

Higher Order Mining

John F. Roddick¹, Myra Spiliopoulou², Daniel Lister¹ and Aaron Ceglar¹

¹ School of Computer Science, Engineering and Mathematics
Flinders University
GPO Box 2100, Adelaide 5001, South Australia
roddick | daniel.lister | ceglar @csem.flinders.edu.au

² Institute of Technical and Business Information Systems
Faculty of Computer Science
Otto-von-Guericke-Universität Magdeburg
PO Box 4120, D-39016 Magdeburg, Germany
myra@iti.cs.uni-magdeburg.de

ABSTRACT

The value of knowledge obtainable by analysing large quantities of data is widely acknowledged. However, so-called *primary* or *raw* data may not always be available for knowledge discovery for several reasons. First, cooperating institutions that are interested in sharing knowledge may not be willing (or allowed) to disclose their primary data. Second, data in the form of streams are only temporarily available for processing. If stored at all, stream data are maintained in the form of synopses or derived, abstract representations of the original data. Finally, even for non-stream data, there are limits on the computation speed to be achieved – such limits are set by hardware and firmware technologies. This problem can only be partially solved through parallelization and increased processing power. Ultimately, in many cases data must be summarized to be processed efficiently.

In the light of these observations, we anticipate the need for defining and practising data mining without the luxury of primary data. To that end, we formally introduce the paradigm of *Higher Order Mining* as a form of data mining that is applied over non-primary, derived data or patterns. Although Higher Order Mining is a new paradigm, there are already research advances on knowledge discovery methods from patterns rather than data. We discuss them and organize them under the light of the new paradigm. We show that the HOM paradigm reveals further potential for knowledge discovery, including the delivery of rules and patterns with semantics that are closer to human intuition and are thus more appropriate for human inspection.

Keywords

Higher Order Mining, Mining Patterns, Mining Data Synopses, Mining Derived Data, Rule Semantics.

1. INTRODUCTION

Traditional data mining involves the analysis of data stored in files and data warehouses. The *Higher Order Mining* (HOM) paradigm extends this to the analysis of patterns rather than just from primary data. We are motivated by

the fact that in many applications primary data are either not available at all or for only a limited time period. HOM generalizes the idea of data summarization, which is broadly exploited in real-time stream mining.

In this paper, we discuss the potential of HOM for knowledge discovery, enumerate the challenges associated with HOM and elaborate in more detail on the issues of interestingness and performance. HOM encompasses data mining methods that operate upon derived, i.e. non-primary, forms of information. These may be:

- summaries of the original data, such as histograms derived from a stream of values or summarizations of a multidimensional dataset (as proposed, for example, in CACTUS [27]);
- the result of aggregate queries, including projections and selections over one or more data tables;
- the result of previous data mining runs or mining queries over an inductive DBMS; and/or
- models and patterns derived from an arbitrary dataset.

HOM subsumes the notion of meta-learning, as suggested by, among others, Prodromidis et al. [59], where a classifier is trained upon the output of classifiers rather than upon the primary data.

1.1 Mining Without Primary Data

The ultimate goal of data mining is to deliver, in a timely manner, results that support scientific insight and contribute to well-informed decision taking. The data used for data mining are typically assumed to be *primary data* captured by some application, cleaned and prepared according to the demands of the mining algorithm, and occasionally enriched with further, imported data, which may be alien to the original application. For example, consider the extraction of preference profiles from customer transaction data. Such primary data may be enriched with demographic data imported from a statistics agency. These imported data will be most likely aggregated, for example at the level of city block. To ascribe them to the primary data, some mapping function is needed, which can be used to infer, for instance,

the profession of a customer, given the distribution of professions in her city block. Data mining is then performed upon the primary data, not the aggregated data.

In some applications that demand data mining, no primary data are available. This situation emerges whenever institutions recognize that data sharing will lead to better mining results and thus to better insights, but are not willing or not allowed to share their primary data. Such institutions may be companies that participate in product or market alliances, governmental agencies that derive profiles to deal with national security threats, research institutions and health departments that aim to gain insights on diseases and treatments and to maximise health outcomes while minimising costs. One concrete example, fraud detection, is discussed by Prodromidis et al. [59] who point out that if multiple banks share information, their composite predictor is likely to be better than any one built from local data only. Since privacy and confidentiality considerations prohibit the disclosure of the data owned by each institution, the challenge is tackled by training a classifier upon the output of local, black-box classifiers. This is an early example of Higher Order Mining, i.e. data mining over classifiers rather than raw data.

In other applications, primary data are only available for a short time. A characteristic example is stream data, which are not stored at all – they are encountered, processed in real time and deleted [26]. Guha et al. [32] describe the *stream paradigm* for mining demonstrating the gradual replacement of primary data with aggregates, in their case *summarized* data. In their *k*-means stream clustering algorithm, data are buffered for clustering, and primary data are gradually and continuously replaced by the centroids of the clusters they belong to, so that buffer space is freed for new incoming data. As soon as primary data are replaced by a cluster centroid, data mining is performed on data + patterns, an example of Higher Order Mining.

Even if the data are not presented as a stream, they cannot be stored indefinitely. The removal of primary data may be due to legal obligations, which determine how long a data record may be retained, self-obligations (such as a sensitivity to *Freedom of Information* requests) or internal regulations about storage management. While data summarization techniques are usually applied upon streams, the summarization of transactions as part of a long-term data maintenance process is less usual. In all cases, long-term maintenance and an *a posteriori* analysis of the aggregated information is rarely foreseen – it would require algorithms that perform data mining upon aggregated rather than primary data.

1.2 Outline

This paper is structured as follows. In the next section, we introduce the concept of *Higher Order Mining* in a more formal way and provide a taxonomy of related literature. A discussion of advances in HOM from different perspectives, including spatiotemporal mining, stream mining, pattern change detection and meta-learning is presented in Sections 3 and 4 with the first providing a discussion on HOM from a static, non-temporal perspective and the second from a temporal viewpoint. In Section 5, we explore how the HOM concept can contribute to some outstanding problems of conventional data mining and discuss the potential for HOM in creating rules with higher, in some cases more use-

ful, semantics. We also elaborate on the interplay of HOM and of *privacy-preserving data mining*, where knowledge discovery without access to primary data is also necessary. Section 6 concludes with an agenda of open issues.

2. DEFINING HIGHER ORDER MINING

We define Higher Order Mining as knowledge discovery upon previously induced patterns and models. To some degree, any kind of derived information, such as the average value of an attribute or a histogram over an attribute's values, can be perceived as a simplistic pattern. We opt for a tighter definition as follows:

Higher Order Mining (HOM) is the sub-field of knowledge discovery concerned with mining over patterns/models derived from one or more large and/or complex datasets.

More formally, let $\Xi = \{\xi_i | i = 1 \dots n, n \geq 1\}$ be a set of models or patterns, such that ξ_i has been extracted from a dataset D_i . The datasets D_1, \dots, D_n may be distinct or identical, may correspond to a vertical or horizontal partitioning of a distributed dataset or may be snapshots of a data stream. *Higher Order Mining* encompasses the discovery of any new pattern or model $\hat{\xi}$ from the set Ξ through the use of data mining methods.

Using this definition, we can describe some of the current research as follows:

- For the meta-learning approach of Prodromidis et al. [59], ξ_1, \dots, ξ_n are classifiers upon independent datasets. A higher-order classifier $\hat{\xi}$ is trained over the outputs of these classifiers for an agreed, common training set.
- The method of Lent et al. [44] takes as input one pattern ξ_1 , namely a set of association rules, and builds upon it a clustering $\hat{\xi}$.
- Mei and Zhai [50] build a HMM $\hat{\xi}$ upon a graph, the nodes of which are clusters of clusterings built at different timepoints over a text stream, whereby an edge between two clusters $X \in \xi_i$, drawn at timepoint t_i , and $Y \in \xi_j$ of timepoint $t_j > t_i$, indicates that the two clusters are similar.

While most data mining research has assumed primary data, there have been a number of research advances that adhere to the HOM paradigm. We present two taxonomies and identify research advances for the individual taxa. An algorithm-oriented taxonomy is presented in Table 1. The vertical axis encompasses the mining techniques that are applied on the primary data, delivering different kinds of patterns/models. In the horizontal axis we list the mining algorithm applied upon the patterns or models. For each cell, we provide some example research contributions that indicate the wide spectrum of HOM perspectives.

In Table 1, we do not distinguish between conventional data and stream data, nor among different data types of data, such as text, spatial data or clickstreams. In Table 2, we draw a fundamental distinction between temporal and non-temporal data. The temporal category encompasses both algorithms designed explicitly for the demands of streams and algorithms that process accumulating (and possibly

Primary data mining method	Higher Order Mining method				
	Association mining	Clustering	Classification	Trend analysis	Rule structure
Association mining	[54]	[33, 44]	[6, 48]	[7, 15, 35]	[53]
Clustering	[25, 52]	[11]	[23]	[1, 36, 50, 64]	
Classification			[59]	[22]	

Table 1: Algorithmic Categorization of HOM Approaches

ageing) data in predefined snapshots. The algorithms of the non-temporal category compare models/patterns upon datasets drawn in an independent process. Quite naturally, this last category of algorithm concentrates on pairwise comparisons for dissimilarity detection.

In the category *Spatiotemporal data* in Table 2, we have assigned algorithms that make explicit use of the metric space, such as by studying *moving* clusters [36] or by referring explicitly to geographic objects, such as cities [55].

In Section 3, we elaborate on research related to Higher Order Mining for arbitrary static data. Then, in Section 4 we discuss advances in temporal Higher Order Mining. Studies in spatiotemporal data, as depicted in the middle column of Table 2 are also discussed in this section.

3. HIGHER ORDER MINING IN NON-TEMPORAL CONTEXTS

In this section we discuss the discovery of higher order static patterns while Section 4 concentrates on methods that monitor and identify changes in evolving patterns.

3.1 Clustering for Association Rule Discovery

A number of problems lend themselves to using association mining techniques over non-primary data. Many of these relate to the non-discreteness of either the data or their semantics. We focus here on the use of clustering as a preprocessing step to association rule discovery and its objective of formulating appropriate data intervals or object groups, upon which association rules of adequate frequency can be built.

3.1.1 The Challenge of Data Granularity

A significant challenge for association mining is to generate rules from data that contains quantitative or time-series attributes [61]. An example of a quantitative association rule is *... if a customer buys two, three, or four toothbrushes, then he/she also buys between three and six tubes of toothpaste* [72], which can be represented by the rule:

$$\text{Toothbrush} : [2, 4] \rightarrow \text{Toothpaste} : [3, 6]$$

Unless attribute values are aggregated there may be a significantly increase in the number of rules, which can in turn both increase the difficulty of interpretation and reduce the chance of individual rules reaching the required support threshold [52].

Attempts at aggregating values during the mining phase have suffered from a range of difficulties. The predefined static allocation of ranges can be beneficial for data sets where the user has previous knowledge of what will prove to be meaningful. However, in some cases the distribution of data can favour the border between two predefined ranges and static groups can effectively hide an interesting distri-

bution. For example, given age ranges of:

$$(0 \dots 1], (1 \dots 5], (5 \dots 18], (18 \dots 30], (30 \dots 65], > 65$$

a disease such as meningococcal (which is most prevalent in children under 5 and young adults 15-24) may go undetected in the latter range.

There are complexity issues in discovering quantitative rules, with conventional algorithms often sacrificing precision to make the problem tractable [72]. As stated by Fu et al. [25], the idea of semantic distance cannot be accurately described by the imposition of static ranges. There are three possible solutions to this problem:

1. Aggregate the data as a preprocessing task,
2. Combine the clustering of values with the production of association rules, or
3. Cluster the association rules in the result set based on their semantic distance.

Aggregating values into groups effectively *... transforms the quantitative problem into a boolean one* [25], from which association rules can be elicited.

Since clustering algorithms are intended to find dense distributions of data, they are capable of discovering partitions that can provide an improved fit to the distribution of data and thus reduce the chance of missing important associations [25]. The interpretation of intervals is a difficult issue. Miller and Yang [52] contend that partial completeness measures may allow for neighbouring partitions to merge based on ordering and support, without indicating any kind of semantic distance between the values contained in those partitions. The level of measurable semantic distance required determines the nature of the clustering solution possible.

Both Miller and Yang [52] and Fu et al. [25] are able to use the resulting cluster information to generate a set of intervals which can then be used in association rule mining.

3.1.2 Clustering upon Ordinal Values

Fu et al. [25] solve discretisation issues pertaining to values that are more ordinal in nature. They use CLARANS to discover a set of medoids that determine the mid-range values from which partitions are extracted. As partitions extend from a mid-point they are expected to overlap. This use of fuzzy sets allows for items to be members of two partitions based on partial membership.

Compactness and separation can be calculated not only between clusters, but can also be combined to form a single metric to compare different clustering runs over a data set. Gyenesei [34] allows the user to specify a range of possible values that are iteratively used as parameters to a *c*-means algorithm, after which the results are compared to assess the number of clusters that will provide the highest quality clusters. A consequence of this is the inherent time complexity

Data

Non-temporal		Spatiotemporal	Temporal	
Arbitrary	Spatial		Streams	Batches
[9, 28]		[1, 36, 55, 73]	[50]	[22, 29, 64]

Table 2: HOM algorithms for different types of data

in running the clustering algorithm repeatedly with different parameters. Due to the relative complexity of mining multi-dimensional quantitative attributes through previous algorithms which do not use clustering, it is difficult to measure whether the trade off in time complexity would prove more effective than if a different preprocessing scheme were used [72]. The input of the possible range of clusters belies the necessary user requirement to determine some measure of distance, or in a more general sense, what possible values could provide the most meaningful numerical ranges.

Other work includes the SemGrAM algorithms [60] that utilise semantic graphs during association mining to form disjunctive itemgroups (consisting of semantically similar items) within conjunctive itemsets, such as:

[*Tuesday, Wednesday*], *Cappuccino* \rightarrow *Croissant*

3.1.3 Establishing Intervals by Clustering

In contrast to discovering numerical ranges for ordinal values, Miller and Yang [52] ensure that the semantics of continuous interval data are preserved. Using BIRCH, clusters are capable of being identified over a single pass, which is significantly more efficient than previous clustering algorithms [77]. This advantage is significant when the clusters are being generated as a preprocessing task.

The distance-based association rules created by Miller and Yang [52] can be directly elicited from the clusters generated. Instead of using a set of medoids to determine ranges, the clustering features produced by BIRCH provide sufficient information to produce association rules that are the combination of multiple dimensions. Moreover, instead of the more simplistic measures of support and confidence to determine the degree of association, Miller and Yang [52] suggest a measure based on the distance between two clusters as a means of determining how strong an implication is – the larger a distance between two clusters C_x, C_y , the weaker the implication of $C_x \rightarrow C_y$.

In studies over both ordinal and distance-based data it is clear that user influence over parameters is required to preserve semantic representation – for ordinal data it is the determination of how many ranges will be informative, whereas for intervals it is the maximum allowable relative distance.

3.2 Clustering over Patterns

The formulation of groups via clustering is a powerful complement to traditional model learning or pattern discovery. We elaborate here on clustering of patterns and on clustering inside patterns.

3.2.1 Clustering of Association Rules

In some cases a set of association rules can be difficult to manage or understand [66]. A set of rules of the form

$age[16] \rightarrow gov_benefits[yes]$
 $age[17] \rightarrow gov_benefits[yes]$
 $age[18] \rightarrow gov_benefits[yes]$

might be better described as

$age[16 - 18] \rightarrow gov_benefits[yes]$

Lent et al. [44] formalise the issue of clustering association rules using a technique that is similar to the adjacent partition (or bin) merging algorithm of Agrawal and Srikant [3], however, the Lent et al. algorithm is also capable of merging consecutive values or bins into larger entities. As the algorithm can handle either intervals or values it is possible to input either into the binning algorithm. Each bin in the matrix represents an association rule in which the consequent is bound to an individual cell.

Gupta et al. [33] extend this work by looking at distance based clustering of association rules while Perrizo and Denton [57] outline a framework based on partitions to unify various forms of data mining algorithm.

This method can be contrasted against clustering as a preprocessing task, as the clustered representations are required to be frequent, therefore an iterative process is undertaken to ensure that the discovered clusters are sufficiently frequent enough to allow for subsequent association rules to be derived from them. One of the most useful domains for clustering association rules has been for explorative mining [67], in which a user requires on-demand expressive rules that are easily readable – in many cases it would be infeasible to do this as a preprocessing task.

3.2.2 Clustering Clusters

Partition-based clustering algorithms aim to produce clusters that have high intracluster similarity and low intercluster similarity. They are iterative in that their centres are based on their average item membership distance, which will continue to shift until it reaches convergence, according to some criterion. A set of initial points are required at which to allow the mean to be initially calculated, and allow the iterative process to begin.

As illustrated by the work of Bradley and Fayyad [11], if the starting point is sub-optimal, clustering algorithms which use iterative refinement will in many cases reach sub-optimal solutions. They contend that an improved local minima can be discovered through performing clustering over subsamples and then through clustering the combination of solutions arrive at an improved starting point. This form of refinement allows for even a reasonably large subsample (10% of overall data set) to be run in a fraction of the time taken to cluster the entire set of data. If the clustering process has a sub-optimal starting point, it will take longer to reach convergence. The smoothing process reduces the outlier subsample solutions and gives a greater chance that the subsequent clustering algorithm will reach a good solution [11].

3.3 Using Association Rules for Classification

Associative classification uses association rules to build a classifier [48]. The association rules must have the consequent bound to the class label, hence are considered class

association rules (CARs). Once a set of CARs are gathered, they are ranked, with the highest ranked CARs used to create classifiers. Associative classification has two primary benefits:

1. Many preprocessing solutions used within the association rule mining domain have made significant headway into dealing with complex data issues, such as the discretisation of interval or spatial data [13]. Classifiers built with association rules are not required to deal with these complex data issues, and face lesser challenges in handling complex relationships as some of these problems have been handled at the first-order level.
2. Freitas [24] points out that association rule mining generates the same set of rules for every subsequent run over the same data, whereas other classifiers can be unstable, generating radically different classifiers for small changes to the input dataset. Put another way, association rule mining finds *all possible rules that explain the presence of the attribute* whereas most decision tree induction methods provide only *one explanation or rule (the one chosen by the heuristic methods)* [40].

It can be argued that CARs are more understandable to the domain expert who is required to verify the reasoning behind a classifier's choices, which has been a significant shortcoming of previous rule elicitation techniques when building classifiers [56].

Associative classification can be separated into two areas, the first involves the generation and ranking of association rules, with the other performing rule selection and classifier construction.

3.3.1 Rule Generation for Associative Classification

Conventional association rules use a variety of methods to determine which rules are *interesting* and statistically significant within the chosen data set [31].

Clearly, rules must be pruned and ranked during the CAR generation process to reduce the volume of rules generated, and to allow for an informed selection procedure to build the classifiers from the most highly ranked rules. One procedure to evaluate CARs is to rank them in the order of confidence, support, and precedence, respectively [48], also termed the *most-confident-first* (MCF) principle [70].

3.3.2 Building Classifiers From Association Rules

Given a prioritised set of CARs, a classifier is created based on the set of rules that cover the training set [48]. The classifier is iteratively built where, if a rule covers some set of examples in the training set, it is added and those pertinent examples are removed from further rule evaluations. Each rule, therefore, only covers a particular set of examples, and hence only rules that increase the accuracy of the classifier are included. The default class is determined as the rule which covers the largest number of examples.

Due to the CAR set being sorted, it is often the first rule to cover the set of examples that will be the most accurate. It should be noted that the best judge of a rule is not simply the set of examples it successfully classifies, but also the proportion that are classified incorrectly. Pessimistic pruning may remove some rules with a high error rate during the

CAR generation phase, however a lower ranked rule may, upon comparison with other higher precedence rules, offer an improvement to accuracy in the form of a reduced error rate. This error checking is important to avoid issues relating to over and under fitting.

It is difficult to determine which will be the most appropriate rule to determine a class based on precedence and error rate. In contrast to a single rule determining the label, it is possible to use a group of high precedence rules to build a classifier [47]. This has the benefit of reducing the error rate due to the possible subset of examples identified incorrectly by a single high precedence rule being covered correctly by a lower precedence rule if it were in the same group, on the proviso that the correct rule from the group is selected to cover each set of examples. When a new object is classified, the label is determined by the most representative rule.

There is an inherent limitation in using only positive association rules to build classifiers – items not positively correlated with a particular class label may prove to be the deciding factor in respect of which class label an object should be considered [6].

ARC-PAN (Association Rule Classification with Positive And Negative) uses both positive association rules and negative association rules in the same manner, allowing multiple ranked rules to be used much in the same form as the algorithm presented by Li et al. [47] with the strength of a group of rules being determined by average confidence.

There are two complementary ideas presented in an effort to reduce the error rate through multiple rules:

1. Additional information provided through negative rules reduces the inherent complexity in selecting the best group of rules [6].
2. A new measurement with which to assess the best group of rules can allow for both minority and majority classes to be correctly labeled, as previous strength-based measurements favor the majority case [47].

Using multiple rules for the purposes of prediction provides more accurate classifiers. Yin and Han [74] identify the use of multiple rules as crucial due to the lack of full accuracy possible in determining the effectiveness of a rule. Furthermore, as with the strength-based measurements, a single rule will cover the majority class only.

4. HIGHER ORDER MINING FROM A TEMPORAL PERSPECTIVE

The temporal perspective of higher order mining is manifested in frameworks that detect changes upon data that come from a non-stationary distribution, in pattern management suites and, obviously, in temporal mining methods. Many approaches in this area assume that the data change but their schema does not. However, there are also advances on dealing with pattern evolution when the schema changes as well.

4.1 Detecting Changes in Patterns

Ganti et al. [27, 28, 29] proposed an early framework for the detection of changes in patterns to identify and quantify the differences between patterns drawn from different snapshots of the dataset. These snapshots may refer to data selected at different timepoints or at different locations

(spatial data). The framework encompasses three modules, FOCUS, DEMON and CACTUS.

FOCUS [28] compares two models and computes an interpretable qualifiable deviation measure between them. For the comparison, Ganti et al assume that a model consists of a *structure component* and a *measure component*. For example, the structure component of a decision tree is a representation of its nested nodes, while the measure component captures the distribution of classes in each node. To compare two patterns, the patterns are decomposed and *interesting regions* are identified. These are regions where the two patterns disagree in the class distribution. The subset of data belonging to each region are then summarized by the measure component.

FOCUS can be coupled with DEMON [29], the module which monitors data change across the time axis. DEMON distinguishes between systematic and non-systematic changes in the data and identifies the data blocks that must be processed to build new patterns. Finally, CACTUS summarizes clusters into synopses of their statistical properties [27]. Thus, instead of comparing data subsets, pattern comparison can be performed on their summarized representations. Another framework for pattern comparison and the detection of differences has been proposed by Bartolini et al. [9] for the PANDA framework. Similarly to FOCUS, complex objects (patterns) are decomposed into simpler ones, for which similarity/distance functions are available. The decomposition of complex objects, the computation of dissimilarity scores for their components and the aggregation of these scores into scores for complex objects are governed by some *aggregation logic*. Aggregation logics can be specified by the PANDA user, albeit some built-in logics are also available within the framework.

4.2 Trends in Association Rules

The mining of association rules over changing attribute values is a complex challenge due to the dimensionality and semantics of the data involved. Wang et al. [71] define an evolution as *the temporal changes of attribute values of some object*.

To discover temporal associations the temporal dimension must first be divided into subsequences (snapshots in this model) much like the algorithm presented by Das et al. [20]. Due to the large number of dimensions that an object may have, it is sometimes necessary to mine a subset of the possible dimensions. This idea of subspace clustering is covered by Agrawal et al. [4], who describe the discovery of clusters over dimensions that are derived from only a subset of the possible dimensions based on the identification of which dimensions primarily contribute to noise, and those dimensions in which there may be particular interest.

Wang et al. [71] discretise the quantitative attributes into small intervals. It is then possible to cluster these intervals to discover the ranges for each subsequence. To determine if the occurrence of an object is frequent enough, we must use a density metric to ensure that sufficient objects follow this evolution, which can be expressed as a ratio of the number of objects that contain values within the interval. Support and strength measures are used to ensure that a significant number of objects follow the evolution across the subsequences,

so that a rule such as:

$$\begin{aligned} & (\textit{scholarship} \in [1000, 1500]) \\ & \xrightarrow{\textit{meets}} (\textit{scholarship} \in [2000, 3500]) \\ & \xrightarrow{\textit{meets}} (\textit{scholarship} \in [4000, 5500]) \end{aligned}$$

represents the evolution of a significant proportion of objects within the data set. It is evident that through the combination of many first order concepts we can achieve these higher order relationships. Wang et al. [71] introduce a measure of information loss by initially discretising intervals with equal distances to reduce the inherent complexity of combining multiple algorithms.

Harms et al. [35] offer a similar solution. Upon the discovery of frequent episodes (as opposed to shapes), sequential rules can be built of the form:

$$r = \alpha[\textit{win}_a] \rightarrow_{\textit{lag}} \beta[\textit{win}_b] \quad (1)$$

with both α, β being frequent within their respective window *win*, and β occurring for *lag* after α . While support for these rules follows the calculation as for normal association rules, confidence is measured by *the conditional probability that β occurs, given that α occurs, under the time constraints specified by the rule*. Window width is specified by the user, as the duration as to what will be an interesting time metric is variable based on the environment.

Au and Chan [7] propose a different implementation. Association rules are still elicited from sequence data, however a decision tree is built over the collected association rules to allow for prediction based on the present set of rules discovered. The confidence measure used is more complex than Harms et al. [35], with confidence being determined as:

$$\gamma_j(X \rightarrow Y) = \frac{|d_j \in D_j | X \cup Y \subseteq d_j|}{|D_j|}$$

with D_j being the set of transactions for time period t_j . One of the important issues with trend prediction raised by Au and Chan [7] is the fact that rules that do not reach the necessary support and confidence measures will not be returned by an association rule miner. Fuzzy meta-rules are elicited from these association rules by measuring the change of support and confidence between contiguous sequences and quantifying them through simple linguistic terms such as *fairly decrease*. The fuzzy meta-rules are built using a fuzzy decision tree which effectively establishes branches based on information gain. New occurrences can then be classified based on traversing the tree in a manner that offers the path of best fit (meets the conditions of that leaf). The concept of fuzziness is required due to the lack of a crisp sense of membership; the linguistic terms used to quantify rules allow for this degradation in precision. To discover the meta-rules that make up a decision tree the tree is traversed from the root to the respective leaves.

A pendant to trend prediction with classification algorithms is the use of clustering upon a temporal sequence to identify segments that show the same trend. Das et al. [20] partition a temporal sequence into contiguous subsequences of equal length. Once this subsequence size is specified, subsequences can be clustered into basic *shapes* such as *increase(1)*, *decrease(1)*. Once this collection of shapes has been derived, it is possible to discover associations that are

linked to these basic shapes. The adaptation of the association rule algorithm uses the following notion of confidence:

$$\gamma_{(A\bar{T}B)} = \frac{F(A, B, T)}{F(A)} \quad (2)$$

where F is the frequency of A following B within T . Harms et al. [35] follow a similar methodology in separating a sequence into subsequences based on a sliding window, over which episodes may be derived.

Das et al. [20] elaborate on the importance of selecting proper clustering parameters to ensure that the shapes identified do not contain too many or two few subshapes, as they are required to be applied repeatedly to detect time-based associations. One of the shortcomings of this approach is that tuning clustering algorithms based on previous results is based on the assumption that previous information was correct [39]. A parameterless algorithm may prove more accurate due to the possible range of errors that may arise through a user's *prejudices, expectations, and presumptions*.

4.3 Trend Detection in Clusters

Similarly to association rules, clusters may also exhibit trends of different types. A cluster may shrink or expand, it may split into further clusters or merge with another. Cluster transition detection on the time axis is performed by the MONIC framework [64]. Next to a *cluster transition model*, which lists that types of change that may occur on a cluster, MONIC also encompasses an algorithm that detects *external transitions* – transitions involving multiple clusters, and a set of heuristics that detect and quantify *internal transitions* inside a cluster, such a change in size or homogeneity.

Aggarwal adopts a different perspective upon cluster change [1]. A cluster is a densification of data in a multidimensional space, where a *dimension* is a feature. Intuitive examples for this type of cluster definition can be obviously found in Geographical Information Systems. A town is an area where houses are more dense than in the surrounding areas. The evolution of the town (e.g. growth or shrinkage) can be modeled as a density change. Aggarwal approximates clusters with kernel functions and then computes kernel density changes at each spatial location. Density changes at neighbouring spatial locations are aggregated, so that one can detect areas which change at different *speed* than their surroundings. For each location at timepoint t , the *forward density change* captures kernel density changes in the future of t , while *backward density change* captures density changes in the past of t . The difference between forward and backward density change constitute the *velocity of change* [1].

Kalnis et al propose a special type of cluster change, the *moving cluster*, whose contents may change while its density function remains the same during its lifetime [36]. They find moving clusters by tracing common data records between clusters of consecutive timepoints.

The work of Aggarwal [1] and Kalnis et al. [36] refer to clusters as spatial objects. For many applications, this cluster representation is not intuitive. Particularly for text clustering, one is inclined to derive cluster labels as summaries for the cluster contents. Aggarwal and Yu [2] introduce the notion of a *droplet* as a summary of a cluster over a text stream. A droplet consists of two vectors, one containing all words appearing in the underlying cluster and one containing all pairs of co-occurring words. The words in the first vector are weighted – older records of the stream have lower

weights.

Aggarwal and Yu [2] use the notion of droplet to monitor cluster evolution. A new document is assigned to the cluster to which the droplet is most similar. If a cluster does not receive documents for some time, it becomes *inactive*. If a document cannot be assigned to a new cluster, it becomes itself a cluster, thereby replacing the oldest inactive cluster. This approach assumes that the number of droplets (topics in a stream) are fixed and that old ones decay and are replaced by newer ones.

Cluster trends over a text stream are also addressed by Mei and Zhai [50]. Again, the focus is on detecting topics and studying their evolution. Mei and Zhai perform probabilistic clustering, in which each document contributes to each cluster with a probability that increases with its similarity to the cluster's mean. A *topic* is the mean of a cluster and consists of the words characterizing the cluster's members. For trend monitoring, Mei and Zhai connect clusters that appear in the time axis and have similar topics. This results to a *topic evolution graph*, where a sequence of clusters constitutes a (persisting) *theme*.

4.4 Pattern Management

Pattern management is an orthogonal task to data mining and to higher order mining. However, for the higher order mining paradigm, we should keep in mind that the analysis of patterns presupposes pattern maintenance.

Some systems store previously mined rules for the purpose of performance improvement through incremental mining [42, 65]. It is computationally expensive to continuously rescan the entire database for rules when at a small cost in terms of accuracy, adequate results can be obtained through sampling [43], or by reanalysing only when the difference between the rules stored and the actual information held by the data differs significantly [42].

Pattern maintenance frameworks pay much attention to the management of association rules. Spiliopoulou and Roddick [63] provide a framework for modelling higher order association rules as temporal sequences of conventional rules obtained from different mining sessions. Mining sessions are defined as a 6-tuple, providing a *signature* to which higher order reasoning algorithms can refer. Higher order mining routines are then able to operate over temporal sequences of rulesets. The followup system PAM [8] studies these sequences to detect changes in the statistics of the rules in the ruleset under observation.

Association rules' management is also offered in PANDA [9] and in PSYCHO [49], where some emphasis is also paid in the maintenance of decision trees and of clusters. A specialized system for Web pattern monitoring is proposed in [17]. The system NaviMoz is designed for navigation patterns and pays much emphasis on how the habitual behaviour of users can be reconstructed through pattern querying and inspection. A generalization effort for arbitrary patterns has been proposed by Meo and Psaila [51]. The pattern management model XDM captures data and patterns and builds upon XML as basis for a unifying framework.

5. HIGHER ORDER MINING FOR KNOWLEDGE DISCOVERY

Data mining techniques have been successfully applied in a number of diverse application domains including health

and medicine, biotechnology, natural sciences, telecommunications, commerce, financing and marketing, defence and national security. The diversity of applications and current trends in data management have exacerbated or even given raise to several still unresolved issues. As pointed out in Section 1, some modern applications require data analysis without primary data or without the possibility of maintaining and scanning the primary data more than once. In this section, we discuss how Higher Order Mining can contribute to solutions for conventional but complex applications of knowledge discovery and we elaborate on the associated challenges that are yet to be solved.

5.1 Discovering Patterns for Human Consumption

While advances in data mining encompass very powerful algorithms, there are fewer advances on driving the knowledge discovery process towards results appropriate for human consumption. For example, association rules' discovery methods transform a very large dataset into a very large ruleset that is human understandable but still brings a considerable cognitive load for the human expert. The approaches proposed to this challenge can be categorized into:

- *Condensed representations*: Algorithms of this category produce a reduced set of results, from which valid patterns can be derived. This category is particularly important for association rules' discovery; the techniques used for association mining are discussed in [14].
- *Pattern visualization*: These methods exploit the natural human pattern recognition capability. All commercial KDD products and many prototypes incorporate next to the data mining algorithms also methods for the visualisation of the results.
- *Interestingness Measures*: These measures are intended to improve the fit between the discovered results and the perception of a pattern's "usefulness". Geng and Hamilton [31] provide a survey of methods for association rules.

The aforementioned methods refer to the organisation of the results. Some methods or measures are incorporated to the mining algorithm to ensure the generation of a human-consumable set of results, while others are rather preprocessing filters upon already derived results. However, the study of patterns reveals that experts are often interested in specific higher order patterns and need means for describing them in advance. For example, an analyst that designs a product portfolio may be interested in *competitor* products or products that churn with each other. An analyst that studies customer profiles is likely to care about shrinking profiles that require special attention and growing profiles that indicate a new exploitable market segment. The first example refers to patterns that have some special property, while the latter refers to patterns whose evolution exhibits a pattern – i.e. to Higher Order Mining patterns. Example of such patterns are given in Appendix A.

A promising approach towards specifying the meta-patterns to be exhibited by the resulting patterns comes from the paradigm of database mining. Advances in inductive databases and in (multi-)relational database mining include query languages and engines with which the knowledge discovery process can be driven by the human expert. The query language

of PANDA allows the juxtaposition of patterns with data [9], the query language of PSYCHO supports the detection of trends in patterns [49], the ConQueSt system supports the incorporation of constraints to database mining [12], while the system of Bonchi et al. [10] integrates mining algorithms into the query optimiser. Although much of the research on Higher Order Mining is performed on data that are not stored in a database and without exploiting the potential of database querying facilities, the recent research on pattern comparison and pattern trend discovery in a database context indicates that powerful query languages are already being made available to human experts for Higher Order Mining. Another open area is the development of *pattern languages* that can describe data mining patterns. Such pattern languages have been used successfully in other areas such as in describing software patterns [19].

5.2 Higher Order Mining upon Non-Crisp Datasets

Conventional set theory generally only considers a *crisp* set methodology, whereas in many real world domains that level of precision is not available [41]. Cases such as first-order association rules not meeting thresholds for second-order mining [7], or partitions of interval data not having clear boundaries [52], imply the need for some form of support for fuzziness that cannot be adequately handled based on the rigid constraints of set theory.

Since the accuracy and completeness of information cannot be assumed, due in part to the human element involved in all data handling and the propensity to assume that data stored in a database is accurate, it is evident that higher order mining may be particularly susceptible to the constraints imposed by classical (crisp) sets.

One of the primary methods of dealing with fuzzy sets in a data mining context is to use more generalised terms to describe attribute values. In terms of human readability and comprehension the use of general linguistic terms is beneficial in many contexts where precision is not paramount. Fuzzy sets are also beneficial for the purposes of spatial or interval data, where hard partitions of quantitative data may imply a significant difference.

5.3 Privacy

Privacy is a non-trivial issue if data mining is to gain general acceptance and a variety of approaches have been suggested. While the issue is too large to discuss in detail here (qv. [5, 18, 21, 30, 38, 69, and many others]), it is worth noting that rulesets have a level of anonymity built in as a natural consequence of the mining process.

The demand for privacy-preservation in data mining emerges in two different, though often related contexts. First, personal data must be protected from disclosure *towards everyone*. Second, confidential data must be protected from disclosure *towards partners*. The first case encompasses all kinds of sensitive data, the analysis of which can lead to useful insights, such as understanding the progress of illnesses, the effects and adversary effects of drugs, the preferences for products and the anticipated lifetime value of customers. While there are agents that legitimately express those needs, this does not imply a legitimization for disclosure of the primary data themselves. The second case refers to *data sharing* among institutions, whose internal data are confidential (and occasionally subject to the constraints of

the first case), but who recognize that integrated data mining is of benefit to all participating institutions.

There are a number of situations in which rulesets can be made publicly available but the source datasets cannot. For example, in the case of horizontal partitioned data, Kantarcioglu and Clifton [37] point out that rules mined from a particular source may put the provider at a disadvantage if they are unique to that provider and are sensitive in nature. Using their method of encryption and rule passing the algorithm collects the set of rules and passes them on to the next repository in a secure fashion. Similarly, Vaidya and Clifton [68] create an association rule mining algorithm which can operate over vertically partitioned data in a privacy preserving way. Rozenberg and Gudes [62] offer an improved algorithm which reduces the chance of disclosure through the use of fake transactions, together with a cost-benefit algorithm which assesses the risk of information disclosure during the confidence measuring process.

Two groups of privacy-preserving approaches have emerged in the literature to deal with these challenges. The first group encompasses anonymization methods, which (a) distort the primary data and (b) aggregate them into derived data, from which private information cannot be reconstructed any more. These anonymized, derived data are published - occasionally in the form of publicly accessible files, but also in the form of queryable databases. The published data can then be processed with conventional mining methods. Advances of this category deal with the first case of privacy-preservation.

The second group encompasses interaction and processing protocols that allow the partners to build a shared data mining model without sharing the primary data. This corresponds to the second case above. In a nutshell, privacy-preserving algorithms of this category specify how the data used to build the model should be encrypted and shared among the partners. The output model is based on the original data rather than a distorted version. However, similarly to the first group, the data being encrypted are not primary but rather aggregates, mainly in the form of a sum of a set of values.

Our *Higher Order Mining* paradigm is orthogonal and complementary to the privacy-preservation advances of the second group, but contributes to the same problem. As Prodromidis et al. [59] have shown, a common model can be established by sharing and mining patterns rather than data. For some applications, this level of sharing may be appropriate, more so if the patterns themselves can be treated as black boxes [59] - the confidential data are not shared but the model is built. For other applications, a privacy-preservation protocol may be needed. As Kantarcioglu and Clifton [37] point out, the association rules contributed by one partner may disclose sensitive information if they are unique to that partner. The same authors have provided a privacy-preserving solution to this problem [37]. Hence, Higher Order Mining for *shared* patterns can be enhanced with a privacy-preserving protocol if the application demands so.

5.4 Distributed Higher Order Mining

Li et al. [45] put forward a method to mine data across multiple databases, each with its own schema. Rules may be generated across these heterogeneous schemas on the proviso that there is a unique key for each dataset that may join

these complex sets of data together. Previous work in the field has focused on horizontally or vertically distributed databases, where the global schema has been known [37, 62, 68]. Horizontal partitions involve multiple sites containing a set of the tuples which make up the global set. Vertical partitions contain relations which can be joined to provide information or to create a global schema, or that contain *some elements of a transaction* [68]. In either case each of these relations or tuple sets can be owned by a separate entity, or may have been distributed for the purposes of performance improvements.

Previous work assumes some level of global schema knowledge, however in some cases this may not be possible. Zhang et al. [76] consider the schema differences between a set of distributed databases as one of the largest challenges facing traditional data mining methods. Heterogeneous records are records stored at different sites which each have *data for different set of features possibly with some common features among sites* [46]. The level of record heterogeneity and privacy required for a particular database will determine the nature of the implementation. In an environment where privacy is not a significant issue, they use a similarity-based distributed data mining framework (SBDDM) to integrate the tables into a set of virtual tables based on similarity. Distributed data mining techniques are then applied to these virtual tables. One of the significant limitations of this virtual table method is that it may result in a significant communications overhead for databases with large number of transactions or that the merging process may simply prove infeasible for systems with extremely heterogeneous data sets [75].

Whether it is possible to use higher order semantics to find relationships between highly heterogeneous data sets [45, 58] is an open question.

5.5 Algorithm Tractability

In many cases mining routines are beginning to encounter problems as the volume of data requiring analysis grows disproportionately with the comparatively slower improvements in I/O channel speeds. That is, many data mining routines are becoming heavily I/O bound and this is limiting many of the benefits of the technology. Methods to reduce the amount of data have been presented in the literature including statistical methods, such as sampling or stratification, dimension reduction by, for instance, ignoring selected attributes, and incremental maintenance methods, that analyse the changes to data only [16].

The I/O channel speed problem has meant that in many practical situations, even an algorithm of linear scalability up can show inadequate performance. A large retail, web search or telecommunications company can easily produce multi-GB transaction files each day. Annually, transaction files can exceed many TB. A single scan through such data can take two or more hours, even on high end servers. By comparison, even with support levels set low, the combined size of 365-odd itemsets (i.e. one for each trading day of the year) can be in the MB range.

6. DISCUSSION AND CONCLUSIONS

While higher order mining has already been used in a variety of ways, the identification of HOM as a distinct subfield would be useful as a mechanism for comparing approaches

and identifying open areas for research. Our taxonomy in Table 1 shows that there are areas where little or no overt research has been done to date which might prompt further investigation.

Higher order mining requires a change in perspective for knowledge discovery, from the analysis of data to the analysis of patterns. As we have discussed, this brings new potential, among else in driving the mining process towards human-consumable results and in fostering distributed data analysis among institutions that are not disposed or allowed to share their raw data. However, this potential also requires further research work in expressing the semantics of higher order patterns according to human perception and in line with privacy-preservation requirements.

Higher order mining can contribute to the challenge of applying high-complexity algorithms on enormous data volumes. Data could be partitioned across the time axis or across some other dimensions of the feature space, allowing for conventional mining within each partition and for higher order mining over the patterns extracted from the partitions. This would not only reveal similarities and interesting differences among the partitions but also contribute to a more tractable process. For time-based partitions, stream mining methods and frameworks for change detection can be used, while methods for pattern comparison and spatiotemporal methods are appropriate for other types of partitioning.

The development of HOM techniques also reasks the question of whether plug-compatibility for data mining algorithms might be possible. There is a wealth of conventional data mining algorithms to serve as a basis for higher order mining. Any pattern specification model, be it intended for a mining language or for pattern presentation and exchange, must be compatible with existing advances and make full use of them. The same is desirable for the higher order mining algorithms themselves. Many of the algorithms that operate upon patterns are tightly coupled to their specifications and particularities, calling for a more seamless integration between conventional algorithms and HOM algorithms.

In summary, as discussed in Section 1.1, higher order mining is the sole approach for knowledge discovery in the absence of primary data. In this study, we have shown that there are many advances that adhere to this paradigm but that the overall potential of HOM is still largely unexploited and worthy of further research.

References

- [1] C. C. Aggarwal. On change diagnosis in evolving data streams. *IEEE Transactions on Knowledge and Data Engineering*, 17(5):587–600, 2005.
- [2] C. C. Aggarwal and P. S. Yu. A Framework for Clustering Massive Text and Categorical Data Streams. In *Proceedings of the SIAM conference on Data Mining 2006*, April 2006.
- [3] R. Agrawal and R. Srikant. Mining sequential patterns. In P. Yu and A. Chen, editors, *11th International Conference on Data Engineering (ICDE'95)*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.
- [4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD International Conference on the Management of Data, SIGMOD'98*, pages 94–105, Seattle, WA, USA, 1998. ACM.
- [5] S. Agrawal and J. Haritsa. A framework for high-accuracy privacy-preserving mining. In *21st International Conference on Data Engineering (ICDE 2005)*, pages 193–204. IEEE, 2005.
- [6] M. L. Antonie and O. R. Zaïane. An associative classifier based on positive and negative rules. *9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 64–69, 2004.
- [7] W.-H. Au and K. C. Chan. Mining changes in association rules: a fuzzy approach. *Fuzzy Sets and Systems*, 149(1):87–104, 2005.
- [8] S. Baron, M. Spiliopoulou, and O. Günther. Efficient monitoring of patterns in data mining environments. In *Proc. of 7th East-European Conf. on Advances in Databases and Inf. Sys. (ADBIS'03)*, LNCS, pages 253–265. Springer, Sept. 2003.
- [9] I. Bartolini, P. Ciaccia, I. Ntoutsis, M. Patella, and Y. Theodoridis. A unified and flexible framework for comparing simple and complex patterns. In *8th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2004*, volume 3202 of *LNAI*, Pisa, Italy, 2004. Springer.
- [10] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, R. Trasarti, and I. Isti-Cnr. Conquest: a constraint-based querying system for exploratory pattern discovery. In *22nd International Conference on Data Engineering, 2006. ICDE'06.*, pages 159–159, 2006.
- [11] P. S. Bradley and U. M. Fayyad. Refining initial points for k -means clustering. In J. Shavlik, editor, *15th International Conference on Machine Learning (ICML '98)*, pages 91–99. Morgan Kaufmann, San Francisco, CA, 1998.
- [12] T. Calders, B. Goethals, and A. Prado. Integrating pattern mining in relational databases. In *International Conference Principles of Data Mining and Knowledge Discovery, PKDD'06*, pages 454–461. Springer, 2006.
- [13] M. Ceci, A. Appice, and D. Malerba. Spatial associative classification at different levels of granularity: A probabilistic approach. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 3202 of *LNCS*, pages 99–111. Springer, 2004.
- [14] A. Ceglar and J. F. Roddick. Association mining. *ACM Computing Surveys*, 38(2), 2006.
- [15] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In A. Gupta, O. Shmueli, and J. Widom, editors, *24th International Conference on Very Large Data Bases, VLDB'98*, pages 606–617, New York, NY, USA, 1998. Morgan Kaufmann.

- [16] D. Cheung, J. Han, V. Ng, and C. Wong. Maintenance of discovered association rules in large databases: an incremental updating technique. In S. Su, editor, *12th International Conference on Data Engineering (ICDE'96)*, pages 106–114, New Orleans, Louisiana, USA, 1996. IEEE Computer Society.
- [17] E. Christodoulou, T. Dalamagas, and T. Sellis. Navimoz: Mining navigational patterns in portal catalogs. In *2nd International Workshop on Pattern Representation and Management (PaRMA'06)*, Munich, Germany, 2006.
- [18] C. Clifton and V. Estivill-Castro, editors. *Privacy, Security and Data Mining - Proc. IEEE International Conference on Data Mining Workshop on Privacy, Security, and Data Mining*, volume 14 of *CRPIT*. ACS, Maebashi City, Japan, 2002.
- [19] J. O. Coplien and D. C. Schmidt. *Pattern languages of program design*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995.
- [20] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*. AAAI Press, 1998.
- [21] V. Estivill-Castro, L. Brankovic, and D. Dowe. Privacy in data mining. *Privacy - Law and Policy Reporter*, 6(3):33–35, 1999.
- [22] W. Fan. Systematic data selection to mine concept-drifting data streams. In *10th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2004)*, pages 128–137, Seattle, WA, USA, 2004. ACM.
- [23] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.
- [24] A. A. Freitas. Understanding the crucial differences between classification and discovery of association rules: a position paper. *ACM SIGKDD Explorations Newsletter*, 2(1):65–69, 2000.
- [25] A. Fu, M. H. Wong, S. C. Sze, W. C. Wong, W. L. Wong, and W. K. Yu. Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. In *1st International Symposium on Intelligent Data Engineering and Learning (IDEAL98)*, pages 263–268, Shatin, Hong Kong, China, 1998.
- [26] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: a review. *SIGMOD Record*, 34(2):18–26, 2005.
- [27] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS - clustering categorical data using summaries. In S. Chaudhuri and D. Madigan, editors, *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 73–83, San Diego, CA, 1999. ACM Press.
- [28] V. Ganti, J. Gehrke, and R. Ramakrishnan. A framework for measuring changes in data characteristics. In *18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 126–137, Philadelphia, PA, 1999. ACM.
- [29] V. Ganti, J. Gehrke, and R. Ramakrishnan. DEMON: Mining and monitoring evolving data. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):50–63, 2001.
- [30] J. Gehrke, editor. *Special Issue on Privacy and Security*, volume 4 of *SigKDD Explorations*. ACM, 2002.
- [31] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3), 2006.
- [32] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. OCallaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, 2003.
- [33] G. K. Gupta, A. Strehl, and J. Ghosh. Distance based clustering of association rules. In *Intelligent Engineering Systems Through Artificial Neural Networks, ANIE 1999*, pages 759–764, St. Louis, Missouri, USA, 1999. ASME.
- [34] A. Gyenesei. Fuzzy partitioning of quantitative attribute domains by a cluster goodness index. Technical Report TUCS-TR-368, Turku Centre for Computer Science, 2000.
- [35] S. K. Harms, J. S. Deogun, and T. Tadesse. Discovering sequential association rules with constraints and time lags in multiple sequences. In *13th International Symposium on Foundations of Intelligent Systems, ISMIS '02*, volume 2366 of *LNCS*, pages 373–376, Lyon, France, 2002. Springer.
- [36] P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *9th International Symposium on Spatial and Temporal Databases (SSTD2005)*, volume 3633 of *LNCS*, page 364381, Angra dos Reis, 2005. Springer.
- [37] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1026–1037, 2004.
- [38] M. Kantarcioglu, J. Jiashun, and C. Clifton. When do data mining results violate privacy? In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, pages 599–604, Seattle, WA, USA, 2004. ACM Press, New York, NY, USA.
- [39] E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206–215, Seattle, WA, USA, 2004. ACM Press.

- [40] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. Verkamo. Finding interesting rules from large sets of discovered association rules. In N. Adam, B. Bhargava, and Y. Yesha, editors, *3rd International Conference on Information and Knowledge Management*, pages 401–407, Gaithersburg, Maryland, 1994. ACM Press.
- [41] G. J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.
- [42] S. Lee and D. Cheung. Maintenance of discovered association rules: When to update? In R. Ng, editor, *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'97)*, Tucson, AZ, USA, 1997. ACM.
- [43] S. D. Lee, D. W. Cheung, and B. Kao. Is sampling useful in data mining? a case in the maintenance of discovered association rules. *Data Mining and Knowledge Discovery*, 2(3):233–262, 1998.
- [44] B. Lent, A. Swami, and J. Widom. Clustering association rules. In A. Gray and P.-A. Larson, editors, *13th International Conference on Data Engineering*, pages 220–231, Birmingham, UK, 1997. IEEE Computer Society Press.
- [45] S. Li, T. Wu, and W. M. Pottenger. Distributed higher order association rule mining using information extracted from textual data. *SIGKDD Explorations*, 7(1):26 – 35, 2005.
- [46] T. Li, S. Zhu, and M. Ogihara. A new distributed data mining model based on similarity. *2003 ACM Symposium on Applied Computing (SAC2003)*, pages 432–436, 2003.
- [47] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In N. Cercone, T. Lin, and X. Wu, editors, *IEEE International Conference on Data Mining, ICDM'01*, pages 369–376, San Jose, CA, USA, 2001. IEEE Computer Society.
- [48] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *4th International Conference on Knowledge Discovery and Data Mining, KDD-98*, pages 80–86, New York City, 1998. AAAI Press.
- [49] A. Maddalena and B. Catania. Towards an interoperable solution for pattern management. In *3rd Int. Workshop on Database Interoperability INTERDB'07 (in conjunction with VLDB'07)*, Vienna, Austria, Sept. 2007.
- [50] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text - an exploration of temporal text mining. In R. Grossman, R. Bayardo, and K. Bennett, editors, *11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD05)*, page 198207, Chicago, IL, 2005. ACM Press.
- [51] R. Meo and G. Psaila. An xml-based database for knowledge discovery. In *2nd International Workshop on Pattern Representation and Management (PaRMA'06)*, Munich, Germany, 2006.
- [52] R. Miller and Y. Yang. Association rules over interval data. In J. Peckham, editor, *ACM SIGMOD International Conference on the Management of Data*, pages 452–461, Tucson, AZ, USA, 1997. ACM Press.
- [53] D. Mladenic, W. F. Eddy, and S. Ziolko. Exploratory analysis of retail sales of billions of items. *Computer Science and Statistics*, 33, 2001.
- [54] C. H. Mooney and J. F. Roddick. Mining itemsets - an approach to longitudinal and incremental association rule mining. In A. Zanasi, C. Brebbia, N. Ebecken, and P. Melli, editors, *Data Mining III - 3rd International Conference on Data Mining Methods and Databases*, pages 93–102, Bologna, Italy, 2002. WIT Press.
- [55] D. Neill, A. Moore, M. Sabhnani, and K. Daniel. Detection of emerging space-time clusters. In R. Grossman, R. Bayardo, and K. Bennett, editors, *11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD05)*, pages 218–227, Chicago, IL, 2005.
- [56] M. Pazzani, S. Mani, and W. R. Shankle. Beyond concise and colorful: learning intelligible rules. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Third International Conference on Knowledge Discovery and Data Mining*, page 235238, Newport Beach, CA, USA, 1997. AAAI Press.
- [57] W. Perrizo and A. Denton. Framework unifying association rule mining, clustering and classification. In *International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA03)*, Rio de Janeiro, Brazil, 2003.
- [58] W. M. Pottenger, S. Li, and C. D. Janneck. Distributed higher-order text mining: theory and practice. *2006 International Conference on Digital Government Research*, pages 446–447, 2006.
- [59] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*. AAAI press, 2000.
- [60] J. F. Roddick and P. Fule. SemGrAM - integrating semantic graphs into association rule mining. In *6th Australasian Data Mining Conference (AusDM 2007)*, volume 70 of *CRPIT*, Gold Coast, Australia, 2007. ACS.
- [61] J. F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.
- [62] B. Rozenberg and E. Gudes. Association rules mining in vertically partitioned databases. *Data and Knowledge Engineering*, 59(2):378–396, 2006.

- [63] M. Spiliopoulou and J. F. Roddick. Higher order mining: Modelling and mining the results of knowledge discovery. In N. Ebecken and C. Brebbia, editors, *Data Mining II - 2nd International Conference on Data Mining Methods and Databases*, pages 309–320, Cambridge, UK, 2000. WIT Press.
- [64] M. Spiliopoulou, I. Ntoutsis, Y. Theodoridis, and R. Schult. MONIC: modeling and monitoring cluster transitions. In *12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD06)*, pages 706–711, Philadelphia, USA, 2006. ACM.
- [65] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An efficient algorithm for the incremental updation of association rules. In *3rd International Conference on Knowledge Discovery and Data Mining (KDD 97)*, pages 263–266, New Port Beach, CA, USA, 1997. ACM Press.
- [66] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hattönen, and H. Mannila. Pruning and grouping of discovered association rules. In *ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases*, pages 47–52, Heraklion, Greece, 1995.
- [67] A. Tuzhilin and A. Gediminas. Handling very large numbers of association rules in the analysis of microarray data. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 396–404, Edmonton, Alberta, Canada, 2002. ACM.
- [68] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 639–644, Edmonton, Alberta, Canada, 2002. ACM Press.
- [69] K. Wahlstrom, J. F. Roddick, R. Sarre, V. Estivill-Castro, and D. de Vries. Legal and technical issues of privacy preservation in data mining. In J. Wang, editor, *Encyclopedia of Data Warehousing and Mining*. IGI Publishing, 2nd edition, 2008.
- [70] K. Wang, S. Zhou, and Y. He. Growing decision trees on support-less association rules. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 265–269, Boston, MA, USA, 2000. ACM Press.
- [71] W. Wang, J. Yang, and R. R. Muntz. Tar: Temporal association rules on evolving numerical attributes. In *17th International Conference on Data Engineering, ICDE 2001*, pages 283–292, Heidelberg, Germany, 2001. IEEE Computer Society.
- [72] J. Wijsen and R. Meersman. On the complexity of mining quantitative association rules. *Data Mining and Knowledge Discovery*, 2(3):263–281, 1998.
- [73] H. Yang, S. Parthasarathy, and S. Mehta. A generalized framework for mining spatio-temporal patterns in scientific data. In R. Grossman, R. Bayardo, and K. Bennett, editors, *11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD05)*, pages 716–721, Chicago, IL, 2005. ACM.
- [74] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *SIAM Conference on Data Mining, SDM'03*, pages 369–376, San Francisco, California, USA, 2003. SIAM.
- [75] C. Zhang, M. Liu, W. Nie, and S. Zhang. Identifying global exceptional patterns in multi-database mining. *IEEE Computational Intelligence Bulletin*, 3(1):19–24, 2004.
- [76] S. Zhang, X. Wu, and C. Zhang. Multi-database mining. *IEEE Computational Intelligence Bulletin*, 2(1):5–13, 2003.
- [77] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient clustering method for very large data bases. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 103–114, Montreal, Canada, 1996. ACM.

Appendix A - Examples of Higher Order Mining Patterns

While tightly specifying higher order patterns precisely can be difficult, the advantage of such higher-order descriptors is that they can be closer to the semantics we use when describing objects or object behaviour. Indicative (and loosely described) examples of the sort of higher order pattern that might be found for association rules include:

- *Competitors*: The identification of pairs of items, which while individually frequent, rarely occur together with the frequency that might be expected from independent variables.
- *Clusters*: Sets of items that tend to occur in a transaction together but rarely occur individually.
- *Catalyst*: An item that occurs either independently or with other items but where the other items rarely occur independently.
- *Staple*: An item that occurs with a large subset of the other items in the dataset.
- *Triples*: Three items that occur individually and in triples but rarely only in pairs.
- *Loners*: Items that often occur alone in a transaction.
- *Groupies*: Items that only tend to occur in transactions with large numbers of other items.
- *Gangs*: Items that frequently occur multiple times in a transaction (requires quantitative association rules).

Similarly, based on the cluster transitions we have discussed in Section 4.3, we can also derive descriptions for clusters that *regularly* exhibit a particular behaviour. Such descriptions may include:

- *Poachers*: Two or more clusters that, over time, assume ownership of a set of objects or a region.
- *Oscillators*: A cluster that, over time, breaks into multiple clusters, which then merge again.
- *Pulsars*: Clusters that regularly grow and contract.
- *Black Holes*: Clusters that become more dense over time.

The specification of such patterns is an open area for research.