Math 611 HW4

Jeff Gould

9/26/2020

1. Reading (optional).
   (a) Section $9.2 - 9.4$ discuss hard and soft EM, the rigorous definition of EM, and kmeans
   (b) Section 11.2 discusses Markov chains (not very clearly) and Metropolis-Hastings.

2. In this problem, we will revisit the Hope heights problem of HW 3, but this time we will use EM. Recall, we consider the two component Gaussian mixture model,

$$X = \begin{cases} \mathcal{N}(\mu_1, \sigma_1^2) & \text{with probability } p_1 \\ \mathcal{N}(\mu_2, \sigma_2^2) & \text{with probability } p_2 \end{cases} \tag{1}$$

   where $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution and X models the height of a person when gender is unknown. Let $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_N$ be the sample heights given in the file.
   (a) Take a hard EM approach and compute the parameters of the model.
   (b) Take a soft EM approach and compute the parameters of the model. Compare to your results in (a).
   (c) Given your parameters in (b), use the distribution of X to predict whether a given sample is taken from a man or woman. Determine what percentage of individuals are classified correctly.

3. Attached you will find the file `TimeSeries.csv`. The file contains a 1000 by 20 matrix. Each row represents a sample of a random vector $X \in \mathbb{R}^{20}$, but X represents time series data, so that $X_1, X_2, \ldots, X_{20}$ represent measurements at times $1, 2, \ldots, 20$, respectively. Often, we have a collection of time series samples and would like to separate the samples into similar groups, i.e. cluster. Here we'll do this by using a multivariate normal mixture model.

   To visualize the data, produce a line plot of each sample. In R, you can execute

```
plot(m[1,], type="l", ylim=c(-12,12))
for (i in 2:1000) {
  lines(m[i,])
}
```

   where `m` is the matrix in the csv file. You'll see that the time series are not easy to distinguish. The file `make_timeseries.R` contains the code used to make the data. The data is based on 4 underlying time series found in the file `BaseSeries.csv` which contains a 4×20 matrix. Look through the files and explain how the data was generated.

   Now assume the following model for X

$$X = \begin{cases} \mathcal{N}(\mu^{(1)}, \Sigma^{(1)}) & \text{with probability } p_1 \\ \mathcal{N}(\mu^{(2)}, \Sigma^{(2)}) & \text{with probability } p_2 \\ \vdots & \\ \mathcal{N}(\mu^{(K)}, \Sigma^{(K)}) & \text{with probability } p_K \end{cases} \tag{2}$$

   Each of the $\mu^{(i)} \in \mathbb{R}^{20}$ and each $\Sigma^{(i)}$ is a 20×20 covariance matrix. K is the number of mixtures, which we must choose. (In this case, since you know the solution, you can set $K = 4$.)

(a) To fit this model using EM, you need to know how to derive the MLE of a multivariate normal. Let Z be an n-dimensional multivariate normal with mean μ and covariance matrix Σ. Let $\hat{Z}^{(i)}$ be iid samples from Z for $i = 1, 2, \ldots, N$. Write down the log-likelihood and use it to show that the MLE estimate $\hat{\mu}$ of μ is given by the sample mean of the $\hat{Z}^{(i)}$. Then read Chis Murphy's section 4.1.3 of the book Machine Learning (attached) and in your own words summarize the steps needed to derive the MLE for the variance (or derive it yourself if you prefer). (Bishop does not include the derivation in his book.)

(b) Take a **hard** EM approach to estimating the parameters of the model. When you stop your iteration, plot the $\mu^{(i)}$ and determine if you have recovered the underlying time series used to generate the data.

(c) Now repeat, but take a **soft** EM approach. Compare your result to what you found using a hard EM approach.

4. Consider the hard core model on a 100×100 grid. Let Ω be the set of all configurations and H the set of configurations that do not violate the hard-core restriction (no neighboring 1's). For $w \in \Omega$ let $f(w)$ be the number of positions with a 1 in the grid.

(a) Let X be the r.v. on Ω,

$$P(X = w) = \begin{cases} \frac{1}{|H|} & \text{if } w \in H \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Using the Metropolis-Hasting algorithm, write a sampler for X. Show a sample configurations using the **image** function in R (or equivalent in Python). Using your sampler, generate a histogram for $f(X)/100^2$, the fraction of sites with a 1 under the uniform distribution X. To decide how long to run the MH-algorithm before sampling, plot $f(X)$ as a function of the time step of your chain. If plotted on a long enough time scale, the plot should look noisy. Once you decide how long to run the chain, run the chain many times to produce a histogram. (Each time you sample from the Metropolis-Hastings algorithm you have to rerun the chain.)

(b) Let Y be the r.v. on Ω defined by $P(Y = w) = \alpha(f(w))^2$, where α is a normalizing constant that makes the probabilities sum to 1. Use your sampler to generate a histogram for $f(Y)/100^2$. Compare to part a).