A parameter-free approach for mining robust sequential classification rules

Elias Egho*, Dominique Gay*, Marc Boullé*, Nicolas Voisine* and Fabrice Clérot*
*Orange Labs
2, avenue Pierre Marzin, F-22307 Lannion Cédex, France
Email: firstname.lastname@orange.com

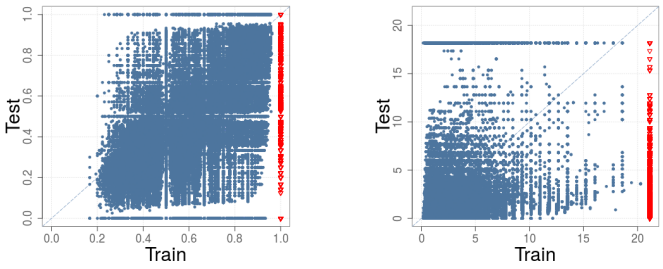
***Abstract*—Sequential data is generated in many domains of science and technology. Although many studies have been carried out for sequence classification in the past decade, the problem is still a challenge; particularly for pattern-based methods. We identify two important issues related to pattern-based 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 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.**

I. Introduction

Sequence classification [1] has many real-world applications in a broad range of domains, such as biology [2], [3], text mining [4] or web mining [5]. 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, that 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) [6], several rule-based approaches have been suggested for sequence classification. Generally, pattern-based classification methods [7] 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 [8], [9], [10] or the mined rules are directly used as new features in a classification process [11], [12], [13]. 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.

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, 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 [14].

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 figure 1, we plot test values of confidence (resp. growth rate) against train values of each mined rule pattern (one point per pattern) for the skater data set [15]. We observe very blurred scatter plots, meaning that interestingness measures 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 when evaluated on test data. Thus, it could be misleading to bet on such rules for classifying new incoming objects.



(a) Confidence (b) Growth rate

Fig. 1. Comparison of confidence (resp. growth rate) values for sequential classification rules in a train-test experiment: 50% train / 50% test for the skater data set.

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:

Towards 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 [16] and classification rule mining in transactional data sets [17]. 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.

A parameter-free approach for mining sequential classification rules. We discuss and present a new algorithm **MiSeRe** for ***Mi**ning **Se**quential Classification **Rule**s*. The main features of MiSeRe are: (i) it is user parameter-free, (ii) it employs an instance-based randomized strategy that promotes diversity mining, (iii) it uses a bitset representation and the Boolean operations, to efficiently mine the sequential classification rules and (iiii) it is anytime – the more time the user grants to the task, the more it learns.

To validate our contributions, we perform an extensive experimental evaluation on a variety of datasets, 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.

II. Preliminaries

Let $\mathcal{I} = \{e_1, e_2, \cdots, e_m\}$ be a finite set of m distinct items. A **sequence** s over $\mathcal{I}$ is an ordered list $s = \langle s_1, \cdots, s_{\ell_s}\rangle$, where $s_i \in \mathcal{I}$; $(1 \leq i \leq \ell_s, \ell_s \in \mathbb{N})$. A sequence $s' = \langle s'_1 \cdots s'_{\ell_{s'}}\rangle$ is a **subsequence** of $s = \langle s_1 \dots s_{\ell_s}\rangle$, denoted by $s' \preceq s$, if there exist indices $1 \leq i_1 < i_2 < \cdots < i_{\ell_{s'}} \leq \ell_s$ 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 $\mathcal{C} = \{c_1, \cdots, c_j\}$ 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., $\mathcal{D} = \cup_{i=1}^{j} \mathcal{D}_{c_i}$). 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:

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

sid	sequence	class
1	$\langle abadc\rangle$	c_1
2	$\langle acbe\rangle$	c_1
3	$\langle badcb\rangle$	c_2
4	$\langle eefcbc\rangle$	c_2

TABLE I. $\mathcal{D}$: A TINY LABELED SEQUENTIAL DATA SET AS AN EXAMPLE.

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:

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

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

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 [19] 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 I as an example. It contains four data sequences (i.e., $n = 4$) over the set of items $\mathcal{I} = \{a, b, c, d, e, f\}$ (i.e., $m = 6$). $\mathcal{C} = \{c_1, c_2\}$ is the set of classes, j equals to 2. The longest sequence of $\mathcal{D}$ is $s = \langle eefcbc\rangle$ (i.e., $\ell_s = \ell_{max}$), ℓ_{max} equals to 6 while k_s equals to 4. Sequence $\langle aad\rangle$ is a subsequence of $\langle abadc\rangle$. Given a sequence $s = \langle ab\rangle$, we have $f(s) = 3$, $\overline{f(s)} = 1$, $f_{c_1}(s) = 2$, $\overline{f_{c_1}(s)} = 0$, $f_{c_2}(s) = 1$ and $\overline{f_{c_2}(s)} = 1$. $\pi : \langle ab\rangle \rightarrow f_{c_1}(\langle ab\rangle) = 2,\ f_{c_2}(\langle ab\rangle) = 1$ is a sequential classification rule.

III. Bayesian framework for sequential pattern

Standard classification rule evaluation criterions 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 [16], 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:

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

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 (SCRM) $\pi : s \rightarrow f_{c_1}(s), f_{c_2}(s), \cdots, f_{c_j}(s)$ 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 SCRMs. Considering the hierarchy of parameters from the definition of SCRM, we use the following hierarchical prior distribution on SCRM 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.

Prior probability. The prior probability of a rule model π is:

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

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 $(q_1, \ldots, q_m)$ such that $\sum_{i=1}^{m} q_i = q$ is $\frac{q!}{q_1! \ldots q_m!} \prod_i (\frac{1}{m})^{q_i}$. 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:

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

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

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:

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

The complete and exact definition of the *cost* of SCRM is then:

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

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 SCRMs.

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

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

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

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

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

IV. Mining Sequential Classification Rules

Mining sequential patterns [20] is a NP-hard problem. The good pruning properties of frequency measure and condensed representations of frequent patterns [21] allows to save computational time though the problem remains hard for large-scale data sets (see [22] for the case of sequential classification rules). Our $level$ evaluation criterion does not hold as good properties as the frequency. Thus, if we look for the whole set of SCRM 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 SCRMs with positive $level$ values".

Algorithm 1: MiSeRe

input : $\mathcal{D}$, a Labeled Sequential Data Set
output: $\mathcal{R}$, a Set of SCRMs

```
1  begin
2      S ={s = ⟨s1⟩; s1 ∈ I} ;
3      R ={π : s → f_c1(s), ··· , f_cj(s); s ∈ S ∧ level(π) > 0}  ;
4      while ¬ StoppingCondition do
5          s= ChooseRandomSequence(D) ;
6          ads= ComputeNumberOfSubsequences(s) ;
7          for i = 1 to log(ads) do
8              s'= GenerateRandomSubsequence(s) ;
9              π : s' → f_c1(s'), ··· , f_cj(s') ;
10             if level(π) > 0 ∧ π ∉ R then
11                 R = R ∪ {π}
12 return R;
```

In the following, we describe our algorithm **MiSeRe** for ***Mi**ning **Se**quential Classification **Rule**s*. Firstly, we generate all SCRMs whose body is made of one single item, such rules with positive $level$ values are chosen (Lines 2-3). The stopping condition Line 4 refers to the running time that the end-user provides to the mining process. At each iteration of the main loop (Lines 4-11), a SCRM 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), we employ the efficient counting procedure presented in [23]. The inner loop (Lines 7-11) 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 $\mathcal{R}$. The main challenge in this algorithm is *"how to efficiently compute the distribution of the sequence s in each class; i.e.,* $f_{c_1}(s), \cdots f_{c_j}(s)$*"*. To achieve this task, we use a **bitset** representation and **Boolean operations** presented in [24] and 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 $O(\ell_s \times n \times log(n))$.

Classification procedure. We suggest to use the SCRMs 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 if 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 e.g., 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.

V. Experiments

In this section, we empirically evaluate our approach.The experiments are designed to discuss the following questions: **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 SCRMs mined with MiSeRe? **Q3:** 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? **Q4:** 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 briefly described in [18]. We also carried out experiments on a large marketing database from the French Telecom company Orange containing sequential information about the behavior of *76564 customers* to predict their propensity to churn. Due to page limitations, we report a detailed interactive visualization of all the results as well as the JAVA code of MiSeRe are publicly available from [18].

A. Stability of the $level$ criterion

To evaluate the stability and robustness of an interestingness measure, we perform train-test experiments. Each data set is in divide in two parts: 50% for training and 50% for testing. Then, for each mined rule, train and test values are compared. We extract frequent sequential patterns from training data set by applying cSPADE [27] 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 set as well as the values of three well-known measures: confidence, growth rate and lift. Notice that for the motivating example skater data of the introduction (figure 1), the $level$ values computed for the mined rules are perceptibly

[1] http://docs.oracle.com/javase/7/docs/api/java/util/BitSet.html

more stable than confidence and growth rate as shown in figure 2. The same observations stand for the other data sets.

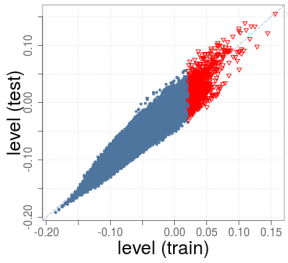


Fig. 2. Level values for mined rules in a train-test experiment for the skater data .

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 [28]. Figure 3 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.

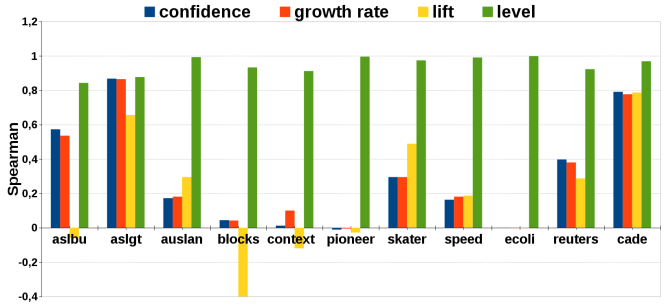


Fig. 3. Agreement between train rank and test rank of the mined rules according to measure values for benchmark data sets.

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 while respecting the original class distribution. As our method *MiSeRe* is controlled by a running time constraint, we run *MiSeRe* for 30 minutes 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**.

B. 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 SCRMs obtained with MiSeRe. We use Naïve Bayes (NB), Random Forest (RF), Decision Tree (C4.5 alias J48), Support Vector Machine (SVM), lazy classifier IBk (a k-Nearest Neighbor) available from the Weka package [29] – all with default parameter values – and the Selective Naïve Bayes[2] (SNB) [30]. The predictive performance results are all obtained with stratified 10-fold cross validation: MiSeRe operates only on the training data folds.

[2]http://www.khiops.com

Although MiSeRe is anytime, for convenience, we set a number of rules to be extracted, say 2^{10}, i.e., 1024 rules. We apply the Friedman test and a post-hoc Nemenyi test as suggested by [31] for comparisons of classifiers over multiple data sets (at significance test $\alpha = 0.05$ for both tests). The null-hypothesis is rejected, meaning the compared classifiers are not equivalent in terms of accuracy. The result of the Nemenyi test is represented by the critical difference (CD) chart shown in figure 4 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, J48, 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 another classifier are available from [18].

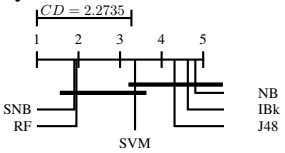


Fig. 4. Critical difference of performance between various classifiers on data using extracted SCRMs.

C. 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 5 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 reache a **plateau** with about a **few hundreds** of mined rules for most of the data sets.

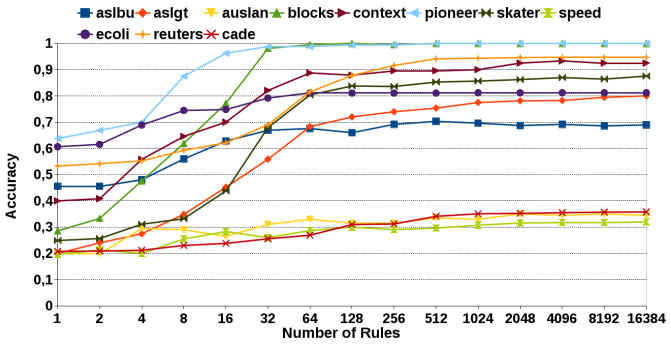


Fig. 5. Evolution of accuracy results per data set w.r.t. number of rules mined

D. SNB-MiSeRe versus state-of-the-art

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 four baseline algorithms: (1) cSPADE [27], (2) SCII [9], (3) Gokrimp [32], and (4)

DeFFeD [10]. The parameters were set for each algorithm as indicated in the original papers. Afterwards, these algorithms extract sequential rules from each training data. Then, we employ six classifiers previously mentioned on the benchmark data sets recorded using sequential classification rules obtained with *MiSeRe*, cSPADE, SCII, Gokrimp and DeFFeD. Figure 6 shows the average accuracy results per data set obtained with stratified 10-fold 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.

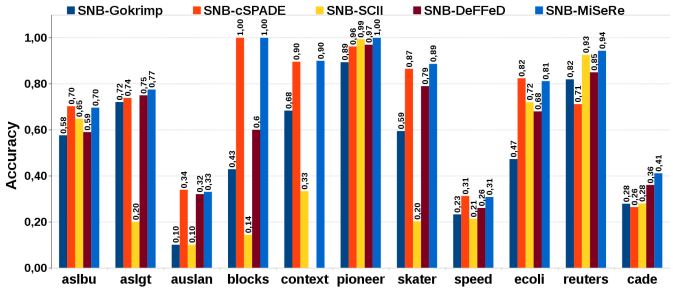


Fig. 6. Comparisons of accuracy results w.r.t. various several extraction methods.

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 [9], BayesFM [11] and CBS [8]. In this experiment we found that there is a difference of performance between *SNB-MiSeRe* and the other competitors as *SNB-MiSeRe* always scores the highest accuracy [18].

VI. Conclusion and Future Work

This paper focuses on the important problem of mining sequential rule patterns for classification purpose. 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.

References

[1] Z. Xing, J. Pei, and E. J. Keogh, "A brief survey on sequence classification," *SIGKDD Explorations*, vol. 12, no. 1, pp. 40–48, 2010.

[2] M. Deshpande and G. Karypis, "Evaluation of techniques for classifying biological sequences," in *PAKDD'02*, 2002, pp. 417–431.

[3] R. She, F. Chen, K. Wang, M. Ester, J. L. Gardy, and F. S. L. Brinkman, "Frequent-subsequence-based prediction of outer membrane proteins," in *ACM SIGKDD'03*, 2003, pp. 436–445.

[4] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[5] P. Tan and V. Kumar, "Discovery of web robot sessions based on their navigational patterns," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 9–35, 2002.

[6] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *ACM SIGKDD'98*, 1998, pp. 80–86.

[7] A. Zimmermann and S. Nijssen, "Supervised pattern mining and applications to classification," in *Frequent Pattern Mining*, 2014, pp. 425–442.

[8] V. S. Tseng and C. Lee, "CBS: A new classification method by using sequential patterns," in *SDM'05*, 2005, pp. 596–600.

[9] C. Zhou, B. Cule, and B. Goethals, "Itemset based sequence classification," in *ECML/PKDD'13*, 2013, pp. 353–368.

[10] P. Holat, M. Plantevit, C. Raïssi, N. Tomeh, T. Charnois, and B. Crémilleux, "Sequence classification based on delta-free sequential patterns," in *ICDM'14*, 2014, pp. 170–179.

[11] N. Lesh, M. J. Zaki, and M. Ogihara, "Mining features for sequence classification," in *ACM SIGKDD'99*, 1999, pp. 342–346.

[12] K. Deng and O. R. Zaïane, "An occurrence based approach to mine emerging sequences," in *DaWaK'10*, 2010, pp. 275–284.

[13] H. T. Lam, F. Moerchen, D. Fradkin, and T. Calders, "Mining compressing sequential patterns," in *SDM'12*, 2012, pp. 319–330.

[14] F. Coenen and P. H. Leng, "The effect of threshold values on association rule based classification accuracy," *Data & Knowledge Engineering*, vol. 60, no. 2, pp. 345–360, 2007.

[15] F. Mörchen and A. Ultsch, "Efficient mining of understandable patterns from multivariate interval time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 181–215, 2007.

[16] M. Boullé, "MODL: A bayes optimal discretization method for continuous attributes," *Machine Learning*, vol. 65, no. 1, pp. 131–165, 2006.

[17] D. Gay and M. Boullé, "A bayesian approach for classification rule mining in quantitative databases," in *ECML/PKDD'12*, 2012, pp. 243–259.

[18] "Companion website. MiSeRe: Mining sequential classification rules," 2015. [Online]. Available: http://misere.co.nf

[19] A. M. Jorge, P. J. Azevedo, and F. Pereira, "Distribution rules with numeric attributes of interest," in *PKDD'06*, 2006, pp. 247–258.

[20] R. Agrawal and R. Srikant, "Mining sequential patterns," in *ICDE'95*, 1995, pp. 3–14.

[21] H. Mannila and H. Toivonen, "Multiple uses of frequent sets and condensed representations (extended abstract)," in *KDD'96*, 1996, pp. 189–194.

[22] E. Baralis, S. Chiusano, R. Dutto, and L. Mantellini, "Compact representations of sequential classification rules," in *Data Mining: Foundations and Practice*, 2008, pp. 1–30.

[23] C. Elzinga, S. Rahmann, and H. Wang, "Algorithms for subsequence combinatorics," *Theoretical Computer Science*, vol. 409, no. 3, pp. 394 – 404, 2008.

[24] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential pattern mining using a bitmap representation," in *KDD'02*. ACM, 2002, pp. 429–435.

[25] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[26] A. Cardoso-Cachopo, "Improving Methods for Single-label Text Categorization," PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

[27] M. J. Zaki, "Sequence mining in categorical domains: Incorporating constraints," in *CIKM'00*, 2000, pp. 422–429.

[28] J. L. Myers and A. D. Well, *Research Design and Statistical Analysis*. New Jersey: Lawrence Erlbaum Associates, 2003.

[29] M. A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[30] M. Boullé, "Compression-based averaging of selective naive bayes classifiers," *Journal of Machine Learning Research*, vol. 8, pp. 1659–1685, Dec. 2007.

[31] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, Dec. 2006.

[32] H. T. Lam, F. Mörchen, D. Fradkin, and T. Calders, "Mining compressing sequential patterns," *Statistical Analysis and Data Mining*, vol. 7, no. 1, pp. 34–52, 2014.