Diego Millan – NetID: diegom3
CS410: Text Information Systems - UIUC

Technology Review

EM Algorithm

Introduction

EM algorithm, or Expectation-Maximization algorithm, has remained a popular approach for clustering data since its introduction in 1977. The general idea of this method is to be able to estimate a local maximum of parameters in a model that depend on unseen variables using a 2-step process implementation.

The first step is called Expectation or E-step in which we calculate an estimate of the unseen data based on our current model parameters.
The second step is called Maximization or M-step where we take the calculations of the E-step and come up with a better estimation of the model parameters using some observed data.

Using this new estimation, we can then update our parameters and feedback to the E-step, lending itself to be iterated until our model converges, where it would have reached a local maximum or maxima.

EM algorithm is not the only one that employs such a model as for example, K-means, works in a similar 2-step analysis. The intention of this review is to showcase the EM algorithm by comparing it to K-means.

Comparison

The EM algorithm shares similar features to other approaches of unsupervised learning algorithms such as K-means which is also very popular due to its simplicity.

In K-means, the first step is to set K centroids, K being the number of expected clusters in the vector space of our observed data. For the first iteration, this centroid can be initialized either randomly or using some knowledge of the data to help converge faster. Next, we calculate the Euclidian distance of all points with respect to the centroids and tag them based on its proximity. What we will end up with is k groups of data points representing each cluster.

The second step of the algorithm is an update step, in which we will take the data and assume they belong to the tagged cluster. This clusters will have a new centroid different from the one initially set, so we can go ahead and calculate and update them.

We can now iterate these 2 steps until our centroids converge, in other words the centroids stop meaningfully moving or not moving at all.

This all sounds pretty much the same as the EM algorithm description, but the main difference between both approaches reside in how strong the tagging of data points is. In K-means, each datapoint is tagged discretely to one of the k clusters defined, while in the EM algorithm they are soft tagged with the probability of the element belonging to each k cluster.

If we think about how this looks in practice, we can imagine that k-means tries to draw a Voronoi diagram over our vector space while the EM algorithm is building a distribution. For this very reason EM is usually a more robust solution as it can fit better different data arrangements.

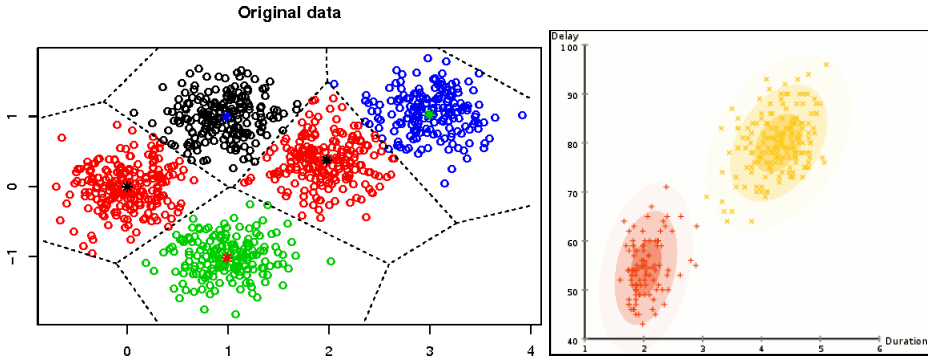

Figure 1 Voronoi diagram on K-means (left), EM algorithm distribution (right)

For starters, K-means has difficulty identifying clusters that do not behave in a spherical fashion, meaning they are not tightly close together. It also has problems with clusters of significant different sizes since in its simpler form we are just looking for distance to the centroid. This creates a division exactly halfway through said distance not considering how wide each cluster is.

EM solves this problem by introducing estimations of representative latent parameters. We can for instance calculate the mean and standard deviation of our data and get a better fit and outline of our data.

To better understand the differences, here is an example using some simple data set taken from electronic devices measurements. On the X axis we have measurements taken in a production environment while the Y axis represents the reference measurement taken on a lab setting of the same devices, in other words their true measurement.

What this plot represents is the correlation between the true and production measurement allowing to find a suitable production offset. But something is not right, the R-square of this system is very low meaning the correlation is weak, we can also see that there seem to be 2 populations that look like they increase monotonically so probably the data is being generated by 2 models.

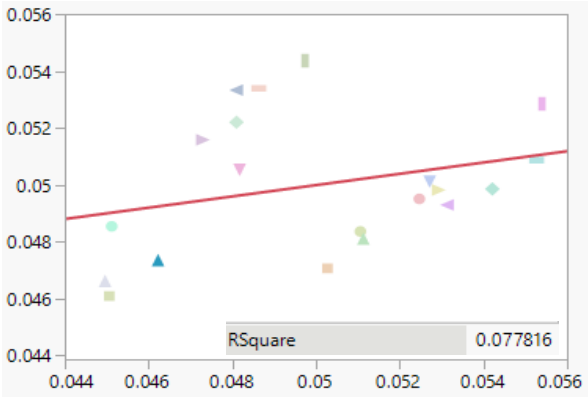

Figure 2 Low R-square due bimodality

This example was deliberately picked for K-means to perform poorly, as the data is both, not spherical and one of the populations seem slightly more compact than the other.

First step for both methods is to initialize the parameters with a current estimation, below is the data plotted with two randomly points selected as initial centroids, one centroid per cluster.

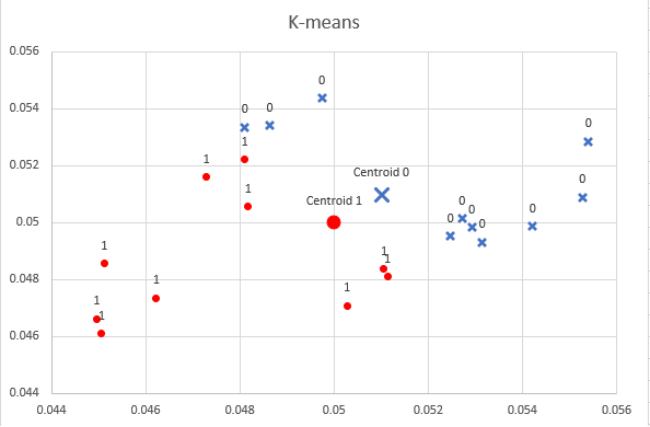

Figure 3 K-means initialization

Each iteration of K-mean we will calculate the Euclidean distances of each point to the centroids and assign a binary variable depending on their closeness.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} .$$

Since the dataset is very simple, after just a few iterations, the algorithm has converged, not too far from where we started, however, we can clearly see a problem with the results. The red centroid ended up being pulled by the 4 dispersed points on the bottom left while the blue one got pulled to the more tightly points on the center right. This caused to erroneously tag the farther up points of the left cluster to the blue cluster and one point that should belong to the blue cluster on the bottom ended up being closer to the red centroid.

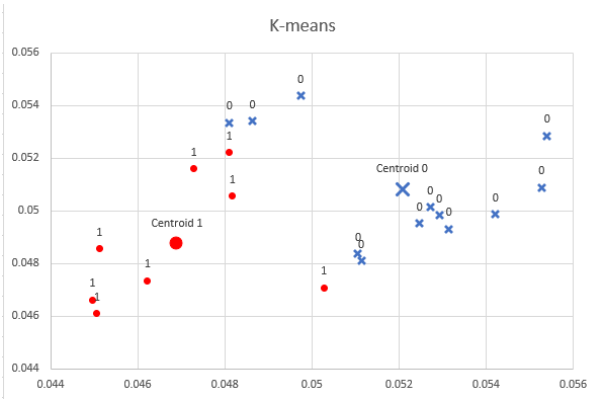

Figure 4 K-means after convergence

How does EM perform comparatively? Using the same data, let us initialize our means using the output of K-means and some appropriate standard deviation randomly. We also need a probability defining the contribution of each of the two theorized models which we will set to 0.5 each since we have the same number of parts per cluster.

Without starting to iterate, we already see different tags on the data points. To keep things simple, points were tagged by the greatest probability out of each cluster.

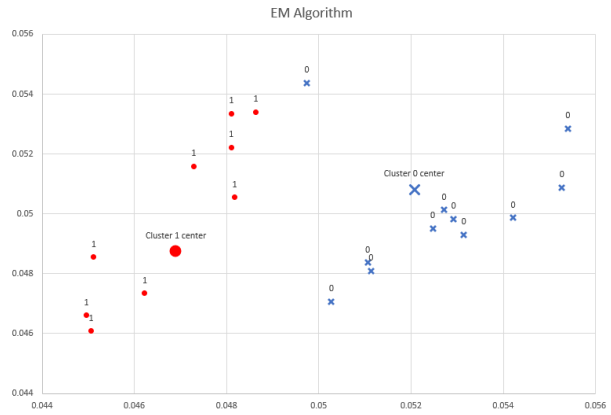

Figure 5 EM algorithm initialization

Each iteration will then calculate the probability of each data point to belong to each model following below formula and represented by ϒ, where π is the probability of a cluster, ϕ is a probability density function, μ is the mean and σ the standard deviation of each cluster.

$$\gamma_i = \frac{\pi\phi(x_i;\mu_2,\sigma_2)}{(1-\pi)\phi(x_i;\mu_1,\sigma_1)+\pi\phi(x_i;\mu_2,\sigma_2)}$$

We then use these estimates to recalculate our initial parameters.

$$\mu_2 = \frac{\sum\gamma_i x_i}{\sum\gamma_i} \qquad \sigma_2 = \frac{\sum\gamma_i(x_i-\mu_2)^2}{\sum\gamma_i} \qquad \pi = \frac{1}{n}\sum\gamma_i$$

After some iterations we converge to a way better result, in fact perfect for this data set and starting parameters.

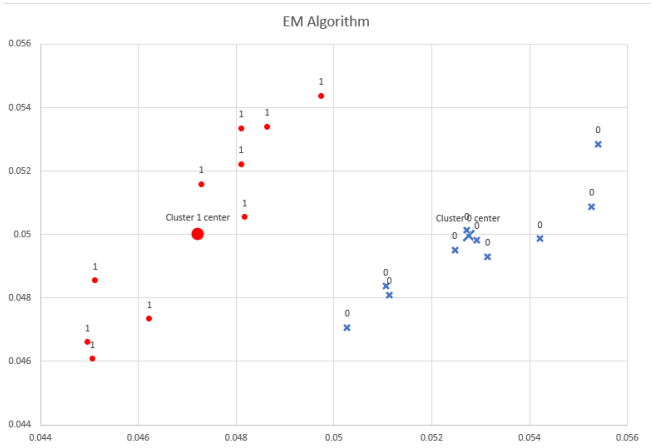

Figure 6 EM algorithm converged

Conclusion

EM algorithm is a versatile approach for clustering data that can fit a wide variety of behaving data and is not as limited as K-means to well defined clusters. Although one must weight in what conclusions we expect to draw from the data and what our system is like, as implementing EM is significantly more complex and expensive. Even in the small and simple set presented here, K-means converged after just a few iterations(less than 5) while EM even with the head start of using K-means out put as initial values, it needed several more iterations (more than 10) to reach the maxima.

References

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological), 39*(1), 1-38. Retrieved November 15, 2020, from http://www.jstor.org/stable/2984875

Charan, R. (2020, July 11). *Expectation Maximization Explained.*
Retrieved from https://towardsdatascience.com/expectation-maximization-explained-c82f5ed438e5

Wolfram MathWorld. (1999) *Distance*
Retrieved from https://mathworld.wolfram.com/Distance.html

Wolfram MathWorld. (1999) *Probability Density Function*
Retrieved from https://mathworld.wolfram.com/ProbabilityDensityFunction.html

Wikipedia (2020, November 5). Expectation-Maximization Algorithm.

Retrieved from https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

Klopotek, R.A., & Kłopotek, M. (2017). On the Discrepancy Between Kleinberg's Clustering Axioms and k-Means Clustering Algorithm Behavior. ArXiv, abs/1702.04577.