Pattern Recognition 48 (2015) 968–975

Contents lists available at ScienceDirect

Pattern Recognition

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

Alternating multiconlitron: A novel framework for piecewise linear classification

Yujian Li*, Qiangkui Leng

College of Computer Science, Beijing University of Technology, Beijing 100124, China

ARTICLE INFO

Article history:
Received 27 March 2014
Received in revised form
17 August 2014
Accepted 6 September 2014
Available online 16 September 2014

Keywords:
Multiconlitron
Alternating multiconlitron
Maximal convexly separable subset
Support alternating multiconlitron algorithm
Piecewise linear classifier

ABSTRACT

Multiconlitron is a general framework for designing piecewise linear classifiers, but it may contain a relatively large number of conlitrons and linear functions. Based on the concept of maximal convexly separable subset (MCSS), we propose alternating multiconlitron as a novel framework for piecewise linear classification. Using the support alternating multiconlitron algorithm, an alternating multiconlitron can be constructed as a series of conlitrons alternately from a subset of one class to the MCSS of the other class. Experimental results show that in practice an alternating multiconlitron generally has a much simpler structure than a corresponding multiconlitron, performing very fast in testing phase with similar or better accuracies.

1. Introduction

As a fundamental problem in pattern recognition, piecewise linear classifier (PLC) design has been a research topic attracting a lot of attention. A PLC extends a linear boundary for approximating an arbitrarily complicated nonlinear boundary [1], with the advantage of easy implementation suitable for small reconnaissance robots, intelligent cameras, imbedded and real-time systems, and portable devices [2]. However, it is a challenging and complicated task to synthesize a PLC with an appropriate number of hyperplanes.

Over the last few decades, many methods have been presented to construct PLCs. In 1980, Sklansky and Michelotti [3] described a local training method to design PLCs. This method first uses Forgy's algorithm to form prototypes, and then finds all close-opposed pairs of prototypes to produce a set of initial hyperplanes for local training with the data in the prototype regions. Based on the local training method, Park and Sklansky [4] developed a Tomek-link-cutting algorithm to design multi-class PLCs, which sometimes suffers from underfitting due to an insufficient number of hyperplanes. To resolve this problem, Tenmoto [5] used the minimum description length (MDL) criterion to choose an appropriate number of linear hyperplanes while keeping the local training error rate under a threshold. In 1996, Cai et al. [6] employed a binary tree structure and genetic algorithm to design PLCs under the guidance of maximum impurity reduction criterion. In 2006, based on the tree division of subregion centroids, Kostin [2] proposed and implemented a simple and fast multi-class PLC with acceptable classification accuracies. In 2010, Gai and Zhang [7] presented a two-step method to construct a piecewise linear model for classification, where the first step samples some boundary points to give a nonparametric decision surface and the second step uses Dirichlet process mixtures (DPM) to simplify this surface for linear surface segmentation.

A PLC can also be constructed by an approach of max–min separability [8–10], where a non-convex and non-smooth error function is minimized by the discrete gradient to construct a piecewise linear function in the form of a max–min of linear functions. But this method requires a prespecified set of integers for describing how to organize linear functions in groups. Theoretically, a max–min of linear functions may also be called a "multiconlitron", which is recently presented as a general framework for constructing PLCs [11] with a union of conlitrons based on the concept of convex separability. Different from a max–min construction, a multiconlitron can be dynamically constructed in training without a prespecified set of integers for grouping linear functions. In fact, using the support conlitron algorithm (SCA) and the support multiconlitron algorithm (SMA) [11], we can construct support conlitrons and support multiconlitrons as a non-kernel extension of support vector machines, since they can respectively separate two convexly separable and commonly separable data sets by the maximum margin without kernel functions. However, a multiconlitron may contain a relatively large number of conlitrons and linear functions.

In this paper, we propose alternating multiconlitron as a novel framework for designing PLCs, based on the concept of maximal convexly separable subset (MCSS). Using the support alternating

* Corresponding author. Tel.: +86 10 6739 2879 2514; fax: +86 10 6739 1742.
E-mail addresses: liyujian@bjut.edu.cn (Y. Li), qkleng@gmail.com (Q. Leng).

http://dx.doi.org/10.1016/j.patcog.2014.09.003
0031-3203/

multiconlitron algorithm (SAMA), we can construct an alternating multiconlitron as a series of conlitrons alternately from a subset of one class to the MCSS of the other class, which generally has a simpler structure than the corresponding multiconlitron.

The rest of the paper is organized as follows. In Section 2, we discuss the concept of maximal convexly separable subset. In Section 3, we explain the meaning and existence of alternating multiconlitron. Then, we present the support alternating multiconlitron algorithm (SAMA) for constructing an alternating multiconlitron in Section 4, and evaluate its performance in Section 5. Finally, we conclude the paper in Section 6.

2. Maximal convexly separable subset

Let $\mathbf{R}^n$ be the n-dimensional Euclidean space. Throughout this paper, we use X and Y to represent two finite nonempty subsets of $\mathbf{R}^n$ unless specified otherwise. For any $X \subset \mathbf{R}^n$, we use $CH(X)$ to denote the convex hull of X, namely

$$CH(X) = \left\{ \mathbf{x} | \mathbf{x} = \sum_{1 \le i \le |X|} \alpha_i \mathbf{x}_i, \sum_{1 \le i \le |X|} \alpha_i = 1, \mathbf{x}_i \in X, \alpha_i \ge 0, \alpha_i \in \mathbf{R} \right\}. \tag{1}$$

In addition, we define two distance functions as follows:

$$d(\mathbf{x}, \mathbf{y}) = \| \mathbf{x} - \mathbf{y} \| = \sqrt{(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})}, \quad \forall \mathbf{x}, \ \mathbf{y} \in \mathbf{R}^n;$$
$$d(\mathbf{x}, Y) = d(Y, \mathbf{x}) = \inf\{d(\mathbf{x}, \mathbf{y}), \mathbf{y} \in Y\}, \quad \forall \mathbf{x} \in \mathbf{R}^n, \ \ \forall Y \subset \mathbf{R}^n. \tag{2}$$

Furthermore, we use $d(\mathbf{x}, CH(Y))$ to stand for the distance from point $\mathbf{x}$ to the convex hull of Y.

We call that X is convexly separable to Y, if there exists a conlitron – Convex Linear Perceptron (CLP) from X to Y [11]. A conlitron is a set of linear functions, i.e.

$$CLP = \{f_l(\mathbf{x}) = \mathbf{w}_l \cdot \mathbf{x} + b_l, (\mathbf{w}_l, b_l) \in \mathbf{R}^n \times \mathbf{R}, \ 1 \le l \le L\}, \tag{3}$$

satisfying the following two conditions:

$$\forall \mathbf{x} \in X, \quad \forall 1 \le l \le L, \quad f_l(\mathbf{x}) \ge 0;$$
$$\forall \mathbf{y} \in Y, \quad \exists 1 \le l \le L, \quad f_l(\mathbf{y}) < 0. \tag{4}$$

The decision function of a CLP is defined as

$$CLP(\mathbf{x}) = \begin{cases} +1, \ \forall 1 \le l \le L, \ f_l(\mathbf{x}) \ge 0, \\ -1, \ \exists 1 \le l \le L, \ f_l(\mathbf{x}) < 0. \end{cases} \tag{5}$$

"Convexly separable" exactly means that X can be surrounded inside by a set of linear functions that constitute a CLP, excluding Y outside. We have Lemma 1 for two convexly separable data sets.

Lemma 1. *Given two finite nonempty data sets X and Y, X is convexly separable to Y if and only if $Y \cap CH(X) = \emptyset$.*

Actually, Lemma 1 is a restatement of Theorem 7 in Ref. [11]. Suppose Z is a subset of Y. If X is convexly separable to Z, and for any bigger subset $Z'(\supset Z)$ of Y, X is convexly nonseparable to Z', then Z is called a maximal convexly separable subset (MCSS) of Y away from X.

Fig. 1 displays an example of MCSS. After proving two theorems about convex hull, we have an important theorem for the existence of MCSS. Note that in the following theorems, $X \cap Y = \emptyset$ means X and Y are two finite nonempty and nonintersecting data sets. Moreover, they should be taken as training samples from two classes.

Theorem 1. *If $X \cap Y = \emptyset$, then $CH(X) \ne CH(Y)$.*

Proof. Supposing $CH(X) = CH(Y)$, we can get that the two convex hulls must have the same set of vertices. That is, each vertex in the set belongs to both X and Y. This contradicts $X \cap Y = \emptyset$. □

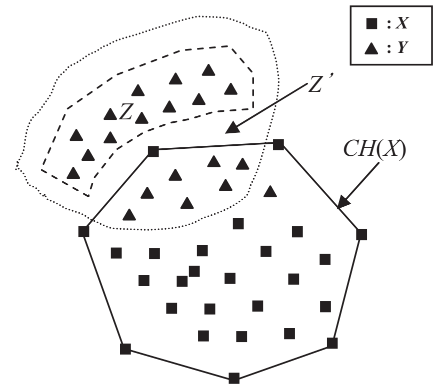

Fig. 1. An example of MCSS.

Notably, Theorem 1 means that if two finite nonempty sets have no common points, then their convex hulls cannot be identical. Accordingly, we obtain Theorem 2.

Theorem 2. *If $X \cap Y = \emptyset$, then $\exists \mathbf{x} \in X$, $\mathbf{x} \notin CH(Y)$ or $\exists \mathbf{y} \in Y$, $\mathbf{y} \notin CH(X)$.*

Proof. Supposing $\forall \mathbf{x} \in X$, $\mathbf{x} \in CH(Y)$ and $\forall \mathbf{y} \in Y$, $\mathbf{y} \in CH(X)$, we have $X \subset CH(Y)$ and $Y \subset CH(X)$. Accordingly, we have $CH(X) \subseteq CH(Y)$ and $CH(Y) \subseteq CH(X)$, i.e., $CH(X) = CH(Y)$. This contradicts Theorem 1. □

Theorem 3. *If $X \cap Y = \emptyset$, there must exist a unique MCSS of Y away from X or a unique MCSS of X away from Y.*

Proof. According to Theorem 2, we might assume that $\exists \mathbf{y} \in Y$, $\mathbf{y} \notin CH(X)$. Thus, we can construct a subset Z of Y, i.e.

$Z = Y - CH(X) = \{\mathbf{y} | \mathbf{y} \in Y, \ \mathbf{y} \notin CH(X)\}$.

According to Lemma 1, X is convexly separable to Z. Therefore, for any bigger subset $Z'(\supset Z)$ of Y, X is convexly nonseparable to Z'. This means Z is a unique MCSS of Y. □

3. Alternating multiconlitron

Using the concept of MCSS, we will define an alternating multiconlitron as a series of conlitrons. If X is convexly separable to Y, they can be separated by a conlitron. In case of $X \cap Y = \emptyset$, they can be separated by a multiconlitron or an alternating multiconlitron.

A multiconlitron is a union of multiple conlitrons. $MCLP = \{CLP_k, 1 \le k \le K\}$ is a multiconlitron from X to Y [11], if and only if

$$\forall \mathbf{x} \in X, \quad \exists 1 \le k \le K, \quad CLP_k(\mathbf{x}) = +1;$$
$$\forall \mathbf{y} \in Y, \quad \forall 1 \le k \le K, \quad CLP_k(\mathbf{y}) = -1. \tag{6}$$

The decision function of a $MCLP$ is defined as

$$MCLP(\mathbf{x}) = \begin{cases} +1, \ \exists 1 \le k \le K, \ CLP_k(\mathbf{x}) = +1, \\ -1, \ \forall 1 \le k \le K, \ CLP_k(\mathbf{x}) = -1. \end{cases} \tag{7}$$

It can be said that each CLP_k in a $MCLP$ from X to Y generates a convex region covering a part of X, and the $MCLP$ is the union of these convex regions for covering the whole set of X. Fig. 2 illustrates the structure of a $MCLP$ from X to Y with one, two or three conlitrons, respectively. Note that if X and Y are commonly separable (i.e., nonintersecting or having no common points), we can always construct multiconlitrons in two directions: one from X to Y (see Fig. 3(a)) and the other from Y to X (see Fig. 3(b)). Furthermore, different conlitrons in a $MCLP$ may overlap each other theoretically (see Fig. 3(a)). However, if one conlitron is nested in another, it will be redundant and can be dropped out. For example, in Fig. 3(a) CLP_2 is redundant, whereas CLP_1 is necessary.