Fast Convergence of Online Pairwise Learning Algorithms

Martin Boissier[†] **Siwei Lyu**[‡] **Yiming Ying**[§] **Ding-Xuan Zhou**[†]

[†]Department of Mathematics
City University of Hong Kong
Hong Kong, China

[‡]Department of Computer Science
SUNY at Albany
Albany, NY 12222, USA

[‡]Department of Mathematics
and Statistics, SUNY at Albany
Albany, NY 12222, USA

Abstract

Pairwise learning usually refers to a learning task which involves a loss function depending on pairs of examples, among which most notable ones are bipartite ranking, metric learning and AUC maximization. In this paper, we focus on online learning algorithms for pairwise learning problems without strong convexity, for which all previously known algorithms achieve a convergence rate of $\mathcal{O}(1/\sqrt{T})$ after T iterations. In particular, we study an online learning algorithm for pairwise learning with a least-square loss function in an unconstrained setting. We prove that the convergence of its last iterate can converge to the desired minimizer at a rate arbitrarily close to $\mathcal{O}(1/T)$ up to logarithmic factor. The rates for this algorithm are established in high probability under the assumptions of polynomially decaying step sizes.

1 INTRODUCTION

This paper is concerned with an important family of learning problems that, for simplicity, we refer to as *pairwise learning*. In contrast to regression and classification, such learning problems involve pairwise loss functions, i.e. the loss function depends on a pair of examples which can be expressed by $\ell(f,(x,y),(x',y'))$ for a hypothesis function $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Many machine learning tasks can be formulated as pairwise learning problems. For instance, bipartite ranking [1, 6, 14] is to correctly predict the ordering of pairs of binary labeled samples, which can be formulated as a pairwise learning problem. It generally involves the use of a misranking loss $\ell(f,(x,y),(x',y')) = \mathbb{I}_{\{(y-y')f(x,x')<0\}}$ or its surrogate loss $\ell(f,(x,y),(x',y')) = (1-(y-y')f(x,x'))^2$, where $\mathbb{I}(\cdot)$ is the indicator function. Apart from bipartite ranking, many other learning tasks fit this pairwise learning framework well, such as metric learning [5, 19, 20, 21] and AUC maximization [7, 9, 24].

In practice, pairwise learning usually involves pairs of training samples that are not independently and identically distributed (i.i.d.). Consequently, standard generalization analysis techniques do not apply to these algorithms. Generalization analysis for pairwise learning algorithms in the batch learning setting has been conducted relying on U-statistics [3, 6, 14] and algorithmic stability [1]. The algorithmic challenge in pairwise learning is the big volume of data in the sense that the number of pairs of examples grows quadratically in the number of examples. Specifically, if we have n examples, then we have n^2 possible pairs of examples. Online learning algorithms are scalable to large scale datasets and have been well studied theoretically in classification and regression, see e.g. [2, 4, 8, 13, 15, 16, 17, 22, 25]. However, there is relatively little work on generalization analysis for online learning algorithms for pairwise learning, in spite of their capability of dealing with large scale datasets. Wang et al. [18] established the first generalization analysis of online learning methods for pairwise learning. In particular, they proved online-to-batch conversion bounds for online learning methods, which are combined with regret bounds to obtain generalization error bounds. This is in the same spirit as the results in [4] for online learning algorithms in classification and regression. Kar et al. [9] derived tighter bounds than those in [18] using an extension of Rademacher complexities instead of covering numbers. Such results are based on the assumption of a uniformly bounded loss function with a rate $\mathcal{O}(1/\sqrt{T})$ in the general convex case and $\mathcal{O}(1/T)$ if, moreover, the loss function is strongly convex.

In this paper we focus on online learning algorithms

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

for pairwise learning without strong convexity. In particular, we study an online pairwise learning algorithm with a least-square loss function in an unconstrained setting. We prove that the convergence of its last iterate can converge to the desired minimizer at a rate arbitrarily close to $\mathcal{O}(\log^2 T/T)$. The rates for this algorithm are established in high probability under the assumptions of polynomially decaying step sizes. In contrast with previous work [9, 18], the algorithm does not require the loss function to be strongly convex nor the the loss function is uniformly bounded.

Apart from the direct implementation of online pairwise algorithms leading to the usual $\mathcal{O}(T^2 d)$ complexity where d is the dimensionality of the data, we introduce an efficient algorithm in $\mathcal{O}(Td^2)$ time and $\mathcal{O}(d^2)$ space complexity. This algorithm directly benefits applications with large amount of data and can be used to tackle cases where the volume of streaming data is such that it cannot be entirely saved in memory beforehand and has to be processed in real time.

The paper is organized as follows. Section 2 illustrates the main result and discusses the related work. Section 3 proves the main result, Section 4 presents experimental results and Section 5 concludes the paper.

2 MAIN RESULTS

Let samples $\mathbf{z} = \{(x_i, y_i),\ i = 1, \ldots, T\}$ be drawn i.i.d. from an unknown distribution ρ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where is $\mathcal{X}$ is compact domain of $\mathbb{R}^d$ and $\mathcal{Y} \subseteq [-M, M]$ with a constant $M > 0$.

For any $\mathbf{w} \in \mathbb{R}^d$, given the pairwise least-square loss $\ell(\mathbf{w}, (x, y), (x', y')) = (\mathbf{w}^\top (x - x') - y + y')^2$, we are interested in solving the expected risk minimization problem, i.e.

$$\inf_{\mathbf{w} \in \mathbb{R}^d} \mathcal{E}(\mathbf{w}), \quad \text{where } \mathcal{E}(\mathbf{w}) = \iint_{\mathcal{Z} \times \mathcal{Z}} (\mathbf{w}^\top (x - x') - y + y')^2 d\rho(x, y) d\rho(x', y').$$

This paper considers the following online learning algorithm: $\mathbf{w}_1 = \mathbf{w}_2 = 0$ and, for $2 \leq t \leq T$,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \gamma_t \Big[\frac{1}{t-1} \sum_{j=1}^{t-1} (\mathbf{w}_t^\top (x_t - x_j) - y_t + y_j)(x_t - x_j)\Big], \quad (1)$$

where $\{\gamma_t : t \in \mathbb{N}\}$ is a sequence of step sizes. The above algorithm is an online learning algorithm as it only needs a sequential access to the training data. Specifically, at each time step $t + 1$, the above algorithm presumes a hypothesis $\mathbf{w}_t$ upon which a new data $z_t = (x_t, y_t)$ is revealed. The quality of $\mathbf{w}_t$ is then estimated on the local empirical error $\frac{1}{2(t-1)} \sum_{j=1}^{t-1} (y_t - y_j - \mathbf{w}_t^\top (x_t - x_j))^2$. The next iterate $\mathbf{w}_{t+1}$ given by equation (1) is exactly obtained by performing a gradient descent step from the current iterate $\mathbf{w}_t$ based on the local empirical error. A similar form of algorithm (1) has been studied in [9, 18, 23]. For instance, a variant of the stochastic gradient descent algorithm was studied in [9, 18] which, at each iteration, requires an additional projection of $\mathbf{w}_t$ to a prescribed bounded ball.

Before stating our main result, consider a minimizer $\mathbf{w}^* = \arg\inf_{w \in \mathbb{R}^d} \mathcal{E}(\mathbf{w})$. The existence of a minimizer itself follows from the calculus of variations' direct method, as $\mathcal{E}(\mathbf{w})$ is lower bounded by zero, coercive after quotienting by the nullspace, and convex. However, the minimizer $\mathbf{w}^*$ may not be unique. To see this, denote the covariance matrix

$$\mathcal{C}_\rho = \iint_{\mathcal{X} \times \mathcal{X}} (x - x')(x - x')^\top d\rho_{\mathcal{X}}(x) d\rho_{\mathcal{X}}(x),$$

where $\rho_{\mathcal{X}}$ is the marginal distribution of ρ on $\mathcal{X}$. Denote by V_0 the eigenspace of $\mathcal{C}_\rho$ associated with the zero eigenvalue. Then, any $\mathbf{w}^* + v_0$ with $v_0 \in V_0$ is also a minimizer. Let $\mathbf{w}^*$ be the minimizer with zero component in the space V_0, denote by λ_ρ the smallest positive eigenvalue of matrix $\mathcal{C}_\rho$ and by κ the quantity $\sup_{x, x' \in \mathcal{X}} \|x - x'\|$.

Theorem 1. *Let $\gamma_t = \frac{t^{-\theta}}{\mu}$ for any $t \in \mathbb{N}$ with some $\theta \in (\frac{1}{2}, 1)$ and $\mu \geq \lambda_\rho + \kappa^2$, and $\{\mathbf{w}_t : t = 1, \ldots, T + 1\}$ be given by algorithm (1). Let $\mathbf{w}^*$ be the minimizer with zero component in the space V_0. Then, with probability $1 - \delta$,*

$$\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 \leq \bar{C}_{\theta, \rho, \mu}\, T^{-(2\theta - 1)} \log^2\big(\frac{4T}{\delta}\big), \quad (2)$$

where $\bar{C}_{\theta, \rho, \mu} > 0$ is a constant depending on θ, μ and λ_ρ of matrix $\mathcal{C}_\rho$ but independent of T (see its explicit form in the proof of the theorem).

In [23], an online learning algorithm for pairwise learning similar to (1) was studied in the setting of a reproducing kernel Hilbert space (RKHS). Specifically, in order to translate the results there in the linear case, for each vector $\mathbf{w} \in \mathbb{R}^d$ we associate the function $f_{\mathbf{w}}(x, x') = \mathbf{w}^\top (x - x')$. Theorem 2 from [23] proved that the convergence rate for

$$\|f_{\mathbf{w}_{T+1}} - f_{\mathbf{w}^*}\|_\rho^2 := \iint_{\mathcal{X} \times \mathcal{X}} |f_{\mathbf{w}_{T+1}}(x, x') - f_{\mathbf{w}^*}(x, x')|^2 d\rho_{\mathcal{X}}(x) d\rho_{\mathcal{X}}(x')$$

is of $\mathcal{O}(\log^2 T / T^{1/3})$. Notice that

$$\|f_{\mathbf{w}_{T+1}} - f_{\mathbf{w}^*}\|_\rho \leq \kappa \|\mathbf{w}_{T+1} - \mathbf{w}^*\|.$$

Consequently, in the linear case, our rate arbitrarily close to $\mathcal{O}(\log^2 T/T)$ is a sharp improvement over the rate of $\mathcal{O}(\log^2 T/T^{1/3})$ in [23].

2.1 Related Work

We now review existing work related to our work. Firstly, we discuss most recent work on online learning algorithms for pairwise learning. Generalization analysis were first done in [18] which provided online-to-batch conversion bounds for online pairwise learning algorithms. In [9], tighter bounds were established using Rademacher complexities. Algorithm (1) is closely related to the algorithm proposed in [18] which, however, needs a projection at each iteration to a bounded domain after the gradient descent step. This, in practice, leads to the difficult problem of selecting a bounded domain beforehand. On the contrary, the update step in Algorithm (1) is performed in the unconstrained setting and theoretically guaranteed to converge when the step sizes are in the form of $\mathcal{O}(t^{-\theta})$ with $\theta \in (1/2, 1)$. In particular, the rate can be arbitrarily close to $\mathcal{O}(1/T)$ when θ is close 1. To the best of our knowledge, this is the first result on the fast convergence of online pairwise learning algorithms without assuming strong convexity for the loss function.

Secondly, we review online learning algorithms in the univariate case. Online learning and stochastic approximation for the univariate loss [2, 4, 8, 11, 13, 15, 16, 22, 23] is well studied. For strongly convex loss, the optimal rate is $\mathcal{O}(1/T)$ [13]. For general convex loss, the convergence rate of the last iterate are $\mathcal{O}(\log(T)/\sqrt{T})$ and $\mathcal{O}(\log(T)/T)$ for strongly convex loss [15]. Recently, it was proved in [2] that online learning with the least-square loss, although being non-strongly convex, still achieves the optimal rate $\mathcal{O}(1/T)$ through an averaging scheme with constant step sizes. In infinite-dimensional RKHSs, convergence of the last iterate of stochastic gradient descent was established for strongly-convex losses [16] and non-strongly convex least-square loss [22]. .

Algorithm 1, as for algorithms in the univariate case, substitutes the true gradient by a computationally-cheap estimator but does not assume the objective function to be strongly convex nor to work in a constrained setting. In these aspects, algorithm (1) is closer to the following stochastic gradient descent in a RKHS $\mathcal{H}_G$ introduced in [22]:

$$\begin{cases} g_1 = 0 \ \text{ and } , \forall t \in 1, 2, \ldots, T \\ g_{t+1} = g_t - \gamma_t (g_t(x_t) - y_t) G_{x_t}. \end{cases}$$

The analysis in [22] heavily depends on the fact that the randomized gradient $(g_t(x_t) - y_t)G_{x_t}$ is an unbiased estimator of the true gradient $\iint_{\mathcal{X}} (g_t(x) - y) G_x d\rho(x, y)$. This is actually the main difficulty of analysing the convergence of algorithm (1) as the randomized gradient $\sum_{j=1}^{t-1} (\mathbf{w}_t(x_t - x_j) - y_t + y_j)(x_t - x_j)$ is not an unbiased estimator of the true gradient $\iint_{\mathcal{X}\times\mathcal{X}} (\mathbf{w}_t(x_t - x_j) - y_t + y_j)(x_t - x_j) d\rho(x, y) d\rho(x', y')$. This is due to the fact that the $T(T-1)/2$ pairs $(x_i - x_j, y_i - y_j)$ are not independent although the sampling is i.i.d itself. It is still possible to obtain $T/2$ independent pairs out of T samples, in that case the pairwise problem can be reduced to the univariate case and analysed using [22, 2]. In practice, people prefer not to discard the potential information contained in those T(T-1)/2 non i.i.d pairs and this trick is not used, Section 4 illustrates the reason.

Lastly, we discuss existing pairwise learning frameworks related to our work. In [1], the pairwise discrete ranking loss

$$\mathbb{I}_{[(y_t - y_j)(f(x_t) - f(x_j)) < 0]} + \frac{1}{2} \mathbb{I}_{[f(x_t) = f(x_j)]}$$

is considered, resulting in a batch learning algorithm minimizing the following empirical risk

$$\frac{1}{\binom{T}{2}} \sum_{t=2}^{T} \sum_{j=1}^{t-1} \max(0, |y_t - y_j| - (f(x_t) - f(x_j)) \operatorname{sgn}(y_t - y_j))$$

where the indicator function was replaced by the hinge loss as a convex surrogate. For AUC maximization, [9] provided an online algorithm aimed at minimizing the following

$$\frac{1}{T-1} \sum_{t=2}^{T} \frac{1}{t-1} \sum_{j=1}^{t-1} \max(0, 1 - (y_t - y_j) w^\top (x_t - x_j)),$$

and where AUC, the underlying quantity quantity being optimised is simply the loss $\mathbb{I}_{[w^\top (x_t - x_j) < 0]}$ when $y_t < y_j$ and 0 otherwise. In [7], the online learning algorithm optimizes the quantity

$$\frac{1}{T-1} \sum_{t=2}^{T} \sum_{j=1}^{t-1} \frac{\mathbb{I}_{[y_t \neq y_j]} (1 - y_t w^\top (x_t - x_j))^2}{2 |\{1 : y_j y_t = -1\}|},$$

which directly corresponds to the empirical AUC risk when the least square loss is used as a convex upper bound of the indicator function. Those frameworks simply differ in the loss functions used and to a certain extent to the penalty received for sample pairs of same label. We note that algorithm (1) relies on a slightly different least square loss formulation based on a similar local empirical error

$$\frac{1}{T-1} \sum_{t=2}^{2} \sum_{j=1}^{t-1} (w^\top (x_t - x_j) - y_t + y_j)^2.$$

In the particular case of the bipartite ranking setting with $\mathcal{Y} = \{0, 1\}$, we remark that $(1 - (y_t - y_j) w^\top (x_t - x_j))^2 = (w^\top (x_t - x_j) - y_t + y_j)^2$ when $y_t \neq y_j$, which can also be seen as an upper bound of the AUC loss.

3 PROOF OF MAIN RESULTS

We now turn our attention to the proof of Theorem 1 by introducing some notations. Let $\hat{\mathcal{C}}_t = \frac{1}{t-1}\sum_{\ell=1}^{t-1}(x_t - x_\ell)(x_t - x_\ell)^\top$, $\widetilde{\mathcal{C}}_t = \frac{1}{t-1}\sum_{\ell=1}^{t-1}\int_{\mathcal{X}}(x - x_\ell)(x - x_\ell)^\top d\rho_{\mathcal{X}}(x)$, and $\mathcal{C}_\rho = \iint_{\mathcal{X}\times\mathcal{X}}(x - x')(x - x')^\top d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x)$. Likewise, let $\hat{S}_t = \frac{1}{t-1}\sum_{\ell=1}^{t-1}(y_t - y_\ell)(x_t - x_\ell)$, $\widetilde{S}_t = \frac{1}{t-1}\sum_{\ell=1}^{t-1}\int_{\mathcal{X}}(f_\rho(x) - y_\ell)(x - x_\ell)d\rho_{\mathcal{X}}(x)$, and $S_\rho = \iint_{\mathcal{X}\times\mathcal{X}}\widetilde{f}_\rho(x,x')(x - x')d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x')$. Here $\widetilde{f}_\rho(x,x') = f_\rho(x) - f_\rho(x')$ with the regression function f_ρ being defined by $f_\rho(x) = \int_{\mathcal{Y}} y d\rho(y|x)$, where $\rho(\cdot|x)$ is the conditional distribution of ρ on $\mathcal{Y}$.

Notice that, for any minimizer $\mathbf{w}^* = \arg\inf_{\mathbf{w}\in\mathbb{R}^d}\mathcal{E}(\mathbf{w})$, there holds

$$\iint_{\mathcal{Z}\times\mathcal{Z}}((x-x')^\top\mathbf{w}^* - y + y')(x - x')d\rho(x,y)d\rho(x',y') = 0,$$

which implies that $\mathcal{C}_\rho\mathbf{w}^* = S_\rho$. We additionally define $\hat{\mathcal{A}}^t = (\widetilde{\mathcal{C}}_t - \mathcal{C}_\rho)\mathbf{w}_t - (\widetilde{S}_t - S_\rho)$, and $\hat{\mathcal{B}}^t = (\hat{\mathcal{C}}_t - \widetilde{\mathcal{C}}_t)\mathbf{w}_t - (\hat{S}_t - \widetilde{S}_t)$. Using the above notations, algorithm (1) can be written as

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= (I - \gamma_t\mathcal{C}_\rho)(\mathbf{w}_t - \mathbf{w}^*)\\ &\quad + \gamma_t(\mathcal{C}_\rho - \hat{\mathcal{C}}_t)\mathbf{w}_t + \gamma_t(\hat{S}_t - S_\rho)\\ &= (I - \gamma_t\mathcal{C}_\rho)(\mathbf{w}_t - \mathbf{w}^*) - \gamma_t\hat{\mathcal{A}}^t - \gamma_t\hat{\mathcal{B}}^t.\end{aligned} \tag{3}$$

Let $\omega_j^t(\mathcal{C}_\rho) = \prod_{\ell=j}^t(I - \gamma_\ell\mathcal{C}_\rho)$ for any $j \le t$, and introduce the conventional notations $\sum_{\ell=t+1}^t\gamma_\ell = 0$ and $\omega_{t+1}^t(\mathcal{C}_\rho) = I$. Then, we can derive from the equality (3), for any $2 \le t \le T$, that

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= -\omega_2^t(\mathcal{C}_\rho)\mathbf{w}^*\\ &\quad - \sum_{j=2}^t\gamma_j\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j - \sum_{j=2}^t\gamma_j\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{B}}^j.\end{aligned} \tag{4}$$

The strong convergence of $\|\mathbf{w}_{t+1} - \mathbf{w}^*\|$ stated in Theorem 1 will be proved by estimating the terms on the righthand of (4). To this end, we needs some lemmas. The first lemma states that $\mathbf{w}_t$ are almost surely orthogonal to the eigenspace V_0. This observation is inspired by the recent study on the randomized Kaczmarz algorithm [10] for regression.

Lemma 1. *Let the learning sequence $\{\mathbf{w}_t : t = 1, 2, \ldots, T+1\}$ be produced by (1). Then, for any t, $\mathbf{w}_t$ is almost surely orthogonal to the eigenspace V_0.*

Proof. We prove the lemma by induction. The result holds true for $t \le 2$ since $\mathbf{w}_1 = \mathbf{w}_2 = 0$. Assume, for some $t \ge 3$, that $\mathbf{w}_t$ is almost surely orthogonal to the eigenspace V_0. We are going to prove a similar result for $\mathbf{w}_{t+1}$ from (3). To see this, for any $v \in V_0$ and $t, j \in \mathbb{N}$, observe that

$$\iint_{\mathcal{X}^2}|v^\top(x_t - x_j)|^2 d\rho_{\mathcal{X}}(x_t)d\rho_{\mathcal{X}}(x_j) = v^\top\mathcal{C}_\rho v = 0.$$

Similarly,

$$\begin{aligned}&|v^\top S_\rho|^2\\ &\le \Big(\iint_{\mathcal{X}\times\mathcal{X}}|\widetilde{f}_\rho(x,x')||v^\top(x - x')|d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x')\Big)^2\\ &\le \Big(\iint_{\mathcal{X}\times\mathcal{X}}|\widetilde{f}_\rho(x,x')|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x')\Big)^2\\ &\qquad\cdot\Big(\iint_{\mathcal{X}\times\mathcal{X}}|v^\top(x - x')|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x')\Big)^2\\ &= \Big(\iint_{\mathcal{X}\times\mathcal{X}}|\widetilde{f}_\rho(x,x')|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x')\Big)^2\big(v^\top\mathcal{C}_\rho v\big)\\ &= 0.\end{aligned}$$

In addition, for any $\ell \le t-1$, there holds

$$\begin{aligned}&\int_{\mathcal{X}}\Big|\int_{\mathcal{X}}v^\top(x - x_\ell)(x - x_\ell)^\top d\rho_{\mathcal{X}}(x)\mathbf{w}_t\Big|d\rho_{\mathcal{X}}(x_\ell)\\ &\le \Big(\int_{\mathcal{X}}\int_{\mathcal{X}}|v^\top(x - x_\ell)|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x_\ell)\Big)^{1/2}\\ &\qquad\cdot\Big(\int_{\mathcal{X}}\int_{\mathcal{X}}|\mathbf{w}_t^\top(x - x_\ell)|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x_\ell)\Big)^{1/2}\\ &= (v^\top\mathcal{C}_\rho v)^{1/2}\\ &\qquad\cdot\Big(\int_{\mathcal{X}}\int_{\mathcal{X}}|\mathbf{w}_t^\top(x - x_\ell)|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x_\ell)\Big)^{1/2} = 0,\end{aligned}$$

and

$$\begin{aligned}&\int_{\mathcal{Z}}\Big|\int_{\mathcal{X}}v^\top(f_\rho(x) - y_\ell)(x - x_\ell)d\rho_{\mathcal{X}}(x)\Big|d\rho(z_\ell)\\ &\le \Big(\int_{\mathcal{Z}}(f_\rho(x) - y_\ell)^2 d\rho(x, y_\ell)\Big)^{1/2}\\ &\qquad\cdot\Big(\int_{\mathcal{X}}\int_{\mathcal{X}}|v^\top(x - x_\ell)|^2 d\rho_{\mathcal{X}}(x)d\rho_{\mathcal{X}}(x_\ell)\Big)^{1/2}\\ &= \Big(\int_{\mathcal{Z}}(f_\rho(x) - y_\ell)^2 d\rho(x, y_\ell)\Big)^{1/2}(v^\top\mathcal{C}_\rho v)^{1/2} = 0.\end{aligned}$$

In summary, the above inequalities imply that $x_t - x_j \perp V_0$, $\int_{\mathcal{X}}(x - x_\ell)(x - x_\ell)^\top d\rho_{\mathcal{X}}(x)\mathbf{w}_t \perp V_0$, $\int_{\mathcal{X}}(f_\rho(x) - y_\ell)(x - x_\ell)d\rho_{\mathcal{X}}(x) \perp V_0$, and $S_\rho \perp V_0$ almost surely, which by the definition of $\hat{\mathcal{A}}^t$ and $\hat{\mathcal{B}}^t$, leads to $\hat{\mathcal{A}}^t \perp V_0$ and $\hat{\mathcal{B}}^t \perp V_0$ almost surely. Consequently, from (3), $\mathbf{w}_{t+1}$ is orthogonal to V_0. This completes the proof of the lemma. □

The above lemma indicates that the error decomposition equality (4) holds true in $V_0^\perp$, the orthogonal complement of V_0 in R^d. Denote by $\omega_{j+1}^t(\lambda_\rho) = \prod_{\ell=j+1}^t(1 - \gamma_\ell\lambda_\rho)$ for any $j \le t$. Then, we have the following result.

Lemma 2. *Assume that $\gamma_\ell\kappa^2 \le 1$ for any $\ell \in \mathbb{N}$. Then, for any $j \le t$, there holds $\|\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j\| \le \omega_{j+1}^t(\lambda_\rho)\|\hat{\mathcal{A}}^j\|$ and $\|\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{B}}^j\| \le \omega_{j+1}^t(\lambda_\rho)\|\hat{\mathcal{B}}^j\|$, where $\|\cdot\|$ denotes the Euclidean norm.*

Proof. Let us prove the first inequality. To this end, recall from the proof of Lemma 1 that $\hat{\mathcal{A}}^j \perp V_0$. For any $v \in V_0$, observe that $v^\top \omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j = v^\top \hat{\mathcal{A}}^j = 0$. Hence, $\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j \perp V_0$. Moreover, we can write $\hat{\mathcal{A}}^j = \sum_{k:\lambda_k>0}(v_k^\top \hat{\mathcal{A}}^j)v_k$, where $\{v_k\}$ and $\{\lambda_k\}$ are the eigenvectors and eigenvalues of $\mathcal{C}_\rho$. Consequently,

$$\begin{aligned}\|\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j\| \quad &= \|\textstyle\sum_{k:\lambda_k>0}(v_k^\top \hat{\mathcal{A}}^j)\omega_{j+1}^t(\mathcal{C}_\rho)v_k\| \\ &= \|\textstyle\sum_{k:\lambda_k>0}(v_k^\top \hat{\mathcal{A}}^j)\omega_{j+1}^t(\lambda_k)v_k\| \\ &= \big(\textstyle\sum_{k:\lambda_k>0}|(v_k^\top \hat{\mathcal{A}}^j)\omega_{j+1}^t(\lambda_k)|^2\big)^{1/2} \\ &\leq \omega_{j+1}^t(\lambda_\rho)\big(\textstyle\sum_{k:\lambda_k>0}|v_k^\top \hat{\mathcal{A}}^j|^2\big)^{1/2} \\ &= \omega_{j+1}^t(\lambda_\rho)\|\hat{\mathcal{A}}^j\|,\end{aligned}$$

where the second to the last inequality used the fact that $\lambda_k \leq \|\mathcal{C}_\rho\| \leq \sup_{x,x'\in\mathcal{X}}\|x-x'\|^2 = \kappa^2$ and $\gamma_\ell\lambda_k \leq \gamma_\ell\kappa^2 \leq 1$ for any $\ell \in \mathbb{N}$. The proof for the second inequality can be done similarly. This completes the proof of the lemma. □

The following lemma gives an upper-bound of the norms of the learning sequence $\{\mathbf{w}_t : t \in \mathbb{N}\}$.

Lemma 3. *Let the learning sequence $\{\mathbf{w}_t : t \in \mathbb{N}\}$ be given by (1) and assume, for any $t \in \mathbb{N}$, that $\gamma_t\kappa^2 \leq 1$. Then, for any $t \in \mathbb{N}$ we have $\|\mathbf{w}_t\| \leq 2M\big(\sum_{j=2}^{t-1}\gamma_j\big)^{\frac{1}{2}}$.*

Proof. For $t = 1$ or $t = 2$, by definition $\mathbf{w}_1 = \mathbf{w}_2 = 0$ which trivially satisfy the desired inequality. It suffices to prove the case of $t \geq 2$ by induction. By recalling the recursive equality (1), we have

$$\|\mathbf{w}_{t+1}\|^2 \leq \|\mathbf{w}_t\|^2 + \frac{\gamma_t^2\kappa^2}{t-1}\sum_j^{t-1}(\mathbf{w}_t^\top(x_t, x_j) - y_t + y_j)^2$$

$$-\frac{2\gamma_t}{t-1}\sum_{j=1}^{t-1}(\mathbf{w}_t^\top(x_t - x_j) - y_t + y_j)\mathbf{w}_t^\top(x_t - x_j).$$

Define a univariate function F_j by $F_j(s) = \kappa^2\gamma_t(s - y_t + y_j)^2 - 2(s - y_t + y_j)s$. It is easy to see that $\sup_{s\in\mathbb{R}} F_j(s) = \frac{(y_t-y_j)^2}{2-\kappa^2\gamma_t} \leq (2M)^2$ since $\gamma_t\kappa^2 \leq 1$ and $|y_j| + |y_t| \leq 2M$. Therefore, from the above estimation we can get, for $t \geq 2$, that

$$\begin{aligned}\|\mathbf{w}_{t+1}\|^2 \quad &\leq \|\mathbf{w}_t\|^2 + \tfrac{\gamma_t}{t-1}\textstyle\sum_{j=1}^{t-1}\sup_j F_j(s) \\ &\leq \|\mathbf{w}_t\|^2 + (2M)^2\gamma_t.\end{aligned}$$

Combining the above inequality with the induction assumption that $\|\mathbf{w}_t\| \leq 2M\sqrt{\sum_{j=2}^{t-1}\gamma_j}$ implies the desired result. This completes the proof of the lemma. □

We also need the following probabilistic inequalities in a Hilbert space. The first one is the Bennett's inequality for random variables in Hilbert spaces, which can be derived from [16, Theorem B4].

Lemma 4. *Let $\{\xi_i : i = 1, 2, \ldots, t\}$ be independent random variables in a Hilbert space $\mathcal{H}$ with norm $\|\cdot\|$. Suppose that almost surely $\|\xi_i\| \leq B$ and $\mathbb{E}\|\xi_i\|^2 \leq \sigma^2 < \infty$. Then, for any $0 < \delta < 1$, there holds, with probability at least $1-\delta$,*

$$\Big\|\frac{1}{t}\sum_{i=1}^t[\xi_i - \mathbb{E}\xi_i]\Big\| \leq \frac{2B\log\frac{2}{\delta}}{t} + \sigma\sqrt{\frac{\log\frac{2}{\delta}}{t}}.$$

The second probabilistic inequality is the Pinelis-Bernstein inequality for martingale difference sequence in a Hilbert space, which is derived from [12, Theorem 3.4].

Lemma 5. *Let $\{S_k : k \in \mathbb{N}\}$ be a martingale difference sequence in a Hilbert space. Suppose that almost surely $\|S_k\| \leq B$ and $\sum_{k=1}^t \mathbb{E}[\|S_k\|^2|S_1, \ldots, S_{k-1}] \leq \sigma_t^2$. Then, for any $0 < \delta < 1$, there holds, with probability at least $1-\delta$,*

$$\sup_{1\leq j\leq t}\Big\|\sum_{k=1}^j S_k\Big\| \leq 2\left(\frac{B}{3} + \sigma_t\right)\log\frac{2}{\delta}.$$

After the above preparations, we can now present the following bounds for the terms on the righthand side of the error decomposition (4).

Theorem 2. *Assume that $\gamma_\ell(\kappa^2 + \lambda_\rho) \leq 1$ for any $\ell \in \mathbb{N}$. Then, for any $0 < \delta < 1$, the following estimations hold true.*

(a) With probability $1-\delta$, there holds

$$\|\sum_{j=2}^t\gamma_j\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j\| \leq 6\sqrt{2}(1+\kappa)\kappa M\log\big(\frac{2t}{\delta}\big)$$

$$\sum_{j=2}^t\frac{\gamma_j\omega_{j+1}^t(\lambda_\rho)}{\sqrt{j}}\big(1 + (\sum_{\ell=2}^{j-1}\gamma_\ell\big)^{1/2}).$$

(b) With probability $1-\delta$, we have

$$\|\sum_{j=2}^t\gamma_j\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{B}}^j\| \leq \frac{32\sqrt{2}}{3}(1+\kappa)\kappa M\log\big(\frac{2}{\delta}\big)$$

$$\Big(\sum_{j=2}^t\gamma_j^2(\omega_{j+1}^t(\lambda_\rho))^2(1 + \sum_{\ell=2}^{j-1}\gamma_\ell)\Big)^{1/2}.$$

Proof. We start with the proof of part (a). From Lemma 2 and Lemma 3, we have

$$\begin{aligned}&\|\sum_{j=2}^t\gamma_j\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{A}}^j\| \leq \sum_{j=2}^t\gamma_j\omega_{j+1}^t(\lambda_\rho)\|\hat{\mathcal{A}}^j\| \\ &\leq \sum_{j=2}^t\gamma_j\omega_{j+1}^t(\lambda_\rho)(\|\mathcal{C}_\rho - \widetilde{\mathcal{C}}_j\|\|\mathbf{w}_j\| + \|\widetilde{S}_j - S_\rho\|) \\ &\leq \textstyle\sum_{j=2}^t\gamma_j\omega_{j+1}^t(\lambda_\rho) \\ &\qquad\cdot\Big(2M\|\mathcal{C}_\rho - \widetilde{\mathcal{C}}_j\|(\textstyle\sum_{\ell=2}^{j-1}\gamma_\ell)^{1/2} + \|\widetilde{S}_j - S_\rho\|\Big),\end{aligned}$$

where, for any $2 \leq j \leq t$, $\|\mathcal{C}_\rho - \widetilde{\mathcal{C}}_j\|$ denotes the Frobenius norm of matrix $\mathcal{C}_\rho - \widetilde{\mathcal{C}}_j$. Applying Lemma 4 with $B = \sigma = \kappa^2$, with probability $1 - \frac{\delta}{t}$ there holds $\|\mathcal{C}_\rho - \widetilde{\mathcal{C}}_j\| \leq \frac{2\kappa^2 \log \frac{2t}{\delta}}{j-1} + \kappa^2 \sqrt{\frac{\log \frac{2t}{\delta}}{j-1}} \leq 3\sqrt{2}\kappa^2 \log(\frac{2t}{\delta})/\sqrt{j}$. Similarly, applying Lemma 4 with $B = \sigma = 2\kappa M$ implies, with probability $1 - \frac{\delta}{t}$, that

$$\begin{aligned} \|\widetilde{S}_j - S_\rho\| \quad & \leq \tfrac{4M\kappa \log \frac{2t}{\delta}}{j-1} + 2\kappa M \sqrt{\log \tfrac{2t}{\delta}/(j-1)} \\ & \leq 6\sqrt{2}\kappa M \log(\tfrac{2t}{\delta})/\sqrt{j}. \end{aligned}$$

Putting these estimation into (3) implies part (a).

For part (b), observe that $\{\xi_j := \gamma_j \omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{B}}^j : j = 2, \ldots, t\}$ is a martingale difference sequence. we will apply Lemma 5 to prove part (b). To this end, it needs to estimate B and σ. Indeed, by Lemma 3, we get that

$$\begin{aligned} \|\hat{\mathcal{B}}_j\| \quad & \leq \|\hat{\mathcal{C}}_j - \widetilde{\mathcal{C}}_j\| \|\mathbf{w}_j\| + \|\hat{S}_j - \widetilde{S}_j\| \\ & \leq 4\kappa^2 M \big(\textstyle\sum_{\ell=2}^{j-1} \gamma_\ell\big)^{\frac{1}{2}} + 2\kappa M \\ & \leq 4\sqrt{2}\kappa(1+\kappa) M \big(1 + \textstyle\sum_{\ell=2}^{j-1} \gamma_\ell\big)^{\frac{1}{2}}. \end{aligned}$$

From Lemma 2 and the above estimation, we have that

$$\begin{aligned} \sigma_t^2 \quad & = \sum_{j=2}^{t} \gamma_j^2 \mathbb{E}(\|\omega_{j+1}^t(\mathcal{C}_\rho)\hat{\mathcal{B}}_j\|^2 | z_1, \ldots, z_{j-1}) \\ & \leq \sum_{j=2}^{t} \gamma_j^2 (\omega_{j+1}^t(\lambda_\rho))^2 \mathbb{E}(\|\hat{\mathcal{B}}_j\|^2 | z_1, \ldots, z_{j-1}) \\ & \leq 32\kappa^2(1+\kappa)^2 M^2 \\ & \qquad \cdot \textstyle\sum_{j=2}^t \gamma_j^2 (\omega_{j+1}^t(\lambda_\rho))^2 (1 + \displaystyle\sum_{\ell=2}^{j-1} \gamma_\ell), \end{aligned} \tag{5}$$

and

$$\begin{aligned} B \quad & = \sup_{2 \leq j \leq t} \big[\gamma_j \omega_{j+1}^t(\lambda_\rho) \|\hat{\mathcal{B}}^j\|\big] \\ & \leq (\textstyle\sum_{j=2}^t \big[\gamma_j \omega_{j+1}^t(\lambda_\rho) \|\hat{\mathcal{B}}^j\|\big]^2)^{1/2} \\ & \leq 4\sqrt{2}\kappa(1+\kappa) M \\ & \qquad \cdot \Big(\textstyle\sum_{j=2}^t \gamma_j^2 (\omega_{j+1}^t(\lambda_\rho))^2 (1 + \displaystyle\sum_{\ell=2}^{j-1} \gamma_\ell)\Big)^{1/2}. \end{aligned} \tag{6}$$

Applying Lemma 5 with the estimation of B and σ_t being given by (5) and (6) implies the desired result in part (b). This completes the proof of the theorem. □

Theorem 1 can be derived from Theorem 2 by using the following technical lemma.

Lemma 6. *Let $\gamma_j = \frac{j^{-\theta}}{\mu}$ for any $j \in \mathbb{N}$ with some $\theta \in (1/2, 1)$. Then, there holds*

$$\begin{aligned} & \textstyle\sum_{j=2}^t \frac{\gamma_j \omega_{j+1}^t(\lambda_\rho)}{\sqrt{j}} \big(1 + \big(\sum_{\ell=2}^{j-1} \gamma_\ell\big)^{1/2}\big) \\ & \leq \tfrac{2 \max(1, (\mu(1-\theta))^{-1/2})}{\mu} \\ & \qquad \cdot \big(1 + \big(\tfrac{\mu 2^{\frac{5\theta}{2}}}{\lambda_\rho} + \Big(\tfrac{\mu(2+3\theta)}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{2+3\theta}{2(1-\theta)}}\big)\big) t^{-\frac{\theta}{2}}, \end{aligned} \tag{7}$$

and

$$\begin{aligned} & \Big(\textstyle\sum_{j=2}^t \gamma_j^2 (\omega_{j+1}^t(\lambda_\rho))^2 (1 + \sum_{\ell=2}^{j-1} \gamma_\ell)\Big)^{1/2} \\ & \leq \tfrac{\max(1, (\mu(1-\theta))^{-1/2})}{\mu} \\ & \qquad \cdot \Big(1 + \Big(\tfrac{\mu 2^{4\theta-1}}{2\lambda_\rho} + \Big(\tfrac{3\mu\theta}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{3\theta}{1-\theta}}\Big)\Big)^{1/2} t^{-(\theta - \frac{1}{2})}. \end{aligned} \tag{8}$$

Proof. The proof needs the elementary inequality (see e.g. [17, Lemma 2]): for any $\nu > 0, a > 0, 0 < q_1 < 1$, and $q_2 \geq 0$, then, for any $t \in \mathbb{N}$,

$$\begin{aligned} \sum_{j=1}^{t-1} j^{-q_2} \exp\big(-\nu \sum_{\ell=j+1}^{t} \ell^{-q_1}\big) \leq \Big(\frac{2^{q_1+q_2}}{\nu} \\ + \Big(\frac{1+q_2}{\nu(1-2^{q_1-1})e}\Big)^{\frac{1+q_2}{1-q_1}}\Big) t^{q_1-q_2}. \end{aligned} \tag{9}$$

To this end, denote the lefthand term of (7) by $\mathcal{I} = \sum_{j=2}^t \frac{\gamma_j}{\sqrt{j}} \big[\prod_{\ell=j+1}^t (1 - \lambda_\rho \gamma_\ell)\big] (1 + (\sum_{\ell=2}^{j-1} \gamma_\ell)^{1/2})$. Indeed, we have

$$\begin{aligned} \mathcal{I} \quad & \leq \tfrac{t^{-\theta-\frac{1}{2}}}{\mu} \big(1 + (\tfrac{1}{\mu(1-\theta)}((t-1)^{1-\theta} - 1))^{1/2}\big) \\ & \quad + \sum_{j=2}^{t-1} \frac{j^{-\theta-\frac{1}{2}}}{\mu} \exp\Big(-\frac{\lambda_\rho}{\mu} \sum_{\ell=j+1}^{t} \ell^{-\theta}\Big) \\ & \qquad \cdot \big[1 + (\tfrac{1}{\mu(1-\theta)}((j-1)^{1-\theta} - 1))^{1/2}\big] \\ & \leq \tfrac{2\max(1, (\mu(1-\theta))^{-1/2})}{\mu} \\ & \qquad \cdot \Big(t^{-\frac{3\theta}{2}} + \sum_{j=2}^{t-1} j^{-\frac{3\theta}{2}} \exp\big(-\frac{\lambda_\rho}{\mu} \sum_{\ell=j+1}^{t} \ell^{-\theta}\big)\Big) \\ & \leq \tfrac{2\max(1, (\mu(1-\theta))^{-1/2})}{\mu} \\ & \qquad \cdot \Big(t^{-\frac{3\theta}{2}} + \Big(\tfrac{\mu 2^{\frac{5\theta}{2}}}{\lambda_\rho} + \Big(\tfrac{\mu(2+3\theta)}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{2+3\theta}{2(1-\theta)}}\Big) t^{-\theta/2}\Big) \\ & \leq \tfrac{2\max(1, (\mu(1-\theta))^{-1/2})}{\mu} \\ & \qquad \cdot \Big(1 + \Big(\tfrac{\mu 2^{\frac{5\theta}{2}}}{\lambda_\rho} + \Big(\tfrac{\mu(2+3\theta)}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{2+3\theta}{2(1-\theta)}}\Big)\Big) t^{-\frac{\theta}{2}}, \end{aligned} \tag{10}$$

where the third to last inequality used inequality 9 with $q_1 = \theta, q_2 = \frac{3\theta}{2}$, and $\nu = \frac{\lambda_\rho}{\mu}$. This completes the estimation of (7).

Now we turn to the estimation of (8) where the term on the lefthand side is denoted by $\mathcal{J}$. Similarly we have

$$\begin{aligned} & (\mathcal{J})^2 \leq \tfrac{1}{\mu^2} t^{-2\theta} \big(1 + \tfrac{(t-1)^{1-\theta} - 1}{\mu(1-\theta)}\big) \\ & + \textstyle\sum_{j=2}^{t-1} \frac{j^{-2\theta}}{\mu^2} \exp\big(-\frac{2\lambda_\rho}{\mu} \sum_{\ell=j+1}^t \ell^{-\theta}\big) \big(1 + \frac{(j-1)^{1-\theta} - 1}{\mu(1-\theta)}\big) \\ & \leq \tfrac{2\max(1, (\mu(1-\theta))^{-1})}{\mu^2} \\ & \qquad \cdot \Big[t^{1-3\theta} + \textstyle\sum_{j=2}^{t-1} j^{-(3\theta-1)} \exp\big(-\frac{2\lambda_\rho}{\mu} \sum_{\ell=j+1}^t \ell^{-\theta}\big)\Big] \\ & \leq \tfrac{2\max(1, (\mu(1-\theta))^{-1})}{\mu^2} \\ & \qquad \cdot \Big(1 + \Big(\tfrac{\mu 2^{4\theta-1}}{2\lambda_\rho} + \Big(\tfrac{3\mu\theta}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{3\theta}{1-\theta}}\Big)\Big) t^{-(2\theta-1)}, \end{aligned}$$

where, in the last inequality, we used 9 with $q_1 = \theta, q_2 = 3\theta - 1$, and $\nu = \frac{2\lambda_\rho}{\mu}$. Hence,

$$\mathcal{J} \leq \frac{\sqrt{2}\max(1, (\mu(1-\theta))^{-1/2})}{\mu}\Big(1+\Big(\frac{\mu 2^{4\theta-1}}{2\lambda_\rho} + \Big(\frac{3\mu\theta}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{3\theta}{1-\theta}}\Big)\Big)^{1/2} t^{-(\theta-\frac{1}{2})}.$$

This completes the proof of the lemma. □

We are finally ready to prove Theorem 1 by using Theorem 2 and Lemma 6.

Proof of Theorem 1. By (4), there holds

$$\|\mathbf{w}_{T+1} - \mathbf{w}^*\| \leq \|\omega_2^T(\mathcal{C}_\rho)\mathbf{w}^*\| + \|\sum_{j=2}^{T}\gamma_j\omega_{j+1}^T(\mathcal{C}_\rho)\hat{\mathcal{A}}^j\| + \|\sum_{j=2}^{T}\gamma_j\omega_{j+1}^T(\mathcal{C}_\rho)\hat{\mathcal{B}}^j\|. \quad (11)$$

In addition, recall that $\mathbf{w}^* \perp V_0$. Then, there holds

$$\|\omega_2^t(\mathcal{C}_\rho)\mathbf{w}^*\| \leq \prod_{j=2}^{t}(1-\lambda_\rho\gamma_j)\|\mathbf{w}^*\|. \quad (12)$$

we have

$$\begin{aligned}
&\|\omega_2^T(\mathcal{C}_\rho)\mathbf{w}^*\| \\
&\leq \exp(-\tfrac{\lambda_\rho}{\mu}\textstyle\sum_{j=2}^{T}\ell^{-\theta})\|\mathbf{w}^*\| \\
&\leq \tfrac{2\kappa M}{\lambda_\rho}\exp\big(-\tfrac{\lambda_\rho}{\mu(1-\theta)}(T^{1-\theta}-2)\big) \\
&\leq \tfrac{2\kappa M}{\lambda_\rho}\exp(\tfrac{2\lambda_\rho}{\mu(1-\theta)})\exp(-\tfrac{\lambda_\rho}{\mu(1-\theta)}T^{1-\theta}) \\
&\leq \tfrac{2\kappa M}{\lambda_\rho}\exp(\tfrac{2\lambda_\rho}{\mu(1-\theta)})\Big(\tfrac{\mu(2\theta-1)}{2\lambda_\rho e}\Big)^{\frac{2\theta-1}{2(1-\theta)}}T^{-(\theta-\frac{1}{2})}.
\end{aligned} \quad (13)$$

The second inequality in the above estimation relies on the fact (from the proof of from Lemma 1) that $\mathcal{C}_\rho\mathbf{w}^*, S_\rho \perp V_0$. This implies that $\|\mathbf{w}^*\| = \|\mathcal{C}_\rho^{-1}S_\rho\|$ holds true in the eigenspace corresponding to non-zero eigenvalues of $\mathcal{C}_\rho$ for which $\mathcal{C}_\rho^{-1}$ is well defined (i.e. it equals to the pseudo inverse of $\mathcal{C}_\rho$). The last inequality of the above estimation used the elementary inequality (see e.g. [17, Lemma 2]): for any $x > 0$, $\exp(-\nu x) \leq (\frac{a}{\nu e})^a x^{-a}$.

Combining (7), (8), and (13) with Theorem 2, we obtain from inequality (11), with probability $1-\delta$, that $\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 \leq \bar{C}^2_{\theta,\rho,\mu}T^{-(2\theta-1)}\log^2\big(\frac{4T}{\delta}\big)$, where

$$\begin{aligned}
\bar{C}_{\theta,\rho,\mu} = {} & (12\sqrt{2}+\frac{128}{3})(1+\kappa)\kappa M\frac{\max(1,(\mu(1-\theta))^{-1})}{\mu} \\
& \cdot\Big[2+\frac{\mu 2^{\frac{5\theta}{2}}}{\lambda_\rho}+\Big(\frac{\mu(2+3\theta)}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{2+3\theta}{2(1-\theta)}}+\Big(\frac{\mu 2^{4\theta-1}}{2\lambda_\rho} \\
& +\Big(\frac{3\mu\theta}{2\lambda_\rho(1-2^{\theta-1})e}\Big)^{\frac{3\theta}{1-\theta}}\Big)\Big)^{1/2}\Big]+\frac{2\kappa M}{\lambda_\rho}\exp(\frac{2\lambda_\rho}{\mu(1-\theta)}) \\
& \cdot\Big(\frac{\mu(2\theta-1)}{2\lambda_\rho e}\Big)^{\frac{2\theta-1}{2(1-\theta)}}.
\end{aligned}$$

This completes the proof of the theorem. □

4 PRELIMINARY EXPERIMENTS

In this section we first introduce an efficient implementation of algorithm (1) and then evaluate its performance on benchmark datasets. We stress that those results are preliminary, aimed at empirically studying the convergence of algorithm (1) on pairwise learning problems.

4.1 Implementation

We remark that algorithm (1) can be implemented in linear time with respect to the number of samples and not quadratic, as the double sum in (1) would suggest, by updating the following four quantities at each iteration: $XX_t = \frac{1}{t-1}\sum_{j=1}^{t-1}x_jx_j^\top$, $X_t = \frac{1}{t-1}\sum_{j=1}^{t-1}x_j$, $Y_t = \frac{1}{t-1}\sum_{j=1}^{t-1}y_j$ and $YX_t = \frac{1}{t-1}\sum_{j=1}^{t-1}y_jx_j$.

The resulting algorithm has a time complexity of $\mathcal{O}(Td^2)$ and a space complexity of $\mathcal{O}(d^2)$. Incidentally, the more straightforward implementation yields a $\mathcal{O}(T^2d)$ time complexity which could be preferred when working with high-dimensional datasets were the number of features far exceeds the sample size.

Algorithm 1

```
Input: $\theta, \mu$
Initialization: $\mathbf{w}_0 = \mathbf{w}_1 = \mathbf{0}, XX_1 = [\mathbf{0}]_{d\times d}, X_1 = \mathbf{0}, Y_1 = 0, YX_1 = \mathbf{0}$
for $t = 2, \ldots, T$ do
    Receive training pair $(x_t, y_t)$
    $XX_t = ((t-2)XX_{t-1} + x_tx_t^\top)/(t-1)$
    $X_t = ((t-2)X_{t-1} + x_t)/(t-1)$
    $Y_t = ((t-2)Y_{t-1} + y_t)/(t-1)$
    $YX_t = ((t-2)YX_{t-1} + y_tx_t)/(t-1)$
    $\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{t^{-\theta}}{\mu}([\mathbf{w}_t^\top(x_t - X_t)]x_t - [\mathbf{w}_t^\top x_t]X_t + XX_t\mathbf{w}_t + (y_t - Y)x_t - YX_t + y_tX_t)$
end for
```

Table 1: benchmark datasets

datasets	T	d	datasets	T	d
sonar	208	60	splice	3175	60
ionosphere	351	34	a9a	32561	123
diabetes	768	8	w8a	49749	300
german	1000	24	ijcnn1	141691	22
svmguide3	1243	22	covtype	581012	54

Table 2: Comparison of AUC values (mean±std) on benchmark datasets

datasets	(1)	SGD	OPAUC
sonar	.8213±.0679	.7968±.0487	.8038±.0574
ionosphere	.9438±.0330	.9352±.0333	.9131±.0419
diabetes	.8278±.0277	.8233±.0237	.8291±.0381
german	.7914±.0318	.7728±.0352	.7962±.0203
svmguide3	.7199±.0438	.7005±.0536	.7078±.0397
splice	.9246±.0092	.9160±.0090	.9179±.0095
a9a	.8960±.0037	.8947±.0042	.8996±.0042
w8a	.9557±.0069	.9524±.0050	.9508±.0049
ijcnn1	.9251±.0033	.9227±.0034	.9365±.0025
covtype	.8230±.0012	.8222±.0016	.8226±.0012

4.2 Comparison on Benchmark Data

We measured the performance on AUC optimization tasks, and report results on 10 standard binary classification datasets of different sample sizes and class imbalance[1]. We compared Algorithm 1 to the online algorithms OPAUC [7] as well as the stochastic version of Algorithm 1 where $T/2$ independent pairs are used. Hyperparameters were selected on the training fold, and AUC values obtained by overaging over five trials of 5-fold cross validation (Table 2) after one pass over the dataset.

Our results show that algorithm (1) fared always better than the sgd variant relying only on $T/2$ truly independent pairs but also competed fairly against OPAUC the state of the art for online AUC algorithms. This is promising as algorithm (1) its current form was not adapted to directly optimize pairs of opposite classes as in other AUC maximization algorithms. While simply minimising over all pairs, algorithm (1) performs well on AUC tasks and enjoys an efficient implementation as well as a fast convergence rate.

In addition, Figure 1 shows the evolution of the AUC over several epochs for the first four datasets. The same experimental protocol was used and the dataset was additionally shuffled between each pass. It is quite clear that reducing pairwise problems to the univariate case by discarding dependent pairs, although having the same assymptotic convergence, is subefficient in practice.

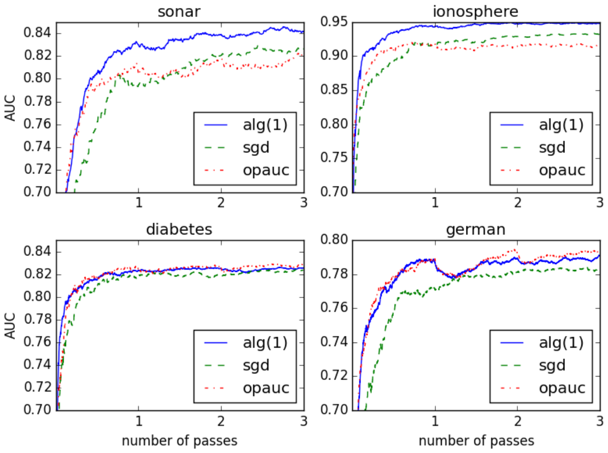

Figure 1: Influence of epochs on AUC

5 CONCLUSION

In this paper, we proved the fast convergence rate for an online pairwise learning algorithm with a non-strongly-convex loss in an unconstrained setting. Specifically, under the assumption of polynomially decaying step sizes, we established that the convergence rate of the last iterate to the minimizer of the true risk is arbitrarily close to $\mathcal{O}(\log^2 T/T)$. We are currently exploring ideas to improve the scalability of algorithm (1). From a practical point of view, algorithm (1) has a linear time implementation that only needs to store the first two moments of the data. However, when the implementation in $\mathcal{O}(T^2 d)$ is favored, algorithm (1) is not a fully online learning algorithm since it needs to store previous samples. One possibility is to work with a truly stochastic update consisting of only a pair of examples at each iteration, or to rely only on a buffering set of past training samples, as used in [9, 18], when computing the gradient estimator. Finally, we notice that our rate $\mathcal{O}(1/T)$ depends on the smallest positive eigenvalue of $\mathcal{C}_\rho$. It would be interesting to exploit strategies such as an averaging scheme of the iterates to relax such a dependency.

References

[1] S. Agarwal and P. Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 10: 441–474, 2009.

[2] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with conver-

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/

gence rate $O(1/n)$. *NIPS*, 2013.

[3] Q. Cao, Z. C. Guo, and Y. Ying. Generalization bounds for metric and similarity learning. *arXiv:1207.5437*, 2012.

[4] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9): 2050–2057, 2004.

[5] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. *ICML*, 2007.

[6] S. Clémençon, G. Lugosi, and N. Vayatis. *The Annals of Statistics*, 36: 844–874, 2008.

[7] W. Gao, R. Jin, S. Zhu, and Z. H. Zhou. One-pass AUC optimization. *arXiv preprint arXiv:1305.1363*, 2013.

[8] E. Hazan, A. Kalai, K. Satyen, and A. Agarwal. Logarithmic regret algorithms for online convex optimization. *COLT*, 2006.

[9] P. Kar, B. K. Sriperumbudur, P. Jain, and H. C. Karnick. On the generalization ability of online learning algorithms for pairwise loss functions. *ICML*, 2013.

[10] J. Lin and D. X. Zhou. Learning theory of randomized kaczmarz algorithm. *To appear in Journal of Machine Learning Research*, 2015.

[11] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[12] I. Pinelis. Optimum bounds for the distributions of martingales in banach spaces. *The Annals of Probability*, 22(4): 1679–1706, 1994.

[13] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *ICML*, 2012.

[14] W. Rejchel. On ranking and generalization bounds. *Journal of Machine Learning Research*, 13(1):1373–1392, 2012.

[15] O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. *ICML*, 2013.

[16] S. Smale and Y. Yao. Online learning algorithms. *Found. Comput. Math.*, 6(2): 145–170, 2006.

[17] S. Smale and D. X. Zhou. Online learning with markov sampling. *Analysis and Applications*, 7(1): 87–113, 2009.

[18] Y. Wang, R. Khardon, D. Pechyony, and R. Jones. Generalization bounds for online learning algorithms with pairwise loss functions. *COLT*, 2012.

[19] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10: 207–244, 2009.

[20] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. *NIPS*, 2003.

[21] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, 13(1): 1–26, 2012.

[22] Y. Ying and M. Pontil. Online gradient descent learning algorithms. *Found. Comput. Math.*, 8(5): 561–596, 2008.

[23] Y. Ying and D. X. Zhou. Online pairwise learning algorithms with kernels. *arXiv preprint arXiv:1502.07229*, 2015.

[24] P. Zhao, R. Jin, T. Yang, and S. C. Hoi. Online AUC maximization. *ICML*, 2011.

[25] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *ICML*, 2003.