Neurocomputing 149 (2015) 86–92

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Network-based supervised data classification by using an heuristic of *ease of access*

Thiago H. Cupertino [a,*], Liang Zhao [b], Murillo G. Carneiro [c]

[a] Institute of Mathematical Sciences and Computing - University of São Paulo, Av. Trabalhador São-carlense 400, São Carlos, SP 13560-970, Brazil
[b] School of Philosophy, Science and Literature in Ribeirao Preto - University of São Paulo, Av. Bandeirantes 3900, Ribeirão Preto, SP 14040-900, Brazil
[c] Faculty of Computing - Federal University of Uberlândia, Av. João Naves de Ávila 2160, Bloco B, Uberlândia, MG 38400-902, Brazil

ARTICLE INFO

Article history:
Received 24 May 2013
Received in revised form
4 February 2014
Accepted 17 March 2014
Available online 30 July 2014

Keywords:
Network-based learning
Data classification
Supervised learning
Random walk
Limiting probabilities
Steady states

ABSTRACT

We propose a new supervised classification technique which considers the *ease of access* of unlabeled instances to training classes through an underlying network. The training data set is used to construct a network, in which instances (nodes) represent the states that a random walker visits, and the network link structure is modified by performing a link weight composition between the unlabeled instance bias and the initial network link weights. Different from traditional classification heuristics, which divide the training data set into subspaces, the proposed scheme uses random walk limiting probabilities to measure the limiting state transitions among training nodes. An unlabeled instance receives the label of the class that is most easily reached by the random walker, that is, the limiting transition to that class is large. Simulation results suggest that the proposed technique is comparable to some well-known classification techniques.

1. Introduction

Supervised machine learning comprises the construction of a predicting model by using information extracted from a training data set. The constructed model defines decision borders that are used to classify unlabeled data [1]. An unlabeled instance is classified depending on its relative position to the decision borders. Due to its importance in various real applications, many classification techniques have been developed, such as Linear Discriminant Analysis (LDA) [1], Neural Networks [2], *k*-Nearest Neighbors (*k* NN) [3], Naive-Bayes [4], Support Vector Machines (SVM) [5] and Decision Trees [6]. These traditional classification techniques divide the data space according to physical features of the training data (similarity, distance, or distribution). In this way, many intrinsic and semantic relations among data items are ignored, for example topological structures and pattern formation.

On the other hand, the usage of an underlying network can do take into account the previously mentioned relationships among data, that is, topological structures and pattern formation. Networks are powerful tools for complex system modeling and for data representation. By using this representation, the structure, dynamics and functions of the system that it represents are unified. Also besides describing the interaction among nodes (structure) and the evolution of such interactions (dynamics), it also reveals how "structure + dynamics" affects the overall function of the network [7]. Due to these characteristics, the usage of network-based methods in learning tasks has been increasing over the past years and has become a very active research area with a myriad of applications, such as semi-supervised learning [8,9], clustering [10–12], regression [13], feature selection [14], and dimensionality reduction [15], among others.

Another relevant advantage when using networks in learning tasks is that it can perform quite different classification heuristics. Traditional classification techniques divide the data space into subspaces, each one representing a data class. These subspaces are not overlapped in the case of crisp classification, but they can be slightly overlapped in the case of fuzzy classification. In either way, strong twisting or largely overlapped subspaces are not permitted. However, when using networks many are the classification approaches. For instance, the authors in [16] consider that unlabeled instances belong to a sub-network (class) which results in the lowest modularity value [17] after connecting it to a network constructed from the unlabeled set. In their approach, the classification process is performed by considering the *connectivity pattern* of the training data. In a recent study [18], the authors propose the use of node centrality for data classification. Their technique is capable of classifying multiple observations where each pattern is represented by a group of invariant transformations. The classifier must predict

* Corresponding author.
E-mail address: thcupertino@gmail.com (T.H. Cupertino).

http://dx.doi.org/10.1016/j.neucom.2014.03.071
0925-2312/

the pattern this group belongs to. In this approach, the classification is conducted by analyzing *how central* or *how important* is a test instance to each class.

Here, we propose a new network-based classification technique which considers the *ease of access* of unlabeled instances to training classes. Different from the previous works, the proposed technique uses the dynamical process measure called the random walk limiting probabilities. Limiting probabilities are applied to random walk processes to measure the limiting state transitions through an underlying network [19]. In the proposed scheme, the training data set is used to construct the network, in which instances (nodes) represent the states that a random walker visits during the process. An unlabeled instance is considered to be belonging to the class that is most easily reached, that is, the limiting transition probability for a random walker to that class, after the insertion of the unlabeled instance link bias into the training network, is large. As a consequence of the dynamical processes, both local and global relationships among nodes are taken into account.

This paper is organized as follows: Section 2 describes the model for the supervised classification technique, and Section 3 analyses the algorithm complexity. In Section 4, simulation results and comparisons to others techniques are presented. Finally, Section 5 concludes the paper.

2. Technique description

The general idea of the proposed technique is explained as follows. Random walk theory can be understood in terms of Markov chains [19]. A Markov chain is formed by a sequence states visited by a random walker in which the probability to visit a given state is independent of past visits given that the current state is known. The probability of moving from one state to another is called the transition probability. It can be shown that, under some conditions, after an infinite number of transitions, the random walk process reaches the stationary state, or the limiting probabilities, which is independent of the initial state [20]. In this situation, states that have larger transition probabilities to other states result in larger limiting probabilities, that is, the random walk has some preference to visit them. In other words, by representing the states and transition probabilities as network nodes and link weights, respectively, we can say that nodes which are better linked to the other nodes or have stronger link weights result in larger limiting probabilities. In this case, the random walker prefers to visit some nodes in the network in detriment of others, that is, some nodes are easily accessed than others. To classify a given unlabeled instance, a set of labeled instances is considered as network nodes, or the state space set, that is, each node (labeled instance) is a possible state for the random walker. This network of labeled nodes is modified by a specific link weight composition which takes into account the bias information of the unlabeled instance to be classified. The bias information changes the network structure by affecting the link weights among nodes, resulting in a structure such that the most easily reached labeled nodes represent the class label of the unlabeled instance after the calculation of the limiting probabilities in the biased network. The mathematical formulation is as follows.

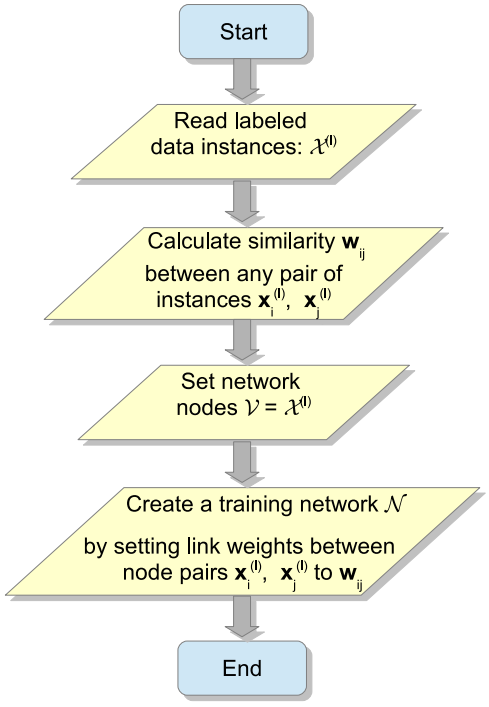

Fig. 1. Flowchart for the training phase.

The classification problem concerned within this paper requires a given labeled data set, $\mathcal{X}^{(l)} = \{\mathbf{x}_i^{(l)},\ i = 1, \ldots, n\}$, where each instance is described by q attributes $\mathbf{x}_i^{(l)} = \{x_{i1}, x_{i2}, \ldots, x_{iq}\}$. Each instance in this set has a single assigned label $l \in \{1, 2, \ldots, \mathcal{L}\}$. It is also given an unlabeled data set, $\mathcal{X}^{(u)} = \{\mathbf{x}_i^{(u)},\ i = 1, \ldots, m\}$, containing instances that will be assigned to labels after classification. The proposed technique can then be divided into two phases, training and classification, as it is described next.

Training phase: In the training phase, a weighted and undirected network $\mathcal{N} = \{\mathcal{V}, \mathcal{E}\}$ is constructed without self-loops. Nodes represent data instances, $\mathcal{V} = \mathcal{X}^{(l)}$, and link weights represent similarities among instances, $\mathcal{E} = [\mathcal{W}_{ij}],\ i, j = 1, \ldots, n$. The similarity between any pair of instances $\mathbf{x}_i^{(l)}$ and $\mathbf{x}_j^{(l)}$ is denoted by w_{ij}. The network similarity matrix $\mathcal{W} = \{w_{ij}\}$ can be calculated by using any distance function. Specifically, we use the Euclidean distance in all experiments in this paper. At the end of this phase, we get a network $\mathcal{N}$ called the training network. The flowchart in Fig. 1 depicts the training phase.

Classification phase: To classify an unlabeled instance $\mathbf{x}^{(u)}$, a weight vector $\mathcal{S} = [s_1, s_2, \ldots, s_n]$ is calculated containing the link weights between $\mathbf{x}^{(u)}$ and all other nodes $\mathbf{x}_i^{(l)}$. That is, node $\mathbf{x}^{(u)}$ is inserted into the training network $\mathcal{N}$ by calculating the link weights to all other nodes into this network, and is subsequently removed from $\mathcal{N}$. Then, an asymmetric and $n \times n$ modified similarity matrix $\hat{\mathcal{W}}$ is constructed by the following composition:

$$\hat{\mathcal{W}} = \mathcal{W} + \epsilon \hat{\mathcal{S}}, \tag{1}$$

where ϵ is a non-negative parameter and $\hat{\mathcal{S}}$ is the following $n \times n$ matrix:

$$\hat{\mathcal{S}} = \begin{bmatrix} \mathcal{S}_{(1)} \\ \mathcal{S}_{(2)} \\ \vdots \\ \mathcal{S}_{(n)} \end{bmatrix}.$$

It can be observed in Eq. (1) that the weight biases of the unlabeled instance $\mathbf{x}^{(u)}$, encoded in matrix $\hat{\mathcal{S}}$, are applied over all links $\mathcal{W}$ of the training network $\mathcal{N}$, that is, the weight of each link is linearly added up with its corresponding weight bias. The idea behind this operation is that the distance between any pair of nodes is modified due to the new weights of network routes introduced by the insertion of the link biases from the unlabeled instance links. The higher the similarity between the unlabeled instance and a node, say node i, the more strengthened the connections from all other nodes to node i are after this operation. The parameter ϵ controls the influence of the weight biases in the