arXiv:1111.1136v2 [cs.LG] 14 Nov 2011

# Universal MMSE Filtering With Logarithmic Adaptive Regret

Dan Garber
Technion - Israel Inst. of Tech.
dangar@cs.technion.ac.il

Elad Hazan
Technion - Israel Inst. of Tech.
ehazan@ie.technion.ac.il

November 16, 2011

**Abstract**

We consider the problem of online estimation of a real-valued signal corrupted by oblivious zero-mean noise using linear estimators. The estimator is required to iteratively predict the underlying signal based on the current and several last noisy observations, and its performance is measured by the mean-square-error. We describe and analyze an algorithm for this task which:

1. Achieves logarithmic adaptive regret against the best linear filter in hindsight. This bound is assyptotically tight, and resolves the question of Moon and Weissman [1].
2. Runs in linear time in terms of the number of filter coefficients. Previous constructions required at least quadratic time.

## 1 Introduction

We consider the problem of filtering: designing algorithms for the causal estimation of a real valued signal from noisy observations. The filtering algorithm observes at each iteration a noisy signal component, and is required to estimate the corresponding underlying signal component based on the current and past noisy observations alone.

We consider finite fixed-length linear filters that combine the current and several last noisy observations for prediction of the current underlying signal component. Performance is measured by the mean square error over the entire signal. Following the setting in [1], we assume that the underlying signal is an arbitrary bounded signal, possibly even adversarial, and that it is corrupted by an additive zero-mean, time-independent, bounded noise with known constant variance [1].

The approach taken in this paper is to construct a *universal* filter - i.e. an adaptive filter whose performance we compare to an optimal offline filter with full knowledge of the signal and noise. The metric of performance is thus regret - or the difference between the total mean squared error incurred by our adaptive filter, and the total mean square error of the offline benchmark filter.

The question of competing with a fixed offline filter was successfully tackled in [1]. In this paper we consider a more challenging task: competing with the best offline changing filter, where restrictions are placed on how often this optimal offline filter is allowed to change. A more stringent metric of performance what fully captures this notion of competing with an adaptive offline benchmark is called *adaptive regret*: it is the maximum regret incurred by the algorithm on any subinterval.

We present and analyze simple, efficient and intuitive algorithms that attain logarithmic adaptive regret. This bound is tight, and resolves a question posed by Moon and Weissman in [1]. Along the way, we introduce a simple universal algorithm for filtering, improving the previously known best running time from quadratic in the number of filter coefficients to linear.

## 1.1 Related Work

There has been much work on the problem of estimating a real-valued signal from noisy observations with respect to the MMSE loss over the years. Classical results assume a model in which the underlying signal is stochastic with some known parameters, i.e. the first and second moments, or require the signal to be stationary, such as the classical work of [2]. The special case of linear MMSE filters has received special attention due to its simplicity [3]. For more recent results on MMSE estimation see [4, 5, 6, 7].

In this work we follow the non-stochastic setting of [1]: no generating model is assumed for the underlying signal and stochastic assumptions are made on the added noise (that it is zero-mean, time-independent with known fixed variance). In this setting, while considering finite linear filters, [1] presented an online algorithm that achieves logarithmic expected regret with respect to the entire signal. The computational complexity of their algorithm is proportional to a quadratic in the linear filter size.

Henceforth we build on recent results from the emerging online learning framework called online convex optimization [8, 9]. For our adaptive regret algorithm, we use tools from the framework presented in [10] to derive an algorithm that

[1] The justification of [1] for assuming that the variance is a known constant is that this variance could be learned by sending a training sequence in the beginning of transmission.

achieves logarithmic expected regret on any interval of the signal.

# 2 Preliminaries

## 2.1 Online convex optimization

In the setting of online convex optimization (OCO) an online algorithm $\mathcal{A}$ is iteritevly required to make a prediction by choosing a point $x_t$ in some convex set $\mathcal{K}$. The algorithm then incurs a loss $l_t(x_t)$, where $l_t(x) : \mathcal{K} \to \mathbb{R}$ is a convex function. The emphasis in this model is that on iteration $t$, $\mathcal{A}$ has only knowledge of the loss functions in previous iterations $l_1(x), ..., l_{t-1}(x)$ and thus $l_t(x)$ may be chosen arbitrarily and even adversely. The standard goal in this setting is to minimize the difference between the overall loss of $\mathcal{A}$ and that of the best fixed point $x^* \in \mathcal{K}$ in hindsight. This difference is called regret and it is formally given by,

$$R_T(\mathcal{A}) = \sum_{t=1}^{T} l_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} l_t(x)$$

A stronger measure of performance requires the algorithm to have little regret on any interval $I = [r, s] \subseteq [T]$ with respect to the best fixed point $x_I^* \in \mathcal{K}$ in hindsight in this interval. This measure is call adaptive regret and it is given by ,

$$AR_T(\mathcal{A}) = \sup_{I=[r,s]\subset[T]} \{\sum_{t=r}^{s} l_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=r}^{s} (l_t(x)\}$$

## 2.2 Problem Setting

Let $x_t$ be a real-valued, possibly adversarial, signal bounded in the range $[-B_X ... B_X]$. The signal $x_t$ is corrupted by an additive zero-mean time independent noise $n_t$ bounded in the range $[-B_N ... B_N]$ with known time-invariant variance $\sigma^2$. An estimator observes on time $t$ the noisy signal $y_t = x_t + n_t$, and is required to predict $x_t$ by taking a linear combination of the observations $y_t, y_{t-1}, ..., y_{t-d+1}$ where $d$ is the order of the filter. That is, the estimator chooses on time $t$ a filter $w_t \in \mathbb{R}^d$ and predicts according to $w_t^\top Y_t$ where $Y_t \in \mathbb{R}^d$ and $Y_t(i) = y_{t-i+1}$, $1 \leq i \leq d$. The loss of the estimator after $T$ iterations is given by the mean-square-error $\frac{1}{T} \sum_{t=1}^{T} (x_t - w_t^\top Y_t)^2$.
In case $x_t$ is observable to the online algorithm, minimizing the regret and the adaptive regret is fairly easy using the framework of OCO with the loss functions $l_t(w_t) = (x_t - w_t^\top Y_t)^2$. However in our case, the algorithm only observes the noisy signal $y_t$ and thus online convex optimization algorithms could be directly

used. Denoting $\hat{l}_t(w) = (y_t - w^\top Y_t)^2 + 2w^\top c$ where $c \in \mathbb{R}^d$, $c = (\sigma^2, 0..., 0)$, it was pointed out in [1] that if $w_t$ depends only on the observations $y_1, ..., y_{t-1}$, then for any $w \in \mathbb{R}^d$ it holds that,

$$\mathbb{E}\left[\sum_{t=1}^{T} \hat{l}_t(w_t) - \sum_{t=1}^{T} \hat{l}_t(w)\right] = \mathbb{E}\left[\sum_{t=1}^{T} l_t(w_t) - \sum_{t=1}^{T} l_t(w)\right] \tag{1}$$

Thus by using OCO algorithms with the estimated loss functions $\hat{l}_t(w)$ we may minimize the expected regret with respect to the actual losses $l_t(w)$. Thus a simple algorithm such as [8] immediately gives a $O(\sqrt{T})$ bound on the expected regret as well as on the expected adaptive regret with respect to the true losses $l_t(w)$, as long as we limit the choice of the filter to a euclidean ball of constant radius.

### 2.3 Using Strong-Convexity and Exp-Concavity

Given a function $f(x) : \mathcal{K} \rightarrow \mathbb{R}$ we denote by $\nabla f(x)$ the gradient vector of $f$ at point $x$ and by $\nabla^2 f(x)$ the matrix of second derivatives, also known as the Hessian, of $f$ at point $x$. $f(x)$ is convex at point $x$ if and only if $\nabla^2 f(x) \succeq 0$, that is its Hessian is positive semidefinite at $x$.
We say that $f$ is *$H$-strongly-convex*, for some $H > 0$, if for all $x \in \mathcal{K}$ it holds that $\nabla^2 f(x) \succeq H\mathbf{I}$, where **I** is the identity matrix of proper dimension. That is all the eigenvalues of $\nabla^2 f(x)$ are lower bounded by $H$ for all $x \in \mathcal{K}$.
We say that $f$ is *$\alpha$-exp-concave*, for some $\alpha > 0$, if the function $\exp(-\alpha f(x))$ is a concave function of $x \in \mathcal{K}$. It is easy to show that given a function $f$ such that $f \succeq H\mathbf{I}$ and $\max_{x\in\mathcal{K}} \|\nabla f(x)\|_2 \leq G$ it holds that $f$ is $\frac{H}{G^2}$-exp-concave.
In case all loss functions are $H$-strongly-convex or $\alpha$-exp-concave, there exists algorithms that achieve logarithmic regret and adaptive regret [9, 10].
In our case, the Hessian of the loss function $\hat{l}_t(w)$ is given by the random matrix $\nabla^2 \hat{l}_t(w) = 2Y_tY_t^\top$ which is positive semidefinite and it holds that

$$\mathbb{E}\left[Y_tY_t^\top\right] = \mathbb{E}\left[X_tX_t^\top + N_tX_t^\top + X_tN_t^\top + N_tN_t^\top\right] = X_tX_t^\top + \sigma^2\mathbf{I} \succeq \sigma^2\mathbf{I} \tag{2}$$

Nevertheless, in worst case, $\hat{l}_t(w)$ need not be strongly-convex or exp-concave and thus algorithms such as [9, 10] could not be directly used in order to get logarithmic expected regret and adaptive regret.

## 3 A Simple Gradient Decent Filter

In this section we describe how the problem of the loss functions $\hat{l}_t$ not necessarily being strongly-convex or exp-concave could be overcome and introduce a simple gradient decent algorithm based on [9] that achieves $O(\log T)$ expected regret. For time $t$ and filter $w \in \mathbb{R}^d$ we define the following loss functions.

$$L_t^k(w) = \sum_{\tau=t-k+1}^{t} \hat{l}_t(w) + (w - w_t)^\top \left( (k-d+1)\sigma^2 \mathbf{I} - \sum_{\tau=t-k+d}^{t} Y_t Y_t^\top \right) (w - w_t) \quad (3)$$

where $w_t$ is the filter that was used by the algorithm for prediction in time $t$ and $k \in \mathbb{N}^+$ is a parameter.

Our Gradient Decent filtering algorithm is given below.

**Algorithm 1** GDFilter

```
Input: $k \in \mathbb{N}^+$, $H \in \mathbb{R}^+$, $R \in \mathbb{R}^+$.
Let $w_1 = \mathbf{0}_d$
for $c = 1...$ do
    for $t = (c-1)k + 1...ck$ do
        predict: $x_t = w_c^\top Y_t$.
    end for
    $\eta_c \leftarrow \frac{1}{Hc}$
    $\tilde{w}_{c+1} \leftarrow w_c - \eta_c \nabla L_c^k(w_c)$.
    if $\|\tilde{w}_{c+1}\| > R$ then
        $w_{c+1} \leftarrow \tilde{w}_{c+1} \cdot \frac{R}{\|\tilde{w}_{c+1}\|}$.
    else
        $w_{c+1} \leftarrow \tilde{w}_{c+1}$.
    end if
end for
```

We have the following theorem and corollary.

**Theorem 1.** *Let $w_t$ be the filter used by algorithm 1 for prediction in time $t$. Let $k$ = $2d$ and $H = d\sigma^2$. Algorithm 1 achieves the following regret bound,*

$$\mathbb{E}\left[\sum_{t=1}^{T} l_t(w_t)\right] - \min_{w \in \mathbb{R}^d, \|w\| \leq R} \mathbb{E}\left[\sum_{t=1}^{T} l_t(w)\right] = O\left(\frac{d^3 R^2 (B_X + B_N)^4}{\sigma^2} \log T\right)$$

**Corollary 1.** *Let $w_t$ be the filter used by algorithm 1 for prediction in time $t$. Let $k = 2d$, $H = d\sigma^2$ and let $R = \frac{\sqrt{d} B_X^2}{\sigma^2}$. It holds that,*

$$\mathbb{E}\left[\sum_{t=1}^{T} l_t(w_t)\right] - \min_{w \in \mathbb{R}^d} \mathbb{E}\left[\sum_{t=1}^{T} l_t(w)\right] = O\left(\frac{d^4 B_X^4 (B_X + B_N)^4}{\sigma^6} \log T\right)$$

Basically the new loss function (3) sums several consecutive losses and adds a regularization expression. We show that since the regularization expression depends on the actual choices of the filtering algorithm, achieving low regret with respect to $L_t^k(w)$ implies low regret with respect to the losses $l_t(w)$. Moreover, as we will show, the combination of summing several losses and adding regularization, insures that $L_t^k(w)$ is always strongly-convex for a proper choice of $k$, and thus we can use the algorithms in [9, 10] to get logarithmic regret.
It holds that,

$$\begin{aligned}
\nabla^2 L_t^k(w) &= \sum_{\tau=t-k+1}^{t} \nabla^2 \hat{l}_t(w) + 2\left((k-d+1)\sigma^2\mathbf{I} - \sum_{\tau=t-k+d}^{t} Y_t Y_t^\top\right) \\
&= 2\sum_{\tau=t-k+1}^{t} Y_t Y_t^\top + 2(k-d+1)\sigma^2\mathbf{I} - 2\sum_{\tau=t-k+d}^{t} Y_t Y_t^\top \\
&\succeq 2(k-d+1)\sigma^2\mathbf{I} \qquad (4)
\end{aligned}$$

Thus for $k \geq d$, $L_t^k(w)$ is always $2(k-d+1)\sigma^2$-strongly-convex and $2(k-d+1)\sigma^2/G^2$-exp-concave where $G = \max_{w,t} \|\nabla L_t^k(w)\|$.
We thus use the gradient decent algorithm in [9] by partitioning the iterations into disjoint blocks of length $k$ each, and our algorithm updates its filter every $d$ iterations according to the loss function $L_t^k(w)$ for $t = ck$, $c \in \mathbb{Z}$ and predicts using the same filter on all iterations in the same block. The value of $k$ is assumed to be a constant independent of $T$.
Abusing notation, we switch between $L_c^k(w)$ and $L_{ck}^k(w)$ interchangeably where we use $L_c^k(w)$ to refer to the loss on block number $c$ of length $k$.
The following Lemma plays a key part in our analysis.

**Lemma 1.** *Let $\mathcal{A}$ be a filtering algorithm that updates its filter every $k$ iterations. Denote by $w_t$ the filter used for prediction on iteration $t$ and denote by $w_c$ the filter used to predict on the entire block $c$, that is on iterations $((c-1)\cdot k+1)...c \cdot k$. It holds that*

$$\mathbb{E}\left[\sum_{t=1}^{T} l_t(w_t) - \sum_{t=1}^{T} l_t(w)\right] \leq \mathbb{E}\left[\sum_{c=1}^{T/k} L_{ck}^k(w_c) - \sum_{c=1}^{T/k} L_{ck}^k(w)\right]$$

*Proof.* First we assume w.l.o.g. that $T = b \cdot k$ for some $b \in \mathbb{N}^+$. Otherwise it holds that $T = b \cdot k + a$ where $0 < a < k$ and thus the regret on the additional $a$ iterations is a constant independent of $T$ and we can ignore it in the regret bound.
We now have,

$$\begin{aligned}
& \sum_{c=1}^{T/k} L_{ck}^{k}(w_c) - \sum_{c=1}^{T/k} L_{ck}^{k}(w) \qquad (5) \\
= \;& \sum_{c=1}^{T/k} \left( \sum_{t=(c-1)k+1}^{ck} \hat{l}_t(w_c) + (w_c - w_c)^\top \left( (k-d+1)\sigma^2 \mathbf{I} - \sum_{\tau=ck-k+d}^{ck} Y_\tau Y_\tau^\top \right) (w_c - w_c) \right) \\
- \;& \sum_{c=1}^{T/k} \left( \sum_{t=(c-1)k+1}^{ck} \hat{l}_t(w) + (w - w_c)^\top \left( (k-d+1)\sigma^2 \mathbf{I} - \sum_{\tau=ck-k+d}^{ck} Y_\tau Y_\tau^\top \right) (w - w_c) \right) \\
= \;& \sum_{t=1}^{T} \left( \hat{l}_t(w_t) - \hat{l}_t(w) \right) - \sum_{c=1}^{T/k} (w - w_c)^\top \left( (k-d+1)\sigma^2 \mathbf{I} - \sum_{\tau=(c-1)k+1}^{ck} Y_\tau Y_\tau^\top \right) (w - w_c)
\end{aligned}$$

Since $\mathcal{A}$ updates its filter every $k$ iterations, we have that $w_{ck}$ depends only on the random variables $n_1, ..., n_{(c-1)k}$. Thus using (2) we have for all $c$ we that,

$$\begin{aligned}
& \mathbb{E}\left[ (w - w_c)^\top \left( (k-d+1)\sigma^2 \mathbf{I} - \sum_{\tau=(c-1)k+1}^{ck} Y_\tau Y_\tau^\top \right) (w - w_c) \right] \\
= \;& (k-d+1)\sigma^2 \mathbb{E}[\|w - w_c\|^2] - \mathbb{E}\left[ \sum_{\tau=(c-1)k+1}^{ck} Y_\tau Y_\tau^\top \right] \circ \mathbb{E}\left[ (w - w_c)(w - w_c)^\top \right] \\
= \;& (k-d+1)\sigma^2 \mathbb{E}[\|w - w_c\|^2] \\
- \;& \left( \sum_{\tau=(c-1)k+1}^{ck} X_\tau X_\tau^\top + (k-d+1)\sigma^2 \mathbf{I} \right) \circ \mathbb{E}\left[ (w - w_c)(w - w_c)^\top \right] \\
= \;& - \sum_{\tau=(c-1)k+1}^{ck} X_\tau X_\tau^\top \circ \mathbb{E}\left[ (w - w_c)(w - w_c)^\top \right] \leq 0
\end{aligned}$$

Overall by taking expectation over (5) we get

$$\mathbb{E}\left[ \sum_{c=1}^{T/k} L_{ck}^{k}(w_c) - \sum_{c=1}^{T/k} L_{ck}^{k}(w) \right] \geq \mathbb{E}\left[ \sum_{t=1}^{T} \hat{l}_t(w_t) - \hat{l}_t(w) \right]$$

The lemma now follows from (1). □

According to Lemma 1 we can reduce our discussion to algorithms that predict in disjoint blocks of length $k$ and achieve low regret with respect to the loss

function $L_c^k(w)$
In order to derive precise regret bounds we give a bound on $G = \max_{w,t} \|\nabla L_t^k(w)\|$.

$$\nabla L_t^k(w) = 2 \sum_{\tau=t-k+1}^{t} Y_t(y_t - w_t^\top Y_t) + 2 \left( (k-d+1)\sigma^2 \mathbf{I} - \sum_{\tau=t-k+d}^{t} Y_\tau Y_\tau^\top \right) (w - w_t)$$

Thus by simple algebra we have,

$$\begin{aligned} G^2 &= O\left(k^2 d(B_X + B_N)^2 R^2 d(B_X + BN)^2 + k^2 d^2 (B_X + B_N)^4 R^2\right) \\ &= O\left(k^2 d^2 R^2 (B_X + B_N)^4\right) \end{aligned}$$

Where $R$ is a bound on the magnitude of the filter. That is we consider only filters $w \in \mathbb{R}^d$ such that $\|w\|_2 \leq R$. $R$ needs to be bounded since the regret of online convex optimization algorithms grows with $G$.
As pointed out in [1], for

$$w^* = \arg\min_{w \in \mathbb{R}^d} \mathbb{E}\left[(1/T) \sum_{t=1}^{T} \left(x_t - w^\top Y_t\right)^2\right]$$

It holds that $\|w^*\| \leq \frac{\sqrt{d} B_X^2}{\sigma^2}$.
We denote by $G(k, R)$ an upper bound on $\max_{w,t} \|\nabla L_t^k(w)\|$ parametrized by $k, R$.
For the complete proof of the theorem and corollary the reader is referred to the appendix.

# 4 An Adaptive Algorithm

In this section we present an algorithm that is based on the framework from [10] and achieves logarithmic expected regret on any interval $I = [r, s] \subseteq [T]$. Our algorithm is given below. We have the following theorem and corollary.

**Theorem 2.** *Let $w_t$ be the filter used by algorithm 2 for prediction in time $t$. Let $k = 2d$ and let $\alpha = \frac{d\sigma^2}{G(2d,R)^2}$. For all $I = [r, s] \subseteq [T]$, algorithm 2 achieves the following regret bound,*

$$\mathbb{E}\left[\sum_{t=r}^{s} l_t(w_t)\right] - \min_{w \in \mathbb{R}^d, \|w\| \leq R} \mathbb{E}\left[\sum_{t=r}^{s} l_t(w)\right] = O\left(\frac{d^3 R^2 (B_X + B_N)^4}{\sigma^2} \log T\right)$$

**Algorithm 2** AdaptiveFilter

```
Input: $k \in \mathbb{N}^+$, $\alpha \in \mathbb{R}^+$.
Let $E^1, ..., E^T$ be online convex optimization algorithms.
Let $p_1 \in \mathbb{R}^T, p_1^{(1)} = 1, \forall j : 1 < j \leq T, p_1^{(j)} = 0$.
for $c = 1...$ do
    $\forall j \leq c, w_c^{(j)} \leftarrow E^j(L_1^k, ..., L_{(c-1)}^k)$ (the filter of the j'th algorithm).
    $w_c \leftarrow \sum_{j=1}^{c} p_c^{(j)} w_c^{(j)}$.
    for $t = (c-1)k + 1...ck$ do
        predict: $x_t = w_c^\top Y_t$.
    end for
    $\hat{p}_{c+1}^{(c+1)} = 0$ and for $i \in [c]$,
        $$\hat{p}_{c+1}^{(i)} = \frac{p_c^{(i)} e^{-\alpha L_c^k(w_c^{(i)})}}{\sum_{j=1}^{c} p_c^{(i)} e^{-\alpha L_c^k(w_c^{(i)})}}$$
    $p_{c+1}^{(c+1)} = 1/(c+1)$ and for $i \in [c] : p_{c+1}^{(i)} = (1 - (c+1)^{-1})\hat{p}_{c+1}^{(i)}$ (adding
    expert $E^{(c+1)}$).
end for
```

**Corollary 2.** *Let $w_t$ be the filter used by algorithm 2 for prediction in time $t$. Let $k = 2d$, $R = \frac{\sqrt{d}B_X^2}{\sigma^2}$ and let $\alpha = \frac{d\sigma^2}{G(2d,R)^2}$. For all $I = [r, s] \subseteq [T]$, algorithm 2 achieves the following regret bound,*

$$\mathbb{E}\left[\sum_{t=r}^{s} l_t(w_t)\right] - \min_{w \in \mathbb{R}^d} \mathbb{E}\left[\sum_{t=r}^{s} l_t(w)\right] = O\left(\frac{d^4 B_X^4 (B_X + B_N)^4}{\sigma^6} \log T\right)$$

As in the previous section, we take the approach of partitioning the iterations into disjoint blocks of length $k$ and optimizing over the loss functions $L_t^k$.

The algorithm is based on the well known experts framework where each expert in our case, is a gradient descent filter presented in the previous section. On each block $c$, the algorithm adds a new expert that starts producing predictions from block $c + 1$ an onward. The experts algorithm predicts on each iteration by combining the filters of all experts using a weighted sum according to the weight of each expert. The key idea behind this framework is that an expert added at block $r$ achieves low regret on all intervals starting in $r$. Given such an interval, the experts algorithm itself achieves low regret on the interval with respect to this specific expert, and thus has low regret on the interval.

Expert $E^r$ could be thought of as an algorithm that plays $w_c = 0$ for all $c < r$ and starting at block $r$ plays according to algorithm 1.

For the complete proof of the theorem and corollary the reader is referred to the appendix.

## References

[1] Taesup Moon and Tsachy Weissman. Universal fir mmse filtering. *IEEE Transactions on Signal Processing*, 57(3):1068–1083, 2009.

[2] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series, With Engineering Applications*. New York: Wiley, 1949.

[3] Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*.

[4] H. Vincent Poor. On robust wiener filtering. *IEEE Trans. Automatic Control*, AC-25:521–526, Jun 1980.

[5] Yonina C. Eldar and Neri Merhav. A competitive minimax approach to robust estimation of random parameters. *IEEE Trans. Signal Processing*, 52:1931–1946, July 2004.

[6] Yonina C. Eldar, Aharon Ben-Tal, and Arkadi Nemirovski. Linear minimax regret estimation of deterministic parameters with bounded data uncertainties. *IEEE Trans. Signal Processing*, 52:2177–2188, Aug 2004.

[7] Simon Haykin. *Unsupervised Adaptive Filtering: Volume I, II*. New York:Wiley, 2000.

[8] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

[9] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[10] Elad Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *ICML*, page 50, 2009.

## A Proof of Theorems 1, 2

The proofs are based on [9, 10] and are brought here in full detail for completeness.

**Theorem 3.** *Let $w_t$ be the filter used by algorithm 1 for prediction in time $t$. Let $k$ = $2d$ and $H = d\sigma^2$. Algorithm 1 achieves the following regret bound,*

$$\mathbb{E}\left[\sum_{t=1}^{T} l_t(w_t)\right] - \min_{w\in\mathbb{R}^d, \|w\|\le R} \mathbb{E}\left[\sum_{t=1}^{T} l_t(w)\right] = O\left(\frac{d^3R^2(B_X+B_N)^4}{\sigma^2}\log T\right)$$

*Proof.* Again we assume w.l.o.g that $T = b\cdot k$ for some $b\in\mathbb{N}^+$. Consider some $w\in\mathbb{R}^d$ such that $\|w\|_2 \le R$. Define $\nabla_c = \nabla L_c^k(w_c)$ and $\nabla_c^2 = \nabla^2 L_c^k(w_c)$, $G = G(2d, R)$. Writing the Taylor series approximation of $L_c^k(w)$ around $w_c$ we have,

$$L_c^k(w) \;=\; L_c^k(w_c) + \nabla_c^\top(w-w_c) + \frac{1}{2}\nabla_c^2 \circ (w-w_c)(w-w_c)^\top$$

According to (4), $\nabla_c^2 \succeq 2(k-d+1)\sigma^2\mathbf{I}$ and we have,

$$L_c^k(w) \ge L_c^k(w_c) + \nabla_c^\top(w-w_c) + (k-d+1)\sigma^2\|w-w_c\|_2^2 \tag{6}$$

Following the analysis in [8, 9] we upper bound $\nabla_c^\top(w-w_c)$ by,

$$2\nabla_c^\top(w-w_c) \le \frac{\|w_c-w\|^2 - \|w_{c+1}-w\|^2}{\eta_{c+1}} + \eta_{c+1}G^2 \tag{7}$$

Summing over (7) for all $c$, using (6) we have,

$$\begin{aligned} 2\sum_{c=1}^{T/k} L_c^k(w_c) - L_c^k(w) \;&\le\; \sum_{c=1}^{T/k}\|w_c-w\|^2\left(H(c+1) - Hc - (k-d+1)\sigma^2\right) \\ &+\; G^2\sum_{c=1}^{T/k}\frac{1}{Hc} \end{aligned}$$

Plugging $H = d\sigma^2$ yields

$$\sum_{t=c}^{T/k} L_c^k(w_c) - L_c^k(w) = O\left(\frac{G^2}{d\sigma^2}\log T\right)$$

The theorem now follows from (1) and plugging $G = G(2d, R)$. □

In order to prove Theorem 2 we need two simple claims first. In what follows we assume that $L_c^k(w)$ is $\alpha$-exp-concave.

**Claim 1.** *1. For $i < c$,*

$$L_c^k(w_c) - L_c^k(w_c^{(i)}) \le \alpha^{-1}$$

*2.* $L_c^k(w_c) - L_c^k(w_c^{(c)} \le \alpha^{-1}(\ln \hat{p}_{c+1}^{(c)} + \ln c)$

*Proof.* Using the $\alpha$-exp concavity of $L_c^k$ we have

$$e^{-\alpha L_c^k(w_c)} = e^{-\alpha L_c^k(\sum_{j=1}^{c} p_c^{(j)} x_c^{(j)}} \ge \sum_{j=1}^{c} p_c^{(j)} e^{-\alpha L_c^k(x_c^{(j)}}$$

Taking logarithm,

$$L_c^k(w_c) \le \alpha^{-1} \ln \sum_{j=1}^{c} p_c^{(j)} e^{-\alpha L_c^k(w_c^{(j)})}$$

Thus,

$$\begin{aligned}
&L_c^k(w_c) - L_c^k(w_c^{(i)}) \\
&\qquad \le \alpha^{-1}\left(\ln e^{-\alpha L_c^k(w_c^{(i)})} - \ln \sum_{j=1}^{c} p_c^{(j)} e^{-\alpha L_c^k(w_c^{(j)})}\right) \\
&\qquad = \alpha^{-1} \ln \frac{e^{-\alpha L_c^k(w_c^{(i)})}}{\sum_{j=1}^{c} p_c^{(j)} e^{-\alpha L_c^k(w_c^{(j)})}} \\
&\qquad = \alpha^{-1} \ln \left(\frac{1}{p_c^{(i)}} \cdot \frac{p_c^{(i)} e^{-\alpha L_c^k(w_c^{(i)})}}{\sum_{j=1}^{c} p_c^{(j)} e^{-\alpha L_c^k(w_c^{(j)})}}\right) \\
&\qquad = \alpha^{-1} \ln \frac{\hat{p}_{c+1}^{(i)}}{p_c^{(i)}} \qquad (8)
\end{aligned}$$

Now, by definition it holds that for $i < c$, $p_c^{(i)} = (1 - 1/c)\hat{p}_c^{(i)}$. Also, $p_c^{(c)} = 1/c$. Plugging these two equalities into (8) yields the claim. □

**Claim 2.** *For any two integers $r, s$ such that $s > r$, it holds that*

$$\sum_{c=r}^{s} L_c^k(w_c) - L_c^k(w_c^{(r)}) \le \frac{4}{\alpha} \ln T$$

*Proof.* Using the previous claim we have,

$$\begin{aligned}
& \sum_{c=r}^{s} L_c^k(w_c) - L_c^k(w_c^{(r)}) \\
= \; & (L_r^k(w_r) - L_r^k(w_r^{(r)})) + \sum_{c=r+1}^{s} L_c^k(w_c) - L_c^k(w_c^{(r)}) \\
\leq \; & \alpha^{-1}\left(\ln \hat{p}_{r+1}^{(r)} + \ln r + \sum_{c=r+1}^{s} \ln \hat{p}_{c+1}^{(r)} - \ln \hat{p}_c^{(r)} + 2/c\right) \\
= \; & \alpha^{-1}\left(\ln r + \ln \hat{p}_{s+1}^{(r)} + \sum_{c=r+1}^{s} 2/c\right)
\end{aligned}$$

Since $\hat{p}_{s+1}^{(r)} \leq 1$, $\ln \hat{p}_{s+1}^{(r)} \leq 0$. This implies that the regret is bounded by $\frac{4}{\alpha} \ln T$. □

We can now prove Theorem 2.

**Theorem 4.** *Let $w_t$ be the filter used by algorithm 2 for prediction in time $t$. Let $k = 2d$ and let $\alpha = \frac{d\sigma^2}{G(2d,R)^2}$. For all $I = [r, s] \subseteq [T]$, algorithm 2 achieves the following regret bound,*

$$\mathbb{E}\left[\sum_{t=r}^{s} l_t(w_t)\right] - \min_{w \in \mathbb{R}^d, \|w\| \leq R} \mathbb{E}\left[\sum_{t=r}^{s} l_t(w)\right] = O\left(\frac{d^3 R^2 (B_X + B_N)^4}{\sigma^2} \log T\right)$$

*Proof.* Given an interval $I = [r, s] \subseteq [T]$, let $r = c_r \cdot k - b_r$, $s = c_s \cdot k + b_s$ such that $c_r, b_r, c_s, b_s \in \mathbb{N}$ and $0 \leq b_r, b_s \leq k - 1$.
Since $k$ is a constant independent of $T$, we ignore the first $b_r$ iterations and last $b_s$ iterations, since they only add a constant to the regret.
According to Claim 2 we have,

$$\sum_{c=c_r}^{c_s} L_c^k(w_c) - L_c^k(w_c^{(r)}) \leq \frac{4}{\alpha} \ln T = O\left(\frac{G(2d,R)^2}{d\sigma^2} \log T\right)$$

Since $E^r$ achieves low regret on all block-intervals beginning in block $r$ we have for all $w \in \mathbb{R}$ such that $\|w\|_2 \leq R$, □

$$\sum_{c=c_r}^{c_s} L_c^k(w_c^{(r)}) - L_c^k(w) = O\left(\frac{G(2d,R)^2}{d\sigma^2} \log T\right)$$

Thus we have,

$$\sum_{c=c_r}^{c_s} L_c^k(w_c) - L_c^k(w) = O\left(\frac{G(2d,R)^2}{d\sigma^2}\log T\right)$$

Again, the theorem now follows from (1) and plugging $G = G(2d, R)$.