Neural Networks 60 (2014) 17–24

Contents lists available at ScienceDirect

Neural Networks

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

Simple randomized algorithms for online learning with kernels

Wenwu He [a,b,*], James T. Kwok [b]

[a] Department of Mathematics and Physics, Fujian University of Technology, Fuzhou, Fujian 350118, China

[b] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 9 April 2014
Received in revised form 7 July 2014
Accepted 17 July 2014
Available online 28 July 2014

Keywords:
Online learning
Kernel methods
Stochastic strategies
Budget

ABSTRACT

In online learning with kernels, it is vital to control the size (budget) of the support set because of the curse of kernelization. In this paper, we propose two simple and effective stochastic strategies for controlling the budget. Both algorithms have an expected regret that is sublinear in the horizon. Experimental results on a number of benchmark data sets demonstrate encouraging performance in terms of both efficacy and efficiency.

1. Introduction

Online learning is a popular and natural approach for solving real-time and life-long learning problems, where instances arrive sequentially. Online learning is also advantageous in large-scale learning as it is often efficient and highly competitive (Shalev-Shwartz, 2007, 2011). At each iteration t, an online learning algorithm produces a function estimate $f_t \in \mathcal{F}$, and then suffers a loss $\ell_t(f_t)$. Here, we assume that the function space $\mathcal{F}$ is closed and convex, and $\ell_t(\cdot)$ is convex. To evaluate the performance of the algorithm, it is customary to measure its regret $R_T = \sum_{t=1}^{T}(\ell_t(f_t) - \ell_t(f))$, where T is the horizon, w.r.t. a competitor $f \in \mathcal{F}$.

A standard online learning algorithm is the gradient descent (GD) (Zinkevich, 2003), which updates f_t as

$$f_{t+1} = \Pi_{\mathcal{F}}(f_t - \eta g_t). \tag{1}$$

Here, g_t is the gradient (or subgradient) of ℓ_t w.r.t. f_t, $\Pi_{\mathcal{F}}$ is the Euclidean projection onto $\mathcal{F}$, and η is the stepsize. Its regret is $O(\sqrt{T})$, and cannot be improved in general (Abernethy, Bartlett, Rakhlin, & Tewari, 2008). To extend linear models for nonlinear function learning, the kernel trick has been widely used (Kivinen, Smola, & Williamson, 2004; Schölkopf & Smola, 2002). An input instance $\mathbf{x}$ is first mapped to $\phi(\mathbf{x})$ in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, where ϕ is a feature map induced by the kernel $\kappa(\cdot,\cdot)$ of $\mathcal{H}$. The inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ between two instances $\mathbf{x}_i, \mathbf{x}_j$ in the linear algorithm is then replaced by $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$.

A bottleneck for kernelized online learning is that the set S_t of support vectors (SV's) in f_t keeps expanding as learning proceeds. Subsequently, so are the memory and time costs. It is thus necessary to keep $|S_t|$ under control with the use of budget (Crammer, Kandola, & Singer, 2003; Weston, Bordes, & Bottou, 2005). In recent years, various budget algorithms have been proposed, such as the Projectron and its variants (He & Wu, 2012; Orabona, Keshet, & Caputo, 2009), randomized budget perceptron (RBP) (Cesa-Bianchi & Gentile, 2006; Sutskever, 2009) and the simplified Forgetron (Dekel, Shalev-Shwartz, & Singer, 2008; Sutskever, 2009). These algorithms are mainly for classification and only mistake bounds are provided. A framework for algorithms with sublinear regret (as in (1)) is lacking and highly desirable in the budget online learning literature. With such a regret guarantee, we can directly generalize existing budget algorithms to regression and other problems. Moreover, though some budget algorithms (such as the Projectron) have remarkable classification accuracies, they have to update the inverse of a Gram matrix in each iteration. This costs $O(B^2)$ time and memory, where B is the budget. Besides, setting an appropriate budget for a particular learning problem can be difficult.

In this paper, we propose a simple but effective stochastic strategy for online learning with budget. The idea is to keep the excess regret of the budget algorithm over its non-budget counterpart small, while still controlling the growth of the budget. In particular, two algorithms, both with $O(B)$ memory and time, will be presented.

* Corresponding author at: Department of Mathematics and Physics, Fujian University of Technology, Fuzhou, Fujian 350118, China. Tel.: +86 13290930180.
E-mail addresses: hwwhbb@163.com, hwwhbb@gmail.com (W. He), jamesk@cse.ust.hk (J.T. Kwok).

http://dx.doi.org/10.1016/j.neunet.2014.07.006
0893-6080/

Algorithm 1 Online learning with kernels (OLK).

Input: Learning rate sequence $\{\eta_t > 0\}$.
Initialize: $S_1 = \emptyset, f_1 = 0$.
for $t = 1, 2, \ldots, T$ **do**
  receive input $\mathbf{x}_t$;
  suffer loss $\ell_t(f_t)$;
  compute the subgradient $g_t \in \partial \ell_t(f_t)$;
  if $g_t = \mathbf{0}$ **then**
    $f_{t+\frac{1}{2}} = f_t$;
    $S_{t+1} = S_t$;
  else
    $f_{t+\frac{1}{2}} = f_t - \eta_t g_t$;
    $S_{t+1} = S_t \cup \{t\}$;
  end if
  $f_{t+1} = \Pi_{\mathcal{F}}(f_{t+\frac{1}{2}})$.
end for

- The first one is based on dynamically increasing the budget in a stochastic manner. This yields a sublinear expected regret of $O(T^{\frac{1+\gamma}{2}})$ (where $0 < \gamma < 1$), and a budget of (variable) size $O(T^{1-\gamma})$ which is also sublinear in T.
- The second algorithm allows the use of a fixed budget, which may be more convenient in some applications. When the pre-assigned budget is exceeded, the algorithm randomly removes an existing SV. This also yields a sublinear expected regret of $O(\sqrt{T})$ and a budget of size $O(\sqrt{T})$.

The rest of this paper is organized as follows. Section 2 presents a baseline algorithm for non-budget kernelized online learning. The stochastic strategy for (kernelized) online learning with budget, with two specific budget algorithms, is demonstrated in Section 3. A discussion on the related work is in Section 4. Experimental results are reported in Section 5, and the last section gives some concluding remarks.

2. Basic algorithm for Online Learning with Kernels

Given a sequence $\{(\mathbf{x}_t, y_t)\}$, with $\mathbf{x}_t$ coming from the input domain $\mathcal{X} \subseteq \mathbb{R}^d$, y_t from output domain $\mathcal{Y} \subseteq \mathbb{R}$, and $t \in [T] \equiv [1, 2, \ldots, T]$, the learner aims to learn the underlying function $f : \mathcal{X} \to \mathcal{Y}$ in an online manner. Let $\mathcal{H}$ be the RKHS associated with the kernel function κ, such that $\kappa_{ij} \equiv \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi_i, \phi_j \rangle$ and $\phi_t \equiv \phi(\mathbf{x}_t) \in \mathcal{H}$. We assume that

$$\|\phi_t\| \leq 1, \quad \forall t \in [T]. \tag{2}$$

Unless explicitly stated, all the norms are ℓ_2-norms. Moreover, let $\mathbf{K}$ be the Gram matrix with elements κ_{ij}'s,

$$\mathcal{F} = \{f \in \mathcal{H} \mid \|f\| \leq U\} \tag{3}$$

for some given $U > 0$, and $f_t \in \mathcal{F}$ be the function learned at iteration t. Denote the loss of f_t at iteration t by $\ell_t(f_t)$, and $g_t \in \partial \ell_t(f_t)$ be its gradient (or subgradient). For any $f \in \mathcal{H}$, the operation

$$\Pi_{\mathcal{F}}(f) = \arg\min_{g \in \mathcal{F}} \|g - f\| = \min\left\{1, \frac{U}{\|f\|}\right\} f \tag{4}$$

projects f onto $\mathcal{F}$.

Algorithm 1 shows a basic kernel extension of GD in (1), and will be called *Online Learning with Kernels* (OLK) in the sequel. Let $\{f_t \in \mathcal{F}\}_{t \in [T]}$ be a sequence obtained by OLK. The following regret can be easily obtained from (Zinkevich, 2003).

Theorem 1. (i) *Using a constant stepsize $\eta > 0$, the regret of OLK is bounded as*

$$R_T \leq \frac{\|f_1 - f\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2. \tag{5}$$

Assume that $\max_{t \in [T]} \|g_t\|^2 \leq G$. We have $R_T \leq 2U\sqrt{GT}$ on setting $\eta = 2UG^{-\frac{1}{2}} T^{-\frac{1}{2}}$.

(ii) *With a dynamic stepsize*

$$\eta_t = \eta t^{-\frac{1}{2}}, \tag{6}$$

we have

$$R_T \leq \frac{\sqrt{T} \max_{t \in [T]} \|f_t - f\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \frac{\|g_t\|^2}{\sqrt{t}}. \tag{7}$$

On setting $\eta = \sqrt{2}UG^{-\frac{1}{2}}$, we have $R_T \leq \sqrt{2}\left(2U\sqrt{GT}\right)$.

The constant stepsize η can be difficult to set unless T is known. The OGD algorithm (Kivinen et al., 2004; Zhao, Wang, Wu, Jin, & Hoi, 2012), which uses a constant stepsize, also suffers from the same problem. Moreover, OGD has a regret of

$$R_T \leq \frac{\lambda T + \eta^{-1}}{2} \|f_1 - f\|^2 + \eta \sum_{t=1}^{T} \|g_t\|^2,$$

where $\lambda > 0$ is a regularization parameter. It is worse than the bound in (5) by a factor of $\frac{1}{2}$ on setting $\lambda T = \eta^{-1}$ (Zhao et al., 2012).

On the other hand, with the dynamic stepsize scheme in (6), η_t does not depend on T, and the learner can tune η for optimal performance based on a subset of samples. The price to pay is that its regret is only $\sqrt{2}$-competitive with that of the fixed stepsize.

We assume that at iteration t, the change which may be added to f_t can be written as $\alpha_t \phi_t$ for some $\alpha_t \in \mathbb{R}$, i.e.,

$$-\eta_t g_t = \alpha_t \phi_t. \tag{8}$$

This holds for many commonly used loss functions. For example, for the hinge loss $\ell_t(f_t) = \max\{0, 1 - y_t f_t(\phi_t)\}$, we have $g_t = -y_t \phi_t$ when $\ell_t(f_t) > 0$. Similarly, for the square loss $\ell_t(f_t) = \frac{1}{2}(y_t - f_t(\phi_t))^2$, we have $g_t = -(y_t - f_t(\phi_t))\phi_t$. Hence, f_t can be written as $f_t = \sum_{\tau \in S_t} \alpha_\tau \phi_\tau$. On non-separable problems or noisy data, the number of support vectors involved in f_t may thus increase with t, and both the update and prediction time will become unlimited. This hinders the application of online learning with kernels to large-scale problems.

When a new SV is to be added, $f_{t+\frac{1}{2}} = \sum_{\tau \in S_t} \alpha_\tau \phi_\tau + \alpha_t \phi_t$. Note that the projection $\Pi_{\mathcal{F}}(f_{t+\frac{1}{2}})$ (defined in (4)) involves computing $\|f_{t+\frac{1}{2}}\|^2$, which can be easily obtained as:

$$\begin{aligned} \left\|f_{t+\frac{1}{2}}\right\|^2 &= \|f_t\|^2 + \alpha_t^2 \kappa_{tt} + 2\alpha_t \sum_{\tau \in S_t} \alpha_\tau \kappa_{\tau t} \\ &= \|f_t\|^2 + \alpha_t^2 \kappa_{tt} + 2\alpha_t f_t(\mathbf{x}_t). \end{aligned} \tag{9}$$

Here, $f_t(\mathbf{x}_t)$ is the prediction at iteration t and needs to be computed anyway. By storing $\|f_t\|^2$ in each iteration, computing (9) is inexpensive in terms of both time and memory.

3. Randomized strategies for online learning with budget

In online learning with budget, we restrict each f_t to have a maximum of B SV's, where $B > 0$. The budget version of OLK can be obtained by replacing $f_{t+\frac{1}{2}}$ in Theorem 1 with $f^B_{t+\frac{1}{2}}$, whose expression is to be specified.

Proposition 1. *For any stepsize sequence $\{\eta_t\}$ (with $\eta_t > 0$), the regret for the budget version of OLK is bounded by*

$$R^B_T \leq R^{\sim B}_T + \sum_{t=1}^{T} \frac{1}{2\eta_t} \left(\|e_t\|^2 + 2\left\langle f_{t+\frac{1}{2}} - f, e_t \right\rangle\right), \tag{10}$$

where $e_t = f^B_{t+\frac{1}{2}} - f_{t+\frac{1}{2}}$, and $R^{\sim B}_T = \sum_{t=1}^{T} \frac{\|f_t - f\|^2 - \|f_{t+1} - f\|^2}{2\eta_t} + \frac{\eta_t \|g_t\|^2}{2}$.