Complex Patterns in Streams (COMPASS)

Open Competition Project NWO

Accepted July 2009

1 Abstract

In recent years there has been a growing interest in the study and analysis of flows of so-called data streams. Typical examples of such streams include Internet traffic data and continuous sensor readings. Traditional data mining approaches are not suitable for mining such streams, because they assume static data stored in a database, whereas streams are continuous, high speed, and unbounded. Therefore, streams must be analyzed as they are produced and high quality, online results need to be guaranteed.

Until now, most pattern mining techniques focus either on non-streaming data, or only consider very simple patterns, such as identifying the hot items from one stream, or constantly maintaining the frequencies in a window sliding over the stream. The challenging task we set forward in this project is to extend the existing state-of-the-art techniques into two, orthogonal directions: on the one hand, the mining of more complex patterns in streams, such as sequential patterns and evolving graph patterns and on the other hand, more natural stream support measures taking into account the temporal nature of most data streams.

The developed techniques will be tested on real-life data, such as social network data and the World-Wide Web. Next to those datasets, in the project we will have access to the data streams generated by a sensor network mounted on a large bridge in The Netherlands.

Keywords: Data mining, data streams, sensor networks

2 Composition of the Research Team

Name	University	Role
T.G.K. Calders, dr.	TU Eindhoven	Project leader
W. Kosters, dr.	U Leiden	Project co-promotor
M. Pechenizkiy, dr.	TU Eindhoven	Collaborator
P.M.E. De Bra, prof. dr.	TU Eindhoven	Collaborator (Thesis advisor AiO1)
J.N. Kok, prof. dr.	U Leiden	Collaborator (Thesis advisor AiO2)
A.J. Knobbe, dr.	U Leiden	Collaborator
H. Blockeel, dr.	U Leiden/KU Leuven	Collaborator
B. Obladen	Strukton	Collaborator (Expert *Hollandse Brug*)
AIO1 (N.N.)	TU Eindhoven	PhD student
AIO2 (N.N.)	U Leiden	PhD student

The research team consists of two strong research groups in the area of data mining. The competencies of the two groups are complementary, and both of crucial importance for the research project: the Leiden group is especially experienced in the graph mining domain as

witnessed by the recent PhD dissertations guided by prof. J. Kok in this area by Siegfried Nijssen [37] and Edgar de Graaf [10], whereas the group in Eindhoven brings in expertise on condensed representations for pattern mining and concept drift and some preliminary works on new support measures for for frequent patterns in streams. The official thesis advisors for the AIOs will be the professors De Bra and Kok.

3 Research School

The research schools involved are, for the TU/e, SIKS (School voor Informatie- en Kennissystemen) and, for ULeiden, IPA (Institute for Programming research and Algorithmics).

4 Full Proposal

Description of the Proposed Research

The topic of this research proposal lies at the intersection of data stream processing and data mining, and aims at the development of complex pattern mining techniques for the growing domain of stream analysis. We first briefly sketch the relevant related research areas and state the central problem motivating the project. After that we discuss our proposed approach and methodology.

Introduction to the relevant research areas

Stream processing [35] In some applications we are confronted with a continuous flow of data (a data stream) that need to be processed as they arrive ("online"), because either it is impossible or impractical to store all these data, or because the results of the data processing are required without delay. Typical examples are data streams generated by monitoring network traffic, sensor networks in scientific or medical applications, etc. Algorithms for stream processing must fulfill a number of conditions: they must be fast (to keep up with the incoming data), space-efficient (memory requirements of a stream processing technique are typically allowed to grow at most logarithmically in the length of the stream), and any-time (able to produce up-to-date results whenever requested).

Data mining [19, 43] Data Mining refers to the analysis of typically large amounts of data with the aim of discovering interesting (ir)regularities in these data. Typical data mining tasks include classification, clustering, and pattern mining. In this project, we will focus on frequent pattern mining, where the goal is to discover patterns that occur frequently in the data. Such a pattern may be the co-occurrence of items (a so-called itemset), a particular subsequence that frequently re-occurs in a longer sequence, a subgraph that frequently re-occurs in either one large graph (single-graph setting) or in a database of several graphs, etc.

Stream mining [14, 15] The classical pattern mining algorithms do not fulfill the constraints imposed on stream processing algorithms, as they require too much space and time and assume a static database. Mining data streams, or stream mining, is therefore a challenging task. It gained a lot of interest in the research literature. The most popular techniques that have been developed so-far are randomization and approximation, sampling, sketches, and summaries. Randomization and approximation techniques render stream mining algorithms sufficiently fast, at the expense of no longer guaranteeing exactness. Sampling implies that a small sample of the data stream

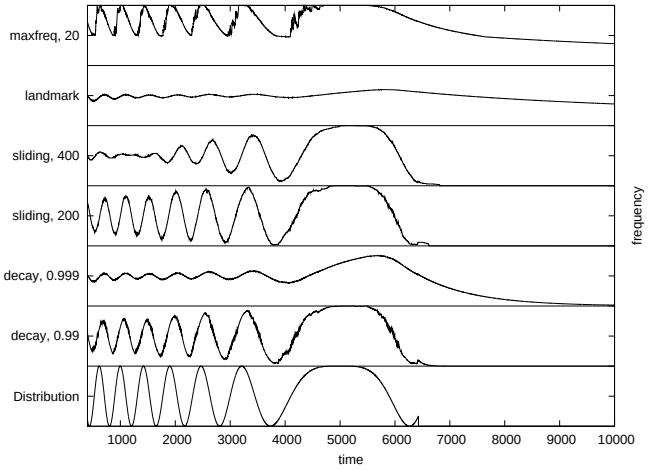


Figure 1: Comparison between the different measures for item support

is taken, and costly algorithms are run on the sample. Sketches and summaries help in dealing with the abundance of data. Instead of storing the complete data stream, which is infeasible, a summary of the relevant features is kept. This summary allows for answering queries about the stream approximately.

The important tasks studied in stream mining include the detection of change (or concept drift) in a stream [20, 24, 9, 42]. By detecting change, we can detect anomalies in the stream, for example attacks in a network.

The other stream mining tasks can roughly be divided into two groups: searching for local patterns and finding global models. The first category is the most relevant one for this project proposal and recent work includes the mining of frequent items [8, 34] and sets of items [33, 17]. Different variations of this problem have been studied, based on how the frequency of an item(set) is counted. These measures can roughly be divided into three categories: sliding window models [12, 18, 21, 22, 32, 45, 11], time fading models [31], and landmark models [21, 22, 46]. In the time-fading model, the entire stream is taken into account when counting frequency, but more recent transactions contribute more than older ones. This is achieved by introducing a decay factor $0 < d < 1$. The closer to 1 the decay, the more the history is taken into account. In the sliding window model, at every time point only the data in the most recent window of a fixed length is considered, whereas in the landmark model, particular time points, called landmarks, are fixed. The second category of mining problems over streams; i.e., finding global models [16, 39, 40] for classification, clustering or prediction is of less relevance for this proposal.

Shortcomings of the Current State-of-the-Art Techniques

We do, however, identify the following two important shortcomings in the current works:

1. The sequential and temporal nature of the stream is taken into account in a rather crude way: only the last entries of the stream are considered to be important, and older entries are either completely discarded, or their influence diminishes very rapidly. The stream is actually considered as if it is a regular database, constantly being updated; entries leaving the window are deleted, entries entering the window are added to the database.

As a result, the time aspect is treated similarly for all items in the stream. In many situations, however, different items behave quite differently and have different life-spans. For example, different products in a store may behave very differently over time; some may show recurring patterns bounded to time of the day (e.g., newspapers), to days of the week (e.g., magazines), whereas others to seasons or holidays (e.g., ice-cream). As such it is not possible to select one unique best window length for all items.

For more complex data types, this difference becomes even more prevalent; e.g., for large sequences of items, it makes sense to consider a larger horizon than for smaller sequences. With the current methods, making such distinctions is not possible. In Figure 1 different support measures have been illustrated for item a in a stream with 10,000 timepoints. The item a is produced at random with the probability given by the bottom line. As can be seen, the actual value of support highly depends on the exact parameter setting.

2. The current methods cover the discovery of frequent items or itemsets, but only few works [13, 2, 29] consider the discovery of more complex patterns such as sequences or subgraphs. In many settings, however, it is natural to consider time-changing graphs where edges are constantly being deleted or added. For example, the World-Wide Web can be seen as a constantly evolving graph where links are being added and deleted. Another example are social networks where the structure of the network is changing as new people enter the network while others leave, and constantly new links are being created. To study the evolution of such networks, dynamic graph patterns need to be considered.

   Patterns in a graph may be identified at different levels. There may be correlations between nodes of the graph at the same point in time, for instance: when the output of any sensor x of type A is high, that of sensors of type B connected to x tends to be low. We call these static patterns, as they refer to a particular state of the graph. But also the short-term evolution of these graphs may be important, for instance: when a certain graph pattern P occurs at time point t, some other pattern P' will occur in the neighborhood of P at time $t + 1$. (Think of a truck driving over a bridge, and imagine sensors simply measure the vibrations of the road surface: if a sensor measures the passing-by of the truck, nearby sensors will measure the same event at a later time point; which sensors these are depends on the direction in which the truck is driving, and the time difference depends on the speed of the truck). This type of patterns we call dynamic graph patterns, and they may be represented as sequences of subgraphs, partial orders over nodes, ... Finally, the long-term evolution of both static and dynamic graph patterns (which we can refer to as "concept drift") is relevant. In Figure 2 two hypothetical examples of very simple, unlabeled graph patterns have been given. In a social network context where the nodes represent people and the edges friendship ties, these two patterns could be interpreted as: "*A new person entering the social network connecting to a person who is part of a group of friends is likely to connect to the other persons in the group as well at a later time*" (top one) and "*If, in a group of friends, the ties between two persons are broken, it is likely that the group falls apart*" (bottom one.)

   Notice that similar problem settings have been considered in [13, 2, 29]. But, the solutions proposed there are insufficient as these works either consider only small subproblems which can be reduced to classical data mining techniques [13], or, the patterns to be discovered are either periodical occurrences of the same pattern [29]; i.e., there is no graph evolution, or the window in which a pattern needs to occur is fixed [2].

Problem Statement

In the project we want to tackle these two problems, and try to push frequent pattern mining on streams beyond the current state-of-the-art techniques, by providing a better handling of the sequential and temporal nature of the streams, without treating all items similarly and by extending these methods so that they cover more complex patterns. The result of this research project will be the introduction of new support measures for sequences and evolving graphs in data streams, algorithms to mine all patterns that are frequent according to this new frequency measure, and implementations of these algorithms. In order to allow verification of the results, the implementations will be made freely available for the research community and the performance will be tested on freely available benchmark data sets, allowing for verification and reproducibility of the results by the research community. Furthermore, the practical relevance will be illustrated by comparing traditional techniques with the newly developed techniques for a practical case, *Hollandse Brug* (see also Section 6b: Application Perspective).

Solving these two problems will enable new applications for analyzing dynamic graph patterns. Such patterns are becoming increasingly more important with the ever growing availability of large social networks data and sensor networks. In Section 6b: Application Perspective some concrete examples are given where the techniques developed within the project can help.

Approach

Subprojects We identify two major sub-projects:

1. Redefine pattern mining tasks to make them more appropriate and feasible for the stream mining domain. More concretely, this involves the development of new frequency measures for patterns such as sequences and graphs that take into account the sequential and temporal nature of data streams, and that allow for different patterns to be evaluated against different time horizons. In [3, 4] a promising new frequency measure for frequent item sets was introduced. This new measure is able to find sudden bursts in data streams, while still taking into account the history of the data stream. It will be extended to other types of patterns as well, with frequent sequential patterns as first candidate. To give a rough idea of how the support measure could be applied for, e.g., scoring sequences in a stream, suppose the following stream arrived (abc denotes that items a, b, and c arrived at the same time): $\langle\ a, b, b, a, ab, ac, a, bc, abc, c\ \rangle$
   What is the frequency of a followed by b? The most traditional answer would be to divide

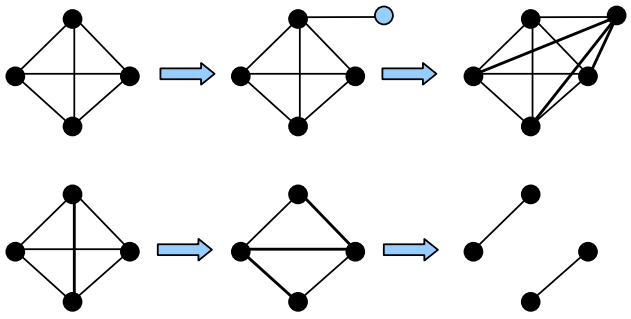


Figure 2: Two examples of potential graph patterns.

the stream into windows of fixed length and count the number of windows in which a followed by b occurs. But how does one need to set this length? And, also, depending on the items in the stream, other window lengths might be more appropriate. Hence, following a similar strategy as in [3, 4], we could decide to take as a division the one that would maximize the frequency of the sequence, whence, giving it "the benefit of the doubt." Although this might seem counter-intuitive, for itemsets it turned out to be quite effective and computationally tractable [3, 4].

2. Constructing algorithms for more complex patterns, such as sequential patterns, partial orders or general graph structures, given the new definitions and notions developed in subproject 1. While many results already exist on finding frequent subgraphs, frequent subsequences, etc., the kind of graph patterns that we are looking for is much more complex, as it generalizes over static subgraphs as well as sequences of items or itemsets. We are interested in finding graph patterns that extend over several time points, and of which sequences of graphs, partial orders over nodes, graph grammars, etc. are special cases.

   Based on algorithms for graph theory and operations research, we can formulate the search for patterns as an optimization problem under constraints. For linear or quadratic optimization functions efficient optimization under linear constraints is possible using mathematical programming techniques, such as linear and quadratic programming, as long as the domains are real-valued. We plan to use submodular optimization, which can be conceived as a discrete analogue of convex optimization. For the complex problems, we will also incorporate local methods.

   There is currently no research on discovering this kind of graph patterns. The Leiden group has expertise on multi-relational mining [25, 26] and the mining of static graph patterns [36, 41], and is currently performing research on discovery of graph grammars, which may describe graph evolution [1]. Moreover, the group is well-connected with the research lab of professor Luc De Raedt in Leuven, Belgium, where methods for mining sequences of complex data are studied [44, 23, 30].

Solving the Problems Approximately One of te main challenges we are confronted with in this project is to deal with the contradicting requirements of having streaming data which requires fast processing of the data on the one hand, and the complexity of mining the numerous graph patterns in the stream. To make this discovery of patterns more efficient, condensed representations [6] will be considered. For the itemset domain this was applied successfully [7]. For the other pattern types, recently an extension of the so-called Non-Derivable Itemset based representations [5] towards the sequence domain was proposed [38]. Next to exact condensed representations, we will also have to consider qualitative approximate solutions, based on maximally informative itemsets and pattern teams [27, 28]. There already exists very efficient summary-based models for mining frequent items in a stream with high accuracy. These are often based on maintaining a remarkably small summary of the stream that yet allows for making highly accurate estimations of the frequencies of the top-K items in the stream [35].

Methodology We start the project into two parallel directions. The first one is the development of the new support measures and in the second one it will be explored how pattern mining on streams can be performed efficiently. In order to assess the usefulness of the new definitions, with controlled experiments it will be shown that the new measures are able to capture patterns where other measures fail to spot them. Then, new algorithms will be developed for mining on streams with the new support measures, joining again the two parallel tracks. Simultaneously with the development of new algorithms for mining all patterns, condensed representations will

be developed for them. The mining algorithms will then be refined using the newly developed condensed representations. Because of the delicate interleaving of the subtasks, apart from the electronic contacts, the teams of TU/e and ULeiden will frequently meet to discuss the progress of the project. In order to test the developed algorithms, implementations will be made and will be tested on publicly available datasets such as the stream of page and link additions and deletions on the World-Wide Web or social network data. Next to the publicly available datasets, in the project we will have access to the "*Hollandse Brug* dataset" which will allow validation by domain experts of the usefulness of the new algorithms.

Dissemination *All implementations will be made freely available for the research community in order to stimulate both validation of the results as well as to encourage dissemination.* We aim for publishing the results of this research in high-level data mining conferences, such as ACM SIGKDD, IEEE ICDM, ECML/PKDD, SIAM DM, and data mining journals, e.g., DAMI, KAIS, and ACM TKDD.

Application Perspective

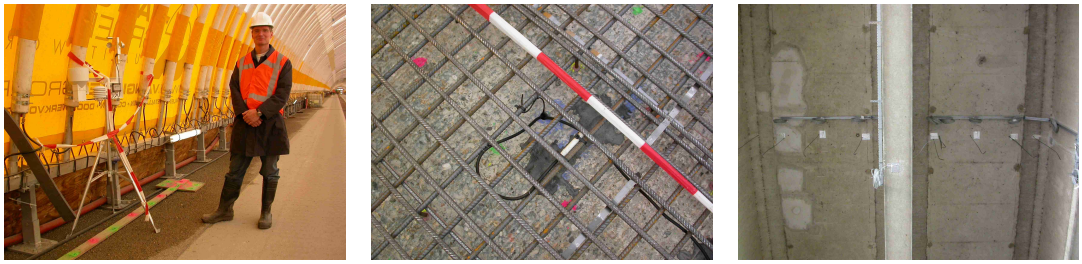


Figure 3: The *Hollandse Brug*. Left: a weather station. Middle: sensors in the top layer of the bridge. Right: cables for transporting the signals below the bridge.

In the project we have access to data generated by the *Hollandse Brug*. The *Hollandse Brug* (Holland Bridge) is the connecting bridge between Flevoland and North-Holland, located at the place where Lake Gooi becomes Lake IJ. Under normal circumstances this bridge is heavily used by car traffic. At this moment, this bridge is being renovated and broadened by the combination "Hollandse Brug," consisting of the construction companies Strukton Civiel BV and Reef Infra BV and *Bouwdienst Rijkswaterstaat* (Dutch National body for road maintenance). These partners are investing in a sensor-network on the bridge. Figure 3 shows a weather station (left picture) and the implementation of the sensors in the surface of the bridge (middle and right picture). This construction is a first step towards an *intelligent bridge*; a bridge constantly monitoring its conditions, processing the data and communicating with the outside world.

We list a couple of problems identified by the domain experts and sketch how the techniques developed inside the project might help solving them:

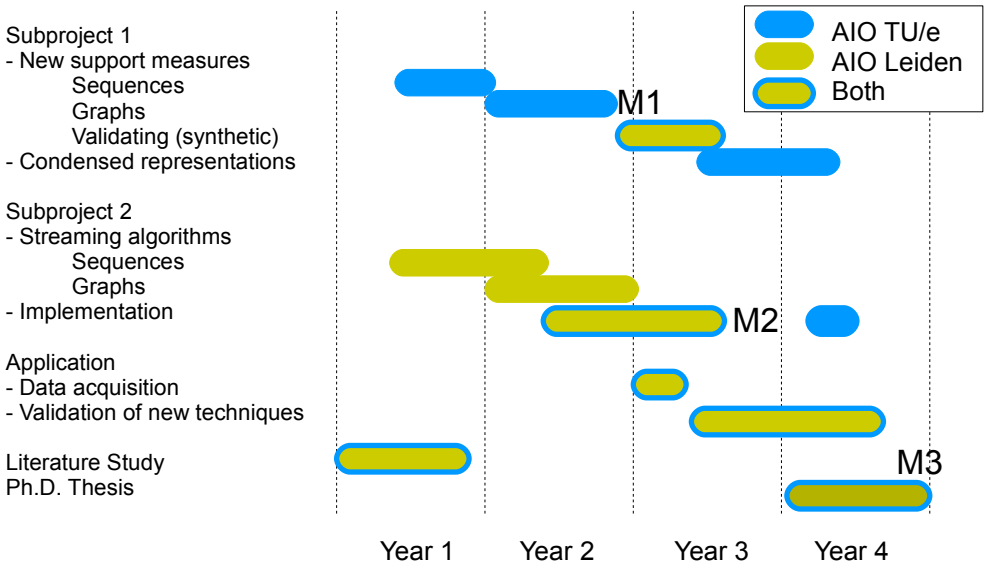
- *How can we analyse the data coming from the bridge in an intelligent way; i.e., which data is important, how can we downsize the amount of information needed to be stored ?* This question is similar the condensed representation problem: a condensed representation stores only the information really needed and omits infrequent and redundant information.
- *What is the influence of the traffic on the bridge, and are there triggers that can be used to predict an increase in the traffic on the bridge?* The influence of the traffic on the bridge can be described using frequently occurring patterns. By identifying patterns regularly reoccurring in combination with a starting increase in traffic, future predictions of upcoming heavy traffic might be made.
- *Can we detect, in an early stage, if the construction of the bridge is degrading?* This question is related to the notion of concept drift. We can assume that the degradation of the bridge will result in a slight shift in the pattern being observed; e.g., one can easily imagine a scenario in which the presence of microscopic cracks affects the time vibrations need to travel from one sensor to another or it might even change the frequencies of the vibrations. The mining of frequent dynamic graph patterns can help in the detection of this type of drift.

For deploying potential solutions developed in the project in an operational setting, the participants plan subsequent or even partly parallel applications for funding at, e.g., *STW* (*Stichting voor Technologie en Wetenschap*—A Dutch fund for supporting application-oriented technical and scientific research).

5 Project Planning

Timeline

The following chart indicates the timeline of the project. The blue bars indicate activities carried out by the AIO at the TU/e, guided by the TU/e team members, whereas the yellow bars indicate activities carried out in Leiden by the other AIO. Bi-colored bars indicate joint activities.



As already pointed out in the Subsection "Approach" of Section "6a: Description of Proposed Research," we distinguish two sub-projects:

1. Define new support measures for the pattern mining tasks sequence mining and dynamic graph mining on streams.

2. Construct algorithms for the more complex patterns types on streams.

The two sub-projects will not be executed sequentially, but in parallel and in close cooperation between the TU/e and ULeiden. The TU/e members will take the lead for the first subproject, and the second sub-project will be the responsibility of the ULeiden team. The domain expert of Strukton will be involved in the tasks "Data Acquisition" and "Validation." Although every pattern type studied in the sub-projects has its own peculiarities and will involve its own set of problems, we reasonably expect to be able to build on the previous results.

The timeline closely follows the chronology of this approach. Both PhD students start their work with literature study and end the project by writing their PhD dissertations. In between, three important milestones can be distinguished:

M1 The definition of new support measures for sequences and graphs. This milestone consists of two subtasks and will be finished at the end of the second year of the project.

M2 The algorithms for mining the more complex graph pattern types in streams are developed and implemented; half of year 3.

M3 The end of the project: the developed techniques have been tested and validated in co-operation with the domain expert. The results of the project are described in two PhD dissertations.

During the course of the project the following deliverables will be produced:

D1 Implementations of the algorithms developed in the project. These implementations will be made public and freely available for research purposes.

D2 Dissemination: several publications at high-level conferences and journals.

D3 & D4 Two PhD dissertations.

References

[1] H. Blockeel and S. Nijssen. Induction of node label controlled graph grammar rules. In *Proceedings of the 6th International Workshop on Mining and Learning with Graphs*, pages 1–4, 2008.

[2] K.M. Borgwardt, H.P. Kriegel, P. Wackersreuther, and G. Munich. Pattern mining in frequent dynamic subgraphs. In *Proceedings of the Sixth International Conference on Data Mining*, pages 818–822. IEEE Computer Society Washington, DC, USA, 2006.

[3] T. Calders, N. Dexters, and B. Goethals. Mining frequent itemsets in a stream. In *Proceedings IEEE International Conference on Data Mining (ICDM)*, pages 83–92, 2007.

[4] T. Calders, N. Dexters, and B. Goethals. Mining frequent items in a stream using flexible windows. *Intelligent Data Analysis*, 12(3), 2008.

[5] T. Calders and B. Goethals. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1):171–206, 2007.

[6] T. Calders, C. Rigotti, and J-F. Boulicaut. A survey on condensed representations for frequent sets. In J-F. Boulicaut, L. de Raedt, and H. Mannila, editors, *Constraint-Based Mining*, volume 3848 of *LNCS*. Springer, 2006.

[7] Y. Chi, H. Wangt, PS Yu, and RR Muntz. Moment: maintaining closed frequent itemsets over a stream sliding window. In *Data Mining, 2004. ICDM 2004. Proceedings. Fourth IEEE International Conference on*, pages 59–66, 2004.

[8] G. Cormode and S. Muthukrishnan. What's hot and what's not: tracking most frequent items dynamically. *ACM Transactions on Database Systems (TODS)*, 30(1):249–278, 2005.

[9] G. Cormode and S. Muthukrishnan. What's new: finding significant differences in network data streams. *IEEE/ACM Transactions on Networking (TON)*, 13(6):1219–1232, 2005.

[10] E.H. de Graaf. *Mining semi-structured data, theoretical and experimental aspects of pattern evaluation*. PhD thesis, Leiden Institute of Advanced Computer Science, Faculty of Science, Leiden University, 2008.

[11] E.H. de Graaf, J.M. de Graaf, and W.A. Kosters. Using consecutive support for genomic profiling. In M. Hilarion and C. Nédellec, editors, *ECML/PKDD-2006 Workshop on Data and Text Mining for Integrative Biology*, pages 16–27, 2006.

[12] E. D. Demaine, A. L.-O., and J. I. Munro. Frequency estimation of internet packet streams with limited space. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA)*, pages 348–360, 2002.

[13] P. Desikan and J. Srivastava. Mining temporally evolving graphs. In *Proceedings of the the Sixth WEBKDD Workshop in conjunction with the 10th ACM SIGKDD conference*, volume 22, 2004.

[14] M.M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: a review. *ACM SIGMOD Record*, 34(2):18–26, 2005.

[15] J. Gama, J. S. Aguilar-Ruiz, and R. Klinkenberg. Knowledge discovery from data streams. *Intell. Data Anal.*, 12(3):251–252, 2008.

[16] J. Gama and P. P. Rodrigues. Stream-based electricity load forecast. In *Proceedings Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 446–453, 2007.

[17] C. Giannella, J. Han, J. Pei, X. Yan, and P.S. Yu. Mining frequent patterns in data streams at multiple time granularities. In *Next Generation Data Mining*, pages 191–212. Cambridge, Mass: MIT Press, 2003.

[18] L. Golab, D. DeHaan, E.D. Demaine, A. Lopez-Ortiz, and J.I. Munro. Identifying frequent items in sliding windows over on-line packet streams. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 173–178. ACM New York, NY, USA, 2003.

[19] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[20] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM New York, NY, USA, 2001.

[21] R. Jin and G. Agrawal. An algorithm for in-core frequent itemset mining on streaming data. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 210–217, 2005.

[22] R.M. Karp, S. Shenker, and C.H. Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems (TODS)*, 28(1):51–55, 2003.

[23] K. Kersting, L. De Raedt, B. Gutmann, A. Karwath, and N. Landwehr. Relational sequence learning. In *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Computer Science*, pages 28–55. Springer, 2008.

[24] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very Large Data Bases*, pages 180–191. VLDB Endowment, 2004.

[25] A. Knobbe. *Multi-Relational Data Mining*. IOS Press, Amsterdam, the Netherlands, 2006.

[26] A. Knobbe. Safarii multi-relational data mining environment. http://www.kiminkii.com/safarii.html, 2006.

[27] A. Knobbe and E. Ho. Maximally informative k-itemsets and their efficient discovery. In *Proceedings KDD'06*, pages 237–244, Philadelphia, PA, 2006.

[28] A. Knobbe and E. Ho. Pattern teams. In *Proceedings PKDD'06*, pages 577–584, Berlin, Germany, 2006. Springer-Verlag.

[29] M. Lahiri and T. Y. Berger-Wolf. Mining periodic behavior in dynamic social networks. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, December 2008.

[30] N. Landwehr and L. De Raedt. r-grams: Relational grams. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 907–912, 2007.

[31] D. Lee and W. Lee. Finding maximal frequent itemsets over online data streams adaptively. In *ICDM*, pages 266–273, 2005.

[32] C.H. Lin, D.Y. Chiu, Y.H. Wu, A.L.P. Chen, and T. Hsinchu. Mining frequent itemsets from data streams with a time-sensitive sliding window. In *Proceedings of the 5th International Conference on Data Mining*. Society for Industrial Mathematics, 2005.

[33] G.S. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 346–357. VLDB Endowment, 2002.

[34] A. Metwally, D. Agrawal, and A. El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *Proceedings of the 10th ICDT International Conference on Database Theory*, pages 398–412. Springer, 2005.

[35] S. Muthukrishnan. *Data Streams: Algorithms And Applications*. Now Publishers Inc, 2005.

[36] S. Nijssen and J. N. Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 647–652, 2004.

[37] S.G.R. Nijssen. *Mining Structured Data*. PhD thesis, Leiden Institute of Advanced Computer Science, Faculty of Mathematics and Natural Sciences, Leiden University, 2006.

[38] C. Raïssi, T. Calders, and P. Poncelet. Mining conjunctive sequential patterns. *Data Mining and Knowledge Discovery*, 17(1):77–93, 2008.

[39] P. P. Rodrigues, J. Gama, and L. M. B. Lopes. Clustering distributed sensor data streams. In *Proceedings Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 282–297, 2008.

[40] P. P. Rodrigues, J. Gama, and J. P. Pedroso. Hierarchical clustering of time-series data streams. *IEEE Trans. Knowl. Data Eng.*, 20(5):615–627, 2008.

[41] L. Schietgat, J. Ramon, M. Bruynooghe, and H. Blockeel. An efficiently computable graph-based metric for the classification of small molecules. In *Proceedings of the 11th International Conference on Discovery Science*, volume 5255 of *LNAI*, pages 197–209, 2008.

[42] E. J. Spinosa, A. C. P. de Leon Ferreira de Carvalho, and J. Gama. Olindda: a cluster-based approach for detecting novelty and concept drift in data streams. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 448–452, 2007.

[43] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[44] I. Thon, N. Landwehr, and L. De Raedt. A simple model for sequences of relational state descriptions. In *ECML/PKDD (2)*, volume 5212 of *Lecture Notes in Computer Science*, pages 506–521. Springer, 2008.

[45] R. C.-W. Wong and A. W.-C. Fu. Mining top-k frequent itemsets from data streams. *Data Min. Knowl. Discov.*, 13(2):193–217, 2006.

[46] J.X. Yu, Z. Chong, H. Lu, Z. Zhang, and A. Zhou. A false negative approach to mining frequent itemsets from high speed transactional data streams. *Information Sciences*, 176(14):1986–2015, 2006.