Advances in Applied Mathematics 应用数学进展, 2023, 12(9), 4066-4075
Published Online September 2023 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2023.129399

Hans汉斯

基于投影策略的改进在线非线性分类算法

孟　依，宋爱民*

大连交通大学理学院，辽宁　大连

收稿日期：2023年8月14日；录用日期：2023年9月8日；发布日期：2023年9月15日

摘　要

在线学习自提出以来已经发展出许多算法及其改进，根据模型是线性还是非线性，将在线学习算法分为两大类：在线线性学习算法和基于核的在线学习算法。核方法是指利用特征映射，将样本映射到高维再生核希尔伯特空间，从而将线性不可分问题转化成线性可分问题。并行投影算法是用自适应投影次梯度方法(APSM)推导的，其优点是收敛速度快，但该算法的稀疏方式是当模型的规模达到一定容量时，可能将最远端某些重要的数据点删除，从而导致算法性能严重退化；以及存在偏移量使得分类效果变差。为了改进以上问题，本文是在并行投影算法的基础上提出了一种新的算法，其稀疏方式是利用最新的数据向字典中的数据做投影，从而在保证误分类率低时，本文算法的字典规模较小。仿真数据和真实数据的实验结果表明，与其他七种经典在线分类算法相比，本文算法在字典规模较小时，误分类率最低。

关键词

在线学习，并行投影，核方法，稀疏

Improved Online Nonlinear Classification Algorithm Based on Projection Strategy

Yi Meng, Aimin Song*

School of Science, Dalian Jiaotong University, Dalian Liaoning

Received: Aug. 14th, 2023; accepted: Sep. 8th, 2023; published: Sep. 15th, 2023

Abstract

Online learning has developed many algorithms and their improvements since it was proposed. According to whether the model is linear or nonlinear, online learning algorithms are divided into two categories: online linear learning algorithms and kernel-based online learning algorithms. The kernel method is to map the sample to a high-dimensional regenerated kernel Hilbert space by us-

*通讯作者。

文章引用: 孟依, 宋爱民. 基于投影策略的改进在线非线性分类算法[J]. 应用数学进展, 2023, 12(9): 4066-4075.
DOI: 10.12677/aam.2023.129399

ing feature mapping, so that the linear indivisible problem can be transformed into a linear separable problem. The parallel projection algorithm is derived by the adaptive projection subgradient method (APSM), which has the advantage of fast convergence, but the sparse way of the algorithm is that when the scale of the model reaches a certain capacity, some important data points at the farthest end may be deleted, resulting in serious degradation of the algorithm performance. And the existence of offset makes the classification effect worse. In order to improve the above problems, this paper proposes a new algorithm based on the parallel projection algorithm. Its sparse method is to use the latest data to project the data in the dictionary, so that the dictionary size of the algorithm in this paper is small when the classification error rate is low. The experimental results of simulation data and real data show that compared with other online algorithms, the proposed algorithm has the lowest classification error rate when the dictionary size is small.

Keywords

Online Learning, Parallel Projection, Kernel Method, Sparse

Copyright © 2023 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

机器学习算法分为批处理算法和在线学习算法，批处理算法可以一次性批量输入给学习算法，并在小规模数据上取得了巨大成功，但当数据规模较大时，批处理算法的计算复杂度高，并且不适合动态变化的环境。与批处理算法相比，在线学习算法适用于大规模数据和动态环境，并且能够降低学习算法的空间复杂度和时间复杂度。在大数据时代，数据高速增长的特点为机器学习带来了更加严峻的挑战，而在线学习算法可以有效地解决该问题，因此引起了人们的广泛关注。

设计良好的在线学习算法只需较少的计算，就能达到与批处理算法相同的效果。在线学习算法可以分为在线线性学习算法和基于核的在线学习算法，当样本线性可分时，在线线性学习算法可以有效分类，而在基于核的在线学习算法中，由于样本线性不可分，常需要利用特征映射，将样本映射到高维再生核希尔伯特空间，从而将线性不可分问题转化成线性可分问题，来降低分类任务的难度。在过去几年中，开发了很多基于核的在线学习算法，并在很多问题上也展现了很好的性能。

1958 年，Rosenblatt 首次提出了感知机(Perceptron)算法，并证明了对于线性可分问题，该算法总是收敛的[1]。文献[2]基于核的感知机算法，其主要思想是通过特征映射将非线性分类问题转化成特征空间的线性问题，并在特征空间应用感知机算法求解。2004 年，J.Kivinen 等人提出了核最小均方算法[3]。Engel 等人提出了核递归最小二乘算法，由 Mercer 核诱导的高维特征空间中执行线性回归，因此可以用于递归地构造非线性最小二乘问题的最小均方误差解，并且首次将核方法扩展到自适应滤波[4]。2004 年，Isao Yamada 和 Nobuhiko Ogura 首次提出自适应投影次梯度方法，该算法是把分类问题转化为一个凸优化问题[5]。2008 年，Konstantinos Slavakis 等人引入了自适应投影次梯度法，利用基于投影的自适应滤波工具，导出一种新的用于再生核希尔伯特空间(RKHS)分类的在线算法[6]。

本文是在文献[6]中并行投影算法的基础上进行改进，而并行投影算法具有收敛速度快的优点，但是也存在缺点。第一，由于并行投影算法存在偏移量，会导致分类效果差；第二，并行投影算法的稀疏方式是当样本容量小于预设的预算时，数据点将会进入字典；当样本容量大于预设的预算时，将最远端的数据点删除，而该数据点可能是最重要的数据点，从而导致算法性能严重退化。

基于核的在线学习算法面临的问题是随着训练次数的增加，导致模型的规模随之增大，需要的计算量和存储空间也越来越多，为了改善以上缺点，需要对模型进行稀疏处理。在算法中加入稀疏处理，能够有效地改变字典规模的大小。因此，本文提出了新的算法：基于投影策略的改进在线非线性分类算法，本文的算法在误分类率(misclassification rate)和字典规模方面都具有一定的优势。

为了更清楚地表述，本文用 Z，$Z_{\geq 0}$，$Z_{>0}$，$\mathbb{R}$ 表示所有整数，非负整数，正整数和实数。此外，对于任意整数 $j_1 \leq j_2$，定义 $\overline{j_1, j_2} := \{j_1, j_1+1, \cdots, j_2\}$。

2. 问题描述

考虑一个在线二元分类问题，假设 $(\boldsymbol{x}_t)_{t\in Z_{\geq 0}} \in \mathcal{X}$ 为输入向量序列，其中 $\mathcal{X} \in \mathbb{R}^m$ 为输入向量集合。用给出的数据对 $(\boldsymbol{x}_t, y_t)$，训练一个非线性分类器 $f: \mathbb{R}^d \to \mathbb{R}$，其中 t 表示第 t 轮，d 是维数，$y_t \in \{+1, -1\}$ 是真实的类别标签。

对于在线分类问题，可以通过最小化软边际损失函数 $l(\cdot): \mathcal{H} \to \mathbb{R}$ 来训练分类器 $f \in \mathcal{H}$：

$$l((\boldsymbol{x}_t, y_t); f) = \max(0, \rho - y_t f(\boldsymbol{x}_t)),$$

其中 ρ 为边缘参数。

在线非线性的分类算法在特征空间 $\mathcal{H}$ 中形成分类器(点)的形式为

$$f := \sum_{t=0}^{\infty} \alpha_t k(\boldsymbol{x}_t, \cdot) \in \mathcal{H}.$$

基于核的在线学习算法的目的是学习关于核的预测模型 $f(\boldsymbol{x})$，用于如下分类新数据点 $\boldsymbol{x} \in \mathbb{R}^d$：

$$f(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t k(\boldsymbol{x}_t, \boldsymbol{x}),$$

其中 T 为处理数据点的数量，α_t 表示第 t 个数据点的系数，$k(\cdot,\cdot)$ 表示核函数。

本文的目的是在并行投影算法的基础上进行更新，下面小节将给出本文算法。

3. 本文算法

在本文中，主要讨论了基于投影策略的改进在线非线性分类算法。第一部分介绍了基于并行投影的在线分类算法的推导，第二部分介绍了算法的稀疏。

3.1. 基于并行投影的在线分类算法的推导

本节主要利用文献[6]中并行投影算法的推导，下面将简单的介绍一下并行投影算法的推导过程。

$\forall t \in Z_{>0}$，每个点的索引集 $J_t \subset \overline{0,t}$，对于 $j \in J_t$ 有闭半空间

$$\Pi_{j,t}^{+} = \left\{ \hat{f} \in \mathcal{H} \times \mathbb{R} : y_j \hat{f}(x_j) \geq \rho_j \right\}.$$

给定分类器 $u_t \in \mathcal{H}$，任意函数 $u \in \mathcal{H}$，定义函数列为

$$\Theta_t(u) = \begin{cases} \sum_{j\in J_t} \dfrac{w_j d(u_t, \Pi_{j,t}^{+})}{L_t} d(u, \Pi_{j,t}^{+}), & L_t \neq 0 \\ 0, & L_t = 0 \end{cases},$$

其中 $L_t := \sum_{j\in J_t} w_j d(u_t, \Pi_{j,t}^{+})$，权重 $w_j \geq 0$，$\forall j \in J_t$ 且 $\sum_{j\in J_t} w_j = 1$。

根据距离公式 $d(f,c) = \| f - P_c(f) \|$，其中点 $f \in \mathcal{H}$，闭凸集 $c \subset \mathcal{H}$，$P_c(f)$ 表示 f 向 c 投影，可知

$$d\left(u,\Pi_{j,t}^{+}\right)=\left\|u-P_{\Pi_{j,t}^{+}}\left(u\right)\right\|.$$

把 u_t 代入 $\Theta_t\left(u\right)$ 中，可得

$$\Theta_t\left(u_t\right)=\frac{1}{L_t}\sum_{j\in J_t}w_j d^2\left(u_t,\Pi_{j,t}^{+}\right).$$

该函数 $\left\{d\left(\cdot,\Pi_{j,t}^{+}\right)\right\}_{j\in J_t}$ 是闭凸集的度量距离函数，是连续的和次可微的。

对 $d\left(u,\Pi_{j,t}^{+}\right)$ 求次梯度，可知

$$\partial d\left(u,\Pi_{j,t}^{+}\right)=\left\{\frac{u-P_{\Pi_{j,t}^{+}}\left(u\right)}{d\left(u,\Pi_{j,t}^{+}\right)}\right\}=\frac{u-P_{\Pi_{j,t}^{+}}\left(u\right)}{\left\|u-P_{\Pi_{j,t}^{+}}\left(u\right)\right\|}.$$

对 $\Theta_t\left(u_t\right)$ 求导知

$$\Theta_t'\left(u_t\right)=\frac{1}{L_t}\sum_{j\in J_t}w_j d\left(u_t,\Pi_{j,t}^{+}\right)\frac{u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)}{\left\|u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right\|}.$$

因为 $d\left(u_t,\Pi_{j,t}^{+}\right)=\left\|u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right\|$，所以有

$$\Theta_t'\left(u_t\right)=\frac{1}{L_t}\sum_{j\in J_t}w_j\left(u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right), \tag{1}$$

由 APSM 算法的迭代公式可知

$$u_{t+1}=u_t-\lambda_t\frac{\Theta_t\left(u_t\right)}{\left\|\Theta_t'\left(u_t\right)\right\|^2}\Theta_t'\left(u_t\right).$$

将(1)式代入 APSM 算法的迭代公式中，可知

$$u_{t+1}=u_t-\lambda_t\frac{\sum_{j\in J_t}w_j d^2\left(u_t,\Pi_{j,t}^{+}\right)}{\left\|\sum_{j\in J_t}w_j\left(u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right)\right\|^2}\sum_{j\in J_t}w_j\left(u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right).$$

令

$$\mu_t=\lambda_t\frac{\sum_{j\in J_t}w_j d^2\left(u_t,\Pi_{j,t}^{+}\right)}{\left\|\sum_{j\in J_t}w_j\left(u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right)\right\|^2},$$

故

$$u_{t+1}=u_t-\mu_t\sum_{j\in J_t}w_j\left(u_t-P_{\Pi_{j,t}^{+}}\left(u_t\right)\right). \tag{2}$$

令

$$\boldsymbol{\beta}_j=w_j\cdot y_j\left(\rho-y_j f_t\left(\boldsymbol{x}_j\right)\right)^{+}.$$

则算法过程中(2)式可以等价地写为

$$f_{t+1}=f_t+\mu_t\sum_{j\in J_t}\beta_j k\left(\boldsymbol{x}_j,\cdot\right). \tag{3}$$

3.2. 算法的稀疏

随着训练次数的增加，会导致利用(3)式更新的算法中的字典规模越来越大，因此人们引入了各种在线稀疏方法，例如滑动窗技术[6]、贪婪法[7]、随机法[8]、量化法[9]等等。在本节中，当最新的数据可以利用字典中的元素近似表示时，最新的数据将不进入字典，此时需要用最新的数据向字典中的数据做投影，利用求矩阵的逆的方法进行更新；否则，最新的数据将进入字典。

设$\left(\boldsymbol{x}_t\right)_{t\in z_{\geq 0}}\in\mathcal{X}$为输入向量序列，其中$\mathcal{X}\in\mathbb{R}^m$为输入向量集合，$y_t\in\{+1,-1\}$为标签。在$t-1$时刻，字典为$D_{t-1}=\left\{\tilde{\boldsymbol{x}}_j\right\}_{j=1}^{m_{t-1}}$。当获得最新数据$\boldsymbol{x}_t$时，本文算法需要判断是否满足近似线性相关(ALD)条件

$$\delta_t=\min_{\boldsymbol{a}}\left\|\sum_{j=1}^{m_{t-1}}a_j\boldsymbol{\Phi}\left(\tilde{\boldsymbol{x}}_j\right)-\boldsymbol{\Phi}\left(\boldsymbol{x}_t\right)\right\|^2\leq\eta,$$

即将最新数据$\boldsymbol{\Phi}\left(\boldsymbol{x}_t\right)$向字典$D_{t-1}$做投影，其中投影系数为

$$\boldsymbol{a}_t=\tilde{\boldsymbol{K}}_{t-1}^{-1}\tilde{\boldsymbol{k}}_{t-1}\left(\boldsymbol{x}_t\right).\tag{4}$$

若最新数据$\boldsymbol{x}_t$满足 ALD 条件，即投影误差$\|\delta_t\|$小于给定的阈值η，因此可以利用字典中已有的元素近似表示，所以最新的数据将不进入字典；若最新数据$\boldsymbol{x}_t$不满足 ALD 条件，即投影误差$\|\delta_t\|$大于给定的阈值η，因此最新数据不能利用字典中的元素近似表示，会导致很大的误差，所以最新数据进入字典。

无论数据是否进入字典，都需要将数据向字典做投影，从而得到每个投影系数，将所有展开的投影系数组成系数矩阵，本节的系数矩阵是由最新的q个数据向字典做投影，可以得到最新q个数据展开的投影系数，然后将其组成系数矩阵，用$\boldsymbol{A}_t$表示

$$\boldsymbol{A}_t=\begin{bmatrix}\boldsymbol{a}_{t-q+1}^{\mathrm{T}}\\ \boldsymbol{a}_{t-q+2}^{\mathrm{T}}\\ \vdots\\ \boldsymbol{a}_t^{\mathrm{T}}\end{bmatrix}.$$

在下一次迭代时，因为不再考虑$k\left(\boldsymbol{x}_{t-q+1},\cdot\right)$，所以不再考虑$k\left(\boldsymbol{x}_{t-q+1},\cdot\right)$在字典$D_{t-1}$上的投影，因此引入矩阵

$$\bar{\boldsymbol{A}}_{t-1}=\begin{bmatrix}\boldsymbol{a}_{t-q+2}^{\mathrm{T}}\\ \vdots\\ \boldsymbol{a}_{t-1}^{\mathrm{T}}\end{bmatrix}.$$

故(3)式中$\sum_{j\in J_t}\beta_jk\left(\boldsymbol{x}_j,\cdot\right)$可写成如下形式

$$\sum_{j\in J_t}\beta_jk\left(\boldsymbol{x}_j,\cdot\right)\approx\left(\beta_{t-q+1},\beta_{t-q+2},\cdots,\beta_t\right)\boldsymbol{A}_t\begin{pmatrix}k\left(\boldsymbol{x}_{t-q+1},\cdot\right)\\ k\left(\boldsymbol{x}_{t-q+2},\cdot\right)\\ \vdots\\ k\left(\boldsymbol{x}_t,\cdot\right)\end{pmatrix}.$$

下面将分情况讨论最新数据是否进入字典，投影系数、字典和字典维数分别是如何更新的。

当$\|\delta_t\|>\eta$时，数据$k\left(\boldsymbol{x}_t,\cdot\right)$不能利用字典中的元素所表示，会导致很大的误差，因此数据$k\left(\boldsymbol{x}_t,\cdot\right)$要向自己做投影，可以得到其投影系数为$\boldsymbol{a}_t=\left(0,0,\cdots,1\right)$，所以矩阵$\boldsymbol{A}_t$的更新为$\boldsymbol{A}_t=\begin{bmatrix}\bar{\boldsymbol{A}}_{t-1} & 0\\ 0 & 1\end{bmatrix}$，系数更新为

$\Upsilon_t' = \begin{bmatrix} \Upsilon_{t-1} & 0 \end{bmatrix}$，$\Upsilon_t = \Upsilon_t' + \boldsymbol{\beta}_t \boldsymbol{A}_t$。此时，最新数据将进入字典，字典的更新为 $D_t = D_{t-1} \cup \{\boldsymbol{x}_t\}$，字典维数更新为 $m_t = m_{t-1} + 1$。

当 $\|\delta_t\| \le \eta$ 时，数据 $k(\boldsymbol{x}_t, \cdot)$ 可以利用已有字典中的元素表示，因此数据 $k(\boldsymbol{x}_t, \cdot)$ 要向字典 D_{t-1} 做投影，得到其投影系数为 $\boldsymbol{a}_t = \tilde{\boldsymbol{K}}_{t-1}^{-1} \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)$，所以矩阵 $\boldsymbol{A}_t$ 的更新为 $\boldsymbol{A}_t = \begin{bmatrix} \bar{\boldsymbol{A}}_{t-1} \\ \boldsymbol{a}_t^{\mathrm{T}} \end{bmatrix}$，系数更新为 $\Upsilon_t' = \Upsilon_{t-1}$，$\Upsilon_t = \Upsilon_t' + \boldsymbol{\beta}_t \boldsymbol{A}_t$。此时，最新数据不会进入字典，字典不变，字典更新为 $D_t = D_{t-1}$，字典维数更新为 $m_t = m_{t-1}$。

综上，引入稀疏化环节后，本文的更新公式为

$$f_t(\cdot) = \sum_{j=1}^{m_t} \Upsilon_t(j) k(\boldsymbol{x}_j, \cdot).$$

至此，本文得到了新算法，新算法的伪代码如下。

基于投影策略的改进在线非线性分类算法

初始化：$f_0 = 0$

输入：数据对 $\{\boldsymbol{x}_t, y_t\}$，损失函数 $l_t = \max\{0, \rho - y_t f(\boldsymbol{x}_t)\}$，权重 w_j，$\sum_{j \in J_t} w_j = 1$，阈值 η，并行投影个数 q

for $t = 1, \cdots, T$

  if $t \le q$

$\boldsymbol{\beta}_t = w_t \cdot y_t (\rho - y_t f(\boldsymbol{x}_t))^+$

$\boldsymbol{A}_t = \begin{bmatrix} \bar{\boldsymbol{A}}_{t-1} & 0 \\ 0 & 1 \end{bmatrix}$，其中 $\bar{\boldsymbol{A}}_{t-1} = \boldsymbol{A}_{t-1}$

系数更新：$\Upsilon_t' = \begin{bmatrix} \Upsilon_{t-1} & 0 \end{bmatrix}$ $\Upsilon_t = \Upsilon_t' + \boldsymbol{\beta}_t \boldsymbol{A}_t$

维数更新：$m_t = m_{t-1} + 1$

  end

  if $t > q$

$\boldsymbol{\beta}_t = w_t \cdot y_t (\rho - y_t f(\boldsymbol{x}_t))^+$

if $\|\delta_t\| > \eta$

$\boldsymbol{A}_t = \begin{bmatrix} \bar{\boldsymbol{A}}_{t-1} & 0 \\ 0 & 1 \end{bmatrix}$，其中 $\bar{\boldsymbol{A}}_{t-1} = \begin{bmatrix} \boldsymbol{a}_{t-q+2}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_{t-1}^{\mathrm{T}} \end{bmatrix}$

系数更新：$\Upsilon_t' = \begin{bmatrix} \Upsilon_{t-1} & 0 \end{bmatrix}$ $\Upsilon_t = \Upsilon_t' + \boldsymbol{\beta}_t \boldsymbol{A}_t$

维数更新：$m_t = m_{t-1} + 1$

else

计算矩阵 $\tilde{\boldsymbol{K}}_t^{-1}$ 的更新

$\boldsymbol{A}_t = \begin{bmatrix} \bar{\boldsymbol{A}}_{t-1} \\ \boldsymbol{a}_t^{\mathrm{T}} \end{bmatrix}$，其中 $\boldsymbol{a}_t = \tilde{\boldsymbol{K}}_{t-1}^{-1} \tilde{\boldsymbol{k}}_{t-1}(\boldsymbol{x}_t)$，$\bar{\boldsymbol{A}}_{t-1} = \begin{bmatrix} \boldsymbol{a}_{t-q+2}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_{t-1}^{\mathrm{T}} \end{bmatrix}$，

系数更新：$\Upsilon_t' = \Upsilon_{t-1}$ $\Upsilon_t = \Upsilon_t' + \boldsymbol{\beta}_t \boldsymbol{A}_t$

维数更新：$m_t = m_{t-1}$

end

  end

end

输出：$f_t(\cdot) = \sum_{j \in J_t} \Upsilon_t(j) k(\boldsymbol{x}_j, \cdot)$

4. 实验结果

在本节中，新算法与其他不同算法分别在仿真数据和真实数据下进行性能比较，并将算法参数的设置以表格的形式展示，各个算法的误分类率和字典维数以图的形式展示。由于误分类率可以作为衡量不同算法性能的重要指标，在下面的仿真数据实验中，利用测试集上的误分类率来衡量不同算法的性能，其定义式如下：

$$\text{误分类率} = \frac{\sum_i I\left(y_i \neq \hat{y}_i\right)}{n},$$

上式中 n 表示测试数据集的长度，y_i 表示 i 时刻的期望输出，$\hat{y}_i$ 表示 i 时刻对测试数据的实际输出。

4.1. 仿真数据比较

本文使用的仿真数据与文献[6]相同，其中输入数据对 $(\boldsymbol{x}_t, y_t)$，$\boldsymbol{x}_t \in \mathbb{R}^4$，$y_t \in \{+1,-1\}$。为了算法比较的公平性，本文算法与 APSM 算法[6]，SPA 算法[10]，Perceptron 算法[2]，OLRU 算法[11]，OLRD 算法[11]，Projectron 算法[12]，Projectron++算法[12]等算法进行性能对比，对比的算法都具有良好的鲁棒性，具体的参数选择在下面表 1 中说明。

Table 1. Parameter setting of each algorithm
表 1. 各个算法的参数设置

算法	参数
APSM	$\sigma^2 = 0.5 \quad \delta = 5 \quad q = 16 \quad L = 800$
SPA	$\sigma^2 = 0.5$
Perceptron	$\sigma^2 = 0.5 \quad \eta = 1$
OLRU	$\sigma^2 = 0.5 \quad \eta = 0.9 \quad c = 2 \quad \alpha = 0.2$
OLRD	$\sigma^2 = 0.5 \quad \eta = 0.9 \quad c = 2$
Projectron	$\sigma^2 = 0.5 \quad \eta_t = 0.1$
Projectron++	$\sigma^2 = 0.5 \quad \eta_t = 0.1$
Proposed	$\sigma^2 = 0.5 \quad \eta_t = 0.5 \quad q = 16$

根据上述参数的设置，本文算法与各个算法进行比较，下图展示各个算法的误分类率及字典维数 D。图 1 显示，本文提出的算法比其他算法相比，在字典规模较小时，本文提出的算法的误分类率最低。

4.2. 真实数据比较

真实数据集是来自 LIBSVM 数据存储库的二分类数据集，其中 LIBSVM 是一个支持向量机的库，它由台湾大学(Taiwan University)的林智仁(Chih-Jen Lin) 教授和他的团队所开发的。该库提供了一种简单且高效的方式，用于解决回归(regression)、分类(classification)和分布估计(distribution estimation)等问题。

下面在真实数据 a3a 以及 a7a 的各个算法进行性能比较，其中两组数据的参数相同，都用了以下表 2 参数的设置。为了算法比较的公平性，选择对比的算法都具有良好的鲁棒性，具体的参数选择在下面表 2

中说明。

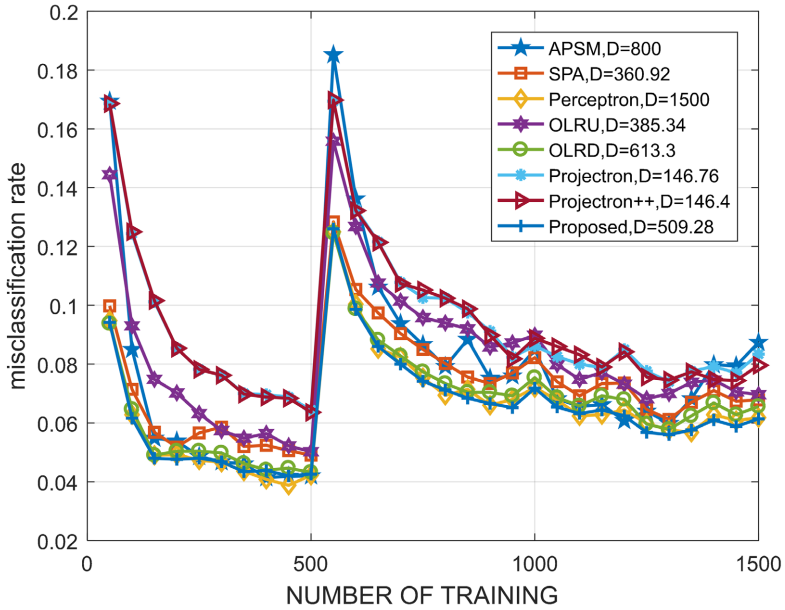

Figure 1. The performance of the proposed algorithm is compared with other algorithms under simulation data
图 1. 本文算法和其他算法在仿真数据下的性能比较

Table 2. Parameter setting of each algorithm
表 2. 各个算法的参数设置

算法	参数
APSM	$\sigma^2=1 \quad \delta=5 \quad q=16 \quad L=800$
SPA	$\sigma^2=1$
Perceptron	$\sigma^2=1 \quad \eta=1$
OLRU	$\sigma^2=1 \quad \eta=0.5 \quad c=2 \quad \alpha=0.2$
OLRD	$\sigma^2=1 \quad \eta=0.9 \quad c=2$
Projectron	$\sigma^2=1 \quad \eta_t=0.01$
Projectron++	$\sigma^2=1 \quad \eta_t=0.01$
Proposed	$\sigma^2=1 \quad \eta_t=0.99 \quad q=16$

根据上述参数的设置，将各个算法与本文提出的算法分别在 a3a 以及 a7a 数据中，进行了比较，下图给出各个算法的误分类率以及字典维数 D。

图 2 运用的数据是 a3a，测试数据为 20,000 个，训练数据为 3000 个，核长均为 1。在 a3a 数据中，本文提出的算法与其他算法相比，新算法在字典规模较小时，误分类率最低。

图 3 运用的数据是 a7a，测试数据为 16,000 个，训练数据为 10,000 个，核长均为 1。在 a7a 数据中，这八种算法是在期望的参数下进行的对比，本文提出的算法与其他算法相比，新算法在字典规模较小时，误分类率最低。

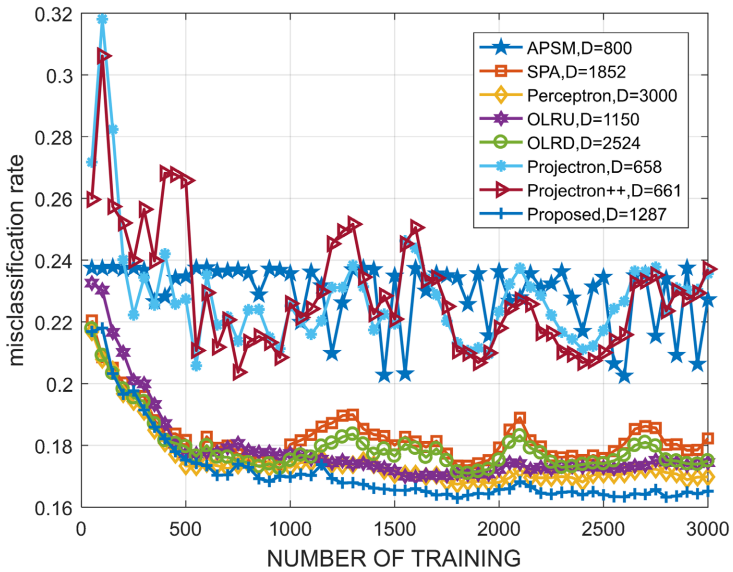

Figure 2. Performance comparison of the proposed algorithm and other algorithms under a3a data
图 2. 本文算法和其他算法在 a3a 数据下的性能比较

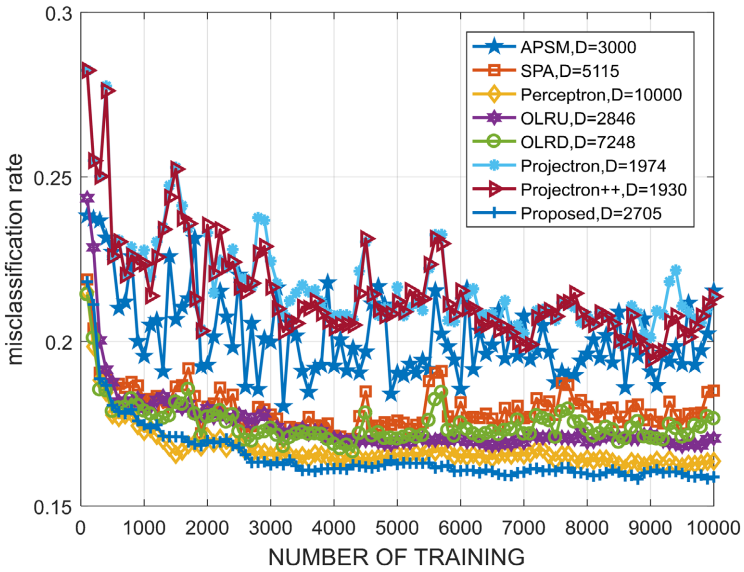

Figure 3. Performance comparison of the proposed algorithm and other algorithms under a7a data
图 3. 本文算法和其他算法在 a7a 数据下的性能比较

5. 总结

基于核的在线学习算法面临的问题是随着训练次数的增加，导致模型的规模随之增大，需要的计算量和存储空间也越来越多，为了改善以上缺点，需要对模型进行稀疏处理。本文是在并行投影算法的基础上提出了一种新的算法为基于投影策略的改进在线非线性分类算法，其稀疏方式是最新的数据要向字典中的数据做投影。字典维数是否增加，和投影误差有关。当投影误差大于阈值时，新数据进入字典，字典维数增加；反之，小于阈值时，新数据不进入字典，新的系数可以用字典中原有的系数和基底近似表示，字典维数不变。仿真数据和真实数据的实验结果表明，与其他七种经典的在线分类算法相比，本文算法在字典规模较小时，误分类率最低。

基金项目

辽宁省自然科学基金(项目编号 2019-ZD-0106，2019-ZD-0087)。

参考文献

[1] Rosenblatt, F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, **65**, 386-408. https://doi.org/10.1037/h0042519

[2] Freund, Y. and Schapire, R.E. (1999) Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, **37**, 209-217. https://doi.org/10.1023/A:1007662407062

[3] Kivinen, J., Smola, A.J. and Williamson, R.C. (2004) Online Learning with Kernels. *IEEE Transactions on Signal Processing*: *A Publication of the IEEE Signal Processing Society*, **52**, 2165-2176. https://doi.org/10.1109/TSP.2004.830991

[4] Engel, Y., Mannor, S. and Meir, R. (2004) The Kernel Recursive Least-Squares Algorithm. *IEEE Transactions on Signal Processing*, **52**, 2275-2285. https://doi.org/10.1109/TSP.2004.830985

[5] Yamada, I. and Ogura, N. (2004) Adaptive Projected Subgradient Method for Asymptotic Minimization of Sequence of Nonnegative Convex Functions. *Numerical Functional Analysis and Optimization*, **25**, 593-617. https://doi.org/10.1081/NFA-200045806

[6] Slavakis, K., Theodoridis, S. and Yamada, I. (2008) Online Kernel-Based Classification Using Adaptive Projection Algorithms. *IEEE Transactions on Signal Processing*, **56**, 2781-2796. https://doi.org/10.1109/TSP.2008.917376

[7] Cawley, G.C. and Talbot, N.L.C. (2002) Reduced Rank Kernel Ridge Regression. *Neural Processing Letters*, **16**, 239-302. https://doi.org/10.1023/A:1021798002258

[8] Gilbarg, D. and Trudinger, N.S. (1983) Elliptic Partial Differential Equations of Second Order. Springer, Berlin, Heidelberg, 1-517.

[9] Chen, B.D., Zhao, S.L., Zhu, P.P. and Príncipe, J.C. (2012) Quantized Kernel Least Mean Square Algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, **23**, 22-32. https://doi.org/10.1109/TNNLS.2011.2178446

[10] Lu, J., Sahoo, D., Zhao, P.L. and Hoi, S.C.H. (2018) Sparse Passive-Aggressive Learning for Bounded Online Kernel Methods. *ACM Transactions on Intelligent Systems and Technology*, **9**, 1-27. https://doi.org/10.1145/3156684

[11] He, W.W. and Kwok, J.T. (2014) Simple Randomized Algorithms for Online Learning with Kernels. *Neural Networks*, **60**, 17-24. https://doi.org/10.1016/j.neunet.2014.07.006

[12] Orabona, F., Keshet, J. and Caputo, B. (2009) Bounded Kernel-Based Online Learning. *Journal of Machine Learning Research*, **10**, 2643-2666.