

Interpretable Machine Learning

Feature Importance

Leave One Covariate Out (LOCO)

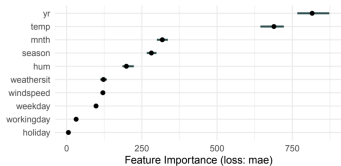
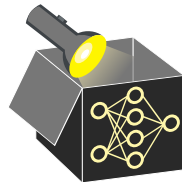


Figure: Bike Sharing Dataset

Learning goals

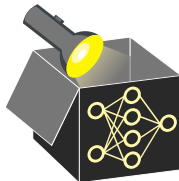
- Definition of LOCO
- Interpretation of LOCO

LOCO

► “Lei et al.” 2018

► “Tibshirani” 2018

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



LOCO

► “Lei et al.” 2018

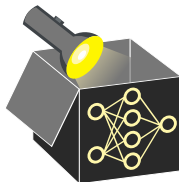
► “Tibshirani” 2018

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

Definition: Given train and test data $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, a learner \mathcal{I} , and model $\hat{f} := \mathcal{I}(\mathcal{D}_{\text{train}})$, the LOCO importance for feat $j \in \{1, \dots, p\}$ is computed by:

- 1 Learn model on $\mathcal{D}_{\text{train}, -j}$ where feature x_j was removed, i.e.

$$\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train}, -j})$$



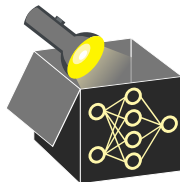
LOCO idea: Remove the feature from data, refit model on reduced data, and measure the loss in performance compared to model fitted on complete data.

Definition: Given train and test data $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, a learner \mathcal{I} , and model $\hat{f} := \mathcal{I}(\mathcal{D}_{\text{train}})$, the LOCO importance for feat $j \in \{1, \dots, p\}$ is computed by:

- ❶ Learn model on $\mathcal{D}_{\text{train}, -j}$ where feature x_j was removed, i.e.

$$\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train}, -j})$$
- ❷ Compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e.

$$\Delta_j^{(i)} = \left| y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)}) \right| - \left| y^{(i)} - \hat{f}(x^{(i)}) \right| \text{ with } i \in \mathcal{D}_{\text{test}}$$



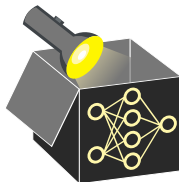
LOCO idea: Remove the feature from data, refit model on reduced data, and measure the loss in performance compared to model fitted on complete data.

Definition: Given train and test data $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, a learner \mathcal{I} , and model $\hat{f} := \mathcal{I}(\mathcal{D}_{\text{train}})$, the LOCO importance for feat $j \in \{1, \dots, p\}$ is computed by:

- ❶ Learn model on $\mathcal{D}_{\text{train}, -j}$ where feature x_j was removed, i.e.

$$\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train}, -j})$$
- ❷ Compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e.

$$\Delta_j^{(i)} = \left| y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)}) \right| - \left| y^{(i)} - \hat{f}(x^{(i)}) \right| \text{ with } i \in \mathcal{D}_{\text{test}}$$
- ❸ Compute importance score by $\text{LOCO}_j = \text{med}(\Delta_j)$



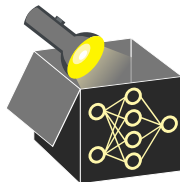
LOCO idea: Remove the feature from data, refit model on reduced data, and measure the loss in performance compared to model fitted on complete data.

Definition: Given train and test data $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, a learner \mathcal{I} , and model $\hat{f} := \mathcal{I}(\mathcal{D}_{\text{train}})$, the LOCO importance for feat $j \in \{1, \dots, p\}$ is computed by:

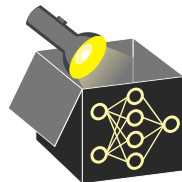
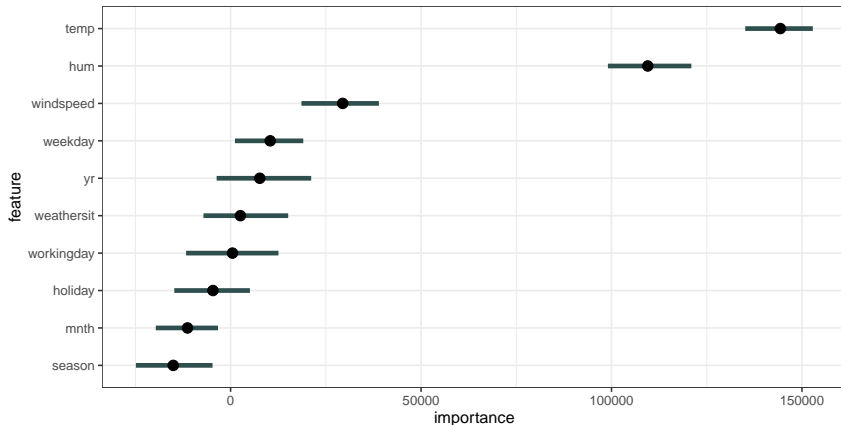
- ❶ Learn model on $\mathcal{D}_{\text{train}, -j}$ where feature x_j was removed, i.e.
 $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train}, -j})$
- ❷ Compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e.
 $\Delta_j^{(i)} = \left| y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)})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

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

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



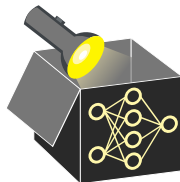
BIKE SHARING EXAMPLE



- Trained random forest (default hyperparams) on 70% of bike sharing data
- Performance measure: mean squared error (MSE)
- Computed LOCO on test set for all features, measuring increase in MSE
- temp was most important: removal increased MSE by approx. 140.000

INTERPRETATION OF LOCO

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



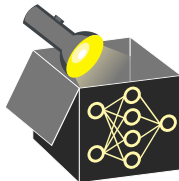
Can we get insight into whether the ...

- ❶ feature x_j is causal for the prediction \hat{y} ?
 - In general, no, also because we refit the model (counterexample on the next slide)
- ❷ feature x_j contains prediction-relevant information?
 - In general, no (counterexample on the next slide)
- ❸ 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 $\epsilon \sim N(0, 2)$
- Trained LM: $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



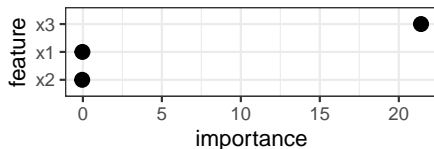
INTERPRETATION OF LOCO

Example: Sample 1000 observations with

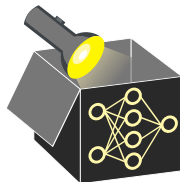
- $x_1, x_3 \sim N(0, 5)$, $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$
- Trained LM: $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Correlation matrix



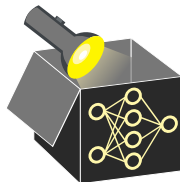
LOCO importance from LM trained on 70% of data, evaluated on remaining 30%



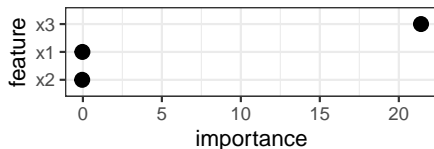
INTERPRETATION OF LOCO

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 $\epsilon \sim N(0, 2)$
- Trained LM: $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Correlation matrix



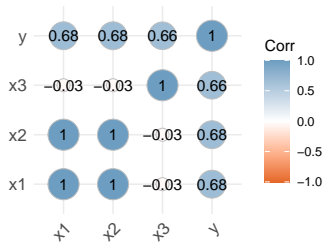
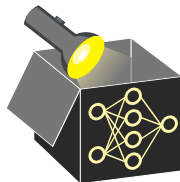
LOCO importance from LM trained on 70% of data, evaluated on remaining 30%

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

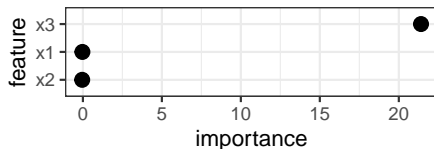
INTERPRETATION OF LOCO

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 $\epsilon \sim N(0, 2)$
- Trained LM: $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Correlation matrix



LOCO importance from LM trained on 70% of data, evaluated on remaining 30%

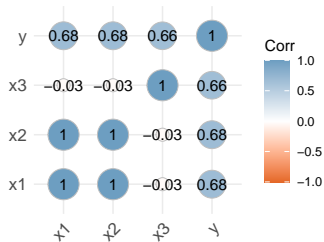
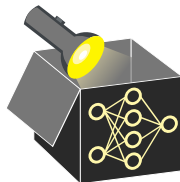
⇒ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coef. of x_2 is 2.05)

⇒ We also can't infer (2), e.g., $\text{Cor}(x_2, y) = 0.68$ but $\text{LOCO}_2 \approx 0$

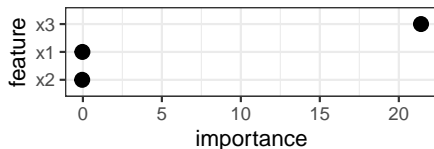
INTERPRETATION OF LOCO

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 $\epsilon \sim N(0, 2)$
- Trained LM: $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Correlation matrix



LOCO importance from LM trained on 70% of data, evaluated on remaining 30%

- ⇒ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coef. of x_2 is 2.05)
- ⇒ We also can't infer (2), e.g., $\text{Cor}(x_2, y) = 0.68$ but $\text{LOCO}_2 \approx 0$
- ⇒ We can get insight into (3): x_2, x_1 highly corr. with $\text{LOCO}_1 = \text{LOCO}_2 \approx 0$
 - ↪ x_2 and x_1 take each others place if one of them is left out (unlike 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:

- Provides insight into a learner on specific data, not a specific model
 - + for algorithm-level insight
 - for model-specific insights
- Model training is a random process and LOCO estimates can be noisy
 - ~> Limits inference on model and data, or multiple refittings necessary?
- Requires re-fitting the learner for each feature
 - ~> Computationally intensive compared to PFI

