

Machine Learning

Data Train Workshop
October 2021

Marvin N. Wright

wright@leibniz-bips.de

Lukas Burk

burk@leibniz-bips.de

Introduction

Example: House Prices

Predict the price for a house in a certain area

Features x				Target y
square footage of the house	number of bedrooms	swimming pool (yes/no)	...	house price in US\$
1,180	3	0	...	221,900
2,570	3	1	...	538,000
770	2	0	...	180,000
1,960	4	1	...	604,000



Introduction

Example: Length of hospital stay

Predict days a patient has to stay in hospital

Features x					Target y
diagnosis category	admission type	gender	age	...	Length-of-stay in the hospital in days
heart disease	elective	male	75	...	4.6
injury	emergency	male	22	...	2.6
psychosis	newborn	female	0	...	8
pneumonia	urgent	female	67	...	5.5



Introduction

Example: Life Insurance

Predict risk category for a life insurance customer

Features x				Target y
job type	age	smoker	...	risk group
carpenter	34	1	...	3
stuntman	25	0	...	5
student	23	0	...	1
white-collar worker	39	0	...	2



Introduction

Learning outcomes

- Understand basic concepts of machine learning
 - Supervised learning
 - Models and learners
 - Overfitting and underfitting
 - Hyperparameter tuning
 - Performance evaluation
- Know the major machine learning methods
 - k-nearest neighbors
 - Decision trees
 - Random forests
 - Boosting
 - Support vector machines
 - Artificial neural networks
- Be able to perform machine learning analyses in R

Outline (Day 1)

Morning

1. k-nearest neighbors (kNN)
2. General concepts
3. Decision trees

Afternoon

4. Random forests
5. Model evaluation and resampling
6. Boosting

Outline (Day 2)

Morning

7. Support vector machines
8. Hyperparameter tuning and performance comparison
9. Artificial neural networks

Afternoon

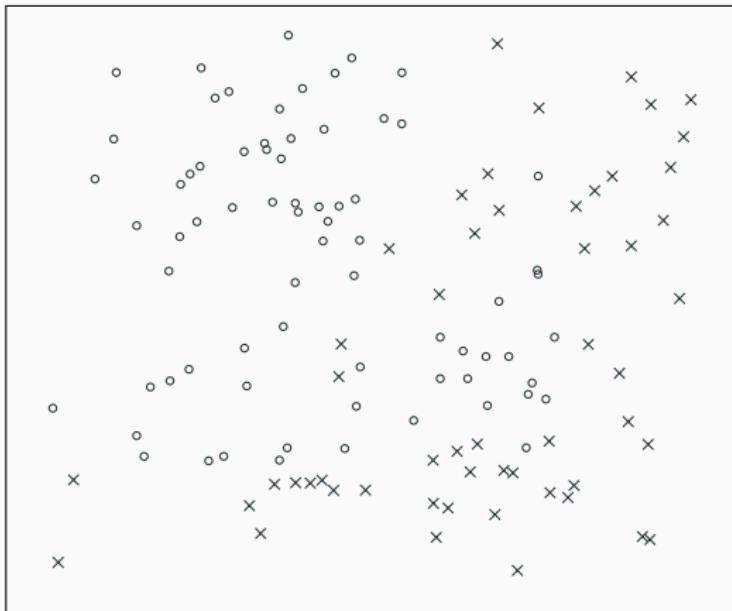
10. Specific endpoints
11. Variable importance and selection
12. Discussion

Part 1

k-nearest neighbors (kNN)

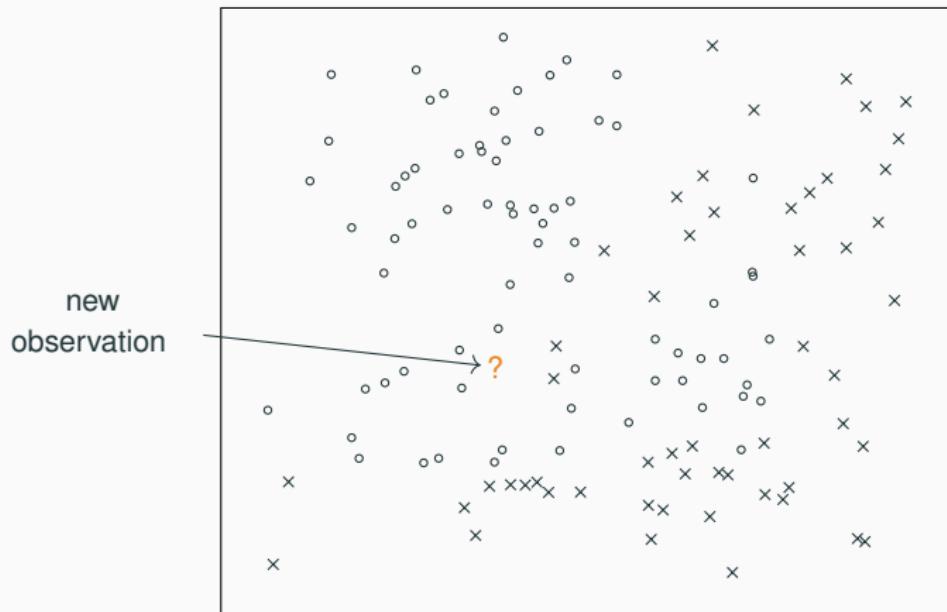
k-nearest neighbors (kNN)

What is k-nearest neighbors (kNN)?



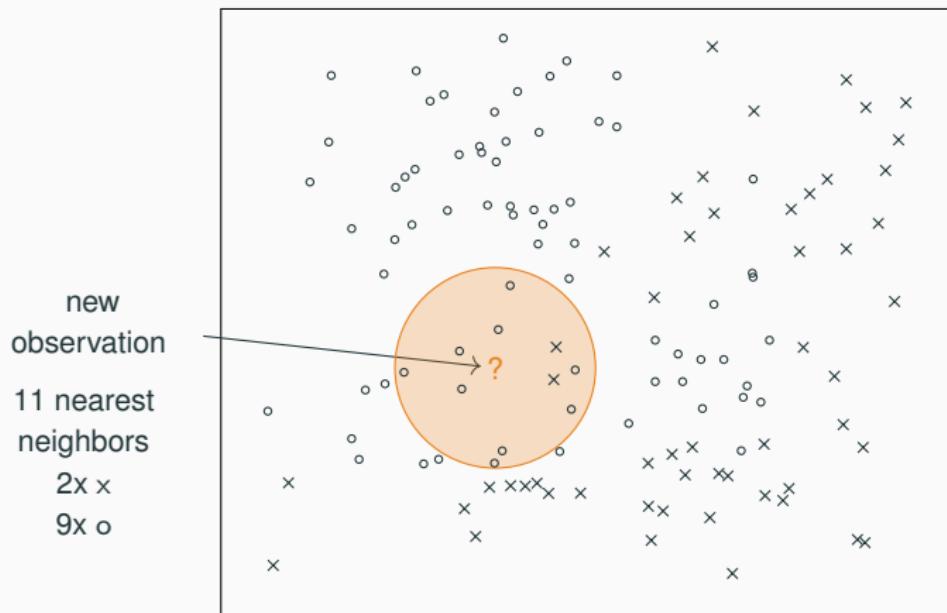
k-nearest neighbors (kNN)

What is k-nearest neighbors (kNN)?



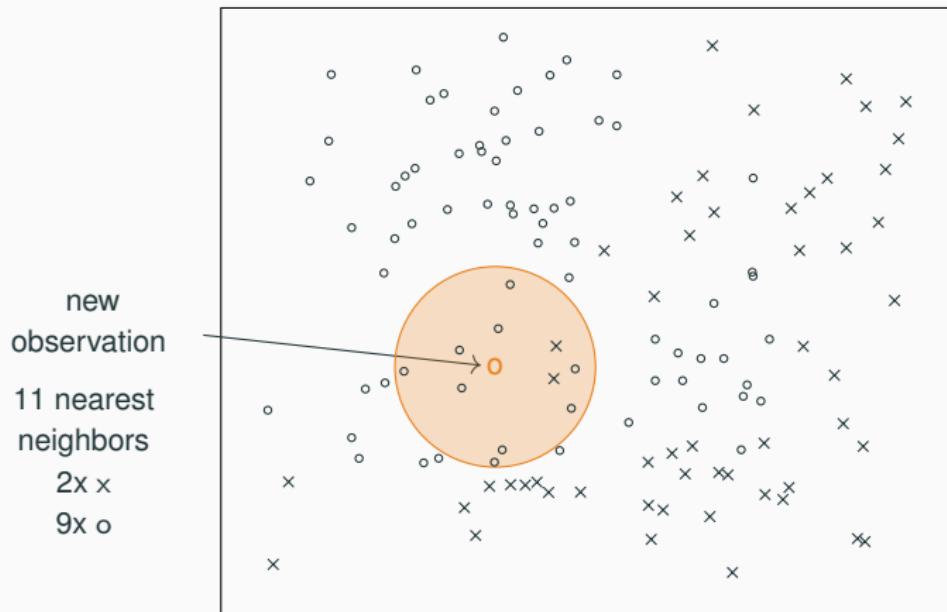
k-nearest neighbors (kNN)

What is k-nearest neighbors (kNN)?



k-nearest neighbors (kNN)

What is k-nearest neighbors (kNN)?



k-nearest neighbors (kNN)

What is kNN formally?

- $N_k(\mathbf{x})$ neighborhood of \mathbf{x} defined by k closest points \mathbf{x}_i in training data
- $\hat{y} = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y_i$
- Closeness implies metric
- Standard metrics: Euclidian, Mahalanobis distance
- Generalization: Weighting schemes, e.g. $w = \frac{1}{d(x, x_i)}$
- kNN assumes: Regression function $\mathbb{E}(y | \mathbf{x})$ well approximated by locally constant function

k-nearest neighbors (kNN)

What are the statistical properties of kNN?

- **Universal consistency (Stone's theorem)**

kNN is universally consistent if $k \rightarrow \infty, n \rightarrow \infty, k/n \rightarrow 0$

- **Convergence rate**

Under suitable regularity conditions the rate of convergence of kNN is faster than the minimax rate

- **Asymptotic normality**

Under suitable regularity conditions the kNN estimator is asymptotically normal

k-nearest neighbors (kNN)

What are the advantages of kNN?

- Fast training
- Simple and easy to understand
- Robust to noisy training data
- Effective if training sample size is large
- Many generalizations available

k-nearest neighbors (kNN)

What are the disadvantages of kNN?

- No rule to transfer to new data
- Results sensitive to choice of metric
- Tuning of k required → discussed later
- Works not well in high dimensional problems

Part 2

General concepts

General concepts

What is supervised learning?

Learn a functional relationship between **features** x and **target** y

Features x		Target y
People in Office (Feature 1) x_1	Salary (Feature 2) x_2	Worked Minutes Week (Target Variable)
4	4300 €	2220
12	2700 €	1800
5	3100 €	1920

$n = 3$

$p = 2$

$x_1^{(2)}$

$x_2^{(1)}$

$y^{(3)}$

General concepts

What is supervised learning?

Use labeled data to learn a model f

Use model f to predict target y of new data



General concepts

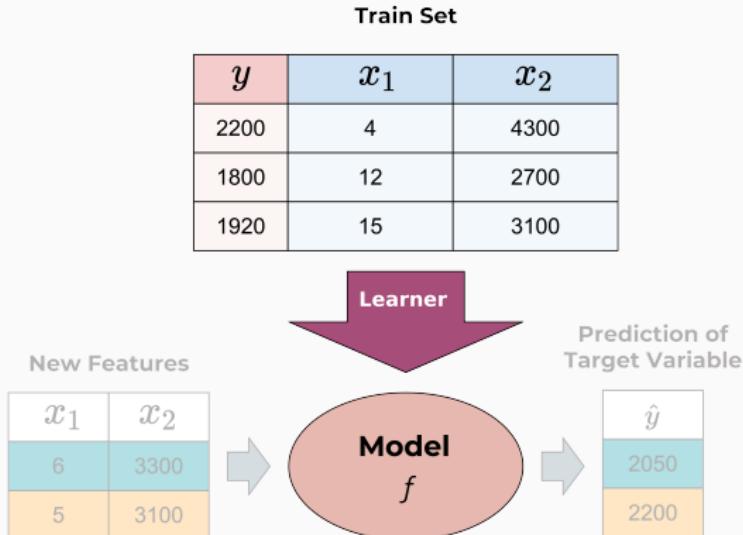
What is the difference between model and learner?

Model

Functional relationship between **features** x and **target** y

Learner (or inducer)

Algorithm for finding model



General concepts

Example

- Learner: Artificial neural network (as a concept)
- Model: Actual network with learned weights

Models differ in size and complexity

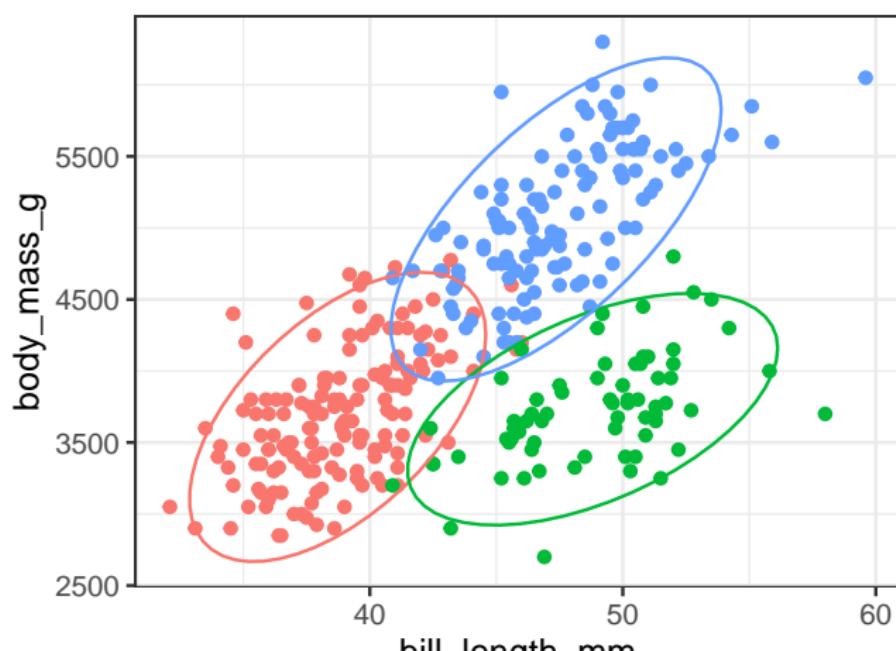
- Linear model: Coefficients β
- Neural network: Weights for all units in all layers
- Decision trees: Many binary splits
- k-nearest neighbors: Complete training data

General concepts

What is unsupervised learning?

No **target** y available

Search for patterns in the data x , e.g. clustering:

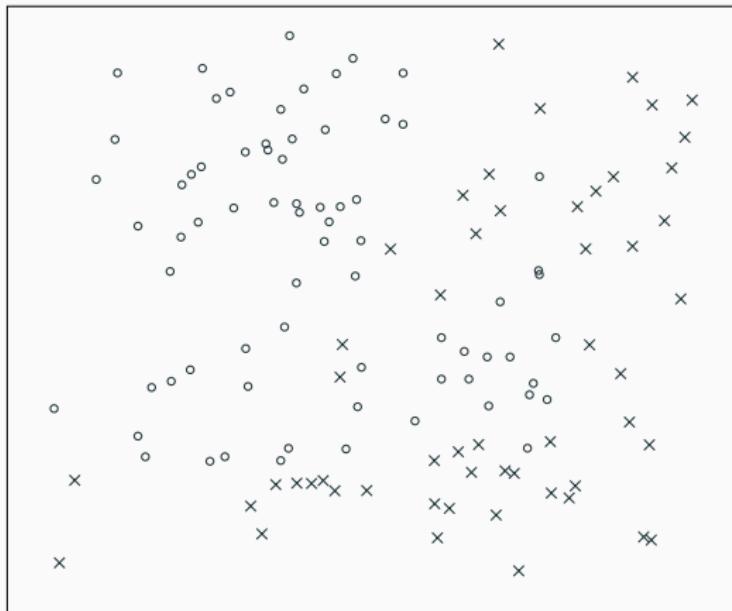


Part 3

Decision trees

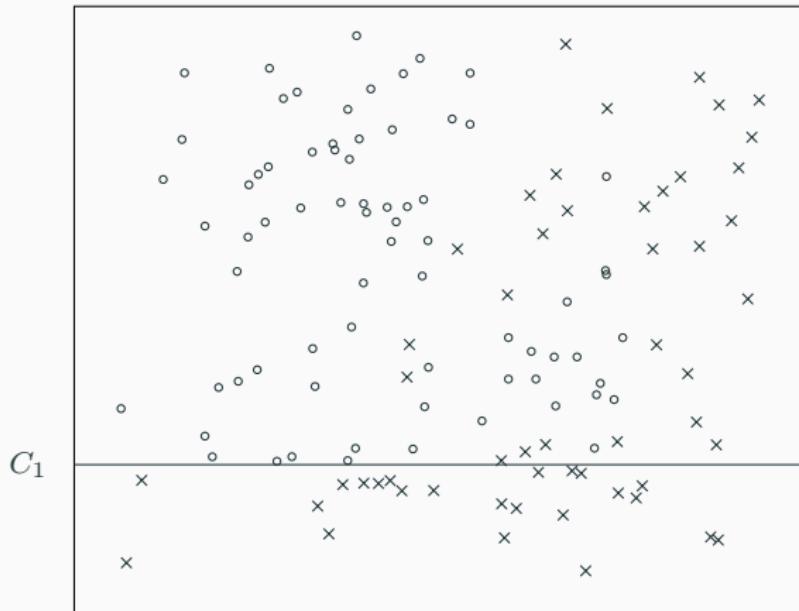
Classification and probability estimation trees

What are classification and probability estimation trees?



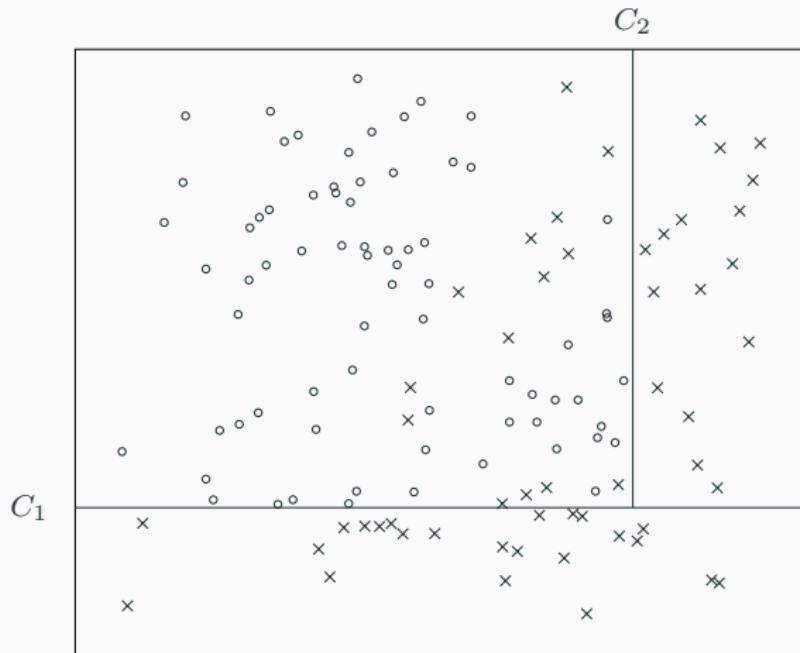
Classification and probability estimation trees

What are classification and probability estimation trees?



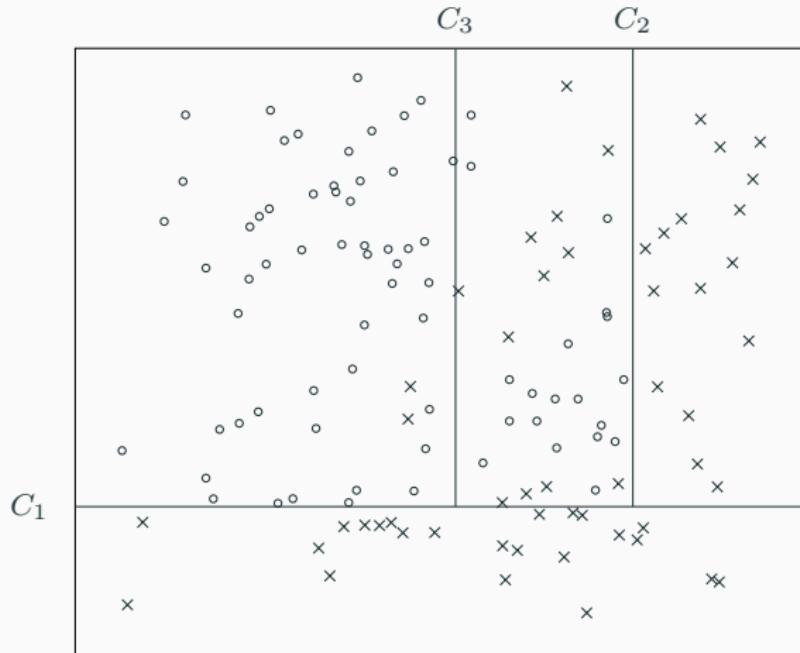
Classification and probability estimation trees

What are classification and probability estimation trees?



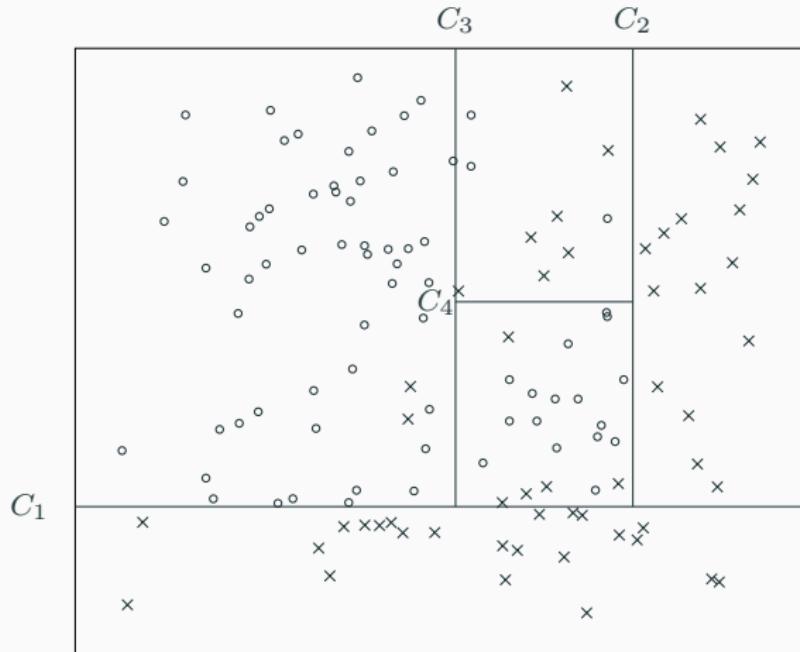
Classification and probability estimation trees

What are classification and probability estimation trees?



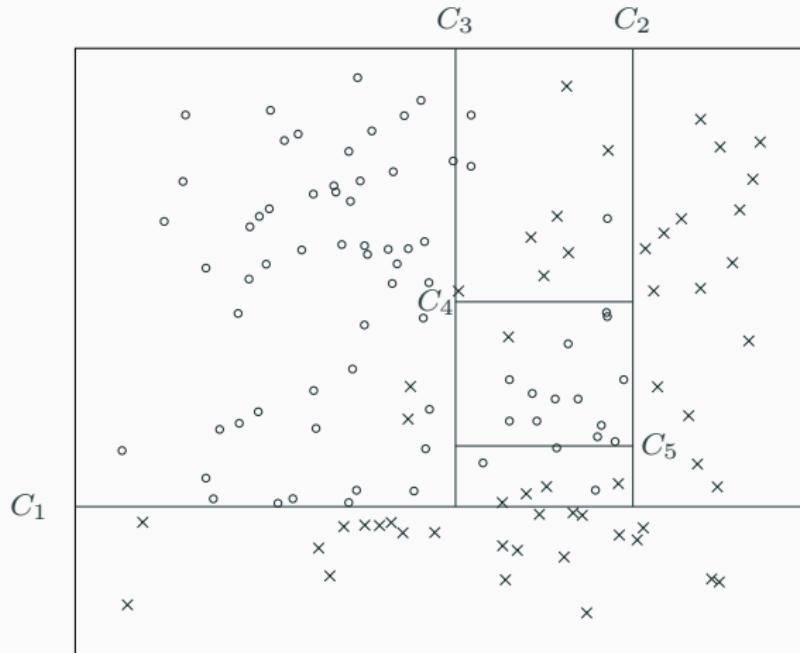
Classification and probability estimation trees

What are classification and probability estimation trees?



Classification and probability estimation trees

What are classification and probability estimation trees?



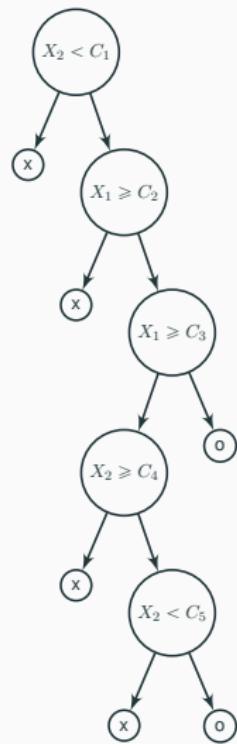
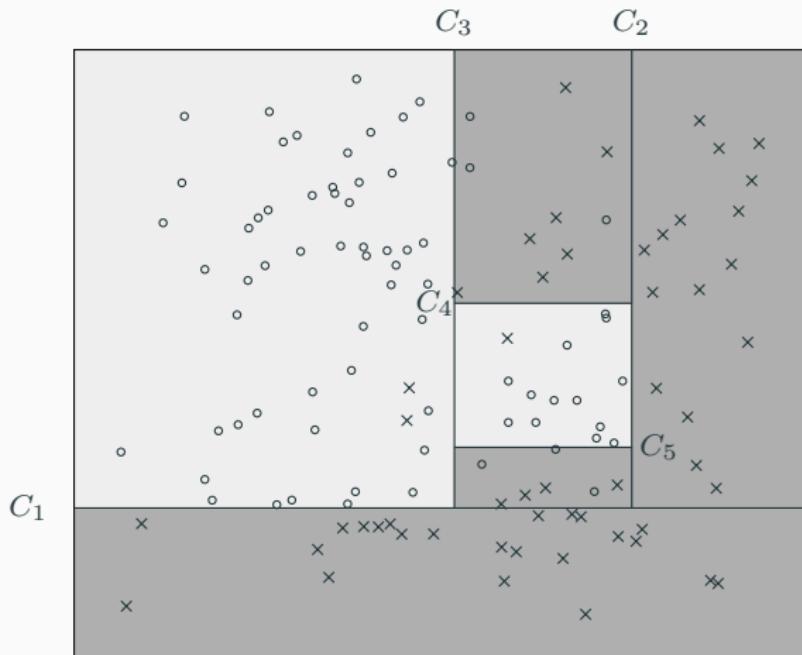
Classification and probability estimation trees

What are classification and probability estimation trees?



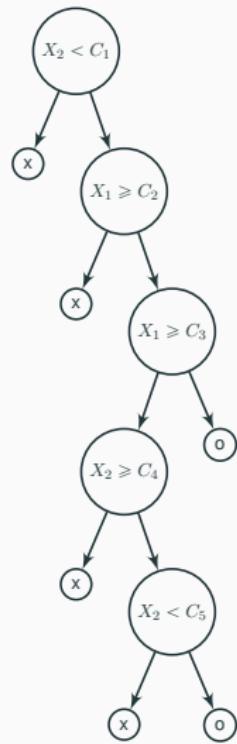
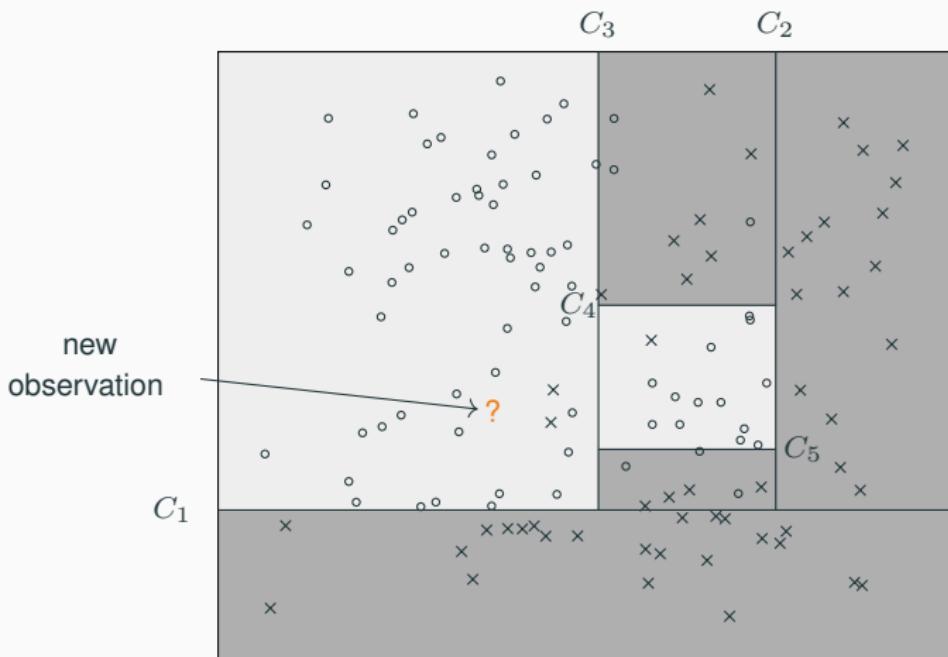
Classification and probability estimation trees

What are classification and probability estimation trees?



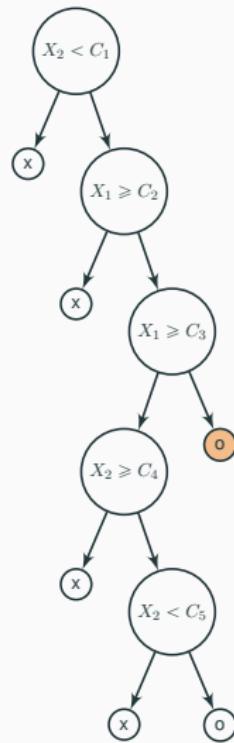
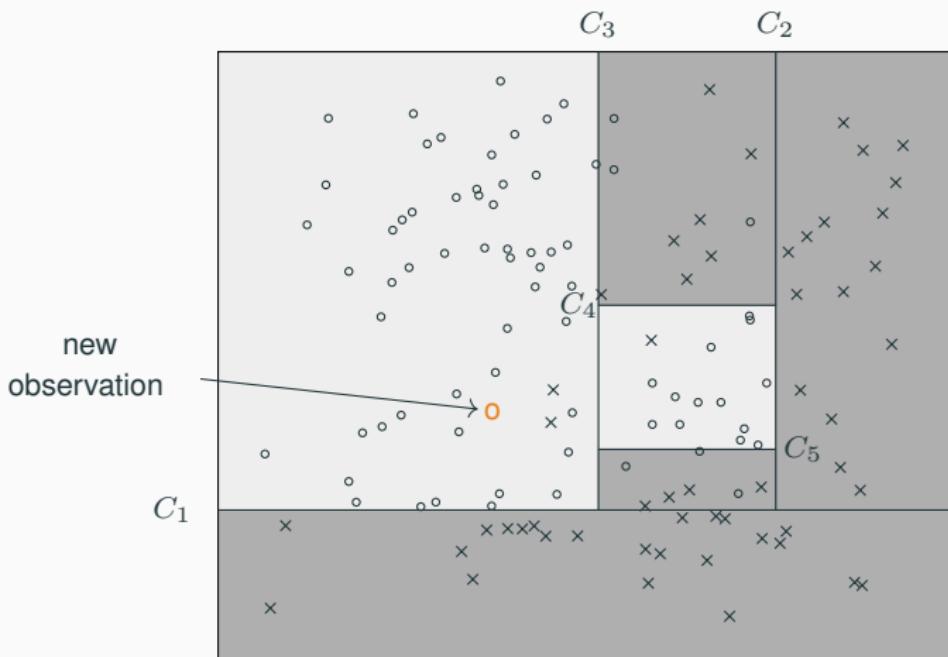
Classification and probability estimation trees

What are classification and probability estimation trees?



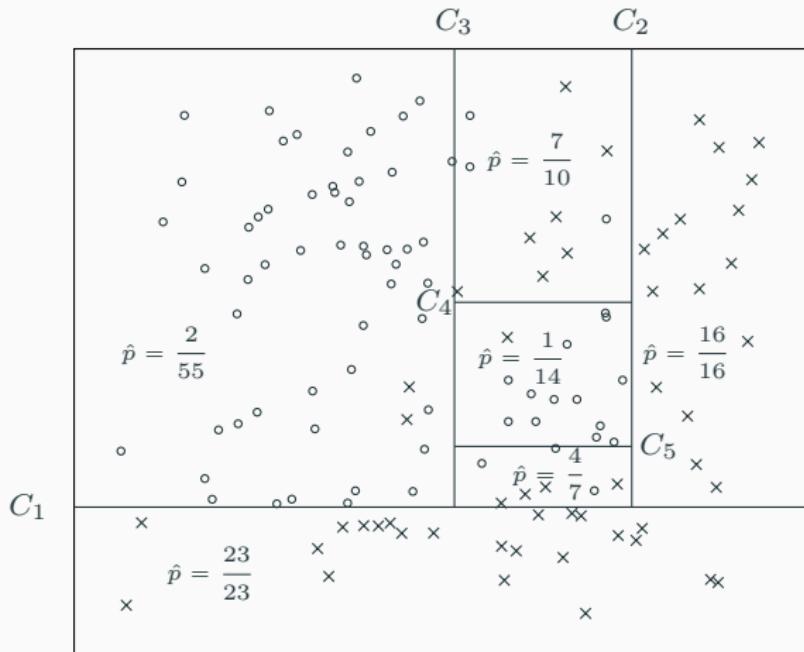
Classification and probability estimation trees

What are classification and probability estimation trees?



Classification and probability estimation trees

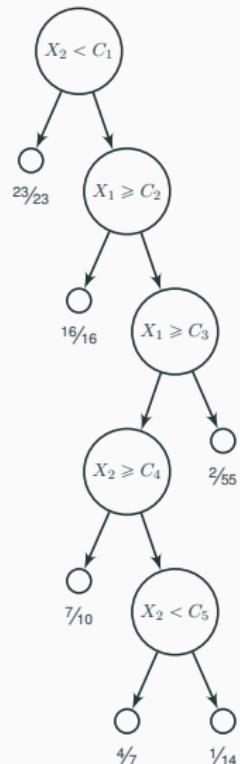
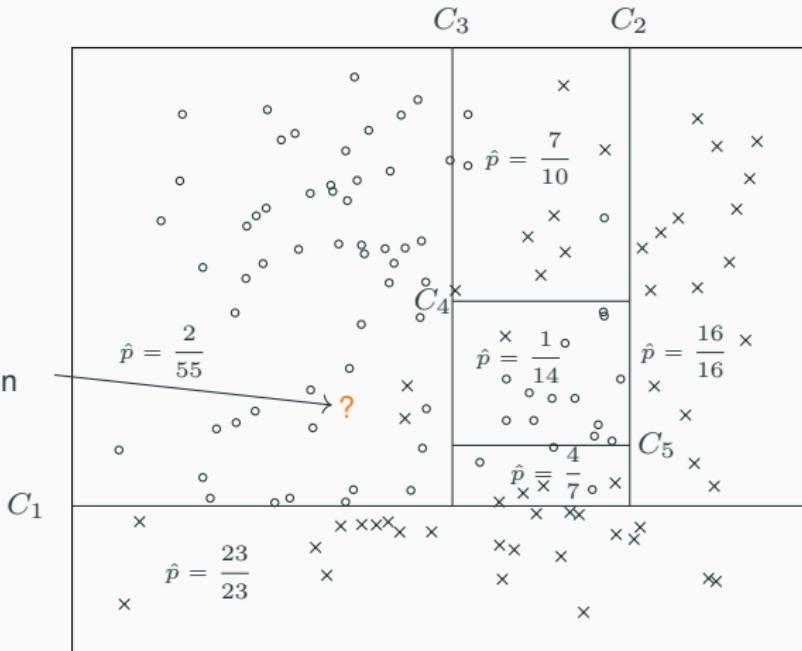
What are classification and probability estimation trees?



Classification and probability estimation trees

What are classification and probability estimation trees?

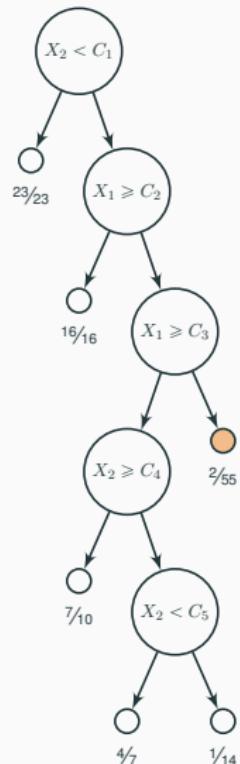
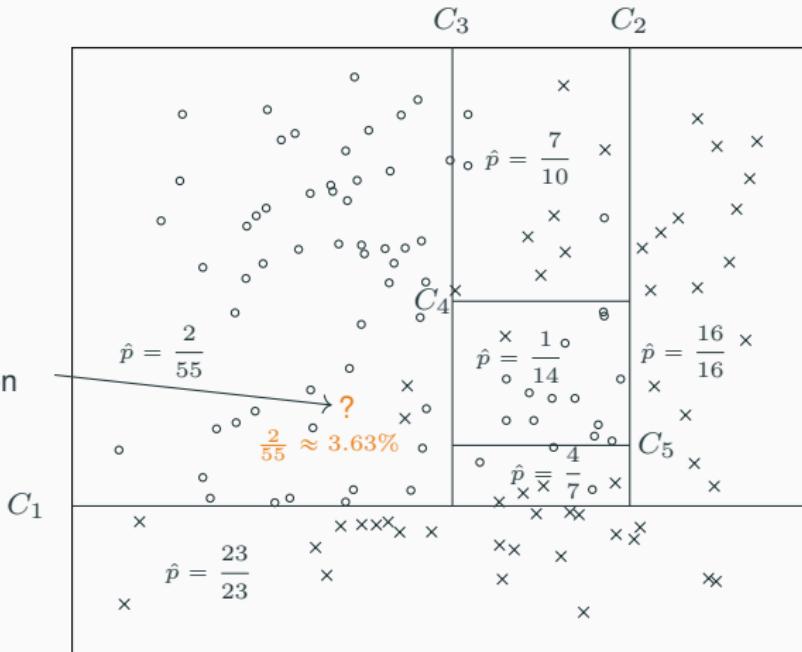
new observation



Classification and probability estimation trees

What are classification and probability estimation trees?

new observation



Classification and probability estimation trees

What is the standard algorithm for classification and probability estimation trees?

1. Grow tree
2. Stop tree growing process
3. Prune back branches
4. Select optimal tree

Classification and probability estimation trees

What is the standard algorithm for classification and probability estimation trees?

1. Grow tree : Recursively split nodes
 - a) For each independent variable x_j , consider each possible binary split (partition), compute child node impurity
 - b) Select variable x_j and split point yielding largest decrease in impurity
 - c) Split in exactly two child nodes at optimal split point
2. Stop tree growing process
3. Prune back branches
4. Select optimal tree

Classification and probability estimation trees

What is the standard algorithm for classification and probability estimation trees?

1. Grow tree
2. Stop tree growing process
 - a) In a node with only identical outcome (pure node)
 - b) In a node with only identical variable values (no split possible)
 - c) If external limit on tree complexity, tree depth or node size reached
3. Prune back branches
4. Select optimal tree

Classification and probability estimation trees

What is the standard algorithm for classification and probability estimation trees?

1. Grow tree
2. Stop tree growing process
3. Prune back branches
 - a) Tradeoff between complexity and accuracy
 - b) Estimate accuracy in test data set
 - c) Time consuming
4. Select optimal tree

Classification and probability estimation trees

What are standard split criteria for classification trees?

Measures of impurity for classification trees

- Misclassification error: $1 - \max(p, 1 - p)$
- Gini index: $2p(1 - p)$
- Deviance: $-p \cdot \ln(p) - (1 - p) \cdot \ln(1 - p)$

p : proportion of subjects with $y = 1$ in node

Dichotomous probability estimation

Gini index splitting and variance splitting identical

What is the standard algorithm for classification and probability estimation trees?

Number of possible splits for a variable

- Dichotomous categorical variable, e.g., sex:
 1 possible split point
- Ordered categorical variable, e.g., age in years:
 $m - 1$ possible split points
- Continuous independent variable, e.g., biomarker:
 $n - 1$ possible split points
- Unordered categorical variable, e.g., hair color:
 $2^m - 1$ possible split points

Classification and probability estimation trees

How can the computational complexity be reduced for unordered categorical variables?

- Idea: separate high case prob. categories from low case prob. categories
- Order categories by proportion of 1's
- Split as ordered categorical variable
- Split points shown to be optimal for binary classification and regression
- Approximations for multiclass classification available

Classification and probability estimation trees

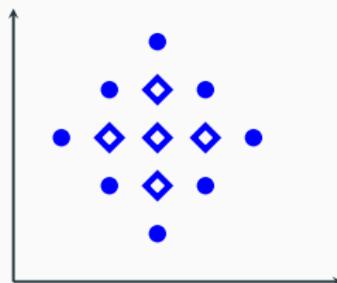
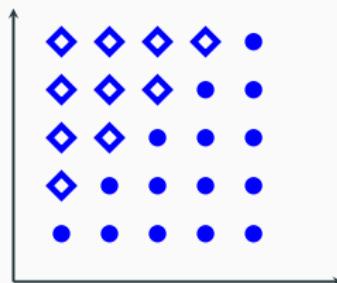
What are the advantages of classification and probability estimation trees?

- Procedure intuitive
- Small trees simple to interpret
- Intrinsic variable selection
- Simple handling of outliers
- Fast training
- Usually better prediction performance than kNN

Classification and probability estimation trees

What are the disadvantages of classification and probability estimation trees?

- Trees unstable
- Pruning can be computationally intensive
- Usually worse prediction performance than random forests and boosted trees (covered later)
- Problematic data sets



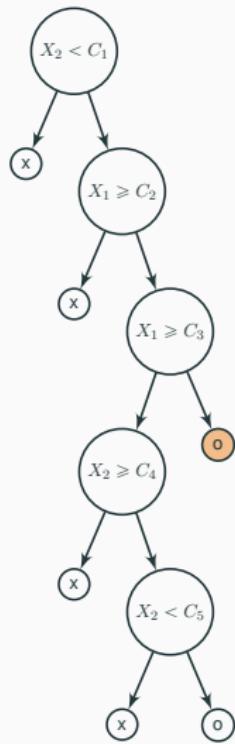
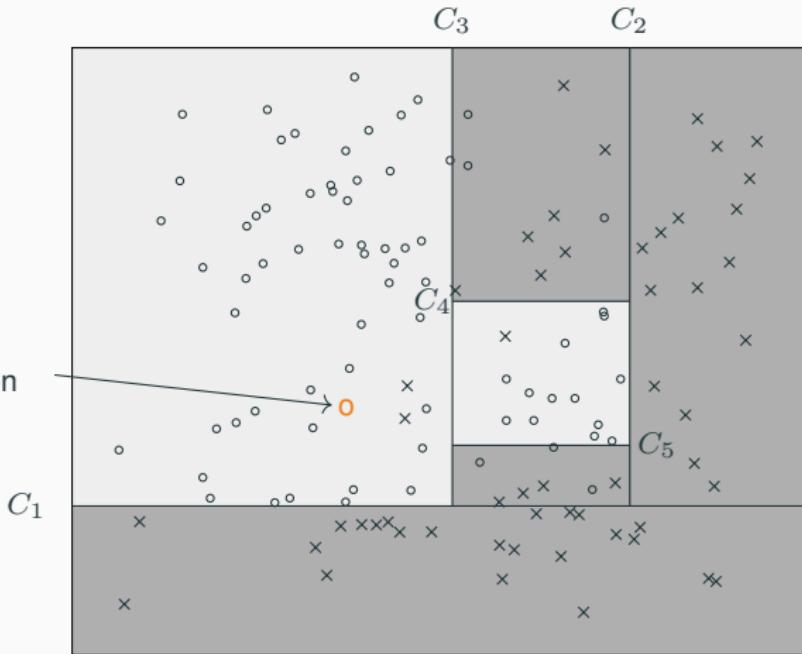
Part 4

Random forests

Random forests

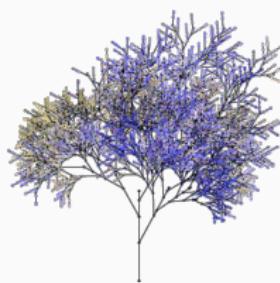
What are classification and probability estimation trees?

new
observation



Random forests

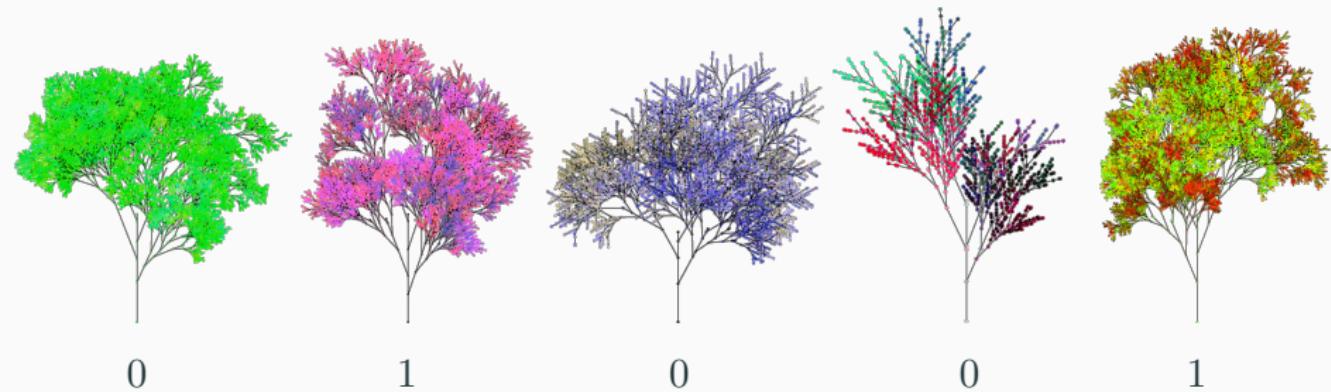
What are random forests?



0

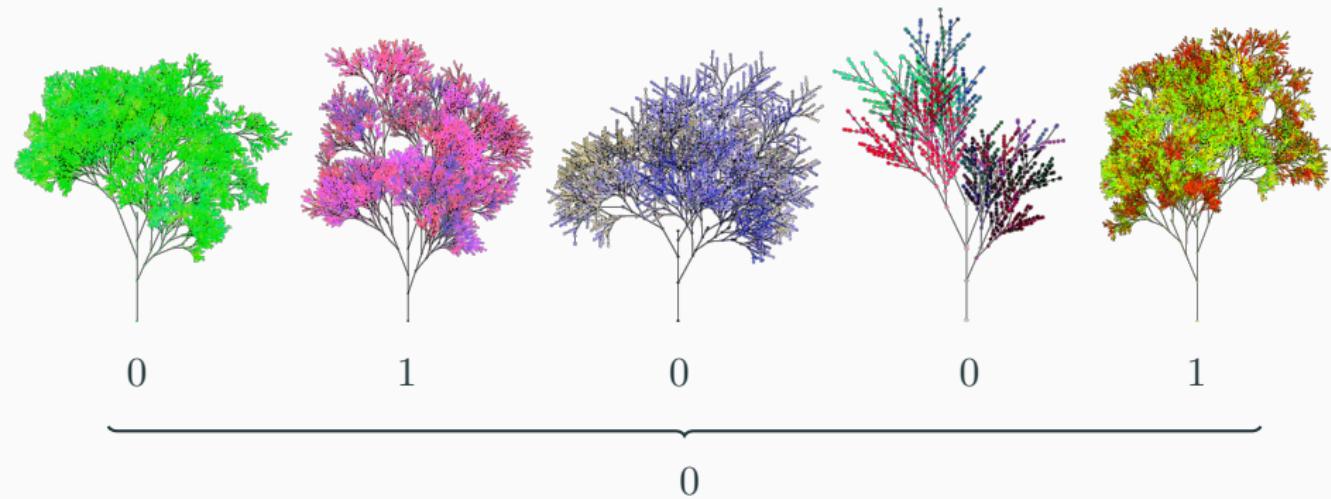
Random forests

What are random forests?



Random forests

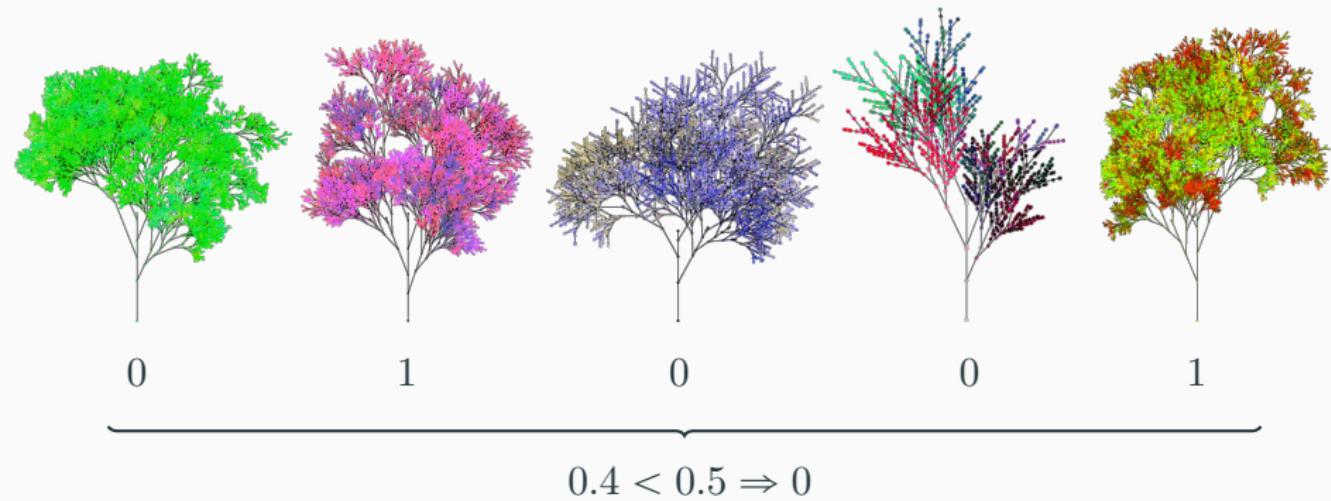
What are random forests?



Classification: **majority vote** over all trees

Random forests

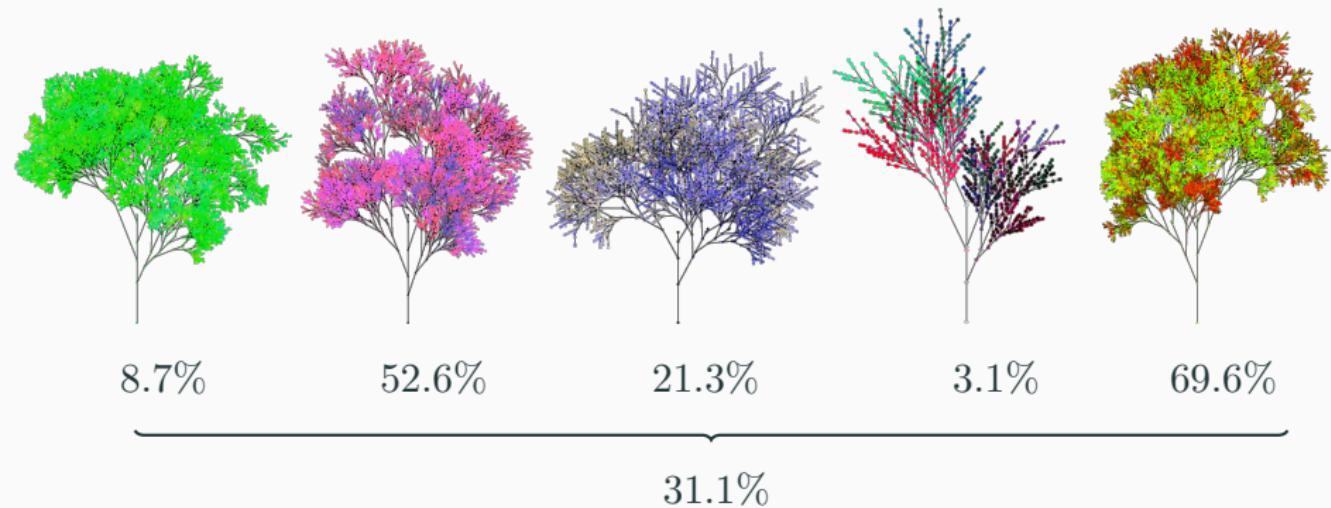
What are random forests?



Classification: **majority vote** over all trees
Identical to average over all trees, cut point 0.5

Random forests

What are random forests?



Probability estimation: Average over all trees

Random forests

What are random forests?

Two components of randomization

- Data manipulation in rows: bootstrapping / subsampling
- Data manipulation in columns: feature subsampling

Random forests

What is bootstrap aggregating (bagging)?

- Ensemble = committee of experts
- Single weak learner = single committee member
- Ensemble decision = committee decision

Fundamental idea of boosting (later) and bagging (bootstrap aggregating)

Any machine can be used as *base learner*, e.g. kNN or tree

What are bootstrapping and subsampling?

- Bootstrapping
 - Sampling **with** replacement
 - Original sample size n , resampled sample size n
 - On average $\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n \approx 0.632 \approx 2/3$ resampled
- Subsampling
 - Sampling **without** replacement
 - Original sample size n , resampled sample size $< n$
 - Standard: resampling of $0.632n$

What is feature subsampling?

At a node consider only subset of features

- Trees vary
- “Experts” differ in their opinion
- Reduce correlation between trees

Number of features considered at split

$mtry = \sqrt{d}$, $\ln d$ or $d/3 \rightarrow$ Tuning possible (later)

Random forests

What is the standard random forest algorithm?

For each tree

1. Draw bootstrap sample with replacement
2. Grow tree
 - a) Use random subset of variables (`mtry`) at each node
 - b) Stop if minimum node size reached
3. Determine proportion of '1' in each terminal node

New subject

1. Drop down subject in each single tree
2. Store proportion from all trees
3. Average proportion of '1's over all trees

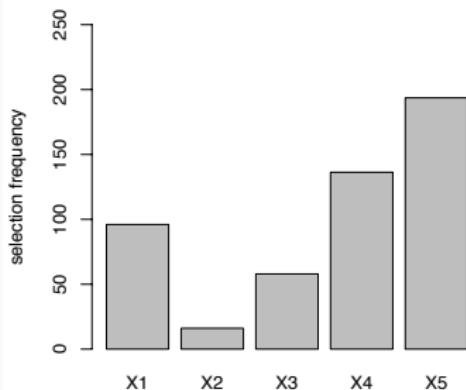
Random forests

Which variables are selected if they have no effect?

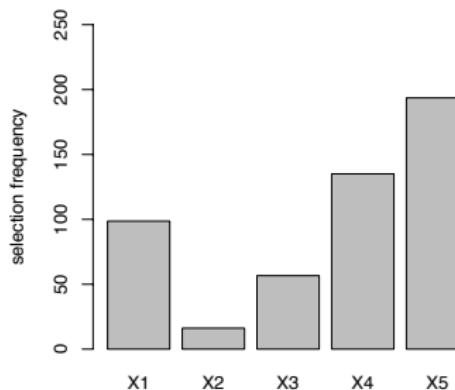
Standard random forest: Bias in split variable selection

- All independent variables unimportant
- $X_1 \sim N(0,1)$, $X_2 \sim M(2)$, $X_3 \sim M(4)$, $X_4 \sim M(10)$,
 $X_5 \sim M(20)$

Bootstrap



Subsample

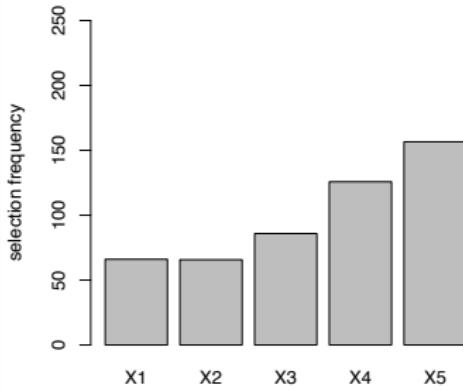


Random forests

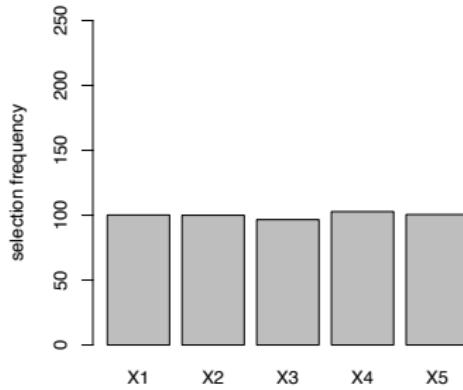
Which variables are selected if they have no effect?

Conditional inference forest with subsampling: NO bias in split variable selection

Bootstrap



Subsample



Random forests

What are conditional inference forests?

Standard random forest

- One-step procedure: maximize gain in purity over all independent variables and all possible split points
- No natural stop criterion

Conditional inference forest (CIF)

- Two-step procedure:
 - Step 1: select variable to split on (association test)
 - Step 2: select optimal split point
- Natural stop through p -value criterion
- Adjustment for multiple testing of split points

What are the statistical properties of trees and RF?

- **Consistency**
 - Single tree not consistent
 - RF consistent if $a_n/n \rightarrow 0$ and $a_n \rightarrow \infty$ (a_n/n : Subsampling rate)
- **Convergence rate**
 - Single trees slower than minimax rate
 - RF achieves minimax rate. If more than 54% of variables have no effect, convergence rate faster than minimax
- **Asymptotic normality**
 - Single tree predictions asymptotically normally distributed
 - RF predictions asymptotically normally distributed for subsampling

Few results for sampling with replacement

Random forests

What are the advantages of random forests?

- As with trees: Procedure intuitive, intrinsic variable selection, simple handling of outliers, fast training
- Work well with high dimensional data
- Work well without (or with only a little) tuning
- Usually better prediction performance than a single tree

Random forests

What are the disadvantages of random forests?

- Not simple to interpret
- Usually worse prediction performance than well tuned boosted trees

Part 5

Model evaluation and resampling

Model evaluation and resampling

What is model evaluation?

How good is a prediction model?

Compare true target y with predicted target \hat{y}

Examples

- How many patients correctly diagnosed?
- How many emails correctly detected as ham or spam?
- How close is the predicted price of a house to the true value?
- How close is the length of hospitalization to the true value?

Model evaluation and resampling

What are standard performance measures?

Dichotomous (binary) outcome

- Proportion of correct classifications (PC):

$$\widehat{PC} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{y_i = \hat{y}_i}$$

- ROC, AUC: $\hat{\mathbb{P}}(y = 1 \mid x)$

- Brier score (BS), i.e., MSE of probability estimates; also

probability score (PS): $\widehat{BS} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{\mathbb{P}}(y_i = 1 \mid x_i))^2$

- Root mean square error (RMSE): $\widehat{RMSE} = \sqrt{\widehat{BS}}$

- Brier skill score (BSS), i.e., Brier score normalized to reference BS_{ref} : $\widehat{BSS} = \frac{\widehat{BS}}{BS_{ref}}$

- Mean absolute error (MAE):

$$\widehat{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{\mathbb{P}}(y_i = 1 \mid x_i)|$$

Model evaluation and resampling

What are standard performance measures?

Multicategory outcome

- Proportion of correct classifications (PC)
- Averaged class-wise PC
- ROC, AUC: several extensions

Continuous outcome

- MSE: $\widehat{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
- MAE: $\widehat{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$
- RMSE: $\widehat{RMSE} = \sqrt{\widehat{MSE}}$
- Explained variance: $\hat{R}^2 = \frac{1 - \widehat{MSE}}{\widehat{Var}(y)}$

What are standard performance measures?

Survival outcome

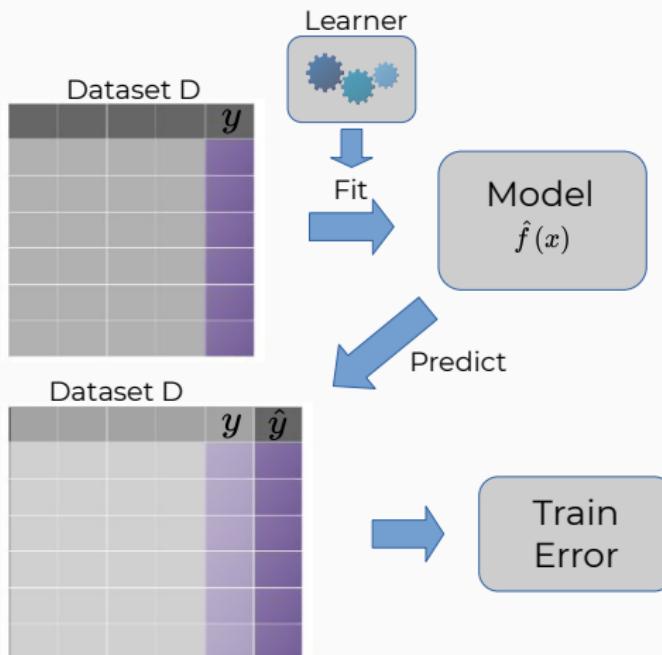
- C-Index
- Time-dependent Brier Score
- Integrated Brier score

Model evaluation and resampling

What is model evaluation?

Training error

Evaluate performance on training data

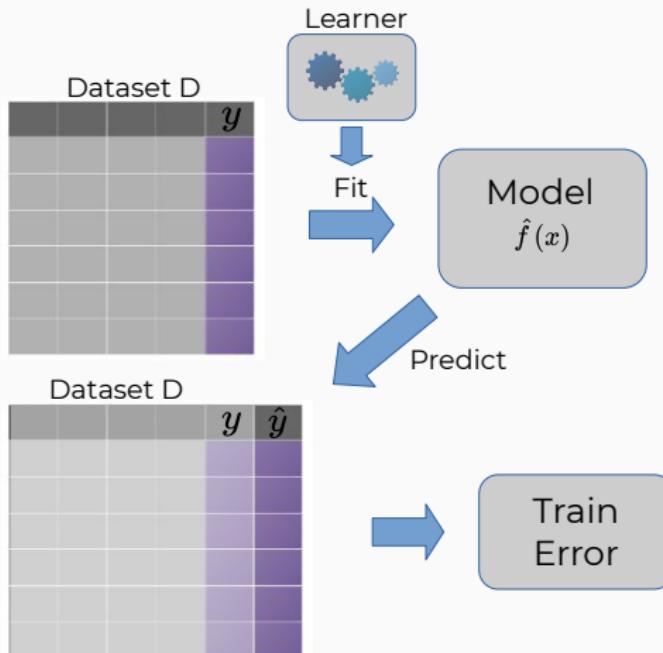


Model evaluation and resampling

What is model evaluation?

Training error

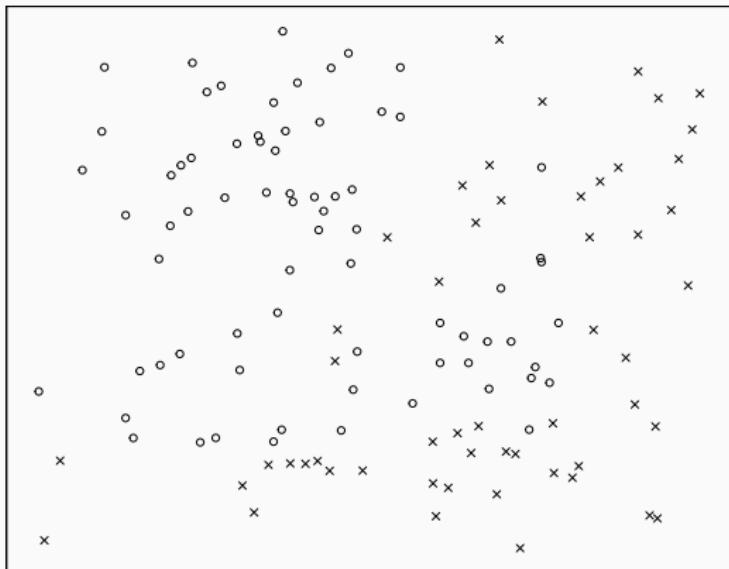
Evaluate performance on training data



Problem:
Overfitting

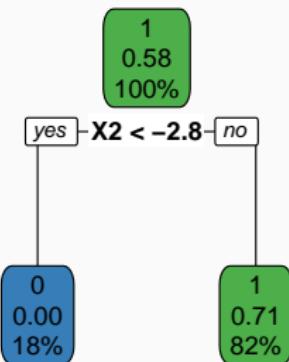
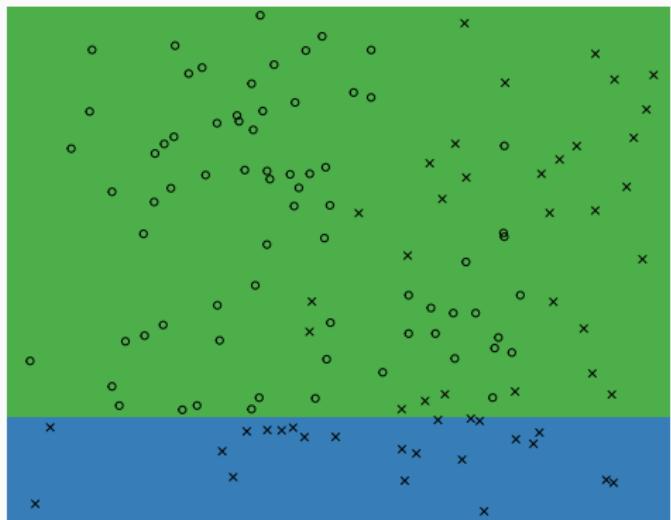
Model evaluation and resampling

What is overfitting?



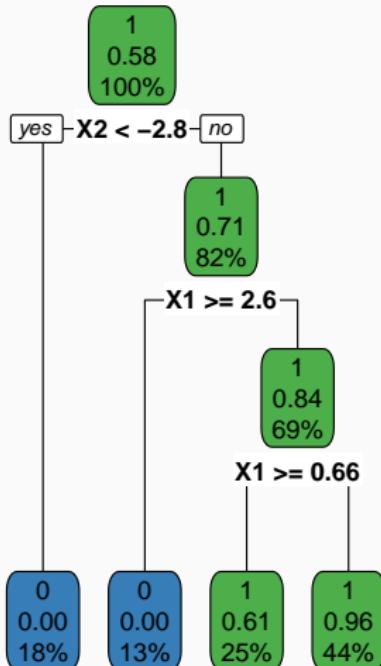
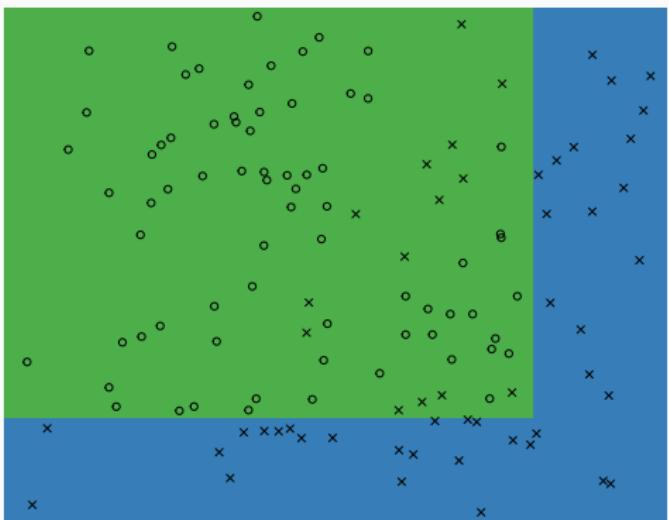
Model evaluation and resampling

What is overfitting?



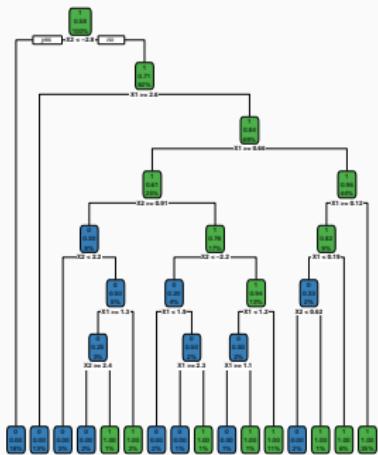
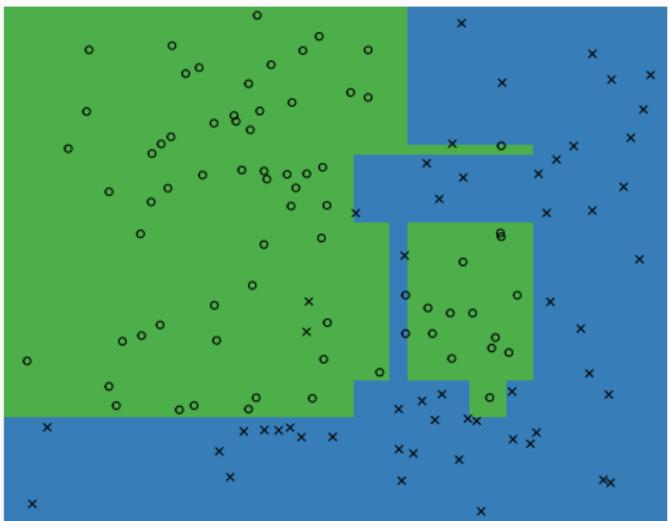
Model evaluation and resampling

What is overfitting?



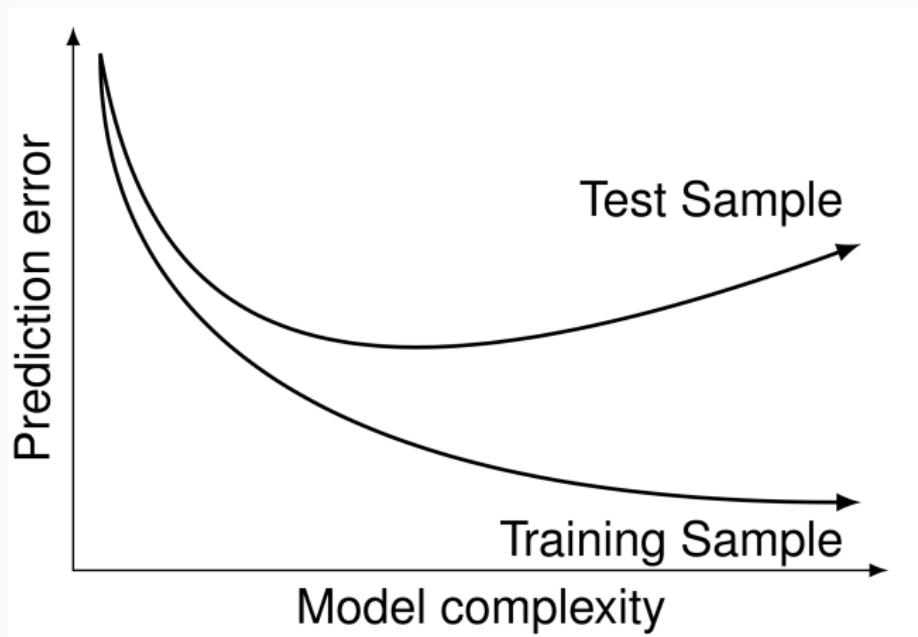
Model evaluation and resampling

What is overfitting?



Model evaluation and resampling

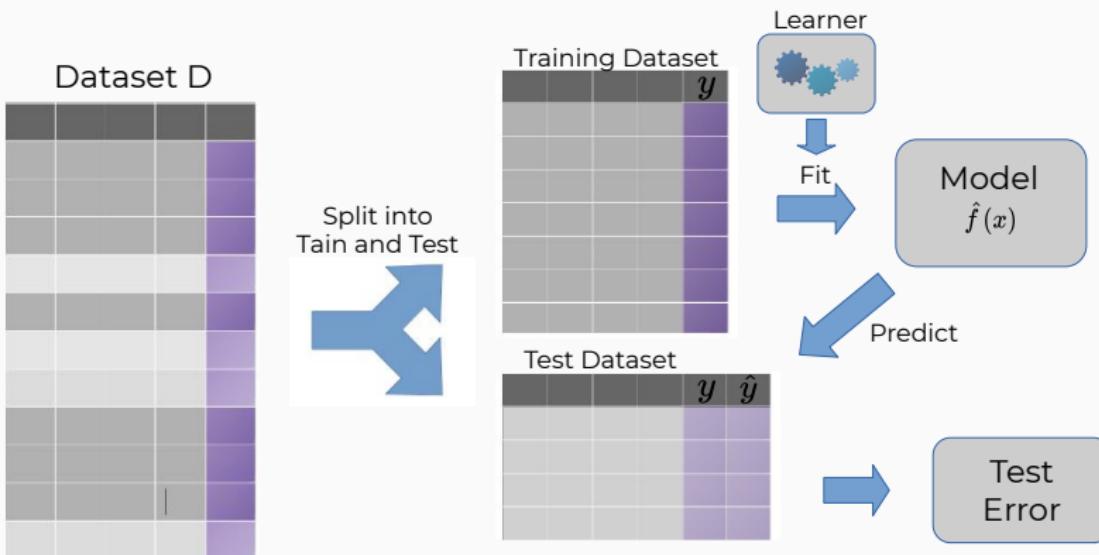
What is overfitting?



Model evaluation and resampling

What is model evaluation?

Test error



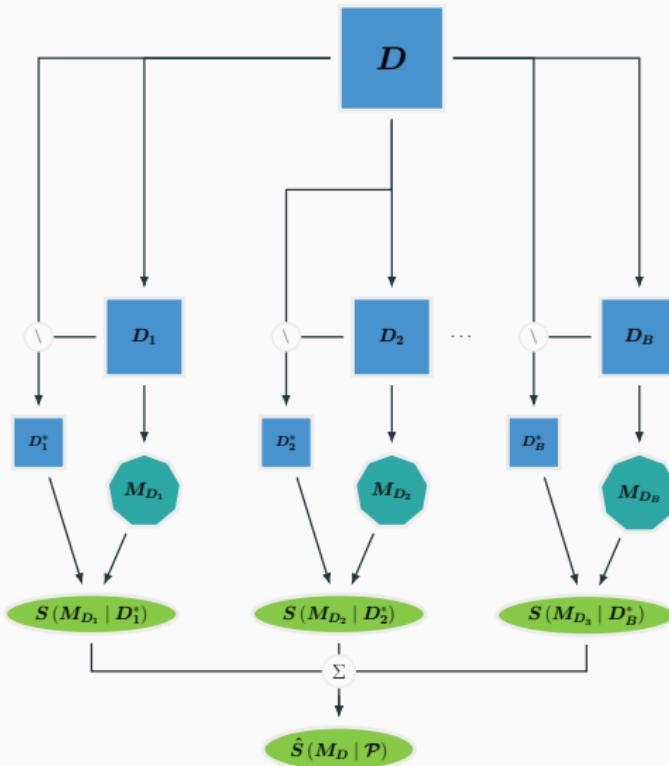
What is model evaluation?

Training and test error

- Training error heavily biased
- Test error (almost) unbiased but variance unknown

Model evaluation and resampling

What is resampling?



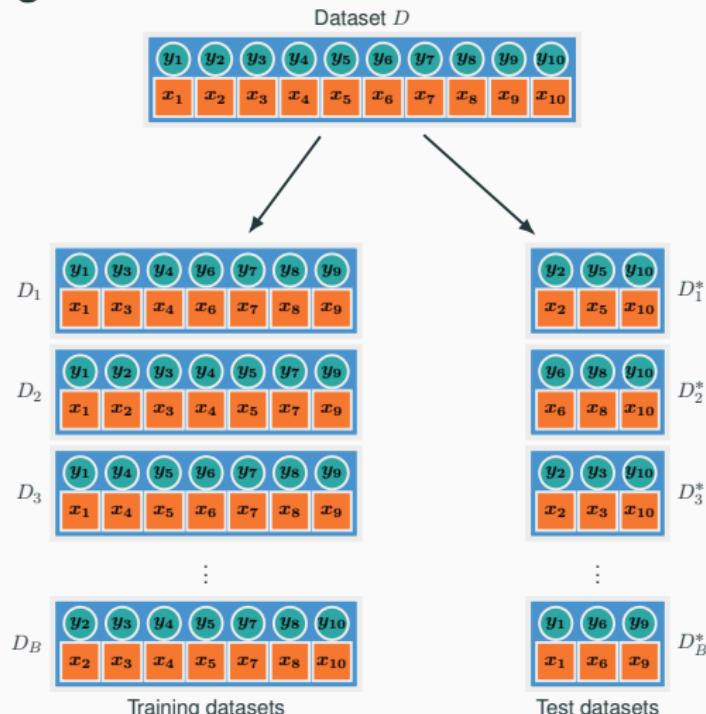
What is resampling?

- Estimate performance on independent data
- Used for
 - Performance estimation
 - Hyperparameter tuning
 - Model selection
- Resampling based performance estimation
 1. Split dataset in several (smaller) datasets D_b
 2. On each dataset D_b :
 - 2.1 Train machine
 - 2.2 Estimate performance on $D_b^* = D \setminus D_b$
 3. Aggregate performance estimates

Model evaluation and resampling

What is resampling?

Subsampling



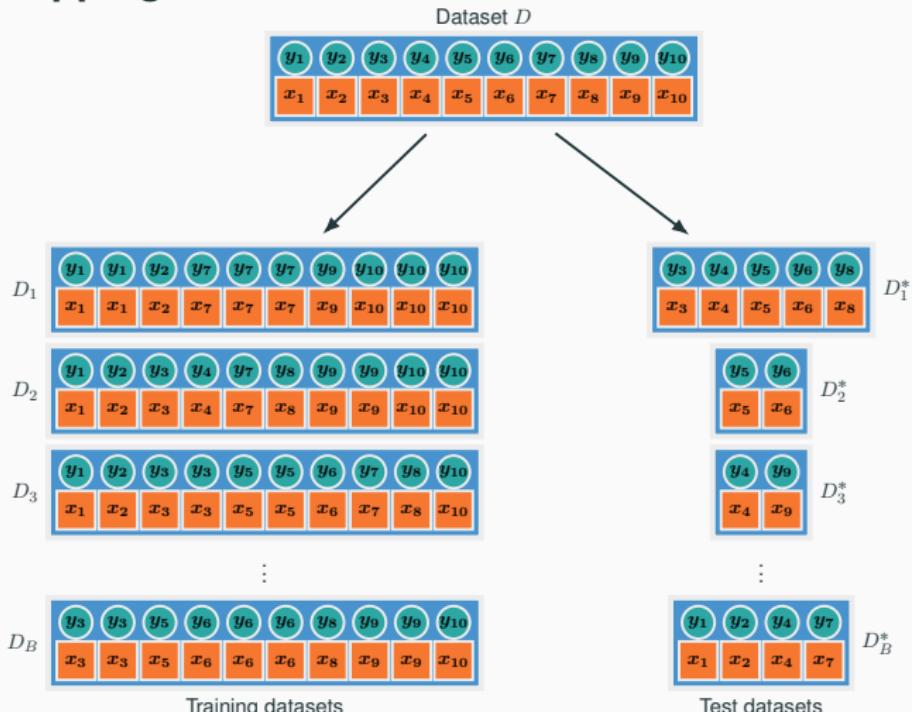
What is resampling?

Subsampling

- Sample B training datasets D_b from D without replacement, usually $n_b = \frac{2}{3}n$
- Use $D_b^* = D \setminus D_b$ as test datasets
- D_b and D_b^* disjunct
- D_1 and D_2 not disjunct
- D_1^* and D_2^* not disjunct
- Performance estimator biased
- No optimal B , usually $100 < B < 1000$
- Special case with $B = 1$: Single train/test split (holdout)

Model evaluation and resampling

What is resampling? Bootstrapping



What is resampling?

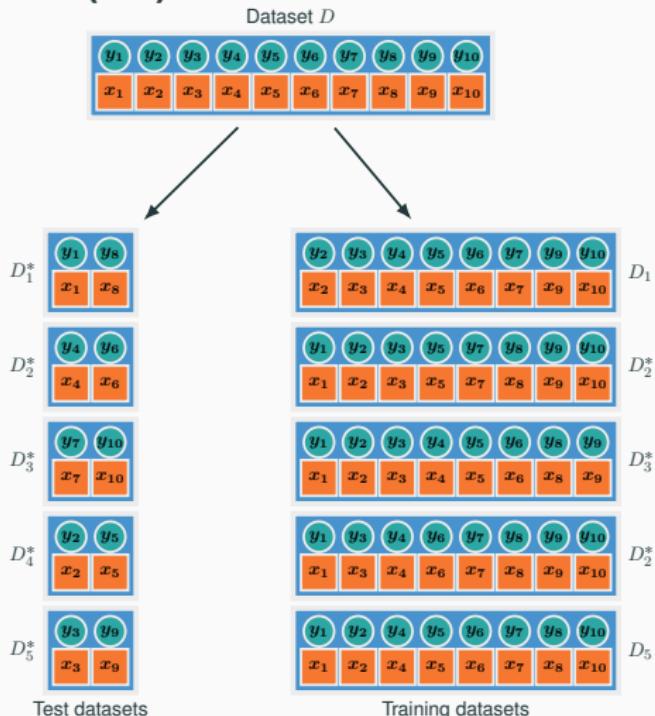
Bootstrapping

- Sample B training datasets D_b from D with replacement, usually $n_b = n$
- Use $D_b^* = D \setminus D_b$ as test datasets
- D_b and D_b^* disjunct
- D_1 and D_2 not disjunct
- D_1^* and D_2^* not disjunct
- Performance estimator biased
- Adaptive weighting of performance estimators to reduce bias (.632+ bootstrap)
- Small variance (large B)
- No optimal B , usually $100 < B < 1000$

Model evaluation and resampling

What is resampling?

Cross validation (CV)



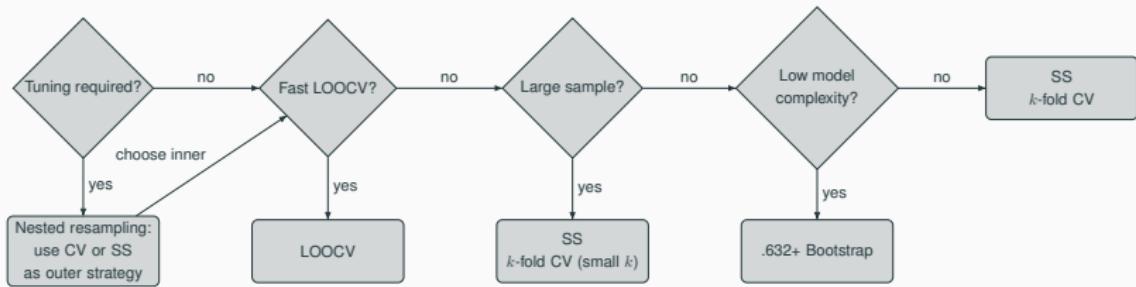
What is resampling?

Cross validation (CV)

- Split D in B test datasets D_b^*
- Use $D_b = D \setminus D_b^*$ as training datasets
- D_b and D_b^* disjunct
- D_1 and D_2 not disjunct
- D_1^* and D_2^* disjunct
- Special case with $B = n$: Leave-one-out CV (LOOCV)
 - Small bias, high variance
 - Long runtime
- No optimal B , usually $B = 5, 10$
 - Slightly more bias than LOOCV, but lower variance
 - Lowest B of all resampling methods → fast computation

Model evaluation and resampling

How to choose the resampling method?



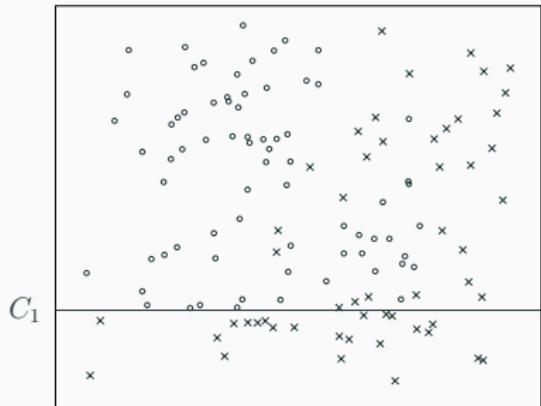
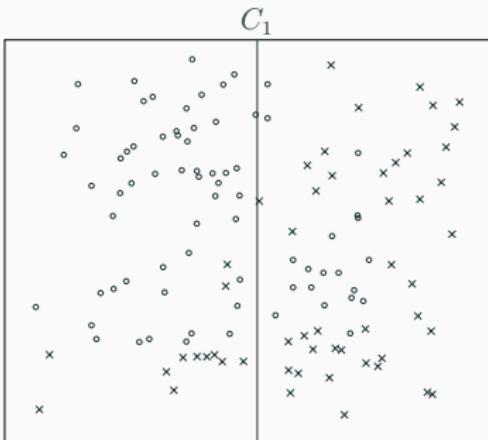
Part 6

Boosting

Boosting stumps

What is boosting stumps?

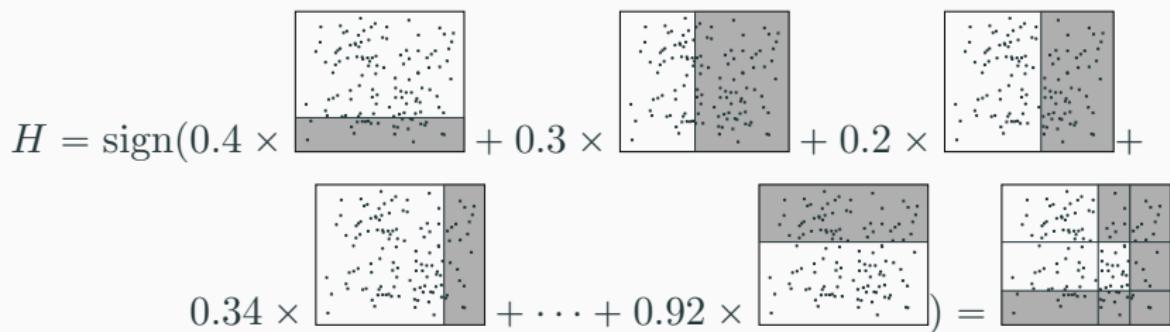
- Use 1-level decision tree, i.e., a stump
Geometrically: only one horizontal or one vertical split



Boosting stumps

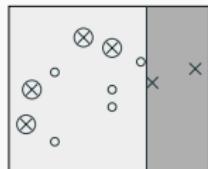
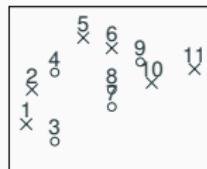
What is boosting stumps?

- Stump might not be able to classify all data points correctly
- If stumps are done repeatedly, classification might be improved
- Stumps often used as base learner in boosting



AdaBoost

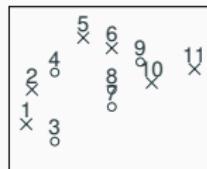
What is discrete AdaBoost?



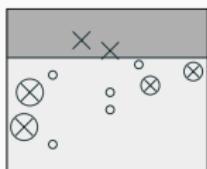
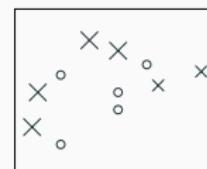
i	1	2	3	4	5	6	7	8	9	10	11	sum	err	alpha
$w_{i,1}$	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	1	0.36	0.5596
$w_{i,2}$	0.159	0.159	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.09	0.09			

AdaBoost

What is discrete AdaBoost?



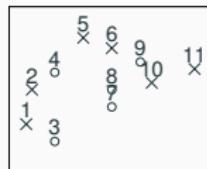
i	1	2	3	4	5	6	7	8	9	10	11	sum	err	alpha
$w_{i,1}$	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	1	0.36	0.5596
$w_{i,2}$	0.159	0.159	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.09	0.09			



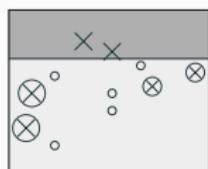
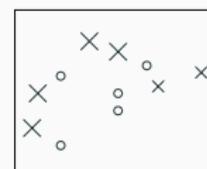
i	1	2	3	4	5	6	7	8	9	10	11	sum	err	alpha
$w_{i,2}$	0.159	0.159	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.09	0.09	1.27	0.3928	0.4356
$w_{i,3}$	0.2458	0.2458	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.09	0.1405	0.1405		

AdaBoost

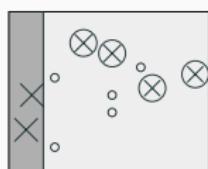
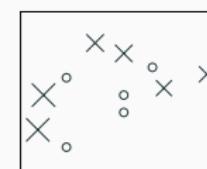
What is discrete AdaBoost?



i	1	2	3	4	5	6	7	8	9	10	11	sum	err	alpha
$w_{i,1}$	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	1	0.36	0.5596
$w_{i,2}$	0.159	0.159	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.09	0.09			



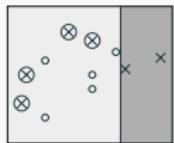
i	1	2	3	4	5	6	7	8	9	10	11	sum	err	alpha
$w_{i,2}$	0.159	0.159	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.09	0.09	1.27	0.3928	0.4356
$w_{i,3}$	0.2458	0.2458	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.1405	0.1405			



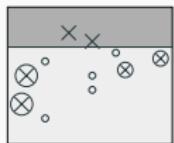
i	1	2	3	4	5	6	7	8	9	10	11	sum	err	alpha
$w_{i,3}$	0.2458	0.2458	0.09	0.09	0.159	0.159	0.09	0.09	0.09	0.1405	0.1405	1.5451	0.3877	0.457
$w_{i,4}$	0.2458	0.2458	0.09	0.09	0.2511	0.2511	0.09	0.09	0.09	0.2219	0.2219			

AdaBoost

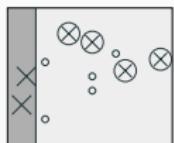
What is discrete AdaBoost?



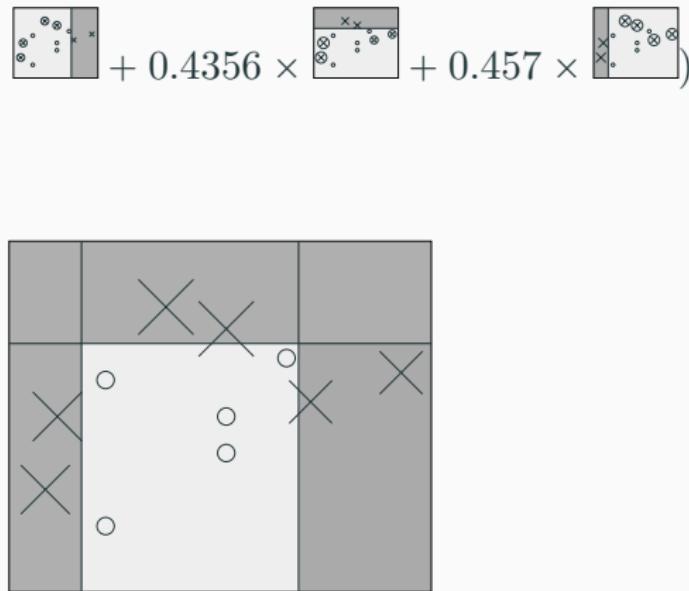
$$\frac{\text{alpha}}{0.5596} H = (0.5596 \times \text{grid 1}) + 0.4356 \times \text{grid 2} + 0.457 \times \text{grid 3})$$



$$\frac{\text{alpha}}{0.4356}$$



$$\frac{\text{alpha}}{0.457}$$



What is discrete AdaBoost?

1. Start with equal weights w_i for all subjects
2. For all boosting steps $b = 1$ to B do
 - a) Fit a classifier $G_b(x)$ to the training data using the above weights
 - b) Compute error weights $err_b = \frac{1}{\sum_i w_i} \sum_{i:G_b(x_i) \neq y_i} w_i$
 - c) Compute log odds of non error $\alpha_b = \ln(\frac{1-err_b}{err_b})$
 - d) For all misclassified subjects increase weights using α_b
3. Output: weighted mean of classifiers

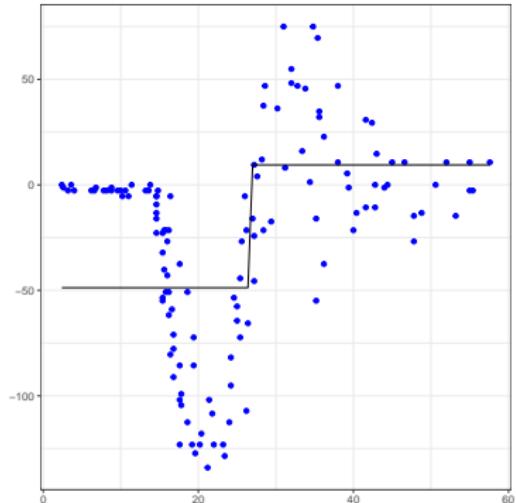
$$G(x) = \text{sign} \left(\sum_{b=1}^B \alpha_b G_b(x) \right)$$

Gradient boosting

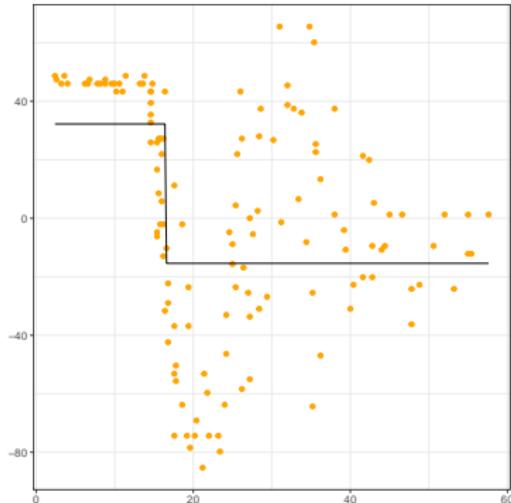
What is gradient boosting?

- Learn a dependent variable **using any learner**
- Compute residual (more general: any loss function)
- Learn the residual

Learn a simple model - here: a stump



Try to correct its error

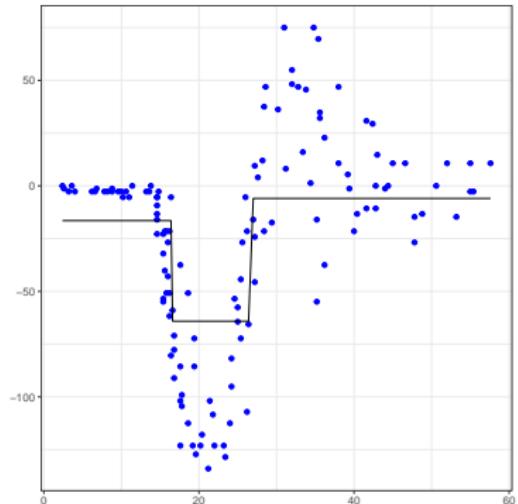


Gradient boosting

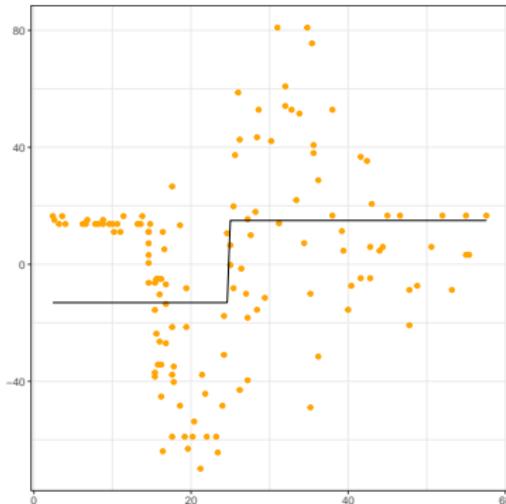
What is gradient boosting?

- Learn a dependent variable **using any learner**
- Compute residual (more general: any loss function)
- Learn the residual

Combining improves prediction



Try to correct its error once again

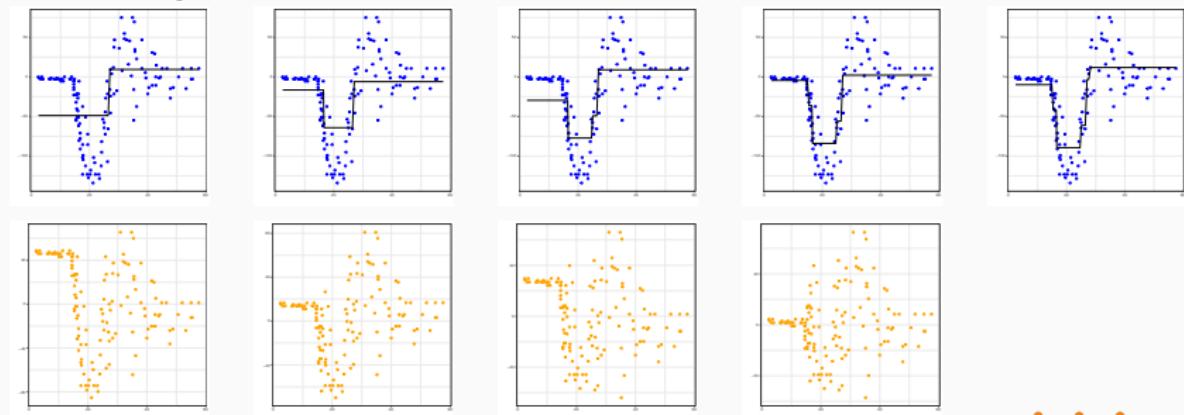


Gradient boosting

What is gradient boosting?

- Learn a dependent variable **using any learner**
- Compute residual (more general: any loss function)
- Learn the residual

Data and **prediction** function and **residual**



What are similarities and differences of AdaBoost and gradient boosting?

Identical fundamental idea

Boost performance of simple base-learner by iteratively shifting the focus towards problematic observations that are 'difficult' to predict

Difference

- AdaBoost: up-weight observations misclassified before
- Gradient boosting: difficult observations have large residuals

Gradient boosting

What are extensions of gradient boosting?

Component-wise boosting

Fit an additive model, e.g. regression

- Base learner: regression of single independent variable
- Select base learner with best fit among all single components, add to model

Likelihood boosting

Maximize likelihood with penalty term

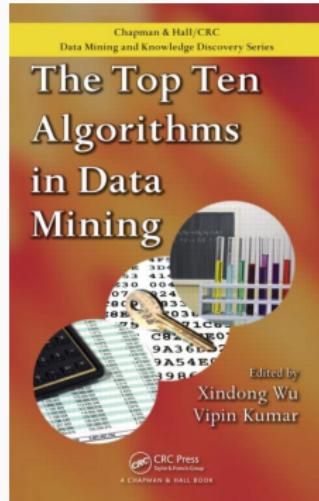
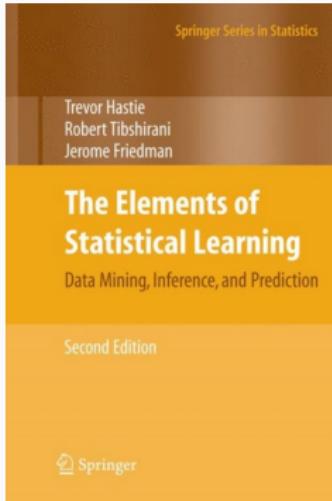
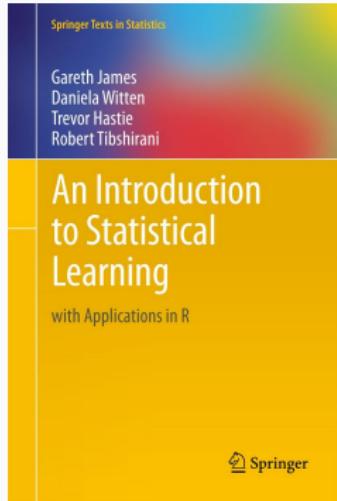
What are the advantages of boosting?

- High prediction accuracy
- Fast training
- Optimize any differentiable loss function
- Intrinsic variable selection

What are the disadvantages of boosting?

- Not simple to interpret (except component-wise boosting)
- Not easily parallelized (but possible)
- Prone to overfitting
- Requires proper parameter tuning

Literature



<https://www-bcf.usc.edu/~gareth/ISL/> (free!)

<https://web.stanford.edu/~hastie/ElemStatLearn/> (free!)

<http://www.crcpress.com/product/isbn/9781420089646>

Outline (Day 2)

Morning

7. Support vector machines
8. Hyperparameter tuning and performance comparison
9. Artificial neural networks

Afternoon

10. Specific endpoints
11. Variable importance and selection
12. Discussion

Part 7

Support vector machines

What are support vector machines?

Fundamental idea

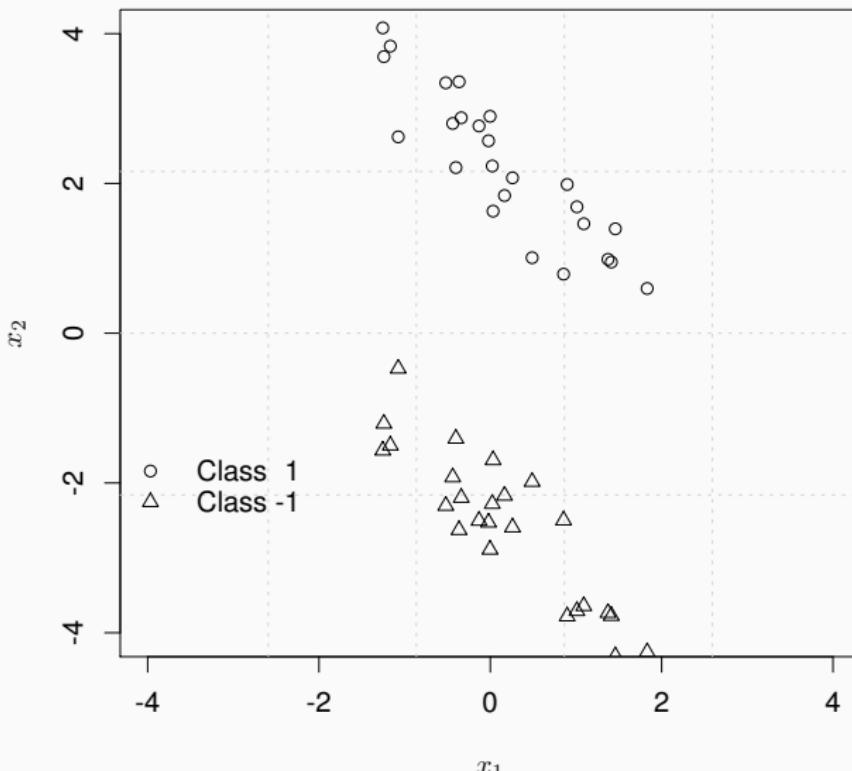
- Find the hyperplane which best separates the data
- SVM similar to discriminant analysis, but SVM works without likelihood

Warning

Mathematically most complex, not possible without formulae!

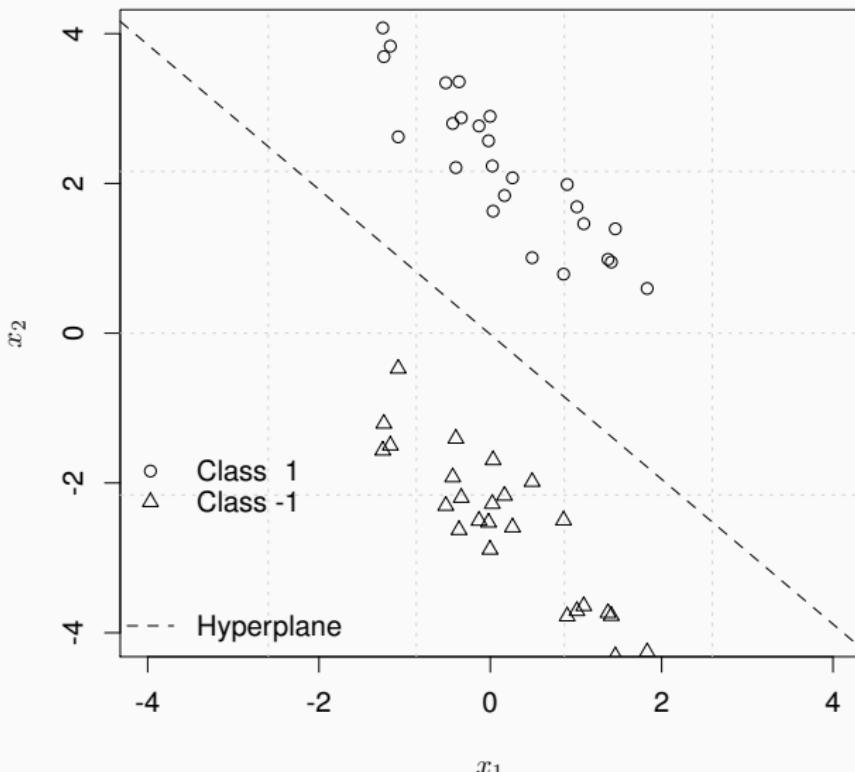
Support vector machines

What are support vector machines?



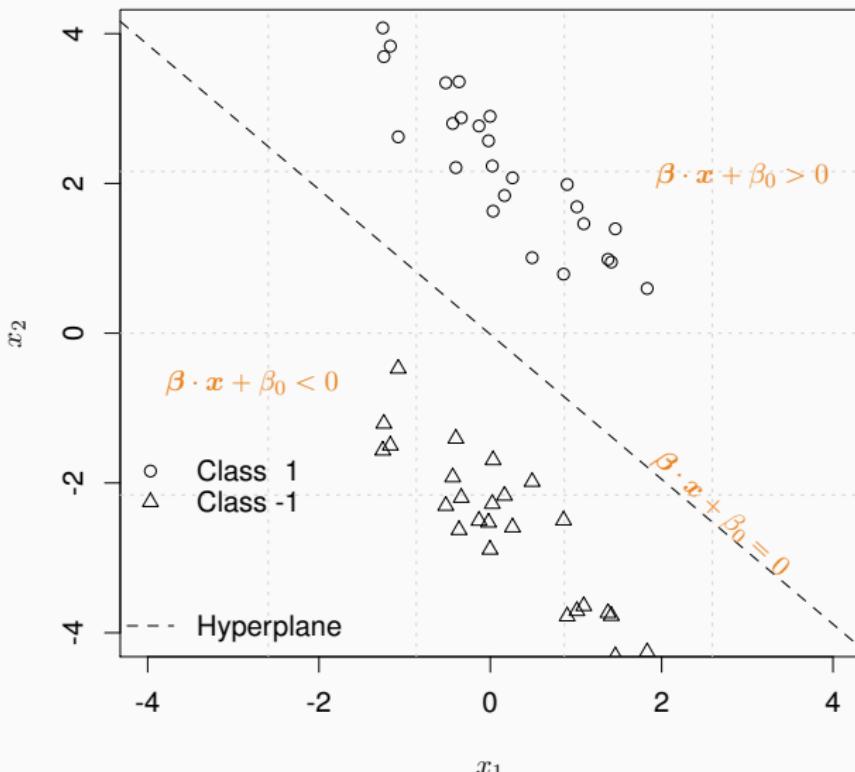
Support vector machines

What are support vector machines?



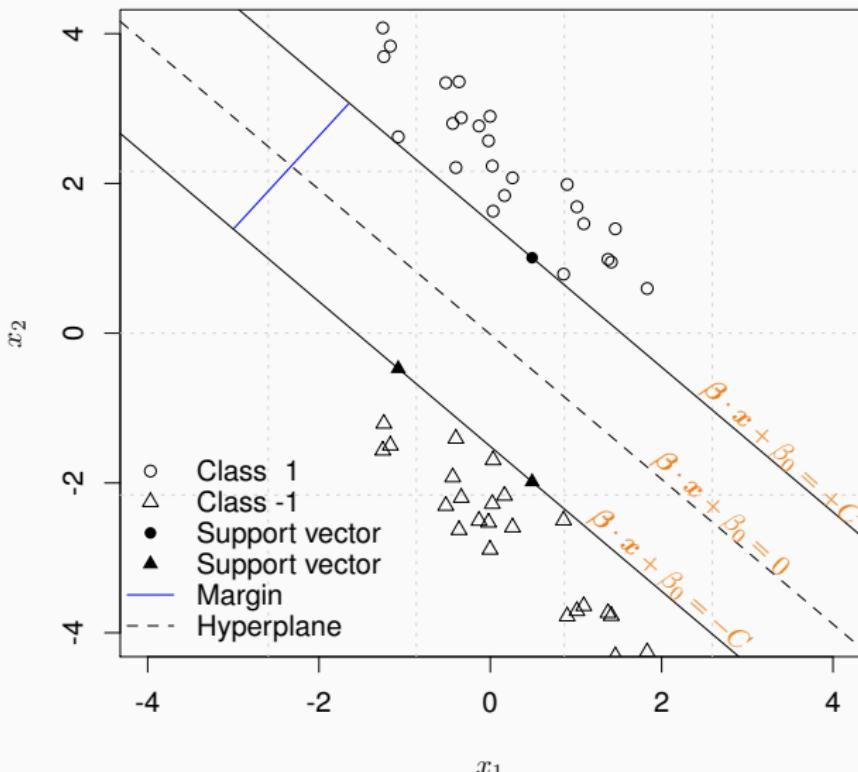
Support vector machines

What are support vector machines?



Support vector machines

What are support vector machines?



Support vector machines

What are support vector machines?

- Hyperplane given by $f(\mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{x} + \beta_0$
- Separation given by $y_i f(\mathbf{x}_i) > 0$ for $i = 1, \dots, n$
- Number of Hyperplanes: **infinite**
- Uniqueness guaranteed by maximizing the **margin C** between groups
- Width of margin : $2C$
- Maximization problem:

$$\max_{\boldsymbol{\beta}, \beta_0, \|\boldsymbol{\beta}\|=1} C$$

subject to $y_i(\boldsymbol{\beta} \cdot \mathbf{x}_i + \beta_0) \geq C, \quad i = 1, \dots, n$

Support vector machines

What are support vector machines?

- Maximization problem:

$$\max_{\beta, \beta_0, \|\beta\|=1} C$$

subject to $y_i(\beta \cdot x_i + \beta_0) \geq C, \quad i = 1, \dots, n$
is equivalent to

$$\min_{\beta, \beta_0} \|\beta\|$$

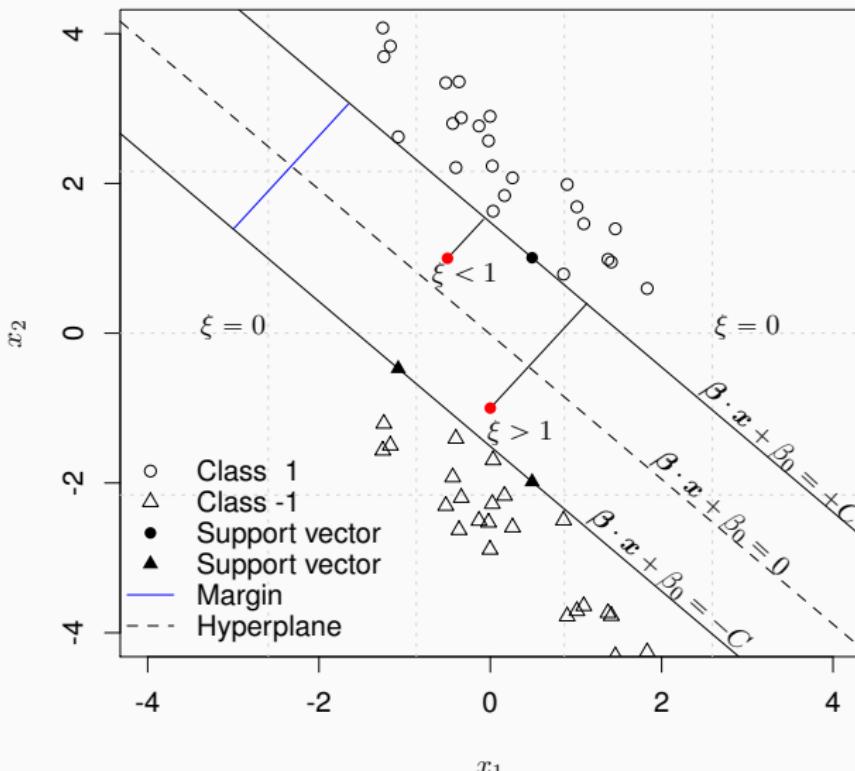
subject to $y_i (\beta \cdot x_i + \beta_0) \geq C, \quad i = 1, \dots, n$

- Classification rule for an observation: x^*

$$\text{sign}(f(x^*)) = \text{sign}(\beta \cdot x^* + \beta_0)$$

Support vector machines

What are support vector machines?



Support vector machines

What are support vector machines?

- Slack variables ξ_i
- Subject on correct side of margin $\Rightarrow \xi_i = 0$
- Subject on correct side, but within margin $\Rightarrow 0 < \xi_i < 1$
- Subject on wrong side of the margin $\Rightarrow \xi_i \geq 1$

Restriction: $\sum_{i=1}^n \xi_i \leq \gamma$

- Tuning Parameter γ
- Proportion of predictions on the wrong side of the margin

Support vector machines

What are support vector machines?

Maximization problem:

$$\max_{\beta, \beta_0, \|\beta\|=1} C$$

subject to $y_i (\beta \cdot x_i + \beta_0) \geq 1 - \xi_i$

and

$$\xi_i \geq 0$$

$$\sum_{i=1}^n \xi_i \leq \gamma \quad i = 1, \dots, n$$

Support vector machines

What are support vector machines (SVMs)?

Idea for solving optimization problem

- Quadratic problem with linear inequalities
→ convex optimization problem
- Simplify optimization problem by transfer to the dual
- Together with Karush-Kuhn-Tucker conditions
→ unique solution

Advantages of the dual form

- Does not involve β
- Involves maximization problem with inner product
 $x_i \cdot x_j = \langle x_i, x_j \rangle$
- Quadratic programming solver with Lagrange multipliers

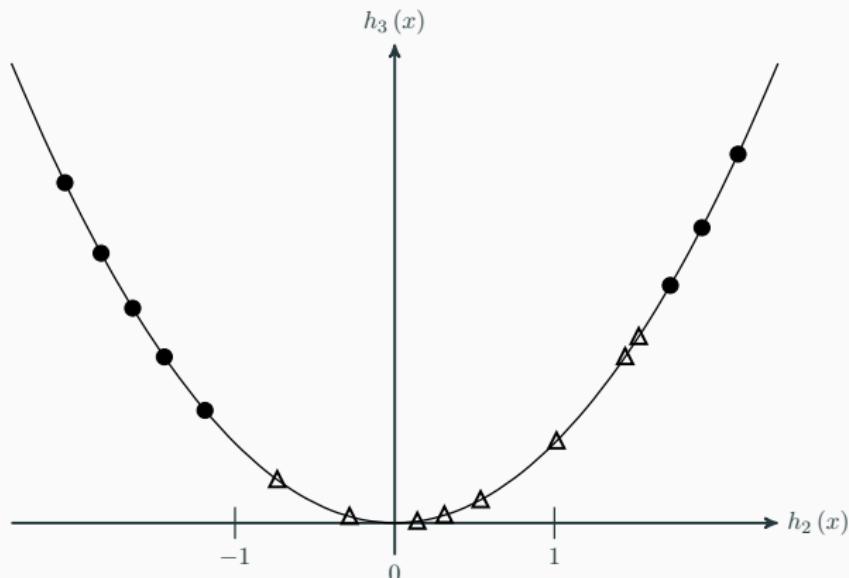
Support vector machines

What is the kernel trick?



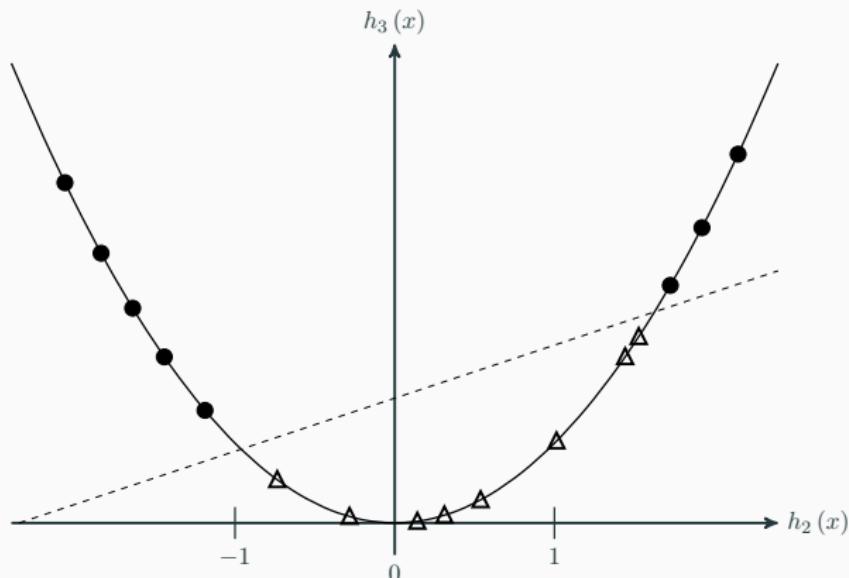
Support vector machines

What is the kernel trick?



Support vector machines

What is the kernel trick?



What is the kernel trick?

- Separation by hyperplane impossible
- BUT separation by hyperplane might be possible in higher dimensional space → feature space $h(x)$
- Separating hyperplane given by

$$\beta \cdot h(x) + \beta_0$$

Disadvantage

- Drastic increase in CPU time

Kernel trick

- Reduction of CPU time by using specific kernel functions

Support vector machines

What is the kernel trick?

Solution of the optimization problem f for new observation \boldsymbol{x}^*

$$\begin{aligned}\text{sign}(f(\boldsymbol{x}^*)) &= \text{sign}(\boldsymbol{\beta} \cdot \boldsymbol{x}^* + \beta_0) \\ &= \text{sign}\left(\sum_{i=1}^n \alpha_i y_i \boldsymbol{x}_i \cdot \boldsymbol{x}^* + \beta_0\right) \\ &= \text{sign}\left(\sum_{i=1}^n \alpha_i y_i \langle \boldsymbol{x}^*, \boldsymbol{x}_i \rangle + \beta_0\right)\end{aligned}$$

With feature space as input space

$$\text{sign}(f(\boldsymbol{x}^*)) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i \langle \boldsymbol{h}(\boldsymbol{x}^*), \boldsymbol{h}(\boldsymbol{x}_i) \rangle + \beta_0\right)$$

What is the kernel trick?

Trick: choose kernel function to evaluate

$$K(\mathbf{x}^*, \mathbf{x}_i) = \langle \mathbf{h}(\mathbf{x}^*), \mathbf{h}(\mathbf{x}_i) \rangle$$

without explicit calculation of $h(\cdot)$

What is the kernel trick?

Standard properties of kernel functions $K(\cdot, \cdot)$

- Symmetric
- Positive definite

Support vector machines

What is the kernel trick?

Classical choices of kernel functions

- Inhomogeneous polynomial kernel
$$K(\mathbf{x}^*, \mathbf{x}_i) = (c + \langle \mathbf{x}^*, \mathbf{x}_i \rangle)^b$$
 with $b \in \mathbb{N}$ and $c \geq 0$
- Sigmoid kernel (artificial neural network kernel)
$$K(\mathbf{x}^*, \mathbf{x}_i) = \tanh(\kappa \langle \mathbf{x}^*, \mathbf{x}_i \rangle + v)$$
 with $\kappa > 0$ and $v < 0$
- Gaussian radial kernel
$$K(\mathbf{x}^*, \mathbf{x}_i) = \exp\left(\frac{1}{2\sigma^2} \|\mathbf{x}^* - \mathbf{x}_i\|\right)$$
, with $\sigma^2 > 0$
- Bessel kernel
$$K(\mathbf{x}^*, \mathbf{x}_i) = -\text{Bessel}_{\nu+1}^n\left(\sigma \|\mathbf{x}^* - \mathbf{x}_i\|^2\right)$$

How can SVMs be used for probability estimation?

Approach 1

Fit parametric sigmoid link function between conditional probability and SVM classification function

⇒ Imposes parametric assumptions, probably unrealistic for applications

How can SVMs be used for probability estimation?

Approach 2

- SVM consistent estimation of Bayes rule
- Weighted SVM allows consistent estimation of probabilities at other thresholds

Concept

- Fit weighted SVM classifiers
- SVM estimates correspond to point probabilities
- **Bracket probability:** observation exactly between two SVM estimates
- Average probability of lower and upper limits

What are the statistical properties of SVMs?

- Summary of findings: Moguerra & Muñoz
- Consistency for binary SVM: Cucker & Smale
- Consistency of multiclassification SVM:
Doğan et al.
- Convergence rate:
 - Surrogate loss: Zhang
 - Hinge loss: Bartlett et al.
 - Gaussian kernel binary SVM: Steinwart & Scovel
 - Multiclass: Zhang, Wang & Shen, Zhang & Liu
 - Probability SVM: Wang et al.
 - Multiclass probability SVM: Not aware of any result

What are the advantages of SVMs?

- Work well with high dimensional data
- Work well on non-linear data (kernel trick)
- Simple handling of outliers
- Beautiful theory

What are the disadvantages of SVMs?

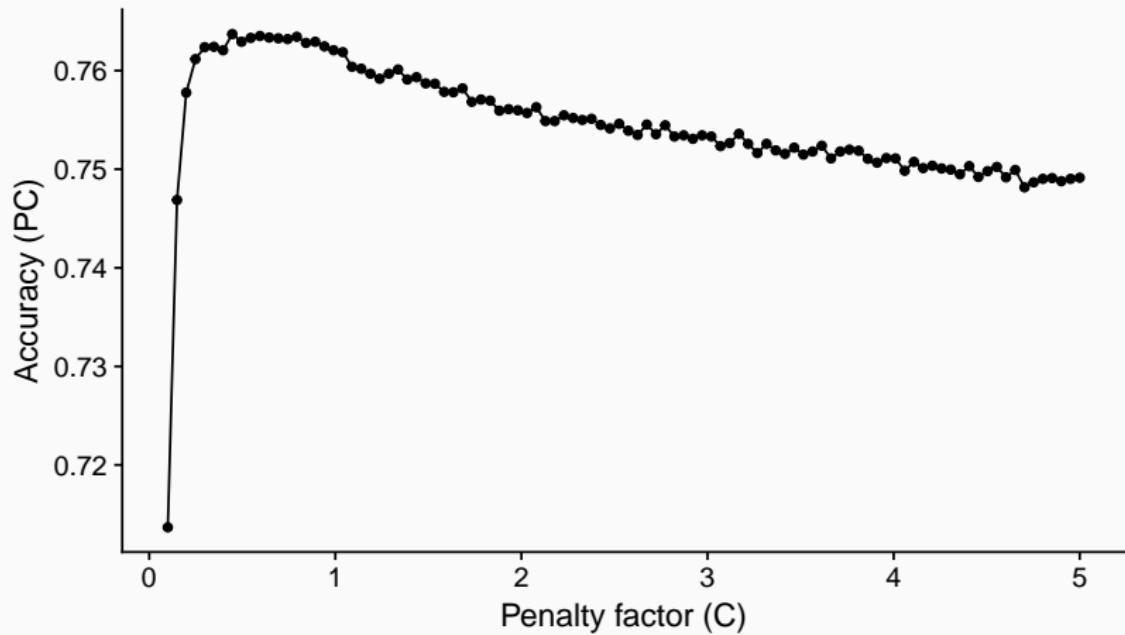
- Not simple to interpret
- Difficult to guess good kernel on real data
- Require proper hyperparameter tuning
- Computationally intensive on large data sets
- Optimized for binary classification and numerical features

Part 8

Hyperparameter tuning and performance comparison

Hyperparameter tuning

Why is parameter tuning necessary?



What is parameter tuning?

Hyperparameters

Learners have hyperparameters, e.g.:

- Number of nearest neighbors k
- Depth of a tree
- Number of features to consider in each split of a random forest (mtry)
- Number of boosting iterations
- Kernel of SVM

Most learners have several hyperparameters

Have to be jointly optimized

What is parameter tuning?

Search entire parameter space

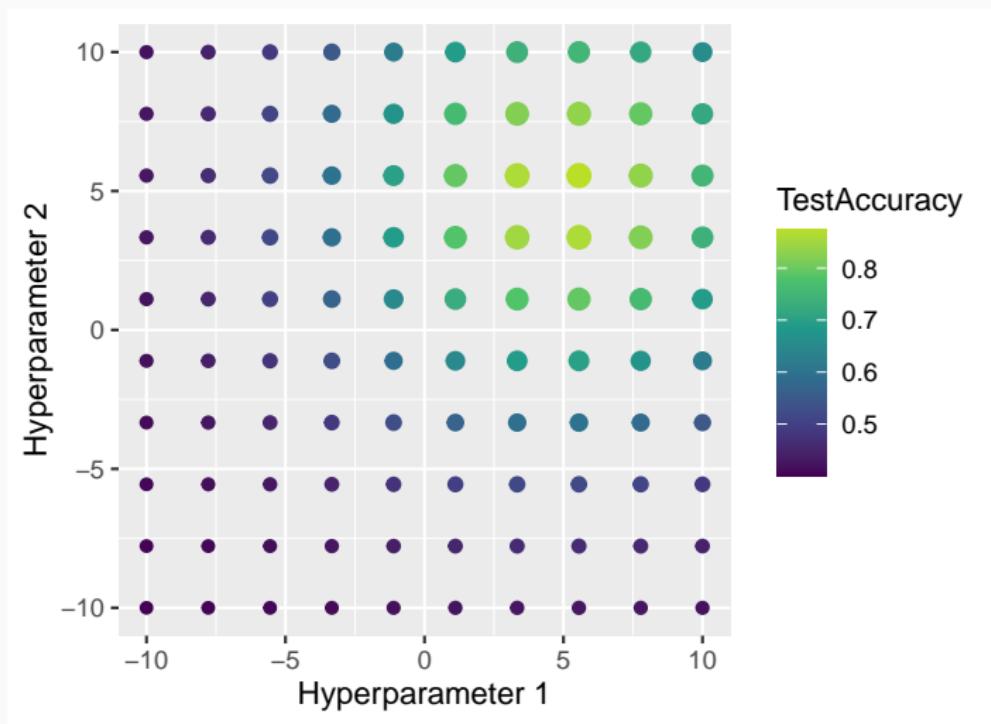
- All possible combinations
- Grid search
- Randomly select combinations
- Model-based optimization

Use resampling

- Evaluate each parameter combination on all resampling iterations/folds
- Choose parameter maximizing aggregated performance measure

Hyperparameter tuning

What is grid search?



Grid search

Advantages

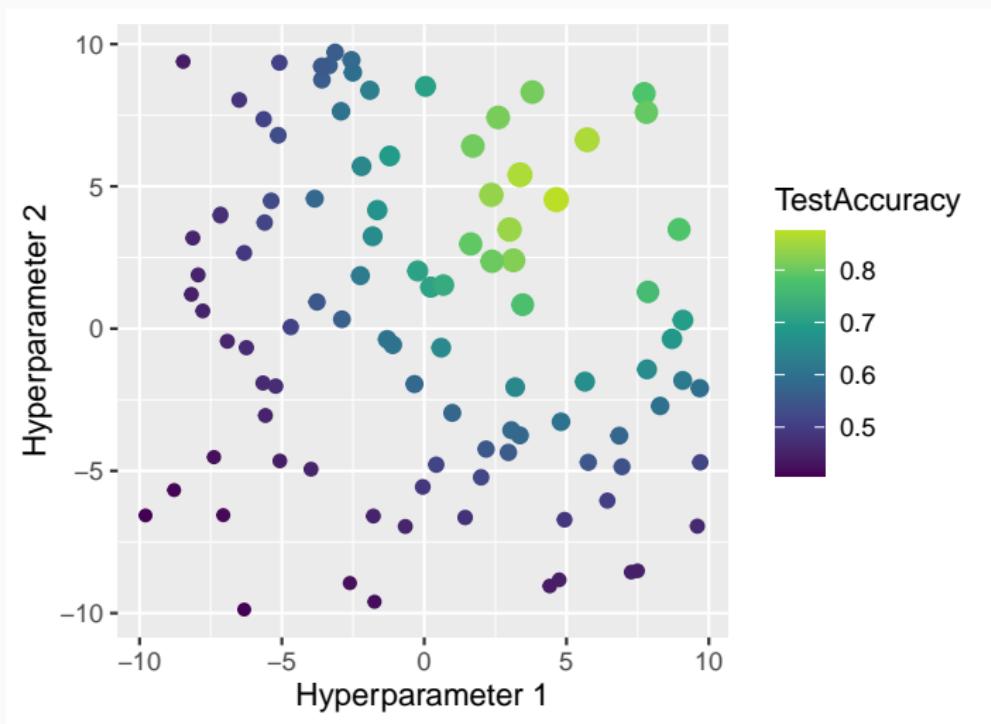
- Easy to implement
- All parameter types possible
- Easily parallelized

Disadvantages

- Computationally intensive
- Inefficient: Searches large irrelevant areas
- Arbitrary: Which values / discretization?

Hyperparameter tuning

What is random search?



Random search

Advantages

- Same as grid search: Easy to implement, all parameter types possible, trivial parallelization
- Easy to adjust to computational budget
- No discretization
- Superior performance compared to grid search

Disadvantages

- Computationally intensive
- Inefficient: Searches large irrelevant areas

What is model-based optimization?

Surrogate model

Learn relationship between hyperparameters and prediction performance

Algorithm

1. Pick initial configuration (e.g. random)
2. Learn surrogate model
3. Predict new configuration with surrogate model
4. Repeat steps 2 and 3

Model-based optimization

Advantages

- All parameter types possible
- Efficient: Focus on promising areas
- Superior performance compared to grid and random search

Disadvantages

- Computationally intensive
- Non-trivial parallelization
- Harder to implement

Performance comparison

How can performance be compared?

Be fair!

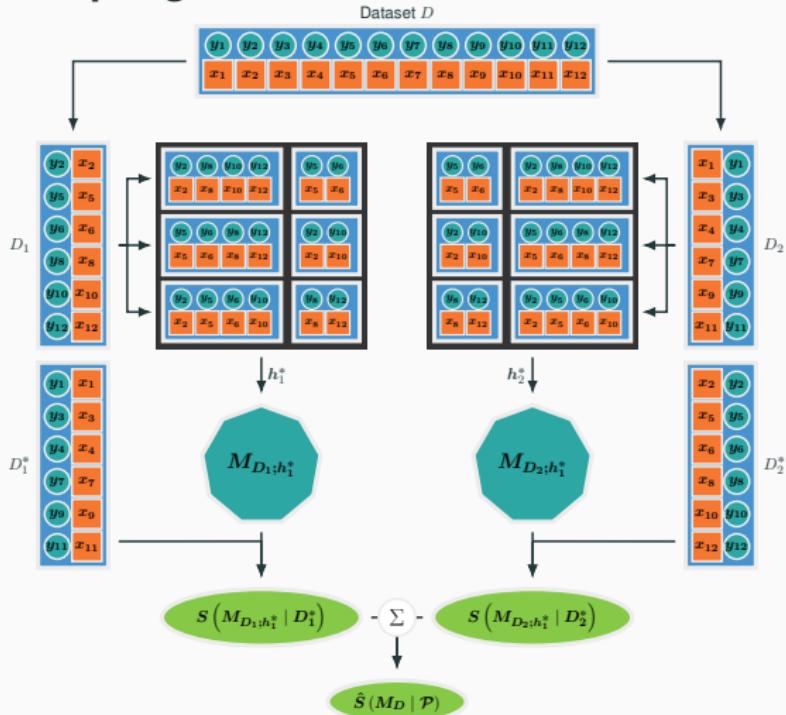
- Compare all learners and models on same data
- Tune parameters of all learners
- Don't overfit
- Don't publish over-optimistic results

Never learn, tune or evaluate on same data!

Performance comparison

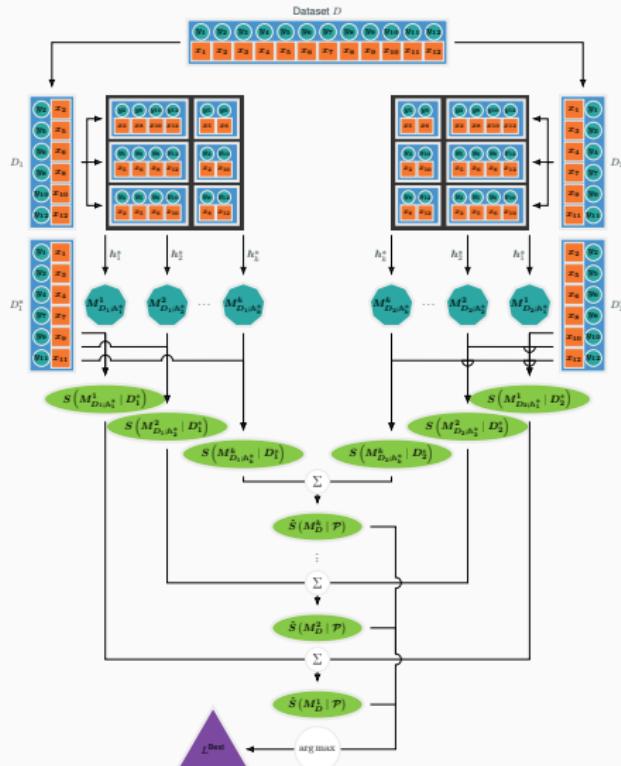
How can performance be compared?

Nested resampling



Performance comparison

How can performance be compared?



How to build a final model?

1. Select best learner with nested resampling
2. Find optimal hyperparameters of best learner with resampling
3. Train best learner with optimal hyperparameters on full data

How can differences in performance be statistically tested?

One dataset

- McNemar test for simple holdout validation
- 5×2 CV test
- Corrected t-test for (repeated) CV

Several datasets

- 2 machines: Wilcoxon signed rank test
- Several machines: Friedman test

How can differences in performance be statistically tested?

Problems

- Same as statistical vs. clinical relevance
- Many other parameter to consider
 - number of hyperparameters
 - number of variables
 - runtime
 - ...

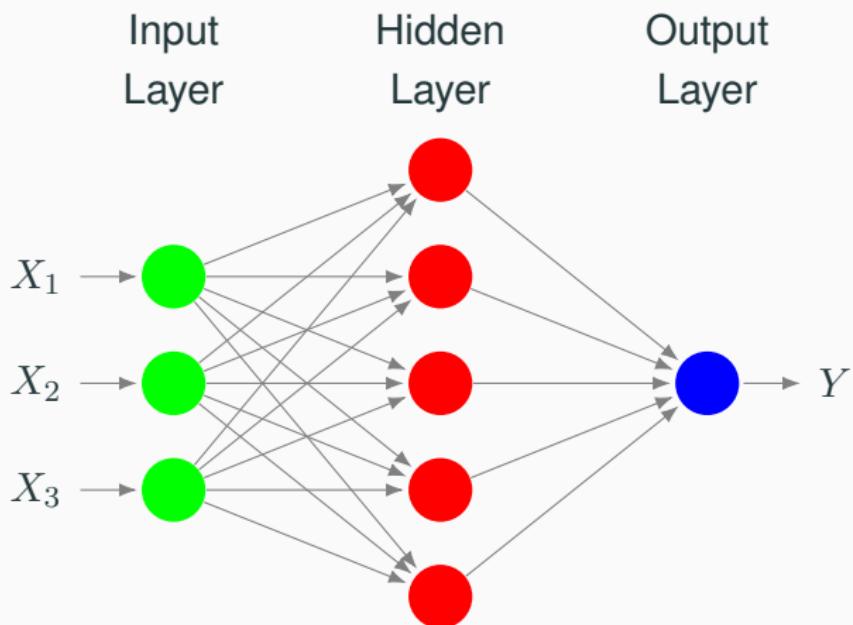
⇒ Practical relevance?

Part 9

Artificial neural networks

Artificial neural networks

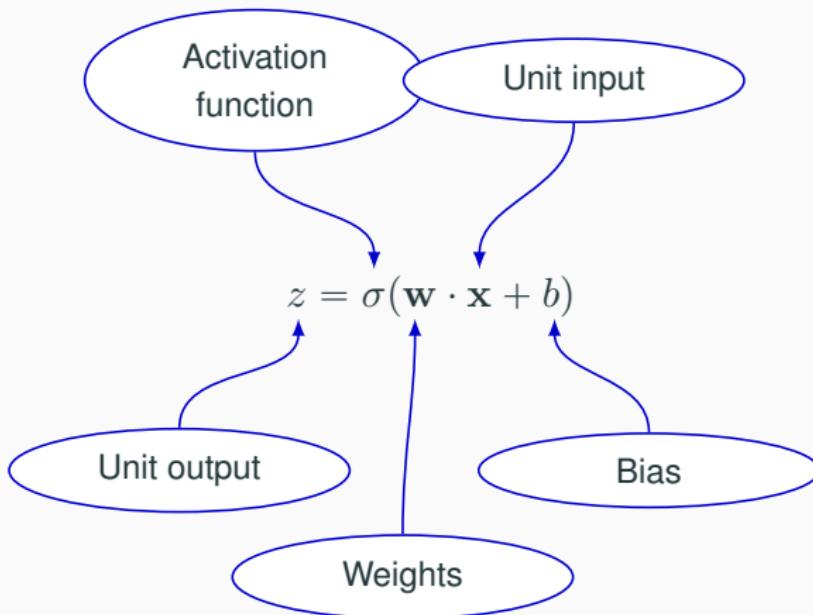
What are artificial neural networks?



Artificial neural networks

What are artificial neural networks?

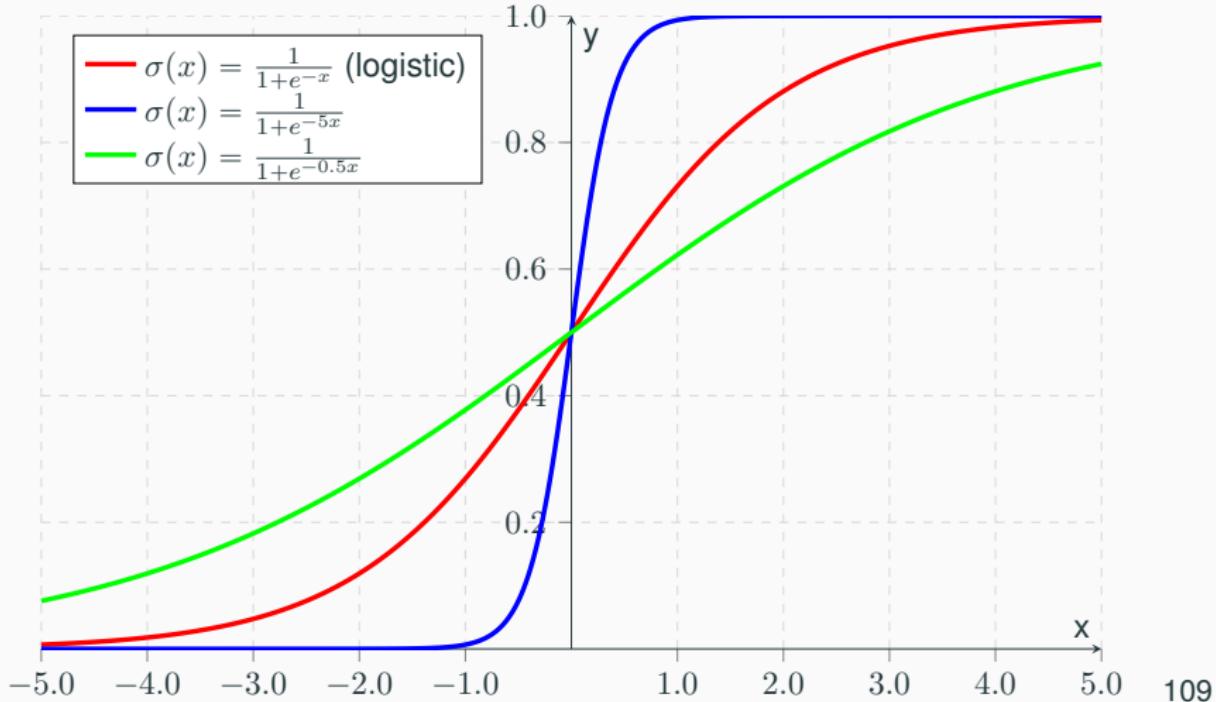
For each hidden unit:



Artificial neural networks

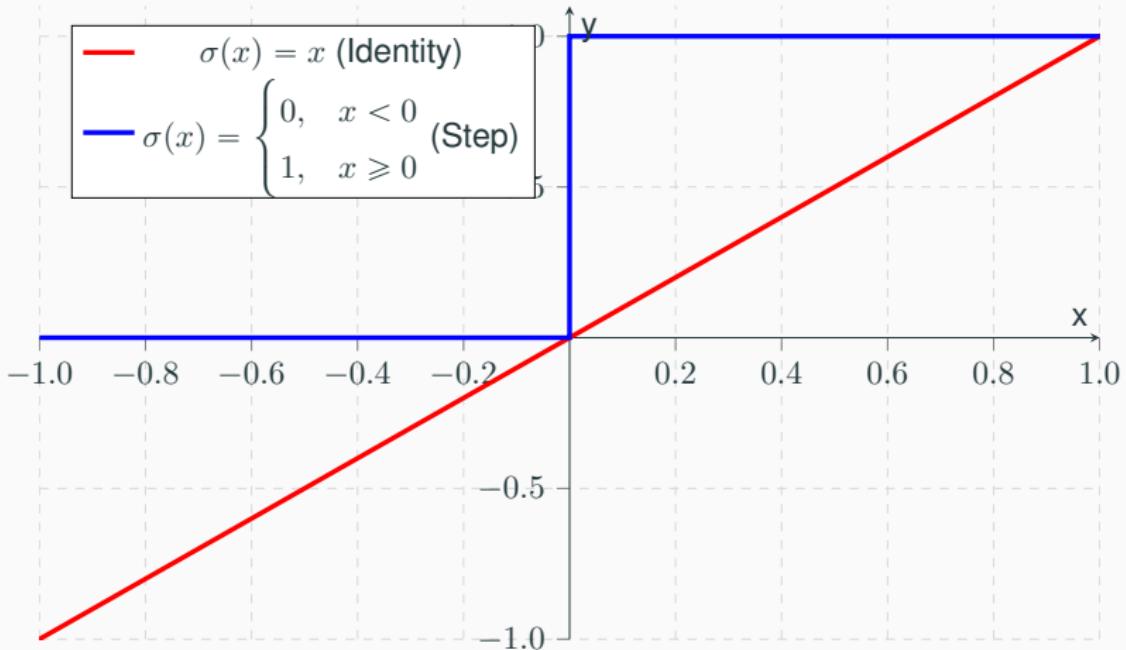
What is an activation function?

Typical choice: Sigmoid (e.g. logistic)



Artificial neural networks

What are special cases of activation functions?



Identity: Linear model

Step: Perceptron 110

Artificial neural networks

How to fit a neural network?

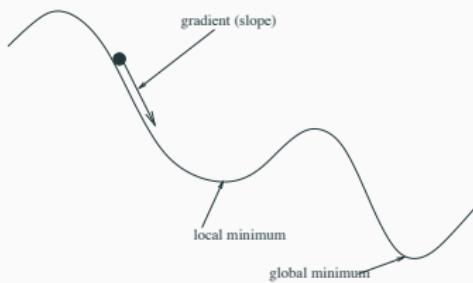
Loss function: Error as function of network weights, e.g,

$$L(\mathcal{W}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Aim: Find weights that minimize error

Gradient descent

Adjust weights in direction with steepest descent

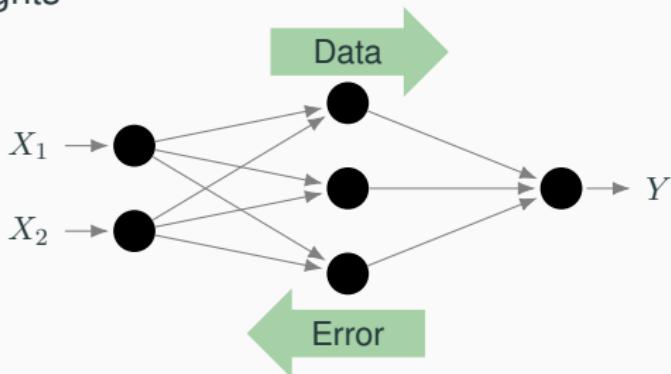


Artificial neural networks

How to fit a neural network?

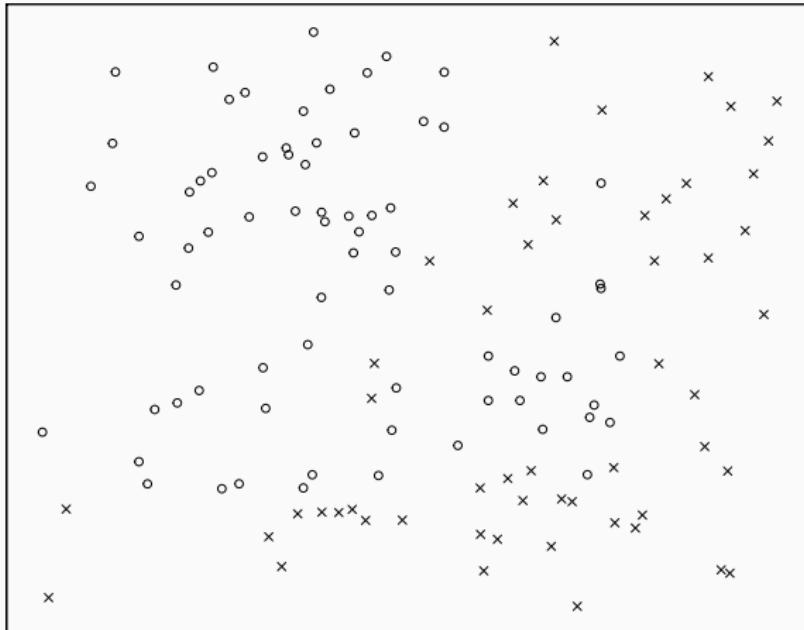
Backpropagation

1. Initialize weights randomly
2. For k iterations repeat
 - a) Compute error function
 - b) Adjust weights in output layer
 - c) Propagate error backwards through network and adjust weights



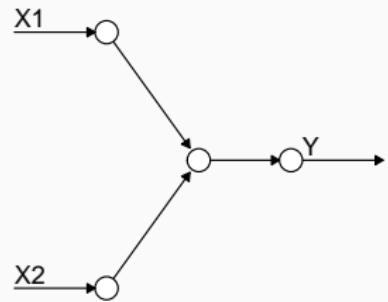
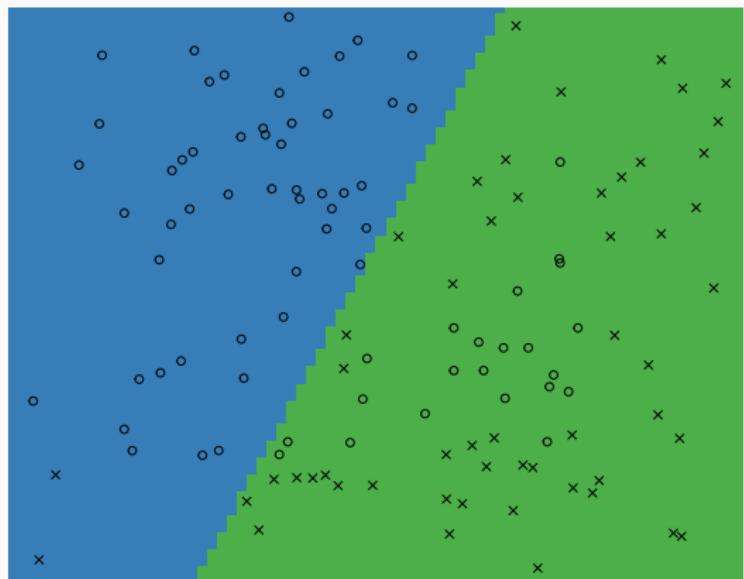
Artificial neural networks

Do neural networks overfit?



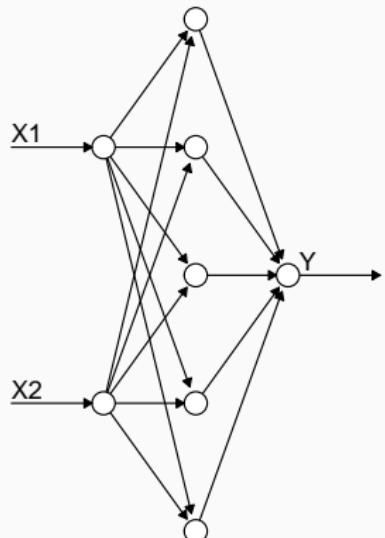
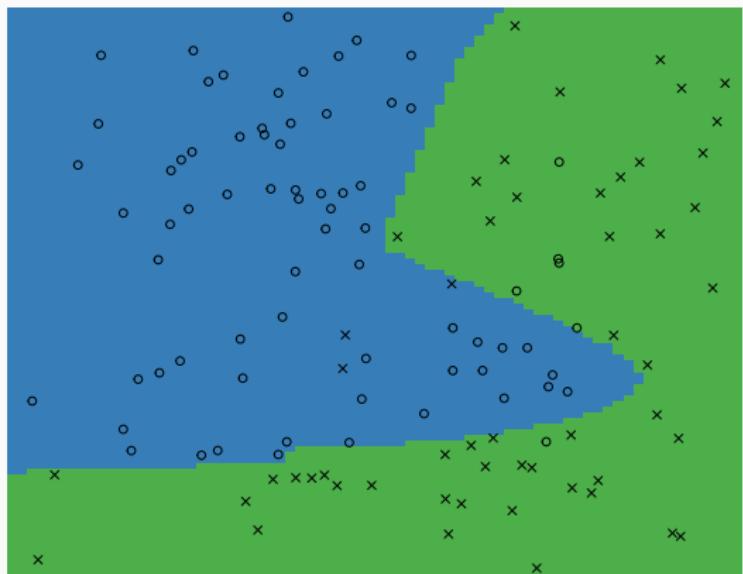
Artificial neural networks

Do neural networks overfit?



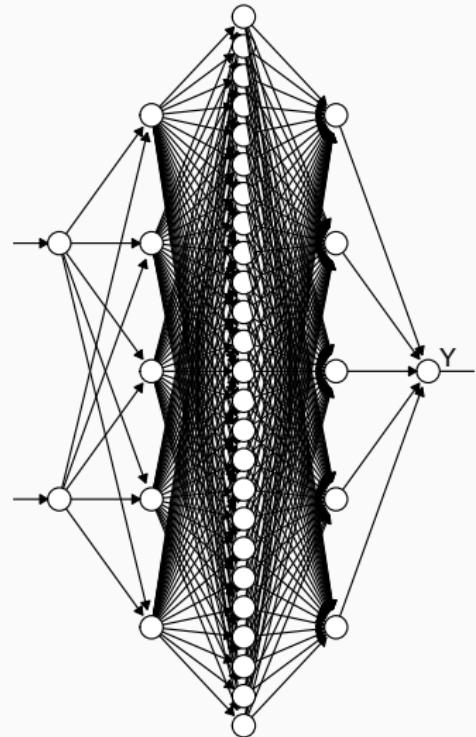
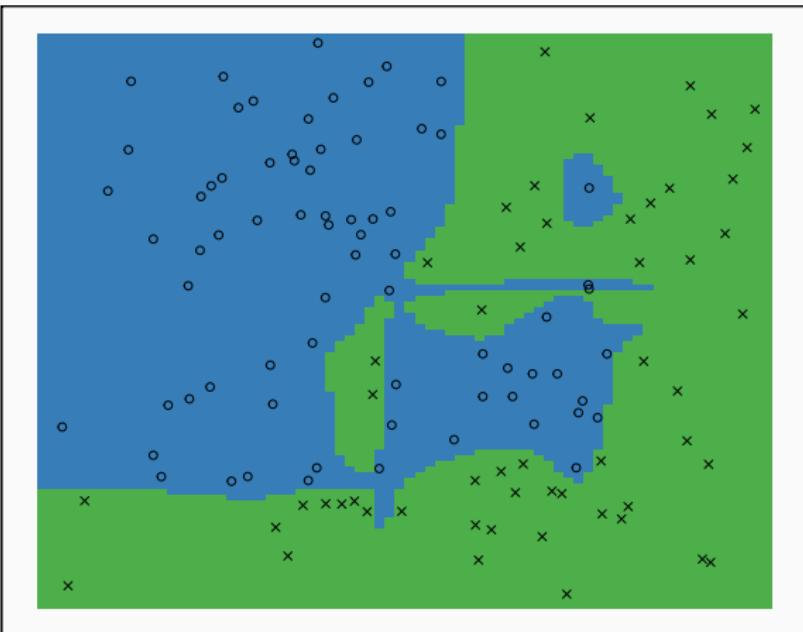
Artificial neural networks

Do neural networks overfit?



Artificial neural networks

Do neural networks overfit?



What is regularization?

L2 regularization

$$L(\mathcal{W}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{\mathbf{w} \in \mathcal{W}} \mathbf{w}^2$$



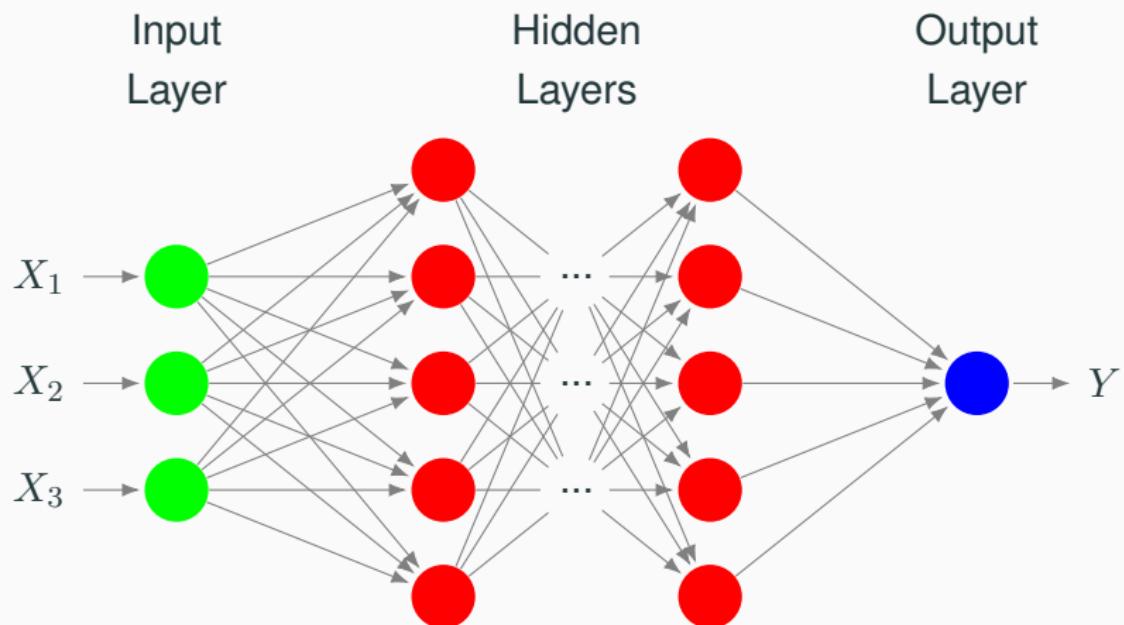
Alternative: Dropout

Temporarily remove units while fitting

Other alternative: Early stopping

Artificial neural networks

What is deep learning?

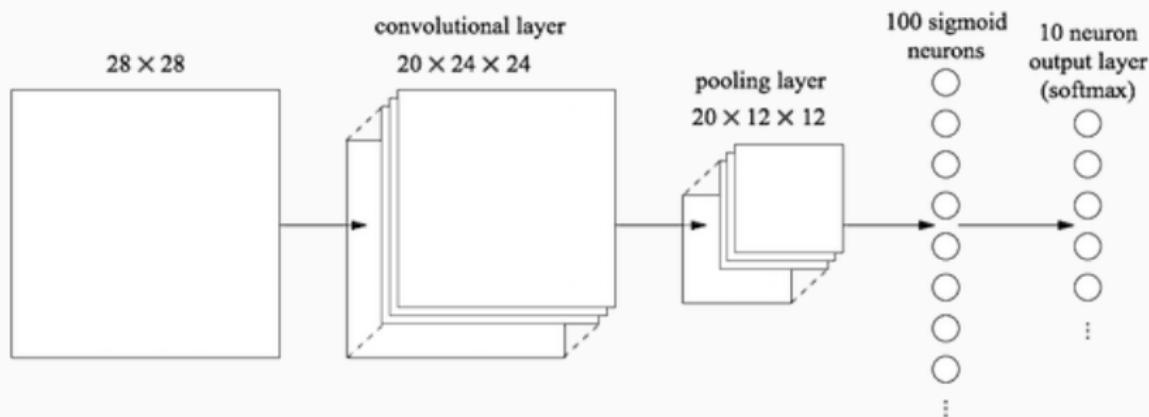


Artificial neural networks

What is deep learning?

Idea: Build hierarchy of concepts (representations)

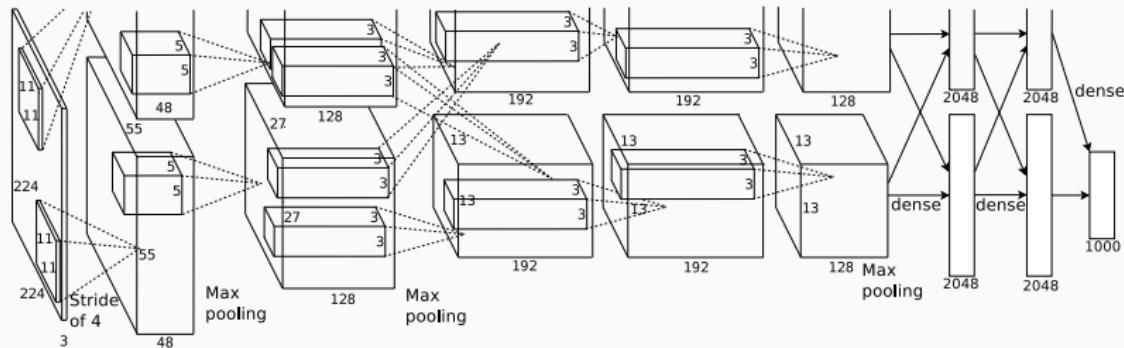
Convolutional neural networks



Artificial neural networks

What is deep learning?

Example



What are the advantages of neural networks?

- Can fit any complicated function almost perfectly
- Learns representations
- Online learning possible
- Prediction very fast (matrix multiplication)
- Can be parallelized (GPU computing)

What are the disadvantages of neural networks?

- Prone to overfitting
- Difficult to design good networks
- Interpretation very difficult (black box)
- Learning can be slow
- Statistical properties not well studied

Part 10

Specific endpoints

How can kNN be used with multi-category outcome?

- Algorithms unchanged
- Classification: most frequent category among kNN
- Probability estimation: estimate proportions of categories among kNN

How can random forests be used with multi-category outcome?

- Random forest algorithm unchanged
- Classification: use of extension of impurity measure to ordered and unordered multi-category outcome
 - Entropy → straightforward
 - Gini Index $\sum_{c=1}^C p_c(1 - p_c) = 1 - \sum_{c=1}^C p_c^2$
- Probability estimation: analogously to classification with Gini Index as criterion

How can boosting be used with multi-category outcome?

AdaBoost

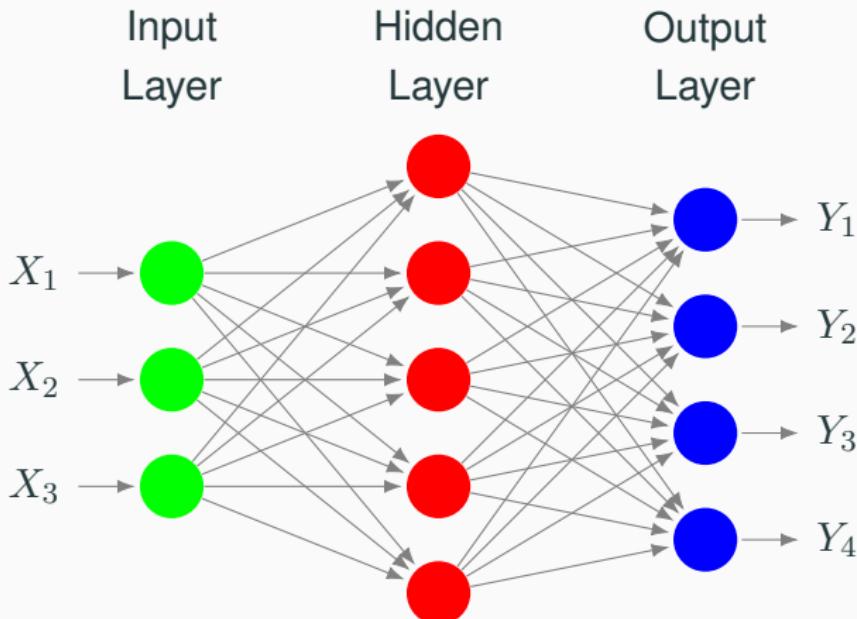
Essentially the same, but use base learner better than C -class random guessing, i.e., accuracy better than $1/C$.

Gradient boosting

- Use base learner supporting multi-category outcomes
- Use e.g. multinomial loss function

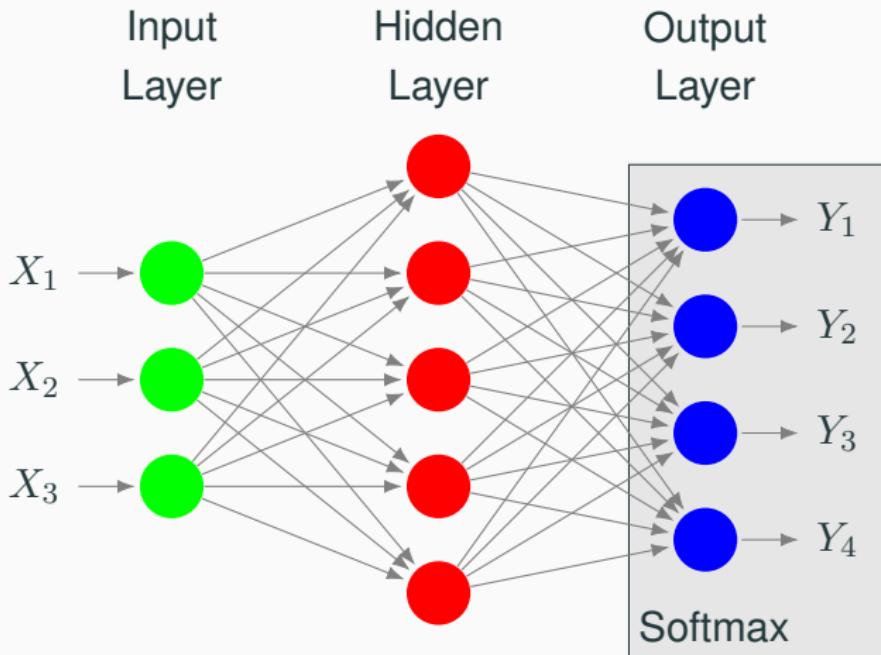
Machine learning for multi-category (multi-class) endpoints

How can neural networks be used with multi-category outcome?



Machine learning for multi-category (multi-class) endpoints

How can neural networks be used with multi-category outcome?



How can neural networks be used with multi-category outcome?

- One output unit per class
- Use softmax activation function in output layer:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{c=1}^C e^{x_c}}, j = 1, \dots, C$$

- Outputs Y_1, \dots, Y_C guaranteed to sum to 1

How can support vector machines be used with multi-category outcome?

Two approaches in case of C classes

1. One versus all(OVA)
 - a) Fit c different 2-class SVM classifiers $\hat{f}_c(x)$
 - b) Each class versus rest
 - c) Classify x to the class c for which $\hat{f}_c(x)$ is largest
2. One versus one (OVO) "pairwise coupling approach"
 - a) Fit all $\binom{C}{2}$ pairwise classifiers $\hat{f}_{cc'}(x)$
 - b) Classify x to the class that wins the most pairwise competitions

If C is not too large, use OVO.

How can support vector machines be used for probability estimation in the multi-category case?

Approach 1

Step 1 OVO approach (pairwise coupling)

Step 2 Combine probability estimates

Approach 2

Extend bracketing approach to multiclass probability estimation

Approach 3

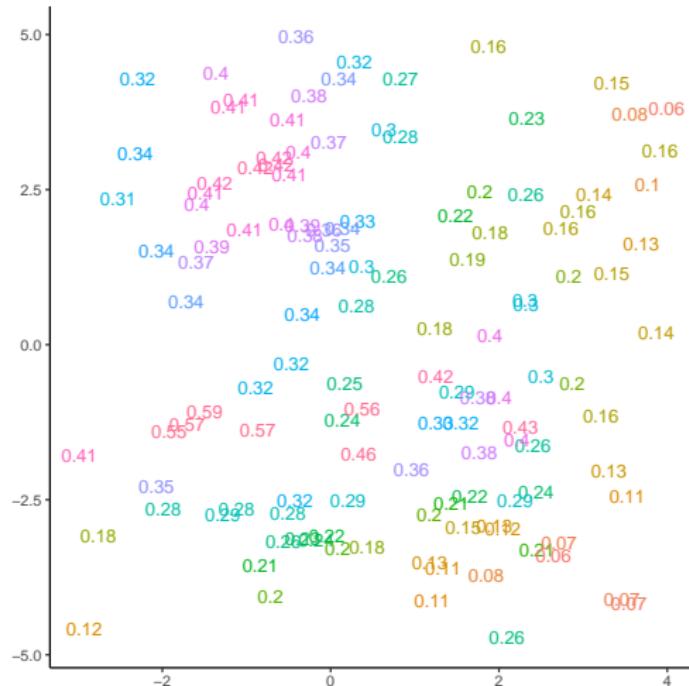
Re-fit specific Large-margin Unified Machine (LUMs)

Do the statistical properties of binary classification problems automatically generalize to the multi-category setting?

- No
- No explicit proofs available for kNN, random forests
- Erroneous proof available for support vector machines
- Sufficient and necessary conditions available for consistency and wide class of models

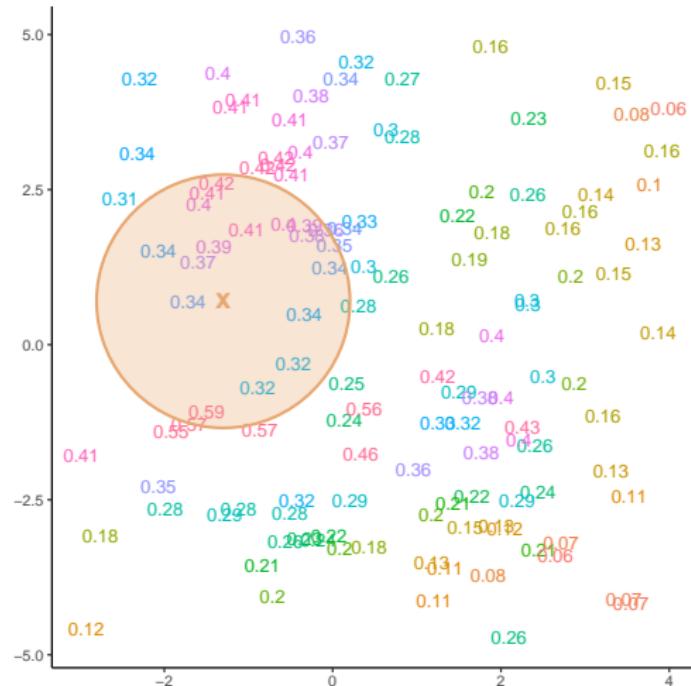
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with kNN?



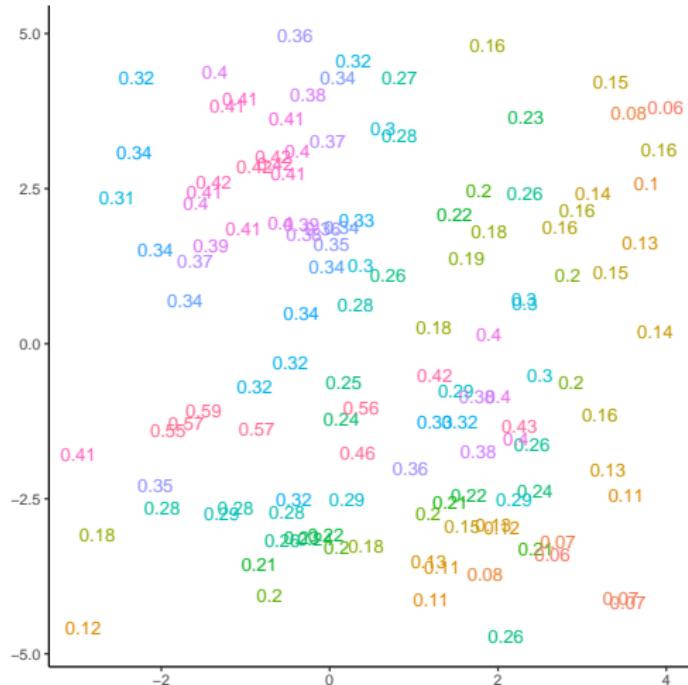
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with kNN?



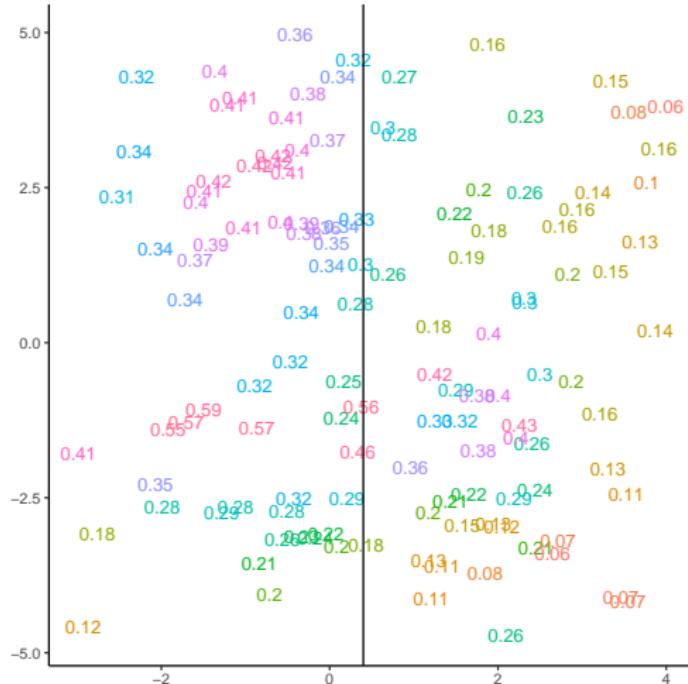
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with random forests?



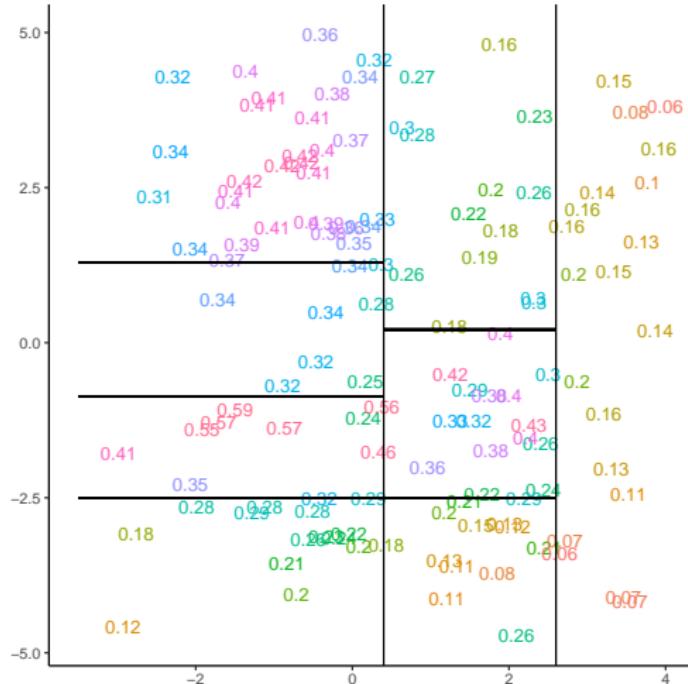
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with random forests?



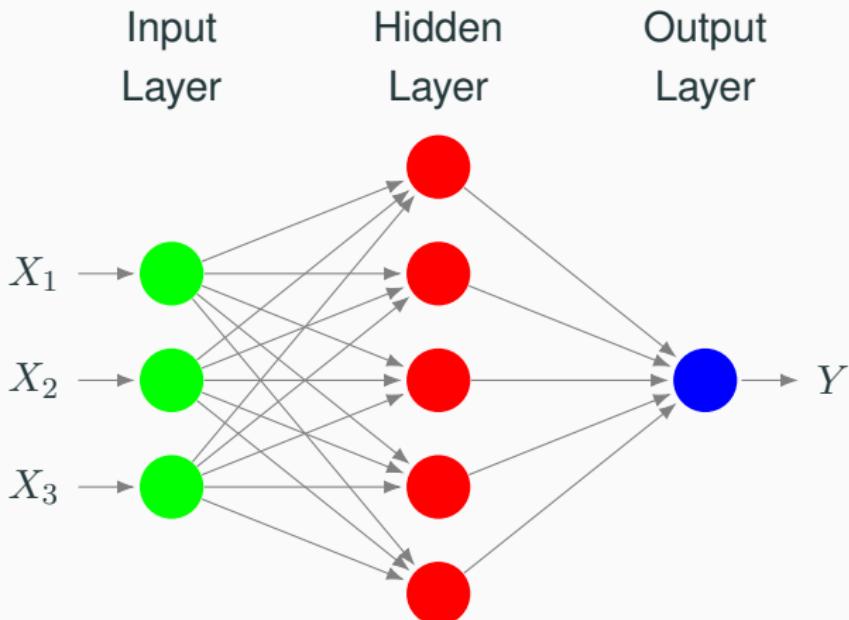
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with random forests?



Machine learning for continuous endpoints

How can continuous endpoints be analyzed with neural networks?



Machine learning for continuous endpoints

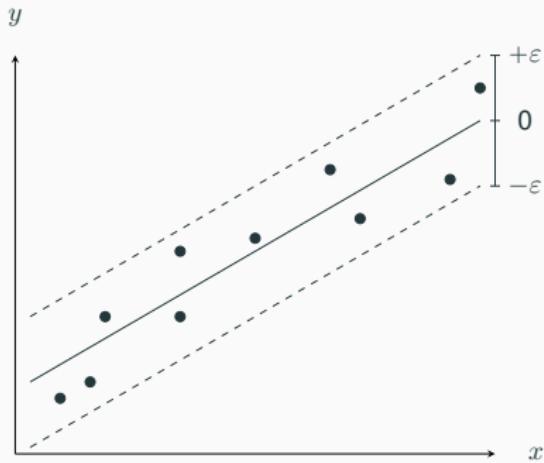
How can continuous endpoints be analyzed with neural networks?

- One output unit
- Use identity activation function in output layer:

$$\sigma(x) = x$$

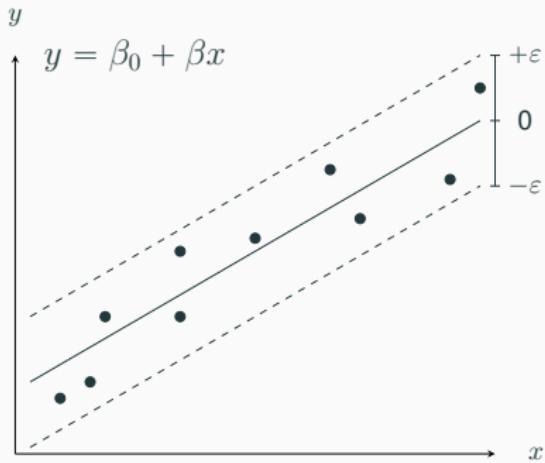
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?



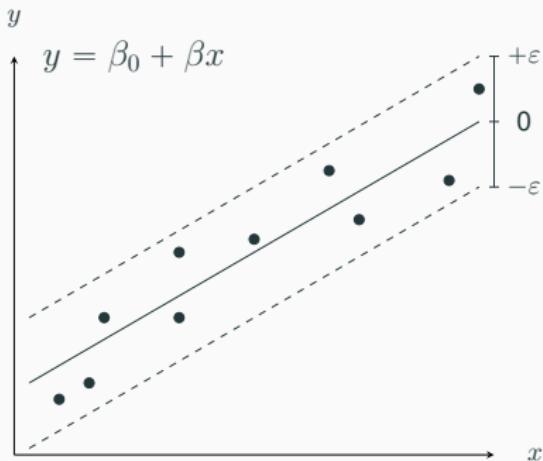
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?



Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?



Solution:

$$\min \frac{1}{2} \|\boldsymbol{\beta}\|^2$$

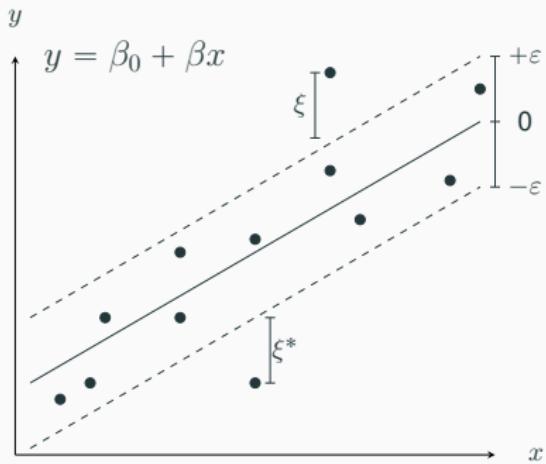
Constraints:

$$y_i - \beta_0 - \boldsymbol{\beta} \cdot \mathbf{x}_i \leq \epsilon$$

$$\beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}_i - y_i \leq \epsilon$$

Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?



Solution:

$$\min \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*)$$

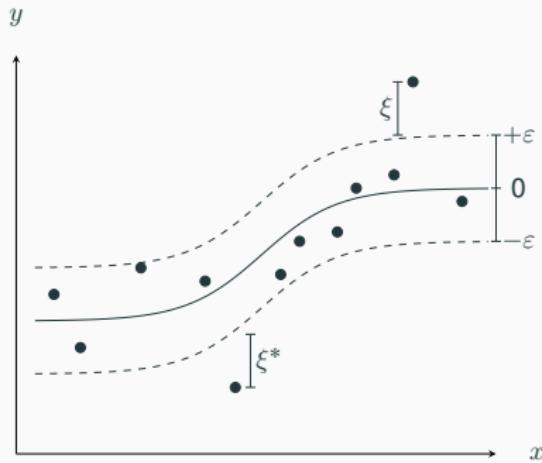
Constraints:

$$y_i - \beta_0 - \boldsymbol{\beta} \cdot \mathbf{x}_i \leq \epsilon + \xi_i$$

$$\beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}_i - y_i \leq \epsilon + \xi_i^*$$

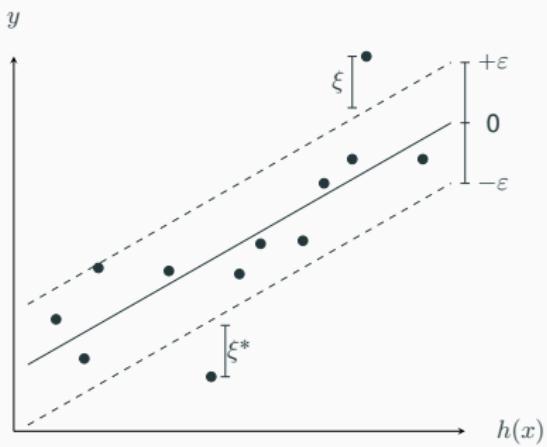
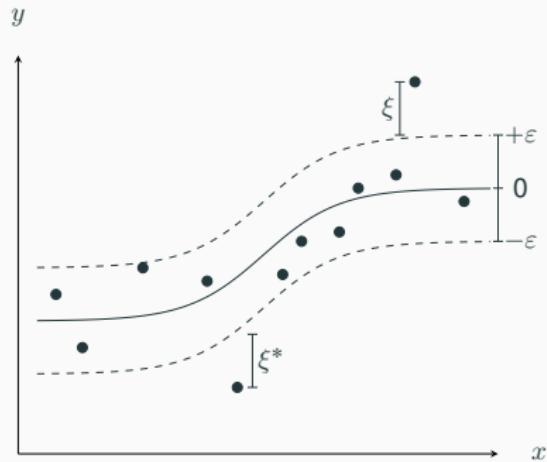
Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?



Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?



Machine learning for continuous endpoints

How can continuous endpoints be analyzed with support vector regression?

Linear kernel support vector regression

$$y = \beta_0 + \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \mathbf{x}_i, \mathbf{x}_i^* \rangle$$

Non-Linear kernel support vector regression

$$y = \beta_0 + \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}_i^*) \rangle$$

$$y = \beta_0 + \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}_i^*)$$

Machine learning for survival endpoints

How can nearest neighbor methods be used for survival analysis?

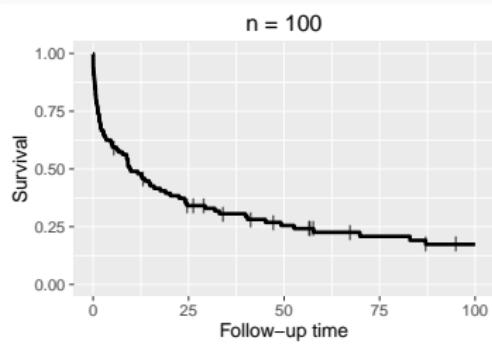
Algorithm unchanged

Estimate function of interest using kNN, e.g.,

- Kaplan-Meier estimator
- Nelson-Aalen estimator
- Median survival time
- ...

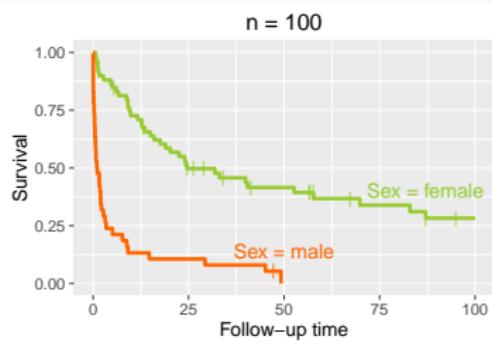
Machine learning for survival endpoints

How can random forests be used for survival analysis?



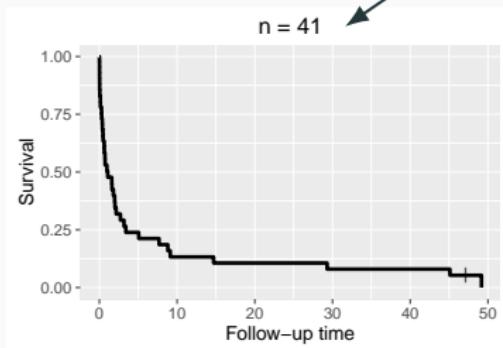
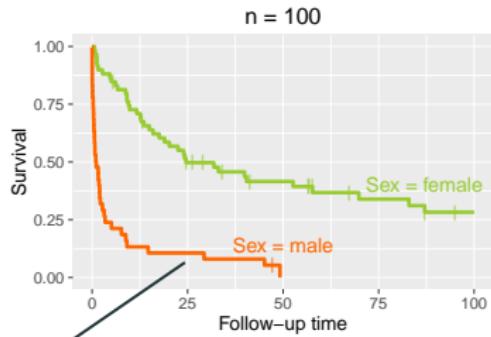
Machine learning for survival endpoints

How can random forests be used for survival analysis?



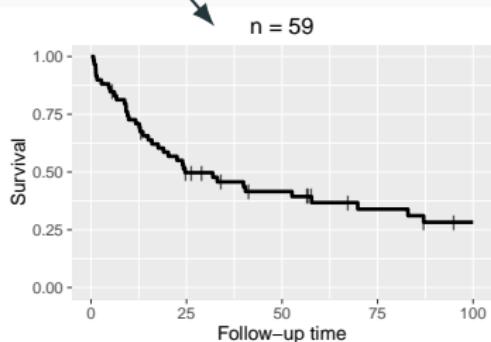
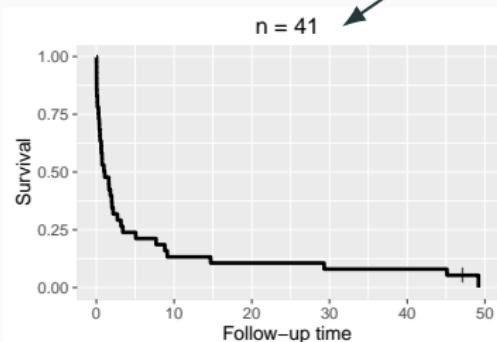
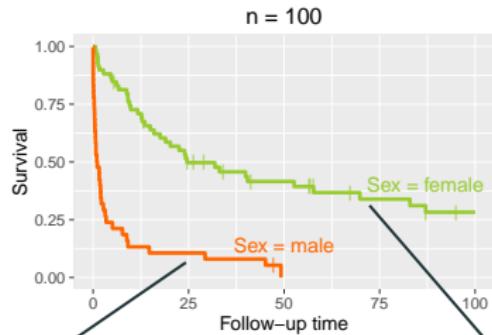
Machine learning for survival endpoints

How can random forests be used for survival analysis?



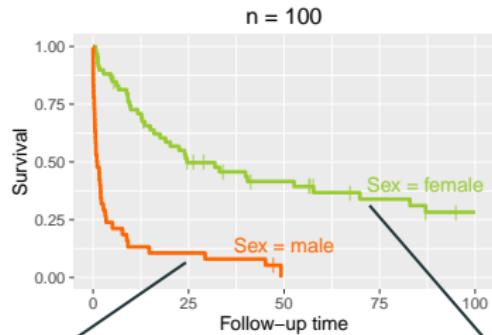
Machine learning for survival endpoints

How can random forests be used for survival analysis?

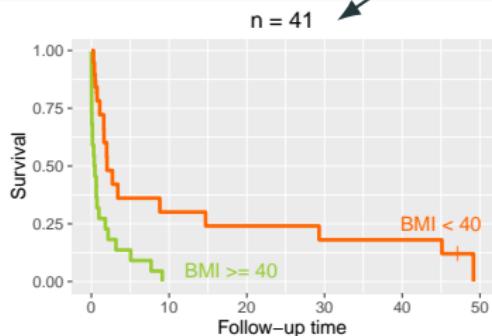


Machine learning for survival endpoints

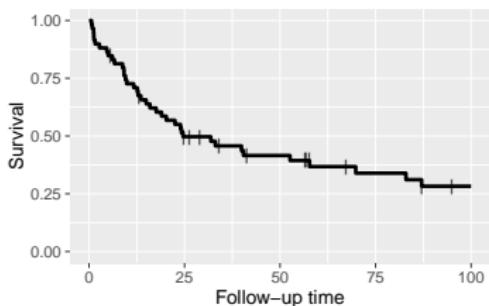
How can random forests be used for survival analysis?



n = 41

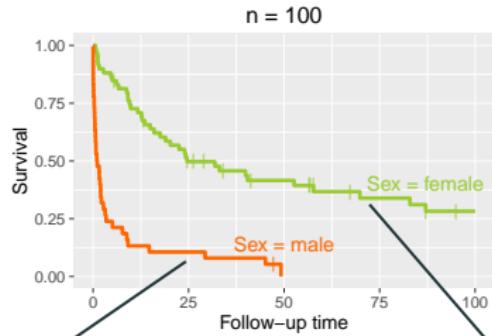


n = 59

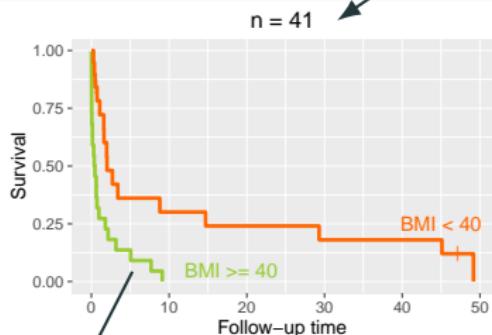


Machine learning for survival endpoints

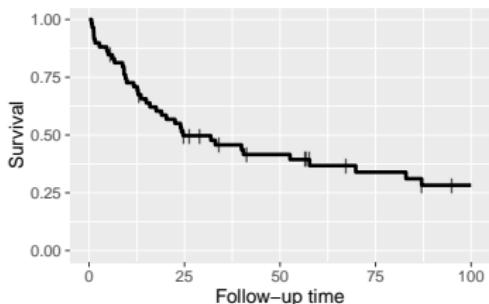
How can random forests be used for survival analysis?



n = 41



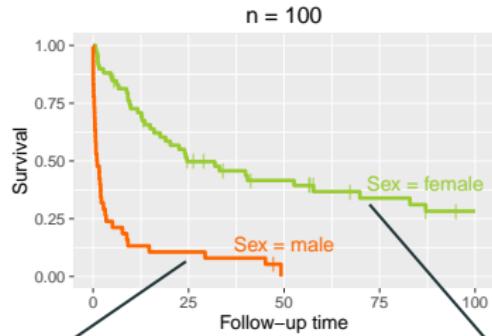
n = 59



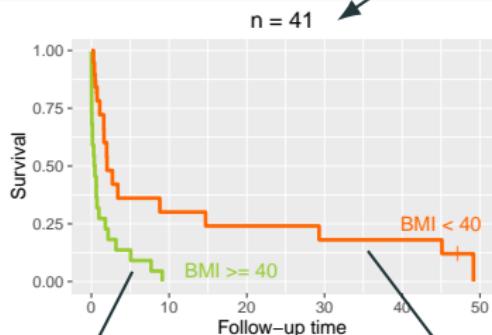
...

Machine learning for survival endpoints

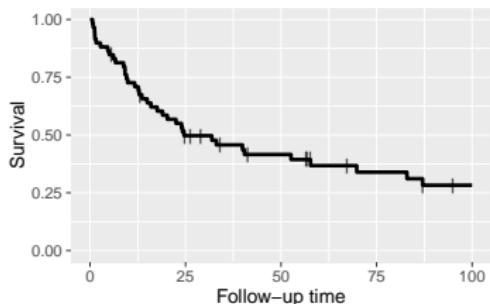
How can random forests be used for survival analysis?



n = 41

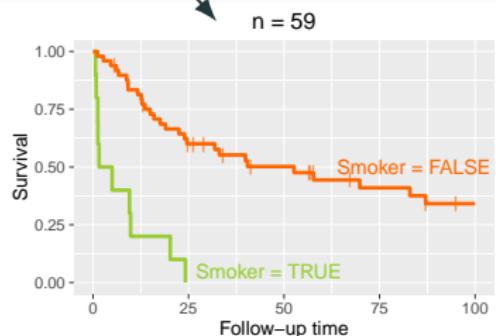
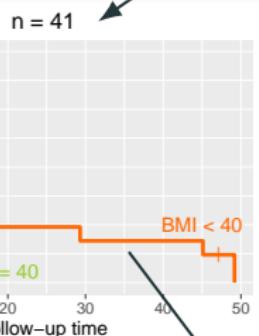
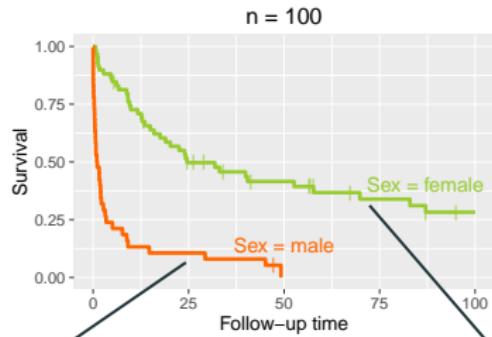


n = 59



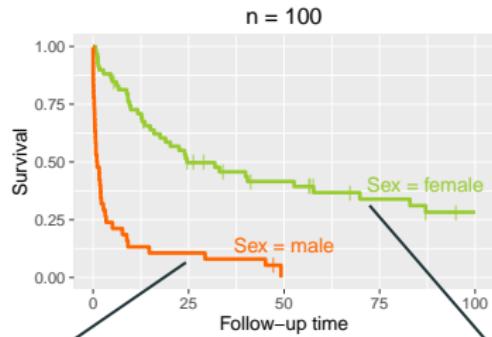
Machine learning for survival endpoints

How can random forests be used for survival analysis?

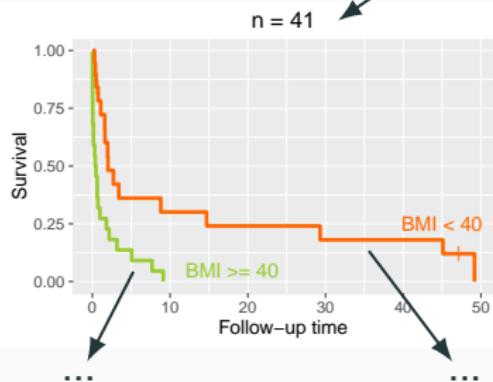


Machine learning for survival endpoints

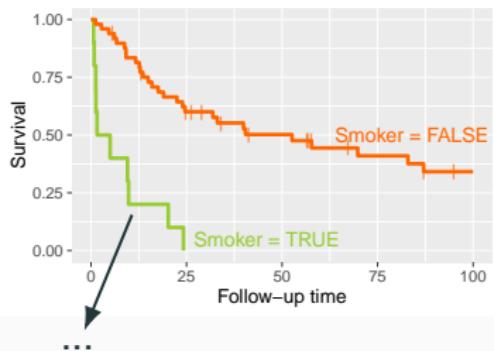
How can random forests be used for survival analysis?



n = 41

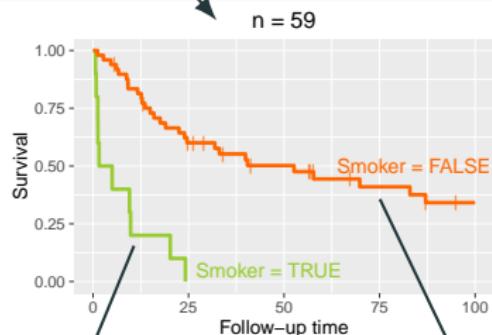
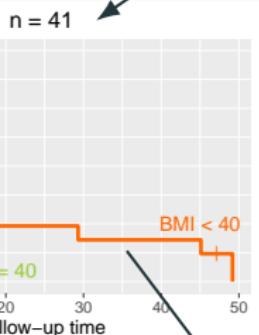
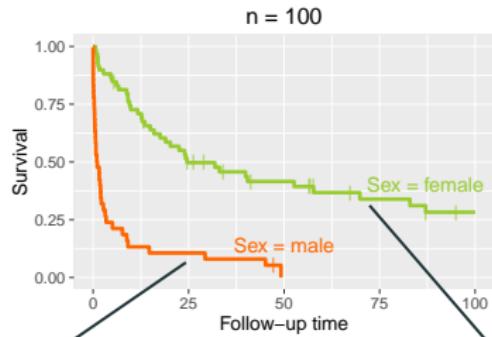


n = 59



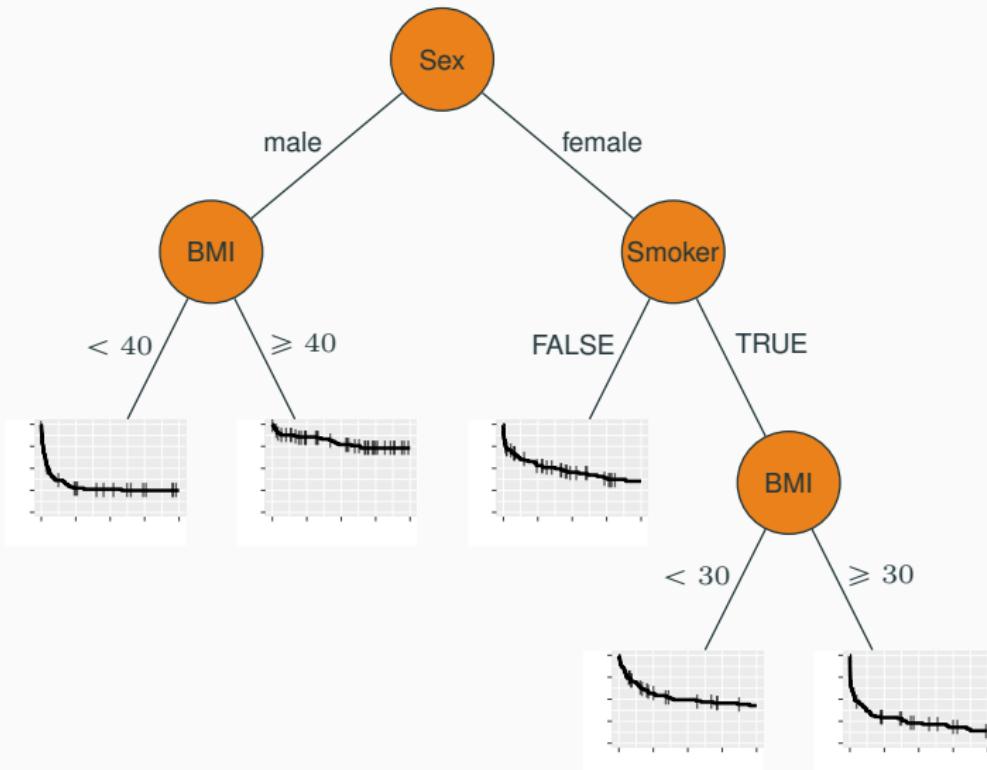
Machine learning for survival endpoints

How can random forests be used for survival analysis?



Machine learning for survival endpoints

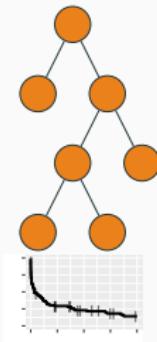
How can random forests be used for survival analysis?



Machine learning for survival endpoints

How can random forests be used for survival analysis?

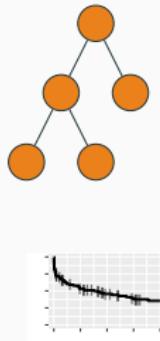
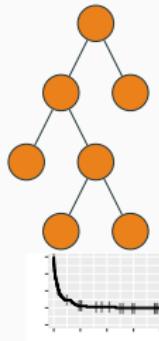
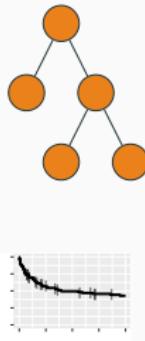
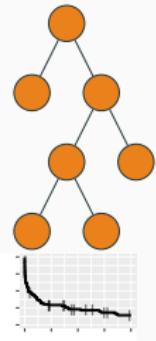
Bootstrap 1



Machine learning for survival endpoints

How can random forests be used for survival analysis?

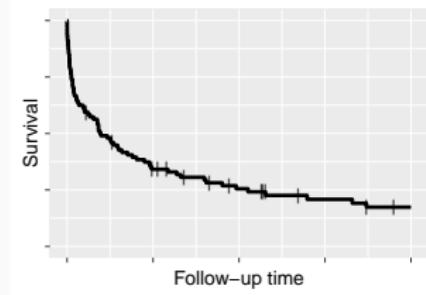
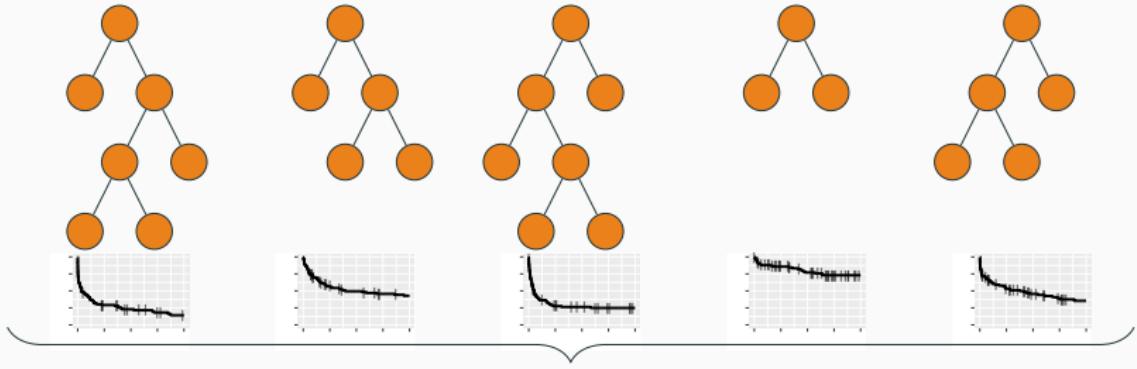
Bootstrap 1 Bootstrap 2 Bootstrap 3 Bootstrap 4 Bootstrap 5



Machine learning for survival endpoints

How can random forests be used for survival analysis?

Bootstrap 1 Bootstrap 2 Bootstrap 3 Bootstrap 4 Bootstrap 5



Which random forest approach and which split criterion should be used for random forest with survival outcome?

Biased split point selection

As before: continuous covariates have more possible split points

Examples as before

- Sex: 2 unique values
- Medication type: few unique values
- Age (in years): many unique values
- Biomarker: n unique values

Which random forest approach and which split criterion should be used for random forest with survival outcome?

Approaches

- Standard random forests with
 - Log-rank statistic as split criterion
 - Harrell's C index as split criterion
- p -value based 2-step random forests
 - Conditional inference forests with linear rank statistic
 - Maximally selected rank statistic

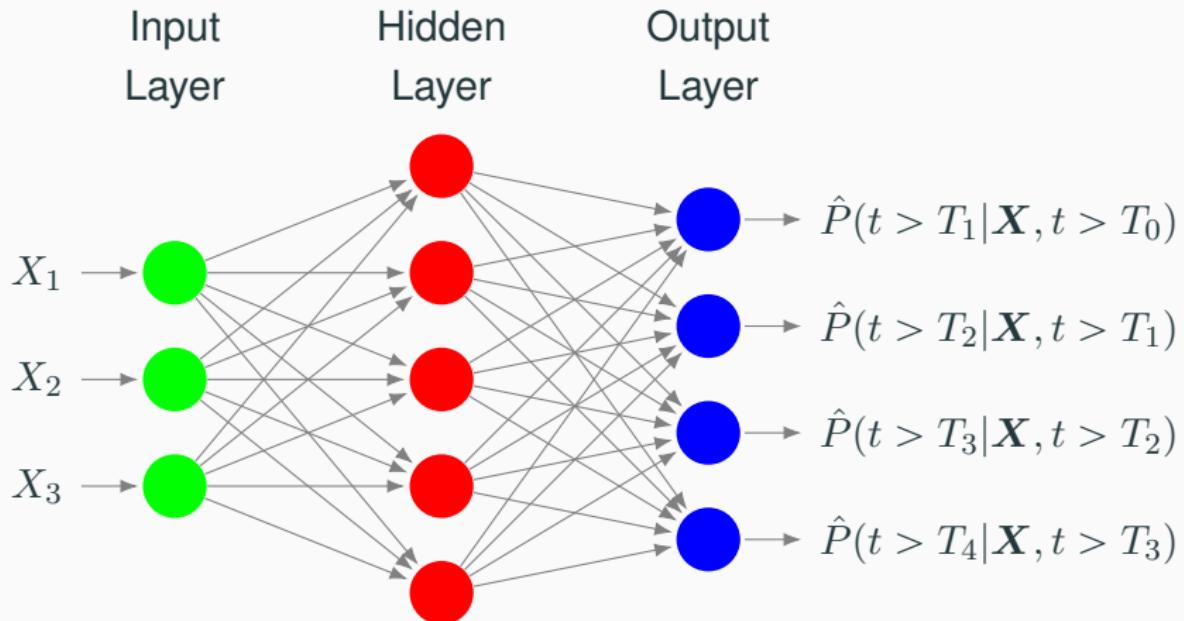
Which random forest approach and which split criterion should be used for random forest with survival outcome?

Summary

1. Log-rank to be used in case of many noise variables
2. Harrell's C index to be used in case of few variables
3. Standard random forest biased
4. Use 2-step subsampling random forest
 - 4.1 CIF slow but excellent
 - 4.2 Maximally selected rank statistics fast but with small bias

Machine learning for survival endpoints

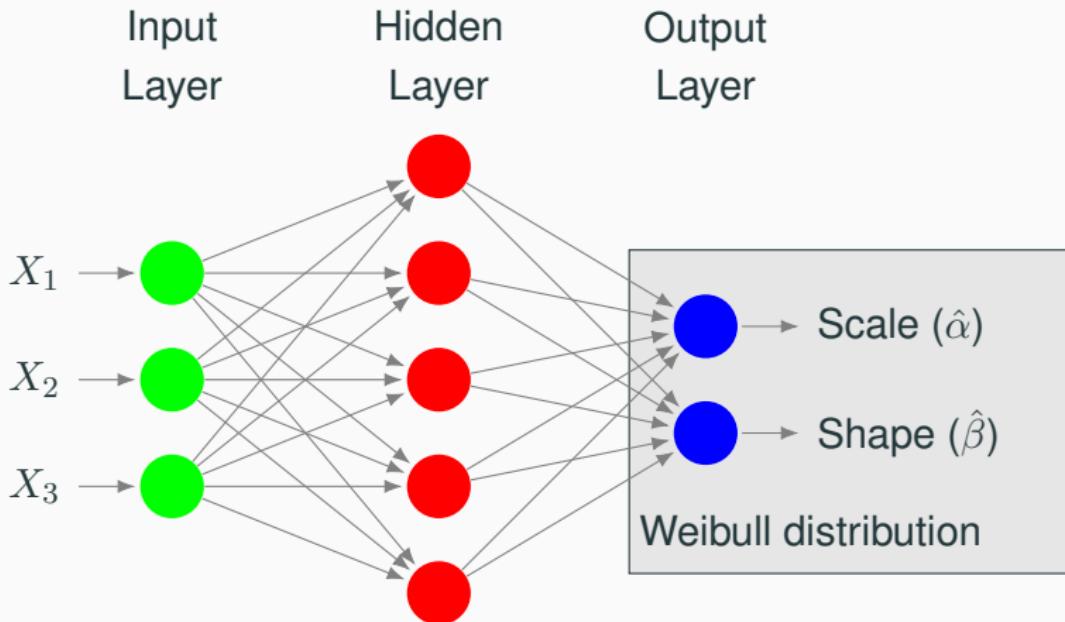
How can neural networks be used for survival analysis?



Loss function: Log-likelihood of discrete hazard model

Machine learning for survival endpoints

How can neural networks be used for survival analysis?



Loss function: Log-likelihood of Weibull distribution

Machine learning for survival endpoints

How can SVMs be used for survival analysis?

Approaches

- Ranking: Compare pairs of observations, classify larger survival time
- Regression: Penalization for incorrect predictions of censored survival times if predicted time is lower than observed time
- Hybrid: Minimize weighted sum of ranking and regression functions

Remarks

- Hybrid approach performs best
- Long runtimes for medium and large datasets

Part 11

Variable importance

Variable selection

What are general ideas for variable importance?

Permutation importance

Idea: The larger the error after permutation, the more important the variable

Algorithm for variable j :

1. Fit model on original data
2. Determine error e (e.g. test data, cross validation)
3. Permute values x_j of variable j
4. Determine error after permutation e_j
5. Permutation importance: $VIM_j = e_j - e$

What are ideas for model-specific variable importance?

kNN

No specific methods known

Decision trees

- Count number of splits on a variable
- Sum of impurity reduction of all splits on a variable

Boosted trees

Same as single decision trees

What are ideas for random forest variable importance?

Impurity importance

As in single decision trees → Gini importance

Permutation importance

- Use out-of-bag (OOB) performance estimate for each tree
- Average tree results

Bias-corrected impurity importance

1. Create pseudo-variables by permutation
2. Subtract pseudo-variable importance from original importance

Fast but requires re-fitting for prediction

Variable importance

What are permutation importance (PI) and scaled PI?

1. For all $b = 1$ to $B = \text{ntree}$ do
 - a) Grow tree
 - b) Determine error on OOB data (classification error, Brier score, ...) e_b^{OOB}
 - c) Permute values x_j of variable j
 - d) Determine error on OOB data using permuted data $e_{b,j}^{OOB}$
 - e) Determine difference in errors $e_b^{OOB} - e_{b,j}^{OOB}$

2. Permutation importance of variable j

$$PI_j = \frac{1}{B} \sum_{b=1}^B (e_b^{OOB} - e_{b,j}^{OOB})$$

3. Standard deviation s_j of differences

4. Scaled importance (scaled PI) of variable j

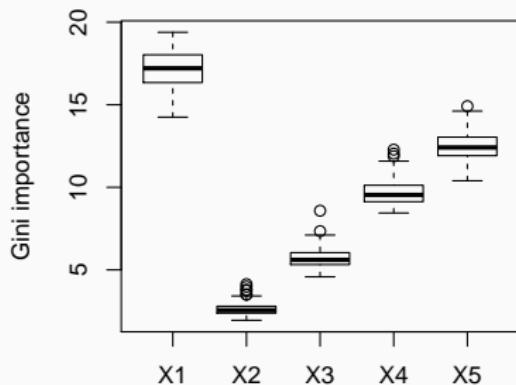
$$zPI_j = \frac{PI_j}{s_j / \sqrt{B}}$$

Variable importance

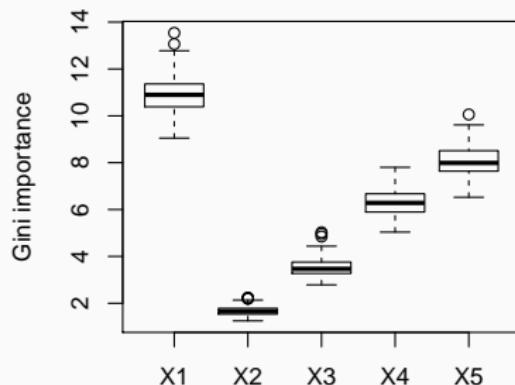
How does the Gini importance perform?

- All independent variables unimportant
- $X_1 \sim N(0,1)$, $X_2 \sim M(2)$, $X_3 \sim M(4)$, $X_4 \sim M(10)$,
 $X_5 \sim M(20)$

Bootstrap



Subsample



Also applies to single and boosted trees

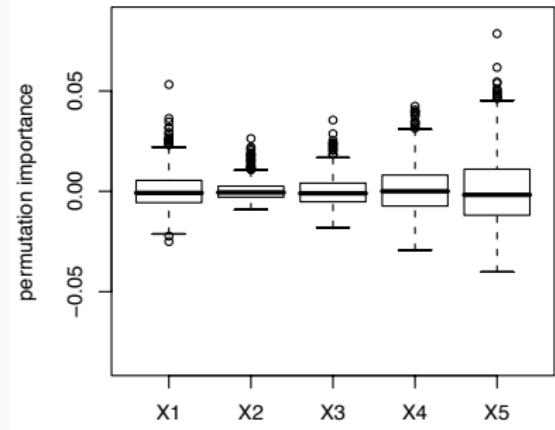
Variable importance

How does the permutation importance perform?

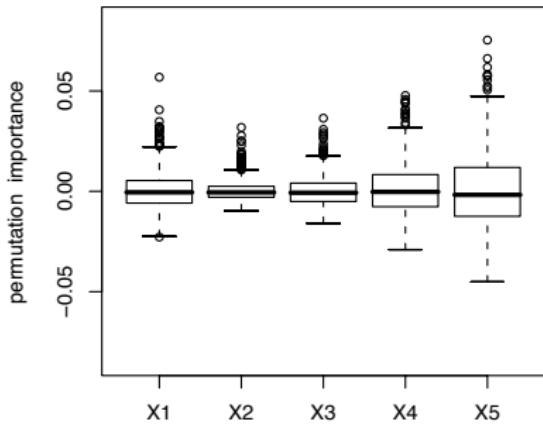
- All independent variables unimportant
- $X_1 \sim N(0,1)$, $X_2 \sim M(2)$, $X_3 \sim M(4)$, $X_4 \sim M(10)$,
 $X_5 \sim M(20)$

Standard random forest

Bootstrap



Subsample

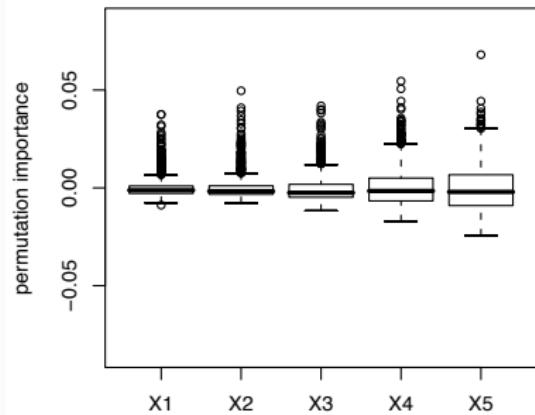


Variable importance

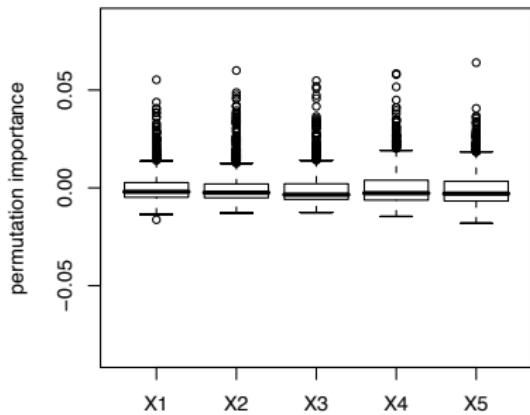
How does the permutation importance perform?

Conditional inference forest

Bootstrap



Subsample

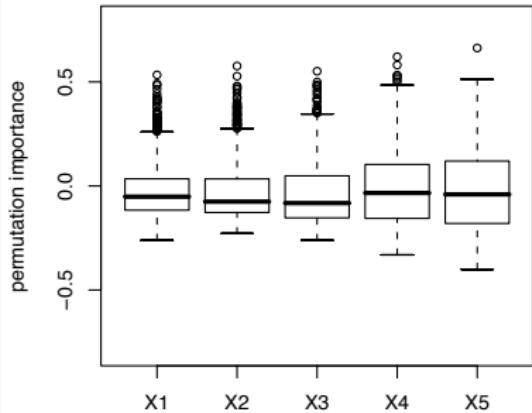


Variable importance

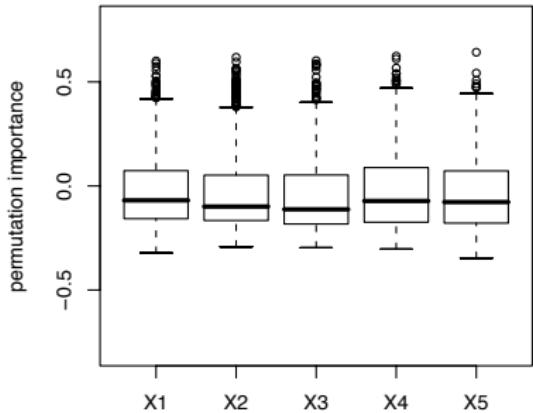
How does the scaled importance perform?

- Distribution skewed
- Not asymptotically normal
- Depends on number of trees in random forest
- Magnitude not interpretable

Bootstrap



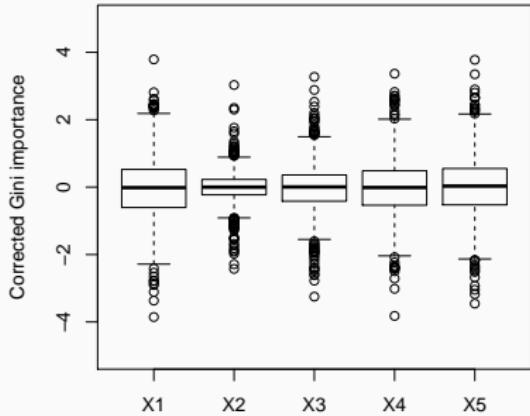
Subsample



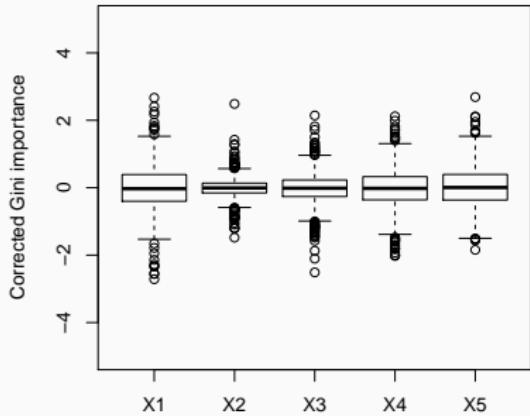
Variable importance

How does the corrected Gini importance perform?

Bootstrap



Subsample



What are ideas for model-specific variable importance?

SVM

Use parameters β as in regression

→ Only possible for linear kernel or without kernel trick

Neural networks

Idea: Partition output unit weights to each input unit

Newer methods available

Which importance measure should be used?

- Random forest measures most elaborated
- Gini importance *not* recommended
- Scaled permutation importance *not* recommended
- Standard permutation importance recommended
- Corrected Gini importance recommended
- Performance depends on number of trees, `mtry` and split criterion

How can importance be statistically tested?

Permutation approach

1. Fit model on original data set
2. Estimate variable importance
3. For $perm = 1$ to $Perm$
 - a) Permute y
 - b) Fit model on permuted data set
 - c) Estimate variable importances
4. p -value obtained as proportion of permuted importance scores at least as large as importance from original data

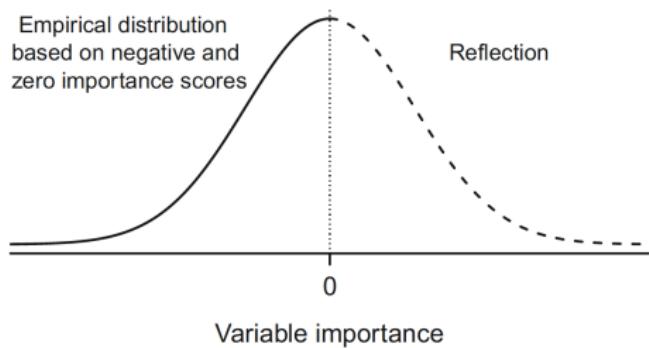
Only shown for random forests

Variable importance

How can importance be statistically tested?

Null importance test

- Under H_0 distribution of importance expected to be symmetric around 0
- Idea: Reconstruct distribution of null importance with importance of variables likely to be irrelevant:
 - Zero importance variables
 - Negative importance variables



How can importance be statistically tested?

Null importance test

- Only possible for high dimensional data
- Shown to work with random forests
 - Holdout importance (modification of permutation importance)
 - Corrected Gini importance (faster)
- No results for other learners

What are ideas for variable selection?

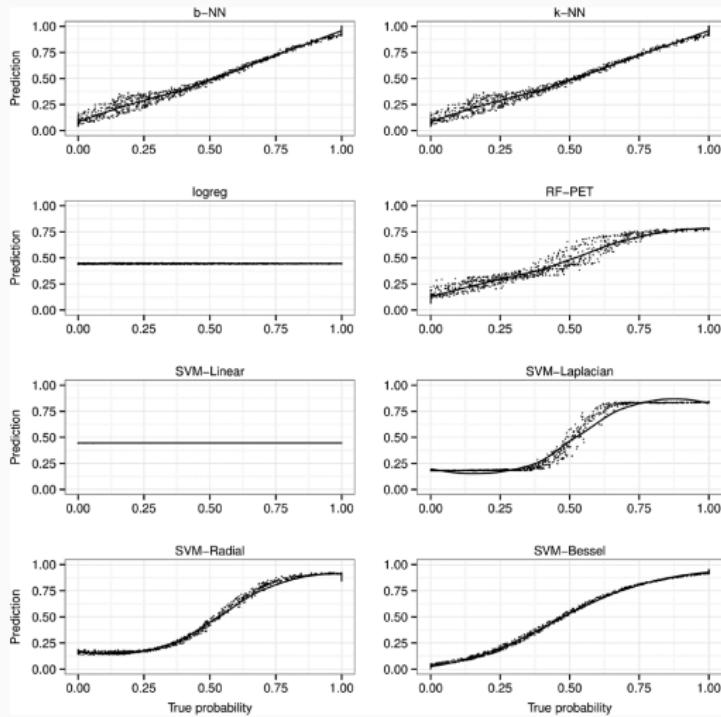
- Backward elimination
- Forward selection
- Use variable importance
 - Rank based
 - Test based
- *Intrinsic* variable selection in some learners
- Possible to use different method for variable selection
- Multivariate approaches recommended

Part 12

Discussion

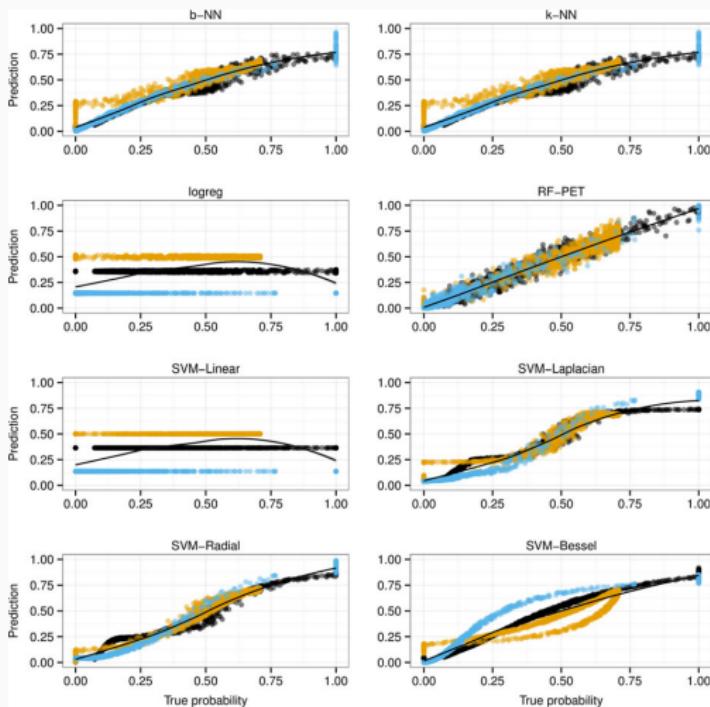
Discussion

Is there a single best learner? Mease model I



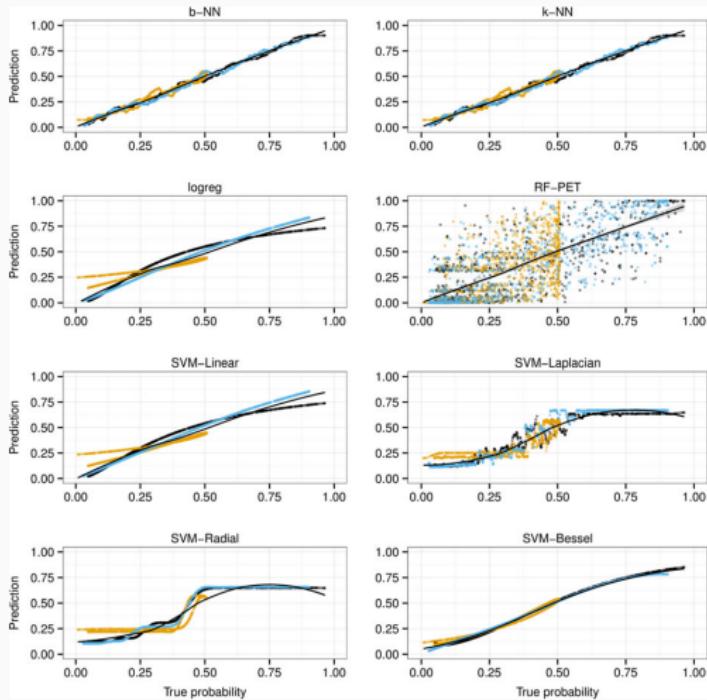
Discussion

Is there a single best learner? Mease model II



Discussion

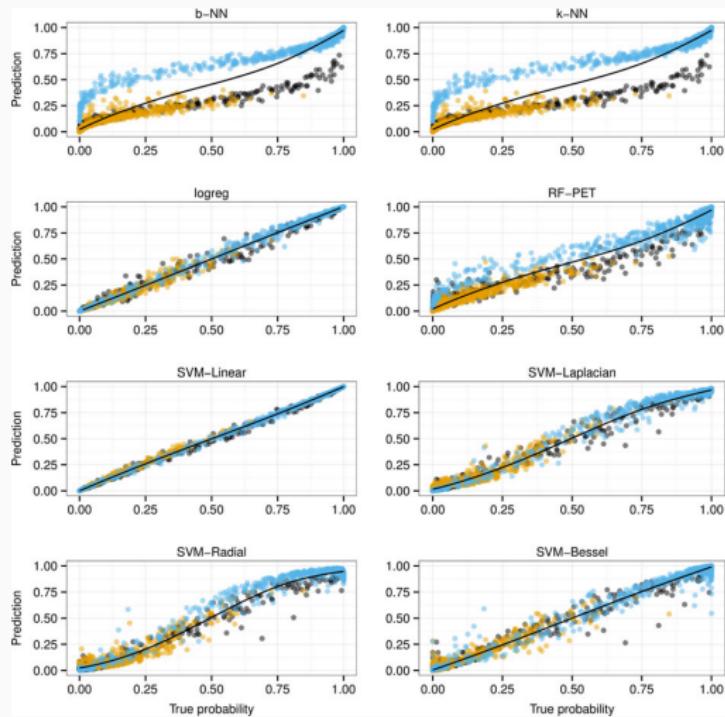
Is there a single best learner? Lee model



Discussion

Is there a single best learner?

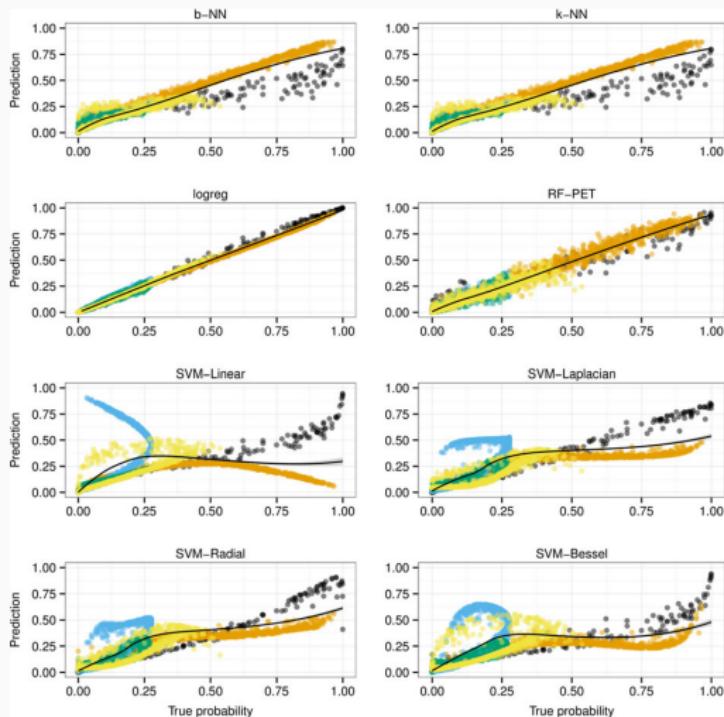
Li model I



Discussion

Is there a single best learner?

Li model II



Discussion

Is there a single best learner?

General recommendations

- Typically RF > Tree > kNN
- RF robust, easy to tune and fast
- Boosting often slightly better than RF on tabular data (when properly tuned)
- SVM good alternative for binary classification with numerical features (when properly tuned)
- Image, text and speech data → Deep Learning

Discussion

What are important aspects when applying machine learning?

- Subsampling statistically superior over bootstrapping
- Subsampling and bootstrapping lead to same prediction performance
- Interpretability versus performance
- Transferability

Discussion

What are important aspects when applying machine learning?

- Never use default parameter settings
- Tune parameters!
- Tune parameters jointly!
- Parameter tuning simple and straightforward for
 - kNN
 - bNN
 - random forests
- Parameter tuning complex and not straightforward for
 - SVM: parameters depend on kernel
 - ANN: tuning of architecture
- Use adequate resampling strategy
- Gold standard: nested cross validation

Discussion

What are important aspects when applying machine learning?

- Extensions required for
 - Matched samples
 - Longitudinal endpoints
 - Family data
- Development is triggered by other research areas for ANN and deep learning (image, signal)

Three parts of data science

	Statistics	Machine learning
Exploration	+/-	+/-
Effect estimation	+	-
Prediction	-	+

Acknowledgements

Some figures from course Introduction to Machine Learning
(I2ML)

Bernd Bischl, Fabian Scheipl, Daniel Schalk, Heidi Seibold et al.

https://github.com/compstat-lmu/lecture_i2ml

License: 

Develop lectures similar to open source software