Software Warehouse

Honghua Dai
Deakin University, Australia

INTRODUCTION

A software warehouse is a facility providing an effective and yet efficient mechanism to store, manage, and utilize existing software resources (Dai, 2003, 2004a, 2004b; Dai & Li, 2004). It is designed for the automation of software analysis, testing, mining, reuse, evaluation, and system design decision making. It makes it easier to make use of existing software for solving new problems in order to increase software productivity and to reduce the cost of software development.

By using a software warehouse, software assets are systematically accumulated, deposited, retrieved, packaged, managed, and utilized, driven by data mining and OLAP technologies. The design perspectives and the role of a software warehouse in modern software development are addressed in Dai (2003).

BACKGROUND

With the dramatic increase in the amount and size of available software, it is naturally important to consider an effective and yet efficient way to store, manage, and make best use of existing software. A software warehouse is proposed to meet such a demand. In many cases, software analysis is a necessary step to system development for new applications. Such analysis is required for the provision of answers or solutions to many challenging and practical questions, such as the following:

1. Is there any software available for solving a particular problem? What are the software products? Which one is better?
2. In history, how did people solve a similar problem?
3. What are the existing software components that can be used in developing a new system?
4. What is the best design for a set of given system requirements?

To provide a satisfactory answer to these questions, the following conditions need to be met:

1. A comprehensive collection of both historical and current software.
2. An architecture and organization of the collected software are needed for effective and yet efficient access to the target software.
3. A reliable and feasible management and access strategy is needed for management and for making use of the software.

In short, the significance of establishing a software warehouse includes:

1. Effective and efficient software storage and management.
2. Software design study.
3. Software development audit and management.
4. Software reuse.
5. Software analysis and software review.
6. Software reverse engineering using data mining.
7. Software development decision making.
8. Software design recovery to facilitate software design.
9. Support automatic software engineering.
10. Provide essential material to software factory in an organized, systematic, effective, and efficient way.

Since the invention of computers, almost all software analysis tasks have been completed by human experts. Such analysis normally was done based on a very small portion of the information, due to the limitation of available resources. Such resource limitation is not due to the lack of resources but to the lacking of a way to effectively manage the resources and make use of them. With regard to software development, in today's software industry, the analysis, design, programming, and testing of software systems are done mostly by human experts, while automation tools are limited to the execution of preprogrammed action only. Evaluation of system performance is also associated with a considerable effort by human experts, who often have imperfect knowledge of the environment and the system as a whole.

SOFTWARE WAREHOUSE TECHNOLOGY

A software warehouse is an extension of the data warehouse (Barquin & Edelstein, 1997; Dai, Dai & Li, 2004).

Spectral Methods for Data Clustering

Wenyuan Li
Nanyang Technological University, Singapore

INTRODUCTION

With the rapid growth of the World Wide Web and the capacity of digital data storage, tremendous amount of data are generated daily from business and engineering to the Internet and science. The Internet, financial real-time data, hyperspectral imagery, and DNA microarrays are just a few of the common sources that feed torrential streams of data into scientific and business databases worldwide. Compared to statistical data sets with small size and low dimensionality, traditional clustering techniques are challenged by such unprecedented high volume, high dimensionality complex data. To meet these challenges, many new clustering algorithms have been proposed in the area of data mining (Han & Kambr, 2001).

Spectral techniques have proven useful and effective in a variety of data mining and information retrieval applications where massive amount of real-life data is available (Deerwester et al., 1990; Kleinberg, 1998; Lawrence et al., 1999; Azar et al., 2001). In recent years, a class of promising and increasingly popular approaches — spectral methods — has been proposed in the context of clustering task (Shi & Malik, 2000; Kannan et al., 2000; Meila & Shi, 2001; Ng et al., 2001). Spectral methods have the following reasons to be an attractive approach to clustering problem:

- Spectral approaches to the clustering problem offer the potential for dramatic improvements in efficiency and accuracy relative to traditional iterative or greedy algorithms. They do not intrinsically suffer from the problem of local optima.
- Numerical methods for spectral computations are extremely mature and well understood, allowing clustering algorithms to benefit from a long history of implementation efficiencies in other fields (Golub & Loan, 1996).
- Components in spectral methods have the naturally close relationship with graphs (Chung, 1997). This characteristic provides an intuitive and semantic understanding of elements in spectral methods. It is important when the data is graph-based, such as links of WWW, or can be converted to graphs.

In this paper, we systematically discuss applications of spectral methods to data clustering.

BACKGROUND

To begin with the introduction of spectral methods, we first present the basic foundations that are necessary to understand spectral methods.

Mathematical Foundations

Data is typically represented as a set of vectors in a high-dimensional space. It is often referred as the matrix representation of the data. Two widely used spectral operations are defined on the matrix.

- EIG(A) operation: Given a real symmetric matrix $A_{n\times n}$, if there is a vector $x \in R^n \neq 0$ such that $Ax=\lambda x$ for some scalar λ, then λ is called the eigenvalue of A with corresponding (right) eigenvector x. EIG(A) is an operation to compute all eigenvalues and corresponding eigenvectors of A. All eigenvalues and eigenvectors are real, that is, guaranteed by Theorem of real schur decomposition (Golub & Loan, 1996).
- SVD(A) operation: Given a real matrix $A_{m\times n}$, similarly, there always exists two orthogonal matrices $U \in R^{m\times m}$ and $V \in R^{n\times n}$ ($U^TU= I$ and $V^TV=I$) to decompose A to the form $A=USV^T$, where $S = \text{diag}(\sigma_1,\cdots,\sigma_r) \in R^{r\times r}$, r=rank(A) and $\sigma_1 \ge \sigma_2 \cdots \ge \sigma_r = \cdots = \sigma_n = 0$. Here, the σ_i are the singular values of A and the first r columns of U and V are the left and right (respectively) singular vectors of A. SVD(A) is called Singular Value Decomposition of A (Golub & Loan, 1996).

Typically, the set of eigenvalues (or singular values) is called the spectrum of A. Besides, eigenvectors (or singular vectors) are the other important components of spectral methods. These two spectral components have