Online adaptation of the number of particles in sequential Monte Carlo methods

Víctor Elvira⋆

joint work with Joaquín Míguez† and Petar M. Djuric‡

⋆ IMT Lille Douai (France)

† University Carlos III of Madrid (Spain)

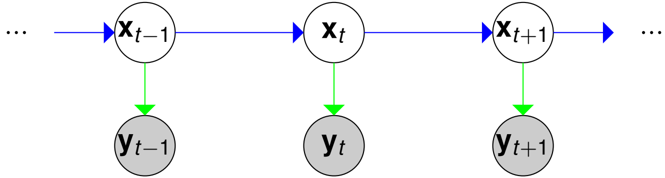
‡ Stony Brook University (USA)

August 30, 2017, Uppsala.

Table of contents

Inference in State-Space Models

- Let us consider:
  - a set of hidden states $\mathbf{x}_t \in \mathbb{R}^{d_x}$, $t = 1, ..., T$.
  - a set of observations $\mathbf{y}_t \in \mathbb{R}^{d_y}$, $t = 1, ..., T$.



$$
\begin{aligned}
\mathbf{x}_t &= g(\mathbf{x}_{t-1}, \mathbf{u}_t) \rightarrow \quad p(\mathbf{x}_t|\mathbf{x}_{t-1}) \\
\mathbf{y}_t &= h(\mathbf{x}_t, \mathbf{v}_t) \quad\; \rightarrow \quad p(\mathbf{y}_t|\mathbf{x}_t)
\end{aligned}
$$

where g and h are known and $\mathbf{u}_t$ and $\mathbf{v}_t$ have known distribution

Optimal Filtering

- Filtering Problem:
  - **Filtered** distribution of $\mathbf{x}_t$ given all the obs. $p(\mathbf{x}_t|\mathbf{y}_{1:t})$
  - **Recursively** from $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ updating with the new $\mathbf{y}_t$
- Optimal filtering:
  1. **Prediction** step:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}$$

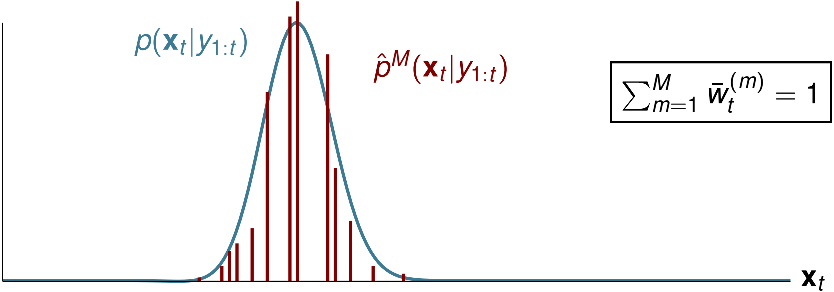
  2. **Update** step:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}$$

- Usually the posterior cannot be analytically computed!
- Interest in integrals of the form: $(f, p_t) = \int f(\mathbf{x}_t)p(\mathbf{x}_t|y_{1:t})d\mathbf{x}_t$

Particle Filtering (Sequential Monte Carlo)

- The distributions are approximated by a random measure of M particles and associated normalized weights
  $\mathcal{X} = \{\mathbf{x}_t^{(m)}, \bar{w}_t^{(m)}\}_{m=1}^{M}$
  - $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \hat{p}^M(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{m=1}^{M} \bar{w}_t^{(m)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(m)})$



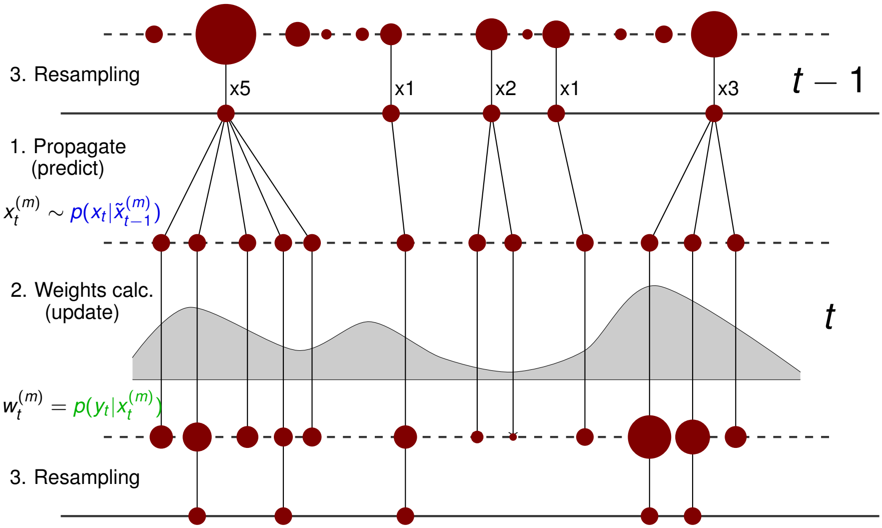
$\boxed{\sum_{m=1}^{M} \bar{w}_t^{(m)} = 1}$

A Basic Particle Filter in a Nutshell

- Bootstrap PF $\equiv$ Sequential Importance Resampling (based on importance sampling) [Gordon93]
- The filtered distribution at time t is recursively approximated from $\tilde{\mathcal{X}}_{t-1} = \{\tilde{\mathbf{x}}_{t-1}^{(m)}, \tilde{w}_{t-1}^{(m)}\}_{m=1}^{M}$
- At each time step t and for $m = 1, ..., M$
  1. Propagate (**Prediction**): $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}^{(m)})$
  2. Weights calculation (**Update**): $w_t^{(m)} = \tilde{w}_{t-1}^{(m)} p(y_t|\mathbf{x}_t^{(m)})$

     and normalization: $\bar{w}_t^{(m)} = \frac{w_t^{(m)}}{\sum_{m=1}^{M} w_t^{(m)}}$
  3. **Resampling** (optional but necessary):
     - Sample M times from the random measure $\mathcal{X}_t = \{\mathbf{x}_t^{(m)}, \bar{w}_t^{(m)}\}_{m=1}^{M}$,
     - New random measure $\tilde{\mathcal{X}}_t$ of M particles with equal weights.

Bootstrap Particle Filter



Quality of the Approximation

- The integral of interest is **approximated** $(f, p_t) \approx (f, p_t^M)$ as

$$\int f(\mathbf{x}_t) p(\mathbf{x}_t | y_{1:t}) d\mathbf{x}_t \approx \sum_{m=1}^{M} \bar{w}_t^{(m)} f(\mathbf{x}_t^{(m)})$$

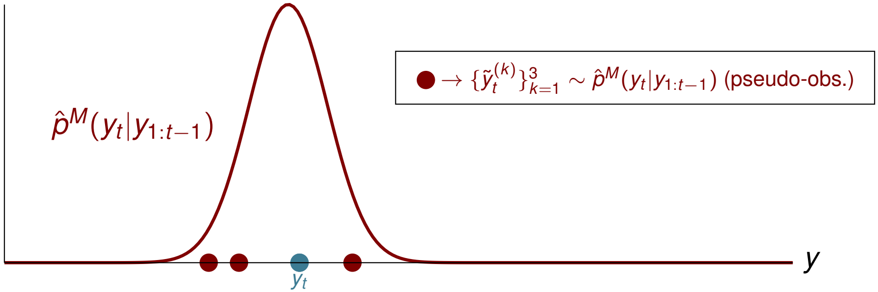
- Some **convergence results** under regularity assumptions:
  - Limit: $\lim_{M\to\infty} |(f, p_t^M) - (f, p_t)| = 0 \qquad$ a.s.
  - Convergence rate: $\mathbb{E}\left[\left((f, p_t^M) - (f, p_t)\right)^2\right] \leq \frac{c_t ||f||_\infty}{M}$
- Peformance/computational cost **tradeoff**
  - Very large M: good approximation but very expensive
  - Reducing M: deteriorates the performance

So, how is M chosen?

Convergence Assessment in Particle Filtering

- **Goal**: in **real time** and for **any SSM**:
  1. **Evaluate the convergence** (quality of the approximation)
  2. **Adapt the number of particles**
- **Intuition**: check whether the received observations "make sense" with the approximated predictive distributions
- **Challenges**:
  - At each time step just one observation y_t available
  - The predictive $\hat{p}^M(y_t|y_{1:t-1})$ is evolving with time
- **Proposed method**: At each time step t
  - Generate K fictitious observations $\tilde{y}_t^{(k)}$ from $\hat{p}^M(y_t|y_{1:t-1})$
    1. $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}^{(m)})$ (prediction step of BPF, **for free**)
    2. $\tilde{y}_t^{(k)} \sim \frac{1}{M}\sum_{m=1}^{M} p(y_t|\mathbf{x}_t^{(m)})$,$k = 1, ..., K$ (**cheap** $K << M$)
  - Compare them with the actual observation y_t.
    - Implicitly, we compare $\hat{p}^M(y_t|y_{1:t-1})$ and $p(y_t|y_{1:t-1})$

Position Matters

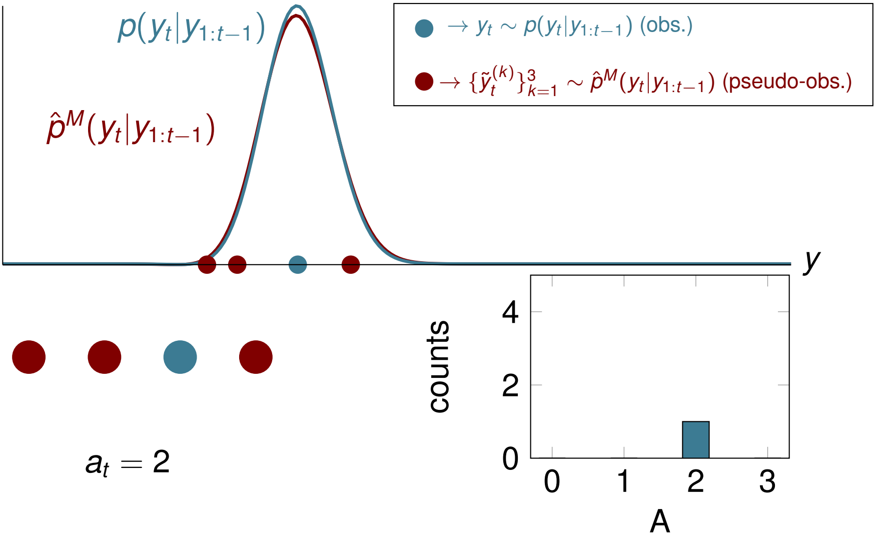


- A_t: number of fictitious observations, $\{\tilde{y}_t^{(k)}\}_{k=1}^3$, smaller than y_t

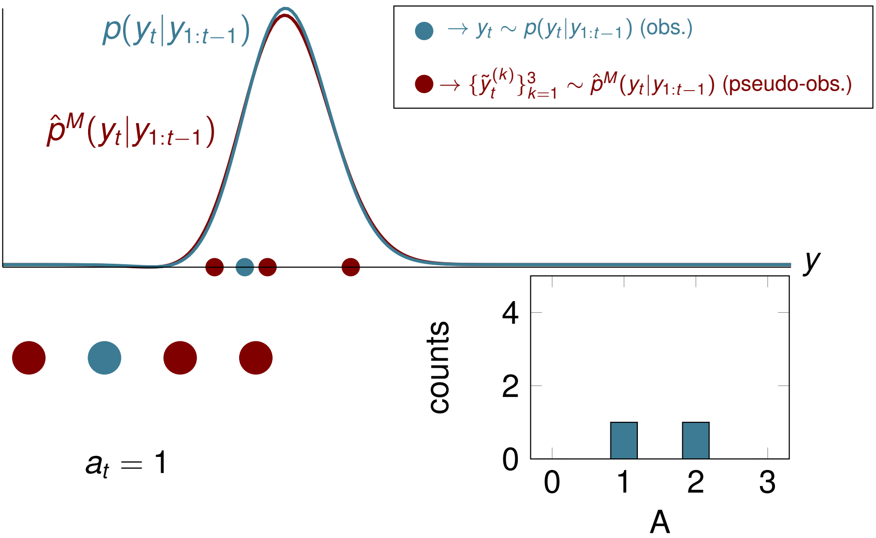
$a_t = 2$

- We can iteratively compute a_t

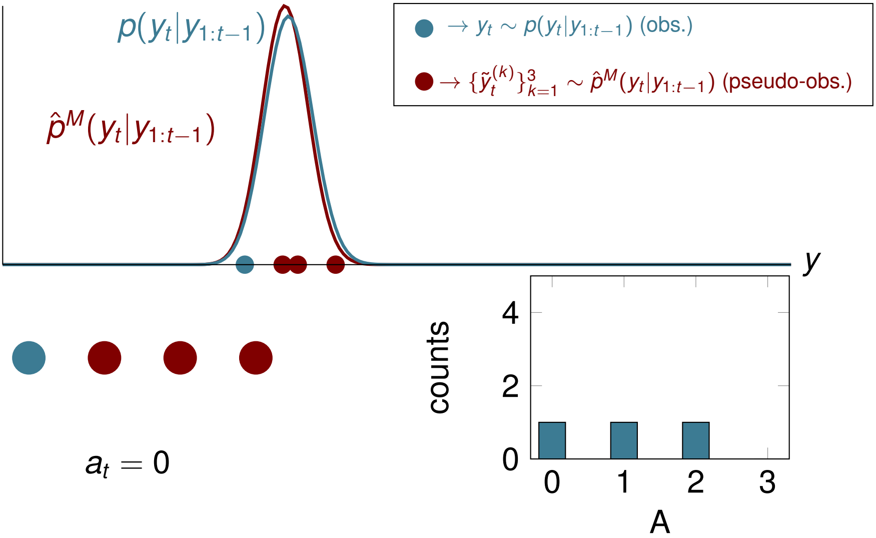
Good Approximation, $t = 1$



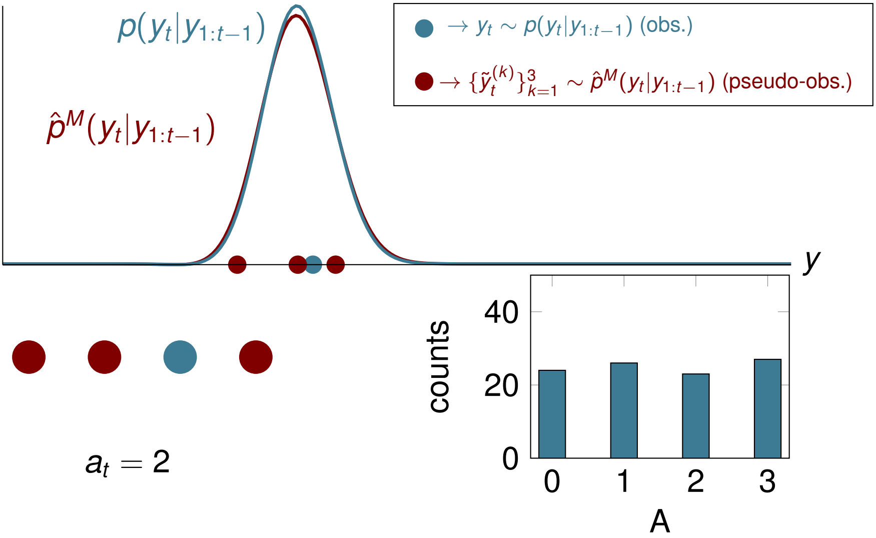
Good Approximation, $t = 2$



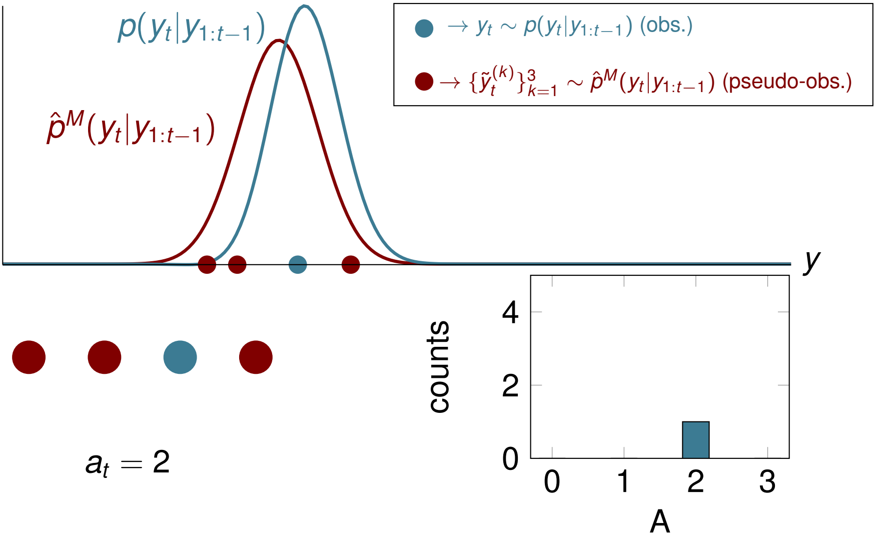
Good Approximation, $t = 3$

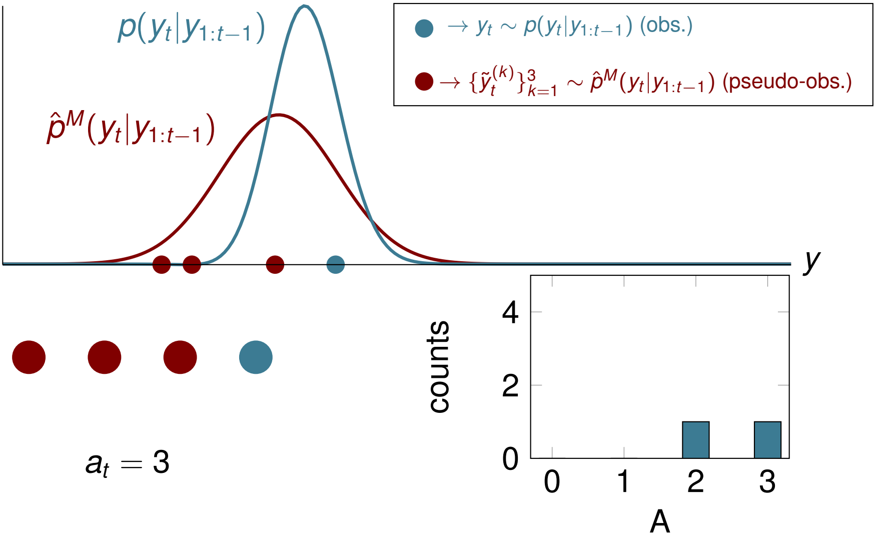


Good Approximation, $t = 100$



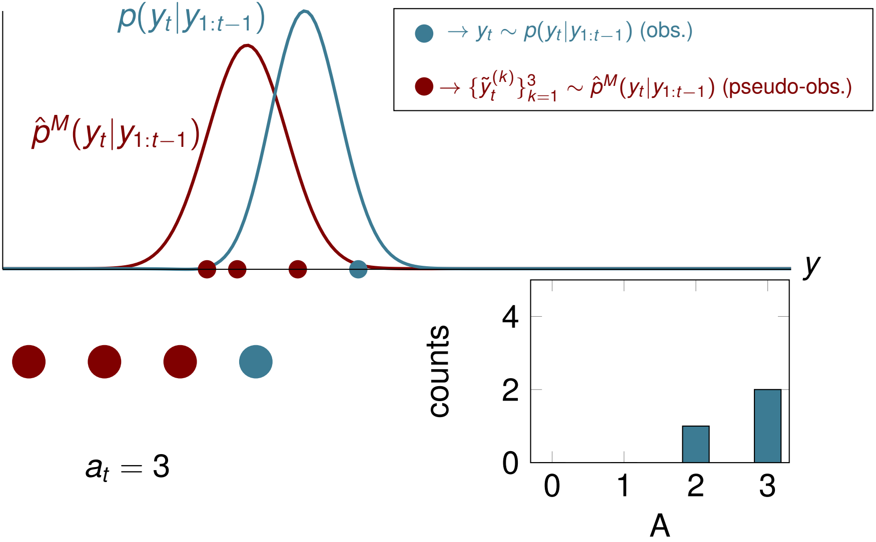
Bad Approximation, $t = 1$



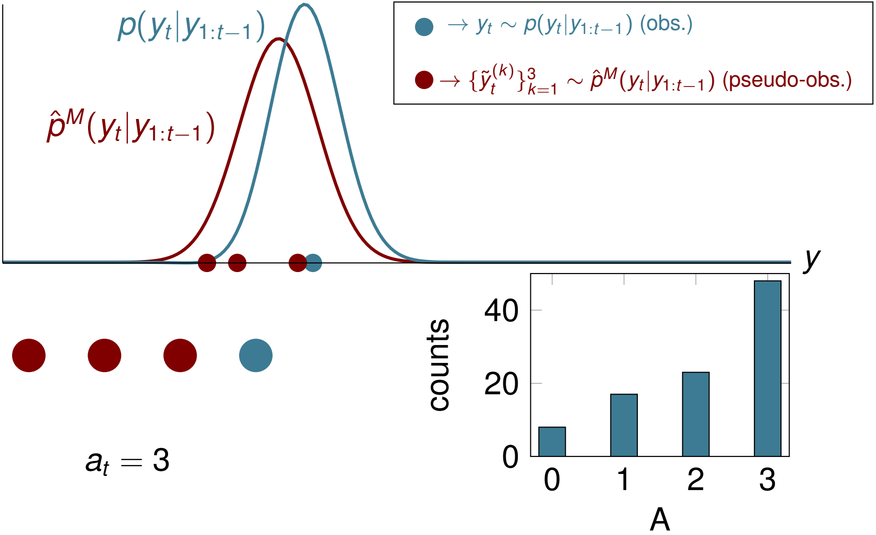


$p(y_t|y_{1:t-1})$
$\hat{p}^M(y_t|y_{1:t-1})$
$\to y_t \sim p(y_t|y_{1:t-1})$ (obs.)
$\to \{\tilde{y}_t^{(k)}\}_{k=1}^3 \sim \hat{p}^M(y_t|y_{1:t-1})$ (pseudo-obs.)
y
$a_t = 3$
counts
4
2
0
0
1
2
3
A

Bad Approximation, $t = 3$



Bad Approximation, $t = 100$



Methodology Summary and Properties

- **Methodology**: At each time step:
  - simulate $\tilde{y}_t^{(k)} \sim \hat{p}^M(y_t|y_{1:t-1}), \qquad k = 1, ..., K$
  - build the r.v. $A_{K,t} := |\mathcal{A}_{K,t}| \in \{0, 1, ..., K\}$, where $\mathcal{A}_{K,t} := \{y \in \{\tilde{y}_t^{(k)}\}_{k=1}^{K} : y < y_t\}$
- **Properties**: Under the hypothesis of **perfect approximation**:
  - $\mathcal{J}_t := \{y_t, \tilde{y}_t^{(1)}, \ldots, \tilde{y}_t^{(K)}\}$ is a set of i.i.d. samples from a **common** continuous probability distribution $p_t(y_t)$, then:

Proposition 1: *the pmf of the r.v. $A_{K,t}$ is **uniform**:*

$$\mathbb{Q}_K(n) = \frac{1}{K+1}, \qquad n = 0, ..., K.$$

Proposition 2: *the r.v.'s A_{K,t_1} and A_{K,t_2} are **independent**, $\forall t_1, t_2 \in \mathbb{N}$ with $t_1 \neq t_2$.*

- **Invariant wrt the state space model!**

Theoretical Results

- **Theoretical analysis**:
  - **convergence** of the predictive pdf of the observations:

$$\lim_{M\to\infty} \left(f, \hat{p}^M(y_t|y_{1:t-1})\right) = \left(f, p(y_t|y_{1:t-1})\right) \qquad \text{a.s.},$$

    with explicit **convergence rate**
    - extends the existing results of pointwise convergence of $\hat{p}^M(y_t|y_{1:t-1})$ to $\hat{p}(y_t|y_{1:t-1})$
    - holds for **multidimensional** observations
    - key for the statistical analysis of $A_{K,t}$
  - **convergence** of the p.m.f. of $A_{K,t}$ to a discrete uniform distribution

$$\frac{1}{K+1} - \varepsilon_M \leq \mathbb{Q}_K(n) \leq \frac{1}{K+1} + \varepsilon_M, \qquad n = 0, ..., K,$$

    with $\lim_{M\to\infty} \varepsilon_M = 0$ a.s.

[1] V. Elvira, J. Mguez, and P. M. Djuric, "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment", IEEE Transactions on Signal Processing, vol. 65, no. 7, pp. 1781-1794, 2017.

Adapting the number of particles

- **Generic framework** for online convergence assessment
- **Embedded** in your favorite PF
- Exploit the **properties** of $A_{K,t}$
- Different algorithms:
  - **Algorithm 1** [1][2]: exploit Prop. 1 $\Rightarrow$ **uniformity** of $A_{K,t}$
  - **Algorithm 2** [3]: exploit Prop. 2 $\Rightarrow$ **independence** of $A_{K,t}$

---

[1] V. Elvira, J. Mguez, and P. M. Djuric, "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment", IEEE Transactions on Signal Processing, vol. 65, no. 7, pp. 1781-1794, 2017.

[2] V. Elvira, J. Mguez, and P. M. Djuric, "Online adaptation of the number of particles of sequential Monte Carlo methods", IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2016), Shanghai, China, March, 2016.

[3] V. Elvira, J. Mguez, and P. M. Djuric, "A Novel Algorithm for Adapting the Number of Particles in Particle Filtering", Sensor Array and Multichannel Signal Processing Workshop (SAM 2016), Rio de Janeiro, Brazil, 2016.

Summary of the algorithms

- The algorithms work in windows of W time steps.

(Alg 1) Check the **uniformity** of the W consec. statistics [1][2]

- chi-square on the uniformity of $A_{K,t}$

(Alg 2) Check the **autocorrelation** of the W consec. statistics [3]

- t-test on the first-order autocorrelation

- The statistical test produces a p-val $p^*_{K,n}$ at each window:
  - If $p^*_{K,n} < p_\ell$, reject the null hypothesis, **increase** M
  - If $p^*_{K,n} > p_h$, **decrease** M
  - Otherwise, **keep** the same M
- **Intuition**: if the filter is lost, the predictions are biased in the same direction.

---

[1] V. Elvira, J. Mguez, and P. M. Djuric, "Adapting the number of particles in sequential Monte Carlo methods through an online schme for convergence assessment", submitted to IEEE Transactions on Signal Processing, 2015.

[2] V. Elvira, J. Mguez, and P. M. Djuric, "Online adaptation of the number of particles of sequential Monte Carlo methods", IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2016), Shanghai, China, March, 2016.

[3] V. Elvira, J. Mguez, and P. M. Djuric, "A Novel Algorithm for Adapting the Number of Particles in Particle Filtering", Sensor Array and Multichannel Signal Processing Workshop (SAM 2016), Rio de Janeiro, Brazil, 2016.

Algorithms: beyond

- $A_{K,t}$ can be exploited for novel algorithms
- Desirable properties:
  - **stability**: for a specific model, the algorithm should behave always similarly
  - **few parameters**: almost no parameters to be tuned (self-adaptive algorithms)
    - with **physical meaning**, e.g. pre-selected operation point in the complexity-performance trade-off
- Ongoing novel solutions:
  - Check **jointly** the uniformity and the autocorrelation at the same time $\Rightarrow$ they detect different malfunctioning situations
  - **Auto-tune** algorithm parameters $\Rightarrow$ freezing the adaptation
  - **Go back** in time (if your application can afford it)
  - **...**

Alg. 1 (uniformity): Lorenz63 System

- 3-dimensional dynamical system defined by

$$
\begin{aligned}
dX_1 &= -\mathrm{s}(X_1 - X_2), \\
dX_2 &= \mathrm{r}X_1 - X_2 - X_1X_3, \\
dX_3 &= X_1X_2 - \mathrm{b}X_3,
\end{aligned}
$$

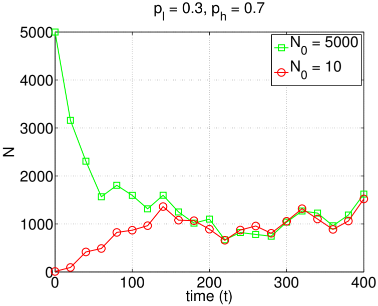
- Time discrete version using Euler's method with

$$
\begin{aligned}
X_{1,n} &= X_{1,n-1} - \Delta\mathrm{s}(X_{1,n-1} - X_{2,n-1}) + \sqrt{\Delta}U_{1,n}, \\
X_{2,n} &= X_{2,n-1} + \Delta(\mathrm{r}X_{1,n-1} - X_{2,n-1} - X_{1,n-1}X_{3,n-1}) + \sqrt{\Delta}U_{2,n}, \\
X_{3,n} &= X_{3,n-1} + \Delta(X_{1,n-1}X_{2,n-1} - \mathrm{b}X_{3,n-1}) + \sqrt{\Delta}U_{3,n},
\end{aligned}
$$

- $U_{i,n} \sim \mathcal{N}(0,1)$, $\Delta = 10^{-3}$, and $(\mathrm{s},\mathrm{r},\mathrm{b}) = \left(10, 28, \frac{8}{3}\right)$

Alg. 1 (uniformity): Lorenz System

- Algorithm checking the uniformity of the statistic.
- $K = 7$ fictitious observations and $W = 20$



Alg. 2 (autocorrelation): Stochastic Growth Model

- Stochastic growth model:

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(\phi t) + u_t, \quad (1)$$

$$y_t = \frac{x_t^2}{20} + v_t, \quad (2)$$

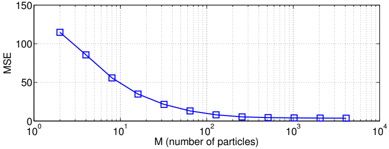
with $\phi = 0.4$, u_t and v_t are i.i.d. zero-mean univariate Gaussian r.v.'s with variance $\sigma_u^2 = 2$ and $\sigma_v^2 = 0.1$. $T = 10^4$.

- Algorithm parameters: $K = 7$, $W = 25$

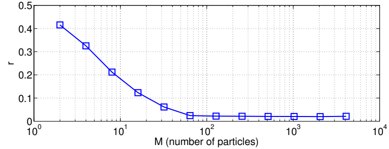
$[p_l - p_h]$	$[0.2 - 0.6]$	$[0.25 - 0.65]$	$[0.35 - 0.75]$	$[0.4 - 0.8]$	$[0.45 - 0.85]$
MSE	21.62	13.83	4.90	3.62	3.39
M	144	386	1933	2841	3255
ex. time (s)	18.9	233.4	285.7	441.5	536.1

Alg. 2 (autocorrelation): Stochastic Growth Model

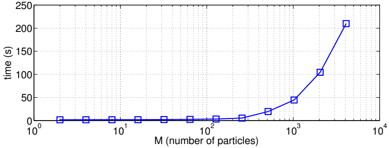
(a) **MSE** with fixed number of particles



(b) **Autocorrelation** with fixed number of particles



(c) **Execution time** with fixed number of particles



Conclusions

- **Different SSM** require **different** M for operating at the same level of accuracy
  - even the same model can require different M at different stages
- We propose a **convergence assessment** method for PF in **real time**
  - model invariant
  - theoretically sound
- Algorithms for **adapting** the number of particles
  - based on the **independence** and **autocorrelation** of $A_{K,t}$
  - **robust** to initialization and choice of parameters
  - allow for the selection of the **operation point** at the performance/computational **tradeoff**

Thank you for your attention!