Journal of Machine Learning Research 20 (2011) 33–46 Asian Conference on Machine Learning

Nonlinear Online Classification Algorithm with Probability Margin

Mingmin Chi* and **Huijun He** and **Wenqiang Zhang**
School of Computer Science, Fudan University, 825 Zhang Heng Road, Shanghai, 201203, China

Editor: Chun-Nan Hsu and Wee Sun Lee

Abstract

Usually, it is necessary for nonlinear online learning algorithms to store a set of misclassified observed examples for computing kernel values. For large-scale problems, this is not only time consuming but leads also to an out-of-memory problem. In the paper, a nonlinear online classification algorithm is proposed with a probability margin to address the problem. In particular, the discriminant function is defined by the Gaussian mixture model with the statistical information of all the observed examples instead of data points. Then, the learnt model is used to train a nonlinear online classification algorithm with confidence such that the corresponding margin is defined by probability. When doing so, the internal memory is significantly reduced while the classification performance is kept. Also, we prove mistake bounds in terms of the generative model. Experiments carried out on one synthesis and two real large-scale data sets validate the effectiveness of the proposed approach.

Keywords: Nonlinear online classification, probability margin, Probability product kernel, Gaussian mixture models

1. Introduction

Linear online learning algorithm is a well-studied and popular classification algorithm, where a theoretical analysis can be obtained. In real applications, however, problems are much more complex and often are not linearly separable, such as face tracking and robot navigation. Like offline nonlinear classification problems, generally nonlinear online learning algorithms were introduced by using kernel functions proposed by Aizerman et al. (1964) for addressing nonlinearly separable problems.

The common way for recently proposed nonlinear online learning algorithms is to store a set of the observed examples which are misclassified on each round (Freund and Schapire, 1999; Li and Long, 2002; Gentile, 2001) or have a low prediction confidence by an online algorithm (Dekel et al., 2006; Crammer et al., 2006; Orabona et al., 2008). After a nonlinear transform, a linear online learning algorithm can be applied while at the expense of an increase of the input dimensionality. A rapid growth of the set leads to the explosion of internal memory. Also, from a computational point of view, computing the kernel values can become prohibitively hard.

To address the problem, the most of the recent proposed nonlinear approaches used a fixed budget of the set by forgetting the observed examples out of budget window size, such as the algorithms (Kivinen et al., 2004; Cheng et al., 2006; Crammer et al., 2004) (where

* Correspondence author: mmchi@fudan.edu.cn.

no mistake bound was derived) and the Forgetron (Dekel et al., 2006), Random Budget Perceptron (RBP) (Cavallanti et al., 2007) (where a relative mistake bound was derived). A different approach is the Projectron (Orabona et al., 2008), where the hypothesis is projected onto the subspace spanned by the set and so the size of the internal memory is bounded. Also, a relative mistake bound can be derived for the Projectron.

In the paper, we propose a new algorithm for nonlinear online classification which combines the Gaussian mixture model and an online discriminative learning method, where the margin is defined in terms of probability. Our algorithm takes advantage of statistical modeling of a sequence of instances with the Gaussian mixture model that is linearly separable in the probabilistic space. Compared to (Freund and Schapire, 1999; Dekel et al., 2006; Crammer et al., 2006; Orabona et al., 2008), ours is much more efficient in terms of internal memory, and also in terms of computational time. Also, we prove mistake bounds in terms of the mixture model. Experiments carried out on one synthesis and two real large-scale data sets validate the effectiveness of the proposed approach: the classification accuracies provided by our algorithm is superior to the Forgetron and the Projectron on the same problems, while saving significantly spatial and computational complexities.

The rest of the paper is organized as follows. The next section describes the proposed Nonlinear Online Classification Algorithm with probability margin (abbreviated as NOCApm). Section 3 gives a theoretical analysis both on the internal memory and the mistake bound. Section 4 gives the data used in the experiments, reports and discusses the results provided by different algorithms. Finally, conclusions and discussion are given in Section 5.

2. The NOCApm algorithm

Online learning takes place in rounds. Assume that the initial hypothesis be zero, i.e., $f_0 = \mathbf{0}$ and the one at round t be denoted by $f(\mathbf{x}_t) \doteq f_t$. At each time, the algorithm receives an instance $\mathbf{x}_t \in \mathcal{R}^D$ in D-dimensional space and the corresponding label y_t, and the prediction result is computed by $\hat{y}_t = \text{sign}(f(\mathbf{x}_t))$. Usually, there are two ways for updating the prediction hypothesis function. The first one is classification error driven, i.e., if the predicted label $\hat{y}_t$ is different from the real one y_t, the hypothesis is updated, e.g., perceptron-like algorithms in a linear case, $f_{t+1} = f_t + y_t\mathbf{x}_t$. Alternatively, the confidence, the absolute of margin $|y_t f(\mathbf{x}_t)|$, is used to decide whether the prediction hypothesis should be updated or not. Usually, a convex loss is defined for the learning problem, e.g., the hinge loss, where if $y_t f(\mathbf{x}_t) \leq \rho$, the prediction function is updated; and otherwise it is kept, i.e., $f_{t+1} = f_t$.

The proposed nonlinear online classification algorithm falls in the margin error driven category. Usually, the main difference among the online algorithms driven by margin error is of the definition of an updating rule.

2.1. The Problem Setting

The common difficulty for nonlinear online learning algorithms is to store kernel functions whose size grows with the learning process. This can lead to explosion of internal memory. Moreover, kernel values should be recomputed on each round, which is time-consuming. In the paper, the Gaussian mixture model learned from the statistical information of data sequence is used for online learning instead of data points such that the size of memory

can be significantly reduced. Therefore, computational complexity and memory explosion problems can be avoided.

Assuming that for each class in online learning, data are represented by a distribution of K-Gaussian mixture:

$$\begin{aligned} P(\mathbf{x}|\boldsymbol{\Theta}) &= \sum_{k=1}^{K} p_k(\mathbf{x}|\boldsymbol{\theta}_k) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

where $\boldsymbol{\theta}_k = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ denotes the k^{th} component with the prior π_k, the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$ (*for the ease of computation, only the diagonal components are computed and the non-diagonal components are set to zero*), and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k | k = 1, \cdots, K\}$. We also give each component of Gaussian a label z_k. Then, a conditional probability for an incoming instance $\mathbf{x}_t$ assigned to the class c is expressed in the form:

$$P_c(\mathbf{x}|\boldsymbol{\Theta}) = \frac{\sum_{z_k=c} p_k(\mathbf{x}|\boldsymbol{\theta}_k)}{\sum_{j=1}^{K} p_j(\mathbf{x}|\boldsymbol{\theta}_j)}$$

where z_k denotes the class label of $\boldsymbol{\theta}_k$.

In the analysis hereafter, without loss of generality, we focus on the case of binary classification, i.e., each instance has label $y_t \in \{-1, 1\}$ and $z_t \in \{-1, 1\}$ for each component. We can define the decision hyperplane as

$$P_+(\mathbf{x}|\boldsymbol{\Theta}) - P_-(\mathbf{x}|\boldsymbol{\Theta}) = 0. \tag{1}$$

It is possible to find a maximum-margin separating hyperplane having the same distance from the two classes. If we use the label as a sign indicator, (1) becomes

$$\sum_{k=1}^{K} z_k \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0 \tag{2}$$

by omitting $\sum_{j=1}^{K} p_j(\mathbf{x}|\boldsymbol{\theta}_j)$ as it is the same for all the classes.

To link the probability distribution with the kernel theory, we first define the distribution for a single instance $\mathbf{x}_t$ as $p(\mathbf{x}_t|\boldsymbol{\theta}_{\mathbf{x}_t})$, $\boldsymbol{\theta}_{\mathbf{x}_t} := \{\pi = 1, \boldsymbol{\mu} = \mathbf{x}_t, \boldsymbol{\Sigma} = \alpha^2\mathbf{I}\}$, where $\mathbf{I}$ denotes an identity matrix and $\alpha \propto 0$. Thus, the classification hypothesis can be represented by `probability product kernel` (PPK) (Jebara et al., 2004) as:

$$f(\mathbf{x}_t) = \sum_{k=1}^{K_t} z_k \boldsymbol{\kappa}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{\mathbf{x}_t}). \tag{3}$$

where

$$\begin{aligned} \boldsymbol{\kappa}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{\mathbf{x}_t}) &= \int_{\mathbb{R}^D} \boldsymbol{\theta}_k \boldsymbol{\theta}_{\mathbf{x}_t} d\mathbf{x} \\ &= p_k(\mathbf{x}_t|\boldsymbol{\theta}_k). \end{aligned} \tag{4}$$

Here, K_t is the number of mixture components in round t. Due to online properties, K_t is adaptive and varies with rounds. Note that the definition aforementioned is actually a Dirac function.

Theorem 1 (Reproducing property) *The hypothesis $f(\mathbf{x})$ is defined in a Hilbert space $\mathcal{H}$ with the probability product kernel that satisfies the reproducing property*

$$\langle f, \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}).$$

Proof Considering the hypothesis

$$f(\cdot) = \sum_{k=1}^{K_t} z_k \boldsymbol{\kappa}(\boldsymbol{\theta}_k, \cdot) \tag{5}$$

and a special case for an instance $\mathbf{x}_t$ on the round t, $\boldsymbol{\theta}_{\mathbf{x}_t}$, using the definition of the PPK, we have

$$\boldsymbol{\kappa}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_{\mathbf{x}_t}) = p_k(\mathbf{x}_t|\boldsymbol{\theta}_k),\ k = 1, \ldots, K_t.$$

Therefore, we have

$$\begin{aligned} f(\mathbf{x}_t) &= \sum_{k=1}^{K_t} z_k p_k(\mathbf{x}_t|\boldsymbol{\theta}_k) \\ &= \langle f, \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)\rangle_{\mathcal{H}}. \end{aligned}$$

■

With the PPK-form definition of hypothesis, the mixture of Gaussian model is ready for nonlinear online classification.

2.2. The Updating Rule

The way to evaluating the prediction of a hypothesis is via the `hinge-loss` function, which is defined as

$$L_\rho(f; (\mathbf{x}, y)) = \begin{cases} 0 & yf(\mathbf{x}) \geq \rho \\ \rho - yf(\mathbf{x}) & \text{otherwise} \end{cases} \tag{6}$$

where ρ is a margin parameter and can be predefined or adjusted during learning. The Gaussians $\boldsymbol{\theta}_k$ are hidden in f, i.e., $L_\rho(f; (\mathbf{x}, y)) = L_\rho(f; (\mathbf{x}, y), \boldsymbol{\Theta})$. Accordingly, the proposed nonlinear online classification algorithm is with probability margin, i.e., the confidence of prediction value based on the probability. Namely, if $yf(\mathbf{x}, \boldsymbol{\Theta}) \leq \rho$, the prediction suffers from a margin error with the value $\rho - yf(\mathbf{x}, \boldsymbol{\Theta})$, and otherwise there is no error occurred. For brevity, we will omit $\boldsymbol{\Theta}$ from f in the following. Hence, we can obtain the upper margin $f(\mathbf{x}) = +\rho$ and the lower margin $f(\mathbf{x}) = -\rho$, respectively (cf. Fig. 1).

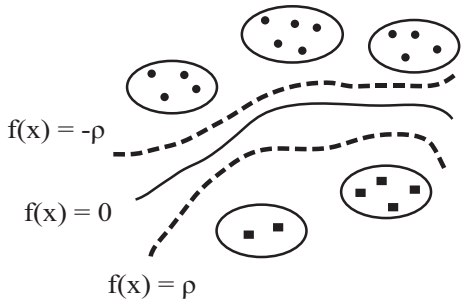

Figure 1: Illustration of the hypothesis and the corresponding margins: The square denotes examples belonging to the positive class and the circle to the negative one. The solid line represents the hyperplane H, which has the same probabilities assigned to the positive class or the negative one. Two dashed lines show the upper margin ρ and the lower $-\rho$ margin, respectively. Namely, for any instance $\mathbf{x}$, if $f(\mathbf{x}) = \rho$, it has a probability $\frac{1}{2} + \rho$ belonging to the positive class, and otherwise if $f(\mathbf{x}) = -\rho$, it has the probability $\frac{1}{2} - \rho$ belonging to the negative class.

If a margin error is occurred, i.e., $yf(\mathbf{x}, \boldsymbol{\Theta}) \leq \rho$, like the marge-based Perceptron (Freund and Schapire, 1999), the updating rule takes the form

$$f_{t+1} = f_t + y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot). \tag{7}$$

If there is no margin error occurred, two cases should be taken into account: if the Euclidean distance of the instance $\mathbf{x}_t$ to the nearest Gaussian (e.g., i-th component) is greater than a threshold ϵ (which is prefixed), the update rule takes the same form as (7); otherwise, the i-th component, i.e., $\boldsymbol{\theta}_i$ in the mixture model is reestimated such that the prediction function is updated as,

$$f_{t+1} = f_t + y_t \boldsymbol{\kappa}(\boldsymbol{\theta}'_i, \cdot) - y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_i, \cdot) \tag{8}$$

where $\boldsymbol{\theta}'_i$ is updated for the i-th component and the previous component was subtracted from the prediction function. The process of reestimation of the mixture model is a sequential update of means and covariance matrices as follows:

$$\begin{aligned} \boldsymbol{\mu}'_k &= \frac{n}{n+1}\boldsymbol{\mu}_k + \frac{1}{n+1}\mathbf{x}_t \\ \boldsymbol{\Sigma}'_k &= \frac{n}{n+1}\boldsymbol{\Sigma}_k + \frac{1}{n+1}\mathbf{x}_t\mathbf{x}_t^T \end{aligned}$$

where n is the number of update occurrence and is kept track of along with the Gaussian parameters.

In summary, when the prediction function suffers from a loss, or the minimal Euclidean distance from the new arrival example $\mathbf{x}_t$ to the existing Gaussian components is greater

than the predefined threshold ϵ (the distance is denoted by $d(x_t, \boldsymbol{\theta}_k)$), a new kernel is generated and added to the prediction for updating, where $z_{K_t+1} = y_t$, and $\boldsymbol{\theta}_{K_t+1} = \boldsymbol{\theta}_{x_t}$. Therefore, the final update rule takes the form:

$$\begin{aligned} f_{t+1} = f_t &+ \sigma_t y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot) \\ &+ (1 - \sigma_t)(y_t \boldsymbol{\kappa}(\boldsymbol{\theta}'_i, \cdot) - y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_i, \cdot)) \end{aligned} \tag{9}$$

where

$$\sigma_t = \begin{cases} 1 & \text{if } L_\rho(f_t; (\mathbf{x}_t, y_t)) \geq 0 \ \vee \ \forall k, d(x_t, \boldsymbol{\theta}_k) > \epsilon \\ 0 & \text{otherwise.} \end{cases}$$

The number of Gaussian components K_t is automatically estimated according to the distribution of data set adaptively. More Gaussian components lead to a higher classification accuracy while more computational complexity. To balance both, the confidence parameter ρ is adjusted on round t when a margin error occurred, and is decreased by a tuning parameter τ ($\tau < 1$) in the form

$$\rho_{t+1} = \tau \rho_t. \tag{10}$$

After rounds of learning, the mixture distribution converges and new kernels have a less and less impact on the prediction hypothesis. Therefore, the setting of τ makes the error margin narrower with rounds, which provides a speed trade-off between model convergence and learning rate.

The proposed Nonlinear Online Classification Algorithm with probability margin (NOCApm) is summarized in Algorithm 1.

3. Analysis

We first define the cumulative hinge loss in (6) as

$$L_{cum,\rho_0}[f, S] = \sum_{t=1}^{T} L_{\rho_t}(f_t, (\mathbf{x}_t, y_t)) \tag{11}$$

and for an arbitrary hypothesis g

$$L_{cum,\rho}[g, S] = \sum_{t=1}^{T} L_{\rho}(g, (\mathbf{x}_t, y_t)). \tag{12}$$

3.1. Internal Memory

In the proposed online algorithm, the number of components (size of kernels) is increased when satisfying the following two conditions: (a) the prediction suffers from a loss defined in (6) (step 9 in Algorithm 1); and (b) the minimal distance $d(\mathbf{x}_t, \boldsymbol{\theta}_k) > \epsilon, k = 1, 2, ..., K_t$ (step 15 in Algorithm 1).

In the first case, we can easily compute the size of kernels given the initial margin parameter ρ_0 and the minimal confidence $\rho_{\min}$. When the prediction suffers from a loss, ρ_t

Algorithm 1 The NOCApm Algorithm

Input: $\rho_0 > 0$, $\tau > 0$ and $\epsilon > 0$
Initialization: $\boldsymbol{\theta}_0 = \emptyset$
for $t = 1$ to T **do**
  Receive the instance $\mathbf{x}_t$
  Make prediction $\hat{y} = \text{sign}(f(\mathbf{x}_t))$
  Get label y_t;
  Compute loss, $L_{\rho_t}(f(\mathbf{x}_t), y_t) = \rho_t - y_t f(\mathbf{x}_t)$
  if $L_{\rho_t} \neq 0$ **then**
    $\rho_{t+1} \leftarrow \tau\rho_t, \quad K_{t+1} \leftarrow K_t + 1$
    Add a new Gaussian $\boldsymbol{\theta}_{K_{t+1}}$ to $\boldsymbol{\Theta}$
    Update the hypothesis f_{t+1} by (7)
  else
    Compute the minimal distance to $\boldsymbol{\Theta}$, $d_{\min}$
    if $d_{\min} > \epsilon$ **then**
      $K_{t+1} \leftarrow K_t + 1$
      Add a new Gaussian $\boldsymbol{\theta}_{K_{t+1}}$ to $\boldsymbol{\Theta}$
      Update the hypothesis f_{t+1} by (7)
    else
      Reestimate the one of components, $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$
      Obtain $\boldsymbol{\theta}_i^{'}$
      Update the hypothesis f_{t+1} by (8)
    end if
  end if
end for

is reduced by ratio of τ (cf. (10)). Therefore, the total number of mixture components is $\log_\tau(\rho_{\min}/\rho_0)$.

In the second condition, the increasing number of kernels depends on the distribution of data set. If the data distribute dispersedly in the feature space, the algorithm could have a big size of kernels; otherwise, a small amount of kernels can be obtained in real applications compared to the counterparts, e.g., (Dekel et al., 2006; Orabona et al., 2008), which can be empirically proved in Section 4.

Finally, the total number of kernels $K_{\max}$ is to combine the numbers in both the above conditions.

3.2. Mistake Bounds

If the algorithm satisfies one of the above conditions, the prediction mistake occurs. Therefore, we analyze the mistake bound in individual cases.

Theorem 2 *Suppose f be generated by (7) in the example sequence $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_T, y_T)\}$. Let X be upper bounded for $\boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}}, \boldsymbol{\theta}_{\mathbf{x}}) \leq X^2$, and set the initial margin parameter to ρ_0. Assume that there exists a prediction function g whose cumulative margin loss $L_{cum,\rho'}(g, S) \leq M$ on ρ', and g satisfies $||g||_{\mathcal{H}} \leq B$. Then, the mistake bound for f is:*

$$M_{\rho_0}(f) \leq \frac{B^2 + 2M + \frac{2\rho_0}{1-\tau}}{2\rho' - X^2}. \tag{13}$$

Proof By adding the error indicator parameter σ_t to (7), the updating rule is modified as

$$f_{t+1} = f_t + \sigma_t y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)$$

.

Here, we purposely ignore the reestimation of the mixture model, which will be analyzed in the following. The relative progress can be defined

$$\begin{aligned}
\Delta_t &= ||g - f_t||^2 - ||g - f_{t+1}||^2 \\
&= 2\sigma_t y_t \langle g - f_t, \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)\rangle - \sigma_t^2 \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \boldsymbol{\theta}_{\mathbf{x}_t}) \\
&= 2\sigma_t y_t (g(\mathbf{x}_t) - f_t(\mathbf{x}_t)) - \sigma_t^2 \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \boldsymbol{\theta}_{\mathbf{x}_t}) \\
&\geq 2\sigma_t(\rho' - L_{\rho'}(g, (\mathbf{x}_t, y_t))) - 2\sigma_t(\rho_t - L_{\rho_t}(f_t, (\mathbf{x}_t, y_t))) \\
&- \sigma_t^2 X^2.
\end{aligned} \tag{14}$$

Therefore, the telescopic sum can be computed as:

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t &= ||g - f_0||^2 - ||g - f_T||^2 \\
&\geq 2L_{cum,\rho_0}(f, S) - 2L_{cum,\rho'}(g, S) \\
&\quad - \frac{2\rho_0(1 - \tau^{M_{\rho_0}(f)})}{1 - \tau} + (2\rho' - X^2) M_{\rho_0}(f).
\end{aligned}$$

Since $||g - f_0||^2 - ||g - f_T||^2$ is upper bounded by $||g||^2 \leq B^2$, we have

$$\begin{aligned}
B^2 &\geq 2L_{cum,\rho_0}(f, S) - 2L_{cum,\rho'}(g, S) \\
&\quad - \frac{2\rho_0}{1 - \tau} + (2\rho' - X^2) M_{\rho_0}(f)
\end{aligned}$$

noticing that $\frac{1 - \tau^{M_{\rho_0}(f)}}{1 - \tau} \leq \frac{1}{1 - \tau}$.

Due to $L_{cum,\rho_0}(f, S) > 0$ and $L_{cum,\rho'}(g, S) \leq M$, the mistake number is bounded by

$$M_{\rho_0}(f) \leq \frac{B^2 + 2M + \frac{2\rho_0}{1-\tau}}{2\rho' - X^2}.$$

■

In the proposed approach, the prediction function is updated at each round. If there is no margin error occurred, (8) is used for the updating rule; if there exists a margin error, a new component is added and so (7) is used for updating. For the latter, mistake bound has been analyzed in Theorem 2. To consider both cases, we can derive the mistake bound in the following.

Theorem 3 *Suppose f be generated by (9) in the example sequence $S = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_T, y_T)\}$. Let $\boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}}, \boldsymbol{\theta}_{\mathbf{x}}) \leq X^2$, $\boldsymbol{\kappa}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_i) \leq W^2$ and set the initial margin parameter to ρ_0. Assume that there exists a prediction function g whose cumulative margin loss $L_{cum,\rho'}(g, S) \leq M$ on ρ', and g satisfies $||g||_{\mathcal{H}} \leq B$. And NOCApm generates K_{max} kernels. Then, the mistake bound for f,*

$$M_{\rho_0}(f) \leq \frac{B^2 + 2M + \frac{2\rho_0}{1-\tau}}{2\rho' - X^2} + \frac{2K_{max}W(B + K_{max}W + 2W)}{2\rho' - X^2}.$$

Proof In round t, the updating rule is (9)

$$\begin{aligned} f_{t+1} = f_t &+ \sigma_t y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot) \\ &+ (1 - \sigma_t) \underbrace{(y_t \boldsymbol{\kappa}(\boldsymbol{\theta}'_i, \cdot) - y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_i, \cdot))}_{\delta_t}. \end{aligned}$$

Due to both the updating possibilities, the prediction functions in two successive rounds should be taken into account. Therefore, the relative progress is changed to

$$\begin{aligned} \Delta_t &= ||g - f_t||^2 - ||g - f_{t+1}||^2 \\ &= 2\langle g - f_t, (1 - \sigma_t)\delta_t + \sigma_t y_t \kappa(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)\rangle \\ &\quad - ||(1 - \sigma_t)\delta_t + \sigma_t y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)||^2 \\ &= 2\sigma_t y_t (g(\mathbf{x}_t) - f_t(\mathbf{x}_t)) + 2(1 - \sigma_t)\langle g - f_t, \delta_t\rangle - \\ &\quad ||(1 - \sigma_t)\delta_t + \sigma_t y_t \boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)||^2 \\ &\geq 2\sigma_t(\rho' - L_{\rho'}(g, (\mathbf{x}_t, y_t))) - 2\sigma_t(\rho_t - L_{\rho_t}(f_t, (\mathbf{x}_t, y_t))) \\ &\quad - 2(1 - \sigma_t)||g|| \cdot ||\delta_t|| - 2(1 - \sigma_t)\langle f_t, \delta_t\rangle \\ &\quad - (1 - \sigma_t)||\delta_t||^2 - \sigma_t||\boldsymbol{\kappa}(\boldsymbol{\theta}_{\mathbf{x}_t}, \cdot)||^2. \end{aligned}$$

The inner product between f_t and δ_t can be upper bounded by

$$\begin{aligned} \langle f_t, \delta_t\rangle &\leq \left(\sum_{j=1}^{K_t} \boldsymbol{\kappa}(\boldsymbol{\theta}_j, \boldsymbol{\theta}_i) + \sum_{j=1}^{K_t} \boldsymbol{\kappa}(\boldsymbol{\theta}_j, \boldsymbol{\theta}'_i)\right) \\ &\leq 2K_t W^2. \end{aligned}$$

Therefore, the process in two successive rounds can be lower bounded by:

$$\begin{aligned} \Delta_t &\geq 2\sigma_t(\rho' - L_{\rho'}(g, (\mathbf{x}_t, y_t))) - 2\sigma_t(\rho_t - L_{\rho_t}(f_t, (\mathbf{x}_t, y_t))) \\ &\quad - 2(1 - \sigma_t)||g|| \cdot ||\delta_t|| - 4(1 - \sigma_t)K_t W^2 \\ &\quad - 2(1 - \sigma_t)W^2 - \sigma_t X^2 \\ &\geq 2\sigma_t(\rho' - L_{\rho'}(g, (\mathbf{x}_t, y_t))) - 2\sigma_t(\rho_t - L_{\rho_t}(f_t, (\mathbf{x}_t, y_t))) \\ &\quad - 2(1 - \sigma_t)(BW + 2K_t W^2 + W^2) - \sigma_t X^2. \end{aligned}$$

Compared to the process (14) by only considering a margin loss, there are three additional terms $-(BW + 2K_t W^2 + W^2)$ by considering both the cases for updating. The prediction

Table 1: Average accuracies with standard deviation, the size of working set and the training times provided by the NOCApm, the Forgetron and the Projectron++ algorithms for synthetic data set.

Algorithm	Accuracy(%)	#Set	Training times(s)
NOCApm	98.95 ± 0.34	31.6 ± 8.6	1.901 ± 0.396
Forgetron	98.33 ± 0.13	167.4 ± 12.5	1.872 ± 0.068
Projectron++	98.65 ± 0.10	20.7 ± 2.16	2.075 ± 0.088

mistakes occur at most $K_{\max}$ times. Therefore, the telescopic sum by considering both the updating conditions can be lower bounded by

$$\begin{aligned}\sum_{t=1}^{T} \Delta_t \geq\ & 2L_{cum,\rho_0}(f,S) - 2L_{cum,\rho'}(g,S) \\ & - \frac{2\rho_0}{1-\tau} + (2\rho' - X^2)M_{\rho_0}(f) \\ & - 2K_{max}W(B + K_{max}W + 2W)\end{aligned}$$

with the upper bound

$$\sum_{t=1}^{T} \Delta_t = ||g - f_0||^2 - ||g - f_T||^2 \leq B^2.$$

Therefore, we can derive the mistake bound,

$$M_{\rho_0}(f) \leq \frac{B^2 + 2M + \frac{2\rho_0}{1-\tau}}{2\rho' - X^2} + \frac{2K_{max}W(B + K_{max}W + 2W)}{2\rho' - X^2}.$$

■

The mistake bound in Theorem 3 has one additional term (the second term in the right-hand side of the expression above) compared to the bound proposed in (Kivinen et al., 2004). Here, $K_{\max}W$ reflects the range of data distribution. NOCApm will control the quantity of $K_{\max}$ in low dimension sequence data, however, the algorithm is prone to being influenced by the curse of dimensionality of data set with respect to the other online algorithms with Budget, i.e., the Forgetron (Dekel et al., 2006) and the Projectron (Orabona et al., 2008).

4. Experiments

In this section, we compare the NOCApm to the recently proposed kernel-based online algorithms: the Projectron (Orabona et al., 2008) and the Forgetron (Dekel et al., 2006). We select the Projectron++ for the comparison as it obtained the best result reported in (Orabona et al., 2008).

Table 2: Average accuracies with standard deviation, the size of working set and the training times provided by the NOCApm, the Forgetron and the Projectron++ algorithms for Adult data set.

Algorithm	Accuracy(%)	#Set	Training times(s)
NOCApm	80.78 ± 0.41	58.2 ± 4.30	11.40 ± 1.77
Forgetron	76.25 ± 0.23	2000	42.78 ± 0.85
Projectron++	79.88 ± 0.13	740.2 ± 5.89	35.09 ± 2.32

The algorithms are tested on one synthetic data set described in (Orabona et al., 2008) and two large-scale real benchmark machine learning data sets:

`Adult` [1]: 1994 Census database, where there are $30,162$ training samples and $15,060$ test samples with 14 features, and the percentage of the positive samples is 24.78%. It is a binary classification problem with the prediction task for determining whether a person makes over $ 50K a year.

`Vehicle` [2]: it contains 78,823 training samples and 19,705 test examples with 50 attributes. It is a multiple classification problem with 3 classes.

4.1. Experimental setting

All the experiments were conducted over 5 different permutations of the training data sets. In the following experiments, average accuracies, average size of the working set, average training times are reported. All the experiments were conducted on a 3.0GHz CPU with the MATLAB implementations.

For both the Projectron++ and the Forgetron, two parameters should be firstly defined, i.e., budget size, B and kernel parameters. Here, we follow the same setting as (Orabona et al., 2008), and so a Radial Basis Function (RBF) kernel is used with the best results Gaussian width $\sigma = 1.0$ for the synthetic data set, $\sigma = \sqrt{2}$ for Adult dataset and $\sigma = 2$ for Vehicle dataset. The budget size is set to $B = 1000$ for the synthetic data set, $B = 2000$ for Adult data set and $B = 4000$ for Vehicle data set.

The selection of parameters ρ_0, τ and ϵ for NOCApm follows the rules: ρ_0 is defined on a probability margin, and unrelated to the scale of various data; τ is used to control the degeneration of margin parameter ρ_0, thus $\tau = 0.9$ is enough; ϵ measures the distance in feature space, which is affected by the scale of data. Based on these points above, the parameters used in the following tables and figure are set as follows: in Table 1 $\epsilon = 1.0$; in Table 2 $\epsilon = 0.39$; and in Table 3 and Fig. 2 $\epsilon = 0.35$, respectively and $\rho_0 = 0.10, \tau = 0.9$ for all the experimental results.

Table 3: Average accuracies with standard deviation, the size of working set and the training times provided by the NOCApm, the Forgetron and the Projectron++ algorithms for Vehicle data set.

Algorithm	Accuracy(%)	#Set	Training times(s)
NOCApm	82.98 ± 0.22	383.87 ± 3.71	124.49 ± 3.34
Forgetron	75.24 ± 0.34	4000	565.36 ± 8.78
Projectron++	80.01 ± 0.18	1494 ± 9.41	451.70 ± 7.25

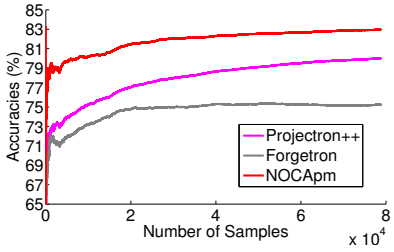

Figure 2: Average accuracies with respect to the size of the observed examples for Vehicle data.

4.2. Experimental results

The average classification accuracies with standard deviation are reported in Table 1 for synthesis data set, in Table 2 for Adult data set and in Table 3 for Vehicle data set, respectively. Also, the corresponding size of the working set and the training times are reported in the same tables with comparison to those provided by the Forgetron and the Projectron++. On the analysis of Table 2 and Table 3, one can see that the NOCApm obtains not only the best classification accuracies and also far less computational times in both binary (i.e., Adult data set) and multiple (Vehicle data set) classification problems.

Besides, we show the average classification accuracies for Vehicle data set with respect to the number of observed examples using the NOCApm, the Forgetron and the Projectron++ algorithms in Fig. 2. The proposed approach smoothly obtains the best classification accuracies with the increase of the size of observed examples.

5. Conclusion and Discussion

In the paper, we propose a kernel-based online classification algorithm which takes advantage of both generative models and discriminative approaches. In online learning, the

1. Available at: http://archive.ics.uci.edu/ml/datasets/Adult.
2. Available at: http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass.html.

statistical modeling of data points with the Gaussian mixture model is adopted for the definition of hypothesis on each round in a kernel manner. Unlike the voted perceptron (Freund and Schapire, 1999), it is not necessary to store all the misclassified observed examples for computing kernel values. Unlike the fixed budget algorithms by forgetting the observed examples out of budget window size, such as (Dekel et al., 2006; Cavallanti et al., 2007), the information of the observed examples for the proposed algorithm is captured and stored in the statistical modeling and thus it is not necessary to discard the observed examples due to the memory problem. Additionally, the size of kernels can be controlled by the parameters of the proposed approach. The mistake bounds are derived in terms of generative models. Experiments carried out on one synthesis and two real large-scale data sets validate the effectiveness of the proposed approach, where the classification accuracies of our algorithm is superior to the Forgetron and the Projectron on the same problems, while saving significantly on spatial and computational complexities.

In the NOCApm algorithm, data are represented by the Gaussian mixture model. This can suffer from the curse of dimensionality problem. The future development will apply a dimensionality reduction technique for statistical modeling.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported in part by Natural Science Foundation of China (No. 61075057), Municipal R&D Foundation of Baoshan district, Shanghai (No. 10511500703), and Shanghai Municipal Committee for the Education (No. 11CXY01).

References

A. Aizerman, E. M. Braverman, and L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25: 821–837, 1964.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Tracking the best hyperplane with a simple budget perceptron. *Mach. Learn.*, 69(2-3):143–167, 2007.

L. Cheng, S. V. N. Vishwanathan, D. Schuurmans, S. Wang, and T. Caelli. Implicit online learning with kernels. In *Advances in Neural Information Processing Systems*, volume 19, Cambridge MA, 2006. MIT Press.

K. Crammer, J. Kandola, and Y. Singer. Online classification on a budget. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, March 2006.

O. Dekel, S. Shalev-Shwartz, and Y. Singer. The forgetron: A kernel-based perceptron on a fixed budget. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 259–266. MIT Press, Cambridge, MA, 2006.

Y. Freund and R.E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.

T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, August 2004.

Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1-3):361–387, 2002.

F. Orabona, J. Keshet, and B. Caputo. The projectron: a bounded kernel-based perceptron. Helsinki, Finland, July 2008. In Proc. of International Conference on Machine Learning (ICML08).