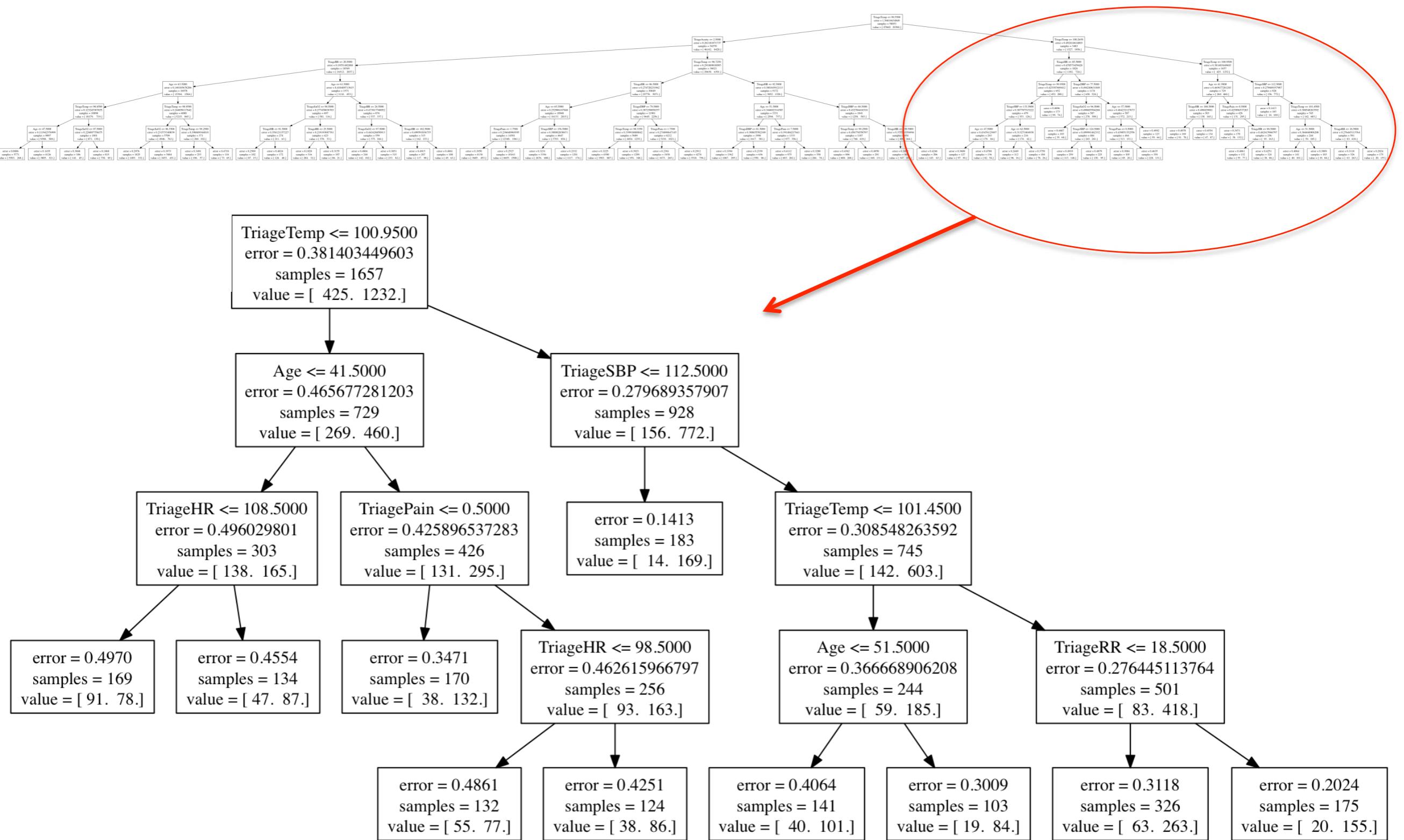


# BBM406

## Fundamentals of Machine Learning

### Lecture 19: What is Ensemble Learning? Bagging Random Forests

# Last time... Decision Trees



# Last time... Information Gain

- Decrease in entropy (uncertainty) after splitting

$$IG(X) = H(Y) - H(Y | X)$$

In our running example:

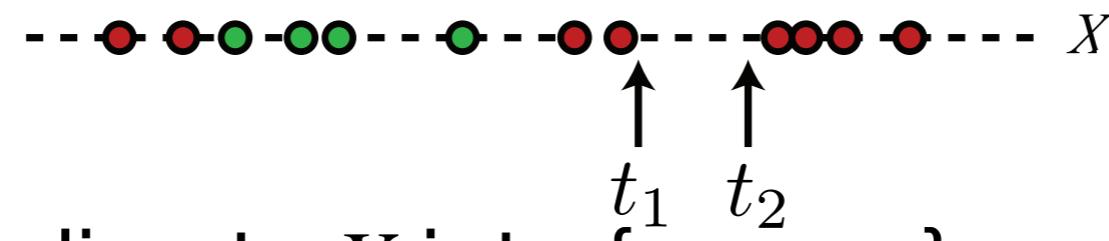
$$\begin{aligned} IG(X_1) &= H(Y) - H(Y|X_1) \\ &= 0.65 - 0.33 \end{aligned}$$

$IG(X_1) > 0 \rightarrow$  we prefer the split!

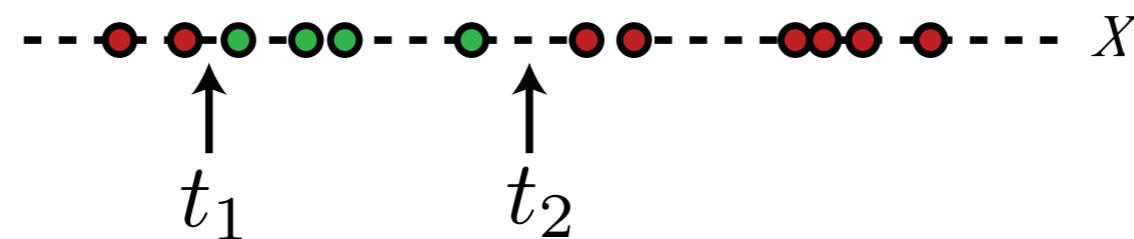
$X_1$	$X_2$	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

# Last time... Continuous features

- Binary tree, split on attribute  $X$ 
  - One branch:  $X < t$
  - Other branch:  $X \geq t$
- Search through possible values of  $t$ 
  - Seems hard!!!
- But only a finite number of  $t$ 's are important:



- Sort data according to  $X$  into  $\{x_1, \dots, x_m\}$
- Consider split points of the form  $x_i + (x_{i+1} - x_i)/2$
- Moreover, only splits between examples from different classes matter!



# Last time... Decision trees will overfit

- Standard decision trees have no learning bias
  - Training set error is always zero!
    - (If there is no label noise)
  - Lots of variance
  - Must introduce some bias towards simpler trees
- Many strategies for picking simpler trees
  - Fixed depth
  - Fixed number of leaves
- Random forests

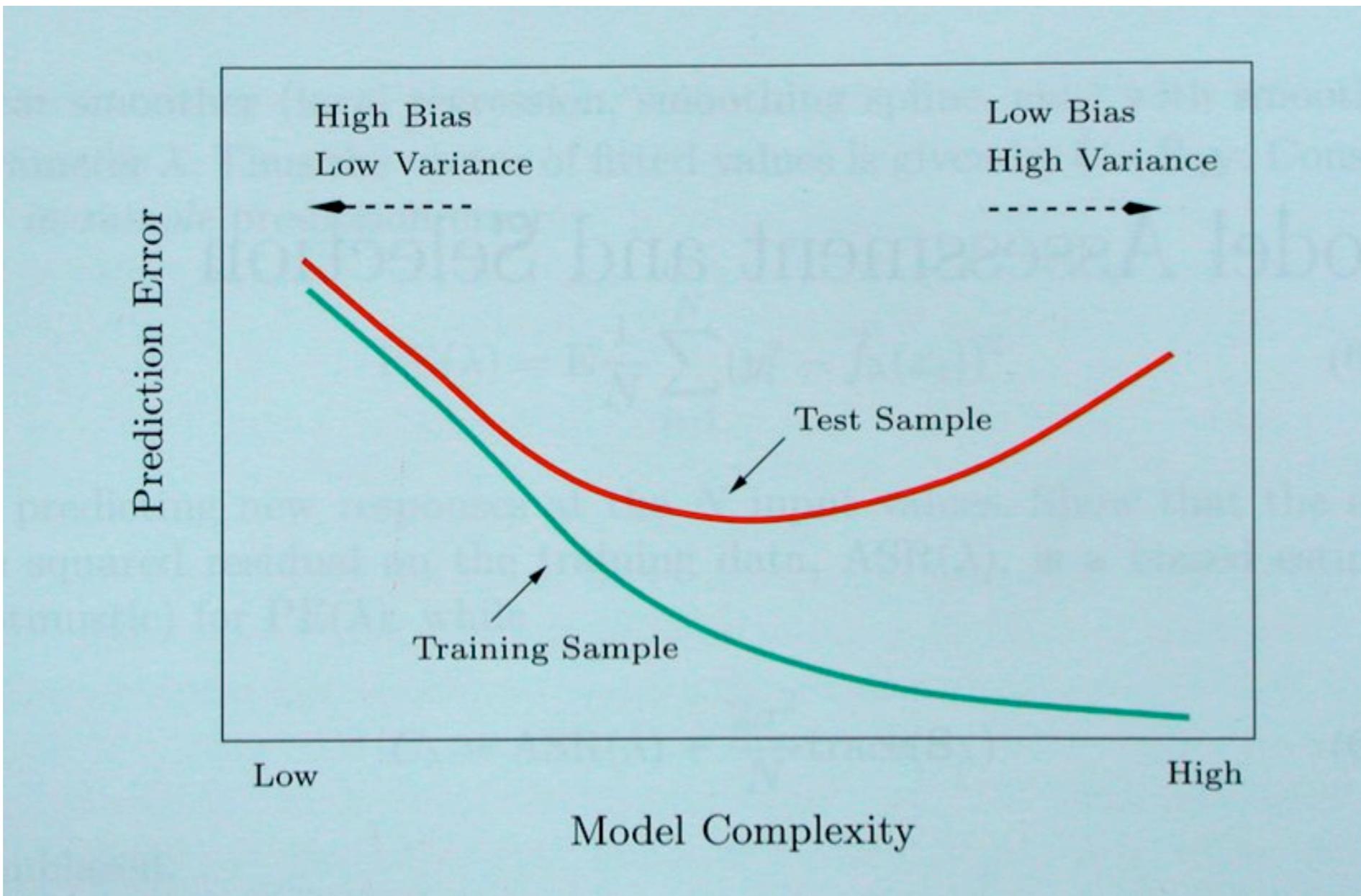
# Today

- Ensemble Methods
  - Bagging
    - Random Forests

# Ensemble Methods

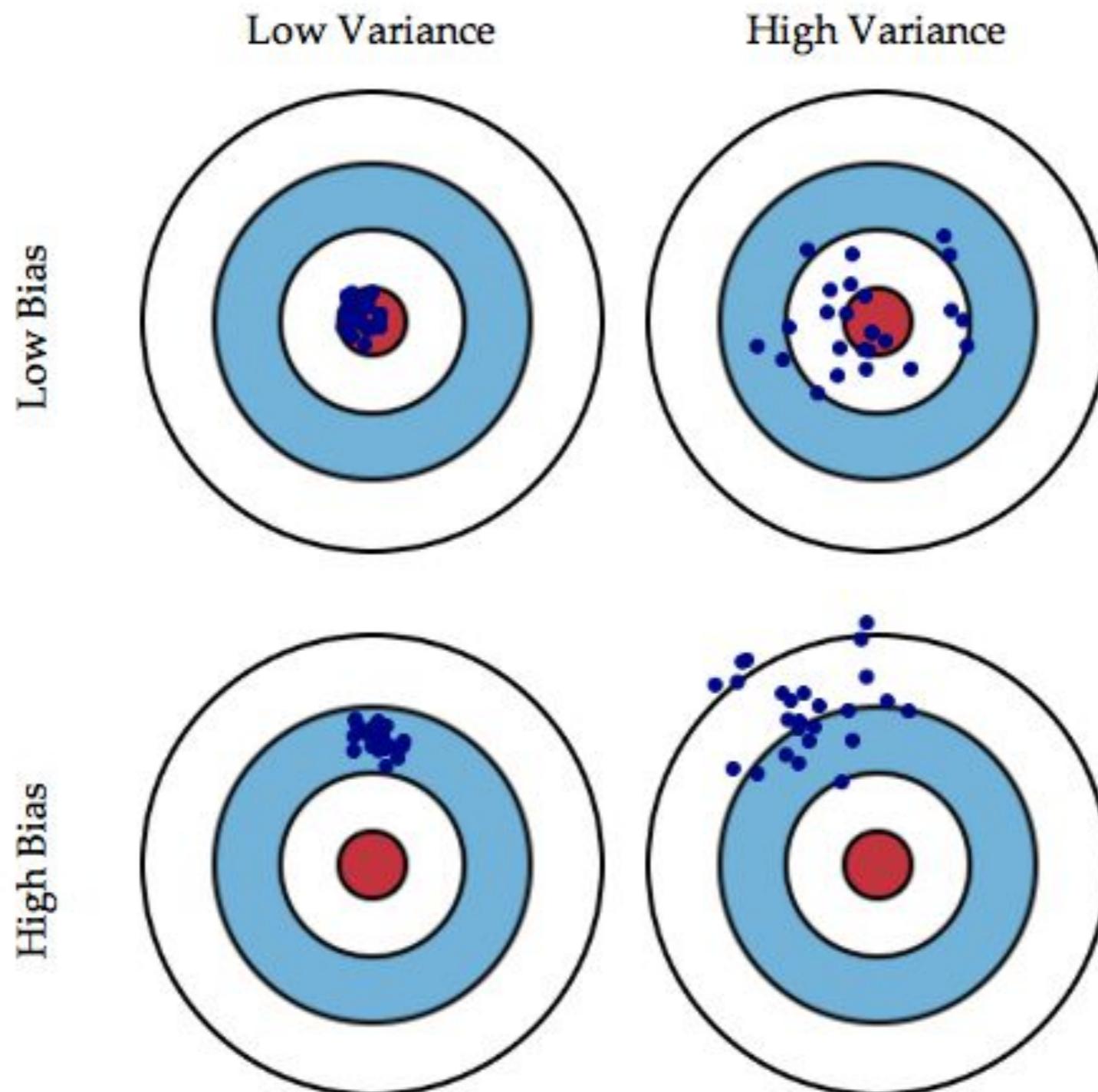
- High level idea
  - Generate multiple hypotheses
  - Combine them to a single classifier
- Two important questions
  - How do we generate multiple hypotheses
    - we have only one sample
  - How do we combine the multiple hypotheses
    - Majority, AdaBoost, ...

# Bias/Variance Tradeoff



Hastie, Tibshirani, Friedman “Elements of Statistical Learning” 2001

# Bias/Variance Tradeoff



Graphical illustration of bias and variance.

<http://scott.fortmann-roe.com/docs/BiasVariance.html>

# Fighting the bias-variance tradeoff

- **Simple (a.k.a. weak) learners are good**
  - e.g., naïve Bayes, logistic regression, decision stumps (or shallow decision trees)
  - Low variance, don't usually overfit
- **Simple (a.k.a. weak) learners are bad**
  - High bias, can't solve hard learning problems

# Reduce Variance Without Increasing Bias

- **Averaging** reduces variance:

$$Var(\bar{X}) = \frac{Var(X)}{N} \quad (\text{when prediction are independent})$$

- Average models to reduce model variance
- One problem:
  - Only one training set
  - Where do multiple models come from?

# Bagging (Bootstrap Aggregating)

- Leo Breiman (1994)
- Take repeated **bootstrap samples** from training set D.
- **Bootstrap sampling:** Given set D containing N training examples, create D' by drawing N examples at random **with replacement** from D.
- Bagging:
  - Create  $k$  bootstrap samples  $D_1 \dots D_k$ .
  - Train distinct classifier on each  $D_i$ .
  - Classify new instance by majority vote / average.

# Bagging

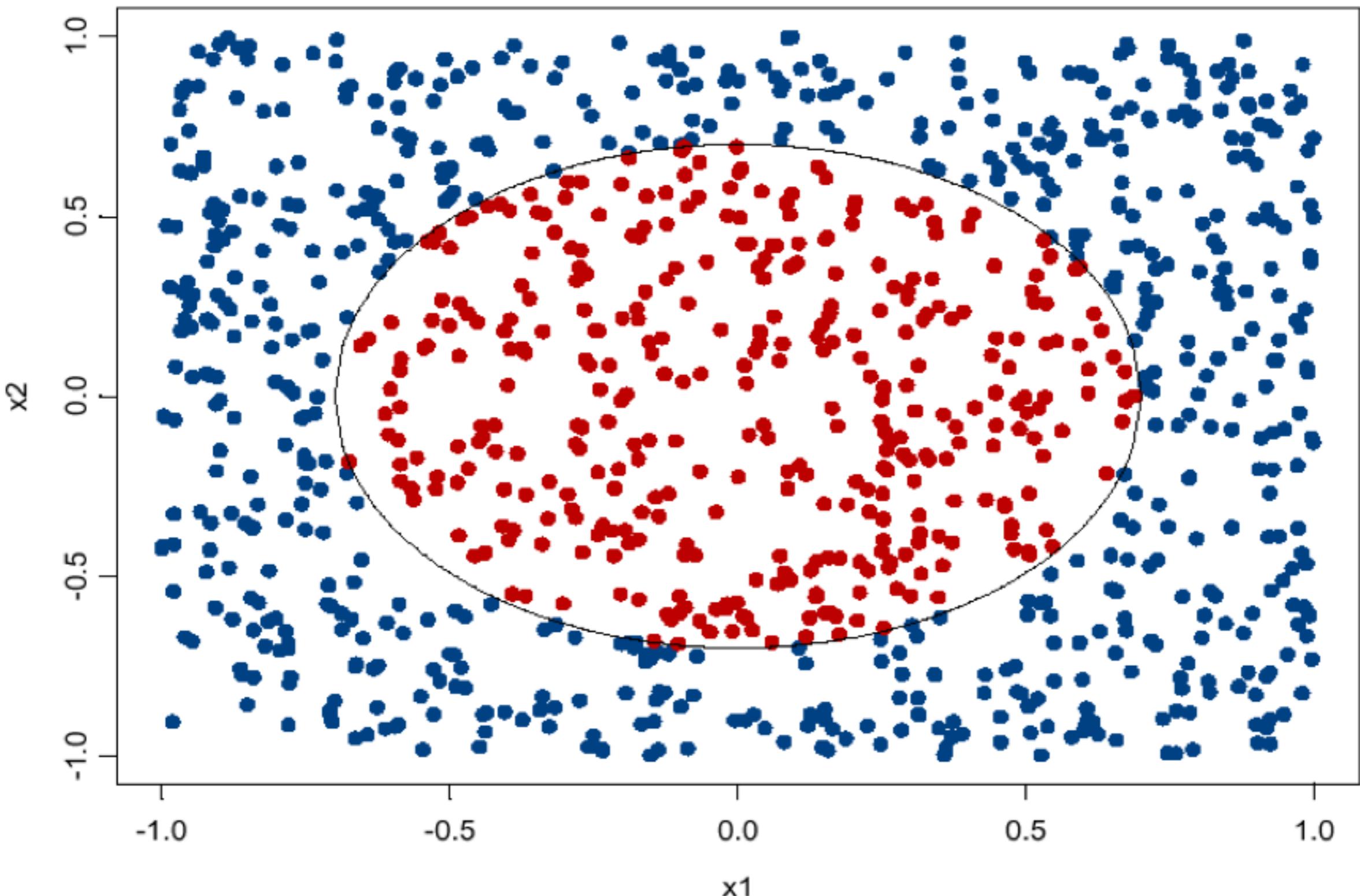
- Best case:

$$Var(Bagging(L(x, D))) = \frac{Var(L(x, D))}{N}$$

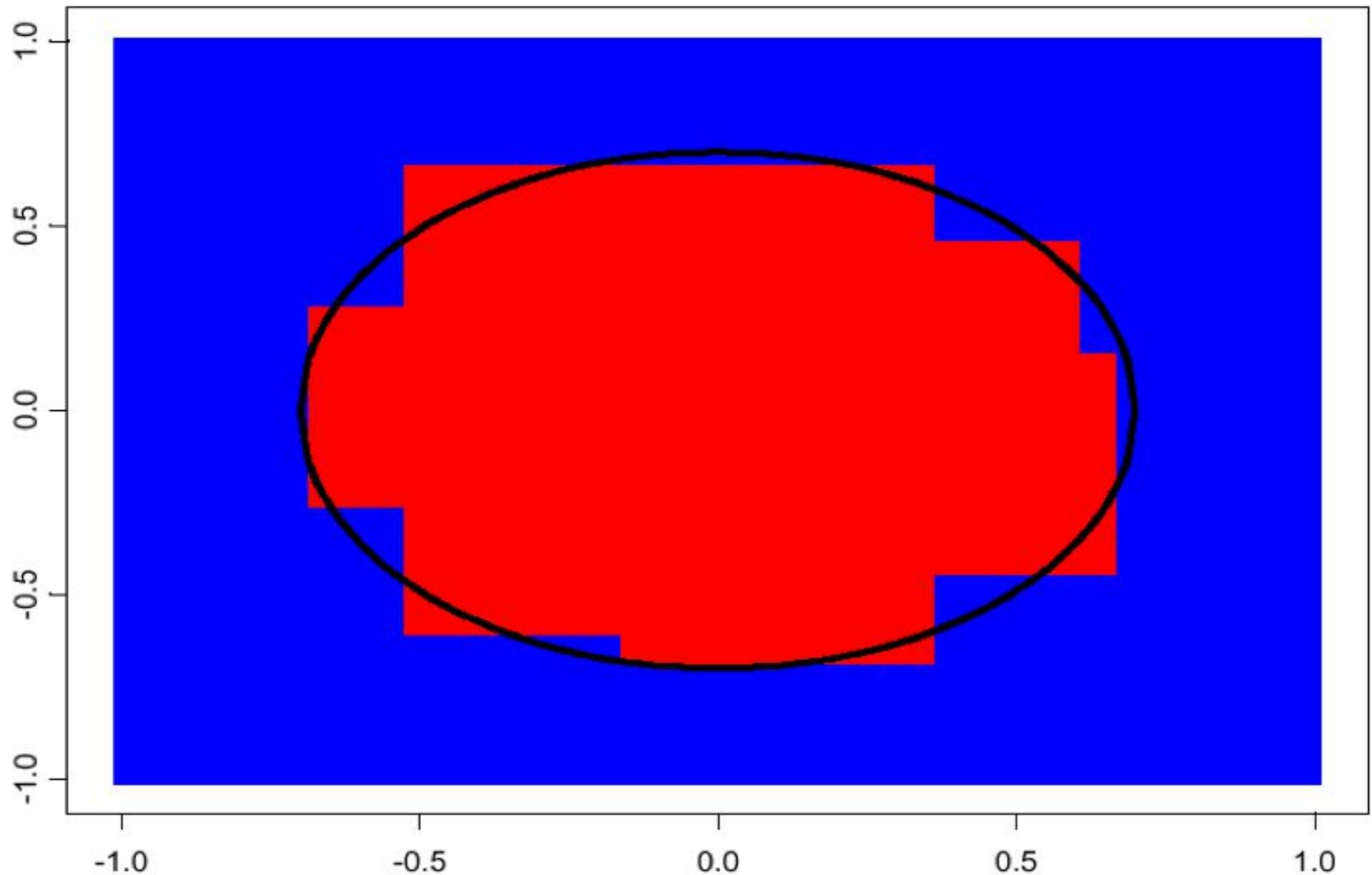
- In practice:

- models are correlated, so reduction is smaller than  $1/N$
- variance of models trained on fewer training cases usually somewhat larger

# Bagging Example

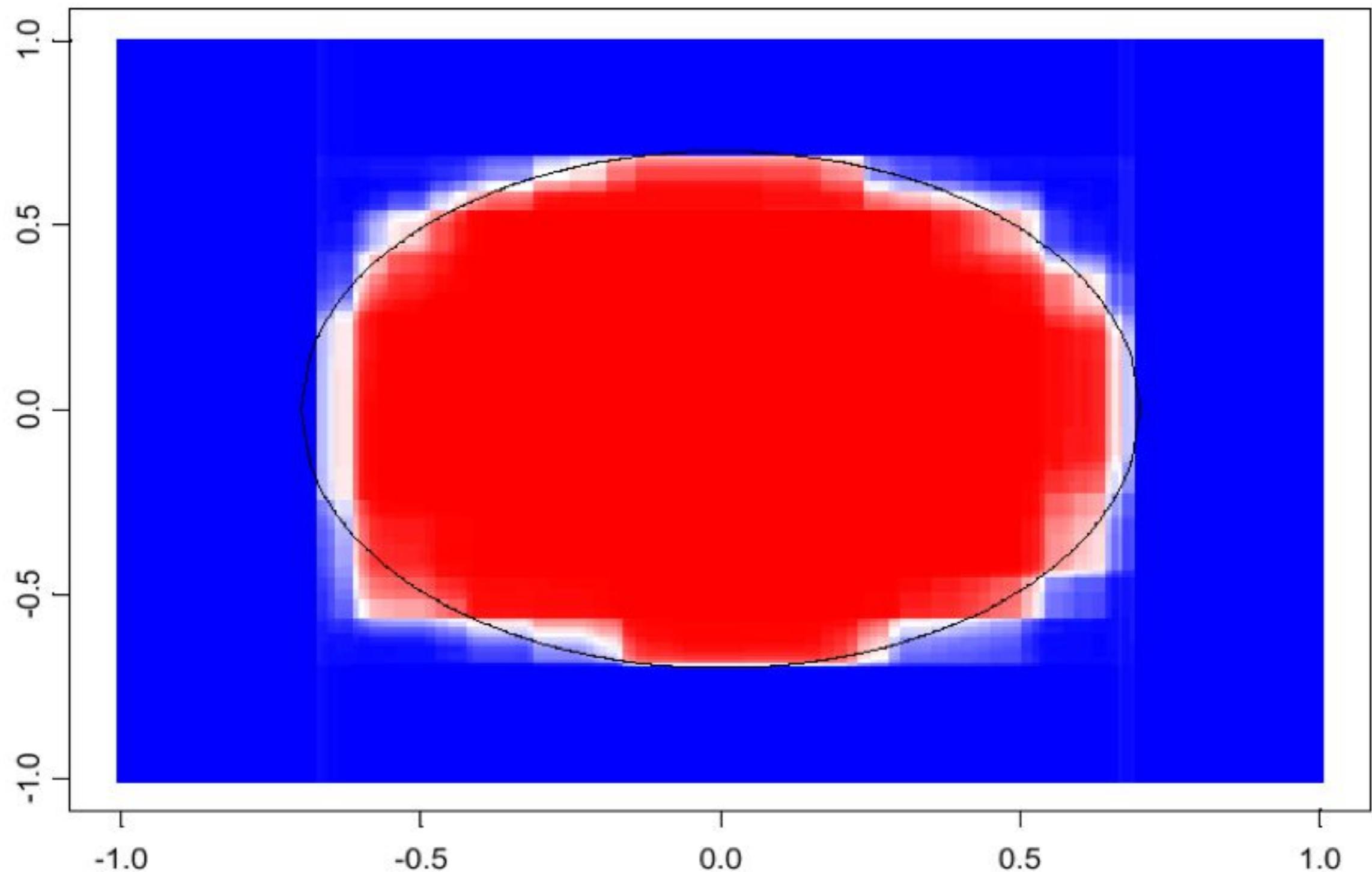


# CART\* decision boundary



\* A decision tree learning algorithm; very similar to ID3

# 100 bagged trees



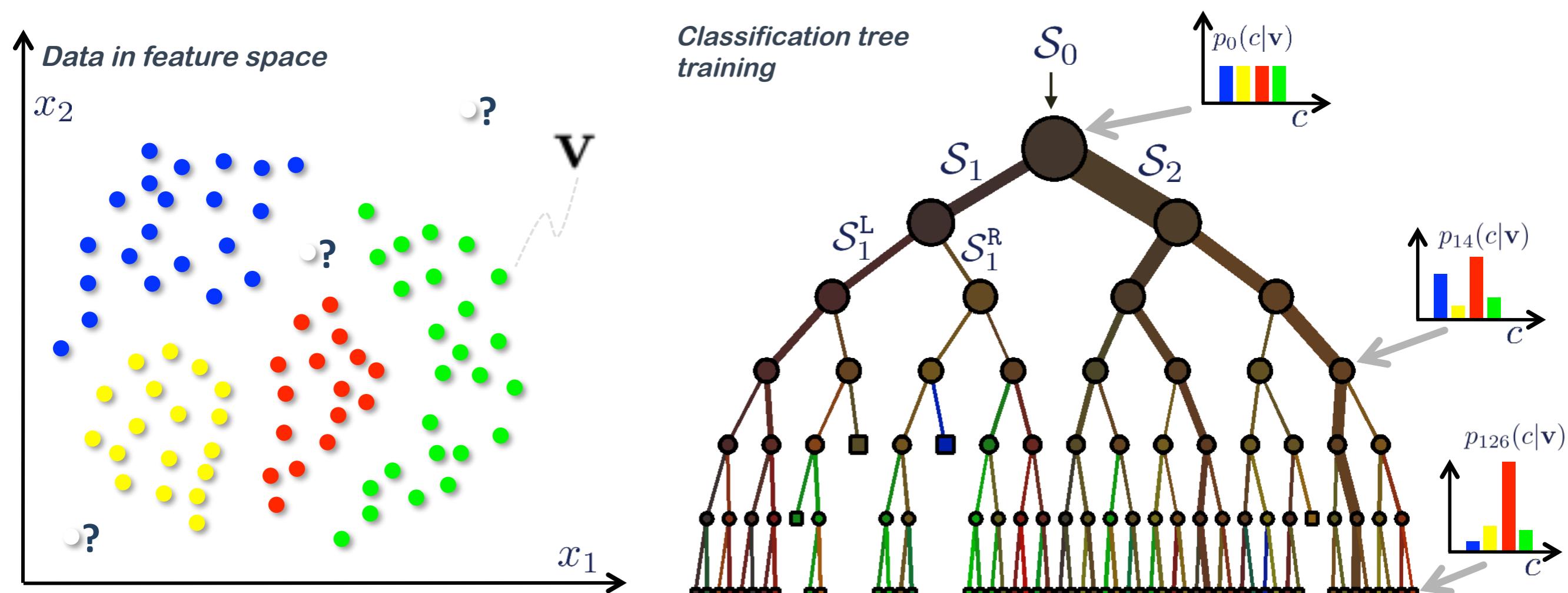
- Shades of blue/red indicate strength of vote for particular classification

# Random Forests

# Random Forests

- Ensemble method specifically designed for decision tree classifiers
- Introduce two sources of randomness: “Bagging” and “Random input vectors”
  - **Bagging method:** each tree is grown using a bootstrap sample of training data
  - **Random vector method:** **At each node**, best split is chosen from a random sample of  $m$  attributes instead of all attributes

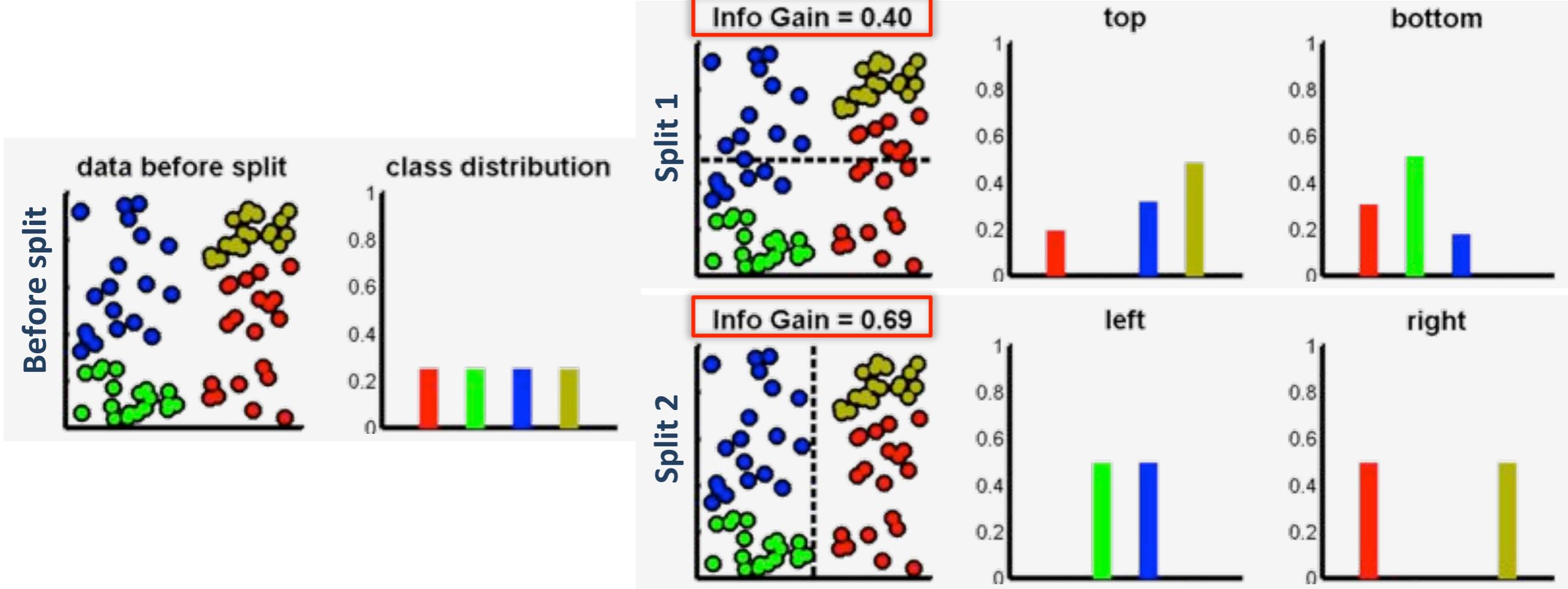
# Classification tree



A generic data point is denoted by a vector  $\mathbf{v} = (x_1, x_2, \dots, x_d)$

$$\mathcal{S}_j = \mathcal{S}_j^L \cup \mathcal{S}_j^R$$

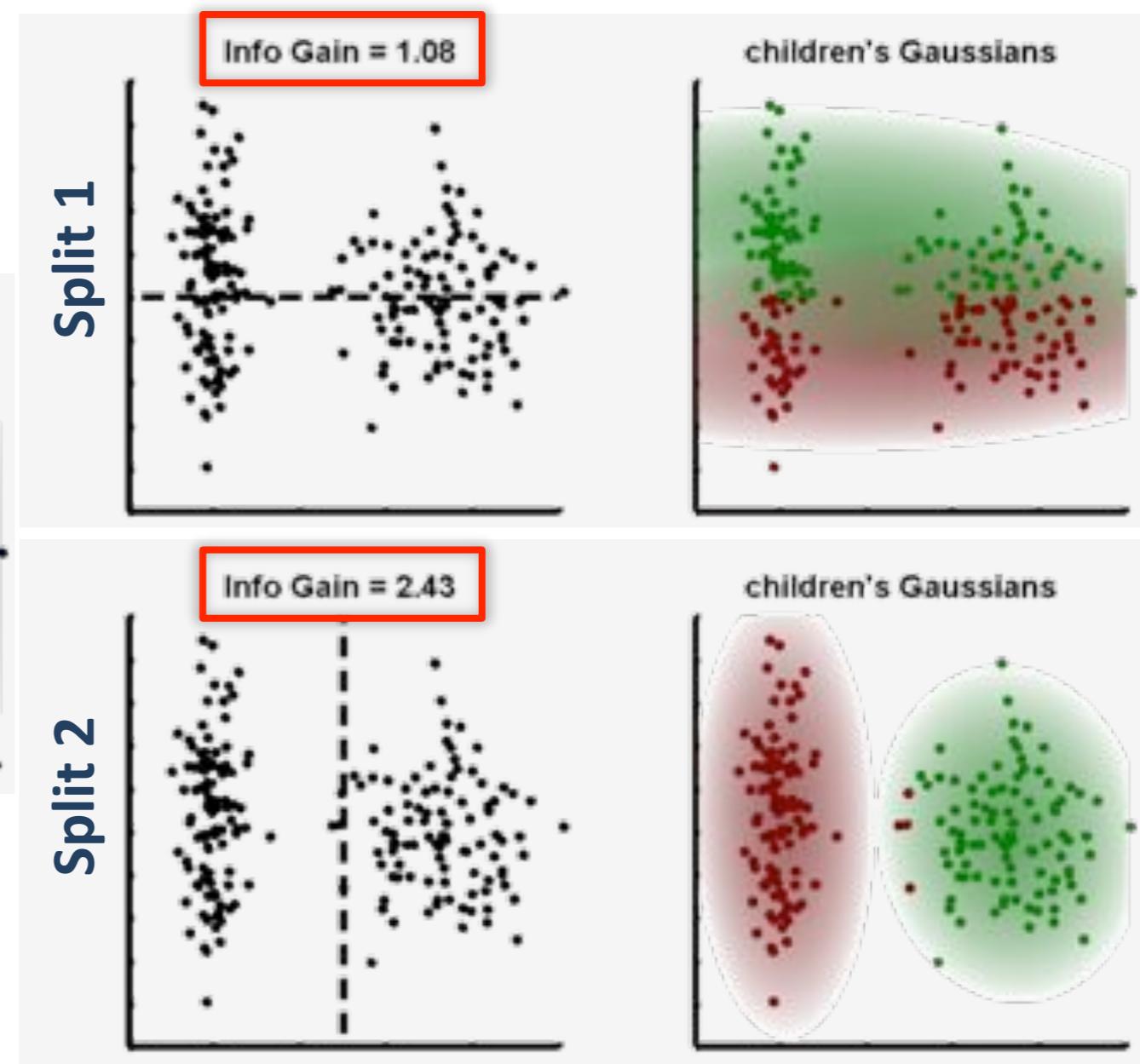
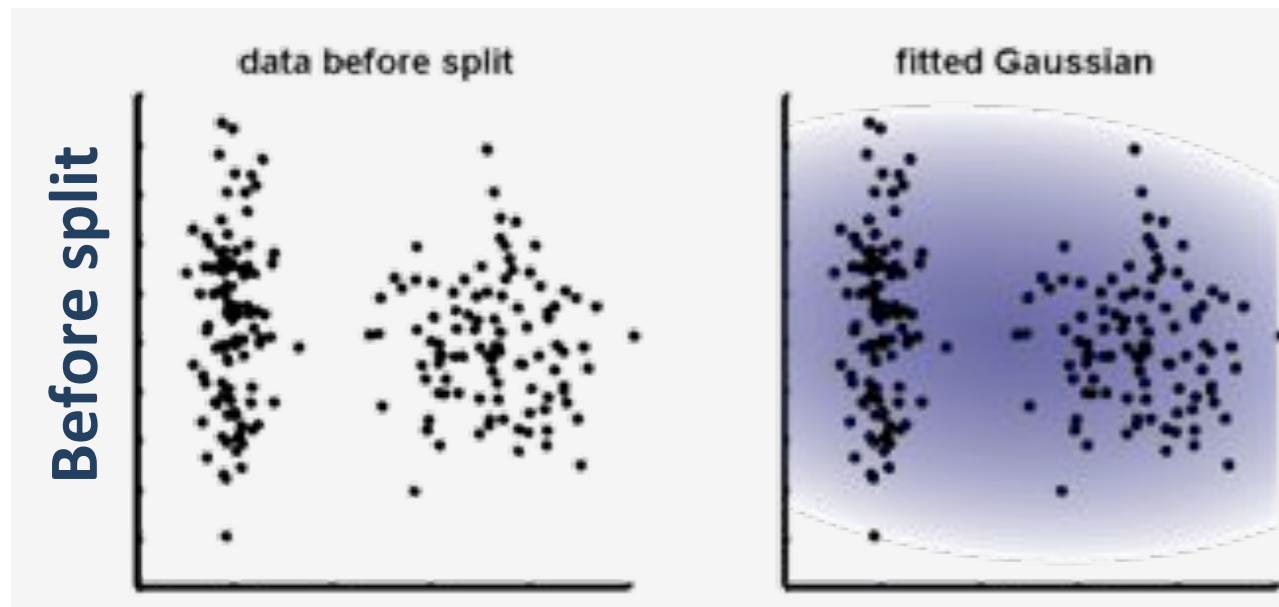
# Use information gain to decide splits



$$I_j = H(\mathcal{S}_j) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i)$$

[Criminisi et al., 2011] 20

# Advanced: Gaussian information gain to decide splits

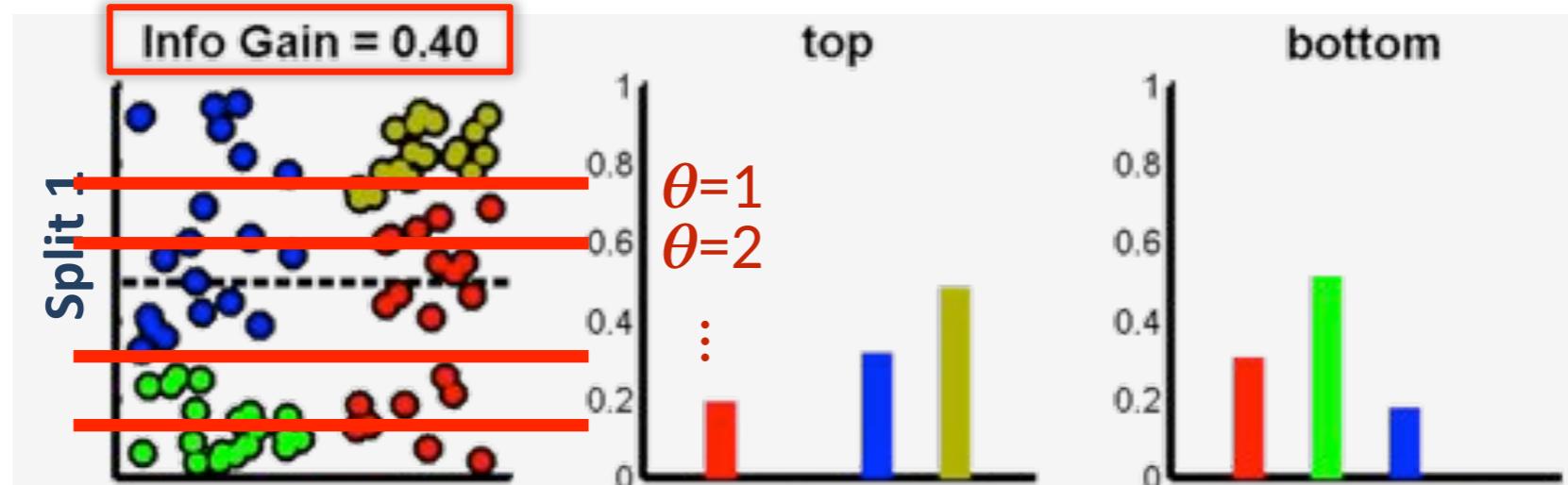


$$H(\mathcal{S}) = \frac{1}{2} \log \left( (2\pi e)^d |\Lambda(\mathcal{S})| \right)$$

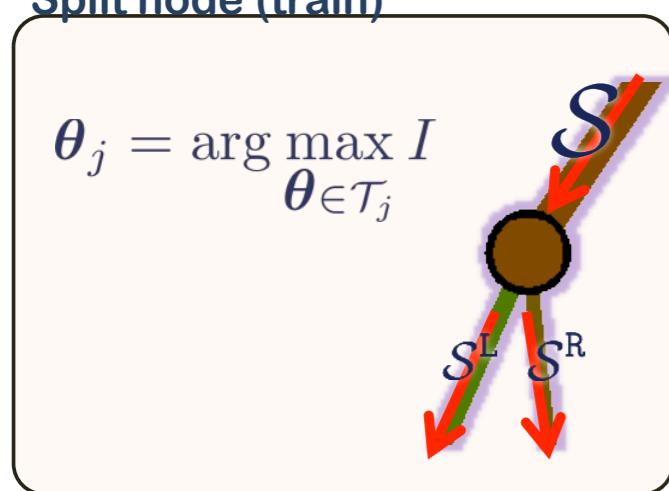
[Criminisi et al., 2011] 21

Each split node  $j$  is associated with a binary split function

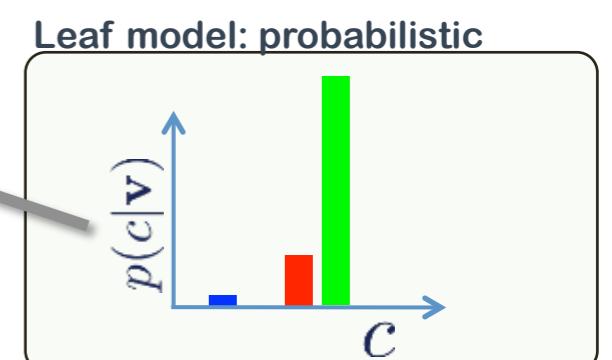
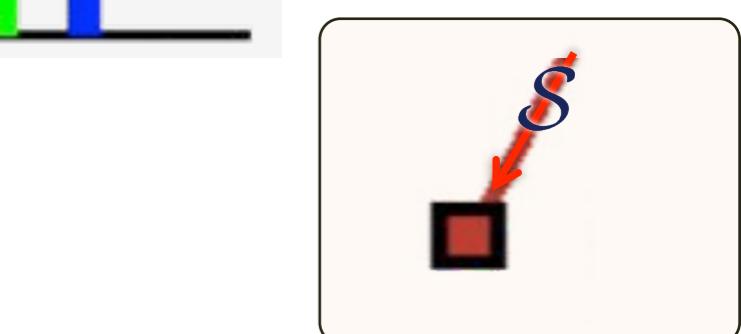
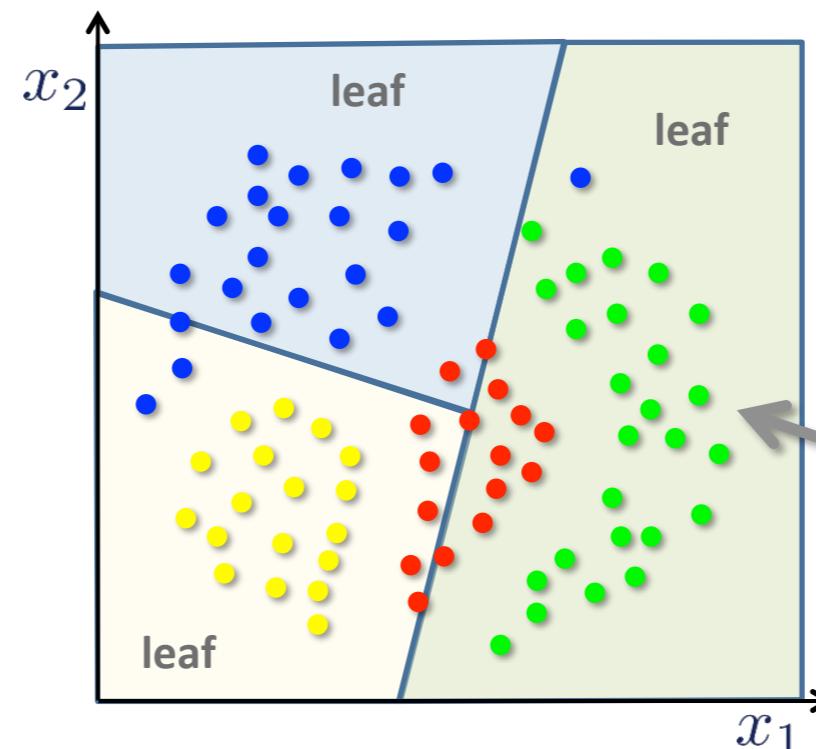
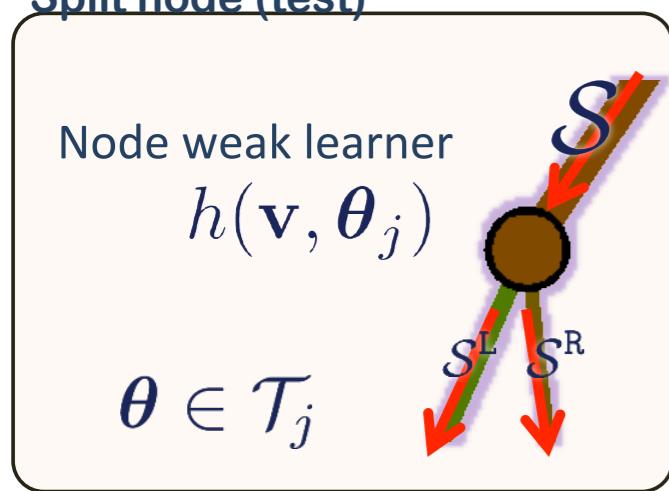
$$h(\mathbf{v}, \theta) \in \{\text{true, false}\}$$



Split node (train)



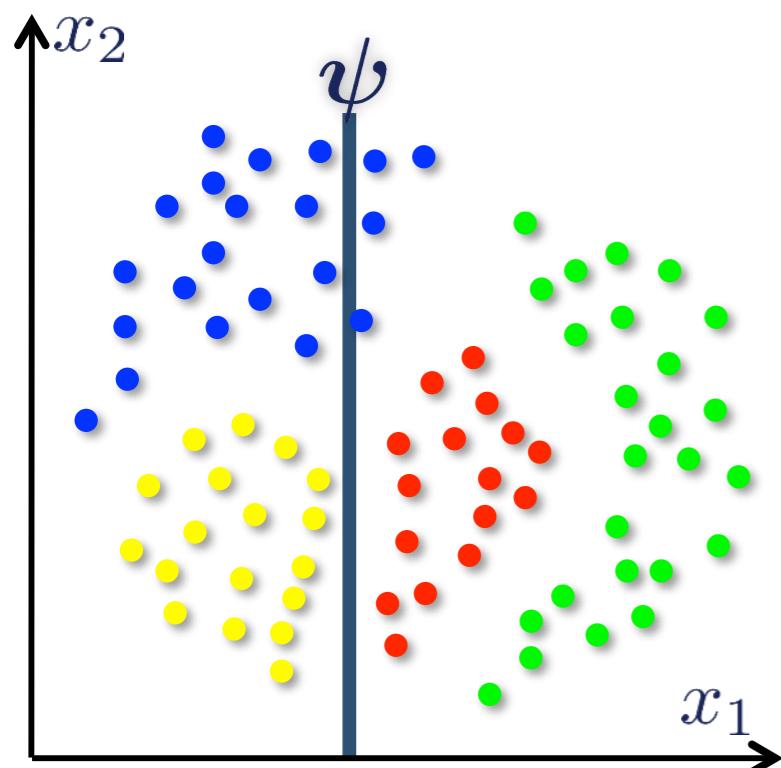
Split node (test)



$$I_j = H(\mathcal{S}_j) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i)$$

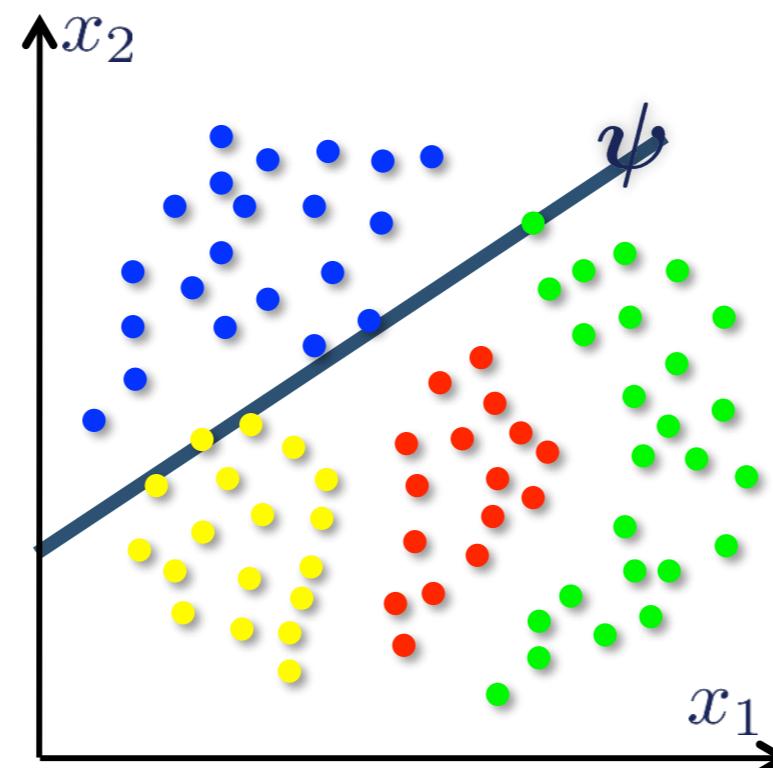
# Alternative node decisions

$$\mathbf{v} = (x_1 \ x_2) \in \mathbb{R}^2 \quad \phi(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$$



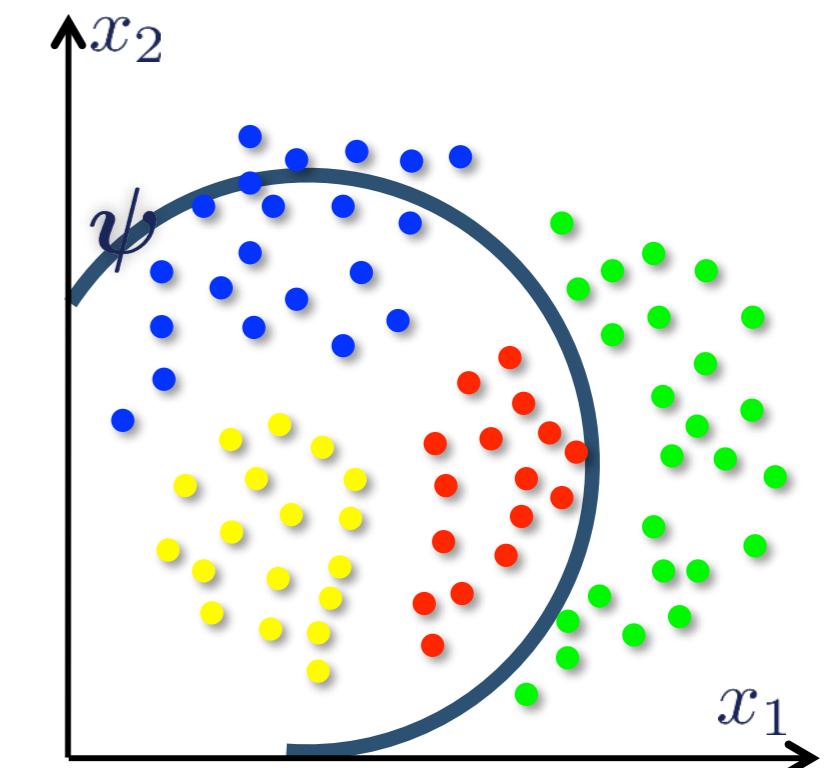
$$h(\mathbf{v}, \theta) = [\tau_1 > \phi(\mathbf{v}) \cdot \psi > \tau_2]$$

axis aligned



$$h(\mathbf{v}, \theta) = [\tau_1 > \phi(\mathbf{v}) \cdot \psi > \tau_2]$$

oriented line



$$h(\mathbf{v}, \theta) = [\tau_1 > \phi^\top(\mathbf{v}) \psi \phi(\mathbf{v}) > \tau_2]$$

conic section

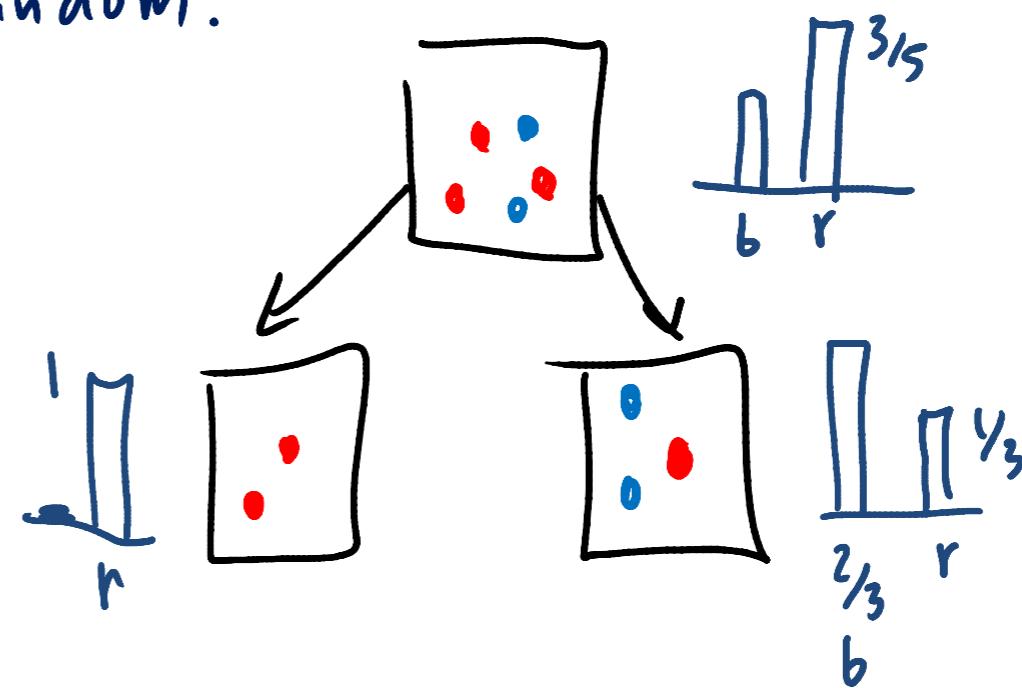
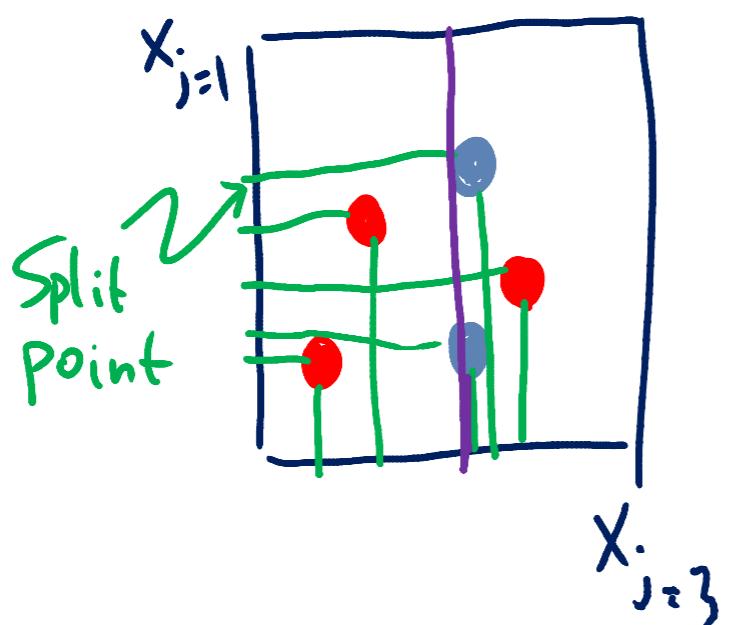
examples of weak learners

# Building a random tree

$d = 3$  features  
 $n = 5$  data

$$X = \begin{bmatrix} i=1 & i=2 & & & i=n=5 \\ j=1 & 1 & 3 & 0 & 8 \\ j=2 & 0 & 6 & 2 & 9 \\ j=3 & 2 & 1 & 4 & 0 \end{bmatrix}$$
$$Y = \begin{bmatrix} i=1 & i=2 & & i=5 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Pick 2 features at random.



# Random Forests algorithm

1. For  $b = 1$  to  $B$ :
  - (a) Draw a bootstrap sample  $\mathbf{Z}^*$  of size  $N$  from the training data.
  - (b) Grow a random-forest tree  $T_b$  to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size  $n_{min}$  is reached.
    - i. Select  $m$  variables at random from the  $p$  variables.
    - ii. Pick the best variable/split-point among the  $m$ .
    - iii. Split the node into two daughter nodes.
2. Output the ensemble of trees  $\{T_b\}_1^B$ .

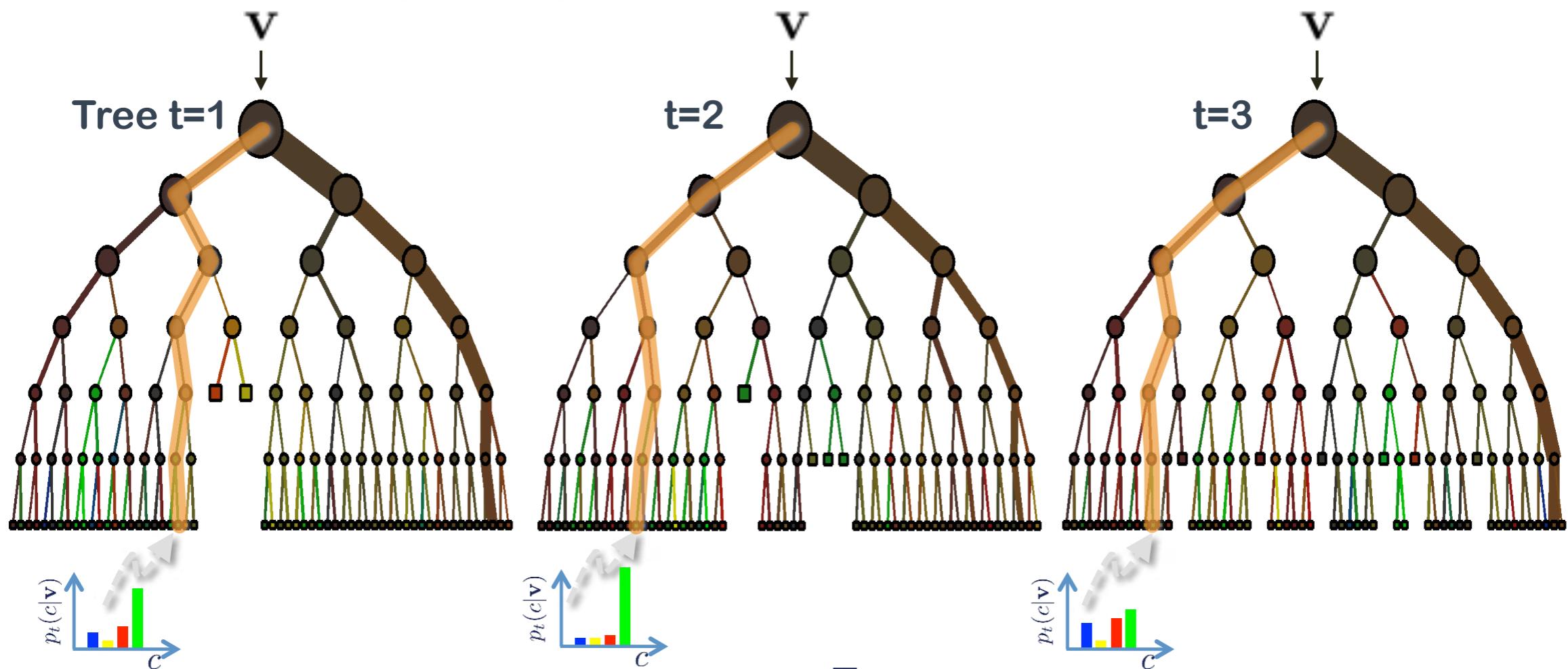
# Randomization

**Randomized node optimization.** If  $\mathcal{T}$  is the entire set of all possible parameters  $\theta$  then when training the  $j^{\text{th}}$  node we only make available a small subset  $\mathcal{T}_j \subset \mathcal{T}$  of such values.

$$\theta_j^* = \arg \max_{\theta_j \in \mathcal{T}_j} I_j.$$

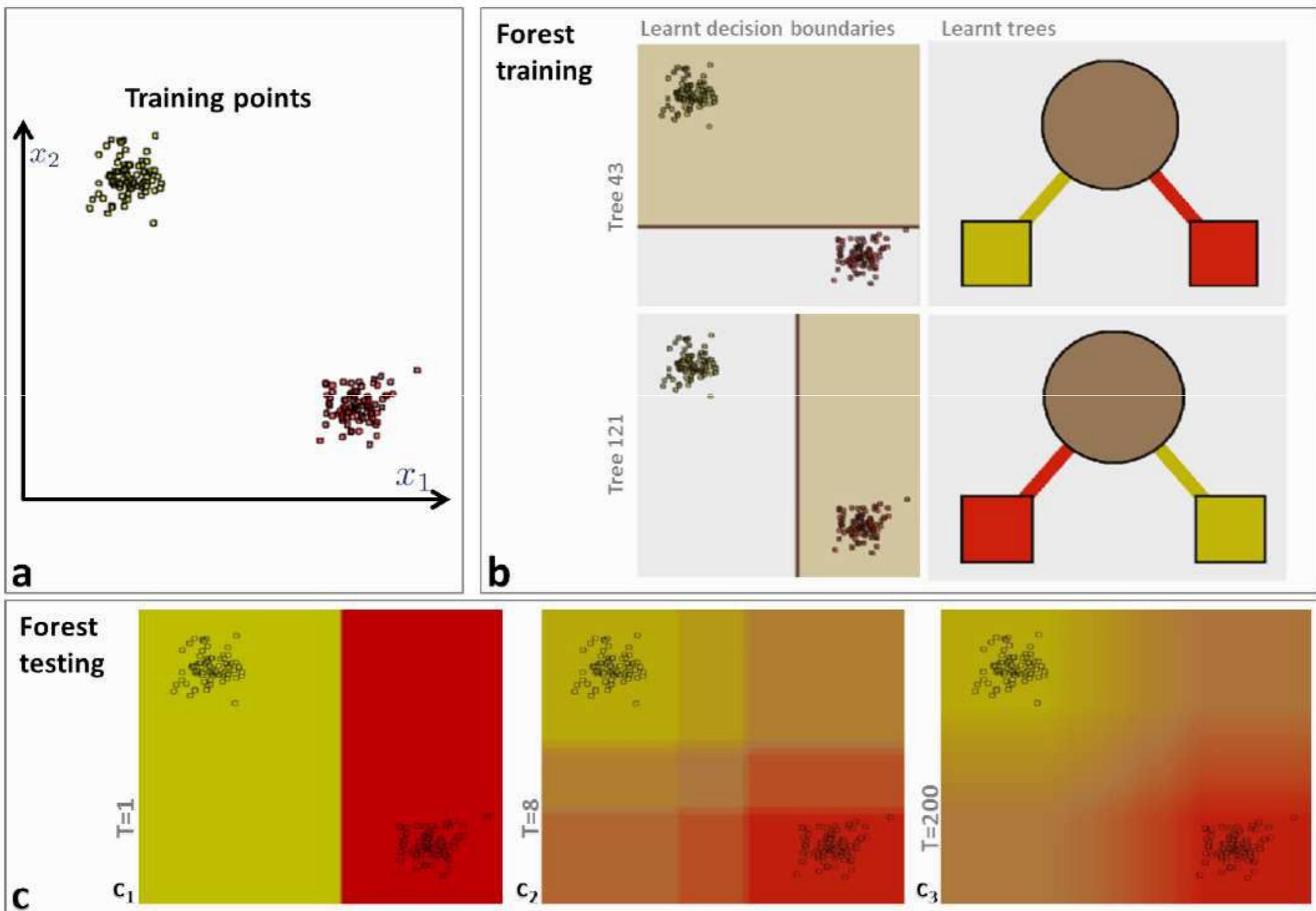
# Building a forest (ensemble)

In a forest with  $T$  trees we have  $t \in \{1, \dots, T\}$ . All trees are trained independently (and possibly in parallel). During testing, each test point  $\mathbf{v}$  is simultaneously pushed through all trees (starting at the root) until it reaches the corresponding leaves.



$$p(c|\mathbf{v}) = \frac{1}{T} \sum_t^T p_t(c|\mathbf{v})$$

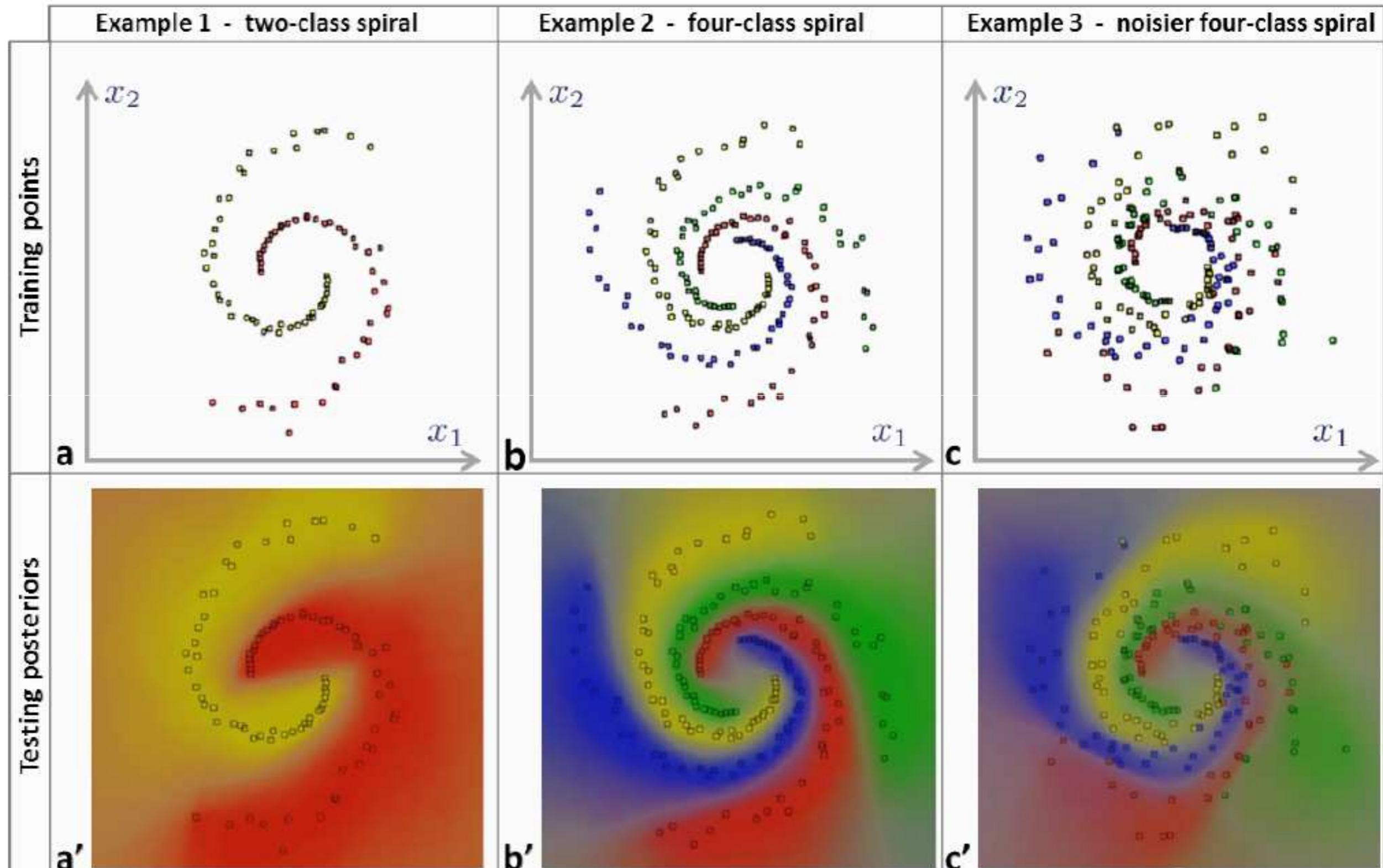
# Effect of forest size



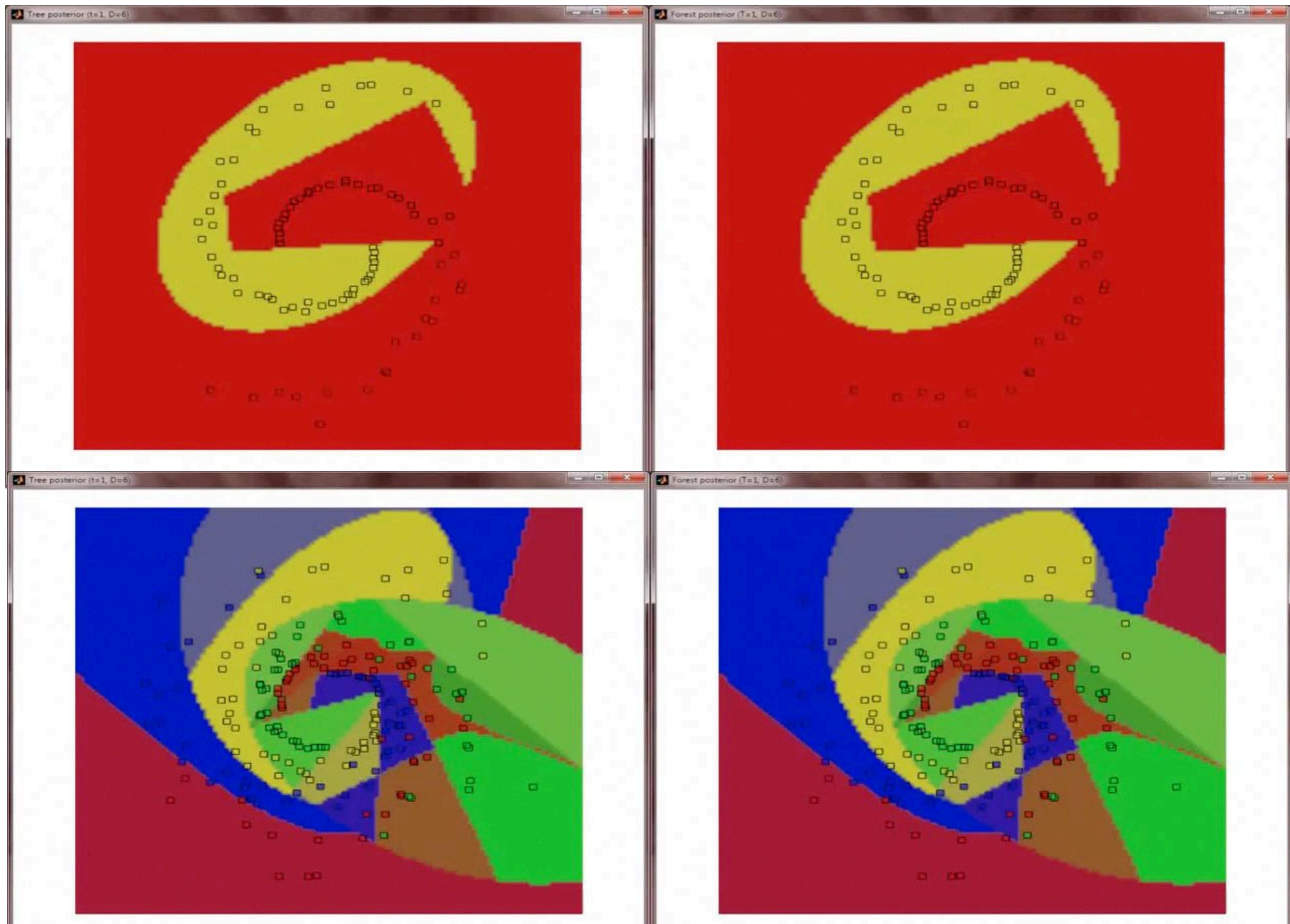
# Effect of forest size



# Effect of more classes and noise

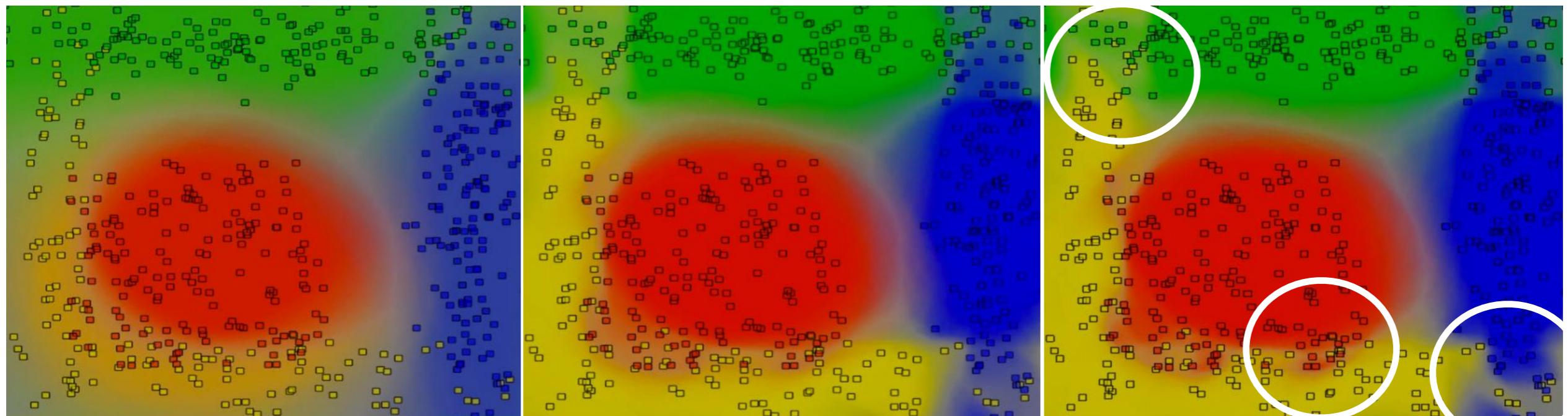
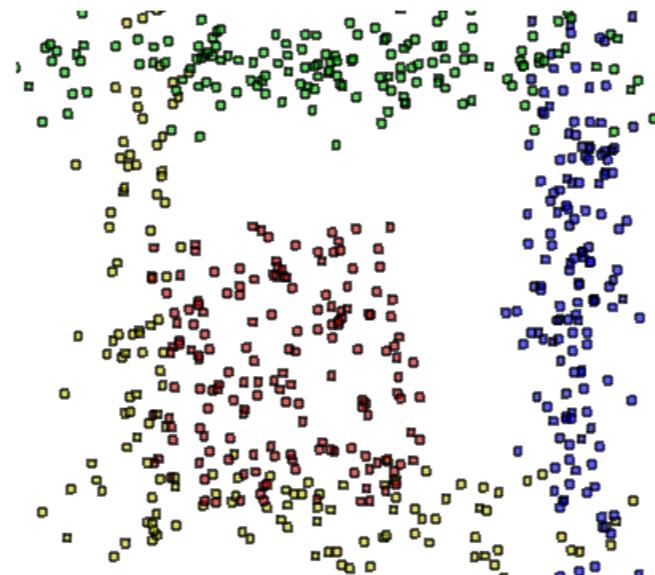


# Effect of more classes and noise



# Effect of tree depth (D)

Training points: 4-class mixed

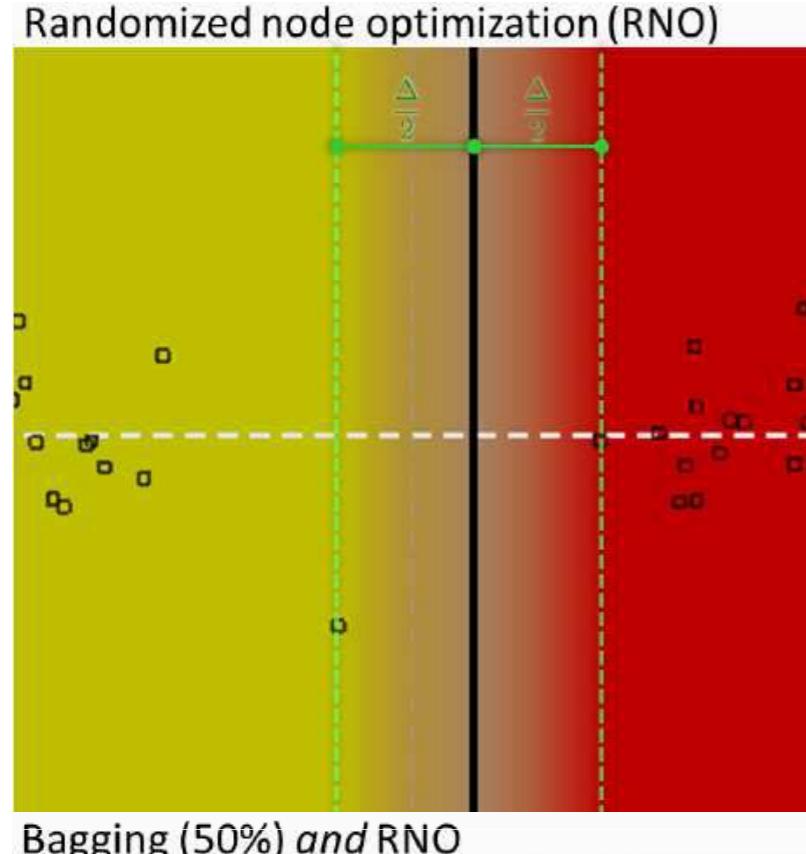


D=3  
(underfitting)

D=6

D=15  
(overfitting)

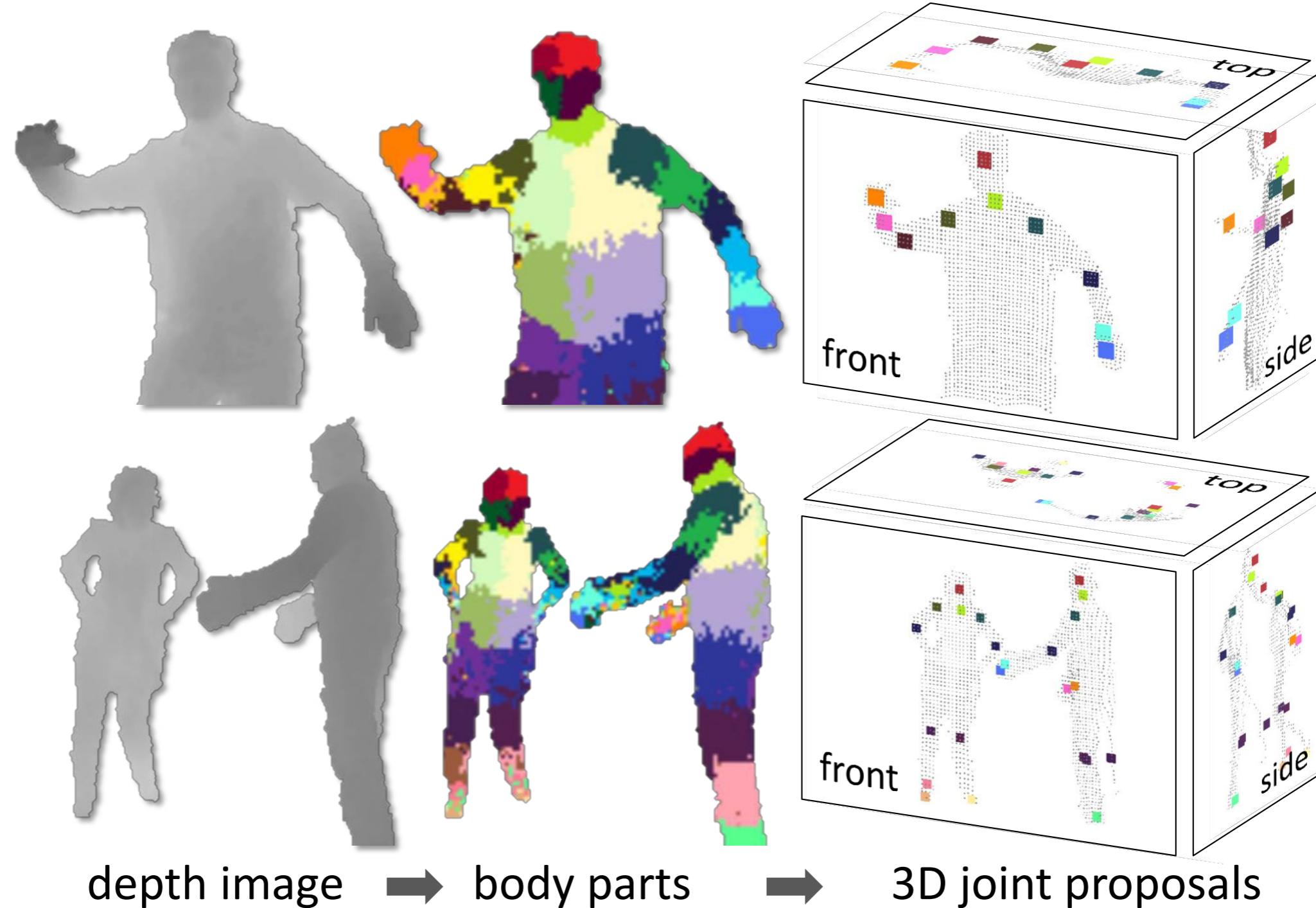
# Effect of bagging



# Random Forests and the Kinect

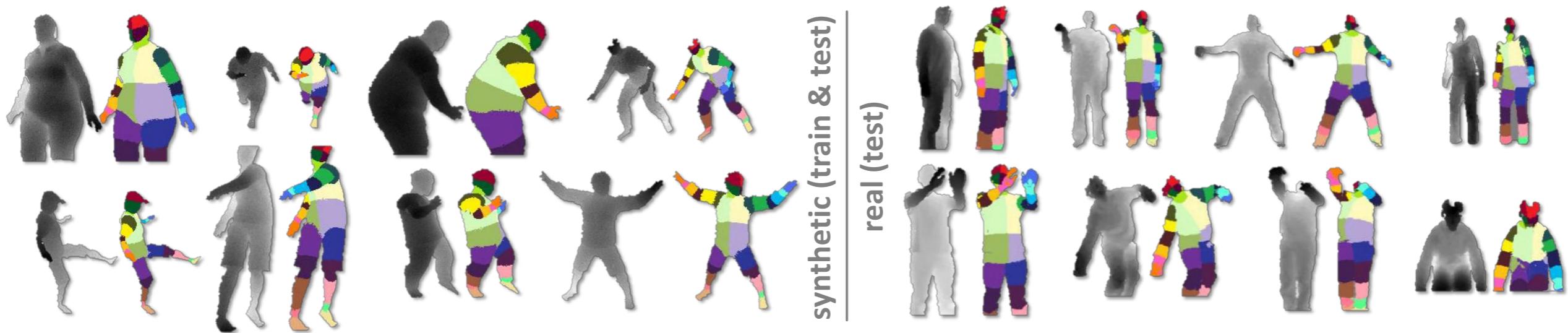


# Random Forests and the Kinect



# Random Forests and the Kinect

- Use computer graphics to generate plenty of data



Real-Time Human Pose Recognition in Parts from Single Depth Images

CVPR 2011

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, Andrew Blake  
Microsoft Research Cambridge & Xbox Incubation

# Reduce Bias<sup>2</sup> and Decrease Variance?

- Bagging reduces variance by averaging
- Bagging has little effect on bias
- Can we average *and* reduce bias?
- Yes: **Boosting**

# **Next Lecture:**

## **Boosting**