A Relevance Criterion for Sequential Patterns

Henrik Grosskreutz, Bastian Lang and Daniel Trabold

Fraunhofer IAIS, Schloss Birlinghoven, 53754 St. Augustin, Germany
{henrik.grosskreutz,bastian.lang,daniel.trabold}@iais.fraunhofer.de

Abstract. The theory of relevance is an approach for redundancy avoidance in labeled itemset mining. In this paper, we adapt this theory to the setting of sequential patterns. While in the itemset setting it is suggestive to use the closed patterns as representatives for the relevant patterns, we argue that due to different properties of the space of sequential patterns, it is preferable to use the minimal generator sequences as representatives, instead of the closed sequences. Thereafter, we show that we can efficiently compute the relevant sequences via the minimal generators in the negatives. Unlike existing iterative or post-processing approaches for pattern subset selection, our approach thus results both in a reduction of the set of patterns and in a reduction of the search space – and hence in lower computational costs.

1 Introduction

Sequential pattern mining is concerned with finding frequent subsequences in sequence databases [1]. These subsequences, or sequential patterns, have many real-world applications. For example, they can be used to characterize sequences of credit card transactions having high fraud probability, or DNA subsequences having particular properties.

Like most pattern mining tasks, sequential pattern mining suffers from the problem that it mostly comes up with huge amounts of patterns. This problem is well-know in the pattern mining community, and various approaches have been proposed to cope with this issue (e.g. [2–5]).

In this paper, we take one particular approach, namely the *theory of relevance* [6, 7], and adapt it to the case of sequential data. Originally, the theory of relevance was developed in the setting of *labeled itemset data*, and assumes that one is interested in characterizing a particular *target class*. The basic idea is related to the concept of Pareto domination: remove all itemsets which are *dominated* by some other itemset, meaning that the dominated itemset is strictly inferior in characterizing the target class. More precisely, an itemset is considered as dominated if there is another, dominating itemset which supports at least all *positives* (i.e. target-class sequences) supported by the dominated itemset, but no additional *negative* (i.e. non target-class sequences).

The theory of relevance not only reduces the size of the resulting set of itemsets, but also allows for efficient algorithms. Unlike iterative or post-processing approaches, the relevant itemsets can be collected by traversing, once only, a small subset of all itemsets. The foundation for these algorithms is a set of properties that relate the relevant itemsets to the itemsets that are closed in a particular subset of the data, namely the positives.

The adaptation of the theory of relevance to sequential data raises interesting challenges. These are due to the different characteristics of the space of sequential patterns, compared with the space of itemsets. One well-known difference, illustrated in Figure 1, is that the closed sequences are no *unique* representatives of their equivalence class. This is unlike in the itemset setting, where the closed itemsets are used as unique representatives for the relevant itemsets [8]. The different characteristics of sequential patterns makes the use of closed (sequential) patterns much less suggestive for this new setting. As we will show, there are other important differences (for example, the relevance of a sequence cannot be checked by considering its generalizations, unlike in the setting of itemsets). Altogether, we make the following contributions:

label	*sequence*
+	abc
+	acb
-	b
-	c

Fig. 1. In this sequence database, the sequences *a*, *ab* and *ac* have the same support set: while the first sequence is a (minimal) generator, the other two sequences are closed. Applying the idea of domination to this example, we see that all sequences but the three abovementioned are dominated: for example, the sequence *b* is dominated by *a*, because *b* supports a superset of the negatives supported by *a* but the same set of positives. In our approach, only the minimal generator *a* will be kept.

– We show that if the concept of domination is transfered from itemsets to sequences, several important properties no longer hold. As a consequence, the standard algorithmic approach cannot be applied to find the relevant sequences (Section 4.2);
– We propose to use generators as representatives for the relevant sequences, instead of closed sequences. Besides the obvious advantage of shorter descriptions, this allows dealing efficiently with maximum pattern length constraints (Section 4.3);
– We show that our new definition of relevance has the consequence that the relevant sequences are a subset of the minimal generator sequences in the negatives (Section 5.1);
– Subsequently, we describe how this connection can be turned into an efficient algorithm (Section 5.2);
– Finally, we experimentally investigate the impact of our new relevance criterion on the number of patterns and the computational costs (Section 6).

The rest of this paper is structured as follows: After discussing related work in Section 2 and introducing our notation in Section 3, we present the contributions listed above, before we conclude in Section 7.

2 Related Work

Sequential pattern mining was first considered by Agrawal and Srikant [1]. The notions closed pattern, minimal generator pattern etc. have been transfered from itemset data to the setting of sequences, and different algorithms have been proposed to find the closed sequences [9, 10], respectively the minimal generator sequences [11, 12].

While the use of closed patterns resp. minimal generators reduces the number of patterns, the outcome can still be huge. A variety of pattern selection approaches have been proposed to cope with this issue. Most of these approaches are either post-processing or iterative solutions. The post-processing approaches expect, as input, a set of patterns, from which they choose a subset, typically in a greedy fashion [2, 4]. The iterative approaches, on the other hand, run a new search in every iteration; The different runs assess the pattern quality differently, taking into account the set of patterns already collected [13–15, 5]. As all these approaches rely, somehow, on an underlying mining algorithm, they are no alternative to our approach but could, instead, be combined with our approach: that is, for labeled sequential data they could rely on our algorithm to enumerate the candidate patterns.

Other approaches exist that reduce the set of patters by relying strongly on the properties and operations that can be performed on itemset data [3, 16]. As these operations are not directly applicable to sequences, there is no easy way to transfer these approaches to the setting of sequential patterns (Note that the theory of relevance considered here falls into this category of approaches).

Different approaches have been proposed to define a closure operator in a sequential data setting. However, none of these approaches is directly applicable to our setting, as they all consider different patterns families, which are only connected via some post-processing to the (classical) sequential patterns we consider. In particular, Garriga has proposed a new closure operator, which however is not defined on individual sequential patterns, but on *sets* of sequential patterns. While this approach allows for advanced summarization [17], it relies on a classical closed sequential pattern miner to produce the patterns to be post-processed respectively summarized. Raïssi et al. [18] presents a similar approach, which also considers sets of sequences instead of individual sequences. Finally, Tatti et al. [19] proposed a new notion of closedness, called *i-closed*. Unlike us, they don't consider sequence databases but consider the episode mining setting (where frequency is defined in terms of sliding windows over a single sequence) and consider patterns taking the form of directed acyclic graphs. Above all, the computation of i-closed episodes is only the first step: the i-closed episodes are a *superset* of the classical closed episodes, from which the closed episodes must then be computed in a second step.

3 Preliminaries

In this section, we review the standard notions from itemset and sequential pattern mining, which will then be used in the remainder of this paper.

3.1 Itemsets, Closed Patterns and Minimal Generators

As the theory of relevance has been defined in the scope of itemsets, we will first review the notions from itemset and closed pattern mining [20].

Itemsets and Itemset Databases An *itemset* over an alphabet Σ is a subset of Σ. A labeled dataset $\mathcal{DB}$ over an alphabet Σ is a collection of records (l, I), where l is a label and I is an itemset. Given a database $\mathcal{DB}$ and an itemset P, the *support set* of P in a dataset $\mathcal{DB}$, denoted by $\mathcal{DB}[P]$, is defined as the set of records $r = (l, I) \in \mathcal{DB}$ such that P is a subset of I. The *support* of an itemset is the size of its support set.

Positives, Negatives, True Positives etc. In the rest of this paper we assume a binary setting where the set of classes consists of "+" and "-". We call the subset of "+"-labeled records the *positives.* Similarly, we call the "-"-labeled records the *negatives.* The term *true positives*, denoted by $TP(\mathcal{DB}, P)$, refers to the support set of P in the positives. The *false positives*, $FP(\mathcal{DB}, P)$, are defined analogously on the negatives.

Equivalence Classes, Closed Itemsets and Generators If two itemsets have the same support set, then the two are said to be *equivalent.* The space of itemsets can thus be partitioned into *equivalence classes*: all itemsets with the same support set belong to the same equivalence class. Within an equivalence class, there are two interesting subsets of itemsets: the *minimal generators* and the *closed* itemsets. The minimal generators are the minimal members of an equivalence class, meaning that any true generalization (i.e. sub-itemset) has a strictly higher support in the dataset. The closed itemsets are their counterpart: they are maximal members of the equivalence class, meaning that any true specialization (i.e. super-itemset) has strictly lower support.

3.2 The Theory of Relevance

We will now turn to the theory of relevance [6–8].

Domination and Relevance The basic idea of the theory of relevance is to reduce the number of itemsets by removing itemsets that are irrelevant for the purpose of characterizing the *target class*, which by convention is the "+" class. An itemset is considered to be irrelevant if there is another itemset, called the *dominating* itemset, which allows characterizing the target class at least as good as the former (dominated) itemset. Formally:

Definition 1. *The itemset P* dominates *the itemset P_{irr} in dataset $\mathcal{DB}$ iff (i) $TP(\mathcal{DB}, P) \supseteq TP(\mathcal{DB}, P_{irr})$ and (ii) $FP(\mathcal{DB}, P) \subseteq FP(\mathcal{DB}, P_{irr})$.*

Note that it is possible that two itemsets dominate each other, however only in the case that they belong to the same equivalence class. Given that for itemsets there is exactly one closed itemset in every equivalence class, the closed itemsets can be used as unique representatives. Garriga et al. [8] thus define the relevant itemsets as follows:

Definition 2. *Itemset P is* relevant *in database $\mathcal{DB}$ iff (i) P is closed and (ii) there is no itemset having a different support set that dominates P (in $\mathcal{DB}$).*

The Connection to Closed-on-the-Positives Garriga et al. [8], have shown that when searching for relevant itemsets, it is sufficient to consider only itemsets that are *closed on the positives*, that is, itemsets that are closed in the subset of the positively labeled records:

Proposition 1 ([8]) *The space of relevant itemsets consists of all itemsets P_{rel} satisfying the following:*

- *P_{rel} is closed on the positives, and*
- *there is no generalization $P \subsetneq P_{rel}$ closed on the positives such that $|FP(\mathcal{DB}, P)| = |FP(\mathcal{DB}, P_{rel})|$.*

The above proposition provides an elegant way to compute the relevant itemsets, sketched in Algorithm 1.

Algorithm 1 CPOS Relevant Itemset Miner

```
Input  : an itemset database $\mathcal{DB}$
Output : the relevant itemsets in $\mathcal{DB}$

1: collect all closed-on-the-positive itemsets (using some closed itemset mining algo-
   rithm, e.g. [21]).
2: remove all itemsets having a (closed-on-the-positives) generalization with the same
   negative support.
```

3.3 Sequences and Sequence Databases

We will now review the most important notions from sequence mining [1].

Sequences and Sequence Databases A *sequence* over a set of items Σ is a sequence of items $i_1, \ldots, i_l$, $i_i \in \Sigma$. The *length* of the sequence is the number of items in the sequence. A sequence $S_a = a_1, \ldots, a_n$ is said to be *contained* in another $S_b = b_1, \ldots b_m$, denoted by $S_a \sqsubseteq S_b$, if $\exists i_1, \ldots i_n$ such that $1 \leq i_1 < \cdots < i_n \leq m$ and $a_1 = b_{i_1}, \ldots, a_n = b_{i_n}$. We also call S_a a *generalization* of S_b.

A *sequence database* $\mathcal{SDB}$ is a collection of labeled sequences. A *labeled sequence* is a tuple (l, S), where S is a sequence and l a label – i.e, "+" or "-". Again, we call the subset of "+"-labeled sequences the *positives*. Similarly we call the "-"-labeled sequences the *negatives*.

The *support set* of a sequence S in a sequence database $\mathcal{SDB}$, denoted by $\mathcal{SDB}[S]$, consists of all labeled sequences in $\mathcal{SDB}$ that contain S. Here, a labeled sequence (l, S) contains a sequence S_a iff $S_a \sqsubseteq S$. Again, the term *true positives*, denoted by $TP(\mathcal{SDB}, S)$, refers to the support set of S in $\mathcal{SDB}$'s positives. $FP(\mathcal{SDB}, S)$ is defined analogously on the negatives. Finally, the *support* denotes the size of the support set.

Patterns, Closed Patterns and Minimal Generators In the rest of this paper, we will use the general term *pattern* to refer to either an itemset or a sequence. In general, patterns have a support set (wrt. a given database), and moreover there is a partial generalization order between patterns (defined via the subset relation for itemsets, resp. the *contained* relation for sequences).

Based upon these generalized definitions of support set and generalization, the terms *closed* and *minimal generator* from Section 3.1 can be carried over to sequences, and can hence be applied to both types of patterns.

4 Relevant Sequences

We will now adapt the definition of relevance to sequential data. While is is straightforward to transfer the concept of domination to sequential patterns, defining relevance will raise subtle issues.

4.1 Domination between Sequences

Unlike the original definition (Definition 1), our definition of domination between sequential patterns explicitly distinguishes between *weak* and *strong* domination. This will be useful in situations where two different patterns dominate each other circularly (in the original definition).

Definition 3. *The sequence* S_d weakly dominates *the sequence* S *iff*

– $TP(\mathcal{SDB}, S_d) \supseteq TP(\mathcal{SDB}, S)$, *and*
– $FP(\mathcal{SDB}, S_d) \subseteq FP(\mathcal{SDB}, S)$.

Moreover, S_d strongly dominates S *iff* S_d *weakly dominates* S *and* $\mathcal{SDB}[S_d] \neq \mathcal{SDB}[S]$.

4.2 Relevant Sequences: Problems and Differences to the Itemset Setting

While we directly carried over the definition of *domination* to sequential patterns, our proposed definition of *relevant sequences* will differ from the definition used in the setting of itemsets. This is due do the fact that several properties that hold in the space of itemsets do not transfer to the space of sequences.

One main issue is the choice of representatives for the patterns that are not strongly dominated. In the itemset setting, Garriga et al. chose to use the closed itemsets as representatives. In the itemset setting, this is very suggestive: it provides unique representatives and allows for efficient computation. In this section, we will argue that in the sequential setting, the use of closed patterns as representatives is much less appealing. Beside the issue that there can be several equivalent closed sequences (as illustrated in the introduction), the use of closed patterns as representatives results in the following issues:

1. The computational approach proposed by Garriga et al. is not applicable, because Proposition 1 does not carry over to sequential patterns;
2. The use of a length limit is problematic, resulting in counter-intuitive results and/or excessive computational costs. In practice, however, specifying a limit for the length of the patterns to be considered is very useful: it allows reducing the computational costs to a reasonable amount of time, and is often more suitable than using a minimum support threshold.

We will now discuss these issues in detail.

Garriga's Computational Approach is not applicable to Sequences Proposition 1 is the foundation for many fast relevant itemset mining algorithms [8, 22]. We will now show that it does not carry over to sequential patterns:

Proposition 2 *There is a sequence database such that a closed-on-the-positives sequence pattern S exists which is strongly dominated, yet not dominated by any of its generalizations.*

The correctness of the above proposition is shown by the example in Table 1. Here, the sequence c is closed on the positives. It is, however, dominated, namely by ab. Yet, c is not dominated by any generalization of itself.

label	*sequence*
+	cab
+	abc
-	c
-	d

Table 1. Example: the closed sequence c is strongly dominated (e.g. by a), yet it is not dominated by any generalization.

The above proposition shows that it is not sufficient to consider *generalizations* to verify the relevance of a sequence. While the above example alone shows that Proposition 1 does not hold, we could still hope that testing for relevance is possible by comparing only other patterns with same negative support (as in Proposition 1). However, this also does not carry over:

Proposition 3 *There is a sequence database such that a closed-on-the-positives sequence S exists which is strongly dominated, yet it is not strongly dominated by any sequence having the same negative support.*

Again, this is illustrated by Example 1. c is closed on the positives and is strongly dominated. However, all strongly dominating sequences (a, b, and ab) have a different negative support.

The above two propositions show that the second step of Algorithm 1 cannot be adapted to the sequential pattern setting: neither can relevance be tested by considering only generalizations; nor is it possible to consider only patterns with same negative support.

Problems with Length Limits We will now turn to the issues that arise if a length limit is introduced and closed patterns are used as representatives for the relevant sequences (Issue 2). Here, instead of considering the space of all sequences, we are only concerned with the space of sequences satisfying the length limit. We wish to remove all sequences that are strongly dominated, and to keep only a set of representatives for the remaining sequences.

Again, the example from Table 1 illustrates the problems that arise if closed patterns are used as representatives: assume that we are searching for relevant sequences with a maximum length limit of 1. Then:

– c is dominated, namely by the patterns a, b and ab. It should thus not be in the result set, because it is dominated by some pattern satisfying the length limit.
– c is, however, not dominated by any closed pattern satisfying the length limit (a and b are not closed). Checking domination would hence require a computationally much more expensive approach, for example considering *all* sequences up to the length limit, not only closed sequences.
– a should not be in the result because it is not closed; same for b. However, ab, which is closed and lies in the same equivalence class as the earlier two sequences, has a too long description. The result is that there is no representative in the result set for this equivalence class. This is somewhat counter-intuitive.

While it might be possible to ensure efficient computation by using a different, computationally-motivated definition of relevance wrt. a length limit, this is likely to result in awkward and unintuitive results.

4.3 The Relevant Sequences

As we have seen in the previous section, the closed sequences are not a particularly suggestive set of representatives for the relevant sequences. Therefore, we propose to use a different set of patterns as representatives: the *minimal generator sequences*:

Definition 4. *Given a sequence database $\mathcal{SDB}$ and a length limit L, the set of* relevant minimal generator sequences *(wrt. $\mathcal{SDB}$ and L) consists of all sequences S that satisfy the following:*

1. *S is a minimal generator in $\mathcal{SDB}$ of length $\leq L$,*
2. *S is not strongly dominated (in $\mathcal{SDB}$) by any other sequence of length $\leq L$.*

In the following, the database and length limit will be clear from the context, so they will not be explicitly listed. Moreover, if no length limit is given, this is handled as if $L = \infty$. Finally, we will use the expression *relevant sequence* to refer to an element of the set of relevant minimal generator sequences.

Using minimal generators as representatives has several advantages. First, it produces shorter descriptions, which can be an important advantage if the patterns are to be read and interpreted by human experts; second, it allows for efficient computation via the minimal-generators-in-the-negatives, as we will describe in Section 5; and finally it allows for maximum length constraints with clear and simple semantics:

Proposition 4 *Let $\mathcal{SDB}$ be a sequence database, L a positive integer and S some sequence of length $\leq L$. Then, there is a relevant sequence S^* in $\mathcal{SDB}$ such that S^* is of length $\leq L$ and S^* weakly dominates S.*

Hence, for every sequence S satisfying the length limit there is a relevant sequence as good as S in characterizing the target class.

Proof. Let S_G be (one of the) minimal generator of S. Obviously, S_g weakly dominates S and satisfies the length limit. If S is a relevant sequence, then $S^* = S_g$ and we're finished. Else, S_g must be strongly dominated. As strong domination is transitive, non-reflexive and the set of minimal generators satisfying the length constraint is finite, there must be (at least one) minimal generator S^* that (i) satisfies the length constraint, (ii) strongly dominates S_g and (iii) is not dominated by any other minimal generator of length $\leq L$.

It remains to show that this pattern S^* dominates S_g and that it is a relevant sequence. The first fact follows by transitivity of weak domination. Concerning the second fact, S^* is a minimal generator and satisfies the length constraint by construction. It remains to show that it is not strongly dominated, which we show by contradiction. Assume it is dominated by a sequence of length $\leq L$, then it would also be dominated by the minimal generators of that pattern. Contradiction with (iii) above. □

5 Computing the Relevant Sequences

We will now present a new approach that allows computing the relevant sequences much more efficiently than by simply checking, for every pair of patterns, the dominance criterion from Definition 3.

5.1 Relevant Sequences and Minimal Generators in the Negatives

This approach is based on the observation that the set of patterns not-strongly-dominated is not only related to the closed-on-the-positives (as investigated by Garriga et al.), but also to their counterpart: namely to the *minimal generators* in the *negatives.*

Proposition 5 *Let $\mathcal{SDB}$ be a sequence database and S some relevant sequence in $\mathcal{SDB}$. Then S is a minimal generator in $\mathcal{SDB}$'s* negatives.

Proof. By contradiction. Assume that S is a relevant sequence but is no minimal generator in the negatives. The latter implies that there is a generalization S' of S with same support in the negatives. Thus, we have that $FP(\mathcal{SDB}, S) \supseteq$

$FP(\mathcal{SDB}, S')$. Just as in the case of classical itemsets, for sequential patterns we have the property that the support is anti-monotonic. That is, the support set of S' in the positives is a superset of the support set of S in the positives. Thus, we also have $TP(\mathcal{SDB}, S) \subseteq TP(\mathcal{SDB}, S')$. The above implies that S' weakly dominates S. Moreover, by the assumption that S is a relevant sequence together with Definition 4, we have that S is a minimal generator, hence S and S' have different support sets. Hence, S' strongly dominates S – which contradicts the assumption that S is relevant. □

Please note that in the above proposition, unlike in the work of Garriga et al. we consider a different pattern type (minimal generators instead of closed patterns) but also a different subset of the data (negatives instead of positives).

label	*sequence*
+	ab
+	abc
-	abc
-	ac
-	c

(a) dataset

seq.	*dominated*	*closed*	*gen*	*g-neg*
a	yes, by b	y	y	y
b	no	-	y	y
c	yes, by b	y	y	-
ab	no	y	-	-
ac	yes, by b	y	y	-
bc	yes, by b	-	y	-
abc	yes, by b	y	-	-

(b) sequential patterns

Fig. 2. Subfigure 2(b) considers all sequential patterns occurring in the dataset in Subfigure 2(a). As the 2nd column shows, all sequences but *b* and *ab* are strongly dominated. These two patterns belong to the same equivalence class, with *b* being a minimal generator (column "gen") and *ab* a closed sequence (column "closed"). As we opted for the minimal generators as representatives, we want to come up with *b*. While this result can be computed using the minimal generators as candidates (column "gen"), using the minimal generators in the negatives (column "g-neg") is more efficient as this yields a smaller candidate set.

We will now illustrate the above proposition and its implications using the example in Figure 2. In this database, only the sequences *b* and *ab* are potentially useful in characterizing the target class. All other sequential patterns are strongly dominated, and should thus be removed as irrelevant. The two un-dominated patterns *b* and *ab* are equivalent, hence we would be happy with just one of these two as representative. More precisely, according to our new approach, we would select *b* as representative, which is the (only) minimal generator. Now Proposition 5 shows that to compute this result, it is sufficient to consider the set of minimal generators in the negatives (i.e. *a* and *b*) as candidates, instead of considering the whole set of minimal generators (which comprises 5 sequences).

5.2 Our Algorithm

The new relation stated in Proposition 5 suggests the following approach: first compute the minimal generators in the negatives (using some standard minimal generator sequence miner, e.g. [11, 12]) and then remove the dominated generators. So far, we have not considered the second step – the removal of the dominated generators. The following proposition shows that it is possible to decide whether a pattern is strongly dominated solely by considering the minimal-generators-in-the-negatives:

Proposition 6 *Let $\mathcal{SDB}$ be a sequence database, L a positive integer, and S_{irr} some minimal generator of length $\leq L$ that is not relevant (wrt. $\mathcal{SDB}$ and L). Then, there is a minimal-generator-in-negatives S_g of length $\leq L$ strongly dominating S_{irr}.*

Proof. Let S_d be one of the sequences strongly dominating S_{irr} and satisfying the length limit. By Proposition 4, there is a relevant sequence S_g weakly dominating S_d and satisfying the length limit. By Proposition 5, S_g is a minimal generator in the negatives. Moreover, by transitivity S_g strongly dominates S_{irr}, which completes the proof. □

Algorithm 2 Relevant Sequence Miner

```
Input  : a sequence database $\mathcal{SDB}$ and optionally a length limit $L$
Output : the relevant sequences

 1: Calculate the set $\mathcal{G}_\mathcal{N}$ of sequences that are minimal generator in the negatives and
    have length $\leq L$;
 2: Group the candidate patterns $\mathcal{G}_\mathcal{N}$ into sets having same extension in $\mathcal{SDB}$. Let $\mathcal{G}_\mathcal{N}^{\equiv}$
    denote the resulting set of equivalence classes
 3: sort the set $\mathcal{G}_\mathcal{N}^{\equiv}$ by (i) descending positive support and (ii) in case of ties ascending
    negative support
 4: let $\mathcal{R}$ be an empty set of equivalence classes
 5: for every class $e$ in $\mathcal{G}_\mathcal{N}^{\equiv}$ do
 6:   if $e$ is not dominated by any class in $\mathcal{R}$ then
 7:     add $e$ to $\mathcal{R}$
 8:   end if
 9: end for
10: return The set of minimal generators in $\mathcal{R}$
```

The above Proposition, together with Proposition 5, is the foundation for our algorithmic approach, sketched in Algorithm 2. In Line 1, the algorithm can make use of any minimal generator sequence miner, e.g. [11, 12] to compute the minimal-generators-in-the-negatives. The rest of the pseudo-code takes care of filtering strongly dominated sequences from this candidate set. Instead of the naive approach – comparing every pair of candidates, which would result in a quadratic number of comparisons – we use a slightly more efficient solution. First, we group the candidate patterns (i.e. the minimal-generators-in-the-negatives) into equivalence classes (Line 2). The reason is that the definition of domination immediately carries over from patterns to equivalence classes and it is thus sufficient to consider those instead of the individual patterns. The second improvement is that we sort the candidates, resp. the equivalence classes, by descending positive support and then, in case of ties, by ascending negative support. This step, done in Line 3, ensures that a pattern can only be dominated by a predecessor in the sorted list. As a consequence, during the following iteration over the candidates one only has to compare a candidate with the predecessors in the sorted list *which have been verified to be relevant.* Hence, the number of comparisons per candidate is limited by the number of relevant patterns.

5.3 Analysis of the new Algorithm

The correctness of our algorithm follows directly from Propositions 5 and 6. We will now turn to its complexity: Let n denote the number of items, m the number of sequences in the database, L the length limit, l the maximum length of the sequences, $|\mathcal{G}_{\mathcal{N}}|$ the number of minimal generators in the negatives, $|\mathcal{G}_{\mathcal{N}}^{\equiv}|$ the equivalence classes including a minimal generator in the negatives, and $|\mathcal{R}^{\equiv}|$ the number of relevant equivalence classes. Then

1. The runtime of the first step in Algorithm 2 – computing the minimal generators – is $O(n^L \cdot m \cdot l)$.
2. The grouping of the candidate sequences can be done using a hash function mapping the support set to an integer. The runtime is then $O(|\mathcal{G}_{\mathcal{N}}| \cdot m \cdot l)$.
3. The runtime for sorting is $O(|\mathcal{G}_{\mathcal{N}}^{\equiv}| \log(|\mathcal{G}_{\mathcal{N}}^{\equiv}|))$.
4. The loop is executed $|\mathcal{G}_{\mathcal{N}}^{\equiv}|$ times. The condition in the if requires to check each equivalence class against at most $|\mathcal{R}^{\equiv}|$ relevant patterns. Every comparison can be done in $O(m)$, assuming that hash-sets are used to check for inclusion of a record. The total runtime is thus $O(|\mathcal{G}_{\mathcal{N}}^{\equiv}| \cdot |\mathcal{R}^{\equiv}| \cdot m)$.

Overall, the runtime for the computation of Algorithm 2 is hence

$$O(n^L \cdot m \cdot l + |\mathcal{G}_{\mathcal{N}}| \cdot m \cdot l + |\mathcal{G}_{\mathcal{N}}^{\equiv}| \log(|\mathcal{G}_{\mathcal{N}}^{\equiv}|) + |\mathcal{G}_{\mathcal{N}}^{\equiv}| \cdot |\mathcal{R}^{\equiv}| \cdot m).$$

As the number of minimal generators is typically much smaller than n^L, the overall runtime is typically dominated by the first summand – that is, the runtime is dominated by the first step which computes the minimal generators in the negatives. This will be confirmed by experiments presented in Section 6.

6 Experimental Evaluation

In this section, we experimentally evaluate the impact of our approach. As several investigations have demonstrated that removing dominated patterns is beneficial for classification purposes [6, 8, 23], we only investigate the effect on the size of the result pattern set and on the computational costs.

6.1 Implementation and Setup

Our approach requires, as a building block, a minimal generator sequence miner. To this end, we have used a (slightly modified) reimplementation of FEAT [11], a state-of-the-art generator sequence mining algorithm. In particular, our implementation allows for *maximum length constraints*. This can easily be realized by stopping the recursive traversal of the candidate space if a pattern violates the length constraint.

We used five sequence datasets in our evaluation. The datasets 'hill-valley', 'libras', 'person-activity', and 'promoter' are publicly available datasets from the UCI repository [24]. The last dataset, 'wlan', is from an ongoing project and cannot be made publicly available. Table 3 shows all datasets together with their most important statistics.

dataset	# seq.	# pos.	# items	max. length
hill-valley	606	301	5	100
libras	360	192	979	89
person	273	198	116	8610
promoter	106	53	4	57
wlan	206	166	15	2920

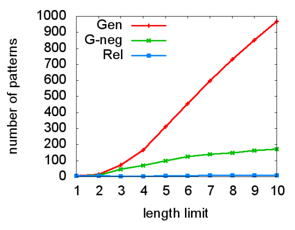
Fig. 3. Datasets

6.2 Results

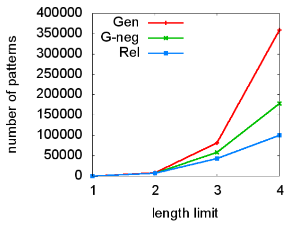
We will now show how the concept of relevance affects the number of patterns obtained. Figures 4(a) to 4(e) show, for different datasets and length limits, the number of minimal generators ("Gen"), the number of minimal generators in the negatives ("G-neg") and of relevant sequences ("Rel"). In the experiments, we also used a minimum support of 10%.

Reduction of the Pattern Set The figures show, first, that the concept of relevance dramatically reduces the number of patterns. At higher length limits, the reduction from all generator sequences ("Gen") to the relevant sequences ("Rel") amounts to several orders of magnitude. The results are similar if the size of the outcome is controlled using a support threshold instead of a length limit. We show a corresponding plot for the 'hill-valley' dataset in Figure 4(f), where we additionally used a length limit of 10 (the result for other datasets are similar and omitted for space reasons). This demonstrates the main benefit of our relevance criterion for sequential patterns: it tremendously reduces the number of sequential patterns.

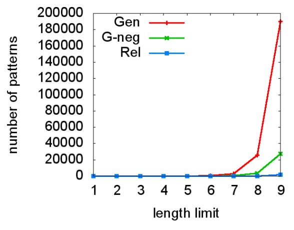
Computational Speedup The second observation is that the computation via the minimal-generators-in-the-negatives reduces the computational costs. Again, this can be seen in Figure 4, which shows the reduction from generators ("Gen") to generators-in-the-negatives ("G-neg"). These numbers are less implementation-dependent than the runtime, and hence a more convenient assessment of the



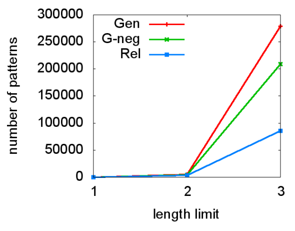
(a) 'hill-valley'



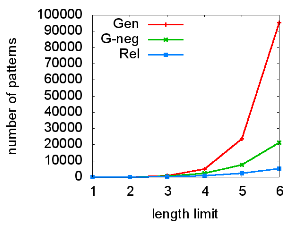
(b) 'libras'



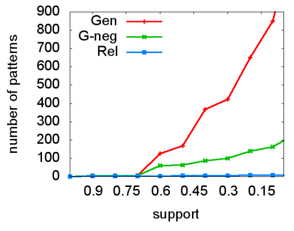
(c) 'promoter'



(d) 'person activity'



(e) 'wlan'

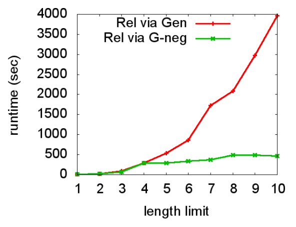


(f) 'hill-valley (by support)'

Fig. 4. The relevance constraint tremendously reduces the number of patterns. Moreover, the generators-in-the-negatives approach significantly reduces the candidate set.

computational costs (for comparison, we also show the runtimes on a Core 2 Duo E8400 for the hill-valley dataset in Figure 5(a)). The experiments show that while the reduction varies between the datasets, it can amount to an order of magnitude.

We also compared the runtimes for the first and second step of our algorithm, namely computing the generators-in-the-negatives and removing the dominated candidates. The result is shown in Figure 5(b). It shows that for all datasets, the computational costs are dominated by the candidate mining step. While the table shows the values for a maximum length of 2 and a support threshold of 30%, the results are similar for other settings.



(a) runtime using the G-neg instead of all generators

dataset	share of 2nd step
hillValley	4.3 %
libras	1.6 %
personActivity	8.6 %
promoter	11.1 %
WLAN	0.3 %

(b) Share of the computational time spent in the filtering step

Fig. 5. Runtime figures showing (a) that the generators-in-the-negatives approach reduces the computation time, and (b) that the overall costs are dominated by the candidate mining step.

7 Conclusions

In this paper, we have adapted the idea of relevance [6] to sequential data. We have shown that several important properties do not carry over from itemset to sequence data. This makes the use of the closed-on-the-positives as representatives less appealing, which motivated our proposal to use, instead, the minimal generator sequences as representatives. Besides coming up with shorter descriptions, this has the important advantage that it allows for a meaningful maximum pattern length constraint, which can be very useful in practical applications.

Subsequently, we have presented a computational approach for mining the relevant sequences. Our approach is based on the relation between relevant sequences and *minimal generators* in the *negatives*. This relation is kind of the counterpart to the relation between relevant itemsets and *closed itemsets* in the *positives*, discovered by Garriga et al. [8].

In the experimental section, we have shown that the concept of relevance results in a tremendous reduction of the number of patterns, and that the generators-in-the-negatives approach reduces the computational costs. Our approach thus improves upon the use of all sequence generators in a similar way as Garriga's approach exceeds over the use of all closed itemsets. For sequence data, computing the minimal generators is, in general, not more demanding than computing closed patterns. Thus, our algorithm would also be a good choice as underlying miner in post-processing [2, 4] or iterative approaches [13–15, 5].

There are several lines in which our research can be extended. For one, it could be adapted to sequences of itemsets, as opposed to the sequences of items considered here. For another, the relation between minimal generators in the negatives and relevant patterns also holds in the case of itemsets. It might be exploited to design algorithms for mining relevant minimal-length itemsets. This would result in shorter itemsets, which is an important advantage if the itemsets are to be read and interpreted by human experts. Another interesting question would be whether the approaches proposing closure operators on patterns taking the form of *sets* of sequences [17, 18] could be combined with the concept of relevance. Finally, it would be interesting to investigate whether the notion of relevance can be further relaxed, following the ideas of [25, 26].

Acknowledgments

This publication has been produced in the context of the EU Collaborative Projects P-Medicine and EURECA, which are co-funded by the European Commission under the contracts ICT-2009-6-270089 respectively ICT-2011-288048.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In Yu, P.S., Chen, A.L.P., eds.: ICDE, IEEE Computer Society (1995) 3–14
2. Knobbe, A.J., Ho, E.K.Y.: Pattern teams. In: PKDD. (2006) 577–584

3. Webb, G.I.: Discovering significant patterns. Mach. Learn. **68** (July 2007) 1–33
4. Bringmann, B., Zimmermann, A.: One in a million: picking the right patterns. Knowl. Inf. Syst. **18**(1) (2009) 61–81
5. Mampaey, M., Tatti, N., Vreeken, J.: Tell me what i need to know: succinctly summarizing data with itemsets. In: KDD. (2011)
6. Lavrac, N., Gamberger, D., Jovanoski, V.: A study of relevance for learning in deductive databases. J. Log. Program. **40**(2-3) (1999) 215–249
7. Lavrac, N., Gamberger, D.: Relevancy in constraint-based subgroup discovery. In: Constraint-Based Mining and Inductive Databases. (2005)
8. Garriga, G.C., Kralj, P., Lavrač, N.: Closed sets for labeled data. J. Mach. Learn. Res. **9** (2008) 559–580
9. Yan, X., Han, J., Afshar, R.: Clospan: Mining closed sequential patterns in large databases. In: SDM. (2003)
10. Wang, J., Han, J.: Bide: Efficient mining of frequent closed sequences. In: ICDE. (2004) 79–90
11. Gao, C., Wang, J., He, Y., Zhou, L.: Efficient mining of frequent sequence generators. In: WWW. (2008) 1051–1052
12. Lo, D., Khoo, S.C., Li, J.: Mining and ranking generators of sequential patterns. In: SDM, SIAM (2008) 553–564
13. Lavrac, N., Kavsek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. Journal of Machine Learning Research **5**(Feb) (2004) 153–188
14. Cheng, H., Yan, X., Han, J., Yu, P.S.: Direct discriminative pattern mining for effective classification. In: ICDE. (2008)
15. Zimmermann, A., Bringmann, B., Rückert, U.: Fast, effective molecular feature mining by local optimization. In: ECML/PKDD (3). (2010) 563–578
16. Grosskreutz, H.: Class relevant pattern mining in output-polynomial time. In: SDM. (2012)
17. Casas-Garriga, G.: Summarizing sequential data with closed partial orders. In: SDM. (2005)
18. Raïssi, C., Calders, T., Poncelet, P.: Mining conjunctive sequential patterns. Data Min. Knowl. Discov. **17**(1) (2008) 77–93
19. Tatti, N., Cule, B.: Mining closed strict episodes. Data Mining, IEEE International Conference on **0** (2010) 501–510
20. Bastide, Y., Pasquier, N., Taouil, R., Gerd, S., Lakhal, L.: Mining minimal non-redundant association rules using frequent closed itemsets. In: Proceedings of the First International Conference on Computational Logic. (2000)
21. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Discovery Science. (2004)
22. Grosskreutz, H., Paurat, D.: Fast and memory-efficient discovery of the top-k relevant subgroups in a reduced candidate space. In: ECML/PKDD (1), Springer (2011)
23. Lemmerich, F., Atzmueller, M.: Fast discovery of relevant subgroup patterns. In: FLAIRS. (2010)
24. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
25. Lemmerich, F., Atzmueller, M.: Incorporating Exceptions: Efficient Mining of epsilon-Relevant Subgroup Patterns. In: Proc. LeGo-09: From Local Patterns to Global Models, Workshop at ECMLPKDD-2009. (2009)
26. Grosskreutz, H., Paurat, D., Rüping, S.: An enhanced relevance criterion for more concise supervised pattern discovery. In: KDD-2012. (2012)