Introduction to Machine Learning for Survey Research

AAPOR Webinar

Christoph Kern[1]

c.kern@uni-mannheim.de

05/26/2022

[1]Thanks to Frauke Kreuter (LMU), Malte Schierholz (LMU)

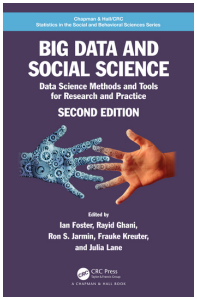
Outline

Machine Learning for Survey Research

Foster, I., Ghani, R., Jarmin, R. S., Kreuter, F., and Lane, J. (Eds.). (2020). *Big Data and Social Science: Data Science Methods and Tools for Research and Practice*. 2nd Edition. Provided by the Coleridge Initiative: `https://textbook.coleridgeinitiative.org/`

Kern, C., Klausch, T., and Kreuter, F. (2019). Tree-based Machine Learning Methods for Survey Research. *Survey Research Methods 13*(1), 73–93. `https://doi.org/10.18148/srm/2019.v1i1.7395`

Buskirk, T. D., Kirchner, A., Eck, A. Signorino, C. S. (2018). An Introduction to Machine Learning Methods for Survey Researchers. *Survey Practice 11*(1). `https://doi.org/10.29115/SP-2018-0004`.

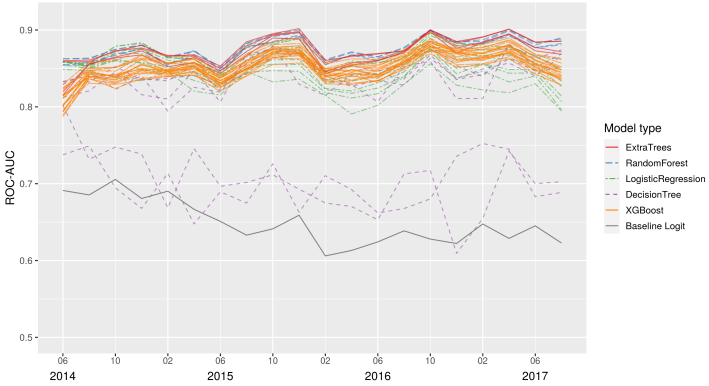


Why ML? Example 1a

Kern et al. (2021)

- Predict nonresponse in the next wave of a panel survey
- Investigate potential of moving from post- to pre-correction of panel dropouts
- Complex prediction problem; many attributes, many waves

Figure: Prediction performance of various models by wave

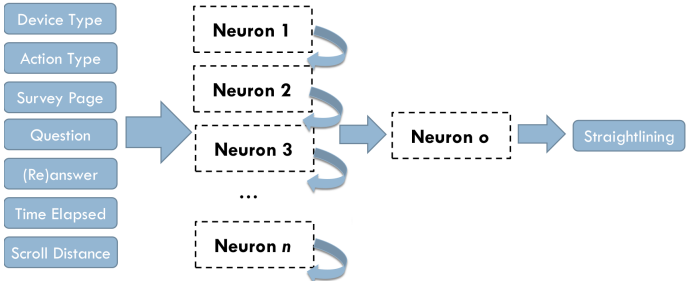


Why ML? Example 1b

Eck et al. (2015); Eck and Soh (2017)

- Prediction of respondent behaviors leading to (various) error in web surveys
- Predict survey breakoff, straightlining, speeding, item non-response
- Could ultimately be used to adapt surveys in real time

Figure: Recurrent Neural Networks for sequential learning



Why ML? Example 2

Schierholz et al. (2018)

- Automated occupation coding during the interview
- ML to predict candidate job categories based on initial answer (text data!)
- Objectives: Reduce coding errors, minimize coding after the interview, save interview time

Figure: Productivity of the coding system

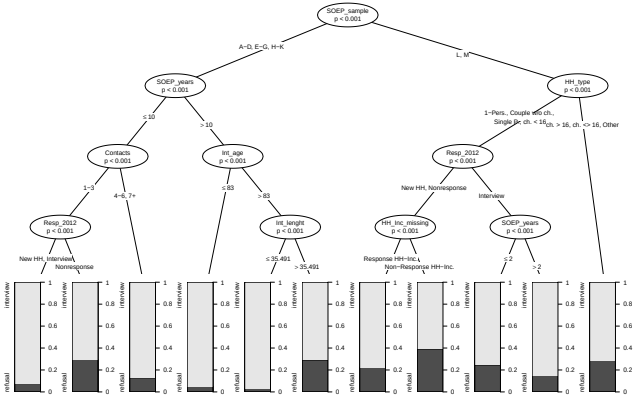
Number of respondents who give a job description	1064	100.0%	
Algorithm provides no job suggestion	106	10.0%	
Algorithm finds possible categories: thereof,	958	90.0%	
... Respondent chooses a job title	770		72.4%
... Respondent chooses 'other occupation'	145		13.6%
... Item non-response	3		0.3%
... Other experimental conditions	40		3.8%

Why ML? Example 3

Kern et al. (2019)

- Semi-exploratory modeling of nonresponse
- Did we miss important interactions (e.g., between respondent and interviewer characteristics)?
- Identify subgroups with unique effect patterns (Zeileis et al., 2008)

Figure: Conditional Inference Tree predicting interview refusal



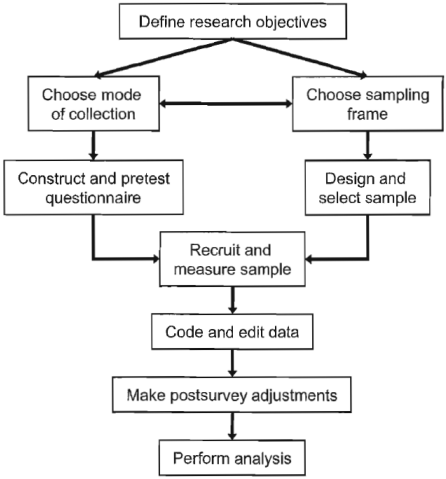
Machine Learning for Survey Research

1. **Prediction** as a means to an end (example 1a, 1b)
   - Informing adaptive survey designs
   - Predicting response propensities for weighting
   - Improving inference from non-probability samples
2. Utilizing high-dimensional **new data sources** (example 2)
   - Text data
   - Sensor data
   - Web-tracking and social media data
3. Accounting for **heterogeneity** (example 3)
   - Uncover interactions, unique subgroups
   - Model heterogeneous (treatment) effects

More Examples

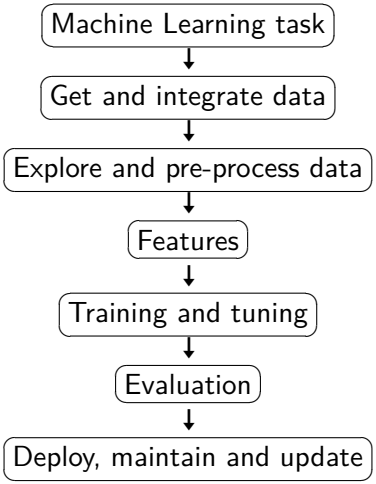
ML along the survey process to...

- Help creating sampling frame (Eckman and Qiu, 2019)
- Improve sampling design (Buskirk et al., 2018)
- Refine Survey Questions (Saris et al., 2011)
- Classify open-ended questions (Schonlau and Couper, 2016)
- Calculate pseudo-weights/ improve nonprob inference (Kern et al., 2020; Kim et al., 2022)
- ...



Foundations of Machine Learning

ML Process & Terminology



Supervised vs. Unsupervised

Unsupervised Learning

- Finding patterns in data using a set of input variables X

Supervised Learning

- Predicting an output variable Y based on a set of input variables X
  1. Learn the relationship between input and output using **training data** (with X and Y)

$$Y = f(X) + \varepsilon$$

  2. Predict the output based on the prediction model (of step 1) for **new test data** ($\sim$only X available)

Supervised vs. Unsupervised

Unsupervised Learning

- Finding patterns in data using a set of input variables X

Supervised Learning

- Predicting an output variable Y based on a set of input variables X
  1. Learn the relationship between input and output using **training data** (with X and Y)

$$Y = f(X) + \varepsilon$$

  2. Predict the output based on the prediction model (of step 1) for **new test data** ($\sim$only X available)

Supervised vs. Unsupervised

Unsupervised Learning

- Finding patterns in data using a set of input variables X

Supervised Learning

- Predicting an output variable Y based on a set of input variables X
  1. Learn the relationship between input and output using **training data** (with X and Y)

$$Y = f(X) + \varepsilon$$

  2. Predict the output based on the prediction model (of step 1) for **new test data** ($\sim$only X available)

Supervised Learning Basics

Supervised Learning: Find function $f(x)$ that makes optimal predictions in a **new data set**

Prerequisites:

- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.
- **Evaluation**: What is the criterion to choose between different functions?
  - Measures predictive performance
  - Examples: Mean Squared Error, Logistic Loss
- **Computation**: How is f actually calculated?
  - Speed and memory space may be limiting factors

Supervised Learning Basics

Supervised Learning: Find function $f(x)$ that makes optimal predictions in a **new data set**

Prerequisites:

- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.
- **Evaluation**: What is the criterion to choose between different functions?
  - Measures predictive performance
  - Examples: Mean Squared Error, Logistic Loss
- **Computation**: How is f actually calculated?
  - Speed and memory space may be limiting factors

Supervised Learning Basics

Supervised Learning: Find function $f(x)$ that makes optimal predictions in a **new data set**

Prerequisites:

- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.
- **Evaluation**: What is the criterion to choose between different functions?
  - Measures predictive performance
  - Examples: Mean Squared Error, Logistic Loss
- **Computation**: How is f actually calculated?
  - Speed and memory space may be limiting factors

Supervised Learning Basics

Supervised Learning: Find function $f(x)$ that makes optimal predictions in a **new data set**

Prerequisites:

- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.
- **Evaluation**: What is the criterion to choose between different functions?
  - Measures predictive performance
  - Examples: Mean Squared Error, Logistic Loss
- **Computation**: How is f actually calculated?
  - Speed and memory space may be limiting factors

ML vs. Regression?

Table: Fitting $f(x)$

Regression methods	(tree-based) ML methods
parametric	non-parametric
linearity, additivity	flexible functional form
prior model specification	"built-in" feature selection
theory-driven	data-driven
$\rightarrow$ Inference	$\rightarrow$ Prediction

Training and Test Error

Training error

- Prediction error based on **training data**
- How well did we fit the original data?

Test error

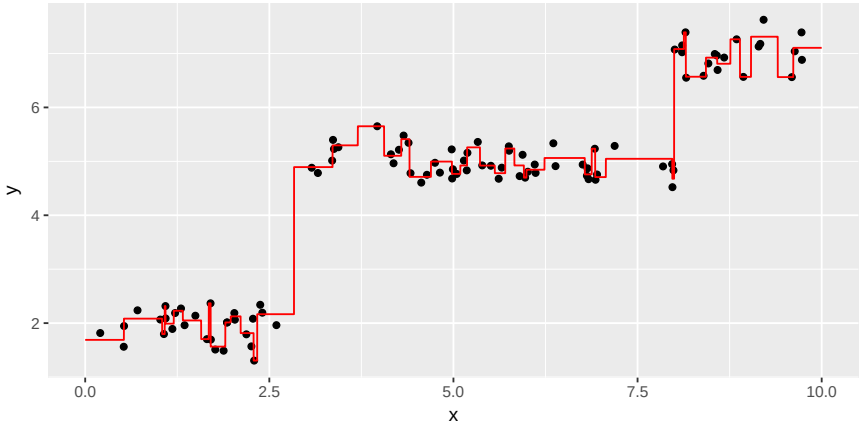
- Prediction error using test data
- Focus: How well do we perform for **new data**?

Test error decomposition

- Minimizing the test error requires
  - Low bias **and**
  - Low variance!

Bias-Variance Trade-Off

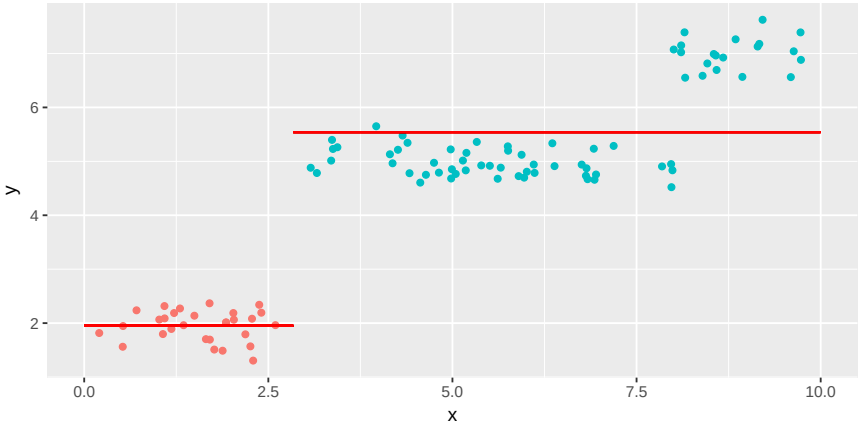
Figure: High Variance in Trees



- High Variance = Different data would lead to a different function
- Overfitting = Poor generalization to new data

Bias-Variance Trade-Off

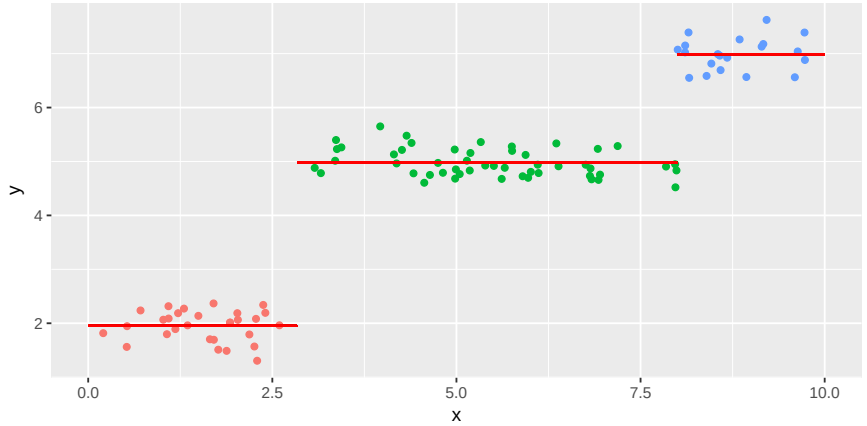
Figure: High Bias in Trees



- High Bias = Blue points are poorly predicted
- Underfitting = Function should adapt better to the data

Bias-Variance Trade-Off

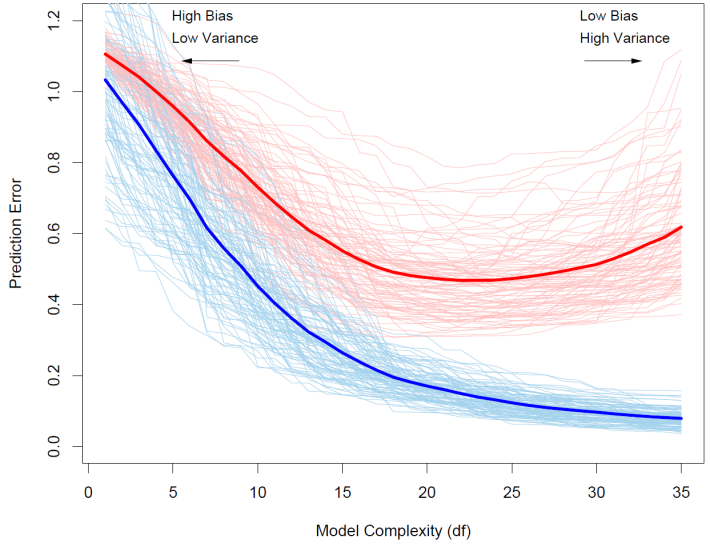
Figure: Optimal Solution



- Goal: Find optimal compromise between bias and variance

Bias-Variance Trade-Off

Figure: Training error and test error by model complexity (Hastie et al. 2009)



Bias-Variance Trade-Off

How to avoid overfitting?

Minimize loss in the training data while **restricting** the complexity of f

- Tree with at most K leaves
- Regression with $\sum |\beta_j| < K$
- General form: $\text{Penalty}(f) < K$

This is **regularization** – in general form:

$$\arg\min_{f \in \mathcal{F}} \text{Loss}(f) + \lambda \cdot \text{Penalty}(f)$$

Train-test splits, Cross-Validation

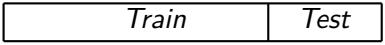
Validation Set Approach

Training set & test set

- Estimate prediction error on new data
  1. Fit model using one part of training data
  2. Compute test error for the excluded section

→ Model assessment

Figure: 80/20 train-test split

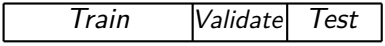


Training set, validation set & test set

- Compare models and estimate prediction error
  1. Fit models with training set
  2. Choose best model using validation set
  3. Evaluate final model using test set

→ Model selection & assessment

Figure: 50/25/25 Train-validation-test split



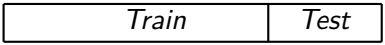
Validation Set Approach

Training set & test set

- Estimate prediction error on new data
  1. Fit model using one part of training data
  2. Compute test error for the excluded section

→ Model assessment

Figure: 80/20 train-test split

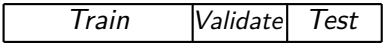


Training set, validation set & test set

- Compare models and estimate prediction error
  1. Fit models with training set
  2. Choose best model using validation set
  3. Evaluate final model using test set

→ Model selection & assessment

Figure: 50/25/25 Train-validation-test split



Leave test data untouched until the end of analysis!

Cross-Validation

LOOCV (Leave-One-Out Cross-Validation)

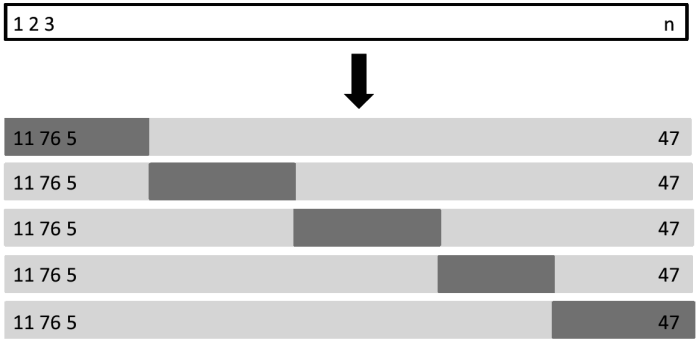
1. Fit model on training data while excluding one case
2. Compute test error for the excluded case
3. Repeat step 1 & 2 n times

k-Fold Cross-Validation

1. Fit model on training data while excluding one group
2. Compute test error for the excluded group
3. Repeat step 1 & 2 k times (e.g. $k = 5$, $k = 10$)

Cross-Validation

Figure: 5-Fold Cross-Validation with training set and validation set (James et al. 2013)



Cross-Validation

More on data splitting

- Simple random splits
  - General approach for "unstructured" data
  - Typically 75% or 80% go into training set
- Stratified splits
  - For classification problems with class imbalance
  - Sampling within each class of Y to preserve class distribution
- Splitting by groups
  - For (temporal) structured data
  - Use specific groups (temporal holdouts) for validation

Performance Evaluation

Performance Measures for Classification

Probabilities, thresholds and prediction for classification

$$y_i = \begin{cases} 1 & if \quad p_i > c \\ 0 & if \quad p_i \leq c \end{cases}$$

Table: Confusion matrix

		Prediction 0	Prediction 1	
Reference	0	True Negatives (TN)	False Positives (FP)	N'
	1	False Negatives (FN)	True Positives (TP)	P'
		N	P	

Performance Measures for Classification

Confusion matrix metrics

- Global performance
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$
  - No Information rate
- Row / column performance
  - Sensitivity (Recall): $\frac{TP}{TP+FN}$
  - Specificity: $\frac{TN}{TN+FP}$
  - Positive predictive value (Precision): $\frac{TP}{TP+FP}$

Probability-based metrics

- ROC-AUC
- PR-AUC

Table: Confusion matrix

		Prediction		
		0	1	
Reference	0	TN	FP	N'
	1	FN	TP	P'
		N	P	

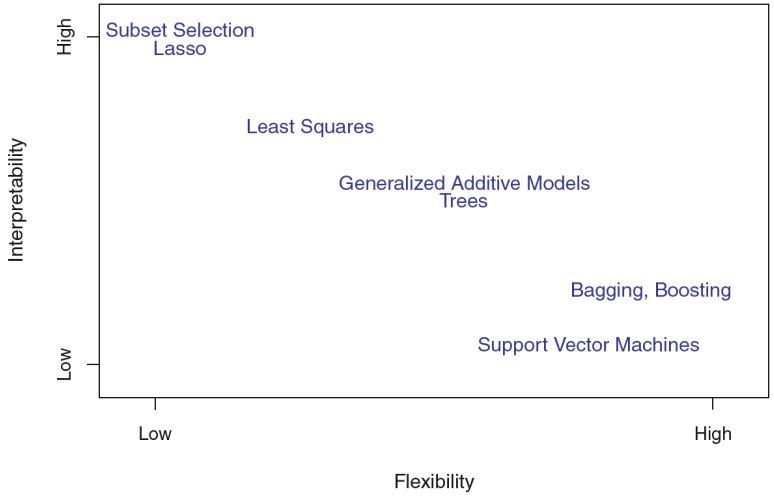
Summary

- Expected test error can be decomposed into bias and variance components
- Bias-Variance Trade-off represents decisive concept in ML
- Aim at model (setup) that generalizes well to new data (vs. over- and underfitting)
- Various types of Cross-Validation can be used for model selection and assessment
- Large number of performance metrics for classification available
- Important to compare against reference level (e.g., no information rate)

Machine Learning Methods

Machine Learning Methods

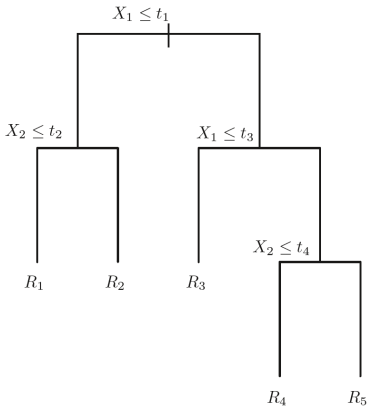
Figure: Flexibility-Interpretability Trade-Off (James et al. 2013)



Classification and Regression Trees (CART)

- Approach for partitioning the predictor space into smaller subregions via "recursive binary splitting"
- Results in a "top-down" tree structure with...
  - Internal nodes within the tree
  - Terminal nodes as endpoints
- Can be applied to regression and classification problems
- Important building block for ensemble methods

Figure: A small tree



Decision Tree Growing

Algorithm 1: Tree growing process

```
Define stopping criteria;
Assign training data to root node;
if stopping criterion is reached then
    end splitting;
else
    find the optimal split point;
    split node into two subnodes at this split point;
    for each node of the current tree do
        continue tree growing process;
    end
end
```

Ensembles

Some limitations of (single) trees

- Difficulties in modeling additive structures
- Lack of smoothness of prediction surface
- High variance / **instability** due to hierarchical splitting process

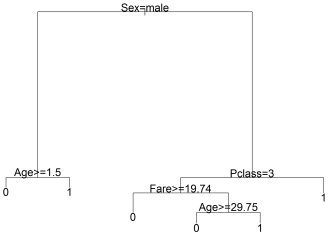
→ **Ensemble methods**

- Address instability via combining multiple prediction models
- Combine diverse models into a more robust ensemble

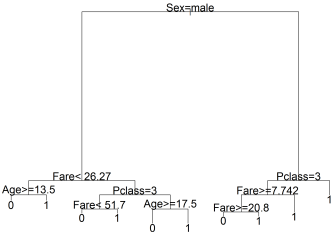
Bagging Trees

Figure: Growing Trees on Bootstrap Samples

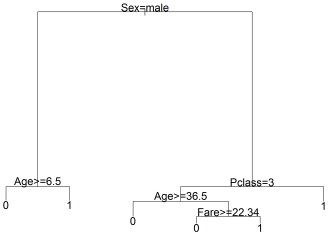
(a) b = 1



(b) b = 2



(c) b = 3



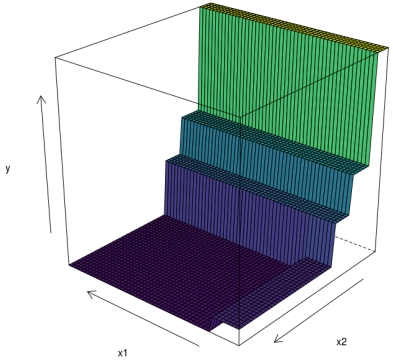
Algorithm 2: Grow a Random Forest

```
Set number of trees B;
Set predictor subset size m;
Define stopping criteria;
for b = 1 to B do
    draw a bootstrap sample from the training data;
    assign sampled data to root node;
    if stopping criterion is reached then
        end splitting;
    else
        draw a random sample m from the p predictors;
        find the optimal split point among m;
        split node into two subnodes at this split point;
        for each node of the current tree do
            continue tree growing process;
        end
    end
end
```

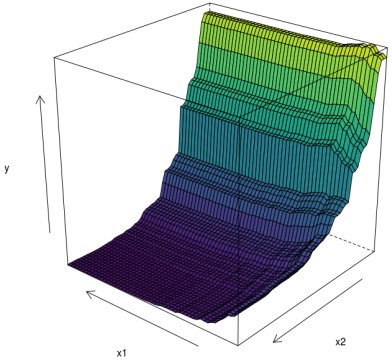
RF vs. CART

Figure: Prediction surface (example)

(a) CART



(b) Random Forest



Summary

- Decision Trees: Divide-and-conquer strategy that splits the data into subgroups
- No need to specify the functional form in advance (unlike regression)
- Non-linearities and interactions are handled automatically
- Limitations of (single) trees: Instability, competition among correlated predictors, biased variable selection
- Bagging, random forest stabilize predictions from high-variance methods (e.g., CART)

Further Readings

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York, NY: Springer.

Machine Learning in Practice

General Recommendations

Model tuning, selection and evaluation can be tricky

- Acknowledge any grouping/temporal structure in data splitting
- Include any feature selection in cross-validation pipeline
- Use test data only for final model evaluation
- Compare classification accuracy against a meaningful baseline
- Default classification threshold (0.5) may be useless for imbalanced outcomes

$\rightarrow$ Use meta packages to code safe and reliable ML pipelines

Resources

Resources for R – ML (meta) packages

- Overview
  - `https://cran.r-project.org/web/views/MachineLearning.html`
- caret
  - `http://topepo.github.io/caret/index.html`
- tidymodels
  - `https://www.tidymodels.org/`
- mlr3
  - `https://mlr3.mlr-org.com/`

Resources

Resources for R – Tree-based methods

- Standard package to build CARTs: `rpart`
- Unified infrastructure for tree representation: `partykit`
- Standard package to grow RFs: `randomForest`
- Fast implementation of RFs: `ranger`

Resources

Machine Learning for Social Science Tutorials

https://github.com/kimbrianj/mlforsocialscience

R tutorials for ML Basics, kNN, regularized regression, decision trees, random forests, boosting...

References I

Buskirk, T. D., Bear, T., and Bareham, J. (2018). Machine made sampling designs: Applying machine learning methods for generating stratified sampling designs. Paper presented at the Big Data Meets Survey Science Conference, Barcelona, Spain.

Eck, A. and Soh, L. K. (2017). Sequential prediction of respondent behaviors leading to error in web-based surveys. Paper presented at the 72nd Annual Conference of the American Association for Public Opinion Research, New Orleans, LA.

Eck, A., Soh, L. K., and McCutcheon, A. L. (2015). Predicting breakoff using sequential machine learning methods. Paper presented at the 70th Annual Conference of the American Association for Public Opinion Research, Hollywood, FL.

Eckman, S. and Qiu, Q. (2019). Detecting housing units from satellite imagery with computer vision. Paper presented at the Annual Conference of the American Association for Public Opinion Research.

Kern, C., Klausch, T., and Kreuter, F. (2019). Tree-based machine learning methods for survey research. *Survey Research Methods*, 13(1):73–93.

Kern, C., Li, Y., and Wang, L. (2020). Boosted Kernel Weighting – Using Statistical Learning to Improve Inference From Nonprobability Samples. *Journal of Survey Statistics and Methodology*. smaa028.

References II

Kern, C., Weiß, B., and Kolb, J.-P. (2021). Predicting Nonresponse in Future Waves of A Probability-Based Mixed-Mode Panel With Machine Learning. *Journal of Survey Statistics and Methodology*.

Kim, M. P., Kern, C., Goldwasser, S., Kreuter, F., and Reingold, O. (2022). Universal adaptability: Target-independent inference that competes with propensity scoring. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 119(4).

Saris, W. E., Oberski, D., Revilla, M., Zavala-Rojas, D., Lilleoja, L., Gallhofer, I., and Gruner, T. (2011). The development of the program sqp 2.0 for the prediction of the quality of survey questions. Technical report, RECSM Working Paper 24.

Schierholz, M., Gensicke, M., Tschersich, N., and Kreuter, F. (2018). Occupation coding during the interview. *Journal of the Royal Statistical Society: Series A*, 181(2):379–407.

Schonlau, M. and Couper, M. (2016). Semi-automated categorization of open-ended questions. *Survey Research Methods*, 10(2):143–152.

Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514.