Machine Learning for Applications in Computer Vision

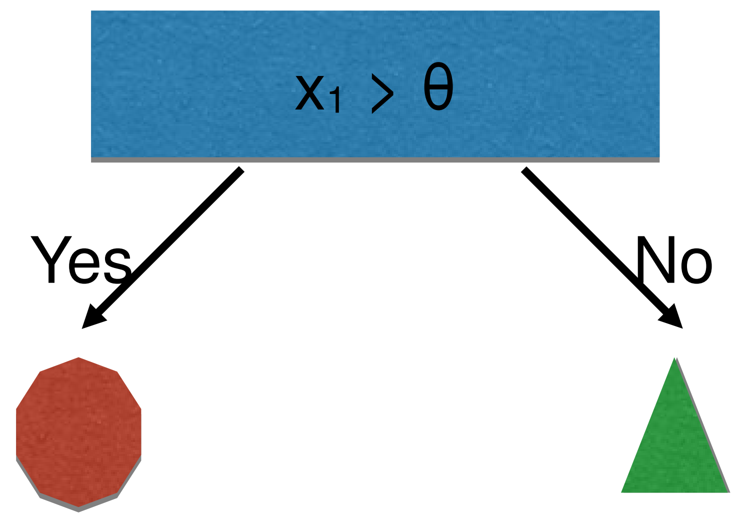
Tree-based Classifiers

Overview

- Decision Stump
- From stumps to trees
  - Growing a tree
  - Pruning a tree
  - Pros and Cons
- From trees to forests
  - Random Forests theory
  - Applications
  - Pros and Cons
- Applications (Learning with trees online)
  - Online learning
  - Online Random Forests
  - Mondrian Forests

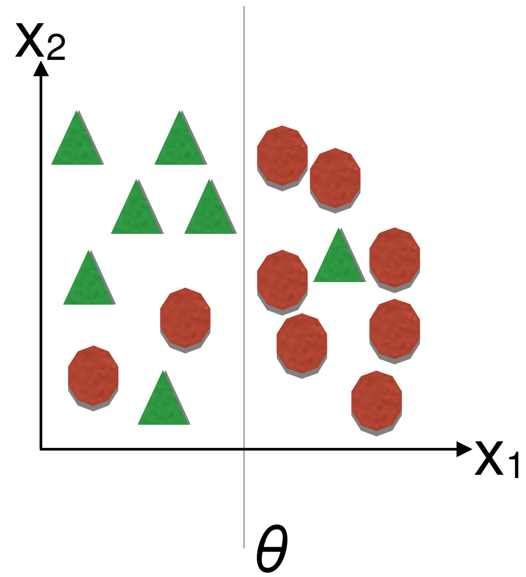
Decision Stump

- One level decision tree
- One internal node (root) connected to its terminal nodes (leaves)



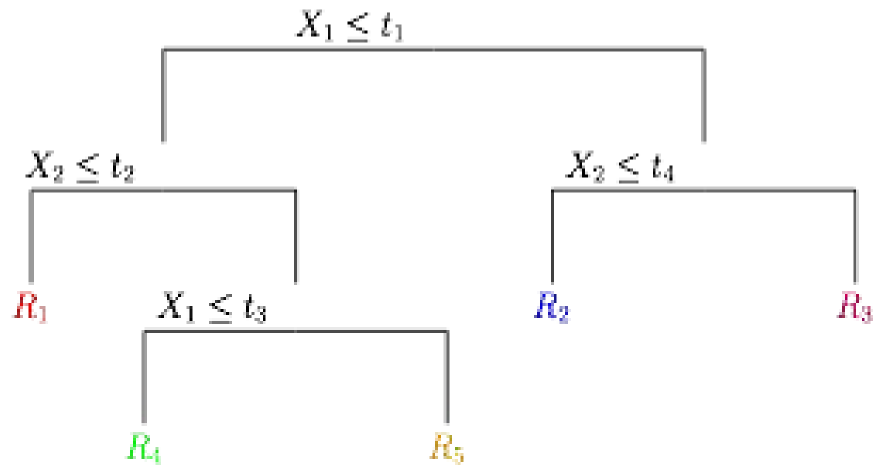
- **Goal:**
  Find axis aligned hyper plane that minimizes the class. error
- Class. error is always better than random guessing (50%),

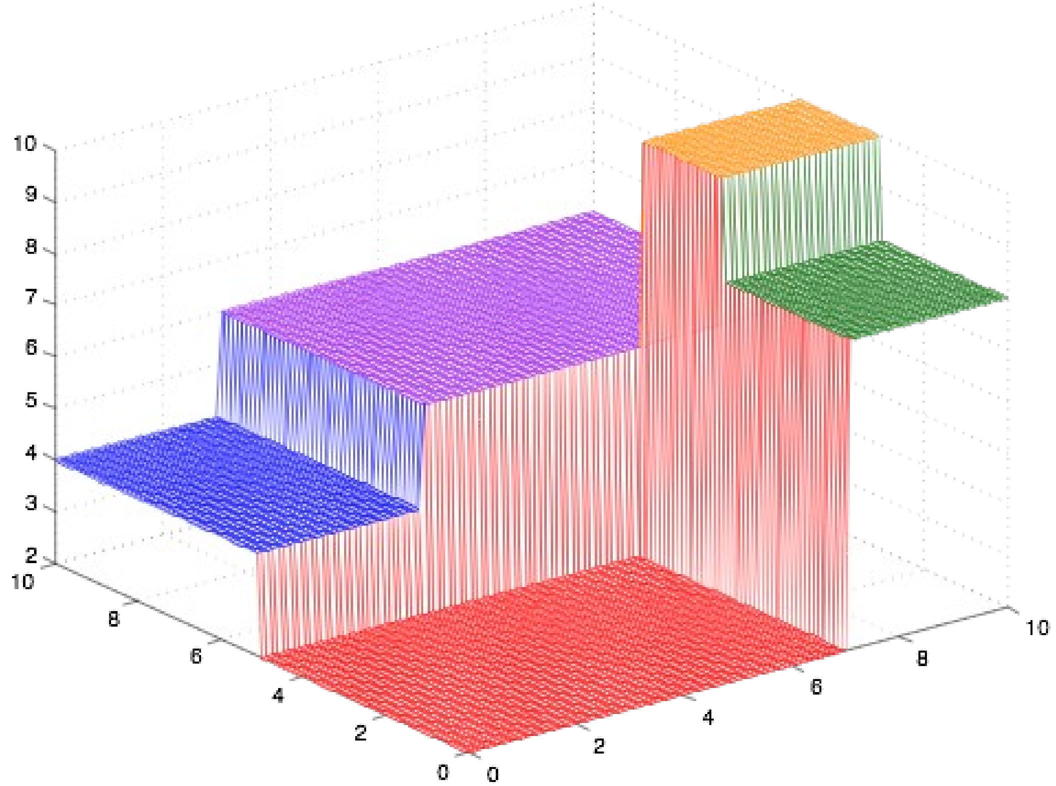
  weak classifier



Decision Trees

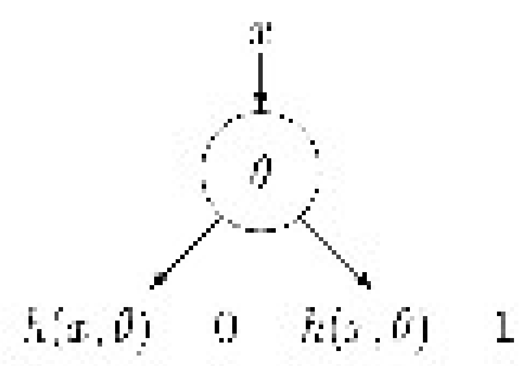
- Classification and Regression Trees (CART)
- Extension of Decision Stump
- Partition the input space recursively
- Define a label for each resulting region of the input space.



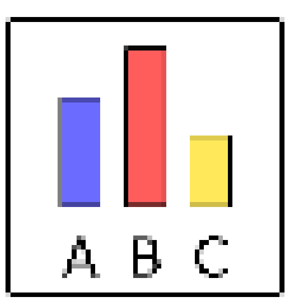


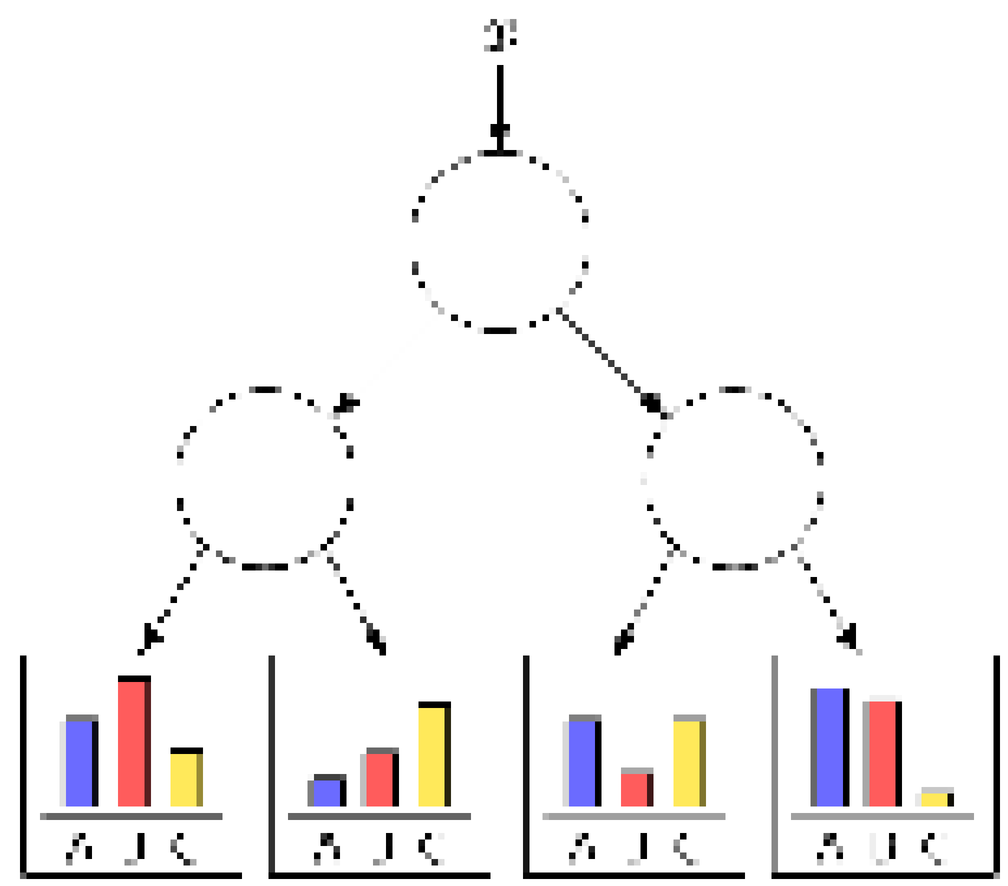
Decision Trees

- Regression: assign mean response to each leaf (piecewise constant surface)
- Classification: store the distribution over class labels in each leaf
  - Inner node:



  - Leaf node:





Growing a Tree

- NP-Complete problem (NP: Non-deterministic Polynomial time)
- Solution is locally optimal

- Minimize a cost function to find the best feature and its best threshold on each node
- Split the data on each node based on the chosen feature and the threshold
- Stopping criteria for growing the tree
  - reduction of cost too small ?
  - maximum depth is reached ?
  - is the distribution in the subtrees homogeneous ? (pure dist.)
  - is the number of samples in the subtrees too small ?

Growing a Tree

- Regression cost:

$$\text{cost}(D) = \sum_{i \in D} (y_i - \bar{y})^2 \qquad \bar{y} = \frac{1}{|D|} \sum_{i \in D} y_i$$

- Classification cost:
  - Misclassification rate:

$$\frac{1}{|D|} \sum_{i \in D} |y_i \neq \hat{y}|$$

  - Entropy:
    - same as maximizing the information gain

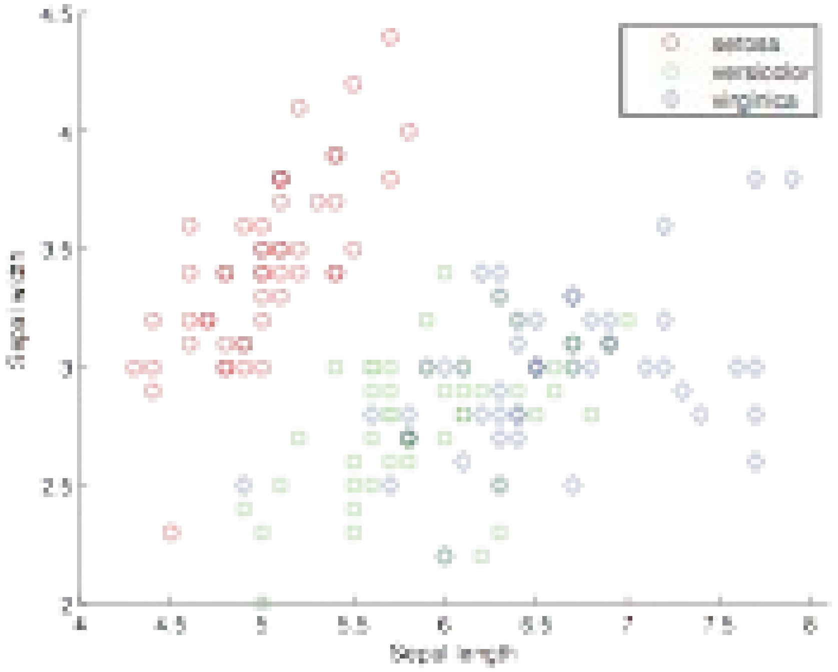
$$H(\hat{P}) = - \sum_{c=1}^{C} \hat{P}_c \log \hat{P}_c$$

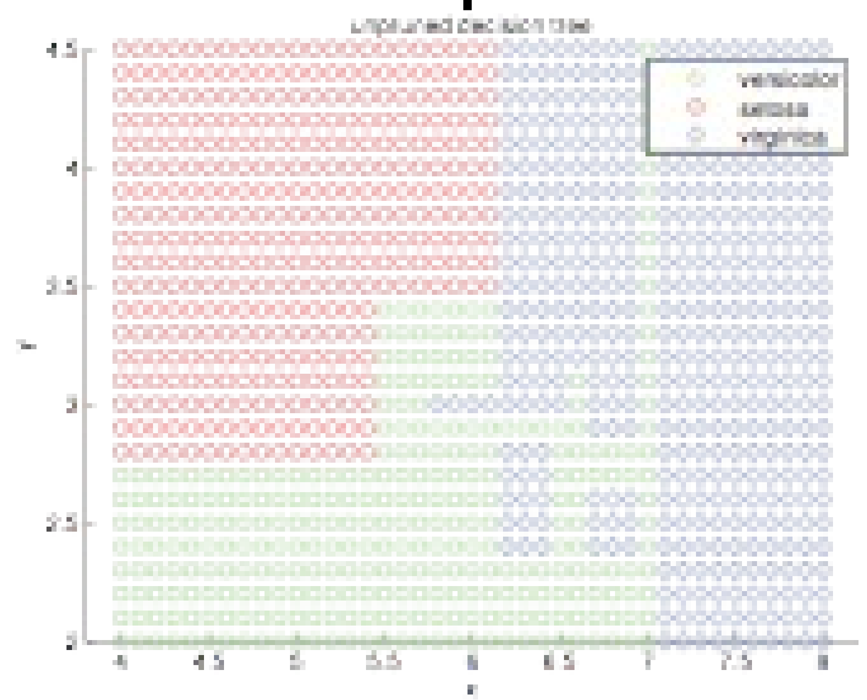
  - Gini Index:
    - expected error rate

$$1 - \sum_{c} \hat{P}_c^2$$

Pruning a tree

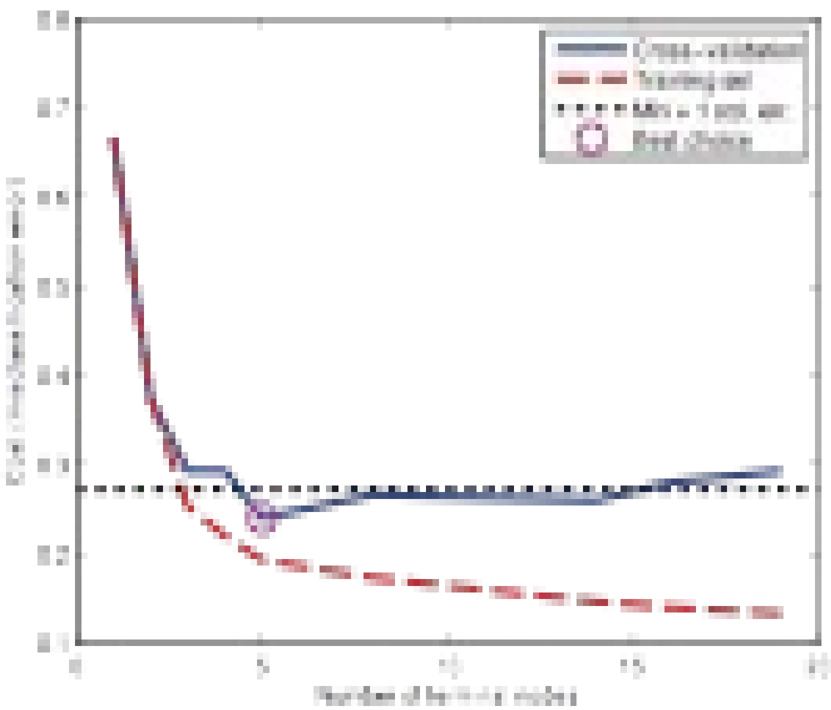
- Growing a tree too large yields overfitting
- Solution: build a full tree and then prune it

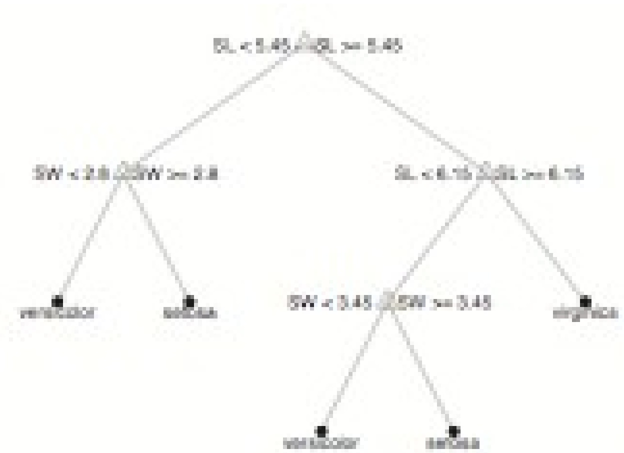


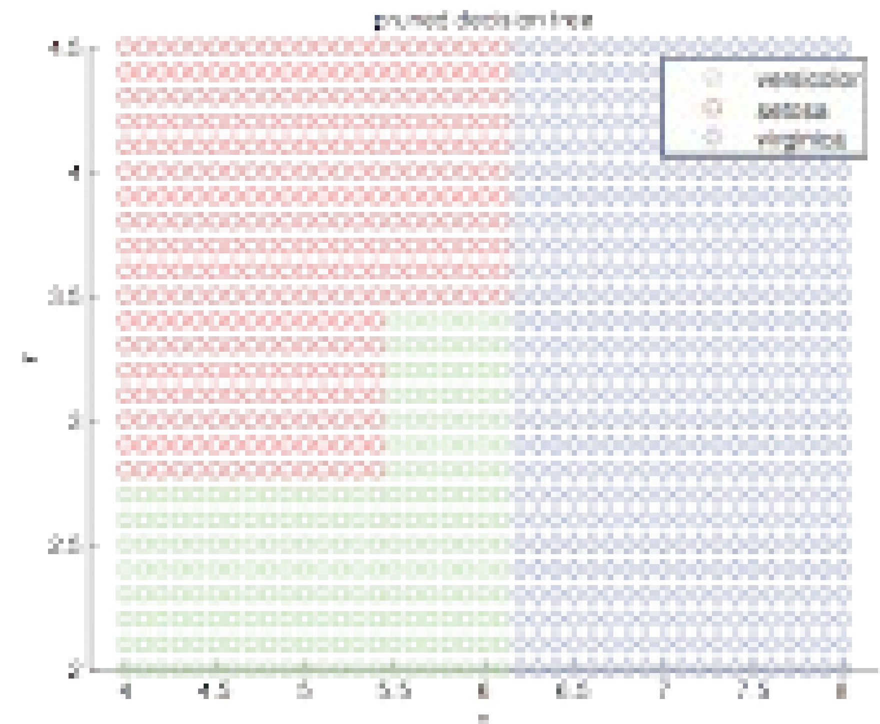


Pruning a tree

- Growing a tree too large yields overfitting
- Solution: build a full tree and then prune it







Pros (CART)

- easy to interpret
- can handle mixed discrete and cont. data
- insensitive to monotone transformations
- CART perform automatic variable selection
- relatively robust to outliers
- scale well to large datasets
- can be modified to handle missing inputs

Cons (CART)

- DO NOT predict very accurately
  - due to the greedy training procedure
- Trees are unstable
  - small change in the input might yield a large effect on the tree structure
- Trees are high variance estimators
  - Solution: Random Forests

Random Forests

- Reduce the variance of estimate by
  - Train M trees on different subsets of the data:

$$f(x) = \sum_{m}^{M} \frac{1}{M} f_m(x)$$

    - !!! highly correlated predictors
  - Solution: Choose data as well as variable (feature) randomly
- Known as Random Forests. RF has a high accuracy and widely used in practical studies.

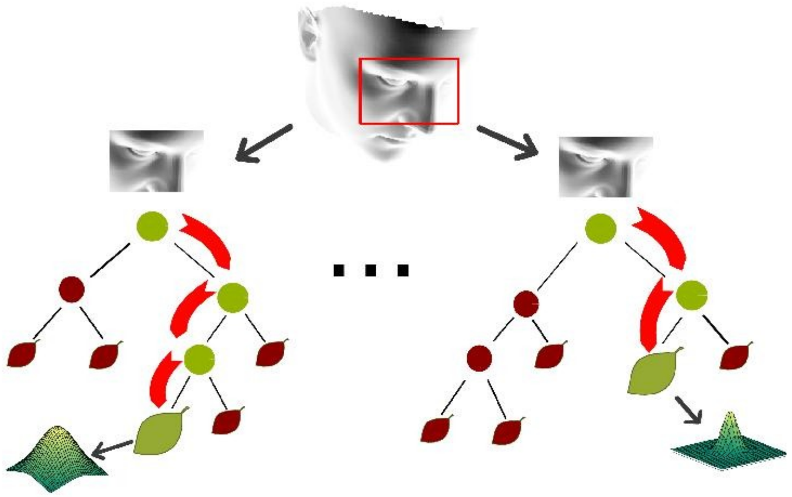
Random Forests

- **Real-Time Object Segmentation with Semantic Texton Forests**
  - James Shotton (winner of CVPR 2008 Demo Prize)

    - https://www.youtube.com/watch?v=oBYnnp-GQqY

Random Forests

- **Real Time Head Pose Estimation with Random Regression Forests** [Fanelli et al. (CVPR 2011)]



- https://www.youtube.com/watch?t=136&v=sxUkGGGtRBU

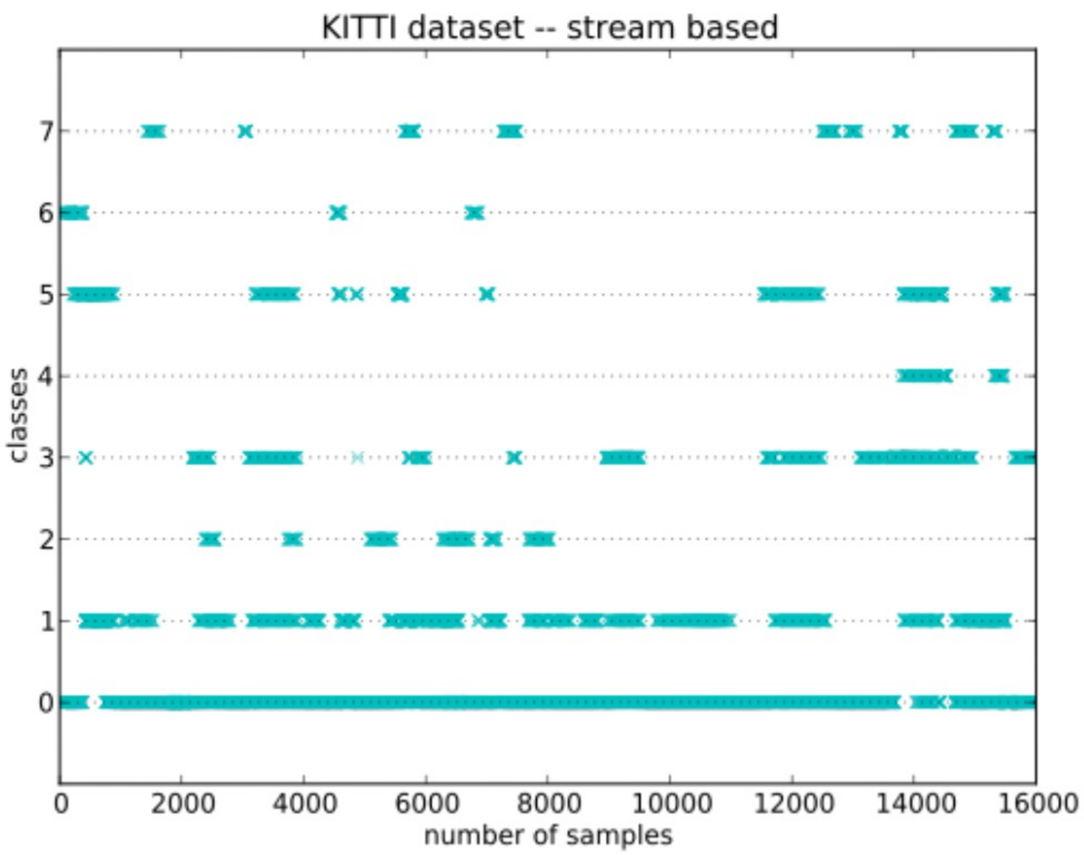
Random Forests (Pros and Cons)

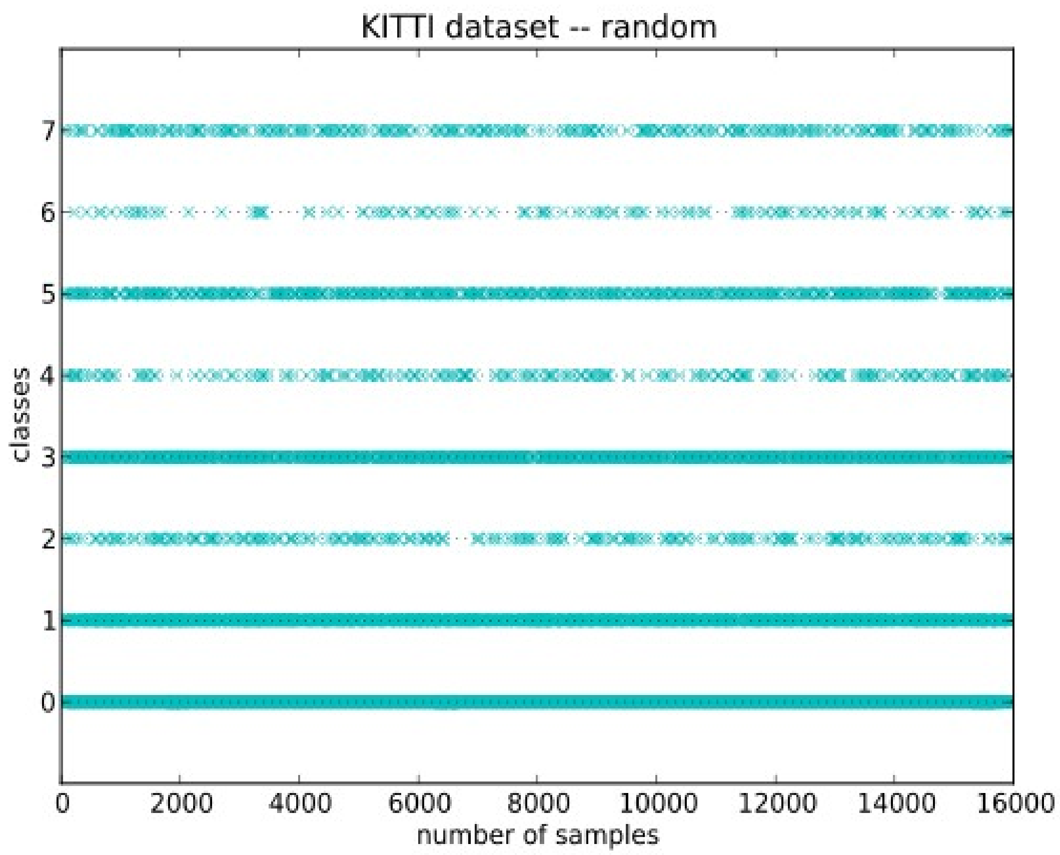
- very good predictive performance
- fast to train and test
- trees can be trained in parallel
- overfitting is avoided

- Not possible to train incrementally
- Retraining periodically is slow
  - And requires access to past data

Online Learning

- We receive data sequentially (in streams)
- The class sizes may vary significantly
- Application: self-driving cars (e.g. RGB-D sensors)





Online Random Forests

- **Online Random Forests** [Saffari et. al (ICCV 2009)]
- Applications:
  - Tracking (ORFs vs OnlineAdaBoost)

  - Interactive image segmentation

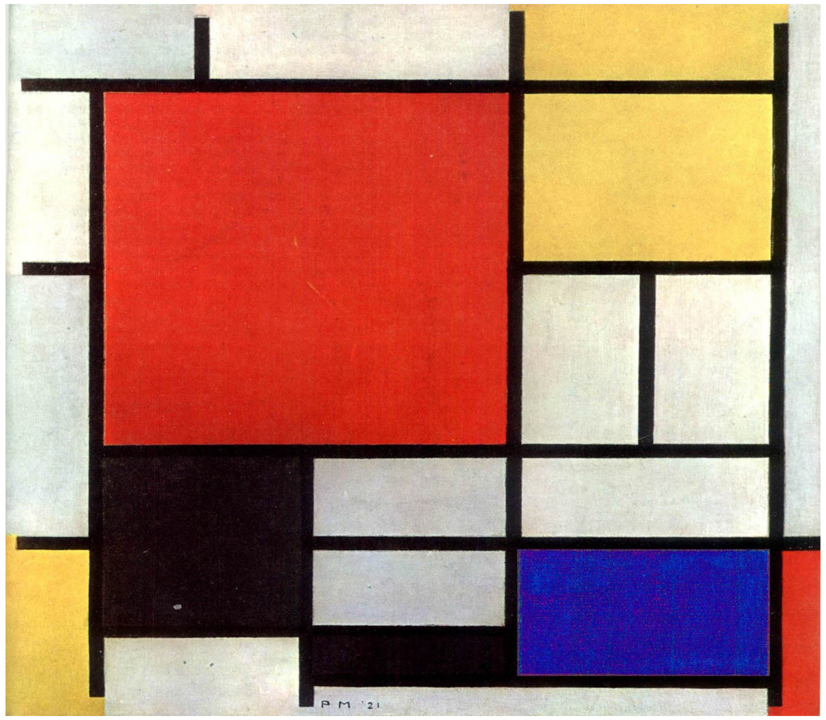
Figure 4. Interactive segmentation results using on-line random forests and a Total Variation based segmentation algorithm.

Mondrian Forests

- **Efficient Online Random Forests**
  [B. Lakshminarayanan, D.M. Roy and Y.W. Teh (NIPS 2014)]

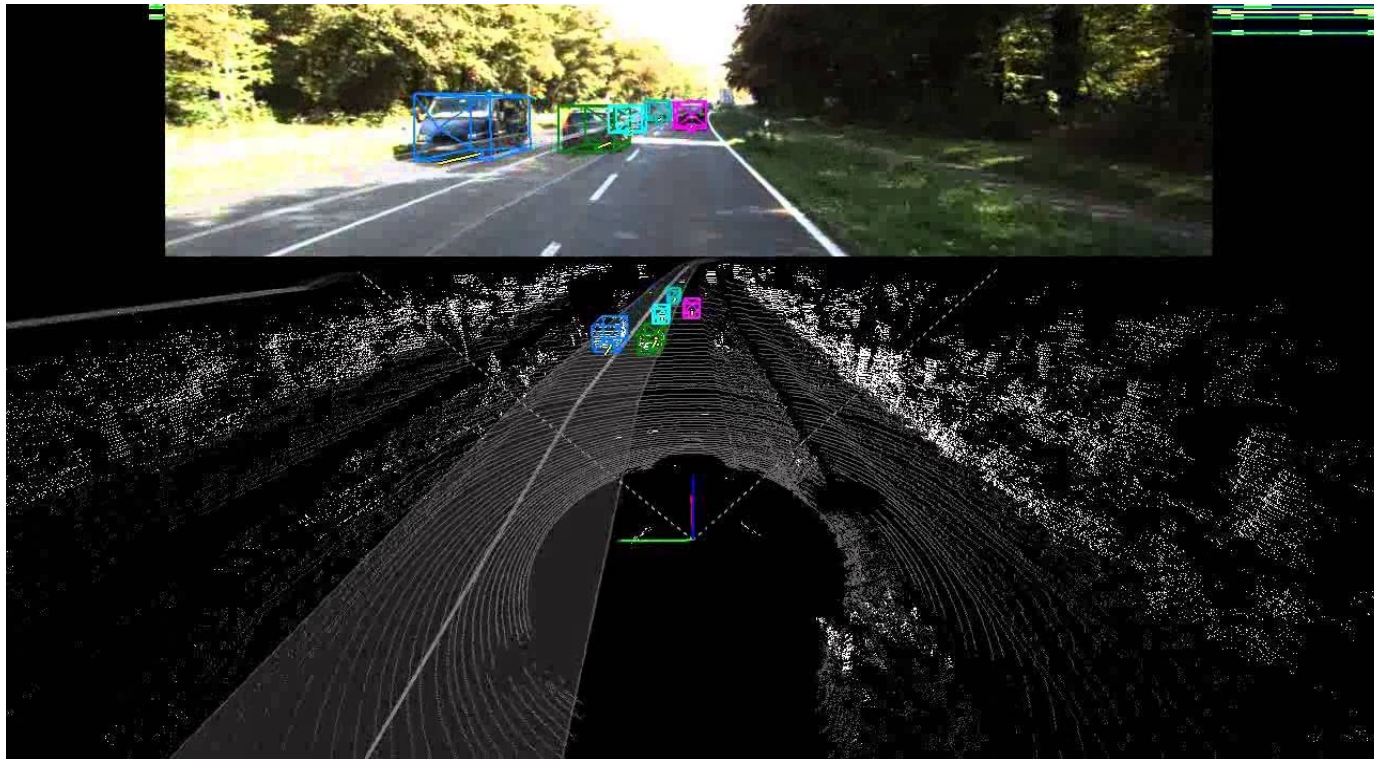
  - store also the **range** of the data in each dimension
  - are independent of the class labels
  - trees can grow upwards as well as downwards

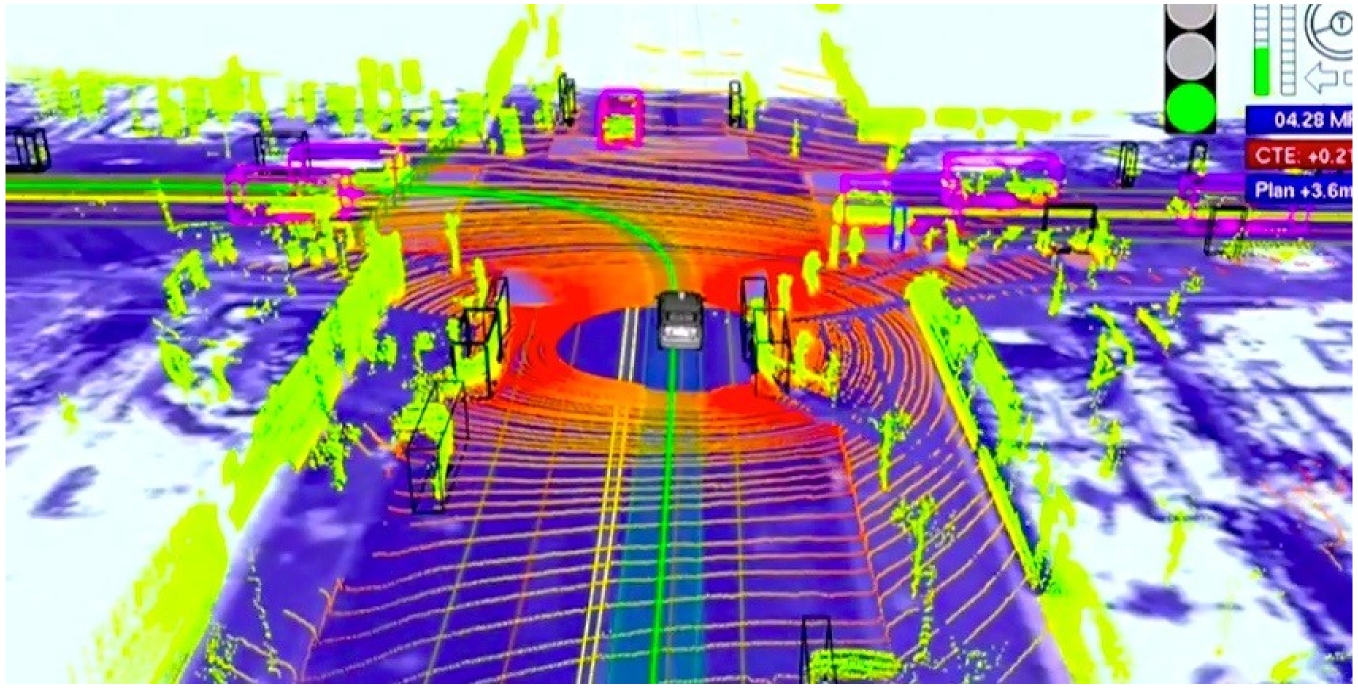
  - Name inspired by Piet Mondrian:
  - "*Composition with Red, Yellow, Blue, and Black*" 1926
    Gemeentemuseum, Den Haag



Mondrian Forests

- **Application to real data "KiTTi benchmark"**
  - 3D pointclouds
  - cars, pedestrians, bikes, trucks, etc





The end

Thank you!

Questions?