Event Trend Aggregation Under Rich Event Matching Semantics

Olga Poppe\(^1\), Chuan Lei\(^2\), Elke A. Rundensteiner\(^3\), and David Maier\(^4\)

\(^1\)Microsoft Gray Systems Lab, \(^2\)IBM Research – Almaden, \(^3\)Worcester Polytechnic Institute, \(^4\)Portland State University

July 3rd, 2019

Supported by NSF grants IIS-1815866, CRI-1305258, IIS-1018443
Algorithmic Trading

**Goal:** Reliable actionable insights about the stream

**Solution:** Each event is considered in the context of other events in the stream

Picture source: http://www.businessxack.com/how-to-know-the-stock-market-trend/1303
Algorithmic Trading

- Single event = *Single* stock value
- Event sequence = Stock down trend of *fixed* length
- Event trend = Stock down trend of *arbitrary* length
Algorithmic Trading

- Single event = *Single* stock value
- Event sequence = Stock down trend of *fixed* length
- Event trend = Stock down trend of *arbitrary* length
Algorithmic Trading

- Single event = *Single* stock value
- Event sequence = Stock down trend of *fixed* length
- Event trend = Stock down trend of *arbitrary* length
Algorithmic Trading

- Single event = *Single* stock value
- Event sequence = Stock down trend of *fixed* length
- Event trend = Stock down trend of *arbitrary* length under the *skip-till-next-match* semantics
Event Trend Aggregation Under Rich Event Matching Semantics

Algorithmic Trading  Ridesharing Service  Cluster Monitoring

**Number** of down-trends per sector ignoring local price fluctuations

**Average speed** of Uber trips per district ignoring irrelevant events

**Total CPU load** per mapper experiencing contiguously increasing load

Skip-till-any-match semantics  Skip-till-next-match semantics  Contiguous semantics


Worcester Polytechnic Institute
Complexity of Event Trend Analytics
Complexity of Event Trend Analytics

Existing trends

\[ e \]
Complexity of Event Trend Analytics

Real-time event trend aggregation despite
- Rich event matching semantics
- Exponential number and arbitrary length of trends
- Complex event inter-dependencies in a trend
Existing Two-Step Approaches

**Step 1:** Event Trend Construction

**Step 2:** Event Trend Aggregation

**Event Trend Aggregation Query**

```
RETURN sector, COUNT(*)
PATTERN Stock S+
WHERE [company, sector] AND S.price > NEXT(S).price
SEMANTICS skip-till-any-match
GROUP-BY sector WITHIN 30 min SLIDE 1 min
```

**Event Stream**

- Sector id
- Company id
- Price
- Time

Coarse-Grained Online Trend Aggregation

**Cogra:**
Coarse-Grained Online Trend Aggregation

Number of aggregates decreases

- Coarse granularity
- Pattern-grained aggregates
- Type-grained aggregates
- Event-grained aggregates

**Quadratic** time & **linear** space complexity

---

**Event Trend Aggregation Query**

```
RETURN sector, COUNT(*)
PATTERN Stock S+
WHERE [company, sector] AND S.price > NEXT(S).price
SEMANTICS skip-till-any-match
GROUP-BY sector WITHIN 30 min SLIDE 1 min
```
Approach Overview

COGRA Framework

Event query ➔

Pattern Analyzer
Predicate Classifier
Granularity Selector

Static Query Analyzer

Cogra configuration

Event stream ➔

Type-Grained Aggregator
Mixed-Grained Aggregator
Pattern-Grained Aggregator

Runtime Executor

Aggregation results ➔
Cogra Template

Nested Kleene Pattern

\[ P = (\text{SEQ}(A^+, B)) + \]

- A’s are preceded by a’s and b’s
- B’s are preceded by a’s

Start type

End type
Online Type-Grained Aggregator for skip-till-any-match semantics

**Table:**

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Online Type-Grained Aggregator for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

A \rightarrow B \text{ (SEQ)}

+ +
Online Type-Grained Aggregator
for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>b2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Online Type-Grained Aggregator
for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

**Event trends:**
(a1,b2)
Online Type-Grained Aggregator
for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>b2</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>a3</td>
<td>3</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Event trends: (a1,b2)
Online Type-Grained Aggregator for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>a3</td>
<td>3</td>
<td></td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Event trends: (a1,b2)
Online Type-Grained Aggregator
for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Event</th>
<th>a.count</th>
<th>b.count</th>
<th>A.count</th>
<th>B.count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>a3</td>
<td>3</td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>a4</td>
<td>6</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>b6</td>
<td></td>
<td>10</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>a7</td>
<td>22</td>
<td></td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>b8</td>
<td></td>
<td>32</td>
<td></td>
<td>43</td>
</tr>
</tbody>
</table>

Event trends:
(a1, b2)
(a1, a3, b6)
(a1, a3, a4, b6)
(a1, b2, a3, a4, b6)
(a1, b2, a2, b6, a7, b8)
(a1, b2, a2, a3, b6, a7, b8)
...

Worcester Polytechnic Institute
# Online Type-Grained Aggregator for skip-till-any-match semantics

<table>
<thead>
<tr>
<th>Idea</th>
<th>Existing Two-Step Approaches</th>
<th>Cogra</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. Construct all trends</td>
<td>One aggregate is kept per event type</td>
</tr>
<tr>
<td></td>
<td>2. Aggregate them</td>
<td></td>
</tr>
<tr>
<td>Time complexity</td>
<td>Exponential in #events per window</td>
<td>Linear in #events per window, i.e., <strong>optimal</strong></td>
</tr>
<tr>
<td>Space complexity</td>
<td>Exponential if all trends are stored</td>
<td>Linear in #event types in the pattern</td>
</tr>
</tbody>
</table>

1. **Cogra**

- **Idea**: One aggregate is kept per event type.
- **Time complexity**: Linear in #events per window, i.e., **optimal**.
- **Space complexity**: Linear in #event types in the pattern.
Online Pattern-Grained Aggregator
for skip-next-any-match & contiguous semantics

<table>
<thead>
<tr>
<th>Idea</th>
<th>Existing Two-Step Approaches</th>
<th>Cogra</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. Construct all trends</td>
<td>One aggregate is kept per pattern</td>
</tr>
<tr>
<td></td>
<td>2. Aggregate them</td>
<td></td>
</tr>
<tr>
<td>Time complexity</td>
<td><strong>Polynomial</strong> in #events per window</td>
<td><strong>Linear</strong> in #events per window, i.e., <strong>optimal</strong></td>
</tr>
<tr>
<td>Space complexity</td>
<td><strong>Polynomial</strong> if all trends are stored</td>
<td><strong>Constant</strong></td>
</tr>
</tbody>
</table>

Cogra enables **real-time in-memory** event trend aggregation.
Experimental Setup

Execution infrastructure:
Java 8, 1 Linux machine with 16-core
3.4 GHz CPU and 128 GB of RAM

Data sets:

- New York city taxi and Uber data set (330 GB)
  - Event trend = Taxi or Uber trip
- Physical activity real data set (1.6 GB)
  - Event trend = Sequence of physical activities
- Stock real data set (1.3 GB)
  - Event trend = Stock market trend

- Unified New York City Taxi and Uber data. https://github.com/toddwschneider/nyc-taxi-data
## Event Aggregation Approaches

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Kleene closure</th>
<th>Event matching semantics</th>
<th>Online sequence\trend aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Skip-till-any-match</td>
<td>Skip-till-next-match</td>
</tr>
<tr>
<td><strong>Flink</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Sase</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Greta</strong></td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>A-Seq</strong></td>
<td>--</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Cogra</strong></td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Flink:** [https://fink.apache.org/](https://fink.apache.org/)


Cogra is a win-win solution that achieves up to $10^6$ speed-up and up to $10^7$ memory reduction compared to state-of-the-art.
Contributions

We are the first to compute aggregation of Kleene pattern matches under rich event matching semantics with optimal time complexity

- Cogra incrementally maintains event trend aggregates at the coarsest granularity
- Cogra guarantees quadratic time complexity and linear space complexity in the number of events in the worst case
- Cogra enables real-time in-memory event trend aggregation as required by time-critical streaming applications
Thank You