CS4786 Midterm (Practice)

Spring 2019

NAME:	
Net ID:	
Email:	

I promise to abide by Cornell's Code of Academic Integrity.

Signature: _____

1 [15] True or False

Please identify if these statements are either True or False.

1. Of the clustering algorithms covered in class, Gaussian Mixture Models used for clustering always outperforms k-means and single link clustering.
   F. Even in practice, the structure in underlying data dictates which algorithm is better for your problem.

2. K-means algorithm always finds the clustering $(C_1, \ldots, C_K)$ that minimizes the objective:

$$\text{Minimize} \sum_{k=1}^{K} \sum_{t \in C_j} \|x_t - r_j\|^2 \qquad \text{where } r_j = \frac{1}{|C_j|} \sum_{t \in C_j} x_t \text{ is the centroid of cluster } j$$

   F. K-means algorithm is only guaranteed to not make the objective worse. So it need not exactly minimize objective. It can get stuck in local minima.

3. If in the ISOMAP algorithm, one changes the nearest neighbor graph to a weighted graph with edge weight between two vertices i, j to be

$$P_{i,j} = \frac{p_{i \mapsto j} + p_{j \mapsto i}}{2n} \qquad \text{where} \quad p_{i \mapsto j} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma^2}\right)}$$

   then one can recover t-SNE algorithm.
   F.

4. PCA can for some dataset approximately preserve pairwise distances between all pairs of points.
   T. If data is low dimensional.

5. Random Projection can always approximately preserve pairwise distances between all pairs of points.
   T JL theorem proves this.

6. Normalized Spectral clustering can only be applied to unweighted and undirected graphs
   F

7. Since PCA computes largest K eigenvectors of the covariance matrix Σ, it requires computation time of at least order d^2
   F. Using SVD, in the case when $n << d$, we can get computation time to be linear in d

8. CCA algorithm is always translation and rotation invariant. T

9. In kernel PCA, the larger the dimensionality of feature space (i.e. dimensionality of $\Phi(x_t)$'s), the longer kernel PCA will take to run. F.

10. EM algorithm can only be used for parameter estimation of mixture models. F.

11. Dimensionality reduction can be used to remove noise from high dimensional data. T.

12. Say data lies on a two dimensional circle embedded in a d-dimensional space, then performing classical PCA and reducing d dimension to 2 dimensions does note lead to loss of information T.

13. Random projection guarantees that distances between all pairs of points is preserved even when data in the higher dimension lies on a non-linear manifold. T.

14. Laplacian matrix of a graph is always positive semidefinite. T. 15. Sample covariance matrix of a set of n points in d dimensions is a $d \times d$ matrix. T.

2 [20] Multiple Choice Questions

Tick the right one(s). More than one choice can be correct.

1. Consider a linear dimensionality reduction method, where we obtain low dimensional (K dimensional) projection y_t of point x_t in d dimensions by setting $y_t^\top = x_t^\top W$ for some projection matrix W of dimension $d \times K$. Further, say W were such that its columns were orthogonal to each other and have unit norm. That is, if W_k is the k'th column of W, then $W_k \perp W_j$ for any j, k and $\|W_k\|_2 = 1$ (Eg. PCA, CCA, ...). Which of the following are true for such linear dimensionality reduction methods:

   A. Any such method can always guarantee that if $K << d$ is large enough, then all pairwise distances between points are always approximately preserved?

   B. It is possible to choose a particular W as above such that if $K < d$ is chosen large enough, then all pairwise distances between points are always approximately preserved?

   C. It is possible to choose a particular W as above such that if $K < d$ is chosen large enough, for some datasets distances are always approximately preserved?

   D. PCA algorithm is one example of linear projection as specified in the question that can, for some dataset preserve all pairwise distances between points approximately.

   A and B are false. If $\|W_k\|_2 = 1$ then in direction of W_k magnitude is exactly preserved and in all other directions, distances always shrink. So no method where $\|W_k\|_2 = 1$ can for all dataset ensure that all distances are preserved. To make it more formal, take the $n = d$ data points to be $W_1, \ldots, W_K$ for the first K points and orthonormal directions to these K directions for the remaining points. We can never preserve distances on remaining points as any W_k producted with these remaining points is going to map to 0.
   C and D are true. D is true when data lives in a K dimensional subspace.

2. Which of the following are true about the ISOMAP algorithm?

   A. It tries to preserve some distances between all pairs of points

   B. It tries to preserve Euclidean distances between all pairs of points

   C. It tries to preserve Euclidean distances between neighboring points

   D. It tries to recover the underlying low dimensional manifold on which the data (approximately) lies.

   A is true with distances being distances on the manifold. D is true, that was the goal of ISOMAP. B is certainly false. C is technically false since we preserve only shortest distances on nearest neighbor graph and not distance to every neighbor.

3. A regular graph is one where every node has exactly same number of neighbors. For a regular graph which of the following are true:

   A. Only for some regular graphs, normalized and unnormalized spectral embedding are equivalent.

   B. Normalized and unnormalized spectral embedding are always equivalent for any regular graph.

   C. Normalized and unnormalized spectral embedding are never the same for any regular graph.

   D. There are non-regular graphs for which normalized and unnormalized spectral embedding yield same results.

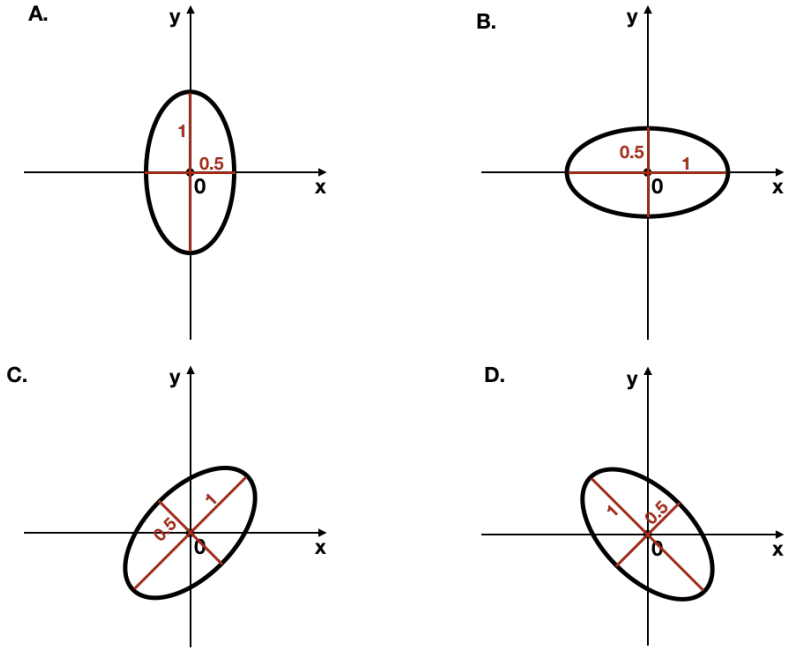
   B is true as for a regular graph, D is a a diagonal matrix with same value on every diagonal entry and so normalized or unnormalized laplacian matrices are both proportional to each other. That is, normalized Laplacian is the unnormalized Laplacian divided by number of neighbors for the nodes in the graph. D is true. Say we have a graph with two connected components and one component is k-regular and the other is k' regular, then the spectral embedding is going to be the same.

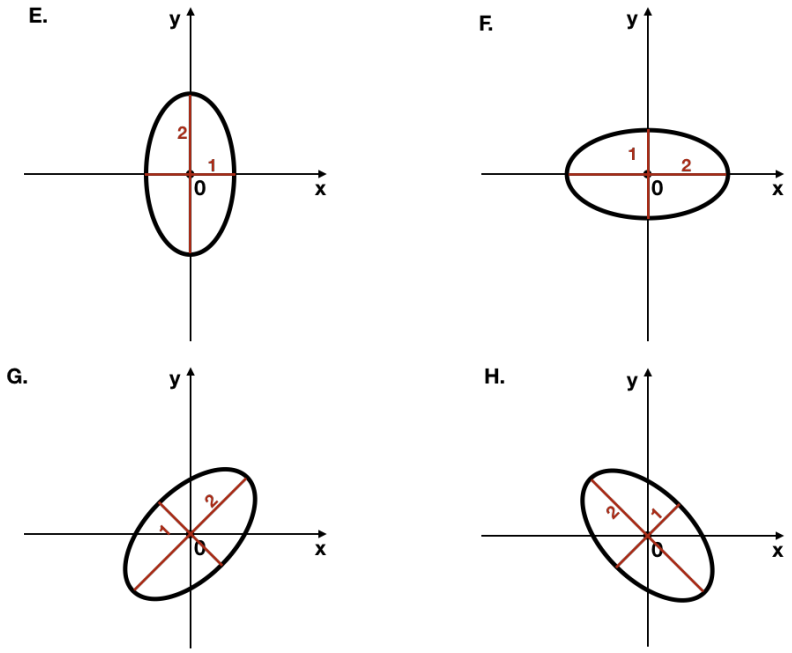
4. Probabilistic Modeling: A high level intuition is that Maximum Likelihood Estimator (MLE) is same as Maximum A Posteriori (MAP) estimator when our prior over models is that all models are equally likely. Which of the following statements are true.

   A. Whenever $P(\theta)$ is the uniform probability distribution over Θ, MLE and MAP estimators are the same.
   B. MLE can always be seen as MAP under the right prior distribution
   C. MLE can always be seen as MAP under the right prior distribution $P(\theta)$ when Θ is a finite set
   D. MLE can be seen as MAP under the right prior distribution $P(\theta)$ *only when* Θ is a finite set

   A is true.
   B is false. Specifically, if the underlying parameter can be any real number, then there is not prio such that the two are same.
   C is true. If Θ is finite, we can always define a uniform prior and so using A it is true.
   D. Not true, if $\Theta = [0, 1]$ the set of all real numbers from 0 to 1, then this set is not finite. None the less, you can define uniform distribution on this set, and using the uniform prior, MLE and MAP are same.

5. Say matrix $M = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$. Let R be the rotation matrix that rotates any given 2 dimensional vector counter-clockwise by $\pi/4$ radians. That is, given a 2 dimensional vector $\vec{z}$, $R\vec{z}$ is the same vector rotated counter clockwise by a 45 degree angle. Now say A is the matrix defined as $A = R^\top MR$. Say $\vec{z} = \begin{bmatrix} x \\ y \end{bmatrix}$ Which one of the following in the curve given by equation $\vec{z}^\top A\vec{z} = 1$ (sorry =1 was missing earlier.)?





D is true. Note that if there was not rotation, then $[1/2, 0]$ satisfies the equation $\vec{z}^{\top} A \vec{z} = 1$. and similarly $[0, 1]$ also satisfies the equation. Hence after the counter-clockwise rotation by $\pi/4$ the answer is D since $[1/2, 0]$ rotated counter clockwise and $[0, 1]$ rotated counter clockwise by 45 degrees are the principle axes of the ellipse in D.

3 [20] Rotation and Translation of Data

Say we take the $n \times d$ data matrix X and subtract from each row, some row vector μ to get new data matrix X', that is $X' = X - \mu$ this X' is said to be a translation of data X. Similarly, if we take data matrix X and rotate every row by applying a $d \times d$ rotation matrix R to get $X'' = XR$, then we say X'' is a rotation of X. Recall that for a rotation matrix $RR^\top = I_{d\times d}$. We say that a method is **translation invariant** if the result of the method is unaltered if we replace X by any translation X' by any μ. Similarly we say that a method is **rotation invariant** if we obtain same output from the method when we replace input X by any rotation X'' of it.

For each of the following algorithms, write down if they are rotation invariant and translation invariant. If they are show how/why they are rotation and/or translation invariant and if not, either via example or otherwise show that they are not rotation and/or translation invariant

1. Single Link clustering with Euclidean metric
2. Single Link clustering with ℓ_1 distance instead of euclidean.
3. K-means clustering algorithm

The ℓ_1 distance between two d dimensional vectors is x, y is given by $\text{Dist}(x, y) = \sum_{i=1}^{d} |x[i] - y[i]|$

1. Single Link clustering with Euclidean metric:

   It is translation invariant. Translating all the points doesn't change their inter-point distances. Specifically, for any $s, t \in \{1, \ldots, n\}$, $\|x_t - x_s\|_2 = \|(x_t - \mu) - (x_s - \mu)\|_2$.

   It is rotation invariant. Rotating all the points by same same rotation matrix does not change inter point distances. $\|Rx_t - Rx_s\|_2^2 = (R(x_t - x_s))^\top R(x_t - x_s) = (x_t - x_s)^\top R^\top R(x_t - x_s) = (x_t - x_s)^\top I(x_t - x_s) = \|x_t - x_s\|_2^2$

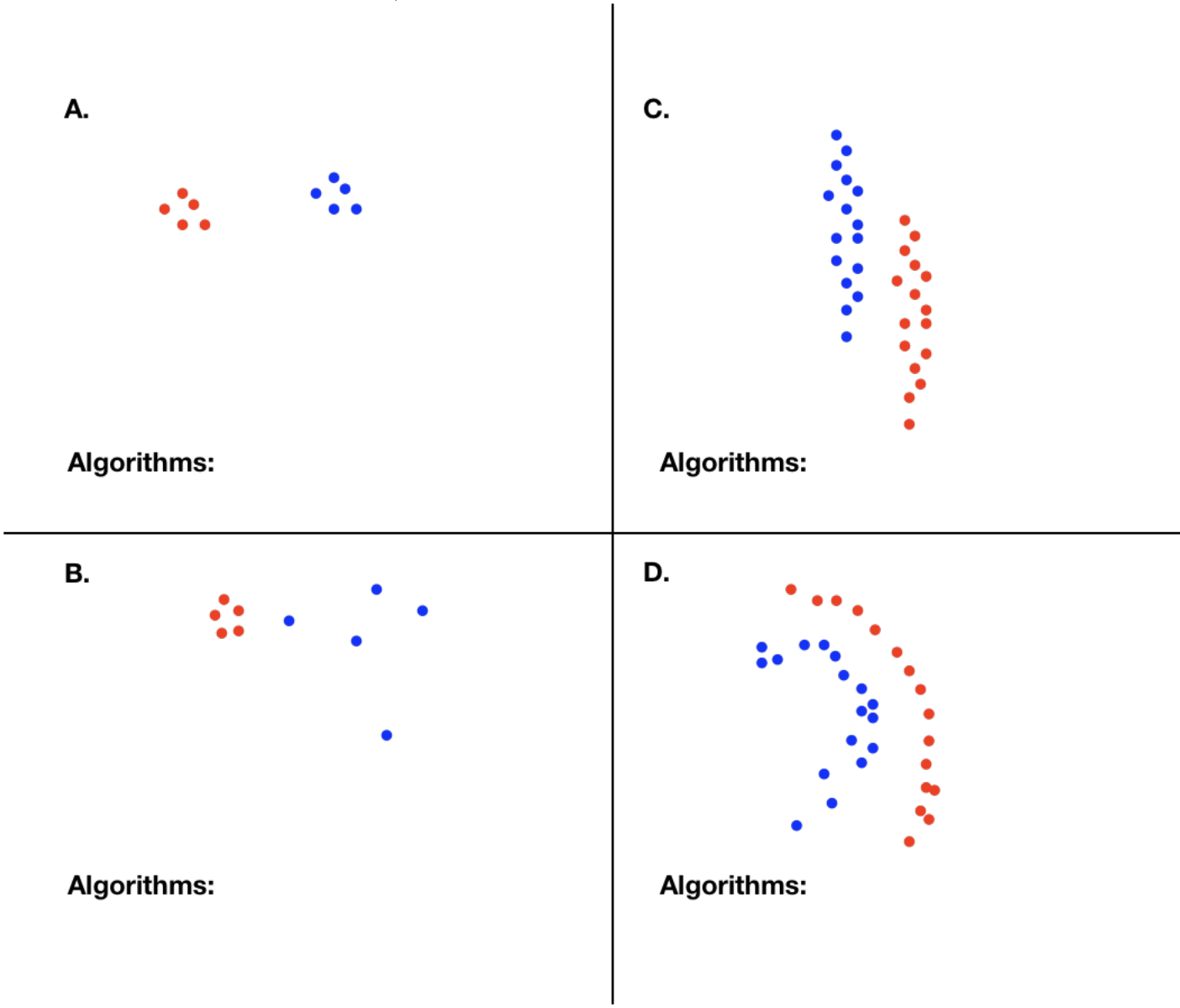
2. Single Link clustering with ℓ_1 distance:

   It is translation invariant. Translating all the points doesn't change their inter-point distances. Specifically, for any $s, t \in \{1, \ldots, n\}$, $\text{Dist}(x_t, x_s) = \sum_{i=1}^{d} |x_t[i] - x_s[i]| = \sum_{i=1}^{d} |x_t[i] - \mu[i] - (x_s[i] - \mu[i])| = d(x_t - \mu, x_s - \mu)$.

   It is NOT rotation invariant. ℓ_1 metric changes with rotation. To see this, say we have two points in 2 dimensions. One is $x_1 = [0, 0]$ and another $x_2 = [1, 0]$. Now say we rotate the points by 45 degrees so we get $[0, 0]$ and $[1/\sqrt{2}, 1/\sqrt{2}]$ Note that ℓ_1 distance before rotation is 1 and after rotation is $2/\sqrt{2} = \sqrt{2}$ between the two points.

4 [20] Which Clustering Algorithm

For each of the following clustering of (2 dimensional) points into two clusters (i.e all run with $K = 2$) (given by red points for cluster 1 and blue color for cluster two), write down which amongst: k-means, single link clustering, gaussian mixture models or none of the three could have led to the cluster assignments shown. (more than one answer could be right, put down all the possible ones,).



A: K-means, single link, gaussian mixture models
B. gaussian mixture models
C. single link, gaussian mixture models
D. Single link

5 [20] The RBF Kernel?

Lets consider points in one dimension. The Radial Basis Functions (RBF) kernel with parameter σ is given by the kernel function

$$k(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$$

show that this is indeed a valid kernel function. Specifically do this by writing down feature mapping $\Phi(x)$ into some high dimensional vector space and show that for any real numbers x, y,

$$k(x,y) = \langle \Phi(x), \Phi(y) \rangle$$

Hint: the only thing you will need is the expansion of the exponential which you can recall is given by: $e^a = 1 + a + \frac{a^2}{2!} + \frac{a^3}{3!} + \ldots$ for any real number a.

Answer: Notice that:

$$k(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) = \exp\left(-\frac{x^2}{2\sigma^2} - \frac{y^2}{2\sigma^2} + \frac{2xy}{2\sigma^2}\right)$$
$$= e^{-\frac{x^2}{2\sigma^2}} \times e^{-\frac{y^2}{2\sigma^2}} \times e^{\frac{2xy}{2\sigma^2}}$$

Now using the fact that $e^a = 1 + a + \frac{a^2}{2!} + \frac{a^3}{3!} + \ldots$ we have,

$$k(x,y) = e^{-\frac{x^2}{2\sigma^2}} \times e^{-\frac{y^2}{2\sigma^2}} \times \left(1 + \frac{xy}{\sigma^2} + \frac{1}{2!}\frac{x^2y^2}{\sigma^4} + \frac{1}{3!}\frac{x^3y^3}{\sigma^6} + \ldots\right)$$

Hence define

$$\Phi(x) = e^{-\frac{x^2}{2\sigma^2}} \times \begin{bmatrix} 1 \\ \frac{x}{\sqrt{1!}\sigma} \\ \frac{x^2}{\sqrt{2!}\sigma^2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{x^i}{\sqrt{i!}\sigma^i} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Notice that

$$\Phi(x)^\top \Phi(y) = e^{-\frac{x^2}{2\sigma^2}} \times e^{-\frac{y^2}{2\sigma^2}} \times \left(1 + \frac{xy}{\sigma^2} + \frac{1}{2!}\frac{x^2y^2}{\sigma^4} + \frac{1}{3!}\frac{x^3y^3}{\sigma^6} + \ldots\right)$$

and hence we can conclude that k is a valid kernel with the above infinite dimensional feature mapping.

6 [20] EM Algorithm

A casino has K regular gamblers. On each day t, one of the K gamblers comes in and plays m_t rounds of blackjack game for that day and wins w_t of those m_t rounds. The casino agrees to share with you just the data of how many rounds of black jack were played on each day and how many of those rounds were won by the gambler and the fact that there are K gamblers playing the game. You however don't know which of the K gamblers played on which day. You decide to use probabilistic model, especially mixture model to model this data. Specifically, for each player k, you model the probability that she wins on any given round by the parameter p_k between 0 and 1. That is, if gambler k plays a round, the probability that she wins the round is p_k. Hence, on day t, if the k'th gambler had played m_t rounds, then the probability that she won w_t of those m_t rounds is given by the binomial distribution by $\binom{m_t}{w_t} p_k^{w_t}(1-p_k)^{m_t-w_t}$. Now with this model, you shall use a mixture of K binomials with parameters $p_1, \ldots, p_K$ to model the data for n days given by $(m_1, w_1), \ldots, (m_n, w_n)$. That is, the generative story is that on day t, we first pick one gambler out of the K at random according to the distribution π as $c_t \sim \pi$. Next, the gambler for day t plays m_t rounds and given the gambler is c_t, the number of wins w_t out of the m_t rounds is given by the binomial distribution, $\binom{m_t}{w_t} p_{c_t}^{w_t}(1-p_{c_t})^{m_t-w_t}$.

Derive the EM algorithm for this problem. Specifically:

1. Write down the E-step update for Q's. That is write down what $Q_t^{(i)}[k]$ is for any given iteration i (in terms of parameters from previous iteration).
2. For any, mixture modes, as we showed in class, the M-step for π on iteration i is given by $\pi^{(i)}[k] = \frac{\sum_{t=1}^n Q_t^{(i)}[k]}{n}$. Derive the M-step update for $p_1^{(i)}, \ldots, p_K^{(i)}$ the K model parameters on iteration i, in terms of data and $Q_t^{(i)}$'s. First write down the maximization problem for the M-step and then solve for $p_1^{(i)}, \ldots, p_K^{(i)}$ showing that they are the maxima for the optimization problem.

E-step:

We set Q_t as

$$\begin{aligned} Q_t^{(i)}[k] &= P(c_t = k | x_t; \theta^{(i-1)}) \propto P(x_t | c_t = k; \theta^{i-1}) \times \pi_k^{(i-1)} \\ &\propto \binom{m_t}{w_t} p_k^{w_t}(1-p_k)^{m_t-w_t} \times \pi_k^{(i-1)} \\ &= \frac{\binom{m_t}{w_t} p_k^{w_t}(1-p_k)^{m_t-w_t} \times \pi_k^{(i-1)}}{\sum_{j=1}^K \binom{m_t}{w_t} p_j^{w_t}(1-p_j)^{m_t-w_t} \times \pi_j^{(i-1)}} \end{aligned}$$

M-step:

We already have that for any mixture model, $\pi^{(i)}[k] = \frac{\sum_{t=1}^n Q_t^{(i)}[k]}{n}$. Now lets optimize for p_k's. We want to maximize the following w.r.t. $p_1, \ldots, p_K \in [0,1]$:

$$f(p_1, \ldots, p_k) = \sum_{t=1}^n \sum_{j=1}^K Q_t^{(i)}[j] \log \left(\binom{m_t}{w_t} p_j^{w_t}(1-p_j)^{m_t-w_t} \right)$$

To find $p_k^{(i)}$, we find p_k such that $\frac{\partial f}{\partial p_k} = 0$ or in other words, taking derivative,

$$\begin{aligned} 0 = \frac{\partial f}{\partial p_k} &= \frac{\partial}{\partial p_k} \sum_{t=1}^n Q_t^{(i)}[k] \log \left(\binom{m_t}{w_t} p_k^{w_t}(1-p_k)^{m_t-w_t} \right) = \frac{\partial}{\partial p_k} \sum_{t=1}^n Q_t^{(i)}[k] \left(\log \left(\binom{m_t}{w_t} \right) + w_t \log(p_k) + (m_t - w_t) \log(1-p_k) \right) \\ &= \sum_{t=1}^n Q_t^{(i)}[k] \left(\frac{w_t}{p_k} - \frac{m_t - w_t}{1-p_k} \right) \end{aligned}$$

Hence we have that

$$\frac{\sum_{t=1}^n Q_t^{(i)}[k] w_t}{p_k^{(i)}} = \frac{\sum_{t=1}^n Q_t^{(i)}[k] (m_t - w_t)}{1 - p_k^{(i)}}$$

Hence we conclude that:

$$p_k^{(i)} = \frac{\sum_{t=1}^{n} Q_t^{(i)}[k] w_t}{\sum_{t=1}^{n} Q_t^{(i)}[k] m_t}$$

Thus for every k we set $p_k^{(i)}$ as above in the M-step.

Question	Points Scored	Max Points
True/False		15
Multiple Choice		10
Rotation and Translation		20
Choose Clustering Algo.		20
RBF kernel		20
EM for mixture Model		20
TOTAL		105

How did it go? Anything you want to share with us (exam related or not)?