Successor PDF for the Metropolis-Hastings Algorithm on a Finite Interval

Bill Menke, January 31, 2024

This may be well-known, but I could not immediately turn up a reference.

The Metropolis-Hastings algorithm requires a pdf, $q(x^*|x^{(i)})$, which specifies the probability of a successor x^* given the current ensemble member $x^{(i)}$, and a pdf $q(x^{(i)}|x^*)$, which specifies the probability of the current position $x^{(i)}$ given the successor x^*. On the open interval, $-\infty < x < +\infty$, one often uses the Normal pdf

$$n(x, \bar{x}, \sigma) \equiv \frac{1}{\sqrt{2\pi}\sigma} \exp\{-½(x - \bar{x})^2/\sigma^2\}$$

Together with the choices

$$q(x^*|x^{(i)}) = n(x^*, x^{(i)}, \sigma) \quad \text{and} \quad q(x^{(i)}|x^*) = n(x^{(i)}, x^*, \sigma)$$

because methods of sampling the Normal pdf to produce a successor are widely available, and because the standard deviation σ is naturally interpreted as the scale of the neighborhood around $\bar{x}$ in which the successor x^* is highly probable. With these choices, an important ratio r used by the algorithm is unity

$$r \equiv \frac{q(x^{(i)}|x^*)}{q(x^*|x^{(i)})} = 1$$

The question is, what choices work for the half-open interval $0 < x < \infty$ and the finite interval $x_L < x < x_R$.

My idea is to use a pdfs that are proportional to a Normal pdf within the interval, but are zero outside of it. This produces a new pdf that is proportional to the Normal pdf

$$n'(x, \bar{x}, \sigma) \equiv F(\bar{x}.\sigma)\, n(x, \bar{x}, \sigma) \quad \text{where} \quad F(\bar{x}.\sigma) \equiv \frac{1}{N(x_R, \bar{x}, \sigma) - N(x_V, \bar{x}, \sigma)}$$

Here, $N(x, \bar{x}, \sigma)$ is the cumulative Normal distribution. Software for evaluating $N(x, \bar{x}, \sigma)$ is widely available. The ratio is then

$$r = \frac{N(x_R, x^{(i)}, \sigma) - N(x_V, x^{(i)}, \sigma)}{N(x_R, x^*, \sigma) - N(x_V, x^*, \sigma)}$$

A remaining issue is how to sample $n'(x, \bar{x}, \sigma)$. My strategy is to repeatedly sample $n(x, \bar{x}, \sigma)$, discarding values outside the $x_L < x < x_R$ interval and repeating until one inside it occurs. In Python

```
while True:
     xstar = np.random.normal(loc=xi,scale=sigma);
     if( (xstar>=xL) and (xstar<=xR):
          break:
```

In practice, one chooses the neighborhood to be a small fraction of the interval; that is, $\sigma \ll (x_R - x_L)$, so when $x^{(i)}$ is near the center of the interval, success usually occurs on the first iteration, and when $x^{(i)}$ is near the end of the interval, it usually occurs with one or two iterations.

I have tested these ideas by using Metropolis-Hastings to sample the exponential pdf

$$p(x) = \frac{1}{s}\exp\{-x/s\} \quad \text{with} \quad 0 < x < +\infty$$

with scale factor $s = 0.75$ and neighborhood parameter $\sigma = 0.50$, with favorable results Fig. 1).

These ideas are easily generalizable to multiple dimension; that is when **x** is a vector.

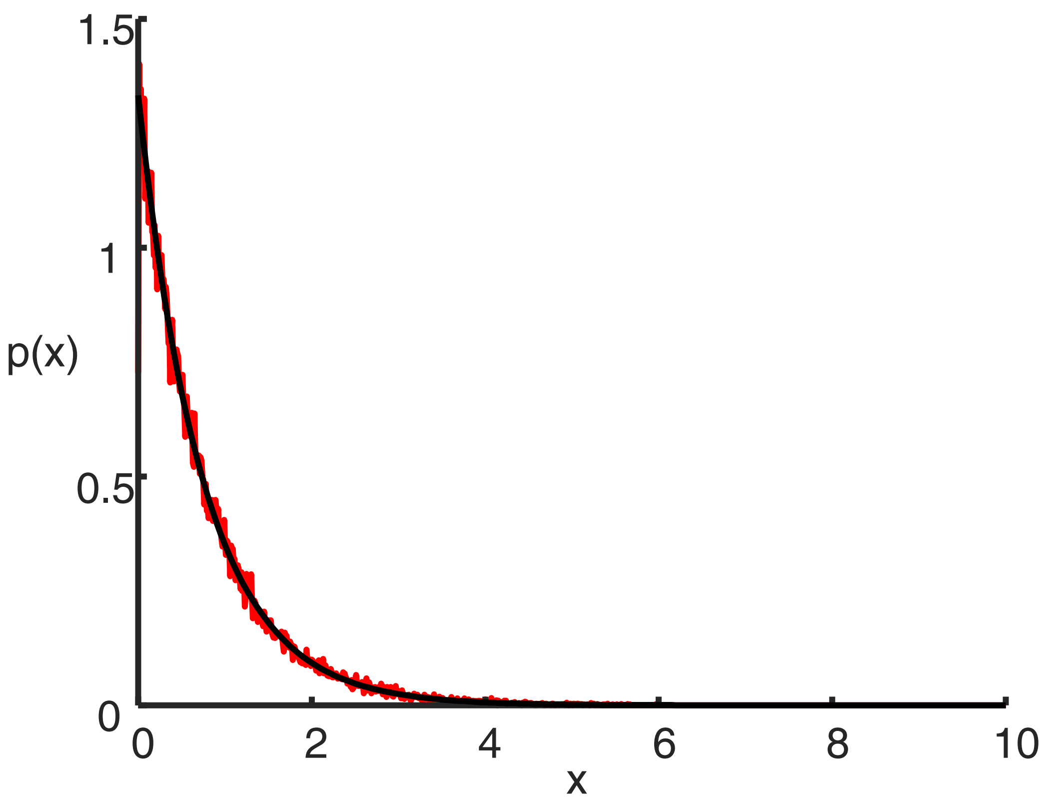

Fig 1. Results of numerical test. The true exponential pdf (black) compared with an empirical pdf (red) created by binning a 100,000-member ensemble created using the Metropolis-Hastings algorithm.