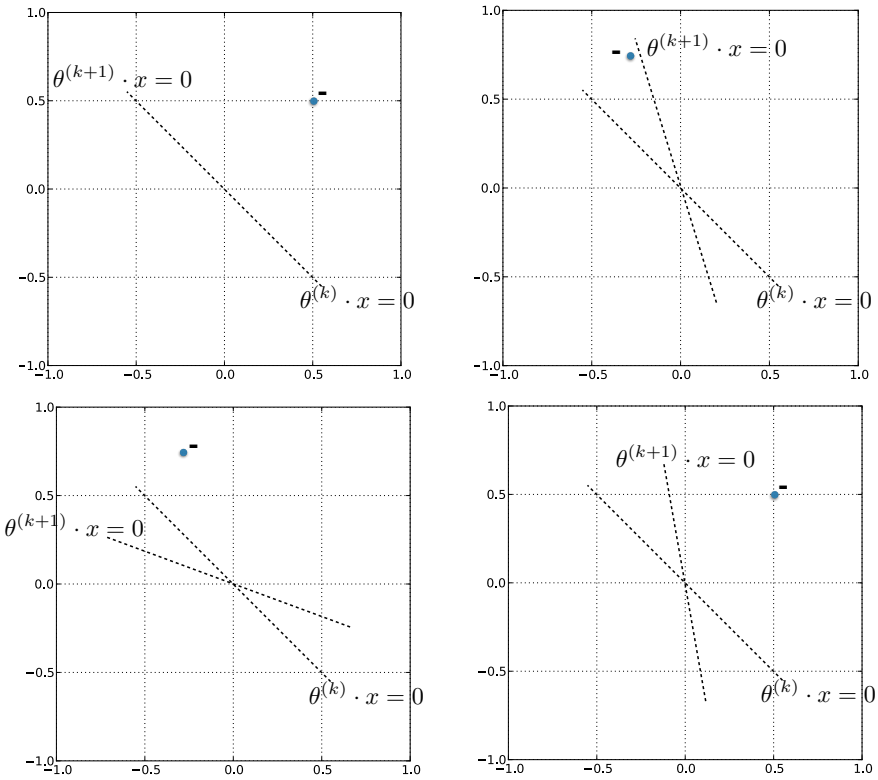
(2.3) **(4 points)** If $\theta^{(0)} = [0, 0]^T$, what is the value of $\theta^{(1)}$ in response to the first labeled training example (x, y)? Select the right expression for η_0 as a function of x, y, and (positive) λ.

( ) $$\eta_0 = \min\left\{ \frac{\text{Loss}(0)}{\|x\|^2}, \frac{1}{\lambda} \right\} \tag{7}$$

( ) $$\eta_0 = \min\left\{ \frac{1}{\|x\|^2}, \frac{1}{\lambda} \right\} \tag{8}$$

( ) $$\eta_0 = \frac{1}{\lambda + \|x\|^2} \tag{9}$$

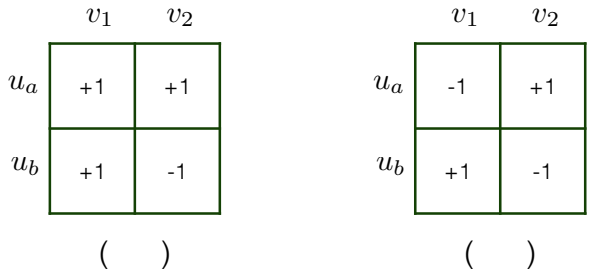
(2.4) **(6 points)** The passive-aggressive algorithm will result in slightly different updates depending on how we set λ. Below you will see four plots representing decision boundaries before/after the update obtained with some value of λ in response to a negatively labeled point shown in the figure. Some of the plots cannot arise in this way, however. Please a) draw the orientation of $\theta^{(k)}$ (parameters before the update) in each figure and b) cross out all the plots which are **NOT** possible with any value $\lambda > 0$.



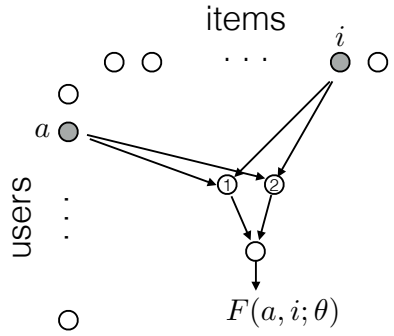
Problem 3 Recommender problems are everywhere, from Netflix, Amazon, to Google News. Here we aim to reconstruct a matrix of user-item preferences using two different methods learned in class, matrix factorization and neural networks. Yes, neural networks find their way here as well...

Suppose we have n users $a \in \{1, \ldots, n\}$ and m items $i \in \{1, \ldots, n\}$. Since each user is likely to provide ratings for only a small subset of possible items, we must heavily constrain the models so as not to overfit. In fact, our goal here is to understand how constrained they really are.

(3.1) **(4 points)** We begin by estimating a simple rank-1 model $X_{ai} = u_a v_i$ where $u_1, \ldots, u_n$ and $v_1, \ldots, v_m$ are just scalar parameters associated with users u_a and items v_i, respectively. Please select which (if any) of the following 2x2 matrices **cannot** be reproduced by the rank-1 model.



We will also try to understand a simple neural network model applied to the same problem. To this end, we introduce an input unit corresponding to each user and each item. When querying about a selected entry, (a, i), only the a^{th} user input unit and i^{th} item input unit are active (set to 1), the rest are equal to zero. Thus only the outgoing weights from these two units matter for predicting the value for (a, i). Figure below provides a schematic representation of the model.



User a has two outgoing weights, U_{a1} and U_{a2}, and item i has two outgoing weights, V_{i1} and V_{i2}. These weights are fed as inputs to the two hidden units in the model. The hidden units evaluate

$$z_1 = U_{a1} + V_{i1}, \quad f(z_1) = \max\{0, z_1\} \tag{10}$$
$$z_2 = U_{a2} + V_{i2}, \quad f(z_2) = \max\{0, z_2\} \tag{11}$$

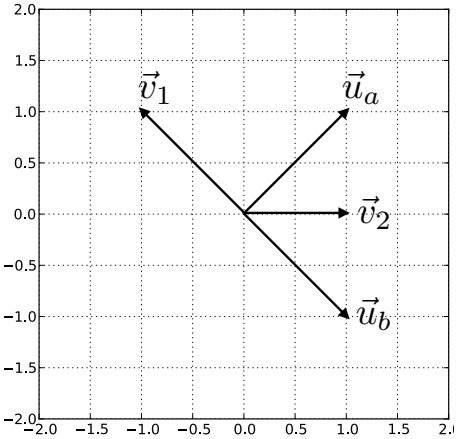
Thus, for (a, i) entry, our network outputs

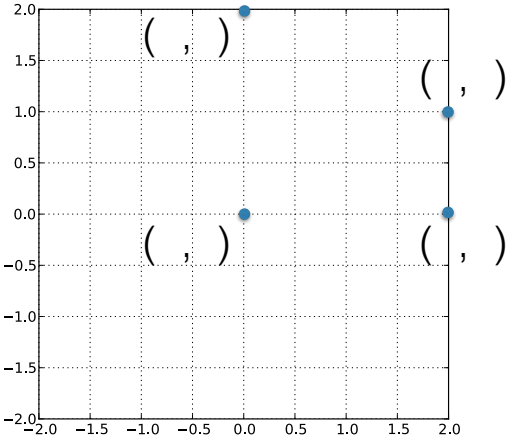
$$F(a, i; \theta) = W_1 f(z_1) + W_2 f(z_2) + W_0 \tag{12}$$

In a vector form, each user a has a two-dimensional vector of outgoing weights $\vec{u}_a = [U_{a1}, U_{a2}]^T$ and each item i has $\vec{v}_i = [V_{i1}, V_{i2}]^T$. The input received by the hidden units is then $\vec{z} = [z_1, z_2]^T = \vec{u}_a + \vec{v}_i$.

Consider again a simple matrix of ratings where we have two users, $\{a, b\}$, and two items $\{1, 2\}$. We will fix the first layer weights as shown in the Figure below.

(3.2) **(4 points)** Each of the four user-item pairs in the 2x2 matrix are mapped to a corresponding feature representation $[f(z_1), f(z_2)]^T$ (hidden unit activations). Please mark the points on the right with the correct pair, e.g., (a,1), that it corresponds to.





(3.3) **(4 points)** If we keep the input to hidden layer weights fixed as shown above, and only estimate the weights in the output layer, which (if any) of the following matrices the neural network **cannot** reproduce.

	v_1	v_2
u_a	+1	+1
u_b	+1	-1

( )

	v_1	v_2
u_a	-1	+1
u_b	+1	-1

( )

Problem 4 In this question, we will consider a kernel-based K-means clustering algorithm. The distance measure for this algorithm is defined exactly as before but now evaluated in terms of the feature vectors, i.e., $\|\phi(x^{(i)}) - \phi(x^{(j)})\|^2$.

(4.1) **(2 points)** Suppose we have a cluster of points $\{x^{(i)}, i \in C\}$. Provide an expression for the corresponding centroid z in feature coordinates.

(4.2) **(6 points)** Show that when we reassign points to clusters, we don't have to explicitly compute the means $z^{(j)}$ or feature vectors $\phi(x^{(i)})$, because the cost of assigning a point $x^{(i)}$ to a cluster $z^{(j)}$, i.e., $||\phi(x^{(i)}) - z^{(j)}||^2$, can be expressed in terms of the kernel function $K(x, x') = \phi(x) \cdot \phi(x')$.

(4.3) **(4 points)** Suppose we are given the kernel matrix $K_{ij} = K(x^{(i)}, x^{(j)})$ evaluated between any pair of training examples. What is the number of operations that a single iteration of kernel k-means takes as a function of n number of points and k number of clusters.

Problem 5

(5.1) **(4 points)** Consider weighted training points shown in Figure 1. The figure also shows two possible decision stumps to consider at this round of boosting: A and B. Which one of the stumps will be selected? Briefly justify your answer.

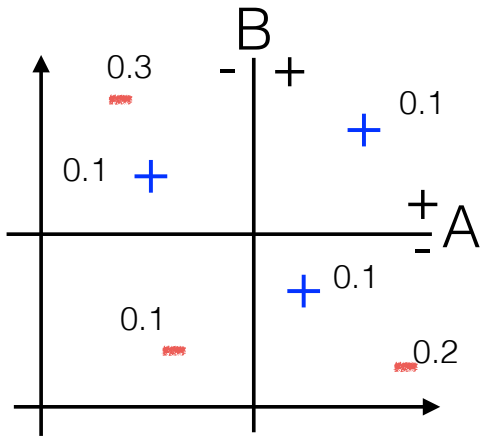


Figure 1: Training set and two candidate decision stumps

(5.2) **(6 points)** Figure 2 shows a set of four labeled points. Construct an ensemble of decision stumps that correctly classifies these points. Use the smallest number of decision stumps. Clearly indicate the decision boundary and the positive/negative direction of the stumps, including their votes (if not uniform).

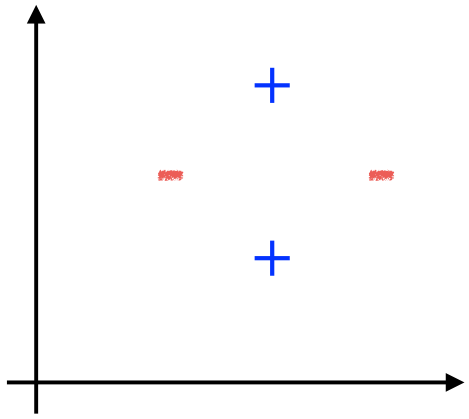


Figure 2: Training set of four points

(5.3) **(6 points)** Consider training set in Figure 3. Assume that the points are equally weighted, i.e., each have weight 1/4.

(a) What is the weighted error of the stump shown in the Figure? What is the corresponding votes α that this stump receives?

(b) Will AdaBoost succeed in finding an ensemble that correctly classifies these points? Briefly justify your answer.

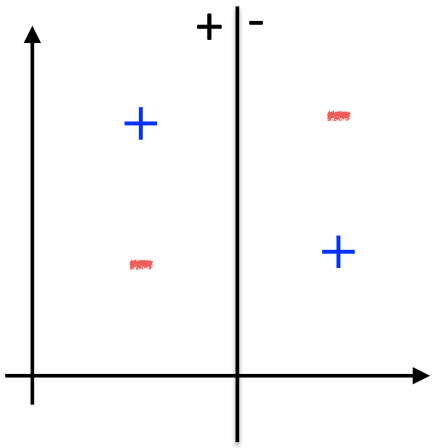


Figure 3: Training set for Question 5.3