2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Confidence-based Ant Random Walks

Ping He, Lin Lu, Xiaohua Xu*, Kanwen Li, Heng Qian, and Wei Zhang

Abstract— **To facilitate the computer-aided medical applications, this paper tries to build better intelligent diagnosis systems with the help of swarm intelligence method. As to the clinical data, a built-in graph structure is constructed with training samples being mapped as labeled vertices and test samples being unlabeled vertices. On the basis of the iterative label propagation algorithm, this paper first introduces a confidence-based random walk learning model, where unlabeled vertices that consistently show high probability (above the confidence threshold) in belonging to one class is treated as labeled vertices in the next iteration. Later motivated by the swarm intelligence, this model is further improved by treating the labeled vertices as real ants in nature and the predefined classes as different ant colonies. A novel labeled ant random walk algorithm is introduced by incorporating the history information of random walk in the form of aggregation pheromone. The proposed algorithms are evaluated with a synthetic data as well as some real-life clinical cases in terms of diagnostic accuracy. Experimental results show the potentiality of the proposed algorithms.**

I. Introduction

In the field of human decision, medical diagnosis is naturally a classification or prediction problem for the sake of determining further treatment. With the development of diagnostic knowledge and herb formula of Traditional Chinese Medicine (TCM) in China, there are a huge set of clinical cases are available up to now. Some of these cases are represented by tests obtained from modern equipments, while the others are represented by symptoms from the TCM methods (e.g. Inspection, auscultation and olfaction, inquiry and palpation). Therefore, researches on these clinical cases may help promote the development of diagnosis technology, and make a contribution to the objection and modernization of TCM.

The computer-aided disease diagnosis can be easily molded as classification problem [1] in artificial intelligent (AI) domain. For example, by analyzing the symptoms of a patient based on the empirical clinical cases, the computer can provide a prediction of an initial diagnosis, which further helps the doctor for final decisions. The classification problem has been studied by many machine learning methods for a long time. Machine learning is the study of computer algorithms that improve automatically through experience and has been central to AI research since the field's inception [2].

A crucial problem of machine learning is to automatically learn to recognize complex samples and make intelligent decisions based on previous data. Due to their effectiveness and good performance in real-world applications, many algorithms[3], [4] have been developed and largely employed to solve problems in application such as pattern recognition, medical diagnosis, bioinformatics, and even syndrome differentiation [5] in TCM.

To solve this problem, this paper first studies the graph presentation of data samples. Graph is a very important method for data modeling, since many real-world data sets have built-in graph topology. So is the disease data, where each case can be a point in high dimensional space denoted by multiple symptoms or properties. Provided a graph structure for the data points, it is easy to introduce local consistency to previous algorithms by modifying the objective function under the regularization framework or even to present novel classification algorithm with better performance.

Learning with graph is quite appealing recently, as for the classification, label propagation [6] via random walks on graph is one typical graph-based machine learning method. In such method, the training data points are considered as the labeled vertices and the test data points are treated as unknown vertices to be labeled. The labeled ones, during their Markov random walk [7] on this graph, are able to propagate their labels to the unlabeled vertices. This idea is intuitive and iteration-based, though the analysis equation can be derived from the iteration equation, it involves complex matrix operation that affects its scalability on large data set.

Therefore, we improve the basic iterative solution of the label propagation process and propose a confidence-based random walks (CRW) Model. In this model, a confidence value is pregiven so that after a certain number of iterations, some of the highly reliable unlabeled vertices that consistently belong to one certain class are added into the training set for the next iteration. By doing so, the unlabeled vertices will be gradually assigned a label and later contribute to the successive label propagation process and also ensure early convergence as long as there is no more label assignment.

Besides the CRW model, we note that the label propagation process doesn't treat the labeled nodes as any intelligent agent such as the ant, bee and bird in swarm intelligence, while they share many in common. In view of this, we then try to combine the the swarm intelligence idea for further improvement of the CRW algorithm. Swarm intelligence (SI)[8] represents a class of bio-inspired algorithms for complex optimization problems. It studies the collective behavior of systems composed of many individuals inter-

* Corresponding author: Department of Computer Science, Yangzhou University, Yangzhou 225009, Ph: +8651487978309, Fax: +8651487887937, e-mail: arterx@gmail.com

This work was supported in part by the Chinese National Natural Science Foundation under Grant nos. 61003180, 61379066 and 61103018, Natural Science Foundation of Education Department of Jiangsu Province under contracts 13KJB520026 and 09KJB20013, Natural Science Foundation of Jiangsu Province under contracts BK2010318, Research Innovation Program for College Graduates of Jiangsu Province (CXLX12_0917), and New Century Talent Project of Yangzhou University.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

acting locally with each other and with their environment. Swarms inherently use forms of decentralized control and self-organization to achieve their goals. SI algorithms observe the social behaviors of different swarm of animals and insects such as ants, termites, bees, birds, fishes in nature.

In addition, the individuals in such biological swarms, are by no means sophisticated engineers, but instead are simple creatures with limited cognitive abilities and limited means to communicate. Yet the complete swarm exhibits intelligent behavior, thus providing efficient solutions. Providing simple intelligence and random walk rule, we believe the label propagation process can be more efficient and intelligent. For instance, the labeled vertex can be modeled as an ant from one ant colony representing a class of label, it moves randomly to neighboring vertices and leaves the pheromone on them. This specific pheromone thus like the smell or excreta indicating the possession of this vertex. In other words, this unlabeled vertex will probably be labeled as that ant's colony.

As a result, we present the confidence-based ant random walk model. To begin with, the model is first trained with a small amount of labeled (samples) ants. Afterwards, the classifier is used to classify the unlabeled samples; and then among the unlabeled vertices, the high confidence ones are determined in the same manner as the CRW. Then they are added (together with their predicted labels) to the corresponding (class) colony in the training set. The classifier is re-trained (using the newly formed training set) and this procedure is repeated until colony formation is stabilized. In this way, a new enlarged training set is built. Once the colony formation is stabilized, in the last, each test sample is evaluated to assign to the colony for which the average aggregation pheromone density is more. At last, both our proposed models are evaluated and applied in real medical diagnostic scenarios for the sake of correctness and effectiveness demonstration.

The remainder of this paper is organized as follows. We first provide some preliminary knowledge in section 2, and then present our initial algorithm Confidence-based Random Walk in section 3, and the further-improved algorithm Confidence-based Ant Random Walk in section 4. They are followed by the performance evaluation in section 5. Section 6 concludes the whole paper.

II. Prerequisites

A. Problem definition

Usually, the background of classification is defined as follows. Given a data set $X = X_m \bigcup X_u = x_1 ... x_m, x_{m+1}, ..., x_n$, where $X_m = (x_i)_{i=1...m}$ is labeled, corresponding to the known label subset $Y_m = (y_i)_{i=1...m}$, and $Xu = (x_i)_{m+1...n}$ is unlabeled, corresponding to the unknown label subset $Y_u = (y_i)_{m+1...n}$. The number of test data is u satisfying $m + u = n$. The label y_i of each sample x_i is chosen within a set of c predefined labels, which are supposed to be available in a fixed permutation order.

Then without caring about the type (Integer, String or others) of specific labels, we let Y be a vector with its element denoting the position of corresponding label in this ordered label set. That is to say, $y_i = j$ indicates that the label of sample x_i is the j^{th} label in the given permutation order. The classification problem then turns to predicting the Y_u given the above prior knowledge.

B. Graph representation and construction

Graph-based methods construct a graph where the vertices designate the labeled and unlabeled samples of the data set and edges represent the similarity (distance) of samples. These methods are naturally nonparametric, discriminative, and transductive, and thus triggered some graph based learning methods like harmonic [11], local and global consistency [9], and our lately introduced component random walk model [10]. To our knowledge, a few recent graph-based semi-supervised classifiers are proposed particularly for graph construction [12], handling multiple graphs in gene networks [13], neighborhood graph construction [14], betweenness computation on large sparse directed graphs [15] and protein localization [16].

Constructing a graph from raw data is usually based on a well-defined distance (or similarity) measure between a pair of data points. Most traditional learning algorithms require that all the data points are numerical vectors of the same length. For example, in text classification, a document has to be converted to a vector representation before classifiers can be applied.

Suppose a graph $G = (V, E, w)$ consists of a vertex set V and an edge set E. In machine learning problems, the vertex set is equal to the sample set, i.e. $V = X$. And the graph is represented as an affinity matrix W. A popular weight function is the Radius Basis Function (RBF) kernel, which is defines as:

$$w_{ij} = w(x_i, x_j) = e^{-\gamma d_{ij}^2} \tag{1}$$

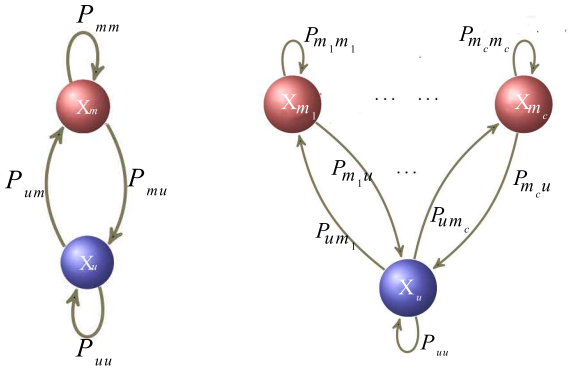
where d_{ij} is the distance between x_i and x_j and γ is a predefined parameter. Due to the existence of multiple labels, a $n \times c$ label indicating matrix $F = [F_m^T \quad F_u^T]^T = (f_{ij})_{n \times c}$ is defined, where F_m and F_u respectively denote the states of the known labeled set X_m and the unknown set X_u. At the beginning, F is initialized so that $f_{ij} = 1$ if and only if $y_i = j$, and $f_{ij} = 0$ if $y_i \neq j$ or $y_i = 0$.

III. Confidence-based Random Walks

A. Random walks on graph

Recent studies on spectral graph theory and manifold learning theory have demonstrated that the local geometric structure can be effectively modeled through a nearest neighbor graph on a scatter of data points. Basically, the functions and operators on a manifold can be approximated by their discrete counterparts on the corresponding graph. For example, graph Laplacians have been successfully applied in semi-supervised learning, clustering, and data representation.

Given a partially labeled graph, the process of assigning labels is iteratively done by randomly propagating the labels of already labeled vertices to those unlabeled vertices. The



(a) (b)

Fig. 1. Different views of data set

label propagation via random walks is usually based on a probabilistic transition matrix P. Here, Each element of P, $p_{ij} \in [0, 1]$, contains the possibility of propagating the class information from vertex v_i to vertex v_j. Usually the construction of P involves a normalized similarity matrix $P = D^{-1}W$, where $D = diag(W\mathbf{1}_{n\times n})$, $\mathbf{1}_{n\times n}$ is a $n \times n$ square matrix with all entries are one.

The evolution of F depends on transition matrix P can be represented in the iterative form

$$F^{t+1} \leftarrow PF^t \tag{2}$$

where F^t represents the state of F at time step t, similarly F^{t+1} represents the state of F at time step $t + 1$.

In the context of semi-supervised learning, since labeled samples bear discriminative information while unlabeled samples do not, then the transition probabilities of labeled samples are distinct from those of unlabeled samples. Consequently, The transition matrix P is decomposed into four sub-blocks:

$$P = \begin{bmatrix} P_{mm} & P_{mu} \\ P_{um} & P_{uu} \end{bmatrix} \tag{3}$$

where, as can be seen in Fig 1(a), P_{mm} is the transition submatrix from F_m to F_m, P_{uu} is the transition submatrix from F_u to F_u, P_{mu} and P_{um} are the transition submatrices from F_m and F_u to each other.

Under such circumstances, the state transition Eq. 2 then can be decomposed into two iterative parts:

$$F_m^{t+1} = P_{mm}F_m^t + P_{mu}F_u^t \tag{4}$$
$$F_u^{t+1} = P_{um}F_m^t + P_{uu}F_u^t \tag{5}$$

To ensure the labeled data unaffected by the unlabeled ones, we set $P_{mm} = I$ and $P_{mu} = O$. As a result, we have $F_m^{t+1} = F_m^t$, further it can be deduced that $F_m^t = F_m^0$. Such iterative process is intuitively shown in Fig. 2, as can be seen, the next state of F_u^{t+1} is determined by its previous state F_u^t and the initial state of labeled samples F_m^0. Our goal is to predict F_u when this process converges.

This diagram is the basis of our firstly proposed algorithm [10], to be discussed later. When the evolution of state

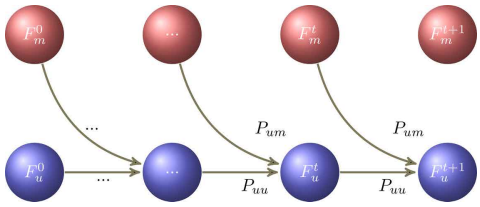


Fig. 2. The state transition diagram

matrix approximates infinity, it is easy to deduce the analytic solution of the above iterative solution, which goes that

$$F_u = (I - P_{uu})^{-1}P_{um}F_m \tag{6}$$

As to this equation, the inverse operation only works when $(I - P_{uu})$ is non-singular, resulting in a limited classification performance and application.

B. Confidence-based random walks

According to eq. (6), when the inverse of $(I - P_{uu})$ in the analytic solution is intractable, the iterative solution of label propagation algorithm is more useful. Therefore, instead of studying the analytic solution of label propagation via random walks, we pay more attention to improving the iterative solution, based on the observation that some unlabeled nodes continuously display high probability in belonging to one class during the iteration.

High probability grants us with high confidence. Hence, we take the inspiration that after a period of iteration, it is better to label such 'high confidence' unlabeled nodes and add them into training set for next iteration and for better and faster random walk learning. To achieve this, we add a self-training stage into the basic model. The detailed algorithm description is illustrated in Algorithm 1.

In the beginning, all entries in the unknown state matrix are initially assigned with equal possibility ($\epsilon = 1/c$) to make sure that each row sums to 1 (step 1). Then we proceed to the random walks, where we repeat Δt times of the state evolution (step 4-8) at each time step t to avoid a premature convergence. This local loop also helps to obtain more stable states.

During the self-training stage (step 10-16), the highly reliable and stable unlabeled vertices are selected as training samples. In detail, we search for the largest and second largest values f_{ij} and f_{ik} of each row $F_u^{t+1}(i, :)$. The two values indicate the two most possible classes j and k that v_i belongs to. The ratio f_{ik}/f_{ij} represents the probability of the unlabeled vertex v_i belonging to class k in comparison with class j. The smaller the ratio, the higher confidence of the unlabeled data belonging to the most probable class j. The larger the ratio, the less confidence in assigning the unlabeled data to a specific class. Therefore, a confidence value θ is introduced so that if such ratio between the two largest values is less than or equal to θ , the unlabeled vertex becomes a 'high confidence' vertex that will be added to the training set for the next iteration. Otherwise, it will not be included in any class.

Algorithm 1 Confidence-based Random Walks (CRW)

Input: $X, Y_m, \Delta t$ and confidence value θ
Output: Y_u.

Initialize $F_u^0 = \mathbf{1}_{u\times u} * \epsilon$, build graph $G = (V, E, w)$
$t = 0$;
while not StoppingCriteria **do**
  $F_u^{\tilde{t}} = F_u^t$ (For local iteration)
  for $\tilde{t} = 1$ to Δt **do**
    $F_u^{\tilde{t}+1} = P_{um}^t F_m^t + P_{uu}^t F_u^{\tilde{t}}$
    Row-Normalize $F_u^{\tilde{t}+1}$ according to Eq. 7:

$$F_u^{\tilde{t}+1}(i,:) = \frac{F_u^{\tilde{t}}(i,:)}{\sum_{k=1}^{c} F_u^{\tilde{t}}(i,k)} \tag{7}$$

  end for
  $F_u^{t+1} = F_u^{\tilde{t}}$
  for each row $F_u^{t+1}(i,:)$ **do**
    Find the largest value $F_u^{t+1}(i,j)$ and second largest value $F_u^{t+1}(i,k)$.
    if $F_u^{t+1}(i,k)/F_u^{t+1}(i,j) \le \theta$ **then**
      Add this row into F_m^t as new known state matrix:

$$F_m^{t+1} = \begin{bmatrix} F_m^t \\ F_u^{t+1}(i,:) \end{bmatrix} \tag{8}$$

      Remove the i^{th} row from F_u^{t+1}
    end if
  end for
  Update P_{mm}^{t+1}, P_{mu}^{t+1}, P_{um}^{t+1}, P_{uu}^{t+1} with respect to F_u^{t+1} and F_m^{t+1}
  $t = t + 1$
end while
Assign labels to V according to F_u^t and F_m^t

The training phase of this algorithm stops when there is no (re)assignment of 'high confidence' unlabeled vertices. Then the label of an arbitrary vertex v_i is computed as

$$y_i = \text{argmax}_j \ f_{ij} \tag{9}$$

IV. Confidence-based Ants Random Walks

A. Combining historic information in form of Aggregation Pheromone

The random walk on graph holds the memoryless property: The future behavior of a Markov chain depends only on its current state, but not on how it arrived at the present state. However, the intermediate states denoting how the initial vertex transits to the current vertex is also worthy of consideration. In our concern, keeping memory of the walking trace can help select optimal transition paths during the iteration and thus help produce a better classification performance. This is proved by many swarm intelligence algorithms [8], such as ant colony optimization (ACO) [17] and Aggregation Pheromone Systems (APS) [18], which maintain a pheromone matrix for recording the historic random walk (searching) information so as to find an optimal solution.

ACO and APS are computational algorithms modeled on the behavior of ant colonies. ACO algorithms are designed to emulate ants' behavior of laying pheromone on the ground while moving to solve optimization problems. Pheromone is a type of chemical emitted by an organism to communicate between members of the same species. Aggregation pheromone is termed due to the clumping or clustering behavior in a species that brings individuals into closer proximity. Thus, aggregation pheromone causes individuals to aggregate around good positions which in turn produces more pheromone to attract individuals of the same species.

The further improving idea of our first proposed CRW algorithm is triggered based on the aggregation pheromone density. In earlier work, attempts have been made for solving clustering [19] and classification [20] showing encouraging results. In view of the recent research [21], the rest of this paper will propose a novel confidence-based ant random walks (CARW) classification algorithm using aggregation pheromone.

B. Model building

We take into account of different classes for rebuilding our graph. Since the training data are divided into c classes and a small number of labeled data from each class, by our assumption, forms C homogeneous groups or colonies of ants in the training/labeled set X_m. Therefore, it is more reasonable to partition the whole training set into multiple subsets according to the label distribution, as is illustrated in Fig. 1(b). Each subset with the same label l denoted by $X_{m_l}(\subset X_m)$ is treated as an ant colony C_l. Let, $x_1^l, x_2^l, \cdots, x_{|C_l^0|}^l$ be the given original training data or labeled data samples in the l^{th} initial training class C_l^0. These samples are considered as a population of $|C_l^0|$ number of ants represented as $a_1^l, a_2^l, \cdots, a_{|C_l^0|}^l$. Hence, an ant a_i^l represents the i^{th} training data in the l^{th} initial training class $x_i^l \in C_l^0$.

Due to the fact that different labeled subset may contain different number of instances thus causing the imbalance of training data, instances belonging to same class are integrated into a single nest vertex v_l. In other words, we let vertex v_l be the representation of labeled subset X_{m_l} or colony C_l, and let vertex set $V_c = (v_l)_{1,..,c}$ contains the c nest points. Meanwhile, for the unlabeled instances, we let them be food resources scattered around these nests. Instead of being a single vertex, a connected graph is constructed over X_u. Accordingly, we let $V_u = (v_i)_{c+1,..,c+u}$ be the set of unlabeled vertices. Then given that P_{uu} is the transition submatrix from unlabeled instances to themselves, we assume that there is an edge linking a pair of vertices only if pairwise transition probability exists. Fig. 3 shows a simple model composed of two nests (labeled data sets belonging to different classes) and three resources (unlabeled data).

However, the transition probabilities from nests to all the resources are still unknown. We define P_{cu} to be the transition matrix from c nests to u resources, as we have known that P_{mu} denotes the transition probabilities from

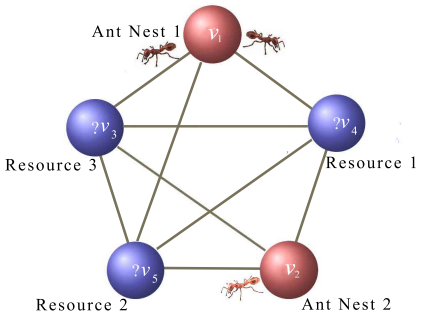


Fig. 3. A simple example of binary-class ant colonies

all labeled instances to unlabeled instances, so we use the following equation to compute P_{cu} with the supervised information of F_m

$$P_{cu} = (P_{um}F_m)^T \tag{10}$$

C. Aggregation pheromone definition

In this paper, we focus on the collective strength (aggregation pheromone) rather than the individual pheromone. The pheromone matrix is defined as a $u \times c$ matrix τ, where each column vector is assumed to be the accumulation of pheromone on all resources by one nest. For instance, given a nest label l at the t^{th} generation, $\tau^t_{*l} = \tau^t(:, l)$ is the corresponding column vector with its element $\tau^t_{jl} = \tau^t(j, l)$ indicates the vertex v_j's pheromone left by the ants from nest l. At first, the pheromone matrix is initialize as $\tau^0 = F^0_u$ (see in Algorithm 1). By doing so, once the algorithm converges, we can directly assign the labels to unlabeled data according to this matrix. Since ants from different nests may attach different types of pheromone to a vertex, when choosing the next target, ants are not only attracted by its type of class pheromone but also resisted by other classes. Therefore, the relative pheromone level is more useful, and the pheromone matrix is normalized in the following way so as to consider such influence.

$$\tau^t(j, l) = \frac{\tau^t(j, l)}{\sum_{k=1}^{c} \tau^t(j, k)} \tag{11}$$

This operation is done after the pheromone update and before the next generation of colony random walk. After many generations of such colony random walk, one type of pheromone from a certain nest may dominate that vertex. Thus for the other population of ants, they are unlikely to choose this vertex though the heuristic transition probability is appealing.

Consider ants from one colony or samples with the same label, since we pay more attention to aggregation pheromone rather than individual pheromone, now we define the increment of aggregated pheromone on the j^{th} unlabeled vertex v_j due to l^{th} training colony at iteration t as $\Delta\tau^t_{jl}$. We also define $\Delta\tau^t$, whose column vector $\Delta\tau^t(:, l)$ indicates the increment of the collective emitted pheromone on all unlabeled vertices by the ant colony l at iteration t.

Fig. 5 depicts the ants random walk mechanism of computing the aggregated pheromone increment. Thereafter, the pheromone density τ^t_{jl} is updated at iteration $t+1$ using the following equation:

$$\tau^{t+1} = (1 - \rho)\tau^t + \Delta\tau^t \tag{12}$$

Particularly, for an output of a nest with label l, the update equation is:

$$\tau^{t+1}(j, l) = (1 - \rho)\tau^t(j, l) + \Delta\tau^t(j, l) \tag{13}$$

where $\rho \in [0, 1)$ is the evaporation constant. With smaller values of ρ, the model uses more information of the pheromone density of the past cycles. Larger value of ρ indicates that the effect of the pheromone emitted in the present iteration is more important than the pheromone in the previous iterations. ρ acts as a trade-off factor of the emitted pheromone in the previous and present iterations. Instead of keeping it constant throughout the self-training process, we make it vary with time. Then ρ becomes a function of time and we define it as:

$$\rho = \frac{1}{2 + log(t + 1)} \tag{14}$$

By doing so, the initial evaporation value at $t = 0$ is 0.5 and it will decrease with time goes by.

D. Probabilistic transition rule

For metaheuristic algorithms, the heuristic evaluation function is necessary because mostly the initial pheromone matrix can not guide the optimization process, while heuristic evaluation functions can provide a quality measurement for different solutions. Traditional ACO for traveling salesman problem treats the inverse of pairwise distance to be the heuristic value. However in our paper, the transition probability of random walk on graph is more appropriated to be the heuristic value. Specifically we let P_{cu} and P_{uu} be the heuristic values, which respectively indicate the transition matrix from V_c to V_u and P_{uu} and from V_u to V_u. Therefore, the heuristic transition matrix from $V_c \cup V_u$ to V_u at iteration t becomes

$$\eta^t = \begin{bmatrix} P^t_{cu} \\ P^t_{uu} \end{bmatrix} \tag{15}$$

If an ant starts to move out from its nest l, then the heuristic value guiding it to the resource vertex v_j is $\eta(l, j) = P_{cu}(l, j)$. If an ant starts from the vertex v_i, then the heuristic value guiding it to its neighbor v_j is $\eta(i, j) = P_{uu}(i, j)$. Moreover, for an arbitrary nest l, let $\eta(l, :)$ be the l^{th} row of matrix P_{cu}, its related transition matrix is

$$\eta^t_l = \begin{bmatrix} P^t_{cu}(l, :) \\ P^t_{uu} \end{bmatrix} \tag{16}$$

Finally, the ants determine their paths based on the combination of the heuristic value and the strength of pheromone trail. That is to say, given a vertex pair $\langle v_i, v_j \rangle$, the probability of ants moving from v_i to v_j at the t^{th} generation is

$$q^t_l(v_i, v_j) = \tau(j, l)^t + \eta^t_{i,j} \tag{17}$$

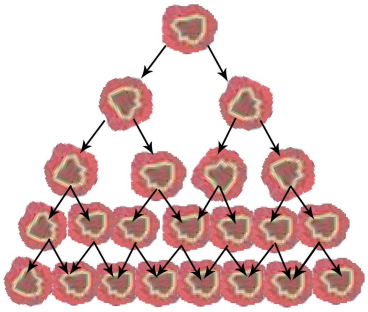


Fig. 4. The spanning tree of breast cancer cells

Each nest maintains a transition rule for its random walk; we also provide the integrated transition matrix form for nest l at generation t

$$Q_l^t = [\mathbf{1}_{(u+1)\times 1}(\tau_{*l}^t)' + \eta_{l*}^t] = \begin{bmatrix} \mathbf{1}_{(u+1)\times 1}(\tau_l^t)' + P_{uu}^t \\ (\tau_l^t)' + P_{cu}^t(l,:) \end{bmatrix} \quad (18)$$

Ants from different colonies may attach different kinds of pheromone to a vertex. With the increase of iterations, one class of pheromone from a certain colony will dominate that vertex. Thus for the other population of ants, they are unlikely to choose this vertex.

E. Ant random walks from one colony

Each labeled ant emits pheromone at its visited edges and vertices. The intensity of the pheromone emitted by the i^{th} individual labeled ant $a_i^l \in C_l^t$ located at v_i^l at iteration t decreases with increase in its distance from v_i^l. Thus, the pheromone left on the data points closer to v_i^l is intenser than that left on the data points far from v_i^l. Since we enforce an ant only visit one vertex only once, the shorter the path that travels through all the vertices, the more deposition of pheromone left on the unlabeled vertices. Then the random walk result of labeled ants turns out to be a tree rooting from the nest vertex that links all the vertices.

For instance, the Fig. 4 displays some breast mass cells organized with a binary spanning tree, from which we can easily recognize the intimacy degree between cells by the parent-children relationship. In addition, if the root cell is already labeled as malignant or benign, then its direct children cells are likely to be malignant or benign, but this similarity is less compared to its indirect children cells.

In our nests-and-resources model, a nest l and all resource vertices constitute a fully connected sub-graph G_l, which allows the ants to visit all the resources in at least one step. However, due to the existence of other different kinds of pheromone, an ant will also addresses the intensity of pheromone belonging to its kind when choosing the nest transition target. If an ant detects that a candidate choice shows inferior pheromone intensity produced by its nest to that produced by other nests, it is unlikely to move to this vertex despite a favorable heuristic probability.

As a result, the goal for a nest's tree traversing is finding a maximum spanning tree (MST), or specifically in our paper a spanning tree which maximizes the sum of transition probability and relative pheromone intensity. The reason for choosing the structure of spanning tree is because that it can indicate the intrinsic structure of the input data space. On one hand, spanning tree and its derived tree algorithms serve as a common introductory example of both graph algorithms and greedy algorithms due to their simplicity. On the other hand, there are many spanning tree-based segmentation, recognition and classification applications. They have been proved to outperform the other spanning trees in the setting of weighted graph prediction [22]. Therefore, it is reasonable to perform such ants random walk for classifying medical data sets.

The algorithm starts with a tree consisting of the nest vertex, and continuously increases its size one edge at a time, until it spans all unlabeled vertices. This one nest random walk is sort of like the PRIME algorithm for finding the minimal spanning tree. Detailed description of our ants maximal spanning tree (AMST) algorithm is given in Algorithm 2.

Algorithm 2 Ants Maximal Spanning Tree (AMST)

Input: The nest label l and a connected subgraph $G_l = (V^t, E, Q_l^t)$, where $V = \{v_l\} \cup V_u^t$, Q_l^t is the probabilistic rule at generation t.

Output: V_{new} and E_{new}, which describe a MST; $\Delta\tau$

1. **Initialization:** $V_{new} \leftarrow \{v_l\}$, $E_{new} \leftarrow \emptyset$ and $\Delta\tau(:,l) = 1$.
2. **repeat**
3. Choose an edge $\langle v_j, v_k \rangle$ from V^t with maximal value in Q_l^t such that $v_j \in V_{new}$ and v_k is not. If there are multiple edges with the same value, we choose one of them randomly.
4. Add v_k to V_{new} and $\langle v_j, v_k \rangle$ to E_{new}; meanwhile, compute the pheromone increment as follows.

$$\Delta\tau^t(k,l) \leftarrow \Delta\tau^t(k,l) \times \eta^t(k,j) \quad (19)$$

5. **until** $V_{new} = V$

For each labeled ant with label l, it starts from the labeled vertex and search for a spanning tree over the unlabeled vertices under the guidance of transition probability. Taking the Fig. 5 as an example, the left figure presents the ant random walk environment–a weighted form of Fig. 3. The ants start with a tree rooting at the labeled vertex, and continuously increases its size one edge at a time, until it spans all unlabeled vertices. The right of Fig. 5 shows the random walk result of ants starting at nest vertex v_1.

F. Confidence-based Ants Random Walk Learning

The detailed description of our CARW algorithm is given in Algorithm 3. At each generation, guided by both η and τ, every colony will continuously send ants for foraging mission: occupying the already visited vertices by pheromone

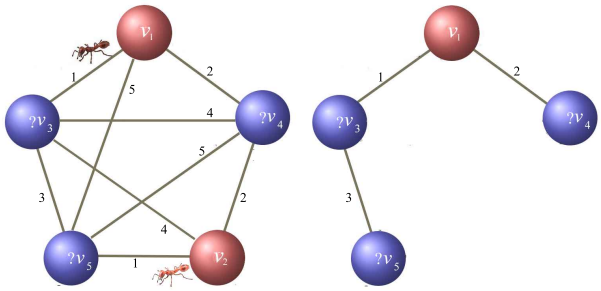


Fig. 5. An example of ants random walk

densities and exploring new ones. After a MST is found, the pheromone increment can be immediately obtained for updating corresponding column in τ.

Next, step 12-19 illustrate the selection of 'high confidence' unlabeled vertices. Similar to CRW algorithm in Fig. 1, the highly reliable unlabeled vertex will be added into its corresponding colony as new member for next generation of label prediction.

G. Stopping criterion

The stopping criterion is based on the computation of ant colony centers. If the colony centers do not change in two successive iterations, then we can say that there is no (re)partition. At that time, the colony formation on the unlabeled vertices is stabilized. It means that either the unlabeled vertices have joined any colony with sufficient reliability, or they have not joined any colony (with sufficient confidence). The unlabeled vertices, which have joined in any colony are now considered as training samples, and thus, the size of the training set is increased.

Algorithm 3 Confidence-based Ant Random Walks (CARW)

Input: $X, Y_m, \gamma, \Delta t$ and confidence value θ
Output: Y_u
Input:

1. Initialize parameters $t = 0$
2. Construct initial graph $G^0 = (V^0, E, Q^0)$
3. **while** not StoppingCriteria **do**
4. **for** each nest vertex v_l **do**
5. Build the ants walking environment $G_l^t = (V^t, E, Q_l^t)$
6. Generate a MST using AMST method
7. Traversing this tree and compute the aggregation pheromone increment $\Delta\tau_{*l}^t$
8. Update the pheromone density τ_{*l}^{t+1} using $\Delta\tau_{*l}^t$
9. **end for**
10. Normalize the matrix τ^{t+1} according to Eq. 11
11. Let $V_u^{t+1} = V_u^t$ and $C^{t+1} = C^t$
12. **for** each unlabeled vertex $v_i \in V_u^t$ **do**
13. Search row τ_{i*}^{t+1} for the largest value τ_{ij}^{t+1} and second largest value τ_{ik}^{t+1}.
14. **if** $\tau_{ik}^{t+1}/\tau_{ij}^{t+1} \leq \theta$ **then**
15. Assign label j to v_i
16. $C_j^{t+1} = C_j^{t+1} \cup v_i$
17. $V_u^{t+1} = V_u^{t+1}/v_i$
18. Remove the i^{th} row of τ^{t+1}
19. **end if**
20. **end for**
21. $t = t + 1$
22. Update the graph $G^t = (V^t, E, Q^t)$
23. **end while**
24. Assign labels to Y_u according to τ^t and C^t

V. Experiments

To evaluate the performance of our proposed models on real-world data sets, we adopt five typical diagnostic data sets, including SPECT, Hepatitis, Hypothyroid, Heart and Breast Cancer. Most of them are binary-class classification problems except the Heart data set.

The missing values of original data sets are filled in the following way: continuous features are set to the average value and those nominal features are set to the majority value. To produce the training data set, a labeled rate is given, then samples are randomly taken out to form the initial training set, and the rest samples are considered as the test set. In this paper, we adopt five labeled rate from 1% to 5%. The test results produced by our CRW and CARW methods are shown in Table I and Table II, while the test results produced by Label Propagation algorithm is given in Table III. In these tables, each row reports the average accuracies and standard deviations for 10 simulation runs given different labeled ratios.

Generally, with the increase of labeled rate, both methods gradually improve the classification accuracy on all the data sets, but not always the case as we can see from both tables. Then our methods share on a fifty-fifty basis on SPECT and Hepatitis data sets, but CARW outperforms CRW on Hypothyroid and Breast Cancer. Further comparisons to our algorithms on these data sets can be obtained from Table III. We adopt the analytical solution of label propagation (LP) as bench marks.

On one hand, CRW performs not worse than the LP algorithm, that is to say, our CRW method is comparable in classification performance but more scalable in large data set. On the other hand, CARW method outperforms CRW and LP algorithms in terms of classification accuracy. Therefore, the combination of ant pheromone and random walk mechanism is successful and effective.

VI. Conclusion

In this paper, we aim to build disease diagnose models over the clinical cases using graph-based methods and swarm intelligence. With a partially labeled graph representation of the raw input medical data under the classification background, this paper first proposes a confidence-based random walks algorithm using the self-training trick over both labeled and unlabeled data. Later, the swarm intelligence algorithm is incorporated to improve the former proposed method by

TABLE I

CLASSIFICATION ACCURACY OF CRW ON FIVE DATA SETS

Labeled Ratio	0.01	0.02	0.03	0.04	0.05
SPECT	79.25% ± 0.07%	79.25% ± 0.05%	79.23% ± 0.04%	79.22% ± 0.01%	79.22% ± 0.01%
Hepatitis	79.22% ± 0.04%	79.08% ± 0.03%	78.95% ± 0.01%	79.33% ± 0.02%	79.05% ± 0.01%
Hypothyroid	95.29% ± 1.70%	95.47% ± 0.99%	95.21% ± 0.93%	95.23% ± 0.77%	96.71% ± 0.76%
Heart	53.97% ± 0.05%	53.85% ± 0.07%	54.05% ± 0.02%	53.92% ± 0.01%	53.84% ± 0.01%
Breast Cancer	80.39% ± 3.57%	83.75% ± 3.31%	85.04% ± 3.01%	89.52% ± 3.32%	89.23% ± 2.94%

TABLE II

CLASSIFICATION ACCURACY OF CARW ON FIVE DATA SETS

Labeled Ratio	0.01	0.02	0.03	0.04	0.05
SPECT	79.25% ± 0.07%	79.25% ± 0.05%	79.23% ± 0.04%	79.22% ± 0.01%	79.22% ± 0.01%
Hepatitis	79.22% ± 0.04%	79.08% ± 0.03%	78.95% ± 0.01%	79.33% ± 0.02%	79.05% ± 0.01%
Hypothyroid	95.29% ± 1.70%	95.47% ± 0.99%	95.21% ± 0.93%	95.23% ± 0.77%	96.71% ± 0.76%
Heart	53.97% ± 0.05%	53.85% ± 0.07%	54.05% ± 0.02%	53.92% ± 0.01%	53.84% ± 0.01%
Breast Cancer	80.39% ± 3.57%	83.75% ± 3.31%	85.04% ± 3.01%	89.52% ± 3.32%	89.23% ± 2.94%

TABLE III

CLASSIFICATION ACCURACY OF LP ON FIVE DATA SETS

Labeled Ratio	0.01	0.02	0.03	0.04	0.05
SPECT	79.25% ± 0.03%	79.39% ± 0.01%	67.96% ± 0.03%	65.25% ± 0.07%	65.27% ± 0.07%
Hepatitis	79.22% ± 0.07%	79.08% ± 0.05%	78.95% ± 0.03%	79.33% ± 0.01%	78.89% ± 0.01%
Hypothyroid	95.21% ± 1.69%	95.23% ± 1.01%	95.21% ± 0.93%	95.21% ± 0.84%	95.21% ± 0.78%
Heart	53.97% ± 0.04%	53.85% ± 0.03%	54.05% ± 0.03%	53.92% ± 0.02%	53.61% ± 0.02%
Breast Cancer	62.77% ± 3.77%	62.72% ± 3.44%	62.75% ± 2.46%	62.71% ± 4.32%	62.73% ± 3.03%

using the metaphor of the aggregation pheromone found in natural behavior of real ants.

For our present models, the classification process is tested on five real data sets, and the test results and comparisons on these real data sets show that our proposed algorithms are competing and impressive, thus demonstrating practical value for medical diagnosis application.

REFERENCES

[1] R. O. Duda, P. E. Hart, D. G. Stork. Pattern Classification. *NewYork:Wiley*, 2000.

[2] Ray Solomonoff. An Inductive Inference Machine, *IRE Convention Record, Section on Information Theory,* Part 2, pp.56C62, 1957.

[3] D. J. Hand. Pattern detection and discovery. In *Pattern Detection and Discovery, ser. Lecture Notes in Computer Science*, vol. 2447, pp. 1-12, 2002.

[4] B. Baesens. Developing intelligent systems for credit scoring using machine learning techniques. *Ph. D. dissertation*, K. U. Leuven, Leuven, Belgium, 2003.

[5] Guo-Zheng Li, Shixing Yan, Mingyu You, Sheng Sun, Aihua Ou Intelligent ZHENG Classification of Hypertension Depending on ML-kNN and Feature Fusion *Evidence-based Complementary and Alternative Medicine (eCAM)*, 2012.

[6] X. Zhu and Z. Ghahramani Learning from labeled and unlabeled data with label propagation *Technical Report CMUCALD-02-107*, 2002.

[7] Lovász L. Random Walks on Graphs: A survey. *Combinatorics*, Paul Erdös is Eighty (Vol. 2), Keszthely(Hungary), 2:01-46, 1993.

[8] David Martens, Bart Baesens, Tom Fawcett. Editorial survey: swarm intelligence for data mining. *Mach. Learn.* , (2011)82:1-42, 2011.

[9] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schöelkopf. Learning with local and global consistency. In *Advances in Neural Information Processing System 16*, 2004.

[10] Xiaohua Xu, Ping He, Lin Lu, Zhoujin Pan, Ling Chen. Component Random Walk. In *proc. of Emerging Intelligent Computing Technology and Applications*, Vol. 304, pp. 62-66, 2012.

[11] X. Zhu. Semi-supervised learning literature survey (technical report 1530). Technical report, Computer Science, University of Wisconsin-Madison, 2007.

[12] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, J. You, Semi-supervised classification based on random subspace dimensionality reduction, *Pattern Recognition* 45(3): 1119C1135, 2012.

[13] M. Shiga, H. Mamitsuka, Efficient semi-supervised learning on locally informative multiple graphs, *Pattern Recognition* 45(3): 1035C1049, 2012.

[14] M.H. Rohban, H.R. Rabiee, Supervised neighborhood graph construction for semi-supervised classi?cation, *Pattern Recognition* 45(4): 1363C1372, 2012.

[15] A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, M. Saerens, Semi-supervised classification and betweenness computation on large, sparse, directed graphs, *Pattern Recognition*, 44(6): 1212C1224, 2011.

[16] Xiaohua Xu, Lin Lu, Ping He and Ling Chen Protein localization prediction using random walks on graphs *BMC Bioinformatics* 2013, 14(Suppl 8):S4

[17] M. Dorigo and T. St$^{'}$utzle. Ant Colony Optimization. *Prentice Hall of India Private Limited*, New Delhi, India, 2005.

[18] S. Tsutsui. Ant colony optimization for continuous domains with aggregation pheromones metaphor. In *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, 207-212, United Kingdom, December 2004.

[19] Ghosh, A., Halder, A., Kothari, M., Ghosh, S. Aggregation pheromone density based data clustering. *Information Sciences 178(13)*, 2816C2831, 2008

[20] K. M. Salama, A. M. Abdelbar, F. E. Otero Utilizing multiple pheromones in an ant-based algorithm for continuous-attribute classification rule discovery. *Applied Soft Computing*, 13(2013) 667-675, 2013.

[21] Anindya Halder, Susmita Ghosh, Ashish Ghosh Aggregation pheromone metaphor for semi-supervised classification. *Pattern Recognition 46(2013)*, 2239-2248, 2013.

[22] Nicoló Cesa-bianchi, Claudio Gentile, Fabio Vitale, Giovanni Zappella Random Spanning Trees and the Prediction of Weighted Graphs, *Journal of Machine Learning Research*, 2013