Network Structural Optimization Based on Swarm Intelligence for Highlevel Classification

Murillo G. Carneiro
Faculty of Computing,
Federal University of Uberlândia,
Uberlândia, 38400-902, Brazil
mgcarneiro@ufu.br

Liang Zhao
Department of Computing
and Mathematics,
University of São Paulo,
Ribeirão Preto, 14040-901, Brazil
zhao@usp.br

Ran Cheng and Yaochu Jin
Department of Computer Science,
University of Surrey,
Guildford, GU2 7XH, United Kingdom
{r.cheng,yaochu.jin}@surrey.ac.uk

Abstract—**While most part of the complex network models are described in function of some growth mechanism, the optimization of a goal or certain characteristics can be desirable for some problems. This paper investigates structural optimization of networks in the highlevel classification context, where the classification produced by a traditional classifier is combined with the classification provided by complex network measures. Using the recently proposed social learning particle swarm optimization (SL-PSO), a bio-inspired optimization framework, which is responsible to build up the network and adjust the parameters of the hybrid model while conducting the optimization of a quality function, is proposed. Experiments on two real-world problems, the Handwritten Digits Recognition and the Semantic Role Labeling (SRL), were performed. In both problems, the optimization framework is able to improve the classification given by a state-of-the-art algorithm to SRL. Furthermore, the optimization framework proposed here can be extended to other machine learning tasks.**

I. Introduction

Graphs (or networks) are an important tool to model real systems. They are very applied to a wide range of problems which include social, biological, energy and technological networks [1], [2]. As the topological features of these networks are non-trivial, i.e. neither purely regular nor purely random, they are called complex networks. Some of the most well-known models of complex networks are the small-world and the scale-free networks which present specific structural features [3], [4]. The former is characterized by high clustering and short path lengths and the latter by a power law. As these examples, the complex network literature contains many network models described in function of some growth mechanism. Alternatively, some real-world problems are better explained by optimizing a goal or certain characteristics, which is called structural optimization of the network. A very intuitive example presented in [5] is the hub-and-spoke design of the airline networks, which optimizes the efficiency by ensuring that flights in and out of minor destinations are to and from major hubs, ensuring fuller planes while still giving the passengers a short journey.

As the literature contains many complex network measures to characterize nodes, sub-network and the whole network [6], most works have been largely concentrated in explore these measures while the structural optimization (also called network formation in this paper) has been a barely explored topic. The main study related to it is presented in [7]. In that paper, authors investigate structural optimization of networks through a quality function composed by two opposite measures: the number of edges and the mean geodesic distance. As result, the authors found four major types of networks: sparse exponential-like, scale-free, highly dense and star. Here, we investigate network structural optimization on the supervised learning context.

Complex networks have been widely applied to unsupervised and semi-supervised learning tasks [8]–[14]. In the former, the techniques usually try to exploit the topological features of the network, by using for example the well-known modularity measure, to detect communities (or clusters). In the latter, the idea is to exploit some label or other information propagation process in the network to classify the unlabeled data. Recently, network-based techniques have been also designed to the supervised learning [15]–[19] and their results are promising.

In this paper, structural optimization is studied on the high-level classification model proposed in [16]. In that paper, the authors proposed a hybrid model that combines traditional and network-based classification techniques, where the traditional one captures physical features of each class (such as support vector machine for instance) and the network-based one is able to classify data instances by verifying the conformation of data pattern formed by the training data, which means a test instance receives the label from the data network of which structure is kept unmodified or is barely modified after the insertion of the test instance.

In order to employ some complex network theory, it is usually required the data set be represented as a graph. However, it is known the biggest part of the data sets in machine learning are available as vector-based feature data. As a consequence, to be able to exploit complex network properties in machine learning, a graph is usually formed from the vector-based feature data by using a graph formation technique. This would imply in many works about structural optimization, but it is not the case in supervised learning. While graph-based unsupervised and semi-supervised learning have been extensively studied, there are still few reported graph-based supervised learning techniques proposed directly to handle vector-based data. The few graph-based supervised learning methods proposed in the literature are usually based on the k-nearest neighbor network [10], [15]. Unlikely of setting a specific value of connections by all vertices (or groups of vertices), our study propose a bio-inspired optimization framework where connections between vertices are iteratively updated by the recently proposed social learning particle swarm optimization (SL-PSO), which has been chosen due its robust performance on high dimensional problems [20]. To the best of our knowledge, there is no such kind of approach in the literature. Perhaps the most related works are those designed to community detection task, in which bio-inspired algorithms are employed to search the best partition of network communities (also called groups, clusters or modules) which optimizes a fitness function usually based on the modularity measure [21].

To be specific, the proposed bio-inspired optimization framework combines traditional and highlevel classification. The framework is expected to build up the network and tune the parameters of the hybrid model while conducting the optimization of a quality function, denoted by the predictive accuracy here. The structural optimization is also evaluated by combining the quality function and complex network measures, such as closeness and assortativity [22], [23]. Empirical simulations have been performed on two real-world problems, the Handwritten Digits Recognition (HDR) and the Semantic Role Labeling (SRL). Both simulations show the optimization framework is able to improve the performance of a rigorously tunned state-of-the-art algorithm to SRL task.

The remainder of the paper is organized as follows. Section II introduces network-based supervised learning, highlevel classification and the SL-PSO algorithm. The framework proposed for structural optimization is detailed in Sect. III. Computer simulations and discussions about the results obtained on the real-world data sets are presented in Sect. IV; and Sect. V concludes the paper.

II. Background

In this section, we present a brief description about the techniques related to the paper. Firstly, we define the network-based data classification. Following, we describe the hybrid framework to perform highlevel classification. Finally, we present a quick overview about the SL-PSO algorithm used in this paper.

A. Problem Definition

In the data classification problem concerned here, the algorithms receive as input a training data set usually denoted by $X_{train} = \{(x_1, l_1), \dots, (x_n, l_n)\}$, where n means the number of labeled instances. Each instance $x_i = (a_1, \dots, a_d)$ represents the attributes, also called features, of a d-dimensional data item and $l_i \in \mathcal{L}$ represents the target class or label associated to that data item. The goal in the training phase is to induce a classifier $x \rightarrow l$ by using the training data X_{train}. In the testing phase, the induced classifier is used to predict the label of the new input instances, also called test instances, which are usually denoted by $X_{test} = \{(x_{n+1}, ?), \dots, (x_{n+m}, ?)\}$, where "?" means the label to be predict.

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ the set of edges. In network-based supervised learning, each vertex $v_i \in \mathcal{V}$ represents a labeled instance $x_i \in X_{train}$ and $e_{ij} \in \mathcal{E}$ represents a link between node v_i and v_j, which usually indicate some similarity. In general, the information required to perform data classification is obtained from $\mathcal{G}$ by using complex network measures, which means that the network is the key to the results of prediction as the pattern formation of the classes are directly extracted from it. Therefore, the optimization framework proposed here performs the structural optimization of $\mathcal{G}$ in order to reach some goal or certain characteristics, which is related to the predictive performance in this paper.

B. Highlevel Classification via Complex Network Measures

Traditional techniques such as neural networks, support vector machine, Bayesian learning, instance-based learning, etc., perform classification considering only the physical features of the data (e.g. distance or similarity). By contrast, complex networks are expected to detect the global semantic characteristics of the data by taking also its topological structure into consideration, which is known as highlevel classification. This is the principle behind the hybrid model proposed in [16], in which complex network measures are applied to improve the final prediction of traditional classifiers. In the model, the classification of a test instance y is formally described by:

$$\mathcal{M}_y^{(c)} = (1 - \lambda)\mathcal{C}_y^{(c)} + \lambda \mathcal{H}_y^{(c)}, \tag{1}$$

with $\mathcal{M}_y^{(c)}$ denoting the association produced by traditional and high level algorithms when evaluating instance y for the class c. Also in the equation, $\mathcal{C}_y^{(c)} \in [0, 1]$ establishes the association produced by a traditional classifier between the instance y and the class c; and $\mathcal{H}_y^{(J)} \in [0, 1]$ points to an association produced by the highlevel technique. The contribution of the traditional and highlevel techniques in the final classification is given by $\lambda \in [0, 1]$, which is an user-controllable variable.

To be specific, the highlevel classification of a new instance y for a given class c, is presented by:

$$\mathcal{H}_y^{(c)} = \frac{\sum_{u=1}^{Z} \delta(u)[1 - b_y^{(c)}(u)]}{\sum_{g \in L} \sum_{u=1}^{Z} \delta(u)[1 - b_y^{(g)}(u)]}, \tag{2}$$

where $\mathcal{H}_y^{(c)} \in [0, 1]$, u is related to the network measures employed in the high level algorithm, $\delta(u) \in [0, 1]$, $\forall u \in \{1 \dots, Z\}$ is an user-controllable variable that indicates the influence of each network measure in the classification process and $b_y^{(c)}(u)$ provides an answer whether the test instance y presents the same patterns of the class c or not, considering the u-th network measure applied. The denominator term is only for normalization. There is also a constraint about $\delta(u)$, where (2) is valid only if $\sum_{u=1}^{Z} \delta(u) = 1$. About $b_y^{(c)}(u)$, it is given by:

$$b_y^{(c)}(u) = \Delta G_y^{(c)}(u)\ p^{(c)}, \tag{3}$$

in which $\Delta G_y^{(c)}(u) \in [0,1]$ represents the variation that occurs in a network measure whenever a new instance y is inserted into the network related to class c, with $p^{(c)} \in [0,1]$ denoting the proportion of instances that belongs to the class c.

It can concluded that, in the highlevel classification, each network represents a unique class and the classification is conducted by inserting the test data y into different networks. As a consequence, the classification results are determined by the values returned by $\mathcal{H}$ which indicates the degree of variations: a large $\mathcal{H}$ value indicates that y is in conformity with the pattern of the corresponding network (class), and vice versa. To formulate $\mathcal{H}$, this paper employs three complex network measures: average closeness, assortativity and clustering coefficient. A more detailed view about these measures is provided in the next section.

C. Social Learning Particle Swarm Optimization

The social learning particle swarm optimization algorithm (SL-PSO) is a recently proposed swarm intelligence paradigm for solving numerical optimization algorithms [20]. Unlike traditional particle swarm optimization (PSO) algorithms [24], [25], the SL-PSO does not memorize the historical best positions – neither the global best position $gebst$ nor the personal best position $pbest$. Instead, in SL-PSO, the swarm is sorted according the fitness values of the particles, and as a consequence, each particle is made to learn from any better particles in the current swarm as follows:

$$s_{i,j}(t+1) = \begin{cases} s_{i,j}(t) + vel_{i,j}(t+1), & \text{if } p_i(t) \leq Prob_i^L \\ s_{i,j}(t), & \text{otherwise} \end{cases} \tag{4}$$

where t is the generation counter, $s_{i,j}(t)$ is the j-th ($j \in \{1,2,3,...,D\}$) dimension in the position vector of particle i ($i \in \{1,2,3,...,m\}$), with m and D denoting the swarm size and number of decision variables, respectively; $s_{i,j}(t)$ is updated according to velocity $vel_{i,j}(t+1)$ based on a *learning probability* $Prob_i^L$ for each particle i in the sorted swarm; ϵ is a parameter known as the social influence factor; $F_1(t)$, $F_2(t)$ and $F_3(t)$ are three coefficients randomly generated within $[0,1]$. In detail, $vel_{i,j}(t+1)$ is generated as follows:

$$\begin{aligned} vel_{i,j}(t+1) &= F_1(t) \cdot vel_{i,j}(t) \\ &\quad + F_2(t) \cdot I_{i,j}(t) + F_3(t) \cdot \varepsilon \cdot S_{i,j}(t), \end{aligned} \tag{5}$$

where $vel_{i,j}(t+1)$ consists of three components: the *inertia component* $F_1(t) \cdot vel_{i,j}(t)$, the *imitation component* $I_{i,j}(t)$ and the *social influence component* $S_{i,j}(t)$. The inertia component is similar to that in traditional PSO, while the the imitation component and the social influence component are inspired from the social learning theories as:

$$\begin{cases} I_{i,j}(t) = s_{k,j}(t) - s_{i,j}(t), \\ S_{i,j}(t) = \bar{s}_j(t) - s_{i,j}(t). \end{cases} \tag{6}$$

where $s_{k,j}(t)$ is the j-th dimension in the position vector of particle k, known as a demonstrator, which has a better fitness value than particle i; $\bar{s}_j(t) = \frac{\sum_{i=1}^{m} s_i^j}{m}$ is the mean position of the swarm in generation t.

The experimental results in [20] has demonstrated that SL-PSO has robust performance on a variety of 47 benchmark functions scaling from 30 to 1000 dimensions, in comparison with several state-of-the-art PSO variants.

III. Model Description

In this section, the proposed bio-inspired optimization framework (optimization framework for short hereafter) is described in details. The major contribution of this investigation is regarding how to obtain the network through a structural optimization process. To the best of our knowledge, this is the first attempt to adopt bio-inspired algorithms to perform structural optimization in the network-based data classification context. In addition, we also exploit the optimization process to adjust the parameters of the hybrid model $\mathcal{M}$. Sub-Sects. III-A and III-B present the optimization framework and the complex network measures employed in this work, respectively.

A. Optimization Framework

The optimization framework proposed in this paper is divided in two phases. The first one is the optimization phase (or training) which compreheends network and parameter optimization. In this phase, SL-PSO algorithm is used to construct the graph from the training data X_{train} according to the optimization of a given quality function f under a given validation data set X_{valid}. The second phase is the testing phase, where the best result in the training phase is employed by f to classify every new instance $y \in X_{test}$. Here, the function f gives the predictive accuracy of the hybrid model described by (1). The main steps of the general framework for structural optimization is illustrated by Fig. 1. Initially, SL-PSO creates a population of particles, where each particle P_i is composed by a graph G_i and a set of parameters R_i. Then, at each generation t, the particles are evaluated and updated (Δ) according to a quality function f. At the end, SL-PSO returns the particle with the best fitness value, which contains the graph and the parameters to be adopted in the testing phase to classify the unknown instances in X_{test}.

1) Social Learning - PSO: The optimization phase in the proposed framework is driven by the SL-PSO algorithm, which is a recently proposed bio-inspired algorithm for large scale optimization. Given a population of m particles $P = \{P_1, \ldots, P_m\}$, each particle $P_i \in P$ can be represented as follows:

$$P_i = \{\mathcal{G}_i, \mathcal{R}_i\}, \tag{7}$$

where $\mathcal{G}_i$ denotes the network and $\mathcal{R}_i$ the parameters. At each generation, SL-PSO evaluate and update the particles according to the better particles in a way that each particle learns from any other better ones. The algorithm stops when the predefined maximum number of generations is reached.

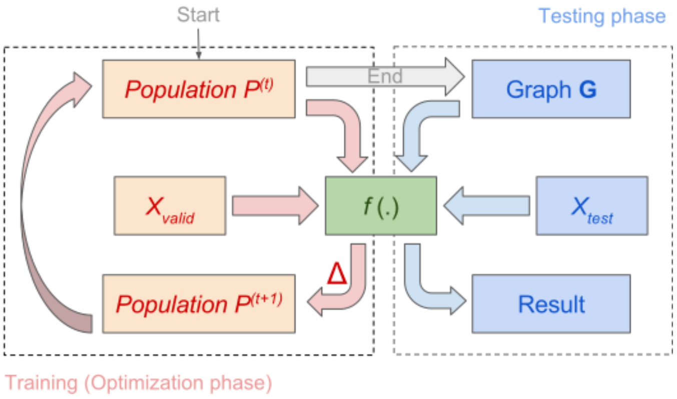


Fig. 1. General framework for structural optimization in network-based supervised learning.

More details of SL-PSO can be referred to Sub-Sect. II-C, as well as [20].

2) Network Optimization: In the data representation designed for the optimization framework, each network $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$ is represented as the following vector:

$$V_i = \{v_1, \dots, v_n\}, \tag{8}$$

where n means the number of instances (or vertices) each vertex $v_j \in \mathcal{V}_i$ denotes a labeled instance $x_j \in X_{train}$ and its links are represented by:

$$v_j = \{e_{j1}, \dots, e_{jq}\}, \tag{9}$$

with q denoting the maximum number of links and e_{jk} denoting the probability of a link between node v_j and node v_k. Values of e_{jk} are continuous and vary between $[0, 1]$ in order to be manipulated by SL-PSO. In addition, the complex network measures encoded with binary values, denoted as $e'_{jk} \in \mathcal{E}_i$, which can be obtained as:

$$e'_{jk} = \begin{cases} 1, & \text{if } e_{jk} \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Given a graph with n vertices, since the total number of possible edges is n^2, the complexity of the search space is $\mathcal{O}(n^2)$. In complex networks, however, since n can be as large as hundreds or even thousands, a search complexity of $\mathcal{O}(n^2)$ is infeasibly expensive. To address this issue, our framework employs a mapping heuristic that creates a sub-dimensional space based on the features of the given data set X_{train}, which reduces the search complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ ($\mathcal{O}(n \cdot q), q \ll n$). The heuristic comprises the following steps:

1) Compute the similarity among the instances;
2) Select for each vertex v_i its q more similar vertices;
3) Create $Map_{n \times q}$ matrix, where:

$$Map_{jz} = \begin{cases} v_z, & \text{if } v_z \text{ and } v_j \text{ belong to the same class,} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Figure 2 shows an example about the transformation of the solutions from the vector-based probability, which is manipulated by SL-PSO, to the network, which is used by the complex network measures in f. Let us consider a given vertex v_j as example. Initially, each link probability related to v_j, denoted by e_{jk} in SL-PSO, is transformed to a binary vector using (10). Then, a simple multiplication between v_j and Map_j give us the vertices which v_j is connected with, as illustrated in the figure.

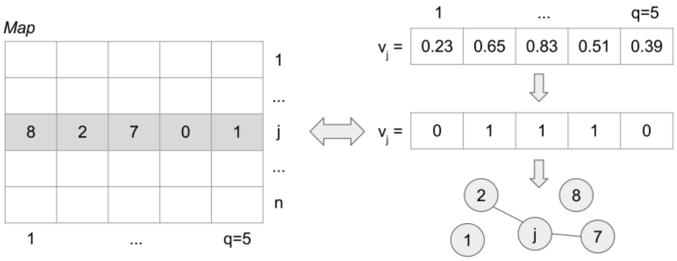


Fig. 2. Example of converting the representation: from vector-based probability to graph.

3) Parameters: In addition to the graph $\mathcal{G}_i$, a particle $\mathcal{P}_i$ is also composed by a set of parameters of the hybrid model, denoted by $\mathcal{R}_i$:

$$\mathcal{R}_i = \{\epsilon, \delta, \lambda\}, \tag{12}$$

where:

- ϵ is the similarity distance in which each new instance is virtually connected to the vertices in the network. In our framework, ϵ is optimized between the smallest and the biggest similarity value in X_{train};
- δ denotes the contribution of each network measure employed by the highlevel technique. It takes into account a number Z of network measures such that $\sum_{h=1}^{Z} \delta(h) = 1$;
- $\lambda \in [0, 1]$ represents the contribution of each classification, traditional and highlevel, in the combined model described by (1).

4) Quality Function f: In this paper, the quality function f computes the predictive accuracy of the classification produced by the hybrid model described by (1). Especially for the optimization phase, we have also designed some structural optimization strategies based on complex network measures as follows:

$$f^{(\cdot)} = (1 - \alpha) f + \alpha f^{(\cdot)}, \tag{13}$$

where $\alpha = 0.001$ provides a very smooth bias in the quality function, and $f^{(\cdot)}$ can be:

- f^{AC+}, which increases the average closeness while maximizing the predictive accuracy;
- f^{r+}, which increases the assortativity patterns while maximizing the predictive accuracy.

In population-based optimization, it is usually expected that there should be some gradient information that guides the convergence of the candidate solutions towards global optimum. In this way, (13) gives a smooth bias for some network characteristics, such as the assortativity patterns of the network when using f^{r+}.

B. Complex Network Measures

This section describes the complex network measures employed in the work, which are expected to provide the high-level classification in the hybrid model. While *assortativity* and *clustering coefficient* have already been used in [16], the introduction of the *average closeness* in the hybrid model is a major contribution of this work.

1) Average Closeness - $\Delta G_y^{(c)}(1)$: This centrality mesure is based on the length of the average shortest path between a vertex and all vertices in the graph [22]. The normalized closeness with regards to each class c is given by:

$$AC_i^{(c)} = \frac{V_c - 1}{\sum_{j=1}^{V_c} d(i,j)}, \tag{14}$$

$$AC^{(c)} = \frac{1}{V_c}\sum_{i=1}^{V_c} AC_i^{(c)}, \tag{15}$$

in which $AC^{(c)} \in [0,1]$, $d(i,j)$ is the shortest distance between vertices i and j, and V_c denotes the number of vertices in class c. Regarding the membership value of a test instance y with respect to class c, it is given by:

$$\Delta G_y^{(c)}(1) = \frac{\mid AC'^{(c)} - AC^{(c)} \mid}{\sum\limits_{u \in U} \mid AC'^{(u)} - AC^{(u)} \mid}, \tag{16}$$

where the denominator is only for normalization.

2) Assortativity - $\Delta G_y^{(c)}(2)$: This measure analyzes whether a link occurs preferentially between vertices with similar degree or not [23]. The assortativity with regards to each class c of the data set is given by:

$$r^{(c)} = \frac{L^{-1}\sum_{u\in U_c} i_u k_u - [L^{-1}\sum_{u\in U_c}\frac{1}{2}(i_u + k_u)]^2}{L^{-1}\sum_{u\in U_c}\frac{1}{2}(i_u^2 + k_u^2) - [L^{-1}\sum_{u\in U_c}\frac{1}{2}(i_u + k_u)]^2}, \tag{17}$$

where $r^{(c)} \in [-1,1]$, U_c encompasses all the edges within the class c and i_u, k_u indicate the vertices at each end of the edge $u \in U_c$. Therefore, the membership value is given by:

$$\Delta G_y^{(\alpha)}(2) = \frac{\mid r'^{(\alpha)} - r^{(\alpha)} \mid}{\sum\limits_{u \in U} \mid r'^{(u)} - r^{(u)} \mid}, \tag{18}$$

3) Clustering Coefficient - $\Delta G_y^{(c)}(3)$: This measure quantifies the degree to which local nodes in a network tend to cluster together [3]. The clustering coefficient with regards to each class c is given by:

$$CC_i^{(c)} = \frac{\mid \{e_{us} : v_u, v_s \in V_c, e_{us} \in \mathcal{E}\} \mid}{k_i(k_i - 1)}, \tag{19}$$

$$CC^{(c)} = \frac{1}{V_c}\sum_{i=1}^{V_c} CC_i^{(c)}, \tag{20}$$

in which $CC_i^{(c)} \in [0,1]$, k_i is the average degree of vertex i, and V_c denotes the number of vertices in the class c. The membership value of a test instance y is given by:

$$\Delta G_y^{(c)}(3) = \frac{\mid CC'^{(c)} - CC^{(c)} \mid}{\sum\limits_{u \in U} \mid CC'^{(u)} - CC^{(u)} \mid}. \tag{21}$$

IV. Results

This section presents empirical simulations to analyze the performance of the proposed technique. It is divided into three subsections: Subsect. IV-A provides an illustrative example about how the proposed framework works; Subsects. IV-B and IV-C discuss simulations on real-world data sets, showing that the proposed optimization framework not only improves the performance of a rigorously tuned state-of-the-art classifier, but also outperforms a widely-used network formation method.

A. Toy Example

Let us demonstrate the general idea of the proposed work through the toy example presented in Fig. 3. In the figure, there are two classes, in which the data items are denoted by *gray* and *black* colors. There is also an unlabeled instance y, denoted by *green* color, which needs to be classified. One can see each class has a clear pattern. However, traditional techniques have troubles to classify y into the *gray* class because they considers only the physical features of the input data in the classification process. Unlikely traditional techniques, complex network measures can detect the pattern formation of the data by analyzing also its topological structure. Table I shows the probabilities associated to each class by a traditional classifier $\mathcal{C}$ (Logistic Regression algorithm was used here), the complex network measures $\mathcal{H}$ and the combination between them $\mathcal{M}$, after the optimization process conducted by our technique. The parameters adjusted by the optimization framework are presented in Table II.

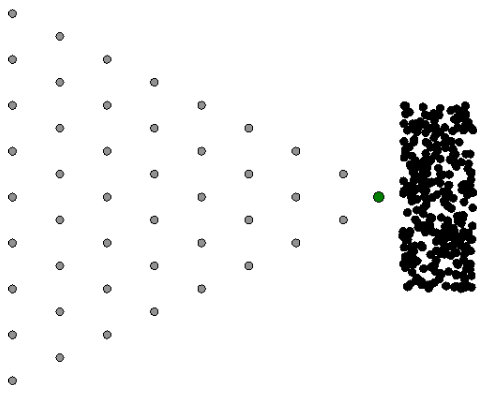


Fig. 3. Illustrative data set with two classes, in which gray class data items form a triangle pattern. The green color item represents a test instance y which needs to be classified.

Figure 4 shows the virtual insertion of y into the *gray* class (Fig. 4a) and into the *black* class (Fig. 4b), when the variation of the complex network measures are calculated. The results of the network measures before and after the insertion of y

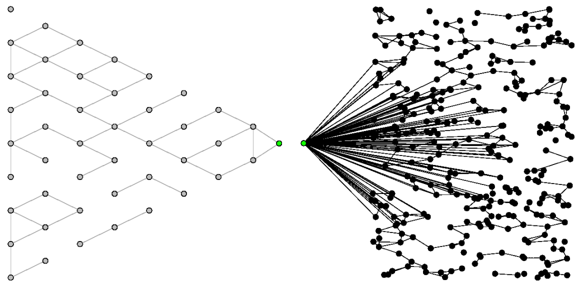
TABLE I
CLASSIFICATION RESULTS OBTAINED BY THE OPTIMIZATION FRAMEWORK FOR THE ILLUSTRATIVE EXAMPLE.

Alg.	*Gray* Class	*Black* Class
$\mathcal{C}$ (Logistic Regression)	0.093	**0.907**
$\mathcal{H}$ (Network measures)	**0.796**	0.204
$\mathcal{M}$ (Combination)	**0.605**	0.395

TABLE II
PARAMETERS OPTIMIZED BY THE FRAMEWORK FOR THE ILLUSTRATIVE EXAMPLE.

Parameter	Value
ϵ	0.135
w_1	0.431
w_2	0.239
w_3	0.330
λ	0.729

are presented in Tab. III. By the table, it is possible to see that the optimization framework adjust the connections and the parameters in order to detect the pattern.



(a) *Gray* network (class). (b) *Black* network (class).

Fig. 4. Virtual insertion of the test instance y in each class of the illustrative example.

TABLE III
HIGHLEVEL CLASSIFICATION THROUGH COMPLEX NETWORK MEASURES ON THE ILLUSTRATIVE DATA SET. THE CALCULATED AVERAGE CLOSENESS, ASSORTATIVITY, AND CLUSTERING COEFFICIENT FOR EACH CLASS.

	Complex Network Measures					
Class	Avg. Closeness		Assortativity		Clustering Coeff.	
	$AC^{(\cdot)}$	$AC'^{(\cdot)}$	$r^{(\cdot)}$	$r'^{(\cdot)}$	$CC^{(\cdot)}$	$CC'^{(\cdot)}$
Gray	0.115	0.114	0.064	0.031	0.036	0.035
Black	0.015	0.080	-0.064	-0.136	0.296	0.438

B. Real-World Applications

Here, we give some details about the real-world data sets in which the optimization framework is applied and about the conduction of the experiments. Table IV presents a brief description in terms of number of instances, features and classes of the real-world data sets considered in this work. As a data preparation, each instance attribute vector was normalized to have a magnitude of one and the euclidean distance was used in all simulations as the distance measurement.

TABLE IV
BRIEF DESCRIPTION OF THE REAL-WORLD DATA SETS IN TERMS OF THE NUMBER OF INSTANCES, NUMBER OF ATTRIBUTES AND NUMBER OF CLASSES.

Name	#Instances	#Features	#Classes
AlphaDigits	1404	320	36
PBbr-do	397	2118	8

Each simulation was performed by using a 10-fold stratified cross-validation process. In this process, the data set is split in 10 disjoint sets and, in each run, 9 sets are used as training data and 1 set is used as the test data, resulting in a total of 10 runs. In each run, the training data is divided: 75% as sub-training (X_{train}) and 25% as validation (X_{valid}), such is summarized by Fig. 5. By doing this, we assure an unbiased learning as the testing data is outside of the learning process.

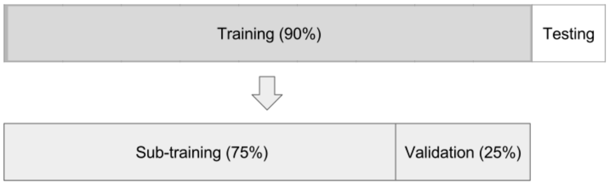


Fig. 5. Division of the training and testing data in the computer simulations.

The proposed optimization framework only has three parameters to be setted, which are related to the SL-PSO algorithm: the size of the swarm population m, the number of generations gen and the maximum number of possible links q. For the computer simulations on the real-world data sets, we use $m = 100$, $gen = 100$ and $q = 5$.

As the hybrid model is also composed by a traditional technique, we have selected the logistic regression (or maximum entropy) algorithm (LR) [26], which is a state-of-the-art algorithm to the Semantic Role Labeling task [27]. The LR parameters are rigorously tunned by using the grid-search method over a large number of combinations: $p = \{l1, l2\}$ for the norm used in the penalization and $C = \{2^{-2}, 2^{-1}, \ldots, 2^{12}\}$ for the regularization strength.

1) Handwritten Digits Recognition: The first problem addressed here is the Handwritten Digits Recognition (HDR), which is a well-known task in data classification. The data set used was the Binary Alphadigits, which is available online[1] and contains binary 20×16 digits of “0” through “9” and capital “A” through “Z”, with 39 samples (images) of each class. Figure 6 shows some sample images of this data set. In the simulations, each image was mapped as a vertex into an underlying a network.

2) Semantic Role Labeling: The second problem addressed here is the Semantic Role Labeling (SRL) [28], which is the task of automatically identifying and classifying the arguments of a predicate (verb) with roles. Such roles indicate meaningful relations among the arguments, as who did what to whom,

[1] http://www.cs.nyu.edu/~roweis/data.html

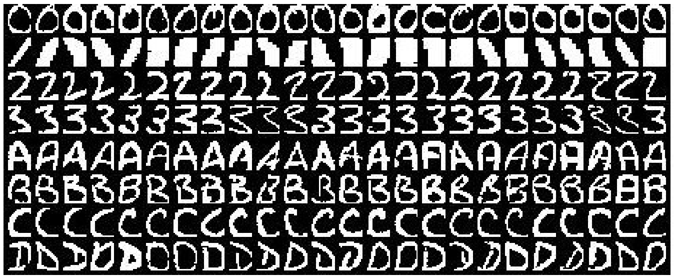


Fig. 6. Sample images of the Handwritten Digits data set.

where, when and how. Motivated by its potential to improve applications in a wide range of Natural Language Processing (NLP) tasks, such as machine translation, question answering, and so on, SRL has received much attention in the last years. In addition, many lexical resources, such as PropBank and FrameNet, have been built to allow the development of efficient semantic role labelers. In the PropBank for example, each predicate is associated with a set of core roles (named A0, A1, and so on) whose interpretations are specific to that predicate and a set of adjunct roles (e.g., manner or location) whose interpretation is common across predicates [29]. Following we present an example about the SRL task through the sentences 1, 2 and 3. In 2, the arguments are identified, and the argument classification is showed in 3.

1. *Seymour Cray can* <u>*do*</u> *it again.*[2]

2. [*Seymour Cray*$_{arg}$] [*can*$_{arg}$] <u>*do*</u> [*it*$_{arg}$] [*again*$_{arg}$].

3. [*Seymour Cray*$_{A0}$] [*can*$_{mod}$] <u>*do*</u> [*it*$_{A1}$] [*again*$_{tmp}$].

In this paper, we focus on the argument classification task using the PropBank.br [30], which is a Brazilian Portuguese corpus that follows the PropBank style. For the experiment, we select all sentences related to the predicate "to do", one of the most frequent verbs in the corpus, and extract the attributes by using a set of features from the literature[3] [28], [31], [32] . As a pre-processing step, argument classes smaller than ten instances were excluded. The metadata of the data set obtained, name here "PBbr-do", can be see in Tab. IV. In the simulations, each argument was mapped as a vertex into an underlying a network.

C. Discussion

Table V presents the average accuracy obtained for the two real-world data sets in the computer simulations. In the table, $\mathcal{C}(LR)$ denotes the logistic regression classifier and $\mathcal{M}(.)$ is the combined classification between LR and the complex network measures (given by (1)), where f means the networks are optimized by the proposed framework; and kNN denotes a baseline in which the network is obtained directly from the kNN network formation method and only the parameters of the hybrid model are optimized by the proposed framework. The results in the table reveal (i) the optimization framework is able to improve the classification given by a state-of-the-art algorithm, which was rigorously tuned; and (ii) in the SRL task, the network obtained through the optimization process provided better result than those obtained from the kNN method. The difference between the results obtained by f and kNN in the HDR task is small. Unlikely "PBbr-do", "AlphaDigits" is a balanced data set, i.e. each class has the same number of images. We believe this can be a relevant information about when to use structural optimization and when the kNN network network could be preferable.

TABLE V
CLASSIFICATION ACCURACY OF LOGISTIC REGRESSION (LR), THE FRAMEWORK WITH NETWORK OPTIMIZATION ($\mathcal{M}(f)$) AND WITHOUT NETWORK OPTIMIZATION ($\mathcal{M}(kNN)$, WITH $k = q$) .

Alg.	AlphaDigits	PBbr-do
LR	68.68 ± 5.04	74.80 ± 4.97
$\mathcal{M}(f)$	**69.72 ± 4.66**	**76.82 ± 3.89**
$\mathcal{M}(kNN)$	69.65 ± 4.74	75.50 ± 5.49

Table VII presents the results of another experiment where f^{r+} and f^{AC+}, which are defined by (13), describe the optimization of the network by combining the predictive accuracy with assortativity and average closeness measures, respectively. As result, one can see the increase of the assortativity or the average closeness while optimizing the predictive accuracy is outperformed by the quality function considering only the predictive accuracy. This give us some evidence that both measures, when composing the quality function, lead the algorithm to converge for local maximum solutions which have lower generalization ability.

TABLE VI
CLASSIFICATION ACCURACY OF THE OPTIMIZATION FRAMEWORK USING DIFFERENT COMPOSITIONS OF THE QUALITY FUNCTION.

Alg.	PBbr-do
$\mathcal{M}(f)$	**76.82 ± 3.89**
$\mathcal{M}(f^{r+})$	76.35 ± 5.27
$\mathcal{M}(f^{AC+})$	76.32 ± 5.04

Certainly, the framework proposed here represents a good effort to promote an applied study of network structural optimization. Furthermore, its results are promising and suggest it is possible to obtain a reasonable improvement in terms of predictive performance, even over rigorously tuned state-of-the art algorithms, by using structural optimization.

Besides the LR technique, the optimization framework is also evaluated with other traditional technique, the Naive Bayes classifier (NB). The results are presented in Tab. VII which reveals the proposed approach is able to consistently improve the classification provided by NB.

Finally, in terms of time complexity, the optimization framework for highlevel classification can be analyzed in three steps: creation of Map matrix which takes $\mathcal{O}(n^2)$ as the euclidean

[2] http://verbs.colorado.edu/propbank/framesets-english/do-v.html

[3] The features used were: FirstForm+FirstPostag, FirstLemma, Head, HeadLemma, TopSequence, PostagSequence, PredLemma+PhraseType, LastForm+LastPostag, PredLemma+Path, FirstPostag, LeftHead, RightHead, VoicePosition, LeftHeadPostag, RightPhrase, and PredLemma.

TABLE VII
CLASSIFICATION ACCURACY USING NAIVE BAYES (NB) AND THE OPTIMIZATION FRAMEWORK. THE LIKELIHOOD OF THE FEATURES IN NB IS ASSUMED TO BE GAUSSIAN.

Alg.	AlphaDigits	PBbr-do
NB	50.23 ± 4.40	56.20 ± 8.00
$\mathcal{M}$	**55.58 ± 3.85**	**60.58 ± 7.53**

distance is calculated between each pair of instances; SL-PSO algorithm which the time complexity in the proposed framework lies on $\mathcal{O}(mn)$; and highlevel classification which is dependant of the network measures adopted (for example, assortativity measure takes $\mathcal{O}(n)$ as the optimized networks are sparse). Generally speaking, the complexity of the proposed framework is low what allows its application for a big number of data classification problems.

V. CONCLUSION

In this work, we presented a bio-inspired framework for structural optimization of networks, which forms the graph by conducting the optimization of a quality function. The framework is applied to the highlevel classification context, where the classification produced by a traditional classifier is combined with the classification provided by complex network measures. Results on experiments conducted on two real-world data sets reveal the network optimization process not only improve the classification given by a rigorously tuned state-of-the-art algorithm, but also outperforms a very well-known and widely-used network formation method. Forthcoming works include the study and application of the optimization framework to other machine learning tasks. We believe there are many graph-based techniques which can benefit from such kind of structural optimization, such as outline detection and dimensionality reduction.

ACKNOWLEDGMENT

Authors thank the financial support given by the São Paulo State Research Foundation - FAPESP (grants numbers 2012/07926-3, 2011/50151-0 and 2013/07375-0). Authors also acknowledge support from the Brazilian Coordination for the Improvement of Higher Education - CAPES, the Brazilian National Council for Scientific and Technological Development - CNPq, and the Minas Gerais State Research Foundation - FAPEMIG.

REFERENCES

[1] M. E. Newman, “The structure and function of complex networks,” *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, “Complex networks: Structure and dynamics,” *Physics reports*, vol. 424, no. 4, pp. 175–308, 2006.

[3] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-worldnetworks,” *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[4] A.-L. Barabási, R. Albert, and H. Jeong, “Scale-free characteristics of random networks: the topology of the world-wide web,” *Physica A: Statistical Mechanics and its Applications*, vol. 281, no. 1, pp. 69–77, 2000.

[5] M. Newman, *Networks: An Introduction*. Oxford University Press, Inc., 2010.

[6] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, “Characterization of complex networks: A survey of measurements,” *Advances in Physics*, vol. 56, no. 1, pp. 167–242, 2007.

[7] R. F. i Cancho and R. V. Solé, “Optimization in complex networks,” in *Statistical mechanics of complex networks*. Springer, 2003, pp. 114–126.

[8] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[9] O. Chapelle, B. Schölkopf, A. Zien *et al.*, *Semi-supervised learning*. MIT press Cambridge, MA:, 2006, vol. 2.

[10] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40 –51, 2007.

[11] X. Zhu, “Semi-supervised learning literature survey,” University of WisconsinMadison, Tech. Rep., 2008.

[12] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3–5, pp. 75 – 174, 2010.

[13] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, “A survey of statistical network models,” *Foundations and Trends® in Machine Learning*, vol. 2, no. 2, pp. 129–233, 2010.

[14] T. C. Silva and L. Zhao, “Network-based stochastic semisupervised learning,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 451–466, 2012.

[15] J. R. Bertini, L. Zhao, R. Motta, and A. A. Lopes, “A nonparametric classification method based on k-associated graphs,” *Information Sciences*, vol. 181, no. 24, pp. 5435–5456, 2011.

[16] T. C. Silva and L. Zhao, “Network-based high level data classification,” *IEEE Transactions on Neural Networks*, vol. 23, pp. 954–970, 2012.

[17] M. G. Carneiro, T. H. Cupertino, and L. Zhao, “K-associated optimal network for graph embedding dimensionality reduction,” in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 1660–1666.

[18] M. G. Carneiro, J. L. Rosa, A. A. Lopes, and L. Zhao, “Network-based data classification: combining k-associated optimal graphs and high-level prediction,” *Journal of the Brazilian Computer Society*, vol. 20, no. 1, p. 14, 2014.

[19] T. H. Cupertino, L. Zhao, and M. G. Carneiro, “Network-based supervised data classification by using an heuristic of ease of access,” *Neurocomputing*, vol. 149, pp. 86–92, 2015.

[20] R. Cheng and Y. Jin, “A social learning particle swarm optimization algorithm for scalable optimization,” *Information Sciences*, vol. 291, pp. 43–60, 2015.

[21] Cai, Qing and Ma, Lijia and Gong, Maoguo and Tian, Dayong, “A survey on network community detection based on evolutionary computation,” *Int J Bio-Inspired Comput*, vol. 45, no. 2, pp. 167–256, 2014.

[22] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social networks*, vol. 1, no. 3, pp. 215–239, 1979.

[23] M. E. Newman, “Assortative mixing in networks,” *Physical review letters*, vol. 89, no. 20, p. 208701, 2002.

[24] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[25] J. Kennedy, *Swarm Intelligence*. Springer, 2006.

[26] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[27] M. Bazrafshan and D. Gildea, “Semantic roles for string to tree machine translation.” in *ACL (2)*, 2013, pp. 419–423.

[28] D. Gildea and D. Jurafsky, “Automatic labeling of semantic roles,” *Computational Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[29] M. Palmer, D. Gildea, and P. Kingsbury, “The proposition bank: An annotated corpus of semantic roles,” *Computational Linguistics*, vol. 31, pp. 71–106, 2005.

[30] M. S. Duran and S. M. Alusio, “Propbank-br: a brazilian treebank annotated with semantic role labels,” *International conference on Language Resources and Evaluation (LREC)*, pp. 1–6, 2012.

[31] F. E. Alva-Manchego and J. L. G. Rosa, “Semantic role labeling for brazilian portuguese: A benchmark,” in *Advances in Artificial Intelligence–IBERAMIA 2012*. Springer, 2012, pp. 481–490.

[32] S. S. Pradhan, W. Ward, and J. H. Martin, “Towards robust semantic role labeling,” *Computational Linguistics*, vol. 34, no. 2, pp. 289–310, 2008.