Tackling Semantic Shift in Industrial Streaming Data Over Time

Lisa Ehrlinger, Christian Lettner, Johannes Himmelbauer

DI Lisa Ehrlinger
Senior Researcher Data Analysis Systems / Software Competence Center Hagenberg
Senior Researcher / Johannes Kepler University Linz
+43 50 343 836
lisa.ehrlinger@scch.at
www.scch.at
Data Quality Research at JKU and SCCH

Johannes Kepler University (JKU) Linz
- Senior researcher in research group of a.Univ.-Prof. Wolfram Wöß
- DQ tool DQ-MeeRKat: https://github.com/lisehr/dq-meerkat
- Talks at MIT Chief Data Officer and Information Quality Symposium 2019 and 2020

Software Competence Center Hagenberg GmbH (SCCH)
- Lead of research focus “Data Management and Data Quality”
- Research on DQ issues with industrial companies (e.g., KTM)
- DQ tool: A DaQL to Monitor Data Quality in Machine Learning Applications
The Software Competence Center Hagenberg (SCCH) is part of the software park Hagenberg.

Non-profit organization for data science and software science

Founded 1999

~ 80 employees

> 7 Mio. € turnover

COMET Center

At JKU / Open Innovation Center

Union of SCCH partner companies
Semantic Shift – A Data Quality Problem

- Linguistics: “semantic shift”
  - Also: “semantic change”, “semantic drift”
  - Evolution of word meaning over time (Bloomfield 1933)

- Machine learning (ML) research: “concept drift”
  - Drift in the target variable predicted by a ML model (Widmer & Kubat 1996)

- Data quality (DQ) research
  - A lot of research into DQ dimensions (cf. Wang & Strong 1996)
  - Related DQ terms:
    - “identity” and “rigity”, referring to the stability of a variable (Guarino and Welty 2002)
    - “timeliness”, which describes how current data is for a task at hand (Heinrich et al. 2018)
Industrial Application Scenario (1)

- Austrian manufacturing company
  - Mass production of plastic and multi-material parts with injection molding machines
  - Injection molding = complex physical-chemical process

- ML Project with SCCH: monitor the stability of the production process
  - Avoid machine damage & perform countermeasures as early as possible
  - Data-driven solution to ensure production quality using …
    - Stream data processing, classical ML algorithms, outlier detection, causal discovery, etc.
  - Requirement for ML: algorithms expect data to be in a standardized format
Injection molding machines almost exclusively from same vendor
- Shipped with standardized API → high level of data consistency
- Process data logged into the “MES system”

Issues with semantic shift
- There exist different machine types and versions
- Identical machines (same type + version) might still have different firmware
- Variables in process log schema undergo semantic shift (over time)
  - Example: with a firmware update, measurements of pressure sensor are changed from storage in bar to millibar (updated for higher granularity)
- Ignoring semantic shift yields to wrong ML results!
L* System Architecture

- **Message Queue**
- **Stream Engine**
- **Operational Datastore**
- **Data Analysis**
- **Online Datastore**
- **Batch Environment**
- **Reports**

- Kafka
- Spark Streaming
- Cassandra
- R

Feedback

10/Sec

Compressed Archive Files

TTL=7d

Σ
**L* Online Datastore**

- **Apache Cassandra**
  - Column-based
  - Optimized for large amounts of data

```
create table MDavro (  
  jahr int,  
  seriennummer int,  
  interval int,  
  zeitpunkt timestamp,  
  value blob,  
  primary key((jahr, seriennummer, interval), zeitpunkt));
```

```
create table MD (  
  jahr int,  
  seriennummer int,  
  metric text,  
  zeitpunkt timestamp,  
  value text,  
  primary key((jahr, seriennummer, metric), zeitpunkt));
```
**L* Data Processing to handle Semantic Shift**

- **3 Spark jobs**
  - Data preprocessed according to defined rules to handle semantic shift (cf. next slide)

- **Stream engine**
  - Encoded with Apache Avro data serialization
  - LoadMD-Avro receives machine data → decode → store to Cassandra
  - PreProMDStream receives machine data → decode → preprocessing with rules → returns data to Confluent

- **Batch environment**
  - PreProMDBatch reads data from Cassandra (requires start and end point as batch interval) → preprocessing with rules → returns data to Cassandra
<table>
<thead>
<tr>
<th>MD_paramname</th>
<th>Process_paramname</th>
<th>Machine_type</th>
<th>Scale</th>
<th>Offset</th>
<th>Lag</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process_value_1</td>
<td>Mode stopped</td>
<td>T1, T2, T3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Bool</td>
</tr>
<tr>
<td>Process_value_2</td>
<td>Mote Starting</td>
<td>T1, T2, T3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Bool</td>
</tr>
<tr>
<td>Process_value_3</td>
<td>Mode Production</td>
<td>T1, T2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Bool</td>
</tr>
<tr>
<td>Process_value_4</td>
<td>Product Counter</td>
<td>T1, T2, T3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Long</td>
</tr>
<tr>
<td>Process_value_5</td>
<td>Process Temperature 1</td>
<td>T1, T2</td>
<td>1.8</td>
<td>32</td>
<td>0</td>
<td>Float</td>
</tr>
<tr>
<td>Process_value_6</td>
<td>Process Pressure</td>
<td>T1, T2</td>
<td>1.8</td>
<td>32</td>
<td>0</td>
<td>Float</td>
</tr>
<tr>
<td>Process_value_3</td>
<td>Mode Production Phase 1</td>
<td>T3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Bool</td>
</tr>
<tr>
<td>Process_value_7</td>
<td>Mode Production Phase 2</td>
<td>T3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Bool</td>
</tr>
<tr>
<td>Process_value_5</td>
<td>Process Temperature 1</td>
<td>T3</td>
<td>1.8</td>
<td>32</td>
<td>0</td>
<td>Float</td>
</tr>
<tr>
<td>Process_value_6</td>
<td>Process Temperature 2</td>
<td>T3</td>
<td>1.8</td>
<td>32</td>
<td>0</td>
<td>Float</td>
</tr>
<tr>
<td>Process_value_7</td>
<td>Process Temperature 1 Previous</td>
<td>T3</td>
<td>1.8</td>
<td>32</td>
<td>1</td>
<td>Float</td>
</tr>
<tr>
<td>Process_value_8</td>
<td>Process Pressure</td>
<td>T3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Float</td>
</tr>
</tbody>
</table>
Deployment in productive environment → handle Big Data

Performance evaluation: 28.8 million records

Avg.: 42.2 measurement values / record

<table>
<thead>
<tr>
<th>Spark Data Stream</th>
<th>Unit</th>
<th>Throughput (unit/sec)</th>
<th>Storage (byte/unit)</th>
<th>Storage (disk space in GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoadMDAavro</td>
<td>Records</td>
<td>358</td>
<td>182</td>
<td>5.01 GB</td>
</tr>
<tr>
<td>PreProMDBatch</td>
<td>Values</td>
<td>174,343</td>
<td>4.6</td>
<td>6.49 GB</td>
</tr>
<tr>
<td>PreProMDStream</td>
<td>Values</td>
<td>4,816</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Outlook

- Data preprocessing system L* to handle semantic shift in data streams
- Rule-based solution most common in DQ tools (cf. Ehrlinger et al. 2019)

Ongoing and future work

- Extend rule-based system with **semantic solution** to achieve a higher degree of automation
- Investigate **DQ assessment for streaming data** from a more general viewpoint → develop DQ metrics specific for data streams
DI Lisa Ehrlinger
Senior Researcher Data Analysis Systems
+43 50 343 836
lisa.ehrlinger@scch.at
www.scch.at
References