Bayesian Statistics

Simulation

Nan Lin
Department of Mathematics
Washington University in St. Louis

Motivation

- Posterior distribution is often known up to an unknown normalizing constant
- How do we compute posterior mean, median or variance based on a not-fully-known distribution?
- Simulation is a general approach to solve the problem if we could generate random numbers from the posterior distribution.

Law of large numbers (LLN)

- Given a random sample of size n, sample mean converges in probability to the population mean as the sample size n approaches infinity.
- Application:
  - Suppose that we are interested in the random variable $Y = g(X)$ and X has distribution is $p(x)$
  - Treat it as the population distribution and generate a large set of independent random numbers from $p(x), x_1, \ldots, x_B$
  - Then $y_i = g(x_i)$ are a set of independent random numbers of from the distribution of Y.
  - For example, to learn $E(Y)$, we can approximate it by $\bar{y}$. Based on the LLN, it can be made very accurate by making B large.

Random number generation: discrete distributions

- Suppose the distribution is supported at $\{1,2, \ldots k\}$ and the corresponding probabilities are p_i.
  - Note that the values in the support have no numerical meaning. They are just a set of labels.
- We can think the problem as randomly sampling some subjects from a population with k subjects.
  - But note, the same subject can appear multiple times, so, this is "sampling with replacement".
  - Second, in basic sampling setup, every subject is given equal chance to be selected. It is NOT the case here.
  - So, our problem is "weighted random sampling with replacement".

How to do sampling in R?

- Use the function sample()
- **Usage:** sample(x, size, replace = FALSE, prob = NULL)
- In our notation,
  - x: $\{1,2, \dots k\}$
  - size: B
  - replace: TRUE
  - prob: $\{p_1, \dots p_k\}$
- Exercise

Random number generation: continuous distributions

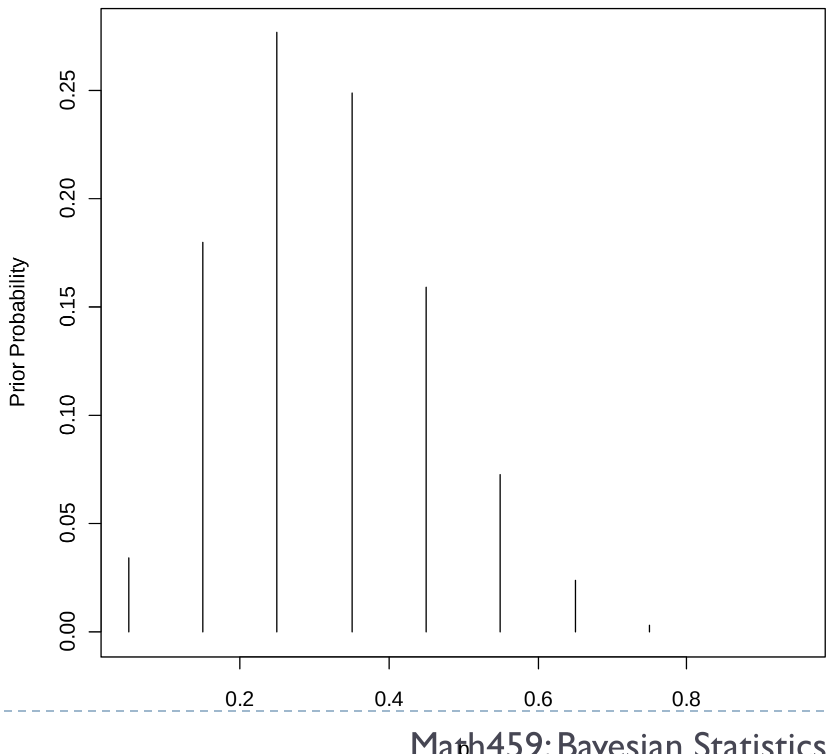
- Probability integral transformation
  - If a continuous random variable X has cdf $F(x)$, then $U = F(X)$ has distribution $U(0,1)$.
- If we start from U, and F is invertible, we will have $X = F^{-1}(U)$.
- Almost all computer languages have a random number generator for $U(0,1)$.
- So, we just need to figure out F^{-1}.
- Exercise: simulate random numbers from an exponential distribution with mean $\lambda = 1$.

How to deal with the unknown normalizing constant?

- The approaches we discussed so far all require the distribution is fully known.
- But in Bayesian inference, we usually do not know the normalizing constant. What do we do then?
- Let's see the following example.

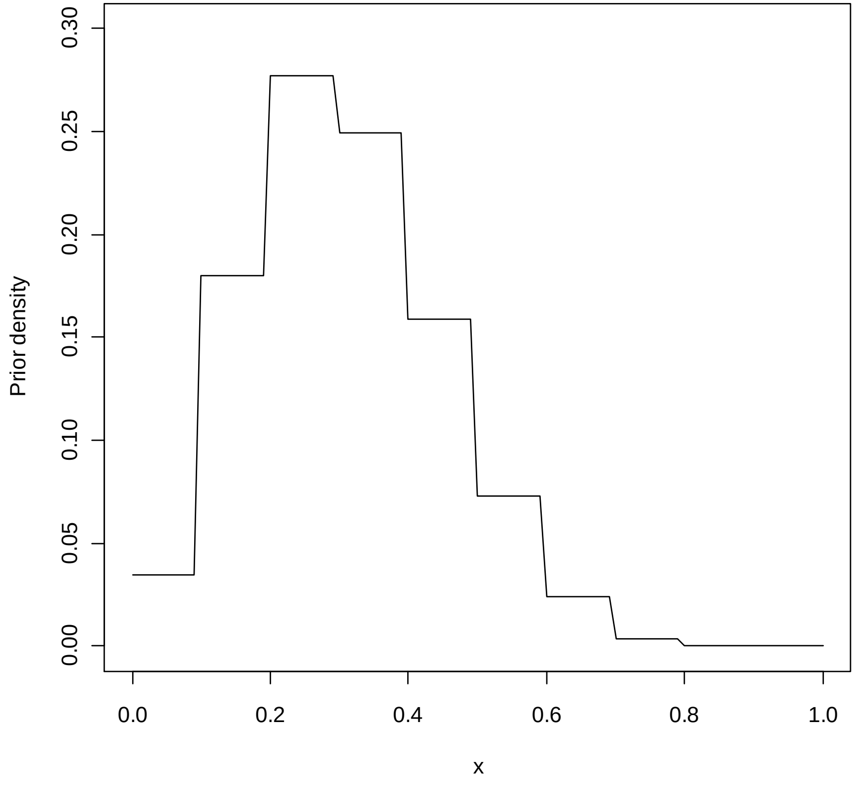
Discrete prior

- Model: $X \sim Binomial(n, p)$
  - $x = 11, n - x = 16$
- Support of p:
  - {0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95}

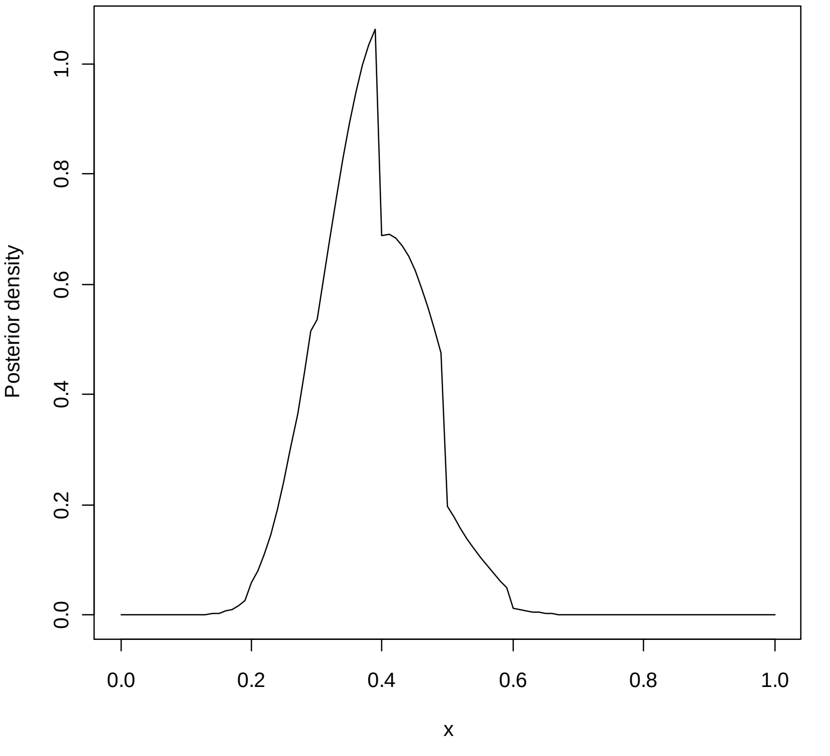


Histogram prior

- Step function



Posterior



A “brute-force” method

- How to summarize posterior computation for an arbitrary prior density $g(p)$?
  - Choose a grid of values of p over an interval that covers the posterior density
  - Compute the product of the likelihood $L(p)$ and the prior $g(p)$ on the grid
  - Normalize by dividing each product by the sum of the products. In this step, the posterior density is approximated by a discrete probability distribution on the grid
  - Using the R function sample(), take a random sample with replacement from the discrete distribution.
  - The resulting simulated draws are an approximate sample from the posterior distribution.