MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.036—Introduction to Machine Learning
Spring 2015

Project 3: Which movies do I like? Issued: Tues., 4/15 Due: Fri., 4/25 at 9am

Project Submission: Please submit two files—a *single* PDF file containing all your answers, code, and graphs, and a *second* .zip file containing all the code you wrote for this project, to the Stellar web site by 9am, April 25th.

Introduction

Your task is to build a mixture model for collaborative filtering. You are given a data matrix containing movie ratings made by users; we have sampled this matrix from the Netflix database. Not all movies have been rated by all users, and the goal of this project is to use mixture modeling to predict the missing ratings. You will explore this task by using the Expectation Maximization (EM) algorithm that uses the hidden structure across different users, and with the help of this hidden structure, you will be able to predict the missing ratings. In other words, we have a partially observed rating matrix, and we will use the EM algorithm to complete (fill in) the missing entries.

1. **Part 1** Warm up example.

   For this part of the project you will compare clustering obtained via K-means to the (soft) clustering induced by EM.

   (a) Use the toy data set (`blah.csv`) and the K-means code (`kmeans.py`) provided to plot different clusters for cluster sizes $K = [5, 10, 15]$. Notice that when using K-means, each data point is fully assigned to a single cluster, that is, each point can have only one cluster label.

   (b) Let x be a data point. Recall the mixture model presented in class:

   $$P(x|\theta) = \sum_{j=1}^{K} p(j|\theta)p(x|j,\theta),$$

   where θ denotes the parameters of the model. A data point may be generated by first choosing a cluster, and then choosing a data point x according to that cluster (What does this mean?). Explain how once you have cluster assignments given by K-means would the data generation process based on a K-means model would differ from data generation via a mixture model. [Hint: K-means partitions the space into clusters, whereas a mixture model allows weighted memberships across different clusters].

   (c) Consider a mixture model that uses a Gaussian as the conditional distribution given the hidden label. That is, $p(x|j,\theta) = N(x|\mu^{(j)}, \sigma_j^2 I)$, where $\mu^{(j)}$ and $\sigma_j^2 I$ are the unknown parameters for mixture component j (label j).

   The goal of the EM algorithm is to estimate these unknown parameters by making use of observed data, say $x^{(1)}, \ldots, x^{(N)}$. Starting with some initial guess at the unknown parameters, the E-Step keeps the model fixed (i.e., for each component j, its parameters $\mu^{(j)}$ and the σ_j^2 are held fixed; this idea is similar to K-means holding the centroids fixed in one of its steps), and computes the soft-assignments for each data point. Thus, for each data point $x^{(t)}$ $(1 \leq t \leq N)$ the E-step computes the posterior probabilities $p(j|x^{(t)})$. The M-step takes these soft-assignments

as fixed, and computes the maximum-likelihood estimates of the parameters $\mu^{(j)}$ and $\sigma_j^2 I$ for each component j ($1 \leq j \leq K$) (notice the analogy to K-means, which, given assignment of data points to their clusters, computes the centroids of each cluster).

Task: Implement the EM algorithm using a Gaussian mixture model (as recalled above), and run it in the toy data set with a random initialization: WE have to provide the initialization so that the students can checkpoint their results: e.g., after running 100 iterations of EM, using our initialization, on this data, the log-likelihood they obtain should match the one we provide.

The question below uses the word clustering, when really the students are only running K-means induced clustering. Perhaps good to clarify that hard clustering is meant.

(d) Your next task is to understand how K-means clustering differs from the soft-clustering induced by learning a mixture model. Using the parameters estimated in part (c) (obtained after 100 iterations of EM), **plot** the clusters (**visualize** the soft-clusters?) by drawing 3 contours for each of the 2D? Gaussians. **Explain** why choosing contour curves that are evenly spaced is not a good visualization of these Gaussians. Further, **explain** why is having some of the area covered by these contour curves cover intersect a good visualization for mixture models This is a vague task.... Also **explain** why in the mixture model it does not make sense to deterministically assign any data point to a Gaussian and hence highlight its difference from K-means clustering. Do points that are very far away from cluster centroids still have a chance to be assigned to any cluster in EM? What about in K-means?

(e) Now we will try to choose the number of mixture components (K) that EM should learn. **Explain** why choosing a value of K that achieves the highest log-likelihood might not be the best criterion for selecting K.

(f) One way to avoid the issues addressed in part (e) is to penalize a high number of parameters. **Explain** how the Bayesian Information Criterion (BIC) addresses the issue brought up in part (e) and why it might be a better function for choosing K.

(g) Implement the Bayesian Information Criterion (BIC) for selecting the number of mixture components. Choose the best value of K from the choices $\{5, 10, 15, 20, 30\}$.

I have highlighted the deliverables for the questions using **bold**; please ensure that this is consistent, or at least for each question we clearly indicate what they have to deliver.

2. **Part 2** EM for predicting movie ratings via *matrix completion*.

In this part of the project we will use the EM algorithm for matrix completion. Let X denote the $N \times D$ data matrix. The rows of this matrix correspond to users and the columns correspond to movies. A single entry $x_j^{(i)}$ (Better to write: x_{ij}) indicates the rating person user i gave to movie j, and this rating is a single number that lies in the set $\{1, 2, 3, 4, 5\}$.

In a realistic setting, most of the entries of X are missing, because a user will not have watched most of the movies so he/she would have not rated the unwatched movies. Thus, we use the set C_u to denote the collection of movies (column indices) that user u has rated. Also, let H_u denote the set of movie indices that a user has not watched. Notice that $C_u \cup H_u = \{1, \ldots, D\}$. To denote a subset of the movies a particular user has watched we write $x_{C_u}^{(u)}$, which is a vector with $|C_u|$ entries. Similarly $x_{H_u}^{(u)}$ denotes the vector of hidden / unknown entries (the ratings we wish to estimate).

For example, if user 1 has the ratings vector $x^{(1)} = (5, 4, ?, ?, 2)$, then $C_1 = \{1, 2, 5\}$ and $H_1 = \{3, 4\}$ and $x_{C_1}^{(1)} = (5, 4, 2)$.

Our goal is to use a mixture model to generate the missing entries of the matrix X (thus the name "matrix completion"). We will estimate the parameters of the mixture model using EM.

(a) The mixture model from Part 1 assigns the probability density $P(x^{(u)}|\theta) = \sum_{j=1}^{K} p_j N(x^{(u)}; \mu^{(j)}, \sigma_j^2 I)$ to the vector $x^{(u)}$. However, since we have missing entries, i.e., not all entries of $x^{(u)}$ are known, we will just use only the observed data and compute $P(x_{C_u}^{(u)}|\theta)$. **Argue** that the correct expression for $P(x_{C_u}^{(u)}|\theta)$ is

$$P(x_{C_u}^{(u)}|\theta) = \sum_{j=1}^{K} p_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I).$$

[Hint: note that the covariance matrix is a multiple of the identity].

(b) Using the mixture density from part (a) for a partial data point $x_{C_u}^{(u)}$, we are ready to write the log-likelihood and maximize to derive the M-step of the EM algorithm.

The following text in blue color is not needed and should be deleted Recall from the notes that the complete log-likelihood using the hard-assignment:

$$\sum_{u=1}^{n} \left[\sum_{j=1}^{K} \delta(j|u) \log(p_j N(x_{C_u}^{(u)}|\mu_{C_u}^{(j)}, \sigma_j^2 I) \right]$$

Notice that we used the probability we derived in part (a). In this case the purpose of the indicator function is to select the probability of data points that actually generated them. In this case it would be like knowing the true assignments and hence we choose the exact (log) mixture component that generated it. However, the whole point is to learn such a hidden assignment without knowing it. Hence, to this goal we will be more conservative and instead replace the hard-assignment with soft assignments.

To that end, we will maximize the following log-likelihood:

$$\sum_{u=1}^{N} \left[\sum_{j=1}^{K} p(j|u) \log\big(p_j N(x_{C_u}^{(u)}|\mu_{C_u}^{(j)}, \sigma_j^2 I)\big) \right]$$

where the posterior probability $p(j|u)$ can be interpreted as the soft-assignment of the data point $x_{C_u}^{(u)}$ to the mixture component j. To maximize the above log-likelihood, we keep the probabilities $p(j|u)$ (the soft-assignments) fixed, and we maximize over the model parameters. **Derive** the M-step that results after maximizing the above likelihood with respect to the variables $\mu_{C_u}^{(j)}$, σ_j^2 and p_j (remember $\sum_{j=1}^{K} p_j = 1$ is a constraint). Are we really asking them to do this derivation? This requires using Lagrangians etc., are the students supposed to be able to do that?

(c) In the E-Step of the EM algorithm, one uses the updated model parameters to re-estimate the soft-assignments $p(j|u)$. **Write** down the formula for the E-Step that shows how to update $p(j|u)$ [Hint: the parameters $\mu_{C_u}^{(j)}$, σ_j^2, p_j are to be held fixed in this step].

(d) Next, **implement** the EM algorithm for running on the partially observed model. Use the E- and M- steps you derived in parts (b) and (c) above.

(e) Finally, **run** your EM algorithm using the initialization (`file.csv?, read_init.py`) to learn the model parameters on the incomplete data set (`netflix.csv`). And once you have learned the parameters using EM, you have at your hands a *generative model*, which you should apply to complete the matrix (generate the missing entries).

(f) Report the squared distance between entries you complete and the true entries. This requires knowing the true entries? Fix this question / make more precise.