Short Course on Statistics and Uncertainty Part V

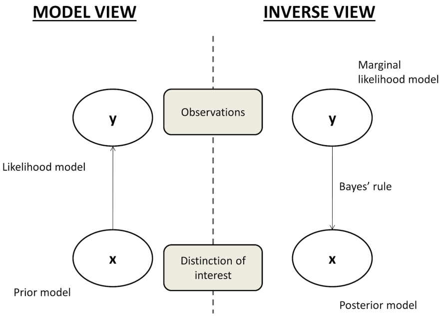
Jo Eidsvik

Department of Mathematical Sciences, NTNU, Norway

Topics

- Sequential methods for data conditioning
- State space models
- Bayesian filtering
- Kalman filter, ensemble Kalman filter

Model and inversion



Model for $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})$$

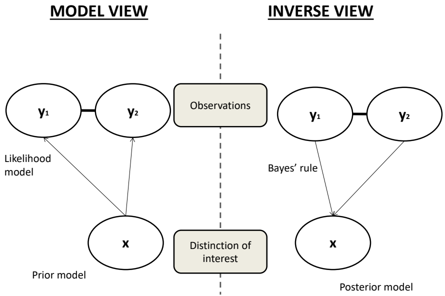
For the analysis, the main interest is in the conditional $p(\boldsymbol{x}|\boldsymbol{y})$.

Bayes rule

$p(\boldsymbol{x})$ from a priori knowledge, $p(\boldsymbol{y}|\boldsymbol{x})$ from data acquisition. Bayes' rule gives the posterior:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y})}$$

Inversion of multiple data



Inversion of multiple data

Model for $\boldsymbol{x}$, and $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is

$$p(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2) = p(\boldsymbol{x})p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x})$$

$p(\boldsymbol{x})$ from a priori knowledge, $p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x})$ from data acquisition.
Bayes' rule gives the posterior:

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \boldsymbol{y}_2) = \frac{p(\boldsymbol{x})p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x})}{p(\boldsymbol{y}_1, \boldsymbol{y}_2)} \propto p(\boldsymbol{x})p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x})$$

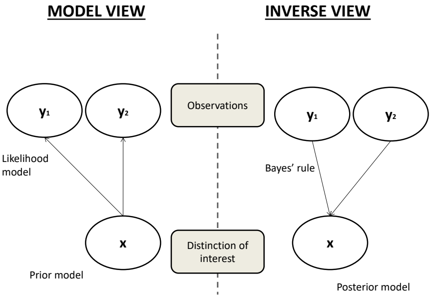
Conditional independence

$$p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x}) = p(\boldsymbol{y}_1|\boldsymbol{x})p(\boldsymbol{y}_2|\boldsymbol{x})$$

Bayes' rule gives the posterior:

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \boldsymbol{y}_2) = \frac{p(\boldsymbol{x})p(\boldsymbol{y}_1|\boldsymbol{x})p(\boldsymbol{y}_2|\boldsymbol{x})}{p(\boldsymbol{y}_1, \boldsymbol{y}_2)} \propto p(\boldsymbol{x})p(\boldsymbol{y}_1|\boldsymbol{x})p(\boldsymbol{y}_2|\boldsymbol{x})$$

Inversion of multiple data



Sequential Bayesian inversion

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \boldsymbol{y}_2) = \frac{p(\boldsymbol{x}|\boldsymbol{y}_1)p(\boldsymbol{y}_2|\boldsymbol{x})}{p(\boldsymbol{y}_2|\boldsymbol{y}_1)} \propto p(\boldsymbol{x}|\boldsymbol{y}_1)p(\boldsymbol{y}_2|\boldsymbol{x})$$

Generalization, $t = 1, \ldots, T$ data sources:

$$p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) \propto p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x})$$

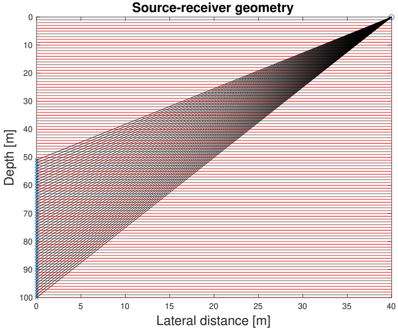
This is often called sequential updating or data assimilation.

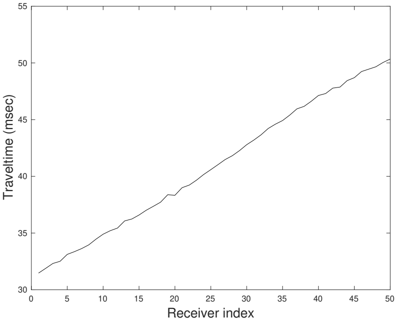
Approaches for sequential Bayesian inversion

- Monte Carlo samples from $p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$ are updated or re-weighted to get samples from $p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$.
- Closed form solution for $p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t) \propto p(\boldsymbol{x}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x})$. Gaussian-linear situation, or a discrete set of classes for $\boldsymbol{x}$.

Example of sequential assimilation of multiple data

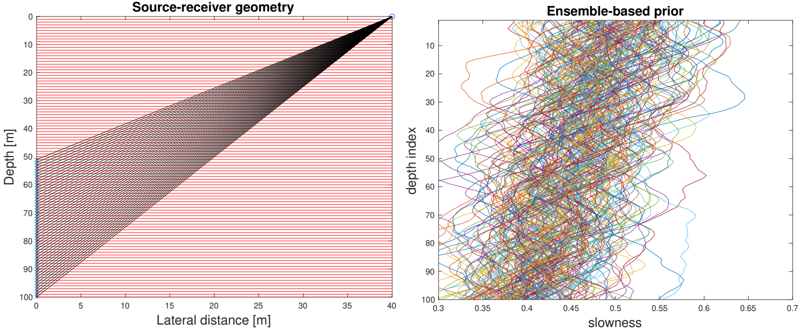
- Seismic experiment with 50 receiver depths in well and 1 source on surface (known locations).
- Use traveltime data to predict slowness in the subsurface.



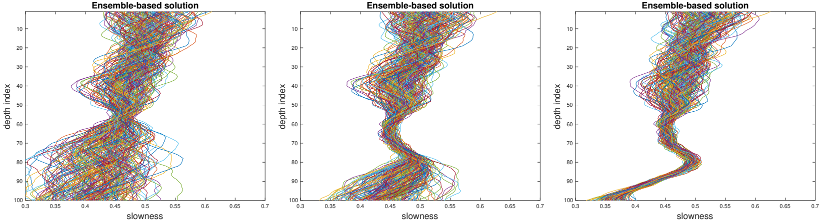


Example with traveltime data

- Traveltime $= \dfrac{\text{Distance}}{\text{Velocity}} =$ Distance $\cdot$ Slowness.
- Assimilate traveltime data $y_1, \ldots, y_{50}$ sequentially.
- Predict distribution for slowness. Initial slowness ensembles from prior model $p(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, \ldots, x_{100})$.



Example : Inversion after 10, 30 and 50 steps



State space models

- Variable $\boldsymbol{x}_t$ can change with index t. Index t is often time, but could be along a road, along a well, etc.
- We have a model for how $\boldsymbol{x}_t$ change with time (often assuming dependence only on the previous time).
- We have a model for how data $\boldsymbol{y}_t$ relates to $\boldsymbol{x}_t$ (often assuming conditional independence in the data).

A Common Type of State space model

Conditional independence in process (state) model:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_1, ...\boldsymbol{y}_{t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$

Conditional independence in measurement model:

$$p(\boldsymbol{y}_t|\boldsymbol{y}_1, ...\boldsymbol{y}_{t-1}, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_1) = p(\boldsymbol{y}_t|\boldsymbol{x}_t)$$

Filtering and Prediction

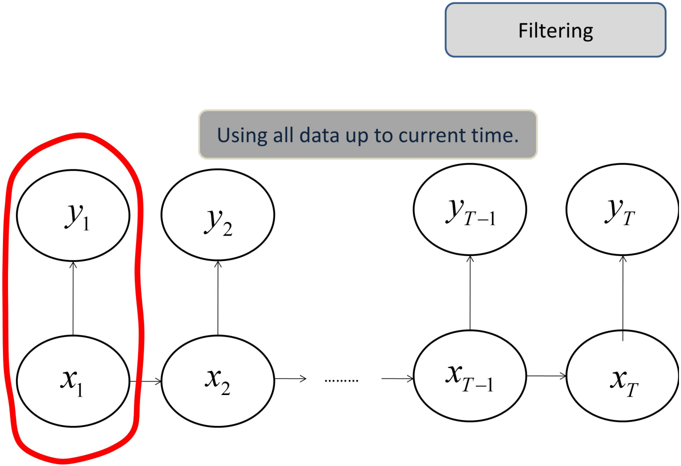
Filtering goal:

$$p(\boldsymbol{x}_t | \boldsymbol{y}_1, \dots \boldsymbol{y}_t)$$

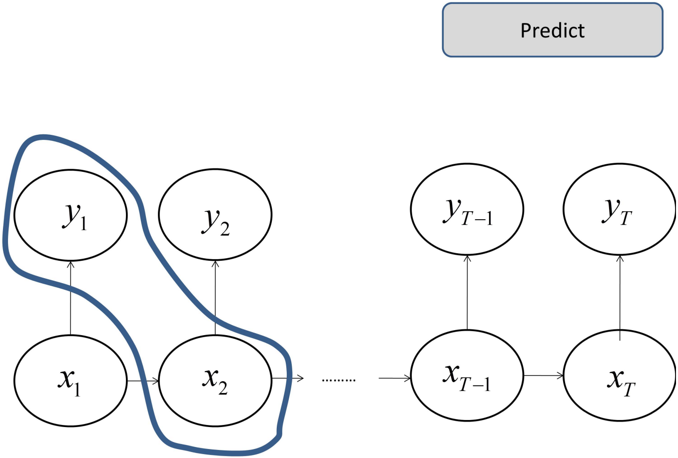
Prediction goal $s > t$:

$$p(\boldsymbol{x}_s | \boldsymbol{y}_1, \dots \boldsymbol{y}_t)$$

Filter


Filtering
Using all data up to current time.
y_1
y_2
y_{T-1}
y_T
x_1
x_2
....
x_{T-1}
x_T

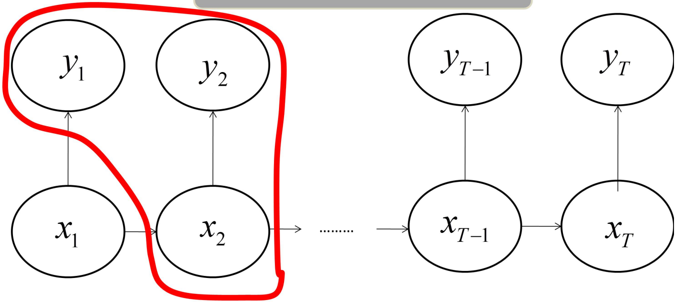
One step prediction



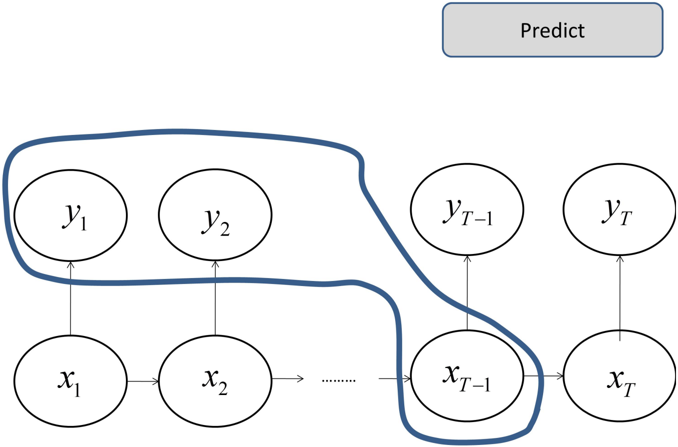
Filter

Filtering

Using all data up to current time.



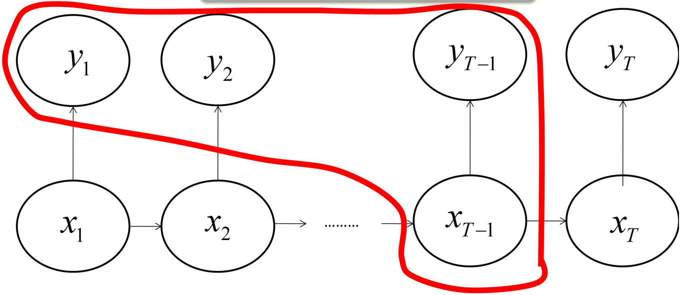
Predict



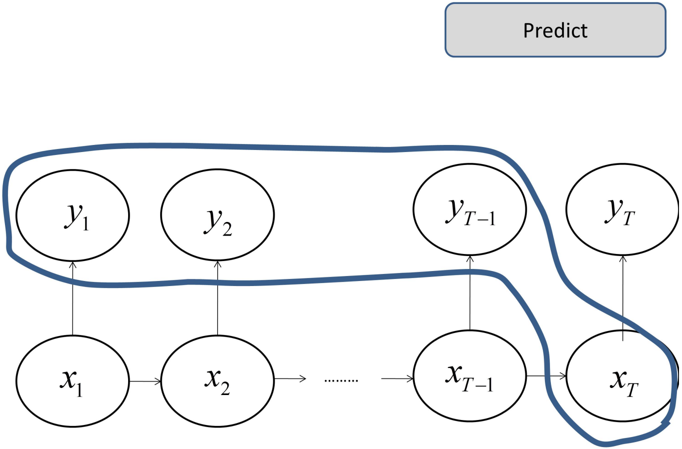
Filter

Filtering

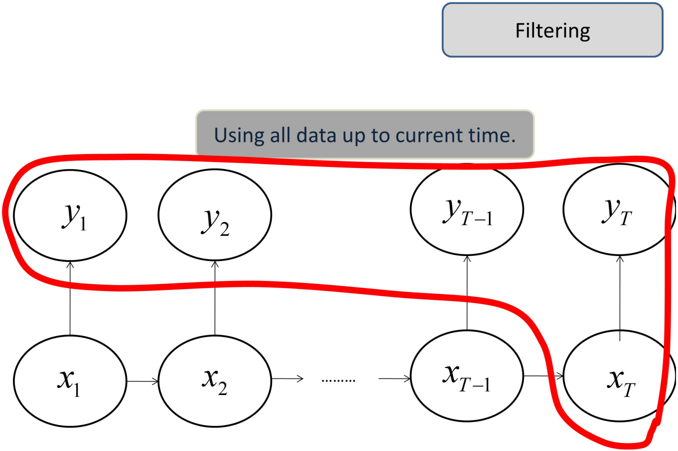
Using all data up to current time.



Predict



Predict


Filtering
Using all data up to current time.
y_1
y_2
y_{T-1}
y_T
x_1
x_2
....
x_{T-1}
x_T

Prediction, filtering and smoothing

$$p(\boldsymbol{x}_t|\boldsymbol{y}_1, \dots \boldsymbol{y}_{t-1}) = \int p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1, \dots, \boldsymbol{y}_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_1, \dots \boldsymbol{y}_t) = \frac{p(\boldsymbol{x}_t|\boldsymbol{y}_1, \dots, \boldsymbol{y}_{t-1}) p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}, \dots, \boldsymbol{y}_1)}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_1, \dots \boldsymbol{y}_T), \qquad p(\boldsymbol{x}|\boldsymbol{y}_1, \dots, \boldsymbol{y}_T)$$

Exact closed-form solutions:

- Discrete state space models (Markov chain)
- Gaussian linear models (Kalman filter, smoother)

Not so easy for other models. Need Monte Carlo methods.

Special case: Linear Gaussian model assumptions

Conditional independence in process (state) model:

$$\boldsymbol{x}_t | \boldsymbol{x}_{t-1} \sim N(\boldsymbol{F}_t \boldsymbol{x}_{t-1}, \boldsymbol{Q}_t)$$

Simplest setting (static model): $\boldsymbol{x}_t = \boldsymbol{x}_{t-1}$

Conditional independence in measurement model:

$$\boldsymbol{y}_t | \boldsymbol{x}_t \sim N(\boldsymbol{G}_t \boldsymbol{x}_t, \boldsymbol{R}_t)$$

Kalman filter

Elegant form for building the Gaussian distribution for prediction and filtering/analysis/assimilation:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_1, ...\boldsymbol{y}_{t-1}) = \int p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})d\boldsymbol{x}_{t-1}$$

$$\boldsymbol{x}_t|\boldsymbol{y}_1, ...\boldsymbol{y}_{t-1} \sim N(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$$

$$\boldsymbol{x}_t|\boldsymbol{y}_1, ...\boldsymbol{y}_t \sim N(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$$

Kalman filter : Prediction step

With linear expectation and Gaussian additive noise, the models remain Gaussian. Need mean and covariance.
$\boldsymbol{\delta}_t \sim N(0, \boldsymbol{Q}_t)$

$$\boldsymbol{\mu}_{t|t-1} = E(\boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{\delta}_t | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}) = \boldsymbol{F}_t \boldsymbol{\mu}_{t-1|t-1}$$

$$\boldsymbol{\Sigma}_{t|t-1} = \mathrm{Var}(\boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{\delta}_t | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}) = \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1|t-1} \boldsymbol{F}_t^T + \boldsymbol{Q}_t$$

Kalman filter update : Joint Gaussian

$p(\boldsymbol{x}_t, \boldsymbol{y}_t | \boldsymbol{y}_1, ...\boldsymbol{y}_{t-1})$ is joint Gaussian

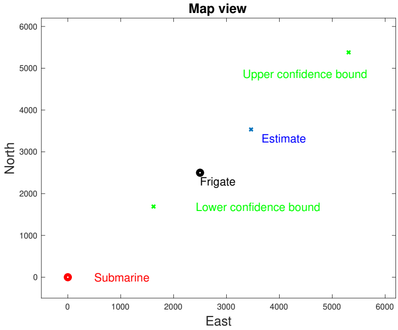
$$\begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{y}_t \end{pmatrix} | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1} \sim N\left[\begin{pmatrix} \boldsymbol{\mu}_{t|t-1} \\ \boldsymbol{G}_t \boldsymbol{\mu}_{t|t-1} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{t|t-1} & \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{G}_t^T \\ \boldsymbol{G}_t \boldsymbol{\Sigma}_{t|t-1} & \boldsymbol{G}_t \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{G}_t^T + \boldsymbol{R}_t \end{pmatrix}\right]$$

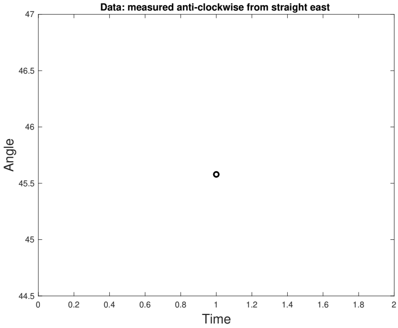
$$[\boldsymbol{x}_t | \boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_1] \sim N(\boldsymbol{\mu}_{t|t-1} + \boldsymbol{K}_t(\boldsymbol{y}_t - \boldsymbol{G}_t \boldsymbol{\mu}_{t|t-1}), \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{K}_t \boldsymbol{G}_t \boldsymbol{\Sigma}_{t|t-1})$$

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{G}_t^T [\boldsymbol{G}_t \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{G}_t^T + \boldsymbol{R}_t]^{-1}$$

Example from Target Tracking

- A submarine measures the bearing to a target (frigate).
- From bearings-only data, it attempts to track the frigate.





Example from Target Tracking : Model

Position of frigate: $\boldsymbol{x}_t = (\text{North}_t, \text{East}_t, \text{NorthVelocity}_t, \text{EastVelocity}_t)'$.
$\boldsymbol{x}_1 \sim N(0, \boldsymbol{Q}_0)$.
Dynamical model:

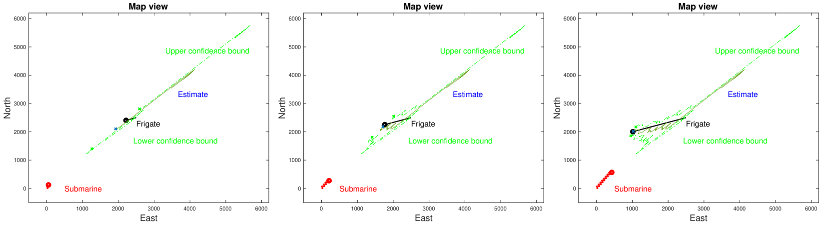
$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1 & 0 & \delta & 0 \\ 0 & 1 & 0 & \delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_t + \boldsymbol{v}_t, \boldsymbol{v}_t \sim N(0, \boldsymbol{Q})$$

Data equation: $y_t = \arctan \frac{\text{North}_t - \text{NorthSUB}_t}{\text{East}_t - \text{EastSUB}_t} + w_t,\ w_t \sim N(0, r^2)$

Example from Target Tracking : Filtering distribution

- The nonlinear equation model is linearized in the solution - Extended Kalman filter.
- Filtering density $p(\boldsymbol{x}_t | y_1, \ldots, y_t)$ is approximate Gaussian.

Example from Target Tracking : Results



Monte Carlo sampling for filtering

Common sequential Monte Carlo methods:

- Particle filtering : re-weighting of realizations based on data-match. Pros: exact in the asymptotic limit. Cons: challenging to make it work for high-dimensional methods.
- Ensemble Kalman filtering : moves realizations based on correlations with data. Pros: often works well in high-dimensional systems. Cons: no guarantee of performance, even in the number of ensembles go to infinity.

Ensemble Kalman filter approach

- Method for highly nonlinear dynamical models or measurements models.
- The forward models are black-box models. (No explicit form.)
- Using Monte Carlo realizations to represent probability distribution.
- The updating of ensembles is based on correlations between state variables and data.

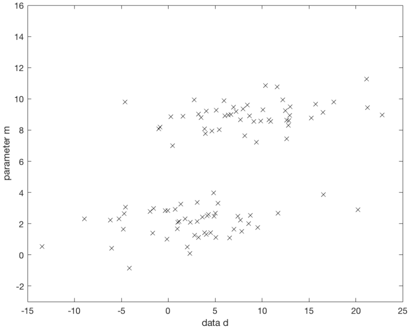
Ensemble-based Kalman approximation

- Ensemble size B. Repeat for $t = 2, \ldots, N$
- $\boldsymbol{x}_{t-1}^b$, $b = 1, \ldots, B$ approximately from $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$.
- Predictive realizations $\boldsymbol{x}_t^b = f(\boldsymbol{x}_{t-1}^b; \boldsymbol{\delta}_t)$, $b = 1, \ldots, B$.
- Predictive data $\boldsymbol{y}_t^b = \boldsymbol{g}(\boldsymbol{x}_t^b) + \boldsymbol{\epsilon}_t$, $\boldsymbol{\epsilon}_t \sim N(0, \boldsymbol{R}_t)$.
- Kalman weight matrix $\hat{\boldsymbol{K}}_t = \hat{\boldsymbol{\Sigma}}_{xy,t}\left(\hat{\boldsymbol{\Sigma}}_{yy,t} + \boldsymbol{R}_t\right)^{-1}$ determined empirically from forecast ensembles $(\boldsymbol{x}_t^b, \boldsymbol{y}_t^b)$, $b = 1, \ldots, B$.
- Kalman update of bth ensemble member at step t

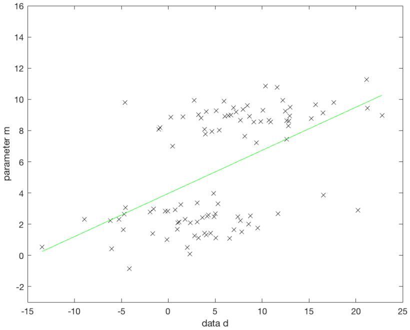
$$\boldsymbol{x}_t^b = \boldsymbol{x}_t^b + \hat{\boldsymbol{K}}_t(\boldsymbol{y}_t - \boldsymbol{y}_t^b)$$

Univariate example - forecast samples

$$x^b \sim p(x), \qquad y^b = x^b + N(0, 5^2)$$

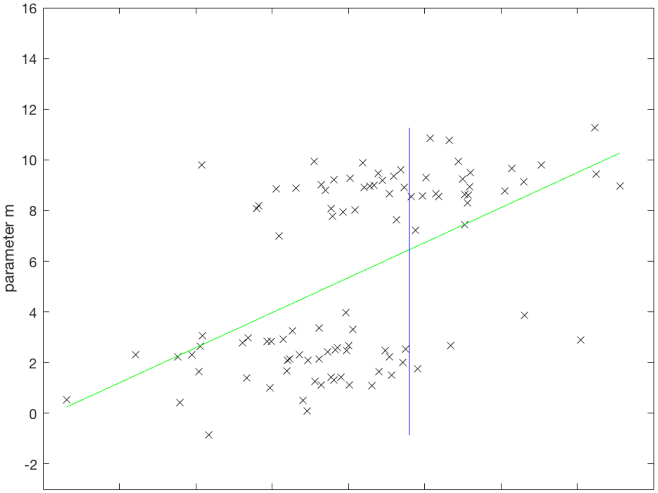


Univariate example - regression fit

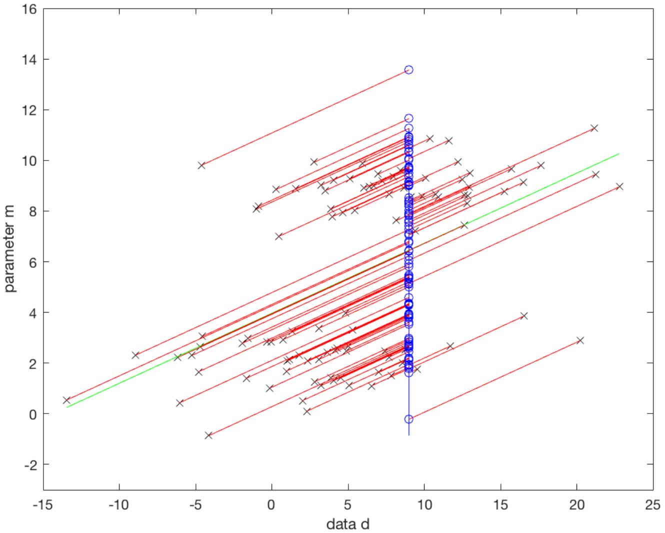


Univariate example - observation

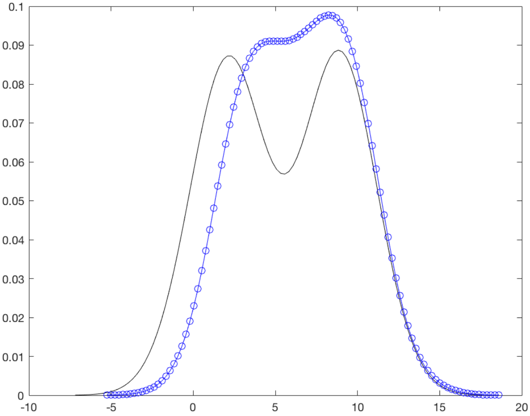
$y = 9.$



Univariate example - analysis or update step



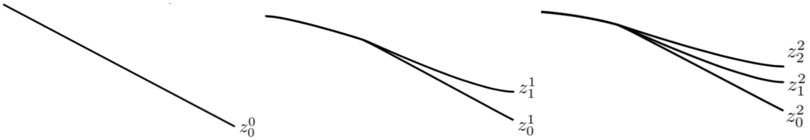
Univariate example - prior and posterior



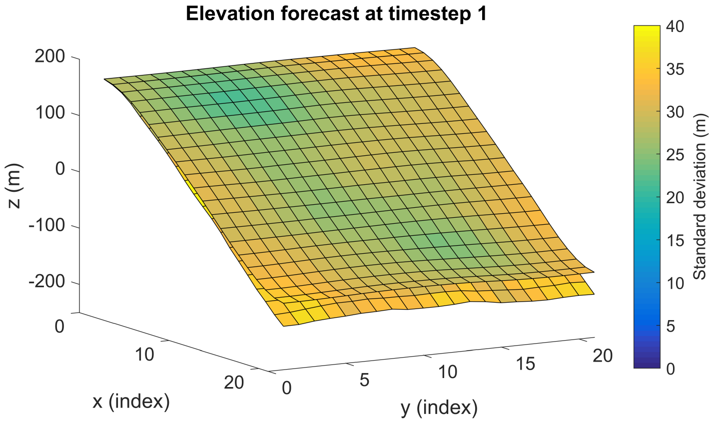
Geologic process models

Differential equation for sedimentation, corrected with data.

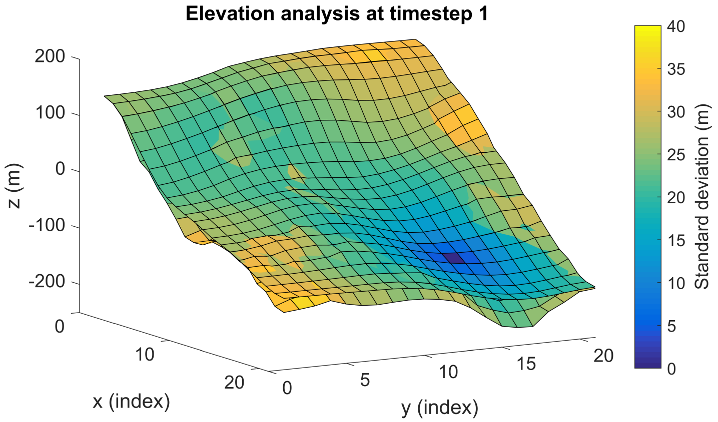
- Start with initial surface z_0^0 at time t_0
- Surface will "diffuse" to yield new top surface z_1^1 at time t_1
- Elevation of surface j at time t_k is z_j^k for $j = 1, \dots, k$



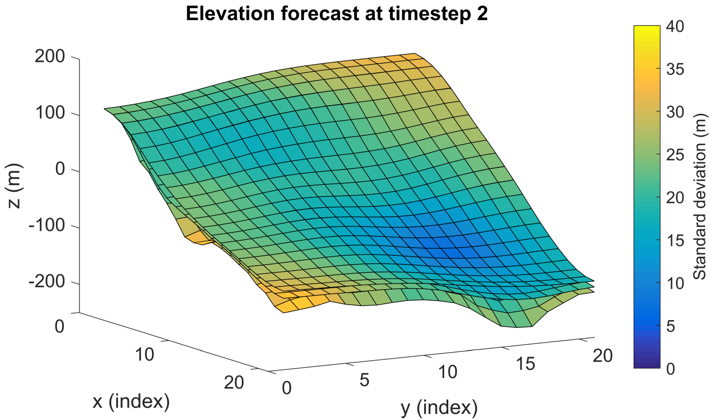
Time evolution of ensemble



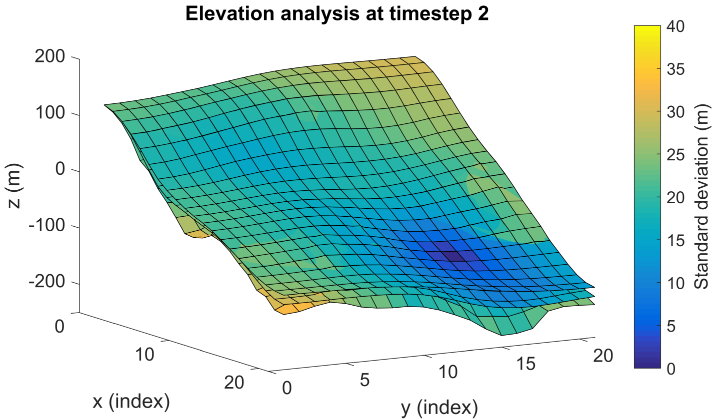
Time evolution of ensemble



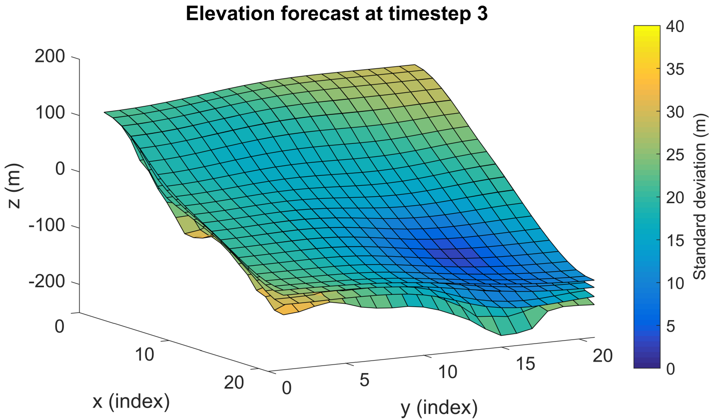
Time evolution of ensemble



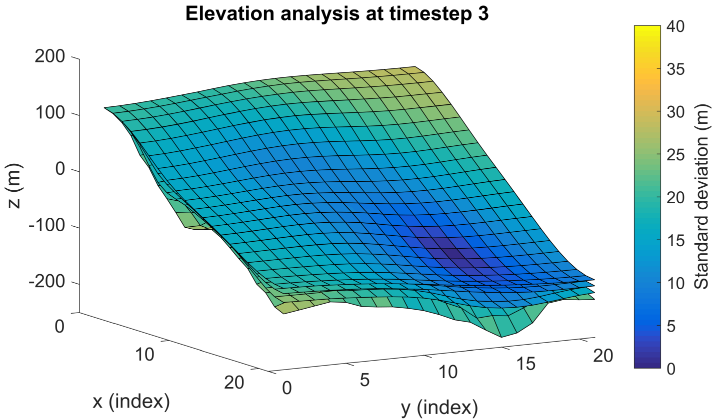
Time evolution of ensemble



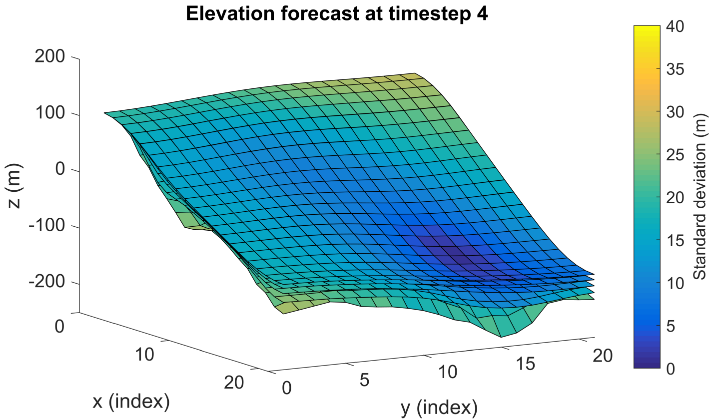
Time evolution of ensemble



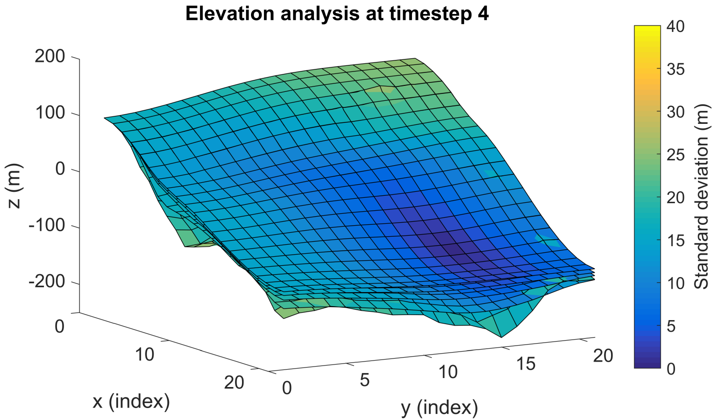
Time evolution of ensemble



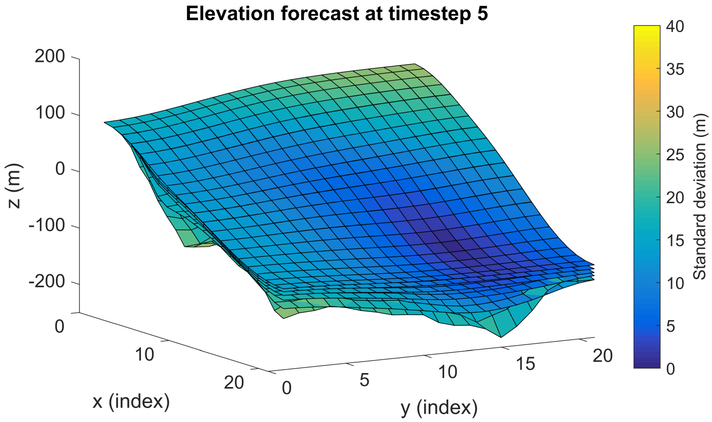
Time evolution of ensemble



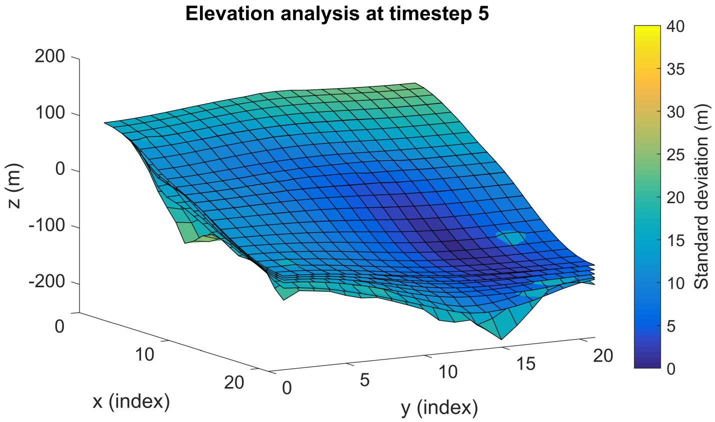
Time evolution of ensemble



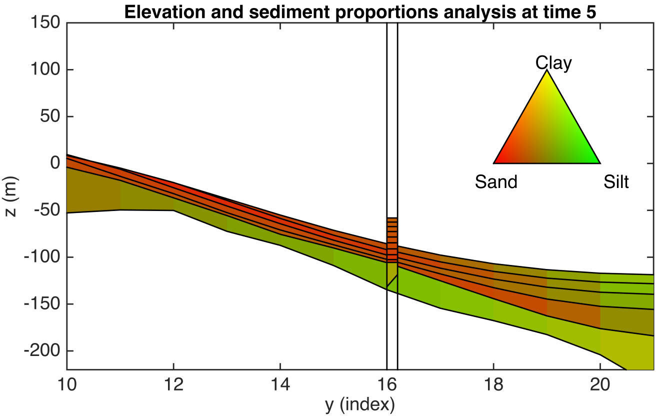
Time evolution of ensemble



Time evolution of ensemble

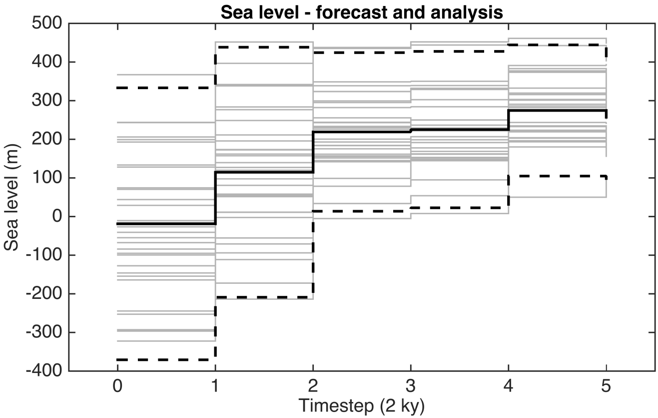


Time evolution of ensemble

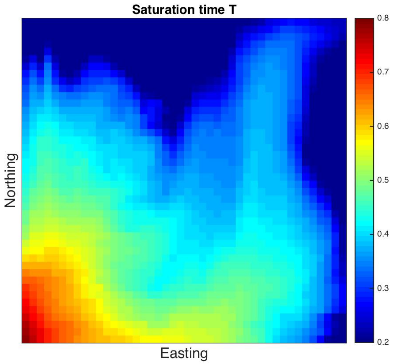


Time evolution of ensemble

Sea level parameter - constant: $\theta(t) = \theta_0,\ t_{\text{start}} \leq t \leq t_{\text{end}}$



Reservoir simulation example

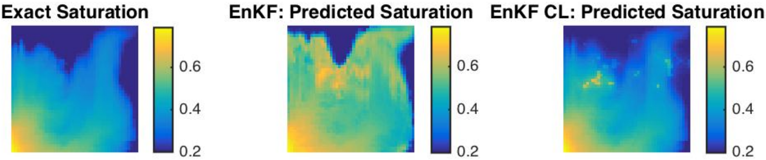


Repeated seismic data assimilation.

EnKF approximation

- Generate B realizations of porosities, permeabilities and initial saturation. Repeat the following over time:
- Forecast saturations with fluid flow simulator, for all realizations.
- Forecast seismic data for all realizations, using geophysical relations.
- Use forecast reservoir variables and seismic data to train the Kalman gain.
- Update ensemble members using the Kalman update and the observed seismic response.

Reservoir example results (standard and localized version)



Course summary

- Statistical models and concepts
- Statistical dependence and graphical models
- Linear Bayesian inversion
- Markov chain Monte Carlo sampling.
- State space models and Bayesian filtering.