Chapter

MINING HIGH-DIMENSIONAL DATA

Wei Wang[1] and Jiong Yang[2]

1. *Department of Computer Science, University of North Carolina at Chapel Hill*

2. *Department of Electronic Engineering and Computer Science, Case Western Reserve University*

Abstract: With the rapid growth of computational biology and e-commerce applications, high-dimensional data becomes very common. Thus, mining high-dimensional data is an urgent problem of great practical importance. However, there are some unique challenges for mining data of high dimensions, including (1) the curse of dimensionality and more crucial (2) the meaningfulness of the similarity measure in the high dimension space. In this chapter, we present several state-of-art techniques for analyzing high-dimensional data, e.g., frequent pattern mining, clustering, and classification. We will discuss how these methods deal with the challenges of high dimensionality.

Key words: High-dimensional data mining, frequent pattern, clustering high-dimensional data, classifying high-dimensional data

1. INTRODUCTION

The emergence of various new application domains, such as bioinformatics and e-commerce, underscores the need for analyzing high dimensional data. In a gene expression microarray data set, there could be tens or hundreds of dimensions, each of which corresponds to an experimental condition. In a customer purchase behavior data set, there may be up to hundreds of thousands of merchandizes, each of which is mapped to a dimension. Researchers and practitioners are very eager in analyzing these data sets.

Various data mining models have been proven to be very successful for analyzing very large data sets. Among them, frequent patterns, clusters, and classifiers are three widely studied models to represent, analyze, and summarize large data sets. In this chapter, we focus on the state-of-art techniques for constructing these three data mining models on massive high-dimensional data sets.

2. CHANLLENGES

Before presenting any algorithm for building individual data mining models, we first discuss two common challenges for analyzing high-dimensional data. The first one is the curse of dimensionality. The complexity of many existing data mining algorithms is exponential with respect to the number of dimensions. With increasing dimensionality, these algorithms soon become computationally intractable and therefore inapplicable in many real applications.

Secondly, the specificity of similarities between points in a high dimensional space diminishes. It was proven in [3] that, for any point in a high dimensional space, the expected gap between the Euclidean distance to the closest neighbor and that to the farthest point shrinks as the dimensionality grows. This phenomenon may render many data mining tasks (e.g., clustering) ineffective and fragile because the model becomes vulnerable to the presence of noise. In the remainder of this chapter, we present several state-of-art algorithms for mining high-dimensional data sets.

3. FREQUENT PATTERN

Frequent pattern is a useful model for extracting salient features of the data. It was originally proposed for analyzing market basket data [2]. A market basket data set is typically represented as a set of transactions. Each transaction contains a set of items from a finite vocabulary. In principle, we can represent the data as a matrix, each row represents a transaction and each column represents an item. The goal is to find the collection of itemsets appearing in a large number of transactions, defined by a support threshold t. Most algorithms for mining frequent patterns utilize the Apriori property stated as follows. If an itemset A is frequent (i.e., present in more than t transactions), then every subset of A must be frequent. On the other hand, if an itemset A is infrequent (i.e, present in less than t transactions), then any superset of A is also infrequent. This property is the basis of all level-wise search algorithms. The general procedure consists of a series of iterations

beginning with counting item occurrences and identifying the set of frequent items (or equivalently, frequent 1-itemsets). During each subsequent iteration, candidates for frequent *k*-itemsets are proposed from frequent (*k*-1)-itemsets using the Apriori property. These candidates are then validated by explicitly counting their actual occurrences. The value of *k* is incremented before the next iteration starts. The process terminates when no more frequent itemset can be generated. We often refer to this level-wise approach as the breadth-first approach because it evaluates the itemsets residing at the same depth in the lattice formed by imposing the partial order of subset-superset relationship between itemsets.

It is a well-known problem that the full set of frequent patterns contains significant redundant information and consequently the number of frequent patterns is often too large. To address this issue, Pasquier et al. [9] proposed to mine a selective subset of frequent patterns, called *closed frequent patterns*. If the number of occurrences of a pattern is the same to all its immediate subpatterns, then the pattern is considered as a closed pattern. In [10], the CLOSET algorithm is proposed to expedite the mining of closed frequent patterns. CLOSET uses a novel frequent pattern tree (FP structure) as a compact representation to organize the data set. It performs a depth-first search, that is, after discovering a frequent itemset *A*, it searches for superpatterns of *A* before checking *A*'s siblings.

A more recent algorithm for mining frequent closed pattern is CHARM [14]. Similar to CLOSET, CHARM searches for patterns in a depth-first manner. The difference between CHARM and CLOSET is that CHARM stores the data set in a vertical format where a list of row IDs is maintained for each dimension. These row ID lists are then merged during a "column enumeration" procedure that generates row ID lists for other nodes in the enumeration tree. In addition, a technique called *diffset* is used to reduce the length of the row ID lists as well as the computational complexity of merging them.

All previous algorithms can find frequent closed patterns when the dimensionality is low to moderate. When the number of dimensions is very high, e.g., greater than 100, the efficiency of these algorithms could be significantly impacted. CARPENTER [8] is therefore proposed to solve this problem. It first transposes the matrix representing the data set. Next, CARPENTER performs a depth-first row-wise enumeration on the transposed matrix. It has been shown that this algorithm can greatly reduce the computation time especially when the dimensionality is high.

4. CLUSTERING

Clustering is a widely adopted data mining model that partitions data points into a set of groups, each of which is called a *cluster*. A data point has a shorter distance to points within the cluster than those outside the cluster. In a high dimensional space, for any point, its distance to its closest point and that to the farthest point tend to be similar. This phenomenon may render the clustering result sensitive to any small perturbation to the data due to noise and make the exercise of clustering useless. To solve this problem, Agrawal et. al. proposed a subspace clustering model [1]. A subspace cluster consists of a subset of objects and a subset of dimensions such that the distance among these objects is small within the given set of dimensions. The CLIQUE algorithm [1] is proposed to find the subspace clusters.

In many applications, users are more interested in the objects that exhibit a consistent trend (rather than points having similar values) within a subset of dimensions. One such example is the bicluster model [4] proposed for analyzing gene expression profiles. A bicluster is a subset of objects (U) and a subset dimensions (D) such that objects in U have the same trends (i. e., fluctuating simultaneously) across dimensions in D. This is particular useful in analyzing gene expression levels in a microarray experiment since the expression levels of some genes may be inflated/deflated systematically in some experiments. Thus, the absolute value is not as important as the trend. If two genes have similar trends across a large set of experiments, they are likely to be co-regulated. In the bicluster model, the mean squared error residue is used to qualify a bicluster. In [4], Cheng and Church used a heuristic randomized algorithm to find biclusters. It consists of a series of iterations, each of which locates one bicluster. To prevent the same bicluster from being reported again in subsequent iterations, each time when a bicluster is found, the values in the bicluster are replaced by uniform noise before the next iteration starts. This procedure continues until a desired number of biclusters are discovered.

Although the bicluster model and algorithm have been used in several applications in bioinformatics, it has two major drawbacks: (1) the mean squared error residue may not be the best measure to qualify a bicluster. A big cluster may have small mean squared error residue even if it includes a small number of objects whose trends are vastly different in the selected dimensions; (2) the heuristic algorithm may be interfered by the noise artificially injected after each iteration and hence may not discover overlapped clusters properly. To solve these two problems, the authors of [12] proposed the *p-cluster* model. A p-cluster consists of a subset of objects U and a subset of dimensions D where for each pair of objects u_1 and u_2 in U and each pair of dimension d_1 and d_2 in D, the change of u_1 from d_1

to d_2 should be similar to that of u_2 from d_1 to d_2. A threshold is used to evaluate the dissimilarity between two objects on two dimensions. Given a subset of objects and a subset of dimensions, if the dissimilarity between every pair of objects on every pair of dimensions is less than the threshold, then these objects constitute a p-cluster in the given dimensions. In [12], a novel deterministic algorithm is developed to find all maximal p-clusters, which utilizes the Apriori property held on p-clusters.

5. CLASSIFICATION

The classification is also a very powerful data analysis tool. In a classification problem, the dimensions of an object can be divided into two types. One dimension records the class type of the object and the rest dimensions are attributes. The goal of classification is to build a model that captures the intrinsic associations between the class type and the attributes so that an (unknown) class type can be accurately predicted from the attribute values. For this purpose, the data is usually divided into a training set and a test set, where the training set is used to build the classifier which is validated by the test set. There are several models developed for classifying high dimensional data, e.g., naïve Bayesian, neural networks, decision trees [7], SVMs, rule-based classifiers, and so on.

Supporting vector machine (SVM) [11] is one of the newly developed classification models. The success of SVM in practice is drawn by its solid mathematical foundation that conveys the following two salient properties. (1) The classification boundary functions of SVMs maximize the margin, which equivalently optimize the general performance given a training data set. (2) SVMs handle a nonlinear classification efficiently using the kernel trick that implicitly transforms the input space into another higher dimensional feature space. However, SVM suffers from two problems. First, the complexity of training an SVM is at least $O(N^2)$ where N is the number of objects in the training data set. It could be too costly when the training data set is large. Second, since an SVM essentially draws a hyper-plain in a transformed high dimensional space, it is very difficult to identify the principal (original) dimensions that are most responsible for the classification.

Rule-based classifiers [6] offer some potential to address the above two problems. A rule-based classifier consists of a set of rules in the following form: $A_1[l_1, u_1] \cap A_2[l_2, u_2] \cap \ldots \cap A_m[l_m, u_m] \rightarrow C$, where $A_i[l_i, u_i]$ is the range of attribute A_i's value and C is the class type. The above rule can be interpreted as that, if an object whose attributes' values fall in the ranges in the left hand side, then its class type is likely to be C (with some high probability). Each

rule is also associated with a confidence level that depicts the probability that such a rule holds. When an object satisfies several rules, either the rule with the highest confidence (e.g., CBA[6]) or a weighted voting of all valid rules (e.g., CPAR[13]) may be used for class prediction. However, neither CBA nor CPAR are targeted for high dimensional data. In [5], an algorithm called FARMER is proposed to generate rule-based classifiers for high dimensional data set. It first quantizes the attributes into a set of bins. Each bin is treated as an item subsequently. FARMER then generates the closed frequent itemsets using a method similar to CARPENTER. These closed frequent itemsets are the basis to generate rules. Since the dimensionality is high, the number of possible rules in the classifier could be very large. FARMER finally organizes all rules into compact rule groups.

6. REFERENCES

1. Agrawal, Gehrke, Gunopulos, Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. Proceedings of ACM SIGMOD International Conference on Management of Data, 1998.
2. Agrawal, Srikant. Fast Algorithms for mining association rules in large databases. Proceedings of International Conferences on Very Large Data Bases, 1994.
3. Beyer, Goldstein, Ramakrishnan, Shaft. When is “nearest neighbor” meaningful? Proceedings of International Conference on Database Theory, 1999.
4. Cheng, Church. Biclustering of expression data. Proceedings of International Conference on Intelligent System for Molecular Biology, 2000.
5. Cong, Xu, Pan, Tung, Yang. FARMER: finding interesting rule groups in microarray datasets. SIGMOD International Conference on Management of Data, 2004.
6. Liu, Ma, Wong. Improving an association rule based classifier. Proceedings of International Conference on Principles of Knowledge Discovery and Data Mining, 2000.
7. Mitchell. Machine Learning. WCB McGraw Hill, 1997.
8. Pan, Cong, Tung, Yang, Zaki. CARPENTER: finding closed patterns in long biological data sets. Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
9. Pasquier, Bastide, Taouil, Lakhal. Discovering frequent closed itemsets for association rules. Proceedings of International Conference on Database Theory, 1999.
10. Pei, Han, Mao. CLOSET: an efficient Algorithm for mining frequent closed itemsets. Proceedings of ACM Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000.
11. Vapnik. Statistical Learning Theory. John Wiley and Sons, 1998.
12. Wang, Wang, Yang, Yu. Clustering by pattern similarity in large data sets. Proceedings of ACM SIGMOD International Conference on Management of Data, 2002.

13. Yin, Han. CPAR: classification based on predictive association rules. Proceedings of SIAM International Conference on Data Mining, 2003.
14. Zaki, Hsiao. CHARM: an efficient algorithm for closed association rule mining. Proceedings of SIAM International Conference on Data Mining, 2002.