In The Name of God, The Compassionate, The Merciful

Name: ...**Problems' keys**...

Student ID#: .....

Statistical Pattern Recognition (CE-725)
Department of Computer Engineering
Sharif University of Technology

Final Exam Solution - Spring 2012
(150 minutes – 100+5 points)

1) Basic Concepts (15 points)

a) True or false questions: For each of the following parts, specify that the given statement is true or false. In the case of true, provide a brief explanation, otherwise, propose a counter example.

i- ..... The kernel $K(x_1, x_2)$ is symmetric, where x_i and x_j are the feature vectors for i-th and j-th examples.

ii- ..... Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could in principle be reproduced with an SVM and a polynomial kernel.

iii- ..... After training a SVM, we can discard all examples which are not support vectors and can still classify new examples.

b) What would happen if the activation function at hidden and output layer in MLP be linear? Explain why this simpler activation function is not normally used in MLPs although it would simplify and accelerate the calculations for the back-propagation algorithm?

Sol: a)

i- True. $K(x_1,x_2)=\phi(x_1)\phi(x_2)=\phi(x_2)\phi(x_1)=k(x_2,x_1)$.

ii- True. Since class-conditional Gaussians always yield quadratic decision boundaries, they can be reproduced with an SVM with kernel of degree less than or equal to two.

iii- True. Only support vectors affect the boundary.

b) If we use this activation function, MLP becomes like Perceptron (a linear classifier). To be more specific, we can write the weights from the input to the hidden layer as a matrix W^{HI}, the weights from the hidden to output layer as W^{OH}, and the bias at the hidden and output layer as vectors b^H and b^O. Using vector and matrix multiplication, the hidden activations can be written as $H = b^H + W^{HI} * I$. And, the output activations can be written as

$$\begin{aligned} O &= b^O + W^{OH} * H \\ &= b^O + W^{OH} * (b^H + W^{HI} * I) \\ &= (b^O + W^{OH} * b^H) + (W^{OH} * W^{HI}) * I \\ &= b^{OI} + W^{OI} * I;\ b^{OI} = b^O + W^{OH} * b^H,\ W^{OI} = W^{OH} * W^{HI} \end{aligned}$$

Therefore, the same function can be computed with a simpler network, with no hidden layer, using the weights W^{OI} and bias b^{OI}.

2) Support Vector Machines (20 points)

Consider the following data points and labels:

Data point	Label
$x_1 = (1,1)$	1
$x_2 = (2,1)$	1
$x_3 = (2,0)$	1
$x_4 = (1,2)$	-1
$x_5 = (2,2)$	-1
$x_6 = (1,-3)$	-1

Suppose that we use following embedding function to separate two classes by a large margin classifier.

$$\phi(x) = (x_1^2 + x_2^2, x_1 - x_2)$$

a) Find the support vectors, visually.

b) Find the parameters of the SVM classifier *(w, w_0, λ_i)*.

c) Introduce an embedding function from 2-D to 1-D that separates original data points linearly.

Sol:

a) Transformed data points are:

Original	Transformed
$x_1 = (1,1)$	$x'_1 = (2,0)$
$x_2 = (2,1)$	$x'_2 = (5,1)$
$x_3 = (2,0)$	$x'_3 = (4,2)$
$x_4 = (1,2)$	$x'_4 = (5,-1)$
$x_5 = (2,2)$	$x'_5 = (8,0)$
$x_6 = (1,-3)$	$x'_6 = (10,4)$

Then, x2, x4 and x6 are support vectors.

b) $y(w^T x + w_0) = 1$ for all support vectors, then $w=(-1,1)$, $b=5$. $\sum y_i \lambda_i = 0$, then $\lambda_1 - \lambda_2 - \lambda_3 = 0$. In addition $w = \sum \lambda_i y_i x_i$, then $\lambda_1 (5,1)^T - \lambda_2 (5,-1)^T - \lambda_3 (10,4)^T = (1,-1)$, and $(\lambda_1, \lambda_2, \lambda_3) = (-1, -4/5, -1/5)$.

c) The 1-D embedding function is $\varphi(x,y) = |y|$.

3) Graphical Methods (25 points)

a) Consider an HMM with three nodes $\{S_1, S_2, S_3\}$, outputs $\{A,B\}$, initial state probabilities {1,0,0}, state transition probability matrix *A*, and output probability matrix *B*. Compute $P(O_1=B, O_2=B, \ldots, O_{200}=B)$ in the given HMM.

$$A = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 1 & 0 \end{pmatrix}$$

b) Given the following graphical model, which of the following statements are true, regardless of the conditional probability distributions?

b1) $P(D,H) = P(D)P(H)$

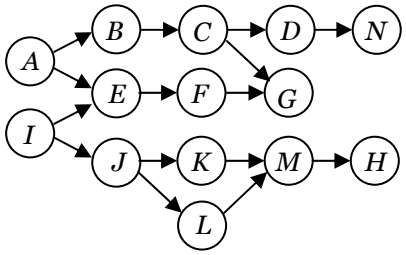
b2) $P(A, I) = P(A)P(I)$

b3) $P(A, I|G) = P(A|G)P(I|G)$

b4) $P(J,G|F) = P(J|F)P(G|F)$

b5) $P(J,M|K,L) = P(J|K,L)P(M|K,L)$

b6) $P(E,C|A,G) = P(E|A,G)P(C|A,G)$



Sol:

a) We can write the probability as:

$$P(O_1 = B, \ldots, O_{200} = B) = \sum_i P(O_1 = B, \ldots, O_{200} = B \mid q_{200} = S_i) P_{200}(S_i)$$

And we have,

$$P(O_1 = B, \ldots, O_{200} = B \mid q_{200} = S_1) = P(O_1 = B, \ldots, O_{200} = B \mid q_{200} = S_2) = 2^{-200}$$
$$P(O_1 = B, \ldots, O_{200} = B \mid q_{200} = S_3) = 0$$

$$P_{200}(S_1) = 2^{-199}, P_{200}(S_2) = P_{200}(S_3) = \frac{1}{2}(1 - P_{200}(s_1))$$

Then, $P(O_1=B, O_2=B, \ldots, O_{200}=B) = 2^{-399} + 2^{-201}(1-2^{-199})$

b)

b1) True. Because there is no active trails in any possible paths from *D* to *H* (*DCGFEIJKMH*, *DCGFEIJLMH*, *DCBAEIJKMH* and *DCBAEIJLMH*).

b2) True. Because there is no active trails in any possible paths from *A* to *I* (*AEI* and *ABCGFEI*).

b3) False. There is an active trail on the path *ABCGFEI*.

b4) False: There is an active trail on the path *GCBAEIJ* (*E* is descendant of *F*).
b5) True: Because there is no active trails in any possible paths from *J* to *M* (*JKM* and *JLM*).
b6) False: There is an active trail on the path *EFGC*.

4) Expectation and Maximization (20 points)

Consider a random variable *x* that is categorical with *M* possible values {1,2,…,M}. Suppose that *x* is represented as a vector such that *x(i)=1* if *x* takes the *i*-th value, and $\sum_i x(i)=1$. The distribution of *x* is represented by a mixture of *K* discrete multinomial distributions such that:

$$p(x)=\sum_{k=1}^{K}\pi_k p(x\mid\mu_k) \text{ and } p(x\mid\mu_k)=\prod_{j=1}^{M}\mu_k(j)^{x(j)}$$

π_K denotes the mixing coefficient for the *k*-th component (or the prior probability that the hidden variable $z = k$), and $\mu_k(j)$ represents the probability $P(x(j)=1|z=k)$. Observed data points $\{x_i\}$, $i=1,\ldots,n$ derive the E and M steps to estimate π_K and $\mu_k(j)$ for all values of *k*, and *j*.

Sol: The hidden variables are z_{ij}s. z_{ij} is a binary variable which is 1 if x_i is drawn from the *j*-th distribution.

E Step:

$$\begin{aligned}P(z_{ik}=1\mid x_i,\theta)&=\frac{P(x_i\mid z_{ik}=1,\theta)P(z_{ik}=1\mid\theta)}{P(x_i\mid\theta)}\\&=\frac{P(x_i\mid\mu_k)\pi_k}{\sum_{j=1}^{K}P(x_i\mid\mu_j)\pi_j}\\&=\frac{\pi_k\prod_{j=1}^{M}\mu_k(j)^{x_i(j)}}{\sum_{j=1}^{K}\pi_j\prod_{l=1}^{M}\mu_j(l)^{x_i(l)}}\end{aligned}$$

M Step:

$$\begin{aligned}P(X,Z\mid\theta)&=\prod_{i=1}^{n}P(x_i,z_i\mid\theta)\\&=\prod_{i=1}^{n}P(x_i\mid z_i,\theta)P(z_i\mid\theta)\end{aligned}$$

We have,

$$P(z_{ik}=1\mid\theta)=\pi_k\Rightarrow P(z_i\mid\theta)=\prod_{k=1}^{K}\pi_k^{z_{ik}}$$
$$P(x_i\mid z_{ik}=1,\theta)=P(x_i\mid\mu_k)\Rightarrow P(x_i\mid z_i,\theta)=\prod_{k=1}^{K}P(x_i\mid\mu_k)^{z_{ik}}=\prod_{k=1}^{K}\prod_{j=1}^{M}\left(\mu_k(j)^{x_i(j)}\right)^{z_{ik}}$$

Substituting the above two values in the likelihood results in:

$$\begin{aligned}P(X,Z\mid\theta)&=\prod_{i=1}^{n}P(x_i\mid z_i,\theta)P(z_i\mid\theta)\\&=\prod_{i=1}^{n}\left(\prod_{k=1}^{K}\prod_{j=1}^{M}\left(\mu_k(j)^{x_i(j)}\right)^{z_{ik}}\right)\left(\prod_{k=1}^{K}\pi_k^{z_{ik}}\right)\\&=\prod_{i=1}^{n}\prod_{k=1}^{K}\left(\pi_k\prod_{j=1}^{M}\mu_k(j)^{x_i(j)}\right)^{z_{ik}}\end{aligned}$$

The log likelihood is:

$$L(X,Z\mid\theta)=\ln P(X,Z\mid\theta)=\sum_{i=1}^{n}\sum_{k=1}^{K}\left(z_{ik}\ln\pi_k+z_{ik}\sum_{j=1}^{M}x_i(j)\ln\mu_k(j)\right)$$

To estimate π_ks, fixing z_{ij}s and considering the constraint $\Sigma\pi_k=1$, and using Lagrangian multiplier we must optimize the following objective function:

$$L(\pi_k)=L(X,Z\mid\theta)+\lambda\left(\sum_{l=1}^{K}\pi_l-1\right)$$

Setting the differentiation to zero we have,

$$\frac{\partial L}{\partial \pi_k} = \sum_{i=1}^{n} \frac{z_{ik}}{\pi_k} + \lambda = 0 \Rightarrow \pi_k = \frac{1}{-\lambda} \sum_{i=1}^{n} z_{ik}$$

To calculate the value of λ we use the fact that $\sum_{l=1}^{K} \pi_l = 1$. Then,

$$\sum_{l=1}^{K} \frac{\sum_{i=1}^{n} z_{il}}{-\lambda} = 1$$

This equation results in $-\lambda = \sum_{l=1}^{K} \sum_{i=1}^{n} z_{il}$. We complete estimation by substituting the value of λ in the previous obtained equation for π_k. The final result is:

$$\pi_k = \frac{\sum_{i=1}^{n} z_{ik}}{\sum_{l=1}^{K} \sum_{i=1}^{n} z_{il}} = \frac{1}{n} \sum_{i=1}^{n} z_{ik}$$

In the same way, to estimate μ_ks, fixing z_{ij}s and considering the constraint $\Sigma_l \mu_k(l)=1$, and using Lagrangian multiplier we must optimize the following objective function:

$$L(\mu_k) = L(X, Z \mid \theta) + \lambda \left(\sum_{l=1}^{M} \mu_k(l) - 1 \right)$$

Setting the differentiation to zero we have,

$$\frac{\partial L}{\partial \mu_k(j)} = \sum_{i=1}^{n} \frac{z_{ik} x_i(j)}{\mu_k(j)} + \lambda = 0 \Rightarrow \mu_k(j) = \frac{1}{-\lambda} \sum_{i=1}^{n} z_{ik} x_i(j)$$

To calculate the value of λ we use the fact that $\sum_{l=1}^{M} \mu_k(l) = 1$. Then,

$$\frac{1}{-\lambda} \sum_{l=1}^{M} \sum_{i=1}^{n} z_{ik} x_{il} = 1$$

This equation results in $-\lambda = \sum_{l=1}^{M} \sum_{i=1}^{n} z_{ik} x_{il}$. We complete estimation by substituting the value of λ in the previous obtained equation for μ_k. The final result is:

$$\mu_k(j) = \frac{\sum_{i=1}^{n} z_{ik} x_{ij}}{\sum_{i=1}^{n} \sum_{l=1}^{M} z_{ik} x_{il}} \overset{\sum_{l=1}^{M} x_i(l)=1}{=} \frac{\sum_{i=1}^{n} z_{ik} x_{ij}}{\sum_{i=1}^{n} z_{ik}}$$

5) Clustering (15 points)

a) Assume we are trying to cluster the points $2^0, 2^1, 2^2, \ldots, 2^n$ (a total of $n+1$ points where $n+1=2^N$) using hierarchical clustering. We break ties by combining the two clusters in which the lowest number resides. For example, if the distance between clusters *A* and *B* is the same as the distance between clusters *C* and *D* we would chose *A* and *B* as the next two clusters to combine if *min{A,B} < min{C,D}* where *{A,B}* are the set of numbers assigned to *A* and *B*.

a1) If we are using Euclidian distance, draw a sketch of the hierarchical clustering tree we would obtain for each of single/complete linkage methods.

a2) Now assume we are using the distance function *d(p,q) = max(p,q)/min(p,q)*. Which of the single/complete linkage methods will result in a different tree from the one obtained in (a) when using this distance function? If you think that one or more of these methods will result in a different tree, sketch the new tree as well.

b) Consider the following algorithm to partition the data points to *K* clusters:

1. Calculate the pairwise distance $d(P_i, P_j)$ between every two data points P_i and P_j in the set of data points to be clustered and build a complete graph on the set of data points with edge weights = corresponding distances.
2. Generate the Minimum Spanning Tree of the graph i.e. Choose the subset of edges E' with minimum sum of weights such that $G' = (P,E')$ is a single connected tree.

3. Throw out the *K-1* edges with the heaviest weights to generate *K* disconnected trees corresponding to the *K* clusters.

Identify which of the clustering algorithms you saw in the class corresponds to the mentioned algorithm.

Sol:

a1) All linkage methods leads to the same tree shown in figure (a).

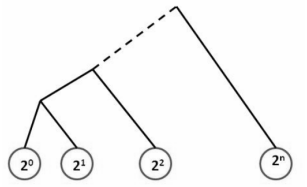


Fig. a

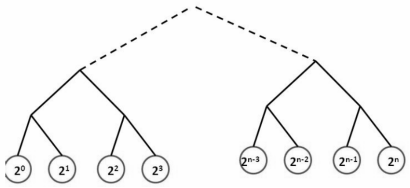


Fig. b

a2) Single link does not change. Complete link changes to the graph shown in Fig. b.

b) The clustering corresponds to single-link bottom-up clustering. The edges used to calculate the cluster distances for the single link bottom up clustering correspond to the edges of the MST (since all points must be clustered, and the cluster distance is single link and chooses the min weight edge joining together two so far unconnected clusters). Thus, the heaviest edge in the tree corresponds to the top most clusters, and so on.

6) Semi-Supervised Learning (10 points)

Consider the following figure (a) which contains labeled (class 1 filled black circles class 2 hollow circles) and unlabeled (squares) data. We would like to use two methods (re-weighting and co-training) in order to utilize the unlabeled data when training a Gaussian classifier.

Fig. a

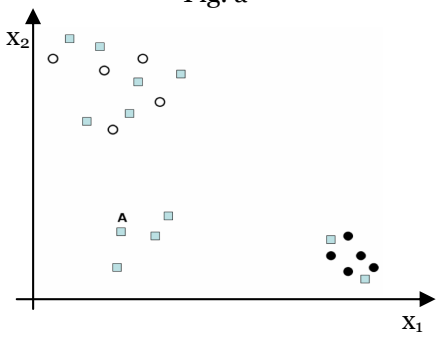
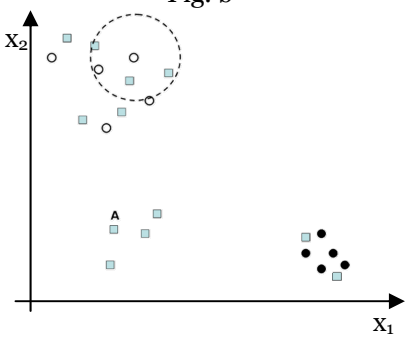


Fig. b



a) How can we use co-training in this case (what are the two classifiers)?

b) We would like to use re-weighting of unlabeled data to improve the classification performance. Re-weighting method is done by placing the dashed circle (shown in figure b) on each of the labeled data points and counting the number of unlabeled data points in that circle. Next, a Gaussian classifier is run with the new weights computed.

b1) To what class (hollow circles or full circles) would we assign the unlabeled point *A* is we were training a Gaussian classifier using only the labeled data points (with no re-weighting)?

b2) To what class (hollow circles or full circles) would we assign the unlabeled point A is we were training a classifier using the re-weighting procedure described above?

Sol:

a) Co-training partitions the feature space into two separate sets and uses these sets to construct independent classifiers. Here, the most natural way is to use one classifier (a Gaussian) for the *x* axis and the second (another Gaussian) using the *y* axis.

b1) Hollow class. Note that the hollow points are much more spread out and so the Gaussian learned for them will have a higher variance.

b2) Again, the hollow class. Re-weighting will not change the result since it will be done independently for each of the two classes, and will produce very similar class centers to the ones in (b1) above.

Good Luck!