Random Forest

a non-probabilistic discriminative classifier

Contents

Bootstrapping

- Original question: how to determine an accuracy measure for a classifier if only (relatively) few training data are available?
- **Generation of bootstrap data sets:**
  - Out of N samples $\mathbf{x}_n$ of the training data set, $N_1 < N$ samples are selected randomly
  - This is done **with replacement**, i.e. a point $\mathbf{x}_n$ may be included several times in the bootstrap data set
  - An certain number B of these boostrap data sets is generated. These data sets are considered to be independent
  - The classifier is trained using each data set
  - Uncertainty measure, derived from the variation of the results

Application of the Bootstrap Principle

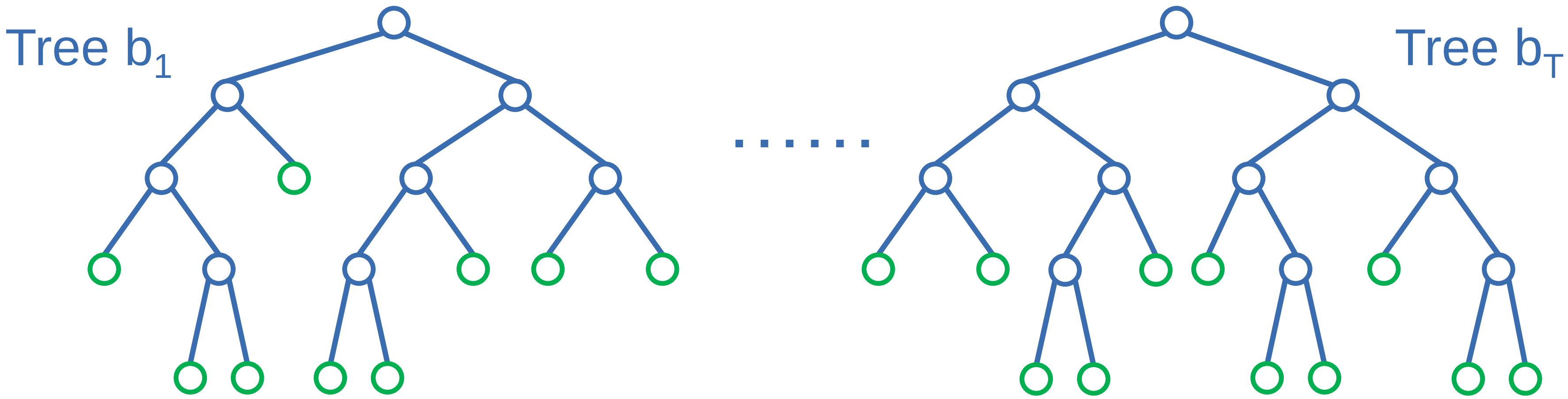
- Using of bootstrap data sets for classification
  → **Bagging** (Bootstrap AGGregatING):
  - Training:
    - Generate B bootstrap data sets
    - Each data set b is used to train a classifier $f_b(\mathbf{x})$
  - Classification:
    - For each classifier f_b: Determine result $C_b = f_b(\mathbf{x})$
    - Result: The class that gets the highest number of “votes“

Application of the Bootstrap Principle

- Bagging can be applied to any classifier (… but for some it might not make much sense)
- However, the classifiers f_b usually are of the same type
- Bagging can improve results of unstable classifiers
- CART are "unstable" in the sense that small changes in the input data may lead to major changes in the class boundaries
- Bootstrapping for CART → **Random Forests**

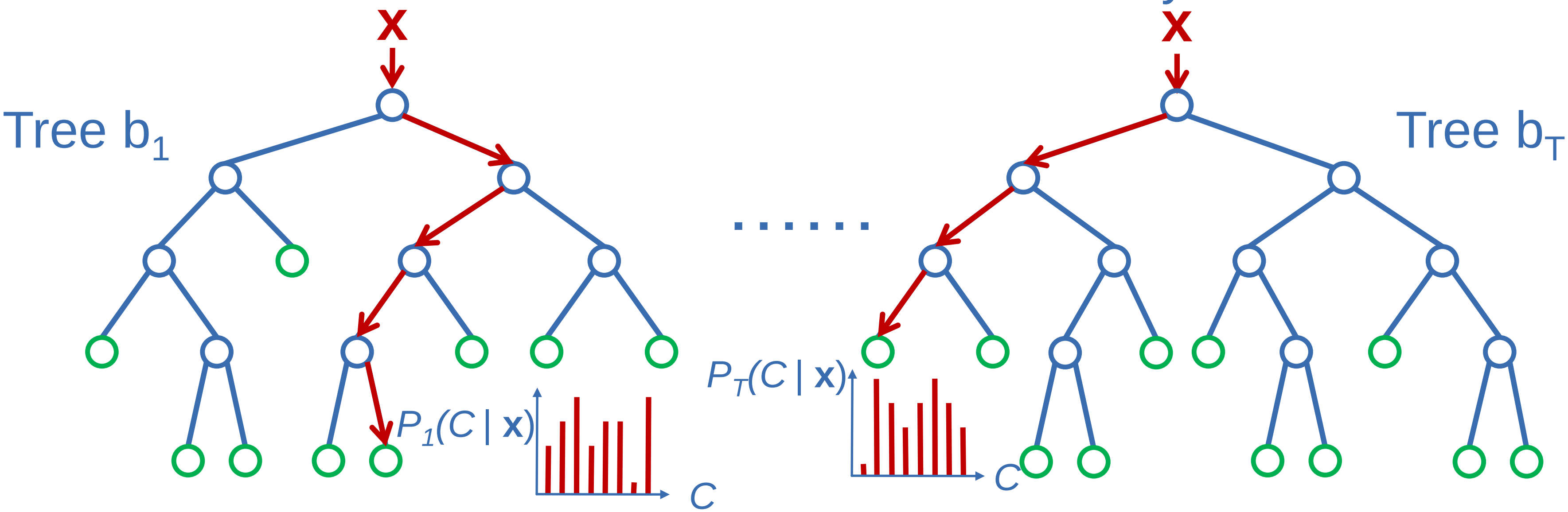
Random Forests

- A Random Forest [Breiman, 2001] consists of *T* decision trees (CART)
- Classification: A feature vector **x** is classified by each tree



Random Forests

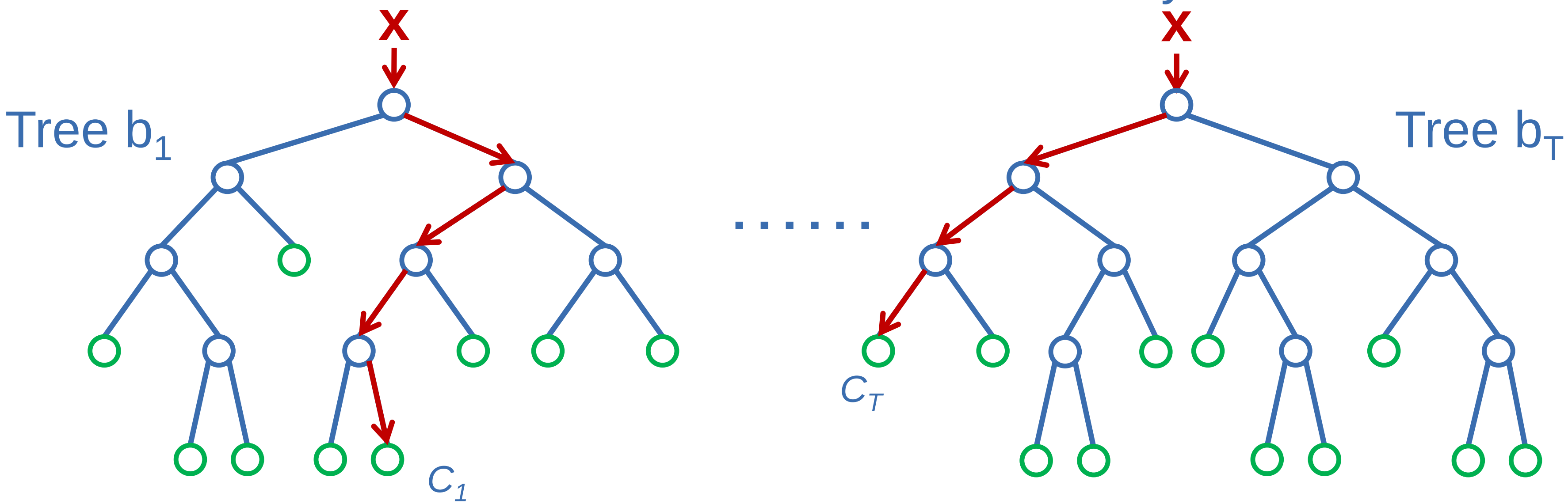
- A Random Forest [Breiman, 2001] consists of T decision trees (CART)
- Classification: A feature vector **x** is classified by each tree



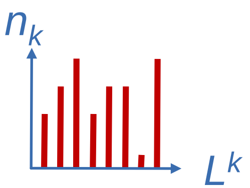
- In every tree t: posterior distribution $P_t(C \mid \mathbf{x})$ according to tree t
- Posterior: average probability: $P(C \mid \mathbf{x}) = \frac{1}{T} \cdot \sum_{t=1}^{T} P_t(C \mid \mathbf{x})$

Random Forests

- A Random Forest [Breiman, 2001] consists of T decision trees (CART)
- Classification: A feature vector **x** is classified by each tree



- Alternative: each tree casts a vote for one class
- Posterior: relative number of votes: $P\left(C = L^k \mid \mathbf{x}\right) = \frac{n_k}{T}$



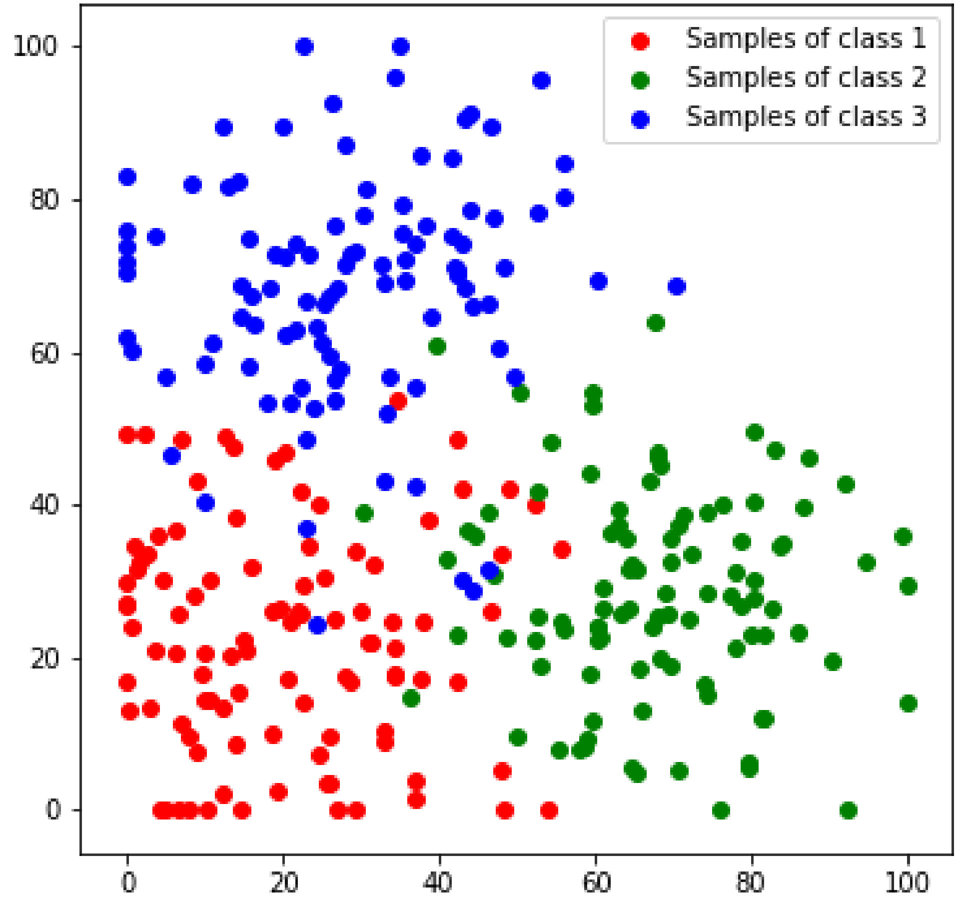
Histogram of votes

Training for Random Forests

- Generation of T bootstrap data sets (e.g. $T = 50$)
- Using each data set t one tree b_t is trained (cf. CART training)
- Important: Independent and random selection of the bootstrap data sets
- Due to the combination of several trees :
  - Better generalization
  - Better stability
- Can be parallelised easily due to the independence of the bootstrap data sets and the trees.

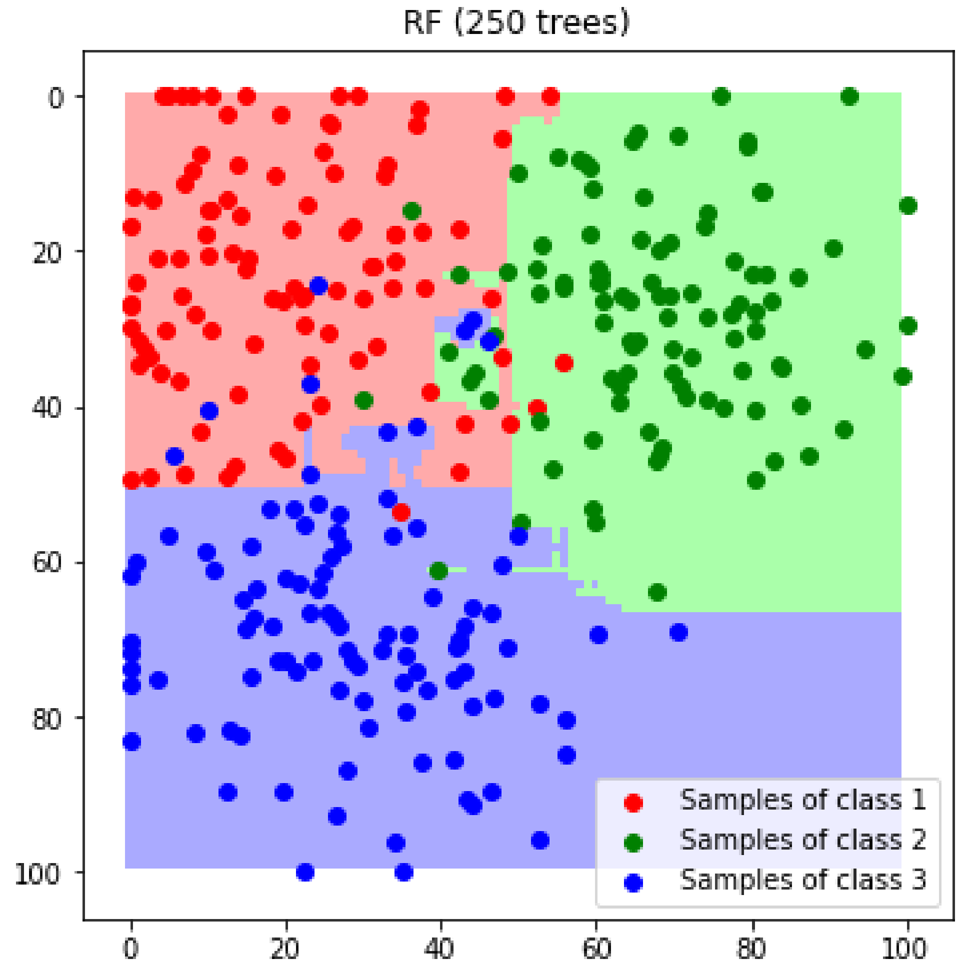
Example

Example with three classes and two features



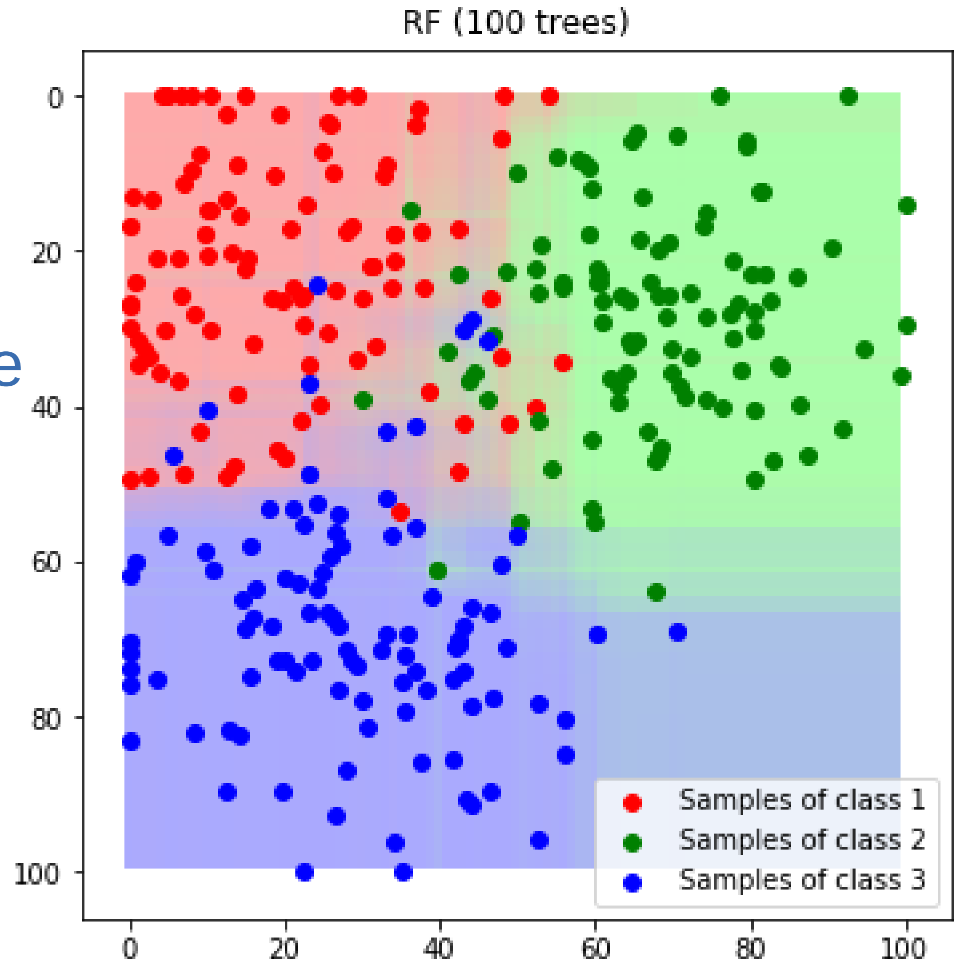
Example

- Depth of 7
- Completed feature space is visualized with final class label
- Classification result after combing 250 trees
- No strong overfitting



Example

- Classification result after combing 100 trees
- The completed feature space is visualized with the predicted probabilities
- Good decision boundaries



Random Forests: Discussion

- Random forests are considered to be one of the best local classifiers today
- Similar quality to the results as SVM, but much faster
- Training and classification can be parallelized easily
- Can be applied to multi-class problems by design
- Probabilistic interpretation of the outputs → can be used in subsequent processing steps
- Random forests may consume a lot of memory
- There are implementations in Matlab, openCV, …

Random Forests: Application Examples

- Classification of aerial images and DSM [Schindler, 2012]
- Matching of SIFT features by classification [Lepetit et al., 2006]
- Classification of laser scanner data [Niemeyer et al., 2013]
  - Features: Height above ground, local distribution of points, full-waveform features
  - Classification of each individual point with RF