Machine learning methods applied to the analysis of central exclusive production events in ALICE

Sebastian Ratzenböck[1]

[1]Stefan Meyer Institut
Österreichische Akademie der Wissenschaften

26. April 2018

ÖAW ÖSTERREICHISCHE AKADEMIE DER WISSENSCHAFTEN

Outline

ML: an overview

In general ML represents a contrast to a *rule based systems*

Rule-based system

System that uses rules to make deductions or choices

- Domain-specific expert system
- Knowledge base: facts & rules (if $\rightarrow$ then statement)
- Rules manually specified (by expert) $\rightarrow$ expensive, incomplete

ML: an overview

In general ML represents a contrast to a *rule based systems*

Rule-based system

System that uses rules to make deductions or choices

- Domain-specific expert system
- Knowledge base: facts & rules (if $\rightarrow$ then statement)
- Rules manually specified (by expert) $\rightarrow$ expensive, incomplete

ML: an overview

In general ML represents a contrast to a *rule based systems*

Rule-based system

System that uses rules to make deductions or choices

- Domain-specific expert system
- Knowledge base: facts & rules (if $\rightarrow$ then statement)
- Rules manually specified (by expert) $\rightarrow$ expensive, incomplete

ML: an overview

In general ML represents a contrast to a *rule based systems*

Machine learning

- Alorithms that learn from *data* & make predictions on *data*
- Automatic methods $\rightarrow$ no human needed
- Human work required for defining problem & assessing the data

ML: an overview

In general ML represents a contrast to a *rule based systems*

Machine learning

- Alorithms that learn from *data* & make predictions on *data*
- Automatic methods → no human needed
- Human work required for defining problem & assessing the data

ML: an overview

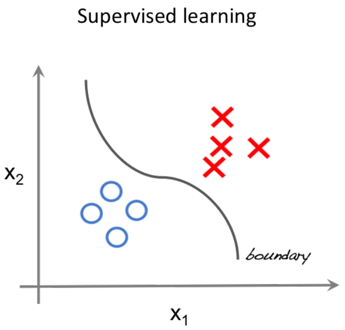
In general ML represents a contrast to a *rule based systems*

Machine learning

- Alorithms that learn from *data* & make predictions on *data*
- Automatic methods → no human needed
- Human work required for defining problem & assessing the data

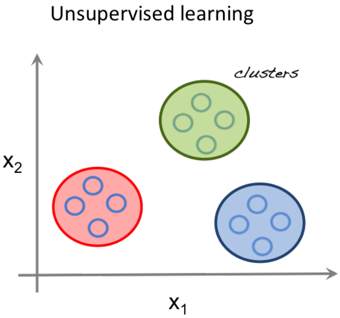
Types of ML

- Supervised
  - Classification
  - Regression
- Unsupervised



Types of ML

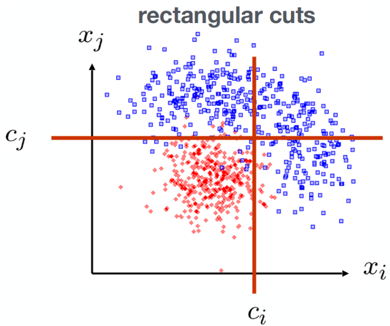
- Supervised
  - Classification
  - Regression
- Unsupervised



Rectangular cuts

Standard cut in one variable

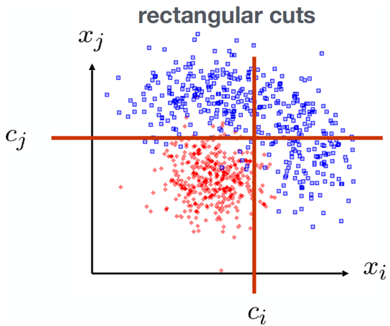
- Cuts only in lower-dimensional subspaces
- Ignores possible dependencies between the input variables
- Signal might behave like BG in several observables
  $\rightarrow$ misclassification



Rectangular cuts

Standard cut in one variable

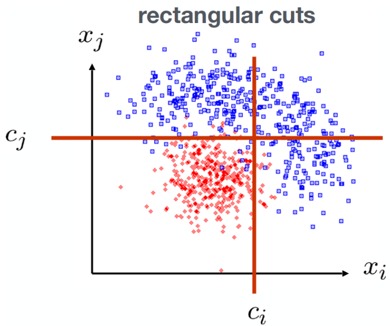
- Cuts only in lower-dimensional subspaces
- Ignores possible dependencies between the input variables
- Signal might behave like BG in several observables
  $\rightarrow$ misclassification



Rectangular cuts

Standard cut in one variable

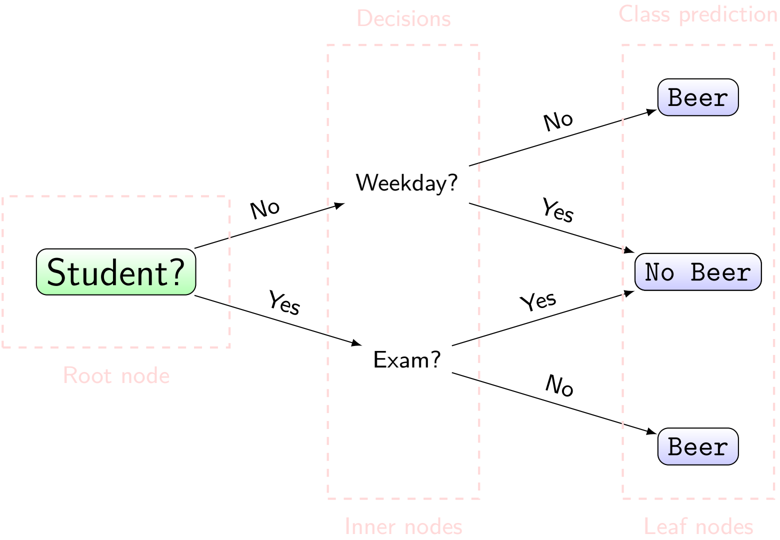
- Cuts only in lower-dimensional subspaces
- Ignores possible dependencies between the input variables
- Signal might behave like BG in several observables
  $\rightarrow$ misclassification



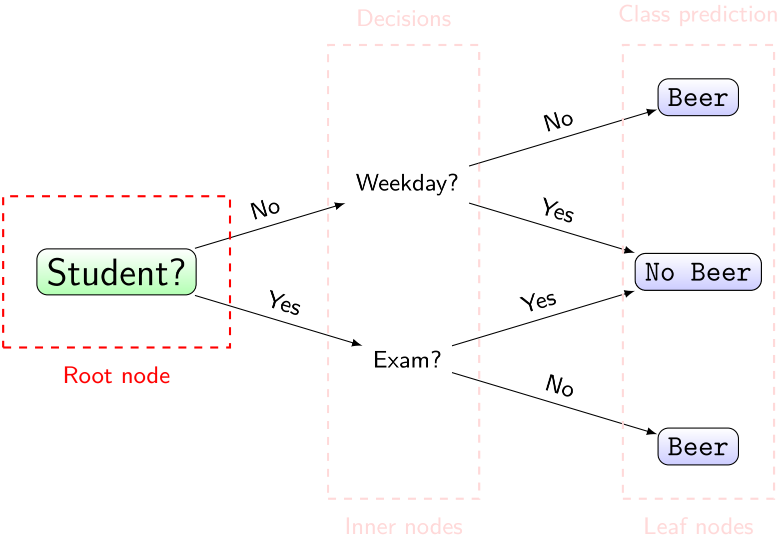
Rectangular cuts with *decision trees*

- Tree-like graph $\rightarrow$ flowchart
- Easy to understand
- Either be manually modelled by experts or learned from training data

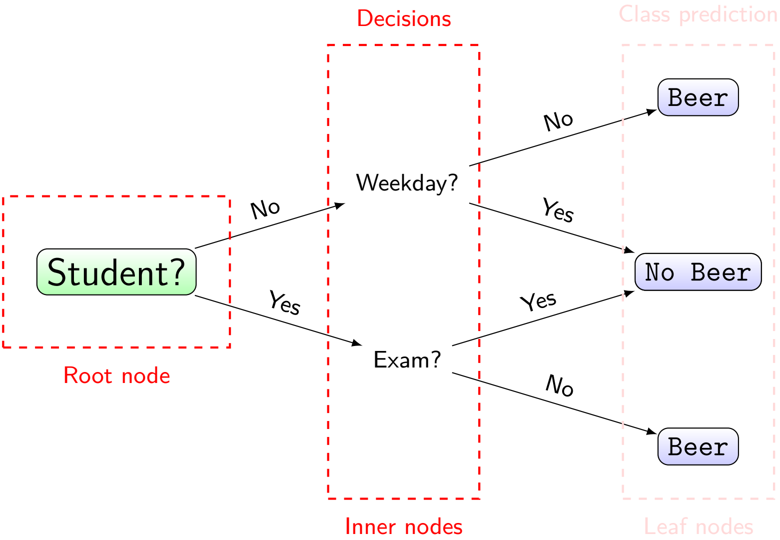
Rectangular cuts with *decision trees*



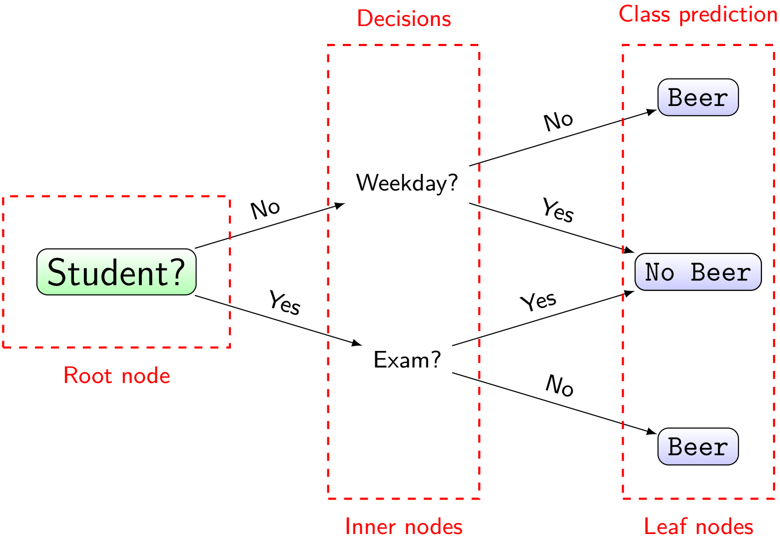
Rectangular cuts with *decision trees*



Rectangular cuts with *decision trees*



Rectangular cuts with *decision trees*



Decision tree learning

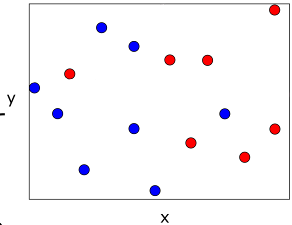
Training

Recursively split feature space into sub-spaces at each step
$\rightarrow$ Measures to evaluate split

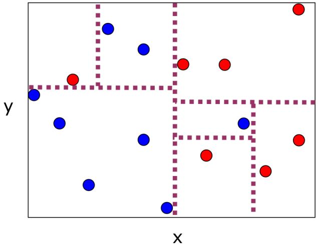
- Error rate
- Information gain
- Gini index

Decision tree learning

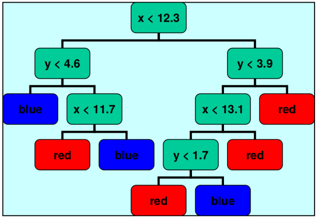
Feature space



Classification



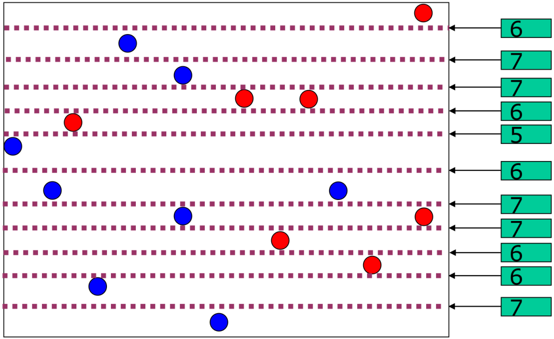
Decision tree



Decision tree learning

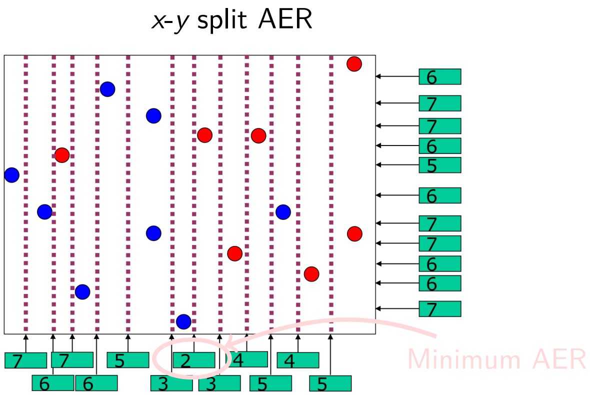
1) We compute a measure for *each possible split* in each feature
   $\rightarrow$ here **absolute error rate** (AER)

y split AER



Decision tree learning

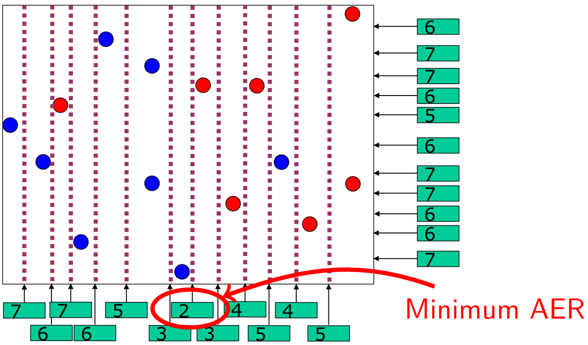
1) We compute a measure for *each possible split* in each feature
$\rightarrow$ here **absolute error rate** (AER)



Decision tree learning

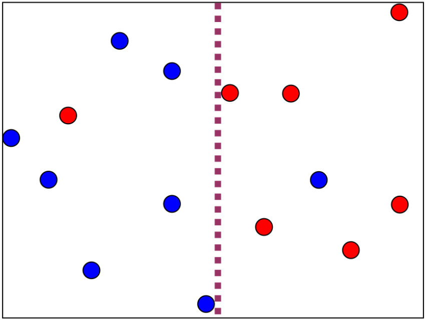
1) We compute a measure for *each possible split* in each feature
   → here **absolute error rate** (AER)

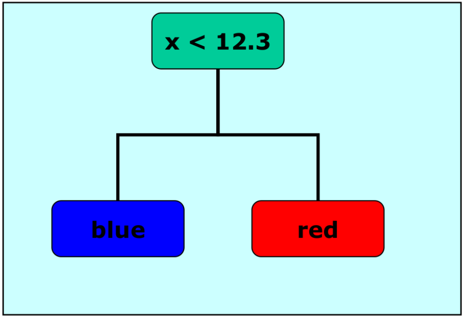
x-*y* split AER



Decision tree learning

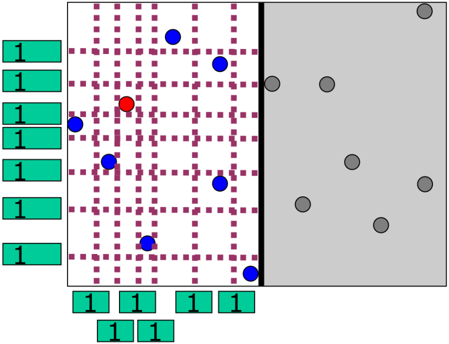
1) We compute a measure for *each possible split* in each feature
   $\rightarrow$ here **absolute error rate** (AER)
2) Recursively repead step (1) for each subspace until AER $\rightarrow$ 0

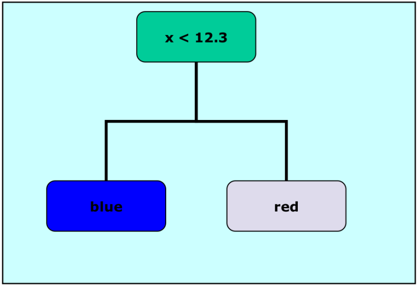




Decision tree learning

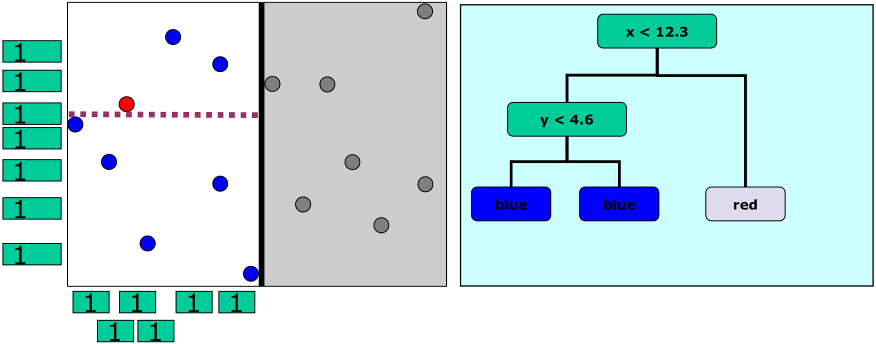
1) We compute a measure for *each possible split* in each feature
   $\rightarrow$ here **absolute error rate** (AER)
2) Recursively repead step (1) for each subspace until AER $\rightarrow$ 0





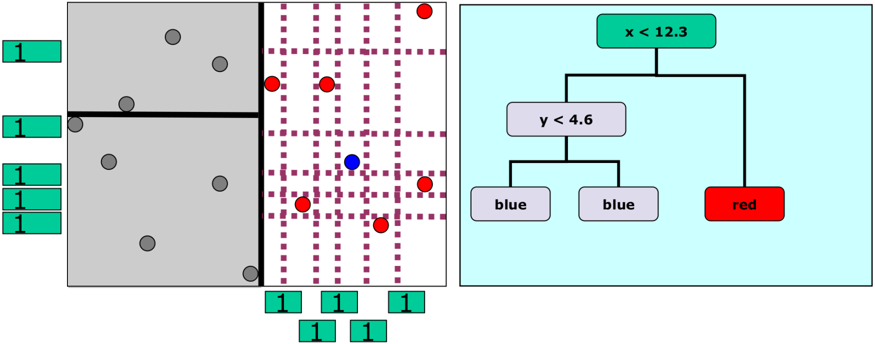
Decision tree learning

1) We compute a measure for *each possible split* in each feature
   $\rightarrow$ here **absolute error rate** (AER)
2) Recursively repead step (1) for each subspace until AER $\rightarrow$ 0



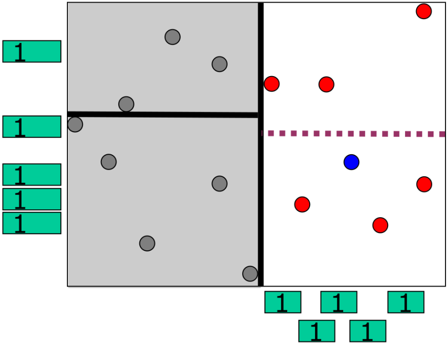
Decision tree learning

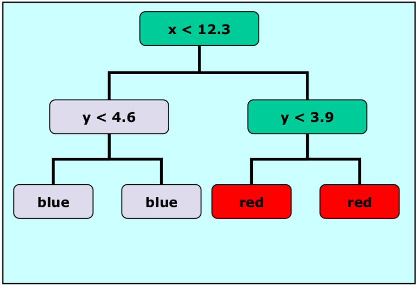
1) We compute a measure for *each possible split* in each feature
   $\rightarrow$ here **absolute error rate** (AER)
2) Recursively repead step (1) for each subspace until AER $\rightarrow 0$



Decision tree learning

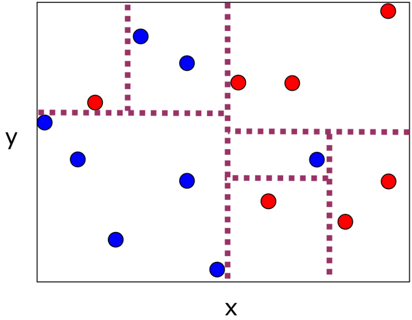
1) We compute a measure for *each possible split* in each feature
   → here **absolute error rate** (AER)
2) Recursively repead step (1) for each subspace until AER → 0

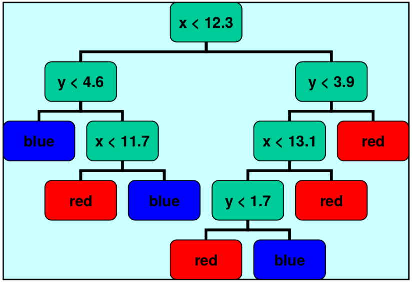




Decision tree learning

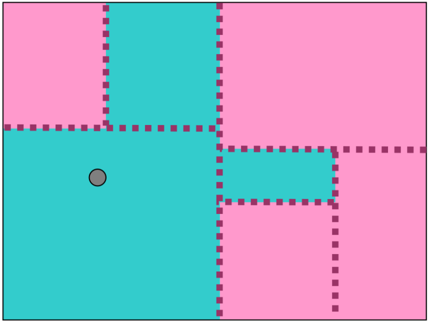
1) We compute a measure for *each possible split* in each feature
   $\rightarrow$ here **absolute error rate** (AER)
2) Recursively repead step (1) for each subspace until AER $\rightarrow 0$

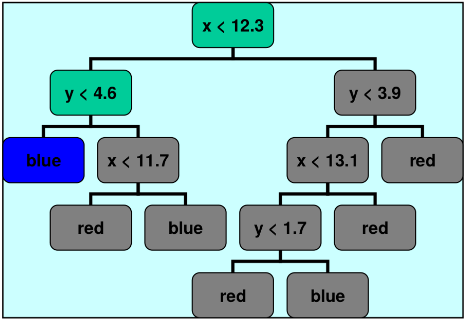




Decision tree classification

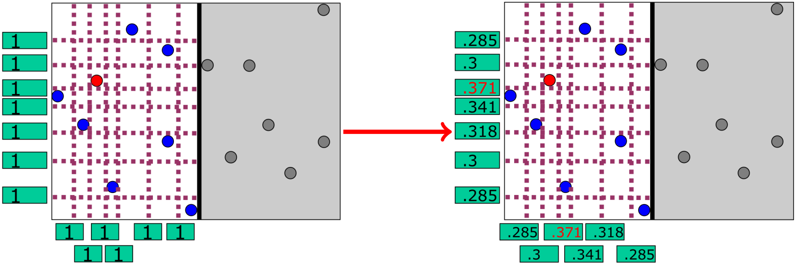
3) Classification





Decision tree improvements I

- Use more sophisticated split measures
  - *Information gain* $\leftrightarrow$ (im-)purity of splitted sub-sets
  - Gini index
- Pruning



Absolute error rate

Information gain

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features

  → large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- → Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    → large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- → Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    → large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- → Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    $\rightarrow$ large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- $\rightarrow$ Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    → large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- → Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    $\rightarrow$ large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- $\rightarrow$ Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    $\rightarrow$ large number of rather shallow trees
- Classify data by majority voting of individual trees

Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- $\rightarrow$ Subsequent predictors learn from the mistakes of the previous predictors

Decision tree improvements II

Random forest

- Ensemble of DTs
- For each tree use:
  - Random sub-sample (=*bootstrapping*)
  - Random number of the original features
    $\rightarrow$ large number of rather shallow trees
- Classify data by majority voting of individual trees

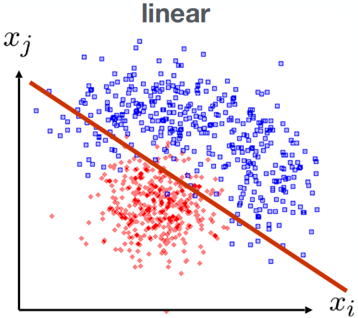
Boosted DT

- Sequential ensemble of evolving DTs
- Output of each tree is given weight relative to accuracy
- $\rightarrow$ Subsequent predictors learn from the mistakes of the previous predictors

Linear cuts

More flexible than rectangular cut

- Simple white box methods
- Can become very powerful by using *kernel trick*



Linear cuts

More flexible than rectangular cut

- Simple white box methods
- Can become very powerful by using *kernel trick*