Neurocomputing 73 (2009) 438–448

Contents lists available at ScienceDirect

Neurocomputing

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

ACO-based hybrid classification system with feature subset selection and model parameters optimization

Cheng-Lung Huang *

Department of Information Management, National Kaohsiung First University of Science and Technology, 2, Juoyue Rd., Nantz District, Kaohsiung 811, Taiwan, ROC

ARTICLE INFO

Article history:
Received 19 February 2008
Received in revised form
21 April 2009
Accepted 27 July 2009
Communicated by T. Heskes
Available online 19 August 2009

Keywords:
Ant colony optimization
Feature selection
Support vector machine

ABSTRACT

This work presents a novel hybrid ACO-based classifier model that combines ant colony optimization (ACO) and support vector machines (SVM) to improve classification accuracy with a small and appropriate feature subset. To simultaneously optimize the feature subset and the SVM kernel parameters, the feature importance and the pheromones are used to determine the transition probability; the classification accuracy and the weight vector of the feature provided by the SVM classifier are both considered to update the pheromone. The experimental results indicate that the hybridized approach can correctly select the discriminating input features and also achieve high classification accuracy.

1. Introduction

Many practical pattern classification tasks require the learning of a classification function that assigns a given input pattern, usually represented by a vector of attribute values, to a finite set of classes. Feature selection is employed to identify a powerfully predictive subset of fields in a database and reduce the number of fields presented to the data mining. Significant computation time can thus be saved and the constructed models can generalize well for unseen data by extracting as much information as possible from a given data set while using the smallest number of features [1]. The choice of features used to represent patterns that are presented to a classifier influences several aspects of pattern classification, including the accuracy of the learned classification algorithm, the time required to learn a classification function, the number of examples required for learning, and the cost associated with the features [2]. The feature subset selection algorithms have been classified into two categories based on whether feature selection is performed independently of the learning algorithm that constructs the classifier—the filter approach and the wrapper approach [3–5]. The filter approach initially selects important features and then the classifier is used for classification. The wrapper approach either modifies the classifier to select important features or combines the classifier with other optimization tools to select features.

A range of machine learning approaches have been developed to build classifiers, including artificial neural networks, k-nearest neighbor algorithms and support vector machines (SVM). SVM [6] has recently been adopted to solve a range of problems. Feature subset selection is an important issue in building an SVM-based classification model (as well as other classification models, such as the neural networks). In addition to the feature selection, a kernel function, kernel parameters and a soft margin constant C (also called the regularization parameter) [6] must be determined to construct an accurate SVM classifier. To identify the best subset of features, a wrapper-based system typically combines a classifier with stochastic optimization techniques, including simulated annealing, genetic algorithms and ant colony optimization.

Ant colony optimization (ACO) [7–9] is an artificial system inspired by the behavior of real ant colonies and is used to solve discrete combinatorial optimization problems [10,11]. Ants deposit pheromones along their trail to a food source. At a decision point, they make a probabilistic choice based on the amount of pheromone along each search branch. ACO has been used as a search procedure for selecting features; and it has been combined with the artificial neural network classifier [12–14], the nearest neighbor classifier [15], rough set theory [16–18] or SVM [19–22]. As well as feature selection, the proper setting of parameters for the classifier can also increase classification accuracy. Both feature subset selection and model parameter setting substantially influence classification accuracy [23,24]. The optimal feature subset and model parameters must be determined simultaneously, since feature subset selection affects the appropriate model (kernel) parameters and vice versa [24]. Since research on the simultaneous optimization of the feature subset

* Tel.: +886 7 6011000x4127; fax: +886 7 6011042.
E-mail address: clhuang@ccms.nkfust.edu.tw

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.07.014

and model parameters using ACO is lacking, this investigation proposes an intelligent system that incorporates two newly developed techniques, ACO and SVM, to solve this problem.

In designing an ACO-based system, the construction of a traversal path for an ant, the pheromone update strategy, and the design of the transition probability are critical. In the proposed scheme, the pheromone and heuristic information are used to determine the transition probability; a solution is generated for an ant that uses the transition probability to determine the classifier's model parameters and the input features; the SVM classifier evaluates the quality of the solution in terms of the classification accuracy and the weight vector of the SVM model; the solution quality is then used to update the pheromone tables. The proposed ACO–SVM is a wrapper-based hybrid system; thus the overall classification accuracy and the feature importance provided by the classifier can be integrated together into the ACO algorithm. But a filter-based approach, which selects feature subset first and then trains the classifier, cannot tightly combine the overall classification accuracy and feature importance provided by the classifier. While the filter approach is generally computationally more efficient than the wrapper approach, however, its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm that is used to construct the classifier [25]. The proposed hybrid system can avoid this drawback.

This paper is organized as follows. Section 2 addresses work related to the basic SVM and ACO concepts. Section 3 describes the proposed ACO–SVM hybrid system. Section 4 presents the experimental results obtained using a simulated dataset and three public datasets. Section 5 draws conclusions.

2. Brief introduction to the ant colony algorithm and support vector machines

2.1. Support vector machines

Given a training set of instance-label pairs $(\mathbf{x}_i, y_i)$, $i = 1, 2, \ldots, m$ where $\mathbf{x} \in R^n$ and $y \in \{+1, -1\}$, the SVM finds an optimal separating hyperplane, $d(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$, by solving the following optimization problem:

$$\underset{\mathbf{w},b,\xi}{\text{Minimize}}\ \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\xi_i$$

$$\text{Subject to}: y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) + \xi_i - 1 \geq 0; \xi_i \geq 0 \tag{1}$$

where C is a regularization parameter associated with the training error; ξ_i is the non-negative slack variable, and ϕ is a mapping function that maps the training samples from the input space into a higher-dimensional feature space.

The optimal hyperplane provides the minimum number of training errors (to keep the constraint violation as small as possible), and can be solved by introducing Lagrange multipliers for the dual optimization model [26–28]. After the optimal hyperplane has been solved, the decision function (classifier) is given by,

$$\text{sign}(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \tag{2}$$

or

$$\text{sign}\left(\sum_{i=1}^{m}\alpha_i y_i[\phi^T(\mathbf{x}_i)\phi(\mathbf{x})] + b\right) \tag{3}$$

In Eq. (3), the required scalar product $\phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ is calculated directly by computing the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ for given training data in an input space. The radial basis function (RBF) is a common kernel function, as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2) \tag{4}$$

The multi-class classifier is based on two major multi-class SVM classification strategies—"one-against-all" and "one-against-one." (1) In the one-against-all strategy [29], ν binary SVM decision functions are constructed for an ν-class problem. The jth ($j = 1, 2, \ldots, \nu$) decision function is trained by labeling all of the examples in the jth class with positive labels, and all of the examples that are not in the jth class with negative labels. A new $\mathbf{x}$ is classified into the class that has the largest decision function. (2) In the one-against-one strategy [30,31], $C_2^{\nu} = \nu(\nu-1)/2$ classifiers are constructed, and each classifier is trained using two classes (such as class c_i vs. class c_j). A new $\mathbf{x}$ is classified into the majority class that is voted on by all of the decision functions.

2.2. Ant colony optimization

The ant-based algorithm, used to solve mainly combinatorial optimization problems, was inspired by observations of the foraging behavior of real ants. The first ACO was developed to solve the classical traveling salesman problem (TSP) [8,9]. This section introduces the basic concepts based on which the standard ACO algorithm is used to solve the TSP problem. More details can be found elsewhere [32].

(1) *Transition probability*: In the TSP problem, the transition probability from city i to city j for the kth ant at time step t is expressed as,

$$PROB_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{j\in I_i^k}[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}} & \text{if } j \in I_i^k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\tau_{ij}(t)$ is the amount of pheromone trail on edge (i,j) at time t; η_{ij} is the *a priori* available heuristic information; α and β are two factors that specify the relative effects of pheromone trail and heuristic information; and I_i^k is the set of feasible neighborhood cities that have not yet been visited by ant k.

(2) *Pheromone update*: The solution is generated after each ant has completed a tour. Then, the pheromone trails are updated by initially lowering them with a constant evaporation rate and then allowing each ant to deposit pheromone on the arcs that are part of its tour, as indicated in the following equation:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{NumberOfAnts}\Delta\tau_{ij}^k \tag{6}$$

where ρ is the pheromone trail evaporation rate $(0<\rho<1)$. The parameter ρ is used to prevent unlimited accumulation of the pheromone trails and enables the algorithm to "forget" previously made bad decisions. On arcs that are not selected by the ants, the associated pheromone strength declines exponentially with the number of iterations. $\Delta\tau_{ij}^k$ is the quantity per unit of length of the trail substance that is laid on edge (i, j) by the kth ant.

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k} & \text{if ant } k \text{ uses edge}(i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where L_k denotes the tour length, and Q is a predefined constant.