Sequential pattern mining algorithm for automotive warranty data

Jirachai Buddhakulsomsiri\textsuperscript{a,}\textsuperscript{*}, Armen Zakarian\textsuperscript{b}

\textsuperscript{a}School of Manufacturing Systems and Mechanical Engineering, Sirindhorn International Institute of Technology, Thammasat University, Thammasat-Rangsit, P.O. Box 22, Pathumthani 12121, Thailand

\textsuperscript{b}Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, USA

\begin{abstract}
This paper presents a sequential pattern mining algorithm that allows product and quality engineers to extract hidden knowledge from a large automotive warranty database. The algorithm uses the elementary set concept and database manipulation techniques to search for patterns or relationships among occurrences of warranty claims over time. These patterns are represented as IF--THEN sequential rules, where the IF portion of the rule includes one or more occurrences of warranty problems at one time and the THEN portion includes warranty problem(s) that occur at a later time. Once sequential patterns are generated, the algorithm uses rule strength parameters to filter out insignificant patterns, so that only important (significant) rules are reported. Significant patterns provide knowledge of one or more product failures that leads to future product fault(s). The effectiveness of the algorithm is illustrated with the warranty data mining application from the automotive industry. A discussion on the sequential patterns generated by the algorithm and their interpretation for the automotive example are also provided.
\end{abstract}

\section{Introduction}

Many industries, including the automotive industry are faced with the tasks of improving product quality and minimizing warranty costs. Product quality is by-product of the effectiveness of product development processes and their production systems. Thus, product quality can be improved through continuous improvements in product design and development of robust manufacturing and assembly systems. However, no matter how well a product is designed and manufactured, it may fail in the usage environment, either by chance or by some assignable causes. When a product fails within a certain time period, the warranty is a manufacturer’s assurance to a buyer that the product will be repaired without a cost to the customer. In a service environment where dealers are more likely to replace than to repair, the cost of component failure during the warranty period can easily equal three to ten times the supplier’s unit price (Baird 2000; Feng, Wang, & J. 2001; Cali 1993). Consequently, companies invest significant amounts of time and resources to monitor, document, and analyze product warranty data.

Product quality problems are monitored during the warranty period through the claims filed against the products. This process generates large volumes of warranty data records, such as product problems in the form of repair related labor codes, problem descriptions, actions taken, repair dates, and repair costs (labor and parts). Sequential pattern analyses of these data records may provide significant benefits to product manufacturers. A sequential pattern analysis searches for patterns or relationships between data objects in a database that occur over time. The analysis is particularly of interest to automotive Original Equipment Manufacturers (OEM), because it identifies important sequential relationships between various product faults. For example, sequential pattern analysis results may reveal a fault pattern that shows how previous product failures may have led to other product fault(s) at a later time. This knowledge enables companies to effectively predict or discover the root causes of failures that are caused by, or are associated with, the earlier problems. This helps in formulating an action plan to remedy the problems and improve product quality, which leads to significant savings in warranty costs and the attainment of product goodwill.

In this paper, a sequential pattern mining algorithm for automotive warranty data is presented. The proposed algorithm is based on the elementary set concept and database manipulation techniques. The algorithm is constructed to search for significant sequential patterns in preprocessed data sets that are obtained from a large automotive warranty database. The sequential patterns are represented in a form of IF--THEN association rules, where the IF portion of the rule includes quality/warranty problems, represented as labor codes, that occurred in an earlier period, and the THEN portion includes labor codes that occurred at a later time. Once a set of unique sequential patterns is generated, the algorithm applies a set of thresholds to evaluate the significance of
the rules and the rules that pass these thresholds are reported in the solution. The major differences of the proposed approach and those reported in the literature are presented at the end of this section.

Several association rule mining algorithms (Agrawal & Srikant 1994; Agrawal & Shafer 1996; Han & Kamber 2006) and sequential pattern mining algorithms (Agrawal & Srikant 1995; Thomas & Sarawagi 1998; Pei et al. 2004) have been reported in the literature. Agrawal and Srikant (1994) introduced an Apriori algorithm that generates significant association rules between items in a database such that support and confidence of the rules are greater than the user-specified thresholds. However, the algorithm generates a large number of candidate itemsets, whose sizes grow exponentially with the size of a database. To overcome this problem, Agrawal and Srikant (1995) introduced three different Apriori algorithms that define the problem of sequential pattern mining as finding the maximal (longest) sequences of items that have a certain user-specified minimum support. These algorithms use candidate generation technique to address the scalability related shortcomings of their previous approach. Bayardo and Agrawal (1999) proposed metrics for ranking association rules and introduced an algorithm that uses rule support and confidence for extracting best rules from the large data-sets. Pei et al. (2004) proposed the efficient PrefixSpan approach for sequential pattern mining. In PrefixSpan, the global database is projected into a set of smaller (local) databases and sequential patterns are constructed by exploring frequently occurring datasets of local databases.

Many efficient algorithms are proposed to mine sequential patterns. The differences between these algorithms are mostly related to how they improve computational time by imposing some constraints on the mining process, or in some subtle differences in how they handle the sequential mining process. For example, Yun (2008) uses weight constraints to reduce the number of unimportant patterns, Chen, Cao, Li, and Qian (2008) incorporate user-defined constraints so that the discovered knowledge better meets user needs, Massegla, Poncelet, and Tissseire (2008) introduce time constraints in early stages of the data mining process, and Chen and Huang (2008), Fiot, Laurent, and Tissseire (2007) use fuzzy set techniques and the K-means algorithm (Kuo, Chao, & Liu 2009) to achieve better computational efficiency.

Kum, Chang, and Wang (2006) proposed a new sequential pattern mining method based on multiple alignment (rather than the usual support-based approach) for mining multiple databases. Multiple databases are mined and summarized at the local level, and only the summarized patterns are used in the global mining process. Laur, Symphor, Nock, and Poncelet (2007) introduced statistical supports to maximize mining precision and improve the computational efficiency of the incremental mining process. Kum, Chang, and Wang (2007) benchmarked the effectiveness of sequential pattern mining methods by comparing a support-based sequential pattern model with an approximate pattern model based on sequence alignment. Chen and Hu (2007) introduced concepts of recency (an ability to quickly adapt to changes in a database) and compactness, which can cause reasonable time spans for discovering data patterns. They have proposed algorithms that use these concepts to adapt to the frequency of changes in discovered patterns in the database. Lin, Chen, Hao, Chueh, and Chang (2008) introduced the notion of positive and negative sequential patterns, where positive patterns include the presence of an itemset of a pattern, and negative patterns are the ones with the absence of an itemset. Ren, Sun, and Guo (2008) developed an incremental sequential pattern mining process that stores the results from the previous mining and uses them to efficiently mine the database when additional data are added.

Typically, warranty data are strictly confidential for most companies because they relate to product quality, reliability, and are therefore critical to consumers' product goodwill. As a result, literature on the warranty data analysis of real-life applications is limited to a few published reports (see Blischke & Murthy 1994; Majeske and Herrin 1995, and Lu 1998). Most models and algorithms developed in warranty analysis studies involve warranty cost analysis and can be divided into two categories: (1) one-dimensional studies, which model product failures and warranty costs as a function of the warranty period (see Blischke & Murthy 1996; Sahin & Polatoglou 1998), and (2) two-dimensional studies, which model failures and perform warranty analysis by considering both warranty period and length or frequency of usage (see Murthy, Djamaludin, & R.J. 1995; Singpurwalla & Wilson 1998; Majeske 2007). In most studies, the warranty analysis concentrates on: (a) modeling of failure patterns to estimate the number of occurrences (or recurrences) of failures (components, subassemblies, or systems) over the warranty period, assuming all the usage conditions are statistically similar and all the warranty claims are reported with no delay, (b) modeling of rectification costs incurred by failures, and (c) modeling of the expected warranty costs (see Karim, Yamamoto, & Suzuki 2001; Lawless 1998; Polatoglou & Sahin 1998; Suzuki, Yamamoto, Karim, & Wang 2000; Suzuki, Karim, & Wang 2001; Majeske 2007; Fredette & Lawless 2007; and Kulkarni & Resnick 2008). Several studies developed empirical models based on the manufacturer's field data (i.e., failures and costs over the warranty period) for the warranty cost analysis (see Robinson & McDonald 1991; Lawless & Kalbfleisch 1992; Hu & Lawless 1996). Others use probability distribution functions and statistical models for estimating warranty costs with the incomplete data (see Karim et al. 2001; Wang & Suzuki 2001). More recent studies are: Gutierrez-Pulido, Aguirre-Torres, and Christen (2006), which used a utility-function-based method to determine the appropriate warranty length of a product (brake linings), and Jung and Bai (2007), which applied a bivariate reliability model to estimate the lifetime distribution for products. A comprehensive literature review on warranty data analysis can be found in Murthy and Djamaludin (2002).

Although a number of research studies have been reported on warranty analysis, most of them use statistical approaches for cost and/or reliability analysis (Majeske, Lynch, & Herrin 1997; Kalbfleisch, Lawless, & Robinson 1991; Hu & Lawless 1996; Lawless 1998), while very few have applied data mining techniques to warranty data (Hotz et al. 1999, 2001, Buddhakulsomsiri, Siradehyan, Zakarian, & Li 2006). Hotz et al. (1999) implemented a data mining support environment for planning warranty and goodwill costs in the automotive industry. Regression analysis and back-propagation neural network were used to construct an automatic prediction tool based on the historical warranty data and goodwill costs. Hotz et al. (2001) later developed statistical and machine learning methods for detecting deviation of warranty costs and for the analysis of warranty and goodwill cost statements. Buddhakulsomsiri et al. (2006) implemented a data mining approach to explore the potential benefits of data mining in automotive warranty data analysis. Potential data mining tasks were identified, based on the type of knowledge to be mined. An association rule generation algorithm was developed for important mining tasks. The algorithm was applied to automotive warranty data to illustrate its effectiveness.

In this paper, a new data mining algorithm is presented that uses the elementary set concept of rough set theory (Pawlak 1997) with some important modifications and database manipulation techniques for identifying significant sequential patterns from a large automotive warranty database. Specifically, the algorithm considers all the possible rules that may be generated from a data set rather than the rules determined from the upper and lower approximations of rough set theory. Furthermore, the algorithm proposed in this paper uses important database set operations to
reduce computation time of the rule generation (Buddhakulsomsiri et al. 2006; Siradeghyan, Zakarian, & P. 2008). In addition, sequential mining of warranty data has some unique characteristics not encountered in typical data mining problems. Meaning, the same product problem can occur more than once in a given day, which may result in a significant number of duplicate rules during the rule generation process. The proposed algorithm introduces an important procedure (Step 2 of the proposed algorithm) that effectively combines duplicate rules and improves the algorithm’s computational efficiency. We demonstrate the effectiveness of this procedure by showing the number of rules generated by the algorithm with and without the use of this procedure. Finally, this paper presents a unique and perhaps the first data mining application to the automotive warranty problems that arise over time.

The remainder of the paper is organized as follows: Section 2 provides a discussion on the source and characteristics of automotive warranty data and the data preprocessing process used to extract necessary data attributes for the sequential pattern mining. Section 3 presents the sequential pattern mining algorithm. Section 4 presents computation results of the algorithm when applied to a larger automotive warranty data set, with a detailed discussion on sequential pattern generation and interpretation. Conclusions and future research directions are provided in Section 5.

2. Source of automotive warranty data and data preprocessing

The automotive warranty database contains vehicle attributes and warranty problem related data. Typically, automotive warranty data are obtained from: (1) manufacturing and assembly plants (e.g., vehicle identification number (VIN), production date, product options (attributes), plant ID, supplier data, and so on); (2) automobile dealerships (e.g., VIN, sales date); and (3) repair shops (e.g., repair-related labor code, repair date, mileage-at-repair, labor and part costs, and so on). These different pieces of data are stored in a warranty data warehouse maintained by a claim processing department and are used by the quality engineering experts to perform data analysis and provide recommendations for design and manufacturing improvements. Warranty data used in sequential pattern mining are product repair data representing quality problems that occur within the product warranty period. The characteristics of the repair data can be from the following four types: temporal (i.e., repair dates), numerical (i.e., mileage at repair, labor cost, part cost), categorical (i.e., labor code), and textual (e.g., problem description, repair actions taken). Table 1 shows sample repair data used in the analysis.

For the purpose of sequential pattern mining, the original warranty data is preprocessed at the outset to filter out irrelevant (non-warranty-related) data attributes and obtain a database with warranty relevant attributes that include repair related labor codes, their associated warranty costs (part and labor), and repair dates. The data are then properly arranged to facilitate the mining operations. Sample preprocessed warranty data is shown in Table 2 and explained next.

Each data object (row) in Table 2 corresponds to a vehicle with a unique VIN. Each object contains a condition labor code set, a decision labor code set, their associated costs, and inter-occurrence time (in days). The condition set includes labor codes that occur together and are recorded when the vehicle was serviced for repair, and the decision set includes labor codes for the same vehicle when it was serviced at a later time. The condition and decision costs are the total repair costs (labor and parts) to the manufacturer for the two (i.e., condition and decision) labor code sets. A repair cost for a labor code may vary from one vehicle to another, depending on the extent of damage. Inter-occurrence time is the time interval between repair dates for the condition and decision labor codes of a vehicle.

Another important parameter in preprocessed data is the inter-occurrence time limit, which specifies the maximum inter-occurrence time between two labor code sets. For example, setting the inter-occurrence time limit to one month would generate a preprocessed data table that contains records of vehicles that were repaired at least twice within a month. The value for this parameter may be specified by the user or data analyst to generate several data sets with different inter-occurrence time limits. In addition, one may see from Table 2, there are occurrences of multiple labor codes during a single vehicle service. In the situation where the same labor code occurs more than once, it is combined into a unique labor code with the corresponding costs combined into a single repair cost.

For example, the data object represented in row 1 of Table 2 can be interpreted as follows. A vehicle was first serviced for two problems, A0110 and G2200, with the corresponding repair costs of $29.47 and $41.37, respectively. Ten days later, the same vehicle was serviced for two different problems, K9325 and M2441, which incurred $44.21 and $126.35 repair costs, respectively.

3. Sequential pattern mining algorithm

The goal of the sequential pattern mining algorithm is to determine associations between two sets of labor codes that occur sequentially and frequently. Such associations provide knowledge about the temporal relationships between diverse product quality problems. The algorithm developed in this study is an extension of the association rule generation algorithm reported in Buddhakulsomsiri et al. (2006). The algorithm includes three different stages. Stage 1 uses the elementary set concept and database manipulation techniques to generate sequential relationships between the labor codes of the condition and decision sets. Each relationship is represented as an association rule that consists of IF (condition) and THEN (decision) statements. For example, the rule (IF Labor code = A0110 THEN Labor code = M0511) extracted from a 30-day inter-occurrence data set suggests that the warranty problems A0110 and M0511 may occur sequentially within a 30-day period. Stage 2 of the algorithm aggregates duplicate rules (described earlier) from Stage 1 into unique rules. Once rules are aggregated, the Stage 3 algorithm computes rule strength parameters to evaluate the significance of the rules. Rules that overcome the significance test are reported in Stage 3.

Table 1

<table>
<thead>
<tr>
<th>VIN</th>
<th>Labor code</th>
<th>Problem description</th>
<th>Action taken</th>
<th>Repair date</th>
<th>Repair cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1GPAA0001</td>
<td>A0110</td>
<td>...</td>
<td>...</td>
<td>01/01/03</td>
<td>$29.47</td>
</tr>
<tr>
<td>1GPAA0001</td>
<td>G2200</td>
<td>...</td>
<td>...</td>
<td>01/01/03</td>
<td>$41.37</td>
</tr>
<tr>
<td>1GPAA0001</td>
<td>K9325</td>
<td>...</td>
<td>...</td>
<td>01/11/03</td>
<td>$44.21</td>
</tr>
<tr>
<td>1GPAA0001</td>
<td>M2441</td>
<td>...</td>
<td>...</td>
<td>01/11/03</td>
<td>$126.35</td>
</tr>
</tbody>
</table>


3.1. Stage 1: Sequential pattern generation

Notations used in the algorithm are presented next.

Notation
- \( C = \{c_1, c_2, \ldots, c_n\} \) denotes condition labor code set
- \( D = \{d_1, d_2, \ldots, d_m\} \) denotes decision labor code set

Elementary set
- \( C_i \) denotes the set of labor codes in set \( C \) or \( D \)
- \( D_j \) denotes the set of labor codes in set \( D \)
- \( V(C_i, c_k) \) denotes value of column \( c_k \) in the elementary set \( C_i \)
- \( V(D_j, d_l) \) denotes value of column \( d_l \) in the elementary set \( D_j \)
- \( X_{ij} \) denotes intersection of elementary sets \( C_i \) and \( D_j \)
- \( |C_i|, |D_j|, |X_{ij}| \) denotes absolute support of \( C_i \), \( D_j \), and \( X_{ij} \), where absolute support is the number of data objects in the respective sets.

C-cost, D-cost
- Repair costs of a rule for condition and decision labor code sets

Steps of the sequential pattern generation algorithm

Step 1.1: Initialize condition \( C = \{c_1, c_2, \ldots, c_n\} \) and decision \( D = \{d_1, d_2, \ldots, d_m\} \) labor code sets.

Step 1.2: Compute elementary sets of \( C \) and \( D \) and intersection \( X_{ij} \) of all elementary sets of \( C \) and \( D \): \( X_{ij} = C_i \cap D_j \) for \( i = 1, \ldots, p \) and \( j = 1, \ldots, q \).

Step 1.3: For each \( X_{ij} \neq \emptyset \), generate the following rule: IF \( c_1 = V(C_i, c_1) \) AND \( \ldots \) AND \( c_n = V(C_i, c_n) \) THEN \( d_1 = V(D_j, d_1) \) AND \( \ldots \) AND \( d_m = V(D_j, d_m) \). Compute absolute support \( |C_i|, |D_j|, |X_{ij}| \) and repair costs C-cost and D-cost for each rule.

Step 1.1 initializes sets \( C \) and \( D \) that contain column indices of labor codes of preprocessed data table, where \( c_i, k = 1, \ldots, n \) and \( d_j, l = 1, \ldots, m \), are the \( n \) and \( m \) columns of labor codes in the condition and decision sets, respectively. Step 1.1 specifies the number of labor codes in the condition and decision states of the generated rules. For the automotive warranty data, the maximum number of labor codes in the condition (or decision) statements is equal to the maximum number of labor codes recorded at one time when a vehicle is serviced. Step 1.2 identifies elementary sets and their intersections. The elementary set of \( C \) (or \( D \)) includes data objects (i.e., records in the preprocessed data table) that have the same value for the labor codes in \( C \) (or \( D \)). For example, assume set \( C \) includes only one labor code column with \( p \) possible values. Then, there are \( p \) elementary sets, where each elementary set \( C_i, i = 1, \ldots, p \) contains a list of data objects with the same value of labor codes.

Similarly, if \( C \) contains \( n \) labor code columns, the number of elementary sets of \( C \) is equal to the number of unique combinations of values that can be obtained from \( n \) labor code columns (see example 1 below). In Step 1.3, for each non-empty \( X_{ij} \), a sequential rule is generated, and the rule support \( |C_i|, |D_j|, |X_{ij}| \) and repair costs (C-cost and D-cost) are computed.

The Stage 1 algorithm is somewhat similar to rough set-based algorithms (Pawlak 1999) since both use the equivalence relations to form elementary sets. The difference is in the process of generating the rules from the databases. In rough set-based algorithms, decision rules are generated by computing the lower and upper approximations of a set, whereas in the proposed approach, the rules are generated from the intersections of elementary sets using database manipulation operations. The main advantage of the proposed approach is that it only requires the database to be scanned once, which significantly reduces computation time. In addition, rules generated from the rough set approach could be disjunctive (i.e., contain both “and” and “or” Boolean expressions) and conjunctive (contain only “and” expression) while proposed algorithm only generates the conjunctive rules.

To illustrate the steps of the Sequential Pattern Generation algorithm, consider the sample data set shown in Table 3. Assume a data analyst applies the Stage 1 algorithm to generate all of the possible 2-to-1 sequential patterns between two condition labor code columns \( (c_1, c_2) \) and one decision labor code column \( d_1 \) or \( d_2 \). In Step 1.1, the condition labor code sets \( C = \{c_1, c_2\} \) and \( D = \{d_1\} \) are initialized. In Step 1.2, the elementary sets \( C_i, i = 1, 2, 3, \) and \( D_j, j = 1, 2, 3, \) are obtained and their corresponding intersections \( X_{ij} \) are calculated and shown below. Note, since 2-to-1 rules are of interest, data objects 1 and 2 are omitted from the calculations.
In Step 1.3, five rules (see Table 4) are generated from non-empty intersections of the elementary sets \(X_{ij}\) according to the values of the elementary sets \(V(c_i,c_j)\) and \(V(d_i,d_j)\).

\[
\begin{align*}
V(c_1,c_1) & = A0110, \quad V(c_2,c_1) = K9325, \quad V(c_1,c_1) = M0511 \\
V(c_1,c_2) & = M0511, \quad V(c_2,c_2) = G2200, \quad V(c_1,c_2) = G2200 \\
V(d_1,d_1) & = M0511, \quad V(d_2,d_1) = K9325, \quad V(d_3,d_1) = A0110
\end{align*}
\]

Now, assume decision labor code column \(d_2\) is used to generate the rules instead of \(d_1\). In Step 2, \(C_1, C_2,\) and \(C_3\) remain the same, while there would be only two elementary sets of \(D, D_1 = \{2, 7\}\) and \(D_2 = \{8\}\). This results in \(X_{31} = C_3 \cap D_1 = \{7\}\), \(X_{32} = C_3 \cap D_2 = \{8\}\) (note that the remaining \(X_{ij} = \emptyset\) and only two rules are generated in Step 1.3, according to the values of the elementary sets (see Table 5).

\[
\begin{align*}
V(c_1,c_1) & = M0511, \quad V(c_2,c_2) = G2200; \quad V(d_1,d_2) = M0511, \\
V(d_2,d_2) & = G2200
\end{align*}
\]

Efficiency of the Stage 1 algorithm is further improved with the use of database projection function when computing the absolute support of all the intersections of elementary sets \(X_{ij}\). For more details on database projection operation, see Garcia-Molina, Ullman, and Widom (2001), and for its detailed implementation in data mining applications, see Buddhakulsomsiri et al. (2006).

3.2. Stage 2: Identical rule aggregation

Stage 2 algorithm searches and combines identical rules obtained from the previous stage. Two rules are considered identical if they include identical values of labor codes in their respective condition and decision statements, regardless of their column indices or the order of their appearance in IF-THEN statements. Two possible causes of identical rules are: (1) the same values of labor codes in identical rules are from different labor code columns, and (2) there are at least two labor codes in condition or decision statements, and the same set of values of labor code may come from the same set of columns, but the order of appearance may be different in the statement. For example, Rule 4 and Rule 6 presented in Tables 4 and 5 are identical rules. Although the decision labor codes are from different columns, \(d_1\) for Rule 4 and \(d_2\) for Rule 6, both rules include the same labor codes. Therefore, these two rules can be aggregated and represented with a single rule as follows: If \(M0511\) and \(G2200\) THEN \(M0511\). When aggregating identical rules, the values of \(X_{ij}\), \(C\)-cost, and \(D\)-cost are always combined, whereas the values of \(|C|\) and \(|D|\) are combined so that only \(|C|\) and \(|D|\) with the same labor code values from different columns are aggregated. To describe the Stage 2 algorithm, additional notation is presented next.

Notation

- \(R\): The number of rules generated from Stage 1 algorithm
- \(C\)-key\(_{ij}\): A set containing values of labor codes in the condition statement from rule \(r\), \([V(c_i,c_j) \mid V(c_i,c_j) \neq \emptyset]\)
- \(D\)-key\(_{ij}\): A set containing values of labor codes in the decision statement from rule \(r\), \([V(d_i,d_j) \mid V(d_i,d_j) \neq \emptyset]\)
- \(C\)-col\(_{ij}\): A set containing column indices for the labor code values in the condition statement from rule \(r\), \([c_i, c_j, \ldots, c_n]\)
- \(D\)-col\(_{ij}\): A set containing column indices for the labor code values in the decision statement from rule \(r\), \([d_i, d_j, \ldots, d_m]\)
- \(|X_{ij}|\): Absolute support of rule \(r\)
- \(C\)-cost\(_{ij}\): Condition cost of rule \(r\), and decision cost of rule \(r\)
- \(D\)-cost\(_{ij}\): respectively
- \(M_X\): A set storing unique rules (after aggregation)
- \(S_X\): A set storing the corresponding \(|X_{ij}|\), \(C\)-cost\(_{ij}\), and \(D\)-cost\(_{ij}\) of each \([C\text{-key}_{ij}, D\text{-key}_{ij}]\) in \(M_X\)
- \(M_C\): A set storing \([C\text{-key}, C\text{-col}]\) of all unique rules
- \(S_C\): A set storing \([C\text{-key}, |C|]\) of all unique rules
- \(M_D\): A set storing \([D\text{-key}, D\text{-col}]\) of all unique rules
- \(S_D\): A set storing \([D\text{-key}, |D|]\) of all unique rules

Steps of identical rule aggregation algorithm

Step 2.1: Identical rule detection and \(|X_{ij}|\) aggregation

Initialize \(M_X = S_X = \{\}\) for each rule \(r = 1\) to \(R\)

Generate \([C\text{-key}_{ij}, D\text{-key}_{ij}, |X_{ij}|, C\text{-cost}_{ij}, D\text{-cost}_{ij}]\)

Check for identical rules and update \(|X_{ij}|, C\text{-cost}_{ij}, D\text{-cost}_{ij}\

If \([C\text{-key}_{ij}, D\text{-key}_{ij}]\) in \(M_X\) (identical rule is found) Update \(|X_{ij}|, C\text{-cost}_{ij}, D\text{-cost}_{ij}\

Else (a new rule is found) Add \([C\text{-key}_{ij}, D\text{-key}_{ij}]\) to \(M_X\) Add \(|X_{ij}|, C\text{-cost}_{ij}, D\text{-cost}_{ij}\) to \(S_X\)

Step 2.2: \([C\text{-key}, C\text{-col}]\) and \(|C|\) aggregation

Initialize \(M_C = S_C = \{\}\) for each rule \(r = 1\) to \(R\)

Generate two sets: \([C\text{-key}, C\text{-col}]\), \([C\text{-key}, |C|]\), and \([D\text{-key}, D\text{-col}]\), \([D\text{-key}, |D|]\) Update \(|C|\) If \([C\text{-key}_{ij}]\) in \(S_C\) AND \([C\text{-key}_{ij}, C\text{-col}_{ij}]\) \(\neq M_C\) (identical condition labor code values are found, but they are from different condition labor code columns) (Note, \([C\text{-key}_{ij}]\) \(\notin S_C\) implies that the condition labor code values for rule \(r\) have already included in \(S_C\) from previously considered rules; but \([C\text{-key}_{ij}, C\text{-col}_{ij}]\) \(\notin M_C\) indicates that the condition labor code values are from different labor code columns and therefore, \([C]\) and \(M_C\) must be updated) Update \(|C|\) in \(S_C\): \(|C| = |C| + |C|_{(r)}\) Add \([C\text{-key}_{ij}, C\text{-col}_{ij}]\) to \(M_C\) Else (\([C\text{-key}_{ij}]\) \(\notin S_C\) new condition labor code values are found) Add \([C\text{-key}_{ij}, |C|_{(r)}]\) to \(S_C\) Add \([C\text{-key}_{ij}, C\text{-col}_{ij}]\) to \(M_C\) Update \(|D|\) in \(S_D\): \(|D| = |D| + |D|_{(r)}\) Add \([D\text{-key}_{ij}, D\text{-col}_{ij}]\) to \(S_D\) Else (\([D\text{-key}_{ij}]\) \(\notin S_D\) new decision labor code values are found) Add \([D\text{-key}_{ij}, |D|_{(r)}]\) to \(S_D\) to \(M_D\)

Step 2.3: Generate unique rules from \(M_X\) with the corresponding\(|X_{ij}|, C\text{-cost}, D\text{-cost}\) from \(S_X\). Use \(C\text{-key}_{(r)}\) and \(D\text{-key}_{(r)}\) to fetch \(|C|\) from \(S_C\) and \(|D|\) from \(S_D\), respectively.

To illustrate the steps of the Stage 2 algorithm, consider the 1-to-1 rules in Table 6 generated from the data in Table 3. Note, Rules 7, 8, and 9 are identical rules and can be aggregated into Rule 10 using the Stage 2 algorithm described next.

Step 2.1 initializes \(M_X = \{\}\) and \(S_X = \{\}\) and creates sets \([[G2200, M0511], [2, 164.49, 53.21]]\) from Rule 7. Since both \(M_X\) and \(S_X\) are initially empty, the algorithm updates \(M_X = [[G2200, M0511]]\) and \(S_X = [2, 164.49, 53.21]\). The algorithm repeats this step for Rules 8...
Table 4
Five 2-to-1 sequential pattern rules obtained using $C = \{c_1, c_2\}$ and $D = \{d_1\}$.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Absolute support</th>
<th>C-cost</th>
<th>D-cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF $c_1 = A0110$ AND $c_2 = M0511$ THEN $d_1 = A0110$</td>
<td>$</td>
<td>c_1</td>
</tr>
<tr>
<td>2</td>
<td>IF $c_2 = K9325$ AND $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_2</td>
</tr>
<tr>
<td>3</td>
<td>IF $c_2 = K9325$ AND $c_2 = G2200$ THEN $d_1 = K9325$</td>
<td>$</td>
<td>c_2</td>
</tr>
<tr>
<td>4</td>
<td>IF $c_1 = M0511$ AND $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_1</td>
</tr>
<tr>
<td>5</td>
<td>IF $c_1 = M0511$ AND $c_2 = G2200$ THEN $d_1 = A0110$</td>
<td>$</td>
<td>c_1</td>
</tr>
</tbody>
</table>

Table 5
2-to-1 sequential pattern rules obtained using $C = \{c_1, c_2\}$ and $D = \{d_2\}$.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Absolute support</th>
<th>C-cost</th>
<th>D-cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>IF $c_1 = M0511$ AND $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_1</td>
</tr>
<tr>
<td>7</td>
<td>IF $c_1 = M0511$ AND $c_2 = G2200$ THEN $d_1 = G2200$</td>
<td>$</td>
<td>c_1</td>
</tr>
</tbody>
</table>

Table 6
Identical 1-to-1 rules.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Absolute support</th>
<th>C-cost</th>
<th>D-cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>IF $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_2</td>
</tr>
<tr>
<td>8</td>
<td>IF $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_2</td>
</tr>
<tr>
<td>9</td>
<td>IF $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_2</td>
</tr>
<tr>
<td>10</td>
<td>IF $c_2 = G2200$ THEN $d_1 = M0511$</td>
<td>$</td>
<td>c_2</td>
</tr>
</tbody>
</table>

and 9, but there are no new entries in $M_3$ since both rules are identical to Rule 7, while $S_3$ is updated to $S_3 = \{4, 113.45, 55.88\}$. Step 2.2 initializes $M_4 = S_4 = \{\}$. From Rule 7, it creates sets $\{G2200\}, \{4, c_1\}$, and $\{M0511, 2, d_1\}$. Then, updates both $|C|$ and $|D|$ by setting $M_4 = \{G2200, c_2\}, S_4 = \{G2200, 4\}, M_4 = \{M0511, 1, d_1\}$, and $S_4 = \{M0511, 3\}$. From Rule 8, it creates sets $\{G2200\}, \{4, c_2\}$, and $\{M0511, 2, d_2\}$. No update for $|C|$, thus $M_4$ and $S_4$ remain the same, while $|D|$ is updated by setting $M_4 = \{M0511, 1, d_1\}, \{M0511, 2, d_2\}, \{M0511, 3\}$, and $S_4 = \{M0511, 5\}$. Finally, from Rule 9, it creates sets $\{G2200\}, \{1, c_1\}$, and $\{M0511, 2, d_2\}$ and updates $|C|$ by setting $M_4 = \{G2200, c_2\}, c_2, c_1\}$, and $S_4 = \{G2200, 5\}$. Note, no update for $|D|$, thus $M_4$ and $S_4$ stay the same.

Step 2.3 creates Rule 10 (i.e., IF $G2200$ THEN $M0511$) and aggregates the appropriate values of the support and costs.

3.3. Stage 3: Rule filtering

The goal of the Stage 3 algorithm is to filter out insignificant rules so that only important rules are reported. Several rule strength parameters, proposed in various data mining studies, are considered in this stage. These parameters are used to rank the rules that are uniquely aggregated from Stage 2 algorithm. Their definitions are given next.

| $|X_1|$ | absolute support of a rule represents the number of data objects that support the rule. |
| $\text{Cost} = \text{C-cost} + \text{D-cost}$ | Total repair cost of a rule is the sum of the costs of condition and decision labor codes. |
| $\text{Sup}(C_1) = \frac{|X_1|}{N}$ | Support of a rule is the percentage of data objects that support the rule, where $N$ is the total number of data objects in the data set. |
| $\text{Sup}(D_1) = \frac{\text{ps}}{N}$ | $P$ and $Q$ represent the percentage of objects in $C_1$ and $D_1$ that correspond to a rule. |
| $P = \text{Sup}(C_1) = \text{Sup}(D_1)$ | Measures rule confidence. |

Q represents the percent of vehicles with the decision labor code(s) that previously had specific condition labor code(s).

$\text{lift} = \frac{|X_1|}{N}$ Lift or strength (Dhar & Tushilin 1993) measures the ratio of the observed number of data objects supporting the rule to the expected number of data objects that should have supported the rule if the rule were to be not significant.

$\text{ps} = |X_1| - \frac{|X_1|}{N}$. Represents the difference between the observed number of data objects supporting the rule and its expected number (Piatetsky-Shapiro 1991).

$X^2 = \frac{\text{ps}^2}{\text{Sup}(C_1) \times \text{Sup}(D_1) \times \text{Sup}(C_1) \times \text{Sup}(D_1)}$. The chi-square $X^2$ statistic with one degree of freedom measures the significance of the association (sequential pattern) between condition and decision labor codes (Buddhakulsomsiri et al. 2006).

There is no clear indication as to which parameter is a better evaluator of the rule significance. However, close examination of these parameters indicates that they measure the importance of a rule somewhat similarly. Meaning, a rule is considered more significant when the values of these parameters increase. More specifically, these parameters are monotonically increasing functions of $|X_1|$ and monotonically decreasing functions of $|C_1|$ and $|D_1|$. These parameters are used collectively in the algorithm for comparing and evaluating the significance of the rules produced in Stage 2. The Stage 3 algorithm includes two steps and presented next.

Steps of rule filtering algorithm

Step 3.1: Compute the rule strength parameters for each rule obtained from the Stage 2 algorithm.

Step 3.2: Specify the threshold values for the rule strength parameters. Report rules whose parameters are higher than the thresholds.

To illustrate the use of rule parameters and their interpretations, consider Rule 10 (from Table 6) strength parameters calculations presented below.

$|X_1| = 4, \text{Cost} = 386.69 + 129.63 = 516.32$

$\text{ Sup}(C_1) = 5/8 = 0.625$, $\text{ Sup}(D_1) = 5/8 = 0.625$

$P = 4/5 = 0.8$, $Q = 4/5 = 0.8$

$\text{lift} = \frac{(8) \times (0.8)}{5} = 1.28$, $ps = 4 - \frac{(5) \times (0.8)}{5} = 0.875$

$\text{leverage} = \frac{0.875}{8} = 0.11$, $X^2 = \frac{8 \times (0.875/8)^2}{0.625 - 0.625 - 0.375 - 0.375} = 1.74$

The above Rule 10 parameters can be interpreted as follows: From eight data objects in Table 3, four objects ($|X_1| = 4$) or 50% of all the data objects ($\text{Sup}(r) = 50\%$) support Rule 10. The total war-
The warranty cost of this rule is $516.32. Furthermore, 80% of all vehicles (\(P = 80\%\)) with the G2200 problem developed the M0511 problem within 30-day period. Also, 80% of all vehicles (\(Q = 80\%\)) with the M0511 problem had the G2200 problem within the previous 30 days. The number of objects supporting the rule is 1.28 times as many as it should be if the rule is considered to be not significant \((\text{lift} = 1.28)\). The difference between the number of supporting objects and its expected value is 0.875 objects \((ps = 0.875)\). Finally, the sequential pattern between G2200 and M0511 within the 30-day period is not statistically significant, because the \(X^2\) statistic \((1.74)\) for this rule has the \(p\)-value of 0.19.

### 4. Computational results

All three stages of the Sequential Pattern Mining algorithm have been coded in the C#.NET programming environment and the Oracle 9i database is used to organize and manipulate the automotive warranty data. The computation study presented in this section is conducted on actual automotive warranty data sets of a vehicle model that were collected over a 27-month period. A Pentium 4, 2.8 GHz, 512 Mb RAM, personal computer is used in the experiment. The warranty data are analyzed in three-month increments. That is, the algorithm is used to mine the first three months warranty data after the vehicle model was released. Then, the mining algorithm is executed again after six months of data are collected, and this process is repeated in three-month increments of data. This creates nine incremental data sets, ranging from 3-month, 6-month, and so on, to 27-month data sets. In each of these data sets, \textit{inter-occurrence time limit} parameter, which specifies interoccurrence time intervals between the condition and decision labor codes, is set at four different values: 1-month, 3-month, 6-month, and 12-month. Thus, a total of 33 data sets are created for this study. For example, applying the algorithm to 12-month incremental data with a 6-month inter-occurrence time interval would generate sequential patterns from the first 12 months of warranty data after the vehicle model was released. These sequential patterns would be generated from all warranty records where the decision labor codes occurred no more than 6 months after the occurrences of the condition labor codes for the same vehicles. Table 7 contains the number of data objects for 33 data sets after the preprocessing step is completed.

Thirty-three data sets are mined using the three-stage algorithm. To demonstrate the capability of the algorithm, three sets of results are provided in Fig. 2, including 1–1, 1–2, and 2–1 sequential rules obtained after the Stage 2 algorithm. Note, that a rule with condition and decision labor codes as lengthy as 13–13 can be generated by the algorithm, because thirteen different labor

<table>
<thead>
<tr>
<th>Incremental data (month)</th>
<th>Inter-occurrence limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-Month</td>
</tr>
<tr>
<td>3</td>
<td>1738</td>
</tr>
<tr>
<td>6</td>
<td>4869</td>
</tr>
<tr>
<td>9</td>
<td>9887</td>
</tr>
<tr>
<td>12</td>
<td>15,677</td>
</tr>
<tr>
<td>15</td>
<td>23,934</td>
</tr>
<tr>
<td>18</td>
<td>34,624</td>
</tr>
<tr>
<td>21</td>
<td>47,022</td>
</tr>
<tr>
<td>24</td>
<td>58,433</td>
</tr>
<tr>
<td>27</td>
<td>65,227</td>
</tr>
</tbody>
</table>

![Graph](image-url)
codes did occur in one warranty record for at least one vehicle during the 27-month period. However, such a rule would not be significant, since it is supported by only one data object.

The numbers of rules shown in Fig. 1 are from the Stage 2 algorithm, where all the identical rules generated in Stage 1 have already been aggregated. The effectiveness of the Stage 2 algorithm is measured with a percent reduction in the number of rules generated with the aggregation procedure. As an example, Fig. 2 illustrates the percent reduction in 1–1 rules for all 33 data sets. As expected, the identical rule aggregation procedure (Stage 2) is more effective when the number of data objects and the inter-occurrence time limit increase.

Table 8 provides summary results when threshold parameters are used to filter rules for 27-month incremental data.

In Table 8, \( |X_{ij}| \geq 100 \) (meaning at least 100 data objects support the rule) is used in combination with each of the other rule parameter thresholds to filter the rules. That is, each column \( P \), \( Q \), total cost, \( \text{lift} \), \( \text{ps} \), and \( X^2 \) statistic gives the number of 1–1 rules when each filter is applied separately and uses the condition \( |X_{ij}| \geq 100 \). The threshold value for each rule parameter is also given in Table 8. For example, there are six 1–1 rules that have \( P \geq 25\% \) and \( |X_{ij}| \geq 100 \) for 1-month inter-occurrence time limit, and there are twenty-seven 1–1 rules with the total cost of each rule at least $100,000 and \( |X_{ij}| \geq 100 \).

For the total cost, \( \text{ps} \), and \( X^2 \), four different threshold values are used, one for each inter-occurrence time limit. Note that these minimum thresholds are chosen arbitrarily for demonstration purposes and can be adjusted by the user in the real usage environment.

Examples of significant rules are provided in Fig. 3. These rules can be interpreted as follows: Rule 1 indicates that there are 633 instances where the vehicle problem described by the labor code B4890 is followed by the labor code M5200 within a 12-month time interval. This amounts to a large warranty cost of $457,678. Two conditional probability measures, \( P \) and \( Q \), indicate that 5.93% of all warranty claims that had labor code B4890 later developed a problem described by the labor code M5200, while 13.35% of all the claims with the M5200 labor code are preceded by the problem described by the labor code B4890.

For the total cost, the \( \text{ps} \), and \( X^2 \), four different threshold values are used, one for each inter-occurrence time limit. Note these minimum thresholds are chosen arbitrarily for demonstration purposes and can be adjusted by the user in the real usage environment.

Examples of significant rules are provided in Fig. 3. These rules can be interpreted as follows: Rule 1 indicates that there are 633 instances where the vehicle problem described by the labor code B4890 is followed by the labor code M5200 within a 12-month time interval. This amounts to a large warranty cost of $457,678. Two conditional probability measures, \( P \) and \( Q \), indicate that 5.93% of all warranty claims that had labor code B4890 later developed a problem described by the labor code M5200, while 13.35% of all the claims with the M5200 labor code are preceded by the problem described by the labor code B4890.
described by the labor code B4890. A large value of $X^2$ statistic suggests that this rule is significant. The \textit{lift} value indicates that the number of objects supporting the rule is 4.49 times as many as it should be if the rule is to be considered not significant. The difference between the number of supporting objects and its expected value is 493 objects ($ps$). In addition, although the data (labor codes and costs) are masked due to the confidentiality agreement with the industrial data provider, the actual description of labor codes B4890 and M5200 do not have obvious relationships. This suggests new knowledge (sequential relationship) between two warranty problems that may seem unrelated. On the other hand, Rule 2 depicts a very significant sequential pattern of a repeated warranty problem with the large cost of $246,393. This rule implies that the problem B4890 was not diagnosed or serviced correctly during the initial repair visit and recurred again within a one-year period.

Rules 3 and 4 are the examples of 1–2 rules. Aside from the rule strength parameters interpretation, examination of actual labor codes of Rule 3 indicates a significant sequential relationship between one unrelated labor code (i.e., Z5500) that precedes two highly related labor codes (i.e., N0210 and N0220). On the contrary, Rule 4 has one of the decision labor codes (B4720) related to the condition labor code (B4890). Rules 5 and 6 are the examples of 2–1 rules. Their interpretation is similar to Rules 3 and 4. Note that the rules shown in Fig. 4 are selected based on Cost filter, which is a major driver in the automotive industry.

An interesting observation can be made by examining the trend of key strength parameters of the rules. Table 9 provides Rule 1 (see Fig. 3) strength parameters for the incremental data, where the inter-occurrence time limit is set to 12-months. Fig. 4 shows three graphical plots of the data in Table 9. Fig. 4(a) shows trends of $P$ and $Q$ values as data are incrementally collected and analyzed over time. While the $P$ value trend is relatively constant over time, the $Q$ value shows increasing trend. This implies that the occurrences of B4890-related problems that led to the M5200 problems do not change noticeably over time; however, the percentage of occurrences of M5200 problems that have proceeded with the B4890 problem has increased over time. Fig. 4(b) shows the number of data objects that support the rule for each 3-month period and for the incremental data. The number of data objects that support this rule exhibit a “rapid” increase during months 18–21 (i.e., a sudden spike in the number of occurrences supporting the Rule 1). Fig. 4(c) shows the incremental cost and the cost of the rule for each 3-month period. Consistent with Fig. 4(b), the warranty cost associated with this rule during months 18–21 is the highest.

Finally, it needs to be emphasized that due to the unique nature of the proposed approach comparative computation studies
of the proposed algorithm with those available in the literature is practically most. Most of the well known data mining algorithms, such as, ID3, C4.5 (decision tree algorithms), or CMAR (Li, Han, Pei, “CMAR: Accurate, & ” ICDM 2001), CPAR (Yin and Hun 2003), that are readily available and widely used in the comparative computation studies cannot handle the sequential data mining applications described in this paper. Most of the sequential pattern mining algorithms reviewed in this manuscript are new and their pseudo codes are not readily available to perform a valid comparative analysis. This is the reason we present the computational time study of the three-stage algorithm (see Table 10) so the reader can form an opinion on its computational efficiency. Table 10 shows the computation time (in seconds) of the Sequential Pattern Mining algorithm for the automotive data mining example. The results show that the algorithm is capable of accurately analyzing large databases within reasonable amount of time (less than 12 min). From the results in Table 10 one may see that most automotive warranty data sets considered in this study can be mined within twelve minutes, with the exception of 27-month incremental data with 12-month inter-occurrence time that requires about seven hours to produce a result.

5. Conclusion

This paper presents a data mining algorithm for extracting significant sequential patterns from a large automotive warranty database. The algorithm used the elementary set concept and database manipulation techniques to search for patterns or relationships among occurrences of warranty claims over time. Significant patterns provided knowledge of one (or more) product failures that led to future product fault(s). These patterns were represented as IF–THEN sequential rules, where the IF portion of the rule included one or more occurrences of warranty problems at one time and the THEN portion included warranty problem(s) that occur at a later time. The algorithm developed in this paper consists of three stages. The Stage 1 algorithm used the elementary set concept and database manipulation operations to generate all the possible sequential patterns from the warranty database. In Stage 2, the Identical Rule Aggregation algorithm was used to identify and aggregate identical rules obtained in Stage 1. Because the size of the typical automotive warranty database is large, the total number of possible rules generated in Stage 2 of the algorithm may still be large. Therefore, once sequential patterns were generated, in Stage 3 multiple rule strength parameters were used to filter out insignificant patterns so that only the important rules were reported. The effectiveness of the algorithm was illustrated with the warranty data mining application from the automotive industry. A discussion on sequential patterns generated by the algorithm and their interpretation for the automotive example was also provided.

Finally, it needs to be emphasized that the important and challenging requirement for warranty data mining is an efficient data integration system that can combine data from different sources (i.e., assembly plants, dealerships, and repair shops) into one unified warranty data warehouse. In addition, the development of effective text mining algorithms that can extract knowledge from the textual warranty report may also significantly enhance the data gathering and analysis process. Finally, a warranty data mining decision support system should incorporate data integration, data preprocessing, and graphical display capabilities to be an effective tool in an industrial arena.

References


Table 10

<table>
<thead>
<tr>
<th>Inter-occurrence (month)</th>
<th>Incremental data (month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2, 9, 25, 29, 52, 72, 41, 104</td>
</tr>
<tr>
<td>3</td>
<td>2, 6, 41, 28, 124, 165, 255, 296</td>
</tr>
<tr>
<td>6</td>
<td>– 9, 54, 56, 210, 306, 377, 519</td>
</tr>
<tr>
<td>12</td>
<td>– –, 55, 184, 215, 371, 274, 2301, 25200</td>
</tr>
</tbody>
</table>