An Optimal Reduction of TV-Denoising to Adaptive Online Learning

Dheeraj Baby
dheeraj@ucsb.edu

Xuandong Zhao
xuandongzhao@ucsb.edu

Yu-Xiang Wang
yuxiangw@cs.ucsb.edu

Dept. of Computer Science, UC Santa Barbara

Abstract

We consider the problem of estimating a function from n noisy samples whose discrete Total Variation (TV) is bounded by C_n. We reveal a deep connection to the seemingly disparate problem of *Strongly Adaptive* online learning (Daniely et al., 2015) and provide an $O(n \log n)$ time algorithm that attains the near minimax optimal rate of $\tilde{O}(n^{1/3}C_n^{2/3})$ under squared error loss. The resulting algorithm runs online and optimally *adapts* to the *unknown* smoothness parameter C_n. This leads to a new and more versatile alternative to wavelets-based methods for (1) adaptively estimating TV bounded functions; (2) online forecasting of TV bounded trends in time series.

1 Introduction

Total variation (TV) denoising (Rudin et al., 1992) is a classical algorithm originated in the signal processing community which removes noise from a noisy signal y by solving the following regularized optimization problem

$$\min_f \|f-y\|_2^2 + \lambda \mathrm{TV}(f).$$

where $\mathrm{TV}(\cdot)$ denotes the total variation functional which is equivalent to $\int |f'(x)|dx$ for weakly differentiable functions. In discrete time, TV denoising is known as "fused lasso" in the statistics literature (Tibshirani et al., 2005; Hoefling, 2010), which solves

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \sum_{i=1}^n (\theta_i - y_i)^2 + \lambda \sum_{i=2}^n |\theta_i - \theta_{i-1}|. \qquad (1)$$

where θ_i is the element at index i of the vector $\boldsymbol{\theta}$. Unlike their L2-counterpart, the TV regularization functional is designed to promote sparsity in the number of change points, hence inducing a "piecewise constant" structure in the solution.

Over the three decades since the advent of TV denoising, it has seen many influential applications. Algorithms that use TV-regularization has been deployed in every cellphone, digital camera and medical imaging devices. More recently, TV denoising is recognized as a pivotal component in generating the first image of a super massive black hole (Akiyama et al., 2019). Moreover, the idea of TV regularization has inspired a myriad of extensions to other tasks such as image debluring, super-resolution, inpainting, compression, rendering, stylization (we refer readers to a recent book (Chambolle et al., 2010) and the references therein) as well as other tasks beyond the context of images such as change-point detection, semisupervised learning and graph partitioning.

In this paper, we focus on the *non-parametric statistical estimation* problem behind TV-denoising which aims to estimate a function $f : [0,1] \to \mathbb{R}$ using observations of the following form:

$$y_i = f(x_i) + \epsilon_i, i \in [n] := \{1, \ldots, n\},$$

where ϵ_i are iid $N(0, \sigma^2)$ and the function f belongs to some fixed non-parametric function class $\mathcal{F}$. The exogenous variables x_i belongs to some subset $\mathcal{X}$ of $\mathbb{R}$. The above setup is a widely adopted one in the non-parametric regression literature (Tsybakov, 2008). In this work, we take $\mathcal{F}$ to be the Total Variation class: $\{f | \mathrm{TV}(f) \leq C_n\}$ or its discrete counterpart

$$\mathcal{F}(C_n) := \left\{ f \middle| \sum_{t=2}^n |f(x_t) - f(x_{t-1})| \leq C_n \right\}.$$

We are interested in finding algorithms that generate estimates $\hat{y}_t, t \in [n]$ such that the total square error

$$R_n(\hat{y}, f) := \sum_{t=1}^n \mathbb{E}[(\hat{y}_t - f(x_t))^2],$$

is minimized. Throughout this paper, when we refer to *rate*, we mean the growth rate of R_n as a function of n

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

and C_n. The family $\mathcal{F}(C_n)$ we consider here features a rich class of functions that exhibit spatially heterogeneous smoothness behavior. These functions can be very smoothly varying in certain regions of space, while in other regions, it can exhibit fast variations (see for eg. Fig. 5) or abrupt changes that may even be discontinuous. A good estimator should be able to detect such local fluctuations (which can be short lived) and adjust the amount of "smoothing" to apply according to the level of smoothness of the functions in each local neighborhood. Such estimators are referred as *locally adaptive estimators* by Donoho (Donoho et al., 1998).

We are interested in algorithms that achieve the minimax optimal rates for estimating functions in $\mathcal{F}(C_n)$ defined as:

$$R_n^*(C_n) = \inf_{\{\hat{y}_t\}_{t=1}^n} \sup_{f \in \mathcal{F}(C_n)} R_n(\hat{y}, f),$$

which is known to be $\Theta(n^{1/3}C_n^{2/3})$(Donoho et al., 1990; Mammen, 1991).

There is a body of work in *Strongly Adaptive online learning* that focuses on designing online algorithms such that its regret in any local time window is controlled (Daniely et al., 2015). Hence the notion of local adaptivity is built into such algorithms. This makes the problem of estimating TV bounded functions, a natural candidate to be amenable to techniques from Strongly Adaptive online learning. However, it is not clear that whether using Strongly Adaptive algorithms can lead to minimax optimal estimation rates. By formalizing the intuition above, we answer it affirmatively in this work.

We reserve the phrase *adaptive estimation* to describe the act of estimating TV bounded functions such that R_n of the estimator/algorithm can be bounded by a function of n and C_n without any prior knowledge of C_n. An *adaptively optimal* estimator $\hat{y}$ is able to estimate an arbitrary function f with an error

$$R_n(\hat{y}, f) = \tilde{O}\Big(\inf_{C_n \text{ such that } f \in \mathcal{F}(C_n)} R_n^*(C_n)\Big).$$

A TV bounded function will be referred as a Bounded Variation (BV) function henceforth for brevity.The notation $\tilde{O}(\cdot)$ hides poly-logarithmic factors of n.

It is well known that *all linear estimators* that output a linear transformation of the observations attain a suboptimal $\Omega(\sqrt{nC_n})$ rate (Donoho et al., 1990). This covers a large family of algorithms including the popular methods based on smoothing kernels, splines and local polynomials, as well as methods such as online gradient descent (see a recent discussion from Baby and Wang, 2019). Wavelet smoothing (Donoho et al., 1998) is known to attain the near minimax optimal rate of $\tilde{O}(n^{1/3}C_n^{2/3})$ for R_n without any prior information about C_n. Recently the same rate is shown to be achievable for the online forecasting setting by adding a wavelets-based adaptive restarting schedule to OGD (Baby and Wang, 2019).

In this paper, we provide an alternative to wavelet smoothing by a novel reduction to a strongly adaptive regret minimization problem from the online learning literature. We show that the resulting algorithm achieves the same adaptive optimal rate of $\tilde{O}(n^{1/3}C_n^{2/3})$. The algorithm is more versatile than wavelet smoothing for three reasons:

1. Our algorithm is based on aggregating experts that performs local predictions. The experts we use perform online averaging. However, one may use more advanced algorithms such as kernel/spline smoothing, polynomial regression or even deep learning approaches as experts that can potentially lead to better performance in practice. Hence our algorithm is highly configurable.

2. Our algorithm accepts a learning rate parameter that can be set without prior knowledge of C_n to obtain the near optimal rate of $\tilde{O}(n^{1/3}C_n^{2/3})$ (see Theorem 5). However, this learning rate can also be tuned using heuristics that can lead to better practical performance (see Section 5).

3. It can also handle a more challenging setting where the data are streamed sequentially in an online fashion.

To the best of our knowledge, we are the first to formalize the connection between strongly adaptive online learning and the problem of local-adaptivity in nonparametric regression. By establishing this new perspective, we hope to encourage further collaboration between these two communities.

1.1 Problem Setup

Though we are primarily motivated to solve the offline/batch estimation problem, our starting point is to consider a significant generalization of the batch problem as shown in Fig. 1. Any adaptively optimal algorithm to this online game immediately implies adaptive optimality in the batch/offline setting. For example, to solve the batch problem, adversary can be thought of as revealing the indices isotonically, i.e $i_t = t$. However, note that in the online game, adversary can even query the same index multiple times. The term "forecasting strategy" in step 1 of Fig. 1, is used to mean an algorithm that makes a prediction at current time point only based on the historical data.

Solving the online problem has an added advantage that the resulting algorithm can be applied to various instances of time series forecasting like financial markets, spread of contagious disease etc.

Assumption 1 $|f(x_i)| \leq B, \forall i \in [n]$ for some known B.

Though this constraint is considered to be mild and natural,

1. Player (we) declares a forecasting strategy
2. Adversary chooses an $\mathcal{X} = \{x_1 < x_2 < \ldots < x_n\}$ and reveals it to the player.
3. Adversary chooses $f(x_1), \ldots, f(x_n)$ such that $\sum_{t=2}^{n} |f(x_t) - f(x_{t-1})| \leq C_n$.
4. Adversary fixes an ordered set $\{i_1, \ldots, i_n\}$ where each $i_j \in [n]$.
5. For every time point $t = 1, ..., n$:
   (a) Adversary reveals i_t.
   (b) We play $\hat{y}_t$.
   (c) We receive a feedback $y_t = f(x_{i_t}) + \epsilon_t$, where ϵ_t is $N(0, \sigma^2)$.
   (d) We suffer loss $(\hat{y}_t - y_t)^2$
6. Our goal is to minimize $\sum_{t=1}^{n} \mathbb{E}[(\hat{y}_t - f(x_{i_t}))^2]$.

Figure 1: *Online interaction protocol*

we note that standard non-parametric regression algorithms do not make this assumption.

1.2 Notes on novelty and contributions

To the best of our knowledge, in non-parametric regression literature, only wavelet smoothing [1] (Donoho et al., 1998) is able to *provably* attain a near optimal $\tilde{O}(n^{1/3} C_n^{2/3})$ rate for estimating BV functions in batch setting without knowing the value of C_n. There are model-selection techniques based on information-criterion, which often either incurs significant practical overhead or comes with no optimal rate guarantees (We will review these approaches in Section 1.3).

The contributions of this work is mainly theoretical. Our primary result is a novel reduction from the problem of estimating BV functions to Strongly Adaptive online learning (Daniely et al., 2015). This reduction approach results in the development of a new $O(n \log n)$ time algorithm that is: 1) *minimax optimal* (modulo log factors) 2) *adaptive* to C_n and 3) can be used to tackle *both* online and offline estimation problems thereby providing new insights. To elaborate slightly, this is facilitated by few fundamentally different viewpoints than those adopted in the wavelet literature. In particular, we exhibit a specific partitioning of TV bounded function into consecutive chunks that incurs low total variation such that total number of chunks is $O(n^{1/3} C_n^{2/3})$. Then by designing a strongly adaptive online learner, we ensure an $\tilde{O}(1)$ cumulative squared error in each chunk of that partition. This immediately implies an estimation error rate of $\tilde{O}(n^{1/3} C_n^{2/3})$ when summed across all chunks. To the best of our knowledge, this is the *first* time a connection between strongly adaptive online learning and estimating BV functions has been exploited in literature.

Experimental results (see Section 5) indicate that our algorithm can outperform wavelet smoothing in terms of its cumulative squared error incurred in practice. We demonstrate that the proposed algorithm can be used without any hyperparameter tuning and incurs very low computational overhead in comparison to model selection based approaches for the fused lasso problem (see Eq. (1)).

Before closing this section, we remind the reader that this work shouldn't be viewed only as providing yet another solution to a classical problem but rather one that provides *a fundamentally new set of tools* that adds new insight to this decades-old problem that might have a profound impact in many extensions of the basic setting we consider and other downstream tasks such as estimating higher-dimensional BV functions, fused lasso on graphs, image deblurring, trend filtering and so on.

1.3 Related Work

As noted before, the theoretical analysis of estimating BV functions is well studied in the rich literature of non-parametric regression. Apart from wavelet smoothing (Donoho et al., 1990; Donoho and Johnstone, 1994a,b; Donoho et al., 1998), many algorithms such as Trend Filtering (Kim et al., 2009; Tibshirani, 2014; Wang et al., 2016; Sadhanala et al., 2016; Guntuboyina et al., 2017) and locally adaptive regression splines (Mammen and van de Geer, 1997) can be used for estimation. However, one drawback of these algorithms is that they require the TV of ground truth C_n as an input to the algorithm to guarantee minimax optimal rates. For example, the solution to fused lasso (Eq. (1)) is minimax optimal only when one chooses the hyperparameter λ optimally. It is shown in (Wang et al., 2016) that optimal choice of λ depends on the variational budget C_n which may be unknown beforehand.

Theoretically one may tune the choice C_n (or λ) as a hyperparameter using criteria like AIC, BIC, Stein-Unbiased Risk Estimate (SURE)-based approaches or the use techniques presented in (Birge and Massart, 2001). However, such model selection based schemes often have statistical or computational overheads that make them impractical. The most relevant is the effective degree of freedom (dof) approach (See Eq.(8) and Eq.(9) in (Tibshirani and Taylor, 2012)). It requires solving fused lasso with many λ (computational overhead). The estimate of dof is unstable in some regimes (statistical overhead). Generally, these methods may work well in practice but often do not come with theoretical guarantees of adaptive optimality. Moreover, we are not aware of any such model-selection technique that can solve the online version of the problem.

There is also a body of work that focuses on the computation

[1] Though (Baby and Wang, 2019) proposes a minimax policy for forecasting TV bounded sequences online, they heavily rely on the adaptive minimaxity of wavelet smoothing.

of solving problem (1) and their higher-dimensional extensions (see (Chambolle and Lions, 1997; Barbero and Sra, 2011), and the excellent survey therein). This is complementary to our focus, which is to minimize the error against the (unobserved) ground truth. Computationally, (Johnson, 2013)'s dynamic programming has a worst-case $O(n)$ time-complexity, but only for a fixed λ. Our algorithm runs in $O(n \log n)$-time while avoids choosing the λ parameter all together.

The closest to us is perhaps (Baby and Wang, 2019) which indeed has motivated this work. They consider an online protocol similar to Fig. 1 with the adversary constrained to reveal the indices i_t isotonically (i.e $i_t = t$) and propose an adaptive restart scheme based on wavelets. However such techniques are not useful to compete against a more powerful adversary which can query indices in any arbitrary manner — for example when the exogenous variables $x \in \mathcal{X}$ are sampled iid from a distribution and revealed online. Further, their proof critically relies on adaptive minimaxity of wavelets. We aim to build a radically new algorithm that is agnostic to the results from wavelet smoothing literature.

A strongly adaptive online learner (Daniely et al., 2015; Adamskiy et al., 2016), incurs low static regret in *any* interval. This is accomplished by maintaining a pool of sleeping experts that are static regret minimizing algorithms which are awake only in some specific duration. Then an aggregation strategy to hedge over the experts is used to guarantee low regret in any interval. This work was preceded by the notion of weakly adaptive regret in (Hazan and Seshadhri, 2007). To the best of our knowledge, the efficient reduction of TV-denoising to strongly-adaptive online learning is new to this paper. We defer further discussions on related work to Appendix A.

2 Preliminaries

In this section, we briefly review the elements from online learning literature that are crucial to the development of our algorithm.

2.1 Geometric Cover

Geometric Cover (GC) proposed in (Daniely et al., 2015) is a collection of intervals that belong to $\mathbb{N}$ defined below. In what follows $[a, b]$ denotes the set of natural numbers lie between a and b, both inclusive.

$$\mathcal{I} = \bigcup_{k \in \mathbb{N} \cup \{0\}} \mathcal{I}_k,$$

where $\forall k \in \mathbb{N} \cup \{0\}$, and $\mathcal{I}_k = \{[i \cdot 2^k, (i+1) \cdot 2^k - 1] : i \in \mathbb{N}\}$. Define AWAKE$(t) := \{I \in \mathcal{I} : t \in I\}$.By the construction of Geometric Cover $\mathcal{I}$, it holds that

$$|\text{AWAKE}(t)| = \lfloor \log t \rfloor + 1. \quad (2)$$

Let's denote $\mathcal{I}|_J := \{I \in \mathcal{I} : I \subseteq J\}$ for an interval $J \subseteq \mathbb{N}$. The GC has a very nice property recorded in the following Proposition.

Proposition 1. *(Daniely et al., 2015) Let $I = [q, s] \subseteq \mathbb{N}$. Then the interval I can be partitioned into two finite sequences of disjoint consecutive intervals $(I_{-k}, \ldots, I_0) \subseteq \mathcal{I}|_I$ and $(I_1, \ldots, I_p) \subseteq \mathcal{I}|_I$ such that,*

$$\frac{|I_{-i}|}{|I_{-i+1}|} \le \frac{1}{2}, \forall i \ge 1 \qquad \text{and} \qquad \frac{|I_i|}{|I_{i-1}|} \le \frac{1}{2}, \forall i \ge 2.$$

2.2 Sleeping Experts and Specialist Aggregation Algorithm (SAA)

In the problem of learning from expert advice with outcome space $\mathcal{O}$ and action space $\mathcal{A}$, there are K experts who provide a list of actions $a_{t,:} = [a_{t,1}, ..., a_{t,K}] \in \mathcal{A}^K$ at time $t = 1, ..., n$. The learner is supposed to takes an action $a_t \in \mathcal{A}$ based on the expert advice[2] before the outcome $o_t \in \mathcal{O}$ is revealed by an adversary. The player then incurs a loss given by $\ell(a_t, o_t)$, where ℓ is a loss function.

In the most basic setting, $\mathcal{A}, \mathcal{O}$ are discrete sets, ℓ can be described by a table, and we assign one constant expert to each $a \in \mathcal{A}$, then this becomes an online version of Von Neumann's linear matrix game. More generally, $\mathcal{A}$ can be a convex set, describing parameters of a classifier, $o \in \mathcal{O}$ could denote a feature-label pair in which case the loss could be a square loss or logistic loss that measures the performance of each classifier.

Our result leverages a variant of the learning from expert advice problem which assumes an arbitrary subset of K experts might be sleeping at time t and the learner needs to compete against an expert only during its awake duration. The learner chooses a distribution $\boldsymbol{w}_t$ over the awake experts and plays a weighted average over the actions of those awake experts. It then incurs a surrogate-loss called "MixLoss" which is a measure of how good the distribution $\boldsymbol{w}_t$ is. (See Figure 2 for details.) This setting is different from the classical prediction with experts advice problem in two aspects: 1) The adversary is endowed with more power of selecting an awake expert set in addition to the actual outcome o_t at each round. 2) Instead of the loss $\ell(a_t, o_t)$, the learner is incurred a surrogate loss on the distribution chosen by the learner at time t.

Consider the protocol of learning with sleeping experts shown in Fig. 2. Assume an expert pool of size K.

Lemma 2. *(Adamskiy et al., 2016) Regret R_n^j of SAA (Fig. 3) w.r.t. any fixed expert $j \in [K]$ satisfies,*

$$R_n^j := \sum_{t \in [n]} \mathbf{1}\{j \in A_t\} \left(-\log\Big(\sum_{k \in A_t} w_{t,k} e^{-\ell_{t,k}}\Big) - \ell_{t,j} \right) \\ \le \log K,$$

[2] Could be $a_{t,k}$ for some $k \in [K]$ or any other points in $\mathcal{A}$

```
For $t = 1, \dots, n$
  1. Adversary picks a subset $A_t \subset [K]$ of awake experts.
  2. Learner choose a distribution $\boldsymbol{w}_t$ over $A_t$.
  3. Adversary reveals loss of all *awake* experts,
     $\boldsymbol{\ell}_t \in (-\infty, \infty]^{|A_t|}$.
  4. Learner suffers MixLoss:
     $-\log(\sum_{k \in A_t} w_{t,k} e^{-\ell_{t,k}})$.
```

Figure 2: *Interaction protocol with sleeping experts. The expert pool size is K.*

```
Initialize $u_{1,k} = 1/|\mathcal{S}|$ for all $k$ in an index set $\mathcal{S}$ used to
index the expert pool.
For $t = 1, \dots, n$
  1. Adversary reveals $A_t \subseteq \mathcal{S}$.
  2. Play weighted average action wrt distribution:
     $w_{t,k} = \frac{u_{t,k} \mathbf{1}\{k \in A_t\}}{\sum_{j \in A_t} u_{t,j}}$.
  3. Broadcast the weights $w_{t,k}$.
  4. Receive losses $\ell_{t,k}$ for all $k \in A_t$.
  5. Update:
     • $u_{t+1,k} = \frac{u_{t,k} e^{-\ell_{t,k}}}{\sum_{j \in A_t} u_{t,j} e^{-\ell_{t,j}}} \sum_{j \in A_t} u_{t,j}$
       if $k \in A_t$.
     • $u_{t+1,k} = u_{t,k}$ if $k \notin A_t$.
```

Figure 3: *Specialist Aggregation Algorithm (SAA).*

where $\mathbf{1}\{\cdot\}$ is the indicator function, $\ell_{t,k} := \mathcal{L}(a_{t,k}, o_t)$ and $a_{t,k}$ is the action taken by expert k at time t.

Note that $\ell_{t,j} = \mathrm{MixLoss}(\boldsymbol{e}_j)$ where $\mathbf{e}_j$ selects j with probability 1. The regret measures the performance of the learner against any fixed expert in terms of the MixLoss in the subsequence where she is awake.

Definition 3. *$\mathcal{L}(a, x)$ is η exp-concave in a for each x if $\sum_{k=1}^K w_k e^{-\eta \mathcal{L}(a_k, x)} \leq e^{-\eta \mathcal{L}(\sum_{k=1}^K w_k a_k, x)}$, for $w_k \geq 0$ and $\sum_{k=1}^K w_k = 1$.*

A MixLoss regret bound is useful because it implies a regret bound on any exp-concave losses for learners playing the weighted average action $a_t = \sum_{k \in A_t} w_{t,k} a_{t,k}$. To see this, let $\mathcal{L}'(a, o)$ be η exp-concave in its first argument $a \in \mathcal{A}$. By the definition of exp-concavity it follows that if SAA is run with losses $\mathcal{L}(a, o) = \eta \mathcal{L}'(a, o)$, then,

$$\sum_{t \in [n] : j \in A_t} \left(\eta \mathcal{L}' \left(\sum_{k \in A_t} w_{t,k} a_{t,k} \ , \ o_t \right) - \eta \mathcal{L}'(a_{t,j}, o_t) \right) \leq R_n^j,$$

where $a_{t,k}$ is the action taken by expert k at time t.

We refer to Chapter 3 of (Cesa-Bianchi and Lugosi, 2006) and (Adamskiy et al., 2016) for further details on SAA.

3 Main Results

In this section, we present our algorithm and its performance guarantees.

3.1 Algorithm

As noted in Section 1, our goal is to explore the possibility that a Strongly Adaptive online learner can lead to minimax optimal estimation rate. Consequently the algorithm that we present is a fairly standard Strongly Adaptive online learner that can guarantee logarithmic regret in any interval.

Our algorithm ALIGATOR (**A**ggregation of on**LI**ne averaGes using **A** geome**T**ric c**O**ve**R**) defined in Fig.4 can be used to tackle both online and batch estimation problems. The policy is based on learning with sleeping experts where expert pool is defined as follows.

Definition 4. *The expert pool is $\mathcal{E} = \{\mathcal{A}_I : I \in \mathcal{I}|_{[n]}\}$, where $\mathcal{I}|_{[n]}$ is as defined in Section 2.1 and $\mathcal{A}_I$ is an algorithm that perform online averaging in interval I. Let $\mathcal{A}_I(t)$ denote the prediction of the expert $\mathcal{A}_I$ at time t, if $I \in AWAKE(t)$.*

Due to relation (2), we have $|\mathcal{E}| \leq n \log n$. Our policy basically performs SAA over $\mathcal{E}$.

```
ALIGATOR:Inputs - time horizon $n$, learning rate $\eta$
  1. Initialize SAA weights $u_{1,I} = 1/|\mathcal{E}|, \forall I \in \mathcal{I}|_{[n]}$.
  2. For $t$ = 1 to $n$:
     (a) Adversary reveals an arbitrary $x_{i_t} \in \mathcal{X}$.
     (b) Let $A_t = \mathrm{AWAKE}(i_t)$. Pass $A_t$ to SAA.
     (c) Receive $w_{t,I}$ from SAA for each $I \in A_t$.
     (d) Predict $\hat{y}_t = \sum_{I \in A_t} w_{t,I} \mathcal{A}_I(t)$.
     (e) Receive $y_t = f(x_{i_t}) + \epsilon_t$.
     (f) Pass losses $\ell_{t,I} = \eta(y_t - \mathcal{A}_I(t))^2$,
         for each $I \in A_t$ to the SAA.
```

Figure 4: *The* ALIGATOR *algorithm*

The precise definition of $\mathcal{A}_I(t)$ used in our algorithm is

$$\mathcal{A}_I(t) = \begin{cases} \frac{\sum_{s=1}^{t-1} y_s \mathbf{1}\{i_s \in I\}}{\sum_{s=1}^{t-1} \mathbf{1}\{i_s \in I\}} & \text{if } \sum_{s=1}^{t-1} \mathbf{1}\{i_s \in I\} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where i_s is the index of the exogenous variable x_{i_s} in step 2(a) of Fig. 4. This particular choice of experts is motivated by the fact that performing online averages lead to logarithmic static regret under quadratic losses. As shown later, this property when combined with the SAA scheme leads to logarithmic regret in *any* interval of $[n]$.

3.2 Performance Guarantees

Theorem 5. *Consider the online game in Fig. 1. Under Assumption 1, with probability atleast $1-\delta$,* ALIGATOR *forecasts $\hat{y}_t$ obtained by setting $\eta = \frac{1}{8\left(B+\sigma\sqrt{\log(2n/\delta)}\right)^2}$, incurs a cumulative error*

$$\sum_{t=1}^{n}(\hat{y}_t - \theta_t)^2 = \tilde{O}(n^{1/3}C_n^{2/3}),$$

where $\tilde{O}(\cdot)$ hides the dependency of constants B, σ and poly-logarithmic factors of n and δ.

Proof Sketch. We first show that ALIGATOR suffers logarithmic regret against any expert in the pool $\mathcal{E}$ during its awake period. Then we exhibit a particular partition of the underlying TV bounded function such that number of chunks in the partition is $O(n^{1/3}C_n^{2/3})$ (Lemma 3 in Appendix B). Following this, we cover each chunk with atmost $\log n$ experts and show that each expert in the cover suffers a $\tilde{O}(1)$ estimation error. The Theorem then follows by summing the estimation error across all chunks of the partition. In summary, the delicate interplay between Strongly Adaptive regret bounds and properties of the partition we exhibit leads to the adaptively minimax optimal estimation rate for ALIGATOR. □

Remark 6. *We note that under the above setting,* ALIGATOR *is minimax optimal in n and C_n, and adaptive to unknown C_n.*

Remark 7. *If the noise level σ is unknown, it can be robustly estimated from the wavelet coefficients of the observed data by a Median Absolute Deviation estimator (Johnstone, 2017). This is facilitated by the sparsity of wavelet coefficients of BV functions .*

Remark 8. *In the offline problem where we have access to all observations ahead of time, the choice of $\eta = 1/(8\hat{\nu}^2)$ where $\hat{\nu} = \max\{|y_1|, \ldots, |y_n|\}$ results in the same near optimal rate for R_n as in Theorem 5. This is due to the fact that $B + \sigma\sqrt{\log(2n/\delta)}$ is nothing but a high probability bound on each $|y_t|$. Hence we don't require the prior knowledge of B and σ for the offline problem.*

Remark 9. *The authors of (Donoho et al., 1998) use the error metric given by the L2 function norm in a compact interval $[0,1]$ defined as $\int_0^1 \left(\hat{f}(x) - f(x)\right)^2 dx$ in an offline setting, where $\hat{f}(x)$ is the estimated function. A common observation model for non-parametric regression considers $x_{i_t} = t/n$ (Tibshirani, 2014). When $x_{i_t} = t/n$,* ALIGATOR *guarantees that the empirical norm $\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - f(t/n))^2$ decays at the rate of $\tilde{O}\left(n^{-2/3}C_n^{2/3}\right)$. For the TV class, it can be shown that the empirical norm and the function norm are close enough such that the estimation rates do not change (see Section 15.5 of (Johnstone, 2017)).*

Remark 10. *Note that conditioned on the past observations, the prediction of* ALIGATOR *is deterministic in each round. So in the online setting, we can compete with an adversary who chooses the underlying ground truth in an adaptive manner based on the learner's past moves. With such an adaptive adversary, it becomes important to reveal the set of covariates $\mathcal{X}$ ahead of time. Otherwise there exists a strategy for the adversary to choose the covariates x_{i_t} that can enforce a linear growth in the cumulative squared error. We refer the readers to (Kotłowski et al., 2016) for more details about such adversarial strategy.*

Proposition 11. *The overall run-time of* ALIGATOR *is $O(n \log n)$.*

Proof. On each round $|\text{AWAKE}(t)|$ is $O(\log n)$ by (2). So we only need to aggregate and update the weights of $O(\log n)$ experts per round which can be done in $O(\log n)$ time. □

4 Extensions

Motivated from a practical perspective, we discus two direct extensions to ALIGATOR below. These extensions highlight the versatility of ALIGATOR in adapting to each application.

Hedged ALIGATOR. In our theoretical results, we found that choosing learning rate η conservatively according to Theorem 5 or Remark 8 ensures the minimax rates. In practice, however, one could use larger learning rates to adapt to the structure of every input sequence.

We propose to use a hedged ALIGATOR scheme that aggregates the predictions of ALIGATOR instantiated with different learning rates. In particular, we run different instances of ALIGATOR in parallel where an instance corresponds to a learning rate in the exponential grid $[\eta, 2\eta, \ldots, \max\{\eta, \log_2 n\}]$ which has a size of $O\left(\log\left((B^2+\sigma^2)\log n\right)\right)$ (recall that η is chosen as per Theorem 5 or Remark 8). Then we aggregate each of these instances by the Exponential Weighted Averages (EWA) algorithm (Cesa-Bianchi and Lugosi, 2006). The learning rate of this outer EWA layer is set according to the theoretical value. By exp-concavity of squared error losses, this strategy helps to match the performance of the best ALIGATOR instance. Since the theoretical choice of learning rate is included in the exponential grid, the strategy can also guarantee optimal minimax rate. We emphasize that Hedged ALIGATOR is adaptive to C_n and requires no hyper-parameter tuning.

ALIGATOR with polynomial regression experts. This extension is motivated by the problem of identifying trends in time series. Though in Section 3.1 we use online averaging as experts, in practice one can consider using other algorithms. For example, if the trends in a time series are piecewise-linear, then experts based on online averaging can lead to poor practical performance because the TV budget C_n of piecewise linear signals can be very large. To alleviate this, in this extension, we propose to use Online Polynomial Regression as experts where a polynomial of a fixed degree d is fitted to the data with time points as its exogenous variables. This is similar to the idea adopted in (Baby and Wang, 2020) where they construct a policy that performs restarted online polynomial regression where the restart schedule is adaptively chosen via wavelet based methods. They show that such a scheme can guarantee estimation rates that grow with (a scaled) L1 norm of higher order differences of the underlying trend which can be much smaller than its TV budget C_n. This extension can be viewed as a variant to the scheme in (Baby and Wang, 2020) where the "hard" restarts are replaced by "soft restarts" via maintaining distributions over the sleeping experts.

5 Experimental Results

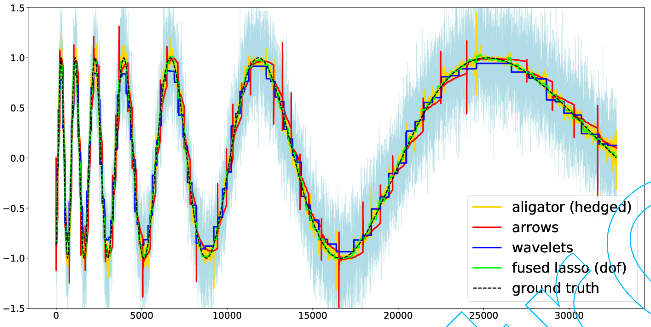


Figure 5: *Fitted signals for Doppler function with noise level σ= 0.25*

For empirical evaluation, we consider online and offline vesrions of the problems separately.

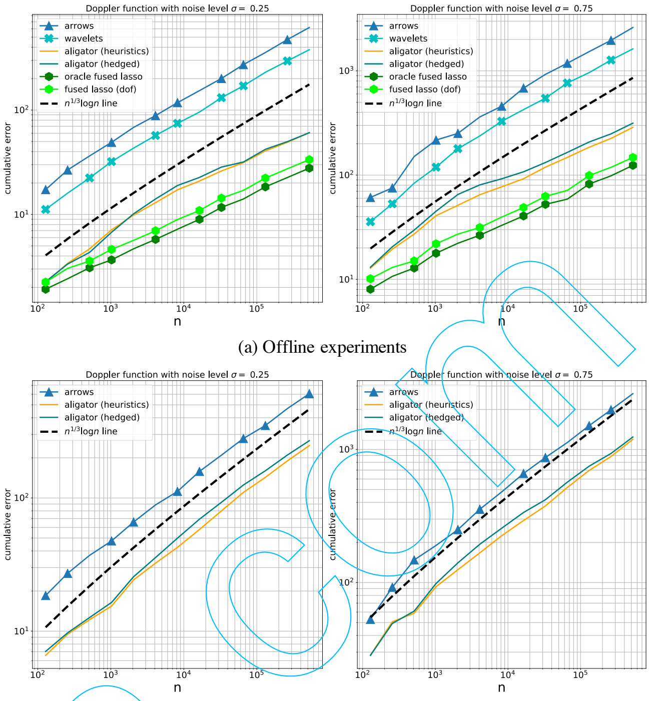
Description of policies. We begin by a description of each algorithm whose error curve is plotted in the figures.

ALIGATOR *(hedged):* This is the extension described in Section 4

ALIGATOR *(heuristics):* For this hueristics strategy, we divide the loss of each expert by $2(\sigma^2 + \sigma^2/m)$ where m is the number of samples whose running average is compued by the expert. This loss is proportional to the notion of (squared) z-score used in hypothesis testing. Intuitively, lower (squared) z-score corresponds to better experts. The multiplier 2 in the previous expression is found to provide good performnace across all signals we consider.

arrows: This is the the policy presented in (Baby and Wang, 2019), which runs online averaging with an adaptive restarting rule based on wavelet denoising results.

wavelets: This is the universal soft thresholding estimator from (Donoho et al., 1998) based on Haar wavelets which is



(a) Offline experiments

(b) Online experiments

Figure 6: *Cumulative squared error rate of various algorithms on offline setting and online setting.* ALIGATOR *achieves the optimal $\tilde{O}(n^{1/3})$ rate while performing better than wavelet based methods. In particular, in the offline setting, it achieves a performance closer to that of dof based fused lasso while only incurring a cheap $\tilde{O}(n)$ run-time overhead.*

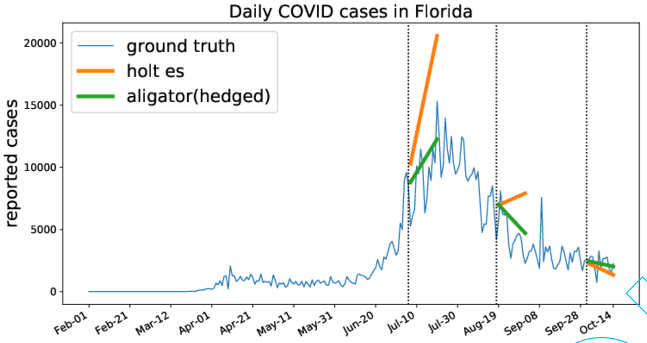


Figure 7: *A demo on forecasting COVID cases based on real world data. We display the two weeks forecasts of hedged* ALIGATOR *and Holt ES, starting from the time points identified by the dotted lines. Both the algorithms are trained on a 2 month data prior to each dotted line. We see that hedged* ALIGATOR *detects changes in trends more quickly than Holt ES. Further, hedged* ALIGATOR *attains a 20% reduction in the average RMSE from that of Holt ES (see Section 5).*

known to be minimax optimal for estimating BV functions.

oracle fused lasso: This estimator is obtained by solving (1) whose hyper-parameter is tuned by assuming access to an oracle that can compute the mean squared error wrt actual ground truth. The exact ranges used in the hyper-parameter grid search is described in Appendix C. Note that the oracle fused lasso estimator is purely hypothetical due to absence of such oracles described before in reality and is ultimately impractical. It is used here to facilitate meaningful comparisons.

fused lasso (dof): In this experiment, we maintain a list of λ for the fused lasso problem (Eq. (1)). Then we compute the Stein's Unbiased Risk estimator for the expected squared error incurred by each λ by estimating its degree of freedom (dof) (Tibshirani and Taylor, 2012) and select the λ with minimum estimated error.

Experiments on synthetic data. For the ground truth signal, we use the Doppler function of (Donoho and Johnstone, 1994a) whose waveform is depicted in Fig. 5. The observed data are generated by adding iid noise to the ground truth. For offline setting, we have access to all observations ahead of time. So we run Arrows and both versions of ALIGATOR two times on the same data, once in isotonic order (i.e $i_t = t$ in Fig. 1) and other in reverse isotonic order and average the predictions to get estimates of the ground truth. For online setting such a forward-backward averaging is not performed. This process of generating the noisy data and computing estimates are repeated for 5 trials and the average cumulative error is plotted. As we can see from Fig.6 (a), ALIGATOR versions attains the $\tilde{O}(n^{1/3})$ rate and incurs much lower error than wavelet smoothing. Further, performance of hedged and heuristics versions of ALIGATOR is in the vicinity to that of the hypothetical fusedlasso estimator while the policies arrows and wavelets violate this property by a large margin. Even though the dof based fused lasso comes very close to the oracle counterpart, we emphasize that this strategy is not known to provide theoretical guarantees for its rate and requires heavy computational bottleneck since it requires to solve the fused lasso (Eq. 1) for many different values of λ.

For the online version of the problem, we consider the policy Arrows as the benchmark. This policy has been established to be minimax optimal for online forecasting of TV bounded sequences in (Baby and Wang, 2019). We see from Fig.6 (b) that all the policies attains an $\tilde{O}(n^{1/3})$ rate while ALIGATOR variants enjoy lower cumulative errors.

Experiments on real data. Next we consider the task of forecasting COVID cases using the extension of Aligator with polynomial regression experts as in Section 4. The data are obtained from the CDC website (cdc).

We address a very relevant problem as follows: Given access to the historical data, forecast the evolution of COVID cases for the next 2 weeks. We compare the performance of hedged ALIGATOR and Holt Exponential Smoothing (Holt ES), on this problem, where the later is a common algorithm used in Time Series forecasting to detect underlying trends. For ALIGATOR, we use Online Linear Regression as experts where a polynomial of degree one is fitted to the data with time points as its exogenous variables. For each time point t in [Apr 20, Sep 27], we train both hedged ALIGATOR and Holt ES on a training window of past 2 months. Then we calculate a 2 week forecast for both algorithms. For ALIGATOR this is achieved by linearly extrapolating the predictions of experts awake at time t and aggregating them. Following this, we compute the Root Mean Squared Error (RMSE) in the interval $[t, t+14)$ for both algorithms. These RMSE are then averaged across all t in [Apr 20, Sep 27].

We choose data from the state of Florida, USA, as an illustrative example. We obtained an average RMSE of 1330.12 for hedged ALIGATOR and 1671.77 for Holt ES. Thus hedged ALIGATOR attains a 20% reduction in forecast error from that of Holt ES. A qualitative comparison of the forecasts is illustrated in Fig. 7. As we can see, the time series is non-stationary and has a varying degree of smoothness. ALIGATOR is able to adapt to the local changes quickly, while Holt ES fails to do so despite having a more sophisticated training phase. Similar experimental results for some of the other states are reported in Appendix C.

The training step of hedged ALIGATOR involves learning the weights of all experts by an online interaction protocol as shown in Fig. 1 with $i_t = t$. It is remarkable that *no* hyper-parameter tuning is required by ALIGATOR for its training phase. The slowest learning rate to be used in the grid for hedged ALIGATOR is computed as follows. First we calculate the maximum loss incurred by each expert for a one step ahead forecast in its awake duration. Then we take the maximum of this quantity across all experts in the pool. Let this quantity be β. The slowest learning rate in the grid is then set as $1/(2\beta)$. The learning rate of the outer layer of EWA is also set the same. This is justifiable because the quantity $4\left(B + \sigma\sqrt{\log(2n/\delta)}\right)$ in the denominator of the learning rate in Theorem 5 is a high probability bound on the loss incurred by any expert for a one step ahead forecast.

We defer further experimental results to Appendix C.

An important caveat for practitioners. Though ALIGATOR is able to detect non-stationary trends in the COVID data efficiently, we do *not* advocate using ALIGATOR *as is* for pandemic forecasting, which is a substantially more complex problem that requires input from domain experts.

However, ALIGATOR could have a role in this problem, and other online forecasting tasks. Estimating (and removing) trend is an important first step in many time series methods (e.g., Box-Jenkins method). Most trend estimation methods only apply to offline problems (e.g., Hodrick-Prescott filter or L1 Trend Filter) (Kim et al., 2009), while Holt ES is a common method used for online trend estimation. For instance, Holt ES is being used as a subroutine for trend estimation in a state-of-the-art forecasting method (Jin et al., 2021) for COVID cases that CDC is currently using. We expect that using ALIGATOR instead in such models that use Holt ES will lead to more accurate forecasting, but that is beyond the scope of this paper.

6 Concluding Discussion

In this work, we presented a novel reduction from estimating BV functions to Strongly Adaptive online learning. The reduction gives rise to a new algorithm ALIGATOR that attains the near minimax optimal rate of $\tilde{O}(n^{1/3}C_n^{2/3})$ in $O(n \log n)$ run-time. The results form a parallel to wavelet smoothing in terms of optimal adaptivity to unknown variational budget C_n. However, our algorithm is more versatile than wavelets in terms of its configurability and practical performance. Further, for offline estimation, ALIGATOR variants achieves a performance closer (than wavelets) to an oracle fused lasso while incurring only an $\tilde{O}(n)$ run-time with no hyper parameter tuning. This is in contrast to degree of freedom based approaches of tuning the fused lasso hyper parameter that requires significantly more computational overhead and is not known to provide guarantees on its rate.

Acknowledgment

The research is partially supported by NSF Award #2029626 and a start-up grant from UCSB CS department. XZ was supported by UCSB Chancellor's Fellowship.

References

Centers for disease control and prevention (cdc). https://www.cdc.gov/.

Dmitry Adamskiy, Wouter M. Koolen, Alexey Chernov, and Vladimir Vovk. A closer look at adaptive regret. *Journal of Machine Learning Research*, 2016.

Kazunori Akiyama, Katherine Bouman, and David Woody. First m87 event horizon telescope results. i. the shadow of the supermassive black hole. *Astrophysical Journal Letters*, 2019.

Dheeraj Baby and Yu-Xiang Wang. Online forecasting of total-variation-bounded sequences. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Dheeraj Baby and Yu-Xiang Wang. Adaptive online estimation of piecewise polynomial trends. *Neural Information Processing Systems (NeurIPS)*, 2020.

Alvaro Barbero and Suvrit Sra. Fast Newton-type methods for total variation regularization. In *International Conference on Machine Learning (ICML-11)*, volume 28, pages 313–320, 2011.

Lucien Birge and Pascal Massart. Gaussian model selection. *Journal of the European Mathematical Society*, 3(3):203–268, 2001.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

Antonin Chambolle and Pierre-Louis Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.

Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411, 2015.

David Donoho and Iain Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994a.

David Donoho and Iain Johnstone. Minimax risk over ℓ_p-balls for ℓ_q-error. *Probability Theory and Related Fields*, 99(2):277–303, 1994b.

David Donoho, Richard Liu, and Brenda MacGibbon. Minimax risk over hyperrectangles, and implications. *Annals of Statistics*, 18(3):1416–1437, 1990.

David L Donoho, Iain M Johnstone, et al. Minimax estimation via wavelet shrinkage. *The annals of Statistics*, 26 (3):879–921, 1998.

Adityanand Guntuboyina, Donovan Lieu, Sabyasachi Chatterjee, and Bodhisattva Sen. Adaptive risk bounds in univariate total variation denoising and trend filtering. 2017.

Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007.

Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

Xiaoyong Jin, Yu-Xiang Wang, and Xifeng Yan. Inter-series attention model for covid-19 forecasting. *SIAM International Conference on Data Mining (to appear)*, 2021.

Nicholas Johnson. A dynamic programming algorithm for the fused lasso and L_0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, 2013.

Iain M. Johnstone. *Gaussian estimation: Sequence and wavelet models*. 2017.

Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2): 339–360, 2009.

Wojciech Kotłowski, Wouter M. Koolen, and Alan Malek. Online isotonic regression. In *Annual Conference on Learning Theory (COLT-16)*, volume 49, pages 1165–1189. PMLR, 2016.

Enno Mammen. Nonparametric regression under qualitative smoothness assumptions. *Annals of Statistics*, 19(2):741—759, 1991.

Enno Mammen and Sara van de Geer. Locally apadtive regression splines. *Annals of Statistics*, 25(1):387–413, 1997.

Leonid Rudin, Stanley Osher, and Emad Faterni. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

Veeranjaneyulu Sadhanala, Yu-Xiang Wang, and Ryan J Tibshirani. Graph sparsification approaches for laplacian smoothing. In *AISTATS'16*, pages 1250–1259, 2016.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.

Ryan J Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics*, 42(1): 285–323, 2014.

Ryan J. Tibshirani and Jonathan Taylor. Degrees of freedom in lasso problems. *Annals of Statistics*, 40(2):1198–1232, 2012.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008.

Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan J Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105):1–41, 2016.