An OLAM Operator for Multi-Dimensional Shrink

Stefano Rizzi, Department of Computer Science and Engineering, University of Bologna, Bologna, Italy
Matteo Golfarelli, Department of Computer Science and Engineering, University of Bologna, Cesena, Italy
Simone Graziani, Department of Computer Science and Engineering, University of Bologna, Cesena, Italy

ABSTRACT

Shrink is an OLAM (On-Line Analytical Mining) operator based on hierarchical clustering, and it has been previously proposed in mono-dimensional form to balance precision with size in the visualization of cubes via pivot tables during OLAP analyses. It can be applied to the cube resulting from a query to decrease its size while controlling the approximation introduced; the idea is to fuse similar facts together and replace them with a single representative fact, respecting the bounds posed by dimension hierarchies. In this paper the authors propose a multi-dimensional generalization of the shrink operator, where facts are fused along multiple dimensions. Multi-dimensional shrink comes in two flavors: lazy and eager, where the bounds posed by hierarchies are respectively weaker and stricter. Greedy algorithms based on agglomerative clustering are presented for both lazy and eager shrink, and experimentally evaluated in terms of efficiency and effectiveness.

Keywords: Hierarchical Clustering, OLAP, OLAP Visualization, Pervasive BI, Query Personalization

1. INTRODUCTION

Business intelligence (BI) is a bundle of tools and techniques for timely detecting key business factors and effectively solving strategic decisional problems. The “new wave” of BI, often called BI 2.0, aims at specifically addressing more sophisticated user needs; among the characterizing trends of BI 2.0 we focus on pervasive BI, where information can be easily and timely accessed through devices with different computation and visualization capabilities, and with sophisticated and customizable presentations, by everyone in the organization (Rizzi, 2012).

In the context of pervasive BI, one of the key factors that rule the effectiveness of analysis is the achievement of a satisfactory (from the users’ viewpoint) compromise between the precision and the size of the information being displayed while analyzing multi-dimensional cubes. The OLAP paradigm gives a significant

DOI: 10.4018/IJDWM.2015070104
support in this direction by enabling users to interactively slice, dice, and aggregate cube facts, but this is not always sufficient: more detail gives more information, but at the risk of missing the overall picture, while focusing on general trends may prevent users from observing specific small-scale phenomena (Marcel, Missaoui, & Rizzi, 2012). This is also strictly related to the “information flooding” problem, that may happen because the user drilled down a cube up to a very detailed level, where a huge number of facts are to be returned. In this case, it may be very hard for the user to browse and analyze the results, especially if the device used has limited visualization and data-transmission capabilities.

Different approaches can be taken to cope with this issue. For instance, in \textit{query personalization} there is an attempt to tune the size and pertinence of facts returned by considering the users’ preferred aggregation levels, measures, and slices (Golfarelli, Rizzi, & Biondi, 2011). In \textit{approximate query answering}, the focus is on quickly returning an answer at the price of some imprecision in the returned values (Vitter & Wang, 1999). In \textit{intensional query answering}, the set of facts returned by a query is summarized with a concise description of the properties shared by those facts (Marcel, Missaoui, & Rizzi, 2012). Other papers couple the OLAP paradigm with data mining techniques to create an \textit{OLAM approach} where cubes can be mined “on-the-fly” to extract concise patterns for user’s evaluation (Han, 1997).

The \textit{shrink} approach is a form of OLAM based on hierarchical clustering, specifically aimed at balancing precision with size in visualization of multi-dimensional cubes via pivot tables like the one shown in Figure 1. The shrink operator can be applied during an OLAP session to the cube resulting from a query to decrease its size while controlling the approximation introduced, like sketched in Figure 2. The idea is to fuse similar facts together and replace them with a single representative fact (computed as their average), respecting the bounds posed by dimension hierarchies.

A mono-dimensional version of the shrink operator has been proposed by Golfarelli, Graziani, & Rizzi (2014). In that work one shrink dimension is explicitly chosen by the user, and cube slices are fused together along that dimension until a user-specified precision/size trade-off is achieved. Though the mono-dimensional version has been shown to be quite effective and efficient in delivering compact visualizations of cubes, it suffers from two main drawbacks: (i) since the shrink dimension is fixed a priori, some possibly more effective directions for shrinking may be lost; and (ii) the approach is subject to the user’s discretion in choosing the shrink dimension.

In this paper we propose a multi-dimensional generalization of the shrink operator, where facts are fused along multiple dimensions. Multi-dimensional shrink come in two flavors: lazy and eager.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{pivot_table.png}
\caption{A simple pivot table showing data per city and year}
\end{figure}
Kernal Width Selection for SVM Classification: A Meta-Learning Approach
[www.igi-global.com/article/kernal-width-selection-svm-classification/1760?camid=4v1a](www.igi-global.com/article/kernal-width-selection-svm-classification/1760?camid=4v1a)

Efficient Compression and Storage of XML OLAP Cubes
Doulkifli Boukraa, Mohammed Amin Bouchoukh and Omar Boussaid (2015).
*International Journal of Data Warehousing and Mining* (pp. 1-25).
[www.igi-global.com/article/efficient-compression-and-storage-of-xml-olap-cubes/129522?camid=4v1a](www.igi-global.com/article/efficient-compression-and-storage-of-xml-olap-cubes/129522?camid=4v1a)