Machine Learning in Medical Imaging

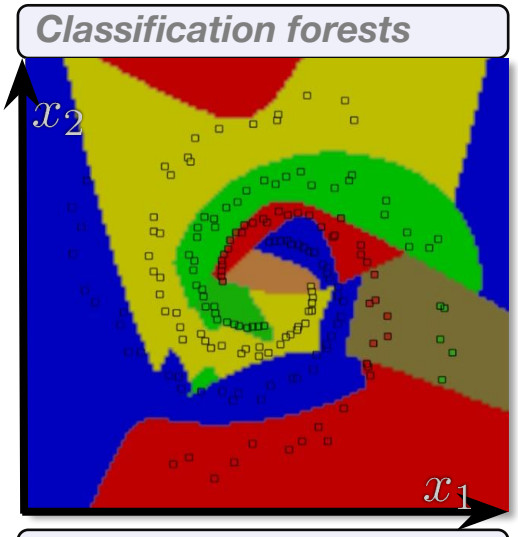
Random Forests

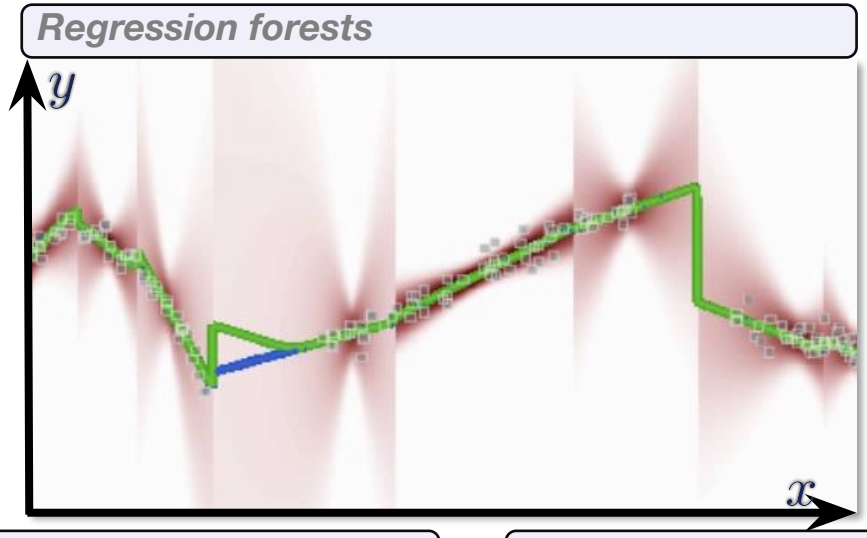
Mai Bui
Graduate Research Assistant | PhD Candidate
mai.bui@tum.de

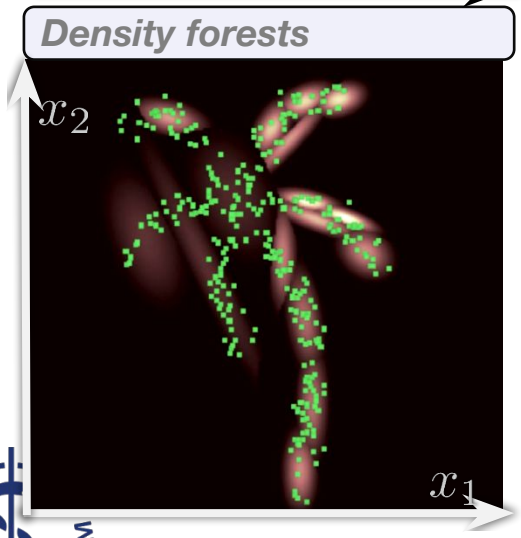
Based on the tutorial on Decision Forests from A. Criminisi, J. Shotton
https://www.microsoft.com/en-us/research/project/decision-forests/

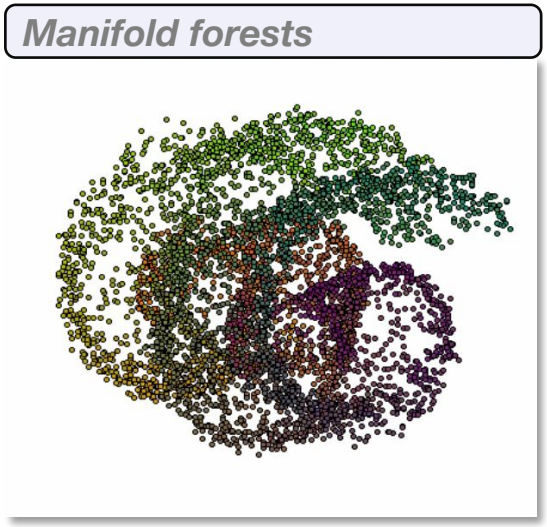
Technische Universität München
+
Johns Hopkins Whiting School of Engineering

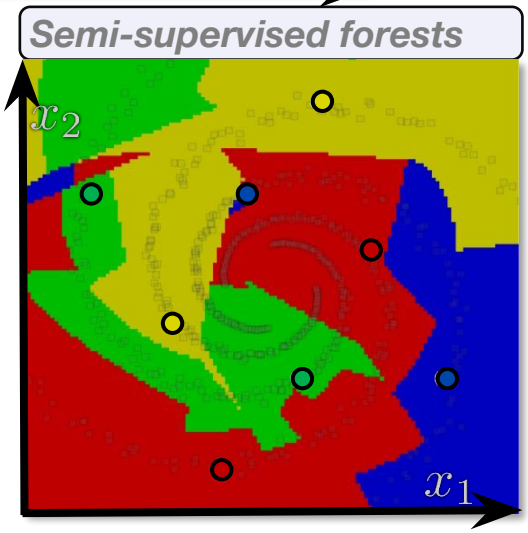
What can forests do?











Images from A. Criminisi

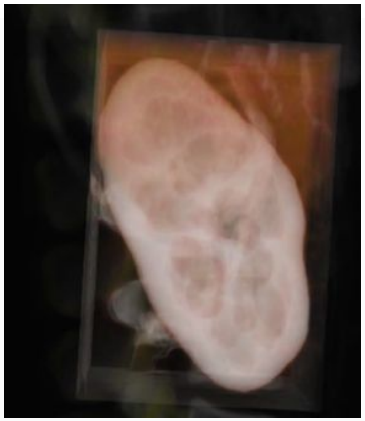
What can forests do?

Classification forests

semantic segmentation

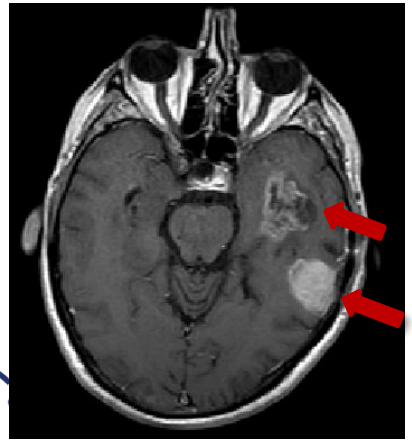
Regression forests

object localization



Density forests

novelty detection



Manifold forests

dimensionality reduction

Semi-supervised forests

semi-sup. semantic segmentation

Images from A. Criminisi

Agenda

- Generic tree and Forest model
- Classification Forests
- Regression Forests

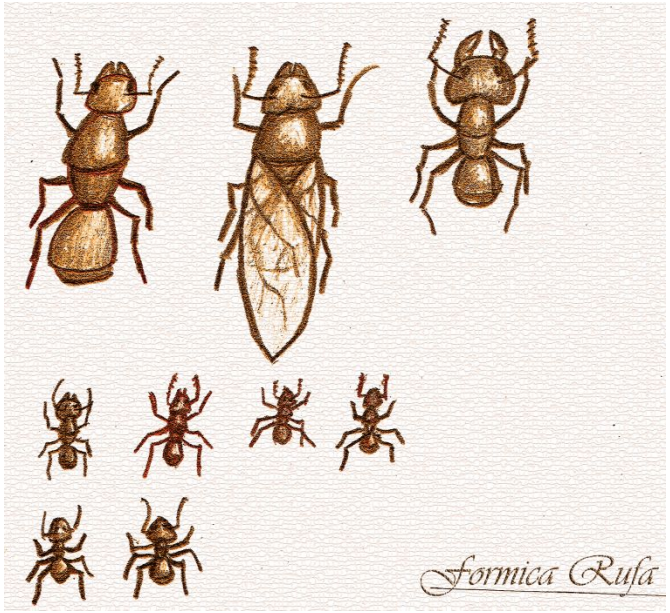
Generic Tree Model

Hierarchical learning

Given a set of observations...

Goal

(i) gain knowledge
(ii) discover similar patterns
(iii) model observations
(iv) perform categorization
(v) perform predictions

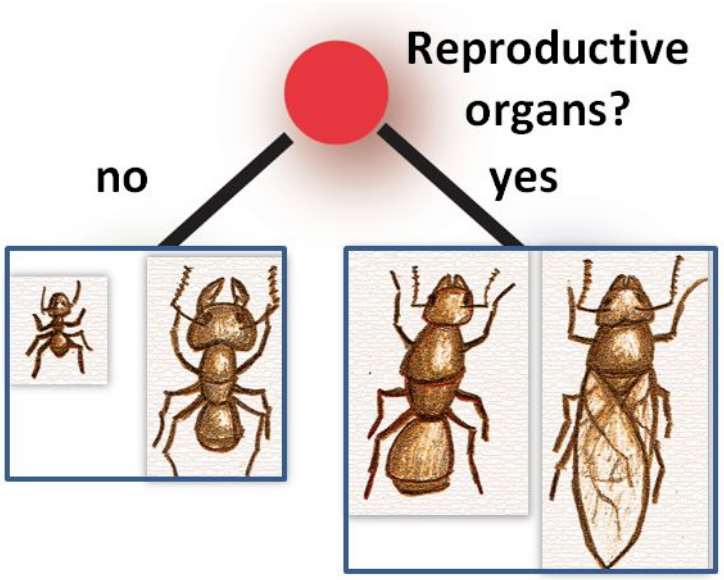


Hierarchical learning

IDEA: Use a divide and conquer strategy

(1) Divide:
Subdivide your observations using simple rules

(2) Conquer:
Model each subgroups of observations

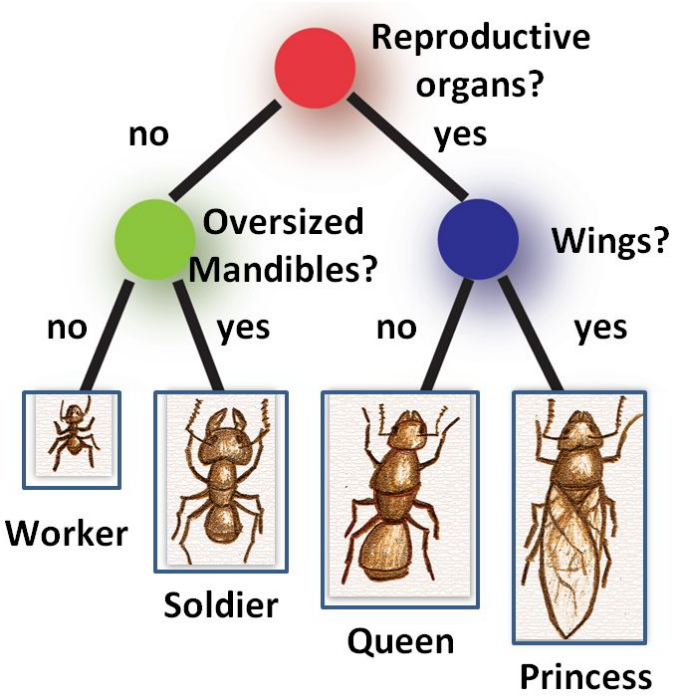


Hierarchical learning

IDEA: Use a divide and conquer strategy

(1) Divide:
Subdivide your observations using simple rules

(2) Conquer:
Model each subgroups of observations



A tree is a hierarchical learner

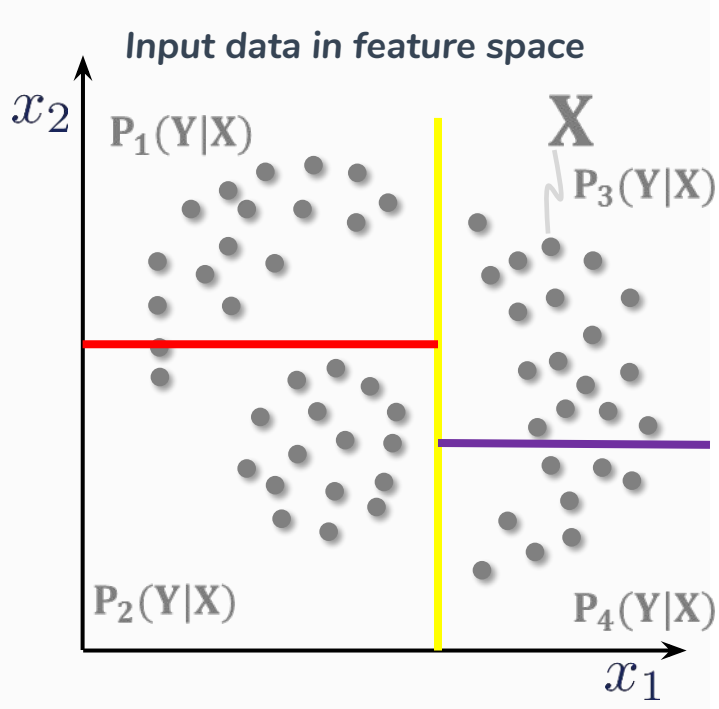
GOAL: Learn the (posterior) distribution of your observations

Divide and Conquer

(1) Partitioning
Create a partition of the feature space

(2) Modeling
Model the posterior locally in each part of this space

A tree provides a piece-wise approximation of $P(\mathbf{Y}|\mathbf{X})$



The generic tree model

TREE = directed acyclic graph

Two types of nodes

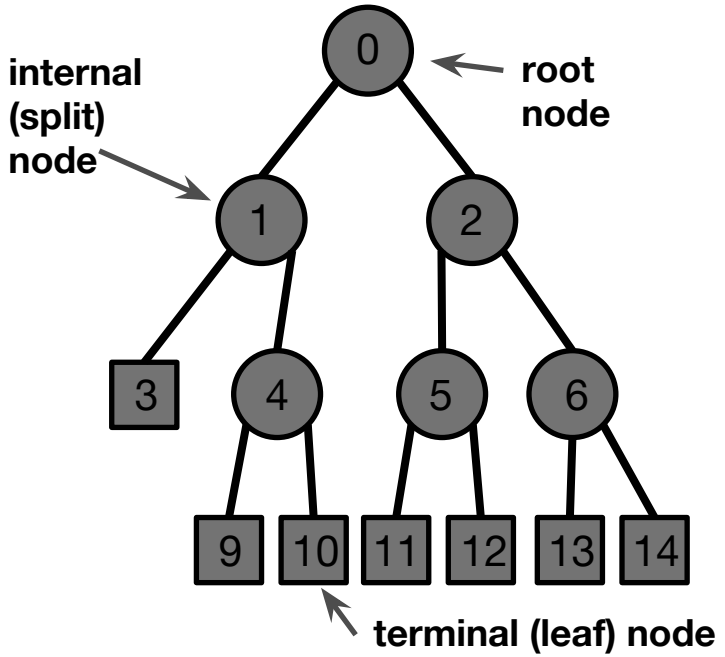
(1) Split node

consists in a decision function f_θ that splits observations in two subsets

(2) Leaf node

model the output, i.e. $P(\mathbf{Y}|\mathbf{X})$
corresponds to a part of the space

A general tree structure

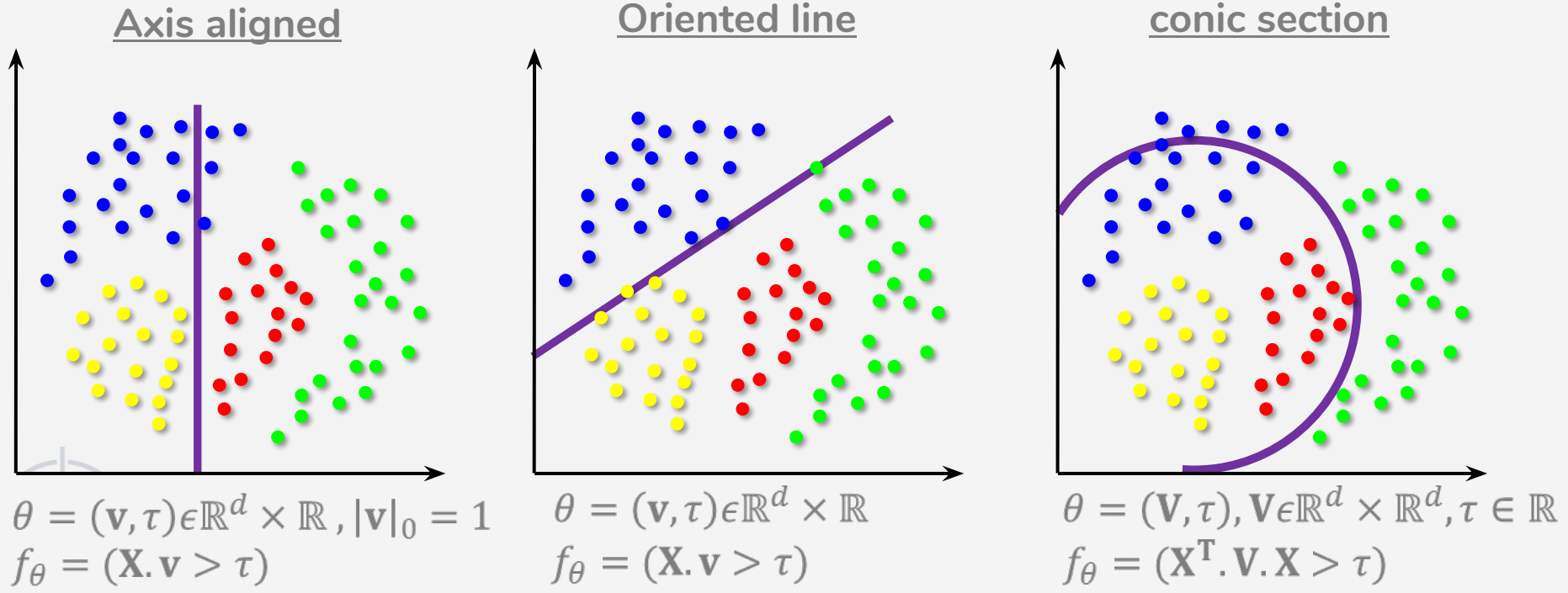


The generic tree model

Each split node is equipped with a decision function

Split the observations in 2 subsets, i.e. the feature space in two parts

Different possibilities for f_θ



Materials from A. Criminisi

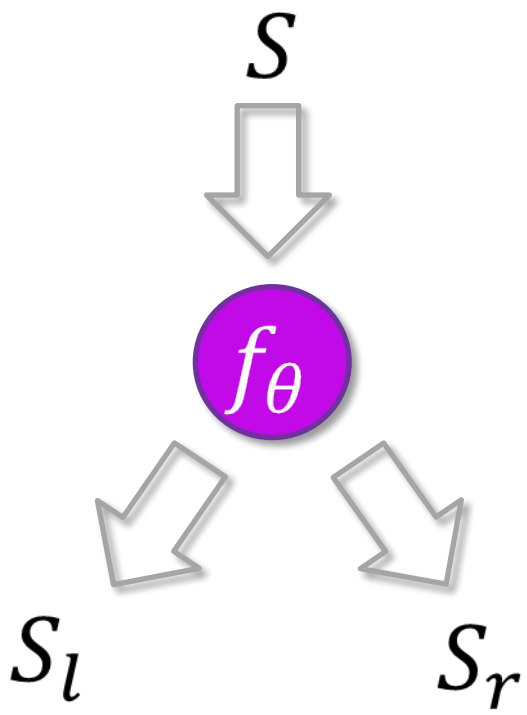
Tree training

Given a training set $\{(\mathbf{X_1}, \mathbf{Y_1}), \ldots, (\mathbf{X}_n, \mathbf{Y}_n), \ldots$

Iteratively split the training set

Greedy node optimization

- Generate a set of decision functions
- Choose the best according to a predefined objective function $\mathbf{I}(\mathbf{S}, \mathbf{S_l}, \mathbf{S}$
- Split data and send to left/right children



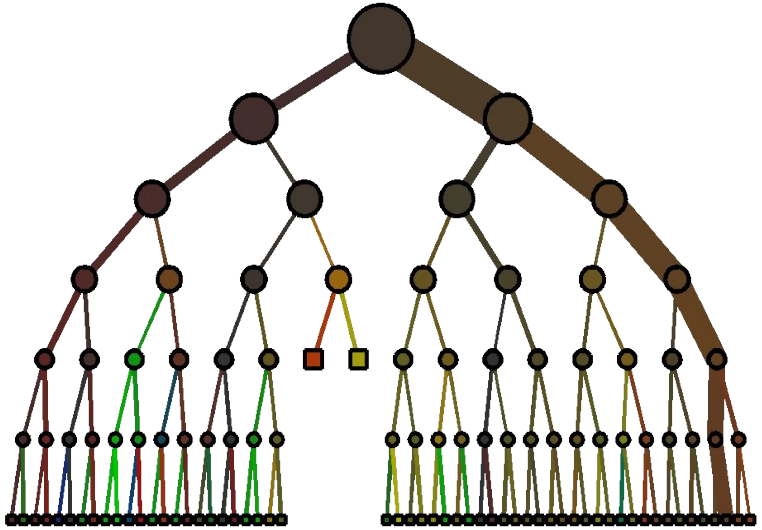
Tree training

Given a training set $\{(\mathbf{X_1}, \mathbf{Y_1}), \ldots, (\mathbf{X}_n, \mathbf{Y}_n), \ldots$

Iteratively split the training set

Stopping criteria

- **Maximum depth** reached
- **Objective function** lower than a predefined threshold
- **Number of training instances** below than a predefined threshold



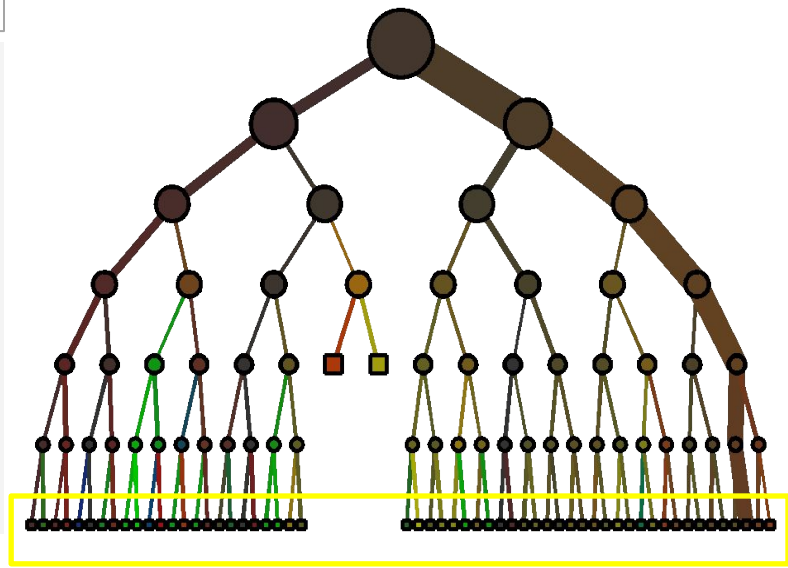
Tree training

Given a training set $\{(\mathbf{X_1}, \mathbf{Y_1}), \ldots, (\mathbf{X}_n, \mathbf{Y}_n), \ldots$

Modeling the data

Leaf Model

- Task-specific **posterior model**
- **Learned** from training instances reaching the leaf

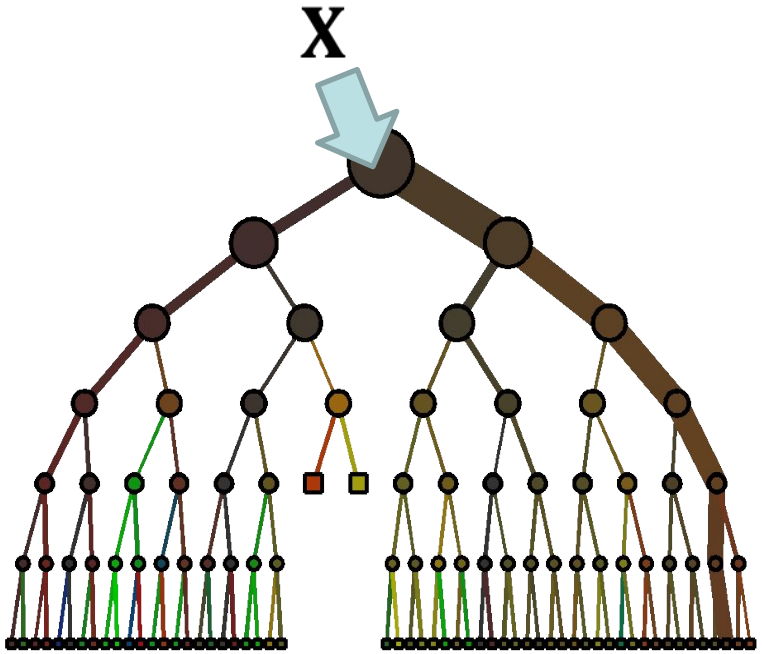


Tree prediction

Given an unseen observation $\mathbf{X}$

A test time

- **Push observation through the tree** according to the test results at each visited node.
- **When a leaf is reached,** gather the posterior model stored within



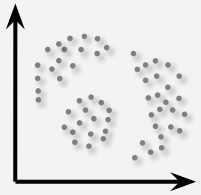
Prediction using maximum a posteriori:

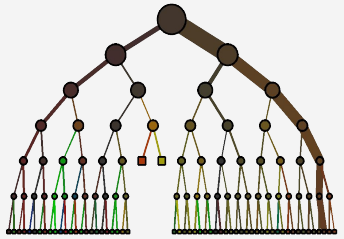
$$\hat{Y} = \mathbf{argmax}_Y \, P(Y|X)$$

Summary: tree model

Input: $\mathbf{X} \in \mathbb{R}^d$

Output: $\mathbf{Y} \in \{c_k\}_{k=1}^K$ or $\mathbf{Y} \in \mathbb{R}^{d'}$





Decision Tree

Components

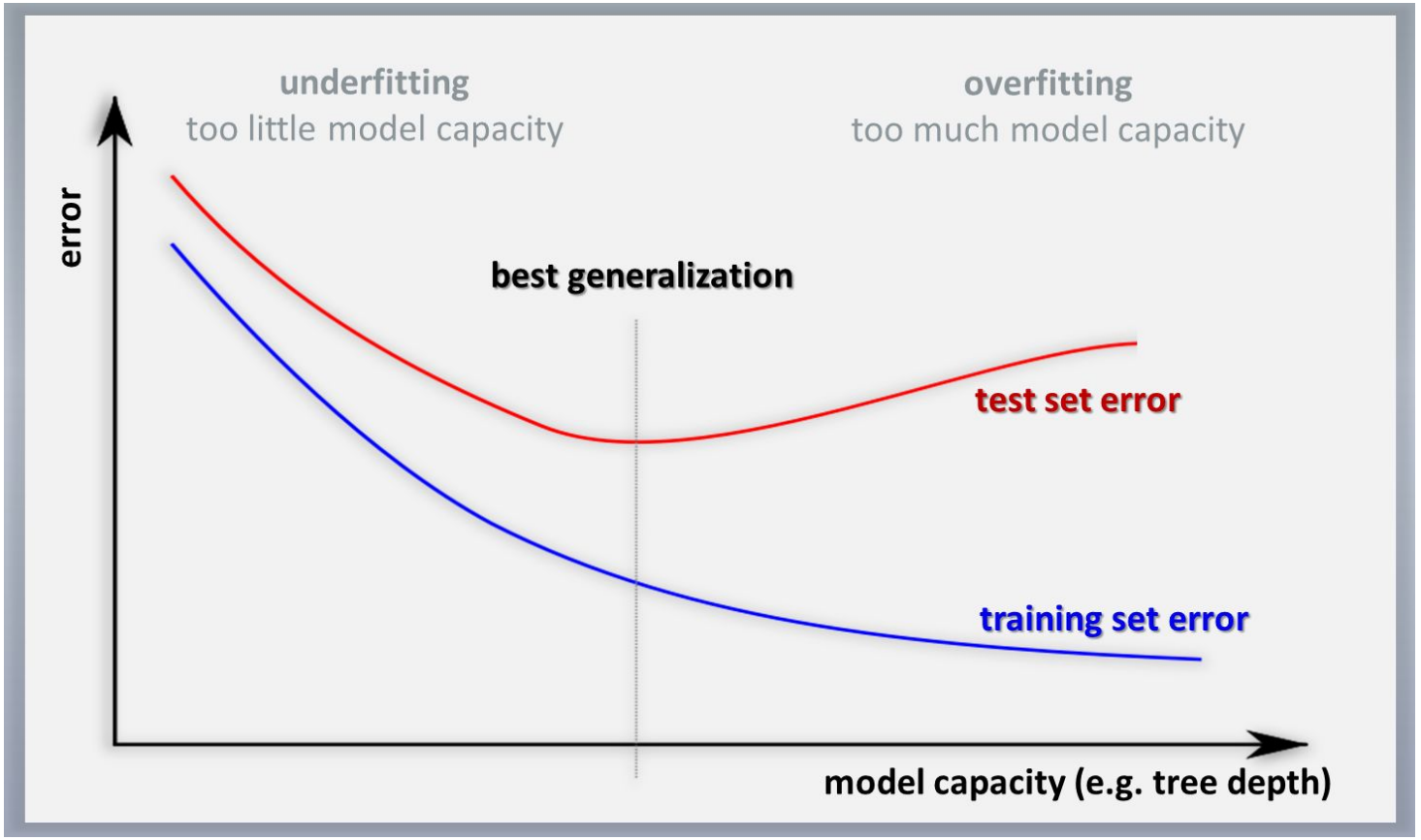
- Node decision function: f_θ
- Decision function param: θ
- Objective function $\mathbf{I}(\mathbf{S}, \mathbf{S_l}, \mathbf{S_r})$
- Leaf prediction model $\mathbf{P}(\mathbf{Y}|\mathbf{X})$

Parameters

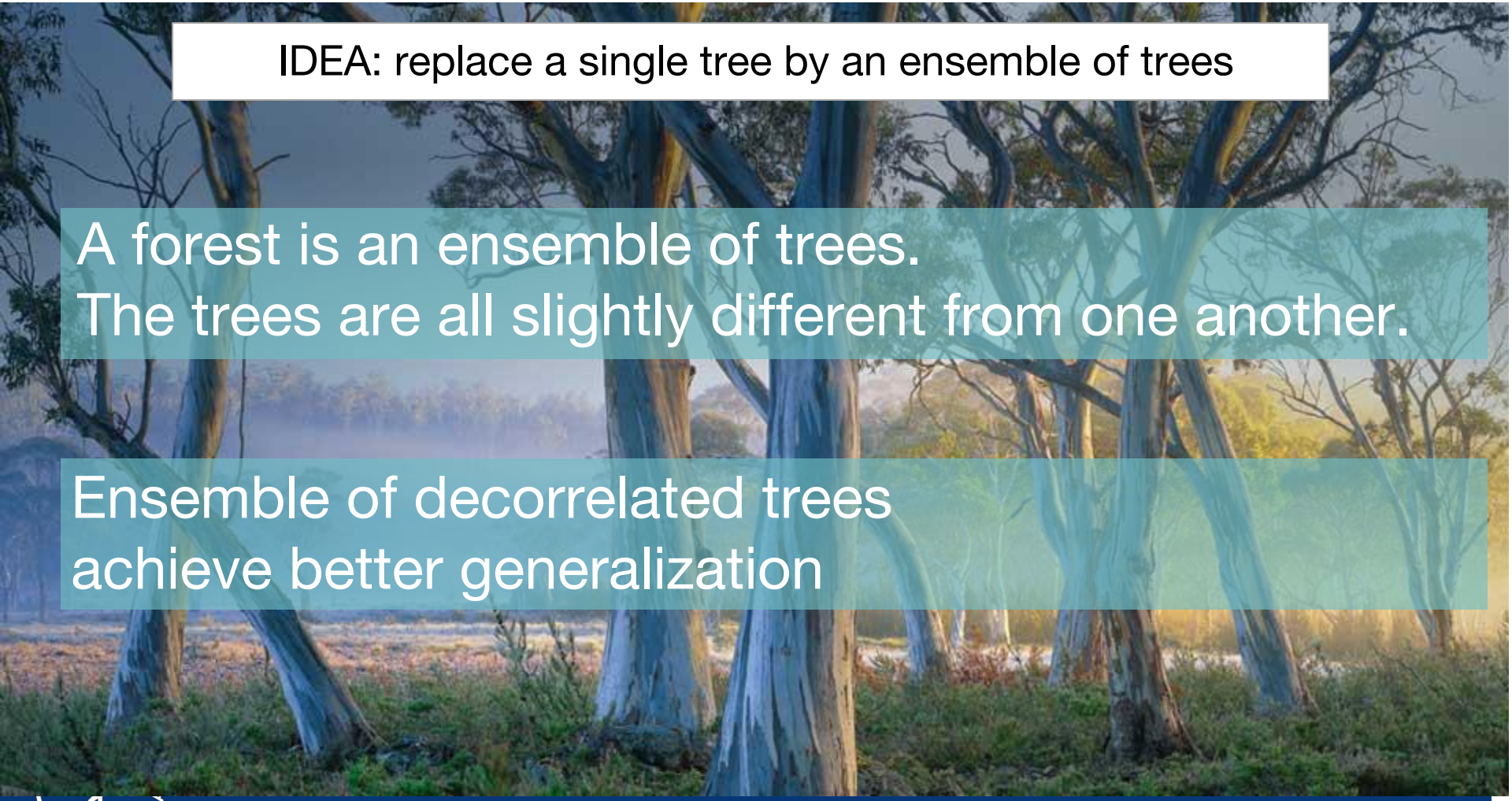
- Maximal tree depth: D
- Minimum Population: MinPop
- Minimum I: Imin

Limitation

A decision tree is prone to overfitting



From trees to forest



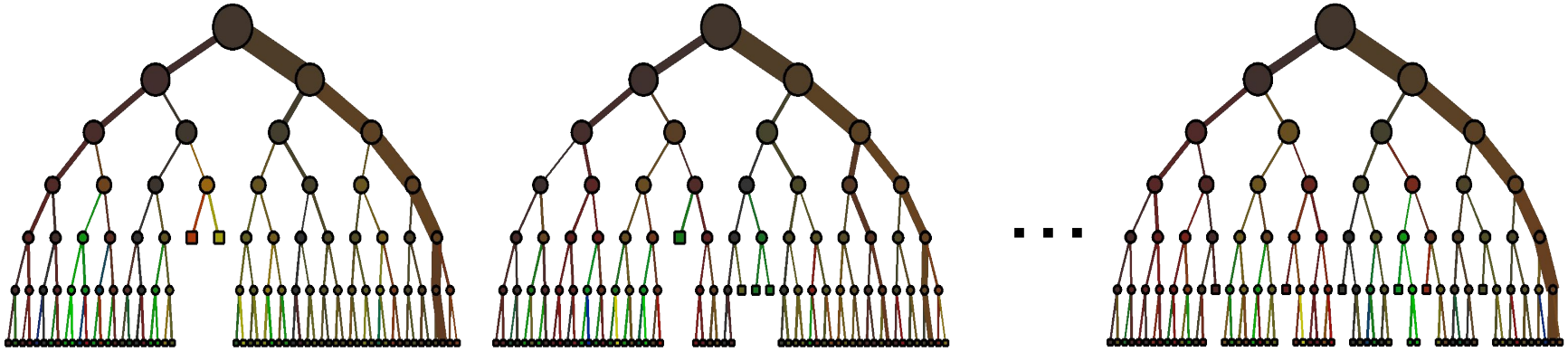
IDEA: replace a single tree by an ensemble of trees

A forest is an ensemble of trees.
The trees are all slightly different from one another.

Ensemble of decorrelated trees
achieve better generalization

From trees to forest

IDEA: replace a single tree by an ensemble of trees



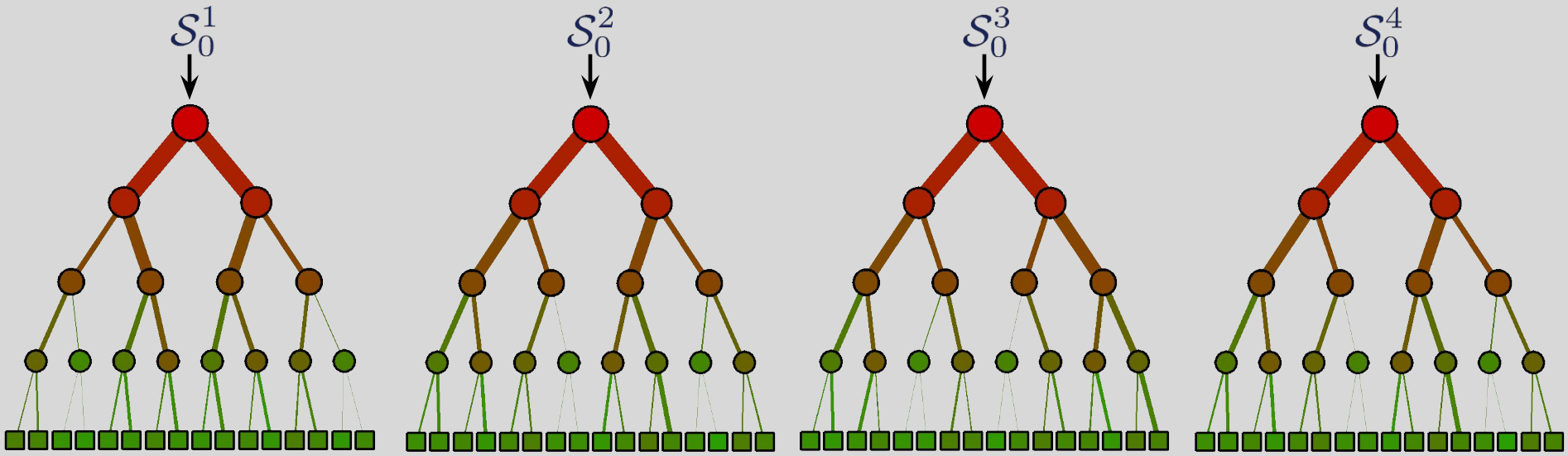
- How to create decorrelated trees?
- How to combine their output?
- How many trees?

From trees to forest

How to create decorrelated trees?

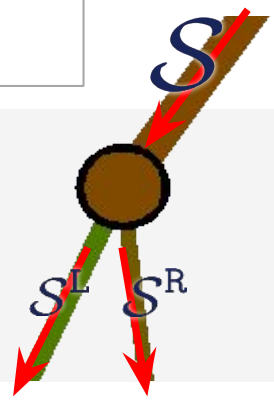
- Bootstrap Aggregating: Bagging
  - *Create **random subsets** of the training set* S_0
  - *Train each tree using **one** of these random subsets*

Forest training



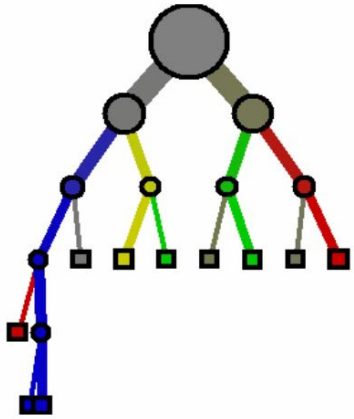
From trees to forest

How to create decorrelated trees?

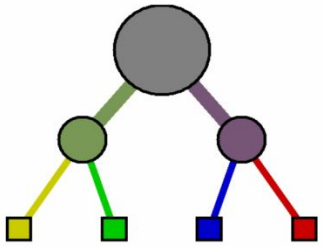


- Random Node Optimization
  - *Set of possible decision function parameters* Γ
  - *Create a random subset* $\mathcal{T} \subset \Gamma$
  - *Choose the best function from* $\mathcal{T}$

$|\mathcal{T}|$ *small, low correlation*



$|\mathcal{T}|$ *high, high correlation*

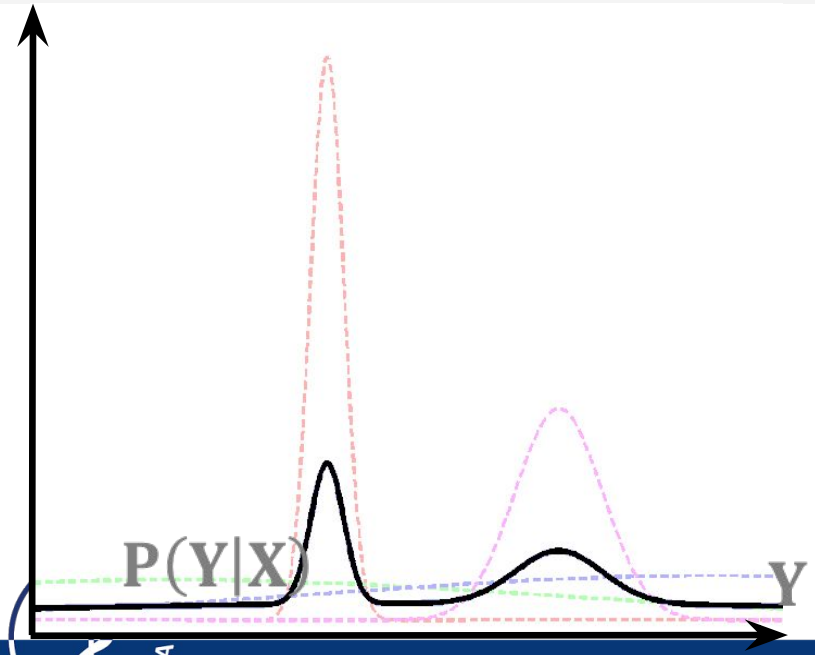


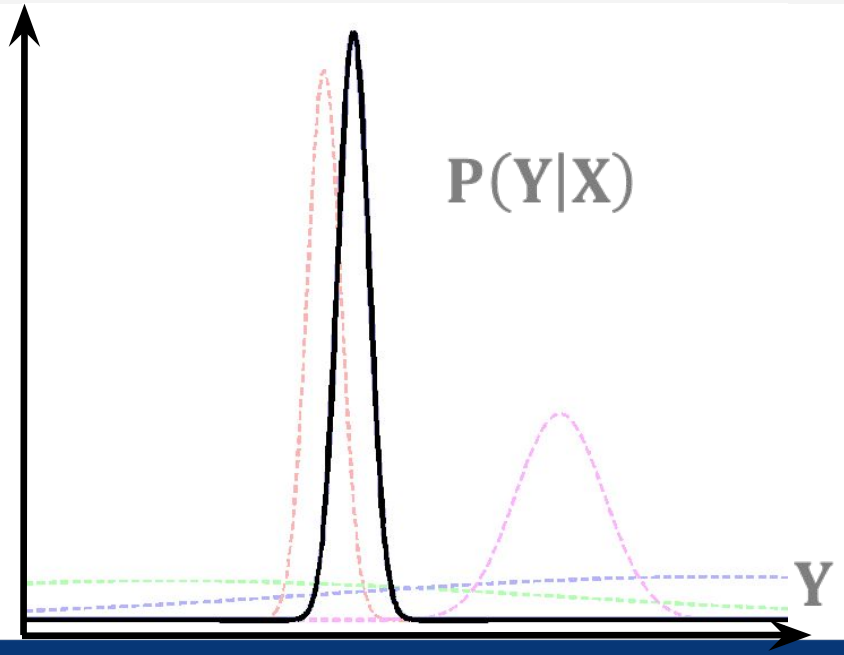
From trees to forest

How to combine their outputs?

$$\mathbf{P(Y|X)} = \frac{1}{T}\sum_{t=1}^{T} P_t(\mathbf{Y}|\mathbf{X})$$

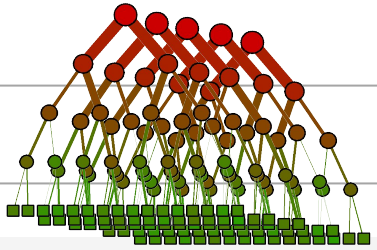
$$\mathbf{P(Y|X)} = \frac{1}{Z}\prod_{t=1}^{T} P_t(\mathbf{Y}|\mathbf{X})$$





Summary: forest model

Decision Forest



Components

- Node decision function: f_θ
- Decision function param: θ
- Objective function $\mathbf{I}(\mathbf{S}, \mathbf{S_l}, \mathbf{S_r})$
- Leaf prediction model $\mathbf{P}(\mathbf{Y}|\mathbf{X})$
- Forest prediction model

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{\mathbf{1}}{\mathbf{T}} \sum_{t=1}^{\mathbf{T}} P_t(\mathbf{Y}|\mathbf{X})$$

Parameters

- Maximal tree depth: D
- Minimum Population: MinPop
- Minimum Energy: Imin
- Bagging ratio $\mathrm{R} = |S_0^t|/|S_0|$
- Number of tries/node $|\mathcal{T}|$
- Number of trees T

Classification forests

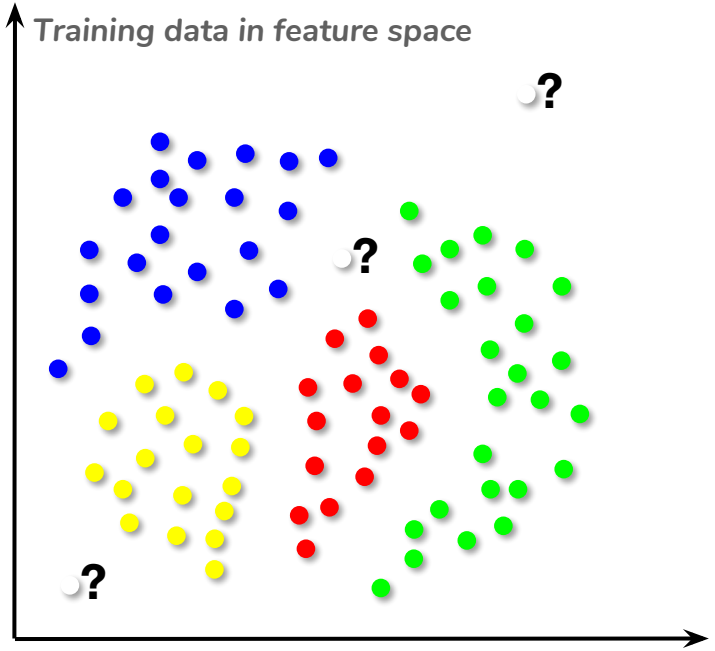
Supervised classification

Observation

$$\mathbf{X} = [x_1 \ \dots \ x_i \ \dots \ x_d]^T, \mathbf{X} \in \mathbb{R}^d$$

Class label

$$\mathbf{Y} \in \{c_1, \dots, c_k, \dots, c_K\}$$



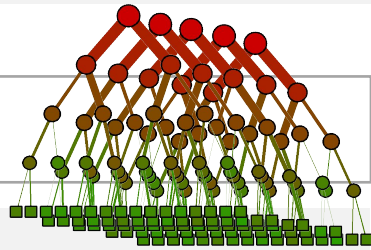
Given a training set: $\{(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_n, \mathbf{Y}_n), \dots$

GOAL: learn the conditional distribution $P(\mathbf{Y}|\mathbf{X})$

Perform prediction using $\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}} \, \mathbf{P}(\mathbf{Y}|\mathbf{X})$

Classification forest model

Decision Forest



Components

- Node decision function: f_θ
- Decision function param: θ
- Objective function $\mathbf{I(S, S_l, S_r)}$
- Leaf prediction model $\mathbf{P(Y|X)}$
- Forest prediction model

$$\mathbf{P(Y|X)} = \frac{1}{T}\sum_{t=1}^{T} P_t(\mathbf{Y|X})$$

Parameters

- Maximal tree depth: D
- Minimum Population: MinPop
- Minimum Energy: Imin
- Bagging ratio: $\mathrm{R} = |S_0^t|/|S_0|$
- Number of tries/node $\mathrm{Ntry} = |\mathcal{T}|$
- Number of trees: T

Objective function

Node training

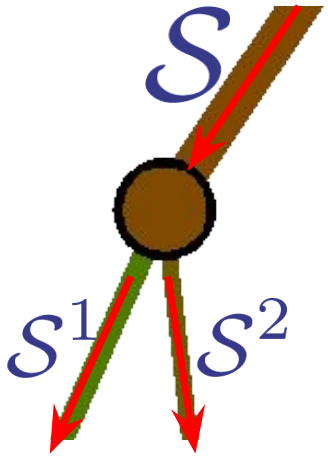
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\mathcal{T}} I$$

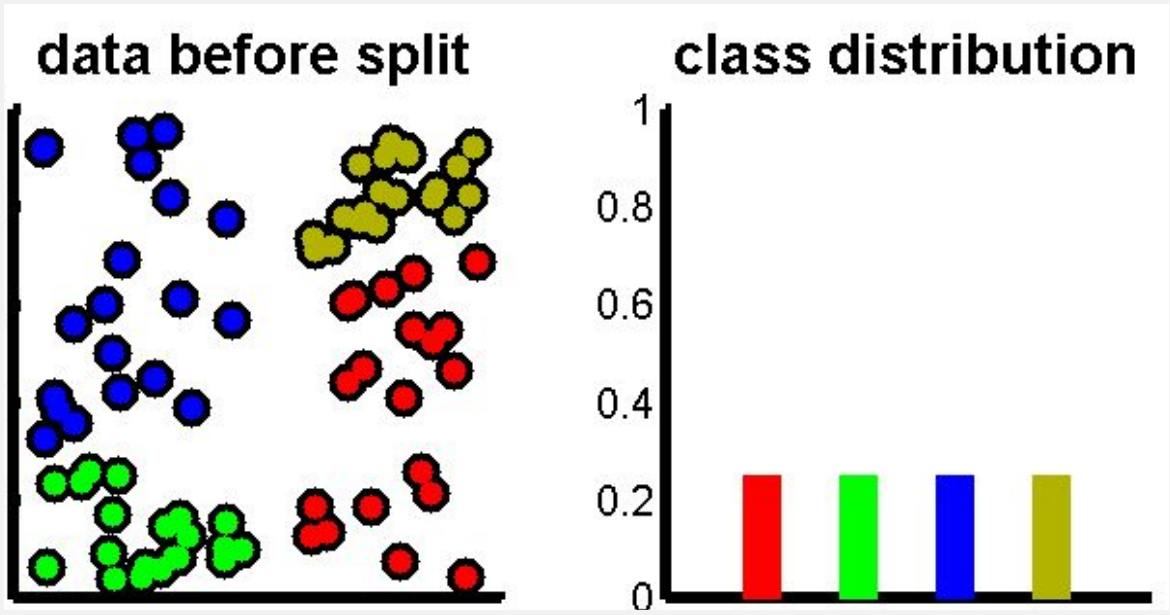
Information gain

$$I = H(\mathcal{S}) - \sum_{i\in\{1,2\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H(\mathcal{S}^i)$$

Shannon's entropy

$$H(\mathcal{S}) = -\sum_{c\in\mathcal{C}} p(c)\log(p(c))$$





Objective function

Node training

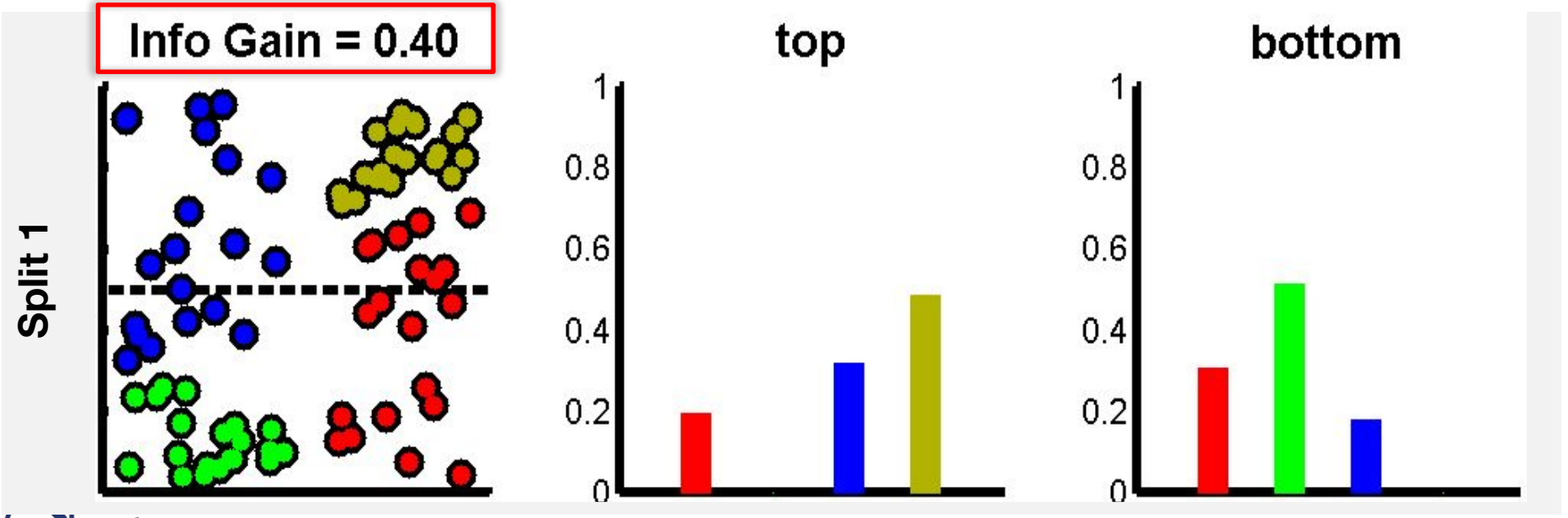
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \mathcal{T}} I$$

Information gain

$$I = H(\mathcal{S}) - \sum_{i \in \{1,2\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H(\mathcal{S}^i)$$

Shannon's entropy

$$H(\mathcal{S}) = -\sum_{c \in \mathcal{C}} p(c) \log(p(c))$$



Objective function

Node training

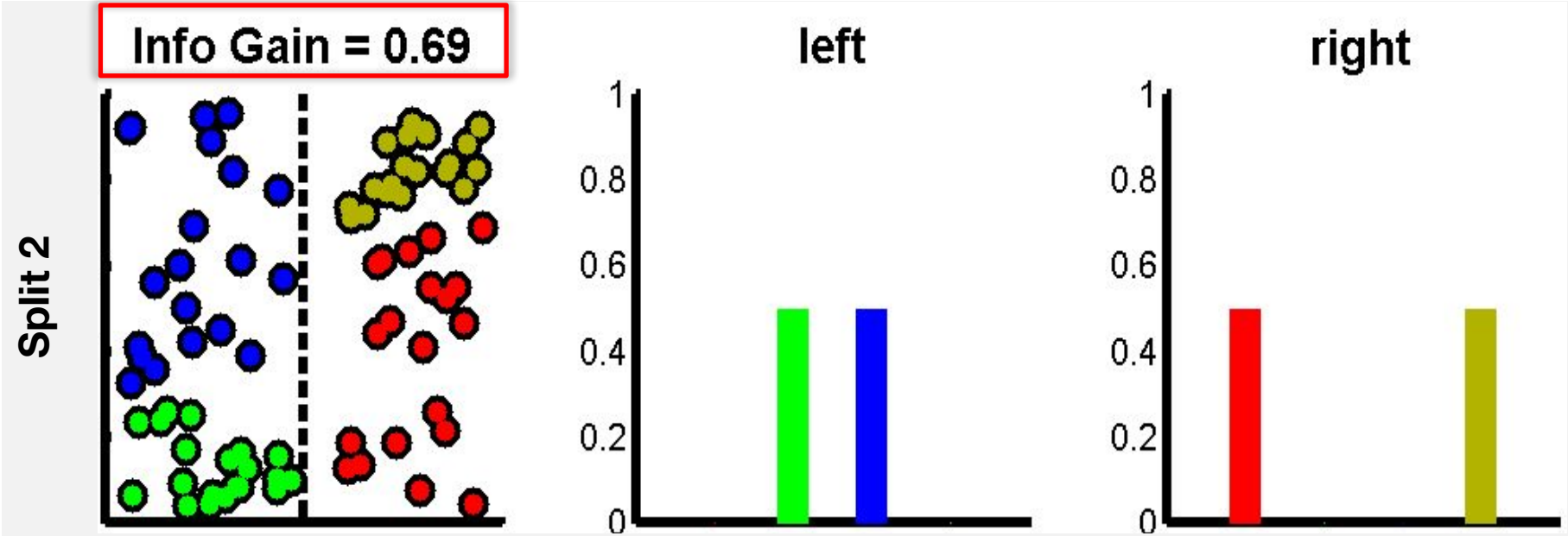
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\mathcal{T}} I$$

Information gain

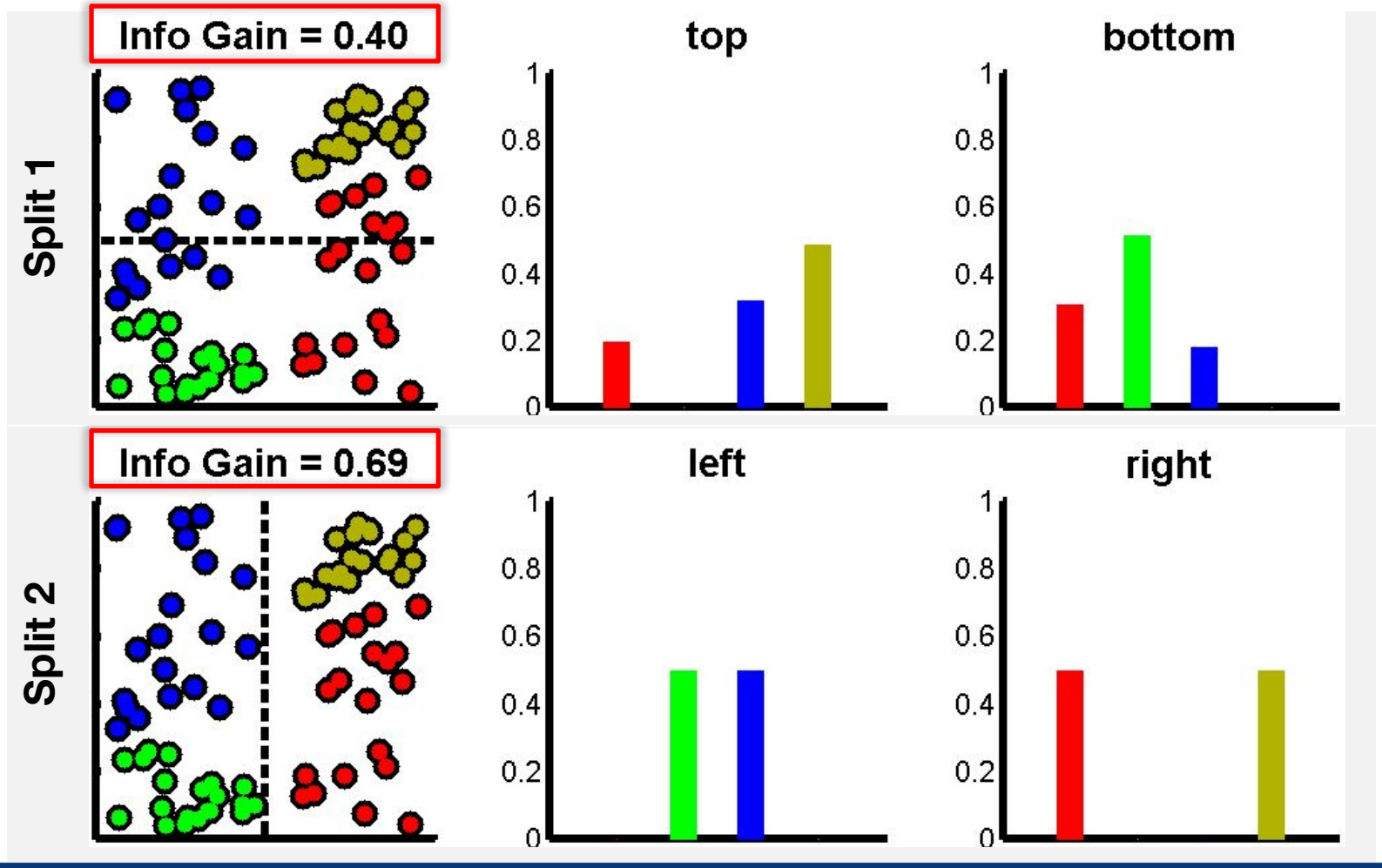
$$I = H(\mathcal{S}) - \sum_{i\in\{1,2\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H(\mathcal{S}^i)$$

Shannon's entropy

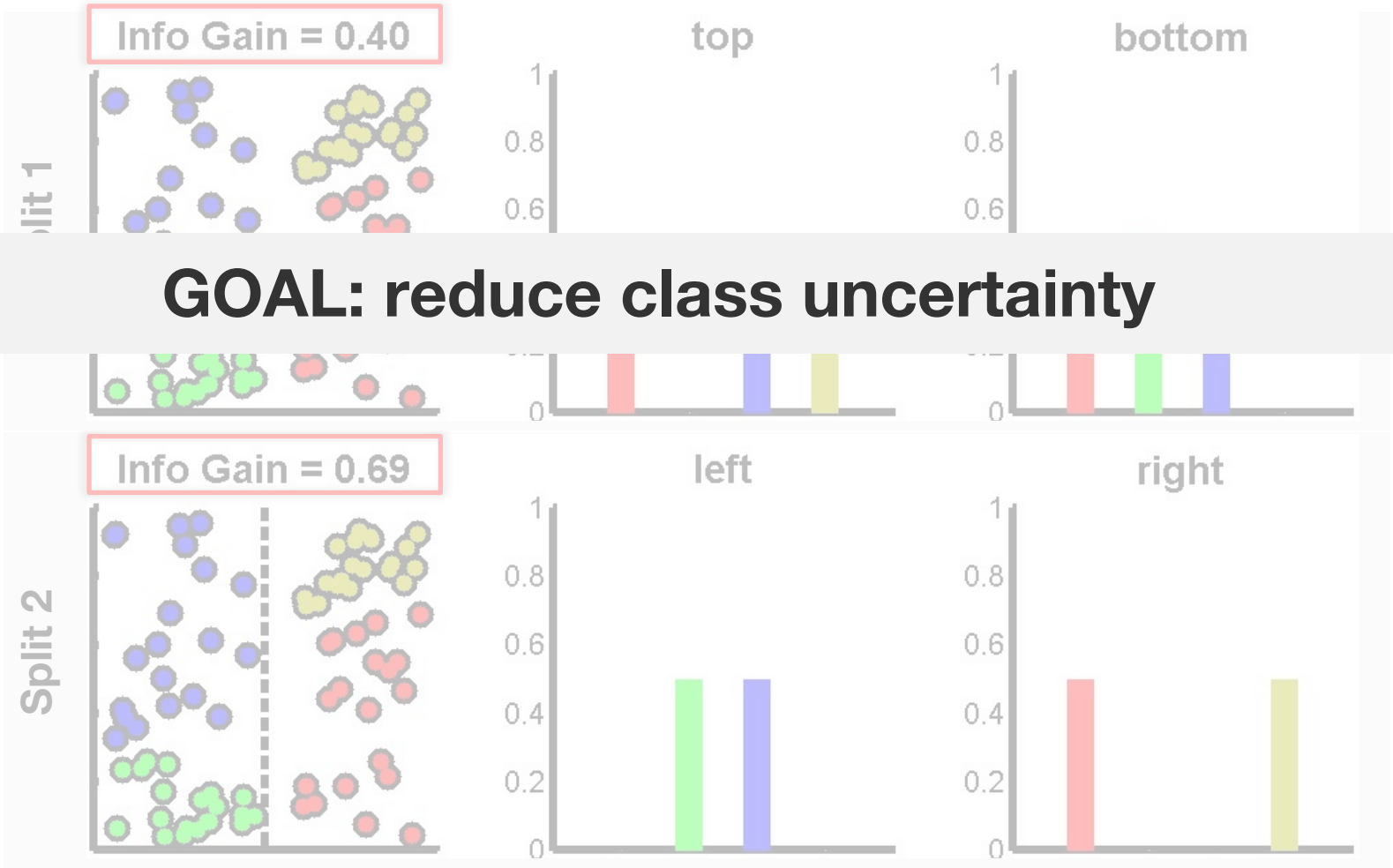
$$H(\mathcal{S}) = -\sum_{c\in\mathcal{C}} p(c)\log(p(c))$$



Objective function

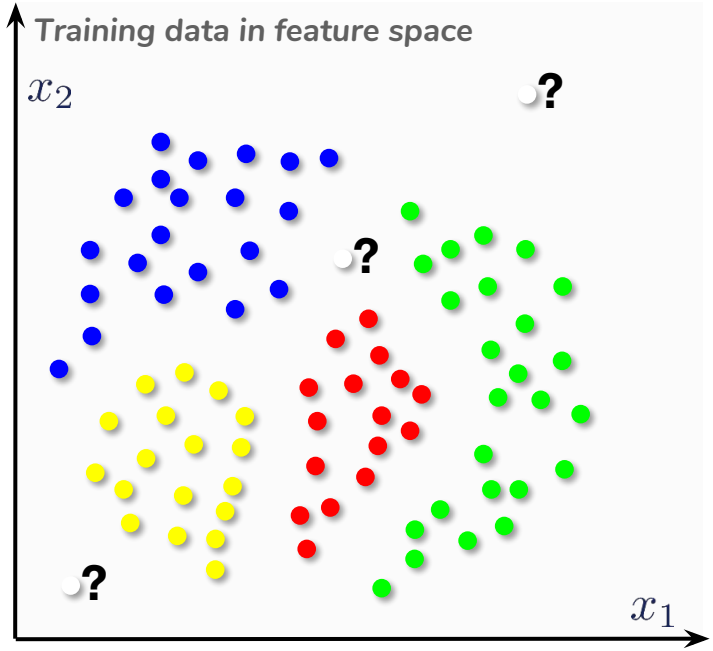


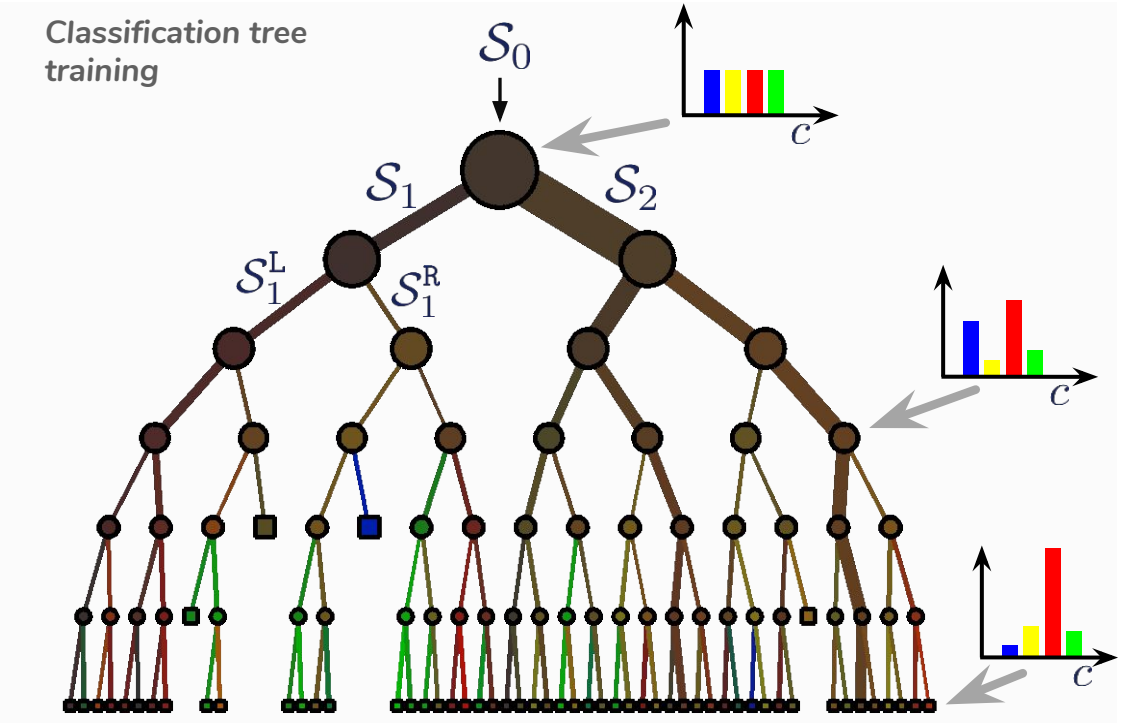
Objective function



GOAL: reduce class uncertainty

Objective function





Objective function

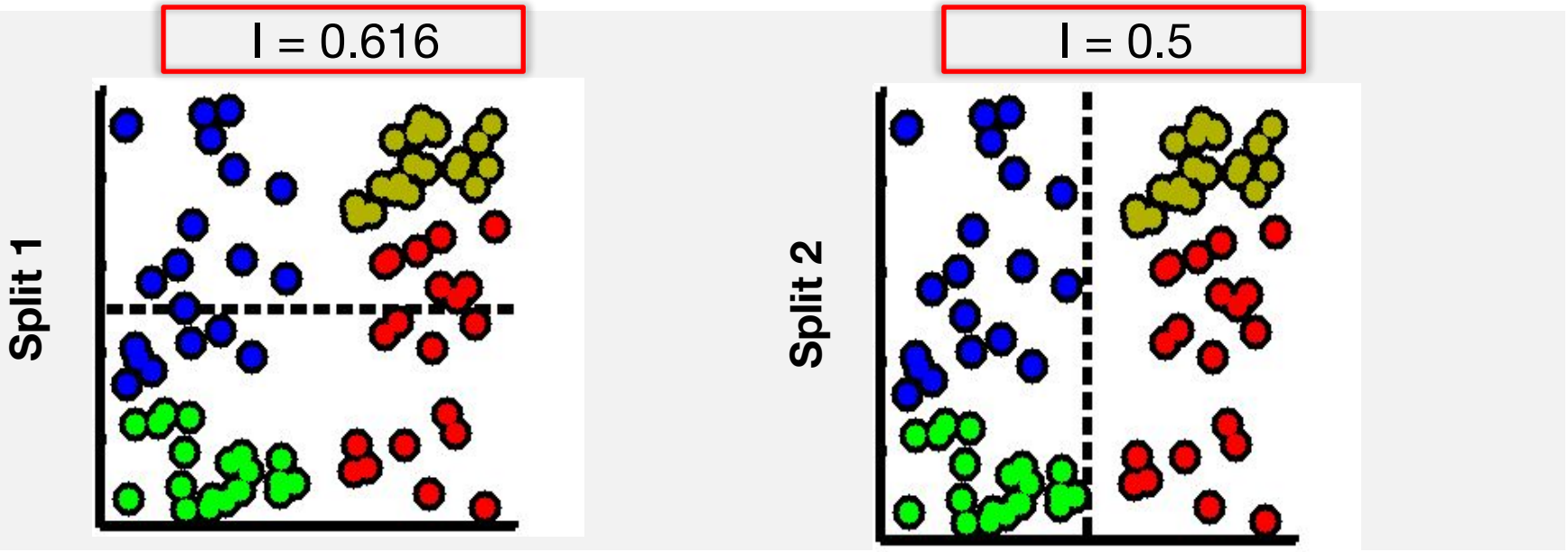
Node training

$$\theta^* = \underset{\theta \in \tau}{\operatorname{argmin}}\, I$$

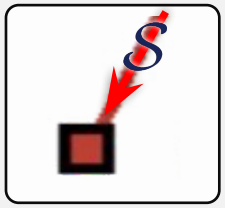
Gini Impurity

$$G(S) = 1 - \sum_{\forall c} p_c^2$$

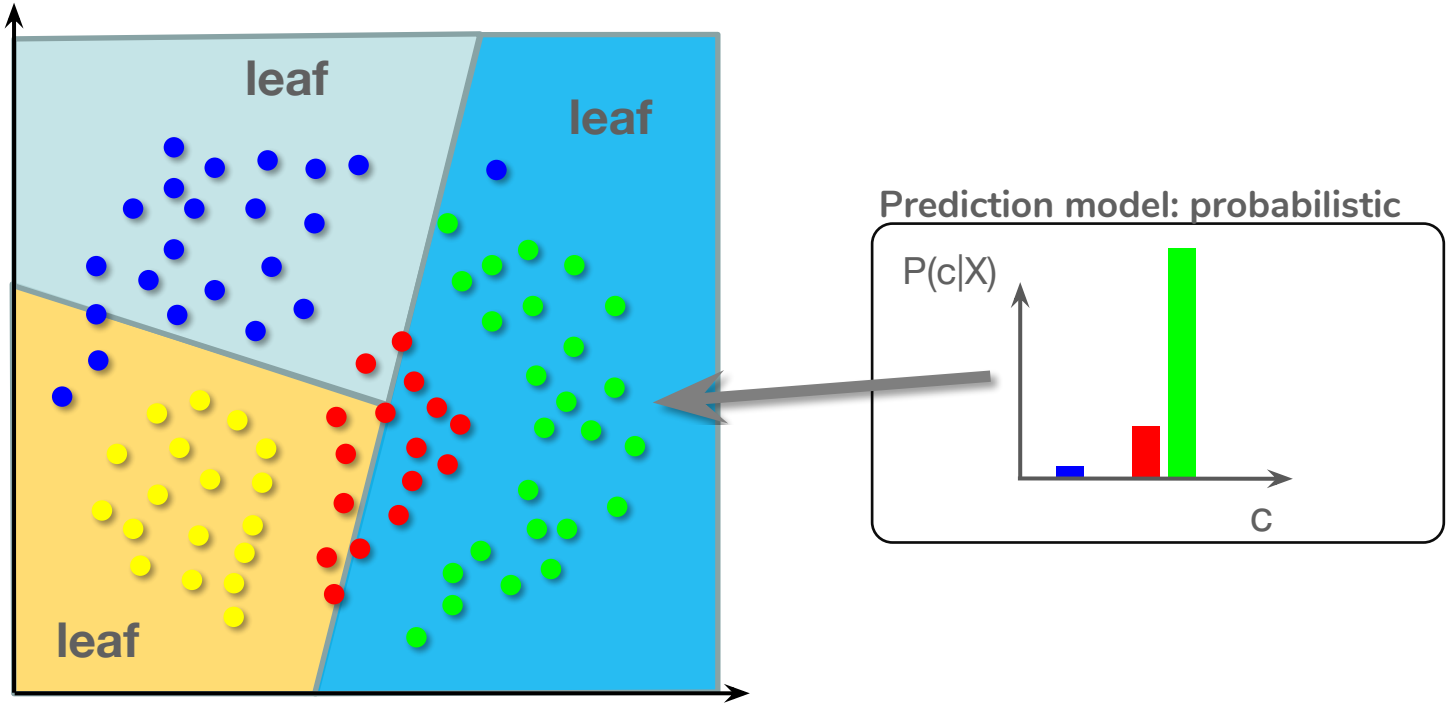
$$I_{gini} = \sum_{n \in [l,r]} \frac{|S_n|}{|S|} G(S_n)$$



Leaf model: class posterior



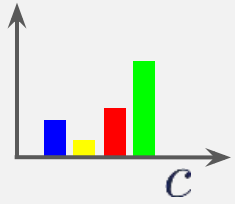
What do we at a leaf?
Compute **class histogram**
from set of points **reaching the leaf**

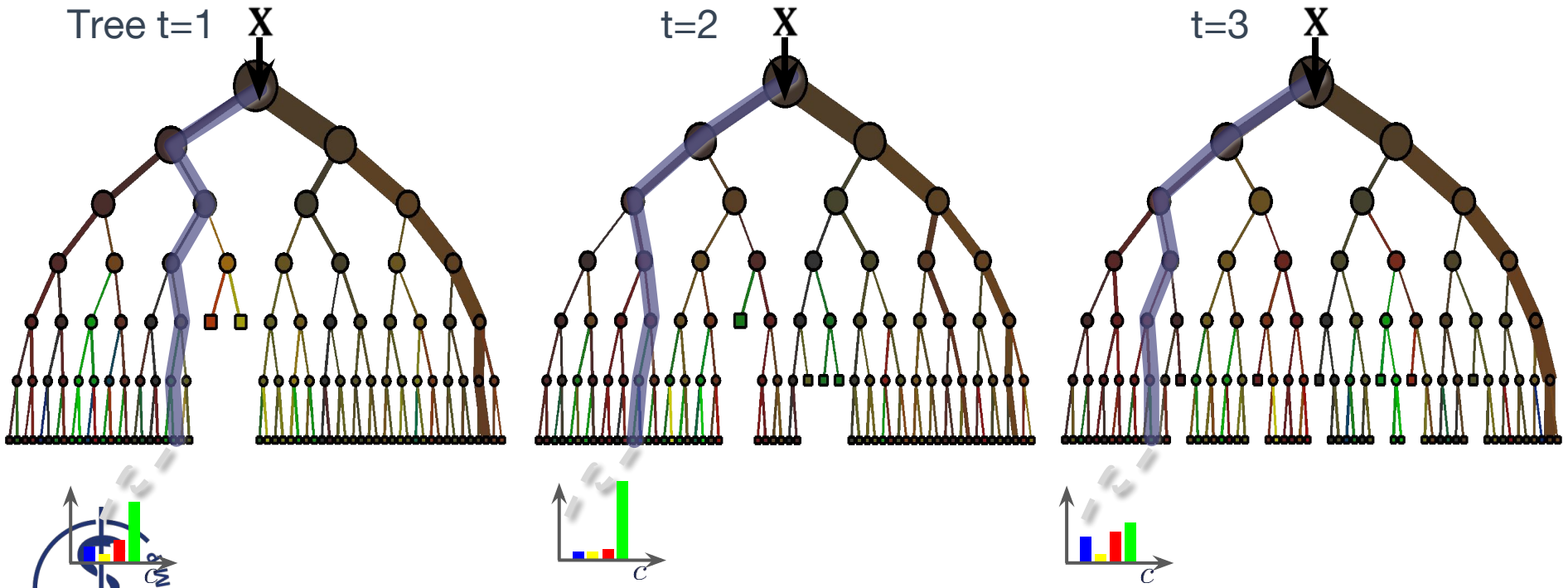


Forest prediction: posterior averaging

Forest output probability

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} P_t(\mathbf{Y}|\mathbf{X})$$





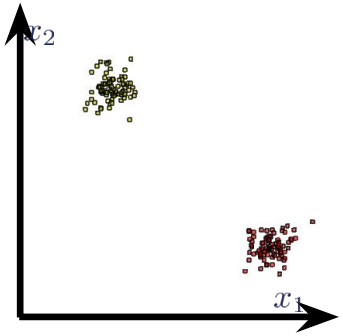
Forest prediction: posterior averaging

Forest output probability $\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} P_t(\mathbf{Y}|\mathbf{X})$

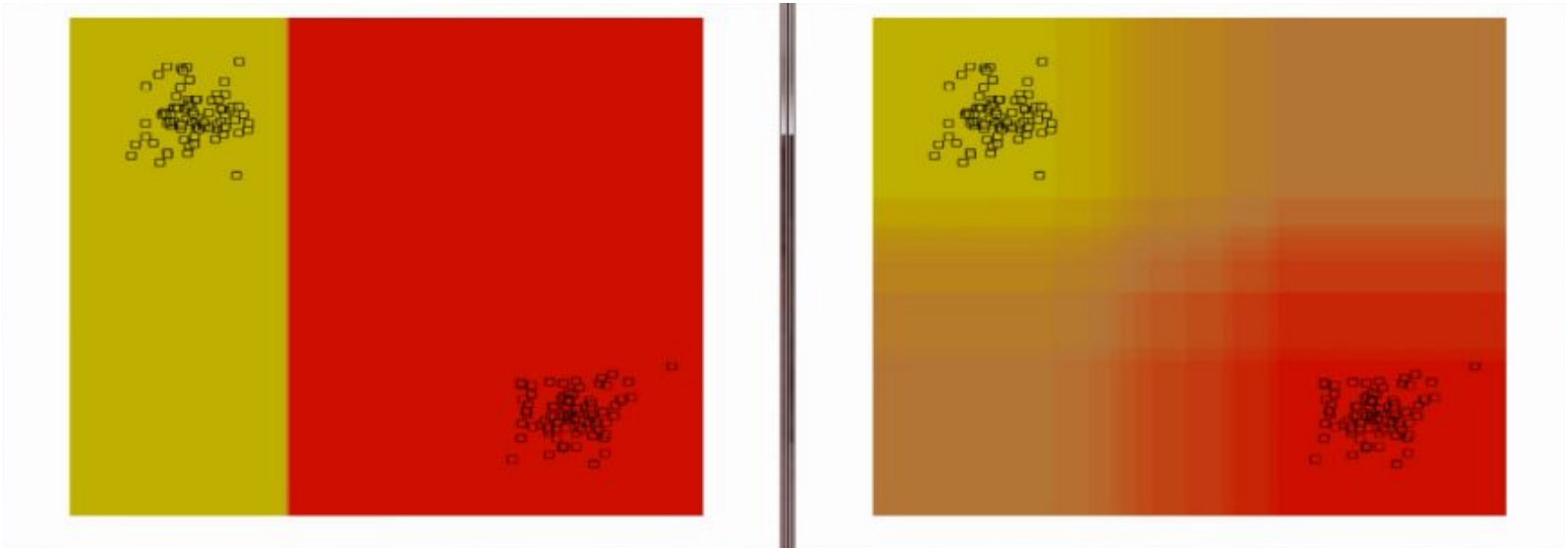
Maximum a posteriori:

Perform prediction using $\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}}\, \mathbf{P}(\mathbf{Y}|\mathbf{X})$

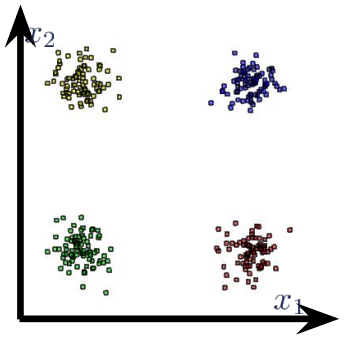
Toy example: effects of the decision function



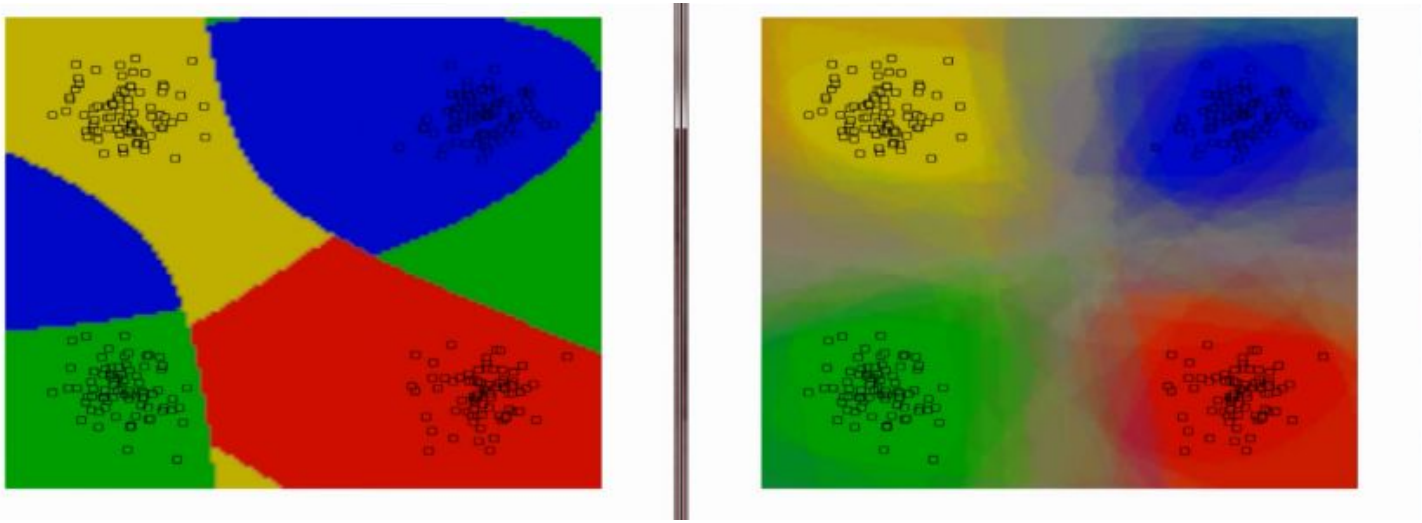
- 2 classes problem - linearly separable
- Axis aligned decision function
- Generalization - confidence



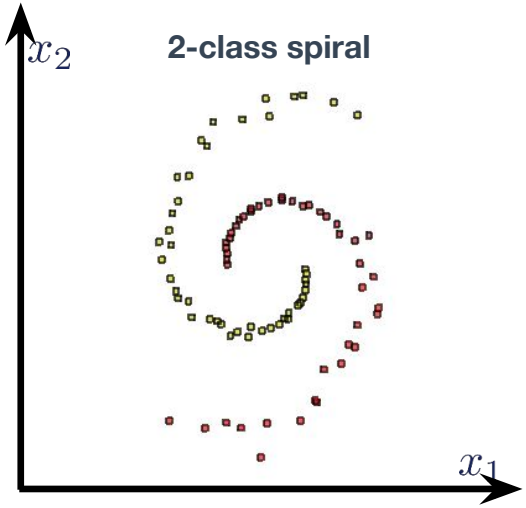
Toy example: multi-class problem

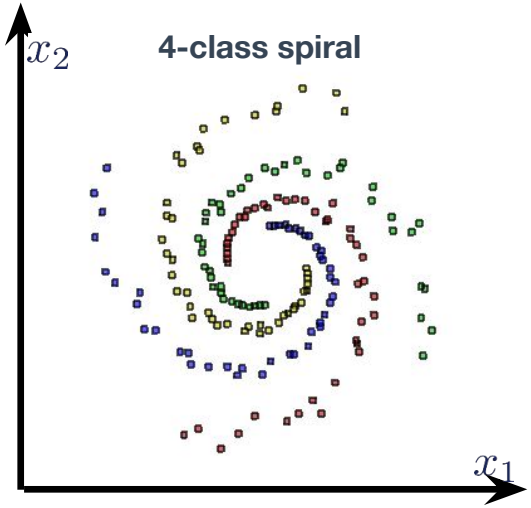


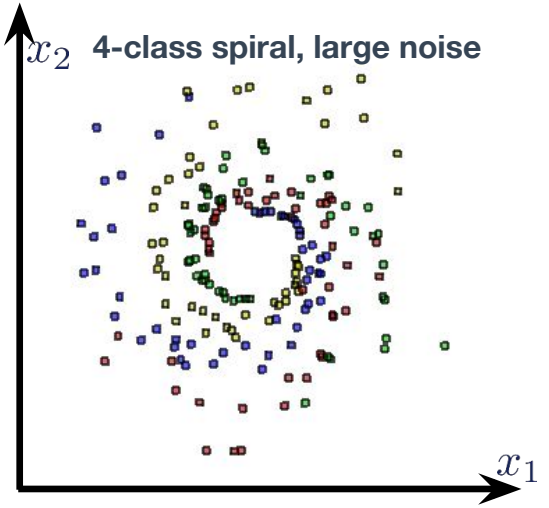
- Forests are inherently multi-class
- 4 classes problem – linearly separable
- Conic section decision functions



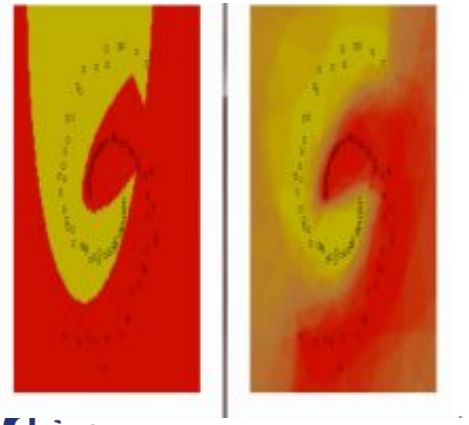
Toy example: non-linear classification

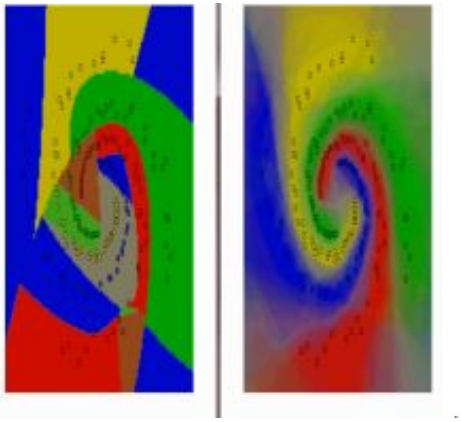


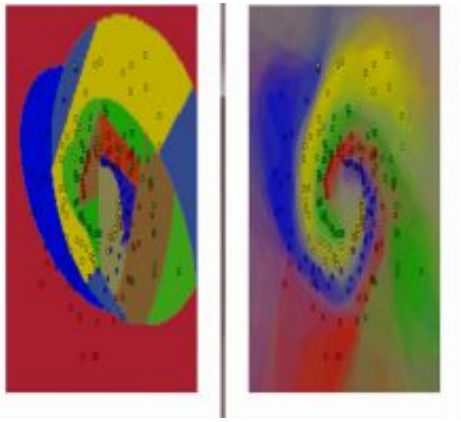




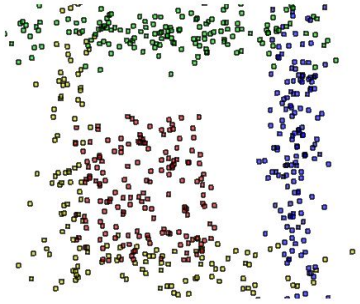
Testing posteriors





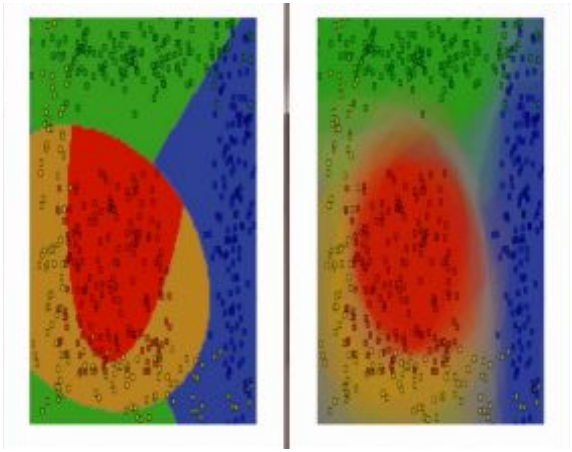


Toy example: effects of the tree depth

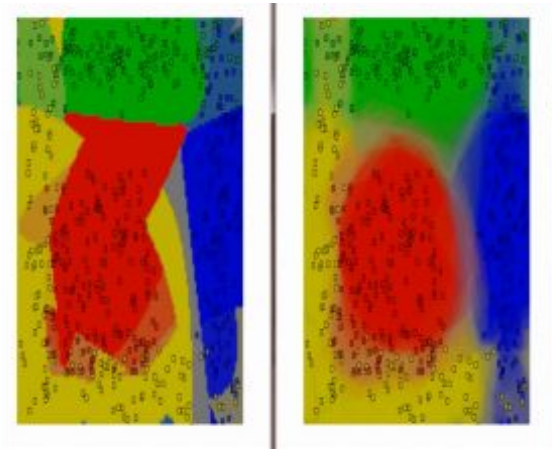


- Tree depth is a crucial parameter
- Influence on the generalization

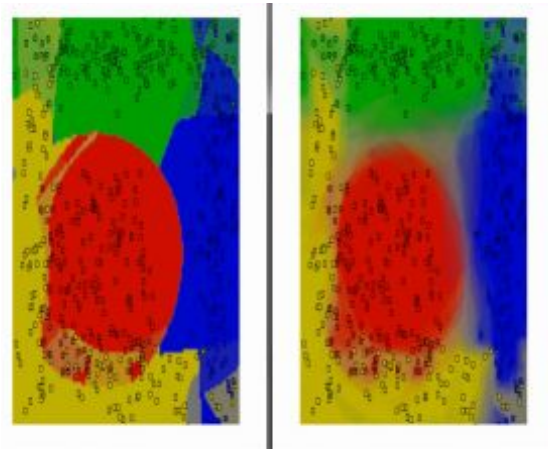
max tree depth, D



T=200, D=3, w. l. = conic



T=200, D=6, w. l. = conic

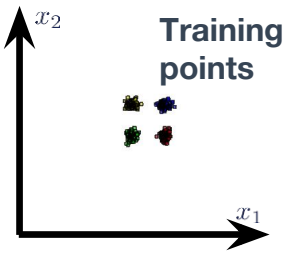


T=200, D=15, w. l. = conic

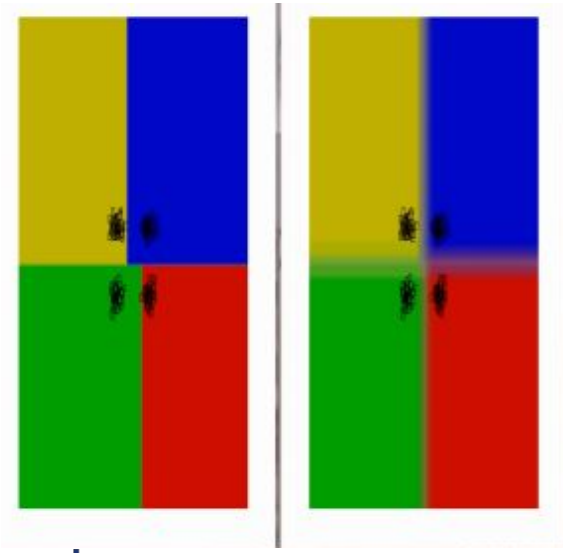
underfitting

overfitting

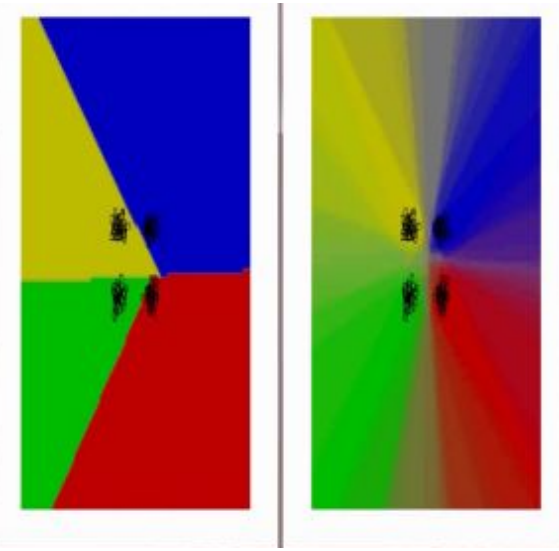
Toy example: Generalization



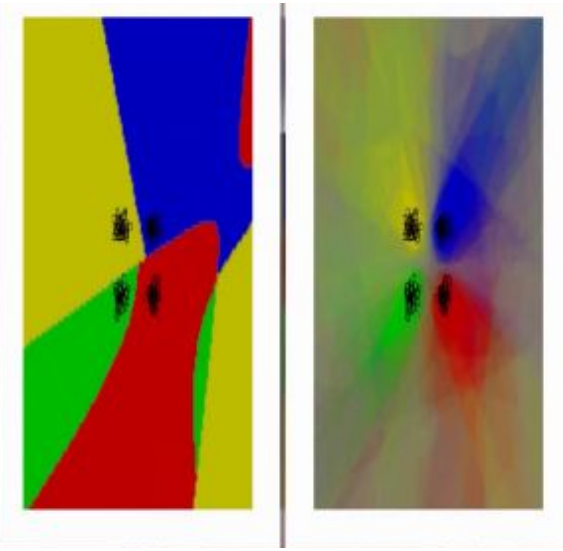
Decision function: axis aligned



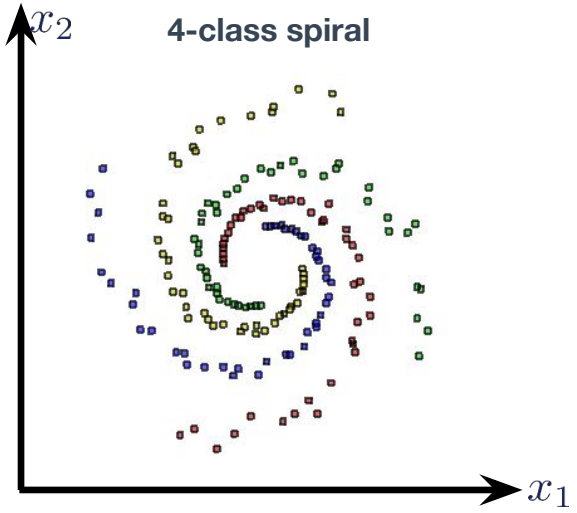
Decision function: oriented line

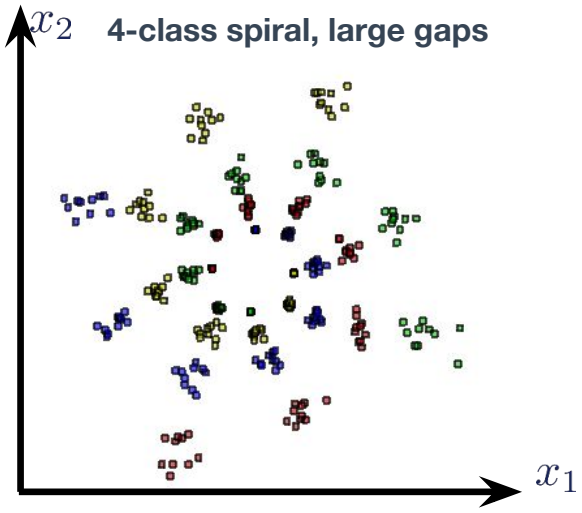


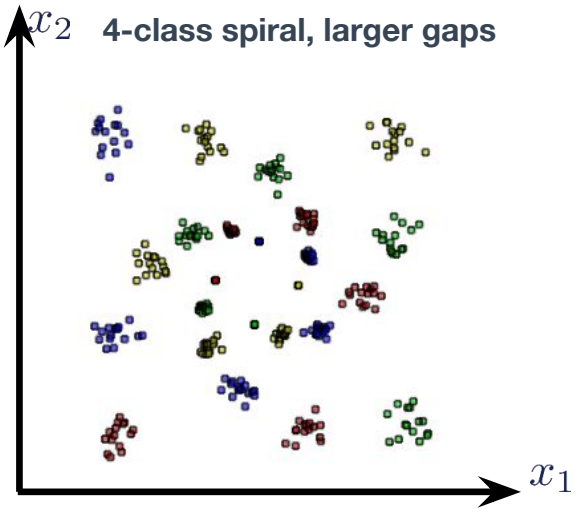
Decision function: conic section

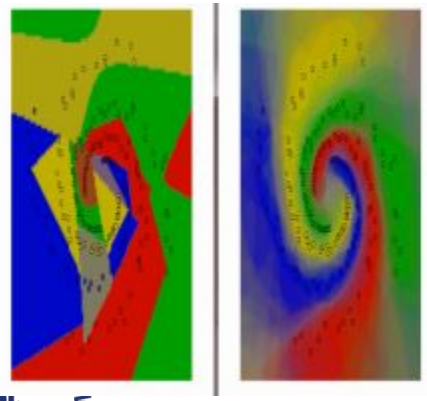


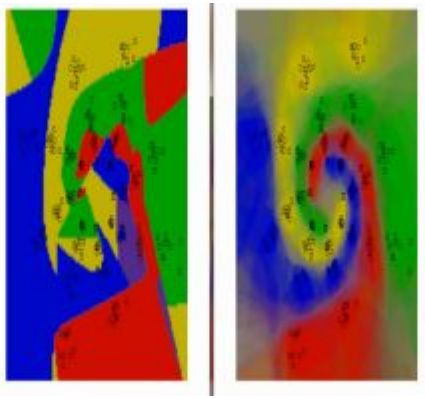
Toy example: Generalization

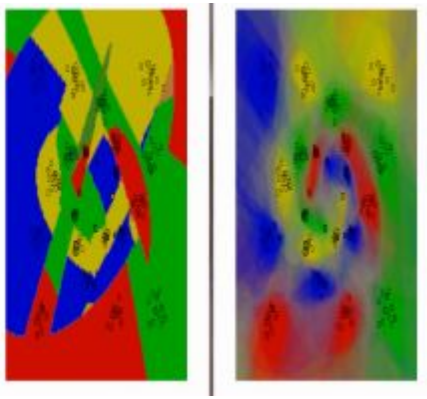






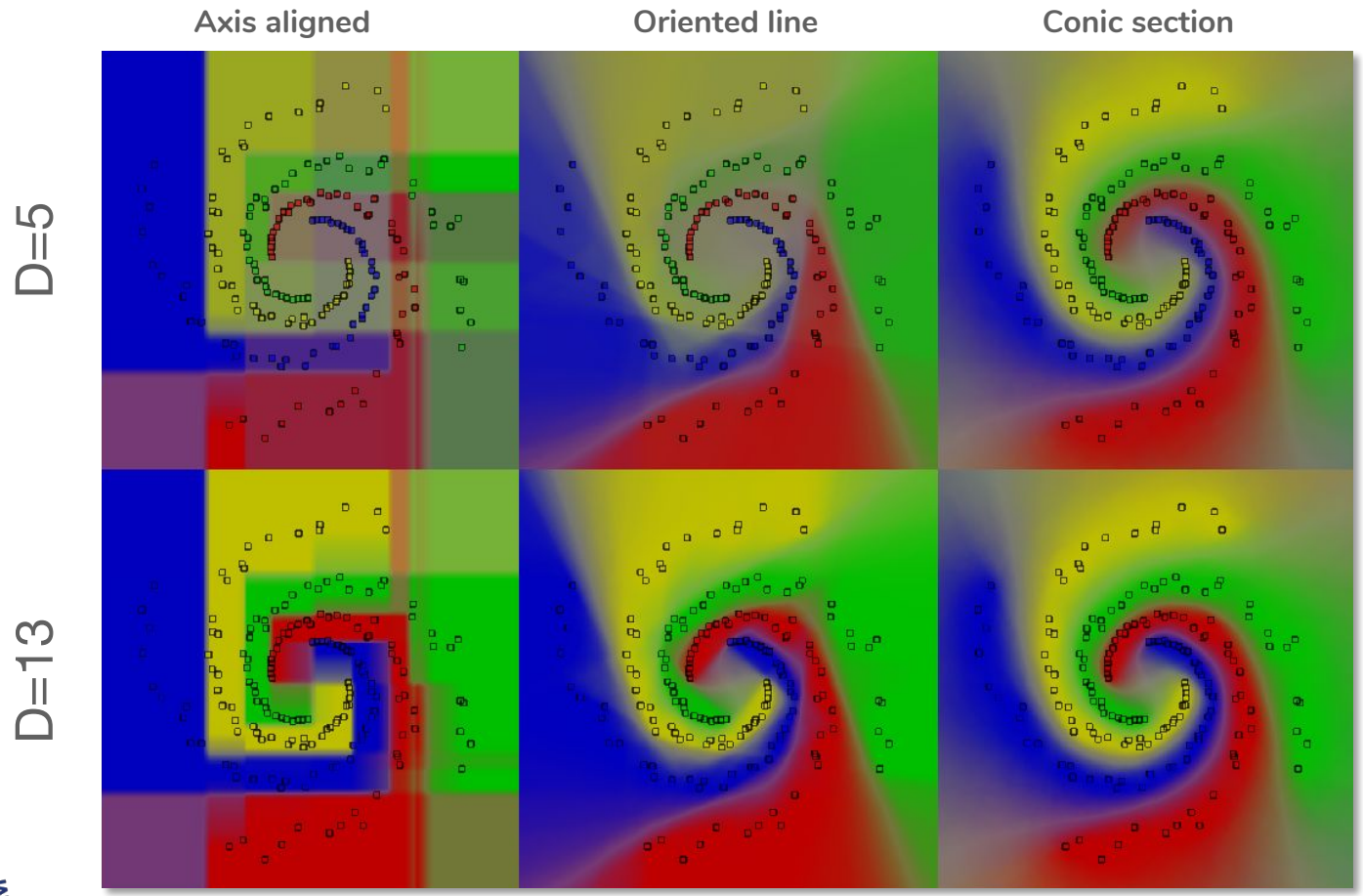






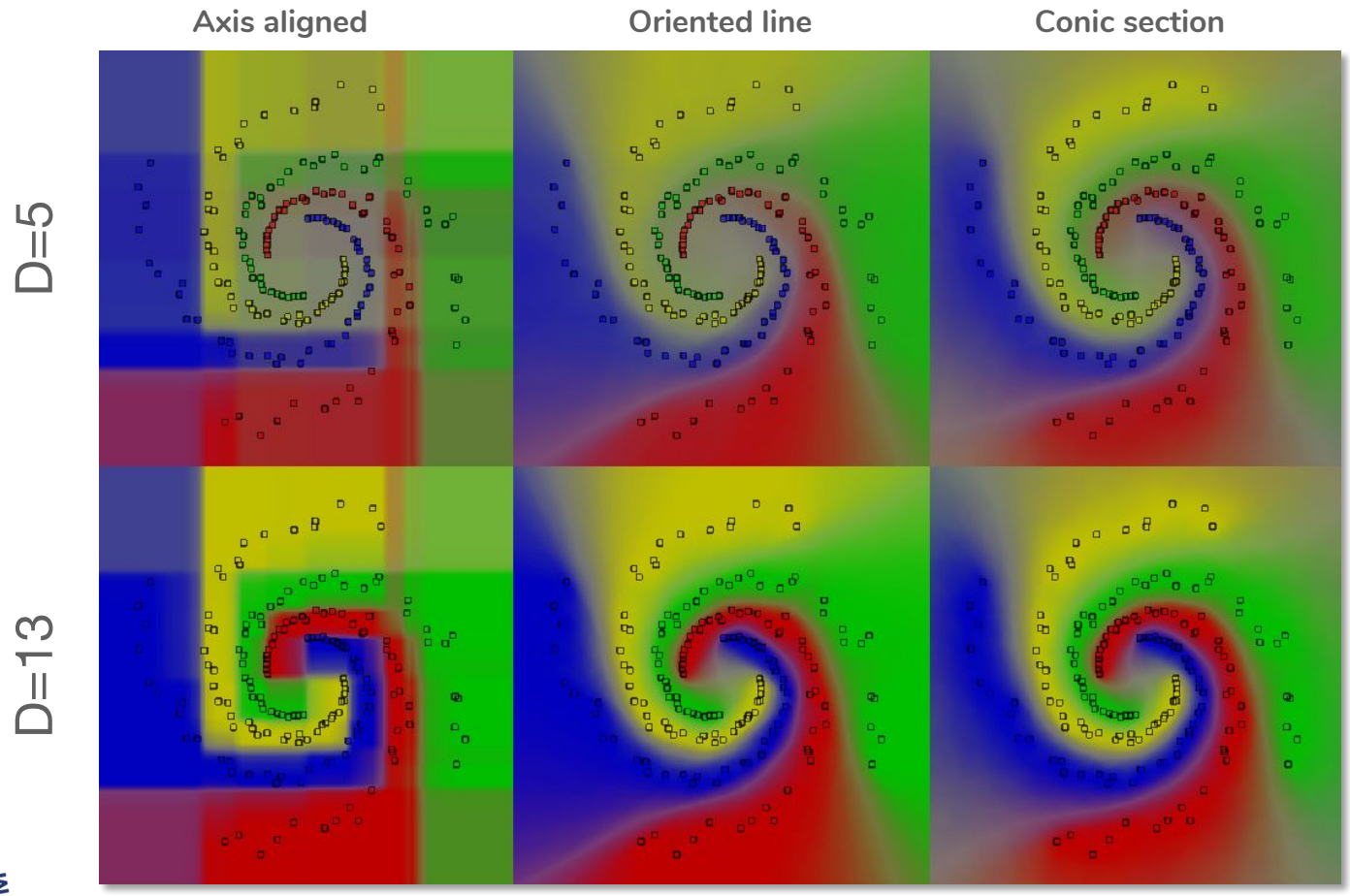
Toy example: randomness effects

Increasing Randomness



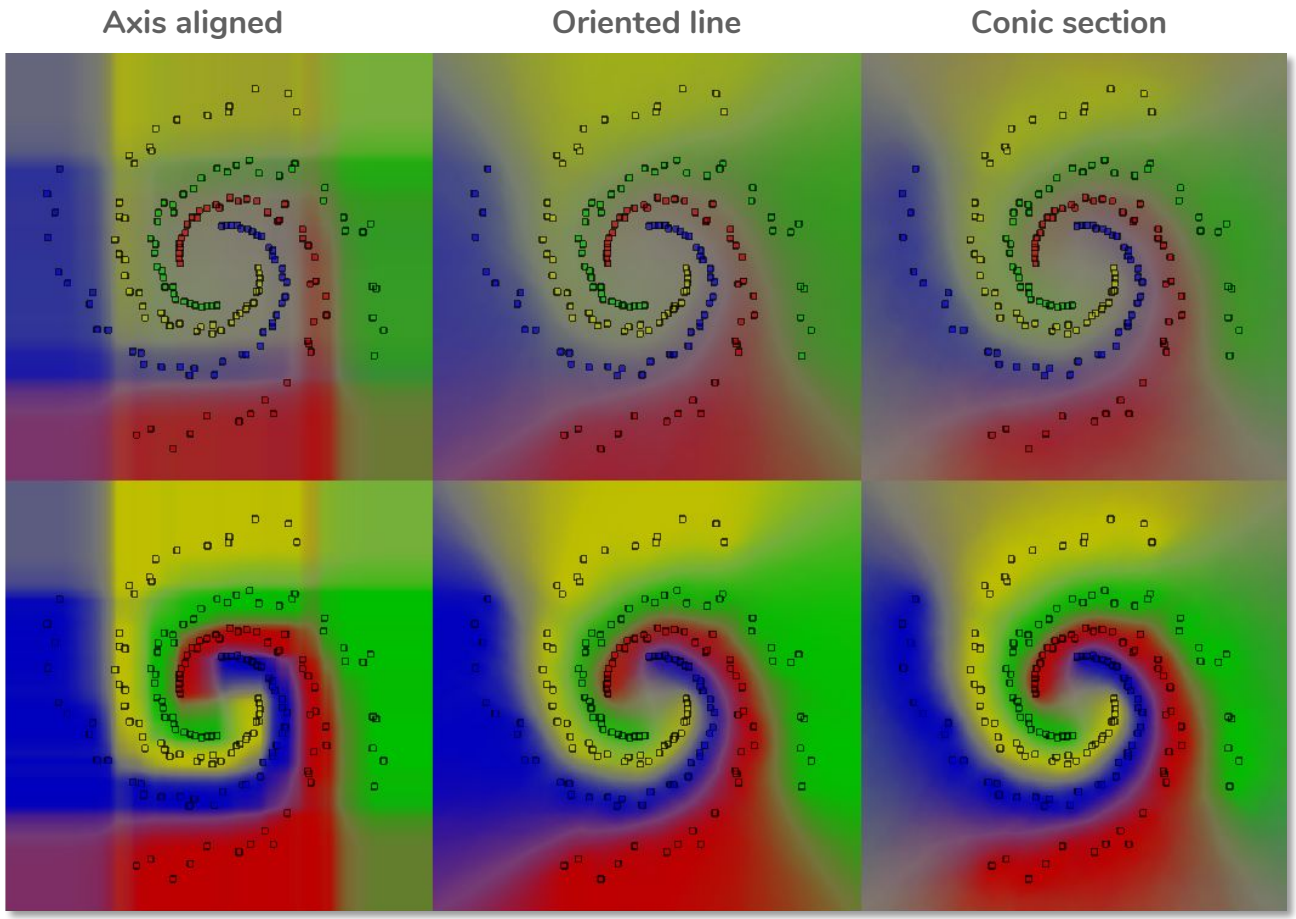
Toy example: randomness effects

Increasing Randomness



Toy example: randomness effects

Increasing Randomness



Classification forests in practice

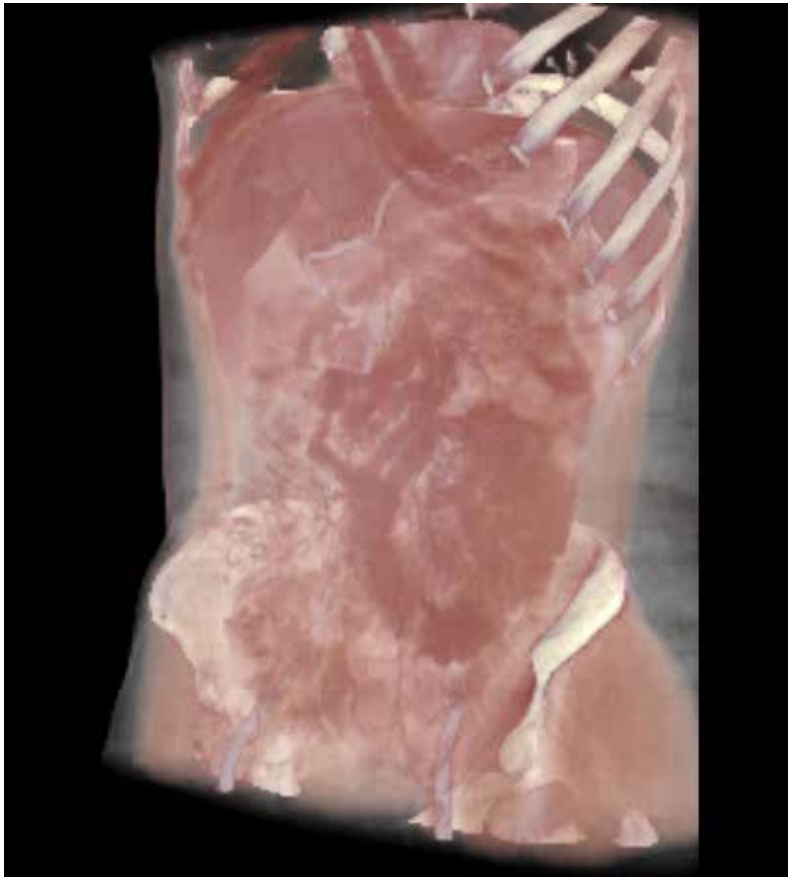
Body tracking in Microsoft Kinect for XBox 360

Input depth image (bg removed)

Inferred body parts posterior

Classification forests in practice

Multi-organ segmentation



Regression forests

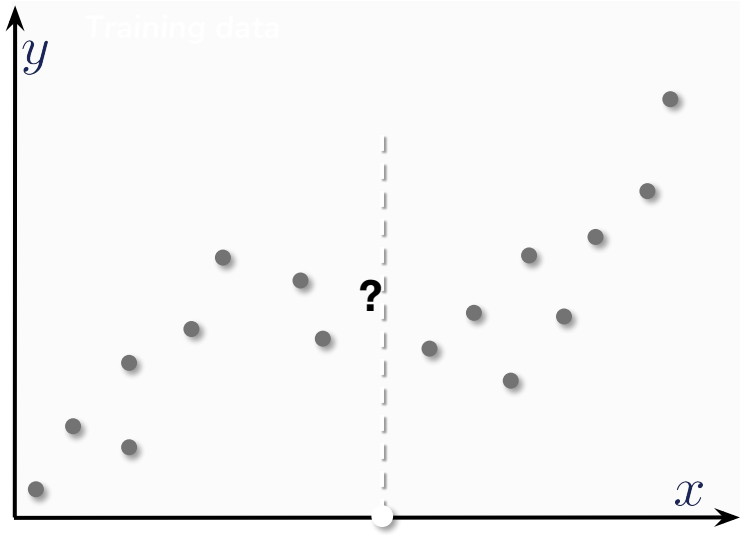
Supervised regression

Observation

$$\mathbf{X} = [x_1 \ \dots \ x_i \ \dots \ x_d]^T, \mathbf{X} \in \mathbb{R}^d$$

Output

$$\mathbf{Y} \in \mathbb{R}^{d'}$$

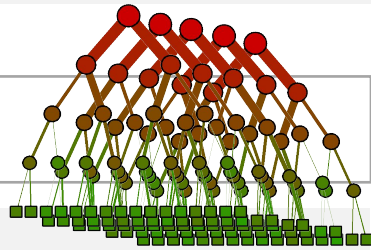


Given a training set: $\{(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_n, \mathbf{Y}_n), \dots$

GOAL: learn the conditional distribution $P(\mathbf{Y}|\mathbf{X})$

Regression forest model

Decision Forest



Components

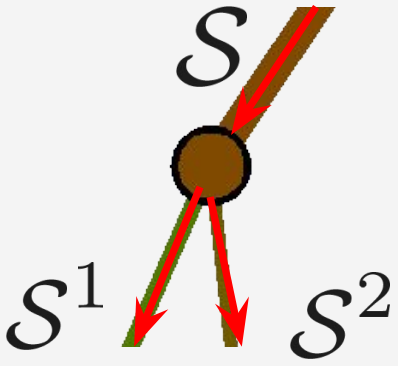
- Node decision function: f_θ
- Decision function param: θ
- Objective function $\mathbf{I(S, S_l, S_r)}$
- Leaf prediction model $\mathbf{P(Y|X)}$
- Forest prediction model

$$\mathbf{P(Y|X)} = \frac{1}{T}\sum_{t=1}^{T} P_t(\mathbf{Y|X})$$

Parameters

- Maximal tree depth: D
- Minimum Population: MinPop
- Minimum Energy: Imin
- Bagging ratio: $\mathrm{R} = |S_0^t|/|S_0|$
- Number of tries/node $\mathrm{Ntry} = |\mathcal{T}|$
- Number of trees: T

Objective function: reducing uncertainty



Information gain

$$I = H(\mathcal{S}) - \sum_{i \in \{1,2\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H(\mathcal{S}^i)$$

$$H(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \int_y p(y|x) \log p(y|x)\, dy$$

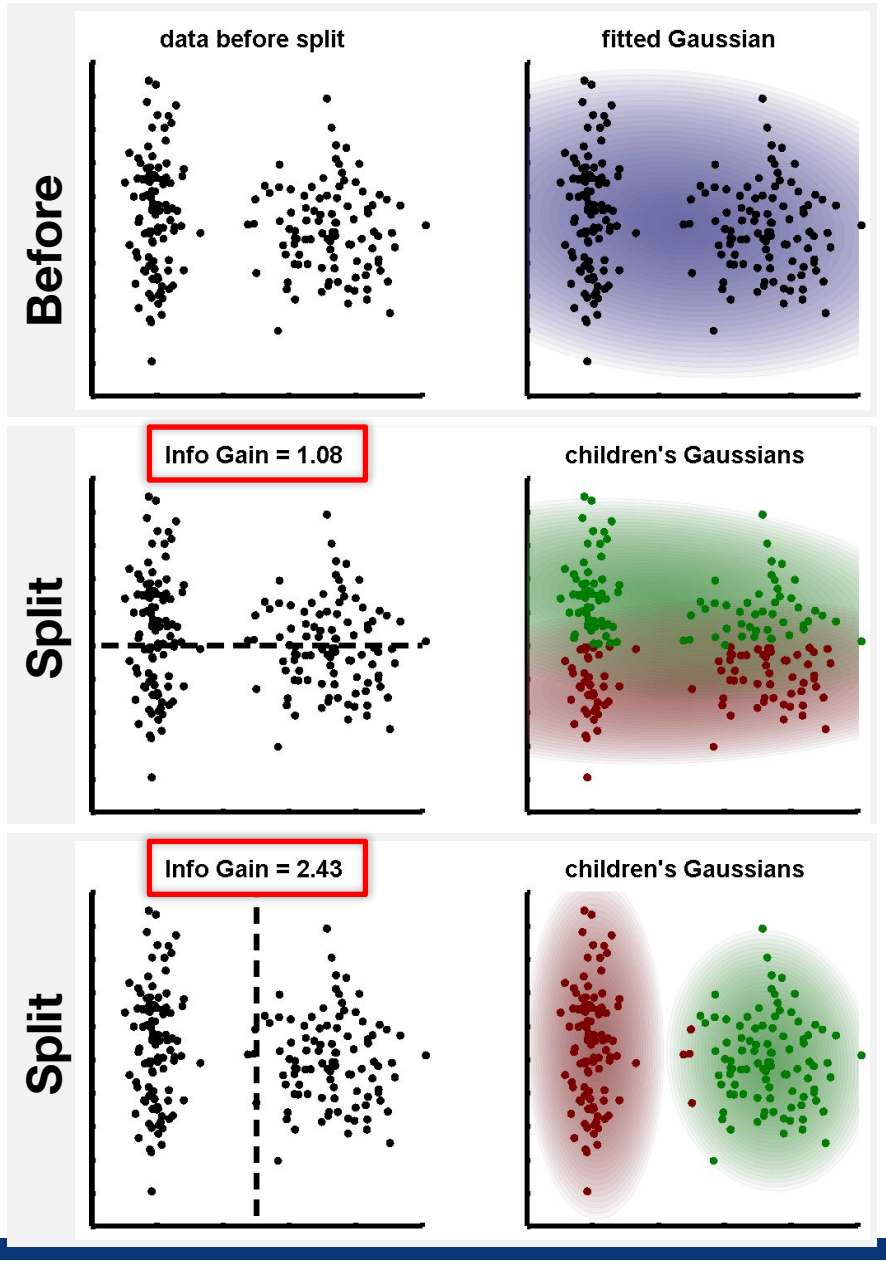
$$p(y|x) \sim N\left(y; \overline{y}, \sigma_y^2(x)\right)$$

Differential entropy of Gaussian

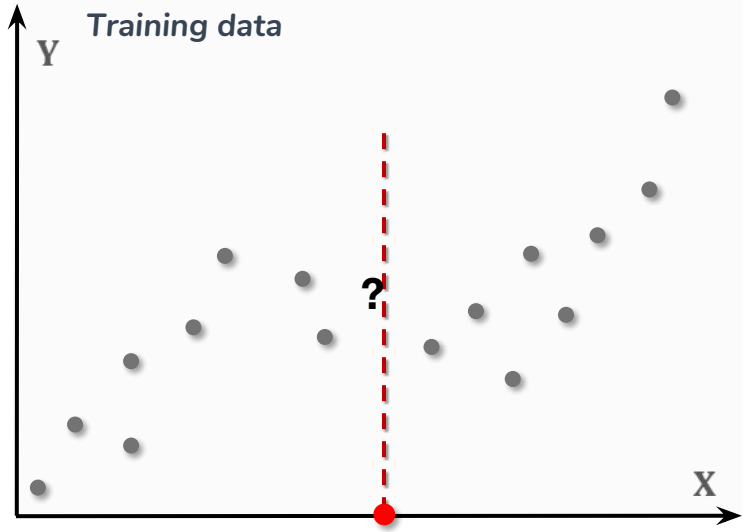
$$H(\mathcal{S}) = \frac{1}{2} \log\left((2\pi e)^d |\Lambda(\mathcal{S})|\right)$$

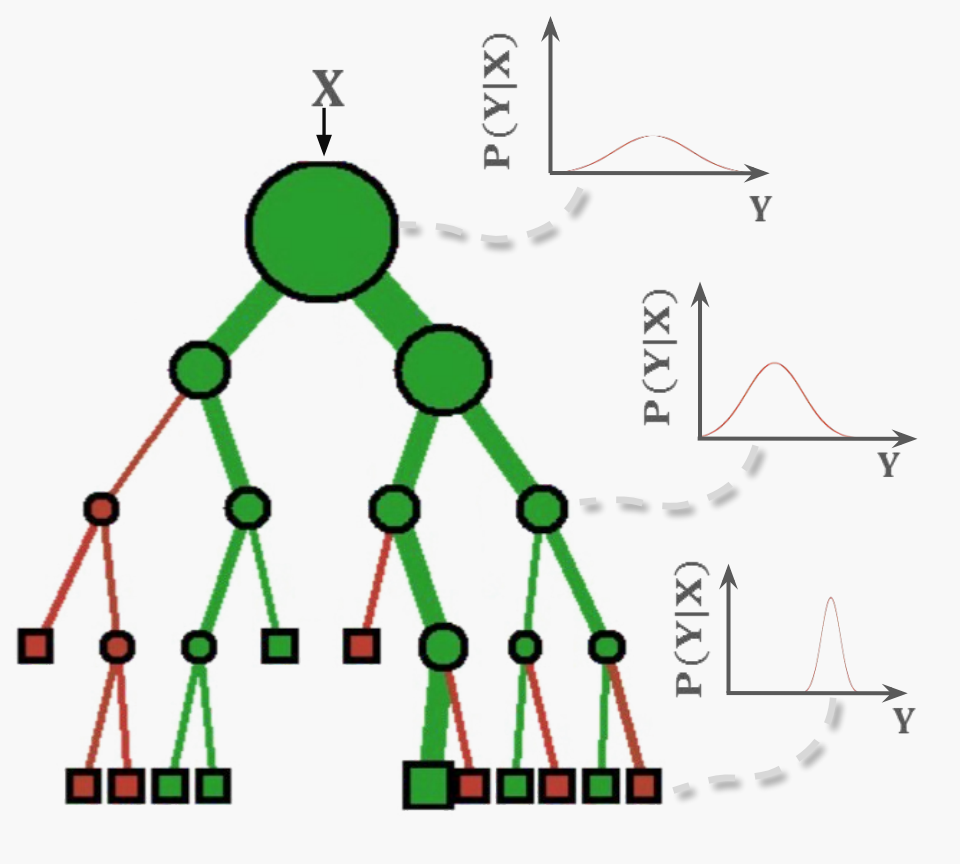
Node training

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \mathcal{T}} I$$

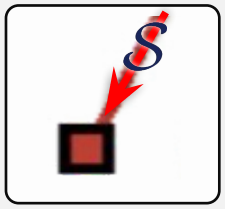


Objective function: reducing uncertainty



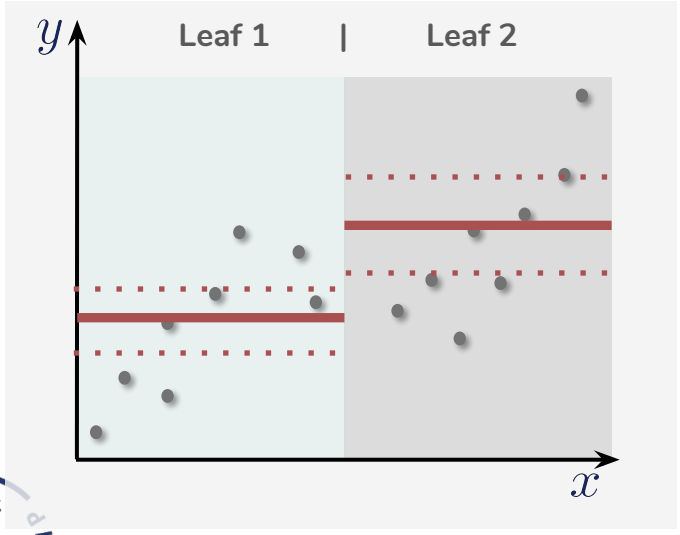


Leaf model

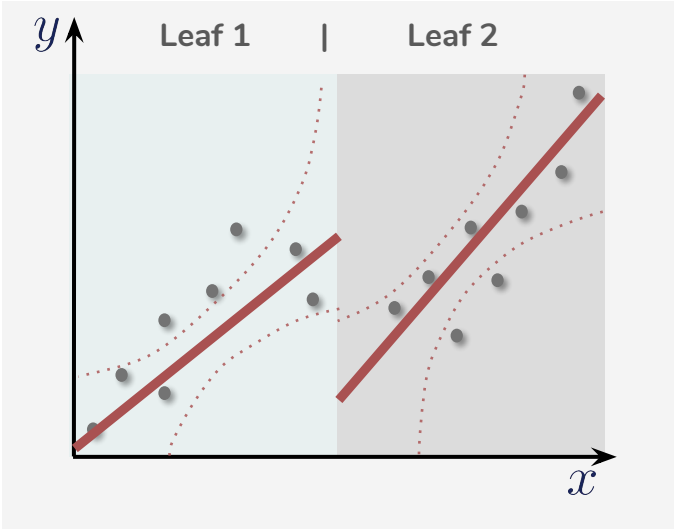


What do we at a leaf?
Probabilistic model
Parameters estimated from set of points **reaching the leaf**

Probabilistic constant



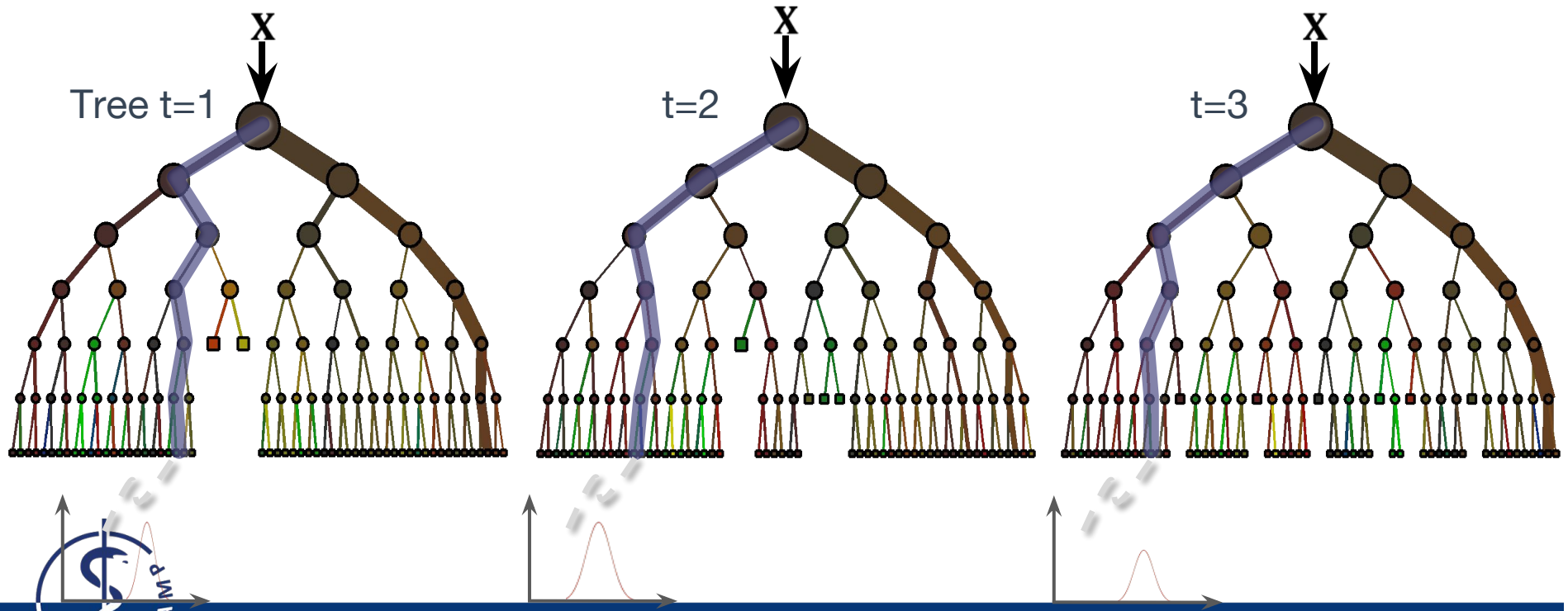
Probabilistic linear



Forest prediction: posterior averaging

Forest output probability

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} P_t(\mathbf{Y}|\mathbf{X})$$



Forest prediction: posterior averaging

Forest output probability $\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t}^{T} \boldsymbol{P}_t(\mathbf{Y}|\mathbf{X})$

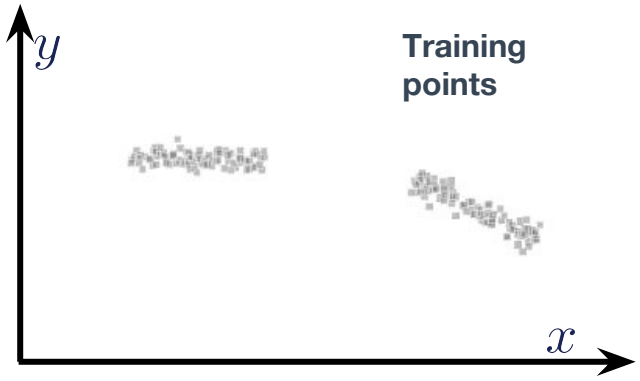
- Perform prediction using mathematical expectation:

$$\hat{\mathrm{Y}} = E(\mathrm{Y}|\mathrm{X}) = \int_{\mathrm{Y}} \mathrm{Y}\, \mathrm{P}(\mathrm{Y}|\mathrm{X})\mathrm{dY}$$

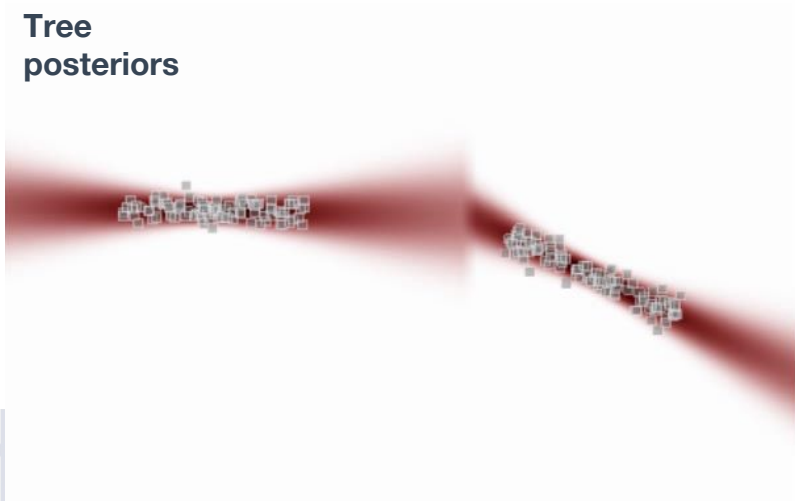
- Perform prediction using first mode:

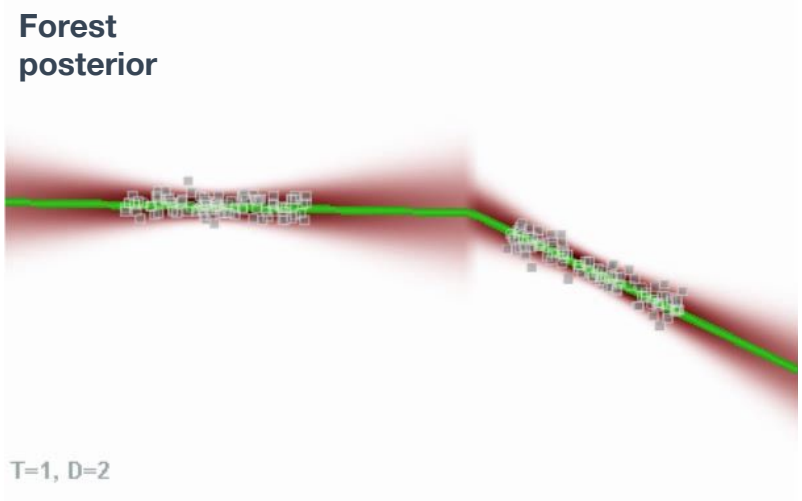
$$\hat{\mathrm{Y}} = \mathrm{argmax}_{\mathrm{Y}}\, \mathrm{P}(\mathrm{Y}|\mathrm{X})$$

Toy example: probabilistic non-linear regression

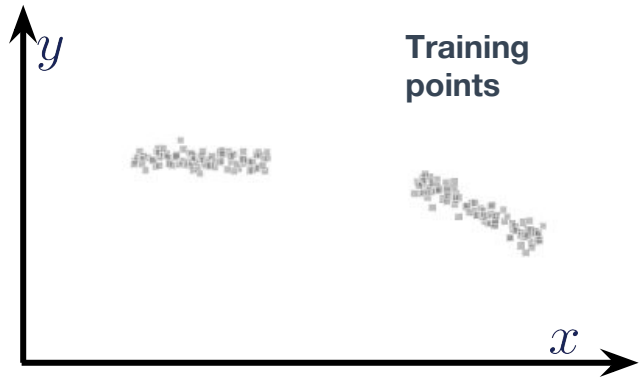


- Axis-aligned decision functions
- Smooth interpolation
- Confidence decreases far from data

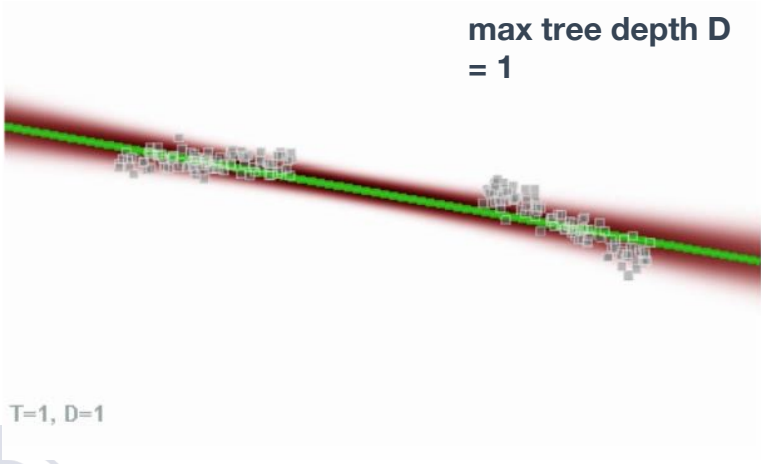


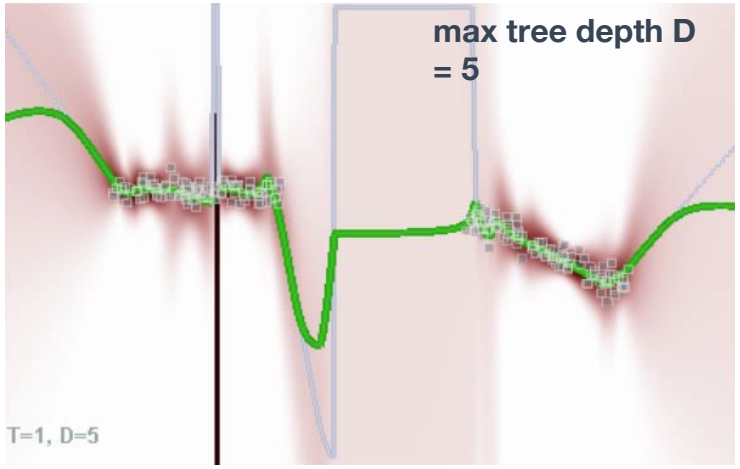


Toy example: effects of the tree depth

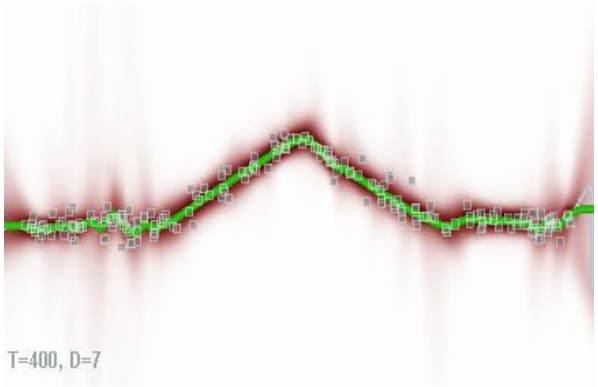


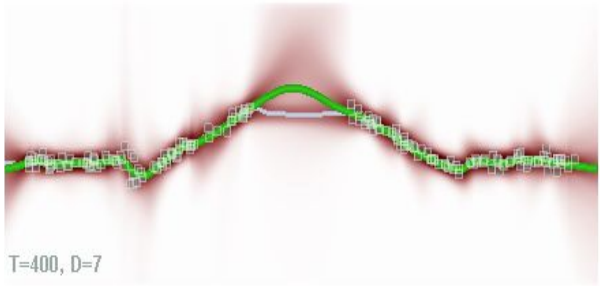
- Axis-aligned decision functions
- Smooth interpolation
- Confidence decreases far from data

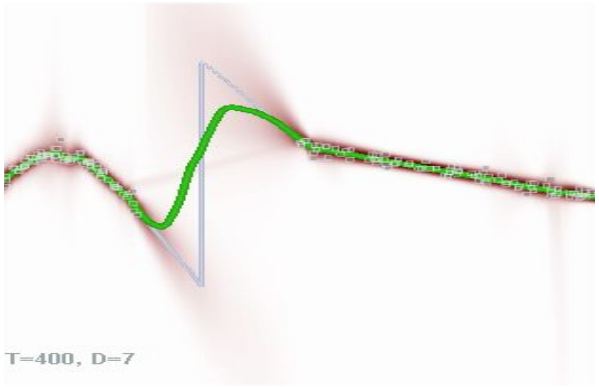


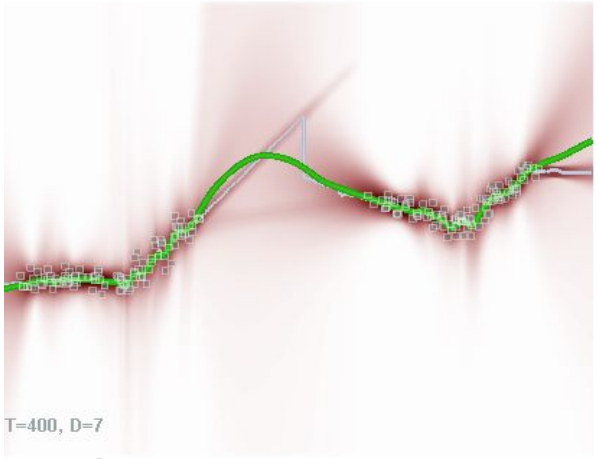


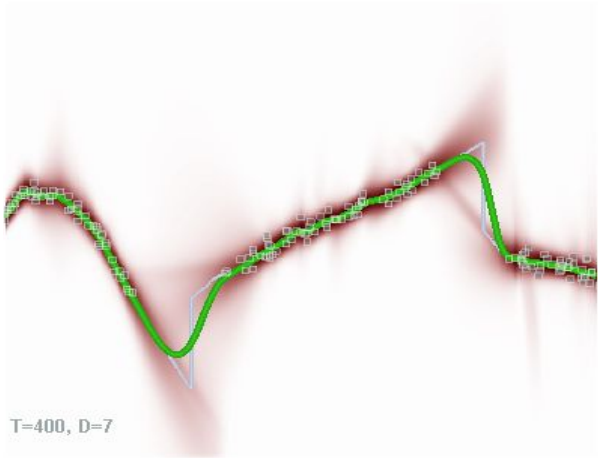
Toy example: effects of the tree depth

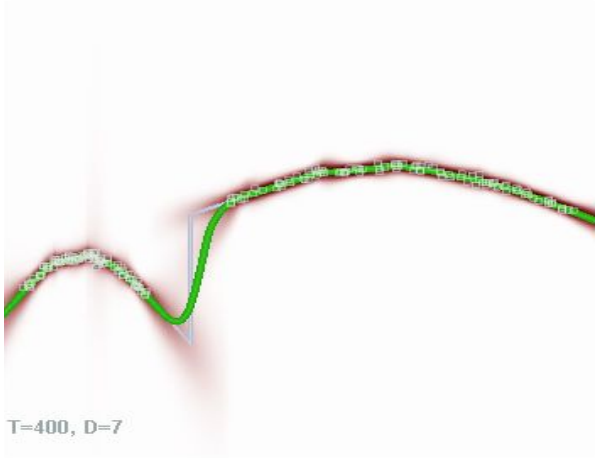




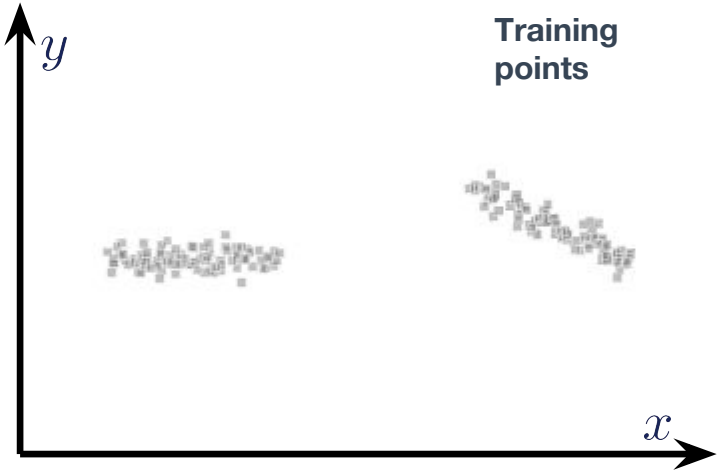




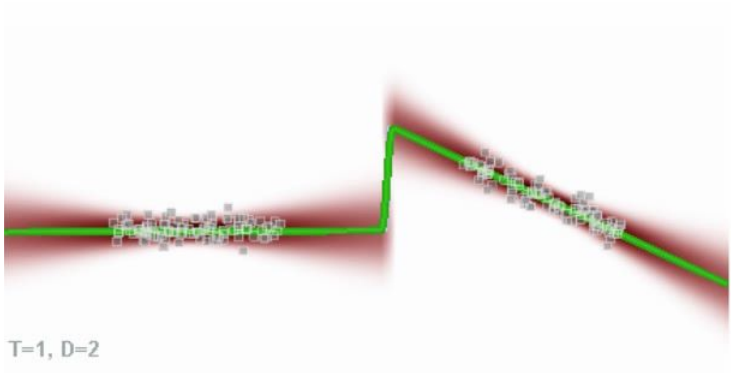


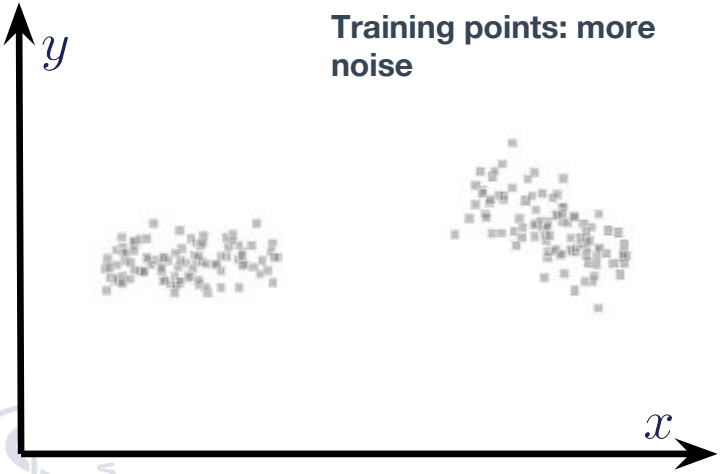


Toy example: effects of noise

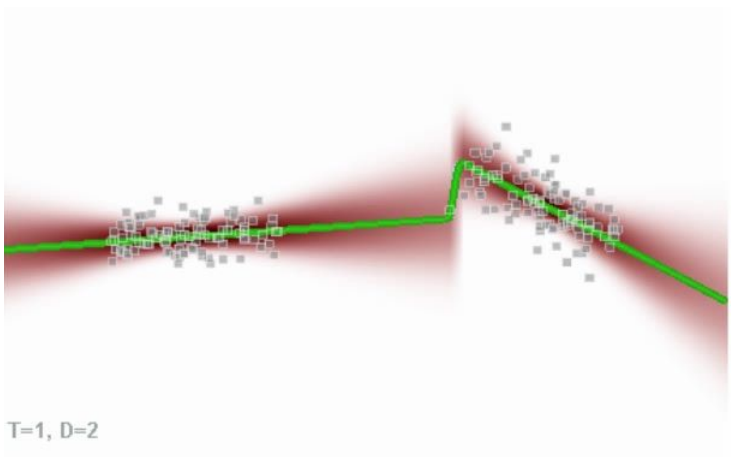


Training points



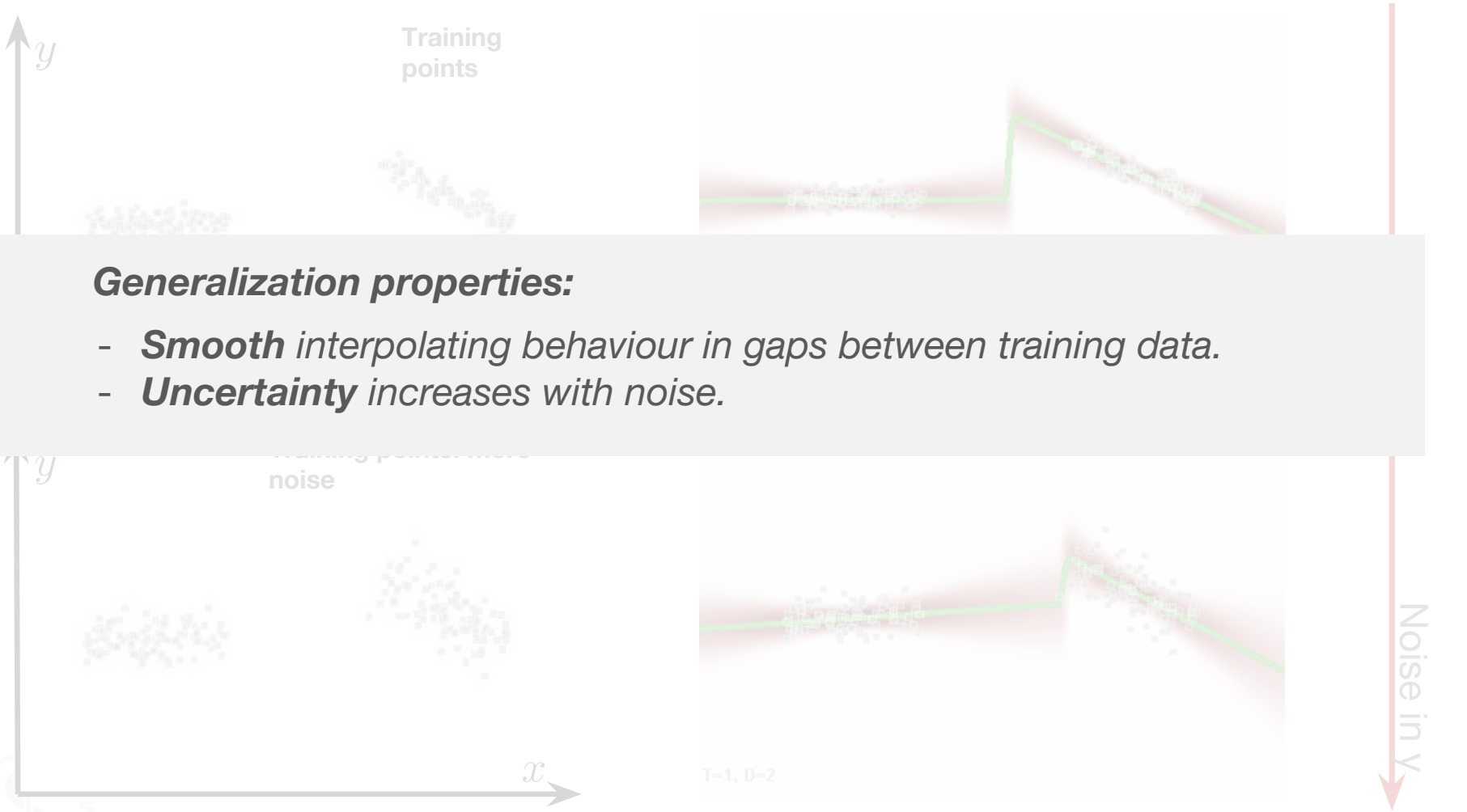


Training points: more noise



Noise in y

Toy example: effects of noise

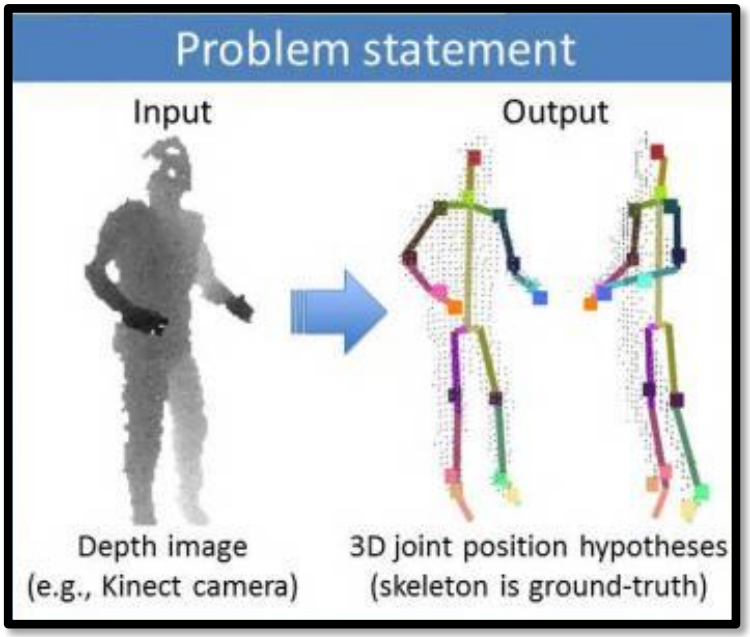


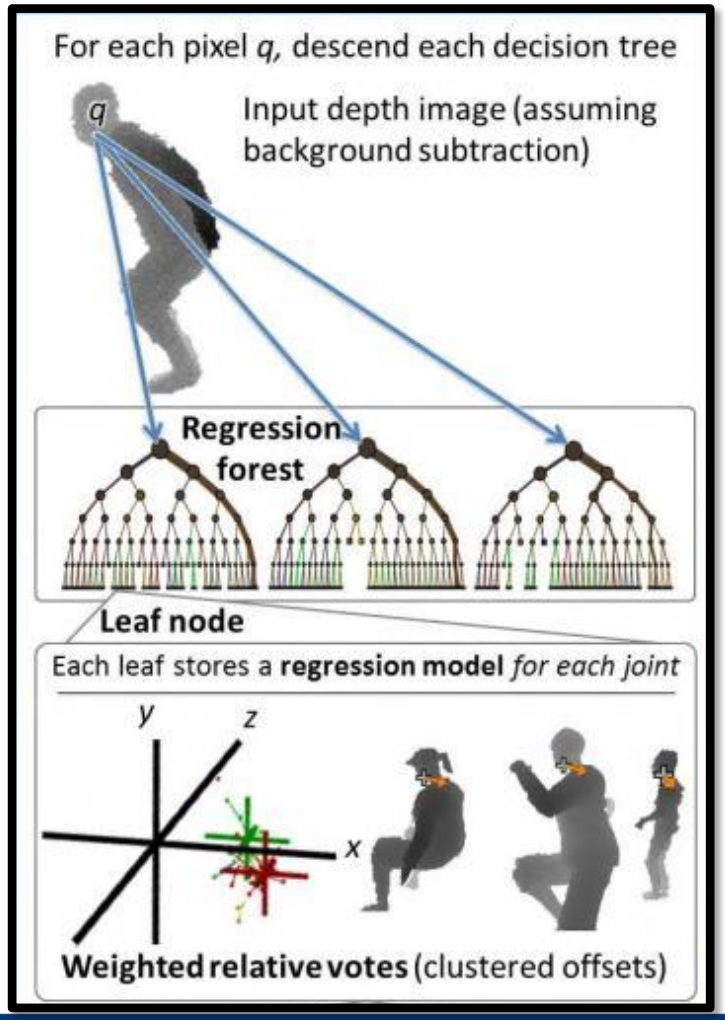
Generalization properties:

- ***Smooth*** *interpolating behaviour in gaps between training data.*
- ***Uncertainty*** *increases with noise.*

Regression forests in practice

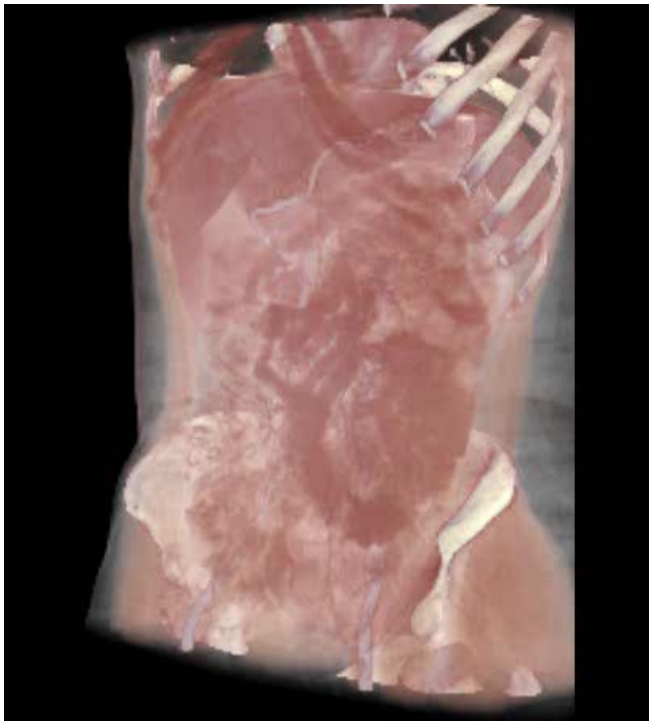
Body tracking in Microsoft Kinect for XBox 360



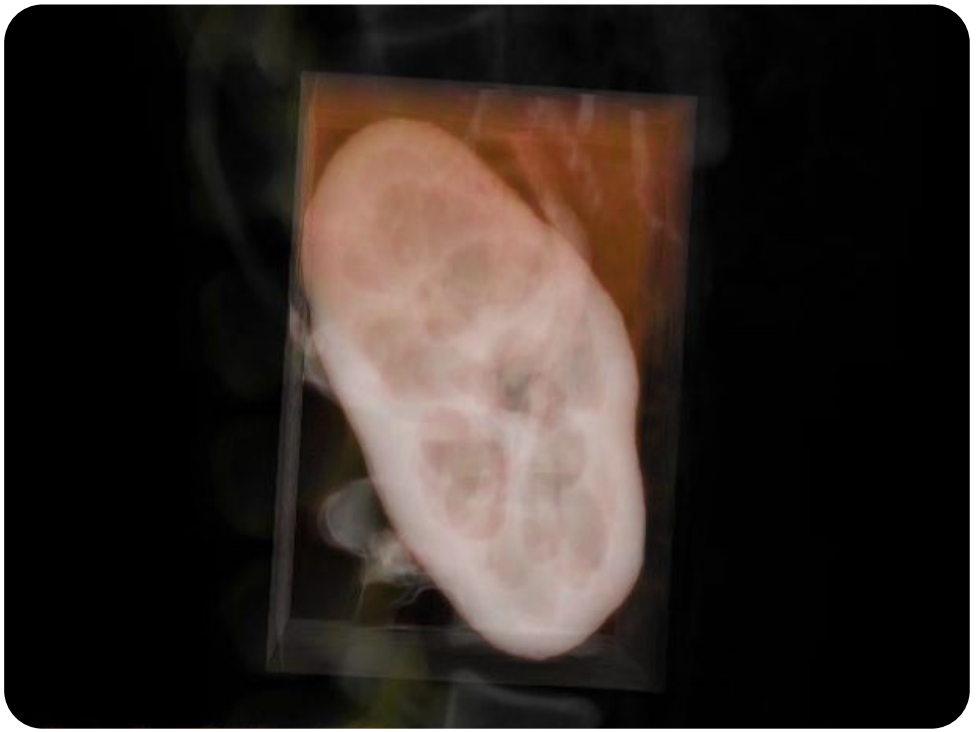


Regression forests in practice

Efficient multi-organ localization



Input CT scan (volumetric rendering)



Output anatomy localization

References

[Criminisi, A., Shotton, J., Konukoglu, E.: **Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning**, Technical Report MSR-TR-2011-114, Microsoft Research (2011)]

[Criminisi, Antonio, et al. "**Regression Forests for Efficient Anatomy Detection and Localization in CT Studies.**" *MCV* 2010 (2010): 106-117.]

[L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. **Classification and Regression Trees (CART)**. 1984]