Mining short sequential patterns for hepatitis type detection

Sujeevan ASEERVATHAM and Aomar OSMANI

Université de Paris-Nord, Laboratoire LIPN-CNRS UMR 7030
F-93430 Villetaneuse Cedex, France

Abstract. This paper presents the framework we have developed to classify patients according to the type of hepatitis. To detect the type of virus, once the data have been prepared and encoded in a suitable way, we have extracted the sequential patterns for each virus. Temporal differences between hepatitis B and C can then be seen as patterns that are frequent in one data series and infrequent in the other. The B virus and C virus patterns, extracted under specific constraints, were used to classify patients according to the hepatitis virus. In this work, we especially studied the use of very short patterns for virus type detection. However, the framework allows the mining and the use of longer patterns.The results have shown that in more than half of cases, short patterns can reveal the type of virus with a low level of errors.

1 Introduction

In this paper, we present the work that we have realized in order to discover temporal differences between the hepatitis B and C. This problem was raised by the Chiba University hospital and constitutes one of the subjects of the PKDD Discovery Challenge 2004 and 2005. A hepatitis dataset was provided by the Chiba hospital. It contains long time-series data on the laboratory examinations of hepatitis B and C infected patients. The examinations were realized between 1982 and 2001 on 771 patients.
Our approach to discover temporal differences is based on the extraction of short frequent sequential patterns from the in-laboratory examinations only. To reach our aim, we have developed a framework, which discretizes numerical data into categorical data and uses a constraints incorporated frequent sequence mining program based on the SPADE [1] algorithm. The frequent patterns are checked against the class labels and then pruned according to their confidence values. Finally, the obtained rules are evaluated on a test database. The results of the experiments are encouraging and can be improved.

Objectives. In this study, our aim was to discover temporal differences between the hepatitis B and C. Given two datasets, one containing the observations of patients infected with hepatitis B and the other with hepatitis C, we can define a temporal difference as a temporal pattern that is frequent in one dataset and infrequent in the other.
According to this definition, we have concentrated our work on the analysis of short sequences. Our framework, developed in C++, is able to extract, under user-defined constraints, frequent sequences from the hepatitis dataset. In this work, we were only interested in a classification system that uses short sequences

with the constraint that the period between two examinations should not exceed three months.

Organization. In section 2, we describe the data preprocessing used to generate a dataset of sequences. The sequential pattern generating algorithm and the classification system are defined in section 3. An experimental evaluation of our classification system is presented in section 4. Finally, we conclude in section 5.

2 Data Preprocessing

The hepatitis dataset was provided by the Chiba University Hospital. It contains the results of examinations realized from 1982 to 2001 on 771 patients infected with hepatitis B and C viruses.

2.1 Dataset cleaning

For the data cleaning operation and the statistical analysis, we used the R environment [2]. Using R, we did the following operations on the data set :

1. cleaning the data of incorrect values,
2. merging the 7 tables of the initial dataset into one table,
3. acquiring basic statistic knowledge of the dataset,
4. selecting relevant variables for hepatitis type discrimination.

The final table was constituted of 548 patients and 37 variables[1]. The distribution of patients according to their sex and the type of virus infection is shown in the table 1.

Patient's sex	Hepatitis B	Hepatitis C	Interferon
Female	49	97	60
Male	204	198	136

Table 1. Final *inlab* Patient Distribution Table

2.2 Data Transformation

To extract patterns and classify patients, we needed to transform the dataset into a collection of suitable sequence formatted data.

Definition 1. *We define a sequence event as a non-empty unordered*[2] *set of measurements realized at the same time. Each event is composed of the pair* $(variable, measurement)$*. We will use the term examination as a synonym of event.*
An event with n *measurements will be denoted as* $\{e_1, \ldots, e_n\}$*.*

[1] Variables were scaled according to the interval of "normality". An interval of "normality" for a variable is defined by a lower and upper bounds values between which measurements are considered as being normal. This interval is defined for each variable in the *labn* table.
For each measurement x of a variable with an interval $[L, H]$, x was scaled using the following formula : $x_{scaled} = \frac{x-L}{H-L}$.

[2] However, according to [3, 4, 1] we can assume without loss of generality that the events are lexicographically sorted.

Definition 2. *We shall say that a sequence is a non-empty time-ordered collection of events. A sequence S composed of n examinations and k measurements will be called a k-sequence and it will be denoted as $S = s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_n$ [1]. Indeed, a k-sequence S of n elements verifies : $k = \sum_{s_j \in S} \mid s_j \mid$*

The processed *inlab* database was transformed into a collection of sequences. For this, each patient's examinations identified by the value of the MID variable were grouped together and sorted according to the date of examination.
Nevertheless, most of the sequential mining algorithms processing only symbolic data, we discretized measurements into a finite number of intervals. Each interval being a categorical value for a variable.

For the discretization algorithm, a supervised [3] global[4] method based on a minimal entropy heuristic described in [5] was used. This method uses the class information entropy to find the best cut points. Given a data set S, the attribute A to discretize, a class attribute C, a cut point T, the set $S_1 \subseteq S$ with A-values$(S_1) \leq T$ and the set $S_2 = S - S_1$, the best cut point is the one that minimizes $E(A, T; S)$:

$$E(A, T; S) = \frac{\mid S_1 \mid}{\mid S \mid} Ent(S_1) + \frac{\mid (S_2) \mid}{\mid S \mid} Ent(S_2)$$

$$Ent(S) = -\sum_i P(C_i, S) \log_2(P(C_i, S))$$

Once the best cut point is found and accepted, the method recursively partitions S_1 and S_2. The Minimum Description Length Principle (MDLP) is used as a stopping criterion. Thus, the recursion is stopped and the cut point is rejected iff :

$$Gain(A, t; S) \leq \frac{\log_2(\mid S \mid -1)}{\mid S \mid} + \frac{\Delta(A, T; S)}{\mid S \mid}$$

with $Gain(A, t; S) = Ent(S) - E(A, T; S)$ and
$\Delta(A, T; S) = \log_2(3^k - 2) - (kEnt(S) - k_1 Ent(S_1) - k_2 Ent(S_2))$ where k_i is the number of class labels in the set S_i.

Once all the transformations were done, we have obtained a collection of patients sequences composed of events in which all measurements were encoded with integers. An examination or event was, then, defined as a set of integers with each integer corresponding to a pair $(variable, measurement)$.

3 Sequence Mining

3.1 Mining Algorithm

For the criteria of speed and class decomposition, we have used cSPADE [6] in our framework for mining patterns.
cSPADE is based on the SPADE algorithm [1]. The only difference is that SPADE does not handle the time constraints.

[3] A supervised method uses the class information to partition the data.
[4] Global discretization methods produce, for an attribute, partitions that are applied to the entire dataset independently of the other attributes.

The main key of cSPADE is the use of a vertical database instead of the common horizontal database. In the horizontal database, sequences are represented in a quite natural way : measurements (items) are grouped together by events and events are grouped by patients in a chronological order. Whereas in the vertical database, measurements (items) are lexicographically ordered[5] and with each measurement is associated a set of pairs of patient id and examination date. Therefore, for each measurement, thanks to the patient id we can identify the sequences in which it appears and for each sequence we can identify, thanks to the date, the examinations to which the measurement belongs.
cSPADE, using a bottom-up lattice[6] traversal, starts at the first level[7] with the set of frequent 1-sequences. It joins the sequences between them and jumps to the minimal upper level.

The vertical database is well suited for the join operations. The support counting for a sequence resulting of a join operation is straightforward.
Another advantage of cSPADE is that it can decompose the lattice into small equivalence classes and then mine the frequent patterns, independently, in each classes.

3.2 Rules generation

Given $\mathcal{D}_\mathcal{T}$, the complete training dataset, for each class C, once the frequent sequences have been extracted from the specific class training data $\mathcal{D}_\mathcal{C} \subseteq \mathcal{D}_\mathcal{T}$, we have calculated for each extracted patterns m the confidence of the rule $m \Rightarrow C$ from the formula below [6] :

$$conf(m \Rightarrow C) = \frac{\sigma_{\mathcal{D}_\mathcal{C}}(m)}{\sigma_{\mathcal{D}_\mathcal{T}}(m)}$$

where $\sigma_\mathcal{D}(m)$ is the number of sequences in $\mathcal{D}$ that contain m.

Then, given a user-specified minimum confidence threshold, all the rules that had a confidence value below the threshold were eliminated.

Moreover, as we are dealing with short sequences, we can expect that many rules will be triggered by an input-sequence[8]. Therefore we cannot rely on the rules confidences. Thus, we have proposed an heuristic that relies on the ratio of the number of rules triggered for the class to the total number of rules for this class.

For an input-sequence, we compare, for each class, the ratio of triggered rules. Then, we label the input-sequence with the label of the class that had the highest ratio. Nevertheless if the difference between the ratios are less than 10%

[5] Measurements can also be ordered according to other criteria and even not ordered at all.

[6] It is not exactly a lattice but a hyper-lattice. Indeed, the join of two elements of the same level gives three elements. For example, the join of elements A and B will results in elements $\{\{A, B\}, A \rightarrow B, B \rightarrow A\}$.

[7] In fact, an optimization approach is to start at the second level after having determined all the frequent 1-sequences as well as all the frequent 2-sequences.

[8] A rule is triggered by a sequence if the rule's antecedent is a subsequence of the sequence according to the various constraints.

then the input-sequence cannot be labelled as the extracted information are not sufficient to classify such sequences.

4 Experimental Evaluation and Results

The *inlab* table has been partitioned into three datasets for 3-fold cross-validation. Each subset was extracted from the *inlab* table by randomly selecting sequences. However, we have extracted the data according to the proportion of the patients infected with hepatitis B. Thus, in *inlab* there were 41% of the patients who were infected with B virus and the others with C virus. This ratio was kept in each subset. Moreover, we have separated the patients according to the type of virus they were infected with.

For each class B and C, we have extracted patterns using the parameters defined in table 2. As we wanted short sequences, we have selected a high value for the minimum support since we know that higher is the minimum support and shorter will the frequent patterns be. Moreover, since we're interested in detecting the hepatitis virus type within a short period of examinations, we have set, for the desired patterns, the maximum time distance between two successive examinations to 90 days.
Nonetheless, we have set a low value for the minimum confidence value of the generated rules because as the extracted patterns will be short, the probability for sequences to contain the patterns will be important. Thus, we cannot expect high confidence values for the rules that will be generated from short sequences.

The statistics about the results of the patterns extraction and rules generation are summarized in table 4. As we have expected, the frequent patterns are very short. Their average length is about 4.1 examinations per pattern.
Due to the sliding window and the maximal gap that increase the combinatorial possibilities, the number of generated patterns are very important. However, the rules confidences are low and near 94% of the generated patterns were not suitable to be used in classification rules.

To check the validity of the extracted rules, we have applied them on the test data. The result of the test is shown in table 4. Our classification program has successfully detected the type of virus for about 59% of the tested patients and for nearly 32.7% the rules weren't enough discriminatory and that can be explained by the fact that the extracted patterns are too general to detect specific cases. However, when the discrimination was possible, the system has, correctly, classed 86% of the patients.

	Hep. B	Hep. C
minimum support	85%	85%
sliding window	45 days	45 days
minimum gap	0 day	0 day
maximum gap	90 days	90 days
minimum rule confidence	60%	60%

Table 2. Pattern extraction parameters

	Training Data		Patterns		Rules	
	Hep. B	Hep. C	Hep. B	Hep. C	Hep. B	Hep. C
Number of sequences	110	153	453706	333988	36531	10618
Number of unique measurements	182	178.33	30.67	18	22.66	12.33
Average sequence length	64.44	42.01	4.61	3.54	4.73	2.52
Average event length	23.81	26.98	1.83	2.15	1.74	2.25

Table 3. Average 3-fold cross-validation statistics for the generated rules

	Hep. B	Hep. C
Classed as B	57.19%	11.49%
Classed as C	7.96%	61.02%
Classed as Unknown	34.85%	27.49%

Table 4. Average 3-fold cross-validation classification results

5 Conclusion

We have presented a method for extracting short sequential patterns in order to detect the type of hepatitis virus infection. We focused our work on the problematic of detecting the type of hepatitis within a short period of time. The test done on the hepatitis dataset provided by the Chiba University Hospital allowed us after a learning step, to extract very short patterns, which were sufficient, in half of cases, to detect hepatitis virus type with a low rate of errors. The results were encouraging and can be improved by relevantly setting the various parameters of the mining and classification programs.

As future work, we plan, in a first time, to improve the discretization part of the data transformation. In a second time, we want to test our system using the same database but with different problematic and especially we want to analyse the performance of the system when attempting to extract more specific patterns i.e. long patterns.

References

1. M. J. Zaki: Efficient enumeration of frequent sequences. In: CIKM'98, 7th International Conference on Information and Knowledge Management. (1998)
2. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. (2005) ISBN 3-900051-07-0.
3. R. Agrawal, R. Srikant: Mining Sequential Patterns. In: ICDE'95,11th International Conference on Data Engineering. (1995) 3–14
4. R. Srikant, R. Agrawal: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology. (1996) 3–17
5. U. M. Fayyad, K. B. Irani: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI. (1993) 1022–1029
6. M. J. Zaki: Sequence mining in categorical domains: Incorporating constraints. In: CIKM'2000, 9th International Conference on Information and Knowledge Management. (2000) 422–429