Lecture: Advanced Bayesian Computing

Importance Sampling and Sequential Monte Carlo

Accept - Reject Sampling

In many scenarios, sampling from a function $g(x)$ can be challenging (in that we cannot use `rg.function()` to sample from it. The general idea of accept-reject sampling is to simulation observations from another distribution $f(x)$ and accept the response if it falls under the distribution $g(x)$.

Formally the algorithm for the Accept-Reject Method follows as:

1. Generate $X \sim f, U \sim Unif[0, 1]$
2. Accept $Y = X$ if $U \leq g(x)/Mf(x)$,

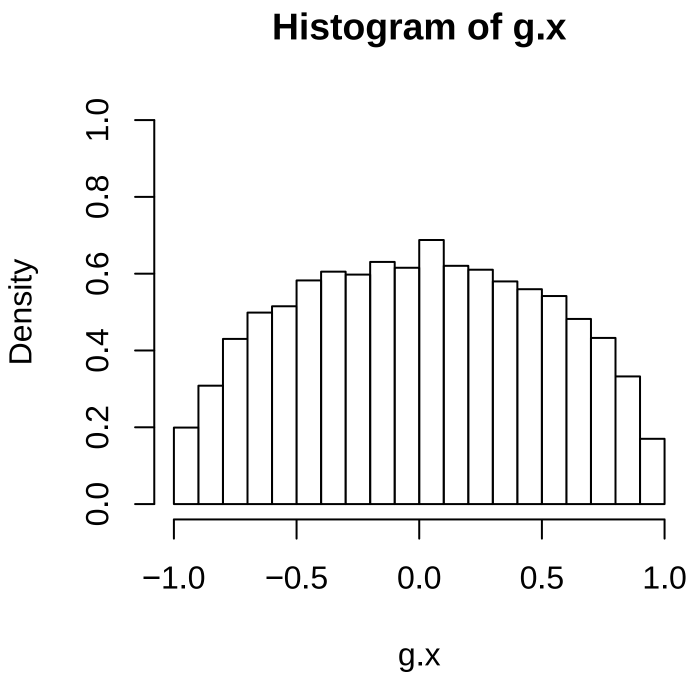
where $f(x)$ and $g(x)$ are normalized probability distributions and M is a constant ≥ 1.

Suppose we want to draw samples from a normalized form of $g(x) = \frac{\sqrt{1-x^2}}{\pi/2}$. Then $f(x) = \frac{1}{2}$ for $x \in [-1, 1]$ and $M = \frac{4}{\pi}$.

```r
num.sims <- 10000
x <- runif(num.sims,-1,1) # simulate samples
u.vals <- runif(num.sims)
M <- 4/pi
f.x <- 1/2

accept.ratio <- (sqrt(1-x^2) / (pi/2)) / (f.x*M) # g/fM
accept.index <- (1:num.sims)[u.vals < accept.ratio]
g.x = x[accept.index]

hist(g.x, prob = T, ylim = c(0,1), breaks = 'FD')
```



Now we have samples from $g(x)$ and could, for example, compute $I = \int xg(x)dx$

Without the use of packages such as `rtnorm()` how would you draw a sample from a truncated normal distribution?

One possibility is to use accept-reject sampling. Simulate points from a normal distribution with the same mean and variance. Let M $= \frac{1}{\Phi(c;\mu,\sigma^2)}$, where $\Phi(.;\mu,\sigma^2)$ is the cdf function a normal random variable and c is the truncation point. Then all values of x greater than c are accepted and all values less than c are rejected.

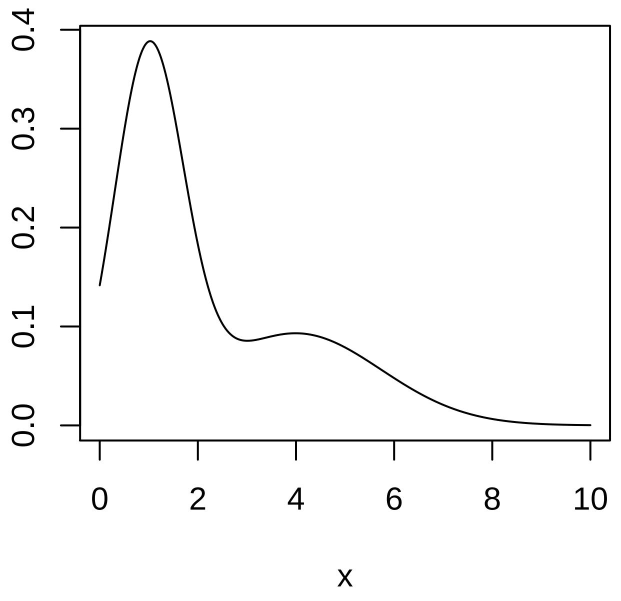
Importance Sampling

Importance sampling is related to the idea of accept-reject sampling, but emphasizes focusing on the "important" parts of the distribution and uses all of the simulations. In accept-reject sampling, computing the value M can be challenging in its own right. We can alleviate that problem with importance sampling.

Again the goal is to compute some integral, say $I = \int h(x)g(x)dx = E[h(x)]$. We cannot sample directly from $g(x)$ but we can evaluate the function. So the idea is to find a distribution that we can simulate observations from, $f(x)$, that ideally looks similar to $g(x)$. The importance sampling procedure follows as:

1. Draw $x_1, \dots, x_n$ from trial distribution $f(x)$.
2. Calculate the *importance weight*:
   $w_j = g(x_j)/f(x_j)$, for $j = 1, \dots, n$
3. Approximate $I = \frac{w_1 h(x_1) + \cdots + w_n h(x_n)}{w_1 + \cdots + w_n}$

Example. Compute the mean of a mixture of truncated normal distributions. Let $X \sim (.4)N(4,3)_{+(0)} + (.6)N(1,.5)_{+(0)}$.



Let the trial distribution be an Exponential(.5) distribution. Then the importance sampling follows as:

```
f <- rexp(100000, rate = .5)
w <- ( .6*dnorm(f,1,sqrt(.5))/ (1-pnorm(0,1,sqrt(.5))) +
+        .4*dnorm(f,4,sqrt(3))/ (1-pnorm(0,4,sqrt(3))) ) /
+       dexp(f,.5)

sum(w*f) / sum(w)
```

```
## [1] 2.277717
```

Note the `rtnorm()` function has this implementation using importance sampling with an exponential distribution.

Sequential Importance Sampling

Sequential Importance Sampling is commonly used in time series settings with state-space models.

For some context, consider the following model, describe the parameters in this model and explain what the model does. (This will look familiar to 536 students)

$$\begin{aligned} y_t &\sim Bernoulli(\pi_t) \\ logit(\pi_t) &= x_t\beta_t \\ \beta_t &= \beta_{t-1} + w_t \end{aligned}$$

where $w_t \sim N(0, W)$.

An algorithm known as a particle filter is implemented to learn the values of β. For simplicity we will assume that W is known, but this is not necessary.

A particle filter for time series data has a few common steps:

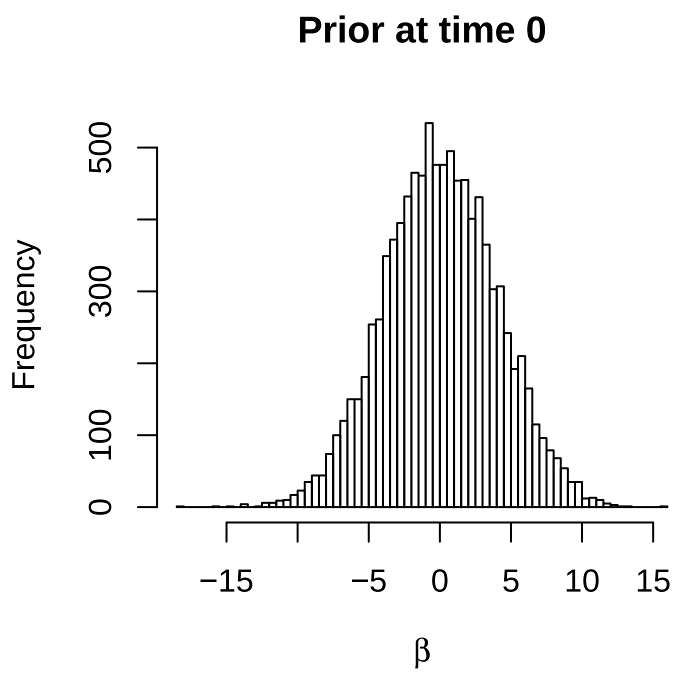
1. Prior distribution: For a model of this sort, we only need a prior distribution on β at time 0. Let $\beta_0 \sim N(0, 4^2)$

2. Evolution distribution: This controls the change in β from time t to $t+1$ and ultimately ends up as our proposal distribution.

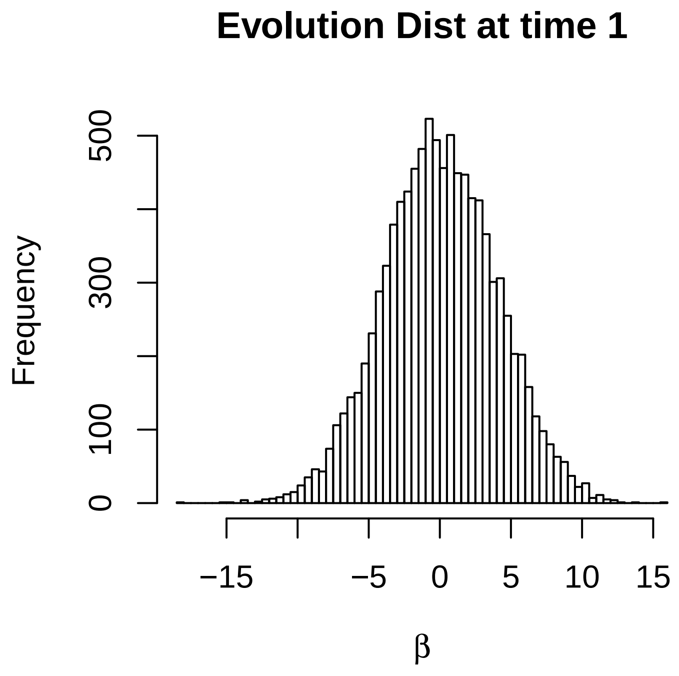
3. (Optionally) A predictive distribution for y_{t+1} can be constructed here.

4. Importance Weights and Resampling: Importance weights are calculated, that is how well do the particles from the evolution distribution match the next data point. The resampled distribution becomes the prior distribution for the next time point.

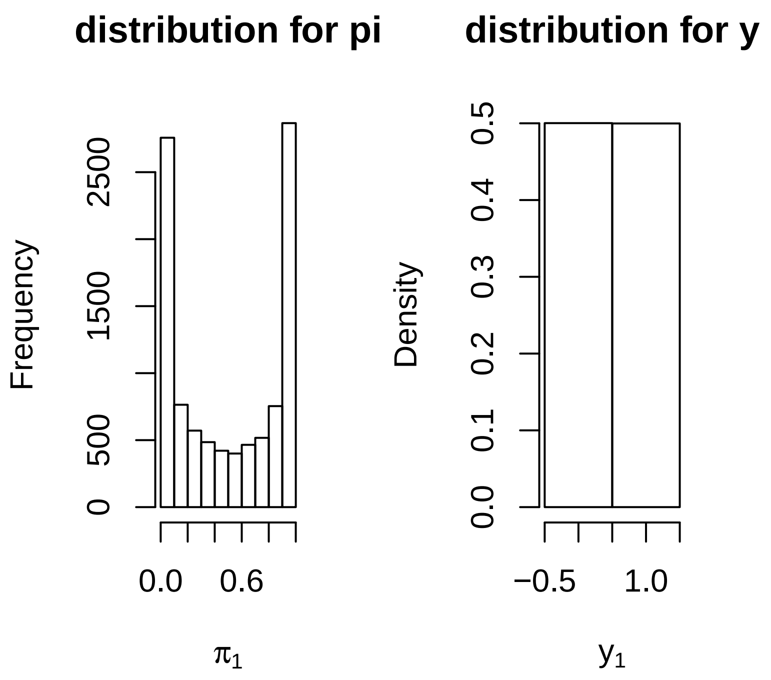
prior distributions



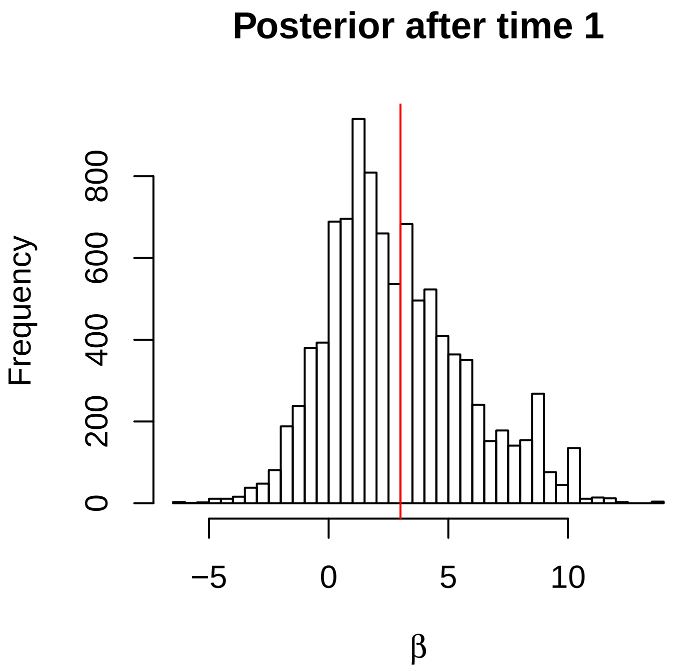
evolution distributions



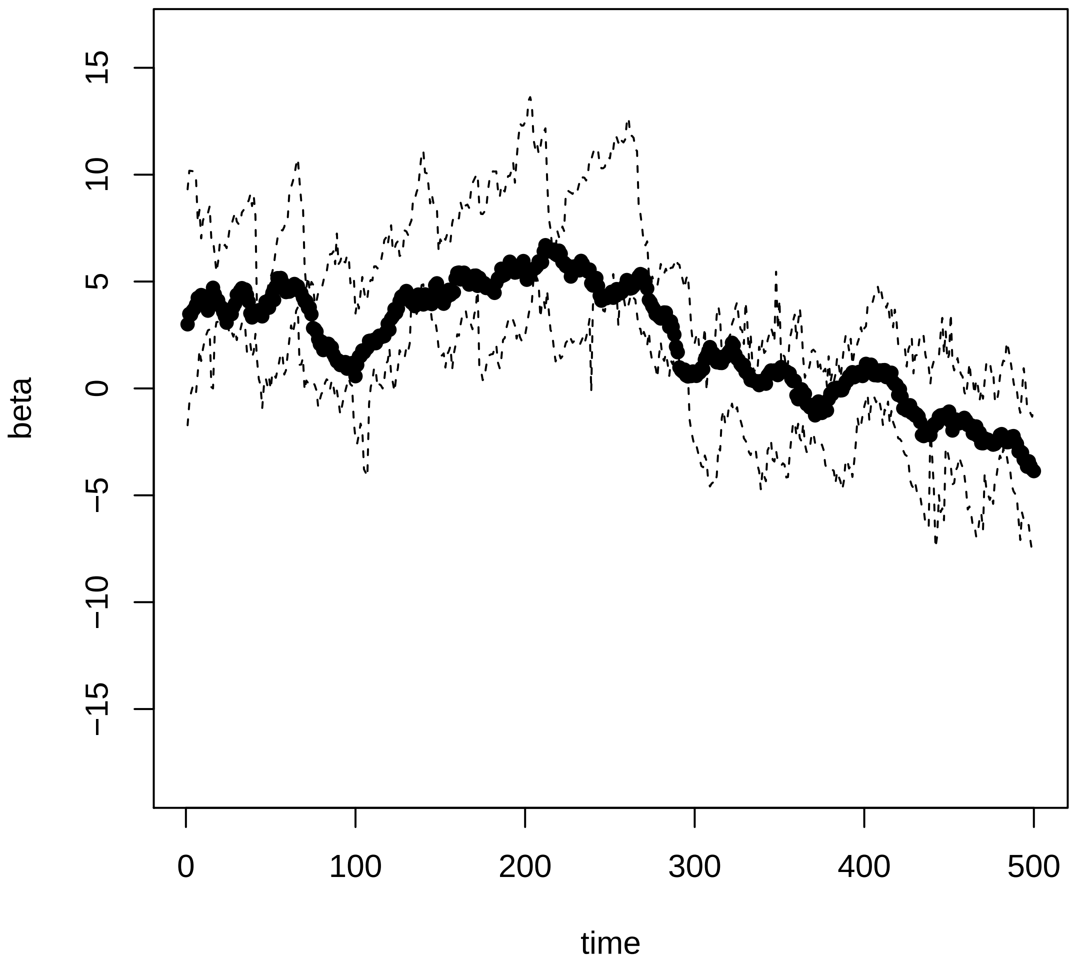
predictive distributions



compute importance weights and resample



Example for all time points



Variational Bayes

First, let's define some notation. Assume that $y = \{y_1, \ldots, y_n\}$ are observations and $\theta = \{\theta_1, \ldots, \theta_p\}$ are parameters.

In general in Bayesian statistics, the goal is to learn the posterior distribution:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)d\theta}$$

.

In many situations, the posterior is difficult to compute, either analytically or computationally. With MCMC we approximate the integral using algorithmic procedures.

An alternative to MCMC, is to use variational methods to approximate the posterior distribution. Define a new distributional family, say $q(\theta|\nu)$, with it's own variational parameters ν.

Then, assuming q is reasonably close to the posterior distribution, p, then q is used in place of p.

The Kullback-Leibler (KL) divergence is used to measure the similarity between the two distributions.

Typically with variational inference the variational distributions are assumed to factorize such that, $q(\theta_1, \ldots, \theta_p) = \prod q(\theta_i)$, each variable is independent. Typically the true posterior does not permit this factorization, as dependence between parameters often cause some of the issues with difficulties in dealing with the posterior in the first place.

Coordinate ascent algorithms are iteratively update each variational distribution (similar to a Gibbs sampler). In fact, the variational distributions often have the same distributional family as the full conditional distributions.

Variational methods are typically considerably faster than MCMC and scale to larger data sets.

For additional details, see an article by David Blei "Variational Inference: A Review for Statisticians" https://arxiv.org/pdf/1601.00670.pdf.