Ensemble Learning

Concepts and Algorithms of Artificial Intelligence (WS2022)

FH University of Applied Sciences TECHNIKUM WIEN

Lecturer: Sharwin Rezagholi
Authors: Stefan Lackner, Bernhard Knapp, Sharwin Rezagholi

Regression
KNN regression
Regression trees
Linear regression
Multiple regression
Ridge and Lasso regression
Neural networks

Classification
KNN classification
Classification trees
Ensembles & boosting
Random Forest
Logistic regression
Naive Bayes
Support vector machines
Neural networks

Supervised learning

AI

Clustering
k-means
Hierachical clustering
DB-scan

Non-supervised learning

Data handling
EDA
Data cleaning
Feature selection
Class balancing
etc

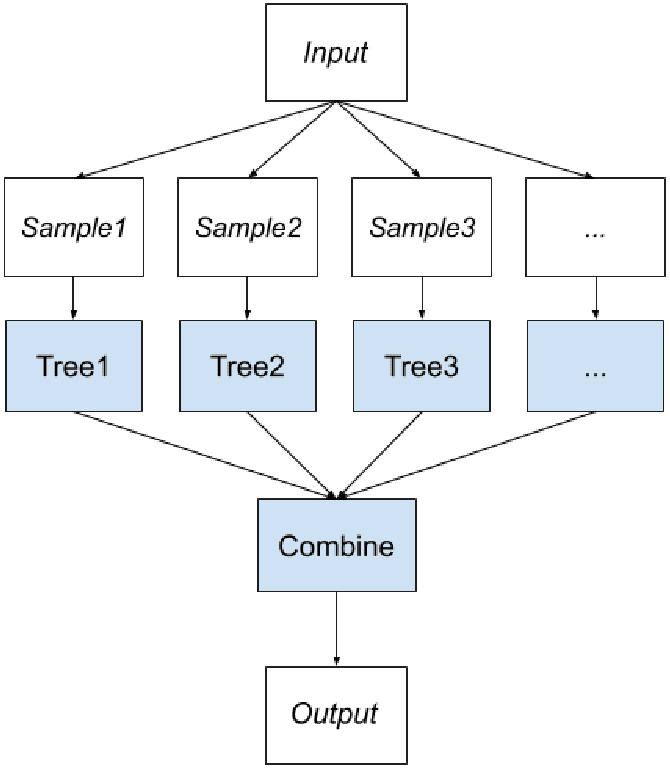
Dimensionality reduction
PCA / SVD
tSNE
MDS

Reinforcement learning
Covered in a separate lecture.

Contents

FH University of Applied Sciences TECHNIKUM WIEN

Ensemble Learning: Basic Idea



Motivation

- Ensemble learning is a process of combining multiple models (classifiers or regressors) to solve an ML problem.
- Ensembles (especially random forests) and boosting are powerful and deliver competitive results.
- Ensembles are conceptually easy to understand.
- Given sufficient computational power, ensemble learning is easy to implement.
- The bagging method of ensemble learning is easy to parallelize.
- There are many ensemble methods with different aims and properties.

FH University of Applied Sciences TECHNIKUM WIEN

Demotivation

- Ensemble methods increase training and test time.
- Performance increases level out. In some domains (e.g. computer vision, natural language) best results are obtained by deep learning, usually without ensemble methods.
- Ensemble methods turn interpretable base-learners (more on terminology later) into black box models. Additional analytical methods are needed to extract „the meaning“ of an ensemble.

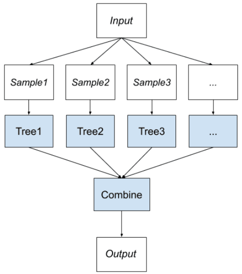
FH University of Applied Sciences TECHNIKUM WIEN

Contents

FH University of Applied Sciences TECHNIKUM WIEN

Ensemble Learning

- The **combination of different models** to obtain a final result.
- “Different models“ may refer to
  - **different algorithms** (e.g. trees, SVMs, …),
  - **same algorithm trained differently** (e.g. trees trained on different subsets of the data, or on differently weighted data).
- The **output of all models** is used to obtain a final result.
- Single models are called **base-learners**.
- **Applicable to many problems**: Regression, classification, and clustering (consensus-clustering).



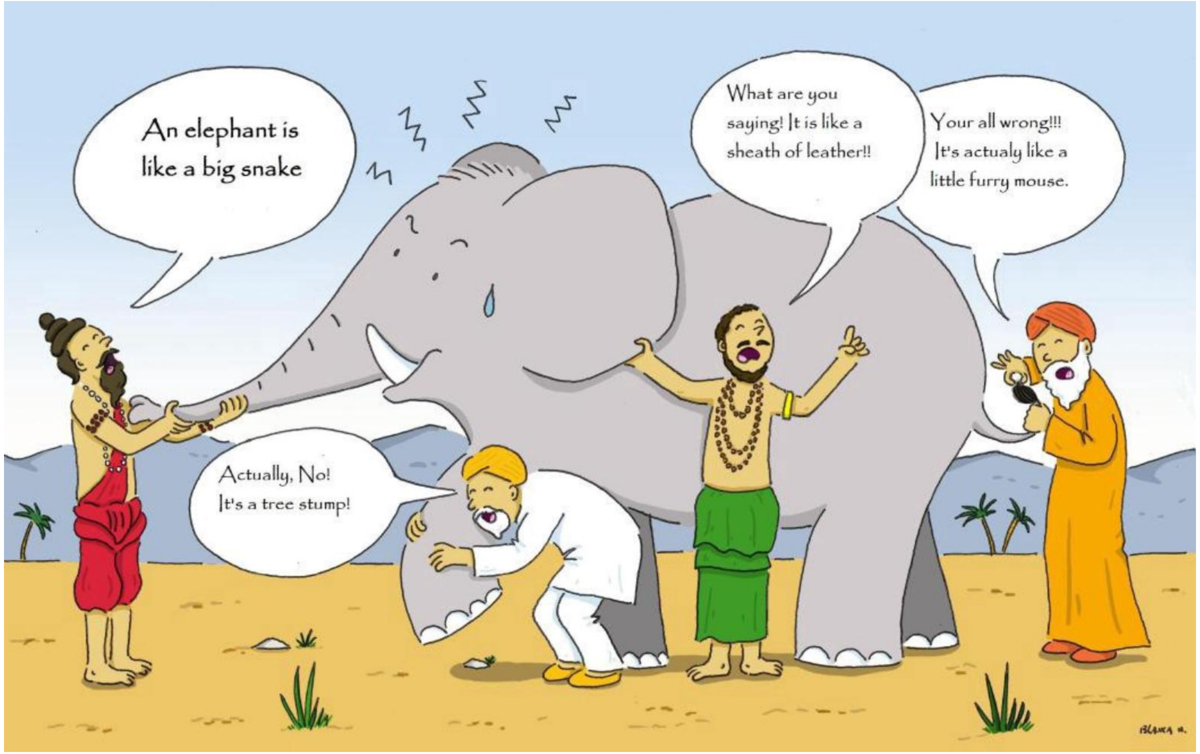
FH University of Applied Sciences TECHNIKUM WIEN

Weak Base Learners vs. Strong Ensembles

- In ensemble learning **base-learners do not need to be strong**.
- A **strong model** is a **high-accuracy-classifier/regressor.**
- **Weak learners may only be slightly better than chance** (e.g. only reaching 0.51 accuracy).
- **Combining many weak learners can lead to a strong ensemble** if base learners are sufficiently independent.

Many weak learners together can do a great job

(… while one single strong learner might achieve less than the sum of many weak learners …)



[https://medium.com/ml-research-lab/ensemble-learning-the-heart-of-machine-learning-b4f59a5f9777]

Contents

FH University of Applied Sciences TECHNIKUM WIEN

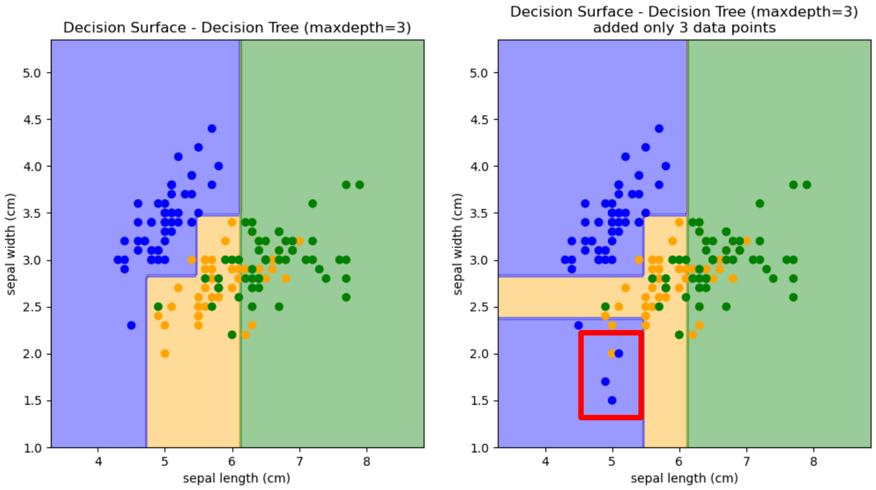
Diversity of Classifiers

Diversity among base learner can be reached by:

(1) using **different algorithms**,

(2) using the same **high-variance algorithm** on different training sets (e.g. resampling).

Recall: A high-variance model changes much if the data changes little (example: decision trees).



High variance:
3 new datapoints lead to a very different tree.

Bias-Variance Decomposition

- Consider a regression.
- We make errors when using the fitted model for prediction.
- If we use a sufficiently large test set we can approximate the expected mean squared error (MSE) of the model.
- The expected MSE on test data can be **additively decomposed into 3 quantities**:
- (1) The **variance of the model**,
- (2) the **squared bias**,
- (3) the **variance of the error**.

Bias-Variance Decomposition

Error = Variance + Bias + Noise

$$\mathbb{E}\Big(y_i - \hat{f}(x_i)\Big)^2 = \mathrm{Var}\Big(\hat{f}(x_i)\Big) + \mathrm{bias}\Big(\hat{f}(x_i)\Big)^2 + \mathrm{Var}(\epsilon)$$

(x_i, y_i), a tuple from the test set

$\mathbb{E}\Big(y_i - \hat{f}(x_i)\Big)^2$, expected MSE on test data if model is trained om different training sets

$\mathrm{Var}\Big(\hat{f}(x_i)\Big)$, variance of the model (how different the model is when using different training sets)

$\mathrm{bias}\Big(\hat{f}(x_i)\Big)$, error due to choice of model class (example: linear model for non-linear data-generating process)

$\mathrm{Var}(\epsilon)$, irreducible variance of the error term

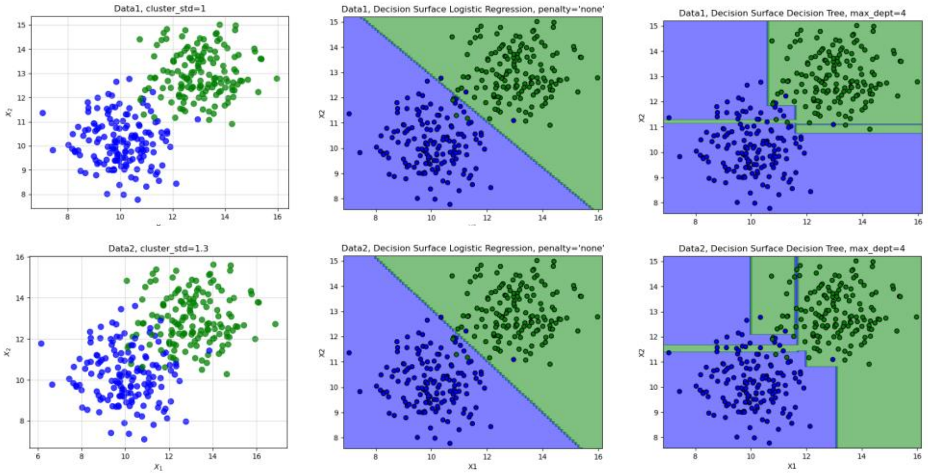
*We cannot do anything about the **irreducible** part but we can handle **bias** and **variance**.*

FH University of Applied Sciences TECHNIKUM WIEN

High vs. Low-Variance Models

- Decision tree: $$\mathbb{E}\left(y_i - \hat{f}(x_i)\right)^2 = \boxed{\mathrm{Var}\left(\hat{f}(x_i)\right)} + \boxed{\mathrm{bias}\left(\hat{f}(x_i)\right)^2} + \mathrm{Var}(\epsilon)$$
- Logistic regression: $$\mathbb{E}\left(y_i - \hat{f}(x_i)\right)^2 = \boxed{\mathrm{Var}\left(\hat{f}(x_i)\right)} + \boxed{\mathrm{bias}\left(\hat{f}(x_i)\right)^2} + \mathrm{Var}(\epsilon)$$

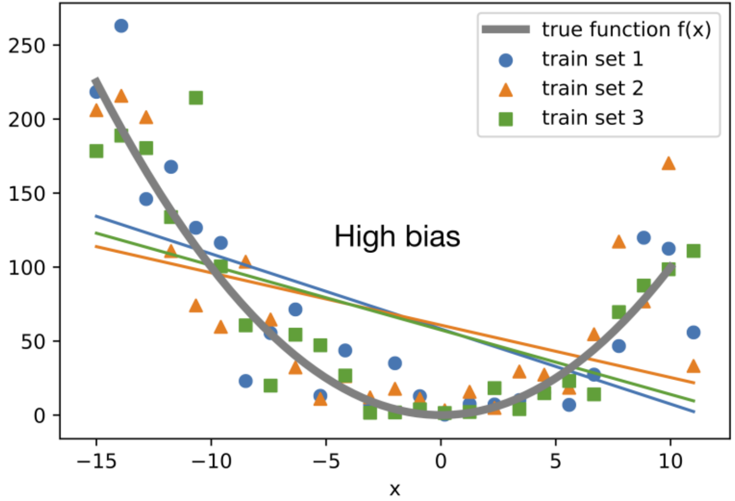
Data | Log. reg. | Tree



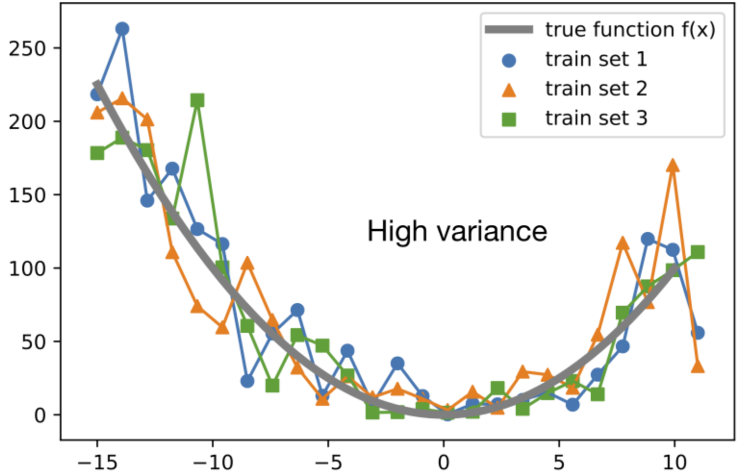
University of Applied Sciences FH TECHNIKUM WIEN

High vs. Low-Variance Models

http://rasbt.github.io/mlxtend/user_guide/evaluate/bias_variance_decomp/



Also known as **underfitting**.

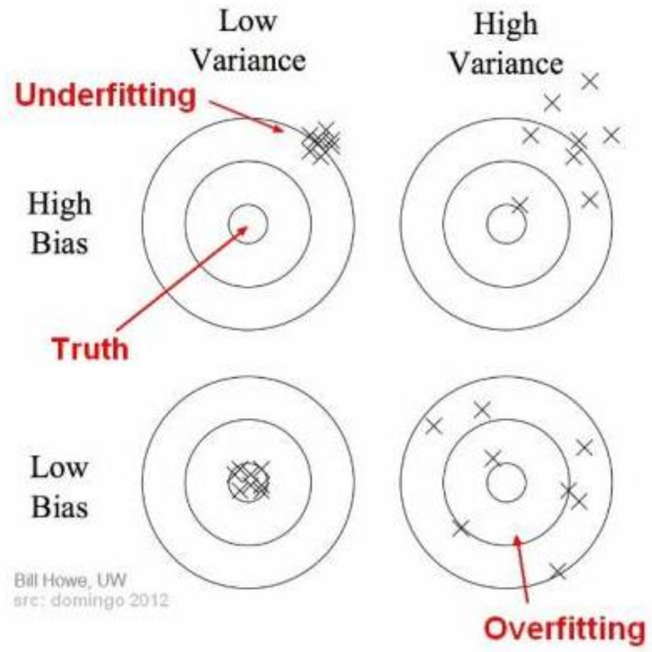


Also known as **overfitting**.

- The **bias** stems from **erroneous assumptions in the learning algorithm**. High bias can cause an algorithm to miss the relevant relations between features and target outputs (**underfitting**).
- The **variance** stems from **sensitivity to small changes** in the training data. High variance may result from an algorithm modeling the noise in the training data (**overfitting**).

FH University of Applied Sciences TECHNIKUM WIEN

High vs. Low-Variance Models



https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229
https://medium.com/ml-research-lab/ensemble-learning-relation-with-bias-and-variance-431cdc0a3fc9

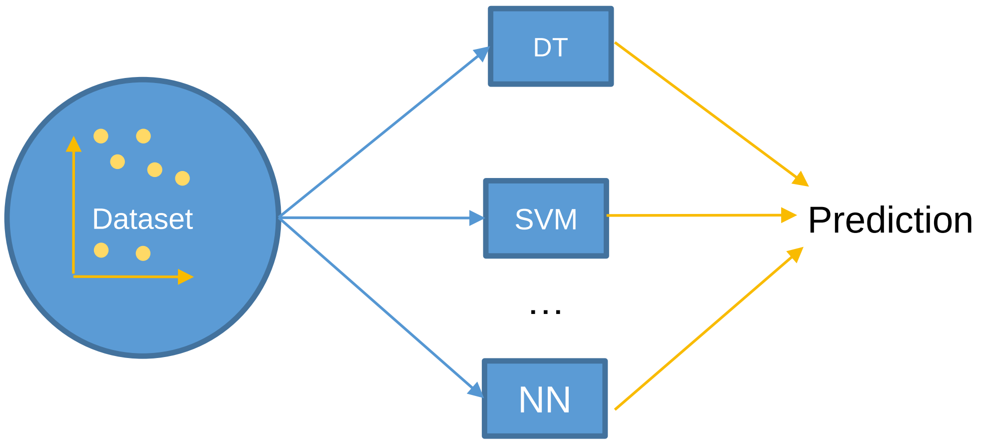
Contents

FH University of Applied Sciences TECHNIKUM WIEN

Ensemble Method: Voting and Averaging

Combine the results of several classifiers by “letting them vote”.

- **Classification:**
  - **Hard voting**: Predict the plurality class.
  - **Soft voting**: Take the average of the estimated class probabilities.
- **Regression:**



Simple Voting & Bias Reduction

- The **combination of different classifiers by voting** leads to a **reduction in bias** since different classifier algorithms are biased in different ways.
- The amount of reduction depends on the data and the difference in biases of single models.
- We assume here that each classifier is trained using the same trainin g data. But we can do better (see next slides).

$$\mathbb{E}\left(y_i - \hat{f}(x_i)\right)^2 = \boxed{\mathrm{Var}\left(\hat{f}(x_i)\right)} + \boxed{\mathrm{bias}\left(\hat{f}(x_i)\right)^2} + \mathrm{Var}(\epsilon)$$

Voting ensemble

$$\mathbb{E}\left(y_i - \hat{f}(x_i)\right)^2 = \boxed{\mathrm{Var}\left(\hat{f}(x_i)\right)} + \boxed{\mathrm{bias}\left(\hat{f}(x_i)\right)^2} + \mathrm{Var}(\epsilon)$$

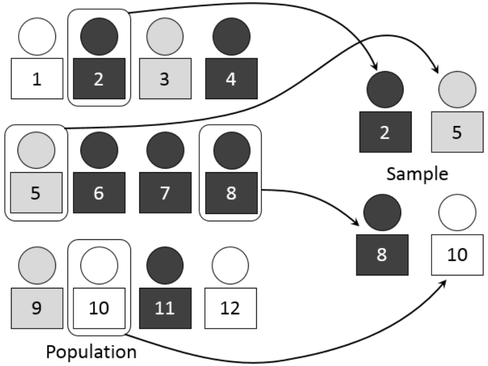
Contents

- Motivation and Demotivation
- Ensemble Learning (Basic Idea)
- Bias-Variance Decomposition
- Simple Voting
- **Bagging and Pasting**
- Random Patches and Random Subspaces
- 2 Popular Ensembles Techniques (Overview)
- Stacking
- Recap & Exercises

FH University of Applied Sciences TECHNIKUM WIEN

Bagging and Pasting

- We do not use all the training data at once.
- **Bagging** and **Pasting** are **resampling techniques:**
  - **Bagging** (**b**ootstrap **agg**regat**ing**): Repeatedly draw random samples from a training set **with replacement**.
  - **Pasting**: Draw random samples **without replacement** (each observation in the sample can be used once). Requires a sufficiently large dataset.
- Resampling techniques are used to **reduce variance.**



Bagging and High-Variance Classifiers

$$\mathbb{E}\Big(y_i - \hat{f}(x_i)\Big)^2 = \boxed{\mathrm{Var}\Big(\hat{f}(x_i)\Big)} + \boxed{\mathrm{bias}\Big(\hat{f}(x_i)\Big)^2} + \mathrm{Var}(\epsilon)$$

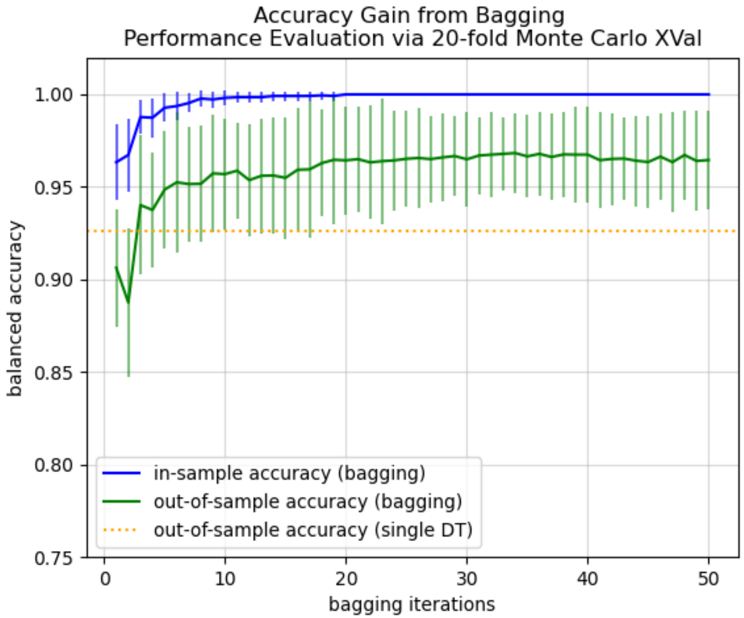
$$\Big\downarrow \text{Bagging}$$

$$\mathbb{E}\Big(y_i - \hat{f}(x_i)\Big)^2 = \boxed{\mathrm{Var}\Big(\hat{f}(x_i)\Big)} + \boxed{\mathrm{bias}\Big(\hat{f}(x_i)\Big)^2} + \mathrm{Var}(\epsilon)$$

- Making your **base models as independent as possible** benefits the procedure.
- Increasing the variance of base-learners may **increase the generalization capability** of the ensemble.
- In practice one needs to **balance variance and the number of base-learners**.
- Low-variance models will not benefit (much) from bagging.

FH University of Applied Sciences TECHNIKUM WIEN

Increased Generalization Capability due to Bagging

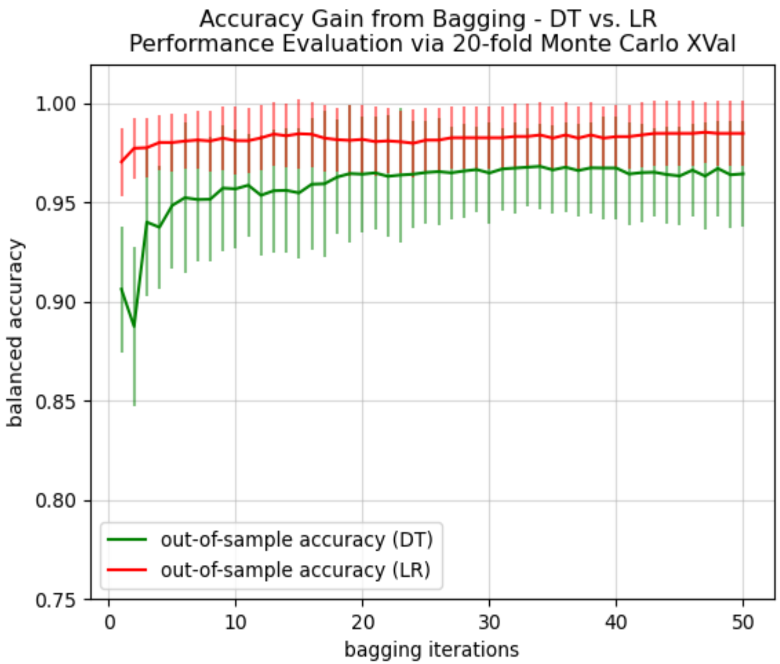


Note: The function BaggingClassifier() in SK-Learn performs soft voting if predict_proba() is available.

- **Gains** from bagging **arise quickly** and **taper out fast**.
- Training accuracy reaches 1.
- Test accuracy increased by 0.05 by bagging 20 base learners.
- No tuning was used for the single decision trees (default parameters are used).
- Unrestricted decision trees (which have very high variance) are used.

FH University of Applied Sciences TECHNIKUM WIEN

Bagging: Decision Trees vs. Logistic Regressions



- Log. reg. performs better than bagged decision trees on this data.
- **Log. reg. is a low variance model**, hence it does not gain much from bagging.
- However: Be careful and try different things on your own datasets.

FH University of Applied Sciences TECHNIKUM WIEN

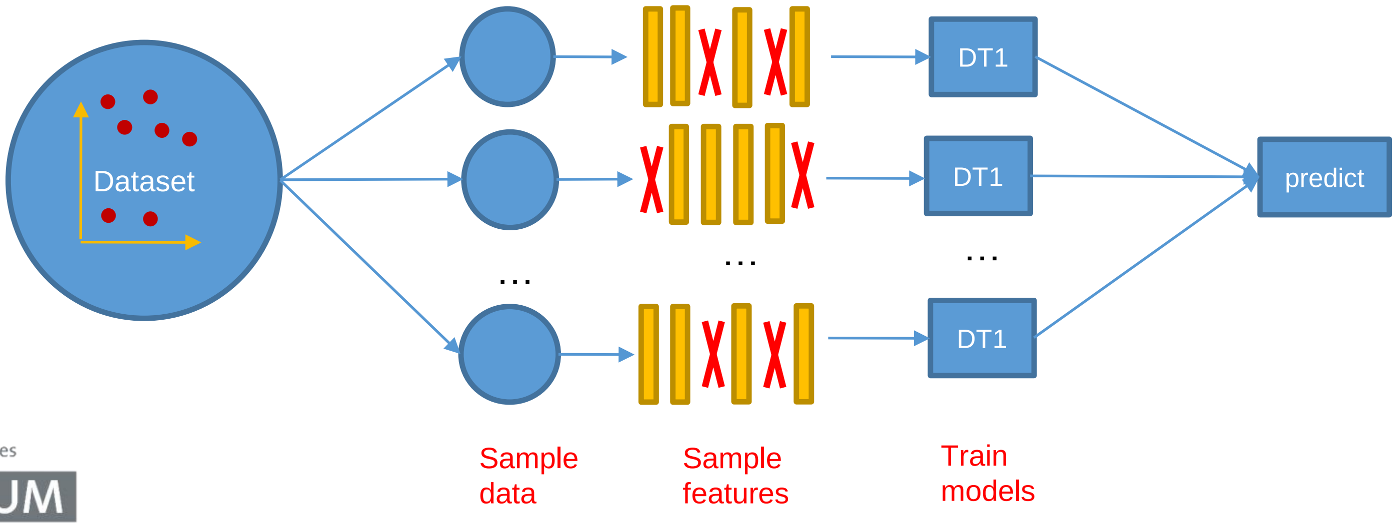
*James, Witten, Hastie, Tibshirani (2017), p. 37.

Contents

FH University of Applied Sciences TECHNIKUM WIEN

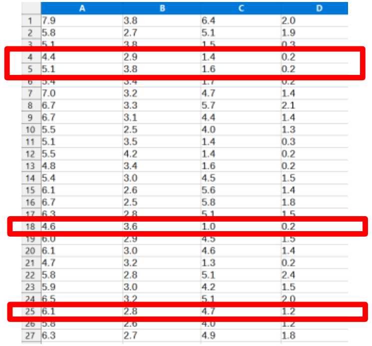
Random Patches and Random Subspaces

- Bagging & Pasting reduce variance and work best when base learners are diverse.
- Idea: Increase diversity further by **sampling not only the data but also the features used in training base learners**?
- Doing this **in combination** with bagging is called **random patches**. Doing this **in isolation**, this is called **random subspaces.**

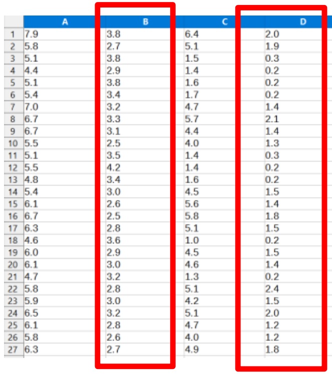


Summary

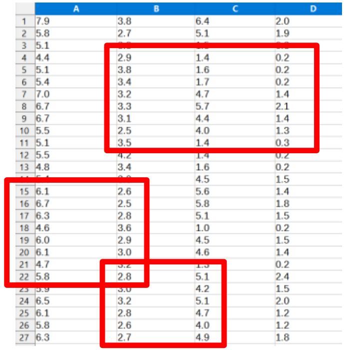
- **Bagging and pasting**:
  - Resampling from data (observations).
- **Random Subspaces**:
  - Resampling features (columns).
  - Sometimes called “feature bagging”.
- **Random Patches**:
  - Resampling from data and features (observations and features).
  - Sometimes called “random subspace plus bagging”.



Bagging



Random subspace

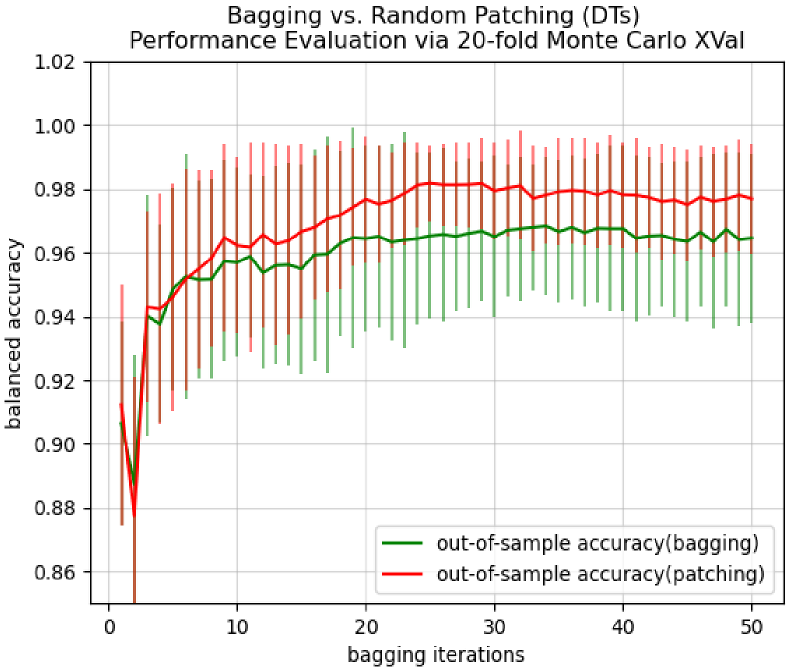


Random patch

These illustrations are very simplified.
Random choice is key in resampling.

FH University of Applied Sciences TECHNIKUM WIEN

Bagging vs. Random Patches



- Patching increases the diversity of base learners.
- Increased diversity leads to higher gains from bagging.
- Since the diversity of base learners is higher, gains from bagging taper out later (roughly at ensemble size 25).

Contents

FH University of Applied Sciences TECHNIKUM WIEN

Random Forests

- Random forests are **ensembles of unrestricted decision trees** obtained by **a variant of random patches**.
- Instead of sampling features after bootstrapping a sample, **features for spitting are resampled at every node** (for every split).
- The intensity of resampling increases from bagging, to random patches, to random forests.
- Random forests will be the focus of a dedicated lecture in this course.

FH University of Applied Sciences TECHNIKUM WIEN

Boosting

Base learners are trained **sequentially** (not in parallel as in bagging).

Aim: Reduction of both, bias and variance.

Two approaches:

1. Critical datapoints are reweighted to emphasize “complicated” obervations.
2. Single models are fitted to the residuals of the previous models.

Boosting will be the focus of a dedicated lecture in this course.

Contents

FH University of Applied Sciences TECHNIKUM WIEN

Stacking

- Stacking is an **extension of the voting classifier/regressor** that **replaces plurality-vote/average by an additional classifier/regressor called the blender**.
- This is **vaguely similar to the intuition behind deep learning**: Layers of predictors are learned sequentially, each layer takes the predictions of the earlier layer as input.
- As in voting methods different models classes are combined.

FH University of Applied Sciences TECHNIKUM WIEN

[PDF] Stacked generalization | Semantic Scholar

Stacking

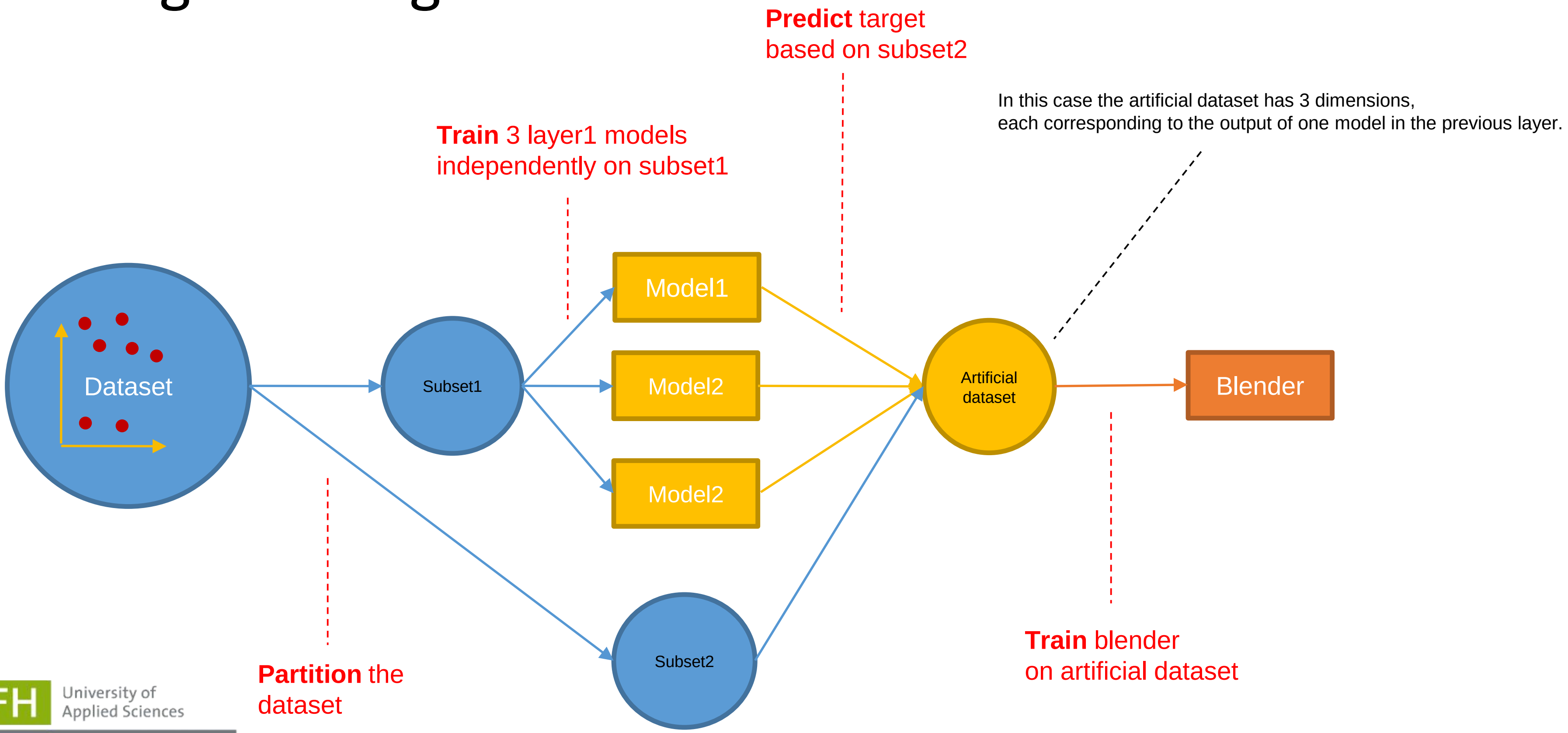
Procedure:

1) On a **dedicated data subset**, several **base learners are fitted** (perhaps using different algorithms).
2) On a **second data subset**, the **blender** is trained. The blender is another model trained on the predictions of the other classifiers.

- By partitioning the training data into m subsets, one can stack m layers.
- **Partitioning is nescessary** to ensure that inputs to subsequent layers are not overfit.

Online resource: https://www.youtube.com/watch?v=sBrQnqwMpvA&t=10s

Stacking: Training

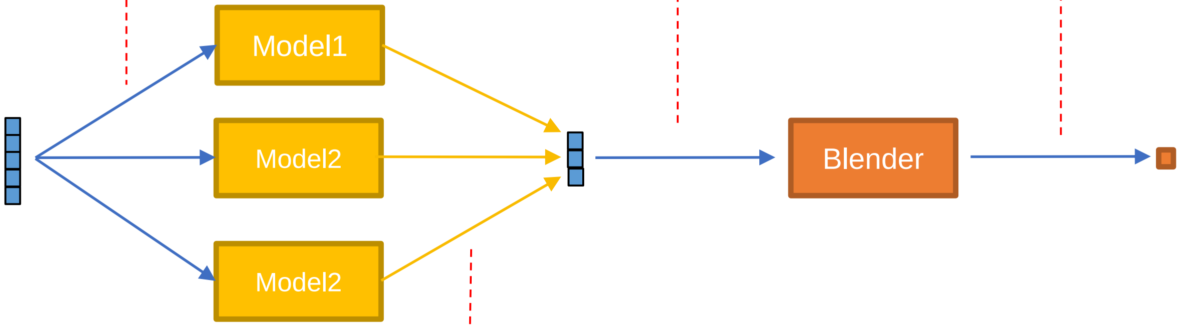


Stacking: Prediction

A new datapoint with unknown label
is routed to the first layer of models.

The stacked prediction vector
is routed to the blender.

The blender makes the final
Prediction based on the
3-dimensional vector from
the first layer.



The 3 models make a prediction,
when stacked together this yields
a 3-dimensional vector.

Note: Stacking-like ideas are often called cascading.
E.g. [Cascaded Random Forest for Fast Object Detection | SpringerLink]

Stacking: Regression vs. Classification

- In regression models the first layer contains models that output real-valued predictions.
- In classification models the first layer contains models that output **categorical values**.
- In classification using SK-Learn **use predict_proba()** in layer1.

Parallel vs. Sequential Implementation

- Bagging, Pasting, Random Patches, and Random Subspaces can be implemented in **parallel**, therefore they are **easily scalable**.
- Boosting and Stacking are **sequential** in nature: **Scaling is computationally expensive**.

Black-Box vs. White-Box

- Using ensembles, (simple) base learners can be combined into powerful models.
- The downside is that their **interpretability is lost** to a large extent. They become **black-box models**.
- Example: While a small decision tree is a white-box model, interpreting the behavior of an ensemble of decision tree

FH University of Applied Sciences TECHNIKUM WIEN

Contents

- Motivation and Demotivation
- Ensemble Learning (Basic Idea)
- Bias-Variance Decomposition
- Simple Voting
- Bagging and Pasting
- Random Patches and Random Subspaces
- 2 Popular Ensemble Techniques (Overview)
- Stacking
- **Recap & Exercises**

FH University of Applied Sciences TECHNIKUM WIEN

Recap

- **Ensemble learning is powerful**. It is the only way to obtain models for certain datasets.
- There is **no guarantee that ensembles will improve performance**. Experience shows that simple **voting** and **stacking** are less likely to increase performance than **ensembles based on resampling**.
- Ensembles that allow for parallel estimation are computationally efficient

Assignment: Random patches.

a) Explain (and compare, for example in a table) a plurality-voting ensemble, bagging/pasting, random patches, random subspaces, stacking, and boosting.

Use simple self-made images or even hand drawings (of which you take a photo).

Use self-written explanations. Do not copy from the lecture slides or the internet (neither text nor images).

b) Implement a version of random patches without the use of a library. Use a base learner of your choice (e.g. a tree) and compare the performance with a single learner.

Use a dataset of your choice. You will need a dataset with a large number of observations and features.

FH University of Applied Sciences TECHNIKUM WIEN

References

- Géron A. (2017): Hands-On Machine Learning with Scikit-Learn & Tensorflow. – O‘Reilly.
- James G., Witten D., Hastie T., Tibshirani R. (2017): An introduction to Statistical Learning. – Springer.
- Kuhn M., Johnson K. (2016): Applied Predictive Modeling. – Springer.
- Berk R. (2016): Statistical Learning from a Regression Perspective. – Springer.
- Bagging predictors | SpringerLink
- The random subspace method for constructing decision forests | IEEE Journals & Magazine | IEEE Xplore

FH University of Applied Sciences TECHNIKUM WIEN