Gaussian Mixture Model on Student Performance Data

Shubham Barudwale
School of Computing Sciences and Engineering, VIT Chennai, Tamilnadu, India 600127
Email: barudwaleshubham.dinesh2015@vit.ac.in

Abstract-

Gaussian mixture models are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning.

For example, in modeling human height data, height is typically modeled as a normal distribution for each gender with a mean of approximately 5'10" for males and 5'5" for females. Given only the height data and not the gender assignments for each data point, the distribution of all heights would follow the sum of two scaled (different variance) and shifted (different mean) normal distributions. A model making this assumption is an example of a Gaussian Mixture Model (GMM), though in general, a GMM may have more than two components. Estimating the parameters of the individual normal distribution components is a canonical problem in modeling data with GMMs.

1. Introduction, Algorithm and Methodology

If the number of components is known, **expectation maximization** is the technique most commonly used to estimate the mixture model's parameters. In frequentist probability theory, models are typically learned by using maximum likelihood estimation techniques, which seek to maximize the probability, or likelihood, of the observed data given the model parameters. Unfortunately, finding the maximum likelihood solution for mixture models by differentiating the log likelihood and solving for is usually analytically impossible. Expectation maximization (**EM**) is an numerical technique for maximum likelihood estimation, and is usually used when closed form expressions for updating the model parameters can be calculated (which will be shown below). Expectation maximization is an iterative algorithm and has the convenient property that the maximum likelihood of the data strictly increases with each subsequent iteration, meaning it is guaranteed to approach a local maximum or saddle point.

EM for Gaussian Mixture Models

Expectation maximization for mixture models consists of two steps.

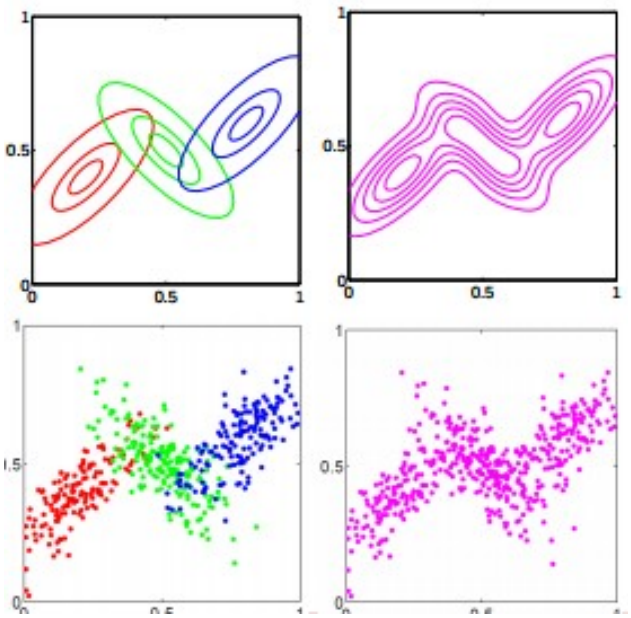
The first step, known as the **Expectation**, or **E**, step, consists of calculating the expectation of the component assignments Ck for each data point given the model parameters .

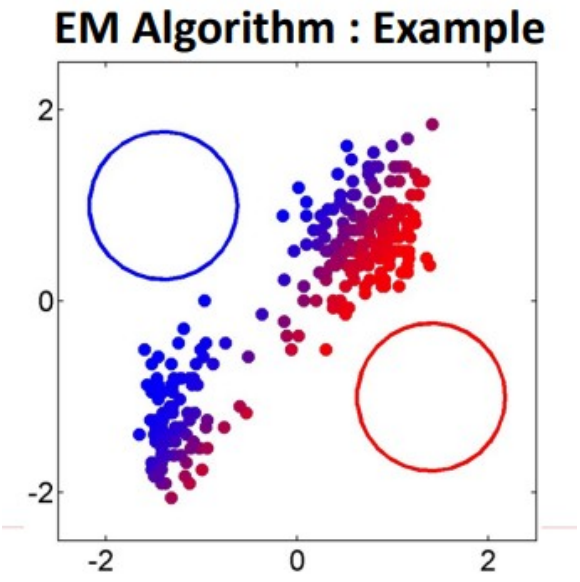
The second step is known as the **Maximization**, or **M**, step, which consists of maximizing the expectations calculated in the E step with respect to the model parameters.

The entire iterative process repeats until the algorithm converges, giving a maximum likelihood estimate. Intuitively, the algorithm works because knowing the component assignment for each makes solving values and easy, while knowing and makes inferring easy. The expectation step corresponds to the latter case while the maximization step corresponds to the former. Thus, by alternating between which values are assumed fixed, or known, maximum likelihood estimates of the non-fixed values can be calculated in an efficient manner.

The expectation maximization algorithm for Gaussian mixture models starts with an initialization step, which assigns model parameters to reasonable values based on the data. Then, the model iterates over the

Expectation (E) and Maximization (M) steps until the parameters' estimates converge, i.e. for all parameters at iteration,for some user-defined tolerance.





The main difficulty in learning Gaussian mixture models from unlabeled data is that it is one usually doesn't know which points came from which latent component (if one has access to this information it gets very easy to fit a separate Gaussian distribution to each set of points).Expectation-maximization is a well-founded statistical algorithm to get around this problem by an iterative process. First one assumes random components (randomly centered on data points, learned from k-means, or even just normally distributed around the origin) and computes for each point a probability of being generated by each component of the model. Then, one tweaks the parameters to maximize the likelihood of the data given those assignments. Repeating this process is guaranteed to always converge to a local optimum.

3. Data

Student performance dataset is the dataset from which we are trying to gain some information from 650 student's academic performance data and various attributes which are affecting the performance. Attributes include sex, age, study time, free time, family information etc.

We will be using the Logistic regerssion method to learn and predict the performance. Out of 650 I have taken 455 data entries for lerning and rest for prediction i.e testing the performance of the algorithm.(70:30). The labels or target functions are mentioned in the 'new' field which is added manually. bad(0-7), medium(8-14), good(15-20) .

Univariate gaussian Distribute

$$\mathcal{N}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

mean variance

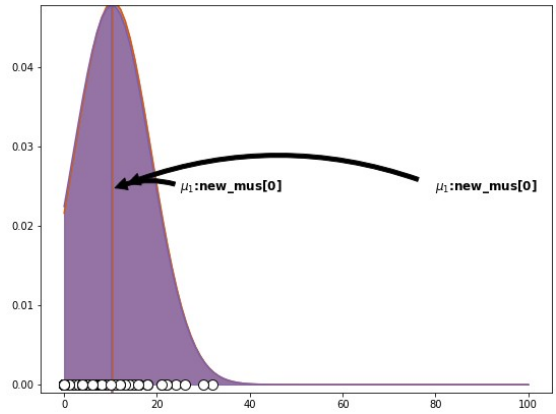
Multi-Variate Gaussian Distribution

$$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}$$

mean covariance

5. Experiments

```
Iteration:  9

                1          2
mus   11.992655  11.572048
sigs  59.231139  62.976005
log_likelihood: {ll_new:3.4f}

Iteration:  10

                1          2
mus   12.431546  11.947120
sigs  55.676539  60.338667
log_likelihood: {ll_new:3.4f}

Iteration:  11

                1          2
mus   13.029712  12.45602
sigs  50.225179  56.29549
log_likelihood: {ll_new:3.4f}

Iteration:  12

                1          2
mus   13.881852  13.180722
sigs  41.251764  49.611750
log_likelihood: {ll_new:3.4f}
```



6. Conclusion

The median in this case is overlapping. So there is no such significant difference in the factors. The factors are significant to each other. In other models we can see the GMM can produce the good clusters out of complex data and gives the probability of each data point to belong to specific cluster.

7. Referances

http://www.cse.iitm.ac.in/~vplab/courses/DVP/PDF/gmm.pdf

http://scikit-learn.org/stable/modules/mixture.html