Nonparametric Sequential Prediction While Deep Learning the Kernel

Guy Uziel
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel

Abstract

The research on online learning under stationary and ergodic processes has been mainly focused on achieving asymptotic guarantees. Although all the methods pursue the same asymptotic goal, their performance varies when handling finite sample datasets and depends heavily on which predefined density estimation method is chosen. In this paper, therefore, we propose a novel algorithm that simultaneously satisfies a short-term goal, to perform as good as the best choice in hindsight of a data-adaptive kernel, learned using a deep neural network, and a long-term goal, to achieve the same theoretical asymptotic guarantee. We present theoretical proofs for our algorithms and demonstrate the validity of our method on the online portfolio selection problem.

1 Introduction

In the traditional online learning setting, and in particular in sequential prediction under uncertainty, the learner is evaluated by a loss function that is not entirely known at each iteration (Cesa-Bianchi and Lugosi, 2006). In this work, we study online prediction but instead of focusing on the well-studied i.i.d. and adversarial settings, we consider nonparametric sequential prediction, which focuses on the challenging case where the unknown underlying process is stationary and ergodic, thus observations depend on each other arbitrarily.

Nonparametric sequential prediction under stationary and ergodic sources has been considered in many papers and various application domains. For example, in online portfolio selection, Györfi and Schäfer (2003); Györfi *et al.* (2006, 2007); Li *et al.* (2011); Uziel and El-Yaniv (2017) proposed nonparametric online strategies that guarantee, under mild conditions, convergence to the best possible outcome. Biau *et al.* (2010); Biau and Patra (2011) considered the setting of time-series prediction. Another line of research worth noting is that of Györfi and Lugosi (2005) regarding the online binary classification problem under such processes.

Although the above strategies are very appealing due to their ability to handle very general processes, they require the use of a countably infinite set of experts, and the guarantees provided for these strategies are always asymptotic. This is no coincidence, as it is well known that finite sample guarantees for these methods cannot be achieved without additional strong assumptions on the source distribution (Luxburg and Schölkopf, 2008; Devroye *et al.*, 2013). Approximate implementations of nonparametric strategies (which apply only a finite set of experts), however, turn out to work exceptionally well and, despite the inevitable approximation, are reported to significantly outperform strategies designed to work in an adversarial or i.i.d., no-regret setting, in various domains (Györfi and Schäfer, 2003; Györfi *et al.*, 2006, 2008; Li *et al.*, 2011).

A common theme in all of these algorithms is that the asymptotically optimal strategies are constructed by combining the predictions of simple experts. The experts are constructed using a single predefined density estimation method where a well-adapted choice of an underlying density method will suppress the performance of a vanilla method on a finite sample dataset. For example, Li *et al.* (2011) carefully designed a kernel suitable for online portfolio selection that outperformed a naive kernel choice used by Györfi *et al.* (2006).

In this paper, we focus on a commonly used density estimation method, the kernel density estimation method (Rosenblatt, 1956; Watson, 1964; Nadaraya, 1964) and on the problem of making sequential predictions while learning a suitable kernel in a data-dependent way.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

We, therefore, propose Online Nonparametric Learning with Kernels (ONLK), which simultaneously learns a set of kernels using a deep learning model while making predictions. The algorithm, as we will prove later on, satisfies two goals simultaneously: a short-term goal—to achieve a bounded regret from the best data-dependent kernel choice in hindsight, and a long-term goal—to produce the same asymptotic guarantee as previous algorithms did.

The paper is organized as follows: In Section 2, we define the nonparametric sequential prediction framework under a jointly stationary and ergodic process and we define the short-and long-term goals of the learner. In Section 3, we present ONLK and provide proofs of its guarantees. In Section 4 we demonstrate our approach to the online portfolio selection problem, comparing ONLK's performance with several well-known algorithms.

2 Problem formulation

We consider the following prediction game. Let $\mathcal{X} \triangleq [-D, D]^d \subset \mathbb{R}^d$ be a compact *observation space* where $D > 0$. At each round, $t = 1, 2, \ldots$, the player is required to make a prediction $y_t \in \mathcal{Y}$, where $\mathcal{Y} \subset \mathbb{R}^m$ is a compact and convex set, based on past observations, $X_1^{t-1} \triangleq (x_1, \ldots, x_{t-1})$ and, $x_i \in \mathcal{X}$ (X_1^0 is the empty observation). After making the prediction y_t, the observation x_t is revealed and the player suffers a loss, $l(y_t, x_t)$, where l is a real-valued continuous function and strongly convex w.r.t. its first argument. We view the player's prediction strategy as a sequence $\mathcal{S} \triangleq \{S_t\}_{t=1}^{\infty}$ of forecasting functions $S_t : \mathcal{X}^{(t-1)} \to \mathcal{Y}$; that is, the player's prediction at round t is given by $S_t(X_1^{t-1})$ (for brevity, we denote $S(X_1^{t-1})$).

Throughout the paper we assume that $x_1, x_2, \ldots$ are realizations of random variables $X_1, X_2, \ldots$ such that the stochastic process $(X_t)_{-\infty}^{\infty}$[1] is jointly stationary and ergodic and $\mathbb{P}(X_i \in \mathcal{X}) = 1$. The player's goal is to play the game with a strategy that minimizes the average l-loss,

$$\frac{1}{T}\sum_{t=1}^{T} l(S(X_1^{t-1}), x_t).$$

The well-known result of Algoet (1994) states that the lowest achievable loss for any online strategy is (without the power of hindsight):

$$\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} l(S(X_1^{t-1}), x_t) \geq \mathbb{E}\left[\max_{y\in\mathcal{Y}()} \mathbb{E}_{\mathbb{P}_\infty}\left[l(y, X_0)\right]\right],$$

where $\mathbb{P}_\infty$ is the regular conditional probability distribution of X_0 given $\mathcal{F}_\infty$ (the σ-algebra generated by the infinite past $X_{-1}, X_{-2}, \ldots$) and the maximization is over the $\mathcal{F}_\infty$-measurable functions. Therefore, we define the optimal quantity as follows:

$$\mathcal{V}^* \triangleq \mathbb{E}\left[\max_{y\in\mathcal{Y}()} \mathbb{E}_{\mathbb{P}_\infty}\left[l(y, X_0)\right]\right].$$

We focus our attention on processes with mixing properties. By mixing property we mean that the process depends weakly on its past. Below we restate the definition of an α-mixing process:

Definition 1 (α-mixing process)**.** *Let $\sigma_m = \sigma(X_1^m)$ and $\sigma_{m+t} = \sigma(X_{m+t}^{\infty})$ be the sigma-algebras of events generated by the random variables $X_m = (X_1, X_2, \ldots, X_m)$ and $X_{m+t}^{\infty} = (X_{m+t}, X_{m+t+1}, ...)$, respectively. The coefficient of absolute regularity, α_n, is given by*

$$\alpha_t = \sup_{m\in\mathbb{N}, A\in\sigma_m, B\in\sigma_{m+t}} |\mathbb{P}(B\cap A) - \mathbb{P}(A)\mathbb{P}(B)|.$$

A stochastic process is said to be α-mixing (or strong mixing), if $\alpha_t \to 0$ as $t \to \infty$.

If also

$$\alpha_n \leq ct^{-r}$$

for some positive constants c, r, then we say that the process has an algebraic mixing rate.

Using the above definition, we assume, similar to several earlier papers (e.g., Modha and Masry, 1998; Meir, 2000), that our process is an α-mixing process with an algebraic mixing rate[2]. Additionally, we assume that the underlying process has a bounded density function.

2.1 Nonparametric sequential prediction using kernel density estimation

In recent years many papers have proposed methods for asymptotically converging to $\mathcal{V}^*$ in several significant machine learning problems such as classification (Györfi and Schäfer, 2003), online portfolio selection (Györfi *et al.*, 2007; Li *et al.*, 2011), quantile prediction, and regression (Györfi and Lugosi, 2005).

The common ground for all these methods is that in order to asymptotically converge to the optimal quantity, $\mathcal{V}^*$, they consist of building estimators for $\mathbb{P}_\infty$.

[1] By Kolmogorov's extension theorem, the stationary and ergodic process $(X_n)_1^{\infty}$ can be extended to $(X_n)_{-\infty}^{\infty}$ such that the ergodicity holds for both $n \to \infty$ and $n \to -\infty$ (see, e.g., Breiman, 1992).

[2] Our algorithms apply without these assumptions, with a slight modification in the theoretical analysis.

Building those estimators for such a general stationary and ergodic process is a challenging task and requires the learner to maintain an infinite number of experts at each given time (or a doubly infinite array without the mixing assumption).

This array of experts, which throughout the paper we call *Markovian experts*, are estimators for the conditional probability $\mathbb{P}_{\left\{X_0|X_{-k}^{-1}\right\}}$, $k = 1, 2, \ldots$. All are built based on the available information up to round t. We denote these experts by h_k, $k = 1, 2, \ldots$.

Each Markovian expert, equipped with a window of length k, is looking for similarities between the current context X_{-k}^{-1} and the set of past k-length observations $\{X_{1-t}^{k-t}, \ldots, X_{-k}^{-1}\}$. More specifically, the learner is using a predefined kernel $K(\cdot, \cdot)$, a non-negative bounded and integrable function[3], which in fact serves as a similarity measure between the different observations. Using the kernel, we can estimate the conditional probability in the following way:

First, for a fixed context $X \in \mathbb{R}^{k \times t}$, we define the similarity weights:

$$\omega_i^k = K(X_{i-k}^{i-1}, X) \qquad 1 - t + k \leq i \leq 0.$$

Using these weights we can define a probability measure over $\mathcal{X}$:

$$\mathbb{P}_t^k(A) \triangleq \frac{\sum_{i=1-t+k}^{0} \omega_i^k 1_A(X_i)}{\sum_{i=1-t+k}^{0} \omega_i^k}, \tag{1}$$

where 1_A denotes the indicator function of the set $A \subset [-B, B]^n$. In other words, $\mathbb{P}_t^k(A)$ is the kernel-weighted relative frequency of the vectors among $X_{1-j+k}, \ldots, X_0$ that falls in the set A. Thus, we can define the predictions of the k-Markovian expert to be:

$$h_k(X_{1-t}^{-1}) \triangleq \arg\min_{y \in \mathcal{Y}} \left(\mathbb{E}_{\mathbb{P}_t^k} \left[l(y, x) \right] \right) = \arg\min_{y \in \mathcal{Y}} \left(\frac{\sum_{i=1-t+k}^{0} \omega_i^k l(y, x_i)}{\sum_{i=1-t+k}^{0} \omega_i^k} \right). \tag{2}$$

By definition, $h_k(X_{1-t}^{-1})$ is the minimum of l w.r.t. $\mathbb{P}_t^k$.

Since in general the memory of the underlying stochastic process is unknown (it might not even be Markovian), one has to aggregate an infinite number of such Markovian experts. The aggregation of the experts is done using standard online learning algorithms.

By aggregating the Markovian experts properly, one can guarantee, using an observation made by Algoet (1994), that the average loss will asymptotically converge to $\mathcal{V}^*$.

[3]We assume throughout the paper that the kernel's bandwidth is chosen in accordance with Theorem 2 in Hansen (2008) and thus we ignore it.

It should not be surprising that the empirical performance depends heavily on the selection of the of the kernel K (Györfi and Schäfer, 2003; Györfi *et al.*, 2007; Li *et al.*, 2011). As stated before, previous algorithms considered how to choose the kernel estimator K in a data-dependent way. We, on the other hand, propose a novel mechanism that can handle the online selection of those estimators in a data-dependent way.

In our approach, and as will be discussed in the next section, using the intuition that different time scales exhibit different patterns, for each Markovian expert we assign a deep neural network. Each network produces at each round J-kernels. These kernels are called throughout the paper *kernel experts*. These experts allow the player to choose and aggregate different choices of data-dependent kernels.

Summarizing the above, we define the goals of the learner to be as follows: Generate a sequence of predictions $y_{ALG} \triangleq y_1, y_2 \ldots$, without the power of hindsight, such that the following goals are satisfied simultaneously:

Short-term goal: To generate a sequence of predictions competing with the best kernel K chosen in hindsight, from a reference class of kernels denoted by $\mathcal{K}$. We, therefore, define the regret w.r.t. reference class $\mathcal{K}$ as follows:

$$\textbf{Regret}(y_{ALG}, \mathcal{K}, T) \triangleq \sum_{t=1}^{T} l(y_t, X_t) - \min_{K \in \mathcal{K}} \sum_{t=1}^{T} l(y_t^K, X_t),$$

where y_t^K are the predictions generated by the learner's algorithm using kernel K. Accordingly, we aim to achieve sublinear regret w.r.t $\mathcal{K}$.

Long-term goal: To ensure that the sequence of predictions converges asymptotically to $\mathcal{V}^*$:

$$\liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} l(y_t, X_t) = \mathcal{V}^*, \tag{3}$$

a.s., and, therefore, maintain the same guarantee as existing algorithms.

3 Online Nonparametric Learning with Kernels (ONLK)

We now present our algorithm Online Nonparametric Learning with Kernels (ONLK). ONLK is illustrated in Algorithm 1 and comprises the three components discussed below.

Algorithm 1 Online Nonparametric Learning with Kernels (ONLK)

Input: Memory bound $\hat{d}$
For $t = 0$ **to** T
  Play y_t.
  Nature reveals x_t
  Suffer loss $l(y_t, x_t)$.
  For $i = 1$ **to** $\hat{d}$
    For $j = 1$ **to** M
      Update the cumulative loss for (i,j)-kernel $l_t^{i,j}$
      Update the kernels using Equation (5) and the predictions $y_{t+1}^{i,j}$ using Equation (2)
      Update the kernel's weights

$$\beta_t^{(i,j)} \triangleq \exp\left(-\frac{1}{\sqrt{t}} l_t^{i,j}\right) \quad p_t^{(i,j)} \triangleq \frac{\beta_t^{(i,j)}}{\sum_{s=1}^{M} \beta_t^{(i,s)}}$$

    Update y_{t+1}^i as follows

$$y_{t+1}^i = \sum_{s=1}^{M} p_{t+1}^{(i,s)} y_{t+1}^{i,s}$$

    Update the loss of the Markovian expert l_t^i and the experts' weights

$$\beta_t^i \triangleq \exp\left(-\frac{1}{\sqrt{t}} l_t^i\right) \quad p_{t+1}^k \triangleq \frac{\beta_{t+1}^i}{\sum_{s=1}^{\hat{d}} \beta_{t+1}^s}$$

  Choose y_{t+1} as follows

$$y_{t+1} = \sum_{s=1}^{\hat{d}} p_{t+1}^s y_{t+1}^s$$

End For

Aggregation between the Markovian experts At each given round we aggregate the predictions of the Markovian experts. This is done using an instance of the well-known Weak Aggregating Algorithm (WAA) (Kalnishkan and Vyugin, 2005; Vovk, 2007)[4], the algorithm puts more weight on more successful experts.

Aggregation between the kernel experts Each Markovian kernel possesses J kernel experts, each of which is a different kernel choice learned in a data-dependent way. The aggregation between the different choices is done by using a second instance of the WAA algorithm.

Learning the kernel experts The kernel experts are learned using recent innovations in the field of deep metric learning, which is an emerging field in metric learning, whose goal in general is to learn a similarity function or a distance between samples from training data. Metric learning and, in particular, deep metric learning is widely applied in various computer vision tasks [5]. Recently, Li *et al.* (2018b); Gao *et al.* (2019) suggested methods for learning deep metric presentations in an online manner. Both approaches consist of stacking several fully connected layers where each layer represents a different similarity measure. This method exploits the fact that shallow layers tend to converge faster than deeper ones, thus generating the learner metrics that are suitable for early stages in the game. Later on, the deeper layers converge as well, suggesting to the learner rich and deep patterns. This approach is more suitable for the online learning setting, for several reasons. First, in the online setting, validation data is missing, and thus it is hard to train a

[4]The choice of WAA is arbitrary and could have been replaced with any other no-regret expert learning algorithms such as EG or ONS (Helmbold *et al.*, 1998; Hazan *et al.*, 2007).

[5]For an extensive survey on this field see, e.g., Gao *et al.* (2019) and the references within

neural model as done in the batch setting. Second, the approach allows training the neural network fast and on-the-fly, and third, stacking the layers allows them to share information.

More formally, for each Markovian expert h_k, we assign a neural network with J fully connected layers to exploit different patterns that may occur on different timescales. We denote the corresponding matrix of layer j by M_j. At time t, and given the context window $s = X_{t-k}^{t-1}$, we first create a triplet, (s^0, s_p^0, s_n^0), where s_p^0 denotes an example that is desired to be close to the context window and s_n^0 that is supposed to be far from this window. In practice, the triplets can be generated using one data pass as described in Li *et al.* (2018a).

The triplet serves as an input to the first fully connected layer. The matrix is decomposed into $M_1 = L_1^T L_1$ and a transformed triplet is created,

$$(s^1 = L_1^T s^0, s_p^1 = L_1^T s_p^0, s_n^1 = L_1^T s_n^0). \tag{4}$$

This triplet passes through a ReLU activation and enters the second fully connected layer, and so on until the final layer, layer J. Resulting, at the end of the process, in $M_1, \ldots, M_J$ matrices. To each layer we attach the following local loss,

$$f(M, (s^0, s_p^0, s_n^0)) \\ = \max\{0, 1 + D_M(s^0, s_p^0) - D_M(s^0, s_n^0)\},$$

where $D_M(\cdot, \cdot)$ is the Mahalanobis distance induced by matrix M. This loss encourages the layer to find matrix M, which separates the two examples with a large margin. Inspired by the method of online passive-aggressive updates (Crammer *et al.*, 2006), Li *et al.* (2018b) suggested to minimize f using gradient updates, the following closed form formula should occur,

$$M_t = \begin{cases} M_{t-1} - \gamma A_t & f > 0 \\ M_{t-1} & f = 0. \end{cases} \tag{5}$$

for $A_t = (s^0 - s_p^0)(s^0 - s_p^0)^T - (s^0 - s_n^0)(s^0 - s_n^0)^T$.

Moreover, as proved in Li *et al.* (2018b), by carefully choosing γ, we can guarantee that the matrices remain semi-positive during the updates, and thus can be used to induced a metric.

The induced matrices, $M_{i,1}, \ldots, M_{i,J}$, are used to construct the following Mahalanobis kernel for a given bandwidth θ,

$$K_{i,j}(x, y) = \begin{cases} D_{M\,i,j}(x, y) & D_{M\,i,j}(x, y) < \theta \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where we denote $K_{i,j}$ as the kernel generated by layer j of the network attached to Markovian expert i. Using Equation (2), prediction $y_t^{i,j}$ can be generated from $K_{i,j}$.

The architecture that was described above is illustrated in Figure 1.

We note that despite the fact that a global loss (e.g., triplet loss) can be attached to the last layer as well, and be trained using backpropagation, the authors found that training each network layer using the updates presented above is beneficial and produces solid results on several benchmarks.

Summarizing the above, at time step t, each Markovian expert h_i has J possible predictions $y_{t+1}^{i,j}$ $j = 1, \ldots, J$, generated by its kernel experts. The aggregation between the different kernel experts is done, as explained before, by applying an instance of the WAA, resulting in the prediction of the Markovian expert y_{t+1}^i. The aggregation of all the predictions of the Markovian expert, by applying another instance of the WAA, results in the final prediction of the algorithm y_{t+1}^i

We now describe the pseudo-code of ONLK. ONLK gets, as a hyperparameter, a memory bound $\hat{d}$, which is needed for finite execution time. After a new observation is revealed (line 3), the cumulative loss up to time t for each kernel expert, $l_t^{i,j}$ $1 \le j \le J, 1 \le i \le \hat{d}$, is updated (line 8). Afterwards, each kernel updates its prediction $y_{t+1}^{i,j}$ using Equation 2 (line 9). For each Markovian expert, $1 \le i \le \hat{d}$, we run $\hat{d}$ instances of the WAA algorithm (lines 10-11) to aggregate the kernel expert predictions, resulting in the final predictions $y_{t+1}^1, \ldots, y_{t+1}^{\hat{d}}$ of the Markovian experts (line 11). The final prediction of ONLK, for the next round, y_{t+1}, is received after the aggregation of $y_{t+1}^1, \ldots, y_{t+1}^{\hat{d}}$ using the outer WAA instance (lines 12-14).

3.1 Theoretical Guarantee

We state and prove the theoretical guarantee of ONLK. Showing that it indeed satisfies the short-term goal and the long-term goal simultaneously.

Theorem 1 (ONLK). *Let $y_1, y_2, \ldots$ be the predictions generated by ONLK when applied on a set of kernels $\mathcal{K} = \{K_1 \ldots, K_{J*\hat{d}}\}$. Then the following holds:*

$$\sum_{t=1}^{T} l(y_t, X_t) - \min_{K \in \mathcal{K}} \sum_{t=1}^{T} l(y_t^K, X_t) \le O(\sqrt{T}), \tag{7}$$

where $y_1^K, y_2^K, \ldots$ are the predictions made by kernel $K \in \mathcal{K}$. If, moreover, we apply ONLK using the doubling trick over the parameter $\hat{d}$, then ONLK will satisfy the long-term goal as well, generating predictions such that:

$$\liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} l(y_t, X_t) = \mathcal{V}^* \quad a.s. \tag{8}$$

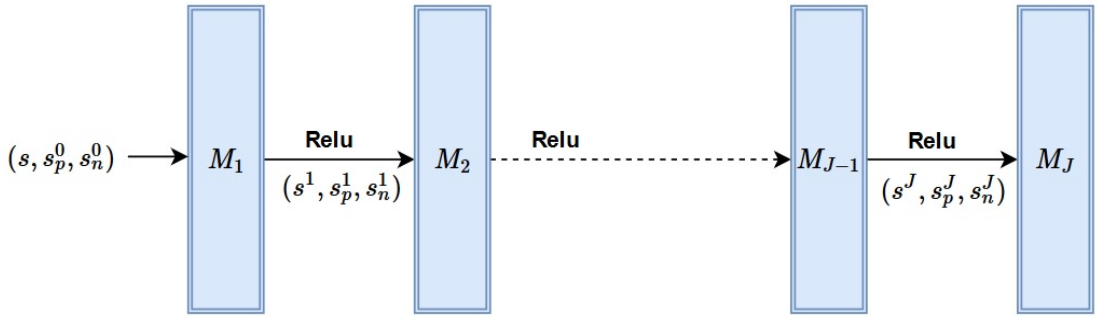

Figure 1: The architecture of ONLK, which was used in our experiments. At time t, for a Markovian expert h_k, the context window s serves as input to the first network layer. The first layer M_1 is used as the first kernel expert. Afterwards, by using a decomposition of M_1, we get a new presentation for the triplets that serves as an input to the second layer, resulting in J different kernels. The weights over the kernel experts are set by an instance of the WAA algorithm.

The proof is presented below. In the proof, we use the following lemma, which is known as Breiman's generalized ergodic theorem:

Lemma 1 (Ergodicity, Breiman, 1957)**.** *Let* $\mathbf{X} = \{X_i\}_{-\infty}^{\infty}$ *be a stationary and ergodic process. For each positive integer* i*, let* T^i *denote the operator that shifts any sequence by* i *places to the left. Let* $f_1, f_2, \ldots$ *be a sequence of real-valued functions such that* $\lim_{t\to\infty} f_t(\mathbf{X}) = f(\mathbf{X})$ *a.s., for some function* f*. Assume that* $\mathbb{E}\sup_t |f_t(\mathbf{X})| < \infty$*. Then,*

$$\lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{t} f_i(T^i\mathbf{X}) = \mathbb{E}f(\mathbf{X})$$

a.s.

Proof. The proof is divided into two parts. In the first part, we show that the short-term goal is indeed satisfied by using the guarantees of the WAA algorithm and by showing that the outer algorithm performs as good as the best Markovian expert. In the second part of the proof, we show that $\mathcal{V}^*$ is achievable by the Markovian experts and, therefore, can be achieved by ONLK.

First step The first step of the proof is to show that the predictions generated by the inner algorithm perform as good as the predictions made by using the best-fixed kernel at time T. Since we use the WAA as a sub-routine for every $1 \le i \le \hat{d}$ and by applying Lemma 11 in Kalnishkan and Vyugin (2005):

$$\frac{1}{T}\sum_{t=1}^{T} l(y_t^i, x_t) \le \min_{j\in[1,J]} \frac{1}{T}\sum_{t=1}^{T} l(y_t^{i,k}, x_t) + \frac{C'_{i,k}}{\sqrt{T}}, \quad (9)$$

where $C'_{i,k} > 0$ is a constant, independent of T. By applying the same theorem again but now on the outer algorithm, we get that

$$\frac{1}{T}\sum_{t=1}^{T} l(y_t, x_t) \le \min_{i\in[1,\hat{d}]} \min_{j\in[1,J]} \frac{1}{T}\sum_{t=1}^{T} l(y_t^{i,k}, x_t) + \frac{C_{i,k}}{\sqrt{T}}, \quad (10)$$

and thus ONLK satisfies the short-term goal.

Second step We just showed that we perform as good as any other expert. As discussed earlier in the paper, since the process is not assumed to have bounded memory, we must aggregate all the Markovian experts. This is done using the doubling trick over the parameter $\hat{d}$, i.e., dividing the timeline into intervals of length $T = 2^i, i > 0$ and applying ONLK on each interval with $\hat{d} = 2^i$ (see, e.g., Cesa-Bianchi and Lugosi, 2006), resulting in a bounded regret from any Markovian kernel. In particular, using Equation (10),

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} l(y_t, x_t) \le \inf_{k,j}\left(\frac{1}{T}\sum_{t=1}^{T} l(y_t^{k,j}, x_t) + \frac{C_{k,j}}{\sqrt{T}}\right).$$

Therefore, we get

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}l(y_t,x_t)\leq$$
$$\limsup_{T\to\infty}\inf_{k,j}\left(\frac{1}{T}\sum_{t=1}^{T}l(y_t^{k,j},x_t)+\frac{C_{k,j}}{\sqrt{T}}\right)\leq$$
$$\inf_{k,j}\limsup_{T\to\infty}\left(\frac{1}{T}\sum_{t=1}^{T}l(y_t^{k,j},x_t)+\frac{C_{k,j}}{\sqrt{T}}\right)\leq$$
$$\inf_{k,j}\limsup_{T\to\infty}\left(\frac{1}{T}\sum_{t=1}^{T}l(y_t^{k,j},x_t)\right)\leq$$
$$\limsup_{k\to\infty}\min_{j}\limsup_{T\to\infty}\left(\frac{1}{T}\sum_{t=1}^{T}l(y_t^{k,j},x_t)\right), \qquad (11)$$

where in the last inequality we used the fact that lim sup is sub-additive. Concluding the above, it is enough to show that

$$\limsup_{k\to\infty}\min_{j}\limsup_{T\to\infty}\left(\frac{1}{T}\sum_{t=1}^{T}l(y_t^{k,j},x_t)\right)=\mathcal{V}^*. \qquad (12)$$

To show it, first analyze the asymptotic average loss of a fixed expert (k,j) for $1\leq k, 1\leq j\leq J$. As we saw in Equation (1), this kernel forms a (random) probability measure $\mathbb{P}_t^{(k,j)}$. Using the uniform (weak) convergence of kernels for α–mixing processes (Theorem 2 in Hansen, 2008), we get that

$$\mathbb{P}_t^{(k,j)}\to\mathbb{P}_{\left\{X_0|X_{-k}^{-1}\right\}} \qquad (13)$$

weakly, as t tends to ∞. Since l is strongly convex, $\arg\min_{y\in\mathcal{Y}}\left(\mathbb{E}_{\mathbb{P}_{\left\{X_0|X_{-k}^{-1}\right\}}}\left[l(y,x)\right]\right)$ is a singleton. Thus, by using Lemma 2 in Algoet (1994) we get that a.s.

$$\arg\min_{y\in\mathcal{Y}}\left(\mathbb{E}_{\mathbb{P}_t^{(k,j)}}\left[l(y,x)\right]\right)\to$$
$$\arg\min_{y\in\mathcal{Y}}\left(\mathbb{E}_{\mathbb{P}_{\left\{X_0|X_{-k}^{-1}\right\}}}\left[l(y,x)\right]\right).$$

Thus, we can apply Lemma 1 and conclude that as T approaches ∞,

$$\frac{1}{T}\sum_{t=1}^{T}l(y_t^{k,j},X_t)\to\mathbb{E}\left[l(y_k^*,X_0)\right]$$

a.s., where $y_k^*\triangleq\arg\min_{y\in\mathcal{Y}}\left(\mathbb{E}_{\mathbb{P}_{\left\{X_0|X_{-k}^{-1}\right\}}}\left[l(y,x)\right]\right)$, i.e., the optimal selection w.r.t. $\mathbb{P}_{\left\{X_0|X_{-k}^{-1}\right\}}$.

To finish the proof, we apply the supermartingale convergence theorem (see, e.g., Stout, 1974). First note that the sequence

$$Z_k\triangleq\mathbb{E}\left[l(y_k^*,X_0)\mid X_{-k}^{-1}\right]$$

is a supermartingale. We can see this by using the tower property of conditional expectations,

$$\mathbb{E}[Z_{k+1}\mid X_{-k}^{-1}]=\mathbb{E}\left[\mathbb{E}\left[Z_{k+1}\mid X_{-k-1}^{-1}\right]\mid X_{-k}^{-1}\right],$$

and since Z_{k+1} is the optimal choice in $\mathcal{Y}$ w.r.t. to X_{-k-1}^{-1},

$$\leq\mathbb{E}\left[\mathbb{E}[Z_k\mid X_{-k-1}^{-1}]\mid X_{-k}^{-1}\right]=\mathbb{E}[Z_k\mid X_{-k}^{-1}]=Z_k.$$

Note also that $\mathbb{E}[Z_k]$ is uniformly bounded. Therefore, we can apply the supermartingale convergence theorem and get that $Z_k\to Z_\infty$ a.s. as k tends to ∞, where,

$$Z_\infty=\mathbb{E}\left[l(y_\infty^*,X_0)\mid X_{-\infty}^{-1}\right]=\mathcal{V}^*,$$

and by using Lebesgue's dominated convergence theorem, also $\mathbb{E}[Z_k]\to\mathbb{E}[Z_\infty]=\mathcal{V}^*$, which concludes the proof.

□

4 Empirical Study

4.1 Online Portfolio Selection

Online portfolio selection (Cover, 1991) is a challenging and a long-standing problem. In this problem the learner maintains an online allocation vector, called a portfolio, specifying the fraction of wealth to be invested in all the stocks in the market. At the start of each trading period (e.g., day), the learner receives the current prices of the stocks and submits his next day's portfolio to his broker with the hope that his chosen stocks will rise in price. Formally speaking, we are given a market with n stocks. On each day t, the market is represented by a *market vector* $\mathbf{X}_t$ of relative prices, $\mathbf{X}_t\triangleq(x_1^t,x_2^t,...,x_n^t)$, where for each $i=1,\ldots,n$, $0<c_1\leq x_i^t\leq c_2$ for some constants c_1,c_2, the *relative price* of stock i is defined to be the ratio of its closing price on day t relative to its closing price on day $t-1$.

The algorithm's *portfolio* for day t is $\mathbf{b}_t\triangleq(b_1^t,b_2^t,\ldots,b_n^t)$, where $b_i^t\geq 0$ is the wealth allocation for stock i. We require that the portfolio satisfy $\sum_{i=1}^m b_i^t=1$. Thus, $\mathbf{b}_t$ specifies the online player's wealth allocation for each of the n stocks on day t, and b_i^t is the fraction of the total current wealth invested in stock i on that day.

At the start of each trading day t, the algorithm chooses a portfolio $\mathbf{b}_t$. Thus, by the end of day t, the player's wealth is multiplied by $\langle\mathbf{b}_t,\mathbf{X}_t\rangle=\sum_{i=1}^n b_i^t x_i^t$ and, assuming an initial wealth of \$1, the player's cumulative wealth by the end of the game is

$$R(\mathbf{B},\mathbf{X}_0^{T-1})\triangleq\prod_{t=0}^{T-1}\langle\mathbf{b}_t,\mathbf{X}_t\rangle, \qquad (14)$$

Table 1: Cumulative wealth of known algorithms and ONLK assuming an initial wealth of \$1

Dataset	Index	UCRP	BCRP	UP	$\mathbf{B}_k$	CORN	BK	BM	ONLK
NYSE	1.47	1.52	1.77	1.51	1.89	2.16	2.55	2.52	2.48
SP500	1.79	1.81	2.03	1.86	2.14	2.42	3.01	2.93	2.91

where $\mathbf{B} \triangleq \mathbf{b}_0, \ldots, \mathbf{b}_{T-1}$ is the sequence of T portfolios played by the algorithm for the entire game.

We compare the performance of ONLK to the following benchmark algorithms: (i) Index: Setting a uniform weight over the stocks, with the allocation afterwards remaining untouched; (ii) UCRP: Uniform portfolio, where the allocation is rebalanced back to equal weights each time the prices are revealed; (iii) BCRP: The best fixed rebalancing portfolio calculated in hindsight, which is the best strategy in i.i.d. markets; (iv) UP: Universal portfolios (Cover and Ordentlich, 1996), an algorithm that tracks the BCRP; (v) $\mathbf{B}_k$: The kernel algorithm suggested by Györfi and Lugosi (2005) for stationary and ergodic markets using a naive choice of a Euclidean kernel; (vi) CORN: An improvement over the $\mathbf{B}_k$ algorithm, by using handcrafted kernels; (vii) BM: The best Markovian expert calculated in hindsight. (vii) BK: The best kernel expert calculated in hindsight. All the algorithms were implemented with the same parameters that were suggested by their authors.

Table 2: Some properties of the datasets

Dataset	Starting day	# Days	# stocks
NYSE	1/1/2009	2000	20
SP500	1/1/2009	2000	30

We used two datasets, NYSE and SP500. The datasets contain stocks with the largest market value at the start period. These datasets span several types of market conditions and amounts of stocks (see Table 2).

4.2 Implementation and results

The main objectives of our experiments are to check how well ONLK tracks the best kernel expert and the best Markovian expert and to examine the usefulness of exploiting a data-dependent kernel instead of using a naive kernel or handcrafted one, especially in the case where the data is hard to model. To apply our ONLK strategy, and for learning the kernels, we used a 5-layer DNN with ReLU activation between the layers. The size of each hidden layer was equal to the size of the input vector (number of stocks) in the corresponding dataset. The network was trained using the updates in Equation (5) with $\gamma = 0.01$ as suggested by Li *et al.* (2018b). ONLK was applied with $\hat{d} = 5$, resulting in five Markovian experts (as suggested by Györfi *et al.*, 2008). The bandwidth of all the kernels was chosen to be $\theta = \frac{1}{\sqrt{t}}$. The experiments were implemented using Keras (Chollet and others, 2015).

Moreover, since the network is using triplets as an input, we used the observation made by several heuristic algorithms that exploiting statistical phenomenon, such as mean reversion, in the markets can be beneficial. One approach was an algorithm called Anticor (Borodin *et al.*, 2000). Accordingly, to exploit this observation for a given context window, we generated a positive example, which is the most correlated sample, and a negative, one which possesses the lowest correlation, all measured using the Pearson correlation.

In Table 1, we present the final wealth achieved by the different algorithms in the two datasets that were used. It can be seen that our approach to making predictions while learning the kernel results demonstrates superior performance compared to the index, UP and BCRP algorithms and to the the choice of the naive kernel $\mathbb{B}_k$ and to the handcrafted kernel CORN. Moreover, by comparing the result of ONLK to the results attained by the best layer BL and by the best Markovian expert, BM (both calculated in hindsight), we can observe that our algorithm tracks both successfully.

5 Conclusions

In this paper we tackled the problem of learning and using a data-dependent kernel, learned online using a neural network, in the context of nonparametric sequential prediction under stationary and ergodic processes. This success reduces the need for pre-choosing a well-suited kernel for the specific problem. We presented ONLK, which can achieve the short-term and long-term goals simultaneously, performing as good as the best choice in hindsight of a kernel, while maintaining the same asymptotic guarantee as previous algorithms.

In future work, we wish to extend our approach to deal with another well-known density estimation method, the nearest neighbour method (Györfi *et al.*, 2008), where one has to pre-choose a metric. Online metric choosing might need a different theoretical analysis and thus may not be directly deduced from the proof presented in this paper.

References

P.H. Algoet. The strong law of large numbers for sequential decisions under uncertainty. *IEEE Transactions on Information Theory*, 40(3):609–633, 1994.

G. Biau and B. Patra. Sequential quantile prediction of time series. *IEEE Transactions on Information Theory*, 57(3):1664–1674, 2011.

G. Biau, K. Bleakley, L. Györfi, and G. Ottucsák. Nonparametric sequential prediction of time series. *Journal of Nonparametric Statistics*, 22(3):297–317, 2010.

A. Borodin, R. El-Yaniv, and V. Gogan. On the competitive theory and practice of portfolio selection. In *LATIN 2000: Theoretical Informatics*, pages 173–196. 2000.

L. Breiman. The individual ergodic theorem of information theory. *The Annals of Mathematical Statistics*, 28(3):809–811, 1957.

Leo Breiman. Probability, volume 7 of classics in applied mathematics. *Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA*, 1992.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

François Chollet et al. Keras. `https://keras.io`, 2015.

T.M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996.

T.M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Y. Gao, Y. Li, S. Chandra, L. Khan, and B. Thuraisingham. Towards self-adaptive metric learning on the fly. In *The World Wide Web Conference*, pages 503–513. ACM, 2019.

L. Györfi and G. Lugosi. Strategies for sequential prediction of stationary time series. In *Modeling Uncertainty*, pages 225–248. Springer, 2005.

L. Györfi and D. Schäfer. Nonparametric prediction. *Advances in Learning Theory: Methods, Models and Applications*, 339:354, 2003.

L. Györfi, G. Lugosi, and F. Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337–357, 2006.

L. Györfi, A. Urbán, and I. Vajda. Kernel-based semi-log-optimal empirical portfolio selection strategies. *International Journal of Theoretical and Applied Finance*, 10(03):505–516, 2007.

L. Györfi, F. Udina, and H. Walk. Nonparametric nearest neighbor based empirical portfolio selection strategies. *Statistics & Decisions, International Mathematical Journal for Stochastic Methods and Models*, 26(2):145–157, 2008.

B. Hansen. Uniform convergence rates for kernel estimation with dependent data. *Econometric Theory*, 24(3):726–748, 2008.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

D.P. Helmbold, R.E. Schapire, Y. Singer, and M.K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

Y. Kalnishkan and M. Vyugin. The weak aggregating algorithm and weak mixability. In *International Conference on Computational Learning Theory*, pages 188–203. Springer, 2005.

B. Li, S.C.H Hoi, and V. Gopalkrishnan. Corn: Correlation-driven nonparametric learning approach for portfolio selection. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):21, 2011.

W. Li, W. Gao, L. Wang, L. Zhou, J. Huo, and Y. Shi. Opml: A one-pass closed-form solution for online metric learning. *Pattern Recognition*, 75:302–314, 2018.

W. Li, J. Huo, Y. Shi, Y. Gao, L. Wang, and J. Luo. Online deep metric learning. *arXiv preprint arXiv:1805.05510*, 2018.

U.V. Luxburg and B. Schölkopf. Statistical learning theory: Models, concepts, and results. *arXiv preprint arXiv:0810.4752*, 2008.

R. Meir. Nonparametric time series prediction through adaptive model selection. *Machine Learning*, 39(1):5–34, 2000.

D. Modha and E. Masry. Memory-universal prediction of stationary random processes. *IEEE Transactions on Information Theory*, 44(1):117–133, 1998.

E. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, pages 832–837, 1956.

W. Stout. Almost sure convergence, vol. 24 of probability and mathematical statistics, 1974.

G. Uziel and R. El-Yaniv. Growth-optimal portfolio selection under cvar constraints. *arXiv preprint arXiv:1705.09800*, 2017.

V. Vovk. Competing with stationary prediction strategies. In *International Conference on Computational Learning Theory*, pages 439–453. Springer, 2007.

G. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.