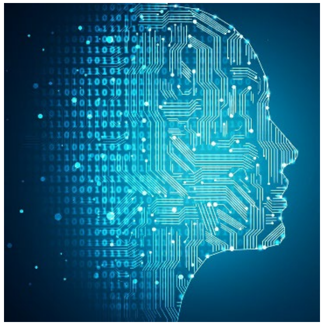
Machine Learning

Random Forests

Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Outline

1. Random forests review
2. Random forest predictions
3. Variable importance
4. Random forest proximities
    1. Different definitions
    2. Data Imputation
    3. Outlier detection
    4. Dimensionality reduction (DR)

Random Forests (RF) Review

- Grow a **forest** of many trees. (R default is 500).
- Grow each tree on an independent **bootstrap sample** (sample N cases at random with replacement) from the training data
- At each node:
  1. Select ***m*** variables **at random** out of all ***M*** possible variables (independently for each node)
  2. Find the best split on the selected ***m*** variables
- Grow the trees to maximum depth (classification)
- Vote/average the trees to get predictions for new data
- Let's look at some RF properties in detail

Random Forests Benefits

- Work for classification and regression
- Train quickly
- Little to no parameter tuning
- Provide an estimate of generalization error
- Trivially parallelizable
- Handle mixed variable types (e.g. categorical)
- Unaffected by monotonic transformations
- Insensitive to outliers in the predictor space
- Scale to small and large datasets
- Model nonlinear interactions

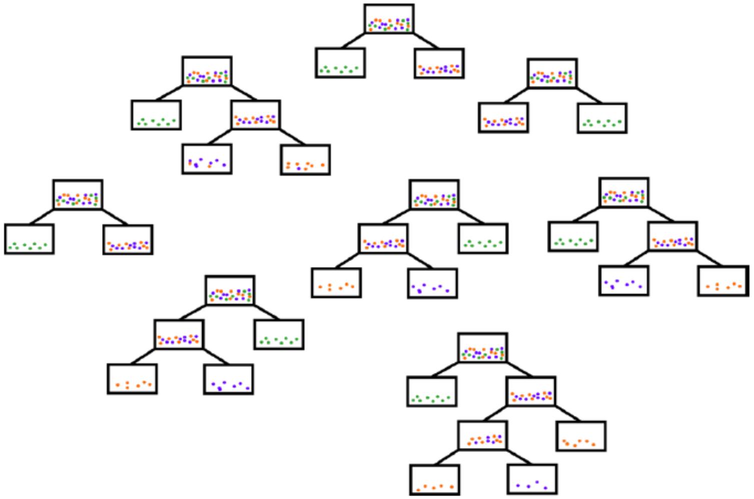
(Breiman 2001; Cutler et al. 2012)

Random Forest Uses

- Predicting surface water salinity (Ali Khan et al. 2022)
- Assessing shear strength of soft clays (Zhang et al. 2021)
- Analyzing building structure on CO2 emissions (Lin et al. 2021)
- Modeling the heterogeneity of water quality (Wang et al. 2021)
- Raman Spectra Classification(Zhang et al. 2020)
- Patient health prediction (COVID-19) (Iwendi et al. 2020)
- Spatio-temporal COVID-19 case estimation (Ye, silkanat 2020)
- Landslide susceptibility mapping (Nhu et al. 2020)
- Cardiovascular disease prediction (Yang et al. 2020)
- Deforestation rate prediction (Saha et al. 2020)
- Nanofluid viscosity estimation (Gholizadeh et al. 2020)
- Rural credit assessment (Rao et al. 2020)
- Wearable-sensor activity classification (Badar ud din Tahir et al. 2020)

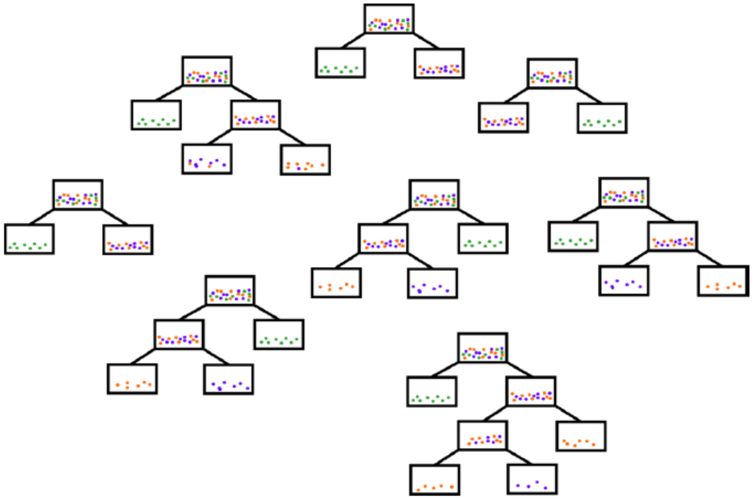
RF Predictions

- Assume we have training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, $\boldsymbol{x}_i \in \mathbb{R}^d$
- Bootstrapping gives us a natural test error estimate
- Consider a bootstrap sample I_t used to train tree t
  - *In-bag* samples are present in I_t
  - *Out of bag* (OOB) samples are NOT present in I_t
- What is the RF prediction for training points $\boldsymbol{x}_i$ and test points $\boldsymbol{x}$?



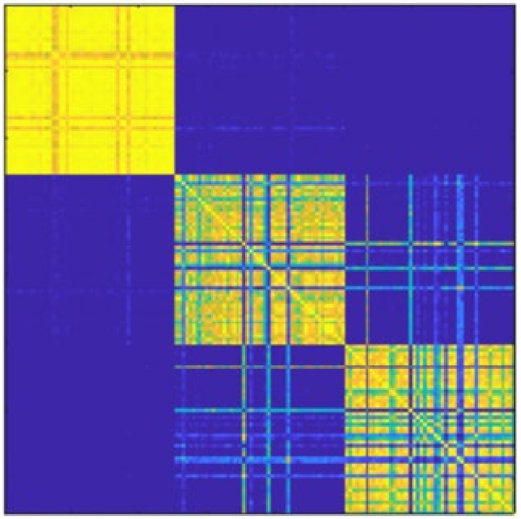
RF Predictions

- Suppose we fit 1000 trees and the sample $\boldsymbol{x}_i$ is OOB in 339 of them
- The prediction for $\boldsymbol{x}_i$ is the majority vote or average prediction of the 339 trees
- The OOB error rate is the average OOB error of the RF predictions of the training set
  - Gives an estimate of generalization error
  - No test set required, although still useful if there's enough data
- For test points, use all of the trees to obtain a prediction



Additional RF Uses

- Assessing variable importance
  - Variable selection
- Providing a notion of similarity (proximity)
  - Outlier detection
  - Data visualization and dimensionality reduction
  - Data imputation
  - Multi-modal learning



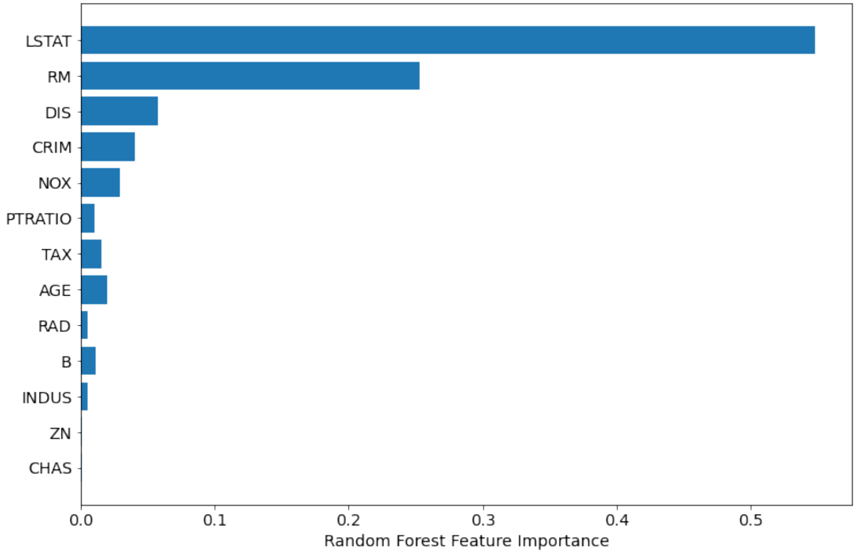
Variable Importance

Variable Importance

- Variable importance measures how important a variable is for the supervised task
- Useful for data exploration, feature selection, model interpretation, etc.
- Two main approaches in RF:
  1. Mean decrease in the impurity score
  2. Mean decrease in accuracy (permutation importance)
- Different methods will give somewhat different results
  - Both methods above tend to overstate the importance of correlated variables (an area of research)
  - Typically best to run both and compare

Decrease in Impurity Score

- Recall that each split in a decision tree is decided based on which split decreases the impurity score the most
- One measure of variable importance is to simply sum up the total impurity decrease for each variable and divide by the number of trees
- Actual importance scores aren’t meaningful
  - Variables are compared to each other
- Computationally cheap
- Biased towards continuous variables and variables with many categories
- Variable importance plots (Boston housing data):



Plot from https://mljar.com/blog/feature-importance-in-random-forest/

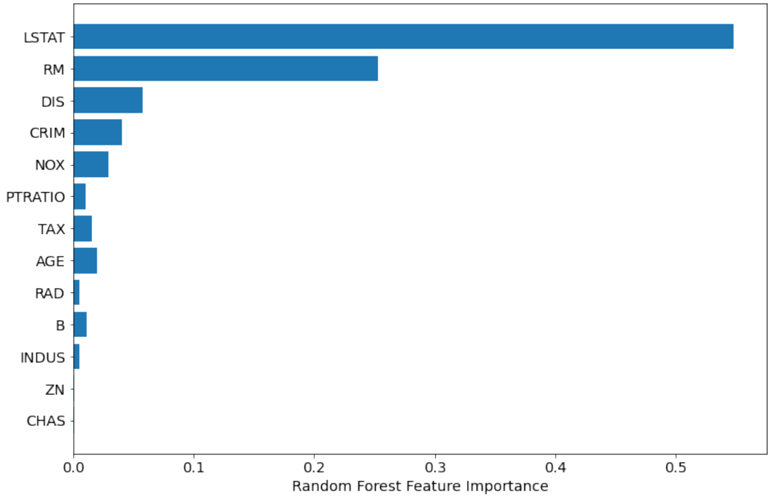
Permutation Variable Importance

Permutation variable importance is model agnostic

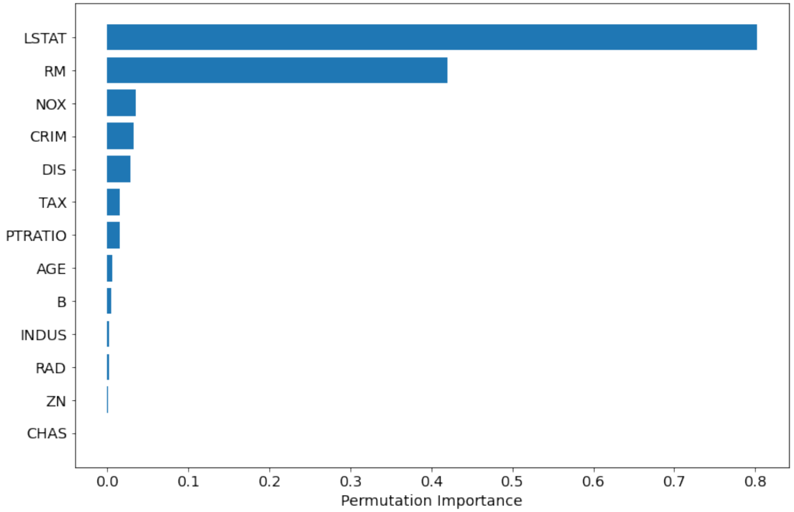
1. For each variable (one at a time), randomly shuffle its values across the samples
2. Compute the OOB error with the shuffled data
3. Compute the decrease in accuracy compared to the real data

- Actual importance scores have some meaning
- Can be computationally expensive

Gini Importance Score (Boston housing data)



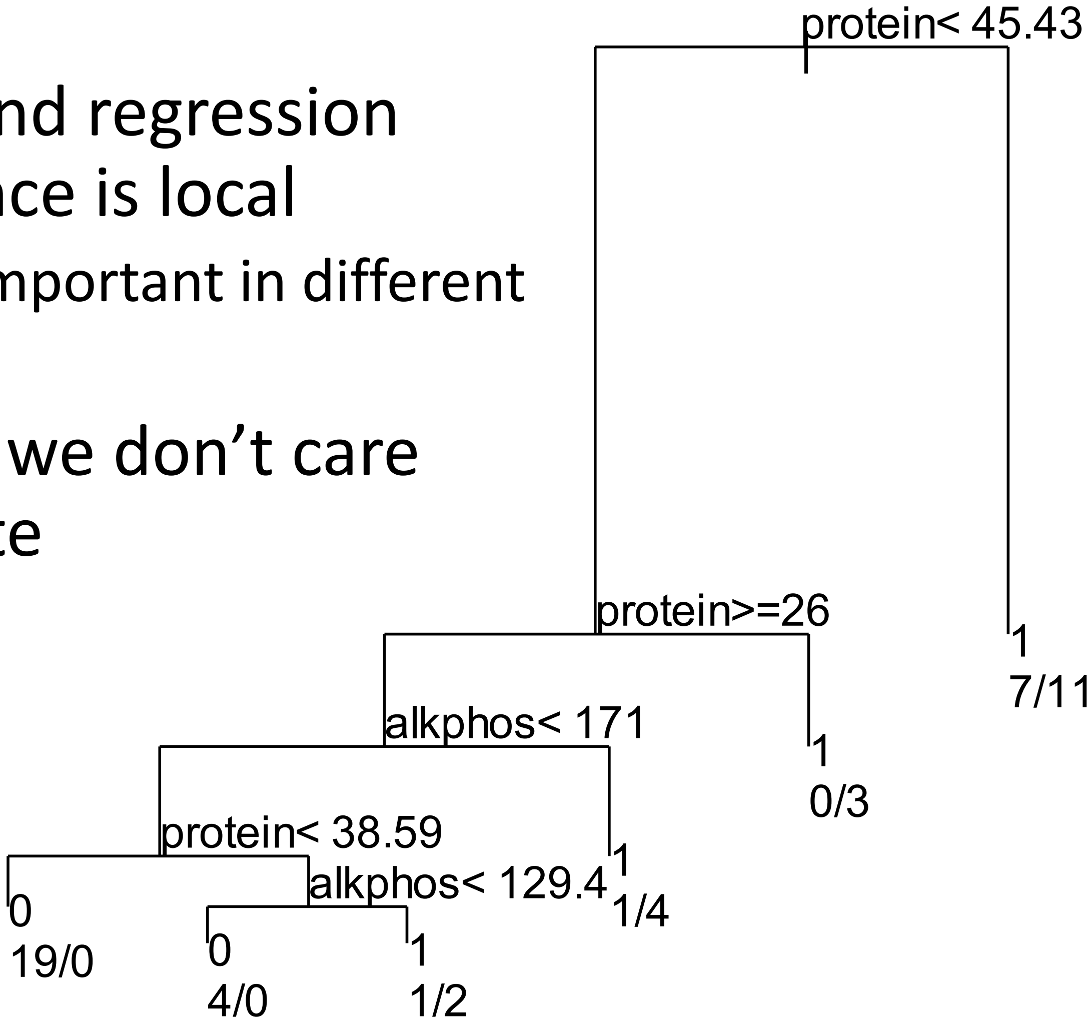
Permutation Importance Score (Boston housing data)



Plots from https://mljar.com/blog/feature-importance-in-random-forest/

Local Variable Importance

- Typically think about global variable importance
- In CART (classification and regression trees) variable importance is local
  - Different variables are important in different regions of the data
- If protein is high or low, we don't care about alkaline phosphate
- For intermediate values of protein, alkaline phosphate is important



Local Variable Importance

One (unpublished) approach:

1. For each tree, consider OOB data:
    1. Randomly permute the values of variable j
    2. Pass the perturbed data down the tree
2. For sample $\boldsymbol{x}_i$ and variable j find

$$\left\{\begin{array}{c}\text{error rate with}\\ \text{variable } j \text{ permuted}\end{array}\right\} - \left\{\begin{array}{c}\text{error rate with}\\ \text{no permutation}\end{array}\right\}$$

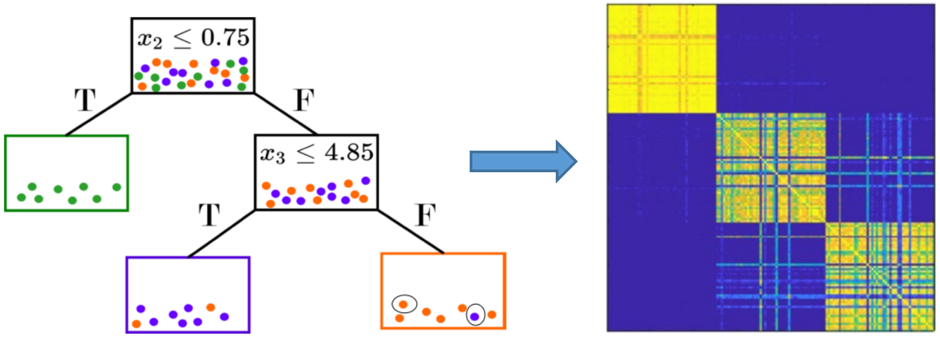
- Error rates are taken over all trees for which $\boldsymbol{x}_i$ is OOB

- We’ll see more on variable importance later

RF Proximities

Proximities

- Many ML algorithms rely on pairwise distances or affinities/proximities
  - **Examples**: SVM, manifold learning, clustering, nearest neighbor methods
- Most proximity measures are unsupervised
  - E.g. kernel functions such as in PHATE, DM, t-SNE, UMAP, etc.
- **Goal**: Construct supervised proximities that take into account label information
  - Ideally, gives a measure of similarity between the variables relevant for the supervised task while ignoring irrelevant variables
- We'll use RF to construct proximities



RF Proximities – Original

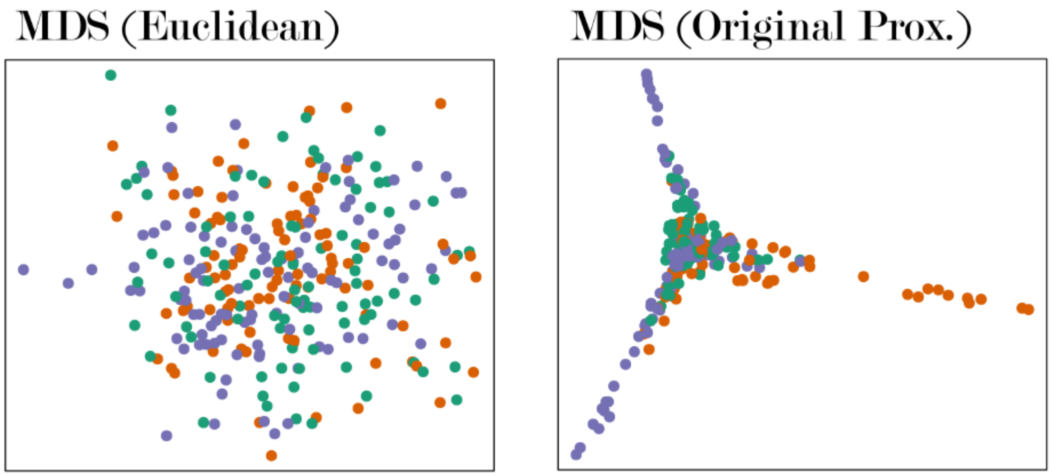
- (Breiman 2001) The random forest proximity between observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$p_{OR}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{T}\sum_{t=1}^{T} 1\left(\boldsymbol{x}_j \in v_i(t)\right)$$

  - T = # of trees in the forest
  - $v_i(t)$ = terminal node (leaf) in tree t containing $\boldsymbol{x}_i$
- I.e., the proximity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is the proportion of trees in which they reside in the same leaf

RF Proximities – Original

- These proximities exaggerate class separation
- A random sample of 300 points generated from bivariate normal distribution with random classes:



- Reason: trees are grown until pure
- In-bag samples of opposing classes end in different nodes

$$\Pr(\boldsymbol{x} \in \text{bootstrap sample}) = 1 - 1/e \approx \frac{2}{3}$$

RF Proximities – OOB

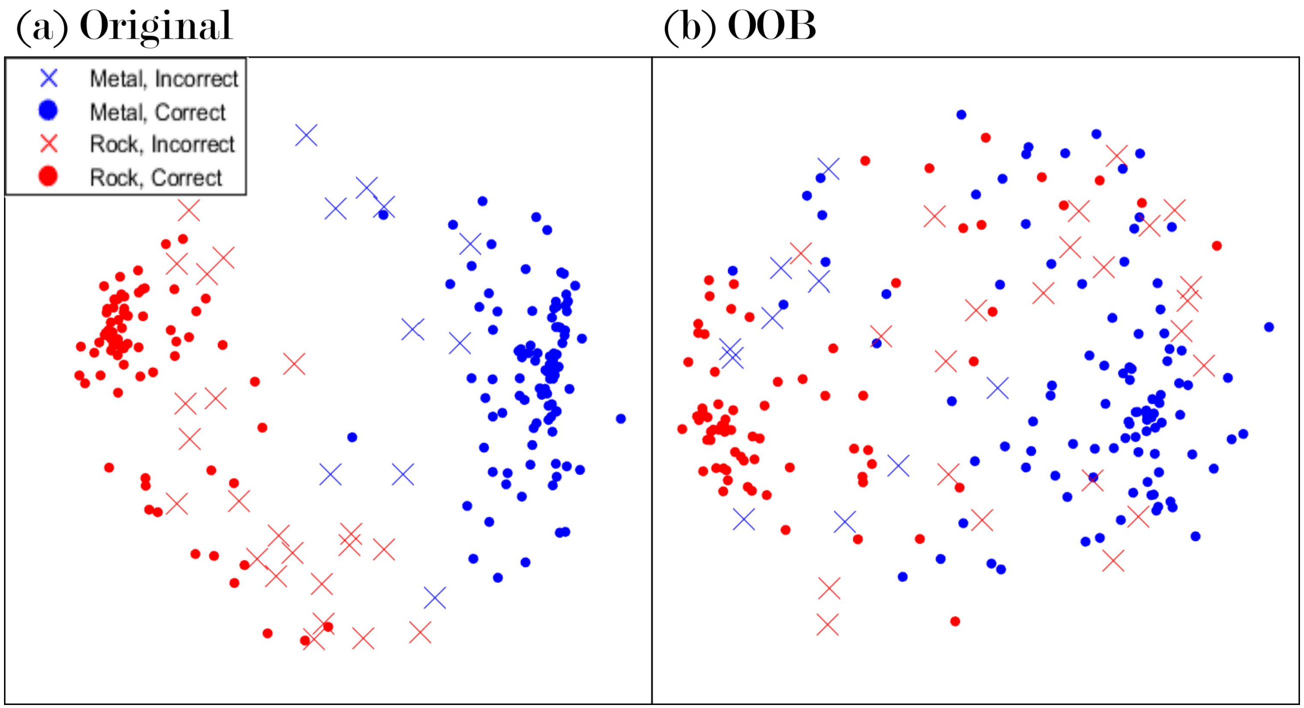
- (Hastie et al. 2009) The OOB proximity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$p_{OOB}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \frac{\sum_{t\in S_i} 1\left(\boldsymbol{x}_j \in \left(O(t) \cap v_i(t)\right)\right)}{\sum_{t\in S_i} 1\left(\boldsymbol{x}_j \in O(t)\right)}$$

  - $O(t) =$ set of OOB observations in tree t
  - $S_i =$ set of trees for which $\boldsymbol{x}_i$ is OOB
- I.e. the proximity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is the proportion of trees in which they reside in the same leaf when both are OOB

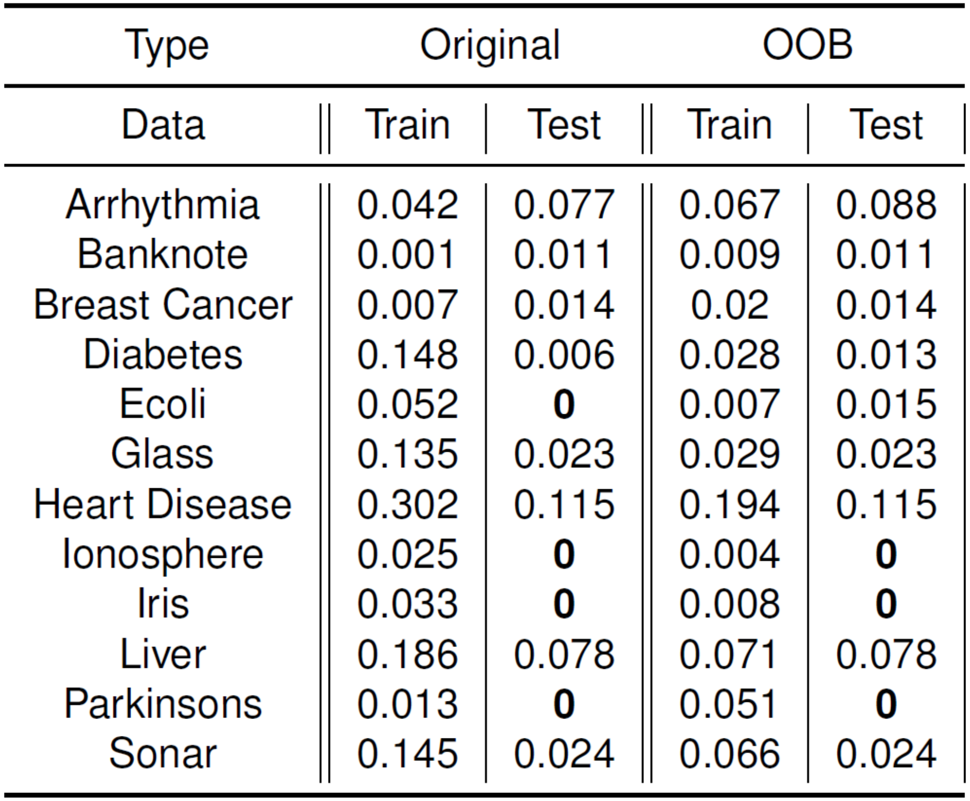
RF Proximities - OOB

- The OOB proximities do better than the original at preserving geometry
- MDS applied to the proximities on the Sonar dataset
  - RF accuracy was 84.13%



RF Geometry

- Ideally, the RF proximities should encode RF learning
- Construct a proximity-weighted nearest neighbor classifier/regressor
  - How often does it match the RF performance?



Type	Original		OOB	
Data	Train	Test	Train	Test
Arrhythmia	0.042	0.077	0.067	0.088
Banknote	0.001	0.011	0.009	0.011
Breast Cancer	0.007	0.014	0.02	0.014
Diabetes	0.148	0.006	0.028	0.013
Ecoli	0.052	**0**	0.007	0.015
Glass	0.135	0.023	0.029	0.023
Heart Disease	0.302	0.115	0.194	0.115
Ionosphere	0.025	**0**	0.004	**0**
Iris	0.033	**0**	0.008	**0**
Liver	0.186	0.078	0.071	0.078
Parkinsons	0.013	**0**	0.051	**0**
Sonar	0.145	0.024	0.066	0.024

Difference between the RF error and proximity-weighted error

What went wrong?

Proximity constructions are pairwise while the RF predictions are not

- **Original**
  - If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are both in-bag, $class_i \neq class_j \rightarrow p_{OR}^{t}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$
  - Finds signal within the noise
- **OOB**
  - Only OOB examples are used
  - $\approx \frac{1}{9}$ of observation pairs are OOB
  - Additional trees required for stability
  - Does not take into account in-bag samples, which were used to construct the trees

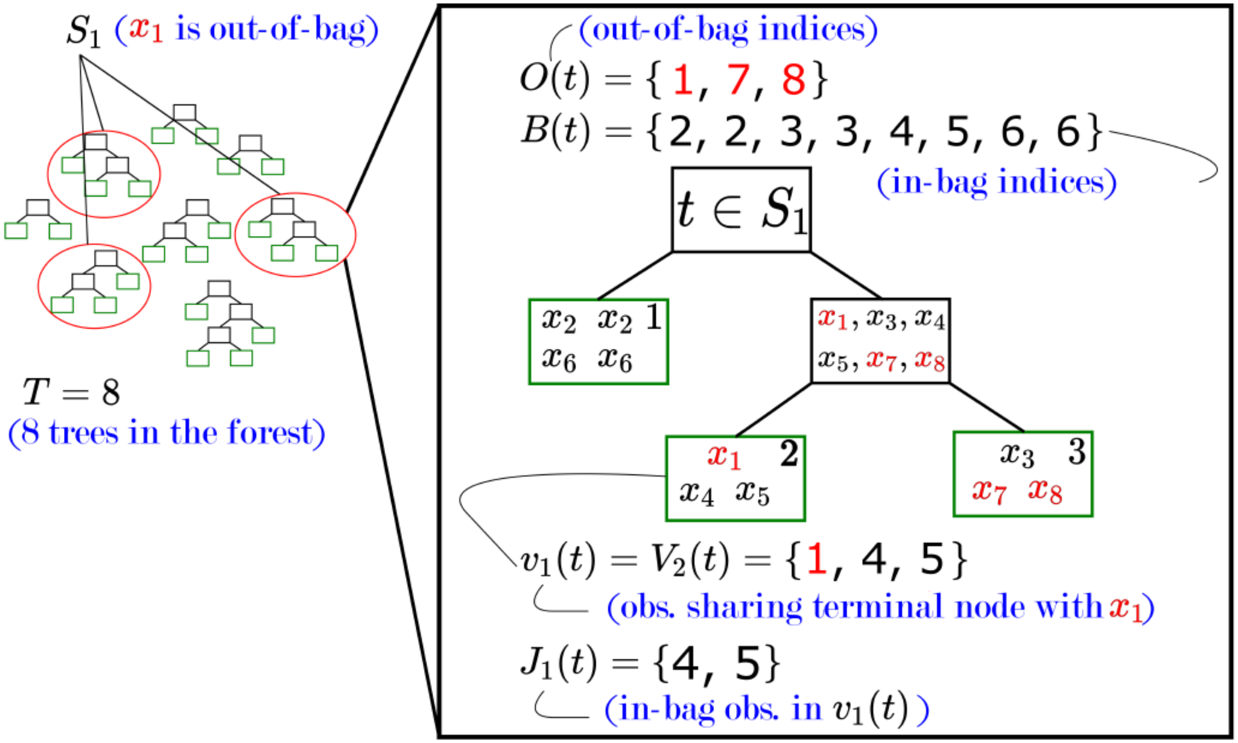
RF-GAP

- Idea: weight in-bag and OOB samples appropriately to match the RF performance
- (Rhodes et al. 2022) The RF-GAP proximities are:

$$p_{GAP}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \frac{1}{|S_i|} \sum_{t \in S_i} \frac{1\left(\boldsymbol{x}_j \in J_i(t)\right)}{|J_i(t)|}$$

  - $B(t) =$ multiset of in-bag samples
  - $J_i(t) = B(t) \cap v_i(t) =$ set of in-bag samples which share the leaf node with $\boldsymbol{x}_i$ in tree t
- I.e. the proximity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is the average proportion of in-bag samples in the shared leaf node of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in all trees where $\boldsymbol{x}_i$ is OOB and $\boldsymbol{x}_j$ is in-bag.

RF-GAP Notation



S_1 (x_1 is out-of-bag)

$T = 8$

(8 trees in the forest)

(out-of-bag indices)

$O(t) = \{1, 7, 8\}$

$B(t) = \{2, 2, 3, 3, 4, 5, 6, 6\}$

(in-bag indices)

$t \in S_1$

$v_1(t) = V_2(t) = \{1, 4, 5\}$

(obs. sharing terminal node with x_1)

$J_1(t) = \{4, 5\}$

(in-bag obs. in $v_1(t)$)

$$p_{GAP}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{|S_i|}\sum_{t\in S_i}\frac{1\left(\boldsymbol{x}_j \in J_i(t)\right)}{|J_i(t)|}$$

Proximity-Weighted Prediction

Theorem (Proximity-Weighted Classification)

For a given training data set $\mathcal{S} = \{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_N, y_N)\}$, with $y_i \in \{1, \cdots, K\}$ for all $i \in \{1, \cdots, N\}$, the random forest OOB classification prediction is determined by the weighted-majority vote using RF-GAP proximities as weights.

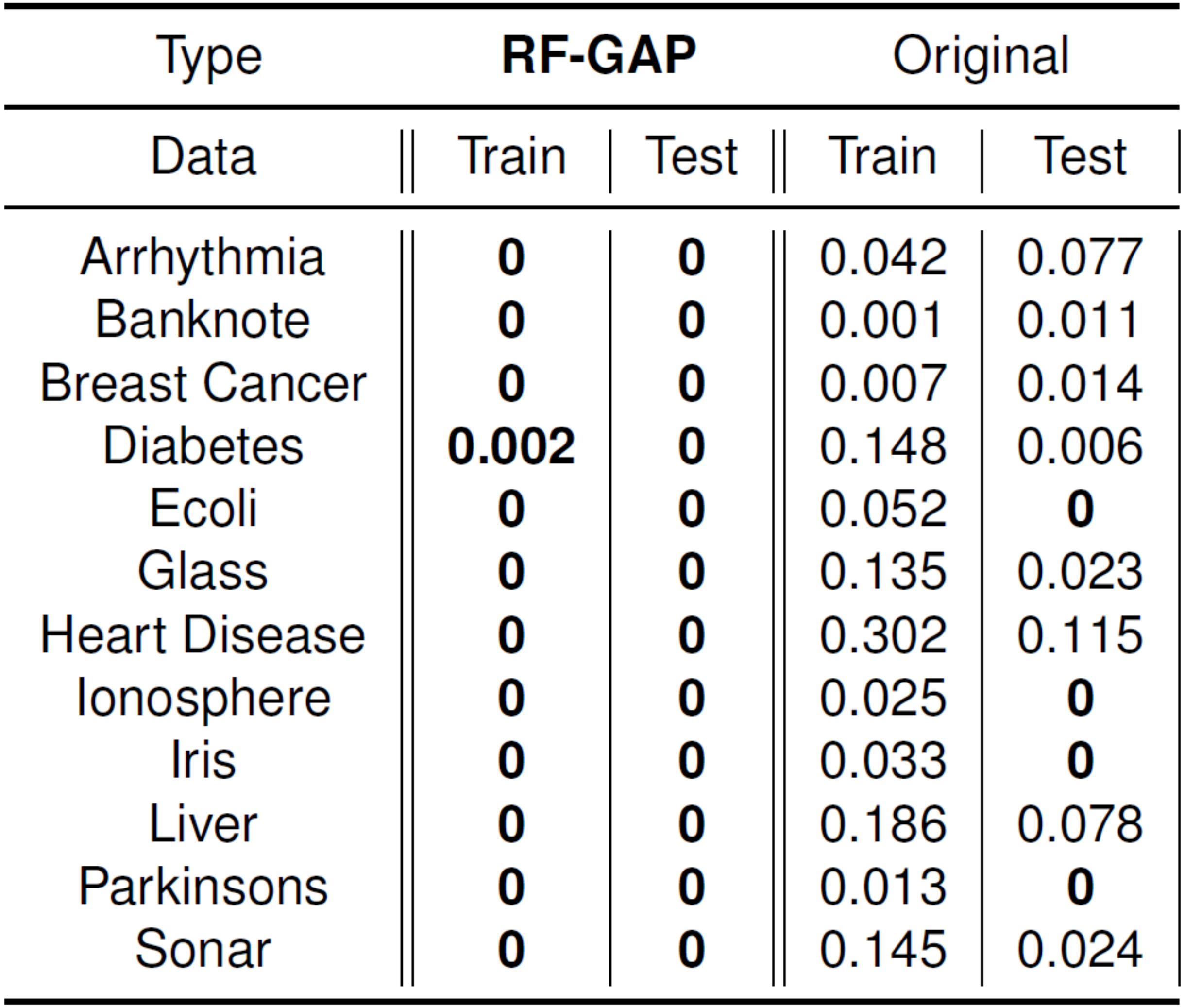
Theorem (Proximity-Weighted Regression)

For a given training data set $\mathcal{S} = \{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_N, y_N)\}$, with $y_i \in \mathbb{R}$, the random forest OOB regression prediction is determined by the proximity-weighted sum using RF-GAP proximities as weights.

Proofs of theorems found in (Rhodes et al. 2022).

Proximity-Weighted Prediction

Difference between RF error and proximity-weighted error



Type	**RF-GAP**		Original	
Data	Train	Test	Train	Test
Arrhythmia	**0**	**0**	0.042	0.077
Banknote	**0**	**0**	0.001	0.011
Breast Cancer	**0**	**0**	0.007	0.014
Diabetes	**0.002**	**0**	0.148	0.006
Ecoli	**0**	**0**	0.052	**0**
Glass	**0**	**0**	0.135	0.023
Heart Disease	**0**	**0**	0.302	0.115
Ionosphere	**0**	**0**	0.025	**0**
Iris	**0**	**0**	0.033	**0**
Liver	**0**	**0**	0.186	0.078
Parkinsons	**0**	**0**	0.013	**0**
Sonar	**0**	**0**	0.145	0.024

(Rhodes et al, 2022)

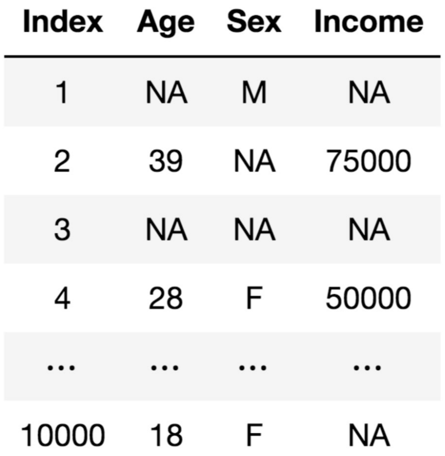
RF Proximities wrap-up

- Unsupervised proximities measure pairwise samples considering all variables
- RF proximities consider mostly the variables that are important for the supervised task
  - Two samples that have **different** labels might have **large** proximity if they differ only on variables that are **unimportant**
  - Two samples that have **similar** labels might have **small** proximity if they differ on variables that are **important**
- RF-GAP accurately reflects what the RF has learned based on the nearest neighbor geometry
  - Thus much more likely to reflect these variable relationships

RF Data Imputation

To impute missing data:

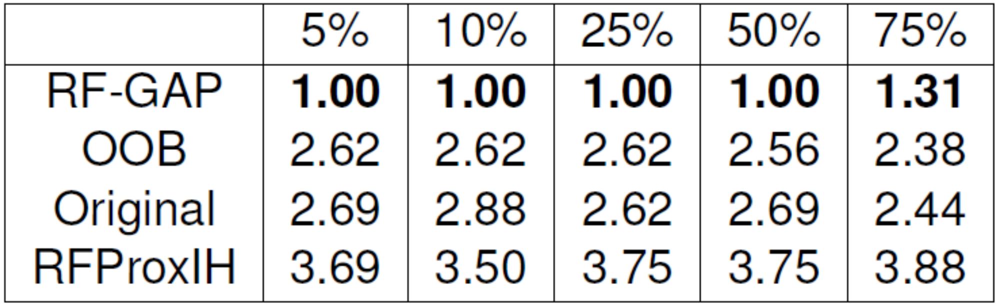
1. Initialize with the median (continuous) or mode (discrete)
2. Train a RF on the imputed dataset
3. Construct the proximities from the RF
4. Replace the missing values with the proximity weighted sum (continuous) or majority vote (discrete)
5. Repeat steps 2-4 until convergence



Index	Age	Sex	Income
1	NA	M	NA
2	39	NA	75000
3	NA	NA	NA
4	28	F	50000
...	...	...	...
10000	18	F	NA

RF Data Imputation

- Comparison across 25 UCI repository datasets and five percentages of missing values
- Each experiment repeated 100 times
- Reported average rank of four proximity measures



	5%	10%	25%	50%	75%
RF-GAP	**1.00**	**1.00**	**1.00**	**1.00**	**1.31**
OOB	2.62	2.62	2.62	2.56	2.38
Original	2.69	2.88	2.62	2.69	2.44
RFProxIH	3.69	3.50	3.75	3.75	3.88

- RF-GAP vastly outperforms the others

(Rhodes et al, 2022)

RF Outlier Detection

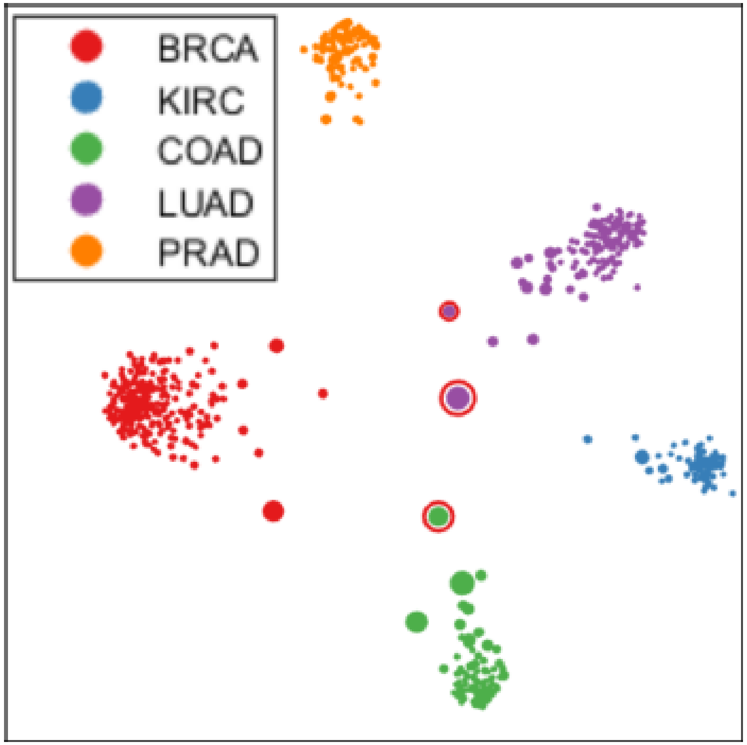
- Outliers are generally defined to be samples that are dissimilar from all/most observations
- In the supervised context, can consider dissimilarity with same-class observations

Approach:

1. Compute the raw score: $\sum_{j \in class(i)} \frac{n}{prox^2(i,j)}$
2. Standardize using the median score and mean absolute deviation (class-wise)

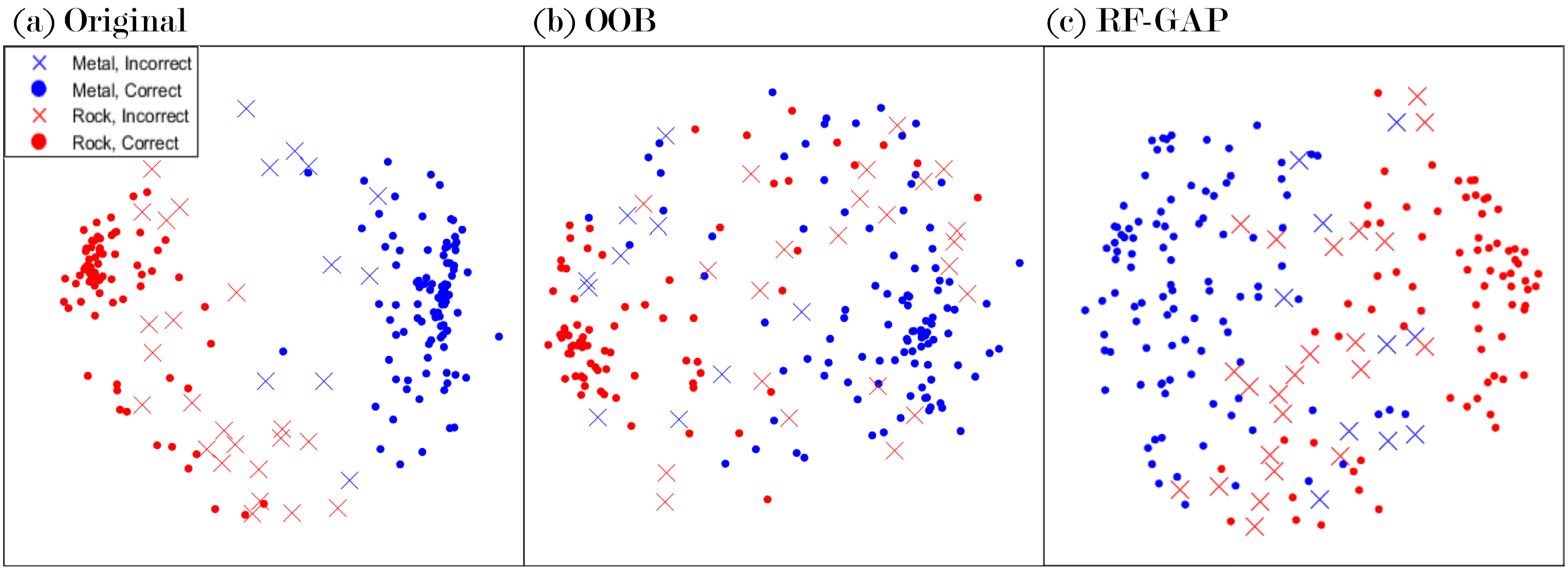
RF Outlier Detection

- MDS applied to RF-GAP proximities generated from gene expression cancer dataset (Dua and Graff, 2017)
- Point size is scaled by outlier score



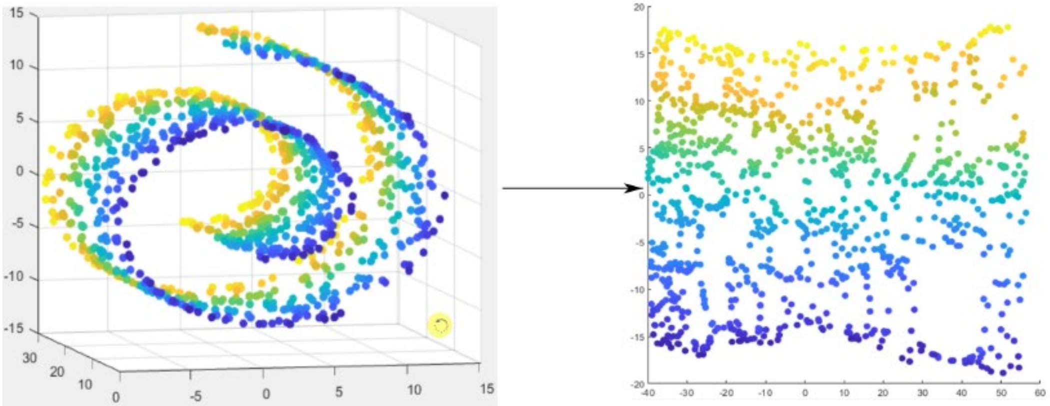
MDS with RF Proximities

- MDS applied to RF proximities on the Sonar dataset
- RF accuracy is 84.1%
- RF-GAP best displays this accuracy



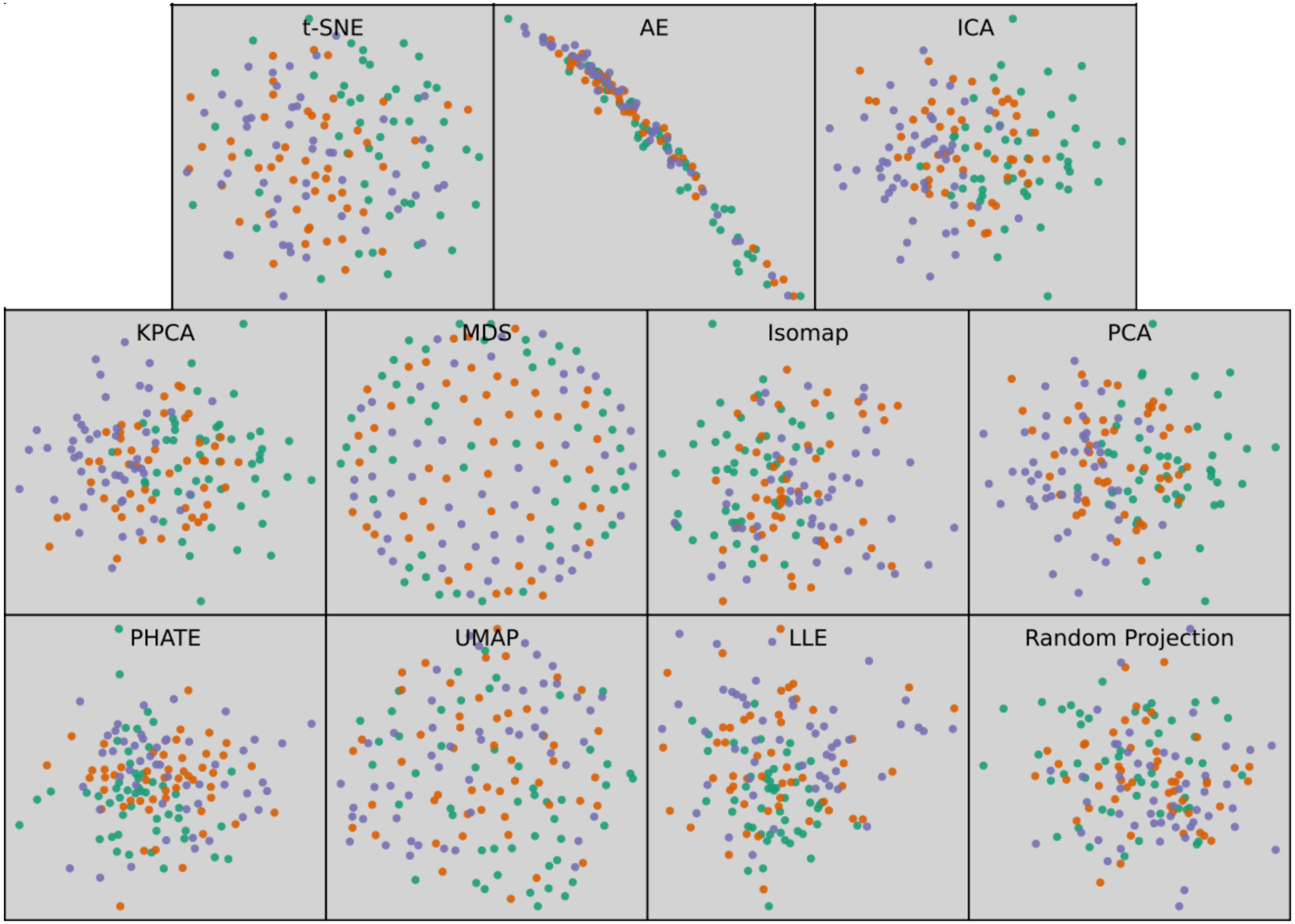
Dimensionality Reduction

- Recall our discussion of dimensionality reduction, manifold learning, and visualization
- All of those methods are **unsupervised**
  - E.g. PCA, t-SNE, PHATE, UMAP, DM, etc.
  - Tend to reveal the dominating structure present in all variables
- What if we want to focus on variables relevant to a supervised task
- Do supervised dimensionality reduction!



Supervised Dimensionality Reduction

- Unsupervised methods fail when there are noise variables
- Iris dataset with 500 added noise variables:



Supervised Dimensionality Reduction

- Common approach: modify distance based on labels

$$\text{D'}\left(x_i, x_j\right) = \begin{cases} \sqrt{1 - e^{\frac{-D^2\left(x_i, x_j\right)}{\beta}}} & y_i = y_j \\ \sqrt{e^{\frac{D^2\left(x_i, x_j\right)}{\beta}}} - \alpha & y_i \neq y_j \end{cases}$$

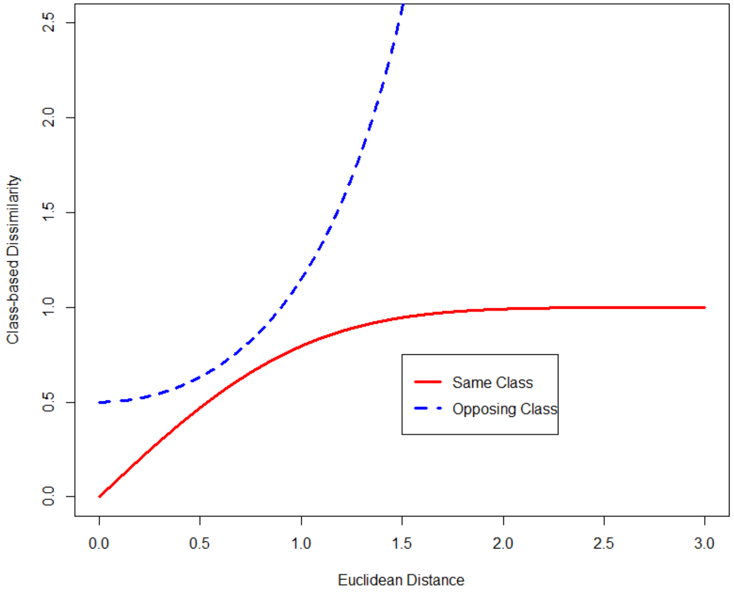
$$\text{D'}(\text{x}_i, x_j) = \begin{cases} \frac{1}{\alpha} D\left(x_i, x_j\right) & y_i = y_j, \quad \alpha > 1 \\ D\left(x_i, x_j\right) & y_i \neq y_j \end{cases}$$

$$\text{D'}(\text{x}_i, x_j) = \begin{cases} D\left(x_i, x_j\right) & y_i = y_j \\ D\left(x_i, x_j\right) + \alpha \max \mathcal{D} & y_i \neq y_j, \quad 0 \leq \alpha \leq 1 \end{cases}$$

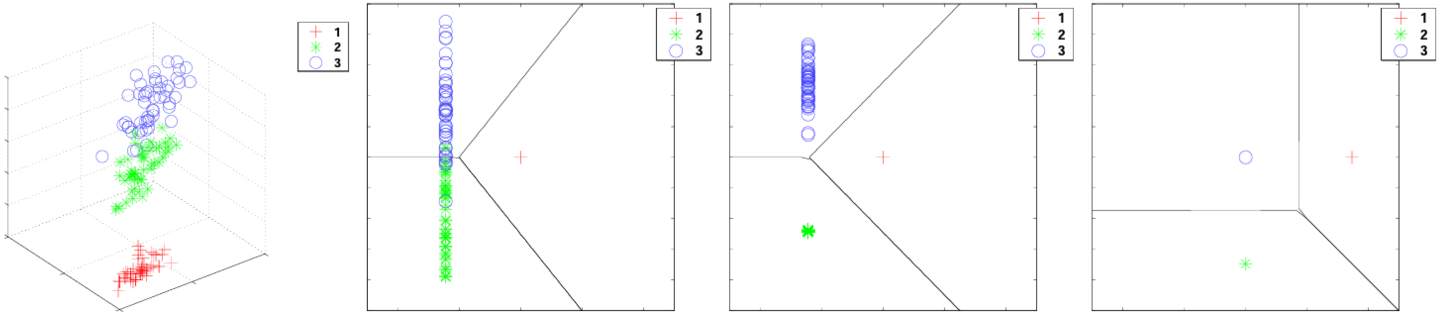
Where α, β are parameters and $\mathcal{D}$ is a set of pairwise distances.

Supervised Dimensionality Reduction

- This tends to exaggerate class separation
- Distance measure used in supervised LLE, ISOMAP, t-SNE, and Laplacian eigenmaps (right)



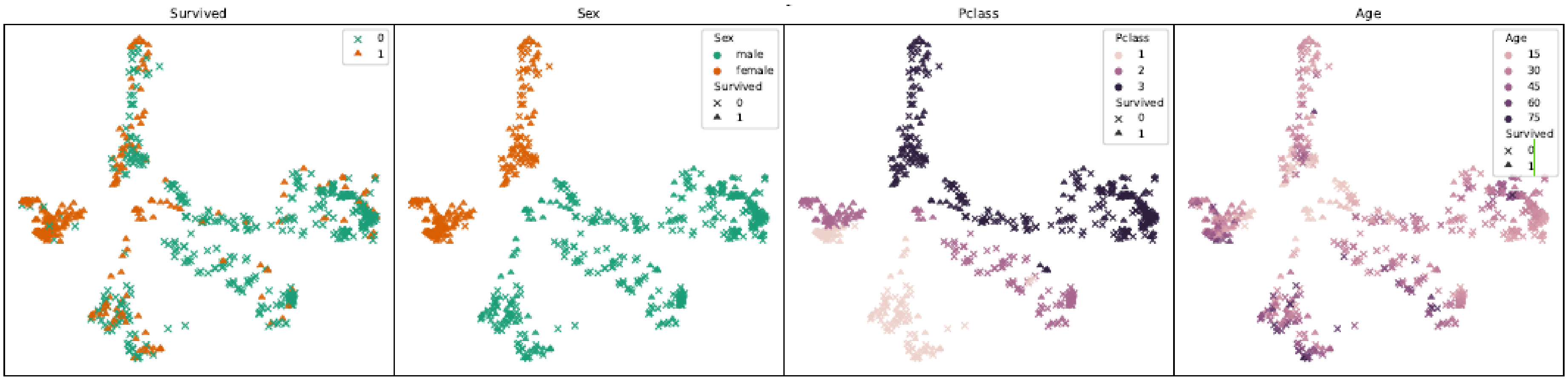
- Effect of α in S-LLE (de Ridder et al. 2003) (below)



(a) Original (b) LLE (c) 0.01-SLLE (d) 1-SLLE

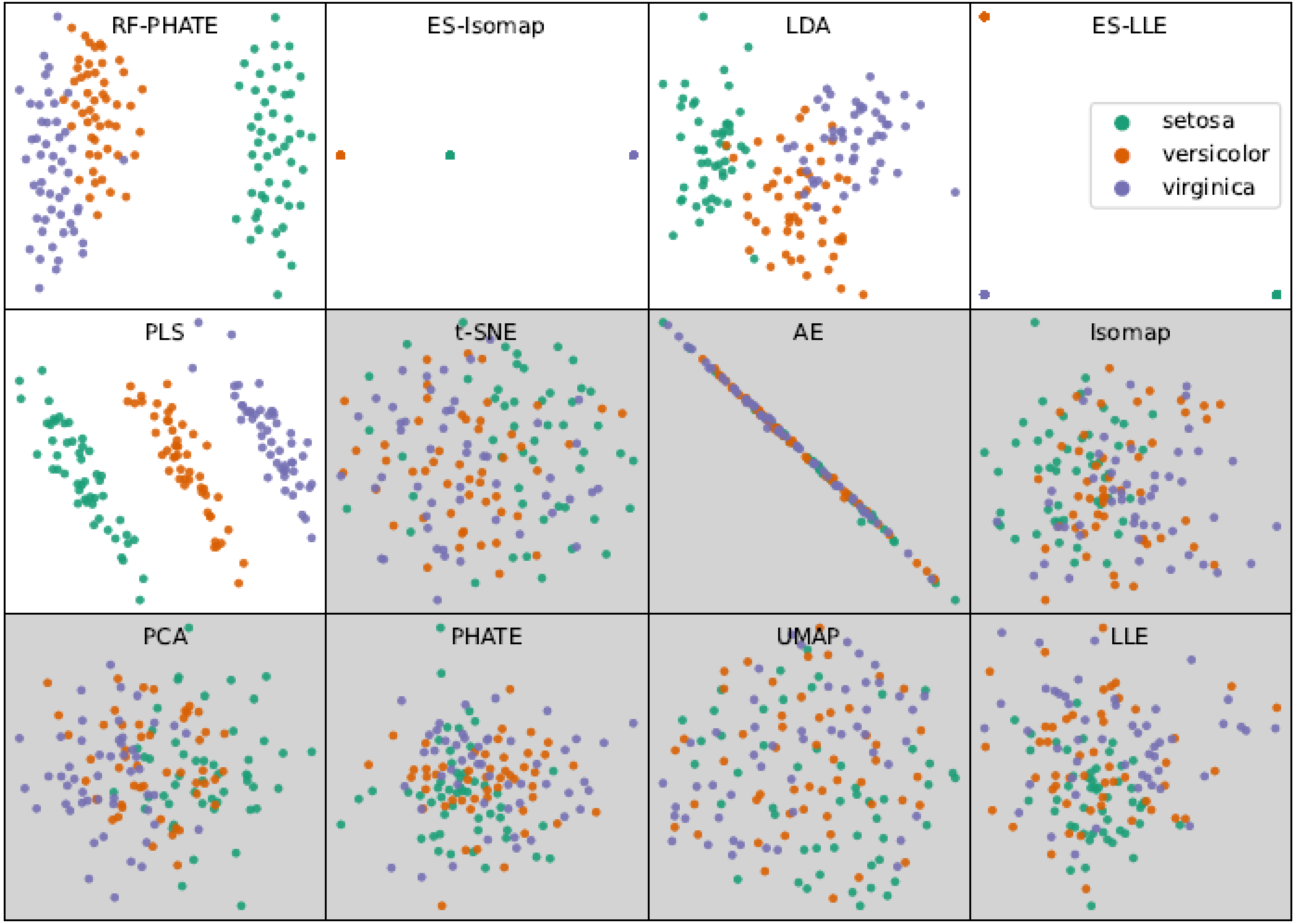
RF-PHATE

- Replace the α-decay kernel with RF-GAP proximities
  - Denoises the data while focusing on important variables
- RF-PHATE (Rhodes et al., 2021) applied to the Titanic dataset:



RF-PHATE

- Iris dataset with 500 noise variables added

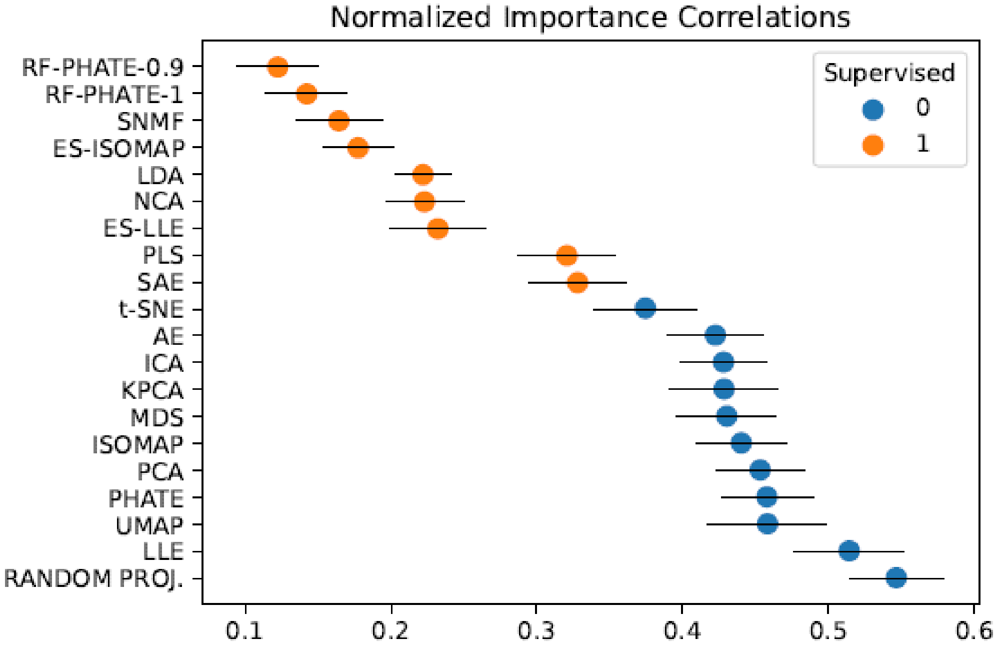


RF-PHATE

Supervised DR should capture important variable geometry

1. Compute k-nn permutation importance on original space
2. Compute k-nn permutation importance on embeddings
3. Determine correlation between importance scores
4. Normalize each correlation: $\bar{\rho} = \rho_{max} - \rho$

- Averaged across multiple datasets



(Rhodes et al, 2021)

Takeaways

- Random forests are generally the best out of box classifier
- Random forests have many other uses
- Proximities from random forests help us in a lot of applications
- RF-GAP captures the geometry learned by the random forest
- RF-PHATE outperforms other supervised DR methods
- We also applied RF-PHATE to multiple sclerosis clinical data
    - Applied to time series using dynamic time warping

Further reading

- Breiman (2001). Random forests. *Machine Learning.*
- Cutler et al. (2012). *Random Forests*, Springer US, Boston, MA.
- Rhodes et al. (2021). Random forest-based diffusion information geometry for supervised visualization and data exploration. *SSP*
- Rhodes et al. (2023). Random Forest Geometry and Accuracy Preserving Proximities. *IEEE Transactions PAMI*
- Rhodes et al. (2024). Gaining biological insights through supervised data visualization. *bioRxiv*