Time Series Analysis

4. State space models and Kalman filtering

Andrew Lesniewski

Baruch College
New York

Fall 2023

Outline

1. Warm-up: Recursive Least Squares Regression
2. Kalman Filter
3. Nonlinear State Space Models
4. Particle Filtering

OLS regression

- As a motivation for the reminder of this lecture, we consider the standard linear model

$$Y = X^{\mathrm{T}}\beta + \varepsilon, \tag{1}$$

  where $Y \in \mathbb{R}$, $X \in \mathbb{R}^k$, and $\varepsilon \in \mathbb{R}$ is noise.
- This includes the model with an intercept as a special case in which the first component of X is assumed to be 1.
- Given n observations $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of X and Y, respectively, the ordinary least square least (OLS) regression leads to the following estimated value of the coefficient β:

$$\widehat{\beta}_n = (\mathcal{X}_n^{\mathrm{T}}\mathcal{X}_n)^{-1}\mathcal{X}_n^{\mathrm{T}}\mathcal{Y}_n. \tag{2}$$

OLS regression

- The matrices $\mathcal{X}$ and $\mathcal{Y}$ above are defined as

$$\mathcal{X} = \begin{pmatrix} x_1^{\mathrm{T}} \\ \vdots \\ x_n^{\mathrm{T}} \end{pmatrix} \in \mathrm{Mat}_{\mathrm{n,k}}(\mathbb{R}) \tag{3}$$

and

$$\mathcal{Y}_n = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \tag{4}$$

respectively.

Recursive least squares regression

- Suppose now that X and Y consists of a streaming set of data, and each new observation leads to an updated value of the estimated β.
- For large data sets it may be suboptimal to redo the entire calculation above after a new observation arrives in order to find the updated value.
- Namely, this calculation may lead to both large memory and large storage requirements.
- Instead, we shall derive a recursive algorithm that stores the last calculated value of the estimated β, and updates it to incorporate the impact of the latest observation.
- To this end, we introduce the notation:

$$\begin{aligned} P_n &= (\mathcal{X}_n^{\mathrm{T}} \mathcal{X}_n)^{-1}, \\ B_n &= \mathcal{X}_n^{\mathrm{T}} \mathcal{Y}_n, \end{aligned} \tag{5}$$

and rewrite (2) as

$$\widehat{\beta}_n = P_n B_n. \tag{6}$$

Recursive least squares regression

- Using (3) and (4), we can write

$$\mathcal{X}_{n+1} = \begin{pmatrix} \mathcal{X}_n \\ x_{n+1}^{\mathrm{T}} \end{pmatrix},$$
$$\mathcal{Y}_{n+1} = \begin{pmatrix} \mathcal{Y}_n \\ y_{n+1} \end{pmatrix}.$$

- Therefore, P_n and B_n obey the following recursive relations:

$$P_{n+1}^{-1} = P_n^{-1} + x_{n+1} x_{n+1}^{\mathrm{T}},$$
$$B_{n+1} = B_n + x_{n+1} y_{n+1}.$$

Recursive least squares regression

- We shall now invoke the *matrix inversion formula*:

$$(A+BD)^{-1} = A^{-1} - A^{-1}B(I+DA^{-1}B)^{-1}DA^{-1}, \tag{7}$$

  valid for a square invertible matrix A, and matrices B and D such that the operations above are defined.

- *Proof:*

$$\begin{aligned}
&(A+BD)(A^{-1} - A^{-1}B(I+DA^{-1}B)^{-1}DA^{-1}) \\
&\qquad = I - B(I+DA^{-1}B)^{-1}DA^{-1} + BDA^{-1} - BDA^{-1}B(I+DA^{-1}B)^{-1}DA^{-1} \\
&\qquad = I - B\Big((I+DA^{-1}B)^{-1} - I + DA^{-1}B(I+DA^{-1}B)^{-1}\Big)DA^{-1} \\
&\qquad = I.
\end{aligned}$$

Recursive least squares regression

- This yields the following relation:

$$\begin{aligned} P_{n+1} &= P_n - P_n x_{n+1}(1 + x_{n+1}^{\mathrm{T}} P_n x_{n+1})^{-1} x_{n+1}^{\mathrm{T}} P_n \\ &= P_n - K_{n+1} x_{n+1}^{\mathrm{T}} P_n. \end{aligned}$$

  where we have introduced the notation

$$K_{n+1} = P_n x_{n+1}(1 + x_{n+1}^{\mathrm{T}} P_n x_{n+1})^{-1}.$$

- Now, define the *a priori error*

$$\widehat{\varepsilon}_{n+1} = y_{n+1} - x_{n+1}^{\mathrm{T}} \widehat{\beta}_n.$$

Recursive least squares regression

- Then the recursion for B_n can be recast as

$$B_{n+1} = B_n + x_{n+1}x_{n+1}^{\mathrm{T}}\widehat{\beta}_n + x_{n+1}\widehat{\varepsilon}_{n+1}.$$

- Using (6), we see that

$$\begin{aligned} P_{n+1}^{-1}\widehat{\beta}_{n+1} &= P_n^{-1}\widehat{\beta}_n + x_{n+1}x_{n+1}^{\mathrm{T}}\widehat{\beta}_n + x_{n+1}\widehat{\varepsilon}_{n+1} \\ &= (P_n^{-1} + x_{n+1}x_{n+1}^{\mathrm{T}})\widehat{\beta}_n + x_{n+1}\widehat{\varepsilon}_{n+1} \\ &= P_{n+1}^{-1}\widehat{\beta}_n + x_{n+1}\widehat{\varepsilon}_{n+1}. \end{aligned}$$

Recursive least squares regression

- In other words,
$$\widehat{\beta}_{n+1} = \widehat{\beta}_n + P_{n+1} x_{n+1} \widehat{\varepsilon}_{n+1}.$$
- However, from the definition of K_{n+1},
$$P_{n+1} x_{n+1} = K_{n+1},$$
and so
$$\widehat{\beta}_{n+1} = \widehat{\beta}_n + K_{n+1} \widehat{\varepsilon}_{n+1}.$$

Recursive least squares regression

- The algorithm can be summarized as follows.
- Initialize $\widehat{\beta}_0$ (e.g. $\widehat{\beta}_0 = 0$), and P_0 (e.g. $P_0 = \mu I$), and iterate:

$$\begin{aligned} \widehat{\varepsilon}_{n+1} &= y_{n+1} - x_{n+1}^{\mathrm{T}} \widehat{\beta}_n, \\ K_{n+1} &= P_n x_{n+1} (1 + x_{n+1}^{\mathrm{T}} P_n x_{n+1})^{-1}, \\ P_{n+1} &= P_n - K_{n+1} x_{n+1}^{\mathrm{T}} P_n, \\ \widehat{\beta}_{n+1} &= \widehat{\beta}_n + K_{n+1} \widehat{\varepsilon}_{n+1}. \end{aligned} \tag{8}$$

- As n increases, $\widehat{\beta}_n$ converges to the "true" $\widehat{\beta}$, regardless of the initial values $\widehat{\beta}_0$ and P_0.

Recursive least squares regression

- Note that
  - (i) we no longer have to store the (potentially large) matrices $\mathcal{X}_n$ and $\mathcal{Y}_n$,
  - (ii) the computationally expensive operation of inverting the matrix $\mathcal{X}_n\mathcal{X}_n^{\mathrm{T}}$ is replaced with a small number of simpler operations.
- One can also introduce a learning rate $0 < \lambda < 1$ so that the older observations are "forgotten" exponentially fast.
- We now move on to the main topic of these notes, the Kalman filter and its generalizations.

State space models

- A *state space model* (SSM) is a time series model in which the time series Y_t is interpreted as the result of a noisy observation of a stochastic process X_t.
- The values of the variables X_t and Y_t can be continuous (scalar or vector) or discrete.
- Graphically, an SSM is represented as follows:

$$\begin{array}{ccccccccc} X_1 & \rightarrow & X_2 & \rightarrow & \dots & \rightarrow & X_t & \rightarrow & \dots \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \dots \\ Y_1 & & Y_2 & & \dots & & Y_t & & \dots \end{array} \quad \begin{array}{l} \text{state process} \\ \\ \text{observation process} \end{array} \tag{9}$$

- SSMs belong to the realm of *Bayesian inference*, and they have been successfully applied in many fields to solve a broad range of problems.
- Our discussion of SSMs follows the book [2].

State space models

- It is usually assumed that the state process X_t is Markovian, i.e. X_t depends on the history only through X_{t-1}, and Y_t depends only on X_t:

$$\begin{aligned} X_t &\sim p(X_t|X_{t-1}), \\ Y_t &\sim p(Y_t|X_t). \end{aligned} \tag{10}$$

- The most well studied SSM is the *Kalman filter*, for which the processes above are linear and and the sources of randomness are Gaussian.
- Namely, a *linear state space model* has the form:

$$\begin{aligned} X_{t+1} &= GX_t + \varepsilon_{t+1}, \\ Y_t &= HX_t + \eta_t. \end{aligned} \tag{11}$$

State space models

- Here, the *state vector* $X_t \in \mathbb{R}^r$ is possibly unobservable and it can be observed only through the *observation vector* $Y_t \in \mathbb{R}^n$.
- The matrices $G \in \mathrm{Mat_r}(\mathbb{R})$ and $H \in \mathrm{Mat_{n,r}}(\mathbb{R})$ are assumed to be known.
- For example, their values may be given by (economic) theory, or they may have been obtained through MLE estimation.
- In fact, the matrices G and H may depend deterministically on time, i.e. G and H may be replaced by known matrices G_t and H_t, respectively.
- We also assume that the distribution of the initial value X_1 is known and Gaussian.

State space models

- The vectors of residuals $\varepsilon_t \in \mathbb{R}^r$ and $\eta_t \in \mathbb{R}^n$ satisfy

$$\begin{aligned} \mathrm{E}(\varepsilon_t \varepsilon_s^{\mathrm{T}}) &= \delta_{ts} Q, \\ \mathrm{E}(\eta_t \eta_s^{\mathrm{T}}) &= \delta_{ts} R, \end{aligned} \tag{12}$$

  where δ_{ts} denotes Kronecker's delta, and where Q and R are known positive definite (covariance) matrices.
- We also assume that the components of ε_t and η_s are independent of each other for all t and s.
- The matrices Q and R may depend deterministically on time.
- The first of the equations in (11) is called the *state equation*, while the second one is referred to as the *observation equation*.

Inference for state space models

- Let T denote the time horizon.
- Our broad goal is to make inference about the states X_t based on a set of observations $Y_1, \ldots, Y_t$.
- Three questions are of particular interest:
  - (i) *Filtering:* $t < T$. What can we infer about the current state of the system based on all available observations?
  - (ii) *Smoothing:* $t = T$. What can be inferred about the system based on the information contained in the entire data sample? In particular, how can we back fill missing observations?
  - (iii) *Forecasting:* $t > T$. What is the optimal prediction of a future observation and / or a future state of the system?

Kalman filter

- In principle, any inference for this model can be done using the standard methods of multivariate statistics.
- However, these methods require storing large amounts of data and inverting $tn \times tn$ matrices.
- Notice that, as new data arrive, the storage requirements and matrix dimensionality increase.
- This is frequently computationally intractable and impractical.
- Instead, the Kalman filter relies on a *recursive* approach which does not require significant storage resources and involves inverting $n \times n$ matrices only.
- We will go through a detailed derivation of this recursion.

Kalman filter

- The purpose of filtering is to update the knowledge of the system each time a new observation is made.
- We define the *one period predictor* μ_{t+1}, when the observation Y_t is made, and its covariance P_{t+1}:

$$\begin{aligned} \mu_{t+1} &= \mathrm{E}(X_{t+1}|Y_{1:t}), \\ P_{t+1} &= \mathrm{Var}(X_{t+1}|Y_{1:t}), \end{aligned} \tag{13}$$

as well as the *filtered estimator* $\mu_{t|t}$ and its covariance $P_{t|t}$:

$$\begin{aligned} \mu_{t|t} &= \mathrm{E}(X_t|Y_{1:t}), \\ P_{t|t} &= \mathrm{Var}(X_t|Y_{1:t}). \end{aligned} \tag{14}$$

- Our objective is to compute these quantities recursively.

Kalman filter

- We let
$$v_t = Y_t - \mathrm{E}(Y_t | Y_{1:t-1}) \tag{15}$$
denote the *one period prediction error* or *innovation*.
- In Homework Assignment #5 we show that v_t's are mutually independent.
- Note that
$$\begin{aligned} v_t &= Y_t - \mathrm{E}(HX_t + \eta_t | Y_{1:t-1}) \\ &= Y_t - H\mu_t. \end{aligned}$$

Kalman filter

- As a consequence, we have, for $t = 2, 3, \ldots,$

$$\begin{aligned} \mathrm{E}(v_t | Y_{1:t-1}) &= \mathrm{E}(HX_t + \eta_t - H\mu_t | Y_{1:t-1}) \\ &= 0. \end{aligned} \tag{16}$$

- Now we notice that

$$\begin{aligned} \mathrm{Var}(v_t | Y_{1:t-1}) &= \mathrm{Var}(HX_t + \eta_t - H\mu_t | Y_{1:t-1}) \\ &= \mathrm{Var}(H(X_t - \mu_t) | Y_{1:t-1}) + \mathrm{Var}(\eta_t | Y_{1:t-1}) \\ &= \mathrm{E}(H(X_t - \mu_t)(X_t - \mu_t)^{\mathrm{T}} H^{\mathrm{T}} | Y_{1:t-1}) + \mathrm{E}(\eta_t \eta_t^{\mathrm{T}} | Y_{1:t-1}) \\ &= HP_t H^{\mathrm{T}} + R. \end{aligned}$$

Kalman filter

- For convenience we denote

$$F_t = \mathrm{Var}(v_t | Y_{1:t-1}), \tag{17}$$

  and we will assume in the following that the matrix F_t is invertible.
- The result of the calculation above can thus be stated as:

$$F_t = HP_tH^{\mathrm{T}} + R. \tag{18}$$

- This relation allows us to derive a relation between μ_t and $\mu_{t|t}$.

Lemma

- First, we will establish the following *Lemma*.
- Let X and Y be Gaussian jointly distributed random vectors with

$$\mathrm{E}\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix},$$

and

$$\mathrm{Var}\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{XY}^{\mathrm{T}} & \Sigma_{YY} \end{pmatrix}$$

Then

$$\mathrm{E}(X|Y) = \mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y), \tag{19}$$

and

$$\mathrm{Var}(X|Y) = \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{XY}^{\mathrm{T}}. \tag{20}$$

Lemma

- *Proof*: Consider the random variable

$$Z = X - \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y).$$

- Since Z is a linear in X and Y, the vector (Y, Z) is Gaussian jointly distributed.
- Furthermore,

$$\begin{aligned} \mathrm{E}(Z) &= \mu_X, \\ \mathrm{Var}(Z) &= \mathrm{E}((Z - \mu_X)(Z - \mu_X)^{\mathrm{T}}) \\ &= \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{XY}^{\mathrm{T}}. \end{aligned}$$

Lemma

- Finally,

$$\begin{aligned}\mathrm{Cov}(Y, Z) &= \mathrm{E}(Y(Z - \mu_X)^{\mathrm{T}}) \\ &= \mathrm{E}(Y(X - \mu_X)^{\mathrm{T}} - Y(Y - \mu_Y)^{\mathrm{T}}\Sigma_{YY}^{-1}\Sigma_{XY}^{\mathrm{T}}) \\ &= 0.\end{aligned}$$

- This means that Z and Y are independently distributed!
- Consequently, $\mathrm{E}(Z|Y) = \mathrm{E}(Z)$ and $\mathrm{Var}(Z|Y) = \mathrm{Var}(Z)$.

Lemma

- Since
$$X = Z + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y),$$
we have
$$\mathrm{E}(X|Y) = \mu_X + \Sigma_{XY}\Sigma_{YY}^{-1}(Y - \mu_Y),$$
which proves (19).
- Also, conditioned on Y, X and Z differ by a constant vector, and so
$$\begin{aligned}\mathrm{Var}(X|Y) &= \mathrm{Var}(Z|Y)\\ &= \mathrm{Var}(Z)\\ &= \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{XY}^{\mathrm{T}},\end{aligned}$$
which proves (20). QED

Kalman filter

- Now, going back to the main calculation, we have

$$\begin{aligned}\mu_{t|t} &= \mathrm{E}(X_t|Y_{1:t}) \\ &= \mathrm{E}(X_t|Y_{1:t-1}, v_t),\end{aligned}$$

and

$$\begin{aligned}\mu_{t+1} &= \mathrm{E}(X_{t+1}|Y_{1:t}) \\ &= \mathrm{E}(X_{t+1}|Y_{1:t-1}, v_t).\end{aligned}$$

- Applying the Lemma to the joint distribution of X_t and v_t conditioned on Y_{t-1} yields

$$\mu_{t|t} = \mathrm{E}(X_t|Y_{1:t-1}) + \mathrm{Cov}(X_t, v_t|Y_{1:t-1})\mathrm{Var}(v_t|Y_{1:t-1})^{-1}v_t. \qquad (21)$$

Kalman filter

- Note that

$$\begin{aligned} \mathrm{Cov}(X_t, v_t | Y_{1:t-1}) &= \mathrm{E}\big(X_t(HX_t + \eta_t - H\mu_t)^{\mathrm{T}} | Y_{1:t-1}\big) \\ &= \mathrm{E}\big(X_t(X_t - \mu_t)^{\mathrm{T}} H^{\mathrm{T}} | Y_{1:t-1}\big) \\ &= \mathrm{E}\big((X_t - \mu_t)(X_t - \mu_t)^{\mathrm{T}} | Y_{1:t-1}\big) H^{\mathrm{T}} \\ &= P_t H^{\mathrm{T}}, \end{aligned} \tag{22}$$

  by the definition (13) of P_t.
- This allows us to rewrite equation (21) in the form

$$\mu_{t|t} = \mu_t + P_t H^{\mathrm{T}} F_t^{-1} v_t, \tag{23}$$

  where F_t is defined by (17).

Kalman filter

- Next, we conclude from the Lemma that

$$\begin{aligned}\mathrm{Var}(X_t|Y_{1:t}) &= \mathrm{Var}(X_t|Y_{1:t-1}, v_t) \\ &= \mathrm{Var}(X_t|Y_{1:t-1}) - \mathrm{Cov}(X_t, v_t|Y_{1:t-1})\mathrm{Var}(v_t|Y_{1:t-1})^{-1}\mathrm{Cov}(X_t, v_t|Y_{1:t-1})^{\mathrm{T}}.\end{aligned}$$

From the definition (14) of $P_{t|t}$, this can be stated as the following relation:

$$\begin{aligned}P_{t|t} &= P_t - P_t H^{\mathrm{T}} F_t^{-1} H P_t \\ &= (I - K_t H) P_t.\end{aligned} \tag{24}$$

where

$$K_t = P_t H^{\mathrm{T}} F_t^{-1}. \tag{25}$$

- The matrix K_t is referred to as the *Kalman gain*.
- Now we are ready to establish recursions for μ_t and P_t.

Kalman filter

- From the state equation (the first equation in (11)) we have

$$\begin{aligned} \mu_{t+1} &= \mathrm{E}(GX_t + \varepsilon_{t+1} | Y_{1:t}) \\ &= G\mathrm{E}(X_t | Y_{1:t}) \\ &= G\mu_{t|t}. \end{aligned} \tag{26}$$

- Furthermore,

$$\begin{aligned} P_{t+1} &= \mathrm{Var}(GX_t + \varepsilon_{t+1} | Y_{1:t}) \\ &= G\mathrm{Var}(X_t | Y_{1:t})G^{\mathrm{T}} + \mathrm{Var}(\varepsilon_{t+1} | Y_{1:t}). \end{aligned} \tag{27}$$

- Substituting (24) and (25) into (27) we find that

$$P_{t+1} = GP_{t|t}G^{\mathrm{T}} + Q. \tag{28}$$

- Relations (26) and (28) are called the *prediction step* of the Kalman filter.

Kalman filter

- Using this notation, we can write the full system of recursive relations for updating from t to $t+1$ in the following form:

$$\begin{aligned} v_t &= Y_t - H\mu_t, \\ F_t &= HP_tH^{\mathrm{T}} + R, \\ K_t &= P_tH^{\mathrm{T}}F_t^{-1}, \\ \mu_{t|t} &= \mu_t + K_tv_t, \\ P_{t|t} &= (I - K_tH)P_t, \\ \mu_{t+1} &= G\mu_{t|t}, \\ P_{t+1} &= GP_{t|t}G^{\mathrm{T}} + Q, \end{aligned} \tag{29}$$

for $t = 1, 2, \ldots$.

Kalman filter

- The initial values μ_1 and P_1 are assumed to be known, and consequently, $X_1 \sim N(\mu_1, P_1)$.
- If the matrices G, H, Q, R depend deterministically on t, the formulas above remain valid with G replaced by G_t, etc.

Kalman filter with mean adjustments

- It is sometimes necessary to consider a linear SSM with mean adjustments:

$$\begin{aligned} X_{t+1} &= GX_t + C_t + \varepsilon_{t+1}, \\ Y_t &= HX_t + D_t + \eta_t, \end{aligned} \tag{30}$$

  where $C_t \in \mathbb{R}^r$ and $D_t \in \mathbb{R}^n$ are deterministic (known).
- Following the arguments above, we can derive the following Kalman filter for (30):

$$\begin{aligned} v_t &= Y_t - H\mu_t - D_t, \\ F_t &= HP_tH^{\mathrm{T}} + R, \\ K_t &= P_tH^{\mathrm{T}}F_t^{-1}, \\ \mu_{t|t} &= \mu_t + K_tv_t, \\ P_{t|t} &= (I - K_tH)P_t, \\ \mu_{t+1} &= G\mu_{t|t} + C_t, \\ P_{t+1} &= GP_{t|t}G^{\mathrm{T}} + Q, \end{aligned} \tag{31}$$

for $t = 1, 2, \ldots$.

State smoothing

- *State smoothing* refers to the process of estimation of the values of the states $X_1, \ldots, X_T$, given the *entire* observation set.
- The objective is thus to (recursively) determine the conditional mean

$$\widehat{X}_t = \mathrm{E}(X_t | Y_{1:T}) \tag{32}$$

and the conditional variance

$$V_t = \mathrm{Var}(X_t | Y_{1:T}). \tag{33}$$

- Since all distributions are normal, $X_t | Y_{1:T} \sim N(\widehat{X}_t, V_t)$.
- As before, we assume that $X_1 \sim N(\mu_1, P_1)$ with known μ_1 and P_1.

State smoothing

- An analysis, similar to the derivation of the Kalman filter leads to the following result (see [2]) for the derivation).
- The smoothing process consists of two phases:
  - (i) *forward* sweep of the Kalman filter (29) for $t = 1, \ldots, T$,
  - (ii) *backward* recursion

$$\begin{aligned} R_{t-1} &= H^{\mathrm{T}} F_t^{-1} v_t + L_t^{\mathrm{T}} r_t, \\ N_{t-1} &= H^{\mathrm{T}} F_t^{-1} H + L_t^{\mathrm{T}} N_t L_t, \\ \widehat{X}_t &= \mu_t + P_t R_{t-1}, \\ V_t &= P_t - P_t N_{t-1} P_t, \end{aligned} \tag{34}$$

  where $L_t = G(I - K_t H)$, for $t = T, T-1, \ldots$, with the terminal condition $R_T = 0$ and $N_T = 0$.
- This version of the smoothing algorithm is somewhat unintuitive but computationally efficient.

Forecasting with Kalman filter

- The forecasting problem consists in predicting the values of X_{t+d} and Y_{t+d} given the observations $Y_{1:t}$.
- As discussed in Lecture Notes #1, the optimal forecasts of the state variable and observation are given by:

$$X^*_{t+d} = \mathrm{E}(X_{t+d}|Y_{1:t}), \tag{35}$$

with variance

$$P^*_{t+d} = \mathrm{E}\big((X^*_{t+d} - X_{t+d})(X^*_{t+d} - X_{t+d})^{\mathrm{T}}|Y_{1:t}\big), \tag{36}$$

and

$$Y^*_{t+d} = \mathrm{E}(Y_{t+d}|Y_{1:t}), \tag{37}$$

with variance

$$V^*_{t+d} = \mathrm{E}\big((Y^*_{t+d} - Y_{t+d})(Y^*_{t+d} - Y_{t+d})^{\mathrm{T}}|Y_{1:t}\big), \tag{38}$$

respectively.

Forecasting with Kalman filter

- The forecast is straightforward for $d = 1$:

$$X^*_{t+1} = G\mu_{t|t}, \tag{39}$$

with

$$P^*_{t+1} = GP_{t+1}G^{\mathrm{T}} + Q, \tag{40}$$

and

$$Y^*_{t+1} = H\mu_{t+1}, \tag{41}$$

with

$$V^*_{t+1} = HP_{t+1}H^{\mathrm{T}} + R. \tag{42}$$

Forecasting with Kalman filter

- For $d > 1$ we obtain the recursions:

$$X^*_{t+d} = GX^*_{t+d-1}, \tag{43}$$

with

$$P^*_{t+d} = GP^*_{t+d-1}G^{\mathrm{T}} + Q, \tag{44}$$

and

$$Y^*_{t+d} = HX^*_{t+d}, \tag{45}$$

with

$$V^*_{t+d} = HP^*_{t+d}H^{\mathrm{T}} + R. \tag{46}$$

MLE estimation of the parameters

- We have left a number of parameters that may have not been specified, namely:
  - (i) the initial values μ_1 and P_1 that enter the probability distribution $N(\mu_1, P_1)$ of the state X_1,
  - (ii) the matrices G and H,
  - (iii) the variances Q and R of the disturbances in the state and observation equations, respectively.
- We denote the unspecified model parameters collectively by θ.
- If μ_1 and P_1 are known, the remaining parameters θ can be estimated by means of MLE.

MLE estimation of the parameters

- To this end, we consider the joint probability of the observations:

$$p(Y_{1:T}|\theta) = \prod_{t=1}^{T} p(Y_t|Y_{1:t-1}), \tag{47}$$

where $p(Y_1|Y_0) = p(Y_1)$.

- Hence,

$$-\log \mathcal{L}(\theta|Y_{1:T}) = \frac{1}{2} \sum_{t=1}^{T} \left(\log \det(F_t) + v_t^{\mathrm{T}} F_t^{-1} v_t \right) + const, \tag{48}$$

where v_t denotes the innovation.

MLE estimation of the parameters

- For each t, the value of the log likelihood function is calculated by running the Kalman filter.
- Searching for the minimum of this log likelihood function using an efficient algorithm such as BFGS, we find estimates of θ.
- In the case of unknown μ_1 and P_1, one can use the *diffuse log likelihood* method, which is discussed in detail in [2].
- Alternatively, one can regard μ_1 and P_1 *hyperparameters* of the model.

Nonlinear state space models

- The recursion relations defining the Kalman filter can be extended to the case of more general state space models.
- Some of these extensions are straightforward (such as adding constant terms to the right hand sides of the state and observation equations in (11)), others are fundamentally more complicated.
- In the remainder of this lecture we summarize these extensions following [2].

Nonlinear state space models

- In general, a state space model has the form:

$$\begin{array}{ccccccccc} X_1 & \rightarrow & X_2 & \rightarrow & \dots & \rightarrow & X_t & \rightarrow & \dots \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \dots \\ Y_1 & & Y_2 & & \dots & & Y_t & & \dots \end{array} \tag{49}$$

  where the state and observed variables may be continuous or discrete.
- It is also not required that the distributions of the residuals are Gaussian.
- Such models include linear and non-linear Kalman filters, hidden Markov models, stochastic volatility models, etc.
- As in the case of a linear SSM, the *filtering problem* is to estimate *sequentially* the values of the unobserved states X_t, given the values of the observation process $Y_1, \cdots, Y_t$, for any time step t.

Nonlinear state space models

- We assume that the states X_t and the observations Y_t can be modeled in the following form.
- $X_1, X_2, \cdots$, is a Markov process on $\mathbb{R}^n$ that evolves according to the transition probability density $p(X_t \mid X_{t-1})$:

$$X_t \mid X_{t-1} \sim p(X_t \mid X_{t-1}). \tag{50}$$

- On the other hand, Y_t depends only on the value of the state variable X_t:

$$Y_t \sim p(Y_t \mid X_t). \tag{51}$$

- Usually, these two relations are stated in explicit functional form:

$$\begin{aligned} X_t &= G(X_{t-1}, \varepsilon_t), \\ Y_t &= H(X_t, \eta_t), \end{aligned} \tag{52}$$

where ε_t and η_t are noises.

Stochastic volatility model

- An example of a nonlinear SSM is the stochastic volatility model

$$\begin{aligned} X_{t+1} &= a + X_t + \varepsilon_{t+1}, \\ Y_t &= \exp(X_t)\eta_t, \end{aligned} \tag{53}$$

  where $X_t, Y_t \in \mathbb{R}$, $a \in \mathbb{R}$, $\varepsilon_t \sim N(0, \alpha^2)$, and $\eta_t \sim N(0, 1)$.

- This model can be thought of as follows:
  - (i) the (hidden) state variable X_t drives the stochastic volatility process $\sigma_t = \exp(X_t)$,
  - (ii) the volatility process is observed through the change $Y_t = F_{t+1} - F_t$ in the market observable F_t (such as asset price or forward rate).
- One can view this model as a discretized version of a continuous time stochastic volatility model such as SABR.

Extended Kalman filter

- The *extended Kalman filter* (EKF) consists in approximating a nonlinear SSM by a linear SSM followed by applying the Kalman filter.
- Namely, assume that the state and observation equations are given by

$$\begin{aligned} X_{t+1} &= G(X_t) + \varepsilon_{t+1}, \\ Y_t &= H(X_t) + \eta_t, \end{aligned} \tag{54}$$

respectively, where

(i) $G(x)$ and $H(x)$ are differentiable functions on $\mathbb{R}^r$,
(ii) the disturbances ε_t and η_t are mutually and serially uncorrelated with mean zero and covariances $Q(X_t)$ and $R(X_t)$, respectively (we do not require that their distributions are Gaussian),
(iii) X_1 has mean μ_1 and variance P_1, and is uncorrelated with all noises.

Extended Kalman filter

- We denote by G'_t and H'_t the matrices of first derivatives (Jacobi matrices) of $G(X_t)$ and $H(X_t)$ evaluated at μ_t and $\mu_{t|t}$, respectively:

$$G'_t = \nabla G(X_t)|_{X_t=\mu_{t|t}},$$
$$H'_t = \nabla H(X_t)|_{X_t=\mu_t}.$$

- We now expand the matrix functions G, H, Q and R in Taylor series to the orders indicated:

$$G(X_t) = G(\mu_{t|t}) + G'_t(X_t - \mu_{t|t}) + \ldots,$$
$$H(X_t) = H(\mu_t) + H'_t(X_t - \mu_t) + \ldots,$$
$$Q(X_t) = Q(\mu_{t|t}) + \ldots,$$
$$R(X_t) = R(\mu_t) + \ldots,$$

and disregard the higher order terms denoted by $\ldots$.

Extended Kalman filter

- As a result of this approximation we obtain a linear SSM with mean adjustment:

$$\begin{aligned} X_{t+1} &= G_t' X_t + \left(G(\mu_{t|t}) - G_t' \mu_{t|t}\right) + \varepsilon_{t+1}, \\ Y_t &= H_t' X_t + \left(H(\mu_t) - H_t' \mu_t\right) + \eta_t. \end{aligned} \tag{55}$$

- Applying the Kalman filter formulas to this SSM we obtain the following EKF recursion:

$$\begin{aligned} v_t &= Y_t - H(\mu_t), \\ F_t &= H_t' P_t H_t'^{\mathrm{T}} + R(\mu_t), \\ K_t &= P_t H'^{\mathrm{T}} F_t^{-1}, \\ \mu_{t|t} &= \mu_t + K_t v_t, \\ P_{t|t} &= (I - K_t H_t') P_t, \\ \mu_{t+1} &= G(\mu_{t|t}), \\ P_{t+1} &= G_t' P_{t|t} G_t'^{\mathrm{T}} + Q(\mu_{t|t}), \end{aligned} \tag{56}$$

for $t = 1, 2, \ldots$.

Extended Kalman filter

- The EKF works well if the functions G and H are weakly nonlinear, for strongly nonlinear models its performance may be poor.
- Other extensions of the Kalman filter have been developed, including the *unscented Kalman filter* (UKF).
- It is based on a different principle than the EKF: rather than approximating G and H by linear expressions, one matches approximately the first and second moments of a nonlinear function of a Gaussian random variable, see [2] for details.
- Another approach to inference in nonlinear SSMs is via Monte Carlo (MC) techniques *particle filters*, a.k.a. *sequential Monte Carlo*.

Particle filtering

- Estimation of complex time series models requires evaluation of complex expected values, often expressed as high dimensional, analytically intractable integrals.
- Particle filters provide a method for calculating such integrals approximately via carefully crafted MC techniques.
- In this approach, a continuous PDF is approximated by a discrete PDF made of weighted outcomes called *particles*.
- Particle filter algorithms are formulated recursively, very much in the spirit of the Kalman filter.
- They are far reaching generalizations of the Kalman filter to nonlinear, non-Gaussian SSMs.
- Since particle filtering is based on MC methods, its performance or accuracy does not much that of the Kalman filter.

Nonlinear state space models

- The probability distributions in the following depend on some parameters θ. In order to streamline the notation, we will suppress θ from all the formulas.
- All (Bayesian) inference about X_t is encoded in the *posterior* PDF $p(X_t \mid Y_{1:t})$.
- The particle filter methodology provides an approximation of these conditional probabilities using the *empirical measure* associated with a sampling algorithm.
- The objective of a particle filter is to estimate the posterior PDF of the (unobserved) state variables given a time series of observations.
- Distribution properties of the state variable can be captured by the *joint smoothing distribution*, which is defined as

$$p(X_{1:t} \mid Y_{1:t}) = \frac{p(X_{1:t}, Y_{1:t})}{p(Y_{1:t})} \,. \tag{57}$$

Joint smoothing distribution

- We derive the following recursion relation for the joint smoothing distribution:

$$
\begin{aligned}
p(X_{1:t} \mid Y_{1:t}) &= \frac{p(Y_t \mid X_{1:t}, Y_{1:t-1}) p(X_{1:t}, Y_{1:t-1})}{p(Y_t, Y_{1:t-1})} \\
&= \frac{p(Y_t \mid X_{1:t}, Y_{1:t-1}) p(X_t \mid X_{1:t-1}, Y_{1:t-1})}{p(Y_t \mid Y_{1:t-1})} \, p(X_{1:t-1} \mid Y_{1:t-1}) \qquad (58) \\
&= \frac{p(Y_t \mid X_t) p(X_t \mid X_{t-1})}{p(Y_t \mid Y_{1:t-1})} \, p(X_{1:t-1} \mid Y_{1:t-1}).
\end{aligned}
$$

- This recursion will be approximated by numerically tractable expressions.

Filtering recursion

- An alternative to working directly with the joint smoothing distribution is to find recursive relations for the one-period predictive and filtering distributions.
- This is analogous to the approach we took when deriving the Kalman filter.
- Assume that the initial distribution $p(X_1)$ is known.
- The one-period prediction distribution is given by

$$p(X_t \mid Y_{1:t-1}) = \int p(X_t \mid x_{t-1}) p(x_{t-1} \mid Y_{1:t-1}) dx_{t-1}. \tag{59}$$

Filtering recursion

- The filtering distribution is calculated based on the arrival of the new observation Y_t.
- Namely, applying Bayes' rule, and the fact that Y_t depends on X_t only,

$$\begin{aligned} p(X_t \mid Y_{1:t}) &= \frac{p(Y_t, X_t \mid Y_{1:t-1})}{p(Y_t \mid Y_{1:t-1})} \\ &= \frac{p(Y_t \mid X_t, Y_{1:t-1}) p(X_t \mid Y_{1:t-1})}{\int p(Y_t \mid x_t) p(x_t \mid Y_{1:t-1}) dx_t} \\ &= \frac{p(Y_t \mid X_t) p(X_t \mid Y_{1:t-1})}{\int p(Y_t \mid x_t) p(x_t \mid Y_{1:t-1}) dx_t} . \end{aligned} \tag{60}$$

Filtering recursion

- The difficulty with this recursion is clear: there is a complicated integral in the denominator, which cannot in general be calculated in closed form.
- In some special cases this can be done: for example, in the case of a linear Gaussian state space model, this integral is Gaussian and can be calculated.
- The recursion above leads then to the Kalman filter.
- Instead of trying to evaluate the integral numerically, we will develop a Monte Carlo based approach for approximately solving recursions (59) and (60).

Importance sampling

- Suppose we are faced with Monte Carlo evaluation of the expected value

$$\mathrm{E}(f(X_{1:t}) \mid Y_{1:t}) = \int f(x_{1:t}) p(x_{1:t} \mid Y_{1:t}) dx_{1:t}. \tag{61}$$

- The straightforward approach would be to generate a number of samples $x_{1:t}^j,\ j = 1, \ldots, N$, from the distribution $p(x_{1:t} | Y_{1:t})$, evaluate the integrand $f(x_{1:t}^j)$ on each of these samples, and take the average of these values.
- This approach may prove impractical if the density $p(x_{1:t} \mid Y_{1:t})$ is hard to simulate from.
- Instead, we use the method of *importance sampling* (IS).
- We proceed as follows:

Importance sampling

1. Choose a *proposal distribution* $g(X_{1:t} \mid Y_{1:t})$, and write

$$\mathrm{E}(f(X_{1:t}) \mid Y_{1:t}) = \int f(x_{1:t}) \frac{p(x_{1:t} \mid Y_{1:t})}{g(x_{1:t} \mid Y_{1:t})} g(x_{1:t} \mid Y_{1:t}) dx_{1:t}. \tag{62}$$

   The proposal distribution should be chosen so that it is easy to sample from it.

2. Draw N samples of paths $x_{1:t}^1, \ldots, x_{1:t}^N$ from the proposal distribution, and assign to each of them a weight proportional to the ratio of the target and proposal distributions:

$$w_t^j \propto \frac{p(x_{1:t}^j \mid Y_{1:t})}{g(x_{1:t}^j \mid Y_{1:t})}. \tag{63}$$

Importance sampling

3. Given the sample, we define the estimated expected value by

$$\widehat{\mathrm{E}}_N(f(X_{1:t}) \mid Y_{1:t}) = \sum_{j=1}^{N} \widehat{w}_t^j \, f(x_{1:t}^j), \tag{64}$$

where the *importance weights* $\widehat{w}_t^j$, $j = 1, \ldots, N$, are given by

$$\widehat{w}_t^j = \frac{w_t^j}{\sum_{j=1}^{N} w_t^j}. \tag{65}$$

- The efficiency of IS depends essentially on how closely the proposal distribution $g(X_{1:t} \mid Y_{1:t})$ matches the target distribution.
- One could, for example, settle on a parametric distribution such as Gaussian and fine tune its parameters by minimizing its KL divergence from $p(x_{1:t} \mid Y_{1:t})$.

Sequential importance sampling

- Another serious limitation of IS is that it is computationally very expensive to generate $x^j_{1:t}$, and that this cost increases with t.
- To mitigate it, the method of *sequential importance sampling* (SIS) has been developed.
- In this approach we retain the previously simulated values $x^j_{1:t-1}$ and generate the value of x^j_t only.
- In order to implement this idea, the samples $x^j_{1:t}$ are simulated from a sequence of conditional distributions rather than a joint proposal distribution.
- The proposal distribution can be factored into two pieces as follows :

$$\begin{aligned} g(X_{1:t} \mid Y_{1:t}) &= \frac{g(X_{1:t}, Y_{1:t})}{g(Y_{1:t})} \\ &= \frac{g(X_t \mid X_{1:t-1}, Y_{1:t}) g(X_{1:t-1}, Y_{1:t})}{g(Y_{1:t})} \\ &= g(X_t \mid X_{1:t-1}, Y_{1:t}) g(X_{1:t-1} \mid Y_{1:t}). \end{aligned}$$

Sequential importance sampling

- Once the sample of $X_{1:t-1}$ has been generated from $g(X_{1:t-1} \mid Y_{1:t-1})$, its value is independent of the observation Y_t, and so $g(X_{1:t-1} \mid Y_{1:t}) = g(X_{1:t-1} \mid Y_{1:t-1})$.
- We can thus write the result of the calculation above as the following recursion:

$$g(X_{1:t} \mid Y_{1:t}) = g(X_t \mid X_{1:t-1}, Y_{1:t}) g(X_{1:t-1} \mid Y_{1:t-1}). \tag{66}$$

- The second factor on the RHS of this equation, $g(X_{1:t-1} \mid Y_{1:t-1})$, is the proposal distribution built out of the paths that have already been generated in the previous steps.
- A new set of samples $x_t^1, \ldots, x_t^N$ is drawn from the first factor $g(X_t \mid X_{1:t-1}, Y_{1:t})$.
- We then append the newly simulated values $x_t^1, \ldots, x_t^N$ to the simulated paths $x_{1:t-1}^1, \ldots, x_{1:t-1}^N$ of length $t-1$.
- We thus obtain simulated paths $x_{1:t}^1, \ldots, x_{1:t}^N$ of length t.

Sequential importance sampling

- The weights (63) can be computed as follows. Using (58) and (66),

$$
\begin{aligned}
w_t^j &\propto \frac{p(Y_t \mid x_t^j)p(x_t^j \mid x_{t-1}^j)p(x_{1:t-1}^j \mid Y_{1:t-1})}{p(Y_t \mid Y_{1:t-1})g(x_t^j \mid x_{1:t-1}^j, Y_{1:t})g(x_{1:t-1}^j \mid Y_{1:t-1})} \\
&= \frac{p(Y_t \mid x_t^j)p(x_t^j \mid x_{t-1}^j)}{p(Y_t \mid Y_{1:t-1})g(x_t^j \mid x_{1:t-1}^j, Y_{1:t})} \times \frac{p(x_{1:t-1}^j \mid Y_{1:t-1})}{g(x_{1:t-1}^j \mid Y_{1:t-1})} \qquad (67) \\
&\propto \frac{p(Y_t \mid x_t^j)p(x_t^j \mid x_{t-1}^j)}{g(x_t^j \mid x_{1:t-1}^j, Y_{1:t})} w_{t-1}^j \\
&= \widetilde{w}_t w_{t-1}^j,
\end{aligned}
$$

  where the factor $\widetilde{w}_t$ is defined by

$$
\widetilde{w}_t = \frac{p(Y_t \mid x_t^j)p(x_t^j \mid x_{t-1}^j)}{g(x_t^j \mid x_{1:t-1}^j, Y_{1:t})}. \qquad (68)
$$

- We initialize this distribution with $w_1^j = 1$.

Sequential importance sampling

- The densities $p(Y_t \mid X_t)$ and $p(X_t \mid X_{t-1})$ are determined by the state and observation equations (52).
- The only quantity that needs to be computed at each iteration is the ratio of weights $\widetilde{w}_t$.
- As a result of each iteration, SIS produces N Monte Carlo paths $x_{1:t}^1, \ldots, x_{1:t}^N$ along with the *unnormalized* importance weights $w_t^1, \ldots, w_t^N$.
- These paths are referred to as *particles*.
- We define the normalized weights by

$$\widehat{w}_t^j = \frac{w_t^j}{\sum_{i=1}^N w_t^i} \,. \tag{69}$$

Sequential importance sampling

- The joint smoothing PDF is estimated as follows:

$$\widehat{p}(X_{1:t} \mid Y_{1:t}) = \sum_{j=1}^{N} \widehat{w}_t^j \, \delta(X_{1:t} - x_{1:t}^j), \tag{70}$$

where δ denotes Dirac's delta function, and so

$$\widehat{\mathrm{E}}(f(X_{1:t}) \mid Y_{1:t}) = \sum_{j=1}^{N} \widehat{w}_t^j \, f(x_{1:t}^j). \tag{71}$$

- The estimated contribution to the likelihood function at time t is equal to

$$\widehat{p}(Y_t \mid Y_{1:t-1}) = \sum_{j=1}^{N} \widehat{w}_{t-1}^j \widetilde{w}_t^j. \tag{72}$$

Sequential importance sampling with resampling

- It has been observed that in practice SIS suffers from the *weight degeneracy* problem.
- This manifests itself in the rapid increase of the variance of the distribution of the importance weights as the number of time steps t increases.
- As t increases, all the probability density gets eventually allocated to a single particle.
- That particle's normalized weight converges to one, while the normalized weights of the other particles converge to zero, and the SIS estimator becomes a function of a single sample.

Sequential importance sampling with resampling

- A remedy to this problem is *resampling*, a process in which a new population of particles is replicated from the existing population in proportion to their normalized importance weights.
- This algorithm is called *sequential importance sampling with resampling* (SISR).
- A new population of particles is generated by sampling from the existing population:
  - (i) The probability of selecting a particle is proportional to its normalized importance weight.
  - (ii) Once the resampled particles are selected, their weights are set equal (to $1/N$). This prevents the weights from degenerating as in SIS.
- We proceed as follows:

1. Initialize the filter: draw N samples $x_1^j \sim g(X_1)$ and define the weights

$$w_1^j = \frac{p(x_1^j)}{g(x_1^j)}.$$

Sequential importance sampling with resampling

2. For $t = 2, \ldots, T$:
   (i) Generate N samples $x_t^j \sim g(X_t \mid x_{t-1}^j, Y_{1:t})$ and compute the importance weights
   $$w_t^j \propto \frac{p(Y_t \mid x_t^j)\, p(x_t^j \mid x_{t-1}^j)}{g(x_t^j \mid x_{t-1}^j, Y_{1:t})}\, w_{t-1}^j.$$
   (ii) Normalize the importance weights:
   $$\widehat{w}_t^j = \frac{w_t^j}{\sum_{j=1}^N w_t^j}. \tag{73}$$
   (iii) Resample N particles with probabilities $\widehat{w}_t^1, \ldots, \widehat{w}_t^N$, and define $w_t^j = 1/N$.

- After every iteration, once the particles have been generated, quantities of interest can be estimated.

Sequential importance sampling with resampling

- The joint smoothing PDF at time t is estimated as follows:

$$\widehat{p}_N(X_{1:t} \mid Y_{1:t}) = \sum_{j=1}^{N} \widehat{w}_t^j \delta(X_{1:t} - x_{1:t}^j), \tag{74}$$

where the normalized weights are given by (73).

- The estimate of the expected value of a function $f(X_{1:t})$ of the path $X_{1:t}$ is given by

$$\widehat{\mathrm{E}}_N(f(X_{1:t}) \mid Y_{1:t}) = \sum_{j=1}^{N} \widehat{w}_t^j \delta(X_{1:t} - x_{1:t}^j). \tag{75}$$

- The contribution to the likelihood function at time t is estimated as follows:

$$\begin{aligned} \widehat{p}_N(Y_t \mid Y_{1:t-1}) &\approx \int p(Y_t \mid x_t) p(x_t \mid Y_{1:t-1}) dx_t \\ &\approx \frac{1}{N} \sum_{j=1}^{N} \widehat{w}_t^j. \end{aligned} \tag{76}$$

Bootstrap filter

- The efficacy of the algorithms presented above depends on the choice of the proposal distribution.
- The simplest choice of the proposal distribution is

$$g(X_t \mid X_{t-1}, Y_t) = p(X_t \mid X_{t-1}). \tag{77}$$

  This choice is called the *prior kernel*, and the corresponding particle filter is called the *bootstrap filter*.
- The bootsstrap filter resamples by setting the incremental weight ratios equal to $\widetilde{w}_t = p(Y_t \mid X_t)$.
- The prior kernel is an example of a *blind proposal*: it does not use the current observation Y_t.
- Despite this, the bootstrap filter performs well in a number of situations.
- Another popular version is the *auxiliary particle filter*, see [1] and [2].

References


[1] Creal, D.: A survey of sequential Monte Carlo methods for economics and finance, *Economic Reviews*, **31**, 245 - 296 (2012).


[2] Durbin, J., and Koopman, S. J.: *Time Series Analysis by State Space Methods*, Oxford University Press (2012).