Mining Sequences in Distributed Sensors Data for Energy Production

Mehmed Kantardzic and John Gant

Computer Engineering and Computer Science, The University of Louisville
mmkant01@louisville.edu
jdgant01@louisville.edu

Abstract

The desire to predict power generation at a given point in time is essential to power scheduling, energy trading, and availability modeling. The research conducted within is concerned with sequence mining on power generation data and has the intent of modeling power generation. The data streams analyzed are average hourly power generation that is reported to the EPA. A global statistical model is proven impractical for the data streams, and local modeling via sequence mining is performed. The methodology presented, Uniform Sequence Discovery, implements the idea of uniform population coding, stream mining, and cross-stream mining. 1671 streams from years 2002 through 2004 are coded, mined for sequences, and cross-mined for matching sequences. 486 and 270 frequent sequences were extracted from the learning and testing data respectively. Association rules and the accompanying confidence and support values are used to create local models for power generation prediction. 159 local models were confirmed in the testing phase with a minimum confidence of 0.60. Power traders, concerned with predicting available generation, would then use the local models for prediction of natural gas-fired power generation.

1. Introduction

The estimation of point values is a widely studied problem [6,7,12]. To predict a point value one typically uses a model that will encompass all factors that affect predictability. These factors are gathered into distinct features via many well-studied feature extraction methods. A single model that is typically linear, of a polynomial form $y = a_1 x_1^n + a_2 x_2^n + ... + b$, is also considered to be a global model as its predictive capability includes value prediction for the entire set of distinct features. Association rules are models in the sense that they provide point prediction capabilities, but do not span the entire distinct feature set that is covered by a global model. The inability to cover the entire set of distinct features localizes their predictive capability; therefore, association rules are considered to be local models. A combination of local models can be used to gain a full understanding of the entire dataset. A ***data stream*** is an ordered sequence of data, and typically consists of an infinite set of point value, timestamp pairs which are recorded from a sensor [13,15]. The point value and timestamp pair must be unique, with regards to time. A ***sensor*** is a device that reports an infinite stream of data, which are usually recorded from streams supplied by multiple sensors [13,16]. A collection of sensors that supply the streams of data is deemed a sensor network. Streaming generation data are observed for frequently occurring sequences, or patterns, which, if common and of high reliability, result in the creation of association rules. The problem of mining sequences in streaming data is highly studied [12-18]. Optimizing sequence mining algorithms is of high importance for real-time assessment of local models [3,7,13-18]. Following previous discussions provided by [2,3,4], we pursue the idea of a normal population for the dataset. Once the normal population assumption is proven infeasible, we introduce Uniform Sequence Discovery (USD). USD is a coding and mining process that is applied to data with the intent of reliable local model construction. The process involves a discrete coding of each raw point value and an exhaustive sequence mining operation. Data from years 2002 through 2004 are analyzed, reliable local models are created from the combination of discovered association rules, and reliable support and confidences of large numeric value.

2. Previous Research

The goal of the research reviewed within this paper was to create a modeling scenario for prediction of natural gas-fired power generation. As a result, power traders and regulatory officials may use the model to predict load. To extract useful features from the data, one may use techniques such as correlation, principal components analysis, and clustering algorithms. If local modeling is proven appropriate, sequence mining is the technique of choice. Statistical analysis provides information that will be used during the determination of the modeling constraints.

Eammon Keogh and his colleagues contribute multiple ideas, to sequence mining, that include: normal population theory, the idea that streams contain “don’t cares”, and segmented data sequencing. Normal population is the idea that all temporal data are sampled from a normal population [4]. “Don’t cares” promote the idea that data might be missing (an empty timestamp) or invalid values

are present (data out of the range) [1]. Segmentation is idea of approximating multiple data points, from a stream, using a linear segment which allows for the elimination of noise in data presented to the sequencing algorithm [3].

Frank Höppner contributes the idea of time series representation using derivatives [6], specifically that seven basic shapes exist which derivatives represent therefore creating a coding schema. He explores the first and second derivatives on a point basis, and recommends specific techniques for noise avoidance. Later, to combat noise he introduces the idea of kernel smoothing with various filters [11]. With higher filtering constants he is able to extract features from which he assembles quantitative rules.

Researchers at Harvard introduce the idea of using stochastic dictionaries to build sequences [5]. The dictionary is preloaded with common sequences that might be encountered. They discuss that a probability word matrix (PWM), which represents the product of probabilities of a code appearing in sequence, is used as a guide to validating the probability that a sequence will occur. Also included are the ideas of deletions and phase shifts between matching sequences while using the PWM.

Researchers from AT&T, The University of Maryland, Carnegie Mellon University, The University of Michigan, and The University of Virginia introduce the MUSCLES algorithm. The MUSCLES algorithm [7] uses the idea of least squares approximation where sequences are the independent factors used in approximation of the regression coefficient. The MUSCLES algorithm has three noted uses that include correlation detection, outlier detection, and interpolation.

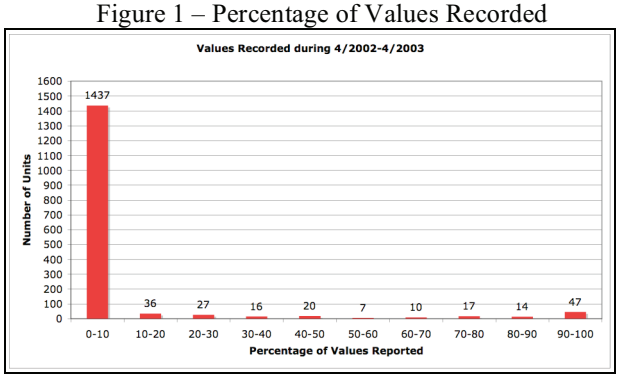
We propose an analysis to determine the constraints of modeling power generation. To define modeling constraints we analyze possible feature extraction using correlation, principal components analysis, and k-means clustering. Once local modeling is proven appropriate, the normal population assumption is examined. Uniform sequence discovery is introduced and applied resulting in the construction of local models using association rules and summary statistics.

3. Data Analysis

In this section we briefly overview the operation of power generating units, describe generation data, analyze the normal population assumption, and define the constraints of the model to be developed. To define the constraints we attempt to discover distinct features in the data using correlation, principal components analysis, and k-means clustering. Finally we conclude with a brief review of the results and recommendations for the model constraints. The Environmental Protection Agency, EPA, is in charge of levying fines to corporations who defy pollution limits, by analyzing the data reported by SCADA systems. Supervisory Control and Data Acquisition, or SCADA, provide recording and reporting of operational and emissions related data within a power plant setting. The data streams from distributed generating units, to the EPA, which are connected to the nationwide power grid. The data are hourly averages of power (MWH). Generation data must be submitted if the unit has a capacity, or maximum theoretical output, greater than 25 MWH. The data recorded are from generating units of many different fuel types, but the data studied are from pipeline natural gas-fired generating units. These units run solely off of natural gas fed through pipelines and exclude cogeneration units, which recover heat and use it for another process. The power industry, since deregulation, has adapted its operational characteristics to provide optimum profit. The restructuring of the operations of power generation result in demand oriented patterns. From the author's domain experience there are some basic patterns existing in power generation. First a daily cyclic pattern exists, and operates in a bell shape starting with its peak at 13:00 EST. Generators are ramped up around 6:00 EST to prepare for the early morning demand, and tend to ramp down after 20:00 EST. The second pattern is a weekly pattern. Monday through Friday are high power usage days and on the weekend the power usage drops. The weekly pattern is disrupted by holidays, when customers are at home and businesses are not operating. The third and final pattern, which is heavily influenced by national and regional temperature, is downtime. Usually power stations have a downtime during the off-peak season, which is usually after October and before April. During the off-peak season output will be lower unless a cold streak occurs in which electric heaters will be operational. Within the scope of power generation units exist a few distinct operational characteristics based upon fuel type. First a typical coal, or oil-fired unit is used to help make up the base load. Base load is an estimated expected load. These units are chosen due to the cost of fuel, i.e. coal is the most cost effective way to generate power. Both oil and coal-fired plants can change generation output fairly quickly. Next is hydroelectric or nuclear generation, which has a very stable and smooth output, who also make up base load. This means that neither of these generation types change output quickly as they require more time to reduce or increase load. The last type of unit is a peaker unit, which is natural gas-fired. Peaker units can be started and stopped quickly (within minutes), allowing for spontaneous generation of the unpredicted power needs above base load. These units are the most unpredictable from past data and are the type we attempt to model within this document. The units that make up base load do follow the cyclic patterns mentioned earlier, therefore; cyclic patterns will not be studied in the analysis. The *learning phase* includes data from natural gas-fired average hourly power generation data for the timeframe of 04/01/2002 through and including 03/31/2003. The *testing phase* includes data from natural gas-fired average hourly power generation data for the timeframe of 04/01/2003 through and

including 03/31/2004. The sampled data are sparse when looking at individual streams. Figure 1 describes the scarcity of data by listing each unit's stream and their corresponding percentage of values recorded for the learning dataset. Each stream is associated with a bin, based upon its percentage recorded.

Figure 1 – Percentage of Values Recorded



The learning phase contains 281 streams of 1631, or 17.2%, that was recorded during the year. The testing phase contains 341 streams of 1631, or 20.8%, that was recorded during the year. The percentages of values reported during the testing phase are similar to those in Figure 1. We concentrated our research on pipeline natural gas-fired power generating units and, as a result, 1631 is the maximum number of unit streams that could have been mined, which is 35.6% of the 4582 units connected to the grid in 2004. Pipeline natural gas-fired units were chosen to allow focus on modeling units whose operational patterns are of high complexity.

3.1 Normal Population Assumption

Eamonn Keogh introduced in [4] the idea of discrete time series being sampled from a normal population. Although looking at individual unit streams confirms the lack of normality, a statistical analysis was performed. To compare the population of the sampled data to the assumed normal population, the Chi-Square "Goodness of Fit" test was performed on the learning dataset. To perform a test on the data, a "typical" example must be extracted from the sparse set of streams. A "typical" unit stream is one that reports for a large period of the year, exhibits a wide range, has high arithmetic mean, and low sample standard deviation. Unit 6A at Barry Generating Station located in Bucks, Alabama was chosen. Table 1 includes some summary statistics about the data from unit 6A, which include count (N), arithmetic mean ($\bar{x}$), sample variance (S^2), sample standard deviation (S), minimum, and maximum range values.

Table 1 – Summary Statistics

N	8759
$\bar{x}$	152.15
S^2	8133.68
S	90.19
Range min	0
Range max	312
Number of bins	10
Bin width	31.2

The Chi Square test statistic (χ_0^2) is exceptionally large in comparison with the expected table statistic ($\chi^2_{(1-\alpha,df)}$). The hypothesis, H_0, where data from the EDR dataset are sampled from a normal population, is rejected:

$$\chi_0^2 = 17551.63$$

$$\chi^2_{(1-\alpha,df)} = \chi^2_{(0.05,9)} = 23.59$$

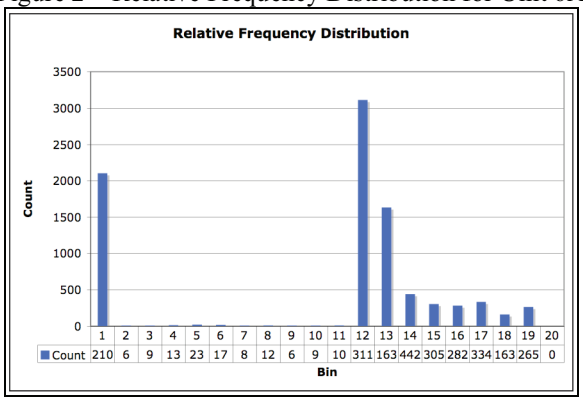
$$\chi_0^2 > \chi^2_{(0.05,9)}, \text{ Reject } H_0$$

Figure 2 is a relative frequency plot of the data represented by the stream used in the Chi Square "Goodness of Fit" test. The square of the Pearson correlation coefficient (r) was used to build a cross correlation matrix for the learning data. The cross correlation matrix was condensed into Table 2, which lists the count of the stream pairs for each rounded correlation coefficient.

Table 2 – Condensed Cross Correlation Matrix for Unit Streams

r^2	Count
0	2652000
0.01	6362
0.02	150
0.03	8
0.04	8
0.05	2
1	1631

Figure 2 – Relative Frequency Distribution for Unit 6A



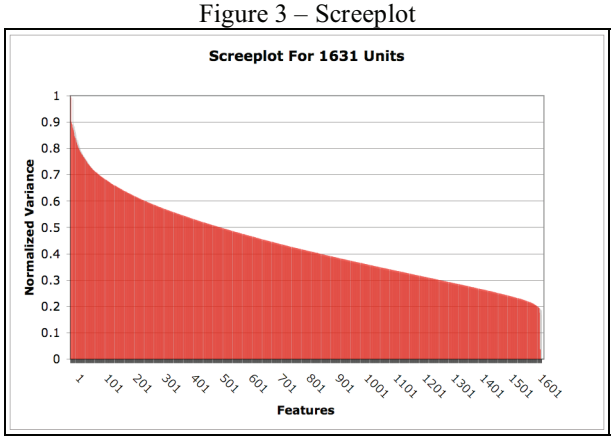
The r^2 values of 1.0 refer to the self-correlations, and as can be seen from the Table 2, there are no numerically similar streams. To extract the dominant features to use for

clustering, which will be used as independent factors in a multivariate regression model, principal components analysis (PCA) was performed. To collect the features that would represent the dataset to maximum efficiency, we expect that the extracted features represent 90% of the variance in the dataset [8]. The analysis was performed on the learning dataset and Figure 3 represents the screeplot, or a plot that exposes each feature and its normalized variance. Typically a researcher looks for a point at which the variance drops sharply, and this point is considered to be the "knee" of the screeplot. As can be seen from Figure 4, the screeplot lacks a "knee". Therefore we now look to account for 90% of the variance in the data, and to do so required that 1159 features be used. 1159 unit streams represent 71% of the 1631 available streams. As a result, a global model would contain 1159 independent factors with no certification that true independence exists. Clustering was chosen as a feature extraction alternative analysis to correlation and principal components analysis. Due to the PCA results, clustering was not performed over the 1159 features extracted and instead 100 clusters were chosen. 100 clusters were chosen as a large, but feasible, number of factors to include in a linear global model. Clustering was performed on the learning data and included using Euclidean distance measure.

Table 3 – Within-Cluster Error Statistics

Minimum	2,715,492
Maximum	2,471,214,401
Median	4,202,800
Sample Mean	148,377,543
Sample Stddev	422,559,223

Figure 3 – Screeplot



Due to the high median, mean, and standard deviation, represented in Table 3, the choice of developing a global model with as few as 100 independent variables would result in high errors between predicted and actual values. Initially there was an assumption that all temporal data is sampled from a normal population distribution, and this has been proven false for the EDR data. The desired result of the research was to build a global model to predict generation and this has been proven infeasible. The PCA results indicate that no optimum feature set can be extracted and used effectively. Clustering also shows that when a feasible number of factors are applied to the data in hopes of generating a global model, it cannot be accomplished with desirable results. The prediction of generation can only be done on a stream level for a short period of time. As a result of the exploratory analysis, local modeling has been chosen as the technique to be used in generation prediction.

4. Uniform Sequence Discovery

The normal population theory has been proven infeasible, and to properly code numerical values (based on Keogh approach) there must be a theoretical population distribution. We propose the idea that all values have equal probability, and as a result, we introduce Uniform Sequence Discovery, or USD. USD requires mining the entire space of sequences in search of frequent sequences. The following steps define Uniform Sequence Discovery: Uniform population coding, Stream sequence mining, cross sequence discovery, and finally local model construction.

Let $D = \{(v_0,t_0),(v_1,t_1),..,(v_{n-1},t_{n-1})\}$ be a set of (value, timestamp) pairs, where the set D of values is considered a ***sequence*** [18,1,2,3,4,]. There can be no duplicate timestamps; therefore, t_i must be unique in the set D. The coding process used during the USD is defined by

$$c_i = floor\left(\frac{v_i - \min(range_V)}{num\_bins}\right) \quad (1)$$

Where $i = 0..n-1$ and $v_i \in V_{range}$, v_i is the raw value, c_i the code for each raw data point in the stream, and num_bins are user-defined. Once each stream had been coded, sequence discovery was performed over the entire space of sequences as defined by USD. The entire space of sequences is infinitely large, and must be contained; as a result the user is responsible for using domain knowledge to determine a "practical and sufficient" minimum and maximum length, $L_{\min}$ and $L_{\max}$ respectively, for a sequence. After coding, cross series sequence mining is performed. This requires that each temporal sequence is discovered and compared with every other stream's sequences. There exists a set of all units, SU, and within that set each unit, U, contains a set of temporal sequences, TS_U. Cross series discovery can be defined formally as,

$$\forall U_i, U_j \in SU \text{ Where } i, j = 0..n-1, j \neq i \quad (2)$$
$$\exists NS = U_i \cap U_j$$

Where NS represents the matching sequences between sets U_i and U_j. Confidence and support values for each discovered sequence determine the validity of defining local models. The ***support***, S, of a sequence, V, is defined as count of sequence V contained within stream St [18,8,13], and can be written as

$$S_{St}(V) = St.count(V) \qquad (3)$$

The ***confidence***, Cf, that sequence A is present at the same time sequence as B in stream St [8] is defined by

$$Cf_{St}(A \mid B) = \frac{count(A \cap B)}{count(B)} \qquad (4)$$

An ***association rule*** where sequence coexists in streams St_1, and St_2 with confidence Cf_{St} [8,17] is be defined by $Ar \rightarrow$ *If sequence* A *exists in stream* St_1*, it will coexist in stream* St_2 *with confidence* Cf.

Local models are constructed from association rules that are defined in the training phase and confirmed in the testing phase by high confidence and support values. Due to the requirements of USD, which include exploring the entire space, an optimal algorithm is preferred for real-time use. For simplicity and trustworthiness, an off-line brute-force approach was used to discover the sequences for each stream and is defined below:

Algorithm: Uniform Sequence Discovery
P= Blank Pattern
foreach stream, $S_i \in Streams$
 for i in numCodes
  buildAndMine (P)
 end #inner for loop
end #outer for loop

Function: buildAndMine
buildAndMine(Pattern P, Stream S_i) (
 if $P.length() \geq L_{\min}$ and $P.length() <= L_{\max}$
  foreach code, $C_i \in \{-1,0,1,2,3\}$
   if $P.isOkByUser()$
    P = concat (P,C_i)
    mineStreamForPattern(S_i,P)
   else
    break
   end #inner if
  end #inner for loop
 end #outer if)

The algorithm contains a process where all possible sequences are built, and each stream is then mined for all possible sequences. The complexity of the algorithm, which is the number of mining operations performed, is defined in equation 5.

$$N_{operations} = N_{streams} * N_{sequences} \qquad (5)$$

Where $N_{streams}$, and $N_{sequences}$ represent the number of streams per phase and the number of possible sequences, respectively. Frequent sequences are extracted along with confidence and support values, which are then used to construct association rules. The algorithm is recursive and terminates only when the maximum length has been exceeded or the user does not verify the sequence. P.isOkByUser() is the method where the user can certify that the sequence does not contain numerous "don't cares", and is not saturated by a code. To use the software effectively the user is required to set some operational parameters prior to execution that include numThreads, minimumNumberOfMatches, minLength, maxLength, miniumPattern, maximumPattern, sequenceSaturationPercentage, bannedPatternList, and leadOrLagCount.

5. Experimental Results

The tuned initial values for operational parameters, used during the sequence discovery during the learning phase, are listed below:

1. minimumPattern = -1
2. maximumPattern = 3
3. minimumPatternLength = 3
4. maximumPatternLength = 8
5. numThreads = 10
6. sequenceSaturationPercentage = 80
7. bannedPatternList = -1,000
8. leadOrLagCount = 0
9. minimumNumberOfMatches = 5

The total number of possible sequences, $N_{sequences}$, is defined by, $N_{sequences} = C^{L_{\max}} - C^{L_{\min}}$. Where $L_{\max}$, $L_{\min}$, and C are the maximum sequence length, minimum sequence length, and number of codes, respectively. There are 390,500 possible sequences of length 3 up to length 8 for a single stream. Of the 109,730,500 possible sequences during the learning phase, 4702 (0.004%) were discovered and 486 (10.336%) of those were frequent and of high confidence (Cf > 0.60). Of the 133,160,500 possible sequences during the testing phase, 3609 (0.003%) were discovered and 270 (7.481%) of those were frequent and of high confidence (Cf > 0.60). 159 sequences were discovered in the learning phase and reoccurred, with high confidence and support, in the testing phase. 159 association rules were created from the patterns, and a sample of the association rules is represented in Table 4. Table 4 includes the unit identifier, pattern, learning support, learning confidence, and testing confidence.

6. Conclusion

This research was focused on creating a modeling scenario for power generation. In doing so we were able to prove that distinct features used in global modeling were not

prominent using correlation, PCA, and clustering. To perform sequence mining according to Uniform Sequence Discovery, we codified the raw data using a uniform population distribution, once the normal population assumption was disproved. 486 frequent sequences were discovered in the learning phase, and 159 reoccurred with high confidence and support in the testing phase. 159 local models were constructed using association rules with a minimum confidence of 0.60. Using the models in a business setting would involve updating and creating new models based upon current data. Future research should focus on identifying a more accurate population distribution along with optimizing the mining algorithm in preparation for real-time mining.

Table 4 – Sample of Association Rules

u1	u2	Sequence	Learning Phase		Testing Phase
			Sup(u1)	Conf(u1|u2)	Conf(u1|u2)
1	2	3,3,3,3,3,3,2,2	76	0.79	0.78
2	1	3,3,3,3,3,3,2,2	81	0.74	0.78
1	2	3,3,3,3,3,2,3,2	7	0.86	1
2	1	3,3,3,3,3,2,3,2	8	0.75	1
1	2	3,3,3,3,3,2,2,2	81	0.83	0.63
2	1	3,3,3,3,3,2,2,2	87	0.77	0.7

References

1. Chiu B, Keogh E, Lonardi S. Probabilistic Discovery of Time Series Motifs. In Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.

2. Keogh, E, Lonardi S, Chiu B. 2002. Finding Surprising Patterns in a Time Knowledge Discovery and Data Mining; 2002; Jul 23-26; Edmonton, Alberta (Canada).

3. Keogh, Eamonn. A Fast and Robust Method for Pattern Matching in Time Series Databases. 1997.

4. Lin J, Keogh E, Lonardi S, Patel P. Finding Motifs in Time Series. Proceedings of the Second Workshop on Temporal Data Mining. 2002: Edmonton, Alberta (Canada).

5. Gupta M, Liu S. Discovery of Conserved Sequence Patterns Using a Stochastic Dictionary Model. American Statistical Association, Journal of the American Statistical Association. 2003;Vol. 98, No. 461, Theory and Methods.

6. Höpper F. Learning Dependencies in Multivariate Time Series. European Conference on Artificial Intelligence. 2002.

7. Byoung-Kee Y, Sidiropoulos N, Johnson T, Jagadish H, Faloutsos C, Biliris A. Online Data Mining for Co-Evolving Time Sequences. In Proceedings of the IEEE Sixteenth International Conference on Data Engineering, pages 13--22, 2000.

8. Kantardzic M. Data Mining: Concepts, Models, Methods, and Algorithms. Wiley – Institute of Electrical and Electronics Engineers. 2003.

9. Gant J. Mining Sequences in Distributed Sensors Data for Energy Production [thesis]. Louisville (KY): University of Louisville, Speed School of Engineering; 2006 Sept 15. 152 p.

10. Höpper F. Time Series Abstraction Methods – A Survey. In Proceedings GI Jahrestagung Informatik, Workshop on Knowledge Discovery in Databases, Lecture Notes in Informatics, pages 777--786, Dortmund, Germany, 2002.

11. Papadimitriou S, Sun J, Faloutsos, F. Steaming Pattern Discovery in Multiple Time Series. Proceedings of the 31st Very Large Database Conference, Trondheim, Norway, 2005.

12. Loo K, Tong I, Kao B, Cheung D. Online Algorithms for Mining Inter-Stream Associations From Large Sensor Networks.

13. Govindaraju N, Raghuvanshi N, Maocha D. Fast and Approximate Stream Mining of Quantiles and Frequencies Using Graphics Processors. International Conference on Manangement of Data Proceedings of the 2005 ACM SIGMOD.

14. Talia D, Trunfo P, Orlando S, Perego R, Silvestri C. Systems and techniques for distributed and stream data mining, CoreGRID Technical Report Number TR-0045, 2006.

15. Sakurai Y, Papdimitriou S, Faloutsos C. AutoLag: Automatic Discovery of Lag Correlations in Stream Data. Proceedings of the 21st International Conference on Data Engineering, 2005.

16. Jiang N, Gruenwald L. Research Issues in Data Stream Association Rule Mining. Association of Computing Machinery Special Interest Group on Management of Data, Volume 35, Number 1, 2006.

17. Marascu A, Masseglia F. Mining Sequential Patterns from Temporal Streaming Data. Proceedings of the 1st ECML/PKDD Workshop on Mining Spatio-Temporal Data, 2005.