Comparing Massive High-dimensional Data Sets

Theodore Johnson and **Tamraparni Dasu**
johnsont@research.att.com and tamr@research.att.com
Database Research and ML and IR Research
AT&T Labs - Research
Florham Park, NJ 07932

Abstract

The comparison of two data sets can reveal a great deal of information about the time-varying nature of an observed process. For example, suppose that the points in a data set represent a customer's activity by their location in n-dimensional space. A comparison of the distribution of points in two such data sets can indicate how the customer activity has changed between the observation periods. Other applications include data integrity checking. An unexpected change in a data set can indicate a problem in the data collection process.

We propose a fast, inexpensive method for comparing massive high dimensional data sets that does not make any distributional assumptions. The method adapts the power of classical statistics for use on complex, high dimensional data sets. We generate a map of the data set (a *DataSphere*), and compare data sets by comparing their DataSpheres. The DataSphere can be generated in two passes over the data set, stored in a database, and aggregated at multiple levels. We illustrate the use of our set comparison technique with an example analysis of data sets drawn from AT&T data warehouses.

Introduction

Data warehouses provide access to large volumes of detailed information about an important function (e.g., sales, customer behavior, internal operations, etc). The warehoused data must be analyzed and summarized to be useful, hence the recent surge of interest in data mining techniques. Data mining algorithms (association rules, decision trees, etc.) try to find "rules" that describe the data in a data set. In this paper, we discuss data set comparison as a data mining technique. A collection of data points (i.e., tuples) have *categorical* attributes and *value* attributes. Each data point represents a unique object (e.g., a customer). The categorical attributes define an a priori classification of the objects into subpopulations, while the value attributes are descriptions of the object's behavior. One of the categorical attributes (the *set definition* attribute) is two-valued, and defines the two sets to be compared. We expect that for each subpopulation, its behavior (distribution) is the same in both data sets. We want to discover which subpopulations are different in the two data sets.

Determining the change in distribution of a large, high-dimensional data set is a difficult problem. Naive approaches, such as comparing componentwise mean values, will not detect many types of distribution changes. Classical statistical tests usually apply to small data sets, and face three problems with large high-dimensional data. First, it is not possible to fit all the data into memory at once. Sampling is a common refuge. Sampling, however well designed, comes with it's pitfalls, particularly where outliers are concerned. Second, even if the samples are small in size, dimensionality becomes an obstacle. Existing techniques for non-parametric analysis of multivariate data such as clustering, multidimensional scaling, principal components analysis and others become expensive as the number of dimensions increase. Furthermore, some of these methods are hard to interpret and visualize. Third, classical multivariate statistics has centered heavily around the assumption of normality, primarily due to analytical convenience in terms of closed form solutions. When the data sets are large, assumptions of homogeneity (e.g., i.i.d. observations) that underlie classical statistical inference might not be valid.

In this paper, we propose a technique, which we call *DataSphere*, for summarizing a data set. Given two very large, high dimensional data sets, a DataSphere partitions the data points into sections, and represents each section as a set of summaries, or *profiles*, that one can use to make meaningful statistically valid comparisons. We apply standard statistical tests to the profile to determine which data sets have changed and where. A DataSphere avoids the curse of dimensionality by using *data pyramids* to represent directional information.

The sectioning technique provides a fine grained summary, without making any a priori distributional assumptions. The summary enables three levels of analysis, as opposed to the one shot test of comparing the overall centers. The DataSphere can be effectively used to: provide an accurate and detailed representation of data sets that can be used for making statistically valid comparisons of centers, distribution in space and interactions among variables; isolate interesting subpopulations; and identify interesting variables

Our algorithms (discussed in the appendix) require only two passes over the the data set. The profiles are small, so data set comparison using the profiles is very fast. The profiles we use are summable. Hence, the profiles can be treated as associative aggregates and managed as a data cube (Gray *et al.* 1996). A single profile collection permits set comparisons at different levels of aggregation. In addition, the data in the profiles permits a deeper analysis of the trends that cause the subpopulations to diverge.

The DataSphere enables fast comparison of data sets. For instance, given two or more data sets, how can we establish in a quick inexpensive fashion whether they are produced by the same statistical process? The question arises often in several contexts. For example, comparing customer behavior by region, by group, by month; comparing the output of automated data collecting devices to establish uniformity; Another important application lies in detecting inconsistencies between data sets. An automated fast screening of data sets for obvious data integrity issues could save valuable analysis time.

In order to answer the above questions, we need to define a metric for similarity of two or more data sets. Ideally, we should be testing the hypothesis that the joint distribution of the variables is the same in the two data sets. However, we address the problem by testing a set of weaker hypotheses that covers different aspects of the distribution, using just the profile information. In this paper, similarity is tested in a two step fashion. First, we test the proportion of points in each of the sections within each subpopulation, using a multinomial test. The test would establish whether the points are distributed in a similar fashion among the sections for the two data sets. Second, we compare the multivariate means of the points that fall in the sections using the Mahalanobis D^2 test. In addition, we could use tests for each variable individually to see which variable is driving the difference.

Problem Definition

We start with some definitions. A *data set* is a collection of tuples $D = \{t_1, t_2, \ldots, t_n\}$. Each tuple t_i is composed of $r = v + c + 1$ attributes. Of the r attributes, v are *value* attributes, c are *categorical* attributes, and one of the attributes is the *set definition* attribute. The categorical attributes define subpopulations of D that will be compared to each other. The value attributes take real values and embed the tuple in a v-dimensional space. The set definition attribute takes its value from the set $\{1, 2\}$. Let S_i be the set of tuples in D such that the set definition attribute has value i. Let C be a particular value of the categorical variables that exists in D. Then we define *subpopulation* $D[C_i]$ to be the tuples in D that have value C in their categorical variables and value i in their set definition attribute, $i \in \{1, 2\}$. Finally we define $V[C_i]$ to be the projection of $D[C_i]$ to the value attributes.

The problem we address is, for each value of C that occurs in D, is the distribution of $V[C_1]$ significantly different than the distribution of $V[C_2]$. In particular, we are interested in answering the following questions.

1. Which subpopulations have changed their behavior in S_2 as compared to S_1.
2. Of the subpopulations which changed their behavior, which sections show the most change?
3. Of the subpopulations which changed their behavior, which variables exhibit the most change?

Data Sectioning

In a previous work, we describe the use of data sectioning for exploratory data analysis (Dasu & Johnson 1997). In this section, we describe briefly the method that produces the profiles used for data set comparison. The DataSphere reduces a large data set D to a smaller and more manageable representation that, ideally, is easy to compute; is easy to interpret; is amenable to further analysis; and requires no prior assumptions.

The method we propose partitions a data set D into k layers $\{D_i\}_{i=1}^k$. Each layer represents a subset of V that is more homogeneous than the entire data set. We have found that a good method for sectioning data into layers is to use the distance of the points from a center of the data cloud. Data points within the same distance range $[d_i, d_{i+1}]$ from the center c, belong to the same layer. Here the index i refers to the layer number, and d_i represents the distance cut off between layers. The partitioning implies that we get to the "typical" data (closest to the center) first and expand outwards to more atypical points. We compute the cutoff points d_i using a fast quantiling algorithm, so each layer contains the same number of data points.

Directional information is incorporated using the concept of *pyramids* (Berchtold, Bohm, & Kriegel 1998). Briefly, a d dimensional set can be partitioned into $2d$ pyramids $P_{i\pm}, i = 1, \ldots, d$ whose tops meet at the center of the data cloud. If $x_1, x_2, \ldots, x_d$ represents a data point p and if $y_1, y_2, \ldots, y_d$ is the corresponding normalized vector, then

$$p \in P_{i+} \quad \text{if } |\, y_i \,|>|\, y_j \,|,\ y_i > 0 \quad j = 1, \ldots, d\ j \neq i$$
$$p \in P_{i-} \quad \text{if } |\, y_i \,|>|\, y_j \,|,\ y_i < 0 \quad j = 1, \ldots, d\ j \neq i$$

We define a *section*, $\mathcal{S}(D_i, P_{i\pm}, \mathcal{C})$ to be the tuples with categorical attributes $\mathcal{C}$ such that the value attributes lie in layer D_i, pyramid $P_{i\pm}$. Each section $\mathcal{S}(D_i, P_{i\pm}, \mathcal{C})$ is summarized using a profile $\mathcal{P}(D_i, P_{i\pm}, \mathcal{C})$. In Figure 1, we show a two dimensional illustration of sectioning with data pyramids. The dotted diagonal lines represent the pyramid boundaries. The black and white dots might correspond to two different subpopulations such as male and female.

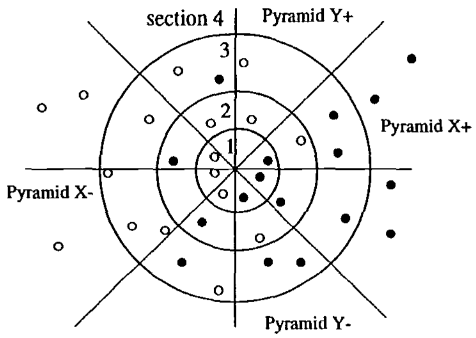

Figure 1: Data Sectioning with Pyramids.

Profile

A profile $\mathcal{P}(D_i, P_{i\pm}, \mathcal{C})$ is a set of statistics, both scalars and vectors, that summarizes the data points in a section. In order to be a member of the profile, a statistic should be easy to compute, be easy to combine across sections, and have the same interpretation when combined across sections or data sets.

These properties ensure that we can compute the summaries in only two passes over the data set. For the purposes of data set comparison, the statistics in the profile are the count of the data points, the vector of means (of the value attributes), and the covariance matrix. Including the count ensures that the mean vector and the covariance matrix are summable. A similar approach is taken in the BIRCH clustering algorithm (Zhang, Ramakrishnan, & Livny 1996), which use profiles to represent clusters in a way that allows them to be combined easily.

A collection of profiles is a *data map* of a data set. The data map has several nice properties. It is compact, transparent, and easy to understand representation of the data set, enabling a visualization of the overall data set. Each profile represents a small homogeneous data space, making profile analysis amenable to classical statistical tests. Outlier analysis is easy, because the outliers are usually in the outermost layer. Finally, analysis of data maps is very fast because of their small size.

Computing the Profiles

Once the data section boundaries are determined, computing profiles is simple. Determining section boundaries requires that a center of the data set be found, and that the quantiles of the distance of points from the center be determined.

Finding the "center" or the "central area" of a multivariate data cloud is an active area of research. Approaches include Liu's simplices (Liu 1990) and the halfplane depth of Rousseeuw and Ruts (Rousseeuw, , & Ruts 1996). Most of these methods suffer when the number of dimensions gets large.

The results in this paper are based on using the vector of means as a center. However, given its extreme sensitivity to outliers, a vector of trimmed means is a superior alternative. We ran experiments to determine the sensitivity of the mean to the proportion of data trimmed. We found that the data center changed only slightly after more than 1 to 2 percent of the per-dimension outlying data points were removed. These points can be accumulated in memory, summed and subtracted from the total mean to get the trimmed mean.

We rank each data point by its Mahalanobis from the center, which helps to scale the data. Computing the Mahalanobis distance requires the standard deviation of the data along each dimension. We compute these quantities while we compute the trimmed mean.

Quantiling the data set into layers requires a partial sort, which in turn requires many passes through the data. We use a recently proposed 1 pass approximate quantiling algorithm recently proposed by Alsabti, Ranka, and Singh (Alsabti, Ranka, & Singh 1997). This algorithm is fast, parallelizable, does not depend on a priori properties of the data set, and provides bounds on the error of the approximation. We modified the quantiling algorithm to compute profiles while the quantiles are computed, so in total computing profiles requires only two passes through the data set. A more precise computation of the profiles can be made with a third pass after the distance quantiles are computed. Profile computation is also easily parallelized, although we did not parallelize the code used in the experiments reported in this paper.

Statistical Tests for Change

Given a collection of section profiles of a subpopulation from two data sets, we need to determine whether the subpopulations of the data sets are similar, without the need to store the raw data. We use two complementary statistical tests that use only summable profile information. The first test is the *Multinomial test for proportions* (Rao 1965), which compares the proportion of points falling into each section within a subpopulation. The second test is the *Mahalanobis* D^2 test (Rao 1965), which we use to establish the closeness of the multivariate means of each layer within each subpopulation, for the two data sets. We use the distance between the multivariate means as a measure of similarity. We note that it is sufficient but not necessary that the joint distribution of the two data sets be the same to pass these tests, and hence we use a strategy of multiple tests.

Application

We applied the DataSphere to data sets obtained from an AT&T data warehouse. The first data set describes customer interactions with an AT&T service. The data consists of over six million observations collected at nine evenly spaced intervals. Every data point consists of two categorical variables (VarA and VarB) and six quantitative variables (Var0 through Var5). VarA defines the type of service or product a customer has, while VarB categorizes the customer's subscription time. The data set is divided into subpopulations based on the levels of VarA and VarB. Each subpopulation is subdivided into 20 layers using the quantiles and each layer into 12 pyramids. A profile is computed for each such segment. We hypothesize that in aggregate, the customers in a subpopulation should have the same behavior from month to month. Any differences are interesting.

We analyzed the data set by applying a DataSphere to each successive pair of monthly data. The total time to compute profiles for the data sets suitable for pairwise comparisons was about 2.5 hours. Once the profiles were computed, performing a comparison test required about 1 second.

In Figure 2, we plot the distribution of data points from one combination of the categorical variables among different section. The X axis indicates increasing layers (one is the central data, 9 is the outlier data, and negative layers indicate negative pyramids), and the Y axis indicates different pyramids. The size of a circle is proportional to the number of points that lie in the indicated section. Note that the points are not distributed evenly among the layers because we plot the distribution for a subset of the data points.

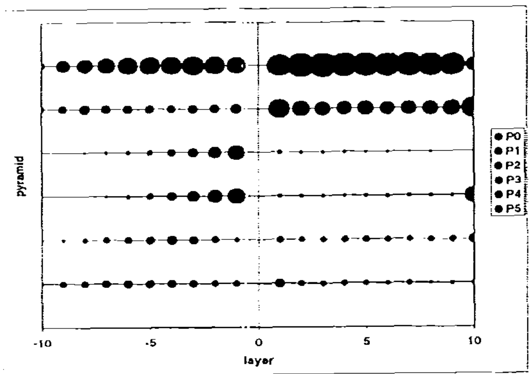

Figure 2: Distribution of points from the customer data set among sections. Dot size indicates the number of points in the section.

We compared data sets from successive observation periods to determine if customer behavior changes between periods. In Figure 3, we plot the results of the multinomial test. The Y axis indicates the earlier of the two data sets compared, i.e. the fourth line indicates the results of comparing the fourth and fifth data sets. From VarB and the collection time of the data set, we can compute the age of a customer, which we plot as the X axis. We selected a value of VarA, and for each age and data set comparison combination we plot a dot whenever the multinomial test indicates a significant difference. This figure indicates that in aggregate new customers change their behavior, then settle into a more stable pattern. Further, customer behavior became more settled in later observations. We obtain similar results for other values of VarA. Such information is useful for determining how best to provide service to AT&T customers.

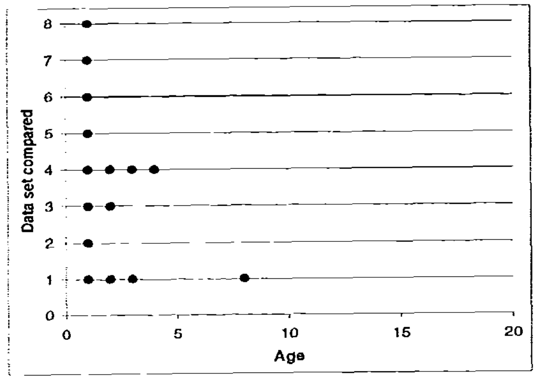

Figure 3: Multinomial test on the customer data for a value of A. Dots indicate significant differences.

We picked the comparison of data sets 8 and 9 (line 8 in Figure 3) and plot the results of the multivariate means test in Figure 4. This comparison also shows that recent customers change their behavior, while older customers are more stable. This chart is similar to charts derived for other successive comparisons and other values of VarA.

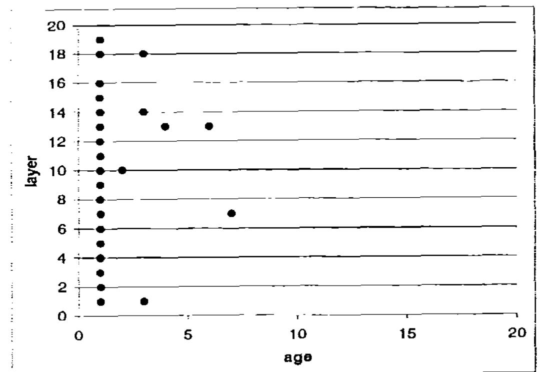

Figure 4: Multivariate means test on the customer data for a value of A. Dots indicate significant differences.

Our second data set is a description of network traffic. The data set consists of 600,000 data points with two categorical variables (VarA and VarB) and three quantitative variables. VarA has a cardinality of four, while VarB has a cardinality of nine. We collected two data sets from two different observation periods.

We are interested in determining if the data sets are different at different observation periods. Our analysis indicated that there is no significant difference between the data sets collected at different time periods, indicating that the characteristics of network use are stable over time.

Next, we are interested in determining if network use is different at different levels of VarA. We repeated our test, but using VarA as the set definition attribute, making pair-wise comparisons. We found some significant differences. For an example, we will focus on comparing data sets where VarA=0 and VarA=2. In Figure 5, we show the results of the multinomial test for each pyramid (a significance level larger than 1 indicates a statistically significant difference). Pyramid P0- shows the most significant difference in data sets. In Figure 6 we plot the distribution of points of P0- to the layers in the pyramid. The distributions of points from differenct values of VarA among layers in P0- show the same basic shape, but some subtle differences. For a more refined analysis, we show the multinomial means test in Figure 7. Significant differences show up only for Pyramid P0-. Hence we conclude that the data sets with VarA=0 and VarA=2 are essentially the same, except for points with low values of Var0.

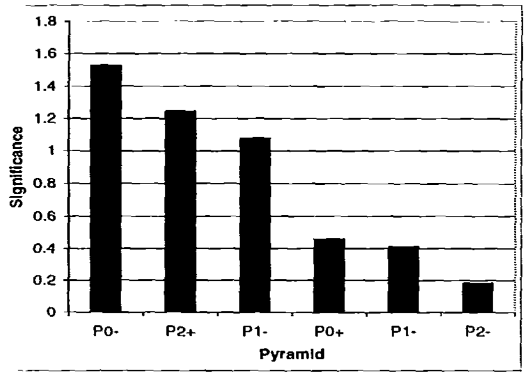

Figure 5: Multinomial analysis of the network data, comparing data sets with VarA = 0 and 2. A significance level of 1 indicates a significant difference.

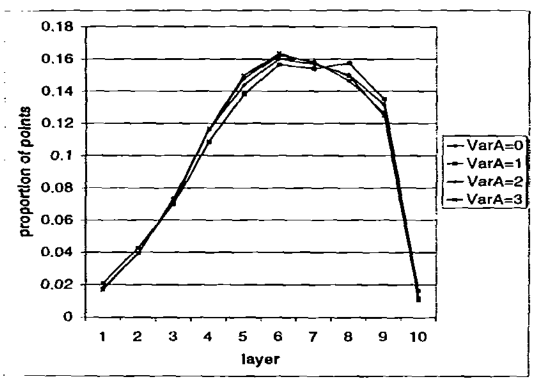

Figure 6: Distribution of network data set points to layers in pyramid P0-.

Conclusions

The DataSphere represents a powerful framework for the analysis and comparison of large data sets. The data elements are divided into homogeneous sections of space. The data sectioning of the DataSphere is scalable with an increasing number of dimensions. While a conventional sectioning by hyperplanes creates 2^d partitions, data pyramids create only $2d$ partitions. incorporating distance information is also scalable. If each dimension is sectioned into l quantiles by hyperplanes, then l^d partitions are created, as opposed to the $2ld$ partitions in a DataSphere. The DataSphere is a kind of dimensionality reduction, taking a d-dimensional data set into 2 dimensional summary.

By sectioning the data sets, we can use well-developed non-parametric statistical tests to compare data sets. Data set comparison can yield interesting and useful information about a dataset, indicating whether it is changing, and how. The data maps produced by the DataSphere are informative summaries of the data set that can be used for visualization among other things.

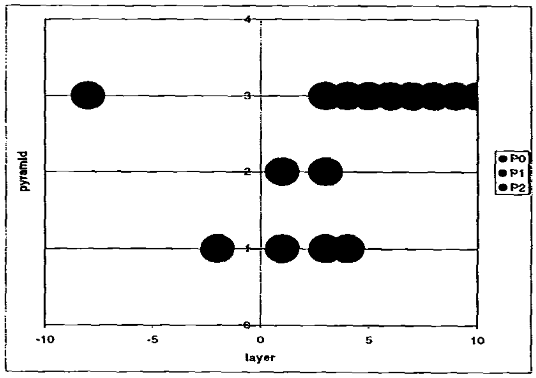

Figure 7: Multivariate means analysis of the network data, comparing data sets with VarA=0 and VarA=2. A circle indicates a significant difference.

References

Alsabti, K.; Ranka, S.; and Singh, V. 1997. A one-pass algorithm for accurately estimating quantiles for disk-resident data. In *Int'l. Conf. on Very Large Databases.*

Berchtold, S.; Bohm, C.; and Kriegel, H. 1998. The pyramid-tree: Breaking the curse of dimensionality. In *Proc. 1998 ACM SIGMOD Conf.*, 142–153.

Dasu, T., and Johnson, T. 1997. An efficient method for representing, analyzing and visualizing massive high dimensional data sets. In *Interface '97.*

Gray, J.; Bosworth, A.; Layman, A.; and Pirahesh, H. 1996. Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In *Proc. of the 12th Intl. Conf. on Data Engineering*, 152–159.

Liu, R. 1990. On a notion of data depth based random simplices. *Annals of Statistics* 18:405–414.

Rao, C. 1965. *Linear Statistical Inference and Its Applications.* John Wiley.

Rousseeuw, P.; ; and Ruts, I. 1996. Computing depth contours of bivariate clouds. *Computational Statistics and Data Analysis* 23:153–168.

Zhang, T.; Ramakrishnan, R.; and Livny, M. 1996. BIRCH: An efficient data clustering method for very large databases. In *Proc. 1996 ACM SIGMOD Conf.*, 103–114.