MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.036—Introduction to Machine Learning
Spring 2014

Project 3: Pun With Words   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 mixture model for collaborative filtering. In this project you will be given a fraction of the data matrix containing users and movie ratings from the netflix database. We have processed a subset of Netflix's data such that you get access to all the ratings for a subset of movies. The goal of this project will be to use the hidden structure in the different types of users that exist using the EM algorithm, then with the knowledge and this hidden structure we hope to be able to complete a partially observed rating matrix as an end goal.

1. **Part 1** Toy Data Set

   For this part of the project you will explore the connections of the results that clustering gives versus the clusters that EM algorithm gives.

   (a) Use the toy data set and the k-means code we provided to plot different clusters for cluster sizes $k = [5, 10, 15]$. Notice that each data point is fully assigned to a single cluster. Meaning, that each point can only have one underlying hidden label for this model.

   (b) Recall the mixture model that we was presented in class $P(x^{(t)}|\theta) = \sum_{j=1}^{K} p(j|\theta)p(x^{(t)}|j,\theta)$. A data point may be generated by first choosing a cluster and then choosing a data point x according to that cluster. Conceptually, explain how once you have cluster assignments given by k-means, how the data generation process for your model for k-means would differ from one from a mixture model. [Hint: k-means divides the plane with Voronoi diagrams but a mixture model allows weighted averages across cluster types].

   (c) Consider a mixture model that uses a Gaussian as the conditional distribution given the hidden label i.e. $p(x^{(t)}|j,\theta) = N(x^{(t)}|\mu^{(j)}, \sigma_j^2 I)$. We will try to understand the EM algorithm on this model. Recall that the EM algorithm aims to learn these parameters by sequential minimization (similar to how we presented clustering and boosting). In the E-step we will have the model fixed (i.e. hold the $\mu^{(j)}$ and the σ_j^2 fixed, similar to holding the centroids fix) and compute the soft count assignments for each data point i.e. the posterior probability $p(j|x^{(t)}) = p(j|t)$. The M-step will receive these soft-counts and treat them as fixed. Now that we have these soft counts, we will compute the maximum-likelihood estimates for our model but with the hard-counts replaced by the soft count.

   Now that you've recalled the EM algorithm, please implement it and run it in the toy data set with a random initialization.

   (d) In this section we will try to understand how clustering is different with the mixture model and the learned parameters from the EM algorithm. With the learned parameters from part (c), plot the clusters by providing 3 contour curves for each of the gaussians learned. Explain why choosing contour curves that are evenly spaced out is not a good visualization of these gaussians.

Furthermore explain why having some of the area that these contour curves cover intersect, a good visualization for the mixture models. Moreover, 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 clustering. Do points that are very far away from clusters still have a chance to be assigned to any cluster in EM? What about in clustering?

(e) Now we will try to choose a good number of mixing components for the EM algorithm to learn the parameters. Explain why choosing a value of k that achieves the highest log-likelihood might not be the best criterion for selecting the number of mixing components.

(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 any of the issue brought up in part (e) and why it might be a better function for choosing the number of mixture components.

(g) Implement the Bayesian Information Criterion (BIC) for selecting the number of mixture components. Choose the best value of k in the range $[5, 10, 15, 20, 30]$.

2. **Part 2** Matrix Completion

For this part of the project we will use the EM algorithm for matrix completion. We will use X to denote the data matrix. It will be an $n \times d$ matrix, meaning that there will be n rows and d columns. The rows of the data matrix indicate users and the columns indicate movies. A single entry $x^{(i)}_j$ of the matrix will indicate the rating person i gave to movie j and the rating will be in the set from 1 to 5, i.e. $x^{(i)}_j \in \{1, ..., 5\}$.

However, in a real life setting, most of the entries will be missing i.e. a user will not have watched most of the movies so he would have not rated any of those movies. To indicate that, we will use the set C_u which is the collection of movie indices that user u has rated. Also, denote H_u as the set of movie indices that a user has not watch. Notice that $C_u \cup H_u = \{1, ..., d\}$. To denote a subset of the movies a particular user has watch we will use the notation $x^{(u)}_{C_u}$ which is a vector with only $|C_u|$ entries. Similarly $x^{(u)}_{H_u}$ will be the vector that contains the hidden entries.

For example, if we have the user vector $x^{(1)} =< 5, 4, ?, ?, 2 >$, then its complete entries are $C_1 = \{1, 2, 5\}$ and $H_1 = \{3, 4\}$. Also, its complete vector would be $x^{(1)}_{C_1} =< 5, 4, 2 >$.

In this case, we will have a different mixture model and using it, we will derive the EM algorithm for the missing entires matrix completion problem.

(a) Recall the mixture model from part 1: $P(x^{(u)}|\theta) = \sum_{j=1}^{K} p_j N(x^{(u)}; \mu^{(j)}, \sigma_j^2 I)$. However, in the missing entries case, we would actually be interested in finding $P(x^{(u)}_{C_u}|\theta)$ (i.e. the vector without the missing entries since those are not observed). Explain (or prove) why the correct expression for $P(x^{(u)}_{C_u}|\theta)$ is$P(x^{(u)}_{C_u}|\theta) = \sum_{j=1}^{K} p_j N(x^{(u)}_{C_u}; \mu^{(j)}_{C_u}, \sigma_j^2 I)$ [hint: notice that the covariance matrix has the identity matrix].

(b) Recall from the notes the complete log-likelihood using the hard-assignment:

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

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. i.e. instead we will try to maximize the following function:

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