MATH 728D: Machine Learning Lab #10: Gaussian Mixture Models

John Burkardt

December 7, 2018

I know my data forms k clusters; I'll guess they're Gaussian; how do I sort my data?

In our previous work with clustering, the only assumption we made about the data we received was that maybe it could be organized into a small number k of clusters. But suppose we can make stronger assumptions about the data, such as:

- the data represents samples from k separate Gaussian distributions $N(\mu_i, \sigma_i^2)$;
- each distribution has a separate mean μ_i and variance σ_i^2, which we do not know;
- we know the value of k;

It should be clear that if we carry out the k-means process on this data, the resulting patterns will give us evidence suggesting good guesses for all the values of μ and σ. In fact, if we trust our assumptions, there is even more that we can do.

1 Estimate Parameters of a Normal PDF

Let's begin with a simple problem. Suppose we have a set `x` of scalar data values, and we believe that $x \sim N(\mu, \sigma^2)$. We are familiar with the idea that the values of x can be used to estimate μ and σ^2:

$$\mu \approx \hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} x_i$$

$$\sigma^2 \approx \hat{\sigma^2} = \frac{1}{m} \sum_{i=1}^{m} (x_i - \hat{\mu})^2$$

Exercise 1:

1. Iterate for $\mathtt{n} = 10^1, 10^2, ..., 10^7$;
   (a) Set `x = 12.3 + 4.5 * randn(n,1);`;
   (b) Compute estimates for mean and variance;
   (c) Print `n` and errors in mean and variance estimates;

2 Evaluate and Sample a 1D Gaussian Mixture PDF

In 1D, a *Gaussian mixture model* (GMM) is a probability density function (PDF) which is a set of k Gaussian distributions, each with a weight p_j, and its own mean μ_j and variance σ_j^2:

$$gmm(x) = \sum_{j=1}^{k} \frac{p_j}{\sqrt{2\pi\sigma_j^2}} \quad e^{-\frac{x-\mu_j}{\sigma_j^2}}$$

We assume the weights sum to 1, and so p_j represents the probability that Gaussian distribution j will be selected on a particular sample. To sample a GMM, you first use the weights to randomly choose the particular Gaussian j. Then it's easy to sample x_i from $N(\mu_j, \sigma_j^2)$:

```
x = mu(i) + sigma(i) * randn();
```

How do the weights p control the choice of distribution? If the weights were equal, we could simply call MATLAB's `randi([1,k])` to select evenly. To deal with unequal weights, we can call `randsample(k,1,true,p)` to return a value between 1 and k according to the weights.

Exercise 2:

- Define a GMM with three components:

```
k = 3;
p = [ 0.2, 0.3, 0.5 ];
mu = [ -1, 1, 3 ];
sigma = [ 0.4, 0.6, 1.5 ];
```

- Plot the GMM PDF over $-2 \le x \le 7$.
- Sample 10000 values from the GMM, and create a histogram of the results.

Do your PDF and histogram plots correspond?

3 Evaluate and Sample a 2D Gaussian Mixture Model

In dimension m, the formula for a single Gaussian distribution depends on a mean m-vector μ, and an $m \times m$ positive definite symmetric covariance matrix Σ:

$$pdf(x) = \frac{1}{(2\pi)^{\frac{m}{2}} \det(\Sigma)} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}$$

The matrix Σ describes how sample values deviate from the mean. In 2D, if Σ is the identity matrix, then sample values are scattered in a circular pattern around the mean; if Σ is a diagonal matrix, the circle becomes a vertical or horizontally oriented ellipse; nonzero off-diagonals of Σ produce more complicated scattering patterns.

Just as in 1D, a Gaussian mixture model can be constructed which assigns weights p to k distinct Gaussian distributions. In this exercise, we will evaluate and sample a 2D GMM with just two components.

Exercise 3:

- Define a 2D GMM with two components:

```
k = 2;
p = [ 0.35, 0.65 ];
mu1 = [ 1.0, 2.0];
```

```
mu2 = [ 4.0, 1.0];
sigma1 = [ 1.0, 0.0;
           0.0, 2.0 ];
sigma2 = [ 5.0, 3.0;
           3.0, 2.0 ];
```

- Create an 2×500 array `x` of samples from the GMM. If component 1 is picked, for instance, your j-th sample would be:

  ```
  xs(1:2,j) = mu1 + sigma1 * randn(2,1);
  ```

- Also record in the vector `y` a 1 or a 2, depending on which distribution was sampled;
- Use the command `gscatter ( x(:,1), x(:,2), y )` to plot your data points, using color to distinguish the two distributions.

Discuss the relationship between the observed data on the plot and the values of μ and Σ that you used.

4 Use gmdist() to Estimate a 2D GMM

If we assume that data is generated by a GMM with k components, there are algorithms to estimate the mixture coefficients p, and the corresponding μ vectors and Σ matrices associated with the components. The MATLAB command `gmdist()` can be used for this purpose. We can also use the `ezcontour()` command to show contour levels of the probability density associated with each distribution.

Exercise 4:

1. Use `xlsread()` to create the array `data` from *climate_data.xls*;
2. Display the data with `gscatter ( data(:,1), data(:,2) )`;
3. Estimate the GMM with the command

   ```
   k = 2;
   gmm = fitgmdist ( data, k );
   ```

4. Print the estimated means and Σ matrices:

   ```
   gmm.mu
   gmm.Sigma
   ```

5. Estimate the cluster assignment for each data item:

   ```
   data_cluster = cluster ( gmm, data );
   ```

6. Recreate the scatterplot, but now with cluster assignments:

   ```
   gscatter ( data(:,1), data(:,2), data_cluster );
   ```

7. Display the shape of the distributions:

   ```
   hold on
   ezcontour ( @(x1,x2) pdf ( gmm, [x1,x2] ), [ 0 45 0 30 ] );
   hold off
   ```