UNIVERSITY DE SÃO PAULO
FACULTY OF SCIENCES AND LETTERS OF RIBEIRÃO PRETO
DEPARTMENT OF COMPUTING AND MATHEMATICS

ESTEBAN WILFREDO VILCA ZUÑIGA

Network-based high-level classification using betweenness centrality and attribute-attribute interaction

Classificação de alto nível baseada em redes usando betweenness centrality e interação atributo-atributo

Ribeirão Preto–SP

2021

ESTEBAN WILFREDO VILCA ZUÑIGA

Network-based high-level classification using betweenness centrality and attribute-attribute interaction

Original Version

Dissertation presented to Faculty of Sciences and Letters of Ribeirão Preto (FFCLRP) from the University de São Paulo (USP), as part of the requirements to hold the Master of Science degree.

Field of Study: Applied Computing.

Supervisor: Zhao Liang

Ribeirão Preto–SP

2021

Esteban Wilfredo Vilca Zuñiga
Network-based high-level classification using betweenness centrality and attribute-attribute interaction. Ribeirão Preto–SP, 2021.
66p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras
de Ribeirão Preto da USP, como parte das exigências para
a obtenção do título de Mestre em Ciências,
Área: Computação Aplicada.

Supervisor: Zhao Liang

1. Complex Networks. 2. Supervised Learning. 3. High-level classification.

Esteban Wilfredo Vilca Zuñiga

Network-based high-level classification using betweenness centrality and attribute-attribute interaction

Modelo canônico de trabalho monográfico acadêmico em conformidade com as normas ABNT.

Supervisor:
Orientador

Professor
Convidado 1

Professor
Convidado 2

Ribeirão Preto–SP
2021

This is dedicated to my family for their support in my journey, to my brothers as an example of dedication and rebellion against the obstacles of life, to my smart students on YouTube, and to my beauty partner for their belief in our rocky relationship that survived the distance and this pandemic.

Acknowledgements

I would like to acknowledge the University of Sao Paulo for its high-quality education in novel technologies like machine learning, complex networks, and data science.

I thank the government of Brazil for their economical help through the CAPES scholarship that supports my research and my paper publications.

Finally, my deepest thanks to my supervisor Zhao Liang for their guidance, help, and patient in my research.

"Love your neighbor as yourself. No other commandment is greater than these.
(Holy Bible, Mark 12, 31)

Resumo

A democratização da tecnologia proporcionada pela internet, tecnologia em nuvem e mídia social aumentou drasticamente a quantidade de dados coletados. Aprendizado de máquina é um campo de inteligência artificial que produz informações valiosas a partir de dados. Especificamente, aprendizado supervisionado é um tipo de aprendizado de máquina que se concentra no uso de dados rotulados para aprender a prever o rótulo de dados futuros. Neste tópico, os algoritmos de classificação de alto nível usam a estrutura e relação entre os dados para classificar em vez de atributos físicos como distância. Redes complexas são uma estrutura de dados que fornece métricas para avaliar os dados como um sistema. Eles fornecem medidas para a conectividade, comunicabilidade ou dispersão da interação de dados. Neste estudo, exploramos propriedades de rede complexas e interação atributo-atributo para desenvolver novas técnicas de classificação de alto nível. Em primeiro lugar, aplicamos métrica de betweenness centrality, que captura características globais e locais de rede. Este método reduz o número de métricas avaliadas e executa uma melhoria na accuracy em comparação com outros algoritmos de alto nível. Em seguida, exploramos uma nova metodologia de construção de rede complexa para capturar medidas estruturais usando a interação atributo-atributo. Essa interação constrói e avalia cada atributo independentemente e os combina usando uma equação ponderada otimizada. Finalmente, analisamos os resultados obtidos por essas métricas em conjuntos de dados sintéticos e reais e os comparamos com outros algoritmos clássicos de baixo e alto nível. As técnicas propostas apresentam algumas características promissoras como a redução das métricas utilizadas para classificação, resiliência diante de dados não normalizados e uma nova métrica de avaliação de rede derivada da metodologia de construção.

Palavras-chave: aprendizado de máquina, aprendizado supervisionado, classificação de alto nível, redes complexas.

Abstract

The democratization of technology caused by the internet, cloud technology, social media increase dramatically the quantity of data collected. Machine learning is an artificial intelligence field that produces valuable information from data. Supervised learning is a type of machine learning that focuses on using labeled data to learn to predict the label of future data. In this area, the High-level classification algorithms use the structure of the relation between the data for classification instead of physical attributes like distance. Complex networks are a data structure that provides metrics to evaluate the data as a system. They provide measures to the connectivity, communicability, or sparseness of the data interaction. In this study, we explore complex network properties and attribute-attribute interaction to develop new high-level classification techniques. Firstly, we exploit a mixed metric betweenness centrality that captures global and local characteristics for classification. This method reduces the number of metrics evaluated and performs an improvement compared to other high-level algorithms. Then, we explore a new complex network building methodology to capture structural measures using attribute-attribute interaction. This interaction builds and evaluates each attribute independently and combining them using an optimized weighted equation. Finally, we analyze the results obtained by these metrics in synthetic and real datasets and compare them to other classical low-level and high-level algorithms. The proposed techniques present some promising characteristics as the reduction of metrics used for classification, resilience in front of non-normalized data, and a new network evaluation metric derived from the building methodology.

Keywords: machine learning, supervised learning, high-level classification, complex networks.

List of figures

List of tables

List of abbreviations and acronyms

NBHL	Network-Based high-level classification method
NBHL-BC	Network-Based high-level classification method using betweenness centrality
kNN	k nearest neighbors
ϵ-$radius$	Radius neighbors
DT	Decision tree
MLP	Multi layer perceptron
RF	Random forest

List of symbols

G	Network measure
I	Impact measure
G	Graph
V	Nodes
E	Edges
$\mathcal{N}$	Neighborhood
k	Node degree
D	Density
Q	Modularity
B	Betweenness centrality

Summary

1

Introduction

1.1 Context

In the last years, we live a revolution in technology. Now, we can collect tons of data using cloud services . However, the transformation of this data into valuable information is still an challenge problem (ALBRIGHT; WINSTON, 2015) (WITTEN et al., 2016).

Machine learning uses math and computational resources through simple or complex transformations to help on decision making in several topics like churn prediction, diagnosis, spam identification, price prediction, client recommendation, and so on (KREYENHAGEN et al., 2014) (KELLEHER; NAMEE; D'ARCY, 2015) (PATIL et al., 2017) (ANUPRIYA et al., 2018). Machine learning can be defined as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict the future data, or perform other kinds of decision making under uncertainty (MURPHY, 2013).

Usually, machine learning is divided into three main paradigms: *supervised learning*, *unsupervised learning* and *semi-supervised learning* (GéRON, 2017). Supervised learning uses labeled data to detect patterns and predict future cases. According to the kind of labels, the prediction is called *classification* for categorical labels and *regression* for numerical labels. Unsupervised learning focuses on the analysis of data without a label. It is used in clustering, visualization, dimensionality reduction, association rule learning, etc. (GéRON, 2017). Semi-supervised learning is a combination of the last two types of paradigms. It considers a small number of labeled data and a large quantity of non-labeled data to predict the labels of new examples (OLIVIER; BERNHARD; ALEXANDER, 2006).

One of the major topics to be investigated in supervised learning is data classification. It aims at generating a map from the input data to the corresponding desired output, for a given training set. The constructed map, called a classifier, is used to predict new input instances. It can be applied to solve a large amount of real-world problems, such

as recommendation system, for example, personalized cloth recommendations according to brands (KREYENHAGEN et al., 2014), predicting witch costumer who will stop buying on a company (churn prediction) (PATIL et al., 2017), credit card fraud detection (ANUPRIYA et al., 2018), image and video recognition, medical diagnoses, just to name a few.

Many classification techniques have been developed (WŁADYSŁAW; WITOLD, 2018) (GOPINATH; AJAY; SANJAY, 2019) (HIMANSHU, 2019), such as k-Nearest Neighbors (kNN), Bayesian decision theory, neural networks (Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), Radial Basis Function (RBF), decision trees (DT), neural networks (NN), and so on. In essence, all these techniques train labeled data items and consequently classify unlabeled data items according to the physical features (e.g., distance, similarity or distribution) of the input data. These kind of techniques divide the training data space into sub-spaces, each one corresponds to a class. The shapes of the sub-spaces should be as regular as possible. Complex shapes, such as twisted patterns, are not permitted or very hard to be determined. The techniques that predict class labels using only physical features are called *low-level* classification techniques (SILVA; ZHAO, 2012). Human (animal) brain performs both low and high orders of learning and it has facility of identifying patterns according to the semantic meanings of the input data. In general, however, this kind of task is still hard to be performed by computers. Data classification by considering not only physical attributes but also pattern formation is referred to as *high-level* classification (SILVA; ZHAO, 2012) (SILVA; ZHAO, 2015) (COLLIRI et al., 2018) (CARNEIRO; ZHAO, 2018b) (CARNEIRO et al., 2019).

The high-level classification techniques developed so far transform training data set into complex networks and use network topology to represent data patterns. Specifically, the classification is performed by verifying whether a testing data instance conforms a data pattern of a class presented in the training set, i.e., a testing instance will be classified to a class, say class C, which its insertion to the sub-network formed by data items of class C, results in a smaller change of the sub-network structure than its insertion to the sub-networks of other classes. High-level classification through pattern formation has already been shown useful in several applications. In the papers (FERREIRA; ZHAO, 2016) and (SOUTO et al., 2019), time series trends are modeled by network communities to estimate financial market risks. In the works (BACKES; CASANOVA; BRUNO, 2013) and (LIMA et al., 2019), the authors use network patterns to represent shapes of images to enhance the classification accuracy. In the papers (COCA; ZHAO, 2016a) and (COCA; ZHAO, 2016b), the authors use the pattern formation in networks to identify musical rhythm and musical scales. These investigations show that the high-level data classification algorithms are important to capture patterns in a wide range of data, such as time series and images.

Complex networks are large scale graphs with nontrivial connection patterns (ALBERT; BARABÁSI, 2002). Complex networks have the ability to describe spatial, functional, and topological relations among the elements (vertices and links). Such a research field has become one of the major themes in complex systems and has been used to study a wide range of problems (VITO; VINCENZO; GIOVANNI, 2017) (MARIA et al., 2019) (SEAN et al., 2019) (JING; HUSSEIN; KAY, 2019) (MILOš; MIRJANA; LAKHMI, 2019). Many network measures have been defined and each one of them characterizes network structure from a particular viewpoint. In the category of degree-related measures, we have *degree distribution*, *degree-degree correlation measures* like *density* that represents how strong the nodes connections are (SILVA; ZHAO, 2016), *assortativity* degree that represents the attraction of nodes with a similar degree (high with high degree and low with low degree) (NEWMAN, 2003), and *normalized rich-club coefficient* that measure the connection strength of hubs (ZHOU; MONDRAGON, 2004). In the category of *structural measures*, we have *clustering coefficient* that represents the connection strength between the neighbors of a node (WATTS; STROGATZ, 1998), *modularity* that measure the connectivity of a portion of the network (CLAUSET et al., 2004), and *cyclic coefficient* that determinate the degree of node circulation (KIM; KIM, 2005). In the category of *centraliy measures*, we have *betweenness* that measure the node importance for communication on the network (FREEMAN, 1977), *closeness vitality* that measure the impact of a network communication if a node is removed (SILVA; ZHAO, 2016), and so on.

There are three main parts in the high-level classification algorithms: the complex network building technique, the graph measures evaluation for classification method, and the optimization phase. In the first phase, there are some building methodologies like kNNG (method based on k nearest neighbors), kNN+ϵNG(method based on knn and e-radius technique). In the second phase, there are a variety of metrics and methods to exploit the network structure like importance measure used in (CARNEIRO; ZHAO, 2017), impact measure used in (COLLIRI et al., 2018), or link prediction used in (FADAEE; HAERI, 2019). In the third phase, we have and optimization phase to improve the original structure like in (CARNEIRO et al., 2019) using a social learning particle swarm optimization (SL-PSO), or as a metric (CHIRE-SAIRE, 2020) using ant colony optimization.

The current research provides a new methodology that explores the three above-mentioned points: Firstly, we use betweenness centrality as a classification metric measuring the communicability of a network once inserted a new data. Second, we introduce the concept of attribute-attribute interaction to produce a complex network for each attribute to evaluate them independently and evaluate each one using modularity to remove the networks that provide noise to the classification process. Finally, we introduce an optimization method to combine each network prediction as a weighted equation using particle swarm optimization.

1.2 Motivation and Objectives

The principal motivation of this research is to explore and exploit the concepts related to the structure, characteristics, and patterns generated by the data as a system. As we will explain in the last section, this field has many applications like business decisions, music generation, time series analysis, etc. So, the research focused on the improvement of high-level classification techniques is an important task. The proposed technique introduces new concepts to the current literature like the next ones:

- Building a network for each attribute exploits the attribute-attribute interaction. Following this method, we avoid compressing all the information of one instance into one single node, but we will have a node for every single piece of data.
- Using betweenness centrality, we use just one metric to capture the local and global characteristics of the networks.
- We introduce particle swarm optimization to reduce the impact of noisy networks.

These points introduce an improvement in each part of classification methodology: building complex networks, evaluation metrics, and optimization.

The main objective in this research is the development and evaluation of a new high-level classification technique that exploits the attribute-attribute interaction, uses a mixed metric to capture local and global network properties, and optimize the participation of each network into the classification.

1.3 Organization

This research presents the subsequent chapters:

- In chapter 2, titled "Review of Relevant Concepts and Techniques", we introduce three main concepts. First, we explore the complex networks and their main properties. Then, we describe the building techniques proposed in the literature. Finally, we review the current high-level classification algorithms.
- In chapter 3, titled "Proposed Models", we explain the steps of the proposed algorithms. First, we describe the use of betweenness centrality as a classification metric. We explore the attribute-attribute interaction. Finally, we describe the optimization of each network participation in the classification process.

- In chapter 4, titled "Results", we present the results obtained by the algorithm in synthetic and existing data sets and compare them in front of other classical and high-level algorithms.
- In chapter 5, titled "Conclusions", we draw some conclusions and point out further works.

2

Review of Relevant Concepts and Techniques

In the chapter 1, we have talked about high-level classification algorithms. Those algorithms generate a complex network from training and testing data and use the network topology to characterize data patterns for classification. In this chapter, we give a short review on complex networks, network building methodologies, and high-level classification methods.

2.1 Complex Networks

Data is irrelevant if we do not transform it into information. First, we need a data structure to describe and capture the behavior of the data interaction. Complex networks exploit these characteristics. They are graphs with a non-trivial structure. Usually, they are used to describe systems like the Internet, flight routes, country roads, people relationship, and more. We will describe some of the principal metrics used in this research.

2.1.1 Basic concepts

A graph is represented by $G = (V, E)$ where V is a set of vertices or nodes of the graph and E is a set of edges. Each node is usually represented by a number $i = 1, 2, 3, ..., n$ where n is the size of V and each edge $E \subseteq \{(i, j)|i, j \in V\}$ connects a pair of nodes (i, j). There are some other special characteristics:

- Undirected graph: The graph presents a symmetric relationship between edges, i.e., if there is a link (i, j), then exists (j, i).
- Directed graph: The graph does not present a symmetric relationship between edges. If there is a link (i, j), not necessarily exists (j, i).

- Weighted graph: The graph presents an additional numeric measure in their links $W(i, j)$, where for each $(i, j) \in E$, there is a weight $W(i, j)$.

There is some important basic metrics related to this graphs:

- Neighborhood of a node ($\mathcal{N}$): It is the set of nodes that are connected to a specific node i.

$$\mathcal{N}(i) = \{j : (i, j) \in E\}$$

- Degree of a node (k): It is the number of nodes on the neighborhood of a specific node i.

$$k_i = |\mathcal{N}(i)|$$

- Average degree ($\hat{k}$): It is the average of all the degree nodes on the network.

$$\hat{k} = \frac{1}{|V|} \sum_{i \in V} k_i$$

.

2.1.2 Complex network models

As we mentioned before, complex networks are large-scale graphs with non-trivial connection patterns. Some well-known complex network models are:

- Random networks: These are networks with random connections, which start with nodes without links and, for each node, a link is created with a probability p to another node (ERDöS; RéNYI, 1959).
- Small-world networks: Many networks, like social networks, have quick information spreading feature, known as small-world property. Such kinds of networks can be generated using a regular structure and a probability p to relocate a link between nodes (BARABáSI; PóSFAI, 2016). In the regular network, the average distance between nodes is high. After a few link relocations, the average distance is drastically reduced, while the local structure of the regular network is still maintained. This behavior is called the small-world effect.
- Scale-Free networks: Some systems present a concentration of links in small groups of nodes called hubs (BARABASI; ALBERT, 1999). Usually, this feature presents in Internet, where some web pages are very well connected like Google or Facebook.

Complex networks are powerful structures to represent and simulate the real structure of complex systems.

2.1.3 Complex network measures

Many measures have been developed to describe complex networks from different viewpoints. We here briefly review the following ones, which are related to the present project.

- Density: this measure represents the force of connection between the nodes (SILVA; ZHAO, 2016).

$$D = \frac{2|E|}{|V|(|V|-1)}$$

  Where $|E|$ and $|V|$ represent the number of edges and nodes respectively. This measure is in the interval $[0, 1]$. Where 0 means that the undirected graph G is an empty graph (without edges) and 1 means that G is complete (all nodes are fully connected).

- Assortativity: It represents the preference of the nodes to be connected with nodes of the same degree (NEWMAN, 2003).

$$r = \frac{|E|^{-1}\sum_{e\in E} u_e v_e - [\frac{|E|^{-1}}{2}\sum_{e\in E}(u_e+v_e)]^2}{\frac{|E|^{-1}}{2}\sum_{e\in E}(u_e^2+v_e^2) - [\frac{|E|^{-1}}{2}\sum_{e\in E}(u_e+v_e)]^2}$$

  Where u_e and v_e are the degrees of the two extreme nodes of the edge e. The value of this measure varies between -1 and 1, where a low value means a high relation between nodes of different degrees and a high represents a strong relation between nodes of equal degree.

- PageRank: It is a measure to rank the importance of a node. It was used by Google to rank the web pages (PERRA; FORTUNATO, 2008).

$$p(i) = \frac{q}{V} + (1-q)\sum_{j\in V: j->i} \frac{p(j)}{k_j^{(out)}} \tag{2.1}$$

  Where $p(i)$ is the pagerank value of the node i,$k_j^{(out)}$ is the out-degree of node j, and $j \in V : j- > i$ represents all the nodes pointing i. The constant $q \in [0, 1]$ is a probability of random walks and random jumps.

- Modularity: It represents the community structure of a subset of the network (CLAUSET et al., 2004).

$$Q = \frac{1}{2|E|}\sum_{i,j\epsilon V}(A_{ij} - \frac{k_i k_j}{2|E|})$$

Where A_{ij} is the weight of the edge that links the vertex i and j. The modularity is in the interval $[0, 1]$ where 0 means a lack of community structure and 1 means a completely modular structure.

- Betweenness Centrality (BC): This metric capture the communication between nodes using the shortest paths (SILVA; ZHAO, 2016). For each node we will calculate the number of geodesic path where this node is present. A node with higher BC present an important role in the network communication.

$$B(i) = \sum_{s \neq i \in \mathcal{V}} \sum_{t \neq i \in \mathcal{V}} \frac{\eta_{st}^{i}}{\eta_{st}} \tag{2.2}$$

where η_{st}^{i} is 1 when the node i is part of the geodesic path from s to t and 0 otherwise. η_{st} is the total number of shortest paths between s and t.

2.2 Building methodology

In order to capture the instance interactions, we need to represent the data in a network structure. Different authors present building methodologies using the k nearest neighbors algorithm (CARNEIRO; ZHAO, 2018a), which transform each instance into a node and the k nearest neighbors of node i are connected to i.

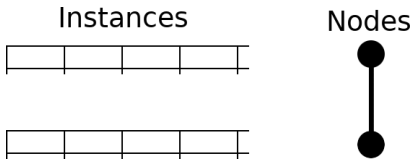


Figure 1 – Figure of two related instances transformed in two linked nodes.

kNN is extensively used in the complex network building methodologies (SILVA; ZHAO, 2012)(SILVA; ZHAO, 2015)(COLLIRI et al., 2018)(FADAEE; HAERI, 2019). Given a training data set, a complex network is generated for each class of data. Each data instance is mapped to be a vertex. The edges are generated according to the similarity between each pair of data instance. Specifically, the following rule was introduced to establish edges (SILVA; ZHAO, 2012)(SILVA; ZHAO, 2015)(COLLIRI et al., 2018):

$$\mathcal{N}(x_i) = \begin{cases} \epsilon\text{-}radius(x_i, y_i), & \text{if } |\epsilon\text{-}radius(x_i, y_i)| > k. \\ kNN(x_i, y_i), & \text{otherwise.} \end{cases} \tag{2.3}$$

where (x_i, y_i) represents a pair of data instance x_i and its corresponding label y_i. For each vertex x_i, $N(x_i)$ is the set of vertices to be connected to it. $\epsilon\text{-}radius(x_i, y_i)$ returns the set of vertices $\{x_j, j \in \mathcal{V} : d(x_i, x_j) < \epsilon \wedge y_i = y_j\}$ i.e. the set of vertices x_j whose

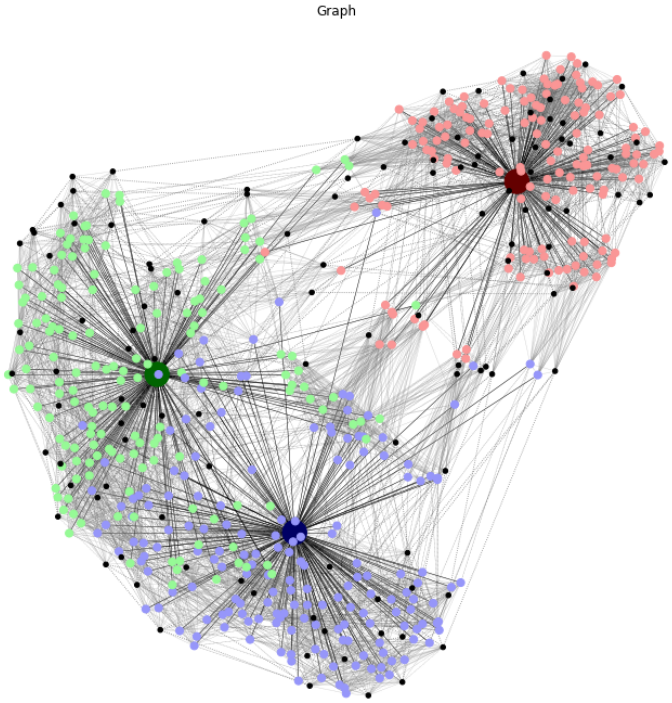


Figure 2 – Figure of the UCI Iris Dataset transformed into complex networks.

similarity with x_i is beyond a predefined value ϵ and have the same class label y_i. Here, d is a distance function (e.g. euclidean distance). $kNN(x_i, y_i)$ returns the set containing the k nearest neighbors of x_i. Note that the ϵ-radius criteria is used for dense regions ($|\epsilon - radius(xi)| > k$), while the kNN is employed for sparse regions. With this mechanism, it is expected that each class will have a network component (SILVA; ZHAO, 2012) (SILVA; ZHAO, 2015) (COLLIRI et al., 2018). Another technique (FADAEE; HAERI, 2019) shows us that we can combine all networks adding a class node for each label value and connect to them the training nodes. Once the networks are generated, we can use Eq. (3.1) again to insert the testing instance into the networks.

In Figure 2, we can observe the iris data set transformed into a complex network, where each class is transformed into a network (red, green, and blue). The complexity of the graph is high even if the number of instances and attributes in the iris dataset is small.

2.3 Network-based classification Methods

There are many ways to exploit this structure to build a classification model. We will explore some of them related to our research.

- Calculate the distance, structure and centrality measures of the network after and

before the testing nodes insertion and evaluate the fluctuation that they creates on each component network. This is called impact measure.

$$I_i^{(j)}(x)(u) = \triangle G_i^{(j)}(u) p^{(j)} \tag{2.4}$$

where $\triangle G_i^{(j)}(u)$ is the variation of the measure u of the network j caused by the test instance i. The component network that present the minimum impact will be the network class of the test node. The metrics used in the article (COLLIRI et al., 2018) were average degree, assortativity, average shortest path length and second moment of degree distribution. The method presents a good performance but the use of many global metrics affects the execution time of the algorithm. If we want to use the attribute-instance as a node, the size of the network will increase dramatically and the processing time also.

- Using the Page Rank technique on the training networks to predict the testing nodes based on the importance of the neighbors was proposed on (CARNEIRO; ZHAO, 2018b).

$$I_j^{(t+1)} = \sum_{i->j} \beta \frac{I_i^{(t)}}{d_i} + (1-\beta)\frac{1}{n} \tag{2.5}$$

where $I_j^{(t+1)}$ is the importance measure of a node on time $t+1$ ($I_j^{(0)} = \frac{1}{n}$), β is a constant (0.85), n is the total number of nodes. The value of t is defined by a threshold value. According to this measure, the test node will be classify where it has more important neighbors.

- Using a class node and connecting all the training nodes to their respective ones. In this way, it is possible to predict the label of a testing node using *link prediction* that measures the probability of a link creation between two nodes. In this case, this measure will be calculated on the edges between test and training nodes.

$$\lambda_{i,j} = \sum_{\lambda \in \Gamma_i \cap \Gamma_j} \frac{1}{log|\Gamma_\lambda|} \tag{2.6}$$

$$\lambda_{i,j} = \sum_{\lambda \in \Gamma_i \cap \Gamma_j} \frac{1}{|\Gamma_\lambda| - |\Gamma_\lambda \cap \Gamma_j|} + \frac{1}{|\Gamma_\lambda| - |\Gamma_\lambda \cap \Gamma_i|} \tag{2.7}$$

The equations 2.6 & 2.7 give a probability according to the number the neighbors in common of two nodes(test and training)(FADAEE; HAERI, 2019). This is a local measure and the time consuming is lower than the used for global measures.

- Additionally, many strategies combine the classification method with classical algorithms like SVM, kNN, CART, etc.

$$M_i^{(j)} = (1-\lambda)T_i^{(j)} + \lambda C_i^{(j)} \tag{2.8}$$

where $T_i^{(j)}$ is the probability that an instance x_i is part of the class j produced by an arbitrary traditional (low level) classifier, $C_i^{(j)}$ represents the same membership function produced by the high-level algorithm, and λ is the constant called *compliance term* that balance the two classification decisions.

There are others main concepts that could improve the performance of the high-level classification algorithms:

1. The optimization proposed on (CARNEIRO et al., 2019) use particle swarm to remove some links of the k-nearest neighbors that are noisy. This particular technique will be vital to reduce the number of links on a large network.

2. The detection of community on a multi-layer network proposed on (GAO et al., 2019) could help us to find some common structures on attribute networks, identify important edges and detect noisy networks without community structure.

3

The Proposed Methods

In this chapter, we propose three new improvements to the high-level classification techniques: 1) Using betweenness centrality to generate a new classification method; 2) Presenting an Ensemble Method for Complex Network Building; 3) proposing an Evaluation Metric Method - Lost Links. The explanation of each of these new methods will be exposed in this chapter and the experimental results will be shown in the next chapter.

3.1 High-Level Classification Based on Betweenness Centrality

In this section, we present the Network-Based High-level Classification method using Betweenness Centrality (NBHL-BC).

3.1.1 Mixed network measure

Several network measures have been applied to high-level classification, like clustering coefficient, assortativity, average degree. However, there are mixed measures that can capture local and global features by a unique measure, like betweenness centrality. Other mixed measures, like PageRank, have been applied to high-level classification provide promising results. Here, we will present a new method using betweenness centrality.

3.1.2 Overview of the Model

Each complex network consists of a set of nodes or vertices $\mathcal{V}$ and a set of links or edges $\mathcal{E}$ between each pair of nodes. The input data $\mathcal{D}$ of N elements for *supervised learning*

contains two parts: the attributes $\mathcal{X}$ and the labels $\mathcal{Y}$.

In *supervised learning*, the dataset $\mathcal{D} = \{(X_1, y_1), ..., (X_n, y_n)\}$ where $X_i = (x_1, ..., x_d)$ represents the d attributes, and y_i represents the label of the instance X_i. The values of $y_i \in \mathcal{L} = \{l_1, ..., l_c\}$ where $\mathcal{L}$ is the possible labels of the instance. The goal of *supervised learning* is to predict the y_i values using the instances X_i. This could be considered as function approximation $f(X_i) \approx y_i$ where the function f is our algorithm. To evaluate the model, it is required to split the data in training and testing datasets. The $X_{training}$ dataset will be used to build our model and the $X_{testing}$ dataset will be used for evaluation.

In the training phase, we will build complex networks using the training dataset. The instances in the dataset will be the nodes and the links will represent the similarity between these nodes. Therefore, we will have $\mathcal{D} \mapsto \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V} = \{1, ..., N\}$ is the set of nodes and $\mathcal{E}$ is the set of links in the complex network $\mathcal{G}$. The links could be created using kNN and $\epsilon - radius$ or personalized relation metrics like friendship on social data, flight routes, or city connections.

The network $\mathcal{G}$ will be built using $X_{training}$ to produce the nodes $\mathcal{V}$ and kNN and $e - radius$ as relation metric for links $\mathcal{E}$. Then, we remove the links between nodes with different labels y_i. Following this strategy, we will have one network component $\mathcal{G}^i$ for each label in $\mathcal{L}$.

In the testing phase, we insert a node from $X_{testing}$ into each component $\mathcal{G}^i$ following the same kNN and $e - radius$ rules of training phase. Then, we calculate the *betweenness measure* of this node in each $\mathcal{G}^i$. This measure is compared to the others from each network component $\mathcal{G}^i$. So, the differences are saved in a new list for each $\mathcal{G}^i$.

Finally, we get the average of the b lowest values for each list and we classify the new node to the $\mathcal{G}^i$ with the lowest average. Then, we remove this node from the other components. In the case that the average differences of two or more lists are equal, we use the number of links connected to this new node in each component as a second difference measure. Moreover, we use a ratio α to combine this high-level method with other techniques; in our case, we use the number of links connected to the new node inserted. This ratio is between 0 when we use only the number of links and 1 when we use just our method.

3.1.3 Network-Based High-level Classification Using Betweenness Centrality (NBHL-BC)

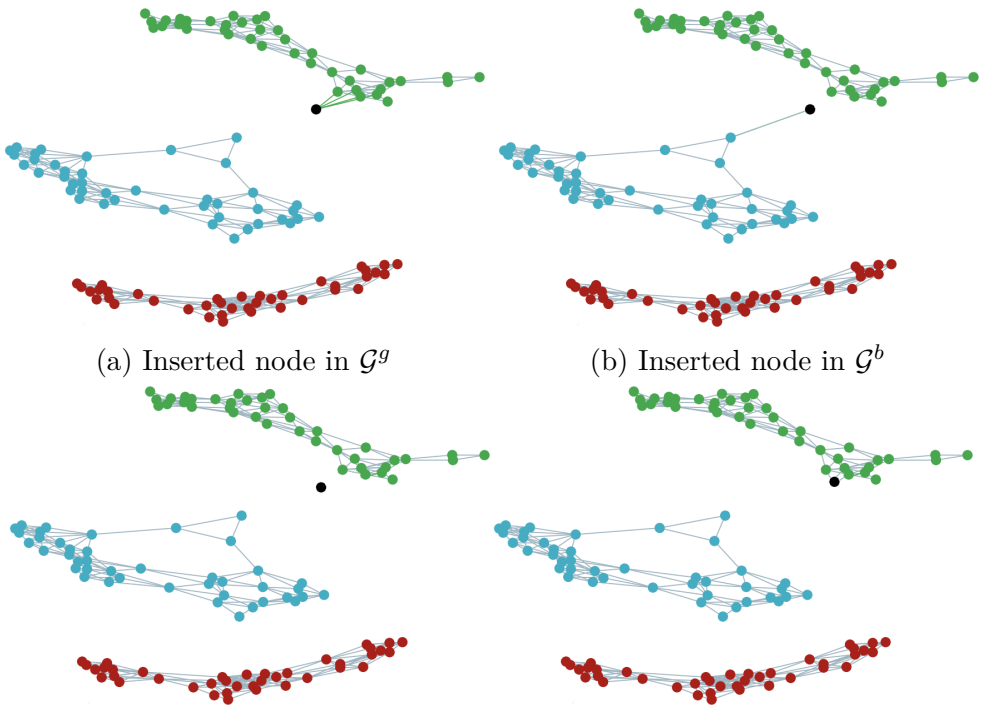
The proposed high-level classification algorithm, which will be referred as NBHL-BC, has four parameters k, e, b, and α. Where k is the number of neighbors used in the kNN,

e is the percentile into $kNN_{distances}$ used to calculate ϵ, b is the number of nodes with similar *betweenness* used for classification, and α is the weight to balance between links and *betweenness centrality*.

During the training phase, we build the network using the $X_{training}$ and $Y_{training}$ where $X_{training} \mapsto \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$. Each node in $\mathcal{V}$ is related with one instance in $X_{training}$ and each link in $\mathcal{E}$ is defined following these two techniques:

$$\mathcal{N}(X_i) = \begin{cases} \epsilon\text{-}radius(X_i, y_i), & \text{if } |\epsilon\text{-}radius(X_i, y_i)| > k \\ kNN(X_i, y_i), & \text{otherwise} \end{cases} \tag{3.1}$$

where (X_i, y_i) represents a pair of data instance X_i and its corresponding label y_i. For each instance X_i, $\mathcal{N}(X_i)$ is the set of nodes to be connected to it, its neighborhood. $\epsilon\text{-}radius(X_i, y_i)$ returns the set of nodes $\{X_j, j \in \mathcal{V} : distance(X_i, X_j) < \epsilon \wedge y_i = y_j\}$ i.e. the set of nodes X_j whose similarity with X_i is beyond a predefined value ϵ and have the same class label y_i. Here, *distance* is a similarity function like euclidean distance. $kNN(X_i, y_i)$ returns the set containing the k nearest neighbors of X_i. The value ϵ is the percentile e of the $kNN_{distances}$ in the sub graph of y_i. Note that the ϵ-radius criteria is used for dense regions ($|\epsilon - radius(Xi)| > k$), while the kNN is employed for sparse regions. With this mechanism, it is expected that each label will have an independent sub graph $\mathcal{G}^c$ (SILVA; ZHAO, 2012) (SILVA; ZHAO, 2015) (COLLIRI et al., 2018).



(a) Inserted node in $\mathcal{G}^g$ (b) Inserted node in $\mathcal{G}^b$

(c) Inserted node in $\mathcal{G}^r$ (d) Final Classification

Figure 3 – Figure of classification of a new instance (dark node) into the iris dataset graph $\mathcal{G}$ using with $k = 5$ $e = 0.5$ $b = 5$ $\alpha = 1.0$

In Figure 3d, we can see the network $\mathcal{G} = \{\mathcal{G}^r, \mathcal{G}^g, \mathcal{G}^b\}$ where r, g, b represent the sub graphs with nodes red, green and blue. At the testing phase, we insert each $X_{testing}$ instance (dark node) to each component $\mathcal{G}^i$ following the same rule defined by equation (3.1), and assuming that the node will be inserted in each sub-network.

Figure 3 shows an example where there are three network components $\mathcal{G}^i$ and the node to be tested is inserted to each one. In Figures 3a 3b 3c, the testing node uses its k-nearest neighbors with the same label. In this case with $k = 5$, there are 4 nodes in $\mathcal{G}^g$, 1 in $\mathcal{G}^b$, and 0 in $\mathcal{G}^r$. The $\epsilon - radius$ with the $e = 0.5$ (median of $kNN_{distances}$) is less than 5, because the current inserted node presents a sparse behavior; for this reason, we will use just kNN. Moreover, due to the condition of the same label, the algorithm will produce one sub-network for each possible label.

Now we calculate the *betweenness centrality* for the node in each component when the new node is inserted. Following this rule, the inserted node will have different values for each component.

The *betweenness centrality* is a mixed measure (global and local) that captures how much a given node is in the shortest paths of others nodes (SILVA; ZHAO, 2016). This measure captures the influence of a node in the communication of the network (NEEDHAM; HODLER, 2019). We capture not only the characteristics of a node but also the behavior of their neighborhood. So, we have a metric that provides local and global network characteristics. This metrics is defined in the equation 3.2.

$$B(i) = \sum_{s \neq i \in \mathcal{V}} \sum_{t \neq i \in \mathcal{V}} \frac{\eta_{st}^i}{\eta_{st}} \tag{3.2}$$

where η_{st}^i is 1 when the node i is part of the geodesic path from s to t and 0 otherwise. η_{st} is the total number of shortest paths between s and t.

Then, we calculate the difference of this measure between the inserted node and the other nodes in each component $\mathcal{G}^i$. The Line 14 of the algorithm 2 shows this step. In the algorithm 1, we can appreciate how an inserted node will present a different *betweenness centrality* for each sub graph.

These values will be inserted into an independent list for each component $\mathcal{G}^i$. We will calculate the average of the b lower values on each list. In the 2 on line 19, we can appreciate how we get just the b lower elements on NB previously sorted on line 16. The results are stored on the array $\mathcal{W} = \{w_1, ..., w_c\}$ where each w_i represents the average difference of the b nearest betweenness node values on the sub-network $\mathcal{G}^i$. This process is represented in the algorithm 2 on line 27 and 28.

$$\mathcal{W}^n = \frac{1 - \mathcal{W}}{\sum_{w_i \in \mathcal{W}} 1 - w_i} \tag{3.3}$$

Algorithm 1 Node Insertion

```
1: function NodeInsertion(G, instance, index, k, e)
2:     ⟨V, E⟩ ← G                ▷ index is the number of nodes in the graph +1
3:     V ← V ∪ {index}
4:     edges ← {}
5:     for X_i ∈ X_training do
6:         if X_i ∈ N(instance, k, e) then
7:             edges ← edges ∪ (i, index)
8:         end if
9:     end for
10:    E ← E ∪ {edges}
11:    G ← ⟨V, E⟩
12:    return G
13: end function
```

where $\mathcal{W}^n$ is the normalized version of $\mathcal{W}$. In order to avoid conflicts of probabilities with the same value $w_i \in \mathcal{W}^n$, we calculate the number of links of the inserted node with respect to each sub-network $\mathcal{G}^i$ on the array $\mathcal{T}$. Then, we follow a similar process of equation 3.3 for $\mathcal{T}$ normalization. This process is represented in the algorithm 2 on line 29.

$$\mathcal{T}^n = \frac{\mathcal{T}}{\sum_{t_i \in \mathcal{T}} t_i} \tag{3.4}$$

Finally, once we normalize these values, we calculate the sum of $\mathcal{T}^n, \mathcal{W}^n$ and made a final normalization.

$$\mathcal{H} = \frac{(\alpha)\mathcal{W}^n + (1-\alpha)\mathcal{T}^n}{\sum_{t_i \in \mathcal{T}^n, w_i \in \mathcal{W}^n} (\alpha)w_i + (1-\alpha)t_i} \tag{3.5}$$

where $h_i \in \mathcal{H}$ represents the probability of a node i to be inserted in the sub graph $\mathcal{G}^i$, and α controls the weights between structural information and number of links. If $\alpha = 1.0$, we just capture information using *betweenness centrality*, and if $\alpha = 0.0$, we just capture information about number of links. The fully algorithm is described in algorithm 2.

Algorithm 2 Classification Algorithm

function CLASSIFICATION($\mathcal{G}$,$instance, k, e, b, \alpha$)
  $index \leftarrow n + 1$ ▷ n is the number of nodes in $\mathcal{G}$
  $\mathcal{G} \leftarrow$NodeInsertion($\mathcal{G}, instance, index, k, e$)
  $\mathcal{W} \leftarrow \{\}$
  $\mathcal{T} \leftarrow \{\}$
  for $\mathcal{G}^i \in \mathcal{G}$ **do** ▷ Where each $\mathcal{G}^i$ is a subgraph
    $NB \leftarrow \{\}$ ▷ NB is a list of node betweenness differences
    $\langle \mathcal{V}^i, E^i \rangle \leftarrow \mathcal{G}^i$
    $Links \leftarrow 0$
    for $j \in \mathcal{V}^i$ **do**
      if $j \in \mathcal{N}(index, k, e)$ **then**
        $Links \leftarrow Links + 1$
      end if
      $NB \leftarrow NB \cup \{B(index) - B(j)\}$ ▷ B is betweenness centrality
    end for
    Sort(NB) ▷ NB has the differences between the nodes in $\mathcal{G}^i$ and the new node
    $Total \leftarrow 0$
    $count \leftarrow 0$
    while $count < b$ **do**
      $Total \leftarrow Total + NB[count]$
      $count \leftarrow count + 1$
    end while
    $Total \leftarrow \frac{Total}{b}$
    $\mathcal{W} \leftarrow \mathcal{W} \cup Total$
    $\mathcal{T} \leftarrow \mathcal{T} \cup Links$
  end for
  $\mathcal{W}^n \leftarrow 1 - \mathcal{W}$
  $\mathcal{W}^n \leftarrow \frac{\mathcal{W}^n}{sum(\mathcal{W}^n)}$
  $\mathcal{T}^n \leftarrow \frac{\mathcal{T}}{sum(\mathcal{T})}$
  $\mathcal{H} \leftarrow (\alpha)\mathcal{W} + (1 - \alpha)\mathcal{T}$
  $\mathcal{H} \leftarrow \frac{\mathcal{H}}{sum(\mathcal{H})}$
  return MaxIndexValue($\mathcal{H}$) ▷ $\mathcal{H}$ has each class probability
end function

3.2 Attribute-attribute Interactions

In this section, we will explore the attribute-attribute interaction to develop a new build network method (Quipus).

3.2.1 Hidden Links

There are different methodologies to build the networks from the dataset that combines variations of kNN and e-radius techniques. However, following these strategies, we are ignoring some hidden links between the data. Each instance that is represented as a node is usually a complex structure with more than four attributes. So, we cannot ignore the fact that each instance is a network.

3.2.2 Quipus

The Quipus is an accounting tool used for the historical American Inca Empire to measure the number of people, food, animals, or other information that they have. It is based on strings for each thing to measure and knots to capture the quantity (SCHMIDT; SANTOS LUIZ DOS SANTOS, 2017). Inspired by this concept, we split the dataset for each attribute to capture the relations inside them. Also, each final attribute network presents a form of a string as it is shown by Figures 7 and 8.

3.2.3 Attribute-attribute interactions

We analyse how each instance is represented as a node but this approach ignore some hidden patterns between attribute-attribute interaction.

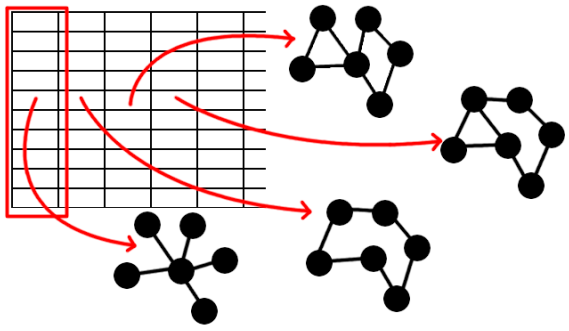


Figure 4 – Figure of 4 graphs that capture the interactions into each attribute.

In this method, we create a network for each attribute to detect this hidden patterns. In Figure 4, we can appreciate how each attribute is represented as an independent network. Using this approach, we can capture the attribute-attribute interaction. Since we are using attributes that have the same scale to produces the networks, our method increases its resistance to non-normalized data. However, there are attributes that by themselves do not provide relevant information and require others to be useful. Thus, we will use the Q to evaluate each network.

3.2.4 Training phase

In the training phase, we split the training dataset $X_{training}$ in two X_{net} and X_{opt} because we will use them for optimization phase. The proportion depends on the quantity of data available usually. Then, we follow the next steps:

- First, we build a network for each attribute to capture their hidden patters following the equation (3.1) on X_{net}. Then, we build the main network using all attributes to capture the instance-instance interaction.

- Second, we calculate the Q for each network. To avoid possible noise of attributes without relevant information, we ignore the networks with modularity lower than the main network.

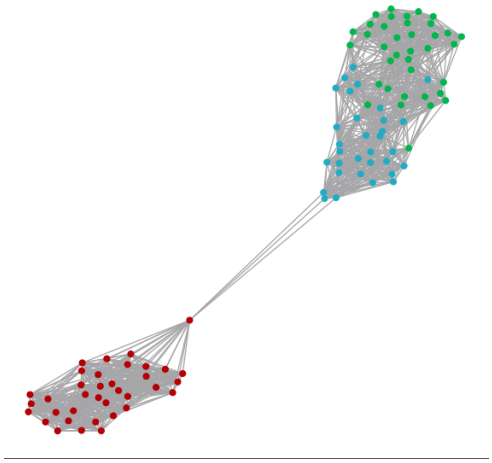
- Third, we insert each instance from X_{opt} to the networks following the same strategy described in step 1. However, we will keep the links between different labels because we want to simulate a real insertion. Then, we introduce each attribute into the correspondent network and the complete instance (with all attributes) in the instance-instance network (main network).

- Fourth, we obtain the probability to be part of each class in each network using the high-level algorithm HLNB-BC. For example, in a dataset with 3 classes and 4 attributes, it will give us a list with three probabilities for each network (12 probabilities in total).

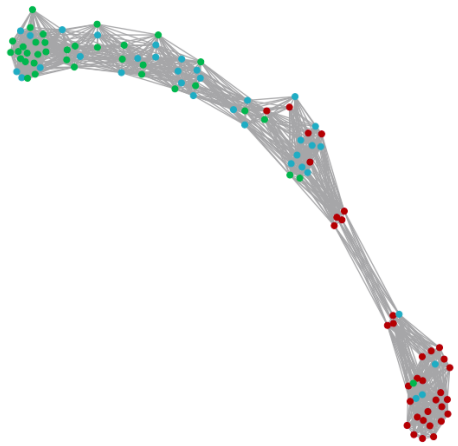
- Fifth, we give a weight for each network from 0 to 1. This will give us a way to reduce or increase the classification probability of the networks. Then, we use an optimization algorithm like particle swarm to determinate the better weights for each network to increase the accuracy of the predicted instances in X_{opt}.

- Finally, we save the weights and produce the final networks following the same procedure in step 1 with $X_{training}$.

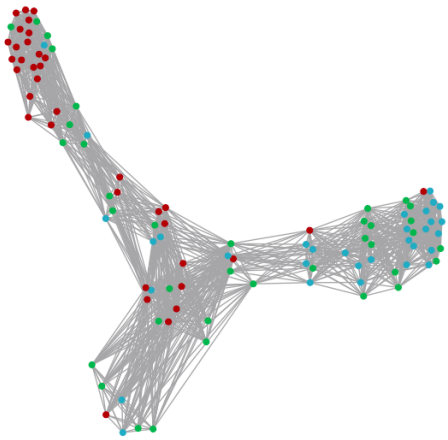
We use Q to reduce the networks with low community structure because some attributes introduce noise in the classification process. In figure 5, we observe that the first and second attributes present a clear community structure for the three classes but with some noise between green and blue nodes. However, the other two attributes do not have a clear structure. We use modularity to remove the weak networks. The modularity of the main network (with all attributes) is used as a delimiter.



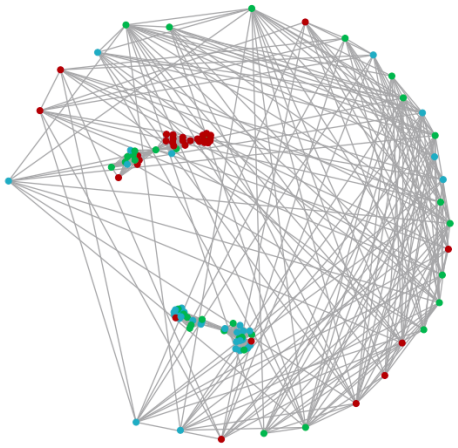
(a) Figure of iris attribute 1 network



(b) Figure of iris attribute 2 network



(c) Figure of iris attribute 3 network



(d) Figure of iris attribute 4 network

Figure 5 – Figure of the four attribute-attribute networks in iris dataset.

3.2.5 Testing phase

In testing phase, we use the final networks built in section 3.2.4 and predict each label in testing dataset $X_{testing}$.

- First, we insert each instance from $X_{testing}$ to the networks following the same strategy described in equation (3.1). However, we will keep the links between different labels because we want to calculate them. We introduce each attribute into its corresponding network and the complete instance in the main network.
- Second, we obtain the probability to be part of each class in each network using the high-level algorithm HLNB-BC.
- Third, we multiply these probabilities for their corresponding optimized weights calculated in section 3.2.4.
- Finally, the labels with higher probability will capture the new node.

4

Experimental Results

In this section, we present the experimental results of our algorithm in different scenarios. i) We explore the proposed techniques using synthetics datasets to explore the behavior of the high level technique in context with structured data. ii) We explore the proposed technique on UCI datasets to explore the algorithm parameters.

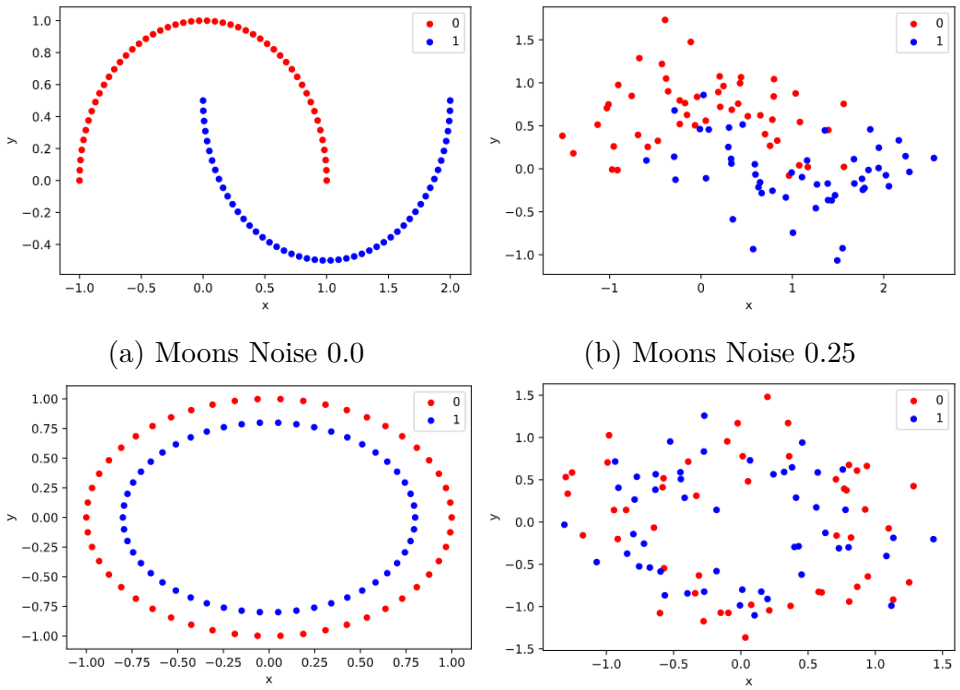
We compare the proposed algorithm with others high-level and classical classification algorithms using 5-folds cross validation 10 times technique and parameter optimization with grid search. The classical algorithms are: 1) Nearest Neighbors (optimized with k between 1 to 30); 2) Multi Layer Perceptron (input - dense layer - dense layer -output with 420 nodes); 3) Random Forest (trees between 100 to 500); 4) High-Level Classification (optimized with k between 1 to 30); 5) High-Level Link Prediction (optimized with k between 1 to 30) ; 6) Quipus. In Quipus, we search k from 1 to 30, the percentile ϵ is tested with the values $[0.1, 0.2, 0.3, 0.4, 0.5]$, b nearest nodes from 1 to 7 and α with these values $[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$. The results of cross validation show us improvements of the proposed technique in the tests.

4.1 Synthetic Datasets

In this section, we present the classification performance of our algorithm in toy datasets and compare the results with other algorithms using python as programming language and Scikit-learn library for algorithms (PEDREGOSA et al., 2011). Specifically, we test our algorithm against Multi Layer Perceptron (MLP) (RIEDMILLER; BRAUN, 1993), Decision Tree C4.5 (DT) (SHAFER; AGRAWAL; MEHTA, 2000), and Random Forest (RF) (BREIMAN, 2001). The algorithms are tested using cross validation 10-folds, executed 10 times, and we use a grid search to select the hyper parameters that give the best accuracy for all the algorithms.

The toy datasets are Moons and Circle with 0.0 and 0.25 of Gaussian standard

deviation noise added to the data 6. The NBHL-BC parameter values are shown in table 1, and the classification accuracy results are shown in table 2. These datasets are used because they present clear data patterns where traditional algorithms have their effectiveness reduced in these cases. In the case of Decision tree, we use gini index as quality measure without pruning method. In the case of Random Forest, we use gini index as split criterion and 100 trees. In the case of MLP, we use 2 hidden layer with 10 nodes and 100 interactions for dataset without noise and 500 interactions with noise.



(a) Moons Noise 0.0

(b) Moons Noise 0.25

(c) Circle Noise 0.0

(d) Circle Noise 0.25

Figure 6 – Figure of synthetic Datasets for Testing

Dataset	k	e	b	α	accuracy
Moons 0.00	5	0.5	5	1.0	**100.0**
Moons 0.25	8	0.0	10	1.0	**97.0**
Circle 0.00	1	0.5	1	1.0	**100.0**
Circle 0.25	5	0.5	1	1.0	64.0
Moons 0.00	5	0.5	5	0.5	**100.0**
Moons 0.25	9	0.0	10	0.5	96.0
Circle 0.00	1	0.0	1	0.5	**100.0**
Circle 0.25	5	0.5	1	0.5	**65.0**
Moons 0.00	5	0.5	5	0.0	**100.0**
Moons 0.25	9	0.0	10	0.0	96.0
Circle 0.00	1	0.0	1	0.0	**100.0**
Circle 0.25	5	0.5	1	0.0	64.0

Table 1 – Parameter values used by our algorithm (NBHL-BC) in Toy Datasets

We use in the first group $\alpha = 1.0$ because we want to evaluate just the structural

methodology using *betweenness centrality*. In the second group, we combine both strategies with $\alpha = 0.5$ and we got a small improvement on the dataset Circle 0.25. In the last group, we use just the number of links and we got similar results but there is a reduction of the accuracy in Moons 0.25. In some cases, we need to remove the property of $\epsilon - radius$ using $e = 0.0$ and increase the quantity of k neighbors in Moons with 0.25 noise. The b similar *betweenness centrality* nodes were kept in all the tests because other values reduce accuracy.

	MLP	DT	RF	NBHL-BC
Moons 0.00	94.0	95.0	98.0	**100.0**
Moons 0.25	84.0	85.0	91.0	**97.0**
Circle 0.00	90.0	92.0	91.0	**100.0**
Circle 0.25	62.0	56.0	56.0	**65.0**

Table 2 – Classification accuracy of the NBHL-BC compared to Multi Layer Perceptron (MLP), Decision Tree (DT), and Random Forest (RF) in Toy Datasets

In this simulations, our algorithm presents the best results in all the datasets. Specially in the the circle dataset with noise 0.25, which is the most difficult case, our algorithm presents better classification accuracy than other techniques under comparison.

4.2 UCI Datasets

In this section, we are going to present the results of the NBHL-BC technique on UCI classification datasets (DUA; GRAFF, 2017) . Also, we will compare our results with other algorithms. We test our algorithm against Multi Layer Perceptron (MLP) (RIEDMILLER; BRAUN, 1993), Decision Tree C4.5 (DT) (SHAFER; AGRAWAL; MEHTA, 2000), Random Forest (RF) (BREIMAN, 2001), and the Network Base High Level Technique (NBHL) (COLLIRI et al., 2018).

The algorithms are tested splitting each dataset in two sub data sets, for training and testing with a proportion of 75% and 25% respectively following an stratified sampling using python as programming language and Scikit-learn library for algorithms .

The datasets used are shown in Table 3 with the number of instances, attributes and classes. These datasets are selected because the previous high-level algorithm used them. The NBHL-BC parameter values are given in Table 4, and classification accuracy results are presented in Table 5.

Our algorithm presents good performance in all the datasets compared to other algorithms. In four cases, our algorithm presents the best results. Just in case of Iris dataset, another high level classification algorithm NBHL is better than the proposed one.

Dataset	Instances	Attributes	Classes
Glass	214	9	6
Iris	150	4	3
Pima	768	8	2
Teaching	151	5	3
Wine	178	13	3
Yeast	1484	8	10
Zoo	101	16	7

Table 3 – Information about the UCI classification dataset used on these project.

Dataset	k	e	b	α
Glass	1	0.0	1	1.0
Iris	7	0.0	3	1.0
Pima	8	0.0	4	1.0
Teaching	5	0.0	5	1.0
Wine	12	0.0	5	1.0
Yeast	14	0.0	3	0.5
Zoo	1	0.0	1	1.0

Table 4 – Parameter values used by our algorithm (NBHL-BC) in UCI datasets.

	MLP	DT	RF	NBHL	NBHL-BC
Glass	69.231	63.077	**75.385**	66.700	69.231
Iris	93.333	93.333	93.333	**97.400**	95.556
Pima	74.892	69.264	**77.056**	73.400	**77.056**
Teaching	52.174	52.174	60.870	55.300	**65.217**
Wine	96.296	92.593	**98.148**	80.000	**98.148**
Yeast	59.641	48.430	**61.883**	36.700	54.036
Zoo	96.774	96.774	96.774	**100.00**	**100.00**

Table 5 – Classification accuracy results of the NBHL-BC compared to Multi Layer Perceptron (MLP), Decision Tree C4.5 (DT), Random Forest (RF), and Network Base High Level Classification (NBHL) using the testing dataset.

Moreover, the α parameter that regulates the weight between the *betweenness* measure and number of links in 6 of the 7 datasets is 1.0 that means that the algorithm just use the *betweenness*. In the dataset Yeast, it is required an $\alpha = 0.5$ that means that give same importance between *betweenness* and number of links. In Table 6, we test UCI Wine dataset (DUA; GRAFF, 2017) using 10-fold cross validation with different values for α. The accuracy with only links number $\alpha = 0.0$ is quite lower than $\alpha = 1.0$, and the best result is mixing both techniques with $\alpha = 0.4$. The b parameter that evaluates the number of nodes with the lower *betweenness centrality* difference with respect to the inserted node were kept constant.

The datasets used, their attributes, instances, and classes are described in Table 3.

In Section 3.2.4, we split the training data in X_{net} and X_{opt}. In our tests, we use a

Dataset	k	e	b	α	accuracy
Wine	8	0.5	5	0.0	95.492
Wine	8	0.5	5	0.1	96.619
Wine	8	0.5	5	0.2	96.619
Wine	8	0.5	5	0.3	96.619
Wine	8	0.5	5	0.4	**97.175**
Wine	8	0.5	5	0.5	96.619
Wine	8	0.5	5	0.6	96.063
Wine	8	0.5	5	0.7	96.048
Wine	8	0.5	5	0.8	95.508
Wine	8	0.5	5	0.9	96.619
Wine	8	0.5	5	1.0	96.048

Table 6 – Results of 10-folds cross validation in UCI Wine dataset with the training dataset.

stratified random split 80% and 20% respectively. This value could be modified according to the quantity of data. Then, we build a network for each attribute and one network for instance-instance interactions. We calculate their modularities (Q) and compare each attribute network with the instance-instance network. The networks with lower modularity will be ignored in the rest of the process. Table 7 show us the modularities for each network.

Table 7 – Modularities of attribute networks in UCI Wine Dataset

Network	Modularity Q
Instance-instance	0.3181
Attribute 1	0.3189
Attribute 2	0.0924
Attribute 3	0.0500
Attribute 4	0.1689
⋮	⋮
Attribute 10	0.2288
Attribute 11	0.3008
Attribute 12	0.3333

For instance, the modularity of the networks attribute 1 and attribute 12 are higher than modularity of instance-instance network. So, these networks will be used for optimization, classification, and insertion. The others will be ignore because do not have a high community structure.

The insertion of the nodes into each graph follow the equation (3.1), but preserving the links with nodes of different labels. Given that we want to capture the insertion probability for each class. Then, we create a weight for each graph probabilities and start an optimization phase. We use a particle swarm optimization from Pyswarms library

with these parameters $\{c_1 = 0.5, c_2 = 0.1, w = 0.9, iterations = 500\}$. Theses could be optimized but we use these fixed values for these experiments.

For example, in one interaction, our algorithm capture the weight in Table 7.

Table 8 – Modularities and weights of attribute networks in UCI Wine Dataset

Network	Modularity Q	Weights	Ignored
Instance-instance	0.3181	0.9083	False
Attribute 1	0.3189	0.8065	False
Attribute 2	0.0924	-	True
Attribute 3	0.0500	-	True
Attribute 4	0.1689	-	True
⋮	⋮		
Attribute 10	0.2288	-	True
Attribute 11	0.3008	-	True
Attribute 12	0.3333	0.1746	False

Once the weights are defined, we proceed to rebuild the graphs but using the entire training dataset $X_{training}$. Finally, the classification phase, will follow the same process that optimization phase but using the optimized weights.

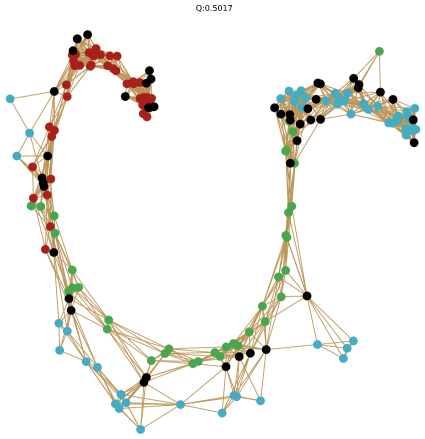


Figure 7 – Figure of instance-instance network from one interaction in UCI wine dataset classification.

In Figure 7, we can observe the instance-instance network from one instance of wine dataset classification. The black nodes represents instances classified. It present an structure where the red nodes are in one side, the blue nodes in the other side, and the green nodes in the middle of them. In the figure 8, we observe the network from the first attribute of wine datset that had a modularity of 0.3189. Once the nodes are inserted this graphs present a higher modularity $Q = 0.6553$. Without this methodology, we will lose these attribute-attribute interactions. These networks gives us an accuracy of 91.11%.

In Table 10, we observe the accuracy of Quipus against the literature network building technique kNN+ϵ-*radius*. This current technique present problems related to data

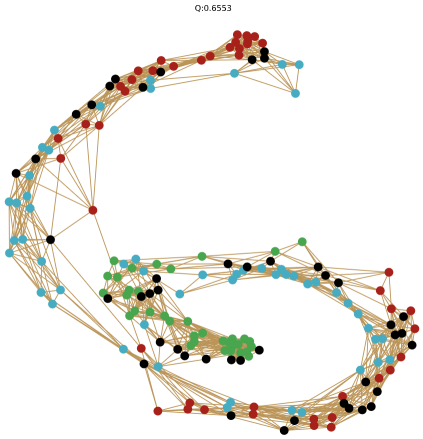


Figure 8 – Figure of attribute-attribute 1 network from one interaction with higher modularity than its instance-instance network in UCI wine dataset classification.

Table 9 – parameter values used by NBHL-BC with Quipus methodology in uci datasets

Dataset	k	e	b	α
Glass	1	0.0	1	0.5
Iris	12	0.0	3	1.0
Pima	8	0.0	4	0.5
Teaching	1	0.0	1	0.5
Wine	7	0.0	3	1.0
Yeast	14	0.5	1	0.5
Zoo	1	0.0	1	1.0

non-normalized like wine UCI dataset. However, using Quipus, we reduce this problem. Due to the attribute networks build their relations in the same scale, the optimized weight manage the probability force, and reduce its impact in the final classification.

Table 10 – Table with accuracy and one standard deviation of different building methodologies and UCI datasets without normalization.

Results of 10 times using 10-folds cross validation			
Dataset	Prediction	Building (k)	Accuracy
Glass	HLNB-BC	kNN+ϵ-*radius* (1) Quipus(1)	58.87 ± 10.73 57.94 ± 7.04
Iris	HLNB-BC	kNN+ϵ-*radius* (7) Quipus (12)	95.33 ± 05.92 95.80 ± 04.68
Pima	HLNB-BC	kNN+ϵ-*radius* (8) Quipus(8)	70.93 ± 3.18 72.96 ± 3.46
Teaching	HLNB-BC	kNN+ϵ-*radius* (1) Quipus(1)	60.24 ± 24.70 58.08 ± 23.48
Wine	HLNB-BC	kNN+ϵ-*radius* (1) Quipus (7)	75.84 ± 09.57 93.03 ± 06.54
Yeast	HLNB-BC	kNN+ϵ-*radius* (14) Quipus(14)	41.10 ± 2.77 41.61 ± 3.22
Zoo	HLNB-BC	kNN+ϵ-*radius* (1) Quipus(1)	96.36 ± 06.49 96.87 ± 02.485

5

Conclusion

The results presented in this research to classify using an attribute-attribute interaction with betweenness centrality show another perspective related to high-level classification. In some cases, the hidden patterns in data produce some promising results in non-normalized data. Also, in some cases, our algorithm presents a higher accuracy.

The toy dataset like circle and spiral explores that the betweenness centrality metric could describe complex structures. Moreover, the performance obtained in real datasets shows us that our algorithm can be used in a real business context.

This research shows us evidence that capturing more structures in data could improve the performance of the high-level algorithms. Other possible hidden patterns are in images where each instance presents a complex composition. More investigation will be needed to exploit these instance-attribute patterns in a high-level classification.

5.1 Publications

The research presented generated two scientific papers:

One titled "A Network-Based High-Level Data Classification Algorithm Using Betweenness Centrality", published in "XVII ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL", 2020, Porto Alegre, Brasil, pp. 188-198. Where we explore the Betweenness Centrality as a classification Technique.

The second paper "A new network-based high-level data classification methodology (Quipus) by modeling attribute-attribute interactions", presented in "INTERNATIONAL CONFERENCE OF DIGITAL TRANSFORMATION AND INNOVATION TECHNOLOGY INCODTRIN 2020", 2020, Quito, Ecuador. Where we explore the attribute-attribute interaction.

5.2 Future works

According to the presented results, we explore that a high-level classification technique using attribute-attribute interaction can capture the structure hidden in each attribute. Also, we exploit the community structure to identify the networks with the best performance.

We will explore other metrics to evaluate the performance of a network instead of modularity. This metric produces competent results to decide which network to use. However, we need to measure each network by each class independently.

Another problem is the high number of nodes and subgraphs that we must use to capture these hidden patterns. Reduce the number of attributes and capture the variation between them could improve the current algorithm. Non-supervised learning methods to reduce the dimensionality could provide an improvement in the algorithm.

Also, we want to introduce these algorithms in business problems like sales classification, time series analysis, and data visualization.

References

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, American Physical Society, v. 74, p. 47–97, Jan 2002. Disponível em: <https://link.aps.org/doi/10.1103/RevModPhys.74.47>.

ALBRIGHT, C.; WINSTON, W. L. *Business analytics: Data analysis and decision making*. [S.l.]: Cengage Learning, 2015.

ANUPRIYA, K. et al. Eshopping scam identification using machine learning. In: *2018 International Conference on Soft-computing and Network Security (ICSNS)*. [S.l.: s.n.], 2018. p. 1–7.

BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Texture analysis and classification: A complex network-based approach. *Information Sciences*, v. 219, p. 168 – 180, 2013. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025512004677>.

BARABASI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, v. 286, n. 5439, p. 509–512, 1999. Disponível em: <http://www.sciencemag.org/cgi/content/abstract/286/5439/509>.

BARABáSI, A.-L.; PóSFAI, M. *Network science*. Cambridge: Cambridge University Press, 2016. ISBN 9781107076266 1107076269. Disponível em: <http://barabasi.com/networksciencebook/>.

BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001.

CARNEIRO, M. et al. Particle swarm optimization for network-based data classification. *Neural Networks*, v. 110, p. 243–255, 2019.

CARNEIRO, M.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. *IEEE Transactions on Neural Networks and Learning Systems*, PP, p. 1–13, 08 2017.

CARNEIRO, M.; ZHAO, L. Analysis of graph construction methods in supervised data classification. In: *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2018. p. 390–395. ISSN null.

CARNEIRO, M.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. *IEEE Transactions on Neural Networks and Learning Systems*, v. 29, p. 3361–3373, 2018.

CHIRE-SAIRE, J. E. *New feature for Complex Network based on Ant Colony Optimization for High Level Classification*. 2020.

CLAUSET et al. Finding community structure in very large networks. *Physical Review E*, p. 1– 6, 2004.

COCA, A. E.; ZHAO, L. Musical rhythmic pattern extraction using relevance of communities in networks. *Information Sciences*, v. 329, p. 819 – 848, 2016. ISSN 0020-0255. Special issue on Discovery Science. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025515006842>.

COCA, A. E.; ZHAO, L. Musical scales recognition via deterministic walk in a graph. In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2016. p. 151–156. ISSN null.

COLLIRI, T. et al. A network-based high level data classification technique. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. p. 1–8. ISSN 2161-4407.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>.

ERDöS, P.; RéNYI, A. On random graphs i. *Publicationes Mathematicae Debrecen*, v. 6, p. 290, 1959.

FADAEE, S. A.; HAERI, M. A. Classification using link prediction. *Neurocomputing*, 2019.

FERREIRA, L. N.; ZHAO, L. Time series clustering via community detection in networks. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 326, n. C, p. 227–242, jan. 2016. ISSN 0020-0255. Disponível em: <https://doi.org/10.1016/j.ins.2015.07.046>.

FREEMAN, L. C. A set of measures of centrality based upon betweenness. *Sociometry*, v. 40, p. 35–41, 1977.

GAO, X. et al. Particle competition for multilayer network community detection. In: *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*. ACM, 2019. (ICMLC '19), p. 75–80. ISBN 978-1-4503-6600-7. Disponível em: <http://doi.acm.org/10.1145/3318299.3318320>.

GOPINATH, R.; AJAY, R.; SANJAY, C. *An Introduction to Machine Learning*. [S.l.]: Springer, 2019. ISBN 978-3-030-15729-6.

GéRON, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491962291, 9781491962299.

HIMANSHU, S. *Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Pattern Recognition Using Python*. 1st ed.. ed. [S.l.]: Apress, 2019. ISBN 978-1-4842-4148-6,978-1-4842-4149-3.

JING, L.; HUSSEIN, A.; KAY, C. T. *Evolutionary Computation and Complex Networks*. [S.l.]: Springer, 2019. ISBN 978-3-319-60000-0.

KELLEHER, J. D.; NAMEE, B. M.; D'ARCY, A. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. [S.l.]: The MIT Press, 2015. ISBN 0262029448, 9780262029445.

KIM, H.-J.; KIM, J. M. Cyclic topology in complex networks. *Phys. Rev. E*, American Physical Society, v. 72, p. 036109, Sep 2005. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.72.036109>.

KREYENHAGEN, C. D. et al. Using supervised learning to classify clothing brand styles. In: *2014 Systems and Information Engineering Design Symposium (SIEDS)*. [S.l.: s.n.], 2014. p. 239–243.

LIMA, G. V. de et al. Classification of texture based on bag-of-visual-words through complex networks. *Expert Systems with Applications*, v. 133, p. 215 – 224, 2019. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417419303483>.

MARIA, A. L. et al. *Complex Networks and Their Applications VII: Volume 1 Proceedings The 7th International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2018*. 1st ed.. ed. [S.l.]: Springer International Publishing, 2019. (Studies in Computational Intelligence 812). ISBN 978-3-030-05410-6,978-3-030-05411-3.

MILOš, S.; MIRJANA, I.; LAKHMI, J. *Complex Networks in Software, Knowledge, and Social Systems*. 1st ed.. ed. [S.l.]: Springer International Publishing, 2019. (Intelligent Systems Reference Library 148). ISBN 978-3-319-91194-6,978-3-319-91196-0.

MURPHY, K. P. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. ISBN 9780262018029 0262018020. Disponível em: <https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2>.

NEEDHAM, M.; HODLER, A. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, Incorporated, 2019. ISBN 9781492047681. Disponível em: <https://books.google.com.br/books?id=UwIevgEACAAJ>.

NEWMAN, M. E. J. Mixing patterns in networks. *Phys. Rev. E*, American Physical Society, v. 67, n. 2, p. 026126, fev. 2003.

OLIVIER, C.; BERNHARD, S.; ALEXANDER, Z. *Semi-Supervised Learning*. [S.l.]: MIT Press, 2006. (Adaptive computation and machine learning). ISBN 0262033585,978-0-262-03358-9.

PATIL, A. P. et al. Customer churn prediction for retail business. In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. [S.l.: s.n.], 2017. p. 845–851.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PERRA, N.; FORTUNATO, S. Spectral centrality measures in complex networks. *Phys. Rev. E*, American Physical Society, v. 78, p. 036107, Sep 2008. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.78.036107>.

RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: *IEEE International Conference on Neural Networks*. [S.l.: s.n.], 1993. p. 586–591 vol.1.

SCHMIDT, P.; SANTOS LUIZ DOS SANTOS, J. The application of inca khipu as an accountability and managerial control tool. *Revista Brasileira de Gestao de Negócios*, scielo, v. 19, p. 613 – 626, 12 2017. ISSN 1806-4892.

SEAN, C. et al. *Complex Networks X: Proceedings of the 10th Conference on Complex Networks CompleNet 2019*. 1st ed.. ed. [S.l.]: Springer International Publishing, 2019. (Springer Proceedings in Complexity). ISBN 978-3-030-14458-6,978-3-030-14459-3.

SHAFER, J.; AGRAWAL, R.; MEHTA, M. Sprint: A scalable parallel classifier for data mining. *VLDB*, 08 2000.

SILVA, T. C.; ZHAO, L. Network-based high level data classification. *IEEE Transactions on Neural Networks and Learning Systems*, v. 23, n. 6, p. 954–970, June 2012. ISSN 2162-237X.

SILVA, T. C.; ZHAO, L. High-level pattern-based classification via tourist walks in networks. *Information Sciences*, v. 294, p. 109 – 126, 2015. ISSN 0020-0255. Innovative Applications of Artificial Neural Networks in Engineering. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025514009608>.

SILVA, T. C.; ZHAO, L. *Machine Learning in Complex Networks*. Springer International Publishing, 2016. Disponível em: <http://dx.doi.org/10.1007/978-3-319-17290-3>.

SOUTO, P. C. et al. Capturing financial volatility through simple network measures. In: AIELLO, L. M. et al. (Ed.). *Complex Networks and Their Applications VII*. Cham: Springer International Publishing, 2019. p. 534–546. ISBN 978-3-030-05414-4.

VITO, L.; VINCENZO, N.; GIOVANNI, R. *Complex Networks: Principles, Methods and Applications*. 1. ed. [S.l.]: Cambridge University Press, 2017. ISBN 1107103185,9781107103184.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. *Nature*, Nature Publishing Group, Department of Theoretical and Applied Mechanics, Cornell University, Ithaca, New York 14853, USA. djw24@columbia.edu, v. 393, n. 6684, p. 440–442, jun. 1998. ISSN 0028-0836. Disponível em: <http://dx.doi.org/10.1038/30918>.

WITTEN, I. H. et al. *Data Mining: Practical Machine Learning Tools and Techniques*. 4. ed. [S.l.]: Morgan Kaufmann, 2016. (Morgan Kaufmann Series in Data Management Systems). ISBN 0128042915,9780128042915.

WŁADYSŁAW, H.; WITOLD, P. *Pattern Recognition. A Quality of Data Perspective*. [S.l.]: Wiley, 2018. ISBN 9781119302834.

ZHOU, S.; MONDRAGON, R. J. The rich-club phenomenon in the internet topology. *IEEE Communications Letters*, v. 8, n. 3, p. 180–182, March 2004. ISSN 1089-7798.