

粒子フィルタ

データ同化入門

Toshikazu Tayanagi

¹ 和田研

² 小暮研

³ 村井研 (SL)

November 9, 2017

目次 I

- 1. 目次
- 2. データ同化とは
- 3. 粒子フィルタに入る前に
 - 3.1 状態空間モデル
 - 3.2 カルマンフィルタ
 - 3.3 粒子フィルタとは
- 4. 粒子フィルタ
 - 4.1 粒子フィルタのメリット・デメリット
 - 4.2 近似
 - 4.3 アンサンブル近似
 - 4.4 一期先予測
 - 4.5 フィルタリング
- 5. 実装
- 6. 補題

データ同化とは

シュミレーションとデータ分析を融合させたもの。

- ▶ シュミレーション

初期値が正しければ正確な結果が帰ってくる。初期値さえ正しければ予測も高精度。演繹的。

- ▶ データ分析

内挿的なものは得意だが外挿は苦手。手持ちのデータがこうだからデータが取得できていない状況はこうだろうという帰納的な考え方。



- ▶ データ同化

シュミレーションしたものをデータが取得されるたびに逐次的に修正する。

状態空間モデル

粒子フィルタに入る前に僕が愛してやまない状態空間モデルについて。

$$\alpha_t = F\alpha_{t-1} + \varepsilon_t$$

$$y_t = H\alpha_t + Wz_t + v_t$$

(1) 式が状態方程式、(2) 式が観測方程式。

y は目的変数、 z は外性変数、 α は潜在変数、 ε と v は誤差項。

α を推定する手法としてオンラインならカルマンフィルタ、オフラインなら MCMC などが有る。

カルマンフィルタ

線形ガウス状態空間モデルの推定手法の一つ。

もともとは制御系の技術。

アポロはカルマンフィルタとファミコン以下の計算能力とお金の力で月に行った。らしい。

詳しいアルゴリズムは今回はパスで。興味ある人は過去の main チャンネルをあさってください。

粒子フィルタとは

非線形非ガウス状態空間モデルの推定手法の一つ。

データ同化の手法の一つで主に地震や津波、気象などの分析に使われることが多いと思われる。

アルゴリズム的には状態 x_t の分布を粒子 $x_t^{(i)}$ とその点に対応する尤度 $w_t^{(i)}$ の集合で近似するもの。

粒子フィルタのメリット・デメリット

▶ メリット

1. 非線形でも推定可能
2. 非線形なモデルを微分などせず、発生させた乱数を代入するだけでなので数値計算が楽
3. アルゴリズム自体はそんなに複雑じゃない

▶ デメリット

1. 次元が大きくなると発生させる乱数が指数的に増える
2. SIR の際に更新がうまく行かなくなる場合がある
3. 粒子数によっても結果が変わる

カルマンフィルタと粒子フィルタの違い

シュミレーション以外にも大きな違いが一つある。

一期先予測からフィルタリング推定量に更新する際の挙動である。

- ▶ カルマンフィルタ

一期先予測に観測値 y_t と観測値の一期先予測 $y_{t|t-1}$ の残差にカルマンゲインと呼ばれる重みをかけたものを足すことで更新する。

- ▶ 粒子フィルタ

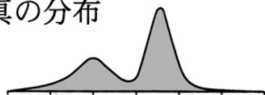
アンサンブルの尤度を計算し、尤度の比率で重複を許して再サンプリングをおこなう (SIR)

カルマンフィルタは全体を動かすイメージ、粒子フィルタは尤度が小さいものを切り捨てること変形させるイメージ。

近似

複雑な分布は解析的に求められないので近似することで数値計算できるようにする。

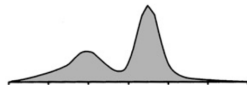
真の分布



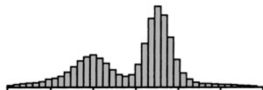
ガウス分布近似



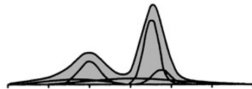
線形(一次)スプライン近似



階段関数近似



ガウス和近似



モンテカルロ(粒子)近似



2005 電子情報通信学会誌 樋口知之『粒子フィルタ』

アンサンブル近似 (モンテカルロ近似)

アンサンブル近似ではある確率変数 x の確率分布を N 個のサンプル集合 $\{x^{(i)}\}$ を用いて以下の様に近似する.

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

x が $p(x)$ に従うとき以下のように近似できる.

$$\begin{aligned} E[f(x)] &= \int f(x)p(x)dx \\ &\doteq \int f(x) \left(\frac{1}{N} \sum_{i=1}^N \delta(x - x^{(i)}) \right) dx \\ &= \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \end{aligned}$$

粒子フィルタの流れ

[一期先予測]

フィルタリングアンサンブル $x_{t-1|t-1}^{(i)}$ と $\varepsilon_t^{(i)} \sim p(\varepsilon_t)$ に従うシステムノイズから予測アンサンブル $x_{t|t-1}^{(i)}$ を得る。



[フィルタリング]

フィルタリングを行い一期先予測に重み付けする。



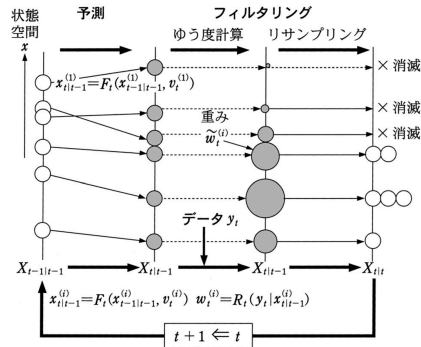
[リサンプリング]

尤度の比率にしたがって SIR(Sampling importance resampling) を行い $x_{t|t}^{(i)}$ を得る。



$t-1 = t$ にして一期先予測に戻る

粒子フィルタの流れ(イメージ図)



2005 電子情報通信学会誌 樋口知之『粒子フィルタ』

一期先予測

$x_{t-1|t-1}$ が既知のとき以下のようになる

$$x_{t|t-1}^{(i)} = f_t(x_{t-1|t-1}^{(i)}, v_t^{(i)}), \quad v_t^{(i)} \sim q_t(v_t)$$

$$\begin{aligned} p(x_t | y_{1:t-1}) &= \int p(x_t | x_{t-1}, y_{1:t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int p(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int \left\{ \int p(x_t, v_t | x_{t-1}) dv_t \right\} p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int \left\{ \int p(x_t | x_{t-1}, v_t) p(v_t | x_{t-1}) dv_t \right\} p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int \int p(x_t | x_{t-1}, v_t) p(v_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1} dv_t \end{aligned}$$

一期先予測

$$\begin{aligned}
 p(x_t|y_{1:t-1}) &\doteq \frac{1}{N} \sum_{i=1}^N \int \int p(x_t|x_{t-1}, v_t) \delta \left(\begin{pmatrix} x_{t-1} \\ v_t \end{pmatrix} - \begin{pmatrix} x_{t-1}^{(i)} \\ v_t^{(i)} \end{pmatrix} \right) dx_{t-1} dv_t \\
 &= \frac{1}{N} \sum_{i=1}^N p \left(x_t | x_{t-1} = x_{t-1}^{(i)}, v_t = v_t^{(i)} \right)
 \end{aligned}$$

x_{t-1} と v_t を与えると x_t は一点に決まるので、

$$\begin{aligned}
 p(x_t|y_{1:t-1}) &\doteq \frac{1}{N} \sum_{i=1}^N \delta \left(x_t - f_t(x_{t-1}^{(i)}, v_t^{(i)}) \right) \\
 &= \frac{1}{N} \sum_{i=1}^N \delta(x_t - x_{t|t-1}^{(i)})
 \end{aligned}$$

フィルタリング

$$\begin{aligned} p(x_t | y_{1:t}) &= \frac{p(y_t | x_t) p(x_t | y_{1:t-1})}{\int p(y_t | x_t) p(x_t | y_{1:t-1}) dx_t} \\ &\doteq \frac{p(y_t | x_t) \frac{1}{N} \sum_i^N \delta(x_t - x_{t|t-1}^{(i)})}{\int p(y_t | x_t) \frac{1}{N} \sum_j^N \delta(x_t - x_{t|t-1}^{(j)}) dx_t} \\ &= \frac{1}{\sum_{j=1}^N p(y_t | x_{t|t-1}^{(j)})} \sum_{i=1}^N p(y_t | x_{t|t-1}^{(i)}) \delta(x_t - x_{t|t-1}^{(i)}) \\ &= \frac{1}{\sum_{j=1}^N \lambda_t^{(j)}} \sum_{i=1}^N \lambda_t^{(i)} \delta(x_t - x_{t|t-1}^{(i)}) \end{aligned}$$

フィルタリング(つづき)

$$\begin{aligned} p(x_t | y_{1:t}) &= \sum_{i=1}^N \frac{1}{\sum_{j=1}^N \lambda_t^{(j)}} \lambda_t^{(i)} \delta(x_t - x_{t|t-1}^{(i)}) \\ &= \sum_{i=1}^N \beta_t^{(i)} \delta(x_t - x_{t|t-1}^{(i)}) \end{aligned}$$

β の確率で重複を許してリサンプリングする (SIR)

実装してみた

モデルは以下である。

$$\alpha_t = \begin{pmatrix} \alpha_{0,t} \\ \alpha_{1,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{0,t-1} \\ \alpha_{1,t-1} \end{pmatrix} + \nu_t \quad \text{[状態方程式]}$$

$$y_t = \alpha_t + \omega_t \quad \text{[観測方程式]}$$

$$\nu_t \sim N\left(0, \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}\right)$$

$$\omega \sim N(0, \sigma^2)$$

粒子数はそれぞれ 10,000。初期値は

$$a0_{0|0}^{(i)} \sim N(7, 1)$$

$$a1_{0|0}^{(i)} \sim N(0, 0.1)$$

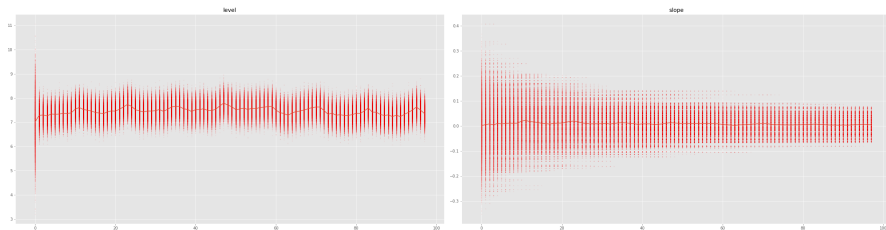
頑張って実装した

線形ガウスでトレンドとレベルを持つモデル。パラメータは RMSE が最小になるものを選んだ。つまり。

推定結果は $\hat{R} = 0, \hat{\sigma}^2 = 7.1$

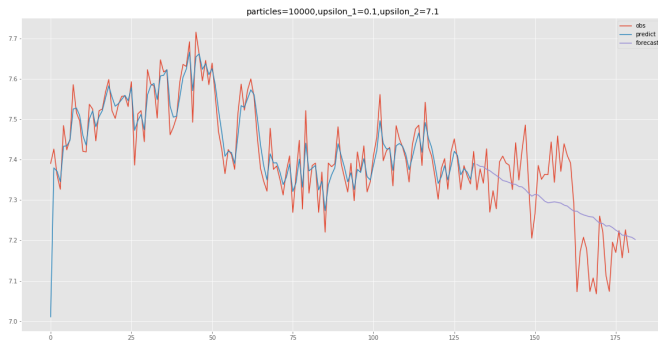
頑張って実装した

レベルとトレンドの粒子と推定値



頑張って実装した

観測値と推定と予測。



補題

デルタ関数

$$\delta(x) = \begin{cases} +\infty (x = 0) \\ 0 (x \neq 0) \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

$$\int_{-\infty}^{\infty} f(x) \delta(x) dx = f(0)$$

$$\int_{-\infty}^{\infty} f(x) \delta(x - a) dx = f(a)$$