Data Characterization for Meta-Learning

Prof. Adriano Rivolli
rivolli@utfpr.edu.br

Departamento de Computação (DACOM)
Universidade Tecnológica Federal do Paraná (UTFPR)

June 2024

Outline

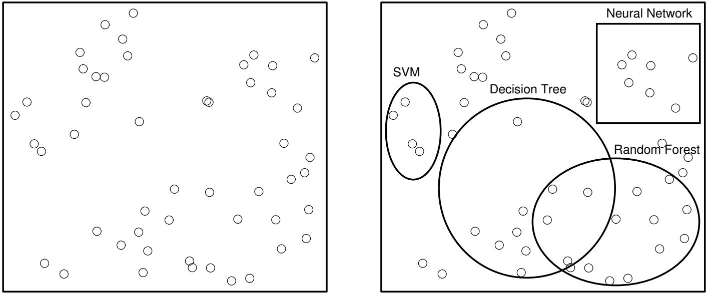
1. Introduction
2. Meta-Learning
3. Meta-Feature Extractor (MFE)
4. Standard Analysis
5. Prospective work

Introduction

Bias Definition

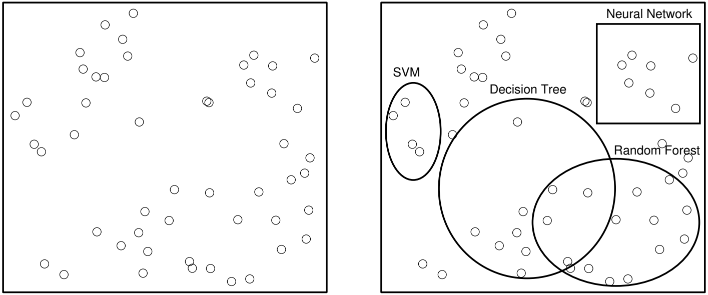
Bias has been defined as the choice of a specific generalization **hypothesis** over others, restricting the **search space** and model representation, making learning from data possible [Mitchell, 1997].

Hypothesis and search space



(a) Search space. (b) Preference bias of ML algorithms.

Hypothesis and search space

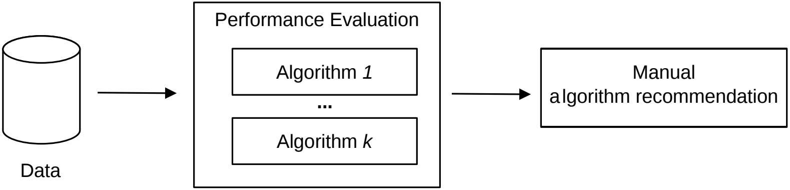


(c) Search space.

(d) Preference bias of ML algorithms.

The effect of bias for Data Science is that several algorithms are usually tried. This is called **trial-and-error approach.**

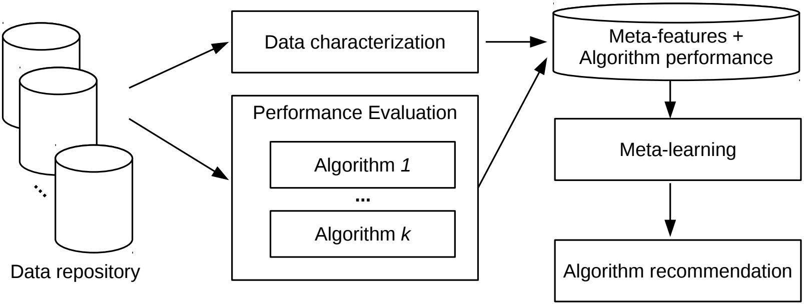
Trial-and-error approach



Trial-and-error approach

- Laborious and subjective;
- Increase the training time;
- Can cause overfitting;
- Decrease the experimental reproducible.

Meta-Learning (MtL) approach



Meta-Learning (MtL) approach

- Laborious but objective;
- Remove the training time;
- Can avoid overfitting;
- Towards the experimental reproducible.

Open gaps

- Increase the reproducible in MtL;
- Improve data characterization with new meta-features;
- Improve the MtL performance;
- Management of bias.

Outline

1. Introduction
2. Meta-Learning
3. Meta-Feature Extractor (MFE)
4. Standard Analysis
5. Prospective work

Meta-learning

MtL Definition

Study of methods that explore **metaknowledge** in order to improve or to obtain more efficient ML solutions [Brazdil et al., 2009].

Algorithm Selection Applications:

- Optimization [Kanda et al., 2011];
- Time series analysis [Rossi et al., 2014];
- Gene expression tissue classification [de Souza et al., 2010];
- SVM parameter tuning [Mantovani et al., 2015].

Algorithm Selection Framework

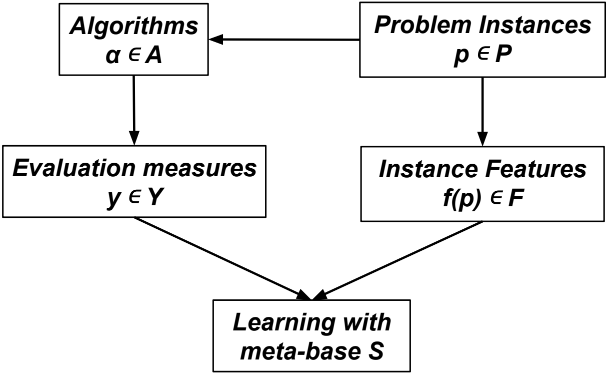


Figure: Algorithm selection framework. (Adapted from [Smith-Miles, 2008])

Problem Instances (P)

The problem instances P are datasets p that will be used to generate the meta-base. They can be collected from:

- UCI [Lichman, 2013];
- Keel [Alcalá-Fdez et al., 2011];
- OpenML [Vanschoren et al., 2013];
- Artificial datasets [Vanschoren and Blockeel, 2006];
- Datasetoids [Prudêncio et al., 2011].

Instance Features (F)

The meta-features F are designed to extract general properties of datasets $f(p)$. They are able to provide evidence about the future performance of the investigated techniques [Soares et al., 2001].

Instance Features (F)

The main groups of meta-features are:

- **General:** Extract simple and basic information;
- **Statistical:** Capture data distribution indicators;
- **Information-theoretic:** Capture the amount of information in the data and their complexity;
- **Model-based:** Extract characteristics like the shape and size of a Decision Tree (DT) model induced from a dataset.
- **Landmarking:** Represents the performance of simple and efficient learning algorithms.

Instance Features (F)

The **general meta-features** are basic information directly extracted from the dataset:

- number of attributes, instances and classes;
- frequency of instances in each class.

Instance Features (F)

The **statistical meta-features** extract information about the data distribution:

- correlation and covariance matrix;
- skewness and kurtosis.

Instance Features (F)

The **information-theoretic meta-features** capture the amount of information in the datasets:

- entropy;
- mutual information;
- noise signal ratio.

Instance Features (F)

The **model-based meta-features** are information extracted from a DT model:

- tree depth;
- distribution of the leaves in the tree;
- number of nodes.

Instance Features (F)

The **landmarking meta-features** are the performance of a set of fast and simple learners:

- Linear Discriminant;
- Elite-Nearest Neighbor;
- One node DT-models.

Algorithms (A)

They represent a set of the algorithms α that will be applied to the datasets $\alpha(p)$ in the algorithm selection process.

- Classifiers, regressors and clustering algorithms [Garcia et al., 2018, Pimentel and de Carvalho, 2019]
- Pre-processing algorithms [Garcia et al., 2016]
- Hyperparameters [Mantovani et al., 2015]
- Optimization [Kanda et al., 2011]
- ...

Evaluation Measures (Y)

The models induced by the algorithm α can be evaluated by different measures to the datasets $y(\alpha(p))$. They are mainly:

- Accuracy, F_β, AUC and kappa for classification;
- MSE, RMSE for regression problems;
- ...

Meta-base (S)

The meta-base S is a collection of meta-examples. A meta-example is the characterization measures from the datasets $f(p)$ plus the evaluation of the algorithms $y(\alpha(p))$ for these dataset.

Meta-base (S)

Meta-{classification, regression and ranking}:

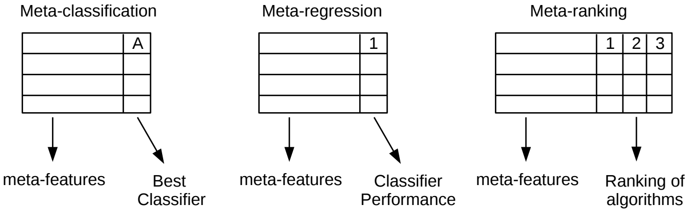


Figure: Example of meta-bases.

Recommendation System based on MtL

Predicting the classifier performance:

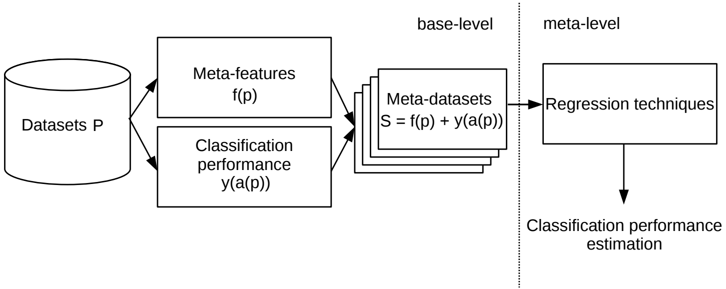


Figure: Example of MtL system to predict classifiers performance.

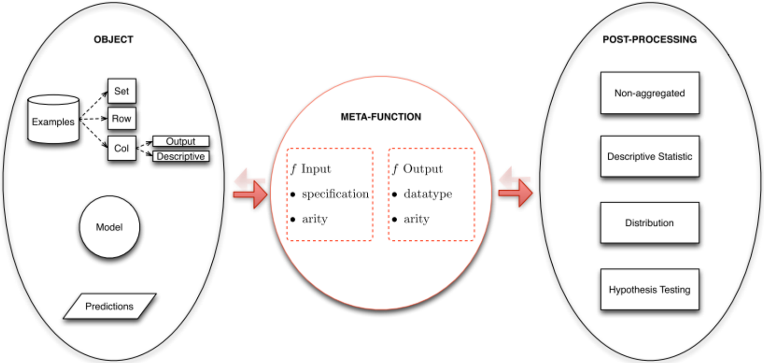
Outline

1. Introduction
2. Meta-Learning
3. Meta-Feature Extractor (MFE)
4. Standard Analysis
5. Prospective work

Introduction

- An R (MFE) and Python (PyMFE) package for dataset characterization
- Contains the implementation of the main characterization measures available
- It also can be used for analyzing of machine learning algorithms' performance

Software Architecture



PyMFE

https://pymfe.readthedocs.io/en/latest/

Open Issues

- Development of new characterization measures
- Understanding the behavior/semantic of current measures
- Development of techniques for features reduction and summarization
- Development of tools
- Creation of synthetic datasets based on specific features values range

Outline

1 Introduction

2 Meta-Learning

3 Meta-Feature Extractor (MFE)

4 Standard Analysis

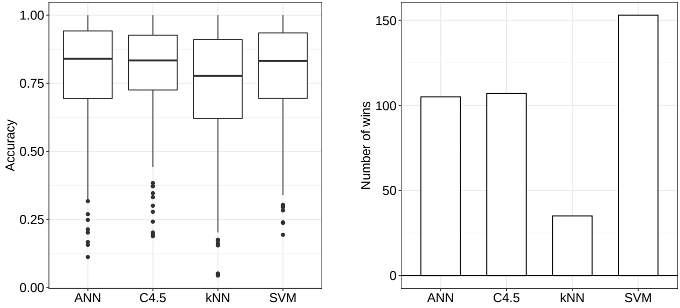
5 Prospective work

Standard Analysis

Evaluating the MtL to predict the classifier performance:

- **Meta-base Analysis:** Distribution of the algorithms in the meta-base and etc...
- **Meta-level Analysis:** Error of the meta-regressors to predict the performance of each classifier.
- **Base-level Analysis:** Performance of the meta-regressors to predict the best classifier for a dataset.
- **Execution time:** Difference of execution time between trial-and-error and MtL approach.

Meta-base Analysis



(a) Distribution of accuracies. (b) Winning classifiers.

Figure: Performance of the base-classifiers.

Meta-level Analysis

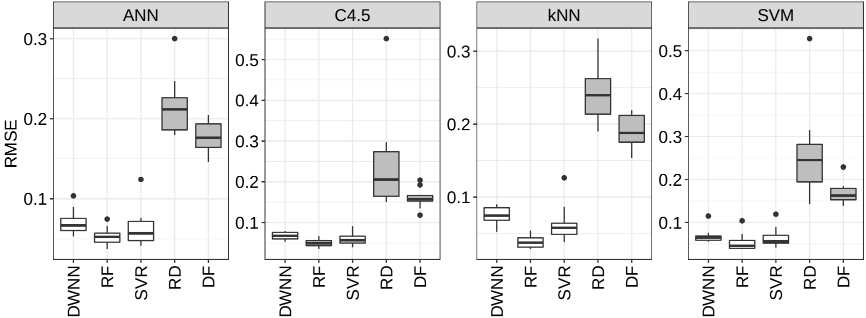


Figure: RMSE of each meta-regressor for each classifier.

Base-level Analysis

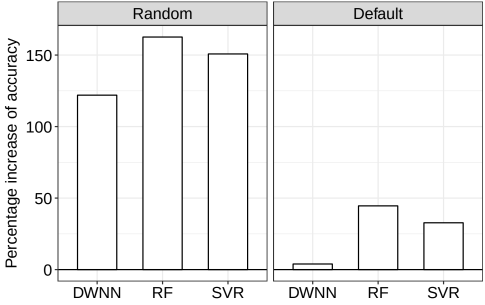
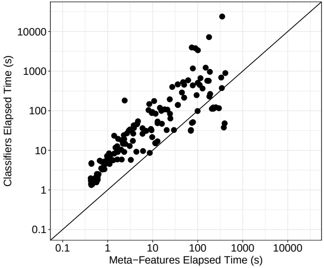
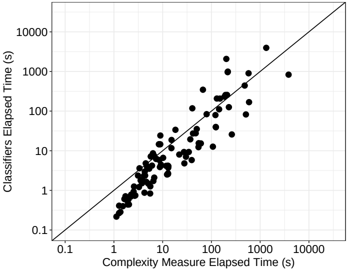


Figure: Improvement of base-classifier accuracies over baselines.

Execution time



(a) Average time elapsed to compute the meta-features and classifiers.



(b) Average time elapsed to compute the complexity measures and classifiers.

Meta-features Importance

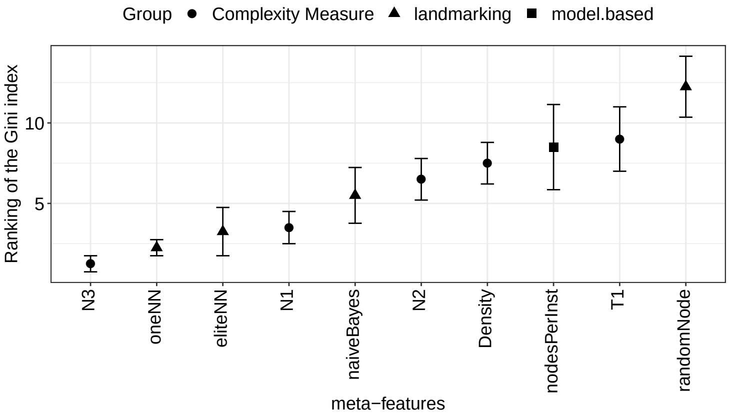


Figure: Top-ranked meta-features selected by the RF meta-regressor

Outline

1. Introduction
2. Meta-Learning
3. Meta-Feature Extractor (MFE)
4. Standard Analysis
5. Prospective work

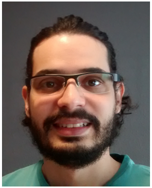
Prospective work

Main interests:

- Proposing a framework to extract meta-features;
- Simulating the Complexity Measures;
- Investigating new measures like Clustering Indexes and types of model-based
- Constructing meta-models for AutoML;
- Solving real problems with MtL.

Collaborations


Andre (USP)


Luís Paulo (UNB)


Davi (USP)


Carlos (FEUP)

MtL for AutoML

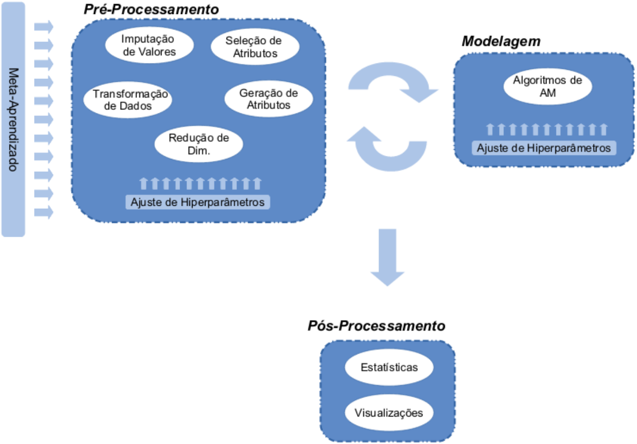


Figure: Defining AutoML pipelines with MtL.

Prospective work

Main papers

- Rivolli, A., Garcia, L., Soares, C., Vanschoren, J., de Carvalho, A., (2022). "Meta-features for Meta-learning". Knowledge-Based Systems
- Pio, P., Garcia, L. P. F., Rivolli, A. (2022). Meta-Learning Approach for Noise Filter Algorithm Recommendation. Knowledge Discovery, Mining and Learning (KDMiLe)
- Meskhi, M. M., Rivolli, A., Mantovani, R. G., Vilalta, R. (2021). Learning abstract task representations. AAAI Workshop on Meta-Learning and MetaDL Challenge
- Rivolli, A., Garcia, L. P. F., Lorena, A. C., Carvalho, A. C. P. L. F. (2021). A Study of the Correlation of Metafeatures Used for Metalearning. International Work-Conference on Artificial Neural Networks
- Garcia, L. P. F., Campelo, F., Ramos, G. N., Rivolli, A., Carvalho, A. C. P. L. F. (2021) Evaluating Clustering Meta-features for Classifier Recommendation. Brazilian Conference on Intelligent Systems
- Alcobaça, E., Siqueira, F., Garcia, L., Rivolli, A., & de Carvalho, A. (2020). "MFE: Towards reproducible meta-feature extraction". Journal of Machine Learning Research.
- Garcia, L., Rivolli, A., Alcobaça, E., Lorena, A., & de Carvalho, A. (2020). "Boosting Meta-Learning with Simulated Data Complexity Measures." Intelligent Data Analysis

Prospective work

Packages

- Rivolli, A., Garcia, L., & de Carvalho, A. (2017). "mfe: Meta-Feature Extractor". R package version 0.1.3. https://CRAN.R-project.org/package=mfe.
- Alcobaça, E., Siqueira, F., Garcia, L., & de Carvalho, A. (2019). "pymfe: Python Meta-Feature Extractor". Python package version 0.0.3. https://pypi.org/project/pymfe/.

Acknowledgements

References I

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., and García, S. (2011).
Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework.
Multiple-Valued Logic and Soft Computing, 17(2-3):255–287.

Brazdil, P., Giraud-Carrier, C. G., Soares, C., and Vilalta, R. (2009).
Metalearning - Applications to Data Mining.
Springer, 1 edition.

de Souza, B. F., de Carvalho, A. C. P. L. F., and Soares, C. (2010).
Empirical evaluation of ranking prediction methods for gene expression data classification.
In *12th Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, volume 6433, pages 194–203.

Garcia, L. P. F., Lorena, A. C., and de Carvalho, A. C. P. L. F. (2016).
Ensembles of label noise filters: a ranking approach.
Data Mining and Knowledge Discovery, 30(5):1192 – 1216.

Garcia, L. P. F., Lorena, A. C., de Souto, M. C. P., and Ho, T. K. (2018).
Classifier recommendation using data complexity measures.
In *24th International Conference on Pattern Recognition (ICPR)*, pages 874–879.

Kanda, J., de Carvalho, A. C. P. L. F., Hruschka, E. R., and Soares, C. (2011).
Selection of algorithms to solve traveling salesman problems using meta-learning.
International Journal of Hybrid Intelligent Systems, 8(3):117–128.

Lichman, M. (2013).
UCI machine learning repository.
http://archive.ics.uci.edu/ml.

References II

Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., and de Carvalho, A. C. P. L. F. (2015).
To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning.
In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Mitchell, T. M. (1997).
Machine Learning.
McGraw Hill series in computer science. McGraw Hill.

Pimentel, B. A. and de Carvalho, A. C. P. L. F. (2019).
A new data characterization for selecting clustering algorithms using meta-learning.
Information Sciences, 477:203 – 219.

Prudêncio, R. B. C., Soares, C., and Ludermir, T. B. (2011).
Uncertainty sampling-based active selection of datasetoids for meta-learning.
In *21th International Conference on Artificial Neural Networks (ICANN)*, volume 6792, pages 454–461.

Rossi, A. L. D., de Carvalho, A. C. P. L. F., Soares, C., and de Souza, B. F. (2014).
Metastream: A meta-learning based method for periodic algorithm selection in time-changing data.
Neurocomputing, 127:52–64.

Smith-Miles, K. A. (2008).
Cross-disciplinary perspectives on meta-learning for algorithm selection.
ACM Computing Surveys, 41(1):6:1 – 6:25.

Soares, C., Petrak, J., and Brazdil, P. (2001).
Sampling-based relative landmarks: Systematically test-driving algorithms before choosing.
In *10th Portuguese Conference on Artificial Intelligence (EPIA)*, pages 88 – 95.

References III

Vanschoren, J. and Blockeel, H. (2006).
Towards understanding learning behavior.
In *15th Annual Machine Learning Conference of Belgium and the Netherlands*, pages 89–96.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. (2013).
Openml: Networked science in machine learning.
SIGKDD Explorations, 15(2):49–60.