Final Exam
EECS 545: Machine Learning
Winter, 2016

Name:

UM uniqname:

- **Closed book. Five sheets of paper of notes are allowed. No computers, cell phones or calculators.**
  - Showing your work makes partial credit possible.
    If you write nothing at all, it's hard to justify any score but zero.
  - Feel free to use the back of each sheet for scratch paper, but not for credit.
    IMPORTANT: Your exam will be scanned and we will not include the back pages, so do not write anything you want for credit there.
  - Write clearly. If we can't read your writing, it will be marked wrong.
- This course operates under the rules of the College of Engineering Honor Code. Your signature endorses the pledge below. **After** you finish your exam, please sign below:
  I have neither given nor received aid on this examination, nor have I concealed any violations of the Honor Code.

DO NOT WRITE BELOW THIS LINE

Problem	1	2	3	4	5	6	7	8	Total
Points									
Max Points	10	6	4	4	4	8	6	2	44

Problem 1 (True/False). Are the following statements true or false? Please circle one. (No need for explanations unless you feel the question is ambiguous and want to justify your answer).

(a) Directed and undirected graphical models are capable of representing exactly the same independence assumptions. That is, every directed model can be converted to an equivalent undirected model and back.

True (False)

(b) Assume we have a directed graphical model with two different nodes A and B, where B is neither a child nor a parent of A. Then it is necessarily the case that A and B are conditionally independent *given* the parents and the children of A. That is, $A \perp B | (\text{parents}(A) \cup \text{children}(A))$.

True (False)

(c) Relative entropy (also called Kullback-Leibler Divergence) between distributions P and Q is symmetric. That is, $\text{KL}(P, Q) = \text{KL}(Q, P)$ for every P and Q.

True (False)

(d) The parameters computed by the Expectation-Maximization algorithm are guaranteed converge to a global maximum of the likelihood function.

True (False)

(e) Let X and Y be two random variables on a discrete space $\{1, \ldots, n\}$. If X and Y are **not** independent, then their mutual information $I(X; Y)$ will be strictly greater than 0.

(True) False

(f) Assume we have a distribution $\mathcal{D}$ whose mean is μ. We take n samples $X_1, \ldots, X_n$ from this distribution, where $n > 0$ is an even integer. We use these samples to estimate the mean, and we consider three separate estimators:

$$\hat{\mu}_1 = \frac{1}{n/2} \sum_{i=1}^{n/2} X_i \qquad \hat{\mu}_2 = \frac{1}{n} \sum_{i=1}^{n} X_i \qquad \hat{\mu}_3 = 3.14159 + \frac{1}{n} \sum_{i=1}^{n} X_i.$$

It is possible that $\hat{\mu}_1$ has lower bias than $\hat{\mu}_2$.

True (False)

(g) Following the exact same setup as in the previous problem, it is possible that $\hat{\mu}_3$ has higher variance than $\hat{\mu}_1$.

True (False)

(h) It is possible for a data set that is **not** linearly separable to become linearly separable after projection to a lower dimensional space via PCA.

True (False)

(i) For every input dataset, the final clusters output by the k-means algorithm do not depend on the initialization of k-means. (That is, the output is the same for every initial choice of cluster-means and assignments.)

True (False)

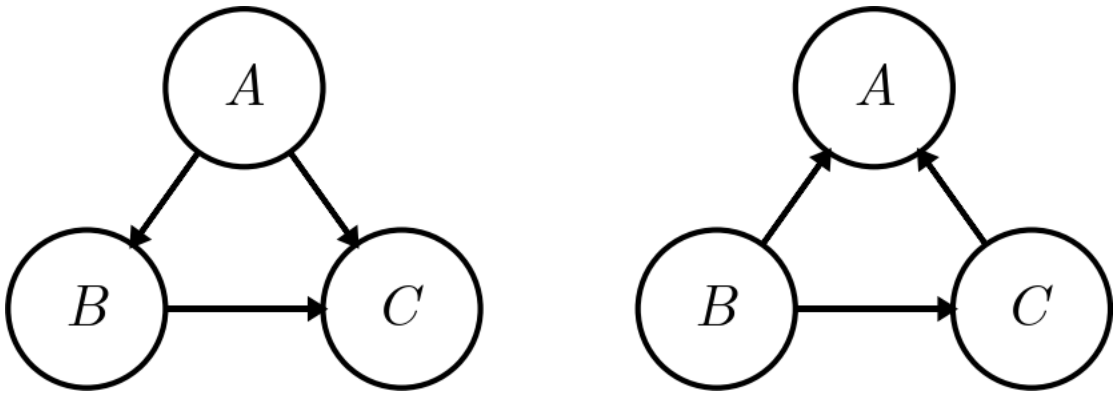
(j) In **every** hidden Markov model, we assume that the sequence of *hidden* states $Z_1, Z_2, \ldots, Z_n$ satisfies the Markov property, i.e.$P(Z_i|Z_1, Z_2, \ldots, Z_{i-1}) = P(Z_i|Z_{i-1})$ for every $i = 2, \ldots, n$. It is also true that the sequence of *observed* states $X_1, \ldots, X_n$ satisfies the Markov property. (That is, $P(X_i|X_1, X_2, \ldots, X_{i-1}) = P(X_i|X_{i-1})$ for all i)

True (False)

Problem 2 (Bayesian Networks).

(a) Consider three random variables A, B, and C. Suppose there are no conditional independencies between the variables.

(i) Draw a directed graphical model for this scenario. Two possibilities:



(ii) Write down the factorization of $P(A, B, C)$ according to your graph.

$$P(A, B, C) = P(A)P(B|A)P(C|B, A) = P(B)P(C|B)P(A|B, C)$$

(iii) Is your picture in part (i) unique? If so, explain why. If not, draw an equivalent graph.

Not unique! By the chain rule, we can factor $P(A, B, C)$ in six different ways! This gives six possibilities for the corresponding graphical model.

(b) For the graphical model shown in Figure 1, which of the following independencies hold? Please circle the **correct** independence assumptions.

(i) $\alpha \perp \eta$ – True

(ii) $\alpha \perp \beta_k \mid W_{dn}$

(iii) $\alpha \perp \beta_k \mid \eta$ – True

(iv) $\theta_d \perp \beta_k \mid W_{dn}$

(v) $\theta_d \perp \beta_k$ – True

(vi) $\theta_1 \perp \theta_2$

(vii) $\theta_1 \perp \theta_2 \mid \alpha$ – True

(viii) $Z_{dn} \perp W_{dn}$

(ix) $Z_{d1} \perp Z_{d2}$

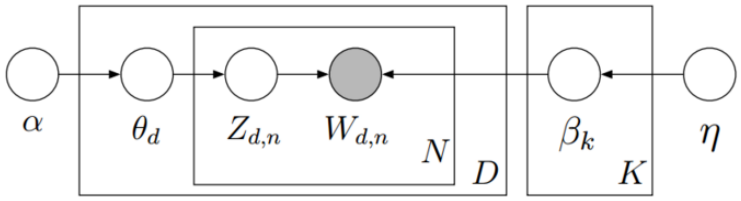


Figure 1: Latent Dirichlet Allocation

(c) Write down the factorized joint distribution for the model shown in Figure 1.

$$p(\alpha)p(\eta)\left(\prod_{k=1}^{K} p(\beta_k|\eta)\right)\left(\prod_{d=1}^{D} p(\theta_d|\alpha)\left(\prod_{n=1}^{N} p(Z_{dn}|\theta_d)p(W_{dn}|Z_{dn},\beta_1,\ldots,\beta_K)\right)\right)$$

Problem 3 (Neural Networks). Draw a neural network with specific weights for each edge that satisfies the following goal. You may choose the weights to be any real number. The activation function for a given node is the threshold of a weighted linear combination of the input values, and you can assume that the offset threshold b can also be chosen. That is, with input values $\mathbf{x}$ and weights $\mathbf{w}, b$, the output takes the form $\mathbf{1}_{\{\sum_i w_i x_i \geq b\}}$.

(a) Design a two-input neural network, where the inputs are specified by binary variables A and B, and the output of the network will be the Boolean function $A \wedge \neg B$.

Solution:

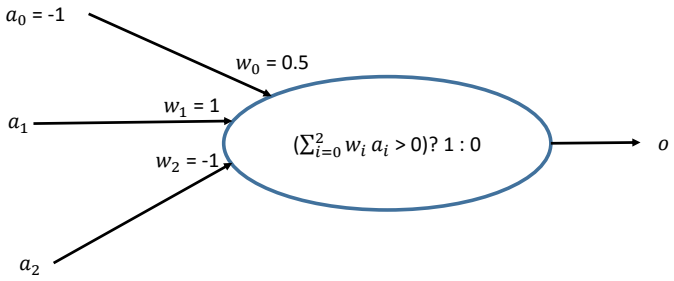


Figure 2: $A \wedge \neg B$

(b) Design a two-input neural network, where the inputs are specified by binary variables A and B, that implements the Boolean function $A \oplus B$ (where $\oplus$ represents XOR).

Solution:

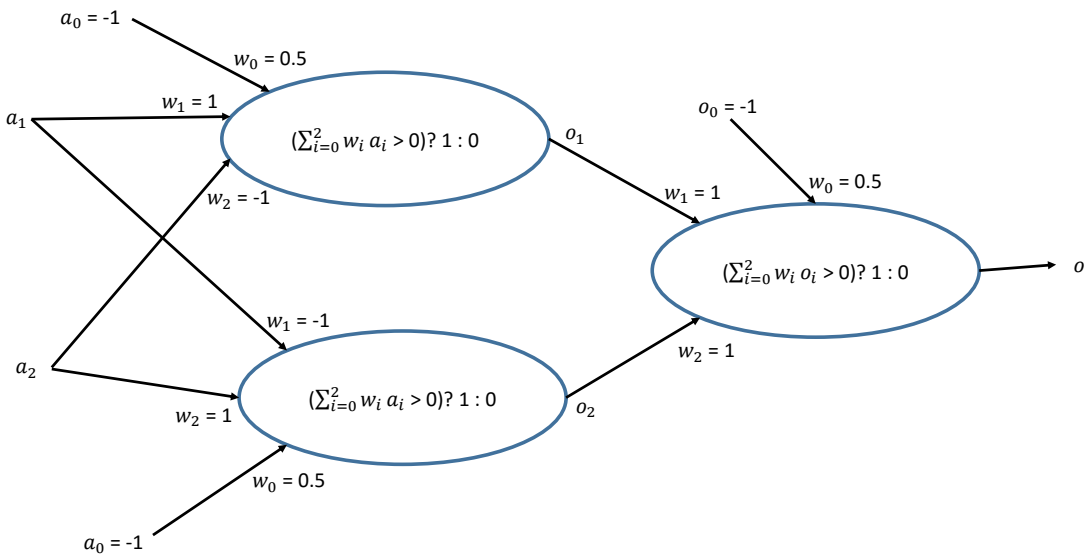


Figure 3: $A \oplus B$

Problem 4 (PCA). Indicate (roughly) the first and second principle component directions for each of the collections of points below (8 points in the left figure, and 4 points in the right figure). Draw your direction vectors starting from the origin. Make sure to **label** each eigenvector with "1st" or "2nd".

Problem 5 (Boosting). Consider the labeled training points in Figure 1, where the circles and triangles denote positive and negative labels, respectively. We wish to apply Adaboost with decision stumps to solve the classification problem. In each boosting iteration, we select the stump that minimizes the weighted training error, breaking ties arbitrarily.

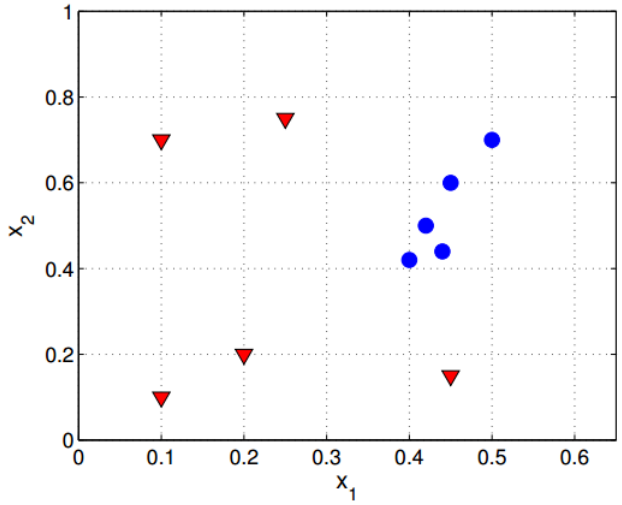
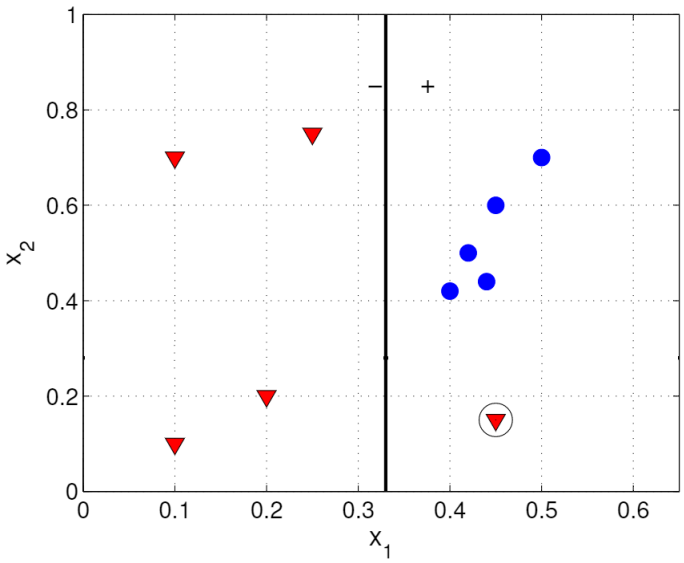


Figure 4: Triangles are negative (-) points and circles are positive (+) points.

(a) In Figure 4, draw the decision boundary corresponding to the first decision stump that the boosting algorithm would choose. Indicate +/- side of the decision boundary.

(b) In Figure 4, circle the point(s) that have the highest weight after the first boosting iteration.

Solution:



Problem 6 (EM for Naive Bayes). Suppose that we are given labeled data $\{(\mathbf{x}^{(1)}, t^{(1)}), ..., (\mathbf{x}^{(l)}, t^{(l)})\}$ and unlabeled data $\{\mathbf{x}^{(l+1)}, ..., \mathbf{x}^{(l+u)}\}$, where the input dimension is D. For every unlabelled example $\mathbf{x}^{(l+i)}$ there is a latent variable $t^{(l+i)}$ which we do not observe.

We have the following additional assumptions:

- The data is binary valued: i.e., each coordinate x_i takes a value of 0 or 1 for all $i \in \{1, ..., D\}$.
- The target variable t can take one out of $\{1, ..., C\}$.
- The data distribution is parameterized by θ, π, and follows the Naive Bayes assumption:

$$\begin{aligned} P(\mathbf{x}, t) &= P(t) \prod_{d=1}^{D} P(x_d|t) \\ P(t = c) &= \pi_c, \qquad \text{for } c = 1, ..., C \\ P(x_d = 1|t = c) &= \theta_{dc}, \qquad \text{for } d = 1, ..., D \text{ and } c = 1, ..., C, \end{aligned}$$

  where π_c are non-negative, multinomial probabilities (i.e., $\sum_{c=1}^{C} \pi_c = 1$), and θ_{dc} are Bernoulli probabilities. Since we have unlabeled data, our goal is to maximize the following hybrid objective function, where $\lambda > 0$ is some parameter.

$$\sum_{i=1}^{l} \log P(\mathbf{x}^{(i)}, t^{(i)} \mid \theta, \pi) + \lambda \sum_{i=l+1}^{l+u} \log P(\mathbf{x}^{(i)} \mid \theta, \pi)$$

  This problem can be solved by applying EM algorithm.

(A) Derive the E-step for this problem. *Hint: consider the target variables of the unlabeled data as the latent variables.*

Solution: We only need to compute the posterior of t given $\mathbf{x}$ for unlabeled data. Specifically, in E-step we compute the following for $i \in l + 1, ..., l + u$:

$$Q_c^{(i)} = P(t^{(i)} = c|\mathbf{x}^{(i)}) = \frac{\pi_c \prod_{d=1}^{D} P(x_d^{(i)}|c)}{\sum_{c'} \pi_{c'} \prod_{d=1}^{D} P(x_d^{(i)}|c')}$$

(B) Assuming you have already computed the correct conditional probabilities above, derive the solution for π_c in the M-step for this problem. (You do not have to specify the values you obtained in part (A) precisely)

Solution: In M-step, we can write the objective function (data-completion log-likelihood) as follows:

$$f(\pi,\theta) = \sum_{i=1}^{l} logP(\mathbf{x}^{(i)}, t^{(i)}) + \lambda \sum_{i=l+1}^{l+u} \sum_{c} Q_c^{(i)} logP(\mathbf{x}^{(i)}, c)$$

$$= \sum_{i=1}^{l} log \prod_{c=1}^{C} (\pi_c \prod_{d=1}^{D} (\theta_{dc})^{I(x_d^{(i)}=1)} (1-\theta_{dc})^{I(x_d^{(i)}=0)})^{I(t^{(i)}=c)}$$

$$+\lambda \sum_{i=l+1}^{l+u} \sum_{c} Q_c^{(i)} log \prod_{c=1}^{C} (\pi_c \prod_{d=1}^{D} (\theta_{dc})^{I(x_d^{(i)}=1)} (1-\theta_{dc})^{I(x_d^{(i)}=0)})$$

$$= \sum_{i=1}^{l} \sum_{c=1}^{C} I(t^{(i)}=c)(log\pi_c + \sum_{d=1}^{D} [I(x_d^{(i)}=1) log(\theta_{dc}) + I(x_d^{(i)}=0) log(1-\theta_{dc})])$$

$$+\lambda \sum_{i=l+1}^{l+u} \sum_{c=1}^{C} Q_c^{(i)} (log\pi_c + \sum_{d=1}^{D} [I(x_d^{(i)}=1) log(\theta_{dc}) + I(x_d^{(i)}=0) log(1-\theta_{dc})])$$

Taking the derivative with respect to π_c, we consider a Lagrange multiplier for the constraint $1 - \sum_{c=1}^{C} \pi_c = 0$. The Lagrangian is written as:

$$L = f(\pi,\theta) + \mu(1 - \sum_{c=1}^{C} \pi_c)$$

Taking a derivative with respect to π_c:

$$\frac{\partial L}{\partial \pi_c} = \frac{1}{\pi_c} \sum_{i=1}^{l} I(t^{(i)}=c) + \frac{\lambda}{\pi_c} \sum_{i=l+1}^{l+u} Q_c^{(i)} - \mu = 0$$

Therefore, we get

$$\pi_c = \frac{1}{\mu} (\sum_{i=1}^{l} I(t^{(i)}=c) + \lambda \sum_{i=l+1}^{l+u} Q_c^{(i)})$$

Since $\sum_{c=1}^{C} \pi_c = 1$, we finally get:

$$\pi_c = \frac{\sum_{i=1}^{l} I(t^{(i)}=c) + \lambda \sum_{i=l+1}^{l+u} Q_c^{(i)}}{\sum_{c'} (\sum_{i=1}^{l} I(t^{(i)}=c') + \lambda \sum_{i=l+1}^{l+u} Q_{c'}^{(i)})} = \frac{\sum_{i=1}^{l} I(t^{(i)}=c) + \lambda \sum_{i=l+1}^{l+u} Q_c^{(i)}}{l + \lambda u}$$

Problem 7 (Clustering). Assume we are given n examples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Given K, recall that the standard K-means algorithm is an iterative procedure that sequentially updates the cluster means and the cluster assignments for each cluster. That is, it chooses some initial means $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$ and then, for $t = 1, 2, \ldots$:

1. Update assignments for all $i = 1, \ldots, n$: $\quad a_i^t := \arg\min_{k \in [K]} \|\mathbf{x}_i - \mu^t{}_k\|$.
2. Update cluster centers for all $k = 1, \ldots, K$: $\quad \mu_k^{t+1} := \frac{1}{|\{i : a_i^t = k\}|} \sum_{i : a_i^t = k} \mathbf{x}_i$.

Now consider an alternative algorithm, *slower-K-means*, which is a slightly modified version: on step 2 of the above procedure, slower-K-means only moves "halfway" to the new mean. I.e., it sets

$$\mu_k^{t+1} := \frac{1}{2}\mu_k^t + \frac{1}{2}\frac{1}{|\{i : a_i^t = k\}|} \sum_{i : a_i^t = k} \mathbf{x}_i.$$

Recall that the K-means algorithm minimizes a certain objective function $J(\mu_1, \ldots, \mu_K; \mathbf{x}_1, \ldots, \mathbf{x}_n)$, and it can be shown that $J(\cdot)$ decreases (or stays constant) after every step t of the K-means algorithm. Question: is it also true that $J(\cdot)$ never increases after any step t of slower-K-means? State your answer and prove it.

The objective function for K-means clustering is

$$J(\mu_1, \ldots, \mu_K; \mathbf{x}_1, \ldots, \mathbf{x}_n) := \sum_{i=1}^n \min_{j=1,\ldots,k} \|\mathbf{x}_i - \mu_j\|^2.$$

We can write this also as a joint optimization $J(\boldsymbol{\mu}, \mathbf{a}; \mathbf{x}_{1:n}) := \sum_{i=1}^n \|\mathbf{x}_i - \mu_{a_i}\|^2$, where here we use the symbol $\boldsymbol{\mu}$ to refer to the joint vector $(\mu_1, \ldots, \mu_K)$ and $\mathbf{a}$ as the assignments $a_1, \ldots, a_n$. The K-means algorithm can be thought of as an alternating minimization where first we optimize $J(\boldsymbol{\mu}, \mathbf{a}; \mathbf{x}_{1:n})$ in $\mathbf{a}$ (updating cluster assignments), and then we optimize over $\boldsymbol{\mu}$ (updating cluster means). Both algorithms in question set the assignments a_i^t in the same way, but the original K-means algorithm would update the cluster means to be $\tilde{\mu}_k^{t+1} := \frac{1}{|\{i : a_i^t = k\}|} \sum_{i : a_i^t = k} \mathbf{x}_i$.

Using this $\tilde{\mu}$ notation, we see that we can write the slower-K-means algorithm as taking the *average* between the current cluster means and the cluster means that would have been selected by the K-means algorithm. That is:

$$\boldsymbol{\mu}^{t+1} := \frac{1}{2}\boldsymbol{\mu}^t + \frac{1}{2}\tilde{\boldsymbol{\mu}}^{t+1},$$

Because the K-means algorithm (non-strictly) decreases the objective function on each round, we have (omitting the $\mathbf{x}$'s):

$$J(\tilde{\boldsymbol{\mu}}^{t+1}, \mathbf{a}^t) \leq J(\boldsymbol{\mu}^t, \mathbf{a}^t).$$

Let's now fix $\mathbf{a}$, and let us note that $J(\boldsymbol{\mu}, \mathbf{a})$ is convex in $\boldsymbol{\mu}$ (it's just a sum of quadratics!). So combining this with the above (and omitting $\mathbf{a}$), we have

$$J(\boldsymbol{\mu}^{t+1}) = J\left(\frac{1}{2}\tilde{\boldsymbol{\mu}}^{t+1} + \frac{1}{2}\boldsymbol{\mu}^t\right) \overbrace{\leq}^{\text{convexity!}} \frac{1}{2}J(\tilde{\boldsymbol{\mu}}^{t+1}) + \frac{1}{2}J(\boldsymbol{\mu}^t) \leq \frac{1}{2}J(\boldsymbol{\mu}^t) + \frac{1}{2}J(\boldsymbol{\mu}^t) = J(\boldsymbol{\mu}^t).$$

This proves that the slower-K-means algorithm also (non-strictly) decreases $J()$, as desired.

Problem 8 (Decision Tree). You are tasked with learning to automatically classify candy bars as "Popular" or "Unpopular". The following eight candy bars with features (Size, Sweetness, and HasChocolate) and binary labels (Popular or Unpopular) are given:

Size	Sweetness	HasChocolate	Classification
L	Very	Y	Popular
L	Very	N	Unpopular
M	Very	N	Unpopular
L	Mild	Y	Popular
S	Medium	Y	Unpopular
L	Very	Y	Popular
S	Mild	Y	Popular
S	Mild	N	Unpopular

Construct (however you want) a binary decision tree for these data that achieves zero training error and uses the fewest number of decision nodes.

Solution: