International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 17 (2025) pp. 99–116

Article

A Study of Transductive Graph-Based Regression

Renan Guilherme Nespolo [1,*], Alan Demétrius Baria Valejo [2] and Alneu de Andrade Lopes [1]

[1] Institute of Mathematics and Computer Science, University of São Paulo, 400, Trabalhador São-Carlense Avenue, São Carlos 13566590, São Paulo, Brazil; alneu@icmc.usp.br

[2] Department of Computing, Federal University of São Carlos (UFSCar), km 235, Washington Luís Road, São Carlos 13565905, São Paulo, Brazil; alanvalejo@ufscar.br

* Correspondence author: renan.nespolo@usp.br

Received date: 4 April 2024; Accepted date: 6 June 2024; Published online: 6 January 2025

Abstract: Regression methods play an important role in many real-world applications such as econometric, pattern recognition, and prediction of protein chains to cite a few tasks. Although some studies have been exploring graph-based prediction of continuous values also called regression problems, in the semi-supervised context, they have not yet explored the formalism and potential of complex network theory and usually, they address only the classification problem (discrete labels). Beyond that, those graph-based approaches do not address factors such as the impact of using different graph construction methods on the regression result, exploration of topological features of the adopted graph representation, how the inference is carried out, and what type of graph-based label propagation strategy is adopted. Here, in the proposed approach, all the relevant dimensions of the technique, such as knowledge and data representations, inference strategy, and evaluation criteria are addressed. First, we combine two well-known network construction models, spectral and k-NN. Separately, the k-NN network generates a regular degree connected network and the spectral network generates a network with groups of vertices densely connected internally, however, it leads to several unconnected groups (or sub-networks), which harms methods such as Random Walks and label propagation. Therefore, we combine spectral and k-NN to ensure a hybrid connected network. Second, as the inference strategy, we use a transductive method for label propagation by using a robust network technique combining Random Walks for the regression task, which turns forward the propagation of values by the network in unlabeled objects using the least-squares regression method. An empirical analysis of different data sets shows that our strategy surpasses traditional approaches considering the measures used in the evaluation.

Keywords: semi-supervised; graph-based; graph laplacian; transductive; propagation

1. Introduction

For any new Artificial Intelligence technique proposal, at least 4 interdependent dimensions must be addressed. They are: (1) how data, and (2) knowledge are represented, (3) how the inference is carried out considering the chosen data and knowledge representations, and last but not least (4) how to evaluate the new technique. Indeed, Other dimensions can be analyzed depending on the domain and application involved. Here we propose a graph-based approach for the regression problem and all those dimensions are discussed.

Usually, propositional data, i.e., attribute-value (AVL) tables are used to represent data in the regression tasks. Such a representation is effective for representing objects by their characteristics, but it does not, naturally, represent relationships between objects. Graphs on the other hand effectively represent objects and the relationships between them. It is important to make clear that some problems in which parts interact with each other are simply and straightforwardly represented by graphs. In these cases, parts are represented by vertices and interactions by edges between vertices. In other problems, there is only the information of the instances and eventually their labels, however, even in this second case, graphs can be constructed from these data so that vertices represent data instances and edges represent similarity between these instances, i.e., in this graph, edges will connect similar instances. Once this representation is obtained it is important to note that graph representation can easily capture (i) the topological structure of connectivity among data leading to an interesting approach for dealing with clustering and community detection tasks; (ii) hierarchies, i.e., groups and subgroups of vertices can be identified;

(iii) density in the vertices connections, enabling detection of groups with arbitrary shapes; and (iv) manifold in data, i.e., vertices in the graph can be connected via a path through high-density regions in the data manifold [1]. When building the network (In this study, henceforward, the terms network and graph are considered synonymous). from AVL data, there exist different techniques for connecting the nodes (see Berton et al. [2]), but usually, the rationale behind them is that the probability of connection between two vertices is proportional to the similarity between them. Therefore, close instances tend to be more densely linked together than to the rest of the data. Hence, this justifies the number of complex network approaches for clustering and community detection problems [3–5].

Another common graph-based application is semi-supervised classification. In this case, edges are used to propagate information from labeled to unlabeled vertices through the entire network [6,7]

This wide usage of complex networks in unsupervised (clustering or community detection problems) and semi-supervised learning does not reflect its use in the regression tasks. To minimize this gap and be motivated by the richness of graph representation as well as the formalism provided by complex networks theory, this paper presents a new graph-based regression method.

Regression is a technique originating from statistical theory and used mainly to analyze practical problems in several areas such as economics, physics, and computing. The main characteristic of the models that apply this technique is that a deterministic function does not build the relationship between the target variables and the covariates; instead, they are based on random errors. This implies that the target variable is a random variable, the distribution of which depends on interpretative variables [8]. This technique was initially formulated for supervised learning with data represented as attribute-value tables or propositional representations [9].

Regression is commonly adopted when the nature of the problem has characteristics of continuous value data, i.e., real values in the labels. By working with covariates, this technique is molded for supervised learning, since it needs labeled data for its functioning. In hypotheses with labels with unknown values, some approaches are necessary to perform a regression, such as: eliminating unknown values, correlation between variables, and definition of similarity measure, among others [10]. These additional tasks make the semi-supervised approach a complicated environment for regression techniques.

Berton et al. [2] studied how the graph's construction has a direct impact on the classification task. This study used a specific network topology to perform the classification. This approach can be advantageous in using regression in semi-supervised learning compared to propositional methods since it is possible to select specific data through connections between them to make the regression.

Although some works have used graphs in regression problems, such as [11–13], they have not yet explored the full potential and robustness of complex network theory. Current works only use the adjacency matrix, which represents the graph, but they do not address the impact of using different graph construction methods (such as epsilon, spectral, k-NN, Mk-NN, to cite a few methods) in the regression. Moreover, current methods, while modeling data from the adjacency matrix, use traditional regression strategies. Therefore, the inference does not explore the structural and topological features of the graph which model the problem. For example, label propagation or techniques for label diffusion, widely used in networks, that operate directly on the network structure and explore vertex neighborhood and connectivity.

We propose a transductive semi-supervised approach for regression with the following characteristics. The data representation is graph-based combining two topologies (k-NN network and spectral network) in order to better handle the volume of data. Inference by label propagation is realized by a combination of Random Walks [14] and linear regression. The Evaluation of the proposed approach follows the commonly adopted measures for regression tasks: MAE and RMSE.

The paper's contributions can be summarized as follows.

1. Proposal of a hybrid network approach for solving transductive regression problems.
2. New transductive inference technique based on Random Walks for regression issues.
3. Direct use of network features, such as vertices, edges, and neighborhoods in the inference process.

The remainder of the paper is organized as follows. Section 2 introduces the basic concepts and definitions. Section 3 presents related work. Section 4 presents our semi-supervised regression framework. Section 5 describes the empirical results. Finally, Section 6 summarizes our findings and discusses future work.

2. Background and Fundamental Concepts

Regression is a set of statistical techniques for estimating the relationships between a dependent variable (often called an explanatory variable) and one or more independent variables (called explained variables). The technique

commonly used is linear regression, in which the intersection that best fits the data following a chosen criterion. For example, the method of ordinary least squares calculates a single intercept that minimizes the sum of the square differences between the data and the intercept. Regression is mainly adopted in two situations. The first is for the prediction of values, which is used in the field of machine learning and artificial intelligence. The second is to infer causal relationships between independent and dependent variables [15]. In the machine learning area, regression is considered a supervised technique because it is highly dependent on the values mapped between the explanatory variables and the explained variables, much adopted for the economic field for forecasting values or percentage [10], and in chemistry and biology fields, in protein chains [13]. Its evaluation is usually given by approximation metrics such as MSE, RMSE, and MAE, presented in Section 2.7.

For the initial organization of the data, two dimensionality reduction techniques are presented in Section 2.1, the principal component analysis (PCA) and, the t-distributed stochastic neighbor embedding (t-SNE).

In the data representation approach two different network types are defined: the k-NN network, in Section 2.2, a network with uniform features; and the spectral network, in Section 2.3, a weighted fully-connected network.

In the inference, three methods are described, which are: Random Walks, presented in Section 2.5, and least squares regression, presented in Section 2.6. In addition to the method of transductive learning, presented in Section 2.4.

2.1. Dimensionality Reduction

Dimensionality reduction is a technique that seeks to select or extract features, visualize spaces, or simplify a problem. The method commonly used is principal component analysis (PCA). This method generates a projection given by a linear transformation based on second-order spectral analysis, called a correlation matrix. To carry out the PCA it is necessary to create a covariance matrix [16]. So given a X dataset, the covariance matrix is generated by:

$$C = \frac{1}{n-1} M^T M \tag{1}$$

where M is given by:

$$M = [m_1, m_2, m_j, \cdots, m_n] \tag{2}$$

where m_j is the mean of each column of X. Thus the main components are:

$$pca(X) = det(C - \Lambda I) \tag{3}$$

where each eigenvalue corresponds to the importance of information contained in each dimension of the dataset X and the eigenvectors (σ_i) generate the projection matrix P:

$$P = \langle \sigma, C \rangle \tag{4}$$

where it will be possible to select only the main components designed in small dimensions.

Another dimensionality reduction technique is the t-distributed stochastic neighbor embedding (t-SNE). This technique has the characteristic of minimizing the divergence between two distributions: a distribution that measures the similarity between pairs of objects given as input and a distribution that measures the similarity between corresponding pairs in low dimensional points of the embedding. Assuming that a set is given as input (high dimensionality), the objects $D = \{x_1, x_2, \cdots, x_N\}$ and a function $d(x_i, x_j)$ which calculates the distance between a pair of objects. The goal is to learn a s-dimensional integration, where each object is represented by a point, $\epsilon = \{y_1, y_2, \cdots, y_N\}$ in which $y_i \in \mathbb{R}^s$ (s is commonly defined as 2 or 3). In embedding s-dimensional ϵ, the similarities between two points y_i and y_j (that is, the low dimensional models of x_i and x_j) are measured using a standardized kernel. Specifically, the q_{ij} embedding similarity between the y_i and y_j points is computed as a normalized kernel t-Student with a single degree of freedom [17]:

$$q_{ij} = \frac{(1 + \left\| y_i - y_j \right\|^2)^{-1}}{{}_k(1 + \left\| y_i - y_j \right\|^2)^{-1}}. \tag{5}$$

The t-SNE is highly complex and it is highly recommended to use another method of dimensionality reduction, such as the PCA for situations where the data have dense characteristics [17]. The PCA technique is characterized by

the transformation of space while maintaining the global structure of the data, generating characteristics of dispersion in the data.

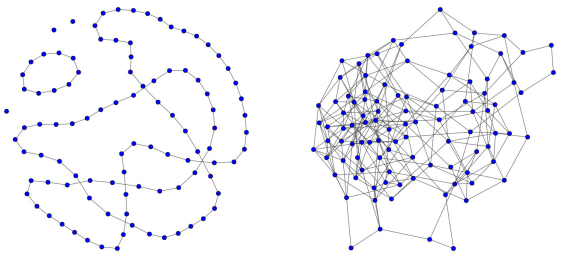
2.2. k-NN Network

Given a dataset $\mathbf{X}$, where each instance of that set is given by x such that $x \in \mathbf{X}$, the nearest neighbors of an instance are defined in terms of the Euclidean distance [18]. Commonly, the distance between two instances x_i and x_j is defined as $d(x_i, x_j)$, which:

$$d(x_i, x_j) \equiv \sqrt{(a_r(x_i) - a_r(x_j))^2}, \tag{6}$$

which x_1 and x_2 are instances of the set X. However, other distance metrics can be used [18]. Thus, the second step of the algorithm is performed using the parameter k chosen arbitrarily. This parameter defines which instances affect a given instance, if x_i is the selected instance then the nearest k instances, following the distance, given by the Equation (6), making these instances the k-nearest neighbors of x_i.

A low value of k can generate a disconnected network, as shown in Figure 1a, while when the value of k increases the density of edges in the network is also increased, as shown in Figure 1b [2].



(**a**) k=1 (**b**) k=3

Figure 1. Visualization of a k-NN network with 100 instances by changing the parameter to (**a**) k=1 and (**b**) k=3.

2.3. Spectral Network

Among the traditional models in the literature, the model that stands out in regression inference is the spectral topology described in [16]. The network used is built as follows.

Given the dataset:

$$\mathbf{X} = \{\mathbf{x_i}\}_{i=1}^{\mathbf{N}} \tag{7}$$

with N input data, both labeled and unlabeled, a weighted non-directed graph is constructed that consists of N vertices, in which each vertex represents an instance of the input data. The link between vertices is as follows. Each pair of vertices i and j are connected, as long as the Euclidean distance between x_i and x_j satisfies the given condition:

$$w_{ij} = exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2}), \tag{8}$$

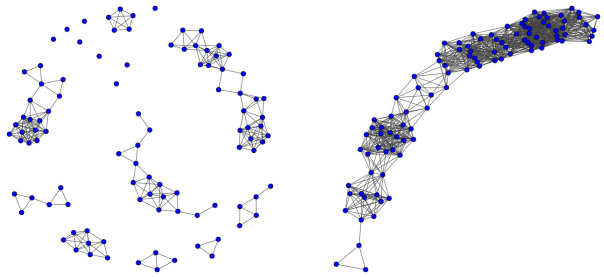
which σ^2 is an arbitrarily defined constant. The interesting features about the adjacency criterion are: geometric vision and a naturally symmetric metric [16]. However, choosing the value for the constant σ is difficult since it impacts directly the number of connections. The value w_{ij} is the weight of an undirected edge connecting the vertices i and j. The weights in the graph as a whole are usually real numbers, the selection of which must satisfy three conditions:

1. symmetry, in which $w_{ij} = w_{ji}$ for all pairs (i, j);
2. connectivity, in which the weight w_{ij} is different from zero if the vertices i and j are connected and zero if not;
3. non-negative, that is, $w_{ij} \geq 0$ for all pairs (i, j).

Therefore, an array of weights $\mathbf{N} \times \mathbf{N}$ is given as:

$$\mathbf{W} = w_{ij} \tag{9}$$

which, by definition, is a symmetric and non-negative matrix, with all its elements being non-negative. The rows and columns of the **W** matrix are indexed to the vertices of the graph and their order is not important [16]. Since a small value of σ leaves the network unconnected, as shown in Figure 2a, for a connected network a much larger value of σ is necessary (according to the scale of the algorithm), generating a network with regions very dense, as shown in Figure 2b.



(**a**) σ=0.5 (**b**) σ=2

Figure 2. Visualization of a Spectral network with 100 instances by changing the parameter σ: (**a**) σ = 0.5, and (**b**) σ = 2.

2.4. Transductive Learning

The goal of transductive inference is to estimate values of unknown functional dependencies in data points of interest without necessarily having to estimate the functions themselves. The consequence of the process is not only solving a problem but also using all the information available at the points of interest, information that is ignored in the inductive inference. In this machine learning process, the generalization step imposed by inductive inference is unnecessary to generate the same information as given by transductive inference [19], i.e., no model that could be employed in unseen data is built.

There are two transductive learning models: the first is based on a space-vector model and the second is network-based. A condition for algorithms based on the space-vector model with good performance, is to have well-separated classes in which the separating hyperplane is in a low-density region [20]. If these non-main conditions are met, the algorithms may have an unsatisfactory classification performance [20]. Network-based transductive learning algorithms have emerged as an alternative to algorithms based on the space-vector model and have been shown to remedy the aforementioned deficiencies. Transductive classification using networks aims to classify unlabeled objects considering information from neighboring objects, such as the attributes [21,22] or the class information itself [6,23].

Among the network-based transductive learning strategies, collective classification can significantly increase classification accuracy when compared to supervised inductive classification based on the space-vector model. This strategy was taken from Relational Learning [24], and normally considers networks generated by explicit information, such as hyperlinks and citations [20].

2.5. Random Walks

The algorithm based on label propagation in graphs is the Random Walks. Initially designed by Szummer & Jaakkola [25] applying Markov's random walks into a graph with a probability of transition between the instances i and j, given by:

$$p_{ij} = \frac{\mathbf{W}_{ij}}{_k \mathbf{W}_{ik}}, \tag{10}$$

with the objective of estimate probabilities of labels. W_{ij} is build by a Gaussian kernel for neighbors and 0 otherwise. Each data point x_i is associated with a probability $P(y = 1 \mid i)$ of being class 1. Given a point x_k, the probability P^t is calculated $(y_{start} = 1 \mid k)$ that started from a labeled point (of the class $y_{start} = 1$) until it reaches x_k after t steps of random walks, given by:

$$P^t = (y_{start} = 1 \mid k) = \sum_{i=1}^{n} P(y = 1 \mid i) P_{0|t}(i \mid k), \tag{11}$$

where $P_{0|t}(i \mid k)$ is the initial probability of x_i given until it reaches k after t step of random walks. x_k is then classified as 1 if $P^(t)(y_{start} = 1|k) > 0.5$ and -1 otherwise [6].

2.6. Least Squares Regression

The inference given as least squares regression (LSR) considers the sum of squares of errors made by a model in which any two values are chosen for v_0 and v_1, given in the equation:

$$\Upsilon = \sum_{i=1}^{n} (\epsilon)^2 = (y_i - v_0 - v_1 x_i)^2, \tag{12}$$

Deriving the expression given by the Equation (12), in relation to v_0 and v_1, and equaling zero, the equations are given:

$$\hat{v_0} + \hat{v_1} \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{13}$$

and

$$\hat{v_0} \sum_{i=1}^{n} x_i + \hat{v_1} \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \tag{14}$$

which corresponds to the system of normal equations, in which the solution of this system of equations provides the least squares estimators for the parameters v_0 and v_1, given by:

$$\hat{v_0} = \overline{y} - \hat{v_1} - \overline{x} \tag{15}$$

and

$$\hat{v_1} = \frac{\sum_{i=1}^{n} x_i y_i + n\overline{xy}}{\sum_{i=1}^{n} x_i^2 - n\overline{x}^2} \tag{16}$$

reaching the approximation of the parameters, enabling the prediction of the response through the model [8]:

$$\hat{y_x} = \hat{v_0} + \hat{v_1} x. \tag{17}$$

There are many other types of regression, but for classification problems, one type of regression used is the conditional probability regression [11,26]. The classification is performed using the conditional probability given by Equation (18):

$$\xi_{ij} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{j \neq i} exp(-\|y_i - y_j\|^2)}, \tag{18}$$

2.7. Evaluation Measures

The error generated between the predicted values and the labeled values follows certain measures depending on the learning method applied. For supervised and semi-supervised classification methods, the commonly adopted measure is accuracy, given by the Equation (19):

$$acc = \frac{tp + tn}{tp + tn + fp + fn}, \tag{19}$$

where tp, true positive, tn true negative, fp, false positive and fn false negative, and all variables presented in the Equation (19) must contain integer values.

For the regression methods, the metrics commonly adopted are: the absolute mean of the error (MAE), given by Equation (20) and the root-mean-square error (RMSE), presented in the Equation (21):

$$MAE = \frac{\sum_{i}^{n} |\hat{y}_i - y_i|}{n}, \tag{20}$$

where $\hat{y}_i$ is the predicted value, y_i is the correct value and n the number of elements.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}, \tag{21}$$

3. Related Work

The concept of semi-supervised graph-based learning was first presented in 2005 by Zhu, X. [27]. This paper introduces a neighborhood-based propagation method for classification, and shortly afterward the main challenges tasks in the area [28], such as transformation from propositional representations to graph-based representations, model generation, kernel generation, vertices connection methods, regression, regularization, and graph construction.

Studies addressing semi-supervised learning are mainly driven by the fact that in many problems, the cost to acquire labeled data can be quite high, and semi-supervised learning needs only a small set of labeled data to achieve results as good as supervised approaches [29]. However, regression, in the context of machine learning, is usually considered as supervised learning. By using covariates to predict results. Regression suffers a drawback in the semi-supervised context when it is performed using propositional representation since the information on the labels is almost absent. Graph-based methods on the other hand can mitigate this disadvantage by using strategies that take advantage of the structure of relationships between objects, i.e. by using the information about relationships among objects.

Ni, B. et al. [11], in 2012, presented one of the main graph-based regression approaches for the regression problem. The work presented was a framework based on propagation learning, for two essential data mining tasks: classification and regression. First, it transforms the space using Principal Component Analysis (PCA) to decrease the dimensions of the space. The information propagation was performed using conditional probability regression. A limited evaluation was carried out on three different data sets, two of them are for classification purposes: Wine and Iris; and one for regression purposes: FG-NET. The regression problems predict real values instead of discrete classes represented by integer values. The FG-NET contains the faces of several people, and for each person, there are several images representing that same person with different ages in his life. The performance measure used was the mean absolute error (MAE), presented in Section 2.7, Equation (20).

Kraus et al. [12] presented an approach for semi-supervised regression problems. The method uses the Laplacian method to create the graph using an exponential function, which is the network following the spectral topology, presented in Section 2.3. The function used to predict the values was Laplacian regularization. The work was also divided into two parts, the first using simpler data sets such as Wine and Concrete data sets [30] set. He made comparisons of the method presented with propositional methods, found in the literature, that work with semi-supervised regression such as SVR and Random Forest. The metric adopted was the root-mean-square error (RMSE), presented in Section 2.7, Equation (21). The second part was attributed to the effectiveness of the algorithm presented to reconstruct the installation data of the Lyon Metropolis network sewer pipes.

Sheikhpour, R. et al. [13], in 2018, proposed a framework that makes a selection of semi-supervised characteristics using normalization to calculate the sparse transformation vector for regression problems. The work used two semi-supervised dispersion matrices based on the Laplacian matrix model to represent the labeled and unlabeled data for the regression problem. The performance of the framework was evaluated using regression data sets for QSAR [31] problems. The metric used was RMSE given in the Equation (21). The framework used three sets of protein network data to attest its effectiveness: ROCK, FYN, and PLK3, in which the values of the labels to be predicted were active values of a given protein represented by real values.

Zhang et al. [32] proposed a matrix-based regression algorithm for classification, in which the input matrices to be classified are used directly to learn two regression matrices for each input matrix order. The network was built using the k-NN network topology. The method used a ranking criterion to select which object will perform the inference first, the objects that have a neighborhood with a greater number of labeled objects have the inference priority. The performance measure used was the accuracy, presented in the Section 2.7, Equation (19).

In our research, the works presented in the literature commonly apply the dimensionality reduction PCA to data visualization and reducing the dimensions complexity for using solutions based on vector-space [11,32–36] and the Laplacian similarity method, (pure or variations) [12,31,37–39]. The method shows good results, but when used as a network generates a fully connected topology, with more connections than necessary to perform the inference of a given object, which can represent each element connected on average to 25% of all other elements. This scenario impairs the regression in the semi-supervised context since almost all of the elements do not contain labels, *i.e.* most of the connected elements do not have labels.

These previous graph-based regression works did not discuss important dimensions that such relational representation influences. The papers cited do not address relevant aspects which has a deep influence on the performance of the algorithms such as how networks are built, how inference (label propagation) is made or how the network is used, and how the convergence of the optimization process of information propagation is achieved.

Some studies used inductive inference strategies [11,13,31]. To the extent of our knowledge previous studies have not yet explored transductive graph-based inference strategies, neither inference considering Random Walks [40], presented in Sections 2.4 and 2.5. Furthermore, the Random Walks have the characteristic of being retained in neighborhoods with dense connectivity, traversing them in a complete way and then passing them to other regions in the network.

Another feature not explored by the related works is the reliability using network methods. For this problem, it is necessary to determine if the variance of the method remains low even with different initializations of the network. In this way, this work focuses on the construction of a method graph-based for the regression problem in semi-supervised environments using two different network topologies and the use of robust methods to prioritize the inference of the method.

4. Proposal: Semi-Supervised Transductive Graph-based Regression (TGBR)

In this section, we present a semi-supervised transductive graph-based regression (TGBR) method. Firstly, we describe a 'hybrid' method for network construction, Section 4.1, and then a label propagation method, Section 4.2.

A summary of the TGBR method is depicted as a flowchart in Figure 3. First, the PCA or t-SNE dimensionality reduction is applied to the data. Then, the Hybrid Network is built, connecting vertices according to similarities between them, and a spectral weight is associated with each edge. The second stage, the Label Propagation addresses issues related to inference by using the Random Walks method to walk over the network and then the LSR regression to assign values to the vertices. In this step, labeled vertices are initialized in random order. After all objects are labeled, the results are stored. The Label Propagation process is repeated γ times. After all the stored results, the evaluation is carried out, using the metrics defined in Section 4.3.

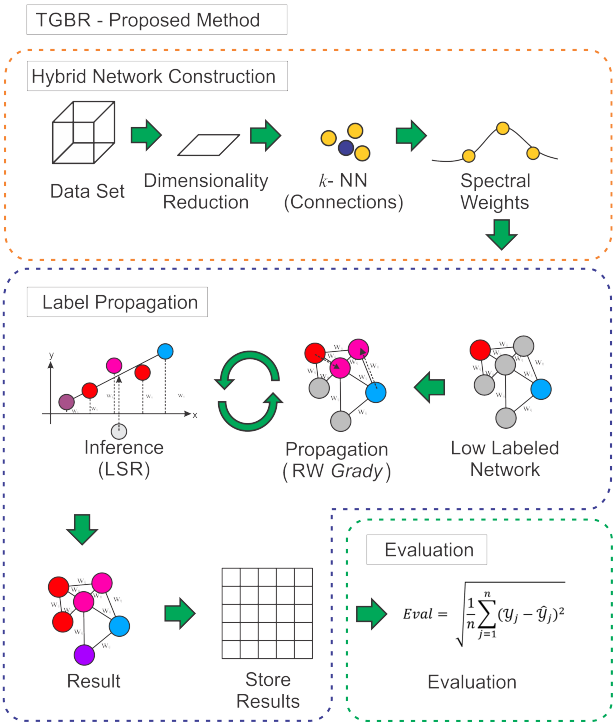
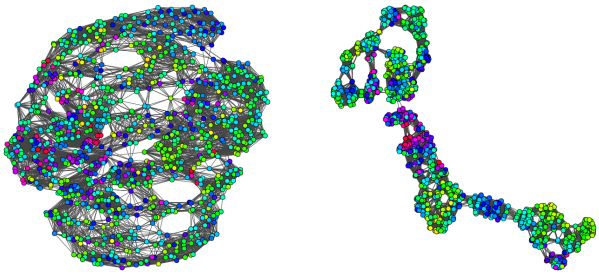


Figure 3. flowchart of the proposed method TGBR.

4.1. Hybrid Network Construction

The first step of TGBR is the reduction dimensionality transformation. For this task, we adopt two different techniques, PCA and t-SNE, and compare both since, as commented in Section 2.1.

The PCA technique, by definition [17], is characterized by the transformation of space while maintaining the global structure of the data, generating characteristics of dispersion in the data as presented in Figure 4a. In contrast, the t-SNE has the characteristic of maintaining the local structure of the data, generating clusters or community structures, as shown in Figure 4b.



(**a**) PCA (**b**) t-SNE

Figure 4. Visualization of the built network of the Concrete set using dimensionality reduction. The colors on the vertices are defined by the activation of the nodes.

After the transformation of the propositional data, usual models seek to generate a $n \times n$ similarity matrix, which will be used to connect the elements of the graph. These connections will be determined by the topological model used, each topological model generally containing a parameter that controls the degree of connections of each vertex. Among these models, the spectral model topology stands out in regression inference described in [16], presented in Section 2.3.

The spectral model leads to good results, considering that σ is set to a minimum value that guarantees a connected network. Nonetheless, this value can lead to a densely connected network. The semi-supervised propagation strategy may be impaired due to the lack of labeled vertices in the neighborhood of the target variables, forcing the inference to work with little data to infer the correct prediction, affecting the performance of the algorithm.

Among the topological models, one that generates a network with a balanced distribution of connections is the k-NN networks, presented in Section 2.2. In this network the vertices i, j are connected by an edge if i is one of k-neighbors closest to j or vice versa. k is a hyperparameter that controls the density of the network. k-NN has the property of adaptive scales, as the neighborhood radius is different in regions of low and high data density. A small k can result in a disconnected graph. For label propagation, this is not a problem if each connected vertex has some labeled vertices, in the semi-supervised transductive context [41]. Thus, the rationale behind the proposed framework was to combine the topologies of the k-NN and spectral networks, in which connections are generated using the k-NN model weighted using the spectral model.

For instance, if the set has 4,000 vertices, if we adopt only the spectral network, considering that σ is configured for a minimum value that guarantees a connected network, as presented in Figure 4a, we will take a vertex as an example. Firstly the calculation of the weights between all vertices with the chosen vertex, each calculation contains two operations with powers, as presented in the Equation (8), Section 2.3. At the end of this operation, an average of 3,000 of these calculated weights will be equal to zero and are discarded, according to the system given in Section 2.3, while the 1,000 that are different from zero will be connected to that chosen vertex. Assuming that k is defined as $k = 30$, although the number of operations is the same, the complexity of the calculation is lesser since the weight of that defined in k-NN is contained in the calculation of the spectral network. In this way, after executing the k-NN, the auxiliary structure will have a size of $30 \times 4{,}000$, and when applying the spectral weight we take advantage of the values already stored in this auxiliary structure and all the calculated weights are used and connected at the chosen vertex. This model also assists the realization of the choices of the vertices in the inference that will be $k = 30$ instead of 1,000.

4.2. Label Propagation

In this Section, we show the great differential of the proposed approach because this inference methodology is only possible for a graph-based model and differs from all previous works presented previously. We propose a robust network walking technique for propagating values and a method for attesting the reliability of the generated result for comparison with the results generated by the works contained in the literature.

To determine the order of propagation of the vertices, an adaptation of the Random Walks [6,42], presented in Section 2.5, was adopted.Initially, a labeled vertex is selected, and then one of its neighbors unlabeled is selected at random, with probability given by the Equation (22):

$$p_{ij} = \frac{w_{ij}}{\varrho_i}, \tag{22}$$

which w_{ij} is the associated weight and ϱ_i the associated degree for each vertex selected. After the choice of the vertex is performed and the LSR, considering only the labeled neighbors, as presented in Section 2.6, repeating this process until the entire network is labeled, as shown in Figure 5, where the gray vertices are vertices unlabeled, the red and blue vertices are initially labeled and the vertices of other colors is labeled by LSR. If all the analyzed neighbors of the labeled vertex are also labeled, a new labeled vertex is randomly selected and the process is restarted until the network is fully propagated. The TGBR algorithm is presented in the Algorithm 1.

Algorithm 1 : TGBR

Input: ℓ, k, D

/* D is the similarity matrix of the distance between the pairs, k is the hyperparameter that controls the number of connections between vertices, and ℓ is the array with discretized (labeled) values initially. */

Output: φ

/* is the array with the predicted values. */

1 $V \leftarrow \ell$

2 $E \leftarrow GL1(D, k)$

3 $W \leftarrow GL2(E, D)$

4 $G \leftarrow (V, E, W)$

/* V is the vertices, E is the edges, W is the weights associated with the edges, and ℓ is the array with discretized (labeled) values initially. */

5 $\varphi \leftarrow Randon_W alks LSR(G)$

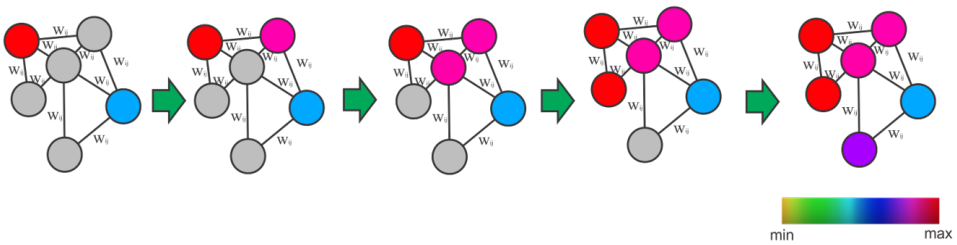


Figure 5. Randon Walks propagation.

In line 1, the initial set of values is assigned to the vertices. In line 2, the connections are defined according to the hyper-parameter k and the distances between the pairs. In line 3 the second network layer is built, defined as the spectral network. In line 4 the two networks are combined into a single network. Finally, in line 5, inference is performed, using a combination of Random Walks to select the object and linear regression to predict this object (vertex). At the end, the values are stored in φ.

To certify the reliability in the graph-based environment, in which the initializations (labeled elements) are carried out at random, several times defining the probability of certainty in that each element has a certain label. To certify the reliability γ must be defined at least $\gamma \geq 500$ [14], however in networks with many elements, for example, 5,000 elements, the execution time is a drawback. After a prediction, the values are stored and released so that the entire procedure can be carried out again, choosing new values randomly. After the end of the predictions, the average of all stored predictions is calculated, according to the adopted metric, as presented in Section 4.3.

4.3. Evaluation

Works with prediction of continuous values or regression tasks adopt approximation metrics instead of accuracy or precision measures. The evaluation can be obtained by comparing the values generated by the prediction with the real values of the target variables, and calculating some average error measure from this comparison [10]. The evaluation measures commonly adopted, presented in Section 3, for regression tasks are MAE [11,13] and RMSE [12], presented in Section 2.7.

Both measures express the average error of the predictive model, relative to the original data. The difference between MAE and RMSE is in the situation where the errors have a greater variance, such as the presence of outliers, for example, the RMSE has a greater error growth in relation to the MAE that maintains the presented error in absolute terms.

5. Results

The tested datasets are: WineQuality(white part) [12], a dataset with quality characteristics of white wines, containing 4,898 instances and 12 variables; FG-Net [11], the dataset of faces of people at different times of their lives, containing 1,002 instances with 784 variables; Concrete [12], a dataset with the elements to form the concrete mixture, containing 1,030 and 9 variables; FYN [31], a protein chain set, with 207 instances and 7 dimensions; and Wind(78 part) [43], a wheater forecast set, with 365 instances and 14 dimensions.

The experiments follow this pattern: The k hyper-parameter, which controls the degree of use between the vertices, was defined with an arbitrary $k = 30$. The datasets are tested in three different scenarios by changing the number of objects with initially discretized (labeled) values. In the first scenario, objects with initially discretized values correspond to 5% of the dataset. For the second scenario, the value of objects with discretized values initially corresponds to 10% and in the last scenario, this value is 15%. Each dataset is tested with PCA and t-SNE dimensionality reductions. The method also is compared with the following methods: FLP [11], which uses PCA dimensionality reduction, spectral graph, and linear regression; S3FSGL-2 [13], which uses the Laplacian graph and the k-nn graph and L2 regularization; LapS3L [12], which uses spectral graph and Ridge regression, local regression [44] and transductive SVR regression. The results of the proposed method in the present study compared with the other methods present in the literature are presented in the results: Tables 1–3, using the WineQuality dataset; Tables 4–6, using the Concrete dataset; Tables 7–9, using the FG-Net dataset;Tables 10–12, using the FYN dataset; and Tables 13–15, using the Wind dataset.

Table 1. Performance comparison of different methods on WineQuality dataset with 5% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0801	0.2831	0.0118	0.1085	0.011
TGBR-PCA	0.0615	0.2481	0.0065	0.0806	0.0008
S3FSGL-2	0.5773	0.7598	0.341	0.584	4.18×10^{-5}
LapS3L	0.5434	0.7371	0.3074	0.5545	0.0102
FLP	0.5433	0.7371	0.3074	0.5544	0.0102
LR	0.5775	0.7599	0.3412	0.5842	4.84×10^{-6}
SVR	0.545	0.7382	0.3048	0.5521	1.37×10^{-5}

Table 2. Performance comparison of different methods on WineQuality dataset with 10% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0967	0.3109	0.0179	0.1337	0.018
TGBR-PCA	0.0601	0.2451	0.0061	0.0781	0.0009
S3FSGL-2	0.5676	0.7534	0.3301	0.5745	7.80×10^{-5}
LapS3L	0.4997	0.7069	0.2653	0.515	0.0181
FLP	0.4999	0.7071	0.2655	0.5153	0.0181
LR	0.5676	0.7534	0.3301	0.5745	1.06×10^{-5}
SVR	0.5448	0.7381	0.3047	0.552	1.32×10^{-5}

Table 3. Performance comparison of different methods on WineQuality dataset with 15% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.1114	0.3338	0.0232	0.1524	0.0247
TGBR-PCA	0.0566	0.2380	0.0055	0.0744	0.0010
S3FSGL-2	0.5567	0.7461	0.3177	0.5637	0.0001
LapS3L	0.4593	0.6777	0.2286	0.4781	0.0236
FLP	0.4593	0.6777	0.2285	0.4781	0.0236
LR	0.5567	0.7461	0.3176	0.5636	2.02×10^{-5}
SVR	0.5446	0.7380	0.3045	0.5518	3.03×10^{-5}

Table 4. Performance comparison of different methods on Concrete dataset with 5% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.1138	0.3373	0.0204	0.1428	0.0062
TGBR-PCA	0.1142	0.3379	0.0203	0.1425	0.0049
S3FSGL-2	0.3563	0.5969	0.1548	0.3934	2.36×10^{-6}
LapS3L	0.3376	0.5811	0.1408	0.3752	0.0056
FLP	0.3376	0.5811	0.1408	0.3752	0.0056
LR	0.3563	0.5969	0.1548	0.3934	8.91×10^{-7}
SVR	0.3540	0.5950	0.1532	0.3914	1.47×10^{-8}

Table 5. Performance comparison of different methods on Concrete dataset with 10% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.1126	0.3355	0.0208	0.1443	0.0092
TGBR-PCA	0.1114	0.3338	0.0199	0.1411	0.0036
S3FSGL-2	0.3543	0.5953	0.1535	0.3918	4.95×10^{-6}
LapS3L	0.3188	0.5646	0.1280	0.3578	0.0098
FLP	0.3188	0.5646	0.1280	0.3578	0.0098
LR	0.3544	0.5953	0.1534	0.3917	1.51×10^{-6}
SVR	0.3539	0.5949	0.1531	0.3913	1.48×10^{-8}

Table 6. Performance comparison of different methods on Concrete dataset with 15% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.1158	0.3403	0.0218	0.1478	0.0115
TGBR-PCA	0.1056	0.3250	0.0176	0.1325	0.0044
S3FSGL-2	0.3514	0.5928	0.1511	0.3887	1.35×10^{-5}
LapS3L	0.2980	0.5459	0.1129	0.3360	0.0140
FLP	0.2980	0.5459	0.1129	0.3360	0.0140
LR	0.3517	0.5930	0.1512	0.3888	4.52×10^{-6}
SVR	0.3539	0.5949	0.1531	0.3913	6.14×10^{-8}

Table 7. Performance comparison of different methods on FG-NET dataset with 5% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0972	0.3118	0.0173	0.1315	0.0023
TGBR-PCA	0.0946	0.3075	0.0153	0.1239	0.0020
S3FSGL-2	0.1566	0.3958	0.0410	0.2025	1.1×10^{10}
LapS3L	0.1494	0.3865	0.0378	0.1943	0.0014
FLP	0.1479	0.3846	0.0373	0.1931	0.0014
LR	0.1501	0.3875	0.0380	0.1949	1.41×10^{-3}
SVR	0.1522	0.3902	0.0394	0.1986	7.5×10^{10}

Table 8. Performance comparison of different methods on FG-NET dataset with 10% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0984	0.3136	0.0187	0.1366	0.0019
TGBR-PCA	0.0893	0.2988	0.0154	0.1242	0.0013
S3FSGL-2	0.1555	0.3943	0.0406	0.2016	3.20×10^{10}
LapS3L	0.1425	0.3775	0.0354	0.1882	0.0021
FLP	0.1403	0.3746	0.0347	0.1863	0.0020
LR	0.1438	0.3792	0.0359	0.1893	2.11×10^{-3}
SVR	0.1500	0.3872	0.0387	0.1967	1.52×10^{-4}

Table 9. Performance comparison of different methods on FG-NET dataset with 15% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0973	0.3120	0.0183	0.1352	0.0024
TGBR-PCA	0.0847	0.2910	0.0136	0.1165	0.0017
S3FSGL-2	0.1543	0.3928	0.0402	0.2006	2.82×10^{10}
LapS3L	0.1343	0.3664	0.0319	0.1787	0.0033
FLP	0.1306	0.3614	0.0308	0.1756	0.0032
LR	0.1361	0.3690	0.0325	0.1803	3.35×10^{-3}
SVR	0.1472	0.3837	0.0377	0.1941	2.63×10^{-4}

Table 10. Performance comparison of different methods on FYN dataset with 5% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0928	0.3046	0.0130	0.1141	0.0156
TGBR-PCA	0.0918	0.3029	0.0120	0.1096	0.0047
S3FSGL-2	0.5324	0.7297	0.3042	0.5515	5.9×10^{-5}
LapS3L	0.5025	0.7089	0.2718	0.5213	0.0126
FLP	0.5026	0.7089	0.2719	0.5214	0.0126
LR	0.5283	0.7269	0.2980	0.5459	4.46×10^{-3}
SVR	0.5005	0.7075	0.2712	0.5207	2.6×10^{-7}

Table 11. Performance comparison of different methods on FYN dataset with 10% of discretized values (labeled).

	MAE	RMAE	MSE	RMSE	Variance
TGBR-t-SNE	0.0846	0.2908	0.0131	0.1144	0.0243
TGBR-PCA	0.0620	0.2490	0.0061	0.0782	0.0040
S3FSGL-2	0.5243	0.7241	0.2948	0.5430	1.26×10^{-4}
LapS3L	0.4651	0.6820	0.2341	0.4838	0.0212
FLP	0.4651	0.6820	0.2339	0.4836	0.0213
LR	0.5128	0.7161	0.2801	0.5293	8.94×10^{-3}
SVR	0.5003	0.7073	0.2709	0.5205	5.60×10^{-8}

Table 12. Performance comparison of different methods on FYN dataset with 15% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0878	0.2964	0.0151	0.1230	0.0305
TGBR-PCA	0.0617	0.2483	0.0062	0.0789	0.0035
S3FSGL-2	0.5150	0.7176	0.2842	0.5331	2.38×10^{-4}
LapS3L	0.4201	0.6482	0.1908	0.4368	0.0302
FLP	0.4210	0.6488	0.1913	0.4374	0.0307
LR	0.4959	0.7042	0.2601	0.5100	1.52×10^{-2}
SVR	0.5003	0.7073	0.2709	0.5205	1.32×10^{-7}

Table 13. Performance comparison of different methods on Wind dataset with 5% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0336	0.1833	0.0021	0.0458	0.0021
TGBR-PCA	0.0329	0.1815	0.0018	0.0430	0.0008
S3FSGL-2	0.1662	0.4077	0.0324	0.1799	1.38×10^{-6}
LapS3L	0.1572	0.3965	0.0290	0.1703	0.0015
FLP	0.1572	0.3965	0.0290	0.1703	0.0015
LR	0.1661	0.4076	0.0323	0.1798	2.87×10^{-6}
SVR	0.1635	0.4043	0.0315	0.1774	8.56×10^{-8}

Table 14. Performance comparison of different methods on Wind dataset with 10% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0351	0.1874	0.0023	0.0479	0.0026
TGBR-PCA	0.0307	0.1753	0.0016	0.0401	0.0006
S3FSGL-2	0.1646	0.4057	0.0318	0.1783	3.29×10^{-6}
LapS3L	0.1471	0.3836	0.0255	0.1598	0.0026
FLP	0.1471	0.3835	0.0255	0.1597	0.0026
LR	0.1645	0.4055	0.0317	0.1781	6.85×10^{-6}
SVR	0.1634	0.4043	0.0316	0.1774	6.00×10^{-7}

Table 15. Performance comparison of different methods on Wind dataset with 15% of discretized values (labeled).

	MAE	**RMAE**	**MSE**	**RMSE**	**Variance**
TGBR-t-SNE	0.0374	0.1934	0.0025	0.0502	0.0033
TGBR-PCA	0.0291	0.1706	0.0014	0.0380	0.0005
S3FSGL-2	0.1628	0.4035	0.0312	0.1766	6.16×10^{-6}
LapS3L	0.1371	0.3703	0.0223	0.1492	0.0036
FLP	0.1372	0.3704	0.0222	0.1491	0.0036
LR	0.1626	0.4032	0.0311	0.1762	1.31×10^{-5}
SVR	0.1634	0.4042	0.0314	0.1773	2.42×10^{-6}

The results show smaller errors compared to the methods found in the literature, especially the results of datasets WineQuality and Wind. Comparing only our method, the variance of the results with 10 and 15% of valued elements are much smaller compared to the test done with only 5%. Dimensionality reduction methods have very close results, which benefits the PCA method by having a faster process to be generated.

6. Conclusions

In this work, we address the problem of transductive regression in a semi-supervised context. In recent years, some graph-based methods have emerged to treat regression, but they do not discuss aspects such as: how the inference was performed, what type of learning is covered, how the initialization in the network is handled, and how to choose the elements that will be valued.

The dimensionality reduction factors, robust graphical walking techniques, and the construction of the network are good solutions to the regression problem in a semi-supervised context. Among these, the construction of the network and the selection of elements become interesting subjects to be further explored. The construction of the connections was a factor that improved the method operation, emphasizing that the reduction of dimensionality is important to reduce cost complexity.

We now intend to employ the TGBR in practical applications such as geolocated indicators. In addition, in future work, we plan to extend the TGBR in real networks, such as freshwater sensing networks in rivers. Another issue that deserves further attention is its application to real state valuation. There is room for further investigation. For instance, graph-based regression methods did not address the impact of different network construction strategies on the regression process.

Author Contributions
The authors contributed equally to this work.All authors have read and agreed to the published version of the manuscript.

Funding
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of Interest
The authors declare no conflicts of interest.

Data Availability Statement
All datasets used in this study are available at https://anonymous.4open.science/r/TGBR-DataSet/

Acknowledgments
This work was partially supported by the following agencies: Author Renan Guilherme Nespolo: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) under the process 88882.328833/2019-01 and Finance Code 001; author Alan Demétrius Baria Valejo: FAPESP under grant numbers 22/03090-0 and 21/06210-3 and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) Finance Code 001; author Alneu de Andrade Lopes: grants FAPESP 2020/09835-1, 2022/09091-8 and the Brazilian National Research Council (CNPq) research fellowships 303588/2022-5.

References

1. Lopes, A.A., Bertini, J.R., Motta, R., Zhao, L.: Classification based on the optimal k-associated network. In *Complex Sciences*; Zhou, J. (Ed.); pp. 1167–1177. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02466-5-117
2. Berton, L., de Andrade Lopes, A.: Neighborhood graph construction for semi-supervised learning. *AI Matters* 2(3), 14–15 (2016). https://doi.org/10.1145/2911172.2911176
3. Valejo, A., Ferreira, V., Fabbri, R., Oliveira, M.C.F.d., Lopes, A.d.A.: A critical survey of the multilevel method in complex networks. *ACM Computing Surveys (CSUR)* 53(2), 1–35 (2020), number: 2 Publisher: ACM New York, NY, USA
4. Valejo, A., de Oliveira, M.C.F., Geraldo Filho, P.R., de Andrade Lopes, A.: Multilevel approach for combinatorial optimization in bipartite network. *Knowledge-Based Systems* 151, 45–61 (2018), publisher: Elsevier
5. Xu, Z., King, I., Lyu, M.R., Jin, R.: Discriminative Semi-Supervised Feature Selection via Manifold Regularization. *IEEE Transactions on Neural Networks* 21(7), 1033–1047 (2010)
6. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (chapelle, o. et al., eds.; 2010). *IEEE Transactions on Neural Networks* 20(3), 542–542 (2010)
7. Rossi, R.G., de Paulo Faleiros, T., de Andrade Lopes, A., Rezende, S.O.: Inductive model generation for text categorization using a bipartite heterogeneous network. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*. pp. 1086–1091. IEEE (2012)
8. Fahrmeir, L., Kneib, T., Lang, S., Marx, B.D.: Regression models. In *Regression: Models, Methods and Applications*, pp. 23–84. Springer (2022)
9. Stańczyk, U., Zielosko, B., Jain, L.C. *Advances in Feature Selection for Data and Pattern Recognition*; Springer (2018)
10. Torgo, L.: Predicting Algae Blooms, pp. 39–94. chapman and hall/CRC, New York (2011). https://doi.org/10.1201/9780429292859
11. Ni, B., Yan, S., Kassim, A.: Learning a Propagable Graph for Semisupervised Learning: Classification and Regression. *IEEE Transactions on Knowledge and Data Engineering* 24(1), 114–126 (Jan 2012). https://doi.org/10.1109/TKDE.2010.209
12. Kraus, V., Benkabou, S., Benabdeslem, K., Cherqui, F.: An Improved Laplacian Semi-Supervised Regression. In *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. pp. 564–570 (Nov 2018). https://doi.org/10.1109/ICTAI.2018.00092
13. Sheikhpour, R., Sarram, M.A., Sheikhpour, E.: Semi-supervised sparse feature selection via graph Laplacian based scatter matrix for regression problems. *Information Sciences* 468, 14–28 (Nov 2018). https://doi.org/10.1016/j.ins.2018.08.035
14. Johansen, A.M.: Markov Chain Monte Carlo. In: Peterson, P., Baker, E., McGaw, B. (Eds.) *International Encyclopedia of Education*, (Third Edition), pp. 245–252. Elsevier, Oxford (Jan 2010). https://doi.org/10.1016/B978-0-08-044894-7.01347-6
15. Freedman, D.A. *Statistical Models: Theory and Practice*. cambridge university press (2009)
16. Haykin, S. *Spectral Graph Theory*, pp. 350–352. No. v. 10, New York: Prentice Hall,, New York (2009). https://doi.org/10.5555/541500
17. Van Der Maaten, L.: Accelerating T-SNE Using Tree-Based Algorithms. *The Journal of Machine Learning Research* 15(1), 3221–3245 (2014). https://doi.org/10.5555/2627435.2697068
18. Michalski, R.S., Carbonell, J.G., Mitchell, T.M. *Machine Learning: An Artificial Intelligence Approach*; Springer Science & Business Media (2013)

19. Silva, M.M., Maia, T.T., Braga, A.P.: An evolutionary approach to transduction in support vector machines. In *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*; pp. 6–pp. IEEE (2005)
20. Rossi, R.G., de Andrade Lopes, A., Rezende, S.O.: Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing & Management* **52**(2), 217–257 (2016). https://doi.org/10.1016/j.ipm.2015.07.004
21. Jensen, D., Neville, J., Gallagher, B.: Why collective inference improves relational classification. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 593–598 (2004). https://doi.org/10.1145/1014052.1014125
22. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. *Acm. Sigmod Record* 27(2), 307–318 (1998). https://doi.org/10.1145/276305.276332
23. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment* 2(1), 718–729 (2009). https://doi.org/10.14778/1687627.1687709
24. Getoor, L., Taskar, B. *Introduction to Statistical Relational Learning*; MIT press Cambridge, Cambridge, MA 02142 (2007). https://doi.org/10.7551/mitpress/7432.001.0001
25. Szummer, M., Jaakkola, T.: Partially Labeled Classification with Markov Random Walks. In *Advances in Neural Information Processing Systems*; pp. 945–952 (2002). https://doi.org/10.5555/2980539.2980661
26. Kobayashi, T., Watanabe, K., Otsu, N.: Logistic Label Propagation. *Pattern Recognition Letters* 33(5), 580–588 (2012)
27. Zhu, X., Lafferty, J., Rosenfeld, R.: Semi-Supervised Learning with Graphs. PhD Thesis, Carnegie Mellon University, Language Technologies Institute, school of Computer Science Pittsburgh, PA (2005)
28. Zhu, X. *Semi-Supervised Learning Literature Survey*; Computer Science, University of Wisconsin-Madison 2(3), 4 (2006)
29. Huang, T., Kecman, V., Kopriva, I. *Semi-Supervised Learning and Applications*, pp. 125–173. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/3-540-31689-2-5
30. Asuncion, A., Newman, D.: UCI Machine Learning Repository (2007)
31. Sheikhpour, R., Sarram, M.A., Rezaeian, M., Sheikhpour, E.: QSAR Modelling Using Combined Simple Competitive Learning Networks and RBF Neural Networks. *SAR and QSAR in Environmental Research* 29(4), 257–276 (2018). https://doi.org/10.1080/1062936X.2018.1424030
32. Zhang, J., Jiang, J., Han, Y.: Semi-supervised regression with optimized rank for matrix data classification. *IEEE Transactions on Cybernetics* 49(9), 3443–3456 (2019). https://doi.org/10.1109/TCYB.2018.2844860
33. Dornaika, F., Traboulsi, Y.E., Zhu, R.: Robust and Flexible Graph-Based Semi-Supervised Embedding. In *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*. pp. 465–470 (Aug 2018). https://doi.org/10.1109/ICPR.2018.8545239
34. Han, C., Cao, X., Stanojevic, M., Ghalwash, M., Obradovic, Z.: Temporal Graph Regression via Structure-Aware Intrinsic Representation Learning pp. 360–368 (May 2019). https://doi.org/10.1137/1.9781611975673.41
35. Pan, X., Li, G., Zheng, Y.: Ensemble transductive propagation network for semi-supervised few-shot learning. Entropy 26(2), 135 (2024)
36. Xu, X., Yang, Y., Deng, C., Nie, F.: Adaptive Graph Weighting for Multi-View Dimensionality Reduction. *Signal Processing* 165, 186–196 (Dec 2019). https://doi.org/10.1016/j.sigpro.2019.06.026
37. Doquire, G., Verleysen, M.: Graph laplacian for semi-supervised feature selection in regression problems pp. 248–255 (2011). https://doi.org/10.1007/978-3-642-21501-8-31
38. Doquire, G., Verleysen, M.: A Graph Laplacian Based Approach to Semi-Supervised Feature Selection for Regression Problems. *Neurocomputing* 121, 5–13 (2013). https://doi.org/10.1016/j.neucom.2012.10.028
39. Guo, Z., Ao, X., He, Q.: Transductive semi-supervised metric network for reject inference in credit scoring. *IEEE Transactions on Computational Social Systems* (2023)
40. Grady, L., Schiwietz, T., Aharon, S., Westermann, R.: Random Walks for Interactive Alpha-Matting. In *Proceedings of the VIIP*. vol. 2005, pp. 423–429 (2005). https://doi.org/10.5555/1659167
41. Zhu, X., Goldberg, A.B. *Graph-Based Semi-Supervised Learning*; pp. 43–55. Springer International Publishing, Cham (2009). https://doi.org/10.1007/97830310154895
42. Mihalcea, R., Radev, D. *Graph-Based Natural Language Processing and Information Retrieval*; Cambridge university press, Cambridge (2011). https://doi.org/10.1017/CBO9780511976247

43. Chen, J., Zeng, G.Q., Zhou, W., Du, W., Lu, K.D.: Wind speed forecasting using nonlinear learning ensemble of deep learning time series prediction and extremal optimization. *Energy Conversion and Management* 165, 681–695 (2018)
44. Cortes, C., Mohri, M.: On Transductive Regression. In: Advances in Neural Information Processing Systems. pp. 305–312 (2007). https://doi.org/10.5555/2976456.2976495

Author Biographies

Renan Guilherme Nespolo Graduated in Business IT Technology from college of technology of the state of São Paulo (2011) and master's at Computer Science from Universidade Estadual Paulista Júlio de Mesquita Filho (2016). Has experience in Computer Science, focusing on Graphical Processing (Graphics). He is currently PhD candidate at the University of São Paulo.

Alan Demétrius Baria Valejo Adjunct Professor at the Department of Computing at the Federal University of São Carlos (DC-UFSCar). Graduated with a Bachelor's degree in Informatics from ICMC-USP in 2012. He obtained a Master's and Doctorate in Computer Science and Computational Mathematics from ICMC-USP in 2014 and 2019, respectively. In 2020, he completed a Post-Doctorate at the University of São Paulo (FFCLRP-USP) through FAPESP. Accredited professor in the Postgraduate Program in Computer Science at the Department of Computing (PPGCC) at UFSCar. Researcher in the Data Mining and Applications research group (MIDAS) and the Center for Sociopolitical Studies of Algorithms and Artificial Intelligence (Interfaces). Coordinator of the Data Processing and Analysis Group at UFSCar (PANDA). He works in the area of Graph Machine Learning and is interested in problems related to text mining and large-scale social network analysis.

Alneu de Andrade Lopes Graduated in Civil Engineering from the Federal University of Mato Grosso do Sul (1985), a master's degree in Computer Science and Computational Mathematics from the University of São Paulo (1995) and a PhD in Computer Science from the University of Porto (2001). He is currently a professor at the University of São Paulo. He works in the area of Artificial Intelligence and Propositional and Relational Machine Learning, mainly on the following topics: Data Mining and Complex Network Mining.