Towards Enabling Outlier Detection in Large, High Dimensional Data Warehouses*

Konstantinos Georgoulas and Yannis Kotidis

Athens University of Economics and Business
76 Patission Street, Athens, Greece
{kgeorgou,kotidis}@aueb.gr

Abstract. In this work we present a novel framework that permits us to detect outliers in a data warehouse. We extend the commonly used definition of distance-based outliers in order to cope with the large data domains that are typical in dimensional modeling of OLAP datasets. Our techniques utilize a two-level indexing scheme. The first level is based on Locality Sensitivity Hashing (LSH) and allows us to replace range searching, which is very inefficient in high dimensional spaces, with approximate nearest neighbor computations in an intuitive manner. The second level utilizes the Piece-wise Aggregate Approximation (PAA) technique, which substantially reduces the space required for storing the data representations. As will be explained, our method permits incremental updates on the data representation used, which is essential for managing voluminous datasets common in data warehousing applications.

1 Introduction

Assuring quality of data is a fundamental task in information management. It becomes even more critical in decision making applications, where erroneous data can mislead to disastrous reactions. The data warehouse is the cornerstone of an organization's information infrastructure related to decision support. The information manipulated within a data warehouse can be used by a company or organization to generate greater understanding of their customers, services and processes. Thus, it is desirable to provision for tools and techniques that will detect and address potential data quality problems in the data warehouse.

It is estimated that as high as 75% of the effort spent on building a data warehouse can be attributed to back-end issues, such as readying the data and transporting it into the data warehouse [1]. This is part of the Extract Transform Load (ETL) processes, that extract information pieces from available sources to a staging area, where data is processed before it is eventually loaded in the data warehouse local tables. Processing at the data staging area includes cleansing, transformation, migration, scrubbing, fusion with other data sources etc.

In this paper, we propose a novel framework for identifying outliers in a data warehouse. Outliers are commonly defined as rare or atypical data objects that do not behave

* This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

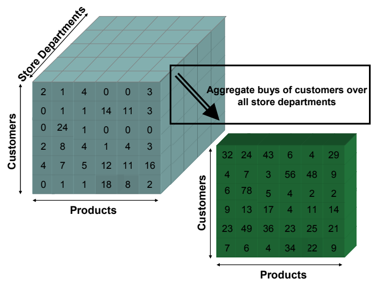
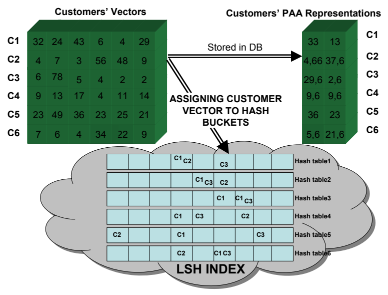


Fig. 1. Customer dimension projected onto the Product dimension



Customers' Vectors
Customers' PAA Representations
Stored in DB
ASSIGNING CUSTOMER VECTOR TO HASH BUCKETS
LSH INDEX

Fig. 2. Framework's Overview

like the rest of the data. Often, erroneous data points appear as outliers when projected on a properly derived feature space. In our work, we exploit the dimensional modeling used in a data warehouse and let the user examine the data under selected dimensions of interest. This way, our definition of what constitutes an outlier has a natural interpretation for the policy makers that interact with this data. Moreover, our techniques are tailored for the massive and periodic schedule of updates that occur with each ETL process. Clearly, techniques that require substantial pre- or post- processing of data are not suitable for handling massive datasets such as those in a data warehouse.

2 A Framework for Detecting Outliers in a Data Warehouse

Given a data warehouse with multiple dimensions d_1, d_2,..., d_n, each organized by different hierarchy levels h_k the data warehouse administrator my select a pair (d_{aggr}, h_{aggr_level}) so as to define the requested aggregation and, similarly, a pair (d_{proj}, h_{proj_level}) in order to denote the space that these aggregates should be projected upon. For instance the aggregate dimension can be customer at the hierarchy level of customer-type and the projected dimension product at the product-brand level. These pairs indicate our intention to compare different customer types based on cumulative sales of the brand of products they buy in order to search for outliers. Clearly, a data warehouse administrator may define multiple such pairs of dimensions in order to test the data for outliers. An example presented in Figure 1 where the customer (aggregate) dimension is projected onto the product dimension. This projection leads to a high dimensional vector for each customer that summarizes all his buys over the whole list of products.

An $O(D, M)$ distance based outlier is defined [2] as a data item O in a dataset with fewer than M data items within distance D from O. The definition, in our domain, suggests that range queries need to be executed in the data space defined by the projected dimension in order to compute the number of data items that lay inside a range of D from item O. As has been explained, the projected space can have very high dimensionality (i.e. equal to the number of all products in the data warehouse, which is in the order of thousands), which renders most multidimensional indexing techniques ineffectual, due to the well documented curse of dimensionality [3].

In order to address the need to compare data items on a high-dimensionality space when looking for outliers, we adapt a powerful dimensionality reduction technique called LSH [4]. LSH generates an indexing structure by evaluating multiple hashing functions over each data item (the resulting vector when projecting a customer on the space of products she buys). Using the LSH index, we can estimate the k nearest neighbors of each customer and compute outliers based on the distances of each customer from its k neighbors. We thus propose an adapted approximate evaluation of distance-based outliers that treats a data item O as an outlier if less than M of its k nearest neighbors are within distance M from O. Please notice that this alternative evaluation permits us to utilize the LSH index for a k-NN query (with $k > M$) and restrict the range query on the k results retrieved from the index. Thus, the use of the LSH index permits effective evaluation of outliers, however it introduces an approximation error, because of collisions introduced by the hashing functions. There have been many proposals on how to tune and increase performance of LSH (e.g. [5, 6]), however such techniques are orthogonal to the work we present here.

The use of LSH enables computation of outliers by addressing the curse of dimensionality. Still, an effective outlier detection framework needs to address the extremely high space required for storing the resulting data vectors. The size of these vectors is proportional to the size of a data cube slice on the selected pair of dimensions. Moreover, these vectors need to be updated whenever the data warehouse is updated with new data. We address both these issues (space overhead, update cost) using the PAA representation instead of the original vectors. Utilizing PAA, we store vectors of lower dimensionality than the real ones, thus gaining in space. We can also compute the distances between each data item and its nearest neighbours through their PAA representations much faster than using the real data items without losing too much in accuracy as we will show in our experimental evaluation. PAA [7] represents a data item of length n in $\mathbb{R}^N$ space (where $n > N$) achieving a dimensionality reduction ratio N:n. Given that each data item X is a vector with coordinates $x_1, .., x_n$, its new representation will be a new vector $\bar{X}$ of length N and coordinates the mean values of the N equisized fragments of vector X. So according to PAA a vector $X = x_1, .., x_n$ is represented by $\bar{X} = \bar{x}_1, .., \bar{x}_N$ where $\bar{x_i} = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j$.

Beside the space savings provided by PAA, its adaptation has another important advantage in our application. Because of its definition, PAA representations are linear projections that permit incremental updates whenever new data arrives at the data warehouse. Let PAA_{old} denote the PAA vector of a customer and PAA_{delta} the PAA representation of the customer's buys in the newly acquired updates. Then, in order to compute the new representation PAA_{new} for this customer we can simply add the two vectors, i.e. PAA_{new}=PAA_{old}+PAA_{delta}. This property is vital for data warehouses, where incremental updates are of paramount importance [8].

3 Experiments and Concluding Remarks

In our experimental evaluation, we used a clustered synthetic dataset that represents orders of 10,000 customers over a list of 1200 products. Each cluster contains customers with similar behavior. In particular, the customers within each cluster have a randomly

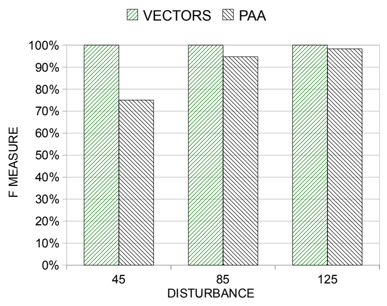


Fig. 3. F-Measure

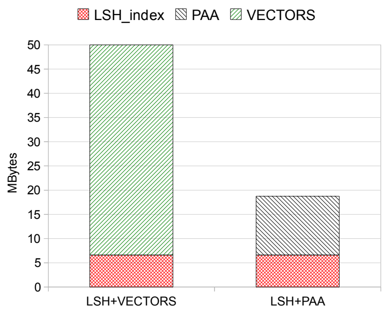


Fig. 4. Space for the LSH Index, the Vectors or PAAs

(pre)selected set of "hot" products that represent 20% of the whole product list. For a customer in the cluster, 80% of her orders are on that 20% subset of products. Different clusters have different set of hot products. The frequencies of customers' orders follow the normal distribution with different mean values per cluster. In order to evaluate the performance of our method we injected in the dataset outliers in the form of spurious orders. We created three infected datasets. In the first one, the spurious orders add low disturbance (measured by the number of spurious orders) to the original data. while in the second medium and in the third large. In Figure 3 we depict the f-measure ($\frac{2 \times recall \times precision}{recall + precision}$) in detecting the injecting outliers for the tree datasets. We compare two variants. The first uses the LSH index and the data vectors generated by projecting the customers on the product dimension. The second setup, instead of the original vectors, it only stores their PAA representations of one quarter of the original vector length. In the Figure we observe that the accuracy of the PAA method is quite similar to the other method that stores the actual vectors, while the storage of PAA is much smaller, as it shown in Figure 4. Moreover, the size of the LSH index is very small, while its accuracy in computing distance-based outliers is at least 98%.

References

1. Kimball, R.: The Data Warehouse Toolkit. John Wiley & Sons (1996)
2. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online Outlier Detection in Sensor Data Using Non-Parametric Models. In: VLDB. (2006)
3. Korn, F., Pagel, B.U., Faloutsos, C.: On the 'Dimensionality Curse' and the 'Self-Similarity Blessing'. IEEE Trans. Knowl. Data Eng. **13**(1) (2001) 96–111
4. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: FOCS, IEEE Computer Society (2006) 459–468
5. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search. In: VLDB. (2007) 950–961
6. Georgoulas, K., Kotidis, Y.: Distributed Similarity Estimation using Derived Dimensions. VLDB J. **21**(1) (2012) 25–50
7. Keogh, E.J., Chakrabarti, K., Pazzani, M.J., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowl. Inf. Syst. **3**(3) (2001) 263–286
8. Roussopoulos, N., Kotidis, Y., Roussopoulos, M.: Cubetree: Organization of and Bulk Updates on the Data Cube. In: SIGMOD Conference. (1997) 89–99