Feature Importance

David S. Rosenberg

NYU: CDS

April 15, 2021

Contents

Introduction

Our approach to model interpretation

- "Model interpretation" is not a well-defined problem.
- Our goal will be to try to get some easy-to-understand explanation for how
  - our prediction function depends on each feature (or small sets of features).
- We'll discuss:
  - partial dependence plots (PDP)
  - individual conditional expectation (ICE) plots
- And we'll discuss caveats for the use of each.

Feature Importance

- Feature importance is closely related to model interpretation..
- Also not particularly well-defined.
- Many modeling packages give back "feature importance" scores for features.
- These are usually model specific, e.g.
  - Linear methods: absolute value of weights, p-values
  - Trees: Mean decrease in impurity
    - weighted combinations for tree ensembles
- We'll discuss methods that are model agnostic:
  - permutation feature importance
  - leave one covariate out (LOCO) importance
- As well as issues and limitations.

Why bother with this stuff?

- Feature selection / domain understanding
- Sanity checking models
  - detection leakage
  - detecting dependencies that won't generalize
- Fairness considerations
  - detecting dependencies that you **don't want**
- Explaining predictions

Permutation feature importance

Breiman's random forest paper (2001)

Machine Learning, 45, 5–32, 2001

Random Forests

LEO BREIMAN
Statistics Department, University of California, Berkeley, CA 94720

- Leo Breiman introduced random forests in [Bre01] (tech report from Sept 1999).
- Still one of the most widely used machine learning methods.
- In a couple pages in Section 10, Breiman describes a procedure that is now referred to as **permutation feature importance**.

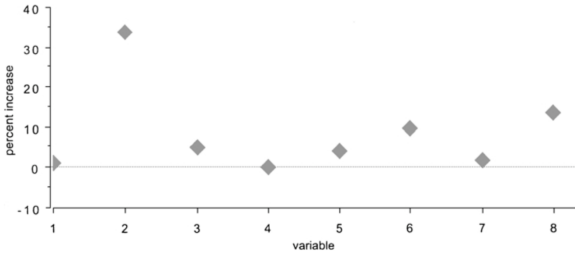
Permutation feature importance

- [Bre01, Sec 10] describes method specifically for random forest.
  - Based on out-of-bag predictions. (A special attribute of RF models.)
- Here we'll present the obvious generalization.
- Given:
  - Prediction function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_d \to \mathcal{A}$. (Arbitrary type for each of d features.)
  - Data set $\mathcal{D} = (x_1, \ldots, x_d)$, where $x_i \in \mathcal{X}_i^n$ is the ith feature "column".
  - Performance metric: $m : (f, \mathcal{D}) \to \mathbb{R}$
- Let $\mathcal{D}_i$ be $\mathcal{D}$ with x_i replaced by Shuffle(x_i).
- **Permutation feature importance** of feature i is some comparison between $m(f, \mathcal{D})$ and $m(f, \mathcal{D}_i)$
  - e.g. $m(f, \mathcal{D})/m(f, \mathcal{D}_i)$ or $m(f, \mathcal{D}) - m(f, \mathcal{D}_i)$ or $(m(f, \mathcal{D}_i) - m(f, \mathcal{D}))/m(f, \mathcal{D})$.

Discussion

- Shuffling feature i
  - breaks any relationship between x_i and the response $f(x)$
  - also breaks any relationship between any interaction feature $g(x_i, x_j)$ and $f(x)$
  - may also take us "off the data manifold" or "out of distribution" (more on this later)
- Some theory on what permutation feature importance corresponds to for additive models can be found in [GMSP15, Sec 2 and App A].

Breiman's example (I)



- Percent increase in misclassification rate from shuffling each variable.
- Variable 2 seems most important.
- Followed by 8 and then 6.

Figure 4 from [Bre01, p. 24].

Breiman's example (II)

- Let's fit new models using different variable subsets.
- All variables: 23.1% error
- Only variable 2: 29.7% error
- Only variables 2 and 8: 29.4% error
- Conclusion: variable 8 doesn't add much information to what variable 2 provides
  - (at least that could be extracted by the random forest used for the model)
- Variables 2 and 6: 26.4% error

Discussion

- So what does permutation importance tell us?
- First: it's a result about **the prediction function itself**.
- The results may change substantially if
  - we have a different random training set, or slightly perturbed data
  - we use a different ML model or training algorithm
- With a rather literal reading of permutation importance, it tells us
  - the effect of replacing a particular input value with a random draw from the marginal distribution of that variable.
- Perhaps we can take this as a proxy for the sensitivity to errors in that variable.
  - errors of a very particular distribution

Leave-one-covariate out (LOCO)

- Suppose we want to see if x_1 is an important features.
- We do the following:
  - Remove x_1 from the training set.
  - Retrain the model.
  - Compare performance of model with x_1 and without x_1.
- This is a "leave-one-covariate-out" or LOCO method [LGR+18, Sec 6].
- What's different compared to the feature importance methods described previously?

Feature importance for what?

- We can think of the importance of a feature for
  - a particular prediction function $\hat{f}(x)$.
  - a particular learning method (e.g. linear models or random forest with particular hyperparameters)
  - the Bayes optimal prediction function
- Suppose $x_1 = g(x_2)$ for some nonlinear function g.
- If $\hat{f}(x)$ uses x_1 but not x_2, then x_1 is important for $\hat{f}$ and x_2 is not.
  - Permutation importance sampling will detect this.
- If our learning method cannot extract the required information from x_2 but it can from x_1,
  - LOCO would identify x_1 is important but not x_2.
- The Bayes optimal prediction function will do at least as well with just x_2 than just x_1.

Feature importance in prediction function vs in problem

- Variable importance for the problem in general:
  - without x_1 no prediction function can be good
  - (or at least not with our training method – perhaps a more powerful hypothesis space could do better)
  - LOCO methods are the most direct way to get to this.
- vs.
  - This particular prediction function needs x_1 to get good performance.
  - This is what permutation importance is getting at most directly.
- In practice, LOCO is usually much more expensive than permutation importance
  - Requires retraining rather than just another evaluation on the test set.

How to figure out which features to buy/generate?

- Suppose we're in a world where using more features has a cost
  - e.g. to buy, to compute, to keep in memory, to reduce training time
- This seems like a problem for LOCO.
- If we LOCO is too expensive, can try permutation feature importance as an approximation
  - but this has issues, as discussed below...
- If features come in groups, leave out a group at a time.
  - we can do this with permutation importance as well.
  - This is called a **grouped variable importance measure** in [GMSP15, Sec 2]

What can go wrong with permutation?

Hooker and Mentch paper (2019)

Please Stop Permuting Features
An Explanation and Alternatives

Giles Hooker*and Lucas Mentch†

- This paper [HM19] has a catchy title.
- It "advocates against permute-and-predict (PaP) methods for interpreting black box functions."
- We need to dig in a bit to see when and why PaP methods may not be a good idea.

Main argument in paper

1. Even when prediction function $\hat{f}$ is an excellent approximation to $\mathbb{E}[Y \mid X = x]$ on the training distribution, it may be quite **different** out of distribution (not "bad", because we usually don't care what happens outside of the training distribution.)
2. **When features are not independent**, PaP methods can end up evaluating $\hat{f}$ on many out-of-distibution points.
3. The PaP importance measures can be influenced significantly by what $\hat{f}$ does out-of-distribution.
4. For **nonlinear** $\hat{f}$, behavior can be quite different in-distribution and out-of-distribution.

Possible mitigations

- If we can identify groups of dependent features, we can permute them together.
  - Reduces out of sample evaluation.
- Suppose there's reason to believe (for scientific reasons, say) that
  - our model $\hat{f}(x)$ is a good **causal** model.
  - in other words, it's reliable even "out-of-distribution"
- If we're interested in $\hat{f}(x)$ to predict the response to interventions,
  - which are almost definitionally "out-of-distribution",
  - then perhaps this criticism seems less relevant.

Partial dependence

Friedman's gradient boosting paper (1999/2001)

The Annals of Statistics
2001, Vol. 29, No. 5, 1189–1232

1999 REITZ LECTURE

GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE[1]

BY JEROME H. FRIEDMAN

Stanford University

- Jerry Friedman presented gradient boosting in 1999 (published 2001).
- The foundation of xgboost, lightGBM, catboost, and other SOTA tree-based methods.
- Gradient boosting is for learning a non-linear function $f : \mathbb{R}^d \to \mathbb{R}$.
- If $d = 1$ or $d = 2$, we can interpret f by visualization.
- In [Fri01, Sec 8.2], Friedman presents **partial dependence plots**
  - as a way to use visualization to help interpret **high-dimensional** non-linear functions.

The partial dependence function

- Suppose $f : \mathbb{R}^d \to \mathbb{R}$.
- Let $S \subset \{1, \ldots, d\}$ index a target subset of features.
- Let $C = \{1, \ldots, d\} - S$ index the complement set of features.
- We'll write $X_S \in \mathbb{R}^{|S|}$ for the subset of features indexed by S.
- Similarly for $X_C \in \mathbb{R}^{|C|}$.
- The **partial dependence function** of f on x_S is given by

$$f_S(x_S) = \mathbb{E}_{X_C}[f(x_S, X_C)] = \int f(x_S, x_C)\, dP(x_C).$$

- In words, f_S is the function f after integrating out over X_C
  - with respect to the **marginal distribution** of X_C.

Estimating the partial dependence function

- The obvious estimate of the partial dependence function is

$$\hat{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^{n} f(x_S, x_{Ci}),$$

  where $(x_{C1}, \ldots, x_{Cn})$ are the n instantiations of x_C in a dataset $\mathcal{D}$.
- This can be used on any "black box" prediction function.
- When S consists of just one or two features, we can visualize $\hat{f}_S(x_S)$.
- Fix a set of points P of points in the domain of x_S
  - e.g. the observed values of x_S themselves in dataset $\mathcal{D}$.
- Then plot $\left\{ \left(x_S, \hat{f}_S(x_S) \right) : x_S \in P \right\}$.

PDPs for 1-dimensional dependencies

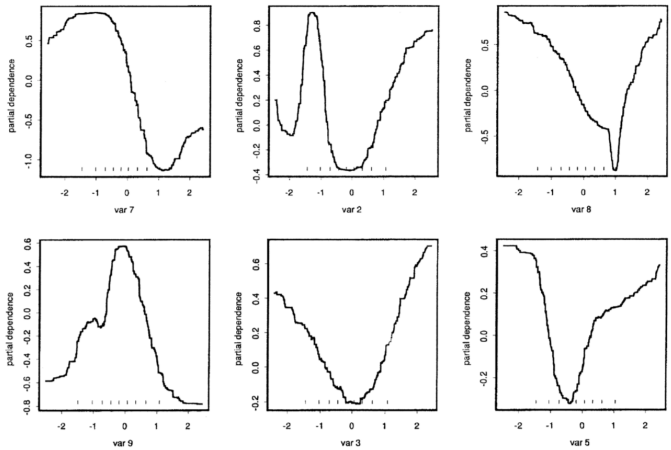


Figure 8 from [Fri01, p. 1223].

PDPs for 2-dimensional dependencies

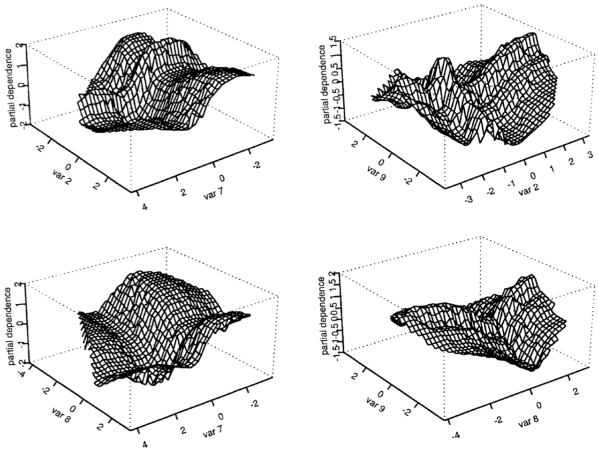


Figure 9 from [Fri01, p. 1224].

Suppose f is additive

- Suppose the function f decomposes additively as

$$f(x_S, x_C) = h_S(x_S) + h_C(x_C),$$

  for some arbitrary h_S and h_C.
- Then the partial dependence function is

$$\begin{aligned} f_S(x_S) &= \mathbb{E}_{X_C}[h_S(x_S) + h_C(X_C)] \\ &= h_S(x_S) + \mathbb{E}[h_C(X_C)] \end{aligned}$$

- Thus $f_S(x_S) - h_S(x_S) = \mathbb{E}[h_C(X_C)]$.
- Since $\mathbb{E}[h_C(X_C)]$ is a constant independent of x_S,
  - the partial dependence function recovers $h_S(x_S)$ up to an additive constant.

Additive f and GAMs

- Suppose the function f decomposes additively as

$$\begin{aligned} f(x_S, x_C) &= h_S(x_S) + h_C(x_C) \\ &= f_S(x_S) + h_C(x_C) + \text{constant} \end{aligned}$$

- Then the partial dependence $f_S(x_S)$ tells us
  - how $f(x_S, x_C)$ changes as a function of x_S with x_C fixed.
- So partial dependence plots are most useful when we can
  - f decomposes additively on small subsets of features (ideally 1 or 2 features).
- Note that generalized additive models (GAMs) are exactly of this form [HT86]...

Generalized additive models

- A generalized additive model is a model that looks like

$$
\begin{aligned}
f(x) &= f_1(x_1) + f_2(x_2) + \cdots + f_d(x_d) \\
\text{OR:} \quad f(x) &= f_1(x_1) + f_{2,3}(x_2, x_3) + f_4(x_4) + \cdots + f_d(x_d) \\
\text{MORE GENERALLY: } f(x) &= \sum_{i=1}^{G} f_i(x_{S_i}),
\end{aligned}
$$

  where each x_{S_i} is a subset of features – usually just one feature, sometimes two features.
- The f_i's are estimated from the data.
- GAMs are often considered "white box" ML models, because they are directly interpretable.
  - Basically by looking at the PDPs that **exactly** represent the GAM.

Interpretability, PDPs, and GAMs

- Using PDPs to explain a model fits into a broader approach to "interpretability".
- Broader approach:
  1. Find a black box model $f(x)$ that has the performance you like.
  2. Approximate $f(x)$ with $\hat{f}(x)$ using some "interpretable" ML model.
  3. Claim that $f(x)$ is basically $\hat{f}(x)$, and so we now can interpret $f(x)$.
- The most common "white box" models used for this are
  - shallow regression trees and
  - GAMs (via PDPs)

Not the partial dependence function

- The **partial dependence function** of f on x_S is given by

$$f_S(x_S) = \mathbb{E}_{X_C}[f(x_S, X_C)] = \int f(x_S, x_C)\, dP(x_C).$$

- Note that this is **different from**

$$f_S(x_S) = \mathbb{E}[f(X_S, X_C) \mid X_S = x_S] = \int f(x_S, x_C)\, dP(x_C \mid x_S).$$

- When might we prefer one or the other?

- One thing worth noting is that, in the special case that f decomposes additively, the partial dependence function can recover that additive decomposition, while the conditional form will not.
- What happens if $f(x) = x_1$ but X_1 and X_2 are highly correlated? Suppose that we have access to the true $f(x)$, or that we have managed to fit it very well. Then

$$f_2(x_2) = \mathbb{E}_{X_1}[f(X_1, x_2)] = \mathbb{E}_{X_1} X_1,$$

which is a constant, showing no partial dependence on x_2. On the other hand, suppose we take the conditional form, and suppose that X_1 and X_2 are jointly distributed Gaussian random variables with identical marginal distributions $\mathcal{N}(0,1)$ and correlation ρ. Then $\mathbb{E}[X_1 \mid X_2] = \rho X_2$. So

$$\begin{aligned} \mathbb{E}[f(X_1, x_2) \mid X_2 = x_2] &= \mathbb{E}[X_1 \mid X_2 = x_2] \\ &= \rho x_2. \end{aligned}$$

Now we see a linear dependence on x_2.

- Are both results meaningful?

Nonadditive example

- Consider the following distribution

$$Y = 0.2X_1 - 5X_2 + 10X_2\mathbb{1}[X_3 \geqslant 0] + \varepsilon,$$

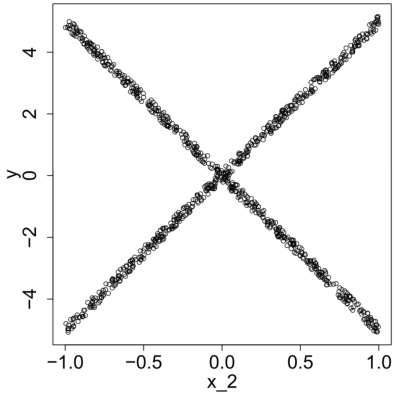
  where $\varepsilon \sim \mathcal{N}(0,1)$ and independent of X_1, X_2, X_3, which i.i.d. Unif$(-1,1)$.
- A gradient boosting model is fit to an i.i.d. sample of size 1000 from this distribution.
- Note that $f(x) := \mathbb{E}[Y \mid X = x]$ does not decompose additively for X_2 or X_3.
- The partial dependence function for f on x_2 is

$$\begin{aligned} f_2(x_2) = \mathbb{E}_{X_{\{1,3\}}}[f(x_2, X_1, X_3)] &= \mathbb{E}[0.2X_1] - 5x_2 + 10x_2\mathbb{P}(X_3 \geqslant 0) \\ &= 0 - 5x_2 + 10x_2\left(\frac{1}{2}\right) = 0. \end{aligned}$$

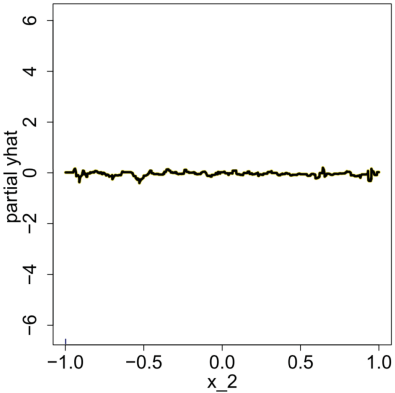
- Yet f has a linear x_2 term: either $-5x_2$ or $+5x_2$, depending on the value of x_3.

Example from from [GKBP15, p. 47].

Nonadditive example



(a) Scatterplot of Y versus X_2



(b) PDP

Figure 1 from [GKBP15, p. 47].

- On the left, we have a scatterplot of Y versus X_2 for the sample of size 1000 from the data generating distribution. We can clearly see the two different linear relationships with x_2, which we know depends on the sign of x_3.
- On the right we see a plot of the estimated partial dependence on x_2, which is approximately flat. This agrees with our calculation on the previous slide.
- This is an example of how a partial dependence plot will fail to capture the relationship between a function value and a particular input when the function does not decompose additively with respect to the input.

Individual conditional expectations (ICE)

Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation

Alex GOLDSTEIN, Adam KAPELNER, Justin BLEICH, and Emil PITKIN

- Really expanded and improved on the idea of PDP.
- The rest of this module will be based primarily on this paper [GKBP15].

Estimating the partial dependence function

- Recall our estimate for the partial dependence function:

$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} f(x_S, x_{Ci}),$$

  where $(x_{C1}, \ldots, x_{Cn})$ are the n instantiations of x_C in a dataset $\mathcal{D}$.
- Let's define the **individual conditional expectation** (ICE) for example i.
- Rather than integrating out over X_C, we condition on a specific value:

$$f_S^{(i)}(x_S) := \mathbb{E}[f(x_S, X_C) \mid X_C = x_{Ci}] = f(x_S, x_{Ci}).$$

- This tells us how f changes as we vary x_S, while keeping $x_C = x_{Ci}$.
- We can plot **all** the ICE functions $f_S^{(1)}, \ldots, f_S^{(n)}$ to get a better sense of the dependency.
- Of course the partial dependence function estimate is just the average: $\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} f_S^{(i)}$.

ICE plots

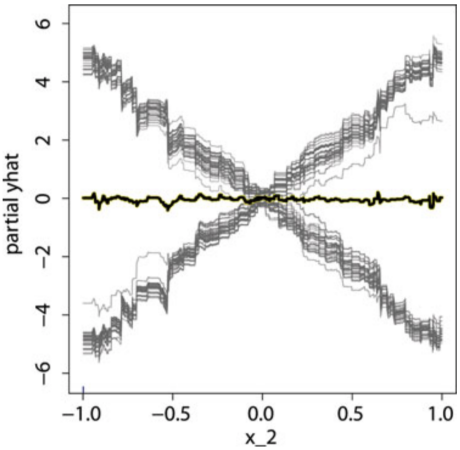
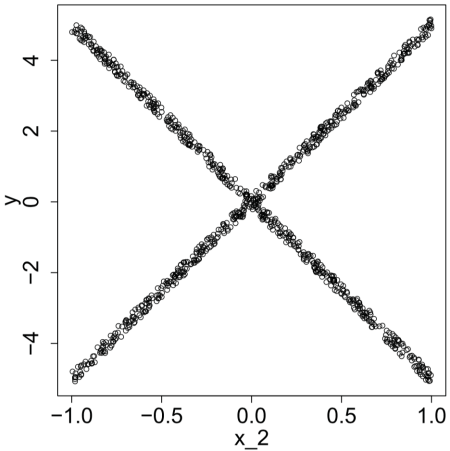
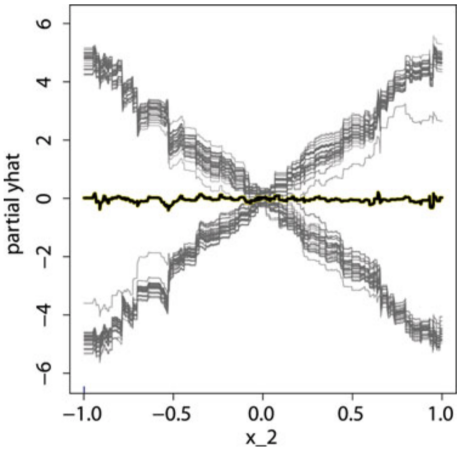


Figure 2 from [GKBP15, p. 48].

- Here we're taking $f(x)$ to be a gradient boosting fit to a our sample of size 1000. What's plotted are $f_2^{(1)}(x_2), \ldots, f_2^{(1000)}(x_2)$. Let's review how $f_2^{(i)}(x_2)$ is plotted. Suppose $x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}\right)$ is the input for example i. $f_2^{(i)}(x_2)$ is the function that shows how $f(x^{(i)}) = f(x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$ changes as we change $x_2^{(i)}$ while leaving $x_1^{(i)}$ and $x_3^{(i)}$ fixed. That is, $f_2^{(i)}(x_2) = f(x_1^{(i)}, x_2, x_3^{(i)})$. So if $x_3^{(i)} \geqslant 0$, and if $f(x) = \mathbb{E}[Y \mid X = x]$, then $f_2^{(i)}(x_2) = 0.2x_1^{(i)} + 5x_2$, and if $x_3^{(i)} < 0$, then $f_2^{(i)}(x_2) = 0.2x_1^{(i)} - 5x_2$. Of course, in what's plotted, we're taking $f(x)$ to be the gradient boosting fit to $\mathbb{E}[Y \mid X = x]$, so we just get na approximation.
- In the plots, note that some lines have a negative linear dependence on x_2 and some have a positive linear dependence on x_2, depending on the value of x_3. Note also that there's some variation in the starting locations of the lines. For the idealized $f(x) = \mathbb{E}[Y \mid X = x]$, this variation will depend on the value of $0.2x_1^{(i)}$. Since $X_1 \sim \text{Unif}(-1, 1)$, the range of that variation should be about $0.2 - (-0.2) = 0.4$. The variation we see in the plot is at least twice as large as this. Presumably, this is coming from the fact that the gradient boosting fit to the data is not a perfect match for $f(x)$.
- The roughly horizontal line in the middle is the partial dependence plot, which is the pointwise average of the 1000 other functions plotted here.

Scatterplot vs ICE plot





These two plots look quite similar in this case... will they always be similar?

Figures 1 and 2 from [GKBP15, p. 47-48].

- In the scatter plot, we're only ever plotting $f(x) = f(x_1, x_2, x_3)$ for input points $x^{(i)}$ that come from the data generating distribution.
- In the ICE plot on the right, we're plotting [our gradient boosting approximation to] f at points $\left(x_1^{(i)}, x_2, x_3^{(i)}\right)$, which do not come from the data generating distribution. x_2 is set to various values to create the plot. For this particular example, since X_1, X_2, X_3 are i.i.d., as long as we're setting x_2 to values from the marginal distribution of X_2, there is no difference. But in other data generating distributions, where X_1, X_2, X_3 are not independent, this can lead to differences between the scatter plot and the ICE plot.
- For example, suppose $X_2 = X_3$. Then in the scatter plot, we'd only see the positive-sloped line of points for $x_2 > 0$ and the negative slope for $x_2 < 0$. The dots would be in the form of a V. It's hard to predict what the gradient boosting fit would look like for this dataset. But if we plotted the partial dependence plot of the true regression function $\mathbb{E}[Y \mid X = x]$, it would again look like a full X.

A real example

- Another example from [GKBP15, p. 48-50].
- Regression problem from Boston Housing Data (BHD).
- Trying to predict a census tract's median house price from features of the census tract.
- Fit with random forests.
- We want to examine the dependency of the fit $\hat{f}$ on the feature "age".
  - "age" is the average age of homes in the census tract

ICE Plot for RF on BHD

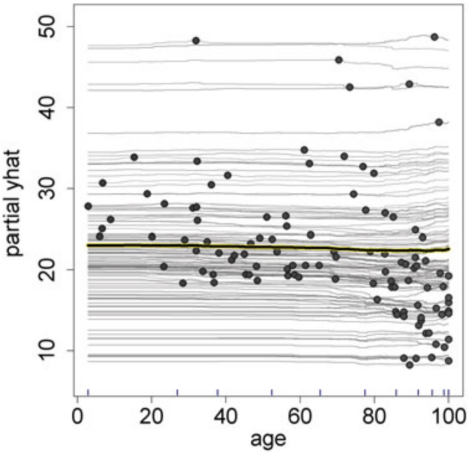


Figure 3 from [GKBP15, p. 49].

- The PDP is the highlighted line in the middle. It seems essentially flat.
- The black dot on each curve corresponds to the actual data point that the curve was built around.
- It doesn't seem like there's much of an effect of age on the response, even in the ICE plots... except, perhaps there's something going on in the high end of age? A lot of curves seem to have a slightly positive dependence on age starting around 85. But then other curves seem to have a slightly negative dependence on age, starting around 85. We'll revisit this.
- Also note the wide range of starting positions for these curves. This makes sense. There are many other factors that will have a large effect on the predicted price of the house besides age. These effects are visible in this range of starting values.

The centered ICE plot [GKBP15, Sec 3.2]

- It can be easier to visually appreciate the relative shapes of the ICE plots
  - if we shift each plot so that they all intersect at a single point.
- They suggest having the curves intersect either at the minimum of maximum of the range of x_S.
- In doing this, we lose two things:
  - Seeing the range of y values.
  - How the eventual y value interacts with the shape of the ICE curve.
- So the original ICE plot can be used in conjunction with the centered ICE or "c-ICE" plot

The c-ICE plot for RF on BHD

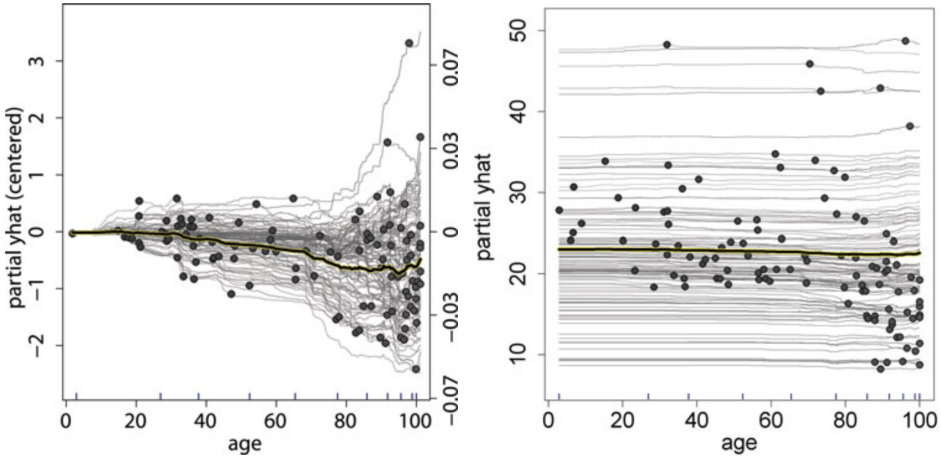


Figure 4 from [GKBP15, p. 50].

- With the "centering" at the left end of age, the heterogeneity of the age effect becomes quite apparent.
- On the other hand, looking at the scale of the y-axis on the left, compared to the scale of the y-axis in the original ICE plot, you can see that the effect of age isn't that large, after accounting for everything else. But 1) there is an effect, and 2) the effect is completely different depending on the settings of other variables.
- The scale on the right side of the c-ICE plot is described as displaying "changes in $\hat{f}$ over the baseline as a fraction of y's observed range." Here's how they get it: find the difference between the maximum and minimum values of $\hat{f}$ over all points evalutated. Then the right scale is just the left scale divided by that difference.
- How could one figure out what determines the relation between drivers the

Color / shade to visualize a second feature

- "rm" denotes the mean number of rooms per house in a census tract.
  - Color lines red if rm $>$ median(rm), blue otherwise.

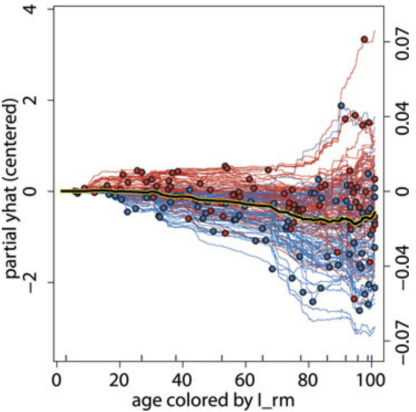


Figure 6 from [GKBP15, p. 52].

- Visually, we seem to have good support for $\mathbb{1}[\text{rm} > \text{median}(\text{rm})]$ being correlated with the type of partial dependence we have between the response and age.
- Could we discover this correlation automatically?
- If we can make a simple rule labeling curves by the pattern of partial dependence we see, perhaps we can try to predict that relationship based on simple, interpretable models, such as decision stumps or sparse linear models.

ICE plots and additivity

- Suppose the function f decomposes additively as

$$f(x_S, x_C) = h_S(x_S) + h_C(x_C),$$

  for some arbitrary h_S and h_C.
- The ICE plot corresponding to the ith point $X_C^{(i)}$ is

$$\left\{ (x_S, h_S(x_S) + h_C(X_C^{(i)}) : x_S \in \mathcal{X}_S \right\}.$$

- So all the ICE plots are shifted versions of $h_s(x_s)$.
- We can check for additive decomposition by "parallel" ICE curves.

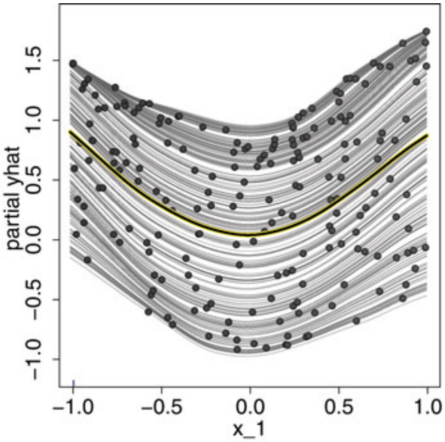
Checking for additivity

- Consider the following data-generating model:

$$Y = X_1^2 + X_2 + \varepsilon, \quad X_1, X_2 \sim \text{Unif}(-1, 1), \quad \varepsilon \sim \mathcal{N}(0, 1)$$

- Suppose we get a sample $\left(X^{(1)}, Y^{(1)}\right), \ldots, \left(X^{(n)}, Y^{(n)}\right)$ i.i.d. from this distribution.
- Using the sample, we want to figure out whether $\mathbb{E}[Y \mid X = x]$ decomposes additively,
  - or whether there's an interaction.
- We fit a model to these points.
- We can check for additivity with the ICE plots...
- Obviously, we can't conclude anything about $\mathbb{E}[Y \mid X = x]$.
- But it suggests how we can restrict whatever model class we're using.

ICE plot check for additivity



- These curves look fairly parallel – i.e. fairly close to additive.
- Not sure why they don't do a c-ICE plot so it would be more clear.

Figure 7 from [GKBP15, p. 53].

- If you find that your fit is essentially additive, what can you do?
- You could use a model that's explicitly additive, such as a GAM.
- You could build an additive model "by hand", by fitting a model on one set of features, and then [in the case of regression] fitting the residual using the other set of features. Something similar can be done for generalized additive models as well.

Extrapolation experiment

- Consider the following model (from [GKBP15, Sec 4.3]):

$$\begin{aligned} Y &= 10X_1^2 + \mathbb{1}[X_2 \geqslant 0] + \varepsilon \\ \varepsilon &\sim \mathcal{N}(0, 0.1^2) \\ (X_1, X_2) &\sim \text{Unif}(S), \end{aligned}$$

  where $S = (-1,1) \times (-1,1) - (0,1) \times (0,1)$.
- That is, $S \subset \mathbb{R}^2$ is a square with the upper right quadrant removed.
- So $\mathbb{P}(X_1 > 0, X_2 > 0) = 0$.
- For $f(x) = \mathbb{E}[Y \mid X = x]$, we'd expect the ICE plots for x_1 to be
  - two parabolas: $10x_1^2$ and $10x_1^2 + 1$.
- We draw 1000 observations and fit a random forest model $\hat{f}(x)$.
- Let's look at the ICE plots for $\hat{f}$ w.r.t. x_1...

All observations

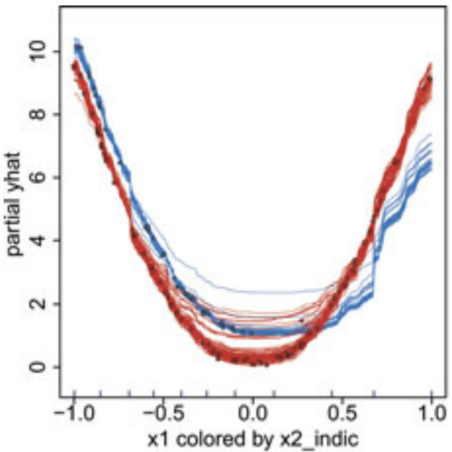


Figure 9(a) from [GKBP15, p. 55].

- Here the ICE plots are colored according to whether or not $X_2 < 0$.
- For $x_1 < 0$, it looks pretty good – that is, we do see two sets of parabolas, one set basically one unit higher than the other.
- For for $x_1 > 0$, the curves actually cross – completely different from what we'd expect for the ICE plot of $f(x) = \mathbb{E}[Y \mid X = x]$.
- If you look closely, you can see that there are no black dots on the blue curves to the right of $x_1 = 0$. This is because $\mathbb{P}(X_1 > 0, X_2 > 0)$=0 . So $\hat{f}$ is in a region of pure extrapolation – that is, there are no training examples in that region.
- What's the takeaway? Well, if we look at the plots naively, they suggest the function $\mathbb{E}[Y \mid X = x]$ has some interaction between X_1 and X_2, since the ICE curves are not parallel. However, this is the wrong conclusion: all we can really conclude is that the fit $\hat{f}$ has an interaction. By noting that $\hat{f}$ is extrapolating in the region that's suggesting an interaction, we realize that we don't have much evidence for an interaction in $\mathbb{E}[Y \mid X = x]$.

Resources

- Some recent work on interpreting GAMs (and by extension, partial dependency plots) in the multiclass setting can be found in [ZTK+18].
- The ESL book has a note on partial dependence plots [HTF09, Sec 10.13.2].
- Limitations of Interpretable Machine Learning Methods has some interesting and relevant chapters.
- Of course, all the papers referenced throughout this module.

References I

[Bre01] Leo Breiman, *Random forests*, Mach. Learn. **45** (2001), no. 1, 5–32.

[Fri01] Jerome H. Friedman, *Greedy function approximation: A gradient boosting machine.*, Ann. Statist. **29** (2001), no. 5, 1189–1232.

[GKBP15] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin, *Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation*, Journal of Computational and Graphical Statistics **24** (2015), no. 1, 44–65.

[GMSP15] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre, *Grouped variable importance with random forests and application to multiple functional data analysis*, Computational Statistics & Data Analysis **90** (2015), no. nil, 15–35.

[HM19] Giles Hooker and Lucas Mentch, *Please stop permuting features: an explanation and alternatives*, CoRR (2019).

References II

[HT86] Trevor Hastie and Robert Tibshirani, *Generalized Additive Models*, Statistical Science **1** (1986), no. 3, 297 – 310.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman, *The elements of statistical learning: Data mining, inference, and prediction, 2nd edition*, Springer Series in Statistics, Springer, 2009.

[LGR+18] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman, *Distribution-free predictive inference for regression*, Journal of the American Statistical Association **113** (2018), no. 523, 1094–1111.

[ZTK+18] Xuezhou Zhang, Sarah Tan, Paul Koch, Yin Lou, Urszula Chajewska, and Rich Caruana, *Axiomatic interpretability for multiclass additive models*, CoRR (2018).