Automatic selection of learning bias for active sampling

Davi P. dos Santos
and André C. P. L. F. de Carvalho
Universidade de São Paulo (USP)
Instituto de Ciências Matemáticas e Computação (ICMC)
São Carlos - SP, Brazil
Email: davips@icmc.usp.br, andre@icmc.usp.br
Phone: +55 16 99190 6776

***Abstract*—The classification task, when performed by machine learning algorithms, requires previous training on labeled instances. In many applications, the data labeling process is expensive and can affect the predictive performance of classification models. A current solution has been the use of active learning, which investigates strategies for data labeling. Its main goal is to decide which instances should be labeled and added to the training set, reducing the overall labeling costs. However, the strategy normally depends on a learning algorithm, which should be chosen by a machine learning specialist - usually based on a cross-validation procedure. Consequently, there is a deadlock: without the complete training set, the algorithm that will present the best learning curve cannot be known in advance. Ideally, some type of automatic selection should be employed to solve this deadlock. This study investigates the use of meta-learning for automatic algorithm selection in active learning tasks. Experimental results show that meta-learning is able to find correspondences between algorithms and dataset features in order to help active learning to reduce the risks of incurring in unexpected labeling costs.**

keywords: machine learning; active learning; meta-learning; recommendation

I. Introduction

Machine learning encompasses several relevant topics with significant impact on everyday life. An almost omnipresent application is the automatic classification of objects. A classification system usually induces a learning model to be able to predict the class of new objects, called *instances*. The induction of such models have been successfully performed by learning algorithms. However, a common problem is that the algorithm must be, somehow, chosen by the machine learning specialist. This algorithm selection can be a difficult task, because there is no learning approach that performs well in all domains [1].

Traditionally, classification algorithms are chosen based on the specialist knowledge and according to an evaluation process that requires the application of *cross-validation* on the training set [2]. Nonetheless, when the training set is not large enough, cross-validation cannot be properly employed - reducing the confidence of the specialist to make decisions. This is specially critical in interactive learning applications, when the classifier should be able to make predictions *during* - not only *after* - learning. Thus, the learning algorithm must be defined, despite the incipience of the training set that is still under construction. This situation is common in the *active learning* setting (Section III), focus of this study.

Active learning is an optimized sampling process that search for the most relevant unlabeled instances to build a representative training set for a given application domain. Frequently, a large pool of unlabeled instances is available to be labeled with their real classes revealed by a supervisor, called *oracle*. This task usually has a cost (human effort, time and/or materials) constrained by a budget that limits the training set size, imposing the need for parsimonious use of the supervision efforts. This problem has been addressed by several active learning strategies [3], [4]. However, since active learning usually needs a classifier, the earlier mentioned choice problem remains unsolved: which algorithm to choose? This leads the specialist to a situation where she/he has to decide with few or no labeled instances. The solution proposed in the present work to deal with this difficulty is to employ a recommendation system based on *meta-learning* (Section IV) to assist the specialist.

Meta-learning aims to discover which algorithms are suitable for a certain task. Previous studies have shown that meta-learning is able to recommend adequate algorithms for new problems (Section IV) when a collection of previous datasets are available for the induction of a *meta-model*. However, *supervised* meta-learning requires the new problem to have a complete training set. Therefore, it is not possible to use it to discover the best algorithm during an active sampling process.

This study deals with the learning algorithm choice problem, proposing and experimentally investigating the use of meta-learning to recommend learning algorithms for the active learning scenario in Section V. Additionally, different meta-learners are evaluated.

This text is organized as follows. Possibly related approaches are revised in Section II. Section III describes the main aspects of active learning. Meta-learning is the subject of Section IV. The proposed method is detailed in Section V. In Section VI, the experimental methodology is explained and the results are reported. Conclusions and future work are presented in Section VII.

II. Related Work

The type of meta-learning most related to this work is the recommendation of clustering algorithms [5], [6]. Another related approach is the use of a set of unsupervised meta-features like the Validity set [7] to recommend classifiers. Since such meta-features do not depend on labels, they can also characterize datasets constrained by the active learning scenario.

Despite the apparent similarity, the scenario of this work is fundamentally different from algorithm recommendation for clustering: while the former targets algorithms that *induce classification models*, the latter targets algorithms that *group similar instances*. No algorithm recommendation for active learning has been reported yet.

Recommender systems can predict the best algorithm for a given problem. They can also predict ranking positions - sorting techniques from the most suitable to the less suitable subject. A well established algorithm for ranking prediction is called *Predictive Clustering Trees* - PCT [8]. Otherwise, if the recommendation intends only to suggest the best option, any single-target classification algorithm can be used.

In the sections III and IV, the two central areas of this study are briefly revised: active learning and meta-learning, respectively.

III. Active Learning

Active learning provides an effective way to selectively label data [3]. To reduce acquisition costs, only the most relevant data for the learning process should be labeled. This study focus on the *pool-based query*, when the learner is given the freedom to choose the most informative instance $\boldsymbol{x}^*$ among several others in a pool $\mathcal{U}$ [9]. After chosen, the instance is sent to an *oracle*, which determines its real class label y from a set Y of possible classes and adds $\langle \boldsymbol{x}^*, y \rangle$ to a growing training set $\mathcal{L}$. This is usually a costly operation.

Several successful strategies have been proposed for the pool-based setting [4]. They usually depend on a learning algorithm, but there are also agnostic strategies. Some of them do not rely on learning algorithms. This specific group of strategies is convenient because, despite their lack of a learning bias, they still allow improvements in *label complexity* [10] over traditional/passive learning. Therefore, they could be a solution to the algorithm choice problem. Nevertheless, agnostic approaches lack the prospective capability that a learning bias could provide to optimize the search for relevant instances.

Some of the active learning strategies are summarized in Table I. Although there are only 11 *base* strategies, they amount to 22, when the variants are considered - for instance: the three distance variants represented by the suffix *. The table also gives the abbreviation adopted for each strategy. In short, Rnd, the simplest strategy, is the selection of instances at random, until the budget is over. Mar and Ent are also simple, but based on the level of uncertainty of the learner: the more ambiguous the prediction, the more informative the instance might be, if queried.

TABLE I
Adopted active learning strategies. * indicates the strategy has variants: *euc* (Euclidian distance), *man* (Manhattan d.), *mah* (Mahalanobis d.), *ent* (entropy criterion) and *acc* (accuracy c.).

Random, Margin and Entropy sampling [3]	Rnd, Mar and Ent
Agnostic and Hybrid Training Utility [11]	ATU* and HTU*
Hierarchical Sampling [12]	HS
Query-By-Committee [13]	QBC
Density Weighted [14]	DW*
Expected Error Reduction [15]	EER*
Density weighted Training Utility [14]	TU*
Multiclass Specific-General hypothesis network [4]	SGmulti
Simple margin for SVM [16]	SVMsim
Balanced k-furthest-first for SVM [17]	SVMbal

Besides random sampling, there are other agnostic strategies: ATU, HS and SGmulti. The remaining strategies are gnostic. They represent distinct paradigms, each with its own sampling bias, which, in turn, is directly related to the learning bias of the algorithm adopted as the learner.

IV. Meta-Learning

The predictive performance of a classification system depends on the bias of the learning algorithm used. Many learning algorithms, with different learning biases, have been proposed and some of them are often adopted as a universal solution in a heterogeneous range of problems. However, as stated by the theorems known as *no free lunch*, there is no bias suitable to all domains [1]. Therefore, the selection of a learning algorithm should follow a judicious decision. Usually, the algorithm is chosen by a machine learning specialist. She/He relies upon her/his knowledge about the data domain and the available algorithms. The decision is made according to a performance metric that associates algorithms and datasets. *Meta-learning* has been employed in such kind of decision problems. This technique can speed up the algorithm selection and allows a less subjective decision.

Meta-learning studies the improvement of learning algorithms by experience [18]. The improvement occurs at the *meta-level*, which is situated one level above conventional learning, named *base-level*. In the base-level, the algorithm bias is fixed; while, in the meta-level, the base-level bias can be dynamically chosen. There are diverse types of meta-learning, e.g., *stacked generalization*, *characterization by model* and *direct characterization*.

Direct characterization is the most appropriate and the choice in this study. It is based on measures taken directly from the instances in a given dataset. Conversely, other approaches might employ intermediate algorithms. Probably, the first dataset characterization was done by [19], aiming to predict accuracy and processing time. It was based on the number of instances available and number of attributes. The next set of attributes was proposed in the project STATLOG [20]. Several additional meta-features were proposed, like entropy, kurtosis, asymmetry and correlation for numeric attributes. Variations of this set are proposed in later work [18], like the adoption of histograms to keep more detailed information [21]; or, like the binarization of the degree of dispersion of the target attribute

in regression tasks [22]. There is also work directed towards optimization [23], data streams [24], ranking prediction [25] and noise detection [26].

Finally, there are approaches to the recommendation of unsupervised algorithms [5], [6]. This is the closest task to the recommendation of learning algorithms for active learning, as already discussed in Section II.

V. Proposed Method

We argue that the learning algorithm can be objectively chosen, despite the fact that it will be used in the process of construction of its own training set. We propose a few steps to solve such deadlock:

1) organize a collection of already labeled datasets[1] from several domains;
2) define an active learning strategy;
3) determine the best learning algorithm for each dataset simulating the application of the strategy[2];
4) characterize all datasets extracting the meta-features values according to our proposal in Table II;
5) generate a meta-instance for each dataset:
   the characterization process provides the attributes;
   the name of the best learner represents the class;
6) train a classifier with the meta-instances generated in the previous step; and,
7) apply the classifier to the target dataset to determine the recommended learner.

An overview of the system is depicted in Figure 1.

The ranking recommendation is analogous, except for small changes in steps 3, 5 and 6, where algorithms must be ranked instead of selected only the best.

A. Meta-learner

We propose to use a random neural network ensemble (RNN) as the meta-learner for the recommender system because it can be viewed as an algorithm that require the adjustment of only a single parameter [27]. The combination of several models into an ensemble is convenient due to the smoothing of potentially overfitted models. A bagging [28] of 1000 PCT is also employed as an alternative meta-learner, since PCT are frequently used in recommender systems and the performance is reported to be equal or better than individual trees [29]. The large size of both (RNN and PCT) ensembles is an attempt to guarantee that there is no accuracy loss due to a lack of members. This does not imply in overfitting nor loss of generalization [30]. The size of the RNN ensemble is the same as PCT's. The aggregation of all 1000 predictions (of class probabilities or ranking) is done by summation after a normalization. A combination of both algorithms in an heterogeneous ensemble is another considered possibility. Indeed, any prediction algorithm could be employed, provided that the meta-learner is suitable to the desired task: classification or ranking prediction.

[1]We constrained the datasets collection by pool size: at least 50 instances.

[2]Simulation should keep the same conditions of the target problem, like initial training set size and available budget.

TABLE II
Metafeatures (MF) description.

MF	Description	Formula
$\#\mathrm{at}$	number of attributes[a]	$\lvert A\rvert$
$\#\mathrm{ex}$	pool size[a]	$\lvert\mathcal{U}\rvert$
$\#\mathrm{nc}$	number of classes[b]	$\lvert Y\rvert$
$\#\mathrm{ea}$	pool size by number of attributes[c]	$\frac{\lvert\mathcal{U}\rvert}{\lvert A\rvert}$
$\%\mathrm{no}$	proportion of nominal attributes[c]	$\frac{1}{\lvert A\rvert}\lvert\{\boldsymbol{a}\in A \mid \mathrm{isnom}(\boldsymbol{a})=1\}\rvert$
lgex	logarithm of pool size[d]	$\log\lvert\mathcal{U}\rvert$
lgea	logarithm of pool size by number of attributes[d]	$\log\frac{\lvert\mathcal{U}\rvert}{\lvert A\rvert}$
$\#\mathrm{no}_*$	number of nominal values:[c]	$\#\mathrm{no}_{\boldsymbol{a}}=\lvert\boldsymbol{a}\rvert, \boldsymbol{a}\in A$
μ_*	mean[c] (*idem*)	$\mu_j=\frac{1}{\lvert\mathcal{U}\rvert}\sum_{\boldsymbol{x}\in\mathcal{U}} x_j \;\; 1\le j\le\lvert A\rvert$
σ_*	standard deviation[b] (*idem*)	$\sigma_j=\frac{1}{\lvert\mathcal{U}\rvert}\sum_{\boldsymbol{x}\in\mathcal{U}}(x_j-\mu_j)^2$
en_*	normalized entropy[b] (*idem*)	$\mathrm{en}_j=\frac{-1}{\log\lvert\mathcal{U}\rvert}\sum_{\boldsymbol{x}\in\mathcal{U}} x_j\log x_j$
ρ_*	correlation between attributes[b] (*idem*)	$\rho_{jk}=$ $(\sigma_j^2\cdot\sigma_k^2)^{-\frac{1}{2}}\sum_{\boldsymbol{x}\in\mathcal{U}}(x_j-\mu_j)(x_k-\mu_k)$
sk_*	skewness[b] (*idem*)	$\mathrm{sk}_j=$ $\frac{n}{(n-1)(n-2)}\sum_{\boldsymbol{x}\in\mathcal{U}}\frac{(x_j-\mu_J)^3}{\sigma_j^3}$
ku_*	kurtosis[b] (*idem*)	$\mathrm{ku}_j=\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{\boldsymbol{x}\in\mathcal{U}}\frac{(x_i-\mu_j)^4}{\sigma_j^4}$ $-3(n-1)^2[(n-2)(n-3)]^{-1}$
cn_{kC}	connectivity[e] *k-means*	cluster validity measure [31]
cn_{hC}	connectivity *hierarc. clust.*	
du_{kC}	Dunn index[e] *k-means*	cluster validity measure [32]
du_{hC}	Dunn index *hierarc. clust.*	
si_{kC}	silhouette[e] *k-means*	cluster validity measure [33]
si_{hC}	silhouette *hierarc. clust.*	

[a] Characterization suggested by [19].
[b] Based on STATLOG project [20].
[c] Based on the feature set by [21].
[c] Adaptation of the summarization proposed by [21].
[d] Meta-features for clustering algorithm recommendation [5].
[e] Meta-features for classification algorithm [7] and for clustering algorithms [6] recommendation.
[obs] *The symbol * indicates that the metafeature is summarized by its maximum (max), minimum (min), mean (mea) and min/max values. The C subscript indicates the adopted number of clusters: $|Y|$, $1.5|Y|$ or $2|Y|$. The set A represents all attributes. The function* isnom *returns 1 if an attribute is nominal or 0 otherwise.*

B. Model selection

RNN has the number of neurons L to be adjusted. This is done, for each ensemble member, by choosing the value that produces the smallest leave-one-out error, which is instantly calculated from the trained network via PRESS statistic [34]. Only sigmoid additive nodes were considered.

VI. Experiments

The proposed method was empirically evaluated. Results are confirmed by the Friedman test with Nemenyi post-hoc test.

Although PCT already represents a common algorithm suitable for classification (and ranking), other metaclassifiers were also tested, like 5NN. Random Forest with 1000 trees was used as a meta-learner alternative to ease reproducibility of the experiments (details in Section VI-B). Additionally,

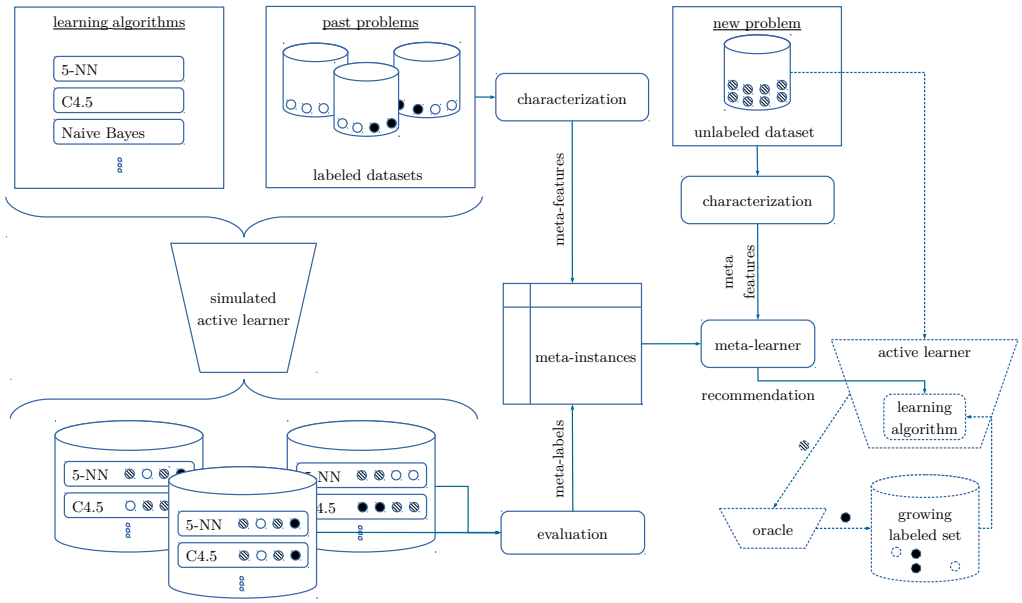

Fig. 1. Overview of the recommender system. *Dashed elements represent the active learning loop, which benefits from the proposed algorithm recommendation.*

two baselines - the proportion of the majoritary class and the mean ranking - were adopted for classification and ranking prediction, respectively.

A. Methodology

We adopted a dataset-dependent budget of ϕ queries, $\phi = \min(\frac{|\mathcal{U}|}{2}, 200)$. Five runs of 5-fold cross-validation were applied for the base learners and ten runs of 10-fold for the meta-learners [2]. Duplicate instances were removed to simulate a consistent oracle. It was assumed that the label of only one instance per class was known before the start of the active sampling process[3] [35]. The performance indicator at the base level was the *Area under the Learning Curve* (ALC) [36] with the kappa measure [37].

B. Datasets and base learners

The evaluation was performed on 94 datasets from the UCI repository [38]. The process was repeated for each one of the 22 strategies and variants (Table I) to show the general applicability of the approach. The employed classifiers as base learners were: 5NN, NB[4], SVM with RBF[5] and RFw[6] [39]–[41]. When not stated otherwise, all parameters used the default values from the Weka library [42]. SVM was of type C-SVC [43] with the following parameters: $\gamma = 0,5$, $C = 1$, cache 200MB and $eps = 0.001$. 5NN was weighted by the complement of the distance.

Datasets are detailed in Table III. They were binarized and standardized or discretized when needed, i.e., for distance calculations or training some of the algorithms (NB, SVM and 5NN). There were no missing attribute values.

[3]Except for HS strategy.
[4]*Naive Bayes*
[5]*Support Vector Machines with Radial Basis Function*
[6]Weka implementation of Random Forest.

TABLE III
DATASETS DETAILS. #EX: NUMBER OF EXAMPLES; #NC: NUMBER OF CLASSES; #AT: NUMBER OF ATTRIBUTES; #NO: NUMBER OF NOMINAL ATTRIBUTES;

Name	#ex	#nc	#at	#no	Name	#ex	#nc	#at	#no
1-abalone 3class	3342	3	8	1	48-phoneme	4316	2	5	0
2-arcene	160	2	998	0	49-pima indians d...	614	2	8	0
3-artificial char...	3890	10	7	0	50-planning relax	141	2	12	0
4-autoUn. au1 1000	798	2	20	0	51-qsar biodegrad...	842	2	41	0
5-autoUn. au7 30...	880	5	12	4	52-ringnorm	5920	2	20	0
6-autoUn. au7 700	560	3	12	4	53-robot fai. lp1	70	4	90	0
7-balance scale	500	3	4	0	54-robot fai. lp4	93	3	90	0
8-banana	4233	2	2	0	55-robot fai. lp5	130	5	90	0
9-banknote authen...	1078	2	4	0	56-robot nav s. r...	4142	4	2	0
10-bupa	273	2	6	0	57-saheart	370	2	9	1
11-car evaluation	1382	4	6	6	58-seeds	168	3	7	0
12-cardiotocograp...	1692	10	35	0	59-semeion	1274	10	256	0
13-cardiotocograp...	1692	3	35	0	60-spambase	3366	2	57	0
14-climate simula...	432	2	20	0	61-spect heart	178	2	22	22
15-connect. mines...	166	2	60	0	62-spectf heart	214	2	44	0
16-connect. vowel...	422	11	10	0	63-statlog austra...	552	2	14	6
17-connect. vowel...	792	11	13	0	64-statlog ge. cre...	800	2	24	0
18-dbworld subje...	50	2	242	242	65-statlog heart	216	2	13	0
19-first order th...	4402	6	51	0	66-statlog i. seg...	1669	7	18	0
20-flare	337	6	12	2	67-statlog land s...	2287	6	36	0
21-glass	170	6	9	0	68-statlog vehi s...	677	4	18	0
22-habermans surv...	226	2	3	0	69-steel plates f...	1553	2	33	0
23-heart di. clev...	242	5	13	2	70-systhetic cont...	480	6	60	0
24-heart di. hung...	234	2	13	0	71-teaching assis...	85	3	5	2
25-hepatitis	124	2	19	13	72-thyroid ann	2967	3	21	0
26-hill valley wi...	970	2	100	0	73-thyroid hypoth...	2468	2	25	18
27-horse colic su...	240	2	27	14	74-thyroid newthy...	172	3	5	0
28-indian liver p...	456	2	10	1	75-thyroid sick	2201	5	26	20
29-ionosphere	280	2	33	0	76-thyroid sick e...	2468	2	25	18
30-iris	118	3	4	0	77-tic tac toe	766	2	9	9
31-kr vs kp	2557	2	36	36	78-turkiye stud...	2667	13	32	0
32-leaf	272	30	15	0	79-twonorm	5920	2	20	0
33-leukemia hasli...	80	2	50	0	80-user knowledge	322	5	5	0
34-mammographic...	514	2	5	0	81-vertebra c. 2c...	248	2	6	0
35-mfeat fourier	1595	10	76	0	82-vertebra c. 3c...	248	3	6	0
36-molecular prom...	85	2	57	57	83-volcanoes a3	1217	5	3	0
37-molecular spli...	2404	3	60	60	84-volcanoes b2	8530	5	3	0
38-monks1	346	2	6	0	85-volcanoes d1	7002	5	3	0
39-monks2	346	2	6	6	86-volcanoes e2	863	5	3	0
40-monks3	346	2	6	0	87-voting	223	2	16	16
41-movement libras	264	15	90	0	88-waveform v2	4000	3	40	0
42-movement l. 10	192	15	90	0	89-wdbc	455	2	30	0
43-mushroom	6499	2	21	21	90-wholesale chan...	352	2	7	0
44-ozone eighthr	2021	2	72	0	91-wilt	3855	2	5	0
45-ozone onehr	2022	2	72	0	92-wine	142	3	13	0
46-page blocks	4314	5	10	0	93-wine quality w...	3149	5	11	0
47-parkinsons	156	2	22	0	94-yeast 4class	1015	4	8	0

C. Experimental results

The experiments are divided in *class prediction* and *ranking prediction*.

D. Class prediction

The four ensembles (RNN, PCT, RFw and RNN+PCT) and the two non-ensemble classifiers (5NN and Maj - majoritary class) are compared in Table IV by balanced accuracy [44]. The Maj column can be considered the baseline, since it is analogous to a random classifier. All ensembles outperformed, often by a large margin, the baseline for all strategies. This means that using the recommender system is considerably better than the best guessing, i.e, it is better than just defaulting to the most frequent winner.

The other non-ensemble classifier, 5NN, although better than Maj, was the worst algorithm for almost all strategies. This suggests that the recommendation can be affected depending on the type of the metaclassifier.

The statistical significance of the advantage of the recommender system is detailed in Table V. RNN and PCT ensembles were better than the baseline and 5NN with a p-value of 0.01. RFw and RNN+PCT outperformed Maj within the same confidence. 5NN was better than Maj as well, but with a p-value of 0.10. Finally, RNN achieved the highest number of victories, but not enough to reach statistical significance when compared to the other ensembles.

These findings, although limited to the 94 datasets employed, are enough to argue that active learning is likely to benefit from meta-learning.

TABLE IV
BALANCED ACCURACY (%)/STANDARD DEVIATION. *Highest values are in bold face. Each mean value is calculated along all runs of cross-validation.*

	RNN	PCT	RFw	RNN+PCT	5NN	Maj
Mar	**49**/17	40/15	37/15	41/13	35/8	34/9
Ent	**53**/15	49/12	43/16	43/14	35/8	33/6
TUman	56/15	56/17	55/15	**57**/15	43/18	30/4
TUmah	**55**/23	47/17	54/24	53/23	40/11	30/4
TUeuc	**50**/19	49/16	44/12	49/15	47/20	28/4
SGmulti	**34**/16	31/14	31/9	30/10	30/8	28/4
Rnd	**40**/22	34/12	31/9	29/8	24/9	27/3
EERent	34/16	**40**/14	33/14	35/13	34/15	27/3
HS	**45**/22	39/11	34/10	34/11	37/11	27/3
DWmah	**48**/17	47/10	44/16	47/19	41/13	26/3
DWeuc	**58**/18	52/16	52/12	49/16	52/12	26/3
ATUeuc	42/16	40/10	**44**/16	42/13	34/12	26/3
SVMsim	30/14	34/14	**37**/13	36/13	30/16	25/0
SVMbal	40/13	**42**/13	36/12	36/11	31/11	25/0
QBCRFw	39/16	**47**/13	42/12	43/11	38/13	25/0
HTUman	45/15	**50**/14	49/13	47/14	40/13	25/0
HTUmah	46/16	49/15	**51**/11	49/12	29/12	25/0
HTUeuc	46/14	**53**/17	45/10	47/11	37/9	25/0
EERacc	**33**/13	27/7	**33**/11	28/9	31/16	25/0
DWman	45/15	**47**/16	38/14	37/13	41/14	25/0
ATUman	33/13	**39**/15	38/12	34/9	33/17	25/0
ATUmah	36/14	**43**/11	**43**/14	42/8	29/14	25/0

E. Ranking prediction

In the case of ranking prediction, the suitable algorithms were RNN, PCT and RNN+PCT ensembles. The mean ranking (called Def - default) was the baseline. The comparison, given in Table VI, is based on Spearman's correlation between predicted and expected rankings; higher values indicate more accurate predictions.

TABLE V
ONE VERSUS ONE BY BALANCED ACCURACY. *Confidence levels according to Friedman test with Nemenyi post-hoc test. Each symbol indicates a p-value:* * (0.01) + (0.05) . (0.10).

	1	2	3	4	5	6
1 - RNN	-				*	*
2 - PCT		-			*	*
3 - RFw			-		.	*
4 - RNN+PCT				-		*
5 - 5NN					-	.
6 - Maj						-

Differently from the prediction of classes, in ranking prediction the RNN ensemble dominated the lowest performance values, whereas RNN+PCT presented almost all highest values. In fact, according to Table VII, RNN+PCT outperformed RNN with high significance (p-value of 0.01). Nevertheless, all ensembles were better than Def with high confidence (p-values of 0.01; and 0.05 for RNN).

TABLE VI
SPEARMAN'S CORRELATION COEFFICIENT/STANDARD DEVIATION. *Highest values are in bold face.*

	RNN	PCT	RNN+PCT	Def
Ent	0.525/0.125	**0.568**/0.074	0.566/0.097	0.528/0.099
Mar	0.525/0.184	0.587/0.156	**0.588**/0.182	0.516/0.130
SGmulti	0.532/0.130	**0.570**/0.107	0.567/0.118	0.494/0.094
EERacc	0.567/0.179	0.609/0.140	**0.611**/0.179	0.460/0.151
Rnd	0.439/0.144	0.453/0.118	**0.467**/0.165	0.430/0.117
HS	0.487/0.174	0.508/0.109	**0.523**/0.140	0.418/0.136
EERent	0.454/0.107	0.444/0.147	**0.475**/0.144	0.364/0.095
TUman	0.466/0.191	0.444/0.185	**0.499**/0.200	0.324/0.133
TUmah	0.502/0.143	0.476/0.139	**0.520**/0.118	0.309/0.119
QBCRFw	0.400/0.150	0.409/0.138	**0.455**/0.169	0.306/0.061
TUeuc	0.484/0.121	0.484/0.172	**0.507**/0.165	0.301/0.182
DWmah	0.402/0.114	**0.457**/0.144	0.455/0.146	0.277/0.101
HTUman	0.392/0.199	0.419/0.200	**0.449**/0.209	0.272/0.121
ATUman	0.316/0.145	**0.378**/0.112	0.351/0.087	0.263/0.155
ATUeuc	0.308/0.110	**0.377**/0.102	0.346/0.111	0.261/0.135
ATUmah	0.337/0.163	0.344/0.160	**0.369**/0.170	0.246/0.151
HTUeuc	0.450/0.116	0.420/0.144	**0.484**/0.113	0.245/0.135
HTUmah	0.396/0.198	0.413/0.197	**0.446**/0.194	0.224/0.212
SVMbal	0.316/0.192	**0.414**/0.183	0.411/0.148	0.203/0.134
DWman	0.441/0.113	0.451/0.121	**0.476**/0.102	0.201/0.113
DWeuc	0.430/0.141	0.446/0.111	**0.460**/0.118	0.190/0.083
SVMsim	0.273/0.181	**0.390**/0.153	0.356/0.159	0.177/0.133

TABLE VII
ONE VERSUS ONE BY SPEARMAN'S CORRELATION. *Details in Table V.*

	1	2	3	4
1 - RNN	-			+
2 - PCT	.	-		*
3 - RNN+PCT	*		-	*
4 - Def				-

VII. CONCLUSION

In this paper, we addressed the algorithm choice problem in active learning. There are several strategies, usually based on model uncertainty measures. However the model induction is only possible after the definition of the learning algorithm. Strategies rely on the machine learning specialist decision

about which is the proper learning algorithm to generate the model.

We proposed to adopt unsupervised meta-learning to avoid the subjectiveness of the specialist decision and the scarceness of training data. Experimentally, we showed that this is not only possible, but it can also be done with a performance superior to important baselines for the collection of 94 datasets employed.

Recommendation of strategies, instead of learners, is intended as future work.

VIII. Acknowledgments

The authors would like to thank CNPq (grant number 150239/2016-5), FAPESP (grant number 2013/07375-0 - CEPID CeMEAI) and CAPES (grant number DS-3136101/D) for the financial support.

References

[1] C. Schaffer, "A conservation law for generalization performance," in *ICML*. Morgan Kaufmann, 1994, pp. 259–265.

[2] R. R. Bouckaert and E. Frank, "Evaluating the replicability of significance tests for comparing learning algorithms," in *PAKDD*, ser. Lecture Notes in Computer Science, vol. 3056. Springer, 2004, pp. 3–12.

[3] B. Settles, *Active Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2012.

[4] D. P. Santos and A. C. P. L. F. Carvalho, "Comparison of active learning strategies and proposal of a multiclass hypothesis space search," in *HAIS*, ser. Lecture Notes in Computer Science, vol. 8480. Springer, 2014, pp. 618–629.

[5] M. C. P. Souto, R. B. C. Prudêncio, R. G. F. Soares, D. S. A. de Araujo, I. G. Costa, T. B. Ludermir, and A. Schliep, "Ranking and selecting clustering algorithms using a meta-learning approach," in *IJCNN*. IEEE, 2008, pp. 3729–3735.

[6] D. G. Ferrari and L. N. de Castro, "Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods," *Information Sciences*, vol. 301, no. 0, pp. 181 – 194, 2015.

[7] B. F. d. Souza, "Meta-aprendizagem aplicada à classificação de dados de expressão gênica," Ph.D. dissertation, Universidade de São Paulo, 2010.

[8] L. Todorovski, H. Blockeel, and S. Dzeroski, "Ranking with predictive clustering trees," in *ECML*, ser. Lecture Notes in Computer Science, vol. 2430. Springer, 2002, pp. 444–455.

[9] D. D. Lewis, "A sequential algorithm for training text classifiers: Corrigendum and additional data," *SIGIR Forum*, vol. 29, no. 2, pp. 13–19, 1995.

[10] A. Beygelzimer, D. Hsu, J. Langford, and T. Z. 0001, "Agnostic active learning without constraints," *CoRR*, vol. abs/1006.2588, 2010.

[11] D. Santos and A. d Carvalho, "Selectively inhibiting learning bias for active sampling," in *2015 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2015, pp. 62–67.

[12] S. Dasgupta, "Two faces of active learning," *Theoretical Computer Science*, vol. 412, no. 19, pp. 1767–1781, 2011.

[13] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging," in *ICML*. Morgan Kaufmann, 1998, pp. 1–9.

[14] B. Settles, "Curious machines: active learning with structured instances," Ph.D. dissertation, University of Madison Wisconsin, 2008.

[15] Y. Guo and R. Greiner, "Optimistic active-learning using mutual information," in *IJCAI*, 2007, pp. 823–829.

[16] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[17] T. T. Osugi, K. Deng, and S. D. Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," in *ICDM*. IEEE Computer Society, 2005, pp. 330–337.

[18] P. Brazdil, C. G. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning - Applications to Data Mining*, ser. Cognitive Technologies. Springer, 2009.

[19] L. A. Rendell, R. Sheshu, and D. K. Tcheng, "Layered concept-learning and dynamically variable bias management," in *IJCAI*. Morgan Kaufmann, 1987, pp. 308–314.

[20] P. Brazdil and R. Henery, "Analysis of results," *Machine learning, neural and statistical classification*, pp. 175–212, 1994.

[21] A. Kalousis, "Algorithm selection via meta-learning," Ph.D. dissertation, Universite de Geneve, 2002.

[22] T. A. F. Gomes, R. B. C. Prudêncio, C. Soares, A. L. D. Rossi, and A. C. P. L. F. Carvalho, "Combining meta-learning and search techniques to select parameters for support vector machines," *Neurocomputing*, vol. 75, no. 1, pp. 3–13, 2012.

[23] J. Kanda, A. C. P. L. F. de Carvalho, E. R. Hruschka, and C. Soares, "Selection of algorithms to solve traveling salesman problems using meta-learning," *Int. J. Hybrid Intell. Syst*, vol. 8, no. 3, pp. 117–128, 2011.

[24] A. L. D. Rossi, A. C. P. de Leon Ferreira de Carvalho, C. Soares, and B. F. de Souza, "Metastream: A meta-learning based method for periodic algorithm selection in time-changing data," *Neurocomputing*, vol. 127, pp. 52–64, 2014.

[25] B. F. de Souza, A. C. P. L. F. de Carvalho, and C. Soares, "Empirical evaluation of ranking prediction methods for gene expression data classification," in *IBERAMIA*, ser. Lecture Notes in Computer Science, vol. 6433. Springer, 2010, pp. 194–203.

[26] L. P. F. Garcia, A. C. P. F. de Carvalho, and A. C. Lorena, "Noise detection in the meta-learning level." *Neurocomputing*, 2015, in press 2015.

[27] W. F. Schmidt, M. Kraaijveld, R. P. Duin *et al.*, "Feedforward neural networks with random weights," in *Proceedings of the 11th International Conference on Pattern Recognition Methodology and Systems*, vol. 2. IEEE, 1992, pp. 1–4.

[28] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[29] D. Kocev, C. Vens, J. Struyf, and S. Dzeroski, "Ensembles of multi-objective decision trees," in *ECML*, ser. Lecture Notes in Computer Science, vol. 4701. Springer, 2007, pp. 624–631.

[30] G. Seni and J. F. E. IV, *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*, ser. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2010.

[31] R. Xu and D. Wunsch, *Clustering*. John Wiley & Sons, 2008, vol. 10.

[32] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[33] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[34] R. H. Myers, *Classical and Modern Regression with Applications*, 2nd ed., ser. The Duxbury Advanced Series in Statistics and Decision Sciences. Boston: PWS–KENT, 1990.

[35] Y. Guo and D. Schuurmans, "Discriminative batch mode active learning," in *NIPS*. Curran Associates, Inc, 2007.

[36] I. Guyon, G. C. Cawley, G. Dror, and V. Lemaire, "Results of the active learning challenge," in *Active Learning and Experimental Design @ AISTATS*, vol. 16. JMLR.org, 2011, pp. 19–45.

[37] B. D. Eugenio and M. Glass, "The kappa statistic: A second look," *Computational Linguistics*, vol. 30, no. 1, pp. 95–101, 2004.

[38] K. Bache and M. Lichman, "UCI repository of machine learning databases," University of California, Department of Information and Computer Science, Irvine, CA, Machine-readable data repository, 2013.

[39] P. E. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[40] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *ECML*, ser. Lecture Notes in Computer Science, vol. 1398. Springer, 1998, pp. 4–15.

[41] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

[42] M. A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[43] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[44] M. Masso and I. I. Vaisman, "Accurate and efficient gp120 V3 loop structure based models for the determination of HIV-1 co-receptor usage," *BMC Bioinformatics*, vol. 11, p. 494, 2010.