On Exploiting the Power of Time in Data Mining

Mirko Böttcher
University of Magdeburg
Magdeburg, Germany
mail@mirkoboettcher.de

Frank Höppner
University of Applied Sciences,
Braunschweig/Wolfsbüttel
Wolfsbüttel, Germany
f.hoeppner@fh-wolfenbuettel.de

Myra Spiliopoulou
University of Magdeburg
Magdeburg, Germany
myra@iti.cs.uni-magdeburg.de

ABSTRACT

We introduce the new paradigm of *Change Mining* as data mining over a volatile, evolving world with the objective of understanding change. While there is much work on incremental mining and stream mining, both focussing on the adaptation of patterns to a changing data distribution, Change Mining concentrates on understanding the changes themselves. This includes detecting when change occurs in the population under observation, describing the change, predicting change and pro-acting towards it. We identify the main tasks of Change Mining and discuss to what extent they are already present in related research areas. We elaborate on research results that can contribute to these tasks, giving a brief overview of the current state of the art and identifying open areas and challenges for the new research area.

1. DATA MINING IN TRANSITION

Data mining has traditionally concentrated on the analysis of a static world, in which data instances are collected, stored and analyzed to derive models and take decisions according to them. More recent research on stream mining has put forward the need to deal with data that cannot be collected and stored statically but must be analyzed on the fly. At the same time, the need to store, maintain, query and update models derived from the data has been recognized and advocated [30]. However, these are only two aspects of the dynamic world that must be analyzed with data mining: The world is changing and so do the accummulating data and, ultimately, the models derived from them. The challenge does not only lay in adapting the models to the changing world but also to analyze *how* the models change and *when* they do so. In this paper, we propose a new paradigm for data mining over the evolving world: *Change Mining* encompasses methods that capture the process of change, analyze how models have changed and predict changes that will emerge.

The need for a paradigm shift is not only dictated by new applications that deal e.g. with streams of data [15] but is also incubent in traditional applications of business (e.g. Customer-Relationship-Management and fraud detection), medicine (e.g. healthcare and epidemiology), natural sciences (e.g. astronomy and meteorology) and many more.

Taking an example from the business world, companies regularly carry out surveys to analyze the relation between past experiences of customers and their attitude towards the business. Association rule discovery is commonly used in this scenario, usually accompanied by clustering or classification methods for the establishment of customer segments, upon which decision makers design the segment-specific products, marketing actions or recommendations. Nowadays, this course of action is a quite straightforward business intelligence task. Yet it suffers from several drawbacks: (i) Since the datasets used to this purpose are very large and grow at fast pace, decision makers are often forced to refrain from powerful but poorly scaling data mining methods or to perform data sampling, thus risking the construction of a non-representative data subset. (ii) Some methods such as association rule discovery almost always return a vast number of rules, many of which are obvious and already known. It requires a strenuous manual inspection to detect those few rules which are of interest because they were not known before. (iii) The segments built by clustering or derived by classification are affected by external factors, such as demographics, but also by the production and the marketing actions of the company itself and of its competitors. So, before exploiting models that may have become obsolete, decision makers need to learn how their earlier actions may have affected/changed them.

All these shortcomings can be alleviated by taking a time-oriented perspective and placing the understanding of *changes* in the centre of knowledge discovery: A mechanism that remembers old association rules, compares them to new ones and highlights those that have emerged or have changed significantly guarantees the distinction between already known and new - interesting or interestingly changed ones. The study of the changes occurring upon customer segments in regular intervals or after the launching of marketing actions provides insights on how customers' behaviour is affected by the actions and sets the basis for pro-active marketing activites and for the prediction of further segment evolution. The study of change does not require the analysis of *whole* datasets, nor sampling over them. In fact, this would be counterproductive since the underlying population is all but stationary.

In this paper, we discuss *Change Mining* as a paradigm that encompasses mechanisms that monitor models and patterns over time, compare them, detect changes and quantify them on their interestingness. In Section 2 we define the notion of Change Mining and elaborate on the methodology associated with this paradigm. In Section 3 we show that Change Mining is already present though implicit in modern research and we discuss associated research fields. Section 4 describes the tasks of Change Mining and elaborates on research studies that have contributed to each task. In the last Section we present a research agenda for Change Mining as a new way of thinking about data mining.

A terminological note: Here and in the following, we use the term "model" for *both* models and patterns, notwithstanding that a pattern, differently from a model, describes only part of the data. Hence, we term classifiers, clusterers *and* sets of association rules over a dataset as models. When we refer to individual clusters, to single association rules or decision tree nodes that describe only part of the data population we use the terms *model component* or *local model*.

2. DEFINING "CHANGE MINING"

Conventional data mining methods observe *one* dataset (or data stream) and learn (local) models upon it. In the Change Mining paradigm, we generalize into *a temporal sequence of datasets*, for which we want to derive the changes effected upon their models during the elapsed time.

DEFINITION 1 (CHANGE MINING). *Change Mining is a data mining paradigm for the study of time-associated data. Its objective is the discovery, modelling monitoring and interpretation of changes in the models that describe an evolving population.*

Change Mining is thus a subdomain of *Higher Order Mining* [37], which encompasses methods for the discovery of knowledge by processing models (instead of data), such as meta-learning, model adaptation, model comparison, temporal mining, mining models (e.g. clustering of association rules) and Change Mining – the discovery of changes in evolving models.

2.1 Formal Definition

Formally, let $T =< t_0, \ldots, t_n >$ be a sequence of timepoints and let D_i be the dataset accummulated during the interval $(t_{i-1}, t_i]$, where D_i may be a static dataset, whose records do not possess timestamps themselves, or a stream of records. Furthermore, let $f()$ be a decay function which determines which data contribute in the learning process and with which weights. For example, $f()$ can express a sliding window of length w, so that all data in the interval $[t_0, t_i - w]$ are forgotten, or $f()$ may be an exponential function of ageing, which assigns weights to the individual records on the basis of their age.

> At each timepoint $t_i, i > 0$, we observe a model (or a set of local models) Ξ_i, derived upon the dataset $\widehat{D_i} := f\left(\cup_{j=1}^{i} D_j\right)$. We define Change Mining as a new paradigm that encompasses (a) methods which describe the changes of Ξ_i to $\Xi_j, j > i > 0$ and (b) methods which build a predictive model over the sequence $< \Xi_1, \ldots, \Xi_n >$.

Hence, similarly to conventional data mining, Change Mining has a *descriptive* and a *predictive* subcategory of algorithms.

2.2 Change Mining Methodology

The *description* of changes among models involves two core tasks. First, it must be decided whether models are indeed different and their differences must be quantified. Second, their differences must be described semantically and interpreted. While there is much work on identifying differences between two models, there is less work on the process of monitoring differences in a sequence of models and also less work on the semantic interpretation of such differences.

The *prediction* of changes in a sequence of models implies building a higher order model, which will decide whether the next member of the sequence will be different from the members seen thus far. A more elaborate aspect of change prediction involves describing in what aspects the next model will be different from already seen ones. In this context, change prediction involves *the description of model changes*, as mentioned above.

In general, the spectrum of Change Mining approaches to be researched is vast: For each type of model used in conventional data mining there is a manifold of ways to analyse change. We propose four generic tasks that constitute a methodological process for Change Mining:

1. Determining the goals of Change Mining:
   - Deciding between description of changes or prediction of change, whereupon the former is also a prerequisite for the latter
   - Determining whether the interest lays with the *result of change* or with the *process of change* itself
2. Specifying a model of time:
   - Partitioning the time axis into intervals for observation
   - Specifying a decay function upon the data
3. Specifying the objects of change:
   - Determining the type(s) of model to be studied
   - Determining the *granularity* level of change to be studied, deciding between the study of a whole model (such as a classifier) and of its components (such as individual classification rules, clusters or association rules) – these are the *objects of change*
   - Identifying the types of change that can occur upon the selected objects of change
4. Designing a monitoring mechanism:
   - Designing a method for tracing models or their components over time, depending on the specified objects of change
   - Designing an algorithm for the identification of model (or model component) changes
   - Designing an algorithm that captures the changes in the model (components)
   - Extracting interesting changes
   - Semantically interpreting change

The process formed by these tasks partially reflects the traditional process of data mining, paying emphasis on the specification of the problem to be solved, i.e. the goal and the objects of change and the types of change that are of interest, and on the appropriate modeling of the data, i.e. the models and their changes.

Similarly to conventional data mining, Change Mining is followed by the task of identifying suitable actions to be taken in response to the discovered changes. This task depends strongly on the application domain and thus lacks the generality of the other tasks which we will discuss in more detail in Section 4. We nonetheless discuss the nature of this task in the example hereafter.

Example: (Change Mining over a Web Log). *In conventional Web usage mining, we derive models that reflect user behaviour and preferences. Assume an online shop that acquires monthly reports from their web hoster. Further, assume that the web hoster does not simply deliver rudimentary statistics but also association rules on products accessed together, as well as clusters of users with similar purchase preferences.*

Task 1 of Change Mining: *Example objectives for the online shop are (a) the discovery and description of differences among the purchases of adjacent and/or non-adjacent months, (b) the discovery and description of the evolution of the user clusters over the last few months and (c) the prediction of rules that are expected to hold in the next month, given the models seen thus far.*

Web hosters deliver activity reports in regular intervals, e.g. months. Then, in terms of Task 2*, the time axis is partitioned in monthly intervals, i.e.* $(t_{i-1}, t_i]$ *is a month and* t_i *denotes the last reported day of the month. At the end of each month* t_i*, the web hoster delivers a clustering of users* Ξ_i *and a set of association rules* $\mathcal{R}_i$*. Both the reports and the clustering have been derived on dataset* $\widehat{D_i}$*.*

If the web hoster has been instructed to build the models upon all data, i.e. from the moment when hosting started, then $f()$ *is the identity function and no data are forgotten. If the web hoster deletes the web server log every two months, then* f *is a sliding window of* 2*, i.e. two months.*

Task 3: *The objects of change are individual clusters of users with similar preferences and individual association rules on items being purchased together.*

Task 4: *According to the objectives in* Task 1*, the expert responsible for Change Mining must specify in what ways a cluster of preferences or an association rule has changed and which of those changes are of interest. Mechanisms for these tasks are described in Section 4.*

3. RELATED FIELDS

The Change Mining paradigm, as defined in Section 2, focusses on *change* in the models or patterns that describe the evolution of data over time. The ultimate objective is the discovery and the understanding of changes. The study of temporal data is a mature field with a veritable amount of valuable findings, but there are also further research areas with a close relationship to Change Mining. We provide a brief overview of these research areas here and determine the scope of Change Mining within and beyond them. Many methods in these areas refer explicitly to patterns, so we use the term “patterns” instead of “models” here.

3.1 Incremental Mining

Incremental mining methods are designed for the updating of patterns as data are inserted, modified or deleted – prominently in a data warehouse. This implies dealing with a very large initial dataset Δ, which should be updated with a batch of insertions δ_+ or deletions δ_-. The bottleneck of pattern actualization is the size of Δ. Accordingly, the goal of incremental methods is to update the patterns in a fast and reliable way, minimizing the accesses to Δ.

Pattern updating may involve the detection of changes in the original patterns. For example, the incremental density-based clustering algorithm IncrementalDBSCAN [17] identifies changes in the neighbourhood of newly inserted or of deleted data points, such as a cluster split or dissolution or the emerging of a bridge that connects two initially disconnected regions (so called: neighbourhoods of data points) into the same cluster. However, the objective is the efficient updating of the clusters with a minimal number of operations, rather than the capturing and understanding of pattern change. For example, the insertion of a data point may cause a cluster to grow, the insertion of the next data point may cause the cluster to merge with another cluster, while the insertion of a third point may cause growth of a different region than the first one. The only information considered by IncrementalDBSCAN in association to these changes is whether the set of cluster members needs to be updated or not. Whether or not a cluster merger is incidentally caused by some noise data and therefore should not be reported as a change in the underlying population is beyond the scope of such algorithms.

A further characteristic of incremental mining methods that disagrees with the nature of Change Mining is the treatment of old patterns. An incremental miner updates a pattern by overwriting its old state. It is obviously trivial to store the old patterns instead of replacing them. However, as mentioned already, the understanding of the differences between an old and a new pattern is beyond the scope of an incremental miner.

3.2 Comparison of Populations

A substantial body of research has been devoted to the detection of *differences* between two datasets. In most cases, the objective is to decide whether the two datasets belong to the same population or, equivalently, have been generated by the same distribution. Some methods of this category like FOCUS [18] and PANDA [6] provide both a qualitative and quantitative description of the differences between two datasets or between two derived models.

The discovery of differences between two datasets is also studied in the field of so-called *emerging patterns*: Goal of these methods is to discover patterns, in particular itemsets, whose support significantly differs between the two datasets [16; 44].

Very recently, Liu and Tuzhilin stressed the need to store, query and analyze models, discover models that underperform and figure out whether models are missing and should be built [30]. They use the term *modelbase* for the venue where models are maintained and propose a first approach for the identification of underperforming models.

The theoretical underpinnings of such methods are obviously relevant for Change Mining, although the methods themselves lack the temporal aspect, which we consider fundamental for Change Mining. On the other hand, combinations of these methods with a utility that deals with temporal data do belong to the field of Change Mining. For example, FOCUS can be combined with the module DEMON [20] developed by the same research group: DEMON monitors evolving data, while FOCUS measures the differences between patterns captured at different timepoints. As a further example, one of the followup frameworks of PANDA [6], the PSYCHO framework [32], has utilities for the treatment of temporal data and for the detection of changes in patterns - prominently of association rules.

3.3 Novelty Detection

Novelty detection is a special area of Change Mining. The objective of novelty detection methods is to decide whether newly arriving data instances agree (are represented by) an existing model or deviate strongly from it. This issue has been studied among else in [21; 24; 31]. Novelty detection methods start with an existing model, which is assumed to be representative of the population and may have been derived from historical data. They detect *events*, i.e. new data instances, which do not fit the model or even invalidate it. These events can signal something not seen before, hence the term "novelty". In general, the focus is on detecting deviations from the model, i.e. abnormalities.

Change Mining encompasses more than novelty detection or abnormality detection. First, Change Mining also covers advances on the monitoring of an evolving population and the identification of changes in individual models or model components, e.g. in an association rule or a cluster; a complete model is not imperative. Second, Change Mining takes a more long-term perspective over the data and includes methods for the regular comparison of derived models and the discovery of *trends* in their changes (e.g. topic evolution, community dynamics).

4. THE TASKS OF CHANGE MINING

In this section, we investigate the steps of the methodological process of section 2 in greater detail.

4.1 Determining the Goals of Change Mining

As pointed out at the beginning of Section 2.2, the objective of Change Mining may be to capture and semantically interpret changes, i.e. *change description*, or the establishment of a higher order model that foresees the changes to come, i.e. *change prediction*. Change prediction obviously builds upon change description.

There are two possible notions of change. Both have their own relevance and both should be accounted for by Change Mining: *change* may denote *the process of change*, such as the evolution of a customer segment and its responsiveness to a marketing strategy that must be aligned again and again to keep it profitable, and *the outcome of change*, such as a sales collapse for a particular product in a certain customer segment.

These notions of change are associated with two intuitive questions in Change Mining: "How is the world changing?" and "When did/will the change occur?" The answer to the first question refers to the evolution of a model across several timepoints $t_i, \dots, t_j$. It describes the nature of the change, such as the shrinking of a cluster or the increase in the support or confidence of an association rule, it captures the properties that have changed and quantifies the importance of the changes, and, ultimately results in a model of change. This higher order model can be a descriptive one, corresponding to the objective of *change description* or be used for proactive *change prediction*. The second question on the timepoint of change is important for the study of the past ("when did a change occur?") and no less for the prediction of the future, i.e. for foreseeing when change as *outcome* will occur and what it will look like.

The nature and frequency of change as well as the distribution of time points $t_0, \dots, t_n$ has an effect on which of the two objectives should be put forward. In a slowly evolving domain, transitions from model Ξ_i to Ξ_j are expected to be smooth. Next to describing them, *prediction of the next change* and its impact can be attempted. Contrary to it, if data are captured sparsely and sudden changes are observed, then *change description* may be the sole accomplishable task. Inducing forthcoming changes from the changes seen thus far may be less appropriate at first. As with prediction in conventional KDD, the description of the data is expected to provide clues about a predictive model at the long term.

Change description is directly or indirectly practised in all methods that study change in models or patterns: Agrawal et al use queries to capture differences between rules seen at different timepoints [4], while Liu et al study capture the interestingness of rules that change [12; 29], next to their support and confidence [29; 5; 8]. Cluster evolution methods first start by describing the types of cluster change that may be observed (cf. Section 4.4 below), including the distinction between clusters and background in a noisy environment [33; 22].

Change prediction is practised less often [33]. Quite intuitively, it presumes that change occurs smoothly rather than abruptly: Aggarwal measures the velocity of change in clusters [1]; evolving clusters are detected against a stationary background in [22]. A stationary background is also assumed in [33], where a HMM is designed for prediction over a document stream.

4.2 Specifying a Model of Time

Change Mining assumes a partitioning of the time axis $T =< t_0, \dots, t_n >$, such that a model Ξ_i is derived at each t_i. This model can be discovered by a conventional static mining method that is applied on the data seen thus far and possibly weighted with help of a decay function $f()$, e.g. of a sliding window. Alternatively, Ξ_i may be the result of adaptation, as is done by stream mining methods which adopt the previous model Ξ_{i-1} to the dataset D_i arriving in $(t_{i-1}, t_i]$ while forgetting older records, again by means of a decay function. In all cases, the model observed at t_i depends on the partitioning of the time axis.

Several aspects must be considered by the specification of the partitioning $T =< t_0, \dots, t_n >$. On the one hand, a long period leads to large datasets and thus enhances the reliability of the estimated model. However, long periods imply a coarse-grain partitioning, in which interesting short-duration changes might be overseen and the precise location of change on the temporal axis gets blurred. On the other hand, short periods force a more frequent re-learning of the model, implying that the model may become less robust. This makes model evaluation more difficult, because the distinction between interesting models and incidental noise gets more challenging [22].

To our knowledge, the approach of Chakrabarti et al [13] on change discovery upon association rules is the only one which *computes* the timepoints $t_0, \dots, t_n$. Rather than assuming pre-defined time periods, Chakrabarti et al determine the best partitioning *for each itemset*, so that the correlation between its constituting items is maximally homogeneous within each partition and inhomogeneous between partitions. The method is inspired by the Minimum Description Length principle: The authors define a binary coding scheme for the time partitions, so that the code length decreases as homegeneity increases. Then, for each itemset they choose the partitioning that minimizes the sum of the code lengths of the partitions. An elegant aspect of this approach is that the code length also reflects volatility – a potential measure of interestingness: Large code lengths indicate higher volatility of the correlation among the items and thus a potentially more interesting itemset.

The disadvantage of the method in [13] is that it derives one partitioning *per itemset*. Beyond the scalability restrictions thus implied, a juxtaposition of the evolution of different rules (e.g. overlapping ones) is more complicated. This disadvantage holds also for traditional methods applied on time series to distinguish between signal and noise, such as Fourier and wavelet transforms. To make the task even more challenging, Change Mining may require the study of multiple properties of the same rule (e.g. confidence, support and lift) or cluster (e.g. cardinality, intra-cluster distance and centroid position), as well as the study of the evolution of properties of multiple objects, e.g. the pairs of inter-cluster distances within a clustering. Hence, single ideal partitioning of the time axis is desirable but is not to be expected in this context.

More pragmatic approaches are often feasible though. For example, many applications are inherently designed around regular time intervals like days, months or years: Customer surveys in marketing are often conducted regularly, so model evolution can be designed across the same intervals. Another option is to fix the size of the dataset D_i studied at each timepoint t_i into a constant c. This is appropriate for applications like the mining of rapid streams that operate on buffers of given size. Datasets of fixed size also imply that model perturbations caused by variations in the dataset size cannot occur and thus require no corrective preprocessing.

4.3 Specifying the objects of change

The objects studied in change mining are themselves models in the traditional sense, i.e. a $\Xi_i, i = 1 \dots n$ may be a classifier (e.g. a decision tree, a neural network, or a support vector machine), a set of clusters (a "clustering") or a set of association rules.

One crucial decision in Change Mining refers to whether a model is observed as a *monolithic black box* that gives no further insights or as the *composite* of individual submodels. Intuitively, the monolithic approach makes sense when the model cannot be decomposed into interpretable components, as e.g. for a neural network, where a decomposition is possible but a *neuron-by-neuron* interpretation is not. The compositional approach is more appropriate when the components of the model are themselves meaningful and of potential interest when studied separately and in relation to each other, as holds e.g. for individual rules or clusters.

4.3.1 Monolithic Approach

Under the monolithic approach, the object of study for change mining is the model as an atomic, non-decomposable entity. Change Mining involves then monitoring the model's evolution over time and the detection, quantification and interpretation of differences between models encountered at different timepoints.

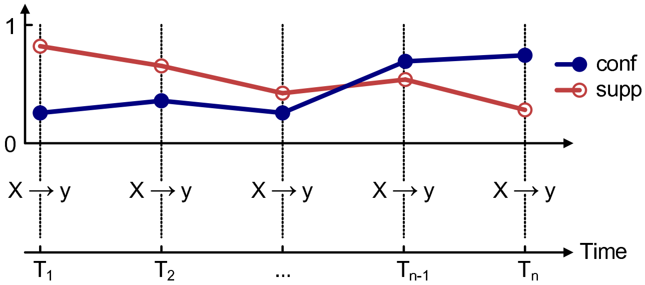

Figure 1: Deriving time series for support and confidence of association rules

Change Mining methods adhering to the monolithic approach devise measures that quantify and assess the model's behaviour or performance at each timepoint. For example, Rissland et al propose the *structural-instability metric* [36] for decision trees: Two decision trees Ξ_{i-1} and Ξ_i, built at t_{i-1}, t_i respectively, are juxtaposed by pairwise comparing their tree nodes and counting (and weighting) (mis)matches. Yang et al propose the *conceptual equivalence metric*, used to decide whether the model presently observed is equivalent to any previously seen model: The equivalence is computed as the count of agreements between the two models when labeling a testset.

In general, the quantification and comparison of the behaviour of two models can be based on measures of the models' predictive power towards a testset, e.g. by monitoring the accuracy, F-measure, precision & recall of a predictive model. Alternatively, Change Mining can concentrate on internal validity measures on homogeneity, separation or stability. Such measures are traditionally used for the evaluation of descriptive models [41]. Since they usually range over a continuous interval of real numbers like $[0, 1]$, the evolution of the model under study can be mapped to the timeseries of each measure's values, as shown in Figure 1 for the case of an association rule. Changes may then be peaks or significant perturbations across the time axis, indicating changes in the analyzed population. Such approaches concentrate mainly on *detecting change* rather than *describing the change* in greater detail.

The monolithic approach lends itself quite intuitively for models that are hard to interpret, such as neural networks or SVMs. Other models, such as decision trees, can be decomposed easily into interpretable components; for example, a decision tree can be decomposed into a set of overlapping decision rules. For such models, it is appealing to consider monitoring the components rather than the whole model. However, the underlying graph theoretic problem to solve, like the inexact graph matching, common subtree and tree editing distance problem, are known to be NP-hard and hard to approximate [7; 43]. Another difficulty is the instability of many learning algorithms: Small changes in input training samples may cause dramatic changes in the produced models. For instance, multi-layer perceptrons and decision trees are known to be unstable [10; 26]. This means that even if the underlying population does not change at all we may observe large changes in the model's structure. Overcoming these difficulties with appropriate methods is still a topic of ongoing research.

4.3.2 Compositional Approach

Under the compositional approach, the objects of study for Change Mining are the *components* that jointly constitute the global model. Such components may be the decision rules or the tree nodes in a decision tree classifier, individual clusters of a clustering or individual rules found by association rules discovery. The objective of Change Mining is to study the evolution of local models over time, to detect, quantify, localize and predict changes among such local models.

Under Change Mining for clustering we find approaches that detect the emergence of new clusters and the decay/dissipation of old ones, the shrinking or expansion of a given cluster in size or with respect to some homogeneity indicator or the movement of a cluster in a topological space (cf. among else [1; 23; 40]), as well as methods that study the relationship among multiple clusters such as cluster split or merge (cf. [17; 33; 40]). In the context of association rules discovery, there are methods that compute the validity interval of individual association rules [35] and methods that focus on (interesting) changes in the statistics of individual rules (cf. [12; 28; 5]). The comparison of classifier components involves comparison of their statistics and of their properties, i.e. (attribute, value)-pairs, as is done in the FOCUS framework of [18]. Kim et al [25] propose a method based on similarity and difference measures for discovering three types of changes from rules derived from a decision tree: the emerging model, the unexpected change and the added/perished rule.

Quite naturally, Change Mining for local models involves the study of the local model's statistical properties, such as the cardinality, homogeneity or distribution (mean, standard deviation, curtosis etc) of a cluster, the accuracy of a decision tree node or a decision rule and the support, confidence or lift of an association rule.

Under the compositional approach we find methods that deal with the object of change (the local model) in fundamentally different ways: Aggarwal proposes a dedicated method that both *builds and monitors* evolving clusters in a spatiotemporal domain [1]. Other methods take as input the models built by an independent, conventional knowledge discovery algorithm [27; 5; 40; 22], sometimes posing certain requirements on the statistics made available for the local models [18]. Frameworks for the comparison of models and their components [6; 32] also delegate the process of model building to an external independent algorithm. Incremental methods [17; 34] and stream miners stand between these ends by adapting a dedicated method with change detection. Many advances on stream mining use change detection to adapt local models to changes in the population [2; 34; 42; 11; 45].

With respect to interpretability, the compositional approach seems advantageous over the monolithic one, since the human expert can understand change in the context of the application domain, e.g. retail marketing [14], exception detection [5] and customer segmentation [9]. However, the compositional approach is more demanding: The model of study must be interpretable itself and decomposable. If this is guaranteed, then different types of change can be be monitored and studied, as proposed e.g. in the *change taxonomies* of [18; 40]. Next to the types of change in statistics of objects, in their interplay with other objects (e.g. cluster split or rule merge) and the evolution in structure of the objects themselves (e.g. cluster movement), it is important to capture also the case where *no changes* have occured at all. In fact, one aspect of change mining is the distinction between evolution/change and stability of the underlying population [36; 28; 22].

4.4 Mechanisms for Monitoring

According to the change mining methodology outlined in subsection 2.2, once the time axis is properly partitioned and the objects of change are defined, mechanisms are due for monitoring the models or their components. They should encompass discovery and interpretation of change and may be descriptive or predictive in nature.

The discovery of change in models is more challenging for the compositional than for the monolithic approach because it involves tracing the components (the *local models* of Section 4.3.2) across time. In particular, if a model Ξ_i discovered at timepoint t_i contains a component X, change mining involves tracing X in the model Ξ_{i+1} built at the next timepoint and deciding that X has disappeared if it cannot be matched to any of the components in Ξ_{i+1}.

For some types of model Ξ_i and for some forms of decomposition, tracing a component $X \in \Xi_i$ may be trivial. For example, association rules have an unambiguous symbolic description, so that tracing e.g. a rule $X = (A \rightarrow B)$ in a later model Ξ_{i+1} is a simple task that can be solved on the syntactic level. Once this task is solved, comparisons on the statistics of the old and the new rule can easily take place.

In other cases, the local models may have a more ambiguous symbolic description that makes tracing a challenge: As pointed out in [40], a cluster X (as component of a clustering Ξ_i) may be observed as a distribution, a formation/area in a feature space or as a set of objects. If it is set that a cluster is a distribution, then it must be decided to what extend a change of the distribution's parameters is tolerated as inherent to the *same cluster*. For example, if cluster X has a mean μ and if we find a cluster $Y \in \Xi_{i+1}$ with mean μ', how close should μ, μ' be to each other to decide that Y is the same as X?

Solutions to the challenge of tracing local model are of two types: First, one may define a *match*-function that returns the closest approximate to a given local model X according to the local model's definition. One may thus specify that two clusters X, Y defined as distributions are the same if their means are within ε_1 of each other *and* their standard deviations with ε_2 of each other. Similarly, if X, Y are defined as sets of objects, the *match*-function may determine that they should share at least $n\%$ objects to be considered the same cluster (as e.g. in [23; 40; 22]). If X, Y are dense areas in a feature space, then they may be expected to overlap for more than τ to be considered identical (as e.g. in [1]). Examples of such *match*-functions for clusters can be found in [40], while model comparison frameworks like FOCUS [18], PANDA [6] and PSYCHO [32] provide both example *match*-functions and generic mechanisms for building user-defined ones.

The second solution to the matching challenge is the association of each local model with a unique identifier. Quite intuitively, this approach is taken in incremental data mining, where each component of the model is adjusted to the arriving new records and to the deleted/forgotten ones. For example, IncrementalDBSCAN checks whether a cluster has lost some of its dense areas (called: neighbourhoods) after data record deletion and may thus be split to more than one clusters or even disappear altogether [17].

The task of tracing a local model, i.e. a component of a model is interwoven with the task of identifying the changes that may have occurred upon it. Most change mining algorithms have an embedded set of changes or *transitions* that they can trace. For example, Kalnis et al trace *location shifts* in clusters over a feature space [23], while Aggarwal's mechanism traces additionally contractions, dissipations and growth of individual clusters and can further capture the velocity of change across different dimensions/features [1]. Additionally, cluster splits and merges can be traced e.g. by the mechanisms of [17; 40]. Moreover, there are mechanisms that map a local model to a condensed representation (a summary) and focus on transitions of this summary, including disappearance, split or merge [19; 33; 3].

The monitoring of change encompasses also the interpretation of change. Methods that map local models into summaries *and* associating such a summary with interpretable semantics do a great step in this direction. For example, cluster evolution in text stream clustering can be mapped to the evolution of human-understandable topics (cluster summaries as weighted vectors of terms), as done among else in [33; 38; 39]. Then, change monitoring can be used to describe change of existing topics and the emergence of new ones [38; 39] or to predict further changes [33].

In many cases, change interpretation is greatly assisted by pictorial or even visual representations. For example, the naming scheme used for the cluster transitions captured in [1; 23] do an excellent service in helping the human expert into a pictorial understanding of change: A cluster may "move", "grow", "shrink", "disappear" etc. As another example, the framework FOCUS incorporates a representation of changes in the nodes of a decision tree classifier [18] which can be visualized in an intutive way.

Change interpretation incurs particular challenges in the context of association rules' discovery. As already mentioned, matching of individual rules across different timepoints is trivial. However, rules may overlap and the changes encountered for one rule may lead to a cascade of changes upon other rules. Liu et al attempt to solve this issue by detecting so-called "fundamental changes" in a set of association rules, i.e. changes that are responsible for all changes seen in the set [28].

A final challenge associated with change monitoring and interpretation is the interplay between change in the population and stability of the mining algorithm used to discover the models (cf. [22]). Some algorithms are very sensitive to small variations of the dataset, among them decision tree classifiers. Others are sensitive to outliers, among them K-Means for clustering. Many algorithms are sensitive to inherent properties of the dataset such as the existence of cluster formations of particular shape. Further, almost all algorithms are sensitive to the selection of the values for their initialization parameters, including random settings (like the initialization seeds for center-based partitioning methods in clustering). For this reason, for any type of change, there is still the open issue of distinguishing between *real change* in the data and change inherent to the instability of a poor model.

5. CONCLUSIONS AND OUTLOOK

We have presented *change mining*, a new perspective on knowledge discovery upon data, which brings forward the volatility and the dynamics of real world applications. The object of study in change mining are models and patterns learned from a non-stationary population. Its objective is to detect and analyze *when* and *how* changes occur, including the quantification, interpretation and prediction of change.

Change mining builds upon research advances on incremental mining, temporal mining, novelty detection, stream mining, pattern maintenance and pattern evolution monitoring. Much research work in all but the last two fields concentrates on the *adaptation* of patterns and models as the data generating process changes. The capturing of the changes themselves is sometimes a by-product of the adaptive process, the interpretation and tracing of change is beyond the scope of adaptive methods. In contrast, change mining observes change as an inherent and indispensable aspect of dynamic environments. Accordingly, it encompasses methods that model and trace patterns across time, detect and quantify changes and provide the underpinnings for change interpretation and prediction.

In this study, we have proposed a fundamental framework for change mining. We have specified an abstract methodological process, described the building blocks for this process and identified exemplary research advances that can be used to implement these blocks.

Despite the wealth of research results that can be used for change mining, a fair amount of research effort is required in this new field, mostly due to the demand of studying evolving patterns rather than static data. In the previous sections, we have identified the need to model patterns as objects across the time axis and to devise mechanisms for monitoring them. We have stressed the difference between studying the evolution of a model (e.g. a classifier) as a whole and studying how a single cluster or classification rule changes within a global model. We have brought forward the necessity of designing methods and measures for capturing, quantifying and predicting change. We have finally stressed the need for distinguishing between real change and artefact, where artefacts can be caused by noise in the data or by instability and evolution of the data mining process itself.

A major objective of knowledge discovery from data is the understanding and prediction of the population which generates the data. Institutions pursue this objective by accummulating data in an ever increasing pace. Data proliferation makes the need for maintaining patterns rather than the data themselves more and more pressing. This implies that knowledge discovery from data should extend to knowledge discovery *from patterns*, of which change mining is a central component.

Change Mining aims at identifying changes in an evolving domain by analyzing how models and patterns change. However, often the data mining process evolves too and the effects of this evolution on a model superimpose those of the underlying domain. Two reasons why the data mining process evolves can be frequently encountered: Firstly, many businesses steadily improving the quality of gathered data. Secondly, the parameters that drive the data mining algorithms are adjusted and tweaked over time. With no doubt, such interventions into the data mining process are important, necessary and useful. Still, they raise the challenge of seperating changes imposed by an evolving domain from those of an evolving data mining process.

6. REFERENCES

[1] C. Aggarwal. On change diagnosis in evolving data streams. *IEEE TKDE*, 17(5):587–600, May 2005.

[2] C. Aggarwal, J. Han, J. Wang, and P. Yu. A framework for clustering evolving data streams. In *Proc. of Int. Conf. on Very Large Data Bases (VLDB'03)*, 2003.

[3] 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.

[4] R. Agrawal and G. Psaila. Active data mining. In M. Fayyad, Usama and R. Uthurusamy, editors, *Proceedings of the 1st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3–8, Montreal, Quebec, Canada, 1995. AAAI Press, Menlo Park, CA, USA.

[5] 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.

[6] I. Bartolini, P. Ciaccia, I. Ntoutsi, M. Patella, and Y. Theodoridis. A unified and flexible framework for comparing simple and complex patterns. In *Proc. of ECML/PKDD 2004*, Pisa, Italy, Sept. 2004. Springer Verlag.

[7] P. Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1-3):217–239, 2005.

[8] M. Boettcher, D. Nauck, D. Ruta, and M. Spott. Towards a framework for change detection in datasets. In *Proceedings of the 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 115–128. Springer, 2006.

[9] M. Boettcher, D. Nauck, D. Ruta, and M. Spott. A framework for discovering and analyzing changing customer segments. In *Proceedings of the 7th Industrial Conference on Data Mining (ICDM2007)*, LNAI 4597, pages 255–268. Springer, 2007.

[10] L. Breiman. The heuristics of instability in model selection. *Annals of Statistics*, 24:2350–2383, 1996.

[11] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-Based Clustering over an Evolving Data Stream with Noise. In *Proc. SIAM Conf. Data Mining*, 2006.

[12] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining Surprising Patterns Using Temporal Description Length. In A. Gupta, O. Shmueli, and J. Widom, editors, *VLDB'98*, pages 606–617, New York City, NY, August 1998. Morgan Kaufmann.

[13] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 606–617. Morgan Kaufmann Publishers Inc., 1998.

[14] M.-C. Chen, A.-L. Chiu, and H.-H. Chang. Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 28(4):773–781, 2005.

[15] G. Dong, J. Han, and L. Lakshmanan. Online mining of changes from data streams - research problems and preliminary results. In *Proceedings of the ACM SIGMOD Workshop on Management and Processing of Data Streams*, June 2003.

[16] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 43–52, 1999.

[17] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental Clustering for Mining in a Data Warehousing Environment. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 323–333, New York City, New York, USA, August 1998. Morgan Kaufmann.

[18] V. Ganti, J. Gehrke, and R. Ramakrishnan. A Framework for Measuring Changes in Data Characteristics. In *Proc. of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 126–137, Philadelphia, Pennsylvania, May 1999. ACM Press.

[19] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS: Clustering categorical data using summaries. In *Proc. of 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '99)*, pages 73–83, San Diego, CA, Aug. 1999. ACM Press.

[20] V. Ganti, J. Gehrke, and R. Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. In *Proc. of the 15th Int. Conf. on Data Engineering (ICDE'2000)*, pages 439–448, San Diego, CA, USA, Feb. 2000. IEEE Computer Society.

[21] V. Guralnik and J. Srivastava. Event detection from time series data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42, New York, NY, USA, 1999. ACM.

[22] F. Höppner and M. Böttcher. Matching partitions over time to reliably capture local clusters in noisy domains. In *Principles and Practice of Knowledge Discovery in Databases PKDD*, pages 479–486, Warsaw, Poland, 2007. Springer.

[23] P. Kalnis, N. Mamoulis, and S. Bakiras. On Discovering Moving Clusters in Spatio-temporal Data. In *Proc. of 9th Int. Symposium on Advances in Spatial and Temporal Databases (SSTD'2005)*, number 3633 in LNCS, pages 364–381, Angra dos Reis, Brazil, Aug. 2005. Springer.

[24] E. Keogh, S. Lonardi, and B. Y. chi' Chiu. Finding surprising patterns in a time series database in linear time and space. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 550–556, New York, NY, USA, 2002. ACM.

[25] J. K. Kim, H. S. Song, T. S. Kim, and H. K. Kim. Detecting the change of customer behavior based on decision tree analysis. *Expert Systems*, 22(4):193–205, 2005.

[26] R.-H. Li and G. G. Belford. Instability of decision tree classification algorithms. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 570–575, New York, NY, USA, 2002. ACM.

[27] B. Liu, W. Hsu, H.-S. Han, and Y. Xia. Mining changes for real-life applications. In *Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery*, pages 337–346, London, UK, 2000. Springer.

[28] B. Liu, W. Hsu, and Y. Ma. Discovering the set of fundamental rule changes. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 335–340, 2001.

[29] B. Liu, Y. Ma, and R. Lee. Analyzing the interestingness of association rules from the temporal dimension. In *Proceedings of the IEEE International Conference on Data Mining*, pages 377–384. IEEE Computer Society, 2001.

[30] B. Liu and A. Tuzhilin. Managing large collections of data mining models. *Communications of ACM*, 51(2):85–89, Feb. 2008.

[31] J. Ma and S. Perkins. Online novelty detection on temporal sequences. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–618, New York, NY, USA, 2003. ACM.

[32] 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.

[33] Q. Mei and C. Zhai. Discovering Evolutionary Theme Patterns from Text - An Exploration of Temporal Text Mining. In *Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'05)*, pages 198–207, Chicago, IL, Aug. 2005. ACM Press.

[34] O. Nasraoui, C. Cardona-Uribe, and C. Rojas-Coronel. Tecno-Streams: Tracking evolving clusters in noisy data streams with an scalable immune system learning method. In *Proc. IEEE Int. Conf. on Data Mining (ICDM'03)*, Melbourne, Australia, 2003.

[35] M. Pěchouček, O. Štěpánková, and P. Mikšovský. Maintenance of Discovered Knowledge. In *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science, pages 476–483, Prague, Czech Republic, September 1999. Springer.

[36] E. L. Rissland and M. T. Friedman. Detecting change in legal concepts. In *ICAIL '95: Proceedings of the 5th International Conference on Artificial Intelligence and Law*, pages 127–136, New York, NY, USA, 1995. ACM.

[37] J. F. Roddick, M. Spiliopoulou, D. Lister, and A. Ceglar. Higher order mining. *submitted for publication*, 2007.

[38] R. Schult and M. Spiliopoulou. Discovering emerging topics in unlabelled text collections. In *Proc. of ADBIS'2006*, Thessaloniki, Greece, Sept. 2006. Springer.

[39] S. Schulz, M. Spiliopoulou, and R. Schult. Topic and cluster evolution over noisy document streams. In F. Masseglia, P. Poncelet, and M. Teisseire, editors, *Data Mining Patterns: New Methods and Applications*. Idea Group, 2007.

[40] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. Monic – modeling and monitoring cluster transitions. In *Proc. of 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'06)*, pages 706–711, Philadelphia, USA, Aug. 2006. ACM.

[41] M. Vazirgiannis, M. Halkidi, and D. Gunopoulos. *Uncertainty Handling and Quality Assessment in Data Mining*. Springer, 2003.

[42] H. Yang, S. Parthasarathy, and S. Mehta. A generalized framework for mining spatio-temporal patterns in scientific data. In *Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'05)*, pages 716–721, Chicago, IL, Aug. 2005. ACM Press.

[43] K. Zhang, J. T. L. Wang, and D. Shasha. On the editing distance between undirected acyclic graphs and related problems. In Z. Galil and E. Ukkonen, editors, *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*, pages 395–407. Springer-Verlag, Berlin, 1995.

[44] X. Zhang, G. Dong, and R. Kotagiri. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 310–314, 2000.

[45] A. Zhou, C. Feng, W. Qian, and C. Jin. Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems*, 2007.