Ensemble Learning for Multi-type Classification in Heterogeneous Networks

Francesco Serafino, Gianvito Pio, Michelangelo Ceci

Abstract—Heterogeneous networks are networks consisting of different types of objects and links. They can be found in several fields, ranging from the Internet to social sciences, biology, epidemiology, geography, finance and many others. In the literature, several methods have been proposed for the analysis of network data, but they usually focus on homogeneous networks, where all the objects are of the same type, and links among them describe a single type of relationship. More recently, the complexity of real scenarios has impelled researchers to design methods for the analysis of heterogeneous networks, especially focused on classification and clustering tasks. However, they often make assumptions on the structure of the network that are too restrictive or do not fully exploit different forms of network correlation and autocorrelation. Moreover, when nodes which are the main subject of the classification task are linked to several nodes of the network having missing values, standard methods can lead to either building incomplete classification models or to discarding possibly relevant dependencies (correlation or autocorrelation). In this paper, we propose an ensemble learning approach for multi-type classification. We adopt the system Mr-SBC, which is originally able to analyze heterogeneous networks of arbitrary structure, within an ensemble learning approach. The ensemble allows us to improve the classification accuracy of Mr-SBC by exploiting *i)* the possible presence of correlation and autocorrelation phenomena, and *ii)* the classification of instances (which contain missing values) of other node types in the network. As a beneficial side effect, we have also that the models are more stable in terms of standard deviation of the accuracy, over different samples used for training. Experiments performed on real-world datasets show that the proposed method is able to significantly outperform the standard implementation of Mr-SBC. Moreover, it gives Mr-SBC the advantage of outperforming four other well-known algorithms for the classification of data organized in a network.

Index Terms—Heterogeneous networks, ensemble learning, multi-type classification

1 Introduction

In the real world we can easily find objects which appear to be connected to each other, thus forming complex networks. Connections among these objects can represent different types of relationships and interactions, which can be found in several fields, including biology, epidemiology, geography, finance, and many others. Most of the works in the literature about mining networked data focus on homogeneous networks, where all the objects are of the same type (and can, accordingly, be represented by a predefined set of features/characteristics) and the links among them describe a single type of relationship. A common example of a homogeneous network can be found in social networks, where objects represent people and links represent the friendship relationship among them. However, real scenarios are more complex due to the presence of multiple types of objects that are connected through different types of links, forming heterogeneous information networks. For example, in well-known databases about movies (e.g., IMDb) we have movies, actors, users, tags, etc. In the bio-medical domain, databases contain genes, proteins, tissues, pathways and diseases. In databases about reviews of travel-related content or about local businesses (e.g. TripAdvisor, Yelp), we can find activities, users, reviews. In all these cases, objects of different types establish different types of relationships.

Consequently, recent works have proposed new data mining methods that work on heterogeneous information networks. For example, in [1] and [2] the authors propose new clustering solutions, whereas in [3] and [4] the authors propose classification/prediction methods. Other methods that were initially proposed for relational data can be almost directly applied for the analysis of heterogeneous information networks [5]. However, existing methods (see Section 2 for a comprehensive overview) suffer from one or more of the following limitations:

1) **Network structure**. They impose strict restrictions on the structure of the network (e.g., star-structured networks, bi-typed networks, etc.), which often is not sufficient to represent real-world scenarios;
2) **Network autocorrelation**. Although they are generally able to exploit the network structure, in order to capture some forms of correlation among different attributes, they are not able to take into account (and possibly exploit) one of the main peculiarities of network data, i.e. the presence of different forms of *autocorrelation* (see [6] and [7]). In particular, in a network, this implies that the value of an attribute at a given node may depend on the values of the same attribute of the nodes it is directly or indirectly connected with. When this phenomenon occurs among class labels, it becomes even more relevant, since capturing it means being able to exploit such dependencies to improve the classification accuracy.
3) **Missing values**. They are not able to consider the possible presence of missing values for some attributes, which can lead to either learning incomplete classification models or to discarding possibly relevant dependencies. This issue is commonly

• *F. Serafino, G. Pio and M. Ceci are in the Department of Computer Science, University of Bari Aldo Moro, Bari (Italy)*
E-mail: name.surname@uniba.it

faced by either ignoring the value of those attributes or by replacing missing values with some computed/predicted values, usually obtained by simple aggregation functions (e.g., average or mode, computed over the same attribute of the other objects).

4) **Multi-type classification**. In some cases we could be interested in classifying objects of different types, where each type can be associated with a different set of possible labels. This task can be considered a generalization of the standard "multi-target classification" task [8]. The difference is that, in our case, objects to be classified belong to different types. Most methods are not able to solve this task (i.e., to predict the labels for all the object types simultaneously), but have to consider several independent classification tasks. More recent works which overcome this limitation [4], however, assume a common set of possible labels for every type of object.

In this paper, we propose a classification method which takes into account all these issues. In particular, the proposed method works on heterogeneous networks with arbitrary structures and is able to capture both correlation and autocorrelation phenomena which involve the target objects (i.e., objects which are the main subject of the classification task). Moreover, we aim at exploiting the same strategy to predict possibly relevant missing values belonging to other objects, appearing strongly related to the target objects. Methodologically, we extend the method Mr-SBC [9], which is only able to deal with issue 1 (it can potentially analyze networks of arbitrary structures), in order to also capture network autocorrelation phenomena (issue 2), handle relevant missing values (issue 3) and perform multi-type classification (issue 4). These last three issues are tackled by resorting to a combined bagging-boosting ensemble learning solution able to exploit information conveyed by objects (also of the same type) directly or indirectly connected to the main subject(s) of the classification task. The way we generate and use the ensemble allows us to naturally take into account the following properties of network data:

- The value of an attribute of an object may depend on the values of the *other attributes* of the same object or of other objects of *any type*, directly or indirectly connected to it (*network correlation*);
- The value of an attribute of an object may depend on the value of the *same attribute* of other objects of the *same type*, (indirectly) connected to it (*autocorrelation*);
- The value of an attribute (also the class attribute) could depend on the attributes of connected objects of any type, whose value is *initially unknown*.

Specifically, we propose two extensions of the Mr-SBC algorithm. The first is able to work with networks of arbitrary structures (issue 1) and to capture network autocorrelation (issue 2). The second also aims at handling relevant missing values and multi-type classification (issues 3 and 4):

- **ST-MrSBC (Self-Training MrSBC)**, which is able to capture possible autocorrelation phenomena by resorting to a variant of the self-training method. In this case, the ensemble consists of the classifiers built over all the iterations of the self-training approach.
- **MT-MrSBC (Multi-Type MrSBC)**, which iteratively analyzes objects of multiple types, in order to predict possible missing values. These missing values can belong either to the target type of the main classification task or to some other object types that are strongly related to the main classification task. In this case, we build an ensemble of classifiers for each target type, which is able not only to catch autocorrelation phenomena, but also dependencies among different types, since each classifier will be built also on the basis of the predictions obtained for other target types in the previous iterations.

It is noteworthy that we consider the network classification task according to the *within-network* setting [10]: objects for which the class is known are linked to objects for which the class must be estimated [11] (which can be either the subject of the main classification task or other objects related to the main classification task). This setting is semi-supervised and differs from the *across-network* setting (considered in the original Mr-SBC), where the problem is learning from one (labeled) network and applying the learned models to a separate, presumably similar network (see [12] and [7]).

The semi-supervised solution, in addition to allowing the classification phase to take advantage of both labeled and unlabeled examples, leads to smooth predictions. In fact, the idea of the popular semi-supervised smoothness assumption (valid in semi-supervised learning) is to smooth the prediction function in highly populated regions. It states that if two points x_i and x_j in a high density region are close, then also their outputs y_i and y_j should be close [13]. In multi-type classification we add an additional mechanism to smooth the prediction function: capturing relationships among objects of different types and, specifically, capturing relationships among labels associated to objects of different types introduces some forms of correlation among labels, with the result of smoothing predictions on different types of objects. In fact, the semi-supervised setting, combined with the ensemble learning approach and the multi-type classification, provides a solution for the problem of marked discontinuities of the prediction function, consequently providing, in principle, simpler models with less overfitting.

In the following section, we report some details about the work related to the present paper. In Section 3, we report some background notions and introduce the system Mr-SBC. In Sections 4 and 5 we describe the proposed framework and its time complexity, while in Section 6 we show the results obtained on some real-world datasets with both the considered variants of Mr-SBC and with some competitor systems. Finally, in Section 7 we draw some conclusions and introduce possible future work.

2 Related work

The method we propose has its roots in the research areas of network data classification and multi-target prediction. In the following, we discuss some related work in both areas.

2.1 Network data classification

In the literature, several approaches have been proposed for network classification. Most of them work in the within-

network setting, that is, they model a partially labeled network and provide estimates of labels for unlabeled nodes. These approaches have been studied in the research fields of collective inference (see [11], [14], [15] and [16]), active inference (see [17]), semi-supervised and transductive inference (see [18], [19] and [20]). All these approaches, however, are designed to work with homogeneous networks. Only recently some works have started to consider the heterogeneity of nodes and links in the networks. For instance, [4] considers a transductive classification task on heterogeneous networks. In particular, the authors propose a graph-based regularization framework, called GNetMine, which models the link structure in arbitrary information networks. GNetMine considers each graph associated with each type of link separately and aims at preserving its consistency. However, in GNetMine, class labels are associated with heterogeneous sub-networks. This means that the set of possible class values is common among all the objects, independently of their type. Although such a characteristic may appear reasonable in many domains, it cannot model those (more general) situations in which different classification schemes should be defined for each type of object.

In [21], the authors proposed the algorithm HENPC, which is able to solve multi-type classification tasks on heterogeneous networks. In particular, it extracts possibly overlapping and hierarchically-organized heterogeneous clusters and exploits them for predictive purposes.

The method proposed in [3] combines ranking and classification tasks on the basis of the intuition that highly-ranked objects within a class should play more important roles in classification or, vice versa, that class membership information is important for determining a good ranking over a dataset. Accordingly, a ranking-based iterative classification framework, called RankClass, is proposed. At each iteration, a graph-based ranking model is built and, on the basis of the current ranking results, the graph structure is adjusted, so that weights of the links in the subnetwork, corresponding to each specific class, are strengthened, while weights of the links in the rest of the network are weakened. Although experiments show the advantages of combining ranking and classification, as in [4], a single classification scheme can be associated with all the types of object.

Recently, in [22] the authors proposed a collective classification approach which aims at classifying objects of the same type in a heterogeneous network, based on the concept of meta-path. A meta-path is a path, between two objects to be classified, consisting of a sequence of link types. This concept is used to effectively assign labels to a group of interconnected instances, by taking into account different meta-path-based dependencies. Classification is probabilistic and is based on feature values of the object to be classified, on meta-paths, on "relational features" associated with meta-paths, as well as on the labels associated with objects traversed in the meta-paths. In this way, the proposed model is able to capture the subtlety of different dependencies among instances, with respect to different meta-paths. Although similar to ours, this approach is specifically designed to classify objects of a single type, similarly to the classical classification problem in relational data mining.

In the latter context, many approaches have been proposed in the literature, which can be considered relevant competitors for the task considered in this paper and that we have already introduced in Section 1. For example, the system Mr-SBC [9], on which the method proposed in this paper is based, adopts the naïve Bayes classification method in the multi-relational setting. This system exploits: *i)* first-order classification rules for the computation of the posterior probability for each class; *ii)* both discrete and continuous attributes by applying a supervised discretization method; *iii)* knowledge on the data model (i.e., the database schema) during the generation of classification rules. More recently, in [23] the authors introduced the tool RelWEKA, which extends the WEKA toolkit with the multi-relational version of classical data mining algorithms (e.g., k-NN and SVMs).

2.2 Multi-target prediction

The construction of different (possibly related) prediction models from the same dataset has received particular attention in the area of Structured Output Prediction (SOP). Specifically, multi-target prediction (more classification than regression) is related to our research, since, in such a task, we have several target attributes associated to the same type of object, with a different domain for each target attribute. In multi-type prediction, the difference is that different target attributes do not necessarily belong to the same object type.

The simplest approach to multi-target prediction is to consider it as multiple single-target prediction tasks and then apply a standard predictive algorithm on each of the single-target tasks (i.e., construct local models). Within this approach, it is possible to use any classification/regression method to obtain the local predictive models and then combine their outputs to obtain the predictions for the multiple target variables. Alternatively, global methods predict the complete structure as a whole. The global methods have several advantages over the local methods. One of the most important advantages is that they exploit the dependencies that exist between the components of the structured output in the model learning phase, which can give better predictive performance. For example, in [8] the authors propose an ensemble (either based on bagging or random forests) of predictive clustering trees and show that multi-target classification outperforms its single-target counterpart.

This is true also for other predictive modeling tasks: multi-target regression and hierarchical multi-label classification. However, most of the works in the literature focus on multi-target regression (MTR) task rather than multi-target classification. Although MTR aims at predicting real values instead of discrete class values, the underlying concepts and methods are very similar to those applicable to multi-target classification. For this reason, in the following we give a brief overview of the works on the MTR task.

In [24], the authors consider two groups of methods for MTR: problem transformation and algorithm adaptation, which correspond to local and global methods for Structure Output Prediction. As for global methods, in [25] the authors extend the standard ridge regression to multivariate ridge regression, while in [26] the authors propose the Curds&Whey method, where relations among the tasks are modeled in a post-processing phase. More recently, some authors have investigated kernel/SVM-based methods for MTR. For example, in [27] the authors extend the kernel

methods to the multi-target setting, using a particular type of kernel. In [28], the authors propose an approach to define the loss functions on the output manifold by considering it as a Riemannian submanifold, in order to include its geometric structure in the learning (regression) process. This approach can be used with any regression algorithm.

A different approach is the adaptation of methods for Multi Label Classification (MLC) towards the task of MTR. In [29], the authors present an ensemble method that constructs new target variables via random linear combinations of existing targets. The augmented output space is then exploited by adapting the MLC algorithm RAkEL for MTR.

Due to its background on naïve Bayes approaches, the method proposed in this paper has also some aspects in common with works that adopt general graphical models. An example can be found in [30], where the authors propose the "double dependent-variable factor graph" to predict simultaneously the age and the gender of users (i.e., features of a single node type) from a large network representing mobile communication activities. Another example can be found in [31], where the authors aim at inferring node labels in a partially labeled homogeneous network, where each node has multiple label types, each with a large number of possible values. In particular, the authors propose the method EDGEEXPLAIN, which is able to represent and catch interactions between properties of label types.

Focusing on the ensemble-based approaches, in [32] the authors propose the adoption of two methods, that are: stacked single-target regression and ensemble of regressor chains. The former corresponds to the binary relevance approach with the addition of meta-models that exploit the estimated values of the other target variables. The latter corresponds to the classifier chains method for MLC [33], which selects a random chain (permutation) of the target variables and builds a predictive model for each target by considering the predictions of the targets earlier in the chain. The ensemble is constructed by multiple random selections of the chains. This last approach also inspired the iterative framework that we present in Section 4.

3 Problem Statement and Background

Before describing the proposed method, in the following we introduce the notation used. In particular, as stated in the introduction, we work on heterogeneous networks, which we formally define as $G = (V, E)$, where V is the set of nodes and E is the set of edges among nodes. Both nodes and edges can be of different types. Moreover:

- Each node type T_p implicitly defines a subset of nodes $V_p \subseteq V$.
- Each node $v' \in V$ is associated with a node type $t_v(v') \in \mathcal{T}$, where $\mathcal{T}$ is the finite set $\{T_p\}$ of all the possible types of nodes in the network.
- A node type T_p defines a set of attributes $\mathcal{X}_p = \{X_{p,1}, X_{p,2}, \ldots, X_{p,m_p}\}$.
- An edge type R_j defines a subset of edges $E_j \subseteq (V_p \times V_q) \subseteq E$, where V_p and V_q are not necessarily based on different types.
- An edge e between two nodes v' and v'' is associated with an edge type $R_j \in \mathcal{R}$, where $\mathcal{R}$ is the finite set $\{R_j\}$ of all the possible edge types in the network. Formally, $e = \langle R_j, \langle v', v'' \rangle \rangle \in E$, where $R_j = t_e(e) \in \mathcal{R}$ is its edge type.

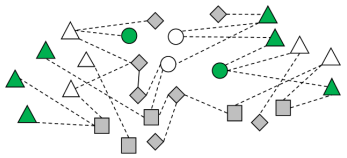

Fig. 1. An example of a heterogeneous network. The shape indicates the type of nodes: triangles and circles are nodes of target types (primary or secondary), and can be labeled (green) or unlabeled (white). Gray nodes belong to task-relevant types.

In the considered task, we define a role for each node type. In particular, we partition the set of node types $\mathcal{T}$ into:

- $\mathcal{T}_t$ (primary targets), which are considered as the targets of the main classification task;
- $\mathcal{T}_{st}$ (secondary targets), which are strongly related to the main classification task, for which a prediction of missing values is considered relevant;
- $\mathcal{T}_{tr}$ (task-relevant), which are the other node types.

An example of a heterogeneous network having nodes of different types with different roles is reported in Figure 1.

Only nodes of target (primary and secondary) types are actually classified, on the basis of all the nodes. However, we are actually interested in the maximization of the prediction accuracy only of objects of the primary target types.

The method we propose iteratively builds an ensemble of Mr-SBC [9] classifiers. For this reason, in the following subsection we report some details about this system.

3.1 The system Mr-SBC

Mr-SBC is a naïve Bayes classifier which is able to work on data stored in a relational database. It relies on a set of first-order rules induced from data stored in the tables belonging to the relational schema. Consequently, it can analyze heterogeneous networks of arbitrary structure, by representing them in a relational database, where:

- each table corresponds to a node type in the network;
- each foreign key constraint represents an edge type in the network;
- each tuple represents a node in the network;
- the attributes of each table represent the attributes associated with each node type.

More formally, let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ be the set of tables for Mr-SBC and let $T_t \in \mathcal{T}_t$ be the considered target type (in multi-type classification we can have several target types). An instance v', such that $t_v(v') = T_t$, is represented as a tuple in the target table according to its attributes $\mathcal{X}_t = \{X_{t,1}, X_{t,2}, \ldots, X_{t,m_t}\}$, joined with all the tuples which are related to v' following a foreign key path. A foreign key path is defined as an ordered sequence of tables $(T_{i_1}, T_{i_2}, \ldots, T_{i_s})$, where: $\forall_{j=1,\ldots,s}\ T_{i_j} \in \mathcal{T}$ and $\forall_{j=1,\ldots,s-1}\ T_{i_{j+1}}$ has a foreign key to the table T_{i_j}. A formal definition of the learning problem solved by Mr-SBC is:

Given:

- A labeled network G represented by means of n relational tables $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ of a relational database D, built from the network G;

- A set of primary key constraints on tables in $\mathcal{T}$;
- A set of foreign key constraints on tables in $\mathcal{T}$;
- A target table $T_t \in \mathcal{T}_t$;
- A target discrete attribute y belonging to T_t, different from the primary key of T_t, with values in $\mathcal{Y}_t$.

Find: a set of first-order rules R_D from the database D and build a naïve Bayesian classifier $\psi_t : V_t \rightarrow \mathcal{Y}_t$, which is able to classify all the unlabeled tuples in T_t, according to R_D.

It is noteworthy that Mr-SBC is not able to classify multiple types of objects, therefore the set of primary target types is limited to only one type, and the concept of secondary target types is not considered (thus not exploited). Another relevant limitation of Mr-SBC is that, in its original version, it explores paths where each foreign key is considered only once. This constraint is known in the literature as the "acyclicity constraint". However, as observed in [34], the acyclicity constraint hinders the representation of many important relational dependencies and, in particular, affects the possibility to capture some relevant autocorrelation phenomena. A simple example can be found in the bibliographic data, where multiple authors collaborate in writing a paper. By considering the table "authors" as the target table, the algorithm would not be able to exploit co-authorship relationships, since it would not be able to consider the path "authors" - "papers" - "authors", following the same foreign key "write". To overcome this limitation, we modified Mr-SBC, in order to also take into account cyclic paths.

In the following, we report some details about the main steps of Mr-SBC, that are the generation of rules and their exploitation for classification.

3.1.1 Generation of first-order rules

In the literature, we can find several studies on first-order naïve Bayes classifiers. In particular, in [35], the authors proposed a method based on a two-stepped process. The first step uses the ILP-R system [36] to learn a hypothesis in the form of a set of first-order rules and then, in the second step, the rules are probabilistically analyzed. During the classification phase, the conditional probability distributions of individual rules are combined naïvely, according to the naïve Bayesian formula. It is noteworthy that the approach adopted by the system ILP-R is very expensive and does not take into account the bias automatically determined by the constraints in the database. In [37], the authors proposed a similar two-stepped method. In this case, there is no learning of first-order rules in the first step. On the contrary, the method generates a set of patterns (first-order conditions) that are used afterwards as attributes in a classical attribute-value naïve Bayesian classifier. This method distinguishes between *structural predicates*, referring to parts of objects (e.g., authors of a paper), and *properties* applying to the objects or to one or several of its parts (e.g., the title of a paper). An elementary first-order feature consists of zero or more structural predicates and one property.

The solution adopted in Mr-SBC is similar to that proposed in [37], since the structure of classification rules is determined on the basis of the structure of the objects. The main difference is in the use of the rules, since Mr-SBC considers the contribution of every predicate only once in the computation of the probabilities, even if they are common to many rules (factorization of atoms rules [38]).

The predicates in the classification rules generated by Mr-SBC are binary and can be of two different types:

- *structural predicates*, which are associated to a table T_i if a foreign key in T_i exists that references a table T_j. The first argument of the predicate represents the primary key of T_j, while the second argument represents the primary key of T_i.
- *property predicates*, which are associated to a table T_i. The first argument of the predicate represents the primary key of T_i, while the second argument represents another attribute in T_i which is neither the primary key of T_i nor a foreign key in T_i.

A first-order classification rule associated to the foreign key path is a clause in the form:

$$p_0(A_1, y) \text{ :- } p_1(A_1, A_2), \ldots, p_{s-1}(A_{s-1}, A_s), p_s(A_s, c),$$

where: p_0 is a property predicate associated to the target table and to the target attribute y; $(T_{i_1}, T_{i_2}, ..., T_{i_s})$ is a foreign key path such that, for each $k = 1, ..., s-1$, p_k is a structural predicate associated to the table T_{i_k}; and p_s is a property predicate associated to the table T_{i_s}.
Rules are generated by means of a breadth-first strategy which starts from the target table. Generated rules are then refined by exploiting possible foreign key paths, that is, each refining step is performed only if the generated first-order classification rule can be associated with a foreign key path. Moreover, the number of refinement steps is upper bounded by a user-defined parameter $maxlen$. An example of a rule associated with a foreign key path is the following:
$business\_type(A, Restaurants)$:- $location(A, B)$, $location\_type(B, city), provides(A, C), goods\_type(C, food)$

3.1.2 Computation of Probabilities

Let R_D be the set of first-order classification rules for all the classes $\mathcal{Y}_t$, and v' be an object of the current target type to be classified. The label to assign to v' is computed by exploiting Bayes theorem, according to the following equation:

$$\psi_t(v') = argmax_c P(Y_c|R_{v'}) = argmax_c \frac{P(Y_c) \cdot P(R_{v'}|Y_c)}{P(R_{v'})}, \quad (1)$$

where:

- $Y_c \in \mathcal{Y}_t$ is a possible class value of the attribute y;
- $R_{v'} \subseteq R_D$ is the subset of first-order rules covering the object v', that is: $R_{v'} = \{r_i \in R_D | r_i \text{ covers } v'\}$.

According to the naïve Bayes assumption, Mr-SBC considers the attributes independent. This assumption is clearly violated for the attributes that are primary keys or foreign keys. This means that the computation of $P(R_{v'}|Y_c)$ in Equation (1) depends on the structures of rules in $R_{v'}$. For instance, assume that two rules R_1 and R_2, related to class Y_c, share the same structure and differ only in the property predicates in their bodies:

$$R1: \ \beta_{1,0} \text{ :- } \beta_{1,1}, ..., \beta_{1,K_1-1}, \beta_{1,K_1}, \quad R2: \ \beta_{2,0} \text{ :- } \beta_{2,1}, ..., \beta_{2,K_1-1}, \beta_{2,K_2},$$

where $K_1 = K_2$, $\forall_{i=1,k_1-1} \beta_{1,i} = \beta_{2,i}$ and $\beta_{0,1} = \beta_{0,2} = Y_c$. Then:

$$P(\beta_{1,K_1} \wedge \beta_{2,K_2} | \beta_{1,0} \wedge \beta_{2,0} \wedge (\beta_{1,1}, ..., \beta_{1,K_1-1}) \wedge Y_c) =$$

$P(\beta_{1,K_1}|(\beta_{1,1},...,\beta_{1,K_1-1}) \wedge Y_c) \cdot P(\beta_{2,K_2}|(\beta_{1,1},...,\beta_{1,K_1-1}) \wedge Y_c)$. (2)

Contrary to [35] and [37], following this approach, Mr-SBC is able to avoid multiple computations of the probability of the structure. In particular, we can compute the numerator of Equation (1) by generalizing the described approach to the set of classification rules $R_{v'}$ as follows:

$$P(Y_c) \cdot P(R_{v'}|Y_c) = P(Y_c) \cdot P(struct) \cdot \prod_{R_j \in R_{v'}} P(R_j|struct), \quad (3)$$

where the term *struct* takes into account the class Y_c and the structure of the rules in $R_{v'}$.

All the factors used in Equation (3) and its expansion (see Equations (4) and (6)) represent the naïve Bayesian model for each class label Y_c and for a single target type. The way these terms are used and estimated in the approach presented in this paper depends on the ensemble learning strategy adopted (and discussed in Section 4).

Computation of P(R_j| struct). If a classification rule $R_j \in R_{v'}$ is in the form $\beta_{j,0}$:- $\beta_{j,1},...,\beta_{j,K_j-1},\beta_{j,K_j}$, where $\beta_{j,0}$ and β_{j,K_j} are property predicates and $\beta_{j,1},\beta_{j,2},...,\beta_{j,K_j-1}$ are structural predicates, then we can compute the term $P(R_j|struct)$ of Equation (3) as follows:

$$\begin{aligned} P(R_j|struct) &= P(\beta_{j,K_j}|\beta_{j,0},\beta_{j,1},...,\beta_{j,K_j-1}) \\ &= P(\beta_{j,K_j}|Y_c,\beta_{j,1},...,\beta_{j,K_j-1}), \end{aligned} \quad (4)$$

where Y_c is the value of the target attribute in the head of the clause ($\beta_{j,0}$). In order to avoid null probabilities in Equation (3), Mr-SBC exploits the Laplace estimation:

$$P(\beta_{j,K_j}|Y_c,\beta_{j,1},...,\beta_{i,K_j-1}) = \frac{\#(\beta_{j,K_j},Y_c,\beta_{j,1},...,\beta_{j,K_j-1})+1}{\#(Y_c,\beta_{j,1},...,\beta_{j,K_j-1})+F}, \quad (5)$$

where F is the number of possible values of the attribute the β_{j,K_j} property predicate is associated with. It is noteworthy that the numerator of Equation (5) is the number of tuples covered by the rule $\beta_{j,0}$:- $\beta_{j,1},...,\beta_{j,K_j-1},\beta_{j,K_j}$, which can be efficiently computed by a "select count (*)" SQL instruction. The value at the denominator is the number of tuples covered by the rule $\beta_{j,0}$:- $\beta_{j,1},...,\beta_{j,K_j-1}$.

Computation of P(struct). Let B=$\{(\beta_{j,1},\beta_{j,2},\ldots,\beta_{j,t})|$ j=1, 2, $\ldots s$ and t=1, ...,$K_j-1\}$ the set of all the distinct sequences of structural predicates in the rules of $R_{v'}$. Then:

$$P(struct) = \prod_{seq \in \mathrm{B}} P(seq). \quad (6)$$

To compute $P(seq)$, Mr-SBC exploits the definition of the probability JP that a join query is satisfied [39]. Let $(T_{i_1},T_{i_2},\ldots,T_{i_s})$ be a foreign key path, then:

$$JP(T_{i_1},T_{i_2},\ldots,T_{i_s}) = \frac{|\triangleright\triangleleft(T_{i_1} \times T_{i_2} ... \times T_{i_s})|}{|T_{i_1}| \times |T_{i_2}| \times ... \times |T_{i_s}|},$$

where $\triangleright\triangleleft (T_{i_1} \times T_{i_2} \times \ldots \times T_{i_s})$ is the result of the join between the tables $T_{i_1},T_{i_2},\ldots,T_{i_s}$.

It is noteworthy that each sequence $seq = (\beta_{j,1},\beta_{j,2},\ldots,\beta_{j,t})$ is associated with a foreign key path. $P(seq)$ can be recursively computed by observing that there could be a prefix of *seq* in B. In particular, by denoting the table related to $\beta_{j,h}$ as T_{j_h} ($h=1,2,\ldots,t$), the probability $P(seq)$ is computed as:

$$P(seq) = \begin{cases} JP(T_{j_1},T_{j_2},...,T_{j_t}) & \text{if } seq \text{ has no prefix in B} \\ \frac{JP(T_{j_1},T_{j_2},...,T_{j_t})}{P(seq')} & \text{if } seq' \text{ is the longest prefix of } seq \text{ in B} \end{cases}$$

This formulation is necessary in order to compute Equation (6) considering both dependent and independent events. Since $P(struct)$ takes into account the class, $P(seq)$ is computed separately for each class.

4 The Proposed Ensemble Learning Method

In this section we describe the solution we propose to solve the considered classification task on data organized in a heterogeneous network. As already stated in Section 1, the proposed method is based on an ensemble of Mr-SBC [9] classifiers, which is able to take into account additional aspects of the network data: *i)* the dependencies among different attributes of possibly different nodes (network correlation); *ii)* the existence of dependencies among values of the same attribute of different nodes (network autocorrelation); *iii)* the existence of dependencies between the class label of the nodes which are subject of the main classification task and attributes belonging to nodes of other types, whose value is unknown. In particular, we propose two different variants: ST-MrSBC, which exploits self-training techniques in order to capture the aspects *i)* and *ii)*, and MT-MrSBC, which also classifies objects of other types, that are not the subject of the main classification task. Therefore, this solution captures all the considered aspects *i)*, *ii)* and *iii)*.

These two solutions share several choices (see the iterative high-level description of the algorithm in Figure 2). Both ST-MrSBC and MT-MrSBC take as input a partially labeled heterogeneous network and work iteratively. At each iteration, they build an ensemble of classifiers from different subsets of labeled nodes (either known or predicted in the previous iterations), whose combination of the output will possibly lead to a stronger prediction model.

In the case of MT-MrSBC, following the idea adopted in [33] (for multi-label classification tasks), we shuffle all the primary and secondary target types. MT-MrSBC then performs a sampling of a subset of nodes of the first target type (which can be either primary or secondary) and builds a weak[1] predictive model through Mr-SBC. This predictive model is applied to classify unlabeled nodes which are then added to the labeled network. The obtained probabilities are also stored for the final combination of the outputs (see the last step of the algorithm in Figure 2). Then, according to the (random) ordering defined over the target types, we select a new target type and repeat the process. It is noteworthy that, at this stage, predictions performed for the previous target types are available to Mr-SBC when building a prediction model for the new target. Target types are shuffled again every time they are all processed. The number of iterations is limited by a user-defined threshold and the outputs obtained for each target type over all the iterations are combined to obtain the final (strong) predictive model.

ST-MrSBC uses a similar process, but works on a single target type. Therefore, it exploits predictions obtained on the

1. It is considered a weak prediction model, since built from a reduced subset of labeled instances.

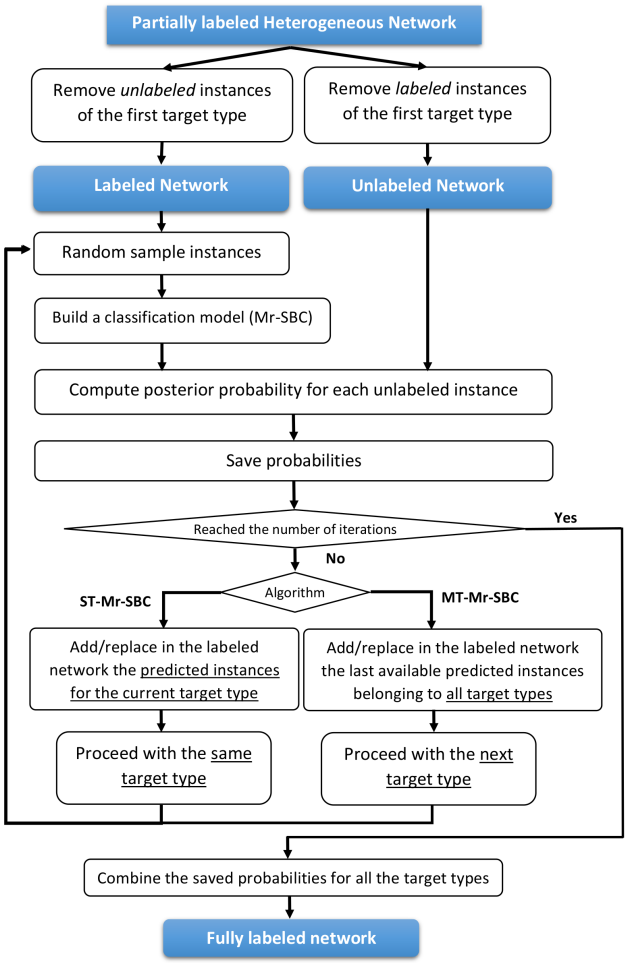

Fig. 2. High-level description of ST-MrSBC and MT-MrSBC.

same target type in the previous iterations when building a prediction model for new iterations. The main difference with respect to the standard self-training method is in the construction of the labeled network when a new iteration starts. Specifically, ST-MrSBC actually builds an ensemble of weak classifiers from different subsets of labeled nodes, instead of considering only the output of the last iteration.

4.1 Definition of target types

As already discussed, at each iteration, the algorithm can work on different target attributes (and types). In ST-MrSBC, this property is not exploited since it works on a single target type, which represents the subject of the classification task. Contrarily, MT-MrSBC needs the set of primary target types $\mathcal{T}_t$ and the set of secondary target types $\mathcal{T}_{st}$. We recall that primary target types are those for which we are interested in maximizing the classification accuracy of their target attribute, since they are the subject of the main classification task. Objects belonging to secondary target types are those that still have a target attribute and, if predicted, this attribute could possibly lead to an improvement in the classification of objects of the primary target types.

Once these two sets are defined, we build the list L_t which contains the union of primary and secondary target types. This list represents the order according to which, in MT-MrSBC, target attributes (and types) are processed. The order of target types in L_t is defined randomly. The choice of a random ordering is motivated by the fact that a predefined ordering of the analysis of target types can negatively affect the classification accuracy, since a wrong decision can inhibit the exploitation of relevant dependencies and, consequently, can possibly enforce the exploitation of irrelevant/wrong dependencies. In the literature, this phenomenon is also observed in random-scan Gibbs sampling, as opposed to systematic-scan Gibbs sampling [40]. The random-scan Gibbs sampling is a Markov chain Monte Carlo method, which considers a random order of variables when performing sampling by exploiting the conditional distribution of variables. It has been proved that the order of the analysis of the variables can affect the effectiveness and the convergence speed of Gibbs sampling and that a random-scan can outperform a systematic scan in terms of convergence speed (number of iterations necessary to approximate the desired probability distribution) [41]. The idea of randomly selecting variables has been also exploited for multi-label classification. Specifically, in [33] the authors propose the analysis of the set of target variables in different (random) orders at each iteration.

We expect that a fixed ordering could lead to very good results when the best ordering is selected, whereas could lead to a decrease in the accuracy when a wrong ordering is considered. In other words, we expect some instability issues in the accuracy results over different datasets. On the contrary, we expect that the randomization of the target types that we adopt should lead to more stable results over different datasets. In order to evaluate the possible influence of the order of target types on the results obtained by the proposed method, in the experiments, we will consider both fixed and random ordering.

4.2 Construction of labeled and unlabeled networks

At each iteration, the algorithm considers the target attribute of the next type in L_t. According to the selected target type, it builds two separate networks: the first contains only labeled nodes, and the second contains only unlabeled nodes. These networks are built by considering the complete network and by removing unlabeled and labeled nodes, respectively. This means that all the nodes of other target types, as well as nodes of task-relevant types, are included in both networks. While the network of labeled nodes, for each target type, changes at different iterations, the network of unlabeled nodes remains stable and the algorithm classifies the same nodes several times.

The way nodes are considered as training instances at each iteration is random. In fact, the algorithm randomly selects, according to a uniform distribution, a given percentage $perc$ of labeled nodes from the labeled network. It is noteworthy that, at a given iteration, the labeled network could contain also nodes that were initially unlabeled but that were classified during a previous iteration. In the case of ST-MrSBC, the last available predictions will regard the same target type, whereas in the case of MT-MrSBC, the last available predictions will regard all the target types (primary and secondary). This behavior is coherent with the assumption that predictions performed on other target

types (primary or secondary) can help in the predictions of the label of nodes of the current target type.

This implies that, at the next iteration, the algorithm will be able to take into account some labels predicted at the previous iterations for the same target type (MT-MrSBC and ST-MrSBC) and all the labels predicted at the previous iterations for a different target type (MT-MrSBC only).

This choice is, apparently, in contrast with most of the works which adopt the self-training framework, where the idea is often to select the most reliable predictions for the next iterations. However, *1)* this solution does not necessarily lead to better results with respect to a random selection [42], and *2)* we do not use predictions as "hard" constraints, but in the next iterations we are able to retract decisions that are not coherent with the new state of the network.

4.3 Estimation of probabilities

At the end of each iteration, it is possible to build a classification model through Mr-SBC (i.e., its variant) for the current target type from the network of labeled nodes. For each unlabeled node, this classification model is exploited to compute the posterior probability (which takes into account both network correlation and network autocorrelation) and its label, according to Equation (1). These labels are then propagated into the labeled network for the subsequent iterations, as described in the previous subsection.

The probabilities are saved and exploited at the end of the entire process. In fact, after the last iteration, the final classifier combines the probabilities computed during all the iterations. In particular, for each target type T_t and for each node v', the method computes the final probability as the average of the probabilities computed over the ensemble. Formally, we extend Equation (1) in the following way:

$$\begin{aligned} \psi_t(v') &= argmax_c \frac{1}{z} \sum_{k=1}^{z} P(Y_c|R_{v'k}) = \\ &= argmax_c \frac{1}{z} \sum_{k=1}^{z} \frac{P(Y_c)P(R_{v'k}|Y_c)}{P(R_{v'k})}, \end{aligned} \quad (7)$$

where z is the number of iterations, i.e., the number of classifiers in the ensemble, for each target type (the total number of iterations is $z \cdot |L_t|$) and $R_{v'k}$ is the set of rules identified from the labeled network at the k-th iteration. Since $P(R_{v'k})$ is independent of the class Y_c, the final classification of an unlabeled node is computed as:

$$\psi_t(v') = argmax_c \frac{1}{z} \sum_{k=1}^{z} P(Y_c)P(R_{v'k}|Y_c). \quad (8)$$

The rationale behind the combination introduced in Equation (8) (and Equation (7)) is twofold: *a)* the predictions obtained at each iteration are based on different training sets, which may focus on different properties of the concept to be learned; *b)* the predictions are made more stable. We will further discuss both aspects in the following subsection.

4.4 Theoretical motivation

The approach we follow is based on the combination of two ensemble learning approaches: bagging and boosting. While bagging produces several training sets by performing a sampling with replacement in the training set, boosting uses all instances at each repetition but maintains a weight for each instance in the training set that reflects its importance [43]. In fact, the principal difference between bagging and boosting [44], is that the latter trains base classifiers in sequence, and each base classifier is trained using a weighted form of the data set in which the weighting coefficient, associated with each data point, depends on the performance of the previous classifiers. Once all the classifiers have been trained, their predictions are then combined through a weighted majority voting scheme.

As observed in [45], boosting is considered stronger than bagging on noise-free data, whereas bagging is much more robust than boosting in noisy settings. Moreover, Breiman [46] showed that bagging can give notable gains in predictive performance, when applied to unstable learners (for which small changes in the training set result in large changes in the predictions[2]). These two observations, together with recent results obtained in [47], where the authors proposed naïve Bayes classifiers in the self-training framework, motivated our choice of combining bagging and boosting in the same ensemble learning algorithm.

In fact, our approach is essentially bagging-based, but it can be considered mixed because we produce several training sets in sequence as in approaches based on boosting. In this way, in principle, we are able to improve the performance of new classifiers that may depend on the predictions of previous classifiers (also on different target types) and, in addition, make predictions more stable [46].

An additional motivation for the solution we propose comes from a practical consideration: predictions obtained at each iteration are based on different training sets. Each training set may focus on different target attributes or on different properties of the concept to be learned. Consequently, we are naturally able to deal with the multi-type classification task, also by boosting predictions of objects of different types.

5 Time Complexity

In order to evaluate the time complexity of the proposed approach, we first analyze the complexity of Mr-SBC. Let:

- k be the average number of types of objects that are related to each type of object (i.e., the average number of foreign key constraints for each table);
- m be the average number of attributes per node;
- n be the average number of nodes of each type (i.e., the average number of tuples in each table);
- q be the average number of distinct values of an attribute;
- $maxlen$ be the user-defined threshold on the length of classification rules.

In the computation of $P(struct)$, in the worst case (i.e., when all the intermediate tables have no attributes and structural intermediate probabilities have not already been computed), the time complexity is:

- 0 for the target table;

2. This can also happen in naïve Bayes classifiers when posterior probabilities for different labels are very similar.

- $O(k \times join\_complexity)$ for tables at distance 1 from the target table; ...
- $O(k^p \times join\_complexity)$ for tables at distance p.

By exploiting index structures on the attributes of both primary and foreign keys, a join among p tables is computed in time $O(p \cdot n)$. Therefore, the time complexity is:

$$\sum_{p=1}^{maxlen} O(k^p \cdot (p+1) \cdot n). \quad (9)$$

This means that the complexity of computing $P(struct)$ is:

$$O\left(k^{maxlen} \cdot (maxlen+1) \cdot n\right) = O\left(k^{maxlen} \cdot n\right). \quad (10)$$

The complexity of the computation of $\prod_j P(R_j|struct)$ is:

- $O(m \cdot q)$ for the target table;
- $O(m \cdot q \cdot k)$ for tables at distance 1 from the target table; ...
- $O(m \cdot q \cdot k^p)$ for tables at distance p.

Therefore, the time complexity for $\prod_j P(R_j|struct)$ is:

$$O(m \cdot q \cdot k^{maxlen}). \quad (11)$$

By combining Equations (10) and (11), we have that the complexity of Mr-SBC is:

$$O(m \cdot q \cdot k^{maxlen} \cdot n). \quad (12)$$

It is noteworthy that k (the average number of tables related to each table[3]) is usually very low $(0-3)$ and that $maxlen$ can be reasonably chosen, in order to obtain a good trade-off between the length of the explored paths and the complexity of the analysis. Therefore, we can conclude that the time complexity of Mr-SBC is generally linear with respect to the number of objects, the number of attributes and the number of distinct values per attribute.
In our ensemble learning solution, we have:

- z is the number of iterations for each target type;
- $|L_t|$ is the number of target types;
- $perc$ is the percentage of nodes for each sample.

The variant ST-MrSBC performs z iterations on each target type independently, on a subset of $perc \cdot n$ nodes, leading to $z \cdot |L_t|$ runs of Mr-SBC. Analogously, MT-MrSBC builds a classification model for each target type in L_t for each iteration, leading to the same number of runs of ST-MrSBC ($z \cdot |L_t|$). Therefore, according to Equation (12), the time complexity of both ST-MrSBC and MT-MrSBC is:

$$O(z \cdot |L_t| \cdot m \cdot q \cdot k^{maxlen} \cdot (perc \cdot n)). \quad (13)$$

Since $perc \cdot n \ll n$, and z and $|L_t|$ are constants with relatively small values, we can conclude that the proposed algorithm does not affect the time complexity, which is similar to that of Mr-SBC.

6 Experiments

In this section, we first describe the considered datasets and the experimental setting. Then, we show the obtained results, discuss them and draw some conclusions.

3. The main difference with respect to the original Mr-SBC is that here, on average, k includes already visited tables.

6.1 Datasets used in the experiments

MOVIE. This dataset is built from MovieLens100k dataset[4] and contains ratings from 1,000 users on 1,700 movies, which are collected by the *movielens* recommender system. We processed the dataset in order to create a network consisting of 4 node types and 3 edge types. The main classification task focuses on movies, that is, $\mathcal{T}_t = \{movies\}$, which are classified into: *comedy*, *thriller*, *drama* and *action*. We considered the users as a secondary target, focusing on their occupation, since this aspect could be strongly related to their preferences of movies. Therefore, $\mathcal{T}_{st} = \{users\}$. Overall, the network contains 2,051 nodes and 59,532 edges.

NBA. This dataset has been released by the Carnegie Mellon University[5] and contains the statistics of basketball players of NBA and ABA, collected during the years 2004 and 2005. In particular, it contains statistics from regular season, playoff and all star games and data related to players, coaches and drafts. The main classification task focuses on teams, that is $\mathcal{T}_t = \{teams\}$, which are classified into: *national* and *american*. We considered the players as a secondary target, i.e. $\mathcal{T}_{st} = \{players\}$, focusing on their role. The network contains 36,593 nodes and 63,924 edges.

YELP. Yelp is a website where it is possible to find, review and talk about business activities. The dataset has been released for academic purposes[6] to supply real-world data for the experimental evaluation of machine learning methods. The data include: businesses, business characteristics, users, check-in information, friendship relationships, tips and reviews. We processed the dataset in order to create a network consisting of 7 node types and 8 edge types. In this dataset, we considered two primary targets, i.e., $\mathcal{T}_t = \{business, users\}$, since we have the ground truth for both of them. The businesses are classified into four categories: *Restaurants*, *Beauty & Spas*, *Health & Medical* and *Shopping*, whereas the users are classified into: *low* and *high*, representing the average rating they gave to the businesses. The network contains 1,387,596 nodes and 1,371,060 edges.

IMDB. This dataset[7] is an extension of the MovieLens10M dataset, published by the GroupLens research group. In particular, it integrates data from the Movielens dataset with data about the pages from Internet Movie Database (IMDb) and reviews from Rotten Tomatoes. We kept only the users with both rating and tagging information. Moreover, we processed the dataset in order to create a network consisting of 13 node types and 16 edge types. The classification task focuses on movies, i.e., $\mathcal{T}_t = \{movies\}$, which are classified into: *comedy*, *thriller*, *drama* and *action*, while, as in the dataset MOVIE, we consider the users as a secondary target, i.e. $\mathcal{T}_{st} = \{users\}$. The network contains 212,039 nodes and 342,161 edges.

STACK. This dataset[8] is an anonymized dump of user-contributed content on the Stack Exchange network. We selected data coming from the Stack Overflow web-site which include posts, users, votes, comments, history and links. In this dataset, the main classification task focuses on posts,

4. http://grouplens.org/datasets/movielens/
5. http://www.cs.cmu.edu/ awm/10701/project/data.html
6. http://www.yelp.com/dataset_challenge
7. http://grouplens.org/datasets/hetrec-2011/
8. https://archive.org/details/stackexchange

i.e., $\mathcal{T}_t = \{posts\}$, which are classified into five categories: *1* (Questions), *2* (Answers), *3* (Wiki), *4* (TagWikiExcerpt) and *5* (TagWiki). Since the users' reputation can be considered strongly related to the type of posts they make (e.g., an expert usually posts answers rather than questions), we considered the users as a secondary target, i.e., $\mathcal{T}_{st} = \{users\}$, focusing on their reputation. The network contains 92,800 nodes and 114,385 edges.

6.2 Experimental setting

The experimental questions we want to answer are:

1) What is the contribution of the different variants of the ensemble learning solution we propose?
2) Is the variant MT-MrSBC able to capture possible (network) correlation of labels of different types?
3) How does the proposed method compare with competitor solutions?
4) Is the ensemble learning solution able to produce more stable predictions?

As concerns the evaluation of the different variants of the ensemble learning solutions we propose, we compared them with the original version of Mr-SBC. This allows us to evaluate the contribution of each aspect we take into account in our method, i.e., the iterative nature of the ensemble-based self-training approach (introduced in **ST-MrSBC**) and the classification of objects of multiple types, both primary and secondary targets (implemented in **MT-MrSBC**). Moreover, we evaluate the performance of MT-MrSBC by considering both the strategies for ordering the target types introduced in Section 4. We call these two variants **LexicographicMT-MrSBC** and **RandomMT-MrSBC**, respectively.

As for the second question, we considered all the known links between objects belonging to target types as a new ground truth and computed the accuracy of the pair-wise prediction of labels. We considered only the first iteration of the framework, in order to exclude the possible influence of the ensemble approach. This allows us to further assess the possible contribution of the multi-type approach and to evaluate whether it really captures, if any, network correlation of labels of different types.

In order to perform a comparison with other systems, we also ran the experiments with four competitor methods. In particular, we considered *i)* the relational version of the nearest neighbour algorithm (**RelIBK**), *ii)* the SVM-based algorithm SMO (**RelSMO**), *iii)* the algorithm **GNetMine** [4], which is natively able to work on heterogeneous networks, and *iv)* the algorithm **HENPC** [21], recently proposed to solve the multi-type classification task in heterogeneous networks. However, RelIBK, RelSMO and HENPC were not able to finish within 3 days of execution, while GNetMine was not able to compute the results for Yelp dataset, since the system went out of memory (on a server with 32GB of RAM). In these cases, we ran the experiments on a reduced set of nodes (about $1,000$ nodes for each target type) for the datasets IMDB, STACK and YELP, by adopting a stratified random sampling. It is noteworthy that, for the dataset YELP (which has two primary target types), ST-MrSBC, RelIBK, RelSMO and GNetMine were run twice in order to learn two different classification models.

Finally, we evaluated the stability of the models in terms of standard deviation of the accuracy, by varying the number of iterations. This comparison was performed among LexicographicMT-MrSBC, RandomMT-MrSBC, ST-MrSBC and Mr-SBC.

The evaluation was performed in terms of average and standard deviation of the accuracy over the results obtained by adopting the 10-fold cross validation strategy. Accuracies were measured on the primary target types. Finally, we performed a Friedman test with the Nemenyi post-hoc tests at p-value = 0.05, to evaluate the significance of the results from a statistical viewpoint. As regards the parameter setting of ST-MrSBC and MT-MrSBC, we considered a random sampling of 20% of nodes for each classifier in the ensemble and the number of iterations for each target type z ranging from 1 to 50. $maxlen$ is set to 999 in order to capture all possible interactions of any length in the network.

6.3 Results

The results obtained by Mr-SBC and the two variants ST-MrSBC and MT-MrSBC (with both the ordering strategies) are shown in Figure 3. They show how the accuracy varies with respect to the number of iterations. As can be observed, the proposed method (in all its variants) is always able to obtain better performance than Mr-SBC. The only exception is the target type *business* of the dataset Yelp, where LexicographicMT-MrSBC obtains the worst result among all the considered approaches. This is probably due to the fact that this approach forced the system to capture a dependency $business \rightarrow users$ which is probably weak, and did not allow the system to exploit a possible dependency $users \rightarrow business$. This observation is confirmed by the result obtained by RandomMT-MrSBC (actually the best for this dataset), which randomly selected either the dependency $business \rightarrow users$ or the dependency $users \rightarrow business$ at each iteration. Another relevant observation is that the improvement increases with the number of iterations, and that the optimal result is reached after only 5 iterations. This means that the proposed method rapidly converges to the best results in few iterations.

By comparing the two proposed variants (ST-MrSBC and MT-MrSBC), we can observe that, on most of the datasets, MT-MrSBC is generally able to obtain better results. This confirms our initial intuition about the exploitation of the predictions performed on related instances of different types. The only cases in which ST-MrSBC outperforms MT-MrSBC are on the datasets MOVIE and STACK. This is probably due to a weak relationship/dependency between the considered primary and secondary target types, whose (possibly wrong) exploitation affected the results. However, the difference in terms of accuracy appears very low. On the contrary, both LexicographicMr-SBC and RandomMr-SBC lead to a significant improvement with respect to ST-MrSBC on the target type *user* of the dataset Yelp and on the dataset NBA (where the advantage is more evident).

All these considerations are confirmed by the charts showing the result of the Nemenyi post-hoc tests performed on single datasets, which are depicted in Figure 5. In particular, we can observe that for the datasets NBA and for the target type *users* of the dataset Yelp, the improvement

Fig. 3. Results in terms of accuracy averaged over the folds of the 10-fold cross validation varying the number of iterations per target attribute.

provided by LexicographicMT-MrSBC and RandomMT-MrSBC over the competitors is statistically significant at p-value$= 0.05$. Focusing on LexicographicMT-MrSBC, we observe two specular and interesting cases: for the dataset IMDB it provides the best result, while for the target type *Business* of the dataset Yelp (as already described) it appears to be the worst approach. This instability is again caused by the static ordering on the target types. Indeed, in the first case, the exploitation of the dependency $movies \rightarrow users$ (which is surely strong, if we observe the result) led LexicographicMT-MrSBC to obtain a better result with respect to RandomMT-MrSBC, which alternatively (and randomly) exploited the dependencies $movies \rightarrow users$ and $users \rightarrow movies$ (which appears weak). On the contrary, in the second case, the static (unlucky) choice of the dependency to be exploited brought LexicographicMT-MrSBC to the bottom of the ranking.

In order to show an overview of the performance of the considered approaches, we also performed a statistical test which evaluates globally the performance of the algorithms over all the datasets. The results are summarized in the last chart of Figure 5, which shows that all the proposed approaches provide a statistically significant advantage over the standard Mr-SBC approach. Moreover, the high instability of LexicographicMT-MrSBC makes it statistically equivalent to ST-MrSBC, whereas RandomMT-MrSBC is (overall) significantly better than all the other approaches.

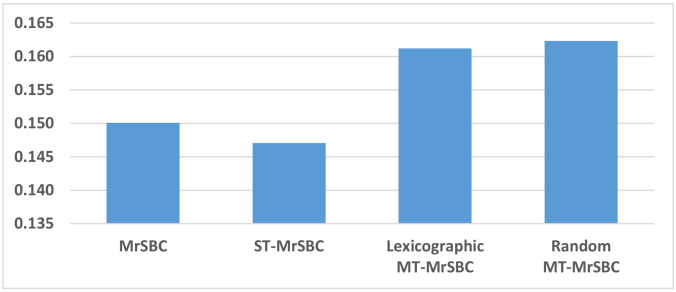

Fig. 4. Average accuracy in the pair-wise prediction of labels of directly linked objects of different types. The results reported are obtained at the first iteration for each target (to exclude the possible influence of the ensemble approach).

In order to further assess the contribution of the multi-type approach (*experimental question 2*), we computed the accuracy of the pair-wise prediction of labels of connected objects of different types. In other words, we evaluated how well the method is able to predict a pair of labels for a pair of linked objects. The average accuracy among all the considered datasets is reported in Figure 4. We can observe that the results of ST-MrSBC are worse than those obtained by Mr-SBC. This is clearly due to the fact that they adopt almost the same approach (since we focus only on the first iteration for each target) and ST-MrSBC only considers a sample

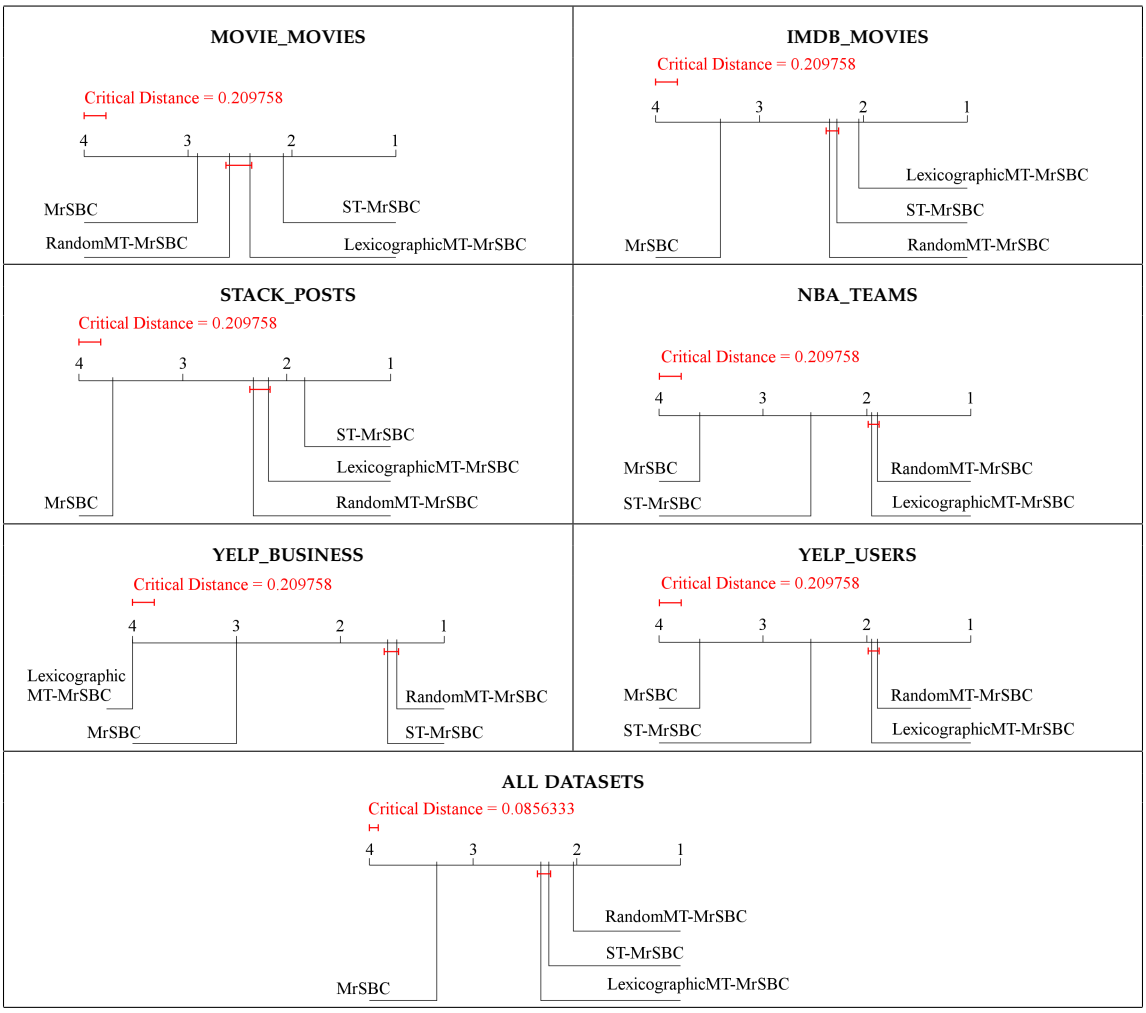

Fig. 5. Results of the Nemenyi post-hoc test on the average accuracy. Better algorithms are positioned on the right-hand side, and those that do not significantly differ in performance (at p-value $= 0.05$) are connected with a line.

of the training data. On the contrary, both the multi-type approaches (especially RandomMT-MrSBC) show a clear improvement over the single-type approaches (MrSBC and ST-MrSBC), which confirms that, if any, the method is able to actually capture label dependencies between multiple types of objects. We note that this behavior is not motivated by the contribution of the ensemble learning approach (which is excluded at the first iteration per target).

Next, we also compared the results obtained by the proposed approach with those of four competitors that can be used for the network classification task (*experimental question 3*): RelIBK, RelSMO, GNetMine and HENPC. The result of the statistical test, reported in Figure 6, shows that the improvement provided by the proposed method is able to give Mr-SBC the advantage of outperforming the competitors[9]. Indeed, the original version of Mr-SBC is not able to outperform the considered competitors, while the ensemble learning (ST-MrSBC) leads to outperform all the competitors (although not statistically w.r.t. HENPC and RelIBk). The clear advantage comes from the fruitful combination of capturing label dependencies between multiple types of objects and of the ensemble learning approach (MT-MrSBC), which significantly improves the accuracy.

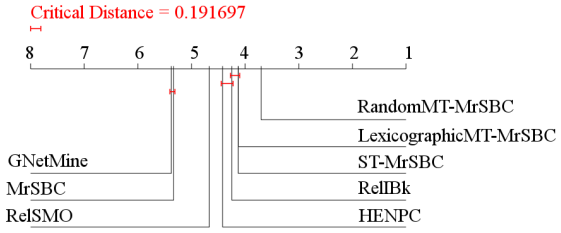

Fig. 6. Results of the Nemenyi Post-Hoc tests for the average accuracy over all the considered datasets. The accuracy values of RelIBK, RelSMO, GNetMine and HENPC, when necessary, are obtained from reduced-size datasets after a stratified random sampling.

Finally, we evaluate the stability of the models (*experimental question 4*). As can be seen in Figure 7, by increasing the number of iterations, the models become more "stable" (i.e. the standard deviation of their accuracy is smaller). While the differences are not statistically significant in the Friedman test (p-value =0.05), the ranking becomes clear

9. We are aware that the significance of this test is negatively affected by the execution of the competitors on smaller datasets (obtained with a stratified random sampling) but, as stated before, this was the only way we had to deal with the high complexity of these algorithms. Moreover, we have also to consider that our method is based on sampling.

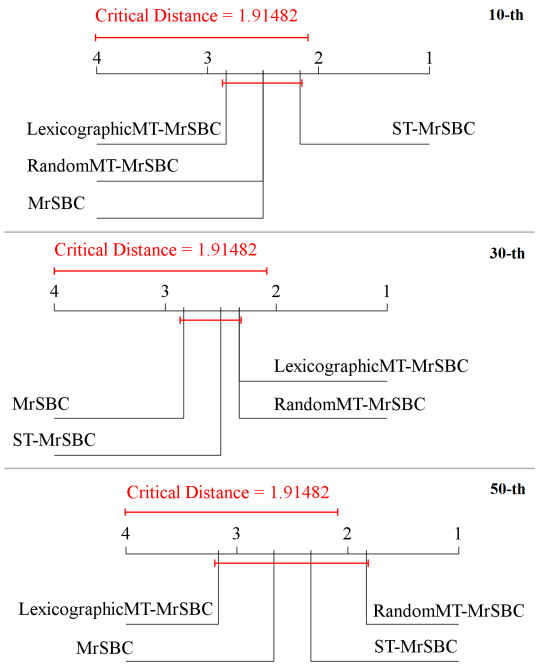

Fig. 7. Results of the Nemenyi Post-Hoc tests for the standard deviation at the *10-th*, *30-th* and *50-th* iterations for each target of the ensemble algorithm, respectively. Better algorithms (i.e. with smaller standard deviation) are positioned on the right-hand side.

after 30 iterations per target attribute: the most unstable predictor is LexicographicMT-MrSBC (ranked, on average, in position 3.3 out of 4). As expected, the best is RandomMT-MrSBC (ranked, on average, in position 1.8 out of 4). This empirically proves that the combination of capturing label dependencies between multiple types of objects and of the ensemble learning approach leads to smooth and stable predictions. The difference between RandomMT-MrSBC and ST-MrSBC (ranked, on average, in position 2.4 out of 4) is only due to the effect of the multi-type learning task.

Overall, we can conclude that the application of the proposed method, which is able to capture both correlation and autocorrelation phenomena, as well as to predict missing values, by exploiting the same classification method adopted for primary targets, can lead to better, more stable predictions when applied to real-world data organized in heterogeneous networks.

6.4 Availability

The proposed method, all the datasets used in the experiments and complete experimental results are publicly available at https://doi.org/10.6084/m9.figshare.4334048.v7

7 Conclusions

In this paper, we proposed an extension of the system Mr-SBC which works on heterogeneous networks and that is able to capture both correlation and autocorrelation phenomena. The presence of these phenomena is fruitfully exploited to improve the accuracy of the classification of nodes of some types, also by exploiting the classification of nodes of other types in the network. Experiments performed on real-world datasets show that both the proposed variants are able to significantly outperform the standard implementation of Mr-SBC, especially when a random ordering of the target types for each iteration is adopted. Moreover, they provide Mr-SBC with the right advantage to outperform four other well-known algorithms for the classification of data organized in a heterogeneous network.

As future work, we plan to perform a close analysis of the performance of the proposed method by considering different aspects that could (positively or negatively) affect its performance, such as the size of the sampling, the possibility to select only a subset of labeled nodes to propagate to the next iteration, or the adoption of a smarter combination strategy for the ensemble of predictions. Moreover, we plan to investigate the possibility of applying the proposed method in some specific domains, such as in bioinformatics, in order to evaluate its possible exploitation for the analysis of complex biological networks consisting of several connected entities. This would also give us the possibility to perform an analysis of the results from a qualitative viewpoint, with the support of biologists.

Acknowledgments

We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944). The authors wish to thank Lynn Rudd for her help in reading the manuscript.

References

[1] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "Rankclus: integrating clustering with ranking for heterogeneous information network analysis," in *EDBT '09*. New York, NY, USA: ACM, 2009, pp. 565–576.

[2] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *ACM SIGKDD*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 797–806.

[3] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *SIGKDD '11*. NY, USA: ACM, 2011, pp. 1298–1306.

[4] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in *ECML PKDD 2010*, ser. LNCS. Springer Berlin, 2010, vol. 6321, pp. 570–586.

[5] C. Loglisci, M. Ceci, and D. Malerba, "Relational mining for discovering changes in evolving networks," *Neurocomputing*, vol. 150, pp. 265–288, 2015.

[6] P. Angin and J. Neville, "A shrinkage approach for modeling non-stationary relational autocorrelation," in *ICDM*. IEEE Computer Society, 2008, pp. 707–712.

[7] D. Stojanova, M. Ceci, A. Appice, and S. Dzeroski, "Network regression with predictive clustering trees," *Data Min. Knowl. Discov.*, vol. 25, no. 2, pp. 378–413, 2012.

[8] D. Kocev, C. Vens, J. Struyf, and S. Dzeroski, "Tree ensembles for predicting structured outputs," *Pattern Recognition*, vol. 46, no. 3, pp. 817–833, 2013.

[9] M. Ceci, A. Appice, and D. Malerba, "Mr-SBC: a multi-relational naive bayes classifier," in *PKDD 2003*. Springer, 2003, pp. 95–106.

[10] C. Desrosiers and G. Karypis, "Within-network classification using local structure similarity," in *ECML PKDD '09*. Berlin: Springer-Verlag, 2009, pp. 260–275.

[11] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *J. Mach. Learn. Res.*, vol. 8, pp. 935–983, May 2007.

[12] Q. Lu and L. Getoor, "Link-based classification using labeled and unlabeled data," in *ICML Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

[13] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, 2nd ed. The MIT Press, 2010.
[14] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos, "Using ghost edges for classification in sparsely labeled networks," in *Proc. 14th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*. ACM, 2008, pp. 256–264.
[15] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data." *AI Magazine*, vol. 29:3, pp. 93–106, 2008.
[16] D. Jensen, J. Neville, and B. Gallagher, "Why collective inference improves relational classification," in *Proc. 10th ACM SIGKDD*. ACM, 2004, pp. 593–598.
[17] M. Bilgic and L. Getoor, "Effective label acquisition for collective classification," in *ACM SIGKDD*. ACM, 2008, pp. 43–51.
[18] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proc. 20th ICML*. AAAI Press, 2003, pp. 912–919.
[19] H. Rahmani, H. Blockeel, and A. Bender, "Predicting the functions of proteins in protein-protein interaction networks from global information," *Journal of Machine Learning Research*, vol. 8, pp. 82–97, 2010.
[20] A. Appice, M. Ceci, and D. Malerba, "An iterative learning algorithm for within-network regression in the transductive setting," in *Discovery Science 2009*. Springer, 2009, pp. 36–50.
[21] G. Pio, F. Serafino, D. Malerba, and M. Ceci, "Multi-type clustering and classification from heterogeneous networks," *Inf. Sci.*, vol. 425, pp. 107–126, 2018.
[22] X. Kong, B. Cao, P. S. Yu, Y. Ding, and D. J. Wild, "Meta path-based collective classification in heterogeneous information networks," *CoRR*, vol. abs/1305.4433, 2013.
[23] A. Woznica, A. Kalousis, and M. Hilario, "Learning to combine distances for complex representations," in *ICML 2007*, Z. Ghahramani, Ed., vol. 227. ACM, 2007, pp. 1031–1038.
[24] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Int. Rev. Data Min. and Knowl. Disc.*, vol. 5, no. 5, pp. 216–233, 2015.
[25] P. J. Brown and J. V. Zidek, "Adaptive multivariate ridge regression," *The Annals of Statistics*, vol. 8, no. 1, pp. 64–74, 1980.
[26] L. Breiman and J. Friedman, "Predicting multivariate responses in multiple linear regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 1, pp. 3–54, 1997.
[27] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.
[28] G. Liu, Z. Lin, and Y. Yu, "Multi-output regression on the output manifold," *Pattern Recognition*, vol. 42, no. 11, pp. 2737–2743, 2009.
[29] G. Tsoumakas, E. Spyromitros-Xioufis, A. Vrekou, and I. Vlahavas, "Multi-target regression via random linear target combinations," in *ECML-PKDD 2014*, ser. LNCS, vol. 8726, 2014, pp. 225–240.
[30] Y. Dong, Y. Yang, J. Tang, Y. Yang, and N. V. Chawla, "Inferring user demographics and social strategies in mobile social networks," in *Proc of ACM SIGKDD 2014*, 2014, pp. 15–24.
[31] D. Chakrabarti, S. Funiak, J. Chang, and S. A. Macskassy, "Joint inference of multiple label types in large networks," in *Proc of ICML 2014*, vol. 32. JMLR.org, 2014, pp. 874–882.
[32] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-target regression via input space expansion: treating targets as inputs," *Machine Learning*, pp. 1–44, 2016.
[33] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, 2011.
[34] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," in *Proc. of the 18th Conference on Uncertainty in Artificial Intelligence*, ser. UAI'02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 485–492.
[35] U. Pompe and I. Kononenko, "Naive bayesian classifier within *ilpr*," in *In Proc. of the 5th Int. Workshop on Inductive Logic Programming*, L. D. Raedt, Ed., Dept. of Computer Science, Katholieke Universiteit Leuven, 1995, pp. 417–436.
[36] U. Pompe and I. Kononenko, "Linear space induction in first order logic with relief," in *CISM Lecture Notes*, R. Kruse, R. Viertl., G. Della Riccia, Ed., Udine, Italy, 1994.
[37] P. Flach and N. Lachiche, "Naive bayesian classification of structured data," *Machine Learning*, vol. 57, no. 3, pp. 233–269, 2004.
[38] M. Ceci and A. Appice, "Spatial associative classification: propositional vs structural approach," *J. Intell. Inf. Syst.*, vol. 27, no. 3, pp. 191–213, 2006.
[39] L. Getoor and L. Mihalkova, "Learning statistical models from relational data," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '11. New York, NY, USA: ACM, 2011, pp. 1195–1198.
[40] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer Publishing Company, Incorporated, 2008.
[41] G. O. Roberts and S. K. Sahu, "Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 2, pp. 291–317, 1997.
[42] Y. Guo, X. Niu, and H. Zhang, "An extensive empirical study on semi-supervised learning," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 186–195.
[43] J. R. Quinlan, "Bagging, boosting, and c4. 5," in *AAAI/IAAI, Vol. 1*, 1996, pp. 725–730.
[44] C. Bishop, "Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn," 2007.
[45] S. Kotsiantis, "Combining bagging, boosting, rotation forest and random subspace methods," *Artificial Intelligence Review*, vol. 35, no. 3, pp. 223–240, 2011.
[46] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
[47] S. Karlos, N. Fazakis, A.-P. Panagopoulou, S. Kotsiantis, and K. Sgarbas, "Locally application of naive bayes for self-training," *Evolving Systems*, pp. 1–16, 2016.

Francesco Serafino is a Ph.D student at the Dept. of Computer Science, University of Bari (Italy). He started his PhD course in 2014 in the Knowledge Discovery and Data Engineering research group (KDDE), a joint research group of the LACAM laboratory, under the supervision of Prof. Michelangelo Ceci. He served as a reviewer in a wide range of international conferences during his Ph.D studies. His research interests include Multi-type Clustering, Multi-type Classification and Structured Output Prediction.

Gianvito Pio, Ph.D, is a Research Fellow at the Dept. of Computer Science, University of Bari (Italy). He has published 17 papers in journals, books and conference proceedings, including 7 papers in journals, such as BMC Bioinformatics, PloS ONE, Information Sciences, the Journal on Computing and Cultural Heritage and the International Journal of Computational Intelligence Systems. He has participated in the scientific committee of international conferences and served as a reviewer in a wide range of international conferences and journals. His interests include bioinformatics, multi-type clustering and classification on heterogeneous networks.

Michelangelo Ceci, Ph.D., is an associate professor at the Dept. of Computer Science, University of Bari, Italy. His research interests are in data mining and machine learning. He has published more than 170 papers in reviewed journals and conferences. He has been the unit coordinator of EU and national projects. He has been in the Program Committee of many conferences, including: IEEE ICDM, SIGKDD, AAAI, IJCAI, ECML-PKDD, SIAM SDM, DS and ISMIS. He is in the editorial board of DMKD, MLJ, JIIS, IJDSN, IJSNM and "Intelligenza Artificiale". He was program (co-)Chair of SEBD2007, Discovery Science 2016, ECML-PKDD2017, ISMIS 2018 and General Chair of ECML-PKDD2017.