

# Uma introdução ao Meta-aprendizado

Prof. Ana C. Lorena e Prof. Luís Paulo F. Garcia

Divisão de Ciência da Computação (IEC)  
Instituto Tecnológico de Aeronáutica (ITA)  
aclorena@ita.br

Departamento de Ciência da Computação (CIC)  
Universidade de Brasília (UnB)  
luís.garcia@unb.br

Agosto de 2020



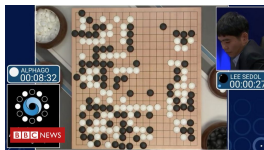
# Agenda

- 1 Introdução
- 2 Meta-Aprendizado
- 3 Análise de resultados
- 4 Conclusões

# Introdução

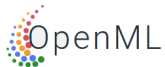
O uso de Aprendizado de Máquina (AM) tem se popularizado, com casos de sucesso reportados em diversas áreas:

- Sistemas de recomendação;
- Segmentação de clientes;
- Identificação de fraudes;
- Robótica;
- Visão computacional;
- Processamento de Linguagem Natural;
- Jogos;
- Etc.



# Introdução

E existem vários repositórios com conjuntos de dados e comparações de algoritmos de AM:



Machine learning, better, together



**KDD-BR** 2020

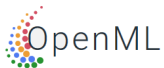
Predicting unavailability of cars in a car rental agency

<https://www.kaggle.com/c/kddbr-2020>



# Introdução

E existem vários repositórios com conjuntos de dados e comparações de algoritmos de AM:



Machine learning, better, together



**KDD-BR** 2020

Predicting unavailability of cars in a car rental agency

<https://www.kaggle.com/c/kddbr-2020>



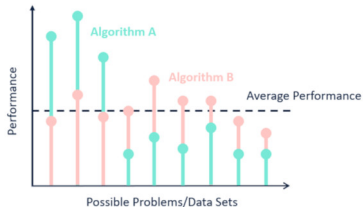
Meta-aprendizado [Vilalta and Drissi, 2002, Vanschoren, 2019]

Extrair (meta-)conhecimento a partir de tarefas resolvidas com sucesso por AM, para usá-lo na solução de novas tarefas.

# Introdução

## *No-free lunch theorem* [Wolpert, 2002, Adam et al., 2019]

- O desempenho médio de dois algoritmos de AM é o mesmo, considerando todos os possíveis problemas.
- Ou seja, não existe um único algoritmo que será adequado para todas as situações.



**Figura:** Sydney Firmin, Alteryx, *There is No Free Lunch in Data Science*, KDDNuggets

# Viés

Cada algoritmo tem um viés que pode ser adequado a uma determinada classe de problemas e não a outra.

## Definição de viés [Mitchell, 1997]

Preferência pela escolha de uma hipótese específica sobre outras, restringindo o **espaço de busca e de representação**, e tornando o aprendizado a partir de dados possível.

# Viés

Cada algoritmo tem um viés que pode ser adequado a uma determinada classe de problemas e não a outra.

## Definição de viés [Mitchell, 1997]

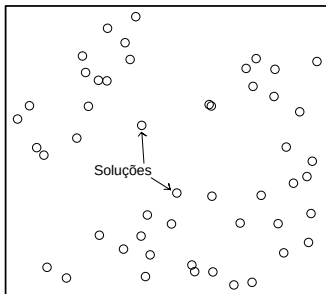
Preferência pela escolha de uma hipótese específica sobre outras, restringindo o **espaço de busca e de representação**, e tornando o aprendizado a partir de dados possível.

## Atenção:

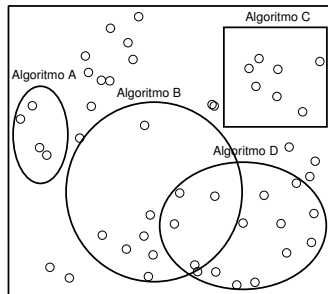
Sem viés não há generalização e, conseqüentemente, aprendizado!



# Hipóteses e espaço de busca



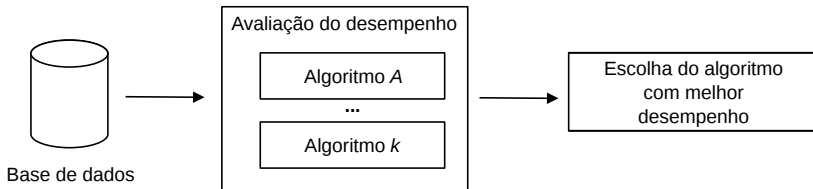
(a) Espaço de busca.



(b) Exemplos de possíveis vieses de alguns algoritmos.

# Necessidade de experimentos

- Dessa forma, geralmente a escolha de algoritmos (e de seus hiper-parâmetros) envolve diversos experimentos.
- Em geral, em uma abordagem de **tentativa e erro**.



# Abordagem de tentativa e erro

- É custosa e subjetiva, dependendo de conhecimento do usuário para definir o que testar;

# Abordagem de tentativa e erro

- É custosa e subjetiva, dependendo de conhecimento do usuário para definir o que testar;
- Aumenta o tempo de geração dos modelos consideravelmente;

# Abordagem de tentativa e erro

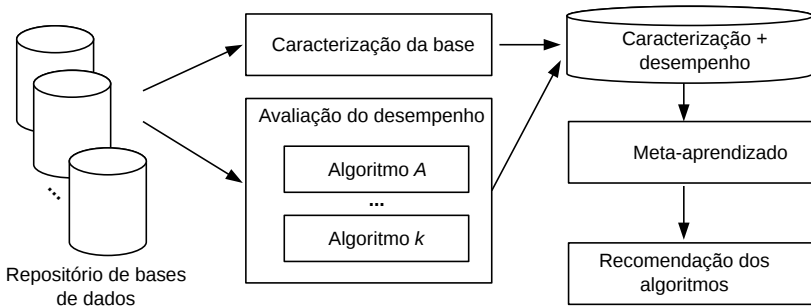
- É custosa e subjetiva, dependendo de conhecimento do usuário para definir o que testar;
- Aumenta o tempo de geração dos modelos consideravelmente;
- Pode levar a super-ajustes (*overfitting*);

# Abordagem de tentativa e erro

- É custosa e subjetiva, dependendo de conhecimento do usuário para definir o que testar;
- Aumenta o tempo de geração dos modelos consideravelmente;
- Pode levar a super-ajustes (*overfitting*);
- Os resultados tornam-se mais difíceis de serem reproduzidos.

# Abordagem de Meta-aprendizado (MtL)

- Usando conhecimento sobre outros problemas para solucionar um novo...



# Abordagem de Meta-aprendizado (MtL)

- Também é trabalhosa, considerando que é necessário ao menos caracterizar uma nova base de dados, mas é objetiva;



# Abordagem de Meta-aprendizado (MtL)

- Também é trabalhosa, considerando que é necessário ao menos caracterizar uma nova base de dados, mas é objetiva;
- O uso do meta-modelo produzido é rápido, eliminando necessidade de treinar múltiplos modelos para um novo conjunto de dados;

# Abordagem de Meta-aprendizado (MtL)

- Também é trabalhosa, considerando que é necessário ao menos caracterizar uma nova base de dados, mas é objetiva;
- O uso do meta-modelo produzido é rápido, eliminando necessidade de treinar múltiplos modelos para um novo conjunto de dados;
- Pode auxiliar a evitar super-ajustes;

# Abordagem de Meta-aprendizado (MtL)

- Também é trabalhosa, considerando que é necessário ao menos caracterizar uma nova base de dados, mas é objetiva;
- O uso do meta-modelo produzido é rápido, eliminando necessidade de treinar múltiplos modelos para um novo conjunto de dados;
- Pode auxiliar a evitar super-ajustes;
- Melhora reprodutibilidade de resultados experimentais;

# Abordagem de Meta-aprendizado (MtL)

- Também é trabalhosa, considerando que é necessário ao menos caracterizar uma nova base de dados, mas é objetiva;
- O uso do meta-modelo produzido é rápido, eliminando necessidade de treinar múltiplos modelos para um novo conjunto de dados;
- Pode auxiliar a evitar super-ajustes;
- Melhora reprodutibilidade de resultados experimentais;
- Auxilia usuários não especializados.

# Alguns pontos importantes

- A partir de que **problemas** o meta-conhecimento será extraído?

# Alguns pontos importantes

- A partir de que **problemas** o meta-conhecimento será extraído?
- Que **características** extrair dos conjuntos de dados?

# Alguns pontos importantes

- A partir de que **problemas** o meta-conhecimento será extraído?
- Que **características** extrair dos conjuntos de dados?
- Que **algoritmos** considerar?

# Alguns pontos importantes

- A partir de que **problemas** o meta-conhecimento será extraído?
- Que **características** extrair dos conjuntos de dados?
- Que **algoritmos** considerar?
- Que **medida de desempenho** será usada na avaliação dos algoritmos?



# Alguns pontos importantes

- A partir de que **problemas** o meta-conhecimento será extraído?
- Que **características** extrair dos conjuntos de dados?
- Que **algoritmos** considerar?
- Que **medida de desempenho** será usada na avaliação dos algoritmos?
- Como será gerado o **meta-modelo**?

# Alguns pontos importantes

- A partir de que **problemas** o meta-conhecimento será extraído?
- Que **características** extrair dos conjuntos de dados?
- Que **algoritmos** considerar?
- Que **medida de desempenho** será usada na avaliação dos algoritmos?
- Como será gerado o **meta-modelo**?
- E como avaliar os resultados obtidos?

# Outline

- 1 Introdução
- 2 Meta-Aprendizado**
- 3 Análise de resultados
- 4 Conclusões

# Meta-aprendizado

## Definição [Brazdil et al., 2009]

Métodos que exploram **meta-conhecimento** para obter soluções de AM mais eficientes e melhores.

# Meta-aprendizado

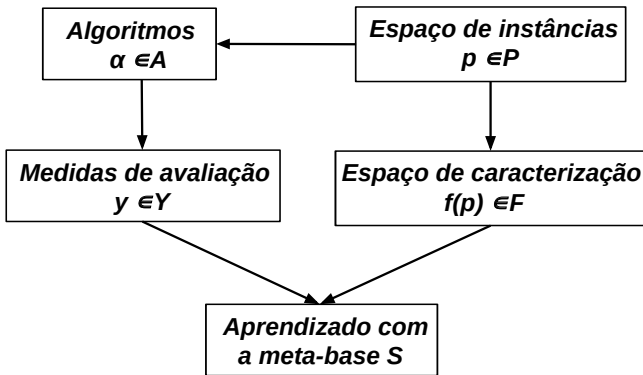
## Definição [Brazdil et al., 2009]

Métodos que exploram **meta-conhecimento** para obter soluções de AM mais eficientes e melhores.

Algumas aplicações:

- Otimização [Kanda et al., 2011];
- Análise de séries temporais [Rossi et al., 2014];
- Recomendação de técnicas de pré-processamento [Garcia et al., 2016a];
- Seleção de classificadores [Garcia et al., 2018a];
- Ajuste de hiper-parâmetros de SVM [Mantovani et al., 2015].

# Arcabouço de Seleção de Algoritmos



**Figura:** Arcabouço de seleção de algoritmos, adaptado de [Smith-Miles, 2008] com base em [Rice, 1976]

# Instâncias do problema ( $P$ )

Em AM,  $P$  será composto por diversos conjuntos de dados  $p$ .

# Instâncias do problema ( $P$ )

Em AM,  $P$  será composto por diversos conjuntos de dados  $p$ .

- Podem ser coletados de repositórios como:
  - UCI [Lichman, 2013];
  - Keel [Alcalá-Fdez et al., 2011];
  - OpenML [Vanschoren et al., 2013];
  - Competições.



# Instâncias do problema ( $P$ )

Em AM,  $P$  será composto por diversos conjuntos de dados  $p$ .

- Podem ser coletados de repositórios como:
  - UCI [Lichman, 2013];
  - Keel [Alcalá-Fdez et al., 2011];
  - OpenML [Vanschoren et al., 2013];
  - Competições.
- E/ou também podem ser gerados:
  - Conjuntos sintéticos [Vanschoren and Blockeel, 2006, Macià and Bernadó-Mansilla, 2014, Muñoz et al., 2018];
  - Datasetoids [Soares, 2009, Prudêncio et al., 2011].

# Instâncias do problema ( $P$ )

Em AM,  $P$  será composto por diversos conjuntos de dados  $p$ .

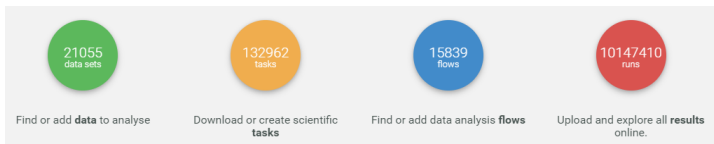
- Podem ser coletados de repositórios como:
  - UCI [Lichman, 2013];
  - Keel [Alcalá-Fdez et al., 2011];
  - OpenML [Vanschoren et al., 2013];
  - Competições.
- E/ou também podem ser gerados:
  - Conjuntos sintéticos [Vanschoren and Blockeel, 2006, Macià and Bernadó-Mansilla, 2014, Muñoz et al., 2018];
  - Datasetoids [Soares, 2009, Prudêncio et al., 2011].

## Objetivo:

Ter diversidade de instâncias do problema.

# Instâncias do problema ( $P$ )

Exemplo: OpenML [Vanschoren et al., 2013]



Explore

Data 3120

Task

Flow

Run

Study

Task type 8

Measure

People

Help

Blog

Contact

Please cite us

3120 results

Only showing active (verified) datasets.

FILTERS

SORT: MOST RUNS

ID'S

TABLE

+ ADD NEW

credit-g (1)

This dataset classifies people described by a set of attributes as good or bad credit risks. This dataset comes with a cost matrix: "" Good Bad (predicted) Good 0 1 (L...

505683 runs ♥ 19 likes ⬇ 227 downloads ⬆ 246 reach ⬆ 22 impact  
1005 instances - 21 features - 2 classes - 0 missing values

blood-transfusion-service-c...

Data taken from the Blood Transfusion Service Center in Hsin-Chu City in Taiwan - this is a classification problem. To demonstrate the RFMTC marketing model (a m...

467724 runs ♥ 5 likes ⬇ 83 downloads ⬆ 85 reach ⬆ 32 impact  
740 instances - 5 features - 2 classes - 0 missing values

monks-problems-2 (1)

Once upon a time, in July 1991, the monks of Concordia Priory were faced with a school held in their priory, namely the 2nd European Summer School on Machine ...

314292 runs ♥ 2 likes ⬇ 22 downloads ⬆ 24 reach ⬆ 21 impact  
601 instances - 7 features - 2 classes - 0 missing values

tic-tac-toe (1)

This database encodes the complete set of possible board configurations at the end of tic-tac-toe games, where 'x' is assumed to have played first. The target conse...

336327 runs ♥ 2 likes ⬇ 74 downloads ⬆ 76 reach ⬆ 4 impact  
958 instances - 10 features - 2 classes - 0 missing values

monks-problems-1 (1)

Once upon a time, in July 1991, the monks of Concordia Priory were faced with a school held in their priory, namely the 2nd European Summer School on Machine ...

338449 runs ♥ 2 likes ⬇ 18 downloads ⬆ 20 reach ⬆ 20 impact  
656 instances - 7 features - 2 classes - 0 missing values

steel-plates-fault (1)

A dataset of steel plates' faults, classified into 7 different types. The goal was to train machine learning for automatic pattern recognition. The dataset consists of 27...

277313 runs ♥ 1 likes ⬇ 42 downloads ⬆ 42 reach ⬆ 19 impact  
1941 instances - 34 features - 2 classes - 0 missing values

kings-hip (1)

1. Title: Chess End-Game - Kingfisher versus KingPawn on a7 (usually abbreviated KRPA7). The pawn on a7 means it is one square away from queening. It is the ...

273452 runs ♥ 0 likes ⬇ 34 downloads ⬆ 36 reach ⬆ 7 impact  
3198 instances - 37 features - 2 classes - 0 missing values

# Prática: coletando bases de dados do OpenML

[https://lpfgarcia.github.io/mtl/coletando\\_bases](https://lpfgarcia.github.io/mtl/coletando_bases)



# Meta-atributos ( $F$ )

- Extraem propriedades gerais dos conjuntos de dados  $p$ , na forma  $f(p)$ .
- Devem prover evidência sobre o desempenho futuro das técnicas investigadas [Soares et al., 2001].
- E distinguir bem problemas com diferentes níveis de dificuldade [Smith-Miles, 2008].
- Idealmente, devem ter baixo custo computacional.

# Meta-atributos ( $F$ )

Os principais grupos de meta-atributos são:

- **Gerais:** informações simples e básicas dos conjuntos de dados;
- **Estatísticos:** indicadores da distribuição dos dados;
- **Informação:** baseados em conceitos de teoria da informação;
- **Baseados em modelos:** usam modelos simples nos dados e extraem características deles.
- **Landmarking:** desempenho de algoritmos simples e eficientes nos dados.
- **Complexidade:** buscam capturar o nível de dificuldade associado à resolução do problema.

# Meta-atributos ( $F$ )

Os meta-atributos **gerais** compreendem informações básicas extraídas dos conjuntos de dados:

- números de instâncias, de atributos, de classes;
- porcentagem de instâncias em cada classe.

## Exemplos:

### Conjunto 'iris' (classificação)

- # instâncias = 150;
- # atributos = 4;
- # classes = 3.

### Conjunto 'swiss' (regressão)

- # instâncias = 47;
- # atributos = 5;

# Meta-atributos ( $F$ )

Os meta-atributos **estatísticos** extraem informações sobre a distribuição dos dados:

- matrizes de correlação e covariância dos atributos;
- obliquidade e curtose dos atributos quantitativos.

## Exemplos:

### Conjunto 'iris'

- obliquidade média = 0.063;
- curtose média = -0.811;

### Conjunto 'swiss'

- obliquidade média = 0.484;
- curtose média = 0.742;



# Meta-atributos ( $F$ )

As medidas baseadas em **teoria da informação** capturam a quantidade de informação no conjunto de dados, por meio de relações entre os atributos.

- Entropia;
- Informação mútua;
- Razão sinal-ruído.

## Exemplos:

Conjunto 'iris'

- informação mútua total = 3.296;

Conjunto 'swiss'

- informação mútua total = 2.823;

# Meta-atributos ( $F$ )

Os meta-atributos baseados em **modelos** extraem informações sobre modelos simples extraídos dos dados, tais como de uma Árvore de Decisão:

- Profundidade da árvore;
- Distribuição das folhas;
- Número de nós.

## Exemplos:

### Conjunto 'iris'

- profundidade AD = 3;
- # nós = 5.

### Conjunto 'swiss'

- profundidade AR = 4;
- # nós = 7.

# Meta-atributos ( $F$ )

Os meta-atributos de **landmarking** provêm medidas do desempenho de modelos simples aplicados aos conjuntos de dados:

- Modelos lineares;
- *Elite-Nearest Neighbor*;
- ADs com um único nó (*Decision Stumps*).

## Exemplos:

### Conjunto 'iris'

- acurácia do discriminante linear = 0.98;
- acurácia do eliteNN = 0.94.

### Conjunto 'swiss'

- MSE do regressão linear = 41.24;
- MSE so eliteNN = 12.07.

# Prática: caracterizando bases de dados

<https://lpfgarcia.github.io/mtl/tradicionais>



# Meta-atributos ( $F$ )

Os meta-atributos de **complexidade** tentam estimar a complexidade do modelo que deve ser ajustado aos dados [Lorena et al., 2018, Lorena et al., 2019]:

- Porcentagem de pontos na fronteira (N1);
- Razão de distâncias médias intra e extra-classe (N2);
- Suavidade da distribuição das saídas (S1);
- Suavidade da distribuição das entradas (S2).

## Exemplos:

### Conjunto 'iris'

- $N1 = 0.10$ ;
- $N2 = 0.19$ .

### Conjunto 'swiss'

- $S1 = 0.12$ ;
- $S2 = 0.25$ .

# Prática: caracterizando bases de dados

<https://lpfgarcia.github.io/mtl/complexidade>



# Algoritmos ( $A$ )

Conjunto de algoritmos  $\alpha$  que serão aplicados aos conjuntos de dados ( $\alpha(p)$ ), candidatos no processo de seleção de algoritmos.

# Algoritmos ( $A$ )

Conjunto de algoritmos  $\alpha$  que serão aplicados aos conjuntos de dados ( $\alpha(p)$ ), candidatos no processo de seleção de algoritmos.

- Classificadores, regressores, algoritmos de agrupamento [Garcia et al., 2018b, Pimentel and de Carvalho, 2019]
- Algoritmos de pré-processamento de dados [Garcia et al., 2016b]
- Hyperparâmetros de algoritmos [Mantovani et al., 2019]
- Algoritmos de otimização [Kanda et al., 2016]
- Análise de séries temporais [Rossi et al., 2017]
- entre outros ...



# Algoritmos (*A*)

- *Random Forest* [Breiman, 2001, Ho, 1998]
- Redes Neurais Artificiais [Haykin et al., 2009]
- Árvores de Decisão [Quinlan, 1986]
- k-vizinhos mais próximos [Cover and Hart, 1967]
- *Support Vector Machine* [Schlkopf et al., 2018]
- ...

# Medidas de Avaliação ( $Y$ )

Avaliação dos modelos obtidos pelos algoritmos  $\alpha$  nos conjuntos de dados coletados  $y(\alpha(p))$

- Classificação: acurácia,  $F_\beta$ , AUC, kappa, etc.
- Regressão: MSE, RMSE, NMSE, etc.
- Agrupamento: ARI, NMI, etc.
- ...

# Prática: induzindo modelos de classificação

<https://lpfgarcia.github.io/mtl/classificadores>



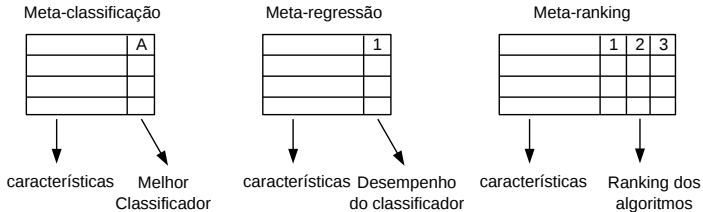
# Meta-modelo ( $S$ )

Uma meta-base é formada a partir da coleção de meta-exemplos:

- Cada meta-exemplo é representado pelas medidas de caracterização  $f(p)$
- E é rotulado de acordo com o resultado da avaliação dos algoritmos  $y(\alpha(p))$

# Meta-modelo ( $S$ )

Meta-{classificação, regressão e ranqueamento}:



Obtendo  $S$  a partir de uma meta-base:

Um novo problema de aprendizado!

# Sistema de recomendação baseado em MtL

Usar técnicas de AM para induzir meta-modelos a partir da meta-base

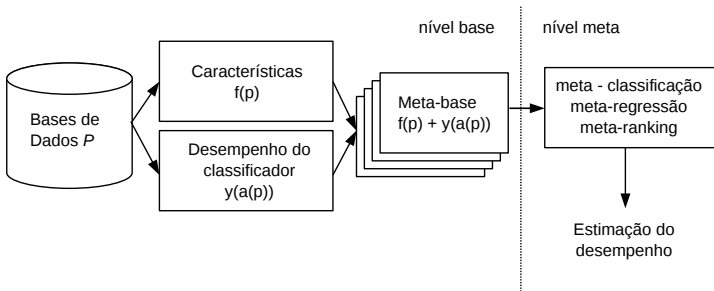
## Procedimento:

Dado um novo conjunto de dados  $p$ :

- Extração dos meta-atributos  $f(p)$
- Consulta ao meta-modelo  $S(f(p))$

# Sistema de recomendação baseado em MtL

Exemplo: recomendando um algoritmo



# Prática: gerando a meta-base

[https://lpfgarcia.github.io/mtl/meta\\_base](https://lpfgarcia.github.io/mtl/meta_base)





# Outline

- 1 Introdução
- 2 Meta-Aprendizado
- 3 Análise de resultados
- 4 Conclusões

# Análise padrão

Avaliando o meta-modelo:

- **Análise da meta-base:** distribuição dos algoritmos na meta-base, etc.

# Análise padrão

Avaliando o meta-modelo:

- **Análise da meta-base:** distribuição dos algoritmos na meta-base, etc.
- **Análise no nível meta:** desempenho dos meta-classificadores na predição do melhor classificador.

# Análise padrão

Avaliando o meta-modelo:

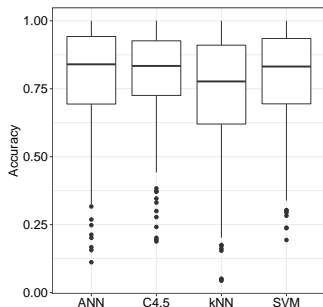
- **Análise da meta-base:** distribuição dos algoritmos na meta-base, etc.
- **Análise no nível meta:** desempenho dos meta-classificadores na predição do melhor classificador.
- **Análise no nível base:** ganho acumulado que os meta-classificadores geram em comparação com a classe majoritária.

# Análise padrão

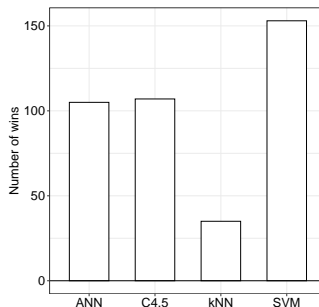
Avaliando o meta-modelo:

- **Análise da meta-base:** distribuição dos algoritmos na meta-base, etc.
- **Análise no nível meta:** desempenho dos meta-classificadores na predição do melhor classificador.
- **Análise no nível base:** ganho acumulado que os meta-classificadores geram em comparação com a classe majoritária.
- **Tempo de execução:** diferença de tempo entre executar MtL e uma abordagem de tentativa e erro para um novo conjunto de dados.

# Análise da meta-base



(a) Distribuição de acurácias.



(b) Classificadores ganhadores.

**Figura:** Desempenho dos classificadores base.

# Análise de nível meta

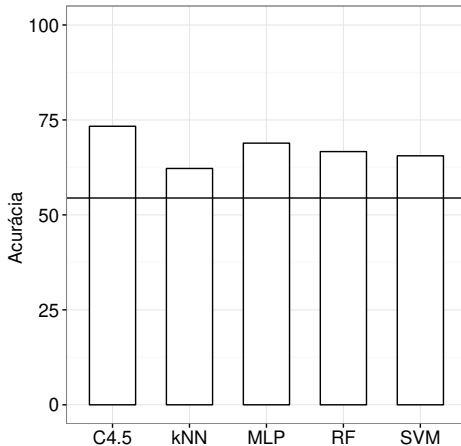
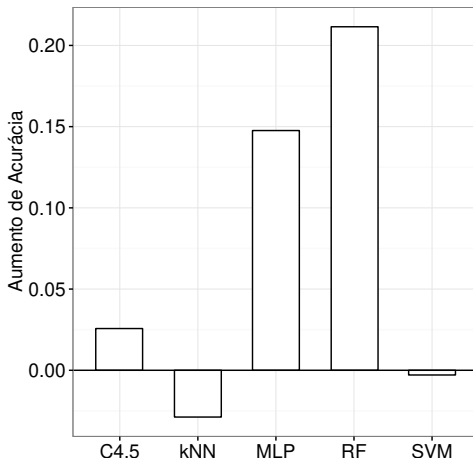


Figura: Desempenho dos meta-classificadores.

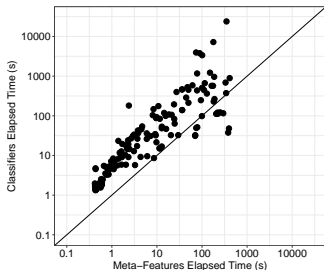
# Análise de nível base



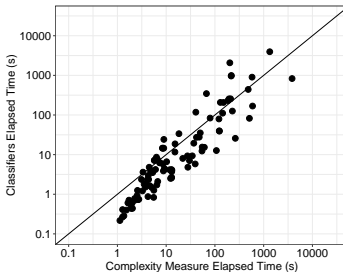
**Figura:** Melhora na acurácia dos classificadores de base sobre *baselines*.



# Tempo de execução



(a) Tempo médio para computar as meta-atributos e consultar os classificadores.



(b) Tempo médio para computar as medidas de complexidade e consultar os classificadores.

# Importância dos meta-atributos

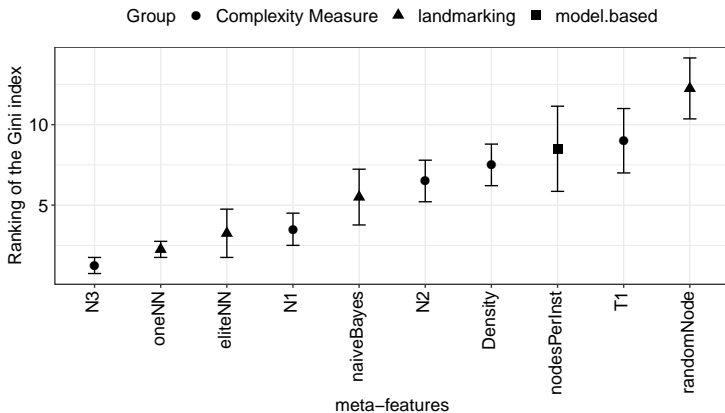


Figura: Meta-atributos melhor ranqueados (meta-regressor RF)

# Outline

- 1 Introdução
- 2 Meta-Aprendizado
- 3 Análise de resultados
- 4 **Conclusões**

# Conclusões

## Principais contribuições de MtL:

- Auxiliar em processos decisórios no *pipeline* de AM, reduzindo tentativas-e-erros
- Prover um melhor entendimento de que tipos de problemas cada técnica de AM possui melhor desempenho
- Melhor entendimento dos conjuntos de dados e de suas semelhanças/diferenças, pelo exame de suas características

# Conclusões

## Principais limitações de MtL:

- Escolhas de  $P$ ,  $F$ ,  $A$ ,  $Y$
- Indução de meta-modelos  $S$  é um problema de AM por si só, que também implica na escolha de um algoritmo e seus hiper-parâmetros
- A obtenção de  $S$  é custosa, embora seu uso seja rápido

# Conclusão

## Alguns pacotes de MtL

- <https://CRAN.R-project.org/package=ECoL>
- <https://CRAN.R-project.org/package=mfe>
- <https://pypi.org/project/pymfe/>
- <https://github.com/byu-dml/metalearn>
- <https://github.com/openml/openml-python>
- <https://github.com/MichaelMMeskhi/MtL-Progress-github.io>

# Agradecimentos



# References I

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., and Vrahatis, M. N. (2019).

No free lunch theorem: a review.

In *Approximation and Optimization*, pages 57–82. Springer.

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.

Breiman, L. (2001).

Random forests.

*Machine learning*, 45(1):5–32.

Cover, T. and Hart, P. (1967).

Nearest neighbor pattern classification.

*IEEE transactions on information theory*, 13(1):21–27.

Garcia, L. P., de Carvalho, A. C., and Lorena, A. C. (2016a).

Noise detection in the meta-learning level.

*Neurocomputing*, 176:14–25.

Garcia, L. P., Lorena, A. C., de Souto, M. C., and Ho, T. K. (2018a).

Classifier recommendation using data complexity measures.

In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 874–879. IEEE.



# References II

Garcia, L. P. F., Lorena, A. C., and de Carvalho, A. C. P. L. F. (2016b).

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. (2018b).

Classifier recommendation using data complexity measures.

In *24th International Conference on Pattern Recognition (ICPR)*, pages 874–879.

Haykin, S. S. et al. (2009).

Neural networks and learning machines/simon haykin.

Ho, T. K. (1998).

The random subspace method for constructing decision forests.

*IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

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.

Kanda, J., de Carvalho, A. C. P. L. F., Hruschka, E. R., Soares, C., and Brazdil, P. (2016).

Meta-learning to select the best meta-heuristic for the traveling salesman problem: A comparison of meta-features.

*Neurocomputing*, 205:393–406.

Lichman, M. (2013).

UCI machine learning repository.

<http://archive.ics.uci.edu/ml>.

# References III

Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C., and Ho, T. K. (2019).

How complex is your classification problem? a survey on measuring classification complexity.

*ACM Computing Surveys (CSUR)*, 52(5):1–34.

Lorena, A. C., Maciel, A. I., de Miranda, P. B., Costa, I. G., and Prudêncio, R. B. (2018).

Data complexity meta-features for regression problems.

*Machine Learning*, 107(1):209–246.

Macià, N. and Bernadó-Mansilla, E. (2014).

Towards uci+: A mindful repository design.

*Information Sciences*, 261:237–262.

Mantovani, R. G., Rossi, A. L. D., Alcobaça, E., Vanschoren, J., and de Carvalho, A. C. P. L. F. (2019).

A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers.

*Information Sciences*, 501:193–221.

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.

Muñoz, M. A., Villanova, L., Baatar, D., and Smith-Miles, K. (2018).

Instance spaces for machine learning classification.

*Machine Learning*, 107(1):109–147.

# References IV

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.

Quinlan, J. R. (1986).

Induction of decision trees.  
*Machine learning*, 1(1):81–106.

Rice, J. R. (1976).

The algorithm selection problem.  
In *Advances in computers*, volume 15, pages 65–118. Elsevier.

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.

Rossi, A. L. D., de Souza, B. F., Soares, C., and de Carvalho, A. C. P. L. F. (2017).

A guidance of data stream characterization for meta-learning.  
*Intelligent Data Analysis*, 21(4):1015–1035.

Scholkopf, B., Smola, A. J., and Bach, F. (2018).

*Learning with kernels: support vector machines, regularization, optimization, and beyond*.  
the MIT Press.

# References V

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. (2009).

Uci++: Improved support for algorithm selection using datasetoids.

In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 499–506. Springer.

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.

Vanschoren, J. (2019).

Meta-learning.

In *Automated Machine Learning*, pages 35–61. Springer, Cham.

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.

Vilalta, R. and Drissi, Y. (2002).

A perspective view and survey of meta-learning.

*Artificial intelligence review*, 18(2):77–95.

# References VI

Wolpert, D. H. (2002).

The supervised learning no-free-lunch theorems.

In *Soft computing and industry*, pages 25–42. Springer.