Transforming the prediction scale

Brandon M. Greenwell

May 11, 2022

Introduction

As of version 0.6.0, **pdp**'s `partial()` function supports the ability to transform the predictions to a different scale via the `inv.link` argument. This (optional) argument takes a function specifying the transformation to be applied to the predictions before the partial dependence function is computed (experimental)l the default is `NULL` (i.e., no transformation). This option is intended to be used for models that allow for non-Gaussian response variables (e.g., counts). For these models, predictions are not typically returned on the original response scale by default. For example, Poisson GBMs typically return predictions on the log scale. In this case setting `inv.link = exp` will return the partial dependence function on the response (i.e., raw count) scale.

```
library(ggplot2)
library (magrittr)
library(pdp)
library(xgboost)

# Set ggplot2 theme
theme_set(theme_bw())

# Fit a simple XGBoost model
set.seed(101)
bst <- xgboost(data = as.matrix(mtcars[, -11]), label = mtcars[, 11],
               objective = "count:poisson", nrounds = 50, verbose = 0)
```

The default...

```
bst %>%  # figure 1
  partial(pred.var = "mpg", train = mtcars[, -11]) %>%
  autoplot() +
  labs(x = "Miles per hour (MPG)", y = "Number of carburetors")
```

If you'd like the y-axis to reflect the original (i.e., count or "inverse link") scale, then you have two options:

- construct a prediction wrapper that computes the average predicted probability on the scale of interest;
- pass a suitable function to the `inv.link` argument.

Both of these approaches are illustrated in the code chunk below. Note that you can also pass in a string to `inv.link` (e.g., `inv.link = "exp"`).

```
# Prediction function that returns the (average) prediction on the original
# response scale
pfun <- function(object, newdata) {
  mean(exp(predict(object, newdata = as.matrix(newdata))))
}
```

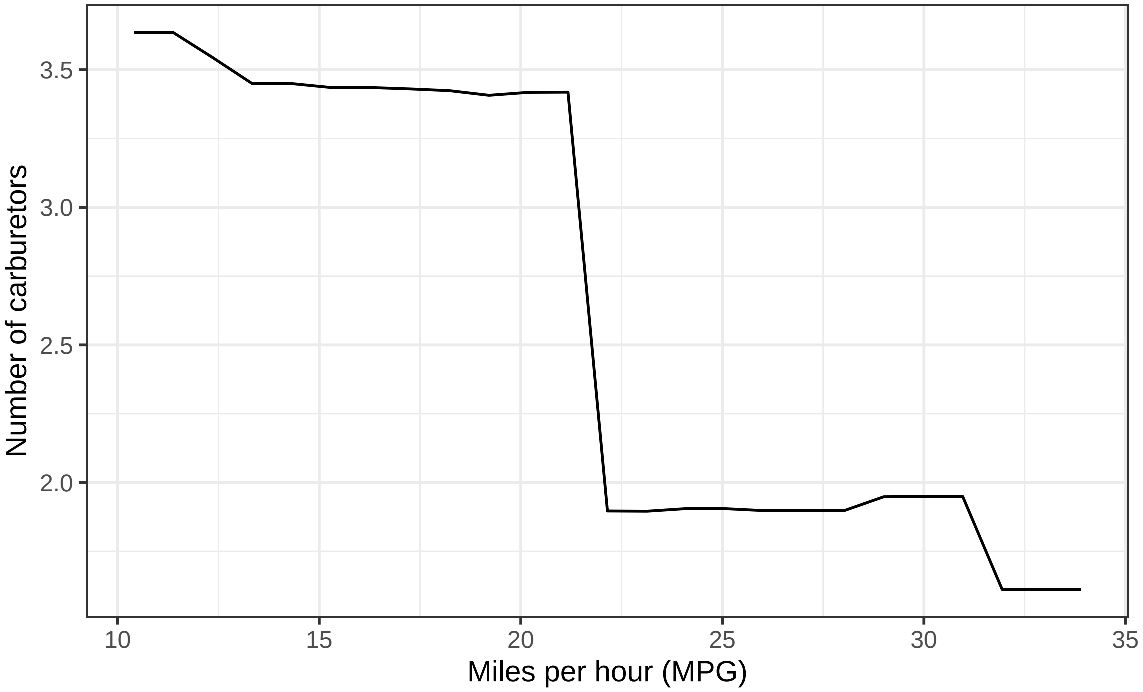

Figure 1: Partial dependence plot of MPG on the log number of carburetors (i.e., the link scale). By default, the y-axis is on the link scale (the log scale, in this case).

```
# Passing a user-supplied prediction wrapper
p1 <- bst %>%
  partial(pred.var = "mpg", pred.fun = pfun, train = mtcars[, -11]) %>%
  autoplot() +
  labs(x = "Miles per hour (MPG)", y = "Number of carburetors")

# Using `inv.link` argument
p2 <- bst %>%
  partial(pred.var = "mpg", inv.link = exp, train = mtcars[, -11]) %>%
  autoplot() +
  labs(x = "Miles per hour (MPG)", y = "Number of carburetors")

# Display PDPs side by side
gridExtra::grid.arrange(p1, p2, nrow = 1)  # figure 2
```

A related example involving logistic regression can be found here: https://github.com/bgreenwell/pdp/issues/125. In this example, it's shown how to produce PDPs for a logistic regression model on the usual logit scale (as opposed to the default class-centered logit described in Greenwell [2017]) and then using the `inv.link` argument to produce a PDP on the probability scale; as of version 0.5.0, **pdp** can generate PDPs for any supported classification model on the probability scale by just setting `prob = TRUE` in the call to `partial()`.

References

Brandon M. Greenwell. pdp: An R Package for Constructing Partial Dependence Plots. *The R Journal*, 9(1): 421–436, 2017. doi: 10.32614/RJ-2017-016. URL https://doi.org/10.32614/RJ-2017-016.

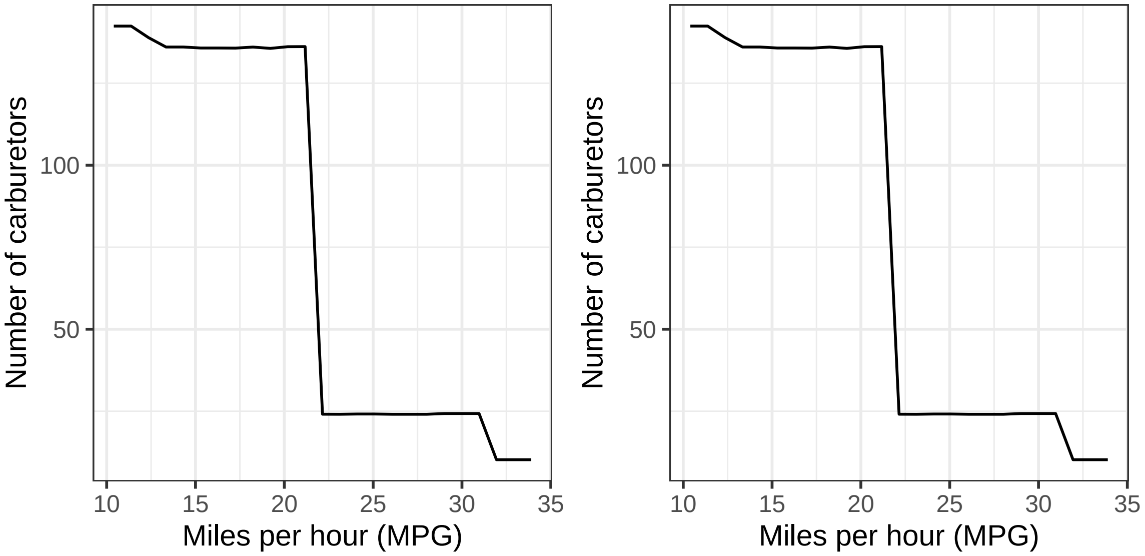

Figure 2: Partial dependence plot of MPG on the number of carburetors (i.e., original or inverse link scale). Left: using a user-supplied prediction wrapper. Right: passing a suitable inverse link function via the ‘inv.link‘ argument.