Modern Digital Communications: A Hands-On Approach

LMMSE Estimation

Dr. Nicolae Chiurtu

- Course material of Prof. Bixio Rimoldi -

Last revision: Dec. 12, 2023

Linear Minimum Mean Squared Error Estimation

In this note, we study the concept of Linear Minimum Mean Squared Error (LMMSE) estimation. We derive the LMMSE estimator via the orthogonality principle.

Setup

Let $\boldsymbol{X} \in \mathbb{C}^m$ and $\boldsymbol{Y} \in \mathbb{C}^n$ be random vectors. We want to estimate $\boldsymbol{X}$ from $\boldsymbol{Y}$.

We are interested in the estimator $\hat{\boldsymbol{X}} : \mathbb{C}^n \to \mathbb{C}^m$ that minimizes

$$E\big[\|\boldsymbol{X} - \hat{\boldsymbol{X}}(\boldsymbol{Y})\|^2\big]. \tag{1}$$

As we have seen, without additional constraints, the solution is the MMSE estimator

$$\hat{\boldsymbol{X}}_{\mathrm{MMSE}}(\boldsymbol{y}) = E[\boldsymbol{X}|\boldsymbol{Y} = \boldsymbol{y}].$$

Finding an expression for $\hat{\boldsymbol{X}}_{\mathrm{MMSE}}(\boldsymbol{y})$ when $\boldsymbol{X}$ and $\boldsymbol{Y}$ are not jointly Gaussian is often a non-trivial task.

We can trade complexity for performance by settling on the LMMSE estimator, namely the one that minimizes

$$E\left[\|\boldsymbol{X} - \hat{\boldsymbol{X}}(\boldsymbol{Y})\|^2\right] \tag{2}$$

among all estimators of the form

$$\hat{\boldsymbol{X}}(\boldsymbol{y}) = A\boldsymbol{y} + \boldsymbol{b},$$

where $A \in \mathbb{C}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{C}^m$.

We begin by assuming that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are zero-mean. Later on we drop this assumption.

The Orthogonality Principle

Let $\boldsymbol{X} \in \mathbb{C}^m$ and $\boldsymbol{Y} \in \mathbb{C}^n$ be zero-mean random vectors. For the LMMSE estimator $\hat{\boldsymbol{X}}_{\mathrm{LMMSE}}(\boldsymbol{Y})$, the estimation error $\boldsymbol{X} - \hat{\boldsymbol{X}}_{\mathrm{LMMSE}}(\boldsymbol{Y})$ is orthogonal to the data:

$$E\Big[\big(\boldsymbol{X} - \hat{\boldsymbol{X}}_{\mathrm{LMMSE}}(\boldsymbol{Y})\big)\boldsymbol{Y}^\dagger\Big] = \boldsymbol{0}. \tag{3}$$

Note that we are assuming column vectors. Hence both sides of the above equality are $m \times n$ matrices.

Proof of the Orthogonality Principle

Suppose that the orthogonality principle holds for the estimator

$$\hat{\boldsymbol{X}}_A(\boldsymbol{y}) = A\boldsymbol{y},$$

and let $\hat{\boldsymbol{X}}_B(\boldsymbol{y}) = B\boldsymbol{y}$ be an alternative estimator.

$$\begin{aligned}
E\big[\|\boldsymbol{X} - B\boldsymbol{Y}\|^2\big] &= E\big[\|(\boldsymbol{X} - A\boldsymbol{Y}) + (A\boldsymbol{Y} - B\boldsymbol{Y})\|^2\big] \\
&= E\big[\|\boldsymbol{X} - A\boldsymbol{Y}\|^2\big] + E\big[\|A\boldsymbol{Y} - B\boldsymbol{Y}\|^2\big] \\
&\quad + 2\Re E\big[(\boldsymbol{X} - A\boldsymbol{Y})^\dagger (A\boldsymbol{Y} - B\boldsymbol{Y})\big] \\
&= E\big[\|\boldsymbol{X} - A\boldsymbol{Y}\|^2\big] + E\big[\|A\boldsymbol{Y} - B\boldsymbol{Y}\|^2\big] \\
&\geq E\big[\|\boldsymbol{X} - A\boldsymbol{Y}\|^2\big],
\end{aligned}$$

where in the third equality we used the assumption on $A\boldsymbol{Y}$ to obtain

$$\begin{aligned}
\Re E\big[(\boldsymbol{X}-A\boldsymbol{Y})^\dagger(A\boldsymbol{Y}-B\boldsymbol{Y})\big] &= \operatorname{trace}\Re E\big[(\boldsymbol{X}-A\boldsymbol{Y})(A\boldsymbol{Y}-B\boldsymbol{Y})^\dagger\big] \\
&= \operatorname{trace}\Re\Big[\underbrace{E\big[(\boldsymbol{X}-A\boldsymbol{Y})\boldsymbol{Y}^\dagger\big]}_{\mathbf{0}}(A-B)^\dagger\Big] \\
&= 0.
\end{aligned}$$

We conclude that the expected squared error of a linear estimator cannot be smaller than that of a linear estimator that satisfies the orthogonality principle (3).

The LMMSE Estimator

We still assume that $\boldsymbol{X} \in \mathbb{C}^m$ and $\boldsymbol{Y} \in \mathbb{C}^n$ are zero-mean random vectors.

We seek a matrix $A \in \mathbb{C}^{m \times n}$ s.t.

$$E\big[\|\boldsymbol{X} - A\boldsymbol{Y}\|^2\big] \tag{4}$$

is minimized.

By the orthogonality principle, A must satisfy

$$E\Big[\big(\boldsymbol{X} - A\boldsymbol{Y}\big)\boldsymbol{Y}^\dagger\Big] = 0,$$

i.e.,

$$E[\boldsymbol{X}\boldsymbol{Y}^\dagger] = AE[\boldsymbol{Y}\boldsymbol{Y}^\dagger],$$

or, changing notation,

$$K_{\boldsymbol{X}\boldsymbol{Y}} = AK_{\boldsymbol{Y}}.$$

Solving for A yields

$$A = K_{\boldsymbol{X}\boldsymbol{Y}}K_{\boldsymbol{Y}}^{-1}.$$

Hence

$$\hat{\boldsymbol{X}}_{\mathrm{LMMSE}}(\boldsymbol{y}) = A\boldsymbol{y} = K_{\boldsymbol{XY}} K_{\boldsymbol{Y}}^{-1} \boldsymbol{y}.$$

Note that we have already found this expression in deriving the MMSE estimator $\hat{\boldsymbol{X}}_{\mathrm{MMSE}}(\boldsymbol{y})$ for the case that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are jointly Gaussian random vectors.

In fact, for jointly Gaussian random vectors, the MMSE estimator is linear, which implies that it is identical to the LMMSE estimator.

Dropping the Zero-Mean Assumption

Lemma. *For a general random vector* $\boldsymbol{Z} \in \mathbb{C}^m$*, the constant vector* $\boldsymbol{b} \in \mathbb{C}^m$ *that minimizes*

$$E\left[\|\boldsymbol{Z} - \boldsymbol{b}\|^2\right]$$

is $\boldsymbol{b} = E[Z]$.

Proof: $\boldsymbol{b}$ can be seen as the MMSE estimate of $\boldsymbol{Z}$ based on an observation $\boldsymbol{W}$ which is a constant – hence independent of $\boldsymbol{Z}$. This estimate is

$$\boldsymbol{b} = E[\boldsymbol{Z}|\boldsymbol{W}] = E[\boldsymbol{Z}].$$

For an alternative proof we write

$$
\begin{aligned}
E\big[\|\boldsymbol{Z}-\boldsymbol{b}\|^2\big] &= E\big[\|\boldsymbol{Z}-E[\boldsymbol{Z}]+E[\boldsymbol{Z}]-\boldsymbol{b}\|^2\big] \\
&= E\big[\|\boldsymbol{Z}-E[\boldsymbol{Z}]\|^2\big] + E\big[\|E[\boldsymbol{Z}]-\boldsymbol{b}\|^2\big] \\
&\quad + 2\Re\ \underbrace{E\big[(\boldsymbol{Z}-E[\boldsymbol{Z}])^\dagger(E[\boldsymbol{Z}]-\boldsymbol{b})\big]}_{(E[\boldsymbol{Z}]-E[\boldsymbol{Z}])^\dagger(E[\boldsymbol{Z}]-\boldsymbol{b})=\mathbf{0}^\dagger(E[\boldsymbol{Z}]-\boldsymbol{b})=0} \\
&= E\big[\|\boldsymbol{Z}-E[\boldsymbol{Z}]\|^2\big] + E\big[\|E[\boldsymbol{Z}]-\boldsymbol{b}\|^2\big] \\
&\geq E\big[\|\boldsymbol{Z}-E[\boldsymbol{Z}]\|^2\big]
\end{aligned}
$$

□

Now consider general random vectors $\boldsymbol{X} \in \mathbb{C}^m$ and $\boldsymbol{Y} \in \mathbb{C}^n$, not necessarily zero-mean.

Define $\tilde{\boldsymbol{X}} = \boldsymbol{X} - E[\boldsymbol{X}]$ and $\tilde{\boldsymbol{Y}} = \boldsymbol{Y} - E[\boldsymbol{Y}]$, both zero-mean.

We are seeking the matrix B and vector $\boldsymbol{b}$ that minimize

$$E\big[\|\boldsymbol{X} - \underbrace{(B\boldsymbol{Y} + \boldsymbol{b})}_{\hat{\boldsymbol{X}}_{\mathrm{LMMSE}}(\boldsymbol{Y})}\|^2\big]. \tag{5}$$

From the above lemma, for a fixed B, the minimizing $\boldsymbol{b}$ is

$$\boldsymbol{b} = E[\boldsymbol{X} - B\boldsymbol{Y}] = E[\boldsymbol{X}] - BE[\boldsymbol{Y}].$$

With this $\boldsymbol{b}$, (5) becomes

$$E\big[\|\boldsymbol{X} - E[\boldsymbol{X}] - (B\boldsymbol{Y} - BE[\boldsymbol{Y}])\|^2\big] = E\big[\|\underbrace{\boldsymbol{X} - E[\boldsymbol{X}]}_{\tilde{\boldsymbol{X}}} - B\underbrace{(\boldsymbol{Y} - E[\boldsymbol{Y}])}_{\tilde{\boldsymbol{Y}}}\|^2\big].$$

The minimum is achieved for

$$B = K_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{Y}}} K_{\tilde{\boldsymbol{Y}}}^{-1} = K_{\boldsymbol{X}\boldsymbol{Y}} K_{\boldsymbol{Y}}^{-1}.$$

Summarizing,

$$\begin{aligned} \hat{\boldsymbol{X}}_{\text{LMMSE}}(\boldsymbol{Y}) &= B\boldsymbol{Y} + \boldsymbol{b} \\ &= B\boldsymbol{Y} - BE[\boldsymbol{Y}] + E[\boldsymbol{X}] \\ &= B(\boldsymbol{Y} - E[\boldsymbol{Y}]) + E[\boldsymbol{X}], \end{aligned} \tag{6}$$

with

$$B = K_{\boldsymbol{X}\boldsymbol{Y}} K_{\boldsymbol{Y}}^{-1}.$$

Notice that (6) is confirming what our intuition should suggest, namely that the LMMSE estimate of $\boldsymbol{X}$ based on $\boldsymbol{Y}$ is $E[\boldsymbol{X}]$ plus the LMMSE estimate of $\tilde{\boldsymbol{X}}$ based on $\tilde{\boldsymbol{Y}}$.

Example: LS vs LMMSE

It is instructive to compare the LS approximation and the LMMSE estimation by means of a simple example.

Let

$$Y = hX + Z$$

be the output of a channel, where X is the channel input symbol, h the channel strength known to the receiver, and Z the additive noise, assumed to be independent of X. For simplicity, we assume that all quantities are scalar and real-valued, and that X and Z are zero-mean.

Suppose that we are interested in an equalizer that tries to undo the effect of the channel by delivering an estimate of the channel input X based on the channel output Y.

The LS approximation is the $\hat{X}(y)$ that minimizes

$$|y - h\hat{X}(y)|^2.$$

In this particular example, the minimum is 0, achieved by setting

$$\hat{X}_{\mathrm{LS}}(y) = \frac{y}{h}.$$

The LMMSE estimate is the $\hat{X}(y)$ that minimizes

$$E|X - \hat{X}(y)|^2,$$

and the minimizer is

$$\hat{X}_{\mathrm{LMMSE}}(y) = \frac{K_{XY}}{K_Y} y = \frac{hP}{h^2P + \sigma^2} y,$$

where $P = E[|X|^2]$ is the signal's energy and $\sigma^2 = E[|Z|^2]$ is the noise variance.

It is important to notice the different objective between the two estimators:

- the LS estimator tries to find the input that "explains" the output with the smallest possible noise. For this scalar case the smallest noise is 0. Notice that there are infinitely many choices for $\hat{x}$ and $\hat{z}$ for which the observed y equals $h\hat{x} + \hat{z}$, and choosing $\hat{z} = 0$ seems arbitrary for the problem at hand.
- the LMMSE estimate better matches the objective of an equalizer, which is to find the input estimate that, in average "best matches" the actual input. To do so, it uses the statistic of the input and of the noise.

As h^2P becomes much larger than σ^2, the output of the LMMSE estimator tends to $\frac{y}{h}$, which is the output of the LS estimator. This is to be expected as $h^2P \gg \sigma^2$ means that the noise is negligible.

Notice that

$$\hat{X}_{\mathrm{LS}}(y) = \frac{y}{h} = x + \frac{z}{h},$$

which explains why we say that the LS estimator boosts the noise when the channel parameter h is small.

LS Approximation Revisited and MATLAB/Python commands

In a more general context, LS approximation is about model fitting in the following sense. We observe data $y_1, \ldots, y_n$ and we assume that the data comes from a parametric model. The objective of the LS approximation is to find the parameters so that the model "best" fits the observations.

For instance, in our communication application, the model is $\mathbf{y} = A\mathbf{x}$ for some given matrix A and unknown vector $\mathbf{x}$ which plays the role of the parameters to be determined. If A is an invertible square matrix, then $\hat{\mathbf{x}} = A^{-1}\mathbf{y}$ constitutes a perfect fit, but this is not the typical situation. In general, as we have seen, the LS approximation is $\hat{\mathbf{x}} = (A^\dagger A)^{-1}A^\dagger\mathbf{y}$.

In another example, the model could be $y_i = p(x_i)$, $i = 1, \ldots, n$, where $p(x) = b_1 + b_2 x + b_3 x^2$ is a polynomial of a specified degree (here of degree 2). The parameters here are b_1, b_2, and b_3. Both the x_i and the y_i, $i = 1, \ldots, n$, are given. Since in general there are more observations than parameters, and the observations are noisy, there is no perfect fit. Notice that we can write the model as

$$\mathbf{y} = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = A\mathbf{b}.$$

We have already encountered this situation in estimating the Doppler frequency of a GPS signal, but in that case the number of observations was the same as the number of parameters, and we could find a perfect fit.

When there is a linear relationship between the parameters and the observations, as in the above examples, we speak of Linear Least Squares, and the parameter vector that minimizes the norm of the error is $(A^\dagger A)^{-1} A^\dagger \mathbf{y}$. The matrix $(A^\dagger A)^{-1} A^\dagger$ is the Moore-Penrose pseudoinverse of A.

If A is a non-singular square matrix, then A^{-1} exists and

$$(A^\dagger A)^{-1}A^\dagger \mathbf{y} = A^{-1}(A^\dagger)^{-1}A^\dagger \mathbf{y} = A^{-1}\mathbf{y}.$$

We see that in this case the Moore-Penrose pseudoinverse of A is the same as the inverse of A. In this case, the LS approximation of $\mathbf{x}$ in the model $\mathbf{y} = A\mathbf{x} + \mathbf{z}$ is the exact solution to the equation $\mathbf{y} = A\mathbf{x}$.

The MATLAB command for the Moore-Penrose pseudoinverse of A is `pinv(A)`.
In Python, one can use `numpy.linalg.pinv(A)`.

When A is a rectangular matrix or a singular square matrix, then, as we know, `pinv(A)*y` is a vector $\hat{\mathbf{x}}$ such that $A\hat{\mathbf{x}}$ is as close as possible (in square norm) to $\mathbf{y}$.

The same is true for the MATLAB backslash command `A\y` but the two results are not necessarily the same when there are multiple solutions. In this case, the Moore-Penrose pseudoinverse leads to the minimum-norm $\hat{\mathbf{x}}$, whereas the backslash operator leads to the $\hat{\mathbf{x}}$ that has the fewest non-zero entries. (See the documentation on `pinv` for an example.)

In Python, when there are multiple solutions, the command `numpy.linalg.lstsq(A, y)` returns the minimum-norm $\hat{\mathbf{x}}$.

When A is a non-singular square matrix, then the inverse, the Moore-Penrose pseudoinverse the backslash operation (applied on $\mathbf{y}$) and `numpy.linalg.lstsq(A, y)` lead to the same unique solution $\mathbf{x}$ that satisfies $\mathbf{y} = A\mathbf{x}$.

Appendix A: The orthogonality Principle and The Projection Theorem

It it instructive to derive the orthogonality principle from the projection theorem.

Recall that a random variable X is a function

$$X : \Omega \to \mathbb{R} \text{ or } \mathbb{C}.$$

We know (e.g. from PDC) that functions can be seen as vectors of an inner-product space.

Let $\boldsymbol{X} = (X_1, \ldots, X_m)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ be random vectors and let $\mathcal{U}$ be the inner-product space spanned by $X_1, \ldots, X_m, Y_1, \ldots, Y_n$, with the inner product defined by

$$\langle X_i, Y_j \rangle = E[X_i Y_j^*].$$

(You should verify that this is indeed a valid inner product.)

In $\mathcal{U}$, the squared norm of X_i is

$$\|X_i\|^2 = \langle X_i, X_i \rangle = E[|X_i|^2].$$

Let $\hat{X}_i$ be the LMMSE of X_i based on $\boldsymbol{Y}$.

Let $\mathcal{V}$ be the subspace of $\mathcal{U}$ spanned by $Y_1, \ldots, Y_n$.

$\hat{X}_i$ is the element of $\mathcal{V}$ for which

$$|X_i - \hat{X}_i|^2$$

is minimized.

By the projection theorem (see e.g. PDC), $\hat{X}_i$ is the projection of X_i onto $\mathcal{V}$.
It has the property that

$$X_i - \hat{X}_i$$

is orthogonal to every element of $\mathcal{V}$, including $Y_1, \ldots, Y_n$.

Hence, for all i and j,

$$\underbrace{\langle X_i - \hat{X}_i, Y_j \rangle}_{E[(X_i - \hat{X}_i) Y_j^*]} = 0.$$

The above equality corresponds to the i, j component of the matrix equality

$$E[(\boldsymbol{X} - \hat{\boldsymbol{X}})\boldsymbol{Y}^\dagger] = 0,$$

which is precisely the orthogonality principle.