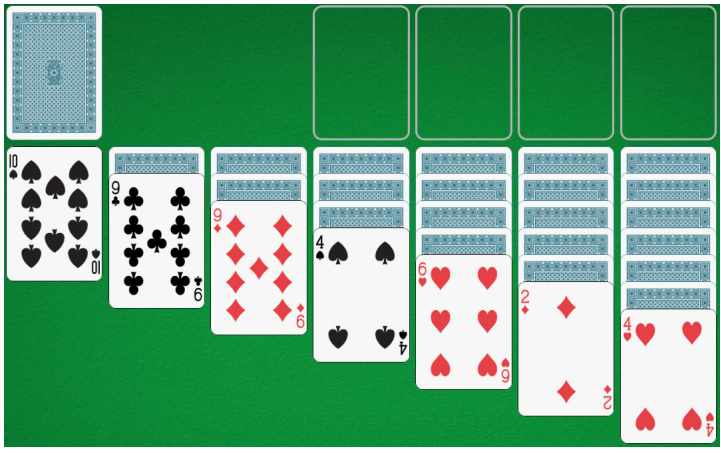
Los comienzos de Monte Carlo y Metropolis

La existencia de los metods de Monte Carlo se la debemos al solitario.



La historia cuenta que Stan Ulam estaba postrado enfermo jugando al solitario y quizo saber la probabilidad que tenia de que el juego fuera exitoso.

El manejo combinatorio es extremadamente complicado sino imposible, por lo que rapidamente desistio, y propuso jugar una gran cantidad de veces y registrar la cantidad de veces que era exitoso.

Si las muestras eran aleatorias, y la cantidad de juegos grande esto se deberia aproximar al valor exacto.

Nicholas Metropolis & Stan Ulam (1949) The Monte Carlo Method, Journal of the American Statistical Association, 44:247, 335-341

Este seminal articulo introduce:

- los métodos de partículas de Monte Carlo tanto para resolver la ecuación de Fokker-Planck a traves de muestras.
- la evaluación de integrales multidimensionales a través de muestras.

El nombre de Monte Carlo también viene de Stan Ulam y en principio la avidez de su tio por jugar a las cartas, en particular al poker en el casino Mónaco.

Aplicaciones de los métodos de Monte Carlo

En inferencia la aplicacion principal seria la evaluacion de integrales de esperanzas:

$$\mathcal{E}_p(f(\mathbf{x})) \triangleq \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$$

Marginalización

$$p(\mathbf{x}|\mathbf{y}) = \int p(\mathbf{x}, \mathbf{z}|\mathbf{y})\mathrm{d}\mathbf{z}$$

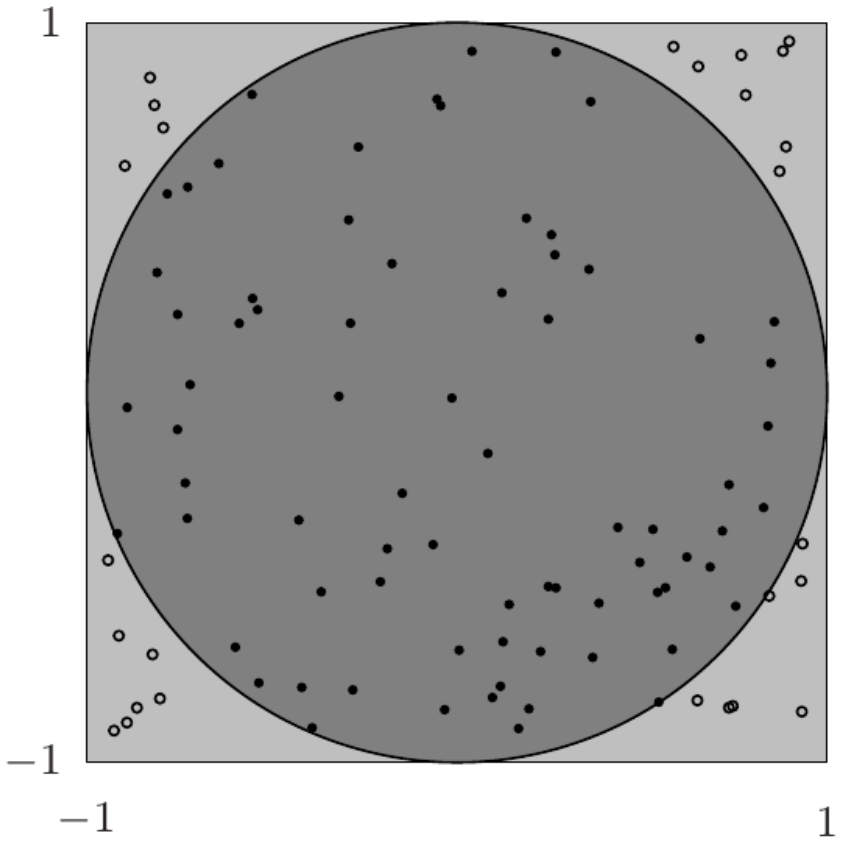
Normalización

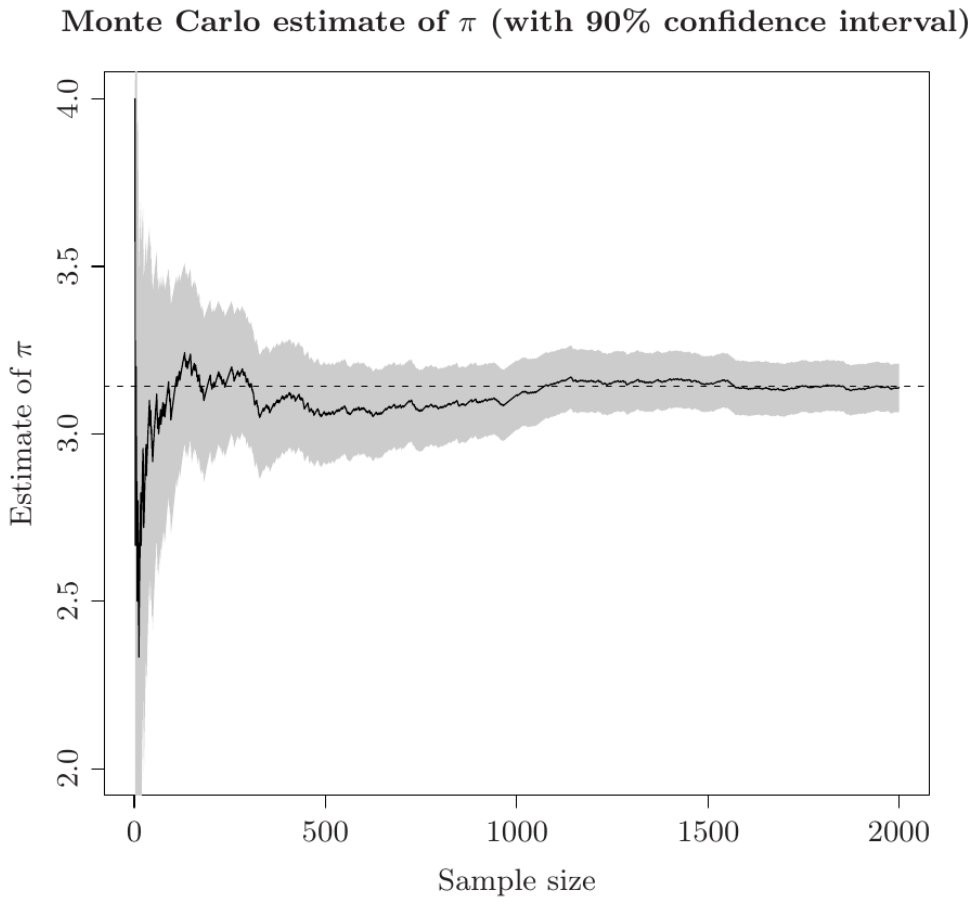
$$F = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$$

También para optimización estocástica y para model selección (por máximo verosimilitud).

Estimación de π con Monte Carlo

$A = \pi(1/2)^2$ entonces $\pi = 4A = 4\frac{n_{cir}}{n}$



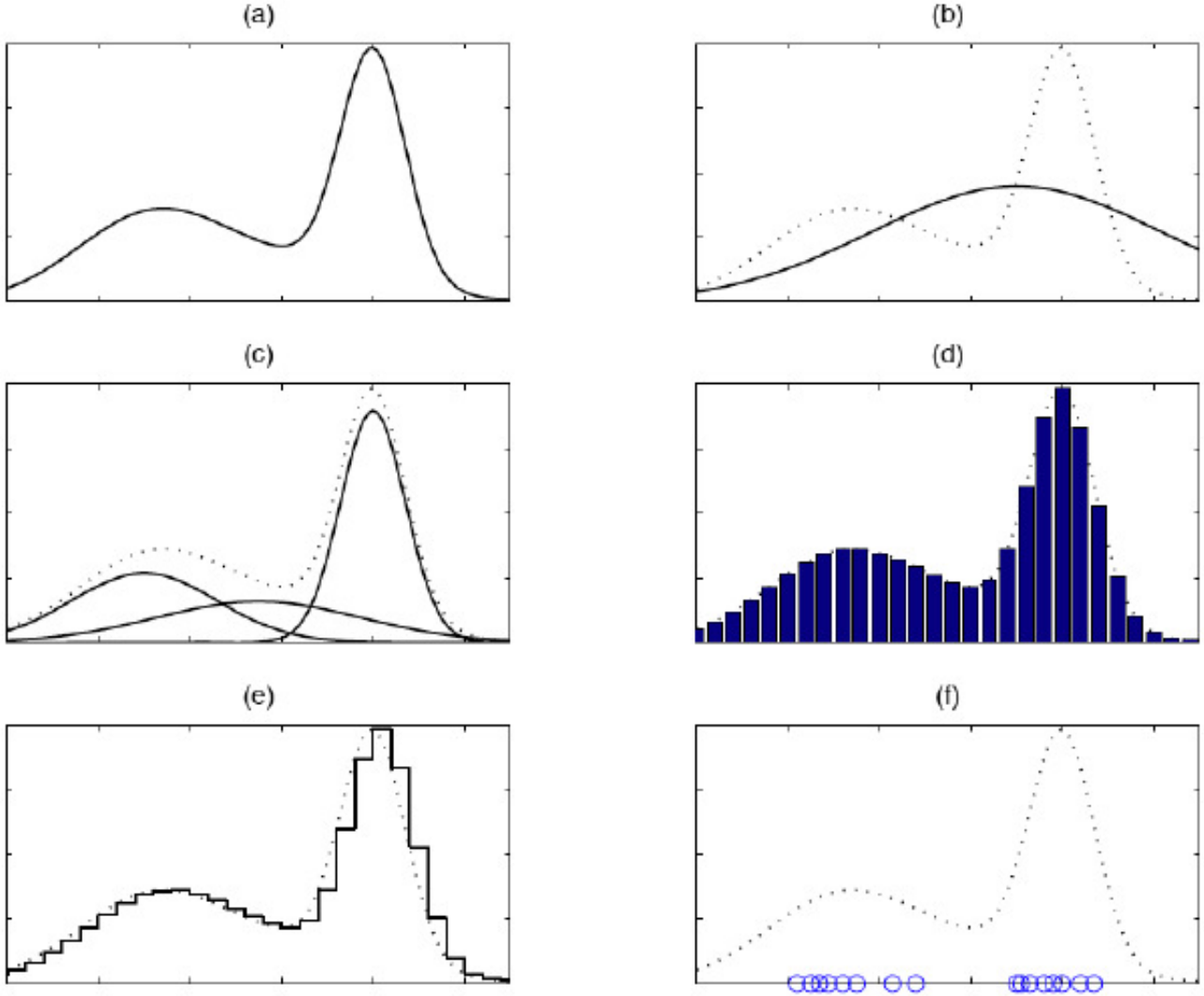


Referencias de Monte Carlo y muestreo (sampling)

- Andrieu, C., N. de Freitas, A. Doucet, and M. I. Jordan, 2003: An introduction to MCMC for machine learning. Machine Learning 50, 5-43.
- Liu, J. S. 2001. Monte Carlo Strategies in Scientific Computing. Springer.
- Cappé, O., Moulines, E. and Rydén, T. 2005. Inference in Hidden Markov Models. Springer, New York, NY.

Muestreo de Monte Carlo

Formas de representar a una densidad de probabilidad.



Muestreo y Monte Carlo

El objetivo es evaluar la integral

$$\mathcal{E}_p(f(\mathbf{x})) \triangleq \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$$

Lo que se hace es tomar una muestra de N realizaciones independientes de la densidad de probabilidad p (i.i.d.).
Tomamos como representación de la densidad de probabilidad a la muestra

$$p(\mathbf{x}) \doteq \frac{1}{N}\sum_{j=1}^{N}\delta(\mathbf{x}-\mathbf{x}^{(j)})$$

Por lo que el estimador de la integral resulta

$$\mathcal{E}_p^{MC}(f(\mathbf{x})) \doteq \frac{1}{N_p}\sum_{j=1}^{N_p} f(\mathbf{x}^{(j)})$$

Error en la estimación de Monte Carlo

Cuando N_p es lo suficientemente grande, el error de aproximación es arbitrariamente pequeño.
Varianza del estimador:

$$\sigma_N^2(p,f) = \frac{1}{N_p}\mathcal{E}_p[(f - \mathcal{E}(f))^2]$$

Asumiendo la muestra $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_p)}\}$ de p

$$\sigma_N^2(p,f) = \frac{1}{N_p}\sum_{j=1}^{N_p}[f(\mathbf{x}^{(j)}) - \mathcal{E}_p^{MC}(f(\mathbf{x}))]^2$$

El intervalo de confianza es $\mathcal{E}_p^{MC}(f(\mathbf{x})) \pm c_\alpha N_p^{-1/2}\sigma_N(p,f)$
Lo importante en la razón de convergencia $N_p^{-1/2}$ es que esta no depende de la dimensionalidad.

Ejemplo integral Monte Carlo

La integral que define la media:

$$\mathcal{E}_p(\mathbf{x}) \triangleq \int \mathbf{x} p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

dadas $\mathbf{x}_k^{1:N_p}$ iid se pueda aproximar por

$$\hat{\mathbf{x}}_k = \frac{1}{N_p} \sum_j^{N_p} \mathbf{x}_k^j$$

Y la covarianza viene dada por

$$\hat{\mathbf{P}}_k = \frac{1}{N_p} \sum_j^{N_p} (\mathbf{x}_k^j - \hat{\mathbf{x}}_k)(\mathbf{x}_k^j - \hat{\mathbf{x}}_k)^\top$$

conocida por sample covariance, o sample estimates. Las cuales convergen para $N_p \to \infty$. El error en la estimación decae como $N_e^{-1/2}$.

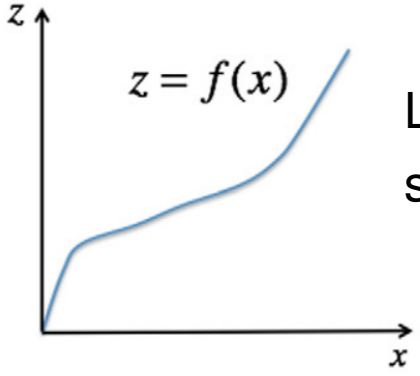
Es una forma alternativa de calcular las covarianzas del forecast y el análisis (iid!).

Detalles para su implementación

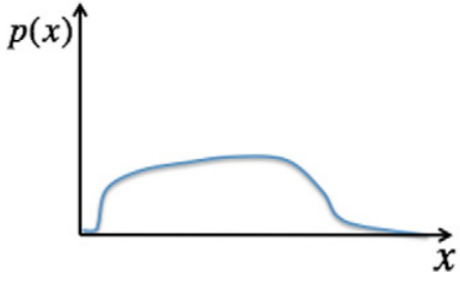
1. Necesitamos un método de muestreo de densidades de probabilidad arbitrarias.
2. Si las muestras no son independientes hay que tener en cuenta la effective sample size.
3. Si $f(\mathbf{x})$ es pequeño donde $p(\mathbf{x})$ es grande entonces la esperanza será dominada por regiones de baja probabilidad por lo que vamos a requerir una gran cantidad de muestras para lograr la resolución requerida.

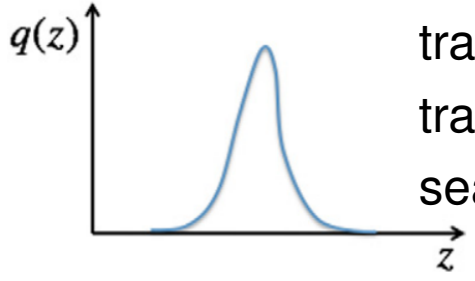
Transformación de densidades

Una forma de muestreo sería obtener muestras de una densidad simple, e.j. la uniforme, y luego transformamos a la densidad deseada,
Asumiendo tengo una variable aleatoria X y quiero transformar esta variable aleatoria con un mapa f entonces tengo una nueva variable aleatoria $Z = f(X)$.



Las densidades de estas variables aleatorias se relacionan por $q_z(z) = p(f^{-1}(z)) \left| \frac{\mathrm{d}f^{-1}(z)}{\mathrm{d}z} \right|$





El problema consiste en encontrar un mapa f que nos permita transformar a la densidad deseada.

Rejection sampling

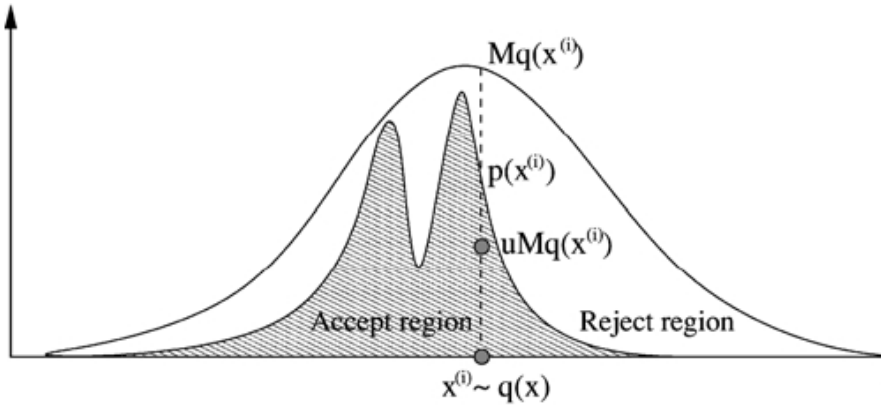
Queremos encontrar muestras de una densidad que se conoce analíticamente, y por lo tanto podemos evaluarla, a menos de una constante de normalización

$$p(\mathbf{x}) = \frac{1}{N}\tilde{p}(\mathbf{x})$$

Tomamos una distribución mas simple a la cual llamamos proposal density, $q(\mathbf{x})$, de la cual si podemos generar muestras aleatorias.
El soporte de q debe incluir al soporte de p.

La idea es que dado un $\mathbf{x}_0$ aleatorio, en base a otro numero aleatorio vemos si corresponde a $[0, p(\mathbf{x}_0)]$ o a $(p(\mathbf{x}_0), q(\mathbf{x}_0)]$.

- Si el segundo numero aleatorio cae en el primer intervalo nos quedamos con la muestra.
- Si cae en el segundo intervalo desechamos la muestra.



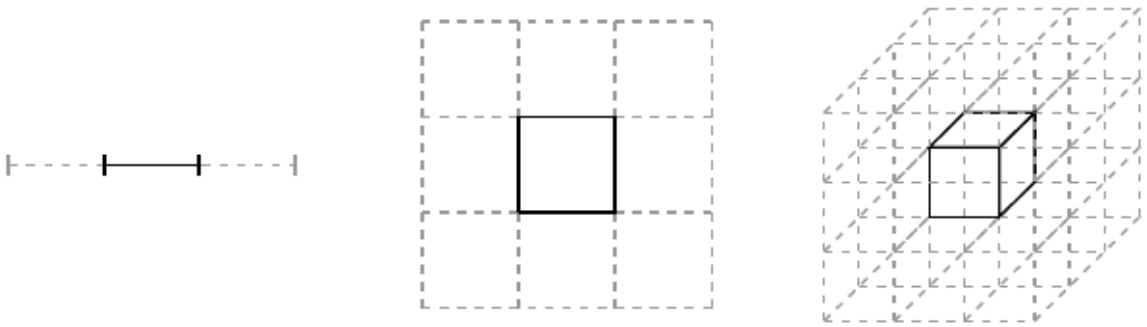
Algoritmo de rejection sampling

1. Se busca una constante k tal que se satisfaga que $kq(\mathbf{x}) \geq \tilde{p}(\mathbf{x})$ para todo $\mathbf{x}$, es decir que la curva de $kq(\mathbf{x})$ siempre cubre a la curva de la densidad de interés $\tilde{p}(\mathbf{x})$. (optimamente el ínfimo de las k posibles).
2. Se generan un número/vector aleatorio $\mathbf{x}_0$ de la distribución $q(\mathbf{x})$.
3. Dada la muestra $\mathbf{x}_0$ se genera otro número aleatorio, u_0, de la distribución uniforme $\mathcal{U}[0, kq(\mathbf{x}_0)]$
4. Una vez que se tienen los dos números aleatorios se compara u_0 con $\tilde{p}(\mathbf{x}_0)$ si $u_0 < \tilde{p}(x_0)$, lo mantenemos como una muestra de $\tilde{p}$ si en cambio $u_0 > \tilde{p}(x_0)$ se rejecta y seguimos buscando nuevas muestras.

Lamentablemente la probabilidad de rejección crece exponencialmente con la dimensionalidad.

El curso de la dimensionalidad

Histogramas. Si estamos representando a las densidades de probabilidad a través de los histogramas, lo que hacemos es dividir al espacio en celdas y contar la cantidad de eventos dentro de cada celda.
En principio para que la técnica funcione se asume en cada celda tenemos muchas realizaciones.
Al aumentar la dimensionalidad aumenta exponencialmente la cantidad de celdas requeridas para representar a un dado volumen. Ergo la cantidad de realizaciones requeridas aumenta también exponencialmente. Si tenemos un intervalo en $\mathbb{R}$ dividido en 50, en un espacio $N_x = 1000$ vamos a tener 50^{1000} celdas.



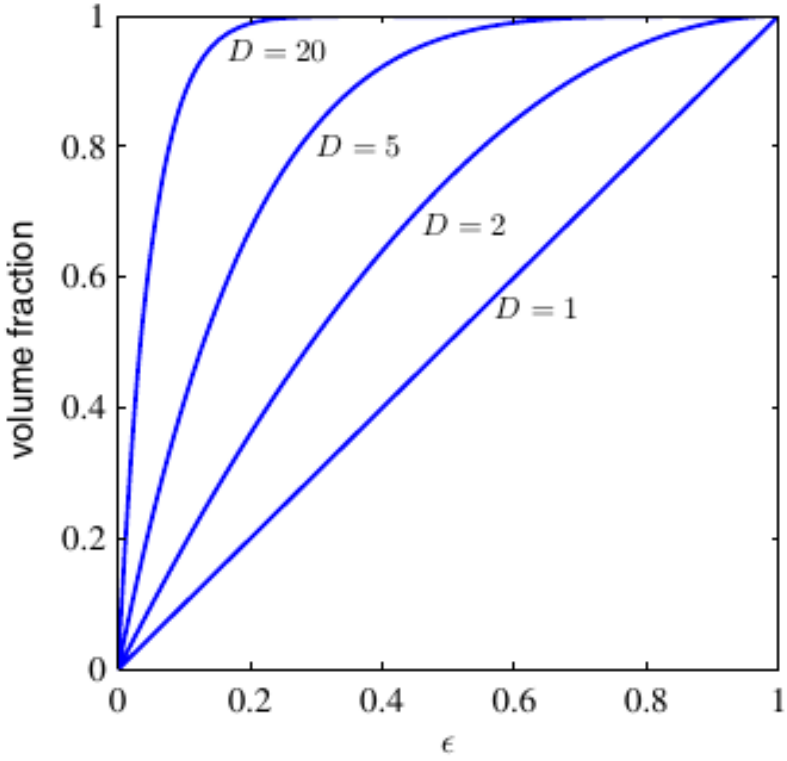
El curso de la dimensionalidad

Supongamos una esfera de radio $r = 1$ en un espacio de N_x dimensiones. Que fracción del volumen se encuentra entre $r = 1 - \epsilon$ y $r = 1$? El volumen es

$$V_{N_x}(r) = K_N r^{N_x}$$

Luego la diferencia de volumen es:

$$\frac{V_{N_x}(1) - V_{N_x}(1-\epsilon)}{V_{N_x}} = 1 - (1-\epsilon)^{N_x}$$

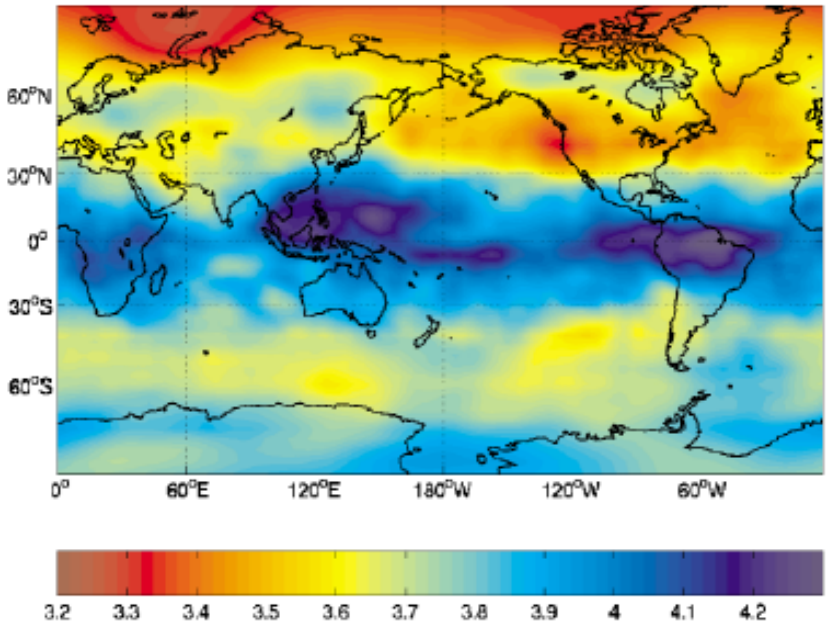


Para N_x grande todo el volumen de la esfera esta concentrado en un cascarón cerca de la superficie.

Las superficies de alta probabilidad en espacios de alta dimensionalidad van a estar en superficies “lejos” de la moda.

El curso de la dimensionalidad

En los sistemas complejos (atmósfera), el estado esta confinado a manifolds de mucha mas pequeña dimensionalidad dentro del espacio, por lo que la dimensionalidad puede ser mucho menor a la del espacio.



Ej. Patil et al 2001 PRL mostraron a través de análisis de componentes principales que el número de bred vectors es relativamente pequeño.

Importance sampling

Si el objetivo es evaluar la integral por Monte Carlo

$$\mathcal{E}_p(f(\mathbf{x})) \triangleq \int f(\mathbf{x})p(\mathbf{x})\mathrm{d}x$$

pero $p(\mathbf{x})$ es una función complicada de samplear.
Utilizamos una proposal density $q(x)$ como en rejection sampling. En principio es simple generar muestras de esta distribución. Si tenemos N_p muestras de la proposal distribución,

$$q(\mathbf{x}) \doteq \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}-\mathbf{x}^{(j)})$$

La esperanza de p resultante es

$$\mathcal{E}_p(f(\mathbf{x})) = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\mathrm{d}x$$

Reemplazando la densidad propuesta por la aproximación muestral de ésta

$$\mathcal{E}_p(f(\mathbf{x})) \doteq \frac{1}{N_p}\sum_{j=1}^{N_p} f(\mathbf{x}^{(j)})\frac{p(\mathbf{x}^{(j)})}{q(\mathbf{x}^{(j)})}$$

El soporte de la proposal debe abarcar al soporte de la densidad de interés.

Constante de normalización en importance sampling

Entonces tenemos que la razón entre las densidades introduce una suma pesada

$$\mathcal{E}_p(f(\mathbf{x})) \doteq \frac{1}{N_p}\sum_{j=1}^{N_p} \tilde{w}^{(j)} f(\mathbf{x}^{(j)})$$

donde $\tilde{w}^{(j)} = \frac{p(\mathbf{x}^{(j)})}{q(\mathbf{x}^{(j)})}$ son los importance weights.

Si a la distribución la conocemos a menos de una constante,

$$\mathcal{E}_p(f(\mathbf{x})) \doteq \frac{1}{Z_p N_p}\sum_{j=1}^{N_p} \tilde{w}^{(j)} f(\mathbf{x}^{(j)})$$

La densidad esta normalizada

$$1 = \int p(\mathbf{x})\mathrm{d}\mathbf{x} \quad \text{o} \quad Z_p = \int \tilde{p}(\mathbf{x})\mathrm{d}\mathbf{x}$$

por lo tanto a esta integral también podemos evaluarla usando importance sampling

$$Z_p \doteq \frac{1}{N_p}\sum_{j=1}^{N_p} \tilde{w}^{(j)}$$

Entonces tenemos que los pesos de la integral son dados por

$$w^{(j)} = \frac{\tilde{w}^{(j)}}{\sum_{j=1}^{N_p} \tilde{w}^{(j)}}$$

Muestras con pesos

Otra forma de interpretar la esperanza

$$\mathcal{E}_p(f(\mathbf{x})) \doteq \sum_{j=1}^{N_p} w^{(j)} f(\mathbf{x}^{(j)})$$

es que nuestra densidad ha sido representada a traves de muestras pesadas:

$$p(\mathbf{x}) \doteq \sum_{j=1}^{N_p} w^{(j)} \delta(\mathbf{x} - \mathbf{x}^{(j)})$$

entonces el importance sampling puede pensarse directamente como una aproximacion a la densidad que queremos representar y terminamos con una muestra con pesos.

Debilidades en importance sampling

- Estamos realizando la razon de dos estimaciones, por lo que el estimador será sesgado, pero bajo ciertas hipótesis se aplica la ley de los grandes números y hay convergencia cuando $N_p \to \infty$. Se puede obtener un teorema central del límite.
- Necesitamos tener muchas muestras en la region donde $p(x)f(x)$ es grande.

Pero si no tenemos una buena densidad proposal entonces vamos a tener unas pocas o ninguna muestra en la region de valores grandes de p.

- El tamaño efectivo de la muestra son las muestras que efectivamente tienen contribuciones a la integral, la cual puede ser un número menor o nulo.

En el caso de que no haya ningun lo que sucederá es que las varianzas de $\tilde{w}$ seran pequeñas y tb las de $\tilde{w}^{(j)} f(\mathbf{x}^{(j)})$. Sin embargo la esperanza puede ser totalmente errónea.

Perdemos entonces la manera de diagnosticar si estamos teniendo una estimación errónea.

Densidad propuesta óptima

La única garantía que tenemos que el método funcione es encontrar una buena proposal density. Una forma de encontrar una proposal density apropiada es tratar de minimizar la varianza del estimador

$$\mathrm{v}ar_q(x)(f(x)w(x)) = \mathcal{E}_q(f^2(\mathbf{x})w^2(x)) - (\mathcal{E}_p(f))^2$$

La q que minimice la varianza de los pesos sera la óptima.

Sampling importance resampling

Entonces en el importance sampling tenemos un conjunto de muestras $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_p)}\}$ cuyos pesos son $\{w^{(1)}, \cdots, w^{(N_p)}\}$, en general estos pesos van a tener una gran varianza (recordar que suman a 1) y nos gustaria que sea lo mas uniforme posible.
El resampling lo que propone es dada la muestra discreta $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N_p)}\}$ cuyas probabilidades son $\{w^{(1)}, \cdots, w^{(N_p)}\}$, entonces podemos generar muestras de esta distribución discreta. Voy a generar nuevamente N_p muestras de esta probabilidad. Como tiene valores discretos lo que sucederá con la nueva muestra es que las muestras viejas con mucho peso tendran varias copias mientras las con peso casi nulo desapareceran.
El error que cometemos en el muestreo de $p(\mathbf{x})$ con este remuestreo disminuye con $N_p \to \infty$.

MCMC

Estos algoritmos pueden producir muestras de una $p(x)$ solo conociendo $f(x)$, $p(x) = Ff(x)$.
- proponen la densidad que se requiere samplear como la densidad de equilibrio de una cadena de Markov.
Dado que son muestras de una cadena de Markov van a estar autocorrelacionadas.

- Se puede comenzar con un ensamble de puntos iniciales.
- Se pueden tomar puntos separados de la cadena.

Algoritmo Metropolis-Hasting

Queremos muestras de $p(x)$ dado $f(x)$.
Dado un punto initial x_0, en cada iteracion i

1. Se genera un nuevo punto con la probabilidad de transición $x' \sim g(x|x_i)$.
2. Se determina la razón $\alpha = f(x')/f(x_i)$
3. Se genera un numero aleatorio $u \sim [0,1]$.
4. Si $u \leq \alpha$ se acepta $x_{i+1} = x'$
5. Si $u > \alpha$ se rechaza $x_{i+1} = x_i$.

Si la densidad de transicion no es simetrica $\alpha = f(x')/f(x_i)g(x'|x_i)/g(x_i|x')$.
Notar que la razón de aceptación es controlada por la covarianza de la transición. Sin embargo un algoritmo con covarianza conservativa tarda en converger y genera alta correlación.

Algoritmo Langevin - Metropolis

Si consideramos la ecuacion de difusión de Langevin

$$x_k = x_{k-1} + \delta t \nabla \log p(x_k) + \sqrt{2\delta t}\zeta_k$$

donde $\zeta_k \sim \mathcal{N}(0, \mathbf{I})$.
En este caso sabemos que $t_k \to \infty$ cuando $q \to p(x)$.
Pero entonces podemos combinar a algoritmo Langevin con Metropolis.
Usamos las particulas de Langevin como muestras de la densidad de transición. La aceptación estará dada por

$$\alpha = \min\left\{1, \frac{p(x'_k)q(x_{k-1}|x'_k}{p(x_{k-1}q(x'_k|x_{k-1})}\right\}$$

donde

$$q(x'|x) \propto \exp\left(-\frac{1}{4\delta t}\|x' - x - \delta t \nabla \log p(x)\|^2\right)$$

Hamiltonian Monte Carlo

Propuesto por Duane et al 1987 y aplicado/readaptado a muestreo en aprendizaje automatizado por Radford Neal.
Precioso review por: Betancourt, M., 2017. A conceptual introduction to Hamiltonian Monte Carlo. arXiv preprint arXiv:1701.02434.
Usa las ecuaciones de la dinámica Hamiltoniana

$$\frac{dq_j}{dt} = \frac{\partial H}{\partial p_j}$$

$$\frac{dp_j}{dt} = \frac{\partial H}{\partial q_j}$$

Reversible. Preserva el volumen. Conserva energía (H).

Hamiltonian Monte Carlo

La distribución (canónica) de mecanica estadística que nos da la probabilidad de los estados viene dada por

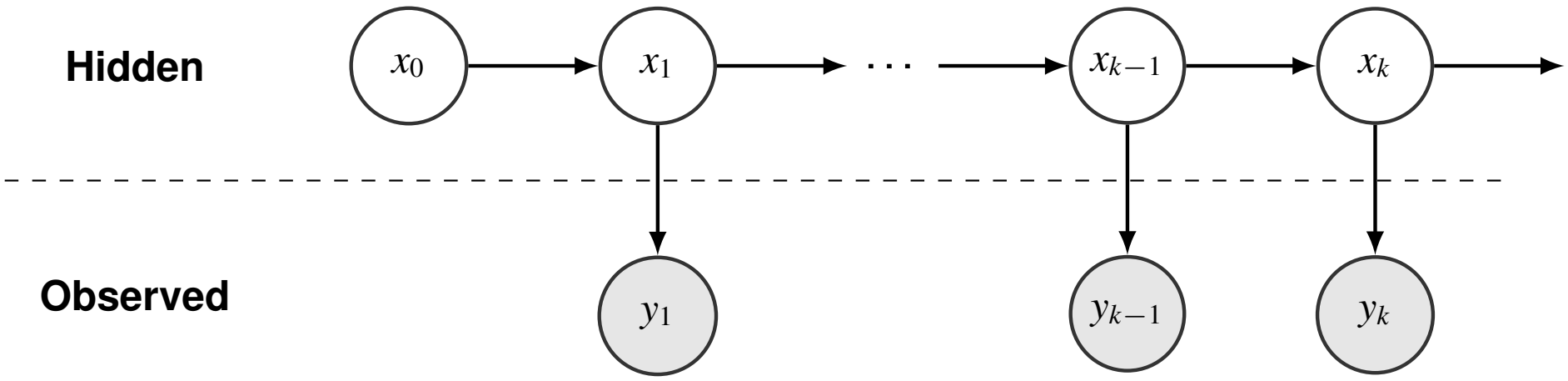
$$p(q,p) = \frac{1}{Z}\exp\left(\frac{-H}{T}\right)$$

T la temperatura del sistema. E Z es la función partición (i.e. la constante de normalización).

La energía potencial viene dada por $U(q) = \log[p_0(q)p(q|y)]$.
Lo usamos como cadena de Markov y combinamos con Metropolis. La razón de aceptación es $\min[1, \exp(-H(q^*,p^*) + H(q,p)]$.
Notar que este esquema evita la caminata aleatoria!.

Secuencial Monte Carlo - Particle filter

El filtro de partículas esta basado en al idea original de Ulam: queremos representar a la densidad posterior secuencial $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ por un conjunto de muestras (particulas) independientes.



Secuencial Monte Carlo - Particle filter

Deberíamos representar dos procesos a través de la muestra:

1. La evolución de la densidad de probabilidad de $k-1$ a k,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\,\mathrm{d}\mathbf{x}_{k-1}$$

2. Asimilación. Se usa la regla de Bayes secuencial:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{F}$$

Podemos usar muestreo de Monte Carlo para representar a las densidades?.
Es estadística totalmente no-paramétrica en alta dimensionalidad (enorme desafio).

Referencias de particle filters

Historica:

- Gordon, N.J., Salmond, D.J. and Smith, A.F., 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In IEE proceedings F, 140, 107-113.

Reviews:

- Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on signal processing, 50(2), pp.174-188.
- Doucet, A. and Johansen, A.M., 2009. A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of nonlinear filtering, 12, 656-704.

Reviews en geociencias:

- Van Leeuwen, P.J., 2009. Particle filtering in geophysical systems. Monthly Weather Review, 137, pp.4089-4114.
- Van Leeuwen, P.J., L Nerger, R Potthast, S Reich, HR Kunsch A review of Particle Filters for Geoscience applications 2019: Q. J. Royal Meteorol. Soc

Naive particle filter

Asumamos que el estado previo $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ es expresado por una muestra—por un conjunto de N_P partículas de estados $\mathbf{x}_{k-1}^{(j)}$. La distribución Monte-Carlo (asumiendo son iid) es dada por

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \doteq \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}_{k-1}-\mathbf{x}_{k-1}^{(j)})$$

Evolucionando a las partículas

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\,\mathrm{d}\mathbf{x}_{k-1} = \frac{1}{N_p}\sum_{j=1}^{N_p}p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)})$$

Para cada particula realizamos un solo muestreo η_{k-1}, obteniendose

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}) &= \int \delta(\mathbf{x}_k-\mathcal{M}(\mathbf{x}_{k-1}^{(j)},\boldsymbol{\eta}_{k-1}))\delta(\boldsymbol{\eta}_{k-1}-\boldsymbol{\eta}_{k-1}^{(j)})\,\mathrm{d}\boldsymbol{\eta}_{k-1} \quad (1)\\ &= \delta(\,\mathbf{x}_k-\mathcal{M}(\mathbf{x}_{k-1}^{(j)},\boldsymbol{\eta}_{k-1}^{(j)})\,) \quad (2)\end{aligned}$$

denotamos $\mathbf{x}_k^{(j)} = \mathcal{M}(\mathbf{x}_{k-1}^{(j)},\boldsymbol{\eta}_{k-1}^{(j)})$.

Naive particle filter

La distribución de k considerando observaciones hasta $k-1$ es dada por

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \frac{1}{N_p}\sum_{j=1}^{N_p}\delta(\mathbf{x}_k - x_k^{(j)})$$

La posterior resultante es entonces

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{1}{N_p}\sum_{j=1}^{N_p}\frac{p(\mathbf{y}_k|\mathbf{x}_k^{(j)})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}\delta(\mathbf{x}_k - \mathbf{x}_k^{(j)})$$

Tenemos un nuevo conjunto de particulas con pesos definidos por

$$w_k^j = \frac{1}{N_p}\frac{p(\mathbf{y}_k|\mathbf{x}_k^{(j)})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} = \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(j)})}{\sum_{j=1}^{N_p} p(\mathbf{y}_k|\mathbf{x}_k^{(j)})}$$

Terminamos con particulas pesadas y habiamos asumido que en el paso anterior todas tenian el mismo peso por lo que hay una incoherencia.

Naive particle filter

Si asumimos que las partículas en $k-1$ eran pesadas, la densidad de pronóstico es

$$p(\mathbf{x}_k|\mathbf{y}_{k-1}) = \sum_{j=1}^{N_p} w_{k-1}^{(j)} \delta(\mathbf{x}_k - x_k^{(j)})$$

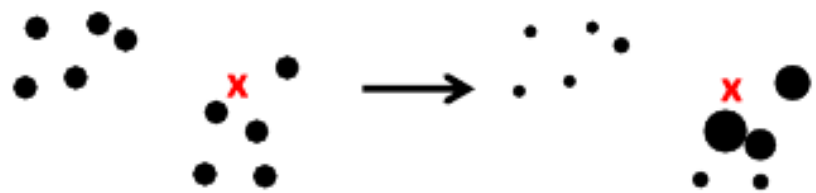
Entonces los nuevos pesos vienen dados por

$$w_k^j = \frac{w_{k-1}^j p(\mathbf{y}_k|\mathbf{x}_k^{(j)})}{\sum_{j=1}^{N_p} w_{k-1}^j p(\mathbf{y}_k|\mathbf{x}_k^{(j)})}$$

Alternativamente consideramos pesos no normalizados

$$\tilde{w}_k^j = w_{k-1}^j p(\mathbf{y}_k|\mathbf{x}_k^{(j)})$$

y luego los normalizamos:
$w_k^j = \tilde{w}_k^j / \sum_{j=1}^{N_p} \tilde{w}_k^j$



Naive particle filter

El gran problema del naive particle filter es que todos los pesos terminaran en una sola partícula (en unas pocas iteraciones).

Dos formas de solucionar esta deficiencia son:

- resampling
- importance sampling

Sequential importance sampling

Filtrado de toda la trayectoria, la posterior es:

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

donde la prior density es $p(\mathbf{x}_{1:K}) = p(\mathbf{x}_1)\prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1})$.
Se deduce la forma secuencial:

$$p(\mathbf{x}_{1:K}|\mathbf{y}_{1:K}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} p(\mathbf{x}_{1:K-1}|\mathbf{y}_{1:K-1})$$

Proponemos que la proposal density sea de la forma:

$$q_K(\mathbf{x}_{1:K}|\mathbf{y}_{1:K}) = q(\mathbf{x}_K|\mathbf{x}_{1:K-1}, \mathbf{y}_{1:K}) q_{K-1}(\mathbf{x}_{1:K-1}|\mathbf{y}_{1:K-1})$$

En ese caso los pesos no-normalizados del paso K son dados por

$$\tilde{w}_K(\mathbf{x}_{1:K}) = \frac{\tilde{p}(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})}{q_K(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})}$$

Pesos con importance sampling

Reescribiendo los pesos en forma secuencial se obtiene

$$\tilde{w}_k^{(j)} = w_{k-1}^{(j)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(j)})p(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)})}{q(\mathbf{x}_K^{(j)}|\mathbf{x}_{1:K-1}^{(j)}, \mathbf{y}_{1:K})}$$

Los pesos luego deben ser renormalizados $w_k^j = \tilde{w}_k^j / \sum_{j=1}^{N_p} \tilde{w}_k^j$.
En el caso del filtro de partículas naive estamos tomando:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:K})) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

No estamos usando las observaciones actuales para construir la proposal

Optimal proposal density

Motivación:

- Si tenemos información de la observaciones en q entonces nos generamos partículas que estan cerca de las observaciones actuales.
- Podemos encontrar la densidad propuesta optima q que nos de una varianza mínima de los pesos? Doucet et al (1998).

Optimal proposal density

Si consideramos los pesos generales de importance sampling:

$$w_k^{*(j)} = \frac{p(\mathbf{x}_{1:k}|^{(j)}\mathbf{y}_{1:k})}{q(\mathbf{x}_{1:k}|^{(j)}\mathbf{y}_{1:k})}$$

Si tomamos que

$$q(\mathbf{x}_{1:k}|\mathbf{y}_{1:k}) = p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k})$$

vamos a tener que

$$\mathcal{E}_q(w^*(x_{1:k})) = 1, \qquad var_q(w^*(x_{1:k})) = 0$$

Para el caso de la posterior tambien vale que

$$p(\mathbf{x}_{1:K}|\mathbf{y}_{1:K}) = p(\mathbf{x}_K|\mathbf{x}_{1:K-1}, \mathbf{y}_{1:K})p(\mathbf{x}_{1:K-1}|\mathbf{y}_{1:K-1})$$

Entonces la densidad propuesta optima secuencial es

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k).$$

De Doucet et al 1998. Aun asi la varianza de los pesos, dadas las observaciones aleatorias, crece con los tiempos k.

Errores Gaussianos

Si asumimos que tenemos errores de modelo y observacionales Gaussianos $\mathcal{N}(0, \mathbf{Q})$ y que $\mathcal{H} = \mathbf{H}$ (lineal).
Entonces la densidad propuesta óptima es fácil de determinar analíticamente:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, \mathbf{y}_k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{x}_{k-1}^{(j)})p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)})}{p(\mathbf{y}_k|\mathbf{x}_{k-1})}$$

Esto es la posterior pero conociendo que la solucion en $k-1$: $\mathbf{x}_{k-1}^{(j)}$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, \mathbf{y}_k) = \mathcal{N}\left[\mathcal{M}(\mathbf{x}_{k-1}^{(j)}) + \mathbf{K}(\mathbf{y}_k - \mathcal{H}\mathcal{M}(\mathbf{x}_{k-1}^{(j)})], (\mathbf{I} - \mathbf{K}\mathbf{H}^\top)\mathbf{Q}\right]$$

Esto es la solucion de OI o KF con $\mathbf{Q}$ como si fuera $\mathbf{P}^b$.
La densidad propuesta óptima es para cada particula la solucion KF.

Resampling

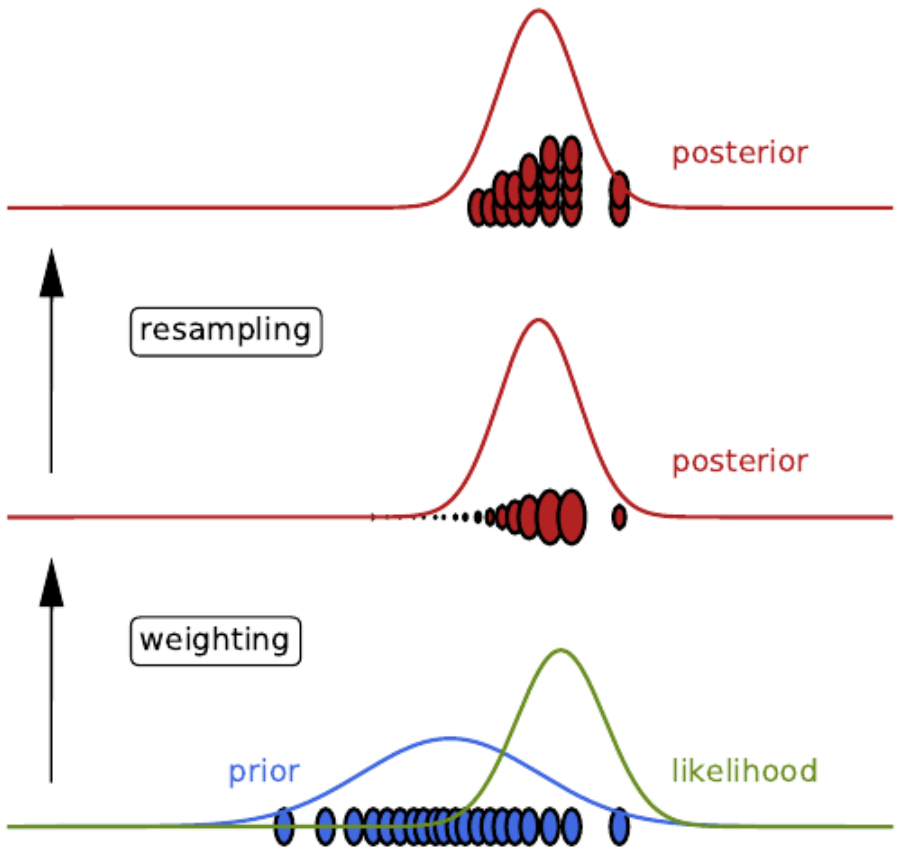
Aun cuando usemos muestreo de importancia el filtro SIS tiene tendencias a degenerarse (una sola particula con todo el peso) después de algun nro de iteraciones.

Generamos un re-muestreo de las partículas (con reemplazo) considerando los pesos.

Se generan copias de las particulas aleatoriamente con una probabilidad dada por los pesos.

De esta forma las particulas con pesos altos tienen mucha dependencia y las con pesos bajos son eliminadas.

Este metodo nos trae aparejado otro problem empobresimiento de la muestra (pérdida de diversidad).



Resampling: Multinomial

La varianza de los pesos crece indefinidamente (Doucet 1998), por lo que es necesar io en algun momento un “resampling”.
Survival: matamos a las particulas debiles.
Idea: Dadas las partículas $\mathbf{x}_k^{1:N_p}$ y los pesos $\mathbf{w}_k^{1:N_p}$ lo que hacemos es generar replicas de las partículas en forma aleatoria de acuerdo al peso. De esta manera la partícula $\mathbf{x}_k^{(j)}$ luego del resampling termina con p robabilidad $\mathbf{w}_k^j$.
Asignamos un orden a las particulas con sus pesos de tal manera que se tenga un intervalo discreto de longitud N_p. Luego generamos N_p números aleatorios de la distribucion discreta $1 : N_p$ con probabilidades $1/N_p$.

Resampling: Multinomial

Generamos $N^{(j)}$ copias de cada partícula $N^{1:N_p}$ una distribución multinomial con parametros $(N, w^{1:N_p})$. La densidad se termina approximando por

$$p(\mathbf{x}_k) \doteq \sum_{j=1}^{N_p} \frac{N^{(j)}}{N_p} \delta(\mathbf{x}_k - \mathbf{x}_k^j)$$

Las partículas con pesos dominantes van a terminar con varias copias mientras las particulas sin peso apreciable mueren.
Luego del resampling las copias de las particulas comienzan con pesos equivalentes $\mathbf{w}_k^{1:N_p} = 1/N_p$.

Número de partículas efectivas: effective sample size (ESS)

La aplicación de resampling genera una fuerte perdida de diversidad de las particulas por lo que es conveniente realizar resampling solo si es necesario.

$$N_{eff} = \frac{1}{\sum_{j=1}^{N_p} (w_k^{(j)})}$$

donde $w_k^{(j)}$ es el peso normalizado de las partículas.
Generalmente se suele poner como condición que si $N_{eff}/N_p < 0.5$ se realice un resampling.
Un buen filtro de partículas debe mantener en todo momento el ESS alto.
Notar que mientras menor es la varianza de los pesos mayor es N_{eff}. En particular si $w_k^{(1:N_p)} = \frac{1}{N_P}$, entonces

$$N_{eff} = N_p$$

Que las partículas tengan pesos equivalentes implica que la ESS sea igual al numero de particulas.

Resampling: Systematic

Existen alternativas al multinomial sampling que permiten disminuir la varianza del multinomial.
Se muestrea de la uniforme $u_1 \sim \mathcal{U}[0, 1/N]$.
Se define $u_i = u_1 + i - 1/N$ con $i = 2, \cdots, N_p$,
Luego se asignan las replicas por

$$N^{(j)} = \{u_j : \sum_{i=1}^{j-1} w_k^i \leq u_i \leq \sum_{i=1}^{j} w_k^i\}$$

Lo unico que estoy eligiendo aleatoriamente es el u_1 en $0, 1/N$ que define cual de las particulas con pesos que no llegan a un multiplo de $1/N$ se le termina haciendo una copia. Todas las particulas con pesos $> 1/N$ van a tener copias, i.e. si una particula tiene peso $3.2/N$ luego tendra 3 copias como minimo (Eventualment 4 dependiendo del u_1).

Algoritmo del SIR filter

Algorithm 1 SIR

Input: Given $\mathbf{x}_{k-1}^{(1:N_p)}$, $\mathbf{y}_k$, $\mathcal{H}(\cdot)$, and $p(\boldsymbol{\eta})$

for $j = 1, N_p$ **do**

$\mathbf{x}_k^{(j)} \leftarrow p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)})$ ▷ En la practica $\mathcal{M}(\mathbf{x}_{k-1}^{(j)} + \eta_k^{(j)})$.

$w_k^{(j)} \leftarrow p(\mathbf{y}_k|\mathbf{x}_k^{(j)})$

end for

$F \leftarrow \sum_j w_k^{(j)}$

for $j = 1, N_p$ **do**

$w_k^{(j)} \leftarrow w_k^{(j)}/F$

end for

if $N_{eff} < N_{umbral}$ **then**

$\mathbf{x}_k^{1:N_p} \leftarrow$ RESAMPLE($\mathbf{x}_k^{1:N_p}$, $w_k^{(1:N_p)}$)

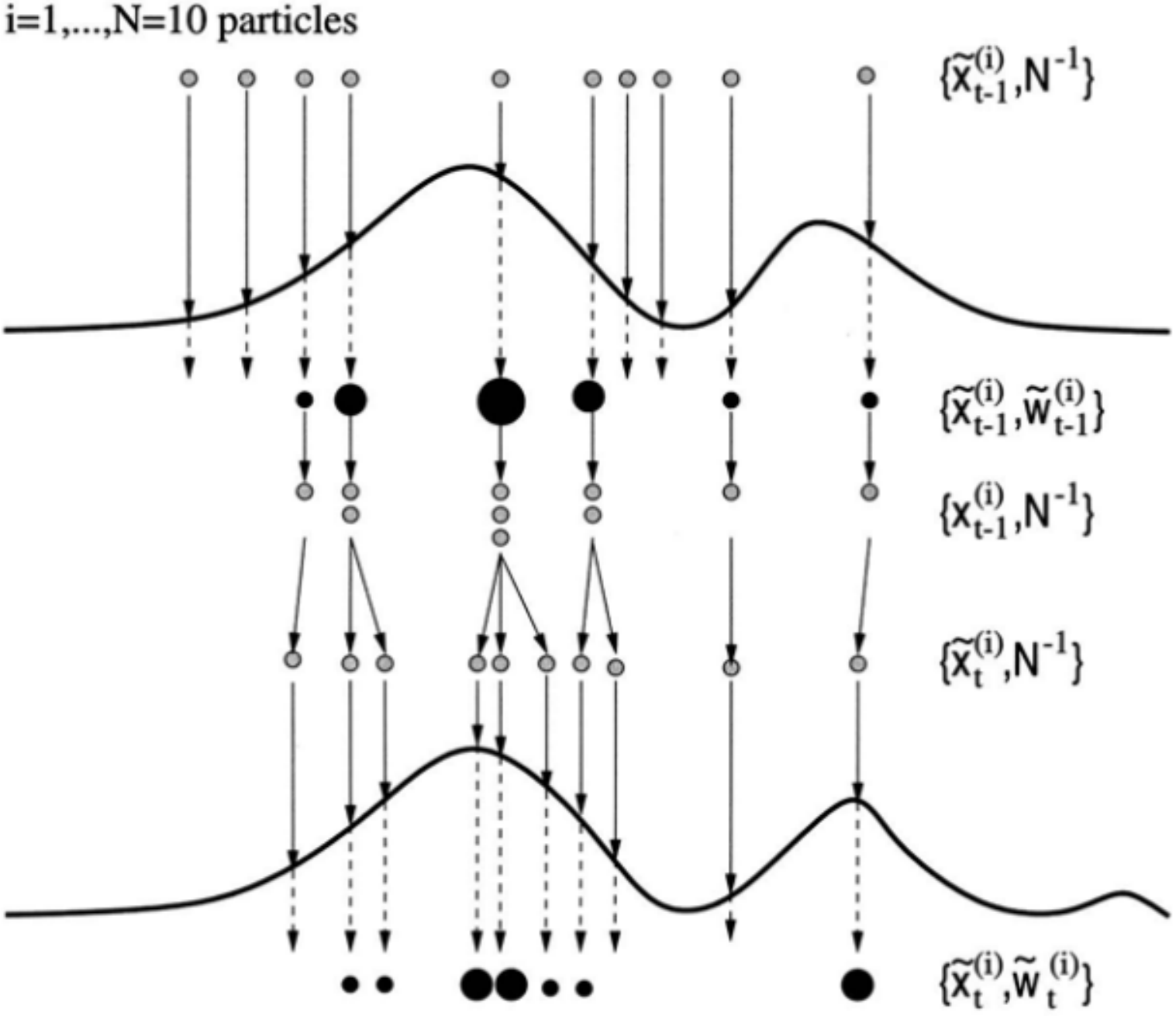
$w_k^{(j)} \leftarrow 1/N_P$

end if

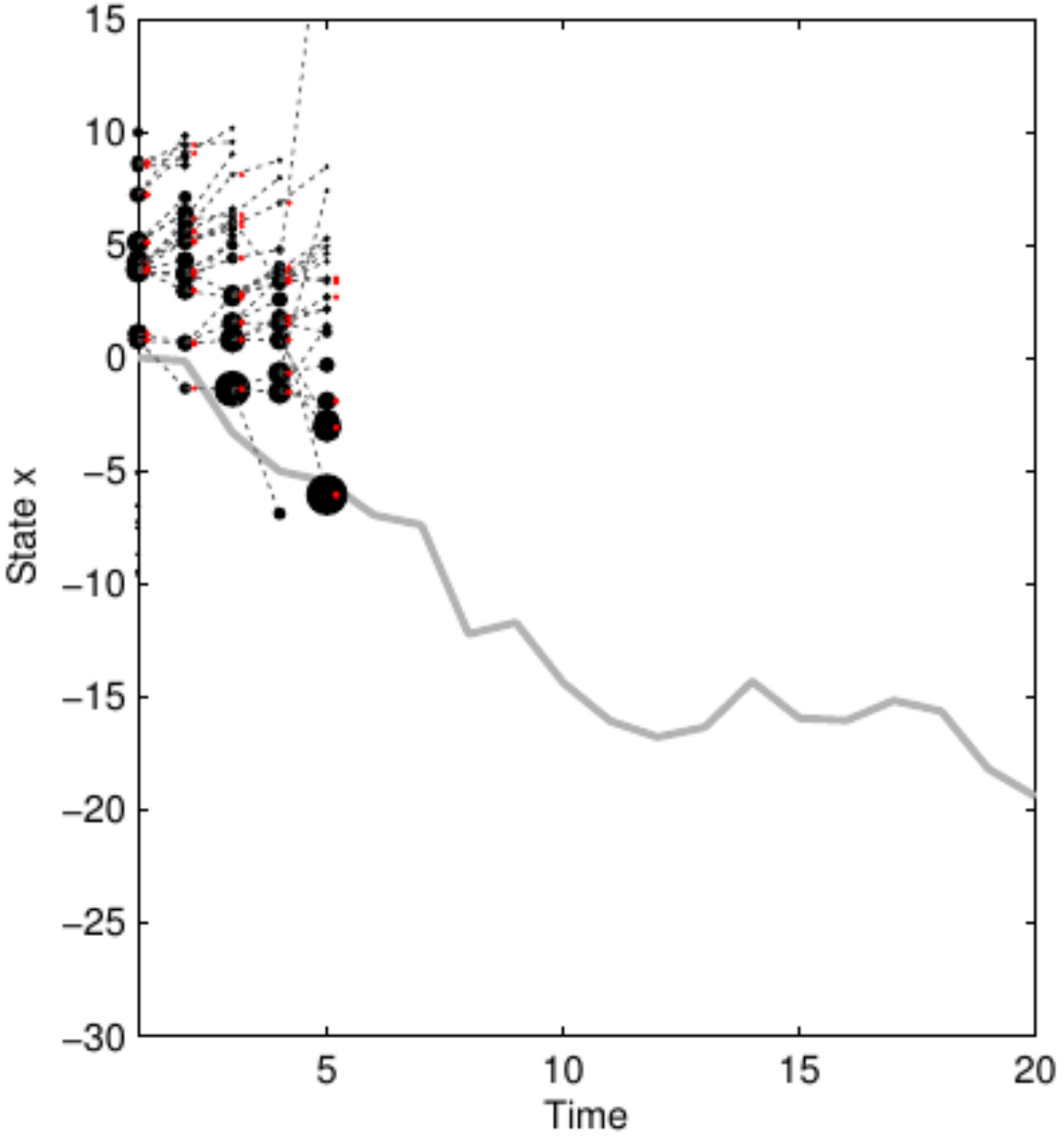
Output: $\mathbf{x}_k^{(1:N_p)}$

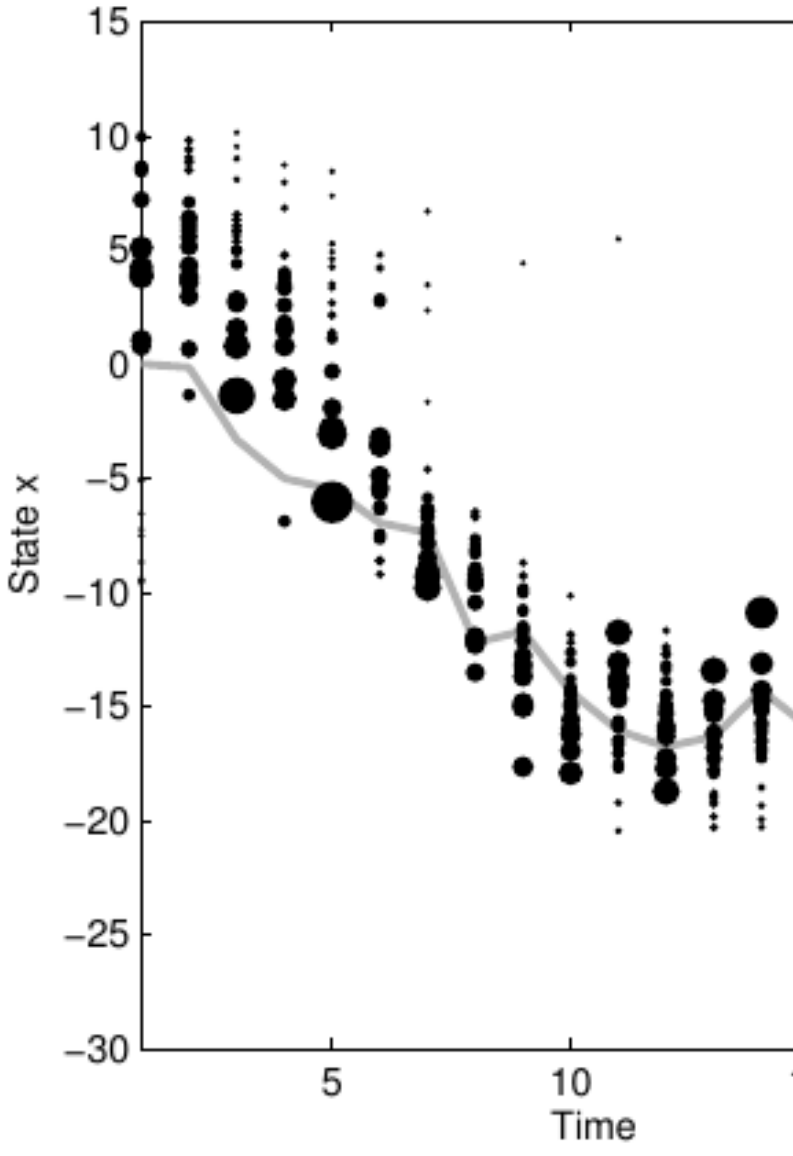
Notar que es un algoritmo de complejidad proporcional a N_p

SIR filter

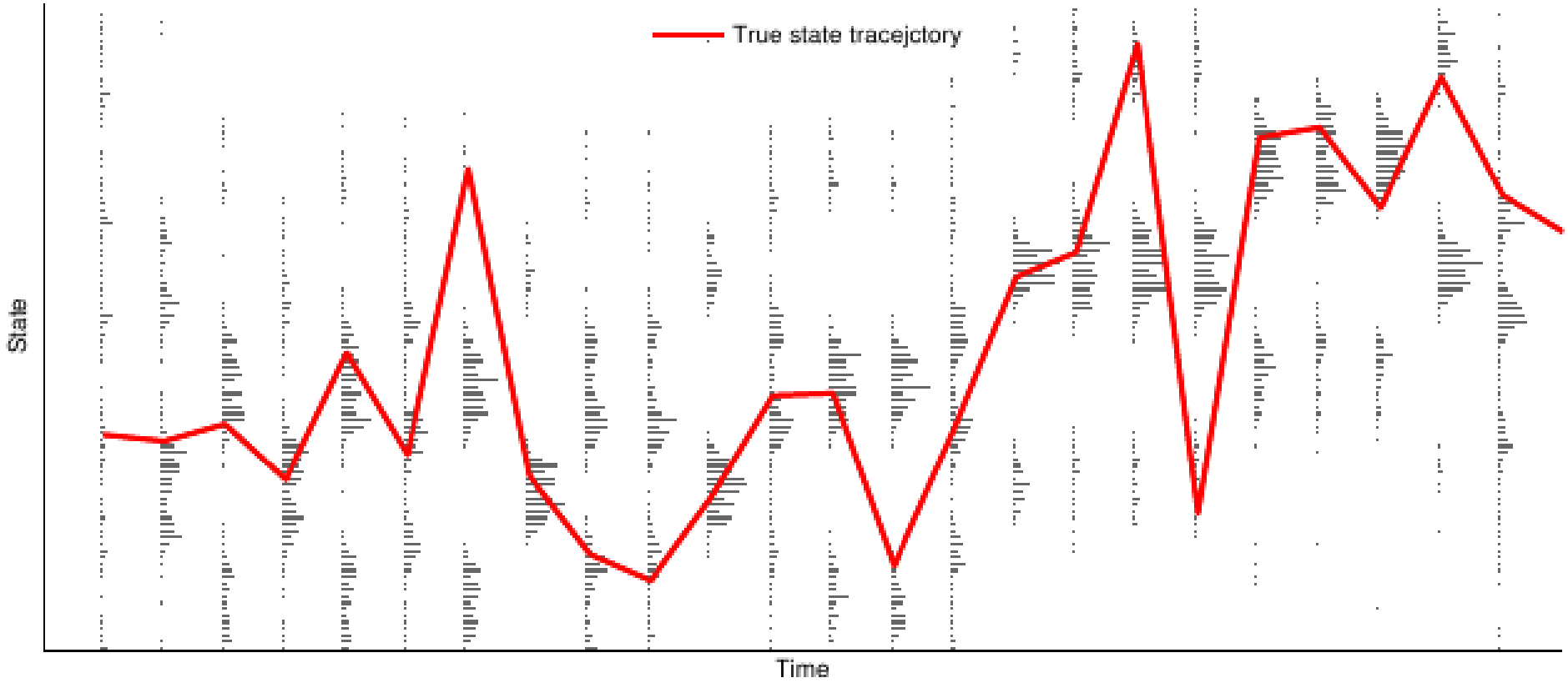

i=1,...,N=10 particles
$\{\tilde{x}_{t-1}^{(i)}, N^{-1}\}$
$\{\tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)}\}$
$\{x_{t-1}^{(i)}, N^{-1}\}$
$\{\tilde{x}_{t}^{(i)}, N^{-1}\}$
$\{\tilde{x}_{t}^{(i)}, \tilde{w}_{t}^{(i)}\}$

Ejemplo


15
10
5
0
−5
−10
−15
−20
−25
−30
State x
5
10
15
20
Time


15
10
5
0
−5
−10
−15
−20
−25
−30
State x
5
10
Time

Ejemplo 2



El filtro de partículas es particularmente adecuado para distribuciones multimodales

Implicit particle filter

Importance sampling pero moviendo a cada particula a las regiones de alta probabilidad.
En los casos que no conocemos analíticamente la densidad propuesta óptima, podemos tomar realizaciones de una Gaussiana y luego transformar las partículas hacia la densidad propuesta óptima.
La idea es entonces:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}) = q(\zeta)J^{-1}$$

donde $q(\zeta) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ y J es el Jacobiano de $\mathbf{x}_k$ a ζ.
La ecuación a resolver para obtener las muestras en $\mathbf{x}_k$ es

$$-\log[p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)})] = F_j(\mathbf{x}_k) = \frac{1}{2}(\boldsymbol{\zeta}^{(j)})^\top \boldsymbol{\zeta}^{(j)} + \phi^{(j)}$$

con $\phi^{(j)} = \min_{\mathbf{x}_k} F_j(\mathbf{x}_k) \propto p(\mathbf{y}_k|\mathbf{x}_{k-1}^{(j)})$.
Sin embargo como hacemos para encontrar el $\mathbf{x}_k$ que satisface la restricción?

Implicit particle filter

Una de las posibilidades es un mapa aleatorio:

$$\mathbf{x}_k^{(j)} = \mathbf{x}^{a(j)} + \lambda^{(j)}(\boldsymbol{\zeta}^{(j)})\mathbf{Q}^{1/2}\boldsymbol{\zeta}^{(j)}$$

donde $\mathbf{x}^{a(j)} = \arg\min F_j(\mathbf{x}_k)$, $\mathbf{Q}$ es el error de modelo.
Entonces hemos transformado a una ecuacion escalar para $\lambda^{(j)}$. Queremos encontrar el salto para cada partícula dada una dirección aleatoria.
Los pesos de las partículas resultantes son:

$$\begin{aligned} w_k^{(j)} &= w_{k-1}^{(j)} \frac{p(\mathbf{y}_k|\mathbf{x}_{k-1}^{(j)})p(\mathbf{x}_k^{(j)}|\mathbf{x}_{k-1}^{(j)}}{p(\mathbf{y}_k)q(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)})} \\ &= w_{k-1}^{(j)} \exp(-\phi^{(j)})J_i \end{aligned}$$

Filtro de Kalman por ensambles como proposal density

Existen varias propuestas de utilizar los miembros del ensamble obtenidos a través de la aproximacion Gaussiana por el EnKF como proposal density para el filtro de partículas.
Frei and Kunsch MWR 2013:

1. Dadas las partículas del forecast, $x_k^{f(j)}$, se realiza un EnKF asumiendo una verosimilitud de $p(\mathbf{y}|\mathbf{x})^{\lambda}$.
2. Luego con las partículas del EnKF, $x_k^{a(j)}$, como proposal del PF, se realiza un pesado considerando $p(\mathbf{y}|\mathbf{x})^{1-\lambda}$ como verosimilitud.
3. Dos problemas: Las copias de las particulas para el EnKF. La adicion estocastica de observaciones perturbadas.

Este tipo de filtro ha sido adaptado para aplicaciones en meteorología, donde se requiere además algun procedimiento de localizacion de las covarianzas (o regresiones). Robert and Kunsch Tellus 2017.
Roberts et al QJ 2018 aplican a un caso de convección. (Fuerte problemas con las discontinuidades en los pesos).