Neurocomputing 179 (2016) 26–36

Contents lists available at ScienceDirect

Neurocomputing

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

Online kernel learning with nearly constant support vectors

Ming Lin [a], Lijun Zhang [b], Rong Jin [b], Shifeng Weng [c], Changshui Zhang [a,*]

[a] State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology(TNList), Department of Automation, Tsinghua University, Beijing 10084, China
[b] Computer Science and Engineering, Michigan State University, East Lansing, MI 48823, USA
[c] School of Electronics and Information, Zhejiang Wanli University NO 8 South Qianhu Road, Ningbo City, Zhejiang Province 315100, China

ARTICLE INFO

Article history:
Received 30 April 2013
Received in revised form
14 September 2015
Accepted 3 October 2015
Communicated by Steven Hoi
Available online 14 October 2015

Keywords:
Online learning
Kernel machine
Nyström
Sample complexity

ABSTRACT

Nyström method has been widely used to improve the computational efficiency of batch kernel learning. The key idea of Nyström method is to randomly sample M support vectors from the collection of T training instances, and learn a kernel classifier in the space spanned by the randomly sampled support vectors. In this work, we studied online regularized kernel learning using the Nyström method, with a focus on the sample complexity, i.e. the number of randomly sampled support vectors that are needed to yield the optimal convergence rate $O(1/T)$, where T is the number of training instances received in online learning. We show that, when the loss function is smooth and strongly convex, only $O(\log^2 T)$ randomly sampled support vectors are needed to guarantee an $O(\log T/T)$ convergence rate, which is almost optimal except for the log T factor. We further validate our theory by an extensive empirical study.

1. Introduction

Kernel machines are powerful tools to handle non-linear data learning tasks. Kernel function improves the flexibility of learning methods by implicitly mapping data to a high dimensional space [1]. Kernel based methods have been successfully applied to classification, dimensionality reduction, and clustering, including kernel SVM [2], kernel logistic regression [3], kernel PCA [4] and spectral clustering [5].

A main drawback of kernel based methods is their high demand on both storage space and computational cycles. Given T training instances, the storage requirement and computational cost are $O(T^2)$. Online learning improves the efficiency of kernel learning by going through the training data once [6–8]. Although it reduces the storage requirement by retrieving training instances one by one in online settings, its time complexity is still $O(T^2)$ because each received training instance can potentially be a support vector. Budget online learning [9–12] ameliorates this problem by limiting number of support vectors of the intermediate classifiers obtained by online learning. But the final classifier obtained by online-to-batch conversion [13] may still include most of the training examples as support vector, leading to a high computational cost in prediction.

An alternative approach to efficient kernel learning is to generate a compact representation for the target kernel classifier. Random Fourier feature [14] and polynomial feature [15,16] are two examples of this category. Both methods approximate kernel function by an expansion of appropriate basis functions. Since the approximation is made independently from data, both schemes are data independent, and therefore often leads to suboptimal performance, according to the analysis in [17].

In this work, we focus on Nyström method [18], another popular scheme for improving the efficiency of kernel learning. It randomly samples M instances as support vectors from a collection of T training examples, and learns a kernel classifier in the subspace spanned by the randomly sampled support vectors. Nyström method was first introduced to kernel learning in [19], and has found applications in kernel classification [20,21], spectral clustering [22], and eigenmap embedding [23].

The generalization performance of Nyström method was examined recently in [17], in which the authors show that Nyström method is overall more effective for batch kernel learning than random Fourier feature because of the data dependence induced by Nyström method. Unlike [17] where the effect of Nyström method was examined in batch learning, we focus on *online* regularized kernel learning where training examples are received sequentially with one at each time, and every training example will be discarded after it is used to update the prediction model. We show that, in online regularized kernel learning, only $O(\log^2 T)$ randomly sampled support vectors are needed to achieve an $O(\log T/T)$ convergence rate. Compared to the optimal convergence rate $O(1/T)$ for online regularized kernel learning, our result is almost optimal except for the $\log(T)$ factor. We verify our theory by an extensive

* Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2015.10.002
0925-2312/

empirical study. To the best of our knowledge, this is the first work that analyzes the performance of Nyström method in online settings, with nearly optimal guarantee.

The rest of this paper is organized as follows. Section 2 discusses the related work on kernel learning and Nyström method. Section 3 describes online regularized kernel learning with Nyström method in details, and present our theoretical guarantees, where the detailed proof can be found in the Appendix. Section 4 demonstrates our theory by an extensive empirical study. Section 5 encloses our paper with future work.

2. Related work

In this section we briefly review the related works on kernel learning.

2.1. Kernel learning

As mentioned in the introduction section, the main challenge arising from kernel learning is its high demand on computational cycles and storage space. Below we list several major efforts in improving the efficiency of kernel learning.

Explicit kernel feature mapping: Explicit kernel feature mapping approximates a kernel similarity function by a finite feature representation of data. When the kernel is shift-invariant, it can be accomplished by random Fourier sampling [14]. It was shown in [24] that the generalization error caused by random Fourier features is bounded by $O(1/\sqrt{M})$, where M is the number of random Fourier features. When kernel is not shift-invariant, polynomial feature representation is often used to approximate the kernel function by a truncated Taylor expansion [25,15,16]. The key limitation of methods in this category is that the kernel approximations are made independently from the data distribution, leading to suboptimal performance as argued in [17].

Batch sparse kernel learning: Sparse kernel learning aims to compute a compact representation of kernel classifier with a limited number of support vectors. A common idea is to confine the support vectors in a reduced set of training data [26,27]. The reduced set is constructed either by a greedy method [3,28] or by minimizing some criterion as a complementary process [26,29,30]. In [31,32], the authors consider approaches for sparse kernel learning by making appropriate changes to the objective function. In [30], the authors propose to first learn a dense kernel SVM through batch learning, and then approximate the learned SVM by a sparse one. Although the output classifier is sparse in support vectors, most methods in this category are expensive in both storage space and computational cost as they have to deal with the full kernel matrix in the first place.

Budget online learning: Budget Online Learning restricts the number of support vectors to a given budget. Crammer et al. [9] propose the first budget online learning, which was refined later on in [10]. The key of these approaches is to remove the support vectors of least significance to maintain the budget. The Forgetron [33] is the first budget online learning with theoretical guarantees. It decreases the weights of support vectors at each iteration of online learning and removes the support vectors with the smallest weight when the number of support vectors exceeds the budget. Randomized Budget Perceptron [34] achieves similar bounds as Forgetron by replacing one of the randomly selected support vectors with new instances. Projectron [35] improves these ideas by making a new training example to be a support vector only when it is far from the space spanned by the existing support vectors. Peilin et al. [8] developed a stochastic gradient descent based method for budget online learning. Very recently, Wang et al. [36] study the convergence rate of online kernel with random Fourier features and Nyström features. Although their analysis is very similar with ours, they only give a linear sampling complexity for Nyström features, which is significantly inferior than the results presented in this paper. It is important to note that our method needs to sample support vectors beforehand and needs independent assumptions, that are not required in conventional analysis. The method proposed in this paper can be viewed as an extension of Projectron with online-to-batch conversion. In our analysis, the optimal convergence rate is only possible with online-to-batch conversion, thus it is not surprising that Projectron cannot provide such guarantee.

Sparse online kernel learning: Sparse online kernel learning maintains a sparse support vector set at each online step and outputs a compact kernel machine without having to take the online-to-batch conversation. Engel et al. [37] proposed a sparse kernel support vector machine for kernel regression, where a new training instance is added into the set of support vectors only when it cannot be linearly approximated by current support vectors. Their method does not provide any guarantee on generalization bounds. Zhang et al. [11] proposed a stochastic gradient method for sparse online kernel learning that shares the similar idea as [8].

2.2. Nyström method

Nyström method was first proposed by Nyström [18]. It was introduced by Williams and Seeger [19] to accelerate kernel learning, followed by [20]. Various sampling schemes have been proposed to improve the effectiveness of Nyström method [38,39].

Nyström method is often viewed as a low rank matrix approximation method: it approximates the kernel matrix K by a low rank matrix $\widehat{K}$. Several analyses have been developed to bound the difference between K and $\widehat{K}$ [20,40–43]. The most interesting result is given in [40,41], which stated that when the rank of kernel matrix is r, only $O(r \log r)$ samplings is needed by Nyström method to achieve a zero error in approximating the kernel matrix. The impact of the low rank approximation made by Nyström method on the generalization performance of kernel learning was studied in [44]. In [17], the authors proved that the generalization error caused by Nyström method is low bounded by $\mathcal{O}(N/M)$, where N is the number of training instances, which can be improved to $O(N/M^{p-1})$ if the eigenvalues of the kernel matrix follow a p power law. Different from the existing studies, we focus on the generalization performance of Nyström method in the online setting.

3. Online kernel learning with nyström method

3.1. Background and notation

Let $\kappa(\cdot,\cdot)$ be a bounded kernel function, i.e., $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, $|\kappa(\boldsymbol{x}, \boldsymbol{x}')| \leq 1$. Denote by $\mathcal{H}$ the *Reproducing Kernel Hilbert Space* (RKHS) endowed with $\kappa(\cdot,\cdot)$. Let $\langle \cdot,\cdot \rangle_{\mathcal{H}}$ be the inner product on $\mathcal{H}$ and $\|\cdot\|_{\mathcal{H}}$ be the corresponding norm. Let $\boldsymbol{z}_t = \{\boldsymbol{x}_t, y_t\}, t = 1, \ldots, T$ be the sequence of training examples received in the online setting, where $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ is a column vector of d dimension and the label $y_t \in \{+1, -1\}$. We assume that all the training examples are i.i.d. samples from an unknown underlying distribution $\mathbb{P}(\boldsymbol{x}, y)$.

Given the sequence of training examples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T\}$, we define the kernel matrix $K \in \mathbb{R}^{T \times T}$ by $K_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Through out the paper, we assume that the kernel to be bounded,

$$|\kappa(\boldsymbol{x}, \boldsymbol{y})| \leq 1.$$

We denote by X the set of training instances, and by V the set of support vectors, i.e.,

$$X \triangleq \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}, \quad V \triangleq \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_M\},$$

where each $\hat{\boldsymbol{x}}_i$ is a support vector used by the kernel classifier and M is the number of support vectors. We define $\mathcal{H}_V$, the subspace