Variance Reduced Online Gradient Descent for Kernelized Pairwise Learning with Limited Memory

Hilal AlQuabeh HILAL.ALQUABEH@MBZUAI.AC.AE
Bhaskar Mukhoty BHASKAR.MUKHOTY@GMAIL.COM
Bin Gu BIN.GU@MBZUAI.AC.AE
Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Pairwise learning is essential in machine learning, especially for problems involving loss functions defined on pairs of training examples. Online gradient descent (OGD) algorithms have been proposed to handle online pairwise learning, where data arrives sequentially. However, the pairwise nature of the problem makes scalability challenging, as the gradient computation for a new sample involves all past samples. Recent advancements in OGD algorithms have aimed to reduce the complexity of calculating online gradients, achieving complexities less than $O(T)$ and even as low as $O(1)$. However, these approaches are primarily limited to linear models and have induced variance. In this study, we propose a limited memory OGD algorithm that extends to kernel online pairwise learning while improving the sublinear regret. Specifically, we establish a clear connection between the variance of online gradients and the regret, and construct online gradients using the most recent stratified samples with a limited buffer of size of s representing all past data, which have a complexity of $O(sT)$ and employs $O(\sqrt{T}\log T)$ random Fourier features for kernel approximation. Importantly, our theoretical results demonstrate that the variance-reduced online gradients lead to an improved sublinear regret bound. The experiments on real-world datasets demonstrate the superiority of our algorithm over both kernelized and linear online pairwise learning algorithms. The code is available at https://github.com/halquabeh/ACML-2023-FPOGD-Code.git.

Keywords: Pairwise learning, AUC maximization, Random Fourier features, Online stratified Sampling

1. Introduction

Pairwise learning is a machine learning paradigm that focuses on problems where the loss function is defined on pairs of training examples. It has gained significant attention due to its wide range of applications in various domains. For instance, in metric learning Kulis et al. (2012), pairwise learning is used to learn a similarity or distance metric between data points. In bipartite learning Kallus and Zhou (2019), it is employed to address fairness concerns when making decisions based on two distinct groups. Multiple kernel learning utilizes pairwise learning to combine multiple kernels and enhance the performance of kernel-based methods Gönen and Alpaydın (2011). AUC maximization involves pairwise learning to optimize the Area Under the ROC Curve, a popular evaluation metric for binary classification Hanley and

McNeil (1982). Pairwise differential Siamese networks utilize pairwise learning to compare and classify pairs of samples Kang et al. (2018); Song et al. (2019).

Online pairwise learning is an effective approach for real-time decision-making, particularly when dealing with large-scale and dynamic datasets. The process involves sequentially processing data points and updating the model using pairwise examples. One technique that has been explored is online gradient descent, which provides computational efficiency and scalability. However, a drawback of online gradient descent in pairwise learning is its time complexity of $O(T^2)$, where T represents the number of received examples. This is due to the requirement of pairing each new data point with all previous points, leading to significant computational complexity. To address this limitation, researchers have been investigating alternative methods such as buffering and sampling strategies Zhao et al. (2011); Kar et al. (2013); Yang et al. (2021). These approaches aim to reduce the computational burden and enable efficient learning on large-scale problems.

Online gradient descent has gained significant attention in the domain of online pairwise learning, leading to the development of various approaches. These methods, including online buffer learning Zhao et al. (2011); Kar et al. (2013), second-order statistic online learning Gao et al. (2013), and saddle point-problem methods Ying et al. (2016); Reddi et al. (2016), have all employed linear models. Moreover, there has been limited exploration of non-linear models in this field, particularly with kernelized learning Ying and Zhou (2015); Du et al. (2016). One noteworthy approach in pairwise learning is online buffer learning, introduced by Zhao et al. Zhao et al. (2011). This method utilizes a finite buffer with reservoir sampling to reduce the time complexity to $O(sT)$, where s denotes the buffer size. By storing a subset of the data and ensuring uniform samples within the buffer, this technique effectively alleviates the computational burden. Furthermore, Yang et al. Yang et al. (2021) achieved optimal generalization with a buffer size of $s = 1$, marking a significant advancement in the field.

The existing frameworks in the literature have primarily focused on linearly separable data, overlooking the challenges associated with non-linear pairwise learning. Moreover, the online buffer methods proposed so far have not adequately addressed the sensitivity of generalization to the variance of the gradient. This limitation restricts their ability to capture the complexity present in real-world datasets. Moreover, there is a lack of extensive research on non-linear pairwise learning, particularly in the context of kernel approximation. Although non-linear methods provide increased expressive power, the computational cost associated with kernel computation, which scales as $O(T^2)$ Lin et al. (2017); Kakkar et al. (2017), poses challenges to their scalability and efficiency in practical applications. In terms of generalization bounds, the analysis of online pairwise gradient descent with buffers and linear models has been extensively explored in previous works Wang et al. (2012); Kar et al. (2013). These studies establish a bound of $O(1/s + 1/\sqrt{T})$ for this approach. However, it is important to note that this bound is only optimal when the buffer size s is approximately $O(\sqrt{T})$, posing challenges for scenarios where a smaller buffer size is desired. Additionally, the generalization analysis in Yang et al. (2021) assumes independent examples in the sequential data, disregarding the temporal nature of the data and the potential ordering and correlation among data points. This assumption may lead to inaccurate performance estimation and unreliable convergence guarantees in online learning scenarios. Taken together, these weaknesses highlight the need for further research and development in the field of online pairwise learning to address the limitations of linear frameworks, explore non-linear

Table 1: Recent pairwise learning algorithms(where T is the iteration number, d: the dimension, D is random features, V.R. denotes variance reduction, and s is a buffer size), note the time complexity is w.r.t. gradients computations.

Algorithm	**Problem**	**Model**	**Scheme**	**V.R.**	**Time**	**Space**
Gu et al. (2019)	AUC	Linear	Offline	NA	$O(T)$	$O(1)$
Natole et al. (2018)	AUC	Linear	Online	NA	$O(T)$	$O(1)$
Ying et al. (2016)	AUC	Linear	Online	NA	$O(T)$	$O(1)$
Zhao et al. (2011)	AUC	Linear	Online	No	$O(sT)$	$O(s)$
Gao et al. (2013)	AUC	Linear	Online	No	$O(T)$	$O(d)$
Yang et al. (2021)	General	Linear	Online	No	$O(T)$	$O(1)$
Kar et al. (2013)	General	Linear	Online	No	$O(sT)$	$O(s)$
Lin et al. (2017)	General	Kernel	Online	NA	$O(T^2)$	$O(T)$
Kakkar et al. (2017)	AUC	Kernel	Offline	NA	$O(T \log T)$	$O(T^2)$
FPOGD (Ours)	General	Kernel	Online	Yes	$O(\frac{sD}{d}T)$	$O(s)$

methods more comprehensively, and overcome the computational challenges associated with kernel computation.

Our approach extends online pairwise learning to handle nonlinear data by incorporating kernelization of the input space. We address the impact of variance on regret through online stratified sampling, selectively updating the model based on cluster relevance. Utilizing random Fourier features, we efficiently estimate the kernel with sublinear error bound, achieving computational savings without sacrificing performance. By combining kernelization, efficient kernel approximation, and online stratified sampling, our method overcomes linear limitations, handles nonlinear data, and mitigates variance impact, resulting in a robust and effective online pairwise learning approach (Table 1). Our main contributions can be summarized as follows:

- We present an online pairwise algorithm for non-linear models with fast convergence. Our algorithm achieves sublinear regret with a buffer size of $O(s)$.
- We address variance impact on regret and propose online stratified sampling to control and improve the regret rate.
- For the case of Gaussian kernel, we approximate the pairwise kernel function using only $O(\sqrt{T}\log T)$ features in comparison to $O(T)$ in previous works, while maintaining a sublinear error bound.
- We demonstrate the effectiveness of our proposed technique on numerous real-world datasets and compare it with state-of-the-art methods for AUC maximization. Our methodology showcases improvements across both linear and nonlinear models for the majority of the examined datasets.

The following sections are organized as follows. Section 2 introduces the problem setting, section 3 presents the proposed method, section 4 provides the regret analysis, section 5

discusses related work, followed by section 6 producing the experimental results, and finally section 7 concludes the paper.

2. Problem Setting

The concept of pairwise learning arises in the context of a subset $\mathcal{X} \subset \mathbb{R}^d$ and a label space $\mathcal{Y} \subset \mathbb{R}$. It can be categorized into two cases:

- Pairwise hypothesis: This case involves learning a pairwise hypothesis, such as in metric learning, where the goal is to determine the relationship or distance between pairs of data points in $\mathcal{X}$. In particular, the hypothesis predicts the distance between pairs of instances i.e. $f : \mathcal{X}^2 \mapsto \mathcal{Y}$, and therefore given n examples, the loss function is a finite sum of $\binom{n}{2}$ terms.
- Pairwise loss function: In this case, the focus is on minimizing a pairwise loss function, such as in AUC (area under curve) maximization. The objective is to optimize the ordering or ranking of pairs of data points based on their labels in $\mathcal{Y}$. In general the hypothesis is pointwise as in SVM, regression and binary deep classification, i.e. $f : \mathcal{X} \mapsto \mathcal{Y}$, however the loss function itself represents the probability of predicting correctly the labels of opposing examples.

In our analysis, we specifically investigate pairwise loss functions from both branches. We establish a connection between the pairwise kernel associated with pairwise hypotheses and regular kernels. This enables us to explore the characteristics of the pairwise loss functions within the framework of regular kernels. Consider an algorithm that learns from examples $z_i := (x_i, y_i) \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, where $i \in [T]$ denotes the number of examples. Let f belong to space $\mathcal{H}$. In this paradigm, the pairwise loss function serves as a performance measure, denoted as $\ell : \mathcal{H} \times \mathcal{Z}^2 \to \mathbb{R}_+$.

Likewise, in online learning with pairwise losses, when a new data point z_t is received, a local error is generated by incorporating the new data point together with all previous $t-1$ points. The local error is then determined based on the chosen pairwise loss function as follows,

$$L_t(f_{t-1}) = \frac{1}{t-1} \sum_{i=1}^{t-1} \ell(f_{t-1}, z_t, z_i) \tag{1}$$

The core objective in online pairwise learning is to create an ensemble of models, denoted as $f_1, f_2, \ldots, f_T$, aimed at minimizing the expected risk. Assuming the data is mapped to a higher-dimensional space where linear separability is achieved, we consider a linear model represented as w. To address the issue of memory requirements, we employ a buffer-based local error denoted as $\bar{L}_t(w_{t-1})$, as defined in Equation 2. At each step t, the buffer, denoted as B_t, contains a limited number of historical example indices, and the cardinality of the buffer is represented as $|B_t|$ (equivalent to s in the existing literature).

$$\bar{L}_t(w_{t-1}) = \frac{1}{|B_t|} \sum_{i \in B_t} \ell(w_{t-1}, z_t, z_i) \tag{2}$$

The buffer plays a critical role in the learning process, being updated at each step using diverse strategies, ranging from randomized techniques like reservoir sampling Zhao et al. (2011);

Kar et al. (2013) to non-randomized approaches like FIFO Yang et al. (2021). However, it is worth noting that there is a noticeable research gap regarding the variance implications of these sampling methods, despite their widespread utilization.

To handle complex real-world data, our pairwise online approach assumes mapping both the hypothesis and the data to a Reproducing Kernel Hilbert Space (RKHS) denoted as $\mathcal{H}$. The associated Mercer pairwise kernel function $k : \mathcal{X}^4 \mapsto \mathbb{R}$ satisfies the reproducing property $\langle k_{(x,x')}, g\rangle = g(x, x')$, where $x, x' \in \mathcal{X}^2$ and $g \in \mathcal{H}$. In the case of pointwise hypothesis but pairwise loss functions, such as AUC loss, the kernel function simplifies to $k : \mathcal{X}^2 \mapsto \mathbb{R}$. The space $\mathcal{H}$ encompasses all linear combinations of the functional mappings $\{k_{(x,x')}|(x, x') \in \mathcal{X}^2\}$ and their limit points.

To address the computational complexity of kernelization in the online setting, we utilize random Fourier features (RFF) as an efficient approximation of the Mercer kernel function. RFF provides a lower-dimensional mapping $r(\cdot)$, which approximates the kernel function , the estimate is denoted as $\bar{k}(\cdot)$. This approximation allows us to perform computations using linear operations, significantly reducing the computational complexity. The space spanned by the new kernel functions is denoted as $\hat{\mathcal{H}}$. Previous work has studied the error of random Fourier approximation in pointwise and offline settings. In the online setting, the minimum number of random features required to ensure sublinear regret has been found to be $O(T)$. In our method, we introduce an error bound for pairwise problems using only $O(\sqrt{T}\log T)$ random features (see Section 5 for more details).

2.1. Assumptions

Before introducing our main theorems, we outline a set of widely accepted assumptions concerning the properties of the loss function and kernels. These assumptions hold significance in the realm of convex optimization and encompass commonly used loss functions such as square loss as well as popular kernels like the Gaussian kernel.

Assumption 1 (M-smoothness) *Assume for any $a \in \mathcal{Z}^2 \times \mathcal{H}$, the gradient of the loss function $\nabla\ell(a)$ is M-Lipschitz continuous, i.e. $\forall w, w' \in \mathcal{H}$,*

$$\|\nabla\ell(a) - \nabla\ell(a')\| \leq M \left\|a - a'\right\|_2 .$$

Assumption 2 (Convexity) *Assume for any $z, z' \in \mathcal{Z}$, the loss function $\ell(\cdot, z, z')$ is convex function, i.e. $\forall w, w' \in \mathcal{H}$,*

$$\ell(w, z, z') \geq \ell(w', z, z') + \nabla\ell(w', z, z')^T(w - w').$$

Assumption 3 (Finite Kernel) *Assume for any ρ-probability measure on $\mathcal{X}^2$ the positive kernel function $k : \mathcal{X}^2 \times \mathcal{X}^2 \to \mathbb{R}$ is ρ-integrable, i.e. for any $(x, x') \in \mathcal{X}^2$,*

$$\int\int_{\mathcal{X}^2} k((x, x'), (\hat{x}, \hat{x}'))d\rho(\hat{x})d\rho(\hat{x}') < \infty.$$

2.2. Preliminaries

In the analysis of buffer-based pairwise online gradient descent algorithms, two key concepts are essential for understanding the relationship between regret and variance in the proposed method.

Variance of Stochastic Gradient. Let us denote the variance of the stochastic gradient as $\mathbb{V}(u_t)$, with $u_t := \nabla \bar{L}_t(w_{t-1})$ is the gradient based on finite buffer. The variance is defined as the trace of the covariance matrix: $\mathbb{V}(u_t) := \mathbb{E}\|u_t - \mathbb{E}u_t\|^2$. The following lemma sheds light on the connection between the variance of the gradient and the distance between inputs and their corresponding expected values.

Lemma 1 *Assuming that the loss function is M-smooth, as mentioned in assumption 1, let z_i denote the i-th sample drawn from a uniform distribution i.e. $i \sim Uniform(1, t-1)$. Then, the variance $\mathbb{V}(u_t)$ of the stochastic gradient is bounded by:*

$$\mathbb{V}(u_t) \leq M^2 \mathbb{E}\|x_i - \mathbb{E}x_i\|^2 \tag{3}$$

where the expectation is w.r.t. the uniform random variable i. Proof in appendix.

Regret Bound and Stochastic Gradient Variance. The variance of the stochastic gradient plays a crucial role in determining the regret bound of pairwise online gradient descent algorithms. The following lemma establishes a connection between regret and the variance of the stochastic gradient.

Lemma 2 *With assumption 2, let $[w_t]_{i=1}^T$ be the sequence of models returned by running any buffer-based algorithm for T time-steps using an online sequence of data. If $\bar{w}^* = \arg\min_{w \in \hat{\mathcal{H}}} \sum_{t=2}^T L_t(w)$, and B_t is sampled from the history of the received examples uniformly and independently, then the following hold,*

$$\sum_{t=2}^{T} L_t(w_{t-1}) \leq \sum_{t=2}^{T} L_t(\bar{w}^*) + \frac{\|\bar{w}^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=2}^{T} (\mathbb{V}(u_t) + \|\mathbb{E}u_t\|^2) \tag{4}$$

where the expectation is w.r.t. the uniform distribution of buffer examples. The detailed derivation is in appendix.

Hence, reducing the variance of the stochastic gradient can improve the regret of buffer online pairwise learning, which can be achieved using online stratified sampling as illustrated in next section.

3. Proposed Method

The proposed method consists of two essential parts that are mutually dependent. Firstly, by mapping non-linearly separable data to the RKHS $\mathcal{H}$, we achieve a transformation that renders the data linearly separable. This mapping serves as the foundation for effectively addressing non-linearity. Secondly, through the strategic implementation of stratified sampling, we potentially reduce the variance, preserve low memory utilization, and achieve sublinear regret.The initial mapping to the RKHS empowers us to seamlessly fulfill the objectives of the second component, ensuring an efficient approach overall as illustrated in figure 1.

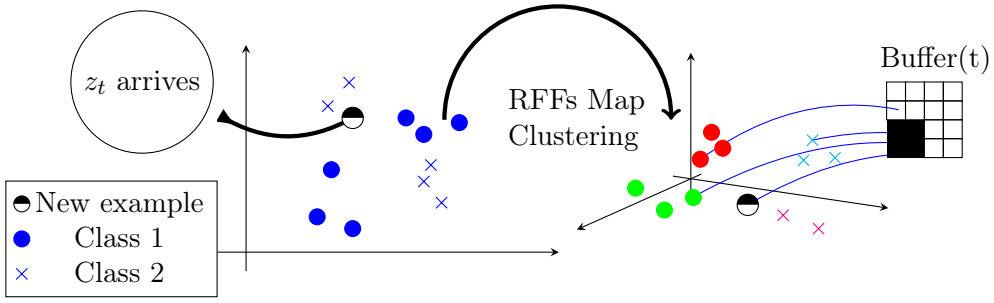

Figure 1: The incoming point z_t from the left are transformed through kernelization and RFFs to a new space. Subsequently, they are clustered and added to the buffer using either a FIFO strategy or with a probability-based approach.

3.1. Non-Linear Mapping to RKHS

In pairwise online gradient descent algorithms, our goal is to handle non-linearly separable data. To achieve this, we employ a transformation $\Phi : \mathcal{X} \mapsto \mathcal{H}$ that maps the input space $\mathcal{X}$ to a high-dimensional RKHS $\mathcal{H}$. By performing this transformation, we project the non-linearly separable data into a higher-dimensional space where it becomes linearly separable.

However, the computational complexity associated with using explicit kernel computations can be prohibitive. To address this challenge, we leverage the RFFs technique, which allows us to approximate the inner products in the target space $\mathcal{H}$ more efficiently. Instead of directly computing the inner products in $\mathcal{H}$, we map the input space to an approximate space $\hat{\mathcal{H}}$ using a randomized mapping function. This approximation enables us to estimate the inner products of the original data points in the approximated space, rather than in the full high-dimensional space.

By applying this non-linear mapping using RFFs to the RKHS, we effectively handle the non-linearity in pairwise online gradient descent algorithms. This approach facilitates better separation of data points in the transformed space $\hat{\mathcal{H}}$, while maintaining a lower computational complexity of $O\left(\frac{sD}{d}T\right)$, compared to the $O(T^2)$ complexity associated with explicit kernel computations.

3.2. Online Stratified Sampling

In addition to the non-linear mapping, we employ stratified sampling to further improve the efficiency and reduce the variance of the stochastic gradient estimates. Online Stratified Sampling (OSS) partitions the input space into balls of radius ϵ, and ensures that each ball is represented by a uniform sample every iteration. By doing so, we achieve several advantages.

Firstly, stratified sampling reduces the variance of the stochastic gradient estimates. By partitioning the input space and uniformly sampling within each partition, we effectively minimize the variance of the stochastic gradient. This is achieved by reducing the expected distance between the sampled variables and their corresponding expected values, as highlighted in Lemma 1.

Secondly, stratified sampling preserves low memory utilization. Instead of storing the entire history of received examples, we maintain one uniform sample from each partition at

every step t. This approach reduces the memory requirement while still providing sufficient information to estimate the gradients accurately.

Finally, stratified sampling enables us to achieve sublinear regret. By reducing the variance and preserving low memory utilization, our method ensures efficient exploration and exploitation of the data, leading to improved regret bounds. To this end, we redefine the loss at each time step by considering the presence of κ_t partitions, denoted as $\mathcal{C}_j^t$, where each partition has a cardinality of $|\mathcal{C}_j^t|$, and its corresponding gradient shown in equation 5.

$$\nabla \bar{L}_t(w_{t-1}) = \sum_{j=1}^{|B_t|} \frac{|\mathcal{C}_j^t|}{t-1} \nabla \ell(w_{t-1}, z_t, z_{B_t[j]}) \tag{5}$$

Note that the gradient mentioned above is unbiased, i.e. $\mathbb{E}\nabla \bar{L}_t(w_{t-1}) = \nabla L_t(w_{t-1})$. In order to reduce the variance in the stochastic gradient estimation, we maintain one uniform sample from each partition at every step t, ensuring that $|B|_t = \kappa_t$. This update approach enables us to achieve lower variance. To accomplish this, we aim to find the optimal partitions at each time step t, which involves solving the following optimization problem or its upper bound based on Lemma 1.

$$\begin{aligned} \min_{\mathcal{C}_j^t} \quad \mathbb{V}(u_t) &= \sum_{j=1}^{\kappa_t} \frac{(|\mathcal{C}_j^t|)^2}{(t-1)^2} \mathbb{E}_{j|j\in\mathcal{C}_j^t} \|\nabla \ell(w, z_t, z_j) - \mathbb{E}_{j|j\in\mathcal{C}_j^t} \nabla \ell(w, z_t, z_j)\|^2 \\ &\leq \sum_{j=1}^{\kappa_t} \frac{(|\mathcal{C}_j^t|)^2 M^2}{(t-1)^2} \mathbb{E}_{j|j\in\mathcal{C}_j^t} \|x_j - \mathbb{E}_{j|j\in\mathcal{C}_j^t} x_j\|^2 = \sum_{j=1}^{\kappa_t} \frac{|\mathcal{C}_j^t| M^2}{(t-1)^2} \sum_{j\in\mathcal{C}_j^t} \|x_j - \mathbb{E}_{j|j\in\mathcal{C}_j^t} x_j\|^2 \end{aligned} \tag{6}$$

The objective in our approach bears resemblance to a conventional clustering problem, where $\mathbb{E}_{j|j\in\mathcal{C}_j^t} x_j = \sum_{j\in\mathcal{C}_j^t} \frac{x_j}{|\mathcal{C}_j^t|}$ represents the centroid of partition j (Note that the partition may be referred to as "cluster" intermittently.). Our approach offers an effective solution by simultaneously addressing memory efficiency and variance reduction.

3.3. The Algorithm

We introduce the algorithm in 1 for general pairwise learning with kernel approximation using RFFs. The centroid update of the OSS algorithm involves minimizing the upper bound given in equation 6. This minimization can be achieved by utilizing the gradient of the upper bound with respect to the centroid c of partition j, denoted as $c := \mathbb{E}_{j|j\in\mathcal{C}_j^t}[x_j]$. The centroid update is then performed as follows:

$$c \leftarrow c + \eta_c(z_t - c) \tag{7}$$

where z_t represents the newly assigned example to partition j, and η_c is the step size. This update ensures that the centroid moves towards the assigned example in order to minimize the upper bound. In the subsequent section, we provide an analysis that decomposes the regret of the algorithm into two distinct components, first is regret of learning $\bar{w}^* := \arg\min_{w\in\hat{\mathcal{H}}} L(w)$, and second regret of the approximation of the kernel function using RFFs mapping $r(\cdot)$.

Algorithm 1 Fourier Pairwise Online Gradient Descent (FPOGD)

Require: Initialization $w_1 \in \hat{\mathcal{H}}$, random feature size D, initial example z_1, Fourier feature distribution p of kernel k, step size η, clustering distance ϵ.

$B_1 \leftarrow \{z_1\}$, $C_1 \leftarrow \{z_1\}$
for $t = 2, \ldots, T$ **do**
  Receive new example $(x_t, y_t) \in \mathcal{Z}$
  Sample Fourier feature $\{q_i\}_{i=1}^{D/2} \sim p$
  Map to new space:

$$z_t = (\frac{2}{\sqrt{D}}[\sin(q_i^T x_t), \cos(q_i^T x_t)]_{i=1}^{D/2}, y_t)$$

  Suffer loss

$$g_t(w_{t-1}) = \sum_{j=1}^{|B_t|} \frac{|\mathcal{C}_j^t|}{t-1} \ell(w_{t-1}, z_t, z_{B_t[j]})$$

  Select $v_t \in \partial g_t(w_{t-1})$
  Update model $w_t = w_{t-1} - \eta v_t$
  Update $B_t, C_t = OSS(B_{t-1}, C_{t-1}, \epsilon, z_t)$
end for
Return w_T

Online Stratified Sampling (OSS) with FIFO Buffer Update

Require: Buffer B, Centroids C, Clustering threshold ϵ, new example $z_t = (x_t, y_t)$

if $\min_{c \in C} \|x_t - c\|^2 \leq \epsilon$ **then**
  Select cluster

$$j := \arg\min_{c \in C} \|x_t - c\|^2$$

  Update buffer $B[j] \leftarrow t$
  Update Centroid j using equation 7
else
  Add new cluster $C \leftarrow C \cup t$
  Add new examples $B \leftarrow B \cup t$
end if
Return B, C

4. Regret Analysis

The regret of online algorithm relative to the optimal hypothesis in the space $\mathcal{H}$, i.e. $w^* = \arg\min_{w \in \mathcal{H}} \sum_{t=2}^{T} L_t(w)$ when running on a sequence of T examples is,

$$\mathcal{R}_{w^*,T} = \sum_{t=2}^{T} L_t(w_{t-1}) - \sum_{t=2}^{T} L_t(w^*), \tag{8}$$

where the local all pairs loss $L_t(\cdot)$ is defined in equation 1. We can decompose the regret in equation 8 by introducing best-in-class hypothesis in the approximated space $\hat{\mathcal{H}}$, e.g: $\bar{w}^* = \arg\min_{w \in \hat{\mathcal{H}}} \sum_{t=2}^{T} L_t(w)$ as follow:

$$\mathcal{R}_{w^*,T} = \underbrace{\sum_{t=2}^{T} L_t(w_{t-1}) - \sum_{t=2}^{T} L_t(\bar{w}^*)}_{T_1} + \underbrace{\sum_{t=2}^{T} L_t(\bar{w}^*) - \sum_{t=2}^{T} L_t(w^*)}_{T_2}$$

We provide the bound on T_1 in Theorem 5 (Section 4.1), and then provide the bound on T_2 in Theorem 8 (Section 4.2). Finally, combining them together, we could provide the main theorem on the regret for the Algorithm 1 as follows.

Theorem 3 *Let $\{z_t \in \mathcal{Z}\}_{t=1}^T$ be sequentially accessed by algorithm 1. Let D be the number of RFFs. And assumptions 1, 2 and 3 hold, with M Lipschitz constant. Then, if the step size $\eta \in (0, 1/M]$, the regret bound compared to w^* is bounded with probability at least $1 - 2^8 \left(\frac{\sigma R}{\delta}\right) \exp(-D\delta^2/(4d+8))$ as follow,*

$$\mathcal{R}_{w^*,T} \leq \frac{M}{2} T \|w^*\|_1^2 \delta^2 + \frac{\|\bar{w}^*\|^2}{2\eta} + \eta \sum_{t=2}^{T} \mathbb{V}(u_t) \tag{9}$$

where $\|w^*\|_1 = \sum_{i,j\neq i}^{T} |a^*_{i,j}|$, δ is the kernel approximation error, κ_t is the number of clusters, σ is the kernel width, $\mathbb{V}(u_t) = tr(cov[u_t])$, R is the input diameter.

Remark 4 *Choosing $\eta = \frac{1}{\sqrt{T}}$ and $\delta = T^{-1/4}$ makes the regret bound sublinear which is optimal. Moreover, if $\mathbb{V}(u_t) = 0$ for all t's, then it's possible to have $\log(T)$ regret by choosing $\eta = \frac{1}{\log(T)}$ and $\delta = \frac{\log(T)}{\sqrt{T}}$, which is similar to the case of full history update.Note that $D = O(\sqrt{T}\log(T))$ in general, but can be as low as $\log^2(T)$ for special kernels, (please refer to appendix B).*

In the following, we provide the analysis to the upper bounds to T_1 and T_2 respectively.

4.1. Regret in the Approximated Space $\hat{\mathcal{H}}$

The choice of buffer updating method, whether randomized (e.g., reservoir sampling) or non-randomized (e.g., FIFO), significantly impacts the analysis, as highlighted by Kar et al. (2013) and Wang et al. (2012). To ensure independence between sampling randomness and data randomness, we begin with a simple FIFO approach, proving T1 bound in Theorem 5 under the i.i.d. assumption. We then introduce reservoir sampling, which uniformly samples from the stream without *i.i.d.* assumption, establishing convergence using the Rademacher complexity of pairwise classes.

Consider the algorithm that has sequential access to the online stream . The following theorem demonstrates that the algorithm achieves optimal regret with memory complexity $O(\kappa_t)$.

Theorem 5 *With assumptions 1 and 2, let $[w_t]_{i=1}^T$ be the sequence of models returned by running Algorithm 1 for T times using the online i.i.d. sequence of data. Then, if $\bar{w}^* = \arg\min_{w\in\hat{\mathcal{H}}} \sum_{t=2}^{T} L_t(w)$, and $\eta \in (0, 1/M]$, the following holds:*

$$\sum_{t=2}^{T} L_t(w_{t-1}) - \sum_{t=2}^{T} L_t(\bar{w}^*) \leq \frac{\|\bar{w}^*\|^2}{2\eta} + \eta \sum_{t=2}^{T} \mathbb{V}(u_t) \tag{10}$$

The proof is in appendix.

Remark 6 *If the original space is assumed to be linearly separable (without kernels) then our algorithm has time complexity of $O(\kappa_T T)$ and offers sublinear regret with $\eta = 1/\sqrt{T}$. Moreover note that for the case of $\epsilon \geq R$ the algorithm is equivalent to Yang et al. (2021) and if $\epsilon \leq R/T$ it matches the algorithm in Boissier et al. (2016). In particular, if the*

clustering radius ϵ is large enough, it will result in only one cluster, similar to Yang's approach. Conversely, if ϵ is small (smaller than the distance between any two examples), it will create a cluster for each example, similar to Boissier's approach.

Remark 7 *In the worst-case scenario, data points in pairwise learning are either assigned to new clusters or grouped within an epsilon distance from the initial centroid. The maximum number of clusters κ_t at step t can be upper-bounded by considering non-overlapping hyperspheres of radius epsilon in the bounded input space. Given the input space's volume V_t^d at time step t, cluster threshold of ϵ, and the gamma function $\Gamma(\cdot)$, we have:*

$$\kappa_t = min\left\{t-1, \left\lceil \frac{V_t^d \Gamma(d/2+1)}{\pi^{(d/2)}\epsilon^d} \right\rceil \right\} \tag{11}$$

For a hypersphere input space with a constant radius R at all time steps, the maximum number of clusters simplifies to $\kappa_t = min\left\{t-1, \lceil (R/\epsilon)^d \rceil\right\}$. In practice, the actual number of clusters obtained may be lower due to the data distribution (For experimental validations, please refer to Appendix B.).

Randomized Buffer Update In practical online learning scenarios, the assumption of an i.i.d. online stream is often impractical since the data can be dependent. Further, uniformly sampling from an online stream is not straightforward, making it challenging to achieve the bound mentioned earlier when the history examples are not readily available in a memory. To address this, we use buffer update strategies that force data independence in the buffer. Stream oblivious methods are particularly useful as they separate the randomness of the data from the buffer construction. To ensure effective buffer update and maintain the desired representation, we adopt reservoir sampling in conjunction with the clustering strategy. This approach treats each partition stream independently. When a new example arrives to cluster $\mathcal{C}_j^t$, the old example is replaced with a probability of $1/|\mathcal{C}_j^t|$, which makes it challenging to establish a uniform distribution among every cluster. Finally, the bound in theorem 5 holds with the assumption of model-buffer independence (refer to appendix for proof).

4.2. Regret of RFFs Approximation

The kernel associated with the pairwise hypothesis in the space $\mathcal{H}$ is a function defined as $k : \mathcal{X}^2 \times \mathcal{X}^2 \mapsto \mathbb{R}^+$ with a shorthand $k_{(x,x')}(\cdot) := k((x,x'),(\cdot,\cdot))$ and can be constructed given any uni-variate kernel $\mathcal{G}$ for any $x_1, x_2, x_1', x_2' \in \mathcal{X}$ as follow,

$$\begin{aligned} k(x_1,x_2,x_1',x_2') &= \mathcal{G}(x_1,x_1') + \mathcal{G}(x_2,x_2') - \mathcal{G}(x_1,x_2') - \mathcal{G}(x_2,x_1') \\ &= \langle \mathcal{G}_{x_1} - \mathcal{G}_{x_2}, \mathcal{G}_{x_1'} - \mathcal{G}_{x_2'} \rangle_{\mathcal{G}} \end{aligned} \tag{12}$$

It's clear that the pairwise kernel k defined above is positive semi-definite on $\mathcal{X}^2$, and therefore it's Mercel kernel if $\mathcal{G}$ does on $\mathcal{X}$ (e.g. see Ying and Zhou (2015)). We further assume there exist a lower dimensional mapping $r : \mathcal{X} \mapsto R^D$, such that $\mathcal{G}_x(\cdot) \approx r(x)^T r(\cdot)$.

The quality of the approximation of the pointwise kernel $\mathcal{G}$ by random Fourier features is studied in literature (see Rahimi and Recht (2007),Bach (2017),Li (2022)), however the approximation of pairwise kernel $k(\cdot)$ needs further analysis. Let the kernel function $\mathcal{G}(\cdot)$

be shift invariant and positive definite, thus using Bochner's theorem, it can be represented by the inverse Fourier transform of a non-negative measure p as $\mathcal{G}(x, x') = \mathcal{G}(x - x') = \int p(q)e^{iq^T(x-x')}du$ where $i = \sqrt{-1}$. For example if the kernel is the Gaussian kernel, the measure is found by Fourier transform to be $p \propto \mathcal{N}(\mathbf{0}, diag(\sigma))$, where $\sigma \in \mathbb{R}^d$ is the kernel width. In other words, the kernel $\mathcal{G}$ can be approximated using Monte Carlo method, denoted as $\hat{\mathcal{G}}$, as follows:

$$\hat{\mathcal{G}}(x, x') = \frac{2}{D}\sum_{i=1}^{D/2}\cos(q_i^T(x - x')) = \langle r(x), r(x')\rangle \tag{13}$$

Where $q_i \sim \mathcal{N}(\mathbf{0}, diag(\sigma))$, and $r(x) := \frac{1}{\sqrt{D/2}}[cos(q_i^T x), sin(q_i^T, x)]_{i=1}^{D/2}$. The following theorem bounds the random Fourier error in equation (9).

Theorem 8 *Given a pairwise Mercer kernel $k_{(x,x')} := k((x, x'), (\cdot, \cdot))$ defined on $\mathcal{X}^2 \times \mathcal{X}^2$. Let $\ell(w, z, z')$ be convex loss that is Lipschitz smooth with constant M. Then for any $w^* = \sum_{i,j\neq i}^{T} a_{i,j}^* k_{(x_i,x_j)}$, and random Fourier features number D we have the following with probability at least $1 - 2^8\left(\frac{\sigma R}{\delta}\right)\exp(-D\delta^2/(4d + 8))$,*

$$\sum_{t=2}^{T} L_t(\bar{w}^*) - \sum_{t=2}^{T} L_t(w^*) \leq \frac{M}{2}T\|w^*\|_1^2\delta^2 \tag{14}$$

where $\|w^*\|_1 = \sum_{i,j\neq i}^{T} |a_{i,j}^*|$. The proof is in appendix.

Remark 9 *Note that $\|w^*\|_1$ is controlled by the regularization, i.e. if there exist more than one optimal solution, then the optimal one has minimal $\|w^*\|_1$.*

5. Related Work

Pairwise scalability poses a challenge in pairwise learning due to the quadratic growth of the problem with the number of samples. To address this issue, researchers have proposed different approaches in the literature. Some examples include offline doubly stochastic mini-batch learning Dang et al. (2020); Gu et al. (2019), online buffer learning Zhao et al. (2011); Kar et al. (2013); Yang et al. (2021), second-order statistic online learning Gao et al. (2013), kernelized learning Hu et al. (2015); Ying and Zhou (2015), and saddle point-problem methods Ying et al. (2016); Reddi et al. (2016). Online gradient descent, while having a time complexity of $O(T^2)$ Boissier et al. (2016); Gao et al. (2013), is impractical for large-scale problems. It pairs a data point (x_t, y_t) received at time t with all previous samples $\{x_{t'}, y_{t'}|1 < t' < t - 1\}$ to calculate the true loss. However, computing the gradients for all received training examples, which increases linearly with t, poses a significant challenge. To address this, the work in Zhao et al. (2011) introduced two buffers, B_+ and B_-, of sizes N_+ and N_-, respectively, using Reservoir Sampling to maintain a uniform sample from the original dataset. While this approach provides a sublinear regret bound dependent on buffer sizes, it is limited to AUC maximization with linear models ($\mathcal{H} = \mathbb{R}^d$) and overlooks the effect of buffer size on generalization error. Researchers have also explored the application of saddle point-problem methods for tackling pairwise learning tasks involving metrics like

Table 2: AUC maximization results (average ± standard error)$\times 10^2$ using different batch and online algorithms on different datasets

Dataset	**FPOGD**	**SPAM-NET**	**OGD**	**S. Kernel**	**Proj++**	**Kar**
diabetes	81.91±0.48	82.03±0.32	82.53±0.31	82.64±0.37	77.92±1.44	79.85±0.28
ijcnn1	**92.32**±0.77	87.01±0.10	83.46±1.25	71.13±0.59	92.20±0.27	83.44±1.21
a9a	**90.03**±0.41	89.95±0.42	88.41±0.42	84.20±0.17	84.42±0.33	77.93±1.55
mnist	**92.98**±0.38	88.57±0.54	88.65±0.34	89.21±0.15	89.82±0.15	84.16±0.15
rcv1	**99.38**±0.20	98.13±0.15	99.05±0.57	96.26±0.35	94.54 ±0.36	97.78 ±0.64
usps	**95.02**±0.84	85.12±0.88	92.88±0.47	91.25±0.84	90.14 ±0.22	91.58±0.25
german	**85.82**±0.24	76.89±2.46	84.20±0.54	80.11±0.44	78.44 ±0.66	84.21±0.45
Reg.	l_2	$l_1 + l_2$	l_2	l_2	l_2	l_2

AUC Ying et al. (2016). By formulating the problem as a saddle point problem and utilizing typical saddle point solvers, this approach achieves a time complexity of $O(T)$ in terms of gradient computations, providing an efficient solution for pairwise learning with reduced computational requirements.

Another approach by the work Kar et al. (2013) is modified reservoir sampling (RSx), which replaces random buffer samples with the received data point using s Bernoulli processes with probability $1/t$. This ensures that s data points are $i.i.d.$ samples from the preceding stream. The authors show that the generalization bound depends on Rademacher complexity and regret, but achieving optimal generalization in terms of buffer loss requires a buffer size of $O(\sqrt{T})$, impractical for large-scale problems. Recently, a study in Yang et al. (2021) claims that a buffer size of $s = 1$, retaining only the last received sample, achieves optimal generalization. However, their proof assumes $i.i.d.$ stream of data, which is impractical in online learning. Their work focuses on linear models ($\mathcal{H} = \mathbb{R}^d$) and lacks explicit regret minimization analysis.

6. Experiments

We perform experiments on several real-world datasets, and compare our algorithm to both offline and online pairwise algorithms. Specifically, the proposed method is compared with different algorithms of AUC maximization, with the squared function as the surrogate loss.

6.1. Experimental Setup

Compared Algorithms. The compared algorithms includes offline and online setting are,

- SPAM-NET Reddi et al. (2016) is an online algorithm for AUC with square loss that is transformed into a saddle point problem with non-smooth regularization.
- OGD Yang et al. (2021), the most similar to our algorithm but with a linear model, that uses the last point every iteration.
- Sparse Kernel Kakkar et al. (2017) is an offline algorithm for AUC maximization that uses the kernel trick.
- Projection ++ Hu et al. (2015) is an online algorithm with adaptive support vector set.

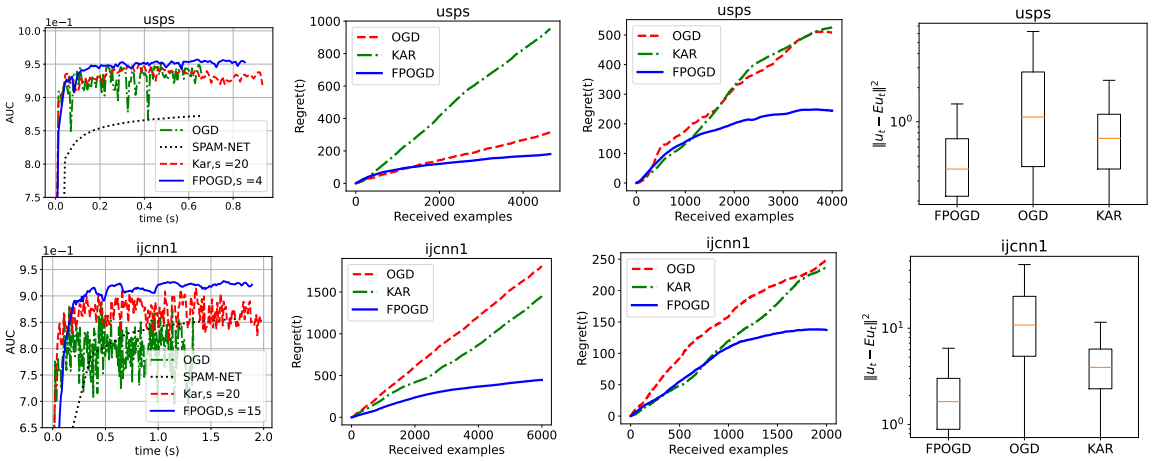

Figure 2: The first column presents AUC vs. time. Columns two and three display regret comparisons, with column two focused on i.i.d. datasets and column three on non-i.i.d. datasets. The fourth column provides insights into gradient variance analysis using a 4-size buffer, offering a glimpse into stochastic gradient behavior across buffer algorithms. Additional datasets are available in the appendix.

- Kar Kar et al. (2013) is an online algorithm with a randomized buffer update policy.

Datasets. The datasets used in this study are sourced from the LIBSVM website Chang and Lin (2011). Appendix B provides an overview of the dataset statistics, including the dataset name, size, feature dimension, and the ratio of negative to positive examples. Non-binary datasets undergo a conversion process into binary by evenly dividing the labels.

Implementation. The experiments were validated for all algorithms through a grid search on the hyperparameters, employing three-fold cross-validation. For instance, in each algorithm, the step size, denoted as η, was varied within the range of $2^{[-8:-1]}$, providing flexibility for fine-tuning. Similarly, the regularization parameters, represented by λ, were explored over the range of $10^{[-8:-1]}$. In the case of the SPAM-NET algorithm, the elastic-net regularization parameter, denoted as λ_2, was determined through a grid search with values ranging from 10^{-8} to 10^{-1}. To ensure a fair comparison, the use of kernelization is excluded when comparing with linear algorithms. All algorithms were executed five times on different folds using Python, running on a CPU with a speed of 4 GHz and 16 GB of memory.

6.2. Experimental Results and Analysis

The effectiveness of our random Fourier pairwise online gradient descent procedure in maximizing the area under the curve (AUC) is confirmed by our results obtained with a squared loss function as illustrated in figure 2. Table 2 clearly demonstrates that our algorithm outperforms both online and offline linear and nonlinear pairwise learning algorithms, yielding enhanced AUC performance particularly on large-scale datasets. Furthermore, in the appendix, we provide experimental results that demonstrate the relationship between the

number of allowed clusters and the convergence of the algorithm. Additionally, we investigate the impact of the number of random features on the algorithm's performance.

7. Conclusion

In this research paper, we introduce a lightweight online kernelized pairwise learning algorithm. Our approach involves maintaining an online clustering mechanism and utilizing it to calculate the online gradient based on the current received sample. Additionally, we approximate the kernel function using random Fourier mapping. As a result, our algorithm achieves a gradient complexity of $O(sT)$ for linear models and $O(\frac{D}{d}sT)$ for nonlinear models, where T represents the number of received examples and D denotes the number of random features.

Moreover, we establish a sublinear regret bound for our algorithm, utilizing only $O(\sqrt{T}\log T)$ random Fourier features. Furthermore, through experimental evaluations, we validate the superiority of our algorithm over both online and offline pairwise learning algorithms.

References

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017.

Martin Boissier, Siwei Lyu, Yiming Ying, and Ding-Xuan Zhou. Fast convergence of online pairwise learning algorithms. In *Artificial Intelligence and Statistics*, pages 204–212. PMLR, 2016.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Zhiyuan Dang, Xiang Li, Bin Gu, Cheng Deng, and Heng Huang. Large-scale nonlinear auc maximization via triply stochastic gradients. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Changying Du, Changde Du, Guoping Long, Qing He, and Yucheng Li. Online bayesian multiple kernel bipartite ranking. In *UAI*, 2016.

Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International conference on machine learning*, pages 906–914. PMLR, 2013.

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

Bin Gu, Zhouyuan Huo, and Heng Huang. Scalable and efficient pairwise learning to achieve statistical accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3697–3704, 2019.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

Junjie Hu, Haiqin Yang, Irwin King, Michael R Lyu, and Anthony Man-Cho So. Kernelized online imbalanced learning with fixed budgets. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Vishal Kakkar, Shirish Shevade, S Sundararajan, and Dinesh Garg. A sparse nonlinear classifier design using auc optimization. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 291–299. SIAM, 2017.

Nathan Kallus and Angela Zhou. The fairness of risk scores beyond classification: Bipartite ranking and the xauc metric. *Advances in neural information processing systems*, 32, 2019.

Bong-Nam Kang, Yonghyun Kim, and Daijin Kim. Pairwise relational networks for face recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 628–645, 2018.

Purushottam Kar, Bharath Sriperumbudur, Prateek Jain, and Harish Karnick. On the generalization ability of online learning algorithms for pairwise loss functions. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 441–449, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/kar13.html.

Brian Kulis et al. Metric learning: A survey. *Foundations and trends in machine learning*, 5 (4):287–364, 2012.

Zhu Li. Sharp analysis of random fourier features in classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7444–7452, 2022.

Junhong Lin, Yunwen Lei, Bo Zhang, and Ding-Xuan Zhou. Online pairwise learning algorithms with convex loss functions. *Information Sciences*, 406:57–70, 2017.

Michael Natole, Jr., Yiming Ying, and Siwei Lyu. Stochastic proximal algorithms for AUC maximization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3710–3719. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/natole18a.html.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

Lingxue Song, Dihong Gong, Zhifeng Li, Changsong Liu, and Wei Liu. Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–782, 2019.

Yuyang Wang, Roni Khardon, Dmitry Pechyony, and Rosie Jones. Generalization bounds for online learning algorithms with pairwise loss functions. In *Conference on Learning Theory*, pages 13–1. JMLR Workshop and Conference Proceedings, 2012.

Zhenhuan Yang, Yunwen Lei, Puyu Wang, Tianbao Yang, and Yiming Ying. Simple stochastic and online gradient descent algorithms for pairwise learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Yiming Ying and Ding-Xuan Zhou. Online pairwise learning algorithms with kernels. *arXiv preprint arXiv:1502.07229*, 2015.

Yiming Ying, Longyin Wen, and Siwei Lyu. Stochastic online auc maximization. *Advances in neural information processing systems*, 29:451–459, 2016.

Peilin Zhao, Steven CH Hoi, Rong Jin, and Tianbo YANG. Online auc maximization. *Proceedings of the 28th International Conference on Machine Learning ICML 2011:*, 2011.