CAVI algorithm for normal mean and precision

Heejung Shim

Simulate data

Simulate 50 samples from $N(50, 2^2)$.

```
set.seed(30027)
X = rnorm(50, 50, 2)
```

Implementation CAVI algorithm

$\mu^*, \sigma^{2*} \Rightarrow q^*_\mu$

$a^*, b^* \Rightarrow q^*_\tau$

```
# X : data
# mu0, lambda0 : prior for mu
# a0, b0 : prior for tau
# initial values for mu*, sigma2*, a*, b*: mu.vi.init, sigma2.vi.init, a.vi.init, b.vi.init
# epsilon : If the ELBO has changed by less than epsilon, the CAVI algorithm will stop
# max.iter : maximum number of iteration
cavi.normal <- function(X, mu0, lambda0, a0, b0, mu.vi.init, sigma2.vi.init, a.vi.init,
                        b.vi.init, epsilon=1e-5, max.iter=100) {

  n = length(X)

  mu.vi = mu.vi.init
  sigma2.vi = sigma2.vi.init
  a.vi = a.vi.init
  b.vi = b.vi.init

  # store the ELBO for each iteration
  elbo = c()

  # I will store mu*, sigma2*, a*, b* for each iteration
  mu.vi.list = sigma2.vi.list = a.vi.list = b.vi.list = c()

  # compute the ELBO using initial values of mu*, sigma2*, a*, b*
  Elogq.mu = -log(sigma2.vi)/2
  Elogq.tau = log(b.vi) - lgamma(a.vi) + (a.vi -1)*digamma(a.vi) - a.vi
  A = sigma2.vi + mu.vi^2 - 2*X*mu.vi + X*X
  B = sigma2.vi + mu.vi^2 - 2*mu0*mu.vi + mu0^2
  Elogp.x.mu.tau = (n+1)/2*(-log(b.vi)+digamma(a.vi))-0.5*a.vi/b.vi*(sum(A) + lambda0*B)
    + (a0-1)*(-log(b.vi)+digamma(a.vi)) - b0*a.vi/b.vi

  elbo = c(elbo, Elogp.x.mu.tau -Elogq.mu - Elogq.tau)
  mu.vi.list = c(mu.vi.list, mu.vi)
  sigma2.vi.list = c(sigma2.vi.list, sigma2.vi)
  a.vi.list = c(a.vi.list, a.vi)
```

$E_{\mu,\tau}(\log q^*_\mu(\mu))$

$\alpha - \frac{1}{2}\log\sigma^{2*}$

$E_{\mu,\tau}(\log q^*_\tau(\tau))$

$\psi(a^*)$

$E_{\mu,\tau}(\log p(x_1 \dots x_n, \mu, \tau))$

** digamma

$\Rightarrow$ log derivative of gamma

$\psi(x) = \frac{d}{dx}\ln(\Gamma(x))$

```r
  b.vi.list = c(b.vi.list, b.vi)

  # set the change in the ELBO with 1
  delta.elbo = 1

  # number of iteration
  n.iter = 1

  # If the elbo has changed by less than epsilon, the CAVI will stop.
  while((delta.elbo > epsilon) & (n.iter <= max.iter)){
                  μ*          σ²*
    # Update mu.vi and sigma2.vi
    mu.vi = (lambda0*mu0 + sum(X))/(lambda0 + n)
    sigma2.vi = b.vi/a.vi/(lambda0 + n)
                 a*        b*
    # Update a.vi and b.vi
    a.vi = (n+1)/2 + a0
    A = sigma2.vi + mu.vi^2 - 2*X*mu.vi + X*X
    B = sigma2.vi + mu.vi^2 - 2*mu0*mu.vi + mu0^2
    b.vi = b0 + 0.5*sum(A) +0.5*lambda0*B

    # compute the ELBO using the current values of mu*, sigma*, a*, b*
    Elogq.mu = -log(sigma2.vi)/2
    Elogq.tau = log(b.vi) - lgamma(a.vi) + (a.vi -1)*digamma(a.vi) - a.vi
    Elogp.x.mu.tau = (n+1)/2*(-log(b.vi)+digamma(a.vi))-0.5*a.vi/b.vi*(sum(A)
               + lambda0*B) + (a0-1)*(-log(b.vi)+digamma(a.vi)) - b0*a.vi/b.vi

    elbo = c(elbo, Elogp.x.mu.tau - Elogq.mu - Elogq.tau)
    mu.vi.list = c(mu.vi.list, mu.vi)
    sigma2.vi.list = c(sigma2.vi.list, sigma2.vi)
    a.vi.list = c(a.vi.list, a.vi)
    b.vi.list = c(b.vi.list, b.vi)

    # compute the change in the elbo
ΔELBO delta.elbo = elbo[length(elbo)]  - elbo[length(elbo)-1]

    # increase the number of iteration
    n.iter = n.iter + 1
  }

  return(list(elbo = elbo, mu.vi.list = mu.vi.list,
              sigma2.vi.list=sigma2.vi.list, a.vi.list=a.vi.list, b.vi.list=b.vi.list))
}
```

Apply the CAVI algorithm to the simulated data set

We will consider the following priors: $\mu_0=0$ $\lambda_0=0.01$ $a_0=2$ $b_0=0.$

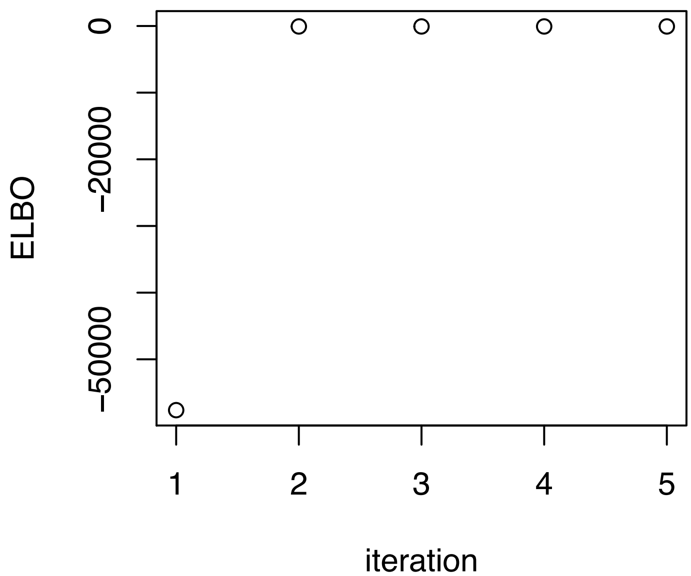
$$\mu|\tau \sim N(0, \frac{1}{0.01\tau}) \quad \tau \sim Gamma(2, 2)$$

Run the CAVI algorithm with different initial values and check that the ELBO increases at each step by plotting them.

```r
mu0=0
lambda0=0.01
a0=2
b0=2
cavi1 = cavi.normal(X, mu0=mu0, lambda0=lambda0, a0=a0, b0=b0, mu.vi.init=2, sigma2.vi.init=4,
                    a.vi.init = 2, b.vi.init=2, epsilon=1e-5, max.iter=100)
cavi.res = cavi1
cavi.res$elbo
```

```
## [1] -57628.02235    -69.47931    -69.14256    -69.14251    -69.14251
```

```r
plot(cavi.res$elbo, ylab='ELBO', xlab='iteration')
```




```r
print(paste("mu* and sigma2* = (",
            round(cavi.res$mu.vi.list[length(cavi.res$mu.vi.list)],2), ",",
            round(cavi.res$sigma2.vi.list[length(cavi.res$sigma2.vi.list)],2), ")", sep=""))
```

```
## [1] "mu* and sigma2* = (49.92,0.08)"
```

```r
print(paste("a* and b* = (",
            round(cavi.res$a.vi.list[length(cavi.res$a.vi.list)],2), ",",
            round(cavi.res$b.vi.list[length(cavi.res$b.vi.list)],2), ")", sep=""))
```

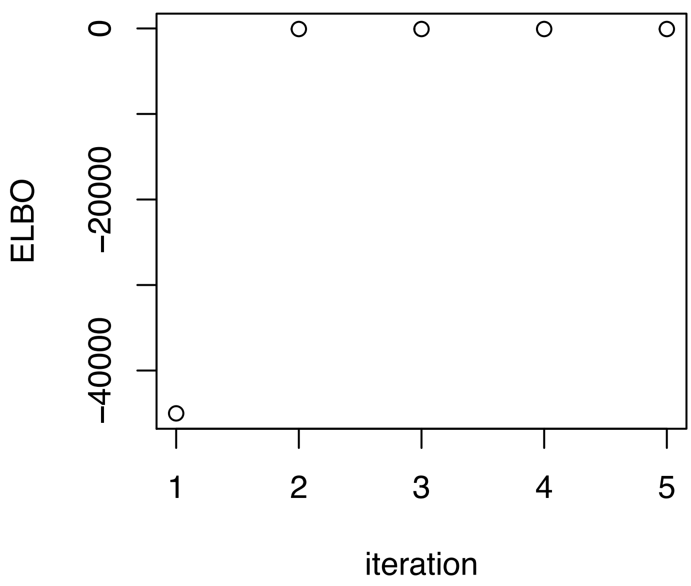
```
## [1] "a* and b* = (27.5,116.38)"
```

CAVI with different initial

```r
cavi2 = cavi.normal(X, mu0=mu0, lambda0=lambda0, a0=a0, b0=b0, mu.vi.init=-10, sigma2.vi.init=4,
                    a.vi.init = 10, b.vi.init=20, epsilon=1e-5, max.iter=100)
cavi.res = cavi2
cavi.res$elbo
```

```
## [1] -45008.17186    -69.25229    -69.14253    -69.14251    -69.14251
```

```r
plot(cavi.res$elbo, ylab='ELBO', xlab='iteration')
```




```
print(paste("Estimate mu* and sigma2* = (",
            round(cavi.res$mu.vi.list[length(cavi.res$mu.vi.list)],2), ",",
            round(cavi.res$sigma2.vi.list[length(cavi.res$sigma2.vi.list)],2), ")", sep=""))
```

```
## [1] "Estimate mu* and sigma2* = (49.92,0.08)"
```

```
print(paste("Estimate a* and b* = (",
            round(cavi.res$a.vi.list[length(cavi.res$a.vi.list)],2), ",",
            round(cavi.res$b.vi.list[length(cavi.res$b.vi.list)],2), ")", sep=""))
```

```
## [1] "Estimate a* and b* = (27.5,116.38)"
```

The two CAVI runs have (equally) highest ELBO. You can see that approximuated posterior distributions from the runs are the same. I will use the output from the first run: $q^*_\mu(\mu)$ is a pdf of $N(49.92, 0.08)$ and $q^*_\tau(\tau)$ is a pdf of $Gamma(27.5, 116.38)$.

$p(\mu,\tau|x) \approx q^*_\mu(\mu) \cdot q^*_\tau(\tau)$

NOTE P20

Let's compare CAVI results with the exact posterior distribution

Sample from the exact distribution

exact posterior

simulate τ

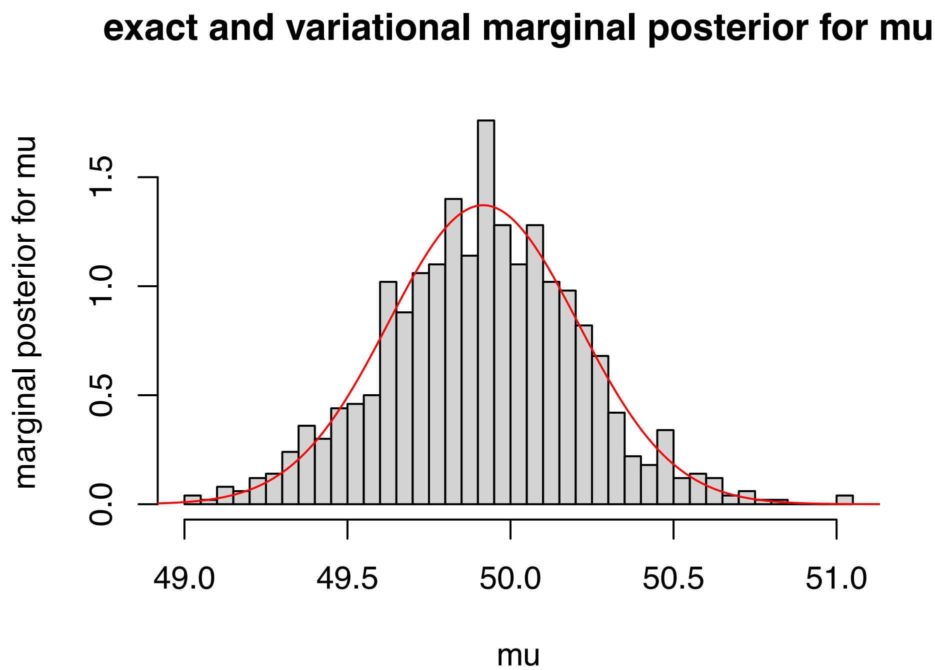
use simulated τ to simulate μ.

```
A = X - mean(X)
n = length(X)

shape = a0 + n/2
rate = b0 + 0.5*sum(A*A) + n*lambda0/(lambda0 + n)*(mean(X)-mu0)*(mean(X)-mu0)/2
tau.true = rgamma(1000, shape =shape, rate = rate)

mean = rep((lambda0*mu0 + sum(X))/(lambda0 + n), 1000)
sd = sqrt(1/(lambda0+n)/tau.true)
mu.true = rnorm(1000, mean, sd)
```

Let's compare the exact and variational marginal posterior distributions.

```
mu.vi = cavi.res$mu.vi.list[length(cavi.res$mu.vi.list)]
sigma2.vi = cavi.res$sigma2.vi.list[length(cavi.res$sigma2.vi.list)]

hist(mu.true, breaks=50, freq=F, xlab="mu",
     ylab="marginal posterior for mu", main="exact and variational marginal posterior for mu")
```

```
xval <- seq(min(mu.true)-1, max(mu.true)+1, 0.01)
lines(xval, dnorm(xval, mu.vi, sqrt(sigma2.vi)), col="red")
```




```
a.vi = cavi.res$a.vi.list[length(cavi.res$a.vi.list)]
b.vi = cavi.res$b.vi.list[length(cavi.res$b.vi.list)]

hist(tau.true, breaks=50, freq=F, xlab="tau",
     ylab="marginal posterior for tau", main="exact and variational marginal posterior for tau")
xval <- seq(min(tau.true)-1, max(tau.true)+1, .01)
lines(xval, dgamma(xval, shape = a.vi, rate = b.vi), col="red")
```