The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Subgroup Discovery in Sequential Databases

Rina Singh, Jeffrey A. Graves, Douglas A. Talbert
{rsingh43, jagraves21}@students.tntech.edu, DTalbert@tntech.edu
Department of Computer Science
Tennessee Technological University
Cookville, TN 38505

Abstract

Sequential pattern mining produces a vast number of frequent patterns due to the combinatorial nature of the problem and redundant information in the results. Several pattern mining techniques (e.g., closed-patterns, maximal-patterns, gap constraints, recency and compactness constraints) have been studied to reduce the size of the results. However, these approaches can still generate large result sets often containing patterns of little use or interestingness to the end users. Even when many interesting results are returned, finding the most useful can be difficult. By applying ideas from subgroup discovery to sequential pattern mining, we have developed exact and heuristic-based algorithms for identifying and ranking the top-k most significant patterns from the complete collection of frequent patterns.

Introduction

Ever since Agrawal and Srikant first published their research, Mining Sequential Patterns (Agrawal and Srikant 1995), sequential pattern mining has been of broad and current interest. In many real world situations, the ordering of events (i.e., the presence of sequences) is significant and the detection of frequent subsequences may be useful. Some examples include the verification and development of clinical pathways (Uragaki et al. 2016), determining product placement within retail stores (Aloysius and Binu 2013), and identifying interesting protein-protein interaction sites (Hsu et al. 2007). Due to the combinatorial nature of the mining task, many sequential pattern mining algorithms suffer from pattern explosion, producing large, redundant pattern sets. Sifting through such results can be difficult and time consuming, so many techniques have been developed to address this.

In their work, Agrawal and Srikant discussed the mining of max patterns (i.e., maximal-patterns), while Yan et al. first tackle the problem of mining closed-patterns (Agrawal and Srikant 1995; Yan, Han, and Afshar 2003). Both approaches reduce the size of the results by reducing redundancy. Several constraint-based mining approaches have been explored over the years, including gap constraints (Antunes and Oliveira 2003; Li and Wang 2008; Srikant and Agrawal 1996), recency and compactness constraints (Chen and Hu 2006), relaxation of itemset/transaction constraints, and taxonomy-based constraints (Srikant and Agrawal 1996). While the majority of these were developed to incorporate domain knowledge or requirements, they also reduce the number of the results. However, none of these consider the *usefulness* or *interestingness* of the results.

In this paper, we describe our attempt to mine the top-k most interesting patterns from a sequential database. It is difficult to find "interesting" results relying solely on frequency and sequence length, so additional information must be incorporated. By assigning class labels to sequences, we are able to apply the well-developed concept of *interestingness* from subgroup discovery to sequential pattern mining and identify the top-k most interesting patterns. Our approach can use many different interestingness measures (e.g., confidence, sensitive, specificity, precision measures, etc.).

Using such measures, we can significantly reduce the result set and present only the most interesting patterns. However, our exact solution still suffers from the same combinatorial time complexity inherent in traditional frequent sequence mining. This, we have developed a heuristic-based approach, relying on the interestingness measure, to reduce the time of the mining task. Using precision measure Q_g (see Eq. 2 on Interestingness Measures), we have compared the results of the heuristic-based approach to those of the exact approach and have found that, while some of the top patterns are lost, the heuristic-based approach performs reasonably well at identifying the top-k patterns.

Related Work

Fournier–Viger and his colleagues provide an excellent survey on sequential pattern mining (Fournier-Viger et al. 2017), and it will be assumed that the reader is familiar with basic definitions, notation, and results from sequential pattern mining, and while there has been considerable research on reducing the number of results from sequential pattern mining, there is significantly less work on identifying "best" patterns. This is most likely due to the difficulty in formally defining what is meant by a best pattern.

Tzvetkov et al. attempt to reduce the number of sequential patterns obtained while eliminating the need to specify a minimum support (Tzvetkov, Yan, and Han 2003). The problem with specifying a minimum support is that, if the minimum support is set too high, the mining task will return few or no results, but if the minimum support is set too low, the mining task will take a long time and produce excessive results. To overcome this, Tzvetkov defined the notion of a *top-k closed sequential pattern*: given a minimum sequence length l and positive integer k, a sequential pattern P is said to be a top-k closed sequential pattern of minimum length l if the length of P is at least l and there are no more than $k-1$ closed sequential patterns of length at least l having a higher support than P. This work is similar to ours, however, their notion of a "top pattern" is based on frequency and length rather than interestingness.

Yin et al., seek to mine high utility patterns as opposed to highly frequent patterns (Yin, Zheng, and Cao 2012). By assigning weights to items in the sequences, a pattern's utility value can be calculated. Such values allow the resulta to be ordered. Because weights can only measure the presence or absence of an item, however, *interestingness* based on item ordering or a property external to the sequence cannot be represented easily, if at all.

Most related to our work is the research of Ji et al., who focus on mining *distinguishing* patterns (Ji, Bailey, and Dong 2005). To do this, the user must specify a positive or negative class label for each sequence and provide two support values: a minimum support for the positive class and a maximum support for the negative class(es). The goal is then to find all patterns that meet the minimum support in the positive class while not exceeding the maximum support in the negative class(es).

These thresholds place hard constraints on the results. One could soften these constraints by mining patterns that maximize the positive/negative class ratio. Thus, with the right interesting measure, our approach can be thought of a generalization of Ji's work using soft constraints.

Subgroup Discovery in Sequences

Subgroup discovery identifies interesting relationships between data entities having a common property (Klösgen 1996). Data entities are often attribute-value pairs and the common property is one or more target attributes. The subgroups discovered are usually reported as a rule or collection of rules that define membership. Numerous quality measures exist to evaluate the interestingness of a subgroup (or its defining rule(s)), including *support*, *confidence*, *coverage*, *lift*, *sensitivity*, *specificity*, and *precision measures* (Herrera et al. 2011; Lavrač, Flach, and Zupan 1999). These measures are often defined in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). In subgroup discovery, data entities which have the property of interest are considered positive while all other data entities are considered negative.

In our work, data entities are sequences, mined patterns are subgroup definitions, and sequences covered by a defining pattern are subgroup members. Many existing quality measures can then be used to rank patterns. However, some quality measures, like *interest* (which is defined in terms of information gain), are ill-suited to subgroups of sequences. Any quality measure that relies solely on true/false positives and true/false negatives can be adapted to subgroup discovery in sequences.

If the property of interest is not explicitly within a sequence, we say the class label is *external*. For instance, consider a database in which the target attribute is buyers between the ages of 18 and 25 years; purchase history sequences associated with buyers in that age range are considered positive, and sequences associated with buyers outside of that age range are considered negative. In this example, the age of the buyer is associated with the sequence but is not part of the sequence itself. If, however, the property of interest is the presence or absence of an item within a sequence, we say the class label is *internal*. Due to some complexities with internal labeling that are beyond the scope of this paper, we only consider sequences with external labels

Interestingness Measures

There are many subgroup quality measures. Here we will introduce a few in the context of subgroup discovery in sequential databases.

One quality measure we could consider is *sensitivity* can be useful if one is interested in patterns that describe the positive class, and is defined by

$$\frac{TP}{TP+FN}\,. \tag{1}$$

Note that sensitivity ignores the negative sequences captured by the pattern. If one is interested in patterns that describe the positive class and not the negative class (i.e., discrimination patterns), *precision measure Q_g* can be used. It is defined as

$$\frac{TP}{FP+g}\,, \tag{2}$$

where g is a nonnegative generalization parameter.

Similar to precision measure Q_g is *specificity*, which can be used to find patterns that focus on excluding the negative class. It is defined as

$$\frac{TN}{TN+FP}\,. \tag{3}$$

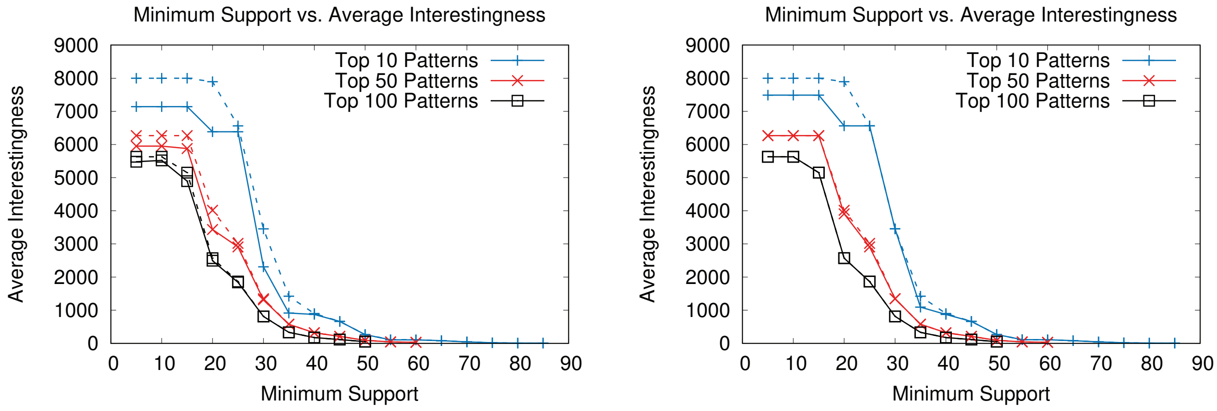
Problem Definition

Many subgroup quality measures do not take subgroup size into consideration. Since a high quality measure of a small subgroup is not very useful in many cases, we require a minimum number of positive examples to be in the subgroup before it is reported. Thus, we define the *subgroup discovery problem in sequences* as follows:

Given a sequential database D, minimum support m, interesting measure $\mathcal{I}$, positive integer k, and a class labeling method, identify the top-k patterns from D, with respect to measure $\mathcal{I}$, that have a minimum support m within the positive class. A pattern P is said to be a top-k pattern if it has minimum support m within the positive class and there are no more than $k-1$ patterns having minimum support m within the positive class with an interestingness measure greater than that of P.

Experimental Evaluation

To test beam search, we created a synthetic dataset in which 100,000 sequences containing the numbers 0-9 were generated using different probability distributions to simulate



(a) Beam Search using Specialization Extensions

(b) Beam Search using all Specializations

Figure 1: Beam Search vs. Exact Solution

different population groups. Sequence lengths were random and between 10 and 20. Each sequence was one of five equally distributed classes based upon the distribution used to generate the items in the sequence. We used binomial distributions with parameters $n = 9$ and p equal to $1/6$, $1/3$, $1/2$, $2/3$, or $5/6$. The distributions with $p = 1/6$ and $p = 1/3$ were chosen to represent the positive class.

We then compared the average interestingness of the top-k patterns against the k patterns returned by our beam search with width of k. We used precision measure Q_g as our interestingness metric. When implementing the beam search, we considered the use of only sequence extensions as refinements and the use of all possible specializations. Figure 1 shows how the average interestingness of the top-k patterns varies with support for both the exact solution (dashed lines) and the heuristic-based results (solid lines).

Full specializations preform only slightly better than extensions alone. Thus, a beam search that relies only on extensions can obtain results quicker without sacrificing too many top patterns. From the plots, it appears that the average interestingness of the beam search results approaches the true average interestingness as the minimum support increases and as the number of top patterns increases. Both of these may be because a large portion of all possible patterns are being included in the results. In the case of larger support values, there are fewer patterns that meet the minimum support. In the case of larger values of k, a larger portion of all possible patterns are included.

The effect of support and the value of k on the average interestingness may be better seen in Table 1, which shows the average interestingness of the beam search as percentage of the average interestingness of the top-k patterns. Note that dashes indicate that there are less than k patterns that meet the given support within the positive class.

Exact and Heuristic-Based Algorithms

Finding the top-k patterns (subgroups) with respect to some interestingness measure can be reduced to standard sequential pattern mining; any one of a variety of sequential pattern mining algorithms can be applied, and those patterns that meet the minimum support within the positive class can then be evaluated using the interestingness measure. This is an exact solution to the problem and suffers from the same combinatorial time complexity as sequential pattern mining.

Near-optimal results are often acceptable if they can be obtained in a shorter time. Since the search space for our problem is exponential, an approximate solution can be beneficial. Gamberger et al. use a beam search to reduce the time required to find subgroups (Gamberger and Lavrac 2002). The algorithm begins by adding the empty sequence to the beam. Then for each iteration, while the beam is not empty, each sequence in the beam is examined and specializations are created. Each specializations is evaluated using the interestingness measure, and then top beam-width sequences are retained for the next iteration.

Sequence specialization refers to generating a new (super)sequence from an existing (sub)sequence by adding a single item. There are many possible ways to specialize a sequence, as shown in Figure 2. We call a sequence specialization an *extension* if the item is added to the end of the sequence. Itemset sequences can also be specialized, but this work focuses only on sequences of items.

$$\mathcal{I} = \{a, b, c\}$$

(a) Itemset

$$S = \langle a, a \rangle$$

(b) Item Sequence

$$\langle a, a, \boldsymbol{a} \rangle \ \langle a, a, \boldsymbol{b} \rangle \ \langle a, a, \boldsymbol{c} \rangle$$

(c) All Extensions

$$\begin{matrix} \langle \boldsymbol{a}, a, a \rangle & \langle a, \boldsymbol{a}, a \rangle & \langle a, a, \boldsymbol{a} \rangle \\ \langle \boldsymbol{b}, a, a \rangle & \langle a, \boldsymbol{b}, a \rangle & \langle a, a, \boldsymbol{b} \rangle \\ \langle \boldsymbol{c}, a, a \rangle & \langle a, \boldsymbol{c}, a \rangle & \langle a, a, \boldsymbol{c} \rangle \end{matrix}$$

(d) All Specializations

Figure 2: Extensions to Item Sequences

Table 1: Average Interestingness of Beam Search
as Percentage of Average interestingness of Exact Results

Support	Top 10	Top 20	Top 30	Top 40	Top 50	Top 100	Top 200	Top 300	Top 400	Top 500
10	0.936	0.936	0.965	0.985	1.0	1.0	0.993	0.986	0.995	0.999
20	0.831	0.801	0.849	0.984	0.973	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	–
40	0.968	0.892	1.0	0.997	1.0	1.0	–	–	–	–
50	1.0	1.0	1.0	1.0	1.0	1.0	–	–	–	–
60	1.0	1.0	1.0	1.0	1.0	–	–	–	–	–
70	1.0	1.0	1.0	–	–	–	–	–	–	–
80	1.0	–	–	–	–	–	–	–	–	–
90	–	–	–	–	–	–	–	–	–	–
100	–	–	–	–	–	–	–	–	–	–

Conclusions and Future Work

We have introduced subgroup discovery in sequential databases. By viewing a sequential pattern as defining a subgroup, we are able to rank the results using various subgroup quality measures. This allows us to reduce the results and return only the most interesting patterns. To help reduce the mining time, we have explored the use of a beam search for extracting patterns while only suffering a small loss in pattern interestingness. In the future, we hope to explore various event-based labeling methods and develop optimal algorithms for mining interesting patterns based on complex internal labeling schemes.

References

Agrawal, R., and Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*.

Aloysius, G., and Binu, D. 2013. An approach to products placement in supermarkets using PrefixSpan algorithm. *Journal of King Saud University-Computer and Information Sciences*.

Antunes, C., and Oliveira, A. L. 2003. Generalization of pattern-growth methods for sequential pattern mining with gap constraints. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*.

Chen, Y.-L., and Hu, Y.-H. 2006. Constraint-based sequential pattern mining: The consideration of recency and compactness. *Decision Support Systems*.

Fournier-Viger, P.; Lin, J. C.-W.; Kiran, R. U.; Koh, Y. S.; and Thomas, R. 2017. A survey of sequential pattern mining. *Data Science and Pattern Recognition*.

Gamberger, D., and Lavrac, N. 2002. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*.

Herrera, F.; Carmona, C. J.; González, P.; and del Jesus, M. J. 2011. An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*.

Hsu, C.-M.; Chen, C.-Y.; Liu, B.-J.; Huang, C.-C.; Laio, M.-H.; Lin, C.-C.; and Wu, T.-L. 2007. Identification of hot regions in protein-protein interactions by sequential pattern mining. *BMC Bioinformatics*.

Ji, X.; Bailey, J.; and Dong, G. 2005. Mining minimal distinguishing subsequence patterns with gap constraints. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*.

Klösgen, W. 1996. Explora: A explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence.

Lavrač, N.; Flach, P.; and Zupan, B. 1999. Rule evaluation measures: A unifying view. In *International Conference on Inductive Logic Programming*.

Li, C., and Wang, J. 2008. Efficiently mining closed subsequences with gap constraints. In *Proceedings of the 2008 SIAM International Conference on Data Mining*.

Srikant, R., and Agrawal, R. 1996. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology*.

Tzvetkov, P.; Yan, X.; and Han, J. 2003. TSP: Mining top-k closed sequential patterns. In *Third IEEE International Conference on Data Mining, 2003. ICDM 2003.*

Uragaki, K.; Hosaka, T.; Arahori, Y.; Kushima, M.; Yamazaki, T.; Araki, K.; and Yokota, H. 2016. Sequential pattern mining on electronic medical records with handling time intervals and the efficacy of medicines. In *2016 IEEE Symposium on Computers and Communication (ISCC),*.

Yan, X.; Han, J.; and Afshar, R. 2003. CloSpan: Mining: Closed sequential patterns in large datasets. In *Proceedings of the 2003 SIAM International Conference on Data Mining*.

Yin, J.; Zheng, Z.; and Cao, L. 2012. USpan: an efficient algorithm for mining high utility sequential patterns. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*.