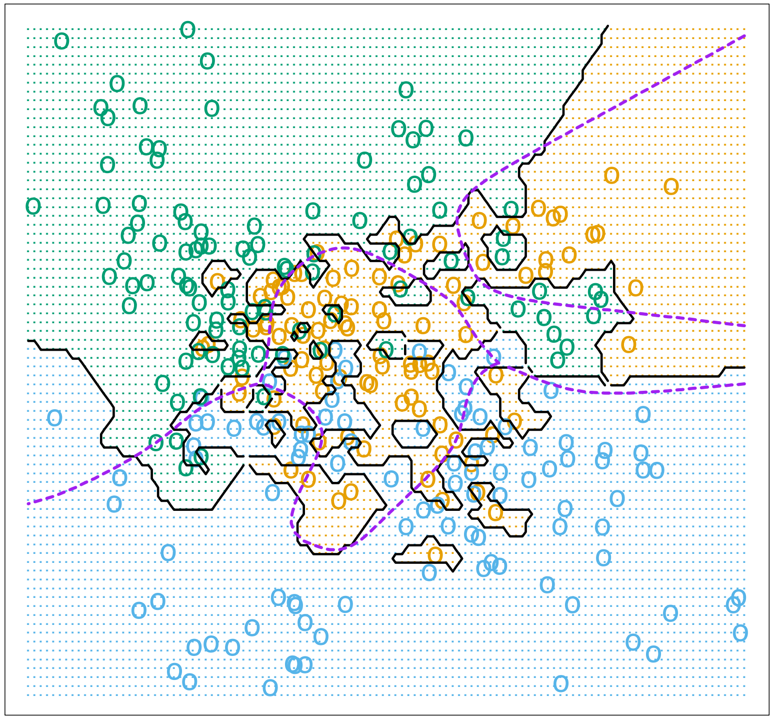
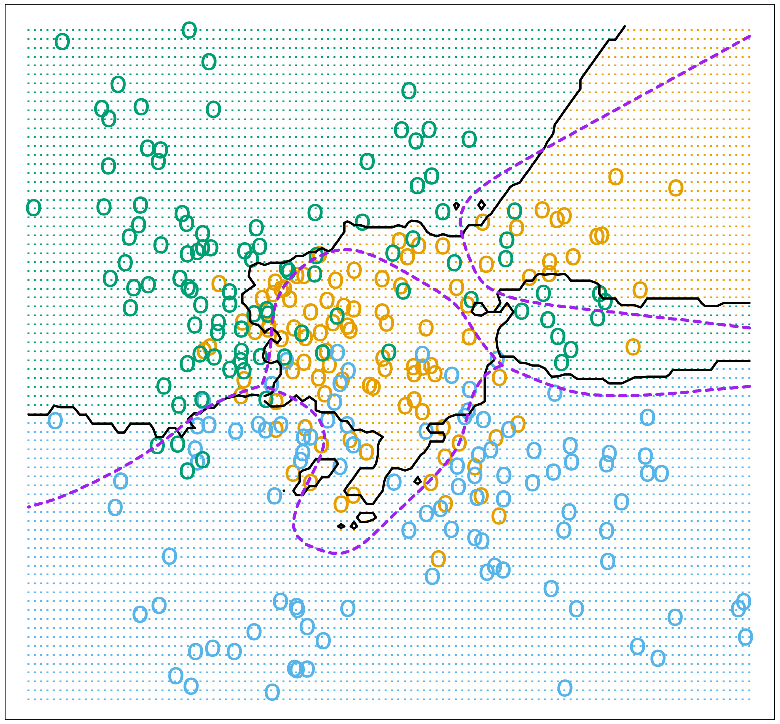
Model Selection and Cross Validation

OVERFITTING

- We had already noted that a classifier can adapt “too closely” to a training data set.
- If the classifier represents idiosyncracies of the training sample rather than the properties of the data source, it achieves small training error, but will not perform well on new data generated by the same data source.
- This phenomenon is called **overfitting**.



1-NN: Overfitting



15-NN: Better generalization

Source: Hastie/Tibshirani/Freedman, “The elements of statistical learning”

Complexity and Overfitting

Overfitting and flexibility

- How prone a classifier is to overfitting depends on how “flexible” it is.
- Linear classifier are not very likely to overfit.

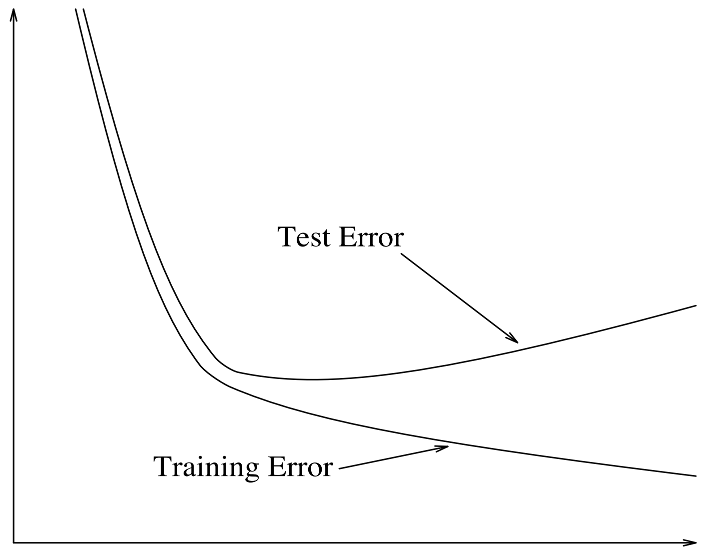
Example: Trees

- A tree of depth 1 is a linear classifier, and will not overfit any reasonably large data set.
- A tree with many splits can subdivide the sample space into small regions.
- Suppose we train a tree with M splits on n training points. If $M \approx n$, the tree can separate almost every training point off into a separate region. That is overfitting: The tree memorizes the training data.

More complex (= flexible) models are more likely to overfit.

Training vs Test Error

Conceptual illustration



Vertical: Error Horizontal: Model complexity (e.g. number of splits in tree)

- If classifier can adapt (too) well to data: Small training error, but possibly large test error.
- If classifier can hardly adapt at all: Large training and test error.
- Somewhere in between, there is a sweet spot.
- Trade-off is controlled by the parameter.

Implications for Training

- If we permit the training algorithm to make a classifier more flexible, it is likely to overfit.
- Example: If the training algorithm for a tree classifier can perform an arbitrary number of splits, it can achieve zero training error.

Avoiding overfitting

- Parameters that control flexibility (like the maximal number of splits in a tree) should be fixed during training.
- We have to develop alternative strategies to choose values for those parameters.

Types of parameters

It is customary to separate parameters into two types:

1. **Model parameters**
   The parameters that specify the solution.
   Examples:
   - Normal vector and offset of a linear classifier
   - Split points and tree structure of a tree classifier.

   (m-nearest neighbor classifiers are an exception: They have no such parameters.)
2. **Hyperparameters**
   Parameters that control the complexity of the solution.
   Examples:
   - Number of splits of tree classifier
   - number m of neighbors in m-nearest neighbor

Hyperparameters cannot be chosen by the training algorithm.

Selecting values for the parameters

- The model parameters are estimated by the training algorithm.
- Hyperparameters are often determined using data splitting methods.

Terminology: Models

Models

- Data mining and machine learning often loosely refers to a method as a *model*. It is difficult to give a definition that is both general and precise.
- A definition that often works for classification is: A **model** is the set of all possible classifiers that a given method can fit to training data. Each individual solution is often called a **hypothesis**.

Examples

- Linear classifier in $\mathbb{R}^d$: Model = all possible affine planes in $\mathbb{R}^d$,
  hypothesis = a specific affine plane.
- Tree classifer in $\mathbb{R}^d$: Model = all possible tree classifiers with a fixed number M of splits,
  hypothesis = classifier defined by one particular tree.

Models and hyperparameters

- Typically, all classifiers within a model should have the same complexity.
- For example: We think of trees with 1 split and trees with 2 splits as two distinct models.
- More generally: Different hyperparameter values define different models.

Cross Validation

Objective

- Cross validation is a method which tries to select the best model (e.g. all tree classifiers with 3 splits) from a given family of models (e.g. all tree classifiers).
- This is done using data splitting. Cross validation is a data splitting “protocol”.
- Assumption: Quality measure is predictive performance.
- "Set of models" can simply mean "set of different parameter values".

Terminology

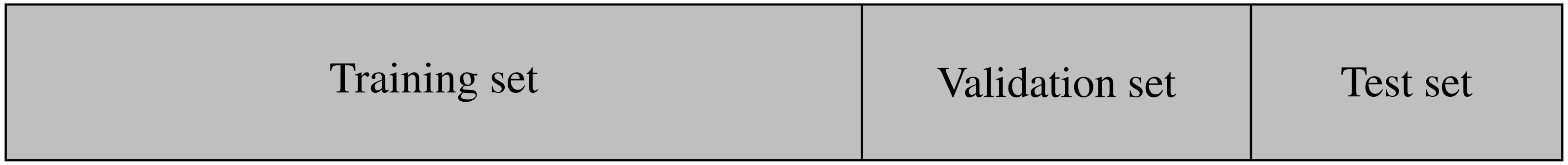
- The process of choosing a good model within a family of models is called **model selection**.

Cross Validation for Model Selection: Procedure

(From now on, we just write γ to denote the entire set of hyperparameters.)

Model selection

1. Randomly split data into three sets: training, validation and test data.



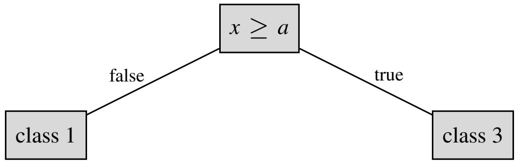
2. Train classifier on training data for different values of γ.
3. Evaluate each trained classifier on validation data (ie compute error rate).
4. Select the value of γ with lowest error rate.

Model assessment

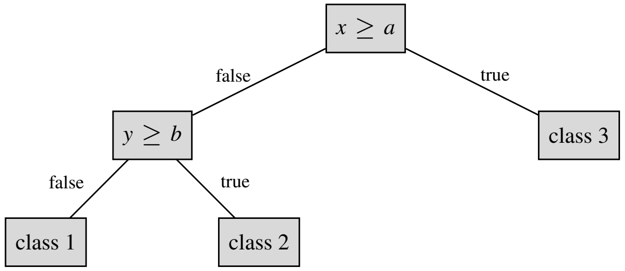
5. Finally: Estimate the error rate of the selected classifier on test data.

BASIC CV FOR A TREE

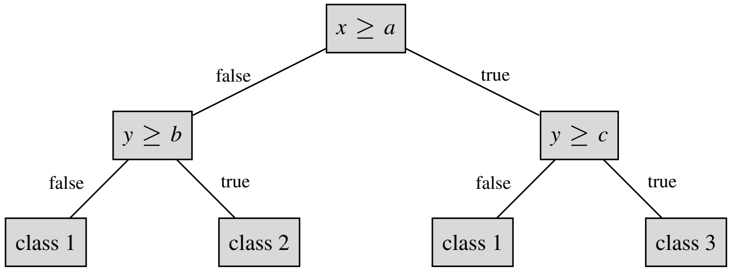
$M = 1$:

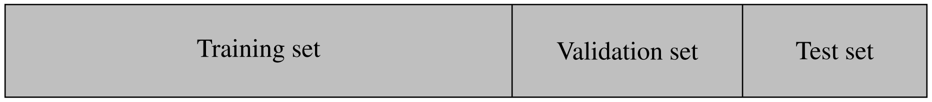


$M = 2$:



$M = 3$:





CV procedure

- Split labeled data into a training, validation and test set.
- For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the training set.
- For each M, estimate the error rate of the trained classifier on the validation set.
- Select the value of M with the smallest error rate; say this is $M = 2$.
- Estimate the error rate for $M = 2$ on the test set.

For prediction on new data, you now use the tree classifier with $M = 2$, and report its estimated error rate as that estimated on the test set.

Statistical Analysis

Meaning

- The quality measure by which we are comparing different classifiers f_γ (for different parameter values γ) is the risk

$$R\big(f_\gamma\big) = \mathbb{E}[L(y, f_\gamma(x))] \; .$$

- Since we do not know the true risk, we estimate it from data as $\hat{R}(f_\gamma)$.

Importance of model assessment step

- We always have to assume: Classifier is better adapted to *any* data used to select it than to actual data distribution.
- Model selection: Adapts classifier to *both* training and validation data.
- If we estimate error rate on any part of the training or validation data, we will in general underestimate it.

CROSS VALIDATION

Procedure in detail

We consider possible parameter values $\gamma_1, \ldots, \gamma_m$.

1. For each value γ_j, train a classifier f_{γ_j} on the training set. (That is: γ_j is fixed, and the training algorithm outputs a fitted classifier f for this value of γ_j.)
2. Use the validation set to estimate $R(f_{\gamma_j})$ as the empirical risk

$$\hat{R}(f_\gamma) = \frac{1}{n_v} \sum_{i=1}^{n_v} L(\tilde{y}_i, f_{\gamma_j}(\tilde{\mathbf{x}}_i)) \; .$$

   n_v is the size of the validation set.
3. Select the value γ^* which achieves the smallest estimated error.
4. Re-train the classifier with parameter γ^* on all data except the test set (i.e. on training + validation data).
5. Report error estimate $\hat{R}(f_{\gamma^*})$ computed on the *test* set.

K-FOLD CROSS VALIDATION

Idea

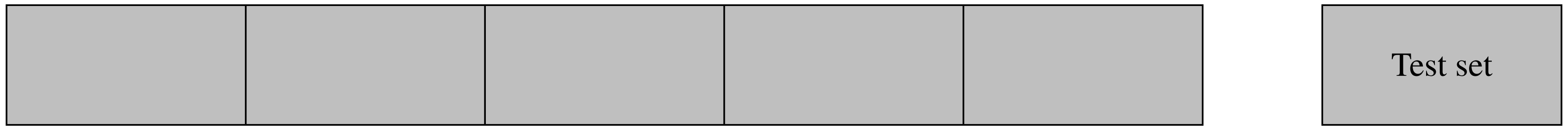
Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy

- Set aside the test set.
- Split the remaining data into K blocks.
- Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

This method is called **K-fold cross validation**.

Example: K=5



K-fold Cross Validation

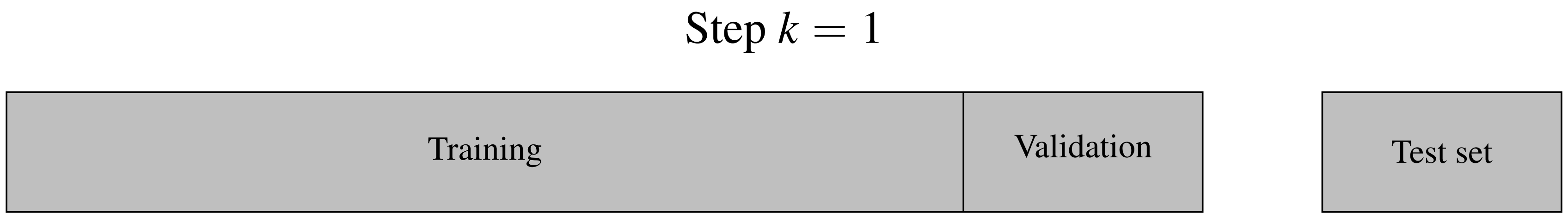
Idea
Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy

- Set aside the test set.
- Split the remaining data into K blocks.
- Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

This method is called **K-fold cross validation**.

Example: K=5



K-FOLD CROSS VALIDATION

Idea

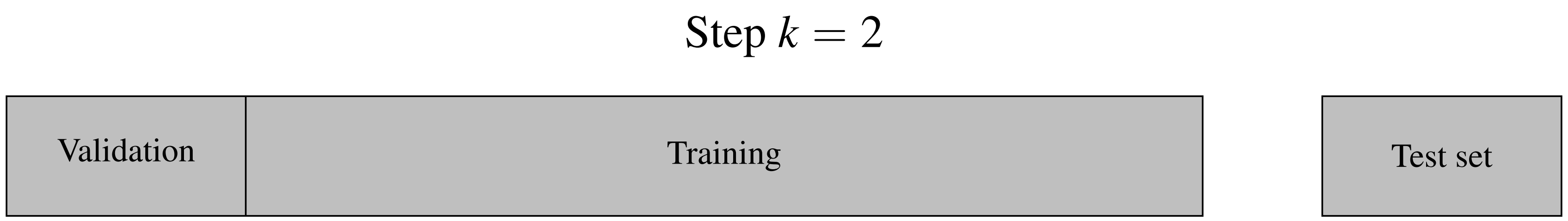
Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy

- Set aside the test set.
- Split the remaining data into K blocks.
- Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

This method is called **K-fold cross validation**.

Example: K=5



K-FOLD CROSS VALIDATION

Idea

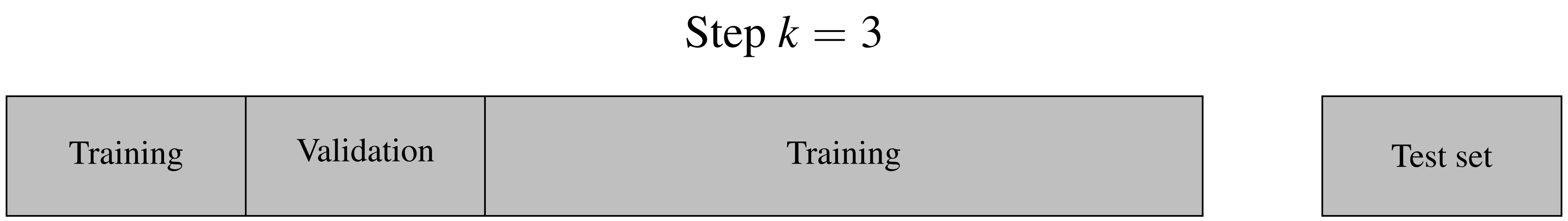
Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy

- Set aside the test set.
- Split the remaining data into K blocks.
- Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

This method is called **K-fold cross validation**.

Example: K=5



K-FOLD CROSS VALIDATION

Idea

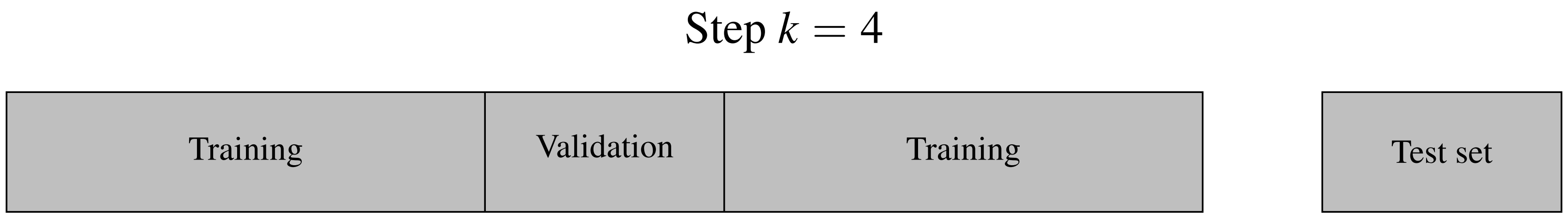
Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy

- Set aside the test set.
- Split the remaining data into K blocks.
- Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

This method is called **K-fold cross validation**.

Example: K=5



K-FOLD CROSS VALIDATION

Idea

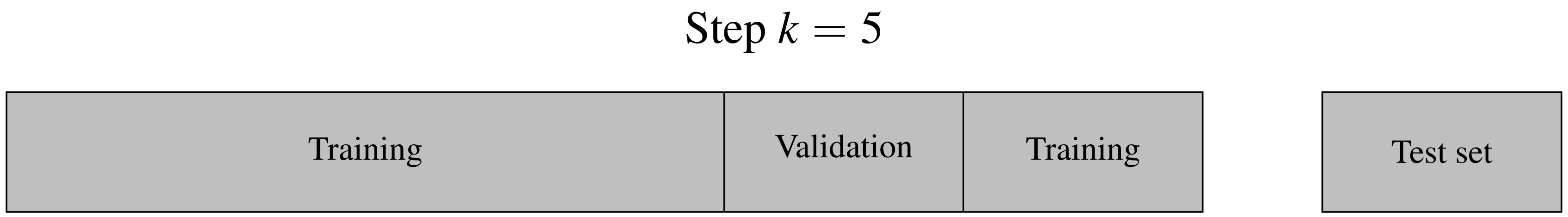
Each of the error estimates computed on validation set is computed from a single example of a trained classifier. Can we improve the estimate?

Strategy

- Set aside the test set.
- Split the remaining data into K blocks.
- Use each block in turn as validation set. Perform cross validation and average the results over all K combinations.

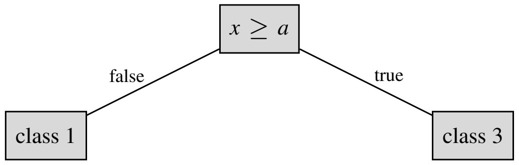
This method is called **K-fold cross validation**.

Example: K=5

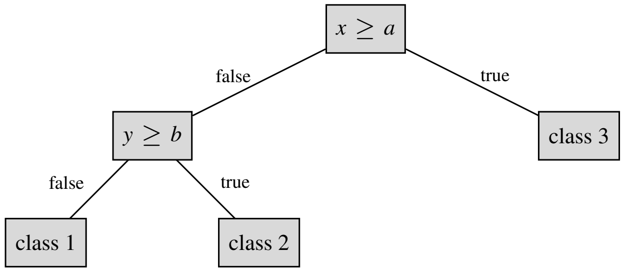


K-FOLD CV FOR A TREE

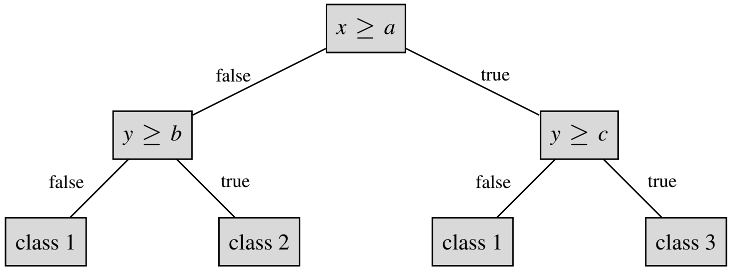
$M = 1$:

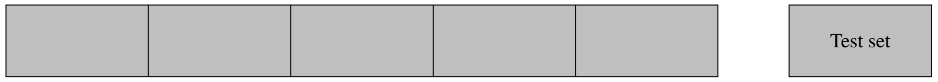


$M = 2$:



$M = 3$:



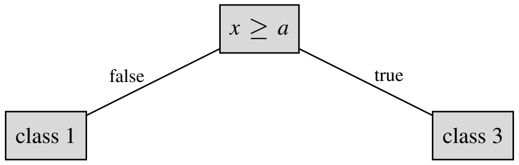


CV procedure (for $K = 5$)

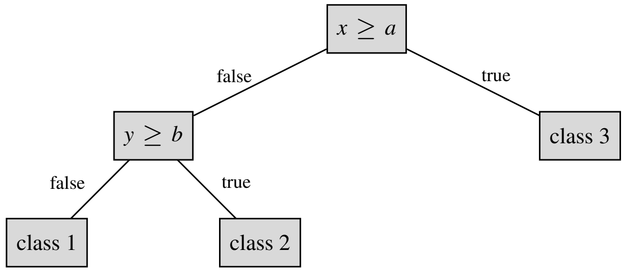
- Split off test data.
- Split remaining data into K equal parts.
- For each $k = 1, \ldots, 5$:
  1. Use kth block as validation set.
  2. For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the remaining blocks.
  3. For each M, estimate the error rate of the trained classifier on the validation block.
- For each M, average the K error rate estimates over all values of k.
- Select the value of M with the smallest average error rate.
- Estimate the error rate for the optimal M on the test set.

K-FOLD CV FOR A TREE

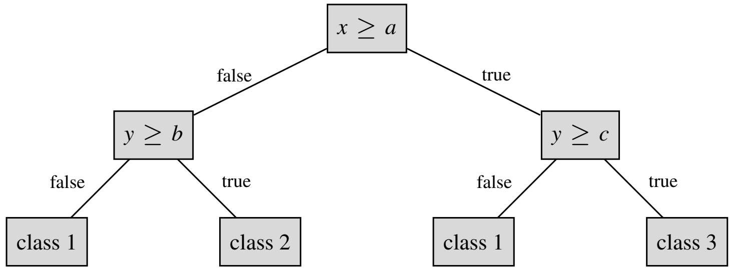
$M = 1$:

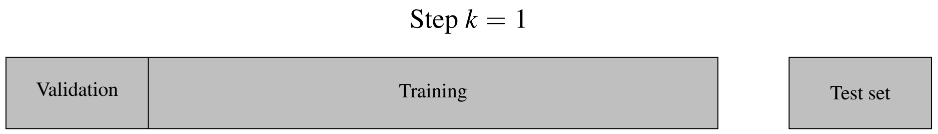


$M = 2$:



$M = 3$:



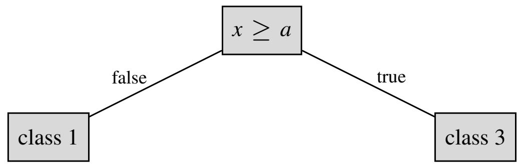


CV procedure (for $K = 5$)

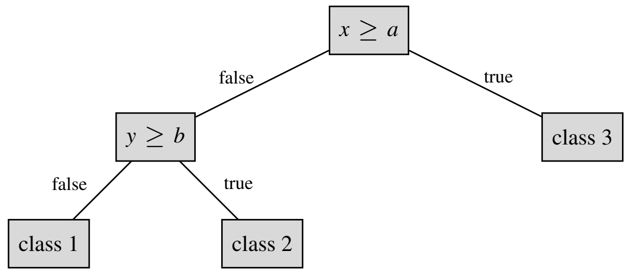
- Split off test data.
- Split remaining data into K equal parts.
- For each $k = 1, \ldots, 5$:
    1. Use kth block as validation set.
    2. For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the remaining blocks.
    3. For each M, estimate the error rate of the trained classifier on the validation block.
- For each M, average the K error rate estimates over all values of k.
- Select the value of M with the smallest average error rate.
- Estimate the error rate for the optimal M on the test set.

K-fold CV for a Tree

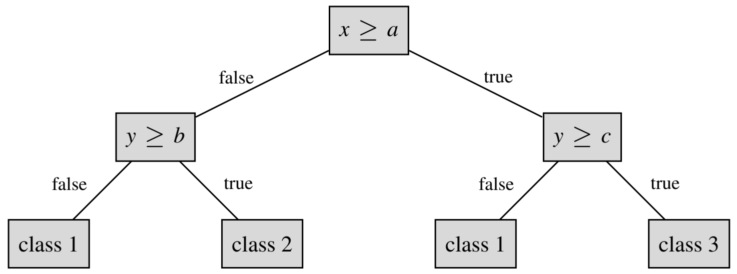
$M = 1$:



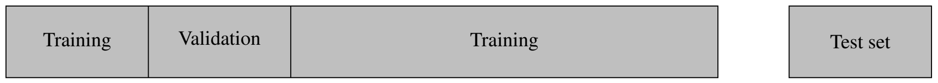
$M = 2$:



$M = 3$:



Step $k = 2$

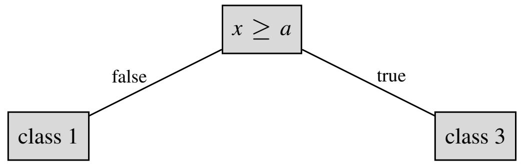


CV procedure (for $K = 5$)

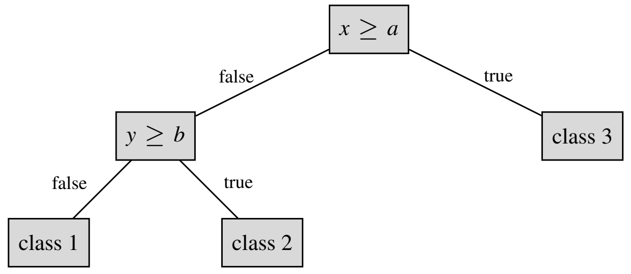
- Split off test data.
- Split remaining data into K equal parts.
- For each $k = 1, \ldots, 5$:
    1. Use kth block as validation set.
    2. For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the remaining blocks.
    3. For each M, estimate the error rate of the trained classifier on the validation block.
- For each M, average the K error rate estimates over all values of k.
- Select the value of M with the smallest average error rate.
- Estimate the error rate for the optimal M on the test set.

K-FOLD CV FOR A TREE

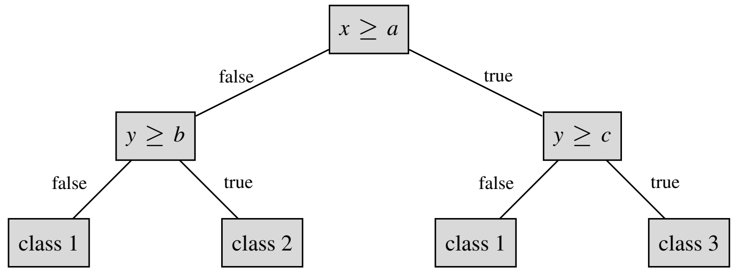
$M = 1$:

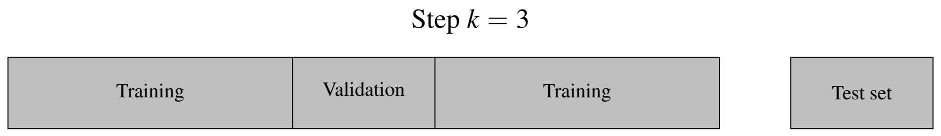


$M = 2$:



$M = 3$:



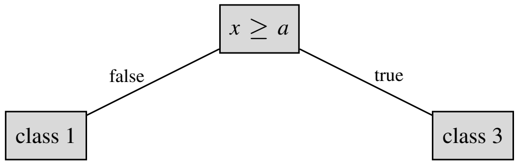


CV procedure (for $K = 5$)

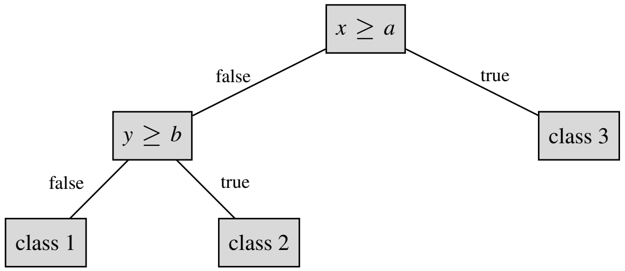
- Split off test data.
- Split remaining data into K equal parts.
- For each $k = 1, \dots, 5$:
  1. Use kth block as validation set.
  2. For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the remaining blocks.
  3. For each M, estimate the error rate of the trained classifier on the validation block.
- For each M, average the K error rate estimates over all values of k.
- Select the value of M with the smallest average error rate.
- Estimate the error rate for the optimal M on the test set.

K-FOLD CV FOR A TREE

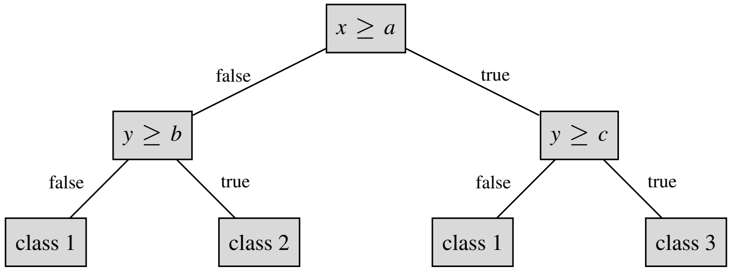
$M = 1$:



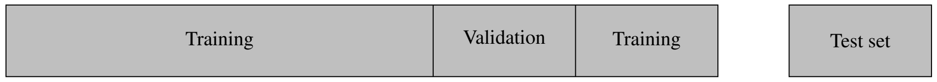
$M = 2$:



$M = 3$:



Step $k = 4$

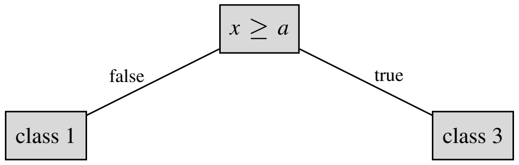


CV procedure (for $K = 5$)

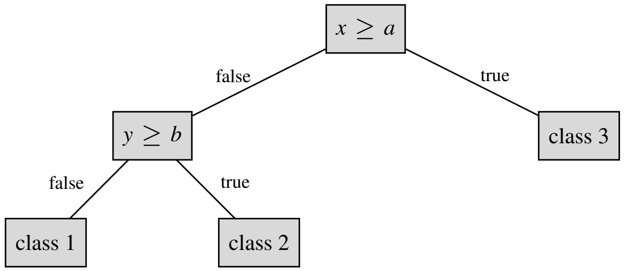
- Split off test data.
- Split remaining data into K equal parts.
- For each $k = 1, \dots, 5$:
  1. Use kth block as validation set.
  2. For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the remaining blocks.
  3. For each M, estimate the error rate of the trained classifier on the validation block.
- For each M, average the K error rate estimates over all values of k.
- Select the value of M with the smallest average error rate.
- Estimate the error rate for the optimal M on the test set.

K-FOLD CV FOR A TREE

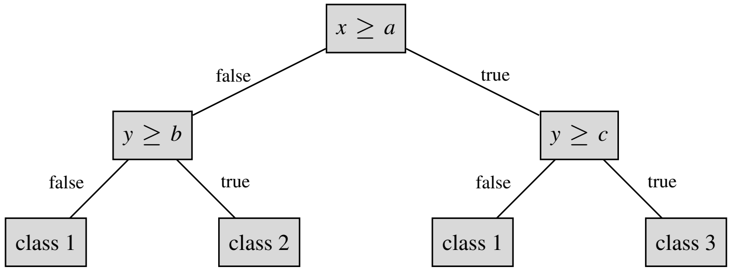
$M = 1$:



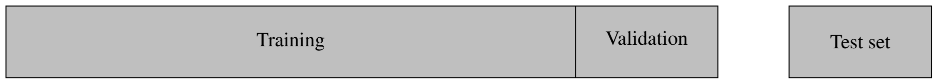
$M = 2$:



$M = 3$:



Step $k = 5$



CV procedure (for $K = 5$)

- Split off test data.
- Split remaining data into K equal parts.
- For each $k = 1, \dots, 5$:
  1. Use kth block as validation set.
  2. For each M in $\{1, 2, 3\}$: Train a tree classifier with M splits on the remaining blocks.
  3. For each M, estimate the error rate of the trained classifier on the validation block.
- For each M, average the K error rate estimates over all values of k.
- Select the value of M with the smallest average error rate.
- Estimate the error rate for the optimal M on the test set.

K-FOLD CROSS VALIDATION: IN DETAIL

Risk estimation

To estimate the risk of a classifier $f(\,.\,,\gamma_j)$:

1. Split data into K equally sized blocks.
2. Train an instance $f_{\gamma_j,k}$ of the classifier, using all blocks except block k as training data.
3. Compute the cross validation estimate

$$\hat{R}_{\text{CV}}(f_{\gamma_j}) := \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|\text{block } k|}\sum_{(\tilde{\mathbf{x}},\tilde{y})\in\text{ block } k} L\big(\tilde{y}, f_{\gamma_j,k}(\tilde{\mathbf{x}})\big)$$

Repeat this for all parameter values $\gamma_1,\ldots,\gamma_m$.

Selecting a model

- Choose the parameter value γ^* for which estimated risk is minimal.

Model assessment

- Report risk estimate for f_{γ^*} computed on *test* data.

How to Choose K?

Extremal cases

- $K = n$, called **leave one out cross validation** (loocv)
- $K = 2$

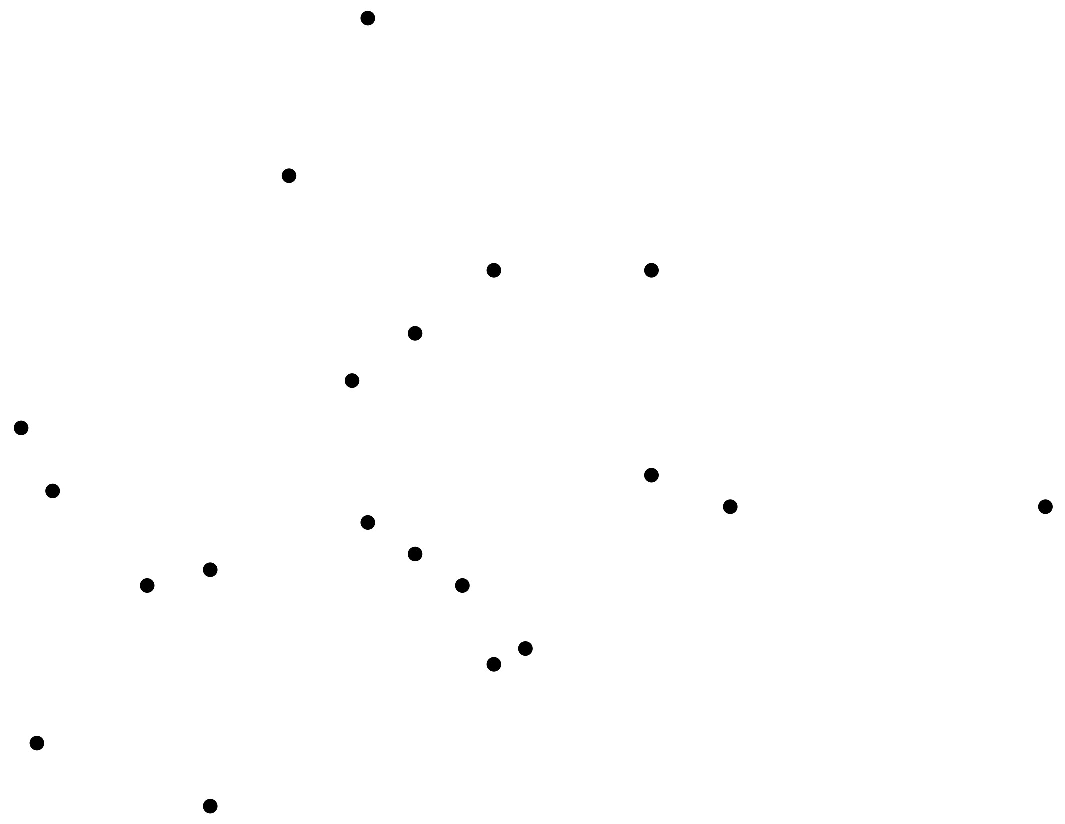
An often-cited problem with loocv is that we have to train many (= n) classifiers, but there is also a deeper problem.

Argument 1: K should be small, e.g. $K = 2$

- Unless we have a lot of data, variance between two distinct training sets may be considerable.
- **Important concept:** By removing substantial parts of the sample in turn and at random, we can simulate this variance.
- By removing a single point (loocv), we cannot make this variance visible.

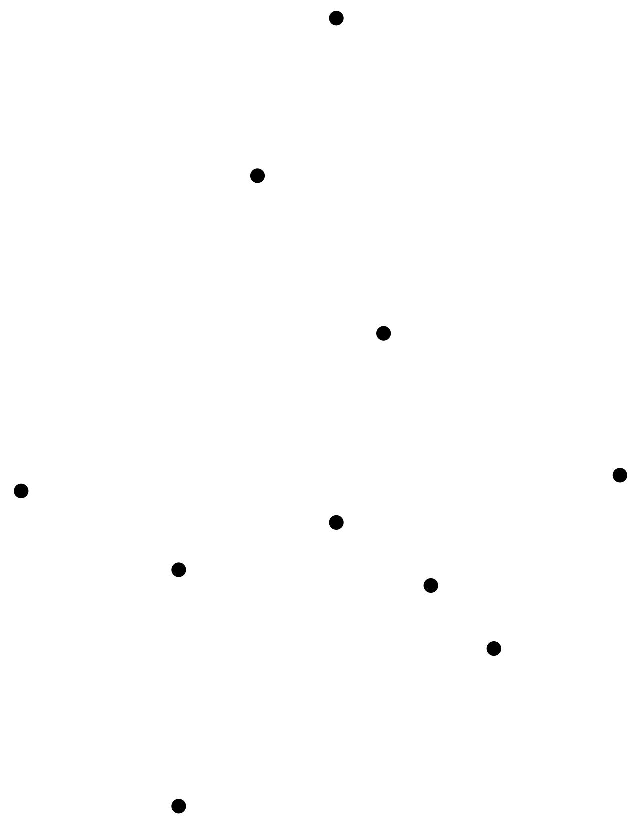
ILLUSTRATION

$K = 2, n = 20$



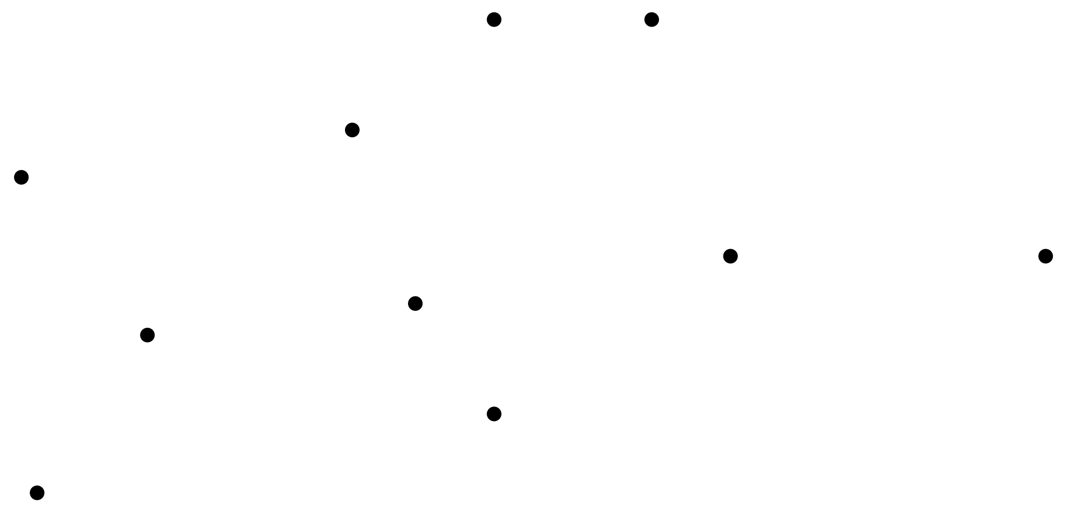
ILLUSTRATION

$K = 2, n = 20$



ILLUSTRATION

$K = 2, n = 20$



How to Choose K?

Argument 2: K should be large, e.g. $K = n$

- Classifiers generally perform better when trained on larger data sets.
- A small K means we substantially reduce the amount of training data used to train each f_k, so we may end up with weaker classifiers.
- This way, we will systematically overestimate the risk.

Common recommendation: $K = 5$ to $K = 10$

Intuition:

- $K = 10$ means number of samples removed from training is one order of magnitude below training sample size.
- This should not weaken the classifier considerably, but should be large enough to make measure variance effects.

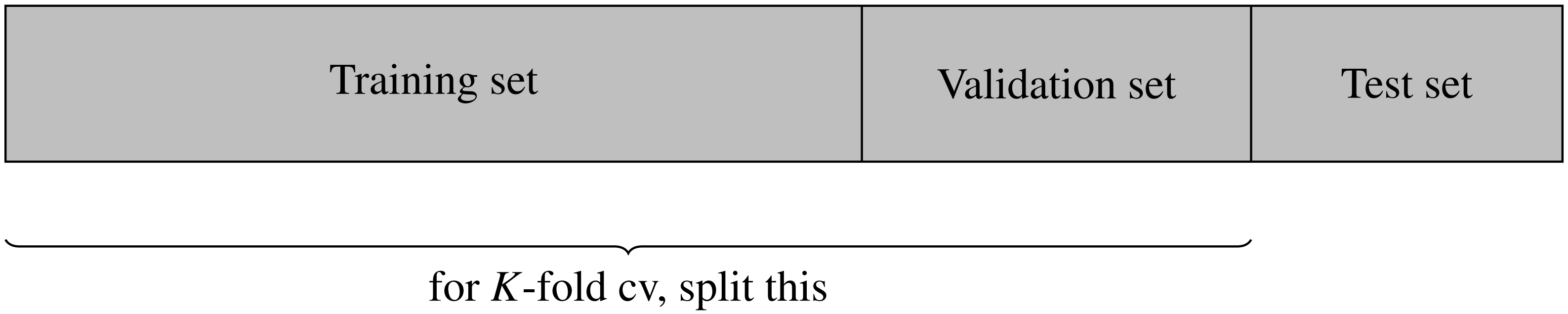
Summary: Cross Validation

Purpose

Estimates the risk $R(f) = \mathbb{E}[L(y, f(x))]$ of a classifier (or regression function) from data.

Application to parameter tuning

- Compute one cross validation estimate of $R(f)$ for each parameter value.
- Example above is margin parameter γ, but can be used for any parameter of a supervised learning algorithm.
- Note: Cross validation procedure does not involve the test data.


Training set
Validation set
Test set
for K-fold cv, split this