Advanced Video Content Analysis and Video Compression (5DD50), Module 9 cont'd

Classification with Random Forests

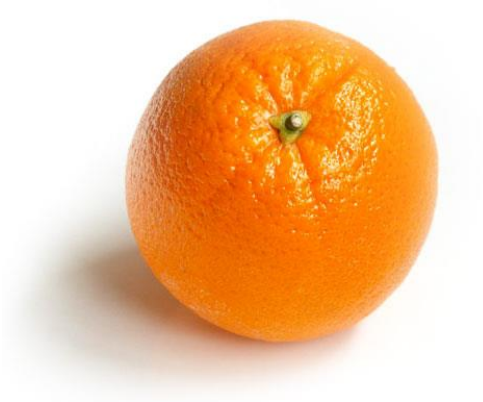
Fons van der Sommen
(f.v.d.sommen@tue.nl)

Classification: recap

- **Catogarized / labeled data**
  - Objects in a picture: chair, desk, person,…
  - Handwritten digits: → 3, → 6, → 5
  - Medical diagnosis: OCT image → T2 colon cancer
- **Goal: identify the class of a new data point**
- **Statistical modeling and machine learning**
- **Disctinctive properties (features)**

Classification: example (1/5)

* **Separate lemons from oranges**



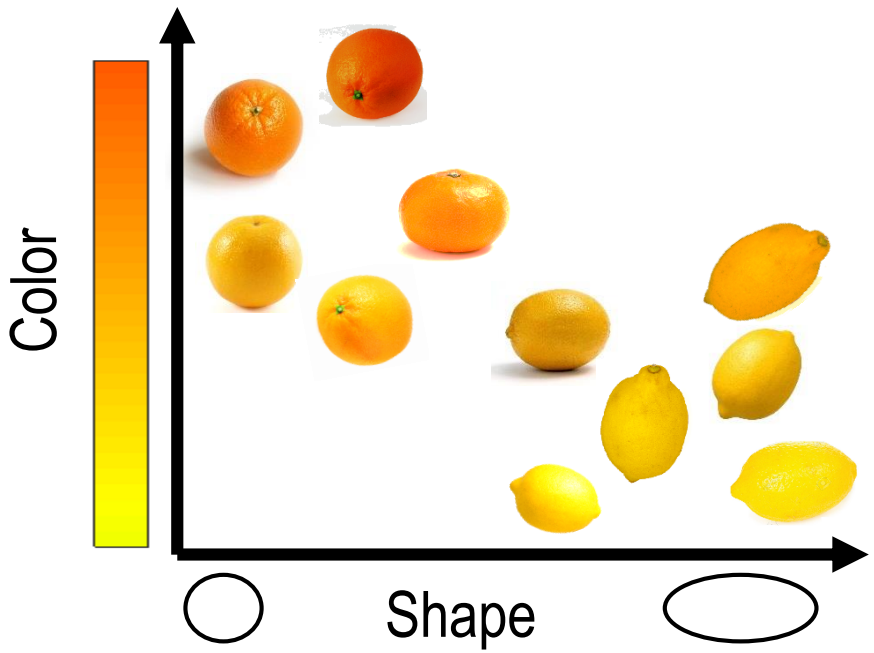
Color: orange
Shape: sphere
Ø: ± 8 cm
Weigth: ±0.1 kg

Color: yellow
Shape: elipsoid
Ø: ± 8 cm
Weigth: ±0.1 kg

* **Use "color" and "shape" as features**

Classification: example (2/5)

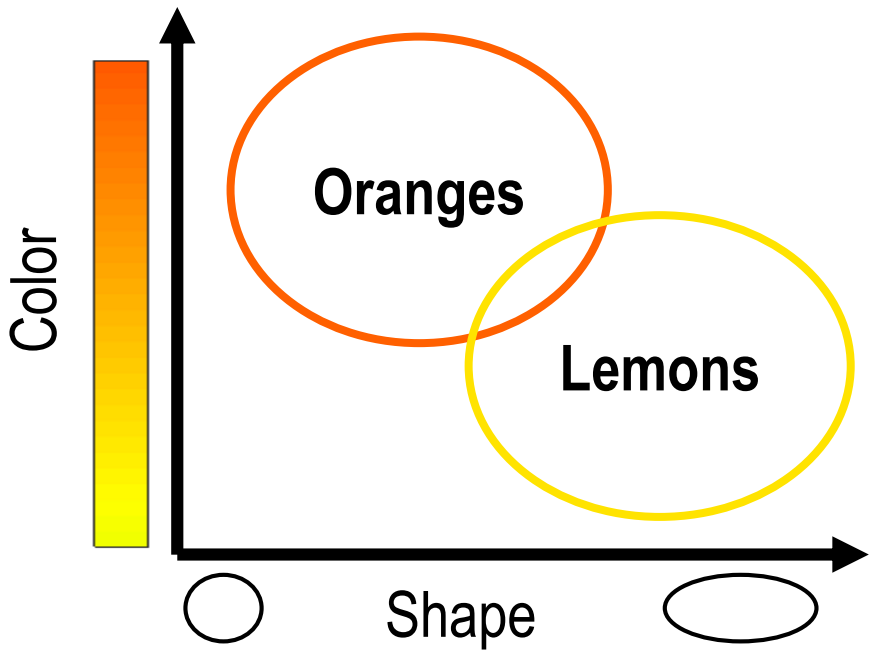
* **Separate lemons from oranges**



Classification: example (3/5)

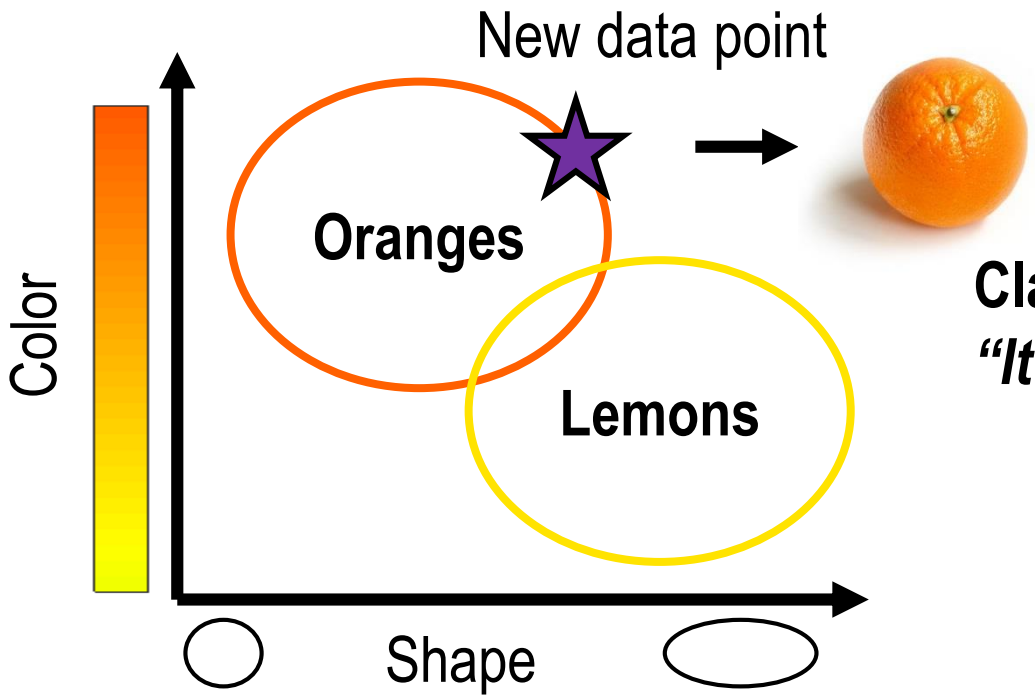
* **Separate lemons from oranges**

Model the given
- *training* - data



Classification: example (4/5)

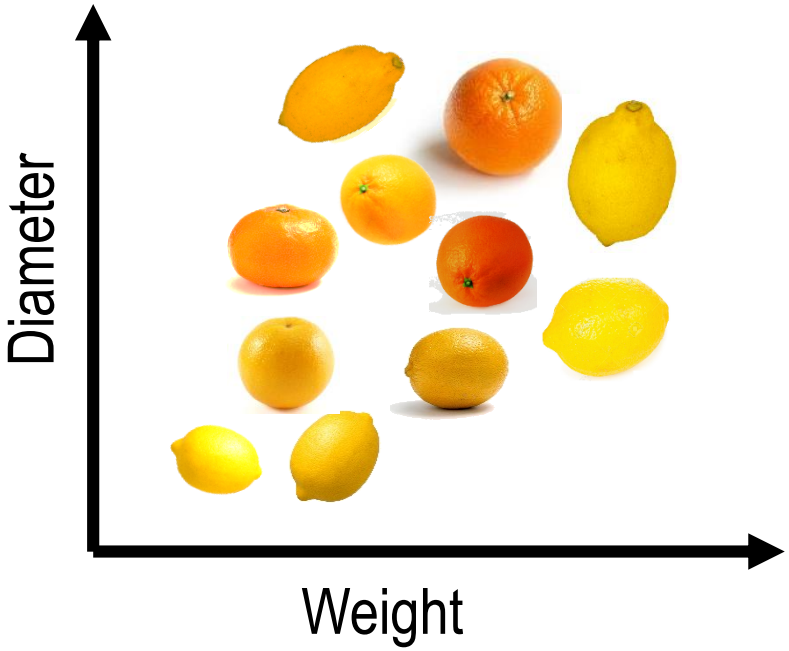
* **Separate lemons from oranges**

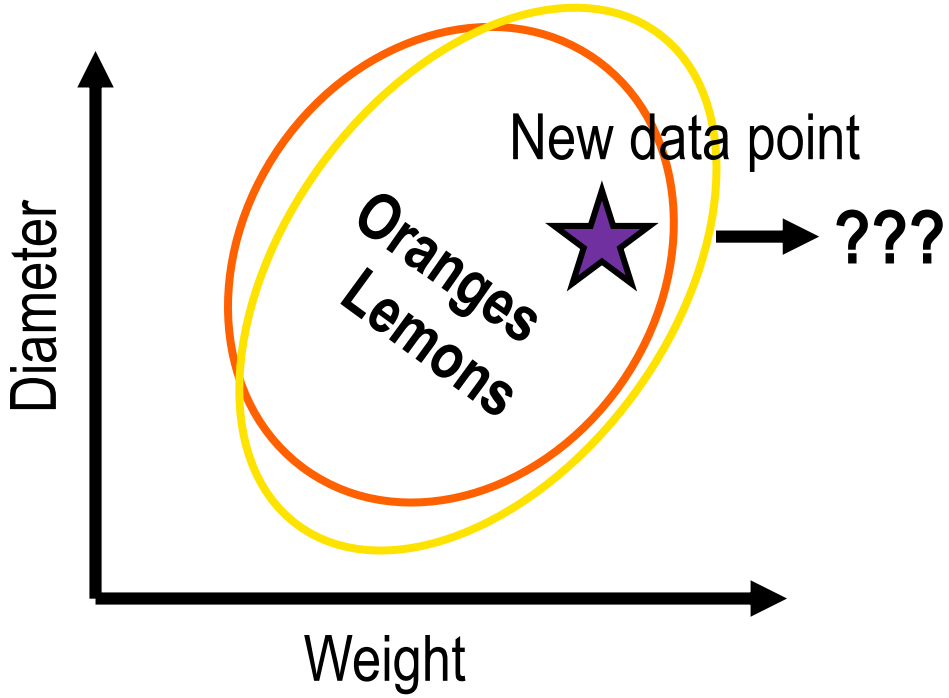


Classifier:
"It's an orange!"

Classification: example (5/5)

* **What if we had chosen the wrong features?**





Classification recap

Summary

* Choose distinctive features
* Make a model based on labeled data (a.k.a. supervised learning)
* Use the *learned* model to predict the class of new, unseen data points

Models for classification

* **Popular method: Random Forests**
* **Other classification methods:**
  - Support Vector Machines (SVM)
  - Neural Networks
  - Boosting
  - Naive Bayes
  - K Nearest Neighbours (k-NN)
  - Logistic regression

The following part of this lecture (slides 10-31) is largely based on a tutorial of Antonio Criminisi (slides 8-23, 30-35), download at:
http://research.microsoft.com/en-us/projects/decisionforests/

Advantages of random forests

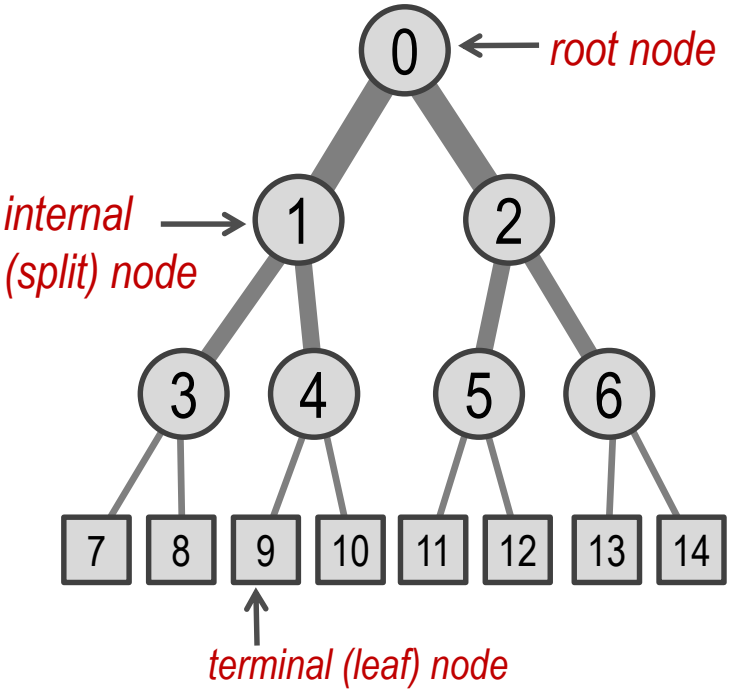
- ***General*** **model for machine learning**
  - Density estimation, regression, classification, …
- **Intrinsically probabilistic output**
  - Measure of confidence / uncertainty
- **Automatic feature selection**
- **Inherently multi-class**
- **Runs efficiently – trees can run in parallel**
- *Efficient implementations freely available…*

Decision trees

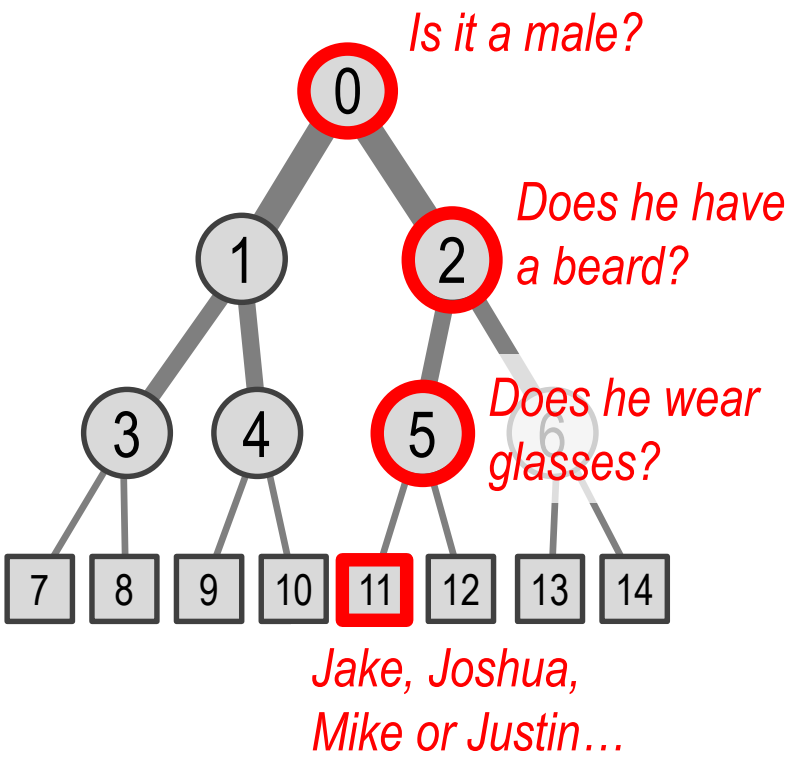
A forest consists of trees

- Start at the root node
- True/false question at each split node
- Stop when a leaf node is reached: prediction

A general tree structure



Decision trees

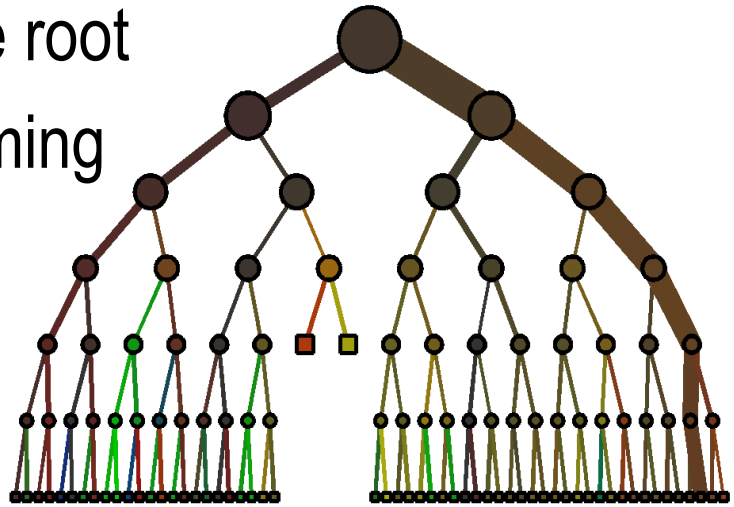


*Example: GUESS WHO**

**Credits to Mark Janse*

Decision trees

* **How to build a tree?**
  - Special type of graph: collection of nodes and edges
    - *Directed Acyclic Graph (DAG)*
  - Internal (split) nodes and terminal (leaf) nodes
  - The upper/start node is called the root
  - Each internal node has one incoming edge and two outgoing edges
  - All nodes (exept the root) have exactly one incoming edge

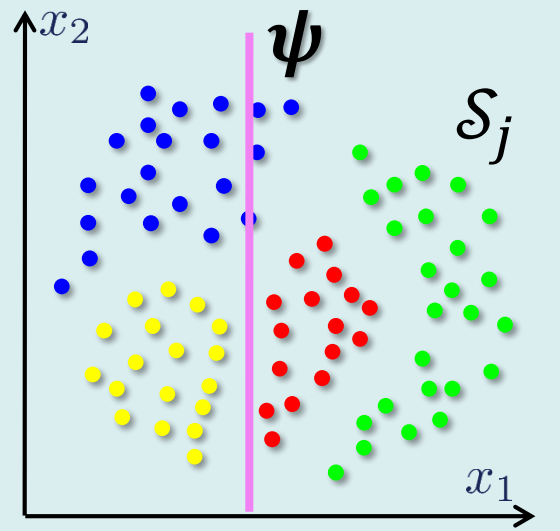


Decision trees

* **Mathematical notation**
  - Data point: $\mathrm{v} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$, label: y
    - Features: x_i, dimensionality: d
  - Binary split function: $h(\mathbf{v}, \boldsymbol{\theta}_j)$: $\mathbb{R}^d \times \mathcal{T} \rightarrow \{0,1\}$
    - Split parameters of node j: $\boldsymbol{\theta}_j \in \mathcal{T}$
    - Set of possible parameters $\mathcal{T}$
  - Training points reaching node j: $\mathcal{S}_j = \mathcal{S}_j^L \cup \mathcal{S}_j^R$ (Left, Right)
    - Complete training set $\mathcal{S}_0$

Decision trees

∗ How to split the data?



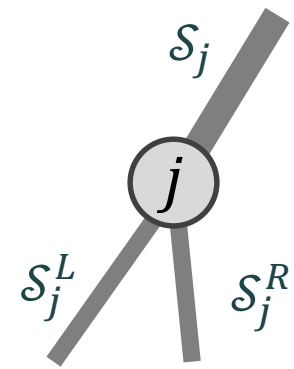
Axis aligned hyperplane

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

For 2D example: $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \quad x_2 \quad 1)^T$

e.g. $\boldsymbol{\psi} = (1 \quad 0 \quad \psi_3)$

Split function $h(\mathbf{v}, \boldsymbol{\theta})$ depends on parameters $\boldsymbol{\theta} = \{\boldsymbol{\phi}(\mathbf{v}), \boldsymbol{\psi}, \boldsymbol{\tau}\}$

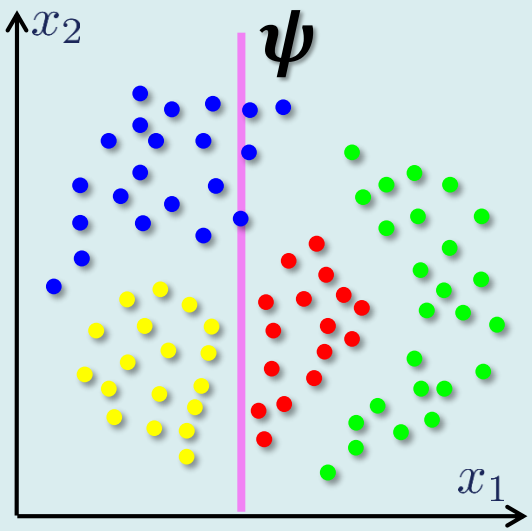


- Feature selection function $\boldsymbol{\phi}(\mathbf{v})$
- Geometric primitive $\boldsymbol{\psi}$ (e.g. a line)
- Thresholds $\boldsymbol{\tau}$
  - *Note that setting either $\tau_1 = \infty$ or $\tau_2 = -\infty$ corresponds to using only one threshold.*

Parameter space $\mathcal{T}$ contains the options that we have for parameters $\boldsymbol{\phi}(\mathbf{v})$, $\boldsymbol{\psi}$ and $\boldsymbol{\tau}$.

Decision trees

* **How to split the data?**

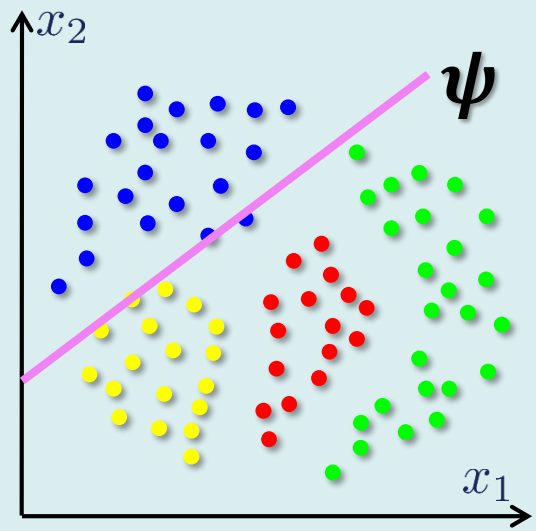


Axis aligned hyperplane

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

For 2D example: $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \quad x_2 \quad 1)^T$

e.g. $\boldsymbol{\psi} = (1 \quad 0 \quad \psi_3)$

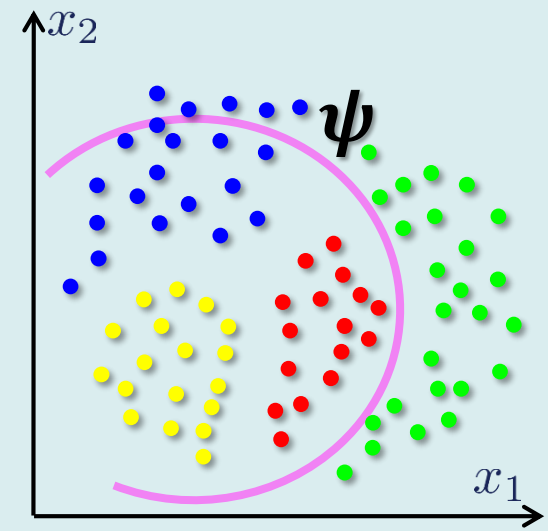


Oriented hyperplane

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

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

$\boldsymbol{\psi} \in \mathbb{R}^3$, *e.g.* $\boldsymbol{\psi} = (3 \quad -2 \quad 4)$



Quadratic surface

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

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

$\boldsymbol{\psi} \in \mathbb{R}^{3\times3}$ representing a conic

Decision trees

* **How to determine the best split?**
  - Maximize information gain*:

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

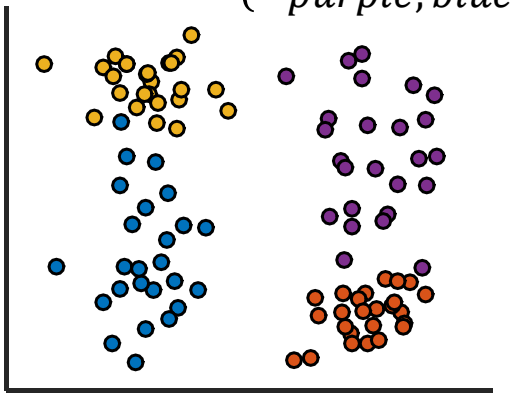
    - Shannon's entropy:

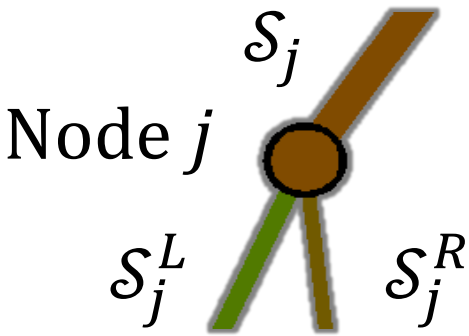
$$H(\mathcal{S}^i) = -\sum_{c\in\mathcal{C}} p(c)\log(p(c))$$

  - Node training

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

$$\mathcal{C} = \begin{Bmatrix} yellow, orange, \\ purple, blue \end{Bmatrix}$$

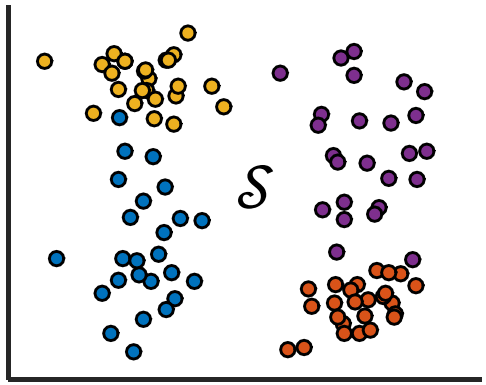


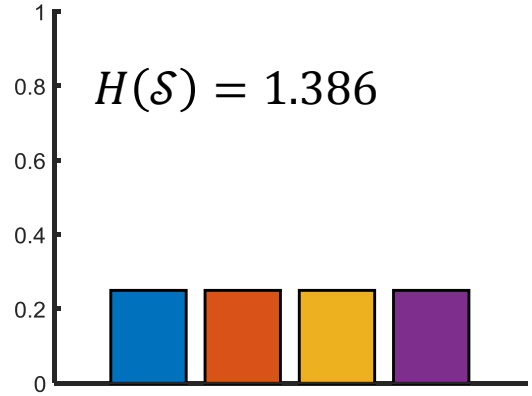


*One of many options. Other popular choices:
(1) Gini's diversity index, (2) Misclassification error

Decision trees

* **What is the best split?**

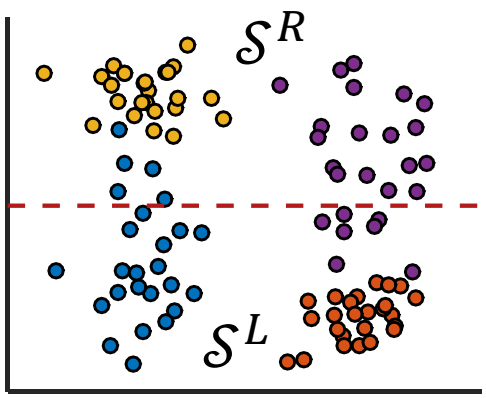


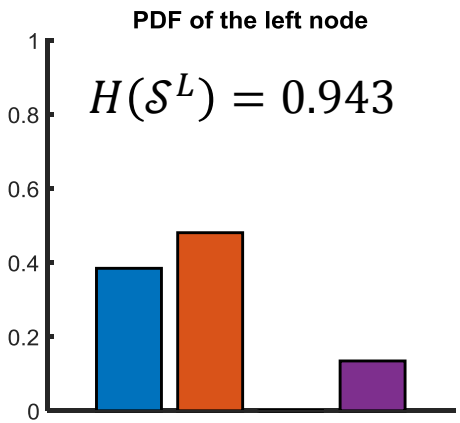


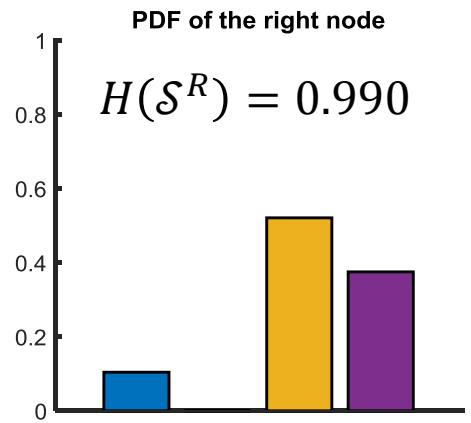
Information gain

$I(\mathcal{S}, \theta) = 0.419$

$|\mathcal{S}^L| = 48,\ |\mathcal{S}^R| = 52$

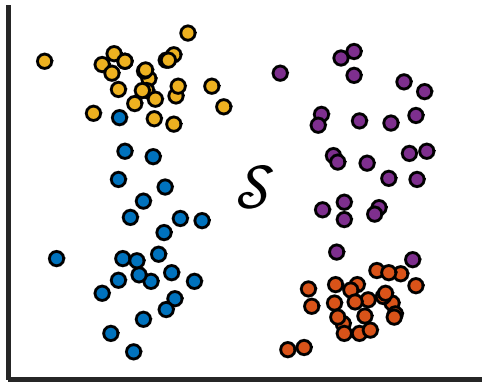


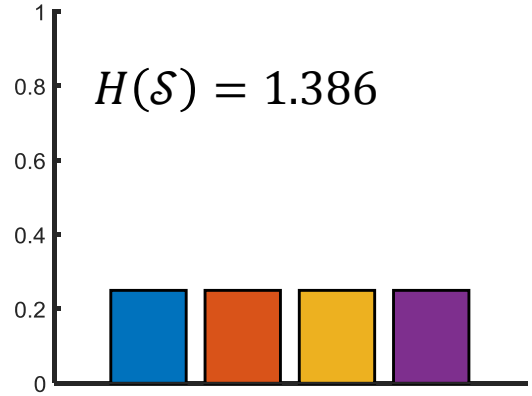




Decision trees

* **What is the best split?**

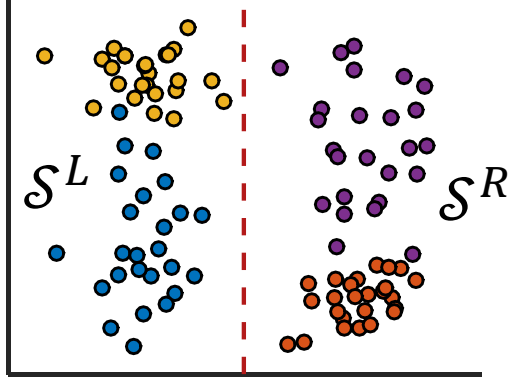


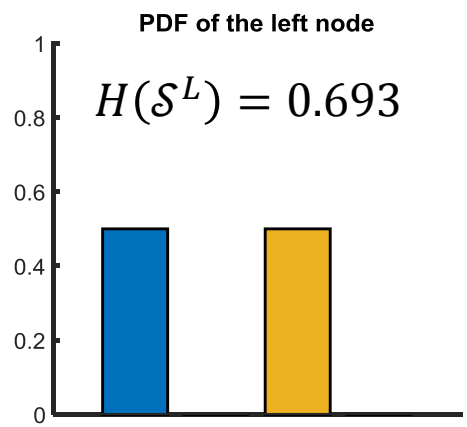


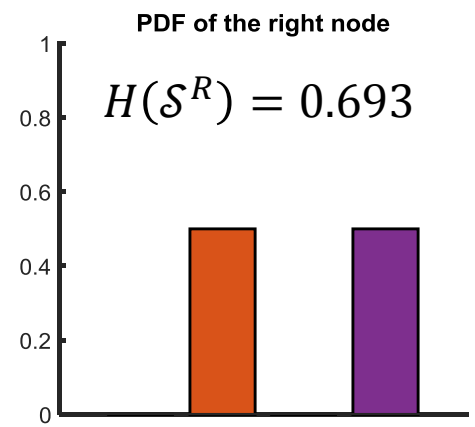
Information gain

$I(\mathcal{S}, \theta) = 0.693$

$|\mathcal{S}^L| = 50, \ |\mathcal{S}^R| = 50$

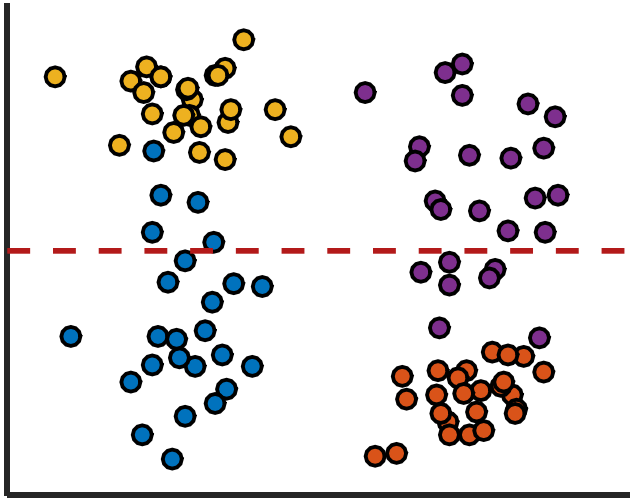






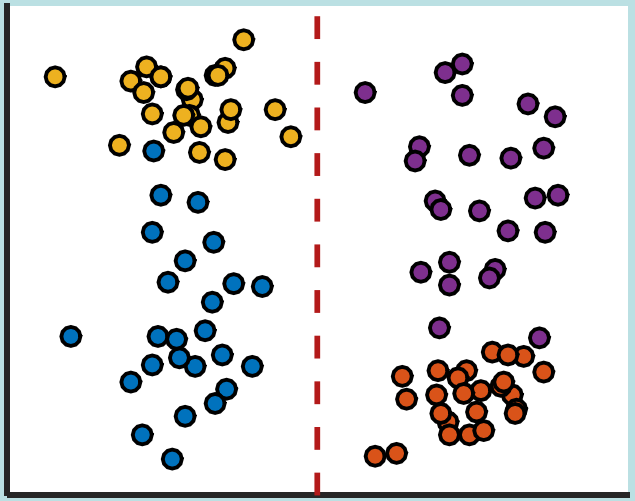
Decision trees

* **What is the best split?**



$I(\mathcal{S}, \theta) = 0.419$

BEST SPLIT !



$I(\mathcal{S}, \theta) = 0.693$

.. of these two options

Decision trees

* **What is the best split?**
  - ➢ The one that yields the highest information gain…

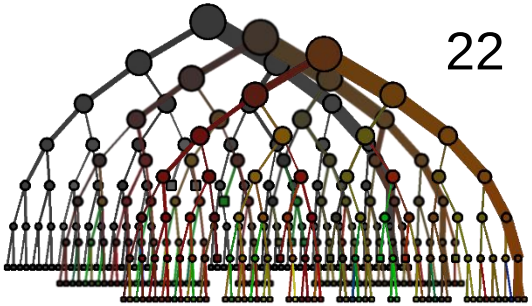
    … from a given set of "candidate splits"

* **Node training**

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

  - Split function parameters $\boldsymbol{\theta}$
  - Limited set of parameter settings $\mathcal{T}_j$

Decision forest model



Node test parameters		$\boldsymbol{\theta} \in \mathcal{T}$	*Features / split function / thresholds*
Node objective function	*e.g.*	$I = I(\mathcal{S}_j, \theta)$	*(Energy) function to minimize*
Node weak learner	*e.g.*	$h(\mathbf{v}, \boldsymbol{\theta}_j) \in \{true, false\}$	*Split node test function*
Leaf predictor model	*e.g.*	$p(c\|\mathbf{v})$	*Point estimate / full distribution*
Randomness model	*e.g.*	Bagging, RNO	*Methods for inserting randomness*
Stopping criteria	*e.g.*	Max tree drepth D	*When to stop splitting the data*
Forest size		T	*Number of trees in the forest*
Ensemble model	*e.g.*	$p(c\|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(c\|\mathbf{v})$	*How to combine the output of all the trees in the forest*

Decision forest model

* **How to add randomness?** *(Randomness model)*

1. **Bagging (randomized training set)**
   - Subset of all data points per tree
2. **Randomized Node Optimization (RNO)**
   - Features *chosen with selection function* $\boldsymbol{\phi}(\boldsymbol{v})$
   - Split function *depending on weak learner orientation* $\boldsymbol{\psi}$
   - Thresholds *given in* $\boldsymbol{\tau}$

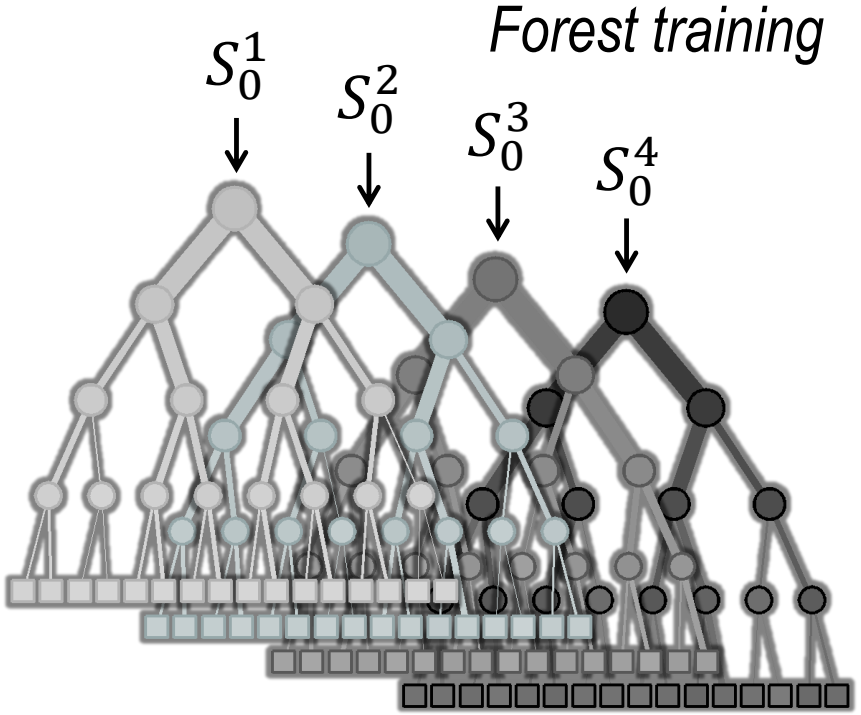
Decision forest model

∗ How to add randomness?

➢ **(1) Bagging**

S_0: Full training set

$S_0^t \subset S_0$: Randomly sampled subset for training tree t



Decision forest model

Node test parameter
$\boldsymbol{\theta} = \{\boldsymbol{\phi}(\mathbf{v}), \boldsymbol{\psi}, \boldsymbol{\tau}\} \in \mathcal{T}$

∗ How to add randomness?

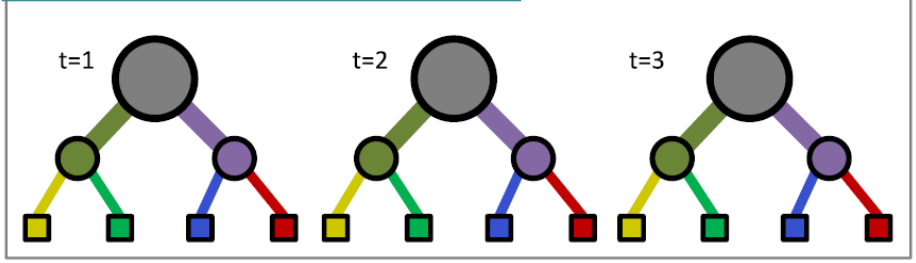
➢ **(2) Randomized Node Optimization (RNO)**

$\mathcal{T}$: Full set of all possible node test parameter values

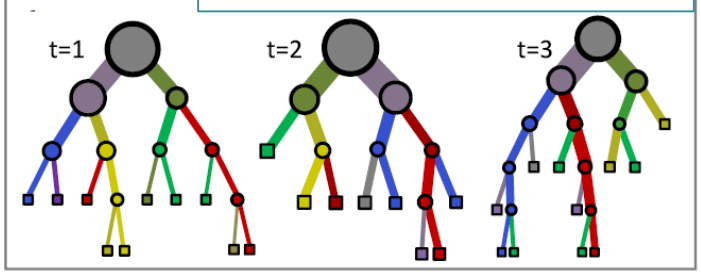
$\mathcal{T}_j \subset \mathcal{T}$: Set of randomly sampled parameter values to train node j

$\rho = |\mathcal{T}_j|$: Randomness control parameter

$\rho \approx |\mathcal{T}|$, *low randomness*



$\rho = 1$, *high randomness*

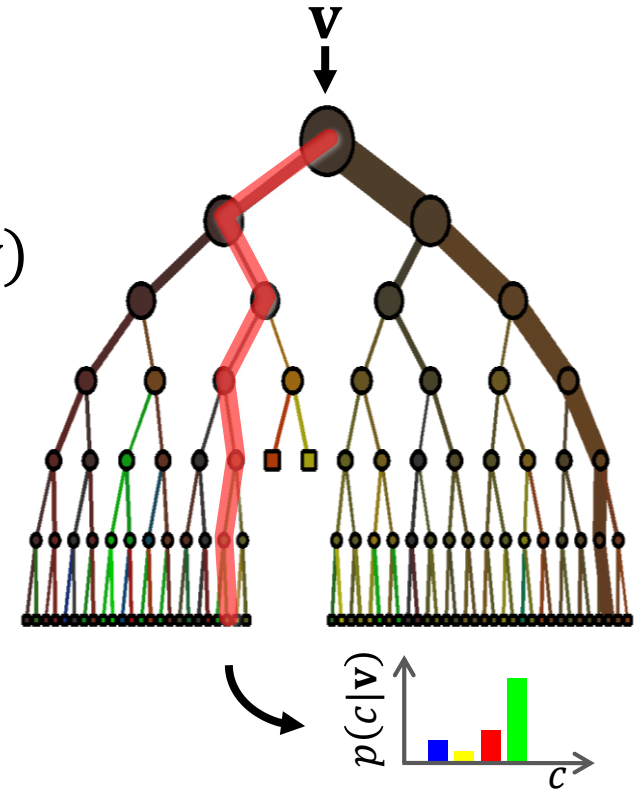


Decision forest model

* **How to compute a prediction from a trained tree?**
  - Probability distribution at leaf: $p(c|\mathbf{v})$
  - Point-estimate, e.g. M.A.P.:

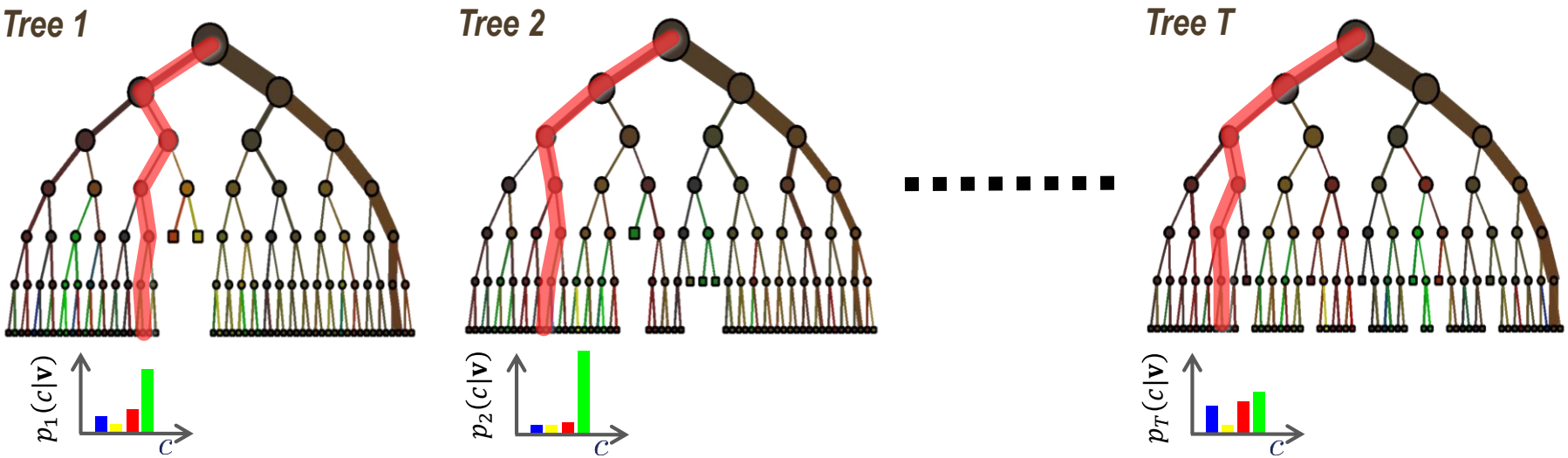
$$c^* = \arg\max_c p(c|\mathbf{v})$$

Generally the full distribution is perserved untill the decision moment to incoroporate uncertainty



Decision forest model

* How to combine tree output?



- **Averaging:** $p(c|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(c|\mathbf{v})$
- **Multiplication:** $p(c|\mathbf{v}) = \frac{1}{Z}\prod_{t=1}^{T} p_t(c|\mathbf{v})$

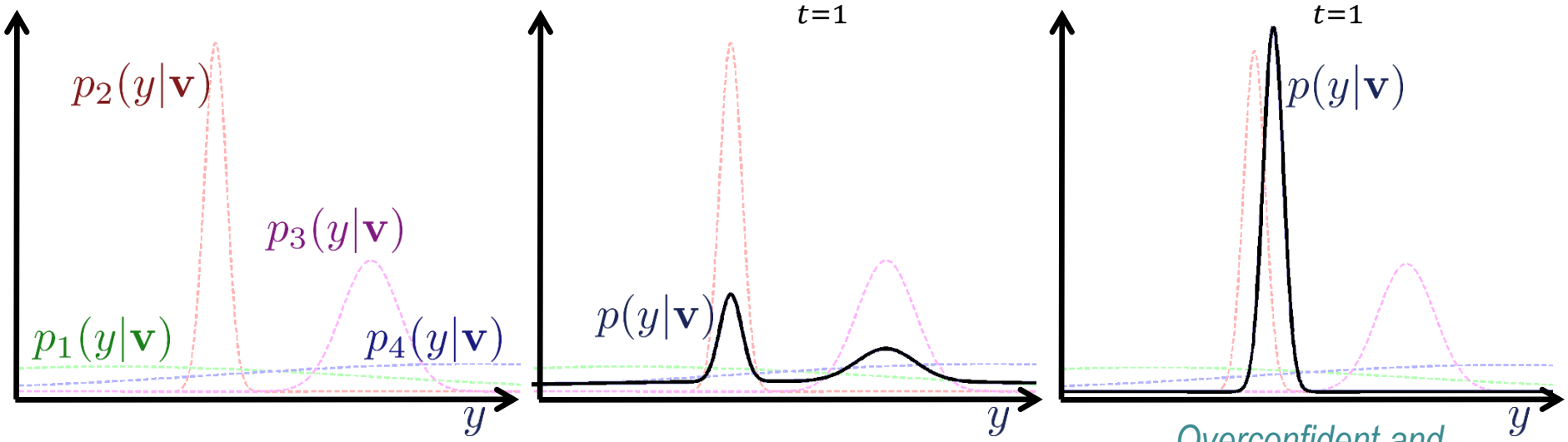
where Z is a partitioning function to ensure probabilistic normalization

Decision forest model

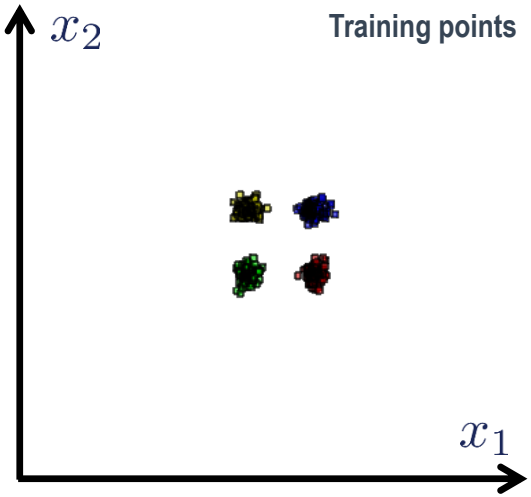
* How to combine tree output?

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

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

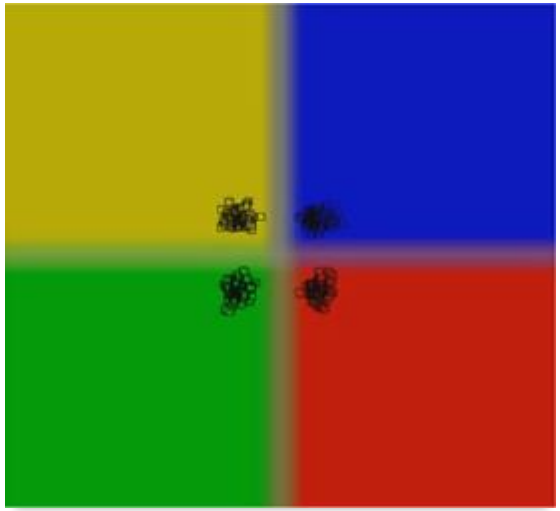


Overconfident and less robust to noise

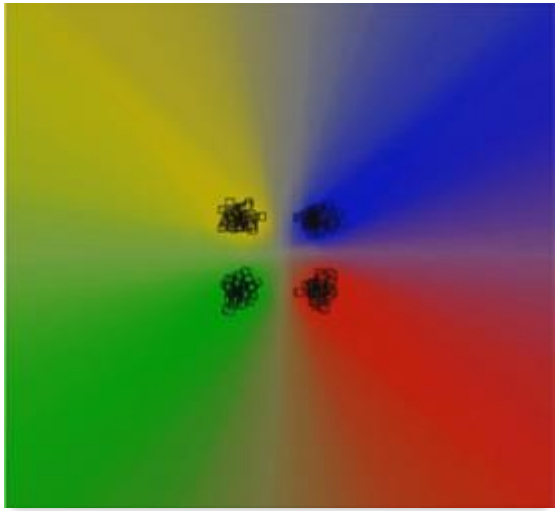


Example: generalization

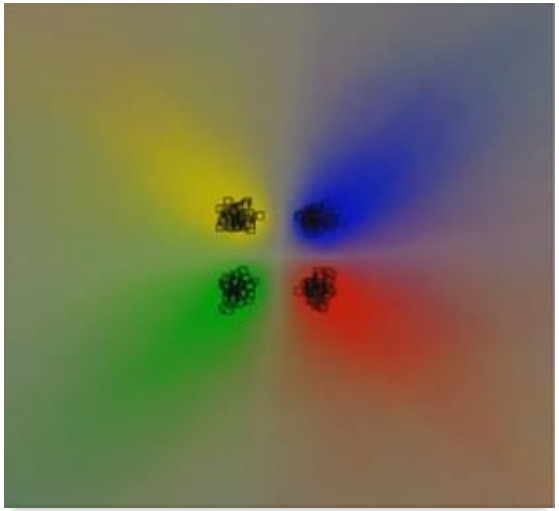
Weak learner: Axis aligned



Weak learner: Oriented line

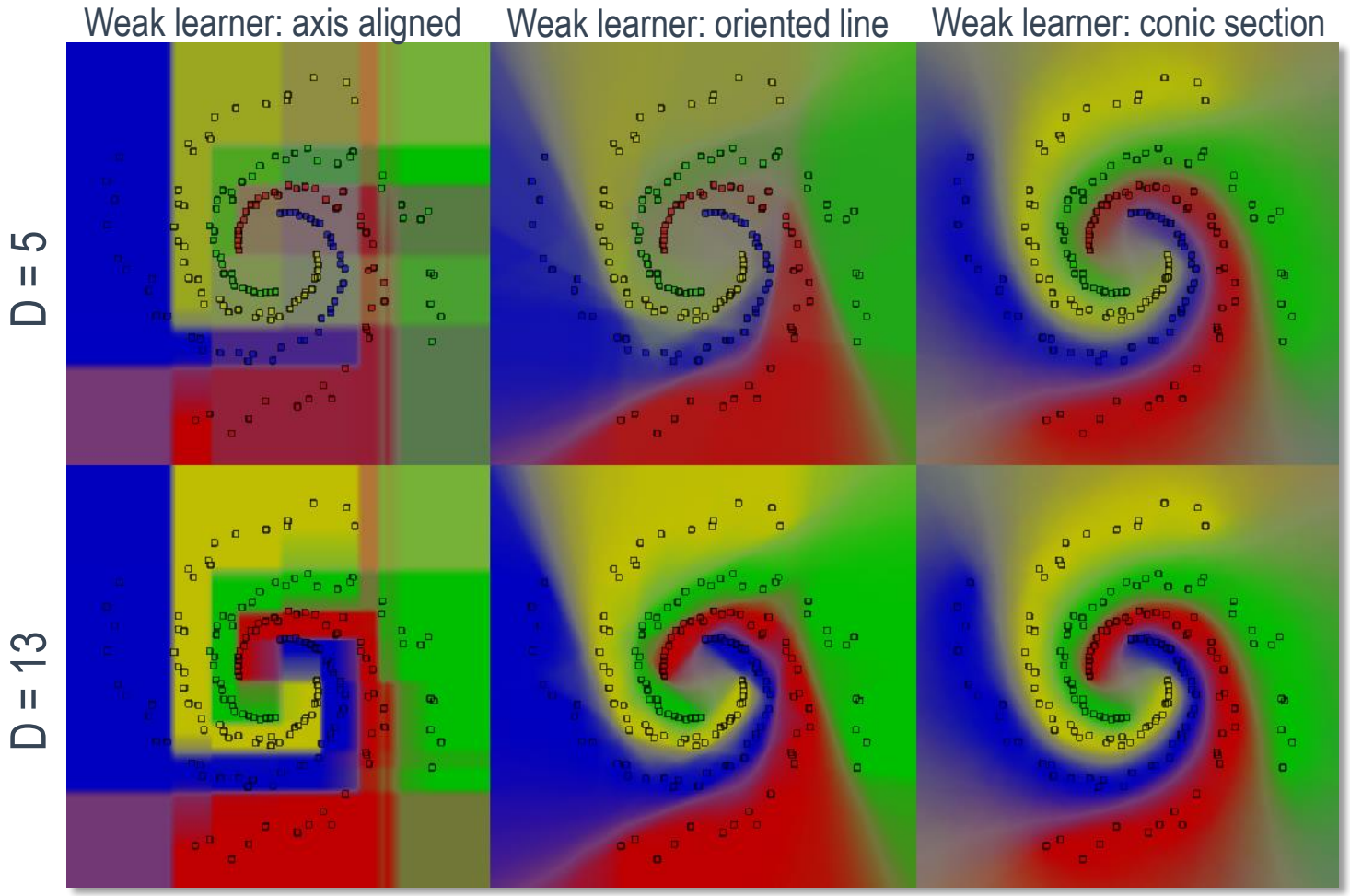


Weak learner: Conic section



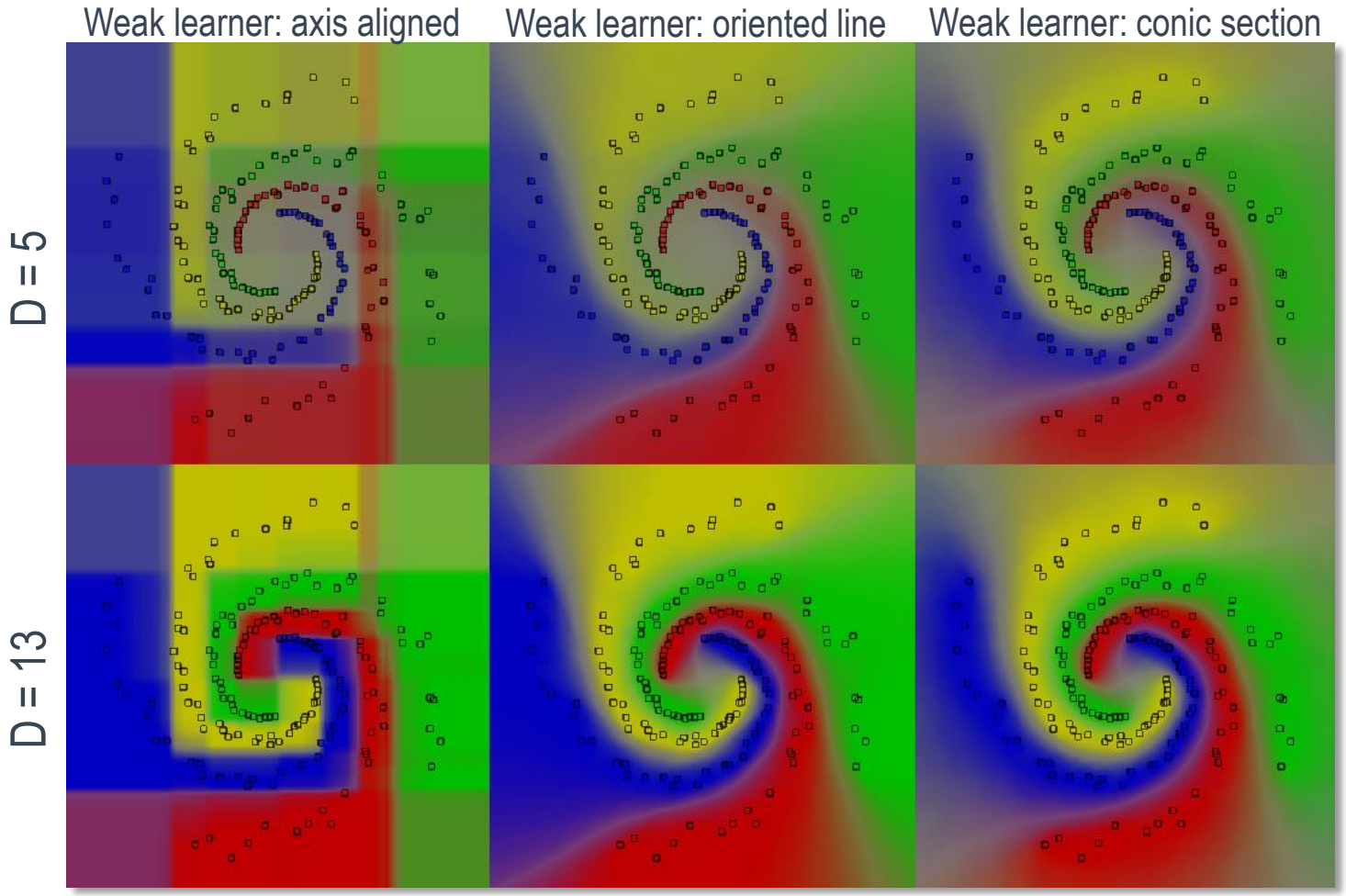
Example: the effect of randomness

$\rho = 500$



Example: the effect of randomness

$\rho = 50$



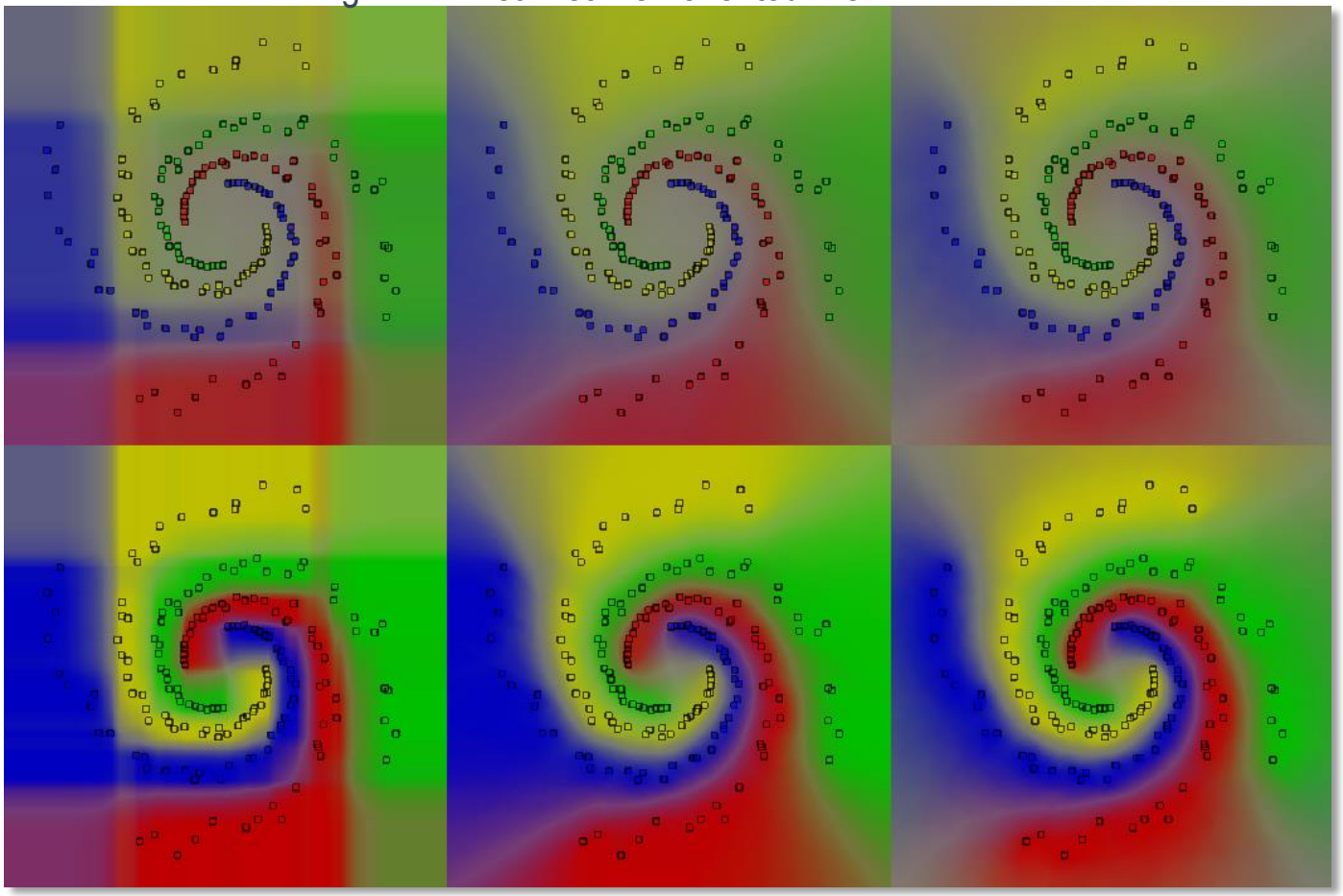
Example: the effect of randomness

$\rho = 5$

Weak learner: axis aligned | Weak learner: oriented line | Weak learner: conic section

D = 5

D = 13



Example: handwritten digit classification

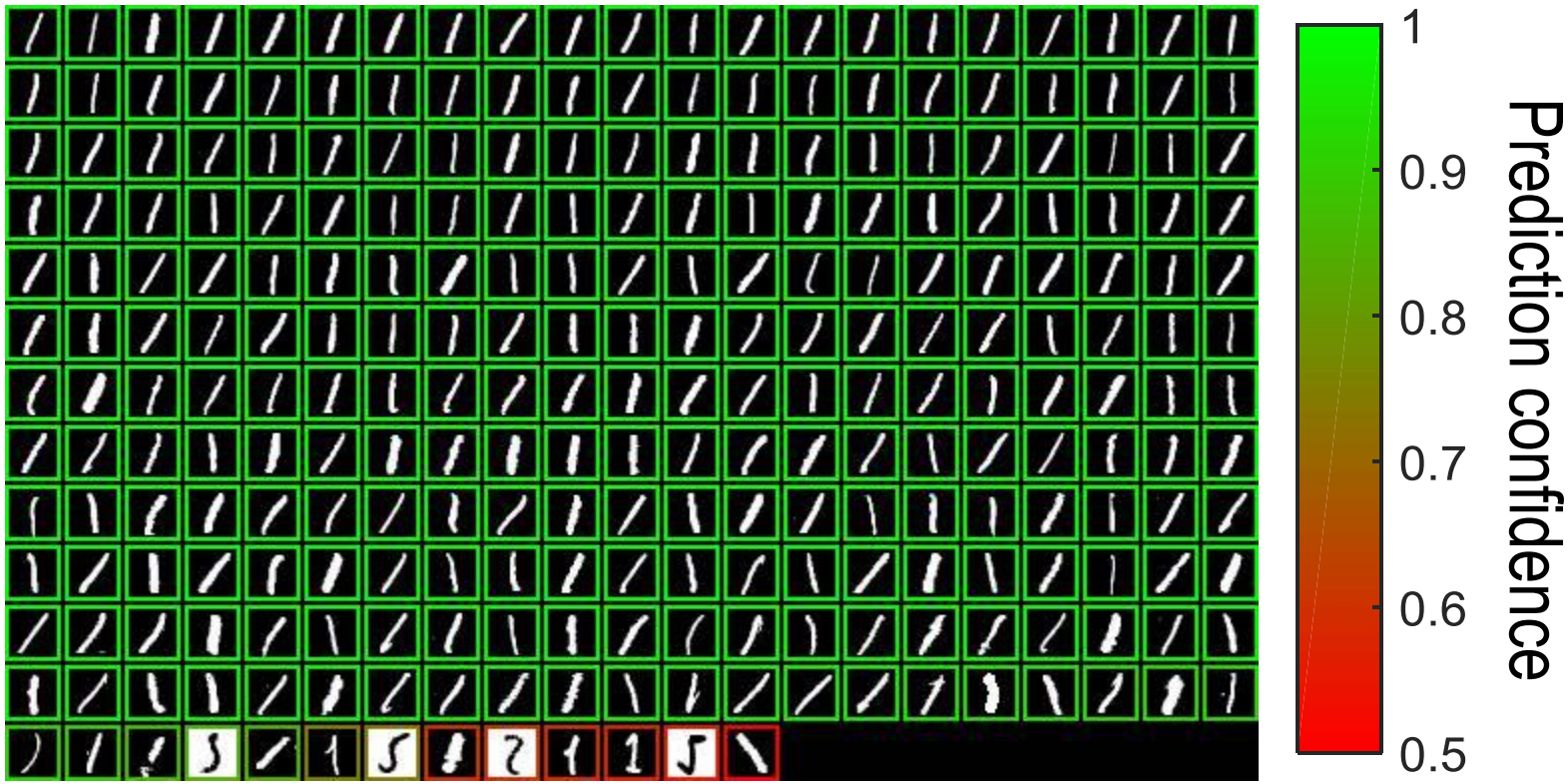
"The MNIST Database of Handwritten Digit Images for Machine Learning Research", DOI:10.1109/MSP.2012.2211477.

Example: handwritten digit classification

* **Task: classify handwritten digits 1, 2, 3, 4, 5**
  - 500 samples per digit: 250 for training, 250 for testing
  - HOG* features as data points $\mathbf{v} \in \mathbb{R}^d$, $d = 144$
  - Forest parameters:
    - Forest size $T = 300$ trees
    - Axis aligned weak learner $\boldsymbol{\psi}$
    - Randomness parameter $\rho = 5$, with $|\mathcal{T}| = 10$
    - Selection functin $\boldsymbol{\phi}(\mathbf{v})$ randomly samples $\sqrt{d}$ dimensions from $\mathbf{v}$

Histogram of Oriented Gradients, Dalal & Triggs, CVPR 2005

Example: handwritten digit classification



Precision = 0.96 / Recall = 0.97

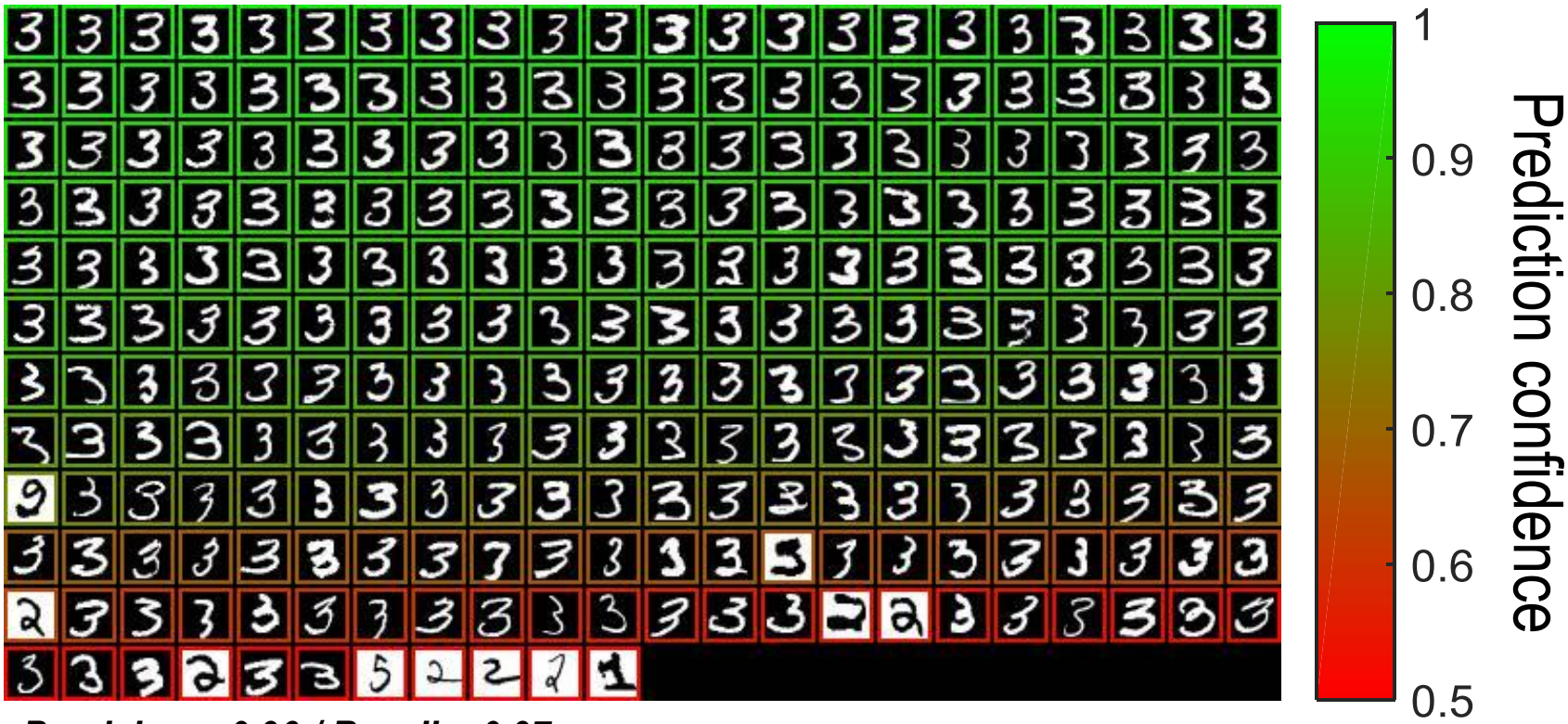
Forest predictions for class '1', sorted by confidence. Inverted digits are wrongly classified.

Example: handwritten digit classification

Precision = 0.99 / Recall = 0.92

Forest predictions for class '2', sorted by confidence. Inverted digits are wrongly classified.

Example: handwritten digit classification



Precision = 0.96 / Recall = 0.97

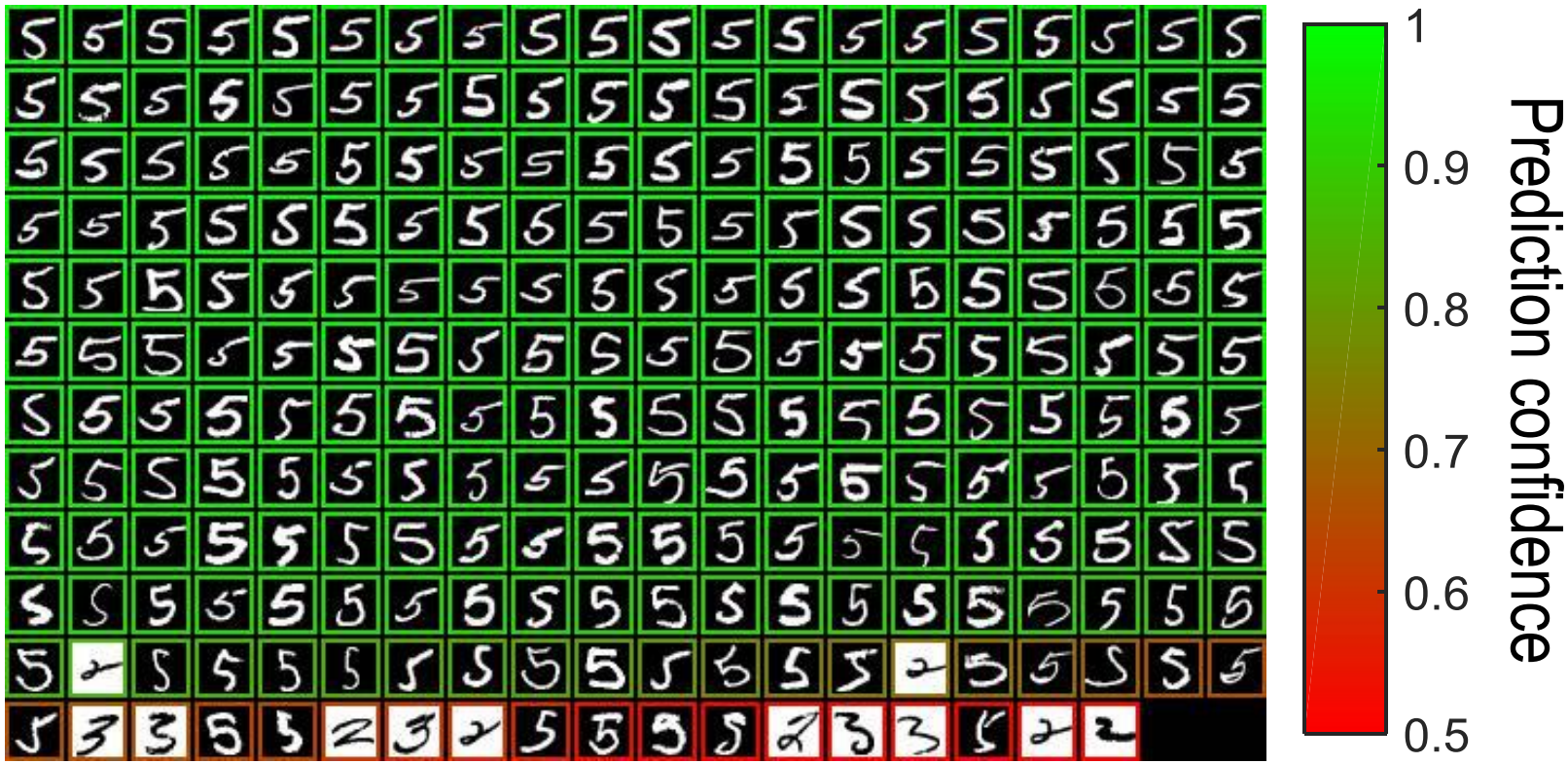
Forest predictions for class '3', sorted by confidence. Inverted digits are wrongly classified.

Example: handwritten digit classification

Precision = 0.97 / Recall = 1.00

Forest predictions for class '4', sorted by confidence. Inverted digits are wrongly classified.

Example: handwritten digit classification

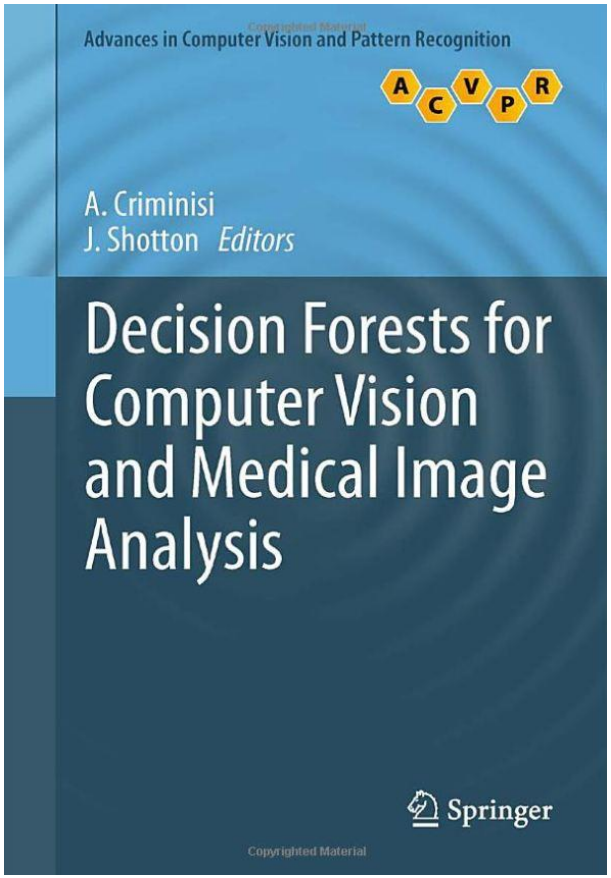


Precision = 0.95 / Recall = 0.97

Forest predictions for class '5', sorted by confidence. Inverted digits are wrongly classified.

Recommended literature

* "*Decision Forests for Computer Vision and Medical Image Analysis*", A. Criminisi, 2013
* Ch.3: Introduction
* Ch.4: Classification Forests
* C++ library: Sherwood
  - http://research.microsoft.com/en-us/projects/decisionforests/
* *Breiman L., "Random forests", Mach. Learn. 45(1), 2001.*
  doi:10.1023/A:1010933404324



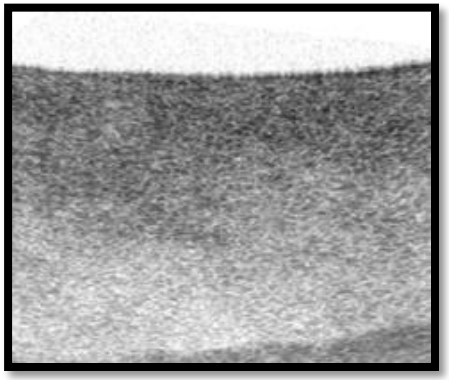
doi:10.1007/978-1-4471-4929-3

Conclusions

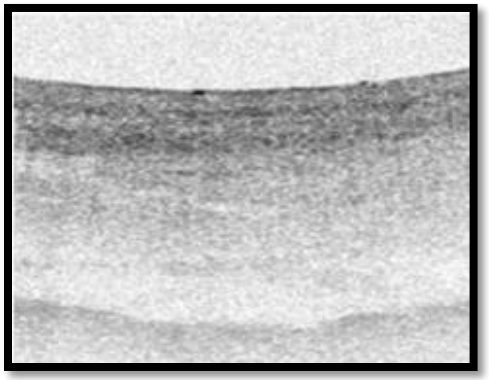
- **Random Forests offer an attractive method classification**
  - Inherently multi-class, probablistic output, efficient implementations available..
- **A forest is a collection of decision trees**
  - Each tree t is trained with a different subset $\mathcal{S}_0^t$ of the training data (**Bagging**)
- **A tree is a collection of nodes and edges**
  - Each internal node splits the incoming data using node split function $h(\mathbf{v}, \boldsymbol{\theta}_j)$
    - *$\boldsymbol{\theta}$ encompasses selection function $\boldsymbol{\phi}(\mathbf{v})$, geometric primitive $\boldsymbol{\psi}$ and thresholds $\boldsymbol{\tau}$*
    - *Each node j receives a random subset $\mathcal{T}_j$ of the parameter space $\mathcal{T}$ for training (**RNO**)*
- **Randomness increases robustness**
  - Randomness control parameter $\boldsymbol{\rho}$ determines the ammount of randomness
    - *Maximum randomness when $\boldsymbol{\rho} = \mathbf{1}$, minumum randomness when $\boldsymbol{\rho} = |\boldsymbol{\mathcal{T}}|$*
  - *Tree depth D controls the forest confidence, hence a high D can lead to overfitting*

Internship / grad. project

- **Computer-aided diagnosis of OCT* images**
  - Pilot study
  - 60 OCT images: 30 cancer / 30 no cancer
  - Frontline research



cancer



no cancer

*Optical Coherence Tomography

Interested? Contact: ir. Fons van der Sommen (f.v.d.sommen@tue.nl)