Rheinische

Friedrich-Wilhelms-Universitt Bonn

Master Thesis Presentation

Distributed Online Learning for Large-scale Pattern Prediction over Real-time Event Streams

Ehab Qadah

First Examiner:
PD. Dr. Michael Mock

Second Examiner:
Prof. Dr. Stefan Wrobel

Outline

Motivation and Problem Formulation

The Proposed Approach

Event Forecasting with Pattern Markov Chains

Our Proposed Approach using Distributed Online Learning

Evaluation of Proposed Approach

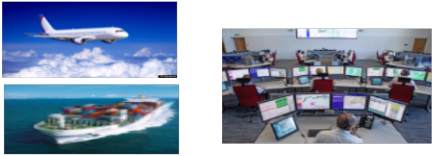
Work Plan

Motivation

A New Era: Big event Data streams

User Activities on Web

Internet of Things (IoT)



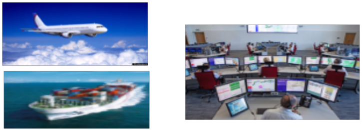
Maritime & Aviation surveillance

Motivation

A New Era: Big event Data streams

User Activities on Web

Internet of Things (IoT)



Maritime & Aviation surveillance

.

- Common Goal: recognition and prediction full matches of complex event patterns in real-time.

Problem Formulation

- Given a set of k real-time streams of events $S = \{s_1, s_2, ..., s_k\}$.
- Each stream $s_i = \langle e_1, e_3, ..., e_t, ...\rangle$ is a time-ordered infinite sequence of events.
- Each event is defined as a tuple of attributes $e_i = (id, type, \tau, a_1, a_2, \ldots, a_n)$, where $type \in \Sigma$ (i.e., event types), $\tau \in \mathbb{R}$, and $id \in \mathbb{N}$.
- A user-defined pattern $\mathcal{P}$ is given in the form of a regular expression over a set of event types. Σ ($\mathcal{P} := E \mid \mathcal{P}_1; \mathcal{P}_2 \mid \mathcal{P}_1 \vee \mathcal{P}_2 \mid \mathcal{P}_1^*$, $E \in \Sigma$).

Problem Formulation

- Goal: the main objective is to predict the pattern $\mathcal{P}$ completion with certain probability in the future over each stream s_i given the current time event e_t.

Problem Formulation

- Goal: the main objective is to predict the pattern $\mathcal{P}$ completion with certain probability in the future over each stream s_i given the current time event e_t.
- (**Pattern Prediction over Multiple Event Streams**)

An Illustrative Application Domain:

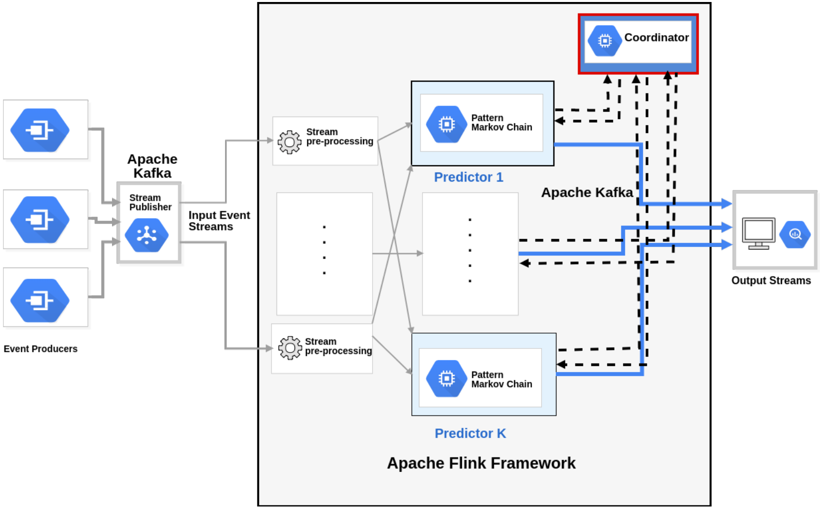
Maritime Surveillance

- Event tuple (i.e., critical points) derived from raw Automatic Identification System (AIS) messages of moving vessels e.g.,


```
{
"timestamp":1443651492000,
"id":"228133000",
"annotation":"change_in_heading",
"latitude":48.117775,
"longitude":-4.4205885,
"distance":323.406,
"heading":264.27
"speed":18.48,
}
```


- Example patterns such as:
  $\mathcal{P}_1 = \mathit{Sailing}$ or
  $\mathcal{P}_2 = \mathit{changeInHeading}; \mathit{gapStart}; \mathit{gapEnd}; \mathit{changeInHeading}$

System Architecture



Scalable Pattern Prediction System [2]

- A scalable and distributed system that provides online pattern prediction over multiple real-time streams of events.
- The proposed system is based on a novel method that combines online probabilistic prediction models based on pattern Markov chain technique [1] with a distributed online learning protocol [5] to learn a global prediction model in a communication-efficient way.
- For large-scale processing support, the system is implemented on top of Apache Flink [2] along with Apache Kafka [4].
- Developed in the context of the datAcron project [1].

[1] http://www.datacron-project.eu/

[2] Source code: `https://goo.gl/BZ2Prk`.

Event Forecasting with Pattern Markov Chains [1]

overview

- The system consumes a single input stream s_i of events.
- The event stream s_i is assumed to be generated by m-order Markov source.
- The event pattern $\mathcal{P}$ is defined in the form of regular expressions over a finite set of event types Σ.
- A probabilistic model provides online forecasting reports when the $\mathcal{P}$ is expected to be completed in future.

Event Forecasting with Pattern Markov Chains

How does it work?

- The pattern $P = Sailing \cdot Stopping$ is converted to *DFA* with $\Sigma = \{Moving, Sailing, Stopping\}$.

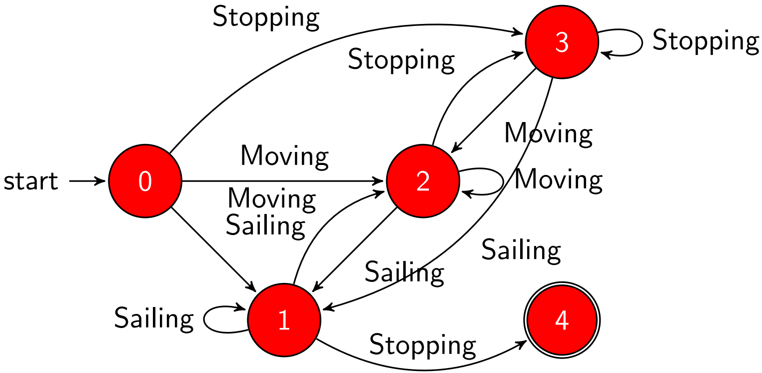


Figure: $Q = \{0, 1, 2, 3, 4\}$ $m = 1$

Event Forecasting with Pattern Markov Chains

How does it work?

- The deterministic finite automa (*DFA*) is used to construct a Markov chain, which is called a Pattern Markov Chain (PMC).
- The states of *DFA* is directly mapped to states of transition probability matrix $\boldsymbol{M}$ $|Q| \times |Q|$ of the *PMC*.
- $$\boldsymbol{M} = \begin{Bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix} \begin{pmatrix} p_{0,0} & . & . & . & p_{0,4} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & p_{4,4} \end{pmatrix}$$
- The maximum-likelihood estimator is used to compute the transition probabilities $p_{i,j}$ of the matrix $\boldsymbol{M}$

$$\hat{p}_{i,j} = \frac{n_{i,j}}{\sum_{k \in Q} n_{i,k}} = \frac{n_{i,j}}{n_i} \tag{1}$$

.

Event Forecasting with Pattern Markov Chains

How does it work?

- The probability distribution of the waiting-time (i.e., time required until the pattern is completed from state i) $P(W_P(i) = n)$, is calculated based on the Markov chain transition matrix.

$$P(W_P(i) = n) = \boldsymbol{\xi_i}^T \boldsymbol{N}^{n-1}(\boldsymbol{I} - \boldsymbol{N})\mathbf{1}$$

where

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{N} & \boldsymbol{C} \\ \mathbf{0} & \boldsymbol{I} \end{pmatrix} \tag{2}$$

- The forecast intervals $I = (start, end)$ (i.e., the completion interval of the pattern P from the current state q) are built using the waiting-time distribution

$$P(I) = \sum_{n \in I} P(W_P(q) = n) \quad \text{and} \quad P(I) \geq prediction\_threshold$$

.

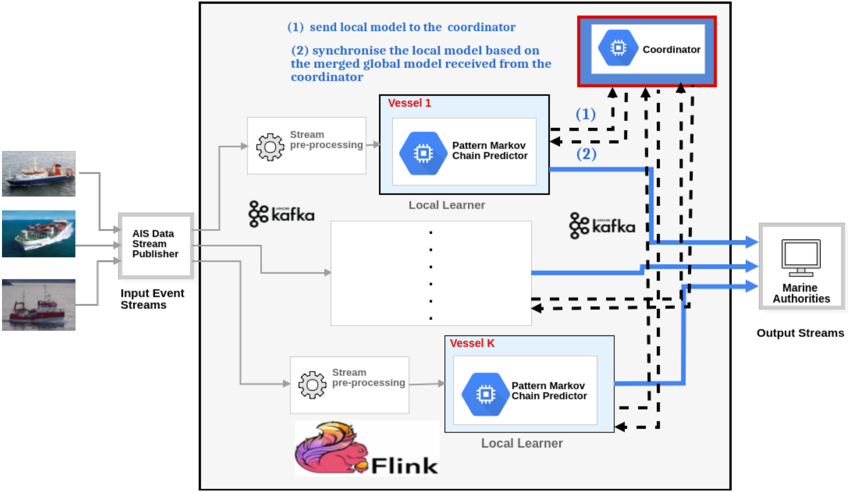
Event Forecasting with Pattern Markov Chains

What is wrong?

- The overhead learning time of the Markov transition matrix for each input stream isolated from the others, how can we accelerate it?
- The low performance of the model for an input event stream with less data.
- How can we enable the communication between the local models and leverage an aggregated global shared model in a distributed and communication efficient fashion while maintaining provable and strong guarantees in service quality?

Distributed Online Learning for Large-Scale Patterns Prediction

Proposed System Model



Communication-Efficient Distributed Online Prediction by Dynamic Model Synchronization

[5]

- A protocol for distributed online prediction over multiple input data streams in a communication efficient manner.
- It allows to combine local models into a global model using a *synchronization operation*.
- The distributed learners exchange their local model with a central coordinator node periodically after observing a fixed number of data points (i.e., mini-batches) [3].
- A dynamic synchronization scheme based on monitoring the local models variance from a global reference model ($\|f_i - r\|^2 \leq \triangle$).

Distributed Online Learning for Large-Scale Patterns Prediction

Proposed Synchronization Operation

- The *synchronization operater* of the local maximum likelihood estimator as the following:

$$\hat{p}_{i,j} = \frac{\sum_{k \in K} n_{k,i,j}}{\sum_{k \in K} \sum_{l \in L} n_{k,i,l}}$$

Evaluation of Proposed Approach

Selected Techniques

- Theoretical Analysis.
- Simulated synthetic data streams.
- Over real-world data streams of AIS [3] messages or derived trajectories critical points of moving vessels [6].

[3] AIS: `www.navcen.uscg.gov/?pageName=AISmain`

Evaluation of Proposed Approach

Performance Measures

- $Precision = \frac{\#\ of\ correct\ predictions}{\#\ of\ total\ predictions}$: how many of the produced predictions are correct.
- $Recall = \frac{\#\ of\ full\ matches\ predicted\ at\ least\ once}{\#\ of\ total\ detected\ full\ matches}$: how many of the detected full matches of the defined pattern does the model predicate at least once in previous prediction.
- *Cumulative prediction loss*: aggregated number of all wrong predictions.

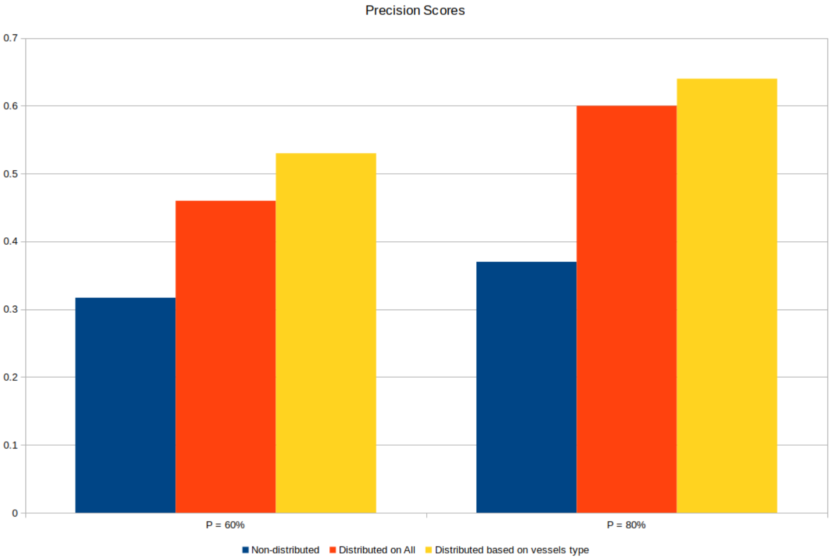
Empirical evaluation

Experimental Setup

- Synopses over raw AIS messages in Brest, France: 1 October 2015 to 31 March 2016.
- $4,684,444$ derived critical points.
- ≈ 5000 vessels.
- Used patterns:
  $P_1 = \mathit{change\_heading} \cdot \mathit{gap\_start} \cdot \mathit{gap\_end} \cdot \mathit{change\_heading}$
  or $P_2 = \mathit{Sailing}$

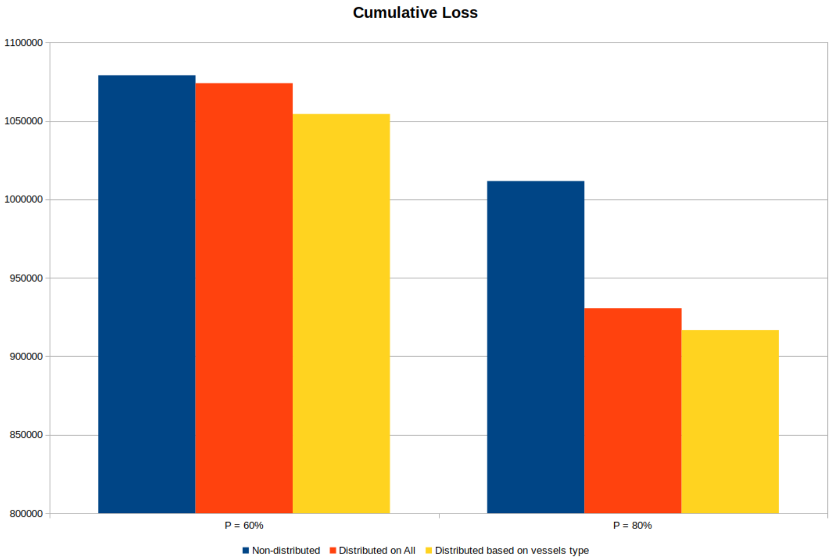
Empirical evaluation

Initial Results ($P = Sailing\ and\ batch\ size = 8$)

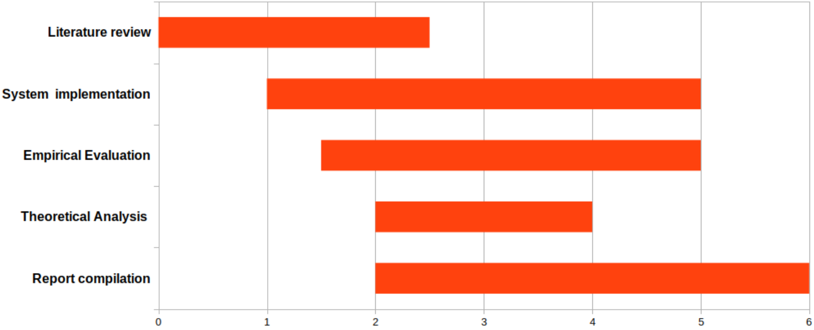


Empirical evaluation

Initial Results ($P = Sailing\ and\ batch\ size = 8$)



Work Plan


Literature review
System implementation
Empirical Evaluation
Theoretical Analysis
Report compilation
0
1
2
3
4
5
6

Bibliography I

[1] Elias Alevizos, Alexander Artikis, and George Paliouras. Event forecasting with pattern markov chains. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, pages 146–157. ACM, 2017.

[2] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4), 2015.

[3] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

[4] The Apache Software Foundation. Apache Kafka. `https://kafka.apache.org/`, 2012.

Bibliography II

[5] Michael Kamp, Mario Boley, Daniel Keren, Assaf Schuster, and Izchak Sharfman. Communication-efficient distributed online prediction by dynamic model synchronization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 623–639. Springer, 2014.

[6] Kostas Patroumpas, Elias Alevizos, Alexander Artikis, Marios Vodas, Nikos Pelekis, and Yannis Theodoridis. Online event recognition from moving vessel trajectories. *GeoInformatica*, 21 (2):389–427, 2017.