Partial Dependency Plots

It's time for an example

- Let's try to predict median housing value in California for neighborhoods (pictures stolen from Elements of Statistical Learning, Chapter 10)...

Figure 14:Real estate?

Feature Importance Review

- Remember that when using an ensemble of trees, we can get a look at which variables are most important by looking at **feature importance**

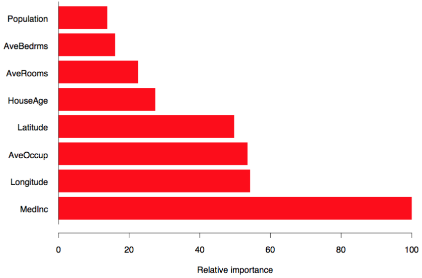

Figure 15:Feature Importances on California Housing

- How do we actually quantify the effects of one of these variables on the response, though?

Partial Dependency Plots - Overview I

- With **partial dependency plots**, we have a useful tool for teasing out and quantifying the effects of an individual variable on our response
- Effectively, **after fitting the model**, we'll cycle over some pre-determined values of the individual variable of interest, predicting on those values and observing how our responses changes
- How the response changes across different values of our variable of interest is the **partial dependency** of the response on that variable

Partial Dependency Plots Visual I

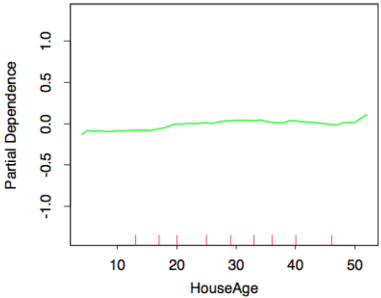

Figure 16:Partial dependence of median house value on median age of houses in the neighborhood

- Here, we can see that once we control for the average effects of all other variables, median house value has a small partial dependence on median age of the house

Partial Dependency Plots Visual II

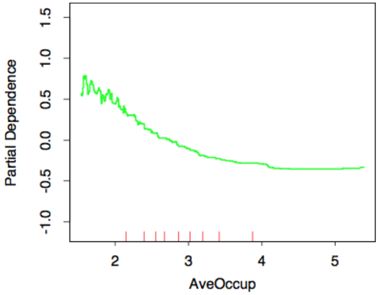

Figure 17:Partial dependence of median house value on average occupancy of houses in the neighborhood

- Here, we can see that once we control for the average effects of all other variables, median house value has a noticeable partial dependence on the average occupancy of houses in the neighborhood.

Partial Dependency Plots - Calculation

- To calculate the partial dependence by hand, a common way of doing it for a single variable is the following:
  1. Fit our machine learning model/algorithm - this should be able to basically tease out all of the average effects of each individual variable
  2. Pick a variable that you would like to calculate the partial dependency of
  3. Pick a range of values that you want to calculate the partial dependency for
  4. Loop over those values, one at a time doing the following for **each value**:
     1. Replace the entire column corresponding to the variable of interest with the current value that is being cycled over (we'll do this with our training set)
     2. Use the model to `predict` (again with the training data)
     3. Average all of the reponses, and calculate the difference of this average to the average calculated in the last iteration of the loop
     4. This (`value`, `difference`) becomes an (`x`, `y`) pair for your partial dependency plot

Partial Dependency Plots - Overview II

- We're first finding the average effects of each individual variable (that's the fitting step, 1)
- Then, we're observing how the response changes as the values of **one** variable change, **holding the effects of the other variables fixed**

Note: See the `Appendix` section for the hand calculation example used in class.

Partial Dependency Plots Visual III

- We can even plot the partial dependency of two variables relative to the response (more than two gets tough):

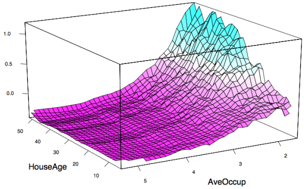

Figure 18:Partial dependence of median house value on median house age and average occupancy

- Here, we see that there is a strong interaction between `HouseAge` and `AveOccup`, which we weren't able to see in looking at either the feature importances or partial dependency plots of a single variable

Partial Dependency Plots Visual IV

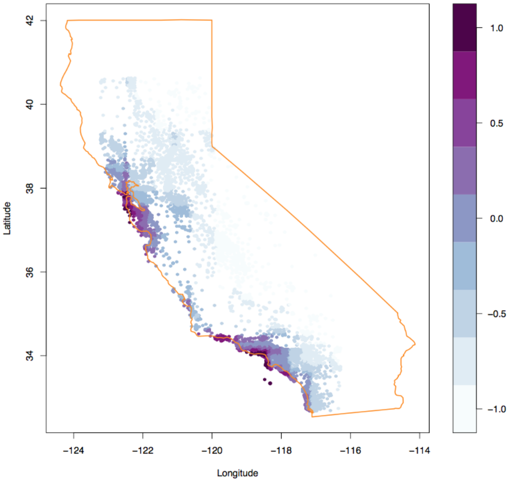

Figure 19:Partial dependence of median house value on latitude and longitude

Partial Dependency Plots - Code

- Let's walk through the code here (sklearn built-in for partial dependency):

Appendix - Partial Dependency Plots

Data

- Let's say we're trying to estimate the median housing value in California neighborhoods, and we have the following **original data** (yes, for now we only have 5 obs. and three columns):

med_value	avg_occup	med_age
3.5	2.3	4.1
4.5	3.1	1.2
3.7	4.9	4.7
2.1	1.6	3.3
1.3	2.8	5.8

med_value: Median housing value in a neighborhood (our response)
avg_occup: Average occupancy of houses in the neighborhood
med_age: Median age of houses in the neighborhood

Partial Dependence Process - A refresher

- To calculate the partial dependence by hand, a common way of doing it for a single variable is the following:
  1. Fit our machine learning model/algorithm - this should be able to basically tease out all of the average effects of each individual variable
  2. Pick a variable that you would like to calculate the partial dependency of
  3. Pick a range of values that you want to calculate the partial dependency for
  4. Loop over those values, one at a time doing the following for **each value**:
     1. Replace the entire column corresponding to the variable of interest with the current value that is being cycled over (we'll do this with our training set)
     2. Use the model to `predict` (again with the training data)
     3. Average all of the reponses, and calculate the difference of this average to the average calculated in the last iteration of the loop
     4. This (`value`, `difference`) becomes an (`x`, `y`) pair for your partial dependency plot

Partial Dependence Process - Steps I, II, and III

1. Fit our machine learning model/algorithm - this should be able to basically tease out all of the average effects of each individual variable
   - Let's say we fit gradient boosted trees
2. Pick a variable that you would like to calculate the partial dependency of
   - Let's go with `avg_occup`
3. Pick a range of values that you want to calculate the partial dependency for
   - Since our observations values range from 1.6 to 4.9, let's go from 1.5 to 5.0, by 0.1

Partial Dependence Process - Step IV

4. Loop over those values, one at a time doing the following for **each value**:
   1. Replace the entire column corresponding to the variable of interest with the current value that is being cycled over (we'll do this with our training set)
   2. Use the model to `predict` (again with the training data)
   3. Average all of the reponses, and calculate the difference of this average to the average calculated in the last iteration of the loop
   4. This (`value`, `difference`) becomes an (`x`, `y`) pair for your partial dependency plot

Partial Dependence Process - Step IV I

- We'll start by replacing the `avg_occup` with our first value (1.5), predicting, and the taking the mean response.

med_value	avg_occup	med_age	preds
3.5	1.5	4.1	3.4
4.5	1.5	1.2	4.6
3.7	1.5	4.7	3.5
2.1	1.5	3.3	2.0
1.3	1.5	5.8	1.4

- The mean prediction is $\frac{(3.4+4.6+3.5+2.0+1.4)}{5} = 2.98$
- Note there is no difference to calculate here becuase this is our first value

Partial Dependence Process - Step IV II

- We'll then by replace the avg_occup with our second value (1.6), predicting, and the taking the mean response.

med_value	avg_occup	med_age	preds
3.5	1.6	4.1	3.9
4.5	1.6	1.2	4.8
3.7	1.6	4.7	4.2
2.1	1.6	3.3	2.5
1.3	1.6	5.8	1.9

- The mean prediction is $\frac{(3.9+4.8+4.2+2.5+1.9)}{5} = 3.46$
- The difference between this and the mean prediction when avg_occup was 1.5 is 0.48, so we plot the point (1.6, 0.48)

Partial Dependence Process - Step IV III

- We'll then by replace the avg_occup with our third value (1.7), predicting, and the taking the mean response.

med_value	avg_occup	med_age	preds
3.5	1.7	4.1	3.7
4.5	1.7	1.2	4.6
3.7	1.7	4.7	4.1
2.1	1.7	3.3	2.6
1.3	1.7	5.8	1.5

- The mean prediction is $\frac{(3.7+4.6+4.1+2.6+1.5)}{5} = 3.3$
- The difference between this and the mean prediction when avg_occup was 1.6 is -0.16, so we plot the point (1.6, -0.16)

Partial Dependence Process - Iterating...

- We continue in this manner all the way up to the last value in the range of values we want to cycle over (we chose 5.0)...
- ....
- ....

Partial Dependence Process - Iterating...

- We'll then finish by replacing the avg_occup with our last value (5.0), predicting, and the taking the mean response.

med_value	avg_occup	med_age	preds
3.5	5.0	4.1	2.1
4.5	5.0	1.2	3.2
3.7	5.0	4.7	3.3
2.1	5.0	3.3	1.8
1.3	5.0	5.8	0.9

- The mean prediction is $\frac{(2.1+3.2+3.3+1.8+0.9)}{5} = 2.26$
- Pretending the last mean prediction (when we used 4.9) was 2.51, the difference between that and our current mean would be -0.25. So, we plot the point (5.0, -0.25)