Chapter 8: Summary and future work

In this dissertation efficiency problems associated with lattice-based approaches have been ameliorated by two strategies: developing faster algorithms and using less concepts in a lattice. The AddAtom algorithm was defined and shown to be a fast lattice construction algorithm whereas the compressed pseudo-lattice data structure that was introduced, support lattices with fewer concepts. The two approaches to more efficient deployment of lattices are complementary.

The AddAtom algorithm efficiently constructs lattices using a tightly focussed search for generator concepts. This search is performed through the use of intent- and extent representative operations. The algorithmic performance of AddAtom is very good both from a theoretical- and an experimental point of view. A worst-case performance bound is $O(||L||.||O||^2.\max(||O'||))$. In experimental comparisons on artificial contexts AddAtom was the best performer in all contexts except those with very high densities or very low densities in which cases it was the second best performer. It was also the best performing incremental algorithm. This indicates that the theoretical complexity bound as stated is not very sharp. In natural contexts the performance advantage of AddAtom was even more pronounced. Initial results suggests that AddAtom has the added advantage of having a relatively tight performance range over contexts of different densities whereas the performance other algorithms that offer good performance differ more significantly over different density contexts.

The compressed pseudo-lattice data structure that was defined is closely related to the line diagram of a lattice and its use as a computational tool in applications such as machine learning, information retrieval and knowledge discovery in databases is discussed. The data structure, essentially a bipartite graph that incorporates an embedded sublattice, combines desirable features of concept lattices in a structure that allows for a flexible mechanism of scaling the size of the embedded sublattice. The scaling is done using defined operations that compress and expand it by removing or adding atoms and coatoms. A compressed pseudo-lattice essentially represents a complete sublattice from which a number of atoms and/or coatoms have been removed. Additionally the relation of the sublattice to the context from which it was derived is preserved. An application-dependent compression strategy or criterion is required to guide this process. It was argued that the removal of concepts from a concept lattice may hold advantages over traditional approaches. Compressed pseudo-lattice shows promise in many field of research due to its close resemblance to that of a formal concept lattice.

The intent- and extent representative operations of a lattice were defined as substitutes for the infimum and supremum operations in compresses pseudo-lattices since the removal of concepts leads to trivial infima and suprema. In both of these areas the notion of the intent- and extent representative operations were shown to be defining in nature. AddAtom uses it to search for generator concepts and, in essence, it repeatedly insert concepts into the lattice in order that $AIR = EIR$.

8.1 POSITIONING AND RELATED RESEARCH

The theoretical experimental comparison in chapter 5 included many of the well-known lattice construction algorithms. The theoretical complexity of the Nourine and Raynaud

(2002) algorithm has the best theoretical complexity of O((||O|| + ||A||).||O||.||L||). Although the cubic nature of the AddAtom theoretical complexity is higher than the quadratic nature of that of Nourine, it is argued that the AddAtom theoretical performance bound is not very sharp – this is confirmed by the experimental results.

The work of Kuznetsov and Obiedkov (2002) indicates that there is no single best algorithm for constructing lattices. This is supported by the findings of chapter 5. A hybrid approach that uses a number of criteria, such a context density, to select an algorithm that would be best to construct the concept lattice is proposed.

Because of the very large data structures associated with lattices the interpretation and use of this data may be obscured by the large amount of detail (often caused by noise in the data). Authors such as Duquenne et al. (2001) have expressed the difficulty in working with large concept lattices and have called for useful approximations of lattices. The approach taken with compressed pseudo-lattices is however not the only approach. A number of alternative approaches for dealing with large lattices have also been proposed:

- Wille (2002) proposes conceptual views that are built using human assistance. Each view represents a small part of the lattice. Since these views are defined by a subset of attributes from the context, they can easily be structured as lattices themselves. This approach supports the idea of browsing a larger lattice where a user can select a conceptual view which is "zoomed" into.
- Hereth and Stumme (2001) generate Iceberg Concept Lattices in which they have purposefully removed nodes to reduce the lattice size. Iceberg Lattices are a specialisation of compressed pseudo-lattices in the sense that a particular compressions strategy is used.
- Pernelle et al. (2002) uses a partial order called nesting. A nested concept lattice is obtained by reducing (through projections) the original lattice. As a consequence it makes the equivalence relation defined on the extents and intents of concepts coarser.
- Godin and Missaoui (1994), proposed ways of reducing concepts in a lattice called a pruned concept lattice. In general, a compressed pseudo-lattice is not directly comparable to a pruned concept lattice.
- Mephu Nguifo (2001) use flexible concept lattices that also do not use the whole concept lattice
- Alternative ideas of reducing concepts are also discussed in (Oosthuizen 1994b).

In general, the first three of these approaches can be supported by compressed pseudo-lattices since they rely on sublattices.

Other approaches focus on reducing or filtering the input context, either in terms of attributes, objects or both such as commonly used in knowledge discovery in databases and information retrieval (e.g. controlled document indexing in Salton (1989), explanation based learning in Oosthuizen (1994b) and Oosthuizen and Avenant (1992)) may also be used to avoid contexts that contain irrelevant attributes and/or erroneous objects which may lead to less effective concept lattice based approaches.

In most instances, FCA-based approaches to problem solving, such as those mentioned in chapter 1, have competing non-lattice based techniques which do not suffer to the same extent from the complexity and size issues as FCA approaches. The future success of FCA-based approaches will thus depend on either having superior predictive or classification performance that outweighs possible time performance issues. Alternatively, approaches resulting in reduced lattice sizes may result in superior time performance.

8.2 FURTHER WORK

Many of the concepts put forward in this dissertation, especially those cantered on the use of compressed pseudo-lattices require further investigation. Below are listed some possible areas for further study related to both AddAtom and compressed pseudo-lattices.

Further work related to AddAtom:

- Further experimental comparison of the AddAtom performance with that of other algorithms not included in this study should be conducted.
- AddAtom can easily be extended to operate on sub-lattices such as those in compressed pseudo-lattices. As was stated in chapter 7, a version of AddAtom with this capability was implementation, but the pseudo-code of the implementation has not been fully documented.
- The experimental comparison in chapter 5 suggests that the performance gap of AddAtom in relation to other algorithms may be the most significant in natural data sets. A wider study is required to support and generalise this observation. Specifically, the extent to which the algorithmic performance of most algorithms running natural data may differ from their performance in terms of artificial data set should be investigated.
- Hybrid approaches combining construction algorithms should be explored whereby various criteria are used to predict a construction algorithm that is most likely to be the best performer. This may even involve a per-object based decision, relying on various incremental lattice construction algorithms to insert objects. Optimisations such as the use of AddCoatom may also be considered.
- As indicated in chapter 7, the developed code is not as efficient as it might be and introduces too many overheads. As a result, there is the need to re-implement and fine-tune the code.

Further work related to compressed pseudo-lattices:

Since the compressed pseudo-lattice is a generic data structure that in essence still uses a lattice (albeit a sublattice), it lends itself to most approaches that rely on FCA. There are, however, a number of areas of research and key research questions that seems most promising. These are listed below.

- In what areas of application are compressed pseudo-lattices beneficial? Specifically, how do compressed pseudo-lattices (and the intent- and extent representative operations) perform in comparison with a formal concept lattice (with the meet and join operations) in areas where the latter has proven successful?
- What compression strategies and criteria should be used and in which areas of application? Specifically, is there a universal compression strategy applicable to many areas of application or are useful compression strategies domain specific?
- What is the relationship of a compressed pseudo-lattice and associated operations to other fields of research in databases, rough sets, etc., given its apparent ability to deal with ambiguity?
- How do various supervised machine learning algorithms perform using compressed pseudo-lattices based on various compression strategies? Here the approach and classifiers proposed in Xie et al. (2002) may be a useful start.

- What is the relationship between the performance gained when fewer concepts are processed, the predictive accuracy of algorithms and the size of the compressed pseudo-lattice?
- To what extent and in what ways may compressed pseudo-lattices be used to support information retrieval?
- A further exploration of the theoretical aspects associated with sublattices would seem to be required.
- A more complete comparison is required of compressed pseudo-lattices with other methods that use sublattices and lattices with a reduced number of concepts.
- There is a need to investigate how compressed pseudo-lattices may be combined with other techniques and approaches.

The research into there and other related issues are on-going.

--------------------oOo--------------------