Assignment 1

CMSC 691 — Graphical and Statistical Models of Learning

Due Friday March 27th, 11:59 PM

<table>
<thead>
<tr>
<th>Item</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due</td>
<td>Friday March 27th</td>
</tr>
<tr>
<td>Topic</td>
<td>EM for Mixture Models</td>
</tr>
<tr>
<td>Points</td>
<td>130</td>
</tr>
</tbody>
</table>

In this assignment you will gain experience deriving, implementing, and experimenting with EM.

You are to complete this assignment on your own: that is, the code and writeup you submit must be entirely your own. However, you may discuss the assignment at a high level with other students or on the discussion board. Note at the top of your assignment who you discussed this with or what resources you used (beyond course staff, any course materials, or public Piazza discussions).

What To Turn In  Turn in a PDF writeup that answers the questions; turn in all requested code necessary to replicate your results. Be sure to include specific instructions on how to build (compile) and run your code. Answers to the following questions should be long-form. Provide any necessary analyses and discussion of your results.

How To Submit  Submit the assignment on the submission site:

https://www.csee.umbc.edu/~ferraro/teaching/691-s20/submit

Be sure to select “Assignment 1.”
Model Description

Consider a dataset \( \{x_1, x_2, \ldots, x_N\} \), where each \( x_i \) is a \( D \)-dimensional vector defined in some space \( H^D \) (\( x_i \in H^D \)). For ease, you can consider \( x_i \in \mathbb{R}^D \) (\( H = \mathbb{R} \)), but it could be that tighter specifications can be given. For example, each component \( x_{i,d} \) could be a binary value (0 or 1), in which case \( x_i \in \{0, 1\}^D \).

Let \( f(\theta) \) be a probability distribution that appropriately supports \( x_i \), and \( \theta \) be a \( J \)-dimensional parameter vector of learnable weights. Note that \( K \) and \( J \) can be different! For example, if \( x_i \) is a \( D \)-dimensional real-valued vector, then \( f(\theta) \) could be a multivariate Gaussian distribution; if \( x_i \) is a \( D \)-dimensional positive-valued vector, then \( f(\theta) \) could be a multivariate log-Normal distribution; and if \( x_i \) is a \( D \)-dimensional count vector, then \( f(\theta) \) could be a multinomial distribution. (For each of these scenarios, there are also other distributions that could be used.)

This entire assignment considers mixture models defined on \( K \) different distributions \( f(\theta_1), \ldots, f(\theta_K) \) that conform to the following generative story:

1. Draw mixing weights \( \pi \sim g(\alpha) \).
2. Let \( \theta_k \in \mathbb{R}^J \).
3. For data points \( i = 1 \ldots N \):
   1. Sample the component to draw from: \( z_i \sim \text{Cat} (\pi) \)
   2. Generate \( x_i \) from the chosen component: \( x_i | z_i \sim f(\theta_{z_i}) \)

Here \( \pi \in \Delta^{K-1} \), and \( g \) is an appropriate distribution parametrized by \( \alpha \). In class, we mentioned the Dirichlet and logistic-Normal distributions as reasonable options. Note that each \( \theta_k \) is a vector.

While \( \pi \) has the simplex constraint, each \( \theta_k \) may have its own constraints. Assume that these can be written as \( h_k(\theta_k) = b_k \). If there are no constraints on \( \theta_k \), then \( h_k(\theta_k) = b_k = 0 \) is a valid setting.

We will call this model the \( f \)-MM (\( f \) Mixture Model).
Questions

We are going to use the above story to model our dataset. Assuming that given \( \pi, \{ \theta_k \} \), each \( x_i \) is i.i.d.:

1. **(5 points)** Just for this question: if \( x_i \in \mathbb{R} \) were a positive, real-valued scalar and \( f \) were a Gamma distribution, specify \( D \) and \( J \), and provide an example of a \( \theta_k \). Identify any constraints on \( \theta_k \).

   (Unless otherwise specified, do not use the above assumptions in the remaining questions.)

2. **(5 points)** Write the formula for the marginal likelihood (or log-likelihood) of a single data point \( x_i \) according to the \( f \)-MM. It may help to first identify the random variables that would need to be marginalized.

3. **(5 points)** Write the formula for the marginal log-likelihood of the data \( \{ x_1, \ldots, x_N \} \) according to the \( f \)-MM.

4. **(5 points)** If we wanted to apply EM to learn this model, we would need to be able to compute the posterior distribution(s). Identify this posterior distribution (or distributions). If it helps, you can think of it as finding appropriate values for \( \square \) (hyperparameters), \( \diamond \) and \( \clubsuit \):

   \[ p(\square | \diamond, \clubsuit). \]

5. **(7 points)** Let \( q^* \) represent the above posterior distribution. For arbitrary \( f \) and \( g \), learning this model is difficult. We can try to apply EM, which optimizes

   \[ \max_{\theta_1, \ldots, \theta_K, \alpha} \mathbb{E}_{q^*} \left[ \log p(\pi, z, x) \right]. \]  \[ \text{[Eq-1]} \]

   However, there’s a difficulty in computing this expectation analytically (in closed form). Specify what the difficulty is. As a hint, it involves \( \pi \).

6. **(13 points)** In order to handle this difficulty, we can resort to maximum a posteriori EM (MAP EM). In MAP EM, we instead solve the following problem:

   \[ \max_{\pi, \theta_1, \ldots, \theta_K} \mathbb{E}_q \left[ \log p(\pi, z, x) \right]. \]  \[ \text{[Eq-2]} \]

   For this question, write out the appropriate Lagrangian objective (i.e., one that takes into account all necessary constraints). Refer to this objective as \( L \). Your formation of \( L \) should expand any expectations into their respective summation (or integral, if appropriate) form, such as in Equation 7 of the GMM handout available on Piazza.

   See Appendix A for a short explanation of MAP EM.

7. **(15 points)** Derive gradients/partial derivatives of \( L \) for all variables. Note that your answer will involve terms such as \( \frac{\partial f(x_i; \theta_k)}{\partial \theta_{k,j}} \), \( \frac{\partial g(\pi, \alpha)}{\partial \alpha_k} \), and \( \frac{\partial h_k(\theta_k)}{\partial \theta_{k,j}} \).

   For example, if \( L \) had only the simplex constraint on \( \pi \), then you’d need to derive \( \frac{\partial L}{\partial \pi_k} \), \( \frac{\partial L}{\partial \alpha_k} \), and the partial derivative of \( L \) related to that simplex constraint.

8. **(15 points)** Assuming that each \( x_i \) is an instance from the MNIST digit dataset, identify appropriate \( f \), \( g \), and \( h_k \) and refine the above gradients/partial derivatives of \( L \) based on these functions. I.e., compute the terms such as \( \frac{\partial f(x_i; \theta_k)}{\partial \theta_{k,j}} \), put them in to the partial derivatives, and simplify. These represent the update equations for your \( f \)-MM.
See Appendix B.1 for an overview of what MNIST is and where to get it, and Appendix B.2 for suggested distributions. If you decide to use a multivariate Gaussian distribution, then while your covariance matrix can be diagonal, it must have separate variables for each diagonal component, and you must provide derivatives for both the mean vectors and covariances matrices.

(9) **(20 points)** Using the equations from [8], implement your f-MM. If you wish, you may provide pseudo-code in your report. In addition:

- Identify any implementation difficulties you had and how you addressed them.
- Discuss how you ensure all constraints were satisfied.
- Picking a reasonable value of $K$, train your f-MM for 10 epochs and plot the marginal log-likelihood of the training set for each epoch. The plot should be legible.
- Discuss the convergence behavior of your implementation. In particular, do 10 epochs seem to be enough? Should more be used, or fewer? Explain how you came to this conclusion.

(10) **(20 points)** Experiment on the training set with at least four different values of $K$, with at least $M = 3$ different initializations, and plot the final training marginal log-likelihood for each $K$. In your report, discuss the impact of $K$ on the ability of the f-MM to fit the data.

Specifically, for $K \in \{K_1, K_2, K_3, K_4, \ldots\}$: (a) train your f-MM from scratch $M$ times using $K$ components, (b) record the final marginal log-likelihood $l_{k,m}$ on for each model on the training set, (c) and plot the average $\hat{l}_k = \frac{\sum_l l_{k,m}}{M}$, along with standard deviation bars for these values vs. $K$. The plot should be legible.

(11) **(20 points)** Pick a $K$ value that provided good performance and train a model with that many components. Use this learned model to sample 3-5 instances per component. Include these sampled images in your report as 28x28 images. Discuss your overall (qualitative) take on the sampled images. Are they better (or worse) than you expected? Speculate on how you might be able to change the model/story to better model the data. (You don’t actually have to make these changes.)

---

1 Or, load up a serialized model.
A MAP EM

MAP EM uses EM to learn certain parameters, but then directly optimizes over other random variables. In effect, it approximates an expectation with a maximization: if there is a random variable $y$ for which an expected value $\mathbb{E}[h(y)]$ is difficult to compute, then the expected value is approximated by the maximum of $h$: $\mathbb{E}[h(y)] \approx \max_y h(y)$.

There are a few things to note about this: (i) The $q$ in [Eq-2] is different from the $q^*$ in [Eq-1]. (ii) $q$ should not contain any of the variables that are being maximized over. (iii) [Eq-2] is no longer maximizing over $\alpha$ (we are considering those values fixed).

B MNIST

B.1 Overview

The MNIST digit dataset is a very popular and well-known ML dataset: in total, there are 70,000 “images” of handwritten digits (each between 0-9). The task is to predict the digit from a 28x28 input grayscale image. 60,000 of these are allocated for training and 10,000 are allocated for testing.

In this assignment’s notation, $D = 28 \times 28 = 784$.

Each component $d$ of the input image $t_i$ is a float ($0 \leq t_{i,d} \leq 1$). This corresponds to

$$x_{i,d} = t_{i,d}. \hspace{1cm} \text{[Eq-3]}$$

It’s also common to threshold values at or above 0.5 to 1 and below 0.5 to 0. This corresponds to

$$x_{i,d} = \begin{cases} 
1 & t_{i,d} \geq 0.5 \\
0 & t_{i,d} < 0.5. 
\end{cases} \hspace{1cm} \text{[Eq-4]}$$

For this assignment you may choose whichever representation you wish (including others not listed here).

The dataset is available from multiple locations, in multiple formats. All of the data in these formats is the same, but don’t assume the order within each set is the same. Which you use is more a question of what is easiest for you.

- Via Scipy/some other dataloader.
- [for Python (2.7)] As a gzipped row-major Pickle file, on GL at
  
  https://www.csee.umbc.edu/~ferraro/teaching/691-s20/materials/mnist_rowmajor.pkl.gz
  
  This maps the keys ‘images_train’, ‘labels_train’, ‘images_test’, ‘labels_test’ to numpy arrays. This Pickle file was created with Python 2.7. To read this file in Python 3, you must specify an encoding (otherwise you will get a loading error):

  ```python
  > import gzip, pickle
  > with gzip.open(‘mnist rowmajor.pkl.gz’, ‘rb’) as data_fh:
  >     data = pickle.load(data_fh, encoding=’latin1’)
  ```

- [for Matlab or Python] As column-major mat files, on GL at
This maps the keys 'images_train', 'labels_train', 'images_test', 'labels_test' to arrays/vectors.

- [for any language] as four compressed IDX files from the “original” source, at http://yann.lecun.com/exdb/mnist/² These are binary files. If you want to use this version, be aware of first the type to read (e.g., 32 bit int vs. unsigned byte), and that ints are in a high endian form!

### B.2 Suggested Distributions

You may use external resources to look up these distributions (or others). Just reference and cite whatever you use!

**Using a float representation** ([Eq-3]) If you use the representation from [Eq-3], then I recommend representing each component \( x_{i,d} \) independently from its own Beta distribution. That is, \( f(x_i; \theta_k) = \prod_d \text{Beta}(x_i; \theta_{k,d}) \).

**Using a binary representation** ([Eq-4]) If you use the representation from [Eq-4], then I recommend representing each component \( x_{i,d} \) independently from its own Bernoulli distribution. That is, \( f(x_i; \theta_k) = \prod_d \text{Bernoulli}(x_i; \theta_{k,d}) \).

²They are also on GL at /afs/umbc.edu/users/f/e/ferraro/pub/678-s19/mnist-data/*idx*.gz. However, you will need to reference the format documentation at the above URL.