Bayes–Kalman update equations.

**Prediction**

\[ p(x_{t+1} | y_t) = \sum_{x_t} p(x_{t+1} | x_t) p(x_t | y_t) \]

**Correction**

\[ p(x_{t+1} | y_{t+1}) = \frac{p(y_{t+1} | x_{t+1}) p(x_{t+1} | y_t)}{p(y_{t+1} | y_t)} \]

with

\[ p(y_{t+1} | y_t) = \sum_{x_{t+1}} p(y_{t+1} | x_{t+1}) p(x_{t+1} | y_t) \]

Problem: Hard to compute Bayes–Kalman update.

If the distributions are Gaussians—prior and likelihood—then both stages can be reduced to algebra, previous lecture.
This motivates a sampling approach based on particle filters (sometimes called Bootstrap Filters).

Represent the distribution, e.g., \( p(x_{t+1}|y_t) \), by a set of particles \( \{x_1^t, \ldots, x^m_t\} \). These are random samples from \( p(x_{t+1}|y_t) \), so big density of particles at places of high probability and small density elsewhere.

\[ \text{many samples} \]  
\[ \text{few samples} \]

(a) Draw samples \( \{x^j_{t+1}; j=1,m\} \) (predict)

from \( p(x_{t+1}|x_t^{(j)}) \), for \( j=1 \ldots m \).

(b) Weight each sample by \( \omega^{(j)} \propto p(y_t|x_{t+1}^{(j)}) \) (uses the new observation \( y_{t+1} \)).

(c) Resample from \( \{x^*_1, \ldots, x^*_m\} \) with probability proportional to \( \omega^{(j)} \) to produce random sample \( \{x^*_1, \ldots, x^*_m\} \) for time \( t+1 \).

Claim: It can be shown that if the \( \{x^{(j)}_t\} \) follow \( P(x_{t+1}|y_t) \) and if \( m \) is sufficiently big, then the \( \{x^*_1, \ldots, x^*_m\} \) follow \( P(x_{t+1}|y_{t+1}) \).
Note: the weights are big for particles which are consistent with the new observation \( y^i_{t+1} \), i.e. for \( x^{*i}_{t+1} \), \( p(y^i_{t+1} \| x^{*i}_{t+1}) \) is large and weights are small for particles which are inconsistent.

Note: this is like importance sampling, you sample from \( p(x_{t+1} \| y_t) \) (like gl(1) in importance sampling) when you really want to sample from \( p(x_{t+1} \| y_{t+1}) \).

Note: you can extend particle filtering to cases where the distribution change with time, e.g. \( p_t(x_{t+1} \| x_t), p_t(y_{t+1} \| x_t) \).

Particle Filters/Bootstrap were developed in the past 10 years. They have had considerable success.

Limitation:

(a) They do not use the current available information \( y_{t+1} \) in the sampling step.
(b) The use of resampling may cause inefficiency.

Some concern about how well they work in high dim.
(5) Bootstrap / Particle Filters are a special case of Sequential Monte Carlo.

(Ch. 3 of Jian Liu).

Justification for Claim that Particle Filter give samples from $p(x_{t+1} | y_{t+1})$.

1. Prediction: $p(b,a) = p(b | a)p(a)$

- samples $\{ (b_i, a_i) \}$ from $p(b,a)$
  - sample $a_i$ from $p(a)$
  - $b_i$ from $p(b | a)$

Then $(b_i)$ are samples from $p(b)$.

2. Correction: $p(a | b) = \frac{p(b | a)p(a)}{p(b)}$

- sample $\{ a_i \}$ from $p(a)$
- accept each sample with prob $\alpha p(b_i | a_i)$
- gives new samples $\{ a_i^* \}$ from $p(a | b)$