


A user parameter-free approach for mining robust sequential classification rules

Elias Egho¹  · Dominique Gay² · Marc Boullé¹ ·
Nicolas Voisine¹ · Fabrice Clérot¹

Received: 24 February 2016 / Revised: 25 August 2016 / Accepted: 3 October 2016 /
Published online: 18 October 2016
© Springer-Verlag London 2016

Abstract Sequential data are generated in many domains of science and technology. Although many studies have been carried out for sequence classification in the past decade, the problem is still a challenge, particularly for pattern-based methods. We identify two important issues related to pattern-based sequence classification, which motivate the present work: the curse of parameter tuning and the instability of common interestingness measures. To alleviate these issues, we suggest a new approach and framework for mining sequential rule patterns for classification purpose. We introduce a space of rule pattern models and a prior distribution defined on this model space. From this model space, we define a Bayesian criterion for evaluating the interest of sequential patterns. We also develop a user parameter-free algorithm to efficiently mine sequential patterns from the model space. Extensive experiments show that (i) the new criterion identifies interesting and robust patterns, (ii) the direct use of the mined rules as new features in a classification process demonstrates higher inductive performance than the state-of-the-art sequential pattern-based classifiers.

Keywords Mining robust sequential rules · Sequence classification · Bayes theory

This work was done while Dominique Gay was a research engineer at Orange Labs.

✉ Elias Egho
Elias.Egho@orange.com

Dominique Gay
Dominique.Gay@univ-reunion.fr

Marc Boullé
Marc.Boulle@orange.com

Nicolas Voisine
Nicolas.Voisine@orange.com

Fabrice Clérot
Fabrice.Clerot@orange.com

¹ Orange Labs, 2, avenue Pierre Marzin, 22307 Lannion Cédex, France

² Université de La Réunion, 2 rue Joseph Wetzell, 97490 Sainte Clotilde, France

1 Introduction

Sequence classification [50] has many real-world applications in a broad range of domains, such as biology [16,44], text mining [42] or web mining [45]. Mining sequential rules for classification has become very popular since the resulting classifier might be interpretable by the domain analyst. A sequential rule is an expression that takes the form of $\pi : s \rightarrow c_i$ where s is the body sequence of the rule and c_i is a value of a class attribute. One can interpret π as “when event sequence s is observed for an object, then it is often an object of class c_i .” An incoming unseen object, which matches a discovered rule pattern, will be more likely of the class indicated by the rule. Adopting the strategy of the pioneering work for transactional data on “Classification Based on Associations” (CBA) [35], several rule-based approaches have been suggested for sequence classification. Generally, pattern-based classification methods [54] follow a similar strategy: Firstly, a sequential rule set is mined w.r.t. an interestingness measure; secondly, either a dedicated classifier, like a decision list or a maximum entropy model, is built upon a selected subset of the mined rules [26,47,53] or the mined rules are directly used as new features in a classification process [15,29,33]. While most of the existing approaches generally lead to good inductive performance, we now highlight two of their weaknesses, namely the curse of parameter tuning and the instability of the interestingness measures.

1.1 The instability of interestingness measures

We justify this claim by considering a motivation example, let us consider three widely used measures for evaluating sequential rules: confidence, growth rate and lift. One can easily show that rule patterns extracted according to these measures are not individually robust. In Fig. 1, we plot test values of confidence (resp. growth rate and lift) against train values of each rule pattern mined by cSPADE [51] (one point per pattern) for the skater data set [39]. We observe very blurred scatter plots, meaning that interestingness measure values are severely unstable from train to test data: A “good” rule w.r.t. an interestingness measure in training phase may turn out to be weak in test phase. Particularly, the top 1000 rules obtained from training data according to each considered measure are clearly not anymore the top 1000

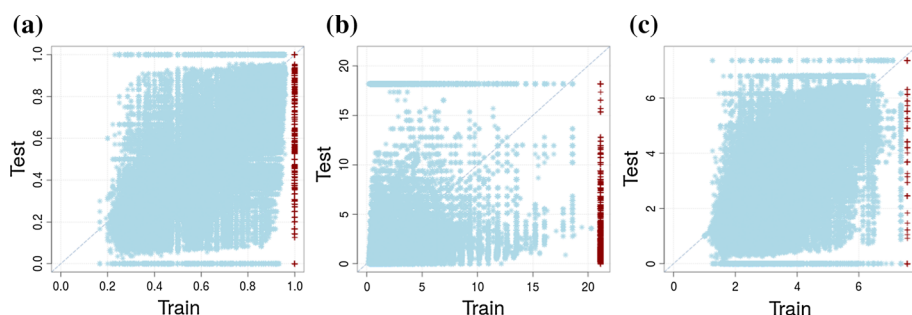


Fig. 1 Comparison of confidence (resp. growth rate and lift) values for sequential classification rules mined by cSPADE [51] in a train–test experiment: 50 % train/50 % test for the skater data set (the red+ are the top 1000 highest confidence (resp. growth rate and lift) values of the mined rules from train data, using a 2 % minimum frequency threshold). Train–test values are expected to be close to the diagonal for a robust interestingness measure, while *blurred scatter plots* are a symptom of unstable measures

when evaluated on test data. Thus, it could be misleading to bet on such rules for classifying new incoming objects.

1.2 The curse of parameter tuning

Most of the existing approaches need parameter tuning. One has to set an interestingness measure threshold (sometimes also with a frequency threshold and a gap constraint) for the mining phase and then choose the number of rules for the final set used for classification. Unfortunately, setting parameters is not an easy task—each application data could require a specific setting. The associated dilemma is well known: For large data sets, low-frequency thresholds lead to an untractable task or a huge number of output patterns many of which are spurious, while high-frequency thresholds produce too few patterns with low class-discrimination power. Moreover, the predictive performance of rule-based classifiers highly depends on these settings [10].

These two weaknesses suggest that there is room for improving inductive performance and ergonomy of pattern-based sequence classification methods. The main contributions of this work tackle these two problems and are summarized as follows:

1.3 Toward a robust criterion for evaluating sequential classification rules

We embrace the Bayes theory and suggest a Bayesian criterion, called *level*, for identifying interesting and robust sequential classification rules. Our suggested framework has been already successfully instantiated for several data mining tasks such as supervised discretization [6] and classification rule mining in transactional data sets [21]. The *level* criterion is based on the a posteriori probability of a rule model given the data and does not require any wise threshold setting.

1.4 A user parameter-free approach for mining sequential classification rules

We discuss and present a new algorithm **MiSeRe** for **Mining Sequential Classification Rules**. MiSeRe is anytime algorithm—the more time the user grants to the task, the more it learns. MiSeRe is a user parameter-free approach and does not require any parameter tuning. Our algorithm employs an instance-based randomized strategy that promotes diversity mining. It uses a bitset representation of the data, to efficiently mine the sequential classification rules.

1.5 Experiments and evaluations

To validate our contributions, we perform an extensive experimental evaluation on a variety of data sets, including biological sequences, web usage logs and text sequences. The main results are unequivocal: (i) the suggested Bayesian criterion identifies interesting and robust sequential patterns; (ii) using the extracted sequential rules as new features in a classification process outperforms state-of-the-art sequential rule-based classifiers in terms of predictive performance. The software source code of MiSeRe, data used for experiments and interactive result visualizations are publicly available from [11] <http://www.misere.co.nf>.

This article is an extended version of our previous work on the same topic [17]. The new contributions of this work can be summarized as follows: (1) We provide an asymptotic behavior study of our criterion *level*, (2) our mining method *MiSeRe* is described with more details, (3) more extensive experiments are presented and (4) we also evaluate our classification system on a real large marketing data from the French Telecom company Orange.

The rest of the paper is organized as follows: Sect. 2 briefly reviews the context and basic definitions. We suggest a new framework for mining sequential classification rules and define a robust criterion for pattern evaluation in Sect. 3. Section 4 describes our mining strategy and parameter-free algorithm. We report an empirical evaluation of our method in Sect. 5. Section 6 reviews the related work with a focus on pattern-based sequence classification before concluding.

2 Preliminaries

Let $\mathcal{I} = \{e_1, e_2, \dots, e_m\}$ be a finite set of m distinct items. A **sequence** s over \mathcal{I} is an ordered list $s = \langle s_1, \dots, s_{\ell_s} \rangle$, where $s_i \in \mathcal{I}$; ($1 \leq i \leq \ell_s$, $\ell_s \in \mathbb{N}$). An atomic sequence is a sequence with length 1. A sequence $s' = \langle s'_1 \dots s'_{\ell_{s'}} \rangle$ is a **subsequence** of $s = \langle s_1 \dots s_{\ell_s} \rangle$, denoted by $s' \preceq s$, if there exist indices $1 \leq i_1 < i_2 < \dots < i_{\ell_{s'}} \leq \ell_s$ such that $s'_z = s_{i_z}$ for all $z = 1 \dots \ell_{s'}$ and $\ell_{s'} \leq \ell_s$. s is said to be a **supersequence** of s' . $\mathbb{T}(\mathcal{I})$ will denote the (infinite) set of all possible sequences over \mathcal{I} . Let $\mathcal{C} = \{c_1, \dots, c_j\}$ be a finite set of j distinct classes. A **labeled sequential data set** \mathcal{D} over \mathcal{I} is a finite set of triples (sid, s, c) with sid is a sequence identifier, s is a sequence ($s \in \mathbb{T}(\mathcal{I})$) and c is a class value ($c \in \mathcal{C}$). The set $\mathcal{D}_{c_i} \subseteq \mathcal{D}$ contains all sequences that have the same class label c_i (i.e., $\mathcal{D} = \bigcup_{i=1}^j \mathcal{D}_{c_i}$). The following notations will be used in the rest of the paper:

- m : Number of items in \mathcal{I} .
- j : Number of classes in \mathcal{C} .
- n : Number of triples (sid, s, c) in \mathcal{D} .
- n_c : Number of triples (sid, s, c) in \mathcal{D}_c .
- ℓ_s : Number of items in the sequence s .
- k_s : Number of distinct items in the sequence s , ($k_s \leq \ell_s$).
- ℓ_{\max} : Number of items in the longest sequence of \mathcal{D} .

Definition 1 (*Support of a sequence*) Let \mathcal{D} be a labeled sequential data set and let s be a sequence. The **support** of s in \mathcal{D} , denoted $f(s)$, is defined as:

$$f(s) = |\{(sid', s', c') \in \mathcal{D} | s \preceq s'\}|$$

The value of $n - f(s)$ can be written as $\overline{f(s)}$. The support of s in \mathcal{D}_c is noted $f_c(s)$ and $\overline{f_c(s)}$ stands for $n_c - f_c(s)$.

Definition 2 (*Standard Classification Rule Model*) Let \mathcal{D} be a labeled sequential data set with j classes. A sequential classification rule π is an expression of the form:

$$\pi : s \rightarrow f_{c_1}(s), f_{c_2}(s), \dots, f_{c_j}(s)$$

where s is a sequence, called body of the rule, and $f_{c_i}(s)$ is the support of s in each \mathcal{D}_{c_i} , $i = 1 \dots j$.

This definition of classification rule is slightly different from the usual definition where the consequent is a class value. It refers to the notion of distribution rule [28] and allows us to access the whole frequency information within the contingency table of a rule π —which is needed for the development of our framework.

Example 1 We use the sequence database \mathcal{D} in Table 1 as an example. It contains four data sequences (i.e., $n = 4$) over the set of items $\mathcal{I} = \{a, b, d, e, f, g\}$ (i.e., $m = 6$). $\mathcal{C} = \{c_1, c_2\}$

Table 1 \mathcal{D} : a tiny labeled sequential data set as an example

SID	Sequence	Class
1	$\langle baadg \rangle$	c_1
2	$\langle agbe \rangle$	c_1
3	$\langle badgb \rangle$	c_2
4	$\langle eefgbg \rangle$	c_2

is the set with two classes (i.e., $j = 2$). The longest sequence of \mathcal{D} is $s = \langle eefgbg \rangle$ (i.e., $\ell_s = \ell_{\max}$), $\ell_{\max} = 6$ while $k_s = 4$. Sequence $\langle aad \rangle$ is a subsequence of $\langle baadg \rangle$. The sequence $\langle a \rangle$ is an atomic sequence. Given the sequence $s = \langle ab \rangle$, we have $f(s) = 2$, $\overline{f(s)} = 2$, $f_{c_1}(s) = 1$, $\overline{f_{c_1}(s)} = 1$, $f_{c_2}(s) = 1$ and $\overline{f_{c_2}(s)} = 1$. $\pi : \langle ab \rangle \rightarrow f_{c_1}(\langle ab \rangle) = 1$, $f_{c_2}(\langle ab \rangle) = 1$ is a sequential classification rule.

3 Bayesian framework for sequential pattern

Standard classification rule evaluation criteria aim at selecting *general* rules (e.g., based on the frequency constraint) and informative rules that characterize classes (e.g., based on confidence or growth rate). However, the trade-off between generality and informativeness is difficult to achieve and usually rely on manual parameter tuning. Using a Bayesian approach, we aim at obtaining a statistical evaluation criterion with the expectation of automatically and optimally finding the best trade-off between generality and informativeness.

Following the framework introduced by [6], from a Bayesian point of view, the problem of sequential classification pattern mining is formulated as a model selection problem. To choose the “best” sequential rule model from the model space, we use a Bayesian Maximum A Posteriori approach: We look for maximizing $p(\pi|\mathcal{D})$, the posterior probability of a rule model π given the data \mathcal{D} . According to Bayes rule, $p(\pi|\mathcal{D})$ is given as :

$$p(\pi|\mathcal{D}) = \frac{p(\pi, \mathcal{D})}{p(\mathcal{D})} = \frac{p(\pi) \times p(\mathcal{D}|\pi)}{p(\mathcal{D})}$$

Considering that $p(\mathcal{D})$ is constant in the current optimization problem, it goes back to the maximization of the expression $p(\pi) \times p(\mathcal{D}|\pi)$. The evaluation criterion, called cost, is based on the negative logarithm of $p(\pi|\mathcal{D})$ and is expressed as follows:

$$\text{cost}(\pi) = -\log(\underbrace{p(\pi)}_{\text{prior}} \times \underbrace{p(\mathcal{D}|\pi)}_{\text{likelihood}}) \propto -\log(\underbrace{p(\pi|\mathcal{D})}_{\text{posterior}})$$

Now to choose the best rule for data \mathcal{D} , we have to minimize the cost of a sequential classification rule.

To compute the prior $p(\pi)$, we complement Definition 2 of sequential classification rules with a hierarchy of parameters that uniquely identifies a given rule in the rule model space:

Definition 3 (*Standard Classification Rule Model*) A sequential classification rule (or SCR model) $\pi : s \rightarrow f_{c_1}(s), f_{c_2}(s), \dots, f_{c_j}(s)$ is defined by:

- the constituent items of the rule body s .
- the order of occurrence the items in the body s .
- the class distribution inside and outside of the body s .

Our working model space is then the space all SCR models. Considering the hierarchy of parameters from the definition of SCR model, we use the following hierarchical prior distribution on SCR models:

1. The number of distinct items k_s in a rule body s is uniformly distributed between 0 and m .
2. The length of the sequence s in a rule body is uniformly distributed between 0 and ℓ_{\max} .
3. For a given number k_s of items, every subset of k_s distinct items of the m items is equiprobable.
4. For a given number of distinct items k_s and for a given number of items in sequence ℓ_s , every ordered set of ℓ_s items of the k_s distinct items is equiprobable.
5. Every distribution of the class values is equiprobable, in and outside of the body.
6. The distributions of class values in and outside of the body are independent.

Notice that such a prior is uniform at each stage of the hierarchy; it does not mean that the hierarchical prior is a uniform prior over the rule space, which would be equivalent to a maximum likelihood approach. From the definition of the model space and its prior distribution, we can now give an expression of the prior probability ($p(\pi)$) of a rule model and the probability ($p(\mathcal{D} \mid \pi)$) of the data given a model π , i.e., the likelihood of π .

3.1 Prior probability

The prior probability of a rule model π is:

$$p(\pi) = p(s) \times p(\{f_{c_i}(s)\}_{i=1}^j \mid \overline{\{f_{c_i}(s)\}_{i=1}^j} \mid f(s), \overline{f(s)})$$

Expanding each term of the prior turns into an enumeration problem. The first two hypotheses assume uniform distribution, which lead to $m + 1$ and $\ell_{\max} + 1$ enumeration terms. The third hypothesis assume the equiprobability of every set of k_s constituent distinct items of the sequence body. The number of combinations $\binom{m}{k_s}$ is a natural candidate to compute this prior term; however, it is symmetric. Adding new items (beyond $m/2$) to the body makes the rule more probable, which is an undesired effect. Indeed, adding spurious items is favored even if it has an insignificant impact on the likelihood of the model. To obtain simpler models, we prefer a parsimonious prior that increases with k_s : Considering a multinomial distribution with q independent trials and m equiprobable outcomes, the likelihood of a draw with counts (q_1, \dots, q_m) such that $\sum_{i=1}^m q_i = q$ is $\frac{q!}{q_1! \dots q_m!} \prod_i (\frac{1}{m})^{q_i}$. If we keep only the draws for which all items are distinct, we obtain $\frac{q!}{m^q}$. The fourth hypothesis promotes the equiprobability of every ordered set of ℓ_s items over k_s distinct items; here we use the exponential term $k_s^{\ell_s}$. We now have $p(s)$:

$$p(s) = \frac{1}{m+1} \times \frac{1}{\ell_{\max}+1} \times \frac{k_s!}{m^{k_s}} \times \frac{1}{k_s^{\ell_s}} \quad (1)$$

Considering the last two hypotheses, enumerating the distributions of the j classes in and outside of the body is a combinatorial problem:

$$p(\{f_{c_i}(s)\}_{i=1}^j \mid f(s), \overline{f(s)}) = \frac{1}{\binom{f(s)+j-1}{j-1}} \quad (2)$$

$$p(\overline{\{f_{c_i}(s)\}_{i=1}^j} \mid f(s), \overline{f(s)}) = \frac{1}{\binom{\overline{f(s)}+j-1}{j-1}} \quad (3)$$

3.2 Likelihood

The probability of the data given the rule model $p(\mathcal{D}|\pi)$ is the probability of observing the data inside and outside of the rule body (w.r.t. $f(s)$ and $\overline{f(s)}$) given the multinomial distribution:

$$p(\mathcal{D}|\pi) = \frac{1}{\frac{f(s)!}{\prod_{i=1}^j f_{c_i}(s)!}} \times \frac{1}{\frac{\overline{f(s)}!}{\prod_{i=1}^j \overline{f_{c_i}}(s)!}} \quad (4)$$

3.3 Cost of SCR

Using the previously defined prior and posterior terms, the complete and exact definition of the cost of SCR is then:

$$\begin{aligned} \text{cost}(\pi) = & \log(m+1) + \log(\ell_{\max}+1) + \log\left(\frac{m^{k_s}}{k_s!}\right) + \log(k_s^{\ell_s}) \\ & + \log\binom{f(s)+j-1}{j-1} + \log\binom{\overline{f(s)}+j-1}{j-1} \\ & + \log(f(s)!) - \sum_{i=1}^j \log(f_{c_i}(s)!) + \log(\overline{f(s)}!) - \sum_{i=1}^j \log(\overline{f_{c_i}}(s)!) \end{aligned}$$

The amplitude of the cost values depends on the number n of sequences and the number m of items in the data set. For convenience, we defined a normalized criterion, called *level*, which plays the role of an interestingness measure to evaluate and compare SCR models.

Definition 4 (Level) Given a SCR model π , the level of π is defined as:

$$\text{level}(\pi) = 1 - \frac{\text{cost}(\pi)}{\text{cost}(\pi_{\emptyset})}$$

where $\text{cost}(\pi_{\emptyset})$ is the cost of the null model (i.e., default rule with empty sequence body). The cost of the default rule π_{\emptyset} is formally:

$$\text{cost}(\pi_{\emptyset}) = \log(m+1) + \log(\ell_{\max}+1) + \log\binom{n+j-1}{j-1} + \log(n!) - \sum_{i=1}^j \log(n_{c_i}!)$$

The *level* naturally highlights the border between the interesting patterns and the irrelevant ones. Indeed, rules π such that $\text{level}(\pi) \leq 0$ are less probable than the default rule π_{\emptyset} . Then using them to explain the data by characterizing classes of sequence objects is more costly than using π_{\emptyset} ; such rules are considered spurious. Rules such that $0 < \text{level}(\pi) \leq 1$ highlight the interesting patterns. In fact, rules with lowest cost (highest *level*) are the most probable arising from the data and show the strongest correlations between the rule body and the class attribute.

As negative log of probabilities can be interpreted as a coding length [43], our model is closely related to the minimum description length (MDL) approach [24,41], which aims at approximating the Kolmogorov complexity [48] for the coding length of the data. The prior term in the *cost* represents the description length of a rule model, whereas the likelihood term represents the coding length of the data given the rule model.

3.4 Asymptotic behavior

The accuracy of the model is indicated by the likelihood term, which is the principal term of the cost, whereas the rest of the terms act as regularization terms. These regularization terms penalize complex models and prevent from over-fitting. When the number of sequences n of the problem is very high, the regularization terms are negligible and the cost function is linked with Shannon class entropy [12].

Theorem 1 *The cost of the default rule π_\emptyset for a data set made of n sequences is asymptotically n times the Shannon class entropy $H(y)$ of the whole data set when $n \rightarrow \infty$.*

$$\text{cost}(\pi_\emptyset) = n \times H(y) + O(\log(n))$$

Proof See Appendix.

Theorem 2 *The cost of the rule π for a data set made of n sequences is asymptotically n times the Shannon conditional class entropy $H(y|s)$ of the whole data set when $n \rightarrow \infty$.*

$$\text{cost}(\pi) = n \times H(y|s) + O(\log(n))$$

Proof See Appendix.

The asymptotic equivalence between the coding length of the default rule π_\emptyset and the class entropy of the data confirms that “rules such that $\text{level} \leq 0$ identify patterns that are not statistically significant” and links our model with the notion of incompressibility of Kolmogorov [38] which defines randomness as the impossibility of compressing the data shorter than its raw form. The asymptotic behavior of the cost function (for a given rule π) confirms that high-level values highlight the most probable rules that characterize classes, since high-level value means high compression of the class data given rule π .

4 Mining sequential classification rules

Mining sequential patterns [1] is a NP-hard problem. The anti-monotone property of frequency measure and condensed representations of frequent patterns [37] allows to save computational time though the problem remains time and memory costly for large-scale data sets (see [5] for the case of sequential classification rules). Our *level* evaluation criterion does not hold the property of anti-monotonicity as the frequency. Thus, if we look for the whole set of SCRs with positive *level* values, an exhaustive exploration of the search space is not conceivable. Indeed, the size of the search space is exponential with m the number of items:

$$\sum_{i=1}^{\ell_{\max}} m^i \equiv O(m^{\ell_{\max}})$$

That’s why we opt for a simpler and more realistic formulation of the problem: “Mining with diversity a subset of SCRs with positive *level* values.”

In the following, we describe our algorithm **MiSeRe** for **Mining Sequential Classification Rules**. The main features of MiSeRe (see Algorithm 1) are:

1. MiSeRe is user parameter-free algorithm.
2. MiSeRe employs an instance-based randomized strategy that promotes diversity mining.

3. MiSeRe uses a bitset representation and Boolean operations, to efficiently mine the sequential classification rules.
4. MiSeRe is anytime—the more time the user grants to the task, the more it learns.

Firstly, we generate all SCRs whose body is an atomic sequence, such rules with positive *level* values are chosen (Lines 2–3). These rules are selected because the short sequences are more probable and preferable as the cost of the rule $c(\pi)$ is smaller for lower ℓ_s and k_s values, meeting the consensus: “*Simpler models are more probable and preferable*”. The stopping condition Line 4 refers to the running time that the end user provides to the mining process. Contrary to common parameters to be set in pattern mining methods (e.g., frequency threshold and interestingness measure threshold), the time constraint (i.e., the time granted to the mining phase) can be more easily managed by domain experts or operational teams since it corresponds to an operational parameter. At each iteration of the main loop (Lines 4–12), a SCR is built and when time is up, the process ends and the current rule set is output. We randomly choose one sequence s from the labeled sequential database \mathcal{D} (Line 5). Then, we count the number of all subsequences that can be generated for s (denoted as ads), and we employ the efficient counting procedure presented in [18]. The inner loop (Lines 7–12) generates randomly $\log(ads)$ subsequences of the chosen sequence s to promote diversity instead of exhaustiveness for the coverage of s . This generation (Line 8) is done by randomly removing z items from s where z is between 1 and $\ell_s - 2$. Then, the rule π is built based on the generated subsequence s' . Finally, the rule π is added to the rule set if its level value is positive and it is not already in \mathcal{R} .

Algorithm 1: MiSeRe

```

input :  $\mathcal{D}$ , a Labeled Sequential Data Set
output:  $\mathcal{R}$ , a Set of SCRs
1 begin
2    $\mathcal{S} = \{s = \langle s_1 \rangle; s_1 \in \mathcal{I}\}$ ;
3    $\mathcal{R} = \{\pi : s \rightarrow f_{c_1}(s), \dots, f_{c_j}(s); s \in \mathcal{S} \wedge level(\pi) > 0\}$ ;
4   while  $\neg StoppingCondition$  do
5      $s = ChooseRandomSequence(\mathcal{D})$ ;
6      $ads = ComputeNumberOfSubsequences(s)$ ;
7     for  $i = 1$  to  $\log(ads)$  do
8        $s' = GenerateRandomSubsequence(s)$ ;
9        $\pi : s' \rightarrow f_{c_1}(s'), \dots, f_{c_j}(s')$ ;
10      if  $level(\pi) > 0$  then
11        if  $\pi \notin \mathcal{R}$  then
12           $\mathcal{R} = \mathcal{R} \cup \{\pi\}$ 
13 return  $\mathcal{R}$ ;

```

4.1 A bitset representation

The main challenge in this algorithm is “*how to efficiently compute the distribution of the sequence s' in each class; i.e., $f_{c_1}(s'), \dots, f_{c_j}(s')$* ”. To achieve this task, we use an efficient technique for computing $f_{c_1}(s'), f_{c_2}(s'), \dots, f_{c_j}(s')$. This technique is based on a bitset representation of subsequences and the use of Boolean operations. Many studies [2, 3] show that using a bitset representation ensures an efficient trade-off between execution speed and

memory usage for mining sequential patterns from sequential data set. To date, there does not exist any approach that uses bitset to efficiently mine sequential classification rules from labeled sequential data set. Accordingly, we describe a bitset structure used for targeting the problems such as (i) how a sequence s' is a subsequence against all the sequences of \mathcal{D} and (ii) how to build the rule $\pi : s' \rightarrow f_{c_1}(s'), \dots, f_{c_j}(s')$ based on the sequence s' .

Definition 5 (Bitset) A bitset B is a sequence of bits which each takes the value 0 or 1. A bitset with k bits is called k -bitset, and B_i refers to the i^{th} bit of B .

This bitset is used to describe how a sequence is a subsequence of another one. Suppose we have two sequences $s = \langle s_1, \dots, s_{\ell_s} \rangle$ and $s' = \langle s'_1, \dots, s'_{\ell_{s'}} \rangle$ where $\ell_{s'} \leq \ell_s$. If the sequence $s' = \langle s'_1 \dots s'_{\ell_{s'}} \rangle$ is a **subsequence** of $s = \langle s_1 \dots s_{\ell_s} \rangle$, we represent it by a bitset with ℓ_s bits. It is defined as follows: Firstly, a bitset is initialized with ℓ_s zeros; then, if there exists a subsequence of s in form of $\langle s_{i_1}, s_{i_2} \dots, s_{i_{\ell_{s'}}} \rangle$ where $s_{i_z} = s'_z$ for all $z = 1 \dots \ell_{s'}$ and $1 \leq i_1 < i_2 < \dots < i_{\ell_{s'}} \leq \ell_s$, then the $i_{\ell_{s'}}^{th}$ bit is set to 1.

Example 2 The sequence $\langle eg \rangle$ is a subsequence of $\langle eefgbg \rangle$, which is presented by a bitset with 6 bits 000101. This bitset indicates how the sequence $\langle eg \rangle$ is a subsequence of $\langle eefgbg \rangle$, with a 1 being turned on in each final position where the sequence $\langle eg \rangle$ is a subsequence of $\langle eefgbg \rangle$.

Given a labeled sequential data set \mathcal{D} and a sequence s' , we need to describe how s' is a subsequence against all the sequences of \mathcal{D} . To achieve that we associate an array of n bitsets with s' where $n = |\mathcal{D}|$, denoted as $s'.arr$. The bitset $s'.arr[sid]$ describes how the sequence s' is subsequence of the sequence $s = \langle s_1, \dots, s_{\ell_s} \rangle$ where $(sid, s, c) \in \mathcal{D}$. *MiSeRe* firstly constructs an array of bitsets for each atomic sequence $s' = \langle s'_1 \rangle$. The bitset $s'.arr[sid]$ is generated as follows: $s'.arr[sid]$ is initialized with ℓ_{\max} zeros; then, we set the bit $s'.arr[sid]_i$ to 1 such that $s_i = s'_1$ where $(sid, s, c) \in \mathcal{D}$.

Example 3 Given the labeled sequential data set \mathcal{D} in Table 1, the atomic sequence $\langle a \rangle$ has an array of 4 bitsets, $\langle a \rangle.arr = [011000, 100000, 010000, 000000]$. $\langle a \rangle.arr[4]$ equals to 000000 as the item a does not appear in the sequence $\langle eefgbg \rangle$. While $\langle a \rangle.arr[1]$ equals to 011000 as the item a appears in the second and third item of the sequence $\langle baadg \rangle$. Figure 2 shows arrays of bitsets of all atomic sequences $\langle a \rangle$, $\langle b \rangle$, $\langle g \rangle$, $\langle d \rangle$, $\langle e \rangle$ and $\langle f \rangle$.

Given a sequence $s' = \langle s'_1, \dots, s'_{\ell_{s'}-1}, s'_{\ell_{s'}} \rangle$ where $\ell_{s'} \geq 2$, the bitset array $\langle s'_1, \dots, s'_{\ell_{s'}-1}, s'_{\ell_{s'}} \rangle.arr$ is generated based on two arrays, the bitset array of the sequence $\langle s'_1, \dots, s'_{\ell_{s'}-1} \rangle$ and the atomic sequence $\langle s'_{\ell_{s'}} \rangle$. Firstly, each bitset in $\langle s'_1, \dots, s'_{\ell_{s'}-1} \rangle.arr$ is transformed, such that the first bit in this bitset with value 1 is set to 0 and all bits after that are set to 1. We denote the new array by $\langle s'_1, \dots, s'_{\ell_{s'}-1} \rangle.\widehat{arr}$. Then, the array $\langle s'_1, \dots, s'_{\ell_{s'}-1}, s'_{\ell_{s'}} \rangle.arr$ is obtained by applying an *and* operation on each bitset of $\langle s'_1, \dots, s'_{\ell_{s'}-1} \rangle.\widehat{arr}$ and $\langle s'_{\ell_{s'}} \rangle.arr$.

Example 4 Given the labeled sequential data set \mathcal{D} in Table 1 and the sequence $\langle ab \rangle$. The bitset array of the sequence $\langle a \rangle$ is $\langle a \rangle.arr = [011000, 100000, 010000, 000000]$ and the sequence $\langle b \rangle$ is $\langle b \rangle.arr = [010000, 001000, 100010, 000010]$. To generate the bitset array of the sequence $s = \langle ab \rangle$, we firstly transform $\langle a \rangle.arr$ to $\langle a \rangle.\widehat{arr} = [001111, 011111, 001111, 000000]$. Then we apply an *and* operation $\langle a \rangle.\widehat{arr} \wedge \langle b \rangle.arr = [000000, 001000, 000010, 000000]$. Figure 3 illustrates a detailed process of generating the bitset array of the sequence $\langle ab \rangle$.

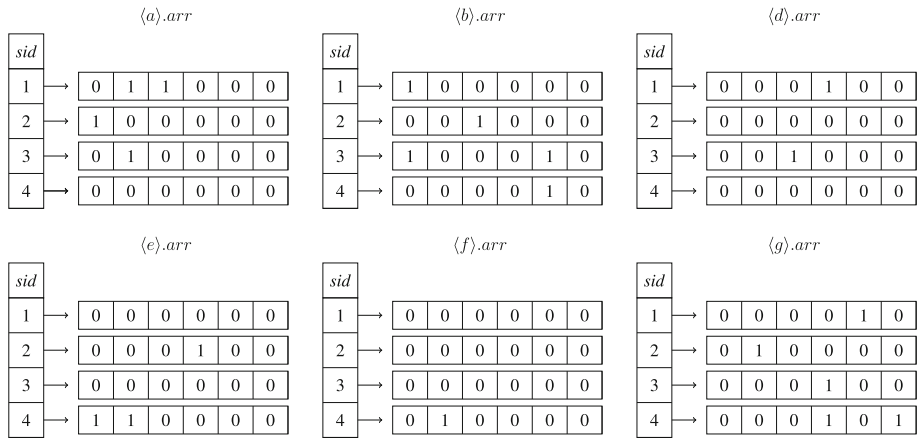


Fig. 2 Arrays of bitsets of all the atomic sequences

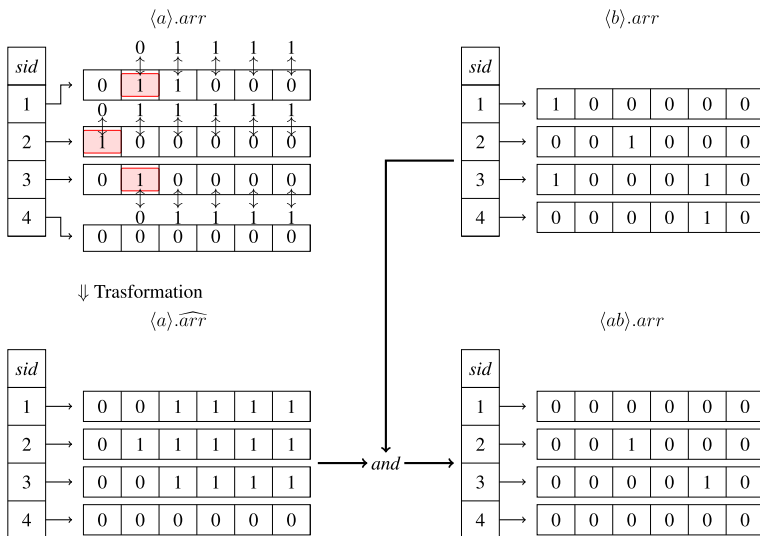


Fig. 3 An example explaining the process of generating the bitset array of the sequence $\langle ab \rangle$

We use $s'.arr$ to generate a new bitset $s'.bitset$ with n bits, which indicates whether the sequence s' is a subsequence of each sequence in \mathcal{D} . This bitset is generated as follows: If $s'.arr[i]$ equals to zeros bits, then the bit $s'.bitset_i$ is set to 0, otherwise, it is set to 1. With each class label $c \in \mathcal{C}$, we also associate a bitset $c.bitset$ with n bits to indicate which sequence in \mathcal{D} is labeled with c . Then, we perform an *and* operation on $s'.bitset$ and $c.bitset$ to compute $f_c(s')$. The value $f_c(s')$ is the number of bits equal to 1 in $s'.bitset \wedge c.bitset$.

Example 5 Given the labeled sequential data set \mathcal{D} in Table 1, we have $c_1.bitset = 1100$ and $c_2.bitset = 0011$. As $\langle ab \rangle.arr = [000000, 001000, 000010, 000000]$ we have $\langle ab \rangle.bitset = 0110$. The value of $f_{c_1}(\langle ab \rangle)$ is 1 because $\langle ab \rangle.bitset \wedge c_1.bitset = 0100$ while $f_{c_2}(\langle ab \rangle) = 1$ as $\langle ab \rangle.bitset \wedge c_2.bitset = 0010$. Figure 4 illustrates a detailed process of computing $f_{c_1}(\langle ab \rangle)$ and $f_{c_2}(\langle ab \rangle)$ using $\langle ab \rangle.arr$.

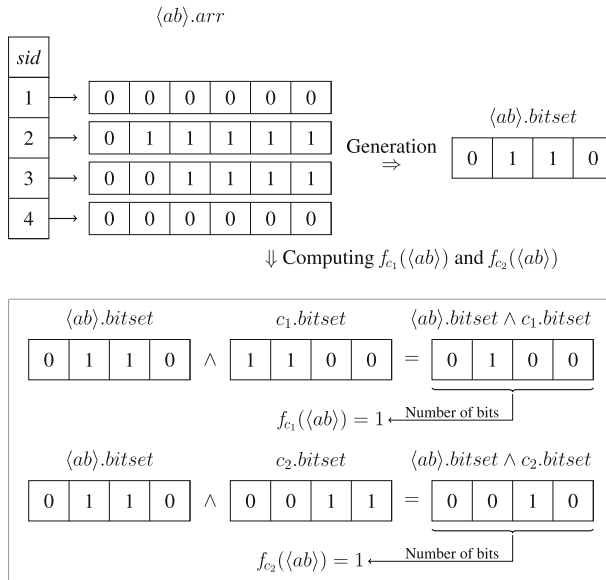


Fig. 4 An example explaining the process to compute $f_{c_1}(\langle ab \rangle)$ and $f_{c_2}(\langle ab \rangle)$ using $\langle ab \rangle.arr$

We benefit from the *BitSet*¹ class in Java in order to efficiently deal with the bitset. Using a **bitset** representation allows us to mine one rule π in time complexity $O(\ell_s \times n \times \log(n))$.

4.2 Classification procedure

We suggest to use the SCRs mined with MiSeRe as new features to recode the sequential data set \mathcal{D} into a binary transactional labeled data set. A new binary feature is created for each mined rule π and takes value 1 for an object (sid, s, c) if s is a supersequence of the body sequence of π , 0 otherwise. This procedure presents two advantages: (i) the full arsenal of existing classification algorithms can be applied to this new recoded data set; (ii) in some real-world data, the sequences are only a part of the description of data objects (together with, for example, classical categorical/numerical attributes), thus replacing the sequential part of the description of the data by relevant binary features enriches the data before using a classification algorithm.

5 Experiments

In this section, we empirically evaluate our approach on real-life data sets. *MiSeRe* is implemented in JAVA. The experiments are carried out on a 3.7 GHz Intel Core i7 computer with 32 GB of RAM Memory under Kubuntu 14. The goal of these experiments is to show the usefulness of the proposed classification system. A detailed interactive visualization of results as well as the JAVA code of MiSeRe is publicly available from [11] <http://www.misere.co.nf>. The experiments are designed to discuss the following questions:

¹ <http://docs.oracle.com/javase/7/docs/api/java/util/BitSet.html>.

Table 2 Experimental data sets description

Data	# Sequences	# Items	Longest sequence	# Classes	% Class majority
<i>aslbu</i>	441	140	58	7	36
<i>aslgt</i>	3493	47	197	40	19
<i>auslan</i>	200	12	24	10	10
<i>blocks</i>	210	8	24	8	14
<i>context</i>	240	54	272	5	21
<i>pioneer</i>	160	92	127	3	64
<i>skater</i>	530	41	260	6	21
<i>speed</i>	530	41	260	7	17
<i>ecoli</i>	106	16	28	2	50
<i>reuters</i>	5459	14529	533	8	51
<i>cade</i>	15,000	111,766	19,763	12	20

- Q1: Is *level* a stable and robust interestingness measure compared with classical measures? And does it avoid spurious patterns?
- Q2: What about the predictive performance of well-known classification algorithms on benchmark data recoded using SCRs mined with MiSeRe?
- Q3: Does *MiSeRe* mine the interesting rules with diversity?
- Q4: Does our method suffer over-fitting? What about spurious patterns?
- Q5: How does the predictive performance of our approach evolve w.r.t. the number of rules extracted? And, what about the time efficiency of *MiSeRe*?
- Q6: How does the predictive performance of our approach compare with state-of-the-art rule-based classifiers?

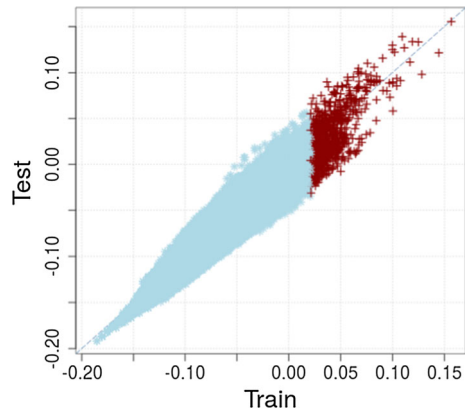
For empirical evaluation, we chose 11 real-life data sets: *aslbu*, *aslgt*, *auslan*, *blocks*, *context*, *pioneer*, *skater* and *speed* data are introduced in [39], *ecoli* data comes from UCI Repository [4], *reuters* and *cade* data,² are introduced in [8]. These data sets have a wide variety in the number of sequences, items, sequence length and classes as well as application domains. A brief description of these data sets is given in Table 2. We also evaluate our classification system on a large marketing database from the French Telecom company Orange containing sequential information about the behavior of 76,564 customers to predict the propensity of customer to churn.

5.1 Stability of the level criterion

To evaluate the stability and robustness of an interestingness measure, we perform train–test experiments. Each data set is divided into two parts: 50 % for training, i.e., learning w.r.t. an interestingness measure and 50 % for testing, i.e., evaluating the measure of the rules on the test data. Then, for each mined rule, train and test values are compared. We extract frequent sequential patterns from training data set by applying cSPADE [51] with a minimum support of 2 % and maximum gap of 2. Sequential classification rules are then generated from these patterns. We compute the *level* values of the mined rules for train and test sets as well as the values of three well-known measures: confidence, growth rate and lift.

² <http://web.ist.utl.pt/acardoso/datasets/>.

Fig. 5 Level values for mined rules in a train–test experiment for the skater data (the *red+* are the top 1000 highest level values of the mined rules from train data)



Notice that for the motivating example skater data of the introduction (Fig. 1), the *level* values computed for the mined rules are perceptibly more stable (closer to the diagonal) than confidence and growth rate as shown in Fig. 5. The same observations stand for the other data sets [11]. Figure 6 shows test values of confidence, growth rate, lift and level against train values of each mined rule pattern for *aslbu*, *pioneer*, *speed*, *auslan* and *ecoli* data—and the same conclusions hold.

To have a global view of the stability of the studied measures on the benchmark data sets, we study the rank agreement of the measure values in the train–test experiments. For a given data set and for each measure, we rank the mined rules according to their measure values in train and test data. Then, the agreement between train and test ranks is analyzed using Spearman correlation coefficient [40]. Figure 7 shows that *level* has the high train–test correlation (coefficient value near 1) and is stable, while the other measures have a weak correlation from train to test data and are thus unstable.

The robustness of the *level* measure is also studied with the help of the following experiment. For each data set, we randomly assign a class label $c \in \mathcal{C}$ to each sequence. As our method *MiSeRe* is controlled by a running time constraint, we run *MiSeRe* for 24 hours for all data sets with random labels. As a result, not one single rule could be extracted as all have a negative level value. Conversely, for most of the data sets, we still could find some sequential classification rules with high confidence, growth rate or lift. Thus, it can be concluded that **level** is a **robust measure**, it **discovers no spurious patterns** and **avoids overfitting**.

5.2 Predictive performance of our approach

To evaluate the predictive performance of our approach, we employ several standard classifiers on the benchmark data sets recoded using SCRs obtained with *MiSeRe*. We use Naïve Bayes (NB), Random Forest (RF), Decision Tree (C4.5), Support Vector Machine (SVM), lazy classifier IBk (a k -Nearest Neighbor) available from the Weka package [25]—all with default parameter values—and the Selective Naïve Bayes³ (SNB) [7]. The predictive performance results are all obtained with stratified tenfold cross-validation: *MiSeRe* operates only on the training data folds. Although *MiSeRe* is anytime and parameter-free, *for convenience and comparison purposes*, we set a number of rules to be extracted, say 2^{10} , i.e., 1024 rules.

³ <http://www.khiops.com>.

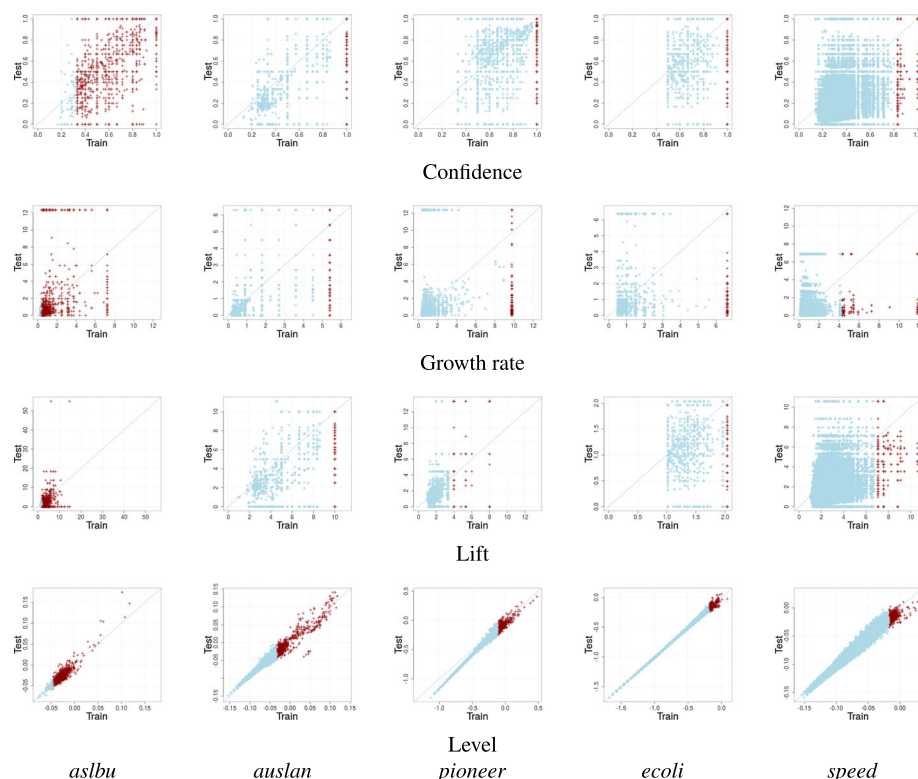


Fig. 6 Comparison of confidence and level values for sequential classification rules in a train–test experiment: 50% train/50% test for each data set (the *red+* are the top 1000 highest confidence (resp. growth rate, lift and level) values of the mined rules from train data)

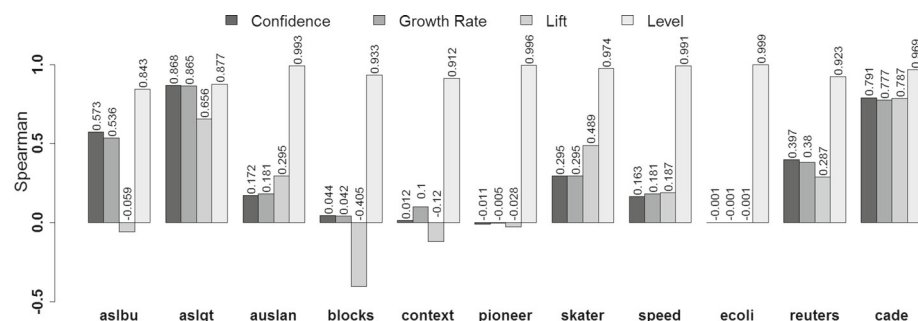


Fig. 7 Agreement between train rank and test rank of the mined rules according to measure values for benchmark data sets

The average accuracy results are reported in Fig. 8: a first look at the results says that SVM, RF and SNB often scores the best accuracy.

To confirm this first impression, we also apply the Friedman test and a post hoc Nemenyi test as suggested by [14] for comparisons of classifiers over multiple data sets (at 95% confidence level for both tests). The null hypothesis is rejected, meaning that the compared classifiers are not equivalent in terms of accuracy. The result of the Nemenyi test is represented

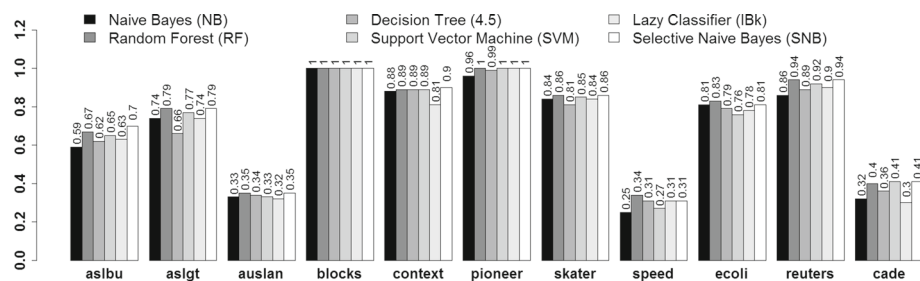
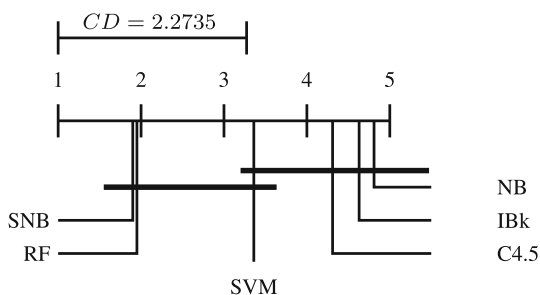


Fig. 8 Average accuracy results over tenfold cross-validation using MiSeRe coupled with several standard classifiers

Fig. 9 Critical difference of performance between various classifiers on data using extracted SCRs



by the critical difference (CD) chart shown in Fig. 9 with $CD \simeq 2.2735$ and where the mean rank of each classifier is plotted. Even if none of the six classifiers is singled out, the chart highlights two different groups of classifiers: {SVM, C4.5, IBk, NB} between which there is no statistical difference of performance, and {SNB, RF}, although they are not statistically better than SVM, they outperform the others. Thus, our recommendation is to use MiSeRe coupled with either SNB or RF. Since, SNB is Bayesian and parameter-free, meeting the characteristics of our framework, we will use SNB-MiSeRe for further inductive performance comparisons with state-of-the-art rule-based classifiers below. Results for MiSeRe coupled with other classifiers are available from [11].

5.3 Binary features versus numerical features

Our classification procedure is based on using the SCRs mined with *MiSeRe* as new features to recode the data into a binary transactional labeled data set. This new binary feature represents the presence or absence of the body of the rule as a subsequence in the object. However, in many applications such as biology and text mining, the body of the rule can appear several times as a subsequence in the same object. Using the number of times, the body of rule appears as a subsequence of an object to recode the data can effect on the predictive performance of our classifier. For this reason, we conduct further experiments to study whether using the binary features has a good predictive performance as compared to numerical features. The idea is to recode the sequential data set \mathcal{D} into a numerical transactional labeled data set, afterward a comparative study of the predictive performance of *SNB* on both binary and numerical features is presented. A new numerical feature is created for each mined rule π by taking the number of times the body of π appears as a subsequence of an object $(sid, s, c) \in \mathcal{D}$. For example, suppose that the sequence $s' = \langle ab \rangle$ is the body of the rule π and $s = \langle abcababadb \rangle$ is a sequence in \mathcal{D} . If the sequence s' appears four times as a

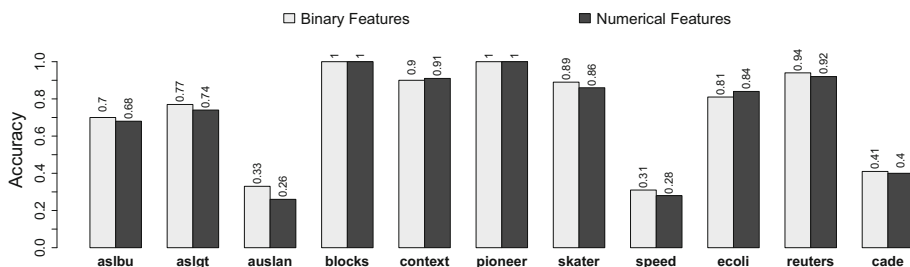


Fig. 10 Comparative study of the predictive performance of SNB on both binary and numerical features

subsequence of s (i.e., $s = \underbrace{ab}_{s'} c \underbrace{ab}_{s'} \underbrace{ab}_{s'} \underbrace{adb}_{s'}$), the sequence s is recoded by using π

into 4. Figure 10 shows the average accuracy results per data set obtained with stratified tenfold cross-validation after applying SNB on a set of binary and numerical features. Here it can be noticed that for most of the data sets the predictive performance of SNB on binary features outperforms on the numerical features. The binary feature construction process is certainly the most straightforward but has also shown good predictive performance for most of the data sets and in several studies [9, 22]. For this reason, we will use the binary recoding in all the experiments.

5.4 Effectiveness and efficiency of MiSeRe

Our mining method *MiSeRe* is controlled by a running time constraint during which a certain number of rules are mined. This section studies the predictive performance of *SNB-MiSeRe* classification system w.r.t. the number of extracted rules. Figure 11a shows the performance in terms of accuracy of *SNB-MiSeRe* based on ρ rules ($\rho = 2^\alpha$; $\alpha \in [0; 14]$). From this figure, it can be observed that the predictive performance increases with the number of rules. Then, it becomes rather stable beyond few hundred of rules. Finally, we can conclude that the accuracy generally reaches a **plateau** with about a **few hundreds** of mined rules for most of the data sets.

Now, we study the scalability of the *MiSeRe* algorithm. Figure 11b reports the runtime in logscale of the algorithm based on the number of mined rules for each data set. For most of the data sets, mining a thousand rules is managed in less than **80 seconds**. For ecoli data set, only 53 rules are extracted with positive level. Thus, *MiSeRe* has a fixed execution time of around 60 seconds (beyond 53 rules) for ecoli regardless of the desired number of extracted rules. For cade data set, *MiSeRe* has a constant execution time around 20 seconds until 2048 mined rules. As cade is a text data set, it has 111776 distinct words; thus, all the 2048 mined rules have a body, which is made of one single item. The execution time is data preparation time (Lines 1–2 in Algorithm 1).

5.5 Diversity of *MiSeRe*

Another experiment was conducted to study the diversity of the mined rules. The maximum number of rules to be mined by *MiSeRe* over each data set was set to 1024. The objective of the experiment is to compute (1) how many times *MiSeRe* randomly chooses a sequence to start local exploration and (2) how many times *MiSeRe* finds a rule with positive level (Line

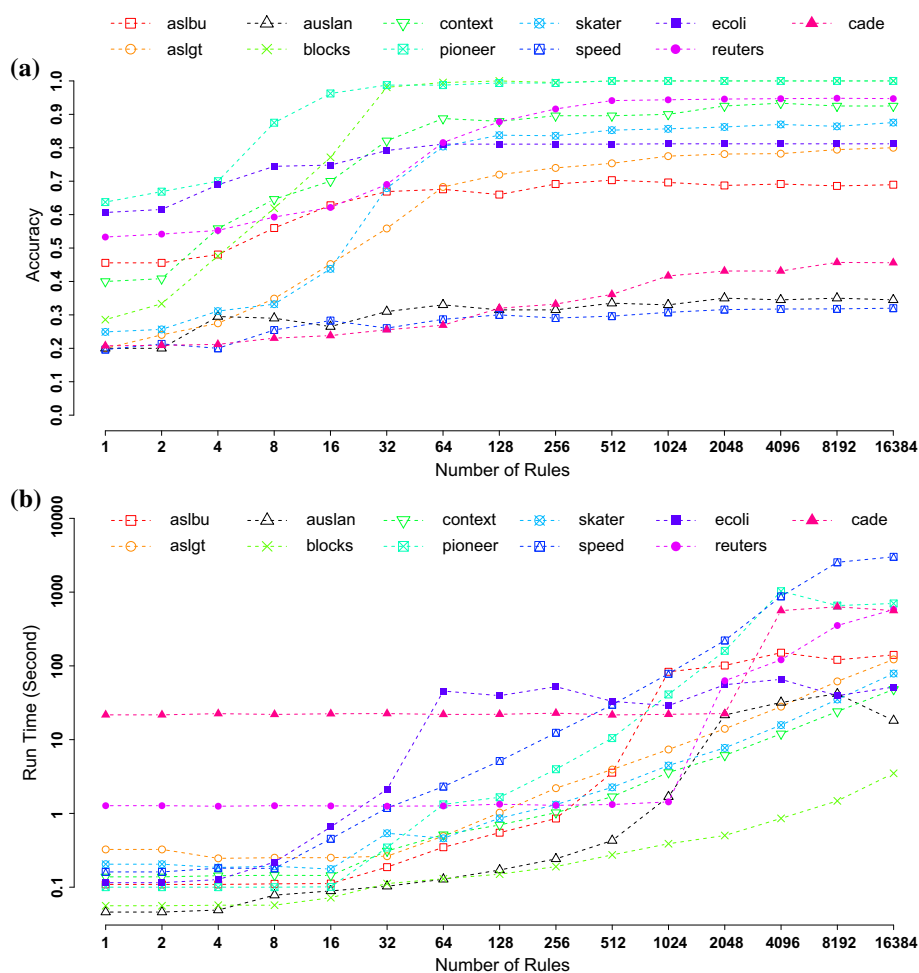


Fig. 11 Evolution of accuracy/execution time results per data set w.r.t. number of rules mined

10 in Algorithm 1). Table 3 gives a summary of these results: $\#randomstart$ is the number of random starts (Line 5 in Algorithm 1), $\%success$ is the percentage of the cases where at least one rule is found, $mean(\#rules)$ and $stdev(\#rules)$ are the mean and the standard deviation of the number of rules found for each starting sequence, while $\#distinctrules$ is the total number of distinct mined rules from the main loop (Lines 4–12 in Algorithm 1, i.e., the rules whose body is made of more than one single item) and $\#ruleswithoneitem$ is the number of rules whose body is made of single item (Line 3 in Algorithm 1).

For cade and reuters, all the 1024 mined rules have a body, which is made of one single item; thus, *MiSeRe* does not enter in the main loop. For the rest of the data sets, the rules are more or less easy to mine. For aslgt, blocks, context and skater, we notice that *MiSeRe* mines the rule with diversity since most random starting sequences produce few rules. In the easiest data set (e.g., blocks), 100 % of the random starts produce rules, and only 236 starts are sufficient to obtain the required rules. On the contrary, for aslbu, auslan, pioneer, speed and ecoli, we have the case where *MiSeRe* has difficulty in finding 1024 rules as the values

Table 3 Summary of the performance of MiSeRe

Data	#Random start	%Success	SD (#rules)	Mean (#rules)	#Distinct rules	#Rules with one item
<i>aslbu</i>	128,49	27	0.65	0.36	566	31
<i>aslgt</i>	205	99	2.01	4.92	977	47
<i>auslan</i>	12,278	66	2.5	2.6	1020	4
<i>blocks</i>	236	100	2.39	6.2	1017	7
<i>context</i>	217	94	2.84	4.69	998	26
<i>pioneer</i>	13,616	18	0.46	0.21	977	29
<i>skater</i>	456	89	1.66	2.33	1003	21
<i>speed</i>	9979	17	0.47	0.2	1011	13
<i>ecoli</i>	6004	6	0.25	0.06	50	3
<i>reuters</i>	0	–	–	–	–	1024
<i>cade</i>	0	–	–	–	–	1024

#randomstart are so high with potentially less proportion of successful starting. In the worst case (e.g., *ecoli*), the required number of rules could not be mined with the time constraints.⁴ Thus, it can be concluded that our randomized strategy allows us to mine interesting rules with diversity which is highly dependent on the data.

5.6 SNB-MiSeRe versus state of the art

Our classification system *SNB-MiSeRe* consists of two steps: pattern mining and classification using *MiSeRe* and *SNB*. This section presents a comparative study of the performance of *MiSeRe* and state-of-the-art competitive rule mining algorithms with several classification methods. We compare the set of rules mined by *MiSeRe* with five baseline algorithms: (1) cSPADE [51], a method for sequential rule mining under different types of constraints, (2) SCII [53], an algorithm for mining sequence classification rules based on interesting itemsets, (3) Gokrimp [30], (4) SQS [46], algorithms for mining sequential patterns by using the minimum description length (MDL) principle, and (5) DeFFeD [26], a very recent method to mine δ -free sequential patterns.

For providing a comparison between all the approaches described above and *MiSeRe*, experiments were conducted over a stratified tenfold cross-validations on the benchmark data sets. As all the compared approaches are unsupervised methods, we apply them independently per class for the related subpart of the data set. The parameters were set for each algorithm as follows: For the cSPADE algorithm, the minimum support threshold is set to 2 %, the maximum gap is set to 3 and the maximum length is set to 6. For the SCII algorithm, the minimum support threshold is set to 2 %, the minimum interestingness threshold is set to 5 %, the maximum length is set to 6 and the minimum confidence threshold is set to 60 %. The DeFFeD algorithm requires the δ -freeness and the minimum support threshold to be defined. For all the data sets when the minimum support threshold is set less than 10 % and δ -freeness equals to 10, we have problems with the execution of DeFFeD due to the space in memory taken for the algorithm. Due to this issue, a different support threshold is used for

⁴ In case if *MiSeRe* has difficulty in finding the required 1024 rules, another constraint based on time is fixed. If *MiSeRe* cannot find any interesting rule 5 minutes after the generation of last interesting rule, the algorithm is terminated.

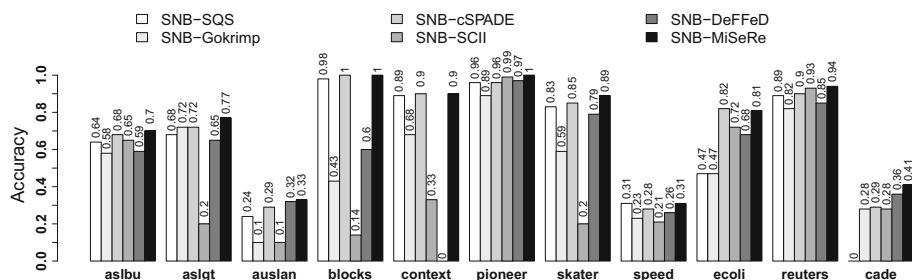


Fig. 12 Comparisons of accuracy results w.r.t. various several extraction methods

each data set. The minimum support threshold is set to 10 % for auslan and cade data, 40 % for reuters data, 55 % for aslbu, blocks, ecoli, pioneer, speed and scatter data, 70 % for context data and 80 % for aslgt data. For context data, we were not able to obtain any results as the minimum support threshold is very high i.e., 70 %. If the support threshold is set to a smaller value, then it causes *memory overload* with DeFFeD. The SQS and GoKrimp algorithms are parameter-free. The GoKrimp algorithm is very fast on all the data sets, while SQS is only fast for small data sets with short sequences. For instance, SQS takes more than 54 hours to complete the extraction process from reuters data, while on cade data, we had to stop SQS after 168 hours without obtaining any results.

Afterward, these algorithms extract sequential rules from each training data. Then, we employ six classifiers: Naïve Bayes (NB), Random Forest (RF), Decision Tree (C4.5), Support Vector Machine (SVM), lazy classifier IBk and the Selective Naïve Bayes (SNB) on the benchmark data sets recorded using sequential classification rules obtained with *MiSeRe*, cSPADE, SCII, Gokrimp, SQS and DeFFeD.

Figure 12 shows the average accuracy results per data set obtained with stratified tenfold cross-validation when we combine SNB with all the extraction methods. The difference of performance between *SNB-MiSeRe* and other methods is clearly noticeable because *SNB-MiSeRe* always has the highest accuracy. Similar observations are made for the other classifiers coupled with all the extraction methods [11].

We apply the Friedman test coupled with a post hoc Nemenyi test (at 95 % confidence level for both tests). The result of this test is represented by the critical difference chart shown in Fig. 13 with computed $CD \simeq 2.2735$. We observe that *MiSeRe* with all the classifiers has the best rank compared with all other extraction methods. Although *MiSeRe* is not statistically singled out, it has remarkable lead over DeFFeD, SCII, SQS and Gokrimp. On the other hand, *MiSeRe* has minor lead over cSPADE.

We also run all the extraction methods over all data sets with random labels (generated in Sect. 5.1). As a result, for all the data sets, the competitive mining methods can extract sequential classification rules while not a single rule can be extracted after running *MiSeRe* for 24 hours. Thus, it can be concluded that *MiSeRe* avoids spurious patterns contrary to all competitive mining methods.

We conclude that *MiSeRe* performs better against other methods for all classifiers because it has the following two advantages: (i) *MiSeRe* mines with diversity a subset of rules; (ii) *MiSeRe* uses a robust measure *level*, which is highly resilient to spurious patterns. To present the power of these two advantages, we conduct the following experiment. We compute the *level* values of the mined rules with cSPADE; then, we select the rules with positive level values. We employ then SNB as a classifier on the selected mined rules. Afterward, we

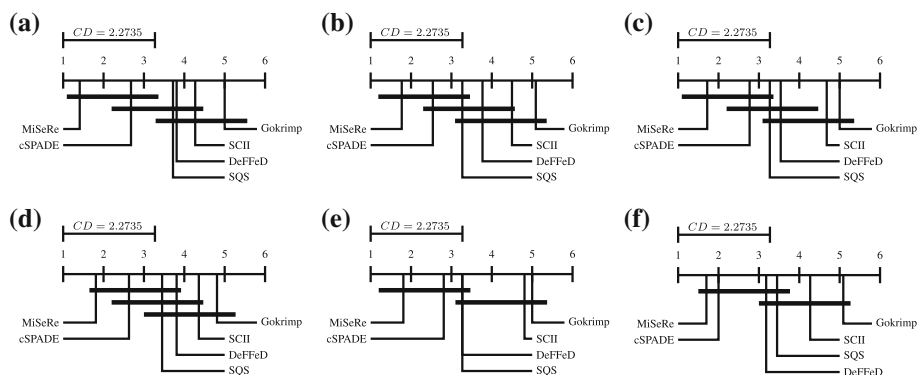


Fig. 13 Comparison of all the extraction methods against each other coupled with six classifiers: **a** Selective Naïve Bayes (SNB), **b** Random Forest (RF), **c** Support Vector Machine (SVM), **d** Decision Tree (C4.5), **e** Lazy Classifier (IBk) and **f** Naïve Bayes (NB) with the Nemenyi test. Groups of extraction methods that are not significantly different (at 95% confidence level) are connected

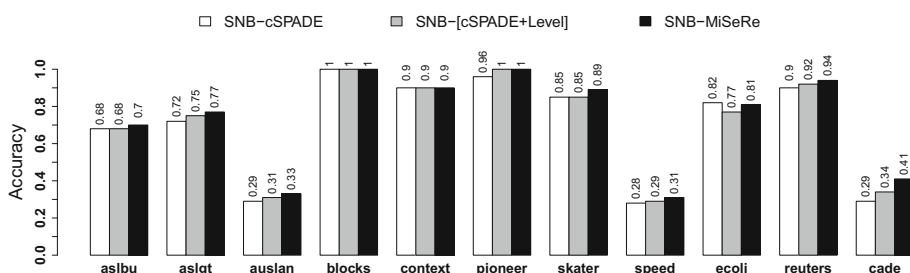


Fig. 14 Comparison of the predictive performance of the rules mined with *MiSeRe* and *cSPADE*

present a comparative study of the predictive performance of *SNB* on the rules mined with *cSPADE*, the selected rules (with positive level values) mined with *cSPADE* and the rules mined with *MiSeRe*. Figure 14 shows the average accuracy results per data set obtained with stratified tenfold cross-validation after applying *SNB*. The results show that, in most of the data sets, using *level* to evaluate the rules mined with *cSPADE* improves its classification results, while *MiSeRe* is still the best as it employs an instance-based randomized strategy promoting diversity mining and it uses a *level* criterion for evaluating the interest of the mined rules.

Another experiment was conducted for comparing the performance of *SNB-MiSeRe* classifier with four state-of-the-art competitive rule-based classifiers: SCII Match, SCII CBA [53], BayesFM [33] and CBS [47]. The experiments were performed with parameters as indicated in the original papers. Average accuracy results per data set obtained with stratified tenfold cross-validation are reported in Fig. 15. It can be conjectured that there is a difference of performance between *SNB-MiSeRe* and the other competitors as *SNB-MiSeRe* always scores the highest accuracy.

To see this difference more clearly, we also apply the Friedman test coupled with a post hoc Nemenyi test (at 95% confidence level for both tests). The result of this test is represented by the critical difference chart shown in Fig. 16 with computed $CD \simeq 1.8392$. In Fig. 16 (a), we observe that *SNB-MiSeRe* has the best rank compared with all the contenders. Even if *SNB-MiSeRe* is not statistically singled out, it gets a significant advantage on SCII Match,

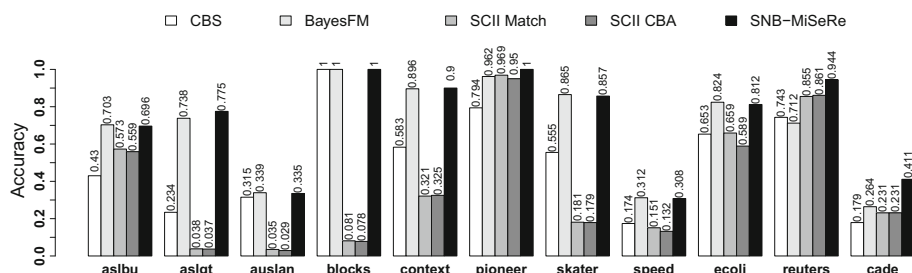


Fig. 15 Comparisons of accuracy results w.r.t. various state-of-the-art rule-based classifiers

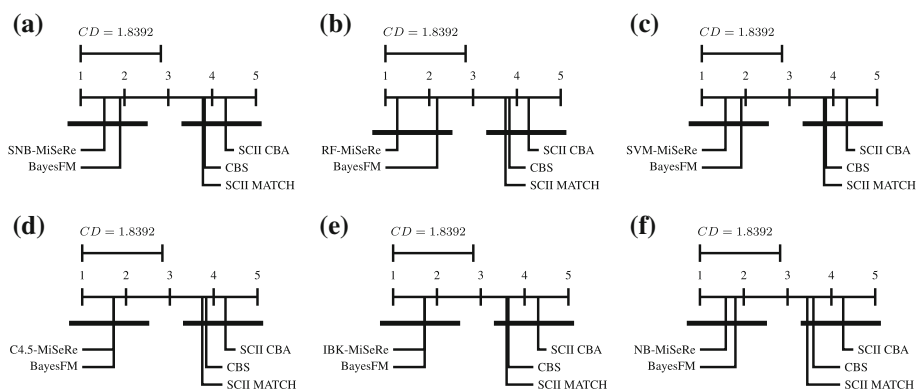


Fig. 16 Comparison of *MiSeRe* with all classifiers against the state-of-the-art competitive rule-based classifiers. Groups of rule-based classifiers that are not significantly different (at 95 % confidence level) are connected

SCII CBA and CBS, whereas BayesFM does not get this advantage. The same observations stand for the other classifiers coupled with *MiSeRe* (as shown in Fig. 16).

5.7 Experiments on a real-world marketing data set

We also carried out experiments on a large marketing database from the French Telecom company Orange containing sequential information about the behavior of 76,564 customers to predict their propensity to churn. Each sequence represents a time-ordered set of actions (or events) between one customer and the company, e.g., visiting the official webpage of the company, visiting its Franchise, calling technical support and buying a new product or service. These customers are classified into three classes. The first class includes 4135 customers who firstly estimate the termination costs, and then, they terminate their contract with the company. The second class consists of 4979 customers who terminate their contract after unblocking their SIM cards, while the third class contains 67,450 customers still having the contract. The goal is to build a model able to classify new customer as accurately as possible. This data set contains 785 distinct actions, the longest sequence is a customer having 7487 actions, while the median length of sequences is 29 actions. Figure 17 shows the distribution of the length of sequences for the number of customers greater than 10.

We experiment with *MiSeRe* and its competitors using a stratified tenfold cross-validation. We had to stop SQS after more than two weeks without obtaining any results as SQS is not efficient on big data sets with long sequences. For this reason, we do not include SQS in this

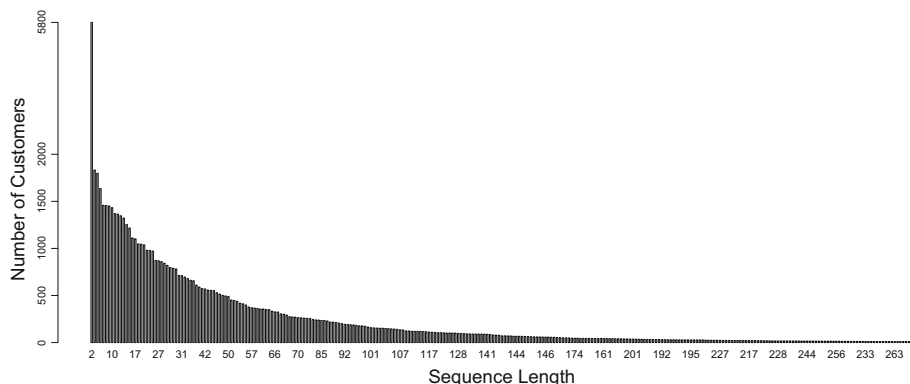


Fig. 17 Distribution of the length of sequences

Fig. 18 Comparison of AUC results for real-world marketing data set



Table 4 Comparison of accuracy results for real-world marketing data set with various state-of-the-art competitive rule-based classifiers

Accuracy				
<i>SNB-MiSeRe</i>	BayesFM	SCII Match	SCII CBA	CBS
0.89	0.67	0.68	0.31	0.16

comparative study. As the distribution of the classes defined above is unbalanced, we use the Area Under the ROC Curve (AUC) to evaluate our results.

Figure 18 reports the heat map of the AUC values for *MiSeRe* and its competitors. The red color scale represents the predictive performance of each classification system. The deviation around mean AUC values lies between 10^{-2} for the highest AUC value and 10^{-3} for the lowest one. From this figure, it can be observed that compared to the other mining methods, *MiSeRe* scores the highest AUC whatever the classifier. It can be also seen that *MiSeRe* with *SNB*, *NB* and *RF* are among the first five best couplings. We also compare the performance of *SNB-MiSeRe* classifier with the state-of-the-art competitive rule-based classifiers over our real-world marketing data set. We choose accuracy for evaluation because the competitors provide only this measure. Table 4 reports accuracy values for *SNB-MiSeRe* and its competitors. It can be observed that *SNB-MiSeRe* has the highest accuracy that gives it a lead over the compared state-of-the-art competitive rule-based classifiers.

Table 5 Characteristics of compared pattern-based sequence classification methods

	Classif method	Extraction	Parameters
BayesFM [33]	Recode+NB	Level-wise	Freq., conf., window, length
CBS [47]	Scoring/ranking	Level-wise	Freq.
SCII [53]	Scoring/ranking	Level-wise	Freq., conf., length
DeFFeD [26]	Max.Ent. Model	Level-wise	Freq., δ -freeness
GoKrimp [30]	Recode+Classif.	Greedy compress.	None
SQS [46]	None	Greedy compress.	None
MiSeRe	Recode+Classif.	Diversity	none

6 Related work and discussion

The topic of sequence classification has been considered under various angles: mainly, (i) through Hidden Markov Models (HMMs) [32] and constructing generative sequence classifiers, (ii) using string kernels as the driving element of Support Vector Machines [34,36], and (iii) combining pattern mining and classification [54]. Xing et al. [50] suggest a brief survey on sequence classification and cover the first two angles. We now focus on the third angle, which is the most related to our work.

One of the first pattern-based classifiers is *BayesFM*, introduced by Lesh et al. [33] and which consists of two steps: Firstly, it mines sequential rules with the help of cSPADE [51], and then, the rules are chosen using a frequency-confidence approach. Secondly, it uses the selected rules as an input for Naive Bayes classification method. As mentioned in [13], *BayesFM* is “unable to extract high-order, long sequential patterns efficiently”—due its level-wise search. That is why our approach takes advantage on BayesFM for the cade and reuters data sets (text data are made of long sequences and large alphabet). However, except for large data sets, BayesFM is very effective compared to subsequent works, as we show in our experiments: CBS [47], SCII [53], DeFFeD [26], SQS [46] and GoKrimp [30]. We report the main characteristics of each approach in Table 5.

Tseng et al. [47] introduced Classify-By-Sequence (CBS) algorithm, which integrates sequential pattern mining with probabilistic induction. CBS mines sequential patterns as sequence rules, keeping the rules with higher frequencies, and then determines the class label of new objects using scoring based on class support or pattern length. Zhou et al [53] introduced a sequence classification method SCII based on frequent cohesive itemsets. In SCII, firstly the interesting itemsets are mined in each class of sequences based on their support and cohesion. Then, the authors propose two different classifiers SCII CBA and SCII MATCH. SCII CBA ranks the rules using their confidence, interestingness and size, while SCII MATCH ranks the rules using the product of the confidence and the cohesion of the rule. Then, the new sequence will be classified into a class of the first matched rule. Using condensed representations of frequent sequences, namely δ -free sequential patterns, Holat et al [26] suggest DeFFeD. The proposed algorithm requires two parameters i.e., frequency and delta δ . Then, they present the utility of δ -free patterns as features in a supervised text classification as well as early prediction task.

Other approaches employ the MDL principle to mine sequential data. Tatti et al [46] propose the SQS method and Lam et al. [30] the Gokrimp method. Both methods mine sequential patterns by compressing the data using a MDL scheme, and then, Gokrimp creates

new features per mined sequential rule as input for a standard classifier. The major difference between SQS and Gokrimp is the encoding scheme. The main differences in our work are as follows. The SQS and Gokrimp methods focus on the encoding scheme, while our approach states the problem as Bayesian Maximum A Posteriori model selection approach, with a specific hierarchical prior distribution. While both approaches can be interpreted directly or indirectly as MDL, they result in very different encoding schemes. Our model is defined to be a single rule with a randomized algorithm to mine a sample of rules, while the SQS and Gokrimp model consist of a set of sequential patterns. Moreover, *MiSeRe* is a supervised method for mining sequential classification rules from a labeled sequential data set, while SQS and Gokrimp are unsupervised methods that mine sequential patterns from sequential data set, applied independently per class for the related subparts of the data set. To summarize, the SQS and Gokrimp methods encode a set of rules and are unsupervised, with a focus on MDL, while our approach is supervised and encodes one single rule at a time, with a focus on Bayesian model selection and explicit prior definition.

It is also worth mentioning other recent relevant works on sequential pattern mining classification purpose: [27] mines contrasting sequential patterns, i.e., patterns that are frequent in the positive class and infrequent in the negative class. Frequency, infrequency and gap parameters have to be carefully set to find useful patterns. Baralis et al. [5] suggest to mine compact representations of sequential classification rule-based condensed representations such as closed and generator patterns. Zaki et al. [52] combine frequent sequence mining and hidden Markov models. Frequency and gap parameters have to be set, and sequential pattern of size 2 is mined. [23] embraces the theory of relevance [31] to mine relevant sequential generator patterns of limited length. In a very recent work, Fradkin & Mörchen [20] extend the BIDE algorithm [49] for directly mining predictive sequential patterns with integration in a tree-based classifier [19].

7 Conclusion and future work

This paper focuses on the important problem of mining sequential rule patterns for classification purpose. We have introduced a space of rule pattern models and a prior distribution defined on this model space. We present a new interestingness measure (*level*) that allows us to naturally mark out interesting and robust classification rules. We develop a parameter-free algorithm that efficiently mines interesting and robust rules. Using the extracted rules as new features in a classification process has demonstrated strong predictive performance. The empirical experiments show that our system demonstrates highly competitive inductive performance compared with state-of-the-art rule-based classifiers while being highly resilient to spurious patterns. As future work, we plan to extend our approach for a labeled multidimensional sequential data set. On the other hand, we are also planning on proposing a parallel and distributed version of *MiSeRe* using Hadoop.

Appendix: Proof of Theorem 1 and 2

Here, we will present a complete Proof of Theorem 1, while theorem 2 can be proven in the same way.

Proof Given the cost of the null model in Eq. 5, the prior terms are neglected when the number of sequences n is very high. Therefore, when $n \rightarrow \infty$ the cost of the null model is written as:

$$\text{cost}(\pi_\emptyset) = \log(n!) - \sum_{i=1}^j \log(n_{c_i}!)$$

Using the approximation $\log(n!) = n(\log(n) - 1) + O(\log(n))$ based on Stirlings formula, the cost of the null model can be written as:

$$\begin{aligned} \text{cost}(\pi_\emptyset) &= n(\log(n) - 1) - \sum_{i=1}^j n_{c_i}(\log(n_{c_i}) - 1) + O(\log(n)) \\ &= n\log(n) - n - \sum_{i=1}^j n_{c_i}\log(n_{c_i}) + \sum_{i=1}^j n_{c_i} + O(\log(n)) \end{aligned}$$

Notice that $n = \sum_{i=1}^j n_{c_i}$. Therefore, we have:

$$\begin{aligned} \text{cost}(\pi_\emptyset) &= \sum_{i=1}^j n_{c_i}\log(n) - n - \sum_{i=1}^j n_{c_i}\log(n_{c_i}) + n + O(\log(n)) \\ &= - \sum_{i=1}^j n_{c_i} \left(\log(n_{c_i}) - \log(n) \right) + O(\log(n)) \\ &= n \times \left(- \sum_{i=1}^j \frac{n_{c_i}}{n} \log\left(\frac{n_{c_i}}{n}\right) \right) + O(\log(n)) \\ &= n \times \left(- \sum_{i=1}^j p(c_i) \log(p(c_i)) \right) + O(\log(n)) \\ &= n \times H(y) + O(\log(n)) \end{aligned}$$

References

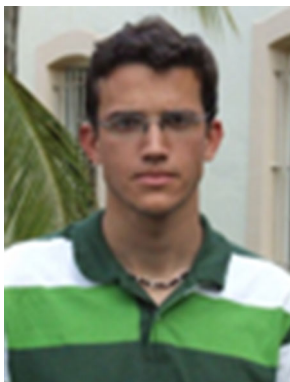
1. Agrawal R, Srikant R (1995) Mining sequential patterns. In: ICDE'95, pp 3–14
2. Aseervatham S, Osmani A, Viennet E (2006) Bitspade: a lattice-based sequential pattern mining algorithm using bitmap representation. In: Sixth International Conference on Data Mining, 2006. ICDM'06. IEEE, pp 792–797
3. Ayres J, Flannick J, Gehrke J, Yiu T (2002) Sequential pattern mining using a bitmap representation. In: KDD'02. ACM, pp 429–435
4. Bache K, Lichman M (2013) UCI machine learning repository. <http://archive.ics.uci.edu/ml>
5. Baralis E, Chiusano S, Dutto R, Mantellini L (2008) Compact representations of sequential classification rules. In: Data mining: foundations and practice, pp 1–30
6. Boullé M (2006) MODL: a Bayes optimal discretization method for continuous attributes. Mach Learn 65(1):131–165
7. Boullé M (2007) Compression-based averaging of selective naive Bayes classifiers. J Mach Learn Res 8:1659–1685
8. Cardoso-Cachopo A (2007) Improving methods for single-label text categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa
9. Cheng H, Yan X, Han J, Hsu CW (2007) Discriminative frequent pattern analysis for effective classification. In: IEEE 23rd international conference on data engineering, 2007. ICDE 2007. IEEE, pp 716–725
10. Coenen F, Leng PH (2007) The effect of threshold values on association rule based classification accuracy. Data Knowl Eng 60(2):345–360
11. Companion website (2015) MiSeRe: Mining sequential classification rules. <http://misere.co.nf>

12. Cover TM, Thomas JA (2006) Elements of information theory (Wiley series in telecommunications and signal processing). Wiley-Interscience, New York
13. Dafé G, Veloso A, Zaki M, Meira W Jr. (2015) Learning sequential classifiers from long and noisy discrete-event sequences efficiently. *Data Min Knowl Discov*, To appear
14. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
15. Deng K, Zaiane OR (2010) An occurrence based approach to mine emerging sequences. In: *DaWaK'10*, pp 275–284
16. Deshpande M, Karypis G (2002) Evaluation of techniques for classifying biological sequences. In: *PAKDD'02*, pp 417–431
17. Egho E, Gay D, Boullé M, Voisine N, Clérot F (2015) A parameter-free approach for mining robust sequential classification rules. In: 2015 IEEE international conference on data mining, *ICDM 2015*, Atlantic City, 14–17 Nov 2015, pp 745–750
18. Egho E, Raïssi C, Calders T, Jay N, Napoli A (2015) On measuring similarity for sequences of itemsets. *Data Min Knowl Discov* 29(3):732–764
19. Fan W, Zhang K, Cheng H, Gao J, Yan X, Han J, Yu PS, Verscheure O (2008) Direct mining of discriminative and essential frequent patterns via model-based search tree. In: *ACM SIGKDD'08*, pp 230–238
20. Fradkin D, Mörchén F (2015) Mining sequential patterns for classification. *Knowl Inf Syst*, To appear
21. Gay D, Boullé M (2012) A Bayesian approach for classification rule mining in quantitative databases. In: *ECML/PKDD'12*, pp 243–259
22. Gay D, Selmaoui N, Boulicaut J-F (2008) Feature construction based on closedness properties is not that simple. In: *Advances in knowledge discovery and data mining*. Springer, pp 112–123
23. Grosskreutz H, Lang B, Trabold D (2013) A relevance criterion for sequential patterns. In: *ECML/PKDD'13*, pp 369–384
24. Grünwald PD, Myung IJ, Pitt MA (2005) *Advances in minimum description length: theory and applications*. MIT press, Cambridge
25. Hall MA, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. *SIGKDD Explor* 11(1):10–18
26. Holat P, Plantevit M, Raïssi C, Tomeh N, Charmois T, Crémilleux B (2014) Sequence classification based on delta-free sequential patterns. In: *ICDM'14*, pp 170–179
27. Ji X, Bailey J, Dong G (2005) Mining minimal distinguishing subsequence patterns with gap constraints. In: *IEEE ICDM'05*, pp 194–201
28. Jorge AM, Azevedo PL, Pereira F (2006) Distribution rules with numeric attributes of interest. In: *PKDD'06*, pp 247–258
29. Lam HT, Moerchen F, Fradkin D, Calders T (2012) Mining compressing sequential patterns. In: *SDM'12*, pp 319–330
30. Lam HT, Mörchén F, Fradkin D, Calders T (2014) Mining compressing sequential patterns. *Stat Anal Data Min* 7(1):34–52
31. Lavrac N, Gamberger D, Jovanoski V (1999) A study of relevance for learning in deductive databases. *J Log Program* 40(2–3):215–249
32. Lawrence R (2002) A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE* 77(2):419–444
33. Lesh N, Zaki MJ, Ogihara M (1999) Mining features for sequence classification. In: *ACM SIGKDD'99*, pp 342–346
34. Leslie CS, Eskin E, Weston J, Noble WS (2002) Mismatch string kernels for SVM protein classification. In: *NIPS'02*, pp 1417–1424
35. Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In: *ACM SIGKDD'98*, pp 80–86
36. Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Christopher JCH (2002) Watkins. Text classification using string kernels. *J Mach Learn Res* 2:419–444
37. Mannila H, Toivonen H (1996) Multiple uses of frequent sets and condensed representations (extended abstract). In: *KDD'96*, pp 189–194
38. Ming L, Paul V (2013) *An introduction to Kolmogorov complexity and its applications*. Springer Science & Business Media, New York
39. Mörchén F, Ultsch A (2007) Efficient mining of understandable patterns from multivariate interval time series. *Data Min Knowl Discov* 15(2):181–215
40. Myers JL, Well AD (2003) *Res Des Stat Anal*. Lawrence Erlbaum Associates, New Jersey
41. Rissanen J (1978) Modeling by shortest data description. *Automatica* 14(5):465–471
42. Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput Surv* 34(1):1–47
43. Shannon CE (2001) A mathematical theory of communication. *ACM SIGMOBILE Mob Comput Commun Rev* 5(1):3–55

44. She R, Chen F, Wang F, Ester M, Gady FL, Brinkman FSL (2003) Frequent-subsequence-based prediction of outer membrane proteins. In: ACM SIGKDD'03, pp 436–445
45. Tan P-N, Kumar V (2002) Discovery of web robot sessions based on their navigational patterns. *Data Min Knowl Discov* 6(1):9–35
46. Tatti N, Vreeken J (2012) The long and the short of it: Summarising event sequences with serial episodes. In: ACM SIGKDD'12, pp 462–470
47. Tseng VS, Lee CH (2005) CBS: a new classification method by using sequential patterns. In: SDM'05, pp 596–600
48. Vitányi P, Li M (2000) Minimum description length induction, Bayesianism, and Kolmogorov complexity. *IEEE Trans Inf Theory* 46(2):446–464
49. Wang J, Han J (2004) BIDE: efficient mining of frequent closed sequences. In: ICDE'04, pp 79–90
50. Xing Z, Pei J, Keogh EJ (2010) A brief survey on sequence classification. *SIGKDD Explor* 12(1):40–48
51. Zaki MJ (2000) Sequence mining in categorical domains: incorporating constraints. In: CIKM'00, pp 422–429
52. Zaki MJ, Carothers CD, Szymanski BK (2010) VOGUE: a variable order hidden Markov model with duration based on frequent sequence mining. *TKDD*, 4(1)
53. Zhou C, Cule B, Goethals B (2013) Itemset based sequence classification. In: ECML/PKDD'13, pp 353–368
54. Zimmermann A, Nijssen S (2014) Supervised pattern mining and applications to classification. In: Frequent pattern mining, pp 425–442



Elias Eggho is currently a research engineer in Profiling and Data Mining team at Orange Labs (France Télécom-Research and Development). He received his Ph.D. in Computer Science from University of Lorraine, Nancy, France, in LORIA-INRIA Nancy Grand Est laboratory in July 2014. His research majorly focuses on developing novel data mining techniques with a special interest in mining sequential patterns for detection and classification of sequential data.



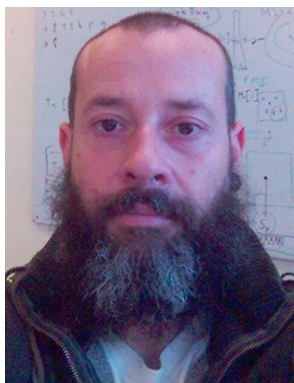
Dominique Gay is currently an Assistant Professor at Université de La Réunion. From 2010 to 2015, he was a research engineer in Profiling and Data Mining team at Orange Labs (France Télécom—Research and Development). In November 2009, he received a Ph.D. degree in Computer Science from Université de la Nouvelle-Calédonie (Nouméa, New-Caledonia) and Institut National des Sciences Appliquées (Lyon, France). His research themes are about Data Mining and Machine Learning with a special interest for local pattern mining and its use for complex data classification purpose.



Marc Boullé was born in 1965 and graduated from Ecole Polytechnique (France) in 1987 and Sup Télécom Paris in 1989. Currently, he is a Senior Researcher in the data mining research group of Orange Labs. His main research interests include statistical data analysis, data mining, especially data preparation and modeling for large databases. He developed regularized methods for feature preprocessing, feature selection and construction, correlation analysis, model averaging of selective naive Bayes classifiers and regressors.



Nicolas Voisine was born in 1972 and obtained a Ph.D. in image and signal processing from the University of Rennes in 2002. During his one-year post doc in ITI Greece his main research interest was video classification. Since 2005, he has been working in the R&D Division of France Tlcom where he became a senior expert in data mining. His main research interests include statistical data analysis, data mining, especially modeling for large databases. He developed regularized methods for Decision Tree and sequence mining.



Fabrice Clérot is currently head of the Profiling and Data mining research team at Orange Labs (France Télécom—Research and Development). His research interests are the application of statistical methods in various areas of interest for telecommunication companies, areas which moved from the semiconductor physics to telecommunication network modeling, traffic prediction, data mining, stream mining and reinforcement learning for the optimization of online targeted advertising and personalized customer relationship management.