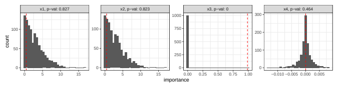
Interpretable Machine Learning

Permutation IMPortance (PIMP)



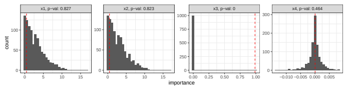
Learning goals

- Understand PIMP and its motivation
- Address multiple testing in feature importance

Interpretable Machine Learning

Feature Importances 1

Permutation IMPortance (PIMP)



Learning goals

- Understand PIMP and its motivation
- Address multiple testing in feature importance

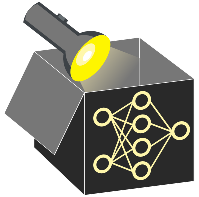
TESTING IMPORTANCE (PIMP) ▸ Altmann et al. (2010)

- PIMP was originally introduced for random forest’s built-in PFI scores

TESTING IMPORTANCE (PIMP) ▸ ALTMANN_2010

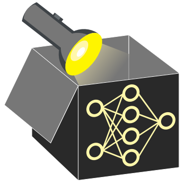
- PIMP was originally introduced for random forest’s built-in PFI scores

TESTING IMPORTANCE (PIMP) ▸ Altmann et al. (2010)



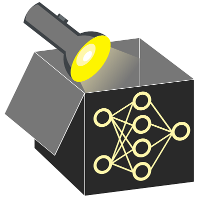
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness

TESTING IMPORTANCE (PIMP) ▸ ALTMANN_2010



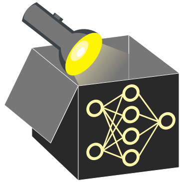
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness

TESTING IMPORTANCE (PIMP) ▸ Altmann et al. (2010)



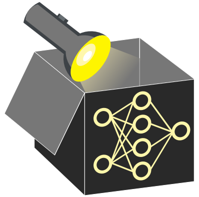
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness
- Null hypothesis H_0: Feature X_j is conditionally independent of y (unimportant)

TESTING IMPORTANCE (PIMP) ▸ ALTMANN_2010



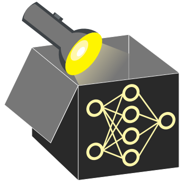
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness
- Null hypothesis H_0: Feature X_j is conditionally indep. of y (unimportant)

TESTING IMPORTANCE (PIMP) ▸ Altmann et al. (2010)



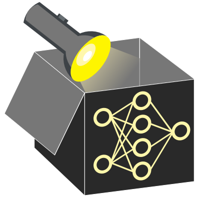
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness
- Null hypothesis H_0: Feature X_j is conditionally independent of y (unimportant)
- Approximate null distribution of PFI scores under H_0 by repeated permutations:
  Permute $y \rightarrow$ retrain model $\rightarrow$ recompute $\widehat{\text{PFI}}_j$ scores for all $j \rightarrow$ repeat B times
  $\Rightarrow$ Permuting y breaks relationship to all features (PFI scores reflect noise only)

TESTING IMPORTANCE (PIMP) ▸ ALTMANN_2010



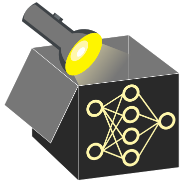
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness
- Null hypothesis H_0: Feature X_j is conditionally indep. of y (unimportant)
- Approximate null distrib. of PFI scores under H_0 by repeated permuts:
  Permute $y \rightarrow$ retrain $\rightarrow$ recompute $\widehat{\text{PFI}}_j$ scores for all $j \rightarrow$ repeat B times
  $\Rightarrow$ Permuting y breaks relationship to all features (PFI scores reflect noise only)

TESTING IMPORTANCE (PIMP) ▸ Altmann et al. (2010)



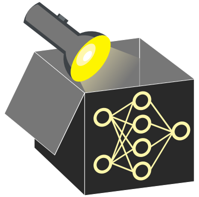
- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness
- Null hypothesis H_0: Feature X_j is conditionally independent of y (unimportant)
- Approximate null distribution of PFI scores under H_0 by repeated permutations:
  Permute $y \rightarrow$ retrain model $\rightarrow$ recompute $\widehat{\text{PFI}}_j$ scores for all $j \rightarrow$ repeat B times
  $\Rightarrow$ Permuting y breaks relationship to all features (PFI scores reflect noise only)
- Assess the significance of PFI scores via tail probability under H_0
  $\Rightarrow$ Use this as a new feature importance score, adjusting for random chance

TESTING IMPORTANCE (PIMP) ▸ ALTMANN_2010



- PIMP was originally introduced for random forest's built-in PFI scores
- PIMP idea: Test if an observed $\widehat{\text{PFI}}_j^{\text{obs}}$ score is *significantly* greater than expected under the null hypothesis of X_j being not important
  $\rightsquigarrow$ Accounts for spurious importance due to randomness
- Null hypothesis H_0: Feature X_j is conditionally indep. of y (unimportant)
- Approximate null distrib. of PFI scores under H_0 by repeated permuts:
  Permute $y \rightarrow$ retrain $\rightarrow$ recompute $\widehat{\text{PFI}}_j$ scores for all $j \rightarrow$ repeat B times
  $\Rightarrow$ Permuting y breaks relationship to all features (PFI scores reflect noise only)
- Assess the significance of PFI scores via tail probability under H_0
  $\Rightarrow$ Use this as a new feat. importance score, adjusting for random chance

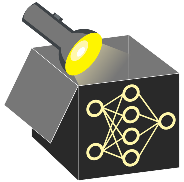
PIMP ALGORITHM



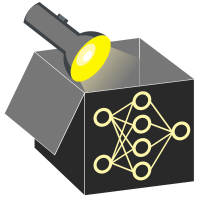
1. For $b \in \{1, \ldots, B\}$:
   - Permute response vector $\mathbf{y}$, denote permuted target as $\mathbf{y}^{(b)}$
   - Retrain model on data $(\mathbf{X}, \mathbf{y}^{(b)})$ with permuted target
   - Compute feature importance $\widehat{\text{PFI}}_j^{(b)}$ for each feature j (under H_0)

PIMP ALGORITHM

1. For $b \in \{1, \ldots, B\}$:
   - Permute response vector $\mathbf{y}$, denote permuted target as $\mathbf{y}^{(b)}$
   - Retrain model on data $(\mathbf{X}, \mathbf{y}^{(b)})$ with permuted target
   - Compute feature importance $\widehat{\text{PFI}}_j^{(b)}$ for each feature j (under H_0)



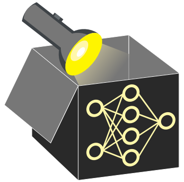
PIMP ALGORITHM



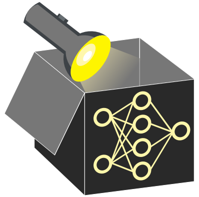
1. For $b \in \{1, \ldots, B\}$:
   - Permute response vector $\mathbf{y}$, denote permuted target as $\mathbf{y}^{(b)}$
   - Retrain model on data $(\mathbf{X}, \mathbf{y}^{(b)})$ with permuted target
   - Compute feature importance $\widehat{\text{PFI}}_j^{(b)}$ for each feature j (under H_0)
2. Train model on original data $(\mathbf{X}, \mathbf{y})$ with unpermuted target

PIMP ALGORITHM

1. For $b \in \{1, \ldots, B\}$:
   - Permute response vector $\mathbf{y}$, denote permuted target as $\mathbf{y}^{(b)}$
   - Retrain model on data $(\mathbf{X}, \mathbf{y}^{(b)})$ with permuted target
   - Compute feature importance $\widehat{\text{PFI}}_j^{(b)}$ for each feature j (under H_0)
2. Train model on original data $(\mathbf{X}, \mathbf{y})$ with unpermuted target



PIMP ALGORITHM



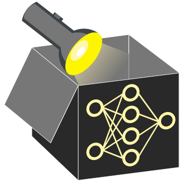
1. For $b \in \{1, \dots, B\}$:
   - Permute response vector $\mathbf{y}$, denote permuted target as $\mathbf{y}^{(b)}$
   - Retrain model on data $(\mathbf{X}, \mathbf{y}^{(b)})$ with permuted target
   - Compute feature importance $\widehat{\text{PFI}}_j^{(b)}$ for each feature j (under H_0)
2. Train model on original data $(\mathbf{X}, \mathbf{y})$ with unpermuted target
3. For each feature $j \in \{1, \dots, p\}$:
   - Compute $\widehat{\text{PFI}}_j^{\text{obs}}$ for the model without permutation of y (under H_1)
   - Fit probability distribution to all PFI scores $\{\widehat{\text{PFI}}_j^{(b)}\}_{b=1}^{B}$ (under H_0) e.g., by assuming Gaussian/lognormal/gamma distribution (parametric)
   - Compute p-value: Probability that null importance exceeds observed:
     - parametric by taking tail probability of assumed distribution

$$\mathbb{P}(\widehat{\text{PFI}}_j^{(m)} \geq \widehat{\text{PFI}}_j^{\text{obs}})$$

     - non-parametric by computing empirical tail probability:

$$p_j := \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}[\widehat{\text{PFI}}_j^{(b)} \geq \widehat{\text{PFI}}_j^{\text{obs}}]$$

PIMP ALGORITHM



1. For $b \in \{1, \dots, B\}$:
   - Permute response vector $\mathbf{y}$, denote permuted target as $\mathbf{y}^{(b)}$
   - Retrain model on data $(\mathbf{X}, \mathbf{y}^{(b)})$ with permuted target
   - Compute feature importance $\widehat{\text{PFI}}_j^{(b)}$ for each feature j (under H_0)
2. Train model on original data $(\mathbf{X}, \mathbf{y})$ with unpermuted target
3. For each feature $j \in \{1, \dots, p\}$:
   - Compute $\widehat{\text{PFI}}_j^{\text{obs}}$ for the model without permutation of y (under H_1)
   - Fit probability distribution to all PFI scores $\{\widehat{\text{PFI}}_j^{(b)}\}_{b=1}^{B}$ (under H_0) e.g., by assuming Gaussian/lognormal/gamma distrib (parametric)
   - Compute p-value: Prob. that null importance exceeds observed:
     - parametric by taking tail probability of assumed distribution

$$\mathbb{P}(\widehat{\text{PFI}}_j^{(m)} \geq \widehat{\text{PFI}}_j^{\text{obs}})$$

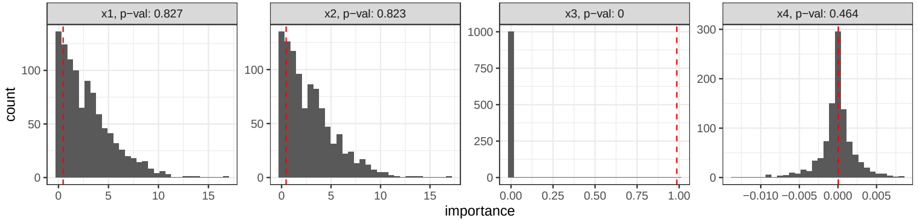
     - non-parametric by computing empirical tail probability:

$$p_j := \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}[\widehat{\text{PFI}}_j^{(b)} \geq \widehat{\text{PFI}}_j^{\text{obs}}]$$

PIMP FOR EXTRAPOLATION EXAMPLE

Recall: Let $y = x_3 + \epsilon_y$, with $\epsilon_y \sim \mathcal{N}(0, 0.1)$.

- $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$ are highly correlated ($\epsilon_1 \sim \mathcal{N}(0, 1)$, $\epsilon_2 \sim \mathcal{N}(0, 0.01)$)
- $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim \mathcal{N}(0, 1)$ and all noise terms ϵ_j are independent
- Fitting a linear model yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$

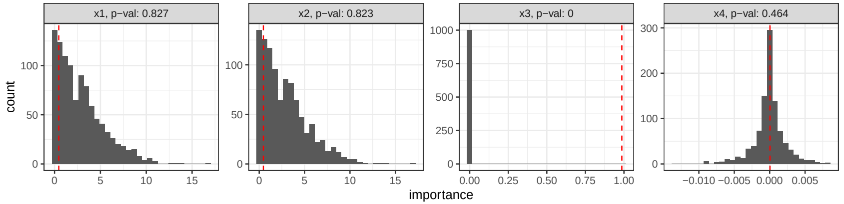


- Histograms: H_0 distribution of PFI scores after permuting y (1000 repetitions)
- Red: Observed PFI score (under H_1) $\rightsquigarrow$ compare against H_0 distribution
- Recall: PFI for x_1, x_2, x_3 is nonzero suggesting they are important (red lines)
- PIMP considers x_1, x_2 not significantly relevant (p-value > 0.05)

PIMP FOR EXTRAPOLATION EXAMPLE

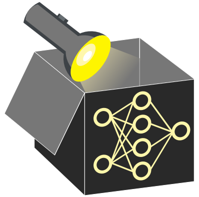
Recall: Let $y = x_3 + \epsilon_y$, with $\epsilon_y \sim \mathcal{N}(0, 0.1)$.

- $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$; highly correlated ($\epsilon_1 \sim \mathcal{N}(0, 1)$, $\epsilon_2 \sim \mathcal{N}(0, 0.01)$)
- $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim \mathcal{N}(0, 1)$; all noise terms ϵ_j are indep.
- Fitting a linear model yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$



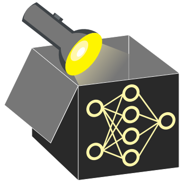
- Histograms: H_0 distrib. of PFI scores after permuting y (1000 repetitions)
- Red: Observed PFI score (under H_1) $\rightsquigarrow$ compare against H_0 distribution
- Recall: PFI for x_1, x_2, x_3 is non-0 suggesting they are important (red lines)
- PIMP considers x_1, x_2 not significantly relevant (p-value > 0.05)

DIGRESSION: MULTIPLE TESTING ▸ Romano et al. (2010)



- When should we reject H_0 for a given feature?
- PIMP conducts one hypothesis test per feature $\Rightarrow$ **multiple testing problem**
- With many tests, rejections of true H_0 just by chance (type-I errors) accumulate
- To account for this, control a suitable error rate, e.g., the **family-wise error rate**
  FWE: probability of making at least one type-I error across all tests
- A classical method is the **Bonferroni correction**:
  reject H_0 if p-value $< \alpha/m$ where m is the number of tests

DIGRESSION: MULTIPLE TESTING ▸ ROMANO_2010



- When should we reject H_0 for a given feature?
- PIMP conducts one hypothesis test per feature
  $\Rightarrow$ **multiple testing problem**
- With many tests, rejections of true H_0 just by chance (type-I errors) accumulate
- To account for this, control a suitable error rate, e.g., the **family-wise error rate**
  FWE: probability of making at least one type-I error across all tests
- A classical method is the **Bonferroni correction**:
  reject H_0 if p-value $< \alpha/m$ where m is the number of tests