THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

A Toolbox for Learning from Relational Data with Propositional and Multi-Instance Learners

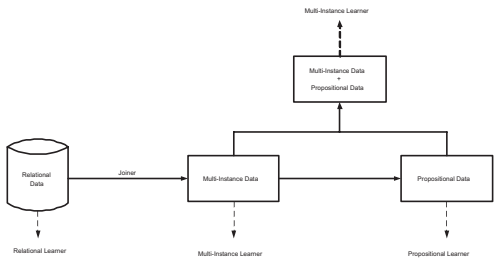
Peter Reutemann, Bernhard Pfahringer and Eibe Frank

Motivation

- **Current data storage**
  predominantly in relational databases
  (SQL databases like MySQL or Oracle)
- **Propositional learners**
  widespread and well-understood
  BUT
  cannot access all data, just one table at a time
  ➔ Tools necessary to work directly on databases

The **Proper Toolbox** represents an attempt to develop a general framework for algorithms to generate propositional and multi-instance data

What data can be produced?



What tools/algorithms are used?

- **Propositionalization**
  - modified version of **RELAGGS** [1]
    calculates aggregates of tables adjacent to the target table
  - **Joiner**
    flattens tree structure of tables into single table
  - **REMILK**
    combines the data produced by RELAGGS and Joiner
- **Learners**
  - **WEKA** (Workbench for Propositional Learners)
  - **MILK** (Multi-Instance Learning Kit)
    using the MIWrapper [2], a meta-scheme for propositional learners

RELAGGS

- uses SQL aggregate functions like SUM, MIN, MAX, AVG and computed standard deviation, quartile and range to capture relational information
- for each value of a nominal column a new attribute is introduced, containing the number of occurrences
- pairs of attributes (one is nominal) are used as GROUP BY conditions for additional aggregations
- determines relations between tables based on name of primary key

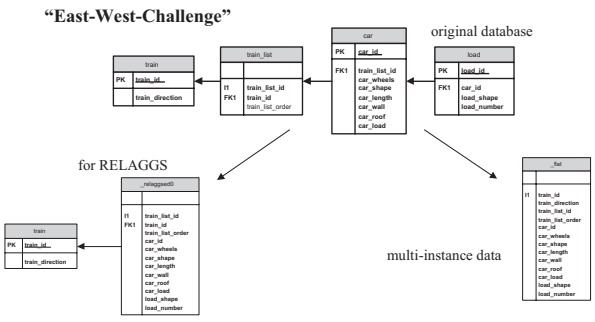
RELAGGS - Limitations

- expects primary key to be integer, which conflicts with chemical domains like mutagenesis where alpha-numeric compound ID functions as key
  ➔ extra table for relation between old key and new integer key
- chemical domains may use the compound ID for defining relationships, but a compound can have several substructures
  ➔ must be INDEX; joins may work differently over indices, which makes a primary key necessary
- NATURAL JOIN can cause loss of information with Prolog data due to closed world assumption
  ➔ LEFT OUTER JOIN necessary
- works only on adjacent tables, possibly missing important data
  ➔ pre-flattening of table structure to access all data

Joiner

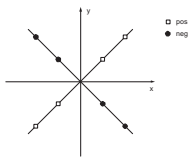
- flattens a tree-structure of related tables into a single table
- can discover relations-tree automatically or process user-defined one
- works in depth-first manner
- limits IO operations by joining small tables first
- join arguments are common columns between tables
- LEFT OUTER JOIN is used, in order not to loose data
- NULL values introduced during join can be set to specific values
- used as pre-processing tool for RELAGGS and to produce multi-instance data

Joiner - Example



MIWrapper

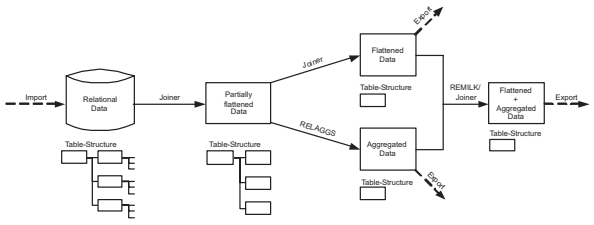
- Joiner can completely flatten data to produce single table (MI data)
- meta-scheme for propositional Learners
- assigns each of the *n* instances in a bag a weight of *1/n*
  ➔ all bags equally weighted
- bag class label is determined by taking the average probabilities over all predicted class labels (by the propositional Learner) in the bag
- advantage over RELAGGS if interactions between attributes important:



MIWrapper	RELAGGS
x < 0	
| y < 0 : pos (4/0)	
| y >= 0 : neg (4/0)	: neg (4/2)
x >= 0	
| y < 0 : neg (4/0)	
| y >= 0 : pos(4/0)	

Artificial dataset and unpruned decision trees for this data

The complete system



Data can be imported as Prolog or CVS, and is exported as an ARFF file for WEKA and MILK

Experiments

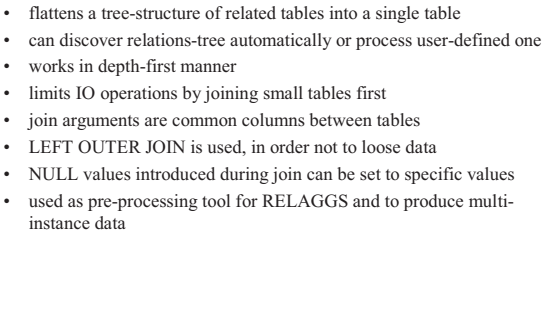
- **Datasets**
  Alzheimer's disease, Drug-Data (pyrimidine + triazine), East-West-Challenge, Genes (KDD01), Musk1/2, Mutagenesis, Secondary structure prediction of Proteins, Suramin analogues
- **Base learning algorithms**
  unpruned decision trees, except for genes_growth where we use LogitBoost/DecisionStump (tree grows too big for memory otherwise)
- **Validation**
  10 runs of 10-fold Crossvalidation, except for suramin/eastwest where we use Leave-One-Out (due to small dataset size)

Detailed comparison of approaches

Dataset	RELAGGS			MILK			REMILK		
alzheimer_amine_uptake	87.59	±	4.31	73.35	±	5.42	87.26	±	4.52
alzheimer_choline	89.18	±	2.73	79.17	±	3.68	89.45	±	2.68
alzheimer_scopolamine	87.84	±	4.23	74.16	±	5.30	87.78	±	4.33
alzheimer_toxic	92.93	±	2.74	88.77	±	3.48	92.39	±	3.00
dd_pyrimidines	92.47	±	2.02	92.46	±	1.94	92.46	±	1.94
dd_triazines	74.76	±	0.85	74.78	±	0.85	74.78	±	0.85
eastwest	80.00	±	41.03	55.00	±	51.04	75.00	±	44.42
genes_growth	31.70	±	1.60	34.00	±	1.01	33.36	±	1.25
genes_growth_bin	84.14	±	0.42	84.33	±	0.22	84.34	±	0.40
genes_nucleus	76.06	±	2.15	57.22	±	2.19	62.55	±	2.03
genes_nucleus_bin	87.28	±	1.39	74.13	±	1.67	78.49	±	1.63
musk1_rel	80.13	±	15.21	81.64	±	14.68	79.50	±	14.58
musk2_rel	72.83	±	13.44	76.82	±	12.61	74.40	±	13.73
mutagenesis3_atoms	79.58	±	8.90	81.86	±	8.23	80.15	±	9.24
mutagenesis3_bonds	85.38	±	7.85	83.62	±	7.87	86.68	±	7.46
mutagenesis3_chains	85.31	±	7.83	84.54	±	7.00	84.85	±	8.32
proteins	59.52	±	4.08	59.12	±	5.08	59.92	±	3.38
suramin	45.45	±	52.22	63.63	±	50.45	45.45	±	52.22

Comparison of accuracy and standard deviation

Graphical comparison of approaches



Graphical comparison of the three approaches

Generated datasets

Dataset	Classes	Attributes	Records	Inst./Bags
alzheimer_amine_uptake	2	237/62/298	686	686
alzheimer_choline	2	251/70/320	1326	1326
alzheimer_scopolamine	2	237/60/296	642	642
alzheimer_toxic	2	251/70/320	886	886
dd_pyrimidines	2	95/90/184	1762	1762
dd_triazines	2	125/118/242	23650	23650
eastwest	2	66/26/91	20/213/213	20
genes_growth	13	27/49/138	4346/14238/14238	4346
genes_growth_bin	2	27/49/138	4346/14238/14238	4346
genes_nucleus	15	27/49/134	4346/14238/14238	4346
genes_nucleus_bin	2	28/49/134	4346/14238/14238	4346
musk1_rel	2	1661/168/1828	92/476/476	92
musk2_rel	2	1661/168/1828	102/6598/6598	102
mutagenesis3_atoms	2	26/12/37	188/1618/1618	188
mutagenesis3_bonds	2	56/18/73	188/3995/3995	188
mutagenesis3_chains	2	88/26/113	188/5349/5349	188
proteins	2	22/22/43	1612	1612
suramin	2	119/22/140	11/2378/2378	11

Overview of the generated datasets, each column shows values for RELAGGS/MILK/REMILK

Interpretation of results

- all three approaches produces similar results for most datasets
- RELAGGS and REMILK almost identical in most of the cases (except in *genes_nuclues_**, possibly because propositional attributes follow after MI ones and decision tree is biased towards MI)
- MILK, i.e. MIWrapper applied to MI data, performs as well as the other approaches on two-thirds of datasets, sometimes slightly better (better performance of RELAGGS on *alzheimer* datasets is due to the fact that it treats NULL values as separate values and not as missing)
- RELAGGS uses the least amount of memory for "true" MI data (*alzheimer* data is propositional), otherwise MILK is better due to fewer columns (no aggregates). REMILK always needs the biggest amount, because it is the combination of both
- generating MI data faces the problem of tables growing too large during joins

Conclusions and future work

Conclusions

- Proper is a practical database-oriented framework
- allows easy integration of other propositionalization algorithms

Future work

- algorithmic improvements for more efficiency
- instead of generating all data before-hand ("bottom-up"), final tuples could be generated one after the other and fed into an incremental Learner ("top-down") ➔ less memory consumption
- replacing expensive joins of tables with propagating only keys

Key references
1. M.-A. Krogel and S. Wrobel. *Facets of Aggregation Approaches to Propositionalization.* In: T. Horváth and A. Yamamoto (eds.) Proceedings of the Work-in-Progress Track at the Thirteenth International Conference on Inductive Logic Programming (ILP), 2003.
2. E. Frank and X. Xu. *Applying Propositional Learning Algorithms to Multi-instance data*. Working Paper 06/03, Department of Computer Science, University of Waikato. 2003.