Auto Regressive Processes

$$x[n] = -a_1 x[n-1] - a_2 x[n-2] + \cdots - a_M x[n-M] + v[n]$$

noise

current value of x is related to previous values of x, we specify how many previous values we want to consider, $\underline{M}$

our prediction is not 100% correct.

what should $-a_1, -a_2, \cdots -a_M$ be it given data, $v[n]$ says how off we are in the estimate.

$\mathcal{E}((v[n])^2) = \sigma_v^2$

$v[n]$ has zero mean, $\mu = 0$ with some variance σ_v^2

How do we expect

$\underbrace{\mathcal{E}(x[n]x[n-k])}_{= r(k)}$ to be related to each other?

2

Estimating $r(k)$ from Data

$$X[1] \ldots X[N]$$

$$r[k] = \frac{1}{N} \sum_{j=1}^{N} x[j]\,x[j-k]$$

WIENER FILTERS

OPTIMAL LINEAR DISCRETE-TIME

filters

$$x[n] \rightarrow \boxed{H} \rightarrow y[n]$$

RANDOM Process

random output

Goal: Design H to Drive $y[n]$ To a Desired output $d[n]$

There will be error! $e[n] = d[n] - y[n]$

We want to minimize this statistically

example of a common signal we might want,

so say we want to predict the next value of the signal.

given all of the $x[n]$ we have so far
we want to predict the next value.

so the desired value is $\underline{x[n+1]}$

To do this we want to estimate
some previous value.

If I know all of the values are
correlated with each other, after seeing
enough previous data, we should be able to
a good enough job ~~to~~ of estimating $x[5]$
based on $x[1] - x[10]$

These are coupled together, so if $x[5]$
was noisy, we can look at the values around
$x[5]$ and estimate a better value of $x[5]$

when the desired value of x is one
of the previous values of x, or one of the future
values of x.

(4) minimize this in a probabilistic way, minimize the expected value of the square of this error.

what does the filter do?

$$Y[n] = \sum_{k=0}^{\infty} h[k] x[n-k]$$

Infinite Impulse Response
causal filter

$X[n]$, $d[n]$ are zero mean

WIDE SENSE STATIONary

(The statistics of a signal do not change as a function of time)

→ auto correlation is time invariant

- we want to minimize

$$J = \mathcal{E}((e[n])^2)$$

$d[n]$ is likely a Past / Future value of

$x[n]$

Take the derivative and set those things equal to zero, - minimizing

---

To minimize J as a function of the $h[k]$, we take the Derivative and (Gradient) set $= 0$

(Filtered coefficients)

$$\frac{\partial J}{\partial h[k]} = 0 \qquad k = 0, 1, 2 \ldots$$

$$J = E\left((e[n])^2\right)$$

$$\frac{\partial J}{\partial h[k]} = E\left(2e[n] \frac{\partial e[n]}{\partial h[k]}\right) \quad \text{chain rule}$$

we know

$$e[n] = d[n] - y[n] = d[n] - \sum_{k=0}^{\infty} h[k]\,x[n-k]$$

$$\frac{\partial e[n]}{\partial h[k]} = -x[n-k]$$

$$\frac{\partial J}{\partial h[k]} = E\left(-2\,e[n]\,x[n-k]\right) \quad \text{recall we set this to zero.}$$

$$= -2\,E\left(e[n]\,x[n-1]\right) = 0$$

(6) error $e[n]$ is orthogonal to all the

inputs $x[n], x[n-1], x[n-2], \dots$

The ~~Goal~~ is ~~to~~ set up a set of

equations to be satisfied,

So write

$$\frac{\partial}{\partial h[n]} J = -2\, \mathcal{E}(e[n]\, x[n-k])$$

$$= -2\, \mathcal{E}\Big(\big(d[n] - \sum_{i=0}^{\infty} h[i]\, x[n-i]\big)\, x[n-k]\Big)$$

$$= -2\, \mathcal{E}\Big(d[n]\, x[n-k] - \sum_{i=0}^{\infty} h[i]\, \mathcal{E}(x[n-i]\, x[n-k])\Big) = 0$$

$$= -2\, \mathcal{E}(d[n]\, x[n-k]) + 2 \sum_{i=0}^{\infty} h[i]\, \underbrace{\mathcal{E}(x[n-i]\, x[n-k])}_{r(i-k)}$$

$= 0$

we are minimizing the error

autocorrelation

So Now

$$\sum_{i=0}^{\infty} h[i]\, r[i-k] = p[-k]$$

$$= \mathcal{E}(x[n-k]\, d[n])$$

We define $P[-k]$ as the expected

Value

$$p[-k] = \mathcal{E}(x[n-k]\,d[n])$$

If d is a different value of the very same signal Then this becomes one of the auto-correlation Numbers

Some other correlation if related to X (Not X though)

Note the WIENER – HOPF equations

$$\boxed{\sum_{i=0}^{\infty} h[i]\,r[i-k] = p[-k]}$$

If we have an IIR filter then it's hard to think about $h[i]$ being infinite (Too much estimations)

If we allow ourselves to deal only with an FIR filter (easier life)

If we only consider a length – M FIR filter

The Wiener – Hopf equations are simpler

$$\sum_{i=0}^{M-1} h[i]\,r[i-k] = p[-k]$$

Looks like Yule – Walker equations

8 matrix form $\quad k = 0, 1$

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r[M-1] \\ r(1) & r(0) & r(1) & & r[M-2] \\ & & & \ddots & \\ & & & & r(1) \\ r(M-1) & & & r(1) & r(0) \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[M-1] \end{bmatrix} = \underbrace{\begin{bmatrix} p[0] \\ p[-1] \\ \vdots \\ p[-(M-1)] \end{bmatrix}}_{p}$$

Autocorrelation

$$Rh = p$$

optimal filter taps

$$h = R^{-1} p = R \backslash p$$

Revisit $\quad J = \mathcal{E}((e[n])^2)$

$$= E\left(\left(d[n] - \sum_{k=0}^{M-1} h[k]\, x[n-k]\right)\left(d[n] - \sum_{i=0}^{M-1} h[i]\, x[n-i]\right)\right)$$

$$= E((d[n])^2) - \sum_{k=0}^{M-1} h[k] \overbrace{\mathcal{E}(x[n-k]\, d[n])}^{p}$$

$$+ \sum_{k=0}^{M-1} \sum_{i=0}^{M-1} h[k]\, h[i] \overbrace{\mathcal{E}(x[n-k]\, x[n-i])}^{R}$$

autocorrelation says

$$= \underbrace{\sigma_d^2}_{\text{Scalere}} - \underbrace{2h^T p}_{\text{sealer}} + \underbrace{h^T R h}_{\text{Scalar as well}}$$

error for any chosen h

→ dot product

best h,

Page 8

At the optimal h, $\hat{h} = R^{-1}p$

$$\hat{J} = \sigma_d^2 - 2\hat{h}^T p + \hat{h}^T R \hat{h}$$

$$= \sigma_d^2 - 2(R^{-1}p)^T p + (R^{-1}p)^T R (R^{-1}p)$$

$$= \sigma_d^2 - 2p^T R^{-1} p + p^T R^{-1} p$$

$$= \sigma_d^2 - p^T R^{-1} p$$

This is the best error we can get (minimum error)

Variance of Desired Signal

We can show

$$J = \sigma_d^2 - 2h^T p + h^T R h$$

Error for any h

$$= \underbrace{(\sigma_d^2 - p^T R^{-1} p)}_{\hat{J}} + (p^T R^{-1} p - 2h^T p + h^T R h)$$

$$J = \hat{J} + (h - \hat{h}) R (h - \hat{h})$$

Some positive # (R is positive Definite matrix)

Error for optimal $\hat{h}$)

optimal filter is unique

as $(h-\hat{h})$ must be zero

for $\hat{J}$

suppose the desired signal, $d[n]$, is one of the original signal input values.

Special case

Linear Prediction

we want to predict $x[n+1]$

$x[n], x[n-1], \ldots \ldots x[n-(M-1)]$

One-step-Forward Linear Predictor

$$\hat{x}[n+1] = \sum_{K=0}^{M-1} h[K]\, x[n-K]$$

prediction of $x[n]$

what is the optimal set of filter taps that give me the best prediction of x

The above case is for the situation where the desired signal is one of the original filter values.

This corresponds to $d[n] = x[n+1]$

Take all of the stuff that happened just before the signal in order to predict the next one

$d[n] = x[n]$ Filtering

$d[n] = x[n + \text{positive } \#]$ prediction

$d[n] = x[n - \text{positive } \#]$ smoothing

For ONE-STEP Ahead Prediction we can find The Answer with a Wiener filter.

$d[n] = x[n+1]$ — our prediction

$e[n] = x[n+1] - \hat{x}[n+1]$

We want to minimize $J = E\left((e[n])^2 \right)$

(12) Wiener Hopf equations require

$$R\hat{h} = P_k \qquad k = 0, 1, 2 — (M-1)$$

$$\mathcal{E}\left(x[n-k] \cdot x[n+1] \right)$$

So set up matrix equations

$$\begin{pmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r(1) & r(0) & & \\ & & \ddots & r(1) \\ r(M-1) & & r(1) & r(0) \end{pmatrix} \begin{pmatrix} h[0] \\ \vdots \\ h[M-1] \end{pmatrix} = \begin{pmatrix} p(1) \\ \vdots \\ r(M) \end{pmatrix}$$

Predict the Taps

our p's are other values of auto correlations

$$p = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

WIENER HOPF for one-step ahead Predictions

Error for the optimal $\hat{h}$

$$\hat{J} = \sigma_d^2 - \underline{p}^T R^{-1} \underline{p}$$

$r = p = [r(1) \ldots r(M)]$

$\mathcal{E}((d[n])^2)$

$= \mathcal{E}((x[n+1])^2) = r(0)$

So $\hat{J} = r(0) - r^T \hat{h}$ $\qquad \hat{h} = R^{-1} \underline{p}$

$(\hat{\sigma})$

Best error and best filter $(\hat{h})$

$$\begin{bmatrix} r(0) & r^T \\ r & R \end{bmatrix} \begin{bmatrix} 1 \\ -\hat{h} \end{bmatrix} = \begin{bmatrix} \hat{J} \\ 0 \end{bmatrix}$$

$r(0) - r^T \hat{h} = \hat{J}$

$r - R\hat{h} = 0 \rightarrow r = R^{-1}\hat{h}$

Like a bigger auto correlation matrix. $(M+1) \times (M+1)$

(14) Say we have an M tap 1-step forward predictor, OR I can give one more tap.

You can Look at one more element in the past to help predict things.

Can we get a next-best filter by updating the filter we alread have

(we don't want to recalculate the filter we already have)

Levinson–Durbin Algorithm

How can we take The optimal Length-M predictor And easily Determine the OPTIMA Length M+1 Predictor?

$$\text{IF} \quad a_m = \begin{bmatrix} 1 \\ \vdots \\ -h \end{bmatrix}$$ ← Length M OPTimal fiter

$$\sum_{l=0}^{n} (a_m)_l \, r(l-i) = \begin{cases} J & i=0 \\ 0 & i=1 \cdots M \end{cases}$$

(Different way of writing the matrix equations

If we chose $d[n] = x[n-M]$ and predicted this from the samples

$x[n], x[n-1], \ldots x[n-(M-1)]$, this would be a similar problem

(smoothing, or backwards prediction)

Wiener-Hopf in this case gives a similar looking system for the optimal filter

$g[0], g[1], \ldots g[M-1]$

Forward Prediction

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ & \ddots & & \\ & & & r(1) \\ r(M-1) & & & r(0) \end{bmatrix} \begin{bmatrix} h[0] \\ \vdots \\ h[M-1] \end{bmatrix} = \begin{bmatrix} r(1) \\ \vdots \\ r(M) \end{bmatrix}$$

flipped upside down

Backward Prediction

$$\begin{bmatrix} & & \\ & R & \\ & & \end{bmatrix} \begin{bmatrix} g[0] \\ \vdots \\ g[M-1] \end{bmatrix} = \begin{bmatrix} r(M) \\ r(M-1) \\ \vdots \\ r(1) \end{bmatrix}$$

(16) The optimal gs are the optimal

hs flipped backwards

$$\cancel{\hat{g}[k] = \hat{h}[(M+1)-k]}$$

$$\hat{g}[k] = \hat{h}[M-k]$$

$\hat{J}$ (Best expected error)

is the same