Distributed Pattern Discovery in Multiple Streams

Jimeng Sun[1], Spiros Papadimitriou[2], and Christos Faloutsos[1]

[1] Carnegie Mellon University
{jimeng, christos}@cs.cmu.edu
[2] IBM Watson Research Center
spapadim@us.ibm.com

Abstract. Given m groups of streams which consist of $n_1, \ldots, n_m$ co-evolving streams in each group, we want to: (i) incrementally find local patterns within a single group, (ii) efficiently obtain global patterns across groups, and more importantly, (iii) efficiently do that in real time while limiting shared information across groups. In this paper, we present a distributed, hierarchical algorithm addressing these problems. Our experimental case study confirms that the proposed method can perform hierarchical correlation detection efficiently and effectively.[1]

1 Introduction

Streams are often inherently correlated and it is possible to reduce hundreds of numerical streams into just a handful of *patterns* that compactly describe the key trends and dramatically reduce the complexity of further data processing. Multiple co-evolving streams often arise in a large distributed system, such as computer networks and sensor networks. Centralized approaches usually will not work in this setting. The reasons are: **(i) Communication constraint**; it is too expensive to transfer all data to a central node for processing and mining. **(ii) Power consumption**; in a wireless sensor network, minimizing information exchange is crucial because many sensors have very limited power. **(iii) Robustness concerns**; centralized approaches always suffer from single point of failure. **(iv) Privacy concerns**; in any network connecting multiple autonomous systems (e.g., multiple companies forming a collaborative network), no system is willing to share all the information, while they all want to know the global patterns. To sum up, a **distributed online algorithm** is highly needed to address all the above concerns.

To address this problem, we propose a hierarchical framework that intuitively works as follows:1) Each autonomous system first finds its local patterns and shares them with other groups. 2) Global patterns are discovered based on the shared local patterns. 3) From the global patterns, each autonomous system further refines/verifies their local patterns.

[1] The technical report [6] is a longer version of this work.

2 Problem Formalization and Framework

Given m groups of streams which consist of $\{n_1, \ldots, n_m\}$ co-evolving numeric streams, respectively, we want to solve the following two problems: (i) incrementally find patterns within a single group (*local pattern monitoring*), and (ii) efficiently obtain global patterns from all the local patterns (*global pattern detection*).

More specifically, *local pattern monitoring* can be modelled as a function,

$$F_L : (S_i(t+1,:), G(t,:)) \rightarrow L_i(t+1,:), \tag{1}$$

where the inputs are 1) the new input point $S_i(t+1,:)$ at time $t+1$ and the current global pattern $G(t,:)$ and the output is the local pattern $L_i(t+1,:)$ at time $t+1$. Details on constructing such a function will be explained in section 3. Likewise, *global pattern detection* is modelled as another function,

$$F_G : (L_1(t+1,:), \ldots, L_m(t+1,:)) \rightarrow G(t+1,:), \tag{2}$$

where the inputs are local patterns $L_i(t+1,:)$ from all groups at time $t+1$ and the output is the new global pattern $G(t+1,:)$.

Now we introduce the general framework for distributed mining. More specifically, we present the meta-algorithm to show the overall flow, using F_L (*local patterns monitoring*) and F_G (*global patterns detection*) as black boxes.

Intuitively, it is natural that global patterns are computed based on all local patterns from m groups. On the other hand, it might be a surprise that the local patterns of group i take as input both the stream measurements of group i and the global patterns. Stream measurements are a natural set of inputs, since local patterns are their summary. However, we also need global patterns as another input so that local patterns can be represented consistently across all groups. This is important at the next stage, when constructing global patterns out of the local patterns; we elaborate on this later. The meta-algorithm is the following:

```
Algorithm DistributedMining
0. (Initialization) At $t = 0$, set $G(t,:) \leftarrow$ null
1. For all $t > 1$
   (Update local patterns) For $i \leftarrow 1$ to $m$, set $L_i(t,:) := F_L(S_i(t,:), G(t-1,:))$
   (update global patterns) Set $G(t,:) := F_G(L_1, \ldots, L_m)$
```

3 Pattern Monitoring

Tracking Local Patterns. We now present the method for discovering patterns within a stream group. More specifically, we explain the details of function F_L (Equation 1). We first describe the intuition behind the algorithm and then present the algorithm formally. Finally we discuss how to determine the number of local patterns k_i.

The goal of F_L is to find the low dimensional projection $L_i(t,:)$ and the participation weights $W_{i,t}$ so as to guarantee that the reconstruction error $\|S_i(t,:) - \hat{S}_i(t,:)\|^2$ over time is predictably small.

The first step is, for a given k_i, to incrementally update the $k \times n_i$ participation weight matrix $W_{i,t}$, which serves as a basis of the low-dimensional projection for $S_i(t,:)$. Later in this section, we describe the method for choosing k_i. For the moment, assume that the number of patterns k_i is given.

The main idea behind the algorithm is to read the new values $S_i(t+1,:) \equiv [S_i(t+1,1),\ldots,S_i(t+1,n_i)]$ from the n_i streams of group i at time $t+1$, and perform three steps: (1) Compute the low dimensional projection $y_j, 1 \le j \le k_i$, based on the *current* weights $W_{i,t}$, by projecting $S_i(t+1,:)$ onto these.(2) Estimate the reconstruction error ($\boldsymbol{e}_j$ below) and the energy.(3) Compute $W_{i,t+1}$ and output the *actual* local pattern $L_i(t+1,:)$.

The term λ is a forgetting factor between 0 and 1, which helps adapt to more recent behavior. For instance, $\lambda = 1$ means putting equal weights on all historical data, while smaller λ means putting higher weight on more recent data.

In practice, we do not know the number k_i of local patterns. We propose to estimate k_i on the fly, so that we maintain a high percentage $f_{i,E}$ of the *energy* $E_{i,t}$. For each group, we have a low-energy and a high-energy threshold, $f_{i,E}$ and $F_{i,E}$, respectively. We keep enough local patterns k_i, so the retained energy is within the range $[f_{i,E} \cdot E_{i,t},\ F_{i,E} \cdot E_{i,t}]$.

```
Algorithm $F_L$
**Input:** new vector $S_i(t+1,:)$, old global patterns $G(t,:)$
**Output:** local patterns ($k_i$-dimensional projection) $L_i(t+1,:)$
1. Initialize $\boldsymbol{x}_1 := S_i(t+1,:)$.
2. For $1 \le j \le k$, we perform the following in order:
      $y_j := \boldsymbol{x}_j W_{i,t}(j,:)^T$                                  ($y_j$ = projection onto $W_{i,t}(j,:)$)
    If $G(t,:) = \text{null}$, then $G(t,j) := y_j$          (handling boundary case)
      $d_j \leftarrow \lambda d_j + y_j^2$                                      (local energy, determining update magnitude)
      $\boldsymbol{e} := \boldsymbol{x}_j - G(t,j)W_{i,t}(j,:)$                  (error, $\boldsymbol{e} \perp W_{i,t}(j,:)$)
    $W_{i,t+1}(j,:) \leftarrow W_{i,t}(j,:) + \frac{1}{d_j}G(t,j)\boldsymbol{e}$   (update participation weight)
      $\boldsymbol{x}_{j+1} := \boldsymbol{x}_j - G(t,j)W_{i,t+1}(j,:)$          (repeat with remainder of $\boldsymbol{x}$).
3. Compute the new projection $L_i(t+1,:) := S_i(t+1,:)W_{i,t+1}^T$
```

Tracking Global Patterns. We now present the method for obtaining global patterns over all groups. More specifically, we explain the details of function F_G.

First of all, what is a global pattern? Similar to local pattern, global pattern is low dimensional projections of the streams from all groups. Loosely speaking, assume only one global group exists which consists of all streams, the global patterns are the local patterns obtained by applying F_L on the global group—this is essentially the centralized approach. In other words, we want to obtain the result of the centralized approach without centralized computation.

The algorithm exactly follows the lemma above. The j-th global pattern is the sum of all the j-th local patterns from m groups.

```
Algorithm $F_G$
Input: all local patterns $L_1(t,:), \ldots, L_m(t,:)$
Output: global patterns $G(t,:)$
0. Set $k := max(k_i)$ for $1 \leq i \leq m$
1. For $1 \leq j \leq k$, set $G(t,j) := \sum_{i=1}^{m} L_i(t,j)$ (if $j > k_i$ then $L_i(t,j) \equiv 0$)
```

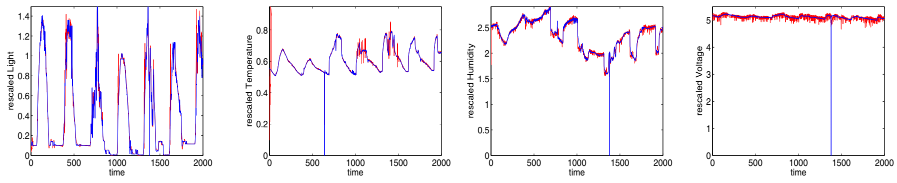
4 Experimental Case Study

The `Motes` dataset consists of 4 groups of sensor measurements (i.e., light intensity, humidity, temperature, battery voltages) collected using 48 Berkeley Mote sensors at different locations in a lab, over a period of a month.

The main characteristics (see the blue curves in Figure 1) are: (1) Light measurements exhibit a clear global periodic pattern (daily cycle) with occasional big spikes from some sensors (outliers), (2) Temperature shows a weak daily cycle and a lot of bursts. (3) Humidity does not have any regular pattern. (4) Voltage is almost flat with a small downward trend.

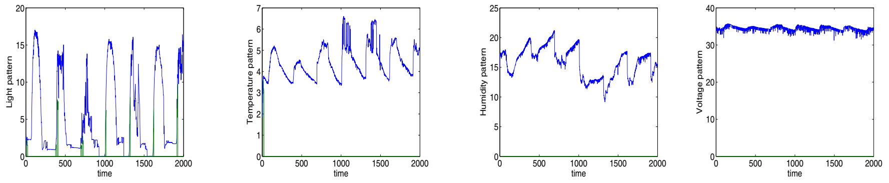
The reconstruction is very good (see the red curves in Figure 1(a)), with relative error below 6%. Furthermore, the local patterns from different groups are correlated well with the original measurements (see Figure 2). The global patterns (in Figure 3) are combinations of different patterns from all groups and reveal the overall behavior of all the groups.

The relative reconstruction error as the evaluation metric. The best performance is obtained when all groups exchange up-to-date local/global patterns at every timestamp, which is prohibitively expensive. One efficient way to deal with this problem is to increase the communication period, which is the number of



(a) Light measurements (b) Temperature (c) Humidity (c) Voltage

Fig. 1. original measurements (blue) and reconstruction (red) are very close



(a) Light patterns (b) Temperature patterns (c) Humidity patterns (d) Voltage patterns

Fig. 2. Local patterns

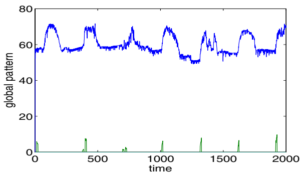


Fig. 3. Global patterns

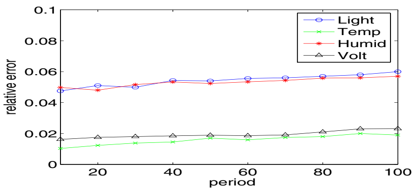


Fig. 4. Error increases slowly

timestamps between successive local/global pattern transmissions.Overall, the relative error rate increases very slowly as the communication period increases (see Figure 4). This implies that we can dramatically reduce communication with minimal sacrifice of accuracy.

5 Related Work

Distributed Data Mining. Most of works on distributed data mining focus on extending classic (centralized) data mining algorithms into distributed environment, such as association rules mining [3], frequent item sets [5]. Web is a popular distributed environment. Several techniques are proposed specifically for that, for example, distributed top-k query [2] But our focus are on finding numeric patterns, which is different.

Privacy Preserving Data Mining. The most related discussion here is on how much privacy can be protected using subspace projection method [1, 4]. Liu et al. [4] discuss the subspace projection method and propose a possible method to breach the protection using Independent component analysis(ICA). All the method provides a good insight on the issues on privacy protection. Our method focuses more on incremental online computation of subspace projection.

6 Conclusion and Acknowledgement

We focus on finding patterns in a large number of distributed streams. More specifically, we first find local patterns within each group, where the number of local patterns is automatically determined based on reconstruction error. Next, global patterns are identified, based on the local patterns from all groups. We evaluated our method on several datasets, where it indeed discovered the patterns. We gain significant communication savings, with small accuracy loss.

Work partially supported by the NSF under Grants No. IIS-0209107 IIS-0205224 INT-0318547 SENSOR-0329549 IIS-0326322 and the Pennsylvania Infrastructure Technology Alliance (PITA) This publication only reflects the authors views.

References

1. C. Agrawal and P. Yu. A condensation approach to privacy preserving data mining. In *EDBT*, 2004.
2. B. Babcock and C. Olston. Distributed Top-K Monitoring. In *SIGMOD*, 2003.
3. D. W. Cheung, V. T. Ng, A. W. Fu, and Y. Fu. Efficient Mining of Association Rules in Distributed Databases. *TKDE*, 8:911–922, 1996.
4. K. Liu, H. Kargupta, and J. Ryan. Multiplicative noise, random projection, and privacy preserving data mining from distributed multi-party data. In *TKDE*, 2005.
5. K. K. Loo, I. Tong, B. Kao, and D. Cheung. Online Algorithms for Mining Inter-Stream Associations From Large Sensor Networks. In *PAKDD*, 2005.
6. Jimeng Sun, Spiros Papadimitriou, and Christos Faloutsos. Distributed pattern discovery in multiple streams. Technical Report CMU-CS-06-100, Carnegie Mellon Univ., 2005.