Predictor Factory
Temporal Propositionalization of Relational Data

Jan Motl

Seminář strojového učení a modelování

If you have a question, ask during the presentation
Outline

1. Introduction
   1. Problem statement
   2. Motivation
   3. Scope of work
   4. Process flow
   5. Results

2. Temporal relational learning

3. Details

4. Discussion
What does Predictor Factory do?

Introduction

Problem statement

Multiple tables → *propositionalization* → Single table → *modeling tool* → Prediction
Why to Automate Data Preprocessing?

Time spent on data preprocessing (relative to the whole datamining task):

- Skytree: 90%
- Jan Spousta: 90%
- FICO: 80%
- Oracle: up to 80%
- Microsoft: 70-90%
- Petr Máša: 70-80%
- Teradata: 70%
- Data Preparator: 60-80%
- Vladimír Kyjonka: 60-80%
- Dorian Pyle: 60%
- IBM: 40-70%
- David Olson: over 50%
- KXEN: 40-50%
- SAS: 43%

On average, 67% of all the time is spent on data preprocessing.
Data Mining Process

Acquisition | Pre-processing | Modeling | Evaluation | Deployment

Visualization | Aggregation | Feature creation | Sampling | Outlier detection | Missing value treatment | Feature transformation | Feature selection

Predictor Factory
Process Flow

1. Data are in a single database

2. Create a *target table* with:
   1. ID (CustomerID...)
   2. $n$ Labels (what to predict)
   3. $n$ Timestamps (when to perform the prediction)

3. Propagate the *target* data into all tables

4. Calculate features on each table

5. Join features into the output table
## Classification accuracy on 10-CV

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Aleph</th>
<th>Predictor Factory</th>
<th>RelF</th>
<th>RSD</th>
<th>Wordification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carcinogenesis</td>
<td>0.55</td>
<td>0.82±0.07</td>
<td>0.60</td>
<td>0.60</td>
<td>0.80</td>
</tr>
<tr>
<td>CS</td>
<td>–</td>
<td>0.96±0.01</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Financial</td>
<td>0.87</td>
<td>0.87±0.06</td>
<td>0.98</td>
<td>0.95</td>
<td>0.90</td>
</tr>
<tr>
<td>Genes</td>
<td>–</td>
<td>0.62±0.02</td>
<td>–</td>
<td>0.84</td>
<td>–</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>0.78</td>
<td>0.74±0.08</td>
<td>0.69</td>
<td>0.59</td>
<td>0.65</td>
</tr>
<tr>
<td>Mondial</td>
<td>–</td>
<td>0.79±0.05</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MovieLens</td>
<td>–</td>
<td>0.78±0.03</td>
<td>0.61</td>
<td>–</td>
<td>0.83</td>
</tr>
<tr>
<td>Mutagenesis</td>
<td>0.81</td>
<td>0.93±0.08</td>
<td>0.87</td>
<td>0.90</td>
<td>0.82</td>
</tr>
<tr>
<td>NBA</td>
<td>–</td>
<td>0.60±0.02</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NCAA</td>
<td>–</td>
<td>0.70±0.02</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PremierLeague</td>
<td>–</td>
<td>0.67±0.03</td>
<td>–</td>
<td>–</td>
<td>0.35</td>
</tr>
<tr>
<td>Trains</td>
<td>0.70</td>
<td>0.95±0.15</td>
<td>0.75</td>
<td>0.80</td>
<td>0.95</td>
</tr>
<tr>
<td>University</td>
<td>–</td>
<td>0.89±0.04</td>
<td>–</td>
<td>0.37</td>
<td>0.84</td>
</tr>
<tr>
<td>UW-CSE</td>
<td>0.85</td>
<td>0.88±0.15</td>
<td>0.81</td>
<td>0.89</td>
<td>0.89</td>
</tr>
</tbody>
</table>
Outline

1. Introduction

2. Temporal relational learning
   1. Leaking data
   2. Temporal constraint

3. Details

4. Discussion
Time Axis

- Time of prediction
- Observation window
- Time forsaken in the prediction
- Time
Time Axis

- Obsolete data
- Time of prediction
- Observation window
- Time forsaken in the prediction
- Time
Temporal relational learning

Leaking data

Time Axis

Obsolete data

Observation window

Time of prediction

Time forsaken in the prediction

Blackout

Time
Temporal relational learning

Temporal constraint

Contains NULL?

Contains future date?

Count dates that fulfil the constraint on the history length and blackout

Has the highest estimate?

Is estimate > 0?

Is temporal constraint

Is not temporal constraint
Outline

1. Introduction
2. Temporal relational learning
3. Details
   1. Data
   2. Foreign key constraint identification
   3. Feature functions
4. Discussion
RELATIONAL DATASET REPOSITORY

Search for datasets

72 datasets found...

**Accidents**
Traffic accident database consists of all accidents that happened in Slovenia's capital city Ljubljana between the years 1995 and 2005.

- 235.5 MB
- 3 Tables
- Government
- Classification
- Missing values
- Numeric
- String
- Temporal

**AdventureWorks**
Adventure Works 2014 (OLTP version) is a sample database for Microsoft SQL Server, which has replaced Northwind and Pub sample databases that were shipped earlier. The database is about a fictitious, multinational bicycle manufacturer called Adventure Works Cycles.

- 234.6 MB
- 71 Tables
- Synthetic
- Retail
- Regression
- Missing values
- Numeric
- String
- LOB
- Temporal
- Spatial

Link: relational.fit.cvut.cz
Collect metadata

Primary key scoring

Primary key optimization

Relationship scoring

Relationship optimization

Relationships identified

**Precision:** ~85%
**Recall:** ~85%
**Runtime:** ~1 minute

Ask for code if you have an application for FKC identification!
Warning: If you do not have questions, I will bore you with another 23 pages.
Problem decomposition

1. Data storage paradigm (e.g. relational database)
2. Information propagation (e.g. join)
3. Feature function (e.g. avg())
4. Refinement (e.g. discrete optimization)
5. Feature selection (e.g. Chi$^2$)
6. Collection (e.g. single table)
How to represent data?

Expressive power:

- **Logical** (predicate logic)
- **Relational** (relational algebra)
- **Graph** (graph theory)

How to connect the data?

Hierarchy

1. Native
2. Target $\rightarrow$ Non-target
3. Non-target $\rightarrow$ Target

Expressive power*:

Native $\supseteq$ Target $\rightarrow$ Non-target $\supseteq$ Non-target $\rightarrow$ Target

* Assuming definitions without extensions
How to calculate features?

Typology:

1. Unirow/multirow
2. Univariate/multivariate
3. Other samples independent/dependent

Other dimensions: data type...
How to iteratively improve features?

Typology:

1. Discrete optimization
2. Continuous optimization

Other dimensions: constrained...
How to select useful features?

Typology:

1. Filter
2. Wrapper
3. Embedded
How to return the features?

Hierarchy:

1. Single table
2. Multi-view learning
3. Attribute based
Duplicate Feature Detection

1. Similarity matrix \( O(n^2) \)
2. Hash \( O(n) \)
3. Locality-sensitive hashing (LSH) \( O(n) \)

in space & time
Chi$^2_{adj}$ as LSH

1. Invariant to shift
2. Already calculated
   - just put it into hash map \( O(1) \) in time on average
3. Works
   - False positive: 0 out of 19 320 features
   - True positive due to shift: \( \sim 3\% \)
Sampling

Two level:

Exploration: 1000 samples per each label class
(univariate feature selection -> #attributes ignored)

Exploitation: The rest only if needed

Hysteresis:
If the target table contains less than 2000 samples per each label class, skip exploration.

Effect:
1. Identifies “lethal joins” agnostic way (rare but lethal)
2. Identifies good features

Alternatives for top-k selection:
Hidden bipartite graphs (Cao, 2015)
• multi-armed bandit does not bound count of samples
• branch&bound does not perform well in presence of loose bounds
Overlapping Data
Overlapping Data

We suggest using overlapping data, OLS to estimate parameters and using Monte Carlo methods to calculate standard errors when the motivation for using overlapping data is errors in the variables (Harri, 2002).
Length of History Biases Aggregates

With the length of history, the count of events can only grow.

Affects count, sum, min, max, exists, avg,...
Remedies

1. Temporal window of a constant length
   - but each customer may generate events with different rate
2. Constant amount of the last $n$ events
   - but not all customers have a history
3. Bootstrapping (repeatedly take $n$ events)
4. Discounting (e.g. exponential)
5. Pass history length & event count to the model
   - > interactions
In Database Processing

Advantages

• No new hardware
• No new software*
• No integration*
• No user training*
• Built in parallelism*

Disadvantages

• Difficult low-level optimization
• Difficult reuse of current non-SQL codebase
• SQL can become cumbersome if pushed outside of its domain

*Almost...
High Cardinality Nominal Attributes

Distance based solution (Claudia Perlich, 2006)

Scales quadratically (with naïve implementation)

Multinomial Naïve Bayes with Laplace’s correction

A fast approach from text mining (scales linearly)

\[
P(\text{token } j | \text{class } k) = \frac{1 + c_{j|k}}{P + c_k},
\]

\[
c_{j|k} = n_k \frac{\sum_{i:y_i \in \text{class } k} X_{ij}W_i}{\sum_{i:y_i \in \text{class } k} W_i}
\]
## Weight of Evidence (WoE)

\[
\text{WoE}(y,x_i) = \log\left[\frac{p(x_i | y^+)}{p(x_i | y^-)}\right]
\]

<table>
<thead>
<tr>
<th>Range</th>
<th>Bins</th>
<th>Non events</th>
<th>Events</th>
<th>% of Non-Events</th>
<th>% of Events</th>
<th>WOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-50</td>
<td>1</td>
<td>197</td>
<td>20</td>
<td>5%</td>
<td>6%</td>
<td>-0.0952</td>
</tr>
<tr>
<td>51-100</td>
<td>2</td>
<td>450</td>
<td>34</td>
<td>12%</td>
<td>10%</td>
<td>0.2002</td>
</tr>
<tr>
<td>101-150</td>
<td>3</td>
<td>492</td>
<td>39</td>
<td>13%</td>
<td>12%</td>
<td>0.1522</td>
</tr>
<tr>
<td>151-200</td>
<td>4</td>
<td>597</td>
<td>51</td>
<td>16%</td>
<td>15%</td>
<td>0.0774</td>
</tr>
<tr>
<td>201-250</td>
<td>5</td>
<td>609</td>
<td>54</td>
<td>17%</td>
<td>16%</td>
<td>0.0401</td>
</tr>
<tr>
<td>251-300</td>
<td>6</td>
<td>582</td>
<td>55</td>
<td>16%</td>
<td>16%</td>
<td>-0.0236</td>
</tr>
<tr>
<td>301-350</td>
<td>7</td>
<td>386</td>
<td>41</td>
<td>11%</td>
<td>12%</td>
<td>-0.1405</td>
</tr>
<tr>
<td>351-400</td>
<td>8</td>
<td>165</td>
<td>23</td>
<td>5%</td>
<td>7%</td>
<td>-0.4123</td>
</tr>
<tr>
<td>&gt;401</td>
<td>9</td>
<td>184</td>
<td>21</td>
<td>5%</td>
<td>6%</td>
<td>-0.2123</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>3662</td>
<td>338</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
WoE is Prone to Overfitting

1. Group rare values together
2. Laplace correction
3. Leave-one-out
Concept Drift

1. Real concept drift refers to changes in $p(y|X)$. Such changes can happen either with or without change in $p(X)$.

2. Virtual drift happens if the distribution of the incoming data changes (i.e., $p(X)$ changes) without affecting $p(y|X)$. 
The lower-left quadrant represents the case in which the sample at hand is drawn from the population at random, this approach also validates the model's performance. In this case, data for building a model are independent cross-sectional data. While such out-of-sample and out-of-time procedures are impractical to create a model using one data set and then test it on a separate ‘hold-out’ data set composed of completely independent cross-sectional data. However, if the data are impaired and over-fitting becomes likely.

A schematic of the framework is shown in Figure 3.
Histogram Comparison

Intersection: 0.66

\[ p(y_i|x) \]
Concept Drift Adjusted $\chi^2$

$\chi^2_{\text{adj}} = \chi^2 * \text{Intersection},$

$\text{Intersection} \in <0,1>$
Temporal Propositionalization

Propositionalization of relational data stored in relational databases with the focus on the *retail & financial domain*. These data are distinct from static data by the inclusion of *temporal* attributes. The propositionalization is designed for purposes of predictive analysis, namely computation of propensity to buy and propensity to churn. These are *classification “multiple snapshots, multiple entities”* tasks distinct from time-series and sequence learning.