Distance Based Subspace Clustering with Flexible Dimension Partitioning

Guimei Liu
National University of Singapore
Singapore
liugm@comp.nus.edu.sg

Jinyan Li Kelvin Sim
Institute for Infocomm Research
Singapore
{jinyan, shsim}@i2r.a-star.edu.sg

Limsoon Wong
National University of Singapore
Singapore
wongls@comp.nus.edu.sg

Abstract

Traditional similarity or distance measurements usually become meaningless when the dimensions of the datasets increase, which has detrimental effects on clustering performance. In this paper, we propose a distance-based subspace clustering model, called nCluster, to find groups of objects that have similar values on subsets of dimensions. Instead of using a grid based approach to partition the data space into non-overlapping rectangle cells as in the density based subspace clustering algorithms, the nCluster model uses a more flexible method to partition the dimensions to preserve meaningful and significant clusters. We develop an efficient algorithm to mine only maximal nClusters. A set of experiments are conducted to show the efficiency of the proposed algorithm and the effectiveness of the new model in preserving significant clusters.

1 Introduction

Clustering seeks to find groups of similar objects based on the values of their attributes. Traditional clustering algorithms use distance on the whole data space to measure similarity between objects. As the number of dimensions in a dataset increases, distance measures become increasingly meaningless [6]. In very high dimensional datasets, the objects are almost equidistant from each other. This is known as *the curse of high dimensionality* [5].

The concept of subspace clustering has been proposed to cope with the problems caused by high dimensionality by discovering clusters embedded in subspaces of high dimensional datasets. Many subspace clustering algorithms use a grid and density based approach [3, 8, 13, 7, 12]. They partition the data space into non-overlapping rectangular cells by discretizing each dimension into a number of bins. A cell is dense if the fraction of total objects contained in the cell is greater than a threshold. Clusters are formed by merging connected dense cells in the same subspace.

In the density based model, objects are clustered based on the distribution of the objects in subspaces instead of the distance between objects, so it is very likely that some objects in a cluster are far apart from each other even in the subspace of the cluster. In many applications, users are more interested in finding groups of objects that are physically close to one another in subspaces. In the density based approach, dense cells contain objects that have similar values on subsets of dimensions, so a simple approach to finding clusters of similar objects in subspaces is to find the dense cells. However, using this approach, a cluster may be divided into several small clusters as illustrated in the following example.

	a	b	c
1	0	10	1
2	**4**	**5**	4
3	**5**	**6**	0
4	**6**	**5**	7
5	9	0	10
6	10	1	6

Table 1. An example missing cluster

Table 1 shows a dataset containing 6 objects and 3 attributes. The value range of the three attributes is [0, 10]. Objects 2, 3 and 4 have similar values on both attributes a and b, so object set {2, 3, 4} and attribute set {a, b} should form a subspace cluster. If we use the grid based approach and partition each attribute to two bins of equal length, then for each attribute, we have two bins [0, 5] and (5, 10]. Object 4 is in different bins with objects 2 and 3 on attribute a, and object 3 is in different bins with objects 2 and 4 on attribute b. We get two smaller clusters ({2, 3}, a) and ({2, 4}, {b}). Algorithms have been proposed to find the cutting points adaptively based on data distribution [13, 7, 12]. However, these algorithms do not allow overlap between different bins either, so it is still possible that objects with similar values on an attribute are placed into different bins, which may cause a cluster to be shattered in different cells.

In this paper, we propose a distance based subspace clustering model called nCluster to overcome the problem discussed above. The nCluster model uses a more flexible method to partition the dimensions, which allows overlap

1-4244-0803-2/07/$20.00 ©2007 IEEE

between different bins of an attribute. This may result in more bins than the grid based algorithms, which increases the complexity of the problem. To make the problem solvable, we consider only those clusters containing a non-trivial number of objects and attributes, and we mine only maximal nClusters to avoid generating too many clusters.

The rest of the paper is organized as follows. Section 2 describes the nCluster model. Section 3 presents an efficient algorithm for mining maximal nClusters. The experiment results are reported in Section 4. Related work is discussed in Section 5. Finally, Section 6 concludes the paper.

2 The nCluster Model

Let $\mathcal{O}$ be a set of objects. Each object has a set of attributes $\mathcal{A}$ and the domains of the attributes in $\mathcal{A}$ are bounded. We use $x, y, \cdots$ to denote an object in $\mathcal{O}$, $a, b, \cdots$ to denote an attribute in $\mathcal{A}$, R_a to denote the value range of an attribute a, and v_{xa} to denote the value of an object x on an attribute a.

The distance of two objects x and y on an attribute a is defined as $|v_{xa} - v_{ya}|$. If the distance is smaller than a predefined threshold, then x and y are called neighbors on attribute a. Similarly, we can define neighbors of an object on a subset of attributes in $\mathcal{A}$, and they are called subspace neighbors. Attributes usually do not have the same value ranges. Therefore, instead of using a constant threshold on all attributes, we use a relative distance threshold which is specified as the ratio of the value range of an attribute.

Definition 1 (Subspace δ-neighbors) *Let x, y be two objects and $D \subseteq \mathcal{A}$ be a subset of attributes. If for every nominal attribute $a \in D$, we have v_{xa}=v_{ya}, and for every continuous attribute $a \in D$, we have $|v_{xa} - v_{ya}| \leq \delta \cdot R_a$, where δ is a predefined threshold, then we say that x and y are δ-neighbors of each other in subspace D.*

If a set of objects T are δ-neighbors of one another on a set of attributes D, then these objects form a cluster on subspace D and we call it a δ-nCluster.

Definition 2 (Subspace δ-nCluster) *Let $T \subseteq \mathcal{O}$ be a set of objects and $D \subseteq \mathcal{A}$ be a set of attributes. If for every two objects $x, y \in T$ and every attribute $a \in D$, objects x and y are δ-neighbors on attribute a, then we say that (T, D) is a subspace δ-nCluster, or simply δ-nCluster.*

Example 1 *Table 2 shows a dataset with 4 attributes and 8 objects. The value ranges of attribute a, b, c and d are [0, 20], [-50, 50], [0, 100] and [0, 30] respectively. If we set δ to 0.1, then $\{1, 2, 4, 6, 8\}$ and $\{a\}$ form a δ-nCluster, and ({1, 6}, {a, b, c}) is a δ-nCluster. In Table 1, if we set δ to 0.2, we can find the subspace cluster ({2, 3, 4}, {a, b}), which may not be found by the grid based approach.*

	a	b	c	d
1	5	0	27	0
2	6	50	75	24
3	3	-29	53	13
4	5	-2	51	30
5	0	1	100	7
6	6	4	29	19
7	20	27	23	1
8	7	-50	0	2

Table 2. An example dataset

Given two nClusters (T_1, D_1) and (T_2, D_2), if $T_1 \subseteq T_2$ and $D_1 \subseteq D_2$, then we say that (T_1, D_1) is a *sub-nCluster* of (T_2, D_2), and (T_2, D_2) is a *super-nCluster* of (T_1, D_1). If either $T_1 \subset T_2$ or $D_1 \subset D_2$ is true, then we say (T_1, D_1) is a proper sub-nCluster of (T_2, D_2). The δ-nClusters have the following property based on their definition.

Property 1 (anti-monotone property) *Let $T \subseteq \mathcal{O}$ be a set of objects and $D \subseteq \mathcal{A}$ be a set of attributes. If T and D form a δ-nCluster, then T forms a δ-nCluster with every subset of D, and D forms a δ-nCluster with every subset of T.*

The number of subspaces of $\mathcal{A}$ is exponential to the number of attributes in $\mathcal{A}$. It is impractical to exhaustively enumerate all the subspaces when $\mathcal{A}$ contains many attributes. For a cluster to be meaningful and useful, the cluster has to contain a non-trivial number of objects and attributes. We use two thresholds mr and mc to constrain the minimum number of objects and attributes, and we are interested in mining only δ-nClusters containing at least mr objects and mc attributes.

Restricting the minimum number of objects and attributes filters out insignificant δ-nClusters, but there still can be a large number of δ-nClusters, and many of them are redundant in the sense that they can be subsumed by some larger δ-nClusters. Based on Property 1, if a set of objects T and a set of attributes D can form a δ-nCluster, then any sub-nCluster of (T, D) can form a δ-nCluster. These sub-nClusters of (T, D) provide no more information than (T, D). To avoid generating too many δ-nClusters, we enumerate only maximal δ-nClusters.

Definition 3 (Maximal δ-nCluster) *Let $T \subseteq \mathcal{O}$ be a set of objects and $D \subseteq \mathcal{A}$ be a set of attributes, and T and D form a δ-nCluster. If there does not exist a δ-nCluster (T', D') such that (T, D) is a proper sub-nCluster of (T', D'), then (T, D) is called a maximal δ-nCluster.*

Example 2 *Let δ=0.1. In the example dataset shown in Table 2, δ-nCluster ({1, 6}, {a, b}) is not maximal because its attribute set can be extended by attribute c and its object set can be extended by object 4. δ-nClusters ({1, 4, 6}, {a, b}) and ({1, 6}, {a, b, c}) are maximal δ-nClusters because neither their object sets can be extended without reducing* *their attribute sets, nor their attribute sets can be extended without reducing their object sets.*

Wang et al. [18] proposed a clustering model called pClusters to find groups of objects that exhibit coherent patterns on subsets of attributes. Given a set of objects T and a set of attributes D, T and D form a δ-pCluster if for any two objects $x, y \in T$, and any two attributes $a, b \in D$, we have $|(v_{xa} - v_{xb}) - (v_{ya} - v_{yb})| \leq \delta$. If ($T$, D) is a δ-nCluster, then it must be a 2δ-pCluster because $|(v_{xa} - v_{xb}) - (v_{ya} - v_{yb})| \leq |v_{xa} - v_{xb}| + |v_{ya} - v_{yb}| \leq 2\delta$. We can use the algorithm for mining 2δ-pClusters to mine δ-nClusters. However, this approach is very inefficient for two reasons. First, we have to use a larger threshold to mine pClusters, which may generate many unqualified nClusters. For example, two objects x, y satisfying $v_{xa} - v_{xb} = 0$ and $v_{ya} - v_{yb} = 2\delta$ can be in some 2δ-pCluster containing attribute a and b, but they cannot be in any δ-nCluster containing attribute b. Secondly, the objective of the pCluster model is to mine groups of objects that exhibit coherent patterns instead of groups of similar objects, so a 2δ-pCluster may contain many objects that are not δ-neighbors. For example, two objects x, y satisfying $v_{xa} - v_{xb} = 10\delta$ and $v_{ya} - v_{yb} = 9\delta$ can be in some 2δ-pCluster containing attribute a and b, but they cannot be in any δ-nCluster containing attribute a or b. Therefore, given the same mr and mc threshold, mining 2δ-pClusters can produce many clusters that are not qualified to be δ-nClusters.

3 Mining Maximal δ-nClusters

In this section, we present an algorithm for mining maximal δ-nClusters containing at least mr objects and mc attributes. We use Property 1 to prune the search space. We start from δ-nClusters containing one attribute, and extend them to find δ-nClusters containing more attributes.

3.1 Finding δ-nClusters with single attribute

We are interested in maximal δ-nClusters, so for every subspace D, we find the maximal object sets that can form δ-nClusters with D. An attribute can form δ-nClusters with multiple maximal object sets. We identify them based on the following observation.

Lemma 1 *Given an attribute a and a set of objects T, (T, $\{a\}$) is a δ-nCluster if and only if $max\{v_{xa}|x \in T\} - min\{v_{xa}|x \in T\} \leq \delta \cdot R_a$.*

Based on the above lemma, we identify the maximal object sets of an attribute using a method similar to the method used in [18] for finding maximal dimension sets (MDS). We sort the objects in $\mathcal{O}$ in ascending order of their values on attribute a, and then we find pairs of positions p_1 and p_2 ($p_1 < p_2$) in the sorted sequence such that the difference of the values at the two positions is no larger than $\delta \cdot R_a$, but the difference between values at $(p_1 - 1)$ and p_2 or at p_1 and $(p_2 + 1)$ is larger than $\delta \cdot R_a$.

If the number of distinct values of an attribute is very large, then the number of maximal object lists generated can be very large. This may pose a difficulty on the mining algorithm. To avoid generating too many highly overlapped maximal object lists on the same attribute, we use a threshold ω to control the overlap. Threshold ω is used as follows. Let T be the current maximal object set discovered on attribute a, and R_T be the span of T, that is $R_T = [min_{x \in T}\{v_{xa}\}, max_{x \in T}\{v_{xa}\}]$. Then the span of the next maximal object set cannot have more than $\omega \cdot |R_T|$ overlap with R_T. When $\omega = 0$, we divide attributes into non-overlapping bins as in the grid based approach.

attr	maximal object sets
a_1	{1, 3, 4}
a_2	{1, 2, 4, 6, 8}
b_1	{1, 4, 5, 6}
c_1	{1, 6, 7}
c_2	{3, 4}
d_1	{1, 7, 8}

Table 3. Maximal object sets of attributes

Table 3 shows the maximal object sets of all the attributes in Table 2. Every attribute and its maximal object set forms a δ-nCluster containing only one attribute. We use these δ-nClusters as starting points to find δ-nClusters containing more attributes.

3.2 Finding maximal δ-nClusters containing more than one attribute

Given a δ-nCluster (T, D) and an attribute $a \notin D$, if there are at least mr objects in T that are δ-neighbors of one another on attribute a, then attribute a can be added to D to form a δ-nCluster with one more attribute. To find all such attribute a, we maintain an attribute list for every object. The attribute list of an object x contains all the attributes on which x has at least $mr - 1$ δ-neighbors. We need to distinguish the different maximal object sets of an attribute. For example, in Table 3, attribute a has two maximal object sets. If we simply add a to the attribute lists of all the objects contained in its two maximal object sets, then we cannot tell which objects are in the same maximal object sets.

To solve this problem, we add a subscript to an attribute name when we add the attribute name to the attribute lists of objects. The attribute lists of the objects in the same maximal object set receive the attribute with the same subscript. We call a subscripted attribute name a symbol of the attribute to distinguish it from the attribute itself. The number of symbols of an attribute is equal to the number of maximal object sets of the attribute, and the frequency of an attribute symbol in the attribute lists is equal to the size of the maximal object set the symbol represents. In the above example, attribute a has two symbols a_1 and a_2, the attribute lists of objects 1, 3 and 4 contain a_1, and the attribute lists of objects 1, 2, 4, 6, and 8 contain a_2. The attribute lists of all the objects in Table 2 are shown in Table 4.

obj	attribute lists
1	a_1, a_2, b_1, c_1, d_1
2	a_2
3	a_1, c_2
4	a_1, a_2, b_1, c_2
5	b_1
6	a_2, b_1, c_1
7	c_1, d_1
8	a_2, d_1

Table 4. Attribute lists of objects

The above transformation is lossless, that is, we can reconstruct Table 3 from Table 4.

Lemma 2 *Two objects are δ-neighbors on an attribute a if and only if the attribute lists of the two objects contain the same symbol of attribute a.*

Since the attribute lists contain the complete information, so we use attribute lists to discover maximal δ-nClusters in the remaining mining. Our mining algorithm is based on the following observation.

Lemma 3 *A set of attributes D forms a δ-nCluster with a set of objects T if and only if the attribute lists of the objects in T all contain the same symbol of every attribute in D.*

If we regard attribute symbols as items, attribute symbol sets as itemsets, and attribute lists as transactions, then mining δ-nClusters can be transformed to mining frequent itemsets from a transaction database [4]. The concept of maximal δ-nClusters is used in the paper to remove redundant δ-nClusters, and it is similar to the frequent closed itemset concept [14], which is used to remove redundant itemsets. An itemset is closed if it is maximal in the set of transactions containing it. If a δ-nCluster is maximal, then its corresponding attribute symbol set is a closed itemset in the attribute lists. We can use frequent closed itemset mining algorithms to mine maximal δ-nClusters.

LCM [17] is one of the most efficient algorithms for mining frequent closed itemsets, so in this paper, we use LCM to mine maximal nClusters and we call it LCM-nCluster. LCM generates only attribute symbol sets, we use a post-processing step to generate the corresponding object sets by intersecting the maximal object sets of the attribute symbols. We accelerate the intersection operation by reusing intermediate intersection results.

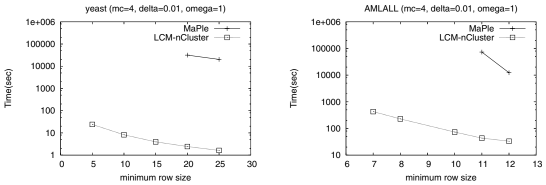
Note that a closed itemset may not always yield a maximal δ-nCluster. For example, $\{a_1, a_2, b_1\}$ is a closed itemset in Table 4, and its corresponding attribute set is $\{a, b\}$ and object set is $\{1, 4\}$, but ($\{1, 4\}$, $\{a, b\}$) is not a maximal nCluster because one of its super-nCluster ($\{1, 4, 6\}$, $\{a, b\}$) is also a δ-nCluster. We remove non-maximal δ-nClusters in a post-processing step.

4 A Performance Study

In this section, we study the efficiency of the LCM-nCluster algorithm and the effectiveness of the δ-nCluster model in preserving significant subspace clusters. Our experiments were conducted on a Windows machine with a 3.0Ghz Pentium IV CPU and 2GB memory. We used two real datasets. One dataset is the gene expression data used in [18], denoted as yeast. The other one named AMLALL is used in [10] to study leukemia. It contains 72 bone marrow samples and 7129 probes from 6817 human genes and is available at `http://research.i2r.a-star.edu.sg/rp/`.

4.1 Mining Efficiency

We compared LCM-nCluster with MaPle [15], which improves the work of Wang et al. [18] by mining only maximal pClusters. We transformed the datasets to make the δ threshold of nClusters and pClusters to be consistent. Figure 1 shows the running time of the two algorithms.



(a) yeast (b) AMLALL

Figure 1. Comparison with MaPle

With the mining parameters specified in the figures, LCM-nCluster is significantly faster than MaPle. The number of pClusters generated by MaPle is orders of magnitude larger than the number of nClusters. It indicates that it is impractical to use the algorithms for mining pClusters to mine nClusters.

4.2 Number of maximal δ-nClusters

We study the effectiveness of the nCluster model in preserving significant subspace clusters by inspecting the number of δ-nClusters generated when varying the overlapping threshold ω. Table 5 shows the number of δ-nClusters under different ω thresholds. The mining parameters are set as mr=25 and δ=0.03 on yeast, and mr=10 and δ=0.01 on AMLALL.

datasets	mc	0	0.25	0.50	0.75	1
AMLALL	4	352	834	7321	28548	92724
AMLALL	6	1	2	502	3230	17889
AMLALL	8	0	0	1	36	952
yeast	4	11915	11915	12051	12052	12052
yeast	6	1721	1721	1721	1721	1721

Table 5. Number of maximal δ-nClusters

The ω threshold controls the overlap between different bins of each dimension. With ω=0, the nCluster model partitions the data space into non-overlapping grids like the grid based approach. Table 5 shows that with the increase of ω, the number of δ-nClusters increases. It indicates that by allowing overlap between different bins of each dimension, more δ-nClusters can be preserved. In particular, on dataset AMLALL, no δ-nClusters containing more 8 attributes can be discovered from AMLALL with ω=0, while 36 δ-nClusters are discovered with ω=0.75 and 952 are discovered with ω=1.

5 Related Work

Besides the density and grid based algorithms [3, 8, 13, 7, 12, 16], there are a number of other subspace clustering algorithms that use a top-down strategy to find non-overlapping subspace clusters [1, 2, 19, 20]. Most of them use greedy or heuristic based approaches, which do not guarantee to find the complete set of subspace clusters.

Jagadish et al. [11] proposed the concept of fascicles, which is similar to δ-nClusters. Their objective is to use fascicles to compress data, so instead of enumerating all fascicles, they find fascicles that minimize data storage. Cheng et al. [9] modeled biclusters as submatrices in expression data that have low mean squared residue scores. Mean squared residue scores do not have the anti-monotone property, which poses difficulties on developing efficient mining algorithms. Therefore, Cheng et al. used a randomized algorithm to find biclusters.

6 Conclusion

In this paper, we have proposed a new subspace clustering model called nClusters to find clusters embedded in subspaces of high dimensional datasets. Compared with the traditional grid based approach, the nCluster model uses a more flexible method to partition dimensions, thus it can find more meaningful and significant clusters.

References

[1] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proc. of the 1999 ACM SIGMOD Conference*, pages 61–72, 1999.

[2] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proc. of the 2000 ACM SIGMOD Conference*, pages 70–81, 2000.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. of the 1998 ACM SIGMOD Conference*, pages 94–105, 1998.

[4] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD Conference*, pages 207–216, 1993.

[5] R. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, 1961.

[6] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is ”nearest neighbor” meaningful? In *Proc. of the 7th ICDT Conference*, pages 217–235, 1999.

[7] J.-W. Chang and D.-S. Jin. A new cell-based clustering method for large, high-dimensional data in data mining applications. In *Proc. of the 2002 ACM symposium on Applied computing*, pages 503–507, 2002.

[8] C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proc. of the 5th ACM SIGKDD Conference*, pages 84–93, 1999.

[9] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proc. of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[10] T. Golub and et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[11] H. V. Jagadish, J. Madar, and R. T. Ng. Semantic compression and pattern extraction with fascicles. In *Proc. of the 25th VLDB Conference*, pages 186–198, 1999.

[12] B. Liu, Y. Xia, and P. S. Yu. Clustering through decision tree construction. In *Proc. of the 9th CIKM conference*, pages 20–29, 2000.

[13] H. Nagesh, S. Goil, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. Technical Report 9906-010, Northwestern University, June 1999.

[14] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. of the 7th ICDT Conference*, pages 398–416, 1999.

[15] J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. Maple: A fast algorithm for maximal pattern-based clustering. In *Proc. of the 3rd ICDM Conference*, pages 259–266, 2003.

[16] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A monte carlo algorithm for fast projective clustering. In *Proc. of the 2002 ACM SIGMOD Conference*, pages 418–427, 2002.

[17] T. Uno, M. Kiyomi, and H. Arimura. Lcm ver. 3: Collaboration of array, bitmap and prefix tree for frequent itemset mining. In *Proc. of the ACM SIGKDD OSDM workshop*, 2005.

[18] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proc. of the 2002 ACM SIGMOD Conference*, pages 394–405, 2002.

[19] K.-G. Woo, J.-H. Lee, M.-H. Kim, and Y.-J. Lee. Findit: a fast and intelligent subspace clustering algorithm using dimension voting. *Information & Software Technology*, 46(4):255–271, 2004.

[20] J. Yang, W. Wang, H. Wang, and P. S. Yu. δ-clusters: Capturing subspace correlation in a large data set. In *Proc. of the 18th IEEE ICDE Conference*, pages 517–528, 2002.