Statistics and ML

MSSP Practicum Discussion

Priam Vyas

2023-01-24

Instructions

Fork the `carvalho/stats-ml-practicum` repository at GitHub, and **create a new branch with your BU login** to store your changes to the document. Start by changing the `author`in the YAML header of the document to state **your name**.

Below we run some analyses and ask questions about them. As you run the code and interpret the results within your group, write your answers to the questions following the analyses, but:

> You should submit your work as a **pull request** to the original repository!

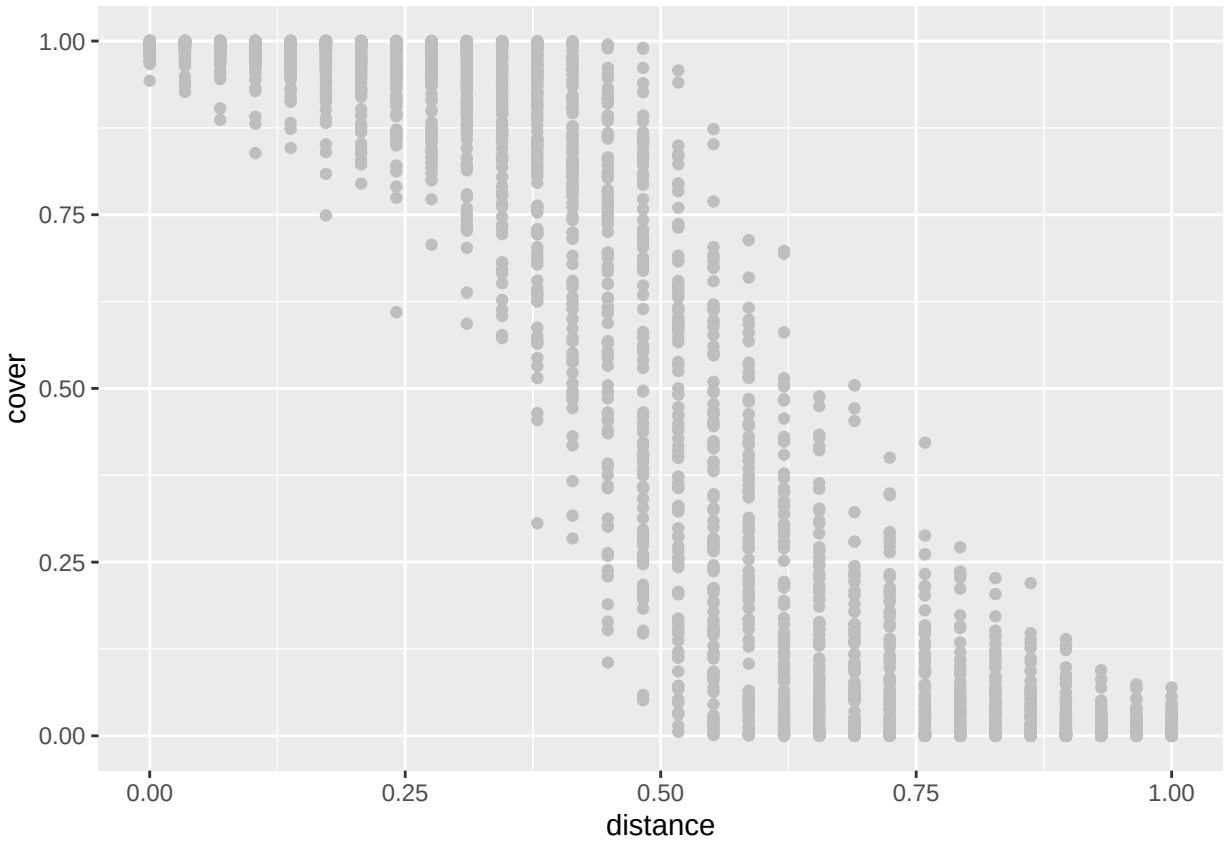
Introduction

In this project we study **tree canopy cover** as it varies with the **relative distance** to a tree line boundary in urban forests. The dataset in `stats-ml-canopy.RData` has three variables: `location` for the urban forest where the canopy cover was observed, `distance` for the relative distance — zero is inside the forest and one is outside (city) — and `cover` for the canopy cover.

```
load("stats-ml-canopy.RData")
(canopy <- as_tibble(canopy))
```

```
## # A tibble: 3,000 x 3
##    location distance cover
##    <fct>       <dbl> <dbl>
##  1 1           0      1.00
##  2 1           0.0345 1.00
##  3 1           0.0690 1.00
##  4 1           0.103  1.00
##  5 1           0.138  1.00
##  6 1           0.172  1.00
##  7 1           0.207  1.00
##  8 1           0.241  0.999
##  9 1           0.276  0.998
## 10 1           0.310  0.993
## # ... with 2,990 more rows
```

```
idx <- order(canopy$distance) # for plots below
ggplot(canopy, aes(distance, cover)) + geom_point(color = "gray")
```



As can be seen, there is a clear pattern here: the canopy cover starts high, closer to 100% when inside the forest, but as the tree line recedes into the city, the canopy cover approaches zero.

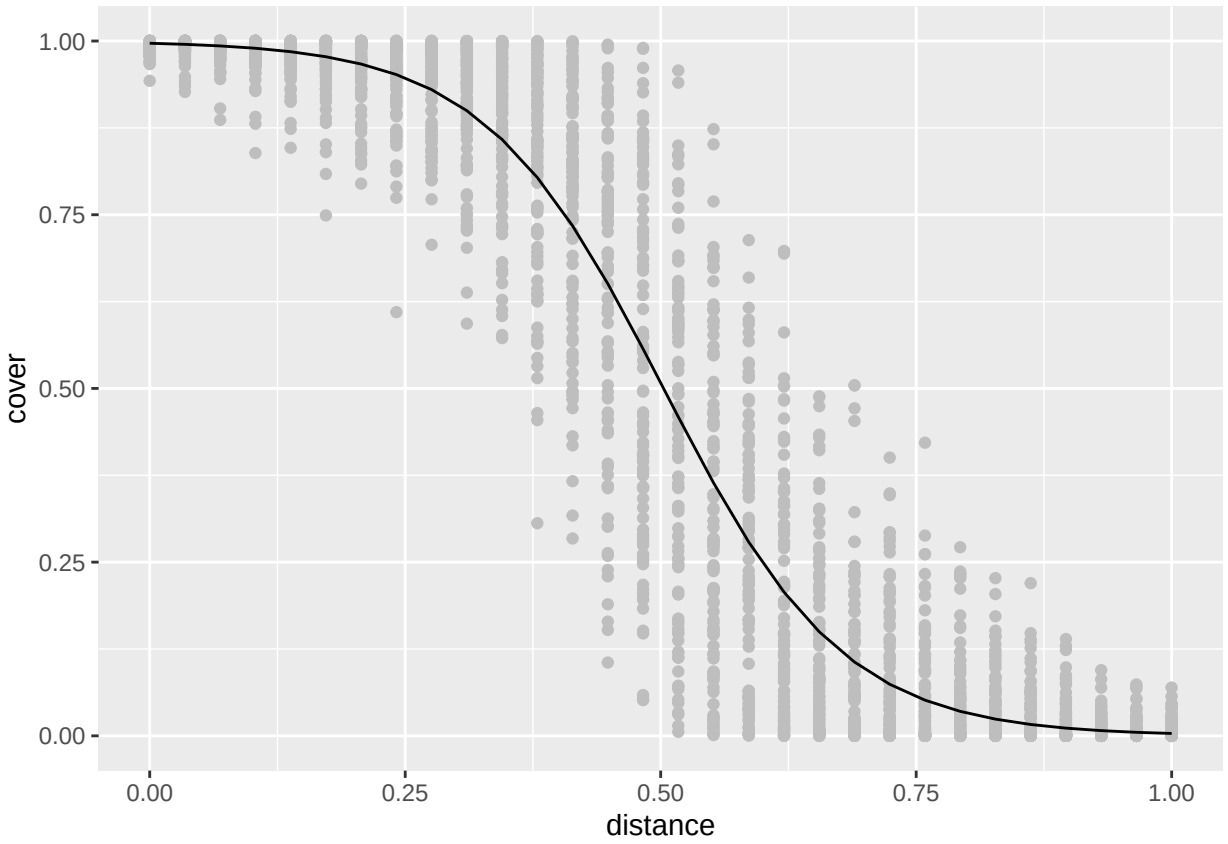
We are interested in two main tasks:

- **Understanding** this relationship more explicitly;
- **Predicting** the canopy cover at the assumed tree line boundary when `distance` is 0.5.

To this end, we explore four approaches below.

Statistics 1: Linear Fit

```
m <- glm(cover ~ distance, data = canopy, family = quasibinomial)
ggplot(canopy, aes(distance, cover)) + geom_point(col = "gray") +
  geom_line(aes(distance[idx], fitted(m)[idx]))
```




```
predict(m, data.frame(distance = 0.5), se = TRUE, type = "response")
```

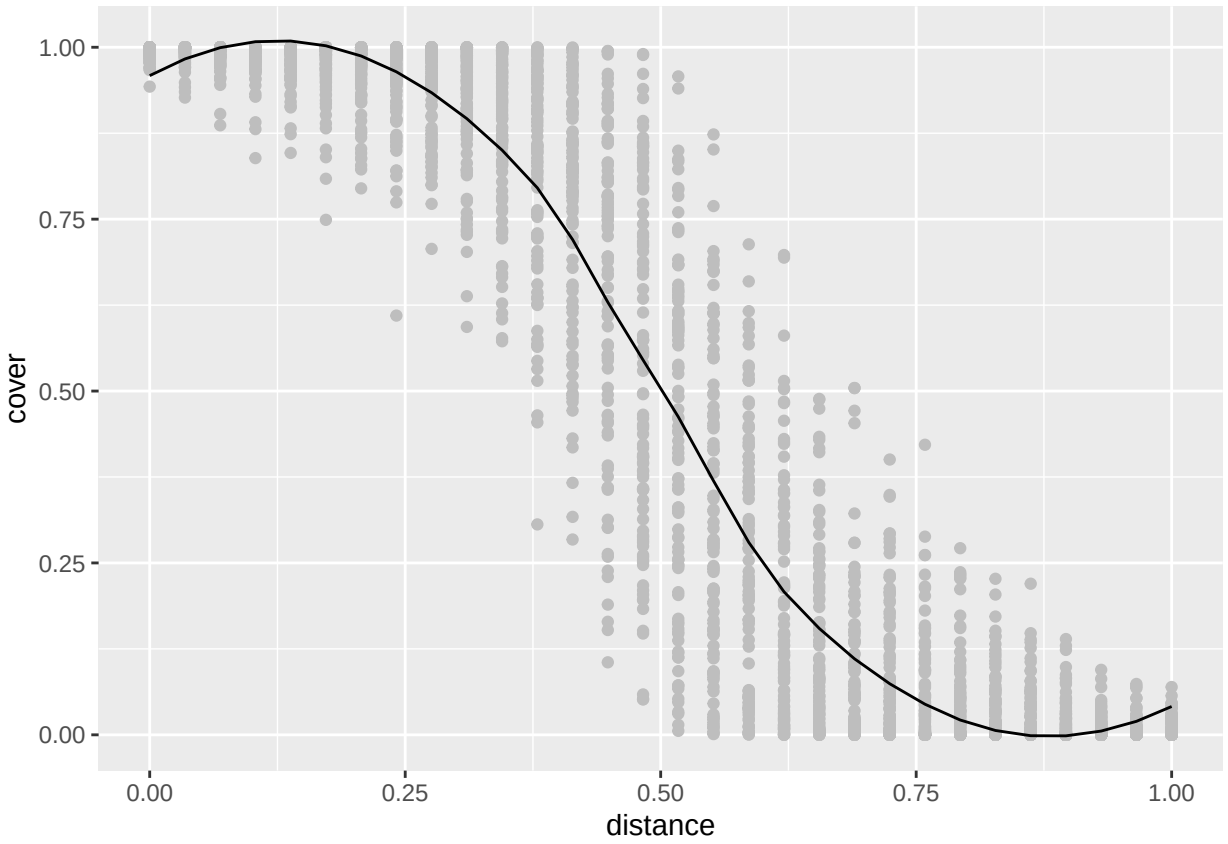
```
## $fit
##         1
## 0.5080519
##
## $se.fit
##           1
## 0.005392449
##
## $residual.scale
## [1] 0.343108
```

Questions and tasks:

- Comment on the fit, plot residuals and comment on them.
- Comment on the prediction; does it seem reasonable?

ML 1: LOESS

```
m <- loess(cover ~ distance, data = canopy)
ggplot(canopy, aes(distance, cover)) + geom_point(col = "gray") +
  geom_line(aes(distance[idx], fitted(m)[idx]))
```




```
predict(m, data.frame(distance = 0.5), se = TRUE)
```

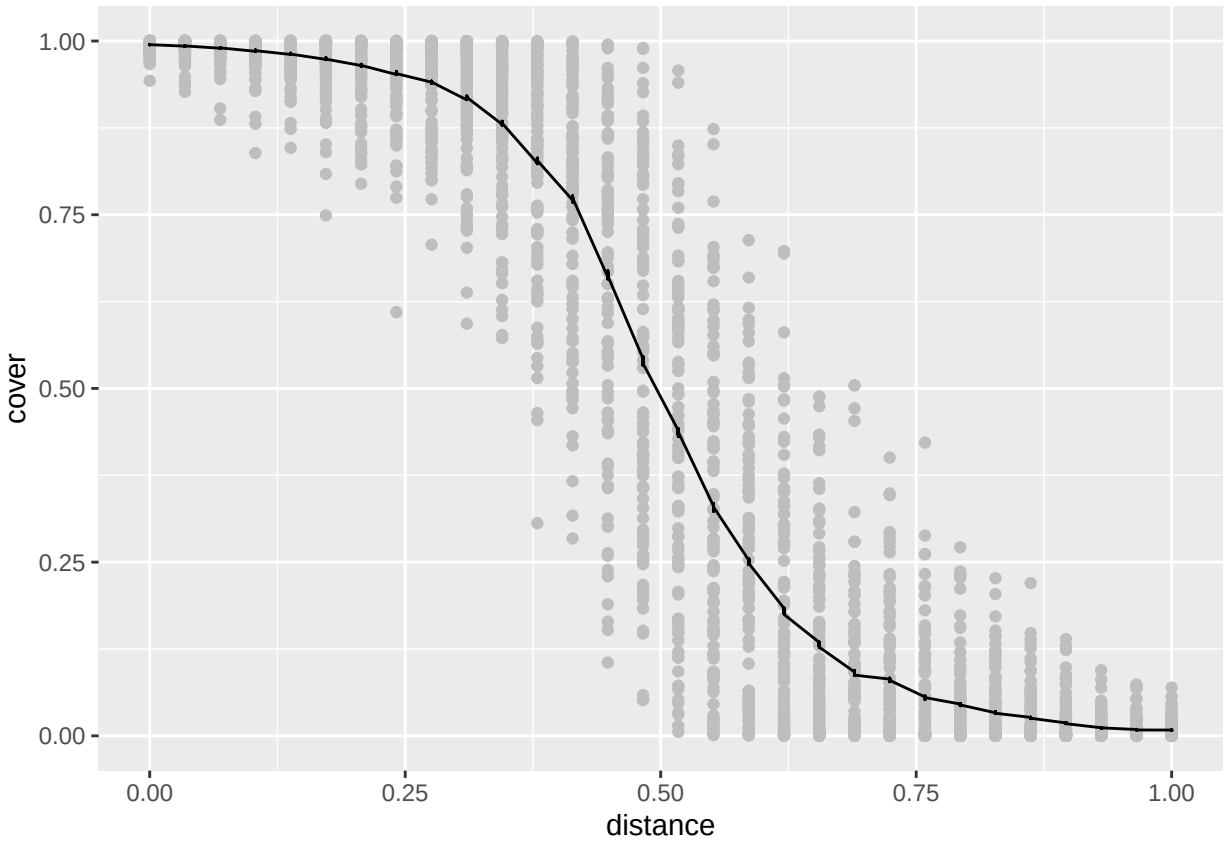
```
## $fit
##         1
## 0.505973
##
## $se.fit
##           1
## 0.004378154
##
## $residual.scale
## [1] 0.1229851
##
## $df
## [1] 2995.204
```

Questions and tasks:

- Check the definition of the `loess` function; how does it differ from the previous approach?
- Comment on the fit; does it seem reasonable?
- Comment on the prediction, including the SE.

ML 2: Random Forest

```
library(randomForest)
m <- randomForest(cover ~ distance, data = canopy)
ggplot(canopy, aes(distance, cover)) + geom_point(col = "gray") +
  geom_line(aes(distance[idx], predict(m)[idx]))
```




```
predict(m, data.frame(distance = 0.5), se = TRUE)
```

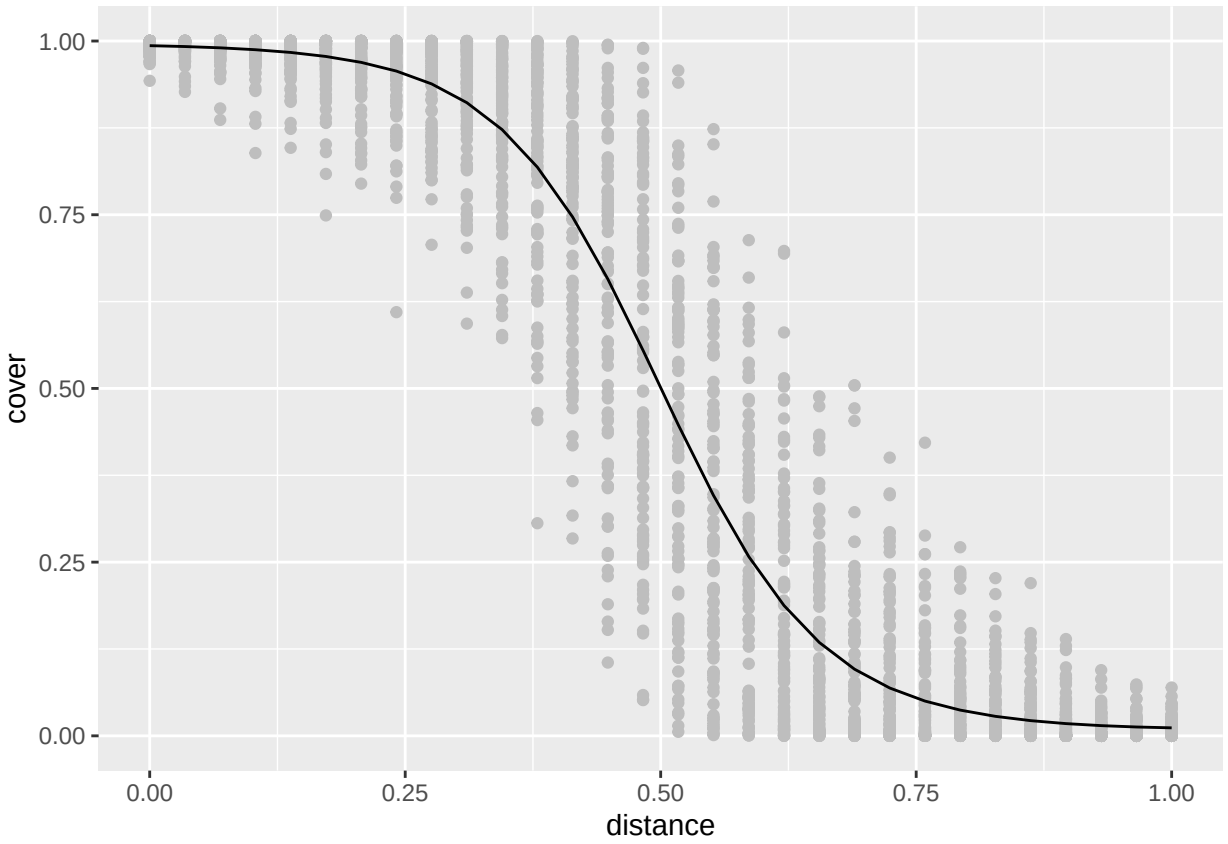
```
##         1
## 0.5389118
```

Questions and tasks:

- Check what `randomForest` does; what is **keyword** here?
- Comment on the fit; how does it differ from the previous fits?
- Comment on the prediction; how would you obtain a measure of uncertainty?

Statistics 2: Cubic Fit

```
m <- glm(cover ~ poly(distance, 3), data = canopy, family = quasibinomial)
ggplot(canopy, aes(distance, cover)) + geom_point(col = "gray") +
  geom_line(aes(distance[idx], fitted(m)[idx]))
```




```
predict(m, data.frame(distance = 0.5), se = TRUE, type = "response")
```

```
## $fit
##         1
## 0.5010702
##
## $se.fit
##           1
## 0.006254468
##
## $residual.scale
## [1] 0.3356464
```

Questions and tasks:

- Comment on the fit and compare it to the first model; plot and check residuals.
- Comment on the prediction and compare it to previous results.
- How would you know that a cubic fit is good enough?

Discussion

Let's try to connect all lessons learned from your work and the discussions. Elaborate more on the following questions:

- How would you know that the predictions are *reliable*?
- How would you test that the cover is exactly 50% at the boundary (`distance` = 0.5)? Which approaches would make the test easier to perform?
- How would you incorporate `location` in your analyses? How would you know that it is meaningful to use it?