Brief Introduction to Probability and Simulation: Part 3 — Bootstrapping and Particle Filter

Elaine Spiller

Marquette University

Data Assimilation Research Programme: Monsoon Summer School

July 11, 2011

Quality of an estimator

Setup:

Suppose $X_1, \ldots, X_n$ are iid random variables where $X_n \sim f$.

Suppose also that we are interested in some parameter of the distribution — could be mean, variance, any other parameter.

Say we have an estimator of $\theta(f)$, $(\bar{x}, s^2, \ldots)$, called $g(X_1, \ldots, X_N)$.

Goal:

We'd like to know how *good* of an estimator g is, so we'll consider the mean square error (MSE), defined as

$$MSE(f) \equiv E_f[(g(X_1, ..., X_N) - \theta(f))^2]$$

Problems

Problem:

The MSE is *also* a parameter of f, but for a generic parameter $\theta(f)$ it is not clear how to estimate it.

To do so, we'll use the idea of a *bootstrap*

Another Problem:

We don't know f, we don't know θ.

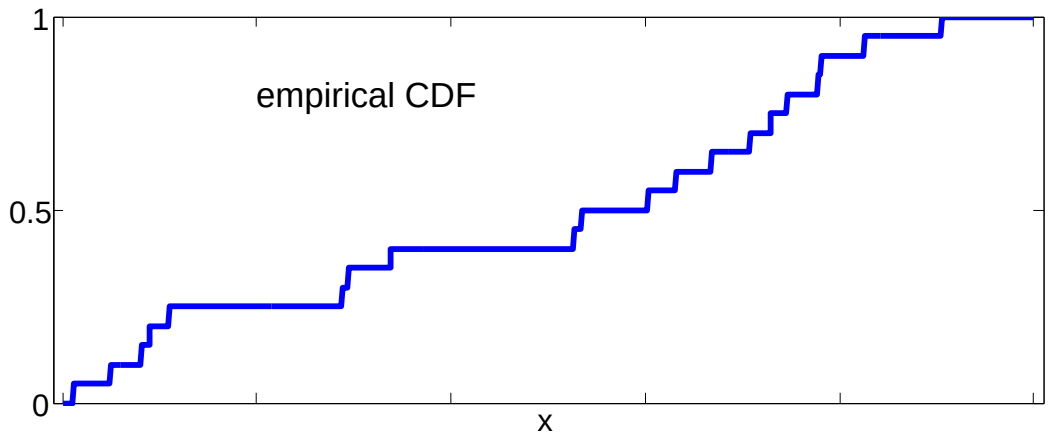
We're going to approximate θ from an empirical distribution, i.e., a distribution constructed from our sampling (or data) and we'd like to know *how good* this approximation is. Hmmm.

In other words, to estimate the sampling distribution of a statistic, we can't sample from the underlying distribution (because we don't know it!).

So instead we sample from an estimate of it — the observed data (e.g. the sample $X_1, \ldots, X_N$).

If we have N samples we can calculate the *empirical cumulative density function*,

$$F_e(x) = \frac{\text{number of } i \ : \ X_i \leq x}{N}.$$



Sampling an empirical distribution

Note that our samples $X_1, \ldots, X_N$ *are* the (discrete) state space f_e (the pmf associated with F_e), i.e. $x = \{X_1, \ldots, X_N\}$.

Since the elements of the state space were random samples from f, we assign them equal probability.

Now to sample from f_e

- Let w be a uniform rv on $(0, 1)$
- Let $I = \lceil Nw \rceil$
- Let $Z = x_I$
- Repeat

Note, this amounts to sampling *with replacement*.

If N is large, then f_e is "close" to f, then $\theta(f_e)$ will be close to $\theta(f)$ and

$$MSE(f) \approx MSE(f_e) = E_{f_e}[(g(Z_1, \ldots, Z_N) - \theta(f_e))^2]$$

How can we find this in practice?

Use all of the samples to estimate θ. That is,

$$\theta(f_e) = g(X_1, \ldots, X_N) \qquad \text{and} \qquad Z_n \sim f_e.$$

To compute $E_{f_e}[(g(Z_1, \ldots, Z_N) - \theta(f_e))^2]$ we need to sum over *all combinations* of $x_i's$. So,

$$MSE(f_e) = \sum_{i_N} \cdots \sum_{i_1} \frac{[f(x_{i_1}, \ldots, x_{i_n}) - \theta]^2}{N^N}$$

The bootstrap method estimates this expectation by taking the sample mean of $MSE(f_e) = E_{f_e}[(g(Z_1, \ldots, Z_N) - \theta(f_e))^2]$. So, we'll compute

$$\begin{aligned} Y_1 &= [g(Z_1^1, \ldots, Z_N^1) - \theta]^2 \\ Y_2 &= [g(Z_1^2, \ldots, Z_N^2) - \theta]^2 \\ Y_3 &= \ldots \\ &\vdots \qquad \text{do this } M \text{ times} \end{aligned}$$

Then, we have

$$MSE(f_e) \approx \frac{1}{M} \sum_{i=1}^{M} Y_i$$

Example

Let W_i be the time the i^{th} customer spends waiting. Say we're interested in the long-run average.

$$\theta \equiv \lim_{n\to\infty} \frac{W_1 + W_2 + \ldots + W_n}{n}$$

We'll rewrite this expression by letting N_i be the number of customers that arrive on the i^{th} data, with the wait on day i is given by

$$\begin{aligned} D_1 &= W_1 + \ldots W_{N_1} \\ D_2 &= W_{N_1+1} + \ldots + W_{N_1+N_2} \\ &\vdots \\ \theta &= \lim_{m\to\infty} \frac{(D_1 + D_2 + \ldots D_m)/m}{(N_1 + N_2 + \ldots + N_m)/m} = \frac{E[D]}{E[N]} \end{aligned}$$

If we have k days of data, we can estimate $E[D]$ and $E[N]$

$$\bar{D} = \frac{D_1 + D_2 + \ldots + D_k}{k} \qquad \text{and} \qquad \bar{N} = \frac{N_1 + N_2 + \ldots + N_k}{k}$$

Recall $\theta = E[D]/E[N]$, so our estimate of θ is

$$g(\mathbf{D}, \mathbf{N}) = \frac{\bar{D}}{\bar{N}} = \frac{D_1 + D_2 + \ldots + D_k}{N_1 + N_2 + \ldots + N_k} \qquad D_i, N_i \sim f$$

And thus we have,

$$MSE(f) = E\left[\left(\frac{\sum_{i=1}^{k} D_i}{\sum_{i=1}^{k} N_i} - \theta\right)^2\right]$$

Sampling f_e

Recall, the samples of f, $\{D_i, N_i\}$ become the state space for f_e, $\{d_i, n_i\}$ for $i = 1, \ldots, k$. And recall,

$$P_{f_e}\{D = d_i, N = n_i\} = \frac{1}{k} \qquad \text{so, we have}$$

$$E_{f_e}[D] = \bar{d} = \sum_{i=1}^{k} \frac{d_i}{k} \qquad \text{and} \qquad E_{f_e}[N] = \bar{n} = \sum_{i=1}^{k} \frac{n_i}{k}.$$

And thus $\theta(f_e) = \bar{d}/\bar{n}$ which gives us

$$MSE(f_e) = E_{f_e}\left[\left(\frac{\sum_{i=1}^{k} D_i}{\sum_{i=1}^{k} N_i} - \frac{\bar{d}}{\bar{n}}\right)^2\right]$$

Probability/simulation tools needed for Particle Filter

- Basic ideas
  - probability distributions, state space, sampling random variables, Bayes theorem
- Importance sampling
  - seek a distribution that "looks like" the one of interest
  - first type discussed, expectation or modes
- Bootstrap
  - sampling an empirical distribution
- Markov property (j is a time index)

$$p(x_{j+1}|x_j, x_{j-1}, \ldots, x_0) = p(x_{j+1}|x_j)$$

shorthand notation $x_{0:j} = \{x_j, x_{j-1}, \ldots, x_0\}$

Markov transitions via model

We're interested in the probability of a state X_j as it evolves over time. Recall, for independent random variables, we have

$$p(x_{0:n}) = p(x_0) \prod_{j=1}^{n} p(x_j | x_{1:j-1})$$

For our case, we'll have some distribution of initial conditions $\mu(x_0)$ (background) and a model to move our state forward in time,

$$X_j | (X_{j-1} = x_{j-1}) \sim m(x_j | x_{j-1})$$

where $m(x_j | x_{j-1})$ is the *transition probability* or the probability that our model would take use from state x_{j-1} to state x_j.

Combining the ideas above gives us

$$p(x_{0:n}) = \mu(x_0) \prod_{j=1}^{n} m(x_j | x_{j-1})$$

Observations and likelihood

Recall, our observations will be related to the state variable by some observation function $y = h(x)$. We can think of observations as random variables distributed as

$$Y_j|(X_j = x_j) \sim g(y|x_j).$$

Or, $Y_j = h(X_j)+$"*noise*".

We call this distribution the *likelihood* — how likely was an observation given the possible states?

With a whole set of observations $\{Y_j\}$ we can write down the likelihood for the time-series of observations

$$p(y_{1:j}|x_{1:j}) = \prod_{j=1}^{n} g(y_k|x_k)$$

Inference: goal for data assimilation

Given a background distribution of initial conditions, $\mu(x_0)$, and observations, $Y_{1:n}$, we want to infer the distribution of physical states $X_{0:n}$.

- Prior

$$p(x_{0:n}) = \mu(x_o) \prod_{j=1}^{n} m(x_j|x_{j-1})$$

- Likelihood

$$p(y_{1:n}|x_{1:n}) = \prod_{j=1}^{n} g(y = h(x_j)|x_j)$$

- Posterior, obtained by Bayes' rule

$$p(x_{1:n}|y_{1:n}) = \frac{p(y_{1:n}|x_{1:n})p(x_{0:n})}{p(y_{1:n})}$$

recall, $p(y_{1:n}) = \int p(y_{1:n}|x_{1:n})p(x_{0:n})dx_{1:n}$

A Monte Carlo simulation or really sampling $p(x_{1:n}|y_{1:n})$

- takes a discrete set of samples from $X_0 \sim p(x_0)$
- moves them forward accord to the model, e.g. samples $X_{0:j} \sim p(x_j|x_{0:j-1})$
- evaluates likelihood between samples and observations

Note, after a few (say $k = 2$ or 3 observations) you will have samples from $X_{0:k} \sim p(x_{0:k}|y_{1:k})$ but they will not be useful.

Sequential Monte Carlo with Importance Sampling (SIS)

Idea — normalize at every step, treat that posterior distribution as an *importance* prior distribution for the next step.

1. Start with $X_o \sim p(x_o)$, each particle $X_o^{(k)}$ has weight $w_1^{(k)} = 1/N$
2. Transition each $X_0^{(k)}$ forward, this gives sample $X_1^{(k)} \sim p(x_1|x_0) = m(x_1|x_0)$
3. Evaluate the likelihood function of each sample ("particle") $X_1^{(k)}$ against Y_1, $g(Y_1|X_1^{(k)})$
4. *Weight* each particle by

$$w_1^{(k)} = \frac{g(Y_1|X_1^{(k)})w_0^{(k)}}{\sum_{k=1}^{N} g(Y_1|X_1^{(k)})w_0^{(k)}}$$

Repeat process transition from X_{j-1} to X_j instead of 0 to 1.

$\tilde{\pi}(x_j|Y_{1:j}) = \{x_j = X_j^{(k)}, w^{(k)}\}$

Problem with SIS and Solution: Resampling

With a large number of samples, SIS works pretty well on moderate (small) dimensional deterministic (perfect model) problems.

Problem:

- A significant problem, though, is that most (or all) of the weight can be taken over by *one particle*

Solution:

- Resampling, e.g., bootstrapping

Strategy:

- Monitor weights, if problematic
- *Resample* or "bootstrap" by treating $\tilde{\pi}_j(x_{0:k}|Y_{1:k})$ as an *importance* empirical distribution
- Set all weights to $w_j^{(k)} = 1/N$
- Transition $j+1$ step, repeating resampling as necessary

The strategy is referred to an *SIR (sequential importance resampling) filter* and also goes by the names *particle filter*, *bootstrap filter*, and *sequential Monte Carlo*.

(Note, if model is deterministic, need some strategy to sample "around" each $X_j^{(k)}$)

Pros and Cons

Pros:

- No linearization, naturally handles nonlinear model and nonlinear observation operator
- No moving state locations in analysis step
- Relatively easy to implement

Cons

- Doesn't work well in large dimensions
- Requires large number of samples
- Lose information on "initial condition problem" when resampling employed

References

General information on particle filters:

- search “A tutorial on particle filtering and smoothing: Fifteen years later” by Doucet and Johansen
- http://www.cs.ubc.ca/∼arnaud/journalsbysubject.html

Particle filters for data assimilation in Geosciences

- “Particle filtering in geophysical systems” (number on 9 on the list) is a review article

  http://www.met.reading.ac.uk/∼xv901096/research/publications.html