Mining Multi-domain Sequential Patterns

Zhung-Xun Liao, Wen-Chih Peng and Xing-Yuan Hu
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, ROC
E-mail:{g9113, wcpeng, hyhu}@cs.nctu.edu.tw

ABSTRACT

In reality, sequential patterns may exist in multiple sequence databases. In this paper, we explore a novel sequential pattern mining problem: mining multi-domain sequential patterns cross multiple domain sequence databases. We propose two algorithms, IndividualMine and PropagatedMine, for efficiently mining multi-domain sequential patterns. In algorithm IndividualMine, sequential patterns in each domain should be discovered and then by iteratively combinating sequential patterns, multi-domain sequential patterns are generated. Algorithm PropagatedMine performs sequential pattern mining in one starting domain and propagates sequential patterns mined to other domain to reduce sequence database sizes of other domains. A comprehensive performance study is conducted and experimental results show the scalability and efficiency of our proposed algorithms.

1: INTRODUCTIONS

Sequential pattern mining has attracted a significant amount of research efforts recently. The problem of sequential pattern mining is that discovering frequent sequences with their occurrence counts larger than or equal to the user-specified number, *min_support*, among a set of sequences [1]. Sequential pattern mining can be applied on business and marketing analysis, Web page browsing behavior, symptomatic pattern of a disease, DNA sequence and hacker invasion detecting. Due to the importance of sequential patterns mining, many efficient sequential pattern mining algorithms have been proposed recently [1][3][7][9].

However, existing sequential pattern mining algorithms only discover sequential behavior (e.g., buying behavior) in one domain, which are not sufficient for behavior analysis. One would like to discover sequential patterns across multiple domains. Such a sequential pattern across multiple domains is referred to a multi-domain sequential pattern in this paper. A multi-domain sequential pattern consists of sequences across multiple domains and for each item of a sequence, the corresponding items having the same order in different domain sequences occur in the same time slot. Note that a multi-domain sequential pattern captures cross relationship among multiple domains. Applications of multi-domain sequential patterns include but are not limited to the following two.

- **User behavior analysis in a mobile computing environment.** Consider a mobile computing environment in Figure 1, where mobile users can access three services (i.e., location tracking service, data searching service, and credit payment service) via mobile devices and each service is referred to one domain in this paper. Given a log of movements of a user from the location tracking service, one would mine user moving patterns referred to those areas in which the user frequently travels. As such, in Figure 1, for each domain, sequential patterns in each domain are discovered by existing algorithms. Note that in order to reflect behavior of a user in such environment; one would like to find more complex sequential patterns cross multiple domains. An illustrative sequential patterns is shown in Figure 1, where a user stays at area *{A}*, searches data items *{1, 2}*, and buys goods *{α, β}*; then moves to area *{B, C}*, searches data *{3, 4, 5}*, and buys goods *{γ}*; and finally moves to area *{D}*, searches data *{6, 7}*, and buys goods *{θ, δ}*. Such a sequential pattern consists of sequences cross multiple domains and provides more information to analyze user behaviors. For example, the user is motivated by the scene at location A and then buys goods *{α, β}* after surfing some web pages of *{1, 2}*.
- **Behavior or event analysis in a sensor network.** Imagine that a large amount of sensors are deployed in a smart home for behavior analysis. Sensors with different sensing capabilities (i.e., water, motion and vibration) are viewed as different domains. Some research projects are conducted to identify the behaviors of users for smart environment. As such, mining a multi-domain sequential pattern could be used to analyze behaviors of users. For example, to recognize one user behavior (i.e., cleaning behavior), one could mine multi-domain sequential patterns, which comprise of sequential patterns in water, motion and vibration domains, from those data logs generated by sensors with various sensing capabilities.

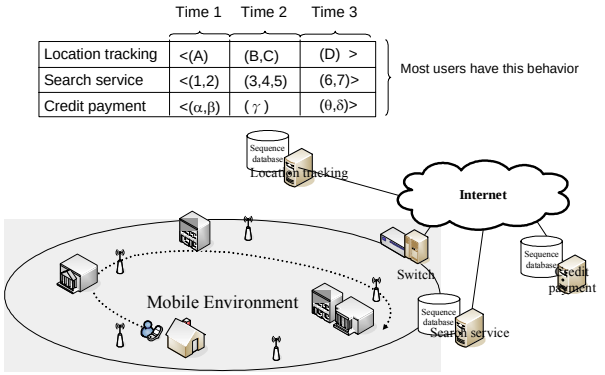

Figure 1. An example of multi-domain sequential patterns.

Though many sequential pattern mining algorithms are able to efficiently mine patterns in one domain, these algorithms cannot directly be applied in mining multi-domain sequential patterns. Existing sequential algorithms suffer from poor performance when being applied in mining multi-domain sequential patterns across multiple domain sequence databases. In order to efficiently mine multi-domain sequential patterns, we propose two algorithms IndividualMine and PropagatedMine. In algorithm IndividualMine, we will perform sequential pattern mining in each domain and then for those sequential patterns in multiple domains have the same time slots will be integrated into multi-domain sequential patterns. Note that mining sequential patterns in each domain is not an efficient way since not all sequential patterns are able to form multi-domain sequential patterns. To reduce the cost of mining sequential patterns in each domain, we propose algorithm PropagatedMine in which we start mining sequential patterns in one domain and then propagate the occurrence times of sequential patterns in this domain to other domains to further reduce the size of databases. Performance study is developed to validate our proposed algorithms. Simulation results show that by exploring propagating, algorithm PropagatedMine outperforms algorithm IndividualMine.

A significant amount of research efforts have been elaborated upon issues of mining sequential patterns[5][14][13][12][11][2]. We mentioned in passing that the authors in [1] formulated the problem of sequential pattern mining and proposed mining algorithms based on Apriori algorithm. By exploring a breadth first search and button-up algorithm, the authors in [10] developed algorithm GSP[10] for mining sequential patterns, whereas the authors in [15] devised algorithm SPADE, which is a depth first search and button-up algorithm with ID-list. The authors in [6][8] exploited the concept of projection in algorithms PrefixSpan and FreeSpan to reduce the volume of data for mining sequential pattern mining. To prevent the candidate generation, the authors in DISC-all[4] used a novel sequence comparison strategy. In addition, the authors in [9] developed several algorithms to mine multi-dimension sequential patterns in which sequential patterns with some category attributes are discovered. To the best of our knowledge, prior works do not fully explore the mining capability for multi-domain sequential patterns, let alone proposing efficient algorithms to mine such sequential patterns. These features distinguish this paper from others. The contributions of this paper are twofold: (1) exploiting a novel and useful sequential patterns (i.e., multi-domain sequential patterns) and (2) devising algorithm PropagatedMine to efficiently mine multi-domain sequential patterns.

The remaining of the paper is organized as follows. In Section 2, some preliminaries are presented. Algorithms for mining multi-domain sequential patterns are developed in Section 3. Performance studies are conducted in Section 4. This paper concludes with Section 5.

2: Preliminary

Let *MDS* be a multi-domain sequence, which is represented as $\begin{bmatrix} M_{11} & M_{12} & .. & M_{1b} \\ M_{21} & M_{22} & .. & M_{2b} \\ : & : & : & : \\ M_{a1} & M_{a2} & .. & M_{ab} \end{bmatrix}$, where each row $[M_{i1}, M_{i2}, ..., M_{ib}]$ for $i = 1, 2, ..., a$, is a sequence and each column $[M_{1j}, M_{2j}, ..., M_{aj}]^T$ for $j = 1, 2, ..., b$, is a vector of itemsets occurring in time period j. In other words, *MDS* consists of the number of a domains and continuous for the number of b time periods.

Definition 2.1 (Contain Relation): Let $M=\begin{bmatrix} m_{11} & m_{12} & .. & m_{1b} \\ m_{21} & m_{22} & .. & m_{2b} \\ : & : & : & : \\ m_{a1} & m_{a2} & .. & m_{ab} \end{bmatrix}$ and $N=\begin{bmatrix} n_{11} & n_{12} & .. & n_{1b'} \\ n_{21} & n_{22} & .. & n_{2b'} \\ : & : & : & : \\ n_{a1} & n_{a2} & .. & n_{ab'} \end{bmatrix}$ be two multi-domain sequences, where $b < b'$. We define that M is contained by N, denoted as $M \subseteq N$, if and only if there exists an integer list $1 \le l_1 < l_2 < ... < l_b \le b'$, such that $m_{ij} \subseteq n_{ilj}$, where $i = 1, 2, ...a$ and $j = 1, 2, ..., b$. For example, let M and N be two multi-domain sequences shown in Table 1. It can be verified that M contains N or N is contained by M, because an integer list $1 < 3$, such that $(a) \subseteq (a)$, $(2) \subseteq (1, 2)$, $(b, c) \subseteq (b, c, d)$, and $(6) \subseteq (6)$. Therefore, a multi-domain sequence database is a set of multi-domain sequences, defined as $MDB = \{M_1, M_2, \ldots, M_n\}$, where $M_i=\begin{bmatrix} m^i_{11} & m^i_{12} & .. & m^i_{1b} \\ m^i_{21} & m^i_{22} & .. & m^i_{2b} \\ : & : & : & : \\ m^i_{a1} & m^i_{a2} & .. & m^i_{ab} \end{bmatrix}$, $i=, 2, ..., n$, is a multi-domain sequence with the number of a domains and b time periods. For example, Table 2 is a multi-domain sequence database which contains 4 multi-domain sequences with 2 domains and 3 to 4 time periods. In addition, a multi-domain sequence database can be

Period Number		P_1	P_2	
D_1	<	(a)	(b,c)	>
D_2	<	(2)	(6)	>

Multi-domain Sequence M

Period Number		P_1	P_2	P_3	P_4	
D_1	<	(a)	(b,c)	(b,c,d)	(e)	>
D_2	<	(1,2)	(2,3)	(6)	(4,5)	>

Multi-domain Sequence N

Table 1. Two multi-domain sequences *M* and *N*, where $M \subseteq N$.

Sequence Id	Multi-domain Sequnces
S_1	(a) (b,c) (b,c,d) (e) (1,2) (2,3) (6) (4,5)
S_2	(a,b) (b,c) (c,e) (1,3) (2,4) (8)
S_3	(a,e) (h) (g,j) (1,6) (5) (9,10)
S_4	(a,b,f) (d) (b,c) (e,f) (1,2,5) (7) (2,3) (4,5,6)

Table 2. An example of multi-domain sequence database.

represented as a set of domains, $MDB = \{D_1, D_2, ..., D_a\}$, where $D_i = \{s_1, s_2, ..., s_n\}$, $s_j = < m_{i1}^j, m_{i2}^j, \ldots, m_{ib}^j >$, $i=1, 2, ..., a$, and $j=1, 2, ..., n$.

Definition 2.2 (Support and Time Instance Set): Let M be a multi-domain sequence and MDB be a multi-domain sequence database. The support of M is the number of multi-domain sequences in MDB that contain M, denoted as $Support(M) = |\{N | N \in MDB \text{ and } M \subseteq N\}|$. The time instance set is the set of all time instances of M, denote as $TIS(M) = \{< N >: \text{the corresponding ordered integer list of } N | N \in MDB \text{ and } M \subseteq N\}$.

Definition 2.3 (Size of Time Instance Set): Let M be a multi-domain sequence and MDB be a multi-domain sequence database. The size of time instance set of M is denoted as $|TIS(M)| = |\{N | N \in MDB \text{ and } M \subseteq N\}|$ and the value of $|TIS(M)|$ is set to $Support(M)$.

For example, assume that $M = \begin{bmatrix} (a) & (b,c) \\ 1 & 2 \end{bmatrix}$ is a multi-domain sequence. The support of M in multi-domain sequence database MDB shown in Table 2 is 3 since three multi-domain sequences, S_1, S_2, S_4, in MDB contain M. Furthermore, $TIS(M) = \{< S_1 : 1, 2 >, < S_2 : 1, 2 >, < S_4 : 1, 3 >\}$ and $|TIS(M)| = |\{S_1, S_2, S_4\}| = 3$.

Definition 2.4 (Frequent): Given a minimum support threshold δ, a multi-domain sequence database MDB, and a multi-domain sequence M. M is a frequent multi-domain sequence in MDB, if and only if $Support(M) > \delta$. In other words, M is a multi-domain sequential pattern of MDB.

For example, given a multi-domain sequence database MDB depicted in Table 2, and the minimum support $\delta = 3$. The multi-domain sequential patterns are $\begin{bmatrix} (a) \\ (1) \end{bmatrix}, \begin{bmatrix} (b) \\ (2) \end{bmatrix}, \begin{bmatrix} (b) \\ (3) \end{bmatrix}, \begin{bmatrix} (c) \\ (2) \end{bmatrix}, \begin{bmatrix} (b,c) \\ (2) \end{bmatrix}, \begin{bmatrix} (a) & (b) \\ (1) & (2) \end{bmatrix}, \begin{bmatrix} (a) & (c) \\ (1) & (2) \end{bmatrix}$, and $\begin{bmatrix} (a) & (b,c) \\ (1) & (2) \end{bmatrix}$.

Problem of mining multi-domain sequential patterns: To facilitate the presentation of multi-domain sequential patterns, Table 2 is used to illustrate an example of multi-domain sequence databases and then we should determine multi-domain sequential patterns from a multi-domain sequence database given. In reality, however, each domain has its own database to store log data in which each date record contains both the occurrence time and data items. To derive a multi-domain sequence database, one should join these data log from each domain database with the join key as the occurrence time. Consequently, deriving a multi-domain sequence database is hard to achieve due to a huge amount of costs incurred when performing joining operations across multi-domain databases. Consider Table 2 as an example, where the set of multi-domain sequence databases is shown in Table 3. As such, the problem of mining multi-domain sequential patterns is that give a set of sequence databases in multiple domains, we should mine multi-domain sequential patterns cross these multiple sequence databases.

3: Multi-domain Sequential Pattern Mining

3.1: Algorithm IndividualMine

Given a set of sequence databases, algorithm IndividualMine consists of two phases: the mining phase and checking phase. Specifically, in the mining phase, one could utilize one of existing sequential pattern mining algorithms to mine sequential patterns and derive the corresponding time instance sets of each sequential patterns in each domain. In the checking phase, for each sequence patterns in each domain, we will enumerate all possible combinations to form candidate multi-domain sequence patterns and then determine support values for each candidate multi-domain sequence patterns. If a candidate multi-domain sequence patterns has its support

Domains	Sequences
D_1	<(a),(b,c),(b,c,d),(e)> <(a,b),(b,c),(c,e)> <(a,e),(h),(g,j)> <(a,b,f),(d),(b,c),(e,f)>
D_2	<(1,2),(2,3),(6),(4,5)> <(1,3),(2,4),(8)> <(1,6),(5),(9,10)> <(1,2,5),(7),(2,3),(4,5,6)>

Table 3. An example of sequence databases in multiple domains.

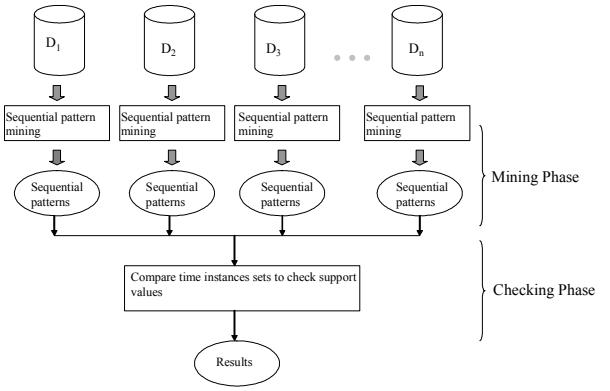

Figure 2. The overview of algorithm IndividualMine.

larger than the threshold value defined, this candidate multi-domain sequence pattern will become a frequent multi-domain sequence pattern.

As described before, in the mining phase of algorithm IndividualMine, one could exploit existing sequential mining algorithms to mine sequential patterns. Then, one could combine these sequential patterns to generate candidate multi-domain sequential patterns. Let $MDB=\{D_1, D_2, ..., D_a\}$ be a multi-domain sequence database. In the mining phase, we can obtain the set of sequential patterns SP_i of domain D_i, where $i=1, 2, ..., a$. Suppose $\forall f_i \in SP_i$, where $i=1, 2, ..., a$ and the lengths of these sequential patterns are the same. In the checking phase, we will iteratively determine whether two sequential patterns from different domains are able to become a multi-domain sequential pattern. For example, we first verify whether is frequent of $\{D_1, D_2\}$ or not. At the k-th iteration, the frequent of $\{D_1, D_2, ..., D_k\}$ and f_{k+1} are combined and we should determine whether is frequent of $\{D_1, D_2, ..., D_{k+1}\}$ or not. In order to efficiently count support values of candidate multi-domain sequential patterns, the time instance sets of sequential patterns are compared. For example, assume that the time instance sets of <(a)(b)> and <(1)(2)> are TIS(<(a)(b)>)={<S_1:1, 2>, <S_1:1, 3>, <S_2:1, 2>, <S_4:1, 3>} and TIS(<(1)(2)>)={<S_1:1, 2>, <S_2:1, 2>, <S_4:1, 3>} respectively. It can be verified that $TIS(\begin{bmatrix}(a) & (b)\\ (1) & (2)\end{bmatrix})$={<$S_1$:1, 2>, <$S_2$:1, 2>, <$S_4$:1, 3>} and $Support(\begin{bmatrix}(a) & (b)\\ (1) & (2)\end{bmatrix})$=3. Therefore, following the above operations, we could derive multi-domain sequential patterns.

3.2: Algorithm PropagatedMine

Algorithm PropagatedMine is designed to reduce the cost of mining sequential patterns in each domain. Moreover, sequential patterns mined in each domain are not necessary to form multi-domain sequential patterns. Hence, algorithm PropagatedMine only performs sequential pattern mining in one domain (referred to as a starting domain) and then propagates sequential patterns mined to other domains to reduce the size of databases

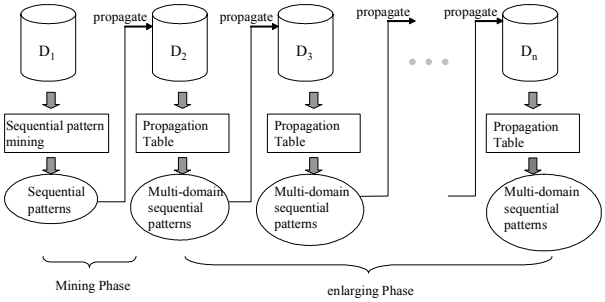

Figure 3. The flowchart of algorithm PropagatedMine.

in other domains. Algorithm PropagatedMine will iteratively propagate sequential patterns to next domains until all domains have been mined. Algorithm PropagatedMine is composed of two phases, the mining phase and the enlarging phase. The flowchart of algorithm PropagatedMine is depicted in Figure 3.

Same as in Algorithm IndividualMine, in the mining phase, algorithm PropagatedMine utilizes traditional sequential pattern mining method to discover sequential patterns in the starting domain (e.g., D_1) and then propagates sequential patterns mined to other domains. By propagating sequential patterns to other domains, we could reduce database size of other domains in that only those items whose occurring times are exactly the same as those sequential patterns propagated are extracted. Thus, for each sequential pattern propagated, we have the corresponding propagated table defined as follows:

Definition 3.1 (Propagated table): Let $\alpha=[a_1\ a_2\ ..\ a_k]^T$ be a frequent multi-domain sequence database $MDB_s = \{S_1, S_2, ..., S_n\}$ consists of k domains where $k > 1$, and $TIS(\alpha) = \{< S_{p1} : l_1 >, < S_{p2} : l_2 >, ..., < S_{pf} : l_f >\}$ where $1 \le p_1 \le p_2 \le ... \le p_f \le n$. Suppose another domain $D_t = \{t_1, t_2, ..., t_m\}$ and $t_i = < X_1^i, X_2^i, ..., X_{e(ti)}^i >$. The propagated table of α in domain D_t is defined as $D_t||_\alpha = \{t'_{p1}, t'_{p2}, ..., t'_{pf}\}$ and $t'_{pi} = X_{li}^{pi}$.

Property of propagated table: Let $D_t||_\alpha$ be a propagated table in domain D_t, α be a frequent k-domain sequence pattern. It can be verified that $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ is a frequent $(k+1)$-domain sequential pattern with the minimum support δ if and only if β is a frequent item of $D_t||_\alpha$ with the same minimum support δ.

For example, Table 4 depicts the propagated table $D_2||_{<(b)>}$ of <(b)> in domain D_2 from the multi-domain sequence database shown in Table 2 and the given minimum support 3. Then, we can easily obtain the 2-atoms $\begin{bmatrix}(b)\\ (2)\end{bmatrix}$ and $\begin{bmatrix}(b)\\ (3)\end{bmatrix}$, where (2) and (3) are the

Sequence Id	(b)
S_1	(2,3)
S_1	(6)
S_2	(1,3)
S_2	(2,4)
S_4	(1,2,5)
S_4	(2,3)

Table 4. The example of propagated table $D_2||_{<(b)>}$.

frequent items of $D_2||_{<(b)>}$ with the minimum support as 3. Through this phase, we can enlarge multi-domain sequential patterns with k domains into $k + 1$ domains, and repeat the same process until there is no more domain that can be propagated.

4: Performance Study

Our simulation is running on a 1.8GHz Athlon PC with 1G main memory, and two algorithms are implemented in Java, where algorithm PrefixSpan is selected as the sequential pattern mining method used in the mining phase of the two algorithms. The datasets were generated by the data generator proposed in [1]. For example, a dataset M5D10kC10T5S4 means there are 5 domains, each domain consists of 10,000 sequences, the average elements in a sequence is 10, the number of items in an element is 5 and the average length of maximal sequential patterns is 4.

4.1: Experimental Results

The experimental results of scalability with minimum supports varied are shown in Figure 4. The dataset is M5D10kC8T8S8, and the minimum supports are 2.5%-10%. The execution time increase while the minimum support decrease especially for IndividualMine, because the cost of performing sequential pattern mining increase rapidly when minimum support decrease.

Figure 5 shows the scalability over the number of domains. The datasets are M2D10kC8T8S8, M3D10kC8T8S8, M4D10kC8T8S8 and M5D10kC8T8S8 while the minimum support is 2.5%. When the number of domains increases, the execution time of both PropagatedMine and IndividualMine tends to increase. By propagating sequential patterns to other domains, algorithm PropagatedMine outperforms algorithm IndividualMine.

5: Conclusions

In this paper, we explored a new sequential pattern, a multi-domain sequential pattern mining. It can be seen that multi-domain sequential patterns are interesting and useful in practice since these patterns clearly reflect the relations of domains hidden across multiple domains. We proposed algorithms IndividualMine and PropagatedMine to efficiently mine multi-domain

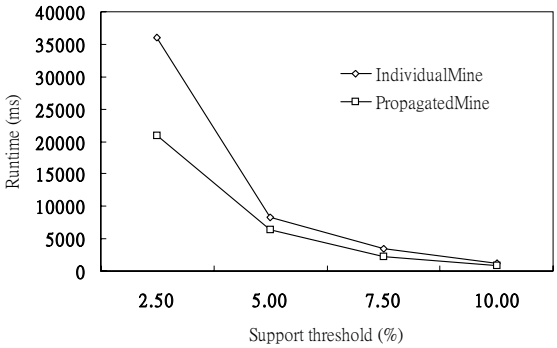

Figure 4. Execution time of two algorithms with minimum support varied.

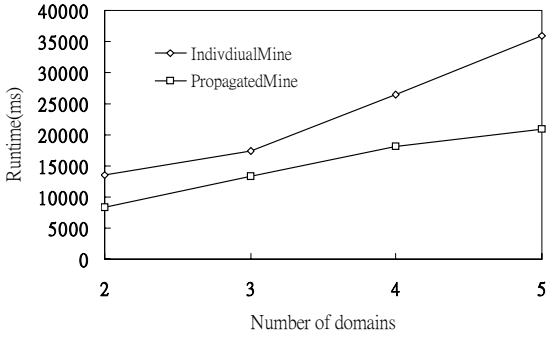

Figure 5. Execution time of two algorithms with the number of domains varied.

sequential patterns. Experimental results show that by propagating sequential patterns mined to other domains, algorithm PropagatedMine is able to more efficiently mine multi-domain sequential patterns. In the future, we will devise an optimal propagation order to further improve the performance of algorithm PropagatedMine.

REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.

[2] G. Chen, X. Wu, and X. Zhu. Sequential pattern mining in multiple streams. In *ICDM*, pages 585–588, 2005.

[3] H. Cheng, X. Yan, and J. Han. Incspan: Incremental mining of sequential patterns in large database. In *KDD*, pages 527–532, 2004.

[4] D.-Y. Chiu, Y.-H.Wu, and A. L. P. Chen. An efficient algorithm for mining frequent sequences by a new strategy without support counting. In *ICDE*, pages 375–386, 2004.

[5] M. N. Garofalakis, R. Rastogi, and K. Shim. Spirit: Sequential pattern mining with regular expression constraints. In *VLDB*, pages 223–234, 1999.

[6] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. Freespan: Frequent pattern-projected sequential pattern mining. In *KDD*, pages 355–359, 2000.

[7] N. Lesh, M. J. Zaki, and M. Ogihara. Mining features for sequence classification. In *KDD*, pages 342–346, 1999.
[8] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In *ICDE*, pages 215–224, 2001.
[9] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *CIKM*, pages 81–88, 2001.
[10] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, pages 3–17, 1996.
[11] P. Tzvetkov, X. Yan, and J. Han. Tsp: Mining topclosed sequential patterns. *Knowl. Inf. Syst*., 7(4):438– 457, 2005.
[12] J.Wang and J. Han. Bide: Efficient mining of frequent closed sequences. In *ICDE*, pages 79–90, 2004.
[13] X.Yan, J. Han, and R.Afshar. Clospan: Mining closed sequential patterns in large databases. In *SDM*, 2003.
[14] J. Yang, W. Wang, P. S. Yu, and J. Han. Mining long sequential patterns in a noisy environment. In *SIGMOD* Conference, pages 406–417, 2002.
[15] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.