A Knowledge Mining Framework for Business Analysts

Themis Palpanas

University of Trento
themis@disi.unitn.eu

Abstract—**Several studies have focused on problems related to data mining techniques, including several applications of these techniques in the e-commerce setting. In this work, we describe how data mining technology can be effectively applied in an e-commerce environment, delivering significant benefits to the business analyst. We propose a framework that takes the results of the data mining process as input, and converts these results into actionable knowledge, by enriching them with information that can be readily interpreted by the business analyst. The framework can accommodate various data mining algorithms, and provides a customizable user interface.**

We experimentally evaluate the proposed approach by using a real-world case study that demonstrates the added benefit of the proposed method. The same study validates the claim that the produced results represent actionable knowledge that can help the business analyst improve the business performance, since it significantly reduces the time needed for data analysis, which results in substantial financial savings.

ACM Categories: H.2.8 Database Applications, H.3 Information Storage and Retrieval

Keywords: data mining, knowledge mining, business analyst

1. INTRODUCTION[1]

Data mining (Hand et al. 2000) has been extensively used in the past with the purpose of helping the analyst quickly examine huge collections of data, and extract new knowledge about the processes that generated those data. Examples of data mining applications include star formation discovery in astronomy data, drug discovery in bioinformatics, targeted advertising in customer relationship management, market basket analysis in retail data, buying patterns identification in online stores, and many others (Han et al. 2006).

More recently, data mining techniques have also been proposed and used in the specific context of e-commerce (Li et al. 2005; Yang et al. 2005). In this case, one of the concerns is how to integrate these techniques with the overall business process, and take advantage of the benefits they have to offer. For example, one of the studies examines the utility of employing data mining for performing risk analysis of customs declaration cargo (Li et al. 2005). The study shows how the results of the analysis helped to transform the associated business process, so that it became more efficient and effective.

We argue that an important problem regarding the use of data mining tools by business analysts is that often, information necessary to make business decisions is absent from data mining. In those cases, the results of the data mining process have to be enriched before they can be transformed into actionable knowledge.

In this work, we describe a framework that demonstrates how data mining technology can be effectively applied in an e-commerce environment, in a way that delivers immediate benefits to the business analyst. The main idea behind our approach is to concentrate on the results of the data mining procedure, and then merge any other related piece of information to these results. The reason for doing this is the following. The data mining results per se are usually rather poor in information content, carrying only the necessary bits of data for achieving a particular mining task. Nevertheless, this amount of information is not sufficient for the business analyst, who needs to have a complete view of the data in order to make business decisions.

Under the proposed framework, all the relevant information is gathered around the mining results, which now represent actionable knowledge. These data are organized according to different dimensions, where each dimension represents a particular aspect of the objects represented in the data mining results.

[1] Work done while the author was a member of the IBM T.J. Watson Research Center. An early version of this work has appeared in the LNCS Proceedings of DEXA (Palpanas et al. 2008).

Once the business analyst has all the above information at hand, she is in a position to interpret the mining results and make informed business decisions. The multi-dimensional organization of the results allows the analyst to view the data from different perspectives, group them according to various business criteria, and analyze them further, just like any other collection of business data.

In order to demonstrate the benefits of the proposed approach, we applied our framework to a real-world scenario, namely to the problem of identifying and managing failures in manufactured products. In the manufacturing industry, it is very common that some of the produced products have defects, or fail during usage because of a wrong design or some inappropriate manufacturing process. In either case, we say that there is a failure in the product. When the products we are examining are complex, e.g., a vehicle or an airplane, it is very important to be able to identify such failures as early as possible, and trace them back to the root cause in the design and manufacturing process.

Our case study, where we examined real data coming from the vehicle manufacturing industry, revealed that there exist many frequent sequences of the same set of failures. That is, many vehicles share a specific set of failures that occur in the same order (in time) for all those vehicles. Mining the data for frequent failure patterns is essentially a way of searching for dependencies among the failures. The failure patterns we identify reveal the trends that have been occurring in the past, but are also indicative of the future trends. These failure patterns are important, because they are entire sequences of the same failures occurring in many different vehicles. This indicates that the specific failures are correlated, and may lead to the identification of deeper problems. Furthermore, these patterns represent actionable knowledge, since they can be correlated to the manufacturing process of the vehicle, and may be translated to errors in the design or the production phases.

Since the final output of our method can be quite sizable, we capture all the detailed results in a database specifically created to facilitate the further inspection and understanding of these results. Subsequently, these results are correlated with relevant information from the other databases in the business, thus, creating a rich context for the analysis of the data mining results.

For example, in our case study, the results of the mining process are correlated with detailed information regarding the manufacturing and the demographics of each one of the vehicles that are part of the mining results. The user can then use a customized interface in order to access a variety of reports that present the results along several different dimensions that are useful to the business analyst. What is important to note here is that in our framework it is very easy to change and customize the interface, so that the reports match the needs of the analysts.

In summary, the method we propose allows the analyst to:

- quickly discover frequent, time-ordered, event-patterns,
- automate the process of event-pattern discovery using a feedback loop,
- enrich these event-patterns with demographics related to the processes that generated them,
- identify the commonalities among these event-patterns,
- trace the events back to the process that generated them, and
- predict future events, based on the history of event-patterns.

The rest of this paper is organized as follows. In Section 2 we review the related work and give some background on frequent sequence mining. We introduce the knowledge mining framework in Section 3, and present a case study, where we apply the proposed framework, in Section 4. Finally, we conclude in Section 5.

2. RELATED WORK

There exists a sizable literature in the area of knowledge extraction from huge datasets (Han et al. 2006), and many studies focus on improving the performance and functionality of the various data mining algorithms (Aggarwal et al. 2004; Agrawal et al. 1994; Evfimievski et al. 2004; Ganti et al. 2002; Keogh et al. 2002; Manerikar et al. 2009; Palpanas et al. 2001; Palpanas et al. 2005; Srikant et al. 1996; Tantoro et al. 2008; Yamanishi et al. 2004). These algorithms include clustering, classification, association rules, frequent sequences, outliers, and others. The main goal of these algorithms is to make the data mining process scale to very large datasets.

In all the above cases, the focus is on devising efficient algorithms that can process large amounts of data in a small timeframe. As a result, these algorithms are targeted to expert users, and are very cumbersome for use by business analysts.

Because of their popularity as data mining techniques, some of the above algorithms are being offered as part of commercial data mining solution packages (DB2 Intelligent Miner 2011; IBM SPSS Modeler 2011; Microsoft SQL Server Business Intelligence 2011; Oracle Data Mining 2011; SAS Enterprise Miner 2011), as well as open source software (Orange 2011; RapidMiner 2011; Weka 2011). These packages hide the intricate details from the users, and only expose graphical user interfaces, which make the use of these algorithms more accessible. The framework we propose in this work proposes a technique for enriching the results of the algorithms with other relevant data, in such a way that it provides several different views that are useful to the business analyst.

The CRISP-DM project (Cross Industry Standard Process for Data Mining 2011) and SAS SEMMA (Paper 2008) have proposed an end-to-end methodology for data mining, starting from understanding the business in which it will be applied, and ending with the assessment of the proposed solution. In this paper, we present a specific framework that addresses some of the problems arising during this process, namely, on how to organize and present to the user the new knowledge gained by applying some data mining techniques. (We further discuss the relationship of the proposed framework to these methodologies in Section 4.) There also exist standards for defining statistical and data mining models, such as the Predictive Model Markup Language (PMML) (Predictive Model Markup Language 2011), which is an XML-based language. The main goal of these standards is to enable the sharing of models among relevant applications.

There are several applications of data mining in the e-business environment (Li et al. 2005; Yang et al. 2005), which prove the usefulness of this kind of techniques for improving the business operations. These studies show that significant improvements can be achieved by the use of the results of data mining. However, the proposed solutions are targeted to the specific tasks for which they were developed. In contrast, we propose a framework that is more general than the above solutions that can be used in a variety of domains.

Finally, there is work on the problem of applying data mining techniques in the context of data warehouses (Palpanas 2000; Sarawagi 2000; Sarawagi et al. 1998; Sathe et al. 2001). Data warehouses are usually employed by business analysts to assist them during the decision-making process. These techniques are very effective in helping the business analyst focus on the interesting parts of the data in the warehouse. However, they are targeted specifically to data warehouses, and are not directly related to the framework we propose in this paper.

A. Frequent Sequence Mining

In the following paragraphs, we provide some necessary background on the problem of frequent sequence mining, which was the focus of our case study.

Data mining for sequential patterns is a problem that has attracted lots of attention (Agrawal et al. 1995; Ayres et al. 2002; Mannila et al. 1997; Srikant et al. 1996). It is closely related to association rules discovery (Agrawal et al. 1994; Evfimievski et al. 2004), and has many applications both for industrial and for scientific purposes.

Mining for sequential patterns requires taking into account the time (in contrast to association rules mining that does not have this requirement). In this case, transactions are sets of items ordered by time, and it is exactly the time sequence, in which events occur, that is of special interest. In sequential pattern mining, we are looking for patterns that occur across transactions. An example of a sequential pattern is the following.

> *5% of customers bought an mp3 player and headsets in one transaction, followed by*
> *speakers for the mp3 player in a later transaction.*

It is not necessary for the second transaction to immediately succeed the first one, as long as it comes after the first one.

1) Formal Model

A formal statement of the problem was proposed in (Agrawal et al. 1995). Let D be a database of customer transactions, each transaction consisting of three fields: customer-id, transaction-time, and the purchased items. Let *itemset* $i = (i_1 i_2 \ldots i_m)$ be a non-empty set of items, and *sequence* $s = < s_1 s_2 \ldots s_n >$ be an ordered list of itemsets. An itemset is large, also called *litemset*, if the fraction of customers who bought all the relevant items, is greater than the minimum support.

A sequence $< a_1 a_2 \ldots a_n >$ is contained in another sequence $< b_1 b_2 \ldots b_n >$ if there exist integers $i_1 < i_2 < \ldots < i_n$, such that a_1 is a subset of b_{i1}, a_2 subset of b_{i2}, …, a_n subset of b_{in}. In a set of sequences, a sequence s is maximal if s is not contained in any other sequence.

A sequence which contains all the transactions of a user ordered by increasing time, is called a customer-sequence. A customer *supports* a sequence if this is contained in the customer-sequence for this customer. A sequence is *large* if the fraction of total customers who support this sequence, is greater than the minimum support threshold. Note that a large sequence is exclusively composed from *litemsets*.

Several algorithms have been proposed for the solution of the mining sequential patterns problem (Agrawal et al. 1995; Ayres et al. 2002; Mannila et al. 1997; Srikant et al. 1996). The above algorithms can also solve more general variations of the problem, by introducing three new features. The first is time constraints that specify a minimum and a maximum time period between adjacent elements in a sequential pattern. The second is that items belonging to different transactions that are close enough in time to one-another, may be considered as belonging to the same transaction. Finally, taxonomies (*is-a* hierarchies) are integrated in the operation of the algorithms, so that the mined sequential patterns may contain items across all levels of the taxonomies. These features offer a much richer processing environment, where patterns of finer granularity of detail can be mined.

Note that the specific choice of algorithm is orthogonal to the framework we propose, and any of the available frequent sequence mining algorithms can be used.

2) *Proposed Algorithms*

We now briefly describe algorithms for solving the frequent sequence mining problem. First, we present the Apriori algorithm (Agrawal et al. 1995), which consists of the following steps.

1. Scan the database to identify all the frequent sequences of unit length, i.e., 1-element sequences.
2. Iteratively scan the database to identify all frequent sequences. During iteration k, discover all the frequent sequences of length k (k-sequences) as follows.
   - Generate the set of all candidate k-sequences by joining two (k-1)-sequences.
   - Prune the candidate sequence if any of its k-1 contiguous subsequence is not frequent.
   - Scan the database to determine if the remaining candidate sequences are frequent.
3. Terminate when no more frequent sequences can be found.

The above algorithm has been extended, in order to solve a more general form of the problem, by introducing three new features (Srikant et al. 1996). First, it adds time constraints that specify a minimum and a maximum time period between adjacent elements in a sequential pattern. Second, it allows for items that belong to different transactions that are close enough in time to one-another, to be considered as belonging to the same transaction. Third, it handles taxonomies (is-a hierarchies), so that the mined sequential patterns may contain items across all levels of the taxonomy.

A slightly different approach is presented in (Mannila et al. 1997). The term *episodes* is used to describe collections of events occurring frequently close to each other. There are serial episodes, where the events have to occur in a specific time order, parallel episodes, where no time constraints are given, and complex episodes which are a combination of the above two.

The discovery of frequent episodes is made by building new candidate episodes and evaluating their actual frequency with a pass over the data. Complex episodes are handled by decomposing then into serial and parallel ones. Events pertaining to an episode should occur within a specific time window.

Two variations of the algorithm are given, **Winepi** and **Minepi**. The former discovers frequent episodes which are constrained in a time window, while the latter relaxes this constraint, and is able to come up with rules bearing two different time windows in the left and right hand side (e.g., “if A and B occur within 15 seconds, then C follows within 30 seconds”).

Other algorithms for solving the frequent sequence mining problem have also been proposed more recently, where the focus is on optimizing the resource consumption (e.g., the SPAM algorithm (Ayres et al. 2002)). These algorithms aim at efficiently searching the candidate space of all possible frequent sequences, and result in performance improvements for several specialized cases of the problem.

For our study, we have used the Apriori algorithm. Nevertheless, the specific choice of algorithm is orthogonal to the framework we propose, and any of the other frequent sequence mining algorithms can be used instead.

3. KNOWLEDGE MINING FRAMEWORK

In this section, we present in more detail the framework we propose for the task of knowledge mining. A high level view of our approach is shown in Figure 1.

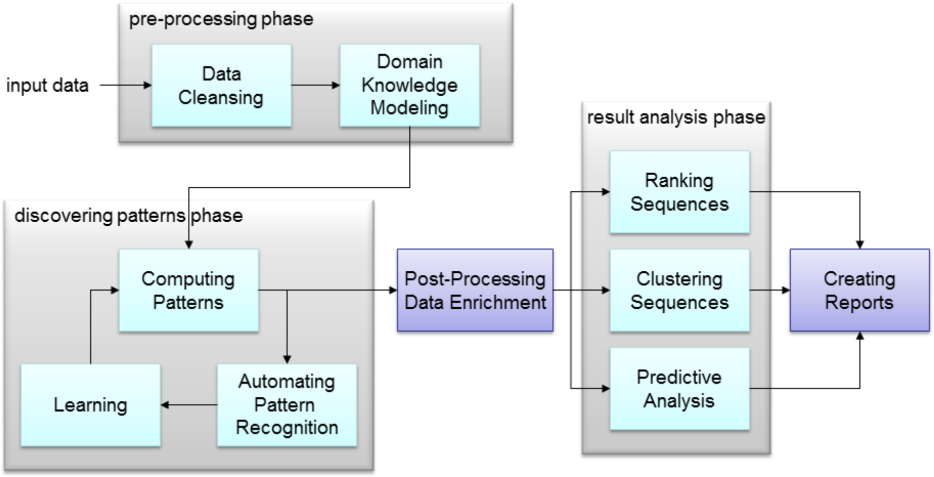


Figure 1. Process overview.

Pre-Processing

At the time when data first enter the system, we apply a series of pre-processing steps that aim at cleaning the data and transforming them in a format suitable for our system. Remember that the incoming data may be generated by several different external systems, according to various specifications and formats. The purpose of these pre-processing steps is to make sure that all data conform to the same standard, as well as that they are semantically equivalent, so that they can be combined in a reasonable manner.

Examples of actions taken during this phase are the following.

- Convert all measures to metric system.
- Convert all codes from different external sources to the same standard.
- Apply domain knowledge to ensure semantic equivalence of data coming from different sources.

Discovering Patterns

In the following set of steps, we analyze the data in order to identify interesting patterns. These patterns should conform to some user-defined constraints that determine the form of the patterns that are going to be computed. For example, the user may focus the analysis by setting minimum and maximum lengths for each of the reported patterns (in the case of frequent sequences). The user may also set a maximum time between the start and the end of the sequence, or between two consecutive items of the same sequence.

Obviously, different settings for the parameters will result in different sets of discovered patterns. The proposed framework allows the users to try out the various alternatives, and customize the mining process to their needs and the specific data they are working with. (The above functionality is in principle similar to exploratory mining (Ng et al. 1998)).

Data Enrichment

When the mining phase of the process is over and the system has produced the final set of frequent sequences, we proceed to the post-processing step of data enrichment. The frequent sequences that come out as the result of the previous step only hold some minimal, vital information: this is the information needed to identify the items that participate in each one of these frequent sequences. The goal of the data enrichment step is to correlate the identified frequent sequences with all the relevant pieces of information that are stored in the system.

In order to achieve this goal, data originating from different parts of the business are gathered and integrated with the data mining results. These data naturally refer to different phases of the lifecycle of each specific item, and as such they enrich the identified frequent sequences with contextual information relevant to the processes that generated them. When all these data are put together, the business analyst can drill down on the root causes for the patterns that have been identified by the mining phase.

The above procedure represents an important step in the framework we describe, since it augments the data mining results with information that renders those results useful for business intelligence. A more concrete discussion on this topic, relevant to the vehicle manufacturing business, follows in Section 4.

Result Analysis

Next, the enriched sequences that are produced during the previous step are analyzed in different ways. In Figure 1, we depict three different analysis modules that we are proposing.

- **Ranking Sequences:** The first module uses various knowledge models and utility functions, and its purpose is to rank the event-patterns according to different criteria. The ranking mechanism combines objective business metrics with subjective domain knowledge. The results of this technique are able to capture the most important business performance issues based on a small but representative fraction of the available information. Therefore, the analyst can examine huge volumes of data, and quickly focus on the data most relevant to a particular aspect of the business. (For a more elaborate discussion, the interested reader should refer to our previous work (Chen et al. 2005))
- **Clustering Sequences:** The purpose of the clustering module is to use the contextual information of each event-pattern in order to identify clusters of similar event-patterns. When a clustering algorithm (for example, K-means MacQueen 1967) or some subspace clustering method (Domeniconi et al. 2004)) is run against the enriched event-patterns, it produces groupings of those patterns that are semantically meaningful within the business context. These groups (or clusters) are a summarized representation of the behavior of the participating event-patterns. Therefore, they help analyze the aggregated behavior of several business objects, and enable the analyst to gain insight on the root causes for each behavior.
- **Predictive Analysis:** The third module aims at using the history of event-patterns in order to predict future events. The identified patterns represent (by definition) an approximation of the past behavior of the items under examination. Given these data, we can make projections for the future behavior of the same items (Pednault 2004).

This kind of analysis is extremely useful to the business analysts, since it enables them to make predictions and anticipate future events (based on the assumption that past behavior is a good indicator of future behavior).

We note that the above list of analysis modules is not exhaustive; other modules can be easily incorporated in the proposed framework as well.

Report Creation

In the final phase of the proposed framework, we produce a set of reports with the aim of summarizing the results of the preceding data analysis phases. To this extent, we have developed a graphical user interface that enables the analyst to access and customize several different report types.

4. Case Study

In order to evaluate the effectiveness of the proposed framework, we applied it to a real-world problem. In this section, we present a case study with one of the two vehicle manufacturing companies we collaborated with.

The manufacturer has data detailing the characteristics of each vehicle that appears in some warranty claim. For each vehicle, we know the model type, the time and place of its assembly, and also the type of some of its components[2], such as the engine. Apart from the above information, we also have data regarding the warranty claims for each vehicle, which describe the behavior of the vehicle after it has left the factory. Thus, we know at which points in time the vehicle failed, and also what the reason of the failure was, as identified by the mechanic. For those incidents, we also record the vehicle parts that failed, and the related costs (that is, parts and labor).

A. Proposed Process

We now elaborate on the process we propose in order to extract new knowledge from the available data, and specifically for the vehicle manufacturing company.

Figure 2 depicts an overview of the approach. Note that here we give a more high-level view than that illustrated in Figure 1. In the description that follows, we focus on a particular task of knowledge extraction and mining, namely, that of frequent sequences.

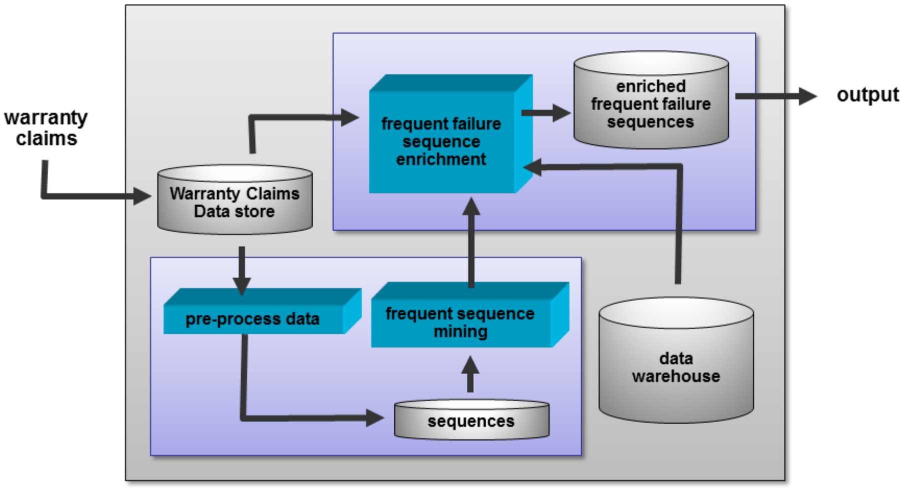

warranty claims
Warranty Claims Data store
frequent failure sequence enrichment
enriched frequent failure sequences
output
pre-process data
frequent sequence mining
sequences
data warehouse

Figure 2. The frequent failure mining process.

Pre-Processing

As shown in Figure 2, the input to our system are the warranty claims associated with each vehicle. Each warranty claim is associated with the following pieces of information.

- The Vehicle Identification Number (VIN).

[2] Of course, vehicle manufacturers also maintain databases with detailed information on the components of each vehicle. However, for this study, we did not have access to those databases.

- The date that the vehicle visited the mechanic for a particular failure (or for a set of failures)[3].
- The mileage of the vehicle at the time of the visit to the mechanic.
- The ids for all the failed vehicle parts, as identified by the mechanic.
- The root cause for each one of the failures, as identified by the mechanic.
- The cost of the visit to the mechanic, broken down by part cost and labor cost.

These warranty claims are gathered from several different sources, they are normalized, and then stored in a database called warranty claims data store. The use of such a database is useful for the subsequent steps of data analysis.

During the next phase, we pre-process the warranty claim data, in order to put them in a form that is more suitable for the frequent sequence discovery step. Let D be the database of warranty claims. We logically divide each claim into four parts.

1. The Vehicle Identification Number (VIN),
2. the date of the claim,
3. the identified failure,
4. and the rest of the attributes.

Frequent Sequence Mining

Following the discussion in Section 2.1, we define failure set $f = (f_1, f_2, \ldots, f_m)$ to be a nonempty set of failures (i.e., a set describing all the known events in which a particular vehicle failed), and failure sequence $s =< s_1, s_2, \ldots, s_n >$ to be an ordered list of failure sets. We call a vehicle-failure sequence the failure-sequence that contains all the failures of a vehicle (identified by the VIN) ordered by the claim date. If the vehicle-failure sequence contains a particular failure sequence, then we say that this vehicle supports this particular failure sequence. Then, the failure pattern mining problem is to discover all failure sequences that are supported by many vehicles. Usually, we are only interested in failure sequences that have a minimum support, that is at least a certain number of vehicles have exhibited the same failure sequence. Note that the support of a failure sequence is defined as the percentage of all the vehicles in the database that have exhibited the particular failure sequence, and as such, it bears a statistical significance.

Example 1: *Assume that in the warranty claims database we have the (simplified) entries shown in Table 1. In this case, the only sequence of failures that is common among more than one of the vehicles is: (300, 500), 450. In this sequence, the first two failures co-occur, that is, they happen on the same date. The vehicles that exhibit the above sequence are the vehicles with VIN numbers 1 and 2. Note that in the series of failures for vehicle 2, there is another failure, 150, that takes place between (300, 500) and 450. This is allowed in our analysis.*

VIN	claim fail date	failure
1	15 Jan 2005	300
1	15 Jan 2005	350
1	15 Jan 2005	500
1	12 Feb 2005	450
2	28 Jan 2006	200
2	5 Mar 2005	300
2	5 Mar 2005	500
2	12 Mar 2005	150
2	29 Mar 2005	450
3	17 Feb 2005	300

Table 1. Example dataset of vehicle failures.

Post-Processing Data Enrichment

The frequent failure patterns analysis step produces a long list of failure sequences, which we enrich with statistics that relate to several of the attributes contained in the warranty claims database. In particular, we associate the following information with each failure pattern produced by the previous step.

[3] For ease of exposition, in this paper we treat the date of the failure, the date of the visit to the mechanic, and the date of the warranty claim as the same.

- Number of vehicles supporting the failure pattern.
- Five most common vehicle models (among the models of the vehicles that support the failure pattern).
- Five most common engine types (among the engine types of the vehicles that support the failure pattern).
- Five most common manufacturing plants (among the plants that manufactured the vehicles that support the failure pattern).
- Five most common makers (among the makers of the vehicles that support the failure pattern).
- Five most common build-years (among the build-years of the vehicles that support the failure pattern).

In addition to the information mentioned above, which is relevant to the entire pattern, we also associate some information to each particular failure (of each failure pattern).

- Five most common cause-codes for the failure (among the cause-codes of the particular failure, for all the vehicles that support the failure pattern).
- Minimum, maximum, average, and standard deviation of the mileage at which the failure occurred (among the mileages when the particular failure occurred, for all the vehicles that support the failure pattern).
- Minimum, maximum, average, and standard deviation of the replacement part cost for the failure (among the part costs for the particular failure, for all the vehicles that support the failure pattern).
- Minimum, maximum, average, and standard deviation of the labor part cost for the failure (among the labor costs for the particular failure, for all the vehicles that support the failure pattern).

Result Analysis - Report Creation

The amount of information produced in the previous steps is such that the use of a database is imperative, in order to effectively organize all the results and help in their analysis in an efficient manner. This database may be used merely to produce a listing with all the failure patterns along with the additional statistics mentioned above. Nevertheless, the real power of this approach stems from the fact that the analyst can interactively query the database, and thus, obtain results relevant to the specific aspects of the failure patterns she is focusing on.

Indeed, we can use the failure pattern database in order to answer any query that correlates any combination of the attributes that the database captures (listed in the previous paragraphs). A small sample of the questions that this database can answer is the following.

(a) Show the k most frequent failure patterns.
(b) Show the failure patterns that involve more than n failures.
(c) Show the k most expensive-to-fix failure patterns.
(d) Show the failure patterns that involve engine components, and the labor cost is greater than the replacement-parts cost.
(e) Show the failure patterns that involve at least k failures whose most common cause code is X.
(f) Show the k most expensive-to-fix failure patterns that on average occur when the vehicle mileage is between p and q miles.
(g) Show the k most frequent patterns for vehicles whose most common engine type is X.
(h) Show the failure patterns with the largest number of failures, for which the most common manufacturing plant of the vehicles that support those failure patterns is Y.

Obviously, these kinds of queries have the power to turn the output of the frequent failure patterns approach into actionable knowledge. They can help interpret failure symptoms, identify underlying problems, and plan corrective actions.

Another benefit of using the above database is that it enables the use of data analytics and reporting tools (DB2 Alphablox 2011). These tools can connect to a database and export a rich graphical user interface for exploring the available data and for producing relevant reports.

B. Evaluation Results

We now present the results of applying the proposed approach to analyze data coming from two different companies in the vehicle manufacturing industry.

The data refer to warranty claims made for vehicles of particular models during the period of the year 2005. The first dataset we examined includes almost 2,000,000 records of warranty claims, referring to almost 1,000 different failure reasons. These claims were made by approximately 250,000 unique vehicles, corresponding to more than 100 different vehicle models.

We should note that this dataset is skewed. For example, 35% of the claims are on vehicles that only failed once, but there are also vehicles that failed more than 10 times during the same time period (i.e., one year). Another example is that 90% of the engine claims refer to 5 specific failures (out of the 167 possible failures).

The second dataset is smaller in size (300,000 warranty claims), but has the same overall characteristics. For the sake of brevity, in this paper we only present results on the first dataset. As mentioned earlier, we provided the interested parties with the toolset to analyze the available data for frequent sequences of failures, and subsequently, to explore the results and view them from several different angles. In the following paragraphs, we discuss some sample results, and we show screenshots from the procedure.

1) Aggregated Behavior Reports

We now present some indicative results on failure sequence statistics related to the aggregated behavior of vehicles. Such results can guide the analyst to prioritize the identified problems, and focus on those that are considered more important than the rest.

The following are example queries that our analysts were interested in. Our framework can be used for answering many other queries as well.

Ranking by total cost.

The analyst is interested in the most expensive failure sequences. In the data we examined, there are sequences that cost the manufacturer more than $6,000. These failure sequences are important, because they cost so much to repair, and renders the vehicle warranty program expensive to maintain. If no action is taken (i.e., all manufacturing parameters remain the same), then we expect to see these failure sequences continue to happen in the future.

Ranking by the difference of part and labor cost for a specific type of failures.

In this example, we are focusing on failure sequences that involve engine failures, for which the labor cost is more expensive than the part cost. Such sequences correspond to an interesting subset of the warranty claims, those for which the part cost is very small, while the labor cost is much higher (around $1,200). Knowledge of these sequences can be useful when redesigning an engine, so as to make sure that the labor cost for repairing the same kind of failures is reduced (e.g., by making particular parts of the engine more easily accessible).

Ranking by frequency of occurrence for a specific engine model.

This query lists the most frequent failure sequences, for which the engine model most commonly involved in the sequences is "A". These sequences reveal the most frequent recurring problems related to a specific engine model. Our experiments with the real data show that more than 2,300 vehicles that are mounted with the specific engine model exhibit the same problems. Such results help identify potential problems in the design or the manufacturing of the particular model of engine.

Ranking by largest number of failures with a specific cause code.

In this example, we are interested in failure sequences that involve multiple failures, whose most common cause code is rust or corrosion. These sequences are important, because they reveal problems attributed to a specific cause. In the data we examined, we found that many vehicles had as many as 13 different parts failing in the same year due to rust or corrosion. Results from this kind of analysis may lead to a new design for particular parts, or to a change in the materials used for the manufacturing of those parts. In both cases, the goal is to increase their life-span.

2) Focused Reports

We now turn our attention to some specific failure sequences. In the following examples, we concentrate on some frequent failure sequences of interest, and we analyze in detail the characteristics of the vehicles that support these failure sequences. These are sequences that may be important to the analyst, because they involve problems on some parts that are under close examination, or because they appear too often, or just because of their sheer cost.

In all cases, the detailed statistics reveal useful information about the vehicles that exhibit these problems, such as the predominant vehicle or engine model, the plant and year where most of these vehicles were manufactured, the main cause of the problems, and others.

It is important to note that the following are just two indicative examples from the many that we used during our case study. The proposed approach enables the analysts to focus on any aspect that is deemed important for their analysis.

Failures in brakes and electrical components.

The first example focuses on vehicles that failed two different times within the same year, for problems related to the brakes and the electrical components. In Table 2, we list the most common causes for each of the failures in the sequence. The results show that in the majority of the cases the failure cause is the same, which indicates that a probable root cause for the problems may be in the design or the manufacturing of the failed parts.

failure X cause code	%	failure Y cause code	%
inoperative	72	leaking	64
shorted	13	rubs	9
leaking	7	broken	4

Table 2. Example 1: Cause code break-down. (All the code ids and descriptions have been replaced by dummy values for reasons of anonymity.)

In Table 3, we can see yet another view of the same data. This table shows the years during which most of the vehicles that exhibit the specific failure sequence were manufactured, their model, the factory where they were manufactured, and the type of engine they have. We observe that many more of the failed vehicles were manufactured in year 2003, than in year 2002. The same is true when we compare factory *A1* to factory *A3*. These results can help the analyst understand the root cause of the problems, and take corrective actions.

bld_dte	%	model	%	plant	%	engine	%
2003	53	M1	17	P1	39	E1	21
2004	29	M2	13	P2	37	E2	19
2002	18	M3	12	P3	16	E3	14

Table 3. Example 1: Demographics break-down. (All the code ids and descriptions have been replaced by dummy values for reasons of anonymity.)

Failures in driving axle, wheels, and brakes.

In the second example, we focus on vehicles that failed three different times during the same year, for problems related to the driving rear axle, the wheels, and the brakes. Observe that in this case all three problems relate to the same sub-system of the vehicle, and that they have occurred one after the other. By examining the causes of these failures (refer to Table 4), it is evident that the main problem is leaking parts. Moreover, we observe that all the vehicles that exhibited those failures were manufactured during 2004, and the vast majority of those, almost 90%, by the same factory (refer to Table 5). This kind of results offer a clear indication to the analyst as to where to focus the efforts necessary for resolving the problems in the vehicles.

failure X cause code	%	failure Y cause code	%	failure Z cause code	%
leaking	100	leaking	47	leaking	21
		loose	5	broken	11
				loose	11

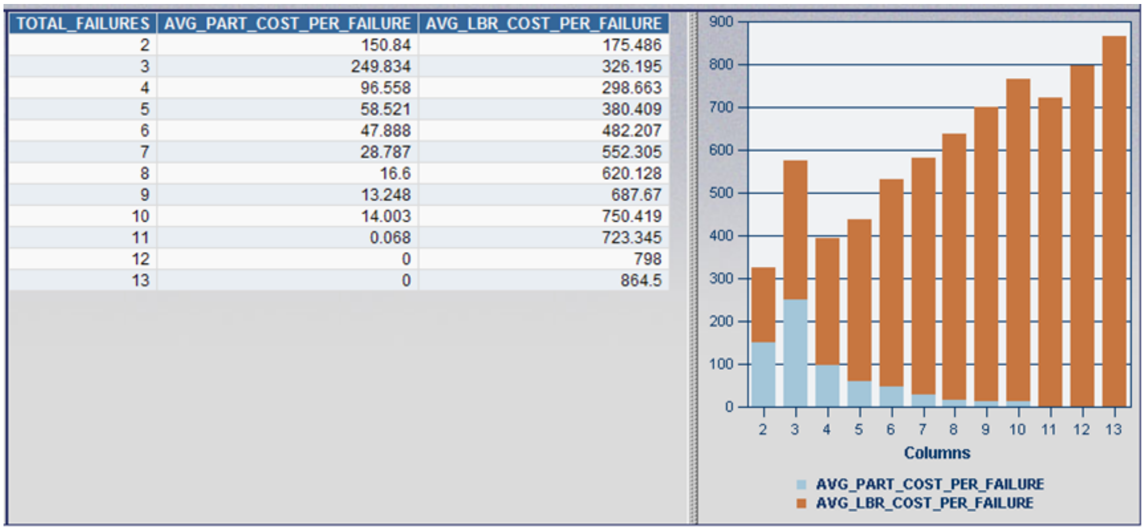
Table 4. Example 2: Cause code break-down. (All the code ids and descriptions have been replaced by dummy values for reasons of anonymity.)

bld_dte	%	model	%	plant	%	engine	%
2004	100	M1	74	P1	89	E1	79
		M2	21	P2	5	E2	16
		M3	5	P3	5	E3	5

Table 5. Example 2: Demographics break-down. (All the code ids and descriptions have been replaced by dummy values for reasons of anonymity.)

3) User Interface

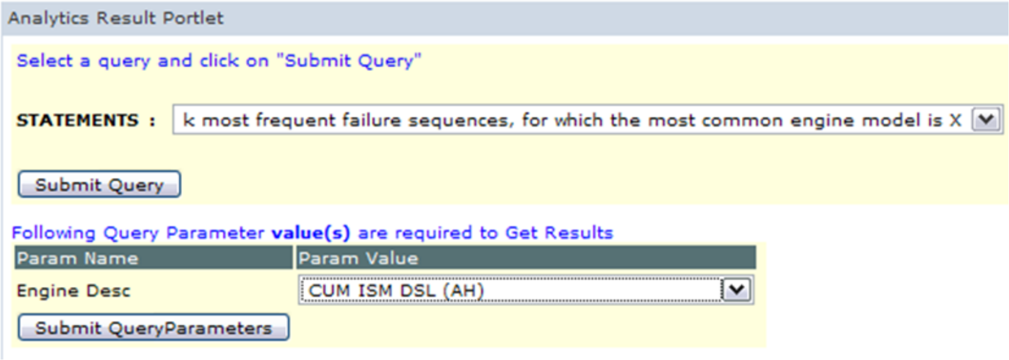
We now present examples from the graphical user interface we have developed. In our case study, we used the Alphablox e-business analysis application (DB2 Alphablox 2011), but any data analytics and reporting platform can be used instead.
In Figure 3 we illustrate the distribution of the average total cost (i.e., replacement part cost plus labor cost) for repairing the failures in the failure sequence (y-axis), as a function of the number of failures in the sequence (x-axis). Remember that each failure sequence refers to a single vehicle. This means that there are several vehicles that repeatedly failed (up to thirteen times) during the time period represented by our data, which is approximately one year. What is interesting to note in this graph is that it reveals a trend regarding the cost for fixing those failures. When there are a few failures, the total cost is split between the part and the labor costs. However, as the number of failures grows, the labor cost dominates the total cost of fixing these failures. The reason for this trend is that the parts per se are in good working condition (so there are little or no replacement part costs), yet, some human error (perhaps during an earlier visit to the mechanic) is causing the vehicle to malfunction.



TOTAL_FAILURES	AVG_PART_COST_PER_FAILURE	AVG_LBR_COST_PER_FAILURE
2	150.84	175.486
3	249.834	326.195
4	96.558	298.663
5	58.521	380.409
6	47.888	482.207
7	28.787	552.305
8	16.6	620.128
9	13.248	687.67
10	14.003	750.419
11	0.068	723.345
12	0	798
13	0	864.5

Figure 3. Average labor and part cost vs total number of failures. (All the code ids and descriptions have been replaced by dummy values for reasons of anonymity.)

Figure 4 depicts two screen-captures that show the process of selecting a particular view for displaying the results of the frequent failure sequences analysis. In this particular example, the analyst is interested in getting information about failures that involve a specific engine type. Therefore, in the picture on the top (Figure 4(a)), the analyst is selecting a view that will display results for engine type “E”. In the next picture (Figure 4(b)), the system is presenting the analyst with detailed information about the exact sequences of failures that are occurring most often for the engine type “E”. As we can see, these sequences of failures have occurred to thousands of vehicles (see second column) within a single year.



Analytics Result Portlet

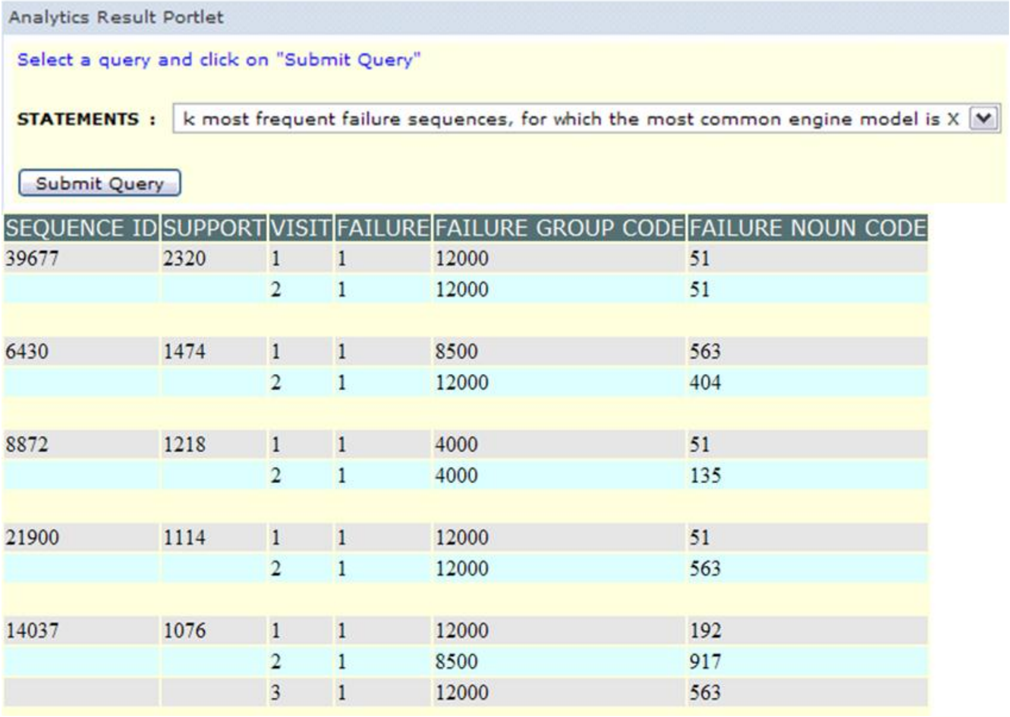
Select a query and click on "Submit Query"

STATEMENTS : k most frequent failure sequences, for which the most common engine model is X

Following Query Parameter **value(s)** are required to Get Results

Param Name	Param Value
Engine Desc	CUM ISM DSL (AH)

(a) view selection



Select a query and click on "Submit Query"

STATEMENTS : k most frequent failure sequences, for which the most common engine model is X

SEQUENCE ID	SUPPORT	VISIT	FAILURE	FAILURE GROUP CODE	FAILURE NOUN CODE
39677	2320	1	1	12000	51
		2	1	12000	51
6430	1474	1	1	8500	563
		2	1	12000	404
8872	1218	1	1	4000	51
		2	1	4000	135
21900	1114	1	1	12000	51
		2	1	12000	563
14037	1076	1	1	12000	192
		2	1	8500	917
		3	1	12000	563

(b) sequence details

Figure 4. Output showing a particular view of the frequent failure sequence details. (All the code ids and descriptions have been replaced by dummy values for reasons of anonymity.)

C. Discussion

As discussed earlier, the analyst can select one of the frequent failure sequences, and get even more detailed information on the vehicles that exhibit that particular failure sequence. For example, the analyst can see a break down of the cause of the failures, the year the vehicles were manufactured, the factory they were manufactured, and others.

When following the process we describe in this work, the analyst can efficiently and effectively focus on the most relevant and important problems. The analyst is able to view, prioritize, and evaluate the corresponding information according to different criteria, such as the cost (relating to the financial aspect of the business), or the frequency of failures (relating to customer satisfaction and the marketing aspect of the business). It is for the above reasons that we claim that the proposed framework leads to actionable knowledge. Our approach makes it possible to quickly analyze huge collections of data, and provide the different interested parties in the business with focused and specific evidence as to which processes are malfunctioning and what the most probable causes may be.

The effectiveness of the proposed approach was also validated by analysts from the two vehicle manufacturing companies with whom we collaborated in our case studies. The processes they were previously following for analyzing their data could not produce results of equivalently high quality and rich content. After the adoption of our framework, they were able to cut down on the time spent on data analysis and interpretation to a small fraction of the time that was previously necessary (from more than 45 days down to a few days). In addition, our framework enabled them to perform more focused analysis and deliver reports with a higher impact factor.

Our experience with the real case studies showed that shortening the vehicle warranty resolution cycle by 10 days can save an Original Equipment Manufacturer (OEM) around $300m and reduce the total number of warranty claims by 5%. Early evidence also suggests that the results of our analysis can be useful for re-designing the problematic parts of vehicles, in order to reduce, or even avoid altogether, some of the occurring failures and sequences of failures.

Finally, we note that the proposed framework is not in competition with other data mining methodologies (e.g., CRISP-DM (Cross Industry Standard Process for Data Mining 2011)), or SEMMA (Paper 2008)), but is rather complementary to those. In particular, what we describe in this paper is a specific example of how these generic data mining methodologies could be formalized and implemented, focusing on the steps of data modeling and result presentation. (We note that these steps are relevant to the *data preparation*, *modeling*, and *evaluation* steps of CRISP-DM, and to the *modify*, *model*, and *assess* steps of

SEMMA.) Our work could also help in the direction of standardizing the interfaces of different commercial data mining tools (e.g., (DB2 Intelligent Miner 2011; IBM SPSS Modeler 2011; Microsoft SQL Server Business Intelligence 2011; Oracle Data Mining 2011; SAS Enterprise Miner 2011)) for the purpose of helping business analysts to explore the produced results along different dimensions. In our case study, we investigated such a report generation technique for a set of queries related to sequences of vehicle failures.

5. CONCLUSIONS

Data mining algorithms have been used for the past several years in various different domains, because they provide the ability to analyze the huge collections of data now available, and gain insight into the processes that generate these data. Nevertheless, these algorithms produce results that are not easy to interpret by non-expert users, and are not easy to use by business analysts.

In this study, we present a framework for enriching the results of the data mining process with additional information necessary for the business analyst. This information is relevant to different aspects of the data mining results, and enables the analyst to manipulate and interpret these results in a principled and systematic way.

We experimentally evaluated our approach with real world case studies, which demonstrated that the proposed framework has value in the e-commerce context. It offers a formulated way that helps to convert the data mining results into actionable knowledge that the business analyst can use to improve the business operations. In the context of our case studies, this translated to changes in the design and manufacturing processes in order to avoid expensive warranty claims for specific vehicle failures.

There are several research directions that we are currently pursuing. We intend to make our framework more general, by incorporating more data mining algorithms. This will give the business analyst a richer toolset for analyzing the available data, and more opportunities for discovering new knowledge. We also plan to make our framework easier to deploy, by making the entire process model-driven.

REFERENCES

Cross Industry Standard Process for Data Mining. http://www.crisp-dm.org/, 2011.

DB2 Alphablox. http://www.alphablox.com/, 2011.

DB2 Intelligent Miner. http://www-306.ibm.com/software/data/iminer/, 2011.

IBM SPSS Modeler. http://www.spss.com/software/modeling/modeler/, 2011.

Microsoft SQL Server Business Intelligence. http://www.microsoft.com/sql/solutions/bi/default.mspx, 2011.

Oracle Data Mining. http://www.oracle.com/technology/products/bi/odm/, 2011.

Orange. http://www.ailab.si/orange/, 2011.

Predictive Model Markup Language. http://www.dmg.org/pmml-v3-0.html/, 2011.

RapidMiner. http://rapid-i.com/, 2011.

SAS Enterprise Miner. http://www.sas.com/technologies/analytics/datamining/miner/, 2011.

Weka. http://www.cs.waikato.ac.nz/ml/weka/, 2011.

Aggarwal C. C., Han J., Wang J., and Yu P. S. A framework for projected clustering of high dimensional data streams. In VLDB, pages 852–863, 2004.

Agrawal R. and Srikant R. Fast Algorithms for Mining Association Rules. In VLDB, pages 487–499, Santiago de Chile, Chile, Sept. 1994.

Agrawal R. and Srikant R.. Mining Sequential Patterns. In ICDE, pages 3–14, Taipei, Taiwan, Mar. 1995.

Ayres J., Flannick J., Gehrke J., and Yiu T.. Sequential Pattern Mining Using a Bitmap Representation. In International Conference on Knowledge Discovery and Data Mining, 2002.

Chen M. and Sairamesh J.. Ranking-Based Business Information Processing. In E-Commerce Technology, 2005.

Domeniconi C., Papadopoulos D., Gunopulos D., and Ma S.. Subspace clustering of high dimensional data. In SDM, 2004.

Evfimievski A.V., Srikant R., Agrawal R., and Gehrke J.. Privacy preserving mining of association rules. Inf. Syst., 29(4):343–364, 2004.

Ganti V., Gehrke J., and Ramakrishnan R.. Mining Data Streams under Block Evolution. SIGKDD Explorations, 3(2), 2002.

Han J. and Kamber M.. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2006.

Hand D.J., Mannila H., and Smyth P.. Principles of Data Mining. MIT Press, 2000.

Keogh E., Lonardi S., and Chiu W.. Finding Surprising Patterns in a Time Series Database In Linear Time and Space. In International Conference on Knowledge Discovery and Data Mining, pages 550–556, Edmonton, Canada, July 2002.

Li Y.-H. and Sun L.-Y.. Study and Applications of Data Mining to the Structure Risk Analysis of Customs Declaration Cargo. In ICEBE, pages 761–764, 2005.

MacQueen J.. Some Methods for Classification and Analysis of Multivariate Observations. In Berkeley Symposium on Mathematical Statistics and Probability, 1967.

Manerikar N., Palpanas T.. Frequent Items in Streaming Data: An Experimental Evaluation of the State-of-the-Art. Data and Knowledge Engineering (DKE) 68(2009), 2009.

Mannila H., Toivonen H., and Verkamo A.I.. Discovery of frequent episodes in event sequences. Technical Report C-1997-15, Department of Computer Science, University of Helsinki, 1997.

Ng R.T, Lakshmanan L.V.S., Han J., and Pang A.. Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. In ACM SIGMOD International Conference, Seattle, WA, USA, June 1998.

Palpanas T.. Knowledge Discovery in Data Warehouses. ACM SIGMOD Record, 29(3):88–100, 2000.

Palpanas T., and Koudas N.. Entropy Based Approximate Querying and Exploration of Datacubes. SSDBM, Fairfax, VA, USA, July 2001.

Palpanas T., Koudas N., and Mendelzon A.. Using Datacube Aggregates for Approximate Querying and Deviation Detection. IEEE Transactions on Knowledge and Data Engineering, 17(11), 2005.

Palpanas T. and Sairamesh J.. Knowledge Mining for the Business Analyst. In DEXA, 2008.

Paper S. I. W.. From Data to Business Advantage: Data Mining, The SEMMA Methodology and the SAS System. Technical report, SAS Institute Inc., 2008.

Pednault E.. Transform Regression and the Kolmogorov Superposition Theorem. Technical Report RC-23227, IBM Research, 2004.

Sarawagi S.. User-Adaptive Exploration of Multidimensional Data. In VLDB International Conference, pages 307–316, Cairo, Egypt, Sept. 2000.

Sarawagi S., Agrawal R., and Megiddo N.. Discovery-driven Exploration of OLAP Data Cubes. In International Conference on Extending Database Technology, pages 168–182, Valencia, Spain, Mar. 1998.

Sathe G. and Sarawagi S.. Intelligent Rollups in Multidimensional OLAP Data. In VLDB International Conference, pages 531–540, Rome, Italy, Sept. 2001.

Srikant R. and Agrawal R.. Mining Sequential Patterns: Generalizations and Performance Improvements. In EDBT, pages 3–17, Avignon, France, Mar. 1996.

Tantono F.I., Manerikar N., Palpanas T.. Efficiently Discovering Recent Frequent Items in Data Streams. SSDBM, Hong Kong, China, July 2008.

Yamanishi K., Takeuchi J., Williams G.J., and Milne P.. On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms. Data Mining and Knowledge Discovery, 8(3), 2004.

Yang X., Weiying W., Hairong M., and Qingwei S.. Design and Implementation of Commerce Data Mining System Based on Rough Set Theory. In ICEBE, pages 258–265, 2005.

Themis Palpanas is a professor of computer science at the University of Trento, Italy. He received the BS degree from the National Technical University of Athens, Greece, and the MSc and PhD degrees from the University of Toronto, Canada. Before joining the University of Trento, he worked at the IBM T.J. Watson Research Center. He has also been a Visiting Professor at the National University of Singapore, worked for the University of California, Riverside, and visited Microsoft Research and the IBM Almaden Research Center. His interests include data management, data analysis, streaming algorithms, and business process management. His research solutions have been implemented in world-leading commercial data management products and he is the author of five US patents, three of which are part of commercial products in multi-billion dollar markets. He is the recipient of two Best Paper awards (ICDE 2010 and ADAPTIVE 2009). He is a founding member of the Event Processing Technical Society, and is serving on the Editorial Advisory Board of the Information Systems Journal and as an Associate Editor in the Journal of Intelligent Data Analysis. He is a General Co-Chair for VLDB 2013, has served on the program committees of several top database and data mining conferences, including SIGMOD, VLDB, ICDE, KDD, and ICDM, and has been a member of the IBM Academy of Technology Study on Event Processing

.