Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.036 Introduction to Machine Learning

Final exam (May 19, 2015)

Your name & ID: Staff solutions

- 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

Prob 1	Prob 2	Prob 3	Prob 4	Prob 5	Prob 6	Total
23	16	18	16	15	6	**94**

Problem 1

(1.1) **(5 points)** VC-dimension is a measure of complexity of a set of classifiers such as the set of linear classifiers. Let $\mathcal{H}$ denote the set of classifiers in question and $d_{VC}(\mathcal{H})$ the corresponding VC-dimension. Select all the true statements.

- (✓) If $\mathcal{H}$ has only one classifier, then $d_{VC}(\mathcal{H}) = 0$
- (✓) If $\mathcal{H} = \{h_1, \ldots, h_K\}$ (finite), then necessarily $d_{VC}(\mathcal{H}) \leq \log_2 K$
- ( ) If $d_{VC}(\mathcal{H}) > n$, then any n labeled points can be correctly classified by some $h \in \mathcal{H}$.
- ( ) No set of $n > d_{VC}(\mathcal{H})$ labeled points can be correctly classified by $h \in \mathcal{H}$.
- (✓) There exists $d_{VC}(\mathcal{H})$ points that can be always correctly classified by some $h \in \mathcal{H}$

(1.2) **(4 points)** Consider a simple set of classifiers in two dimensions where, for each classifier, the positive set is always between two vertical lines or between two horizontal lines. In other words, any $h \in \mathcal{H}$, parameterized by $\theta = \{a, b, i\}$, can be written as

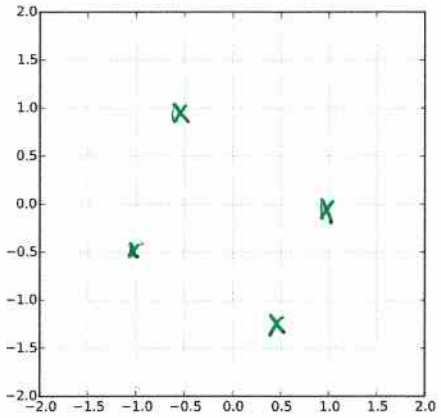
$$h_\theta(x) = \begin{cases} +1 \text{ if } a \leq x_i \leq b, \\ -1 \text{ otherwise} \end{cases}$$

Please write $h_\theta(x)$ where $\theta = \{a, b, i\}$ as an ensemble of decision stumps.

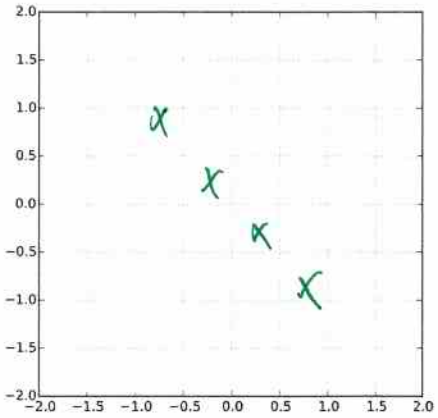
$h_\theta(x) = \text{sign}\left[\text{sign}(x_i - a) + \text{sign}(-x_i + b) + \underbrace{\text{sign}(x_i - \infty)}_{\text{always } -1} \right]$

also $\text{sign}(0) = 1$

(1.3) **(4 points)** In the figure below, draw a set of 4 points that can be shattered (classified in all possible ways) by this set of classifiers (left) and a set of 4 points that cannot (right).



can be shattered



cannot be shattered

(1.4) **(4 points)** So, we got n labeled training examples in $\mathcal{R}^2$ and found the best classifier in the above set $\mathcal{H}$. The resulting training error was zero. We also fit a linear classifier to the same training set and also got zero training error. Which classifier should we prefer? Briefly explain why.

VC-dim of the set of linear clsmf. in 2-d is 3 which is less. We would prefer linear

(1.5) **(3 points)** In practice, we typically resort to cross-validation as a proxy to test error in order to select among different sets of classifiers. Leave-one-out cross-validation takes out each example in turn as a held-out test example, and averages the error on these held-out examples when the classifier is trained on the remaining points. Our training set is given in the figure (CV) below. Suppose we use a kernel perceptron algorithm with a radial basis kernel for this problem. What is the resulting leave-one-out cross-validation error? (1.0)

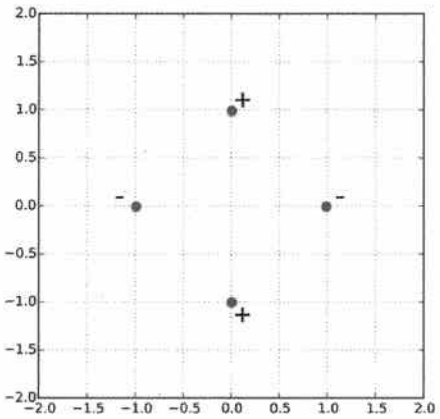
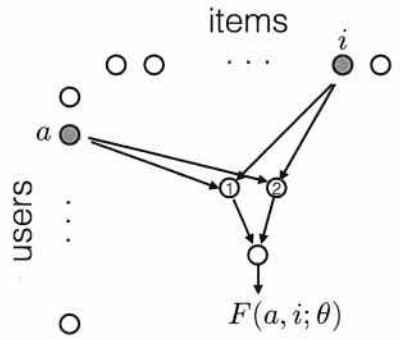


Figure (CV): labeled training set

(1.6) **(3 points)** Specify a feature vector $\phi(x)$ for a linear classifier that would attain the lowest leave-one-out cross-validation error on the training points in figure (CV).

$\phi(x) =$ x_1^2

Problem 2 Suppose we have a recommender problem with n users $a \in \{1, \ldots, n\}$ and m items $i \in \{1, \ldots, m\}$. For simplicity, we will treat the target rating values as class labels, i.e., using $\{-1, 1\}$ ratings (dislikes,likes). Each user is likely to provide feedback for only a small subset of possible items and thus we must constrain the models so as not to overfit. Our goal here is to understand how a simple neural network model applies to this problem, and what its constraints are. To this end, we introduce an input unit corresponding to each user and each item. In other words, there are $n + m$ input units. 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 and will not affect the predictions. Put another way, only the outgoing weights from these two units matter for predicting the value (class label) for entry (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

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

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

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

where θ denotes all the weights U, V, and W. 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, if represented as a vector, is then $\vec{z} = [z_1, z_2]^T = \vec{u}_a + \vec{v}_i$.

Consider a simple version of the problem where we have only two users, $\{a, b\}$, and two items $\{1, 2\}$. So the recommendation problem can be represented as a 2x2 matrix. We will initialize the first layer weights as shown in Figure (NN) below.

(2.1) **(4 points)** Using the initial input-to-hidden layer weights, 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.

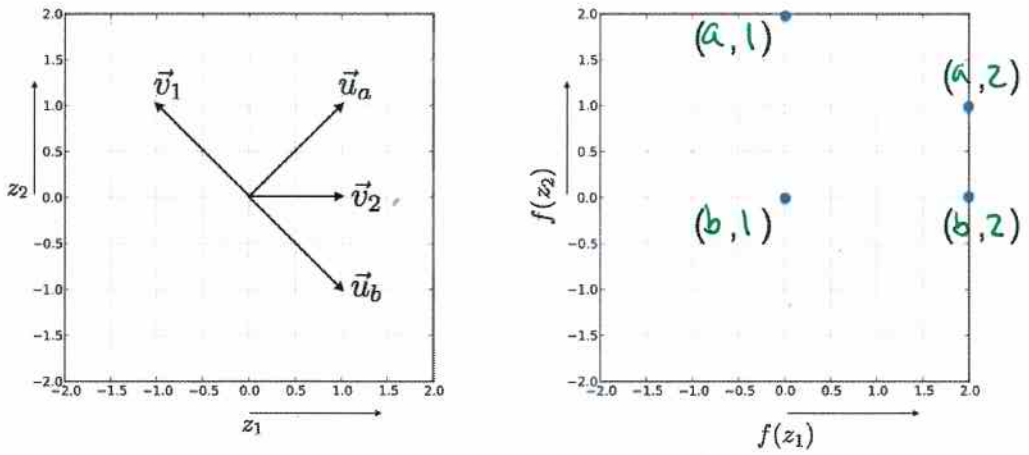


Figure (NN): Outgoing weight vectors from user/item input units (left); hidden layer activations (right)

(2.2) **(4 points)** Suppose we keep the input to the hidden layer weights (U's and V's) at their initial values shown in Figure (NN), and only estimate the weights W corresponding to the output layer. Different choices of the output layer weights will result in different predicted 2x2 matrices of $\{-1, 1\}$ labels. Which (if any) of the following matrices the neural network **cannot** reproduce with any choice of the output layer weights W_1, W_2, and W_0?

	1	2
a	+1	+1
b	+1	-1

( )

	1	2
a	-1	+1
b	+1	-1

(✓)

Learning a new representation for examples (hidden layer activations) is always harder than learning the linear classifier operating on that representation. In neural networks, the representation is learned together with the end classifier using stochastic gradient descent. We initialize the output layer weights as $W_1 = W_2 = 1$ and $W_0 = -1$.

(2.3) **(2 points)** Assume that all the weights are initialized as provided above. What is the class label (+1/-1) that the network would predict in response to $(b, 2)$ (user b, item 2)? (+ |)

(2.4) **(6 points)** Assume that we observe the opposite label from your answer to the previous question. In other words, there is a training signal at the network output. All the weights are initialized as before. Please mark (check the boxes) of all the weights in Figure (SGD) that would change (have non-zero update) based on a single stochastic gradient descent step in response to $(b, 2)$ with our specific weight initialization and the target label. Note that the input units a, b and $1, 2$ are activated with 0's and 1's as shown inside the circles. We are not asking about whether W_0 would change.

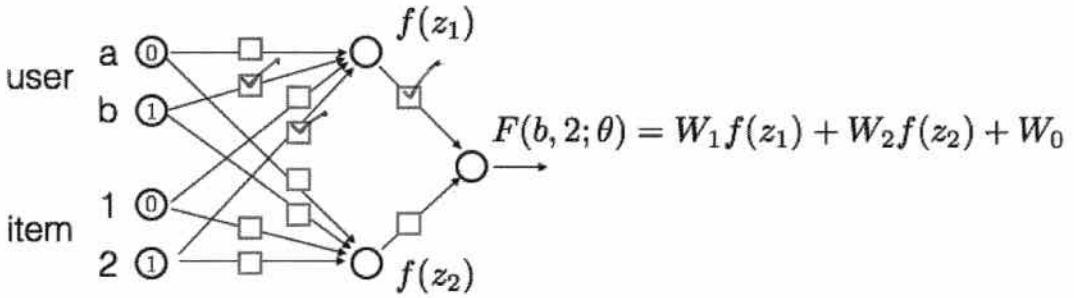


Figure (SGD): Neural network for stochastic gradient descent.

Problem 3 Consider initializing a two component Gaussian mixture model as shown below in the figure. The variances of the two Gaussians are equal $\sigma_1^2 = \sigma_2^2 = 0.5^2$ and they have the same prior probabilities (mixing proportions) $p_1 = p_2 = 0.5$. The two means $\mu^{(1)}$ and $\mu^{(2)}$ are exactly at the grid points shown in the figure (EM).

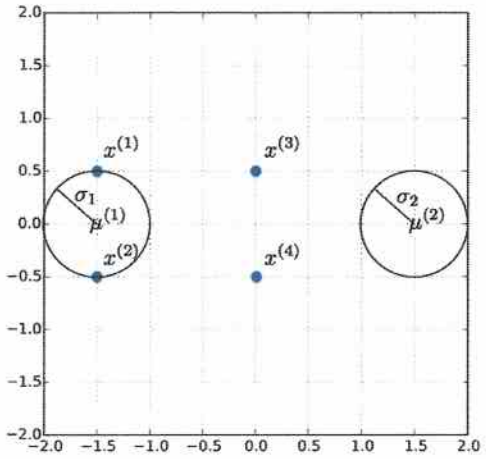


Figure (EM): Initial two component Gaussian mixture model.

(3.1) **(4 points)** Please select the right intervals for the soft assignments of points to mixture components in the first E-step of the EM-algorithm. Here $p(j|i)$ is the posterior probability that component j is assigned to point i. Each point should be marked to exactly one interval

	$p(1\|i) < 0.5$	$p(1\|i) = 0.5$	$p(1\|i) > 0.5$
$x^{(1)}$			✓
$x^{(2)}$			✓
$x^{(3)}$		✓	
$x^{(4)}$		✓	

(3.2) **(3 points)** Let $\hat{\mu}^{(2)}$ be the mean for the 2nd component after the first M-step. Which of the following holds for the horizontal component $\hat{\mu}_1^{(2)}$ of this mean?

(✓) $\hat{\mu}_1^{(2)} < 0$, ( ) $\hat{\mu}_1^{(2)} = 0$, ( ) $\hat{\mu}_1^{(2)} > 0$

(3.3) **(3 points)** Which of the following holds for the variances $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ after the first M-step?

(✓) $\hat{\sigma}_1^2 > 0.5^2$, ( ) $\hat{\sigma}_2^2 > 0.5^2$, (✓) $\hat{\sigma}_1^2 > \hat{\sigma}_2^2$

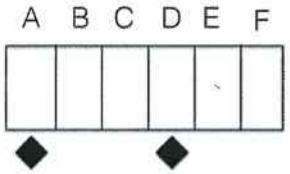
(3.4) **(4 points)** Is $\hat{p}_1 > 2\hat{p}_2$ after the first M-step? Please answer **Y/N ( )**. For ballpark estimation, $\exp(-1) \approx 0.37, \exp(-3) \approx 0.05, \exp(-6) \approx 0.0025$.

(3.5) **(4 points)** If we continue to iterate the E- and M- steps, what are the values of variances and the mixing proportions that the algorithm converges to with the above initialization?

$\sigma^{*2}_1 =$ $\approx \frac{0.5^2}{2}$, $\sigma^{*2}_2 =$ $\approx \frac{0.5^2}{2}$

$p^*_1 =$ 0.5, $p^*_2 =$ 0.5

Problem 4 We will consider here robot localization from measurements. We assume that the robot operates in a simple 1-dimensional grid shown below. Unfortunately, our fancy on-board localization software failed and the robot is left with only measuring and transmitting the temperature of its immediate surroundings, i.e., the temperature at the grid point it is at. It can transmit only a binary value $X =$"cold" or $X =$"hot". Two locations in the grid – A and D – are known to have volcanoes and are therefore "hot". The robot's temperature gauge is 90% accurate whether it is in "hot" or "cold" locations. For example, it would correctly transmit "hot" in A with probability 0.9.



We know that the robot starts in positions A or B on day 1 so $Y_1 = A$ or $Y_1 = B$ with equal probability. The robot always tries to move to the right (towards F) over night. If it is currently in locations A or B, it will succeed in moving to the right overnight. In all other locations, it will either succeed in moving one step to the right overnight (with probability 0.5) or else it would remain in the same location (with probability 0.5). Let the robot's position on day t be $Y_t \in \{A, B, C, D, E, F\}$. We will model the robot's position with an HMM where the state is Y_t and the corresponding observation is the robot's transmission X_t.

(4.1) **(4 points)** Specify the initial state, state transition, and the emission probabilities for this HMM.

Y_1

A	0.5
B	0.5
C	0
D	0
E	0
F	0

Y_{t-1} \ Y_t

	A	B	C	D	E	F
A		1				
B			1			
C			0.5	0.5		
D				0.5	0.5	
E					0.5	0.5
F						1

Y_t \ X_t

	"hot"	"cold"
A	0.9	0.1
B	0.1	0.9
C	0.1	0.9
D	0.9	0.1
E	0.1	0.9
F	0.1	0.9

(4.2) **(4 points)** What are the possible (non-zero probability) sequences of locations (Y_1, Y_2, Y_3) that the robot could have followed up to and including day 3 if it has transmitted $X_1 =$"hot", $X_2 =$"cold", and $X_3 =$"cold"?

ABC
BCC
BCD

(4.3) **(4 points)** Which sequence of locations (Y_1, Y_2, Y_3) is most likely to occur together with $(X_1 =$ "hot", $X_2 =$ "cold", $X_3 =$ "cold"), and what is the corresponding joint probability $P(Y_1, Y_2, Y_3, X_1, X_2, X_3)$?

ABC prob $= 0.5 \cdot 0.9 \cdot 1 \cdot 0.9 \cdot 1 \cdot 0.9$

(4.4) **(4 points)** What is $P(Y_3|X_1 =$ "hot", $X_2 =$ "cold", $X_3 =$ "cold"), i.e., the posterior probability distribution over the robot's possible locations on day 3, given the observations? Please check exactly one box on each row.

$P(Y_3|X_1 =$ "hot", $X_2 =$ "cold", $X_3 =$ "cold")

Y_3	0.0	< 0.1	≈ 0.5	> 0.9
A	✓			
B	✓			
C				✓
D		✓		
E	✓			
F	✓			

Problem 5 Consider a robot that can either stand still and CHARGE (using its solar panels) or it can scurry around and EXPLORE. The robot's state (as we measure it) represents only how charged its battery is and can be EMPTY, LOW or HIGH. The robot is very eager to explore and this is how the rewards are set. The MDP transition probabilities and rewards are specified as shown below.

s	a	s'	T(s,a,s')	R(s,a)
HIGH	EXPLORE	LOW	1.0	+2
LOW	EXPLORE	EMPTY	1.0	+2
EMPTY	EXPLORE	EMPTY	1.0	-10
HIGH	CHARGE	HIGH	1.0	0
LOW	CHARGE	HIGH	1.0	-1
EMPTY	CHARGE	HIGH	1.0	-10

Note that the reward only depends on the robot's current state and action, not the state that it transitions to.

(5.1) **(3 points)** Based on the transitions and rewards (without further calculation), what is the optimal policy for this robot if we set the discount factor $\gamma = 0$?

$\pi_0^*(HIGH)$ = EXPLORE
$\pi_0^*(LOW)$ = EXPLORE
$\pi_0^*(EMPTY)$ = EXPLORE or CHARGE

(5.2) **(4 points)** Could changing the discount factor γ change the optimal action to take in any state? (check all that apply)

( ) when s =HIGH?
(✓) when s =LOW?
(✓) when s =EMPTY? doesn't change if

(5.3) **(4 points)** Let's see how the robot values its states, and recovers those values through value iteration, when the discount factor is set to $\gamma = 0.5$. We start with all zero values as shown in the first value column. Please fill out the table

s	$V_0(s)$	$V_1(s)$	$V_2(s)$
EMPTY	0	-10	-9
LOW	0	2	0
HIGH	0	2	3

(5.4) **(4 points)** If the robot uses values $V_2(s)$ as the true (converged) values, which action would it take in state $s = LOW$? (Show your calculation)

argmax $\begin{cases} +2 + \gamma V_2(\text{EMPTY}) & a = \text{EXPLORE} \\ -1 + \gamma V_2(\text{HIGH}) & a = \text{CHARGE} \end{cases}$ ✓

Problem 6

(6.1) **(4 points)** Two of the guest lectures emphasized the role of "transfer learning" in their applied contexts, especially medicine. Briefly describe what transfer learning is.

using data from one domain/problem to help another

(6.2) **(2 points)** In using medical data for prediction, one may often have to face "small data" rather than "big data" problems. One of the guest lectures emphasized ways to deal with issues arising from applying machine learning methods in the "small data" regime. These include (select all that apply)

- (✓) Dimensionality reduction such as PCA
- (✓) Shrinkage (using estimates tied to broader categories to inform more specific ones)
- ( ) Expanding feature vectors to include more potentially useful features
- ( ) Making high dimensional feature vectors sparse

END OF EXAM