Tutorials Program

Tutorial 1: Wavelets and Their Applications in Databases

Instructors: Daniel A. Keim and Martin Heczko

The roots of wavelet theory reach back to the end of the 19th century. The so-called, developed in 1909 by A. Haar, still serves as the foundation of modern wavelet theory. It took a long time, however, until the wavelet-based hierarchical data decomposition found its widespread application in computer science. Wavelets are seen as the “(re)discovery of the last decade” in Computer Graphics and, in the meantime, they are used in a wide variety of applications including a number of diverse database applications. Examples are: similarity search, data compression, dimensionality reduction, time series analysis, and data clustering. The wavelet theory is well founded and of very high practical impact. The large number of advantages include the strict hierarchical and multiresolutional nature of the wavelet decomposition, the linear time and space complexity of the wavelet transformations, and the high flexibility of different wavelet functions, leading to considerably more effective and efficient solutions of well-known problems. The goal of the tutorial is to make the valuable knowledge about wavelets available to a broader portion of the database research community in order to increase the benefits, which can be gained from using wavelets. The tutorial gives an overview of recent database research projects, which already benefit from the advantages of wavelets. Among the numerous successful applications are: approximation and clustering techniques for large databases, similarity search in image and time series databases, and even standard database applications such as selectivity estimation. The tutorial is structured as follows: After a brief motivation of wavelets, we provide an application-oriented overview of the foundations of wavelet theory and discuss their general advantages. Next, we provide a brief overview of some interesting standard applications of wavelets. In the main portion of the tutorial, we then focus on the recent applications of wavelets in the database area, providing a detailed description and discussion of their main contributions. In concluding the tutorial, we discuss the impact of wavelets for the database area and outline potential future research directions and applications.

Tutorial 2: Similarity Join

Instructor: Christian Boehm

Larger and larger amounts of data are collected and stored in databases, increasing the need for efficient and effective analysis methods to make use of the information contained implicitly in the data. Innumerable approaches for the various data mining tasks such as association rule discovery, classification, clustering, regression, and outlier detection have been proposed from different research communities like statistics and machine learning. An important aspect of contributions from the database research is the scalability of algorithms when facing large data sets. The relational join is one of the most important and most powerful operators of a commercial database system. Both database vendors as well as academic researchers have made every possible effort to implement the join efficiently. Even the whole area of relational query optimization deals primarily with different aspects of joins such as optimizing the join order or selecting the optimal algorithm and parametrization for each join. Recently, it has been recognized that join operations are also a powerful database primitive to support data mining algorithms. Joins do not only provide an easy and universal means to tackle the scalability problem. Moreover, using highly optimized join operations can even accelerate existing mining algorithms by large factors. Of particular interest are mining algorithms, which are based on the notion of the point density. Examples include various clustering algorithms, outlier detection, time series analysis, spatial trend detection, etc. Such algorithms typically issue a large number of similarity queries (i.e. range queries or (k-) nearest neighbor queries) in a multidimensional or metric feature space. Since many queries can be executed simultaneously, the query set can be rewritten as a similarity join between the set of the original query points and the set of the database points. Some data mining algorithms even evaluate a similarity query for each database point. Substituting this massive query set by a single similarity self-join offers a particularly high optimization potential. Due to the high relevance of the similarity join, a large number of different algorithms have been proposed. Our tutorial reviews the state-of-the-art in this area of research. The structure of our tutorial is guided by the intention to bring together the experts of data mining and query processing. First, we will introduce
several representative data mining algorithms and show how to rewrite them on top of a similarity join. Starting from this, we will categorize the different types of similarity joins such as distance range joins, k-nearest neighbor joins, etc. The major part of the tutorial is then dedicated to the various algorithms for evaluating the similarity join. Next, we will go into the details of cost modeling and parameter optimization. A perspective on future research directions will conclude the tutorial.

**Tutorial 3: Data Warehouse Design**  
**Instructors: Stefano Rizzi and Matteo Golfarelli**

Building a data warehouse (DW) for an enterprise is a huge and complex task, which requires accurate planning aimed at devising satisfactory answers to organizational and architectural questions. Despite the pushing demand for working solutions coming from enterprises and the wide offer of advanced technologies from producers, few attempts toward devising a specific, structured methodology for data warehouse design have been made. On the other hand, the statistic reports related to DW project failures state that a major cause lies in the absence of a global view of the design process: in other terms, in the absence of a design methodology. The tutorial aims at introducing a methodological framework for design, addressing the main topics in conceptual, logical and physical design of the data marts, which, assembled in a bottom-up fashion, concur in creating the data warehouse. Among the conceptual models proposed in the literature, we will focus in particular on the Dimensional Fact Model (DFM) as a support for the whole design process.

**Outline:**

The tutorial aims at enabling the participants to understand the basics in data warehousing and the underlying design principles, and more specifically to introduce them to the most critical issues in conceptual, logical and physical design.

This will be achieved by dealing with the following topics:

1. Introduction to Data Warehousing: from operational databases to data warehouses; the multidimensional model; architectural issues; ROLAP and MOLAP solutions.
2. Conceptual design of Data Warehouses: E/R-based models; the Dimensional Fact Model; conceptual design from the operational schemes.
3. Workload-based logical design for ROLAP: defining the workload; star and snowflake schemes; view materialization and fragmentation.
4. Indices for physical design: B-trees, bitmap indices, join indices; selecting the indices for the data mart.

In order to increase the educational efficacy, topic 2 will be supported by a CASE tool designed by the authors.

**Target audience and background:** The tutorial is directed to enterprise analysts and designers, as well as to researchers wishing to get acquainted with data warehousing from the designer’s point of view. A good background on the relational model and on the Entity/Relationship model is required.

**Tutorial 4: Next Generation of Data Mining Tools, Using SDV and Fractals**  
**Instructor: Christos Faloutsos**

What patterns can we find in a bursty Web traffic? On the Web graph itself? How about the distributions of galaxies in the sky, or the distribution of a company’s customers in geographical space? How long should we expect a nearest-neighbor search to take, when there are 100 attributes per patient or customer record? The traditional assumptions (uniformity, independence, Poisson arrivals, Gaussian distributions), often fail miserably. Should we give up trying to find patterns in such settings? This tutorial focuses on two powerful but less known tools, namely on the Singular Value Decomposition (SVD) and on Fractals. SVD is a provably optimal method for dimensionality reduction and feature selection; it is the engine-under-the-hood for breakthrough concepts like the Latent Semantic Indexing (LSI), the Karhunen-Loeve transform and the Kleinberg algorithm for Web site importance ranking, to name a few. Fractals, self-similarity and power laws are extremely successful in
describing real datasets (coast-lines, rivers basins, stock-prices, brain-surfaces, Web and disk traffic, to name a few). Although both tools are impressively general and useful, their introductory papers are typically not tailored toward a database audience, rendering them inaccessible. This tutorial exactly tries to remedy the situation. Specifically, it has two goals: (a) to introduce the most useful concepts from SVD and Fractals, emphasizing the intuition behind them, and avoiding the unnecessary mathematical intricacies and (b) to illustrate the usefulness of SVD and fractals for a variety of database and data mining applications.

**Target Audience:** Researchers working on spatial access methods, on query optimization, and on data mining.

**Prerequisites:** None.

**Benefits to Participants:** The participants will gain the intuition behind these powerful tools, and they will get exposed to numerous settings where SVD and fractals solved the data mining/data base problem at hand.

**Tutorial 5: XML**

**Instructors:** Dana Florescu and Jerome Simeon

XML is a document mark-up language designed for data exchange between Web applications. Developed and promoted by the World Wide Web Consortium (W3C), XML technology has attracted a lot of attention over the last two years, both from the industry and from the research community. But if the XML world was at first limited to a unique, simple, self-contained specification (XML 1.0), the sudden interest for XML has generated an incredible amount of activity. Now, with a multitude of inter-related standards, industry proposals, and research literature, finding its way in the XML maze has become a challenging enterprise. The objective of this tutorial is to draw a clear, simple and meaningful panorama of existing standards and research contributions related to XML. To decode the various XML activities, we will see them through database glasses: we will look at the development of XML technology as a data management problem. The tutorial will be organized in three parts: data models for XML, data definition languages for XML and data manipulation languages for XML. For each of these three aspects, we will introduce the standards and explain their relationship to current state of research. We will notably cover the following material from the W3C: XML 1.0, XML Query Data Model, XML Infoset, XML Schema, DTDs, XPath, XSLT, XML Query Algebra and XML Query Language.

**Tutorial 6: Publish and Subscribe Systems**

**Instructors:** Arno Jacobsen and Francois Llirbat

The publish and subscribe paradigm is a simple to use interaction model that consists of information providers, who publish events to the system, and of information consumers, who subscribe to events of interest within the system. The publish and subscribe system ensures the timely notification of subscribers upon event occurrence. The publish and subscribe paradigm has recently gained great interest in the database community as a solution methodology for information dissemination applications with which the classical request/reply-style communication model (a.k.a. client/server model) fails to cope adequately. Information dissemination applications include applications such as: stock, sports and news tickers, tourist, travel and traffic information systems, as well as emergency notification systems. Common to all of these applications is the need to continuously collect and integrate data distributed among a large set of users, sites, and applications. The application must filter and deliver relevant data to interested users and applications in a timely manner. The classical pull-based approach is not suited to implement these applications for two reasons. First, to approximate “real time” behavior a client would need to continuously increase its frequency of information requests leading to server resource and network overload and congestion. Second, a pure pull-based solution does not support a high volatility of information sources, since new sources can only be discovered by searching the network. This may be very demanding when the network is large and is impossible in mobile and wireless environments where a continuous network access may not always be possible. The objective of this tutorial is twofold. One, we aim to present a comprehensive survey of application domains, system design choices, and existing system implementations to understand scope and applicability of this paradigm. Two, we aim to discuss the strengths and weaknesses of these systems and evaluate what still needs to be done to make the publish and subscribe paradigm a practical solution for large-scale information dissemination applications. To achieve this, the tutorial is organized along four main axes: applications, publish and subscribe systems, algorithms deployed in these systems, and open research questions.