NONLINEAR TIME SERIES PREDICTION

Dr. Yogananda Isukapalli

The Prediction Problem

- Linear prediction
- An Introduction to Nonlinear Predictive Models
- Signal Modeling and Prediction by Neural Networks
- Volterra Filters
- Kalman State Dependent Models
- Nonlinear Time Series Examples
- Neural Network Prediction of Nonlinear Time Series
- A comparative Study and Conclusion

Linear Prediction

- FIR Wiener Filter (Tapped Delay Line)
- AR/ARMA Models
- Linear Optimal Kalman Filters

Nonlinear Predictive Models

- Nonlinear Autoregressive Models

$$X_t = h(X_{t-1}, X_{t-2}....., X_{t-p}) + e_t$$

The best prediction of X_t given X_{t-1}, …… , X_{t-p} is its conditional mean

$$\begin{aligned}\widetilde{X}_t &= E(X_t \mid X_{t-1},.....X_{t-p}) \\ &= h(X_{t-1}, X_{t-2}....., X_{t-p})\end{aligned}$$

where h(.) is an unknown smooth function

Nonlinear Predictive Models

A feedforward network is a nonlinear approximation to h given by

$$\tilde{X}_t = h(X_{t-1}, X_{t-2}\ldots\ldots, X_{t-p}) = \sum_{i=1}^{I} W_i f\left(\sum_{j=1}^{p} W_{ij} X_{t-j}\right)$$

The weight matrix is lower diagonal and will allow no feedback. Thus the feedforward network serves as nonlinear mapping from previous observation onto predictions of future observations. The function f(x) is typically a sigmoid.

The parameters Wi and Wij are estimates from a training sample $x^o_i, \ldots x^o_N$, thereby obtaining an estimate of $\bar{h}$ and h. Estimates are obtained by minimizing the sum of square residuals $\sum_{t=1}^{n}(x_t - \tilde{x}_t)^2$ by backpropogation

Nonlinear Predictive Models

- Nonlinear ARMA Models

$$X_t = h(X_{t-1}, X_{t-2}....., X_{t-p}, e_{t-1}, e_{t-2}....., e_{t-p}) + e_t$$

Predict:

$$\tilde{X}_t = \tilde{h}(X_{t-1}, X_{t-2}....., X_{t-p}, \tilde{e}_{t-1}, \tilde{e}_{t-2}....., \tilde{e}_{t-p})$$

If the $\bar{h}(.)$ is chosen, then a recurrent network can approximate it as

$$\tilde{X}_t = h(X_{t-1}, X_{t-2}....., X_{t-p}) = \sum_{i=1}^{I} W_i f\left(\sum_{j=1}^{p} W_{ij} x_{t-j} + \sum_{j=1}^{q} W'_{ij} (x_{t-j} - \tilde{x}_{t-j}) \right)$$

This model is a special case of the fully connected recurrent network

$$\tilde{X} = \sum_{i=1}^{I} W_i f\left(\sum_{j=1}^{n} W''_{ij} x_{t-j} \right)$$

where w''_{ij} are coefficients of a full matrix

Signal Modeling and Prediction by Neural Networks

- Given (by experimental measurements or by simulation) X(.) at discrete times in some time window containing times less than t, predict X(t+P), where P is some prediction time step in the future

- Use backpropogation and neural network architecture for constructing a function

$$O(t+P) = f(I_1(t), I_2(t-\Delta)....., I_m(t-m\Delta))$$

Where

O(t+P): the output of a single neuron in the output layer

I_1->I_m: Input neurons that take on values x(t), x(t-Δ), .. x(t-mΔ)

Δ: the time delay

Signal Modeling and Prediction by Neural Networks

- The O(t+p) takes the values X(t+P)
- For Feedforward networks, select input values:

$$
\begin{aligned}
I_1 &= x(t_p) \\
I_2 &= x(t_p - \Delta) \\
I_3 &= x(t_p - 2\Delta) \\
&\vdots \\
I_m &= x(t_p - (m-1)\Delta)
\end{aligned}
$$

with associated output values $O = X(t_p + P)$ for discrete times labelled by t_p.

Signal Modeling and Prediction by Neural Networks

- Takens Theorem (1981) provides a guide for choosing m such that

$$d_A < m+1 < 2d_A+1$$

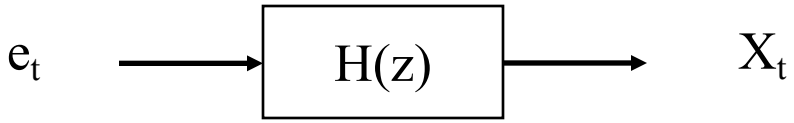
where d_A is the embedding dimension

- Takens theorem provides no information on Δ

- This prediction procedure is the nonlinear extension of the linear Widrow Hoff Algorithm

The Volterra Series

In the case of a linear causal model, the output at any time instant is a linear combination of the present and past values of the input. In other words,



$$X_t = \sum_{u=0}^{\infty} h_u e_{t-u}$$

For a non-linear causal model, the output at any time instant is a non-linear function of the present and the past values of the input.

$$X_t = f(e_{t,} e_{t-1}, e_{t-2}, ...)$$

If we assume that f is sufficiently well behaved so that it can be expanded in a Taylor series about the point O=(0,0,0), then we can expand the RHS of the above expression and write,

The Volterra Series

$$X_t = \mu + \sum_{u=0}^{\infty} g_u e_{t-u} + \sum_{u=0}^{\infty}\sum_{v=0}^{\infty} g_{uv} e_{t-u} e_{t-v}$$

$$+ \sum_{u=0}^{\infty}\sum_{v=0}^{\infty}\sum_{w=0}^{\infty} g_{uvw} e_{t-u} e_{t-v} e_{t-w} + ...$$

where,

$$\mu = f(0), \qquad g_u = \left(\frac{\delta f}{\delta e_{t-u}} \right)_0, \qquad g_{uv} = \left(\frac{\delta^2 f}{\delta e_{t-u} \delta e_{t-v}} \right)_0,$$

$$g_{uv} = \left(\frac{\delta^3 f}{\delta e_{t-u} \delta e_{t-v} \delta e_{t-w}} \right)_0, \quad \text{etc.,.}$$

This expansion is knows as (discrete time) Volterra Series, and it provides an important type of representation for non-linear models.

The Volterra Series (Contd.)

Here we consider the Second order Volterra Filter (SVF), that is the above series reduced to just the first three terms.

Let x(n) be the input to the filter at time instant 'n'. Then the output is,

$$y(n) = A^t X(n) + X^t(n) B(n) X(n)$$

where

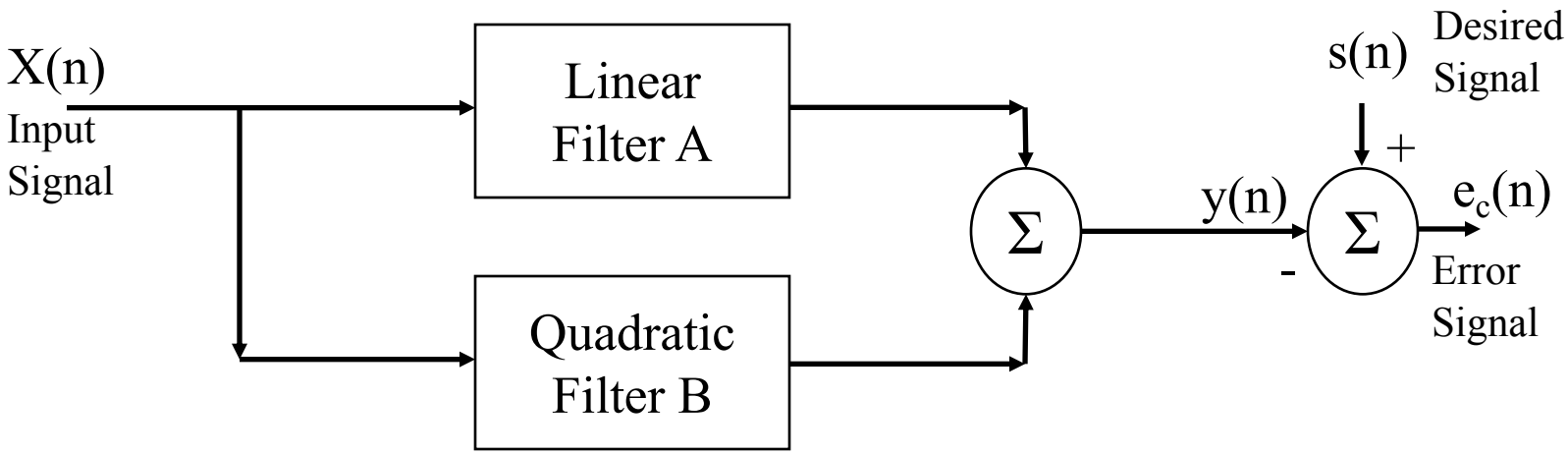
$$X(n) = [x(n), x(n-1), \ldots\ldots, x(n-m+1)]^t$$

A and B are the linear and the quadratic weights of length m and order m respectively.

$$A = [a(0), a(1), \ldots\ldots, a(m-1)]^t$$

$$B = \begin{bmatrix} b(0,0) & \ldots\ldots & b(0,m-1) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots \\ b(m-1,0) & \ldots\ldots & b(m-1,m-1) \end{bmatrix}$$

The Volterra Series (Contd.)



Many algorithms exist for updating the linear and quadratic weights of a SVF. The basic scheme is as shown in the above figure. The input x(n) and its past m-1 samples are fed to the filter. The output of the filter is compared with the desired signal and the error signal obtained is used to update the linear and quadratic weights. The desired signal for the prediction problem will be the time series sample one time instant ahead. The weights are frozen after training.

The State-Dependent Model Approach

Let us assume that the input X_t can be described in terms of finitely many values of the past input $\{X_t\}$ and output $\{e_t\}$
Then we may write,

$$X_t = h(X_{t-1}, X_{t-2}\ldots\ldots, X_{t-p}, e_{t-1}, e_{t-2}\ldots\ldots, e_{t-p}) + e_t$$

Assuming that the function h is analytic, we may expand the RHS of the above equation in a Taylor series about an arbitrary but fixed point t_0. Using only a first order expansion, we obtain

$$X_t = h(X_{t0-1}, X_{t0-2}\ldots\ldots, X_{t0-p}, e_{t0-1}, e_{t0-2}\ldots\ldots, e_{t0-p})$$
$$+ \sum_{u=1}^{k} f_u(x_{t-1})(X_{t-u} - X_{t0-u}) + \sum_{u=1}^{k} g_u(x_{t-1})(e_{t-u} - e_{t0-u}) + e_t$$

where, x_t denotes the state vector at time 't'

$$x_t = [e_{t-l+1}, \ldots\ldots, e_t, X_{t-l+1}, \ldots\ldots X_t]$$

and f_u, g_u depend on the first order partial derivatives of h.

The State-Dependent Model Approach (Contd.)

We can now rewrite the above equation in the form

$$X_t + \sum_{u=1}^{k} \phi_u(x_{t-1}) X_{t-u} = \mu(x_{t-1}) + e_t + \sum_{u=1}^{l} \psi_u(x_{t-1}) e_{t-u}$$

We call this equation a State-dependent Model of order (k,l)
Augmenting the state vector with a constant unity to take care of μ in the SDM, we have,

$$X_t = h[1, e_{t-l+1}, \ldots\ldots, e_t, X_{t-k+1}, \ldots\ldots, X_t]$$

Let us denote the components of x_t by

$$x_t = \left(x_t^{(1)}, x_t^{(2)}, \ldots\ldots, x_t^{(m)}\right)$$

where

$$x_t^{(1)} = 1, \quad x_t^{(2)} = e_{t-l+1} \quad \ldots\ldots, \quad x_t^{(m)} = x_t \qquad (m=k+l+1)$$

The State-Dependent Model Approach (Contd.)

We may then write

$$\Delta x_t = \left(\Delta x_t^{(1)}, \Delta x_t^{(2)}, ..., \Delta x_t^{(m)}\right)^t$$

$$\Delta x_t^{(1)} = 0.$$

The simples non-trivial assumption that we can make about the parameters μ, φ, and ψ is that each is a linear function of the state vector x_t, so that we may write

$$\psi_u(x_{t+1}) = \psi_u(x_t) + \Delta x_{t+1}^T \beta_u^{(t+1)}$$

$$\phi_u(x_{t+1}) = \phi_u(x_t) + \Delta x_{t+1}^T \gamma_u^{(t+1)}$$

$$\mu(x_{t+1}) = \mu(x_t) + \Delta x_{t+1}^T \alpha_u^{(t+1)}$$

The State-Dependent Model Approach (Contd.)

The basic strategy is to allow $\beta_u^{(t)}\gamma_u^{(t)}$ and $\alpha_u^{(t)}$ to wander in the form of 'random walks', and the estimation procedure would then determine, for eat t, those values of $\beta_u^{(t)}\gamma_u^{(t)}$ and $\alpha_u^{(t)}$ which would minimize the discrepancy between the observed value of X_{t+1} and its predictor X_{t+1} , computed from the model. The estimation procedure is thus based on a sequential type of algorithm, similar in the nature to the 'Kalman filter' algorithm.

The time series data considered in the computer simulations all fall under basically "AR" type of SDMs. Hence a Kalman recursion is now developed for this model. The approach followed can readily be extended to the more general "ARMA" type SDM.

An "AR" SDM takes the form

$$X_t = \mu_{t-1} - \phi_{1,t-1}X_{t-1} - \phi_{2,t-1}X_{t-2},.....,-\phi_{k,t-1}X_{t-k} + e_t$$

where

$$\mu t = \mu_{t\text{-}1} + \alpha_1^{(t)}\Delta x_t^{(2)} + \alpha_2^{(t)}\Delta x_t^{(3)}.....+ \alpha_k^{(t)}\Delta x_t^{(k+1)}$$

$$= \mu_{t\text{-}1} + \Delta x_t^{(T)}\alpha^{(t)}$$

And for each u,

$$\phi_{u,t} = \phi_{u,t-1} + \gamma_{u1}^{(t)}\Delta x_t^{(2)} + \gamma_{u2}^{(t)}\Delta x_t^{(3)},.....,+\gamma_{uk}^{(t)}\Delta x_t^{(k+1)}$$

$$= \phi_{u,t-1} + \Delta x_t^{T}\gamma_u^{(t)}$$

and $$(\alpha^{(t)} : \gamma_k^{(t)}....\gamma_k^{(t)}) = (\alpha^{(t-1)} : \gamma_k^{(t-1)}....\gamma_k^{(t-1)}) + V_t$$

The State-Dependent Model Approach (Contd.)

V_t is a matrix having a multivariate normal distribution with zero means and variance-covariance matrix ΣV

We can now rewrite 1 in the form:

$$X_t = H_t \Theta_t + e_t$$

where H_t is a 1 by $(k+1)^2$ row vector given by,

$$H_t = (1, -X_{t-1}, ..., -X_{t-k}; 0, 0, ...0),$$

(there are $k(k+1)$ 0s in the H_t)

and Θ_t is a $(k+1)^2$ by 1 column vector given by,

$$\Theta_t = (\mu_{t-1}, \phi_{1,t-1}, \phi_{2,t-1}, ..., \phi_{k,t-1}; \alpha^{(t)T}, \gamma_1^{(t)T}, \gamma_2^{(t)T} ....., \gamma_k^{(t)T})^T$$

The evolution of Θ_t over time is given by

$$\Theta_t = F_{t-1}\Theta_{t-1} + W_t$$

where the $(k+1)^2$ by $(k+1)^2$ matrix F_{t-1} is given by

The State-Dependent Model Approach (Contd.)

$$F_{t-1} = \begin{bmatrix} [I_{k=1}] & \begin{bmatrix} \Delta x_{t-1}^T & & [0] \\ & \Delta x_{t-1}^T & \\ [0] & & \Delta x_{t-1}^T \end{bmatrix} \\ [0] & [I_{k(k+1)}] \end{bmatrix}$$

$W_t = (0,0,.....0, v_{1,t}^T, ...v_{k+1,t}^T)^T,$

$v_{1,t}, ...v_{k+1,t}$ being the columns of V_t.

If we think Θ_t as the "state" and H_t as the "observation" matrix, then a direct application of the Kalman algorithm to the above equations gives the recursion

$$\hat{\Theta}_t = F_{t-1}\hat{\Theta}_{t-1} + K_t(X_t - H_t F_{t-1}\hat{\Theta}_{t-1})$$

The State-Dependent Model Approach (Contd.)

where Kt is the Kalman "gain" matrix give by

$$K_t = \Phi_t(H_t)(H_t\Phi_t(H_t)^T + \sigma_e^2)^{-1}$$

$$\Phi_t = F_{t-1}C_{t-1}(F_{t-1})^T + \Sigma_W$$

$$C_t = \Phi_t - K_t(H_t\Phi_t(H_t)^T + \sigma_e^2)(K_t)^T$$

The State-Dependent Model Approach (Contd.)

where

$$F_{t-1} = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_v \end{bmatrix}$$

$$\Phi = E((\Theta_t - F_{t-1}\hat{\Theta}_{t-1})(\Theta_t - F_{t-1}\hat{\Theta}_{t-1})^T)$$

$$C_t = E((\Theta_t - \hat{\Theta}_t)(\Theta_t - \hat{\Theta}_t)^T)$$

The initial states are obtained by fitting an AR model for an initial Stretch of data.