EECS E6720: Bayesian Models for Machine Learning

Columbia University, Fall 2018

Homework 2: Due Sunday, October 14, 2018 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, but your code should be your own. Do not submit Jupyter or other notebooks, but the original source code only. 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.* 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. You can resubmit as much as you like, but each time you resubmit be sure to upload **all** files you want graded! 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. (40 points)

In this homework, you will implement an EM algorithm for the object recommendation problem discussed in class. Here, we have a data set of the form $\mathcal{R} = \{r_{ij}\}$ restricted to a subset of pairs $(i,j) \in \Omega$, where i can range from $1,\ldots,N$ and j can range from $1,\ldots,M$ and we let $r_{ij} \in \{\pm 1\}$. The goal of matrix factorization is to find a low rank approximation of this data.

To this end, let the unknown model variables be $u_i \in \mathbb{R}^d$ and $v_j \in \mathbb{R}^d$ for $i = 1,\ldots,N$ and $j = 1,\ldots,M$. Let $U = \{u_i\}$ and $V = \{v_j\}$. The model and priors are,

$$r_{ij}|U,V \overset{ind}{\sim} \text{Bernoulli}(\Phi(u_i^T v_j/\sigma)), \text{ for all } (i,j) \in \Omega, \qquad u_i \overset{iid}{\sim} N(0,cI), \;\; v_j \overset{iid}{\sim} N(0,cI).$$

The function $\Phi(\cdot)$ is the CDF of a standard normal random variable, as discussed in class. (Hint: You will find the probit regression discussion useful for this homework.)

The goal of this homework will be to derive and implement an EM algorithm for maximizing

$$\ln p(\mathcal{R},U,V) \;=\; \underbrace{\int q(\phi) \ln \frac{p(\mathcal{R},U,V,\phi)}{q(\phi)} d\phi}_{\mathcal{L}(U,V)} \;+\; \int q(\phi) \ln \frac{q(\phi)}{p(\phi|\mathcal{R},U,V)} d\phi,$$

where the auxiliary variable ϕ is used in the probit setup similar to that discussed in class for probit regression: $\phi = \{\phi_{ij}\}$ for $(i,j) \in \Omega$ and $r_{ij} = \text{sign}(\phi_{ij})$, $\phi_{ij} \sim \text{Normal}(u_i^T v_j, \sigma^2)$.[1]

[1] Note: In the data, $r_{ij} \in \{-1,+1\}$ instead of $\{0,1\}$. This won't impact the final derivation.

Problem 1 will focus on deriving the algorithm. To this end:

a) Derive $q(\phi)$ for the EM algorithm. Hint: You will be able to show $q(\phi) = \prod_{(i,j)\in\Omega} q(\phi_{ij})$.

b) Derive $\mathcal{L}(U, V)$. You can use the $\mathbb{E}_q[\cdot]$ notation and write the relevant expectation separately.

c) Derive the U, V that optimize $\mathcal{L}(U, V)$. Hint: Derive for one (and therefore all) u_i and v_j.

d) Summarize the final EM algorithm in a list of steps that are easy to read, yet detailed enough for someone to be able to implement it. (Be sure to write out $\ln p(\mathcal{R}, U, V)$.)

Problem 2. (35 points)

In this problem, you will implement the EM algorithm you derived in Problem 1 for finding a local optimal solution to $\ln p(\mathcal{R}, U, V)$. The data is provided on Courseworks along with a README file explaining the data. For your implementation, set $d = 5$, $c = 1$ and $\sigma^2 = 1$. Treat the users as U and the movies as V when implementing the algorithm.[2]

You will need to initialize your algorithm in some way. One possibility is to start by generating each u_i and v_j from $\text{Normal}(0, 0.1I)$. Please use this method for this assignment.

a) Run your algorithm for 100 iterations and plot $\ln p(\mathcal{R}, U, V)$ for iterations 2 through 100.

b) Rerun your algorithm for 100 iterations using 5 different random starting points. Plot the 5 different objective functions for iterations 20 through 100. Note: This is simply a repeat of Problem 2(a), only showing 5 objective functions instead of one and changing the x-axis.

c) Predict the values given in the test set and show your results in a confusion matrix. Show the raw counts in this confusion matrix.

[2] Hint: Depending how you implement it, your code may be very fast or very slow. You may want to change the data format so you aren't looping over the list of ratings. Also, the EM algorithm will work if you change around the orders: e.g., update $q(\phi)$, then update U, then update $q(\phi)$, then update V. We didn't present EM this way—we presented it as doing "E" for everything then "M" for everything in one iteration—but it may help develop your understanding of EM if you think through why this suggestion also works. Your code will work if you ignore this hint, but it may end up being slow, and this hint is probably not the only way to faster code.