

Chapter 5

Multivariate and State Space Models

5.1 Vector Time Series

Univariate methods of time series analysis can be extended to study parallel series.

Data may comprise

- stocks in a sector, indices, foreign exchange rates
- prices of bonds of different maturities
- biomedical measurements of different features
- counts of different disease occurrences across time

all which may exhibit evolution in time in some **dependent** fashion.

5.2 Stationarity and Correlation

Let $\mathbf{X}_t = (X_{1,t}, \dots, X_{K,t})^\top$ be a K -dimensional time series: $\{\mathbf{X}_t\}$ is weakly stationary if

$$\mathbb{E}[\mathbf{X}_t] = \boldsymbol{\mu} \quad \text{Var}[\mathbf{X}_t] = \mathbb{E}[(\mathbf{X}_t - \boldsymbol{\mu})(\mathbf{X}_t - \boldsymbol{\mu})^\top] = \boldsymbol{\Gamma}_0$$

and $\boldsymbol{\mu}$ and $\boldsymbol{\Gamma}_0$ are a constant $(K \times 1)$ vector and $(K \times K)$ matrix respectively.

As usual, the diagonal elements of $\boldsymbol{\Gamma}_0$ give the variances of the K elements of the time series. They will be denoted

$$\Gamma_{11}(0), \dots, \Gamma_{KK}(0)$$

Cross-Correlation

Let $\mathbf{D}$ be a $(K \times K)$ diagonal matrix

$$\mathbf{D} = \text{diag}(\sqrt{\Gamma_{11}(0)}, \dots, \sqrt{\Gamma_{KK}(0)})$$

The lag-zero cross-correlation matrix $(K \times K)$ is

$$\boldsymbol{\rho}_0 = \mathbf{D}^{-1} \boldsymbol{\Gamma}_0 \mathbf{D}^{-1}$$

with (i, j) th element

$$\rho_{ij}(0) = \frac{\Gamma_{ij}(0)}{\sqrt{\Gamma_{ii}(0)\Gamma_{jj}(0)}}$$

This matrix gives the instantaneous correlation between the elements of the vector.

Cross-Correlation

Now consider the covariance **across** series at non-zero lags:

$$\mathbf{\Gamma}_h = \mathbb{E}[(\mathbf{X}_t - \boldsymbol{\mu})(\mathbf{X}_{t-h} - \boldsymbol{\mu})^\top]$$

be the lag- h cross-covariance matrix.

The (i, j) th element of this matrix is

$$\Gamma_{ij}(h) = \mathbb{E}[(X_{i,t} - \mu_i)(X_{j,t-h} - \mu_j)].$$

Cross-Correlation

Define the lag- h cross-correlation matrix as

$$\boldsymbol{\rho}_h = \mathbf{D}^{-1} \boldsymbol{\Gamma}_h \mathbf{D}^{-1}$$

with (i,j) th element

$$\rho_{ij}(h) = \frac{\Gamma_{ij}(h)}{\sqrt{\Gamma_{ii}(0)\Gamma_{jj}(0)}}$$

This matrix gives the instantaneous correlation between the elements of the vector. For stationarity, these quantities can depend on h **but not** t .

Cross-Correlation

- $\rho_{ij}(h)$ measures the (linear) dependence of $X_{i,t}$ on $X_{j,t-h}$ which occurred prior to time t .
- $\rho_{ii}(h)$ is the lag- h autocorrelation for $\{X_{i,t}\}$.
- Note that in general

$$\rho_{ij}(h) \neq \rho_{ji}(h)$$

so Γ_h and ρ_h are not symmetric, but

$$\Gamma_{ij}(h) = \Gamma_{ji}(-h)$$

so $\Gamma_h = \Gamma_{-h}^\top$.

For $h > 0$, ρ_{ij} measures the dependence of $X_{i,t}$ on $X_{j,t-h}$.

Cross-Correlation: Lead-lag relationships

In summary

- if $\rho_{ij}(h) = \rho_{ji}(h) = 0$ for all $h > 0$, then $\{X_{i,t}\}$ and $\{X_{j,t}\}$ have **no linear relationship**.
- if $\rho_{ij}(0) \neq 0$ then $\{X_{i,t}\}$ and $\{X_{j,t}\}$ are **cross-correlated** at the same time point.
- if $\rho_{ij}(h) = 0$ and $\rho_{ji}(h) = 0$ for all $h > 0$, then $\{X_{i,t}\}$ and $\{X_{j,t}\}$ have no **lead-lag** relationship.
- if $\rho_{ij}(h) = 0$ for all $h > 0$ but $\rho_{ji}(h) \neq 0$ for some $h > 0$, then $\{X_{i,t}\}$ does **not** depend on past values of $\{X_{j,t}\}$, but $\{X_{j,t}\}$ **does** depend on past values of $\{X_{i,t}\}$.
- if $\rho_{ij}(h) \neq 0$ for some $h > 0$ and $\rho_{ji}(h) \neq 0$ for some $h > 0$, then $\{X_{i,t}\}$ and $\{X_{j,t}\}$ exhibit a **feedback relationship**.

Sample cross-correlation

We estimate the cross-correlation matrices by the usual moment estimators: for $h > 0$,

$$\hat{\Gamma}_h = \frac{1}{n} \sum_{t=h+1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_{t-h} - \bar{\mathbf{x}})^\top$$

where

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t$$

and

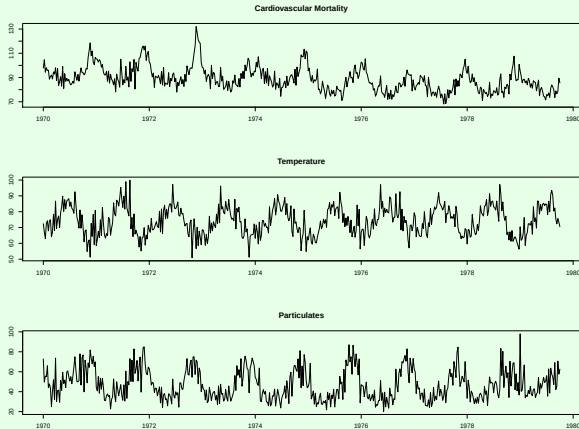
$$\hat{\rho}_h = \hat{\mathbf{D}}^{-1} \hat{\Gamma}_h \hat{\mathbf{D}}^{-1}$$

where

$$\hat{\mathbf{D}} = \text{diag} \left(\sqrt{\hat{\Gamma}_{11}(0)}, \dots, \sqrt{\hat{\Gamma}_{KK}(0)} \right)$$

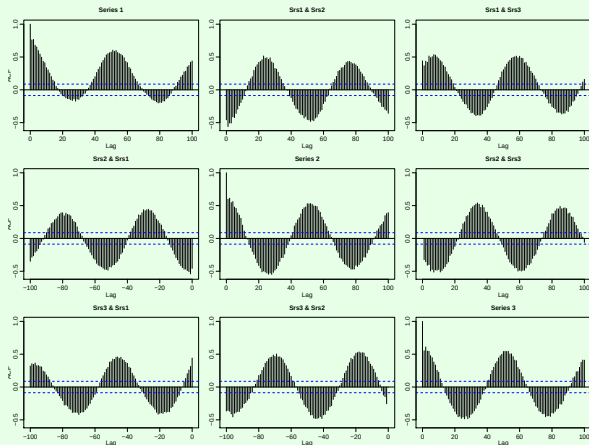
Sample cross-correlation

Example: Cardiac Mortality Data



Sample cross-correlation

Example: Cardiac Mortality Data



5.3 Vector Autoregression Models

A simple model for $\mathbf{X}_t$ takes the form

$$\mathbf{X}_t = \phi_0 + \sum_{j=1}^p \Phi_j \mathbf{X}_{t-j} + \mathbf{Z}_t$$

where

- ϕ_0 is a $K \times 1$ vector
- Φ_j is a $K \times K$ matrix
- $\mathbf{Z}_t$ is zero mean vector white noise process with

$$\mathbb{E}[\mathbf{Z}_t \mathbf{Z}_t^\top] = \Sigma$$

$\mathbf{X}_t$ follows a **vector autoregression** (VAR) model of order p .

5.3 Vector Autoregression Models

Usually

$$\mathbf{Z}_t \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

is a suitable assumption. In this case, the resulting process is a multivariate Gaussian process.

Estimation of the model can proceed using the usual likelihood methods.

Restrictions on the model are required to preserve stationarity.

Example: VAR(1)

Consider the model

$$\begin{aligned}X_{1,t} &= \phi_{10} + \phi_{11}X_{1,t-1} + \phi_{12}X_{2,t-1} + Z_{1,t} \\X_{2,t} &= \phi_{20} + \phi_{21}X_{1,t-1} + \phi_{22}X_{2,t-1} + Z_{2,t}\end{aligned}$$

Then the elements of

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}$$

define the dependence of current elements of $\mathbf{X}_t$ on past elements.

Σ defines the **concurrent** relationship between $X_{1,t}$ and $X_{2,t}$.

Example: VAR(1)

Let

$$\Sigma = \mathbf{LGL}^\top$$

be a Cholesky decomposition where $\mathbf{G}$ is a diagonal matrix, and $\mathbf{L}$ is a lower-triangular matrix. Define $\mathbf{U}_t = \mathbf{L}^{-1}\mathbf{Z}_t$. Then

$$\mathbb{E}[\mathbf{U}_t] = \mathbf{0} \qquad \text{Var}[\mathbf{U}_t] = \mathbf{G}$$

so that the elements of $\mathbf{U}_t$ are uncorrelated. Then

$$\mathbf{L}^{-1}\mathbf{X}_t = \mathbf{L}^{-1}\phi_0 + \mathbf{L}^{-1}\Phi\mathbf{X}_{t-1} + \mathbf{L}^{-1}\mathbf{Z}_t$$

so that

$$\mathbf{L}^{-1}\mathbf{X}_t = \phi^* + \Phi^*\mathbf{X}_{t-1} + \mathbf{U}_t$$

say.

Example: VAR(1)

Thus, as $\mathbf{L}$ is lower-triangular, the K th row of this vector equation is

$$X_{K,t} + \sum_{k=1}^{K-1} L_{Kk} X_{k,t} = \phi_{K0}^* + \sum_{k=1}^K \phi_{Kk}^* X_{k,t-1} + U_{K,t}$$

As the elements of $\mathbf{U}_t$ are uncorrelated, this equation shows explicitly the concurrent dependence of $X_{K,t}$ on $X_{k,t}$, for $k = 1, \dots, K - 1$.

This is termed the **structural equation** for $X_{K,t}$. A similar expression can be found for each of the elements of $\mathbf{X}_t$.

Example: $K = 2$

Let

$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix} + \begin{bmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} + \begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix}$$

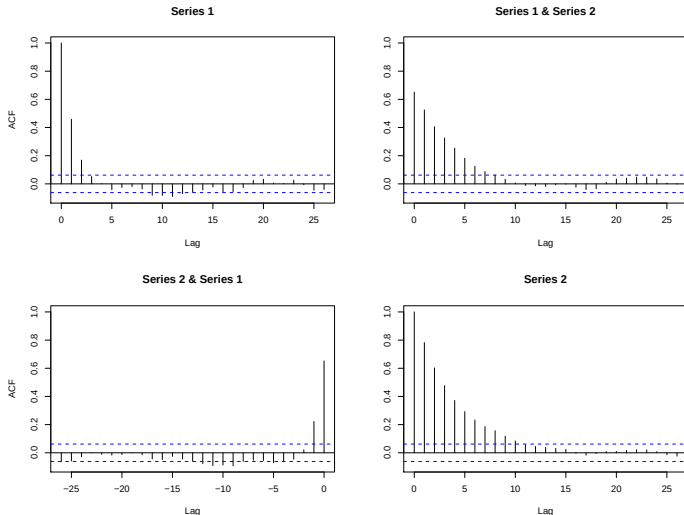
with

$$\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

Then

$$\mathbf{L}^{-1} = \begin{bmatrix} 1.0 & 0.0 \\ -0.5 & 1.0 \end{bmatrix} \quad \mathbf{G} = \mathbf{L}^{-1} \Sigma (\mathbf{L}^{\top})^{-1} = \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}$$

Example: $K = 2$ - cross-correlations



Example: VAR(1)

Using the above structural equation formulation:

$$X_{2,t} = 0.3 + 0.5X_{1,t} - 0.7X_{1,t-1} + 0.95X_{2,t-1} + U_{2,t}$$

and

$$X_{1,t} = -0.2 + 1.0X_{2,t} - 0.8X_{2,t-1} + 0.8X_{1,t-1} + U_{1,t}$$

where $U_{2,t} \sim \mathcal{N}(0, 0.5)$ and $U_{1,t} \sim \mathcal{N}(0, 1)$ are independent random variables.

These equations demonstrate the concurrent linear dependence between $X_{1,t}$ and $X_{2,t}$.

Stationarity for VAR(1)

It follows from

$$\mathbf{X}_t = \phi_0 + \Phi_1 \mathbf{X}_{t-1} + \mathbf{Z}_t$$

that, taking expectations, from stationarity

$$\mu = \phi_0 + \Phi_1 \mu \quad \therefore \quad \mu = (\mathbf{I}_K - \Phi_1)^{-1} \phi_0$$

provided $(\mathbf{I}_K - \Phi_1)$ is non-singular. Then

$$\mathbf{X}_t - \mu = \Phi_1 (\mathbf{X}_{t-1} - \mu) + \mathbf{Z}_t$$

If $\mathbf{Y}_t = \mathbf{X}_t - \mu$, then by recursion

$$\mathbf{Y}_t = \mathbf{Z}_t + \Phi_1 \mathbf{Z}_{t-1} + \Phi_1^2 \mathbf{Z}_{t-2} + \cdots$$

and hence $\text{Cov}[\mathbf{Z}_t, \mathbf{Y}_{t-h}] = \text{Cov}[\mathbf{Z}_t, \mathbf{X}_{t-h}] = \mathbf{0}$, for all $h > 0$.

Stationarity for VAR(1)

In addition,

$$\mathbf{Y}_t \mathbf{Z}_t^\top = \mathbf{Z}_t \mathbf{Z}_t^\top + \Phi_1 \mathbf{Z}_{t-1} \mathbf{Z}_t^\top + \Phi_1^2 \mathbf{Z}_{t-2} \mathbf{Z}_t^\top + \cdots$$

and taking expectations yields

$$\text{Cov}[\mathbf{Y}_t, \mathbf{Z}_t] = \mathbb{E}[\mathbf{Y}_t \mathbf{Z}_t^\top] = \mathbb{E}[\mathbf{Z}_t \mathbf{Z}_t^\top] = \Sigma$$

as $\{\mathbf{Z}_t\}$ is a white-noise process.

Stationarity for VAR(1)

In the representation, $\mathbf{X}_t$ depends on $\mathbf{Z}_{t-j}$ with coefficient matrix Φ_1^j ; this dependence must converge to zero. Therefore, using the **eigendecomposition**

$$\Phi_1 = \mathbf{A}_1 \Lambda_1 \mathbf{B}_1$$

with $\mathbf{A}_1 \mathbf{B}_1 = \mathbf{I}_K$ and $\Lambda_1 = \text{diag}(\lambda_{11}, \dots, \lambda_{1K})$, we have in the usual way

$$\Phi_1^j = \mathbf{A}_1 \Lambda_1^j \mathbf{B}_1.$$

Stationarity for VAR(1)

Thus, for each j ,

$$\Phi_1^j \longrightarrow \mathbf{0} \quad \Longleftrightarrow \quad |\lambda_{1k}| < 1, \quad k = 1, \dots, K$$

as in the univariate case. Note that

$$\det(\lambda \mathbf{I}_K - \Phi_1) = \lambda^K \det(\mathbf{I}_K - \lambda^{-1} \Phi_1)$$

and hence $\lambda_{11}, \dots, \lambda_{1K}$ are **found via** the solutions of

$$\det(\mathbf{I}_K - \Phi_1 z) = 0.$$

Stationarity for VAR(1)

Similar to the univariate case, $\mathbf{X}_t$ is stationary if and only if the roots of

$$\det(\mathbf{z}\mathbf{I}_K - \Phi_1) = 0.$$

are greater than 1 in modulus.

Finally

$$\Gamma_0 = \text{Var}[\mathbf{X}_t] = \text{Var}[\mathbf{Z}_t] + \sum_{j=1}^{\infty} \text{Var}[\Phi_1^j \mathbf{Z}_{t-j}] = \Sigma + \sum_{j=1}^{\infty} \Phi_1^j \Sigma (\Phi_1^\top)^j$$

Stationarity for VAR(1)

Using the equation

$$\mathbf{X}_t - \boldsymbol{\mu} = \boldsymbol{\Phi}_1(\mathbf{X}_{t-1} - \boldsymbol{\mu}) + \mathbf{Z}_t$$

we have for $h > 0$,

$$\begin{aligned}\mathbb{E}[(\mathbf{X}_t - \boldsymbol{\mu})(\mathbf{X}_{t-h} - \boldsymbol{\mu})^\top] \\ = \boldsymbol{\Phi}_1 \mathbb{E}[(\mathbf{X}_{t-1} - \boldsymbol{\mu})(\mathbf{X}_{t-h} - \boldsymbol{\mu})^\top] + \mathbb{E}[\mathbf{Z}_t(\mathbf{X}_{t-h} - \boldsymbol{\mu})^\top]\end{aligned}$$

or equivalently, as the final expectation is zero

$$\boldsymbol{\Gamma}_h = \boldsymbol{\Phi}_1 \boldsymbol{\Gamma}_{h-1} = \boldsymbol{\Phi}_1^h \boldsymbol{\Gamma}_0$$

by recursion.

Stationarity for VAR(1)

Thus

$$\begin{aligned}\rho_h &= \mathbf{D}^{-1} \Phi_1 \Gamma_{h-1} \mathbf{D}^{-1} \\ &= \{ \mathbf{D}^{-1} \Phi_1 \mathbf{D} \} \{ \mathbf{D}^{-1} \Gamma_{h-1} \mathbf{D}^{-1} \} \\ &= \Upsilon_1 \rho_{h-1} \\ &\vdots \\ &= \Upsilon_1^h \rho_0\end{aligned}$$

Note: $\mathbf{D} = \text{diag}(\sqrt{\Gamma_{11}(0)}, \dots, \sqrt{\Gamma_{KK}(0)})$ so square root already taken.

Results for VAR(p) models

We have

$$\mathbf{X}_t - \boldsymbol{\mu} = \sum_{j=1}^p \boldsymbol{\Phi}_j (\mathbf{X}_{t-j} - \boldsymbol{\mu}) + \mathbf{Z}_t$$

and

$$\text{Cov}[\mathbf{X}_t, \mathbf{Z}_t] = \Sigma \qquad \text{Cov}[\mathbf{X}_{t-h}, \mathbf{Z}_t] = \mathbf{0}, \quad h > 0$$

Also, for $h > 0$,

$$\boldsymbol{\Gamma}_h = \sum_{j=1}^p \boldsymbol{\Phi}_j \boldsymbol{\Gamma}_{h-j} \qquad \boldsymbol{\rho}_h = \sum_{j=1}^p \boldsymbol{\Upsilon}_j \boldsymbol{\rho}_{h-j}$$

where $\boldsymbol{\Upsilon}_j = \mathbf{D}^{-1/2} \boldsymbol{\Phi}_j \mathbf{D}^{-1/2}$

Results for VAR(p) models

As for the VAR(1),

$$\mathbf{X}_t = \boldsymbol{\mu} + \mathbf{Z}_t + \boldsymbol{\Psi}_1 \mathbf{Z}_{t-1} + \boldsymbol{\Psi}_2 \mathbf{Z}_{t-2} + \cdots$$

by the usual moving average type construction, where

$$(\mathbf{I}_K - \boldsymbol{\Phi}_1 B - \cdots - \boldsymbol{\Phi}_p B^p)(\mathbf{I}_K + \boldsymbol{\Psi}_1 B + \boldsymbol{\Psi}_2 B^2 + \cdots) = \mathbf{I}_K$$

The matrix $\boldsymbol{\Psi}_j$ is the effect of $\mathbf{Z}_t$ on $\mathbf{X}_{t+j}$; it is termed the **impulse response function** of $\mathbf{X}_t$.

The same trick to orthogonalize the system by writing

$$\boldsymbol{\Sigma} = \mathbf{LGL}^\top$$

and pre-multiplying by $\mathbf{L}^{-1}$ can be used.

Vector MA Models

Let

$$\mathbf{X}_t = \boldsymbol{\theta}_0 + \mathbf{Z}_t + \boldsymbol{\Theta}_1 \mathbf{Z}_{t-1} + \cdots + \boldsymbol{\Theta}_q \mathbf{Z}_{t-q} = \boldsymbol{\theta}_0 + \boldsymbol{\Theta}(B) \mathbf{Z}_t$$

where the $\boldsymbol{\Theta}_j$ are $K \times K$ matrices, and $\boldsymbol{\theta}_0$ is $K \times 1$. Then

- $\text{Cov}[\mathbf{X}_t, \mathbf{Z}_t] = \Sigma$
- $\boldsymbol{\Gamma}_0 = \Sigma + \boldsymbol{\Theta}_1 \Sigma \boldsymbol{\Theta}_1^\top + \cdots + \boldsymbol{\Theta}_q \Sigma \boldsymbol{\Theta}_q^\top$
-

$$\boldsymbol{\Gamma}_h = \begin{cases} \sum_{j=1}^q \boldsymbol{\Theta}_j \Sigma \boldsymbol{\Theta}_{j-h}^\top & 1 \leq h \leq q \\ 0 & h > q \end{cases}$$

where $\boldsymbol{\Theta}_0 = \mathbf{I}_K$.

Vector ARMA Models

Let

$$\Phi(B)\mathbf{X}_t = \phi_0 + \Theta(B)\mathbf{Z}_t$$

where

$$\Phi(B) = \mathbf{I}_K - \Phi_1 B - \cdots - \Phi_p B^p$$

$$\Theta(B) = \mathbf{I}_K + \Theta_1 B + \cdots + \Theta_q B^q$$

This describes a vector ARMA(p, q) (VARMA(p, q)) model.

The marginal processes are ARMA($Kp, (K - 1)p + q$) processes.

5.4 Cointegration

Consider the VARMA(1, 1) model

$$\Phi(B)\mathbf{X}_t = \Theta(B)\mathbf{Z}_t$$

with

$$\Phi(B) = \Phi_1(B) = \begin{bmatrix} 1 - 0.5B & B \\ 0.25B & 1 - 0.5B \end{bmatrix}$$

$$\Theta(B) = \Theta_1(B) = \begin{bmatrix} 1 - 0.2B & 0.4B \\ 0.1B & 1 - 0.2B \end{bmatrix}$$

with

$$\text{Var}[\mathbf{Z}_t] = \Sigma$$

5.4 Cointegration

That is

$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 0.5 & -1 \\ -0.25 & 0.5 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} \\ + \begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} + \begin{bmatrix} -0.2 & 0.4 \\ 0.1 & -0.2 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \end{bmatrix}$$

or

$$\mathbf{X}_t = \mathbf{\Phi} \mathbf{X}_{t-1} + \mathbf{Z}_t + \mathbf{\Theta} \mathbf{Z}_{t-1}.$$

say.

5.4 Cointegration

Then

$$\begin{bmatrix} 1 - 0.5B & B \\ 0.25B & 1 - 0.5B \end{bmatrix} \mathbf{X}_t = \begin{bmatrix} 1 - 0.2B & 0.4B \\ 0.1B & 1 - 0.2B \end{bmatrix} \mathbf{Z}_t$$

Filtering on both sides by

$$\begin{bmatrix} 1 - 0.5B & -B \\ -0.25B & 1 - 0.5B \end{bmatrix}$$

we obtain on the left and right hand sides respectively

$$\begin{bmatrix} 1 - 0.5B & -B \\ -0.25B & 1 - 0.5B \end{bmatrix} \begin{bmatrix} 1 - 0.5B & B \\ 0.25B & 1 - 0.5B \end{bmatrix} = \begin{bmatrix} 1 - B & 1 \\ 0 & 1 - B \end{bmatrix}$$

$$\begin{bmatrix} 1 - 0.5B & -B \\ -0.25B & 1 - 0.5B \end{bmatrix} \begin{bmatrix} 1 - 0.2B & 0.4B \\ 0.1B & 1 - 0.2B \end{bmatrix} = \begin{bmatrix} 1 - 0.7B & -0.6B \\ -0.15B & 1 - 0.7B \end{bmatrix}.$$

5.4 Cointegration

Therefore

$$\begin{bmatrix} 1 - B & 0 \\ 0 & 1 - B \end{bmatrix} \mathbf{X}_t = \begin{bmatrix} 1 - 0.7B & -0.6B \\ -0.15B & 1 - 0.7B \end{bmatrix} \mathbf{Z}_t$$

That is

$$\begin{bmatrix} (1 - B)X_{1,t} \\ (1 - B)X_{2,t} \end{bmatrix} = \begin{bmatrix} Z_{1,t} - 0.7Z_{1,t-1} - 0.6Z_{2,t-1} \\ Z_{2,t} - 0.15Z_{1,t-1} - 0.7Z_{2,t-1} \end{bmatrix}$$

Processes $X_{1,t}$ and $X_{2,t}$ have unit roots, specifically

$$X_{1,t}, X_{2,t} \sim \text{ARIMA}(0, 1, 1).$$

5.4 Cointegration

However, let

$$\mathbf{L} = \begin{bmatrix} 1 & -2 \\ 0.5 & 1 \end{bmatrix}$$

and let

$$\mathbf{Y}_t = \mathbf{L}\mathbf{X}_t \qquad \mathbf{U}_t = \mathbf{L}\mathbf{Z}_t$$

Then

$$\begin{aligned} \mathbf{Y}_t = \mathbf{L}\mathbf{X}_t &= \mathbf{L}\Phi\mathbf{X}_{t-1} + \mathbf{L}\mathbf{Z}_t + \mathbf{L}\Theta\mathbf{Z}_{t-1} \\ &= \mathbf{L}\Phi\mathbf{L}^{-1}(\mathbf{L}\mathbf{X}_{t-1}) + \mathbf{L}\mathbf{Z}_t + \mathbf{L}\Theta\mathbf{L}^{-1}(\mathbf{L}\mathbf{Z}_{t-1}) \\ &= \mathbf{L}\Phi\mathbf{L}^{-1}\mathbf{Y}_{t-1} + \mathbf{U}_t + \mathbf{L}\Theta\mathbf{L}^{-1}\mathbf{U}_{t-1} \end{aligned}$$

or equivalently

$$\mathbf{Y}_t - \Phi^*\mathbf{Y}_{t-1} = \mathbf{U}_t + \Theta^*\mathbf{U}_{t-1}$$

5.4 Cointegration

In this equation

$$\Phi^* = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \Theta^* = \begin{bmatrix} -0.4 & 0 \\ 0 & 0 \end{bmatrix}$$

so that

$$Y_{1,t} - Y_{1,t-1} = U_{1,t} - 0.4U_{1,t-1}$$

$$Y_{2,t} = U_{2,t}$$

Hence $\{Y_{1t}\} \sim \text{ARIMA}(0, 1, 1)$ and $\{Y_{2t}\}$ is a white noise process, and there is no dependence between the two series.

Hence, the unit roots in $\{X_{1,t}\}$ and $\{X_{2,t}\}$ result from the unit root in $\{Y_{1,t}\}$ and a subsequent linear transformation.

5.4 Cointegration

This phenomenon is known as **cointegration**.

Cointegration can be used to analyze co-movements in raw series (asset prices, exchange rates or yields).

The modelling strategy allows the detection of stable or stationary long-run relationships between non-stationary variables; it allows the underlying relationships between the series to be discovered.

5.4 Cointegration

The components of K -vector $\mathbf{X}_t$ are said to be **cointegrated** of order (d, b) , denoted

$$\mathbf{X}_t \sim CI(d, b)$$

if

- (a) the components of $\mathbf{X}_t$ are $I(d)$ (stationary after d -times differencing),

5.4 Cointegration

(b) there exists a linear combination of $\mathbf{X}_t$, $\tilde{X}_t$ say, where

$$\tilde{X}_t = \boldsymbol{\alpha}^\top \mathbf{X}_t \quad \boldsymbol{\alpha} \neq \mathbf{0}$$

such that $\tilde{X}_t \sim I(d - b)$ for some $b > 0$.

The vector $\boldsymbol{\alpha}$ is termed the **cointegrating vector**.

Time series that cointegrate indicate that stable long-run relationships (equilibria) between the series exist, irrespective of the behaviour of the individual series.

Estimation of Cointegrating Series

Consider $\{Y_t\}, \{\mathbf{X}_t\} = \{(X_{1,t}, \dots, X_{K,t})^\top\} \sim I(1)$ that are cointegrated. Suppose that

$$Y_t - (\alpha_1 X_{1,t} + \dots + \alpha_K X_{K,t}) \sim I(0)$$

that is, this process of ‘residuals’ is stationary.

We seek a statistical procedure to analyze this system.

Estimation of Cointegrating Series

1. Fit the regression model

$$Y_t = \alpha_1 X_{1,t} + \dots + \alpha_K X_{K,t} + Z_t^*$$

with (cointegrating) vector

$$\alpha = (1, -\alpha_1, \dots, -\alpha_K)^\top$$

and estimate the parameters in the model (using OLS or maximum likelihood).

It can be shown that the estimators are consistent with convergence rate $o_p(n^{-1})$.

Estimation of Cointegrating Series

2. Under cointegration, residuals $\hat{z}_t^*$ should be $I(0)$ (stationary); this can be tested using Dickey-Fuller procedures.

If $\hat{z}_t^* \sim I(0)$ is acceptable (that is, the unit root hypothesis in the Dickey-Fuller test is **rejected**) then a second regression model known as an **Error Correction Model** (ECM) can be specified.

Estimation of Cointegrating Series

For $K = 1$: Suppose $X_t, Y_t \sim I(1)$, and we conclude from the previous analysis that $\hat{Z}_t^* \sim I(0)$. Let

$$X_t^* = (1 - B)X_t \sim I(0)$$

$$Y_t^* = (1 - B)Y_t \sim I(0)$$

be the differenced series, which are stationary.

Estimation of Cointegrating Series

Consider the regression models

$$Y_t^* = \psi_0 + \gamma_1 \hat{Z}_{t-1}^* + \sum_{j=1}^p \psi_{1j} X_{t-j}^* + \sum_{l=1}^m \psi_{2l} Y_{t-l}^* + \epsilon_{1,t}$$

$$X_t^* = \xi_0 + \gamma_2 \hat{Z}_{t-1}^* + \sum_{j=1}^p \xi_{1j} Y_{t-j}^* + \sum_{l=1}^m \xi_{2l} X_{t-l}^* + \epsilon_{2,t}$$

with estimated using standard OLS/ML techniques.

Estimation of Cointegrating Series

The ECM in the first equation states that changes in the series Y_t are explained by

- the residual in the long-run equilibrium from previous time point (coefficient γ_1),
- lagged changes in the differenced X_t series, via coefficients ψ_1),
- their own history via coefficients ψ_2 .

Estimation of Cointegrating Series

Furthermore

- γ_1 determines the rate of re-adjustment; should have

$$\hat{\gamma}_1 < 0.$$

- Model orders p and m need to be selected; typically start with p, m large, and drop variables according to hypothesis tests or using model selection criteria
- Can extend to systems of cointegrated variables

Two cointegrating series

Example: Cointegrating series

Suppose that

$$X_t \sim I(1)$$

$$Y_t = 0.6X_t + Z_{2,t}$$

where $Z_{2,t} \sim \mathcal{N}(0, 1)$ is an IID series. This implies that

$$Y_t - 0.6X_t = Z_{2,t}$$

is a cointegrating system with $\alpha = (1, -0.6)^\top$.

Two cointegrating series

Example: Cointegrating series

To estimate the system parameters, we regress Y_t on X_t , and then form the error correction models.

Regressions are based on the differenced series

$$Y_t^* = \psi_0 + \gamma_1 \hat{z}_{t-1}^* + \psi_{11} X_{t-1}^* + \psi_{21} Y_{t-1}^* + \epsilon_{1,t}$$

$$X_t^* = \xi_0 + \gamma_2 \hat{z}_{t-1}^* + \xi_{11} Y_{t-1}^* + \xi_{21} X_{t-1}^* + \epsilon_{2,t}$$

where $\{\hat{z}_t^*\}$ are the residuals from the regression of Y_t on X_t .

Simulated Example: R code

Suppose

$$X_t = X_{t-1} + Z_{1,t}$$

$$Y_t = 0.6X_t + Z_{2,t}$$

where $\{Z_{1,t}\}, \{Z_{2,t}\} \sim \text{IID}(0, 1)$.

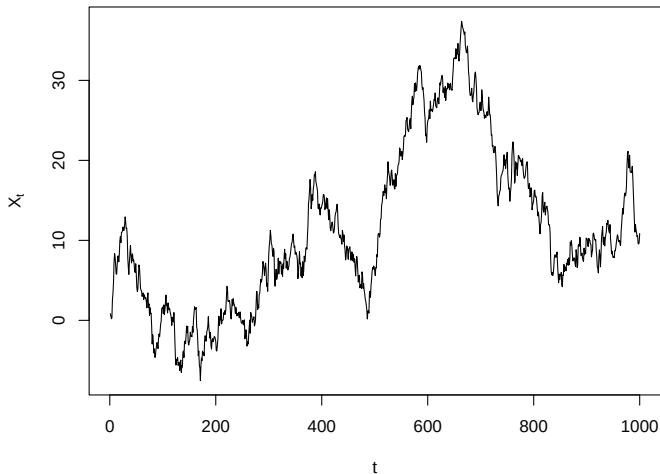
```
set.seed(123456)
n<-1000
z1<-rnorm(n)
z2<-rnorm(n)
x<-cumsum(z1);
y<-0.6*x+z2
lr.reg<-lm(y~x)
error<-residuals(lr.reg)
error.lagged<-error[-c(n-1,n)]
xstar<-diff(x); ystar<-diff(y)
diff.dat<-data.frame(embed(cbind(xstar,ystar),2))
colnames(diff.dat)<-c("xstar", "ystar", "xstar.1", "ystar.1")
```

Simulated Example: R code

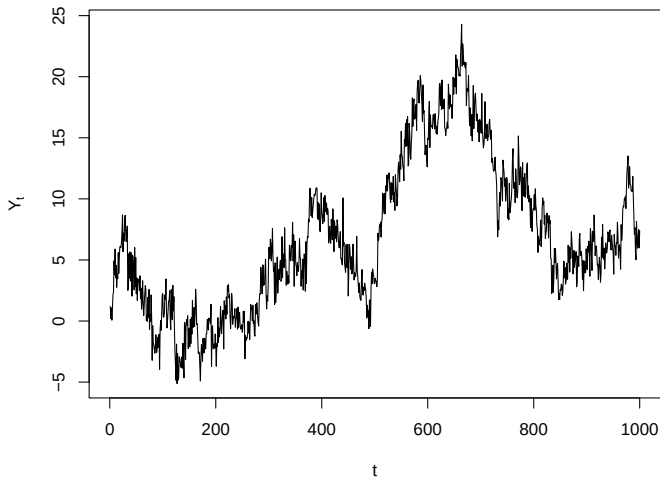
```
ecm.reg<-lm(ystar~error.lagged+xstar.1+ystar.1,data=diff.dat)
summary(ecm.reg)

##
## Call:
## lm(formula = ystar ~ error.lagged + xstar.1 + ystar.1, data = diff.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.1545 -0.8358 -0.0354  0.8193  3.5954
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.005528   0.037437   0.148   0.883
## error.lagged -0.936862   0.051442 -18.212 <2e-16 ***
## xstar.1       0.602592   0.043770  13.767 <2e-16 ***
## ystar.1      -0.972102   0.036773 -26.436 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.183 on 994 degrees of freedom
## Multiple R-squared:  0.4129, Adjusted R-squared:  0.4112
## F-statistic: 233.1 on 3 and 994 DF,  p-value: < 2.2e-16
```

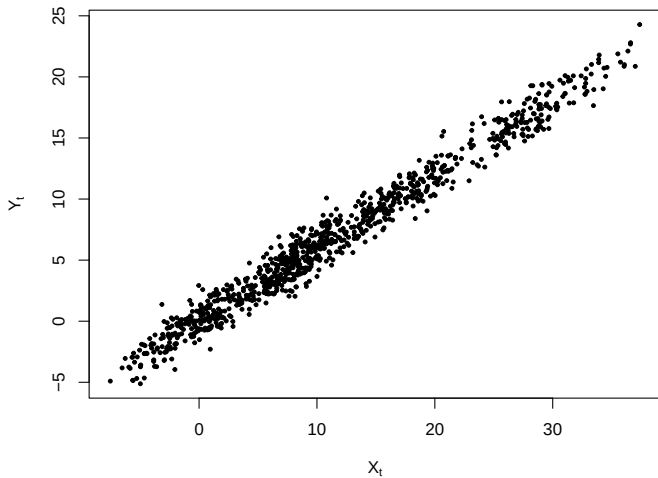
Simulated Example: R code



Simulated Example: R code



Simulated Example: R code



Results

Example: Cointegrating series

```
##           Estimate Std. Error   t value Pr(>|t|)
## (Intercept)  0.005528   0.037437   0.147668 0.882635
## error.lagged -0.936862   0.051442 -18.212078 0.000000
## xstar.1      0.602592   0.043770  13.767172 0.000000
## ystar.1     -0.972102   0.036773 -26.435535 0.000000
```

That is

$$\hat{\psi}_0 = 0.0055 \quad \hat{\gamma}_1 = -0.9369 \quad \hat{\psi}_{11} = 0.6026 \quad \hat{\psi}_{21} = -0.9721$$

5.5 State Space Models

A **state space** model can better represent the actual dynamics of a data generation process.

The principle is that an observed series is in fact generated as the **marginal** process derived from a multivariate time series model, structured so that the model is specified via two equations

- the **state** equation
- the **observation** equation

5.5 State Space Models

The general (Markov) **state equation** takes the form

$$\mathbf{X}_t = \mathbf{g}(\mathbf{x}_{t-1}; \theta_{t-1}) + \mathbf{W}_t$$

and the general **observation equation** takes the form

$$\mathbf{Y}_t = \mathbf{f}(\mathbf{x}_t; \theta_t) + \mathbf{V}_t.$$

The process $\{\mathbf{Y}_t\}$ is observed, the process $\{\mathbf{X}_t\}$ is **latent** or unobserved, and one focus of inferential interest.

Let $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$.

Inference in General State Space Models

Inference goals for state space models are

- **filtering/tracking**: estimate state at time t from observations **up to** time t

$$p(\mathbf{x}_t | \mathbf{y}_{1:t})$$

- **smoothing**: estimate state at time t from **past** and possibly some **future** observations

$$p(\mathbf{x}_t | \mathbf{y}_{1:(t+s)})$$

- **prediction/forecasting**: estimate state at time t from observations up to a previous time point $t - s$

$$p(\mathbf{x}_t | \mathbf{y}_{1:(t-s)})$$

Inference in General State Space Models

The inference mechanism is **probabilistic**; we compute the quantities of interest sequentially in the following recursive calculation:

- Prediction

$$p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_{1:(t-1)}) d\mathbf{x}_{t-1}$$

- Filtering

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})}{p(\mathbf{y}_t | \mathbf{y}_{1:(t-1)})}$$

Initially, we assume that the system parameters θ_t are known.

Inference in General State Space Models

The above equations have an analytical solution if all densities in the state and observation equation are **Gaussian**, and both the state and the observation equation are **linear**.

- If these conditions are met, the **Kalman filter** provides the optimal (Bayesian) solution to the tracking problem.
- If these conditions are not met:
 - ▶ **Extended Kalman filter**: analytical approximation.
 - ▶ **Particle filter**: sampling-based approximation.

Linear Gaussian State Space Models

The general form of a linear Gaussian state space model is given by the equations

$$\mathbf{Y}_t = \mathbf{F}_t \mathbf{X}_t + \mathbf{d}_t + \mathbf{V}_t \quad (1a)$$

$$\mathbf{X}_{t+1} = \mathbf{G}_t \mathbf{X}_t + \mathbf{e}_t + \mathbf{W}_t \quad (1b)$$

where

- $\mathbf{Y}_t$ is $(K \times 1)$
- $\mathbf{X}_t$ is $(L \times 1)$.

Linear Gaussian State Space Models

In (1)

- $\mathbf{F}_t$ is $(K \times L)$,
- $\mathbf{d}_t$ is $(K \times 1)$,
- $\mathbf{G}_t$ is $(L \times L)$,
- $\mathbf{e}_t$ is $(L \times 1)$.

These **system matrices** are assumed to be non-stochastic.

Linear Gaussian State Space Models

We may write

$$\begin{bmatrix} \mathbf{V}_t \\ \mathbf{W}_t \end{bmatrix} = \mathbf{H}_t \mathbf{Z}_t$$

where

$$\mathbf{Z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

is a Gaussian IID process, and $\mathbf{H}_t$ induces covariance between the components of $\mathbf{V}_t$ and $\mathbf{W}_t$.

Linear Gaussian State Space Models

The formulation is completed by the assumption that the initial state vector $\mathbf{X}_1$ is independent of $\mathbf{Z}_t$ at all times t and has unconditional mean and variance $\mathbf{a}_{1|0}$ and $\mathbf{P}_{1|0}$ respectively.

The state space model is characterized by the properties that the system matrices, the disturbance terms $\{\mathbf{Z}_t\}$ and initial state vector $\mathbf{X}_1$ possess.

If the system matrices do not evolve with time, the state space model is called **time-invariant** or **time-homogeneous**.

Linear Gaussian State Space Models

Example: AR(2) Model

Let $\{X_t\}$ be a zero-mean, Gaussian AR (2) process given by

$$X_{t+1} = \phi_1 X_t + \phi_2 X_{t-1} + W_t$$

where $W_t \sim \mathcal{N}(0, \sigma_W^2)$.

Suppose also that we observe the process $\{Y_t\}$ with noise, so that

$$Y_t = X_t + V_t$$

with $V_t \sim \mathcal{N}(0, \sigma_V^2)$ and independent of W_t .

Linear Gaussian State Space Models

Example: AR(2) Model

A state space representation has $K = 1$ and $L = 2$

$$Y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} X_t \\ X_{t-1} \end{bmatrix} + \begin{bmatrix} \sigma_V & 0 \end{bmatrix} \begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix}$$

$$\begin{bmatrix} X_{t+1} \\ X_t \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} X_t \\ X_{t-1} \end{bmatrix} + \begin{bmatrix} 0 & \sigma_W \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix}$$

where $\mathbf{Z}_t = (Z_{1,t}, Z_{2,t})^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$.

The state space model is time-homogeneous and Gaussian, and the observation and state equations are uncorrelated.

5.6 Kalman Filter

In the linear Gaussian case, the Kalman filter is used for

- **Filtering**: computes for all t

$$\mathbb{E}[\mathbf{X}_t | \mathbf{Y}_{1:t} = \mathbf{y}_{1:t}] \quad \text{Var}[\mathbf{X}_t | \mathbf{Y}_{1:t} = \mathbf{y}_{1:t}].$$

- **Smoothing**: computes for all $t < n$

$$\mathbb{E}[\mathbf{X}_t | \mathbf{Y}_{1:n} = \mathbf{y}_{1:n}] \quad \text{Var}[\mathbf{X}_t | \mathbf{Y}_{1:n} = \mathbf{y}_{1:n}].$$

- **Prediction**: computes for all t

$$\mathbb{E}[\mathbf{X}_{t+1} | \mathbf{Y}_{1:t} = \mathbf{y}_{1:t}] \quad \text{Var}[\mathbf{X}_{t+1} | \mathbf{Y}_{1:t} = \mathbf{y}_{1:t}].$$

The Kalman filter consists of a set of recursions that compute these quantities.

5.6 Kalman Filter

For simplicity, we assume that

$$\mathbf{d}_t = \mathbf{e}_t = \mathbf{0}$$

and assume that the processes $\{\mathbf{V}_t\}$ and $\{\mathbf{W}_t\}$ are uncorrelated, with

$$\text{Var}[\mathbf{V}_t] = \mathbf{\Sigma}_t \qquad \text{Var}[\mathbf{W}_t] = \mathbf{\Omega}_t.$$

5.6 Kalman Filter

The Kalman filter recursively computes the quantities

- $\mathbf{a}_{t|t} = \mathbb{E}[\mathbf{X}_t | \mathbf{y}_{1:t}]$,
- $\mathbf{a}_{t+1|t} = \mathbb{E}[\mathbf{X}_{t+1} | \mathbf{y}_{1:t}]$,
- $\mathbf{P}_{t|t} = \text{MSE}[\mathbf{X}_t | \mathbf{y}_{1:t}]$,
- $\mathbf{P}_{t+1|t} = \text{MSE}[\mathbf{X}_{t+1} | \mathbf{y}_{1:t}]$.

where MSE is the **mean square error** which is equivalent to the **one-step ahead prediction variance**.

5.6 Kalman Filter

Then, starting with $\mathbf{a}_{1|0}$ and $\mathbf{P}_{1|0}$, $\mathbf{a}_{t|t}$, $\mathbf{a}_{t+1|t}$ and their MSEs are obtained by running for $t = 1, \dots, n$, the recursions are

$$\mathbf{r}_t = \mathbf{y}_t - \mathbf{F}_t \mathbf{a}_{t|t-1} \quad (2a)$$

$$\mathbf{M}_t = \mathbf{F}_t \mathbf{P}_{t|t-1} \mathbf{F}_t^\top + \Sigma_t \quad (2b)$$

$$\mathbf{a}_{t|t} = \mathbf{a}_{t|t-1} + \mathbf{P}_{t|t-1} \mathbf{F}_t^\top \mathbf{M}_t^{-1} \mathbf{r}_t \quad (2c)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1} \mathbf{F}_t^\top \mathbf{M}_t^{-1} \mathbf{F}_t \mathbf{P}_{t|t-1} \quad (2d)$$

$$\mathbf{a}_{t+1|t} = \mathbf{G}_t \mathbf{a}_{t|t}, \quad (2e)$$

$$\mathbf{P}_{t+1|t} = \mathbf{G}_t \mathbf{P}_{t|t} \mathbf{G}_t^\top + \Omega_t \quad (2f)$$

5.6 Kalman Filter

Notes:

- quantities $\mathbf{r}_t$ and $\mathbf{M}_t$ in (2b) are the one-step-ahead error in forecasting $\mathbf{Y}_t$ conditional on the information set at time $t - 1$ and its MSE. This facilitates computation of the likelihood function.
- the quantities $\mathbf{a}_{t|t}$ and $\mathbf{a}_{t|t-1}$ are optimal estimators of $\mathbf{X}_t$ conditional on the available information, in terms of minimum mean square.
- the latter property holds under Gaussianity, otherwise, $\mathbf{a}_{t|t}$ and $\mathbf{a}_{t|t-1}$ are optimal only within the class of **linear** estimators (the **minimum mean square linear estimators**) of $\mathbf{X}_t$ conditional on $\mathbf{y}_{1:t}$ and $\mathbf{y}_{1:(t-1)}$.

5.6 Kalman Filter

- Equations (2c) and (2e) can be combined to yield recursions that compute the one-step-ahead prediction estimates of $\mathbf{X}_{t+1}$ given $\mathbf{y}_{1:t}$.

Similarly, by taking together (2d) and (2f), a single set of recursions for the MSE is obtained, which goes directly from $\mathbf{P}_{t|t-1}$ to $\mathbf{P}_{t+1|t}$.

- The resulting recursions, along with (2b), are

$$\mathbf{a}_{t+1|t} = \mathbf{G}_t \mathbf{a}_{t|t-1} + \mathbf{K}_t \mathbf{r}_t \quad (3a)$$

$$\mathbf{K}_t = \mathbf{G}_t \mathbf{P}_{t|t-1} \mathbf{F}_t^\top \mathbf{M}_t^{-1} \quad (3b)$$

$$\mathbf{P}_{t+1|t} = \mathbf{G}_t \mathbf{P}_{t|t-1} \mathbf{L}_t^\top + \mathbf{\Omega}_t \quad (3c)$$

$$\mathbf{L}_t = \mathbf{F}_t - \mathbf{K}_t \mathbf{G}_t. \quad (3d)$$

Parameter Estimation

The Kalman filter may be used for estimation of unknown parameters θ that appear in the system matrices.

The likelihood for data $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ can be constructed as

$$\begin{aligned} p(\mathbf{y}_1, \dots, \mathbf{y}_T) &= p(\mathbf{y}_n | \mathbf{y}_{(n-1)}) \cdots p(\mathbf{y}_2 | \mathbf{y}_1) p(\mathbf{y}_1) \\ &= \prod_{t=1}^n p(\mathbf{y}_t | \mathbf{y}_{t-1}) \end{aligned}$$

Parameter Estimation

For a Gaussian model, by taking conditional expectations on both sides of the observation equation, with $\mathbf{d}_t \equiv 0$, then

$$\mathbb{E}[\mathbf{Y}_t | \mathbf{y}_{1:(t-1)}] = \mathbf{F}_t \mathbf{a}_{t|t-1} \qquad \text{Var}[\mathbf{Y}_t | \mathbf{y}_{1:(t-1)}] = \mathbf{M}_t.$$

The one-step-ahead prediction density is a multivariate normal distribution

$$p(\mathbf{y}_t | \mathbf{y}_{1:(t-1)}) \equiv \mathcal{N}(\mathbf{F}_t \mathbf{a}_{t|t-1}, \mathbf{M}_t).$$

Parameter Estimation

Thus, the log-likelihood function is given by

$$-\frac{1}{2} \sum_{t=1}^n \log (\det \mathbf{M}_t) - \frac{1}{2} \sum_{t=1}^n \mathbf{r}_t^\top \mathbf{M}_t^{-1} \mathbf{r}_t$$

where as before

$$\mathbf{r}_t = (\mathbf{y}_t - \mathbf{F}_t \mathbf{a}_{t|t-1})$$

Typically, numerical procedures are used in order to maximize the log-likelihood to obtain the ML estimates of the parameters $\boldsymbol{\theta}$ which are consistent and asymptotically normal.

We may compute

$$\mathbf{a}_{t|n} = \mathbb{E}[\mathbf{X}_t | \mathbf{y}_{1:n}]$$

along with its MSE, $\mathbf{P}_{t|n}$ recursively, for all $t = 1, \dots, n - 1$.

The latter quantities are computed via a set of backward recursions and use the quantities $\mathbf{a}_{t+1|t}$, $\mathbf{a}_{t|t}$ and the MSE matrices $\mathbf{P}_{t|t}$, $\mathbf{P}_{t+1|t}$, which are obtained from (2).

Smoothing

In particular, to obtain $\mathbf{a}_{t|n}$ and $\mathbf{P}_{t|n}$, we start with $\mathbf{a}_{n|n}$ and $\mathbf{P}_{n|n}$ and we run backwards for $t = n - 1, \dots, 0$

$$\mathbf{a}_{t|n} = \mathbf{a}_{t|t} + \mathbf{P}_t^* (\mathbf{a}_{t+1|n} - \mathbf{a}_{t+1|t})$$

$$\mathbf{P}_{t|n} = \mathbf{P}_{t|t} + \mathbf{P}_t^* (\mathbf{P}_{t+1|n} - \mathbf{P}_{t+1|t}) \mathbf{P}_t^*$$

$$\mathbf{P}_t^* = \mathbf{P}_{t|t} \mathbf{G}_t^\top \mathbf{P}_{t+1|t}^{-1}.$$

Non-Linear Non-Gaussian Models

If the state space model is not Gaussian, the likelihood can still be constructed in the same way using the minimum mean square linear estimators of the state vector.

However, the estimators that maximize the likelihood are the **quasi-maximum likelihood** (QML) estimators of the parameters, which are also consistent and asymptotically normal.

Particle Filtering

The particle filter uses **Monte Carlo methods**, in particular **importance sampling**, to construct the approximations. If

$$\mathbb{E}_f[g(X)] = \int g(\mathbf{x})f(\mathbf{x}) \, d\mathbf{x}$$

then

$$\mathbb{E}_f[g(X)] \equiv \mathbb{E}_{f_0} \left[\frac{f(X)g(X)}{f_0(X)} \right] = \int \frac{g(\mathbf{x})f(\mathbf{x})}{f_0(\mathbf{x})} f_0(\mathbf{x}) \, d\mathbf{x}$$

for some **importance sampling** density f_0 .

Particle Filtering

We then approximate the integral by sampling $\mathbf{x}_1, \dots, \mathbf{x}_N$ from f_0 for large N , and constructing the approximation

$$\hat{\mathbb{E}}_{f_N} = \frac{1}{N} \sum_{i=1}^N \frac{g(\mathbf{x}_i) f(\mathbf{x}_i)}{f_0(\mathbf{x}_i)} = \frac{1}{N} \sum_{i=1}^N w_i g(\mathbf{x}_i),$$

say, where, if f_0 is chosen carefully

$$\hat{\mathbb{E}}_{f_N} \longrightarrow \mathbb{E}_f[g(X)]$$

as $N \longrightarrow \infty$.

Particle Filtering

If the function f is known only up to proportionality, the same procedure can work if the weights w_i are suitably normalized to sum to one, that is, if above

$$w_i = \frac{f(\mathbf{x}_i)}{f_0(\mathbf{x}_i)}$$

is replaced by

$$w_i^* = \frac{w_i}{W} \quad \text{with} \quad W = \sum_{i=1}^n w_i.$$

as W approximates the integral

$$\int f(\mathbf{x}) \, d\mathbf{x} = \int \frac{f(\mathbf{x})}{f_0(\mathbf{x})} f_0(\mathbf{x}) \, d\mathbf{x}$$

Particle Filtering

Given a non-linear and non-Gaussian state space model, the goal is to compute the filtering and prediction densities

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) \quad p(\mathbf{x}_{t+1} | \mathbf{y}_{1:t})$$

at each time t via a Monte Carlo approximation.

We aim to maintain a sample of **particles**

$$(\mathbf{x}_{t1}, \dots, \mathbf{x}_{tN})$$

for each t that can be used to approximate these distributions.

Particle Filtering

At each time point t , we perform the

Sequential Importance Sampling

(or SIS) procedure:

Particle Filtering

- Draw random samples from a chosen importance density that can be sampled directly.
- Compute a normalized weight for each particle
- Approximate the target density at time t by the weighted sum of the particles.

Given sufficiently large numbers of particles, this IS characterization converges to the target density.

Particle Filtering

In the filtering problem, we approximate $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ by

- particles $\mathbf{x}_t = (x_{t1}, \dots, x_{tN})$, sampled from density f_0 .
- normalized weights $\mathbf{w}_t^* = (w_{t1}^*, \dots, w_{tN}^*)$

via the discrete distribution $\{(w_{ti}^*, x_{ti}), i = 1 \dots, N\}$.

Particle Filtering

Two key questions are

- What importance density f_0 can or should be chosen?
- How do we compute the weights efficiently (in a recursive fashion) such that for a given weight w_{ti}^* , we just need a new observation $\mathbf{y}_t$ to get an approximation to $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

A simple but effective choice for f_0 sampling density at time t is the **transition density**: $f_0(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t}) \equiv p(\mathbf{x}_t|\mathbf{x}_{t-1})$ as then

$$w_{ti}^* \propto w_{t-1,i}^* p(\mathbf{y}_t|\mathbf{x}_{ti}^*).$$

Particle Degeneracy

A problem with sequential importance sampling is that typically, after a few iterations, all but one particle have negligible weights. This phenomenon is termed **degeneracy**.

The degree of degeneracy is represented by the quantity

$$N_t = \left(\sum_{i=1}^N (w_{ti}^*)^2 \right)^{-1}$$

which is the **effective sample size**.

Particle Degeneracy

Possible solutions to overcome degeneracy effects:

- brute force: very large N
- optimized choice of importance density
- resampling

Resampling

The goal of resampling is to eliminate particles with small weights and replace by new particles drawn from the vicinity of particles with larger weights

- concentrate on particles with large weights
- model important parts of the distribution more precisely

Various efficient algorithms exist for resampling.

- eg **Sampling Importance Resampling (SIR)**

5.7 Hidden Markov Models

If the unobserved state X_t is **discrete**, then state space models are often called **hidden Markov models** (HMMs).

In this case, the state equation is replaced by a **Markov chain** model: if $\{X_t\}$ can take values on $\{1, 2, \dots, S\}$ say, then for $t > 1$

$$p(x_t | x_{1:(t-1)}) \equiv p(x_t | x_{t-1})$$

where

$$\Pr[X_t = l | X_{t-1} = k] = p_{kl}$$

The $S \times S$ transition matrix $\mathbf{P}$, along with the (marginal) distribution of X_1 ($\Pr[X_1 = k], k = 1, \dots, S$) characterize the nature of the state evolution. Note that $\mathbf{P}$ is a **stochastic matrix** whose rows sum to 1.

5.7 Hidden Markov Models

All of the previous types of calculation

- filtering $p(\mathbf{x}_t | \mathbf{y}_{1:t})$
- prediction $p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})$
- smoothing $p(\mathbf{x}_t | \mathbf{y}_{1:n})$
- parameter estimation

are required. The main difference is that the filtering, prediction and smoothing distributions are **discrete**, and therefore somewhat more straightforward to compute.

The observation variable y_t can be either discrete or continuous.

5.7 Hidden Markov Models

Example: $S = 2$

Suppose for $t = 1, 2, \dots, n$

$$Y_t = \mu(X_t) + \epsilon_t$$

and $\{X_t\}$ follows a Markov chain with transition matrix

$$\mathbf{P} = \begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}$$

and the distribution of X_1 is the stationary (or equilibrium) distribution, π , of the chain defined as the solution of

$$\pi = \pi \mathbf{P}$$

5.7 Hidden Markov Models

Example: $S = 2$

We compute π (a 1×2 vector) by

$$[1, 1]^T \pi = \lim_{t \rightarrow \infty} \mathbf{P}^t = \begin{bmatrix} 4/11 & 7/11 \\ 4/11 & 7/11 \end{bmatrix}$$

so $\pi = (4/11, 7/11)$.

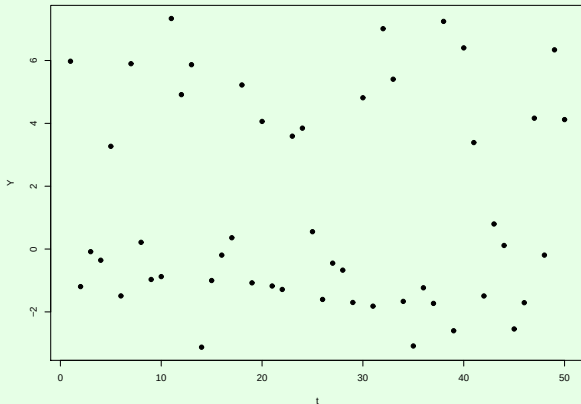
Suppose

$$\mu(\mathbf{x}) = \begin{cases} \mu_1 & \mathbf{x} = 1 \\ \mu_2 & \mathbf{x} = 2 \end{cases}$$

and $\{\epsilon_t\} \sim \mathcal{N}(0, 1)$, IID.

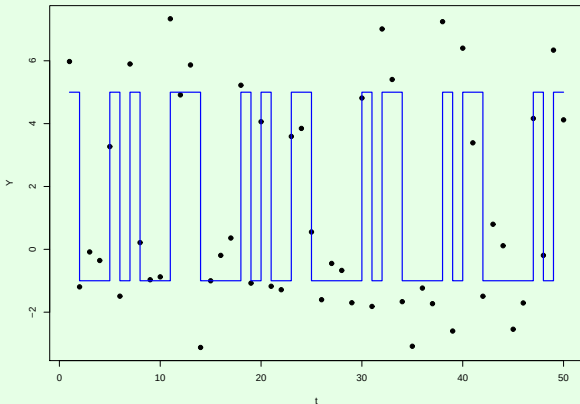
5.7 Hidden Markov Models

Example: $S = 2$



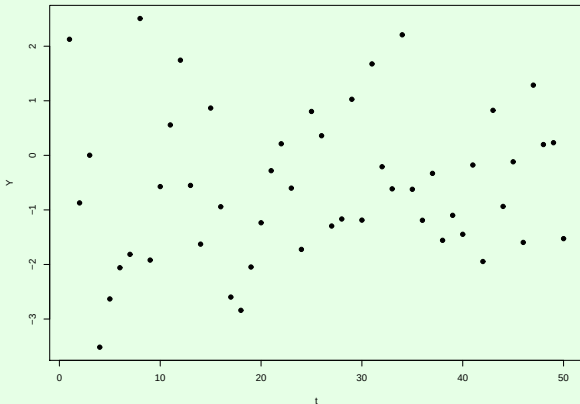
5.7 Hidden Markov Models

Example: $S = 2, \mu_1 = 5, \mu_2 = -1$



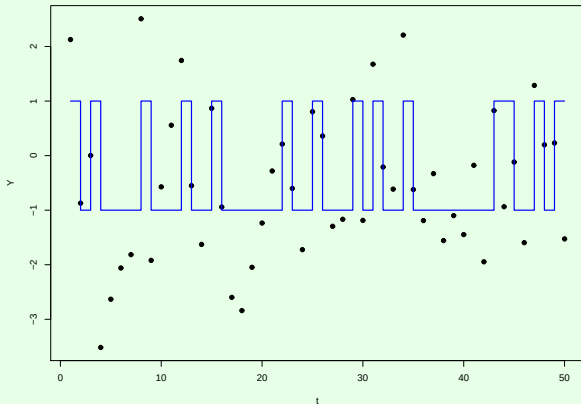
5.7 Hidden Markov Models

Example: $S = 2$



5.7 Hidden Markov Models

Example: $S = 2, \mu_1 = 1, \mu_2 = -1$



5.7 Hidden Markov Models

Filtering: By Bayes theorem

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{y}_{1:t}) &= p(\mathbf{x}_t | \mathbf{y}_t, \mathbf{y}_{1:(t-1)}) \\ &= \frac{p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{1:(t-1)}) p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})}{p(\mathbf{y}_t | \mathbf{y}_{1:(t-1)})} \\ &= \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})}{\sum_{\mathbf{x}} p(\mathbf{y}_t | \mathbf{x}) p(\mathbf{x} | \mathbf{y}_{1:(t-1)})} \\ &\propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)}) \end{aligned}$$

This is a discrete distribution on $\{1, \dots, S\}$ that we can compute exactly.

5.7 Hidden Markov Models

Prediction: By the theorem of total probability

$$p(\mathbf{x}_{t+1} | \mathbf{y}_{1:t}) = \sum_{\mathbf{x}=1}^S p(\mathbf{x}_{t+1} | \mathbf{x}) p(\mathbf{x} | \mathbf{y}_{1:t})$$

- $p(\mathbf{x}_{t+1} | \mathbf{x})$ is determined the x th row of $\mathbf{P}$, for each $x = 1, 2, \dots, S$
- $p(\mathbf{x} | \mathbf{y}_{1:t})$ is the previous filtering distribution.

5.7 Hidden Markov Models

Smoothing: By the theorem of total probability, we can compute **backwards** $t = n, n - 1, \dots, 1$,

$$p(\mathbf{x}_t | \mathbf{y}_{1:n}) = \sum_{\mathbf{x}_{t+1}=1}^S p(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{y}_{1:n}) p(\mathbf{x}_{t+1} | \mathbf{y}_{1:n})$$

$p(\mathbf{x}_{t+1} | \mathbf{y}_{1:n})$ is the previous smoothing distribution.

5.7 Hidden Markov Models

For the first term

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{x}_{t+1}, y_{1:n}) &\equiv p(\mathbf{x}_t | \mathbf{x}_{t+1}, y_{1:t}) \\ &= \frac{p(\mathbf{x}_t | y_{1:t}) p(\mathbf{x}_{t+1} | \mathbf{x}_t, y_{1:t})}{p(\mathbf{x}_{t+1} | y_{1:t})} \\ &= \frac{p(\mathbf{x}_t | y_{1:t}) p(\mathbf{x}_{t+1} | \mathbf{x}_t)}{p(\mathbf{x}_{t+1} | y_{1:t})} \end{aligned}$$

which can be computed from the filtering and prediction distributions.

Discrete HMMs

The HMM structure is commonly implemented when the observation is also **discrete**.

As usual let $x_{1:n}$ denote the **hidden** state sequence, and $y_{1:n}$ denote the **observed** data sequence.

Discrete HMMs

Then the model is specified by

- **transition probabilities**

$$p_{kl} = \Pr[X_t = l | X_{t-1} = k]$$

with initial distribution $\Pr[X_1 = k]$, $k = 1, \dots, S$.

- **emission probabilities**

$$e_k(m) = \Pr[Y_t = m | X_t = k]$$

$e_k(m)$ can be specified for a finite set ($m \in \{1, 2, \dots, M\}$), or on the integers.

Discrete HMMs: Inference Tasks

We seek the **most probable** state path, that is, the most probable sequence of states in light of the data:

$$\begin{aligned} \mathbf{x}_{1:n}^* &= \arg \max_{\mathbf{x}_{1:n}} p(\mathbf{x}_{1:n}, y_{1:n}) \\ &= \arg \max_{\mathbf{x}_{1:n}} p(\mathbf{x}_{1:n} | y_{1:n}) \end{aligned}$$

- $p(\mathbf{x}_{1:n}, y_{1:n})$ is the joint probability or joint likelihood
- $p(\mathbf{x}_{1:n} | y_{1:n})$ is the posterior probability

Discrete HMMs: The Viterbi Algorithm

- (i) Initialize vector $v_k(0) = \Pr[X_1 = k]$, $k = 1, \dots, S$.
- (ii) Recursion: for $t = 1, \dots, n$, compute

$$v_l(t) = e_l(y_t) \max_k (v_k(t-1) p_{kl})$$
$$point_t(l) = \arg \max_k (v_k(t-1) p_{kl})$$

for each $l = 1, 2, \dots, S$

Discrete HMMs: The Viterbi Algorithm

(iii) Termination

$$p(y_{1:n}, \mathbf{x}_{1:n}^*) = \max_k (v_k(n))$$
$$\mathbf{x}_n^* = \arg \max_k (v_k(n))$$

(iv) Trace back the optimal path

$$\mathbf{x}_{t-1}^* = \textit{point}_t(\mathbf{x}_t^*) \quad t = n, n-1, \dots, 2.$$

Discrete HMMs: The Forward Algorithm

The probability of an observed sequence (or likelihood) is

$$p(y_{1:n}) = \sum_{x_{1:n}} p(y_{1:n}, x_{1:n})$$

However, the direct computation of this sum is not possible, so a recursive calculation is needed.

Discrete HMMs: The Forward Algorithm

- (i) Initialize vector $f_k(1) = e_k(y_1)\pi_k$ for $k = 1, 2, \dots, S$
- (ii) Recursion: for $t = 2, \dots, n$, and $l = 1, 2, \dots, S$,

$$f_l(t) = e_l(y_t) \sum_{k=1}^S f_k(t-1) p_{kl}$$

- (iii) Final calculation:

$$p(y_{1:n}) = \sum_{k=1}^S f_k(n).$$

Discrete HMMs: The Backward Algorithm

We seek the posterior (smoothing) probabilities

$$\begin{aligned} p(X_t = k | y_{1:n}) &= \frac{p(X_t = k, y_{1:n})}{p(y_{1:n})} \\ &= \frac{p(y_1, \dots, y_t, X_t = k) p(y_{t+1}, \dots, y_n | X_t = k)}{p(y_{1:n})} \end{aligned}$$

for $k = 1, 2, \dots, S$.

Direct computation is not possible, so a recursive calculation is needed, after completing the Forward Algorithm.

Discrete HMMs: The Backward Algorithm

The backward algorithm works in reverse order through the data:

- (i) Initialize element $b_k(n) = f_k(n)$ for $k = 1, \dots, S$.
- (ii) Backwards recursion: for $t = n - 1, \dots, 1$ compute

$$b_k(t) = \sum_{l=1}^S p_{kl} e_l(y_{t+1}) b_l(t+1)$$

for $k = 1, 2, \dots, S$.

Discrete HMMs: The Backward Algorithm

The forward and backward probabilities, $f_k(t)$ and $b_k(t)$, are in fact the probabilities

$$f_k(t) = p(y_1, \dots, y_t, X_t = k)$$

$$b_k(t) = p(y_{t+1}, \dots, y_n | X_t = k)$$

Thus we have

$$p(X_t = k, y_{1:n}) = f_k(t)b_k(t)$$

and hence, for $k = 1, \dots, S$,

$$p(X_t = k | y_{1:n}) = \frac{f_k(t)b_k(t)}{p(y_{1:n})}$$

Discrete HMMs: Estimation

The algorithms above are all expressed in terms of known transition and emission probabilities. These probabilities can be pre-specified, estimated from observed sequences, or estimated during the state sequence reconstruction.

If the state sequence is **known**, let n_{kl} be the number of transitions from state k to state l , and let $r_k(b)$ be the number of times character b is emitted from state k . Then we may estimate the transition and emission probabilities as follows

$$\hat{p}_{kl} = \frac{n_{kl}}{\sum_{l'} n_{kl'}} \quad \hat{e}_k(b) = \frac{r_k(b)}{\sum_{b'} r_k(b')} \quad (4)$$

The Baum-Welch Algorithm

The Baum-Welch algorithm estimates n_{kl} and $r_k(b)$ as the **expected** number of transitions and emissions, which are computed using the forward and backward algorithms.

The Baum-Welch Algorithm

The probability of a transition $k \rightarrow l$ at time t is

$$p(X_t = k, X_{t+1} = l | y_{1:n}) = \frac{f_k(t) p_{kl} e_l(y_{t+1}) b_l(t+1)}{p(y_{1:n})}$$

Suppose that N replicate observed sequences

$$y_{1:n}^1, \dots, y_{1:n}^N$$

are available. Then the expected number of times this transition will be observed (in light of the data) is

$$n_{kl} = \sum_{i=1}^N \left\{ \frac{\sum_t f_k^i(t) a_{kl} e_l(y_{t+1}^i) b_l^i(t+1)}{p(y_{1:n}^i)} \right\}$$

The Baum-Welch Algorithm

In this formula

- $y_{1:n}^i$ is the i^{th} observed sequence
- $f_k^i(t)$ is the forward probability from the forward algorithm implemented for sequence i
- $b_l^i(t)$ is the backward probability from the backward algorithm implemented for sequence i

Similarly,

$$r_k(b) = \sum_{i=1}^N \left\{ \frac{\sum_{t \in \mathcal{B}_i} f_k^i(t) b_l^i(t)}{p(y_{1:n}^i)} \right\}$$

where $\mathcal{B}_i$ is the set of indices of locations in replicate sequence i where character b is emitted.

The Baum-Welch Algorithm

- (i) Initialize parameter values to some arbitrary values
- (ii) Recursion: Set $n_{kl}^{(0)}, r_k^{(0)}(b)$ to be some small count value, then for $i = 1, \dots, N$: calculate $f_k^i(t)$ using the Forward Algorithm, and $b_k^i(t)$ using the Backward Algorithm, then set

$$n_{kl}^{(i)} = n_{kl}^{(i-1)} + \frac{1}{p(y_{1:n}^i)} \left\{ \sum_{t=1}^n f_k^i(t) p_{kl} e_l(y_{t+1}^i) b_l^i(t+1) \right\}$$

$$r_k^{(i)}(b) = r_k^{(i-1)}(b) + \frac{1}{p(y_{1:n}^i)} \left\{ \sum_{t \in \mathcal{B}_i} f_k^i(t) b_k^i(t) \right\}$$

The Baum-Welch Algorithm

- (iii) Estimate p_{kl} and $e_k(b)$ using (4), and compute the likelihood

$$\prod_{i=1}^N p(y_{1:n}^i)$$

- (iv) Terminate the recursion if the computed likelihood value is virtually unchanged compared to the previous iteration.

Computation for continuous response HMMs

Note that most of the above calculations carry through to the continuous emission case, that is if the emission probabilities

$$e_k(m) = \Pr[Y_t = m | x_t = k]$$

are replaced by densities

$$e_k(y) = p(y | X_t = k)$$

The forward and backward quantities become unnormalized joint densities

Computation for continuous response HMMs

To avoid underflow in the computation, the computations can either be computed on the log scale, or with continual rescaling of the calculation. For example, in the Viterbi algorithm, the computation of the $v_k(t)$ quantities will underflow quickly without rescaling, but rescaling by

$$c_t = \sum_{k=1}^S v_k(t)$$

will prevent this.

The constants c_t should be stored for future use; if the forward and backward probabilities are computed after rescaling, then this rescaling must be undone when, for example, $p(y_{1:n})$ is computed.