3D-2

Using a Dynamic Threshold for Clustering of Nominal Data

Sutheera Puntheeranurak[†]
Graduate School of Engineering
Tokai University[†]

Hidekazu Tsuji[‡]
School of Information Technology and Electronics
Tokai University[‡]

1. Introduction

Clustering is an unsupervised learning process that partitions data such that similar data items are grouped together in sets referred to as clusters. This activity is important for condensing and identifying patterns in data. Despite the substantial effort invested in researching clustering algorithms by the data mining community, there are still many difficulties to overcome in building clustering algorithms. Indeed, as pointed in [1] “there is no clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets”. This situation has generated a variety of clustering techniques broadly divided into hierarchical and partitional; also, special clustering algorithms based on a variety of principles, ranging from neural networks and genetic algorithms, to tabor searches.

Data clustering is one of the key tools of data mining and numerous effective clustering methods have been developed. However, there are still many challenges involved in using clustering for mining massive databases. Some of those issues include:

- Dealing with high dimensional data.
- Scalability with respect to large databases.
- Clustering nominal and mixed data in addition to numerical data.

We are developing optimization based methods that can be applied to nominal data, that is, to data whose attributes have no particular natural ordering. In general clustering, objects to be clustered are represented as points in an n-dimensional space $\Re^n$ and standard distances, such as the Euclidean distance is used to evaluate similarity between objects. For objects whose attributes are nominal (e.g., color, genre, shape, etc.), no such natural representation of objects is possible, which leaves only the Hamming distance as a dissimilarity measure, a poor choice for discriminating among multi-valued attributes of objects.

2 Clustering and center concept

Clustering aims to group similar objects into clusters which can be described by theirs centers. Each object is described by a set of attributes. Each attribute has a domain definition and takes a value in this domain.

The K-means algorithm is one of the most famous algorithms for clustering [2]. We present the classical algorithm dedicated to numerical data, then we introduce a definition of the center concept more adapted to nominal data. We also introduce the associated distance

2.1 The K-means algorithm

The K-means algorithm is an iterative procedure where iteration is given below:

Algorithm 1.

Input: the number of clusters k and item attributes
Output: a set of k clusters that minimizes the squared-error criterion.
(i) Decide on the value of K.
(ii) Start off with K arbitrary centers. They may be chosen randomly, or as the cluster center of arbitrary starting partitions of the case set.
(iii) Consider each case in sequence; find the centre to which the case is closest. Assign the case to that cluster. Recalculate the centre of the new and old clusters as the cluster center of the points in the cluster.
(iv) Repeat until the clusters are stable.
(v) Repeat for different initial centers. Choose the best clustering, in terms of minimum within cluster sum of squares.

Fig. 1. The K-means Clustering

The major drawback of K-means algorithm is that it often terminates on a local optimum and works only on numerical values because it minimizes a cost function calculating the means of clusters. Moreover, it needs to compute centers. The center of a cluster is easy to define on numerical values because the mean makes sense, but for nominal data it is not so simple.

2.2 Aggregate different data type

In this section, we can aggregate mixed type of data (multivariate data) up to N dimension. Here there are binary, nominal, ordinal, and quantitative measurement scales. The goal is to transform this data into aggregated distance matrix.

Here is step by step on how to aggregate multivariate distances:

1. Convert data into coordinate based on measurement scale

   To transform the data into coordinate, we need to consider each features data type.

If the data is quantitative, we don't need to change anything.

If the data is binary, we convert it into 0 and 1.

If the data is ordinal, we get the rank and normalize the rank into range [0, 1].

If nominal data and data is mutually exclusive values then find number of dummy variable by Eq. 1 and convert data into coordinate

$$dv = \left\lceil \frac{\log c}{\log 2} \right\rceil \qquad \text{Eq. 1}$$

Where a number of dummy variables are dv, c is a number of categories.

If nominal scale with multiple choices then assign each value of category into a single binary variable.

2. Determine distance matrix for each features variable based on coordinate
3. Normalize the distance matrix into range of [0, 1]
4. Aggregate the distance matrix

Once we get the distance matrix, we use the distance matrix for K means clustering

In reality, we have very rare of single type measurement scale. Most of cases in real measurements (especially in behavioral survey) may consist of mixed type measurement scale of nominal, ordinal, and quantitative scale. We handle this situation by;

1. Use only normalized distance or similarity (which has value [0, 1]) for all variables.
2. Determine the weight of each feature variable w_{ijk} (usually between 0 and 1)
3. Then, general aggregated similarity and dissimilarity index are simple weighted average of distance matrices of each features variables

$$s_{ij} = \frac{\sum_{k=1}^{n} w_{ijk} s_{ijk}}{\sum_{k=1}^{n} w_{ijk}} \qquad \text{Eq. 2}$$

$$d_{ij} = \frac{\sum_{k=1}^{n} w_{ijk} \delta_{ijk}}{\sum_{k=1}^{n} w_{ijk}} \qquad \text{Eq. 3}$$

Index k represents the features variables. s_{ijk} and δ_{ijk} are similarity and dissimilarity of between object i and j for feature k.

2.3 A new definition for the cluster center concept

After the transformation of the dataset with multi-categorical variables to dataset with only binary variables was used, methods for clustering binary data can be applied. Here is important to take into account variables of both nominal and ordinal nature. When we cluster the binary variables of nominal data, there is a sparse matrix. We then propose to compute the center of a cluster by using the dynamic threshold.

1. Find the number of occurrences of bit “1” of all objects in cluster k :
2. Divide the number of occurrences in (1) by a number of bit and set it to be the dynamic threshold
3. Calculate cluster center by using the dynamic threshold divide the number of occurrences of bit “1” of the considered term.
4. Repeat until all of clusters.

In some works, authors have proposed some definitions for the center of categorical or nominal data. For example, [3] proposes to compute the center of a cluster by using the mode of a set.

3. Conclusion

Although cluster analysis of nominal variable is popular theme in the papers, it is implemented in software packages only rarely. Even K-means clustering is suitable only for simple numerical data. Generally, business data consist of many categorical variables with complex taxonomic domain structure. We then proposed the algorithm for adapted k-means clustering. It can use for all types of data. We apply this algorithm for our proposed system, the multidimensional recommendation system [4].

4. References

[1] A.K.Jain, M.N.Murty, and P.J.Flynn. Data clustering: A review. ACM Computing Surveys, 31:264-323, 1999.

[2] L. Bottou and Y. Bengio. Convergence properties of the K-means algorithms. In G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information

Processing Systems, volume 7, pages 585-592. The MIT Press, 1995.

[3] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 1998.

[4] Sutheera Puntheeranurak and Hidekazu Tsuji, A framework of a multidimensional recommendation system, 第 68 回情報処理学会全国大会、工学院大学 新宿キャンパス、2006 年