Errors in Predictions

November 13, 2018

1 Errors in Predicted Values

This is an attempt to use simulation to characterize the errors in predictions from two different methods of calculating coefficients (and then predicted values) for recentered data.

In principle, the predicted values from the original (uncentered) data/model should be exactly equal to the predicted values from a model estimated from the centered data.

Here we compare two different methods of generating model coefficients for recentered data. In the first method, we actually recenter the data, then re-estimate the model, then calculate predicted values. In the second method, we calculate the model coefficients directly, without re-estimating the model. Then we calculate predicted values (using the same recentered data as in the first method).

The model is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3 + \beta_7 x_1 x_2 x_3$$

```
In [4]: # first load some useful functions
        setwd("z:/r/stdParm-R")
        source("stdParm functions.r") # for direct coefficient estimation
        source("gen_ex_models.r")     # to generate simulated data
        source("kd.plot.overlay.r")   # for plotting our results
```

1.0.1 Main routine

- data simulation
- models estimation
- return difference in predicted values

```
In [5]: sim.3var.center <- function (i, nvals) {
            df <- gen_3x(nvals,
                         means=c(-0.5,0,0.5),
                         sigma=matrix(c(1,.5,.25,.5,1,.3,.25,.3,1), ncol=3),
                         coefs=c(1,2,0,-2,.75,.5,.25,0))

            model1 <- lm(y~x1*x2*x3, data=df)

            b <- coef(model1)
            b.terms <- names(b)
```

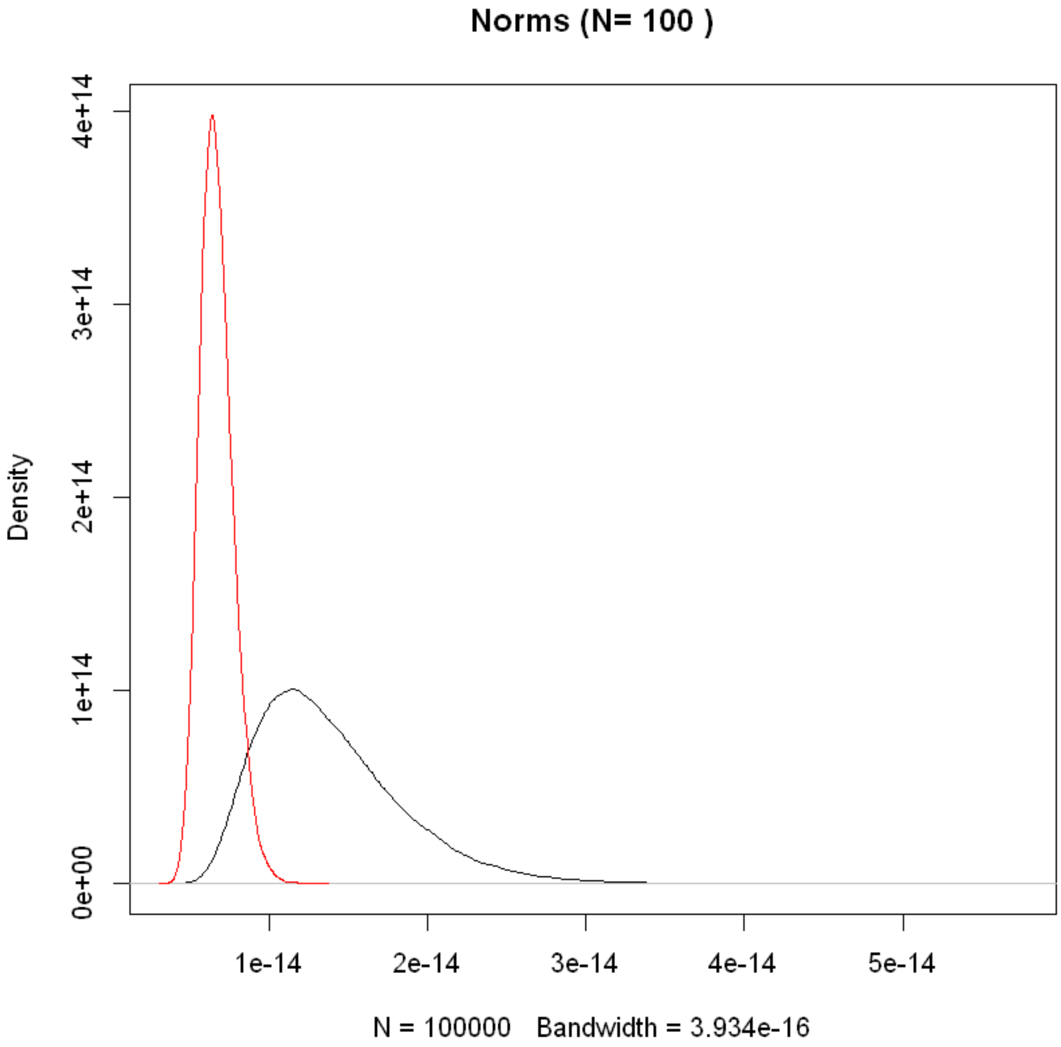
```
          df2 <- df
          df2[,2:4] <- apply(df[,2:4],2,scale, scale=FALSE)
          model2 <- lm(y~x1*x2*x3, data=df2)

          x.means <- colMeans(df[,2:4])
          C <- matrix.build.clean(x.means, b.terms)

          y1 <- predict(model1) # original
          y2 <- predict(model2) # recentered data
          y3 <- cbind(rep(1,nvals), as.matrix(df2[2:4]),
                      df2[["x1"]]*df2[["x2"]],df2[["x1"]]*df2[["x3"]],df2[["x2"]]*df2[["x3"]],
                      df2[["x1"]]*df2[["x2"]]*df2[["x3"]])%*%(C%*%b)

          return(c(sqrt(sum((y1-y2)^2)),  # recenter data
                   sqrt(sum((y1-y3)^2))))        # recenter coefs
        }
```

1.1 Simulation

- Numerical results: first is recentered data, second is directly recentered coefficients
- Graphical results: black is recentered data, red is directly recentered coefficients

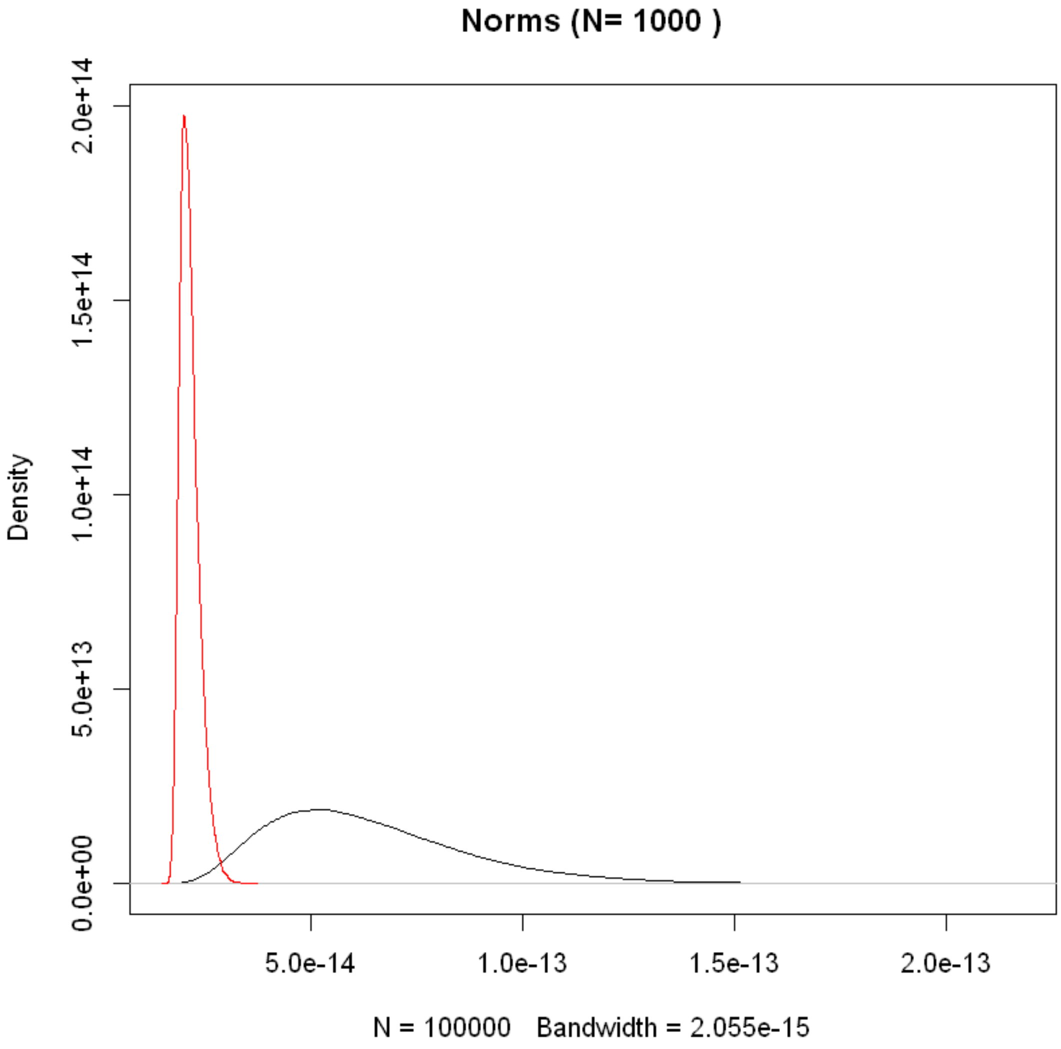
```
In [7]: library(parallel)

        cl <- makeCluster(8)

        nvals <- 100L
        clusterExport(cl, c("nvals", "sim.3var.center", "gen_3x",
                            "mean.to.matrix", "matching.terms", "vars.in.terms", "kron", "matrix
        devnorms <- parSapply(cl, 1:100000, sim.3var.center, nvals)
        rowMeans(devnorms)
        kd.plot.overlay(t(devnorms), nvals)
```

1. 1.38657165242713e-14 2. 6.69900627822363e-15



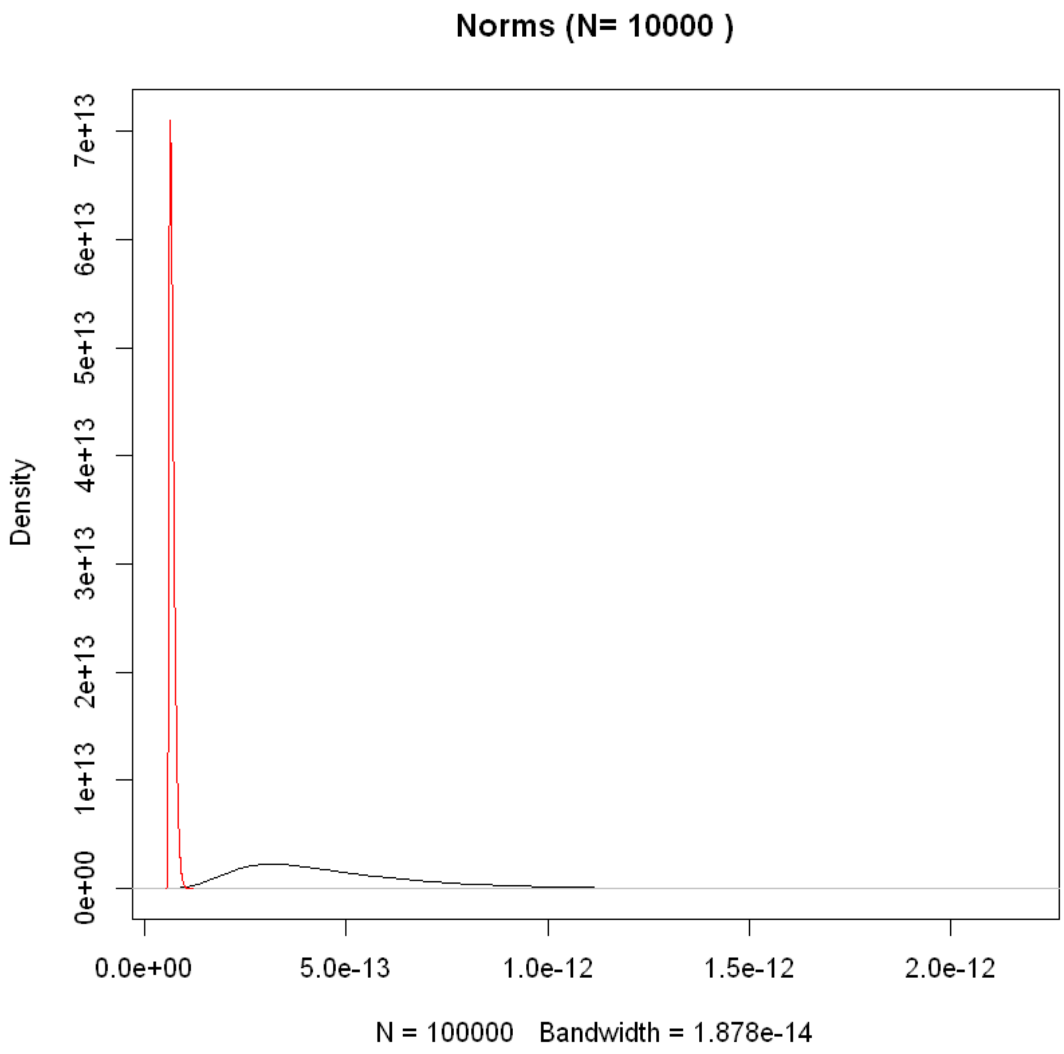
```
In [8]: nvals <- 1000L
        clusterExport(cl, c("nvals", "sim.3var.center", "gen_3x",
                            "mean.to.matrix", "matching.terms", "vars.in.terms", "kron", "matrix
        devnorms <- parSapply(cl, 1:100000, sim.3var.center, nvals)
        rowMeans(devnorms)
        kd.plot.overlay(t(devnorms), nvals)
```

1. 6.38590038661258e-14 2. 2.13792861363091e-14



```
In [9]: nvals <- 10000L
        clusterExport(cl, c("nvals", "sim.3var.center", "gen_3x",
                            "mean.to.matrix", "matching.terms", "vars.in.terms", "kron", "matrix
        devnorms <- parSapply(cl, 1:100000, sim.3var.center, nvals)
        rowMeans(devnorms)
        kd.plot.overlay(t(devnorms), nvals)

        stopCluster(cl)
```

1. 4.58041073857275e-13 2. 6.76170082916634e-14


Norms (N= 10000)
Density
0e+00
1e+13
2e+13
3e+13
4e+13
5e+13
6e+13
7e+13
0.0e+00
5.0e-13
1.0e-12
1.5e-12
2.0e-12
N = 100000 Bandwidth = 1.878e-14