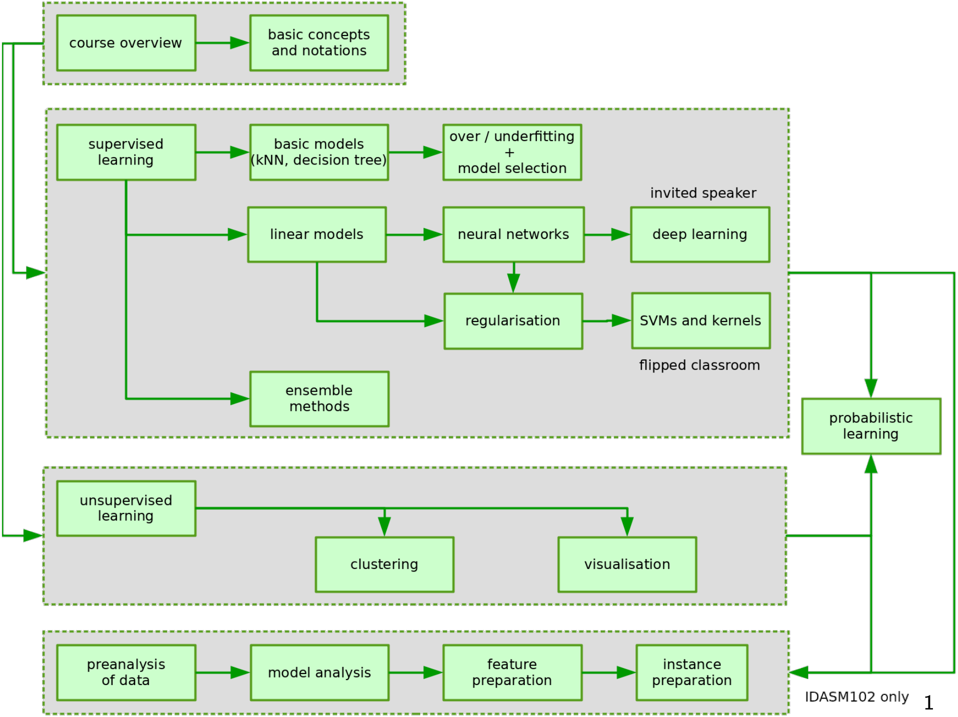
Machine Learning: Lesson 11

Ensemble Methods: from Single Models to Ensemble of Models

Benoît Frénay - Faculty of Computer Science

UNIVERSITÉ DE NAMUR

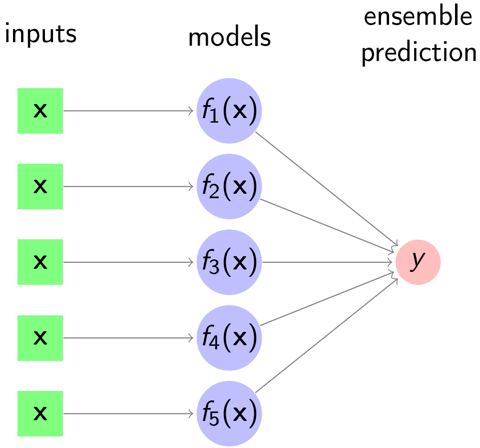

course overview
basic concepts and notations
supervised learning
basic models (kNN, decision tree)
over / underfitting + model selection
invited speaker
linear models
neural networks
deep learning
regularisation
SVMs and kernels
flipped classroom
ensemble methods
probabilistic learning
unsupervised learning
clustering
visualisation
preanalysis of data
model analysis
feature preparation
instance preparation
IDASM102 only

Outline of this Lesson

- ensemble learning
- bootstrap aggregating
- random forests
- adaptive boosting

Ensemble Learning

Combining Machine Learning Models



Combining Machine Learning Models

Motivation

given a set of machine learning models for classification

- each model makes errors on a specific set of instances
- models of different types have different model abilities

what if we find a way to combine "weak models" into a stronger model ?

No free lunch theorems

NFL theorems developed by Wolpert and Macready in late 90's

- "*if algorithm A outperforms algorithm B on some cost functions, then loosely speaking there must exist exactly as many other functions where B outperforms A.*" $\Rightarrow$ there is no "best ever algorithm"
- "[...] *any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class.*"

Combining Machine Learning Models

Motivation

given a set of machine learning models for classification

- each model makes errors on a specific set of instances
- models of different types have different model abilities

what if we find a way to combine "weak models" into a stronger model ?

No free lunch theorems

NFL theorems developed by Wolpert and Macready in late 90's

- "*if algorithm A outperforms algorithm B on some cost functions, then loosely speaking there must exist exactly as many other functions where B outperforms A.*" $\Rightarrow$ there is no "best ever algorithm"
- "[...] *any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class.*"

Pros and Cons of Ensemble Methods

- ✓ ensemble methods are powerful (e.g. random forests)
- × ensemble methods can overfit (e.g. Adaboost)
- × ensemble methods are also affected by NFL theorems
- × in some cases, ensemble may be meaningless (e.g. linear combination of linear models, ensemble of neural networks with linear output, etc.)

Model variability

ensembling models only makes sense if ensembled models are different

- simple strategy: use different training sets to achieve variability
- each model can individually be very weak (low complexity, high error)

synonyms: classifier fusion, multiple classifier system, committees...

Combining Machine Learning Models

Pros and Cons of Ensemble Methods

- ✓ ensemble methods are powerful (e.g. random forests)
- × ensemble methods can overfit (e.g. Adaboost)
- × ensemble methods are also affected by NFL theorems
- × in some cases, ensemble may be meaningless (e.g. linear combination of linear models, ensemble of neural networks with linear output, etc.)

Model variability

ensembling models only makes sense if ensembled models are different

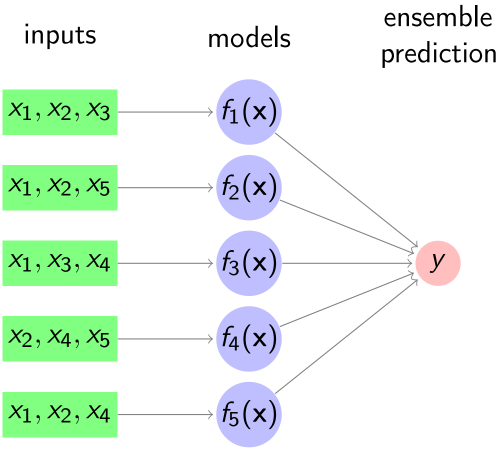
- simple strategy: use different training sets to achieve variability
- each model can individually be very weak (low complexity, high error)

synonyms: classifier fusion, multiple classifier system, committees...

Achieving Model Variability

Feature space

each model uses a subset of features $\Rightarrow$ works on an "aspect" of data



Achieving Model Variability

Classifier architecture/optimisation

- use different types of models (linear classifiers, neural networks, decision trees, k-nearest neighbours, decision stumps, SVMs. . .)
- use different initialisations for non-convex objective functions

Boostrap aggregating (= bagging)

repeated resampling of the training instances

- learn from $\neq$ datasets (yet similar to original)
- ex.: random forests (+ other tweaks)

Boosting (uses very weak base classifiers)

repeated reweighting of the training instances

- focus on $\neq$ parts of the dataset
- ex.: AdaBoost (= adaptive boosting)

Achieving Model Variability

Classifier architecture/optimisation

- use different types of models (linear classifiers, neural networks, decision trees, k-nearest neighbours, decision stumps, SVMs…)
- use different initialisations for non-convex objective functions

Boostrap aggregating (= bagging)

repeated resampling of the training instances

- learn from $\neq$ datasets (yet similar to original)
- ex.: random forests (+ other tweaks)

Boosting (uses very weak base classifiers)

repeated reweighting of the training instances

- focus on $\neq$ parts of the dataset
- ex.: AdaBoost (= adaptive boosting)

Achieving Model Variability

Classifier architecture/optimisation

- use different types of models (linear classifiers, neural networks, decision trees, k-nearest neighbours, decision stumps, SVMs...)
- use different initialisations for non-convex objective functions

Boostrap aggregating (= bagging)

repeated resampling of the training instances

- learn from $\neq$ datasets (yet similar to original)
- ex.: random forests (+ other tweaks)

Boosting (uses very weak base classifiers)

repeated reweighting of the training instances

- focus on $\neq$ parts of the dataset
- ex.: AdaBoost (= adaptive boosting)

Boostrap Aggregating

Learning from Bootstrap Samples

Bootstrap sample

question: how can we obtain $B \gg 1$ similar, yet different training sets ?

- draw n times with replacement from original training set of size n
- instances may appear several times (or not) in the bootstrap sample

goal: obtain an *unstable* procedure to obtain variable training sets

- probability $1 - (1 - 1/n)^n \approx 0.63$ for each instance to be selected
- on average 63% of unique patterns in each bootstrap sample (large n)
- unselected 36% of instances can be used to perform testing

Combining classifiers

- a classifier is trained for each bootstrap sample
- prediction = majority rule applied to the B predictions

Learning from Bootstrap Samples

Bootstrap sample

question: how can we obtain $B \gg 1$ similar, yet different training sets ?

- draw n times with replacement from original training set of size n
- instances may appear several times (or not) in the bootstrap sample

goal: obtain an *unstable* procedure to obtain variable training sets

- probability $1 - (1 - 1/n)^n \approx 0.63$ for each instance to be selected
- on average 63% of unique patterns in each bootstrap sample (large n)
- unselected 36% of instances can be used to perform testing

Combining classifiers

- a classifier is trained for each bootstrap sample
- prediction = majority rule applied to the B predictions

Learning from Bootstrap Samples

Bootstrap sample

question: how can we obtain $B \gg 1$ similar, yet different training sets ?

- draw n times with replacement from original training set of size n
- instances may appear several times (or not) in the bootstrap sample

goal: obtain an *unstable* procedure to obtain variable training sets

- probability $1 - (1 - 1/n)^n \approx 0.63$ for each instance to be selected
- on average 63% of unique patterns in each bootstrap sample (large n)
- unselected 36% of instances can be used to perform testing

Combining classifiers

- a classifier is trained for each bootstrap sample
- prediction = majority rule applied to the B predictions

Learning from Bootstrap Samples

Bootstrap sample

question: how can we obtain $B \gg 1$ similar, yet different training sets ?

- draw n times with replacement from original training set of size n
- instances may appear several times (or not) in the bootstrap sample

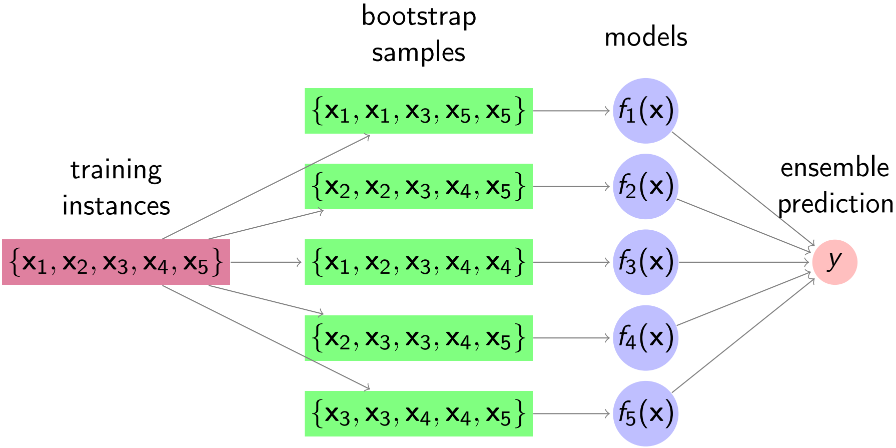
goal: obtain an *unstable* procedure to obtain variable training sets

- probability $1 - (1 - 1/n)^n \approx 0.63$ for each instance to be selected
- on average 63% of unique patterns in each bootstrap sample (large n)
- unselected 36% of instances can be used to perform testing

Combining classifiers

- a classifier is trained for each bootstrap sample
- prediction = majority rule applied to the B predictions

Learning from Bootstrap Samples



Random Forests

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About...a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About…a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

randomisation of the feature choice leads to tremendous diversity in trees

Advantages

- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

If a Decision Tree is Good, What About... a Forest ?

Random forest = bagging + random features

use many decision trees with high variability

- create a bootstrap sample for training (keep unused 36% for test)
- use the bootstrap sample to learn a decision tree (e.g. with ID3)
- at each decision node, pick best feature in a random subset of features

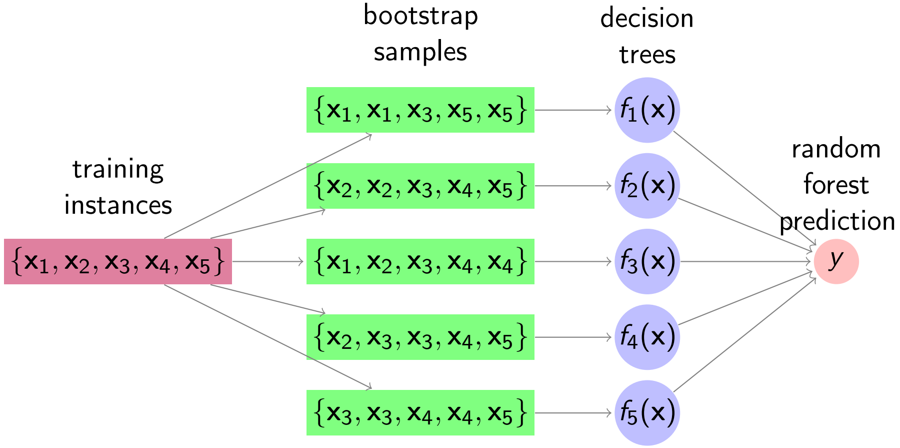
randomisation of the feature choice leads to tremendous diversity in trees

Advantages

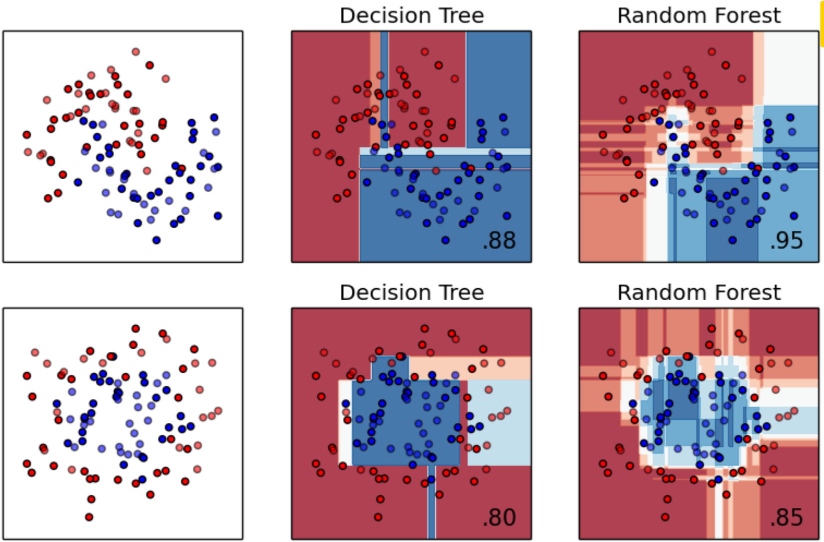
- perform very well on large datasets (ID3 with subset of features)
- straightforward to parallelise (useful if you have many CPUs)
- can take advantages of advanced techniques like Hoeffding trees

Learning Random Forest

the random subset of features is different at each node considered by ID3



Single Decision Tree vs. Random Forest



http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Adaptive Boosting

Intuition Behind Adaptive Boosting

AdaBoost = iterative construction of an ensemble of models

- (very) weak base models (even only slighlty better than random)
- at iteration i, model f_i is added to the set of models $\{f_1, \ldots, f_{i-1}\}$
- the weight α_i of each base model f_i is determined once and for all

at each iteration, f_i is learnt on a weighted version of the dataset

- weights of instances previously misclassified increase afterwards
- opposite for weights of previously correctly classified instances

successive classifiers are forced to focus on misclassified instances !

Intuition Behind Adaptive Boosting

AdaBoost = iterative construction of an ensemble of models

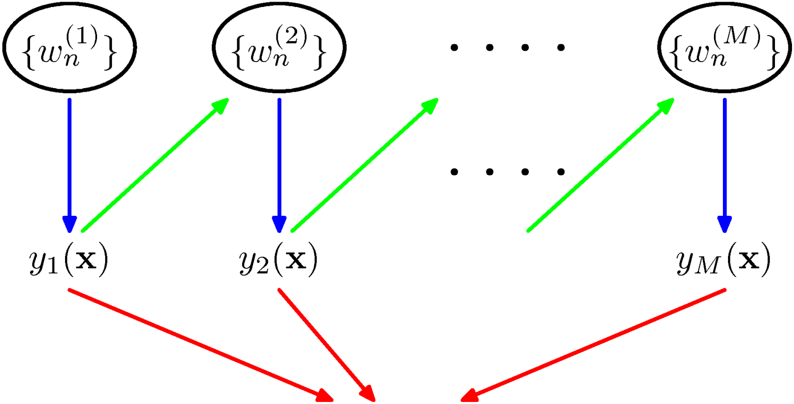
- (very) weak base models (even only slighlty better than random)
- at iteration i, model f_i is added to the set of models $\{f_1, \ldots, f_{i-1}\}$
- the weight α_i of each base model f_i is determined once and for all

at each iteration, f_i is learnt on a weighted version of the dataset

- weights of instances previously misclassified increase afterwards
- opposite for weights of previously correctly classified instances

successive classifiers are forced to focus on misclassified instances !

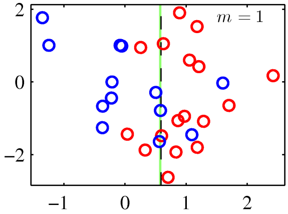
Intuition Behind Adaptive Boosting



$$Y_M(\mathbf{x}) = \operatorname{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

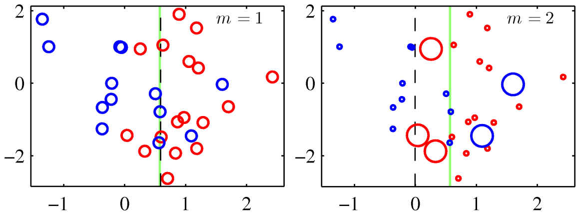
from Pattern Recognition and Machine Learning by Christopher M. Bishop

Intuition Behind Adaptive Boosting



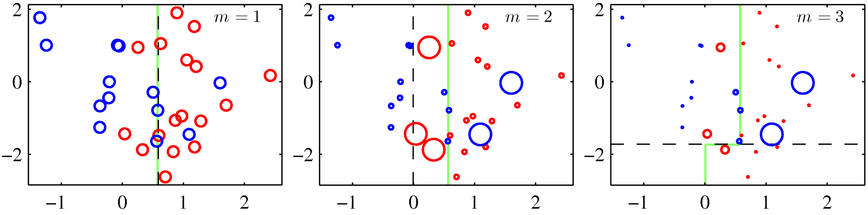
from Pattern Recognition and Machine Learning by Christopher M. Bishop

Intuition Behind Adaptive Boosting



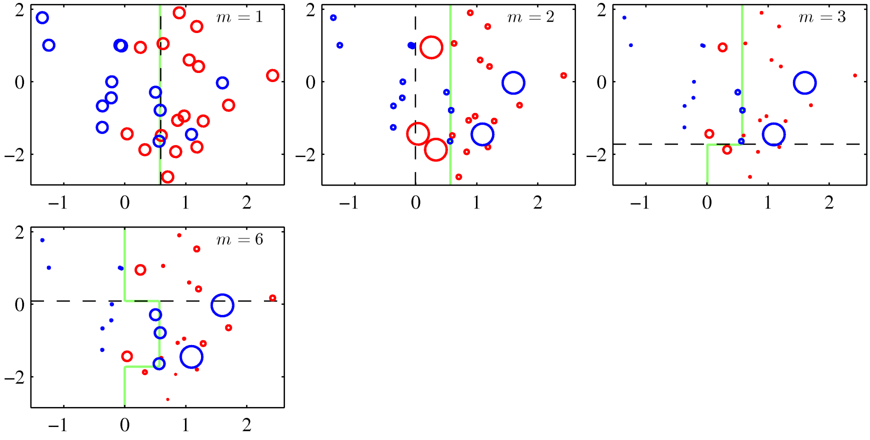
from Pattern Recognition and Machine Learning by Christopher M. Bishop

Intuition Behind Adaptive Boosting



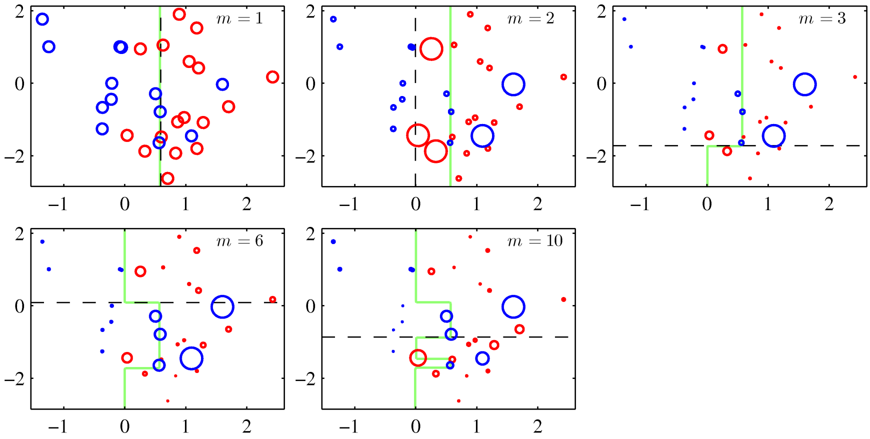
from Pattern Recognition and Machine Learning by Christopher M. Bishop

Intuition Behind Adaptive Boosting



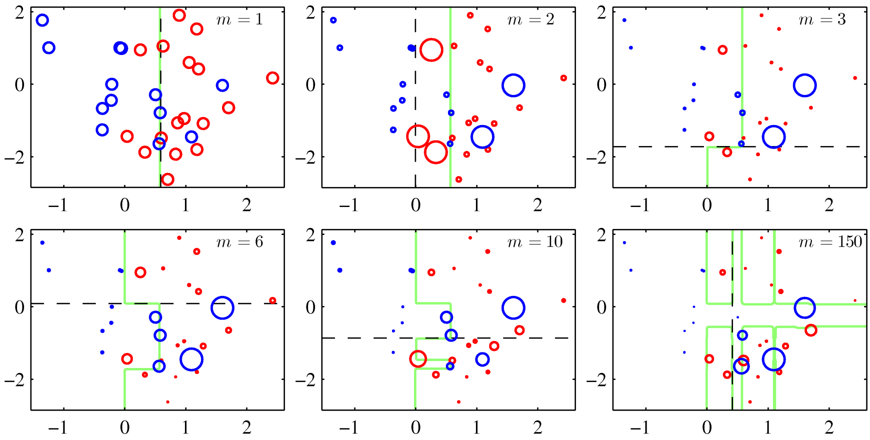
from Pattern Recognition and Machine Learning by Christopher M. Bishop

Intuition Behind Adaptive Boosting



from Pattern Recognition and Machine Learning by Christopher M. Bishop

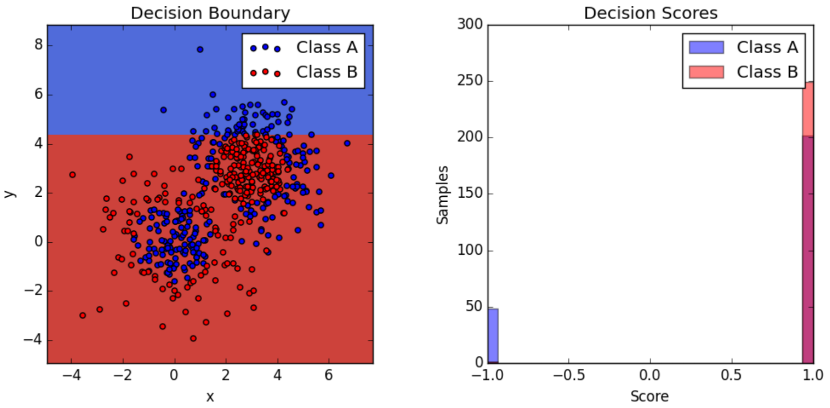
Intuition Behind Adaptive Boosting



from Pattern Recognition and Machine Learning by Christopher M. Bishop

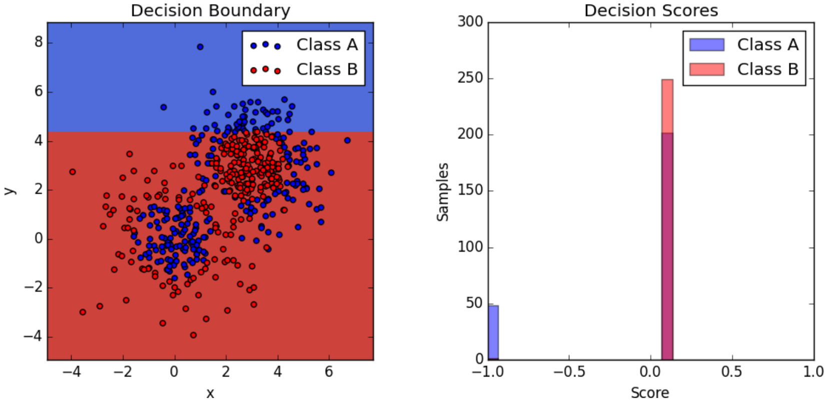
Adaptive Boosting of Decision stumps

single decision stump



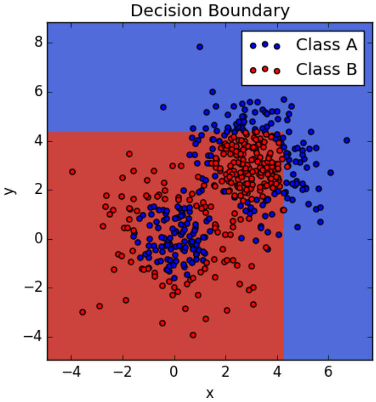
inspired from `http://scikit-learn.org`

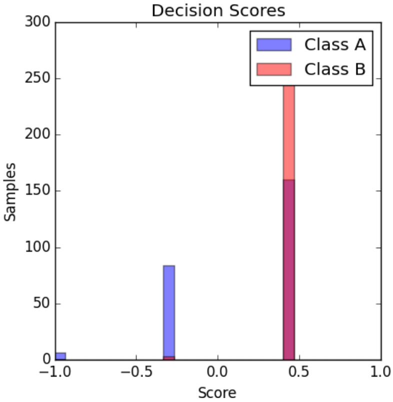
ensemble of 2 decision stumps



inspired from `http://scikit-learn.org`

ensemble of 3 decision stumps

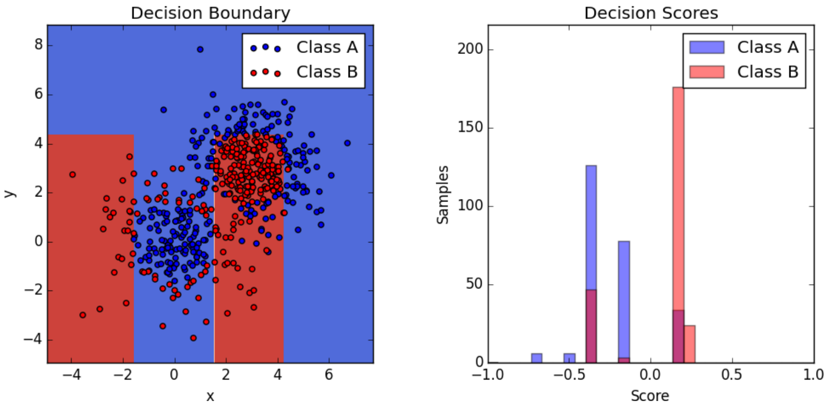




inspired from `http://scikit-learn.org`

Adaptive Boosting of Decision stumps

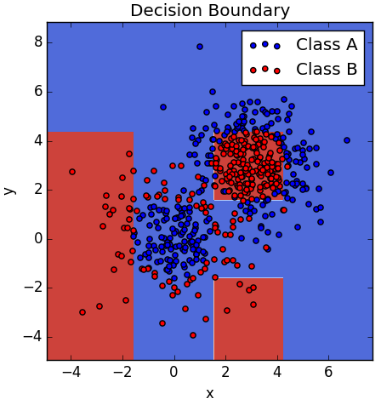
ensemble of 5 decision stumps

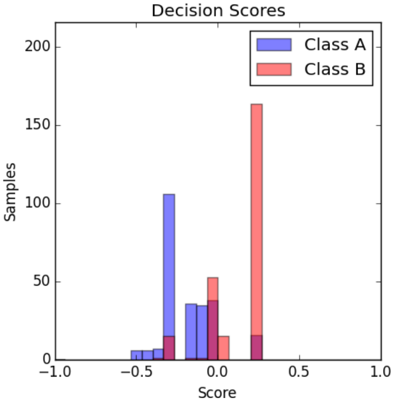


inspired from `http://scikit-learn.org`

Adaptive Boosting of Decision stumps

ensemble of 10 decision stumps

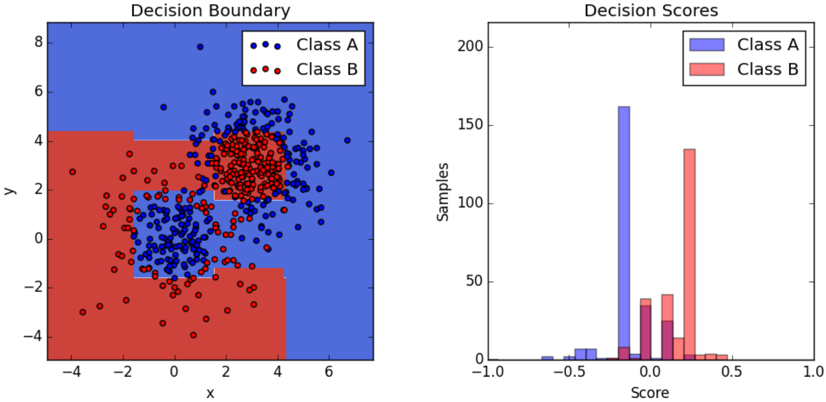




inspired from `http://scikit-learn.org`

Adaptive Boosting of Decision stumps

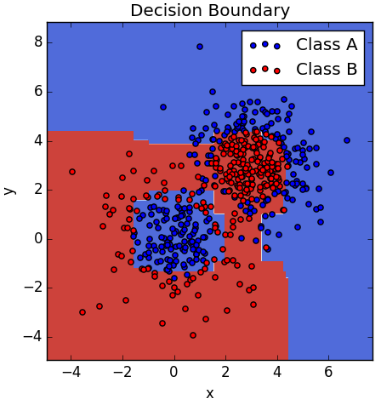
ensemble of 20 decision stumps

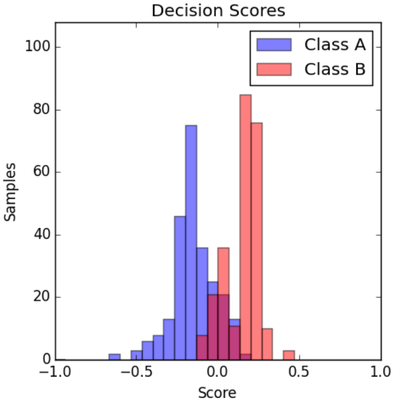


inspired from `http://scikit-learn.org`

Adaptive Boosting of Decision stumps

ensemble of 50 decision stumps

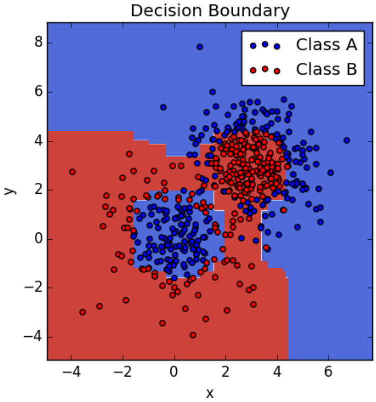


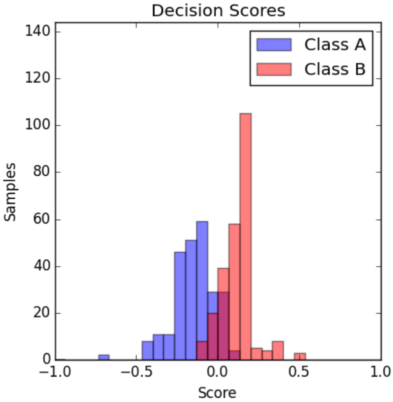


inspired from `http://scikit-learn.org`

Adaptive Boosting of Decision stumps

ensemble of 100 decision stumps

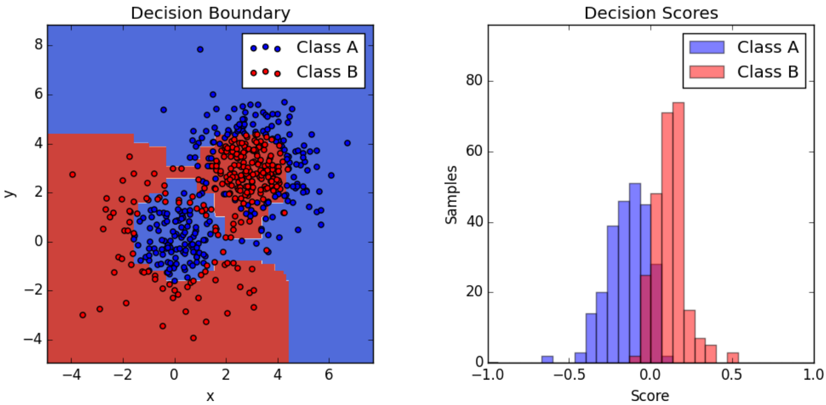




inspired from `http://scikit-learn.org`

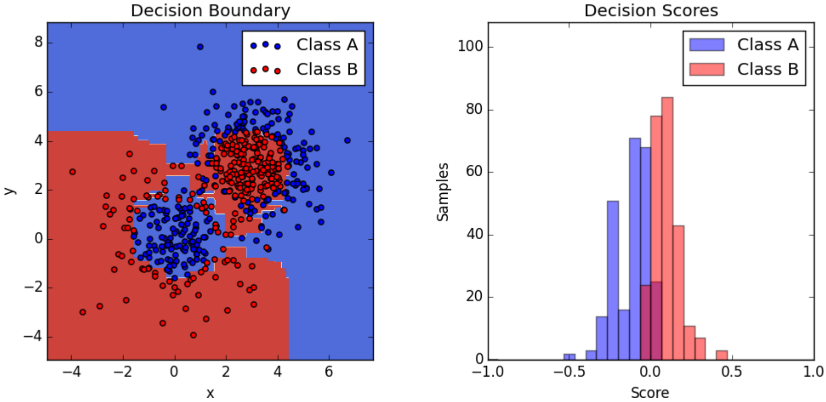
Adaptive Boosting of Decision stumps

ensemble of 200 decision stumps



inspired from `http://scikit-learn.org`

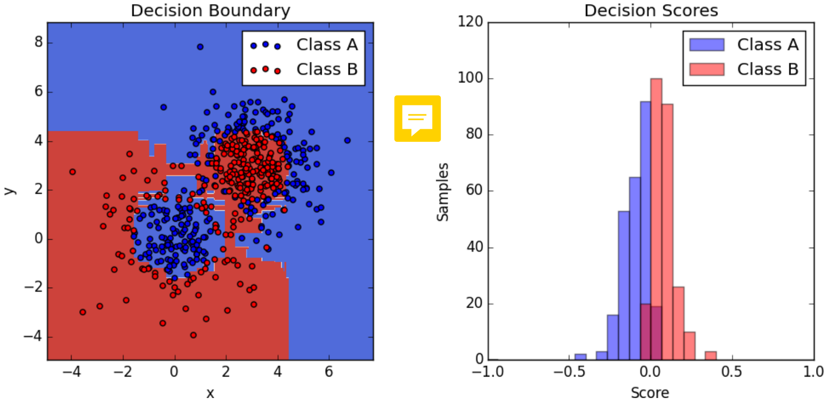
ensemble of 500 decision stumps



inspired from `http://scikit-learn.org`

Adaptive Boosting of Decision stumps

ensemble of 1000 decision stumps

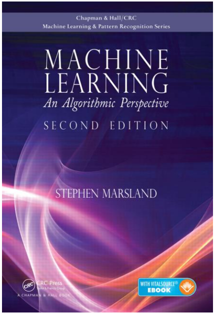


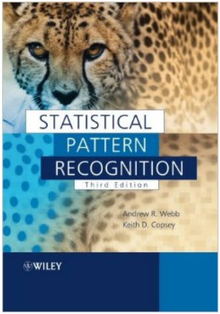
inspired from `http://scikit-learn.org`

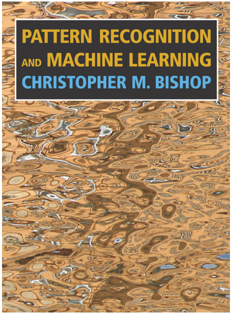
Outline of this Lesson

- ensemble learning
- bootstrap aggregating
- random forests
- adaptive boosting

References


Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series
MACHINE LEARNING
An Algorithmic Perspective
SECOND EDITION
STEPHEN MARSLAND
CRC Press
WITH VITALSOURCE® EBOOK


STATISTICAL
PATTERN
RECOGNITION
Third Edition
Andrew R. Webb
Keith D. Copsey
WILEY


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP