Applied Soft Computing 64 (2018) 497–507

Contents lists available at ScienceDirect

Applied Soft Computing

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

A multi-class learning method for multiconlitron using hybrid binary tree architecture

Qiangkui Leng[a,*], Yuping Qin[b], Yujian Li[c]

[a] College of Information Science and Technology, Bohai University, Jinzhou 121000, China
[b] College of Engineering, Bohai University, Jinzhou 121000, China
[c] College of Computer Science, Beijing University of Technology, Beijing 100124, China

ARTICLE INFO

Article history:
Received 6 May 2017
Received in revised form 7 December 2017
Accepted 25 December 2017
Available online 28 December 2017

Keywords:
Multi-class classification
Hybrid binary tree
Multiconlitron
Piecewise linear classifier
Support vector machine

ABSTRACT

Multiconlitron is a general geometric method for constructing piecewise linear classifiers, but it was initially designed only for two-class problem. In this paper, we propose a multi-class learning method of multiconlitron by using a hybrid binary tree architecture. At each internal node that does not generate leaf nodes, a hyperplane is first created as perpendicular bisectors of line segment linking centroids of the two farthest classes from each other. Then, according to the positive or negative sides on the hyperplane, all the inherited classes are divided into two groups for the next iteration. For an internal node that will generate leaf nodes, the multiconlitron is constructed by support multiconlitron algorithm, which can separate one class from the other class (or group). Generally, the approximate hyperplane by centroids can provide fast division in the early stages of the training phase, whereas the ensemble boundaries with multiconlitron will perform the final precise decision. As a result, a hybrid binary partition tree is created which represents a hierarchical division of given classes. Experimental results show that the proposed method is better than one-versus-one multiconlitron and directed acyclic graph multiconlitron, both in terms of classification effectiveness and computational time. Moreover, comparison with another tree-based multi-class piecewise linear classifier verifies its competitiveness and superiority.

1. Introduction

In pattern recognition, piecewise linear classifier (PLC) is effective when a statistical model cannot express the underlying distribution of samples [1]. It approximates the true classification boundary by a combination of hyperplanes. Since each piece is linear, a PLC is very simple to implement with low memory usage requirements. The advantage makes it very suitable for small reconnaissance robots, intelligent cameras, imbedded and real-time systems, and portable devices [2].

Motivated by the promising applications, many methods have been presented for constructing PLCs. Linear programming [3,4] might be a very early one. It needs to carry out optimization many times, involving a lengthy computation. Locally training [1,5,6] was seen as the stat-of-the-art technology for designing PLCs, but it often makes concessions to over-fitting or under-fitting. Decision tree [2,7] is another strategy to construct PLCs. Nevertheless, it has to pay a lot of effort to keep the error rate under a desirable threshold.

A PLC can also be constructed by an approach of max–min separability [8,9], which is regarded as a generalization of polyhedral separability [10]. In this approach, a non-convex and non-smooth error function is minimized by the discrete gradient method. It is proved that any continuous piecewise linear function can be represented as a max–min of linear functions. Furthermore, the derived result can be employed to design a general framework for description of piecewise linear boundaries between two finite point sets. However, the max–min approach might require a pre-specified set of integers for describing how to organize linear functions in groups.

Despite other existing methods, multiple convex linear perceptron (Abbr. multiconlitron) is presented as a general geometric method for designing PLCs [11]. Theoretically, a multiconlitron can be dynamically constructed without pre-specified number of hyperplanes and distribution assumption. From the viewpoint of large margin, multiconlitron can be regarded as the nonkernel generalization of support vector machines (SVMs). Accordingly, its original construction algorithm is called support multiconlitron

* Corresponding author.
E-mail addresses: qkleng@gmail.com (Q. Leng), jzqinyuping@gmail.com (Y. Qin), liyujian@bjut.edu.cn (Y. Li).

https://doi.org/10.1016/j.asoc.2017.12.035
1568-4946/

algorithm (SMA). Using SMA, a series of hyperplanes are generated, and they are structurally assembled into a multiconlitron.

Multiconlitron was initially designed for binary classification. However, most real-world scenarios, such as handwritten digit recognition, text categorization, and face recognition, correspond to multi-class classification problems. Therefore, an issue worthy of concern is how to effectively extend the multiconlitron to its multi-class version.

Currently, the multi-class classification problem is solved by two main strategies, namely “direct multi-class representation” and “indirect decomposition design” [12,13]. The direct one employs the adaptive multi-class classifiers forthwith to classify multiple labels. It includes some commonly used algorithms, such as multi-class SVMs [14], decision tree [15], neural networks [16], and deep learning (a variant of neural networks) [17]. However, it is difficult to solve the multi-class problem using the direct strategy because of the increase in the number of classes and the size of the training data [18]. In contrast, the indirect one decomposes the original multi-class problem into multiple binary sub-problems [12,19–25]. It is usually easier to build a classifier to distinguish only two classes, since the decision boundaries can be simpler than that of multiple classes [26]. In this paper, we focus on the indirect decomposition methods.

Among these decomposition methods, binary SVMs have been well extended for performing multi-class tasks [19,20]. In general, for a $K(\geq 2)$-class problem, multiple binary SVMs are trained, each of which solves a sub-problem of the original problem. In the testing phase, a testing sample is introduced to all (or part of) the binary classifiers and their responses are combined in order to provide the final classification result [27].

The well-known methods for multi-class binarization SVMs include one-versus-one [20,28], one-versus-all [29,30], directed acyclic graph [31,32], and binary tree architecture [33–35]. For a K-class problem, we here make a brief introduction of these methods.

(1) One-versus-one: It is to build $K(K-1)/2$ binary classifiers, each of which separates a pair of classes. A testing sample goes through all the binary classifiers, and the final decision is made with majority voting (see Fig. 1a).
(2) One-versus-all: Each classifier is trained between one class and the remaining $K-1$ classes. As a result, K binary classifiers are obtained. The decision output is associated with the class whose decision function has the maximum value (see Fig. 1b).
(3) Directed acyclic graph: A rooted binary directed acyclic graph is built. It consists of $K(K-1)/2$ internal nodes and K leaf nodes. When a testing sample reaches the leaf node, the final decision will be made (see Fig. 1c).
(4) Binary tree architecture: It constructs a rooted binary tree by recursively dividing all current classes into two sub-classes. Each leaf node that contains only a single class can make a decision output. A testing sample goes through at most $\log_2{}^n$ binary classifiers (see Fig. 1d).

For one-versus-one and one-versus-all, too many binary tests are required to make a final decision. Although directed acyclic graph may be able to reduce the testing number, it is still somewhat undesirable when the number of classes is large. As for binary tree architecture, it only needs to train $K-1$ binary classifiers and test $\log_2{}^K$ times, where a fast decision is achieved. This architecture, implemented by different researches [33–35], has one feature in common. That is, it adopts a “divide-and-conquer” strategy, and decomposes the original problem into smaller sub-problems.

In this paper, our aim is to generalize the multiconlitron to its multi-class extension. With a reference of Kostin’s work [2], we propose a hybrid binary tree architecture for constructing multi-

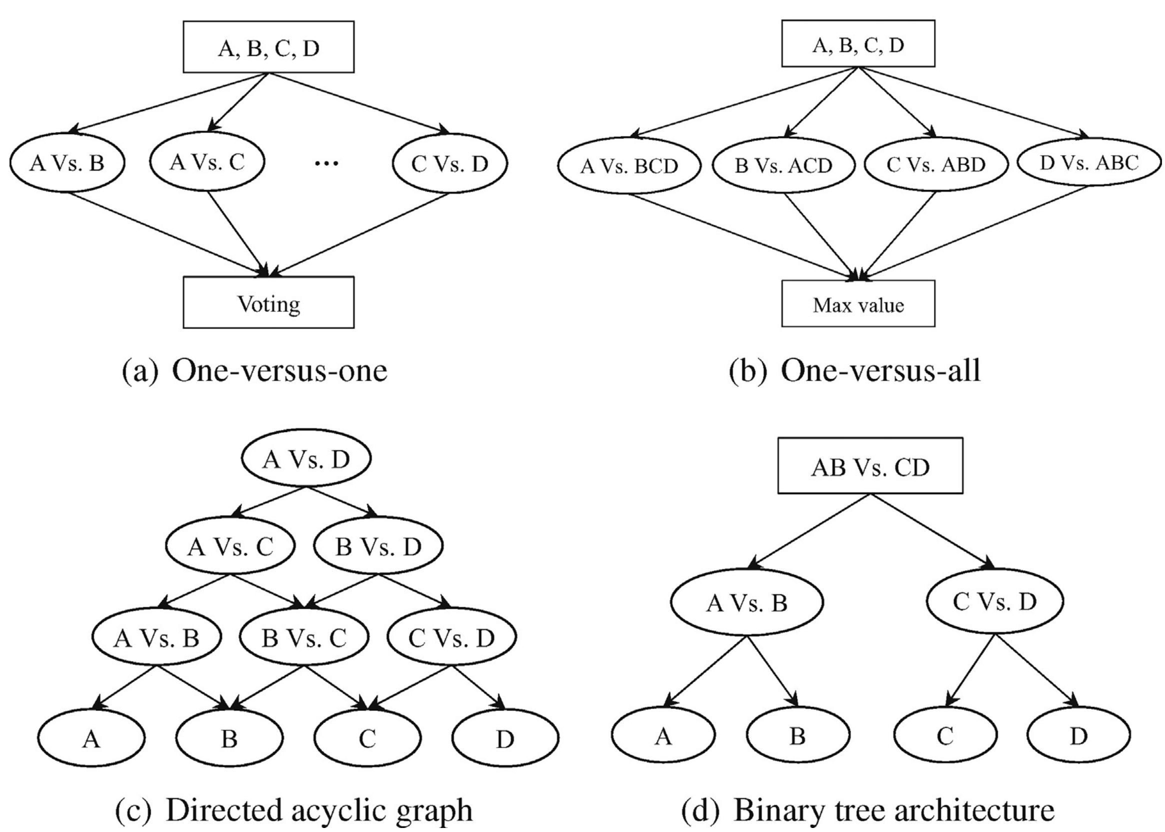

Fig. 1. Multi-class SVMs.