

CSE 446 Machine Learning, Spring 2016

Homework 4

Due: 2016/6/6

1 EM for Gaussian Mixture Model [40 points]

(Extended version of: Murphy Exercise 11.7) In this question we consider clustering 1D data with a mixture of 2 Gaussians using the EM algorithm. You are given the 1-D data points $x = [1 \ 10 \ 20]$.

M step

Suppose the output of the E step is the following matrix:

$$R = \begin{pmatrix} 1 & 0 \\ 0.4 & 0.6 \\ 0 & 1 \end{pmatrix}$$

where entry $R_{i,c}$ is the probability of observation x_i belonging to cluster c (the responsibility of cluster c for data point i). You just have to compute the M step. You may state the equations for maximum likelihood estimates of these quantities (which you should know) without proof; you just have to apply the equations to this data set. You may leave your answer in fractional form. Show your work.

1. [4 points] Write down the likelihood function you are trying to optimize.

2. [8 points] After performing the M step for the mixing weights π_1, π_2 , what are the new values?

3. [8 points] After performing the M step for the means μ_1 and μ_2 , what are the new values?

4. [8 points] After performing the M step for the standard deviations σ_1 and σ_2 , what are the new values?

E step

Now suppose the output of the M step is the answer to the previous section. You will compute the subsequent E step.

1. [4 points] Write down the formula for the probability of observation x_i belonging to cluster c .

2. [8 points] After performing the E step, what is the new value of R ?

2 Expectation Maximization [30 points]

Suppose there are the following four possible outcomes possible when you conduct an experiment: “compound W is present” , “compound X is present”, “compound Y is present”, “compound Z is present”. Suppose we know the probabilities of these events take the form:

- $P(W) = \frac{1}{2}$
- $P(X) = \mu$
- $P(Y) = 2\mu$
- $P(Z) = \frac{1}{2} - 3\mu$

(where μ is unknown). Note that this is valid probability distribution as it sums to 1. We then conduct multiple experiments (done independently) and observe the outcomes of our experiments. Let n_w , n_x , n_y , and n_z be the number of times we observed W , X , Y , and Z , respectively, in our experiments.

1. [10 points] What is the Maximum Likelihood Estimator for μ , given our data?

Now, instead, suppose that we observe some new data where we are not able to observe when W or X occur, but, instead, we only observe if the experiment resulted in either W or X . So we don't know n_w or n_x , but we know the sum $n_{w,x} = n_w + n_x$. As before, we still observe n_y and n_z as before. We now intend to use the Expectation Maximization algorithm to find an estimate of μ .

2. [10 points] **Expectation step:** Given $\hat{\mu}$, a current estimate of μ , what are the expected values of n_w and n_x in terms of the $\hat{\mu}$ and the observed outcomes?

3. [10 points] **Maximization step:** Given \hat{n}_w and \hat{n}_x , the expected values of n_w and n_x , what is the MLE of μ ? Simply use the expression from part 1 with \hat{n}_w and \hat{n}_x , instead of n_w and n_x .

3 Dimensionality Reduction [30 points]

PCA can be hard to visualize in dimensions higher than 3, but the intuition actually translates very well in graphics modeling and image processing. In this question we will walk through dimensionality reduction and reconstruction using the example of "eigenbodies".

We will define an eigenbody (which is related to eigenfaces, eigennumbers, eigencats, eigen-whatever-noun-you-choose) as the decomposition of a 3D model of a collection of human bodies into its principal components. Each data point represents one body, which can be written as a d -dimensional vector of sequential x, y, z coordinates for each 3D vertex.


Note that this representation only makes sense if each index in the vector corresponds to the same vertex position in every body model. (i.e. position $i, i+1, i+2$ always corresponds to x, y, z for the same point on the left eye of every subject)

- (a) [2 points] If each model has v vertices and each vertex is a point (x, y, z) in 3D space, what is the dimensionality d of the data set?

- (b) [4 points] If matrix \mathbf{B} is the data matrix representing a set of n examples each consisting of v vertices, state the dimensions of \mathbf{B} in terms of n and v and write the Singular Value Decompositions of \mathbf{B} for the case when $n = v$. State the dimensionality of each matrix composing the SVD. What is the maximum value of k , the number of principal components you are able to solve for with this data set?

NOTE: You may assume \mathbf{B} is zero-mean.

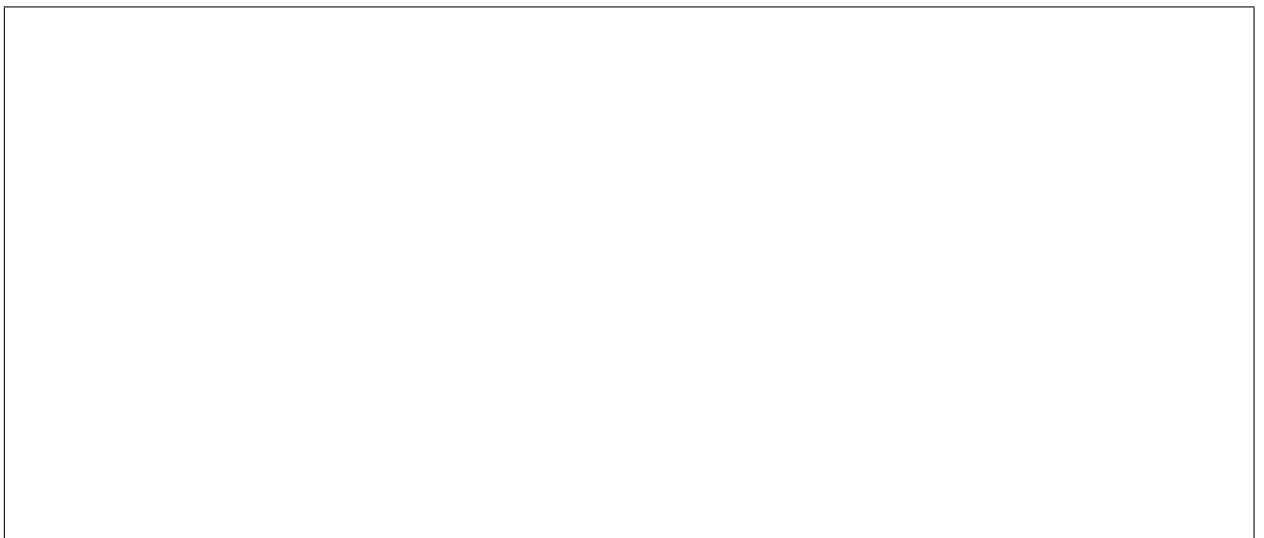
- (c) [5 points] In general, the k principal components of matrix \mathbf{B} are the k eigenvectors corresponding to the k largest eigenvalues of the scaled covariance matrix $\frac{1}{N}\mathbf{B}^T\mathbf{B}$. Prove how the SVD of \mathbf{B} also solves for the eigenvectors of the covariance matrix. Do this by mathematically showing the equivalence of a component of the SVD to the eigenvectors of $\frac{1}{N}\mathbf{B}^T\mathbf{B}$.



- (d) [4 points] Reconstruction of a point in the original dimensionality can be achieved by adding a linear combination of the principal components to the mean of the training set.

$$\mathbf{y} = \bar{\mathbf{B}} + \mathbf{R}\mathbf{w}$$

Where \mathbf{R} is a matrix of your principal component vectors. What is the dimensionality of \mathbf{R} and what is the interpretation of its rows/columns? For the problem of eigenbodies, list three examples of possible qualities the first k principal components may represent. (For example, a principal component of an eigenface may correspond to jaw width)



- (e) [5 points] Suppose you were given a human model represented by a d -dimensional vector \mathbf{y} . It is possible to solve for a weight vector \mathbf{w} that represents the best approximation of \mathbf{y} composed of a linear combination of $k < d$ orthonormal basis vectors. Formulate this as a linear regression problem by writing the objective function as an argmin on \mathbf{w} . Briefly explain the meaning of each variable in your objective function.

- (f) [10 points] In general, the objective of PCA is to minimize the reconstruction error, defined by the difference between the reconstruction using k basis vectors (like you created in the previous section) and the reconstruction possible with all d basis vectors. For a given training matrix \mathbf{Y} of d -dimensional vectors \mathbf{y}_i , minimize the objective:

$$e_{recon} = \sum_{i=1}^N (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2$$

Where

$$\mathbf{y}_i = \bar{\mathbf{Y}} + \sum_{j=1}^d z_j^i \mathbf{r}_j$$

and

$$\hat{\mathbf{y}}_i = \bar{\mathbf{Y}} + \sum_{j=1}^k z_j^i \mathbf{r}_j$$

Define z_j^i as the zero-mean projection of the i -th data point onto the j -th principal component

$$z_j^i = (\mathbf{y}_i - \bar{\mathbf{Y}}) \cdot \mathbf{r}_j$$

Show that this reconstruction error is equivalent to $N \sum_{j=k+1}^d \mathbf{r}_j^T \Sigma \mathbf{r}_j$

where Σ is the zero-mean covariance matrix $\Sigma = \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_i - \bar{\mathbf{Y}})(\mathbf{y}_i - \bar{\mathbf{Y}})^T$

and explain how this shows that the first k principal components of \mathbf{Y} are the k eigenvectors corresponding to the highest eigenvalues of Σ .

4 Programing Question (K-means) [50 points]

4.1 The Data

There are two files with the data on the course website. The first `digit.txt` contains the 1000 observations of 157 pixels (a subset of the original 785) from images containing handwritten digits. The second file `labels.txt` contains the true digit label (either 1, 3, 5, or 7). You can read both data files in with

```
import numpy as np
X = np.genfromtxt('digit.txt')
Y = np.genfromtxt('labels.txt', dtype=int)
```

4.2 The algorithm

Here is a quick review of how K-means algorithm works.

- i. Select k starting centers that are points from your data set. You should be able to select these centers randomly or have them given as a parameter.
- ii. Assign each data point to the cluster associated with the nearest of the k center points.
- iii. Re-calculate the centers as the mean vector of each cluster from (2).
- iv. Repeat steps (2) and (3) until convergence or iteration limit.

Define convergence as no change in label assignment from one step to another **or** you have iterated 20 times (whichever comes first). Please count your iterations as follows: after 20 iterations, you should have assigned the points 20 times.

For Step 1, please make sure that you deep-copy your data points when you select them as starting centers. Otherwise, when you move centers, you will end up moving the data points as well.

Please be reminded that you are NOT allowed to use existing machine learning libraries such as scikitlearn.

4.3 Within group sum of squares

The goal of clustering can be thought of as minimizing the variation within groups and consequently maximizing the variation between groups. A good model has low sum of squares within each group.

We define sum of squares in the traditional way. Let C_k be the k th cluster and let μ_k be the empirical mean of the observations x_i in cluster C_k . Then the within group sum of squares for cluster C_k is defined as:

$$SS(k) = \sum_{i \in C_k} |x_i - \mu_{C_k}|^2$$

Please note that the term $|x_i - \mu_{C_k}|$ is the euclidean distance between x_i and μ_{C_k} , and therefore should be calculated as $|x_i - \mu_{C_k}| = \sqrt{\sum_{j=1}^d (x_{ij} - \mu_{C_{kj}})^2}$, where d is the number of dimensions. Please note that that term is squared in $SS(k)$. If there are K clusters total then the “sum of within group sum of squares” is just the sum of all K of these individual $SS(k)$ terms.

4.4 Mistake Rate

Given that we know the actual assignment labels for each data point we can attempt to analyze how well the clustering recovered this. For cluster C_k let its assignment be whatever the majority vote is for that cluster. If there is a tie, just choose the digit that is smaller numerically as the majority vote.

For example if for one cluster we had 270 observations labeled **one**, 50 labeled **three**, 9 labeled **five**, and 0 labeled **seven** then that cluster will be assigned value **one** and had $50 + 9 + 0 = 59$ mistakes. If we add up the total number of “mistakes” for each cluster and divide by the total number of observations, we will get our total mistake rate, between 0 and 1.

4.5 Questions

When you have implemented the algorithm please report the following:

- i. [10pts] The values of sum of within group sum of squares and mistake rates for $k = 2$, $k = 4$ and $k = 6$. Please start your centers with the first k points in the dataset. So, if $k = 2$, your initial centroids will be the first two lines in **digit.txt**.

- ii. [10pts] The number of iterations that k-means ran for $k = 6$, starting the centers as in the previous item. Make sure you count the iterations correctly. If you start with iteration $i = 0$ and at $i = 3$ the cluster assignments don't change, the number of iterations was 4, as you had to do step 2 four times to figure this out.

- iii. [15pts] A plot of the sum of within group sum of squares versus k for $k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$. Please start your centers randomly (choose k points from the dataset at random). What trend do you expect to have and why?

- iv. [15pts] A plot of total mistake rate versus k for $k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$. Please start your centers randomly (choose k points from the dataset at random). What trend do you expect to have and why?



For iii. and iv., you should generate plots a few times until you get the trend you expect in case you get unlucky with starting centroids. Please submit your code to Catalyst.