Control and Cybernetics

vol. **40** (2011) No. 3

Accuracy of generalized context patterns in the context based sequential patterns mining*

by

Radosław Z. Ziembiński

Poznań University of Technology, Institute of Computing Science
Pl. Marii Skłodowskiej-Curie 5, Poznań, Poland
e-mail: Radoslaw.Ziembinski@cs.put.poznan.pl

Abstract: A context pattern is a frequent subsequence mined from the context database containing set of sequences. This kind of sequential patterns and all elements inside them are described by additional sets of context attributes e.g. continuous ones. The contexts describe circumstances of transactions and sources of sequential data. These patterns can be mined by an algorithm for the context based sequential pattern mining. However, this can create large sets of patterns because all contexts related to patterns are taken from the database. The goal of the generalization method is to reduce the context pattern set by introducing a more compact and descriptive kind of patterns. This is achieved by finding clusters of similar context patterns in the mined set and transforming them to a smaller set of generalized context patterns. This process has to retain as much as possible information from the mined context patterns. This paper introduces a definition of the generalized context pattern and the related algorithm. Results from the generalization may differ as depending on the algorithm design and settings. Hence, generalized patterns may reflect frequent information from the context database differently. Thus, an accuracy measure is also proposed to evaluate the generalized patterns. This measure is used in the experiments presented. The generalized context patterns are compared to patterns mined by the basic sequential patterns mining with prediscretization of context values.

Keywords: knowledge discovery, context based sequential pattern mining, sequential context pattern clustering, pattern accuracy.

1. Introduction

The problem of the sequential pattern mining as defined in Agrawal and Sirkant (1995) introduces a task of finding all frequent subsequences from a set of sequences. The set of sequences constitutes the mined database. In this problem

*Submitted: March 2011; Accepted: August 2011.

definition each sequence is a list of elements where each element is a non-empty set of items. A subsequence must be supported by the sufficient number of sequences to be considered as a pattern. This threshold determines the minimum support value for the mined patterns. A notion of support can be compared to the pattern inclusion in the sequence from the database.

Sequential pattern mining is a popular method of knowledge retrieval with a wide area of practical applications. Complementary mining strategies implemented in algorithms such as breadth-first in AprioriAll from Agrawal and Sirkant (1995) and depth-first in PrefixSpan from Han et al. (2001) made feasible mining in sets of sequences with different frequency characteristics. This method has been successfully used to solve real-life problems in genetics, web mining, network monitoring and medical or financial data analysis. However, the basic version of this method does only allow to process nominal data. Some generalizations or extensions have been proposed in literature e.g. Agrawal and Sirkant (1996) to handle additional information.

Another extension to the basic method is the context based sequential patterns mining as formulated initially in Stefanowski and Ziembiński (2005) and discussed here. The basic sequential pattern mining problem definition does not distinguish between context and transactional information. The context information considered here should be understood as additional information, specifying elements or sequences. It is affecting the mining process by reducing its scope to subsets of data from the database. The notions of the item and the context in a real-life example may be understood through the difference between a car as an object and its current velocity or direction. Another approach utilizing the context for describing whole sequences is the multidimensional sequential patterns mining formulated in Pinto et al. (2001). It was extended in Plantevit and Choong (2005) and Plantevit and Laurent (2008).

Also, it can be noticed that the basic problem definition is cumbersome when context information is expressed by continuous attributes. Continuous attributes have properties that cannot be fully represented by a database built on sets of nominal items. In the basic method each database containing continuous data must be converted, usually by discretization before mining, to produce database compatible to the problem definition. However, the conversion introduces granularity and causes some loss of information. Thus, an option to the conversion is proposed in this paper. Alternatively, context databases can be mined without prediscretization by the context based sequential pattern mining method.

The way of continuous data handling causes that results from the mining may differ even for the same context database and a requested value of the minimal support threshold. Therefore, it is necessary to introduce the notion of accuracy to describe differences in algorithm results. The accuracy will be understood as a degree of representativeness of frequent information existing in the context database by mined patterns. The notion of accuracy forms a new dimension in a comparison of different mining methods for context databases.

This paper begins from a short introduction to the problem of the context based sequential patterns mining. It contains references to related works. Then, the definition of the generalized context pattern is introduced. It is followed by the proposal of an example generalization algorithm. The following section introduces the accuracy measure used in experiments. Results from these experiments validate assumptions taken for the generalization method. Experiments are discussed with some conclusions inferred in the final sections of this paper.

2. Context based sequential pattern mining and the mining algorithm

The definition of the context based sequential pattern mining introduces many changes to the method known from Agrawal and Sirkant (1995). The detailed definition can be found in Stefanowski and Ziembiński (2005). Some algorithms compatible with the definition were proposed in Ziembiński (2007), where some evaluation results of respective computational costs can be found. Then, Stefanowski and Ziembiński (2009) describe an experimental evaluation of the context method accuracy. There, the accuracy was measured in reference to the basic approach with prediscretization of continuous context attribute values. This paper follows that study by introducing a more sophisticated method of context pattern post-processing. The proposed method produces less patterns, but they are more descriptive. They retain much from original context pattern information.

The definition of the context based sequential pattern mining changes data structures and alters the mining process. However, these changes have been introduced in a non-invasive way to ensure maximum compatibility with the basic method. Thus, the sequential pattern mining problem is a boundary case of the context approach.

The first modification involves an introduction of two sets of context attributes describing sequences and elements. However, the set describing sequences is different from the set describing elements. The sequence context is denoted $D = \{d_1, d_2, ...d_v\}$ and the element context is $C = \{c_1, c_2, ..., c_w\}$. The context bound to sequence may be used to store information on patient profile data like age, living place or weight. The element context may be useful if detailed information on patient's current state or drug prescriptions have to be stored. In the following description some Greek symbols are used in the superscript to label object instances.

A comparison of elements with contexts involves their evaluation with similarity functions. A two step procedure leads to the evaluation of a context pair. At the beginning, customized similarity functions related to particular context attributes evaluate pairs of attribute values. Then, a single value from the set of partial evaluations obtained in the previous step is calculated with a similarity aggregation function. Similarity functions dedicated to particular context attributes are denoted $\sigma_k^D(d_k^\alpha, d_k^\beta)$, $k = 1...v$, for the sequence contexts

and $\sigma_k^C(c_k^\alpha, c_k^\beta),\ k = 1...w$, for the element context. These functions return values ranging from 0.0 to 1.0. Attribute values are considered as identical if the similarity function returns 1.0. In the opposite case the result is 0.0. Similarity aggregation functions are denoted $\Theta^D(D^\alpha, D^\beta)$ and $\Theta^C(C^\alpha, C^\beta)$, respectively, for two kinds of contexts. They also return values between 1.0 and 0.0 with the same interpretation. Similarity functions do not have to possess properties of the measure and they even do not have to be symmetric. The similarity evaluation results are continuous values difficult in processing during mining. Thus, a discrete answer on context similarity is required. Hence, two related minimal similarity thresholds have been added for both contexts. If $\Theta^D(D^\alpha, D^\beta) > \theta_{min}^D$ then the pair of sequence contexts is considered as similar. The same is true for the pair of elements' contexts when $\Theta^C(C^\alpha, C^\beta) > \theta_{min}^C$.

The context database has a more complex structure than the basic counterpart. An element with context information has the definition $ce = \langle C, X\rangle$, where X is item set. The element ce^α is supported by the element ce^β in the context approach if $X^\alpha \subseteq X^\beta$ and $\Theta^C(C^\alpha, C^\beta) > \theta_{min}^C$. Sequences are lists of elements and have the definition $cs = \langle D, \{ce_1, ce_2, ..., ce_l\}\rangle$, where l is sequence length. The context database DB contains a finite set of sequences of known lengths. As the pattern is a subsequence of a sequence from the database its definition is identical to the sequence definition. However, the pattern will be denoted as cp to differentiate them. The pattern cp^α is supported by a sequence from the database cs^β if $\Theta^D(D^\alpha, D^\beta) > \theta_{min}^D$ and elements of cp^α are supported by some elements from cs^β in the same way as defined in the basic approach but with an assumption that element contexts are compared as described above. Finally, the subsequence cp^α becomes the pattern if its support is equal or greater than the minimal support threshold $support_{DB}(cp^\alpha) \geq support_{min}$. The goal of the context based sequential pattern mining is to find all context patterns in the DB.

According to the definition, algorithms for context pattern mining named ContextMapping and ContextApriori were proposed in Ziembiński (2007). Experiments conducted there proved that ContextApriori is inefficient. However, computational efficiency of ContextMapping is sufficient for practical applications. ContextMapping is the depth-first algorithm that performs a conversion of the context database to the equivalent database compatible with the basic database structure. It creates a new database containing results from comparisons between contexts. Thus, it does not convert continuous data describing contexts prior to the mining but stores relations between contexts. The context comparison result is binary information and may be mined in a similar way to any other nominal data. However, some modifications had to be introduced in ContextMapping to handle database obtained from conversion efficiently. They make the algorithm different from other algorithms used in the basic approach. The mining using ContextMapping can be summarized as follows:

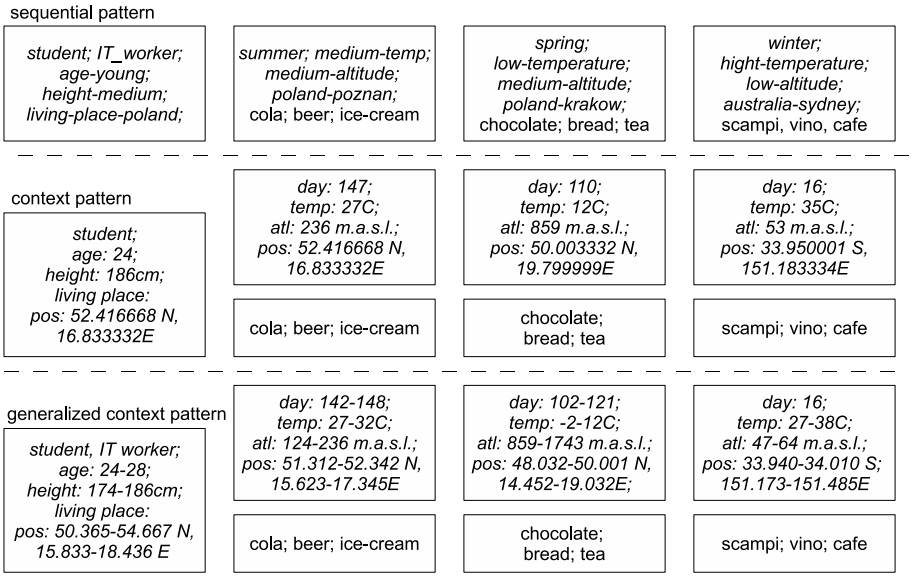

Figure 1. Examples of the non-context basic pattern, the context pattern and the generalized context pattern

1. The context database is converted to its basic equivalent. All contexts from different sequences are replaced by two related artificial items if they are similar to a currently mapped context. The first item is placed instead of the mapped context. However, the second one is put to all elements from other sequences in the mapped database whose equivalent elements in the context database contain contexts similar to those replaced by the first one. Thus, the second item represents similarity of the context pair. The mapping procedure repeats for all element and sequence contexts. However, sequence contexts are mapped to additional artificial elements and related items in the procedure independent from the element contexts processing. Additionally, in this step all infrequent contexts are omitted in the mapping and artificial items are not created for them.
2. All frequent item sets are mapped to artificial items representing them in the mapped database. This step ensures that infrequent item sets are removed from the mapped database and do not undergo further processing.
3. Elements are removed from the mapped database if they do not contain item mapping their frequent context or items representing similarity of contexts or items mapping frequent item sets. Moreover, a whole sequence has to be removed if this happens for the element and items representing sequence context and related similarity mapping. This step saves resources by ensuring that infrequent information is not stored and unnecessarily processed during the mining.

4. The depth-first algorithm mines patterns. During the mining, all items representing similarity of contexts support asymmetrically related items representing contexts. In the output set, all mined patterns have artificial elements with items representing their sequence contexts. Moreover, all elements from each pattern must have a pair of items. The first from them maps the frequent context and the other one maps its frequent item set.
5. All mappings in mined patterns are reverted. Thus, all artificial items in mined patterns are replaced by original contexts and item sets. The mined set of context patterns undergoes maximization where shorter patterns included in the longer ones are removed. Algorithm finishes the processing.

3. Generalized context patterns

The context based sequential pattern mining method tends to enumerate all frequent and distinct subsequences. Thus, it enumerates all contexts in different patterns even if sequences of item sets in these patterns are identical. It significantly increases size of the pattern set for large databases. Although full set of context patterns is an accurate representation of frequent information from the database, the overall legibility may be considered poor because of its impractical size.

The traditional approach suggests use of discretization to process context data before mining. It is possible to convert such context information to a nominal representation. Thus, continuous context information can be represented by one or more artificial items representing discretization folds embracing many similar context values. So, the mined set of patterns may be significantly smaller as the context information is processed and represented as discretization folds. Additionally, the basic mining method described in Agrawal and Sirkant (1995) can be applied to mine patterns without modifications. The prediscretization can be done efficiently with a clustering algorithm. There are many algorithms for clustering proposed in literature, see, e.g., Ng and Han (2002), Guha et al. (2000) or Yang et al. (2003). Recently proposed grid-based algorithms are fast and produce accurate clusters, e.g. Pilevar and Sukumar (2005).

A solution to the problem of the set size can be pattern clustering after mining. It maps information from a huge set of context patterns to a smaller set of generalized context patterns. This method should find all mutually similar subsets of pattern set and define clusters containing them. Then, each cluster of patterns can be compacted and treated as a single generalized context pattern. The definition of the generalized context pattern extends the definition of the context pattern. The generalized context element has a form $ge = \langle pf_C(C^\alpha, C^\beta, C^\gamma, ...), X\rangle$ and the generalized context pattern is $gp = \langle pf_D(D^\alpha, D^\beta, D^\gamma, ...), \{ge_1, ge_2, ..., ge_l\}\rangle$, where pf is a presentation function. The presentation function converts a set of context attribute values to simple and readable form, e.g. some descriptive statistics. The presentation function output may be delivered in the form of value ranges derived from the contexts.

This concept may be considered similar to fold borders in the basic approach. Context pattern clustering and presentation can be realized by different methods. Thus, the content of a generalized context pattern set would depend on the choice of the generalization algorithm and the presentation function. The Fig. 1 contains some examples of different kinds of patterns.

Context pattern generalization causes loss of information. Precise information on context values is distorted after the transformation by the presentation functions. The distortion can be minimized if generalization avoids merging elements with different item sets. Information in item set is related to the transaction itself and is usually more crucial than the context representing transaction circumstances. The proposed generalization method obeys this assumption and two patterns can be merged only if matching item sets in correspondent elements are identical. This method does not exclude merging of a shorter pattern with a longer one. In such cases it must be verified if the shorter pattern matches the longer one in different ways. The generalization method has to account for all these matches during the clustering to improve the accuracy. It is also possible to use minimal similarity thresholds to exclude the compacting of patterns with insufficiently similar contexts.

An experimental study of the mined set accuracy has been done in Stefanowski and Ziembiński (2009). Methods evaluated there were the context based sequential patterns mining and the basic sequential pattern mining with prediscretization. Prediscretization was done by two different methods. The first one was with fold splits of equal widths and the other one was with split ensuring equal frequencies of context occurrences inside folds. Experimental study showed that the first one gave more accurate result.

4. Generalization algorithm

The proposed generalization algorithm may be seen as an example of many possible solutions of the problem. Input data for the algorithm is the set of context patterns and the settings used in mining like thresholds and similarity functions. Output is a set of generalized context patterns approximating the input set of patterns.

In the first step of the algorithm the input set of context patterns is sorted in accordance to pattern lengths and supports. Patterns of the same lengths are sorted as to their support values. Both sorts are decreasing. This sorting step is meant to improve algorithm efficiency. The clustering procedure begins the processing from longer patterns and allows to accumulate shorter patterns in the accretion process. Thus, the sorting solves issues related to pattern order in the clustering.

After sorting, the algorithm creates assignment matrix for all pairs of patterns from the mined set. Each cell of this matrix contains assignment list. The list stores all possible associations between two considered patterns. All associations must fulfill constraints related to the distortion of information mentioned

in the previous section. An assignment from the list is a sequence of pairs of similar elements from compared patterns preserving element order in these patterns. A shorter context pattern can match a longer pattern in many different ways. Thus, an assignment list may have zero or more assignments. This is exemplified by a context patterns list of Fig. 2, where pattern 3 is similar to some parts of pattern 1. Because they do not violate constraints the appropriate list of assignments is created in row 3 and column 1 of the matrix.

In the following step of the algorithm a recursive procedure builds clusters of patterns. The procedure begins from the longest pattern with the greatest support. It seeks relevant assignments in subsequent patterns from the related assignment lists. This approach assumes that considering longer patterns at the beginning gives greater chance to accumulate shorter and similar patterns around them. The procedure reviews the entire assignment matrix. Thus, all assignments matching the longer pattern from the matrix are found. They form a cluster that will undergo conversion to a single generalized context pattern.

All assignments moved to the cluster are removed from the matrix and they do not play a role in subsequent clustering. This step eliminates the possibility of overlapping ranges of the generalized context pattern context values. A pattern already associated to some other pattern is labeled as the visited one and will not create its own cluster in future iterations. If the recursive accretion of assignments has been finished for a given pattern then the next iteration begins. A subsequent yet unvisited pattern is chosen from the sorted list and the accretion procedure repeats until no unvisited pattern can be found on the list. The algorithm is illustrated in Fig. 3.

Clearly, the proposed clustering algorithm forms clusters around seeds from longer patterns with higher support. The longer patterns can be considered as recommended in the output set because they are often the most valuable patterns. Then, it must be noted that the step of building of the assignment matrix can be expensive. It involves detection of all possible matches between each pair of context patterns. However, all matches have to be accounted for to minimize accuracy loss during generalization. Thus, the trade-off between processing costs and output quality privileges accuracy over performance. A different approach, consisting in clustering on the basis of common subsequences was described in Morzy et al. (1999) or Yang and Wang (2003).

5. Evaluation method of the pattern set accuracy

Evaluation with computational efficiency of the context mining algorithms is not relevant for this study. This is true because mined sets of patterns have different qualities in a term of accurate representation of frequent information from the database. Thus, another method has been proposed in Stefanowski and Ziembiński (2009) to solve the issue. It aims at evaluating similarity between two sets of context patterns. Hence, it is useful in cases where context information in patterns is represented as a set of context attribute values or a set

The context patterns list	
1	<1,1><<5,6> (a,b); <8,7> (c); <6,7> (a,b); <1,5> (a, c); <7,4> (c)>
2	<5,4><<4,7>(a,b); <9,8>(e); <1,1>(b)>
3	<2,1><<5,7> (a,b); <9,7> (c)>

The assignment matrix			
	1	2	3
1	-	-	-
2	-	-	-
3	<2,1><<5,7> (a,b); <9,7> (c); *; *; *> <2,1><<5,7> (a,b); * ; *; *; <9,7> (c)> <2,1><; *; *; <5,7> (a,b); *; <9,7> (c)>	-	-

Figure 2. The sorted list of context patterns and the produced assignment matrix

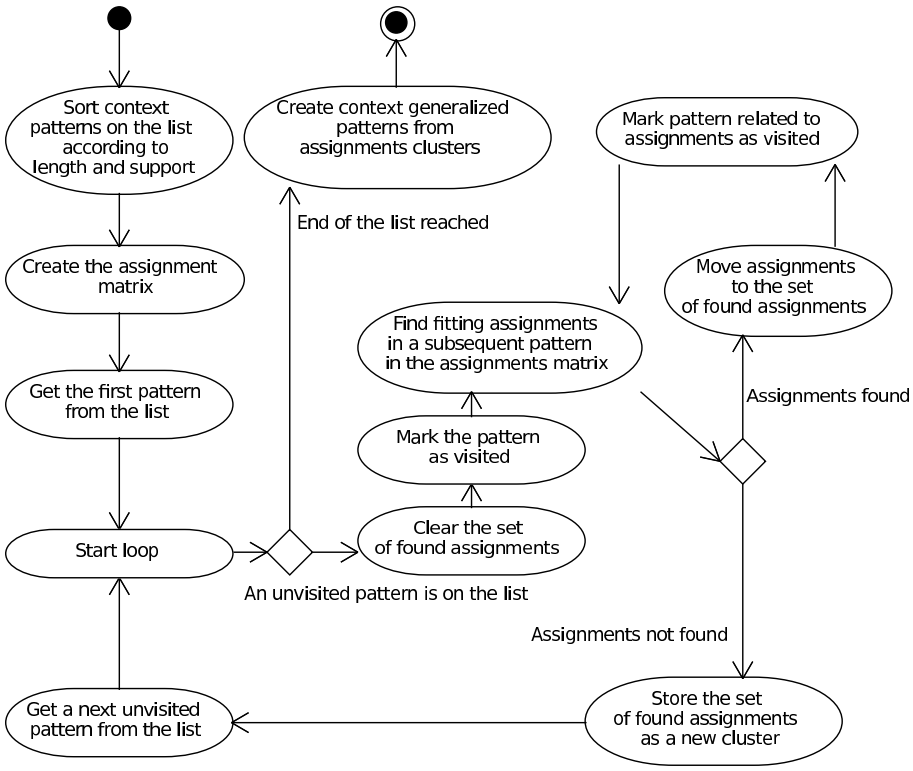

Figure 3. The activity diagram for the clustering algorithm

of discretization folds. The proposed method compares mined patterns to a reference set of context patterns existing in the context database. In experiments discussed later a known set of reference patterns is hidden in the database. The result is a single value between 0 and 1. It reflects accuracy of the reference patterns' description by mined patterns.

The method described in Stefanowski and Ziembiński (2009) compares all pairs of patterns from both sets. It uses a complex equation evaluating differences in values and structures of compared context patterns. In the first step the method finds all possible matches of pairs of elements. If the elements of a pattern match elements of the counterpart then a sequence of pairs of elements called comparison core is created . The core must preserve the order of elements in both patterns. However, there may be many comparison cores if lengths of compared patterns differ. This happens if a shorter pattern matches different layouts of elements of a longer pattern. Hence, this method creates a list of comparison cores similar to the assignment list in pattern clustering. After the matching step, each core from the list is evaluated and its similarity value is calculated from its content. Average evaluation from all comparison cores is treated as similarity value of two compared patterns.

Details of the introduced method for the core evaluation are as follows. The core similarity value is a product of values representing difference in lengths of patterns, similarity of sequence contexts and averaged sum of similarities in pairs of elements associated to the core. The value reflecting the difference in lengths is calculated as the ratio of the length of the comparison core to the length of the longer pattern. The similarity between a pair of elements is a product of values representing similarity between element contexts and similarity of item sets. Hence, the similarity between contexts is calculated using similarity functions chosen to mine these patterns from the database. However, the similarity of item sets is calculated as Jaccard coefficient. A different approach has to be used to compare intervals to a single value. If border values of a fold are compared to a single context value then the average similarity value is calculated between them and the context value. An experimental evaluation of this approach shows that it gives a small error for triangle similarity functions. However, in the case of more complex similarity functions a more precise numeric integration of similarity values could be used. All similarity evaluation values are in the normalized range from 0.0 to 1.0. The value of 0.0 means that compared objects are dissimilar. The value if 1.0 is obtained only for identical objects.

The comparison result for all patterns from both sets is stored in a similarity matrix. The matrix rows represent the reference patterns and columns reflect the mined patterns. A final value representing similarity of two sets of context patterns is an aggregate calculated from the similarity matrix. Two methods of matrix aggregation called reconstruction measure and average similarity measure are given in Stefanowski and Ziembiński (2009). The reconstruction measure averages similarity values of pairs representing the best match of mined patterns to reference patterns. It assigns mined patterns to related reference

patterns only if they are the most similar ones from the whole set of mined patterns. Then similarity evaluation values for selected pairs are averaged. The average similarity measure calculation is simpler and it is the average value of the similarity matrix. Thus, the average similarity value can be equal to 1.0 only if both sets contain the same context pattern.

6. Possible errors in prediscretization

There are some theoretical reasons for the context approach to be more accurate than the basic approach with prediscretization. The first one is related to the way of representation of context information. Context patterns contain exact attribute values from the database while the basic approach can mine patterns with context information expressed as artificial items representing prediscretization folds. Thus, the internal distribution of context values is ignored inside a discretization fold, leading to some errors after mining, related to this simplification. Reasons of these errors are as follows:

- Prediscretization can disperse clusters of context attribute values from the database. Many fast discretization methods create folds that are not optimal as they cannot distinguish clumps of values appropriately and describe them as different clusters, leading to cases where support is shared between neighboring folds. Mining at lower values of the minimal support threshold usually does not help in these cases. It might produce sets containing some random "noisy" elements with low support in the database. Patterns containing such "noise" can be eliminated only if the minimal support threshold is set to a higher value. This case is illustrated in the Fig. 4(A).
- A context with a low support may have its support increased after prediscretization if assigned to a fold with a higher support. As a result, an infrequent element with such context might be mined. The case is shown in Fig. 4(C).
- A context with a high support may have its support decreased after prediscretization if assigned to a fold with a lower support. Then, a frequent element with such context might not be mined. The case is shown in Fig. 4(B).
- A prediscretization algorithm has no knowledge if there are overlapping sub-clusters supporting different elements located in different patterns (or at different locations in the same pattern) until the mining finishes. Prediscretization recognizes such sub-clusters as a single entity forming a single, wider fold, capturing all values regardless of their sequential position in mined patterns. The case is presented in Fig. 5. This indiscernibility leads to a lower discretization accuracy because discretization folds are wider than they could theoretically be. This kind of error is unavoidable if discretization of the context attribute values is done before mining. However, the context approach is immune on this error. It can correctly

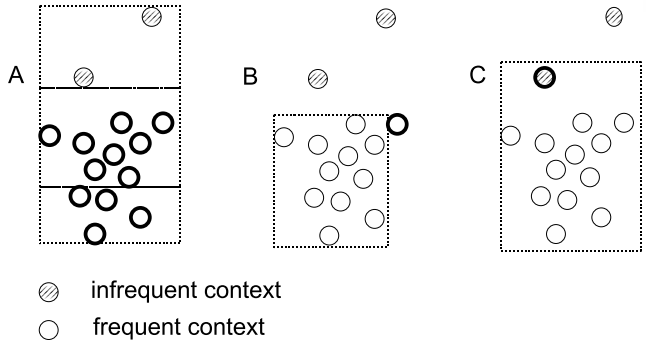

Figure 4. Some basic errors of prediscretization: A – inexact folds, B – omitted frequent context and C – included infrequent context (contexts in question are distinguished with thick lines)

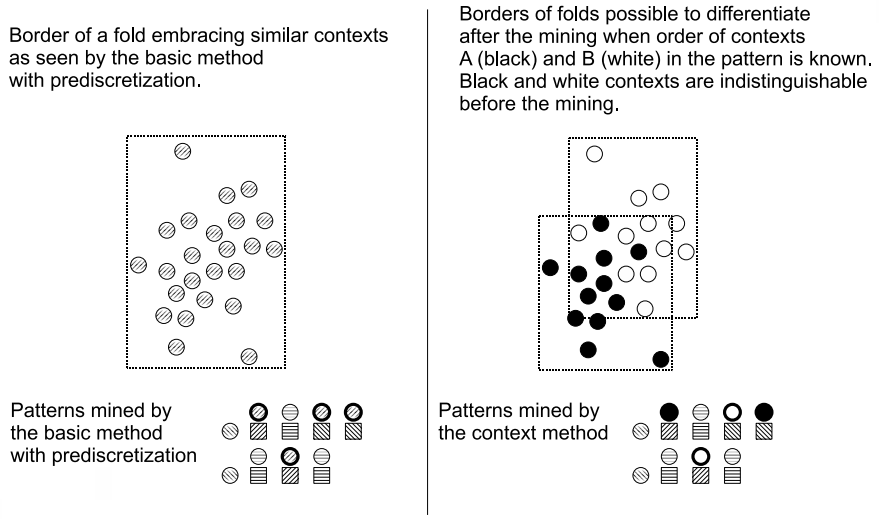

Figure 5. Example of an error in prediscretization resulting from an unknown order of contexts in patterns prior to the mining

identify such sub-clusters with correct borders because it does not use prediscretization.

- The context approach can use different similarity functions (even some non-trivial ones, e.g. rich in knowledge about attribute domains) whereas the basic approach has to rely on the item set inclusion operator. Inclusion may not be sufficient if one wished use a more complex comparison method. Thus, simple replacement of similarity functions by the inclusion operator may lead to further errors.

The experiments described in Stefanowski and Ziembiński (2009) showed the advantage of the context approach in terms of accuracy over the basic approach with a simple prediscretization.

7. Experiments and obtained results

The goal of experiments presented here was to compare the accuracy of generalized context pattern sets with original context pattern sets and to sets of patterns mined using the basic method with the prediscretization. Experiments relied on generated databases created according to the following assumptions:

- The database has a predetermined number of sequences with known lengths.
- The sequence and element contexts have two attributes represented by real numbers. Values from sequence and element contexts are generated independently.
- The generated database contains a hidden set of known context patterns (reference patterns). These patterns have specific item sets and context values. Context values describing patterns stored in particular sequences are generated by random distortion with Gaussian distribution. Actually, context seed values assigned to the reference pattern templates are distorted to retrieve context values stored in hidden patterns. The distortion range is confined within a circle of radius 1.0.
- If a sequence should be longer than a hidden reference pattern then some additional random elements are created to extend the sequence. These elements do not have items similar to those from hidden reference patterns. Their context values are generated outside distortion regions of seed context values to avoid context values overlapping.
- For all attributes in both contexts the similarity functions are triangle functions. They are defined as follows:

$$\sigma^{C,D}(v^\alpha, v^\beta) = \begin{cases} 1.0 - |v^\alpha - v^\beta| & \Rightarrow |v^\alpha - v^\beta| < 1.0 \\ 0.0 & \Rightarrow |v^\alpha - v^\beta| \geq 1.0. \end{cases}$$

- Similarity aggregation functions are averages.

Experiments presented in this paper dealt with the mining of rather small databases. It allowed for providing results on a wide range of mining settings in

a reasonable time. Namely, the computational costs of the sequential patterns mining and the context counterpart grow exponentially in the sequence size or the average number of items in sequences.

Methods used in the mining are the context based sequential patterns mining and the basic sequential pattern mining algorithm with prediscretization based on folds of equal widths. The prediscretization produces folds of equal width because Stefanowski and Ziembiński (2009) have shown that this method is more accurate than creation of folds of equal frequencies. The mined sets of context patterns are generalized with the algorithm described in the previous section. The results are evaluated by accuracy with respect to reference patterns. The presentation function for the generalization converts sets of context attribute values to ranges. As ranges are defined by minimum and maximum values, the evaluation method becomes identical to that used for folds in the basic method.

The context method is labeled in figures by CPT4, CPT6, CPT8 for minimal similarity thresholds values equal to 0.4, 0.6, and 0.8. The produced sets of generalized context patterns are labeled by GCPT4-A, GCPT6-A, GCPT8-A. The basic approach is labeled by ESD3, ESD5, ESD7 for prediscretization at 3, 5 and 7 splits for each dimension what translates to 9, 25 and 49 folds for the context size (dimension) of 2. Each experiment has been repeated on a set of 8 different context databases generated with the same generator settings. Then, each generated database has been mined by all tested mining methods. The results presented are averages. The experiments shown here have following default settings: number of sequences in the database = 256, sequence size = 8, number of hidden patterns = 4, hidden pattern length = 4, support of hidden patterns = 0.25, number of additional random items in element = 2, size of random items pool = 2000, size of sequence context = 2, size of element context = 2 and distance between context value seeds = 1.2.

The first experiment was conducted to verify the accuracy of mined context patterns. The reconstruction measure values were calculated for the set of context patterns, the set of generalized context patterns and set of patterns mined with the basic non-context algorithm with prediscretization. The results of the first experiment have shown that sets of mined context patterns reflect sets of hidden reference patterns in the most accurate way (Fig. 6). However, these sets are the most numerous ones, too. The context method detects a reasonable number of accurate patterns as soon as the minimal support threshold falls below 0.25. This support threshold value reflects support of reference patterns hidden in the database. After generalization the accuracy is still reasonably good. However, sets of generalized pattern sizes have been reduced by one third (Fig. 7). The context method achieved the best results when relatively weak restrictions were imposed on similarities between contexts. The plots representing sizes of mined pattern sets have a characteristic maximum. It is caused by the maximization procedure implemented in the sequential pattern mining algorithms. In the maximization procedure all shorter patterns are removed from the resulting set if they are exact subsequences of a longer pattern. Hence, when

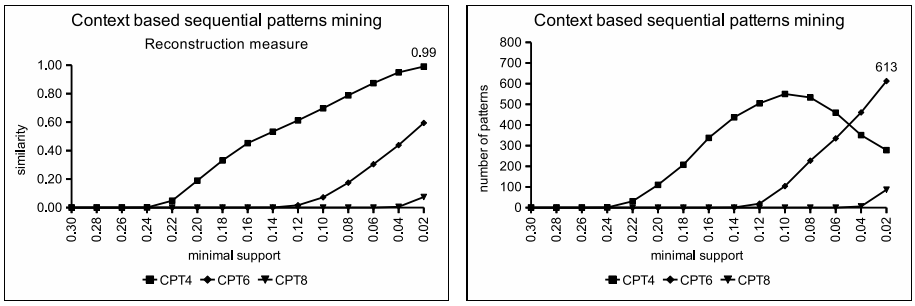

Figure 6. Results from the reconstruction measure and sizes of mined context pattern sets

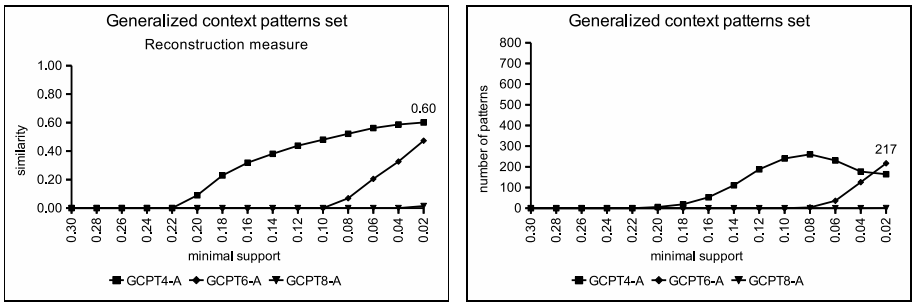

Figure 7. Results from the reconstruction measure and sizes of generalized context pattern sets

the minimal support threshold value decreases, the probability of the longer pattern mining increases. The basic method with prediscretization fared the worst (Fig. 8). Although sizes of mined sets were small but the loss of accuracy was disproportional. The accuracy achieved by the basic method was tenfold lower than for sets of generalized patterns. The sensitivity of the basic method was very low and it detected patterns at the minimal support threshold value equal 0.04.

The subsequent experiments were conducted to verify if conclusions would repeat at different values of the minimal similarity threshold and for different numbers of discretization folds. The results from experiments with the context method for different values of minimal similarity thresholds are presented in Fig. 9. There are cases with and without the context pattern generalization. The figure shows that accuracy of mined context pattern sets increases if the minimal similarity thresholds decrease. This suggests that less restrictive mining provides more accurate sets. The less restrictive mining causes that sets of context attribute values in generalized context patterns have wider ranges

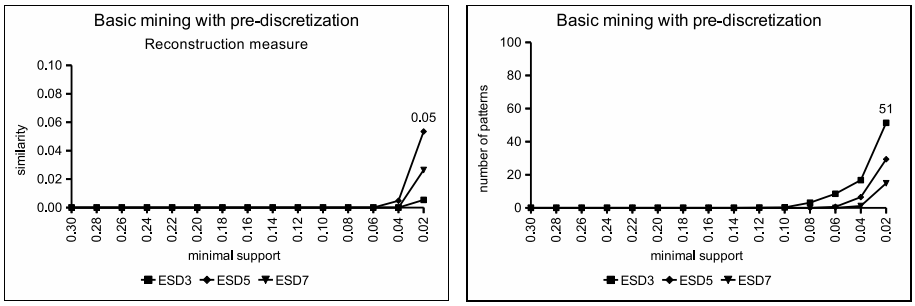

Figure 8. Results from the reconstruction measure and sizes of mined pattern sets for the basic method with the prediscretization

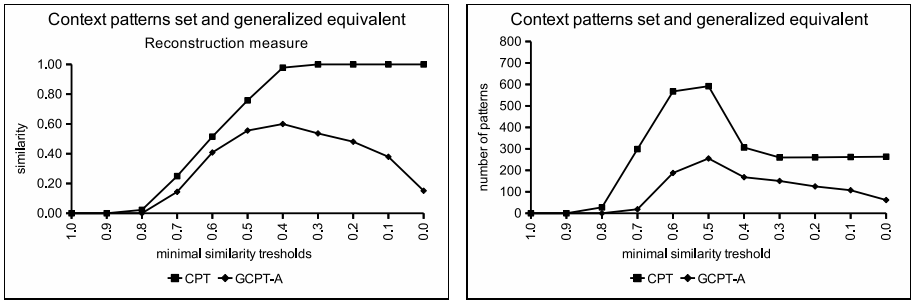

Figure 9. Results from the reconstruction measure and sizes of mined context pattern sets for different values of minimal similarity thresholds

after generalization. However, retrieved sets of generalized context patterns tend to become less accurate if minimal similarity thresholds are set too low and a maximum appears on the plot. The maximum is absent in the case of non-generalized context patterns sets. However, databases containing a lot of "noise" elements mixed with reference patterns data may also cause a maximum on the plot. It happens, because results with "noise" are less accurate. The plots presenting sizes of mined sets have maximums, too. They occur because for more restrictive mining settings each reference pattern from the database may be represented by many shorter patterns in the mined set. However, if the restrictions are eased then these shorter patterns have a greater chance to recombine and create a single longer pattern. Hence, the overall size of the mined pattern set would be smaller after the output set maximization.

The last experiment was conducted for the basic method and variable number of discretization folds (Fig. 10). The results proved that tuning this algorithm by increasing the number of discretization folds would not deliver more accurate results. Apparently, accuracy was getting worse. It can be explained

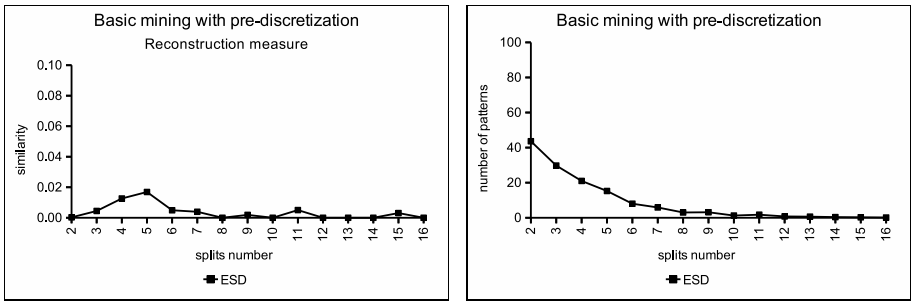

Figure 10. Results from the reconstruction measure and sizes of mined pattern sets for different numbers of discretization folds in the basic approach

in a following way. If the number of discretization folds is growing then the number of contexts embraced by a single fold is getting lower. Such folds have a low support value, entailing computational difficulties, because mining with lower values of minimal support threshold is required. As the experiment was conducted with the constant value of the minimal support threshold, sizes of mined pattern sets were gradually getting smaller with increasing number of folds. The accuracy of mining was at the maximum when 5 splits for each dimension were used. However, even at this setting the accuracy of this method was much worse than in the case of generalized context patterns.

8. Conclusions

This paper has introduced the concept of generalized context patterns together with the algorithm for generalization. Then, the proposed solution was supported by experiments presenting reliability of the algorithm.

The generalization of context patterns is the process of providing smaller sets of patterns accurately representing frequent information from the context database. The generalized context patterns can be obtained by clustering performed on the set of context patterns. In the proposed method clusters of mutually similar context patterns are compacted to generalized context patterns. After this process all associated context values are stored in sets accessible through some presentation functions defined for particular application.

The experiments showed that the accuracy of generalized context patterns was better than of the basic method with prediscretization. However, it was worse than that of the unmodified context patterns (not generalized). The context method may be particularly useful if an accurate method for some continuous attribute values describing elements or sequences in the sequences database is required. Mined patterns are usually used as source for subsequent processing or for data explanation. Success of many real-life applications related, e.g., to medicine, chemistry or mechanics depends on the accuracy of the processed

data. The context based sequential pattern mining is immune to prediscretization errors, delivering more accurate context patterns than the basic approach with prediscretization. Thus, it may be a better choice for some applications than the basic approach.

References

Agrawal, R. and Srikant, R. (1995) Mining sequential patterns. *Proceedings of the 11th International Conference on Data Engineering*. IEEE Computer Society, 3–14.

Guha, S., Rastogi, R. and Shim, K. (2000) ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems*, **25**, 345–366.

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U. and Hsu, M.-C. (2001) PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. *Proceedings of the 17th International Conference on Data Engineering*, IEEE Computer Society, 215–224.

Morzy, T., Wojciechowski, M. and Zakrzewicz, M. (1999) Pattern-Oriented Hierarchical Clustering. *Proceedings of the third East-European Symposium on Advances in Databases and Information Systems – ADBIS'99*, Slovenia, **LNCS 1691**, 179–190.

Ng, R.T. and Han, J. (2002) CLARANS: A Method for Clustering Objects for Spatial Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, **14**, 1003–1016.

Pilevar, A.H. and Sukumar, M. (2005) GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases. *Pattern Recognition Letters*, **26**, 999–1010.

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q. and Dayal, U. (2001) Multi-dimensional sequential pattern mining. *Proceedings of the 10th International Conference on Information and Knowledge Management*, ACM, 81–88.

Plantevit, M., Choong, Y., Laurent, A., Laurent, D. and Teisseire, M. (2005) M2SP: Mining Sequential Patterns Among Several Dimensions. **LNAI 3721**, Springer, 205–216.

Plantevit, M., Laurent, A. and Teisseire, M. (2008) Up and Down: Mining Multidimensional Sequential Patterns Using Hierarchies. *Data Warehousing and Knowledge Discovery*. **LNCS 5182**, Springer, 156–165.

Srikant, R. and Agrawal, R. (1996) Mining Sequential Patterns: Generalizations and Performance Improvements. *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, **L**NCS 1057, Springer-Verlag, 3–17.

Stefanowski, J. and Ziembiński, R. (2005) Mining Context Based Sequential Patterns. *Proceedings of the 3rd International Atlantic Web Intelligence Conference: Advances in Web Intelligence*, **L**NCS 3528, Springer, 401–407.

Stefanowski, J. and Ziembiński, R. (2009) An Experimental Evaluation of Two Approaches to Mining Context Based Sequential Patterns. *Control and Cybernetics*, **31** (1), 27–45.

Yang, Y., Guan, X. and You, J. (2002) CLOPE: a fast and effective clustering algorithm for transactional data. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 682–687.

Yang, J. and Wang, W. (2003) CLUSEQ: Efficient and Effective Sequence Clustering. *Proceedings of the 19th International Conference on Data Engineering*, IEEE Press, 101–112.

Ziembiński, R. (2007) Algorithms for Context Based Sequential Pattern Mining. *Fundamenta Informaticae*, **76** (4), 495–510.