Dimensionality Reduction for the Algorithm Recommendation Problem

Edesio Alcobaça *‡, Rafael G. Mantovani *‡, André L. D. Rossi †, André C. P. L. F. de Carvalho *
‡ The authors have collaborated equally
* Institute of Mathematics and Computer Sciences, University of São Paulo (ICMC/USP), São Carlos, SP, Brazil
† São Paulo State University (UNESP), Campus of Itapeva, SP, Brazil
E-mail: edesio@usp.br, rgmantovani@usp.br, andre.rossi@unesp.br, andre@icmc.usp.br

Abstract—**Given the increase in data generation, as many algorithms have become available in recent years, the algorithm recommendation problem has attracted increasing attention in Machine Learning. This problem has been addressed in the Machine Learning community as a learning task at the meta-level where the most suitable algorithm has to be recommended for a specific dataset. Since it is not trivial to define which characteristics are the most useful for a specific domain, several meta-features have been proposed and used, increasing the meta-data meta-feature dimension. This study investigates the influence of dimensionality reduction techniques on the quality of the algorithm recommendation process. Experiments were carried out with 15 algorithm recommendation problems from the Aslib library, 4 meta-learners, and 3 dimensionality reduction techniques. The experimental results showed that linear aggregation techniques, such as PCA and LDA, can be used in algorithm recommendation problems to reduce the number of meta-features and computational cost without losing predictive performance.**

Index Terms—**Dimensionality reduction, Algorithm recommendation, Meta-learning**

I. Introduction

Due to the increase of using automated systems, Machine Learning (ML) has achieved high popularity in recent years, becoming part of several tools to analyze data and extract useful information from it. As a consequence, the volume of ML applications and experiments have increased, leading to the expansion of Meta-learning (MtL) [1]. MtL investigates the relationship between problems and the performance of ML algorithms when applied to them. This relationship can provide useful information to select the most suitable algorithm for new datasets, commonly referred to as *algorithm recommendation* problem [2].

Many researchers have been proposing recommender systems based on MtL for a wide range of applications, such as to recommend algorithms, to suggest their hyperparameter settings [3], to select noise detection techniques [4] and to speed-up the convergence of optimization techniques [5].

The success of a recommender system depends on how well these problems (datasets) are represented. The MtL literature provides several sets of meta-features proposed to extract data characteristics, varying from simple dataset description measures, like the number of classes and statistics [6], to data complexity measures [3]. More recent approaches go further and propose the systematic and standardized generation of meta-features [7]. Since it is not trivial to define which meta-features better describe the problem represented by a dataset, associated with the many alternatives that can be used to describe a dataset, a large number of meta-features have been proposed. This has brought the "curse of dimensionality" to the meta-level. Besides, it is not difficult to find meta-features computed by different descriptors to be strongly correlated.

Dimensionality Reduction (DR) procedures have been used to reduce the number of features of high-dimensional data to manageable low-dimensional spaces in which analysis can be more effectively performed. They can be divided into two different approaches: Feature Selection (FS) and Feature Extraction (FE) [8] . FS techniques try to find a representative subset of the original features using filters, wrappers or embedding processes. They have extensively been explored in the literature [9].

On the other hand, FE transforms data in the original space of features to a space of fewer dimensions. Principal Component Analysis (PCA) [10] and Linear Discriminant Analysis (LDA) [11] are two traditional FE techniques used to perform linear DR. Recently, a large number of techniques that can deal with complex nonlinear data have been proposed, such as the t-Distributed Stochastic Neighbor Embedding (tSNE) [12]. Although DR have been employed for many domains [13], there is a lack of studies of the effect of these FE techniques in the predictive performance of meta-models induced by ML algorithms in algorithm recommendation tasks.

This study investigates the influence of Dimensionality Reduction (DR) techniques on the predictive performance of ML algorithms employed in algorithm recommendation tasks. For such, experiments were carried out with 15 meta-datasets from "Aslib" repository [2], 3 DR techniques, and 4 different algorithms as meta-learners. The results obtained by each dimensionality reduction technique were also compared with the results obtained using original meta-datasets. In the experiments, the meta-learners were evaluated using default and tuned hyperparameters.

This paper is structured as follows. Section II presents the background on DR and the related works in MtL; section III describes the experimental methodology; the results are discussed in section IV; finally, the conclusions and future works are presented.

II. Background

The *algorithm recommendation problem* was firstly defined by [14]. In [14], the authors defined the idea of selecting an algorithm from a portfolio of options as follows: Given a set $\mathcal{P}$ of problem instances from a distribution $\mathcal{D}$; a space of algorithms $\mathcal{A}$; and a performance measure $\mathcal{M} : \mathcal{P} \times \mathcal{A} \rightarrow \mathbb{R}$; the algorithm recommendation problem is to find a mapping $m : \mathcal{P} \rightarrow \mathcal{A}$ that optimizes the expected performance measure for the instances distributed according to $\mathcal{D}$. In practice, there are some ways to find this mapping between algorithms and problems, and one of them is through the Meta-learning (MtL) [1].

Meta-learning (MtL) is a sub-area of ML that investigates how to exploit past learning experiences in a particular task to improve models and solutions by adapting learning algorithms and data mining processes [1]. This is accomplished using some features extracted from a dataset, named *meta-features*, to represent the main characteristics of a dataset and the performance of ML algorithms when applied to it. This relationship can provide useful information to select the most suitable ML algorithm for new unseen datasets. Thus, ML algorithms are applied to a meta-dataset, whose examples have meta-features as predictive attributes and algorithm performance as a target attribute, to induce a meta-model.

To distinguish this learning process from the conventional application of ML algorithms, it is named meta-level learning. When a ML algorithm recommended by the meta-model is applied to a new dataset, the learning is named base-level learning. Furthermore, the knowledge obtained from past learning tasks is called “meta-knowledge”.

It is often exploited by extracting many meta-features from the datasets and using a learning algorithm to related them to models performance. Commonly measures used to extract these characteristics are grouped into different sets, such as: simple, statistical and information-theoretic measures; model-based; landmarking; data complexity measures, and derived from learning curves [15]. On the other hand, some studies have used meta-features based on properties of the problem domain. For instance, [16] used measures related to propositional satisfiability problem (SAT) problems, such as number of clauses and number of variables.

The suitability of a set of meta-features for a recommendation task is data conformation and problem dependent. Usually, they are extracted in an ad-hoc manner, but more recently systematically generated [7]. However, in both, a large number of meta-features are generated, bringing the “curse of dimensionality” to the meta-level.

A. Dimensionality reduction techniques

The human capacity to extract information from tables is limited as the table dimension increases [17]. An alternative to overcome this limitation is the use of DR techniques, which allow the visualization of high-dimensional data [17]. In ML, DR is mainly used to attenuate the curse of dimensionality and other undesired properties of high-dimensional spaces. DR techniques project the data from a high to a low dimensional feature space. Besides, DR speed-up the learning process and can improve the predictive performance of the induced model.

DR techniques can be categorized as linear or non-linear. The former project the data into a low-dimensional space by a linear transformation, preserving the data linear structures. The benefits of these methods are the easy interpretation because they have solid mathematics definitions, and the fast computation [10], [18].

LDA and PCA are popular examples of linear DR techniques with these characteristics. LDA projects data into a low-dimensional space by maximizing the separation of classes in a supervised way, while PCA performs a rigid rotation by maximizing the data variance captured in a non-supervised way [18].

While linear methods are relatively simple to understanding and computationally inexpensive, they can miss non-linear structures present in the data. On the other hand, non-linear DR techniques, such as tSNE, are computationally more expensive but can deal with complex nonlinear data conformations. tSNE captures non-linear structures by model similarities between data points on joint probabilities and place data points that are similar nearby in the low-dimensional projection [12].

In [19], the authors used PCA with Support Vector Machine (SVM) to extract features from financial datasets. A benchmark of seven DR techniques was performed by [20] using SVMs and k-Nearest Neighbors (kNN) algorithms to classify and visualize sentimental data. PCA was also used by [21] and [5] for feature extraction. The former used PCA for multidimensional unlabeled datasets for change detection while the latter for an Automated Machine Learning (AutoML) tool.

Some studies on algorithm recommendation used DR techniques to improve models predictive performance [22] and for meta-data visualization [23]. However, we did not found any papers evaluating different DR techniques, including PCA, for the algorithm recommendation problem. In some studies, such as [7], the authors proposed to deal with the dimensionality problem of meta-features using correlation and feature selection methods, such as ReliefF [24].

III. Experimental methodology

This section describes the methodology adopted in the experiments carried out to evaluate the DR techniques for algorithm recommendation problems. In addition, it looks for patterns that can support the decision of whether to apply DR and, if needed, which technique should be applied.

Figure 1 provides an overview of the experimental methodology adopted. Firstly, the original meta-datasets are preprocessed by the DR techniques. Then, these meta-datasets are used by different meta-learners for the induction of predictive models. During the learning process, the hyperparameters of the selected algorithms were tuned using a Random Search (RS). The meta-models are applied to the test subset of each meta-dataset and their predictive performance are compared. The next subsections detail these steps.

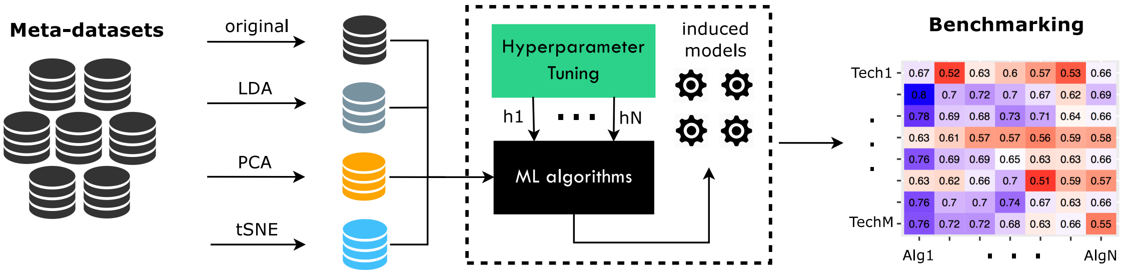


Fig. 1. Experimental methodology adopted to benchmark dimensionality reduction techniques.

A. Meta-datasets

In the experiments we used 15 algorithm recommendation problems from the *Aslib* repository [2]. These meta-datasets are characterized based on problem domain properties, such as propositional satisfiability problem (SAT), traveling salesman problem (TSP), and quantified boolean formula (QBF).

Aslib meta-datasets can be used in classification, regression and clustering tasks. Some of these meta-datasets needed to be preprocessed before the application of DR techniques. Constant and high correlated (corr beq 0.95) meta-features were removed. The remaining meta-features were normalized with Z-Score.

In the experiments we adopted a stratified nested-CV resampling method, considering 10 outer folds (see Section III-C). Thus, meta-classes with very few meta-examples ($<$ 10), referred to as tiny meta-classes, were also dealt with. If the total of meta-examples of these tiny meta-classes exceeds 10, they are joint in a new larger meta-class; otherwise they are discarded. From the available meta-datasets, we selected those with only numerical meta-features, since categorical meta-features would require a more complex preprocessing. Table I summarizes the main characteristics of the 15 meta-datasets.

TABLE I
CLASSIFICATION ASLIB META-DATASETS USED IN THE EXPERIMENTS. FOR EACH META-DATASET IT IS PRESENTED THE NUMBER OF META-EXAMPLES (N), THE NUMBER OF META-FEATURES (F), THE NUMBER OF META-CLASSES (C) AND WHETHER THERE ARE MISSING VALUES (NAS).

Nro	Name	N	F	C	NAs
1	ASP-POTASSCO	1294	140	11	True
2	BNSL-2016	1179	94	8	-
3	CPMP-2015	527	23	4	-
4	CSP-2010	2024	68	2	True
5	CSP-MZN-2013	4636	118	10	-
6	GRAPHS-2015	5723	37	6	-
7	MAXSAT-PMS-2016	596	45	12	-
8	MAXSAT-WPMS-2016	630	54	10	-
9	MAXSAT12-PMS	876	32	6	-
10	MAXSAT15-PMS-INDU	601	58	16	-
11	MIP-2016	214	121	3	-
12	PROTEUS-2014	4021	33	22	True
13	QBF-2011	1368	47	5	-
14	QBF-2014	1248	47	13	True
15	SAT03-16_INDU	2000	140	10	True

Data were retrieved from the Aslib repository using the *aslib* R package[1], while the classification meta-datasets were created using the *llama* R package[2]. Preprocessing and learning tasks were performed by the *mlr*[3].

B. Dimensionality reduction techniques

Three DR techniques were considered for the experiments:

- Linear Discriminant Analysis (LDA) [11], also known as discriminant function analysis, is a statistical technique to find a linear combination of features that characterizes or separates two or more classes of objects;
- Principal Component Analysis (PCA) [10] is a technique that uses an orthogonal transformation to convert a set of possible features into linearly uncorrelated features, called principal components;
- t-Distributed Stochastic Neighbor Embedding (tSNE) [12], is a non-linear technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions. It models each high-dimensional object in such a way that similar objects are modeled by nearby points while dissimilar objects are modeled by distant points.

Different values of the variance explained by the principal components of PCA were considered in the experiments. For instance, PCA.90 means that the meta-dataset contains the principal components that explain 90% of the data variance. Similarly, the tSNE was evaluated considering different reduction rates, e.g., TSNE.30 represents tSNE computed in a meta-dataset, reducing the number of meta-features to 30% of the original number of meta-features.

C. Algorithms and hyperparameter tuning

To investigate the effect of the DR techniques four classification algorithms were used as meta-learners: k-Nearest Neighborss (kNNs), Support Vector Machines (SVMs), Random Forest (RF) and Classification and Regression Tree (CART) (with the '*rpart*' implementation). Except for the experiments with kNN, all the algorithms were also used in the original Aslib study [2], which did not investigate DR techniques.

[1] https://CRAN.R-project.org/package=aslib
[2] https://CRAN.R-project.org/package=llama
[3] https://cran.r-project.org/web/packages/mlr/index.html

The meta-learners' hyperparameters were tuned by an Random Search (RS) [25] technique. Tuning was performed following a stratified nested Cross-validations (CVs) [26] resampling method: 3 inner folds are used to estimate the validation predictive performance, while 10 outer folds assess the test performance. The budget for tuning was set to 250 steps per outer repetition. The tuning task was also executed 10 times using different seeds. Validation and test performances were assessed using the Balanced per class Accuracy (BAC) measure [27], since data collection contains binary and multiclass classification problems and some of them were unbalanced. The hyperparameter spaces of the meta-learners are described in Table II.

IV. Results and Discussions

The main experimental results evaluating the DR techniques for algorithm recommendation are described next.

A. *Overall comparison*

An overview of the main results can be given by the Critical Difference (CD) diagram in Figure 2. The diagram compares the BAC values obtained by the DR techniques for all meta-datasets and meta-learners according to the Friedman-Nemenyi test ($\alpha = 0.05$). According to the figure, the best results were achieved using the original meta-datasets, PCA.95 and LDA, and there were no significant differences among them. The tSNE setups presented the worst BAC values and were not able to improve the predictive performance of the algorithms regarding the linear techniques. The best tSNE variant, namely tSNE.10, was the unique setup with comparable results to PCA (considering all % of variance) and LDA. Since we intend to investigate the behavior of different DR techniques, hereafter we limited our analyzes to the datasets preprocessed by PCA.95, LDA, tSNE.10 and the original dataset.

B. *Predictive performance improvement*

In addition to the overall comparison, Figure 3 shows the average improvement of the predictive performance obtained by the meta-learners (rows) for each meta-dataset (columns) preprocessed by the DR techniques. The values in each cell are the performance differences considering the best variants of the DR techniques and the original meta-dataset. Bold numbers indicate significant differences according to the Wilcoxon paired-test with $\alpha - 0.05$. The white cells denote cases where the best results were obtained from the original meta-datasets. In the other cases, different colors indicate which DR technique outperformed all the others and the original meta-dataset.

In general, most of the improvements were very small, except in the "MIP-2016" meta-dataset. This meta-dataset has three classes and most of its meta-features are linearly correlated. A possible reason for these results is that meta-features' relations and patterns become more evident after the preprocessing performed by the DR techniques, resulting in the predictive performance improvements.

Among the four algorithms selected as meta-learners, kNN and SVM were more sensitive to the use of DR techniques: positive improvements were obtained in 12 of the 15 meta-datasets when using reduced meta-datasets. On the other hand, CART and RF were not often affected by the techniques. This might be due to the fact that these algorithms have an embedded feature selection mechanism during the model induction. This embedded DR process may be harmed by the use of "external" DR techniques.

Comparing the DR techniques, LDA outperformed PCA and tSNE in 18/60 (30%) and 11 of them with statistical significance. PCA was the best in 13/60 (21.67%) of the cases, 5 of them with statistical significance and are usually associated with improvement of the kNN meta-learner. Finally, tSNE was the best DR technique in 9/60 (15%), 4 of them statistically significant, all of them for the SVM meta-learner. In the remaining 20/60 (33.33%) cases, all DR resulted in the induction of meta-models with statistically lower predictive performance.

In short, the experimental results suggest that the use of DR techniques at the meta-level benefit algorithms without an embedded feature selection mechanism. Despite being a simple technique, LDA can capture the linearity between features and improve models independently of the algorithm used as meta-learner. In addition, due to its high computational cost and low effect in the predictive performance, tSNE should not be used.

C. *Dimensionality Reduction*

On average, 50.9% of the meta-features were removed in the preprocessing step because of high correlation ($corr$) (i.e., $corr \geq |0.95|$). Thus, more than half of the information on these meta-datasets were redundant. After removing the correlated attributes, the DR techniques were applied in each meta-dataset. Figure 4 shows the average meta-feature reduction percentage for the different setups of the DR techniques. The horizontal line over each bar represents the standard deviation for each technique.

tSNE.10 was the DR technique that achieved the highest feature space reduction (around 90%). However, in general, it usually reduced the predictive performance of the meta-learners, when compared with the original meta-datasets (Sec. IV-B). LDA reduced the feature space by 75.0%, but with a large standard deviation. This occurred because LDA was set to reduce the number of meta-features to $C - 1$, where C is the number of classes of the problem. PCA reduced around 60%, 54%, 46% and 33% of the feature space, for 80%, 85%, 90% and 95% of the variability explained, respectively.

Regarding the computational cost, Figure 5 shows the boxplots of the average time spent by each DR technique to preprocess datasets. The y-axis shows the runtime in milliseconds (log scale). Figure shows that tSNE was computationally much more expensive than LDA and PCA, even tSNE providing the highest reduction in data dimension. This cost can be prohibitive for many applications, since it took almost one hour, in the worst case, for each meta-dataset. On the other

TABLE II
ALGORITHMS USED AS META-LEARNERS AND THEIR HYPERSPACES EXPLORED IN EXPERIMENTS. THE NOMENCLATURE FOLLOWS THEIR RESPECTIVE R PACKAGES.

Algorithm	Symbol	Hyperparameter	Range	Type	Default	Package
CART	cp	complexity parameter	$(0.0001, 0.1)$	real	0.01	rpart
	minsplit	minimum number of instances in a node for a split to be attempted	$[1, 50]$	integer	20	
	minbucket	minimum number of instances in a leaf	$[1, 50]$	integer	7	
	maxdepth	maximum depth of any node of the final tree	$[1, 30]$	integer	30	
SVM	C	regularized constant	$[2^{-15}, 2^{15}]$	real	1	e1071
	γ	width of the Gaussian kernel	$[2^{-15}, 2^{15}]$	real	$^1/_N$	
RF	ntree	number of trees	$[2^0, 2^{10}]$	integer	500	randomForest
	nodesize	minimum node size of the decision trees	$\{1, 20\}$	integer	1	
KNN	k	number of nearest neighbors	$\{1, 50\}$	integer	7	kknn

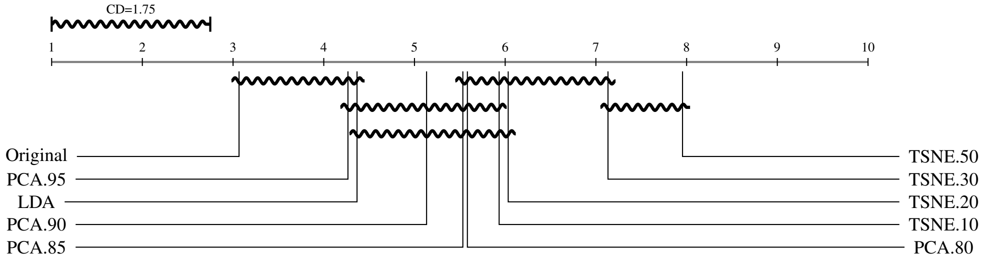
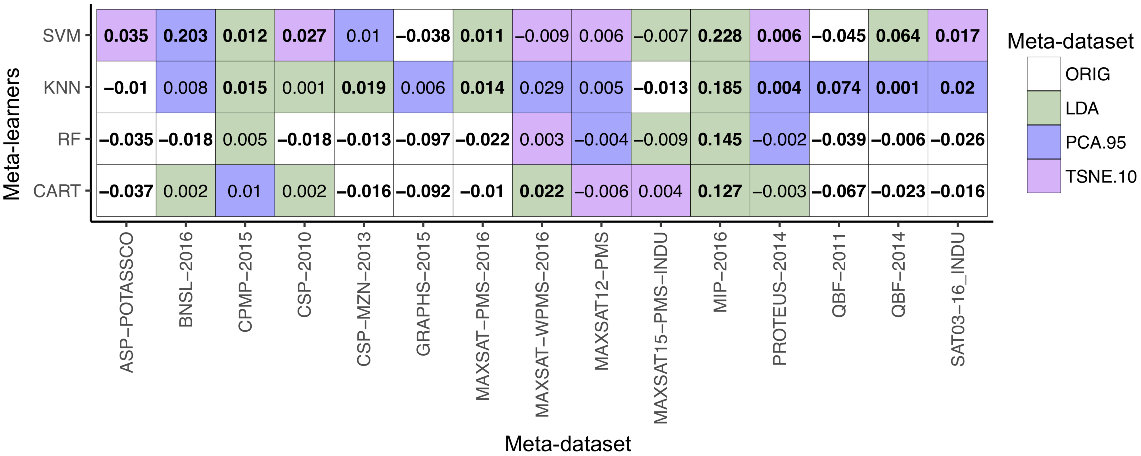


Fig. 2. Critical Difference diagram generated with the different configurations of DR techniques and the original meta-dataset. We set $\alpha = 0.05$ for all experiments.



Meta-learners	ASP–POTASSCO	BNSL–2016	CPMP–2015	CSP–2010	CSP–MZN–2013	GRAPHS–2015	MAXSAT–PMS–2016	MAXSAT–WPMS–2016	MAXSAT12–PMS	MAXSAT15–PMS–INDU	MIP–2016	PROTEUS–2014	QBF–2011	QBF–2014	SAT03–16_INDU
SVM	**0.035**	**0.203**	**0.012**	**0.027**	0.01	**–0.038**	**0.011**	–0.009	0.006	–0.007	**0.228**	**0.006**	**–0.045**	**0.064**	**0.017**
KNN	**–0.01**	0.008	**0.015**	0.001	**0.019**	0.006	**0.014**	0.029	0.005	**–0.013**	**0.185**	**0.004**	**0.074**	**0.001**	**0.02**
RF	**–0.035**	**–0.018**	0.005	**–0.018**	**–0.013**	**–0.097**	**–0.022**	0.003	–0.004	–0.009	**0.145**	–0.002	**–0.039**	**–0.006**	**–0.026**
CART	**–0.037**	0.002	0.01	0.002	**–0.016**	**–0.092**	**–0.01**	**0.022**	–0.006	0.004	**0.127**	–0.003	**–0.067**	**–0.023**	**–0.016**

Fig. 3. Balanced per class accuracy improvement obtained in each meta-dataset exploring data dimensionality reduction techniques. Bold numbers indicate situations where statistical differences were obtained by a paired Wilcoxon test with $\alpha = 0.05$.

hand, PCA was the fastest technique, running in less than one second. Finally, LDA was slightly more expensive than PCA, but also run in less than one second.

V. CONCLUSIONS

This paper investigated the use of data DR techniques for algorithm selection problems. Experiments were carried out with 15 algorithm recommendation meta-datasets from the Aslib library, 4 meta-learners, and 3 different DR techniques following different approaches. It is worth to note that these meta-datasets have many correlated attributes (on average 50.9%), thus some preprocessing procedures were necessary to attenuate the "curse of dimensionality" and other undesired properties of high-dimensional spaces.

The results showed that linear techniques, namely PCA and LDA, can be used in algorithm recommendation problems to reduce the number of meta-features, with low or none performance loss (mainly for SVM and kNN) and are computationally inexpensive. On the other hand, the non-linear

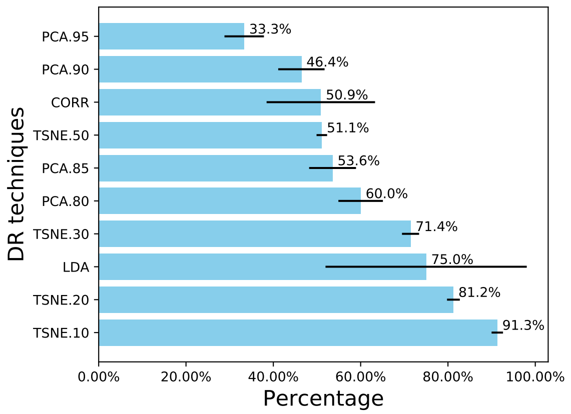


Fig. 4. Average percentage meta-feature reduction considering all 15 meta-datasets.

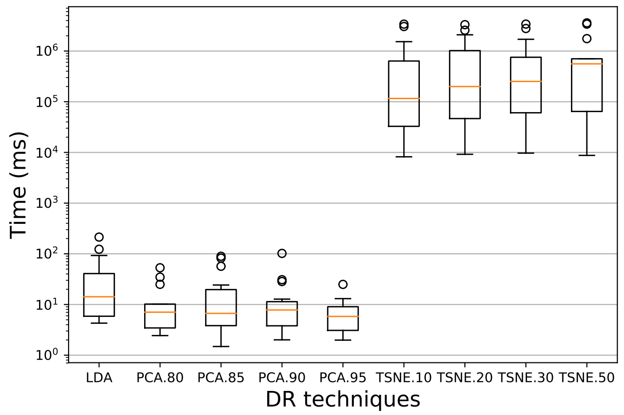


Fig. 5. Average time spent to apply the DR in all meta-examples of each meta-dataset.

technique tSNE presented poor results and presented a high computational cost. Future work includes the expansion of the experimental setup adding more meta-datasets, DR techniques and the development of an automated data cleansing recommender system, exploring other preprocessing techniques.

Acknowledgments

The authors would like to thank CAPES and CNPq (Brazilian Agencies) for their financial support, specially to the grants #2012/23114-9 from São Paulo Research Foundation (FAPESP).

References

[1] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, 2nd ed. Springer Verlag, 2009.

[2] B. Bischl, P. Kerschke, L. Kotthoff, M. T. Lindauer, Y. Malitsky, A. Fréchette, H. H. Hoos, F. Hutter, K. Leyton-Brown, K. Tierney, and J. Vanschoren, "Aslib: A benchmark library for algorithm selection," *Artif. Intell.*, vol. 237, pp. 41–58, 2016.

[3] A. C. Lorena, A. I. Maciel, P. B. C. de Miranda, I. G. Costa, and R. B. C. Prudêncio, "Data complexity meta-features for regression problems," *Machine Learning*, vol. 107, no. 1, pp. 209–246, 2018.

[4] L. P. Garcia, A. C. de Carvalho, and A. C. Lorena, "Noise detection in the meta-learning level," *Neurocomputing*, vol. 176, pp. 14–25, 2016.

[5] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2944–2952.

[6] M. Reif, F. Shafait, and A. Dengel, "Meta-learning for evolutionary parameter optimization of classifiers," *Machine Learning*, vol. 87, pp. 357–380, 2012.

[7] F. Pinto, C. Soares, and J. Mendes-Moreira, "Towards automatic generation of metafeatures," in *Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part I*, 2016, pp. 215–226.

[8] H. Liu and H. Motoda, *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007.

[9] A. Kalousis and M. Hilario, "Feature selection for meta-learning," in *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, ser. PAKDD '01. London, UK: Springer-Verlag, 2001, pp. 222–233.

[10] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[11] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Newark, NJ: Wiley, 2005.

[12] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[13] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, pp. 2859–2900, 2015.

[14] J. R. Rice, "The algorithm selection problem," *Advances in Computers*, vol. 15, pp. 65–118, 1976.

[15] J. N. Van Rijn, S. M. Abdulrahman, P. Brazdil, and J. Vanschoren, "Fast algorithm selection using learning curves," in *International symposium on intelligent data analysis*. Springer, 2015, pp. 298–309.

[16] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Satzilla: Portfolio-based algorithm selection for sat," *J. Artif. Int. Res.*, vol. 32, no. 1, pp. 565–606, 2008.

[17] A. Gisbrecht and B. Hammer, "Data visualization by nonlinear dimensionality reduction," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 2, pp. 51–73, 2015.

[18] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Phil. Trans. R. Soc. A*, vol. 374, no. 2065, 2016.

[19] L. Cao, K. S. Chua, W. Chong, H. Lee, and Q. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1-2, pp. 321–336, 2003.

[20] K. Kim and J. Lee, "Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction," *Pattern Recognition*, vol. 47, no. 2, pp. 758–768, 2014.

[21] L. I. Kuncheva and W. J. Faithfull, "PCA feature extraction for change detection in multidimensional unlabeled data," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 69–80, 2014.

[22] X. Yong, D. Feng, Z. Rongchun, and M. Petrou, "Learning-based algorithm selection for image segmentation," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1059 – 1068, 2005.

[23] M. A. Muñoz, L. Villanova, D. Baatar, and K. Smith-Miles, "Instance spaces for machine learning classification," *Machine Learning*, vol. 107, no. 1, pp. 109–147, Jan 2018.

[24] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[25] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Mar. 2012.

[26] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models." *Journal of cheminformatics*, vol. 6, no. 1, pp. 10+, 2014.

[27] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*. IEEE Computer Society, 2010, pp. 3121–3124.