Homework 10: Gaussian Mixture Models

This homework covers gaussian mixture models (GMMs), and will give you practice implementing expectation maximization algorithms.

1 Goals

In this homework you will:

1. Implement expectation and maximization procedures to train a GMM
2. Use the log-likelihood on the dataset to determine when the GMM has converged
3. Analyze and plot the components of a trained GMM for different numbers of clusters

2 Background

Before attempting the homework, please review the notes on clustering. There, we define a GMM with k components (clusters) as

$$p_X(x) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x|\mu_i, \sigma_i^2)$$

where μ_i, σ_i, π_i are the mean, standard deviation, and weight of the ith Gaussian. To solve this, we introduce an expectation-maximization procedure, which is carried out until the model parameters converge. Two additional components of this procedure are important to know for the homework: initialization and log-likelihood.

2.1 Initializing Parameters

Before the first run of the expectation procedure, we must initialize the π_i, μ_i, and σ_i variables. There are several possible ways to do this, and in fact, no "right" way that will work best for all possible datasets. One way would be to just choose values purely randomly, but that may require a much longer time to converge. A more sophisticated way would be to set the initial centers to the result of another clustering algorithm such as k-means, but this also requires initializing the other clustering algorithm.

In this homework, you will implement a "common sense" initialization approach based on our knowledge of the parameters:

- For the π_i, we know that $\sum_{i=1}^{k} \pi_i = 1$. So, we will initialize them all to the same value: $\pi_i = 1/k$.
- For the μ_i, we at least know that the Gaussians must be centered somewhere within the range of the data, and must be at different points. So, we will spread them uniformly throughout the range: $\mu_i = x_0 + i \cdot (x_n - x_0)/k$, where we have assumed x_0 is the smallest and x_n is the largest number in

the dataset. For example, if we initialized $k = 4$ clusters between $x_0 = 0$ and $x_n = 10$, we would start with $\mu_1 = 2.5, \mu_2 = 5.0, \mu_3 = 7.5, \mu_4 = 10$.

- The σ_i are a bit harder to gain intuition about, so we will initialize them each to 1: $\sigma_i = 1$.

2.2 Log likelihood

The expectation maximization procedure attempts to find the model parameters that maximize the **log-likelihood** of the dataset. Though we already have the equations necessary to do this, we still need to know when we can stop iterating through the procedure, which is best judged by considering successive changes in the log-likelihood: once the change is smaller than some tolerance, we can assume the procedure has converged.

The likelihood of the dataset $x_1, x_2, ..., x_n$ for the current model is given by

$$p_X(x_1, x_2, ..., x_n) = \prod_{j=1}^{n} p_X(x_j) = \prod_{j=1}^{n} \left[\sum_{i=1}^{k} \pi_i \mathcal{N}(x_j | \mu_i, \sigma_i^2) \right]$$

The log-likelihood is just the log of this expression:

$$\log p_X(x_1, ..., x_n) = \log \prod_{j=1}^{n} \left[\sum_{i=1}^{k} \pi_i \mathcal{N}(x_j | \mu_i, \sigma_i^2) \right] = \sum_{j=1}^{n} \log \left(\sum_{i=1}^{k} \pi_i \mathcal{N}(x_j | \mu_i, \sigma_i^2) \right)$$

After each iteration of expectation-maximization, we calculate $\log p_X(x_1, ..., x_n)$ to see how much it has changed from before. The log here can be taken in any base, but it is common to use the natural log ln.

3 Instructions

3.0 Setting up your repository

Click the link on Piazza to set up your repository for HW 10, then clone it.

The repository should contain three files aside from this readme:

1. `gmm-em.py`, a starter file with functions, instructions, and a skeleton that you will fill out in Problem 1.
2. `gmm-visualize.py`, a starter file with functions, instructions, and a skeleton that you will fill out in Problem 2.
3. `data.txt`, a file containing data points that you will use to answer the questions in Problem 2.

3.1 Problem 1: Expectation-Maximization Procedure

In this problem, you will complete several of the functions in `gmm-em.py` related to carrying out the expectation maximization procedure on an input dataset. More specifically:

1. Complete `expectation`, which takes as input the dataset `data` and the current values of `weights` (the π_i), `means` (the μ_i), and `varis` (the σ_i^2) to calculate the new `gammas` (the γ_{ij}).
2. Complete `maximization`, which takes as input the dataset `data` and the current `gammas` to calculate the new `weights`, `means`, and `varis`.

3. Complete the initialization procedure in `train` for `weights`, `means`, and `varis` based on the specifications at the end of Section 2.1.
4. Complete `log_likelihood`, which takes as input the dataset `data` and model parameters `weights`, `means`, `varis` and calculates the log-likelihood (denoted `ll`) according to the specification in Section 2.2. Be sure to use the natural log. (Hint: Use `norm.pdf` from `scipy.stats` to evaluate the probability density functions at a point.)

This completed code will output the `weights`, `means`, `varis`, and `ll` for a gaussian mixture model with k clusters trained on the input dataset at `datapath` until the log-likelihood between two iterations of the expectation-maximization procedure changes by less than `tol`.

3.2 Problem 2: Model Visualization and Comparison

In this problem, you will complete the empty functions in `gmm-visualize.py` that plot a trained gaussian mixture model and the individual components of the model. You will then use the completed code in `gmm-em.py` and `gmm-visualize.py` to visualize mixture models for the input dataset `data.txt`. More specifically:

1. Complete `gauss`, which evaluates the probability density of a Gaussian model with mean `mu` and variance `var` at all the datapoints in `x`.
2. Complete `plot_model`, which uses `matplotlib.pyplot` to plot the individual gaussians in `cluster` and the full mixture model in `model` against the input data range `x`. Be sure to include a legend that differentiates between the different curves plotted. For example, if there are $k = 3$ clusters, the plot should include four curves, one for each of the three clusters and one for the mixture.
3. Run `gmm-em.py` on `data.txt` for $k = 2, 3, 4, 5, 6$ using a tolerance `tol` $= 1$. Write out the fitted mixture model formulas $p_X(x)$ for each k. What do you observe about the log-likelihoods?
4. Input your results from `gmm-em.py` for each value of k into `gmm-visualize.py`. Save your plots and turn them in as part of your writeup. How many clusters does this dataset have? Explain.

4 What to Submit

You should submit three files:

- Your completed `gmm-em.py` file
- Your completed `gmm-visualize.py` file
- A separate PDF file called `writeup.pdf` containing your writeup for Problems 2.3 and 2.4

5 Submitting your Code

Please tag the version of the code that you want to submit with submission, as you did in the previous HW.

Don't forget to commit the code that you want to submit before tagging your submission. You have to do this in two steps.