A Wireless Data Stream Mining Model

Mohamed Medhat Gaber[1], Shonali Krishnaswamy[1], and Arkady Zaslavsky[1]

[1] School of Computer Science and Software Engineering, Monash University,
900 Dandenong Rd, Caulfield East, VIC3145, Australia

Abstract. The sensor networks, web click stream and astronomical applications generate a continuous flow of data streams. Most likely data streams are generated in a wireless environment. These data streams challenge our ability to store and process them in real-time with limited computing capabilities of the wireless environment. Querying and mining data streams have attracted attention in the past two years. The main idea behind the proposed techniques in mining data streams in to develop efficient approximate algorithms with an acceptable accuracy. Recently, we have proposed algorithm output granularity as an approach in mining data streams. This approach has the advantage of being resource-aware in addition to its generality. In this paper, a model for mining data streams in a wireless environment has been proposed. The model contains two novel contributions; a ubiquitous data mining system architecture and algorithm output granularity approach in mining data streams.

1 Introduction

The process of data mining had been centralized, and then distributed among the data sources for bandwidth preservation and privacy contrarians of transferring the data. The dissemination and increasing power of wireless devices have stimulated the need for wireless data mining. In this scenario, the mobile device is receiving or generating a high data rate stream of information that should be analyzed in real time.

Projects found in [5][20][21][23] demonstrate the need for data stream analysis techniques and strategies in a wireless environment that can cope with the high data rate and deliver the analysis results in real time in resource constrained environments.

There is a need for a wireless data mining architecture that efficiently can cope with limited capabilities of the computing and communication power in a wireless environment. RA-UDM is our proposed system architecture in mining data streams in a wireless environment. RA-UDM is a resource-aware ubiquitous data mining system architecture that has the advantage of generality and adaptability. The adaptability of this system is a result of using algorithm output granularity in mining data streams.

The algorithms proposed so far in the literature in mining data streams try to develop approximate solutions that have only one pass or less on the incoming stream. Recently, algorithm output granularity (AOG) [14], [15] has been proposed as an approach in mining data streams. AOG is a resource-aware adaptable approach in mining data streams. In this paper, we present a ubiquitous data mining architecture that incorporates the AOG approach in mining data streams.

The paper is organized as follows. Section 2 presents the related work in mining data streams. The proposed ubiquitous data mining system architecture is discussed in section 3. The algorithm output granularity approach is discussed in section 4. Sec-

Medhat Gaber M., Krishnaswamy S. and Zaslavsky A. (2004).
A Wireless Data Stream Mining Model.
In *Proceedings of the 3rd International Workshop on Wireless Information Systems*, pages 152-160
DOI: 10.5220/0002676301520160

tion 5 shows the application of algorithm output granularity to different mining techniques. Finally, we conclude the paper and show our future work in section 6.

2 Related Work

There are different algorithms proposed to deal with the high speed feature in mining data streams using different techniques. In this section, we present the related work in mining data streams. Clustering data streams has been studied in [16], [10], [7], [1], [27], [22], [3], [6], [9]. Data stream classification has been studied in [11], [19], [12], [30]. Extracting frequent items and frequent itemsets have been studied in [8], [13], [24]. Thorough discussion about data streams could be found in [17], [18], [25], [26]. The main focus of the above algorithms is how to reduce the number of passes and the number of instances being tested in order to have an efficient approximate algorithm. AOG approach in mining data streams is distinct by being resource-aware. In the following section, the wireless data mining system architecture that incorporates AOG approach is illustrated.

3 RA-UDM System Architecture

The research so far in the UDM field does not pay much attention of how to exploit the increasing computation power of mobile devices in the data mining task. Motivated by this fact and the increasing need for resource-aware data analysis systems in commercial and scientific applications for the huge data streams generated continuously, we propose a new resource aware UDM system that incorporates our approach in mining data streams using algorithm output granularity. Figure 1 shows our resource aware ubiquitous data mining (RA-UDM) system architecture. The different components of the systems are explained in details in this section.

Resource-aware Component

Local Resource Information: This module has the ability to inform the system about the mobile device resources' measurement such as the available memory, CPU utilization, battery consumption… etc. The quantification of the performance of data mining algorithms from the energy consumption of the mobile device perspective has been studied experimentally in [4]. Preliminary results for resource-aware data mining in a distributed environment were shown by Parthasarthy [29]. He shows how data mining algorithms could be adapted to network resource constraints.

Context-Aware Middleware: This component can inform the system by the environmental information such as the available communication channels and the effective bandwidth.

Resource Measurements: This module acts as a resource measurement receiver from both local and environmental resources.

Solution Optimizer: This module determines the data mining task scenario according to the available information about the local and environmental resources.

The module is responsible for the initiation of the data mining task and calculation of the initial parameters for the data mining technique.

Mobile Light-Weight Data Analysis Agent

Light-Weight Data Mining Agent: This module is the core of our system. It has the ability to perform the data mining task faster than data stream rate. If the device can not achieve the required accuracy according to the incoming data rate, it sends a data mining request to a data mining server. If the resource measurements indicate that the agent is not able to continue a current process with the specified accuracy, the agent can move to another device to continue this process.

Incremental Learning and Knowledge Integration: This module has the ability to update the current stored results with the incoming new data from data sources or knowledge from the server by interacting with the data mining module.

Data Stream Generator: If the mobile device generates data streams such as context information for a mobile device, or on-board sensor readings in astronomical applications [TAC02].

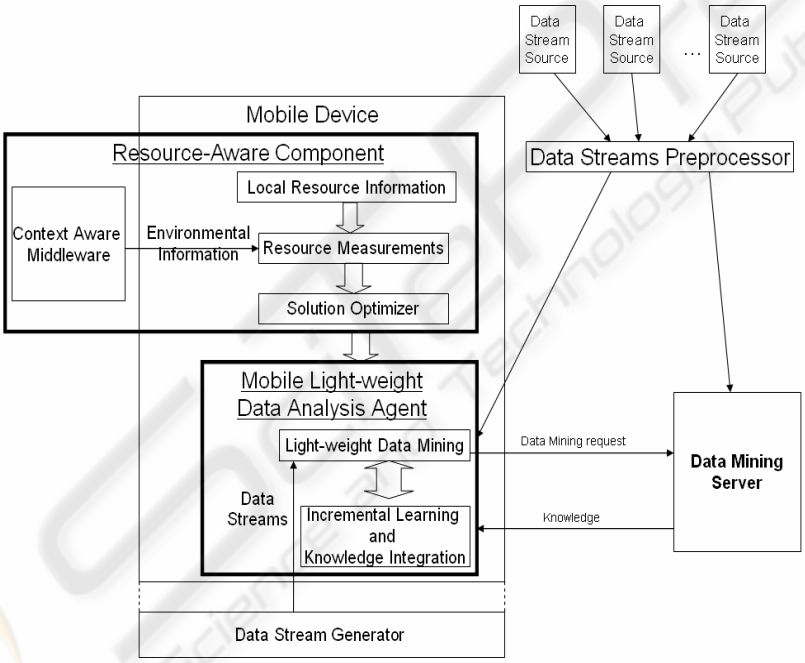

Figure 1: The Proposed System Architecture

Data Mining Server: The server can run the data mining applications upon the mobile device requests. This could be done in two cases: a) when the mobile device does not have the capability to achieve the required accuracy due to the high data rate; b) when the mobile device needs to compare between the accuracy of the light-weight algorithm and the one hosted by the server especially for the very high data rate.

Data Stream Preprocessor: can filter and aggregate the data streams sent to the mobile device or the data mining server.

At the heart of RA-UDM is the light-weight analysis tool. This tool uses AOG approach for its adaptability and being light-weight. The next section discusses the ideas behind AOG approach in mining data streams.

4 Algorithm Output Granularity

The approach uses data rate adaptation from the output side. We use algorithm output granularity to preserve the limited memory size according to the incoming data rate and the remaining time to mine the incoming stream without incremental integration. The algorithm threshold is a controlling parameter that is able to change the algorithm output rate according to the data rate, available memory, algorithm output rate history and remaining time for mining without integration.

The algorithm output granularity approach is based on the following axioms:

a) The algorithm rate (AR) is function in the data rate (DR), i.e., *AR = f(DR).* The number of generated cluster centers per unit time for example depends on the data rate.

b) The time needed to fill the available memory by the algorithm results (TM) is function in (AR), i.e., *TM = f(AR)*. The time needed for example to fill the available memory by cluster centers depends on the algorithm rate.

c) The algorithm accuracy (AC) is function in (TM), i.e., *AC = f(TM)*. That is if the time needed to fill the available memory is enough to the algorithm at the highest data rate without sampling, aggregation or algorithm granularity, this would be the best solution. The higher the algorithm granularity, the more accurate the algorithm output will be.

The controlling threshold is a parameter in each of our light-weight mining algorithm that controls the algorithm rate according to the available memory, the remaining time to fill the main memory without any incremental integration and the data rate. To demonstrate our approach in mining data streams, we define the following terms:

Algorithm threshold: is a controlling parameter built in the algorithm logic that encourages or discourages the creation of new outputs according to three factors that vary over temporal scale:

a) Available memory.
b) Remaining time to fill the available memory.
c) Data stream rate.

Output granularity: is the amount of generated results that are acceptable according to specified accuracy measure. This amount should be resident in memory before doing any incremental integration.

Time threshold: is the required time to generate the results before any incremental integration according to some accuracy measure. This time might be specified by the user or calculated adaptively based on the history of running the algorithm.

Time Frame: is the time between each two consecutive data rate measurements. This time varies from an application to another and from one mining technique to another.

The main steps for mining data streams using our proposed approach:

1) Determine the time threshold and the algorithm output granularity.
2) According to the data rate, calculate the algorithm output rate and the algorithm threshold.
3) Mine the incoming stream using the calculated algorithm threshold.
4) Adjust the threshold after a time frame to adapt with the change in the data rate using linear regression.
5) Repeat the last two steps till the algorithm lasts the time interval threshold.
6) Perform knowledge integration of the results.

After discussing the algorithm output granularity approach in mining data streams, we show the application of this approach in clustering, classification and counting frequent items.

5 Algorithm Output Granularity based Mining Techniques

In the following subsections, we show the application of the algorithm output granularity to clustering, classification and frequent items. All these algorithms have been developed and tested and proved high efficiency in running time accompanied with acceptable accuracy. Empirical results for LWC could be seen in [14][15].

5.1 LWC

In this section, our one-look clustering algorithm (LWC) is explained and discussed. The algorithm has two main components. The first one is the resource-aware RA component that uses the data adaptation techniques to catch up with the high-speed data stream and at the same time to achieve the optimum accuracy according to the available resources. The process starts by checking the minimum data rate that could be achieved using data adaptation techniques with an acceptable accuracy. If the algorithm can catch up with the minimum data rate, the RA component tries to find a solution that maximizes the accuracy by increasing the data rate. Otherwise the algorithm should send a data mining request to a data mining server that can achieve the minimum acceptable accuracy.

The other component is the LWC algorithm. The algorithm follows the following steps:

1- Data items arrive in sequence with a data rate.
2- The algorithm starts by considering the first point as a center.
3- Compare any new data item with the centers to find the distance.
4- If the distance for all the centers is greater than a threshold, the new item is considered as a new center; else increase the weight for the center that has the shortest distance between the data item and the center by 1 and let the new center equals the weighted average.
5- Repeat 3 and 4.

6- If the number of centers = k (according to the available memory) then create a new centers vector.
7- Repeat 3, 4, 5, and 6.
8- If memory is full then re-cluster (integrate clusters) and send to the server if needed.

We have performed experimental evaluation and compared our algorithm with k-means. The results showed that our algorithm outperforms k-means in running time with an acceptable accuracy [14].

5.2 LWClass

In this section, we present the application of the algorithm output granularity to light weight K-Nearest-Neighbors classification LWClass. The algorithm starts with determining the number of instances according to the available space in the main memory. When a new classified data element arrives, the algorithm searches for the nearest instance already in the main memory according to a pre-specified distance threshold. The threshold here represents the similarity measure acceptable by the algorithm to consider two or more elements as one element according to the element attributes' values. If the algorithm finds this element, it checks the class label. If the class label is the same, it increases the weight for this instance by one, otherwise it decrements the weight by one. If the weight becomes zero, this element will be released from the memory. The algorithm granularity here could be controlled by the distance threshold value and could be changing over time to cope with the high speed of the incoming data elements.

5.3 LWF

In this section, we present light-weight frequent items LWF algorithm. The algorithm starts by setting the number of frequent items that will be calculated according to the available memory. This number changes over time to cope with the high data rate. The main idea behind the algorithm is the algorithm output granularity. The AG is represented here by the number of frequent items that the algorithm can calculate as well as the number of counters that will be re-set after some time threshold to be able to cope with the continuous nature of the data stream. The algorithm receives the data elements one by one and tries to find a counter for any new item and increase the item for the registered items. If all the counters are occupied, any new item will be ignored and the counters will be decreased by one till the algorithm reaches some time threshold a number of the least frequent items will be ignored and their counters will be re-set to zero. If the new item is similar to one of the items in memory according to a similarity threshold, the average of both items will be allocated and the counter will be increased by one. The main parameters that can affect the algorithm accuracy are time threshold, number of calculated frequent items and number of items that will be ignored and their counter will be re-set after some time threshold.

6 Conclusions and Future Work

We have described a wireless data stream mining model. The model has two novel contributions that have the advantage of generality and adaptability. The ubiquitous data stream mining architecture is the first contribution. The architecture has the advantage of being applicable to any application.

The algorithm output granularity is the second contribution. AOG is an adaptable data stream mining approach. The implementation of this architecture with the adoption of AOG in mining data streams is our ultimate goal in this project. The potential applications of the system vary from astronomical, web, and business applications.

References

1. C. Aggarwal, J. Han, J. Wang, P. S. Yu, “A Framework for Clustering Evolving Data Streams”, Proc. 2003 Int. Conf. on Very Large Data Bases (VLDB'03), Berlin, Germany, Sept. (2003).
2. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In Proceedings of PODS, (2002).
3. B. Babcock, M. Datar, R. Motwani, L. O'Callaghan: Maintaining Variance and k-Medians over Data Stream Windows, *to appear* in Proceedings of the 22nd Symposium on Principles of Database Systems (PODS 2003).
4. R. Bhargava, H. Kargupta, and M. Powers: Energy Consumption in Data Analysis for On-board and Distributed Applications. Proceedings of the ICML'03 work
   shop on Machine Learning Technologies for Autonomous Space Applications, (2003).
5. M. Burl, Ch. Fowlkes, J. Roden, A. Stechert, and S. Mukhtar, "*Diamond Eye: A distributed architecture for image data mining,*" in SPIE DMKD, Orlando, April (1999).
6. M. Charikar, L. O'Callaghan, and R. Panigrahy. Better streaming algorithms for clustering problems In Proc. of 35th ACM Symposium on Theory of Computing (STOC), (2003).
7. L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. Streaming-data algorithms for high-quality clustering. Proceedings of IEEE International Conference on Data Engineering, March (2002).
8. Graham Cormode, S. Muthukrishnan What's hot and what's not: tracking most frequent items dynamically. PODS 2003: 296-306
9. Mayur Datar, Aristides Gionis, Piotr Indyk, Rajeev Motwani: Maintaining Stream Statistics Over Sliding Windows (Extended Abstract) in Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002).
10. P. Domingos and G. Hulten, "A General Method for Scaling Up Machine Learning Algorithms and its Application to Clustering", Proceedings of the Eight eenth International Conference on Machine Learning, 2001, 106--113, Williams town, MA, Morgan Kaufmann. (2001)
11. P. Domingos and G. Hulten. Mining High-Speed Data Streams. In Proceedings

of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining, pages 71--80, (2000).

12. V. Ganti, Johannes Gehrke, Raghu Ramakrishnan: Mining Data Streams under Block Evolution. SIGKDD Explorations 3(2): 1-10 (2002).
13. C. Giannella, J. Han, J. Pei, X. Yan, and P.S. Yu, "Mining Frequent Patterns in Data Streams at Multiple Time Granularities", in H. Kargupta, A. Joshi, K. Siva kumar, and Y. Yesha (eds.), Next Generation Data Mining, AAAI/MIT, (2003).
14. Gaber, M, M., Krishnaswamy, S., and Zaslavsky, A., Adaptive Mining Techniques for Data Streams Using Algorithm Output Granularity, Proc. of The Australasian Data Mining Workshop (AusDM 2003), Held in conjunction with the 2003 Congress on Evolutionary Computation (CEC 2003), December, Canberra, Australia, Springer Verlag, Lecture Notes in Computer Science (LNCS). (2003)
15. Gaber, M.M., Krishnaswamy, S. and Zaslavsky, A. (2004). Cost-Efficient Mining Techniques for Data Streams. In Proc. Australasian Workshop on Data Mining and Web Intelligence (DMWI2004), Dunedin, New Zealand. CRPIT, 32. Purvis, M., Ed. ACS. (2004)
16. S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In Proceedings of the Annual Symposium on Foundations of Computer Sci ence. IEEE, November (2000).
17. L. Golab and M. Tamer Ozsu. *Issues in Data Stream Management*. In SIGMOD Record, Volume 32, Number 2, June 2003, pp. 5--14.
18. M. Henzinger, P. Raghavan and S. Rajagopalan, Computing on data streams, Technical Note 1998-011, Digital Systems Research Center, Palo Alto, CA, May (1998).
19. G. Hulten, L. Spencer, and P. Domingos. Mining Time-Changing Data Streams. ACM SIGKDD (2001).
20. H. Kargupta. CAREER: Ubiquitous Distributed Knowledge Discovery from Heterogeneous Data. NSF Information and Data Management (IDM) Workshop (2001).
21. H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, M. Klein, K. Sarkar and D. Handy: Vehicle Data Stream Mining (VEDAS): An Experimental Sys tem for Mobile and Distributed Data Stream Mining. Information Mining for Automotive and Transportation Domain workshop. Madrid, Spain (2003).
22. E. Keogh, J. Lin, and W. Truppel. Clustering of Time Series Subsequences is Meaningless: Implications for Past and Future Research. In *proceedings of the 3rd IEEE International Conference on Data Mining.* Melbourne, FL. (2003).
23. Kargupta, H., Park, B., Pittie, S., Liu, L., Kushraj, D. and Sarkar, K. MobiMine: Monitoring the Stock Market from a PDA. ACM SIGKDD Explorations. January (2002). Volume 3, Issue 2. Pages 37--46. ACM Press.
24. G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In Proceedings of the 28th International Conference on Very Large data Bases, Hong Kong, China, August (2002).

25. S. Muthukrishnan , Data streams: algorithms and applications. Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms. (2003)
26. S. Muthukrishnan, Seminar on Processing Massive Data Sets. Available Online:
http://athos.rutgers.edu/%7Emuthu/stream-seminar.html, (2003).
27. Carlos Ordonez. Clustering Binary Data Streams with K-means .*ACM DMKD (2003).*
28. B. Park and H. Kargupta. Distributed Data Mining: Algorithms, Systems, and Applications. To be published in the Data Mining Handbook. Editor: Nong Ye. (2002).
29. S. Parthasarathy: Towards Network-Aware Data Mining. In International Workshop on Parallel and Distributed Data Mining, along with IPDPS (2001).
30. S. Papadimitriou, C. Faloutsos, and A. Brockwell, "Adaptive, Hands-Off Stream Mining", 29th International Conference on Very Large Data Bases VLDB, (2003).