

A Survey on Knowledge Extraction from WSN

Priyanka Pandit¹, Swarupa Kamble²

¹Dept. of Computer Engineering, RMD Sinhgad School of Engineering, Pune, Maharashtra, India

²Assistant Professor, Dept. of Computer Engineering, RMD Sinhgad School of Engineering, Pune, Maharashtra, India

Abstract: Recently, data management and processing for wireless sensor networks are become a topic of active research in several fields of computer science such as distributed system, database systems and data mining. WSN generates large amount of data streams. Extracting information from the data is the most important task in WSN. Many methods are available to extract the information from WSN. These methods generate the association rules using frequency patterns. It creates large number of rules most of which are non informative and fail to reflect true correlation among the sensor data. This paper presents new type of behavioral pattern called as an associated sensor pattern. Associated sensor pattern captures the co-occurrences and temporal correlation of the sensor data. To capture such patterns a compact tree structure, called associated sensor pattern tree (ASP-tree) and mining algorithm are proposed. When data streams flows through the network there is possibility of loss of significance for the current time of the old information. ASP-tree is further enhanced to SWASP-tree to capture significance of recent data by adopting sliding observation window and updating tree structure accordingly.

Keywords: Wireless sensor networks, data mining, knowledge discovery, behavioral patterns, sensor data stream.

1. Introduction

A WSN includes of a large number of heterogeneous or homogeneous nodes usually called as sensor nodes, they communicates via wireless media to the concentrator node called as sink node and works cooperatively to monitor the environment. Wireless sensors are integrated as an ad-hoc fashion to initiate a network, that network is able to deliver detected event in multi-hop transmission to sink periodically. In this mode of transmission networks network creates large amount of data which is in the form of stream. There are data mining techniques which have recently received a great attention to extract intersecting knowledge from these stream data. There are three challenges mining techniques need to be address such as the stream nature of the data, limited resources and distributed nature of sensor.

Data mining techniques have shown to be a promising tool to improve WSN performance and quality of services (QOS). Knowledge discovery in WSN has been used to extract information about the surrounding environment, that are deduce from the data reported by sensor nodes and behavioral patterns about sensor nodes, which are evolved from meta-data describing sensor's behaviors.

Discovering behavioral patterns (i.e., associated patterns) from WSNs can be highly useful in applications that require a fine-grain monitoring of physical environments (e.g., buildings, transportation networks, and battlefield) which may face critical situations like fire, toxic gas leaks and explosion. Behavioral patterns can also be used to predict the source of future events. Knowing the source of a future event may lead to detect faulty nodes, if any, in the network. For example, behavioral pattern mining predicts occurrence of an event from a particular node, but no such event is subsequently reported; this indicates possibility of a node failure. It can also identify the source of the next event when behavioral patterns reveal a chain of related events, e.g., fault in a particular process in an industry may trigger fault in other processes. Behavioral patterns can also reveal a set of temporally correlated sensors. Such patterns can improve

operational aspects (e.g., identifying missed reading) and better management of resources (e.g., sleep awake schedule of nodes) in a WSN.

The main challenges in mining associated sensor patterns are:

- 1) Appropriate formulation to discover associated sensor patterns maintaining the downward closure property to ensure search space reduction;
- 2) Devising a compact tree structure that is able to capture the data content in one scan over sensor datasets and yield better mining performance;
- 3) As older information in the data stream may become unimportant or obsolete for the current time period, it is necessary to make the tree structure adaptive so that it effectively captures the latest information, makes optimal use of memory and extracts complete set of recent associated sensor patterns. To facilitate such tree construction, appropriate data observation window size needs to be determined.

ASP-Tree

The main concept behind our ASP-tree construction is to first build a prefix-tree based on the order of sensors' appearance in the database, then restructure the tree in a frequency descending order, and finally compress the tree by merging the same support sensor nodes in a single node in each branch of the tree. After that, we use a pattern growth approach to mine the associated sensor patterns from our ASP-tree. The ASP-tree construction has two phases: insertion phase and restructuring-compression phase.

For the insertion phase, the ASP-tree arranges the sensors according to sensors' appearance order in the database and is built by inserting every epoch in the database one after another into it.

The purpose of the restructuring-compression phase is to achieve a highly compact ASP-tree which will utilize less memory and facilitate fast mining process. In this phase, we

first sort the SO in frequency-descending order using merge sort and reorganize the tree structure.

Though the ASP-tree and CP-tree bear some similarities in tree construction phase, but the following differences exist:

- 1) ASP-tree performs additional compression that merges the same support sensors in a single node and thereby makes it compact. This ensures that ASP-tree holds fewer nodes than CP-tree.
- 2) Consequently, memory footprint of ASP-tree will be far less than CP-tree.
- 3) CP-tree utilizes FP-growth based mining method to generate frequent patterns.

SWASP-Tree

In sliding window based sensor data stream mining, a window consists of multiple non-overlapping batches and each batch contains a non-empty set of epochs. An example of sensor data stream is shown in Fig. 1, where the data stream of eight epochs are divided into four batches and three batches constitute one window. The SWASP-tree, as described below, captures important knowledge from the stream content for the current window where each node in the tree maintains sensor-id and batch-by-batch support information.

The SWASP-tree construction proceeds window by window. Similar to the ASP-tree, the SWASP-tree construction has two phases: insertion phase and restructuring-compression phase. Additionally, we employ an update phase to reconstruct the tree constructed in the previous window as the window advances.

SWASP-tree differs from DSTree significantly as the latter does not have restructure compression phase and does not perform frequency list update operation for the nodes which are not visited during the new incoming batch, leaving some redundant nodes in the tree structure. This may cause a DSTree to become overloaded with nodes that are no longer relevant for the current window. SWASP-tree structure is also quite different than that of CPS-tree. CPS tree construction is based on a predefined item order (e.g., lexicographic item order) and defines two types of nodes (ordinary and tail) while in SWASP-tree all nodes are treated equally. Moreover, it does not perform compression. The restructuring and compression in SWASP-tree guarantee that SWASP-tree will have less memory requirement than DSTree and CPS-tree.

2. Overview

Proposed a new type of behavioral pattern called as an associated sensor pattern which are linked with such co-occurrences. Associated sensor pattern captures the co-occurrences and temporal correlation of the sensor data. To capture such patterns a compact tree structure, called associated sensor pattern tree (ASP-tree) and mining algorithm are proposed. When data streams flows through the network there is possibility of loss of significance for the current time of the old information. ASP -tree is further enhanced to SWASP-tree to capture significance of recent data by adopting sliding observation window and updating tree structure accordingly. We are also improving the life of sensor nodes by clustering the nodes.

3. Existing Methodology

A wireless sensor network (WSN) (sometimes called a wireless sensor and actor network (WSAN)) are spatially distributed autonomous sensors to monitor physical or environmental conditions, such as as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity.

3.1 Frequent Pattern Mining Using a Prefix-Tree

In this paper, we propose a comprehensive framework for mining Wireless Ad Hoc Sensor Networks (WASNs), which is able to extract patterns regarding the sensors' behaviors. The main goal of determining behavioral patterns is to use them to generate rules that will improve the WASN's Quality of Service by participating in the resource management process or compensating for the undesired side effects of wireless communication. The proposed framework consists of 1) a formal definition of sensor behavioral patterns and sensor association rules, 2) a novel representation structure that we refer to as the Positional Lexicographic Tree (PLT) that is able to compress the data gathered for the mining process and thus allows the fast and efficient mining of sensor behavioral patterns, and 3) a distributed data extraction mechanism to prepare the data required for mining sensor behavioral patterns. Several experimental studies have been conducted to evaluate our PLT structure and our proposed data extraction algorithms for mining wireless sensor networks.

3.2 Behavioral Patterns For Point of Coverage

Recently, Knowledge Discovery Process has proven to be a promising tool for extracting the behavioral patterns of sensor nodes, from wireless sensor networks. In this paper, we propose a new kind of behavioral pattern, named Target-based Association Rules (TARs). TARs aim to discover the correlation among a set of targets monitored by a wireless sensor network at a border area. The major application of the Target-based Rules is to predict the location (target) of a missed reported event. Different data preparation mechanisms for accumulating the data needed for extracting TARs have been proposed. We refer to these mechanisms as

AI-Node, Schedule-Buffer, and Fused-Schedule-Buffer. Several experiment studies have been conducted to evaluate the performance of the three proposed data preparation mechanisms. Results show that the Fused-Schedule-Buffer scheme outperforms the selected schemes in terms of energy consumption.

3.3 Efficient single-pass frequent pattern mining

The FP-growth algorithm using the FP-tree has been widely studied for frequent pattern mining because it can dramatically improve performance compared to the candidate generation-and-test paradigm of Apriori. However,

it still requires two database scans, which are not consistent with efficient data stream processing. In this paper, we present a novel tree structure, called CP-tree (compact pattern tree), that captures database information with one scan (insertion phase) and provides the same mining performance as the FP-growth method (restructuring phase). The CP-tree introduces the concept of dynamic tree restructuring to produce a highly compact frequency-descending tree structure at runtime. An efficient tree restructuring method, called the branch sorting method, that restructures a prefix-tree branch-by-branch, is also proposed in this paper. Moreover, the CP-tree provides full functionality for interactive and incremental mining. Extensive experimental results show that the CP-tree is efficient for frequent pattern mining, interactive, and incremental mining with a single database scan.

4. Conceptual Framework

We propose a new type of behavioral pattern called associated sensor patterns which capture association-like co-occurrences as well as temporal correlations which are linked with such co-occurrences. To capture such patterns a compact tree structure, called associated sensor pattern tree (ASP-tree) and a mining algorithm (ASP) are proposed which use pattern growth-based approach to generate all associated patterns with only one scan over dataset. Moreover, when data stream flows through, old information may lose significance for the current time. To capture significance of recent data, ASP-tree is further enhanced to SWASP-tree by adopting sliding observation window and updating the tree structure accordingly. Finally, window size is made dynamically adaptive to ensure efficient resource usage. We are also improving the life of sensor nodes by clustering the nodes.

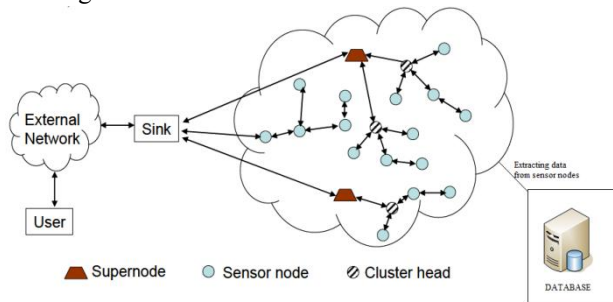


Figure 1: System Architecture

4.1 ASP Tree

The main concept behind our ASP-tree construction is to first build a prefix-tree based on the order of sensors' appearance in the database, then restructure the tree in a frequency descending order, and finally compress the tree by merging the same support sensor nodes in a single node in each branch of the tree. After that, we use a pattern growth approach to mine the associated sensor patterns from our ASP-tree. The ASP-tree construction has two phases: insertion phase and restructuring-compression phase.

For the insertion phase, the ASP-tree arranges the sensors according to sensors' appearance order in the database and is

built by inserting every epoch in the database one after another into it.

The purpose of the restructuring-compression phase is to achieve a highly compact ASP-tree which will utilize less memory and facilitate fast mining process. In this phase, we first sort the SO in frequency-descending order using merge sort and reorganize the tree structure.

Though the ASP-tree and CP-tree bear some similarities in tree construction phase, but the following differences exist:

- 1) ASP-tree performs additional compression that merges the same support sensors in a single node and thereby makes it compact. This ensures that ASP-tree holds fewer nodes than CP-tree.
- 2) Consequently, memory footprint of ASP-tree will be far less than CP-tree.
- 3) CP-tree utilizes FP-growth based mining method to generate frequent patterns.

4.2 SWASP Tree

In sliding window based sensor data stream mining, a window consists of multiple non-overlapping batches and each batch contains a non-empty set of epochs. An example of sensor data stream is shown in Fig. 1, where the data stream of eight epochs are divided into four batches and three batches constitute one window. The SWASP-tree, as described below, captures important knowledge from the stream content for the current window where each node in the tree maintains sensor-id and batch-by-batch support information.

The SWASP-tree construction proceeds window by window. Similar to the ASP-tree, the SWASP-tree construction has two phases: insertion phase and restructuring-compression phase. Additionally, we employ an update phase to reconstruct the tree constructed in the previous window as the window advances.

SWASP-tree differs from DSTree significantly as the latter does not have restructure compression phase and does not perform frequency list update operation for the nodes which are not visited during the new incoming batch, leaving some redundant nodes in the tree structure. This may cause a DSTree to become overloaded with nodes that are no longer relevant for the current window. SWASP-tree structure is also quite different than that of CPS-tree. CPS tree construction is based on a predefined item order (e.g., lexicographic item order) and defines two types of nodes (ordinary and tail) while in SWASP-tree all nodes are treated equally. Moreover, it does not perform compression. The restructuring and compression in SWASP-tree guarantee that SWASP-tree will have less memory requirement than DSTree and CPS-tree.

5. Conclusion

In this project, we have introduced a new type of behavioral patterns called associated sensor patterns that capture co-occurrences as well as temporal correlations among sensors. To extract such patterns, we devised a prefix tree structure

called ASP-tree that stores sensor data in a compact manner, and based upon this tree a mining algorithm called ASP is proposed which effectively mines associated sensor patterns over sensor database in only one scan. Since it is not practical to maintain all elements of sensor data in a tree when data are captured and analyzed for a stream flow, we presented an enhanced tree structure called SWASP-tree which can efficiently discover associated sensor patterns from stream data in a single-pass using sliding observation window. Both trees have the build once and mine many property, making it highly suitable for interactive mining. To optimize resource usage, a technique to dynamically adapt window size is also presented. Comparative performance analyses show that our techniques are very effective and efficient for mining associated sensor patterns over sensor data and outperform existing algorithms in both runtime and memory usage, and also scale well to handle large number of distinct sensors and epochs.

References

- [1] Boukerche D and S. Samarah, "Novel algorithm for mining association rules in wireless ad-hoc sensor networks," IEEE Trans. Parallel Distributed. Syst., vol. 19, no. 7, pp. 865–877, Jul. 2008.
- [2] S. Samarah, B. Azzedine, and S. H. Alexander, "Target association rules: A new behavioral patterns for point of coverage wireless sensor networks," IEEE Trans. Computer., vol. 60, no. 6, pp. 879–889, Jun. 2011
- [3] S. K. Tanbeer, C. F. Ahmed, and B. S. Jeong, "Efficient single-pass frequent pattern mining using a prefix-tree," Inf. Sci., vol. 179, pp. 559–583, 2009.
- [4] L. Gruenwald and M. Halatchev, "Estimating missing values in related sensor data streams," in Proc. 11th Int. Conf. Manage. Data, 2005, pp. 981–987.
- [5] K. K. Loo, I. Tong and B. Kao, "Online algorithms for mining interstream associations from large sensor networks," in Proc. 9th Pacific- Asia Conf. Adv. Knowledge. Discovery data Mining, 2005, pp. 143–149.
- [6] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," Proc. 28th Int. Conf. Very Large Data Bases, 2002, pp. 346–357.
- [7] K. Romer, "Distributed mining of spatial-temporal event patterns in sensor networks," in Proc. Eur.-Amer. Workshop Middleware Sensor Networking. Conjunction with Int'l Conf. Distributed. Computer. Sensor Syst., 2006, pp. 103–116.
- [8] Y. Jennifer, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer. Networking. vol. 52, pp. 2292–2330, 2008.

Author Profile



Priyanka Pandit received B.E. degree in Computer Science in 2012 from Savitribai Phule Pune University and pursuing M.E. from RMDSSOE, Warje, Pune.



Swarupa Kamble is working with RMDSSOE, Warje, Pune as an Assistant Professor. She has experience of 5 yrs in the field of teaching and research and her research interests are Image Processing and Data Mining.