NEW $\mathcal{H}^\infty$ BOUNDS FOR THE RECURSIVE LEAST SQUARES ALGORITHM EXPLOITING INPUT STRUCTURE

Koby Crammer[⋆] *Alex Kulesza*[†] *Mark Dredze*[‡]

[⋆]Department of Electrical Enginering, The Technion, Haifa, Israel
[†]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104
[‡]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD 21211
koby@ee.technion.ac.il kulesza@cis.upenn.edu mdredze@cs.jhu.edu

ABSTRACT

The recursive least squares (RLS) algorithm is well known and has been widely used for many years. Most analyses of RLS have assumed statistical properties of the data or the noise process, but recent robust $\mathcal{H}^\infty$ analyses have been used to bound the *ratio* of the performance of the algorithm to the total noise. In this paper, we provide an additive analysis bounding the *difference* between performance and noise. Our analysis provides additional convergence guarantees in general, and particular benefits for structured input data. We illustrate the analysis using human speech and white noise.

Index Terms— Adaptive estimation, Adaptive signal processing, Machine learning

1. INTRODUCTION

The recursive least squares (RLS) algorithm [1, 2] plays a major role in estimation theory for signal processing and online regression in machine learning. Most previous analyses of the RLS algorithm make statistical assumptions about the data, noise, or model (see, e.g., [3, 4]). Recently, however, there has been a growing interest in adaptive algorithms that can be proven robust to specific properties of the input using $\mathcal{H}^\infty$ theory [4]. Most previous $\mathcal{H}^\infty$ bounds characterize the ratio between the performance of an algorithm and the noise energy; for example, [4] provides such a bound for the Kalman filter (with RLS as a special case). These bounds can be quite loose if the bounding quantity is far from 1.

In this work we provide a new family of bounds that are additive in nature. Our analysis bounds the *difference* between the cumulative performance of the algorithm and the noise energy. Such bounds are more accurate in the presence of large amounts of noise. Furthermore, our bounds can be used to prove convergence of our algorithm to the best (linear) model, and typically also bound the rate of convergence. After presenting a variant of the RLS algorithm, we describe an initial bound that characterizes the tradeoff implicit in the learning rate employed by the algorithm. When the learning rate is too small, the algorithm under-fits, and its performance is bounded by a quantity proportional to the norm of the comparison vector model $\boldsymbol{u}$. On the other hand, if the learning rate is too large, the algorithm over-fits, and its performance is proportional to the cumulative noise. We show that if the learning rate is properly tuned, one can obtain an additive bound with a convergence rate decreasing as $1/\sqrt{N}$, where N is the number of inputs obtained so far. We then show a stronger additive bound with a convergence rate of $\log(N)/N$. This rate is faster if the inputs are structured in the sense that they are captured in a skewed ellipsoid. We conclude the paper by illustrating the bound using speech and white noise signals.

Input parameters : Tradeoff parameter $a > 0$, learning rate r
Initialize: $\boldsymbol{w}_0 = \mathbf{0}$, $P_0 = aI$
For $i = 1, \ldots, N, \ldots,$

- Receive an input-output pair $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$
- Compute prediction $m_i = {\boldsymbol{w}_{i-1}}^\top \boldsymbol{x}_i$
- Define loss $\ell_i = (m_i - y_i)^2$
- Compute:

$$\beta_i = \frac{1}{\boldsymbol{x}_i^\top P_{i-1} \boldsymbol{x}_i + r} \quad ; \quad \alpha_i = (m_i - y_i)\,\beta_i \tag{1}$$

- Update

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \alpha_i P_{i-1} \boldsymbol{x}_i \tag{2}$$

$$P_i = P_{i-1} - \beta_i P_{i-1} \boldsymbol{x}_i \boldsymbol{x}_i^\top P_{i-1} \tag{3}$$

Fig. 1. The RLS algorithm.

2. RECURSIVE LEAST-SQUARES

Adaptive algorithms maintain a function from the input space to the output space, parameterized by $\boldsymbol{w}$, and update the function in iterations or rounds. On iteration i the algorithm receives a training example as a pair: an input vector $\boldsymbol{x}_i \in \mathbb{R}^d$ and a desired output scalar $y_i \in \mathbb{R}$, which is used to update the current parameter $\boldsymbol{w}_i$. Linear functions of the form $f(\boldsymbol{x}) = \boldsymbol{w}_i^\top \boldsymbol{x}$ are commonly used for their simplicity and reliable performance on many practical tasks.

Although we are focusing on adaptive algorithms, it is useful to formulate the estimation problem with a single fixed filter. One common criterion is the regularized least-squares objective, which dates back to Gauss [5]:

$$\min_{\boldsymbol{w}} \left\{ a(\boldsymbol{w}^\top \boldsymbol{w}) + \sum_{i=1}^{N} \left(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i \right)^2 \right\} ,$$

for some tradeoff parameter $a > 0$. The recursive least-squares (RLS) algorithm [1, 2] solves this optimization problem adaptively, using only a single input-output instance at a time. The pseduo code of the algorithm is given in Fig. 1. The variant of the RLS algorithm we use employs a learning rate denoted by r. Using the Woodbury identity we can rewrite the update of the matrix P_i in (3) as

$$P_i^{-1} = P_{i-1}^{-1} + \frac{\boldsymbol{x}_i \boldsymbol{x}_i^\top}{r} . \tag{4}$$

As a consequence we get that the eigenvalues of the update matrix P_i in (3) are monotonically decreasing:

$$P_i \preceq P_{i-1} \quad \Leftrightarrow \quad P_i^{-1} \succeq P_{i-1}^{-1} \,, \tag{5}$$

where $A \preceq B$ if and only if $B - A$ is a PSD matrix. We begin with an analysis of the RLS algorithm.

3. ANALYSIS

Our analysis of the RLS algorithm considers two prediction settings: the first is motivated by adaptive prediction, and the second by online regression in the context of machine learning. In the first case we assume the existence of a linear model $\boldsymbol{u}$ which, together with an additive noise v_i, was used to generate the outputs. That is,

$$y_i = \boldsymbol{u}^\top \boldsymbol{x}_i + v_i \,.$$

In this setting the goal of the algorithm is to do well compared with the (unseen) true output $\boldsymbol{u}^\top \boldsymbol{x}_i$, and our analysis compares the total prediction loss with the total noise. Specifically, we denote the instantaneous *a priori* error with respect to the target output $\boldsymbol{u}^\top \boldsymbol{x}_i$ and the cumulative *a priori* loss, respectively, by

$$p_i = \boldsymbol{w}_{i-1}^\top \boldsymbol{x}_i - \boldsymbol{u}^\top \boldsymbol{x}_i \qquad \mathcal{A}^{(N)} = \sum_{i=1}^{N} p_i^2 \,. \tag{6}$$

We define the instantaneous noise and cumulative noise as, respectively,

$$v_i = \boldsymbol{u}^\top \boldsymbol{x}_i - y_i \qquad \mathcal{V}^{(N)} = \sum_{i=1}^{N} v_i^2 \,. \tag{7}$$

The goal of the bound is to relate the cumulative *a priori* loss $\mathcal{A}^{(N)}$ with the cumulative noise $\mathcal{V}^{(N)}$.

Machine learning online regression offers an alternative view. Here, we do not assume the existence of a model $\boldsymbol{u}$ that generated the (corrupted) outputs. Instead, the goal of the algorithm is to accurately predict the quantities y_i given $\boldsymbol{x}_i$. The algorithm is compared against the best function in some class; here we use the class of linear models. Specifically, we denote the instantaneous error of the algorithm by ℓ_i and the cumulative error by $\mathcal{L}^{(N)}$:

$$\ell_i = \boldsymbol{w}_{i-1}^\top \boldsymbol{x}_i - y_i \qquad \mathcal{L}^{(N)} = \sum_{i=1}^{N} \ell_i^2 \tag{8}$$

Again, in this setting we *do not* assume the existence or knowledge of a specific mechanism that generated the outputs y_i. Since we have no assumption about the way outputs are actually created, we compare our success at predicting y_i with the best possible linear model given the entire sequence of input and outputs. Under this view, the quantity v_i is not the amount of additive noise with respect to a *true* output, but just the amount of error a comparison model $\boldsymbol{u}$ suffers over the same sequence. In this respect the machine learning view is natural as typical bounds are comparing two quantities of the same type ($\mathcal{L}^{(N)}$ and $\mathcal{V}^{(N)}$), both of which measure the total loss over the sequence. The former is the loss of the algorithm (which employs a sequence of linear models) and the later is the loss of a single optimal model. On the other hand, the adaptive prediction view naturally motivates the use of linear models, as we assume that the process which generated the outputs is indeed linear.

A typical $\mathcal{H}^\infty$ bound [4] is multiplicative, having the form

$$\mathcal{A}^{(N)} \leq B \left(\mathcal{V}^{(N)} + (1/a) \boldsymbol{u}^\top \boldsymbol{u} \right) \,.$$

In words, the bound is on the ratio between the algorithm's *a priori* cumulative loss and the sum of the regularization and cumulative noise. The bound tells us that, independently of the choice of target function $\boldsymbol{u}$ and noise $\{v_i\}$, the total loss is not larger than B times the total noise. Such bounds are informative if the total loss is close to zero and if B is relatively small (order of 1), otherwise the quantity $B\mathcal{V}^{(N)}$ is too large and the bound becomes uninformative. A major drawback of these bounds is that they do not guarantee that the performance of the algorithm, in terms of the cumulative (*a priori* or prediction) loss, will eventually converge to the loss obtained by the reference model $\boldsymbol{u}$, even after observing large (even infinite) amounts of data. This is because the ratio of $\mathcal{A}^{(N)}$ and $\mathcal{V}^{(N)}$ is bounded only by B which often is strictly greater than 1.

We now describe and develop bounds on the cumulative loss of the algorithm (either $\mathcal{A}^{(N)}$ or $\mathcal{L}^{(N)}$) that are additive in nature and are of the form

$$\mathcal{A}^{(N)} \leq \mathcal{V}^{(N)} + C^{(N)} \,,\; \mathcal{L}^{(N)} \leq \mathcal{V}^{(N)} + D^{(N)} \,,$$

where $C^{(N)}$ and $D^{(N)}$ are not necessarily constant in the number of input-output pairs N. However, as long as $C^{(N)}$ or $D^{(N)}$ are sub-linear in N we can conclude that the average loss of the algorithm will converge to the average noise (or average loss of the reference model $\boldsymbol{u}$), that is,

$$\mathcal{A}^{(N)}/N \to \mathcal{V}^{(N)}/N \,,\; \mathcal{L}^{(N)}/N \to \mathcal{V}^{(N)}/N \,.$$

The quantities $C^{(N)}$ and $D^{(N)}$ provide the rate of convergence. We start with a general multiplicative bound and show that it can be tuned to get an additive bound. We then give a general additive bound.

3.1. First Bound

Let $\Psi_i = (\boldsymbol{w}_i - \boldsymbol{u})^\top P_i^{-1} (\boldsymbol{w}_i - \boldsymbol{u})$. We start with the following lemma:

Lemma 1 *Let $\boldsymbol{u} \in \mathbb{R}^d$ be an arbitrary weight vector. Then for all i,*

$$\Psi_i - \Psi_{i-1} = -\frac{\ell_i^2}{\chi_i + r} + \frac{v_i^2}{r}, \text{ where } \chi_i = \boldsymbol{x}_i^\top P_{i-1} \boldsymbol{x}_i \,. \tag{9}$$

The proof is omitted due to lack of space. We now prove the first result of this section.

Theorem 2 *Assume that $\sup_i \|\boldsymbol{x}_i\|^2 \leq X^2$ for some $X > 0$ and $P_0 = a\, I$. Let r be rewritten as $r = aX^2 \frac{1-\nu}{\nu}$ for some $\nu \in (0, 1]$. The cumulative loss of the algorithm is bounded as follows:*

$$\mathcal{L}^{(N)} \leq \frac{X^2 \|\boldsymbol{u}\|^2}{\nu} + \frac{\mathcal{V}^{(N)}}{1-\nu} \,. \tag{10}$$

Proof: We write the following telescoping sum,

$$-\Psi_0 \leq \Psi_N - \Psi_0 = \sum_{i=1}^{N} \left(\Psi_i - \Psi_{i-1} \right) \,,$$

where the first inequality follows since $\Psi_N \geq 0$. Substituting Lemma 1 we get $-\Psi_0 \leq -\sum_i \frac{\ell_i^2}{\chi_i + r} + \sum_i \frac{v_i^2}{r}$. Rearranging the terms gives

$$\sum_i \frac{\ell_i^2}{\chi_i + r} \leq \Psi_0 + \sum_i \frac{v_i^2}{r} \,. \tag{11}$$

Substituting $\chi_i \leq a \|\boldsymbol{x}_i\|^2 \leq aX^2$ (from (5)) and $\Psi_0 = \|\boldsymbol{u}\|^2 / a$:

$$\sum_i \frac{\ell_i^2}{aX^2 + aX^2 \frac{1-\nu}{\nu}} = \sum_i \frac{\ell_i^2}{\frac{aX^2}{\nu}} \leq \frac{\|\boldsymbol{u}\|^2}{a} + \sum_i \frac{v_i^2}{aX^2 \frac{1-\nu}{\nu}} .$$

Multiplying both sides with aX^2/ν yields the bound. ∎

A few comments are in order. First, the Least-Mean-Square (LMS) algorithm [6], in which the weights are updated using the rule $\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \eta \left(y_i - \boldsymbol{x}_i \cdot \boldsymbol{w}_{i-1}\right) \boldsymbol{x}_i$ for some $\eta > 0$ has similar loss bounds [7]:

$$\mathcal{L}^{(N)} \leq \frac{X^2 \|\boldsymbol{u}\|^2}{\eta} + \frac{\mathcal{V}^{(N)}}{1-\eta} .$$

The two bounds are equivalent, when identifying η and ν. In practice, however, RLS often outperforms the LMS algorithm. Second, our bound for RLS and the analysis of LMS have a multiplicative factor strictly greater than 1, i.e. $1/(1-\eta)$ or $1/(1-\nu)$ for RLS and LMS, respectively. There is also an additional additive *constant* factor $\|\boldsymbol{u}\|^2 X^2/\eta$ for LMS or $\|\boldsymbol{u}\|^2 X^2/\nu$ for RLS. Third, interestingly, the bound is explicitly independent of the value of the regularization parameter a. A possible explanation is that it does depend on a implicitly via the definition of ν (or r).

Furthermore, we now state a bound for an algorithm that uses an optimal learning rate.

Corollary 3 *Under the conditions of Theorem 2, if the algorithm is run with*

$$\nu = \frac{\sqrt{\|\boldsymbol{u}\|^2 X^2}}{\sqrt{\|\boldsymbol{u}\|^2 X^2} + \sqrt{\mathcal{V}^{(N)}}} , \tag{12}$$

then the cumulative loss it suffers is optimally bounded by

$$\mathcal{L}^{(N)} \leq \mathcal{V}^{(N)} + X^2 \|\boldsymbol{u}\|^2 + 2\sqrt{X^2 \|\boldsymbol{u}\|^2 \mathcal{V}^{(N)}} . \tag{13}$$

The bound holds by substituting (12) in (10), and is optimal since ν was chosen to minimize (10). To actually compute this optimal learning rate ν we need to know or bound the energy of the noise $\mathcal{V}^{(N)}$ and the norm of the reference vector $\boldsymbol{u}$. The former can be actually measured or estimated in many applications, while the later can be bounded or approximated.

We observe that even when we optimize the learning parameter r (or ν) the difference between the cumulative loss of the algorithm and the cumulative loss of a fixed weight vector $\boldsymbol{u}$ is bounded by a quantity proportional to $\sqrt{\mathcal{V}^{(N)}}$, which is on the order of $\sqrt{N}$, the number of input-output pairs. In other words, the rate at which the (averaged) performance of the algorithm goes to the (averaged) performance of any weight-vector is monotonically decreasing on the order of $1/\sqrt{N}$.

Before proceeding to a better bound in which the additive term $D^{(N)}$ will be logarithmic in the number of input-output pairs, we sketch a theorem and proof bounding the cumulative *a priori* loss as opposed to the previously used prediction loss. From (6), (7) and (8) it is easily verified that

$$\ell_i = p_i + v_i . \tag{14}$$

Using [4, Lemma 2] we get that for all $\alpha > 1$,

$$\ell_i^2 \geq \left(1 - \frac{1}{\alpha}\right) p_i^2 + (1-\alpha) v_i^2 . \tag{15}$$

Substituting back in (10),

$$(1-\alpha)\mathcal{V}^{(N)} + \left(1 - \frac{1}{\alpha}\right)\mathcal{A}^{(N)} \leq \mathcal{L}^{(N)} \leq \frac{X^2 \|\boldsymbol{u}\|^2}{\nu} + \frac{\mathcal{V}^{(N)}}{1-\nu} .$$

Algebraic manipulation yields

$$\mathcal{A}^{(N)} \leq \frac{X^2 \|\boldsymbol{u}\|^2}{\nu\left(1 - \frac{1}{\alpha}\right)} + \mathcal{V}^{(N)} \frac{\frac{1}{1-\nu} - (1-\alpha)}{\left(1 - \frac{1}{\alpha}\right)} . \tag{16}$$

At this point we can leave the result as is, optimize the right hand side with respect to α, or set a specific value for α. We choose the latter option by minimizing only the right term of the bound over α and get $\alpha = 1 + \sqrt{\frac{1}{1-\nu}}$. Substituting back in (16) we have

$$\mathcal{A}^{(N)} \leq \frac{X^2 \|\boldsymbol{u}\|^2}{\nu\left(1 - \frac{1}{1+\sqrt{\frac{1}{1-\nu}}}\right)} + \mathcal{V}^{(N)} \left(1 + \sqrt{\frac{1}{1-\nu}}\right)^2 . \tag{17}$$

The coefficient $1/\left[\nu\left(1 - \frac{1}{1+\sqrt{\frac{1}{1-\nu}}}\right)\right]$ of the left term is decreasing in ν while the coefficient $\left(1 + \sqrt{\frac{1}{1-\nu}}\right)^2$ of the right term is increasing in ν, and thus there is, like the bound of Corollary 3, an optimal value of ν that minimizes the bound. To summarize, we have proved the following theorem.

Theorem 4 *Under the conditions of Theorem 2, Eq.* (17) *holds.*

3.2. Second Bound

We now provide a more refined analysis that yields an additive bound with an additive term $C^{(N)}$ or $D^{(N)}$ that is only logarithmic in N if the input data is well behaved. Furthermore, this bound holds for all values of the learning parameter r.

Theorem 5 *Assume that* $\sup_i \|\boldsymbol{x}_i\|^2 \leq X^2$ *for some* $X > 0$ *and* $P_0 = aI$. *The cumulative loss of the algorithm is bounded as follows:*

$$\mathcal{L}^{(N)} \leq \mathcal{V}^{(N)} + A \log\left(\det\left(aI + \frac{1}{r}\sum_i^N \boldsymbol{x}_i \boldsymbol{x}_i^\top\right)\right) + r\frac{\|\boldsymbol{u}\|^2}{a} ,$$

where $A = \sup_i \left\{\ell_i^2\right\}$.

Proof: We start with (11) and use the fact that $\frac{1}{s+t} = \frac{1}{s} - \frac{t}{s(s+t)}$ to obtain

$$\begin{aligned}\sum_i \frac{\ell_i^2}{r} &\leq \sum_i \frac{v_i^2}{r} + \sum_i \frac{\ell_i^2 \chi_i}{r\left(\chi_i + r\right)} + \Psi_0 \\ &\leq \sum_i \frac{v_i^2}{r} + \sup_i\left\{\frac{\ell_i^2}{r}\right\} \sum_i \frac{r\chi_i}{r\left(\chi_i + r\right)} + \Psi_0 . \end{aligned} \tag{18}$$

Using a lemma very similar to [8, Lemma 4],

$$\sum_i \frac{\ell_i^2}{r} \leq \sum_i \frac{v_i^2}{r} + \sup_i\left\{\frac{\ell_i^2}{r}\right\} \log\left(\det\left(P_N^{-1}\right)\right) + \Psi_0 .$$

Setting $\Psi_0 = \frac{\|\boldsymbol{u}\|^2}{a}$ we get,

$$\sum_i \ell_i^2 \leq \sum_i v_i^2 + \sup_i\left\{\ell_i^2\right\} \log\left(\det\left(P_N^{-1}\right)\right) + \frac{r}{a}\|\boldsymbol{u}\|^2 .$$

Substituting the value of P_N^{-1} concludes the proof. ■

We first claim that $A = \sup_i \left(\ell_i^2/r\right)$ is constant in N, i.e., it is finite. Since the eigenvalues of P_i go to zero, the RLS algorithm always converges. Therefore, under the reasonable assumption that the output y_i is bounded (that is $\sup_i |y_i| < \infty$) we get that A is constant with respect to N.

Both the bound of Theorem 2 and the optimal bound in (13) have two terms: a constant term $r\|\boldsymbol{u}\|^2$ or $X\|\boldsymbol{u}\|^2$, and a term dependent on the number of examples. In (13) it grows like the square-root of N. In Theorem 5, however, this term grows like $\log(N)$. To see this, we use convexity and bound

$$\log\det\left(aI + \frac{1}{r}\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^\top\right) \le d\log\left(a + \frac{NX^2}{rd}\right) .$$

Note the that unlike the bound of Theorem 2, which is independent of a, this bound is more refined and does depend explicitly on a. For large values of a the second term increases (the algorithm is forced to estimate $\boldsymbol{w}_i$ close to zero) and the third term decreases (the norm of the competitor $\boldsymbol{w}$ is less relevant), and vice versa for small values of a. Finally, there is a similar tradeoff with the learning rate; for large values of r (slow learning), the second term is small, as the algorithm is less likely to over-fit, and the third term is large, as the comparison model $\boldsymbol{w}$ is more important (the performance depends on the hardness of the problem, and not on the learning process itself). We can derive bounds for the estimation problem analogously to the derivation of Theorem 4. We omit the details due to lack of space.

4. ILLUSTRATION AND CONCLUSIONS

We illustrate some properties of the second bound using a one-second segment of human speech sampled at 16KHz as the input signal $\mathbf{s}$, and compare it to random noise $\mathbf{n}$ of the same length. Both signals were equalized to have the same energy. We simulated filters of various tap sizes $M = 2^m$ for m=3, ..., 11. Each of the two input signals (speech $\mathbf{s}$ and noise $\mathbf{n}$) was transformed into a vector by taking the last M samples, specifically $\boldsymbol{x}_i = [s_i, s_{i-1} \dots s_{i-M+1}]$ for the speech signal, and similarly for the noise signal. We then compute the second term of the bound in Theorem 5, that is, $\log\left(\det\left(aI + \frac{1}{r}\sum_i^N \boldsymbol{x}_i\boldsymbol{x}_i^\top/d\right)\right)$. Note that we normalized each of the instances $\boldsymbol{x}_i$ by the actual dimension d (or M) to compensate for the increased covariance due to increased dimension.

We set $a = 0.01$ and $r = 1$. The value of the bound compared with the tap size M is shown in the left panel of Fig. 2. We observe that for both signals the bound increases as the tap size gets larger. This is because as we increase the memory M we capture more structure in the signal, increase the complexity of the representation, and expect to make more prediction errors. There is a difference between the two signals, however. Random noise, by construction, does not have any structure, and thus the bound increases almost linearly with the tap size M (which is the dimension of the input vectors $\boldsymbol{x}_i$). On the other hand, human speech contains regularities at various scales, and thus the bound is clearly sub linear with the tap size M. We thus expect that the RLS algorithm will perform better on speech signals as opposed to white noise. Finally, the corresponding term of the bound of Theorem 2 is strictly larger than this bound (not shown). This is because Theorem 2 does not take into consideration any properties of the input besides the energy (norm), while Theorem 5 takes additional spectral properties of the covariance matrix into consideration. As a result, we can plot this part of the bound, ignoring the specific task at hand, as the term we plot is a function solely of the input.

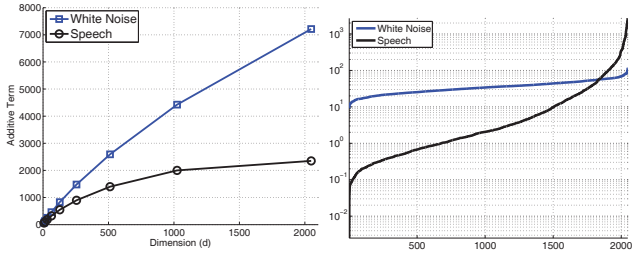

Fig. 2. Left: value of second term in bound of Theorem 5 vs. the tap size M, which is the dimension of the input space d. Right: the eigenvalue of the covariance matrix of both signals for tap size $M = d = 2048$

We further analyze the difference between the two settings in the right panel of Fig. 2 where we plot the eigenvalues of the matrix $aI + \frac{1}{r}\sum_i^N \boldsymbol{x}_i\boldsymbol{x}_i^\top$ for the same values of $a = 0.01$ and $r = 1$. Eigenvalues corresponding to white noise have nearly uniform values, as expected from pure noise. On the other hand, the input signal is structured, as evident from the large difference between the eigenvalues of the covariance matrix corresponding to the speech signal.

In summary, our analysis is additive in nature and provides better convergence bounds than current $\mathcal{H}^\infty$ results. Future work includes modifying RLS to yield improved convergence rates. Specifically, one of our goals is to replace Theorem 5 with a more refined version having smaller and more controlled values of the leading coefficient, denoted by A.

5. REFERENCES

[1] R.L. Plackett, "Some theorems in least-squares," *Biometrika*, vol. 37, pp. 149, 1950.

[2] R.L. Plackett, "The discover of the method of least-squares," *Biometrika*, vol. 59, pp. 239–251, 1972.

[3] Ali H. Sayed, *Adaptive Filters*, Wiley-IEEE Press, 2008.

[4] B. Hassibi and T. Kailath, "H-infinity bounds for least-squares estimators," *IEEE Transactions on Automatic Control*, vol. 46, pp. 309–14, 2001.

[5] G. W. Stwart, "Gauss, statistics, and gaussian elimination," *Journal of Computational and Graphical Statistics*, vol. 4, no. 1, Mar 1995.

[6] B. Widrow and M.E. Hoff Jr., "Adaptive switching circuits," in *Proceedings of IRE WESCON Convention Record 4*, 1960, pp. 96–104.

[7] N. Cesa-Bianchi, P. Long, and M.K. Warmuth, "Worst-case quadratic loss bounds for a generalization of the widrow-hoff rule," in *Proceedings of the Sixth Annual Workshop on Computational Learning Theory*, 1993, pp. 429–438.

[8] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weighted vectors," in *NIPS 23*, 2009.