Chaining Patterns

Taneli Mielikäinen

HIIT Basic Research Unit
Department of Computer Science
University of Helsinki, Finland
Taneli.Mielikainen@cs.helsinki.fi

Abstract. Finding condensed representations for pattern collections has been an active research topic in data mining recently and several representations have been proposed. In this paper we introduce chain partitions of partially ordered pattern collections as high-level condensed representations that can be applied to a wide variety of pattern collections including most known condensed representations and databases. We analyze the goodness of the approach, study the computational challenges and algorithms for finding the optimal chain partitions, and show empirically that this approach can simplify the pattern collections significantly.

1 Introduction

The goal of *pattern discovery* is to find interesting patterns from data sets [1,2,3]. There exist output-efficient algorithms for finding the interesting patterns from a wide variety of different pattern classes [4,5,6].

The most prominent examples of interesting patterns are *frequent sets* and *association rules* [7]. For the frequent sets and the association rules a data set is a finite sequence $d = d_1 \ldots d_n$ of subsets of some finite set R. A set $X \subseteq R$ is interesting if it is σ-frequent in d, i.e.,

$$fr(X, d) = \frac{|\{i : X \subseteq d_i, 1 \leq i \leq n\}|}{n} \geq \sigma \in [0, 1].$$

An association rule $X \Rightarrow Y$ is interesting if it is both σ-frequent and δ-accurate in d, i.e., $fr(X \cup Y, d) \geq \sigma$ and

$$acc(X \Rightarrow Y, d) = \frac{fr(X \cup Y, d)}{fr(X, d)} \geq \delta \in [0, 1].$$

(If $fr(X, d) = 0$ then $acc(X \Rightarrow Y, d)$ is not defined.)

Traditionally the interestingness of a pattern has been a local property of the pattern. For example, the interestingness of an association rule $X \Rightarrow Y$ depends on the frequencies of the sets X and $X \cup Y$. However, the collection of the interesting patterns can have also more global interesting structure.

Some of the structural properties of pattern collections have been exploited in condensed representations of pattern collections, i.e., pattern collections that are irredundant w.r.t. some inference method. The condensed representations of pattern collections have been studied extensively and several condensed representations have been suggested [4,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22]. They can be used, for example, to compress pattern collections, for efficient querying, and to gain insight to the pattern collection (and to the data set).

In this paper we propose a new condensed representation of pattern collections, *pattern chains*, that depends only on a partial order of patterns and thus can be applied to condense a wide variety of pattern collections including many other condensed representations.

The rest of the paper is organized as follows. The new condensed representation is described in Section 2. Algorithmic issues of the representation are discussed in Section 3. In Section 4 we show experimentally that the approach can applied to condense pattern collections in practice. The work is concluded in Section 5.

2 Exploiting the Structure

A collection of interesting patterns and the whole pattern classes, too, usually have some structure. For example, the collection of subsets of R can be structured by the frequency of the subsets in a given data set: every subset of a frequent set is frequent and every superset of an infrequent set is infrequent.

Pattern collections can have also some data-independent structure. A typical data-independent structure of a pattern collection is a partial order. The definitions we use related to partial orders are the following:

- A *partial order* $\prec$ for a finite set $\mathcal{P}$ is a transitive ($p \prec q \wedge q \prec r \Rightarrow p \prec r$) and irreflexive ($p \prec q \Rightarrow p \not\prec q$) binary relation $\prec \subseteq \mathcal{P} \times \mathcal{P}$. For example, any collection of sets is partially ordered by the set inclusion $\subset$.
- Elements p and q of a partially ordered set $\mathcal{P}$ are called *comparable* iff $p \prec q$, $q \prec p$ or $p = q$.
- A *total order* $<$ for a finite set $\mathcal{P}$ is a partial order such that all pairs of elements in $\mathcal{P}$ are comparable. Frequencies of subsets of R determine a total order for the subsets.
- An element $p \in \mathcal{P}$ is *maximal* (*minimal*) in $\mathcal{P}$ if for no element $q \in \mathcal{P}$ holds: $p \prec q$ ($q \prec p$). An example of maximal patterns are maximal frequent sets, i.e., the frequent sets that have no frequent supersets. There is only one minimal pattern in the collection of frequent sets, the empty set, because all the other sets contain the empty set and all subsets of frequent sets are frequent.
- A subset $\mathcal{C}$ (a subset $\mathcal{A}$) of a partially ordered set $\mathcal{P}$ is called a *chain* (an *antichain*) iff any two elements in $\mathcal{C}$ (no two distinct elements in $\mathcal{A}$) are comparable.
- A *chain partition* (an *antichain partition*) of a partially ordered set $\mathcal{P}$ is partition of the set $\mathcal{P}$ to disjoint chains $\mathcal{C}_1, \ldots, \mathcal{C}_m$ (antichains $\mathcal{A}_1, \ldots, \mathcal{A}_m$). A chain partition is minimum iff there is no chain partition of smaller cardinality, and minimal iff there are no two chains in the chain partition such that their union is a chain.

Well-studied examples of partially ordered pattern collections, besides of frequent and maximal patterns, are the collections of closed patterns [8,20,23,24,25,26]. Let $\prec$ be a partial order for the pattern collection $\mathcal{P}$ and let $<$ be a total order determined by the frequencies $fr(p,d)$ of patterns $p \in \mathcal{P}$ w.r.t. a data set d. A pattern $p \in \mathcal{P}$ is *closed* iff

$$p \prec q, q \in \mathcal{P} \Rightarrow fr(p,d) > fr(q,d).$$

The collection of closed patterns in $\mathcal{P}$ is denoted by $Cl(\mathcal{P})$. The closed patterns exploit the data set d more extensively than, e.g., maximal or frequent patterns: the maximal patterns depend only on the pattern collection (which, of course, can depend on the data set) and the frequent patterns depend on the data set only by the maximal frequent patterns.

Usually each closed pattern $p \in Cl(\mathcal{P})$ determines an equivalence class that contains p and all its subpatterns in $\mathcal{P}$ with the frequencies equal to the frequency of p, i.e., all patterns $q \in \mathcal{P}$ such that $q \prec p$ and $fr(p,d) = fr(q,d)$. Unfortunately this does not hold in general since there are pattern collections $\mathcal{P}$ (and data sets d) which contains closed patterns $p, q \in Cl(\mathcal{P})$ such that

$$\{r \in \mathcal{P} : r \prec p, fr(r,d) = fr(p,d)\} = \{r \in \mathcal{P} : r \prec q, fr(r,d) = fr(q,d)\}$$

but $p \neq q$. This can be the case, e.g., when the pattern collection consists of approximate patterns.

Besides of detecting structure in a pattern collection, the found structure can sometimes be further exploited. For example, frequent sets can be stored into an *itemset tree* by defining a total order for R: each frequent set is a path from root to some node. (Itemset trees are known also by several other names, see e.g. [7,26,27,28].) The itemset tree can save space and allow efficient frequency queries. Unfortunately, itemset trees require an explicit order for the elements of R. The order might be an artificial structure that hides the “true” structure of the pattern collection.

The idea of exploiting the structure of the pattern collection, and especially simplifying the partial order structure of the pattern collection, might still be useful although it is not clear whether e.g. the itemset tree makes partial order of the set collection more comprehensible or even more obscure from the human point of view.

To exploit the partial order structure of a pattern collection $\mathcal{P}$, we propose the minimum chain partition $\mathcal{C}_1, \ldots, \mathcal{C}_m$ of the partially ordered set $\mathcal{P}$ as a condensed representation for $\mathcal{P}$. There exists a chain partition for any partially ordered set $\mathcal{P}$. Thus nothing else have to be assumed about the structure of $\mathcal{P}$ in order to be able to find this kind of condensed representation for $\mathcal{P}$. The chain partition can be interpreted as a clustering of the pattern collection. Each chain $\mathcal{C}_i, 1 \leq i \leq m$, as a totally ordered set, can have much simpler structure than the original partially ordered set $\mathcal{P}$. a partition of a partially ordered set $\mathcal{P}$ to minimum number of chains $\mathcal{C}_1, \ldots, \mathcal{C}_m$ corresponds to a structural clustering of $\mathcal{P}$ and that consists of the minimum number of clusters $\mathcal{C}_1, \ldots, \mathcal{C}_m$ and in each cluster $\mathcal{C}_i$ all patterns $p, q \in \mathcal{C}_i$ are comparable.

The minimum chain partition might not be unique but the lack of uniqueness is not necessarily a problem because of the exploratory nature of data mining: Different partitions emphasize different aspects of the pattern collection. Clearly, this can be beneficial when trying to understand the essence of the data set.

The maximum number of chains in a chain partition of $\mathcal{P}$ is $|\mathcal{P}|$ as each element p in $\mathcal{P}$ is itself a chain (and an antichain, too). The minimum number of chains in a chain partition is at least the cardinality of the largest antichain in $\mathcal{P}$ since no two distinct elements of the largest antichain can be in the same chain as they are not comparable. This inequality is strict by the Dilworth's Theorem: a partially ordered set $\mathcal{P}$ can be partitioned into m chains iff the largest antichain in $\mathcal{P}$ is of cardinality m. (For a clear exposition of chains, antichains and partial orders, see [29].) Moreover, as maximal elements of a partial ordered set form an antichain, the number of chains needed is always at least the number of maximal elements in $\mathcal{P}$.

A chain partition can be even more than a structural clustering if the pattern collection has more structure than a partial order. As an example of further exploiting the chain structure, let us consider a collection of weighted sets, e.g., frequent sets with their frequencies or binary matrix as a set collection with integer weights for the sets, and let the partial order relation be determined by the set inclusion. Let the set collection $\mathcal{P}$ be

$$\{\{1\},\{2\},\{1,3\},\{2,4\},\{1,2,3\},\{1,2,4\}\}$$

and the weights of the sets to be $w(\{1\}) = 4$, $w(\{2\}) = 5$, $w(\{1,3\}) = 3$, $w(\{2,4\}) = 4$, $w(\{1,2,3\}) = 2$ and $w(\{1,2,4\}) = 1$. Then the pattern collection can be partitioned into two chains

$$\mathcal{C}_1 = \{\{1\},\{1,3\},\{1,2,3\}\}$$

and

$$\mathcal{C}_2 = \{\{2\},\{2,4\},\{1,2,4\}\}.$$

We can associate to each set its distance from the minimal set in the corresponding chain and then write each chain as one set. If the distance from the minimal element in the chain is denoted as a subscript of an element then the whole chains in the above example can be described as follows:

$$\mathcal{C}_1 = \{1_0, 2_3, 3_2\} = 1_0 2_3 3_2$$

and

$$\mathcal{C}_2 = \{1_2, 2_0, 4_1\} = 1_2 2_0 4_1.$$

This approach to represent pattern chains can be applied to a wide variety of different pattern collections such as substrings and graphs. Besides making the pattern collection more compact and hopefully understandable this approach can also compress the pattern collection: the total size of the sets of a length k ($k \leq |R|$) chain can be $\Theta(k|R| + |w|)$ in the worst case but the size of a chain as one pattern is only $O(|R| \log k + |w|)$, where $|w|$ is the size of the weight function. This is the case when the data set is the collection of all suffixes of R (for any given ordering of elements in R), i.e., the upper triangular binary matrix of size $|R| \times |R|$. The whole set collection can be described as a one chain but the set collection still has $|R|$ distinct sets.

3 Algorithmic Issues

A minimum chain partition $\mathcal{C}_1, \ldots, \mathcal{C}_m$ of a partially ordered set $\mathcal{P}$ can be found efficiently by finding a maximum matching in the bipartite graph corresponding to the partial order [30].

A *bipartite graph* is a triplet $G = (V_l, V_r, E)$, where V_l and V_r are two distinct sets called *vertices*, and E is a subset of $V_l \times V_r$ called *edges*. An edge $e \in E$ is *adjacent* to a vertex $v \in V_l \cup V_r$ iff $e = (p, q)$ or $e = (q, p)$ for some $q \in V_l \cup V_r$. A *matching* M in G is a set of pairwise disjoint edges. A matching M is *maximal* iff there is no edge in E that is disjoint from all edges in M. A matching M is *maximum* iff no matching M' in G is larger than M, i.e., $|M'| \leq |M|$ for all matchings M' in G.

The bipartite graph corresponding to the partial order $\prec$ of $\mathcal{P}$ is $G = (\mathcal{P}, \mathcal{P}, \prec)$. That is, G consists of the partial order $\prec$ and two copies of the pattern collection $\mathcal{P}$. Any matching M in G determines a partition of $\mathcal{P}$ into chains: the matching M partites the pattern collection $\mathcal{P}$ to directed paths in the partial order $\prec$ and each path determines one chain. The number of chains in the matching M is equal to the number of patterns p in $\mathcal{P}$ such that for no $q \in \mathcal{P}$ holds: $(p, q) \in M$. Thus a maximum (maximal) matching corresponds to a minimum (minimal) chain partition.

A maximum matching M in the bipartite graph $G = (V_l, V_r, E)$ can be found in time $O\left(\sqrt{\min\{|V_l|, |V_r|\}}\, |E|\right)$ [30,31]. Thus if the partial order $\prec$ is known explicitly then the minimum chain partition can be found in time $O\left(\sqrt{|\mathcal{P}|}|\prec|\right)$ which is bounded above by $O\left(|\mathcal{P}|^{5/2}\right)$ as there are at most $|\mathcal{P}|^2$ pairs in $\prec \subseteq \mathcal{P} \times \mathcal{P}$.

It is possible to partite $\mathcal{P}$ also to the minimum number of trees or degree-constrained subgraphs in polynomial time in $|\mathcal{P}|$ by finding the maximum b-matching in the corresponding bipartite graph. The *bipartite b-matching* is a generalization of the bipartite matching such that for each vertex $v \in V_l \cup V_r$ there is an upper bound (a lower bound) that determine how many edges adjacent to v can (must) be chosen to M. In the case of the ordinary matching, the upper bound is one and the lower bound is zero.

Another way to generalize the maximum matching is to search for such a maximum matching in a edge-weighted bipartite graph that the sum of the edge weights is smallest in the collection of all maximum matchings for that graph. That minimum weight maximum matching corresponds to a minimum chain partition with the smallest sum of weights for consecutive elements in each chain.

However, finding a good partition of a pattern collection into chains has the following two traits:

1. The pattern collections can be enormously large.
2. The partial order might not be known explicitly.

The first problem can be overcome by searching for a maximal matching instead of a maximum matching. A maximal matching in $G = (V_l, V_r, E)$ can be found in time $O(|E|)$ by trying to add the edges one by one in arbitrary order to the matching. It is easy to show that a maximal matching is at least half of the maximum matching. Unfortunately this does not imply any nontrivial approximation quality guarantees for the chain partition. To see this, consider the set $\{1, 2, \ldots, 2n\}$ and let the partial order $\prec$ be $\{(i, j) : i < j\}$. The maximum matching

$$\{(1, 2), (3, 4), \ldots, (2n - 1, 2n)\}$$

determines just one chain

$$\mathcal{C} = \{1, 2, \ldots, 2n\}$$

whereas the worst maximal matching

$$\{(1, 2n), (2, 2n - 1), \ldots, (n, n + 1)\}$$

determines n chains

$$\mathcal{C}_1 = \{1, 2n\}, \mathcal{C}_2 = \{2, 2n - 1\}, \ldots, \mathcal{C}_n = \{n, n + 1\}.$$

Thus in the worst case the solution found by the greedy algorithm is $|\mathcal{P}|/2$ times worse than the optimal solution.

The quality of a maximal matching, i.e., a minimal chain partition can be improved by finding a total order that conforms to the partial order. If the partial order is known explicitly then a total order conforming it can always be found using topological sorting. Sometimes it is easy to compute a total order even without knowing the partial order explicitly. This is the case, for example, with the frequent sets: the sets can be sorted w.r.t. the cardinalities of the sets. This kind of auxiliary information can significantly reduce the size of the chain partition found using a maximal matching algorithm. The amount of improvement depends on how well the total order is able to capture the essence of the partial order. For example, in the case of the set $\{1, 2, \ldots, 2n\}$ matching the elements greedily in the ascending (or descending) order produces the maximum matching.

If the partial order is given implicitly then its explicit computation might itself be the major computational bottleneck of chaining. The time complexity of the brute force solution, i.e., testing all pairs of patterns in $\mathcal{P}$, is $O\left(|\mathcal{P}|^2\right)$. In the worst case this time bound is asymptotically optimal as the whole set $\mathcal{P}$ can be an antichain: then each pair of patterns in $\mathcal{P}$ must be compared to verify that $\mathcal{P}$ is an antichain.

Partial orders have two useful properties that can be exploited when computing (the explicit representation of) the partial order: transitivity and irreflexivity. Because of transitivity we know that if $p \prec q$ and $q \prec r$ then $p \prec r$. Irreflexivity (together with transitivity) guarantees that the graph $G = (\mathcal{P}, \prec)$ is acyclic.

The partial order can be obtained as a side product of the minimal chain partition as follows:

Init. Init the number m of chains to zero.
Growth. For each $p \in \mathcal{P}$, add p to some of the existing chains $\mathcal{C}_i, 1 \leq i \leq m$, if possible, or create a new chain $\mathcal{C}_{m+1}$ for p and increase m by one.
Comparison. For all chains $\mathcal{C}_i$ and $\mathcal{C}_j$, compute the partial order of $\mathcal{C}_i \cup \mathcal{C}_j$.

To make the complexity analysis of the above procedure simpler we assume that any two patterns in the pattern collection can be compared in a constant time. Then figuring out whether a pattern p can be added to chain $\mathcal{C}_i, 1 \leq i \leq m$, can be computed in time $O(|\mathcal{C}_i|)$. The comparison step can be implemented in several ways. Some of the partial order is revealed already when each pattern $p \in \mathcal{P}$ has been tried to add to existing chains. The partial order for the union $\mathcal{C}_i \cup \mathcal{C}_j$ two chains $\mathcal{C}_i$ and $\mathcal{C}_j$ can be computed from the partial orders of $\mathcal{C}_i$ and $\mathcal{C}_j$ in time $O(|\mathcal{C}_i| + |\mathcal{C}_j|)$. However, as the brute force solution is worst case optimal, the efficiency of different comparison heuristics has to be evaluated experimentally.

4 Experiments

We tested the condensation abilities of the pattern chaining using two data sets: Internet Usage data consisting of 10104 rows and 10674 attributes and IPUMS Census data consisting of 88443 rows and 39954 attributes. The data sets were downloaded from the UCI KDD Repository[1].

From the data sets we computed the closed frequent sets, minimal and the minimum chain partitions of the closed frequent sets, and the maximal frequent sets, with different minimum frequency thresholds. The closed frequent sets were sorted by their cardinalities before finding the minimal chain partitions.

Results are shown in Figures 1 and 2. The number of chains is smaller than the number of closed sets. Thus the idea of finding minimum chain partition seems to be useful in that sense.

Even more interesting results were obtained when comparing the minimal and the minimum chain partitions: the greedy heuristic produced almost as good solutions as the much more computational demanding bipartite matching. We got similar results with other data sets we experimented. However it is not clear whether the quality of maximal matchings is specific to frequent sets or if the results hold for other pattern collections.

It is worth to remember that the fundamental assumption in the frequent set mining is that not very large sets are frequent since also all the subsets of

[1] `http://kdd.ics.uci.edu`

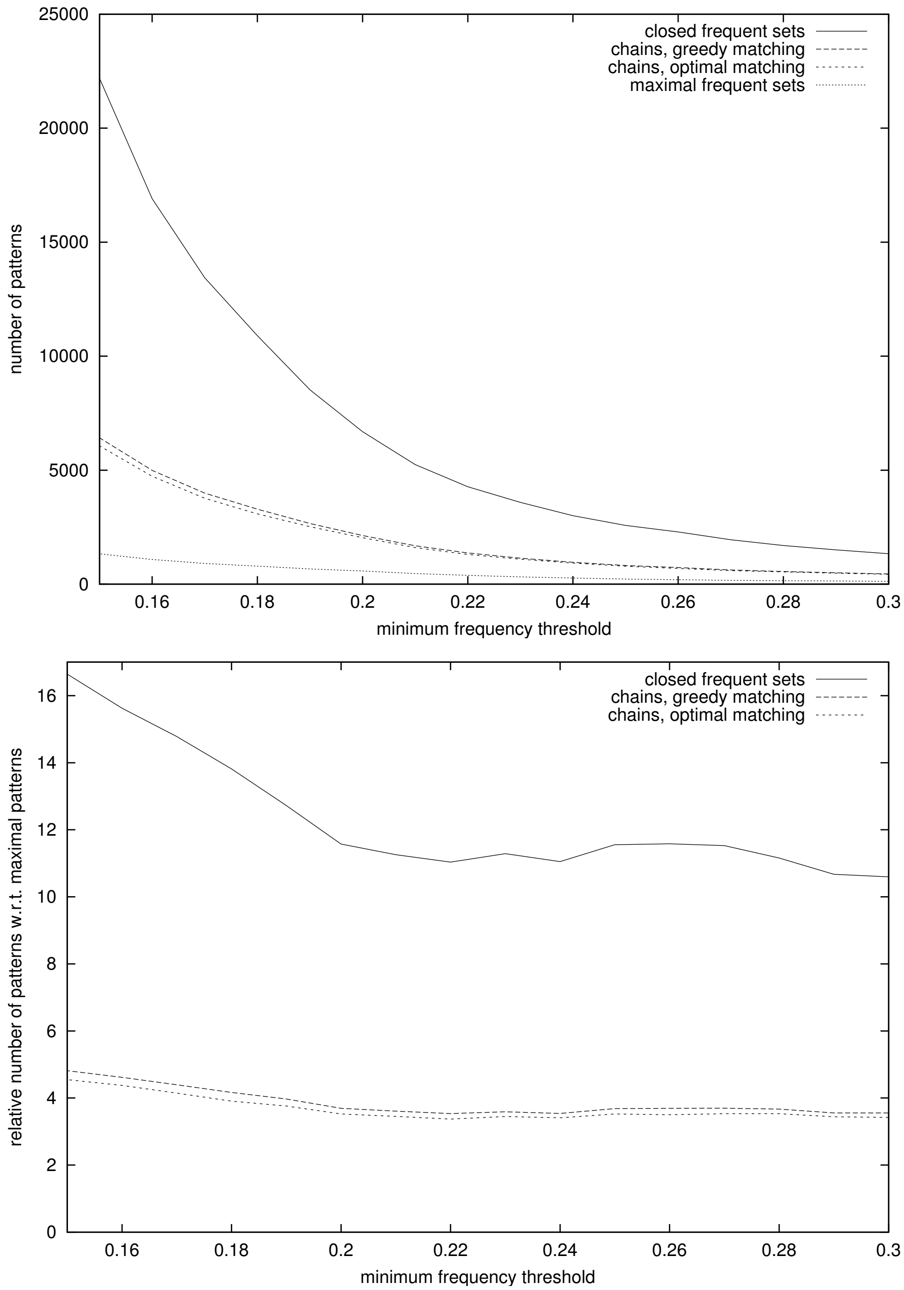

Fig. 1. IPUMS Census data

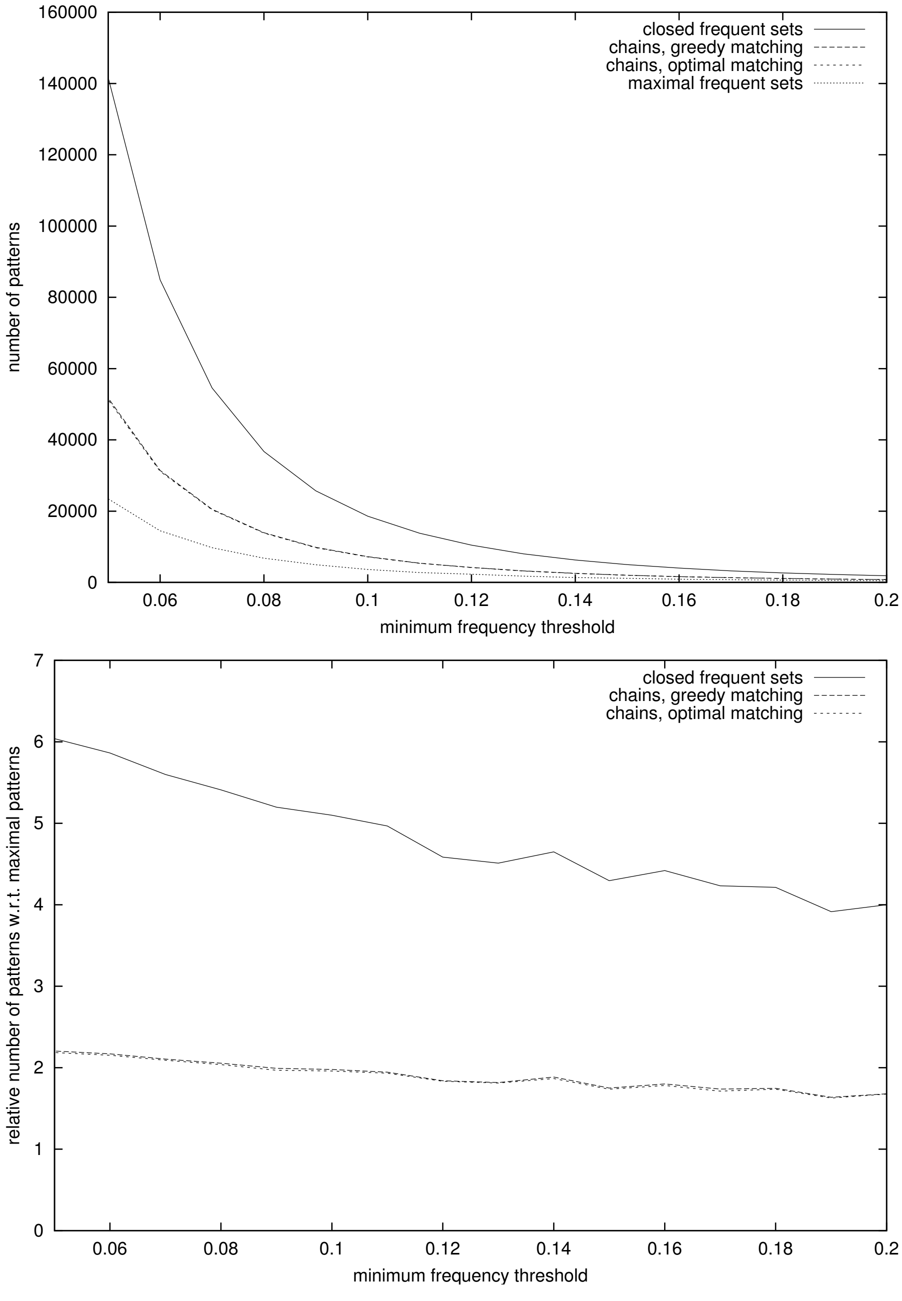

Fig. 2. Internet Usage data

a frequent set are frequent. This implies that the chains cannot be very long as the length of the longest chain is equal to the size of the largest frequent set. This observation makes the results even more satisfactory.

5 Conclusions

In this paper we have introduced the chain partitions of partially ordered pattern collections as a high-level approach to condense and structure pattern collections, even already condensed ones, and also a structural clustering of the pattern collection. We described how the minimum chain partitions can be found and how the computation of the minimum chain partition and obtaining the explicit chain partitions can be made more efficient. Also, we showed that the chain partitions of pattern collections are useful in practice.

However, there are still many important open problems related to pattern chains:

- What kind of additional constraints for the chain partitions are computationally tractable and useful, especially in data analysis?
- Are there efficient approximation algorithms (with approximation quality guarantees) for finding the chain partitions to cope with massive pattern collections?
- Can the minimum chain partitions make some existing pattern discovery algorithms run faster?
- What kind of consensus patterns of pattern collections, besides of chain partitions, are valuable?
- In general, how should the structure of pattern collections be exploited?

Acknowledgments. I wish to thank Matti Kääriäinen, Heikki Mannila and Janne Ravantti for helpful comments on the manuscript and refreshing discussions.

References

1. Hand, D.J.: Pattern detection and discovery. In Hand, D., Adams, N., Bolton, R., eds.: Pattern Detection and Discovery. Volume 2447 of LNAI., Springer-Verlag (2002) 1–12
2. Hand, D.J., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press (2001)
3. Mannila, H.: Local and global methods in data mining: Basic techniques and open problems. In Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R., eds.: Automata, Languages and Programming. Volume 2380 of LNCS., Springer-Verlag (2002) 57–68
4. Gunopulos, D., Khardon, R., Mannila, H., Saluja, S., Toivonen, H., Sharma, R.S.: Discovering all most specific sentences. ACM Transactions on Database Systems **28** (2003) 140–174
5. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining – a general survey and comparison. SIGKDD Explorations **1** (2000) 58–64
6. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery **1** (1997) 241–258
7. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds.: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996) 307–328
8. Boulicaut, J.F., Bykowski, A.: Frequent closures as a concise representation for binary data mining. In Terano, T., Liu, H., Chen, A.L.P., eds.: Knowledge Discovery and Data Mining. Volume 1805 of LNAI., Springer-Verlag (2000) 62–73
9. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Approximation of frequency queries by means of free-sets. In Zighed, D.A., Komorowski, J., Żytkow, J., eds.: Principles of Knowledge Discovery and Data Mining. Volume 1910 of LNAI., Springer-Verlag (2000) 75–85
10. Boros, E., Gurvich, V., Khachiyan, L., Makino, K.: On the complexity of generating maximal frequent and minimal infrequent sets. In Alt, H., Ferreira, A., eds.: STACS 2002. Volume 2285 of LNCS., Springer-Verlag (2002) 133–141
11. Bykowski, A., Rigotti, C.: A condensed representation to find frequent patterns. In: Proceedings of the Twenteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, ACM (2001)
12. Calders, T., Goethals, B.: Mining all non-derivable frequent itemsets. In Elomaa, T., Mannila, H., Toivonen, H., eds.: Principles of Data Mining and Knowledge Discovery. Volume 2431 of LNAI., Springer-Verlag (2002) 74–865
13. Calders, T., Goethals, B.: Minimal k-free representations of frequent sets. In Lavrac, N., Gamberger, D., Todorovski, L., Blockeel, H., eds.: Principles of Knowledge Discovery and Data Mining. LNAI, Springer-Verlag (2003)
14. Geerts, F., Goethals, B., Mielikäinen, T.: What you store is what you get (extended abstract). In: 2nd International Workshop on Knowledge Discovery in Inductive Databases. (2003)
15. Gouda, K., Zaki, M.J.: Efficiently mining maximal frequent itemsets. In Cercone, N., Lin, T.Y., Wu, X., eds.: Proceedings of the 2001 IEEE International Conference on Data Mining. IEEE Computer Society (2001) 163–170
16. Kryszkiewicz, M.: Concise representation of frequent patterns based on disjunction-free generators. In Cercone, N., Lin, T.Y., Wu, X., eds.: Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE Computer Society (2001) 305–312

17. Mannila, H., Toivonen, H.: Multiple uses of frequent sets and condensed representations. In Simoudis, E., Han, J., Fayyad, U.M., eds.: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press (1996) 189–194
18. Mielikäinen, T.: Frequency-based views to pattern collections. In: IFIP/SIAM Workshop on Discrete Mathematics and Data Mining. (2003)
19. Mielikäinen, T., Mannila, H.: The pattern ordering problem. In Lavrac, N., Gamberger, D., Todorovski, L., Blockeel, H., eds.: Principles of Knowledge Discovery and Data Mining. LNAI, Springer-Verlag (2003)
20. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In Beeri, C., Buneman, P., eds.: Database Theory - ICDT'99. Volume 1540 of LNCS., Springer-Verlag (1999) 398–416
21. Pavlov, D., Mannila, H., Smyth, P.: Beyond independence: probabilistic methods for query approximation on binary transaction data. IEEE Transactions on Data and Knowledge Engineering (2003) To appear.
22. Pei, J., Dong, G., Zou, W., Han, J.: On computing condensed pattern bases. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan, IEEE Computer Society (2002) 378–385
23. Mielikäinen, T.: Finding all occurring sets of interest. In: 2nd International Workshop on Knowledge Discovery in Inductive Databases. (2003)
24. Pei, J., Han, J., Mao, T.: CLOSET: An efficient algorithm for mining frequent closed itemsets. In Gunopulos, D., Rastogi, R., eds.: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. (2000) 21–30
25. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with Titanic. Data & Knowledge Engineering **42** (2002) 189–222
26. Zaki, M.J., Hsiao, C.J.: CHARM: An efficient algorithms for closed itemset mining. In Grossman, R., Han, J., Kumar, V., Mannila, H., Motwani, R., eds.: Proceedings of the Second SIAM International Conference on Data Mining, SIAM (2002)
27. Agarwal, R.C., Aggarwal, C.C., Prasad, V.V.V.: A tree projection algorithm for generation of frequent item sets. Journal of Parallel and Distributed Computing **61** (2001) 350–371
28. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In Chen, W., Naughton, J.F., Bernstein, P.A., eds.: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM (2000) 1–12
29. Jukna, S.: Extremal Combinatorics: With Applications in Computer Science. EATCS Texts in Theoretical Computer Science. SPringer-Verlag (2001)
30. Lovász, L., Plummer, M.: Matching Theory. Volume 121 of Annals of Discrete Mathematics. North-Holland (1986)
31. Galil, Z.: Efficient algorithms for finding maximum matchings in graphs. ACM Computing Surveys **18** (1986) 23–38