Gibbs_tfit

We illustrate the data-augmentation Gibbs sampler for fitting a t-density to i.i.d. data. Specifically, the model is

$$x_1, \dots, x_n \mid \mu, \sigma^2 \sim t_\nu(\mu, \sigma^2).$$

Let us assume ν is known for this example, though ν can also be updated inside the Gibbs sampler. Consider priors $\mu \mid \sigma^2 \sim N(0, \lambda^{-1}\sigma^2)$, $\sigma^2 \sim \text{IG}(\alpha/2, \beta/2)$. Consider the following hierarchical specification of the likelihood with data augmentation:

$$\begin{aligned} & x_i \mid \tau_i, \mu, \sigma^2, \nu \sim N(\mu, \tau_i^{-1}\sigma^2), \quad i = 1, \dots, n \\ & \tau_i \sim \text{Gamma}(\nu/2, \nu/2), \quad i = 1, \dots, n. \end{aligned}$$

Exploiting the fact that a t density can be expressed as a scale-mixture of normals, it is clear that one obtains the iid t likelihood by integrating over the τ_is. However, we retain the τ_is to facilitate Gibbs sampling, which cycles through the following steps:

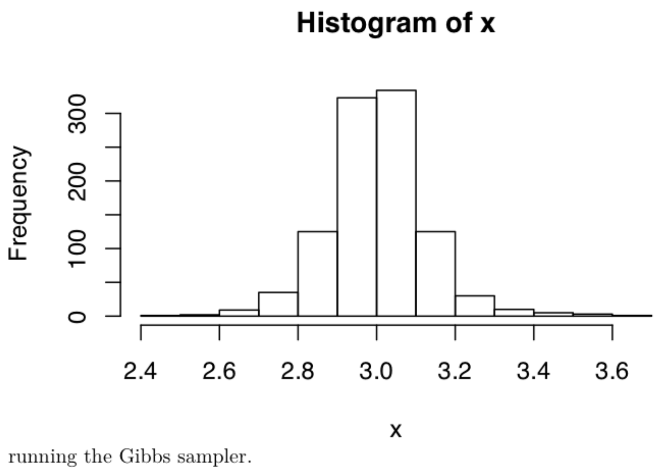
- Sample $\tau_i \mid \mu, \sigma^2, \nu, x$ independently for $i = 1, \dots, n$ from $\text{Gamma}((\nu + 1)/2, (x_i - \mu)^2/(2\sigma^2) + \nu/2)$ distributions.
- Sample $\mu \mid \tau, \sigma^2, \nu, x$ from a $N((\sum_{i=1}^n \tau_i x_i)/(\sum_{i=1}^n \tau_i + \lambda), \sigma^2/(\sum_{i=1}^n \tau_i + \lambda)$ distribution.
- Sample $\sigma^2 \mid \mu, \tau, \nu, x$ from an $\text{IG}((n + 1 + \alpha)/2, \{\sum_{i=1}^n \tau_i(x_i - \mu)^2 + \lambda\mu^2 + \beta\}/2)$ distribution.

We first simulate 1000 data points.

```
# simulate data
n = 1000; x = 3 + 0.1*rt(n,6);
```

Let us plot a histogram of the data first.

```
hist(x)
```



We now define stuff necessary for running the Gibbs sampler.

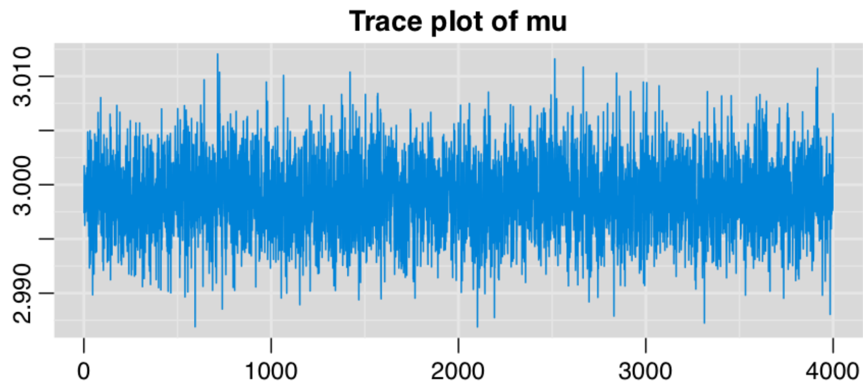
```r
# --- define global parameters --- #
MCMC = 20000; BURNIN = 10000; thin = 5; N=BURNIN+MCMC; effsamp=(N-BURNIN)/thin
# --- set prior hyperparameters --- #
lambda = 0.01; alpha = 0.01; beta = 0.01    # vague prior
# --- output files --- #
muout = rep(0,effsamp); sigmasqout = rep(0,effsamp)
# --- initialize parameters --- #
mu = 0; sigmasq = 1; gammai = rep(1,n); nu = 6
```

Now we implement the Gibbs sampler.

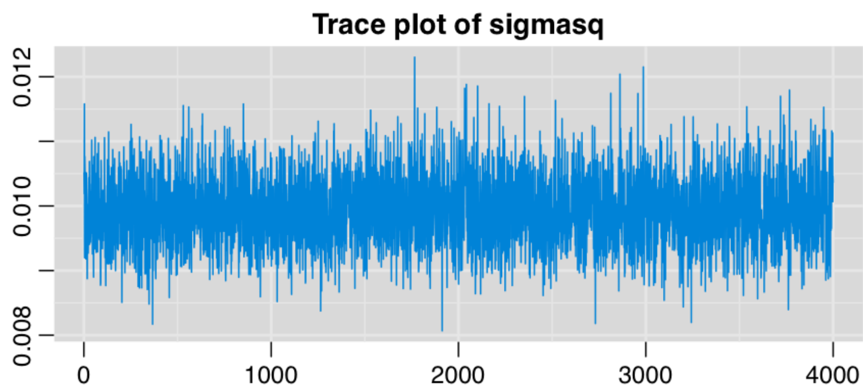
```r
set.seed(86)
# --- start Gibbs sampling --- #

for(g in 1:N){

  # -- update mu -- #
  temp = (sum(gammai) + lambda)
  pvar = sigmasq/temp
  pmean = sum(gammai*x)/temp
  mu = rnorm(1,pmean,sqrt(pvar))

  # -- update sigmasq -- #
  pshape = (n + 1 + alpha)/2; pscale = (sum(gammai*(x-mu)^2) + lambda*mu^2 + beta)/2
  sigmasq = 1/rgamma(1,pshape,scale= 1/pscale)

  # -- update gammai -- #
  pshape = nu/2 + 0.5; pscalei = (x-mu)^2/(2*sigmasq) + nu/2;
  gammai = rgamma(n,pshape,scale = 1/pscalei)

  # -- save output -- #
  if(g > BURNIN && g%%thin== 0)
  {
    muout[(g-BURNIN)/thin] = mu;
    sigmasqout[(g-BURNIN)/thin]=sigmasq;

  }

}
```

Draw traceplots of μ and σ^2.

```r
library(coda)
library(mcmcplots)
muM = as.mcmc(muout); sigsqM = as.mcmc(sigmasqout)
traplot(muM,main = "Trace plot of mu")
```




```r
traplot(sigsqM,main = "Trace plot of sigmasq")
```



Posterior summaries:

```r
summary(muout)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   2.987   2.997   2.999   2.999   3.001   3.012
```

```r
summary(sigmasqout)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## 0.008067 0.009545 0.009912 0.009933 0.010300 0.012300
```