Asymmetric Page Split Generalized Index Search Trees for Formal Concept Analysis

Ben Martin[1] and Peter Eklund[2]

[1] Information Technology and Electrical Engineering
The University of Queensland
St. Lucia QLD 4072, Australia
monkeyiq@users.sourceforge.net
[2] School of Economics and Information Systems
The University of Wollongong
Northfields Avenue, Wollongong, NSW 2522, Australia
peklund@uow.edu.au

Abstract. Formal Concept Analysis is an unsupervised machine learning technique that has successfully been applied to document organisation by considering documents as objects and keywords as attributes. The basic algorithms of Formal Concept Analysis then allow an intelligent information retrieval system to cluster documents according to keyword views. This paper investigates the scalability of this idea. In particular we present the results of applying spatial data structures to large datasets in formal concept analysis. Our experiments are motivated by the application of the Formal Concept Analysis idea of a virtual filesystem [11,17,15]. In particular the libferris [1] Semantic File System. This paper presents customizations to an RD-Tree Generalized Index Search Tree based index structure to better support the application of Formal Concept Analysis to large data sources.

1 Introduction: Information Retrieval and Formal Concept Analysis

Formal Concept Analysis [10] is a well understood technique of data analysis. Formal Concept Analysis takes as input a binary relation I between two sets normally referred to as the object set G and attribute set M and produces a set of "concepts" which are a minimal representation of the natural clustering of the input relation I. A concept is a pair $(A \subseteq G, B \subseteq M)$ such that A cannot be enlarged without reducing $|B|$ and vice versa. The application of Formal Concept Analysis to non binary relations, such as a table in a relational database, can be achieved by first transforming or "scaling" the input data into a binary relation [10,18].

A common approach to document and information retrieval using Formal Concept Analysis is to convert associations between many-valued attributes and objects into binary associations between the same objects G and new attributes M. For example, a many-valued attribute showing a person's income as numeric

F. Esposito et al. (Eds.): ISMIS 2006, LNAI 4203, pp. 218–227, 2006.

data may be converted into three attributes: low-income, middle-income and high-income which are then associated with the same set of people G. The binary relation I between $g \in G$ and $m \in M =$ {low-income, middle-income, high-income} is formed by asserting gIm depending on the level of the numeric income value of person g. The binary relation I is referred to as a formal context in Formal Concept Analysis.

This is the approach adopted in the ZIT-library application developed by Rock and Wille [20] as well as the Conceptual Email Manager [6]. The approach is mostly applied to static document collections (such as newsclassifieds) as in the program Rfca [5] but also to dynamic collections (such as email) as in Mail-Sleuth [2] and files in the Logical File System (LISFS) [17]. In all but the latter two the document collection and full-text keyword index are static. Thus, the FCA interface consists of a mechanism for dynamically deriving binary attributes from a static full-text index. Many-valued contexts are used to materialize formal contexts in which objects are document identifiers.

A specialised form of information retrieval system is a virtual file system [11,17,15]. The idea of using Formal Concept Analysis to generate a virtual filesystem was first proposed by using a logical generalization of Formal Concept Analysis [8,7] and in more recent work using an inverted file index and generating the lattice closure as required by merging inverted lists [17]. In such a system scalability becomes a critical concern because they deal with potentially millions of documents and hundreds/thousands of attributes. For this reason we investigate other forms of indexing data structure more fit to the systems scalability requirement and in particular an analysis of spatial indexing methods which have been applied in the libferris virtual file system [15].

2 Indexing and Scalable Knowledge Processing with Formal Concept Analysis

An index structure allows one to quickly find data by a retrieval key. For example the key might be a person's email address and the data could be a record with other information about that person. The index stores only the key or parts thereof and links to the information itself. Thus index entries are keys.

Typical Formal Concept Analysis queries seek all index entries which either (a) exactly match a given key or (b) are a super set of the given key. As an example of (b) consider Formal Concept Analysis on animal species: one concept might contain the attributes {has-tail, has-fur}, to find the objects which match this concept we will want all known objects which have *at least* these attributes but may include other attributes as well. Both of these common queries can be vastly aided with spatial indexing. Note that even exact match queries present problems for conventional B-Tree indexes due to attribute ordering in index creation [16].

A Generalized Index Search Tree [13,3] abstracts the core operations of a tree index structure into a small well defined collection of functions. A Generalized Index Search Tree is constructed from a collection of pages. A page is usually much larger than individual keys and may contain many keys. Page sizes are