

28.1 Systematic Resampling

Il Systematic Resampling, o ricampionamento a varianza minima, tratta i pesi come variabili casuali continue nell'intervallo $[0, 1]$. Ogni particella viene replicata e il nuovo insieme di campioni scelto in modo da minimizzare la $Var[N_i]$ dove N_i è il numero di repliche della particella i -esima.

Sebbene la fase di resampling riesce a limitare il problema della degenerazione dei pesi essa sfortunatamente introduce una nuova problematica in quanto i campioni che hanno un elevato peso sono statisticamente selezionati molte volte provocando il fatto che l'algoritmo soffra perdita di diversità (*loss of diversity*). Il *multinomial* e il *systematic resampling* differiscono solamente per il metodo di generazione della sequenza di numeri casuali ma risultano molto simili dal punto di vista computazionale. Comunque il systematic resampling migliora la qualità del campionamento.

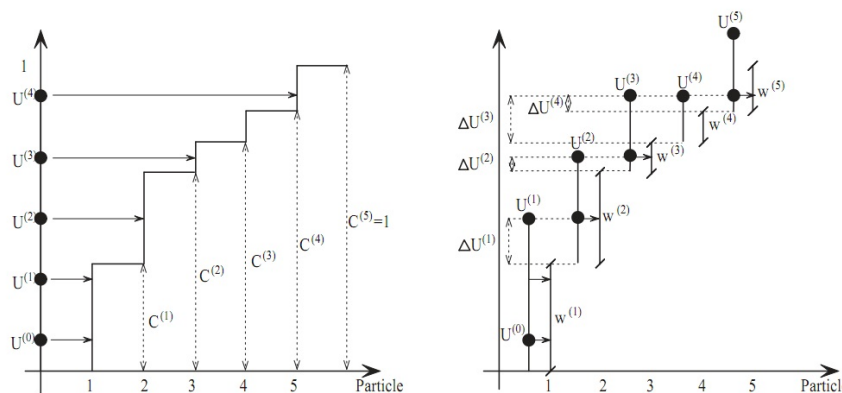


Figura 28.1.

28.2 Filtri particellari

Nelle lezioni precedenti sono stati introdotti i filtri particellari. Questi, come detto, sono utilizzati per trattare sistemi dinamici in cui si presentano d.d.p. “multi-modali”.

Il contesto di applicazione è dunque del tutto generico (sono inclusi anche i modelli altamente non lineari o discontinui) e proprio per questo si riscontra nell'algoritmo implementativo un onere computazionale piuttosto elevato. L'idea di base è l'estremizzazione del concetto di Filtro di Kalman Unscented (U.K.F).

Precisamente, l'obiettivo è quello di approssimare una generica distribuzione di probabilità $p_x(\cdot)$ con tante funzioni $\delta(\cdot)$ ("particelle") centrate nei punti di campionamento ottenuti dalla d.d.p "originale" e tanto più frequenti tanto più ci si avvicina ai campioni con d.d.p maggiore (si veda la prossima figura).

Dati N campioni, ad ogni istante di tempo t si ha un vettore degli stati $x_t \in \mathbb{R}^m$ e si vuole calcolare

$$p(\underbrace{x_t, x_{t-1}, \dots, x_1}_{X_t} | \underbrace{y_t, y_{t-1}, \dots, y_1}_{Y^t})$$

Possiamo riscrivere la p e otteniamo la seguente proporzionalità:

$$p(x_t, x_{t-1}, \dots, x_1 | y_t, y_{t-1}, \dots, y_1) \propto p(y_t | x_t) p(x_t | x_{t-1}) p(X_{t-1} | Y^{t-1}) \quad (28.1)$$

dove sono state sfruttate la formula di Bayes e la markovianità del processo.

Il pseudo codice di un filtro particellare potrebbe essere presentato nel seguente modo:

- **passo 1** Inizializzazione: $x_i \sim \eta(x_i)$ con $i = 1, \dots, N$. I pesi sono dati da

$$\pi_1^i = \frac{p(x_i)}{\eta(x_i^i)}; \bar{\pi}_0^i = \frac{\pi_1^i}{\sum_{i=1}^N \pi_1^i} \quad (28.2)$$

- **passo 2** ci calcoliamo poi ad ogni istante t

$$\tilde{\sigma}_t^i = \frac{p(y_t | x_t^i) p(x_t^i | x_{t-1}^i)}{\eta(x_t^i | x_{t-1}^i, y_t)} \quad (28.3)$$

- **passo 3** peso cumulativo fino a t

$$\pi_t^i = \alpha_t^i \bar{\pi}_{t-1}^i \quad (28.4)$$

- **passo 4** calcolo la normalizzazione del peso fino al tempo t

$$\bar{\pi}_t^i = \frac{\pi_t^i}{\sum_{i=1}^N \pi_t^i} \quad (28.5)$$

- **passo 5** per avere a disposizione una misura che indichi i gradi della degenerazione dei pesi si può utilizzare il numero effettivo delle particelle indicata con N_{eff} .

$$N_{eff} = \frac{N}{1 + Var_q(\pi)} \quad (28.6)$$

Il valore vero di N_{eff} non è in realtà calcolabile quindi si utilizza una sua approssimazione data da:

$$\hat{N}_{eff} = \frac{N^2}{\left(\sum_{i=1}^N \pi^i\right)^2 \cdot \sum_{i=1}^N w_i^2} \simeq \frac{1}{\sum_{i=1}^N w_i^2} \quad (28.7)$$

se N_{eff} scende sotto ad una certa soglia N_t si riutilizza il metodo del ricampionamento (metodo *Sequential Importance Resampling*) per ridurre gli effetti della degenerazione. L'idea di base del ricampionamento è quello di eliminare le particelle che hanno pesi piccoli e di concentrarsi su particelle con pesi più grandi. Se questo non avviene si utilizza il metodo *Sequential Importance Sampling*

- **passo 6** ricalcolo gli stati

$$x_{t+1}^i \sim \eta(x_{t-1}|x_t^i, y_{t+1}) \quad (28.8)$$

e ritorno al passo 2.

L'algoritmo descritto presenta però alcune limitazioni. È importante sottolineare che anche se la distribuzione ottimale $p(x_n|y_n, x_{n-1})$ può essere usata, questo non garantisce che l'algoritmo risulti efficiente. Infatti se la varianza di $p(y_n|x_{n-1})$ è alta allora la varianza dell'approssimazione risultante deve essere alta, di conseguenza deve essere necessario ricampionare più frequentemente.

28.3 Unscented particle filter

Il filtro Unscented Particle Filter (UPF) consiste in un filtro particellare che usa l'Unscented Kalman Filter per generare la "importance proposal distribution". L'UKF permette al filtro particellare di incorporare l'ultima osservazione in una routine di aggiornamento a priori generando la distribuzione a posteriori più verosimile ed è inoltre in grado di generare distribuzioni che hanno maggior significato rispetto al Extended Kalman Filter. Il modello del filtro può essere il seguente

$$\begin{cases} x_{t+1} &= f(x_t, u_t) + w_t \\ y_t &= h(x_t) + v_t \end{cases} \quad (28.9)$$

dove w_t è rumore gaussiano di media nulla e varianza Q , mentre v_t è rumore di misura gaussiano sempre di media nulla e varianza R .

$$p(y_t|x_t) \sim \exp\left(-\frac{1}{2}(y_t - h(x_t))^T R^{-1}(y_t - h(x_t))\right) \quad (28.10)$$

$$p(x_t|x_t) \sim \exp\left(-\frac{1}{2}(x_t - f(x_t, u_t))Q^{-1}(x_t - f(x_t, u_t))\right) \quad (28.11)$$

Descriviamo brevemente l'algoritmo di funzionamento: in $t=0$ si pone l'inizializzazione che consiste nel definire le particelle dalla densità priori $p(x_0)$ e definendo \bar{x}_0 come elemento atteso di x_0 e la matrice varianza P_0 . Per $t \geq 1$ si effettua un campionamento pesato dove ogni particella da $1, \dots, N$ si aggiorna tramite UKF calcolando il passo di aggiornamento e il passo di predizione. Si campionano i nuovi stati e si definiscono le nuove medie e varianze. Per ogni particella si valuta il peso a meno di una costante di normalizzazione

$$\begin{aligned} \sigma_t^i &= \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{\eta(x_t^i|x_{t-1}^i, y_t)} \\ &= \exp\left(-\frac{1}{2}(y_t - h(x_t^i))^T R^{-1}(y_t - h(x_t^i))\right) \exp\left(-\frac{1}{2}(x_t^i - f(x_t^i, u_t))^T Q^{-1}(x_t^i - f(x_t^i, u_t))\right) \end{aligned}$$

Al passo di selezione si moltiplicano le particelle con peso elevato e si eliminano le particelle con peso minore per ottenere ancora N nuove particelle. L'uscita è generata allo stesso modo di un generico filtro particellare.