Concept of rule-based configurator for Auto-WEKA using OpenML

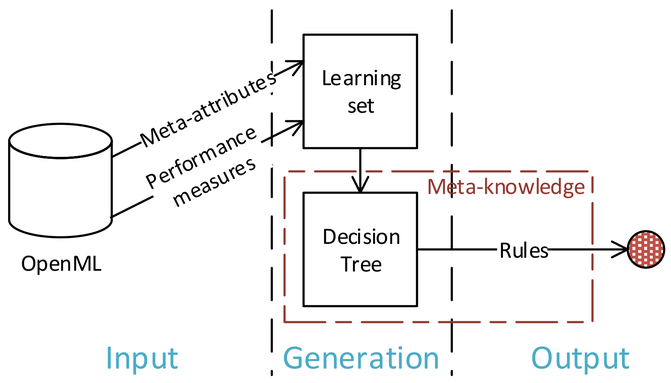
Patryk Kiepas, Szymon Bobek, Grzegorz J. Nalepa
{kiepas, sbobek, gjn}@agh.edu.pl

Contribution

In this poster we describe a mechanism for automatic recommendation of suitable machine learning algorithms and their parameters. This was achieved by use of OpenML database and a rule-based configurator to make an recommendation. Created recommendations are then used to improve Auto-WEKA tool by reducing its search space and in result its computational time [1].

1.Knowledge acquisition

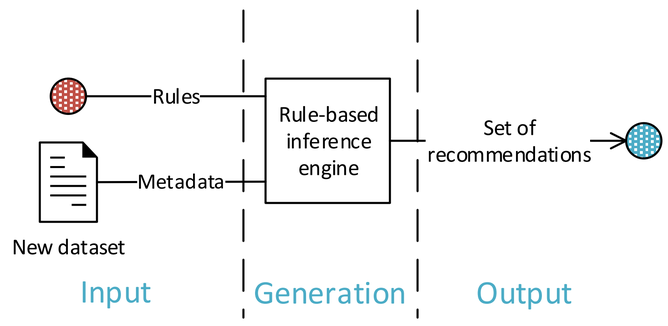
The initial phase is about creating a meta-knowledge that describes dependencies between datasets and performance of machine learning algorithms executed on them.



- Each dataset is described by meta-attributes (e.g. number of classes, attribute entropy)
- Missing meta-attributes are filled with Amelia-II algorithm [2],
- Data about all meta-attributes with corresponding algorithms creates a learning set,
- Building meta-knowledge requires filtering OpenML database described by three parameters (shown in result section),
- Rules are extracted from decision tree created with J48 algorithm on the learning set.

2. Recommendation

In second stage meta-attributes of each new dataset are matched with meta-knowledge in order to build a set consisting of suitable algorithms with their parameters.

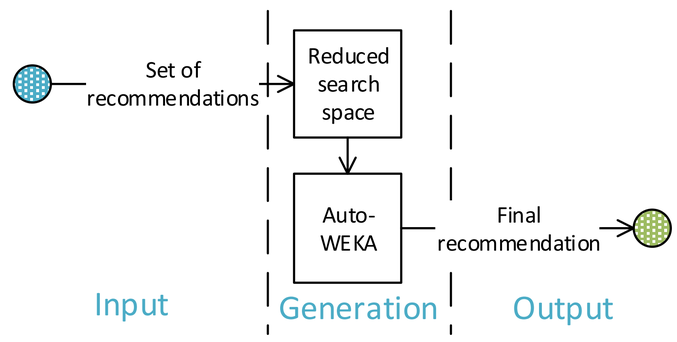


- Configurator is a rule-based inference engine in HeartDroid[a] implementation,
- Inference engine works in fixed-order inference mode,
- Size of set of recommendations depends on number of fired rules by configurator,
- Recommended parameters are fixed within recommended algorithm,
- This phase requires single instance of meta-knowledge.

[a] http://bitbucket.org/sbobek/heartdroid

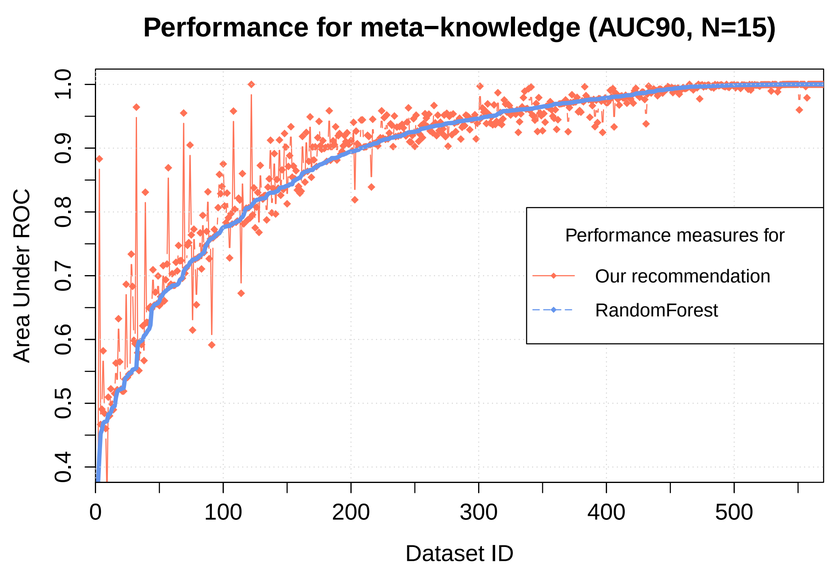
3. Tuning

Tuning phase uses created set of recommendation to reduce search space in hyper-optimization process done by Auto-WEKA tool.



- Auto-WEKA's search space is reduced to recommended algorithms,
- New experiment is prepared with use of BATCH file where list of algorithms to consider and its parameter ranges is storage,
- Output of this phase is a final recommendation consisting of single algorithm with parameters,
- Final recommendation is a result of Auto-WEKA's hyper-parameter optimization,
- Output recommendation can be used directly in WEKA tool.

Results



- Results represents difference in performance of our recommendation for 570 test datasets against performance of *state-of-the-art* algorithm – RandomForest,
- In order to test our system and not Auto-WEKA tool we analyse best recommendations made during second stage,
- Meta-knowledge for this experiment was created with parameters:
  - Selected performance measures: $area\ under\ ROC >= 0.9$,
  - Number of algorithm to consider: 15 most used algorithms from OpenML,
  - Using meta-attributes with less than 20% missing values.
- Results are visibly good where RandomForest is doing poorly (datasets 1-250).

Future work

- Learning and gaining additional meta-knowledge during recommendation phase,
- Improving rules induction from the learning set,
- Adding parameter recommendation in form of value ranges,
- Adding guidance for data preprocessing methods,
- Including more data sources (e.g. MLComp portal, hand-crafted rules),
- Using synthetic datasets for better coverage of meta-attributes space.

References

[1] Thornton, C., Hutter, F., Hoos, H., Leyton-Brown, K.: Auto-WEKA: Combined selection and hyper-parameter optimization of classification algorithms. In: Proc. of KDD-2013. pp. 847–855 (2013)

[2] Honaker, J., King, G., Blackwell, M.: Amelia II: A program for missing data. Journal of Statistical Software 45(7), 1–47 (12 2011), http://www.jstatsoft.org/v45/i07

[3] Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Meta-learning: Concepts and techniques. In: Metalearning: Applications to Data Mining. Springer Publishing Company, Incorporated, 1 edn. (2008)