Predictor Factory

Temporal Propositionalization of Relational Data

Jan Motl

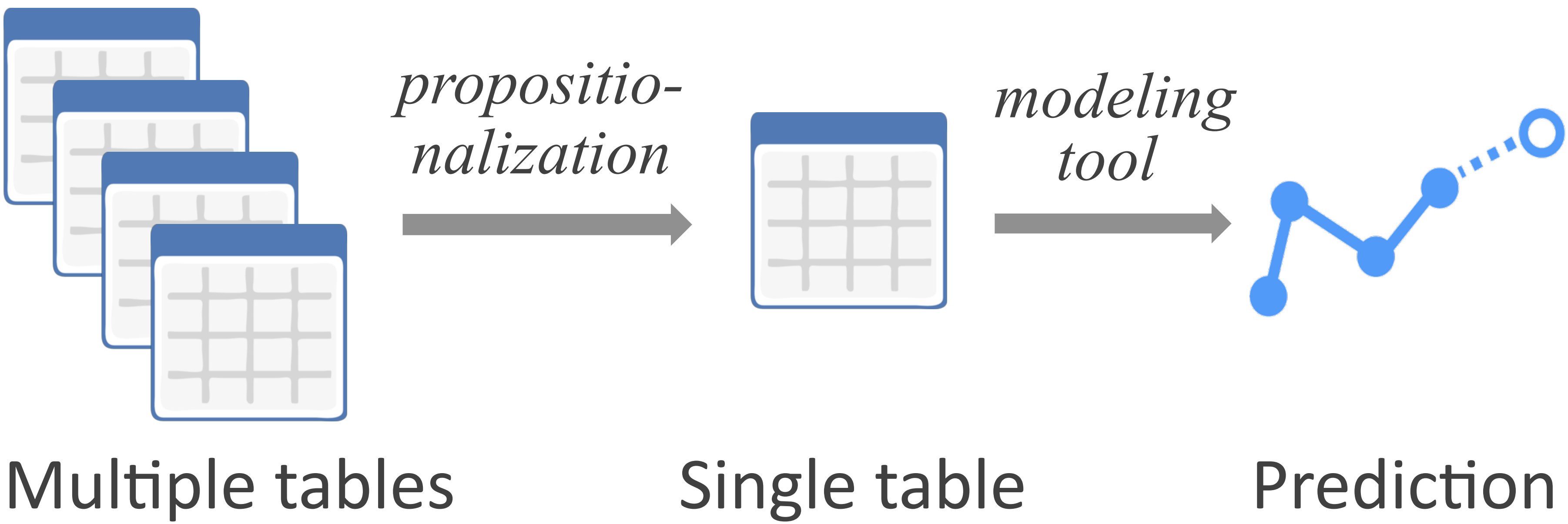
Seminář strojového učení a modelování

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?



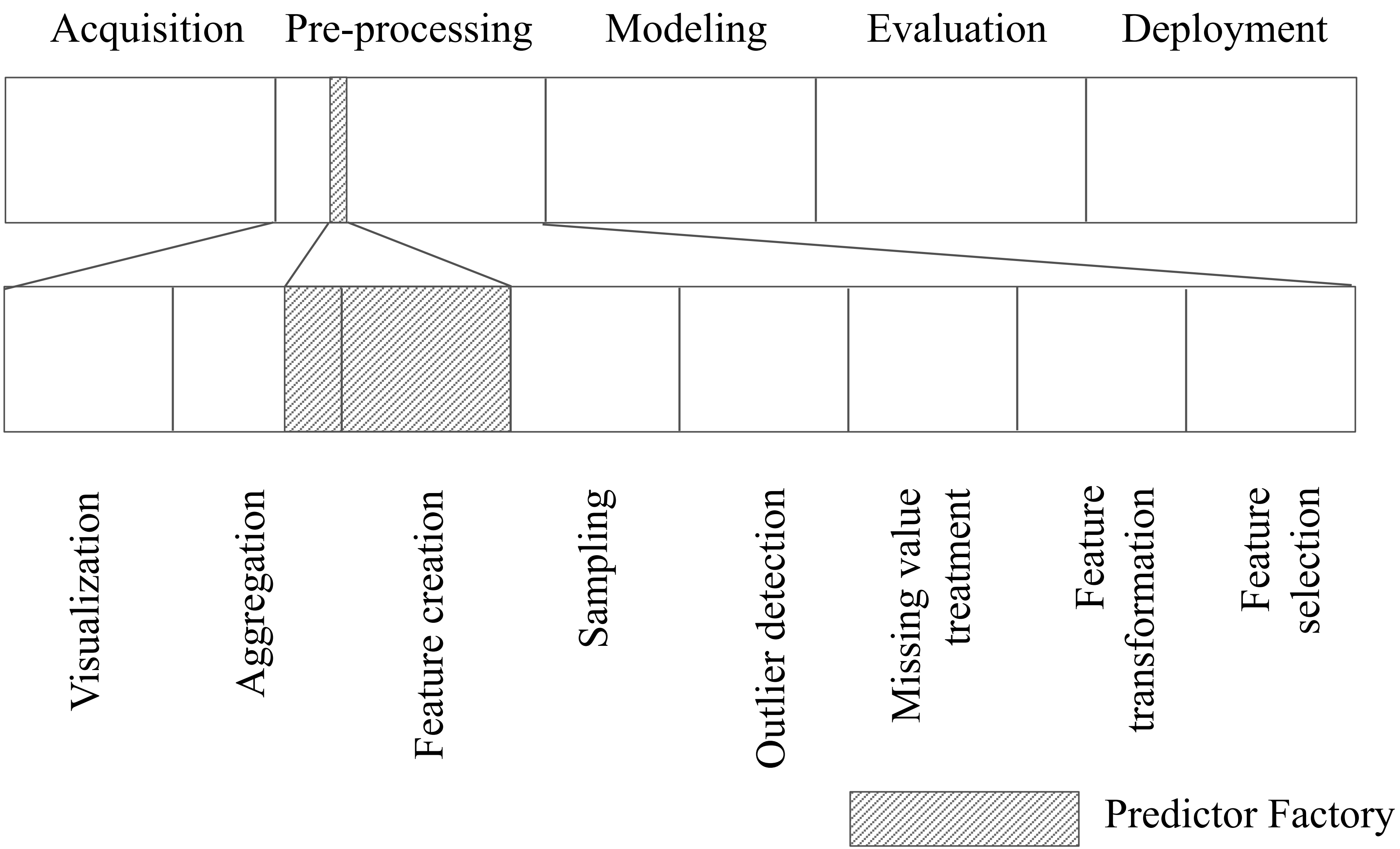
Why to Automate Data Preprocessing?

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



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

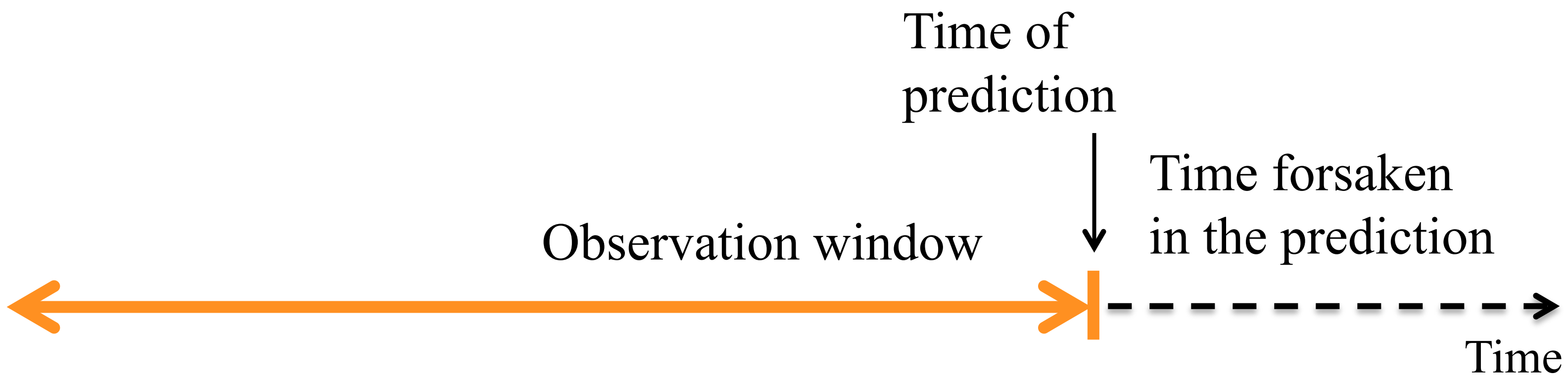
Classification accuracy on 10-CV

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

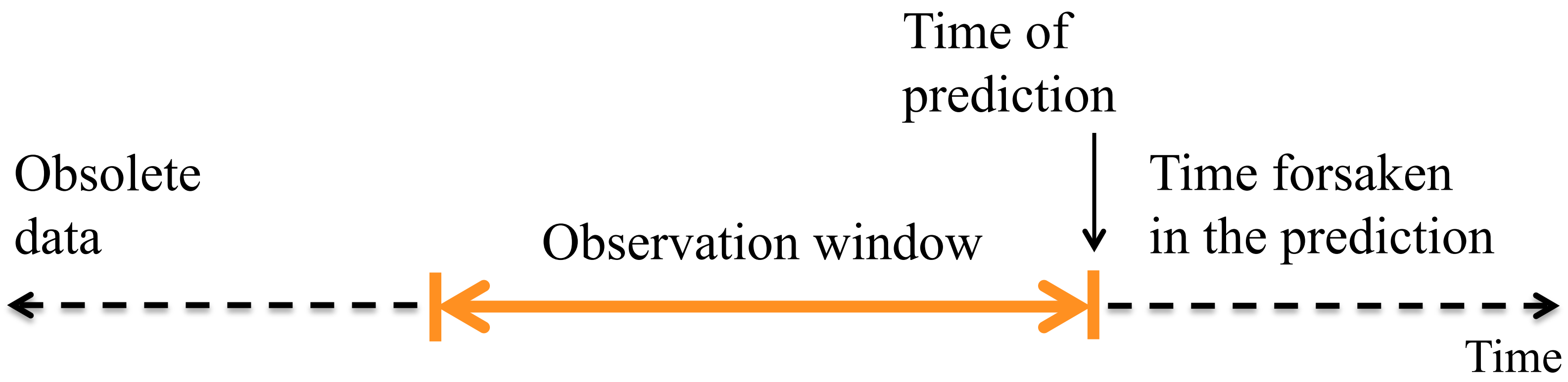
Outline

1. Introduction
2. **Temporal relational learning**
    1. Leaking data
    2. Temporal constraint
3. Details
4. Discussion

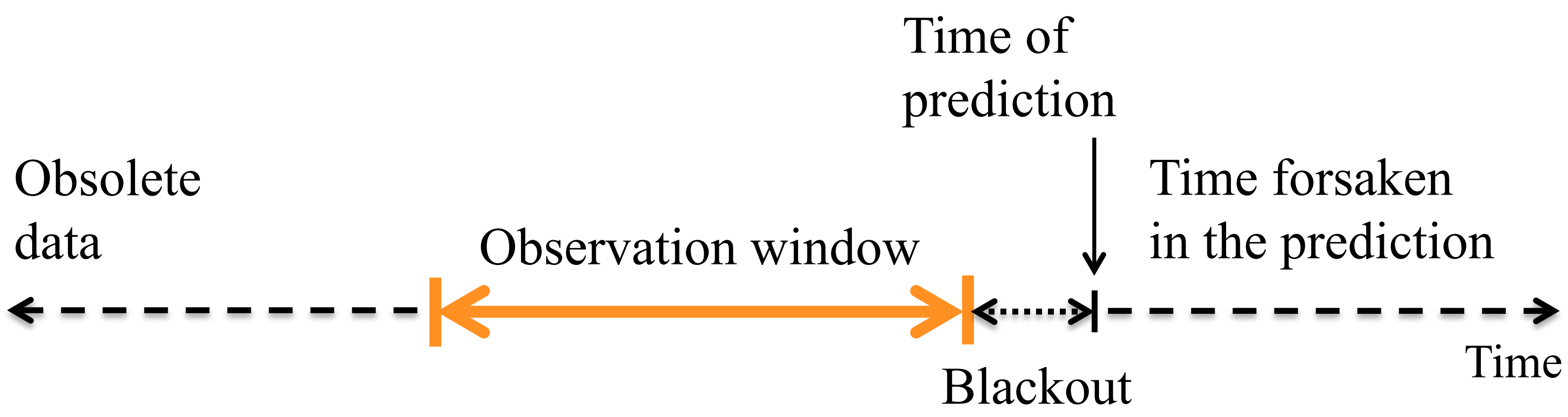
Time Axis

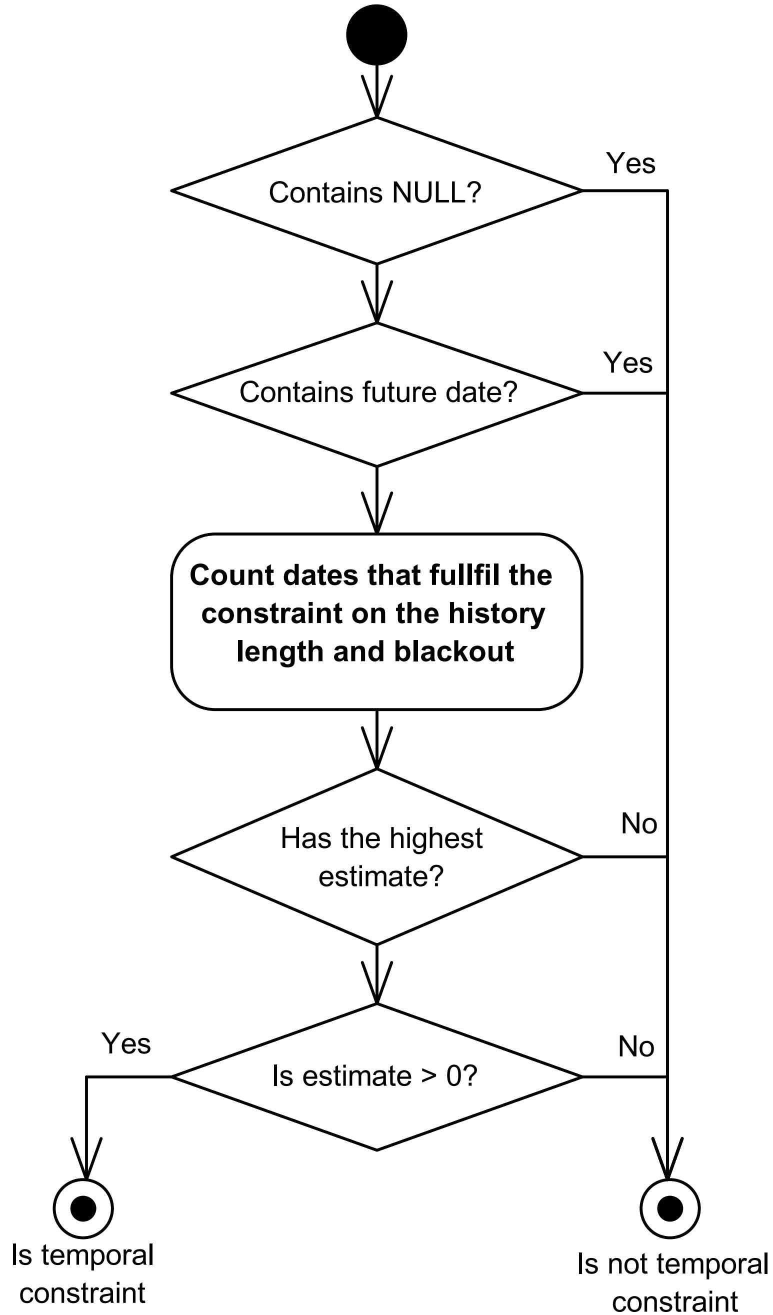

Time of
prediction
Time forsaken
in the prediction
Observation window
Time

Time Axis


Time of
prediction
Obsolete
data
Observation window
Time forsaken
in the prediction
Time

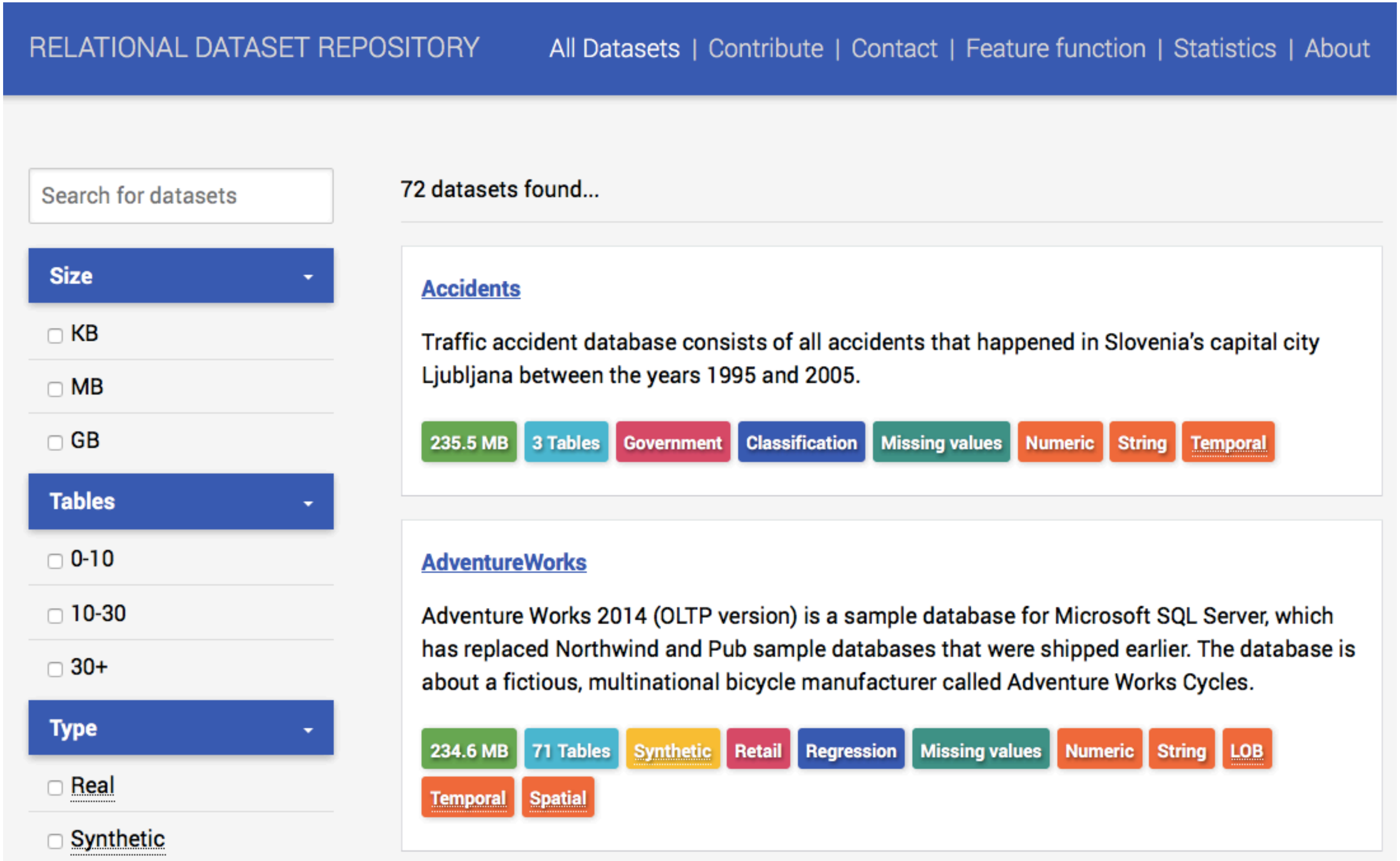
Time Axis


Time of
prediction
Obsolete
data
Observation window
Time forsaken
in the prediction
Blackout
Time

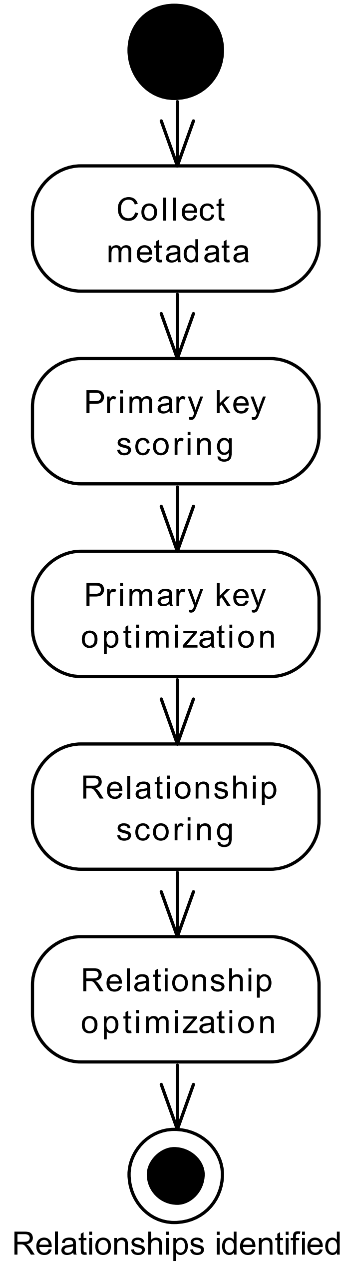

Contains NULL?
Yes
Contains future date?
Yes
Count dates that fullfil the constraint on the history length and blackout
Has the highest estimate?
No
Is estimate > 0?
Yes
No
Is temporal constraint
Is not temporal constraint

Outline

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

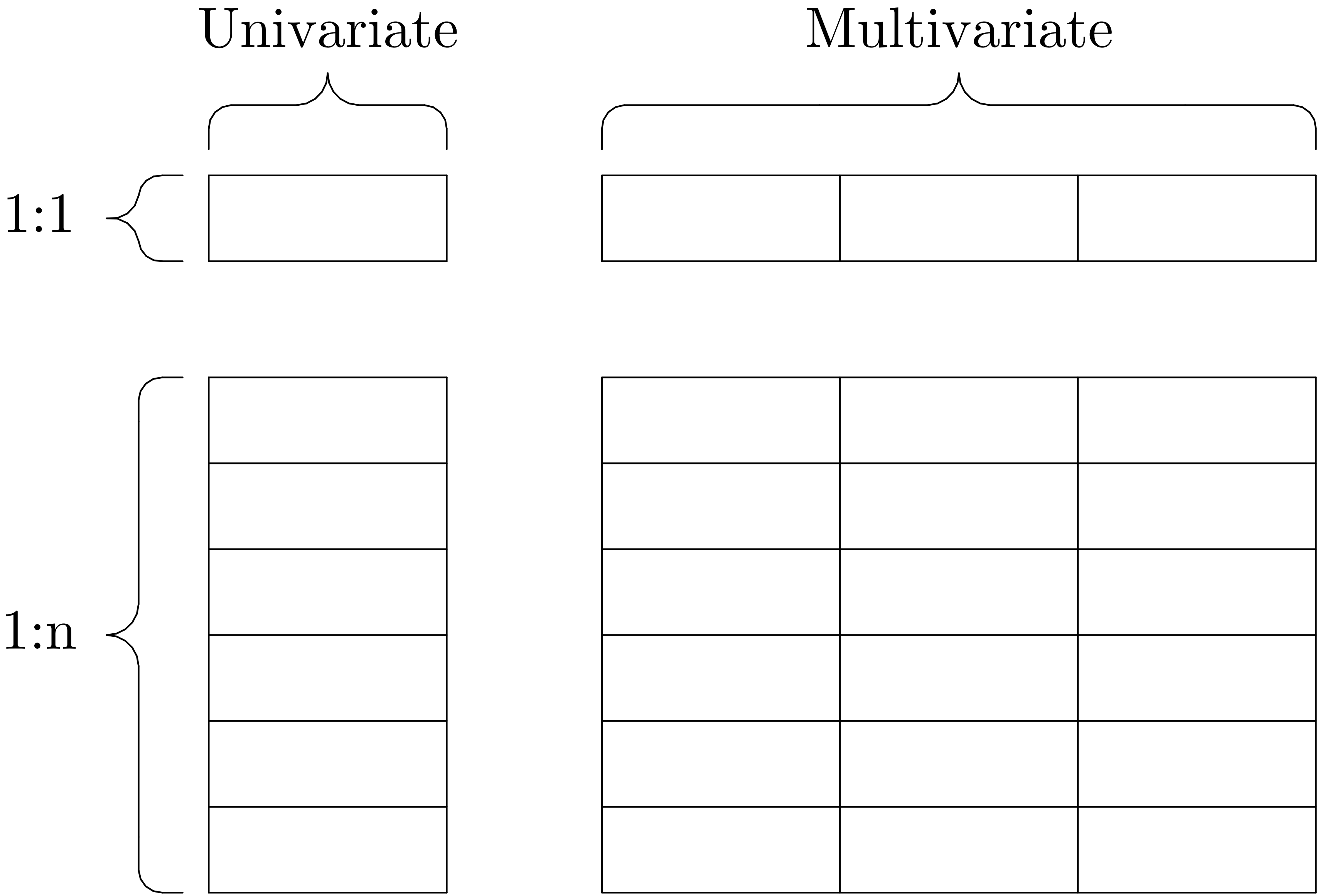


Link: relational.fit.cvut.cz



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

Ask for code if you have an application for FKC identification!


Univariate
Multivariate
1:1
1:n

Discussion

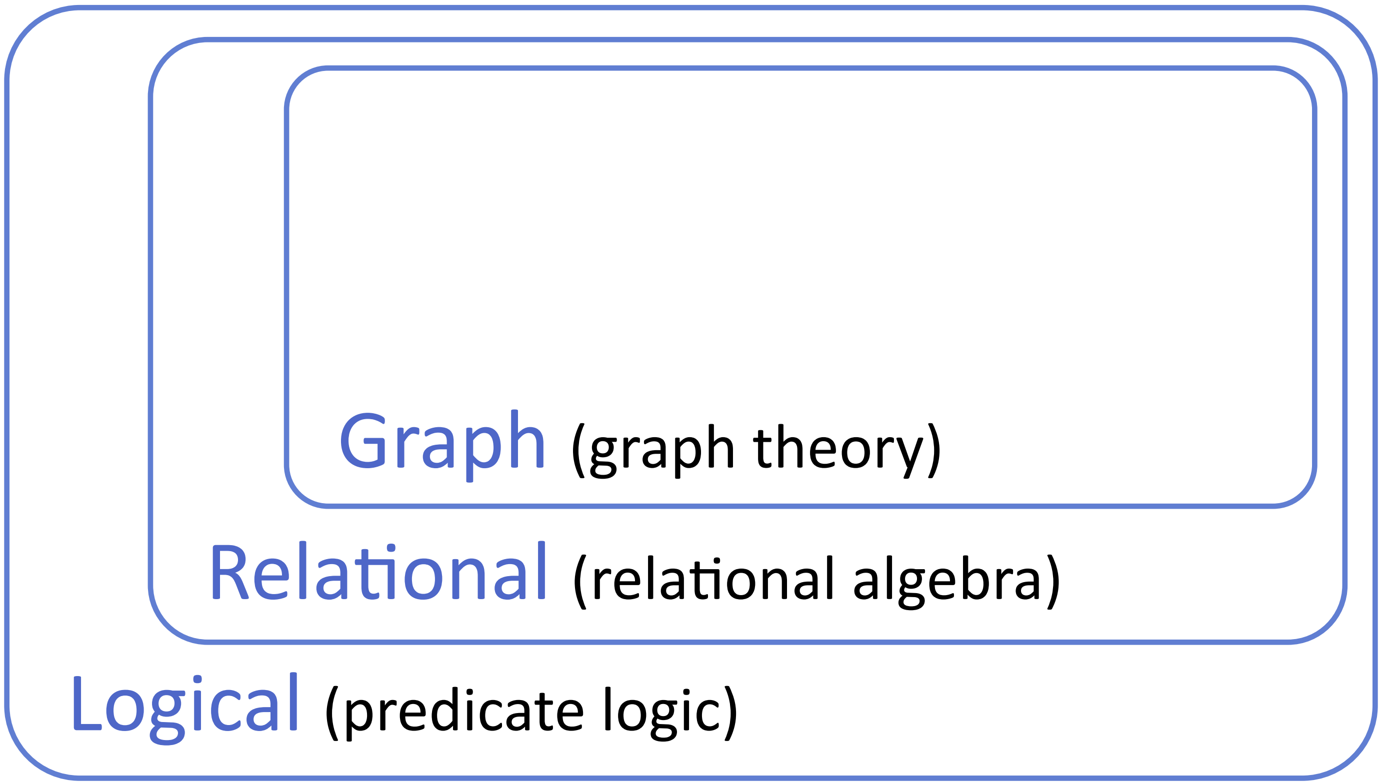
Warning: If you do not have questions,
I will bore you with another 23 pages.

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:



Reference: Dantsin E., Complexity and Expressive Power of Logic Programming, 1997

How to connect the data?

Hierarchy

1. Native
2. Target → Non-target
3. Non-target →Target

Expressive power*:

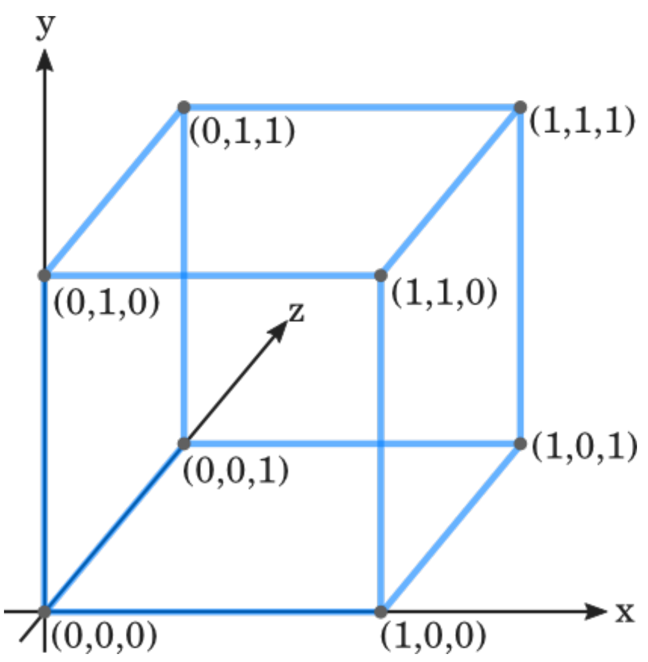
Native ⊇ Target → Non-target ⊇ Non-target →Target

* Assuming definitions without extensions

How to calculate features?

Typology:

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

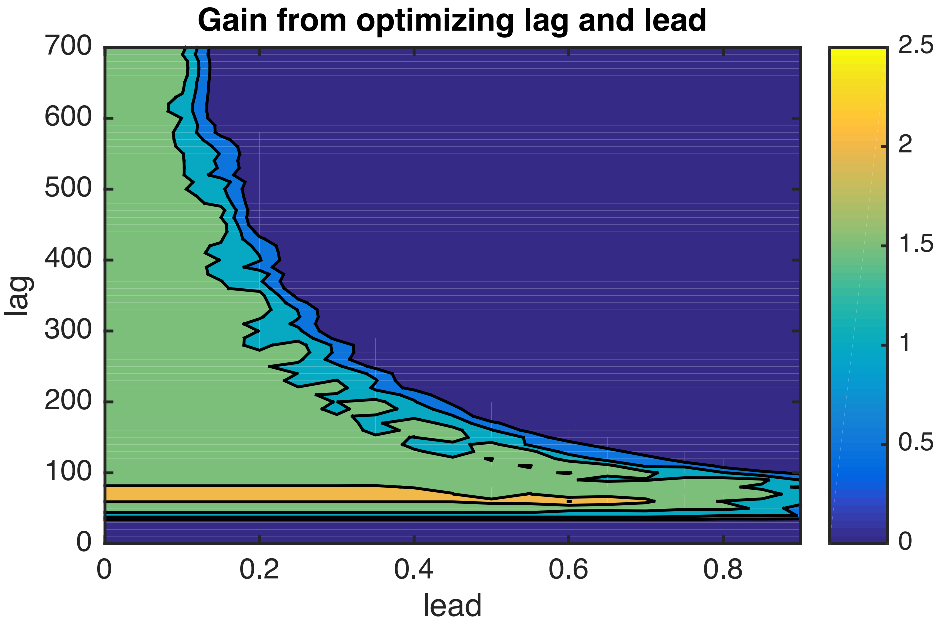


Other dimensions: data type…

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: ~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:

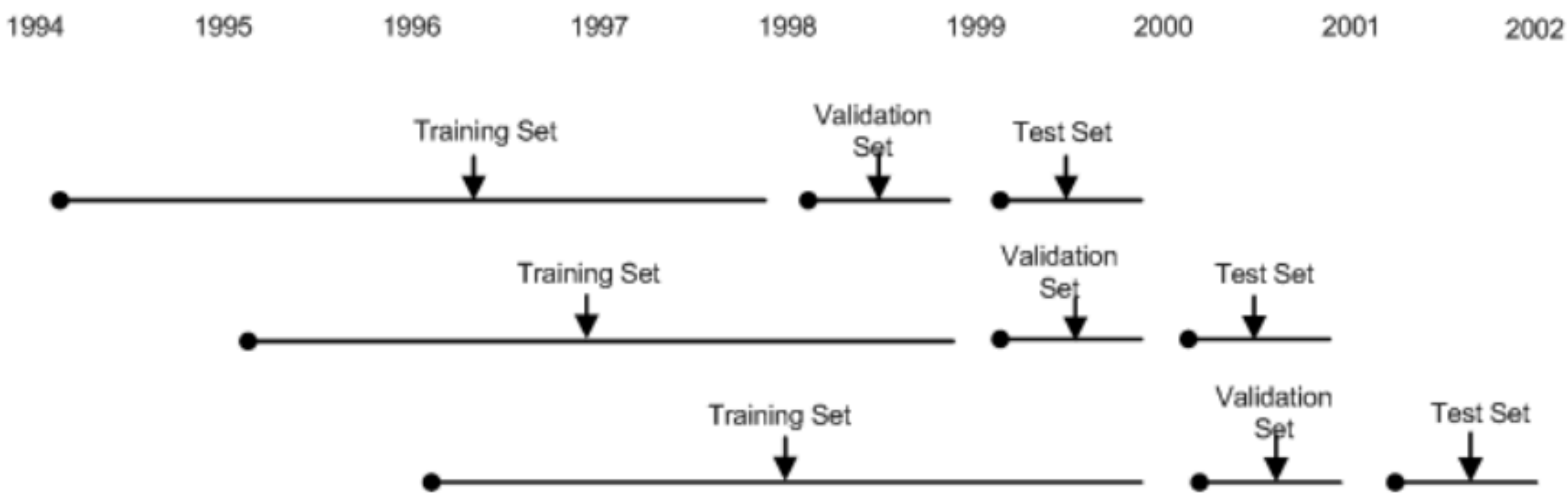
1. Identifies "lethal joins" agnostic way (rare but lethal)
2. Identifies good features

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*

Deployment

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)

$$WoE(y,x_i) = \log[p(x_i|y^+)/p(x_i|y^-)]$$

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

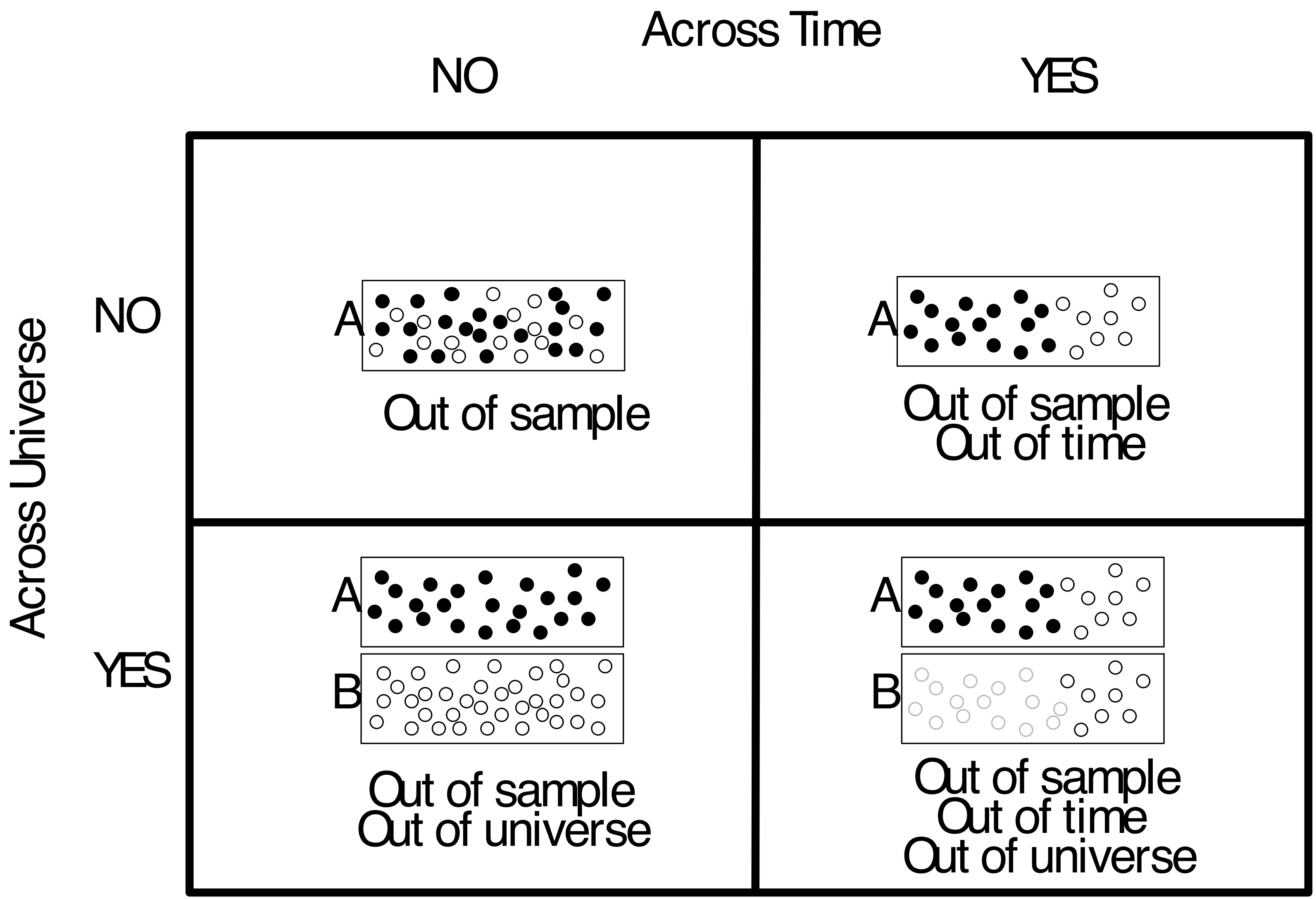
WoE is Prone to Overfitting

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

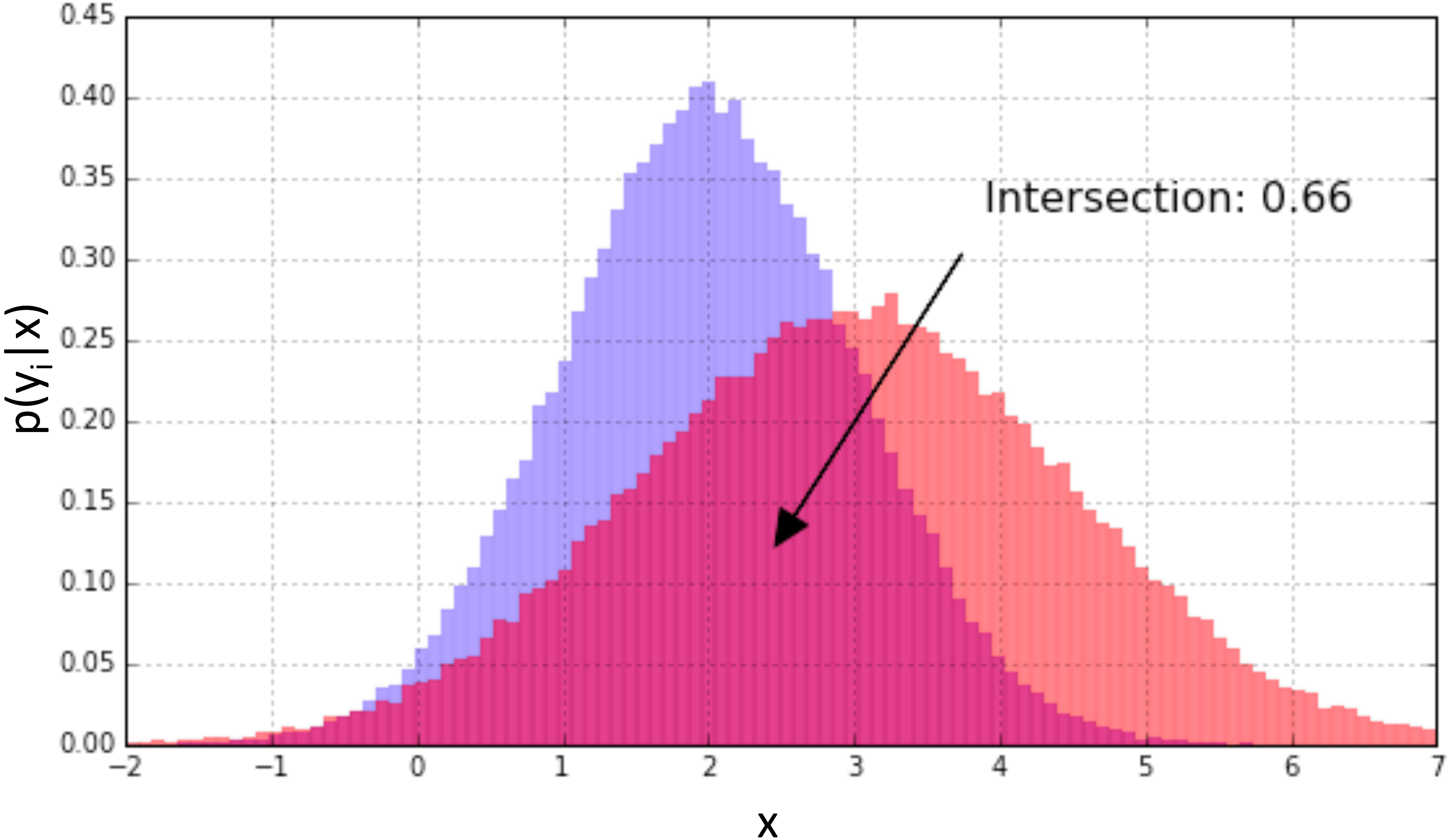
Concept Drift

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).

Training/Testing Split



Histogram Comparison



Concept Drift Adjusted Chi^2

$$Chi^2_{adj} = Chi^2 * Intersection,$$

$$Intersection \in \langle 0,1 \rangle$$

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.