Applications of Random Matrix Theory in Wireless Underwater Communication or Why Signal Processing and Wireless Communication Need Random Matrix Theory

Atulya Yellepeddi[*]

May 12, 2013

Abstract

Recently, there has been an enormous amount of interest in the use of random matrix theory (RMT) methods in signal processing. The application of random matrix methods to the analysis of techniques that have been used and studied for years in signal processing and wireless communications has shed new light onto the behavior of these methods. It has allowed us to tractably explore phenomena that we had not been able to understand before, which is one of the reasons for the immense excitement regarding RMT in the signal processing community. In this paper, I describe one important problem in signal processing to which random matrix methods apply quite naturally- that of sample covariance matrix estimation. This estimation problem is at the heart of many widely used signal processing algorithms. One such algorithm is the least squares channel estimation in wireless underwater communication and. The implications of the random matrix analysis of the SCM on these algorithms is described. It is hoped that this will shed light on the underlying issues and reasons for using RMT in signal processing.

[*] Prepared for MIT Class 18.338- Eigenvalues of Random Matrices, Spring 2013

1 Motivation- A Brief Personal Aside

Of late, it seems that everyone around me is talking about random matrix theory in signal processing. It has emerged as a reliable method to analyze the algorithms that we take advantage of in wireless underwater communication (which is one of my primary research interests) in the regimes that we are interested in. There has been work in various places, including my own research group, that leverages these tools.

My own work has not taken advantage of random matrix methods, but it seemed like it could benefit from them. That was one of the primary motivations for me to take this class- to gain a better understanding of the methods and why they would be applicable to our work. The former is, clearly, a prerequisite to understanding the latter.

With the understanding gained over the semester, I am now in a better position to parse some of the work that has been done in signal processing. That is the prime motivation for me, personally, to do this project- it'll give me a reason to survey some of the results that are in my field. I hope that it will also serve as an interesting exposition of this area.

2 Signal Processing and the Law of Large Numbers

Much of statistical signal processing revolves around detection and parameter estimation. Suppose we want to estimate something about a system, and that has dimension m, i.e., we want to estimate a "population size" m. We have n observations of the population.

A lot of the methods used are based on the asymptotic statistics of the population [1], i.e., the statistics as $n \to \infty$; because as the Law of Large Numbers tells us, the behavior once we have a large number of observations becomes "nice". In other words, the problem of finding and analyzing estimators becomes tractable as $n \to \infty$.

In practice, this kind of behavior generally kicks in when $n >> m$. Unfortunately, in modern signal processing, this assumption is often unrealistic. For instance:

- When we are trying to estimate a parameter of a time-varying system, where the parameter has dimension m. Before the system changes, we only get a small number of observations. Technically, the coherance time of the system, the time within which the system is nearly stationary, might be small, so the number of correlated observations which are useful for estimation is small. The faster the channel varies, the fewer the observations we have. The wireless underwater communication system [2] and the stock market [3] are examples of this.

- The population size might itself grow as the number of observations grows. Systems like social networks and biological networks [4] are good examples of this.
- Where the population size is innately large, for instance fast-moving object tracking with a large number of radar sensors [5].

In all these cases, it is usually fair to assume that $n \sim m$. The study of large dimensional random matrices has proved to be *remarkably* good at making predictions about the behavior of such systems. We will see a simple example in this paper, but a gamut of applications and the corresponding predictions can be found in [6].

2.1 Why Does Random Matrix Theory Help?

Before proceeding to the examples that concretize the notions here, an intuitive explanation of why the predictions are so good may be in order. To my mind, there are 2 main reasons for this.

2.1.1 "For Mathematicians, $30 \approx \infty$"

This comment, made offhand in one of our classes, is almost perfect when applied to signal processing. The parameters that we care about estimating are generally on the order of a few 10s (or more). Large dimensional random matrix theory is, as we have noted in this course, good for matrices of such dimensions! This is at the heart of why it works so well- convergence to the $m \to \infty$ is extremely fast, so for the parameter sizes of interest, this is all we really need to make excellent predictions.

Of course, we could use the exact theory (for finite dimensions), but the fact is that the large dimensional predictions are already so good, and lead to so much simpler predictions, that there's no real benefit of doing so.

2.1.2 "Almost Anything is IID Gaussian"

While it is true that the bulk of Random Matrix Theory results are for IID Gaussian entries in the matrix, we often do not encounter such matrices in practice. It turns out, however, that it does not matter! The results seem to work well for any matrix, as long as the distributions are "reasonably" like Gaussians, and as long as the entries are "somewhat" independent (what these notions mean, of course, needs to be formalized in a given environment). This is the other reason that the predictions are so good- no matter what distribution the samples are actually drawn from, the results of RMT seem to apply.

There are of course a couple of caveats. The convergence rate to these results can be affected, and if the distributions are really far apart, we'd have to take the IID Gaussian ensemble based predictions with a grain of salt. That said, practically speaking, the results have been observed to be very good.

We now proceed to see an example of the ideas above for a particular problem- that of channel estimation in wireless communication. We look at the predictions made and how they shed light on phenomena that have not otherwise been explained.

2.2 A Note on Notation

In what follows, boldface, lowercase math represents vectors (for instance $\boldsymbol{u}$) and boldface uppercase (like $\boldsymbol{U}$) represents matrices. Non-boldface are scalars. Observations at time k are denoted like $u(k)$ (so for instance, $\boldsymbol{u}(k)$ would represent a vector valued observation $\boldsymbol{u}$ at time k). $\mathbb{E}[\cdot]$ denotes expected value and Tr denotes trace. † and ^{T} are Hermitian and Transpose of a matrix, respectively. Finally, $\mathcal{N}(\cdot,\cdot)$ refers to a Gaussian distribution where the arguments are, respectively, the mean and the covariance matrix.

3 Least Squares Channel Estimation and Sample Covariance Matrix Estimation

In this section we describe a specific problem- that of Least Squares Channel Estimation. In the next one, we apply random matrix theory to analyze it and discuss the results.

In communication systems, a *channel* is something that relates a transmitted signal to a received signal. In other words, a transmitted signal passes through the channel, and a distorted version is captured by receivers at the other end. The channel that we refer to is the mathematical model of that distortion.

To take a specific example, suppose that we want to transmit $m \times 1$ vectors at each time $n = 1, 2, \ldots$. We denote the transmitted vector at time n by $\boldsymbol{u}(n)$. Assume that $\boldsymbol{u}(n)$ are random vectors with 0 mean and covariance matrix $\boldsymbol{R}$, which is termed the data covariance matrix. Assume further that these vectors are independent from time to time, so that $\mathbb{E}[\boldsymbol{u}(n)\boldsymbol{u}^\dagger(m)] = \boldsymbol{R}\delta(n-m)$, where δ is the Dirac delta function.

The last constraint, that of independent input observations, is introduced for simplicity. It is unlikely to be met in practice- indeed it is somewhat artificial in the context of communication systems, as inputs are generally vectors composed of the current transmitted symbol, and the past $m-1$ symbols. Independence of such vectors is evidently a rather unrealistic assumption. However, as will be seen, we will be able to relax it, because random matrix theory will give us the tools to deal with the case of a correlated input process (this is certainly another benefit of the random matrix framework for analysis).

We suppose that the $\boldsymbol{u}$s are passed through a finite, time-invariant linear channel $\boldsymbol{w}_0$. So, the output of the channel at some time n is given by the *scalar*:

$$d(n) = \boldsymbol{w}_0^\dagger \boldsymbol{u}(n) + v(n) \tag{1}$$

where $v(n)$ is additive noise. The noise is zero mean, IID, with variance σ_v^2, i.e., $\mathbb{E}[v(n)v(m)] = \sigma_v^2\delta(n-m)$. It is further assumed that the noise process is independent of the input process.

This is the classical model of a linear, time invariant Intersymbol Interference (ISI) channel. It is so called because the operation of $\boldsymbol{w}_0$ causes the various elements of $\boldsymbol{u}(n)$ to interfere with each other.

3.1 Least Squares Channel Estimation

One important problem in communication systems is learning $\boldsymbol{w}_0$ from the $\boldsymbol{u}$s. This is termed *channel estimation.* If the statistics $\boldsymbol{R}$ and $\mathbb{E}[\boldsymbol{u}(n)d^*(n)]$ were known, then the Weiner filter (MMSE) solution to estimating $\boldsymbol{w}_0$ would be given by:

$$\hat{\boldsymbol{w}}_{\text{MMSE}} = \boldsymbol{R}^{-1}\mathbb{E}[\boldsymbol{u}(n)d^*(n)] \tag{2}$$

As these are usually not known, we replace them by their sample values, giving the so-called *Least Squares* solution [7]. This is given by:

$$\hat{\boldsymbol{w}}(n) = \boldsymbol{R}^{-1}(n)\boldsymbol{z}(n) \tag{3}$$

where $\boldsymbol{R}(n)$ is an estimated correlation matrix, computed as:

$$\boldsymbol{R}(n) = \sum_{i=1}^{n} \boldsymbol{u}(i)\boldsymbol{u}^{\dagger}(i) + \delta\boldsymbol{I} \tag{4}$$

This is the famous *Sample Covariance Matrix* (for zero mean inputs). The idea is that as $n \to \infty$, $\boldsymbol{R}(n)/n \to \boldsymbol{R}$, by the Strong Law of Large Numbers.

Also, $\boldsymbol{z}(n)$ is the sample input-output cross-correlation, given by

$$\boldsymbol{z}(n) = \sum_{i=1}^{n} \boldsymbol{u}(i)d^*(i) \tag{5}$$

A small δ is added in (4) as a diagonal loading parameter which enables the algorithm to run from the very first observation.

3.2 Performance Metrics

How well does our estimator of (3) do? We compare performance based on the following:

- The channel estimation error:

$$\epsilon(n) = \boldsymbol{w}_0 - \hat{\boldsymbol{w}}(n) \tag{6}$$

- The signal prediction error:

$$e(n) = d(n) - \hat{\boldsymbol{w}}^{\dagger}(n)\boldsymbol{u}(n) \tag{7}$$

With the assumption of large n made, so that $\frac{1}{n}\boldsymbol{R}(n) \approx \boldsymbol{R}$, it can be shown that:

$$\mathbb{E}[\|\epsilon(n)\|_2^2] \approx \frac{1}{n}\sigma_v^2 tr\{\mathbf{R^{-1}}\} \tag{8a}$$

$$\mathbb{E}[\|e(n)\|_2^2] \approx \frac{1}{n}\sigma_v^2 tr\{\mathbf{R^{-1}}\} + \sigma_v^2 \tag{8b}$$

3.3 What's Wrong with (8)?

Well, it turns out, nothing, provided $n \gg m$. In other words, the approximations of (8) are a good representation of the performance of the system provided that the number of observations is much greater than the channel length (or more generally, the number of parameters we want to estimate).

However, in adaptive signal processing, such as in underwater communication, we can usually only assume that the channel is "time-invariant" for a short time- specifically, until it changes! Generally, what this means is that we end up with $n \sim m$ observations. The predications of (8) don't work well for these cases.

3.4 Why Large-Dimensional Random Matrix Theory?

When we can not assume that $\frac{1}{n}\boldsymbol{R}(n) \approx \boldsymbol{R}$ it becomes important to characterize the statistics of the SCM. The results of large-dimensional random matrix theory seem to work well at such characterizations, as they provide (relatively) tractable ways to deal with the "small" number of observations, for reasons discussed in Section 2.1 . These might give us more insight into the performance of the algorithm as $n \sim m$. So we look at these results next.

4 Random Matrix Theory Results

In [8], some results of interest to this problem were obtained. These are briefly summarized here. First, define the matrix moment:

$$M_k(m,n) = \frac{1}{m}\mathbb{E}\left[\mathrm{Tr}\left(\boldsymbol{R}^{-k}(n)\right)\right] \tag{9}$$

Then the following expressions can be derived for the channel estimation error:

$$\mathbb{E}\left[\|\epsilon(n)\|_2^2\right] = m\left(\sigma_v^2 M_1(m,n) + \delta^2 M_2(m,n) - \delta\sigma_v^2 M_2(m,n)\right) \tag{10}$$

...and for the signal prediction error:

$$\mathbb{E}\left[\|e(n)\|^2\right] = \sigma_v^2 - m\sigma_v^2 M_1(m,n) + m(\delta - \sigma_v^2)\delta M_2(m,n) \tag{11}$$

The question is how to evaluate the moments $M_k(m,n)$ for the matrix $\boldsymbol{R}(n)$. This is what random matrix theory can help us answer. Two distinct cases are considered in the paper.

4.1 Evaluating the Moments

4.1.1 Independent Input Observations

This is the case introduced in the problem statement. Define the matrix $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{u}(1) & \boldsymbol{u}(2) & \dots & \boldsymbol{u}(n)\end{bmatrix}$, and let $\boldsymbol{A} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^{\dagger}$. Then, it's clear that the eigenvalues of $\boldsymbol{R}(n)$ are given by $\lambda_k(\boldsymbol{R}(n)) = n\lambda_k(A) + \delta$. Hence, the eigenvalue density of the matrix $\boldsymbol{R}(n)$ can be approximated, to a high degree of accuracy, by:

$$\mu_{\boldsymbol{R}(n)}(x) \approx \mu_{\boldsymbol{\Phi}}(x) = \frac{1}{n}\mu_{\boldsymbol{A}}\left(\frac{x-\delta}{n}\right) \tag{12}$$

if m, n are large but $m/n = c$ is a constant. This follows from the Marcenko-Pastur Law (and some simple algebra).

Once we have the eigenvalue density, finding the moments is easy- we just "assume" that m is large, and use that, for some $m \times m$ random matrix, $\boldsymbol{B}$:

$$\lim_{m\to\infty} \frac{1}{m}\mathbb{E}[\mathrm{Tr}(\boldsymbol{B}^k)] = \int t^k \mu_{\boldsymbol{B}}(t)\,\mathrm{d}t \tag{13}$$

So, we can compute the moments as

$$M_k(m,n) \approx \int t^{-k}\mu_{\boldsymbol{\Phi}}(t)\,\mathrm{d}t \tag{14}$$

Plugging in yields the following (slightly ugly) expressions for the moments:

$$M_1(m,n) = \frac{\sqrt{\delta^2 + (m-n)^2 + 2\delta(m+n)} - |n-m| - \delta}{2\delta m} \tag{15a}$$

$$\begin{aligned} M_2(m,n) = {} & \frac{(m-n)^2 + \delta(m+n)}{2\delta^2 m\sqrt{\delta^2 + (m-n)^2 + 2\delta(m+n)}} \\ & - \frac{|n-m|\sqrt{\delta^2 + (m-n)^2 + 2\delta(m+n)}}{2\delta^2 m\sqrt{\delta^2 + (m-n)^2 + 2\delta(m+n)}} \end{aligned} \tag{15b}$$

which can be plugged back into (10) and (11) to get the relevant results.

4.1.2 Correlated Input Process

This is usually a mess of a problem. However, it can be simplified considerably using some random matrix theory results, again. We use the $n > m$ (although not significantly more than m) case. In this case, the diagonal loading is essentially irrelevant, so $\delta \approx 0$. Then, all we care about is $M_1(n, m)$.

Now we can use Steiltjes' transform. In this case, define $\boldsymbol{A} = \frac{1}{n}\boldsymbol{T}^{1/2}\boldsymbol{X}\boldsymbol{X}^{\dagger}\boldsymbol{T}^{1/2}$. We further assume that we allow $m, n \to \infty$ but ${}^{m}/_{n} = c < 1$ is a constant.Then, it can be shown that the Steiltjes' transform satisfies:

$$S_A(z) = \int \frac{\mu_T(x)dx}{(1 - c - czS_A(z))\, x - z} \tag{16}$$

Further, we have that:

$$S_A(z) = \lim_{m\to\infty} \frac{1}{m}\mathbb{E}\left[\mathrm{Tr}\left((\boldsymbol{A} - z\boldsymbol{I})^{-1}\right)\right] \tag{17}$$

Using these and (12) (which still applies, as we've made the approximations needed for Marcenko Pastur Law), we may see that

$$M_1(n, m) = \lim_{z\to 0^-} S_{\boldsymbol{R}(n)}(z) \approx \lim_{z\to 0^-} S_{\boldsymbol{\Phi}}(z) \tag{18}$$

which can then be evaluated using (16). Doing that and taking the limit as $z \to 0^-$ yields:

$$S_{\boldsymbol{\Phi}}(0^-) = \frac{1}{mn(1-c)} \sum_{k=1}^{m} \frac{1}{\lambda_k} \tag{19}$$

where the λ_ks are the eigenvalues of the input process. Finally, plugging this back into (10) and (11) gives:

$$\mathbb{E}\left[\|\epsilon(n)\|_2^2\right] = \frac{\sigma_v^2}{n-m} \sum_{k=1}^{m} \frac{1}{\lambda_k} \tag{20}$$

and

$$\mathbb{E}\left[\|e(n)\|_2^2\right] = \left[1 - \frac{1}{n-m} \sum_{k=1}^{n} \frac{1}{\lambda_k}\right] \sigma_v^2. \tag{21}$$

Note that these are for the special case in which $n > m$, $\frac{m}{n} = c$, $\delta = 0$, which is why we end up with this nice simple form.

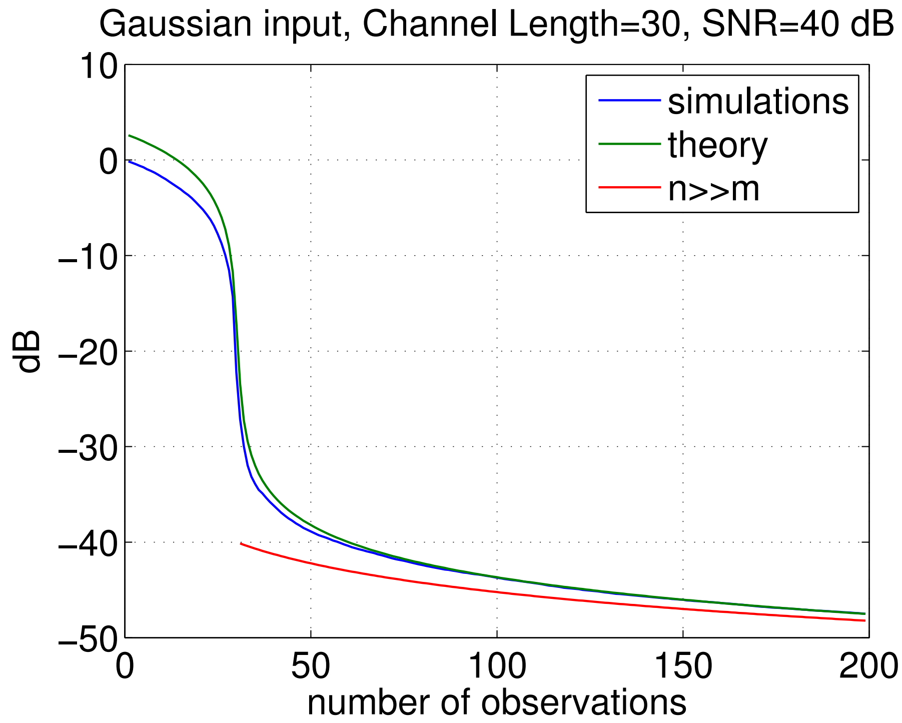

Figure 1: Channel Estimation MSE vs Number of Observations

5 Results

How well do our predictions above do and what do they tell us? In order to answer these questions, some Figures from the original paper have been reproduced here (with permission of the authors). These plots are for $m = 30$, with the noise covariance chosen to match the SNR of each plot.

First, consider the case of independent input vectors. 2 prediction cases are considered- where $\boldsymbol{u}(n)$ are IID Gaussian random vectors, and where they are drawn uniformly from $-1, 1^m$. The plots of the channel estimation error are shown in Figures 1 and 2, respectively. The signal prediction error follows similar trends and is not included.

The figures, first of all, show that the RMT based predictor tracks the performance *much* better than the conventional error expressions (8). That's the first reassuring result. However, in Figure 2, there is a rather strange effect, where the performance deteriorates with as n increases when $n < m$, reaches its worst point at $n = m$ and then gets better again! This is

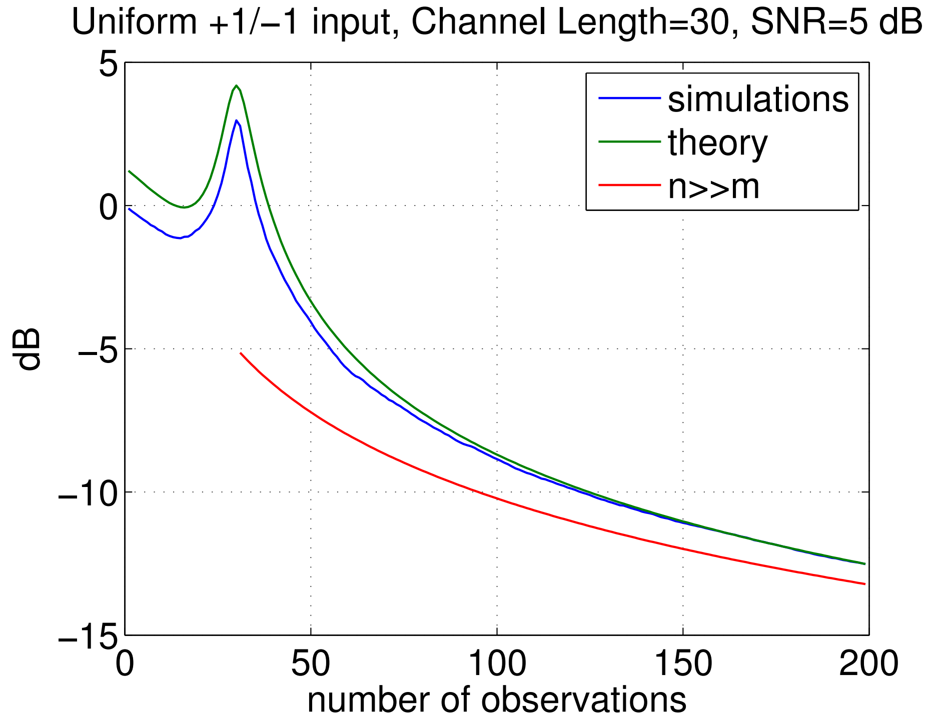

Figure 2: Channel Estimation MSE vs Number of Observations

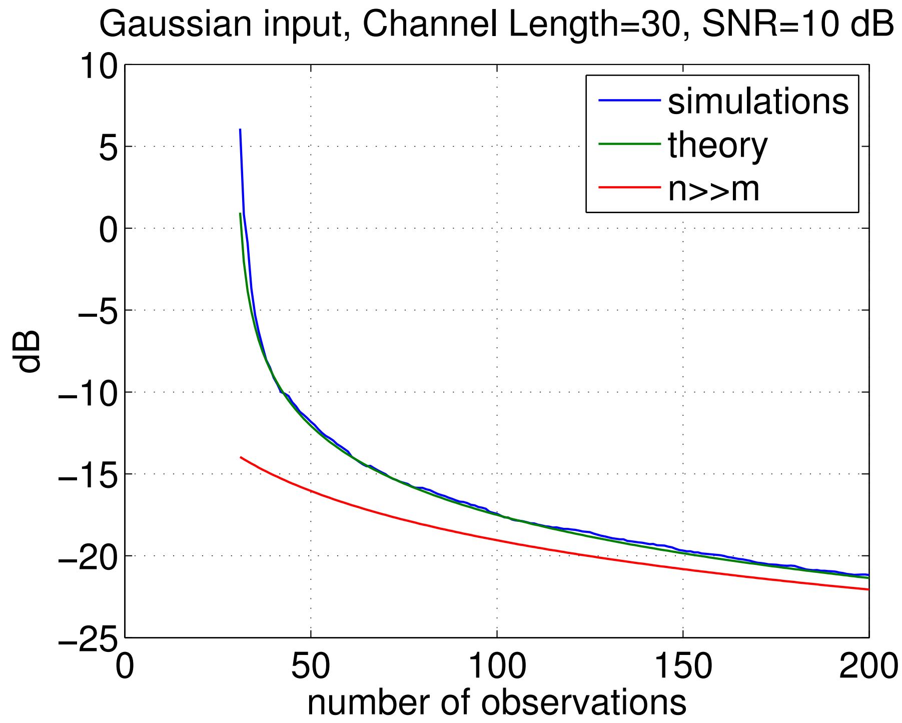

Figure 3: LTI Channel Estimation Error for Correlated Input Process

unexpected- we expect performance to improve with increasing n, but as it's predicted and backed up by simulation, perhaps there is another effect at play. We will come back to this issue.

Before doing so, however, consider the case of the correlated inputs. Figure 3 has the corresponding plot, which again reinforce the point that RMT is good at making predictions about the algorithm performance.

5.1 So What's the Bump?

We now turn to understanding the performance degradation effect when $n \approx m$. First, note that the Least Squares equation can be written in the form:

$$\hat{\boldsymbol{w}}(n) = \hat{\boldsymbol{w}}(n-1) + \boldsymbol{R}^{-1}(n)\boldsymbol{u}(n)(d(n) - \hat{\boldsymbol{w}}^{\dagger}(n-1)\boldsymbol{u}(n))^{*} \tag{22}$$

This structure is what leads to the widely used Recursive Least Squares algorithm.

Define $\boldsymbol{k}(n) = \boldsymbol{R}^{-1}(n)\boldsymbol{u}(n)$. For *this section*, redefine λ_k and $\boldsymbol{q}_k$ to be the eigenvalues and eigenvectors of $\boldsymbol{R}(n)$ (note that this is not what we have been using so far!) It turns out that the $\mathcal{L}_2$ norm of $\boldsymbol{k}(n)$ is what is important when trying to understand the performance degradation. This is because $\boldsymbol{k}(n)$ is what multiplies the error, so if it has a large norm, it will also blow up the noise.

We can write the $\mathcal{L}_2$ norm of $\boldsymbol{k}(n)$ as:

$$\|\mathbf{k}(n)\|_2^2 = \sum_{k=1}^{\min(m,n)} \left| \frac{\boldsymbol{q}_k^\dagger \boldsymbol{u}(n)}{\lambda_k} \right|^2 \tag{23}$$

For $n < m$, the smallest $n - m$ eigenvalues of $\boldsymbol{R}(n)$ are δ, and their eigenvectors are orthogonal to the space spanned by $\boldsymbol{u}(n)$. So the sum only goes to $\min(m, n)$ to account for these “trivial” eigenvalues.

The sum in (23) is dominated by the smallest of the non-trivial eigenvalues. Now suppose the observation vectors are drawn IID, uniformly from the unit sphere. Thus, statistically, each one contributes an equal amount of energy into each of the non-trivial directions spanned by the eigenspace of $\boldsymbol{R}(n)$. However, as $n < m$, a new non-trivial direction is obtained with each observation. The energy along that direction is about $\frac{1}{n}$ of the total energy, so it get smaller and smaller as n approaches m. Hence the minimum eigenvalue gets *smaller*! Once $n > m$, each successive snapshot contributes equally in all directions and the minimum eigenvalue grows. This correspondingly increases (or decreases) the norm of $\boldsymbol{k}(n)$, and a large norm leads to greater error. This explains the performance degradation around $n = m$.

Indeed the paper also uses a rather more technical method involving the moments again to analyze this effect in some more detail. Those details are omitted here.

6 The Upshot

An interesting and important problem in communication - analyzing the performance of the Least Squares Channel Estimator- has been considered using RMT. This significantly improves over classical analysis, and leads to the discovery of a somewhat counterintuitive result- that the performance of such estimators actually degrades as n increases when $n < m$ and the worst performance happens when $n \approx m$. This effect was also analyzed in the paper.

What does this imply about other problems? Clearly, the techniques- at least the basic ones- needed to analyze signal processing algorithms are all there in random matrix theory. This paper makes use of perhaps some of the crudest of these, but still ends up with excellent results. Indeed, such results are sufficient for many applications. We have many more techniques available, of course, and as the algorithms we start to work with get more and more complex, these techniques can get more and more sophisticated. An excellent overview of these is to be found in [6]. Nonetheless, simple though the example considered here may be, it gets at all the basic issues involved in these kinds of analyses- the small number of observations, how to handle correlated observations, which sort of techniques can be useful, and so on. It can serve as an introduction, and even perhaps a good example, to a rich field.

References

[1] A. W. Van der Vaart, *Asymptotic statistics.* Cambridge university press, 2000, vol. 3.

[2] R. Menon, P. Gerstoft, and W. S. Hodgkiss, "Asymptotic eigenvalue density of noise covariance matrices," *Signal Processing, IEEE Transactions on*, vol. 60, no. 7, pp. 3415–3424, 2012.

[3] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud, "Random matrix theory and financial correlations," *International Journal of Theoretical and Applied Finance*, vol. 3, no. 03, pp. 391–397, 2000.

[4] F. Luo, J. Zhong, Y. Yang, R. H. Scheuermann, and J. Zhou, "Application of random matrix theory to biological networks," *Physics Letters A*, vol. 357, no. 6, pp. 420–423, 2006.

[5] X. Mestre and M. Lagunas, "Modified subspace algorithms for doa estimation with large arrays," *Signal Processing, IEEE Transactions on*, vol. 56, no. 2, pp. 598–614, 2008.

[6] R. Couillet and M. Debbah, "Signal processing in large systems: A new paradigm," *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 24–39, 2013.

[7] S. Haykin, *Adaptive filter theory, 4th ed.* Prentice-Hall.

[8] M. Pajovic and J. C. Preisig, "Performance analysis of the least squares based lti channel identification algorithm using random matrix methods," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on.* IEEE, 2011, pp. 516–523.