Mining Actionable Partial Orders in Collections of Sequences

Robert Gwadera, Gianluca Antonini, and Abderrahim Labbi

IBM Zurich Research Laboratory
Rüschlikon, Switzerland

Abstract. Mining frequent partial orders from a collection of sequences was introduced as an alternative to mining frequent sequential patterns in order to provide a more compact/understandable representation. The motivation was that a single partial order can represent the same ordering information between items in the collection as a set of sequential patterns (set of totally ordered sets of items). However, in practice, a discovered set of frequent partial orders is still too large for an effective usage. We address this problem by proposing a method for ranking partial orders with respect to significance that extends our previous work on ranking sequential patterns. In experiments, conducted on a collection of visits to a website of a multinational technology and consulting firm we show the applicability of our framework to discover partial orders of frequently visited webpages that can be actionable in optimizing effectiveness of web-based marketing.

1 Introduction

Mining subsequence patterns (gaps between the symbols are allowed) in a collection of sequences is one of the most important data mining frameworks with many applications including analysis of time-related processes, telecommunications, bioinformatics, business, software engineering, Web click stream mining, etc [1]. The framework was first introduced in [2] as the problem of *sequential pattern mining* and defined as follows. Given a collection of itemset-sequences (sequence database of transactions of variable lengths) and a *minimum frequency (support) threshold*, the task is to find all subsequence patterns, occurring across the itemset-sequences in the collection, whose relative frequency is greater than the minimum frequency threshold. Although state of the art mining algorithms can efficiently derive a complete set of frequent sequential patterns under certain constraints, including mining *closed sequential patterns* [3] and *maximal sequential patterns* [4], the set of discovered sequential patterns is still too large for practical usage [1] by usually containing a large fraction of *non-significant* and *redundant* patterns [5]. As a solution to this problem a method for ranking sequential patterns with respect to significance was presented in [6].

Another line of research to address the limitations of sequential pattern mining was *partial order mining* [7],[8],[9], where a partial order on a set of items (*poset*) is an ordering relation between the items in the set. The relation is called partial

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part I, LNAI 6911, pp. 613–628, 2011.

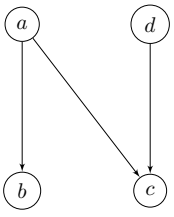

Fig. 1. A partial order

$[a, b, d, c]$
$[a, d, b, c]$
$[a, d, c, b]$
$[d, a, b, c]$
$[d, a, b, c]$

Fig. 2. A corresponding set of total orders

order to reflect the fact that not every pair of elements of a poset are related. Thus, in general the relation can be of three types: (I) *empty*, meaning there is no ordering information between the items; (II) *partial* and (III) *total* corresponding to a sequence. A partial order can be represented as a *Directed Acyclic Graph* (DAG), where the nodes correspond to items and the directed edges represent the ordering relation between the items. Figure 1 presents an example partial order for items $\{a, b, c, d\}$. *The main appeal of partial orders is to provide a more compact representation, where a single partial order can represent the same ordering information between co-occurring items in the collection of sequences as a set of sequential patterns.* As an example of this property of partial orders, consider the partial order in Figure 1 and the corresponding set of total orders that it summarizes in Figure 2. Now imagine that the set of the total orders is the input to algorithm *PrefixSpan* [10] for sequential pattern mining. Setting the minimum relative support threshold $minRelSup = 0.2$ we obtain twenty three frequent sequential patterns, while only one partial order is required to summarize the ordering information expressed by the set of sequences in Figure 2. However, in practice, even a discovered set of frequent closed partial orders (using algorithm *Frecpo* [9]) is still too large for an effective usage. Therefore, we address this problem by proposing a method for ranking partial orders with respect to significance that extends our previous work on ranking sequential patterns [6]. Using the ranking framework we can discover a small set of frequent partial orders that can be actionable (informative) for a domain expert of analyzed data who would like to turn the knowledge of the discovered patterns into an action in his domain (e.g., to optimize web-based marketing for a marketing expert).

1.1 Overview of the Method

Our ranking framework is based on building a probabilistic reference model for the input collection of sequences and then deriving an analytical formula for the expected relative frequency for partial orders. Given such a model we discover partial orders that are *over-represented* with respect to the reference model, meaning they are more frequent in the input collection of sequences than expected in the reference model. The frequency of occurrence alone is not enough to determine significance, i.e., an infrequent partial order can be more significant than a frequent one. Furthermore, an occurrence of a subsequence pattern may be *meaningless* [11] if it occurs in an sequence of an appropriately large size.

The main thrust of our approach is that the expected relative frequency of a partial order in a given reference model can be computed from an exact formula in one step. In this paper we are interested only in over-represented partial orders and our algorithm for ranking partial orders with respect to significance works as follows: (I) we find the complete set of frequent closed partial orders for a given minimum support threshold using a variant of algorithm *Frecpo* [9]; (II) we compute their expected relative frequencies and variances in the reference model and (III) we rank the frequent closed partial orders with respect to significance by computing the divergence (*Z-score*) between the observed and the expected relative frequencies. Given the reference model, a significant divergence between an observed and the expected frequency indicates that there is a dependency between occurrences of items in the corresponding partial order. Note that the only reason we use the minimum relative support threshold is because we are interested in over-represented patterns. In particular, we set a value of the threshold that is low enough to discover low support significant patterns and that is high enough such that the discovered significant patterns are actionable for marketing specialists. Given the rank values we discover actionable partial orders by first pruning non-significant and redundant partial orders and then by ordering the remaining partial orders with respect to significance.

As a reference model for the input collection of sequence we use an independence *mixture model*, that corresponds to a generative process that generates a collection of sequences as follows: (I) it first generates the size w of a sequence to be generated from the distribution of sizes in the input collection of sequences and (II) it generates a sequence of items of size w from the distribution of items in the input collection of sequences. So for such a reference model the expected relative frequency refers to the average relative frequency in randomized collections of sequences, where the marginal distribution of symbols and sequence length are the same as in the input collection of sequences. Note that the reference model can be easily extended to *Markov models* in the spirit of [12]. The reasons we consider the reference model to be the independence model in this paper are as follows: (I) it has an intuitive interpretation as a method for discovering dependencies; (II) it leads to a polynomial algorithm for computing the expected relative frequencies of partial orders in the reference model and (III) it is a reasonable model for our real data set as explained in Section 4.

1.2 Related Work and Contributions

Our ranking framework builds on the work [6] by extending it to the case of arbitrary partial orders, where [6] provided a formula for the expected relative frequency of a sequential pattern, i.e, a pattern that occurs in a collection of itemset-sequences of variable lengths. This paper also fills out the important gap between [11] and [13] by providing a formula for the expected frequency of an arbitrary partial order in a sequence of size w, where [11] analyzed the expected frequency of a *serial pattern* and [13] analyzed the expected frequency of sets of subsequence patterns including the special case of the *parallel pattern* (all permutations of a set of symbols).

In finding the formula for the expected frequency of a partial order we were inspired by the works in [14],[15], [16] that considered the problem of enumerating all *linear extensions* of a poset, where a linear extension is a total order satisfying a partial order. For example, given the partial order in Figure 1 the set of sequences in Figure 2 is the set of all linear extensions of that partial order.

The challenges in analyzing partial orders in comparison to the previous works are as follows: (I) *analytic challenge*: arbitrary partial orders can have complex dependencies between the items that complicate the probabilistic analysis and (II) *algorithmic challenge*: an arbitrary poset may correspond to large set of linear extensions and computing probability of such a set is more computationally demanding then for a single pattern.

The contributions of this paper are as follows: (I) it provides the first method for ranking partial orders with respect to significance that leads to an algorithm for mining actionable partial orders and (II) it is the first application of significant partial orders in *web usage mining*.

In experiments conducted on a collection of visits to a website of a multinational technology and consulting firm we show the applicability of our framework to discover partial orders of frequently visited webpages that can be actionable in optimizing effectiveness of web-based marketing.

The paper is organized as follows. Section 2 reviews theoretical foundations, Section 3 presents our framework for mining actionable partial orders in collections of sequences, Section 4 presents experimental results and finally Section 5 presents conclusions.

2 Foundations

$\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ is an alphabet of size $|\mathcal{A}|$. $\mathcal{S} = \{s^{(1)}, s^{(2)}, \ldots, s^{(n)}\}$ is a set of sequences of size $n = |\mathcal{S}|$, where $s^{(i)} = [s_1^{(i)}, s_2^{(i)}, \ldots, s_{n^{(i)}}^{(i)}]$ is the i-th sequence of length $n^{(i)}$, where $s_t^{(i)} \in \mathcal{A}$.

A sequence $s = [s_1, s_2, \ldots, s_m]$ is a *subsequence* of sequence $s' = [s'_1, s'_2, \ldots, s'_{m'}]$, denoted $s \sqsubseteq s'$, if there exist integers $1 \leq i_1 \leq i_2 \ldots \leq i_m$ such that $s_1 = s'_{i_1}$, $s_2 = s'_{i_2}, \ldots, s_m = s'_{i_m}$. We also say that s' is a *supersequence* of s and s is *contained* in s'.

The *support* (frequency) of a sequence s in $\mathcal{S}$, denoted by $sup_{\mathcal{S}}(s)$, is defined as the number of sequences in $\mathcal{S}$ that contain s as a subsequence. The *relative support* (relative frequency) $rsup_{\mathcal{S}}(s) = \frac{sup_{\mathcal{S}}(s)}{|\mathcal{S}|}$ is the fraction of sequences that contain s as a subsequence. Given a set of symbols $\{s_1, \ldots, s_m\}$ we distinguish the following two extreme types of occurrences as a subsequence in another sequence s': (I) *serial pattern*, denoted $s = [s_1, \ldots, s_m]$, meaning that the symbols must occur in the order and (II) *parallel pattern*, denoted $s = \{s_1, \ldots, s_m\}$, meaning that the symbols may occur in any order. We use the term *sequential pattern* with respect to the serial pattern in the sequential pattern mining framework, where the pattern occurs across a collection of input sequences.

2.1 Reference Model of a Collection of Sequences

Given an input collection of sequence $\mathcal{S} = \{s^{(1)}, s^{(2)}, \ldots, s^{(n)}\}$ its independence reference model corresponds to a generative process that generates a randomized version of $\mathcal{S}$ called $\mathcal{R} = \{r^{(1)}, r^{(2)}, \ldots, r^{(n)}\}$ as follows [6].

Let $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_M]$ be the distribution of sizes of sequences in $\mathcal{S}$ ($\alpha_m = P(|s^{(i)}| = m)$, $M = \max_{i=1}^{n} |s^{(i)}|$), where $\alpha_m = \frac{N_n(|s^{(i)}|=m)}{n}$ and $N_n(|s^{(i)}| = m)$ is the number of occurrences of sequences of size m in $\mathcal{S}$.

Let $\theta = [\theta_1, \theta_2, \ldots, \theta_{|\mathcal{A}|}]$ be the distribution of items in $\mathcal{S}$ ($\theta_j = P(s_t^{(i)} = a_j)$ for $a_j \in \mathcal{A}$), where $\theta_j = \frac{N_n(a_j)}{n_\mathcal{S}}$ and $n_\mathcal{S}$ is the number of items in $\mathcal{S}$ and $N_n(a_j)$ is the number of occurrences of item a_j in $\mathcal{S}$.

Then $r^{(i)} \in \mathcal{R}$ for $i = 1, 2 \ldots, n$ is generated as follows:

1. Generate the size of a sequence w from distribution α
2. Generate sequence $r^{(i)}$ of size w from distribution θ.

Then $P(r^{(i)}, w) = \alpha_w \cdot P(r^{(i)}|w)$, where $P(r^{(i)}|w) = \prod_{t=1}^{w} P(r_t^{(i)})$ is the probability of sequence $r^{(i)}$ to be generated given w.

Given the reference model and a sequential pattern s, $\overline{\Omega}_n(s)$ is the random variable corresponding to the relative frequency of s, where $\mathbf{E}[\overline{\Omega}_n(s)] = P^{\exists}(s)$ and $\mathbf{Var}[\overline{\Omega}_n(s)] = \frac{1}{n} P^{\exists}(s) \cdot (1 - P^{\exists}(s))$, where $P^{\exists}(s)$ is the probability that s occurs (exists) in $\mathcal{R}$. So we use the superscript $\exists$ to denote *at least one occurrence as a subsequence*. $P^{\exists}(s)$ can be expressed as follows

$$P^{\exists}(s) = \sum_{w=|s|}^{M} \alpha_w \cdot P^{\exists}(s|w), \tag{1}$$

where $P^{\exists}(s|w)$ is the probability that pattern s occurs in a sequence of size w [6]. Thus, $P^{\exists}(s)$ is expressed as a *discrete mixture model*, where the *mixing coefficients* $[\alpha_{|s|}, \ldots, \alpha_M]$ model the fact that an occurrence of s as a subsequence in a sequence s' depends on the size of s' and may possibly occur in any sequence $s' \in \mathcal{R}$ for which $|s'| \geq |s|$. Note that (1) is also valid for a parallel pattern or a partial order if $P^{\exists}(s|w)$ refers to a parallel pattern or to a partial order respectively. The significance rank can be expressed as follows $sigRank(s) = \frac{\sqrt{n}\left(rsup_\mathcal{S}(s) - P^{\exists}(s)\right)}{\sqrt{P^{\exists}(s)(1-P^{\exists}(s))}}$.

2.2 Probability of a Serial and Parallel Pattern

Let $\mathcal{W}^{\exists}(e|w)$ be the set of all distinct sequences of length w containing at least one occurrence of serial/parallel pattern e as a subsequence. Then $P^{\exists}(e|w) = \sum_{s \in \mathcal{W}^{\exists}(e|w)} P(s)$, where $P(s)$ is the probability of sequence s in a given reference model (e.g., in a k-order Markov model). $\mathcal{W}^{\exists}(e|w)$ can be enumerated using a *Deterministic Finite Automaton* (DFA) recognizing an occurrence of e.

The DFA for a serial pattern (sequential pattern) $e = [e_1, e_2, \ldots, e_m]$ called $G_<(e)$, whose example is presented in Figure 3 for $e = [a, b, c]$, has the following

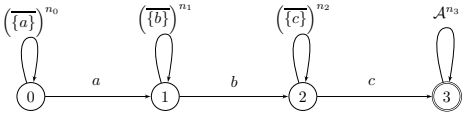

Fig. 3. $G_<(e)$ for serial pattern $e = [a, b, c]$, where $\mathcal{A}$ is a finite alphabet of size $|\mathcal{A}| > 3$ and $\overline{\{a_j\}} = \mathcal{A} \setminus a_j$ is the set complement of element $a_j \in \mathcal{A}$

components [11]. The initial state is 0, the accepting state is m and the states excluding the initial state correspond to indexes of symbols in e. The powers n_i for $i = 1, 2, \ldots m$ symbolize the number of times the i-th self-loop is used on the path from state 0 to m. Thus, $P^{\exists}(e|w)$ is equal to the sum of probabilities of all distinct paths from state 0 to m of length w in $G_<(e)$. $P^{\exists}(e|w)$ for a serial pattern in 0-order Markov reference model can be expressed as follows

$$P^{\exists}(e|w) = P(e) \sum_{i=0}^{w-m} \sum_{\sum_{k=1}^{m} n_k = i} \prod_{k=1}^{m} (1 - P(e_k))^{n_k}, \tag{2}$$

where $P(e) = \prod_{i=1}^{m} P(e_i)$ and $P(e_i)$ is the probability of symbol e_i in the reference model and (2) can be evaluated in $O(w^2)$ using a dynamic programming algorithm [11].

The DFA for a parallel pattern $e = \{e_1, e_2, \ldots, e_m\}$ called $G_{\|}(e)$, whose example is presented in Figure 4 for $e = \{a, b, c\}$, has the following components [13]. The initial state is $\{\emptyset\}$, the accepting state is $\{1, 2, \ldots, m\}$ and the states excluding the initial state correspond to the non-empty subsets of indexes of symbols in e. Let $\mathcal{E}(e)$ be the set of serial patterns corresponding to all permutations of symbols in e. Let $\mathcal{X}$ be the set of all distinct simple paths (i.e., without self-loops) from the initial state to the accepting state and let $Edges(path)$ be the sequence of edges on a path $path \in \mathcal{X}$. Then clearly $\mathcal{E}(e) = \{Edges(path) : path \in \mathcal{X} \in G_{\|}(e)\}$. $P^{\exists}(e|w)$ for a parallel pattern in 0-order Markov reference model can be computed in $O(m!w^2)$ by evaluating (2) for each member of $\mathcal{E}(e)$, where in place of $1 - P(e_k)$ we use the probability of the complement of corresponding self-loop labels of states in the path in $G_{\|}(e)$.

2.3 Partially Ordered Sets of Items

A *poset* is a set $\mathcal{P}$ equipped with an irreflexive and transitive ordering relation $R(\mathcal{P})$ (*partial order*) on set $\mathcal{A}(\mathcal{P})$. An ordered pair $(a, b) \in R(\mathcal{P})$ is denoted $a < b$, where $a, b \in \mathcal{A}(\mathcal{P})$. A *linear extension* of $\mathcal{P}$ is a permutation $p_1, p_2, \ldots, p_m$ such that $i < j$ implies that $p_i < p_j$. $\mathcal{E}(\mathcal{P}) = \{s^{(1)}, \ldots, s^{(n)}\}$ is the set of all linear extensions of $\mathcal{P}$. Elements a, b are *incomparable*, denoted $a \parallel b$, if $\mathcal{P}$ does not contain either $a < b$ or $a > b$. If no pair of elements in $\mathcal{A}(\mathcal{P})$ is incomparable in $\mathcal{P}$ then $\mathcal{P}$ is a *totally ordered set*. $G(\mathcal{P})$ is a DAG, where nodes correspond to the items and the directed edges to the relations such that an edge $a \rightarrow b$ means $a < b$. As an example of the introduced terminology consider poset $\mathcal{P} = \{a < b, a < c, d < c\}$ represented with $G(\mathcal{P})$ in Figure 1, where

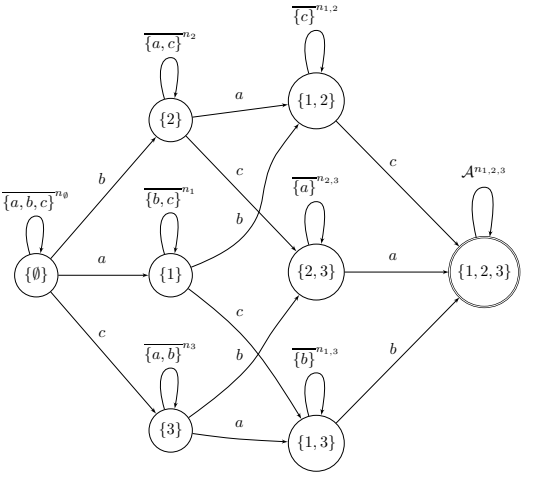

Fig. 4. $G_\parallel(e)$ for parallel pattern $e = \{a, b, c\}$, where $\mathcal{A}$ is a finite alphabet of size $|\mathcal{A}| > 3$ and $\overline{\{a_j\}} = \mathcal{A} \setminus a_j$ is the set complement of element $a_j \in \mathcal{A}$

$\mathcal{A}(\mathcal{P}) = \{a, b, c, d\}$, $R(\mathcal{P}) = \{(a, b), (a, c), (d, c)\}$ and $\mathcal{E}(\mathcal{P})$ is presented in Figure 2. We use the terms poset and partial order interchangeably in the case where all elements of a poset are part of the relation (e.g., as in Figure 1). Clearly, a serial pattern is a *total order* and a parallel pattern is a *trivial order*. A graph G is said to be *transitive* if for every pair of vertices u an v there is a directed path in G from u to v. The *transitive closure* G^T of G is the least subset of $V \times V$ which contains G and is transitive. A graph G^t is a *transitive reduction* of directed graph G whenever the following two conditions are satisfied: (I) there is a directed path from vertex u to vertex v in G^t if and only if there is a directed path from u to v in G and (II) there is no graph with fewer arcs than G^t satisfying condition (I).

A partial order s is *contained* in partial order s', denoted $s \sqsubseteq s'$, if $s \subseteq G^T(s')$. We also say that s' is a *super partial order* of s. The *relative support* $rsup_{\mathcal{S}}(s) = \frac{sup_{\mathcal{S}}(s)}{|\mathcal{S}|}$ is the fraction of sequences in $\mathcal{S}$ that contain s. Given a relative support threshold *minRelSup*, a partial order s is called a *frequent partial order* if $rsup_{\mathcal{S}}(s) \geq minRelSup$. The problem of mining frequent partial orders is to find all frequent partial orders in $\mathcal{S}$ given *minRelSup*. The support has an *anti-monotonic* property meaning that $sup_{\mathcal{S}}(s) \geq sup_{\mathcal{S}}(s')$ if $s \sqsubseteq s'$. A partial order s is called a *closed frequent partial order* if there exists no frequent partial order s' such that $s \sqsubset s'$ and $rsup_{\mathcal{S}}(s) = rsup_{\mathcal{S}}(s')$.

Given a set of symbols $\mathcal{A}' = \{a_1, \ldots, a_m\}$, let s_{serial} be a serial pattern over $\mathcal{A}'$, $s_{partial}$ be a partially ordered pattern over $\mathcal{A}'$ and $s_{parallel}$ be the parallel pattern over $\mathcal{A}'$. Then the following property holds in the independence reference model

$$P^{\exists}(s_{serial}|w) \leq P^{\exists}(s_{partial}|w) \leq P^{\exists}(s_{parallel}|w). \tag{3}$$

Thus, (3) follows from the fact that in a random sequence of size w a serial pattern is the least likely to occur since it corresponds to only one linear extension, while the parallel pattern is the most likely to occur since it corresponds to $m!$ linear extensions. So the probability of existence of a partially ordered pattern, depending on the ordering relation size, is at least equal to $P^{\exists}(s_{serial}|w)$ and at most equal to $P^{\exists}(s_{parallel}|w)$.

3 Mining Actionable Partial Orders

The problem of mining actionable partial orders in collections of sequences can be defined as follows. Given an input collection of sequences $\mathcal{S}$ and a minimum relative support threshold $minRelSup$, the task is to discover actionable partial orders by first ranking discovered partial orders with respect to significance and then by pruning non-significant and redundant partial orders.

Note that in our method the only purpose of the support threshold for partial orders is to discover actionable (exhibited by an appropriately large number of users) patterns for marketing experts.

3.1 Expected Frequency of a Poset

We derive a computational formula for the expected relative frequency of a poset $\mathbf{E}[\overline{\Omega}_n(\mathcal{P}|w)] = P^{\exists}(\mathcal{P}|w)$ as follows. Since a poset $\mathcal{P}$ occurs in a sequence of size w if at least one member of its set of linear extensions $\mathcal{E}(\mathcal{P})$ occurs in that sequence, $P^{\exists}(\mathcal{P}|w)$ is equivalent to $P^{\exists}(\mathcal{E}(\mathcal{P})|w)$ and can be expressed as follows

$$P^{\exists}(\mathcal{P}|w) = \sum_{s \in \mathcal{W}^{\exists}(\mathcal{E}(\mathcal{P})|w)} P(s), \tag{4}$$

where $\mathcal{W}^{\exists}(\mathcal{E}(\mathcal{P})|w)$ is the set of all distinct sequences of length w containing at least one occurrence of at least one member of $\mathcal{E}(\mathcal{P})$ as a subsequence and $P(s)$ is the probability of sequence s in a given reference model (e.g., in a k-order Markov model). Our approach to finding $\mathcal{W}^{\exists}(\mathcal{E}(\mathcal{P})|w)$ is to construct a DFA for $\mathcal{E}(\mathcal{P})$ called $G_{\leq}(\mathcal{P})$. Figure 5 contains a formal definition of $G_{\leq}(\mathcal{P})$, where the states are labeled with sequences $[i^{(1)}, \ldots, i^{(n)}]$ and $i^{(j)}$ denotes the prefix length for serial pattern $s^{(j)} \in \mathcal{E}(\mathcal{P})$. Given $\mathcal{E}(\mathcal{P}) = \{s^{(1)}, \ldots, s^{(n)}\}$, $G_{\leq}(\mathcal{P})$ can be viewed as a *trie* (*prefix tree*) for members of $\mathcal{E}(\mathcal{P})$ with the addition of properly labeled self-loops of nodes. Figure 6 presents an example $G_{\leq}(\mathcal{P})$ for $\mathcal{P} = \{a \rightarrow b, c\}$. Thus, given $G_{\leq}(\mathcal{P})$, the set of sequences of size w containing at least one occurrence of any of its linear extensions $\mathcal{W}^{\exists}(\mathcal{E}(\mathcal{P})|w)$ can be expressed as follows

$$\mathcal{W}^{\exists}(\mathcal{E}(\mathcal{P})|w) = \{Edges(path) : path \in \mathcal{L}_w \in G_{\leq}(\mathcal{P})\}, \tag{5}$$

where $\mathcal{L}_w$ is the set of all distinct paths of length w, including self-loops, from the initial to any of the accepting states.

- the initial state is $[0, \ldots, 0]$ and each out of $n = |\mathcal{E}(\mathcal{P})|$ accepting states corresponds to a member of $\mathcal{E}(\mathcal{P})$
- each non-initial state is $[i^{(1)}, \ldots, i^{(n)}]$ denoting the prefix length for serial pattern $s^{(j)} \in \mathcal{E}(\mathcal{P})$ in a prefix tree of the members of $\mathcal{E}(\mathcal{P})$
- A self-loop from state $[i^{(1)}, \ldots, i^{(n)}]$ to itself exists and has label equal to
  - $\mathcal{A}$ if $\exists i^{(j)} = |\mathcal{A}(\mathcal{P})|$
  - $\mathcal{A} - \bigcup_{i^{(j)}} \{s^{(j)}_{i^{(j)}+1}\}$ if $\forall i^{(j)} < |\mathcal{A}(\mathcal{P})|$.

Fig. 5. Formal definition of $G_{\leq}(\mathcal{P})$

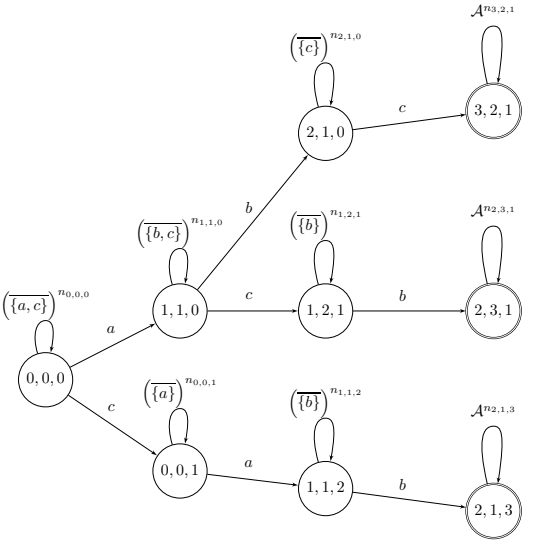

Fig. 6. $G_{\leq}(\mathcal{P})$ for partial order $\mathcal{P} = \{a \rightarrow b, c\}$, where $\mathcal{A}$ is a finite alphabet of size $|\mathcal{A}| > 3$ and $\overline{\{a_j\}} = \mathcal{A} \setminus a_j$ is the set complement of element $a_j \in \mathcal{A}$

3.2 Algorithm for Computing the Probability of a Poset

The main idea of computing $P^{\exists}(\mathcal{P}|w)$ efficiently is to build $G_{\leq}(\mathcal{P})$ in a *depth-first search* manner *on-the-fly* such that at any time only at most $|\mathcal{A}(\mathcal{P})|$ nodes are unfolded that correspond to a member of $\mathcal{E}(\mathcal{P})$. In order to build $G_{\leq}(\mathcal{P})$ on-the-fly we take the inspiration from an algorithm for enumerating linear extensions that can be summarized as follows [16]. Given a poset $\mathcal{P}$ the algorithm proceeds as follows: (I) select a subset of nodes that have no predecessors $\mathcal{N}$ in $G(\mathcal{P})$ and erase all edges (relations) involving them; (II) output the set of all permutations of nodes in $\mathcal{N}$; (III) recursively consider the set of currently existing nodes that have no predecessors $\mathcal{N}'$ and (IV) while backtracking combine the generated sets of nodes using *Cartesian product* operation to obtain the final set of linear extensions. Let $\mathcal{X}$ be the set of all distinct simple paths (i.e., without self-loops) from the start-vertex to any end-vertex in $G_{\leq}(\mathcal{P})$. We build $G_{\leq}(\mathcal{P})$

in lexicographic order of paths from the start-vertex to all end-vertices, i.e., we enumerate members of $\mathcal{E}(\mathcal{P})$ in lexicographic order. Thus, the computation of $P^{\exists}(\mathcal{P}|w)$ corresponds to a depth-first search traversal of the prefix tree of linear extensions as follows:

1. Initialize $P^{\exists}(\mathcal{P}|w) = 0$
2. For every generated $path \in \mathcal{X} \in G_{\leq}(\mathcal{P})$ in a DFS manner such that $path$ corresponds to a member of $\mathcal{E}(\mathcal{P})$ compute the probability of $P^{\exists}(path|w)$ from (2), where in place of $1 - P(e_k)$ use the probability of the complement of corresponding self-loop labels of vertices in the path
3. $P^{\exists}(\mathcal{P}|w) = P^{\exists}(\mathcal{P}|w) + P^{\exists}(path|w)$

The time complexity of the algorithm is $O(|\mathcal{E}(\mathcal{P})|w^2)$ and the space complexity is $O(|\mathcal{A}(\mathcal{P})|)$.

3.3 Pruning Non-significant and Redundant Patterns

We prune under-represented and non-significant partial orders using a *significance pruning* as follows:

1. $sigRank(s) < 0$
2. $pvalue(s) > significanceThreshold$,

where $sigRank(s) < 0$ is true for under-represented partial orders and $pvalue(s) = P(sigRank > sigRank(s))$, where we set $significanceThreshold = 0.05$.

We prune *redundant partial orders* (partial orders having the same semantic information and similar significance rank) partial orders as follows:

1. *Bottom-up pruning*: remove a partial order s if there exits another partial order s' such that $s \sqsubseteq s'$ and $sigRank(s') > sigRank(s)$, meaning s is contained in s' and has lower rank than s' so it is redundant
2. *Top-down pruning*: remove a partial order s if there exits another partial order s' such that $s \sqsubseteq s'$, $sigRank(s) > sigRank(s')$ and

$$\frac{sigRank(s) - sigRank(s')}{sigRank(s')} < pruningThreshold, \tag{6}$$

where we set $pruningThreshold = 0.05$, meaning s is contained in s' and the rank of s is not significantly greater than the rank of s' so s is redundant [5].

3.4 Algorithm for Mining Actionable Partial Orders

Since in practice sequences may contain duplicates in order to provide proper input to the algorithm for discovering closed partial orders we map the original item alphabet $\mathcal{A}$ to a unique alphabet $\mathcal{A}'$, where sequence-wise repeating occurrences of the same symbols are mapped to distinct symbols [7],[8].

Given an input collection of sequences $\mathcal{S} = \{s^{(1)}, s^{(2)}, \ldots, s^{(n)}\}$, where $s^{(i)} = [s_1^{(i)}, s_2^{(i)}, \ldots, s_{n^{(i)}}^{(i)}]$, the ranking algorithm proceeds as follows:

1. Map the collection to a unique alphabet $\mathcal{A}'$
2. Given $minRelSup$ obtain a set of frequent closed partial orders $\mathcal{F}$.
3. Compute $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_M]$, where $\alpha_w = \frac{N_n(|s^{(i)}|=w)}{n}$ and $N_n(|s^{(i)}| = w)$ is the number of sequences of size w in $\mathcal{S}$.
4. Compute $\theta = [\theta_1, \theta_2, \ldots, \theta_{|\mathcal{A}|}]$, where $\theta_j = \frac{N_n(a_j)}{n_{\mathcal{S}}}$ and $n_{\mathcal{S}}$ is the number of items in $\mathcal{S}$ and $N_n(a_j)$ is the number of occurrences of item a_j in $\mathcal{S}$
5. For every frequent closed partial order $\mathcal{P} \in \mathcal{F}$ do the following:
   (a) compute $P^{\exists}(\mathcal{P}|w)$ from (4) using the algorithm from Section 3.2
   (b) compute $P^{\exists}(\mathcal{P})$ from (1) and compute the significance rank as follows $sigRank(\mathcal{P}) = \frac{\sqrt{n}\left(rsup_{\mathcal{S}}(\mathcal{P}) - P^{\exists}(\mathcal{P})\right)}{\sqrt{P^{\exists}(\mathcal{P}))(1-P^{\exists}(\mathcal{P}))}}$.
6. Perform the reverse mapping for the discovered patterns from $\mathcal{A}'$ to $\mathcal{A}$
7. Perform pruning
   (a) remove partial orders containing cycles and redundant edges (selecting only transitive reductions)
   (b) remove non-significant and redundant partial orders using the top-down and bottom-up pruning

4 Experiments

We conducted the experiments on a collection of *Web server access logs* to a website of a multinational technology and consulting firm. The access logs consist of two fundamental data objects:

1. *pageview*: (the most basic level of data abstraction) is an aggregate representation of a collection of Web objects (frames, graphics, and scripts) contributing to the display on a users browser resulting from a single user action (such as a mouse-click). Viewable pages generally include HTML, text/pdf files and scripts while image, sound and video files are not.
2. *visit* (*session*): an interaction by an individual with a web site consisting of one or more requests for a page (pageviews). If an individual has not taken another action, typically additional pageviews on the site within a specified time period the visit will terminate. In our data set the visit timeout value is 30 minutes and a visit may not exceed a length of 24 hours. Thus, a visit is a sequence of pageviews (*click-stream*) by a single user.

The purpose of the experiments was to show that the discovered partial orders of visited pages can be actionable in optimizing effectiveness of web-based marketing. The data contains access logs to the website of the division of the company that deals with *Business Services* (BS) and spans the period from 1-01-2010 to 1-08-2010. The main purpose of the web-based marketing for that division is to advertise business services to potential customers arriving to the web site from a search engine by using relevant keywords in their query strings. Therefore, we considered the following subset of all visits: visits referred from a search engine (Google, Yahoo, Bing, etc.), having a valid query string and at least four distinct pages among their pageviews. As a result we obtained a collection of sequences

of size 13000, where every sequence corresponds to a series of identifiers of pages viewed during a visit. The alphabet $\mathcal{A}$ contains 1501 webpages and the average visit length is equal to 8.7 views, which implies that we should expect to discover partial orders of a rather small cardinality. To our surprise besides the first views in visits also following views often are referred from a search engine, including the internal search engine of the company. This behavior can be explained by the fact that many users who enter the BS site, when the first page does not satisfy their information needs, they get impatient and resort to using either external search engines or the internal search engine. As a result, they land on a different page of the BS site. Such a behavior causes that often consecutive page views in a single visit are not directly connected by HTML links. So this phenomenon suggests that the independence model is a reasonable reference model of the collection of the visits to the BS site.

We set $minRelSup = 0.01$ and obtained 2399 frequent closed partial orders using implemented variant of algorithm *Frecpo* [9], where after ranking and pruning only 185 of them were left. The discovered partial orders were subsequently shown to a domain expert who selected the following partial orders. As a simple baseline we used the method for ranking sequential patterns with respect to significance [6].

Figure 7 presents the most significant partial order (rank 1) that is a total order. The pattern consists of the following pages *Business Analytics* (BA) main page, BA people, BA work and BA ideas. As expected, this pattern is also the most significant sequential pattern for the input collection of sequences using the ranking method from [6]. The most interesting fact about this pattern is that users looking for BA solutions, after arriving at BA main page, choose BA people as the second page. Given that knowledge, the company should enhance the content of the BA people page in order to convince the users that behind the BA solutions stand competent and experienced people. Furthermore, in order to increase the search engine traffic to the BA products the BA people should have personal web pages with links pointing to the BA product pages.

Figure 8 presents the partial order at rank 6 that is the first in decreasing order of that rank that contains the BS main page. This fact clearly confirms that there is a group of users who after arriving at the BS/BA main page from a search engine, look for the people (executive team) who stand behind the BS/BA services. So this partial order reinforces the need for appropriate personal/company pages of the BS/BA people.

Figure 9 presents the partial order at rank 7 that may correspond to users who, after arriving at the BS/BA main page, look for BS/BA people but because of an under-exposed link to the BA people page, tend to perform a random walk

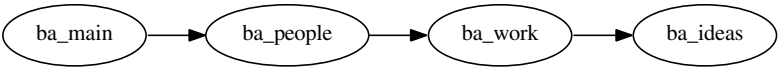

Fig. 7. Partial order at rank 1, where $sigRank = 2.30e+02$ and the relative frequency $\overline{\Omega}_n = 3.09e-02$

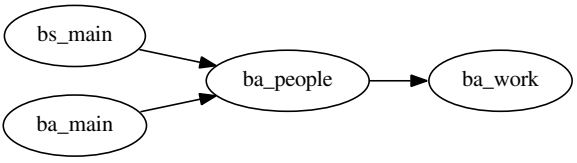

Fig. 8. Partial order at rank 6, where $sigRank = 4.74e+01$ and the relative frequency $\overline{\Omega}_n = 1.56e-02$

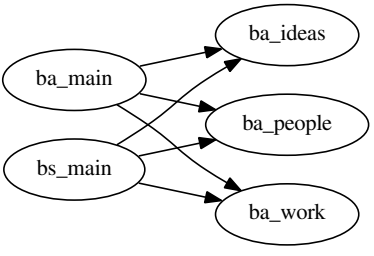

Fig. 9. Partial order at rank 7, where $sigRank = 4.51e+01$ and the relative frequency $\overline{\Omega}_n = 1.02e-02$

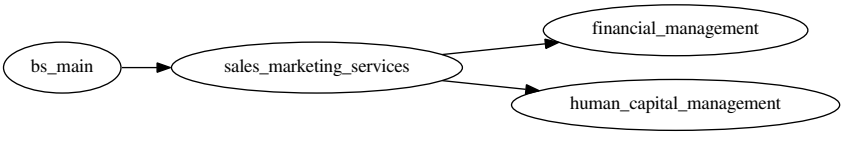

Fig. 10. Partial order at rank 12, where $sigRank = 3.52e+01$ and the relative frequency $\overline{\Omega}_n = 1.03e-02$

including the BA work and ideas page until they arrive at the BA people page. This pattern may suggest that by emphasizing the link to BA people page on the BA/BS main pages the users would directly get to the BA people page.

Figure 10 presents the partial order at rank 12 that is the first in decreasing order of the rank to contain the financial management and the human capital management pages. This partial order may suggests that those two pages should be accessible directly from the BS main page in order to increase their visibility.

Figure 11 presents the partial order at rank 24 that is the first in decreasing order of the rank that contains the strategy planing page. As it turns out it significantly co-occurs with the financial management, the human capital management and the sales marketing services page. This pattern may suggest that the three co-occurring pages with the strategy planing page should contain well-exposed links to the strategy planing page to increase its visibility.

Figure 12 presents a comparison of $P^{\exists}(\mathcal{P})$ against the baseline $P^{\exists}(s_{serial})$ and $P^{\exists}(s_{parallel})$, where s_{serial} is a serial and $s_{parallel}$ is the parallel pattern over symbols of $\mathcal{P}$, where a proper value of $P^{\exists}(s_{partial}) = P^{\exists}(\mathcal{P})$ should satisfy (3). In order to compute $P^{\exists}(s_{serial})$ and $P^{\exists}(s_{parallel})$ for the discovered

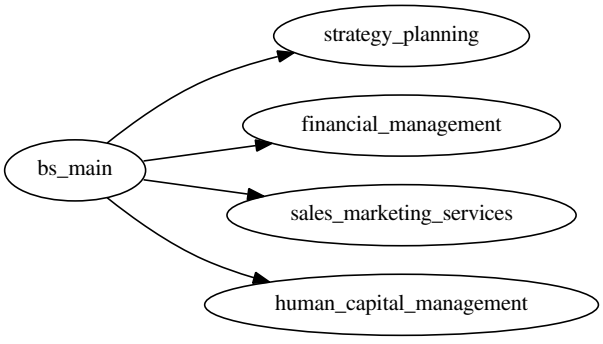

Fig. 11. Partial order at rank 24, where $sigRank = 2.73e + 01$ and the relative frequency $\overline{\Omega}_n = 1.02e - 02$

Partial order	$P^{\exists}(s_{serial})$	$P^{\exists}(s_{partial})$	$P^{\exists}(s_{parallel})$
Figure 7	2.2e-04	2.2e-04	1.9e-03
Figure 8	8.2e-04	1.2e-03	5.7e-03
Figure 9	1.2e-04	5.7e-04	1.7e-03
Figure 10	5.3e-04	8.9e-04	4.0e-03
Figure 11	1.9e-04	1.3e-03	2.6e-03

Fig. 12. $P^{\exists}(s_{partial}) = P^{\exists}(\mathcal{P})$ for the discovered partial orders in comparison to their serialized/paralellized versions $P^{\exists}(s_{serial})/P^{\exists}(s_{parallel})$, where as to be expected for a valid value of $P^{\exists}(s_{partial})$, $P^{\exists}(s_{serial}) \leq P^{\exists}(s_{partial}) \leq P^{\exists}(s_{parallel})$ is satisfied in all cases

partial orders we "serialized"/"parallelized" them appropriately. Thus, for the total order from Figure 7 $P^{\exists}(s_{serial}) = P^{\exists}(s_{partial})$ and for the rest of the partial orders, depending on their structure, their probability is either closer to $P^{\exists}(s_{serial})$ or to $P^{\exists}(s_{parallel})$. For example, for the partial orders from Figure 7 and 10 $P^{\exists}(s_{partial})$ is closer to $P^{\exists}(s_{serial})$ and for the partial order from Figure 11 $P^{\exists}(s_{partial})$ is closer to $P^{\exists}(s_{parallel})$.

Figure 13 compares the number of frequent closed partial orders, the number of final partial orders obtained from our algorithm and effectiveness of pruning methods as a function of the minimum support threshold.

The results can be summarized as follows: (I) the number of final partial orders is equal to 7.8% of the frequent closed partial orders on average; (II) the significance pruning prunes 63.7% of the final partial orders on average; (III) the bottom-up pruning prunes 77.4% of the partial orders left after the significance pruning and (IV) the top-down pruning prunes 4% of the partial orders left after the bottom-up pruning. Furthermore, the variance of the pruning results is rather small suggesting the the pruning methods exhibit a similar behavior across different values of $minRelSup$.

minRelSup	Number of Frequent closed partial orders	Number of Final partial orders	Pruning effectiveness ([%] pruned)		
			Significance pruning	Bottom-up pruning	Top-down pruning
0.2	471	44	53.9	78.3	6.4
0.015	821	85	59.8	73.3	3.4
0.014	983	93	61.3	74.2	5.1
0.013	1163	106	62.7	74.0	6.2
0.012	1450	124	65.0	74.4	4.6
0.011	1819	150	66.2	74.3	5.1
0.01	2399	185	67.6	75.3	3.6
0.009	3124	232	68.2	76.0	2.5
0.008	4566	300	68.7	78.4	2.9
0.007	7104	439	67.6	80.3	2.9
0.006	12217	683	65.4	83.4	3.0
0.005	26723	1439	58.0	86.9	2.4
Average [%]		7.8	63.7	77.4	4.0

Fig. 13. Number of frequent partial orders, number of final partial orders obtained from our algorithm and effectiveness of pruning methods as a function of *minRelSup*. The pruning methods are applied in the following order: significance pruning, bottom-up pruning and top-down pruning

5 Conclusions

We presented a method for ranking partial orders with respect to significance and an algorithm for mining actionable partial orders. In experiments, conducted on a collection of visits to a website of a multinational technology and consulting firm we showed the applicability of our framework to discover partially ordered sets of frequently visited webpages that can be actionable in optimizing effectiveness of web-based marketing.

References

1. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery 15(1) (2007)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: ICDE, pp. 3–14 (1995)
3. Yan, X., Han, J., Afshar, R.: Clospan: Mining closed sequential patterns in large datasets. In: SDM, pp. 166–177 (2003)
4. Guan, E., Chang, X., Wang, Z., Zhou, C.: Mining maximal sequential patterns. In: 2005 International Conference on Neural Networks and Brain, pp. 525–528 (2005)
5. Huang, X., An, A., Cercone, N.: Comparison of interestingness functions for learning web usage patterns. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM 2002, pp. 617–620. ACM, New York (2002)
6. Gwadera, R., Crestani, F.: Ranking sequential patterns with respect to significance. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS(LNAI), vol. 6118, pp. 286–299. Springer, Heidelberg (2010)

7. Mannila, H., Meek, C.: Global partial orders from sequential data. In: KDD 2000: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 161–168. ACM, New York (2000)
8. Casas-Garriga, G.: Summarizing sequential data with closed partial orders. In: Proceedings of the Fifth SIAM International Conference on Data Mining, April 2005, pp. 380–390 (2005)
9. Pei, J., Wang, H., Liu, J., Wang, K., Wang, J., Yu, P.S.: Discovering frequent closed partial orders from strings. IEEE Transactions on Knowledge and Data Engineering 18, 1467–1481 (2006)
10. Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q.: Mining sequential patterns by pattern-growth: The prefixspan approach. TKDE 16 (November 2004)
11. Gwadera, R., Atallah, M., Szpankowski, W.: Reliable detection of episodes in event sequences. In: Third IEEE International Conference on Data Mining, pp. 67–74 (November 2003)
12. Gwadera, R., Atallah, M., Szpankowski, W.: Markov models for discovering significant episodes. In: SIAM International Conference on Data Mining, pp. 404–414 (April 2005)
13. Atallah, M., Gwadera, R., Szpankowski, W.: Detection of significant sets of episodes in event sequences. In: Fourth IEEE International Conference on Data Mining, pp. 67–74 (October 2004)
14. Varol, Y.L., Rotem, D.: An algorithm to generate all topological sorting arrangements. The Computer Journal 24(1), 83–84 (1981)
15. Pruesse, G., Ruskey, F.: Generating linear extensions fast. SIAM J. Comput. 23, 373–386 (1994)
16. Knuth, D.E., Szwarcfiter, J.L.: A structured program to generate all topological sorting arrangements. Inf. Process. Lett