

時系列解析 (13)

－粒子フィルター－
配布用

東京大学 数理・情報教育研究センター
北川 源四郎

非線形・非ガウス型フィルタの計算量

m 状態の次元
 ℓ システムノイズの次元
 k_i 分点数
 k_{q_j} システムノイズの分点数

$$\text{計算量} \sim (k_1 \times \dots \times k_m)(k_{q_1} + \dots + k_{q_\ell})n \sim \ell k^{m+1} n$$

k	1	2	3	4	5	6	7
10	100	1K	10K	100K	1M	10M	100M
30	1K	30K	1M	30M	1G	30G	1T
100	10K	1M	100M	10G	1T	100T	10P
300	100K	30M	10G	3T	1P	300P	
1000	1M	1G	1T	1P			

分布の近似

0. ガウス近似

(拡張)カルマンフィルタ・平滑化

1. 区分的線形(階段)関数近似

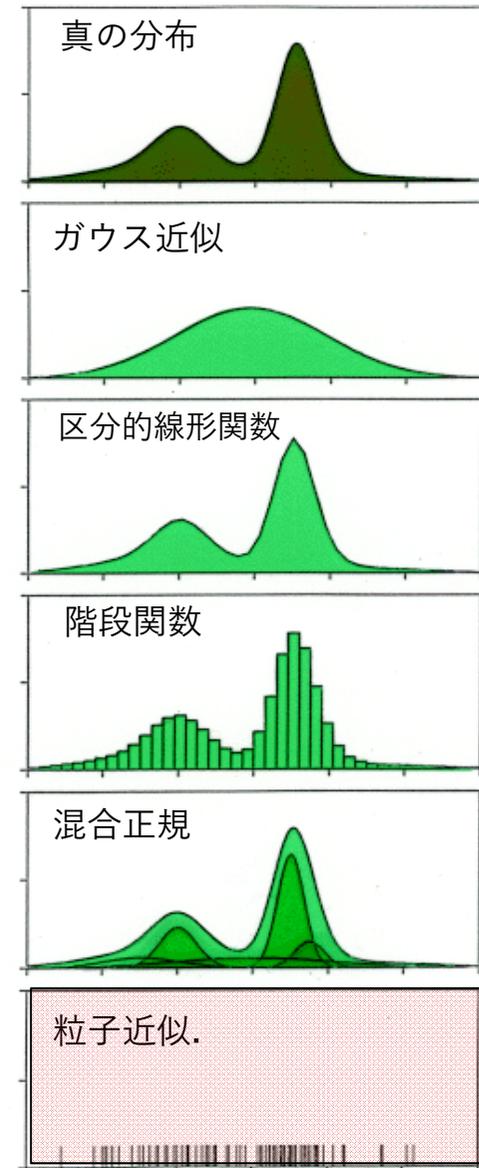
非ガウス型フィルタ・平滑化

2. 混合ガウス近似

ガウス和フィルタ・平滑化

3. 粒子近似

モンテカルロフィルタ・平滑化



モンテカルロ計算の長所, 欠点

長所

- 分布の計算（畳み込み積分, 非線形変換）は大変で時間がかかるが, 粒子ごとの代入計算は簡単で早い
- 状態の次元が増えても計算量はそれほど増えない

問題点

- 常にサンプリング誤差を伴う
- 粒子数 m を増やしても分散減少は $1/m$

粒子による分布の近似

$$\{p_n^{(1)}, \dots, p_n^{(m)}\} \sim p(x_n | Y_{n-1}) \quad \text{予測分布}$$

$$\{f_n^{(1)}, \dots, f_n^{(m)}\} \sim p(x_n | Y_n) \quad \text{フィルタ分布}$$

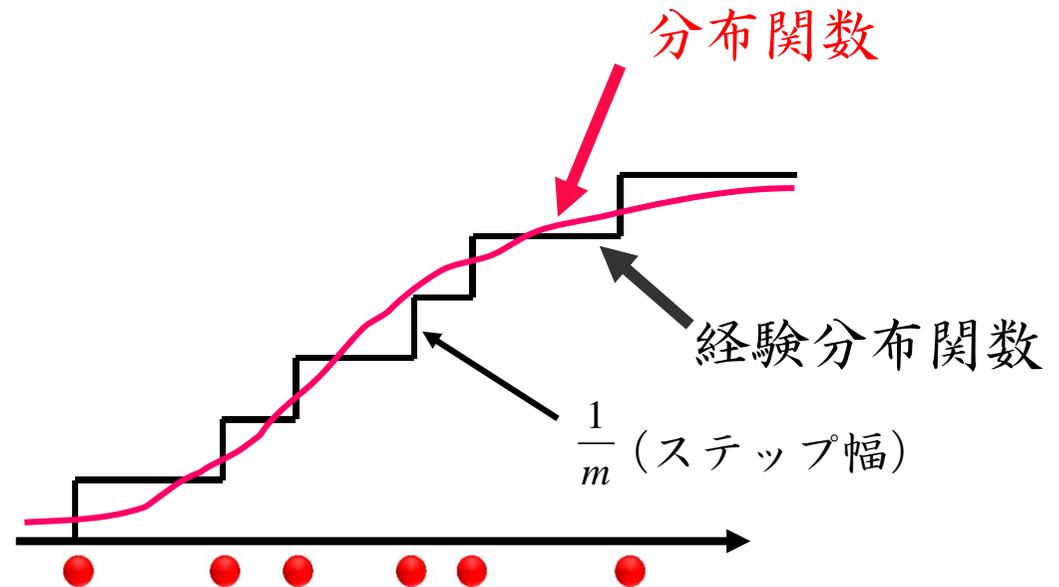
$$\{s_{n|N}^{(1)}, \dots, s_{n|N}^{(m)}\} \sim p(x_n | Y_N) \quad \text{平滑化分布}$$

$$\{v_n^{(1)}, \dots, v_n^{(m)}\} \sim p(v_n) \quad \text{システムノイズ分布}$$

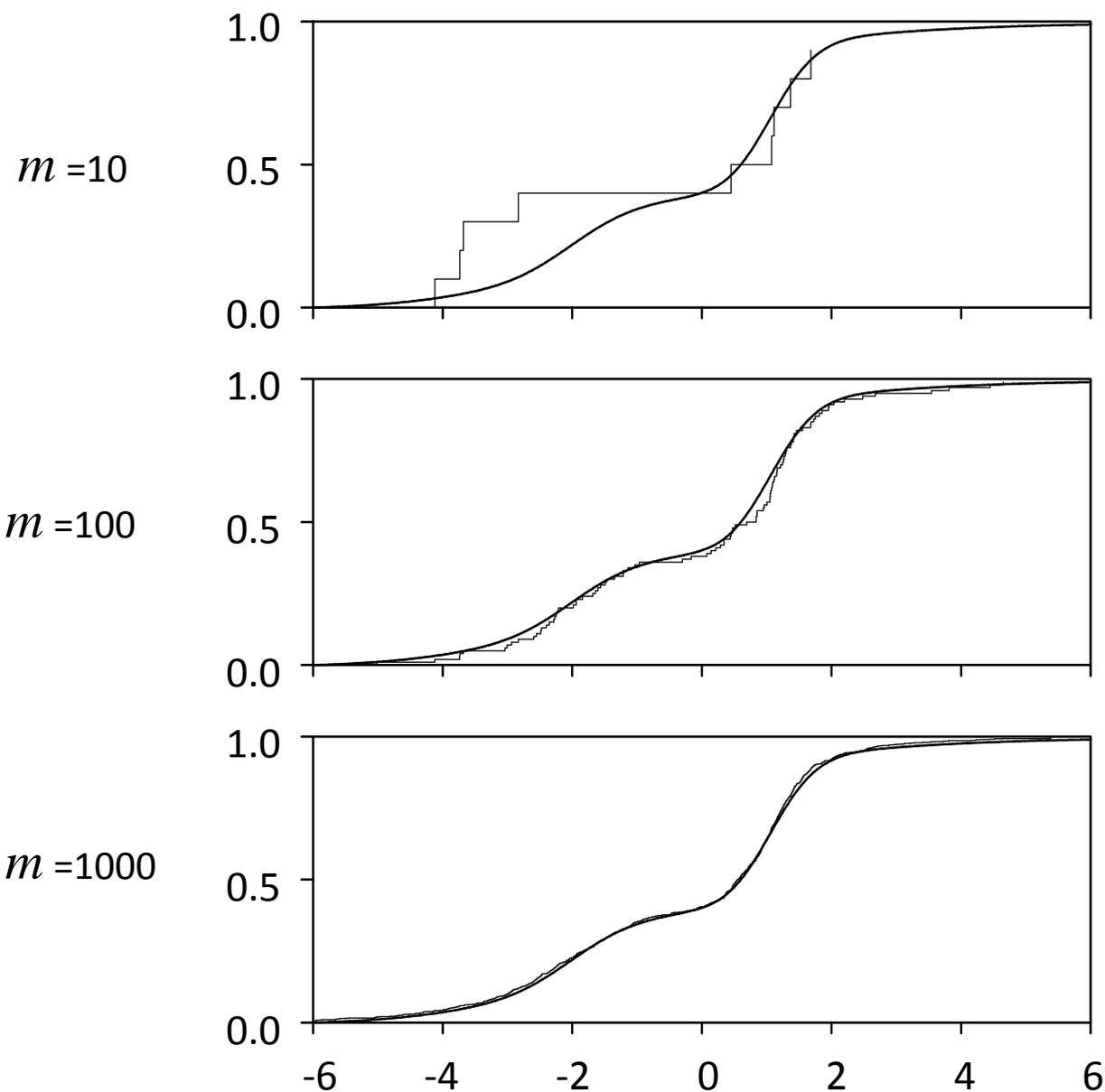
$$\{p_n^{(1)}, \dots, p_n^{(m)}\} \sim p(x_n | Y_{n-1})$$

$$\longleftrightarrow F_n(x) = \frac{1}{m} \sum_{j=1}^m I(x; p_n^{(j)})$$

$$\longleftrightarrow \Pr(X_n = p_n^{(j)} | Y_{n-1}) = \frac{1}{m}$$



経験分布関数による分布の近似



一期先予測の導出

$$\begin{aligned} p(x_n | Y_{n-1}) &= \iint p(x_n, x_{n-1}, v_n | Y_{n-1}) dx_{n-1} dv_{n-1} \\ &= \iint p(x_n | x_{n-1}, v_n, Y_{n-1}) p(v_n | x_{n-1}, Y_{n-1}) p(x_{n-1} | Y_{n-1}) dx_{n-1} dv_{n-1} \\ &= \iint \delta(x_n - F(x_{n-1}, v_n)) p(v_n) p(x_{n-1} | Y_{n-1}) dx_{n-1} dv_{n-1} \end{aligned}$$

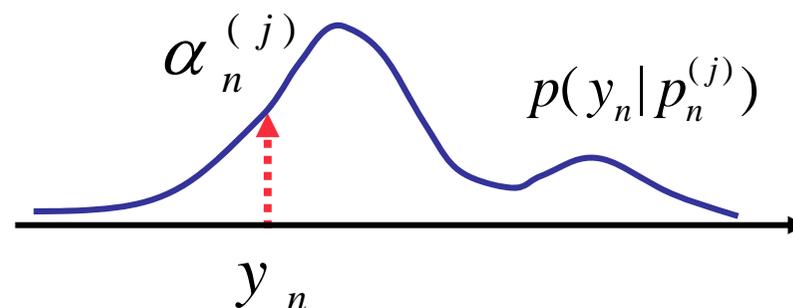
$$\begin{aligned} f_{n-1}^{(j)} &\sim p(x_{n-1} | Y_{n-1}) \\ v_n^{(j)} &\sim p(v) \end{aligned}$$

$$p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)}) \sim p(x_n | Y_{n-1})$$

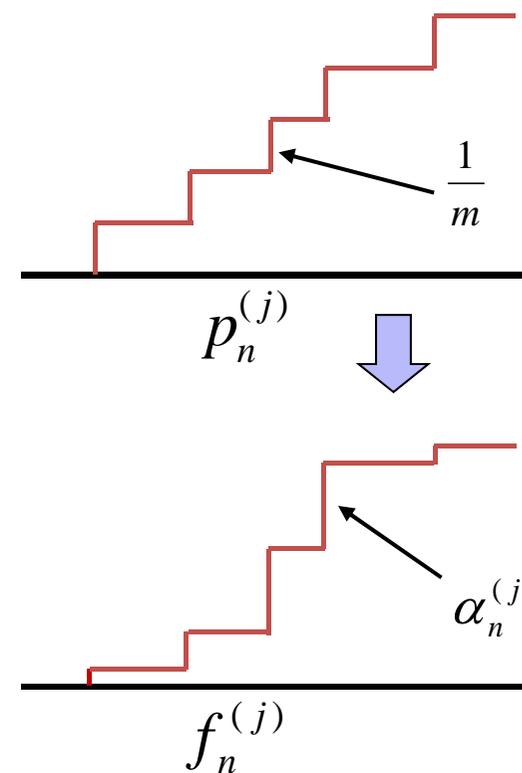
フィルタ

$\alpha_n^{(j)}$: 粒子 $p_n^{(j)}$ の重み係数

$$\alpha_n^{(j)} = p(y_n | X_n = p_n^{(j)})$$



$$\begin{aligned} \Pr(X_n = p_n^{(j)} | Y_n) &= \Pr(X_n = p_n^{(j)} | Y_{n-1}, y_n) \\ &= \frac{\Pr(X_n = p_n^{(j)}, y_n | Y_{n-1})}{\Pr(y_n | Y_{n-1})} \\ &= \frac{\Pr(y_n | X_n = p_n^{(j)}) \Pr(X_n = p_n^{(j)} | Y_{n-1})}{\sum_{i=1}^m \Pr(y_n | X_n = p_n^{(i)}) \Pr(X_n = p_n^{(i)} | Y_{n-1})} \\ &= \frac{\alpha_n^{(j)} \frac{1}{m}}{\sum_{i=1}^m \alpha_n^{(i)} \frac{1}{m}} = \frac{\alpha_n^{(j)}}{\sum_{i=1}^m \alpha_n^{(i)}} \end{aligned}$$



リサンプリング

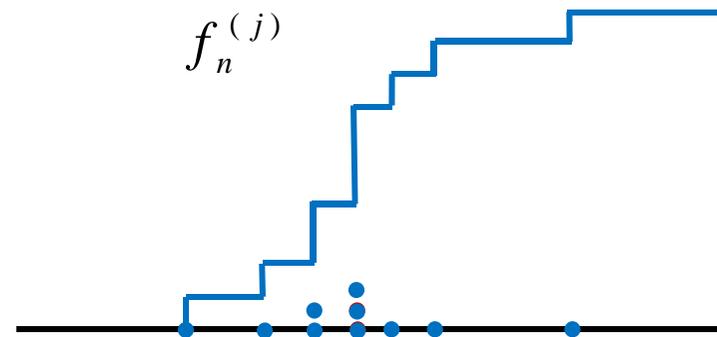
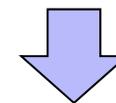
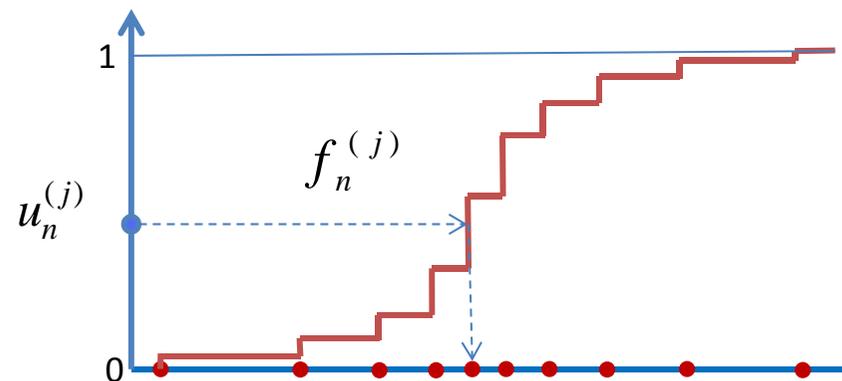
m 個の重み付き粒子で近似されたフィルタ分布を等確率の m 個の粒子で近似しなおす

(a) 一様乱数 $u_n^{(j)} \in U[0,1)$ を生成する。

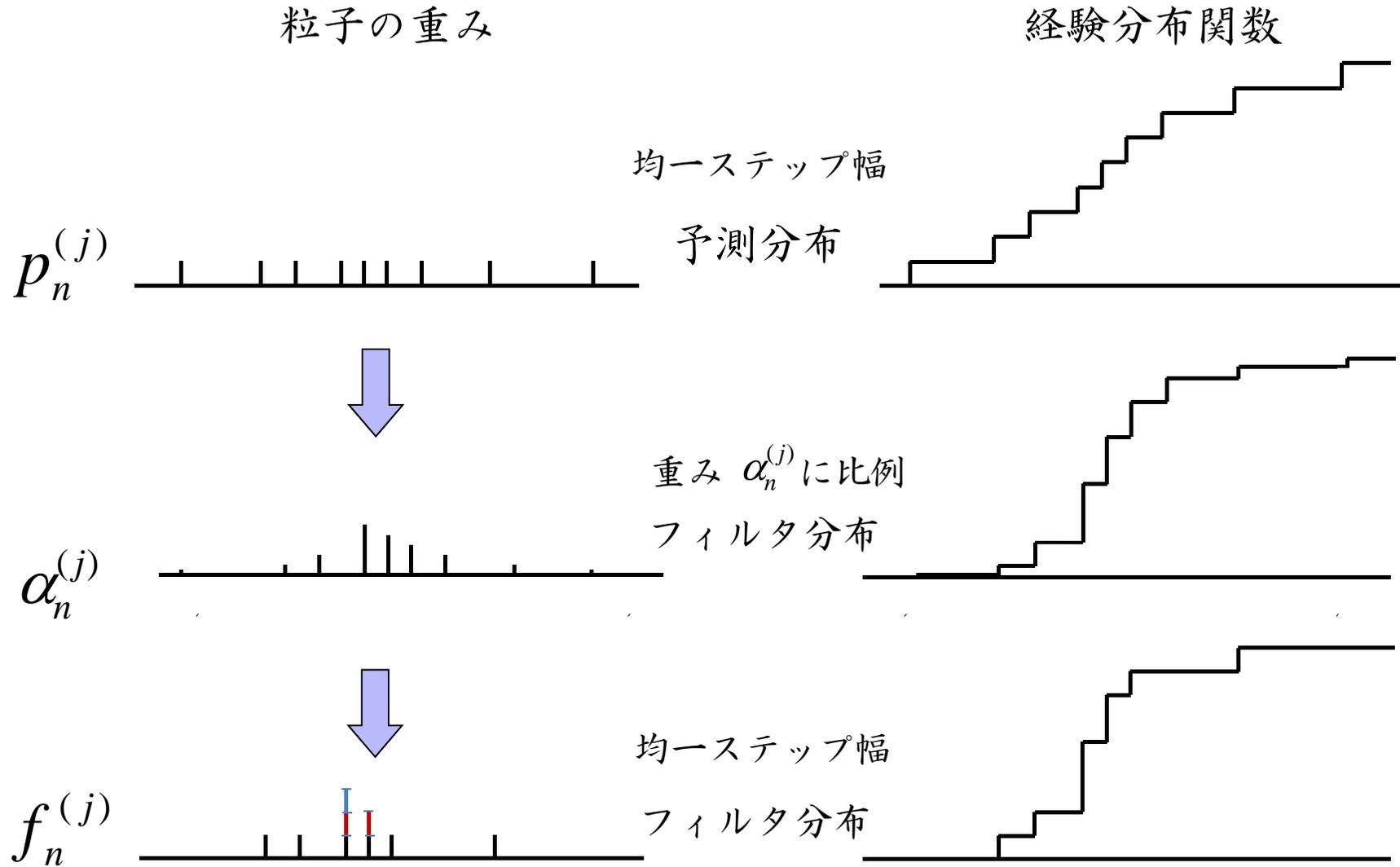
(b) $\frac{1}{C} \sum_{k=1}^{i-1} \alpha_n^{(k)} < u_n^{(j)} \leq \frac{1}{C} \sum_{k=1}^i \alpha_n^{(k)}$ を満たす i を探す。

ただし, $C = \sum_{k=1}^m \alpha_n^{(k)}$

(c) フィルタの粒子を $f_n^{(j)} = p_n^{(i)}$ とする。



フィルタリング



粒子フィルタ (MCF)

- ・ システムノイズ

$$v_n^{(j)} \sim p(v) \quad j = 1, \dots, m$$

- ・ 予測分布

$$p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)})$$

- ・ 重要度 (ベイズ係数)

$$\alpha_n^{(j)} = p(y_n | p_n^{(j)})$$

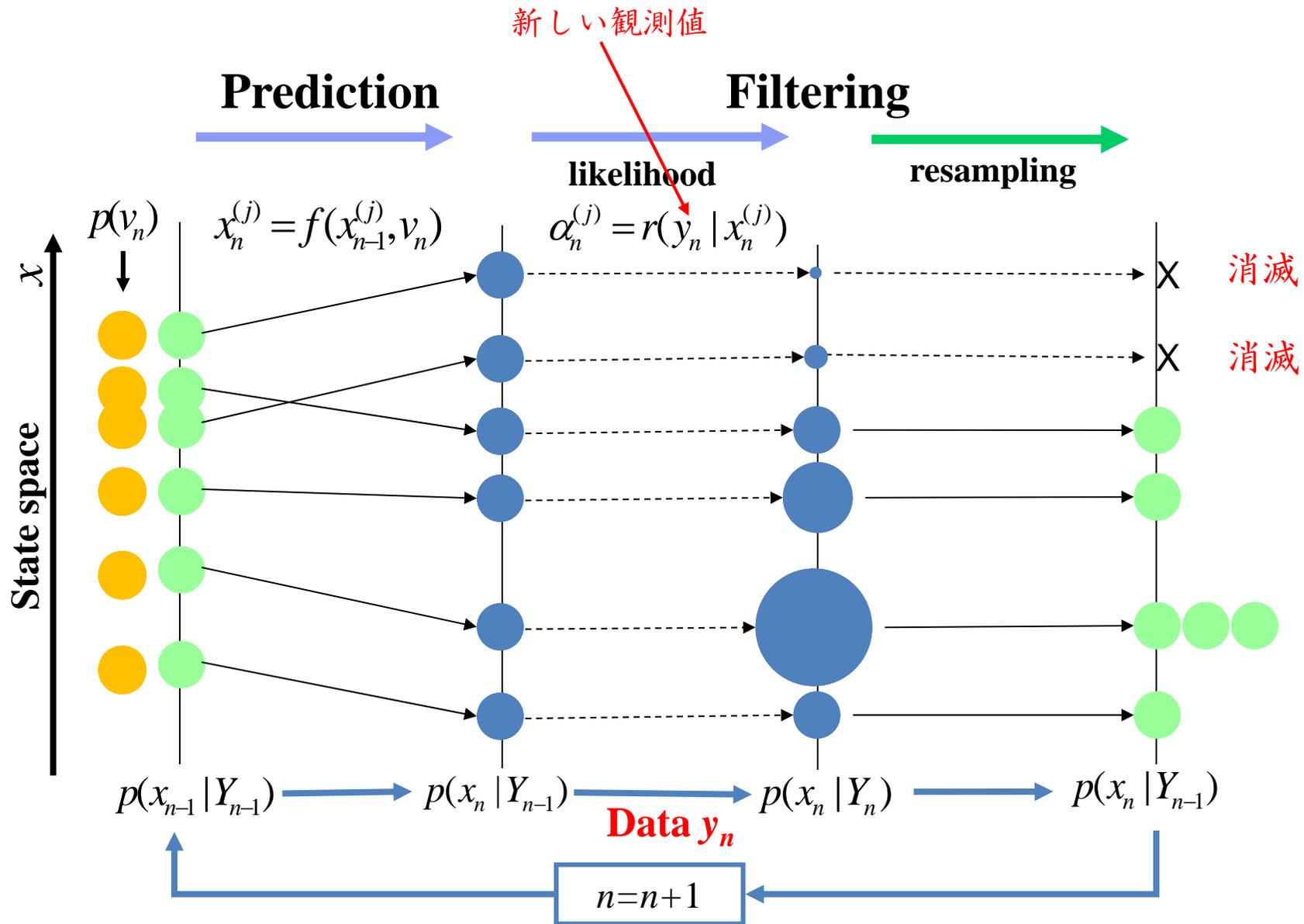
- ・ フィルタ分布のリサンプリング

$$\{p_n^{(j)}\} \quad \longrightarrow \quad \{f_n^{(j)}\}$$

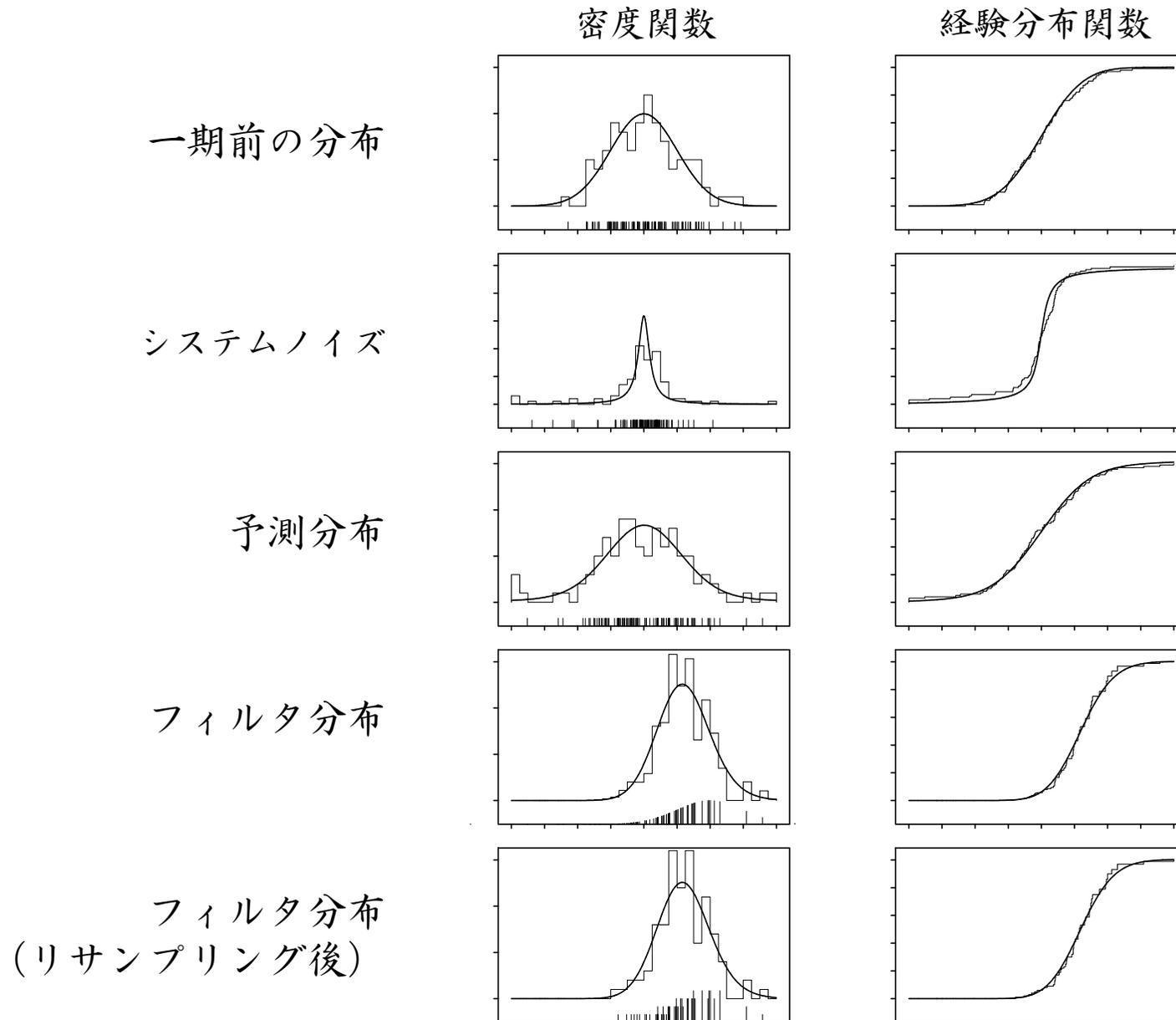
Gordon et al. (1993), Kitagawa (1996)

Doucet, de Freitas and Gordon (2001) “Sequential Monte Carlo Methods in Practice”

One Cycle of Monte Carlo Filtering



One Cycle of Particle Filter

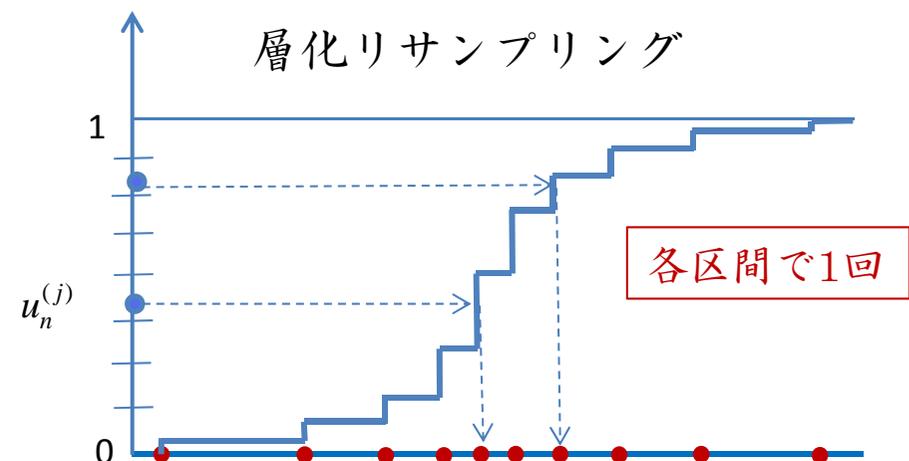
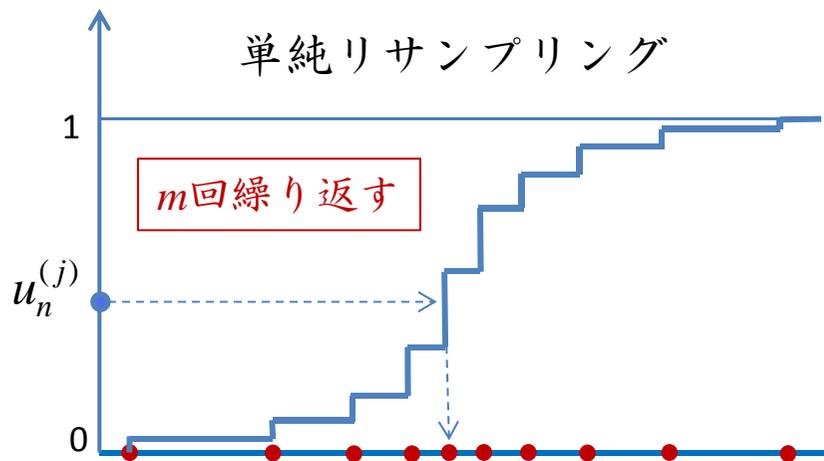


層化リサンプリング

リサンプリングの目的は重み $\{\alpha_n^{(1)}, \dots, \alpha_n^{(m)}\}$ を持つ粒子 $\{p_n^{(1)}, \dots, p_n^{(m)}\}$ で定義される分布関数を，重みが等しい経験分布で表現しなおすことであり，厳密なランダムサンプリングは必ずしも必要ではない。

(a') 一様乱数 $u_n^{(j)} \in U\left[\frac{j-1}{m}, \frac{j}{m}\right)$ を生成する。

(a'') 固定した $\alpha \in [0, 1)$ について， $u_n^{(j)} = \frac{j-1}{m}$ とする。



Comparison of the Resampling Methods

D_f^* : Exact filter distribution

m		$J(D_f, D_p)$			$J(D_p, D_f^*)$
		Random	Stratified	Deterministic	
10	Sort	0.0326	0.00379	0.00176	0.1021
	No-sort	0.0321	0.00859	0.00513	0.1021
100	Sort	0.00381	0.636×10^{-3}	0.311×10^{-4}	0.860×10^{-2}
	No-sort	0.00405	0.939×10^{-3}	0.612×10^{-3}	0.860×10^{-2}
1,000	Sort	0.398×10^{-3}	0.838×10^{-6}	0.407×10^{-6}	0.102×10^{-2}
	No-sort	0.379×10^{-3}	0.982×10^{-4}	0.611×10^{-4}	0.102×10^{-2}
10,000	Sort	0.387×10^{-4}	0.101×10^{-7}	0.498×10^{-8}	0.248×10^{-3}
	No-sort	0.406×10^{-4}	0.936×10^{-5}	0.636×10^{-5}	0.348×10^{-3}

$J(D_f, D_p)$ is negligible

$$J(D_p, D_f) = \int_{-\infty}^{\infty} (D_p(x) - D_f(x))^2 dx$$

$$= \int_{-\infty}^{\infty} \left(\frac{1}{c} \sum_{i=1}^m \alpha_n^{(i)} I(x, p_n^{(i)}) - \frac{1}{m} \sum_{i=1}^m I(x, f_n^{(i)}) \right) dx$$

$$c = \sum_{i=1}^m \alpha_n^{(i)}$$

平滑化 (粒子を保存する方法)

$$(s_{1|i}^{(j)}, \dots, s_{n|i}^{(j)})^T \sim p(x_1, \dots, x_n | Y_i)$$

$$\Pr(x_1 = s_{1|n-1}^{(j)}, \dots, x_{n-1} = s_{n-1|n-1}^{(j)} | Y_{n-1}) = \frac{1}{m}$$

$$v_n^{(j)} \sim q(v)$$

とすると、 n 個の粒子を

$$p_{i|n-1}^{(j)} = \begin{cases} s_{i|n-1}^{(j)} & \text{for } i = 1, \dots, n-1 \\ F(s_{n-1|n-1}^{(j)}, v_n^{(j)}) & \text{for } i = n \end{cases}$$

と定義すると

$$(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})^T \sim p(x_1, \dots, x_n | Y_{n-1})$$

平滑化 (粒子を保存する方法) (2)

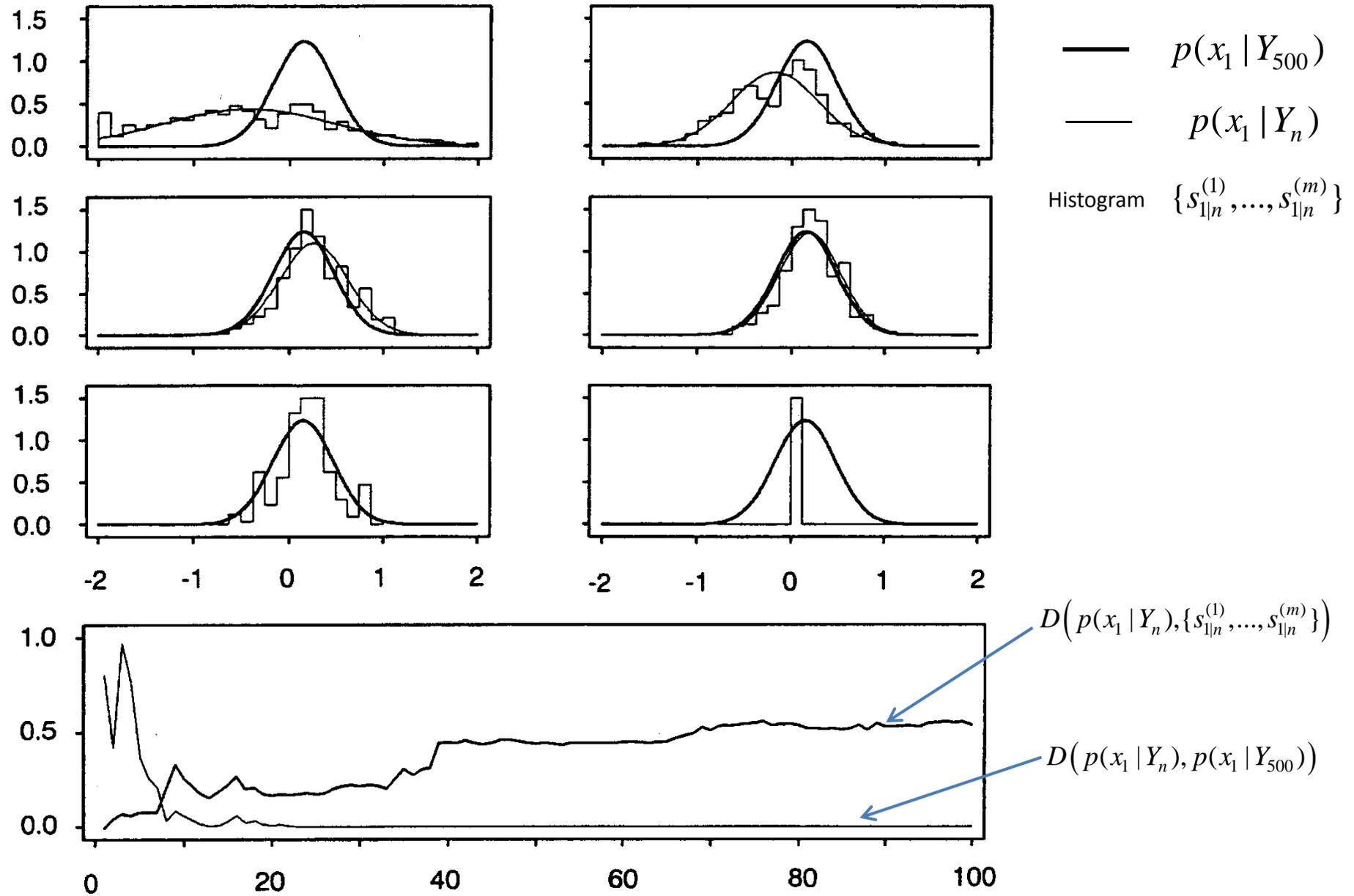
y_n 新しいデータ

$$\begin{aligned} & \Pr(x_i = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_n) \\ &= \Pr(x_i = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1}, y_n) \\ &= \frac{p(y_n | x_i = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)}, Y_{n-1}) \Pr(x_i = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1})}{\Pr(y_n | Y_{n-1})} \\ &= \frac{\Pr(y_n | p_{n|n-1}^{(j)}) \Pr(x_i = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1})}{\Pr(y_n | Y_{n-1})} \end{aligned}$$

ただし、 $p_{n|n-1}^{(j)} = p_{n|n-1}^{(j)}$

$p(x_1, \dots, x_n | Y_{n-1})$ に従う n 次元粒子 $\left\{ \left(s_{1|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_{n|n-1}^{(j)} \right)^T, j = 1, \dots, m \right\}$
のリスAMPLINGによって $p(x_1, \dots, x_n | Y_n)$ に従う等確率の n 次元粒子
 $\left\{ \left(s_{1|n}^{(j)}, \dots, s_{n-1|n}^{(j)}, s_{n|n}^{(j)} \right)^T, j = 1, \dots, m \right\}$ を生成する

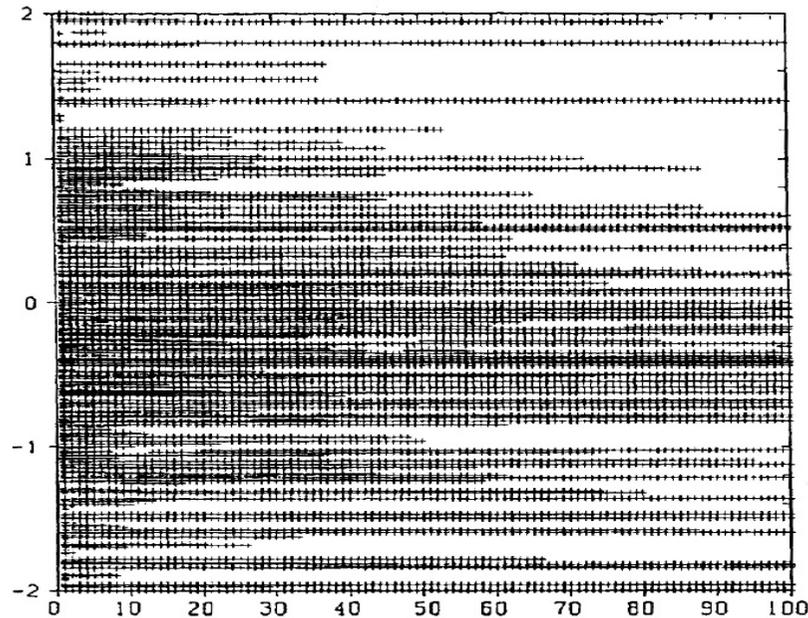
Accuracy vs. lag



平滑化の困難

問題点

- ・リサンプリングによって（実質的）粒子数が単調減少する



解決策

1. 固定ラグ平滑化を用い、ラグを大きくしない
2. 2方向フィルタを用いる

平滑化の改善

(固定区間) 平滑化

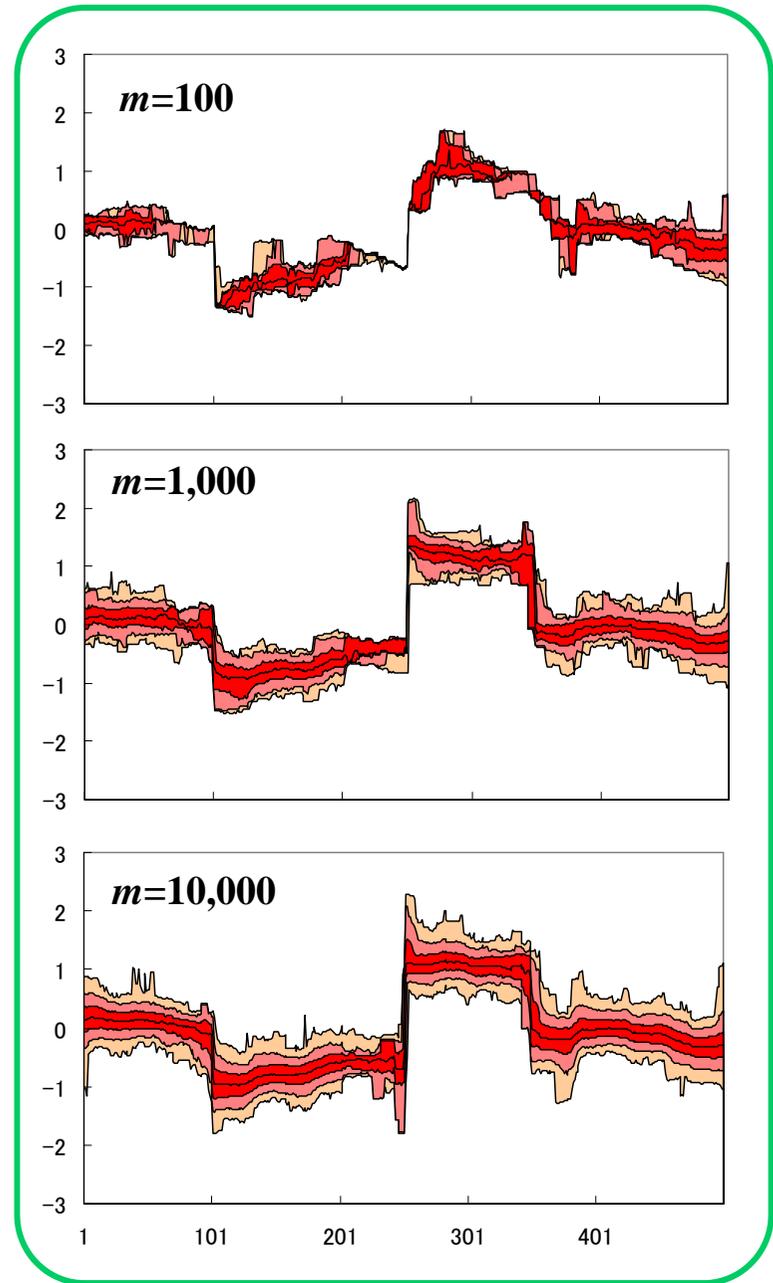
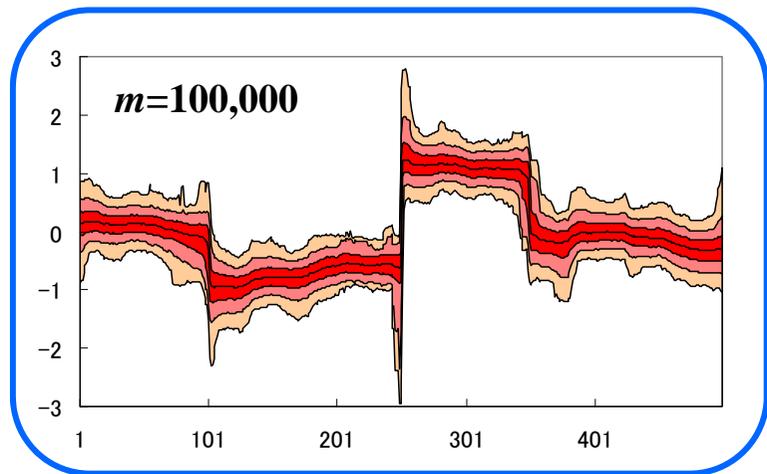
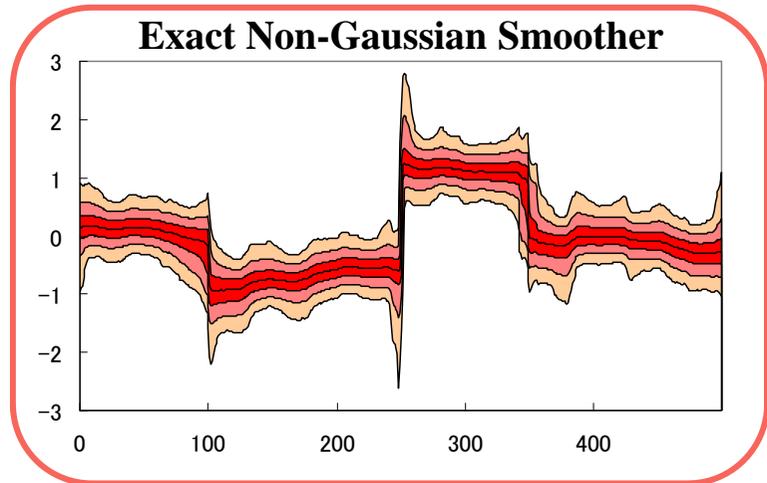
$$\left\{ s_{1|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)} \right\} \Rightarrow \left\{ s_{1|n}^{(j)}, \dots, s_{n|n}^{(j)} \right\}$$

固定ラグ平滑化

$$\left\{ s_{n-L|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)} \right\} \Rightarrow \left\{ s_{n-L|n}^{(j)}, \dots, s_{n|n}^{(j)} \right\}$$

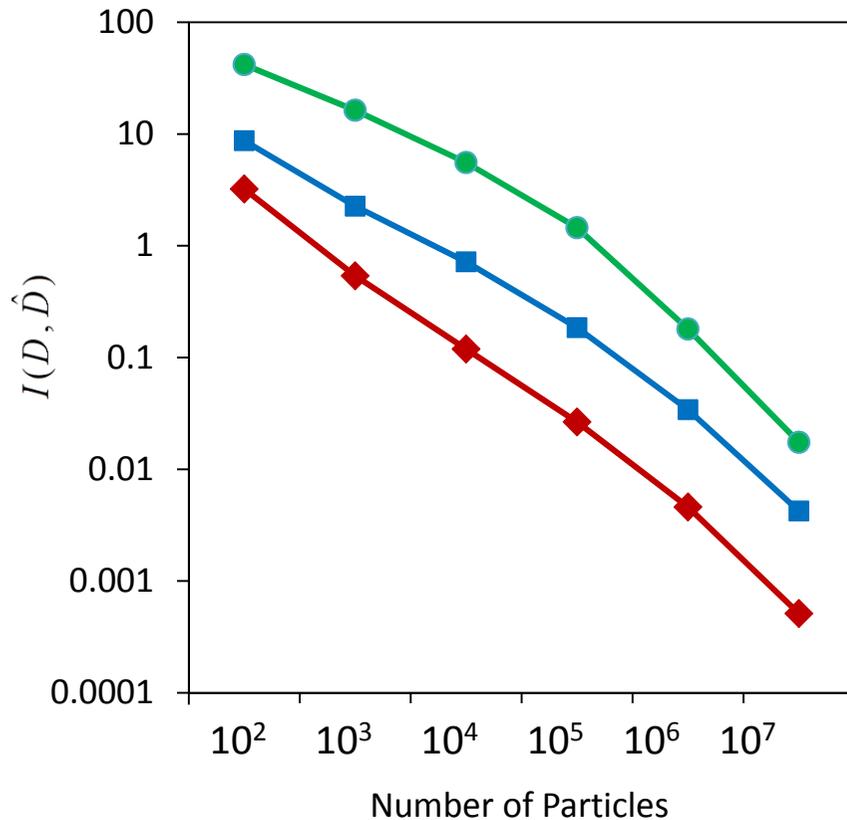
すべての履歴でなく、最近のL個だけをリサンプルすると固定ラグ平滑化が実現できる。

粒子数と精度

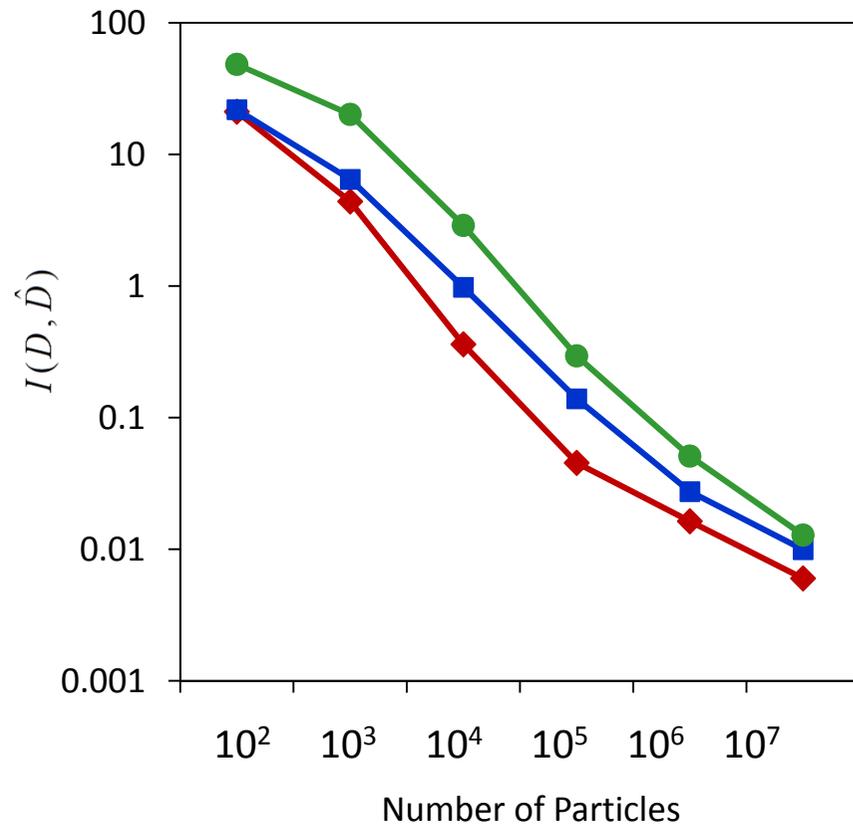


Accuracy vs. Number of Particles

Gaussian model



Cauchy model



—●— Smoothing (max lag)
 —■— Smoothing (best lag)
 —◆— Filter

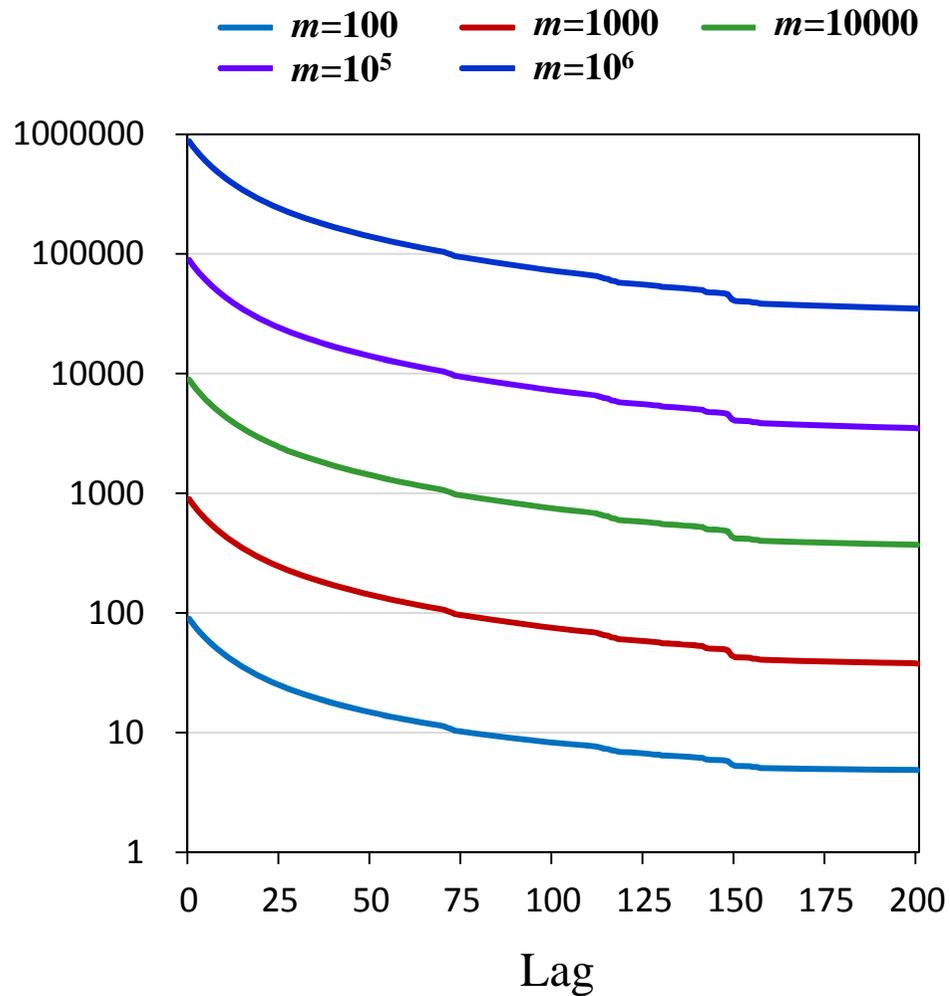
$$I(D, \hat{D}) = \sum_{n=1}^N \sum_{j=1}^L \left\{ D(x_j, n) - \hat{D}(x_j, n) \right\}^2 \Delta x$$

Accuracy in Computing Log-Likelihood

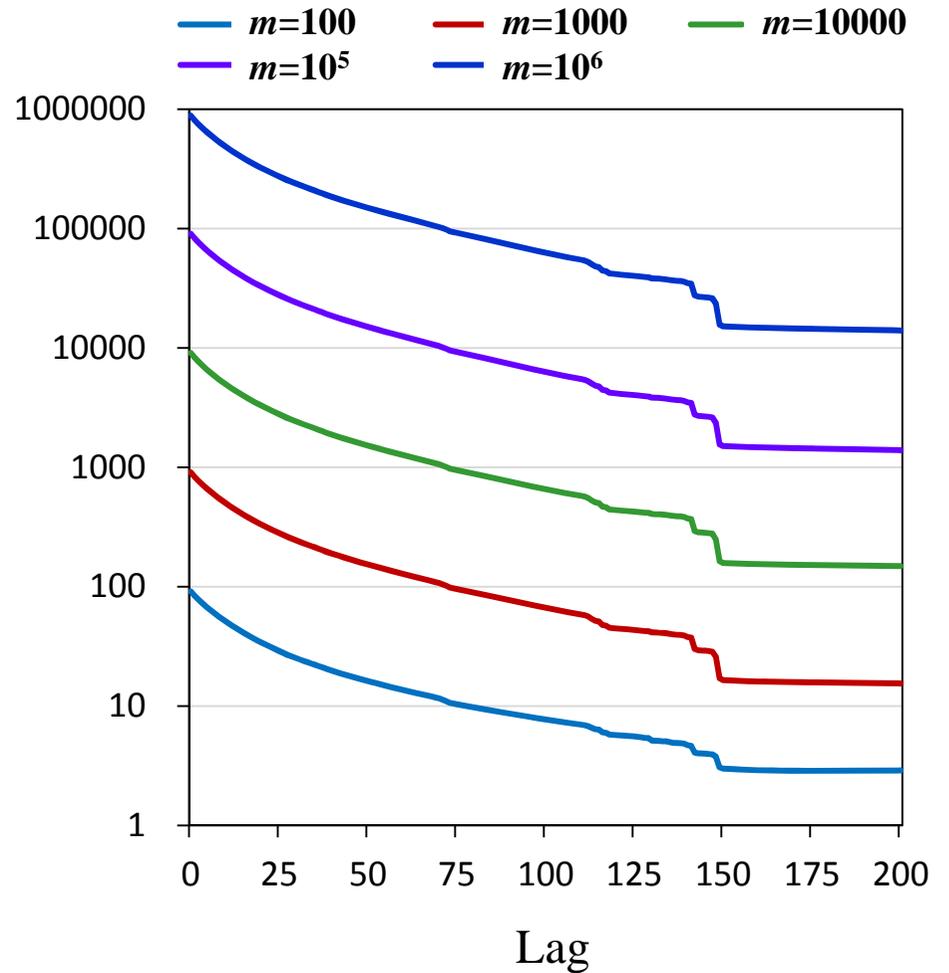
	Gaussian model			Cauchy model		
m	Log-L	S.D.	CPU-time	Log-L	S.D.	CPU-time
10^2	-750.859	2.287	0.02	-752.207	6.247	0.02
10^3	-748.529	1.115	0.06	-743.244	2.055	0.06
10^4	-748.127	0.577	0.58	-742.086	0.429	0.63
10^5	-747.960	0.232	5.84	-742.024	0.124	6.27
10^6	-747.931	0.059	59.41	-742.029	0.038	62.73
10^7	-747.926	0.023	591.04	-742.026	0.013	680.33
10^8	-747.930	0.008	5906.62	-742.026	0.003	6801.55
10^9	-747.928	0.002	59077.35	-742.026	0.001	69255.03

Number of Different Particles in Fixed-Lag Smoothing

Gaussian model

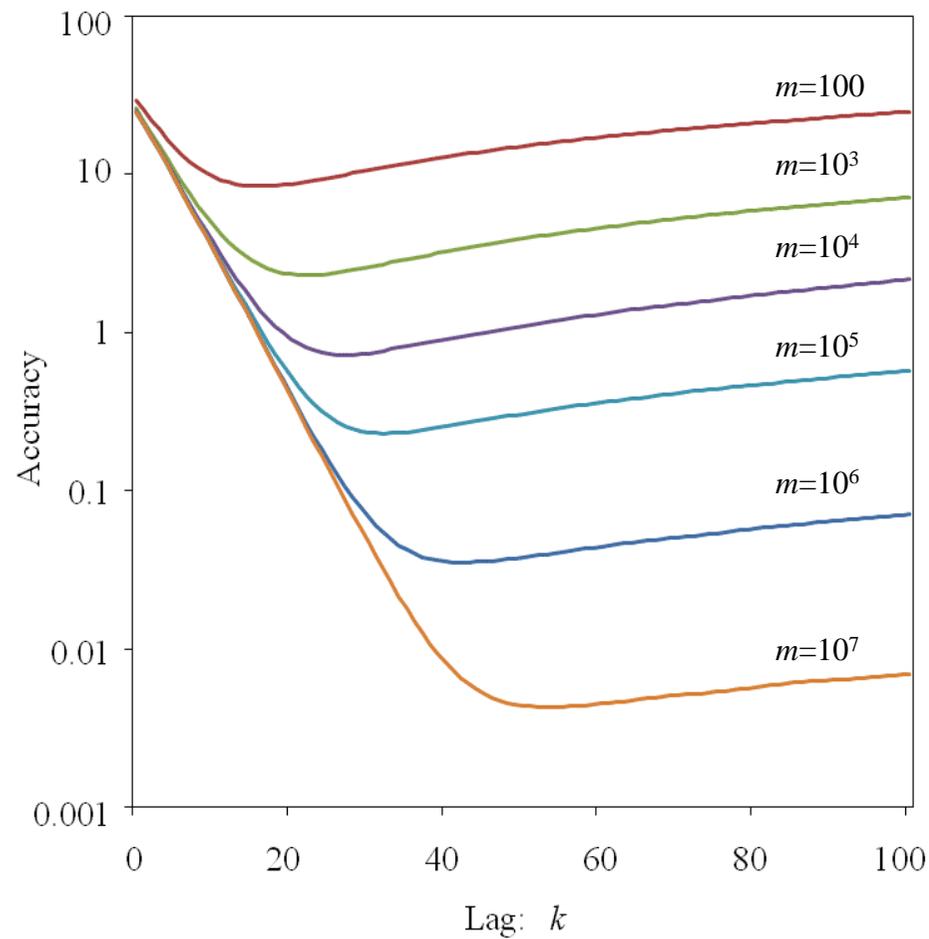


Cauchy model

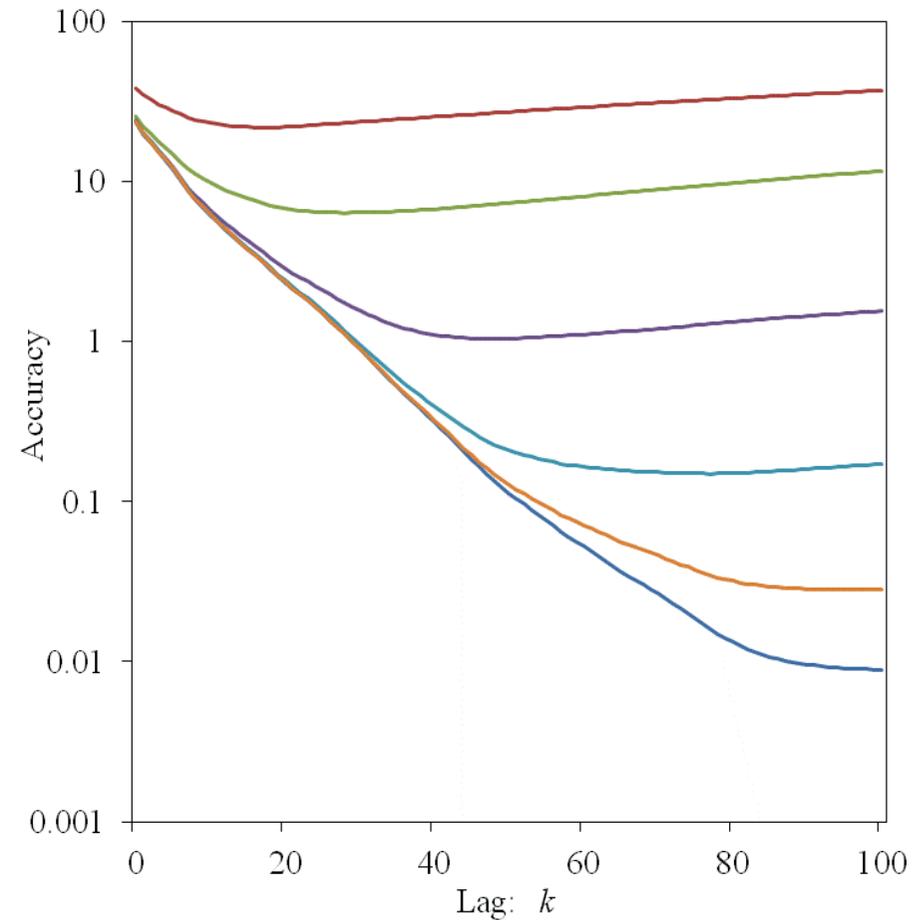


Accuracy of Fixed-Lag Smoother

Gaussian model



Cauchy model



Smoothing by Two Filter Formula

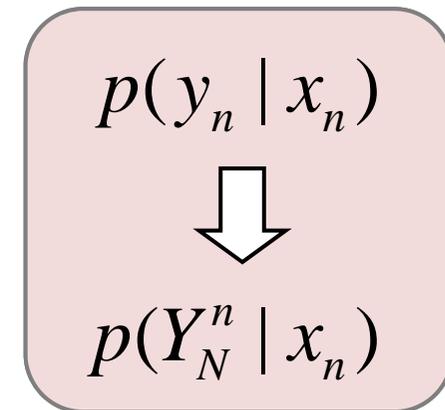
Filter

$$\begin{aligned} p(x_n | Y_n) &= p(x_n | Y_{n-1}, y_n) \\ &\propto p(x_n, y_n | Y_{n-1}) \\ &= \underline{p(y_n | x_n)} p(x_n | Y_{n-1}) \end{aligned}$$

$$\begin{aligned} Y_n &= \{y_1, \dots, y_n\} \\ Y_N^n &= \{y_n, \dots, y_N\} \end{aligned}$$

Smoother

$$\begin{aligned} p(x_n | Y_N) &= p(x_n | Y_{n-1}, Y_N^n) \\ &\propto p(x_n, Y_N^n | Y_{n-1}) \\ &= \underline{p(Y_N^n | x_n)} p(x_n | Y_{n-1}) \end{aligned}$$



Backward Filtering

Initialization

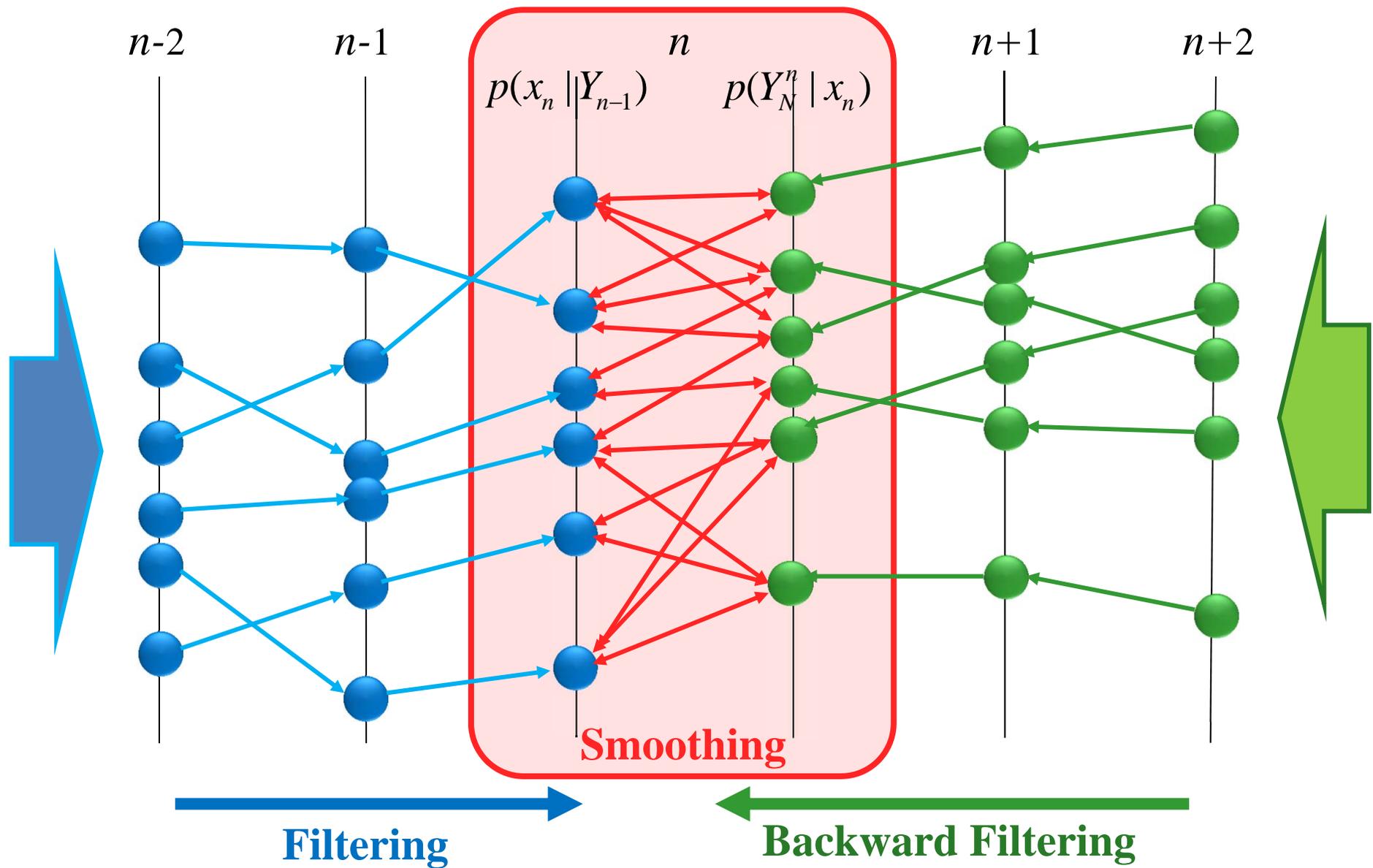
$$p(Y_N^N | x_N) = p(y_N | x_N)$$

Backward Filtering

$$p(Y_N^{n+1} | x_n) = \int p(Y_N^{n+1} | x_{n+1})p(x_{n+1} | x_n)dx_{n+1}$$

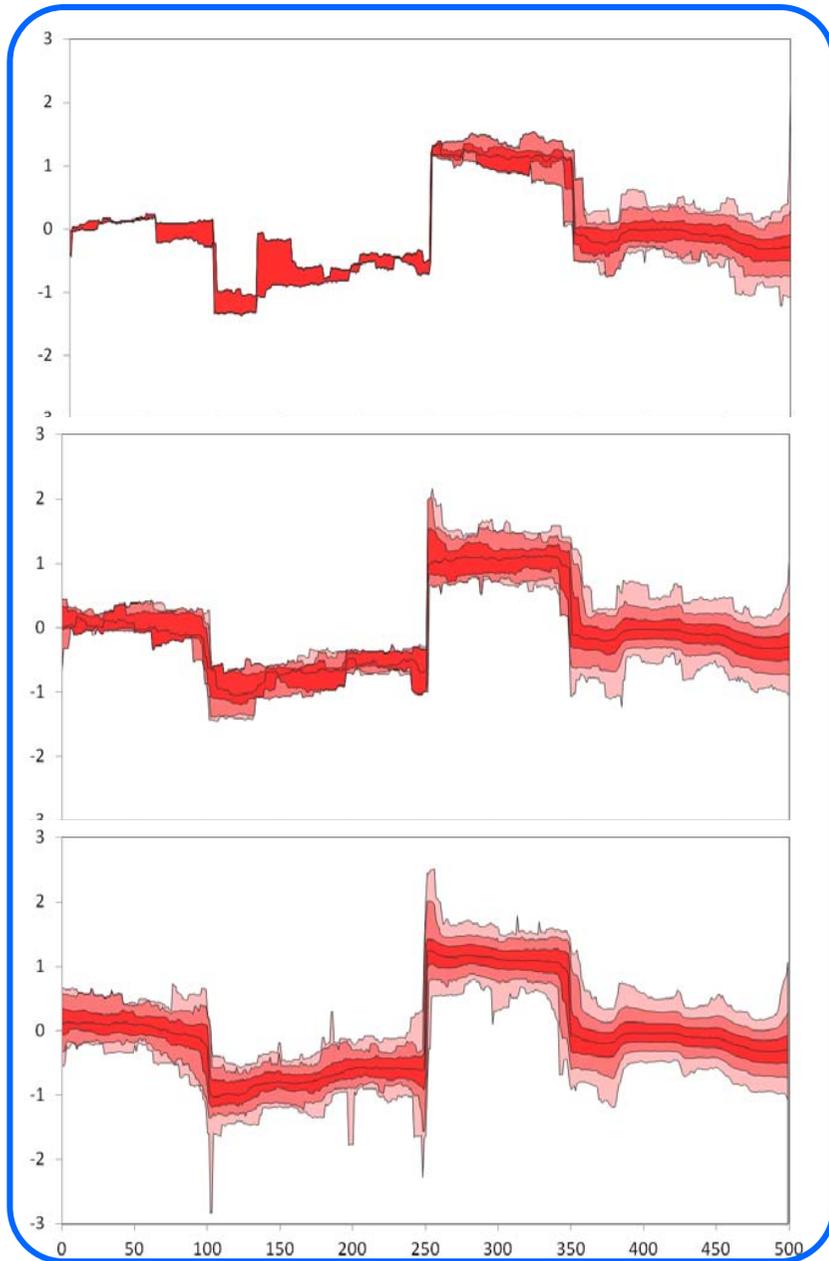
$$p(Y_N^n | x_n) = p(y_n | x_n)p(Y_N^{n+1} | x_n)$$

Two-Filter Formula for Smoothing



Fixed-Lag with LAG=500

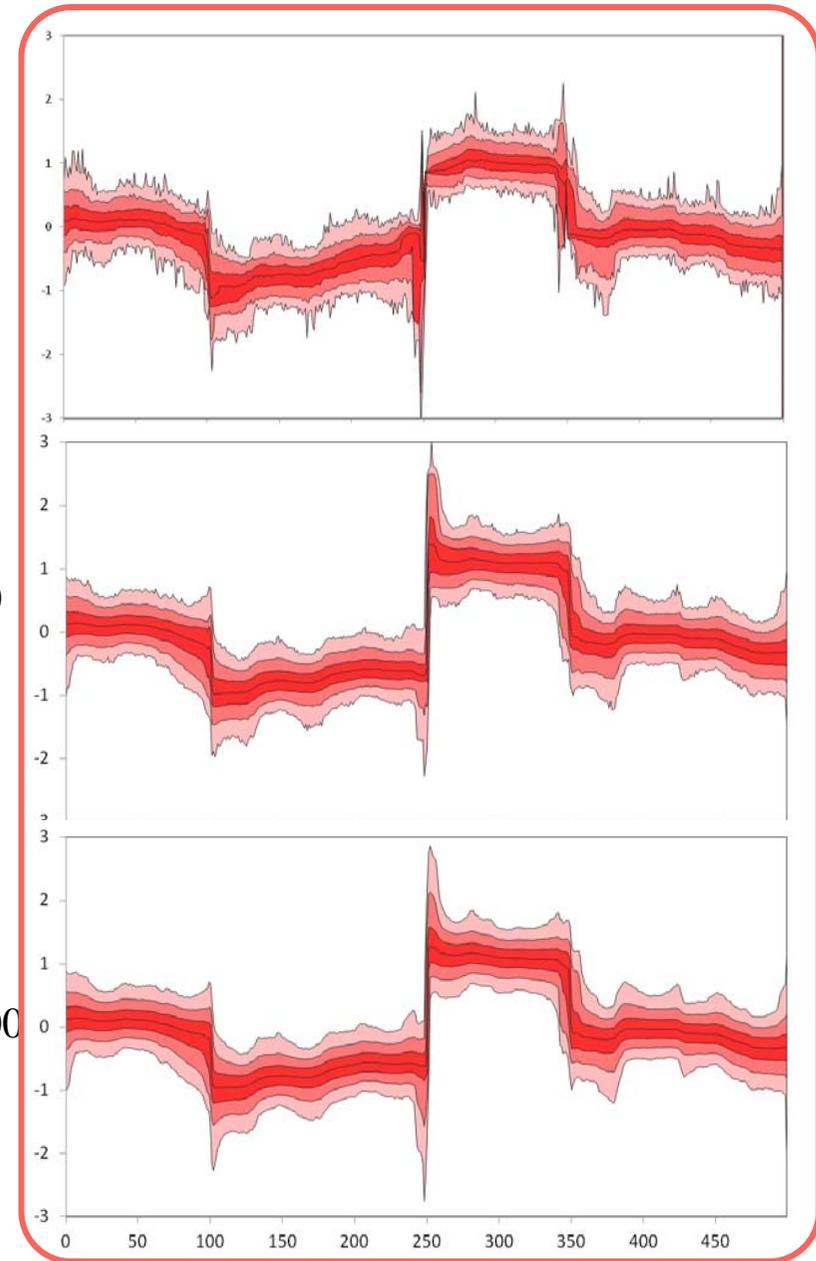
Two-Filter Formula



$m=1,000$

$m=10,000$

$m=100,000$

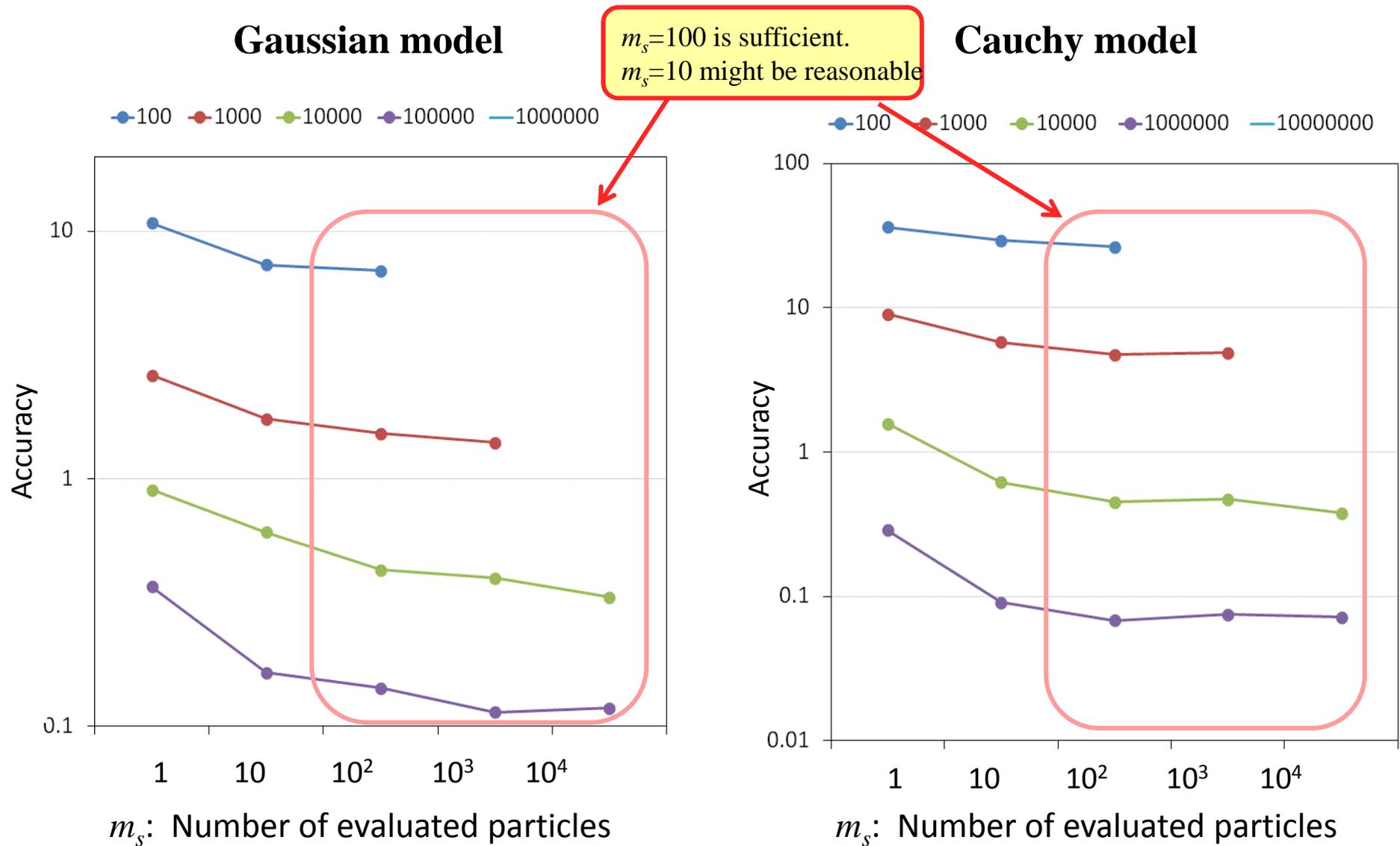


Accuracy of Smoothing

m	Gaussian Model			Cauchy Model		
	Fixed-lag	Fixed-interval	Two-filter	Fixed-lag	Fixed-interval	Two-filter
10^2	8.693	41.723	6.913	21.248	47.881	26.440
10^3	2.259	16.275	1.399	6.042	23.654	4.870
10^4	0.717	5.547	0.333	1.001	3.679	0.378
10^5	0.185	1.448	0.118	0.140	0.380	0.072

Evaluated by $I(D, \hat{D})$

Accuracy of Two-Filter Smoothing Algorithm



粒子フィルタの応用

1. 非ガウス型平滑化

レベルシフト (構造変化)
非ガウス型季節調整
確率的ボラティリティモデル

2. 非線形平滑化

トラッキング
Phase-unwrapping

3. 信号抽出問題

4. 計数データのモデリング

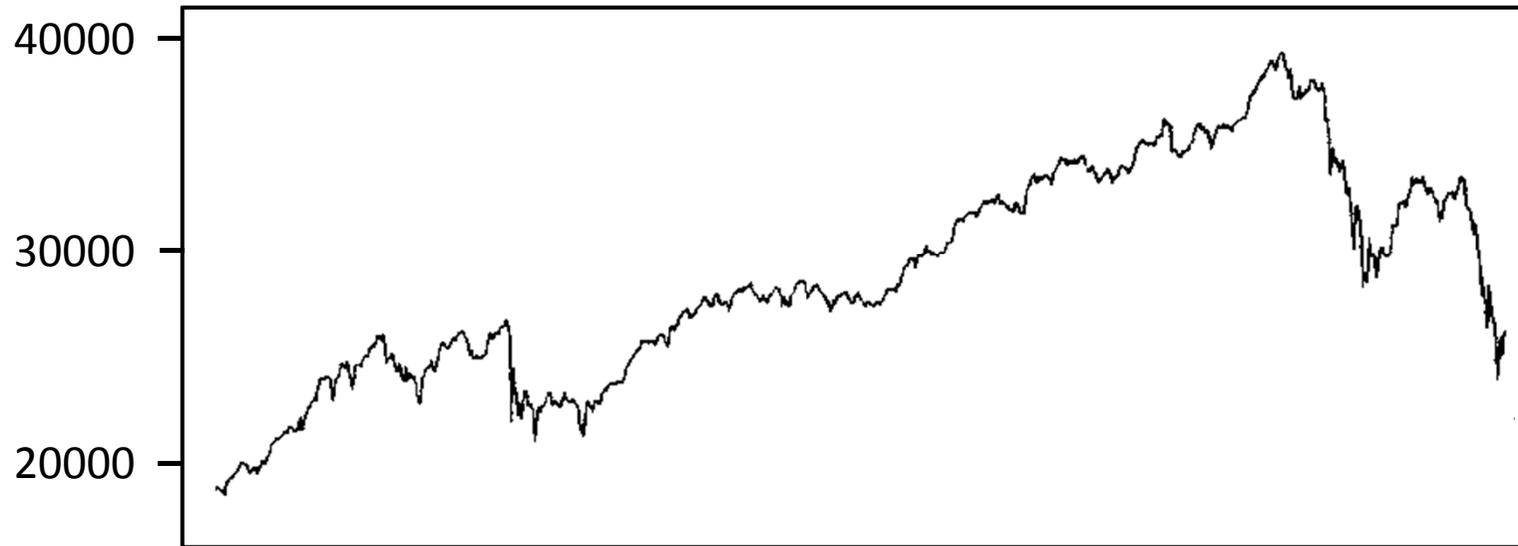
5. 自己組織型状態空間モデル

6. 高次元フィルタ (データ同化)

Gordon et al. (1993), Kitagawa (1996)

Doucet, de Freitas and Gordon (2001) “Sequential Monte Carlo Methods in Practice”

日経225平均株価



- 平均値が変動
- 分散が変動
- 分散変動と平均値の変化に相関の可能性がある

トレンドとボラティリティのモデル化

観測モデル

$$y_n = t_n + \sigma_n w_n, \quad w_n \sim N(0,1)$$

トレンド成分モデル

$$t_n = 2t_{n-1} - t_{n-2} + v_n$$

分散変動成分モデル

$$\log \sigma_n^2 = 2 \log \sigma_{n-1}^2 - \log \sigma_{n-2}^2 + u_n$$

トレンドと分散は独立

状態空間モデル

状態ベクトル

$$x_n = \begin{bmatrix} t_n \\ t_{n-1} \\ \log \sigma_n^2 \\ \log \sigma_{n-1}^2 \end{bmatrix}$$

状態空間モデル

$$x_n = Fx_{n-1} + Gv_n$$

$$y_n = H(x_n, w_n)$$

$$F = \begin{bmatrix} 2 & -1 & & \\ 1 & 0 & & \\ & & 2 & -1 \\ & & 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$H(x_n, w_n) = t_n + \sigma_n w_n$$

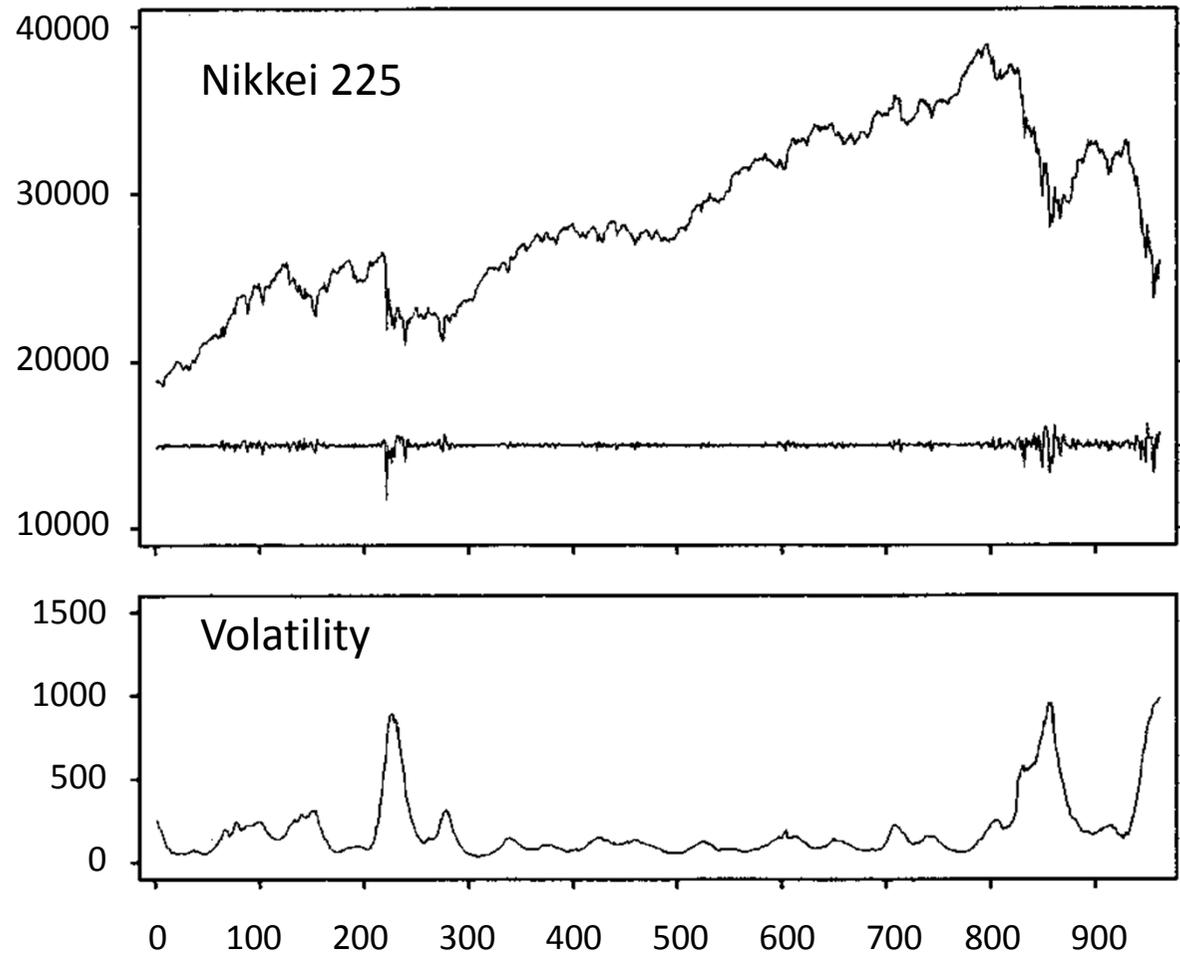
$$v_n = \begin{bmatrix} \varepsilon_n \\ \delta_n \end{bmatrix}$$

確率的ボラティリティ

Model(G,G) $\varepsilon_n \sim N(0, \tau_1^2)$ $\delta_n \sim N(0, \tau_2^2)$

Model(C,G) $\varepsilon_n \sim C(0, \tau_1^2)$ $\delta_n \sim N(0, \tau_2^2)$

Model	(G,G)	(C,G)
τ_1^2	9000	700
τ_2^2	0.0026	0.0005
AIC	13726.02	13765.84



確率的ボラティリティ (自己組織化)

オリジナルの状態空間モデル

$$x_n = \beta x_{n-1} + v_n \quad v_n \sim N(0, \tau^2)$$

$$r_n = e^{\alpha/2} e^{x_n} w_n \quad \text{未知パラメータ} \rightarrow \alpha, \beta, \tau^2$$

同時推定のための状態空間モデル

$$\begin{bmatrix} x_n \\ \alpha_n \\ \beta_n \\ \log \tau_n^2 \end{bmatrix} = \begin{bmatrix} \beta x_{n-1} \\ \alpha_{n-1} \\ \beta_{n-1} \\ \log \tau_{n-1}^2 \end{bmatrix} + \begin{bmatrix} e^{\lambda_n/2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_n \\ \varepsilon_{n1} \\ \varepsilon_{n2} \\ \varepsilon_{n3} \end{bmatrix}$$

$$\lambda_n = \log \tau_n^2$$

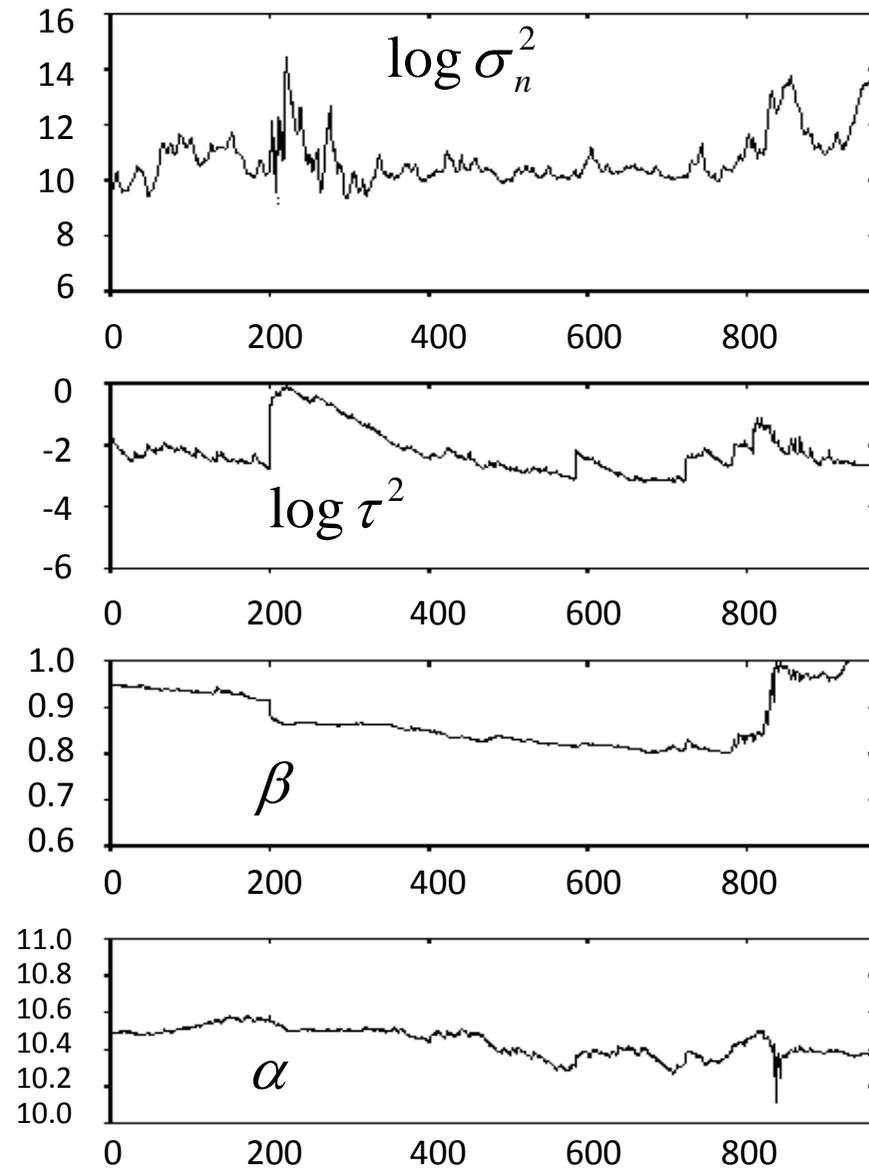
$$r_n \sim e^{x_n} w_n$$

確率的ボラティリティ (自己組織化)

$$\theta_n = (\alpha_n, \beta_n, \log \tau_n^2)^T$$

$$\begin{bmatrix} x_n \\ \theta_n \end{bmatrix} = \begin{bmatrix} \beta x_{n-1} \\ \theta_{n-1} \end{bmatrix} + \begin{bmatrix} e^{\theta_n/2} & & \\ & I_3 & \\ & & \end{bmatrix} \begin{bmatrix} v_n \\ \varepsilon_n \end{bmatrix}$$

$$r_n = e^{x_n} w_n$$

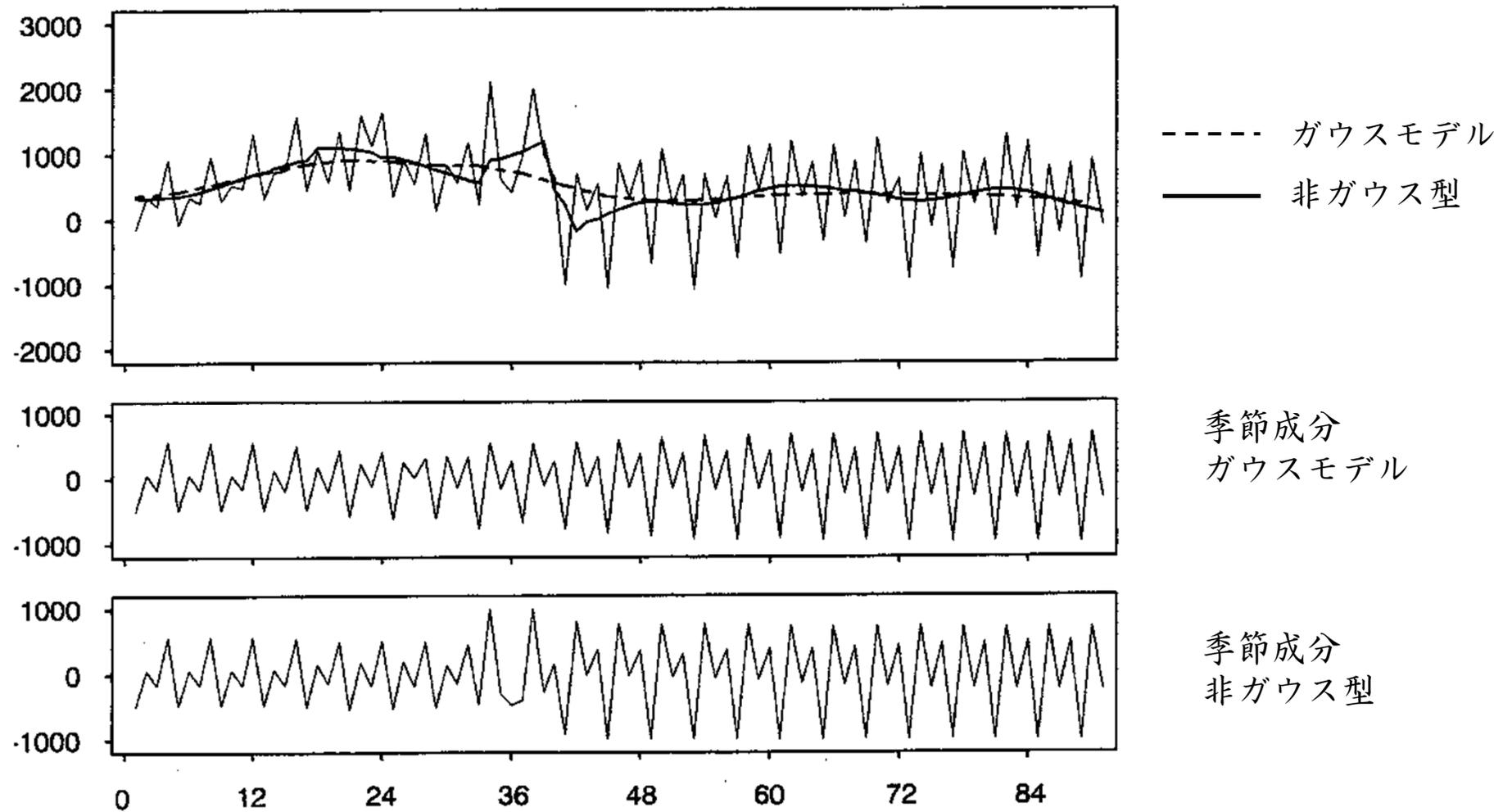


季節調整：構造変化が存在する場合

$$\begin{aligned}
 y_n &= T_n + S_n + w_n & w_n &\sim N(0, \sigma^2) \\
 T_n &= 2T_{n-1} - T_{n-2} + v_n & v_n &\sim \alpha N(0, \tau_1^2) + (1-\alpha)N(0, \zeta_1^2) \\
 S_n &= -(S_{n-1} + \dots + S_{n-p+1}) + u_n & u_n &\sim \beta N(0, \tau_2^2) + (1-\beta)N(0, \zeta_2^2)
 \end{aligned}$$

	$\hat{\sigma}^2$	$\hat{\tau}_1^2$	$\hat{\tau}_2^2$	$\hat{\alpha} = \hat{\beta}$	Log-LK	AIC
Gaussian	0.874×10^5	0.146×10^6	0.365×10^4	0.98	-670.39	1346.8
Non-Gaussian	0.288×10^5	0.528×10^3	0.362×10^3	0.98	-667.01	1342.0

季節調整：異常値

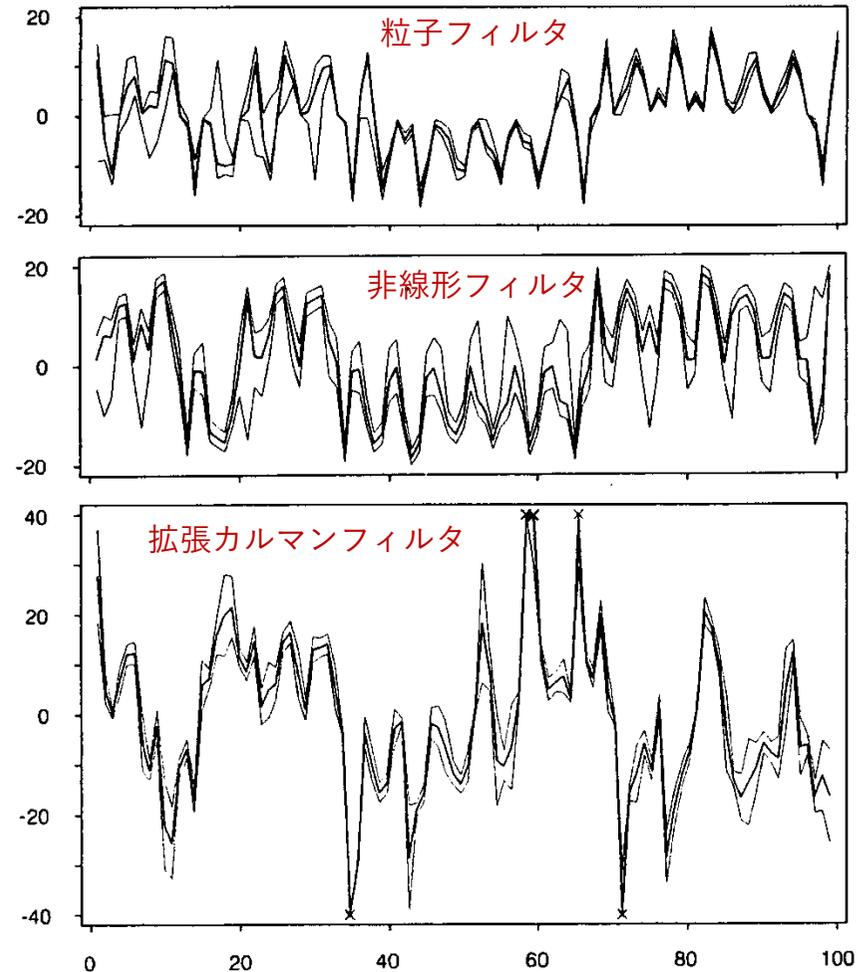
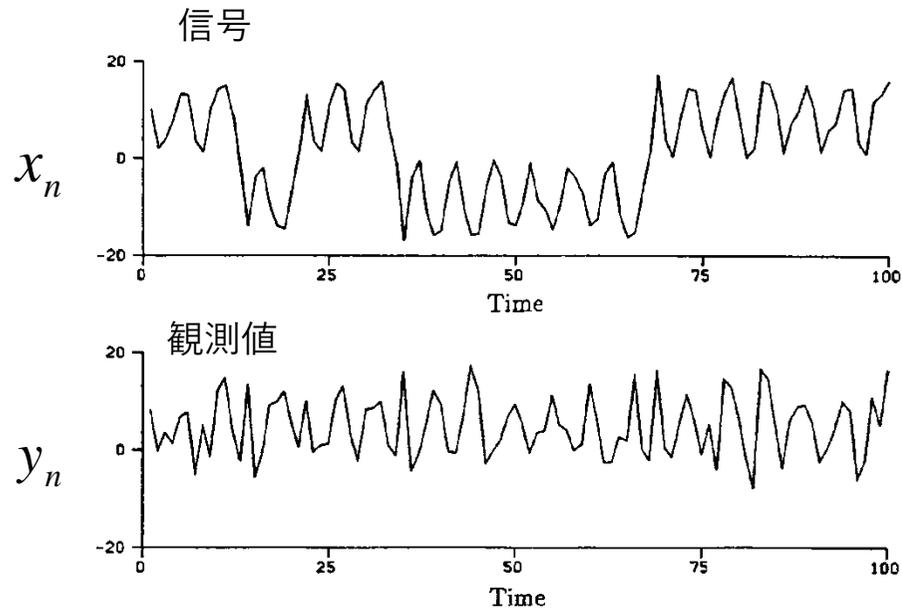


非線形平滑化

$$x_n = \frac{1}{2}x_{n-1} + \frac{25x_{n-1}}{1+x_{n-1}^2} + 8\cos(1.2n) + v_n$$

$$y_n = \frac{x_n^2}{20} + w_n$$

$$v_n \sim N(0,0.1), w_n \sim N(0,1)$$



非線形モデル（自己組織化）

$$x_n = \theta_{n3}x_{n-1} + \frac{\theta_{n4}x_{n-1}}{1+x_{n-1}^2} + \theta_{n5} \cos(1.2n) + v_n$$

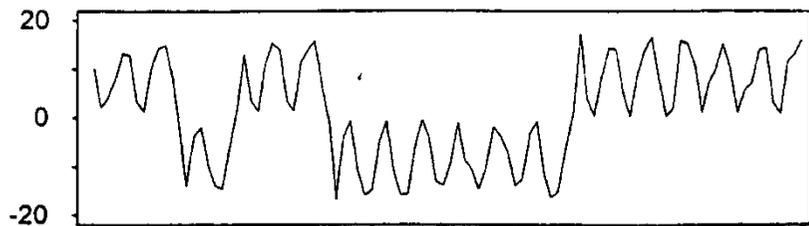
$$y_n = \theta_{n6}x_n^2 + w_n$$

$$v_n \sim N(0, \exp\{\theta_{n1}\}), \quad w_n \sim N(0, \exp\{\theta_{n2}\})$$

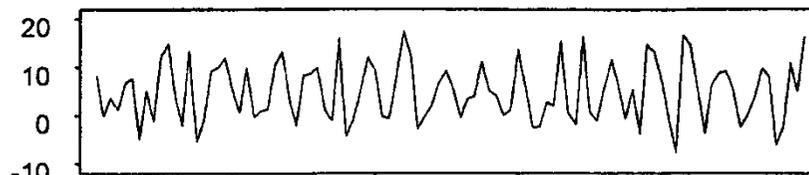
Augmented State Vector

$$z_n = \begin{bmatrix} x_n \\ \theta_{n1} \\ \vdots \\ \theta_{n6} \end{bmatrix}$$

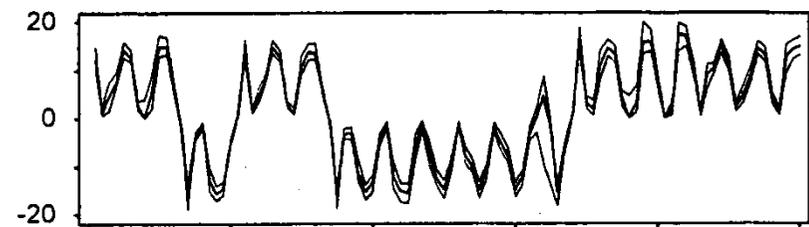
自己組織型平滑化



信号



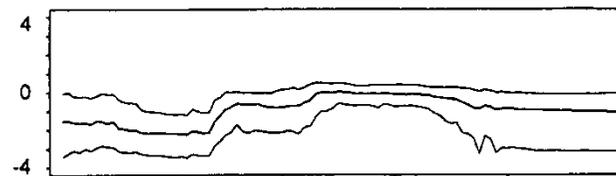
観測値



信号の推定

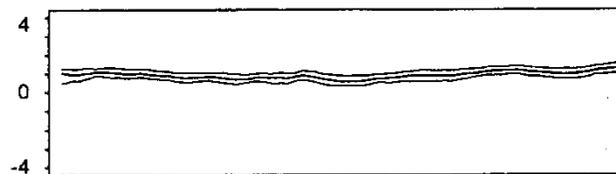
$$x_n = \theta_{n3} x_{n-1} + \frac{\theta_{n4} x_{n-1}}{1 + x_{n-1}^2} + \theta_{n5} \cos(1.2n) + v_n$$

$$y_n = \theta_{n6} x_n^2 + w_n \quad v_n \sim N(0, \exp\{\theta_{n1}\}), w_n \sim N(0, \exp\{\theta_{n2}\})$$



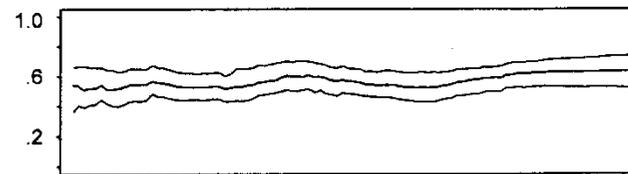
(d)

θ_{n1}



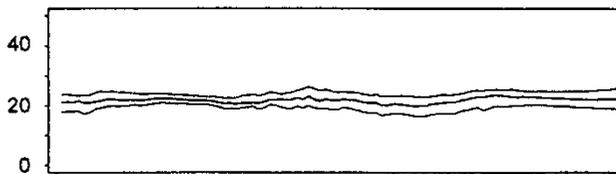
(e)

θ_{n2}



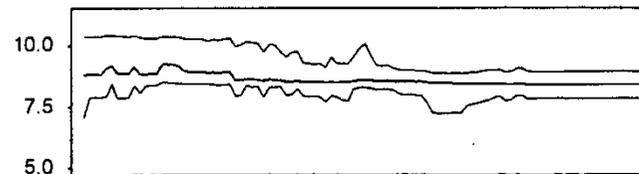
(f)

θ_{n3}



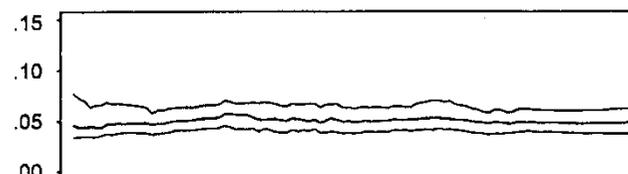
(g)

θ_{n4}



(h)

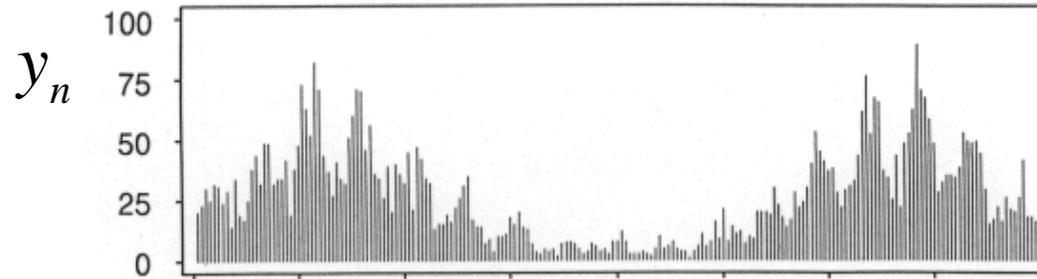
θ_{n5}



θ_{n6}

0 20 40 60 80 100

計数データの季節調整



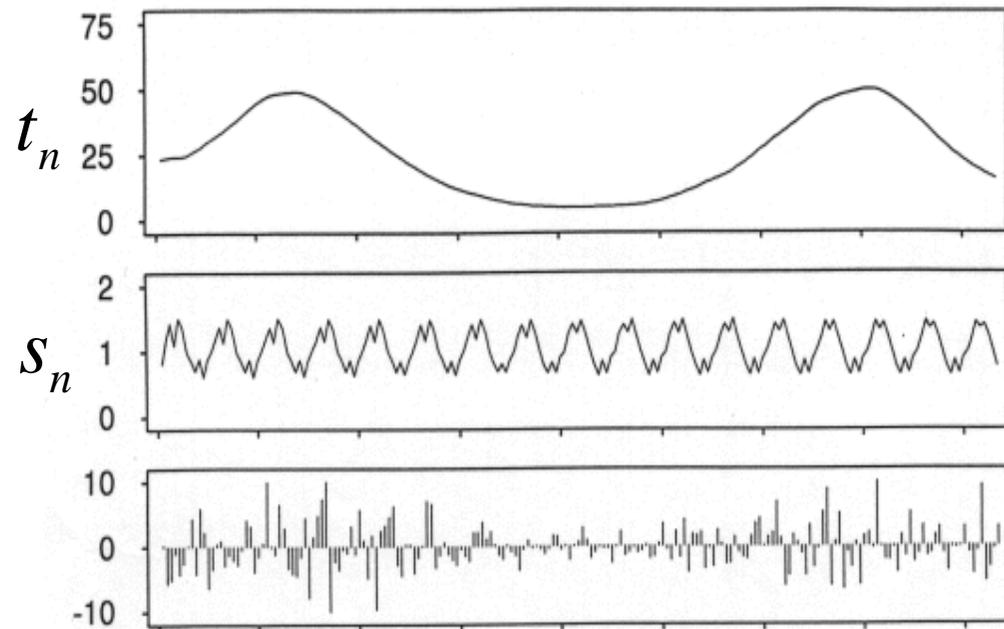
$$\log t_n = 2\log t_{n-1} - \log t_{n-2} + v_n$$

$$\log s_n = -(\log s_{n-1} + \dots + \log s_{n-11}) + u_n$$

$$\lambda_n = t_n \times s_n$$

$$P(y_n | \lambda_n) = \frac{e^{-\lambda_n} \lambda_n^{y_n}}{y_n!}$$

ポアソン分布



データ同化: 非線形状態空間モデル

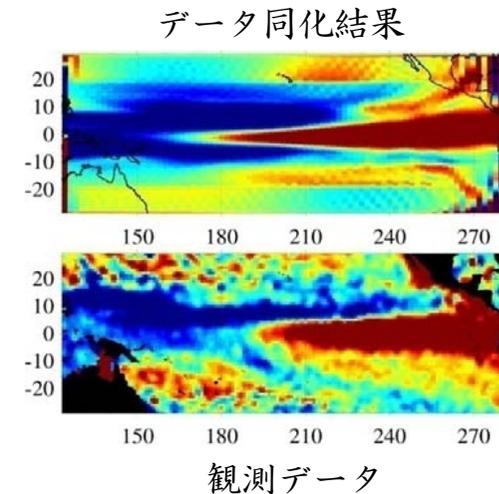


データ同化 = 状態 (x_t) 推定問題

$$x_t = F_t(x_{t-1}, v_t) \quad (\text{システムモデル})$$

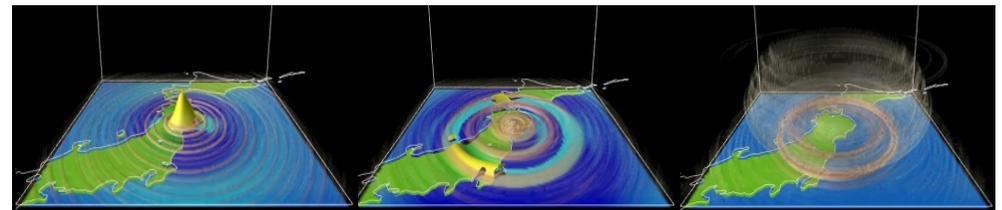
$$y_t = H_t x_t + w_t \quad (\text{観測モデル})$$

- x_t : シミュレーションモデルの全変数
- y_t : 全観測変数
- v_t : システムモデルの不確実性
- w_t : 観測モデル



大地震発生時の微気圧変動のデータ同化結果 (統数研)

超高次元 x_t : $10^4 \sim 10^6$
 y_t : $10^2 \sim 10^5$



```

# Generation of data
## trend model
x <- rep(0,400)
x[101:200] <- 1
x[201:300] <- -1
y <- x + rnorm(400, mean=0, sd=0.5)

#####
# Particle Filter
#####
m <- 10000
tau <- 0.04
sig <- 0.25

# Initial distribution
xf <- rnorm(m,mean=0,sd=2)
n <- length(y)
trend <- rep(0,length=n)

for (ii in 1:n) {
# Prediction step
v <- rnorm(m,mean=0,sd=tau)
xp <- xf + v

# Bayes factor (weights of particles)
alpha <- dnorm(y[ii]-xp,mean=0,sd=sig,log=FALSE)
alpha <- alpha/sum(alpha)

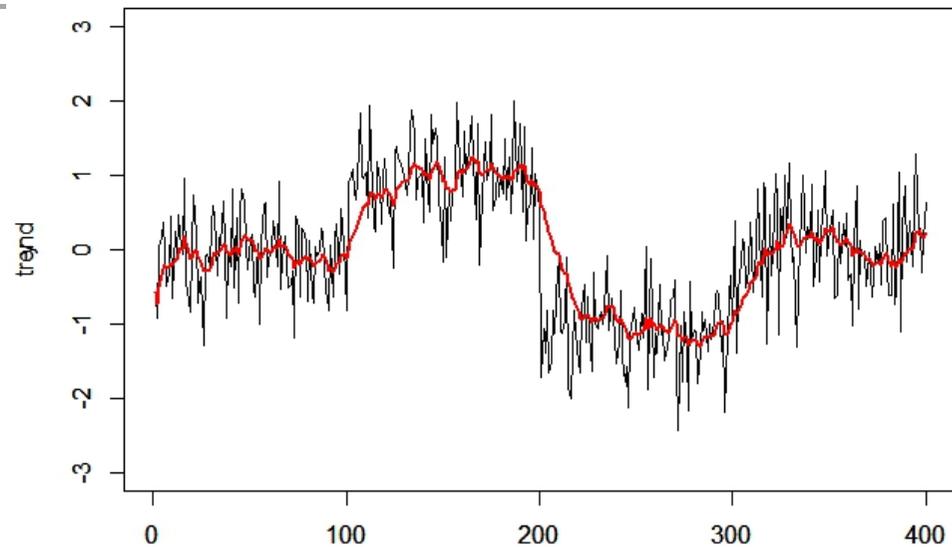
# re-sampling
xf <- sample(xp,prob=alpha,replace=T)

qxf <- quantile(xf)
trend[ii] <- mean(qxf)
}

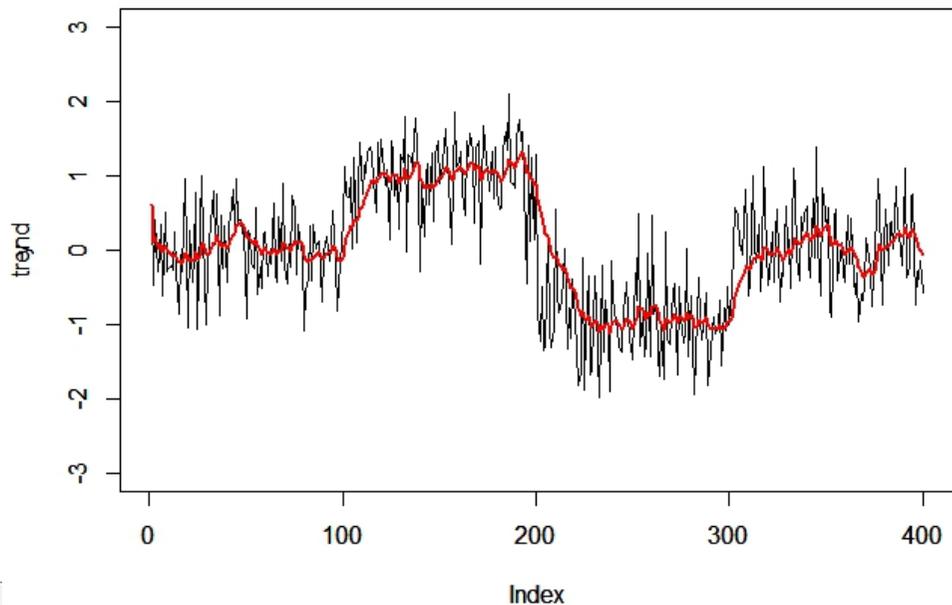
plot(y,type="l",ylim=c(-3,3))
par(new=T)
plot(trend,type="l",col="red",lwd=2,ylim=c(-3,3))

```

m=1,000

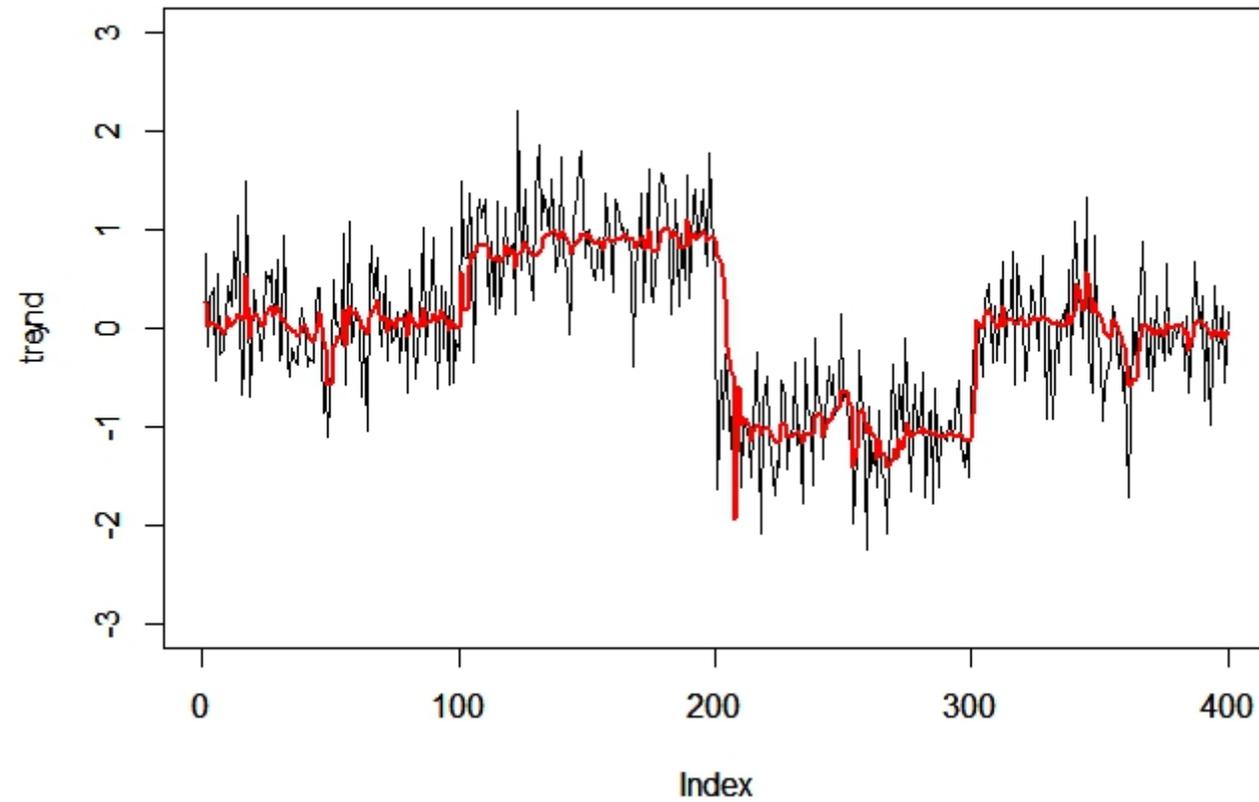


m=10,000



V: Cauchy distribution

`v <- rcauchy(m,location=0,scale=0.0001)`



並列計算

- MCFアルゴリズムの直接並列化
- 単純並列 MCF
 - 単純に k 個の MCFを並列に実行
 - 事後分布を平均
- 重み付き並列 MCF
 - 事後分布の（モデルの尤度に基づく）重み付き平均
 - 交叉を伴う重み付き並列MCF
- 移住を伴う重み付き並列 MCF
 - “良い” MCFから “悪い” MCFへの粒子の移住