ELEN 4903: Machine Learning

Columbia University, Spring 2016

Homework 4: Due April 15, 2016 by 11:59pm

Please read these instructions to ensure you receive full credit on your homework. Submit the written portion of your homework as a *single* PDF file through Courseworks (less than 5MB). In addition to your PDF write-up, submit all code written by you in their original extensions through Courseworks (e.g., .m, .r, .py, etc.). Any coding language is acceptable. Do not wrap your files in .rar, .zip, .tar and do not submit your write-up in .doc or other file type. Your grade will be based on the contents of *one* PDF file and the original source code. Additional files will be ignored. We will not run your code, so everything you are asked to show should be put in the PDF file. Show all work for full credit.

Late submission policy: Late homeworks will have 0.1% deducted from the final grade for each minute late. *Your homework submission time will be based on the time of your **last** submission to Courseworks. I will not revert to an earlier submission!* Therefore, do not re-submit after midnight on the due date unless you are confident the new submission is significantly better to overcompensate for the points lost. Submission time is non-negotiable and will be based on the time you submitted your last file to Courseworks. The number of points deducted will be rounded to the nearest integer.

Problem 1 (K-means) – 35 points

Implement the K-means algorithm discussed in class. Generate 500 observations from a mixture of three Gaussians on $\mathbb{R}^2$ with mixing weights $\pi = [0.2, 0.5, 0.3]$ and means μ and covariances Σ,

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mu_2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mu_3 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

1. For $K = 2, 3, 4, 5$, plot the value of the K-means objective function per iteration for 20 iterations (the algorithm may converge before that).
2. For $K = 3, 5$, plot the 500 data points and indicate the cluster of each for the final iteration by marking it with a color or a symbol.

Problem 2 (Matrix factorization) –85 points

In this problem, you will implement the MAP inference algorithm for the matrix completion problem discussed in class. As a reminder, for users $u \in \mathbb{R}^d$ and movies $v \in \mathbb{R}^d$, we have

$$u_i \sim N(0, \lambda^{-1} I), \quad i = 1, \dots, N_1, \qquad v_j \sim N(0, \lambda^{-1} I), \quad j = 1, \dots, N_2.$$

We are given an $N_1 \times N_2$ matrix M with missing values. Given the set $\Omega = \{(i, j) : M_{ij} \text{ is measured}\}$, for each $(i, j) \in \Omega$ we model $M_{ij} \sim N(u_i^T v_j, \sigma^2)$.

Run your code on the user-movie ratings dataset provided on Courseworks and the course website. For your algorithm, set $\sigma^2 = 0.25$, $d = 10$ and $\lambda = 10$. Train the model on the larger training set for 100 iterations. For each user-movie pair in the test set, predict the rating by mapping the relevant dot product to the closest integer from 1 to 5. Since the equations are in the slides, there's no need to re-derive it.

1. Plot the RMSE of your predictions on the held out test set provided for iteration number 2 to 100.

2. On a separate plot, show the log joint likelihood for iterations 2 to 100.

3. After 100 iterations, pick three reasonably well-known movies from the list provided and for each movie find the 5 closest movies according to Euclidean distance using their respective locations v_j. List the query movie, the five nearest movies and their distances. A mapping from index to movie is provided with the data.

4. After 100 iterations, perform K-means on the vectors $u_1, \dots, u_{N_1}$ learned by your algorithm. Set $K = 20$, which is an arbitrary number. The centroids can be interpreted as personality types (as far as movies are concerned).

   - Pick the 5 centroids corresponding to the 5 clusters that have the most data. For each cluster selected, give the number of users allocated to that cluster.
   - For each of these 5 centroids, list the 10 movies with the largest (most positive) dot product with that centroid. Also give the value of the dot product next to the movie.

5. After 100 iterations, perform K-means on the vectors $v_1, \dots, v_{N_2}$ learned by your algorithm. Set $K = 20$, which is an arbitrary number.

   - Pick the 5 centroids corresponding to the 5 clusters that have the most data. For each cluster selected, give the number of movies allocated to that cluster.
   - For each of these 5 centroids, list the 10 movies with the smallest Euclidean distance to that centroid. Also give the value of the Euclidean distance next to the movie.