Learning Heterogeneous Similarity Measures for Hybrid-Recommendations in Meta-Mining

Phong Nguyen, Jun Wang, Melanie Hilario, Alexandros Kalousis
University Of Geneva
University of Applied Sciences, Western Switzerland

***Abstract*—The notion of meta-mining has appeared recently and extends the traditional meta-learning in two ways. First it does not learn meta-models that provide support only for the learning algorithm selection task but ones that support the whole data-mining process. In addition it abandons the so called black-box approach to algorithm description followed in meta-learning. Now in addition to the datasets, algorithms also have descriptors, workflows as well. For the latter two these descriptions are semantic, describing properties of the algorithms, such as cost functions, learning biases, etc. With the availability of descriptors both for the datasets and the data-mining workflows the traditional modelling techniques followed in meta-learning, typically based on classification and regression algorithms, are no longer appropriate. Instead we are faced with a problem the nature of which is much more similar to the problems that appear in recommendation systems. However on the same time the requirements of the meta-mining tasks make the direct use of tools from recommender systems rather inappropriate. The most important meta-mining requirements are that suggestions should use only the datasets and workflows descriptors and the cold-start problem, e.g. providing workflow suggestions for new datasets.**

In this paper we take a different view on the meta-mining modelling problem and treat it as a recommender problem. In order to account for the meta-mining specificities we derive a novel metric-based-learning recommender approach. Our method learns two homogeneous metrics, one in the dataset and one in the workflow space, and a heterogeneous one in the dataset-workflow space. All learned metrics reflect similarities established from the dataset-workflow preference matrix. The latter is constructed from the performance results obtained by the application of workflows to datasets. We demonstrate our method on meta-mining over biological (microarray datasets) problems. The application of our method is not limited to the meta-mining problem, its formulations is general enough so that it can be applied on problems with similar requirements.

***Keywords*-Semantic Meta-Mining; Meta-Learning;**

I. Introduction

Meta-learning is learning to learn: in computer science, it is the application of machine learning techniques to meta-data describing past learning experience, typically applications of learning algorithms to specific datasets, in order to derive meta-learning models that can support the selection of an appropriate algorithm for a new dataset, [1], [2], [3], [4]. The meta-learning models are usually classification or regression models learned by standard classification and regression algorithms. Until very recently meta-learning was focusing only on the learning part of the data mining process, by trying to model the behavior of different learning algorithms, and was treating the learning algorithms as black-boxes making no effort to describe the concepts that underline them and their properties.

The authors of [5] made an effort to address these limitations by extending the meta-learning process to the whole data mining process resulting in a more comprehensive task which they called *meta-mining*. In addition they made use of a data mining ontology in order to provide detailed descriptions of data mining algorithms in terms of their core components, underlying assumptions, cost functions, optimization strategies, etc, as well as detailed descriptions of data mining workflows, the latter composed of operators implementing data mining algorithms. Even though the introduction of data mining algorithm and workflow descriptors was an important step the authors made rather poor use of them by modelling the meta-mining problem as a classification problem, following thus the traditional meta-learning modelling approach. In this classification problem the meta-mining instances corresponded to data mining experiments, applications of workflows or algorithms on datasets, and they consisted of two types of features, features that described the dataset and features that describe the data mining workflow. The class label was determined on the basis of the performance result estimated by the application of the workflow on the dataset and was indicating the appropriateness or not of the workflow for the dataset.

In this paper we take a different approach on the modelling of the meta-learning and meta-mining tasks. We view them as a matching problem between datasets on the one hand and data mining algorithms or workflows on the other, in which the matching criterion is the performance of the latter when applied on the former. We will address three different meta-mining tasks. Given a new dataset we want to recommend or rank available algorithms or data mining workflows in terms of their expected performance on the specific dataset; we will call this task learning workflow preferences. Symmetrically to this we want, given a new data mining workflow or algorithm to know for which datasets they are most appropriate; we will call this task learning dataset preferences. Finally, given a new dataset and a new workflow or algorithm we want to be able to determine the goodness of their match, i.e. the degree to which the latter will have a good performance when applied to the former; we will call this learning dataset-workflow preferences. It is obvious that all these should be determined without any actual application of the new algorithms on the new datasets but on the basis of some meta-mining model that will be learned from the past mining experience.

These type of problems are similar in nature to problems that appear in recommender systems, where we have users and items and we want to suggest additional items for a given user based on the preferences of users with similar preferences. In the meta-mining and meta-learning case the matrix containing the preferences of users for items is replaced by a performance based matrix of datasets and workflows or algorithms that indicates the performance of the latter applied to the former. This performance-based preference matrix will be one component of our meta-mining data; in addition we will use dataset and workflow or algorithm descriptors. The final meta-mining models will only use the dataset and workflow descriptors to return the preferences. In recommender systems there is a relevant stream of work that makes use of descriptors of users and items, similar to the descriptors of datasets and workflows, that is called hybrid recommendation systems [6], [7], [8], [9]. However there are also a number of differences between the nature of the recommendation problem that we have in meta-mining and the typical recommendation problems. In the latter the preferences matrix is often very sparse, of high dimensionality, and can have hundreds of thousands of rows/users. In contrast, the preferences matrix in meta-mining is rather dense and involves few hundreds of datasets and workflows. The features of datasets and workflows that we use in the meta mining problems are quite informative in contrast to the typical recommendation problems where it is rather hard to get informative features especially in what concerns user descriptions. Finally, in the meta mining setting, the cold-start problem is central, with the most typical example being predicting the workflow preferences for a new dataset. However, in recommendation problem, partly due to the low information content of the features describing users, but also due to the nature of the problem itself, the main focus is in the completion of missing values in the preferences matrix based on historical ratings of items by similar users.

In this paper we present a new metric-learning-based approach to hybrid recommendation for meta-mining, which learns to match dataset descriptors to workflow descriptors. More specifically we will learn three different metrics. One on the dataset descriptor space which will reflect the fact that similar datasets will have similar workflow preferences, as these are given by the performance-based preference matrix. One on the workflow descriptor space which will reflect the fact that similar workflows will have similar dataset preferences again as these are given by the performance-based preference matrix. And a last heterogeneous metric over the two spaces of dataset and workflow descriptors, which will directly give the similarity/appropriateness of a given dataset for a given workflow. We will use these learned metrics, alone or in combination, to address the three meta-mining tasks that we described in the previous paragraphs.

To the best of our knowledge the metric learning approach that we present is the first of its kind, not only for meta-mining, but also for the general context of hybrid recommendation problems. Even though it was developed to address the specific requirements of the meta-mining setting it is not specific to it, and it can be used in any kind of recommendation system that has similar requirements, i.e. preference based matchings of users to items based on descriptions of them and cold-start problem.

The rest of the paper is organized as follows. In section II, we define the meta-mining tasks. In section III, we describe our metric-learning based approach to the problem of learning hybrid recommendations for meta-mining. In section IV, we present briefly the characteristics—features—that we use to describe the datasets and the workflows. In section V, we give the experiments and the evaluation of our approach. In section VI, we discuss the related work and finally we conclude in section VII.

II. Meta-Mining Tasks

Before proceeding to the definition of the different meta-mining tasks that will address let us give some necessary notations. Let $\mathbf{x} = (x_1, \dots x_d)^T \in \mathbb{R}^d$ be the description of some dataset, and $\mathbf{X}$ an $n \times d$ dataset matrix the ith row of which is given by the $\mathbf{x}_i^T$ dataset. Thus the $\mathbf{X}$ matrix is the set of datasets over which the meta-mining will take place. In addition let $\mathbf{a} = (a_1, \dots, a_l)^T \in \mathbb{R}^l$ be the description of some data mining workflow, and $\mathbf{A}$ an $m \times l$ workflow matrix the jth row of which is the $\mathbf{a}_j^T$ workflow, i.e. $\mathbf{A}$ will be the data mining workflow matrix over which the meta-mining will take place. Finally let $\mathbf{R}$ be an $n \times m$ matrix the (i, j) entry of which depends on some performance result obtained by the application of the $\mathbf{a}_j$ data mining workflow on the $\mathbf{x}_i$ dataset. We will use the notation $\mathbf{r}_{\mathbf{x}_i}$ to denote the vector given by the row of $\mathbf{R}$ which corresponds to the $\mathbf{x}_i$ dataset and which contains the performance measures obtained by the application of the m data mining workflows on $\mathbf{x}_i$, and the notation $\mathbf{r}_{\mathbf{a}_j}$ to denote the vector given by the jth column of $\mathbf{R}$ which contains the performance results of the application of the a_j data-mining workflow on the n datasets. Thus the $\mathbf{R}$ matrix relates, based on performance, datasets with workflows and can be seen as giving the appropriateness of workflows for datasets and vice versa.

Since here we will focus only on meta-mining for classification problems the performance measure that we will be using to fill up $\mathbf{R}$ will be based on classification accuracy which we will estimate by ten-fold cross-validation. The accuracies achieved by different workflows are not comparable over different datasets, what is much more important in meta-learning and meta-mining is the relative performance order of a set of data mining workflows or algorithms on a given dataset; this relative order can be compared in a meaningful manner over different datasets. We devise such a relative order in the following way. Given a pair of classification data mining workflows $\mathbf{a}_k$ and $\mathbf{a}_l$ applied on dataset $\mathbf{x}_i$ we compute the statistical significance of their accuracies differences using a McNemar's test, with a p-value of 0.05. If one workflow is statistically significant better than the other it is assigned a score of one and the other a score of zero, in case of no significant difference both are assigned a score of 0.5. For a given dataset $\mathbf{x}_i$ the score of a workflow $\mathbf{a}_k$ will be the sum of the points it gets in all its pairwise comparisons with the other $m-1$ workflows. It is this score that we will use to populate the $\mathbf{R}$ matrix, i.e. its (i, j) entry will be the score obtained by workflow $\mathbf{a}_j$ on dataset $\mathbf{x}_i$, we will also use the notation $r_{\mathbf{x}_i,\mathbf{a}_j}$ to denote the (i, j) entry of $\mathbf{R}$.

Given the above we will now define three different meta-mining tasks. In the first one given a new unseen dataset $\mathbf{x}$, i.e. a dataset with which we have not experimented with, we want to estimate the relative performance order of the m data mining workflows. In other words we want to estimate the relative workflow performance, or workflow preference, vector $\mathbf{r_x}$ for the $\mathbf{x}$ dataset. We will call this task *learning workflow preferences*. The second meta-mining task is the symmetric of the first; here we want to estimate appropriateness of a new unseen workflow $\mathbf{a}$ for the n datasets, i.e. we want to estimate the dataset preference vector $\mathbf{r_a}$ for the $\mathbf{a}$ workflow. We will call this task *learning dataset preferences*, Finally in the third, last and most difficult, meta-mining task we want to estimate the appropriatness of an unseen workflow $\mathbf{a}$ on an unseen dataset $\mathbf{x}$, i.e. estimate the $r_{\mathbf{x},\mathbf{a}}$ value. We will call this meta-mining task *learning dataset-workflow preferences*.

To address all three tasks we will rely on the use of appropriate similarity measures. To learn workflow preferences we will need a dataset similarity measure that given a new dataset $\mathbf{x}$ will establish its most similar datasets in the training set $\mathbf{X}$. From the workflow preference vectors of these datasets we will then estimate the workflow preference vector $\mathbf{r_x}$ of $\mathbf{x}$. In the same manner to learn dataset preferences we need a workflow similarity measure that given a new workflow $\mathbf{a}$ will establish its most similar workflows in the training set $\mathbf{A}$. From the dataset preference vectors of these workflows we will then estimate the dataset preference vector $\mathbf{r_a}$ of $\mathbf{a}$. For the last task we will rely on the use of an *heterogeneous* similarity measure that computes directly similarities between workflows and datasets, which thus given an unseen dataset $\mathbf{x}$ and an unseen workflow $\mathbf{a}$ will produce the $r_{\mathbf{x},\mathbf{a}}$ corresponding to the appropriateness of $\mathbf{a}$ for $\mathbf{x}$.

In the following section we will show how to learn appropriate metric matrices that we will use to compute the similarity measures that we briefly described in the previous paragraph.

III. Learning Similarities for Hybrid-Recommendations in Meta-Mining

Before starting to describe in detail how we will address the three meta-mining tasks let us take a step back and give a more abstract picture of the type of learning setting that we want to address. We have two types of learning instances, $\mathbf{x} \in \mathcal{X}$, and $\boldsymbol{a} \in \mathcal{A}$, and two training matrices $\mathbf{X} : n \times d$ and $\mathbf{A} : m \times l$ respectively. Additionally we also have an instance alignment or preference matrix $\mathbf{R} : n \times m$, the R_{ij} entry of which gives some measure of appropriateness, preference, or match of the $\mathbf{x}_i$ and $\boldsymbol{a}_j$ instances.

We can construct a similarity matrix for the instances of the $\mathbf{X}$ by exploiting the idea that similar instances of the $\mathbf{X}$ should have similar preferences with respect to the instances of the $\mathbf{A}$ matrix. Here we do not rely anymore in the original representation of the $\mathbf{x}$ instances in order to define their similarities but on their preferences with respect to the $\mathbf{a}$ instances[1]. So the $\mathbf{x}$ instances similarity matrix will be the $\mathbf{R}\mathbf{R}^T$ matrix, the $[\mathbf{R}\mathbf{R}^{\mathrm{T}}]_{ij}$ entry of which will give the similarity of the $\mathbf{x}_i$ and $\mathbf{x}_j$ instances. In exactly the same manner we can construct the similarity matrix for the $\mathbf{a}$ instances as $\mathbf{R}^T\mathbf{R}$.

We now want to learn two Mahalanobis metrics one in the $\mathcal{X}$ and one in the $\mathcal{A}$ space which will reflect the instance similarities as these are given by the $\mathbf{R}\mathbf{R}^T$ and $\mathbf{R}^T\mathbf{R}$ similarity matrices respectively. In addition we want to learn a third metric over the two heterogeneous spaces $\mathcal{X}$ and $\mathcal{A}$ which will reflect the similarity/preference of an $\mathbf{x}_i \in \mathbf{X}$ instance to an $\mathbf{a}_j \in A$ instance as this is given by the R_{ij} preference value. Since learning a Mahalanobis metric is equivalent to learning a linear transformation we will see in the following paragraphs that what we actually need to learn is eventually two such linear transformations, one for the $\mathcal{X}$ and one for the $\mathcal{A}$ space, which will optimize the three objective functions that we just skectched.

We should note here that the setting that we just described is not specific to the meta-mining context but is also relevant for any recommendation problem with similar requirements. To the best of our knowledge the metric-based solution which we will present right away is the first of its kind for such settings.

A. Learning a dataset metric

We will now describe how to learn a Mahalonobis metric matrix $\mathbf{W}_{\mathcal{X}}$ in the $\mathcal{X}$ dataset space in a manner that will

[1]This reflects one of the basic assumptions in metalearning, the fact that what we are trying to reflect is a similarity of datasets in terms of the relative performance/appropriateness of different learning paradigms/algorithms for them

reflect datasets similarity in terms of the similarity of their workflow preference vectors. Instead of using the $\mathbf{R}\mathbf{R}^{\mathrm{T}}$ matrix to establish the similarity of two datasets in terms of their preference vectors, under which the dataset similarity is simply the inner product of the workflow preference vectors, we will rely on the Pearson rank correlation coefficient of these preference vectors. The latter is a more appropriate measure of dataset similarity since it focuses on the relative workflow performance which is more relevant when one wants to measure dataset similarity. Nevertheless to simplify notation we will continue using the $\mathbf{R}\mathbf{R}^{\mathrm{T}}$ notation.

We define the following metric learning optimization problem:

$$\begin{aligned} \min_{\mathbf{W}_{\mathcal{X}}} F_1(\mathbf{W}_{\mathcal{X}}) &= ||\mathbf{R}\mathbf{R}^{\mathrm{T}} - \mathbf{X}\mathbf{W}_{\mathcal{X}}\mathbf{X}^{\mathrm{T}}||_F^2 + \mu_1 tr(\mathbf{W}_{\mathcal{X}}) \\ s.t. & \quad \mathbf{W}_{\mathcal{X}} \succeq 0 \end{aligned}$$

where $||.||_F$ is the Frobenius matrix norm, $tr(.)$ the matrix trace, and $\mu_1 \geq 0$ is a parameter controlling the trade-off between empirical error and the metric complexity used to control overfitting, which is a convex optimization problem. As already mentioned learning a Mahalanobis metric matrix is equivalent to learning a linear transformation of the original feature space. Thus we can now rewrite our metric learning problem with the help of a linear transformation as:

$$\min_{\mathbf{U}} F_1(\mathbf{U}) = ||\mathbf{R}\mathbf{R}^{\mathrm{T}} - \mathbf{X}\mathbf{U}\mathbf{U}^T\mathbf{X}^{\mathrm{T}}||_F^2 + \mu_1||\mathbf{U}||_F^2 \quad (1)$$

where $\mathbf{W}_{\mathcal{X}} = \mathbf{U}\mathbf{U}^T$ is the $d \times d$ metric matrix, and $\mathbf{U}$ an associated linear transformation with dimensionality $d \times t$ which projects the dataset description to a new space of dimensionality t. Unlike the previous optimization problem this is no longer convex. We will work with optimization problem (1) because it will make easier the variable sharing between the different optimization problems that we will define. We solve it using gradient descent.

Using the learned metric the similarity of two datasets $\mathbf{x}_i$ and $\mathbf{x}_j$ is $\omega(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i\mathbf{U}\mathbf{U}^T\mathbf{x}_j$. Given some new dataset $\mathbf{x}$ we will use this similarity to establish the set $N_{\mathbf{x}}$ consisting of the N datasets that are most similar to $\mathbf{x}$ with respect to the similarity of their relative workflow preferences as this is computed in the original feature space $\mathcal{X}$. With the help of $N_{\mathbf{x}}$ we can now compute the workflow preference vector of $\mathbf{x}$ as the weighted average of the workflow preference vectors of its nearest neighbors by:

$$\mathbf{r}_{\mathbf{x}} = \zeta_{\mathbf{x}}^{-1} \sum_{x_i \in N_x} \mathbf{r}_{\mathbf{x}_i}\, \omega_{\mathcal{X}}(\mathbf{x}, \mathbf{x}_i) \quad (2)$$

where $\zeta_{\mathbf{x}}$ is a normalization factor given by $\zeta_{\mathbf{x}} = \sum_{\mathbf{x}_i \in X} \omega_{\mathcal{X}}(\mathbf{x}, \mathbf{x}_i)$. Thus using the learned metric we can compute the workflow preference vector $\mathbf{r}_{\mathbf{x}}$ for a new dataset by computing its similarity to the training datasets in the $\mathcal{X}$ feature space, similarity that was learned in a manner that reflects the datasets similarity in terms of their relative workflow preferences.

B. Learning a data mining workflow metric

To learn a Mahalanobis metric matrix $\mathbf{W}$ in the $\mathcal{A}$ data mining workflow space we will proceed in exactly the same manner as we did with the datasets using now the $\mathbf{R}^T\mathbf{R}$ matrix the elements of which will give the rank correlation coefficients of the dataset preference vectors of the workflows, measuring thus the similarity of workflows in terms of their relative performance over the different datasets. More precisely as before we start with the metric learning optimization problem:

$$\begin{aligned} \min_{\mathbf{W}_{\mathcal{A}}} F_2(\mathbf{W}_{\mathcal{A}}) &= ||\mathbf{R}^{\mathrm{T}}\mathbf{R} - \mathbf{A}\mathbf{W}_{\mathcal{A}}\mathbf{A}^{\mathrm{T}}||_F^2 + \mu_1 tr(\mathbf{W}_{\mathcal{A}}) \\ s.t. & \quad \mathbf{W}_{\mathcal{A}} \succeq 0 \end{aligned}$$

which we cast to the problem of learning a linear transformation $\mathbf{V}$ in the workflow space as:

$$\min_{\mathbf{V}} F_2(\mathbf{V}) = ||\mathbf{R}^{\mathrm{T}}\mathbf{R} - \mathbf{A}\mathbf{V}\mathbf{V}^T\mathbf{A}^{\mathrm{T}}||_F^2 + \mu_1||\mathbf{V}||_F^2 \quad (3)$$

where $\mathbf{W}_{\mathcal{A}} = \mathbf{V}\mathbf{V}^T$ is the $l \times l$ metric matrix, and $\mathbf{V}$ an associated linear transformation with dimensionality $l \times t$ that projects workflow descriptions into a new space of t dimensionality. As before this is not a convex optimization problem. We will solve it using gradient descent. Similar to the dataset case using the learned metric the similarity of two workflows $\mathbf{a}_i$ and $\mathbf{a}_j$ is $\omega_{\mathcal{A}}(\mathbf{a}_i, \mathbf{a}_j) = \mathbf{a}_i\mathbf{V}\mathbf{V}^T\mathbf{a}_j$. Given some new workflow $\mathbf{a}$ its workflow neighborhood $N_{\mathbf{a}}$ consists of the N workflows that are most similar to $\mathbf{a}$ with respect to the similarity of their relative dataset preferences as this is computed in the original feature space $\mathcal{A}$. With the help of $N_{\mathbf{a}}$ we can now compute the dataset preference vector of $\mathbf{a}$ as the weighted average of the dataset preference vectors of its nearest neighbors by:

$$\mathbf{r}_{\mathbf{a}} = \zeta_{\mathbf{a}}^{-1} \sum_{a_i \in N_a} \mathbf{r}_{\mathbf{a}_i}\, \omega_{\mathcal{A}}(\mathbf{a}, \mathbf{a}_i) \quad (4)$$

where $\zeta_{\mathbf{a}}$ is a normalization factor given by $\zeta_{\mathbf{A}} = \sum_{\mathbf{A}_i \in A} \omega_{\mathcal{A}}(\mathbf{a}, \mathbf{a}_i)$. Thus using the learned metric we can compute the workflow preference vector $\mathbf{r}_{\mathbf{x}}$ for a new dataset by computing its similarity to the training datasets in the $\mathcal{X}$ feature space, similarity that was learned in a manner that reflects the datasets similarity in terms of their relative workflow preferences.

C. Learning a heterogeneous metric over datasets and workflows

The last metric that we want to learn is one that will relate datasets to data mining workflows reflecting the appropriateness/preference of a given workflow for a given dataset in terms of the relative performance of the former applied to the latter. We will do so by starting with the following metric learning optimization problem

$$\begin{aligned} \min_{\mathbf{W}} F_3(\mathbf{W}) &= ||\mathbf{R} - \mathbf{X}\mathbf{W}\mathbf{A}^{\mathrm{T}}||_F^2 + \mu_1 tr(\mathbf{W}) \\ s.t. & \quad \mathbf{W} \succeq 0 \end{aligned}$$

which if we parametrize the $d \times l$ metric matrix $\mathbf{W}$ with the help of two linear transformation matrices $\mathbf{U}$ and $\mathbf{V}$ with dimensions $d \times t$ and $l \times t$ can be rewritten as:

$$\min_{\mathbf{U},\mathbf{V}} F_3(\mathbf{U},\mathbf{V}) = ||\mathbf{R} - \mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{A}^{\mathrm{T}}||_F^2 + \mu_1||\mathbf{U}||_F^2 + \mu_2||\mathbf{V}||_F^2 \quad (5)$$

Essentially what we do here is to project the descriptions of datasets and workflows to a common space with dimensionality t over which we compute their similarity in a manner that reflects the preference matrix $\mathbf{R}$. We will set t to the $\min(rank(\mathbf{A}), rank(\mathbf{X}))$. In other words we learn a heterogeneous metric which computes similarities of datasets and workflows in terms of the relative performance of the latter when applied on the former. Using the new similarity metric we can now compute directly the match between a dataset $\mathbf{x}$ and a workflow $\mathbf{a}$ as:

$$r_{\mathbf{x},\mathbf{a}} = \mathbf{x}\mathbf{U}\mathbf{V}^T\mathbf{a} \quad (6)$$

Clearly we can use this not only to determine the goodness of match between a dataset and a data mining workflow but also given some dataset and a set of workflows to order the latter according to their appropriateness with respect to the former, thus solving the meta-mining task 1, and vice versa given a workflow and a set of datasets to order the latter according to their appropriateness for the former thus solving meta-mining task 2.

In the objective function of the optimization problem (5) we focus exclusively on trying to learn a metric that will reflect the appropriateness of some workflow for some dataset as this is given by the entries of the $\mathbf{R}$ preference matrix. However there is additional information that we can bring in if we exploit the objective functions of the optimization problems (1) and (3) and use them to additionally regularize the objective function of (5). The overall idea here is that we will learn three different metrics in the spaces of datasets, workflows, and datasets-workflows, all of them parametrized by two linear transformations in a manner that will reflect the basic meta-mining assumptions, namely that similar datasets should have similar workflow preference vectors, similar workflows should have similar dataset preference vectors, and that the heterogeneous metric between datasets and workflows should reflect the appropriateness of datasets for workflows. By combining the three optimization problems of (1), (3), and (5) we get the following metric learning optimization problem that achieves these goals:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} F_4(\mathbf{U},\mathbf{V}) &= \alpha F_1(\mathbf{U}) + \beta F_2(\mathbf{V}) + \gamma F_3(\mathbf{V},\mathbf{U}) \quad (7)\\ &= \alpha||\mathbf{R}\mathbf{R}^{\mathrm{T}} - \mathbf{X}\mathbf{U}\mathbf{U}^T\mathbf{X}^{\mathrm{T}}||_F^2\\ &+ \beta||\mathbf{R}^{\mathrm{T}}\mathbf{R} - \mathbf{A}\mathbf{V}\mathbf{V}^T\mathbf{A}^{\mathrm{T}}||_F^2\\ &+ \gamma||\mathbf{R} - \mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{A}^{\mathrm{T}}||_F^2\\ &+ \mu_1||\mathbf{U}||_F^2 + \mu_2||\mathbf{V}||_F^2\end{aligned}$$

where α, β, γ, are positive parameters that control the importance of the three different optimization terms. As it was the case with optimization problem (5) this optimization problem can also be used to address all three meta-mining tasks. In fact (7) is the most general formulation of the metric-learning based hybrid reccomendation problem and includes as special cases problems (1) and (3).

Matrix factorization, often used in recommender systems, also learns a decomposition of a matrix to component matrices $\mathbf{U}$ and $\mathbf{V}$ under different constraints. However, by its very nature it cannot handle well the out-of-sample problem. The objective function of problem (7) uses as additional constraints the objective functions of (1) and (3) and learns a common space for the datasets and workflows, which are induced by the projection matrices $\mathbf{U}$ and $\mathbf{V}$. As a result, the out-of-sample problem, i.e. cold start problem in recommender system, is naturally handled by the optimization problem (7).

IV. Dataset and workflow descriptors

In the following two sections we will describe the dataset and workflow descriptors that we will use in our meta-mining experiments.

A. Dataset Decriptors

Originally proposed by the STATLOG project [10], the idea of characterizing datasets has been the main stream in meta-learning during these last decades [11], [12], [13], [14]. Various characterizations have been subsequently proposed, from which we have selected the most relevant ones summarized as follows:
statistical measures: number of instances, number of classes, proportion of missing values, proportion of continuous / categorical features, noise signal ratio.
information-theoretic measures: class entropy, mutual information.
geometrical and topological measures [15]: non-linearity, volume of overlap region, maximum fisher's discriminant ratio, fraction of instance on class boundary, ratio of average intra/inter class nearest neighbour distance.
model-based measures: error rates and pairwise $1-p$ values obtained by landmarkers [16] such as ZeroR, one-nearest-neighbor, Naive Bayes, Decision Stumps [17], Random Trees [18], and the linear SVM [19], and the distributions of the weights learned by the Relief [20] and SVMRFE [21] feature selection algorithms.
Overall, we use a large spectrum of dataset characteristics, from very simple ones such as the number of instances to more elaborated ones such as the model-based measures, giving a total number of $d = 150$ dataset characteristics.

B. Workflow descriptors

The ability to describe data mining algorithms and workflows and use these descriptors for meta-learning and meta-mining is a very recent development [5]. There the authors

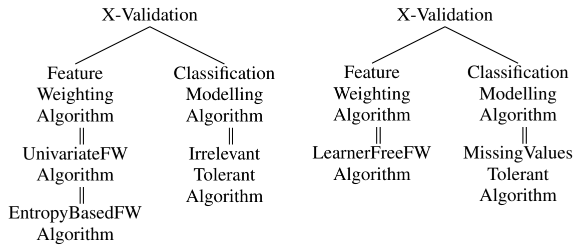

Figure 1. Two workflow patterns with cross-level concepts. Thin edges depict workflow decomposition; double lines depict DMOP's concept subsumption.

used DMOP, a data mining ontology, to describe learning algorithms and data-processing algorithms such as feature selection, discretization and normalization, with respect to the mathematical concepts they implement and different properties, such as their bias/variance profile, their sensitivity to the type of attributes, their learning strategy, etc. In addition the same ontology allows to anotate operators (algorithm implementations) of data mining workflows with their respective concepts. A data mining workflow is typically a direct acyclic graph of data mining operators.

In order to describe the data mining workflows we follow the propositionalization approach used in [5]. We derive from the annotated direct acyclic graphs that describe the data mining workflows a set of frequent closed workflow patterns using the tree-structured apriori algorithm of [22]. The description of a workflow is then given by a binary vector that indicates the presence or absence of each of the frequent patterns; the final workflow description contains $l = 214$ features. In figure 1 we give two examples of workflow patterns that have been abstracted from ground feature selection + classification workflows based on DMOPs algorithm hierarchy. These patterns help us understand how the workflow space is structured by describing frequent workflow structures using the DMOP concepts.

V. Experiments

In this section, we will perform a systematic evaluation to examine the performance of the different metric learning optimization problems for meta-mining that we presented in the previous sections. More precisely we will evaluate the performance of the dataset metric learning optimization problem given in (1) to the meta-mining task of learning workflow preferences for a given dataset; the performance of the workflow metric learning optimization problem of (3) to the meta-mining task of learning dataset preferences; and finally the performance of the two metric learning optimization problems, (5), (7), for all three meta-mining tasks.

A. Base-level Experiments

In order to meta-mine we first need to perform a set of base-level experiments over which we will construct our meta-mining models. To do so we used 65 real world cancer microarray datasets, most of them were taken from the National Center for Biotechnology Information [2]. Microarray datasets are characterized by a high-dimensionality and a small sample size, and a relatively low number of classes, most often two. These datasets have an average of 79.26 instances, 15268.57 attributes, and 2.33 classes. On these datasets we applied a total of 35 classification data mining workflows; 28 of them were workflows that contained one feature selection and one classification algorithm, while the seven remaining ones had only a single classification algorithm. We used the four following feature selection algorithms: Information Gain, *IG*, Chi-square, *CHI*, ReliefF [20], *RF*, and recursive feature elimination with SVM [21], *SVM-RFE*, and fixed the number of selected features to ten. For classification we used the seven following algorithms: one-nearest-neighbor, *1NN*, the *C4.5* [23] and *CART* [24] decision tree algorithms, a Naive Bayes algorithm with normal probability estimation, *NBN*, a logistic regression algorithm, *LR*, and SVM [19] with the linear, SVM_l and the rbf, SVM_r, kernels. We used the implementations of these algorithms provided by the RapidMiner data mining suite [25] with their default parameters. Overall we had a total of $65 \times (28+7) = 2275$ base-level DM experiments, i.e. applications of these workflows on the datasets. To construct the $\mathbf{R}$ preference matrix we estimated the performance of the workflows using 10-fold cross-validation and applied the scoring McNemar based scoring schema described in section II. In table I we give for each of the ten top workflows over the full set of 65 datasets the number of times that these were ranked in the top five positions.

B. Baseline Strategies and Evaluation Methodologies

In order to assess how well the different variants perform we need to compare them with some default and baseline strategies. For the meta-mining task of workflow preference learning, we will use as the default strategy the preference vector given by average of the workflow preference vectors over the different training datasets for a given testing dataset. We should note that this is a rather difficult baseline to beat since the different workflows will be ranked according to their average performance on the training datasets, with workflows that perform consistently well ranked on the top. For the second task of providing a dataset preference vector for a given testing workflow we have a similar default strategy, we will use the average of the dataset preference vectors over the different training workflows. However this strategy for the workflows leads to a trivial constant vector of dataset preferences due to the fact that the total sum of

[2] http://www.ncbi.nlm.nih.gov/

1	2	3	4	5	6	7	8	9	10
LR	*IG+LR*	*RF+LR*	*SVMRFE+SVM$_l$*	*SVMRFE+LR*	*IG+NBN*	*IG+SVM$_l$*	*CHI+NBN*	*SVMRFE+NBN*	*RF+SVM$_l$*
25	12	13	16	13	17	14	10	13	8

Table I
DEFAULT TOP-10 WORKFLOWS AND THEIR FREQUENCY IN THE TOP-5 POSITIONS.

Learning Workflow preferences

	ρ	t5p	mae
def	0.332	77.8	4.50
EC	**0.356**	77.8	**4.39**
δ	32/65 p=1	32/65 p=1	37/65 p=0.321
F_1	**0.366**	**78.6**	4.83
δ	34/65 p=0.804	33/65 p=1	40/65 p=0.082
δ_{EC}	35/65 p=0.620	33/65 p=1	20/65 p=0.003
F_3	0.286	77.1	5.64
δ	23/65 p=0.025	23/65 p=0.025	19/65 p=0.001
δ_{EC}	19/65 p=0.001	27/65 p=0.215	14/65 p=1e-6
F_4	**0.391**	**79.3**	**4.22**
δ	40/65 p=0.082	41/65 **p=0.047**	46/65 **p=0.001**
δ_{EC}	42/65 **p=0.025**	44/65 **p=0.006**	42/65 **p=0.025**

Learning Dataset preferences

	ρ	mae
def	NA	4.84
EC	0.375	**4.29**
δ	NA	30/35 **p=2e-5**
F_2	0.445	**3.90**
δ	NA	29/35 **p=1e-3**
δ_{EC}	23/35 p=0.09	22/35 p=0.176
F_3	0.478	**4.47**
δ	NA	26/35 **p=0.006**
δ_{EC}	22/35 p=0.176	17/35 p=1
F_4	0.491	**4.39**
δ	NA	28/35 **p=0.005**
δ_{EC}	24/35 **p=0.041**	18/35 p=1

Learning DS-WF preferences

	mae
def	4.84
-	-
-	-
-	-
-	-
-	-
F_3	6.46
δ	880/2275 p=0
-	-
F_4	4.99
δ	1070/2275 p=0.005
-	-

Table II
EVALUATION RESULTS. δ AND δ_{EC} DENOTE COMPARISON RESULTS WITH THE DEFAULT (*def*) AND THE EUCLIDEAN BASELINE STRATEGY (*EC*) RESPECTIVELY. ρ IS THE SPEARMAN'S RANK CORRELATION COEFFICIENT, IN T5P WE GIVE THE AVERAGE ACCURACY OF THE TOP FIVE WORKFLOWS PROPOSED BY EACH STRATEGY, AND MAE IS THE MEAN AVERAGE ERROR. X/Y INDICATES THE NUMBER OF TIMES X THAT A METHOD WAS BETTER OVERALL THE EXPERIMENTS Y THAN THE DEFAULT OR THE BASELINE STRATEGY.

workflow points for a given dataset is fixed to $m(m-1)/2$, when we compare m workflows, by the very same nature of the workflow ranking schema for a given dataset. Finally for the last meta-mining task we will use as the default strategy for the prediction for the appropriateness of a workflow for a dataset the average over the values of the preference matrix of the training set. We will denote the default strategy used in the three meta-mining tasks by *def*. In addition we will also have as a baseline strategy the provision of recommendation when we use a simple Euclidean distance, i.e. all attributes are treated equally and there is no learning, which we will denote by *EC*. However this baseline is only applicable to the first two meta-mining tasks, learning workflow preferences and learning dataset preferences, since it cannot be applied to the kind of heterogeneous similarity problem that we have in the third meta-mining task.

As resampling techniques we will use leave-one-dataset-out to estimate the performance on the workflow preference learning task, leave-one-workflow-out for the dataset preference learning task, and leave-one-dataset-and-one-workflow-out for the third task of predicting the appropriateness of a workflow for a dataset.

To quantify the performance we will use a number of evaluation measures. For the first two meta-mining tasks we will report the average Spearman's rank correlation coefficient between the predicted preference vector and the real preference vector over the testing instances. We will denote this average by ρ. This measure will indicate the degree to which the different methods predict correctly the preference order. Note that this quantity is not computable for the default strategy in the case of the learning dataset preferences task, due to the fact that the dataset preference vector that it produces is fixed, as we explained previously, and the Spearman rank correlation coefficient is not computable when one of the two vectors is fixed. In addition to the Spearman rank correlaction coefficient for the meta-mining task of learning workflow preferences we will also report the average accuracy of the top five workflows suggested by each method, measure which we will denote by *t5p*. Finally for the three meta-mining tasks we will also report the mean average error, mae, over the respective testing instances, of the predicted values for $\mathbf{r_x}$, $\mathbf{r_a}$, and $\mathbf{r_{x,a}}$, for learning workflow preferences, dataset preferences, and dataset-workflow preferences, respectively, and the true values. For each measure, method, and meta-mining task, we will give the number of times that the method was better than the respective default and baseline strategies over the total number of datasets, workflows, or dataset, workflow pairs (depending on the meta-mining task), as well as the statistical significance of the result under a binomial test with a statistical significance level of 0.05. The comparison results with the default strategy will be denoted by δ while the comparison to the Euclidean baseline by δ_{EC}.

C. Experiment Results on the Biological Datasets

We will now take a close look on the experimental results for the different meta-mining tasks and objective functions that we have presented to address them. The full results are given in Table II.

Learning Workflow Preferences: Learning algorithm preferences is the most popular formulation in the traditional stream of meta-learning. There given a dataset description we seek to identify the algorithm that will most probably deliver the best results for the given dataset. In that sense this meta-mining task is the most similar to the typical meta-learning task. We have presented three different objective functions that can be used to address this problem. F_1, optimization problem (1), makes makes use of only the dataset descriptors and learns a similarity measure in that space that best approximates their similarity with respect to their relative workflow preference vectors. In traditional meta-learning this similarity is computed directly in the dataset space, it is not learned, and most importantly it does not try to model the relative workflow preference vector, [11], [26]. In our experimental setting the strategy that implements this traditional meta-learning approach is the Euclidean distance-based dataset similarity, *EC*. In addition to the homogeneous metric learning approach we can also use the two heterogeneous metric learning variants to provide the workflow preferences. The simplest one, corresponding to the optimization function F_3, optimization problem 5, uses both dataset and workflow characteristics and tries to directly approximate the relative preference matrix. However this approach ignores the fact that the learned metric should reflect two basic meta-mining requirements, that similar datasets should have similar workflow preferences, and that similar workflows should have similar dataset preferences. The optimization function F_4, optimization problem 7, reflects exactly this bias by regularizing appropriately the learned metrics in the dataset and workflow spaces so that they reflect well the similarities of the respective preference vectors. Before discussing the actual results, given in the left table of Table II, we give the parameter settings for the different variants. F_1: $\mu_1 = 0.5$, $N_{x_n} = 5$; F_3: $\mu_1 = \mu_2 = 0.5$; F_4: $\alpha = 1e^{-10}$, $\beta = 1e^{-3}$, $\gamma = 1e^{-3}$, $\mu_1 = 10$, $\mu_2 = 0$. These parameters reflect what we think are appropriate choices based on our prior knowledge of the meta-mining problem. Better results would have been obtained if we had tuned, at least some of them, via inner cross validation.

Looking at the actual results we see right away that the approach that makes use of only the dataset characteristics, F_1, has a performance that is not statistically significant different neither from the default, nor from the EC baseline with respect to the Spearman's rank correlation coefficient, ρ, and the average accuracy of the top five workflows it suggests, tp5. In addition it is statistically significant worse than the EC with respect to the mean average error criterion, mae, having a lower mae value than *EC* only in 20 out of the 65 datasets. Looking at the performance of the heterogeneous metric that tries to directly approximate the preference matrix $\mathbf{R}$, we see that its results are quite disappointing. It is significant worse than the default strategy and the *EC* baseline for almost all performance measures. So trying to learn a heterogeneous metric that relies exclusively on the approximation of the preference matrix is definitely not an option. However when we turn to the F_4 objective function that learns the heterogeneous metrics in a manner that they do not only reflect the preference manner, but also the fact that similar datasets should have similar workflow preferences and vice versa, there we see that the performance we get is excellent. F_4 beats in a statistically significant manner both the default strategy as well as the *EC* baseline in almost cases, the only exception is the Spearman's correlation coefficient comparison with the default where the level of significance is high, $p = 0.082$, but does not overpass the significance threshold of 0.05. Overall in such a recommendation scenario the best strategy consists in learning a combination of the two homogeneous and one heterogeneous metrics that reflect the similarities of the datasets with respect to the workflow preferences, the similarities of the workflows with respect to the dataset preference vectors, as well as the similarities of workflows-datasets according to the preference matrix.

Learning Dataset Preferences: The goal of this meta-mining task is given a new workflow and a collection of datasets to provide a dataset preference vector that will reflect the order of appropriateness of the datasets for the given workflow. As already mentioned the default strategy provides here a vector of equal ranks thus we cannot compute its Sperman's rank correlation coefficient. We will compare the performance of the F_2 objective function that makes use of only of the workflow descriptors when it tries to approximate the similarity of the dataset preference vectors, and these of F_3 and F_4. We used the following parameter: F_2: $\mu_1 = 10$, $N_{a_n} = 5$; F_3: $\mu_1 = \mu_2 = 10$; F_4: $\alpha = 1e^{-10}$, $\beta = 1e^{-3}$, $\gamma = 1e^{-3}$, $\mu_1 = 0.5$, $\mu_2 = 0$. Looking at the results, middle table of Table II, we see that when it comes to the mean average error, all methods achieve a performance that is statistically significant better than that of the default strategy, suggesting that this meta-mining task is probably easier than the first one. This makes sense since it is easier to describe a workflow similarity in terms of the concepts that these workflows use, than what it is to describe a dataset similarity in terms of the datasets characteristics. Neither F_3 nor F_4 have a mae performance that is statistically significant better than the Euclidean baseline. Nevertheless F_4 is statistically significant better than the Euclidean when it comes to the Sperman's rank correlation coefficient. Thus for this meta-mining task there is also evidence that we should take a more global approach by accounting for all the different constraints on the dataset and workflow metrics as F_4 does.

Learning Dataset-Workflow Preferences: The last meta-mining task is by far the most difficult one. Here we want to predict the appropriateness of a new workflow for a new dataset, i.e. the $r_{\mathbf{x},\mathbf{a}}$ value. The only metric functions that are applicable here are F_3 and F_4, since these are the only ones that are heterogeneous, i.e. they can compute a similarity between a dataset and a workflow. Note also that the Euclidean baseline strategy is no longer applicable because this can only be used between objects of the same type. When it comes to the mean average error F_3 has a very poor performance compared to the default strategy. F_4 has a considerably better performance than F_3, thus providing further support to the incorporation of the additional constraints in the objective function, nevertheless this performance still is significantly worse than the default that of the default strategy.

Overall we tested a number of new metric-learning-based algorithms to solve different variants of the meta-mining problem following a hybrid recommendation approach. We have two metric-based-learning flavors, the homogeneous and the heterogeneous. In the homogeneous flavor we learn a metric in the original space in which some objects are described, here datasets or workflows, which tries to approximate a similarity defined over a different space that of the relative preference vectors. In the heterogeneous approach we learn a metric over the two different spaces that tries to reflect directly the goodness of match between the different objects. As it turns out the best approach comes from the appropriate regularization of the heterogeneous metric by exploiting the additional constrains imposed on each of the original object spaces. In other words we seek for an heterogeneous metric defined over a common projection space of datasets and workflows, where the projection matrices of the datasets and workflows are constrained to reflect vector preference similarities. In the immediate future we want to evaluate the performance of the approach we presented here in recommendations problems other than the meta-mining with similar problem requirements.

VI. Related Work

The meta-mining problem formulation we gave here is closely related with the work of hybrid recommender systems [6], [7], [8], [9]. There the goal is to accurately recommend items to users using information on historical user preferences and descriptors of items and users. Examples of recommender user and item descriptors can be found for instance on the MovieLens dataset where we have demographic or activity information on users such as age, gender and occupation, and taxonomic information on movies such as genre and release date.

State of the art recommender methods [6], [7], [8], rely on matrix factorization methods to directly approximate the preference matrix as we do in the optimization problem (5). In [8], the authors proposed a Bayesian approach where a probabilistic bi-linear rating model is inferred by a combination of expectation propagation and variational message passing. Users and items features are modelled with Gaussian priors into two matrices $\mathbf{U}$ and $\mathbf{V}$ of latent *traits*, the inner product of which defines user-item similarities. Variational approximation on users and items is then used to regularize the latent factors. Their experiments on the MovieLens dataset showed that including user and item descriptors improves performance. [6], [7], propose also a generative probabilistic model where the model fitting is done by a Monte Carlo EM algorithm with no variational approximations. They regularize their model using a regression-based approach on user and item factors, where the latter are determined using topic modelling, [6]. They also experimented on the MovieLens dataset and showed that a model based on the meta-data had a weak predictive performance while their regression-based approach to the latent factors regularization gave the best performance improvements. Our metric-learning-based approach to the problem of hybrid recommendation uses a very different regularization approach in learning the factorization matrices, we focus on constraining them in a manner that they will reflect in the original feature spaces the similarities of the respective preference vectors, approach which in our meta-mining experiments had the best performance. One additional advantage of the use of the two linear projection matrices learned in the dataset and workflow spaces is that we can now naturally handle the out-of-sample problem, i.e. the cold-start problem in recommender systems, which is not the case with the typical matrix factorization models.

VII. Conclusion and Future Work

In this paper we take a new view on the relatively new concept of meta-mining, view that is also relevant for the more traditional work of meta-learning. We model the problem of the selection of the appropriate workflow or algorithm for a dataset as a hybrid recommendation problem, in which suggestions will be provided based on the descriptors of the dataset and the workflow or algorithm. To that end we propose a new metric-learning-based approach to the hybrid recommendation problem, which learns homogeneous metrics in the original dataset and workflow spaces, constrained in a manner that will reflect workflow preference and dataset preference vector similarities, and combines them with an heterogeneous metric in the dataset-workflow space that reflects the appropriateness of a given workflow for a given dataset. The two homogeneous metric-learning problems act as additional, relevant, regularizers for the heterogeneous metric learning problem. In addition thanks to the linear projections that lie at the core of our method, it is able to handle in a natural manner the cold-start problem. The combined use of the three metrics achieves the best results. To the best of our knowledge this the first approach of its kind, not only for the meta-mining problem, but as well as for the more general problem of the hybrid recommendation. Our immediate goal is to experiment with our approach to standard hybrid recommendation problems, such as the MovieLens dataset, and compare its performance with typical recommendation approaches used in such problems.

REFERENCES

[1] C. Giraud-Carrier, R. Vilalta, and P. Brazdil, "Introduction to the special issue on meta-learning," *Mach. Learn.*, vol. 54, no. 3, pp. 187–193, Mar. 2004.

[2] M. L. Anderson and T. Oates, "A review of recent research in metareasoning and metalearning," *AI Magazine*, vol. 28, no. 1, pp. 7–16, 2007.

[3] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[4] R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares, "Using meta-learning to support data mining," *International Journal of Computer Science Applications*, vol. 1, no. 1, pp. 31–45, 2004.

[5] M. Hilario, P. Nguyen, H. Do, A. Woznica, and A. Kalousis, "Ontology-based meta-mining of knowledge discovery workflows," in *Meta-Learning in Computational Intelligence*, N. Jankowski, W. Duch, and K. Grabczewski, Eds. Springer, 2011.

[6] D. Agarwal and B.-C. Chen, "flda: matrix factorization through latent dirichlet allocation," in *Proceedings of the third ACM international conference on Web search and data mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 91–100.

[7] ——, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 19–28.

[8] D. H. Stern, R. Herbrich, and T. Graepel, "Matchbox: large scale online bayesian recommendations," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 111–120.

[9] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[10] R. D. King, C. Feng, and A. Sutherland, "Statlog: Comparison of classification algorithms on large real-world problems," *Applied Artificial Intelligence*, vol. 9, no. 3, pp. 289–333, 1995.

[11] C. Soares and P. Brazdil, "Zoomed ranking: Selection of classification algorithms based on relevant performance information," in *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, ser. PKDD '00. London, UK: Springer-Verlag, 2000, pp. 126–135.

[12] C. Köpf, C. Taylor, and J. Keller, "Meta-analysis: From data characterisation for meta-learning to meta-regression," in *Proceedings of the PKDD-00 Workshop on Data Mining, Decision Support,Meta-Learning and ILP*, 2000.

[13] M. Hilario and A. Kalousis, "Fusion of meta-knowledge and meta-data for case-based model selection," in *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, ser. PKDD '01. London, UK, UK: Springer-Verlag, 2001, pp. 180–191.

[14] A. Kalousis, J. Gama, and M. Hilario, "On data and algorithms: Understanding inductive performance," *Machine Learning*, vol. 54, no. 3, pp. 275–312, 2004.

[15] T. K. Ho and M. Basu, *Data complexity in pattern recognition*. Springer, 2006.

[16] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier., "Meta-learning by landmarking various learning algorithms." *Proc. 17th International Conference on Machine Learning*, pp. 743–750, 2000.

[17] W. Iba and P. Langley, "Induction of one-level decision trees," in *Proceedings of the Ninth International Workshop on Machine Learning*, ser. ML '92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 233–240.

[18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[20] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Mach. Learn.*, vol. 53, no. 1-2, pp. 23–69, Oct. 2003.

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389–422, 2002.

[22] M. J. Zaki, "Efficiently mining frequent trees in a forest: Algorithms and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1021–1035, Aug 2005, special issue on Mining Biological Data.

[23] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, ser. Statistics/Probability Series. Belmont, California, U.S.A.: Wadsworth Publishing Company, 1984.

[25] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, "Yale: rapid prototyping for complex data mining tasks," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006, pp. 935–940.

[26] A. Kalousis and T. Theoharis, "Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection," *Intell. Data Anal.*, vol. 3, no. 5, pp. 319–337, 1999.