Online filtering problems for state space models

Yun, J., Yang, F., and Chen, Y. 2017. *JASA.*

Jonghyun Yun

State space model (SSM)

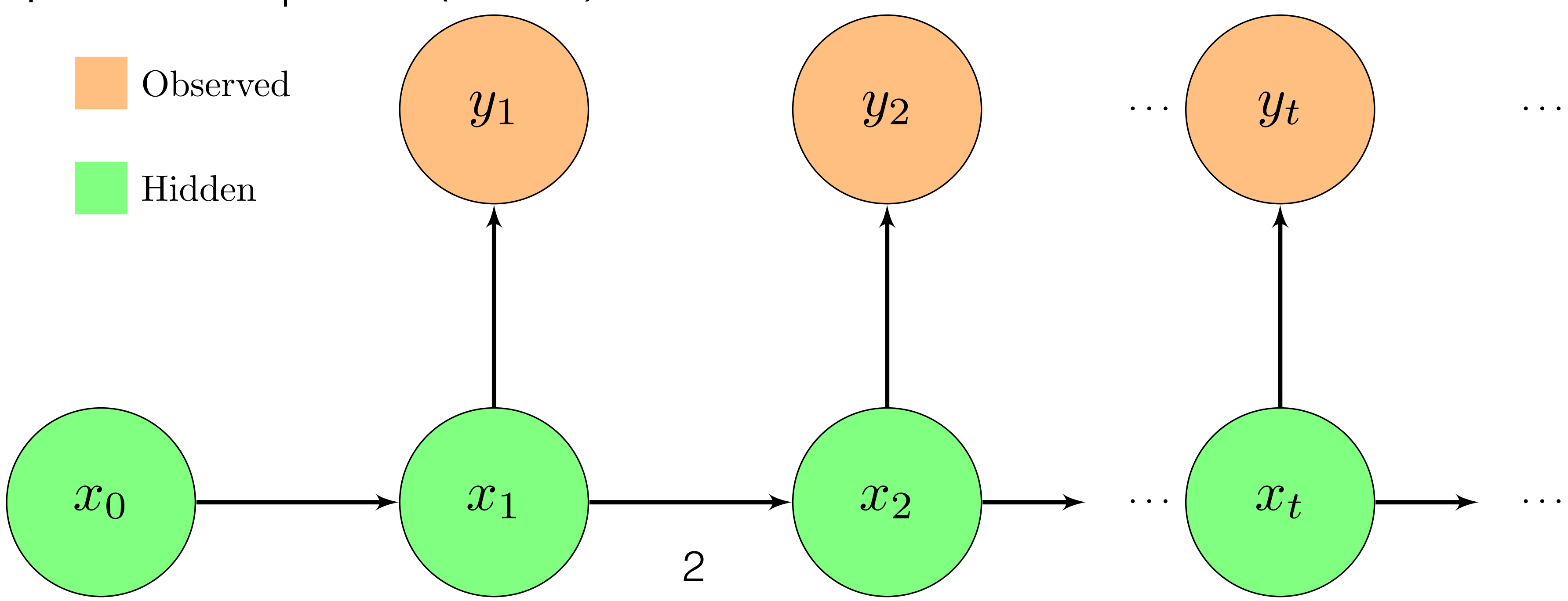
- SSM (also called hidden Markov model) provides flexible modeling framework for spatial or temporal processes.

- Hidden state vector x_t evolves through the state dynamic, and a observation y_t is a noisy function of x_t

  - measurement equation: $y_t | x_t = h_t(x_t, u_t)$

  - state equation: $x_t | x_{t-1} = f_t(x_{t-1}, v_t)$

  -



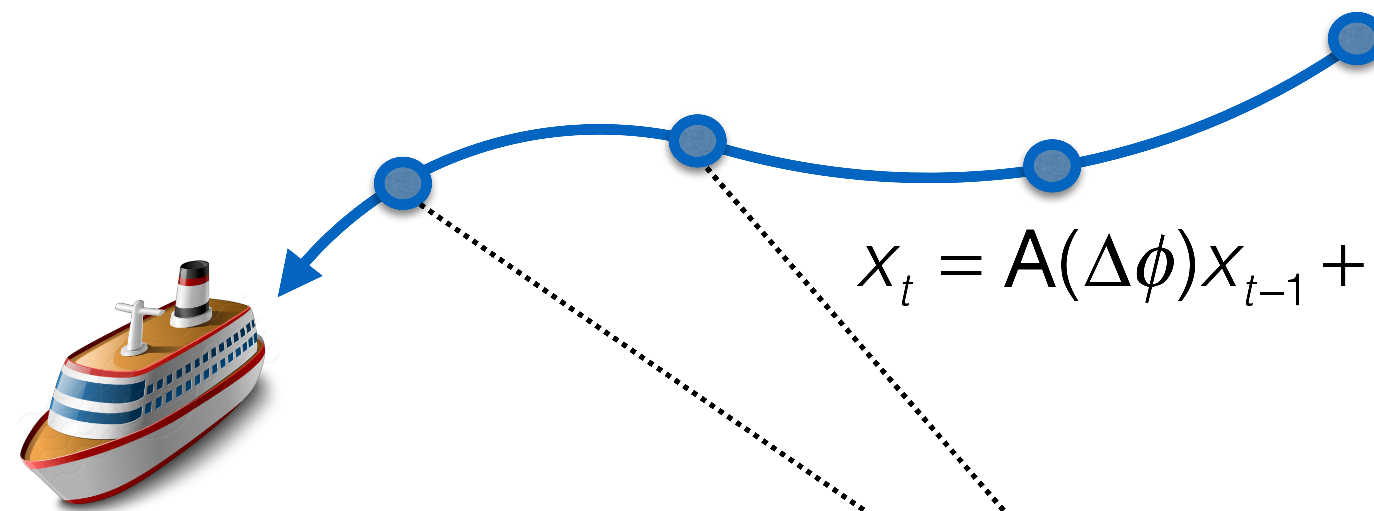
What is online filtering?

- The *online* filtering problem is to obtain $E[g(x_{0:t})|y_{1:t}]$

$x_{0:t}=(x_0,\dots,x_t),\ y_{1:t}=(y_1,\dots,y_t)$

- Speed requirement
  - Data arrive sequentially and we need to update the inference online before the next observation arrives
- In most cases, the target distribution $p(x_{0:t}|y_{1:t})$ or $p(x_t|y_{1:t})$ is not analytically tractable and difficult to sample directly
- The Markov chain Monte Carlo (MCMC) method is not convenient for online inference.

Maneuvering target tracking



$$x_t = \mathrm{A}(\Delta\phi)x_{t-1} + \mathrm{B}(\Delta\phi)v_t$$

$$y_t = h(x_t) + u_t$$

- Goal: estimate $E(x_t|y_{1:t})$ to track a maneuvering target over time
  - Observation y_t: angle and distance of the target from the origin measured by a radar
  - Hidden state x_t: position, velocity and acceleration of the target

Particle filter (PF): how it works?

- PF (also called sequential Monte Carlo methods) is a method to approximate $p(x_{0:t}|y_{1:t})$ by a large number of weighted samples that are sequentially generated over time from distributions proportional to $p(y_t|x_t)p(x_t|x_{t-1})$ using importance sampling and resampling

$$p(x_{0:t} \mid y_{1:t}) = \frac{1}{Z_t} p(x_0) \prod_{k=1}^{t} p(y_k \mid x_k) p(x_k \mid x_{k-1})$$

normalizing constant

$$Z_t = p(y_{1:t})$$

- How the proposal distribution for importance sampling should be chosen?

Reviw: PF

- Initialize particles $x_0^{(i)} \sim p(x_0)$ for $i=1,\ldots,N$

- At each t, draw $x_t^{(i)}$ from the proposal $q(x_t|y_t, x_{t-1}^{(i)})$, and update the importance weight

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t \mid x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)})}{q(x_t^{(i)} \mid y_t, x_{t-1}^{(i)})}$$

normalized weight

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}$$

- Convergence of PF estimates

$$\sum_{i=1}^{N} \tilde{w}_t^{(i)} g(X_{0:t}^{(i)}) \xrightarrow{a.s.} E(g(X_{0:t}) \mid Y_{1:t}) \text{ as } N \uparrow \infty$$

-

Some existing PFs

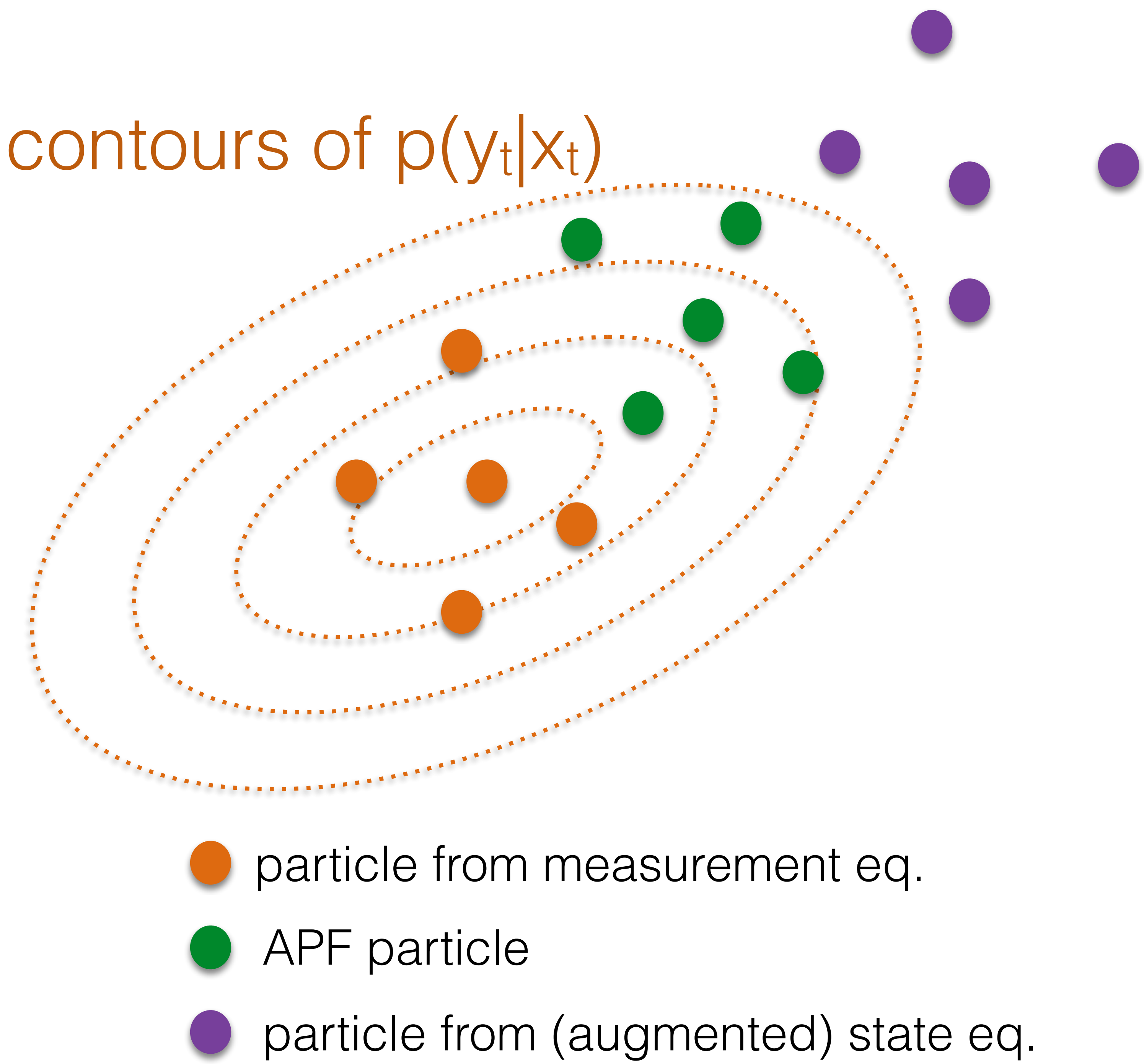
- Efficiency of the PF depends heavily on the quality of the proposal distribution, and it can be measured by the variance of importance weights.

- Naive particle filter (NPF, bootstrap filter) uses the proposal based on the state equation: $q(x_t|y_t,x_{t-1})=p(x_t|x_{t-1})$

- Independent particle filter (IPF, Lin et al. 2005 *JASA*) uses the proposal based on the measurement equation: $q(x_t|y_t,x_{t-1})=q(x_t|y_t)$

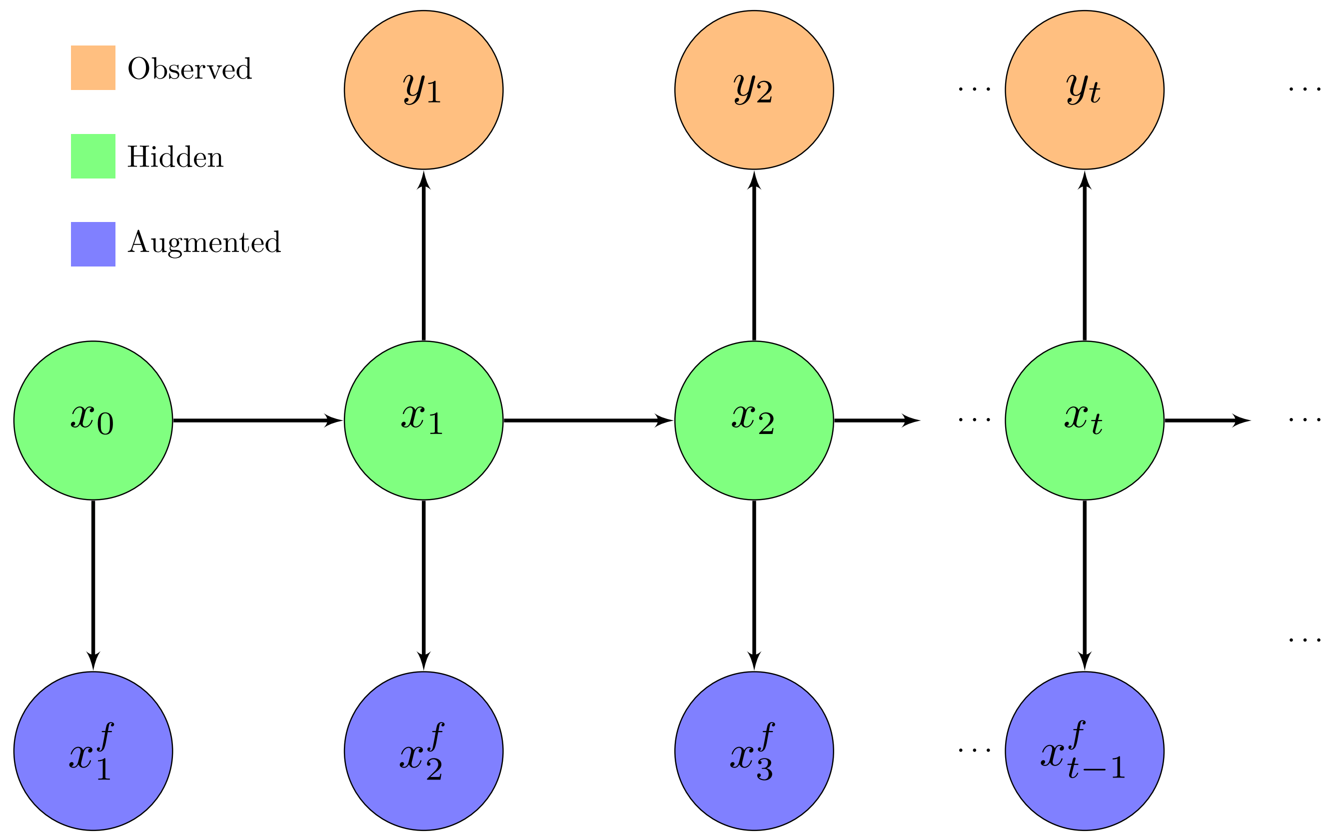
- Optimal particle filter (OPF) uses the proposal based on the two equations: $q(x_t|y_t,x_{t-1}) = p(x_t|y_t,x_{t-1})$

Augmented PF: how it works?

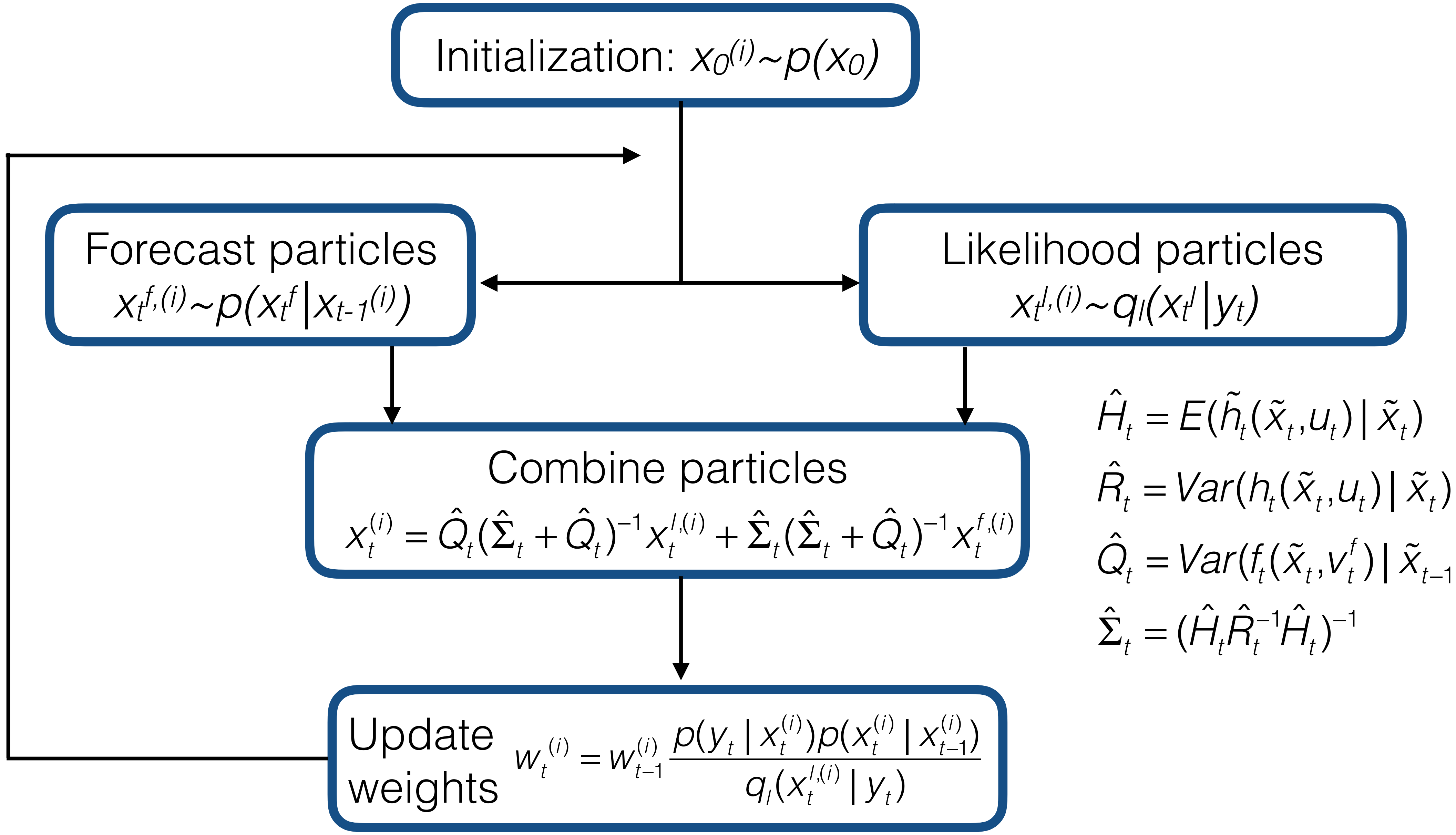


- APF linearly combines two sets of particles: particles from measurement eq. and those from (augmented) state eq.
- Weights for linear combinations are determined by covariance matrices of noise terms.
- APF proposal approximates OPF proposal.

Augmented state space



- $y_t | x_t = h_t(x_t, u_t)$: measurement eq.
- $x_t | x_{t-1} = f_t(x_{t-1}, v_t)$: state eq.
- $x_t^f | x_{t-1} = f_t(x_{t-1}, v_t^f)$: augmented state eq.
- Given x_{t-1}, x_t^f is free of all other components in the SSM



- Theorem. In the APF, the importance weight can be computed recursively as

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t | x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)})}{q_l(x_t^{l,(i)} | y_t)}$$

Remarks on APF

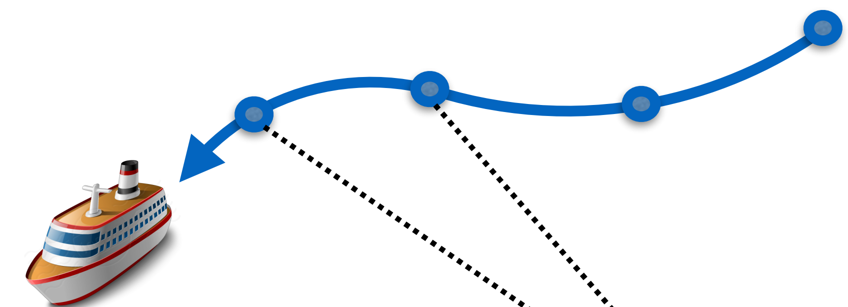
- The marginal posterior of x_t from the APF is the same as the target distribution of x_t under the standard SSM. We can still make valid inference for the hidden states.

- APF can be viewed as a combination of the IPF and the NPF. Unlike the OPF, the APF can be implemented for general SSMs. Augments state vectors facilitate sampling and weight computation in APF

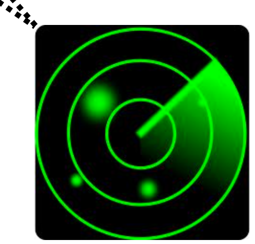
- What will happen if we do not augment the state space? The marginalized proposal density needs to be evaluated through approximation of the integrals for each i

$$
\begin{aligned}
&q(x_t^{(i)} \mid y_t, x_{t-1}^{(i)}) \\
&= \int q_l(x_t \mid y_t, x_t^f) p(x_t^f \mid x_{t-1}^{(i)}) dx_t^f \\
&= |(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1})^{-1} \hat{\Sigma}_t^{-1}| \int q_l(\hat{\Sigma}_t \{(\hat{\Sigma}_t^{-1} + \hat{Q}_t^{-1}) x_t^{(i)} - \hat{Q}_t^{-1} x_t^{f,(i)}\} \mid y_t) p(x_t^{f,(i)} \mid x_{t-1}^{(i)}) dx_t^{f,(i)}
\end{aligned}
$$

Simulation: target tracking



- Lin et al. 2005, *JASA*
- State vector $x_t \in R^6$: position, velocity, and acceleration in 2D plane.



- Observation vector $y_t \in R^2$: angle and distance of the target
- Cauchy noise $v_t \in R^2$: $p(v_{t,i}) = \dfrac{1}{\pi(v_{t,i}^2 + 1^2)}$.

$$\begin{cases} y_t \mid x_t = h(x_t) + u_t; \\ x_t \mid x_{t-1} = A(\Delta\phi)x_{t-1} + B(\Delta\phi)v_t. \end{cases}$$

- $h(x_t) = \left(\arctan\left(\dfrac{x_{t,1}}{x_{t,2}}\right), \sqrt{x_{t,1}^2 + x_{t,2}^2} \right)$. $\quad u_t \sim N(0,R),\ R = \sigma \begin{bmatrix} 10^{-10} & 0 \\ 0 & 10^{-2} \end{bmatrix}$.

- UPF (Merwe et al. 2000 *NIPS*): unscented PF uses the unscented transformation to construct the Gaussian proposal

- ImPF (Chorin et al. 2009 *PNAS*): implicit PF uses the quadratic approximation of *-log[p(y$_t$|x$_t$)p(x$_t$|x$_{t-1}$)]*

- Error measure:

$$\hat{X}_{0:T} = (\hat{E}(X_0 \mid y_1), \hat{E}(X_1 \mid y_{1:2}), \ldots, \hat{E}(X_T \mid y_{1:T}))$$

$$\text{RMSE} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{T}\| \hat{X}_{0:T}^k - X_{0:T}^k \|^2}$$

$$\text{se(RMSE)} = \sqrt{\frac{1}{K}\widehat{Var}\left(\sqrt{\frac{1}{T}\| \hat{X}_{0:T}^k - X_{0:T}^k \|^2}\right)}$$

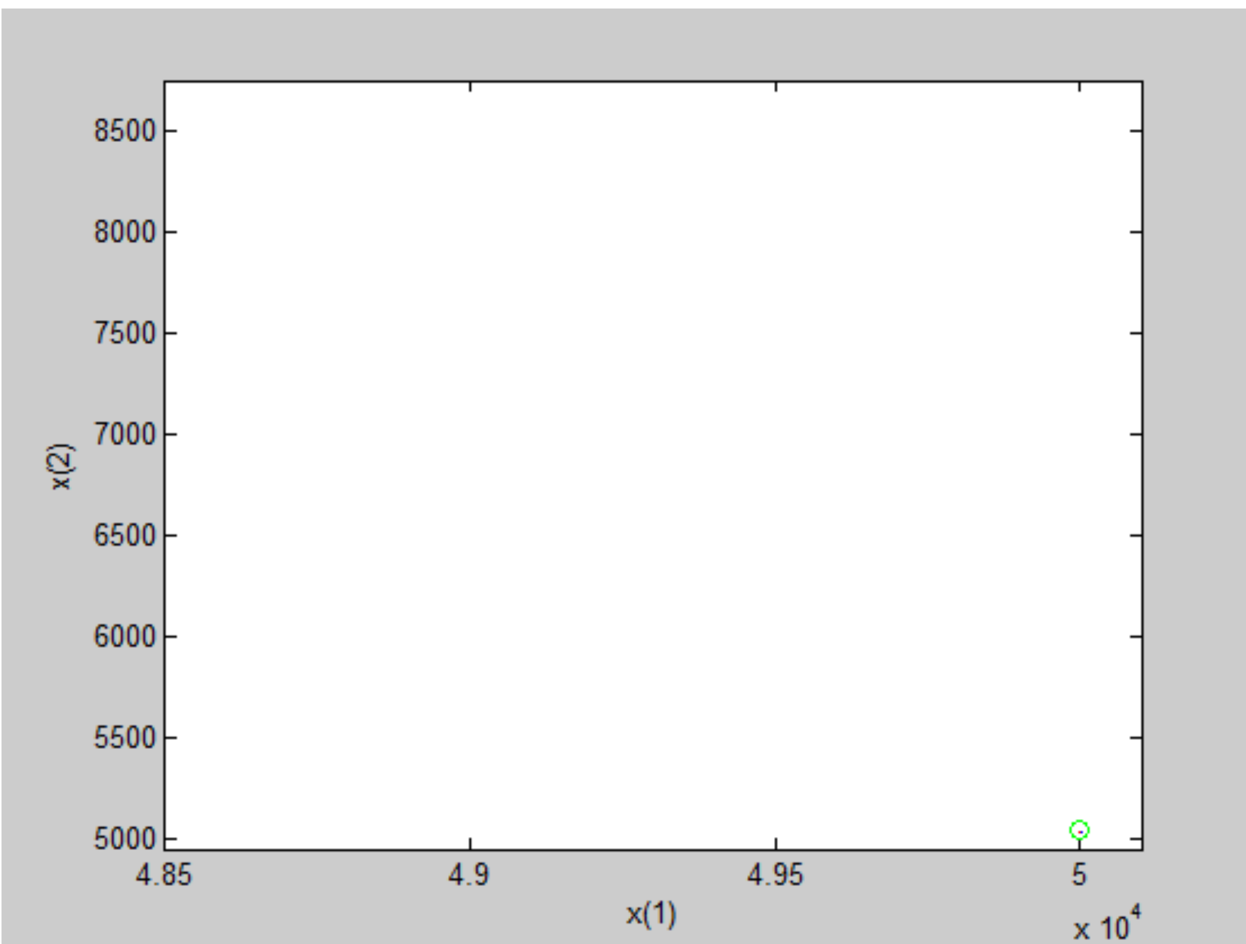
- N=1,000, T=375 (3.75s in real time)

measurement noise level

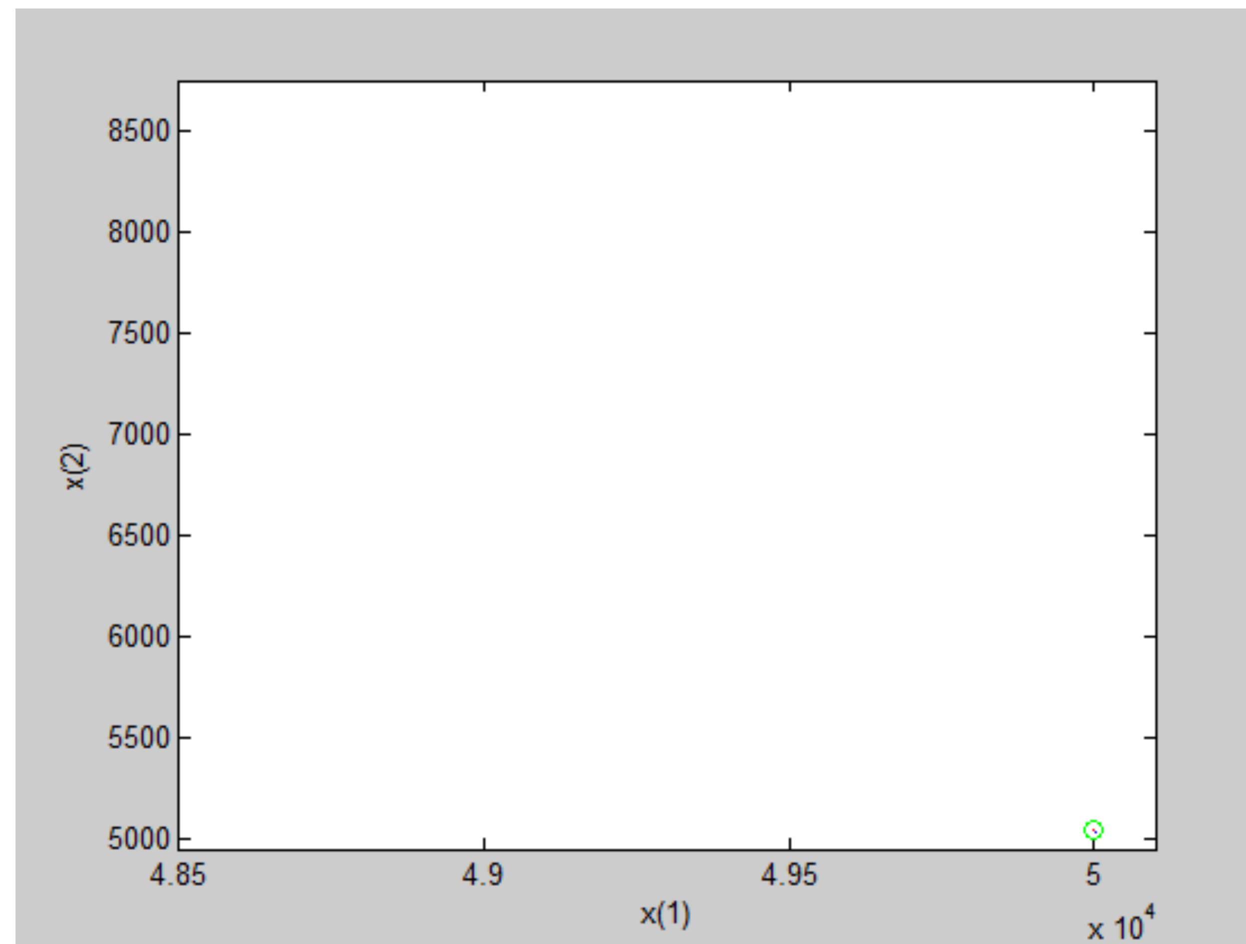
$\sigma = 0.01$	RMSE	se(RMSE)	CPU Time (sec)
APF	1.770	1.101	1.512
IPF	3.183	1.069	1.053
NPF	2820.432	431.089	0.926
UPF	4.889	2.924	282.140
ImPF	188.753	6.125	21991.150
$\sigma = 1$			
APF	1.925	1.075	1.591
IPF	5.670	0.981	1.083
NPF	2590.967	1215.372	0.995
UPF	3.547	0.803	281.150
ImPF	28.650	1.325	28973.821
$\sigma = 100$			
APF	9.494	1.216	1.535
IPF	43.987	2.565	1.056
NPF	520.428	349.574	0.966
UPF	11.042	0.595	274.880
ImPF	16.566	1.104	29715.241

True vs estimated trajectories

APF



NPF



NPF lost track of the target after its sudden change of the direction

Take home messages

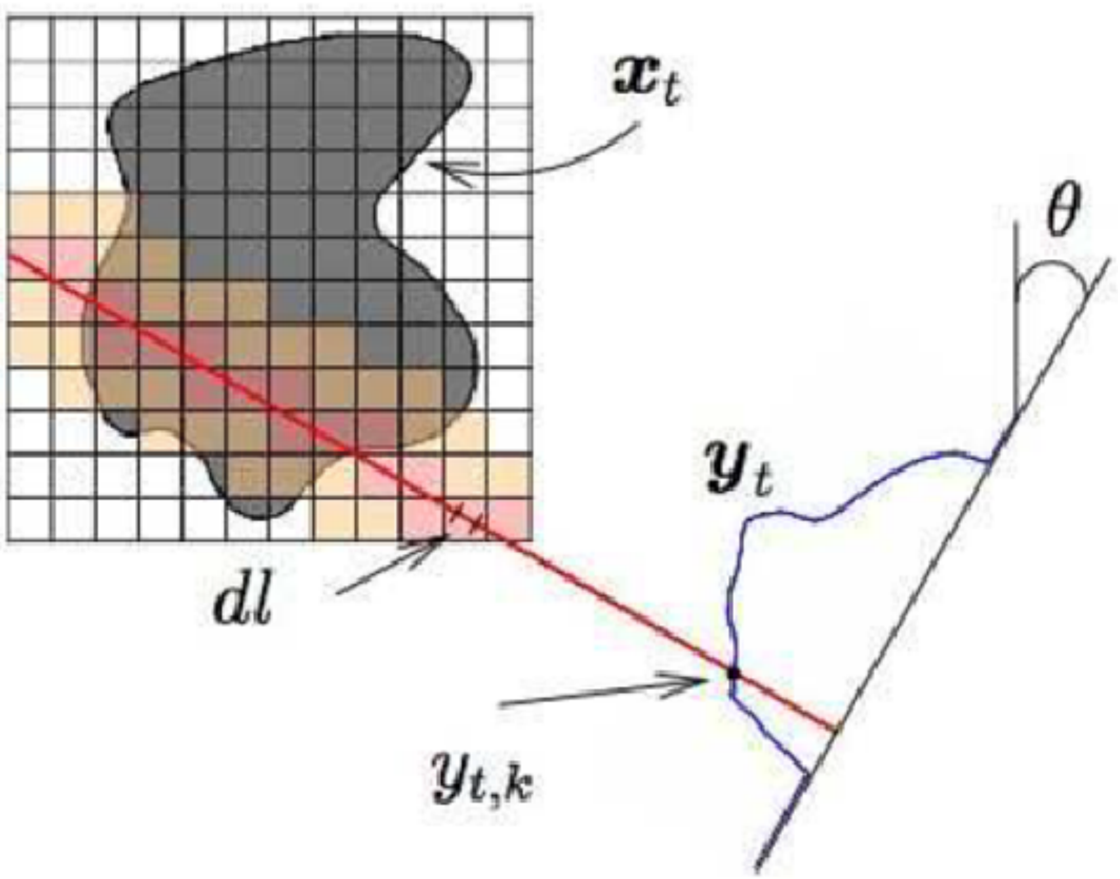
- The augmented state space facilitates the weight computation, and guides to an effective proposal.

- The APF strikes balance between state and measurement equations, and it performs better than other sub-optimal filters.

- The APF outperforms existing methods for high-dimensional SSMs under partially/fully observed noisy state vectors.

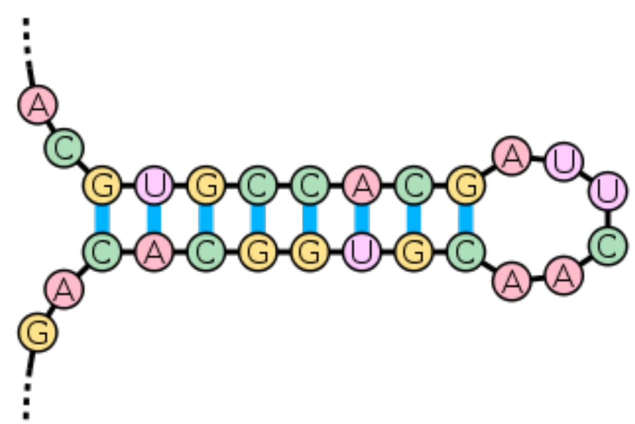
Future Works

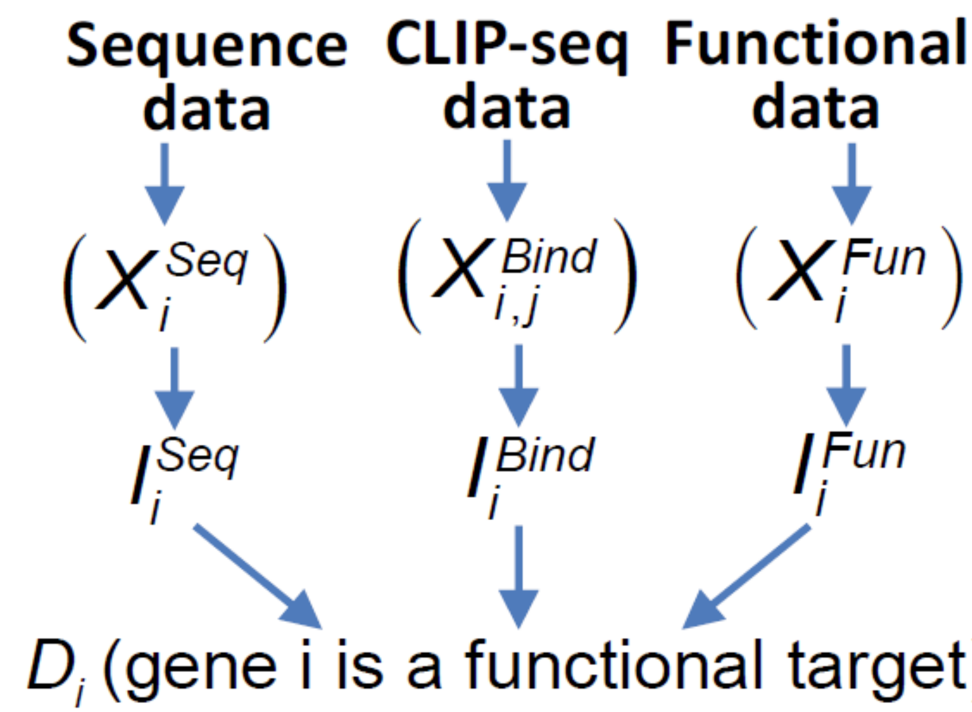
- Localized PF
  - Sparse update of state vectors for high-dimensional particle filtering
  - Each component of y_t is a projection of a few components of x_t
- Response time data
  - To understand behavioral patterns of respondents and interaction with items.
  - To be served as a network model where we have connection type and time.



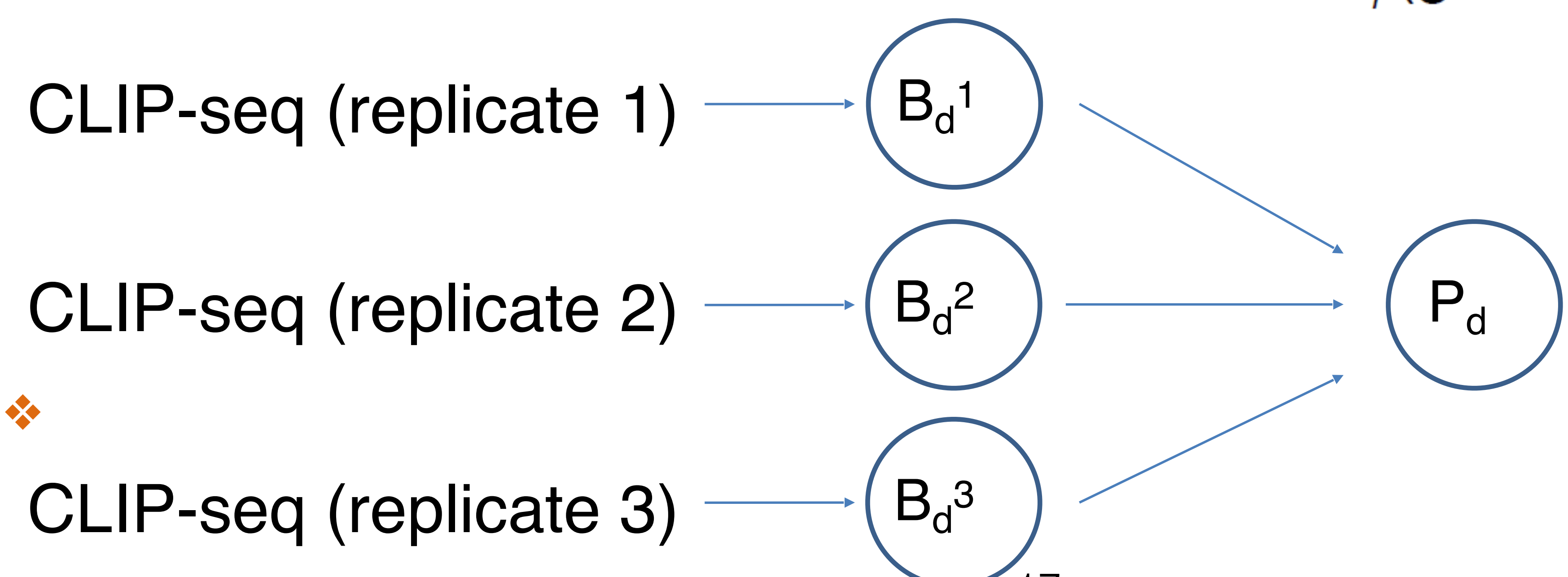
Future Works: NGS analysis

- Integrative analysis of NGS data
    - Motif sequence
    - CLIP-seq
    - Functional data





- CLIP-seq with biological replicates



To identify population level binding sites for each region (d).