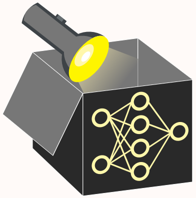
Interpretable Machine Learning

Leave One Covariate Out (LOCO)



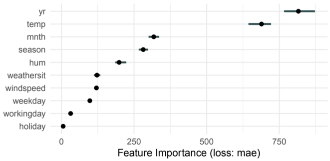


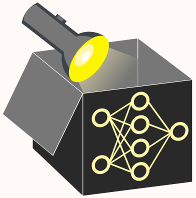
Figure: Bike Sharing Dataset

Learning goals

- Definition of LOCO
- Interpretation of LOCO

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018)

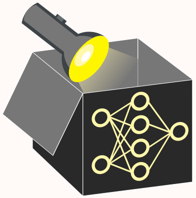
▸ Tibshirani (2018)



LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018)

▸ Tibshirani (2018)



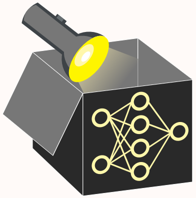
LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \dots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train}, -j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train}, -j})$

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018)

▸ Tibshirani (2018)



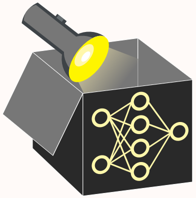
LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \dots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$
2. compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e. $\Delta_j^{(i)} = \left|y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)})\right| - \left|y^{(i)} - \hat{f}(x^{(i)})\right|$ with $i \in \mathcal{D}_{\text{test}}$

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018)

▸ Tibshirani (2018)



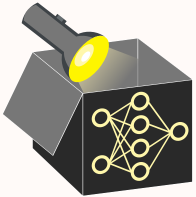
LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \dots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$
2. compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e. $\Delta_j^{(i)} = \left|y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)})\right| - \left|y^{(i)} - \hat{f}(x^{(i)})\right|$ with $i \in \mathcal{D}_{\text{test}}$
3. yield the importance score $\text{LOCO}_j = \text{med}\left(\Delta_j\right)$

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018)

▸ Tibshirani (2018)



LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \dots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$
2. compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e. $\Delta_j^{(i)} = \left|y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)})\right| - \left|y^{(i)} - \hat{f}(x^{(i)})\right|$ with $i \in \mathcal{D}_{\text{test}}$
3. yield the importance score $\text{LOCO}_j = \text{med}\left(\Delta_j\right)$

The method can be generalized to other loss functions and aggregations. If we use mean instead of median we can rewrite LOCO as

$$\text{LOCO}_j = \mathcal{R}_{\text{emp}}(\hat{f}_{-j}) - \mathcal{R}_{\text{emp}}(\hat{f}).$$

BIKE SHARING EXAMPLE

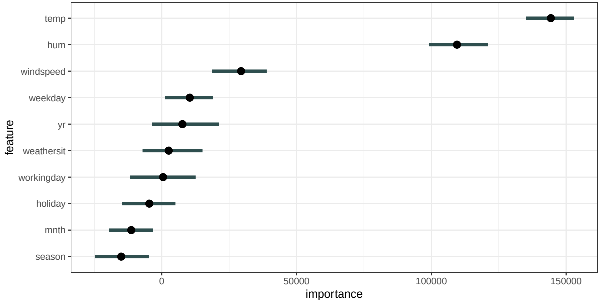
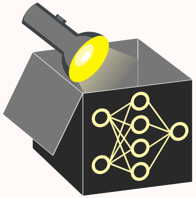


Figure: A random forest with default hyperparameters was fit on 70% of the bike sharing data (training set) to optimize MSE. Then LOCO was computed for all features on the test data. The temperature is the most important feature. Without access to `temp`, the MSE increases by approx. $140,000$.

INTERPRETATION OF LOCO



Interpretation: LOCO estimates the generalization error of the learner on a reduced dataset $\mathcal{D}_{-j}$.

Can we get insight into whether the ...

1. feature x_j is causal for the prediction $\hat{y}$?
   - In general, no also because we refit the model (counterexample on the next slide)
2. feature x_j contains prediction-relevant information?
   - In general, no (counterexample on the next slide)
3. model requires access to x_j to achieve its prediction performance?
   - Approximately, it provides insight into whether the *learner* requires access to x_j

INTERPRETATION OF LOCO

Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

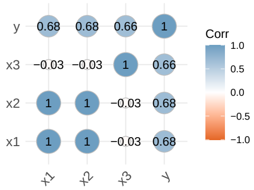
$\Rightarrow$ Fitting a LM yields
$\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$

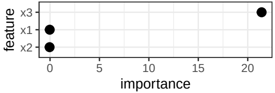
INTERPRETATION OF LOCO

Example: Sample 1000 observations with

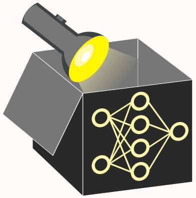
- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields
$\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$

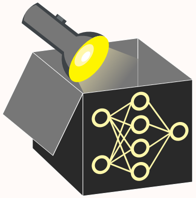




Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations



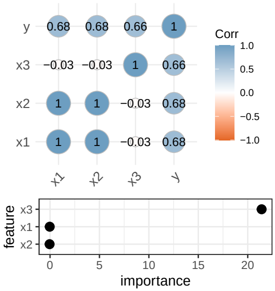
INTERPRETATION OF LOCO



Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields
$\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations

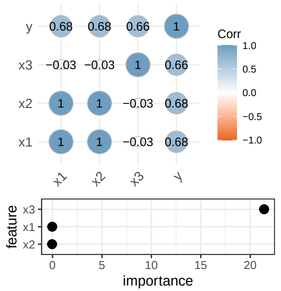
$\Rightarrow$ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coefficient of x_2 is 2.05)

INTERPRETATION OF LOCO

Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields
$\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations

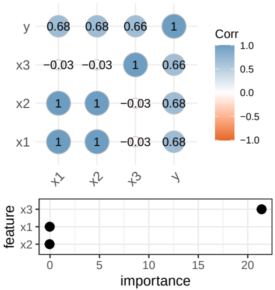
$\Rightarrow$ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coefficient of x_2 is 2.05)
$\Rightarrow$ We also can't infer (2), e.g., $Cor(x_2, y) = 0.68$ but $\text{LOCO}_2 \approx 0$

INTERPRETATION OF LOCO

Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

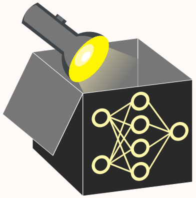
$\Rightarrow$ Fitting a LM yields
$\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations

$\Rightarrow$ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coefficient of x_2 is 2.05)
$\Rightarrow$ We also can't infer (2), e.g., $Cor(x_2, y) = 0.68$ but $\text{LOCO}_2 \approx 0$
$\Rightarrow$ We can get insight into (3): x_2 and x_1 highly correlated with $\text{LOCO}_1 = \text{LOCO}_2 \approx 0$
$\rightsquigarrow$ x_2 and x_1 can take each others place if one of them is left out (not the case for x_3)

PROS AND CONS



Pros:

- Requires (only?) one refitting step per feature for evaluation
- Easy to implement
- Testing framework available in ▸ Lei et al. (2018)

Cons:

- Does not provide insight into a specific model, but rather a learner on a specific dataset
- Model training is a random process, so estimates can be noisy (which is problematic for inference about model and data)
- Requires re-fitting the learner for each feature $\rightarrow$ computationally intensive compared to PFI