Event Correlations in Sensor Networks

Ping Ni, Li Wan and Yang Cai

CIC Building, 4720 Forbes Ave Pittsburgh, Carnegie Mellon University

{pingni,liwan,ycai}@andrew.cmu.edu

Abstract. In this paper we present a novel method to mine the correlations of events in sensor networks to extract correlation patterns of sensors' behaviors by using an unsupervised algorithm based on a hash table. The goal is to discover anomalous events in a large sensor network where its structure is unknown. Our algorithm enables users to select the correlation confidence level and only display the significant event correlations. Our experiment results show that it can discover significant event correlations in both continuous and discrete signals from heterogeneous sensor networks. The applications include smart building design and large network data mining.

Keywords: Correlation; Visualization; Sensor Network

1 Introduction

Discovering salient correlations between events in a large sensor network is valuable for reducing the number of sensors, unnecessary communication and energy consumption. For example, if we know the relations between temperature and humidity we can predict humidity through temperature. Sensors corresponding to humidity can then be removed from a smart building.

Kun-Chent [7] proposes a smart control algorithm to naturally adjust the thermal quality of the environment according to the interior and exterior environmental factors and the behavior of the inhabitants. They analyze correlations between sensors that are known in advance. In Kay's system [10] a user can pose a query to the system using a declarative language. Such a query defines the local events of interest and additional constraints on the sought for frequently occurring event patterns. In this system the frequent patterns are discovered only through some sensors so the calculation resources are constrained. What's more, it only explored the specific patterns of the user's query and did not display relations between sensors automatically. It can't predict events which will emerge in following time interval. Kay [14] focuses on embedded system pattern discovery which characterize the spatial and temporal correlations between events. It only defined the support parameter which is really not enough to discover correlations between events.

Here we address correlations in sensor networks by using an event-driven model for improving efficiency and effectiveness. We extract valuable patterns which are representative patterns of closed patterns [17] instead of complete patterns. Our contributions include: 1) discovering the significant patterns without necessary user specified support definitions, 2) prioritizing event correlations instead of complete correlations, and 3) visualizing the key event correlations in a network diagram.

2 Related work

Mining for correlations [5] is recognized as an extremely important data mining task for its many advantages over data mining using association rules. Instead of discovering co-occurrence patterns in data as does association rule mining, correlation mining identifies the underlying dependency from the data set itself. Those infrequent but significant patterns that are too expensive to be revealed by association rule mining can now be discovered using correlation mining techniques. Zhang [3] discussed how to find two correlation sets. It can extract correlation between multiple series through a variable time window. But obviously it can’t find a dynamic correlation. For example we might know that sometimes the motion of people can lead to the light being turned on. If in a short time there is a high dB sound present, the light can be also turned on. This is a burst event. Zhang’s approach [3] couldn’t find the correlations because there is a very short time correlation between the acoustic event and light being turned on. Indeed, acoustics and light have strong correlation too. Mattew [1] addresses the problem of online detection of unanticipated modes of mechanical failure given a small set of time series under normal conditions, with the requirement that the anomaly detection model be manually verifiable and modifiable. Ke [4] described the relations between the association rule and correlation. The correlation can be obtained when the parameter “support” is ignored by the defined association rule. It will introduce complex time questions and dramatically increases the memory demand if there are lots of items in data set when ignoring the support parameter. Sometimes the value of each attribute is not only of a Boolean type.

How to get exact patterns among attributes of a non-Boolean type presents a scientific challenge. Edith Cohen [6] found interesting associations without support pruning efficiently by using a compressing transformation to get an estimation matrix and then verified the validation of the transformation. Indeed it can find correlations between two columns which hold sparse data efficiently. But it is only for Boolean-type and only for correlations of two columns. With real-world data we need to extract relations between multiple columns like our sensor data set.

2. Problem Definition and Algorithm

In this study, there are over 200 sensors; here we only choose 6 of them . They are acoustic, light, motion, temperature, CO2, humidity. The sensors’ names are abbreviated by their respective first initials (Acoustic=a, Light=l, Motion=m, Temp=t, Co2=c and Humidity=h).

a. Decreasing memory usage

We in fact needn't load the entire data set into the memory because the data set contains sparse data. We will maintain as static the rows which have concurrence events. Since we know that it is a very sparse data set with strong relations among the sensor data set, the emerging event states should have some relations that should be revealed according to common sense. The entire data space could also be compressed using common sense. We use the following data structure to compress our entire data set in a hash table.

Table 1. Hashtable structure

Key (event string)	Value (frequency)
Key1	F1
Key2	F2

From Table 1 we see that the same event string can be assigned the same key and their frequency would be incremented. The whole data set can be compressed like a FP-Growth [16] algorithm in which it compresses data set through FP-tree structure. This is verified using statistics from our entire data sets. Thirty nine patterns including various events took place in 83979 records of six sensors in two months from a real data set. Thirty nine patterns would be loaded into the memory for confidence calculation according to the event sets relation. The satisfied confidence between emerging events set is stored. So it needs only few memory spaces to hold all valuable patterns and we can work only on the compressed data set in hash table.

b. Problem Definition

Here we assumed we have events sets $E=\{E_1,E_2,...E_k...,E_n\}$, E is the event sets of this data set. E_k represent the k^{th} events set in E. The events are stored in a hash table as stated earlier. Any pairs in *E* are not equal. $E_k = \{e_{k1}, e_{k2}, ...e_{ki}..., e_{kj}\}$, e_{ki} represents if *i* takes place e_{ki} would be set corresponding discrete value.

Here we assume $E_1 = \{a\}, E_2 = \{a,c\}, E_3 = \{c,t\}, E_4 = \{l,m\}$. For pair events we will have followed relations.

- **Full-Contained-relation:** all emerging events in an event set are all contained in another event set. For example $E_1 \subset E_2$, $\{a\}$ is contained in $\{a,c\}$.
- **Disjoint relations:** any emerging event in an event set is not contained in another event set. For example E_2 and E_4 ..
- **Part-containing-relation:** At least one but not all events which took place in one set are contained in another event set which hold at least one different event with previous event set. For example E_2 and E_3 .

Here we will illustrate relations between valuable pattern sets and complete pattern sets. Traditionally we would explore whole complete combinations among the events space like $E_1 \rightarrow E_i E_k, E_1 \rightarrow E_i E_k E_j, E_1 \rightarrow E_i$. $E_i E_k$ means the union of the events. Here we examine our questions from $E_1 \rightarrow E_i E_k$ and judge if $E_i E_k$ is valuable. Here it would have followed cases. The following cases are described.

1. $E_i E_k$ is contained in E, it means that there is an event set equal with $E_i E_k$, we assume $E_l = E_i E_k$. For $E_1 \rightarrow E_i E_k$ we can get from $E_1 \rightarrow E_l$ instead of $E_i E_k$.

2. $E_i E_k$ is not contained in E, and not the subset of any element in E. so the $| E_i E_k |$ should be zero. So the $E_1 \rightarrow E_i E_k$ is zero.

3. $E_i E_k$ is not contained in E, but it is the subset of one or more elements in E. for $E_1 \rightarrow E_i E_k$ is not valuable than $E_1 \rightarrow \sup erset(E_i E_k)$. Because $E_i E_k$ is the subset of some event set (assumed E_a) in *E*. So we can get valuable patterns from $E_1 \rightarrow E_a$

For $E_1 \rightarrow E_i ... E_k ... E_j$ we can get similar reasoning recursively.

Obtain the entire valuable pattern set we only explore the correlation of the follows matrix.

$$\begin{array}{c|cccc|} & E_1 & E_2 & ..... & E_n \\ E_1 & & E_1 \rightarrow E_2 & & \\ E_2 & E_2 \rightarrow E_1 & & & \\ ... & & & & \\ E_n & & E_n \rightarrow E_2 & & \end{array} \qquad (1)$$

For each pair of event sets in matrix (1) we will define an operator between them as follows.

1. Full-Contained-relation Confidence: like $\{a\}$ and $\{a, c\}$ we would get Confidence

$$Confidence(\{a\} \rightarrow \{a, c\}) = \frac{| ac |}{| a |} \qquad (2)$$

2. Part-containing-relation Confidence: like {ac}->{ct}, so we get the confidence

$$Confidence(\{a, c\} \rightarrow \{c, t\}) = \frac{| c \rightarrow a |}{| c |} \qquad (3)$$

3. Disjoint relation Confidence: like {c,t} and {m,l}, we get the confidence

$$Confidence(\{c, t\} \rightarrow \{m, l\}) = \frac{| ctml |}{| ct |} \qquad (4)$$

After finishing the definitions of each relation, we give our prune definition for decreasing time cost.

We propose that if events $\{a,c\}$ has weak correlation with events set $\{c,t\}$, then $\{a,c\}$ will have weak correlation with all superset of events $\{a,c,t\}$ and prove it as follows:

If events $\{a,c\}$ occurs $k1$ times in whole data set, $\{c,t\}$ occurs $k2$ times. According to confidence definition of part containing relation,

$\frac{|ac \to ct|}{|ac|} = \frac{|act|}{|ac|} = \frac{k1}{k2} < \varepsilon$, ε represents the minimum confidence. Any superset of $\{a,c,t\}$ event would take place $k3$ times which are less than $k1$. So the confidence between $\{a,c\}$ and superset of $\{a,c,t\}$ would be less than ε .

Events which can't meet the minimum confidence will lead to pruning of their superset. We can summarize our algorithm below as algorithm 1 which incorporates a data preprocess algorithm from [11].

```
Algorithm 1:
Input: min_confidence, cycle, time_slot, support.
//cycle, time_slot, support is referred from [11]
 Output: correlation between sensors.
    data preprocess
    Extract all co-occur events in the data set and add to Hashtable
    which key is the event set, and value is the emergence frequencies
    of combination events.
    For each elements in hash table{
    For each elements in hash table{
    Judge events relations according to above relation definition, we
    acquire events set A.and B.
    Getconfidence(A,B).
    If( $Confidence(A \to B) \le \min\_confidence$ )
    According to pruning rule, the superset of $A \cup B$ would be pruned.
```

3 Performance study

In this section, we evaluate the performance of the proposed algorithm and report the results that we have obtained using a real-world data set.

a. Experimental Results

First, we evaluate our algorithm 1. In algorithm 1 we set the cycle=5 and time_slot=5 minutes and support=0.1, support =0.3, support =0.5, support =0.9, support =1 respectively. Let X axis represents the patterns turn. Y axis represents the confidence between two event sets. See Figure 1.

From the comparison of figure in Fig.1, we can easily find that our sensors indeed have strong correlations as the support increases. When support=0.9 and 1 the relations are very obvious between sensors. From figure 1.e and 1.f we find that our sensors have strong correlations and have weak correlation. They have obvious group features between sensors but figure 1.a and figure 1.b are not obvious in displaying group features because support is too low. We can get effective results through our algorithm 1. From figure 1.e and 1.f we can find clustered information in which sensors can be considered as a strong correlation group. We cannot get the relation between t and h in either an *apriori* algorithm or FP-Growth algorithm when we set support=0.1.

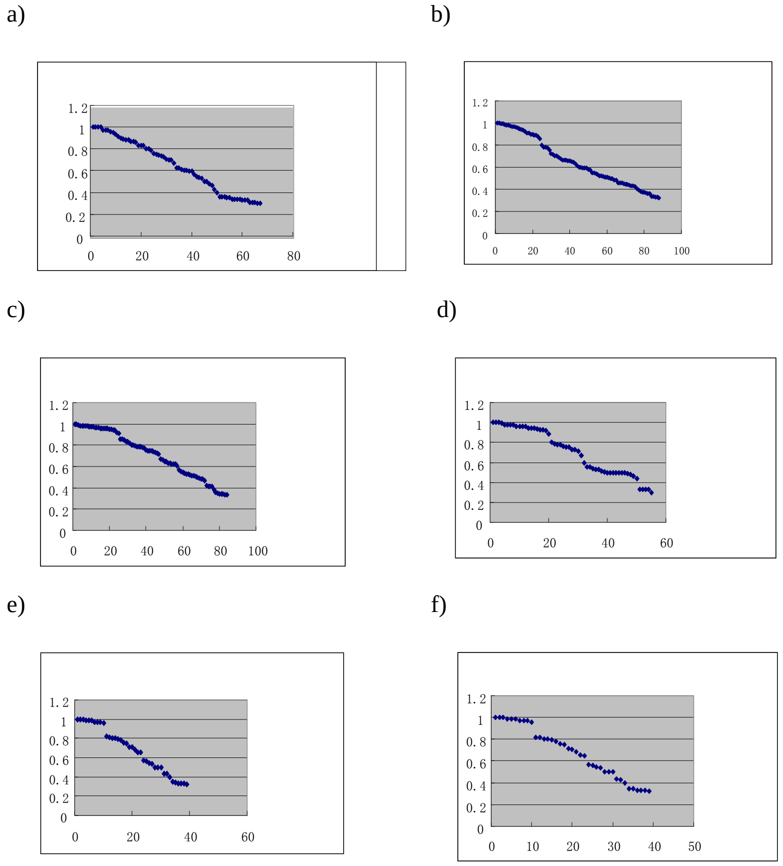

Fig. 1. a) support=0, b) support=0.1, c) support=0.3, d) support=0.5, e) support=0.9, f)support=1

The performance of our algorithm is tested using a laptop in which the CPU processor is 1.50 GHZ, memory is 760M. We split our real dataset into 6 parts (A,2),(B,5),(C,8),(D,10),(E,13), (X,Y) represents size of data set X is Y megabyte. Figure 10 gives the running time of algorithm 1 when confidence =0.1, 0.3, 0.7. As the confidence threshold increases and data size increases, our pruning effect is much more obvious. Our algorithm performs almost linear in time except for the memory limitation.

We use algorithm 1 to exploit a public data set [15]. First we extracted this public data set based on table 2. We list discovered key patterns that are described in table 3.

We can find that variance has strong relations between each other in the same sensor. And we also get a stronger correlation than others between t and v which has been declared by [15].

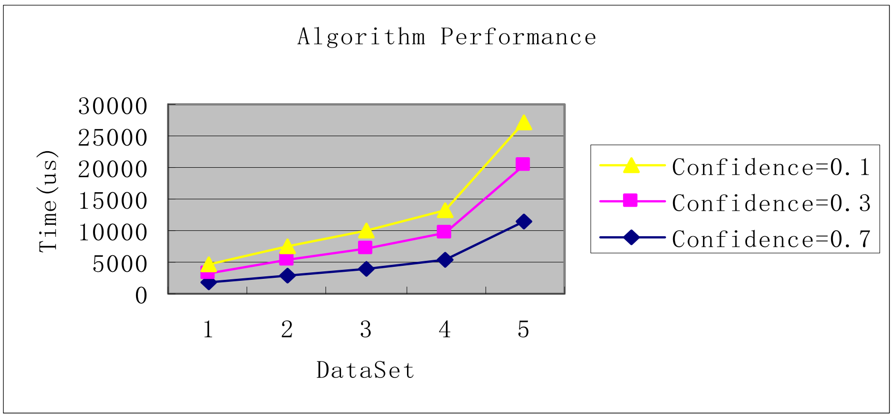

Fig. 2. Performance comparing

Table 2. MIT Data discrete standard

Sensor Name	**Definition**
temperature	Temp rise 10%, value is set 1, versa is 2, no change is 0.
humidity	Humidity rise 3% value is set 1, versa is 2, no change is 0.
Light	lux of light rise 10%, value is set 1, versa is 2, no change is 0.
voltage	Voltage rise 2% value is set 1, versa is 2, no change is 0.

Table 3. Correlation Patterns *Slot_num=5, c=5, support=1*

Patterns	Confidence	Discovered Patterns
t-v	0.76	t->h,v
t,v-h	0.66	v->t
t-h,v	0.51	l,v->t,h
h,v-t	0.36	t->v
l,v-t	0.33	v->h
l,v-h	0.33	
l,v-t,h	0.33	

4 Visualization of Event Correlations

The correlated sensor events can be visualized as a tree shape. Here we set our support parameter 1, confidence is 0.3, cycle is 5 and *time_slot=8*. We get the whole event set state diagram. From figure 3we found some noise existing in it according to common sense. Even though we employed a noise cleaning algorithm from [11] like $l, c \rightarrow m$, c is obviously an occasion event according to common sense since CO2 is a colorless gas which has no effect upon either lighting or motion sensors. We present algortihm2 to compress our patterns. We get our concise state diagram of Figure 4 through algorithm2 as follows.

```
  Algorithm 2 Pattern Compress
  Input: Correlated event sets
  Output: Concised correlated patterns
    Find the nonzero in-degree node from state diagram.
    int patternCount[i];
    //table is two dimension array including each pattern in every rows
    //from state diagram. Like row 1, m,1. Row 2, m,l,h etc, the bit
    //will be set 1 in corresponding place.

    For each col i in table {    for each row j in table{
           patternCount[i] += table.get(j,i);

    }
    if(patternCount[i]/allRows < concise_parameter)
            remove column i from table;
    }
    Return table;
```

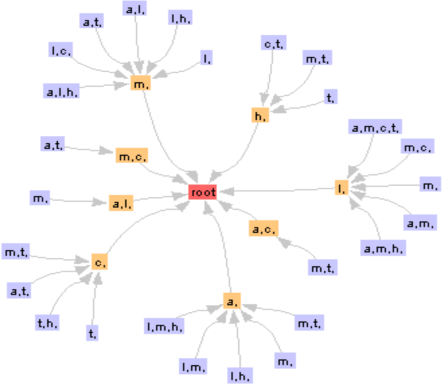

Fig. 3. Event Correlation diagram between sensors

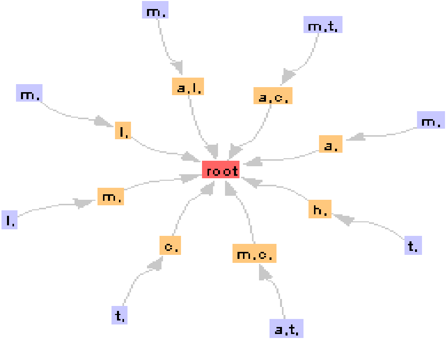

Fig. 4. Trimmed Event Correlation diagram with *concise_parameter=0.7*

Compared the tree structures between the images in Fig. 3 and 4, the state relation is suppressed and present more meaning relations in our sensor network according to commonsense.

5 Conclusions

We here present a novel method to extract correlations from a large number of sensors instead of using a traditional method based on an *apriori* algorithm and pattern growth[16] method. Our method is event-driven and discovers specific valuable patterns instead of a complete pattern set. We incoporate the algorithm from [11] within algorithm 1 in sensor networks to improve efficiency for discovering concise and accurately correlated patterns. We reclean noise from patterns and show concise and meaningful patterns through state diagram illustration. Our experiments verify that our novel method is both highly effective and efficient.

Acknowledgement

The authors would be to thank Brian Zeleznik for his help on reviews and revisions for this paper.

Reference

1. Matthew V. Mahoney, K. Chan, Trajectory Boundary Modeling of Time Series for Anomaly Detection. SIGKDD Explorations Newsletter, Volume 7 Issue 2, pp:132-136, ACM, NY (2005).
2. Morchen Fabian, Unsupervised Pattern Mining from Symbolic Temporal Data. SIGKDD Explorations Newsletter, Volume 9 Issue 1, pp:41-45, ACM, NY (2007).
3. Tiancheng Zhang, Dejun Yue, Yu Gu, Ge Yu. Boolean Representation Based Data-Adaptive Correlation Analysis over Time Series Streams, CIKM'07 Information and knowledge management. Pp:203-212, ACM, NY (2007).
4. YiPing Ke, James Cheng, Correlated Pattern Minging in Quantitative Databases, ACM Transactions on Database Systems, Vol(33), No.3, Article 14, August 2008.
5. Ke Y., Cheng. Correlation search in graph databases. In proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp:390-399, ACM, New York (2007).
6. Edith Cohen, Mayur Datar, Finding Interesting Associations without Support Pruning, IEEE Transaction on Knowledge and Data Engineering, Vol(13), No 1, Feb, 2001
7. Kun-Cheng Tsai, Jing-Tian Sung, An Environment Sensor Fusion Application on Smart Building Skins, IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp: 291-295, June,2008.
8. Sharples S., Callaghan V. and Clarke G. A Multi-agent Architecture for Intelligent Building Sensing and Control, I*nternational Sensor Review Journal*, pp:1-8, May 1999.
9. Hani Hagras, Victor Callaghan, Martin Colley and Graham Clarke, A Hierarchical Fuzzy–genetic Multiagent Architecture for Intelligent Buildings Online Learning, Adaptation and

Control," *Information Sciences (Elsevier)*,vol. 150, pp: 33-57, March 2003.

10. Kay, R. Discovery of Frequent Distributed Event patterns in Sensor Networks. In Berlin, LNCS, vol 4913, pp: 106-124, Springer-Verlag Berlin Heidelberg (2008).
11. Azzedine Boukerche, Samer Samarah, A Novel Algorithm for Mining Association Rules in Wireless Ad Hoc Sensor Networks, IEEE Transactions on Parallel and Distributed Systems, Vol.19 No.7, July, 2008.
12. Agrawal R and Srikant R, Fast Algorithms for Mining Association Rule, Proc. 20th Int'l Conf. Very Large Data Bases(VLDB'94), pp:487-499, Morgan Kaufrmann San Francisco, CA (1994).
13. Agrawal R and Srikant R. Mining sequential patterns. Proceedings of the 11th international Conference on Data Engineering, Pages 3-14. IEEE Press, 1995
14. Shengnan Cong, Jiawei Han, Parallel mining of closed sequential patterns. KDD'05: Eleventh ACM SIGKDD international conference on knowledge discovery in data mining, pp:562-567, Chicago, Illinois (2005).
15. Intel Lab Data, http://berkeley.intel-research.net/labdata/, 2007.
16. J.Han, J. Pei, Y, Yin, Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, Data Mining and Knowledge Discovery, 2000 ACM SIGMOD international conference on Management of data. vol. 8, no.1, pp. 1-12 (2000).
17. M. Zaki and C. Hsiao. Charm: An Efficient Algorithm for Closed Itemset Mining. SDM'02, April 2002.