# 10-701 Midterm Exam, Spring 2010

Professors Eric Xing, Aarti Singh, and Tom Mitchell

March 3, 2010

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be **15** numbered pages in this exam (including this cover sheet).
3. Open book, open notes. No computers, phones, or internet access.
4. If you need more room to work out your answer to a question, use the back of the page and clearly indicate that we should look there.
5. Work efficiently. Some questions are easier, some more difficult. Give yourself time to answer all of the easy ones.
6. You have 80 minutes.
7. Good luck!

**Grading:** The exam was designed by the instructors and graded by the TAs. Beside each question, you will find the name of the TA who graded it. Please read the solution first, and then direct any question or grading concern to the corresponding TA.

| Question | Topic | Max. score | Score |
|---|---|---|---|
| 1 | Short questions | 18 | |
| 2 | Naive Bayes | 15 | |
| 3 | Regression, Model Selection | 19 | |
| 4 | Hidden markov models | 18 | |
| 5 | EM and Gaussian Mixtures | 20 | |
| Total | | 90 | |

# 1 Short answer questions [18 pts] (Graded by Amr )

## 1.1 Decision trees [2 pt]

Give at least one advantage and one limitation of the decision tree algorithm.

A possible answer among many is:
**advantage:** Easy to interpret.
**limitation:** Its greedy procedure does not guarantee optimal solution.

## 1.2 MLE/MAP [2 pt]

In ridge regression we assume $Y = X\beta$ and estimate $\beta$ according to:

$$\hat{\beta} = \arg\min_{\beta} \sum_{j=1}^{N} (Y_j - X_j\beta)^2 + \lambda\beta^T\beta$$

What kind of prior over $\beta$ does this imply?

**Answer:** Zero-mean Guassian distribution.

## 1.3 Bias/Variance [2 pt]

For ridge regression, how will the bias and variance in our estimate $\hat{\beta}$ change as the number of training examples $N$ increases? Assume the regularization parameter $\lambda$ is fixed.

**Answer:** As we have more training data, both bias and variance will decrease. The case for the variance is easy to see. For the bias, note that we are optimizing:

$$\hat{\beta} = \arg\min_{\beta} \sum_{j=1}^{N} (Y_j - X_j\beta)^2 + \lambda\beta^T\beta$$

When $N$ increases, the first term will dominate the second term (since $\lambda$ is fixed). This means that the more data we have, the better it is to move the weight of more features aways from zeros to fit the data. This is true since the gain in decreasing the squared error will outweigh the penalty enforced by the second term. As $N \to \infty$, the second term will be completely ignored. In fact, it is easy to see that $\beta_{ridge} \to \beta_{unregularized}$ on the limit. Recall that $\beta_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T y$. Then as $N \to \infty$, the diagonal of $\mathbf{X}^T\mathbf{X}$, call it $d_{ii}$, will also go to $\infty$ since $d_{ii} = \sum_{n=1}^{N} x_{n,i}^2$. Thus $(\mathbf{X}^T\mathbf{X} + \lambda I) \to \mathbf{X}^T\mathbf{X}$ and $\beta_{ridge} \to \beta_{unregularized}$ which is unbiased.

**Grading note:** If you said that the bias will not change because it doesn't depend on the data, you didn't lose points although this is NOT true in this case. When regularization is involved, the bias of the classifier depends on both the value of the regularize and the amount of the data as we showed above. Since I felt this was a very subtle case, I didn't take points here. We were looking for an answer that argues that as $N$ increases, the second term will be given less weight in the optimization problem. Also, you lost points if you did not explain your answer in one or two words. Overall, almost no one lost more than 1 point in this question.

## 1.4 Model Selection [2 pt]

Answer TRUE or FALSE.

- Algorithm A is better than Algorithm B if the training error of algorithm A is better than that of B

  **False** since classifier A might be overfitting the data. Performance should be measured using test data (i.e. data that was not used in training the classifiers)

- For any classification problem, the Bayes optimal classifier can achieve an error rate of 0.

  **False** The Bayes optimal classifier lower bounds the achievable error by any classifier, but this lower bound need not be 0. For instance, in a binary classification problem , if $P(Y = 1|X = x) = .9$ under the true distribution, then the Bayes optimal classifier will predict $Y = 1$ whenever it sees $X = x$, and thus will make an error 10% of the times $(X = x)$ since $P(Y = 0|X = x) = .1$.

## 1.5 Neural Networks [2 pt]

Answer TRUE or FALSE

- The decision boundary learned by a neural network is always non-linear

  **False** as the decision boundary can be linear if all threshold units are linear.

- Regardless of the size of the neural network, the backpropagation algorithm can always find the globally optimal weights for the neural network.

  **False** It needn't since for a general multi-layer NN with non-linear threshold units, the function optimized by the backpropagation algorithm is not convex and has lots of local minimal points.

## 1.6 Regression [3 pt]

With regard to the regression problem depicted in Figure 1, label each of the following loss functions with the curve that minimizes it.

- $\sum_i |y_i - x_i w_1 - w_0|$
  **L3** since estimating $w$ based on the $|.|$ formulation is more robust to outliers.

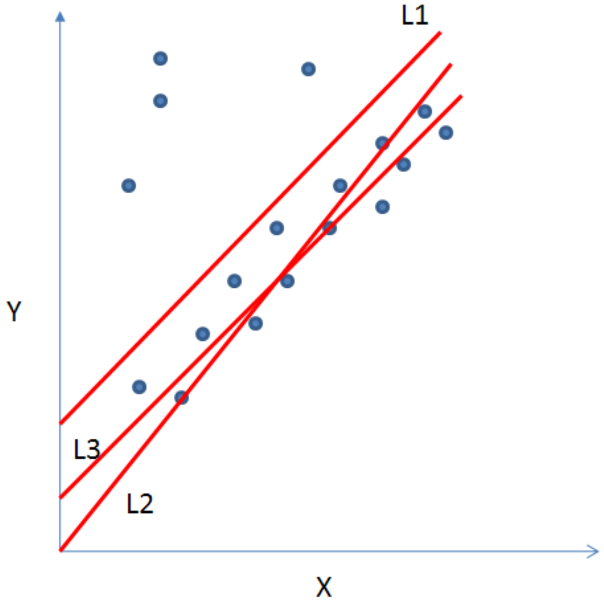


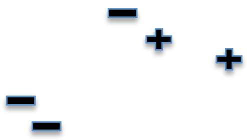
Figure 1: Curves for the regression problem.

- $\sum_i (y_i - x_i w_1 - w_0)^2$
  **L1** since estimating $w$ based on the squared error loss is more affected by outliers.

- $\sum_i |y_i - x_i w_1 - w_0| + 1000^{100}(w_0)^2$
  **L2** since $w_0$ will be set to 0 to avoid the harsh penalty

## 1.7 Cross Validation and Nearest Neighbor [3 pt]

Consider a nearest neighbor classifier that chooses the label for a test point to be the label of its nearest neighboring training example. What is its leave-one-out cross validated error for the following data. (+ and - indicate labels of the points).



**Answer:** 2/5. There are 5 data points, thus we will get 5-folds, each of which has 4 training data points and one test point. The 2 errors are due to the top most two points since they are the nearest neighbor to each other yet they have opposite labels.

## 1.8 EM algorithm [2 pt]

Answer TRUE or FALSE.

- The EM algorithm optimizes a lower bound on its objective function, which is the marginal likelihood $\prod_i P(x_i)$ of the observed data points $x_1, x_2, \ldots x_N$.

  **TRUE**. See slides 42 in lecture 10.

- The EM algorithm is guaranteed to never decrease the value of its objective function on any iteration.

  **TRUE:** See slides 45-50 in lecture 10.

- The objective function optimized by the EM algorithm can also be optimized by a gradient descent algorithm which will find the global optimal solution, whereas EM finds its solution more quickly but may return only a locally optimal solution.

  **False.** The only false part is that the gradient descent algorithm will find the global optimal solution. Gradient descent can also get stuck in a local optima.

| $x_1$ | $x_2$ | $y$ | $p_{\mathcal{D}}(x_1, x_2, y)$ |
|---|---|---|---|
| 0 | 0 | 0 | .15 |
| 0 | 0 | 1 | .25 |
| 0 | 1 | 0 | .05 |
| 0 | 1 | 1 | .08 |
| 1 | 0 | 0 | .1 |
| 1 | 0 | 1 | .02 |
| 1 | 1 | 0 | .2 |
| 1 | 1 | 1 | .15 |

(a) Joint distribution

|  | $x_1 = 0$ | $x_1 = 1$ |
|---|---|---|
| $y = 0$ | .4 | .6 |
| $y = 1$ | .66 | .34 |

(b) $P_{\mathcal{D}}(x_1|y)$

|  | $x_2 = 0$ | $x_2 = 1$ |
|---|---|---|
| $y = 0$ | .5 | .5 |
| $y = 1$ | .54 | .46 |

(c) $P_{\mathcal{D}}(x_2|y)$

| $y$ | $P_{\mathcal{D}}(y)$ |
|---|---|
| $y = 0$ | .5 |
| $y = 1$ | .5 |

(d) $p_{\mathcal{D}}(y)$

Figure 2: Joint and marginal probabilities

# 2 Naive Bayes [15 pt] (Graded by Ni)

## 2.1 Bayes Optimal and Naive Bayes Classifier

Consider the joint probability distribution over 3 boolean variables $x_1, x_2, y$ given in Figure 2(a). Consider also the marginal probabilities for this same distribution, given in Figures 2b,c, and d. Note these distributions would be used by a Naive Bayes classifier.

(1) [2 pt] What is the decision rule used by the Bayes optimal classifier?

**Answer:** if $P(y = 1|x_1, x_2) > P(y = 0|x_1, x_2)$ then $\hat{y} = 1$, or else $\hat{y} = 0$

(2) [2 pt] Express $P_{\mathcal{D}}(y = 0|x_1, x_2)$ in terms of $P_{\mathcal{D}}(x_1, x_2, y = 0)$ and $P_{\mathcal{D}}(x_1, x_2, y = 1)$

**Answer:**

$$P_{\mathcal{D}}(y = 0|x_1, x_2) = \frac{P_{\mathcal{D}}(x_1, x_2, y = 0)}{P_{\mathcal{D}}(x_1, x_2, y = 0) + P_{\mathcal{D}}(x_1, x_2, y = 1)}$$

(3) [2 pt] Write out an expression for the value of $P(y = 1|x_1 = 1, x_2 = 0)$ predicted by the *Bayes optimal classifier*. Your expression should involve numbers from these tables, but please do not bother to perform the multiplications and other arithmetic.

**Answer:**

$$P(y = 1|x_1, x_2) = \frac{P_{\mathcal{D}}(x_1, x_2, y = 1)}{P_{\mathcal{D}}(x_1, x_2, y = 0) + P_{\mathcal{D}}(x_1, x_2, y = 1)} = \frac{.02}{.02 + .1}$$

(4) [2 pt] Write out an expression for the value of $P(y = 1|x_1 = 1, x_2 = 0)$ predicted by the *naive Bayes classifier*. Again, do not bother to perform the multiplications and other arithmetic.

**Answer:**

$$
\begin{aligned}
P(y=1|x_1,x_2) &= \frac{P(x_1,x_2,y=1)}{P(x_1,x_2,y=0)+P(x_1,x_2,y=1)} && (1)\\
&= \frac{P_{\mathcal{D}}(y=1)P_{\mathcal{D}}(x_1|y=1)P_{\mathcal{D}}(x_2|y=1)}{P_{\mathcal{D}}(y=0)P_{\mathcal{D}}(x_1|y=0)P_{\mathcal{D}}(x_2|y=0)+P_{\mathcal{D}}(y=1)P_{\mathcal{D}}(x_1|y=1)P_{\mathcal{D}}(x_2|y=1)} && (2)
\end{aligned}
$$

Therefore

$$P(y=1|x_1=1,x_2=0) = \frac{.5 * .34 * .54}{.5 * .34 * .54 + .5 * .6 * .5} \tag{3}$$

(5) [2 pt] The expressions you wrote down for (3) and (4) should be unequal. Explain why *in one sentence.*

**Answer:** Bayes optimal classifier does not have the assumption of conditional independency.

## 2.2 Beyond Naive Bayes [5 pt]

Suppose you have a data set involving five random variables: $x_1, x_2, x_3, x_4$ and $y$. Variables $x_i$ and $x_j$ are conditionally independent given $y$, for all $i$ and $j$ except for the pair $x_3$ and $x_4$, which are *not* conditionally independent. Therefore, you can't quite use Naive Bayes unless you extend it to handle the dependence between $x_3$ and $x_4$. Write down the decision rule you would use in place of the Naive Bayes rule, to correctly model this data set. (*Hint:* try rederiving the Naive Bayes decision rule, but avoiding the conditional independence assumption for $x_3$ and $x_4$).

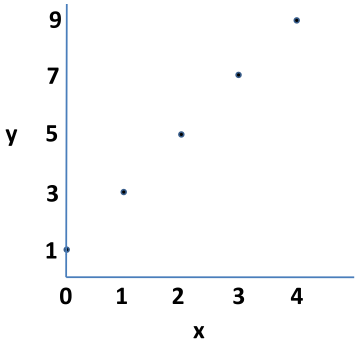
**Answer:** If this quality is larger than 1 then $\hat{y}=1$, or else $\hat{y}=0$

$$\frac{P_{\mathcal{D}}(y=1)P_{\mathcal{D}}(x_1|y=1)P_{\mathcal{D}}(x_2|y=1)P_{\mathcal{D}}(x_3,x_4|y=1)}{P_{\mathcal{D}}(y=0)P_{\mathcal{D}}(x_1|y=0)P_{\mathcal{D}}(x_2|y=0)P_{\mathcal{D}}(x_3,x_4|y=0)} \tag{4}$$

# 3 Regression and Model Selection [19 pt] (Graded by Ni)

1. (a) [2 pt] Find the linear least squares fit $f(x) = a + bx$ to the following training data:

| x | y |
|---|---|
| 0 | 1 |
| 1 | 3 |
| 2 | 5 |
| 3 | 7 |
| 4 | 9 |



What is the training error?

**Answer:** 0

(b) [ 2 pt] What will be the quadratic least squares fit $f(x) = a+bx+cx^2$ to the same training data? What is the training error? Hint: You don't need to do any computations, just think about it.

**Answer:** c=0, and erro=0.

(c) [2 pt] Construct a data set for which the training error of a quadratic least square fit is zero, but the linear least square fit has non-zero training error.

**Answer:** (1,1),(0,0),(-1,1)

(d) [2 pt] Would you always prefer a higher-order polynomial fit? Why/why not?

**Answer:** No. it will overfit.

(e) [2 pt] Propose one model selection procedure to automatically choose the order of the polynomial?

**Answer:** Cross validation

2. (a) [ 5 pt] Recall the Nadaraya-Watson (constant fit) kernel regression

$$\widehat{f}(X) = \frac{\sum_{j=1}^{n} Y_j K(X, X_j)}{\sum_{j=1}^{N} K(X, X_j)}$$

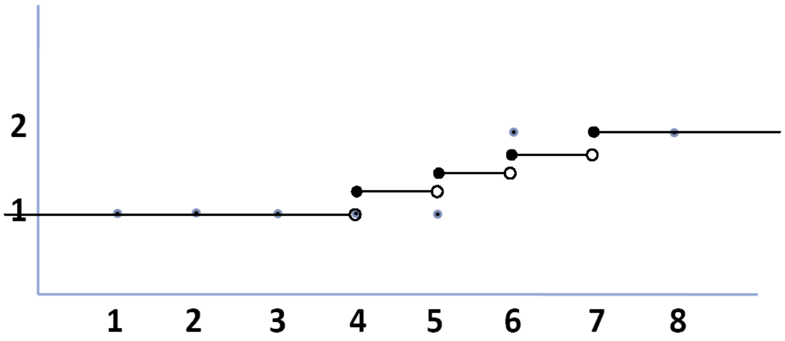
where $K$ denotes the kernel. If we use the boxcar kernel defined as

$$K(X, X_j) = \begin{cases} 1 & if - h \leq X - X_j < h \\ 0 & otherwise \end{cases}$$

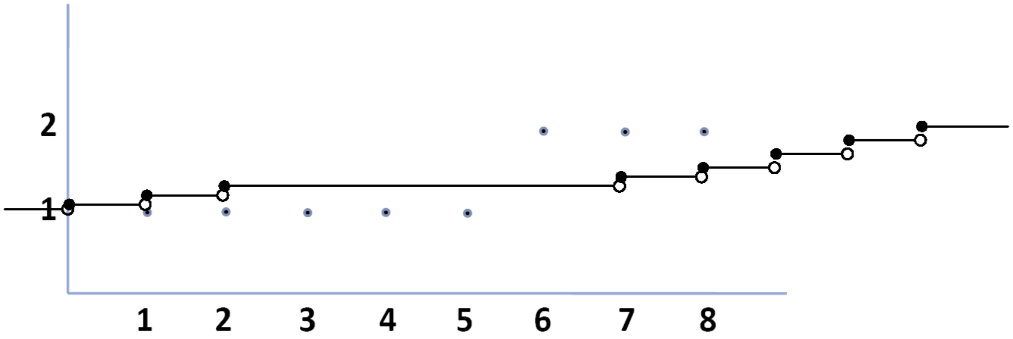
then the kernel regression is equivalent to simply averaging the labels $Y_j$ of points that are within a distance $h$ from the point $X$. Sketch the kernel regression fit for $h = 2, 6, 8$ to the following piece-wise constant training data: (Only a sketch is needed, the values don't need to be exact)

**Answer:**

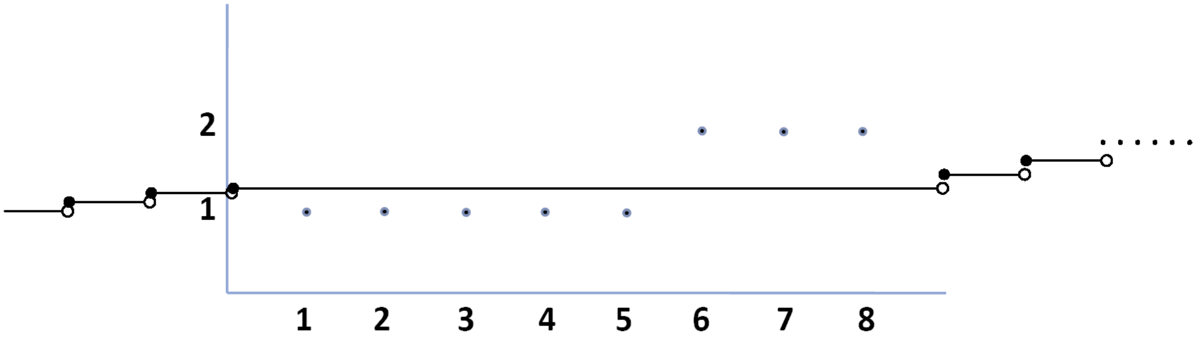
$h = 2$



$h = 6$



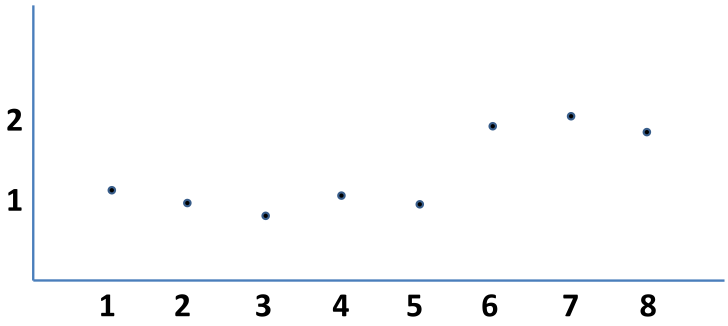
$h = 8$



(b) [2 pt] What value of h (bandwidth) would you choose and why?

**Answer:** Since the has very small noise, we prefer small h (e.g. 1 or 2) to reduce bias.

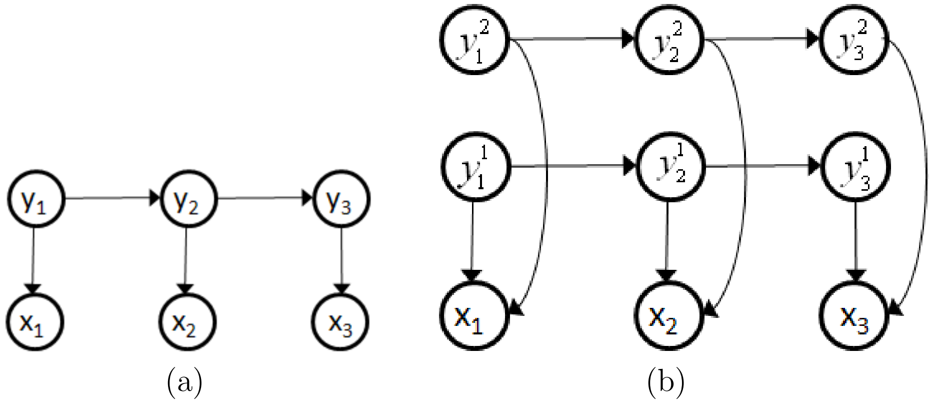
(c) [2 pt] Now consider that the data is corrupted by noise as below.



What value of h (bandwidth) would you choose and why? (No computations needed, only answer qualitiatively)

**Answer:** Since the data is noise, we prefer larger h (e.g. 2,3 or 4) to reduce variance. Even larger h would have too large bias.

# 4 HMM [ 18 pt] (graded by Field)



(a) (b)

Figure 3: (a) HMM. (b) FHMM

1. [2 pt] What is the difference between the Viterbi and the Baum-Welch algorithms?

   **Answer :** The Baum-Welch algorithm uses MLE to learn the parameters of a HMM, whereas Viterbi uses known parameters to find the single most likely sequence of hidden states.

2. [3 pt] Consider the hidden Markov model in Figure 3.a that consists of three observations: $x_1, x_2, x_3$ and three hidden states: $y_1, y_2, y_3$. Write down the joint probability distribution $P(x_1, x_2, x_3, y_1, y_2, y_3)$.

   **Answer :**

   $\pi(y_1)P(x_1|y_1)P(y_2|y_1)P(x_2|y_2)P(y_3|y_2)P(x_3|y_3)$

3. In a HMM, the observations are generated from a sequence of hidden states. However, in some applications, like speech, biology and NLP, the observations are known to be generated from the interaction of multiple independent state sequences. This gives rise to the Factorial HMM (FHMM) model depicted in Figure 3.b. Figure 3.b shows two independent state sequences: $(y_1^1, y_2^1, y_3^1)$ and $(y_1^2, y_2^2, y_3^2)$. At each time step, say $t$, the observation, $x_t$ is generated based on the values of the hidden states in both sequences: $y_t^1$ and $y_t^2$.

   i. [4 pt] Please write down the joint $P(x_1, x_2, x_3, y_1^1, y_2^1, y_3^1, y_1^2, y_2^2, y_3^2)$.

      **Answer :** $(\pi(y_1^1)\pi(y_1^2)P(x_1|y_1^1, y_1^2)) \times (P(y_2^1|y_1^1)P(y_2^2|y_1^2)P(x_2|y_2^1, y_2^2)) \times (P(y_3^1|y_2^1)P(y_3^2|y_2^2)P(x_3|y_3^1, y_3^2))$

ii. [3 pt] The FHMM is an example of a Bayesian Network. Please use the conditional independence property in HMM or Bayesian Networks to give a simpler expression of the conditional probability $P(y_3^1, y_1^2|y_2^1, y_2^2)$. (i.e., in terms of a product of simpler terms such as $P(y_3^1 \cdots)$ and $P(y_1^2 ...)$, you need to write down explicit expressions of these two terms in the form of either a conditional probability or a marginal probability.)

**Answer :** The key thing is that, *if* we do not have any observations, the sequences of hidden states are independent of each other, so $P(y_3^1, y_1^2|y_2^1, y_2^2) = P(y_3^1|y_2^1)P(y_1^2|y_2^2)$ Note that the second of these terms is *not* the transition probability, because we have reversed the direction, but there is still a dependence because knowing $y_2^2$ tells us something about what $y_1^2$ was.

iii. [3 pt] Can we write $P(y_2^1, y_2^2|x_1, x_2, x_3)$ as $P(y_2^1|x_1, x_2, x_3)P(y_2^2|x_1, x_2, x_3)$? Explain your answer.

**Answer :** No, because knowing either of the $y_2^i$ lets us explain away the other one.

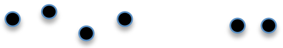
iv. [3 pt] Can we write $P(y_2^1, y_2^2)$ as $P(y_2^1)P(y_2^2)$? Explain your answer.

**Answer :** Yes, because we have no observations.

# 5 EM, Gaussian Mixture Models, Semisupervised Learning [20 pt] (graded by Field)

This question concerns training Gaussian Mixture Models (GMM's). Throughout, we will assume there are two Gaussian components in the GMM. We will use $\mu_0, \mu_1, \sigma_0$ and $\sigma_1$ to define the means and variances of these two components, and will use $\pi_0$ and $(1-\pi_0)$ to denote the mixture proportions of the two Gaussians (i.e., $p(x) = \pi_0 N(\mu_0, \sigma_0 I) + (1-\pi_0) N(\mu_1, \sigma_1 I)$). We will also use $\theta$ to refer to the entire collection of parameters $\langle \mu_0, \mu_1, \sigma_0, \sigma_1, \pi_0 \rangle$ defining the mixture model $p(x)$.

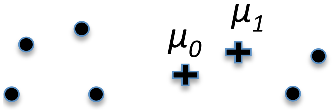
(1) [3 pt] Consider the set of training data below, and two clustering algorithms: K-Means, and a Gaussian Mixture Model (GMM) trained using EM. Will these two clustering algorithms produce the same cluster centers (means) for this data set? In one sentence, explain why or why not.



**Answer :** Ok, almost everybody got this problem wrong, so let me explain carefully. Either algorithm will find the clusters just fine. But the difference lies in that k-means uses hard assignment of each point to a single cluster, whereas GMM uses soft assignment, where every point has non-zero (though possibly small) probability of being in each cluster. So in k-means, the means of the clusters are determined by an average of the points assigned to that cluster, but in GMM the means of each cluster are (differently) weighted averages of all points. This has the effect of skewing the center of the left cluster to the right, and the center of the right cluster to the left.

You could argue that this is a downside of the EM algorithm, that it still give some weight to points that are clearly in the other cluster. On the other hand, each point in the other cluster could just *maybe* be an outlier from the first cluster, so this skewing is not completely unreasonable. Regardless whether you like or dislike this phenomenon, you should be aware of it and understand where it comes from.

(2) Consider applying EM to train a Gaussian Mixture Model (GMM) to cluster the data below into two clusters. The '+' points indicate the current means $\mu_0$ and $\mu_1$ of the two Gaussian mixture components after the k$^{th}$ iteration of EM.



(2a) [2 pt] Draw on the figure the directions in which $\mu_0$ and $\mu_1$ will move during the next M-step.

**Answer :** $\mu_0$ moves to the left, and $\mu_1$ moves to the right.

(2b) [2 pt] Will the marginal likelihood of the data, $\prod_j P(x^j|\theta)$, increase or decrease on the next EM iteration? [note we use superscripts in this question to index training examples.] Explain your reasoning in *one sentence.*

**Answer :** Increase. Each iteration of the EM algorithm increases to likelihood of the data, unless you happen to be exactly at a local optimum.

(2c) [2 pt] Will the estimate of $\pi_0$ increase or decrease on the next EM step? Explain your reasoning in *one sentence.*

**Answer :** Yes. $\pi_0$ is determined by adding the probabilities of all points that they are in cluster 1. In the current configuration, $\mu_1$ is close enough to $\mu_0$ that it will be stealing a lot of this probability mass, so that $\pi_0$ and $\pi_1$ will be pretty close to each other.

(3) Next let's consider the relationship between a Gaussian Naive Bayes (GNB) classifier and the above Gaussian Mixture Model (GMM). It is easy to see that they involve the same probabilistic model. Our usual GNB classifier assumes $p(Y|X)$ is of the form:

$$p(Y|X) = \frac{p(Y)\prod_i p(X_i|Y)}{p(X)}$$

where $Y$ is a Bernoulli random variable (i.e., $P(Y = 0) = \pi_0$). It also assumes each feature $X_i$ is governed by a Gaussian distribution conditioned on $Y$. For simplicity, let's assume all features have the same variance, so

$$P(X_i|Y = k) \sim N(\mu_{k,i}, \sigma)$$

Notice this GNB generative model is identical to that of our GMM above (plus the simplifying assumption of identical $\sigma$'s). In other words, both models assume we generate data points by choosing a $Y$ according to $\pi_0$, then drawing an $X$ according to a Gaussian conditioned on $Y$.

The difference, of course, is that we train GNB using labeled data in which the $Y$ values are known, whereas we train GMM assuming $Y$ values are unknown.

(3a) [3 pt] When training this GNB, we choose the set of parameters $\theta$ that maximize the data likelihood.

$$\arg\max_\theta \prod_j P(x^j, y^j|\theta)$$

where again we use the superscript $j$ to denote the j$^{th}$ training example. Write down the objective that EM seeks to maximize when it trains the same model, without known values for $y^j$.

**Answer :** $\Pi_j P(x^j|\theta) = \Pi_j \left(\sum_y P(x^j, y|\theta)\right)$

(3b) [2 pt] Write down the algorithms for the E and M steps in the standard EM algorithm for mixture of Gaussians.

E:

$$\gamma_{ik} = P(y^k)N(x^i|\mu_k, \sigma_k)$$

M:

$$\begin{aligned} \pi_k &= \frac{\sum_i \gamma_{ik}}{\sum_{ij} \gamma_{ij}} \\ \mu_k &= \frac{\sum_i \gamma_{ik} x^i}{\sum_i \gamma_{ik}} \\ \sigma_k^2 &= \frac{\sum_i \gamma_{ik}(x^i - \mu_k)^2}{\sum_i \gamma_{ik}} \end{aligned}$$

(3c) [3 pt] GNB trains using labeled examples, GMM trains using unlabeled examples. Suppose we have a set of training data in which we have some of each. We have known $y$ values for $x^1, x^2, \ldots x^m$, but have additional unlabeled examples $x^{m+1} \ldots x^{m+n}$ without known values for $y$. How would you propose to train the generative model in this case? Write down your modified E and M steps:

E:

Same as above for $x^{m+1} \ldots x^{m+n}$, and for $i \leq m$, $\gamma_{ij} = \delta_{j,y(i)}$.

M:

Same as above.

(3d) [3 pt] Write down the objective function that your modified EM is maximizing. In your expression, distinguish between the training examples for which $y$ is known and unknown.

**Answer :** $(\Pi_{i=1}^{m} P(x^i, y^i|\theta)) \left(\Pi_{i=m+1}^{m+n} P(x^i|\theta)\right)$