Ensemble Learning for Multi-type Classification in Heterogeneous Networks (Discussion Paper)

Francesco Serafino, Gianvito Pio, and Michelangelo Ceci

University of Bari, Dept. of Computer Science, Via Orabona, 4, 70125 Bari (Italy)

Abstract. In the literature, several methods have been proposed for the analysis of network data, but they usually focus on homogeneous networks. More recently, the complexity of real scenarios has impelled researchers to design classification and clustering methods able to work also on heterogeneous networks, which consist of different types of objects and links. However, they often make assumptions on the structure of the network that are too restrictive or do not exploit different forms of network correlation and autocorrelation. Moreover, when several nodes of the network have missing values, standard methods can lead to either building incomplete classification models or to discarding relevant dependencies (correlation or autocorrelation). In this discussion paper, we describe an ensemble learning approach for multi-type classification that we proposed recently, which is able to exploit *i)* the possible presence of correlation and autocorrelation phenomena, and *ii)* the classification of instances (with missing values) of other node types in the network. Experiments performed on real-world datasets show that the proposed method is able to significantly outperform state-of-the-art algorithms.

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, which can be found in several fields, including biology, epidemiology, geography, finance, etc. 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 is that of social networks: objects represent people and links represent the friendship relationships. 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 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 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 [15] and [16] the

SEBD 2018, June 24-27, 2018, Castellaneta Marina, Italy. Copyright held by the author(s).

authors propose new clustering solutions, whereas in [6] and [7] 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 [9]. However, existing methods suffer from one or more of the following limitations: *i)* they impose strict restrictions on the structure of the network, which must be know apriori; *ii)* 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* [1, 14], according to which two connected nodes in the network tend to share some properties; *iii)* 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; *iv)* they are not able to classify objects of different types, where each type can have a different set of labels.

In this discussion paper, we describe the approach proposed in [13], which is able to work 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 exploit 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 [2], in order to also capture network autocorrelation phenomena, handle relevant missing values and perform multi-type classification. 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. Specifically, we propose two extensions of the Mr-SBC algorithm: ***i)*** **ST-MrSBC (Self-Training MrSBC)**, which is able to capture possible autocorrelation phenomena by resorting to a variant of the self-training method; ***ii)*** **MT-MrSBC (Multi-Type MrSBC)**, which iteratively analyzes objects of multiple types, in order to predict possible missing values, belonging either to the target type of the main classification task or to other object types that are strongly related to the main classification task. We consider the network classification task according to the *within-network* setting [4]: objects for which the class is known are linked to objects for which the class must be estimated [10] (which can be either the subject of the main classification task or other objects related to the main classification task). This semi-supervised setting, which allows the classification phase to take advantage of both labeled and unlabeled examples, leads to smoother predictions [3]. In multi-type classification, we add an additional mechanism to smooth the prediction function: capturing relationships among objects of different types, i.e, capturing correlations among labels of objects of different types.

2 Problem Statement and Background

Before describing the proposed method, in the following we introduce the notation used. We work on heterogeneous networks, which we formally define as

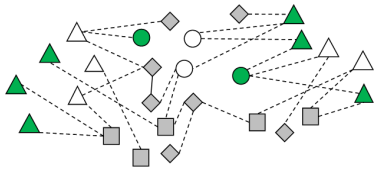

Fig. 1. An example of a heterogeneous network. The shape indicates the type of nodes: triangles and circles are nodes of target types (primary or secondary), and can be labeled (green) or unlabeled (white). Gray nodes belong to task-relevant types.

$G = (V, E)$, 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 possible edge types in the network.

In the considered task, we define a role for each node type: $\mathcal{T}_t$ (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 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.

3 The proposed ensemble learning method

In this section we describe the two solutions ST-MrSBC and MT-MrSBC. Both take as input a partially labeled heterogeneous network and work iteratively. At each iteration, they build an ensemble of Mr-SBC [2] 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 model.

Following the idea in [12] (for multi-label classification tasks), MT-MrSBC shuffles all the target types. This 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 [8].

Once the order of analysis has been defined, MT-MrSBC samples a subset of nodes of the first target type and builds a weak (since built from a subset

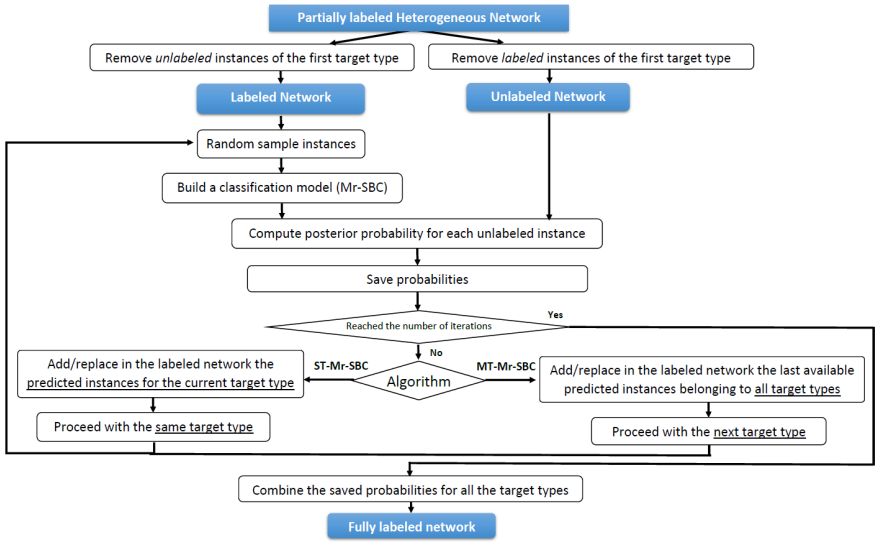

Fig. 2. High-level description of ST-MrSBC and MT-MrSBC.

of instances) 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. Then, we select a new target type and repeat the process. Note 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 same target type in the previous iterations. The main difference with respect to the standard self-training is in the construction of the labeled network when a new iteration starts. Specifically, ST-MrSBC builds an ensemble of weak classifiers from different subsets of labeled nodes, instead of considering only the output of the last iteration.

Construction of labeled and unlabeled networks

At each iteration, we select the target attribute of the next target type and build 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. Note 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 take into account some labels predicted at the previous iterations. This choice is, apparently, in contrast with most of the self-training approaches, which usually 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 [5], 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.

Estimation of probabilities

At the end of each iteration, we build a classification model through Mr-SBC for the current target type from the network of labeled nodes. We remind that Mr-SBC is a naïve Bayes classifier which relies on a set of first-order rules induced from data stored in the tables of a relational database (see [2] for details).

For each unlabeled node v', the identified classification model is exploited to compute the posterior probability (which takes into account both network correlation and network autocorrelation) and its label $\psi_t(v')$, according to the following equation based on the Bayes' theorem:

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

where Y_c is a possible class value of the target attribute y, and $R_{v'} \subseteq R_D$ is the subset of first-order rules, identified by Mr-SBC, covering the object v'.

The labeled nodes are then added to the labeled network for the subsequent iterations, as described before. The probabilities are 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: 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 in the following way:

$$\psi_t(v') = argmax_c \frac{1}{z} \sum_{k=1}^{z} P(Y_c|R_{v'k}) = argmax_c \frac{1}{z} \sum_{k=1}^{z} \frac{P(Y_c)P(R_{v'k}|Y_c)}{P(R_{v'k})}, \quad (2)$$

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. The rationale behind the combination in Equation (2) 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.

Dataset	#Nodes	#Edges	Primary targets	Secondary targets
MOVIE	2,051	59,532	movies	users
NBA	36,593	63,924	teams	players
YELP	1,387,596	1,371,060	business - users	N/A
IMDB	212,039	342,161	movies	users
STACK	92,800	114,385	posts	users

Table 1. Quantitative information about the considered datasets

4 Experiments

We performed our experiments on five heterogeneous networks: MOVIE, NBA, YELP, IMDB and STACK. Some quantitative information about these datasets can be found in Table 4, while additional information can be found in [13].
In order to evaluate 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 of our method, i.e., the iterative nature of the ensemble-based self-training approach (introduced in **ST-MrSBC**) and the multi-type classification of both primary and secondary targets (implemented in **MT-MrSBC**). Moreover, we evaluate the performance of MT-MrSBC by considering both a lexicographic (**LexicographicMT-MrSBC**) and a random ordering (**RandomMT-MrSBC**) of target 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** [7], which is natively able to work on heterogeneous networks, and *iv)* the algorithm **HENPC** [11], 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) of the datasets IMDB, STACK and YELP.

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.

We collected all the classification accuracy values and performed the Nemenyi post-hoc tests on single datasets, on all the datasets and on the reduced datasets for the comparative evaluation. The result of the statistical tests are depicted in Figure 3. In particular, we can observe that for the datasets NBA and for the target type *users* of the dataset Yelp, the improvement 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 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* → *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* → *users* and *users* → *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.

Finally, the results obtained by our comparative evaluation (Figure 3 - last chart) shows that the improvement provided by the proposed method is able to give Mr-SBC the advantage of outperforming the competitors Indeed, the original 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 advantage comes from the combination of capturing label dependencies between multiple types and of the ensemble learning approach (MT-MrSBC), which improves the accuracy.

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.

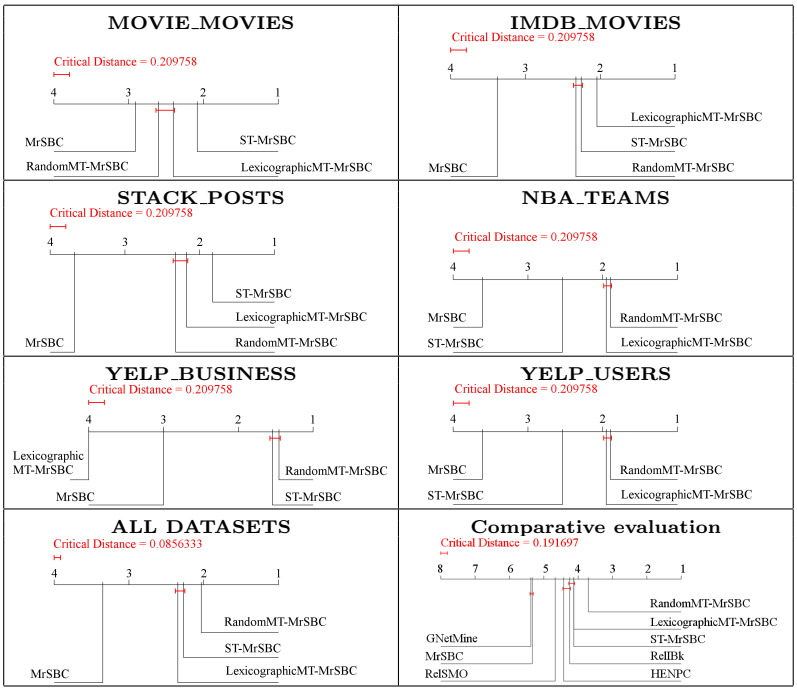

Fig. 3. Results of the Nemenyi post-hoc test on the average accuracy. Better algorithms are positioned on the right-hand side, and those that do not significantly differ in performance (at p-value = 0.05) are connected with a line.

5 Conclusions

In this paper, we proposed an extension of the system Mr-SBC which works on heterogeneous networks and that is able to solve multi-type classification tasks, by capturing both correlation and autocorrelation phenomena. Experiments performed on real datasets show that both the proposed variants are able to significantly outperform the original Mr-SBC, especially with a random ordering of the target types, as well as four other well-known competitor algorithms.

As future work, we plan to study the sensitivity to the sampling size and the possibility to select only a subset of labeled nodes for the next iteration.

Acknowledgments

We would like to acknowledge the European project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (ICT-2013-612944).

References

1. P. Angin and J. Neville. A shrinkage approach for modeling non-stationary relational autocorrelation. In *ICDM*, pages 707–712. IEEE Computer Society, 2008.
2. M. Ceci, A. Appice, and D. Malerba. Mr-SBC: a multi-relational naive bayes classifier. In *PKDD 2003*, pages 95–106. Springer, 2003.
3. O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 2nd edition, 2010.
4. C. Desrosiers and G. Karypis. Within-network classification using local structure similarity. In *ECML PKDD '09*, pages 260–275, Berlin, 2009. Springer-Verlag.
5. Y. Guo, X. Niu, and H. Zhang. An extensive empirical study on semi-supervised learning. In *ICDM*, pages 186–195. IEEE, 2010.
6. M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *SIGKDD '11*, pages 1298–1306, NY, USA, 2011. ACM.
7. M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. In *ECML PKDD 2010*, volume 6321 of *LNCS*, pages 570–586. Springer Berlin, 2010.
8. J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Publishing Company, Incorporated, 2008.
9. C. Loglisci, M. Ceci, and D. Malerba. Relational mining for discovering changes in evolving networks. *Neurocomputing*, 150:265–288, 2015.
10. S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, May 2007.
11. G. Pio, F. Serafino, D. Malerba, and M. Ceci. Multi-type clustering and classification from heterogeneous networks. *Inf. Sci.*, 425:107–126, 2018.
12. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.
13. F. Serafino, G. Pio, and M. Ceci. Ensemble learning for multi-type classification in heterogeneous networks (doi: 10.1109/tkde.2018.2822307). *IEEE TKDE*, 2018.
14. D. Stojanova, M. Ceci, A. Appice, and S. Dzeroski. Network regression with predictive clustering trees. *Data Min. Knowl. Discov.*, 25(2):378–413, 2012.
15. 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*, pages 565–576, New York, NY, USA, 2009. ACM.
16. Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *ACM SIGKDD*, pages 797–806, 2009.