Lecture 4
Least Squares Estimation

- Least Squares Estimation
- Recursive Least Squares
- Curve Fitting

Contents

Motivation

- If the second-order statistics are known, the LMMSE estimator is given by $\hat{\mathbf{s}}_{\text{LMMSE}} = E[\mathbf{s}\mathbf{z}^T](E[\mathbf{z}\mathbf{z}^T])^{-1}\mathbf{z}$;
- In many applications, they aren't known
- Alternative approach is to estimate the coefficients from observed data
- Two possible approaches
    - Estimate required moments from available data and build an approximate LMMSE estimator
    - Build an estimator that minimizes some error functional calculated from the available data

LMMSE VS Least Squares

- Recall that LMMSE estimators are optimal in expectation across the ensemble of all stochastic processes with the same second order statistics
- Least squares estimators minimize the error on a given *block* of data
- No guarantees about optimality on other data sets or other stochastic processes
- If the process is ergodic, the LS estimator approaches the LMMSE estimator as the size of the data set grows.

Principle of Least Squares

- the performance criterion: the sum of squares
- requires a data set where both the inputs and desired responses are known
- the range of possible applications: data fitting, plant modeling for control (system identification), prediction, inverse modeling, interference cancellation
- regularization: Tikhonov regularization (ridge regression), Lasso method (application in compressed sensing)

Least squares problems

- $y(n) \in \mathbb{R}(n = 1, \ldots, N)$ is the target or desired response
- $h_k(n), k = 1, \ldots, M$ represents the inputs
- Assume $y(n) = \mathbf{x}^T \mathbf{h}(n) + v(n)$, in which $v(n)$ is the noise, $\mathbf{h}(n) = [h_1(n), \ldots, h_M(n)]^T, \mathbf{x} = [x_1, \ldots, x_M]^T$
- What we want to do is to estimate $\mathbf{x}$, say $\hat{\mathbf{x}}$
- Assume $\hat{y}(n) = \hat{\mathbf{x}}^T \mathbf{h}(n)$, the estimate $\hat{\mathbf{x}}$ is chosen such the predicted output approaches the measured output

Least squares problems

Estimation error:

$$e(n) = y(n) - \hat{y}(n) = y(n) - \hat{\mathbf{x}}^T \mathbf{h}(n)$$

Sum of squared errors:

$$E_e = \sum_{n=1}^{N} [e(n)]^2$$

Matrix Formulation

$$\begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} - \begin{bmatrix} h_1(1) & \cdots & h_M(1) \\ \vdots & \ddots & \vdots \\ h_1(N) & \cdots & h_M(N) \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_M \end{bmatrix}$$

that is

$$\mathbf{e} = \mathbf{y} - H\hat{\mathbf{x}}$$

What we want to minimize is the sum of squared errors

$$E_e = \mathbf{e}^T\mathbf{e} = \mathbf{y}^T\mathbf{y} - \hat{\mathbf{x}}^T H^T \mathbf{y} - \mathbf{y}^T H\hat{\mathbf{x}} + \hat{\mathbf{x}}^T H^T H\hat{\mathbf{x}}$$

Solving the optimization problem

- necessary condition

$$\frac{\partial E_e}{\partial \hat{\mathbf{x}}} = -\mathbf{y}^T H - \mathbf{y}^T H + 2\hat{\mathbf{x}}^T H^T H = 0,$$

  then we have

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T \mathbf{y}$$

- sufficient condition

$$\frac{\partial^2 E_e}{\partial \mathbf{x} \partial \mathbf{x}^T} = H^T H$$

  has to be positive definite.

Discussion on the rank of H

- Any solution $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ differ by a vector in the nullspace of H, i.e.,

$$H(\hat{\mathbf{x}}_2 - \hat{\mathbf{x}}_1) = 0$$

- $\hat{\mathbf{x}}_{\mathrm{ls}}$ is unique if H has full column rank, which is equivalent to the requirement that $H^T H$ be positive definite.

Unbiasedness

$$E[\hat{\mathbf{x}}] = (H^T H)^{-1} H^T E[y]$$

If the noise is zero-mean, then

$$E[\hat{\mathbf{x}}] = E[\mathbf{x}]$$

the LS estimator is unbiased.

Properties of the LS estimate

- Assumptions:
  - $\mathbf{v}$ is zero-mean white noise, $E(\mathbf{v}\mathbf{v}^T) = \sigma^2 I$
- Conclusion: LS estimate has the minimum mean square error among all the linear unbiased estimate of $\mathbf{x}$.

That is, if $\bar{\mathbf{x}} = L\mathbf{y}$ and $E(\bar{\mathbf{x}}) = E(\mathbf{x})$, we have

$$E\{[\mathbf{x} - \hat{\mathbf{x}}][\mathbf{x} - \hat{\mathbf{x}}]^T\} \preceq E\{[\mathbf{x} - \bar{\mathbf{x}}][\mathbf{x} - \bar{\mathbf{x}}]^T\}$$

Properties of the LS estimate

Sketch of proof:

- As

$$\mathbf{y} = H\mathbf{x} + \mathbf{v}, \quad \hat{\mathbf{x}} = (H^T H)^{-1} H^T \mathbf{y}$$

  we have

$$\begin{aligned} E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] &= (H^T H)^{-1} H^T E(\mathbf{v}\mathbf{v}^T) H (H^T H)^{-1} \\ &= (H^T H)^{-1} H^T \sigma^2 H (H^T H)^{-1} \\ &= \sigma^2 (H^T H)^{-1} \end{aligned}$$

- $LH = I$, prove that the matrix $LL^T - (H^T H)^{-1}$ is positive semidefinite.

Is LS estimate MMSE estimate?

An additional assumption:

- $\mathbf{v}$ is Gaussian white noise

Conclusion: LS estimate has the minimum mean square error among all the unbiased estimate of $\mathbf{x}$

Sketch of proof:

1. Cramer-Rao inequality: for any unbiased estimate $\bar{\mathbf{x}}$, we have

$$E(\bar{\mathbf{x}} - \mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})^T - M^{-1} \succeq 0,$$

   in which M is the fisher information matrix, i.e.,

$$M = E\left(\frac{\partial \ln(f(y_1, \ldots, y_N|x))}{\partial x}\right)\left(\frac{\partial \ln(f(y_1, \ldots, y_N|x))}{\partial x}\right)^T$$

2. According to the assumptions, we have

$$f(y|x) = C \prod_{i=1}^{N} \exp\left\{-\frac{[y_i - \mathbf{h}(i)^T x]^2}{2\sigma^2}\right\}$$

   and $M^{-1} = E(\mathbf{x} - \hat{\mathbf{x}}_{\mathrm{LS}})(\mathbf{x} - \hat{\mathbf{x}}_{\mathrm{LS}})^T$.

Derivation of the likelihood function

$$
\begin{aligned}
&P\left[\begin{pmatrix}\mathbf{y}(1)\\ \vdots \\ \mathbf{y}(N)\end{pmatrix} \le \begin{pmatrix} y_1 \\ \vdots \\ y_N\end{pmatrix} \Big| \mathbf{x} = x\right] \\
&= P\left[\begin{pmatrix}\mathbf{v}(1)\\ \vdots \\ \mathbf{v}(n)\end{pmatrix} \le \begin{pmatrix} y_1 - \mathbf{h}(1)^T x \\ \vdots \\ y_N - \mathbf{h}(N)^T x\end{pmatrix}\right] \\
&= P(\mathbf{v}(1) \le y_1 - \mathbf{h}(1)^T x) \cdots P(\mathbf{v}(n) \le y_N - \mathbf{h}(N)^T x)
\end{aligned}
$$

$$
f(y_1, \ldots, y_N | x) = \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{\left[y_i - \mathbf{h}(i)^T x\right]^2}{2\sigma^2} \right\}
$$

- LS estimate is also the maximum likelihood estimate.

Example

Consider again the Gaussian additive noise problem.

$$\mathbf{z} = \mathbf{s} + \mathbf{v}.$$

Assume we have measurements $z(1), \ldots, z(N)$ and $\mathbf{v}$ is zero mean, then the LS estimate is given by

$$\hat{s}_{\text{LS}} = (H^T H)^{-1} H^T [z(1), \ldots, z(N)]^T = \frac{1}{N}(z(1) + \ldots + z(N))$$

which is the same as the mean filter and is unbiased.

Example: If $\mathbf{v}$ is Gaussian

Derive the maximum likelihood estimate of $\mathbf{s}$.

- The likelihood function can be written as

$$f(z(1),\ldots,z(N)|\mathbf{s}=s) = \prod_i f(z(i)|\mathbf{s}=s) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_v} e^{-(z(i)-s)^2/2\sigma_v^2}$$

- The log-likelihood function

$$\log f(z(1),\ldots,z(N)|\mathbf{s}=s) = C - \frac{1}{2\sigma_v^2}\sum_{i=1}^{N}(z(i)-s)^2$$

- The maximum likelihood estimate is

$$\hat{s}_{\mathrm{ML}} = \hat{s}_{\mathrm{LS}}.$$

Weighted Least Squares

- In previous LS estimate, we assumed that we had an equal amount of confidence in all of our measurements
- Now suppose we have more confidence in some measurements than others.
- A closely related problem is weighted least squares

$$E_e = \sum_{n=1}^{N} w_n^2 [y_n - h(n)^T x]^2 = (y - Hx)^T W (y - Hx)$$

in which $W = \mathrm{diag}\{w_1^2, \ldots, w_N^2\}$ and $y = [y_1, \ldots, y_N]^T$.

Solving the problem

- The cost function E_e can be written as

$$\begin{aligned} E_e &= e^T W e \\ &= y^T W y - x^T H^T W y - y^T W H x + x^T H^T W H x \end{aligned}$$

- Necessary condition:

$$\nabla_x f = \frac{\partial E_e}{\partial x} = -y^T W H + x^T H^T W H = 0$$

- Sufficient condition:

$$\nabla_x^2 f = \frac{\partial^2 E_e}{\partial x \partial x^T} = H^T W H \succ 0$$

Solution

$$\hat{x} = (H^T W H)^{-1} H^T W y$$

$$\hat{\mathbf{x}} = (H^T W H)^{-1} H^T W \mathbf{y}$$

Note that the uniqueness of WLS estimate requires that the matrix $H^T W H$ to be positive definite

Contents

Recursive Least Squares

There is a problem in the LS estimation.

- the H matrix is an $M \times n$ matrix
- if we obtain measurements sequentially and want to update our estimate of $\mathbf{x}$ with each new measurement, we need to augment the H matrix and completely recompute the estimate $\hat{\mathbf{x}}$
- If the number of measurements becomes large, then the computational effort could become prohibitive

Problem formulation

- A linearly recursive estimator can be written in the form

$$\mathbf{y}_k = H_k \mathbf{x} + \mathbf{v}_k$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + K_k(\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k-1})$$

- we compute $\hat{\mathbf{x}}_k$ on the basis of previous estimate $\hat{\mathbf{x}}_{k-1}$ and new measurement $\mathbf{y}_k$
- K_k is the estimator gain matrix to be determined
- the quantity $(\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k-1})$ is called the correction term or innovation

The mean of the estimation error

The estimation error mean can be computed as ($\hat{\mathbf{x}}_k$ is a random variable)

$$\begin{aligned}
E(\epsilon_{x,k}) &= E(\mathbf{x} - \hat{\mathbf{x}}_k) \\
&= E[\mathbf{x} - \hat{\mathbf{x}}_{k-1} - K_k(\mathbf{y}_k - H_k\hat{\mathbf{x}}_{k-1})] \\
&= E[\epsilon_{x,k-1} - K_k(H_k\mathbf{x} + \mathbf{v}_k - H_k\hat{\mathbf{x}}_{k-1})] \\
&= E[\epsilon_{x,k-1} - K_kH_k(\mathbf{x} - \hat{\mathbf{x}}_{k-1}) - K_k\mathbf{v}_k] \\
&= (I - K_kH_k)E(\epsilon_{x,k-1}) - K_kE(\mathbf{v}_k)
\end{aligned}$$

where $\epsilon_{x,k} = \mathbf{x} - \hat{\mathbf{x}}_k$.

Unbiased estimator

- if $E(v_k) = 0$ and $E(\epsilon_{x,k-1}) = 0$, then $E(\epsilon_{x,k}) = 0$
- if the measurement noise v_k is zero-mean for all k, and the initial estimate of $\mathbf{x}$ is set equal to the expected value of $\mathbf{x}$, i.e., $\hat{\mathbf{x}}_0 = E(\mathbf{x})$, then the expected value of $\hat{\mathbf{x}}_k$ is equal to $E(\mathbf{x})$ for all k

This property holds regardless of the value of the gain matrix K_k.

Determination of the optimal value of K_k

The optimally criterion (the sum of the variances of the estimation errors at time k):

$$\begin{aligned}
J_k &= E[(x_1 - \hat{x}_{1,k})^2] + \ldots + E[(x_n - \hat{x}_{n,k})^2] \\
&= E(\epsilon_{x1,k}^2 + \ldots + \epsilon_{xn,k}^2) \\
&= E(\epsilon_{x,k}^T \epsilon_{x,k}) \\
&= E[\mathrm{Tr}(\epsilon_{x,k} \epsilon_{x,k}^T)] \\
&= \mathrm{Tr} P_k
\end{aligned}$$

where $P_k = E(\epsilon_{x,k} \epsilon_{x,k}^T)$ is the estimation error covariance.

Recursive formula for the calculation of P_k

$$\begin{aligned}
P_k &= E(\epsilon_{x,k}\epsilon_{x,k}^T) \\
&= E\left\{[(I - K_kH_k)\epsilon_{x,k-1} - K_kv_k][\cdots]^T\right\} \\
&= (I - K_kH_k)E(\epsilon_{x,k-1}\epsilon_{x,k-1}^T)(I - K_kH_k)^T - \\
&\quad K_kE(v_k\epsilon_{x,k-1}^T)(I - K_kH_k)^T - (I - K_kH_k)E(\epsilon_{x,k-1}v_k^T)K_k^T + \\
&\quad K_kE(v_kv_k^T)K_k^T
\end{aligned}$$

As $\epsilon_{x,k-1}$ is independent of v_k, we have (suppose $R_k = E(v_kv_k^T)$)

$$E(v_k\epsilon_{x,k-1}^T) = E(v_k)E(\epsilon_{x,k-1}^T) = 0,$$

$$P_k = (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T$$

Consistent with intuition

- As the measurement noise increases (i.e., R_k increases), the uncertainty in our estimate also increases (i.e., P_k increases)
- P_k should be positive semidefinite since it is a covariance matrix
- P_k is positive definite provided that P_{k-1} and R_k are positive definite

Find the optimal value of K_k

We choose K_k to make the cost function (the trace of P_k) small then the estimation error will not only be zero-mean, but it will also be consistently close to zero.

$$\frac{\partial J_k}{\partial K_k} = 0$$

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_k H_k)P_{k-1}(-H_k^T) + 2K_k R_k = 0$$

Then

$$K_k R_k = (I - K_k H_k)P_{k-1}H_k^T$$

$$K_k(R_k + H_k P_{k-1} H_k^T) = P_{k-1}H_k^T$$

$$K_k = P_{k-1}H_k^T(H_k P_{k-1} H_k^T + R_k)^{-1}$$

Recursive least squares estimation

1. Initialization: $\hat{\mathbf{x}}_0 = E(\mathbf{x})$, $P_0 = E[(\mathbf{x} - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]$
2. Iteration (for k):
   - obtain the measurement $\mathbf{y}_k$, assuming that $\mathbf{y}_k$ is given by the equation

$$\mathbf{y}_k = H_k \mathbf{x} + v_k$$

   - update the estimate of x and the estimation-error covariance P as follows:

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + K_k (y_k - H_k \hat{\mathbf{x}}_{k-1})$$

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

Important assumptions

- if no knowledge about $\mathbf{x}$ is available before measurements are taken, then $P_0 = \infty I$. If perfect knowledge about $\mathbf{x}$ is available before measurements are taken, then $P_0 = 0$.
- the measurement noise at each time step k is independent, i.e., $E(\mathbf{v}_i \mathbf{v}_k) = R_k \delta_{k-i}$. That is, the measurement noise is white.

Alternate estimator forms

- sometimes it is useful to write the equations for P_k and K_k in alternate forms
- although these alternate forms are mathematically identical, they can be beneficial from a computational point of view

Alternate form for P_k

assume $S_k = (H_k P_{k-1} H_k^T + R_k)$, then

$$K_k = P_{k-1} H_k^T S_k^{-1},$$

substituting for K_k from the above into the expression of P_k, we obtain

$$P_k = [I - P_{k-1} H_k^T S_k^{-1} H_k] P_{k-1} [\cdots]^T + P_{k-1} H_k^T S_k^{-1} R_k S_k^{-1} H_k P_{k-1}^T$$

expand terms to obtain

$$\begin{aligned} P_k =& P_{k-1} - P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} - P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + \\ & P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1} H_k^T S_k^{-1} R_k S_k^{-1} H_k P_{k-1} \end{aligned}$$

Alternate form for P_k

Combining the last two terms in the above equation gives

$$\begin{aligned} P_k &= P_{k-1} - 2P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} S_k S_k^{-1} H_k P_{k-1} \\ &= P_{k-1} - 2P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + P_k H_k^T S_k^{-1} H_k P_{k-1} \\ &= P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} \end{aligned}$$

As $K_k = P_{k-1}H_k^T S_k^{-1}$, we obtain

$$\begin{aligned} P_k &= P_{k-1} - K_k H_k P_{k-1} \\ &= (I - K_k H_k) P_{k-1} \end{aligned}$$

Problems existed in the alternate form for P_k

Numerical computing problems (i.e., scaling issues) may cause this expression for P_k to be not positive definite, even when P_{k-1} and R_k are positive definite.

Matrix inversion lemma:

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1}$$

Another formula for P_k

$$P_k = P_{k-1} - P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}H_kP_{k-1}$$

$$P_k^{-1} = [P_{k-1} - P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}H_kP_{k-1}]^{-1}$$

Applying the matrix inversion lemma:

$$\begin{aligned} P_k^{-1} =& P_{k-1}^{-1} + P_{k-1}^{-1}P_{k-1}H_k^T[(H_kP_{k-1}H_k^T + R_k) - \\ & H_kP_{k-1}P_{k-1}^{-1}(P_{k-1}H_k^T)]^{-1}H_kP_{k-1}P_{k-1}^{-1} \\ =& P_{k-1}^{-1} + H_k^TR_k^{-1}H_k \end{aligned}$$

Another formula for P_k

Inverting both sides of the previous equation gives

$$P_k = [P_{k-1}^{-1} + H_k^T R_k^{-1} H_k]^{-1}$$

This equation for P_k is more complicated in that it requires three matrix inversions, but it may be computationally advantageous in some situations.

Equivalent equation for the estimator gain K_k

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$$

Premultiplying the right side by $P_k P_k^{-1}$ gives

$$K_k = P_k P_k^{-1} P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$$

substituting for P_k^{-1} from

$$P_k = [P_{k-1}^{-1} + H_k^T R_k^{-1} H_k]^{-1}$$

gives

$$K_k = P_k (P_{k-1}^{-1} + H_k^T R_k^{-1} H_k) P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$$

multiply the factor $P_{k-1} H_k^T$ inside the first term in parentheses gives

$$K_k = P_k (H_k^T + H_k^T R_k^{-1} H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1}$$

Equivalent equation for the estimator gain K_k

Now bring H_k^T out to the left side of the parentheses to obtain

$$K_k = P_k H_k^T (I + R_k^{-1} H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1}$$

Now premultiply the first parenthetical expression by R_k^{-1}, and multiply on the inside of the parenthetical expression by R_k, to obtain

$$\begin{aligned} K_k &= P_k H_k^T R_k^{-1} (R_k + H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \\ &= P_k H_k^T R_k^{-1} \end{aligned}$$

General recursive least squares estimation

- The measurement equations:

$$
\begin{aligned}
\mathbf{y}_k &= H_k \mathbf{x} + v_k \\
E(v_k) &= 0 \\
E(v_k v_i^T) &= R_k \delta_{k-i}
\end{aligned}
$$

- The initial estimate of the constant vector x, along with the uncertainty in that estimate

$$
\begin{aligned}
\hat{\mathbf{x}}_0 &= E(\mathbf{x}) \\
P_0 &= E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]
\end{aligned}
$$

General recursive least squares estimation

- The recursive least squares algorithm: For $k = 1, 2, \cdots,$

$$\begin{aligned}
K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\
&= P_k H_k^T R_k^{-1} \\
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + K_k (y_k - H_k \hat{\mathbf{x}}_{k-1}) \\
P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \\
&= (P_{k-1}^{-1} + H_k^T R_k^{-1} H_k)^{-1} \\
&= (I - K_k H_k) P_{k-1}
\end{aligned}$$

From another point of view

- According to least squares estimation, we have

$$\hat{x} = (H^T H)^{-1} H^T y$$

- Assume we have measurements till time k,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} H_1 \\ \vdots \\ H_k \end{bmatrix} \cdot x + \begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix}$$

then the estimate is given by

$$\hat{x}(k) = [H(k)^T H(k)]^{-1} H(k)^T y(k)$$

in which $H(k) = [H_1^T, \ldots, H_k^T]^T$, $y(k) = [y_1, \ldots, y_k]^T$.

From another point of view

- When the time $k+1$ comes, we have

$$\hat{x}(k+1) = [H(k+1)^T H(k+1)]^{-1} H(k+1)^T \mathbf{y}(k+1)$$

in which

$$H(k+1) = \begin{bmatrix} H(k) \\ H_{k+1} \end{bmatrix}.$$

- the problem is how to express the estimate $\hat{x}_{k+1}$ as an incremental expression.

Relationship between $\hat{x}(k+1)$ and $\hat{x}(k)$

- Assume $C(k) = H(k)^T H(k)$
- Then

$$
\begin{aligned}
\hat{x}(k+1) - \hat{x}(k) &= C(k+1)^{-1} \cdot \left[\sum_{i=1}^{k} H_i^T y_i + H_{k+1}^T y_{k+1} \right] - C(k)^{-1} \sum_{i=1}^{k} H_i^T y_i \\
&= [C(k+1)^{-1} - C(k)^{-1}] \cdot \sum_{i=1}^{k} H_i^T y_i + C(k+1)^{-1} H_{k+1}^T y_{k+1} \\
&= C(k+1)^{-1} [C(k) - C(k+1)] C(k)^{-1} \cdot \sum_{i=1}^{k} H_i^T y_i + C(k+1)^{-1} H_{k+1}^T y_{k+1} \\
&= C(k+1)^{-1} \left(-H_{k+1}^T H_{k+1} \right) \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T y_{k+1} \\
&= C(k+1)^{-1} H_{k+1}^T \left(y_{k+1} - H_{k+1} \hat{x}(k) \right)
\end{aligned}
$$

- Equivalently,

$$
\hat{x}(k+1) = \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T \left[y_{k+1} - H_{k+1} \hat{x}(k) \right]
$$

Alleviate the burden for calculating inversion

- As $C(k+1) = C(k) + H_{k+1}^T H_{k+1}$
- According to the matrix inversion lemma, we have

  $C(k+1)^{-1} =$

  $C(k)^{-1} - C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1} H_{k+1} C(k)^{-1}$
- Then

  $C(k+1)^{-1} H_{k+1}^T = C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1}$
- Assume $\tilde{K}(k+1) = C(k+1)^{-1} H_{k+1}^T$, then we have

$$\tilde{K}(k+1) = C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1}$$
$$C(k+1)^{-1} = \left[I - \tilde{K}(k+1) H_{k+1} \right] C(k)^{-1}$$

Another formulation of RLS

- $\tilde{K}(k+1) = C(k)^{-1} H_{k+1}^T \left[I + H_{k+1} C(k)^{-1} H_{k+1}^T \right]^{-1} = C(k+1)^{-1} H_{k+1}^T$
- $C(k+1)^{-1} = \left[I - \tilde{K}(k+1) H_{k+1} \right] C(k)^{-1}$
- $\hat{x}(k+1) = \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T \left(y_{k+1} - H_{k+1} \hat{x}(k) \right)$

RLS 1 VS RLS 2

RLS 1	RLS 2
$\hat{\mathbf{x}}_0 = E(\mathbf{x})$	$\hat{x}(1) = (H_1^T H_1)^{-1} H_1 y(1)$
$P_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]$	Statistical properties of the noise is known or unknown
$\begin{aligned} K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\ &= P_k H_k^T R_k^{-1} \\ P_k &= [I - K_k H_k] P_{k-1} \\ &= (P_{k-1}^{-1} + H_k^T R_k^{-1} H_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1}) \end{aligned}$	$\begin{aligned} \tilde{K}(k) &= C(k-1)^{-1} H_k^T [I + H_k C(k-1)^{-1} H_k^T]^{-1} \\ &= C(k)^{-1} H_k^T \\ C(k)^{-1} &= [I - \tilde{K}(k) H_k] C(k-1)^{-1} \\ &= (C(k-1) + H_k^T H_k)^{-1} \\ \hat{x}(k) &= \hat{x}(k) + \tilde{K}(k)[y_k - H_k \hat{x}(k-1)] \end{aligned}$

Expression: very similar!

Consistency

- For RLS 2, the estimation error at time k is

$$x - \hat{x}(k) = x - C(k)^{-1}H(k)^T y(k) = -C(k)^{-1}H(k)^T \cdot v(k)$$

  in which $v(k) = [v_1, \ldots, v_k]^T$, $H(k)$ and $v(k)$ are independent.

- The expectation of the estimation error is the same as that in the simple LS case

$$E[x - \hat{x}(k)] = 0$$

- The variance is,

$$E\{[x - \hat{x}(k)][x - \hat{x}(k)]^T\} = R_k[H(k)^T H(k)]^{-1}$$

Consistency

- The RLS 2 estimate is consistent, i.e.,

$$\lim_{k\to\infty} E\{[x-\hat{x}(k)][x-\hat{x}(k)]^T\} = 0$$

Sketch of proof. $R_k[H(k)^T H(k)]^{-1} = \frac{R_k}{k}\left[\frac{H(k)^T H(k)}{k}\right]^{-1}$, assume ergodicity

- The RLS 1 estimate is consistent, i.e.,

$$\lim_{k\to\infty} P_k = 0$$

Sketch of proof. $P_k^{-1} = P_{k-1}^{-1} + H_k^T R_k^{-1} H_k$, also assume ergodicity

Interpretation

- stochastic gradient: (k is stochastic)
  - $\hat{x}(k+1) = \hat{x}(k) + \rho H_{k+1}^T (y_{k+1} - H_{k+1}\hat{x}(k))$
  - ρ is the stepsize, gradient decent direction
- RLS 1
  - $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + K_k(y_k - H_k\hat{\mathbf{x}}_{k-1})$,
  - direction: $K_k = P_k H_k^T R_k^{-1}$, relationship with the gradient descent direction
- RLS 2
  - $\hat{x}(k) = \hat{x}(k-1) + \tilde{K}(k)\,(y_k - H_k\hat{x}(k-1))$
  - direction: $\tilde{K}(k) = C(k)^{-1} H_k^T$, relationship with the gradient descent direction
  - A related paper: Stochastic Gauss-Newton Algorithms for Nonconvex Compositional Optimization, ICML 2020.

Example 1

- Consider the problem of trying to estimate the resistance x of an unmarked resistor on the basis of noisy measurement from a multimeter
- however, we do not want to wait until we have all the measurements in order to have an estimate
- we want to recursively modify our estimate of x each time we obtain a new measurement
- At sample time k our measurement is

$$y_k = H_k x + v_k$$

$$H_k = 1$$

$$R_k = E(v_k^2)$$

Example 1

- Assume $R_k = R$;
- Initial estimate:

$$\hat{x}_0 = E(x)$$

$$P_0 = E[(x_0 - \hat{x}_0)(x - \hat{x}_0)^T]$$

If we have absolutely no idea about the resistance value, then $P(0) = \infty$. If we are 100% certain about the resistance value before taking any measurements, then $P(0) = 0$ (but then, of course, there would not be any need to take measurements)

Example 1

- After the first measurement ($k = 1$):

$$K_k = P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1}$$

$$K_1 = P_0(P_0 + R)^{-1}$$

$$\hat{x}_k = \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1})$$

$$\hat{x}_1 = \hat{x}_0 + \frac{P_0}{P_0 + R}(y_1 - \hat{x}_0)$$

$$P_k = (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T$$

$$P_1 = \frac{P_0R}{P_0 + R}$$

Example 1

- Repeating these calculations to find these quantities after the second measurement $(k = 2)$ gives

$$
\begin{aligned}
K_2 &= \frac{P_1}{P_1 + R} = \frac{P_0}{2P_0 + R} \\
P_2 &= \frac{P_1 R}{P_1 + R} = \frac{P_0 R}{2P_0 + R} \\
\hat{x}_2 &= \hat{x}_1 + \frac{P_1}{P_1 + R}(y_2 - \hat{x}_1) \\
&= \frac{P_0 + R}{2P_0 + R}\hat{x}_1 + \frac{P_0}{2P_0 + R} y_2
\end{aligned}
$$

Example 1

- By induction we can find general expressions for P_{k-1}, K_k, and $\hat{x}_k$ as follows:

$$
\begin{aligned}
P_{k-1} &= \frac{P_0 R}{(k-1)P_0 + R} \\
K_k &= \frac{P_0}{kP_0 + R} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - \hat{x}_{k-1}) \\
&= (1 - K_k)\hat{x}_{k-1} + K_k y_k \\
&= \frac{(k-1)P_0 + R}{kP_0 + R}\hat{x}_{k-1} + \frac{P_0}{kP_0 + R} y_k
\end{aligned}
$$

Example 1

- If x is known perfectly a priori (i.e., before any measurements are obtained) then $P_0 = 0$, and then $K_k = 0$ and $\hat{x}_k = \hat{x}_0$, i.e., the optimal estimate of x is independent of any measurements that are obtained
- If x is completely unknown a priori, then $P_0 \to \infty$, and then

$$\begin{aligned}\hat{x}_k &= \frac{(k-1)P_0}{kP_0}\hat{x}_{k-1} + \frac{P_0}{kP_0}y_k \\ &= \frac{k-1}{k}\hat{x}_{k-1} + \frac{1}{k}y_k \\ &= \frac{1}{k}[(k-1)\hat{x}_{k-1} + y_k]\end{aligned}$$

in other words, the optimal estimate of x is equal to the cumulative moving average of the measurements y_k.

Example 1

- the cumulative moving average of the measurements y_k:

$$
\begin{aligned}
\bar{y}_k &= \frac{1}{k}\sum_{j=1}^{k} y_j \\
&= \frac{1}{k}\left(\sum_{j=1}^{k-1} y_j + y_k\right) \\
&= \frac{1}{k}\left[(k-1)\left(\frac{1}{k-1}\sum_{j=1}^{k-1} y_j\right) + y_k\right] \\
&= \frac{1}{k}[(k-1)\bar{y}_{k-1} + y_k]
\end{aligned}
$$

Example 1: using RLS 2

$$\hat{x}(k+1) = \hat{x}(k) + C(k+1)^{-1} H_{k+1}^T (y_{k+1} - H_{k+1}\hat{x}(k))$$

$$H_1 = 1, \hat{x}(1) = y_1, C(1) = 1, C(2) = 2, \hat{x}(2) = \frac{1}{2}[\hat{x}(1) + y_2]$$

$$\hat{x}(k) = \frac{1}{k}\left[\sum_{j=1}^{k-1} y_j + y_k\right]$$

which is the same as RLS 1 if $P_0 = \infty$.

Example 2: computational advantages

- suppose we have a scalar parameter x and a perfect measurement of it, i.e., $H_1 = 1$ and $R_1 = 0$
- suppose that our initial estimation covariance $P_0 = 6$
- suppose that our computer provides precision of three digits to the right of the decimal point for each quantity that it computes

Example 2: computational advantages

- The estimator gain K_1 is:

$$\begin{aligned} K_1 =& P_0(P_0 + R_1)^{-1} \\ =& 6 * 1/6 \\ =& 6 * 0.167 \\ =& 1.002 \end{aligned}$$

- use the first form we obtain

$$\begin{aligned} P_1 &= (1 - K_1)P_0(1 - K_1) + K_1 R_1 K_1 \\ &= (1 - K_1)^2 P_0 + K_1^2 R_1 \\ &= 0 \end{aligned}$$

Example 2: computational advantages

- Using the third term, the covariance update is

$$
\begin{aligned}
P_1 &= (1 - K_1)P_0 \\
&= (-0.002) * 6 \\
&= -0.012
\end{aligned}
$$

  The covariance after the first measurement is negative, which is physically impossible.

- for the first form, the covariance matrix will never be negative, regardless of any numerical errors in P_0, R_1, and K_1.

Contents

Application of recursive least squares theory to the curve fitting problem

- measure data one sample at a time $(y_1, y_2, \cdots)$
- find the best fit of a curve to the data
- the curve that we want to fit to the data could be constrained to be linear, or quadratic, or sinusoid, or some other shape

Example 3: fit a straight line to a set of data points

- the linear data fitting problem can be written as

$$y_k = x_1 + x_2 t_k + v_k$$
$$E(v_k^2) = R_k$$

  $x = [x_1, x_2]^T$, t_k is the independent variable, y_k is the noisy data.

- we want to estimate the constants x_1 and x_2
- the measurement matrix: $H_k = [1 \;\; t_k]$
- linear data fitting equation: $y_k = H_k x + v_k$

Example 3: RLS 1

- initialize our recursive estimator:

$$\hat{x}_0 = E(x)$$

$$P_0 = E[(x - \hat{x}_0)(x - \hat{x}_0)^T]$$

- iteration: for $k = 1, 2, \ldots,$

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$$

$$\hat{x}_k = \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1})$$

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

Example 3: RLS 2

- initialize our recursive estimator:

$$\hat{x}_1 = (H_1^T H_1)^{-1} H_1 y_1$$

$$C(1)^{-1} = (H_1^T H_1)^{-1}$$

- iteration: for $k = 2, \ldots,$

$$\tilde{K}_k = C_{k-1}^{-1} H_k^T (I + H_k C_k^{-1} H_k^T)^{-1}$$

$$\hat{x}_k = \hat{x}_{k-1} + \tilde{K}_k (y_k - H_k \hat{x}_{k-1})$$

$$C(k)^{-1} = (I - \tilde{K}_k H_k) C(k-1)^{-1}$$

Example 4: fit a neural network to a set of data points

- suppose we want to fit a neural network to a set of data points:

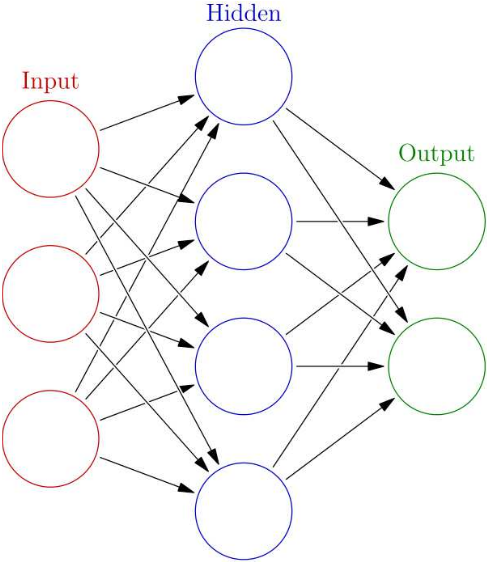
$$y_k = x_0 + \sum_{i=1}^{M} x_i B_i(t_k) + v_k$$

$$E(v_k^2) = R_k$$

  $x = [x_0, x_1, \ldots, x_M]^T$, t_k is the independent variable, y_k is the noisy data, and $B_i(t_k)$ is called the basis (kernel) function.

- we want to estimate the constants $x_0, x_1, \ldots, x_M$
- the measurement matrix: $H_k = [1, B_1(t_k), \ldots, B_M(t_k)]$
- linear data fitting equation: $y_k = H_k x + v_k$

Structure of a neural network



Example 4: fit a neural network to a set of data points

popular basis functions

- linear function; step function
- polynomial function

$$B_i(t_k) = t_k^{\alpha_i}$$

- RBF: (Gaussian)radial basis function:

$$B_i(t_k) = \exp\left\{-\frac{(t_k - \alpha_i)^2}{2\sigma_i^2}\right\}$$

- sigmoid function (S-shape):

$$B_i(t_k) = \frac{1}{1 + e^{-\beta_i t_k}}$$

- Rectified linear units:

$$B_i(t_k) = \max(0, t_k)$$

Example 4: RLS 1

- initialize our recursive estimator:

$$\hat{x}_0 = E(x)$$

$$P_0 = E[(x - \hat{x}_0)(x - \hat{x}_0)^T]$$

- iteration: for $k = 1, 2, \ldots,$

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}$$

$$\hat{x}_k = \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1})$$

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

Example 4: RLS 2

- initialize our recursive estimator:

$$\hat{x}_1 = (H_1^T H_1)^{-1} H_1 y_1$$
$$C(1)^{-1} = (H_1^T H_1)^{-1}$$

- iteration: for $k = 2, \ldots,$

$$\tilde{K}_k = C_{k-1}^{-1} H_k^T (I + H_k C_k^{-1} H_k^T)^{-1}$$
$$\hat{x}_k = \hat{x}_{k-1} + \tilde{K}_k (y_k - H_k \hat{x}_{k-1})$$
$$C(k)^{-1} = (I - \tilde{K}_k H_k) C(k-1)^{-1}$$

Example 4: fit a neural network to a set of data points

The differences in the neural network curve fitting:

- choose a suitable basis function
- determine the parameters $\alpha_i, \sigma^2, \beta_i$, which is the training of the neural network