Estimating the Number of Clusters in a Dataset

Robert Capo
Electrical and Computer Engineering
Rowan University
Glassboro, NJ 08028
Email: robcapo@gmail.com

***Abstract*—Clustering is a popular yet difficult task in machine learning. In the clustering task, the user knows very little information about the data (i.e. membership classes, variability, underlying distributions) and needs to group observations into clusters. Many algorithms exist that constrain the problem and attempt to find optimal solutions. These algorithms usually require the selection of a parameter, and a common choice is K, the number of clusters in the dataset. However, the user may not know the true value of K, so an exhaustive search can be used. We attempt to compare various statistical methods and introduce a heuristic method that is computationally efficient and works well when some general assumptions are held.**

I. Introduction

Unsupervised learning is a popular topic in the machine learning community. The unsupervised scenario is encountered when the learner is tasked to extract meaningful information after being provided with only observations from a dataset. This contrasts supervised learning, in which the learner is also provided with meaningful information for some of the observations and attempts to extend the information to unknown observations drawn from the same distribution. This meaningful information can be a variety of different things. In classification (supervised) and clustering (unsupervised), the information of interest is a set of grouping variables for each of the observations. Observations that are most related to each other should be assigned to the same group (or cluster).

Such a problem is extremely popular, and many clustering algorithms have been developed to solve it [1]–[4]. Many algorithms follow a common paradigm; declare an objective function, $O(X, \theta)$ that measures the goodness of fit for the assigned clustering and choose the clustering that optimizes this function. Here we denote X as an N x d matrix containing N observations in d dimensions and θ as a free parameter that affects the way that the data is clustered (number of clusters, average spread of the data, etc). The objective function can take on different forms, which affect the way it should be optimized. For example, an objective function that measures the likelihood of the assigned clustering would be maximized, while an objective function that measures the error would be minimized.

Assuming the appropriate objective function is chosen, a reasonable approach to finding the best clustering would be to find the solution that analytically maximizes (or minimizes) this function. However, objective functions with closed-form analytical solutions are very difficult to define. The reason, of course, is that the optimal clustering depends on the appropriate value of θ, but there are many possible clusterings for each value of θ. The problem of finding C_i^*, the optimal clustering, can be defined as

$$C_i^* = \underset{i}{arg\,max} L(X, C_i(\theta^*))$$

$$\theta^* = \{\theta : C_i(\theta) = C_i^*(\theta)\}$$

where θ^* is the optimal selection for the parameter.

Since there's no closed form analytical solution, another option is to test all possible values of θ and choose the one which optimizes $O(X, \theta)$. However, for datasets of reasonable size, this option becomes computationally infeasible. To illustrate, let's assume the free parameter is K, the number of clusters in the dataset. This is perhaps the easiest choice of θ, because K is restricted to discrete positive integers between 1 and N. The number of possible clusterings becomes

$$|C| = \sum_{K=1}^{N} \frac{N!}{k!(N-K)}$$

Where $C = \{C_i \forall i\}$ is the set of possible clusterings. For a dataset of size 10, $|C|$ is $1,023$. For a dataset of size 100, $|C|$ becomes $1.2677 * 10^{23}$. Datasets with more than $10,000$ observations are not uncommon, so it quickly becomes evident why an exhaustive search is infeasible.

The rest of this document describes our methodology, results, and conclusions for estimating the optimal number of clusters (and therefore the optimal clustering) in a dataset. Section III goes through various options and explains the types of data that are more suited for each option. Section IV describes our experimental results on some preliminary test datasets. Section V draws conclusions from our results and introduces some avenues for future work.

II. Methodology

A. Expectation Maximization for Gaussian Mixture Models

To this point, we've shown that even if the appropriate objective function is chosen, the optimal solution is very difficult or impossible to find globally. To solve this problem, clustering algorithms make more assumptions that will further constrain the problem and reduce the number of possible solutions. A commonly used algorithm is the expectation maximization procedure for Gaussian Mixture Models (EMGMM) [3].

EMGMM makes the assumption that the data distribution is represented by a mixture of Gaussian distributions. When used for clustering, we assume that every Gaussian in the mixture represents one cluster. The PDF of the model becomes

$$p(\theta) = \sum_{k=1}^{K} p_k \mathcal{N}(\theta_k)$$

where $\theta_k = \{\mu_k, \Sigma_k\}$ defines the parameters of each Gaussian or mode in the mixture. EMGMM requires a specified K, and uses expectation maximization to find the parameters of K modes that maximize the likelihood of the model. The objective function becomes

$$O(X, C_i(k)) = p(X|\theta) = \prod_{j=0}^{N-1} \sum_{k=1}^{K} p_k p(x_j|\theta_k)$$

EMGMM is an iterative two-step procedure that continues until convergence is reached or the maximum number of iterations is reached. The expectation step calculates the probability that distribution θ_k is explained by observing instance x_i. This is often called the membership weight and is defined as

$$p(\theta_k|x_i) = \frac{p_k p(x_i|\theta_k)}{\sum_j p_j p(x_i|\theta_j)}$$

for all k, i.

The maximization step then maximizes the likelihood that the mixture model represents the data based on the current membership weights

$$\mu_k = \frac{\sum_i p(\theta_k|x_i) x_i}{\sum_i p(\theta_k|x_i)}$$

$$\Sigma_k = \frac{\sum_i p(\theta_k|x_i)(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i p(\theta_k|x_i)}$$

$$p_k = \frac{\sum_i p(\theta_k|x_i)}{N}$$

The EMGMM algorithm works fairly well when K is known, but it can converge to the wrong solution if the initial conditions are chosen incorrectly. Furthermore the likelihood of the model generally increases as K increases, so it is unfair to compare two models with different K, as the optimal model would generally appear to be the one with larger K.

B. Information Criteria

In order to compensate the overfitting problem, where GMMs with large K generally have a higher likelihood, some information criteria have been developed to penalize models with large K. Two information criteria used in our experiments are the Akaike Information Criterion (AIC) [5] and the Bayes Information Criterion (BIC) [6].

The AIC is the first information criterion studied and is defined as

$$AIC = 2K + 2\ln(L)$$

where L is the likelihood of the model and K is the number of free parameters or modes in the GMM. And the model with minimum AIC is best suited to describe the data. Minimizing AIC maximizes the likelihood while also penalizing models with many free parameters.

Unfortunately, AIC does not work well in our problem of clustering large datasets. As the size of the dataset increases, more probabilities must be multiplied in the likelihood function, so it decreases, thereby increasing $-\ln(L)$. When $-\ln(L)$ becomes very large, the penalty for increasing K becomes negligible and minimizing AIC is synonymous to maximizing the likelihood. The BIC remedies this problem by scaling the penalty for K with the size of the data. The BIC is given by

$$BIC = -2\ln(L) + K\ln(N)$$

where N is the number of observations and the model with minimum BIC is the best. Scaling the penalty by the size of the dataset allows it to increase as the negative log likelihood increases. Figure 3 shows a comparison of the negative log likelihood, AIC, and BIC for a large dataset. The optimal number of clusters in the dataset is 16.

C. Gap Statistic

Tibshirani, Walther, and Hastie propose the gap statistic for estimating the number of clusters in a dataset [7]. They use the k-means clustering algorithm [2] and sweep from $K = 1$ to N. The objective function

$$O(X, C_j(k)) = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

is defined as the pooled sum of square errors around the cluster means, where

$$D_r = \sum_{i,j \in C_j(k)} d_{ij}$$

is the sum of the pairwise distances of all instances in cluster $C_j(k)$ of size n_r. Of course, this function will decrease as K increases, so the same problem of overfitting exists. However, the authors notice that there is a value of K at which an elbow occurs. At this point, the error function decreases much more slowly.

In order to find the elbow, the authors compare the measured objective function $O(X, C_j(k))$ against the expected value of the objective function $O(X', C_j^*(k))$ on a reference distribution X' for which increasing k is known to be unnecessary. As long as increasing k is useful, the measured objective function should decrease faster than the reference objective function. Therefore, the gap statistic is defined as

$$Gap_n(k) = E_n^*\{\log(O(X', C_j^*(k)))\} - \log(O(X, C_j(k)))$$

The elbow occurs at the maximum value of this function. The authors and references therein express the importance of

choosing the appropriate reference distribution and use the uniform distribution in their experiments. They show that the expected value, assuming uniformly spaced clusterings in k-means, is

$$\log(\frac{dN}{12}) - \frac{2}{d}\log(K) + constant$$

where d is the dimensionality of the data. Figure 4 shows the measured objective function, reference objective function, and gap statistic for a dataset with an optimal K of 16.

D. Heuristic Method

We tested one final heuristic method that relies on hierarchical agglomerative clustering to find the number of clusters in X. Hierarchical clustering is a non-parametric clustering method that sweeps K from N (every instance is a cluster) to 1 (every instance in the same cluster) and sequentially joins the nearest two clusters at each step. The resulting information is a set of N possible clusterings and a set of distances (D_{ij}'s) at which each cluster was joined.

$$D_{ij} = \frac{1}{|c_i| * |c_j|} \sum_{x_i \in c_i} \sum_{x_j \in c_j} d_{ij}$$

This choice of D_{ij} is known as the unweighted pairwise group method with arithmetic mean (UPGMA) for hierarchical clustering. D_{ij} measures the distance between any two clusters as the average of all pairwise distances between all instances in each cluster. The advantages and disadvantages of choosing different distance metrics and a general analysis of hierarchical clustering are found in [4].

When clusters are being joined unnecessarily, the D_{ij} has a sudden increase comparable to the elbow for the gap statistic, except the function increases as K decreases, so the elbow is facing the opposite direction. In order to find the elbow, we took the differential distance

$$\Delta(k) = D_{ij}(k) - D_{ij}(k+1)$$

and find the percentage that it increases from the average of all distances that occurred earlier

$$\Delta'(k) = \frac{\Delta(k) - \frac{1}{N-k+1}\sum_{i=k+1}^{N}\Delta(i)}{\frac{1}{N-k+1}\sum_{i=k+1}^{N}\Delta(i)}$$

We find the maximum of $\Delta'(k)$ and set

$$k^* = \arg\max_k \Delta'(k) + 1$$

Figure 5 shows how finding $\Delta(k)$ and $\Delta'(k)$ effectively finds k^*.

E. Finding k^*

There are several different ways to find k^*, the number of clusters in the dataset. We can observe the values in our objective function for all values of k, and choose the one that maximizes the function. This may be computationally infeasible, especially for algorithms like EMGMM. When an exhaustive search is infeasible, numerical methods like Newton-Raphson and gradient descent are used. However, as we've already seen, our objective function is not actually differentiable and many local maxima exist. The final method that we used in several experiments is to create a range of values for k and find the optimum in that range. Of course, the user has the option to trade off computational complexity with probability of finding the true global optimum.

III. Results

Our experiments consisted of two types of datasets. The first type is a uniform grid of Gaussians with reasonable separability. The means are linearly spaced between 3.5 and $3.5\sqrt{(K)}$, and the covariances are $2.5I$. This experiment can be seen in Figure 1.

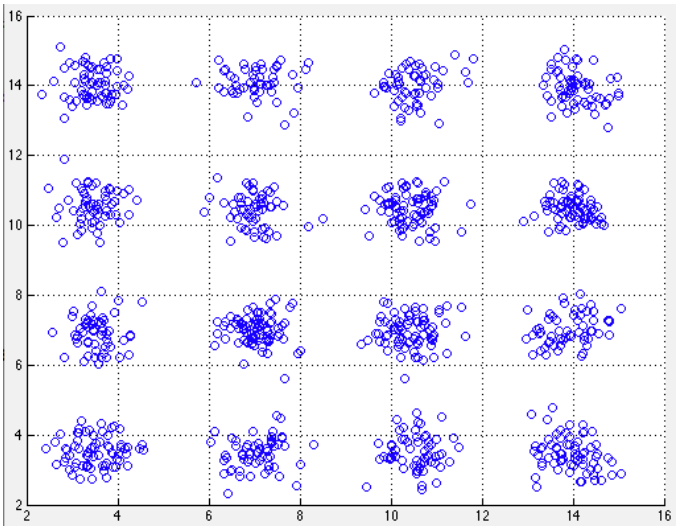

Fig. 1. Experiment 1, Mixture of uniformly spaced Guassians $K = 16$

The second experiment consisted of another mixture of Gaussians, but with randomly spaced means. The means are drawn from a uniform distribution between 0 and $10\sqrt{(K)}$ and the covariances again are $2.5I$. Experiment 2 can be seen in Figure 2.

Figure 3 shows a comparison of AIC, BIC, and $-\log(L)$ in experiment 1. We can clearly see how unpedictable the $-\log(L)$ function is, as there are many local maxima. We can also see that the AIC doesn't have any noticeable advantage over minimizing $-\log(L)$, while BIC's penalty for large K is effective enough that the appropriate value for K is selected.

Figure 4 shows the gap statistic in a search between 10 and 30. The true value, 16 is found by maximizing $Gap(k)$.

Figure 5 shows how the heuristic method finds k^* using the differential distance of cluster joins.

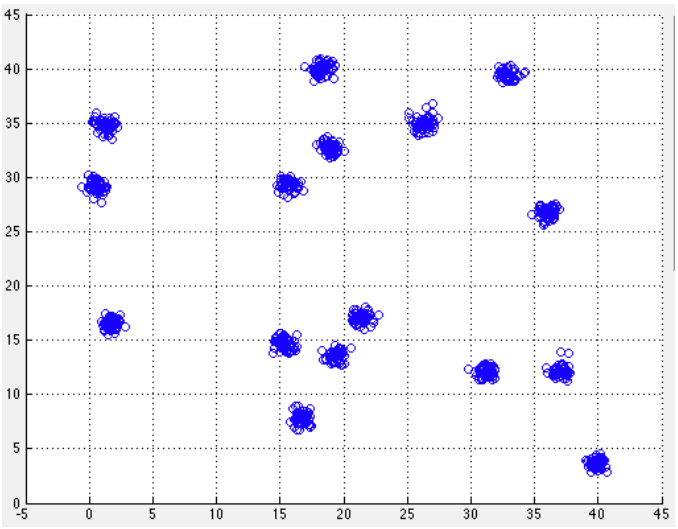

Fig. 2. Experiment 2, Mixture of randomly spaced Guassians

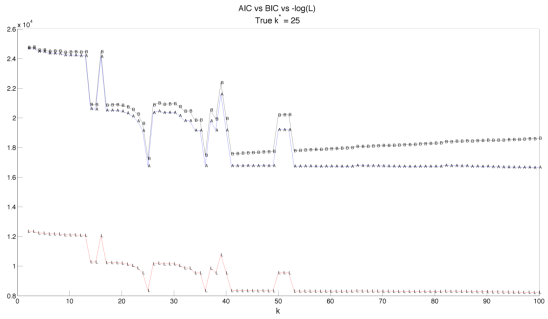

Fig. 3. Information criteria for experiment 1 with $K = 16$

IV. Conclusion

In conclusion, different methods have different advantages and disadvantages associated with them. Statistically proven methods are obviously advantageous by nature, but can still be unpredictable due to the sensitivity to the initial conditions of EMGMM. The information criteria and gap statistic are two viable methods for analyzing the likelihood function and their computational complexity is negligible compared to that of EMGMM. The gap statistic is desirable in its generality, as choosing a different reference distribution and clustering method provides versatility.

Our heuristic method worked well in our experiments, but it could certainly be extended and stress tested further. In addition, the heuristic method is significantly faster than fitting GMMs for all possible values of k. One option for a parameter of this method would be the number of previous clusterings to consider for the average in Δ'. This would further reduce the possibility for joins with large D_{ij} for very small K from appearing as the maximum of $\Delta'(k)$.

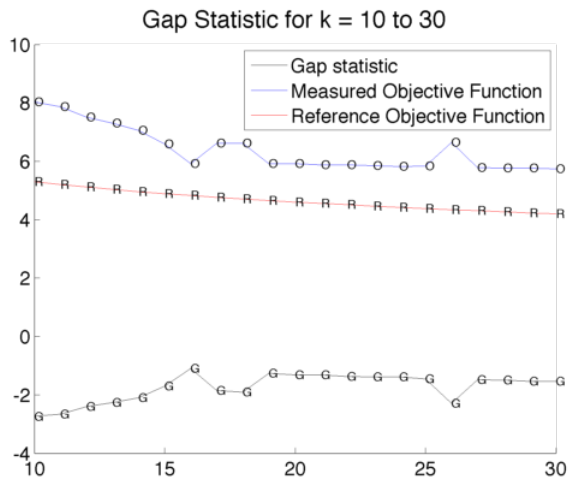

Fig. 4. Gap statistic calculated for experiment 1 with $K = 16$

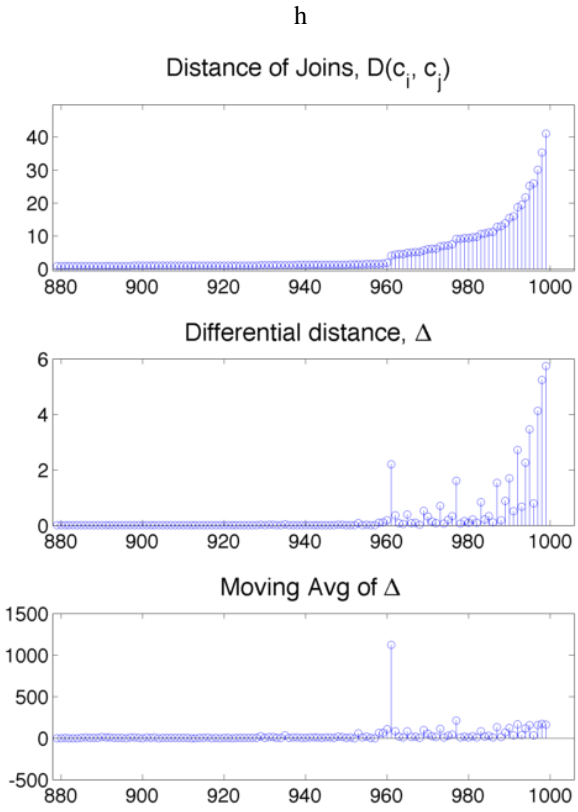

Fig. 5. Heuristic method's for experiment 2 with $K = 16$

References

[1] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1, 1967, pp. 281–297.

[3] D. A. Reynolds, "Gaussian mixture models." in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Springer US, 2009, pp. 659–663. [Online]. Available: http://dblp.uni-

trier.de/db/reference/bio/g.html#Reynolds09
[4] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963. [Online]. Available: http://www.jstor.org/stable/2282967
[5] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, Dec. 1974. [Online]. Available: http://dx.doi.org/10.1109/tac.1974.1100705
[6] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978. [Online]. Available: http://dx.doi.org/10.2307/2958889
[7] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the gap statistic," vol. 63, pp. 411–423, 2000.