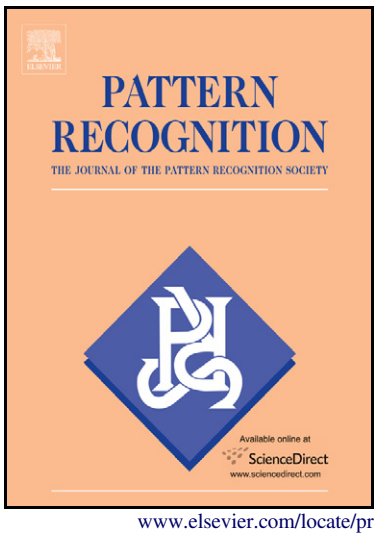


www.elsevier.com/locate/pr

Aggregation pheromone metaphor for semi-supervised classification

Anindya Halder, Susmita Ghosh, Ashish Ghosh

PII: S0031-3203(13)00006-X
DOI: http://dx.doi.org/10.1016/j.patcog.2013.01.002
Reference: PR4690

To appear in: *Pattern Recognition*

Received date: 9 February 2012
Revised date: 16 November 2012
Accepted date: 3 January 2013

Cite this article as: Anindya Halder, Susmita Ghosh, Ashish Ghosh, Aggregation pheromone metaphor for semi-supervised classification, *Pattern Recognition,* http://dx.doi.org/10.1016/j.patcog.2013.01.002

Aggregation pheromone metaphor for semi-supervised classification

Anindya Halder

Department of Computer Application, North-Eastern Hill University, Tura Campus,
Tura 794002, Meghalaya, India
email: anindya.halder@gmail.com

Susmita Ghosh

Department of Computer Science & Engineering, Jadavpur University, Kolkata 700032, India
email: susmitaghoshju@gmail.com

Ashish Ghosh*

Center for Soft Computing Research, Indian Statistical Institute, Kolkata 700108, India

*Corresponding author; email: ash@isical.ac.in; Tel: +913325753110; Fax: +913325788699

Aggregation pheromone metaphor for semi-supervised classification

Anindya Halder[a], Susmita Ghosh[c], Ashish Ghosh[b,*]

[a] *Dept. of Computer Application, North-Eastern Hill University, Tura Campus, Tura 794002, Meghalaya, India*
[b] *Center for Soft Computing Research, Indian Statistical Institute, Kolkata 700108, India*
[c] *Dept. of Computer Science & Engineering, Jadavpur University, Kolkata 700032, India*

Abstract

This article presents a novel 'self-training' based semi-supervised classification algorithm using the property of aggregation pheromone found in real ants. The proposed method has no assumption regarding the data distribution and is free from parameters to be set by the user. It can also capture arbitrary shapes of the classes. The proposed algorithm is evaluated with a number of synthetic as well as real life benchmark data sets in terms of accuracy, macro and micro averaged F_1 measures. Results are compared with two supervised and three semi-supervised classification techniques and are statistically validated using paired t-test. Experimental results show the potentiality of the proposed algorithm.

Keywords: Semi-supervised classification, Self-training, Ant colony, Aggregation pheromone.

1. Introduction

Traditional machine learning methods for pattern classification require sufficient number of labeled data to assign an unlabeled pattern to a cer-

*Corrs. author; email: ash@isical.ac.in; Tel: +913325753110 Fax: +913325788699

tain class. However, labeled patterns are often difficult, costly, and/or time consuming to obtain, as they require the effort of experienced human annotators. On the other hand, unlabeled data may be relatively easy to gather. Semi-supervised learning (classification) [7] methods make use of the large amount of available unlabeled data, along with a small amount of labeled data, to improve classification accuracy. As semi-supervised classification requires less human intervention and produces better accuracy, it is of great interest to the machine learning researchers in recent years.

A variety of semi-supervised learning methods exist in the literature. These can be broadly categorized as follows: self-training [38, 45], co-training [5], transductive support vector machines (TSVM) [2, 8, 9, 44], graph-based methods [3, 4, 47], expectation maximization (EM) with generative mixture models [31] etc. A good review of semi-supervised classification methods is available in [48, 49].

Self-training is a wrapper based method commonly used for semi-supervised learning. In this process a classifier is first trained using a small amount of labeled data. Then unlabeled data patterns are classified using the trained classifier. The classified (unlabeled) pattern whose predicted value is sufficiently high for belonging to a certain class is added to the training set along with its predicted class label. This is done for all the classified (unlabeled) patterns. Thus, the amount of training data increases due to the inclusion of the "high confidence" unlabeled patterns in the original training set. Retraining of the classifier is done using the new enlarged training set; and the procedure is repeated. One can imagine that a misclassification can reinforce itself. Therefore, some algorithms are used to avoid this problem by unlearn-

ing the unlabeled points if the prediction confidence drops below a threshold. Self-training has been applied to several natural language processing tasks [36, 45]. Self-training was also applied to object detection from images [38], and showed that semi-supervised techniques perform better compared to the state-of-the-art object detectors. Self-training based semi-supervised method was used for motion estimation in dynamic systems [22]. Also self-training semi-supervised support vector machine (SVM) was proposed for electroencephalogram (EEG) based brain computer interface system [24].

In co-training [5, 29], features are split into two sets. Following two assumptions are considered in co-training: (i) each sub-set (of features) is sufficient to train a good classifier, and (ii) given a class, the two sets are conditionally independent. Initially, two separate classifiers are individually trained with the labeled data, on the two sub-sets. Co-training then utilizes the unlabeled data by adding the most confident predictions of one classifier to the training set of the other classifier; thereby, effectively allowing each individual classifier to train its counterpart.

Transductive support vector machine (TSVM) [44] is an extension of standard support vector machine for dealing with unlabeled data. In standard SVM, only the labeled data is used, and the goal is to find a maximum margin linear boundary in the Kernel Hilbert Space [10]. In TSVM the unlabeled data is also used to find a labeling of the unlabeled data, so that a linear boundary has the maximum margin on both the original labeled data and the unlabeled data (with predicted label). The decision boundary has the smallest generalization error bound on unlabeled data [43]. Intuitively, unlabeled data guides the linear boundary away from denser regions.

Initially developed TSVM algorithms [2] were not able to handle a large number of unlabeled data efficiently. To overcome this drawback, many variants like SVMLight-TSVM [21], low density separation (LDS) [8], concave-convex procedure (CCCP) for TSVM (CCCP-TSVM) [9] have been proposed in the literature. A semi-supervised support vector classifier designed using a quasi-Newton method for nonsmooth convex functions is proposed in [35].

Graph-based semi-supervised methods construct a graph where the nodes designate the labeled and unlabeled samples of the data set and (weighted or unweighted) edges represent the similarity of samples. These methods are nonparametric, discriminative, and transductive in nature. Some of the graph based methods like mincut [3, 4], harmonic [15], local and global consistency [47], mainfold regularization [1] are discussed in [48, 49]. A few recent graph-based semi-supervised classifiers are proposed particularly for graph construction [46], handling multiple graphs in gene networks [41], neighborhood graph construction [37], and betweenness computation on large sparse directed graphs [27].

Generative models [7] are possibly the oldest semi-supervised learning method. It assumes a probabilistic model where identifiable mixture distribution is known. With a large amount of unlabeled data, the mixture components can be identified. The model using EM method is being used for text classification [31].

In this article, a novel 'self-training' based semi-supervised algorithm is proposed using the aggregation pheromone density which is inspired by the natural behavior of real ants and other social insects.

Different applications originated from the study of different types of

swarms. Among them, most popular ones are ant colony and bird flocks [12]. *Ant Colony Optimization* (ACO) [11] and *Aggregation Pheromone Systems* (APS) [42] are computational algorithms modeled on the behavior of ant colony. ACO [11] 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. Pheromone, which is responsible for clumping or clustering behavior in a species and brings individuals into closer proximity, is termed as aggregation pheromone [42]. Thus, aggregation pheromone causes individuals to aggregate around good positions which in turn produces more pheromone to attract individuals of the same species. In APS [42], a variant of ACO, this behavior of ants is used to solve real parameter optimization problems.

Inspired by the aggregation pheromone system found in ants and other similar agents, in earlier works, attempts were made for solving clustering [13], classification [16], image segmentation [14] problems, and land use map generation from mutispectral remotely sensed images [17] with encouraging results.

Though a large number of techniques exists for ant based unsupervised classification (i.e., clustering) in the literature [19], a few attempts [25, 26, 28, 34] have been made for (supervised) classification. In our earlier (conference) work one preliminary attempt [18] was also made for semi-supervised classification based on ant colony approach with promising results.

Motivated from the promising results, the earlier research [18] has been extended in this article to propose an advanced aggregation pheromone den-

sity based semi-supervised classification (called, APSSC) algorithm.

The proposed APSSC algorithm is 'self-training' in nature and consists of two phases namely, *'self-training'* and *'testing'*. In *self-training* phase the classifier is first trained with the small amount of labeled (patterns) ants. Afterwards, the classifier is used to classify the unlabeled ants; and then among the unlabeled ants, the 'high confidence' ones are determined and 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 *testing* phase, each test (pattern) ant is evaluated to assign to the colony for which the average aggregation pheromone density is more. In this way the classification accuracy obtained with a small amount of labeled ant can be improved by the use of the 'high confidence' unlabeled ants.

The proposed method has the advantage of not having any assumption regarding the data distribution. Moreover, opposed to the earlier version [18], it does not require to set any free parameter manually. In addition, it can better capture the arbitrary shapes of the classes by updating the covariance matrices of the classes with iterations.

The proposed semi-supervised classifier is compared with two conventional supervised classifiers (viz., multi layer perceptron and support vector machine) and three state-of-the-art semi-supervised classifiers (viz., semi-supervised classification by low density separation [8] and concave-convex procedure for transductive support vector machine[9], self-training semi-super-

vised support vector machine [24]) using five real life benchmark data sets and four artificially generated data sets. Performance of each of the methods is evaluated using percentage of overall accuracy, macro averaged F_1 measure, and micro averaged F_1 measure. Results of the investigations of the semi-supervised methods are also statistically validity using paired t-test [23]. Experimental results show the potentiality of the proposed semi-supervised method compared to other techniques for most of the data sets.

The rest of the article is organized as follows. Section 2 provides a detail description of the proposed ant based semi-supervised classification method using aggregation pheromone system. Details of the experiments and analysis of results are provided in Section 3, and finally, conclusions are drawn in Section 4.

2. Proposed Methodology

As mentioned earlier, aggregation pheromone brings individuals into closer proximity. This group forming nature of aggregation pheromone (found in natural behavior of real ants) is being used as the basic idea of the proposed technique.

The proposed aggregation pheromone density based semi-supervised classification (APSSC) algorithm is 'self-training' in nature. It consists of two steps. The first step uses *'self-training'* strategy, where the semi-supervised classifier is (re)trained iteratively using the small number of labeled ants along with the 'high confidence' unlabeled ants (described latter). The second step is *'testing'*. Once *self-training* is over (i.e., colony is stabilized) the new test ants are predicted to assign a particular class (colony) in the *testing*

phase. The details of the proposed methods are described below.

2.1. Aggregation Pheromone Density based Semi-Supervised Classification

Consider a data set with K classes and a small number of labeled data patterns from each class which, by our assumption, forms K homogeneous groups or colonies of ants in the *training/lebeled set* L. Also, there is (a relatively large) $|U|$ number of unlabeled data patterns in the *unlabeled set* U.

Let, $\boldsymbol{x}_1^{l_k}$, $\boldsymbol{x}_2^{l_k}$, ..., $\boldsymbol{x}_{|C_k^0|}^{l_k}$ be the given original *training data* or *labeled data* patterns in the k^{th} *initial training class* C_k^0. These patterns are considered as a population of $|C_k^0|$ number of ants represented as $a_1^{l_k}$, $a_2^{l_k}$, ..., $a_{|C_k^0|}^{l_k}$. Hence, an ant $a_i^{l_k}$ represents the i^{th} training data pattern (in the k^{th} initial training class) $\boldsymbol{x}_i^{l_k} \in C_k^0$.

Consider $\boldsymbol{x}_1^u$, $\boldsymbol{x}_2^u$, ..., $\boldsymbol{x}_{|U|}^u$ to be the unlabeled data patterns represented as *unlabeled ants* a_1^u, a_2^u, ..., $a_{|U|}^u$, correspondingly.

Step1: self-training

At iteration $t = 0$, only labeled ants (patterns) are considered to form the *initial training colony*, i.e., the k^{th} *training colony* C_k^t is the same as k^{th} *initial training class/colony* C_k^0.

Each labeled ant emits pheromone at its neighborhood. The intensity of pheromone emitted by the i^{th} individual labeled ant $a_i^{l_k} \in C_k^t$ located at $\boldsymbol{x}_i^{l_k}$ at iteration t decreases with increase in its distance from $\boldsymbol{x}_i^{l_k}$. Thus, the pheromone intensity at a point closer to $\boldsymbol{x}_i^{l_k}$ is more than those at other points that are farther from it. To achieve this, the pheromone intensity

emitted by ant $a_i^{l_k} \in C_k^t$ at time t is modeled by a Gaussian distribution. Hence, effect of the emitted pheromone density on the j^{th} unlabeled ant a_j^u (located at $\boldsymbol{x}_j^u$) at iteration t due to the i^{th} labeled ant of k^{th} colony ($a_i^{l_k} \in C_k^t$) located at $\boldsymbol{x}_i^{l_k}$ is given by:

$$\Delta\tau^t(\boldsymbol{x}_i^{l_k}, \boldsymbol{x}_j^u) = \frac{1}{(2\pi)^{d/2}\,(det(\Sigma_k^t))^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_j^u - \boldsymbol{x}_i^{l_k})^T(\Sigma_k^t)^{-1}(\boldsymbol{x}_j^u - \boldsymbol{x}_i^{l_k})\right), \tag{1}$$

where, Σ_k^t, $\det(\Sigma_k^t)$, and d represent respectively, the covariance matrix of the k^{th} class at iteration t, the determinant of the covariance matrix Σ_k^t, and number of dimensions of the data set used.

The average effect of emitted (aggregated) pheromone on the j^{th} unlabeled ant a_j^u due to k^{th} *training colony* C_k^t at iteration t is given by:

$$\Delta\overline{\tau}_{jk}^t = \frac{1}{|C_k^t|} \sum_{\boldsymbol{x}_i^{l_k} \in C_k^t} \Delta\tau^t(\boldsymbol{x}_i^{l_k}, \boldsymbol{x}_j^u); \quad \forall\ j, \ \ \forall\ k. \tag{2}$$

Thereafter, pheromone density τ_{jk}^t due to the k^{th} colony C_k^t on the j^{th} unlabeled (pattern) ant at iteration t is updated according to the following equation:

$$\tau_{jk}^t = (1-\rho)\tau_{jk}^{t-1} + \rho\Delta\overline{\tau}_{jk}^t; \quad \forall\ j, \ \ \forall\ k, \tag{3}$$

where, $0 \leq \rho \leq 1$ is the evaporation constant. With smaller values of ρ, the system uses information of the pheromone density of the past cycles more than with the larger values of ρ. Larger value of ρ indicates that the effect of the pheromone emitted in the present iteration is more compared to the pheromone emitted in the previous iterations. ρ acts as a trade-of factor of the emitted pheromone in the previous and the present iterations. Instead

of keeping it constant [18] throughout during the *self-training* process, it is reasonable to vary it with respect to time. As in the *self-training* process, the emitted pheromone (in the current iteration) at the location of an unlabeled ant is computed mainly due to the training ants; also, there is no (or, less) effect of the emitted pheromone from the earlier cycle, therefore, the effect of emitted pheromone ($\Delta\overline{\tau}_{jk}^{t}$) in the current iteration should be high during the initial stage. As time progress the effect of the emitted pheromone in the current iteration should decrease and the effect of the pheromone density of the past cycles (τ_{jk}^{t-1}) should increase. Hence, ρ is a function of time and it is defined as:

$$\rho = \frac{1}{1 + log(t+1)}. \tag{4}$$

After pheromone density is updated, the gradation of belonging of an unlabeled ant a_j^u to colony C_k^t is computed as:

$$\mu_{jk}^{t} = \frac{\tau_{jk}^{t}}{\sum_{k=1}^{K} \tau_{jk}^{t}} \quad \forall\ j, \quad \forall\ k. \tag{5}$$

This μ_{jk}^{t} is nothing but the normalized pheromone density (n.p.d.) at the location of an unlabeled ant a_j^u due to colony C_k^t.

Once the normalized pheromone density (n.p.d.) values of all the unlabeled ants are determined, ants are *evaluated* to be temporarily added to the training set for the next iteration $(t+1)$. Ants, added to the training set, are termed as '*high confidence ants*'. Evaluation of the unlabeled ants is done as follows.

2.1.1. Determination of 'high confidence' ants:

As stated earlier, let μ_{jk}^t be the normalized pheromone density (n.p.d.) value associated with an unlabeled ant a_j^u due to the colony C_k^t. Let the highest normalized pheromone density corresponding to the unlabeled ant a_j^u be $\mu_{jh}^t (= \max_k(\mu_{jk}^t))$. The ratio $\frac{\mu_{jk}^t}{\mu_{jh}^t}$ $(\forall k \neq h)$ represents the degree of similarity of an unlabeled ant a_j^u for belonging to colony C_k and the highest contributing colony C_h. Range of this ratio is in [0,1]. More the value of the ratio, more is the similarity of the unlabeled ant with two colonies C_k and C_h; hence, less is the confidence (of the unlabeled ant for belonging to any colony). Therefore, if all such ratios between $\mu_{jk}^t (\forall k \neq h)$ and the highest n.p.d. value μ_{jh}^t of the unlabeled ant a_j^u are less than equal to $\frac{1}{K}$, (where, K is the number of colony or class) then that unlabeled ant a_j^u becomes a *'high confidence'* ant to be added to the training set for the next iteration $(t+1)$. If for any colony C_k the ratio $\frac{\mu_{jk}^t}{\mu_{jh}^t}$ $(\forall k \neq h)$ is greater than $\frac{1}{K}$, then the corresponding ant is considered to be a less confidence one, and is not added to any colony.

The methodology for determining the 'confidence' of an unlabeled ant a_j^u is summarized in Algorithm 1.

Note that, addition of an ant to the colony C_h is done temporarily for the next iteration. In subsequent iterations it will be added to the appropriate colony depending on its current membership value or it may not be included in any colony. Hence, in each iteration (re)assignment of the initial unlabeled ants occurs.

Algorithm 1 : Determination of the 'confidence' of an unlabeled ant a_j^u

for each n.p.d. μ_{jk}^t $(k \neq h)$ due to k^{th} *training colony* C_k^t at iteration t **do**
  if $(\frac{\mu_{jk}^t}{\mu_{jh}^t = \max\limits_k(\mu_{jk}^t)} \leq \frac{1}{K})$ **then**
    $flag\_variable = 1$
  else
    $flag\_variable = 0$
    break;
  end if
end for
if $(flag\_variable == 1)$ **then**
  Add the unlabeled ant a_j^u to the appropriate training colony (C_h^{t+1}) for the next iteration $(t+1)$ as
  $C_h^{t+1} = C_h^0 \cup \boldsymbol{x}_j^u$
else
  Do not add the unlabeled ant a_j^u to any colony.
end if

2.1.2. Stopping criterion of self-training:

The *self-training* phase of the algorithm stops when there is no (re)assignment. This is done by computing the colony centers. If the colony centers in two successive iterations do not change then it can be said that there is no (re)partition. At that time colony formation by the unlabeled ants is over and the unlabeled ants are stabilized. It means either they have joined any colony with sufficient confidence, or (rest) have not joined any colony (with sufficient confidence). The unlabeled ants, which have joined in any colony are now considered as training samples, and thus, the size of the training set is increased with the help of the unlabeled patterns.

Step2: testing

After the colony formation (by the unlabeled ants) is over, the new ants (patterns) are tested as follows. If the test ant a_n at $\boldsymbol{x_n}$ appears in the system, the average aggregation pheromone density (at the location of the new ant a_n) by the colony C_k^t is given by [as in Equation 2]:

$$\Delta\overline{\tau}_{nk} = \frac{1}{|C_k^t|} \sum_{x_i \in C_k^t} \frac{1}{(2\pi)^{d/2} \left(det(\Sigma_k^t)\right)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{x}_i)^T (\Sigma_k^t)^{-1} (\boldsymbol{x}_n - \boldsymbol{x}_i)\right). \tag{6}$$

The test ant a_n will move towards a colony for which the average aggregation pheromone density (at the location of that test ant) is higher than that of the other colonies. Hence, finally the said ant will join the colony that will be governed by the following equation:

$$ColonyLabel(\boldsymbol{x_n}) = \arg\max_k(\Delta\overline{\tau}_{nk}). \quad (7)$$

As opposed to the previous version [18], the present version of the algorithm does not have any free parameters to be set by the user manually; also in each iteration the algorithm updates the class covariance matrix (of Equation 1) and thereby is able to better capture the shape of the classes. The complete procedure is summarized in Algorithm 2.

3. Experimental evaluation

In this section we report the details of experimental setup, along with the data sets and then analyze the results.

3.1. Data sets used

For the purpose of our study, we used four artificially generated data sets (shown in Figure 1), and five real life data sets, four among them are from the UCI repository [30], and Telugu Vowel data is from [32].

Among the synthetic data sets Annular data set (Figure 1 (a)) has four concentric rings of different radius representing 4 classes having total 1400 data patterns. Ellipse data (Figure 1 (b)) consists of two very close ellipse, partially confound within a half ellipse representing the 3 classes having 300 data patterns. Pat2 data (Figure 1 (c)) [33] consists of 3 classes having 880 patterns. Spiral data (Figure 1 (d)) contains 1000 data points distributed in two spirals shaped classes. All the synthetic data sets have two dimensions. Please note that, all the synthetic data sets are non linearly separable.

Among the real life data sets, the Ionosphere is a radar data which consists of 351 instances each with 34 continuous features distributed in 2 classes

Algorithm 2 : Aggregation Pheromone density based Semi-Supervised Classification (APSSC)

begin_self_training()

Initialize: Iteration counter $t \leftarrow 0$; initial pheromone density $\tau_{jk}^{-1} \leftarrow 0, \forall\, j, \forall\, k$.

repeat

  for each unlabeled ant a_j^u located at $\boldsymbol{x_j^u}$ **do**

    for each *training colony* C_k^t at iteration t **do**

      Calculate the average aggregation pheromone density $\Delta\overline{\tau}_{jk}^t$ on the j^{th} unlabeled ant a_j^u due to all ants in *present training colony* C_k^t at iteration t using equation (2).

      Update pheromone density τ_{jk}^t due to k^{th} colony C_k^t on the j^{th} unlabeled (pattern) ant at iteration t by Equation (3).

    end for

    for each *training colony* C_k^t at iteration t **do**

      Compute the n.p.d. μ_{jk}^t of each unlabeled ant a_j^u due to each colony C_k^t at iteration t using Equation (5).

    end for

    Compute the 'confidence' of the unlabeled ant a_j^u and add or do not add the ant to the appropriate colony for the next iteration $(t+1)$ according to Algorithm 1.

  end for

  $t \leftarrow t+1$.

until $<$ *StoppingCriteria* $>$

end_self_training (P.T.O.)

Algorithm 2: APSSC (continued)

***begin_testing*()**

for each new test ant a_n located at $\boldsymbol{x_n}$ **do**

for each colony C_k^t **do**

Calculate the average aggregation pheromone density $\Delta\overline{\tau}_{nk}$ at location $\boldsymbol{x_n}$ due to all ants in colony C_k^t using Equation (6).

end for

Compute the $ColonyLabel(\boldsymbol{x_n})$ of the ant a_n by Equation (7). // Ties are broken arbitrarily.

end for

end_testing

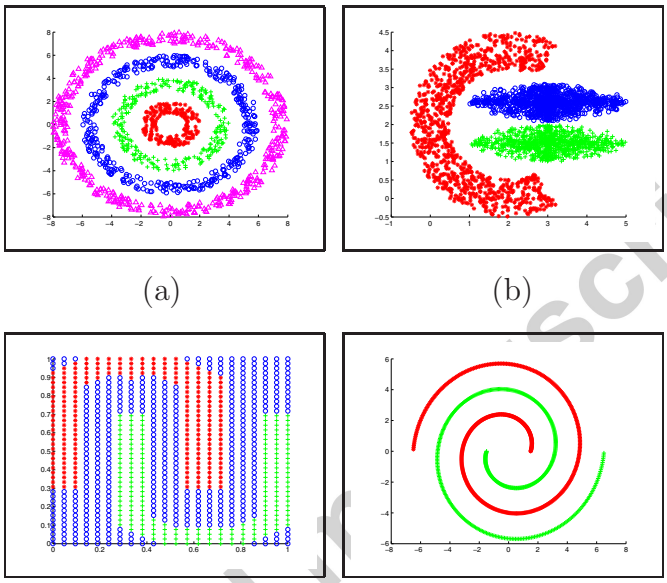
namely "good" and "bad". This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power of the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structures in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. The Indian Telugu vowel data [32] is the formant frequency of sounds in consonant-vowel-consonant context uttered by three speakers in the age group 30-35 years. The data set consists of 871 instances with 3 formant frequencies (features) which were obtained through the spectrum analysis of the speech data. The data patterns are distributed in 6 overlapping classes and their boundaries are ill-defined. Balance scale data was generated to model the psychological experimental results. It has 625 instances described by 4 features, distributed in 3 classes. Sonar data has 208 instances described

by 60 attributes distributed in 2 classes. Wisconsin Breast Cancer (WBC) data contains 699 instances distributed in 2 categories described by 9 features of which 16 instances with the missing values are ignored.

To test the classification accuracy, 5% of data is taken out randomly from a data set to form the initial training set and the rest is considered as the unlabeled set. The process is repeated 10 times. The reported results are obtained considering the unlabeled data as the test set. A summary about the data sets is given in Table 1.

Table 1: Summary of the data sets used for the experiments

Data set	**Classes**	**Dimensions**	**Pattern**	**Labeled pattern**
Synthetic data				
Annular	4	2	1400	5%
Ellipse	3	2	3000	5%
Pat2	3	2	880	5%
Spiral	2	2	1000	5%
Real Life data				
Ionosphere	2	34	351	5%
Telugu vowel	6	3	871	5%
Balance Scale	3	4	625	5%
Sonar	2	60	208	5%
WBC	2	9	683	5%



(a) (b)

(c) (d)

Figure 1: (a) Annular data, (b) Ellipse data, (c) Pat2 data, and (d) Spiral data

3.2. Methods compared with

The proposed method is compared with two traditional classifiers: multi layer perceptron (MLP) [20], and support vector machine (SVM) [40], along with three semi-supervised techniques, namely, semi-supervised classification by low density separation (LDS) [8], concave-convex procedure for transductive support vector machine (CCCP-TSVM) [9] and self-training semi-supervised support vector machine (SS-SVM) [24]. For implementation of SVM and SS-SVM we have used the package as referred in [6] package. We have used the available source codes of LDS, and CCCP-TSVM, respectively, from [8], and [9] for the implementation. Note that, compared methods have a number of parameters. The MLP and APSSC algorithms are implemented in Matlab. We have suitably adjusted the parameters to get the optimum results.

3.3. Performance evaluation measures

In order to evaluate the performance of the proposed classifier, in this article we have used following three kinds of performance measures. Namely, (i) percentage accuracy, (ii) macro averaged F_1 measure, and (iii) micro averaged F_1 measure.

Percentage accuracy: Here we have reported the results on test case accuracy only, that is percentage of correctly classified test patterns out of the total test patterns.

Macro averaged F_1 measure: Macro averaged F_1 is derived from precision and recall [39]. The precision (p_i) of class i is defined as

$$p_i = \frac{\#\ patterns\ correctly\ classified\ into\ class\ i}{\#\ patterns\ classified\ into\ class\ i}, \tag{8}$$

and, recall (r_i) of class i is defined as

$$r_i = \frac{\#\ patterns\ correctly\ classified\ into\ class\ i}{\#\ patterns\ that\ are\ truly\ present\ in\ class\ i}. \tag{9}$$

Then $(F_1)_i$, the harmonic mean between precision and recall, of class i is defined as

$$(F_1)_i = \frac{2 \times p_i \times r_i}{p_i + r_i}. \tag{10}$$

F_1 measure gives equal importance to both precision and recall. The macro averaged F_1 measure is computed by first computing the F_1 scores for each category (class) and then averaging these per-category scores to compute the global means. Macro averaged F_1 gives equal weight to each category.

Macro averaged F_1 measure (denoted as Macro F_1, in short) is defined as:

$$Macro\ \ F_1 = \frac{1}{K}\sum_{i=1}^{K}(F_1)_i, \tag{11}$$

where K is the number of categories (classes).

Micro averaged F_1 measure: It is computed by first creating a global contingency table whose cell values are sum of the corresponding cells in the per-category contingency tables. Then this global contingency table is used to compute the micro averaged performance scores. Micro averaged F_1 gives equal weight on each sample (test case).

Micro averaged F_1 measure (denoted as Micro F_1, in short) is defined as:

$$Micro\ \ F_1 = \frac{2 \times \frac{1}{K}\sum_{i=1}^{K} p_i \times \frac{1}{K}\sum_{i=1}^{K} r_i}{\frac{1}{K}\sum_{i=1}^{K} p_i + \frac{1}{K}\sum_{i=1}^{K} r_i} \tag{12}$$

where K is the number of categories (classes).

Note that, macro-averaged F_1 and micro-averaged F_1 [16] are derived from precision and recall [16], and their values lie between 0 and 1. *Closer the value of macro averaged F_1 and micro averaged F_1 to 1, better is the classification.*

3.4. Statistical significance test

To test the significance of results statistically (in terms of percentage accuracy) of the investigation, paired t-test [23] has been performed with the proposed APSSC versus other semi-supervised methods at 5% level of significance, and results of t-test in terms of p-score are reported in Table 4.

3.5. Experimental results and analysis

The average results and standard deviations (shown in bracket) for 10 simulation runs (with 10 different labeled, unlabeled / test sets) of all the algorithms are reported in Tables 2 & 3 for synthetic and real life data sets, respectively. The CPU (execution) time, in seconds, needed by the algorithms are also given in the table for comparison.

All the algorithms used in this article are implemented in Matlab and simulated in core 2 duo (2.2GHz speed) processor using 2 GB of main memory in Windows environment.

Rank of each algorithm is given depending on its performance index using '#' symbol followed by corresponding rank (from 1 to 3). For example '#1' indicates the best result with respect to the corresponding performance index. The best results are also marked as bold.

It is seen from the experimental outcome of synthetic data (Table 2) that the proposed APSSC algorithm outperforms the other semi-supervised

counterpart (LDS, CCCP-TSVM and SS-SVM), for Annular, Pat2 and Spiral data sets in terms of classification accuracy, macro averaged F_1 (denoted as, Macro F_1 in tables) and micro averaged F_1 (denoted as, Micro F_1 in tables). In particular, the accuracy of the APSSC for the Spiral data is 100%. Also the small standard deviation of the average results produced by APSSC suggest the robustness of the proposed method with variation of the training sets. Though for Ellipse data set LDS performs slightly better than other semi-supervised methods, still the accuracy produced by APSSC is as high as 99.61%. The very high value (> 0.9) of Macro F_1 and Micro F_1 measure produced by APSSC in case of Annular, Ellipse, and Spiral data indicates the very high (classwise) precision, and recall rates. Analyzing the high accuracy produced by the proposed method trained with very limited training samples (only 5% of the total data) on synthetic data sets with arbitrary geometrical classes suggests the effectiveness of the proposed method in capturing the different geometrical shapes having non linearly separable and non convex class distributions. In these cases, the performance of the supervised classifiers (MLP and SVM) trained with limited training samples are very poor.

For real life data sets (Table 3), the proposed APSSC is observed to perform better in terms of classification accuracy, macro averaged F_1 and micro averaged F_1 in three cases namely Telugu Vowel, Balance Scale and Sonar data sets. The improvement found by the proposed method over other methods is significantly high. For example, the improvement of the proposed algorithm in Telugu Vowel, Balance Scale and Sonar data compared to the second best method are 3.07%, 2.78%, and 4.39%, respectively. Whereas,

in case of the WBC data set the accuracy produced by the APSSC is only 0.24% less as compared to the best one (CCCP-TSVM).

It is worth mentioning here that, as expected, all the semi-supervised classifiers clearly dominate the supervised classifiers (MLP and SVM). This is because the use of the unlabeled patterns really helps to gain accuracy in semi-supervised case even though the number of training samples used in both the cases (supervised and semi-supervised) are the same.

As mentioned earlier, results of the investigation (in terms of percentage accuracy) are statistically validated using the paired t-test [23] performed with the proposed APSSC versus other semi-supervised methods at 5% level of significance. Results of paired t-test in terms of p-score are reported in Table 4. Statistically significant results in terms of p-score of the paired t-test (at 5% level of significance) are marked as bold in Table 4. The up-arrow($\uparrow$), and down-arrow ($\downarrow$) are also shown in the table along with the p-scores to indicate, respectively, the significant improvement found by the proposed method APSSC (compared to the other semi-supervised method), and significant improvement found by the other method (compered to the proposed APSSC) during the paired t-test.

From the paired t-test it is found that in case of Annular and Ellipse data sets improvement in performances of the proposed APSSC method is statistically significant (at 5% level) compared to SS-SVM method. However, performance of the LDS method for Ellipse data set is found to be significantly better than that of the APSSC method. Whereas, for Pat2 and Spiral data sets, the proposed method significantly outperformed two other semi-supervised methods, namely, SS-SVM and LDS. Statistical analysis of the

results obtained in real life data sets reveals that the proposed APSSC significantly dominates all the three semi-supervised methods in cases of Telugu Vowel and Sonar data sets. Also for the Balance Scale data, the performance of the proposed APSSC is significantly better compared to those of the LDS and SS-SVM methods. However, CCCP-TSVM produces significant better performances compared to the proposed one in Ionosphere, and WBC data sets. Though, for WBC data, the proposed method showed significant statistical improvement in performance compared to LDS method. In summary, from Table 4 it is found that out of a total of 27 (statistical) tests, in 15 cases (shown in $\uparrow$) the proposed method showed statistically significant improvement in performances compared to the other methods. Whereas, for 8 cases (shown in normal font without arrow), there is no statistical difference of the performances of the proposed one to those of the other ones, and only for 4 cases (shown in $\downarrow$) the other semi-supervised methods performed statistically better than the proposed method.

Note that, the performance of the semi-supervised methods (particularly, LDS and CCCP-TSVM) is sensitive to a number of (manual) parameter settings, and the parameter tuning also varies with data sets. The proposed APSSC on the other hand does not have any free parameter to be set by the users manually. Hence, the proposed method has significant advantage compared to the other semi-supervised methods.

Execution time is the least for CCCP-TSVM for most of the data sets. However, the execution time of the proposed algorithm is moderate.

Table 2: Experimental results for synthetic data

Data	Method	% Accuracy	Macro F_1	Micro F_1	Time (in second)
Annular	MLP	50.75 (5.91744676)	0.444296 (0.135712944)	0.476262 (0.133639028)	15.92 (0.074836414)
	SVM	57.13533 (6.113701984)	0.52047 (0.038274)	0.557013 (0.04104)	3.7 (0.05172)
	LDS	92.8449487 (3.249123293)	0.921006 (0.020370)	0.93052 (0.028701)	18.49 (0.372124248)
	CCCP-TSVM	94.057 #2 (3.805113)	0.940297 #2 (0.034229)	0.944773 #2 (0.0353177)	3.97 (0.18274)
	SS-SVM	92.976 #3 (1.595021421)	0.92507 #3 (0.012046)	0.93516 #3 (0.013301)	57.41 (1.307)
	APSSC	**94.93233** #1 (0.811250596)	**0.948773** #1 (0.0083383)	**0.9517508** #1 (0.00805861)	13.34 (0.244400491)
Ellipse	MLP	85.61713 (3.4172)	0.856166 (0.0025669)	0.866189 (0.0028241)	26.06 (3.703)
	SVM	86.86 (3.4172)	0.860428 (0.0023669)	0.869185 (0.0027075)	10.53 (1.15733)
	LDS	**99.9897126** #1 (0.2401525)	**0.9988** #1 (0.001747)	**0.9994261** #1 (0.002139)	139.68 (8.67633282)
	CCCP-TSVM	99.830522 #2 (0.534772203)	0.998066 #2 (0.005335697)	0.998524 #2 (0.00525842)	9.82 (0.7391)
	SS-SVM	99.227713 (0.474888166)	0.992187 (0.001937)	0.992244 (0.0020625)	92.05 (3.114)
	APSSC	99.61713 #3 (0.2865596)	0.996166 #3 (0.00286905)	0.996189 #3 (0.002839528)	35.29 (2.011096484)
Pat2	MLP	54.88037 (5.375179833)	0.449872 (0.09194651)	0.478673 (0.099698202)	9.02 (0.02964681)
	SVM	52.39 (4.01755)	0.427153 (0.072265)	0.449557 (0.070521)	2.73 (0.1704)
	LDS	67.7976998 (0.812484618)	0.65829 (0.09194651)	0.664493 (0.099698202)	5.25 (0.224279856)
	CCCP-TSVM	69.92207 #2 (3.07411)	0.68853 #2 (0.03285)	0.689472 #2 (0.03432)	2.06 (0.1402)
	SS-SVM	68.497 #3 (2.057976622)	0.667205 #3 (0.02663)	0.675106 #3 (0.02908)	7.35 (0.573)
	APSSC	**71.92585** #1 (0.0905977)	**0.707392** #1 (0.00776266)	**0.711016** #1 (0.006875903)	4.94 (0.19262448)
Spiral	MLP	56.29472 (4.37150531)	0.544723 (0.068508581)	0.564783 (0.046819013)	9.59 (0.027942202)
	SVM	59.537 (3.0627501)	0.574723 (0.048508581)	0.584783 (0.0510839)	2.79 (0.500963617)
	LDS	99.3887511 (0.487125227)	0.993897 (0.008184)	0.99805 (0.0093051)	12.09 (0.125439068)
	CCCP-TSVM	99.90275 #2 (0.037845215)	0.998074 #2 (0.0006382)	0.999143 #2 (0.00065066)	2.91 (0.72095)
	SS-SVM	99.518713 #3 (0.429304046)	0.99614 #3 (0.007914)	0.99872 #3 (0.008605)	28.32 (2.017)
	APSSC	**100** #1 (0)	**1** #1 (0)	**1** #1 (0)	19.45 (6.174734773)

Table 3: Experimental results for real life data

Data	Method	% Accuracy	Macro F_1	Micro F_1	Time (in second)
Ionosphere	MLP	79.69972 (7.430090652)	0.771171 (0.100609496)	0.782185 (0.089924271)	84.10 (0.348435169)
	SVM	77.5736 (5.8205)	0.7572257 (0.057922)	0.764291 (0.061530)	10.66 (1.757)
	LDS	91.2455193 (1.804415569)	0.9125523 (0.020557)	1.9730894 (0.031433)	1.97 (0.128268035)
	CCCP-TSVM	**93.524** #1 (3.170664)	**0.935217** #1 (0.031192)	**0.941882** #1 (0.034704)	0.77 (0.015)
	SS-SVM	92.15 #2 (2.377346794)	0.919527 #2 (0.026204)	0.923172 #2 (0.028137)	5.44 (0.702)
	APSSC	91.77 #3 (1.4300906)	0.9164 #3 (0.008009)	0.9171 #3 (0.0076992)	3.12 (0.226394665)
Telugu Vowel	MLP	70.49638 (5.306315094)	0.650171 (0.064746772)	0.67893 (0.055014359)	16.65 (0.031653682)
	SVM	65.71342 (5.306315094)	0.607968 (0.064746772)	0.619053 (0.055014359)	3.255 (0.571)
	LDS	73.7933292 (4.701421198)	0.708732 (0.037619)	0.720191 (0.039685)	6.39 (0.101618292)
	CCCP-TSVM	78.80682 #3 (3.7703)	0.7669573 #3 (0.05177)	0.7746102 #3 (0.058048)	3.02 (0.1486003)
	SS-SVM	80.1 #2 (2.647850281)	0.7847029 #2 (0.028316)	0.788131 #2 (0.035291)	12.33 (1.28)
	APSSC	**83.17** #1 (1.306315094)	**0.8105** #1 (0.01047467)	**0.8183** #1 (0.01045014)	9.39 (0.112917669)
Balance Scale	MLP	81.3131 (3.450873883)	0.69203 (0.054379174)	0.71 (0.053609467)	11.8327 (0.037680086)
	SVM	76.54858 (3.449413067)	0.58904 (0.013060756)	0.59176 (0.01582411)	0.75 (0.134919276)
	LDS	85.5219 (3.033736355)	0.714331 (0.054379174)	0.72354 (0.053609467)	3.06 (0.16965603)
	CCCP-TSVM	86.4926 #2 (4.01775)	0.73583 #2 (0.0440705)	0.748701 #2 (0.0511604)	0.57 (0.042)
	SS-SVM	85.776 #3 (2.817399116)	0.727012 #3 (0.0265512)	0.73033 #3 (0.026827)	5.07 (1.46)
	APSSC	**89.27** #1 (2.450873883)	**0.7914** #1 (0.013018994)	**0.7977** #1 (0.015757597)	2.39 (0.134919276)

Table 3: Continued.

Data	Method	% Accuracy	Macro F_1	Micro F_1	Time (in second)
Sonar	MLP	59.54546 (9.829256854)	0.566943 (0.124155419)	0.578235 (0.127092305)	103.15 (1.275527151)
	SVM	55.5556 (5.218325508)	0.518773 (0.10806338)	0.525311 (0.1099551)	1.27 (0.266801403)
	LDS	62.8509299 (4.0515366)	0.61840 (0.095227)	0.61982 (0.0988044)	0.33 (0.118601667)
	CCCP-TSVM	65.735 #3 (4.707361)	0.64701 #3 (0.0986613)	0.65491 #3 (0.104279)	0.19 (0.05117)
	SS-SVM	67.035 #2 (2.916623809)	0.66057 #2 (0.072153)	0.66388 #2 (0.072807)	2.55 (0.437)
	APSSC	**71.42** #1 (2.89644287)	**0.6867** #1 (0.021920328)	**0.6894** #1 (0.027902532)	1.27 (0.266801403)
WBC	MLP	92.78891 (4.077845215)	0.918684 (0.048510624)	0.921628 (0.044944388)	25.18 (0.111789683)
	SVM	95.146379 (0.699730146)	0.955182 (0.007923006)	0.955642 (0.007307063)	4.13 (3.007353449)
	LDS	96.8146048 (0.406269611)	0.968061 (0.005105233)	0.971522 (0.005105233)	3.33 (0.154653888)
	CCCP-TSVM	97.8146048 #2 (0.5103577)	0.9779224 #2 (0.0055022)	0.9836402 #2 (0.00591107)	1.10 (0.0602)
	SS-SVM	**97.984** #1 (0.649208)	**0.981307** #1 (0.006207)	**0.987071** #1 (0.006359)	7.91 (1.05)
	APSSC	97.57 #3 (0.6298)	0.970376 #3 (0.0025074)	0.971952 #3 (0.0035915)	5.40 (0.2072)

Table 4: Results of paired t-test performed with proposed APSSC versus other semi-supervised methods in terms of p-score

Data	APSSC Vs LDS	APSSC Vs CCCP-TSVM	APSSC Vs SS-SVM
Annular	0.082	0.4963	**0.001** ↑
Ellipse	**0.0021** ↓	0.3449	**0.0251** ↑
Pat2	$\mathbf{1.58 \times 10^{-7}}$ ↑	0.0835	$\mathbf{7.48 \times 10^{-4}}$ ↑
Spiral	**0.0043** ↑	0.3378	**0.0063** ↑
Ionosphere	0.1123	**0.0285** ↓	0.4398
Telugu Vowel	$\mathbf{9.79 \times 10^{-5}}$ ↑	**0.0173** ↑	**0.0301** ↑
Balance Scale	**0.0107** ↑	0.1223	**0.0086** ↑
Sonar	**0.0014** ↑	**0.0254** ↑	**0.0207** ↑
WBC	**0.0138** ↑	**0.0492** ↓	**0.0085** ↓

4. Conclusions

This article presents a novel 'self-training' based semi-supervised classification algorithm using the metaphore of the aggregation pheromone found in natural behavior of real ants. The performance of the proposed method is compared with two supervised (namely, MLP and SVM) and three semi-supervised classification techniques (namely, LDS, CCCP-TSVM, and SS-SVM). The proposed method has the following advantages. (i) No free parameters need to be set by the users, (ii) during the self-training process in each iteration the method updates the covariance matrix of each class, and thereby it is able to capture the shapes of the classes, (iii) as the algorithm has no assumption regarding the data distribution, therefore, it can be applied for data sets having arbitrary distribution. On the other hand, the other semi-supervised methods have many parameters to be set by the user, and the optimal performance of the algorithm is sensitive to the choice of the parameter values (which varies with different data sets). Therefore, it is extremely difficult and time consuming to find the proper tuning of the parameters. In this respect also, the proposed APSSC has a significant advantage over the other semi-supervised counterpart.

The performance of the proposed semi-supervised algorithm is tested using a number of real life and synthetic data sets. Statistical significance of the experimental results (of different data sets) obtained using various semi-supervised methods is evaluated using paired t-test. Results of investigation justify the potentiality of the proposed APSSC algorithm in terms of classification accuracy, macro and micro averaged F_1 measures consuming moderate execution time. In most of the cases, the improvement in results obtained

by the proposed method are found to be statistically significant compared to its other semi-supervised counterparts.

Future work of the proposed method may be directed towards solving real world problems like microarray gene classification, landuse map generation from multi-spectral remotely sensed images etc.

Acknowledgements

Support of the Center for Soft Computing Research, Indian Statistical Institute, Kolkata to carry out this research is thankfully acknowledged by Mr. Anindya Halder. The authors are also thankful to the reviewers for their thorough and stimulating comments, which helped to improve the quality of the manuscript.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. Technical Report TR-2004-06, University of Chicago, 2004.

[2] K.P. Bennett and A. Demiriz. Semi-supervised support vector machines. In D.A. Cohn M.S. Kearns, S.A. Solla, editor, *Advances in Neural Information Processing Systems*, volume 12, pages 368–374. MIT Press, Cambridge, MA, 1998.

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings* 18^{th} *International Conference on*

Machine Learning, pages 19–26. Morgan Kaufmann, San Francisco, CA, 2001.

[4] A. Blum, J. Lafferty, M.R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In *Proceedings of the 21st International Conference on Machine Learning*, pages 97–104, 2004.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. pages 92–100. Morgan Kaufmann Publishers, 1998.

[6] C.C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines*, 2001. URL: http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[7] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

[8] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the* 10^{th} *International Workshop on Artificial Intelligence and Statistics*, pages 57–64, 2005. Source code available at: http://olivier.chapelle.cc/lds.

[9] R. Collobert, F. Sinz, J. Weston, L. Bottou, and T. Joachims. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7:1687–1712, 2006. Source code available at: http://www.kyb.tuebingen.mpg.de/bs/people/fabee/transduction.

[10] N. Cristianini and J.S. Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, first edition edition, 2000.

[11] M. Dorigo and T. Stützle. *Ant Colony Optimization*. Prentice Hall of India Private Limited, New Delhi, India, 2005.

[12] A.P. Englebrecht. *Computational Intelligence: An Introduction*. Wiley, New York, USA, 2007.

[13] A. Ghosh, A. Halder, M. Kothari, and S. Ghosh. Aggregation pheromone density based data clustering. *Information Sciences*, 178(13):2816–2831, 2008.

[14] S. Ghosh, M. Kothari, A. Halder, and A. Ghosh. Use of aggregation pheromone density for image segmentation. *Pattern Recognition Letters*, 30(10):939–949, 2009.

[15] L. Grady and G. Funka-lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *Proceedings of* 8^{th} *the ECCV 04, Workshop on Computer Vision Approaches to Medical Image Analysis and Mathematical Methods in Biomedical Image Analysis*, pages 230–245, Prague, Czech Republic, 2004. Springer.

[16] A. Halder, A. Ghosh, and S. Ghosh. Aggregation pheromone density based pattern classification. *Fundamenta Informaticae*, 92(4):345–362, 2009.

[17] A. Halder, S. Ghosh, and A. Ghosh. Ant based supervised and unsupervised land use map generation from remotely sensed images. In *Proceedings of the world Congress on Nature and Biologically Inspired Computing (NaBIC'09)*, pages 158–163, Coimbatore, India, 2009. IEEE Computer Socity press.

[18] A. Halder, S. Ghosh, and A. Ghosh. Ant based semi-supervised classification. In *Proceedings of the* 7^{th} *International Conference on Swarm Intelligence (ANTS 2010)*, volume 6234/2010, pages 376–383, Brussels, Belgium, 2010. LNCS, Springer-Verlag.

[19] J. Handl and B. Meyer. Ant-based and swarm-based clustering. *Swarm Intelligence*, 1:95–113, 2007.

[20] Simon Haykin. *Neural Networks: A Comperehensive Foundation*. Prentice Hall, second edition, 1999.

[21] T. Joachims. Transductive inference for text classification using support vector machines. pages 200–209. Morgan Kaufmann, 1999.

[22] M. Kim. Discriminative semi-supervised learning of dynamical systems for motion estimation. *Pattern Recognition*, 44(10-11):2325–2333, 2011.

[23] E. Kreyszig. *Introductory Mathematical Statistics*. John Wiley, 1970.

[24] Y. Li, C. Guan, and Z. Chin. A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. *Pattern Recognition Letters*, 29:1285–1294, 2008.

[25] B. Liu, H.A. Abbass, and B. McKay. Density-based heuristic for rule discovery with ant-miner. In *Proceedings of* 6^{th} *Australasia-Japan Joint Workshop on Intelligence Evololution System*, pages 180–184, Canberra, Australia, 2002.

[26] B. Liu, H.A. Abbass, and B. McKay. Classification rule discovery with ant colony optimization. In *Proceedings of IEEE/WIC International*

Conference Intellelligent Agent Technology, pages 83–88. IEEE Computer Society, 2003.

[27] A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, and M. Saerens. Semi-supervised classification and betweenness computation on large, sparse, directed graphs. *Pattern Recognition*, 44(6):1212–1224, 2011.

[28] D. Martens, M.D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens. Classification with ant colony optimization. *IEEE Transactions on Evolutionary Computing*, 11(5):651–665, 2007.

[29] T. Mitchell. The role of unlabeled data in supervised learning. In *Proceedings of the* 6^{th} *International Colloquium on Cognitive Science*, 1999.

[30] D. J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases. URL: http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[31] K. Nigam, A.K. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[32] S.K. Pal and D. Dutta Majumder. Fuzzy sets and decision making approaches in vowel and speaker recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 7:625–629, 1977.

[33] S.K. Pal and S. Mitra. Fuzzy versions of Kohonen's net and MLP-based classification: Performance evaluation for certain nonconvex decision regions. *Information Sciences*, 76:297–337, 1994.

[34] R.S. Parpinelli, H.S. Lopes, and A.A. Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computing*, 6(4):321–332, 2002.

[35] I. Sathish Reddy, Shirish Shevade, and M.N. Murty. A fast quasi-newton method for semi-supervised SVM. *Pattern Recognition*, 44(10-11):2305–2313, 2011.

[36] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the* 7^{th} *conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 25–32, Stroudsburg, PA, USA, 2003. Edmonton, Canada, Association for Computational Linguistics.

[37] M. H. Rohban and H. R. Rabiee. Supervised neighborhood graph construction for semi-supervised classification. *Pattern Recognition*, 45(4):1363–1372, 2012.

[38] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Proceedings of the* 7^{th} *IEEE Workshop on Applications of Computer Vision*, pages 29–36, 2005.

[39] G. Salton and M.J. McGill. *An Introduction to Modern Information Retrieval*. McGrawHill, 1983.

[40] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

[41] M. Shiga and H. Mamitsuka. Efficient semi-supervised learning on locally informative multiple graphs. *Pattern Recognition*, 45(3):1035–1049, 2012.

[42] 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*, pages 207–212, United Kingdom, December 2004.

[43] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[44] V. Vapnik and A. Sterin. On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control*, 10(3):1495–1503, 1977.

[45] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[46] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, and J. You. Semi-supervised classification based on random subspace dimensionality reduction. *Pattern Recognition*, 45(3):1119 – 1135, 2012.

[47] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.

[48] X. Zhu. Semi-supervised learning literature survey. Technical Report Computer Sciences TR 1530, University of Wisconsin, Madison, July 2008.

[49] X. Zhu and A.B. Goldberg. *Introduction to Semi-supervised Learning.* Morgan & Claypool Publishers, 2009.

Anindya Halder received the B.Sc. (with honors) in Physics from the University of Calcutta, India, in 2000, and the Master of Computer Application form University of Kalyani, in 2005. He is presently working as an Assistant Professor in Department of Computer Application, North-Eastern Hill University, Tura Camus, Meghalaya, India. He worked (towards Ph.D.) as a Research Scholar at the Center for Soft Computing Research, Indian Statistical Institute (ISI), Kolkata, India during August 2007 to July 2012. He was a visiting scientist at Remote Sensing Laboratory, University of Trento, Italy during January 2010 to June 2011. He also worked as project linked personnel in Machine Intelligence Unit, ISI, Kolkata from December 2006 to June 2007. He was a Lecturer in Department of Computer Science at Kalyani Mahavidyalaya (College), Kalyani, India from August 2005 to December 2006. He has already published a couple of research articles in internationally reputed journals and refereed conferences. He won (with A. Ghosh and S. Ghosh) the best innovative paper award in the World Congress on Nature and Biologically Inspired Computing (NaBIC'09), Coimbatore, India during December 9-11, 2009. He has been selected for international travel support scheme for young scientist by Department of Science and Technology (DST), Government of India to present a paper in 7th International Conference on Swarm Intelligence, ANTS 2010, Brussels, Belgium, during September 8-10, 2010. His research interests include, Machine Learning, Pattern Recognition, Swarm intelligence, Soft Computing, Remote Sensing Image Analysis etc.

Susmita Ghosh received the B.Sc. (with honors) in Physics and the B.Tech. in Computer Science and Engineering from the University of Calcutta, India, in 1988 and 1991, respectively, the M.Tech. in Computer Science and Engineering from the Indian Institute of Technology, Bombay, in 1993, and the Ph.D. (Engineering) from Jadavpur University, Kolkata, in 2004. She has visited various universities/academic institutes for collaborative research/ international projects and delivered lectures in different countries including Australia, China, Italy, Japan and the USA. She has published many research papers in internationally reputed journals and referred conferences and has edited 3 books. She was a Lecturer and then a Reader with the Department of Computer Science and Engineering, Jadavpur University since 1997. She is currently an Associate Professor at the same department. Her research interests include genetic algorithms, neural networks, image processing, pattern recognition, soft computing, and remotely sensed image analysis.

Ashish Ghosh is a Professor with the Machine Intelligence Unit and is In-charge, Center for Soft Computing Research, Indian Statistical Institute. He has already published more than 150 research papers in internationally reputed journals and refereed conferences, and has edited eight books. His current research interests include pattern recognition and machine learning, data mining, image analysis, remotely sensed image analysis, video image analysis, soft computing, fuzzy sets and uncertainty analysis, neural networks, evolutionary computation, and bioinformatics. Dr. Ghosh received the prestigious and most coveted Young Scientists Award in Engineering Sciences from the Indian National Science Academy in 1995, and in Computer Science from the Indian Science Congress Association in 1992. He was selected as an Associate of the Indian Academy of Sciences, Bangalore, India, in 1997. He is a member of the founding team that established the

National Center for Soft Computing Research at the Indian Statistical Institute, Kolkata, in 2004, with funding from the Department of Science and Technology, Government of India, and is currently the In-charge of the Center. He is acting as a member of the editorial boards of various international journals.

-> Proposed novel ant based semi supervised classification algorithm

-> The algorithm is self-training in nature

-> Proposed method is parameter free with no assumption regarding data distribution

-> Method can adaptively capture arbitrary shapes of classes

-> Potentiality of the method is justified from the experimental results.