ECE 6123
Advanced Signal Processing
The SVD and its SP Application

Peter Willett

Fall 2017

1 Least Squares Formulation of Wiener Filtering

1.1 The Equations

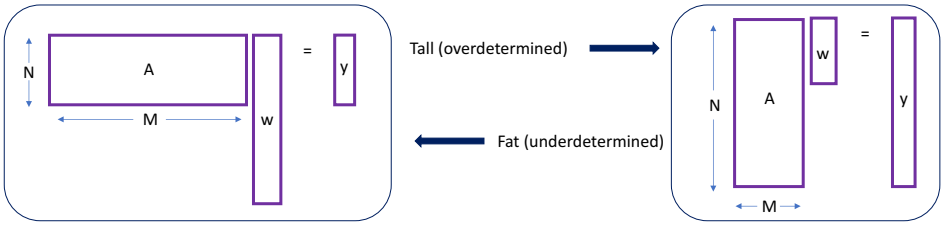
Let's suppose we arrange the data into a matrix:

$$\begin{pmatrix} \longleftarrow & \mathbf{u}_1^H & \longrightarrow \\ \longleftarrow & \mathbf{u}_2^H & \longrightarrow \\ & \vdots & \\ \longleftarrow & \mathbf{u}_2^H & \longrightarrow \end{pmatrix} \begin{pmatrix} w[1] \\ w[2] \\ \vdots \\ w[M] \end{pmatrix} \equiv \mathbf{A}\mathbf{w} = \mathbf{y}^* \equiv \begin{pmatrix} y[1]^* \\ y[2]^* \\ \vdots \\ y[N]^* \end{pmatrix} \quad (1)$$

The Wiener goal is actually the least-squares goal: choose $\mathbf{w}$ to minimize the error

$$J(\mathbf{w}) \equiv ||\mathbf{y} - \mathbf{d}||^2 = \sum_{n=1}^{N} |e[n]|^2 = \sum_{n=1}^{N} |d[n] - y[n]|^2 \quad (2)$$

A lot depends on whether the matrix $\mathbf{A}$ is short and fat or tall and skinny. In the short / fat case the linear system (1) is underdetermined, meaning there are more variables in $\mathbf{w}$ than there are equations to match $\mathbf{y}$ to $\mathbf{d}$.That means that we can make $J(\mathbf{w}) = 0$ with multiple $\mathbf{w}$'s – which one should we choose? In the tall / skinny case (1) is likewise **over**determined, meaning that in any nontrivial case we cannot find $\mathbf{w}$ such that $J(\mathbf{w}) = 0$ – and in that case it makes sense to find the minimizing $\mathbf{w}$. The situations are illustrated below.



If $N = M$ and there is no triviality (linear dependence in columns of $\mathbf{A}$) then we have a unique solution – this is the least interesting case and we will ignore it from now on.

1.2 The Overdetermined Case

Presumably this is familiar. To minimize (2) we apply the *p.o.o.* and see that optimally

$$\mathbf{A}^H (\mathbf{d} - \mathbf{A}\mathbf{w}) \;=\; 0 \tag{3}$$

or

$$\mathbf{w} \;=\; (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{d} \tag{4}$$

unless $(\mathbf{A}^H\mathbf{A})$ is singular. We could write

$$\mathbf{A}^H\mathbf{A} \;=\; \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \begin{pmatrix} \longleftarrow & \mathbf{u}_1^H & \longrightarrow \\ \longleftarrow & \mathbf{u}_2^H & \longrightarrow \\ & \vdots & \\ \longleftarrow & \mathbf{u}_2^H & \longrightarrow \end{pmatrix} \tag{5}$$

$$\;=\; \sum_{n=1}^{N} \mathbf{u}\mathbf{u}^H \tag{6}$$

$$\;=\; N\hat{\mathbf{R}} \tag{7}$$

where the last equation assumes that the covariance matrix $\mathbf{R}$ is estimated by simple averaging. In a similar way we could write

$$\mathbf{A}^H\mathbf{d} \;=\; \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \begin{pmatrix} d[1]^* \\ d[2]^* \\ \vdots \\ d[N]^* \end{pmatrix} \tag{8}$$

$$\;=\; \sum_{n=1}^{N} \mathbf{u}d[n]^* \tag{9}$$

$$\;=\; N\hat{\mathbf{p}} \tag{10}$$

where again the last equation assumes that the cross-correlation vector $\mathbf{p}$ is estimated by simple averaging. Written in this way we have optimally

$$\mathbf{w} \;=\; (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{d} \;=\; (N\hat{\mathbf{R}})^{-1}(N\hat{\mathbf{p}}) \;=\; \hat{\mathbf{R}}^{-1}\hat{\mathbf{p}} \tag{11}$$

meaning that the solution we get by direct dumb least-squares is identical to the Wiener solution with block averaging estimates for the covariances.

2 The Singular Value Decomposition

2.1 Relationship to Eigendecompositions

Let us assume a matrix $\mathbf{A}$ whose dimension is N (rows) by M (columns): $N \times M$. Unless $M = N$ we have no eigendecomposition. But suppose we form left and right products (which are square and of respective dimensions $M \times M$ and $N \times N$. Now eigenstuff is available:

$$(\mathbf{A}^H\mathbf{A})\mathbf{V} = \mathbf{V}\boldsymbol{\Gamma} \quad (12)$$

$$(\mathbf{A}\mathbf{A}^H)\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda} \quad (13)$$

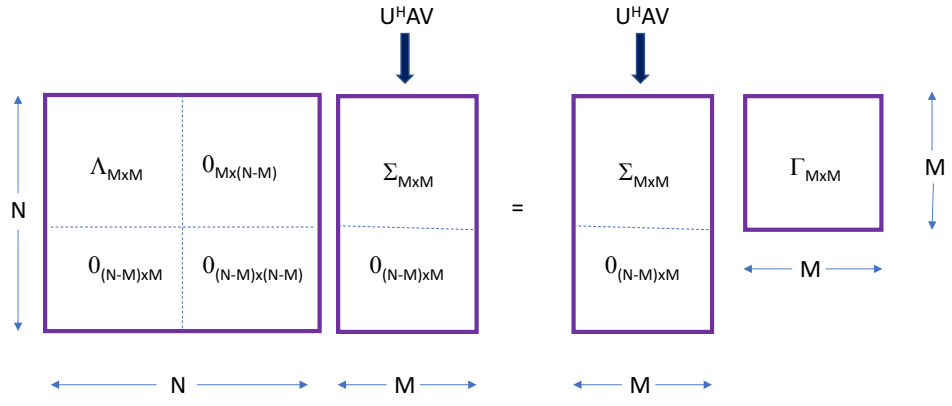
where $\mathbf{V}$ is unitary (means: $\mathbf{V}^H\mathbf{V} = \mathbf{I}$) and of dimension $M \times M$; and likewise $\mathbf{U}$ too is unitary ($\mathbf{U}^H\mathbf{U} = \mathbf{I}$) and of dimension $N \times N$. The matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are diagonal with nonnegative elements. Since the rank of $\mathbf{A}$ is $\min\{M, N\}$ this is also the rank of $(\mathbf{A}^H\mathbf{A})$ and $(\mathbf{A}\mathbf{A})^H$. Hence in the short / fat case $N < M$ there are[1] $M - N$ zeros on the diagonal of $\boldsymbol{\Gamma}$; and likewise in the tall / skinny case $N > M$ there are $N - M$ zeros on the diagonal of $\boldsymbol{\Lambda}$.

What is interesting is to form the identity

$$\mathbf{U}^H\mathbf{A}\mathbf{A}^H\mathbf{A}\mathbf{V} = \mathbf{U}^H\mathbf{A}\mathbf{A}^H\mathbf{A}\mathbf{V} \quad (14)$$

$$\boldsymbol{\Lambda}\mathbf{U}^H\mathbf{A}\mathbf{V} = \mathbf{U}^H\mathbf{A}\mathbf{V}\boldsymbol{\Gamma} \quad (15)$$

where to get (15) we've substituted (13) on the LHS and (12) on the RHS. The situation is as illustrated below.



[1] There could be more zeros if $\mathbf{A}$ is rank-deficient, meaning that some $\mathbf{u}_n$'s are linearly dependent; but this is a trivial case and would be dilatory to explore.

In the above figure we've assumed for concreteness that $N > M$; there is no loss of generality in doing that in this section. Note that we have inserted the fact that the last $N - M$ rows of $\mathbf{U}^H\mathbf{AV}$ have to be zero: the LHS tells us that it must be so. We've also (slightly) changed notation to denote only the northwest $M \times M$ block of the premultiplying matrix on the LHS to be $\mathbf{\Lambda}$. Now, we can also write

$$\mathbf{\Lambda\Sigma} = \mathbf{\Sigma\Gamma} \tag{16}$$

and which implies that $\mathbf{\Sigma}$ is the (unnormalized) matrix of eigenvectors of $\mathbf{\Lambda}$ (or $\mathbf{\Gamma}$). Since $\mathbf{\Lambda}$ is a diagonal matrix we know that its eigenvectors are the Cartesian basis vectors: that is, $\mathbf{\Sigma}$ itself has to be diagonal.

And that's what we wanted to show. Now we know that we have

$$\mathbf{U}^H\mathbf{AV} = \begin{pmatrix} \mathbf{\Sigma} \\ \mathbf{0} \end{pmatrix} \tag{17}$$

$$\mathbf{A} = \mathbf{U}\begin{pmatrix} \mathbf{\Sigma} \\ \mathbf{0} \end{pmatrix}\mathbf{V}^H \tag{18}$$

in the case that $N > M$ and

$$\mathbf{U}^H\mathbf{AV} = \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \end{pmatrix} \tag{19}$$

$$\mathbf{A} = \mathbf{U}\begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \end{pmatrix}\mathbf{V}^H \tag{20}$$

in the case $N < M$. Equations (18) and (20) represent the singular value decomposition (SVD) of the matrix $\mathbf{A}$: the product of a unitary $N \times N$ matrix, a diagonal matrix of dimension $N \times M$ and another $M \times M$ unitary matrix. It's quite general. As will be seen very shortly the matrices can be computed via appropriate eigendecompositions; but there are ways to compute them directly that are far more efficient, especially if $N \gg M$ or $N \ll M$. The SVD is a primary tool in many signal processing tasks; we will soon see an example in the adaptive filtering venue, and then more helping us with spectral estimation.

Again for the case $N > M$ we can also explore

$$\mathbf{AA}^H = \mathbf{U}\begin{pmatrix} \mathbf{\Sigma} \\ \mathbf{0} \end{pmatrix}\mathbf{V}^H\mathbf{V}\begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{21}$$

$$= \mathbf{U}\begin{pmatrix} \mathbf{\Sigma}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{22}$$

and

$$\mathbf{A}^H\mathbf{A} = \mathbf{V}\begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \end{pmatrix}\mathbf{U}^H\mathbf{U}\begin{pmatrix} \mathbf{\Sigma} \\ \mathbf{0} \end{pmatrix}\mathbf{V}^H \tag{23}$$

$$= \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^H \tag{24}$$

meaning that $\boldsymbol{\Sigma}^2$ contains the eigenvalues of $(\mathbf{A}\mathbf{A}^H)$ (or $(\mathbf{A}^H\mathbf{A})$). For the case $N < M$ we have

$$\mathbf{A}\mathbf{A}^H = \mathbf{U}\begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \end{pmatrix}\mathbf{V}^H\mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma} \\ \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{25}$$

$$= \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^H \tag{26}$$

and

$$\mathbf{A}^H\mathbf{A} = \mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma} \\ \mathbf{0} \end{pmatrix}\mathbf{U}^H\mathbf{U}\begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \end{pmatrix}\mathbf{V}^H \tag{27}$$

$$= \mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{V}^H \tag{28}$$

which are the same as (22) and (24), just reversed due to the matrix size.

2.2 The Pseudo-Inverse

The pseudo-inverse, or Moore-Penrose inverse, is defined as

$$\mathbf{A}^\dagger \equiv \mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma}^{-1} & \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{29}$$

if $N > M$ or

$$\mathbf{A}^\dagger \equiv \mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma}^{-1} \\ \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{30}$$

if $N < M$. If some elements of $\boldsymbol{\Sigma}$ are zero the modification is obvious; and if $M = N$ (and $\mathbf{A}$ is full rank) it is easy to see that $\mathbf{A}^\dagger = \mathbf{A}^{-1}$. So what?

Let's begin with the case $N > M$. We have

$$\mathbf{A}^\dagger\mathbf{A} = \mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma}^{-1} & \mathbf{0} \end{pmatrix}\mathbf{U}^H\mathbf{U}\begin{pmatrix} \boldsymbol{\Sigma} \\ \mathbf{0} \end{pmatrix}\mathbf{V}^H \tag{31}$$

$$= \mathbf{V}\mathbf{I}\mathbf{V}^H \tag{32}$$

$$= \mathbf{I}_{M\times M} \tag{33}$$

Now let's examine the case $N < M$. We now have

$$\mathbf{A}\mathbf{A}^\dagger = \mathbf{U}\begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \end{pmatrix}\mathbf{V}^H\mathbf{V}\begin{pmatrix} \boldsymbol{\Sigma}^{-1} \\ \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{34}$$

$$= \mathbf{U}\begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \boldsymbol{\Sigma}^{-1} \\ \mathbf{0} \end{pmatrix}\mathbf{U}^H \tag{35}$$

$$= \mathbf{U}\mathbf{I}\mathbf{U}^H \tag{36}$$

$$= \mathbf{I}_{N\times N} \tag{37}$$

We will use these shortly.

2.3 The SVD and the Overdetermined Case

Here we have $N > M$, the tall / skinny situation. From (33) we write

$$\begin{aligned}
\mathbf{w} &= \mathbf{A}^\dagger \mathbf{d} && (38)\\
&= \mathbf{V} \begin{pmatrix} \mathbf{\Sigma}^{-1} & \mathbf{0} \end{pmatrix} \mathbf{U}^H \mathbf{d} && (39)\\
&= \mathbf{V} \begin{pmatrix} \mathbf{\Sigma}^{-1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{pmatrix} \mathbf{d} && (40)\\
&= \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}_1^H \mathbf{d} && (41)
\end{aligned}$$

For what it is worth, we could start with (4) and use the SVD to get

$$\begin{aligned}
\mathbf{w} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{d} && (42)\\
&= \left[\mathbf{V} \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \end{pmatrix} \mathbf{U}^H \mathbf{U} \begin{pmatrix} \mathbf{\Sigma} \\ \mathbf{0} \end{pmatrix} \mathbf{V}^H\right]^{-1} \mathbf{V} \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \end{pmatrix} \mathbf{U}^H \mathbf{d} && (43)\\
&= \mathbf{V} \begin{pmatrix} \mathbf{\Sigma}^{-1} & \mathbf{0} \end{pmatrix} \mathbf{U}^H \mathbf{d} && (44)\\
&= \mathbf{A}^\dagger \mathbf{d} && (45)\\
&= \mathbf{V} \begin{pmatrix} \mathbf{\Sigma}^{-1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{pmatrix} \mathbf{d} && (46)\\
&= \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}_1^H \mathbf{d} && (47)
\end{aligned}$$

The message: The SVD solves the overdetermined case.

2.4 The SVD and the Underdetermined Case

In the overdetermined case there is no solution to (1), so we found the solution to minimize the error[2]. In the underdetermined (short / fat $\mathbf{A}$) case there is a whole subspace of $\mathbf{w}$'s that solves (1) – which one should we choose? Unless there are other concerns, a good choice might be to select the $\mathbf{w}$ with minimum length. So we have the optimization problem

$$\text{Minimize } \mathbf{w}^H\mathbf{w} \text{ subject to } \mathbf{A}\mathbf{w} = \mathbf{d} \tag{48}$$

We use Lagrange multipliers, and find

$$\mathbf{w} - \mathbf{A}^H\lambda = 0 \tag{49}$$

[2] . . . or the residuals.

at optimality. Substituting for the constraint we have

$$\mathbf{A}\mathbf{A}^H\lambda = \mathbf{d} \tag{50}$$
$$\lambda = (\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{d} \tag{51}$$

hence

$$\mathbf{w} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{d} \tag{52}$$

Note that the matrix can be assumed in nontrivial cases to be nonsingular since $N < M$. Let us substitute for the SVD.

$$\mathbf{w} = \mathbf{A}^H(\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{d} \tag{53}$$
$$= \mathbf{V}\begin{pmatrix}\mathbf{\Sigma}\\ \mathbf{0}\end{pmatrix}\mathbf{U}^H\left[\mathbf{U}\begin{pmatrix}\mathbf{\Sigma} & \mathbf{0}\end{pmatrix}\mathbf{V}^H\mathbf{V}\begin{pmatrix}\mathbf{\Sigma}\\ \mathbf{0}\end{pmatrix}\mathbf{U}^H\right]^{-1}\mathbf{d} \tag{54}$$
$$= \mathbf{V}\begin{pmatrix}\mathbf{\Sigma}\\ \mathbf{0}\end{pmatrix}\mathbf{U}^H\left[\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^H\right]^{-1}\mathbf{d} \tag{55}$$
$$= \mathbf{V}\begin{pmatrix}\mathbf{\Sigma}^{-1}\\ \mathbf{0}\end{pmatrix}\mathbf{U}^H\mathbf{d} \tag{56}$$
$$= \mathbf{A}^\dagger\mathbf{d} \tag{57}$$
$$= \begin{pmatrix}\mathbf{V}_1\\ \mathbf{V}_2\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma}^{-1}\\ \mathbf{0}\end{pmatrix}\mathbf{U}^H\mathbf{d} \tag{58}$$
$$= \mathbf{V}_1\mathbf{\Sigma}^{-1}\mathbf{U}^H\mathbf{d} \tag{59}$$

The message is the same: use the SVD.

2.5 Summary: Applying the Pseudo-Inverse

From (45) and (57) it is clear that the SVD – specifically the pseudo-inverse – can be used to solve both the overdetermined and underdetermined cases: It is always a safe choice. Perhaps more important[3], even if the rows of a short / fat $\mathbf{A}$ or the columns of a tall / skinny $\mathbf{A}$ are linearly dependent, the pseudo-inverse works fine. The only significant difference is that some of the elements of $\mathbf{\Sigma}$ are zero, and when the pseudo-inverse is formed these remain zero when $\mathbf{\Sigma}^{-1}$ is formed. Note that (47) and (59) are not mathematically necessary to include, but computationally they can save effort.

[3]We haven't shown this here because it is messy and irritating, but it it trivial to do.

3 The Normalized LMS Adaptive Filter

This is a nice twist on the LMS that uses the theory we've learnt about the SVD. Suppose we want to make a change in $\mathbf{w}_n \rightarrow \mathbf{w}_{n+1}$ such that

$$\mathbf{w}_{n+1}^H \mathbf{u}_n \;=\; d[n] \tag{60}$$

meaning that the filter error *would* have been zero if the filter had been clairvoyant enough to see $\mathbf{u}_{n+1}$ before it happened. To some extent this seems like making a "rear-view mirror" change. However, the intuition seems solid: it would appear that the filter is moving in the right direction by such a move. Now the concern is that (60) is *too easy*: $\mathbf{w}_{n+1}$ is a vector with M elements, and we are offering only a rank-one constraint by(60).

Let us define

$$\delta_{n+1} \;\equiv\; \mathbf{w}_{n+1} \;-\; \mathbf{w}_n \tag{61}$$

Inserting this to (60) gives us

$$(\delta_{n+1} + \mathbf{w}_n)^H \mathbf{u}_n \;=\; d[n] \tag{62}$$

or

$$\delta_{n+1}^H \mathbf{u}_n \;=\; e[n] \tag{63}$$

$$\mathbf{u}_n^H \delta_{n+1} \;=\; e[n]^* \tag{64}$$

where $e[n]$ is the true (not clairvoyant) filter error. This (63) is really a restatement of (60), but it allows us to see that this is really an underdetermined system, albeit one that is *very* underdetermined down to $N = 1$. If we were to use the pseudo-inverse to "solve" (63) we would find the solution that minimizes $||\delta_{n+1}||$ – and this seems like a reasonable thing to do.

Applying the SVD we have according to (64) $\mathbf{u}_n^H$ taking the role "$\mathbf{A}$"; $N = 1$ and M is the length of the filter tap-weight vector. Since $\mathbf{U}$ and $\mathbf{V}$ are matrices of eigenvectors (unitary matrices, meaning both orthogonal and normalized) the SVD is

$$\mathbf{U} \;=\; 1 \text{ a scalar} \tag{65}$$

$$\mathbf{\Sigma} \;=\; \begin{pmatrix} ||\mathbf{u}_n|| & 0 & 0 & \dots & 0 \end{pmatrix} \text{ a row vector with } (M-1) \text{ zeros} \tag{66}$$

$$\mathbf{V} \;=\; \begin{pmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{pmatrix} \text{ where } \mathbf{V}_1 \text{ is a column-vector} \tag{67}$$

$$\mathbf{V}_1 \;=\; \frac{\mathbf{u}_n}{||\mathbf{u}_n||} \tag{68}$$

and to be clear: $||\mathbf{x}|| \equiv \sqrt{\mathbf{x}^H\mathbf{x}}$ defines the norm. Clearly $\mathbf{V}$ is $M \times M$; but only the first column is important. Applying the pseudo-inverse, then, we have from (59)

$$\begin{aligned} \delta_{n+1} &= \mathbf{V}_1\mathbf{\Sigma}^{-1}\mathbf{U}^H\mathbf{e}^* && (69) \\ &= \frac{\mathbf{u}_n}{||\mathbf{u}_n||^2}e[n]^* && (70) \end{aligned}$$

which means

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{1}{||\mathbf{u}_n||^2}\mathbf{u}_n e[n]^* \quad (71)$$

which for obvious reasons is called the *normalized* LMS (NLMS) update. Notice that (71) is very much like the usual LMS filter update, except that μ is replaced by $\frac{1}{||\mathbf{u}_n||^2}$. One can see that the NLMS update is in a sense more robust than LMS: a large $\mathbf{u}_n$ can force the LMS tap-weight vector $\mathbf{w}_{n+1}$ to make a large step. If that large $\mathbf{u}_n$ were really just an outlying sample (something non-Gaussian, say) then it is doubly-harmful to LMS: both $\mathbf{u}_n$ and $e[n]$ will be large. On the other hand, NLMS de-weights large $\mathbf{u}_n$'s, and that is in a practical sense quite appealing. It is also appealing that there is no need to study convergence to make suggestions for μ, as we had to do with LMS: the step-size is given. The text devotes much time to convergence nonetheless, and that is useful if inserted to a real application.

A concern that is raised in the text actually relates to the opposite of the robustness issues: what happens when $\mathbf{u}_n$ is very small? It is easy to see that the update then can be large. The proposal is rather a bandage:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \left(\frac{\tilde{\mu}}{\delta + ||\mathbf{u}_n||^2}\right)\mathbf{u}_n e[n]^* \quad (72)$$

The result is a far less beautiful algorithm. But it is probably quite practical.

4 Low-Rank Matrix Approximation

The Frobenius norm for a matrix is a logical extension of the vector L_2-norm to matrices:

$$||A||_F^2 \equiv \sum_{n=1}^{N}\sum_{m=1}^{M}|A_{n,m}|^2 \quad (73)$$

meaning that it is the sum of (magnitude-) squares of all the elements. An equivalent way to express the Frobenius norm is

$$||A||_F^2 = Tr\left(\mathbf{A}^H\mathbf{A}\right) \quad (74)$$

If we apply the SVD of $\mathbf{A}$ we get

$$\begin{aligned} ||A||_F^2 &= Tr\left(\mathbf{V}\mathbf{\Sigma}\mathbf{U}^H\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H\right) && (75)\\ &= Tr\left(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^H\right) && (76)\\ &= Tr\left(\mathbf{V}^H\mathbf{V}\mathbf{\Sigma}^2\right) && (77)\\ &= Tr\left(\mathbf{\Sigma}^2\right) && (78)\\ &= \sum_{i=1}^{\min\{M,N\}} \sigma_i^2(\mathbf{A}) && (79) \end{aligned}$$

that is, the Frobenius norm is the sum of squares of the singular values.

The low-rank approximation problem is to find $\hat{\mathbf{A}}_o$ to minimize

$$\hat{\mathbf{A}}_o = \arg\min_{\hat{\mathbf{A}}}\{||\mathbf{A}-\hat{\mathbf{A}}||_F^2\} \quad (80)$$

with the constraint that the rank of $\hat{\mathbf{A}}_o$ is $R < \min\{M,N\}$. Let us assume that the the singular values of $\mathbf{A}$ have been ordered such that we have

$$\sigma_1^2(\mathbf{A}) \geq \sigma_2^2(\mathbf{A}) \geq \ldots \geq \sigma_{\min\{M,N\}}^2(\mathbf{A}) \quad (81)$$

whence it is relatively easy to see that the solution is

$$\hat{\mathbf{A}}_o = \sum_{i=1}^{R} \sigma_i(\mathbf{A})\mathbf{u}_i\mathbf{v}_i^H \quad (82)$$

where

$$||\mathbf{A}-\hat{\mathbf{A}}_o||_F^2 = \sum_{i=R+1}^{\min\{M,N\}} \sigma_i^2(\mathbf{A}) \quad (83)$$

That is, just choose $\hat{\mathbf{A}}_o$ to align with the space corresponding to the R largest singular values of $\mathbf{A}$.

To see this, suppose that $R = 1$. We have that $\hat{\mathbf{A}} = \alpha\mathbf{b}\mathbf{c}^H$ where $\mathbf{b}$ and $\mathbf{c}$ are unit length. Now write

$$\begin{aligned} ||\mathbf{A}-\hat{\mathbf{A}}||_F^2 &= Tr\left((\mathbf{A}-\alpha\mathbf{b}\mathbf{c}^H)^H(\mathbf{A}-\alpha\mathbf{b}\mathbf{c}^H)\right) && (84)\\ &= Tr\left(\mathbf{A}\mathbf{A}^H\right) - 2\Re\left\{\alpha Tr\left(\mathbf{A}^H\mathbf{b}\mathbf{c}^H\right)\right\} \\ &\qquad + |\alpha|^2 Tr\left(\mathbf{b}\mathbf{c}^H\mathbf{c}\mathbf{b}^H\right) && (85)\\ &= Tr\left(\mathbf{A}\mathbf{A}^H\right) - 2\Re\left\{\alpha Tr\left(\mathbf{A}^H\mathbf{b}\mathbf{c}^H\right)\right\} \end{aligned}$$

$$+ \ |\alpha|^2 Tr\left(\mathbf{b}^H\mathbf{b}\mathbf{c}^H\mathbf{c}\right) \tag{86}$$

$$= \ Tr\left(\mathbf{A}\mathbf{A}^H\right) \ - \ 2\Re\left\{\alpha Tr\left(\mathbf{V}\mathbf{\Sigma}\mathbf{U}^H\mathbf{b}\mathbf{c}^H\right)\right\} \ + \ |\alpha|^2 \tag{87}$$

$$= \ \sum_{i=1}^{\min\{M,N\}} \sigma_i^2(\mathbf{A}) \ - \ 2\Re\left\{\alpha Tr\left(\mathbf{c}^H\mathbf{V}\mathbf{\Sigma}\mathbf{U}^H\mathbf{b}\right)\right\} + \ |\alpha|^2 \tag{88}$$

Neither $\mathbf{c}^H\mathbf{V}$ nor $\mathbf{U}^H\mathbf{b}$ can be larger than unity in magnitude. They are maximized when $\mathbf{c}$ and $\mathbf{b}$ are aligned to columns in $\mathbf{V}$ and $\mathbf{U}$, respectively. And the middle term is maximized when aligned to the maximum singular value, yielding

$$||\mathbf{A} - \hat{\mathbf{A}}||_F^2 \ = \ ||\mathbf{A}||_F^2 \ - \ \sigma_1^2(\mathbf{A}) \tag{89}$$

We can continue the process with succeeding rank-one matrices to ascertain (82) and (83).