Fast Relational Learning Using Bounded LGG

Author: Andrea Fúksová 1, Advisor: Ondřej Kuřelka 2
1 Czech Technical University in Prague, 2 KU Leuven

Problem
Sometimes, data expressed by relations make better sense than expressed as vectors of real numbers.

Relational machine learning:
- Subfield of machine learning
- Learning from structured data
- Structures encoded as:
  - Labelled graphs
  - First order logic clauses
  - Relational structures
- So far, most theory based on first order logic formulation (FOL)

Preliminaries
Def. 1. Vocabulary σ is a finite set of relation symbols with associated arities.

Def. 2. Relational structure A of type σ is a pair of universe U A and a sequence of relations R A. There exists one relation R n ∈ R A for each R ∈ σ with the same arity as R.

Def. 3. A homomorphism f from a structure A to a structure B of the same type is a mapping f : U A → U B such that for every m = arg R ∈ σ and every (a 1, ..., a m) ∈ R A we have (f(a 1), ..., f(a m)) ∈ R B. If this homomorphism exists, we denote it by A → B.

Def. 4. A relational structure C is said to be a least general generalization (LGG) of the relational structures A and B if and only if C → A and C → B and for every other relational structure D such that D → A and D → B it holds D → C.

Goal
- Input: sets E + and E − of positive and negative examples
- Examples are relational structures
- Find a classifier: set S of relational structures
- Structure e classified as:
  - positive if ∃s ∈ S : s → e
  - negative otherwise
- If s → e, we say that s covers e

Principle
- Learning based on application of LGG on positive examples
- Homomorphism can be formulated as a Constraint satisfaction problem (CSP)
- Deciding about homomorphism for two structures is NP-complete
- Basic algorithm for finding LGG produces very large structures which need to be reduced
- Reduction without generality loss: find smallest homomorphically equivalent structure
- Result: Basic learning requires a lot of computationally costly homomorphism tests
- Idea: Exploiting polynomial-time local consistency techniques from CSP to test so called bounded homomorphism

Some results
- Effective implementation of in general exponential-time methods based on complete CSP solution is usually faster than solution based on polynomial-time bounded operations (exploiting local consistency techniques)
- Results comparable in accuracy with state-of-the-art algorithms for relational machine learning
- Figure shows accuracy performance of my implementation and state-of-the-arts algorithms on eight datasets.
- Every algorithm has its own color

My work
- Reformulation of theory from FOL into terms of relational structures.
- This formulation should be more accessible for most scientific audience as opposed to FOL
- Effective and complex implementation of the studied methods in Java
- Implementation of a new effective CSP solver
- Investigation of runtime and accuracy performance of the methods

Example
- Results on dataset containing 80 Hexose-binding protein domains (positive examples) and 80 non-Hexose-binding protein domains (negative examples).
- Presented at the workshop Machine Learning in Computational Biology at the conference NIPS 2013
- Equivalent encoding as labelled graphs
- One vertex for every atom (labelled by the atom type + position in the amino acid)
- Edge labelled by a discretized distance (if < 4 Angstroms)
- 10-fold cross-validation accuracy 71.9 ± 5.3
- Picture: structure covering covers 89 positive examples and no negative example

Author: Andrea Fúksová
1 Czech Technical University in Prague, 2 KU Leuven

Vocabulary
- τ is a finite set of relation symbols with associated arities.
- For each relation symbol R ∈ τ, there exists a relation R A in A.
- A homomorphism f : A → B is a mapping f : U A → U B such that for every m = arg R ∈ τ and every (a 1, ..., a m) ∈ R A we have (f(a 1), ..., f(a m)) ∈ R B in B.
- An LGG of A and B is a relational structure C such that C → A and C → B and for every other relational structure D such that D → A and D → B it holds D → C.

Structure
- A structure is a pair (U, R), where U is a set of elements and R is a set of relations.
- A relation is a subset of the Cartesian product of the elements of its domain.
- A homomorphism is a function that preserves the structure of the relations.

Classification
- A classifier is a set of relational structures that are used to predict the class of a given example.
- A classifier is trained on a labeled dataset, where each example is associated with a label from a finite set of possible labels.
- The classifier is then used to predict the label of new, unseen examples.

Accuracy
- The accuracy of a classifier is the proportion of correctly classified examples in the test set.
- The accuracy is calculated by dividing the number of correctly classified examples by the total number of examples in the test set.

Complexity
- The complexity of a classifier is a measure of the computational resources required to train and use the classifier.
- The complexity is often measured in terms of the time and space required to perform the necessary calculations.

Implementation
- The implementation of the classifier is based on the CSP approach.
- The CSP is formulated such that the homomorphism problem is expressed as a set of constraints.
- The CSP is solved using an efficient algorithm that exploits the local consistency techniques.
- The implementation is implemented in Java.

Evaluation
- The performance of the classifier is evaluated on a dataset of 80 Hexose-binding protein domains and 80 non-Hexose-binding protein domains.
- The dataset is divided into a training set and a test set.
- The classifier is trained on the training set and tested on the test set.
- The accuracy of the classifier is calculated on the test set.
- The results are provided in the form of a bar chart and a table.
- The chart shows the accuracy of the classifier compared to state-of-the-art algorithms on eight datasets.
- The table provides a detailed breakdown of the results for each dataset.

Conclusion
- The implementation of the classifier is effective and complex.
- The classifier is shown to be comparable in accuracy with state-of-the-art algorithms for relational machine learning.
- The implementation is available in an open-source repository.

Acknowledgments
- The author would like to thank the anonymous reviewers for their careful reading and constructive comments.
- The author is grateful to the organizers of the NIPS 2013 workshop for the opportunity to present the work.
- The author acknowledges the support of the Czech Technical University in Prague and the KU Leuven.