Merging of Biomedical Decision Decision Diagrams

Christoph Redl

1. Decision Diagrams in Medicine
Decision diagrams are a common aid for decision making
▶ Intuitively understandable
▶ Not only for professionals in information engineering

Medical Examples
▶ Staging of tumor diseases
▶ Severity appraisal in clinical practice
▶ Therapy selection depending on patient’s conditions
▶ Multidimensional indices in case-based reasoning systems

Other Life Sciences and Related Fields
▶ Molecular biology: DNA classification
▶ Psychology: tests for diagnosis of personality disorders

Even Beyond
▶ Numerous applications in economy (e.g., liquidity appraisal)

2. Multiple Decision Diagrams
Sometimes we have more than one diagram with similar content.

Reasons
▶ Different expert opinions
▶ Different training sets (e.g., due to parallel computing settings where each cell works on a subset of the complete training set)
▶ Randomized machine-learning algorithms
▶ Several research groups work on similar projects and come to slightly differing results due to statistical fluctuations

3. Existing Solutions have Drawbacks
Ensemble Learning methods
▶ The object to classify is put into each of the decision diagrams.
▶ Afterwards the individual results are united, e.g., by majority voting.

Disadvantages
▶ Still multiple diagrams: Only the results are merged, but not the diagrams themselves
▶ No compact representation

Other existing solutions
▶ Mostly assume that original training data is still available
▶ Often convert diagrams into sets of rules before merging

4. New Approach
Incorporating of several decision diagrams into a single one
▶ without referring to original training data
▶ without translating them into rules

Technology
▶ dlvhex and mergingplugin (a dlvhex extension) [1]
▶ user-defined operators ensure applicability in many scenarios

Benefits
▶ Easy to experiment with several merging techniques and make comparisons without repeating manual incorporation.

5. Processing Decision Diagrams with dlvhex
1. Input: Human-readable DOT graphs
2. Automatic translation into a machine-readable set of facts
3. Application of algorithms for decision diagram merging
4. Output as answer set and back-translation into a DOT graph
5. Visualization using the DOT tools

6. Case Study: DNA-Classification
Is a sequence over \{A, C, G, T\} protein-coding or junk?

1. Training
Traditional approach
▶ Use a large set of annotated sequences
▶ Compute 20 numeric features for each sequence
▶ Train a single decision tree over the full training set

Our solution
▶ Train multiple decision trees on subsets of the training set
▶ Aim for a diversity among the classifiers (e.g., by randomization or selection of different algorithms)
▶ Merge the classifiers subsequently

2. Classification: Compute the same features for the query sequence and classify it using the merged decision tree

3. Evaluation: Quality of result depends on settings and algorithms; finding the optimum is supported by our framework.

Interesting observations:
▶ Potential to increase accuracy
▶ By combining diagrams from different learning algorithms, number of necessary training samples decreases
These observations could hardly be made without the framework!

7. Further Aspects
▶ Quality of result depends on: training set, machine-learning algorithms and merging procedure
▶ But: Our framework perfectly supports the user when experimenting and evaluating results!

Advantages
▶ Increase of accuracy
▶ Parallel-computing possible
▶ Easy to change the training set or the merging procedure

References