Tutorial/Workshop @ ECML PKDD 2022

Meta-Knowledge Transfer/Communication in Different Systems

Jan N. van Rijn, Pavel Brazdil, Henry Gouk, Felix Mohr

Metalearning for Algorithm/Workflow Selection

Pavel Brazdil
Prof. Emeritus, U.Porto

23 Sept 2022

Overview

Duration ≈ 32 min.

Acknowledgements

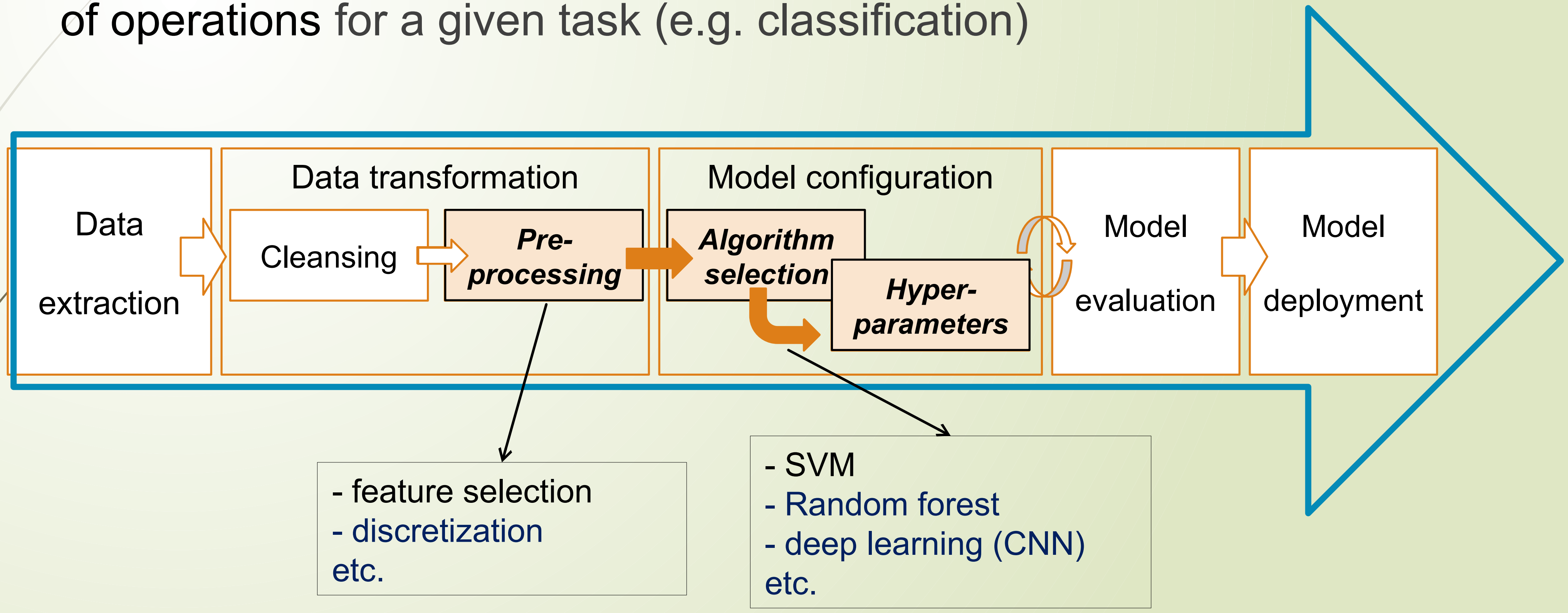
The work presented benefited from joint work with:

Salisu M. Abdulrahman, Miguel V. Cachada, Maria J. Ferreira, Fernando Freitas, João Gama, Rui Leite, Jan van Rijn, Carlos Soares

Introduction (1)

Ch. 2

Typical task is to **recommend a workflow (pipeline)** of operations for a given task (e.g. classification)



There may be many thousands of variants to select from !

Introduction (2)

Distinguish:

1. The current task is another similar problem in the same domain

=> Use **metalearning** methods exploiting information
about past experiments on **similar tasks** (meta-data)

2. The current task is a new problem in a given domain

=> Use **AutoML** exploiting information
about past experiments on the **same task**

Introduction (3)

What is metalearning?

A meta-learning system must include a learning subsystem, which adapts with experience.

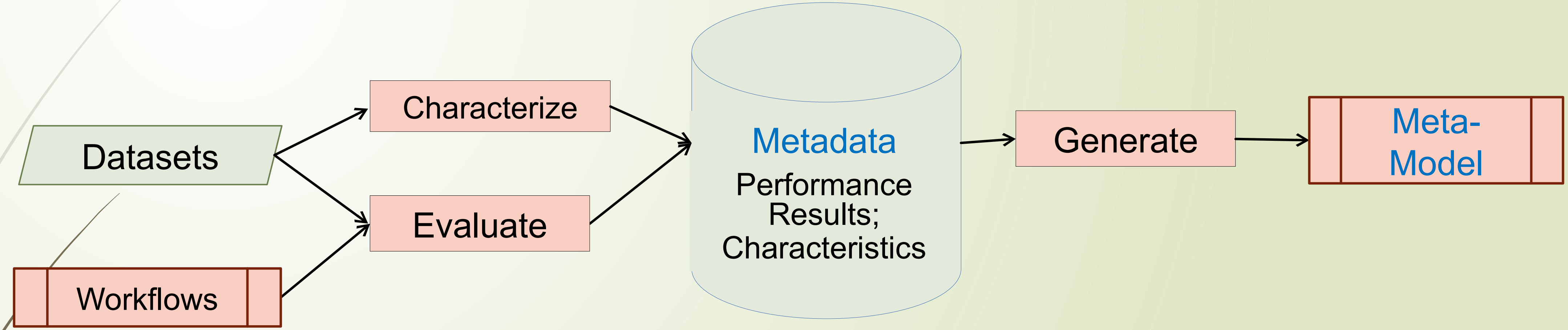
Experience is gained by exploiting metaknowledge extracted:

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

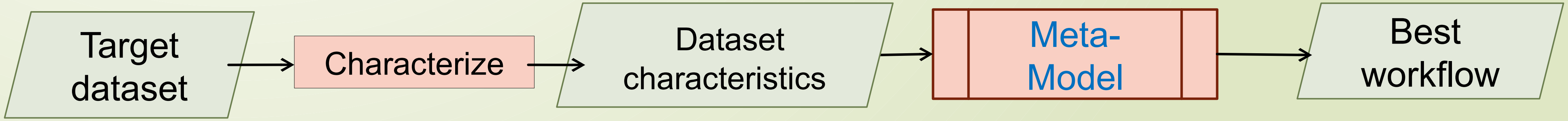
(Lemke et al., 2015)

Introduction (4)

Phase 1. Generate the meta-level model



Phase 2. Apply the meta-level model to the target dataset to **obtain the recommended workflow**



Introduction (5)

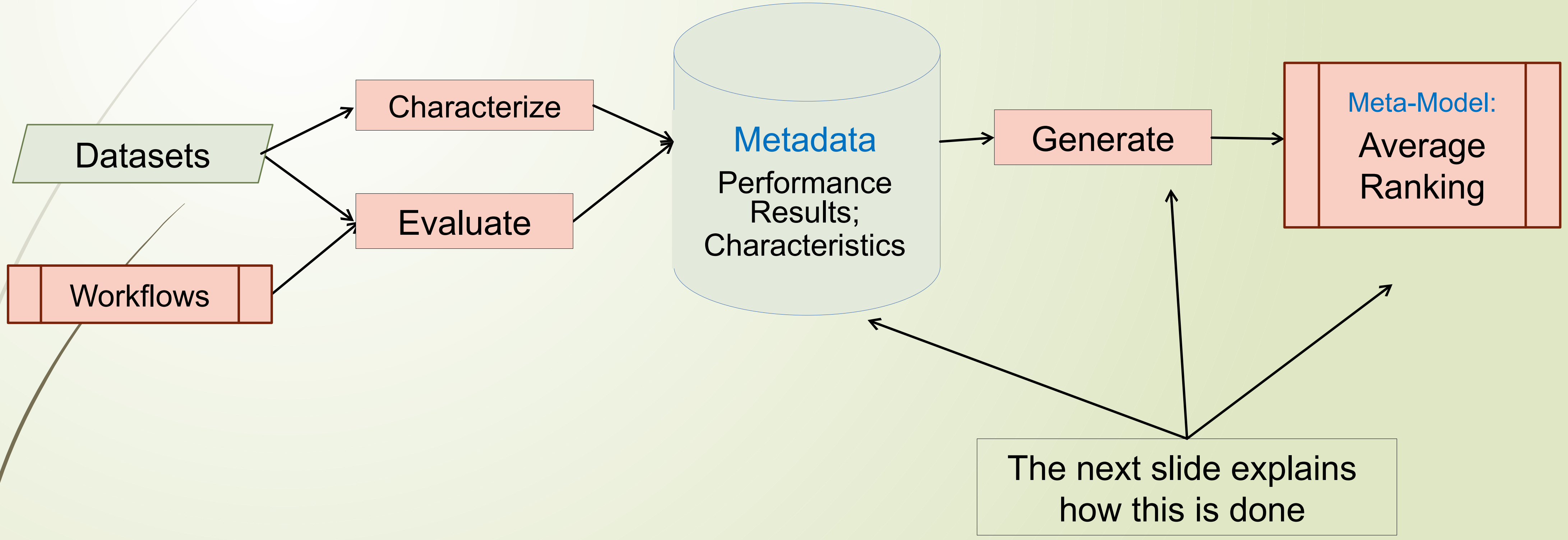
Basic types of meta-level models:

- **Relative performance models**
  - Pairwise comparisons
  - Ranking approaches (e.g. Average ranking) <= Discussed next
- **Empirical performance models (EPM's)** (exploited in AutoML)
  - Regression models, capable of predicting performance;
    They are useful in the search for the best hyperparameter configuration

1. Workflow (WF) Selection with Average Ranking (AR) (1)

Ch. 2

Phase 1. Generate the meta-level model (Average Ranking)



1. WF Selection with Average Ranking (AR) (2)

Example:

Merging rankings R1 and R2 of algorithms/workflows a1.. a6, (obtained on datasets D1 and D2) into *average ranking*:

(1+3)/2 => 2

Rank	R1 on D1	R2 on D2	Average Rank	Rank	Average Ranking
1	a_1	a_2	$r(a_1)=2.0$	1-2	$\mathbf{a_1, a_3}$
2	a_3	a_3	$r(a_2)=2.5$	3	$\mathbf{a_2}$
3	a_4	a_1	$r(a_3)=2.0$	4-5	$\mathbf{a_4, a_6}$
4	a_2	a_6	$r(a_4)=4.5$	6	$\mathbf{a_5}$
5	a_6	a_5	$r(a_5)=5.5$		
6	a_5	a_4	$r(a_6)=4.5$		

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

Phase 2. Apply the meta-level model (AR) to the target dataset **to obtain the recommended workflow**

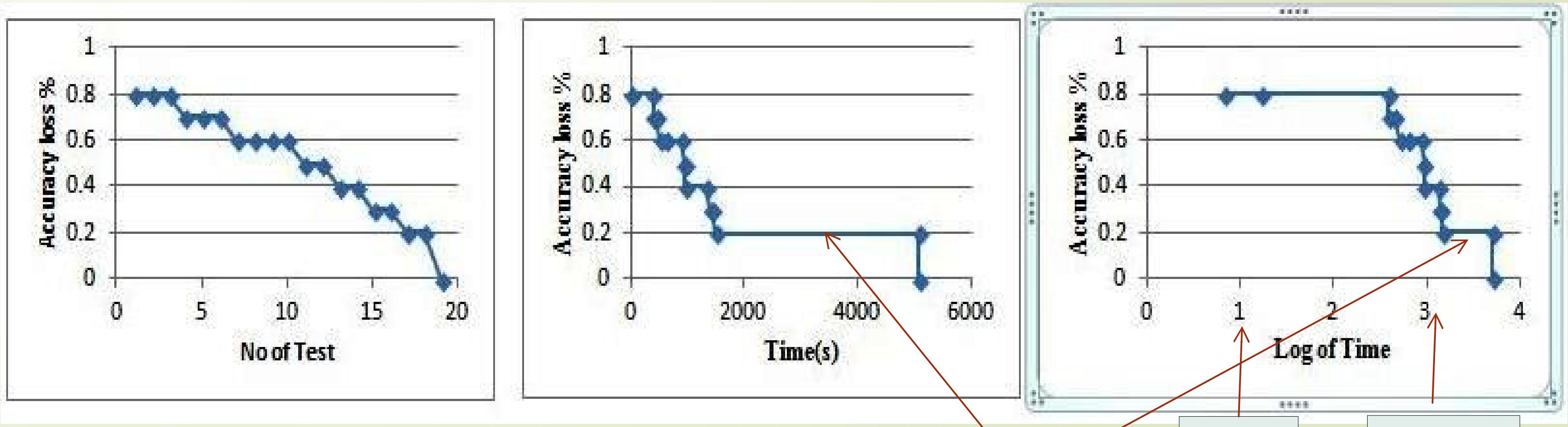
- **Use the top algorithm/workflow** in the average ranking to initialize a_{best} (incumbent)
- **Go through** all algorithms/workflows in the ranking **sequentially** & **evaluate** each one on a validation set
- If some algorithm a_c achieved a better performance than a_{best}, then **update current best** alternative (i.e., set $a_{best} \leftarrow a_c$.)
- After this process has terminated, **return a_{best}**

1. WF Selection with Average Ranking (AR) (4)

Evaluating the AR selection method

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

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



2. WF Selection with Average Ranking (AR) (5)

Questions:

- Is it a good idea to rank workflows on accuracy (or AUC etc.)?
- If not, why not?
- Is there an alternative?

2. Utilizing Accuracy and Runtime as a Measure (1)

Answers:

- **Ranking workflows solely on accuracy** has disadvantages. Accurate workflows can be rather **slow to test.**
- A better alternative is to rank workflows on a **combined measure of accuracy and time** (e.g., A3R').

This permit to identify workflows/algorithms with reasonable performance soon.
This is **important**, if we want to have **any-time result**.

2. Utilizing Accuracy and Runtime as a Measure (2)

- **Ranking workflows solely on accuracy** has disadvantages. Accurate workflows can be rather **slow to test.**
- A better alternative is to rank workflows on a **combined measure of accuracy and time, e.g., A3R':**

$$A3R'^{d_i}_{a_j} = \frac{P^{d_i}_{a_j}}{(T^{d_i}_{a_j})^Q}$$

performance (e.g., accuracy)

di dataset

a_j WF/algorithm

time (rescaled)

Q parameter (e.g., 1/64, i.e., 64^{th} root) controls the importance of time

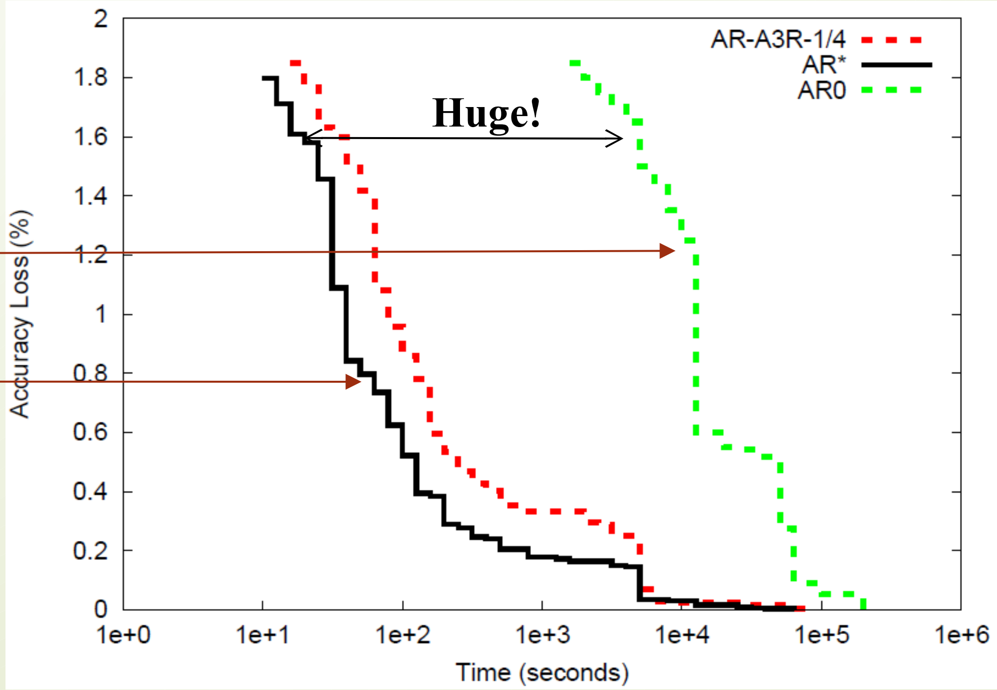
$1^{(1/64)} = 1$
$10^{(1/64)} = 1.037$
$100^{(1/64)} = 1.075$
$1000^{(1/64)} = 1.114$

2. Utilizing Accuracy and Runtime as a Measure (3)

The resulting ranking method is referred to as AR*
It lead to **excellent results**, as the following **loss curves** show:

Average ranking with accuracy

Average ranking AR* with A3R



3. Using Dataset Characteristics (1)

Ch. 3

Observation 1:

Rankings on similar datasets are similar.

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

Basically, it is necessary to select
a subset of similar algorithms/rankings and
rank them and conduct tests

How can we measure dataset similarity?

3. Using Dataset Characteristics (2)

Observation 2:

Dataset characteristics (metafeatures) may help
to discriminate between potentially **good/bad performers.**

This idea was followed in the 90's to **pre-select good performers.**

But let us come back to observation 1 and the question:
How can we measure dataset similarity?

3. Using Dataset Characteristics (3)

Dataset similarity can be established
on the basis of **dataset characteristics (metafeatures)**

These depend on the task:

- Classification
- Regression
- Time series
- Clustering
- OR and Optimization,

Here we focus on classification tasks.

3. Using Dataset Characteristics for Classification (4)

- Simple

No. of examples
No. of attributes
No. of classes
Proportion of discreet attributes
Proportion of missing values
Proportion of outliers, etc.

- Statistical

Skewness of x_i
Kurtosis of x_i
Correlation of x_i and x_j, etc.

- Information-theoretic

Feature entropy of x_i
Class entropy of x_i
Mutual information between x_i and y
Etc.

3. Using Dataset Characteristics for Classification (5)

- Model-based

Decision tree characteristics
No. of nodes
No. of leaves, etc.

- Performance-based

See the next slide

- Concept/complexity-based

Concept variation/roughness in output space
Overlap of features
Separability of classes

3. Using Dataset Characteristics for Classification (6)

Performance-based characteristics (meta-features):

Landmarkers

Performance of simple algorithms, such as:

- decision stump or decision tree
- 1NN,
- linear discriminant

characterizes data separability

characterizes linear separability

Relative landmarkers

difference (or ratio) in performance of

algorithms/workflows a_k and a_{best} (incumbent) on a dataset d_i

3. Using Dataset Characteristics for Classification (7)

Performance-based characteristics (meta-features):

Subsampling landmarkers

performance of algorithms/workflows on different samples of data

Learning curves

performance of algorithms/workflows on different samples of increasing size

3. Using Dataset Characteristics for Classification (8)

Performance-based characteristics (meta-features)

Two (or more) datasets are similar, if the performance characteristics (e.g., relative landmarkers, learning curves, etc.) of a given algorithm on these datasets are similar.

Cosine-based similarity between datasets:

$$Sim_{cos}(d_{new}, d_i) = \frac{p(\mathbf{a}, d_{new}) \cdot p(\mathbf{a}, d_i)}{|p(\mathbf{a}, d_{new})|_2 * |p(\mathbf{a}, d_i)|_2}$$

Dot product of two vectors

Performance of a_k on d_i

$$Sim_{cos}(d_{new}, d_i) = \frac{\sum_{a_k \in \mathbf{a}} p(a_k, d_i) * p(a_k, d_{new})}{|p(\mathbf{a}, d_{new})|_2 * |p(\mathbf{a}, d_i)|_2}$$

This measure was used in one variant of AT (see later)

3. Using Dataset Characteristics for Classification (8)

Performance-based characteristics (meta-features):

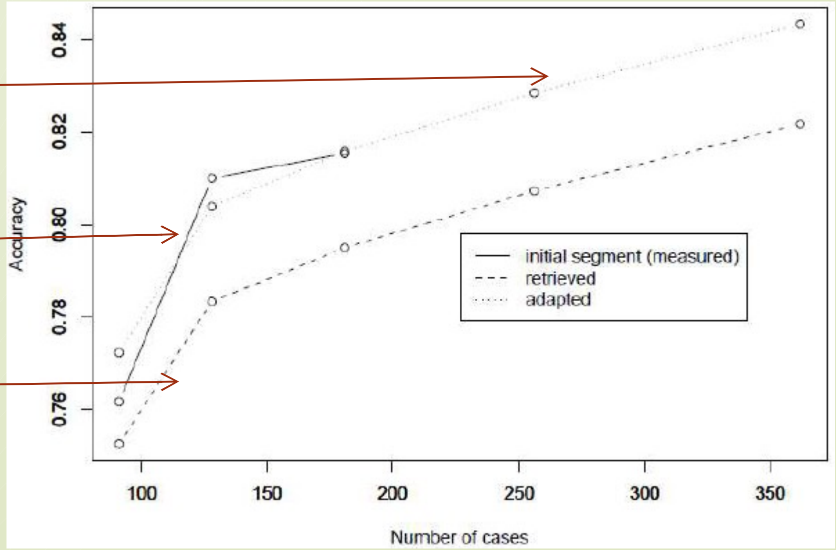
Two (or more) algorithms are similar, if the performance characteristics (e.g., relative landmarkers, learning curves, etc.) of a given dataset (d_i) of these algorithms are similar.

This can be exploited, e.g., to predict the future points of a given partial learning curve:

Predicted points

Partial learning curve on d_{new}

Most similar past learning curve



4. Active Testing (1)

The AR* method has a shortcoming:

It tests the algorithms/workflows in the ranking sequentially.

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

- Algorithms/workflows with sub-optimal performance
- Potentially redundant algorithms/workflows

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

Time can be wasted by testing.

How can this be avoided?

4. Active Testing (2)

Eliminating sub-optimal and potentialy redundant algorithms

- **In pre-processing** stage
  Filter-like method that **reduces the given configuration space** (more details are given later)
- **Incorporated within** a given metalearning/AutoML algorithm
  One particular solution - **Active testing** method

4. Active Testing (3)

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

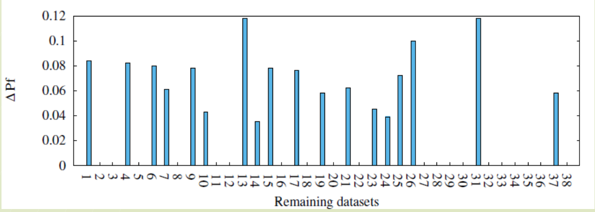
- It does not follow the ranking!
- It jumps to the most promising algorithm $\mathbf{a}_c$,
  based on the ***expected performance gain,*** over $\mathbf{a}_{best}$
  (earlier was called ***relative landmarker***)

4. Active Testing (5)

Performance gain of algorithm/workflow a_j wrt. a_{best} on d_i :

$$\Delta P(a_j, a_{best}, d_i) = A3R^{d_i}_{a_{best}, a_j} = \frac{P^{d_i}_{a_j} / P^{d_i}_{a_{ref}}}{(T^{d_i}_{a_j} / T^{d_i}_{a_{ref}})^Q}$$

Identifying the best competitor requires **summing up for various datasets**:



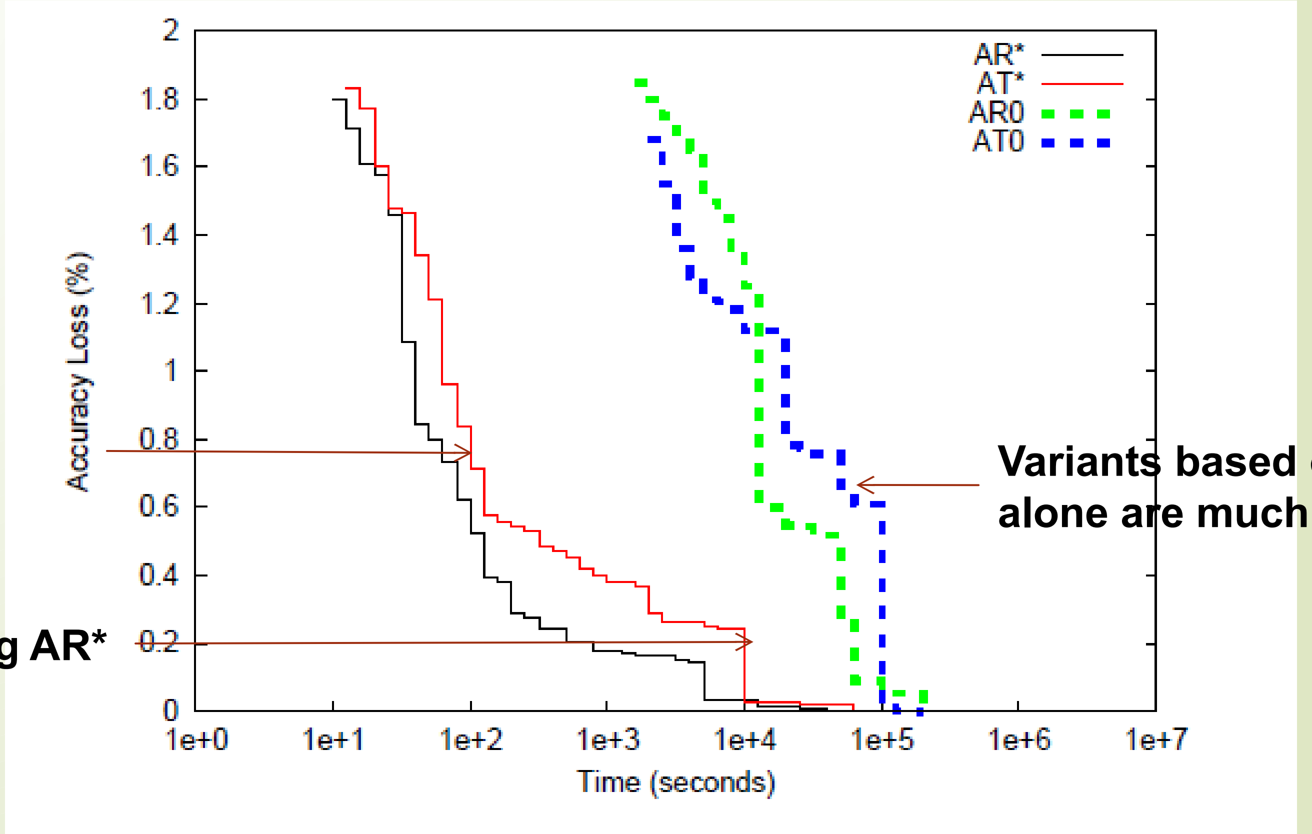
Remarks:

- defined in terms of A3R leads to better results than accuracy (Abdulrahman, et al., 2018)
- It is not necessary to limit the sum to values greater than 1 (Leite, R. and Brazdil, P., 2021)

4. Active Testing (6)

The active testing method leads to good results:
(Abdulrahman, et al., MLJ, 2018):

Active Testing



Average Ranking AR*

Variants based on accuracy alone are much worse!

References

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

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

PB Brazdil, JN van Rijn, C Soares, J Vanschoren, Metalearning: Applications to Automated Machine Learning and Data Mining (2nd ed.), Springer, 2022, Chapters 2,4,5

Leite, R. and Brazdil, P. (2021). Exploiting performance-based similarity between datasets in metalearning. In AAAI Workshop on Meta-Learning and MetaDL Challenge, vol. 140, pages 90–99. PMLR.

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

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