Automatic Selection, Configuration &
Composition of ML Algorithms

Metalearning for Algorithm Selection

Pavel Brazdil
LIAAD Inesc Tec / Univ. Porto

Overview

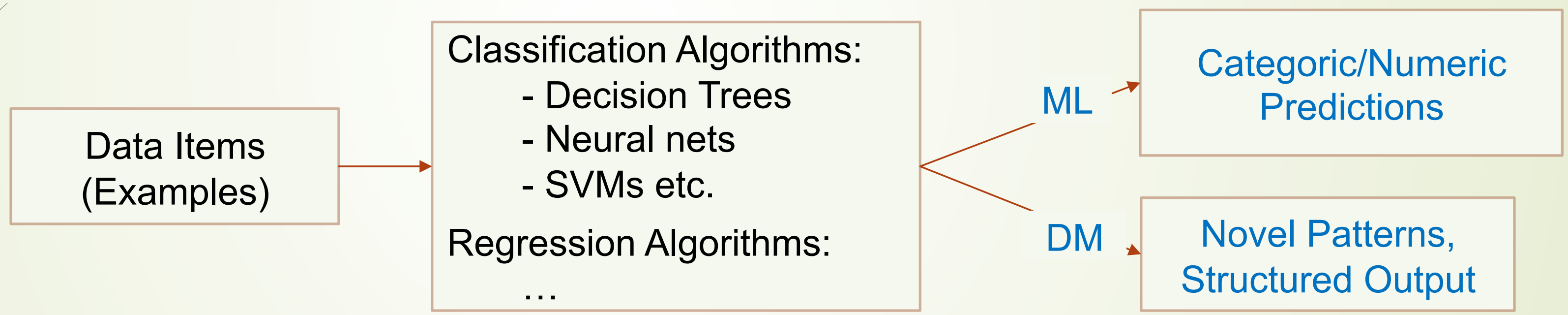
Acknowledgments

Acknowledgements to the following researchers that worked with me on these topics:

- Salisu Abdulrahman
- Miguel Cachada

1. The ML/DM Algorithm Selection & Configuration Problem

Information Flow in Machine Learning (ML) / Data Mining (DM) Systems:



1. The ML/DM Algorithm Selection & Configuration Problem

In general *worklflows*

A large set of algorithms is available in ML:

Classification Algorithms:
- Decision Trees
- Neural nets
- SVMs
- …

Ensembles of algorithms

+ It increases a possibility of finding a good solution.
- It is much harder to find the right algorithm.

This problem is aggravated by:
Many algorithms need hyperparamater settings (NNs, SVMs etc.).

We want methods that
identify/select the algorithm & its configuration with the best performance.

We cannot test all algorithms for computational reasons
(thousands of variants of algorithm + hyperparameter configurations)

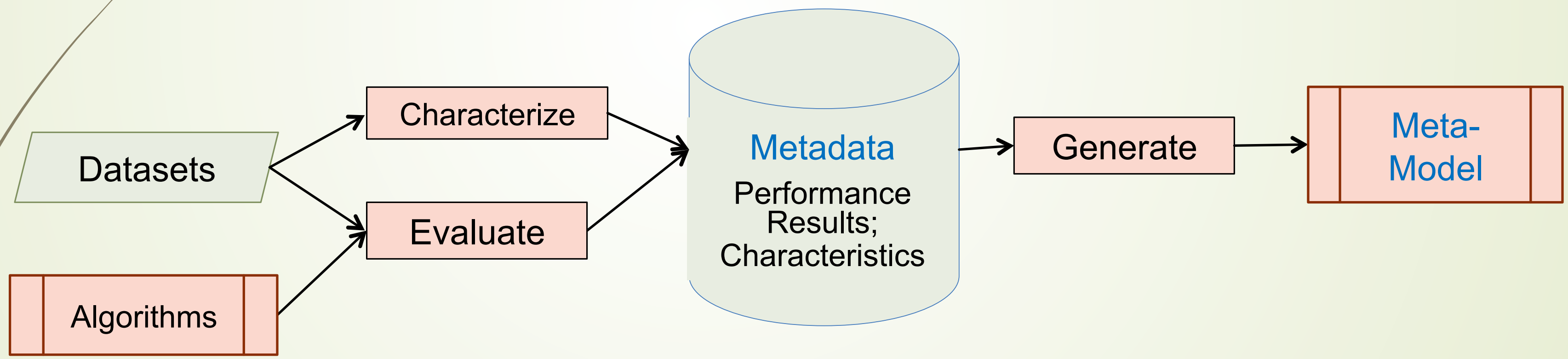
This problem was first formulated by Rice (1976).

2. How can Metalearning Help? (1)

Meta-learning is learning about which method / algorithm is best for which situation

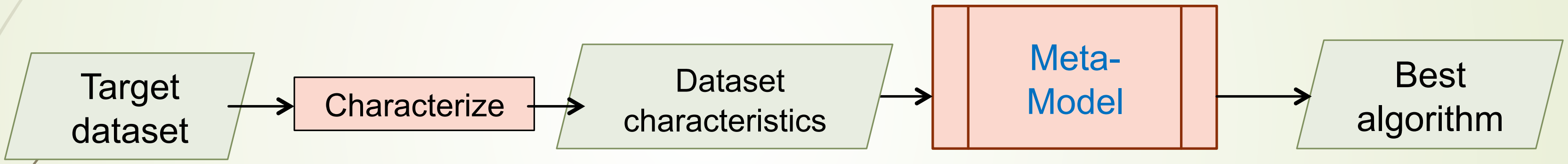
'Classical Approach' (since 1990 until about 10 years ago):

Phase 1. Generation of the meta-level model



2. How can Metalearning Help? (2)

Phase 2. Applying the meta-level model to the target dataset



Definition in Brazdil et al.,
Metalearning: Applications to Data Mining, Springer, 2008:

Metalearning is the study of principled methods that exploit metaknowledge to obtain efficient models and solutions by adapting machine learning and data mining processes.

2. How can Metalearning Help? (3)

Iterative approach (in the last 10 years):

- Some researchers realized that it is useful to carry out limited tests on the target dataset.
- The performance-based chacterization of the target dataset is used to condition the next step in the search for the best algorithm.
- The approach is iterative.

Definition of Lemke et al., *Metalearning: a survey of trends and technologies*, 2015:

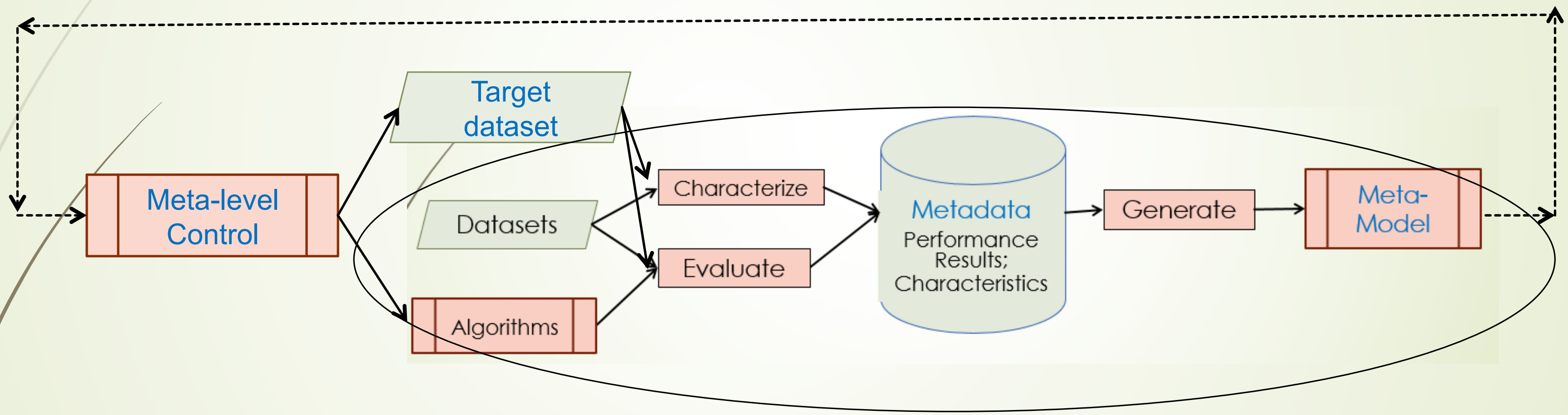
A meta-learning system must include a learning subsystem, which adapts with experience. Experience is gained by exploiting metaknowledge extracted:

a) in a previous learning episode on a single dataset and/or
b) from different domains or problems.

9

2. How can Metalearning Help? (4)

The new iterative approach merges phases 1 and 2
of the ‘classical approach’:



2. How can Metalearning Help? (5)

Two basic types of meta-models:

- **Relative performance models**
- **Empirical performance models (EPM's)**

2. How can Metalearning Help? (5)

Relative performance models

- Typically based on:
  - ***pairwise comparisons*** or
  - ***ranking approaches***
- Useful mainly in the search for the best algorithm (in general workflow)
- Can deal with hyperparameter configurations (but requires careful management of alternatives)
- Advantages: Both the models and the methods are rather simple
- Disadvantages: Meta-level model is defined by extension (enumeration of alternatives)

2. How can Metalearning Help? (5)

Empirical performance models (EPM's)

- Typically some type of ***regression models***, capable of predicting performance;
- Useful mainly in the search for the best hyperparameter configuration
- Can be extended to deal with also with algorithm selection

- Advantages: Intensional models are more appealing than extensional ones
- Disadvantage: Both the models and the methods are more complex than in the extensional approach

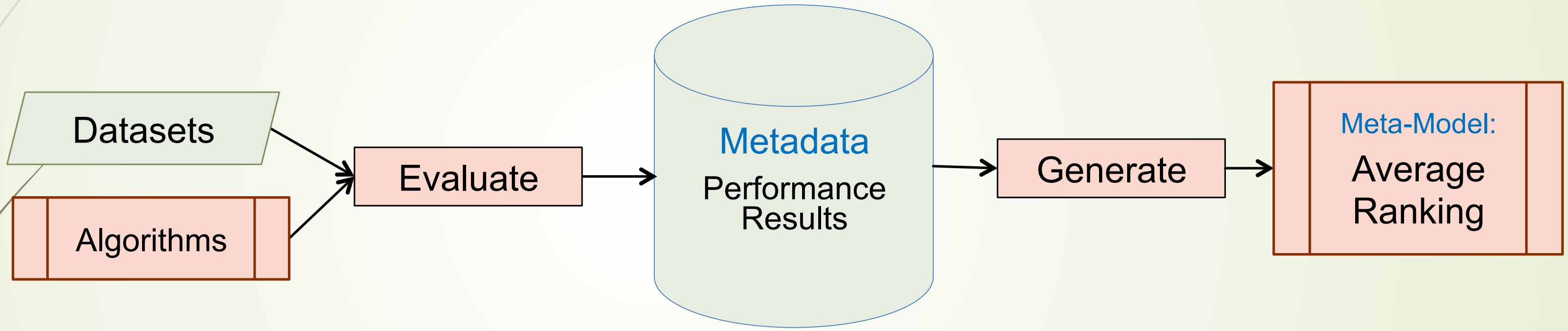
3. Algorithm Selection with Average Ranking (AR) (1)

Why to consider *Average Ranking (AR)* as a Metalearning method?

- AR is a very simple scheme, easy to implement;
- The simple version does not need classical dataset characteristics;
- It is hence applicable to many domains;
- The variant that uses a combined measure of accuracy and runtime achieves excellent results;
- This method can play the role of *straw man / default* method. More complex methods should perform better than this method.

3. Algorithm Selection with Average Ranking (AR) (2)

Using Average Ranking (AR) as the Metalearning method:



3. Algorithm Selection with Average Ranking (AR) (3)

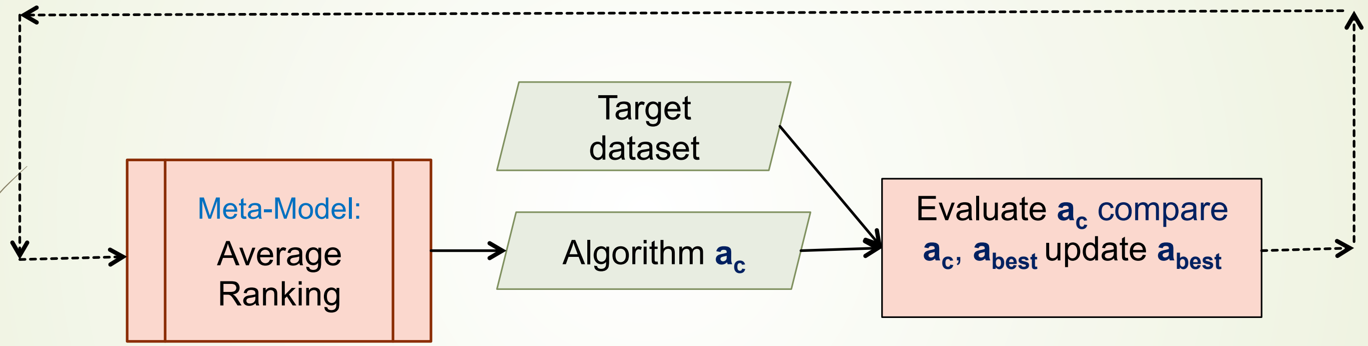
Merging rankings to generate an ***average ranking***:

Example with 2 rankings:

Rank	D1	D2	Average Ranks	Rank	Average Ranking
1	a_1	a_2	$r(a_1)=2.0$	1-2	a_1, a_3
2	a_3	a_3	$r(a_2)=2.5$	3	a_2
3	a_4	a_1	$r(a_3)=2.0$	4-5	a_4, a_6
4	a_2	a_6	$r(a_4)=4.5$	6	a_5
5	a_6	a_5	$r(a_5)=5.5$		
6	a_5	a_4	$r(a_6)=4.5$		

3. Algorithm Selection with Average Ranking (AR) (4)

Conduct Tests to Identify the Best Algorithm (Top-N strategy):



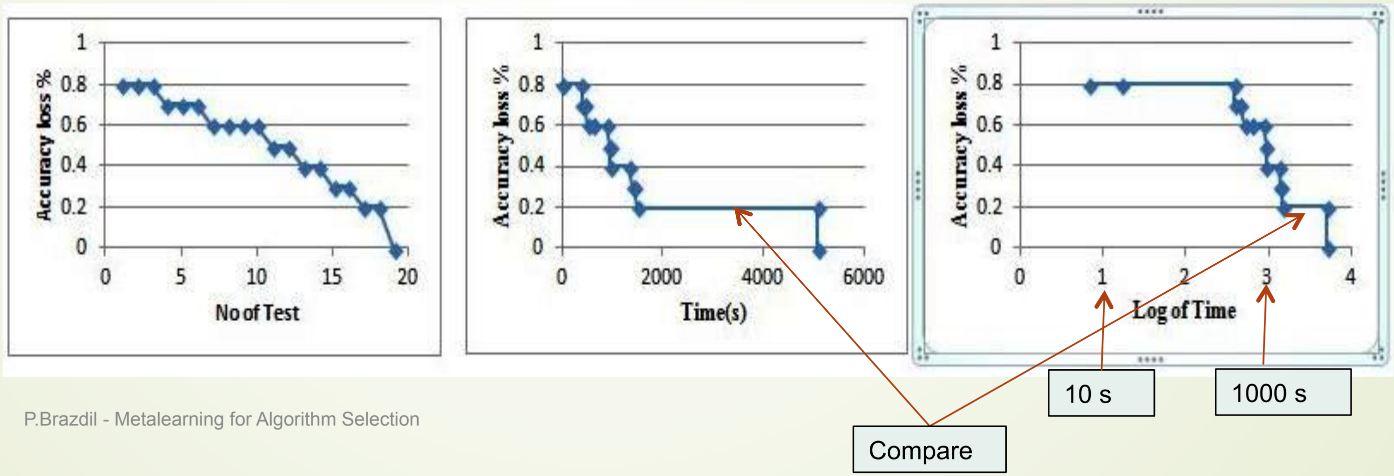
- Use the top algorithm in the average ranking to initialize a_{best}
- Go through all algorithms **sequentially** in the ranking & evaluate each one (e.g. use cross-validation test).
- If some algorithm a_c achieved a better performance than a_{best} , set $a_{best} \leftarrow a_c$.

3. Algorithm Selection with Average Ranking (AR) (5)

Evaluating the AR method

How good is the ranking? How can we evaluate this?

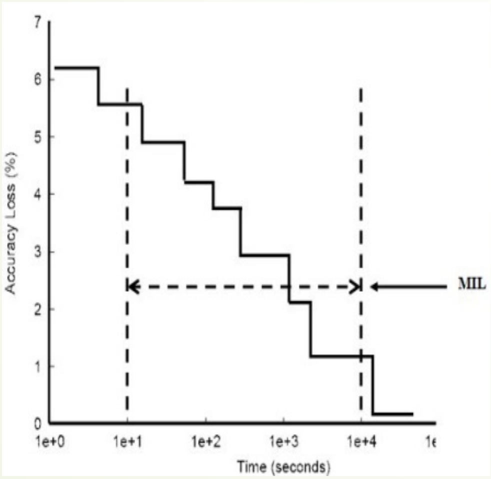
- We need to know in advance the performance of a*,
  the best algorithm in the ranking.
- Calculate **accuracy loss** of each algorithm wrt. a*,
  as we go testing the algorithms in the ranking.



3. Algorithm Selection with Average Ranking (AR) (6)

Mean Interval Loss (MIL)

can be used to characterize each loss curve:



4. AR with a Combined Measure of Accuracy & Runtime (1)

The aim is to consider two different performance measures
- e.g. accuracy (or AUC) and time -

There are two approaches:

- Define a **combined measure**, or
- Carry out multi-objective analysis (e.g. DEA).

This is followed up here

4. AR with a Combined Measure of Accuracy & Runtime (2)

Defining a combined measure

Some authors held a view that for some purposes this measure should not include absolute values, but rather ratios:

- Ratios of accuracies of two algorithms (SR's)
- Ratios of times of two algorithms (T's) (rescaled by parameter P)

One proposal

of S.Abdulrahman, & Brazdil, MetaSel 2014:

$$A3R_{a_{ref},a_q}^{d_i} = \frac{\frac{SR_{a_p}^{d_i}}{SR_{a_{ref}}^{d_i}}}{(T_{a_p}^{d_i}/T_{a_{ref}}^{d_i})^P}$$

4. AR with a Combined Measure of Accuracy & Runtime (3)

Effect of altering parameter P

- The ratio of success rates was fixed to 1
- Suppose that the ratio of times is 1/1000
- We get: $\mathbf{A3R} = \frac{1}{\left(\frac{1}{1000}\right)^{P}}$

P	A3R
1	1000
1/4	5.623
1/16	1.539
1/64	1.114

As P gets smaller, time ratio get more "*squashed*"
In the limit, as P=0, the time ratio is 1 (runtime is ignored).

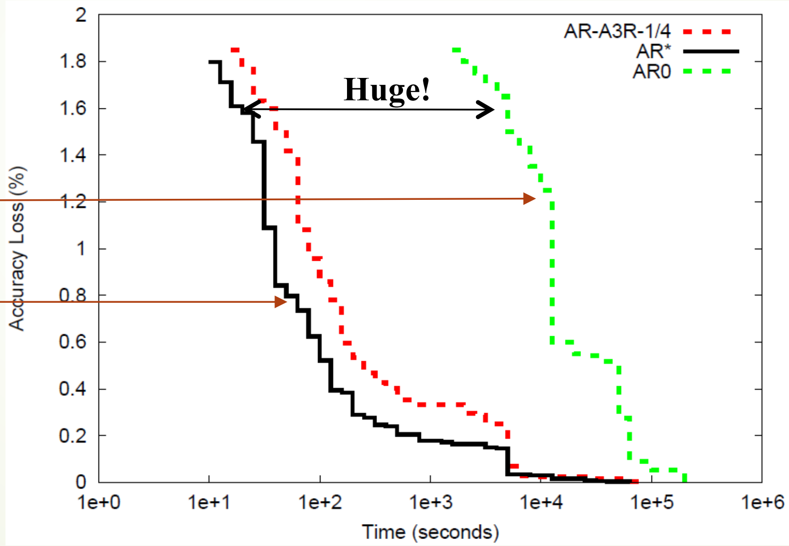
4. AR with a Combined Measure of Accuracy & Runtime (4)

The combined measure A3R was used
to upgrade the average ranking method.

This lead to excellent results:

Average ranking with accuracy

Average ranking AR* with A3R



P	1/4	1/16	1/64	1/128	1/256	0
MIL	0.752	0.626	**0.531**	0.535	0.945	**22.10**

5. Using Dataset Characteristics to Identify Similar Datasets (1)

Observation:

Rankings on similar datasets are similar.

This can be exploited
to generate better rankings and hence better loss curves.

How can we measure dataset similarity?

This area was researched since 1990’s

5. Dataset Characteristics (2)

Dataset characteristics:

- Statistical & Information-theoretic measures
- Landmarks
- Sub-sampling landmarks and learning curves
- Relative landmarks
- Concept-based measures

5. Dataset Characteristics (3)

Statistical and information-theoretic measures:

- Number of classes,
- Class entropy,
- Number of features,
- Ratio of examples to features,
- Degree of correlation between features and target concept,
- etc.

\+ Positive and tangible results (e.g., in projects Statlog and METAL).

\- There is a limit on how much information these can capture.
They are uni- or bi-lateral measures (2 attributes or attribute/class)

5. Dataset Characteristics (4)

Landmarkers

Measure the performance of a set of simple and fast learning algorithms (landmarkers) (e.g. simplified decision tree)

The accuracy of these landmarkers is used to characterize the dataset.

- Which landmarkers should be used is a non-trivial problem

5. Dataset Characteristics (5)

Sub-sampling landmarks and learning curves

Exploit performance information obtained
on **simplified versions of the data (samples).**

Accuracy results on these samples (or sequences of samples)
serve to characterise individual datasets and
are referred to as ***sub-sampling landmarkers*****.**

6. Exploiting Dataset Characteristics in Meta-Models (1)

Focussed AR*

- Given a target dataset D_{target}
- Use (classical) dataset measures to select a subset of *similar datasets*
- Apply AR* on the similar datasets

6. Exploiting Dataset Characteristics in Meta-Models (2)

Determining which of two algorithms (A_p, A_q) is better

(exploits sample-based characterization) (Leite & Brazdil, 2010)

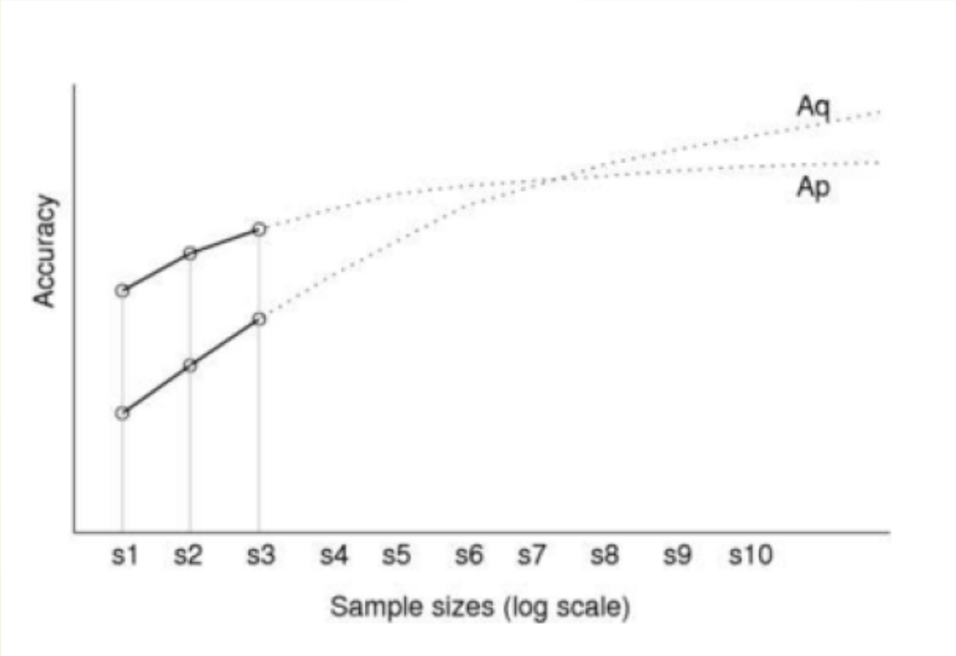
- Given a target dataset D_{target}
- Construct partial learning curves (N samples)
  i.e. performance of the two algorithms on each sample
- Identify the most similar partial learning curve(s)
  Retrieve the curve, adapt & project to obtain estimate of performance.
- Determine which curve achieves better performance

One recent paper :

Jan N. van Rijn et al., Fast Algorithm Selection using Learning Curves, IDA 2015, Springer

6. Exploiting Dataset Characteristics in Meta-Models (3)

Examples of two partial learning curves:



6. Exploiting Dataset Characteristics in Meta-Models (4)

Extending the method for algorithm selection

Jan van Rijn et al., SM Abdulrahman, P Brazdil, J Vanschoren:
Fast Algorithm Selection using Learning Curves, IDA 2015, Springer

Details will be reported by Joaquin Vanschoren

7. ML systems as Meta-level Models

Meta-Model

The **meta-level model** can be in the form of:

- k-NN (used by many - incl. Brazdil, Gama & Henery, 1994)
- Meta-level rules (not very reliable),
- Neural network,
- Approximate Ranking Trees (ART) (Q. Sun, 2013; Sun & Pfahringer, 2013)
- Forests of ARTs(Q. Sun, 2013; Sun & Pfahringer, 2013)

Etc.

8. Active Testing (1)

The AR* method has a shortcoming:

It tests the algorithms in the ranking sequentially.

This gives rise to two problems, as the algorithm portfolio may contain:

- Suboptimal algorithms
- Very similar (redundant) algorithms.

(e.g. variants of the same algorithm with different parameter settings).

Time can be wasted by testing.

How can this be avoided?

8. Active Testing (2)

Eliminating sub-optimal algorithms in pre-processing stage (filter-like method):

- Process all datasets one by one.
- For each dataset mark all algorithms that achieved a competitive result (e.g. best / equivalent to best).
- After all datasets have been processed, drop all unmarked algorithms (they did not win on any dataset)
- Use the remaining algorithms from then on.

Positive results were reported (Brazdil, Soares & Pereira, 2001)

The method needs to be upgraded to deal with the dichotomy of both accuracy and runtime!

8. Active Testing (3)

Eliminating very similar algorithms

Various techniques exist that can identify similar algorithms
by considering performance
e.g. by **identifying algorithms** that commit **correlated errors**
(see e.g. Lee & Giraud-Carrier, 2011)

This could be exploited:
The algorithms that commit similar errors to others could be dropped.

In the next slides we discuss a method of ***active testing (AT)*.**
It deals with these two issues in an *on-line manner*.

8. Active Testing (4)

Active Testing Method (e.g. Leite, Brazdil & Vanschoren, 2012)

- It does not follow the ranking!
- It jumps to the most promising algorithm $\mathbf{a_c}$, based on the expected performance gain (ΔPf) over $\mathbf{a_{best}}$ (earlier ΔPf was called *relative landmarker*)

37

8. Active Testing (5)

Searching for the best competitor:

$$a_c = \underset{a_k}{\operatorname{argmax}} \sum_{d_i \in D} \Delta Pf(a_k, a_{best}, d_i)$$

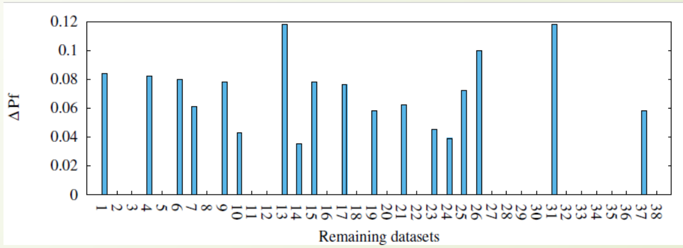
Determining the best competitor among different alternatives

Alg.	$\sum \Delta Pf$
a_1	0.587
a_2	**3.017**
a_3	0.143
a_4	0.247
a_5	1.280

Determining Pf (upgrade of AT for A3R):

$$\Delta Pf(a_j, a_{best}, d_i) = \left(\frac{\dfrac{SR_{a_p}^{d_i}}{SR_{a_{ref}}^{d_i}}}{\left(T_{a_p}^{d_i} / T_{a_{ref}}^{d_i}\right)^P} - 1 \right) \quad \text{for values} > 0 \text{ only}$$

Δ Pf's for a particular algorithm



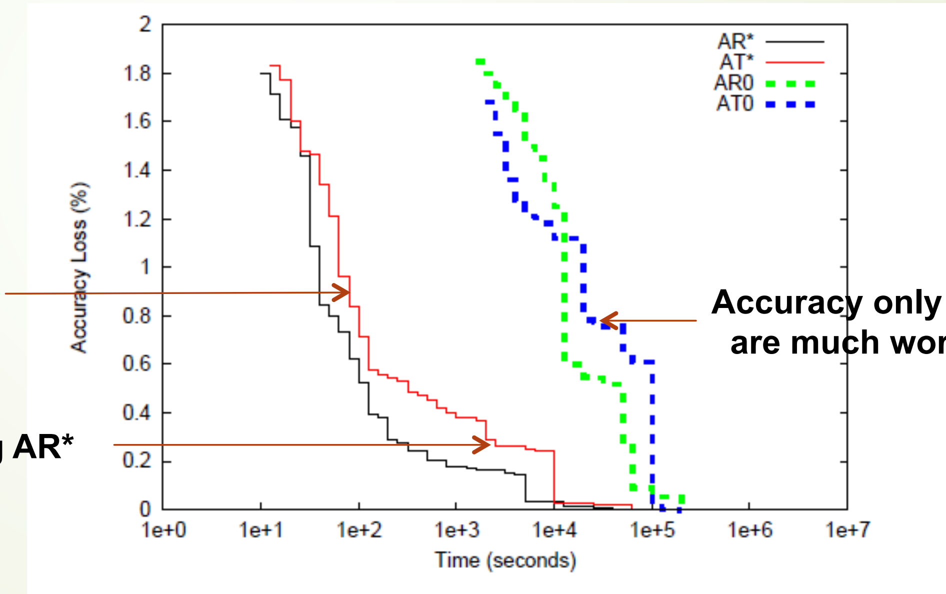
8. Active Testing (6)

The active testing method leads to good results:

(SM Abdulrahman, P Brazdil, J van Rijn, J Vanschoren, to appear in SI on Metalearning, MLJ, 2018):

Active Testing AT*

Average Ranking AR*



Accuracy only variants are much worse!

9. Using AR* on incomplete data (1)

Some Questions:

1. Why should we worry about incomplete meta-data?
2. Can AR* method deal with incomplete meta-data?
3. If not, how can AR* be improved?
4. What implications does this have for metalearning?

9. Using AR* on incomplete data (2)

Question 1:
Why should we worry about incomplete meta-data?

Some test results are often missing.
We want our systems to cope with real-world situations!

Example with 4 algorithms and 5 datasets:

Algorithm	D_1	D_2	D_3	D_4	D_5
a_1	0.98		0.55		0.78
a_2		0.90		0.55	0.79
a_3	0.76	0.61	0.88		
a_4		0.84		0.45	0.38

Additional problem:

The omissions might not be equally distributed across datasets!

9. Using AR* on incomplete data (3)

Question 2:

Can AR* method deal with incomplete meta-data?

R_1	Rank
a_1	1
a_3	2
a_4	3
a_2	4
a_6	5
a_5	6

R_2	Rank
a_2	1
a_1	2

Calculating the **aggregated** rank of a_2 as (4+1)/2 is **not right!**

9. Using AR* on incomplete data (4)

Question 3:

How can AR* be improved? (i.e. to deal with incomplete rankings)

There are many methods that can be used to aggregate incomplete rankings[1].

Here we use a simple method that gives **different weight** according to the **size of the ranking**[2].

Results:

The improved AR* method is not effected by 50% omissions and degrades gracefully afterwards.

[1] S.Lin. Rank aggregation methods. WIREs Computational Statistics, 2:555-570, 2010

[2] SM Abdulrahman, P Brazdil, J van Rijn, J Vanschoren, to appear in SI on Metalearning, MLJ, 2018

9. Using AR* on incomplete data (4)

Question 4

What implications does this have for metalearning?

- We can **conduct fewer tests**, but still obtain a meta-model with similar performance
- Carry out more test on more promising algorithms
  - We have done some experiments that support this
  - Strategy proposed by some in the community studying *multi-armed bandits* problems

Save a lot of effort of setting-up a metalearning system!

References

Books and Survey Articles:

P. Brazdil, Christophe G. Giraud-Carrier, C. Soares, R. Vilalta: *Metalearning - Applications to Data Mining.* Springer, 2009.

K.Smith-Miles: Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection, *ACM Computing Surveys*, 2008

F Serban, J Vanschoren, JU Kietz and A Bernstein: A Survey of Intelligent Assistants for Data Analysis, *ACM Computing Surveys*, 2013

References

SM Abdulrahman, P Brazdil, J van Rijn and J Vanschoren: Speeding up Algorithm Selection using Average Ranking and Active Testing by Introducing Runtime, to appear in *Special Issue on Metalearning and Algorithm Selection*, *Machine Learning Journal*, Jan. 2018

SM Abdulrahman, P Brazdil: Measures for Combining Accuracy and Time for Meta-learning, Proc. of Workshop MetaSel-2014 associated with ECAI-2014, CEUR proceedings, 2014

S Abdulrahman, P Brazdil, J van Rijn, J Vanschoren: Algorithm Selection via Meta-learning and Sample-based Active Testing, Proc. of Workshop MetaSel-2015 associated with ECML/PKDD-2015, CEUR proceedings, 2015

P Brazdil, J Gama, B Henery, Characterizing the applicability of classification algorithms using meta-level learning, *Machine Learning: ECML-94,* 83-102

PB Brazdil, C Soares, JP da Costa: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results, *Machine Learning 50 (3),* 251-277, 2003

P Brazdil, C Soares, R Pereira: Reducing rankings of classifiers by eliminating redundant classifiers, Progress in Artificial Intelligence, 14-21, 2001

M Hilario, P Nguyen, H Do, A Woznica, A Kalousis, "Ontology-based meta-mining of knowledge discovery workflows." in N Jankovski et al. (eds.), *Meta-Learning in Computational Intelligence*. Springer, 2011. 273-315.

References

Kietz, JU, F Serban, A Bernstein, & S Fischer, Designing KDD-workflows via HTN-planning for intelligent discovery assistance. In *5th Planning to Learn Workshop, WS28* at ECAI-2012.

JW Lee, C Giraud-Carrier, A metric for unsupervised metalearning, *Intelligent Data Analysis*, 2011

R Leite, P Brazdil: Active Testing Strategy to Predict the Best Classification Algorithm via Sampling and Metalearning. Proc. of ECAI, 309-314, 2010

R Leite, P Brazdil, J Vanschoren: Selecting Classification Algorithms with Active Testing, Proc. of MLDM, Springer, 2012;

M. Mısır , M. Sebag: Algorithm Selection as a Collaborative Filtering Problem, Inria Research Report N °XX, December 2013, Research Centre Saclay – Île-de-France.

P Nguyen, J Wang and M Hilario, A Kalousis: Learning Heterogeneous Similarity Measures for Hybrid-Recommendations in Meta-Mining, Proc. of ICDM-2012, also in arXiv:1210.1317, 2012

Rice, J. : The algorithm selection problem. *Advances in Computers 15, 65–118*, 1976.

JN van Rijn, SM Abdulrahman, P Brazdil, J Vanschoren, Fast algorithm selection using learning curves, International Symposium on Intelligent Data Analysis XIV, 298-309, 2015

H. Robbins. Some Aspects of the Sequential Design of Experiments. In Bulletin of the American Mathematical Society, volume 55, pages 527–535, 1952.

References

Q Sun: *Meta-Learning and the Full Model Selection Problem*, PhD thesis, U.Waikato, 2013

Q Sun and B Pfahringer. Pairwise Meta-Rules for Better Meta-Learning-Based Algorithm Ranking. *Machine Learning, 93(1):*141-161, 2013.

J Vermorel and M Mohri: Multi-armed Bandit Algorithms and Empirical Evaluation, ECML 2005, LNAI 3720, Springer