

Limited Memory Online Gradient Descent for Kernelized Pairwise Learning with Dynamic Averaging

Hilal AlQuabeh¹, William de Vazelhes¹, Bin Gu^{1,2*}

¹Mohamed bin Zayed University of Artificial Intelligence, UAE

²School of Artificial Intelligence, Jilin University, China

hilal.alquabeh@mbzuai.ac.ae, wdevazelhes@gmail.com, jsgubin@gmail.com

Abstract

Pairwise learning, an important domain within machine learning, addresses loss functions defined on pairs of training examples, including those in metric learning and AUC maximization. Acknowledging the quadratic growth in computation complexity accompanying pairwise loss as the sample size grows, researchers have turned to online gradient descent (OGD) methods for enhanced scalability. Recently, an OGD algorithm emerged, employing gradient computation involving prior and most recent examples, a step that effectively reduces algorithmic complexity to $O(T)$, with T being the number of received examples. This approach, however, confines itself to linear models while assuming the independence of example arrivals. We introduce a lightweight OGD algorithm that does not require the independence of examples and generalizes to kernel pairwise learning. Our algorithm builds the gradient based on a random example and a moving average representing the past data, which results in a sub-linear regret bound with a complexity of $O(T)$. Furthermore, through the integration of $O(\sqrt{T}\log T)$ random Fourier features, the complexity of kernel calculations is effectively minimized. Several experiments with real-world datasets show that the proposed technique outperforms kernel and linear algorithms in offline and online scenarios.

Introduction

The concept of online pairwise learning has attracted significant attention due to its unique characteristics and potential applications. Unlike traditional machine learning, online pairwise learning delves into understanding connections between pairs of entities or data points, rather than isolating individual instances. This methodology holds particular relevance in domains where relative evaluations between entities hold substantial import, as seen in Area Under the ROC Curve (AUC) maximization (e.g. in unbalanced-data classification) (Zhang et al. 2016), metric learning (e.g. in image retrieval) (Kulis et al. 2012), and bipartite ranking (Du et al. 2016). Given that the loss functions in these problems require pairs of examples, a distinctive analysis is needed when compared to other machine learning problems.

At the core of online pairwise learning lies the challenge of efficiently updating models in real-time. This challenge

emerges from the cumulative influence of previous instances on the comprehension of new pairs. In this landscape, Online Gradient Descent (OGD) stands as a significant tool within the realm of online learning. OGD takes on a distinctive role within the sphere of online pairwise learning, entailing a nuanced examination of evolving loss gradients and the impact of prior examples with each incoming instance. This nuanced viewpoint results in a computational complexity translating to $O(T^2)$ of gradient computations.

OGD has garnered considerable attention in the field of online pairwise learning, leading to the development of various methodologies. These approaches include online buffer learning (Zhao et al. 2011; Kar et al. 2013), second-order statistic online learning (Gao et al. 2013), and methods rooted in solving the saddle point problem (Ying, Wen, and Lyu 2016; Reddi et al. 2016), all of which have predominantly harnessed linear models. However, exploration into non-linear models within this domain has been somewhat limited, with kernelized learning receiving minimal attention (Ying and Zhou 2015; Du et al. 2016).

A notable avenue in the realm of pairwise learning is online buffer learning, as introduced by Zhao et al. (Zhao et al. 2011). This technique employs a finite buffer alongside reservoir sampling to mitigate time complexity, reducing it to $O(sT)$, where s signifies the buffer size. By retaining a subset of data and ensuring uniform sampling within the buffer, this strategy effectively mitigates the computational load. Moreover, Yang et al. (Yang et al. 2021) achieved optimal generalization with a buffer size of $s = 1$, all while assuming the sequential data to be iid instances (e.g., stochastic environments). This accomplishment marks a significant milestone in the study of pairwise learning.

The prevailing frameworks in existing literature have predominantly concentrated on data that is linearly separable, disregarding the complexities linked with non-linear pairwise learning. Furthermore, the online buffer techniques introduced thus far have not adequately tackled the sensitivity of generalization to the non-iid data, particularly in adversarial environments, constraining their capacity to grasp the intricacies inherent in real-world scenarios.

Furthermore, the exploration of non-linear pairwise learning, particularly in the realm of kernel approximation, has been relatively limited. While non-linear methods offer enhanced expressive capability, the computational expense at-

*Corresponding author.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Algorithm	Problem	Model	Scheme	iid	Time Complexity	Space Complexity
(Natole, Ying, and Lyu 2018)	AUC	Linear	Online	Yes	$O(T)$	$O(d)$
(Ying, Wen, and Lyu 2016)	AUC	Linear	Online	Yes	$O(T)$	$O(d)$
(Zhao et al. 2011)	AUC	Linear	Online	Yes	$O(sT)$	$O(ds)$
(Gao et al. 2013)	AUC	Linear	Online	Yes	$O(T)$	$O(d^2)$
(Gu, Huo, and Huang 2019)	AUC	Linear	Offline	Yes	$O(T)$	$O(d)$
(Yang et al. 2021)	General	Linear	Online	Yes	$O(T)$	$O(d)$
(Lin et al. 2017)	General	kernel	Online	No	$O(T^2)$	$O(dT)$
(Kakkar et al. 2017)	AUC	Kernel	Offline	Yes	$O(T \log T)$	$O(dT^2)$
(Kar et al. 2013)	General	Linear	Online	Yes	$O(sT)$	$O(ds)$
AOGD	General	Kernel	Online	No	$O(\frac{D}{a}T)$	$O(D)$

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

tributed to kernel calculation – scaling as $O(T^2)$ (Lin et al. 2017; Kakkar et al. 2017) – poses hurdles to their practical applicability in terms of scalability and efficiency.

Transitioning to the examination of generalization bounds, prior research extensively investigated online pairwise gradient descent with buffers and linear models (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’s imperative to acknowledge that this bound attains optimality primarily when the buffer size s approximates $O(\sqrt{T})$. This aspect poses challenges in scenarios where a smaller buffer size is preferred.

Collectively, these vulnerabilities underscore the necessity for further research and advancement in the field of online pairwise learning. This effort should encompass addressing the limitations of iid frameworks, conducting a more comprehensive exploration of non-linear methods, and surmounting the computational obstacles tied to kernel computation. Our method expands the scope of online pairwise learning to accommodate nonlinear data by incorporating kernelization into the input space. We tackle the ramifications of non-iid data on regret by evaluating the gradient of both the moving average and a random sample drawn from the past history. Leveraging random Fourier features, we efficiently estimate the kernel while maintaining a sublinear error bound, thus achieving computational efficiency without compromising performance. Through the integration of kernelization, kernel approximation, and dynamic averaging, our approach transcends linear constraints, effectively addresses non-iid data challenges, and mitigates kernel complexity. As a result, we establish a robust online pairwise learning paradigm (as illustrated in Table 1). Our principal contributions can be summarized as follows:

- We introduce an online pairwise algorithm for non-linear models, characterized by the ability to handle non-iid data. In scenarios featuring a wide margin in the kernel space, our algorithm attains sublinear regret with a buffer size of $O(1)$.
- We confront the influence of non-iid data on regret and introduce a random Bernoulli sampling strategy to manage and enhance the regret rate.
- In the case of Gaussian kernel, we approximate the pair-

wise kernel function utilizing a mere $O(\sqrt{T} \log T)$ features, a significant reduction compared to $O(T)$ features in literature, while keeping a sublinear regret bound.

- We validate the effectiveness of our proposed methodology across a range of real-world datasets and compare its performance against state-of-the-art methods for AUC maximization.

The subsequent sections are structured as follows: Section 2 delineates the problem setting, Section 3 expounds on our proposed method, Section 4 furnishes the analysis of regret, Section 5 delves into related work, followed by Section 6 presents experimental results, and lastly, Section 7 concludes the paper.

Problem Setting

The notion of pairwise learning pertains to a specific subset $\mathcal{X} \subset \mathbb{R}^d$ and a label space $\mathcal{Y} \subset \mathbb{R}$. This concept branches into two categories, as discussed in more detail by (AlQuabeh and Abdurahimov 2022). Our focus lies in analyzing pairwise loss functions from both branches, connecting the pairwise kernel associated with pairwise hypotheses to regular kernels. This link allows us to explore the characteristics of pairwise loss functions within the regular kernel framework. In our study, we examine an algorithmic process that learns from a set of instances denoted as $z_i := (x_i, y_i) \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, where the index i ranges from 1 to T , encompassing the total number of received examples. Assuming that the function f belongs to a specific space denoted as \mathcal{H} , we observe that the pairwise loss function plays a crucial role in evaluating performance. This function is denoted as $\ell : \mathcal{H} \times \mathcal{Z}^2 \rightarrow \mathbb{R}_+$, signifying that it takes pairs of examples from the combined domain \mathcal{Z}^2 , and yields values within the positive real numbers \mathbb{R}_+ .

In online learning with pairwise losses, upon receiving a new data point z_t , a localized error is used as a performance measure (Zhang et al. 2016; Boissier et al. 2016). This error represents the pairing of the new data point with all preceding $t-1$ points. The properties of this localized error depend on the selected pairwise loss function, illustrated as:

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

where $w \in \mathcal{H}$ represents a linear function mapping, i.e. $f_w(x) = \langle w, x \rangle$. To tackle the memory constraints, (Zhao et al. 2011) introduced a buffer-based mechanism for local error computation denoted as $\bar{L}_t(w_{t-1})$, as explicated in Equation 2. At each iteration t , the buffer, designated as B_t , accommodates a finite set of historical-examples indices, and the size of this buffer is $|B_t|$ (or s in literature).

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

This buffer has an important role in the learning journey, its contents updated at each step through diverse strategies that span from randomized methodologies such as reservoir sampling (Zhao et al. 2011; Kar et al. 2013) to non-randomized approaches exemplified by FIFO (Yang et al. 2021). However, despite the wide adoption of these sampling techniques, there exists a discernible research void in understanding their implications within the context of non-iid online examples. It’s imperative to highlight that the issue of buffer size and its update methods becomes pronounced when dealing with adversarial environment scenarios. The choice of buffer size and sampling strategy can significantly impact the model’s performance and ability to generalize in the case of non-iid data streams.

In our pursuit of addressing the intricacies inherent in complex real-world data, our online pairwise approach takes into consideration the mapping of both the hypothesis and the data to a Reproducing Kernel Hilbert Space (RKHS), denoted as \mathcal{H} . The associated Mercer pairwise kernel function, denoted as $k : \mathcal{X}^2 \mapsto \mathbb{R}$, adheres to the reproducing property $\langle k_{(x,x')}, g \rangle = g(x, x')$, where $x, x' \in \mathcal{X}^2$ and $g \in \mathcal{H}$. Notably, for scenarios involving pointwise hypotheses with pairwise losses, such as AUC loss, the kernel function is streamlined to $k : \mathcal{X}^2 \mapsto \mathbb{R}$. The expanse of \mathcal{H} encompasses a spectrum of linear combinations encompassing functional mappings $k_{(x,x')}(x, x') \in \mathcal{X}^2$ and their limit points.

In our quest to circumvent the computational intricacies associated with kernelization in the online realm, we employ the resourceful apparatus of random Fourier features (RFF) as a pragmatic approximation for the Mercer kernel function. RFF introduces a lower-dimensional mapping $r(\cdot)$, serving as an approximation of the kernel function, denoted as $\bar{k}(\cdot)$. This approximation empowers us to execute computations through linear operations, culminating in a significant reduction in computational overhead. The domain defined by the novel kernel functions finds its place in the realm of \mathcal{H} . It’s worth mentioning that earlier investigations have explored the precision of random Fourier approximation within pointwise and offline settings. As we navigate the online landscape, the minimum number of random features necessary to ensure sublinear regret has been ascertained to be $O(T)$. Our method introduces a novel error bound for pairwise predicaments, hinging on the utilization of merely $O(\sqrt{T} \log T)$ random features (elucidated in Section 5 for comprehensive details).

Assumptions

Prior to unveiling our principal theorems, we lay down a foundation comprising a collection of widely embraced assumptions pertaining to the attributes of both the loss function and kernels.

Assumption 1 (Lipschitz Continuity). *Assume for any $z, z' \in \mathcal{Z}$, the loss function $\ell(\cdot, z, z')$ is G -Lipschitz continuous, i.e. $\forall w, w' \in \mathcal{H}$,*

$$|\ell(w, z, z') - \ell(w', z, z')| \leq G \|w - w'\|_2.$$

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

$$\begin{aligned} \ell(w) &\geq \ell(w') + \nabla \ell(w')^T (w - w') \\ &\quad + (2M)^{-1} \|\nabla \ell(w) - \nabla \ell(w')\|^2 \end{aligned}$$

Assumption 3 (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 4 (Finite Kernel). *Assume for any ρ -probability measure on \mathcal{X}^2 the positive kernel function $k : \mathcal{X}^2 \times \mathcal{X}^2 \rightarrow \mathbb{R}$ is ρ -integrable, i.e. for any $(x, x') \in \mathcal{X}^2$,*

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

Methodology

In light of the core problem we aim to address, namely the computational complexity arising from the necessity to accommodate all preceding examples in the local error computation, our proposed methodology centers around mitigating this challenge. We begin by suggesting the utilization of a moving average of previous examples, thus alleviating the burden of handling the complete historical dataset.

However, it’s imperative to note that the simple average approach may lead to misleading outcomes, especially when the data exhibits high variance distributions. Recognizing this potential limitation, we introduce a corrective measure to enhance the reliability of the average-based gradient. This correction mechanism involves introducing a random point from the historical loss, enhancing the robustness of the gradient estimation, and ameliorating the impact of potential average imbalances. This comprehensive strategy effectively tackles the issues associated with handling the entire historical dataset, while also accounting for data distribution complexities that can skew outcomes, and most importantly not requiring the iid assumption. Another parallel research addressing the high variance in the buffer has been proposed recently (AlQuabeh, Mukhoty, and Gu 2023), however with $O(s)$ complexity.

Indeed, the loss function we seek to minimize can be formulated as a linear combination of two distinct losses: one based on the moving average and the other on a randomly chosen example. However, it’s important to recognize that the minimization of this combined loss doesn’t inherently guarantee the minimization of the true local loss. This discrepancy arises because, at the optimal solution where the

gradient is zero, the gradients of both components remain nonzero, depending upon the linear coefficient constant.

To navigate this challenge, an alternating strategy to minimize both losses is studied. In essence, we proceed by iteratively taking steps using the average-based gradient and then employing the random gradient on the transition model derived from the previous step. This alternating optimization strategy accounts for the intricate balance between the two gradient components and fosters a dynamic convergence process. Moreover, the iterative method ensures that both the average-based and random-based gradients are appropriately accounted for in the optimization process. The loss minimization upon receiving new examples z_t can be formulated using the average $\bar{z} = (\sum_{i=1}^{t-1} \frac{x_i}{t-1}, y_{t-1} | y_{t-1} \neq y_t)$ and the random sample \hat{z} from $\{z_1, \dots, z_{t-1}\}$, as follow;

$$\min_w \ell(w - \eta_t \nabla \ell(w, z_t, \bar{z}), z_t, \hat{z}) \quad (3)$$

where η_t is the step size. As detailed in the subsequent section, the magnitude of the step taken in the second update is contingent upon both the variance of the data and the smoothness of the loss function. Interestingly, when the variance is negligible, the step taken based on the average gradient is deemed adequate, rendering the random step unnecessary. The objective function can give more sense if we use second-order Taylor expansion after denoting $\hat{\ell}(\cdot) = \ell(\cdot, z_t, \hat{z})$ and $\bar{\ell}(\cdot) = \ell(\cdot, z_t, \bar{z})$ as follow,

$$\begin{aligned} \ell(w - \eta_t \nabla \ell(w, z_t, \bar{z}), z_t, \hat{z}) &\approx \hat{\ell}(w) - \eta_t \nabla \hat{\ell}(w)^T \nabla \bar{\ell}(w) \\ &\quad + \eta_t^2 \nabla \bar{\ell}(w)^T \nabla^2 \hat{\ell}(w) \nabla \bar{\ell}(w) \end{aligned} \quad (4)$$

Hence, the process of minimizing the objective function involves addressing three integral components. First, there is the endeavor to minimize the loss associated with randomly selected examples. Then, leveraging the convex nature of the loss function and the positive definiteness of its Hessian, the third term—strictly non-negative in nature—ensures that the gradient of the average-based loss is minimized. Of paramount significance is the role of the second term, which guarantees that the gradients of losses, both average-based and random-based, exhibit a maximum inner product. This intricate interplay is pivotal in correcting the moving average gradient, particularly in scenarios involving highly skewed distributions.

In order to handle the intricate patterns of real-world data, our approach to online pairwise learning involves a fundamental assumption: both the hypotheses and the data are transformed into a specialized space known as a Hilbert space, denoted as \mathcal{H} . This space is equipped with a specific type of kernel called a Mercer kernel. Essentially, this kernel is a mathematical function that captures the relationships between data points. This Hilbert space has two key attributes: first, it can reproduce the values of the kernel function, and second, it encompasses various combinations of these kernel functions along with their complete structures.

When we apply traditional buffer-based pairwise algorithms in an online setting, the process of kernelization comes with a complexity that grows as $O(T^2 s)$, where s represents the size of the buffer—a rather unwieldy task

Algorithm 1: Average Online Gradient Descent (AOGD)

Require: Random feature size D , initial solution $w_1 \in \bar{\mathcal{H}}$, initial average and random example $\bar{z} = \hat{z} \in R^D$, kernel function k and its Fourier transform $P(u)$, probability p , step sizes η_t, γ_t and λ .

- 1: Sample Fourier feature $\{u_i\}_{i=1}^{D/2} \sim P(u_i)$
- 2: **for** $t = 2, \dots, T$ **do**
- 3: Receive a new example $z_t = (x_t, y_t) \in \mathcal{Z}$
- 4: Map to the new space:

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

- 5: Suffer loss $\bar{\ell}(w_{t-1}, z_t, \bar{z})$
- 6: Update model $w'_t = w_{t-1} - \eta_t \nabla \bar{\ell}(w_{t-1}, z_t, \bar{z})$
- 7: Suffer loss $\hat{\ell}(w'_t, z_t, \hat{z})$
- 8: Update model $w_t = w'_t - \gamma_t \nabla \hat{\ell}(w'_t, z_t, \hat{z})$
- 9: Update the moving average and the random example,

$$\begin{aligned} \bar{z} &\leftarrow \frac{z * (t-1) + z_t}{t} \\ \hat{z} &\leftarrow z_t \text{ if Bernoulli}(p) = 1 \end{aligned}$$

- 10: **end for**
 - 11: **return** w_T
-

for larger problems. To address this challenge, we introduce a method that approximates the kernel function using a simpler approach. We assume the existence of a lower-dimensional mapping, represented as $r : \mathcal{X}^2 \mapsto R^D$. This mapping enables us to simplify the kernel function's calculations, significantly reducing the complexity of the process. This simplified approach is achieved using what are known as random Fourier features, which are derived from the mathematical Fourier transform. Although these techniques have been studied in previous works, our focus is on adapting them to the online setting, and we establish that as few as $O(\sqrt{T} \log T)$ random features are enough to achieve effective results.

Our method is introduced in algorithm 1 for general pairwise learning, and it includes the Gaussian kernel approximation using random Fourier features. In the upcoming section, we delve into the performance of this online algorithm and dissect its regret into two distinct components: the first is a direct result of learning a model from an online stream of examples, and the second arises from the approximation of the kernel function.

Regret Analysis

The regret incurred by the online algorithm relative to the optimal hypothesis in the original space \mathcal{H} , denoted as $w^* = \arg \min_{w \in \mathcal{H}} \sum_{t=2}^T L_t(w)$, when applied to a sequence of T examples, is formally expressed as:

$$\mathcal{R}_{w^*, T} = \sum_{t=2}^T L_t(w_{t-1}) - \sum_{t=2}^T L_t(w^*), \quad (5)$$

In the context of this equation, the local all-pairs loss $L_t(\cdot)$ is defined according to Equation 1. To delve deeper, we can decompose the regret represented in Equation 5 by introducing the best-in-class hypothesis within the approximated space $\bar{\mathcal{H}}$, denoted as $\bar{w}^* = \arg \min_{w \in \bar{\mathcal{H}}} \sum_{t=2}^T L_t(w)$. This decomposition takes the form:

$$\begin{aligned} \mathcal{R}_{w^*, T} &= \underbrace{\sum_{t=2}^T L_t(w_{t-1}) - \sum_{t=2}^T L_t(\bar{w}^*)}_{T_1} \\ &\quad + \underbrace{\sum_{t=2}^T L_t(\bar{w}^*) - \sum_{t=2}^T L_t(w^*)}_{T_2} \end{aligned} \quad (6)$$

In our analysis, we provide a bound for T_1 in Theorem 2, followed by an error bound for T_2 in Theorem 3. By combining these bounds, we derive the regret bound for Algorithm 1 as follows:

Theorem 1. Let $\{z_t \in \mathcal{Z}\}_{t=1}^T$ be sequentially accessed by Algorithm 1. Let D denote the number of random Fourier features in the kernel mapping from the original space $\mathcal{X} \subset \mathbb{R}^d$. Let η be first step size, $\gamma = O(\Gamma M_t \eta)$ the second step size, and G the Lipschitz constant. Then the regret bound compared to $w^* = \arg \min_{w \in \mathcal{H}} \sum_{t=2}^T L_t(w)$ is bounded with a probability of at least $1 - 2^8 \left(\frac{\sigma \text{Diam}(\mathcal{X})}{\epsilon} \right) \exp(-D\epsilon^2/(4d + 8))$, as follows:

$$\mathcal{R}_{w, T} \leq GT \|w^*\|_1 \epsilon + \frac{\|\bar{w}^*\|^2}{2\eta} + 2\gamma G^2 T \quad (7)$$

where, ϵ denotes the kernel approximation error, σ signifies the kernel width, $\Gamma \geq \text{tr}(\text{cov}[z_t])$, and $M_t : M(w_t)$ stands for the smoothness parameter of ℓ in the neighborhood of w_t .

Remark 1. Selecting $\eta = \frac{1}{\sqrt{T}}$ and $\epsilon = \frac{1}{\sqrt{T}}$ results in a sublinear bound, which is optimal. Notably, $D = O(T)$ in general, yet it can be as low as $\sqrt{T} \log T$ for specific kernels. Furthermore, since $\gamma = O(\Gamma M_t \eta)$, the regret depends on the variance of data and the smoothness of the loss.

Regret in the Approximated Space $\bar{\mathcal{H}}$

Prior approaches (e.g., (Kar et al. 2013), (Zhao et al. 2011)) aimed to attain uniform convergence of buffer loss often relied on the assumption of iid examples. These methods also require maintaining uniform samples from historical data through a Bernoulli process with diminishing probabilities, ultimately approaching zero. Moreover, achieving such uniform convergence necessitated a disentanglement between the model and the buffer to ensure the validity of expectation. For instance, while (Zhao et al. 2011) employed Reservoir sampling to enforce an *iid* buffer, they overlooked the interplay between the model and the data. Similarly, (Kar et al. 2013) utilized modified reservoir sampling and Symmetrization of Expectations to address these challenges. However, their approach demanded a buffer size of $O(\sqrt{T})$,

which proves impractical for larger-scale scenarios. In contrast, (Yang et al. 2021) accomplished sublinear regret with a buffer size of one, focusing on local loss. Nevertheless, their assumption of a fully *iid* training set does not align with real-world online situations. Our forthcoming theorem obviates the necessity for the *iid* assumption by harnessing both moving averages and individual random points.

As previously explained, in each iteration, a single moving average is employed to ascertain the gradient direction, which establishes a connection to the all-pairs loss without relying on the *iid* assumption. This is then followed by a step involving the gradient of a random example. Notably, this sample should not be uniformly chosen from historical data. If we denote the moving average based loss as $\bar{L}_t(w_{t-1})$, then using the convexity of $\ell(\cdot)$ and the linearity of models in \mathcal{H} , we have $\bar{L}_t(w_{t-1}) \leq L(w_{t-1})$ or $\bar{L}_t(w_{t-1}) = L(w_{t-1}) - g(t)$ where $g(t) \geq 0$ is Jensen's gap. The gap is first bounded in lemma 1 using the smoothness constant and the variance of the data.

Lemma 1. Given that the data $\{z_1, \dots, z_t\}$ is linearly separable and have bounded total variance, i.e. $\Gamma \geq \text{tr}(\text{cov}[z_t])$ and a loss function with $M_t : M(w_t)$ smoothness parameters, the Jensen's gap can be bounded as follow,

$$g(t) \leq \frac{1}{2} \Gamma M_t \quad (8)$$

Note that since the model is linear in the space $\bar{\mathcal{H}}$, the smoothness assumption holds for loss gradient w.r.t. z . Therefore we can provide the following bound in theorem 2.

Theorem 2. Assume that assumptions 1,2, and 3 hold. Let $[w_t]_{t=1}^T$ be the sequence of models returned by running the algorithm 1 for T times using the online sequence of data. Then if $\bar{w}^* = \arg \min_{w \in \bar{\mathcal{H}}} \sum_{t=2}^T L_t(w)$, we have,

$$\sum_{t=2}^T L_t(w_{t-1}) \leq \sum_{t=2}^T L_t(\bar{w}^*) + \frac{\|\bar{w}^*\|^2}{2\eta} + 2\gamma G^2 T \quad (9)$$

Note that $\Gamma \geq \text{tr}(\text{cov}[z_t])$ where $\text{cov}[z_t]$ is the covariance of the received examples, and M_t is the smoothness parameter of the loss function.

Approximation Error for Random Features

The kernel associated with 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,

$$\begin{aligned} k(x_1, x_2, x'_1, x'_2) &= \mathcal{G}(x_1, x'_1) + \mathcal{G}(x_2, x'_2) \\ &\quad - \mathcal{G}(x_1, x'_2) - \mathcal{G}(x_2, x'_1) = \langle \mathcal{G}_{x_1} - \mathcal{G}_{x_2}, \mathcal{G}_{x'_1} - \mathcal{G}_{x'_2} \rangle_{\mathcal{G}} \end{aligned} \quad (10)$$

The pairwise kernel k defined earlier demonstrates positive semi-definiteness over \mathcal{X}^2 . This property aligns with the attributes of a Mercer kernel, as indicated in references like (Ying and Zhou 2015), when \mathcal{G} exhibits such characteristics over \mathcal{X} . We further assume there exist a lower dimensional mapping $r : \mathcal{X} \mapsto \mathbb{R}^D$, such that, $\mathcal{G}_x(\cdot) \approx r(x)^T r(\cdot)$

The literature extensively explores the accuracy of approximating the pointwise kernel \mathcal{G} using random Fourier

features, as discussed in references such as (Rahimi and Recht 2007), (Bach 2017), and (Li 2022). For example if the kernel is the Gaussian kernel, i.e., $\mathcal{G}(x - x') = \frac{1}{\sqrt{2\pi d}} \exp -(x - x')^T [\text{diag}(\sigma)](x - x')$, we have $p(u)$ from Fourier transform to be $p(u) \propto \mathcal{N}(\mathbf{0}, \text{diag}(\sigma))$ where $\sigma \in \mathbb{R}^d$ is the kernel width. Thus kernel \mathcal{G} can be given by a Monte Carlo estimate as follow,

$$\hat{\mathcal{G}}(x, x') \approx \frac{2}{D} \sum_{i=1}^{D/2} \cos(u_i^T(x - x')) = \langle r(x), r(x') \rangle \quad (11)$$

Where $u_i \sim \mathcal{N}(0, \text{diag}(\sigma))$, and $r(x) := \frac{1}{\sqrt{D/2}} [\cos(u_i^T x), \sin(u_i^T x)]_{i=1}^{D/2}$. The quality of the estimation is given by (Rahimi and Recht 2007) to be dependent on the number of random features D used to approximate the Fourier transform, which can be extended to pairwise kernels as illustrated in the following corollary.

Corollary 1. *Given $x_1, x_2, x'_1, x'_2 \in \mathcal{X}$, and pairwise kernel k defined on $\mathcal{X}^2 \times \mathcal{X}^2$, the random Fourier estimation of the kernel has mistake bounded with probability at least $1 - 8 \exp \frac{-D^2 \epsilon}{2}$ as follow,*

$$|\hat{k}_{(x_1, x_2)}(x'_1, x'_2) - k_{(x_1, x_2)}(x'_1, x'_2)| \leq \epsilon$$

The following theorem bounds the random Fourier error in equation (6) using the results of corollary 1.

Theorem 3. *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 a convex loss that is Lipschits continuous with constant G . 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 $1 - 2^8 \left(\frac{\sigma \text{Diam}(\mathcal{X})}{\epsilon} \right) \exp(-D\epsilon^2/(4d + 8))$,*

$$\sum_{t=2}^T L_t(\bar{w}^*) - \sum_{t=2}^T L_t(w^*) \leq GT \|w^*\|_1 \epsilon \quad (12)$$

where $\|w^*\|_1 = \sum_{i, j \neq i}^T |a_{i, j}^*|$.

Remark 2. *A trade-off between $\|w^*\|_1$ and ϵ exists when independent support vectors are in the kernel space ($|a_{i, j}^*| > 0$). In such cases, a larger error requires a smaller ϵ , implying a larger D . Regularization can set insignificant support vectors' $a_{i, j}^*$ to zero, a subject for potential future research.*

Related Work

While online gradient descent exhibits a time complexity of $O(T^2)$ (Boissier et al. 2016; Kar et al. 2013; Gao et al. 2013), this remains impractical for large-scale problems. For example, in AUC maximization, pairing a data point $x_t, y_t = 1$ with all preceding $x_{t'}, y_{t'} = -1 |1 < t' < t - 1$ to ascertain the true loss poses a key challenge. This approach necessitates computing gradients $\nabla L_t(f_t - 1)$ for all received training examples, growing linearly with t .

Buffer-based methods have been explored in the literature to manage this challenge. (Zhao et al. 2011) introduced buffers B_+ and B_- , each of size N_+ and N_- , considering

only points within the buffers at each step t . Reservoir Sampling updates the buffers while ensuring uniform sampling. Theorem 1 in (Zhao et al. 2011) establishes sublinear regret linked to buffer sizes. However, this work is limited to AUC with linear models and does not explicitly address the buffer size's impact on generalization error.

In contrast, (Kar et al. 2013) proposed modified reservoir sampling (RSx) by performing s Bernoulli processes with probability $1/t$ to replace each buffer point with the received data point. This approach ensures s data points are iid samples from the stream and independent from w , with the generalization bound depending on Rademacher complexity and regret. Nevertheless, optimal generalization requires $s = O(\sqrt{T})$ for buffer loss.

Another parallel research addressing the high variance in the buffer has been proposed recently (AlQuabeh, Mukhoty, and Gu 2023), coinciding with the publication of our paper. Recently, (Yang et al. 2021) suggested optimal generalization with a buffer size of $s = 1$, assuming *iid* data and independent buffer-data relations. Their work aligns with ours, but we remove the *iid* assumption for convergence proof and consider non-linear model spaces.

Experiments

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

Experimental Setup

Compared Algorithms. The compared algorithms including offline and online settings, as illustrated below,

- SPAM-NET (Reddi et al. 2016) is an online algorithm for AUC with square loss that is converted to 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 an adaptive support vector set.
- Kar (Kar et al. 2013) is online algorithm with buffer of size s .

Implementation. The datasets of investigation are available on LIBSVM website (Chang and Lin 2011). The experiments are validated for all algorithms by performing a grid search on the hyperparameters with three-fold cross-validation. For instance, in all algorithms the step size $\eta \in 2^{[-8: -1]}$, and the regularization parameters $\lambda \in 10^{[-8: -1]}$. All the algorithms have been run five times on different folds on a CPU of 4 GHz speed and 16 GB memory.

Dataset	AOGD	SPAM-NET	OGD	Sparse Kernel	Projection++	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	89.46±1.22	71.13±0.59	92.20±0.27	83.44±1.21
a9a	90.03±0.41	89.95±0.42	90.01±0.42	84.20±0.17	84.42±0.33	77.93±1.55
mnist	96.98±0.38	96.57±0.54	89.56±0.34	95.21±0.15	93.82±0.15	89.16±0.14
rcv1	99.38±0.48	98.13±0.15	99.05±0.57	96.26±0.35	94.54±0.36	97.78±0.64
usps	92.76±0.48	89.12±0.88	89.88±0.47	91.25±0.84	90.14±0.22	89.58±0.25
german	80.75±0.48	78.55±2.75	79.25±0.42	80.11±0.44	78.44±0.66	77.24±0.52
Reg.	l_2	$l_1 + l_2$	l_2	l_2	l_2	l_2

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

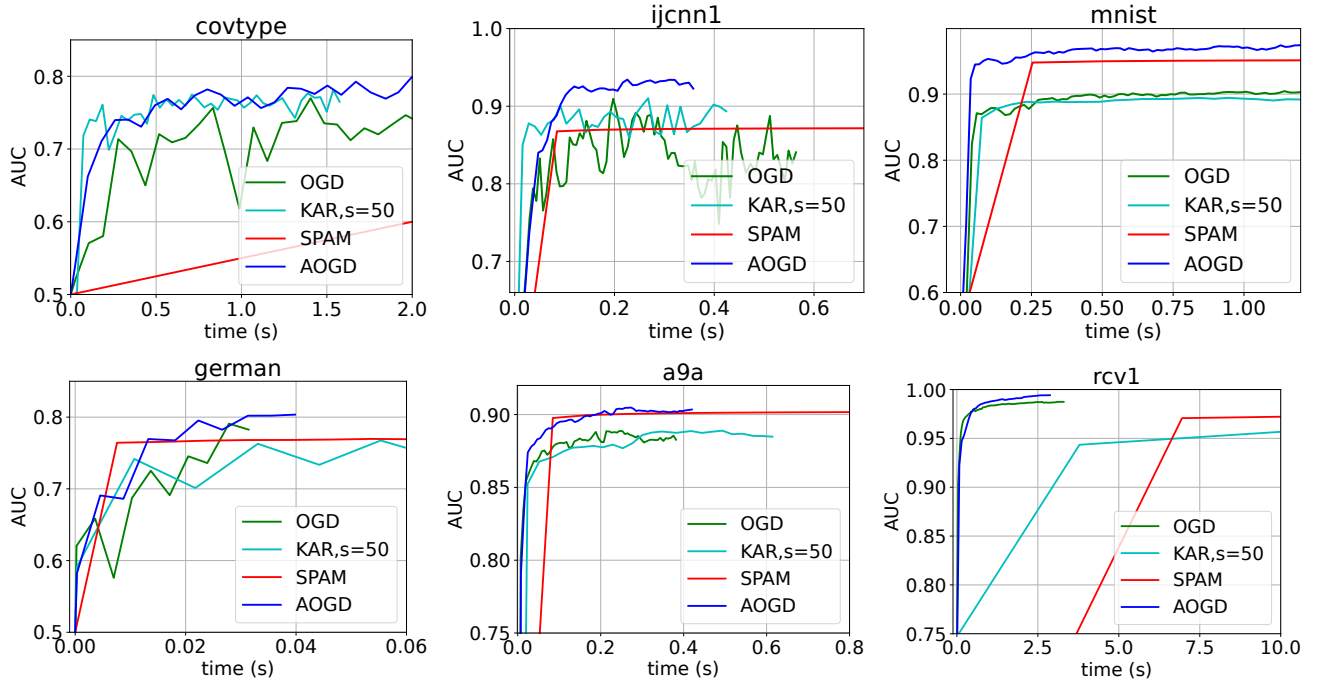


Figure 1: The AUC vs. time comparison of the algorithms in different datasets showing superior performance of the proposed method.

Experimental Results and Analysis

Our results on maximizing the area under the curve (AUC) using a squared loss function verify the efficacy of our random Fourier pairwise online gradient descent procedure. Table 2 shows that when compared to online and offline, linear and nonlinear pairwise learning algorithms, our algorithm achieves improved AUC performance on large-scale datasets.

Figure 1 displays the AUC performance relative to CPU utilization. Results of ijcnn1 and mnist show that compared to other state-of-the-art online pairwise learning methods, our algorithm not only has better AUC performance but also has a faster convergence rate. As compared to the two most popular kernel algorithms, Sparse kernel, and Projection++, our technique provides superior AUC performance across the board. Projection++ is impractical for large-scale datasets because of its hyper-parameters and the requirement to regularly update a support vector set.

Conclusion

In summary, our work introduces a lightweight online kernelized pairwise learning algorithm applicable to both linear and non-linear models. The algorithm efficiently computes the online gradient using a moving average and random examples, with the kernel function approximated through random Fourier mapping. Achieving a gradient complexity of $O(T)$ for linear models and $O(\frac{D}{d}T)$ for nonlinear models, our approach showcases computational efficiency. The incorporation of multiple moving averages, a potential avenue for future research, could further enhance adaptability and robustness. Notably, we address the challenges of non-iid data by dual evaluation of gradients, leveraging random Fourier features for efficient kernel estimation with a sublinear error bound. This integration of kernelization, approximation, and dynamic averaging transcends linear constraints, effectively tackling non-iid data challenges and mitigating kernel complexity.

References

- AlQuabeh, H.; and Abdurahimov, A. 2022. Pairwise Learning via Stagewise Training in Proximal Setting. *arXiv preprint arXiv:2208.04075*.
- AlQuabeh, H.; Mukhoty, B.; and Gu, B. 2023. Variance Reduced Online Gradient Descent for Kernelized Pairwise Learning with Limited Memory. *arXiv preprint arXiv:2310.06483*.
- Bach, F. 2017. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1): 714–751.
- Boissier, M.; Lyu, S.; Ying, Y.; and Zhou, D.-X. 2016. Fast convergence of online pairwise learning algorithms. In *Artificial Intelligence and Statistics*, 204–212. PMLR.
- Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2: 27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Du, C.; Du, C.; Long, G.; He, Q.; and Li, Y. 2016. Online Bayesian Multiple Kernel Bipartite Ranking. In *UAI*.
- Gao, W.; Jin, R.; Zhu, S.; and Zhou, Z.-H. 2013. One-pass AUC optimization. In *International conference on machine learning*, 906–914. PMLR.
- Gu, B.; Huo, Z.; and Huang, H. 2019. Scalable and efficient pairwise learning to achieve statistical accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3697–3704.
- Hu, J.; Yang, H.; King, I.; Lyu, M. R.; and So, A. M.-C. 2015. Kernelized online imbalanced learning with fixed budgets. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Kakkar, V.; Shevade, S.; Sundararajan, S.; and Garg, D. 2017. A sparse nonlinear classifier design using AUC optimization. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 291–299. SIAM.
- Kar, P.; Sriperumbudur, B.; Jain, P.; and Karnick, H. 2013. On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 441–449. Atlanta, Georgia, USA: PMLR.
- Kulis, B.; et al. 2012. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4): 287–364.
- Li, Z. 2022. Sharp Analysis of Random Fourier Features in Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7444–7452.
- Lin, J.; Lei, Y.; Zhang, B.; and Zhou, D.-X. 2017. Online pairwise learning algorithms with convex loss functions. *Information Sciences*, 406: 57–70.
- Natole, M., Jr.; Ying, Y.; and Lyu, S. 2018. Stochastic Proximal Algorithms for AUC Maximization. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3710–3719. PMLR.
- Rahimi, A.; and Recht, B. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.
- Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, 314–323. PMLR.
- Wang, Y.; Khardon, R.; Pechyony, D.; and Jones, R. 2012. Generalization bounds for online learning algorithms with pairwise loss functions. In *Conference on Learning Theory*, 13–1. JMLR Workshop and Conference Proceedings.
- Yang, Z.; Lei, Y.; Wang, P.; Yang, T.; and Ying, Y. 2021. Simple Stochastic and Online Gradient Descent Algorithms for Pairwise Learning. *Advances in Neural Information Processing Systems*, 34.
- Ying, Y.; Wen, L.; and Lyu, S. 2016. Stochastic online AUC maximization. *Advances in neural information processing systems*, 29: 451–459.
- Ying, Y.; and Zhou, D.-X. 2015. Online pairwise learning algorithms with kernels. *arXiv preprint arXiv:1502.07229*.
- Zhang, Z.; Shi, Q.; McAuley, J.; Wei, W.; Zhang, Y.; and Van Den Hengel, A. 2016. Pairwise matching through max-weight bipartite belief propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1202–1210.
- Zhao, P.; Hoi, S. C.; Jin, R.; and YANG, T. 2011. Online AUC maximization. *Proceedings of the 28th International Conference on Machine Learning ICML 2011*.