INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

CLUSTER A CLASS DETECTION DATA USING HIGH DIMENSION WITH NEAREST NEIGHBOR'S POINTS IN DATA MINING

[1]S. RAJESH KUMAR, [2]G.R.SUGANYA
[1]Assistant Professor,
Department of Computer Application, Bharath College of Scinece and Management, Thanjavur, India
[2]Assistant Professor,
Department of Computer Application, Bharath College of Scinece and Management, Thanjavur, India

[1]sivanesanrajesh@gmail.com, [2]sugnayagr@gmail.com

ABSTRACT

Clustering depends critically on density and distance (similarity), but these concepts become increasingly more difficult to define as dimensionality increases. In this paper we offer definitions of density and similarity that work well for high dimensional data (actually, for data of any dimensionality). In particular, we use a similarity measure that is based on the number of neighbors that two points share, and define the density of a point as the sum of the similarities of a point's nearest neighbors. We then present a new clustering algorithm that is based on these ideas. This algorithm eliminates noise (low density points) and builds clusters by associating non-noise points with representative or core points (high density points). This approach handles many problems that traditionally plague clustering algorithms, e.g., finding clusters in the presence of noise and outliers and finding clusters in data that has clusters of different shapes, sizes, and density. We have used our clustering algorithm on a variety of high and low dimensional data sets with good results, but in this paper, we present only a couple of examples involving high dimensional data sets: word clustering and time series derived from NASA Earth science data.

Keywords: Cluster; Detection; High Dimension; Nearest Neighbor; Data mining

1. INTRODUCTION

Cluster analysis tries to divide a set of data points into useful or meaningful groups, and has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining. Cluster analysis is a challenging task and there are a number of well-known issues associated with it, e.g., finding clusters in data where there are clusters of different shapes, sizes, and density or where the data has lots of noise and outliers. These issues become more important in the context of high dimensionality data sets.

For high dimensional data, traditional clustering techniques have sometimes been used. For example, the K-means algorithm and agglomerative hierarchical clustering techniques [DJ88], have been used extensively for clustering document data. While K-means is efficient and often produces “reasonable” results, in high dimensions, K-means still retains all of its low dimensional limitations, i.e., it has difficulty with outliers and does not do a good job when the clusters in the data are of different sizes, shapes, and densities. Any new clustering algorithm must be evaluated with respect to its performance on various data sets, and we present a couple of examples. For the first example, we find clusters in NASA Earth science data, i.e., pressure time series. For this data, our shared nearest neighbour (SNN) approach has found clusters that correspond to well-known climate phenomena, and thus we have confidence that the clusters we found are “good.” Using these clusters as a baseline, we show that the clusters found by Jarvis-Patrick clustering [JP73], an earlier SNN clustering approach, and K-means clustering are not as “good.” For the second example, we cluster document terms, showing that our clustering algorithm produces highly coherent sets of terms. We also show that a cluster consisting of a single word can be quite meaningful.

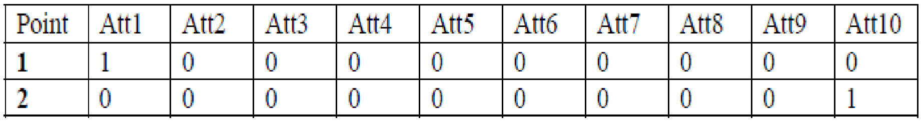
The basic outline of this paper is as follows. Section 2 describes the challenges of clustering high dimensional data: the definition of density and similarity measures, and the problem of finding non-globular clusters. Section 3 describes previous clustering work using the shared nearest neighbour approach, while Section 4 introduces our new clustering algorithm. Section 5 presents a couple of examples: the first example Section finds clusters in NASA Earth science data, i.e., pressure time series, while the second example describes the results of clustering document terms.

2. CHALLENGES OF CLUSTERING HIGH DIMENSIONAL DATA

The ideal input for a clustering algorithm is a dataset, without noise, that has a known number of equal size, equal densities, globular. When the data deviates from these properties, it poses different problems for different types of algorithms. While these problems are important for high dimensional data, we also need to be aware of problems which are not necessarily important in two dimensions, such as the difficulties associated with density and measures of similarity and distance. In this section, we take a look at these two problems and also consider the importance of representative points for handling non-globular clusters.

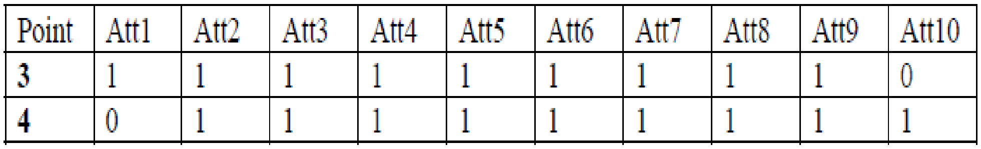
2.1 BEHAVIOR OF SIMILARITY AND DISTANCE MEASURES IN HIGH DIMENSIONS

The most common distance metric used in low dimensional datasets is Euclidean distance, or the L2 norm. While Euclidean distance is useful in low dimensions, it doesn’t work as well in high dimensions. Consider the pair of ten-dimensional data points, 1 and 2, shown below, which have binary attributes.



Point	Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Att9	Att10
1	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1

If we calculate the Euclidean distance between these two points, we get √2. Now, consider the next pair of ten-dimensional points, 3 and 4.



Point	Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Att9	Att10
3	1	1	1	1	1	1	1	1	1	0
4	0	1	1	1	1	1	1	1	1	1

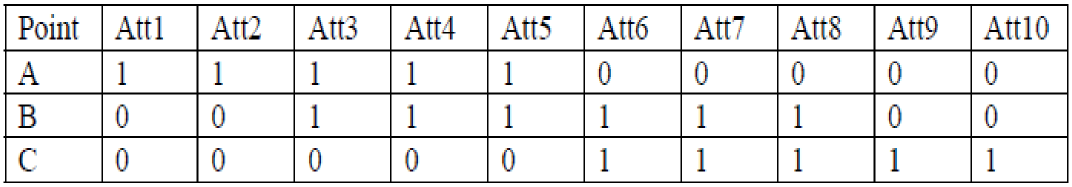
If we calculate the distance between point 3 and 4, we again find out that it's √2. Notice that points 1 and 2 do not share any common attributes, while points 3 and 4 are almost identical. Clearly Euclidean distance does not capture the similarity of points with binary attributes. The problem with Euclidean distance is that missing attributes are as important as the present attributes. However, in high dimensions, the presence of an attribute is a lot more important than the absence of an attribute, provided that most of the data points are sparse vectors (not full), and in high dimensions, it is often the case that the data points will be sparse vectors, i.e. they will only have a handful of non-zero attributes (binary or otherwise).

Different measures, such as the cosine measure and Jaccard coefficient, have been suggested to address this problem. The cosine similarity between two data points is equal to the dot product of the two vectors divided by the individual norms of the vectors. (If the vectors are already normalized the cosine similarity simply becomes the dot product of the vectors.) The Jaccard coefficient between two points is equal to the number of intersecting attributes divided by the number of spanned attributes by the two vectors (if attributes are binary). There is also an extension of Jacquards coefficient to handle non-binary attributes. If we calculate the cosine similarity or Jaccard coefficient between data points 1 and 2, and 3 and 4, we'll see that the similarity between 1 and 2 is equal to zero, but is almost 1 between 3 and 4.

Nonetheless, even though we can clearly see that both of these measures give more importance to the presence of a term than to its absence, there are cases where using such similarity measures still does not eliminate all problems with similarity in high dimensions. We investigated several TREC datasets (which have class labels), and found out that 15-20% of the time, for a data point A, its most similar data point (according to the cosine measure) is of a different class. This problem is also illustrated in [GRS99] using a synthetic market basket dataset.

Note that this problem is not due to the lack of a good similarity measure. Instead, the problem is that direct similarity in high dimensions cannot be trusted when the similarity between pairs of points are low. In general, data in high dimensions is sparse and the similarity between data points, on the average, is very low.

Another very important problem with similarity measures in high dimensions is that, the triangle inequality doesn't hold. Here's an example:



Point	Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Att9	Att10
A	1	1	1	1	1	0	0	0	0	0
B	0	0	1	1	1	1	1	1	0	0
C	0	0	0	0	0	1	1	1	1	1

Point A is close to point B, point B is close to point C, and yet, the points A and C are infinitely far apart. The similarity between A and B and C and B comes from different sets of attributes.

2.2 DEALING WITH NON-GLOBULAR CLUSTERS USING REPRESENTATIVE POINTS

Non-globular cluster cannot be handled by centroid-based schemes, since, by definition, such clusters are not represented by their centroid. Single link methods are most suitable for capturing clusters with non-globular shapes, but these methods are very brittle and cannot handle noise properly. However, representative points are a good way of finding clusters that are not characterized by their centroid and have been used in several recent clustering algorithms, e.g., CURE and DBSCAN.

In CURE, the concept of representative points is used to find non-globular clusters. The use of representative points allows CURE to find many types of non-globular clusters. However, there are still many types of globular shapes that CURE cannot handle. This is due to the way the CURE algorithm finds representative points, i.e., it finds points along the boundary, and then shrinks those points towards the centre of the cluster.

The notion of a representative point is also used in DBSCAN, although the term "core point" is used. In DBSCAN, the density associated with a point is obtained by counting the number of points in a region of specified radius around the point. Points with a density above a specified threshold are classified as core points, while noise points are defined as non-core points that don't have a core points within the specified radius. Noise points are discarded, while clusters are formed around the core points. If two core points are neighbours of each other, then their clusters are joined. Non-noise, non-border points, which are called boundary points, are assigned to the clusters associated with any core point within their radius. Thus, core points form the skeleton of the clusters, while border points flesh out this skeleton.

While DBSCAN can find clusters of arbitrary shapes, it cannot handle data containing clusters of differing densities, since its density based definition of core points cannot identify the core points of varying density clusters. Consider Figure 1. If the user defines the neighbourhood of a point by a certain radius and looks for core points that have a pre-defined number of points within that radius, then either the tight left cluster will be picked up as one cluster and the rest will be marked as noise, or else every point will belong to one cluster.

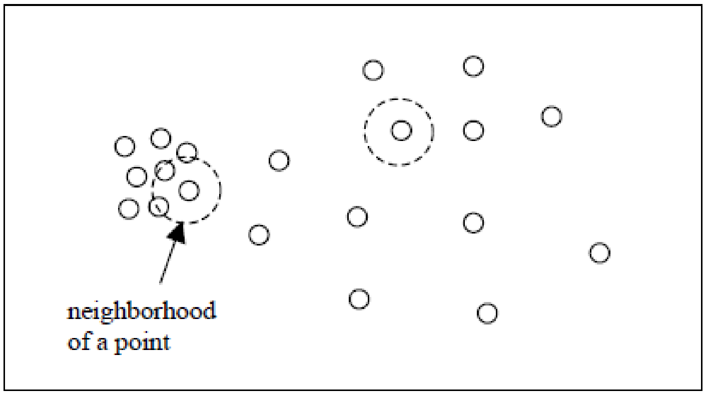
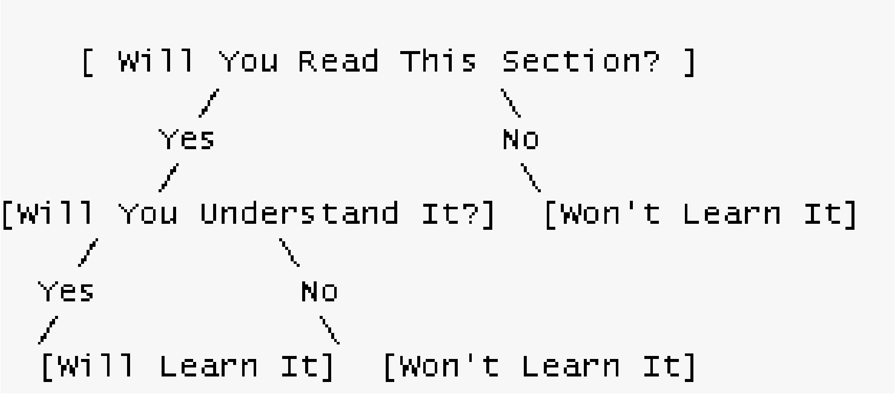


Figure 1. Density Based Neighborhoods

3. CLASSIFICATION

Classification (also known as classification trees or decision trees) is a data mining algorithm that creates a step-by-step guide for how to determine the output of a new data instance. The tree it creates is exactly that: a tree whereby each node in the tree represents a spot where a decision must be made based on the input, and you move to the next node and the next until you reach a leaf that tells you the predicted output. Sounds confusing, but it's really quite straightforward. Let's look at an example.



```
    [ Will You Read This Section? ]
          /             \
        Yes              No
        /                 \
[Will You Understand It?]  [Won't Learn It]
     /          \
   Yes           No
   /              \
 [Will Learn It]  [Won't Learn It]
```

Figure 2: Simple classification tree

4. SHARED NEAREST NEIGHBOR BASED ALGORITHM

An alternative to a direct similarity is to define the similarity between a pair of points in terms of their shared nearest neighbors. That is, the similarity between two points is “confirmed” by their common (shared) near neighbors. If point A is close to point B and if they are both close to a set of points C then we can say that A and B are close with greater confidence since their similarity is “confirmed” by the points in set C. This idea of shared nearest neighbor was first introduced by Jarvis and Patrick [JP73]. A similar idea was later presented in ROCK [GRS99].

In the Jarvis – Patrick scheme, a shared nearest neighbor graph is constructed from the proximity matrix as follows. A link is created between a pair of point's p and q if and only if p and q have each other in their closest k nearest neighbor lists. This process is called k-nearest neighbor sparsification. The weights of the links between two points in the snn graph can either be simply the number of near neighbors the two points share, or one can use a weighted version that takes the ordering of the near neighbors into account. Let i and j be two points. The strength of the link between i and j is now defined as:

$$str(i,j) = \sum (k+1-m)*(k+1-n), \quad \text{where } i_m = j_n$$

In the equation above, k is the near neighbor list size, m and n are the positions of a shared near neighbor in i and j's lists. At this point, all edges with weights less than a user specified threshold are removed and all the connected components in the resulting graph are our final clusters [JP73].

CONCLUSION

Clustering remains a popular method for extracting hypotheses from large amounts of data. One particular advantage is that, unlike some other data mining methods, it does not require any of the input data to be “labeled”, that is, inspected by an expert and tagged with a prognostication. Instead, clustering merely tries to identify groups of similar items within the data and report these back to the user. This falls into the class of “unsupervised learning” techniques, in contrast to “supervised learning”, which requires a training set of data to be made available which is tagged with the appropriate class identifier? Understanding the mechanism of the clustering method is important for the user, so that they may evaluate the significance and meaning of the results of clustering. We have only discussed a few of the clustering methods that have been proposed, and mentioned a few of the factors in their use.

REFERENCES

1. W. Lee and S.J. Stolfo, “Data Mining Approaches for Intrusion Detection” 7th USENIX Security Symposium,Texas, 1998.

2. P. Kumar, P.R. Krishna, B. S Raju and T. M Padmaja, “Advances in Classification of Sequence Data”, Data Mining and Knowledge Discovery Technologies. IGI Global, 2008, pp.143-174.

3. N. Jiang & L. Gruenwald, “CFI-Stream: miningclosed frequent Itemsets in data streams”, Proc.12th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, Philadelphia,PA,USA, 2006, pp. 592–597.

4. J. Cheng, Y. Ke, & W. Ng,” Maintaining frequent itemsets over high-speed data streams”, Proc. 10thPacific-Asia Conf. on Knowledge Discovery and Data Mining, Singapore, 2006, pp.462–467.

5. E.H. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering based on association rule hypergraphs.

6.Technical Report TR-97-019, Department of Computer Science, University of Minnesota,Minneapolis, 1997.

7. A.K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.

8. B. Lent, A. Swami, and J. Widom. Clustering association rules. In Proc. of the 13th Int'l Conf. on Data-Eng.,Birmingham, U.K., 1997.

9. R. Dubes and A.K. Jain. Clustering methodologies in exploratory data analysis. In M.C. Yovits, editor, Advances in Computers. Academic Press Inc., New York, 1980.

10. N. Harris, L.Hunter, and D.States. Mega-classification: Discovering motifs in massive datastreams. In Proceedings of the Tenth International Conference on Artificial Intelligence (AAAI), 1992.

11. R.C.T. Lee. Clustering analysis and its applications. In J.T. Toum, editor, Advances in Information Systems Science. Plenum Press, New York, 1981.

12. R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In Proc.of the 20th VLDB Conference, pages 144-155, Santiago, Chile, 1994.