Poster Paper

A Distributed Mining Framework for Influence in Evolving Entities

Tian Guo
EPFL, Switzerland
tian.guo@epfl.ch

Karl Aberer
EPFL, Switzerland
karl.aberer@epfl.ch

ABSTRACT

Mining dynamic influence in evolving entities, which provides insights into the interaction and causal relations among entities, is an important and fundamental data mining task. Meanwhile, nowadays pervasive sensors in a variety of contexts give rise to the development of many distributed real-time computation systems intended for massive time series streams. In this paper, we focus on mining dynamic influence from time series data generated by entities via such a distribute real-time computation system. The proposed D^{2}InfMiner framework encompasses a statistical lead-lag correlation based influence detection module and an on-line model for dynamic influence inference. We implement D^{2}InfMiner framework based on Apache Storm.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information storage and retrieval

Keywords

Time series, Influence mining, Distribued data procesing

1. INTRODUCTION

For our contemporary interconnected and dynamic changing world, dynamic influence in evolving entities described by time series is fundamental knowledge that helps people understand the behaviours of involved entities. For instance, as massive powerful and various sensors are becoming prevalent in our daily life (e.g., mobile phones, sensor networks, smart meters and etc.), influence among the time series generated by these sensors reflects the real-time status of the carriers and their interactions. Moreover, such quickly and continuously increasing amount of real-time data leads to the development of many distributed real-time computation systems [1], analogous to MapReduce ecosystem designed for large-scale static data.

This paper aims at addressing the problem of mining evolving entity influence based on such a new emerging distributed real-time computation paradigm (DisMineInflu problem). DisMineInflu problem is of great value to various applications such as event/anomaly detection, trend prediction, casualty analysis and so on. For instance, for data-driven event detection in performance monitoring of data centres, when an event is detected from the performance time series (e.g., network I/O) *w.r.t.* a certain server, using our proposed dynamic influence mining framework, operators can quickly identify from large-scale of servers which one(s) are highly probable to be affected by this event in a certain time and then respond in advance. It is also especially applicable in the financial markets and social data analysis [5–7].

Specifically, our proposed distributed data mining framework should be able to tackle the following challenges. Contrary to static influence mining, since new arriving observations of time series from evolving entities are continuously distributed into different computing nodes of a cluster, mining dynamic influence via the distributed real-time computation system requires a computation and communication efficient solution. Otherwise the system would encounter bottlenecks, which lead the influence detection to lag further and further over time and report stale results [1]. Another challenge lies in modeling the dynamic influence through time series. Since the evolving data could be quite volatile during some periods or events [8], the proposed framework should entail a statistical model responsible for providing stable influence inference. How to efficiently on-line maintain this model to capture the dynamic nature of influence is also non-trivial.

Contributions: This paper is the first work that proposes a truly distributed and real-time solution for DisMineInflu problem. The approach in [6] assumes that the underlying influence relationships are static. [5, 7, 8] focus on mining influence in a centralized way and do not the consider the data communication overhead in the distributed environment as well as correlations among time series of entities. Overall, this paper makes the following concrete contributions: we formally define the problem of continuously mine dynamic influence from time series yielded by evolving entities based on a distributed real-time computation system (DisMineInflu problem). The D^{2}InfMiner framework is proposed to optimize both communication and computation cost as well as providing statistical inference of influence for DisMineInflu problem. We implement D^{2}InfMiner framework based on Apache Storm.

2. PROBLEM DEFINITIONS

In this section, we formulate the DisMineInflu problem.

2.1 Distributed Real-time Computation Engine

In a typical cluster of a distributed real-time computation engine [1], *Topology* is a job submitted to the cluster, which is a program-described directed acyclic graph (DAG). The vertexes are user-defined processing elements denoted by *boxes* and the communication between boxes is dictated by the edges in the topology.

A topology is executed to continuously process tuples. Each box has a user specified number of *tasks* denoted by *parallelism* of the box and such tasks are executed in parallel to process the tuples sent to this box. *Shuffling function* is a function between two boxes, which determines to which task of the connected box a tuple from the preceding box should be sent.

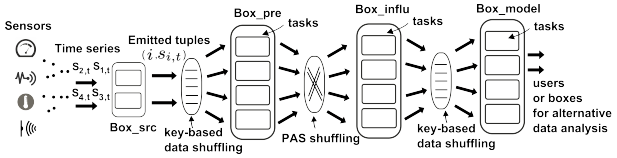

Figure 1: The topology of dynamic influence mining in a distributed real-time computation system

We use n to denote the total number of entities continuously feeding time series streams to the cluster. For an entity i ($1 \leq i \leq n$), let $\boldsymbol{s}_i$ denote the sequence of discrete real-valued observations $s_{i,t}$ (t represents a time instant) of this entity's attribute. The sliding window of length h ending at time instant t of entity i is denoted by $\boldsymbol{s}_i^t = (s_{i,t-h+1}, \cdots, s_{i,t})$ and $\boldsymbol{s}_i^t \in \mathbb{R}^h$.

2.2 Problem Statement

In this paper, we utilize statistical lead-lag correlations, namely Pearson correlation and Spearman correlation to measure the influence between time series of entities [4, 8].

DEFINITION 2.1 (LEAD-LAG CORRELATIONS). *Define a generic correlation function for sliding windows $\boldsymbol{s}_i^{t_1}$ and $\boldsymbol{s}_j^{t_2}$ of time series of two entities i and j as* $corre(\boldsymbol{s}_i^{t_1}, \boldsymbol{s}_j^{t_2}) = \frac{(\boldsymbol{s}_i^{t_1} - \mu(\boldsymbol{s}_i^{t_1})\mathbb{1}) \cdot (\boldsymbol{s}_j^{t_2} - \mu(\boldsymbol{s}_j^{t_2})\mathbb{1})}{(h-1)\sigma(\boldsymbol{s}_i^{t_1})\sigma(\boldsymbol{s}_j^{t_2})}$ *where $\mathbb{1}$ is all one vector ($\mathbb{1} \in \mathbb{R}^h$), $\sigma(\boldsymbol{s}_i^{t_1})$ and $\mu(\boldsymbol{s}_i^{t_1})$ are the sample standard deviation and mean of the elements in $\boldsymbol{s}_i^{t_1}$, respectively [4]. Assume $t_1 < t_2$ and $corre(\boldsymbol{s}_i^{t_1}, \boldsymbol{s}_j^{t_2})$ measures the correlation between time series i and j with lag $\tau = t_2 - t_1$.*

Pearson correlation coefficient $\rho_{i,j}^{t_1,\tau}$, which evaluates the lagged linear relationship, is defined as follows [4]: $\rho_{i,j}^{t_1,\tau} = corre(\boldsymbol{s}_i^{t_1}, \boldsymbol{s}_j^{t_2})$

Spearman correlation $\xi_{i,j}^{t_1,\tau}$, which measures the strength of lagged monotonic relationship is defined as: $\xi_{i,j}^{t_1,\tau} = corre(\boldsymbol{r}_i^{t_1}, \boldsymbol{r}_j^{t_2})$, where the entries of $\boldsymbol{r}_i^{t_1}$ are the ranks of the corresponding entries in original $\boldsymbol{s}_i^{t_1}$ [4].

For a certain application, users can choose either Pearson or Spearman correlation to measure the influence in evolving entities as defined below:

DEFINITION 2.2 (CORRELATION BASED INFLUENCE). *Given an application-specific correlation threshold ϵ, for the time series of entities i and j ($1 \leq i, j \leq n$), entity i has influence on entity j at time t_1 with lag τ if $\rho_{i,j}^{t_1,\tau}$ (or $\xi_{i,j}^{t_1,\tau}$) is significantly above ϵ.*

Intuitively, correlation based influence evaluates to what extent entity j will have a correlated trend with i in time ℓ [3, 8]. For simplicity, we call it influence in the rest of paper.

Now the DisMineInflu problem is formulated as:

DEFINITION 2.3 (DISMINEINFLU PROBLEM). *Assume n time-series streams collected from corresponding n entities are continuously arriving and distributed to different nodes of a distributed real-time computation system. DisMineInflu problem requires to mine the following information for each entity:*

(1) continuously report the entities on which it has influence within a maximum lag ℓ;

(2) on-line maintain a statistical model over the detected dynamic influence such that the probability that certain entities will be impacted within maximum lag ℓ can be inferred.

The maximum lag ℓ represents the temporal range of users' interest to model the dynamic influence and also indicates how far in advance users want to predict the future based on detected influence. In real applications, due to the dynamic nature of evolving entities and observational errors, the yielded time series often embraces volatile or sudden changed influence [7, 8]. The statistical model built on the real-time detected influence from sub-problem (1) enables to discover significant and stable influence in entities. It can also serve for event and anomaly detection [2], influence prediction and so on [5, 7, 8].

3. DISTRIBUTED DYNAMIC INFLUENCE MINER

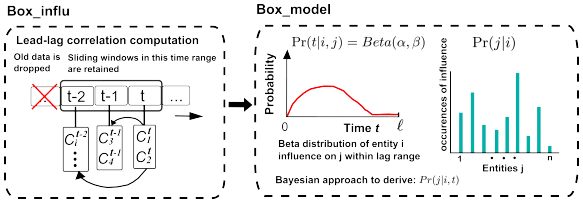

Figure 2: Illustration of key components in the topology

In this section, we briefly describe the proposed D^{2}InfMiner framework whose topology is shown in Figure 1. Box_pre is in charge of maintaining sliding windows and preparing the tuples for PAS-shuffling. Box_influ collects the data sent by PAS-shuffling and calculates the qualified lead-lag correlations based on hypercube computation pruning. Refer [3] for details of Box_pre, Box_influ and PAS-shuffling. Then Box_model builds a beta-distribution based Bayesian approach to estimate the probability of entity i's influence on entity j at certain time instances. Figure 2 depicts some details of D^{2}InfMiner framework.

4. ACKNOWLEDGMENTS

This work was supported by Nano-Tera.ch through the OpenSense II project.

5. REFERENCES

[1] Apache Storm. `https://storm.apache.org/`.

[2] X. C. Chen and et al. Online discovery of group level events in time series. In *SIAM SDM*, 2014.

[3] T. Guo, J.-P. Calbimonte, H. Zhuang, and K. Aberer. Sigco: Mining significant correlations via a distributed real-time computation engine. In *Big Data (Big Data), 2015 IEEE International Conference on*.

[4] D. A. Kenny. Correlation and causality.

[5] C. Liao and et al. Mining influence in evolving entities: A study on stock market. In *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, pages 244–250. IEEE, 2014.

[6] X. Shi and et al. Discovering shakers from evolving entities via cascading graph inference. In *Proceedings of the 17th ACM SIGKDD*, pages 1001–1009. ACM, 2011.

[7] X. Shi, W. Fan, and S. Y. Philip. Dynamic shaker detection from evolving entities. 2011.

[8] D. Wu, Y. Ke, J. X. Yu, S. Y. Philip, and L. Chen. Leadership discovery when data correlatively evolve. *World Wide Web*, 14(1):1–25, 2011.