Random Forests

Günter J. Hitsch

February 21, 2017

```r
library(data.table)
library(ggplot2)
library(ranger)
```

Data

As in the regression tree example, we use household-level private label share data aggregated at the *year* level.

```r
load("./Data/PL_shares_annual.RData")
```

For easier interpretability, we convert income and the Zillow housing index to logs.

```r
PL_shares_DT[, `:=`(income       = log(income),
                    zillow_index = log(zillow_index))]
setnames(PL_shares_DT, c("income", "zillow_index"), c("log_income", "log_zillow"))
```

Using `ranger`

The `ranger` function will throw an error if it detects missing data values(`NA`). Hence, we first remove observations with missing values:

```r
PL_shares_DT = PL_shares_DT[complete.cases(PL_shares_DT)]
```

Now estimate a random forest:

```r
fit = ranger(PL_share ~ . - log_zillow,
             data = PL_shares_DT[, !"household_code", with = FALSE],
             num.trees = 2000,
#            importance = "impurity",
             seed = 1776)
```

Options:

- `num.trees` specifies the number of trees to incorporate. When you first run the algorithm, use a small number to get a sense of how slow or fast the trees are grown (`ranger` will provide some corresponding output, unless you add the option `verbose = FALSE`). The default setting is `num.trees = 500`. In a final production run you may choose a larger number, maybe 1000 or more if time permits.
- Set a `seed` inside ranger to exactly replicate your previous results.
- The `importance = "impurity"` option allows you to later use the `importance` function (e.g. `importance(fit)`) to display the variable importance measures.

Note

Don't exclude variables inside a formula when using `ranger`! E.g., do not use

```r
PL_share ~ . - household_code
```

Instead, exclude the variables directly from the data.table:

```
DT[, !c("var_1", "var_2"), with = FALSE]
```

Random forest prediction

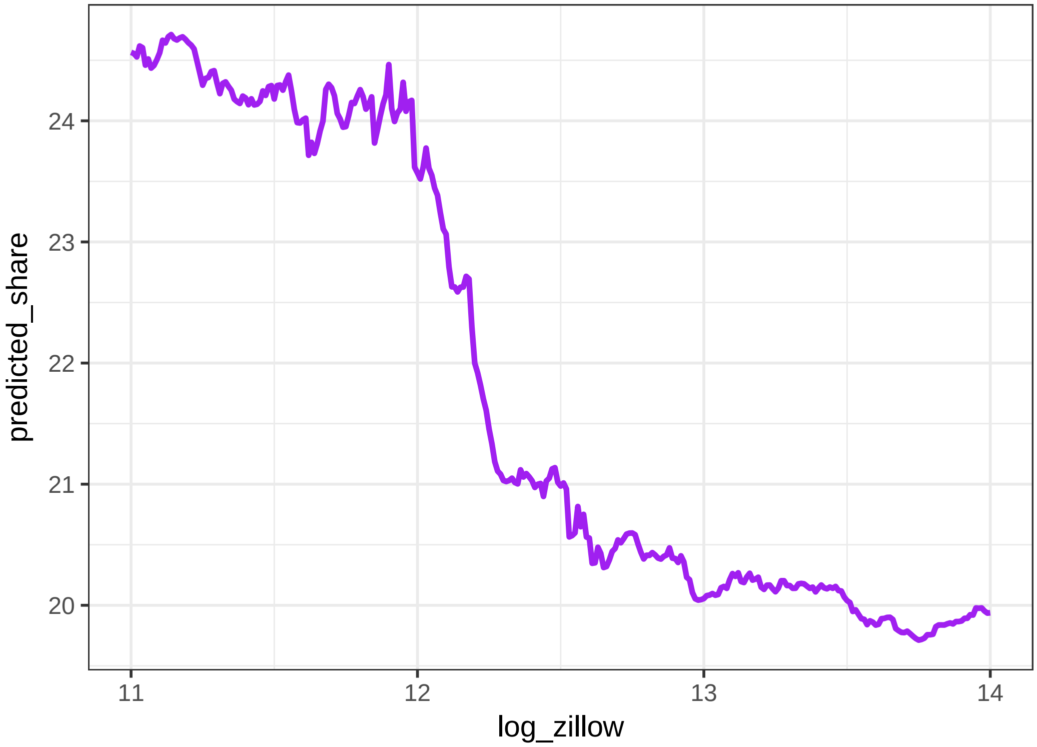
Create a `data.table` with a range of Zillow index values.

```
predict_DT = data.table(log_zillow     = seq(11, 14, 0.01),
                        log_income     = 9.5,
                        year           = 2012,
                        unemployed     = FALSE,
                        education      = "Graduated High School",
                        age            = 55,
                        size           = 2,
                        has_children   = 1,
                        female_head    = FALSE,
                        marital_status = "Married",
                        race           = "White",
                        hispanic_origin= "Hispanic")
```

Predict and graph the predicted private-label share vs. the Zillow index:

```
predict_ranger = predict(fit, data = predict_DT)
predict_DT[, predicted_share := predict_ranger$predictions]
```

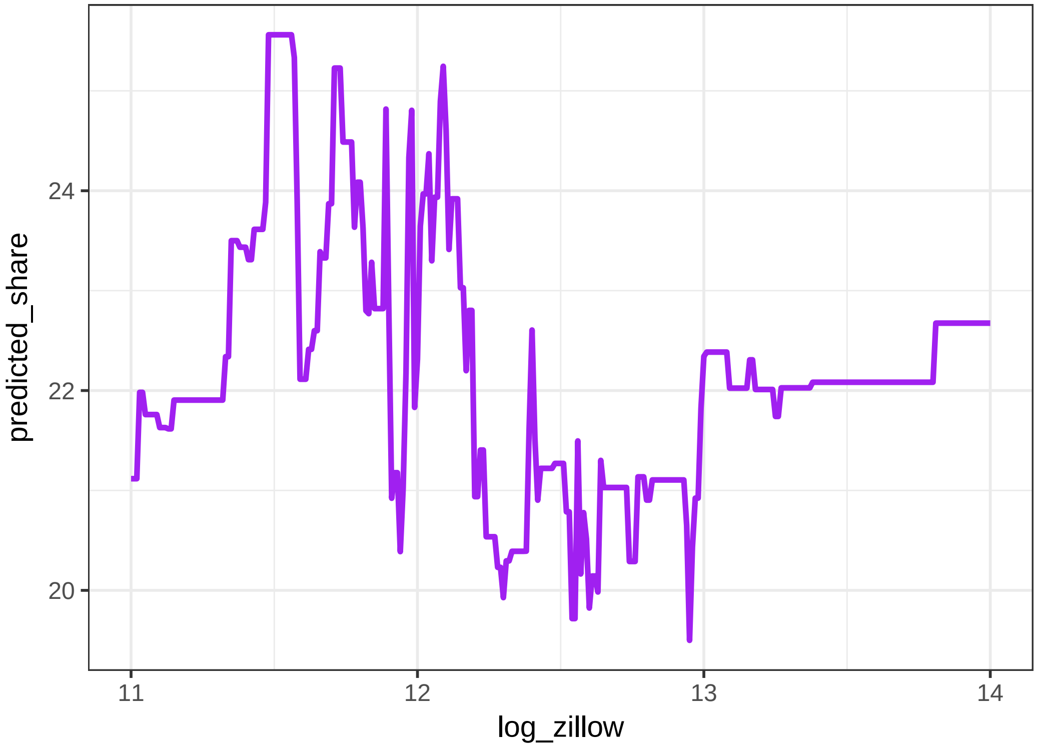
```
ggplot(predict_DT, aes(x = log_zillow, y = predicted_share)) +
   geom_line(color = "purple", size = 1) +
   theme_bw()
```



Here we use only 25 trees in the prediction:

```
predict_ranger = predict(fit, data = predict_DT, num.trees = 25)
predict_DT[, predicted_share := predict_ranger$predictions]
```

```
ggplot(predict_DT, aes(x = log_zillow, y = predicted_share)) +
   geom_line(color = "purple", size = 1) +
   theme_bw()
```



pdf graphs

Predict a matrix with columns for each individual tree using the `predict.all = TRUE` option.

```
predict_ranger = predict(fit, data = predict_DT, predict.all = TRUE)
```

First, plot each individual tree.

```
L = 100

pdf(file = "Indiviual-Trees.pdf", width = 8, height = 6)
for (i in 1:L) {
   predict_DT[, predicted_share := predict_ranger$predictions[, i]]

   plot_i = ggplot(predict_DT, aes(x = log_zillow, y = predicted_share)) +
               geom_line(color = "purple", size = 1) +
               scale_y_continuous(paste0("PL share, tree: ", i)) +
               theme_bw()
   print(plot_i)
}
dev.off()
```

Second, predict based on the average over the first `i` trees:

```r
L = 2000

pdf(file = "Tree-Averages.pdf", width = 8, height = 6)
i = 1
while (i <= L) {
   if (i < 2) predict_DT[, predicted_share := predict_ranger$predictions[, 1]]
   else predict_DT[, predicted_share := rowMeans(predict_ranger$predictions[, 1:i])]

   plot_i = ggplot(predict_DT, aes(x = log_zillow, y = predicted_share)) +
               geom_line(color = "purple", size = 1) +
               scale_y_continuous(paste0("PL share, trees: ", i)) +
               theme_bw()
   print(plot_i)

   cat(i, "\n")
   i = i + 1 + floor(sqrt(i)/5)
}
dev.off()
```