Causal Forest

Günter J. Hitsch

February 19, 2017

Installation

The `causalTree` package is actively being developed and it needs to be installed and compiled from sources. This is not hard, but you first need to install the necessary development environment.

Step 1

Windows: Download and install Rtools.

Mac OS: Download `Xcode` from the App Store. Warning: `Xcode` may take more than one hour to download. Then launch the `Xcode` application and accept the license terms. Open the `Terminal` (command line) application (installed on any Mac and located in `/Applications/Utilities`). In the `Terminal` window type:

```
xcode-select --install
```

An alert box should appear; click *Install* to proceed.

Step 2

Once step 1 is completed reboot your computer (just to be on the safe side). Then install the R package `devtools`.

Now you can install and build the `causalTree` package:

```r
library(devtools)
install_github("walterwzhang/causalTree")
```

Causal forest example

```
library(data.table)
library(ggplot2)
library(causalTree)
library(broom)
library(knitr)
```

We simulate a data set that includes the following variables:

- `target` is the *treatment*, a dummy variable indicating if a customer was targeted (e-mail/catalog). The treatment assignment is random.
- `spend` is observed dollar spending.
- `recency` is the customer recency status (in months), ranging from 1 to 18.
- `web_buyer` is a dummy variable that indicates if a customer is a frequent user of the company's website.

The purchase probability p takes the following form:

1. **Customers who are not targeted**. For `recency` between 1 and 6, $p = 0$. Then, for `recency` between 7 and 12, p increases in `recency` and takes the value $p = 0.03 \cdot (\texttt{recency} - 6)$. For all values of `recency` above 12, $p = 0.03 \cdot 6 = 0.18$.
2. **Customers who are targeted**
   (a) If `web_buyer`, then p is 1.25 times the baseline purchase probability.
   (b) If not `web_buyer`, then p is twice the the baseline purchase probability.

Spending *conditional* on a purchase is uniformly distributed on the values $80, 81, \ldots, 119, 120$, with a corresponding mean of 100. Hence, expected spending is $100 \cdot p$.

In this model, the treatment effect is non-linear in `recency`, and the treatment effect is larger (for `recency` above 6) for customers who are not a `web_buyer`.

```
set.seed(941)

n_obs  = 100000         # Training
n_pred = 100000         # Prediction
n = n_obs + n_pred

customer_DT = data.table(recency   = sample.int(18, size = n, replace = TRUE),
                         web_buyer = rbinom(n, 1, 1/3),
                         target    = rbinom(n, 1, 0.5))

# Define the purchase probability p
customer_DT[recency <= 6,                 p := 0.0]
customer_DT[recency >  6 & recency <= 12, p := 0.03*(recency - 6)]
customer_DT[recency > 12,                 p := 0.03*6]
customer_DT[web_buyer == 1 & target == 1, p := 1.25*p]
customer_DT[web_buyer == 0 & target == 1, p := 2*p]

# Simulate spending data
customer_DT[, purchase   := runif(n) <= p]
customer_DT[, cond_spend := 79 + sample.int(41, size = n, replace = TRUE)]
customer_DT[, spend      := purchase*cond_spend]
```

```
training_DT = customer_DT[1:n_obs]
pred_DT     = customer_DT[(n_obs+1):n]
```

Now we estimate the causal forest. Note that we need to specify the treatment variable.

Note: When first estimating a causal forest, I recommend to set the `num.trees` (number of trees) option to a small value, maybe 10, to get a sense how much computation time is involved.

The `verbose` option was added for your convenience but is not part of the original package. Set to `FALSE` if you don't want to see the output messages indicating the progress in growing the random forest.

```
fit = causalForest(spend ~ recency + web_buyer,
                   treatment = training_DT$target,
                   data = training_DT,
                   num.trees = 1000,
                   verbose = TRUE)
```

Predict spending in the prediction sample.

```
pred_DT[, pred_treatment_effect := predict(fit, pred_DT)]
```

Now we create a table with the true, observed, and predicted conditional average treatment effects (τ) for all values of `web_buyer` and `recency`.

```
summary_DT = pred_DT[, .(tau      = 100*(mean(p[target==1]) - mean(p[target==0])),
                         tau_obs  = mean(spend[target==1]) - mean(spend[target==0]),
                         tau_pred = mean(pred_treatment_effect)),
                     keyby = .(web_buyer, recency)]
```

```r
kable(summary_DT[web_buyer == 0], digits = 2)
```

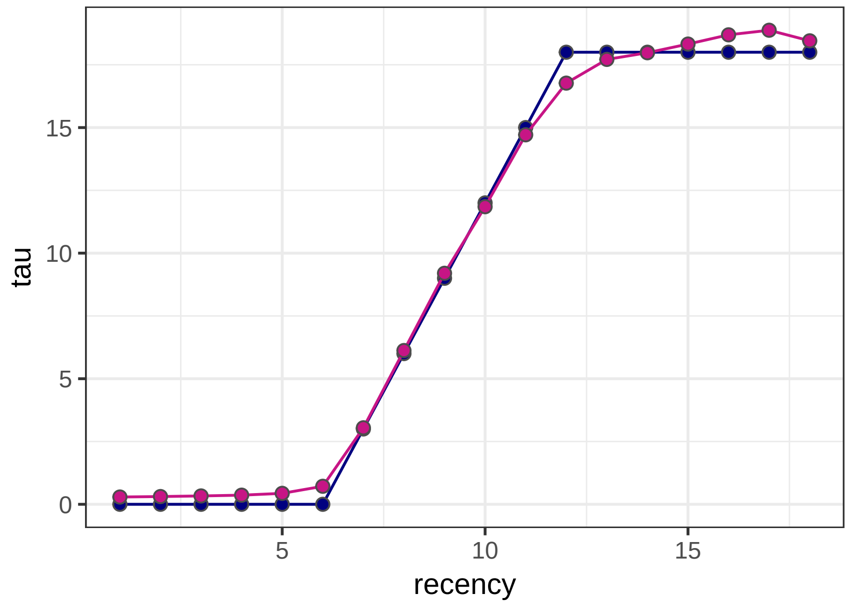
web_buyer	recency	tau	tau_obs	tau_pred
0	1	0	0.00	0.29
0	2	0	0.00	0.31
0	3	0	0.00	0.33
0	4	0	0.00	0.36
0	5	0	0.00	0.44
0	6	0	0.00	0.72
0	7	3	3.07	3.05
0	8	6	5.99	6.12
0	9	9	9.03	9.20
0	10	12	9.24	11.85
0	11	15	16.26	14.71
0	12	18	16.63	16.77
0	13	18	15.56	17.71
0	14	18	20.12	17.98
0	15	18	19.84	18.32
0	16	18	15.87	18.69
0	17	18	17.46	18.88
0	18	18	15.21	18.45

```r
kable(summary_DT[web_buyer == 1], digits = 2)
```

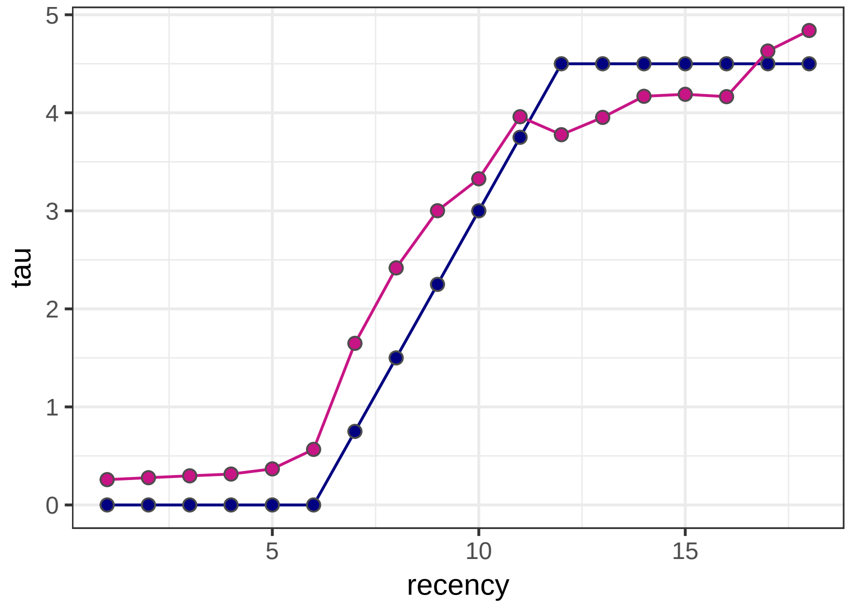
web_buyer	recency	tau	tau_obs	tau_pred
1	1	0.00	0.00	0.26
1	2	0.00	0.00	0.28
1	3	0.00	0.00	0.30
1	4	0.00	0.00	0.32
1	5	0.00	0.00	0.37
1	6	0.00	0.00	0.57
1	7	0.75	2.02	1.65
1	8	1.50	2.25	2.42
1	9	2.25	5.25	3.00
1	10	3.00	0.02	3.33
1	11	3.75	4.66	3.96
1	12	4.50	0.70	3.78
1	13	4.50	4.14	3.95
1	14	4.50	2.84	4.17
1	15	4.50	5.56	4.19
1	16	4.50	2.72	4.16
1	17	4.50	1.90	4.63
1	18	4.50	3.84	4.84

Model fit: causal forest

```r
ggplot(summary_DT[web_buyer == 0], aes(x = recency, y = tau)) +
   geom_line(color = "navyblue", size = 0.5) +
   geom_point(shape = 21, color = "gray30", fill = "navyblue", size = 2, stroke = 0.5) +
   geom_line(aes(x = recency, y = tau_pred),
             color = "mediumvioletred", size = 0.5) +
   geom_point(aes(x = recency, y = tau_pred),
              shape = 21, color = "gray30", fill = "mediumvioletred", size = 2, stroke = 0.5) +

   theme_bw()
```




```r
ggplot(summary_DT[web_buyer == 1], aes(x = recency, y = tau)) +
   geom_line(color = "navyblue", size = 0.5) +
   geom_point(shape = 21, color = "gray30", fill = "navyblue", size = 2, stroke = 0.5) +
   geom_line(aes(x = recency, y = tau_pred),
             color = "mediumvioletred", size = 0.5) +
   geom_point(aes(x = recency, y = tau_pred),
              shape = 21, color = "gray30", fill = "mediumvioletred", size = 2, stroke = 0.5) +

   theme_bw()
```

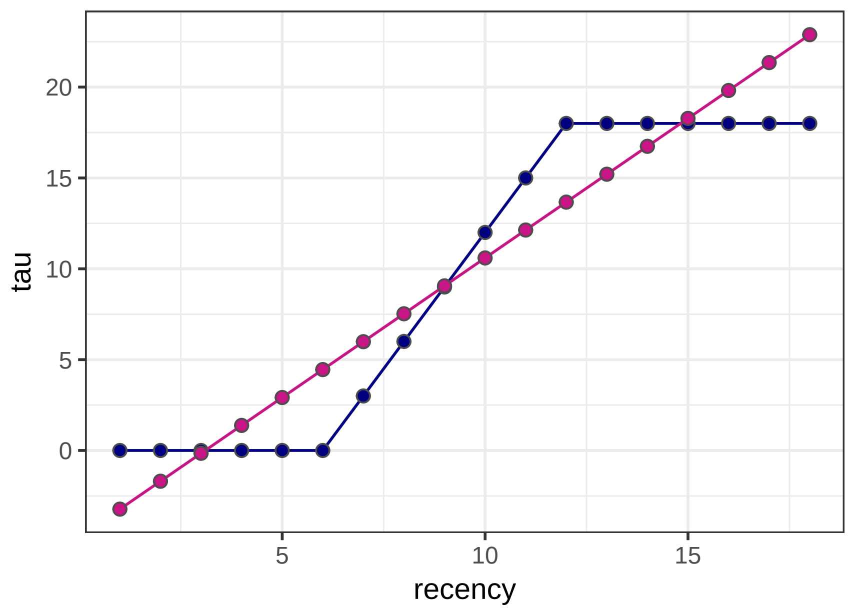


5
4
3
2
1
0
tau
5
10
15
recency

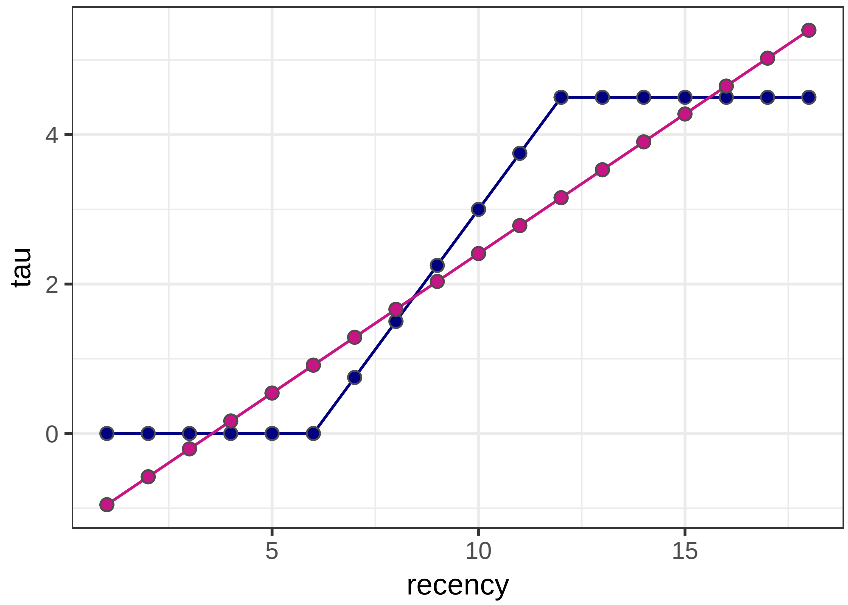
Model fit: OLS

```
minimal_DT = training_DT[, .(spend, recency, web_buyer, target)]

fit_OLS = lm(spend ~ . + .*target + web_buyer:recency*target,
             data = minimal_DT)

pred_DT[, pred_spend_OLS := predict(fit_OLS, pred_DT)]

summary_OLS_DT = pred_DT[, .(tau_pred_OLS = mean(pred_spend_OLS[target==1])
                                          - mean(pred_spend_OLS[target==0])),
                          keyby = .(web_buyer, recency)]
summary_OLS_DT = merge(summary_OLS_DT, summary_DT[, .(web_buyer, recency, tau)],
                       by = c("web_buyer", "recency"))
```

```
ggplot(summary_OLS_DT[web_buyer == 0], aes(x = recency, y = tau)) +
   geom_line(color = "navyblue", size = 0.5) +
   geom_point(shape = 21, color = "gray30", fill = "navyblue", size = 2, stroke = 0.5) +
   geom_line(aes(x = recency, y = tau_pred_OLS),
             color = "mediumvioletred", size = 0.5) +
   geom_point(aes(x = recency, y = tau_pred_OLS),
              shape = 21, color = "gray30", fill = "mediumvioletred", size = 2, stroke = 0.5) +

   theme_bw()
```




```
ggplot(summary_OLS_DT[web_buyer == 1], aes(x = recency, y = tau)) +
   geom_line(color = "navyblue", size = 0.5) +
   geom_point(shape = 21, color = "gray30", fill = "navyblue", size = 2, stroke = 0.5) +
   geom_line(aes(x = recency, y = tau_pred_OLS),
             color = "mediumvioletred", size = 0.5) +
   geom_point(aes(x = recency, y = tau_pred_OLS),
              shape = 21, color = "gray30", fill = "mediumvioletred", size = 2, stroke = 0.5) +

   theme_bw()
```

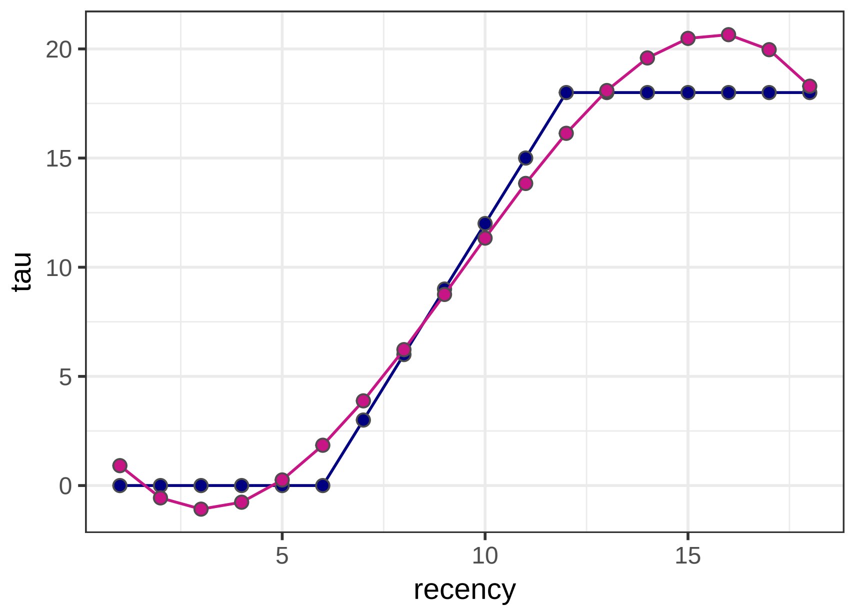


tau
4
2
0
5
10
15
recency

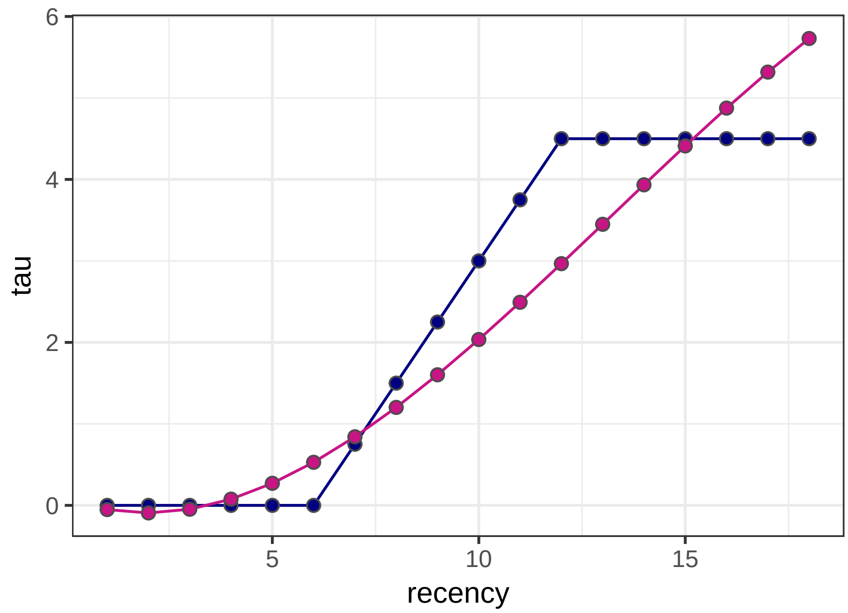
Model fit: OLS with polynomials

```
minimal_DT = training_DT[, .(spend, recency, web_buyer, target)]

fit_OLS = lm(spend ~ target*web_buyer + poly(recency,3)*web_buyer + poly(recency,3):target*web_buyer,
             data = minimal_DT)

pred_DT[, pred_spend_OLS := predict(fit_OLS, pred_DT)]

summary_OLS_DT = pred_DT[, .(tau_pred_OLS = mean(pred_spend_OLS[target==1])
                                          - mean(pred_spend_OLS[target==0])),
                         keyby = .(web_buyer, recency)]
summary_OLS_DT = merge(summary_OLS_DT, summary_DT[, .(web_buyer, recency, tau)],
                       by = c("web_buyer", "recency"))
```

```
ggplot(summary_OLS_DT[web_buyer == 0], aes(x = recency, y = tau)) +
   geom_line(color = "navyblue", size = 0.5) +
   geom_point(shape = 21, color = "gray30", fill = "navyblue", size = 2, stroke = 0.5) +
   geom_line(aes(x = recency, y = tau_pred_OLS),
             color = "mediumvioletred", size = 0.5) +
   geom_point(aes(x = recency, y = tau_pred_OLS),
              shape = 21, color = "gray30", fill = "mediumvioletred", size = 2, stroke = 0.5) +

   theme_bw()
```




```
ggplot(summary_OLS_DT[web_buyer == 1], aes(x = recency, y = tau)) +
   geom_line(color = "navyblue", size = 0.5) +
   geom_point(shape = 21, color = "gray30", fill = "navyblue", size = 2, stroke = 0.5) +
   geom_line(aes(x = recency, y = tau_pred_OLS),
             color = "mediumvioletred", size = 0.5) +
   geom_point(aes(x = recency, y = tau_pred_OLS),
              shape = 21, color = "gray30", fill = "mediumvioletred", size = 2, stroke = 0.5) +

   theme_bw()
```




```
out = tidy(summary(fit_OLS))
kable(out, digits = 2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	9.44	0.17	54.05	0.00
target	9.89	0.25	40.07	0.00
web_buyer	0.37	0.30	1.22	0.22
poly(recency, 3)1	2294.87	55.10	41.65	0.00
poly(recency, 3)2	-247.92	55.22	-4.49	0.00
poly(recency, 3)3	-757.54	55.19	-13.73	0.00
target:web_buyer	-7.69	0.43	-17.98	0.00
web_buyer:poly(recency, 3)1	55.59	95.89	0.58	0.56
web_buyer:poly(recency, 3)2	-64.82	96.03	-0.68	0.50
web_buyer:poly(recency, 3)3	-46.48	95.56	-0.49	0.63
target:poly(recency, 3)1	2529.19	78.01	32.42	0.00
target:poly(recency, 3)2	-46.54	78.02	-0.60	0.55
target:poly(recency, 3)3	-739.05	78.15	-9.46	0.00
target:web_buyer:poly(recency, 3)1	-1919.95	135.45	-14.17	0.00
target:web_buyer:poly(recency, 3)2	154.00	135.43	1.14	0.26
target:web_buyer:poly(recency, 3)3	695.13	135.20	5.14	0.00