Replication Code

Usage of package samMCMC

Suppose we want to sample from a simple normal distribution. This of course can most easily and efficiently be done with the `rnorm` function in the core **stats** package; however, to demonstrate the usage of **samMCMC** we can start with this simple case. To sample from a normal distribution we first define its density

```r
# normal density where `x` is the variable, the first element of `pars` is the mean
# and the second element of `pars` is the S.D.
egFunc <- function(x, pars) {
    dnorm(x, pars[1], pars[2])
}
```

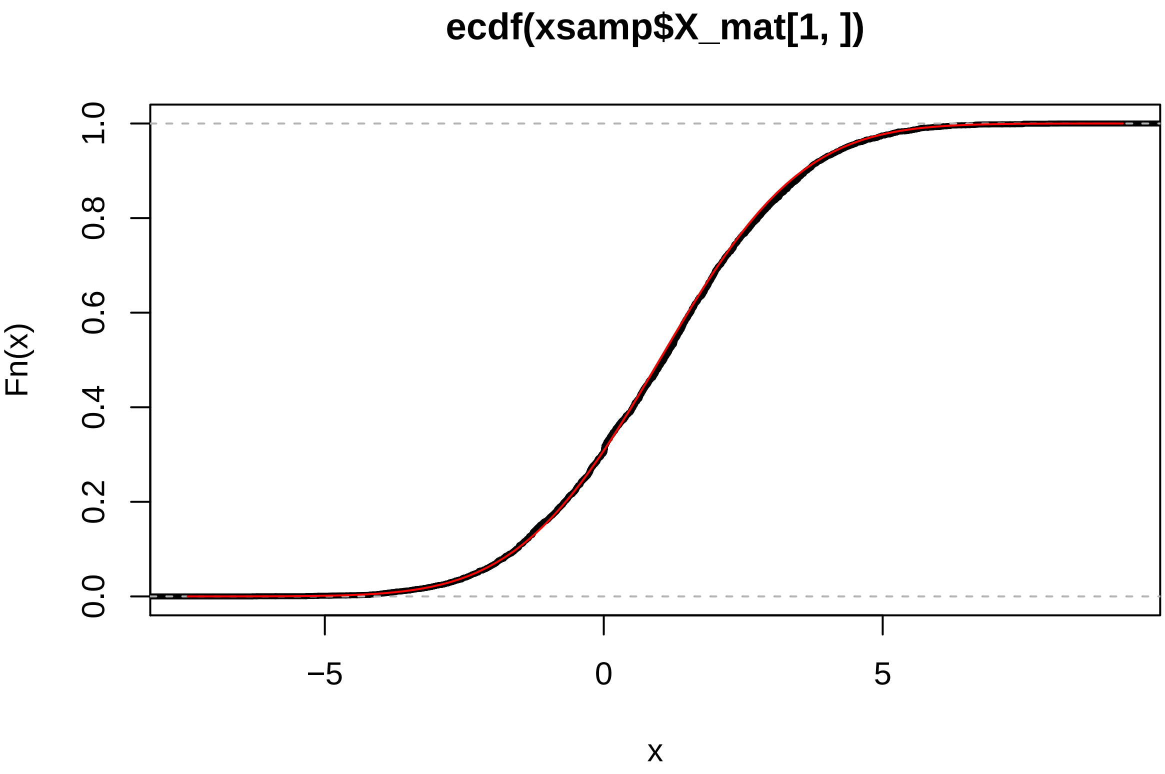
Now we can use of `samMCMC` to sample from, in this example, a normal $N(1,4)$ distribution.

```r
# control parameters for `samMCMC`
numSampBurn <- 100
t0 <- 100
numSamp <- 1000
numSampBurn <- 10000
thinning <- 100

# sample; we set control parameters `direct` to `TRUE` because we are directly
# sampling from the density
xsamp <- samMCMC(egFunc, init = 0, pars = c(1, 2),
                 control = list(numSamp = numSamp, t0 = t0, numSampBurn = numSampBurn,
                                thinning = thinning, direct = TRUE))
```

We can then compare the sample from `samMCMC` to its known generative distribution visually

```r
plot(ecdf(xsamp$X_mat[1, ]), lwd = 3)
curve(pnorm(x, 1, 2), add = TRUE, col = 'red')
```



We can also use `samMCMC` to sample given a cost function as described earlier. To do this we define a cost function (for simplicity we again use a normal distribution) and specify `direct = FALSE` in the control parameters in addition to specifying a temperature.

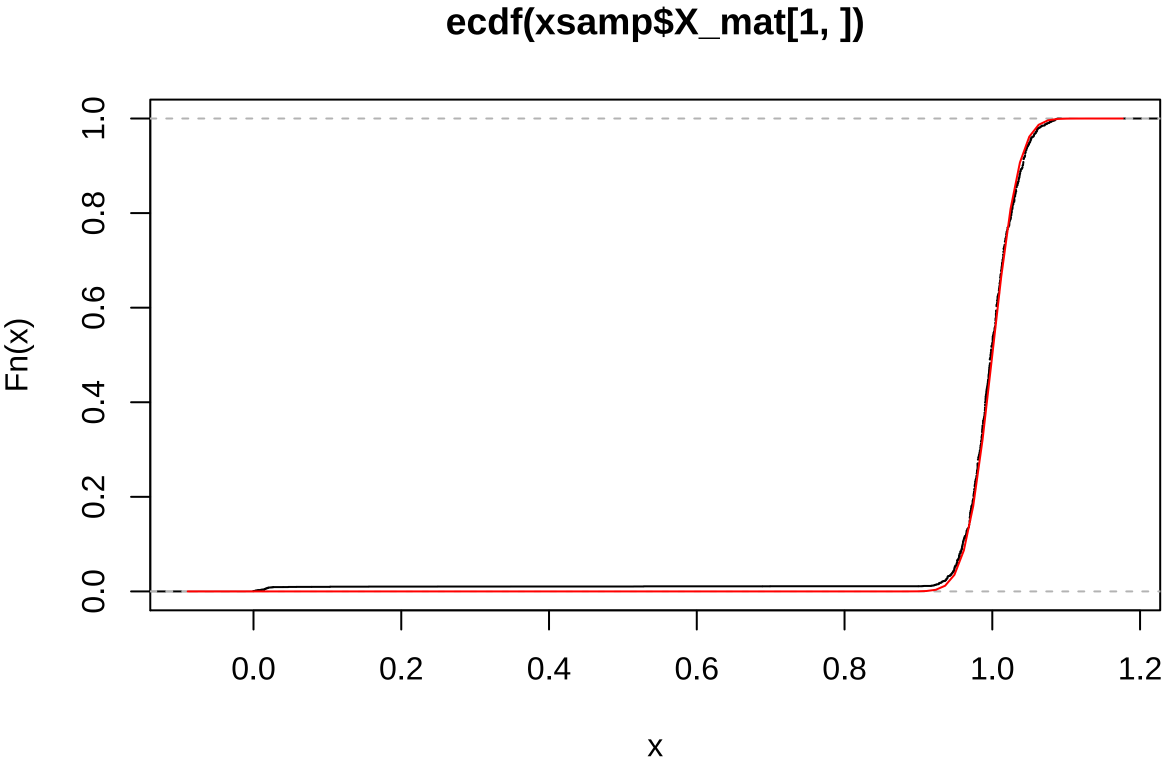
```
# normal likelihood where `m` is the mean (sd is fixed at 2) and `dat` are the data
egCostFunc <- function(m, dat) {
    - sum(dnorm(dat, m, 2, log = TRUE))
}

# example data
egData <- rnorm(5000, 1, 2)

# sample using previously defined control parameters but now with `direct = TRUE` and `temp = 1`
xsamp <- samMCMC(egCostFunc, init = 0, dat = egData,
               control = list(numSamp = numSamp, t0 = t0, numSampBurn = numSampBurn,
                              thinning = thinning, direct = FALSE, temp = 1))
```

Again, we can compare this sample from `samMCMC` with its known distribution, in this case the sampling distribution of the MLE for the mean of a normal distribution with known variance, which is $N(\hat{\mu}, \sqrt{\sigma^2/N})$.

```
plot(ecdf(xsamp$X_mat[1, ]))
mleSD <- sqrt(4 / length(egData))
curve(pnorm(x, mean = 1, sd = mleSD), add = TRUE, col = 'red')
```

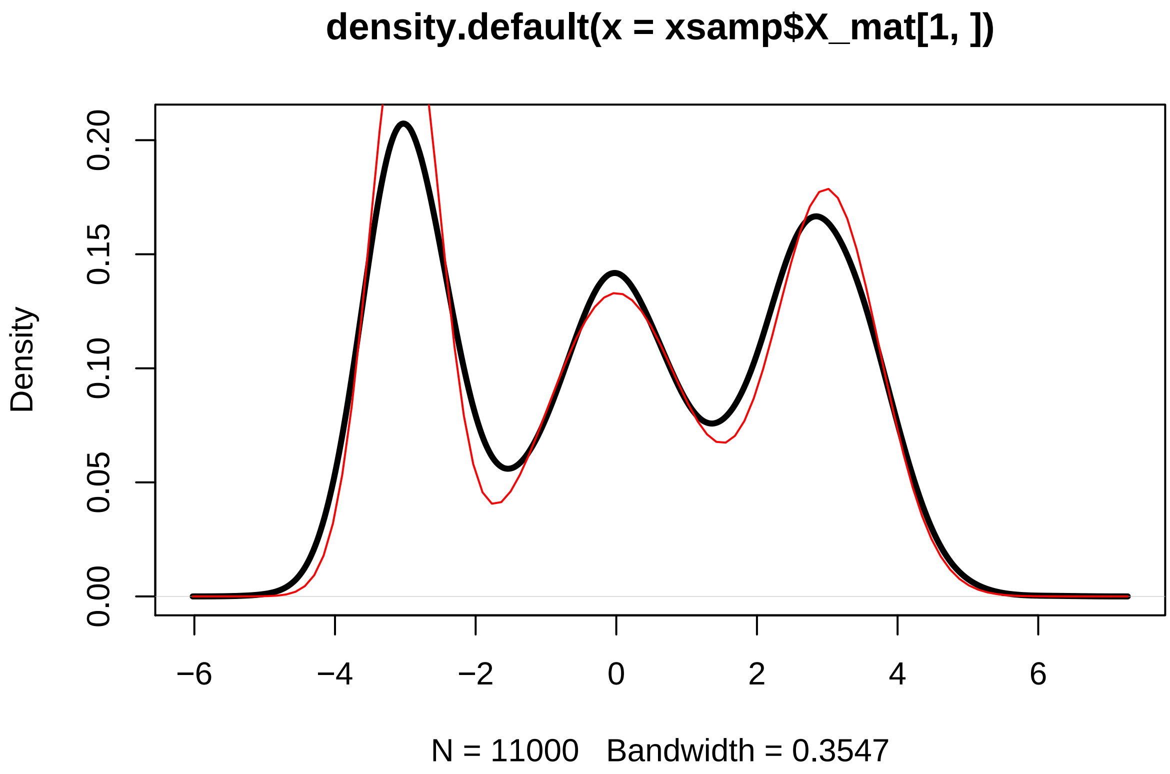


A more realistic use case of samMCMC is when the distribution we would like to sample from cannot be easily inverted, or the distribution we would like to approximate with a sample cannot be expressed analytically. We take as a simple example a mixture of normal distributions. In this example we use an equal mixture of three normals: $N(-1, 0.5)$, $N(0, 1.5)$, $N(1, 0.75)$.

```
mixFunc <- function(x) {
    1/3 * dnorm(x, -3, 0.5) + 1/3 * dnorm(x, 0, 1) + 1/3 * dnorm(x, 3, 0.75)
}

xsamp <- samMCMC(mixFunc, init = 0,
               control = list(numSamp = numSamp, t0 = t0, numSampBurn = numSampBurn,
                              thinning = thinning, direct = TRUE))

plot(density(xsamp$X_mat[1, ]), lwd = 3)
curve(mixFunc(x), add = TRUE, col = 'red')
```



Usage of package parTempeR