Comparing K-value Estimation for Categorical and Numeric Data Clustering

K.Arunprabha
M.C.A.,M.Phil
Assistant Professor,
Department of Computer science,
Vellalar College for Women, Erode-12.

V.Bhuvaneswari
M.Sc., M.Phil
Research Scholar
Vellalar College for Women, Erode-12.

ABSTRACT
In Data mining, Clustering is one of the major tasks and aims at grouping the data objects into meaningful classes (clusters) such that the similarity of objects within clusters is maximized, and the similarity of objects from different clusters is minimized. When clustering a dataset, the right number k of clusters to use is often not obvious, and choosing k automatically is a hard algorithmic problem. We used an improved algorithm for learning k while clustering the Categorical clustering. A Clustering algorithm Gaussian means applied in k-means paradigm that works well for categorical features. For applying Categorical dataset to this algorithm, converting it into numeric dataset. In this paper we present a Heuristic novel techniques are used for conversion and comparing the categorical data with numeric data. The G-means algorithm is based on a statistical test for the hypothesis that a subset of data follows a Gaussian distribution. G-means runs in k-means with increasing k in a hierarchical fashion until the test accepts the hypothesis that the data assigned to each k-means center are Gaussian. G-means only requires one intuitive parameter, the standard statistical significance level α.

Keywords: Data mining, Clustering Algorithm, Categorical data, Gaussian Distribution

1. INTRODUCTION
As a statistical tool, clustering analysis has been widely applied in a variety of scientific areas such as pattern recognition, image processing, information retrieval and biology analysis. In the literature, the k-means is a typical clustering algorithm, which partitions the input data set {Xt}Nt=1 that generally forms k¤ true clusters into k categories (also simply called clusters without further distinction) with each represented by its center. Although the k-means has been widely used due to its easy implementation, it exists a serious potential problem. That is, it needs to pre-assign the number k of clusters. Many experiments have shown that it can work well only when k is equal to k*. However, in many practical situations, it is hard or becomes impossible to know the exact cluster number in advance. Under the circumstances, the k-means algorithm often leads to a poor clustering performance.

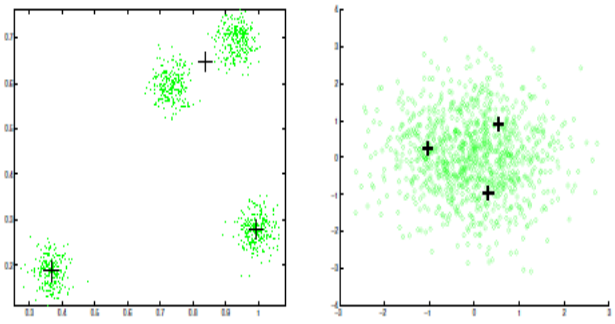

Figure 1: Two clustering's where k was improperly chosen for the dataset being clustered. Dark crosses are k-means centers. On the left, there are too few enters; four should be used. On the right, too many centers are used; one center is sufficient for representing the data.

In this paper we present a simple algorithm called G-means that discovers an appropriate k using a statistical test for deciding whether to split a k-means center into two centers. We present a new statistic for determining whether data are sampled from a Gaussian distribution, which we call the G-means statistic. We also present a new Heuristical novel method for converting categorical data into numeric data. We describe examples and present experimental results that show that the new algorithm is successful [7]. This technique is useful and applicable for many clustering algorithms other than k-means, but here we consider only the k-means algorithm for simplicity. Several algorithms have been proposed previously to determine k automatically.

2. CONVERTING CATEGORICAL DATA
Converting Categorical data into numeric data by using these techniques[5].

Definition 1: Let rij for j = 1,..., ai be an element in the attribute Ai. According to the statement above, rij is converted to a 1-by-l vector

$$\mathbf{V(r^i_j) = [}\mathit{IGR}(A_i)\ \mathit{WCI}(r^i_j)\ \mathbf{p^i_{jk}]_{k=1...l}}$$

***Definition 2*:** Let $O = (c_1, c_2, \ldots c_m)$ be an object in the set D, where $c_i = r^i_j$ for $i = 1,...,m$, and $j = 1,..., a_i$. Assume attributes are independent. Then from Def. 1, O is converted to a vector,

$$V = \sum_{i=1}^{m} v(c_i = r^i_j)$$

Definition 3 Let O and O' are distinct objects from the set D Where $O = (c_1 , c_2 , \ldots, c_m)$

and $O' = (c'_1, c'_2, \ldots, c'_m)$. Following Def. 2, O is converted to V, and O' is converted to V'. The pseudo distance between O and O' is defined by using Euclidean distance:

$$d(O, O') = 2 \|V - V'\|_2$$

So far, we have formally constructed the framework of dissimilarity measure between categorical data. In summary, the proposed clustering process involves three phases:

1. Firstly, we need to estimate the information from interattributes and intra-attributes. This estimation can be obtained from a domain expert or by using a training set of objects.

2. Secondly, we heuristically convert each attribute using the associated information. Therefore each object in the dataset is converted numerically with reasonable concept.

3. Finally, traditionally clustering algorithms can be exploited effectively.

3. G-MEANS ALGORITHM

The Gaussian-means (G-means) algorithm starts with a small number of k-means centers, and grows the number of centers. Each iteration of the algorithm splits into two those centers whose data appear not to come from a Gaussian distribution. Between each round of splitting, we run k-means on the entire dataset and all the centers to refine the current solution. We can initialize with just k = 1, or we can choose some larger value of k if we have some prior knowledge about the range of k. G-Means algorithm gives a pseudo code description [6].

Algorithm 1:

Inputs to the algorithm a dataset X and a confidence level α.

Output as Clustering the dataset.

G-means(X, α)

1. Let C be he initial set of centers (usually C← { $\bar{X}$ }).
2. C←*kmeans*(C, X).
3. Let $\{x_i | \text{class}(x_i)=j\}$ be the set of datapointsassigned to center c_j.
4. Use a statistical test to detect if each $\{x_i | \text{class}(x_i)=j\}$ follow a Gaussian distribution (at confidence level α).
5. If the data look Gaussian, keep c_j. Otherwise replace c_j with two centers split from c_j.
6. Repeat from step 2until no more centers are needed.

G-means repeatedly makes decisions based on a statistical test for the data assigned to each center. If the data currently assigned to a k-means center appear to be Gaussian, then we want to represent that data with only one center. However, if the same data does not appear to be Gaussian, then we want to use multiple centers to model it properly. The algorithm will run k-means multiple times (up to k times when finding k centers), so the time complexity is at most O(k) times that of k-means. An optimization we make in the G-means algorithm is that once we have decided not to split a center cj , we do not test the data belonging to that center again. This enables us to make k statistical tests when finding k centers, rather than up to O(k2) tests if every center is tested at every iteration of G-means(in the worst case scenario). The k-means algorithm implicitly assumes that the data points in each cluster are spherically distributed around the center.

4. TESTING CLUSTERS for GAUSSIAN FIT

To specify the G-means algorithm fully we need a test to detect whether the data assigned to a center are sampled from a Gaussian. The alternative hypotheses are:

• H0: The data around the center are sampled from a Gaussian.
• H1: The data around the center are not sampled from a Gaussian.

If we accept the null hypothesis H0, then we believe that the one center is sufficient to model its data, and we should not split the cluster into two sub clusters. If we reject H0 and accept H1, then we want to split the cluster. In this work we have utilized two tests for normality. Both are one dimensional test which assumes that the data has been z-scored; that is, converted to mean 0 and variance 1. The first is based on a new statistic we call the G-means statistic [3]. This statistic comes from the distortion of the data, defined as

$$r(X, C) = \sum_{i=1}^{|X|} \min_j ||x_i - c_j||^2$$

Where C = {c1, . . . , ck} is the set of k centers. The G-means statistic uses a specific formulation of r(X,C), under constraints of univariate data with k = 2 centers. Specifically, given a one-dimensional set of data and two k-means clusters

$$r_{gm}(X) = \min_{c_1,c_2} \sum_{i=1}^{|X|} \min((x_i - c_1)^2, (x_i - c_2)^2)$$

This is the minimum of the k-means objective function for two centers in Gaussian data (when the null hypothesis is true).

The second test is based on the Anderson-Darling statistic. This one dimensional test has been shown empirically to be the most powerful normality test that is based on the empirical cumulative distribution function (ECDF).

Given a list of values xi that have been converted to mean 0 and variance 1, let x(i) be the ith ordered value. Let zi = F(x(i)), where F is the N(0, 1) cumulative distribution function. Then the statistic is

$$A^2(Z) = -\frac{1}{n} \sum_{i=1}^{n} (2i - 1) [\log(z_i) + \log(1 - z_{n+1-i})] - n$$

Stephens [9] showed that for the case where μ and σ are estimated from the data (as in clustering), we must correct the statistic according to

$$A^2_*(Z) = A^2(Z)(1 + 4/n - 25/(n^2))$$

With these two statistics, and their respective distributions, we will construct statistical tests for normality which will be used in the G-means algorithm [6]. Both these tests are one dimensional test. We have a high dimensional dataset; we reduce the dimensions using dimension reduction method, so we learning true dimension under PCA method.

5. LEARNING TRUE DIMENSION FROM CLUSTERING

We now turn to the general problem of dimension reduction. Many real-world datasets have a high number of dimensions, and in order to work with them it is often beneficial to reduce the dimension of the data prior to using learning algorithms. This is effective because often the structure of the data may be described in far fewer dimensions, and most learning algorithms perform best when the dimension is low. What we would like is an automated way of learning the underlying dimension of high-dimensional data. This is a well-researched area; we approach it from a slightly different perspective. We suppose that we have a black-box algorithm which can tell us how much "structure" exists in a dataset. We will then use this black box in a generate-and-test fashion to repeatedly determine how much structure exists in various reduced-dimension datasets. Starting with a small number of dimensions and increasing, we will look for the point at which no more structure can be discovered in the dataset by increasing the dimension. This critical point will be the true dimension of the data[7].

The intuition behind this algorithm is that when data is reduced to some dimension that is too small, structure that can be found in the original data must necessarily be collapsed, and unavailable. However, as the number of dimensions increases, more structure will unfold to be discovered. If there is some lower dimensional space in which the full structure can be represented, then we can identify that space using our black box. This is related to the reconstruction of dimension in chaotic systems by identifying false nearest neighbors [7] (in chaotic systems parlance, the dimension is actually the number of time steps into the past to observe). Our algorithm for learning dimension is given in Algorithm 2.

To completely define this algorithm, we must fill in a dimension reduction method as well as the black box for measuring structure. To experiment with this technique, we will use single dimension reduction techniques: principal components analysis (PCA), and random linear projection. To measure the structure of the data, we will use the G-means algorithm to find clusters in the data. The number of clusters that it estimates are in the data will be the metric of structure.

We performed several synthetic and real-world experiments to test our dimension reduction algorithm. We generate two datasets: a random dataset with 20 true spherical clusters in 20 dimensions, and a second dataset in the same way, but we add 20 dimensions (for a total of 40) which have uniform noise. We then apply our algorithm, using PCA or random linear projection to reduce the dimension, and using this algorithm.

Algorithm 2: Input of the Algorithm is X as high dimensional dataset.

Output as Single Dimension result.

Learn dimension(X)

1. Let d be the original dimension of X
2. for c € {1,…,d}do
3. Y←reduce-dimension(X,c)
4. S_c←compute-structure(Y)
5. end for
6. return min{c | s_c=max(s)}

G-means is to rank the structure of each reduced-dimension dataset. We used the Anderson-Darling test for Normality[1]

6. CLUSTERING CATEGORICAL DATA

For Clustering Categorical attribute, there are many algorithms used for clustering such as ROCK, BRICH, CATCUS etc. These algorithms are clustered on link based method[10]. In this paper we present G-Means algorithm applied in K-means for categorical attribute. The categorical attribute is converted into numeric data by using Heuristic novel method. Then the converted data is applied in Gaussian distribution as sampled data. Then it is applied in G-Means. Under G-means using dimensionality reduction method to reduce the dimension and testing using Anderson Darling Normality test. It under goes the condition H0 or H1.Finally we can estimate accurate value of K for clustering the categorical data.

We take various categorical dataset such as Congressional votes, Soybeans, Balance scale, and Car evaluation dataset [3]. Congressional Votes data set is the United States Congressional voting records in 1984. Total number of records is 435. Each row corresponds to one Congress mans votes on 16 different issues (e.g., education spending, crime etc.). The data set contains records for 168 Republicans and 267 Democrats. Balance-Scale is a weight and distance Database. The Balance-Scale data set contains 625 data points. Each data point has 4 categorical attributes. The clustering the information about the attribute of this data set is Left-Weight, Left-Distance, Right-Weight and Right-Distance. Attributes are given in numerical from such as 1 to 5. The Soybean data set contains 47 data points on diseases in soybeans. Each data point has 35 categorical attributes and is classified as one of the four diseases. Car Evaluation dataset consists of 1728 instances. All instances completely cover the attribute space. Out of these 7 features last one is a class identifier. Others are corresponding to the Buying (vhigh, high, med, low.), Maintenance (vhigh, high, med, low.), Doors (2, 3, 4, 5 more.), Persons (2, 4, more.), Lug boot (small, med, big.), Safety (low, med, high.). Class identifier has four distinct members, those are unacc, acc, good and very good.

Dataset	Data points	K found	Elapsed time (sec)
Congressional votes	435	2+1.70	0.0378
Balance scale	625	3+0.03	0.0527
Soya beans	47	4+0.00	0.0523
Car Evaluation	1728	4+0.01	0.0529

Table 1 Result of Categorical Data Clustering

7. COMPARISON WITH CATEGORICAL and NUMERIC DATASET

In K-means algorithm only the numeric dataset is applicable. Here we convert the categorical data into numeric data and apply in K-means using the K-value learned from G-means.

The numeric dataset Pendigit is applied and the result shows with elapsed time. Comparison based on elapsed time of both categorical and numeric dataset.

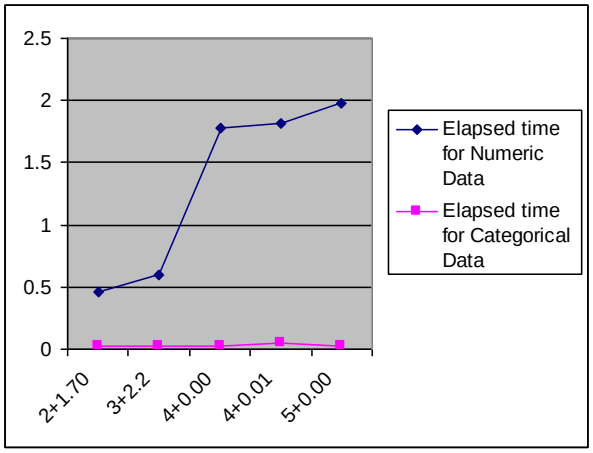

Figure 2. Performance analysis based on Elapsed Time

8. CONCLUSION

The new G-means algorithm is for learning k based on a statistical test for determining whether data points are a random sample from a Gaussian distribution with arbitrary dimension and covariance matrix. We conclude that the G-means algorithm is applied in k-means algorithm for estimating the K value accurately for both numeric value and also a categorical value. By comparing the performance of elapsed time between numeric data and categorical data, categorical data produces the best result. Our future intention is to clustering the mixed numerical dataset and categorical dataset using this technique.

REFERENCES

[1]. "Anderson-Darling: A Goodness of Fit Test for Small Samples Assumptions", START,Vol .10,No.5.

[2]. Ahmed M. Sultan Hala Mahmoud Khaleel., "A new modified Goodness of fit tests for type 2 censored sample from Normal population"

[3]. Blake. C.L. and Merz. C.J. " UCI repository of machine learning databases",1998.

[4]. Chris Ding, Xiaofeng He, Hongyuan Zha, and Horst Simon. "Adaptive dimension reduction for clustering high dimensional data".In Proceedings of the 2nd IEEE International Conference on Data Mining, 2002.

[5]. Dongmin Cai, and Stephen S-T Yau, "Categorical Clustering By Converting Associated Information" International Journal of Computer Science 1;1 2006.

[6]. Greg Hamerly,Charles Elkan, "Learning the k in k means"

[7]. Gregory James Hamerly,"Learning structure and concepts in data through data clustering". 2001.

[8]. Jain,A.K., Murty. M. N., and Flynn. P. J. "Data clustering: a review". ACM Computing Surveys, 1999.

[9]. Stephens. M.A. "EDF statistics for goodness of fit and some comparisons". American Statistical Association, September 1974.

[10]. Zhang. Y. , Fu. A, Cai. C. and Heng. P., "Clustering categorical data" 2000.

[11]. Zhexue Huang, "Extensions to the K-means algorithm for clustering Large Data sets with categorical value", 1998.