UNIVERSITÄT BONN

Prof. Dr. Juergen Gall

Random Forests

MA-INF 2213 - Advanced Computer Vision

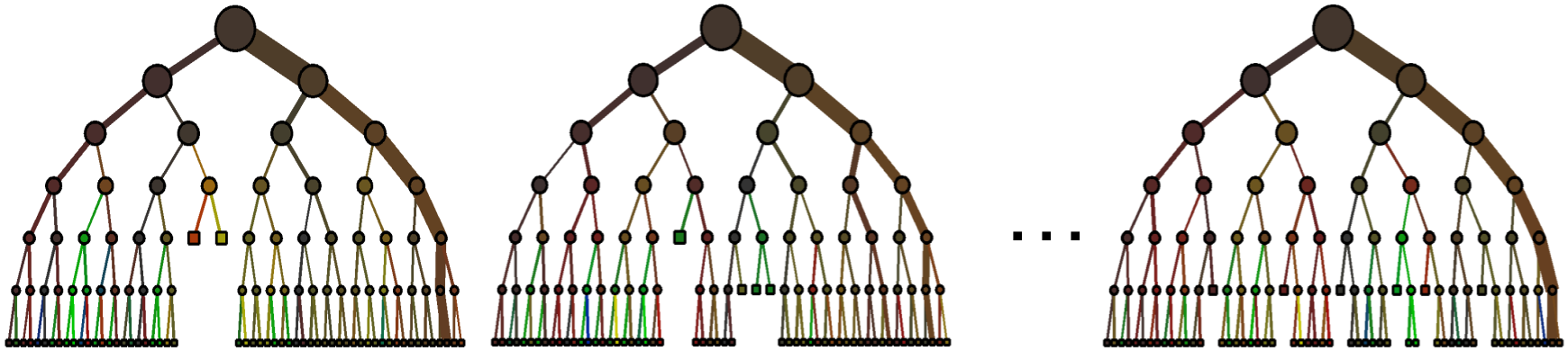
SS25

Organization

UNIVERSITÄT BONN

- Exercise:
  - First exercise on 2.5.
  - Tentative: 5 Sheets
  - Tutorial: PyTorch and GPU cluster Bender on 16.5. (tentative) https://www.hpc.uni-bonn.de/en/systems/bender

Random Forests



[L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. **Classification and Regression Trees (CART)**. 1984]
[T. Ho. **Random Decision Forests.** 1995]
[Y. Amit and D. Geman. **Shape quantization and recognition with randomized trees.** 1997]
[L. Breiman. **Random forests.** 2001]
[A. Criminisi and J. Shotton. **Decision Forests in Computer Vision and Medical Image Analysis.** 2013]

Computer vision applications

[V. Lepetit and P. Fua. **Keypoint recognition using randomized trees.** 2005]

[F. Moosman, B. Triggs and F. Jurie. **Fast discriminative visual codebooks using randomized clustering forests.** 2006]

[J. Shotton, M. Johnson and R. Cipolla. **Semantic Texton Forests for Image Categorization and Segmentation.** 2008]

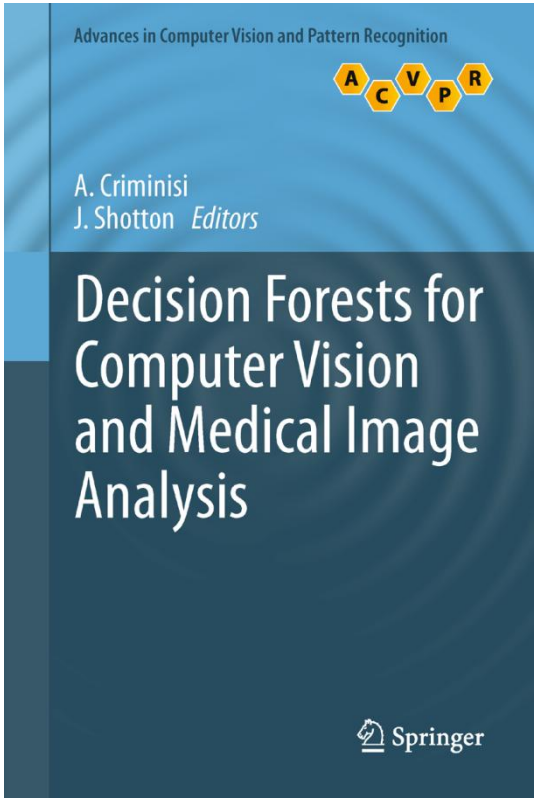
[J. Gall and V. Lempitsky. **Class-Specific Hough Forests for Object Detection.** 2009]

[J. Shotton et al. **Real-Time Human Pose Recognition in Parts from a Single Depth Image.** 2011]

Literature

UNIVERSITÄT BONN

http://research.microsoft.com/en-us/projects/decisionforests/

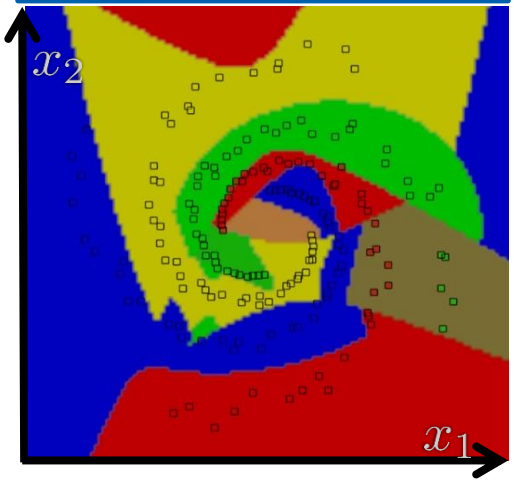


A. Criminisi el al**. Decision Forests for Computer Vision and Medical Image Analysis**, Springer 2013

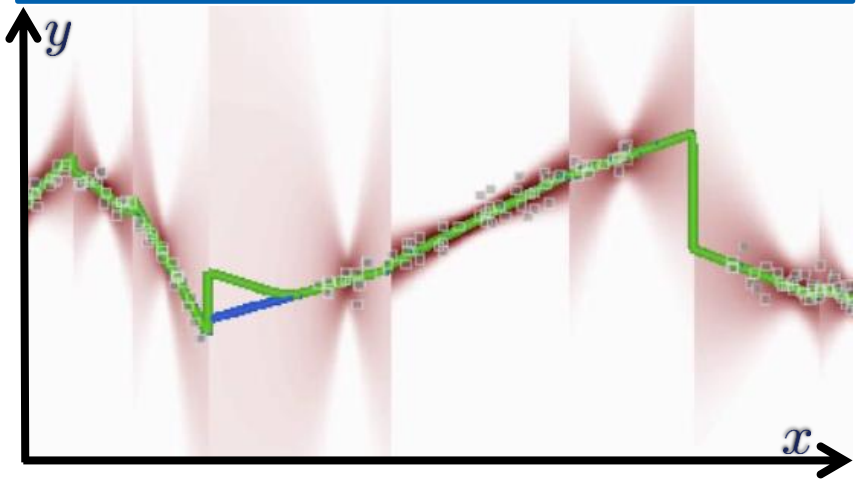
What can decision forests do?

UNIVERSITÄT BONN

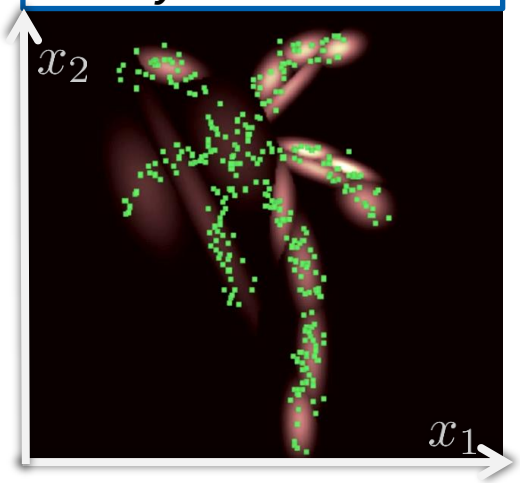
Classification forests



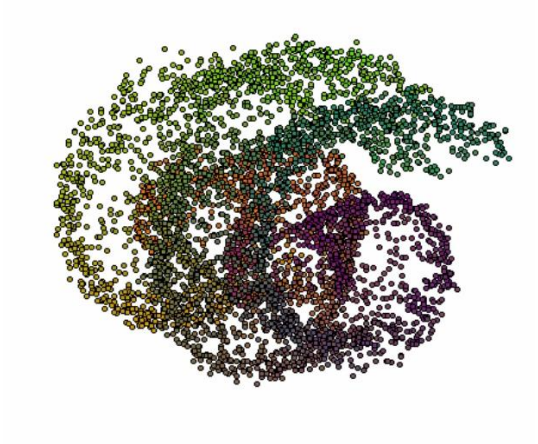
Regression forests



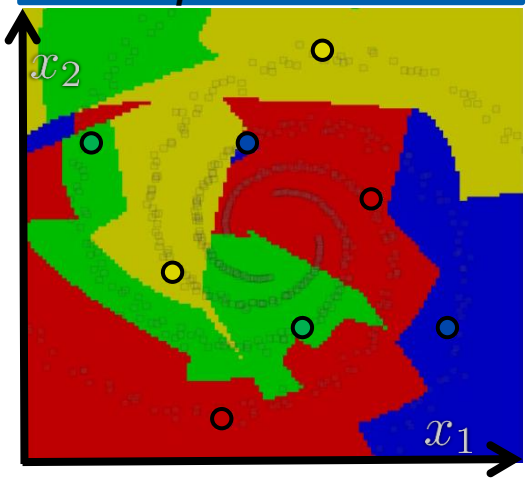
Density forests



Manifold forests



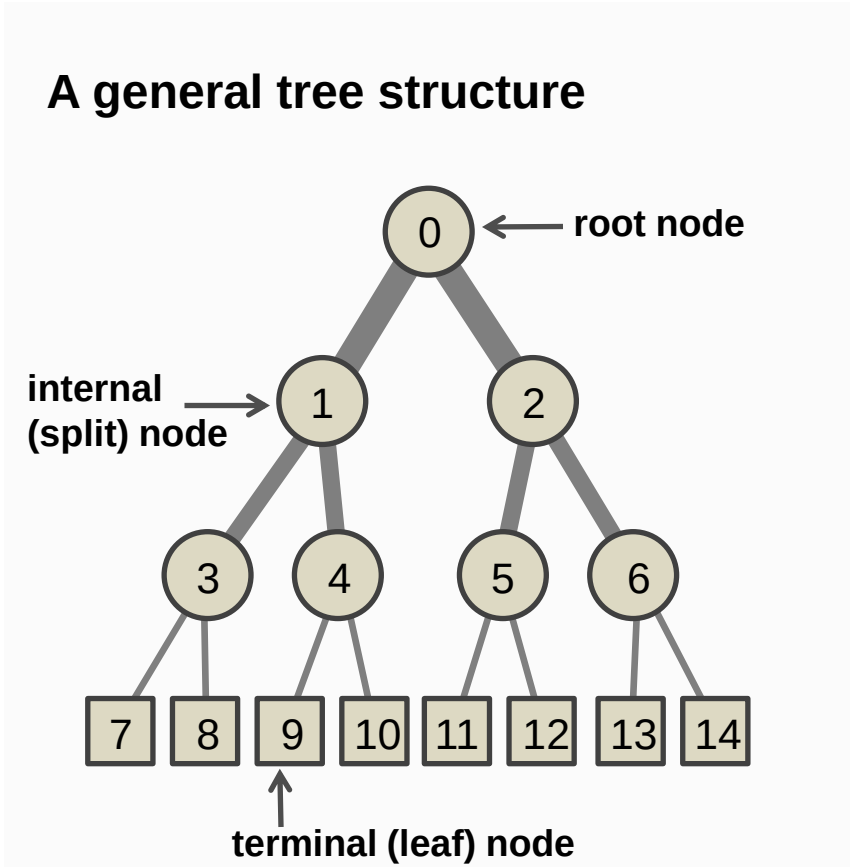
Semi-supervised forests

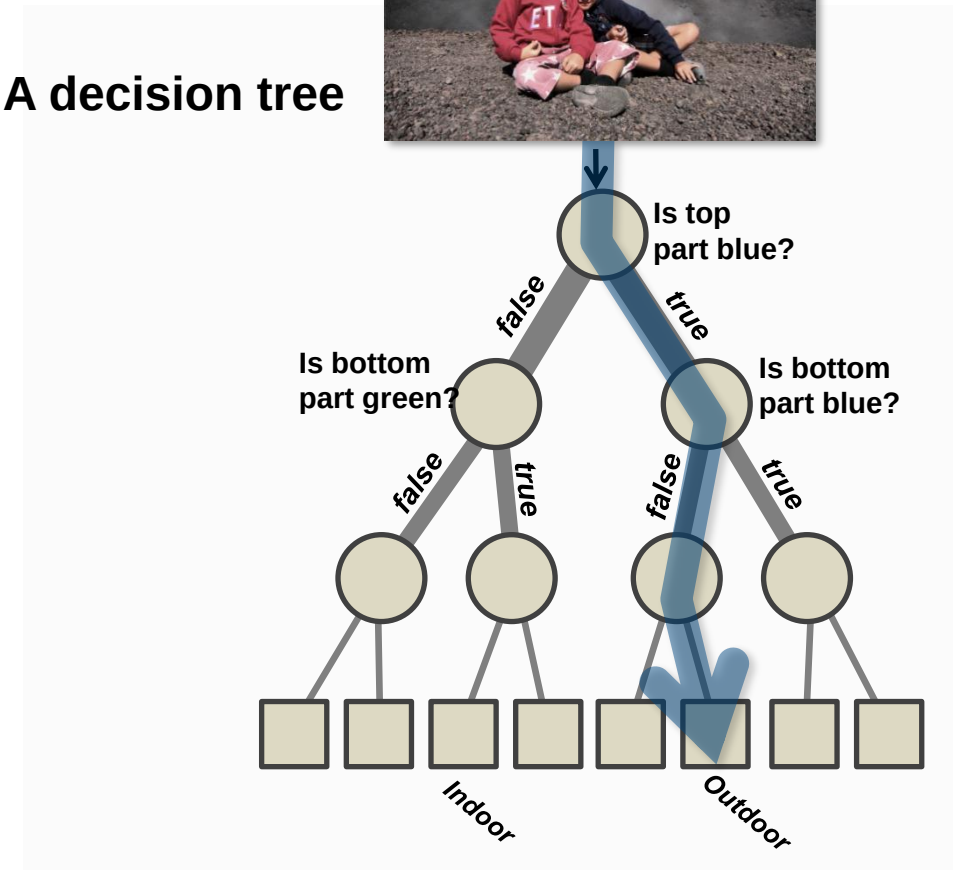


Overview

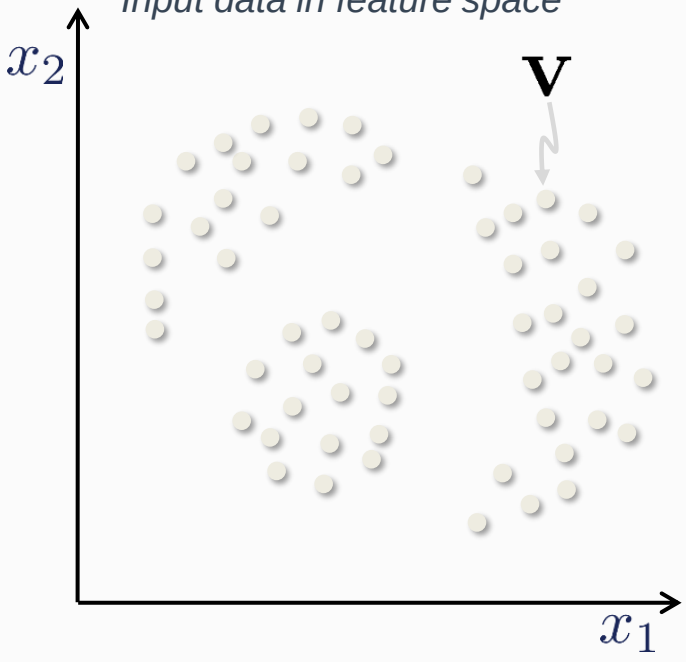
- Classification forests
- Regression forests
- Other variants

Decision trees

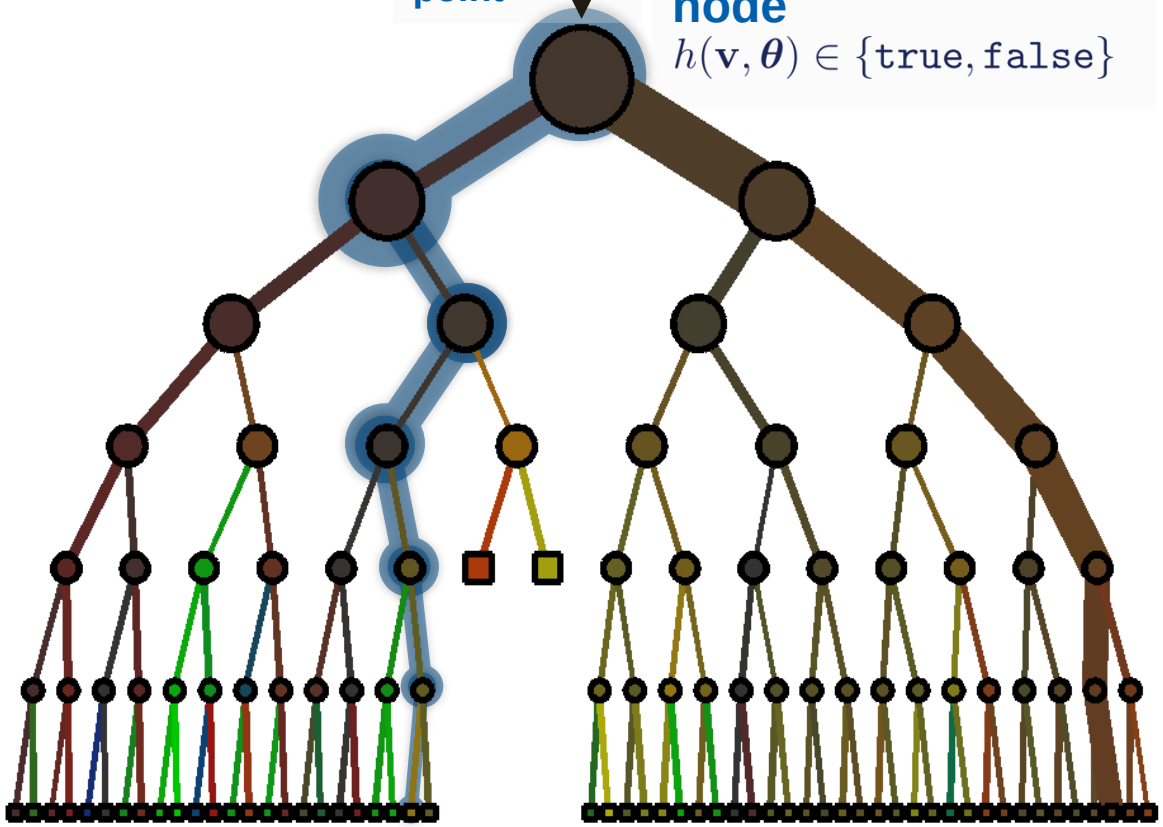




Decision tree



$$\mathbf{v} = (x_1, \cdots, x_d) \in \mathbb{R}^d$$



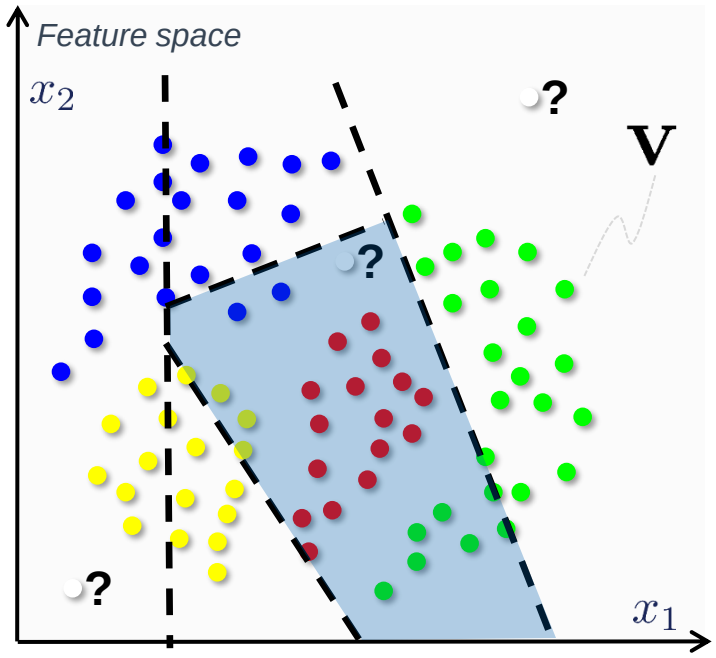
Split the data at node

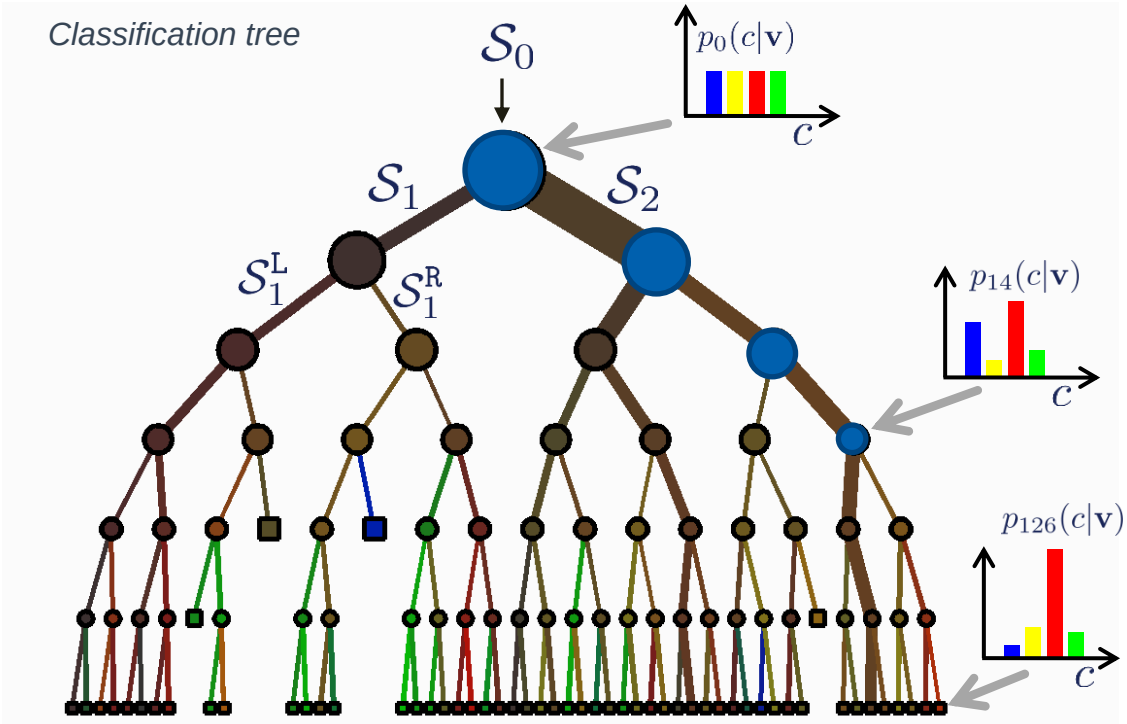
$h(\mathbf{v}, \boldsymbol{\theta}) \in \{\texttt{true}, \texttt{false}\}$

Prediction at leaf

$p(c|\mathbf{v})$

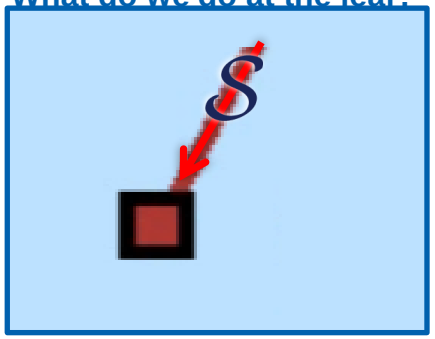
Classification tree

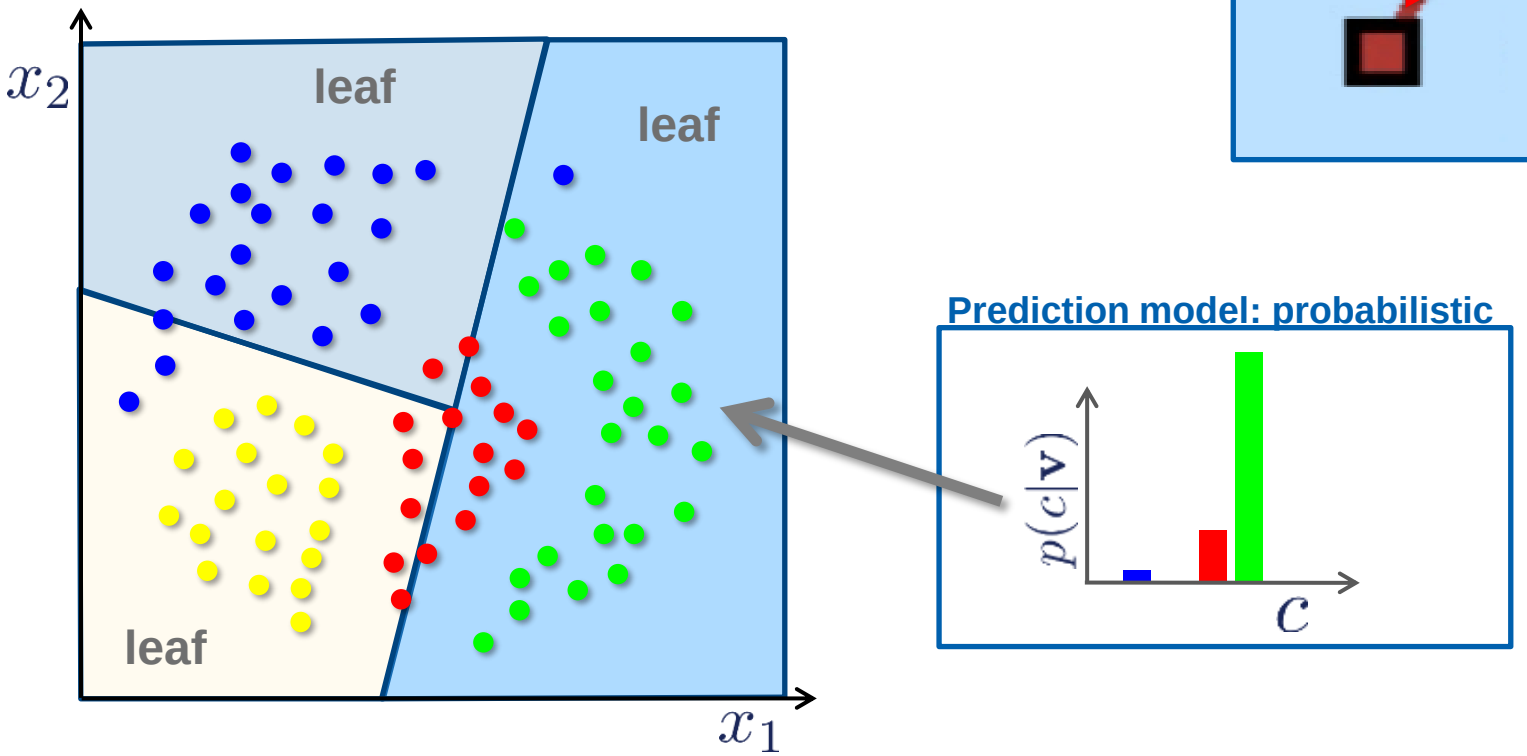




Figures adapted from A. Criminisi

Classification forest: the prediction model

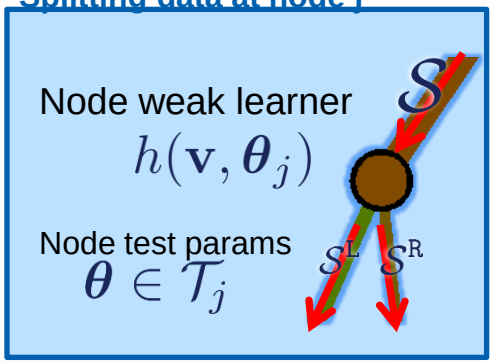




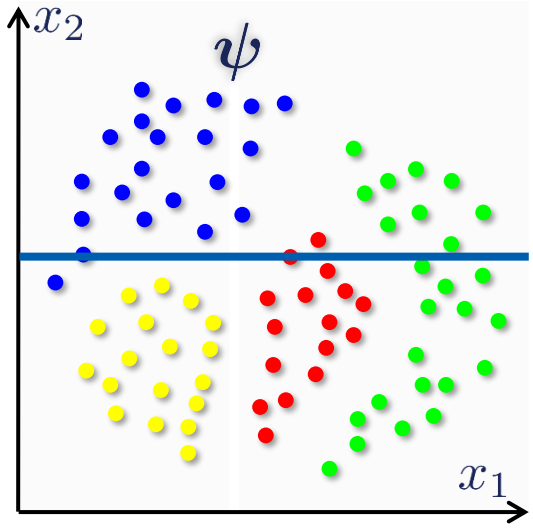
Classification forest: the weak learner model

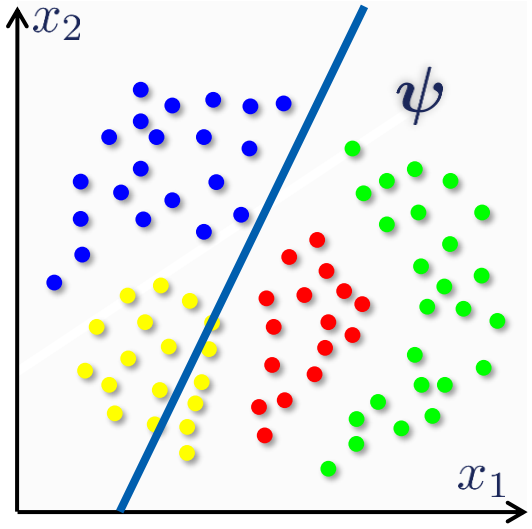
UNIVERSITÄT BONN

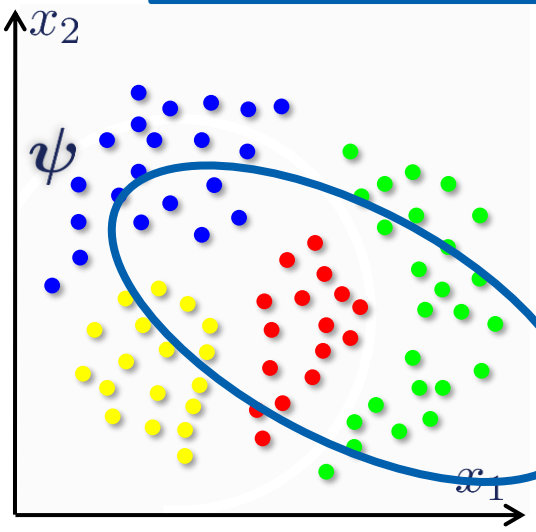
Splitting data at node j



Examples of weak learners







Weak learner: axis aligned

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

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \; x_2 \; 1)^\top$

With $\boldsymbol{\psi} = (1\,0\,\psi_3)$ or $\boldsymbol{\psi} = (0\,1\,\psi_3)$

Weak learner: oriented line

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

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \; x_2 \; 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^3$ a generic line in homog. coordinates.

Weak learner: conic section

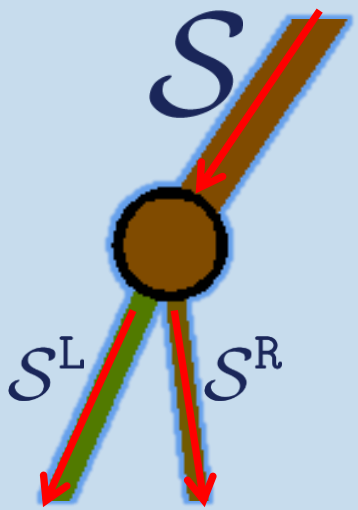
$$h(\mathbf{v}, \boldsymbol{\theta}) = \left[\tau_1 > \boldsymbol{\phi}^\top(\mathbf{v}) \; \boldsymbol{\psi} \; \boldsymbol{\phi}(\mathbf{v}) > \tau_2\right]$$

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \; x_2 \; 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^{3\times 3}$ a matrix representing a conic.

In general $\boldsymbol{\phi}$ may select only a very small subset of features $\boldsymbol{\phi}(\mathbf{v}) : \mathbb{R}^d \rightarrow \mathbb{R}^{d'+1}, \;\; d' << d$

Decision forest model: training and information gain



Information gain

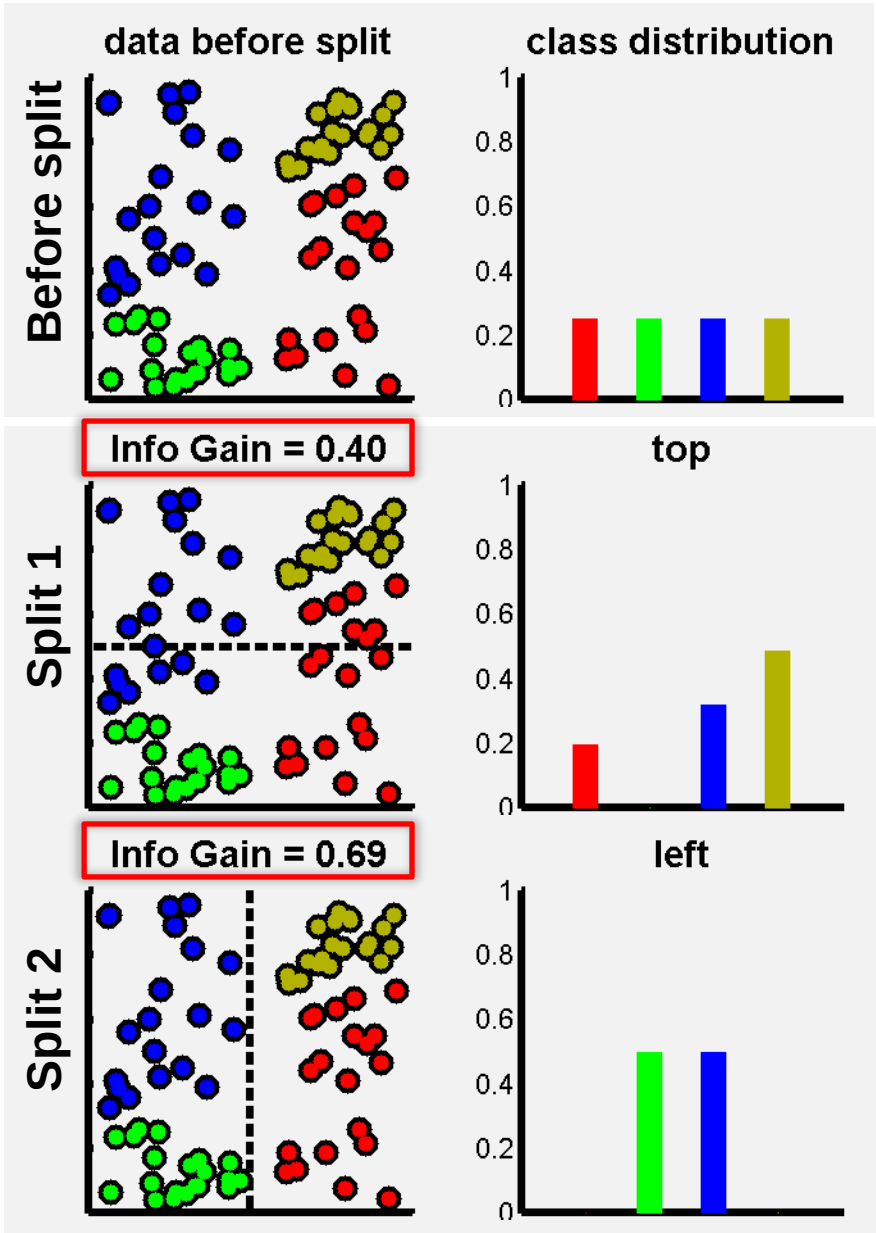
$$I(\mathcal{S}, \boldsymbol{\theta}) = H(\mathcal{S}) - \sum_{i \in \{\mathrm{L},\mathrm{R}\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H(\mathcal{S}^i)$$

Shannon's entropy

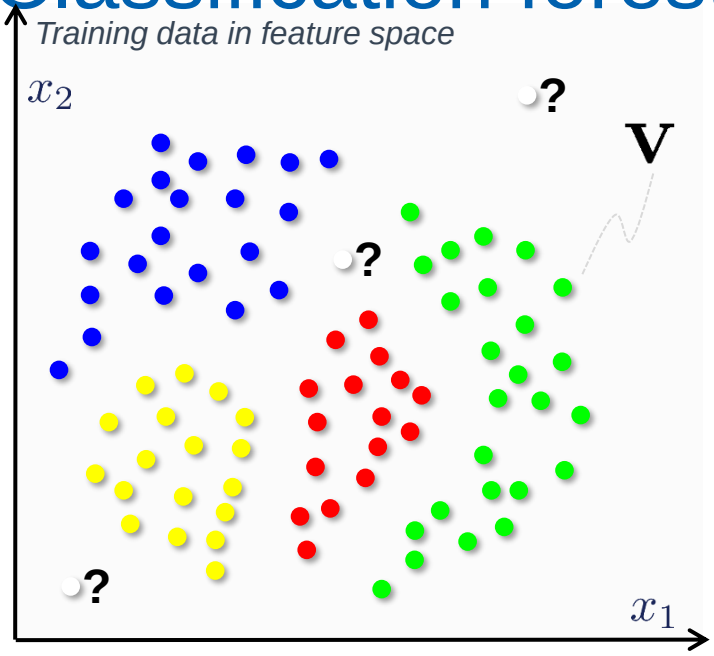
$$H(\mathcal{S}) = -\sum_{c \in \mathcal{C}} p(c) \log(p(c))$$

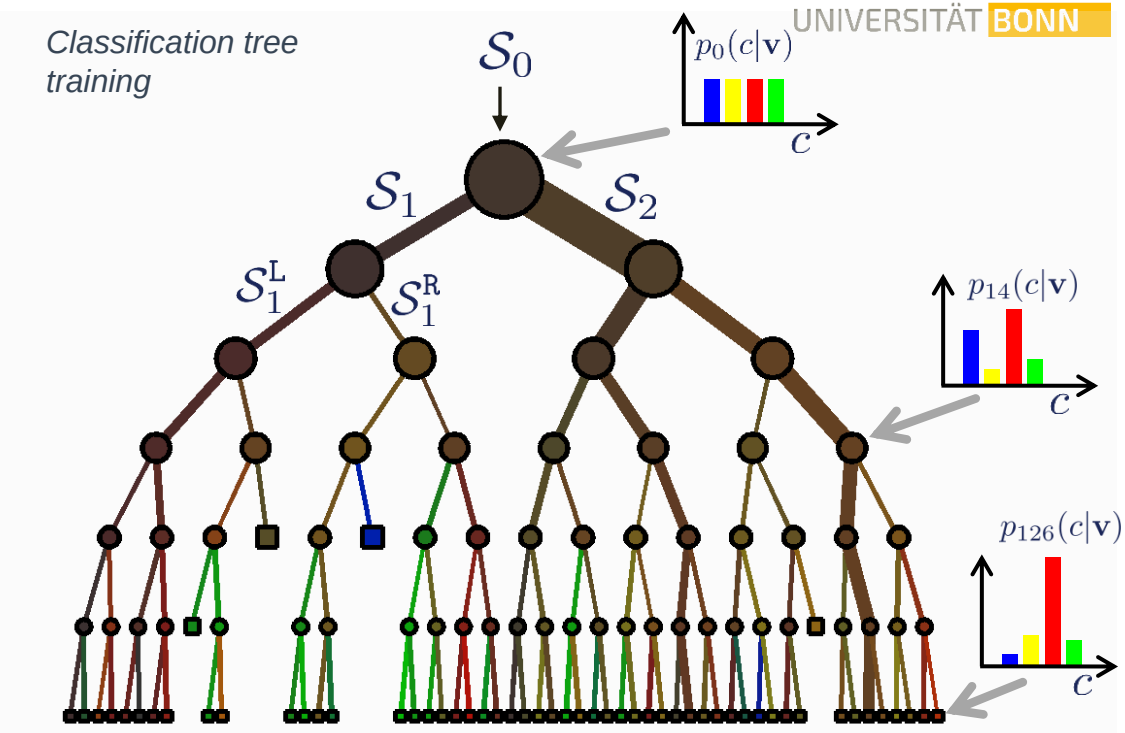
Node training

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta} \in \mathcal{T}_j} I(\mathcal{S}_j, \boldsymbol{\theta})$$



Classification forest





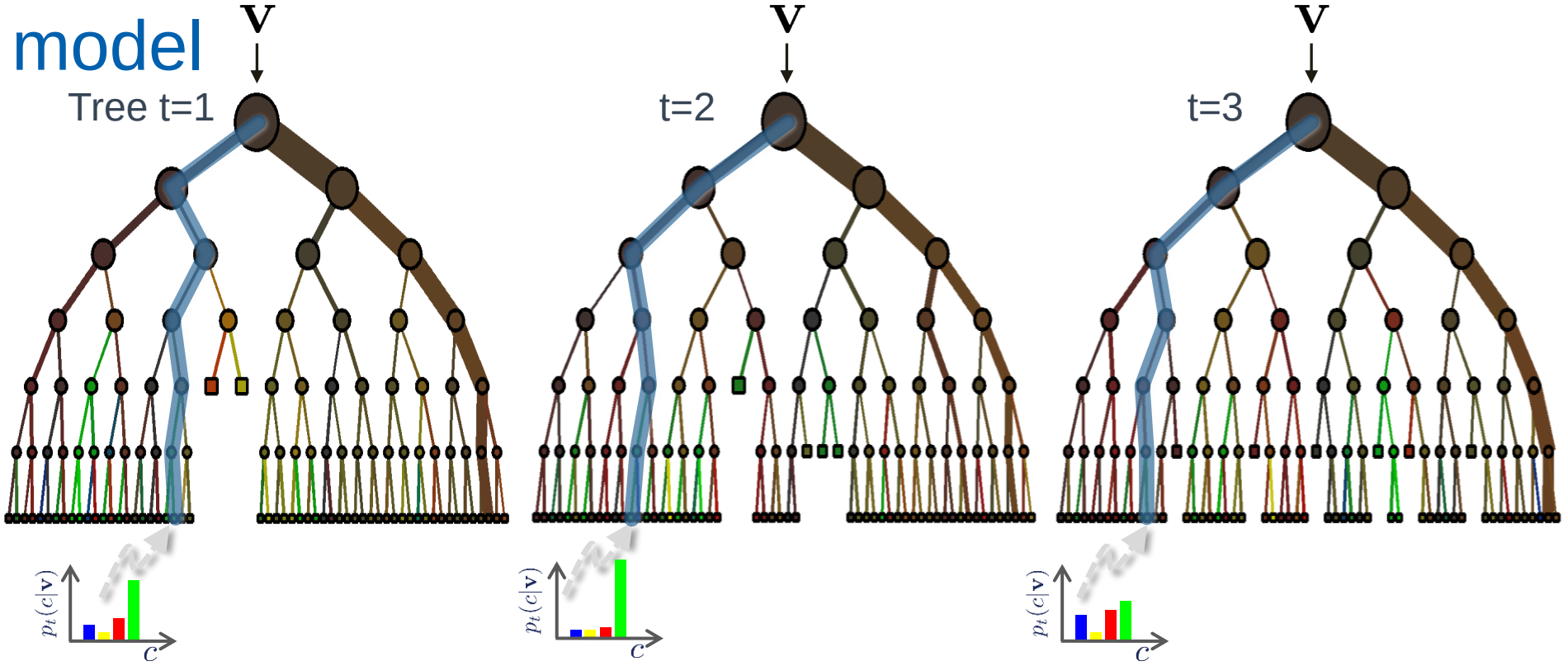
Input: $\mathbf{v} = (x_1, \cdots, x_d) \in \mathbb{R}^d$

Output: $p(c|\mathbf{v}) \quad c \in \mathcal{C} \quad \text{with} \quad \mathcal{C} = \{c_k\}$

Information gain: $I = H(\mathcal{S}_j) - \sum_{i=\mathrm{L,R}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i)$

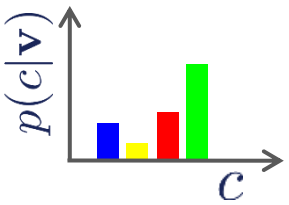
Entropy of discrete distribution: $H(\mathcal{S}) = -\sum_{c \in \mathcal{C}} p(c) \log(p(c))$

Classification forest: the ensemble model



The ensemble model

Forest output probability $$p(c|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(c|\mathbf{v})$$



Decision forest model: the randomness model

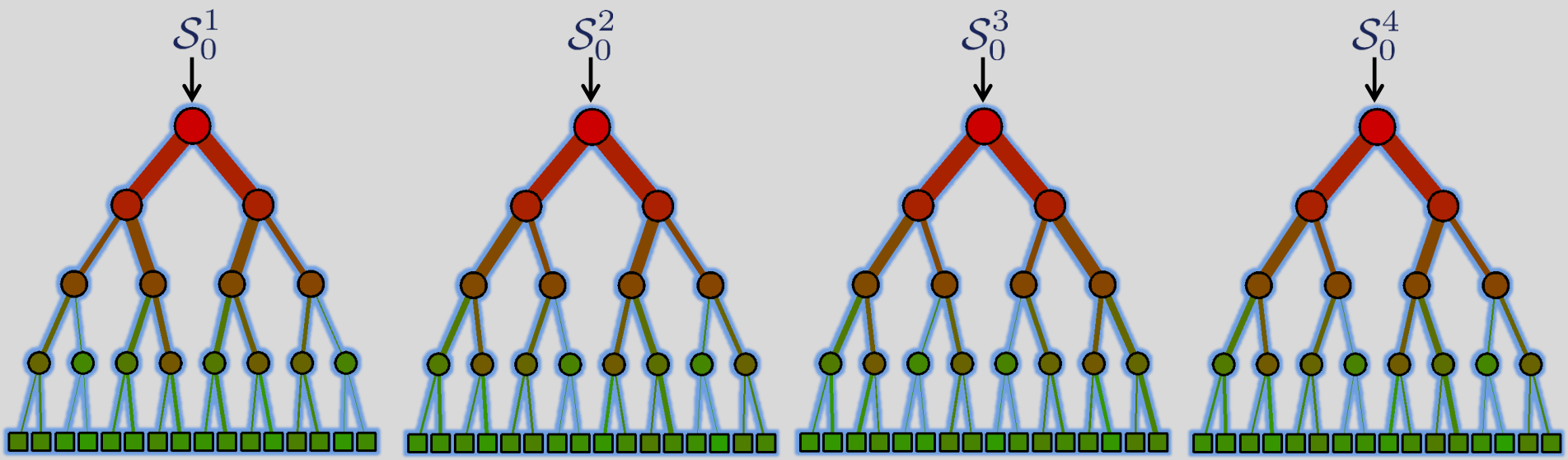
UNIVERSITÄT BONN

1) Bagging (randomizing the training set)

$\mathcal{S}_0$ The full training set

$\mathcal{S}_0^t \subset \mathcal{S}_0$ The randomly sampled subset of training data made available for the tree *t*

Forest training



Efficient training

Decision forest model: the randomness model

UNIVERSITÄT BONN

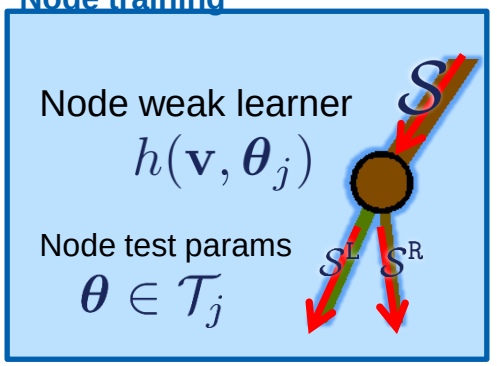
2) Randomized node optimization (RNO)

$\mathcal{T}$ The full set of all possible node test parameters

$\mathcal{T}_j \subset \mathcal{T}$ For each node the set of randomly sampled features

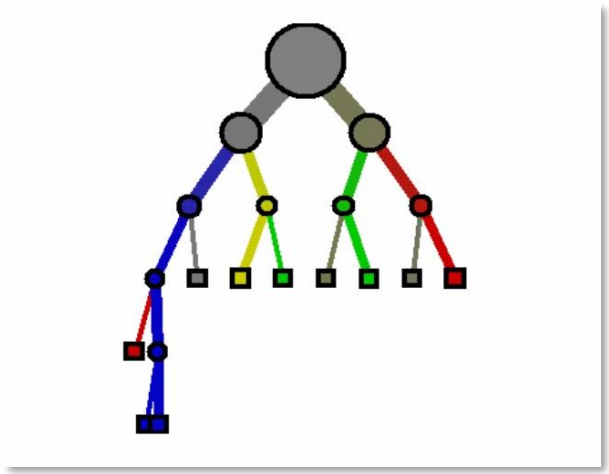
$\rho = |\mathcal{T}_j|$ Randomness control parameter.
For $\rho = |\mathcal{T}|$ no randomness and maximum tree correlation.
For $\rho = 1$ max randomness and minimum tree correlation.

Node training

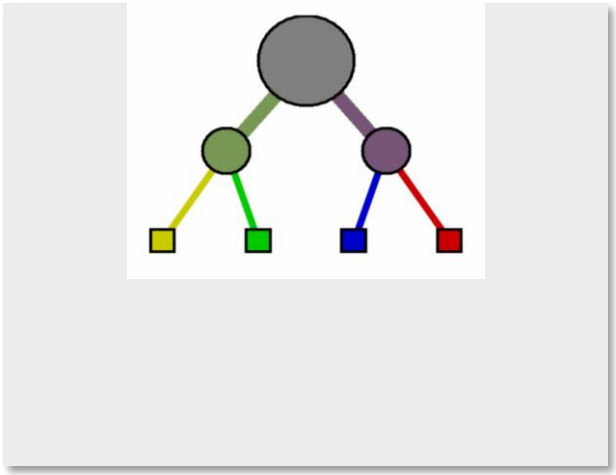


The effect of ρ

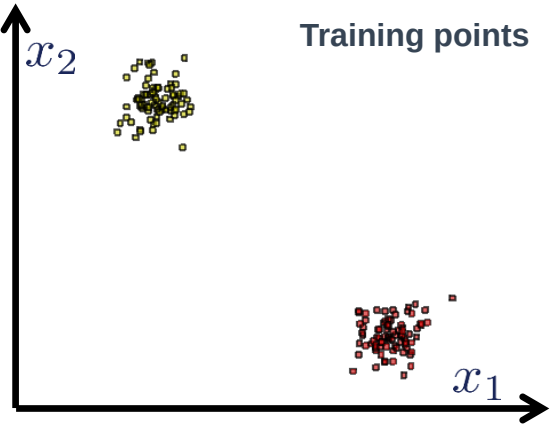
Small value of ρ; little tree correlation.



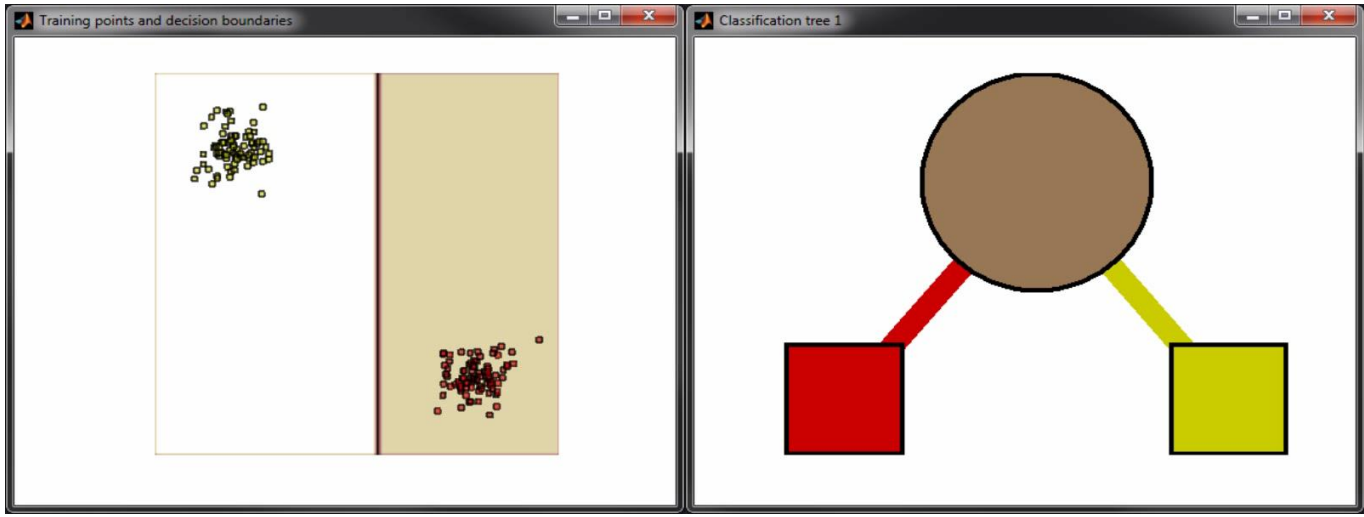
Large value of ρ; large tree correlation.



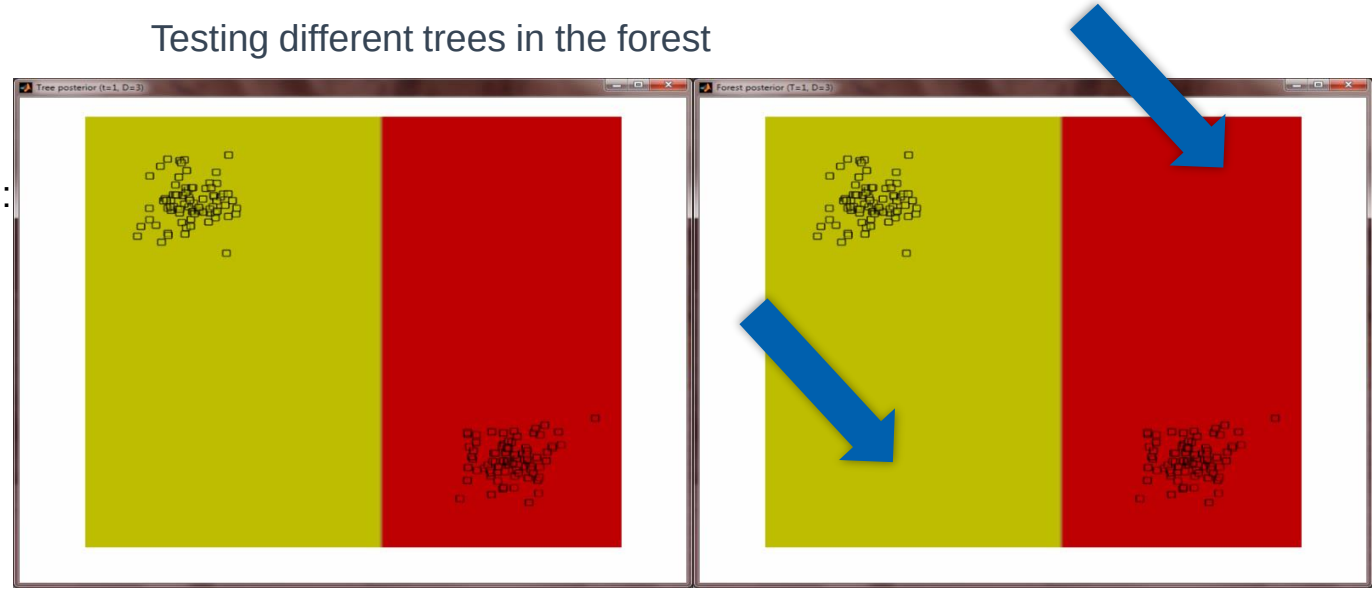
Classification forest: effect of the weak learner model



Training different trees in the forest



Testing different trees in the forest

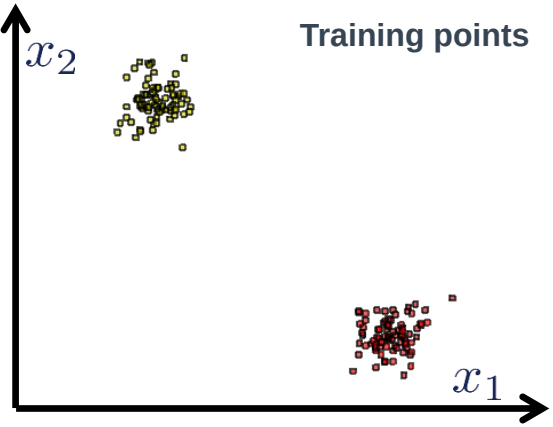


Three concepts to keep in mind:

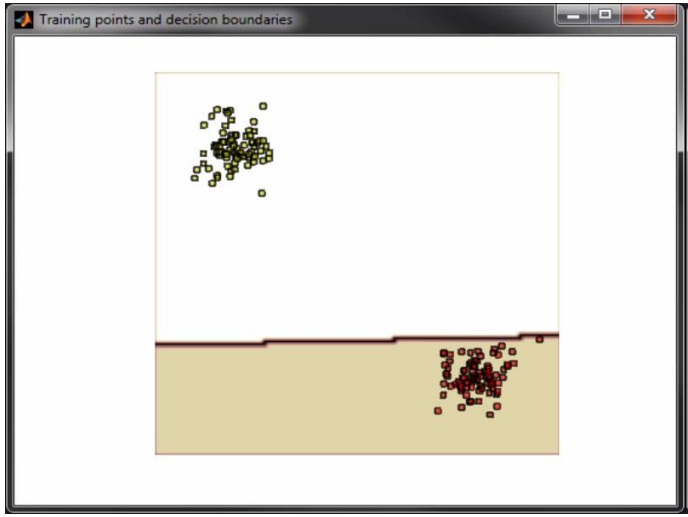
- "Accuracy of prediction"
- "Quality of confidence"
- "Generalization"

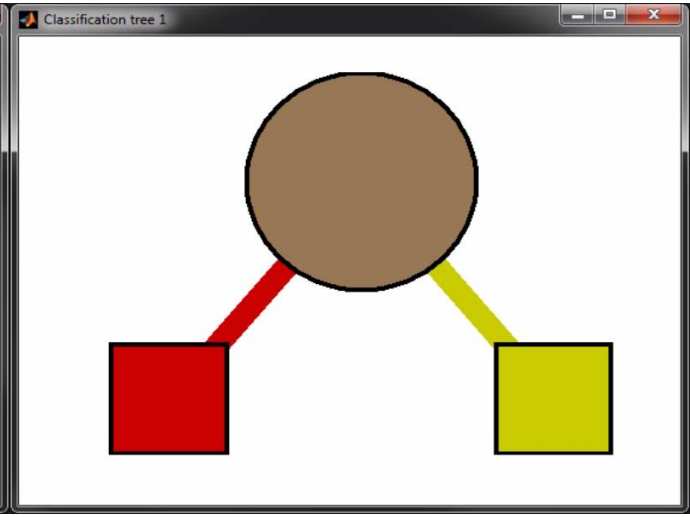
Parameters: T=200, D=2, weak learner = aligned, leaf model = probabilistical

Classification forest: effect of the weak learner model

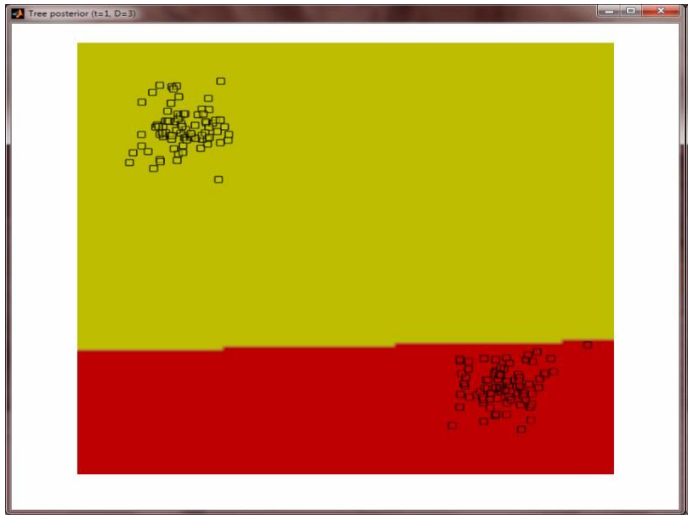


Training different trees in the forest



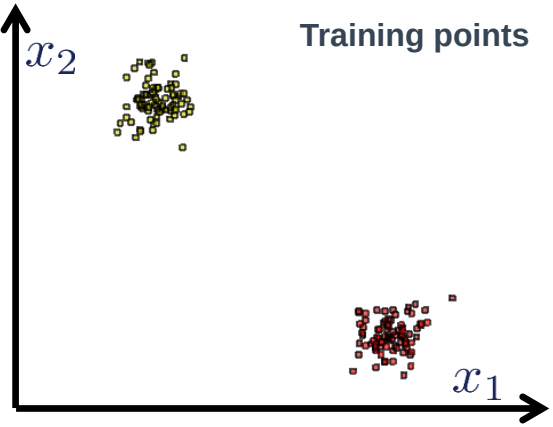


Testing different trees in the forest

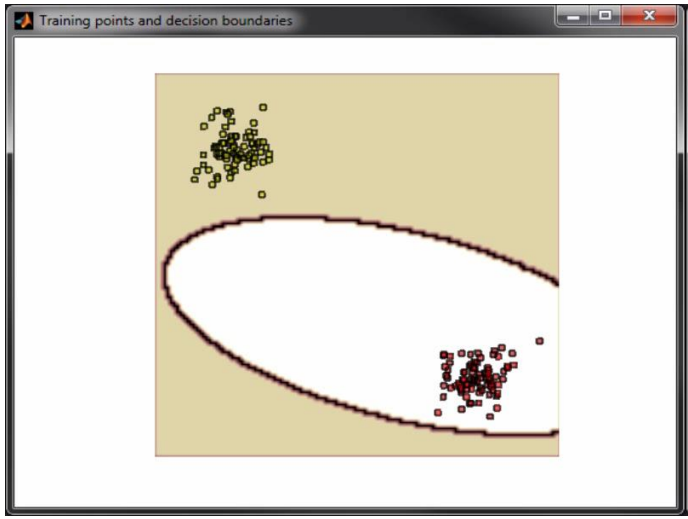


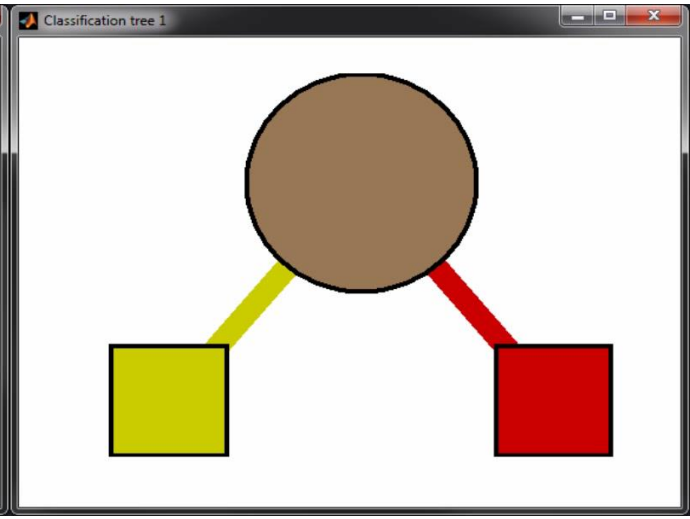
Parameters: T=200, D=2, weak learner = linear, leaf model = probabilistic

Classification forest: effect of the weak learner model

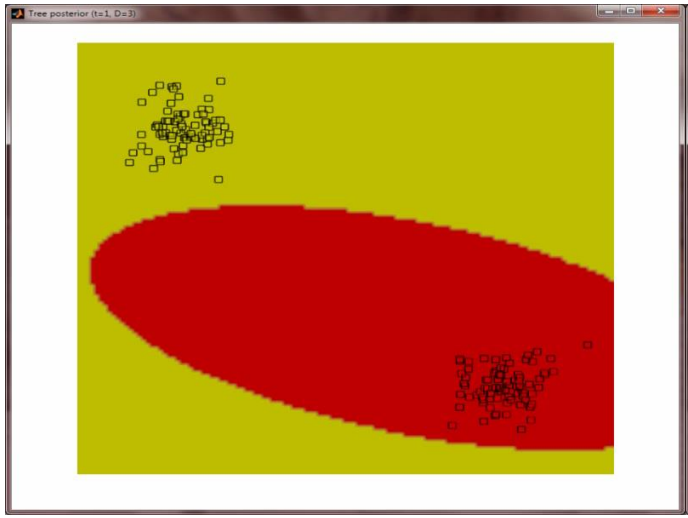


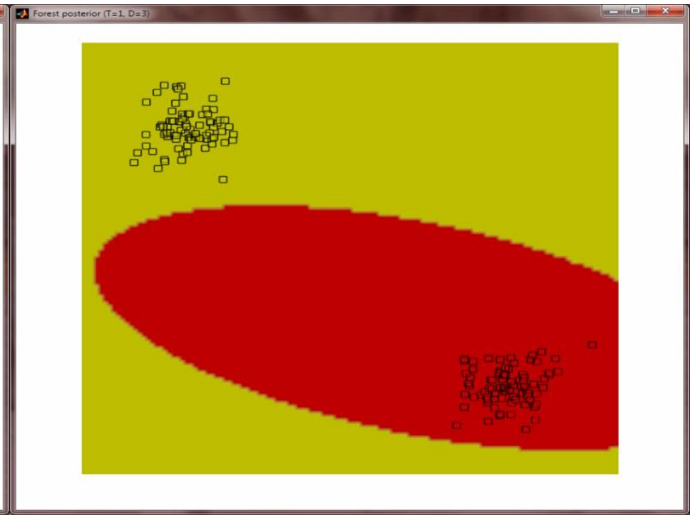
Training different trees in the forest





Testing different trees in the forest

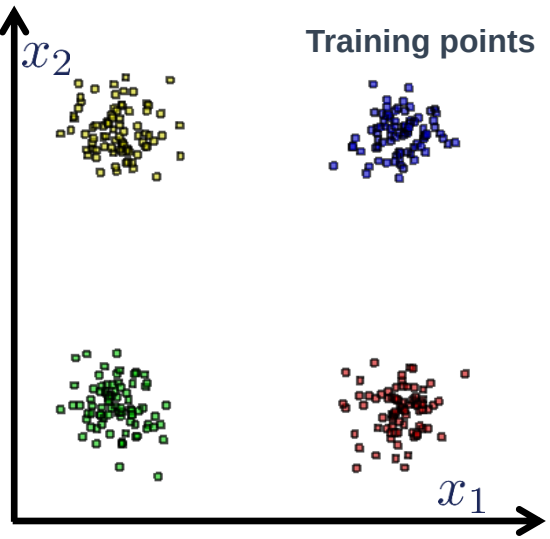


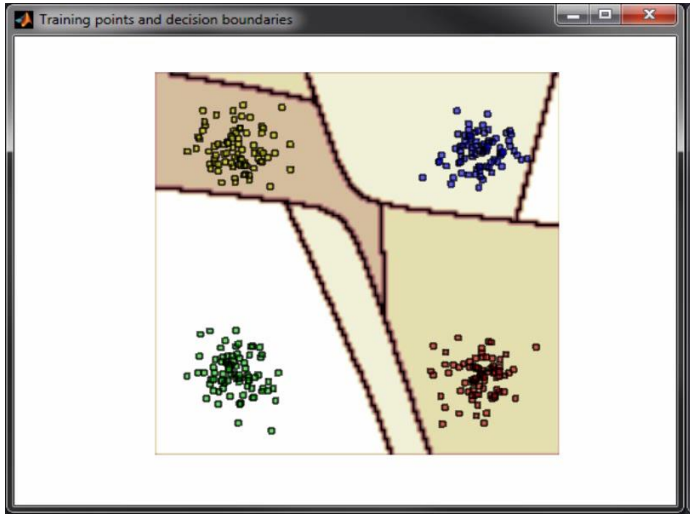


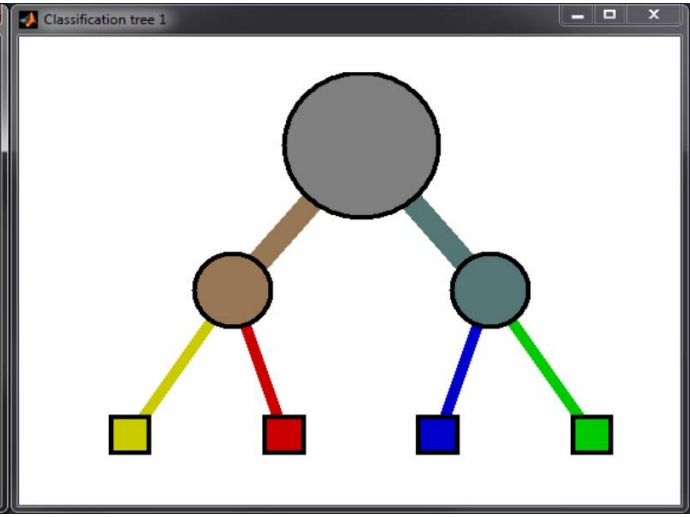
Parameters: T=200, D=2, weak learner = conic, leaf model = probabilistic

Classification forest: with >2 classes

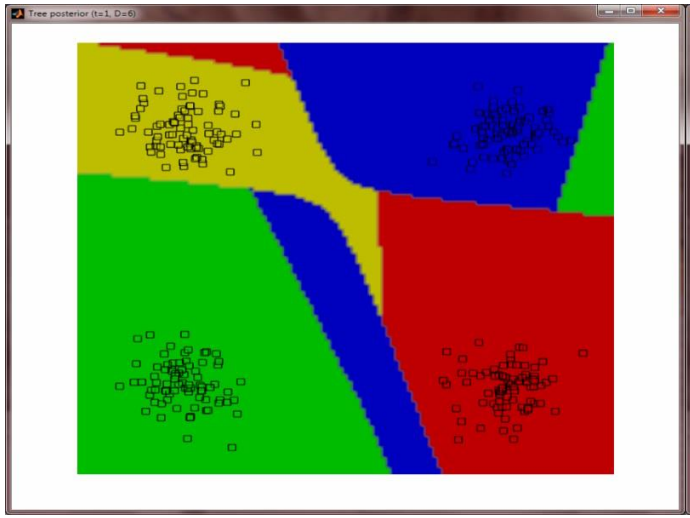
Training different trees in the forest

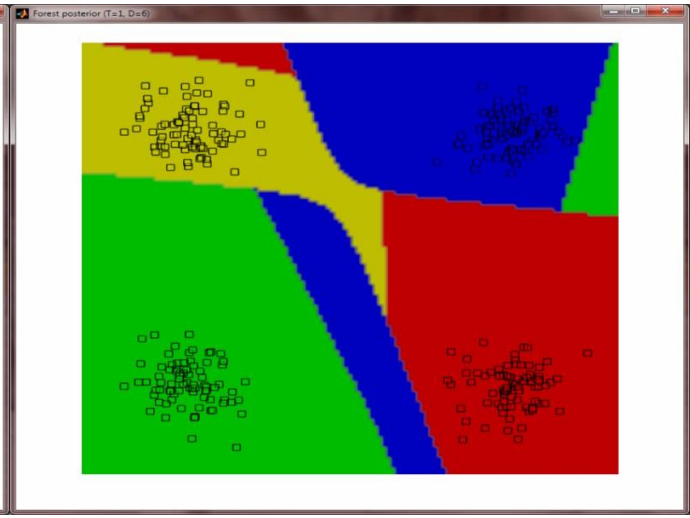






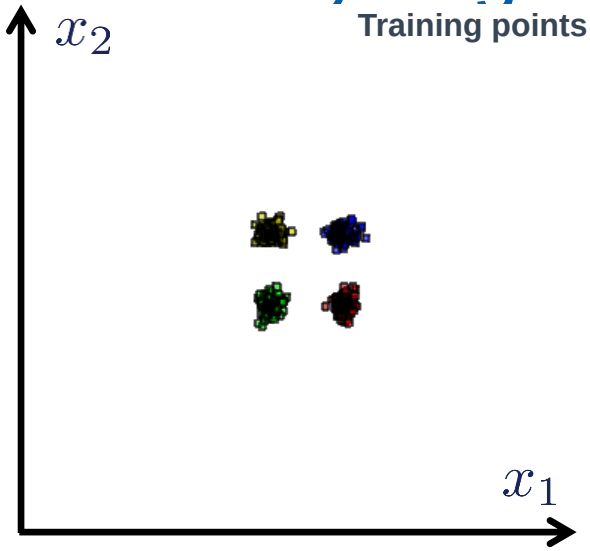
Testing different trees in the forest



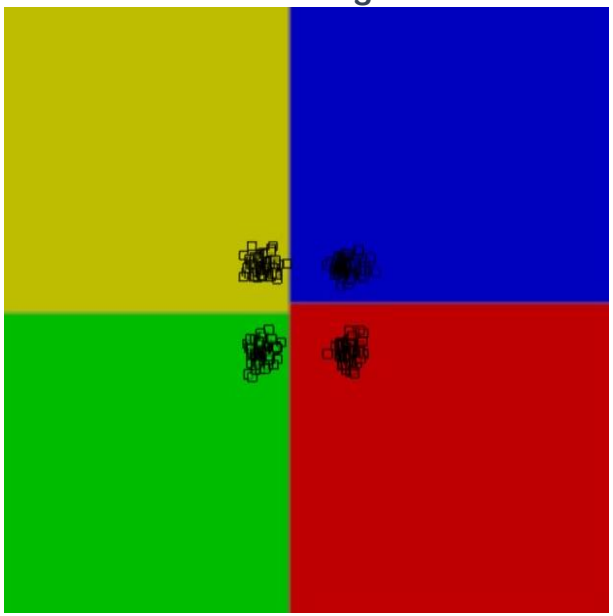


Parameters: T=200, D=3, weak learner = conic, leaf model = probabilistic

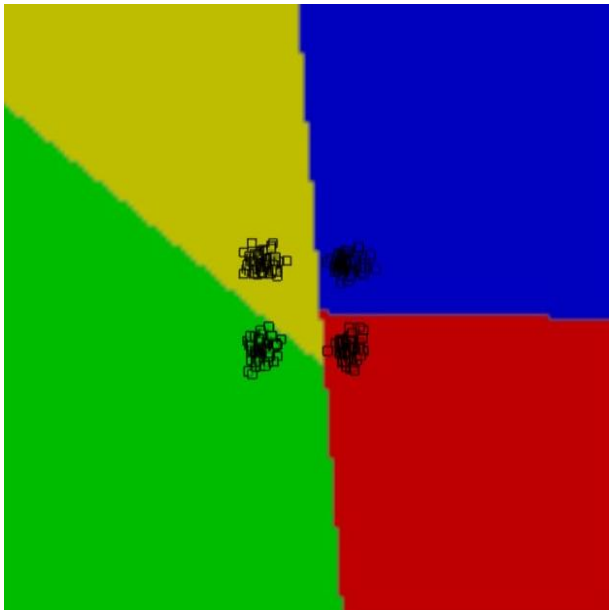
Classification forest: analysing generalization



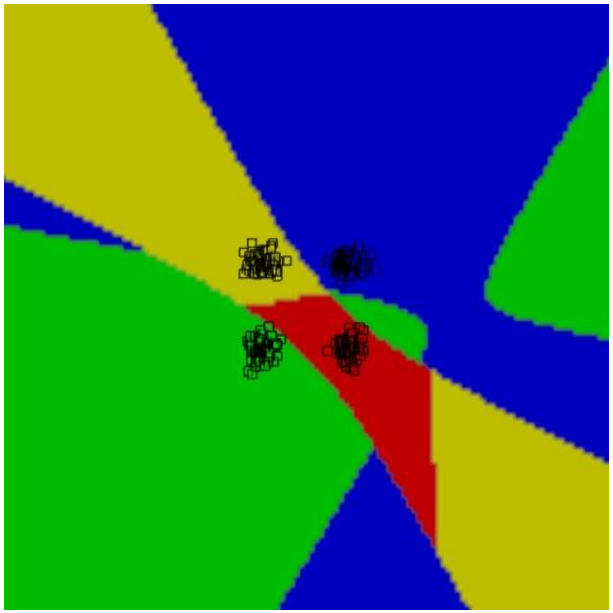
Weak learner: axis aligned



Weak learner: oriented line

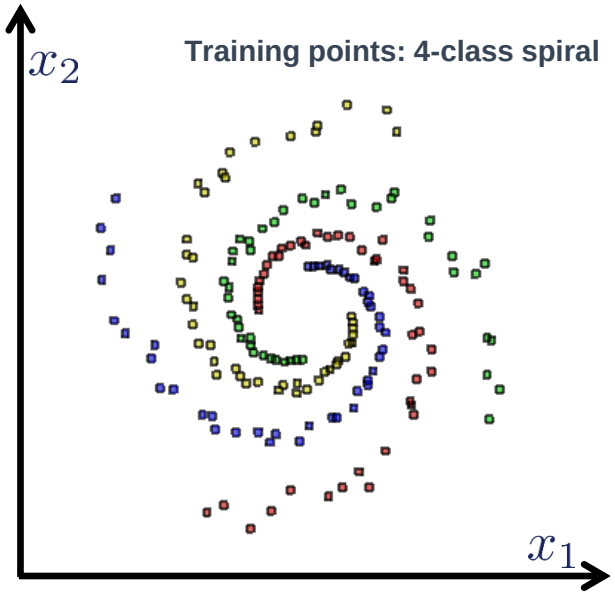


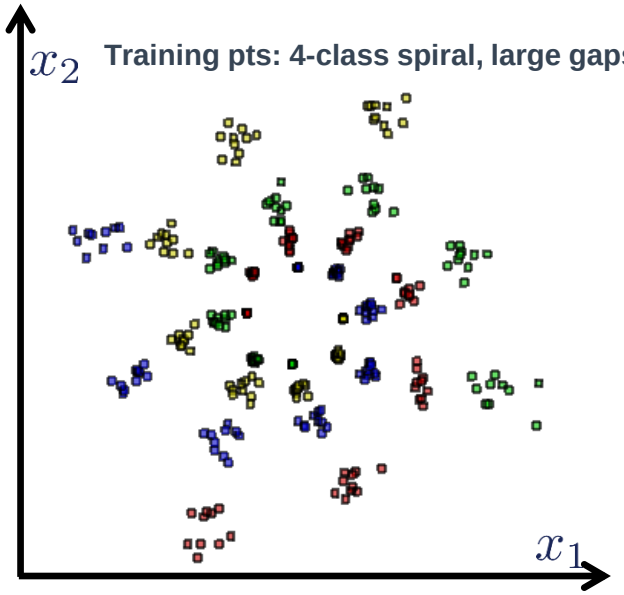
Weak learner: conic section

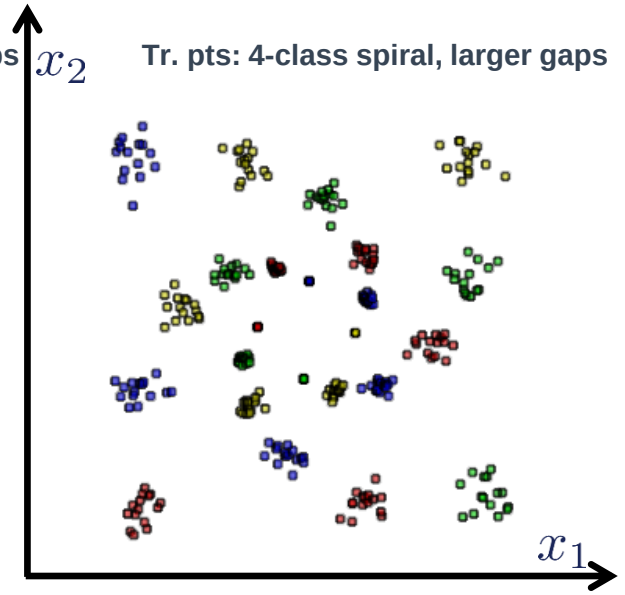


Parameters: T=200, D=3, leaf model = probabilistic

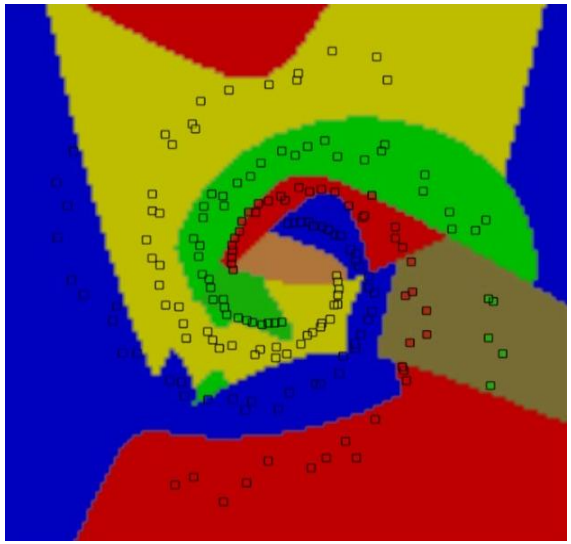
Classification forest: analysing generalization

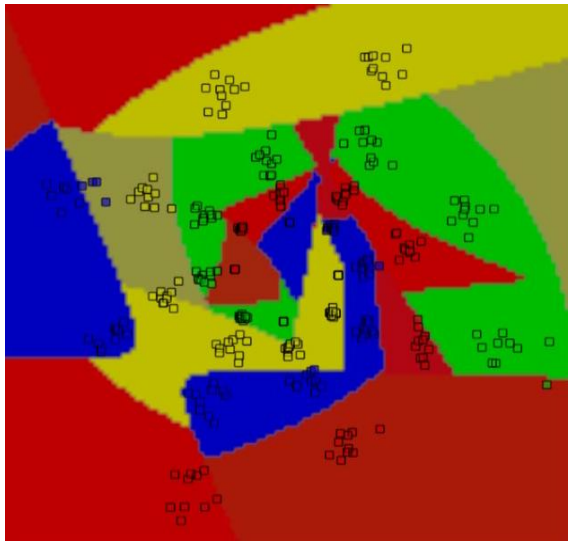


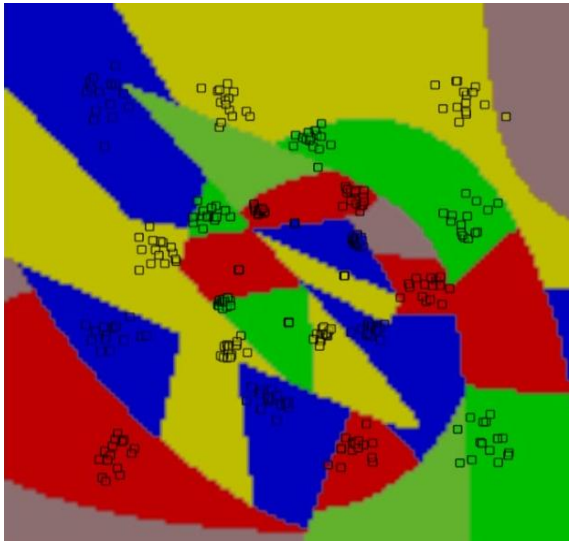




Testing posteriors

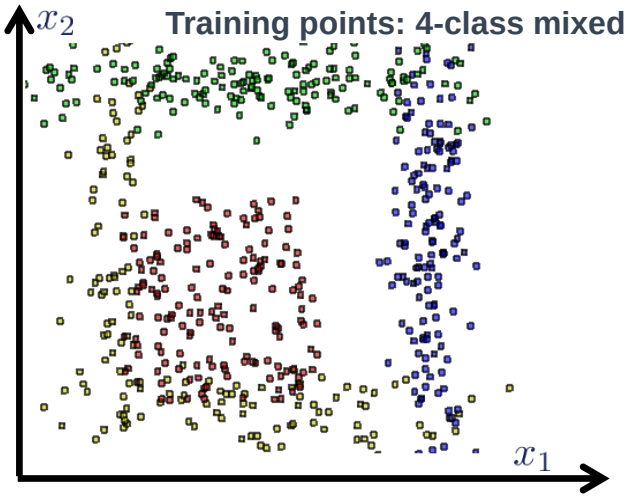


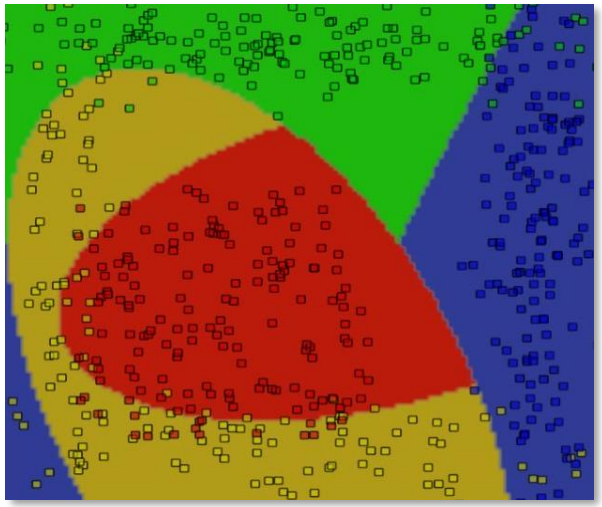




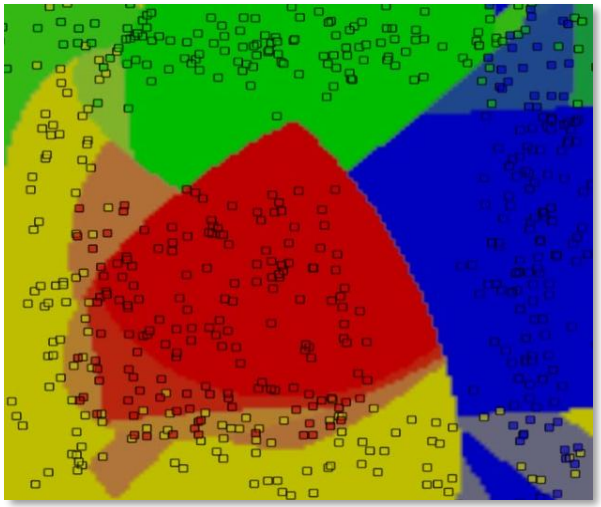
Parameters: T=200, D=13, w. l. = conic, predictor = prob.

Classification forest: effect of tree depth

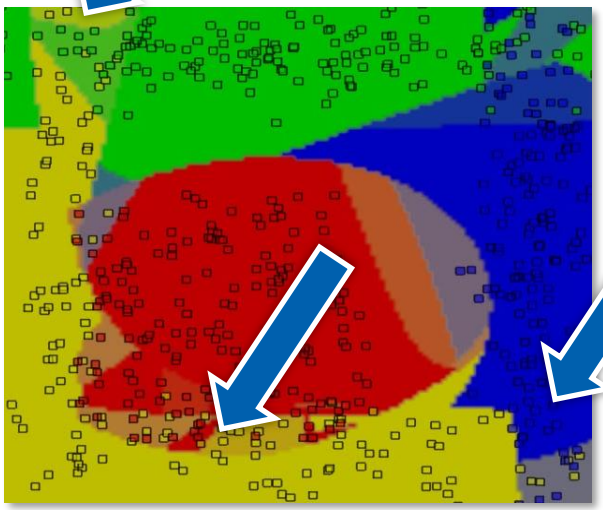




T=200, D=3, w. l. = conic



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



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

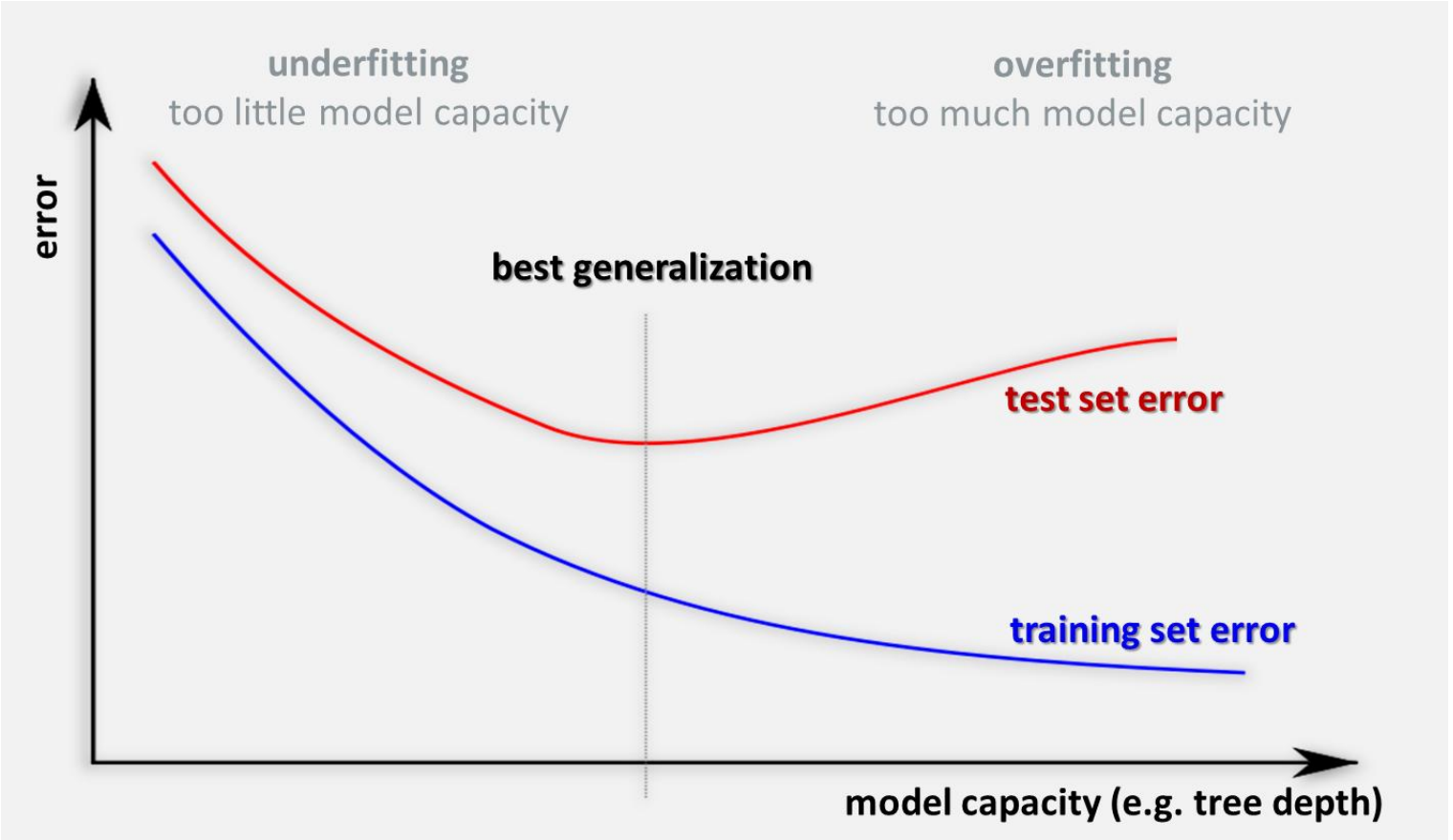
max tree depth, D

underfitting

overfitting

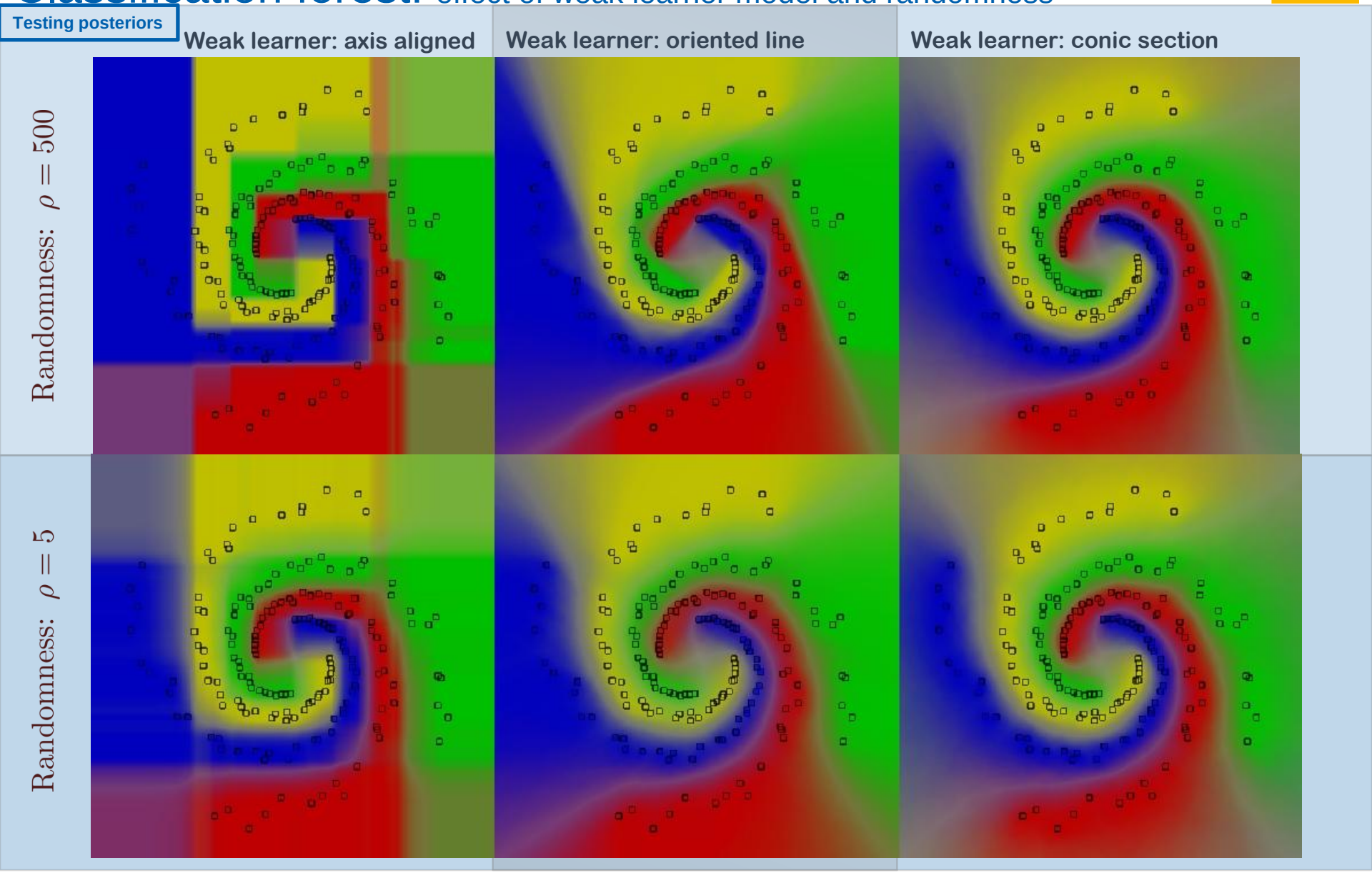
Predictor model = prob.

Background: overfitting and underfitting



Classification forest: effect of weak learner model and randomness

Testing posteriors



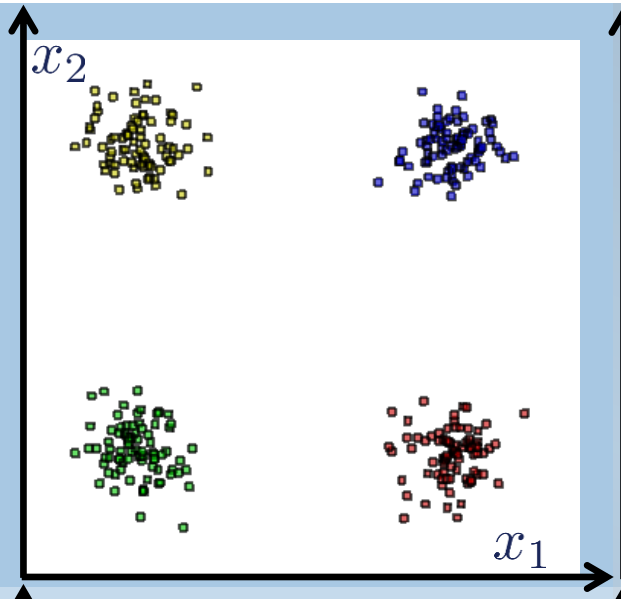
D=13

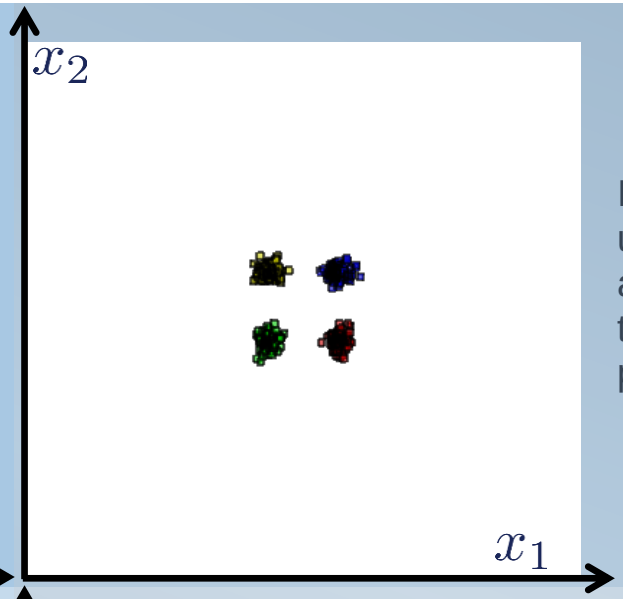
Parameters: T=400 predictor model = prob.

Classification forest: comparison with SVM

UNIVERSITÄT BONN

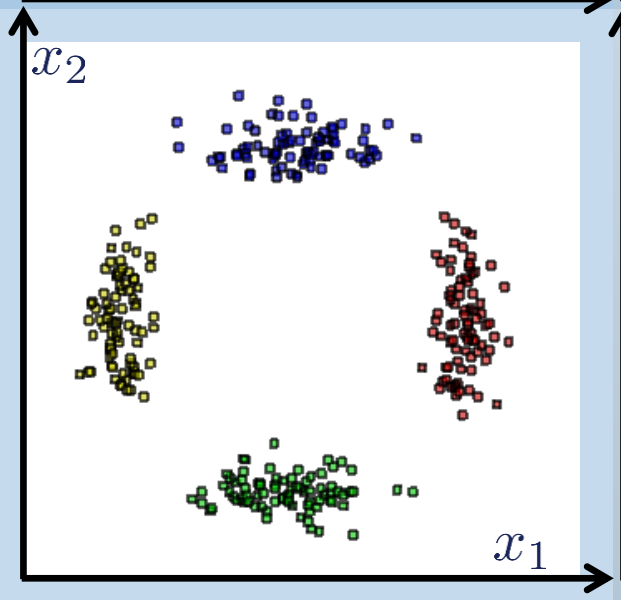
Max-margin like behaviour for multi-class problem

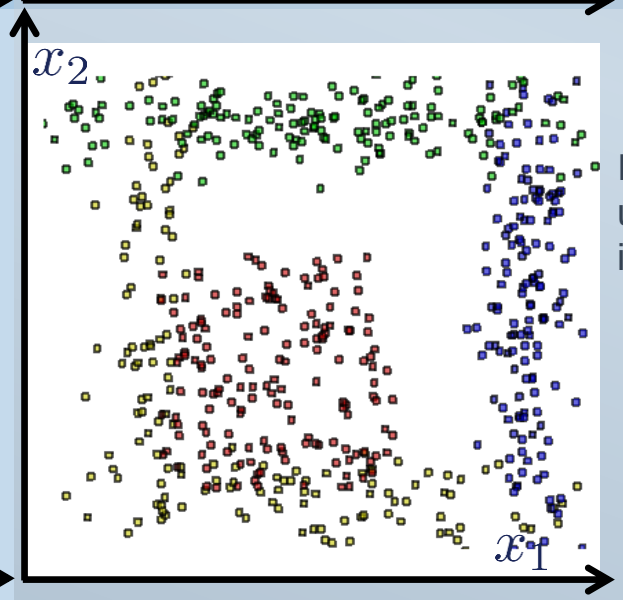




Increased uncertainty away from training points.

Max-margin like behaviour for multi-class problem



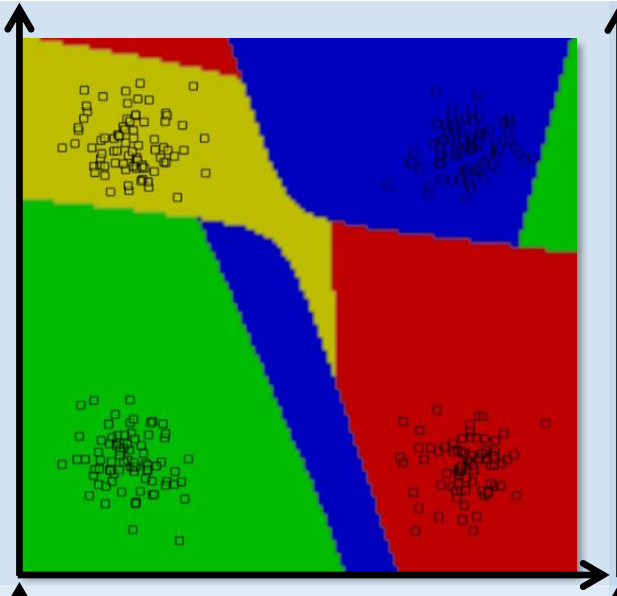


Increased uncertainty in mixed regions

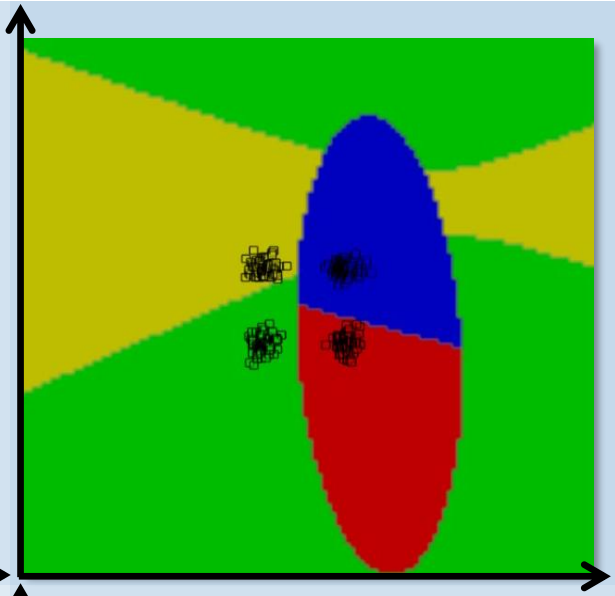
Classification forest: comparison with SVM

UNIVERSITÄT BONN

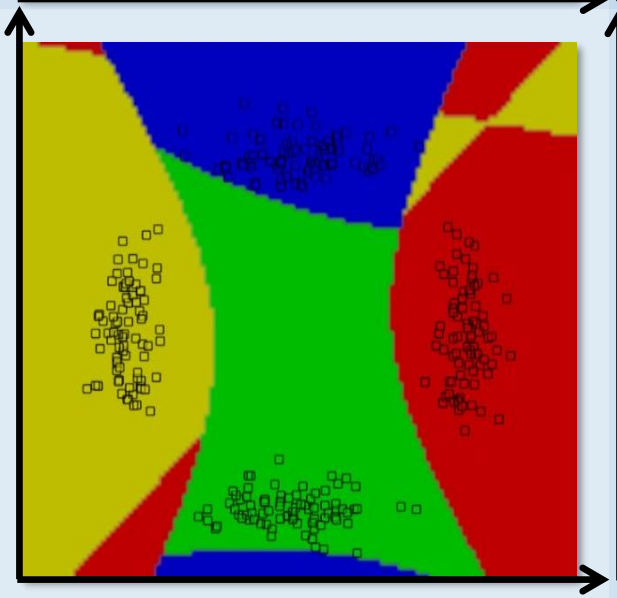
Max-margin like behaviour for multi-class problem



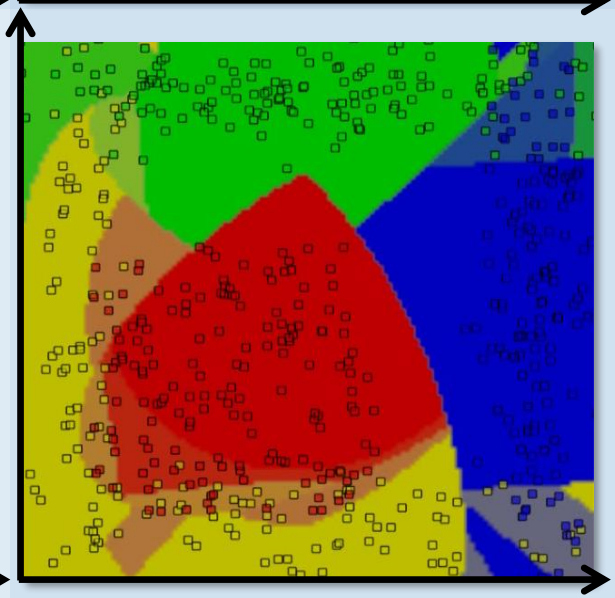
Increased uncertainty away from training points.



Max-margin like behaviour for multi-class problem



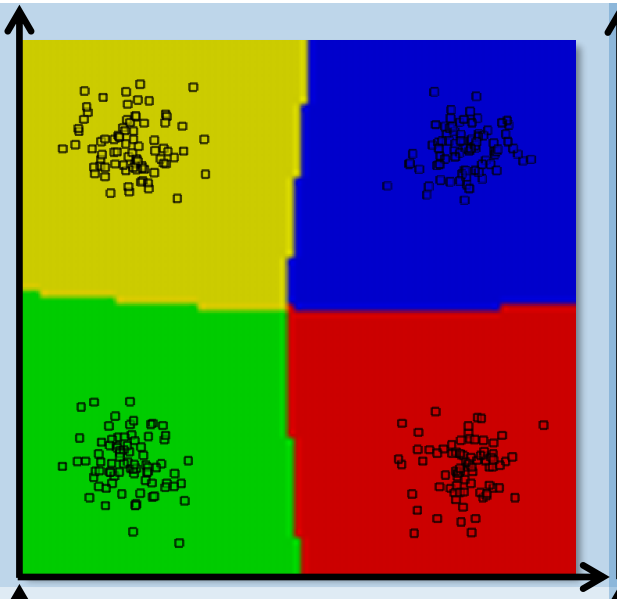
Increased uncertainty in mixed regions

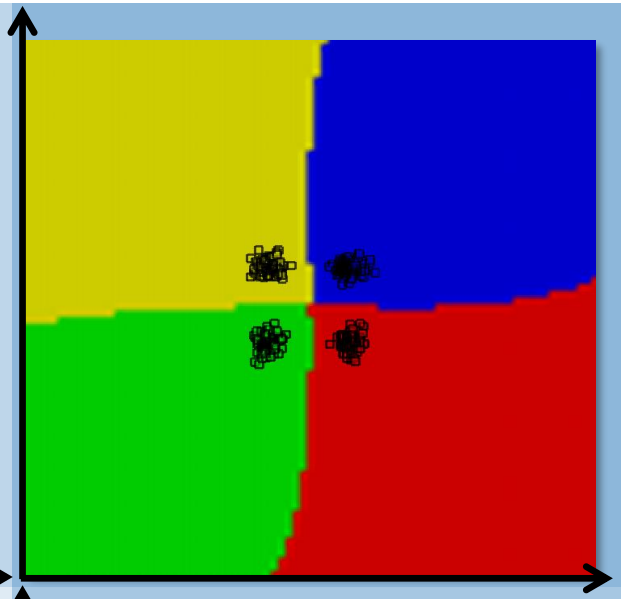


Parameters: T=200, D=6, weak learner = conic, leaf model = probabilistic

Classification forest: comparison with SVM

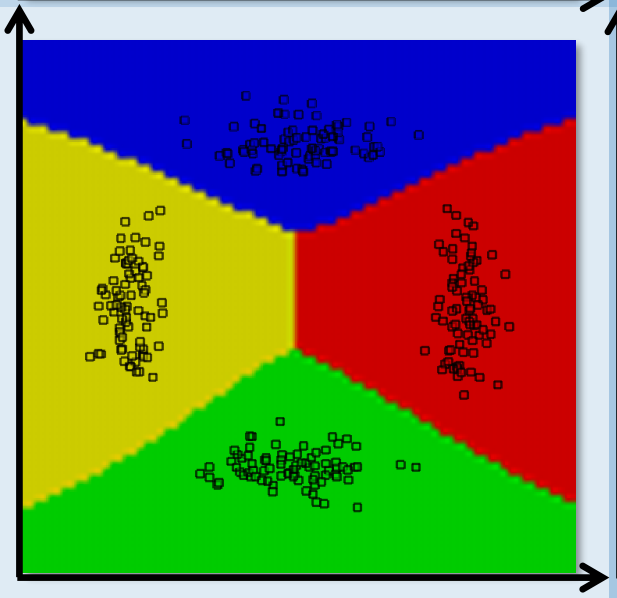
SVM produces nice separation but no confidence information

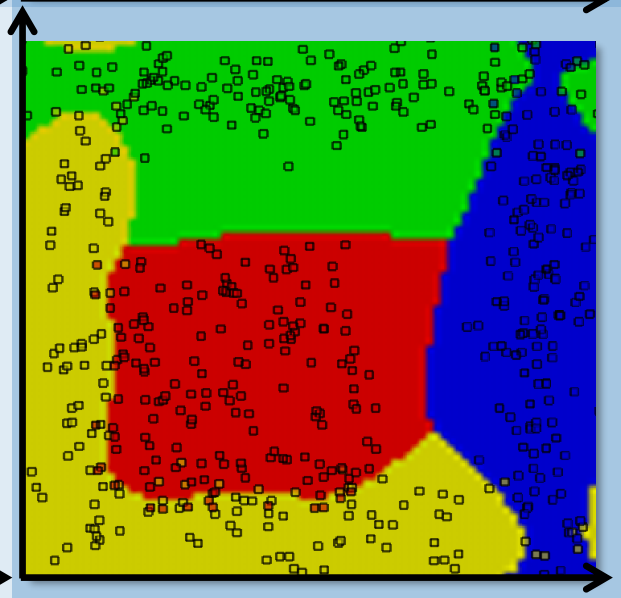




Same high confidence away from training data

Lack of symmetry



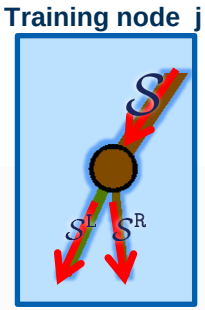


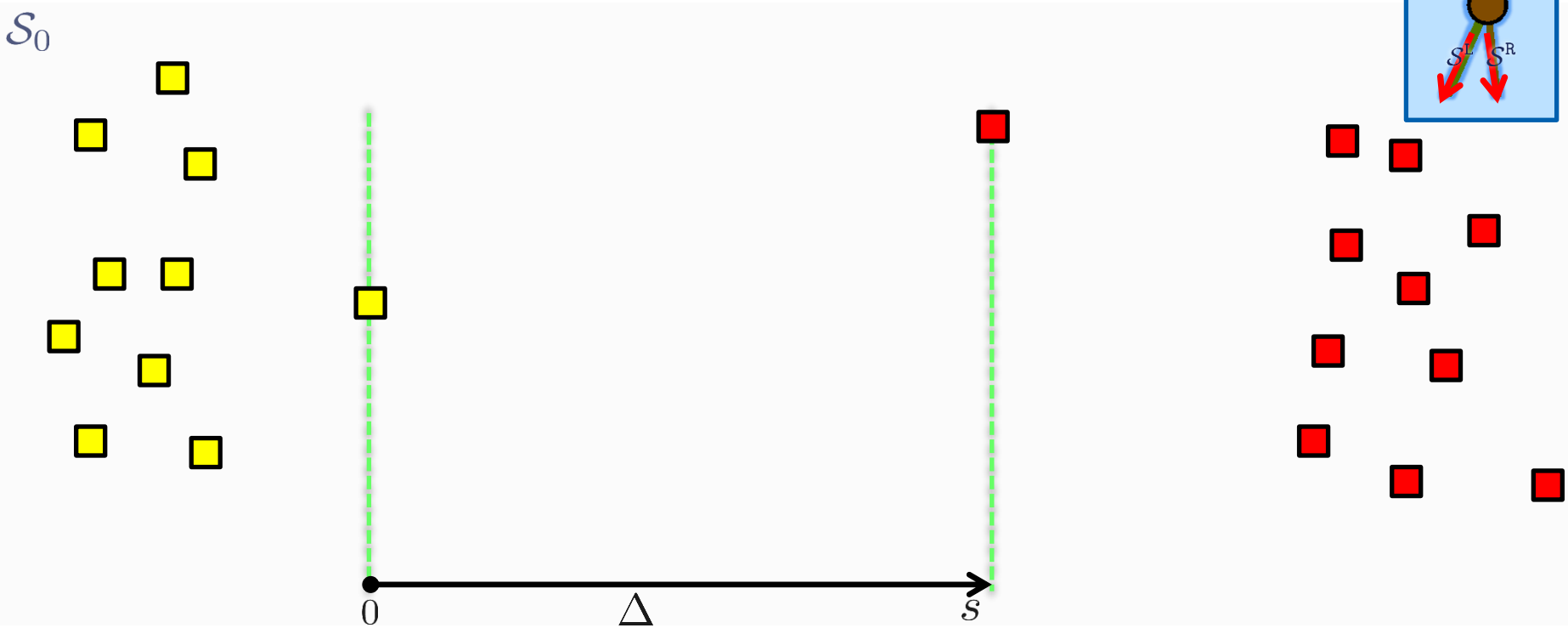
Note overfitting
+
overly confident

[SVM code in http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html]

Classification forest: max-margin properties

Randomness type = Randomized node optimization





Node parameters

$$\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\tau}) \in \mathcal{T}$$

$\boldsymbol{\phi}$ = *which features,*
$\boldsymbol{\psi}$ = *what geometric model,*
$\boldsymbol{\tau}$ = *thresholds*

Vertical weak learner

$$h(\mathbf{v}, \boldsymbol{\theta}) = (\infty > \boldsymbol{\phi}(\mathbf{v}) \cdot \boldsymbol{\psi} > \tau_2)$$

$$\boldsymbol{\psi} = (1\ 0\ 0)$$

$$\boldsymbol{\phi}(\mathbf{v}) = (x_1\ x_2\ 1)^\top$$

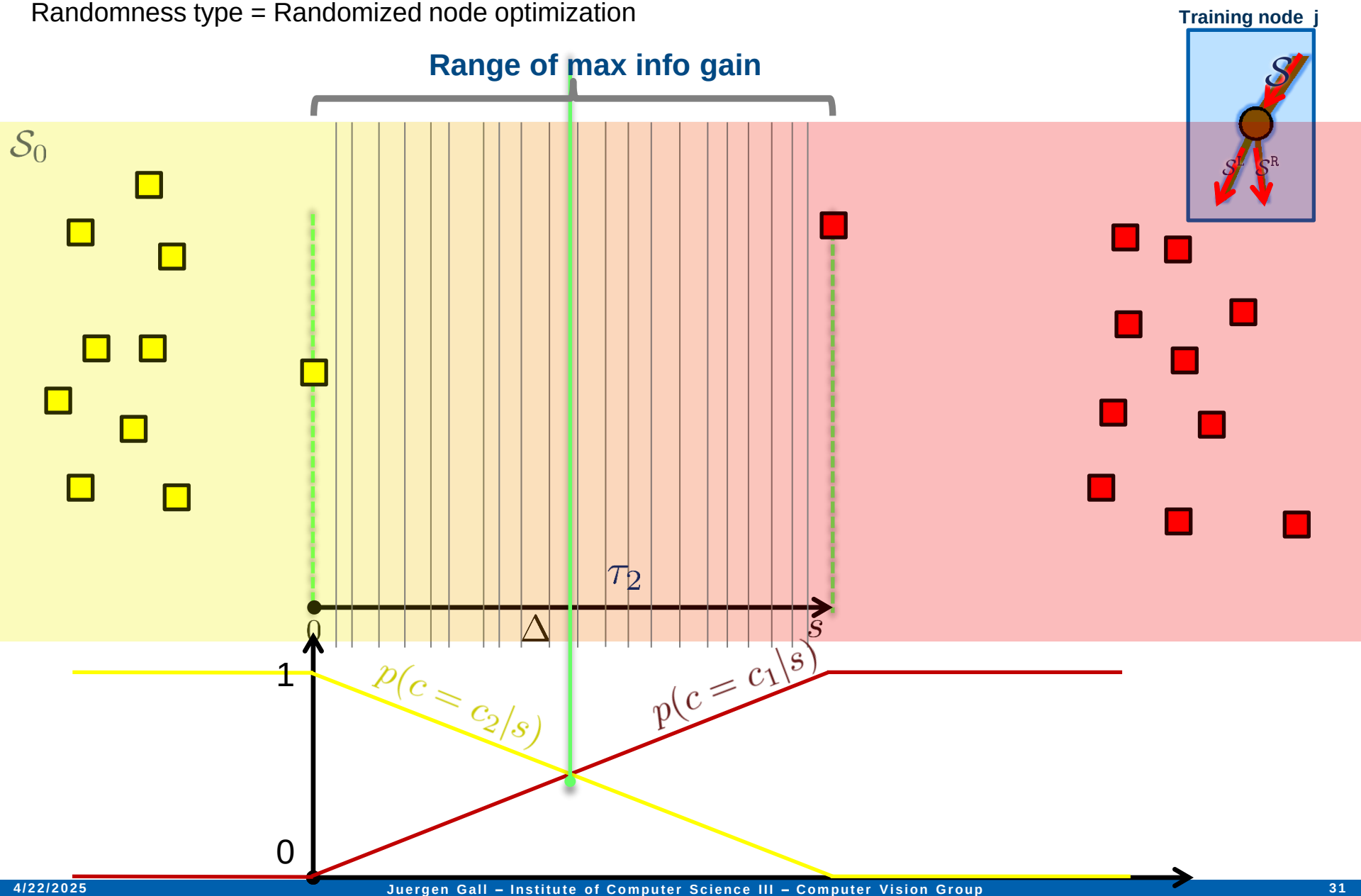
Node training

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

$$\tau_2^* = \arg\max_{\tau_2 \in \mathcal{T}_j} I$$

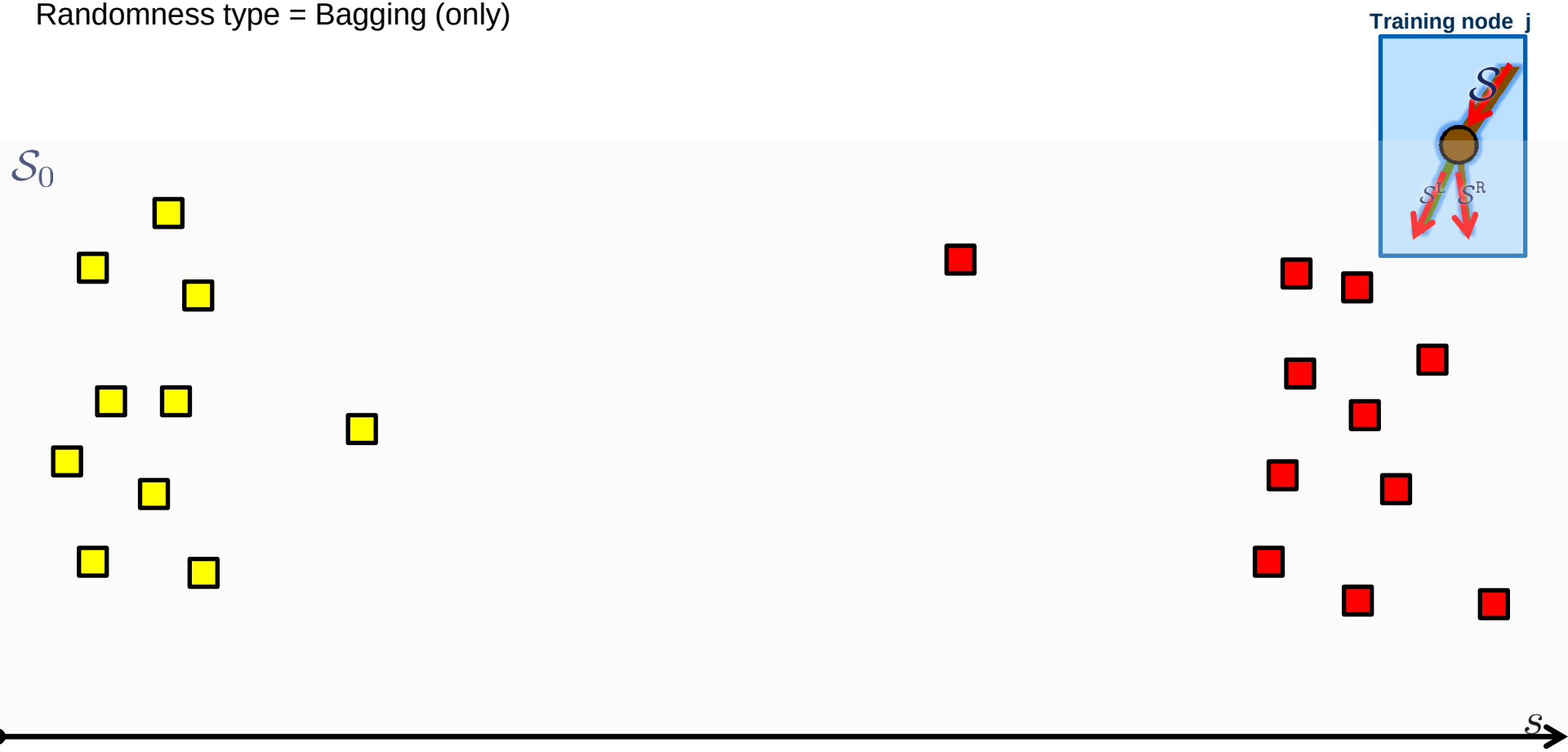
Classification forest: max-margin properties

Randomness type = Randomized node optimization



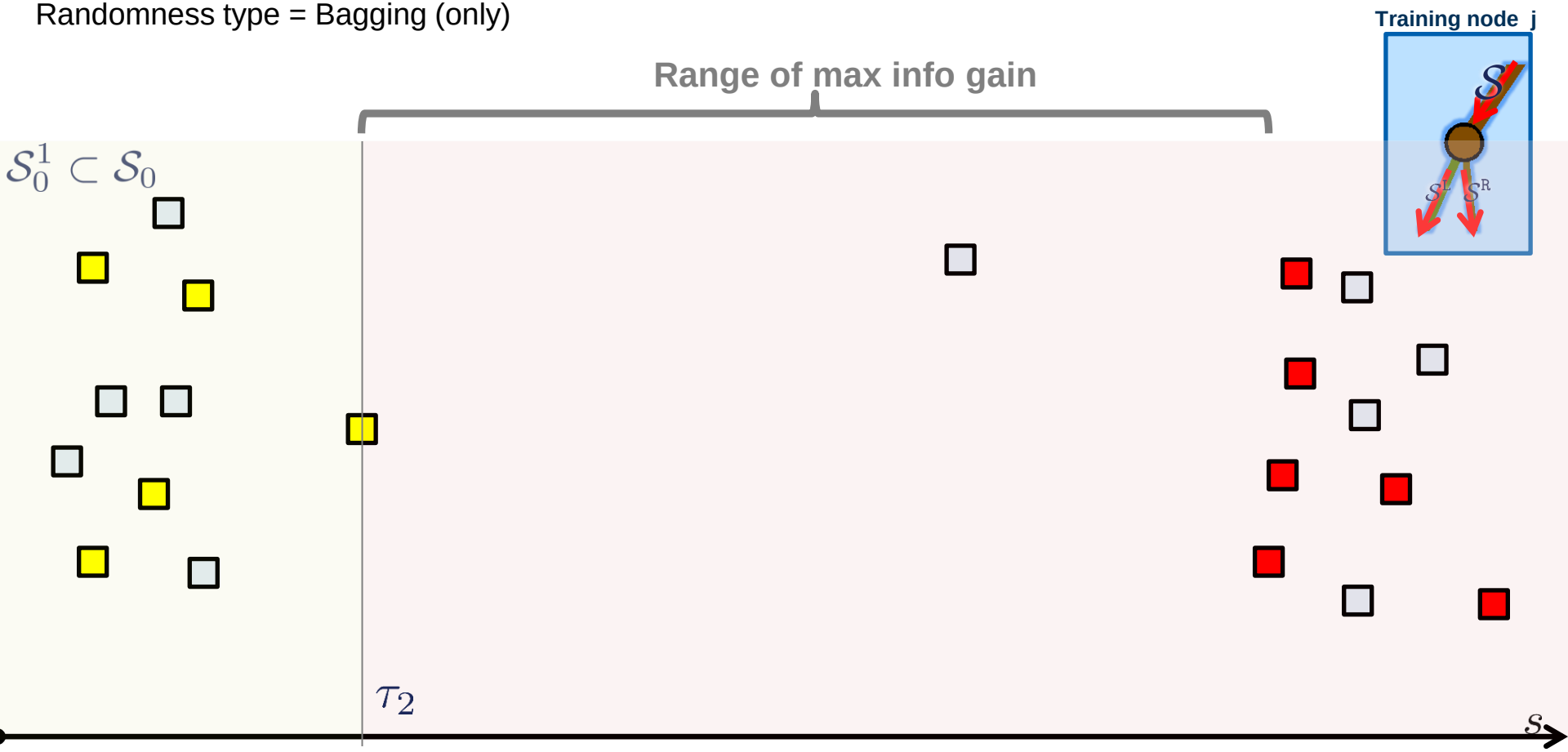
Classification forest: max-margin properties

Randomness type = Bagging (only)



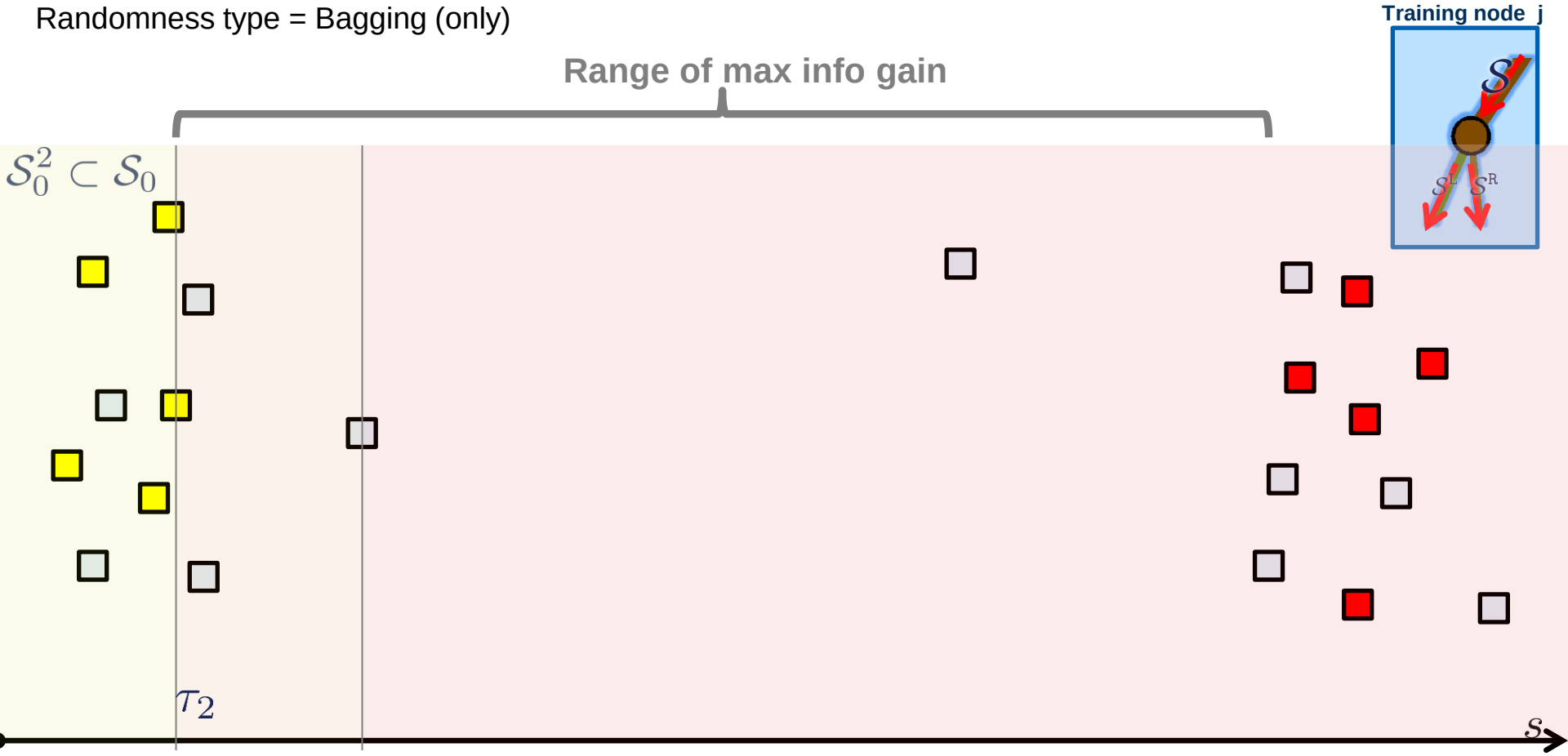
Classification forest: max-margin properties

Randomness type = Bagging (only)



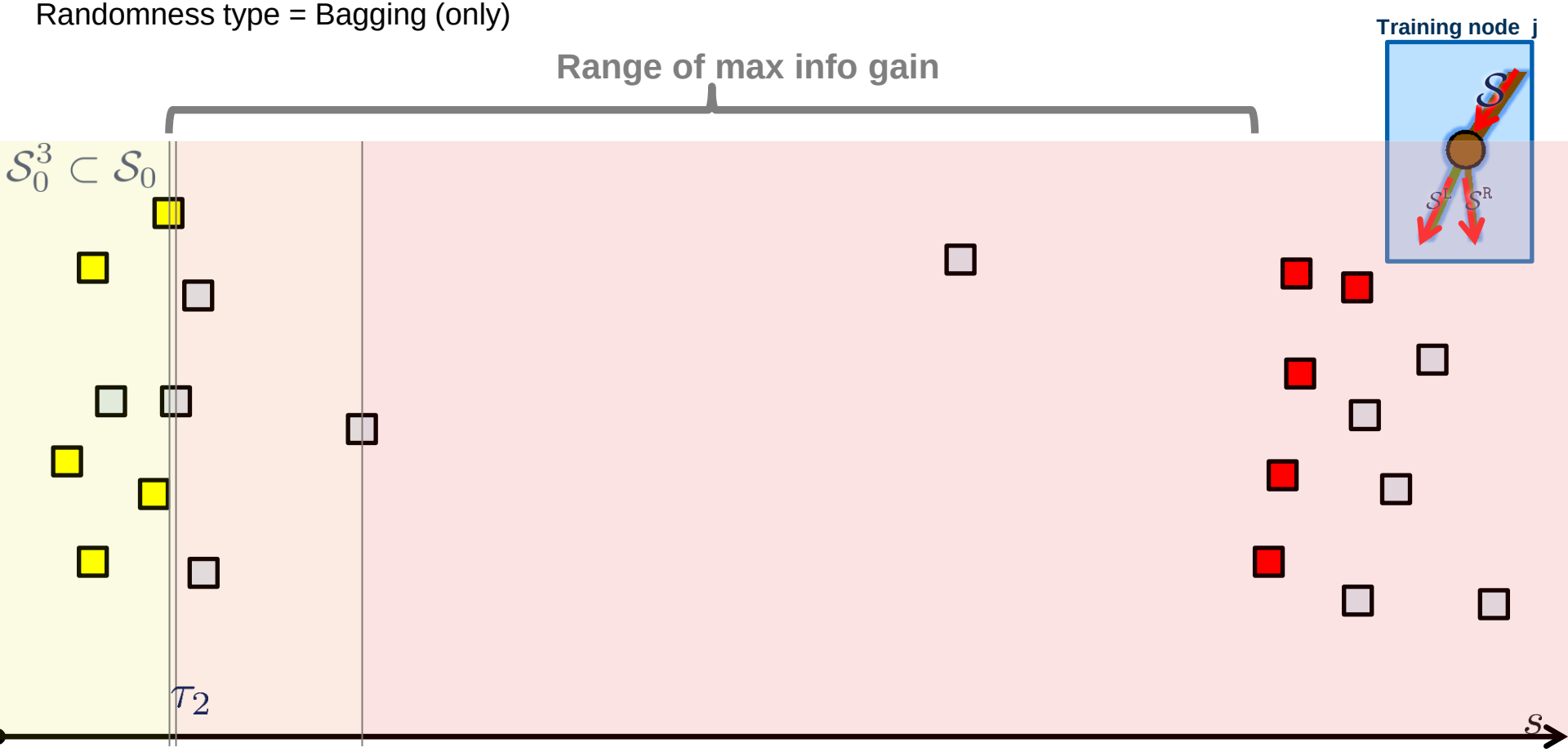
Classification forest: max-margin properties

Randomness type = Bagging (only)



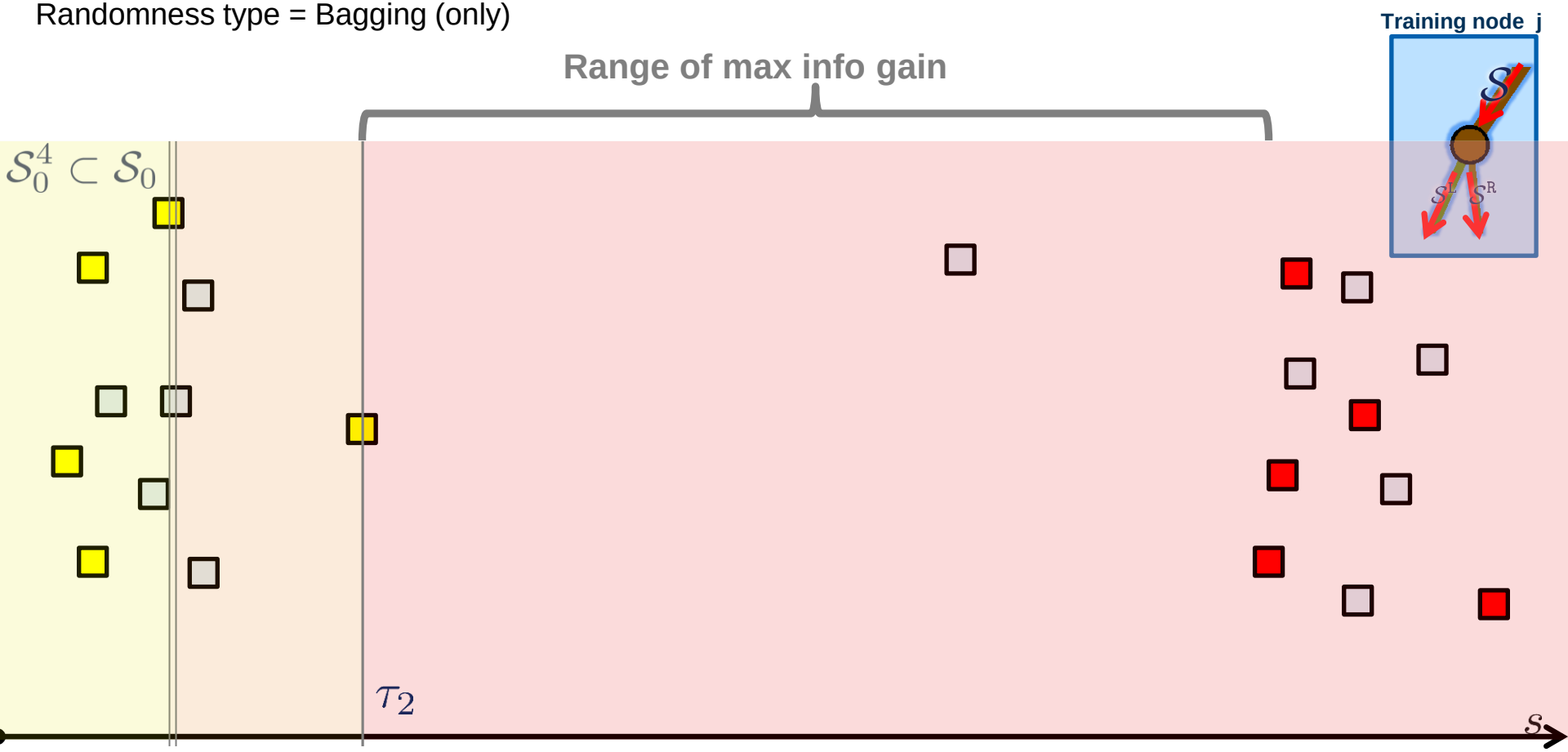
Classification forest: max-margin properties

Randomness type = Bagging (only)



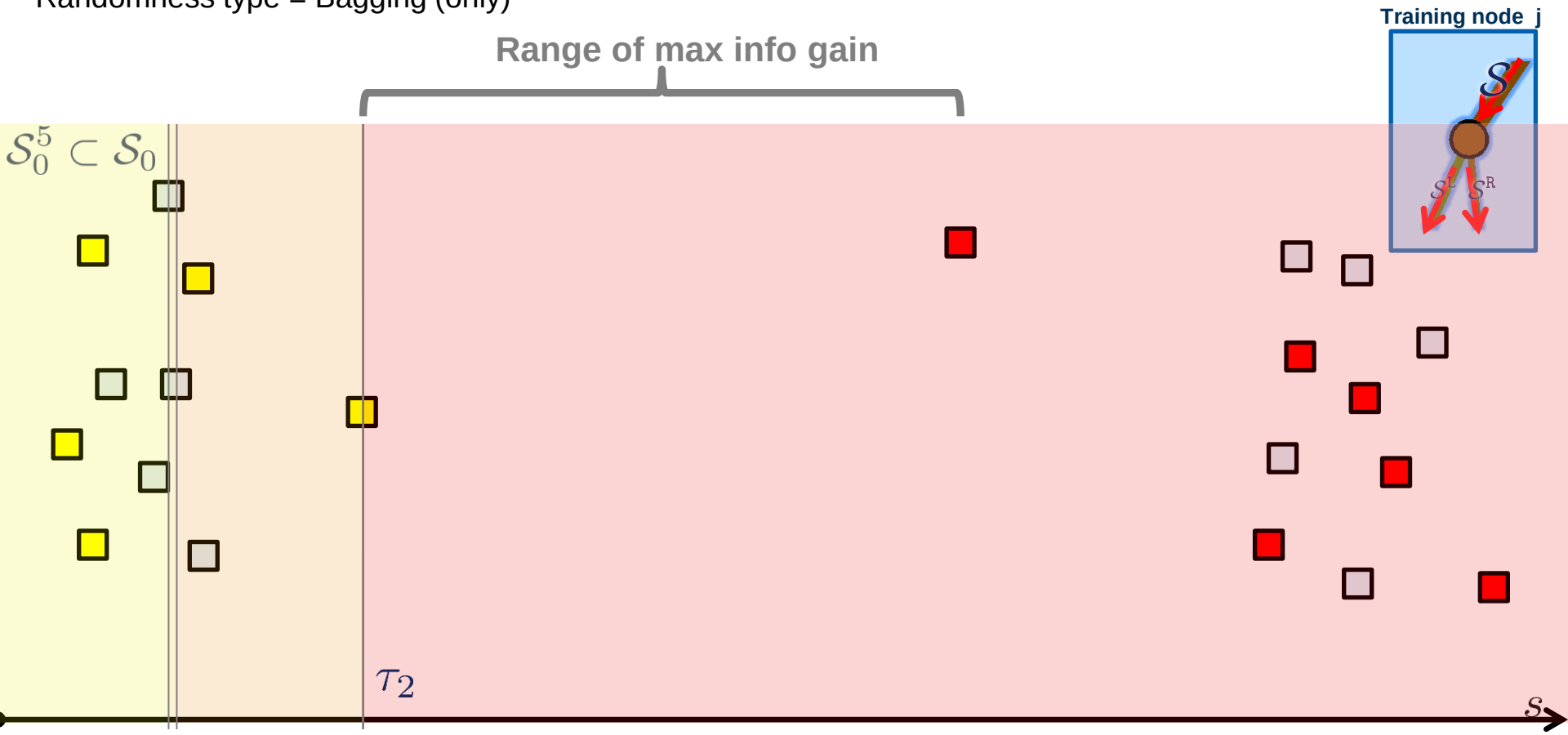
Classification forest: max-margin properties

Randomness type = Bagging (only)



Classification forest: max-margin properties

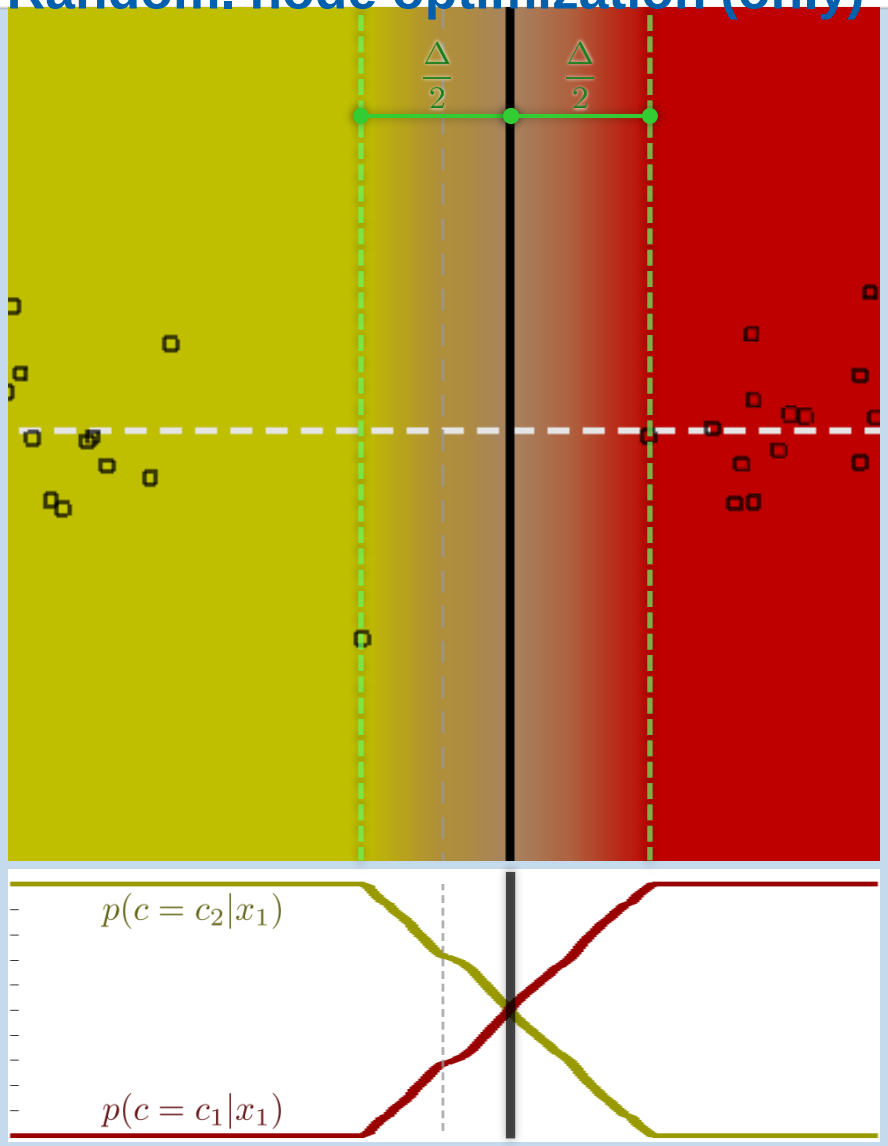
Randomness type = Bagging (only)



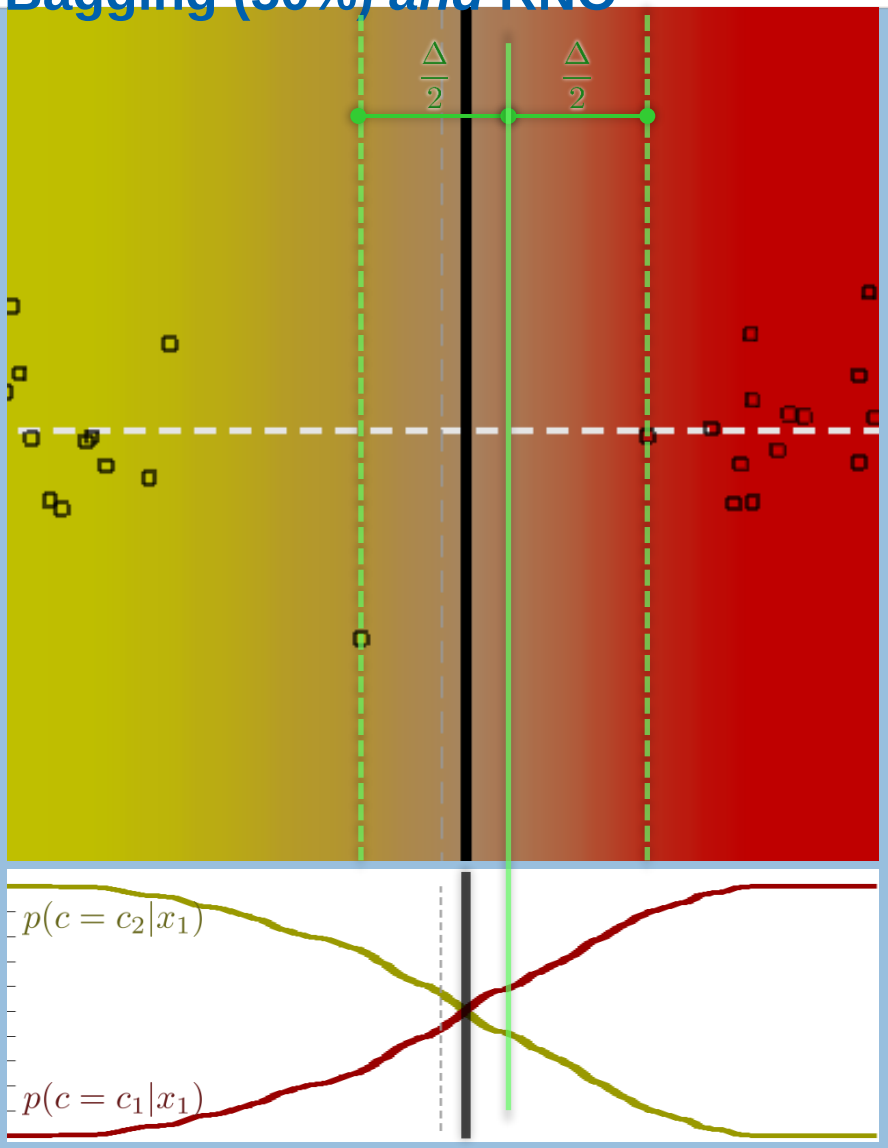
Classification forest: max-margin – comparing with bagging

UNIVERSITÄT BONN

Random. node optimization (only)



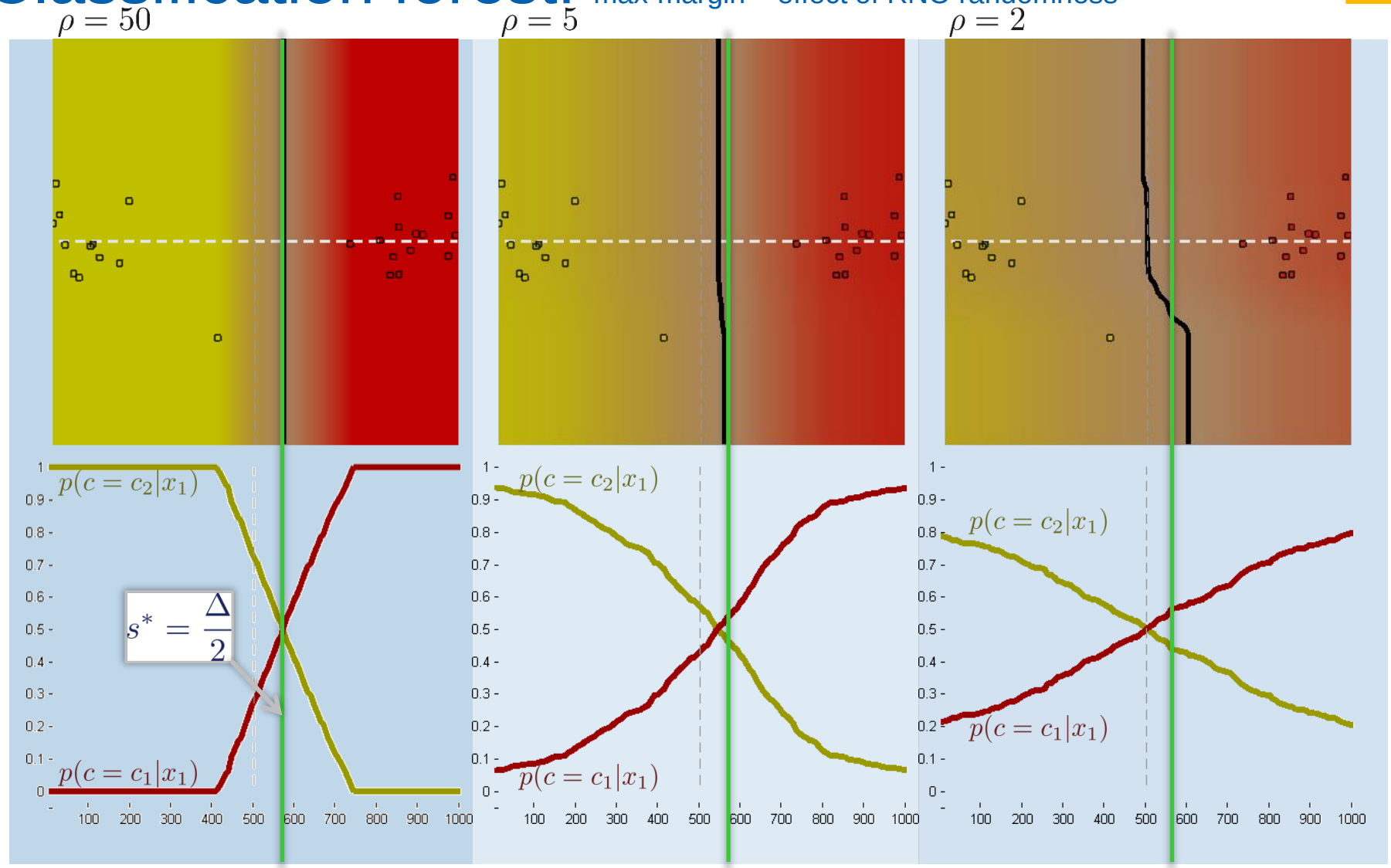
Bagging (50%) *and* RNO



Parameters: $\rho = 500, D = 2, T = 500$, w.l. axis aligned

Classification forest: max-margin – effect of RNO randomness

UNIVERSITÄT BONN



Increased randomness decreases max-margin behaviour.
The separating surface is less affected by individual points.

randomness

Parameters: $D = 2, T = 500$, w.l. axis aligned

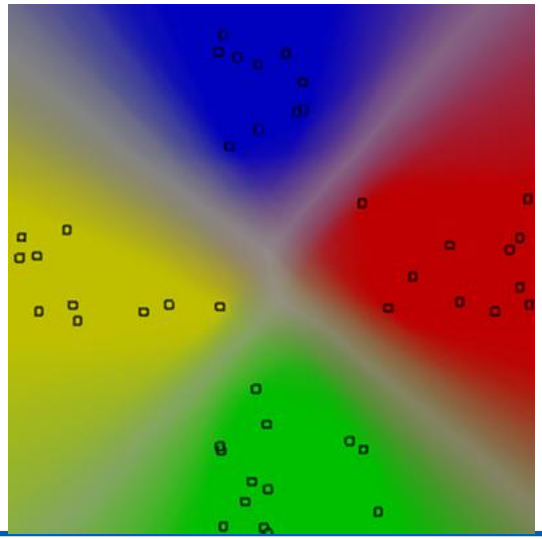
Classification forest: max-margin – for multiple classes

Training points

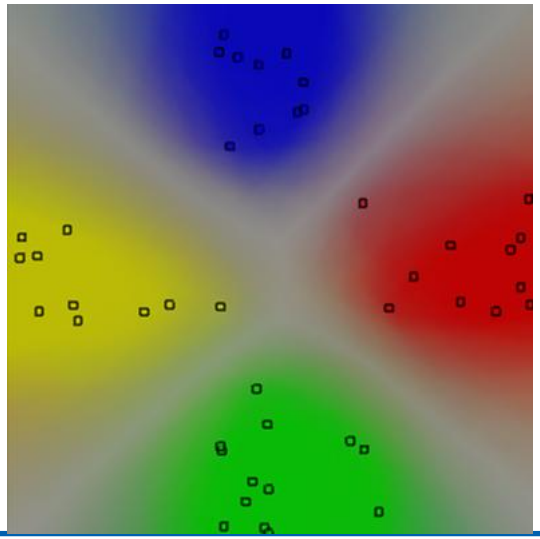
x_2

x_1

weak learner: oriented line



weak learner: conic section



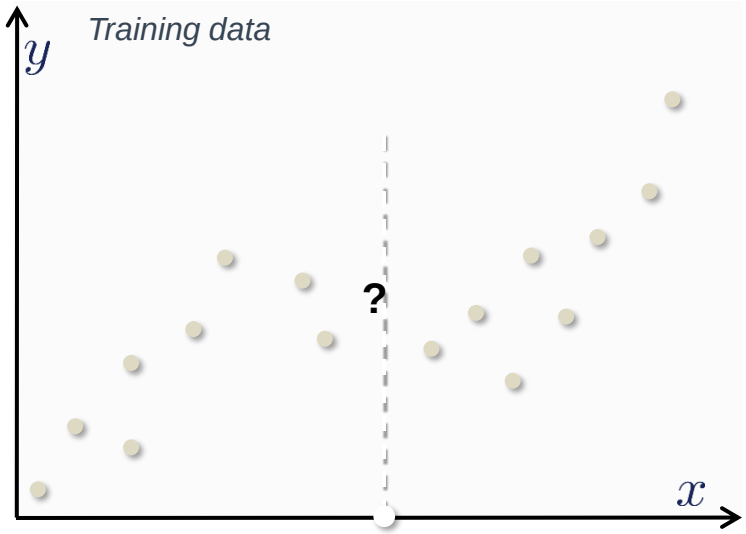
Parameters: $\rho = 50, D = 3, T = 500,$

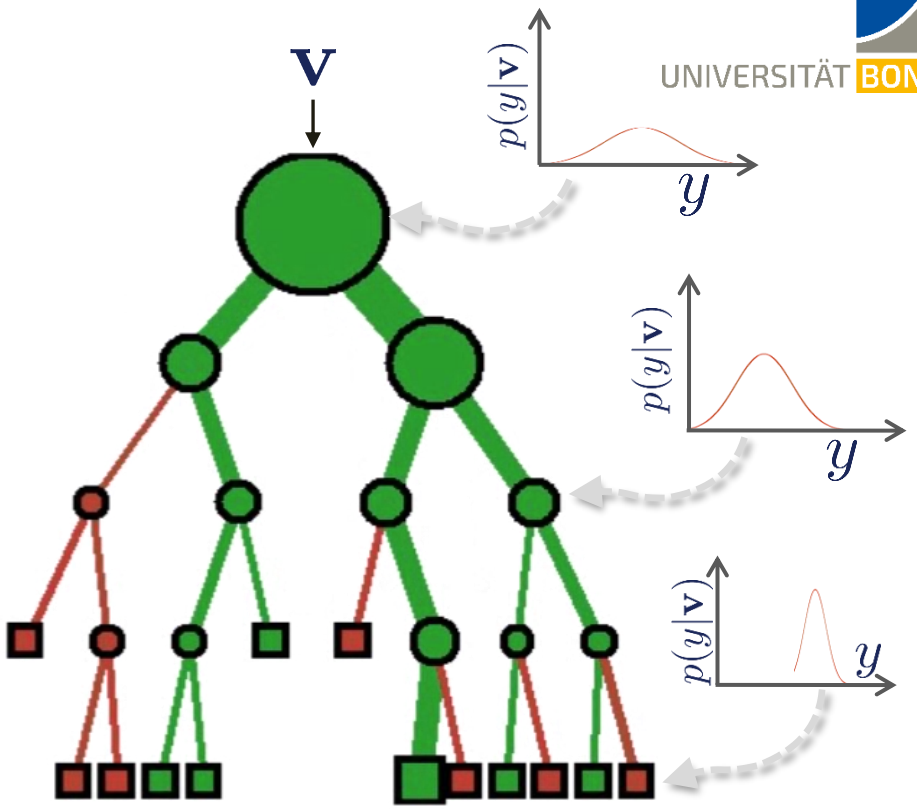
Overview

UNIVERSITÄT BONN

- Classification forests
- Regression forests
- Other variants

Regression forest





Input: $\mathbf{v} = (x_1, \cdots, x_d) \in \mathbb{R}^d$

Output: $p(\mathbf{y}|\mathbf{v}) \quad \mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^n$

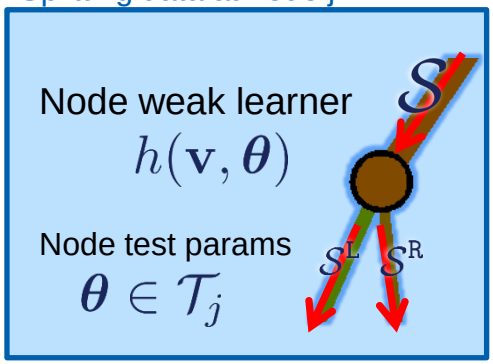
Information gain: $I = H(\mathcal{S}_j) - \sum_{i=\mathrm{L,R}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i)$

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

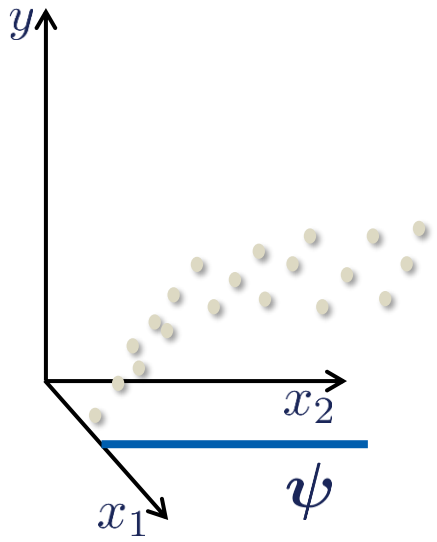
Regression forest: the node weak learner model

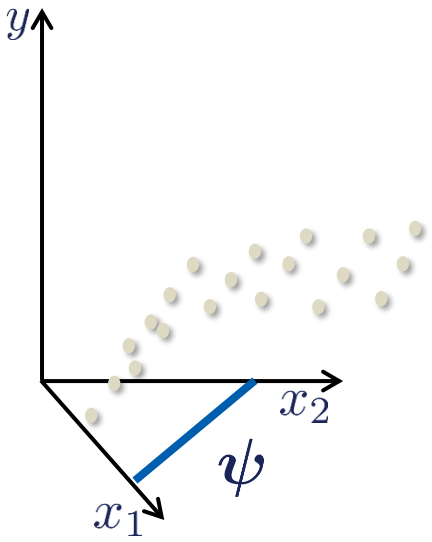
UNIVERSITÄT BONN

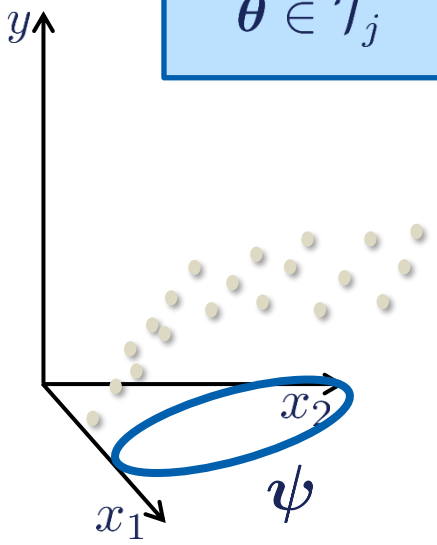
Splitting data at node j



Examples of node weak learners







Weak learner: axis aligned

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

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \; x_2 \; 1)^\top$

With $\boldsymbol{\psi} = (1 \; 0 \; \psi_3)$ or $\boldsymbol{\psi} = (0 \; 1 \; \psi_3)$

Weak learner: oriented line

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

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \; x_2 \; 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^3$ a generic line in homog. coordinates

Weak learner: conic section

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

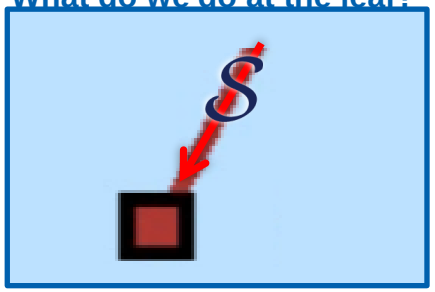
Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \; x_2 \; 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^{3\times 3}$ a matrix representing a conic.

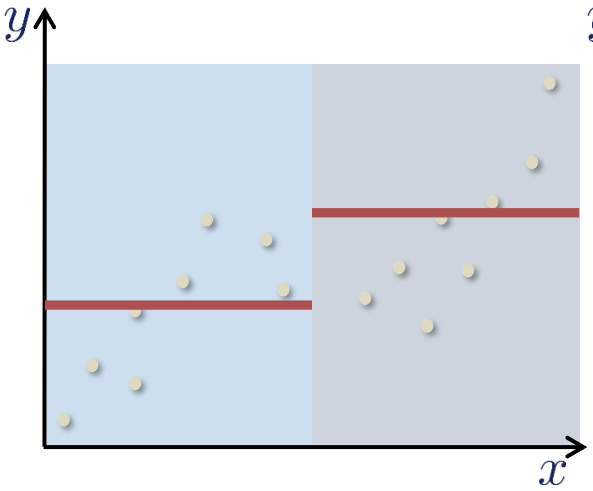
In general $\boldsymbol{\phi}$ may select only a very small subset of features $\boldsymbol{\phi}(\mathbf{v}) : \mathbb{R}^d \rightarrow \mathbb{R}^{d'+1}, \;\; d' << d$

Regression forest: the predictor model

What do we do at the leaf?

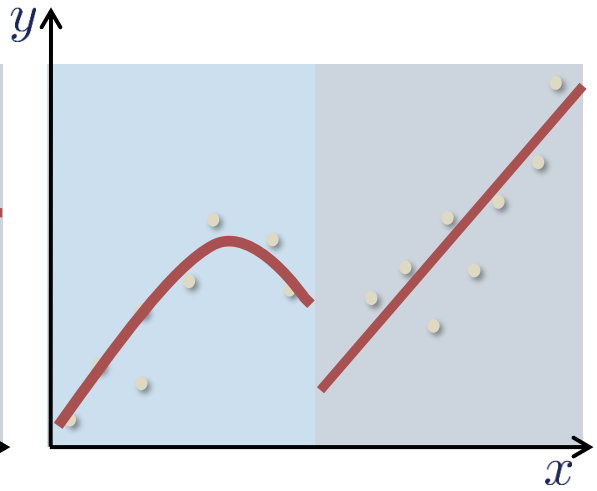


Examples of leaf (predictor) models



Predictor model: constant

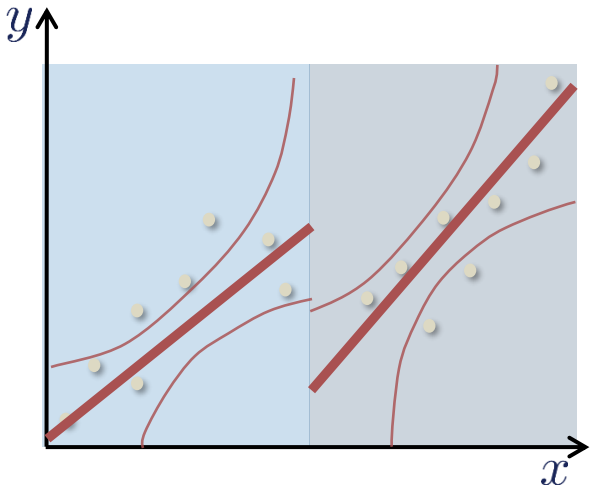
$$y = const$$



Predictor model: polynomial

$$y = \sum_{i=0}^{n} w_i x^i$$

(note: linear for n=1, constant for n=0)

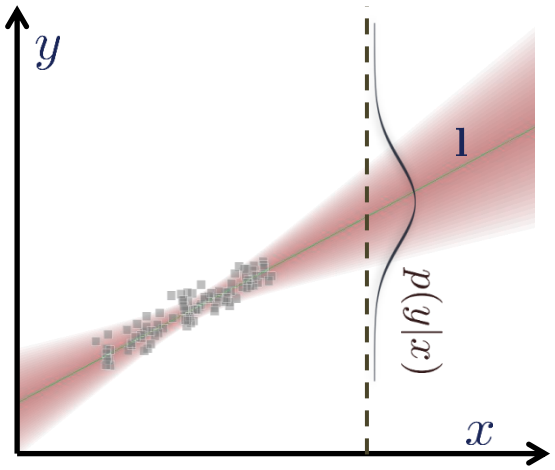


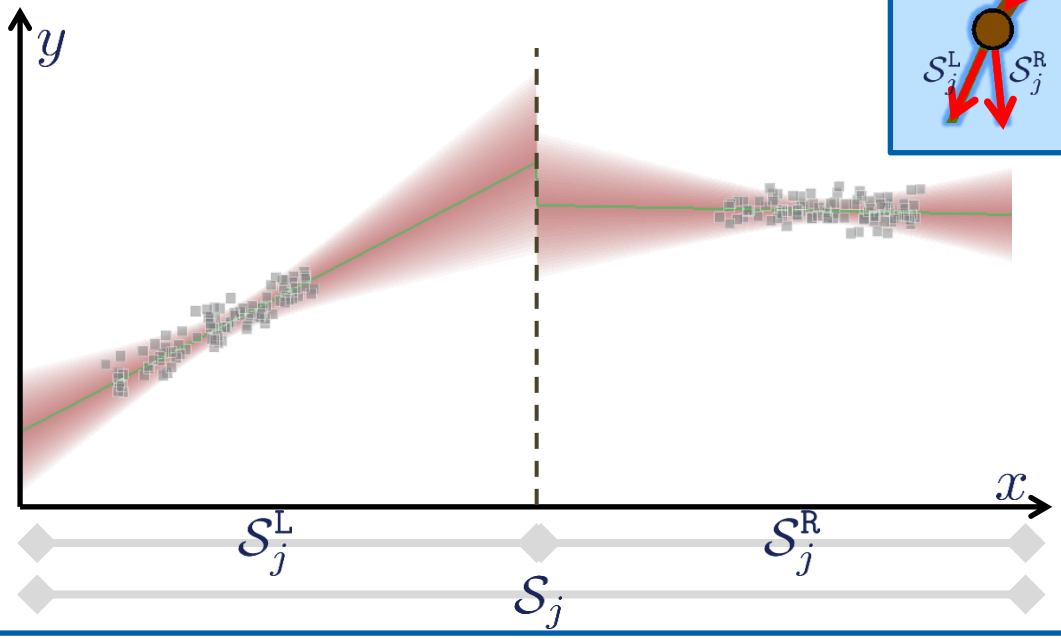
Predictor model: probabilistic-linear

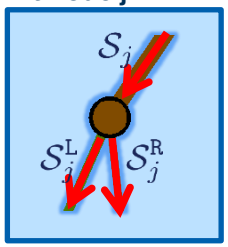
$$y = l_1 x + l_2 \qquad p(y|x)$$

Regression forest: objective function

Computing the regression information gain at node j







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

$$H(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \int_y p(y|x) \log p(y|x)\, dy \qquad p(y|x) \sim N\left(y; \overline{y}, \sigma_y^2(x)\right)$$

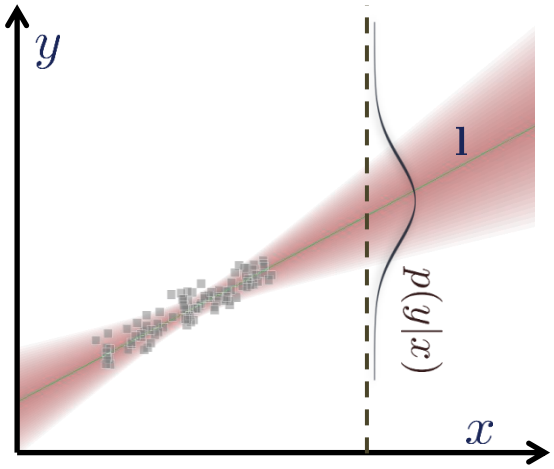
$$H(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \frac{1}{2} \log\left((2\pi e)^2 \sigma_y^2(x)\right)$$

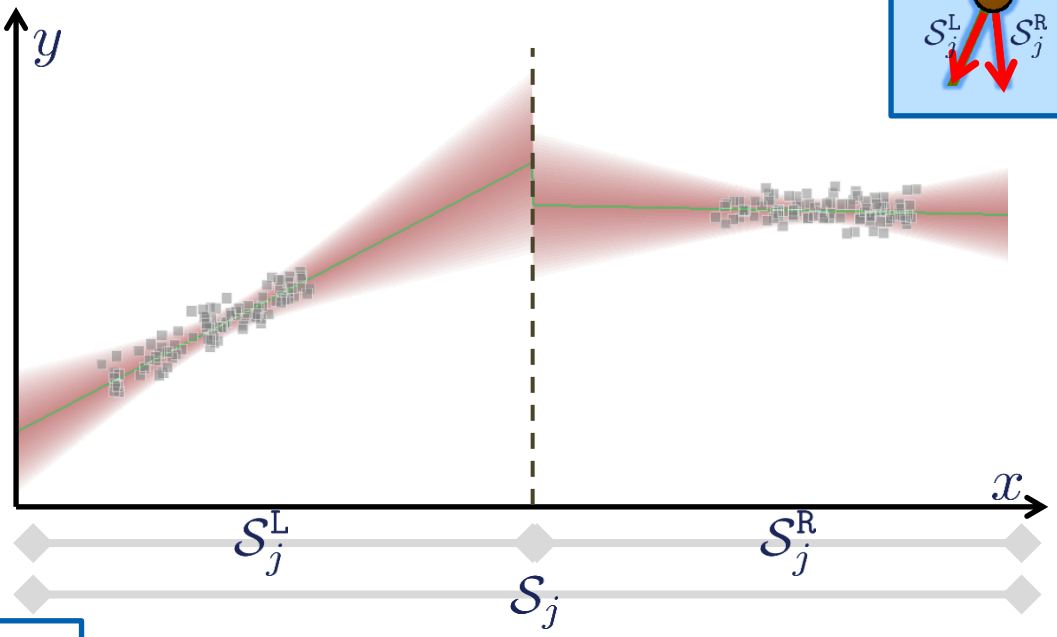
Differential entropy of Gaussian

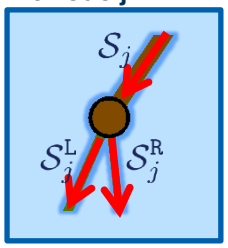
$$|\mathcal{S}_j|\, I = \sum_{(x,y) \in \mathcal{S}_j} \log\left(\sigma_y(x)\right) - \sum_{i \in \{\mathtt{L},\mathtt{R}\}} \left(\sum_{(x,y) \in \mathcal{S}_j^i} \log\left(\sigma_y(x)\right) \right)$$

Regression information gain

Regression forest: objective function







Comparison with Breiman's error of fit (CART)

$$I = \sum_{(x,y)\in\mathcal{S}_j} \log\left(\sigma_y(x)\right) - \sum_{i\in\{\mathtt{L},\mathtt{R}\}} \left(\sum_{(x,y)\in\mathcal{S}_j^i} \log\left(\sigma_y(x)\right) \right)$$

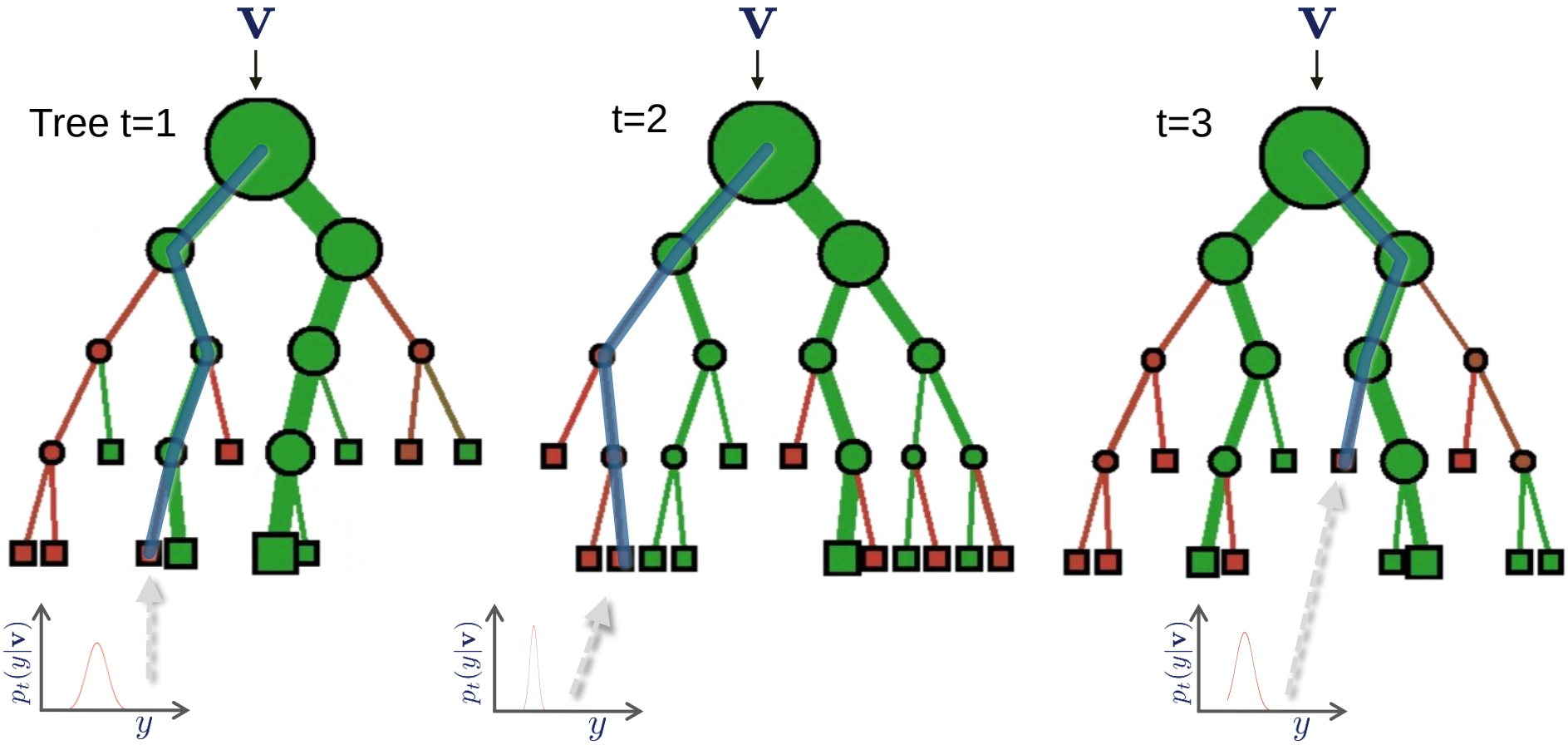
Differential entropy of Gaussian

$$E = \sum_{(x,y)\in\mathcal{S}_j} \left(y - \overline{y}_j\right)^2 - \sum_{\mathtt{i}\in\{\mathtt{L},\mathtt{R}\}} \left(\sum_{(x,y)\in\mathcal{S}_j^{\mathtt{i}}} \left(y - \overline{y}_j\right)^2 \right)$$

Error of fit

The error of fit objective is a special instance of our more general information-theoretical objective function.

Regression forest: the ensemble model



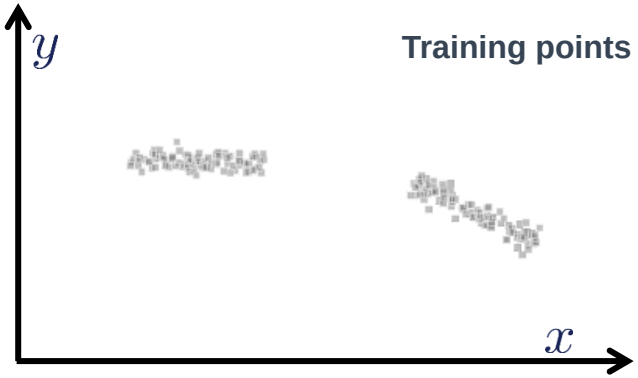
The ensemble model

Forest output probability $$p(\mathbf{y}|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(\mathbf{y}|\mathbf{v})$$

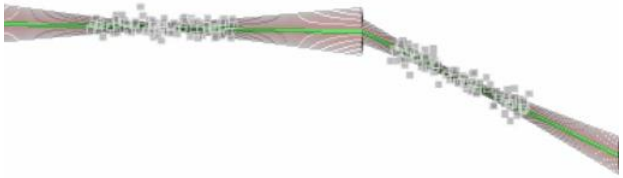
Regression forest: probabilistic, non-linear regression

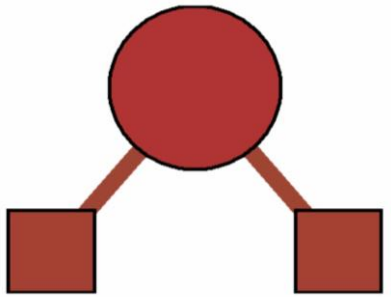
UNIVERSITÄT BONN

Training different trees in the forest (max tree depth D=2)



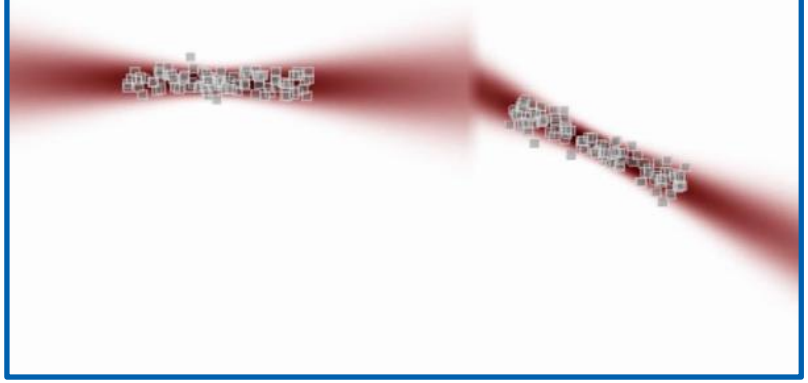
Training points



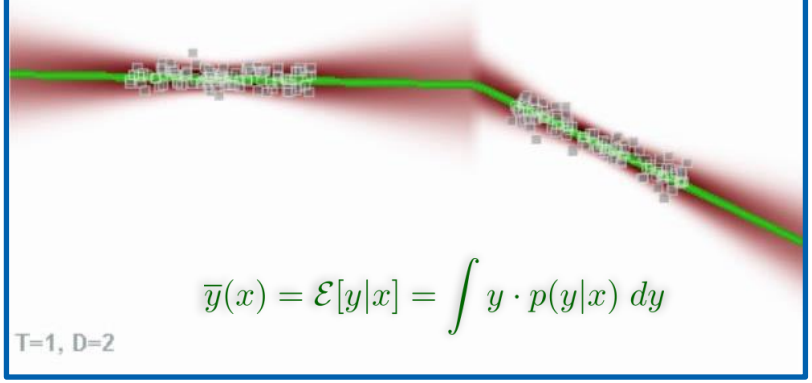


Testing different trees in the forest

Tree posteriors



Forest posterior



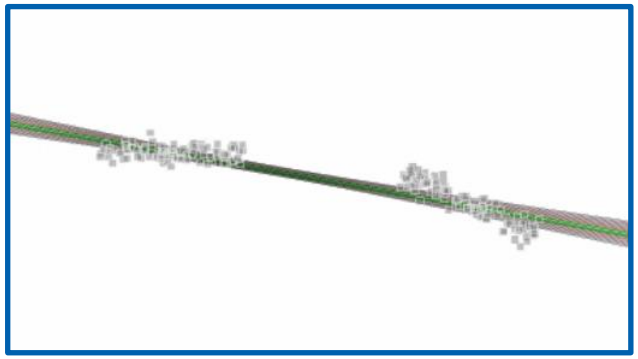
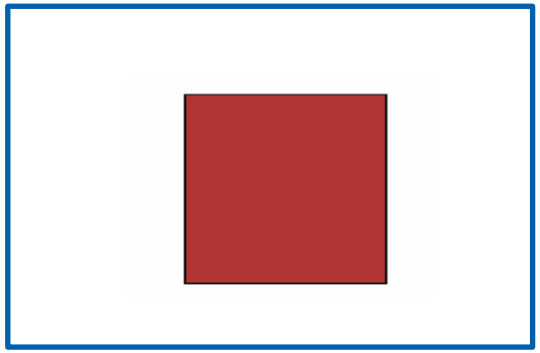
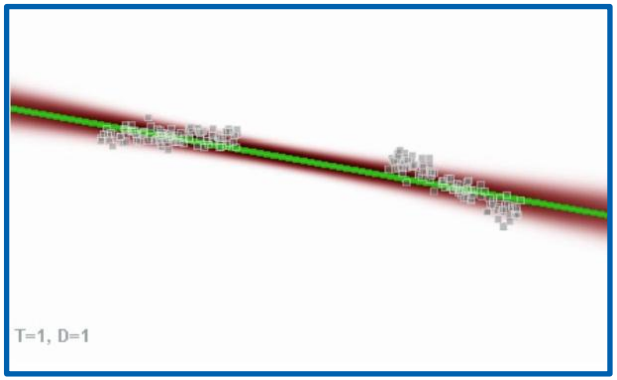
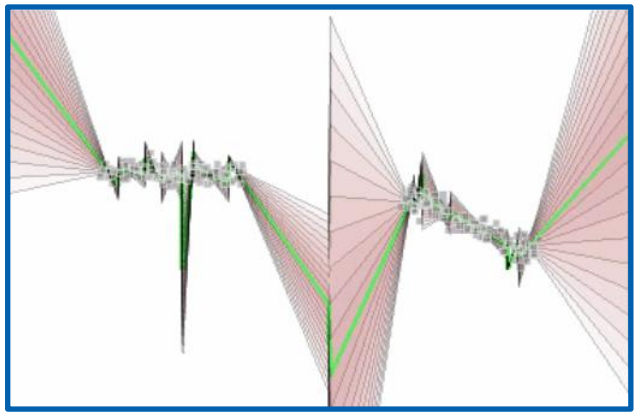
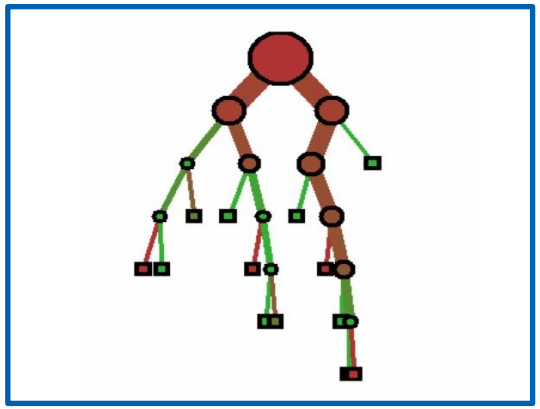
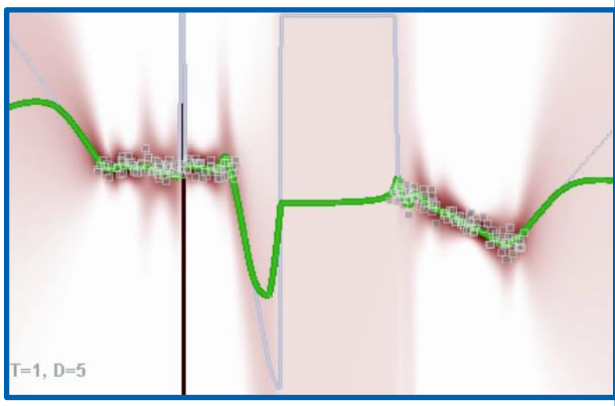
Generalization properties:

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

Parameters: T=400, D=2, weak learner = aligned, leaf model = prob. line

Regression forest: probabilistic, non-linear regression

UNIVERSITÄT BONN

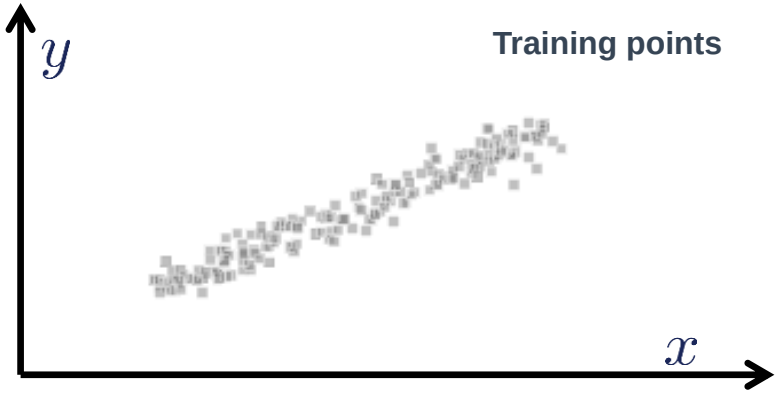
	Training		Testing
underfitting	max tree depth D = 1		max tree depth D = 1
			
overrfitting	max tree depth D = 5		max tree depth D = 5
			

Parameters: T=400, w.l. = aligned, l.m. = prob. line

Regression forest: probabilistic, non-linear regression

UNIVERSITÄT BONN

Generalizing conventional linear regression



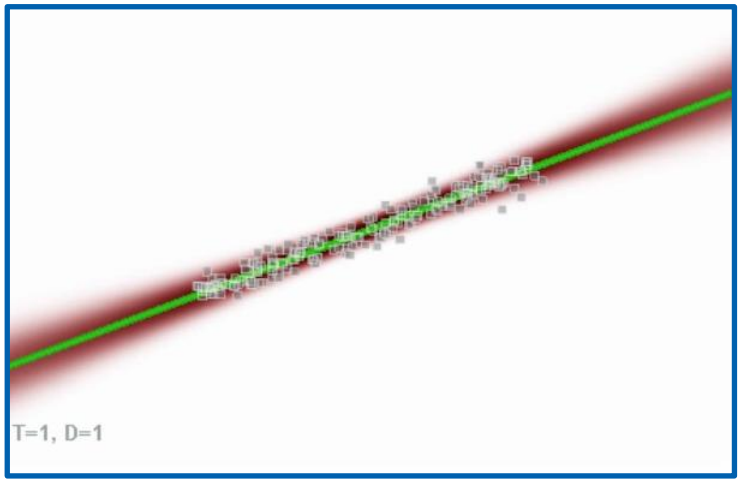
mean

$$\overline{y}(x) = \mathcal{E}[y|x] = \int y \cdot p(y|x)\, dy$$

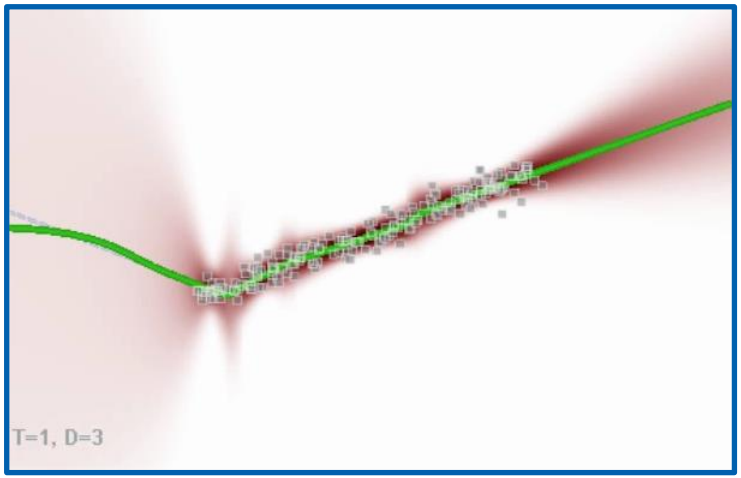
mode

$$\hat{y}(x) = \arg\max_y p(y|x)$$

max tree depth, D = 1



max tree depth, D = 3

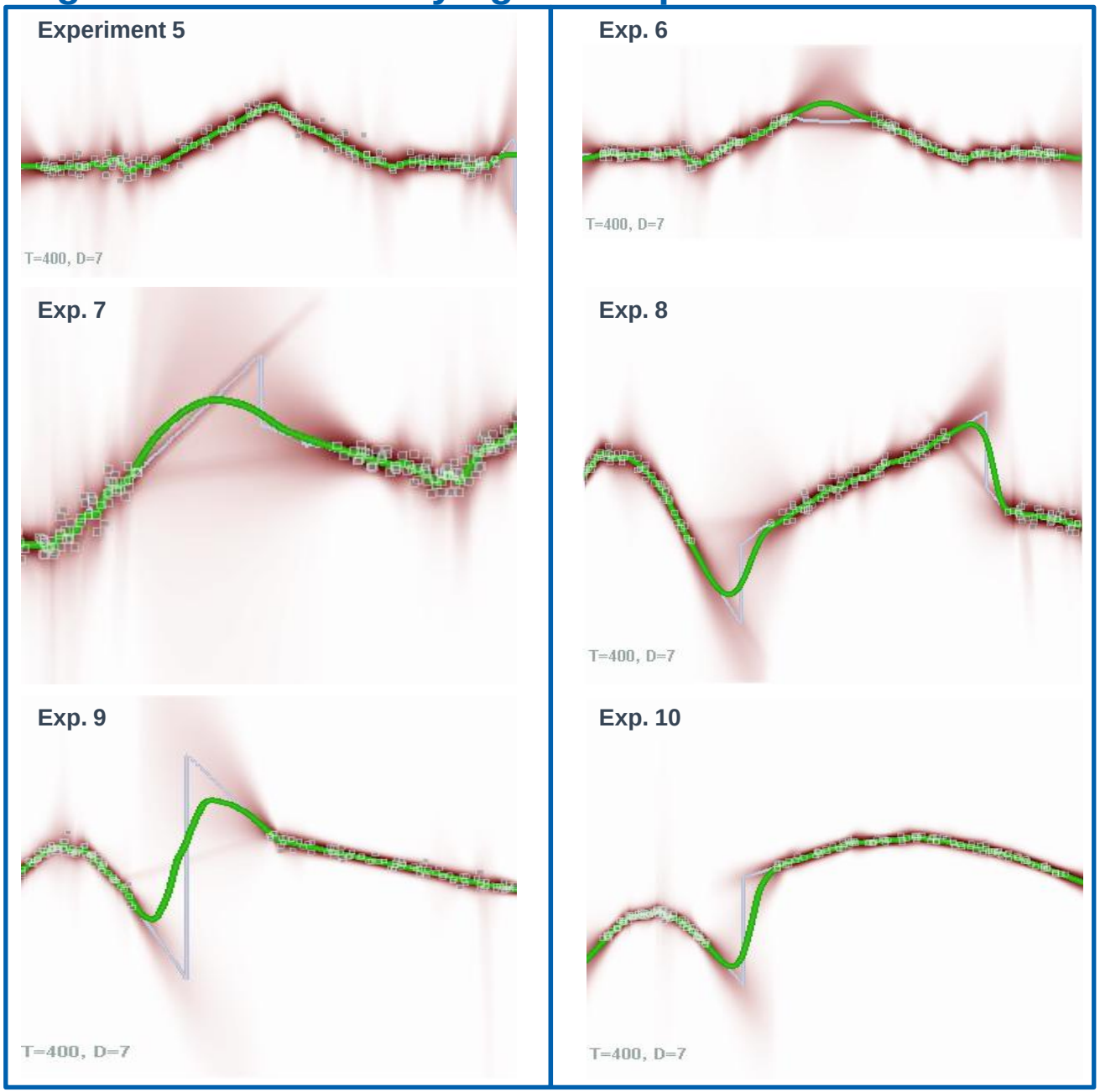


overfitting

Parameters: T=400, w.l. = aligned, l.m. = probabilistic line

(2 videos in this page)

Regression forest: varying tree depth D



Forest posteriors for six different experiments and varying depth D.

mean

$$\overline{y}(x) = \mathcal{E}[y|x] = \int y \cdot p(y|x)\, dy$$

mode

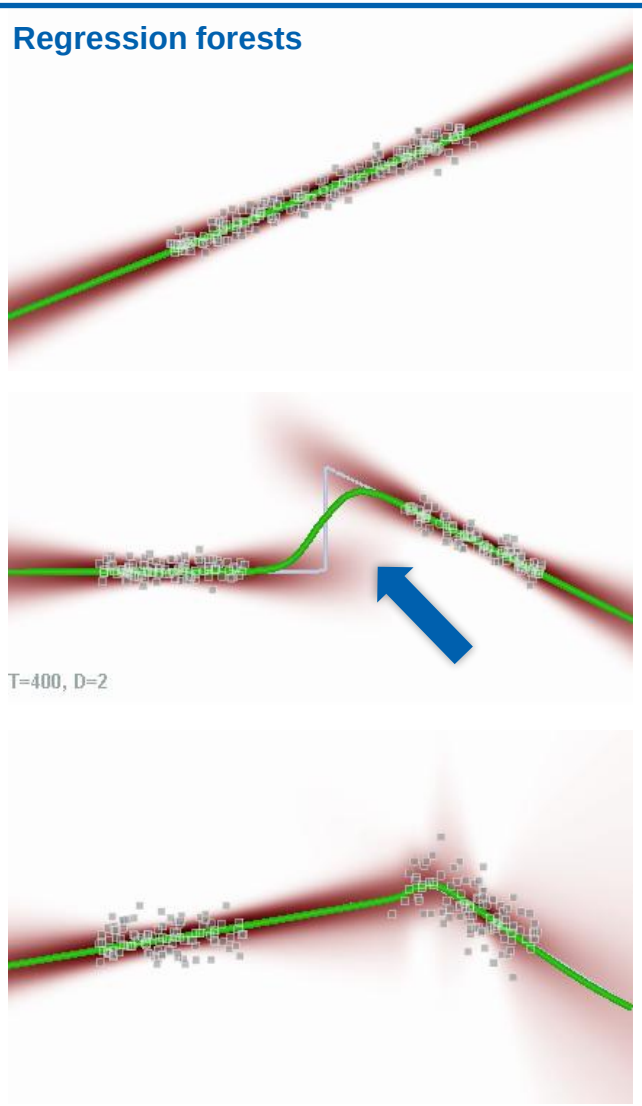
$$\hat{y}(x) = \arg\max_{y} p(y|x)$$

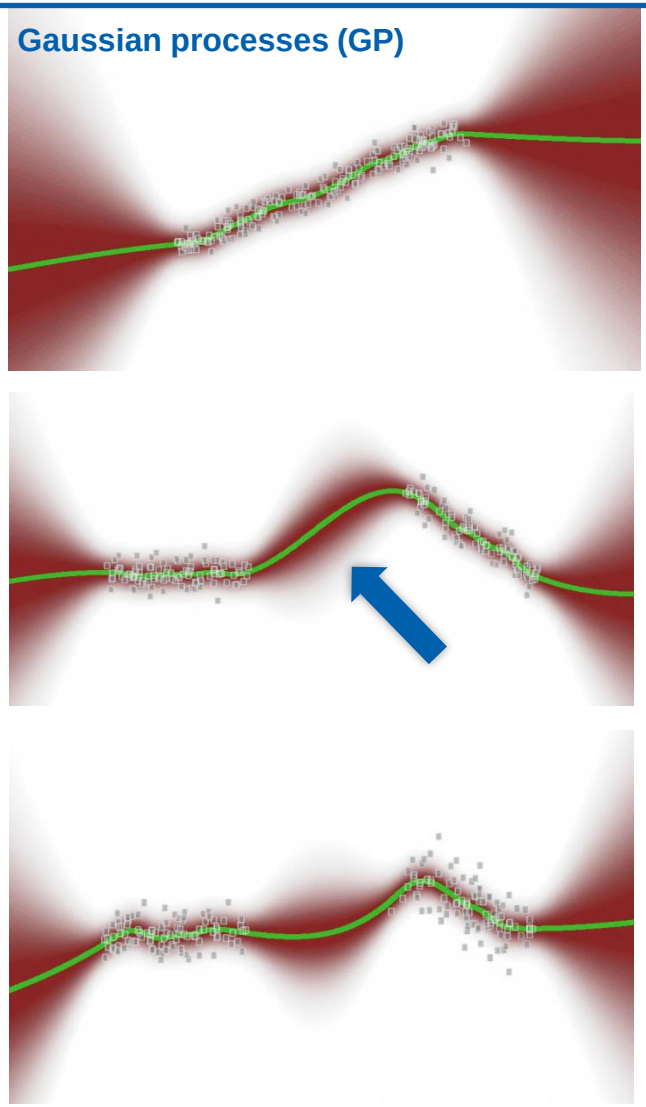
Parameters: T=400, w.l. = aligned, l.m. = prob. line

Regression forest: comparison with Gaussian processes

UNIVERSITÄT BONN

In regr. forests the posterior can be multimodal





In GP the posterior is Gaussian (unimodal)

Observations

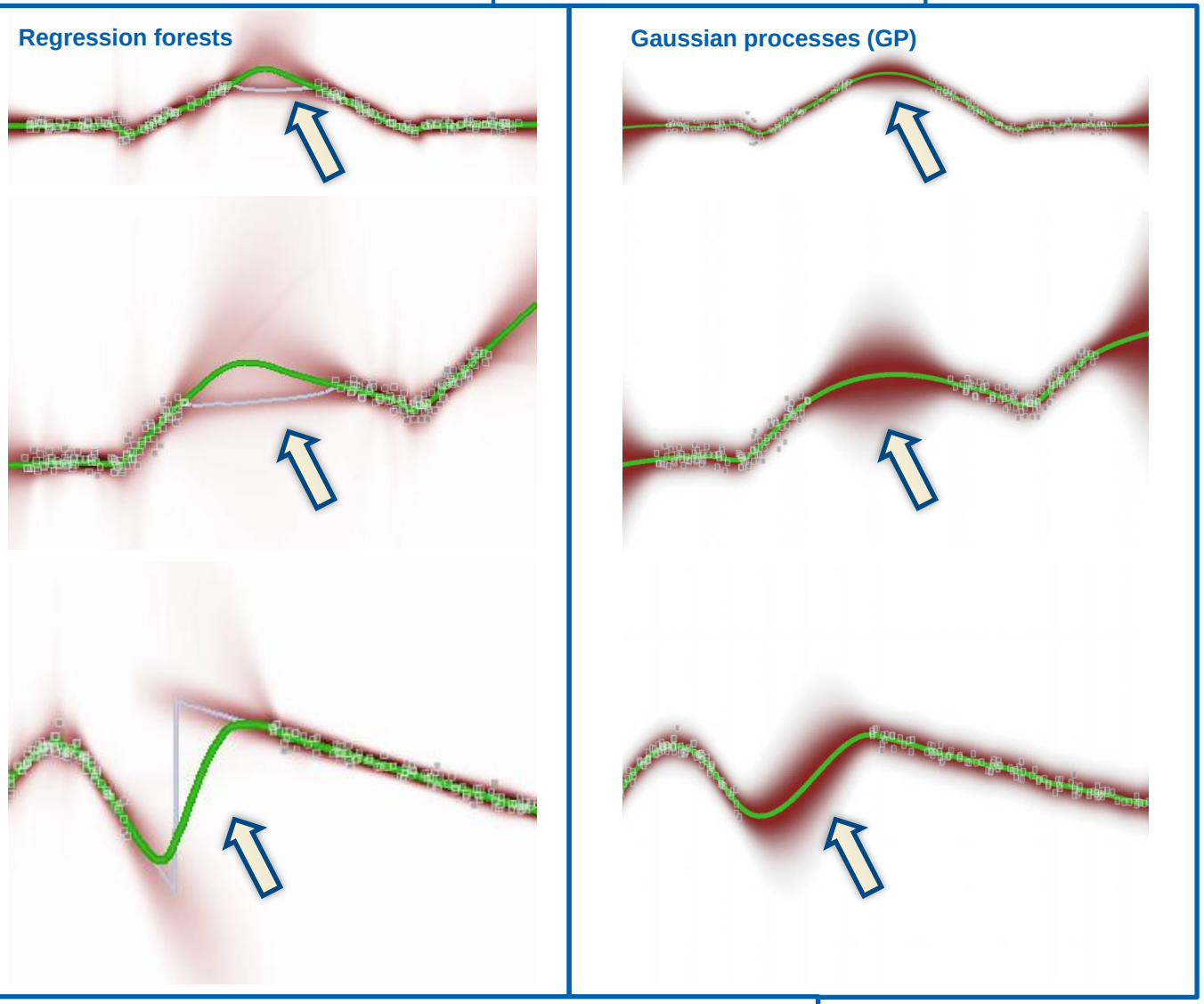
- Similar behaviour of the mean. – Forests can capture multimodality.
- Forests are a parallel algorithm that does not need large matrix inversion.
- In forests the quality and shape of the uncertainty depends on the predictor model.

[GP code from http://gaussianprocess.org/gpml/]

Regression forest: comparison with Gaussian processes

UNIVERSITÄT BONN

In regr. forests the posterior can be multimodal



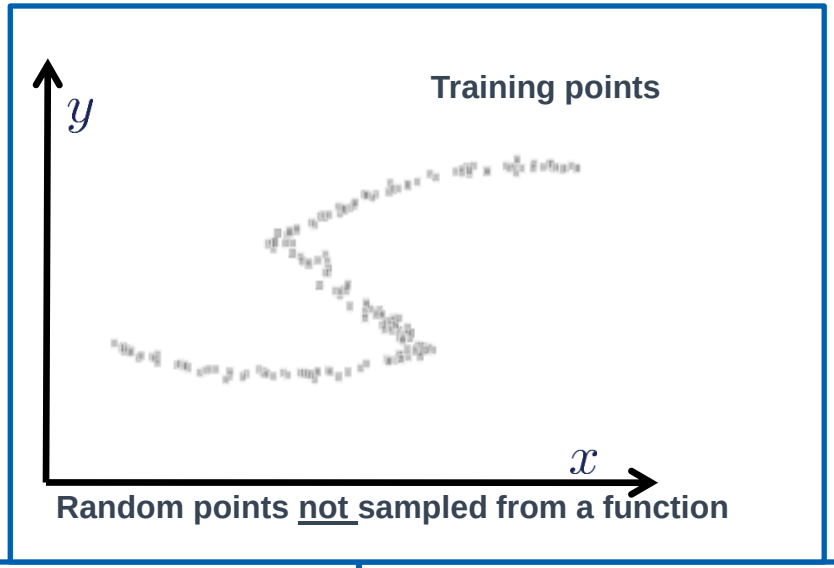
In GP the posterior is Gaussian (unimodal)

Observations

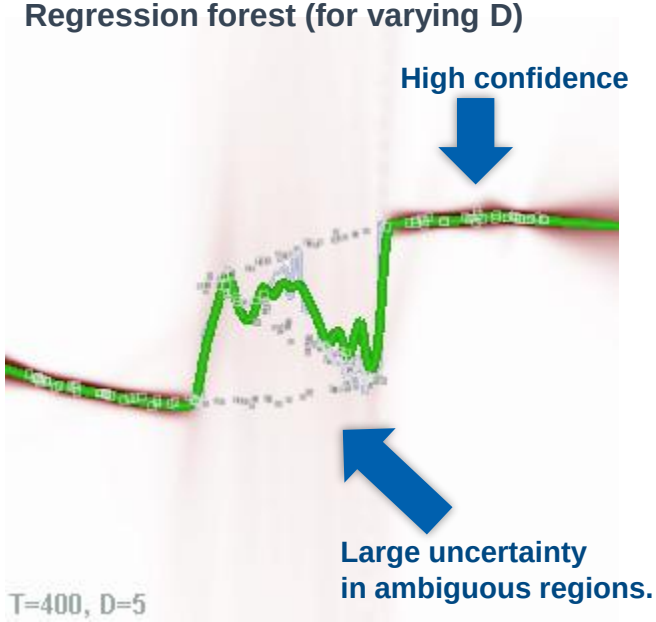
- Similar behaviour of the mean. – Forests can capture multimodality.
- Forests are a parallel algorithm that does not need large matrix inversion.
- In forests the quality and shape of the uncertainty depends on the leaf model.

[GP code from http://gaussianprocess.org/gpml/]

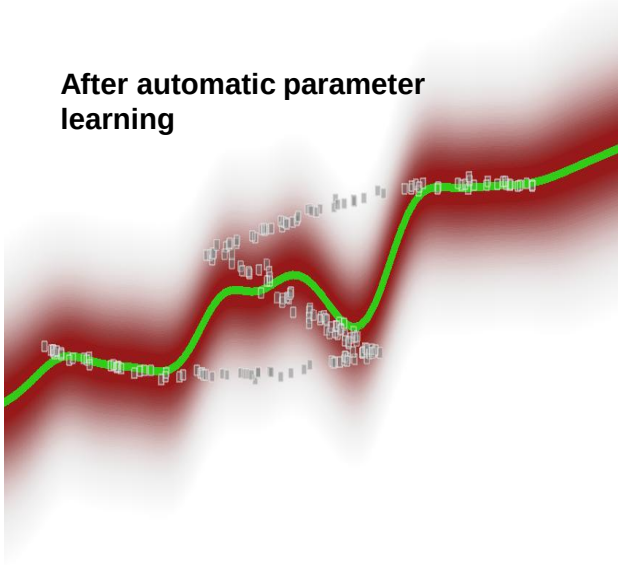
Regression forest: dealing with non-functions



Regression forest (for varying D)



Gaussian process (for varying hyper parameters)



GPs do not seem able to capture the large ambiguity in the middle.

Parameters: T=400, w.l. = aligned, l.m. = prob. line

[GP code from http://gaussianprocess.org/gpml/]

Theory

- Assuming that
  - training samples are i.i.d.
  - trees are trained independently of estimating leaf statistics
- Number of splitting functions per node are sampled from Poisson distribution
- Stopping cirteria: Minimum samples per leaf *k(n)*
- Regression forests converge as the amount of training data $n \rightarrow \infty$

[M. Denil et al. **Narrowing the Gap: Random Forests In Theory and In Practice.** JMLR 2014]

Convergence

Theorem 1 *Suppose that feature X is supported on $\mathbb{R}^d$ and has a density which is non-zero almost everywhere. Moreover, suppose that $\mathbb{E}[\theta|X=x]$, which is the (unknown) mean prediction given x, is bounded and that $\mathbb{E}[\theta^2] < \infty$. Let $A(n)$ denote the training data and Z_t the randomness of a tree. Then*

$$\mathbb{E}_{X,A(n),Z_t}\left[\left|\frac{1}{T}\sum_{t}^{T}\mathbb{E}[\theta|X,A(n),Z_t] - \mathbb{E}[\theta|X]\right|^2\right] \to 0$$

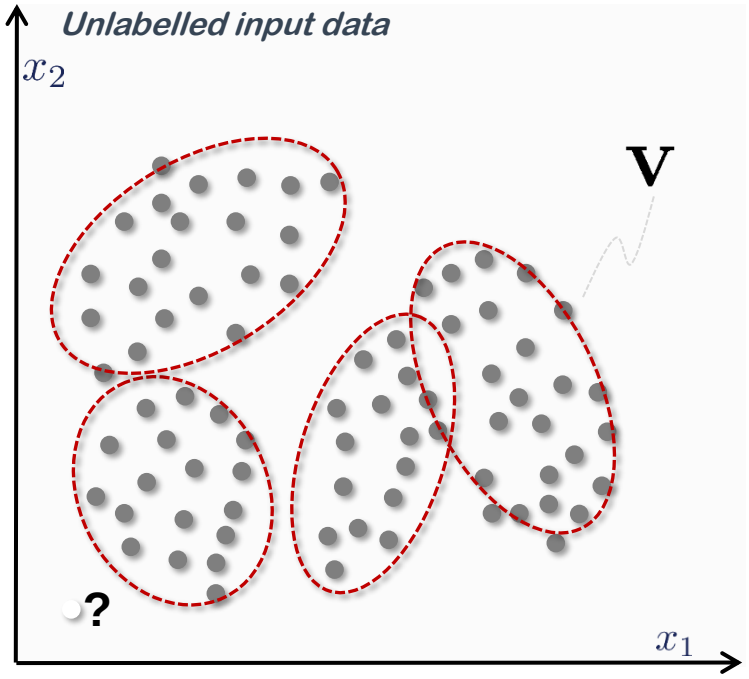
provided that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$.

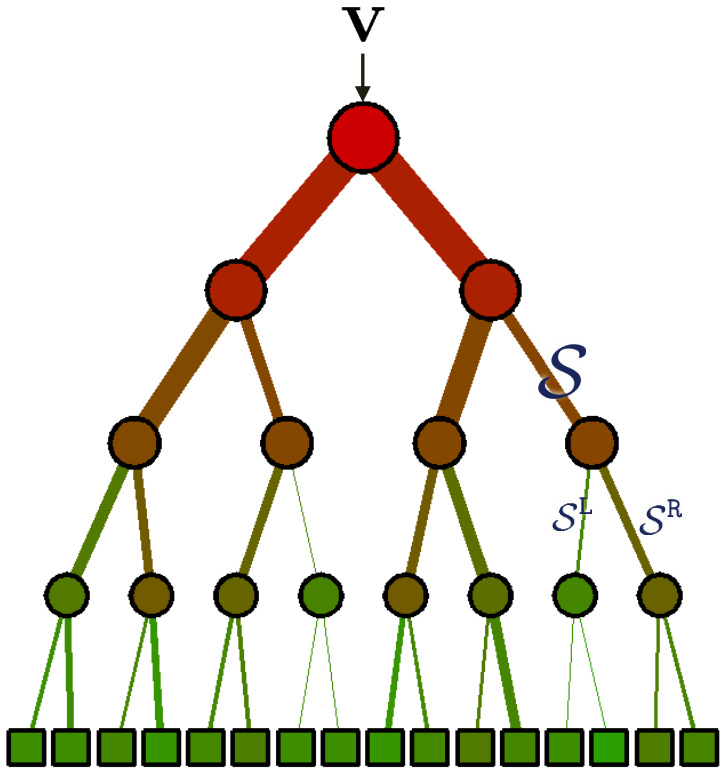
[M. Denil et al. **Narrowing the Gap: Random Forests In Theory and In Practice.** JMLR 2014]

Overview

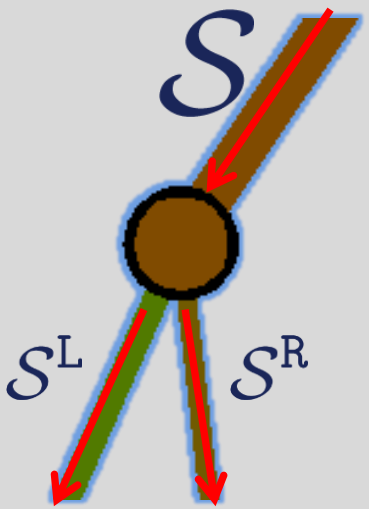
- Classification forests
- Regression forests
- Other variants

Density forest model





Training and information gain



Information gain

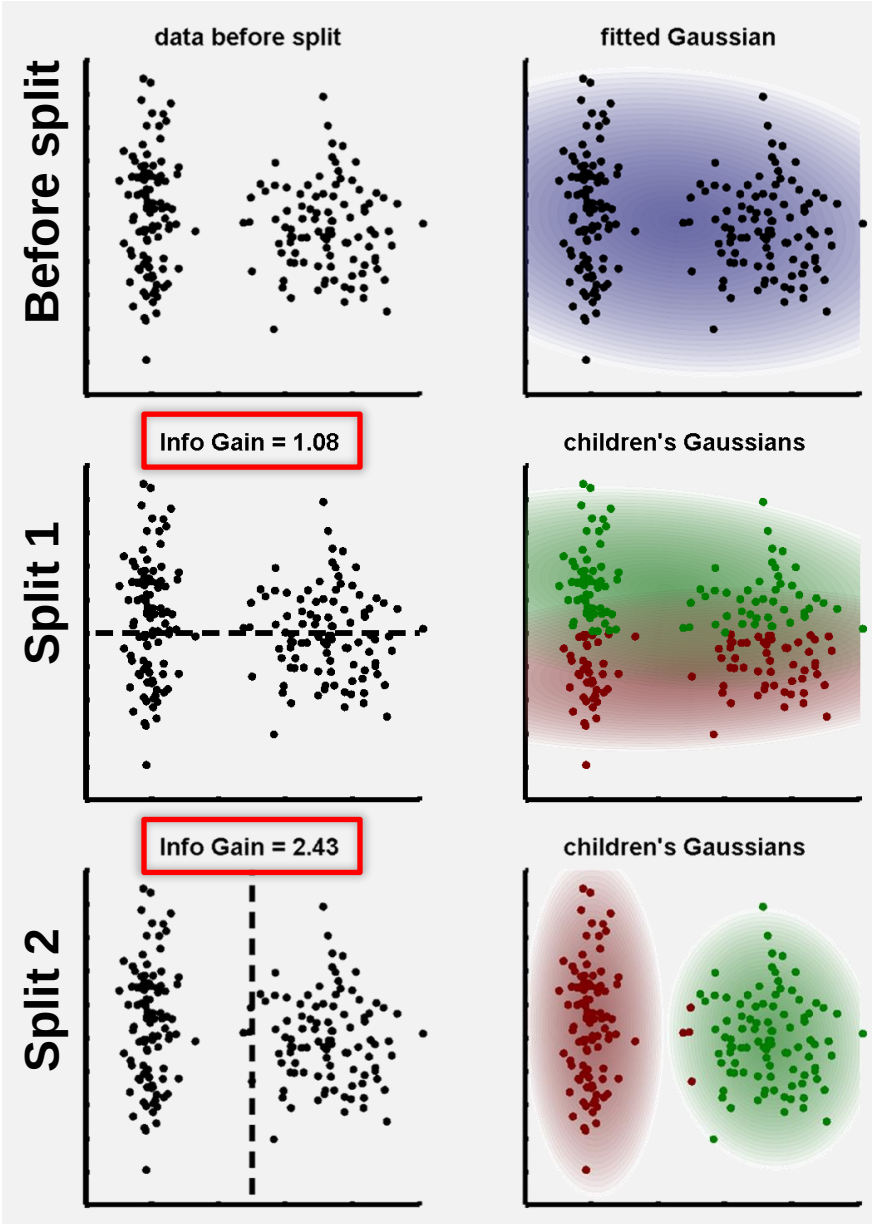
$$I = H(\mathcal{S}_j) - \sum_{i \in \{\mathrm{L},\mathrm{R}\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}|} H(\mathcal{S}_j^i)$$

Differential entropy of Gaussian

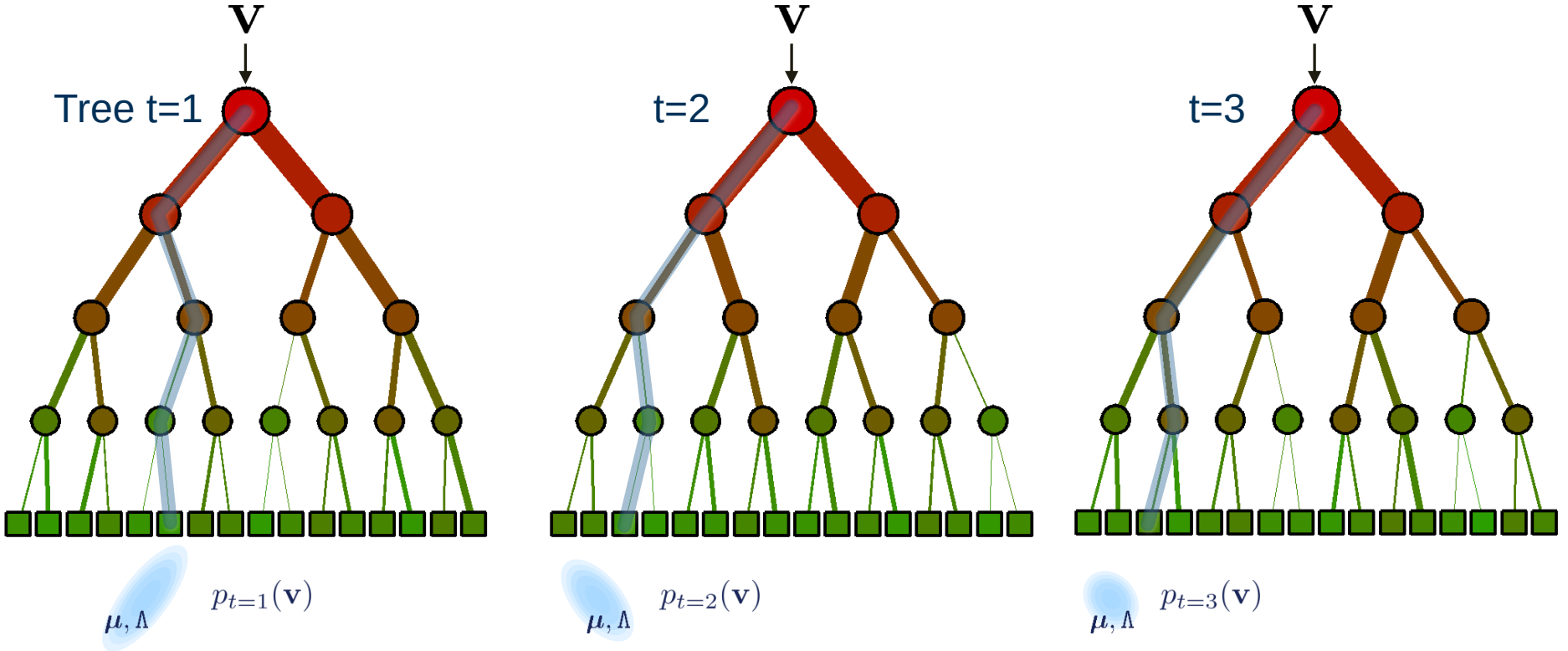
$$H(\mathcal{S}) = \frac{1}{2} \log\left((2\pi e)^d |\Lambda(\mathcal{S})|\right)$$

Node training

$$\boldsymbol{\theta}_j = \arg\max_{\boldsymbol{\theta} \in \mathcal{T}_j} I(\mathcal{S}_j, \boldsymbol{\theta})$$



Density forest model



The prediction and ensemble models

Output of tree $p_t(\mathbf{v}) = \frac{\pi_{l(\mathbf{v})}}{Z_t} \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_{l(\mathbf{v})}, \Lambda_{l(\mathbf{v})})$

with $\pi_{l(\mathbf{v})} = |\mathcal{S}_{l(\mathbf{v})}|/|\mathcal{S}_0|$

The ensemble model $p(\mathbf{v}) = \frac{1}{T} \sum_{t=1}^{T} p_t(\mathbf{v})$

Partition function

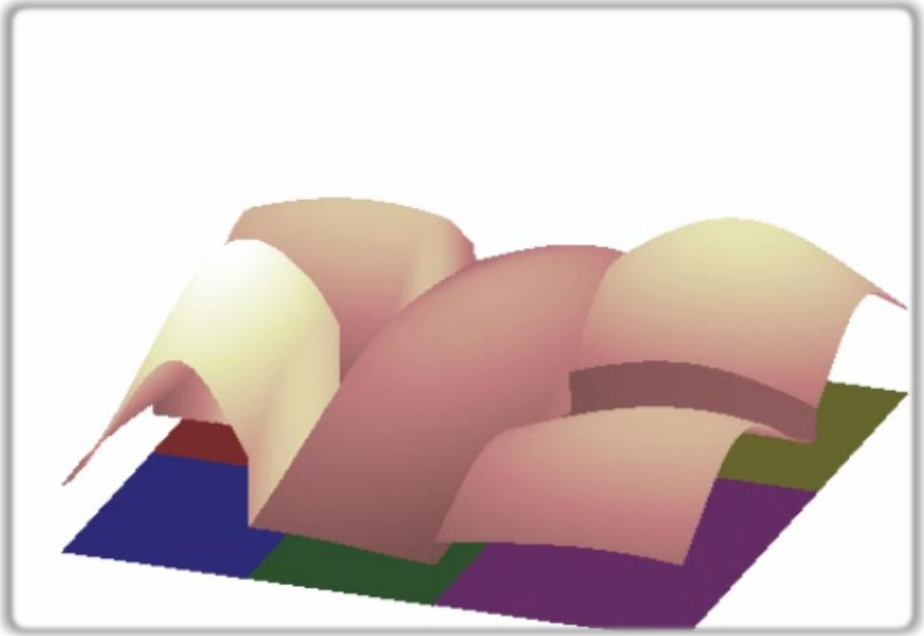
$$Z_t = \int_{\mathbf{v}} \left(\sum_l \pi_l \, \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_l, \Lambda_l) \, p(l|\mathbf{v}) \right) d\mathbf{v}$$

with $p(l|\mathbf{v}) = [\mathbf{v} \in l(\mathbf{v})]$

Partition function

$$Z_t = \int_{\mathbf{v}} \pi_{l(\mathbf{v})} \, \mathcal{N}\left(\mathbf{v}; \boldsymbol{\mu}_{l(\mathbf{v})}, \Lambda_{l(\mathbf{v})}\right) \, d\mathbf{v}$$

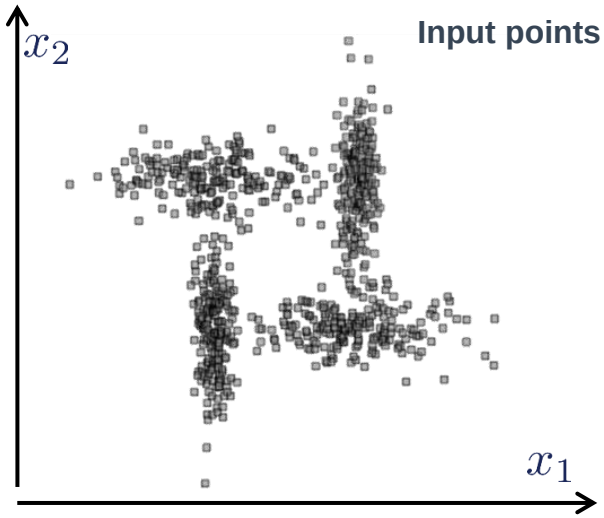
1. Compute via cumulative multivariate normal for ax.-align. w. learn.



2. Grid-based numerical integration

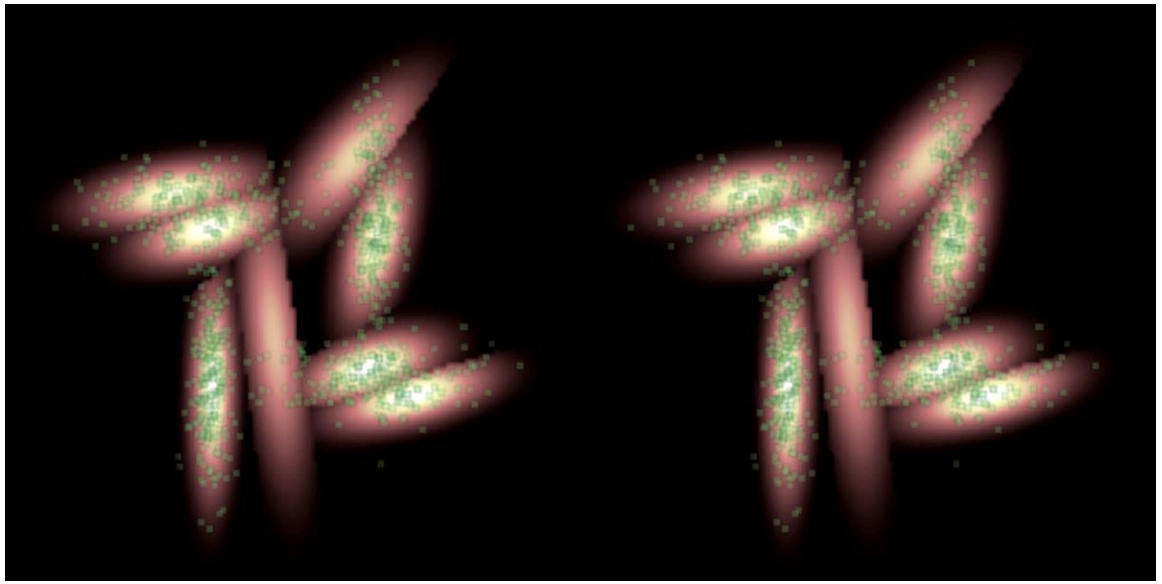
$$Z_t \approx \Delta \sum_i \pi_{l(\mathbf{v}_i)} \; \mathcal{N}(\mathbf{v}_i; \boldsymbol{\mu}_{l(\mathbf{v}_i)}, \Lambda_{l(\mathbf{v}_i)})$$

Density forest: evaluation

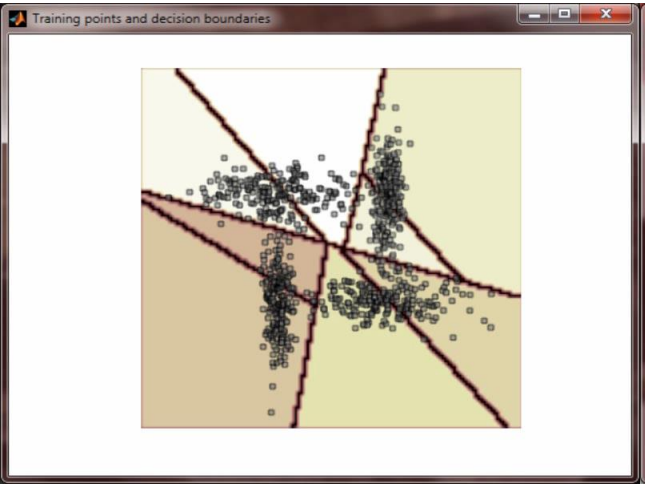


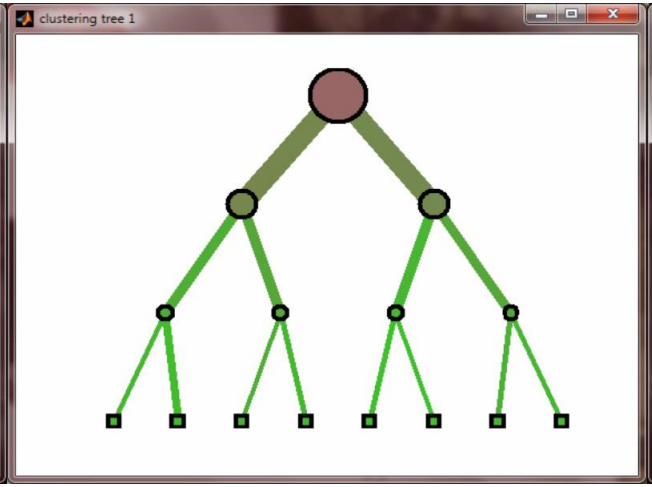
Tree density

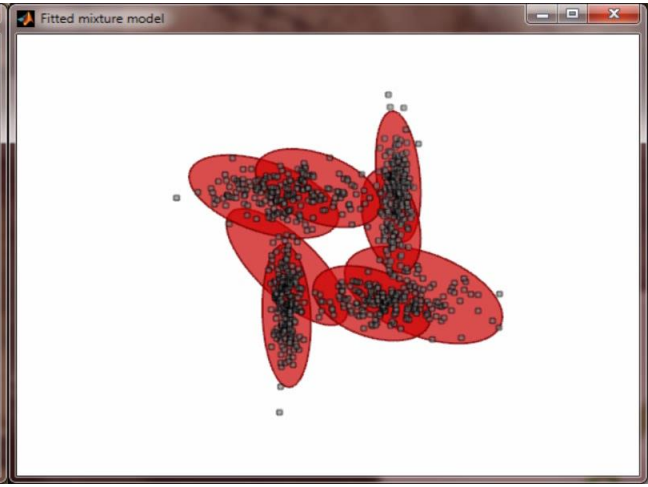
Forest density



Training different trees in the forest



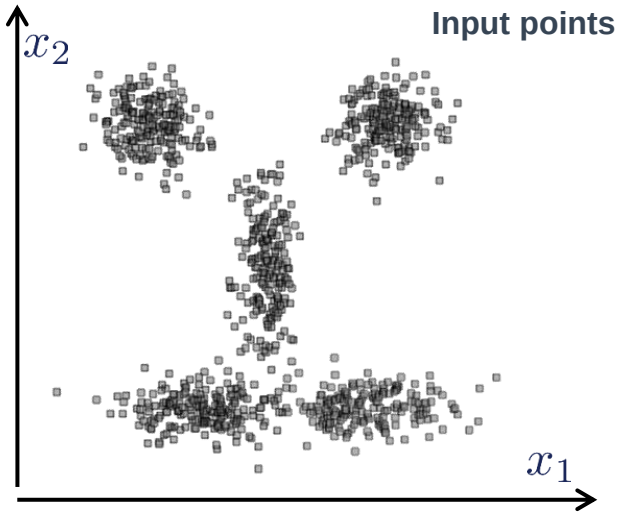




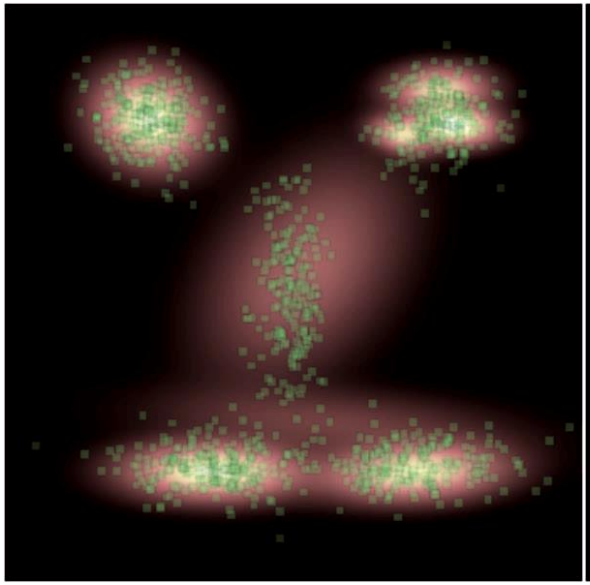
Parameters: T=400, D=4, weak learner = linear, predictor = Gaussian

Density forest: evaluation

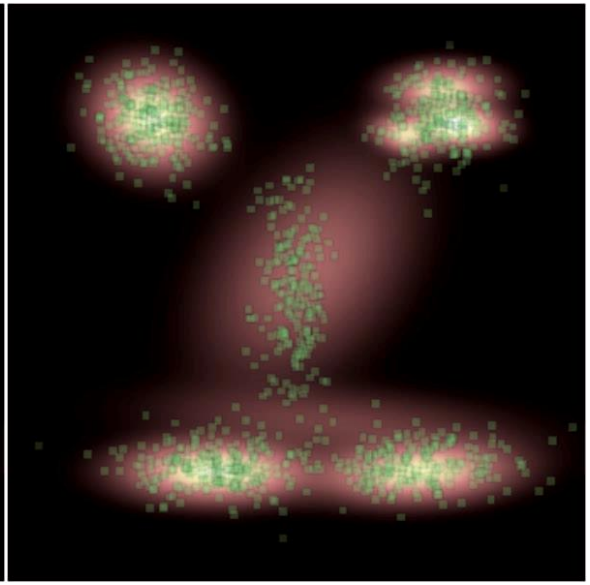
Input points



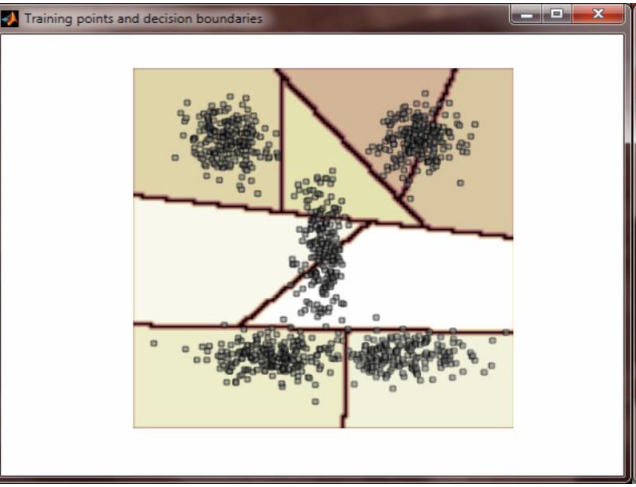
Tree density

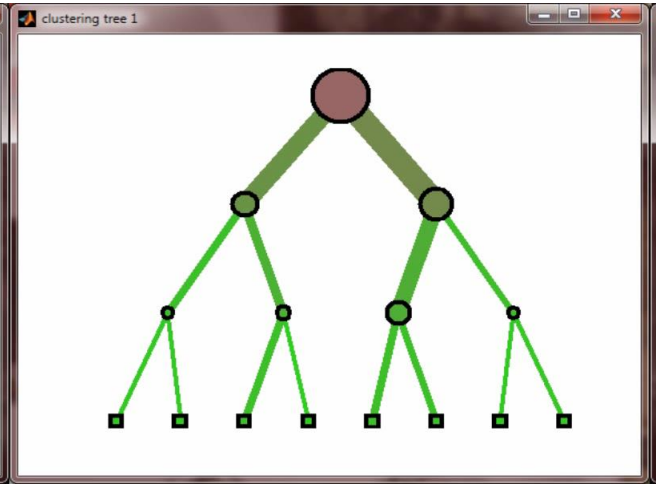


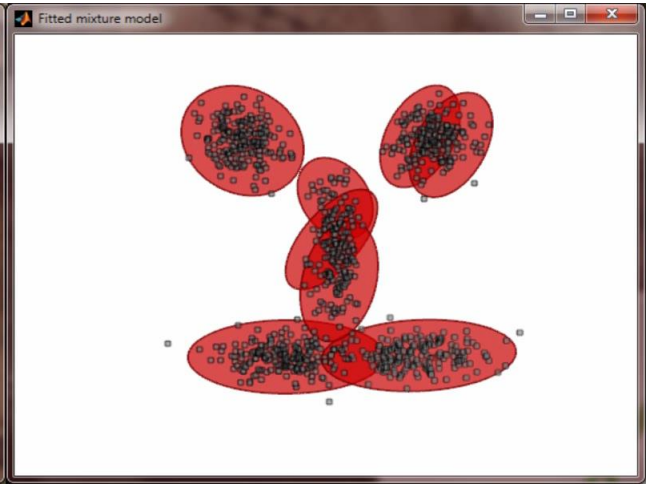
Forest density



Training different trees in the forest

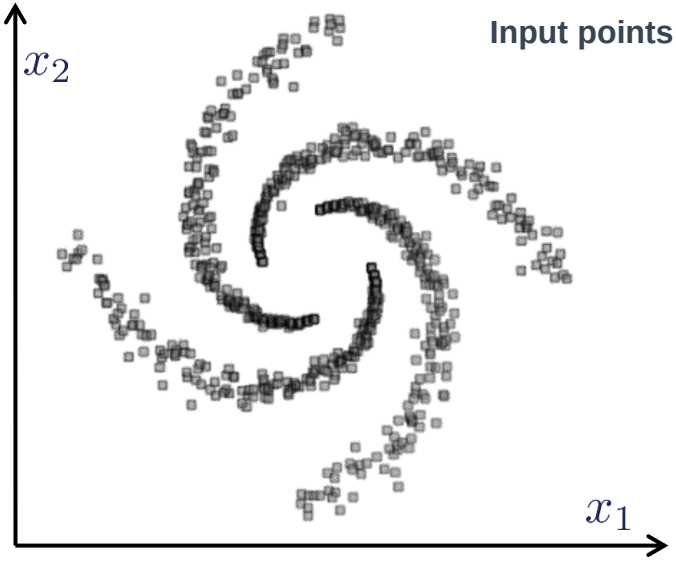


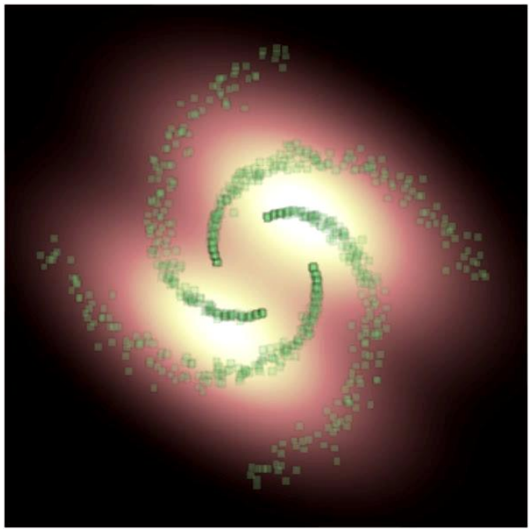




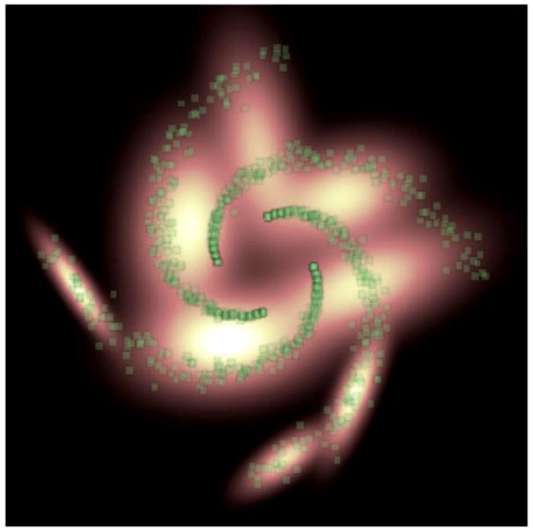
Parameters: T=400, D=4, weak learner = linear, predictor = Gaussian

Density forest: evaluation

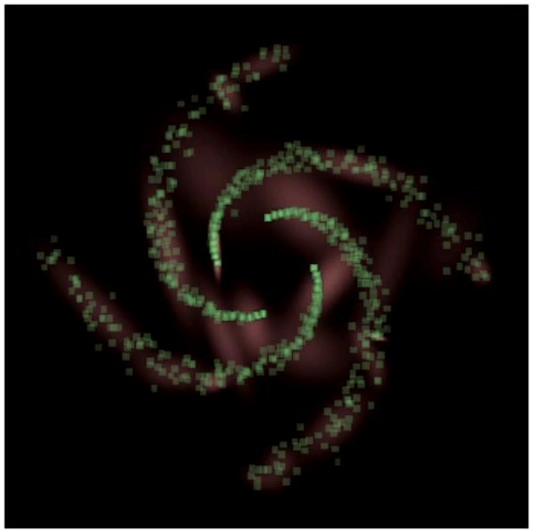




D=2



D=4

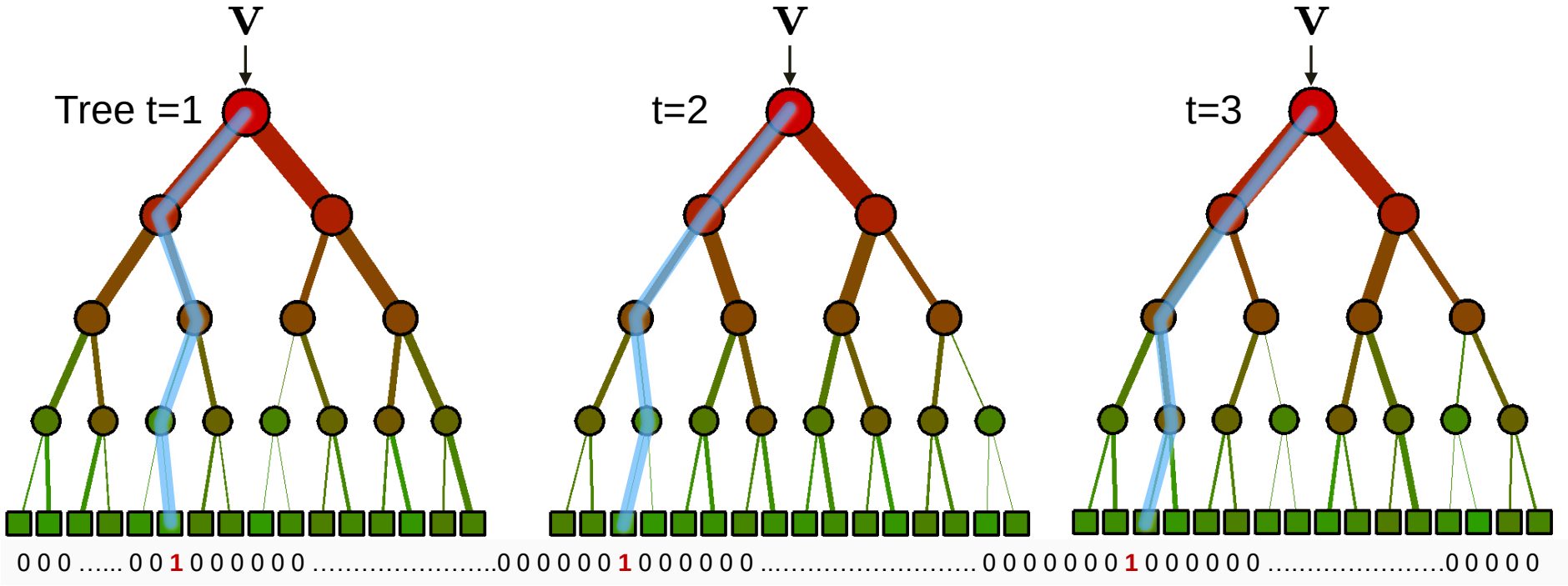


D=6

underfitting

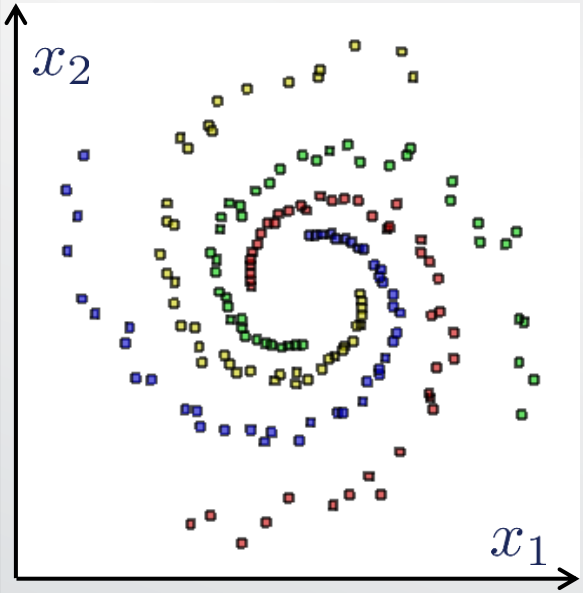
Parameters: T=400, D=4, weak learner = linear, predictor = Gaussian

Clustering trees for coding

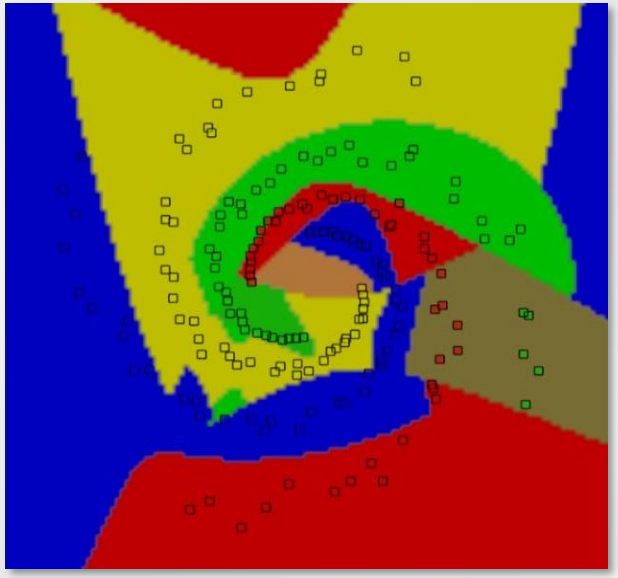


[F. Moosmann, E. Nowak, F. Jurie. **Randomized Clustering Forests for Image Classification.** IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 30(9): 1632-1646 (2008)]

Extremely randomized trees (ERT)

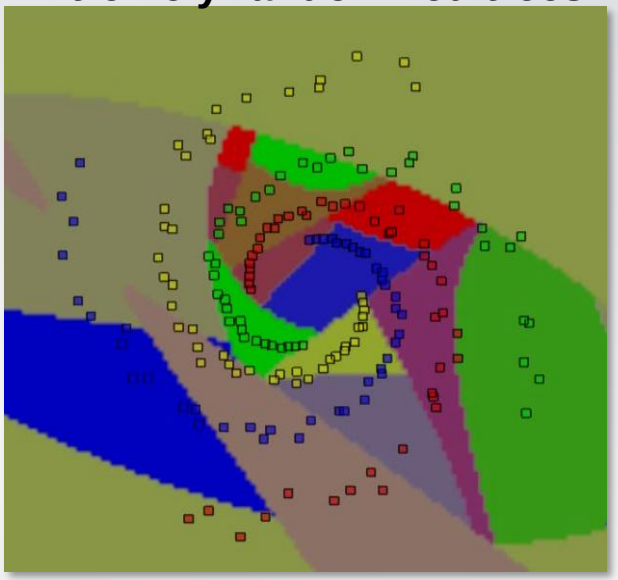


Classification forest



$\rho = 1000$

Extremely randomized trees



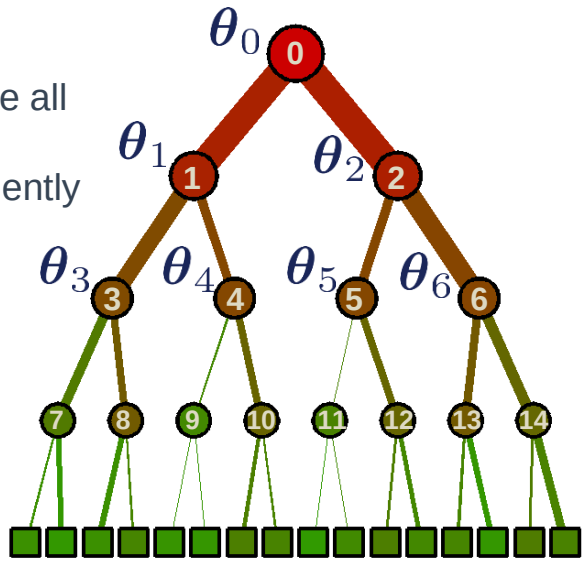
$\rho = 1$

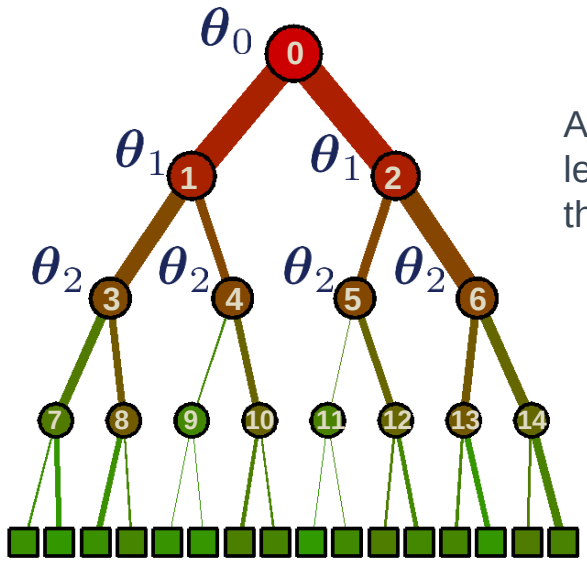
- **ERTs take longer to converge (more trees) and under-confident.**

Parameters: **T=200, D=13, weak learner = conic, predictor = prob.**

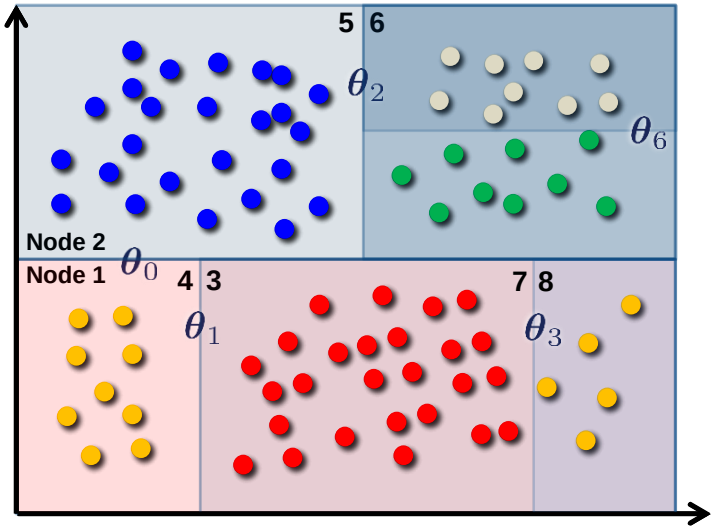
Forests v. ferns

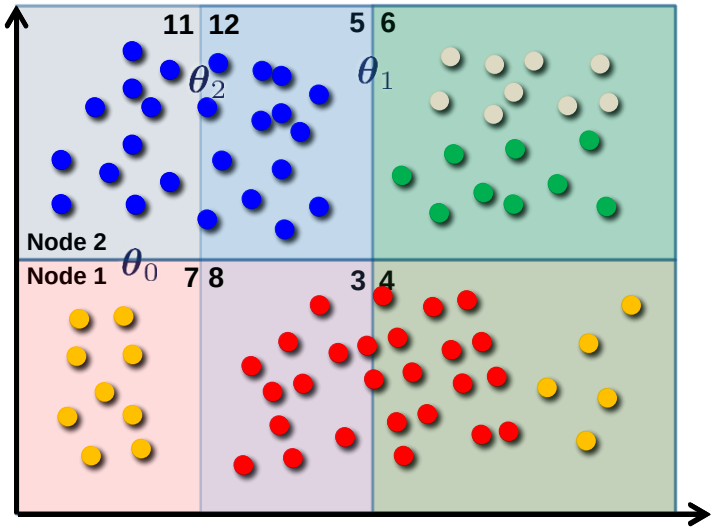
Nodes are all trained independently





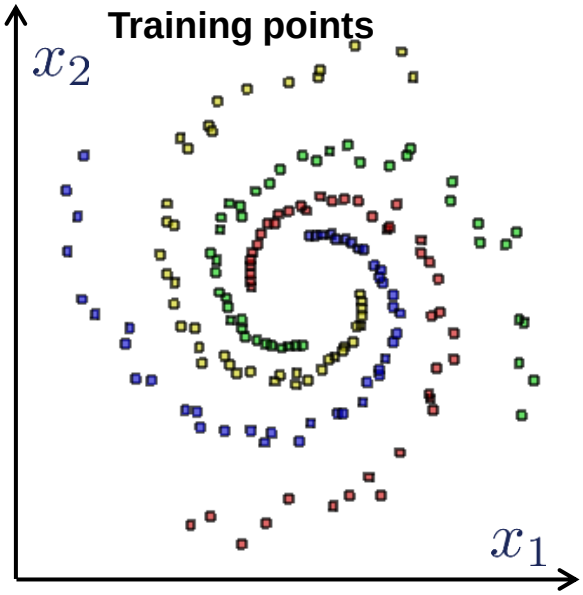
All nodes in the same level l are assigned the same test θ_l



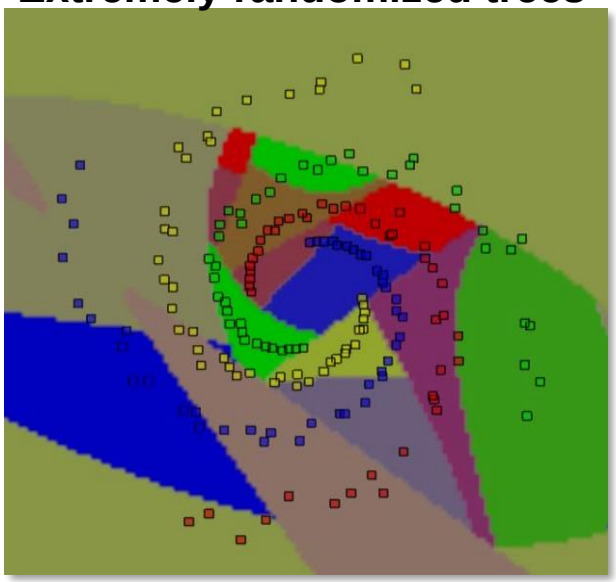


Ferns are simpler but less rich and less flexible.
By level 2 the tree has done more correct data splitting then the fern.

Forests v. ferns

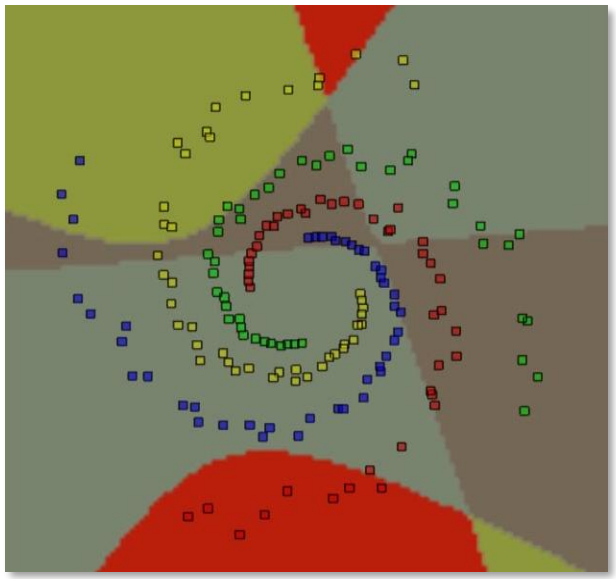


Extremely randomized trees



$\rho = 1$

Fern



$\rho = 1$

- Ferns take longer to converge (more trees) and under-confident.
- Ferns and ERTs have less parameters than forests and thus may over-fit less.

[M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. **Fast keypoint recognition using random ferns.** IEEE Trans. PAMI 32(3), 2001]

Parameters: T=200, D=13, weak learner = conic, predictor = prob.


UNIVERSITÄT
BONN