Outline

- Normal Equations Review
- Augmented Normal Equations
- Fast Methods to Solve Normal Equations
  - Levinson Durbin Algorithm
  - Schur Algorithm

Normal Equations Review

- A method to design linear forward predictor is achieved by minimizing the energy of the error function

$$\begin{aligned} a_p &= \arg\min_{a_p} E\{|f(p)|^2\} \\ &= \arg\min_{a_p} E\{f_p[n] f_p^*[n],\} \end{aligned}$$

where

$$f_p[n] = x[n] - \hat{x}[n] = x[n] + \sum_{k=1}^{p} a_p[k] x[n-k]$$

or

$$f_p[n] = \sum_{k=0}^{p} a_p[k] x[n-k],$$

with $a_p[0] = 1$

Normal Equations Review

- The minimum of this cost function can be found by taking the derivative with respect to $a_p's$ yielding

$$\sum_{k=0}^{p} a_p[k]R_x(l-k) = 0, \quad l = 1, 2, \dots, p$$

- This is called the normal equation
- When the solution of this equation is substituted into the definition of error function, we obtain the minimum error

$$E_p^f = \sum_{k=1}^{p} a_p[k]R_x(-k)$$

Augmented Normal Equations

- If we combine the normal equations with the minimum error resulting from the optimum $a_p' s$, we obtain the augmented normal equations

$$\sum_{k=0}^{p} a_p[k] R_x(l-k) = \left\{ \begin{array}{ll} E_p^f, & l = 0 \\ 0, & l = 1, 2, \ldots, p \end{array} \right\}$$

Fast Methods to Solve Normal Equations

- Normal equations are linear equations in $a_p' s$ and can be solved easily given the second order statistics of the signal $R_x(\tau)$
- This requirement actually explains how a future value can be predicted. We simply know the statistics (related to the likelihood of a certain value)
- Although the solution is simply the solution to a linear set of equations, computation can be high when the signal length p is high
- Therefore, fast methods have been proposed to solve linear equations utilizing the special structure of the autocorrelation function $R_x(\tau)$
- We will study two of these methods
  - Levinson-Durbin algorithm
  - Schur Algorithm

Levinson-Durbin Algorithm

- This is a recursive algorithm that utilizes the toeplitz and symmetry properties of the autocorrelation function
- Let us write the normal equations in the matrix form:

$$\Gamma_p \boldsymbol{a}_p = -\boldsymbol{r}_x$$

where $\boldsymbol{a}_p$ is the vector of uknown coefficients, and Γ_p is the matrix of the autocorrelation R_x, $\boldsymbol{r}_x$ is a vector obtained from elements of $R_x(.)$

$$\Gamma_p = \begin{bmatrix} R_x(0) & R_x(-1) & ... & R_x(1-p) \\ R_x(1) & R_x(0) & ... & R_x(2-p) \\ ... & ... & ... & ... \\ R_x(p-1) & R_x(p-2) & ... & R_x(0) \end{bmatrix}$$

Levinson-Durbin Algorithm (Cont.)

- Let us rewrite Γ_p using the symmetry property of the autocorrelation function $R_x(\tau) = R_x^*(-\tau)$

$$\Gamma_p = \begin{bmatrix} R_x(0) & R_x^*(1) & ... & R_x^*(p-1) \\ R_x(1) & R_x(0) & ... & R_x^*(p-2) \\ ... & ... & ... & ... \\ R_x(p-1) & R_x(p-2) & ... & R_x(0) \end{bmatrix}$$

- This matrix is Hermitian: $\Gamma_p^{\mathrm{T}} = \Gamma_p^*$ and it is toeplitz: $\Gamma_p(i,j) = \Gamma_p(i-j)$
- Normally a matrix inversion takes time $O(p^3)$, but using these properties we will be able to solve the system more efficiently

Levinson-Durbin Algorithm (Cont.)

- Let us start by calculating $a_1(1)$

$$a_1(1) = -\frac{R_x(1)}{R_x(0)}$$

with the minimum mean squared error (MMSE)

$$\begin{aligned} E_1^f &= R_x(0) + a_1(1)R_x(-1) \\ &= R_x(0)(1 - |a_1(1)|^2) \end{aligned}$$

- Now let us calculate the second order predictor coefficients

$$a_2(1)R_x(0) + a_2(2)R_x^*(1) = -R_x(1)$$

$$a_2(1)R_x(1) + a_2(2)R_x(0) = -R_x(2)$$

resulting in the solution

$$\begin{aligned} a_2(2) &= -\tfrac{R_x(2)+a_1(1)R_x(1)}{E_1^f} \\ a_2(1) &= a_1(1) + a_2(2)a_1^*(1) \end{aligned}$$

Levinson-Durbin Algorithm (Cont.)

- Note that the values a_2's are calculated in terms of the previous step
- Continuing in this manner we can obtain all coefficients

$$\boldsymbol{a}_m = \begin{bmatrix} a_m(1) \\ a_m(2) \\ . \\ . \\ . \\ a_m(m) \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_{m-1} \\ . \\ . \\ . \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{d}_{m-1} \\ . \\ . \\ . \\ K_m \end{bmatrix}$$

- What are the values of d_m's and K_m?

Levinson-Durbin Algorithm (Cont.)

- Let us partition Γ_m as follows

$$\Gamma_m = \begin{bmatrix} \Gamma_{m-1} & \boldsymbol{r}_{m-1}^{r*} \\ \boldsymbol{r}_{m-1}^{\mathrm{rT}} & R_x(0) \end{bmatrix},$$

where $\boldsymbol{r}_{m-1}^{r} = [R_x(m-1) R_x(m-2), \ldots, R_x(1)]$

- The normal equations can now be written as

$$\begin{bmatrix} \Gamma_{m-1} & \boldsymbol{r}_{m-1}^{r*} \\ \boldsymbol{r}_{m-1}^{\mathrm{rT}} & R_x(0) \end{bmatrix} \left\{ \begin{bmatrix} \boldsymbol{a}_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{d}_{m-1} \\ K_m \end{bmatrix} \right\} = - \begin{bmatrix} \boldsymbol{r}_{m-1} \\ R_x(m) \end{bmatrix}$$

Levinson-Durbin Algorithm

- We have two equations from two rows:

$$\begin{aligned}\Gamma_{m-1}\boldsymbol{a}_{m-1} + \Gamma_{m-1}\boldsymbol{d}_{m-1} + K_m\boldsymbol{r}^{r*}_{m-1} &= -\boldsymbol{r}_{m-1} \\ \boldsymbol{r}^{\mathrm{rT}}_{m-1}\boldsymbol{a}_{m-1} + \boldsymbol{r}^{\mathrm{rT}}_{m-1}\boldsymbol{d}_{m-1} + K_m R_x(0) &= -R_x[m]\end{aligned}$$

- Solving the first row of this block form yields

$$\begin{aligned}\boldsymbol{d}_{m-1} &= -K_m\Gamma^{-1}_{m-1}\boldsymbol{r}^{r*}_{m-1} \\ &= K_m\boldsymbol{a}^{r*}_{m-1} \qquad (*)\end{aligned}$$

  where r again denotes reversing of the element order since $\Gamma_{m-1}\boldsymbol{a}_{m-1} = -\boldsymbol{r}_{m-1}$.

- Now using second row equation and Eq. (*) we obtain

$$K_m[R_x(0) + \boldsymbol{r}^{\mathrm{rT}}_{m-1}\boldsymbol{a}^*_{m-1}] + \boldsymbol{r}^{\mathrm{T}}_{m-1}\boldsymbol{a}_{m-1} = -R_x(m)$$

- Hence, we obtain K_m

$$K_m = -\frac{R_x(m) + \boldsymbol{r}_{m-1}^{\mathrm{rT}} \boldsymbol{a}_{m-1}}{R_x(0) + \boldsymbol{r}_{m-1}^{\mathrm{rT}} \boldsymbol{a}_{m-1}^{*}}$$

Levinson-Durbin Algorithm (Cont.)

- We have obtained K_m and $\boldsymbol{d}_m$, the whole recursive set of equations are

$$\begin{aligned} a_m(m) &= K_m = -\frac{R_x(m)+\boldsymbol{r}_{m-1}^{\mathrm{T}}\boldsymbol{a}_{m-1}}{R_x(0)+\boldsymbol{r}_{m-1}^{\mathrm{T}}\boldsymbol{a}_{m-1}^{*}} \\ a_m(k) &= a_{m-1}(k) + K_m a_{m-1}^{r*}(m-k) \\ & \quad k = 1, \ldots, m-1 \end{aligned}$$

- The K_m's are the reflection coefficients in the lattice structure

Levinson-Durbin Algorithm (Cont.)

- The expression for the MMSE is

$$\begin{aligned} E_m^f &= R_x(0) + \sum_{k=1}^{m} a_m(k) R_x(-k) \\ &= R_x(0) + \sum_{k=1}^{m} [a_{m-1}(k) + a_m(m) a_{m-1}^*(m-k)] R_x(-k) \\ &= E_{m-1}^f (1 - |K_m|^2) \end{aligned}$$

- The error is monotonically decreasing, makes sense since as new data comes in we should have improving performance
- Computational cost: for each stage we have $O(m)$ multiplications resulting in total $1 + 2 + \ldots + p = p(p+1)/2$ that is $O(p^2)$ operations
- Regular inversion $O(p^3)$ for arbitrary matrix

Schur Algorithm

- Consider the following

$$\alpha_0(z) = \frac{R_x(1)z^{-1} + R_x(2)z^{-2} + \ldots + R_x(p)z^{-p}}{R_x(0) + R_x(1)z^{-1} + R_x(2)z^{2} + \ldots + R_x(p)z^{-p}}$$

and the following that will be calculated recursively

$$\alpha_m(z) = \frac{\alpha_{m-1}(z) - \alpha_{m-1}(\infty)}{z^{-1}[1 - \alpha_{m-1}^*(\infty)\alpha_{m-1}(z)]}$$

- Since $\alpha_0(\infty) = 0$ we can obtain

$$\alpha_1(z) = \alpha_0(z)/z^{-1} = \frac{R_x(1) + R_x(2)z^{-1} + \ldots + R_x(p)z^{-p+1}}{R_x(0) + R_x(1)z^{-1} + R_x(2)z^{2} + \ldots + R_x(p)z^{-p}}$$

Schur Algorithm (Cont.)

- Now with these definitions, we have $\alpha_1(\infty) = R_x(1)/R_x(0)$ which is equal to $-K_1$
- The next step in the recursion

$$\alpha_2(\infty) = \frac{R_x(2) + K_1 R_x(1)}{R_x(0)(1 - |K_1|^2)}$$

  which is equal to K_2

- That is we have the relation $\alpha_m(\infty) = -K_m$
- Calculating $\alpha_m(\infty)$'s is equivalent to solving normal equations

Schur Algorithm (Cont.)

- Let us write

$$\alpha_m(z) = \frac{P_m(z)}{Q_m(z)}$$

where

$$P_0(z) = R_x(1)z^{-1} + R_x(2)z^{-2} + \ldots + R_x(p)z^{-p}$$

$$Q_0(z) = R_x(0) + R_x(1)z^{-1} + R_x(2)z^{-2} + \ldots + R_x(p)z^{-p}$$

- Consider the following block equations for recursive creation of $P_m(z)$ and $Q_m(z)$

$$\begin{bmatrix} P_m(z) \\ Q_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_{m-1} \\ K^*_{m-1}z^{-1} & z^{-1} \end{bmatrix} \begin{bmatrix} P_{m-1}(z) \\ Q_{m-1}(z) \end{bmatrix}$$

Schur Algorithm (Cont.)

- Let us check the recursive relation

$$P_1(z) = P_0(z)$$

$$Q_1(z) = z^{-1}Q_0(z)$$

- Next step

$$\begin{aligned} P_2(z) &= P_1(z) + K_1 Q_1(z) \\ &= [R_x(2) + K_1 Q_1(z) \\ Q_2(z) &= z^{-1}[Q_1(z) + K_1^* P_1(z)] \end{aligned}$$

- We can see that $P_2(z)/Q_2(z)$ is equivalent to the definition of $\alpha_2(z)$
- Recursive applications show that this is true for all m

Schur Algorithm (Cont.)

- The Steps of Schur Algorithm can systematically performed as the following
  - Create a matrix

$$G = \begin{bmatrix} 0 & R_x(1) & r_x(2) & \ldots & R_x(p) \\ R_x(0) & R_x(1) & r_x(2) & \ldots & R_x(p) \end{bmatrix}$$

  - Shift second row to right once with padding zeros for the new elements, the negative ratio of the second column is the reflection coefficient

Schur Algorithm (Cont.)

- And then
  - Multiply this shifted matrix by

$$\begin{bmatrix} 1 & K_m \\ K_m^* & 1 \end{bmatrix}$$

    for $(m = 1, 2, \ldots)$
  - Shift to right by once again, the negative ratio of the second column is K_2
  - Continue untill all K's are calculated