Choosing the right machine learning method for a given problem

Predicting classification decisions with data point based meta-learning

Irene Cramer, Barbara Rauch, Hagen Fürstenau, Dan Shen, Maria Staudte

`Contact:barbara.rauch@LSV.uni-saarland.de`

30-sec summary

- How can we aid non-experts with selection of ML technique?
- Meta-Learning has been used for this in the past: train a meta-learner to predict the performance of a base classifier
- Novelty: we're proposing a change to classic meta-learning which shows promising results in an initial evaluation: instead of making predictions for a data set, predict decision for each *data point*

Feedback we'd like from YOU

- How do *you* select a technique when faced with a problem?
- Do you believe it makes sense to use meta-learning to guide users?
- Do you agree with the basic 'ML toolbox' assumption of this work?

The problem

- Many factors influence choice of ML technique; simplifications: only looking at classification, only considering classification accuracy as selection criterion.
- Given a classification task and a set of applicable ML methods, which algorithm do we expect to achieve the highest accuracy?

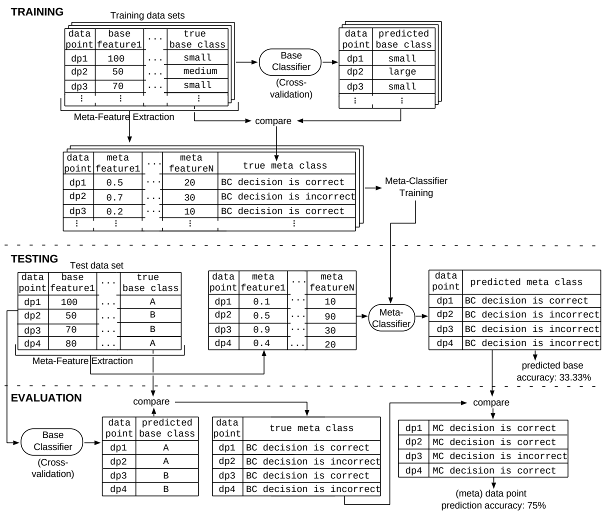
Meta-learning: basic idea

- Basic setup: describe dataset with meta-features, train meta-classifier to predict performance of base classifier
- Input unit = data set
- Output: accuracy prediction, algorithm ranking or suitability decision

Previous work

- Large EU projects: StatLog ('90-'93), MetaL ('98-'02) [REF!]
- Meta-learn. workshops at ECML98, ICML99, ECML00, ICML05
- Special Issue on Meta-Learning, Machine Learning, 03/2004 [2]

Data-point based meta-learning



- Input unit = data point; i.e. train meta-level classifier to predict classification results for a data point
- Result may be converted into ranking or accuracy, i.e. no information loss compared to previous approach
- Reasons for doing it this way:
  - we get more fine-grained information that can be used in e.g. multiple classifier systems or semi-supervised learning
  - intuitively, this level of processing may be more suited to the problem: base classifier's overall accuracy depends on correctness of every single point's classification

Proof-of-concept implementation

- Five classifiers used at base and meta level: decision tree (C4.5), K-nearest neighbour, Naive Bayes, Ripper rule learner, SVM with radial basis function kernel
- Used YALE toolkit [1] implementations
- 19 simple meta-features. Examples:
  - # attributes
  - Proportion of nominal attributes
  - Prop. of undefined values among the values
  - # training data points
  - # classes actually occurring in the training data
  - Prop. of training data with same class as test point
  - Prop. of points with same class over the k nearest training points
  - Normalised Hamming distance to the nearest point

Initial Results: promising

- Evaluated on 127 data sets from Weka archive (UCI KDD / StatLib repositories) = 102 training + 25 test sets
- Majority Baseline ("BC is correct"): 65% prediction accuracy
- Average with simple system: 72.14%
- With best meta-classifier (MC): 83.34%

MC	BC	Acc.	MC	BC	Acc.	MC	BC	Acc.	MC	BC	Acc.	MC	BC	Acc.	Avg.
DT	DT	76.08	DT	KNN	**77.56**	DT	NB	68.93	DT	RL	79.52	DT	SVM	66.86	73.79
KNN	DT	76.79	KNN	KNN	73.25	KNN	NB	**75.21**	KNN	RL	60.38	KNN	SVM	69.12	70.95
NB	DT	70.51	NB	KNN	60.60	NB	NB	64.48	NB	RL	62.66	NB	SVM	81.44	67.94
RL	DT	**79.28**	RL	KNN	69.03	RL	NB	69.87	RL	RL	74.82	RL	SVM	71.12	72.83
SVM	DT	62.66	SVM	KNN	69.55	SVM	NB	59.00	SVM	RL	**93.54**	SVM	SVM	**91.11**	75.17
Average		73.06	Average		70.00	Average		67.50	Average		74.18	Average		75.93	72.14
Baseline		70.69	Baseline		68.53	Baseline		64.07	Baseline		67.89	Baseline		54.77	65.19

Issues with current system

- No tuning or feature selection
- Simplistic meta-features
- Although more data sets used than in most other meta-learning studies, still low number

Conclusions

- blah, blah, blah....
- blah, blah, blah....

Acknowledgements

Thanks to Katharina Morik, Donald Michie, Stefan Evert, Dietrich Klakow, and Miles Osborne for their helpful comments. This research was partially funded by DFG studentships in the International Post-Graduate College "Language Technology and Cognitive Systems".

References

[1] Mierswa, Ingo and Wurst, Michael and Klinkenberg, Ralf and Scholz, Martin and Euler, Timm (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. In *Proc. ACM SIGKDD 2006*, ACM Press.

[2] Special Issue of *Machine Learning* on Metalearning, 54(3), 2004.

UNIVERSITÄT DES SAARLANDES