Partially observed models

Erik Lindström

FMS161/MAFM18 Financial Statistics
Motivation

- Used when regressors are unobservable

- Eg. when the regressor dimension is larger than the observable state dimension (think stochastic volatility).

- Or interest rate models

- Or credit models (hidden jump intensity process)

- Missing observations can be treated in this framework.
Motivation

- Used when regressors are unobservable
- Eg. when the regressor dimension is larger than the observable state dimension (think stoch. vol).

Erik Lindström
Partially observed models
Motivation

▶ Used when regressors are unobservable
▶ Eg. when the regressor dimension is larger than the observable state dimension (think stoch. vol).
▶ or interest rate models
Motivation

- Used when regressors are unobservable
- Eg. when the regressor dimension is larger than the observable state dimension (think stoch. vol).
- or interest rate models
- or credit models (hidden jump intensity process)
Motivation

- Used when regressors are unobservable
- Eg. when the regressor dimension is larger than the observable state dimension (think stoch. vol).
- or interest rate models
- or credit models (hidden jump intensity process)
- Missing observations can be treated in this framework.
Examples

- Stochastic volatility

\[ y_t = \sigma_t \eta_t \] (1)
\[ \log \sigma_t^2 = a_0 + a_1 \log \sigma_{t-1}^2 + e_t \] (2)
Examples

- **Stochastic volatility**

  \[ y_t = \sigma_t \eta_t \quad (1) \]

  \[ \log \sigma_t^2 = a_0 + a_1 \log \sigma_{t-1}^2 + \epsilon_t \quad (2) \]

- **Short rate models**

  \[ dr_t = \alpha(\beta - r_t)dt + \sqrt{\gamma + \delta r_t}dW_t \quad (3) \]

  \[ P(t, T) = A(t, T)e^{-B(t, T)r_t} \quad (4) \]
Let us start with the stoch vol. model.

\[ y_t = \sigma_t \eta_t \]  \hspace{1cm} (5)
\[ \log \sigma_t^2 = a_0 + a_1 \log \sigma_{t-1}^2 + e_t \]  \hspace{1cm} (6)
Let us start with the stoch vol. model.

\[ y_t = \sigma_t \eta_t \]  \hspace{1cm} (5)

\[ \log \sigma_t^2 = a_0 + a_1 \log \sigma_{t-1}^2 + e_t \]  \hspace{1cm} (6)

▷ \( \sigma_t^2 \) is not directly observable
Let us start with the stoch vol. model.

\[ y_t = \sigma_t \eta_t \quad (5) \]

\[ \log \sigma_t^2 = a_0 + a_1 \log \sigma_{t-1}^2 + e_t \quad (6) \]

- \( \sigma_t^2 \) is not directly observable
- but can be estimated.
- Likelihood

\[ p(y_1, \ldots, y_T) = \int p(\sigma_1, y_1, \ldots, \sigma_T, y_T) d\sigma_{1:T}? \]
Let us start with the stoch vol. model.

\begin{align*}
y_t &= \sigma_t \eta_t \quad (5) \\
\log \sigma_t^2 &= a_0 + a_1 \log \sigma_{t-1}^2 + e_t \quad (6)
\end{align*}

- \( \sigma_t^2 \) is not directly observable
- but can be estimated.

- Likelihood

\[
p(y_1, \ldots, y_T) = \int p(\sigma_1, y_1, \ldots, \sigma_T, y_T) d\sigma_{1:T}?
\]

- Dependence structure?
General state space models

All models we use can be written in general state space form

\[
\begin{align*}
    y_t &= h(x_t, \eta_t) \\
    x_t &= f(x_{t-1}, e_t)
\end{align*}
\]

- \(x\) is a hidden (unobservable) Markov process (cf. HMM)
- \(y\) is observed.
- \(y_t|x_t\) is independent of \(y_s, s = 1..t - 1, t + 1 .. T\).
- These rather simple structures can generate complex models!
All models we use can be written in state space form

\[ y_t = h(x_t, \eta_t) \]  
\[ x_t = f(x_{t-1}, e_t) \]

These equations imply transition probabilities, i.e. we can derive \( p(x_t|x_{t-1}) \) and \( p(y_t|x_t) \) from the model setup.

We also need \( p(x_0) \), i.e. initial conditions.
The likelihood can be written as

\[ p(y_1, \ldots, y_T) = p(y_1) \prod_{t=2}^{T} p(y_t | y_{1:t-1}), \]

where \( y_{1:t-1} \) is shorthand notation for \( \{y_1, \ldots, y_{t-1}\} \).
Likelihood

The likelihood can be written as

\[ p(y_1, \ldots, y_T) = p(y_1) \prod_{t=2}^{T} p(y_t|y_{1:t-1}), \]

where \( y_{1:t-1} \) is shorthand notation for \( \{y_1, \ldots, y_{t-1}\} \).

We can write

\[ p(y_t|y_{1:t-1}) = \int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t \]

and

\[ p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \]
The density for the hidden state $x_t$, using the information $y_{1:t}$ is called the *filter density*, $p(x_t|y_{1:t})$. 

We can derive the filter density from $p(x_t|y_{1:t}) = p(y_t|x_t) p(x_t|y_{1:t-1}) p(y_{1:t-1}|y_{1:t-1})$. 

or equivalently $p(x_t|y_{1:t}) = p(y_t|x_t) p(x_t|y_{1:t-1}) \int p(y_{1:t-1}|y_{1:t-1}) dx_t$. 

Filter density
The density for the hidden state $x_t$, using the information $y_{1:t}$ is called the \textit{filter density}, $p(x_t|y_{1:t})$.

We can derive the filter density from

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}.$$  

or equivalently

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t}.$$
The density for the hidden state $x_t$, using the information $y_{1:t-1}$ is called the \textit{predictive density}, $p(x_t|y_{1:t-1})$. 
The density for the hidden state $x_t$, using the information $y_{1:t-1}$ is called the **predictive density**, $p(x_t|y_{1:t-1})$.

We can derive the predictive density from

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}.$$
Recursion

1. We have the filter density \( p(x_0) \) at time 0.
2. At time \( t \), generate the predictive density \( p(x_{t+1}|y_{1:t}) \).
3. At time \( t + 1 \), calculate \( p(y_t|y_{1:t-1}) \) and update the filter density \( p(x_{t+1}|y_{1:t+1}) \). Repeat from step 2.
Essentially there are only 2 recursions known in closed form

- HMM (finite state space)
- Kalman filter (linear, Gaussian models)
Kalman filter

Why does it give closed form recursions?

**Short answer:** The Gaussian density is an exponential, second order polynomial.

Model:

\[
Y_t = CX_t + \eta_t, \quad \eta_t \in N(0, \Gamma)
\]

\[
X_t = AX_{t-1} + e_t, \quad e_t \in N(0, \Sigma)
\]
Kalman filter

Why does it give closed form recursions?

**Short answer:** The Gaussian density is an exponential, second order polynomial.

Model:

\[
Y_t = CX_t + \eta_t, \quad \eta_t \in \mathcal{N}(0, \Gamma)
\]

\[
X_t = AX_{t-1} + e_t, \quad e_t \in \mathcal{N}(0, \Sigma)
\]

[1] Assume initial distribution

\[
p(x_0|\mathcal{F}_0) = \phi(x_0; m_0, P_0)
\]
[2] Calculate the predictive density

\[ p(x_1|\mathcal{F}_0) = \int p(x_1|x_0)p(x_0|\mathcal{F}_0)\,dx_0 \]

Here \( p(x_1|x_0) = \phi(x_1; Ax_0, \Sigma) \), thus giving

\[ \propto \int e^{-\frac{1}{2}(x_1-Ax_0)^T \Sigma^{-1} (x_1-Ax_0)} e^{-\frac{1}{2}(x_0-m_0)^T P_0^{-1} (x_0-m_0)}\,dx_0. \]
[2] Calculate the predictive density

\[ p(x_1|\mathcal{F}_0) = \int p(x_1|x_0)p(x_0|\mathcal{F}_0)dx_0 \]

Here \( p(x_1|x_0) = \phi(x_1; Ax_0, \Sigma) \), thus giving

\[ \propto \int e^{-\frac{1}{2}(x_1-Ax_0)^T\Sigma^{-1}(x_1-Ax_0)} e^{-\frac{1}{2}(x_0-m_0)^TP_0^{-1}(x_0-m_0)}dx_0. \]

Some calculations give

\[ = \phi(x_1; Am_0, AP_0A^T + \Sigma). \]
[3] The filter density is more complicated. We have

\[ p(x_t | y_{1:t}) = \frac{p(y_t | x_t)p(x_t | y_{1:t-1})}{\int p(y_t | x_t)p(x_t | y_{1:t-1})dx_t}. \]

Thus

\[ p(x_1 | y_1) = \frac{p(y_1 | x_1)p(x_1 | \mathcal{F}_0)}{p(y_1 | \mathcal{F}_0)} \propto p(y_1 | x_1)p(x_1 | \mathcal{F}_0). \]
The filter density is more complicated. We have

\[
p(x_t | y_{1:t}) = \frac{p(y_t | x_t)p(x_t | y_{1:t-1})}{\int p(y_t | x_t)p(x_t | y_{1:t-1}) \, dx_t}.
\]

Thus

\[
p(x_1 | y_1) = \frac{p(y_1 | x_1)p(x_1 | \mathcal{F}_0)}{p(y_1 | \mathcal{F}_0)} \propto p(y_1 | x_1)p(x_1 | \mathcal{F}_0).
\]

Note that the likelihood is a normalization, and independent of \(x_1\).
Tedious calculations give

\[ p(x_1|y_1) = \phi(x_1; m_1|0 + K_1(y_1 - Cm_1|0), P_1|0 - K_1 CP_1|0), \]

where

\[
\begin{align*}
\text{var}(\eta) & = \Gamma \\
m_1|0 & = Am_0 \\
P_1|0 & = AP_0 A^T + \Sigma \\
\Omega & = CP_1|0 C^T + \Gamma \\
K_1 & = P_1|0 C^T \Omega^{-1}
\end{align*}
\]

Still Gaussian!
Non-linear models

Approximate non-linear models with a local linear model.
- EKF
- Sigma point filters
- Ensemble filters

Or use Monte Carlo methods (will cover this later)
- Particle filters
- Hybrid particle filters
Extended Kalman filter (EKF)

Use Taylor expansions to approximate the non-linear model with a linear model.

\[ y_{t+1} = h(x_t, \eta_{t+1}) \quad (11) \]
\[ x_{t+1} = f(x_t, e_{t+1}) \quad (12) \]

▶ Prediction

\[ m_{t+1|t} = f(m_{t|t}, 0), \]
\[ P_{t+1|t} = F_t P_{t|t} F_t^T + \Sigma, \]
\[ F_t = f'_x(x_t, 0). \]
Extended Kalman filter (EKF)

While the filtering is given by:

 Filtering

\[
\begin{align*}
m_{t+1|t+1} &= m_{t+1|t} + K_t(y_{t+1} - h(m_{t+1|t}, 0)) \\
P_{t+1|t+1} &= P_{t+1|t} - K_t H_t P_{t+1|t},
\end{align*}
\]

where

\[
\begin{align*}
\Omega_t &= H_t P_{t+1|t} H_t^T + \Gamma \\
K_t &= P_{t+1|t} H_t^T \Omega_t^{-1} \\
H_t &= h'_x(x_t, 0)
\end{align*}
\]

Even better is Iterated EKFs, by reinterpreting the problem as an optimization of the log-posterior.
Other filters

Alternatives include:

- Iterated Kalman filters improve the quality of the linearizations.
Other filters

Alternatives include:

- Iterated Kalman filters improve the quality of the linearizations.

- Sigma point filters (UKF) use clearly sampled points to derive the first and second moment.
Other filters

Alternatives include:

- Iterated Kalman filters improve the quality of the linearizations.
- Sigma point filters (UKF) use clearly sampled points to derive the first and second moment.
- Ensemble filters (EnKF) use Monte Carlo methods to approximate the first and second central moments.
Other filters

Alternatives include:

- Iterated Kalman filters improve the quality of the linearizations.
- Sigma point filters (UKF) use clearly sampled points to derive the first and second moment.
- Ensemble filters (EnKF) use Monte Carlo methods to approximate the first and second central moments.
- These are in general more accurate than the EKF.
The most common method for calibrating options to market data today is some non-linear weighted least squares estimator

$$\theta = \arg \min \sum_i \lambda_i \left( c_{\text{Market}}^t (S_i, K_i, r_i, \tau_i) - c_{\text{Model}}^t (S_i, K_i, r_i, \tau_i; \theta) \right)^2$$  \hspace{1cm} (13)

where $c_{\text{Market}}^t (S_i, K_i, r_i, \tau_i)$ are the market price that depends on the underlying asset $S_i$, strike level $K_i$, interest rate $r_i$ and time to maturity $\tau_i$ and $\lambda_i$ are weights.
Case: Calibration of options

The most common method for calibrating options to market data today is some non-linear weighted least squares estimator

$$\theta = \arg \min \sum \lambda_i \left( c_t^{Market}(S_i, K_i, r_i, \tau_i) - c_t^{Model}(S_i, K_i, r_i, \tau_i; \theta) \right)^2$$ (13)

where \( c_t^{Market}(S_i, K_i, r_i, \tau_i) \) are the market price that depends on the underlying asset \( S_i \), strike level \( K_i \), interest rate \( r_i \) and time to maturity \( \tau_i \) and \( \lambda_i \) are weights.

There are two main (implicitly related) problems with this approach:

- The parameter estimates are noisy,
- Old data is typically discarded, as only the most recent data is used.
Simulated data from the Bates (1996) model

Figure: WLS estimates (red circles) and the true parameters (solid blue line). The WLS works most of the time...
(Lindstrom et al, 2008) rewrites the calibration problems as a filtering problem, augmenting the latent states with the parameter vector

\[ c_t^{\text{Market}}(S_n, K_i, r_i, \tau_i) = c_t^{\text{Model}}(S_n, K_i, r_i, \tau_i; \theta_n) + \eta_n, \quad (14) \]

\[ \theta_n = \theta_{n-1} + e_n. \quad (15) \]

This decomposes the change of the option prices into changes in the underlying state variables (i.e. the index level), changes in the parameters (which is captured by the random walk dynamics) and pure noise due to the ask-bid spread.
Figure: WLS estimates (red circles), the true parameters (solid blue line) and filter estimates (black dots).
We have successfully used

- The EKF (not optimal though...)
- The Iterated EKF (Lindström et. al, 2008)
- The UKF (Wiktorsson & Lindström, 2014)
- The EnKF (Lindström & Guo, 2013)
Filters

We have successfully used

- The EKF (not optimal though...)
- The Iterated EKF (Lindström et. al, 2008)
- The UKF (Wiktorsson & Lindström, 2014)
- The EnKF (Lindström & Guo, 2013)

However, (simple) Monte Carlo filters does not work very well for this problem.
The quadratic hedging problem was given by

\[(\hat{\alpha}, \hat{\beta}) = \arg\min \mathbb{E} \left[ \left( \pi(S(T) - \alpha S(T) - \beta B(T)) \right)^2 \bigg| \mathcal{F}(0) \right]. \tag{16}\]
The quadratic hedging problem was given by

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha, \beta} \mathbb{E} \left[ (\pi(S(T) - \alpha S(T) - \beta B(T))^2 \mid \mathcal{F}(0) \right].$$  

(16)

The solution to this problem is given by the means, variances and covariances between these assets.
The quadratic hedging problem was given by

\( (\hat{\alpha}, \hat{\beta}) = \arg\min \mathbb{E} \left[ (\pi(S(T) - \alpha S(T) - \beta B(T))^2 \middle| \mathcal{F}(0) \right]. \) (16)

The solution to this problem is given by the means, variances and covariances between these assets. Most/All of these are computed in the filter (Lindström & Guo, 2013)!
Proof

Check the derivations!
Proof

Check the derivations!
Example (EKF).

\[ h(m_{t+1}|t,0) \] - Predicted mean \hspace{1cm} (17)
\[ \Omega_t \] - Covariance of the assets \hspace{1cm} (18)

This works if the model is modified by also including the underlying asset \( S \) into the model.

\[ \text{Market}_t(S_n, K_i, r_i, \tau_i) = \text{Model}_t(S_n, K_i, r_i, \tau_i; \theta_n) + \eta_n(S_n, n), (19) \]

\[ S_{\text{Market}} = S_n + \eta(S_n), (20) \]

\[ \theta_n = \theta_{n-1} + e_n. (21) \]
Proof

Check the derivations!
Example (EKF).

\[ h(m_{t+1}|t,0) \text{- Predicted mean} \]
\[ \Omega_t \text{- Covariance of the assets} \]

This works if the model is modified by also including the underlying asset \( S \) into the model.

\[ c_t^{\text{Market}}(S_n, K_i, r_i, \tau_i) = c_t^{\text{Model}}(S_n, K_i, r_i, \tau_i; \theta_n) + \eta_n^{(c)}, \]  
\[ S_t^{\text{Market}} = S_n + \eta_n^{(S)} \]
\[ \theta_n = \theta_{n-1} + e_n. \]