Decision Forests for Classication, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning

A. Criminisi[1], J. Shotton[2] and E. Konukoglu[3]

[1] Microsoft Research Ltd, 7 J J Thomson Ave, Cambridge, CB3 0FB, UK
[2] Microsoft Research Ltd, 7 J J Thomson Ave, Cambridge, CB3 0FB, UK
[3] Microsoft Research Ltd, 7 J J Thomson Ave, Cambridge, CB3 0FB, UK

Josu Maiora

Outline

1. Overview and scope
2. The random decision forest model
   - 2.1 Background and notation
   - 2.2 The decision forest model
3. Classification forest.
   - 3.1 Classification algorithms in the literature
   - 3.2 Specializing the decision forest model for classification
   - 3.3 Effect of model parameters
   - 3.4 Maximum-margin properties
   - 3.5 Comparisons with alternative algorithms
   - 3.6 Human body tracking in Microsoft Kinect for XBox 360
4. Regression forests.
   - 4.1 Nonlinear regression in the literature
   - 4.2 Specializing the decision forest model for regression
   - 4.3 Effect of model parameters
   - 4.4 Comparison with alternative algorithms
   - 4.5 Semantic parsing of 3D computed tomography scans

1. Overview and scope

- Unified, efficient model of random decision forests
  - Applications
    - Scene recognition from photographs,
    - Object recognition in images,
    - Automatic diagnosis from radiological scans
    - Semantic text parsing.

- A brief literature survey
  - Breinman
  - “C4.5” of Quinlan
  - In this early work trees are used as individual entities. However, recently it has emerged how using an ensemble of learners (e.g. weak classiffiers) yields greater accuracy and generalization.[1]

1. Overview and scope

- A random decision forest is an ensemble of randomly trained decision trees.
- Ensemble methods became popular with the face and pedestrian detection papers of Viola and Jones
- Decision forests compare favourably with respect to other techniques
- One of the biggest success stories of computer vision in recent years → the Microsoft Kinect for XBox 360.

2. The random decision forest model

- Problems can be categorized into a relatively small set of prototypical machine learning tasks.
  - Recognizing the type of a scene captured in a photograph can be cast as **classification.**
  - Predicting the price of a house as a function of its distance from a good school may be cast as a **regression** problem.
  - Detecting abnormalities in a medical scan can be achieved by evaluating the scan under a learned probability **density function** for scans of healthy individuals.
  - Capturing the intrinsic variability of size and shape of patients brains in magnetic resonance images may be cast as **manifold learning**.

2. The random decision forest model

- Interactive image segmentation may be cast as a **semi supervised problem**, where the user's brush strokes define labeled data and the rest of image pixels provide already available unlabelled data.

- Learning a general rule for detecting tumors in images using minimal amount of manual annotations is an **active learning** task, where expensive expert annotations can be optimally acquired in the most economical fashion.

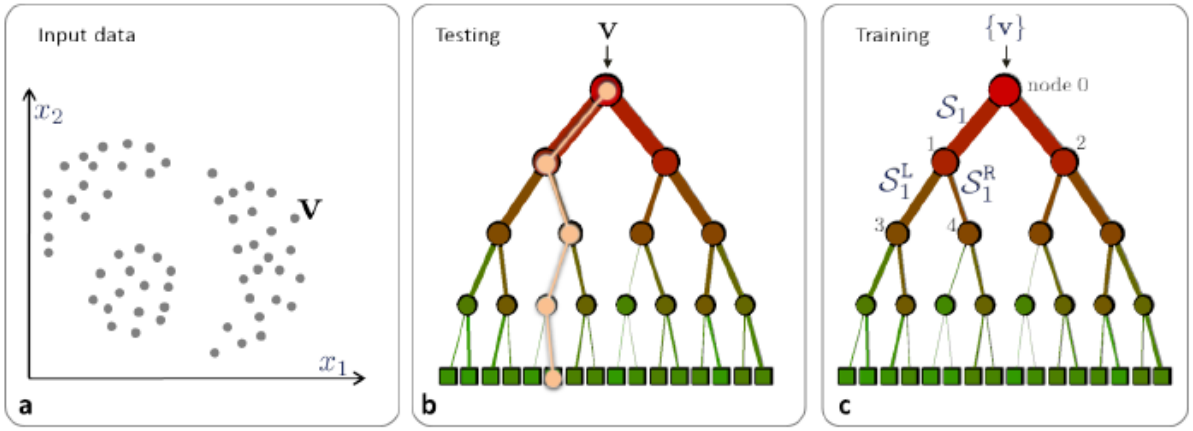
2.The random decision forest model.

2.1 Background and notation

- Their recent revival is due to the discovery that ensembles of slightly different trees tend to produce much higher accuracy on previously unseen data, a phenomenon known as generalization
- A tree is a collection of nodes and edges organized in a hierarchical structure . Nodes are divided into internal (or split) nodes and terminal (or leaf) nodes.
- Mathematical notation
  - vector v = (x_1; x_2; ; x_d) ∈ R_d. x_i represent some scalar feature responses.
  - feature dimensionality *d*
  - Function ϕ (v) selecting a subset of features of interest.

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}, \text{ with } d' << d.$$

- Training and testing decision trees
  - At a high level, the functioning of decision trees can be separated into an off-line phase (training) and an on-line one (testing).



  - Given a training set S0 of data points {v} and the associated ground truth labels the tree parameters are chosen so as to minimize a chosen energy function
  - randomness is only injected during the training process, with testing being completely deterministic once the trees are fixed.

❑ Entropy and information gain

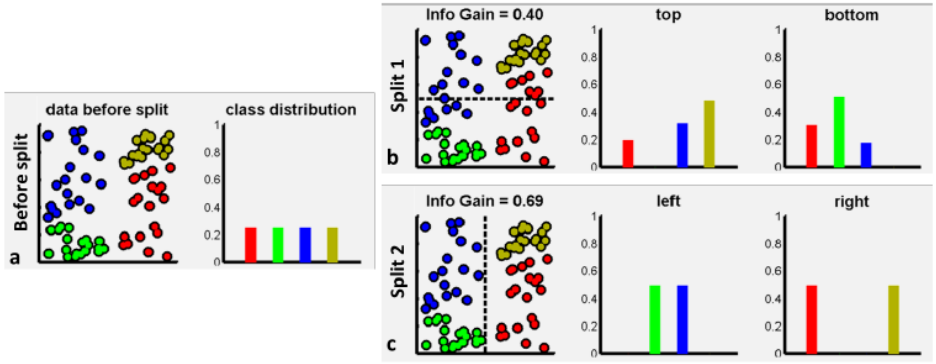


Fig. 2.3: **Information gain for discrete, non-parametric distributions.** (a) Dataset $\mathcal{S}$ before a split. (b) After a horizontal split. (c) After a vertical split.

The gain of information achieved by splitting the data is computed as

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

with the Shannon entropy defined mathematically as: $H(\mathcal{S}) = -\sum_{c \in \mathcal{C}} p(c) \log(p(c))$. In our example a horizontal split does not sep-

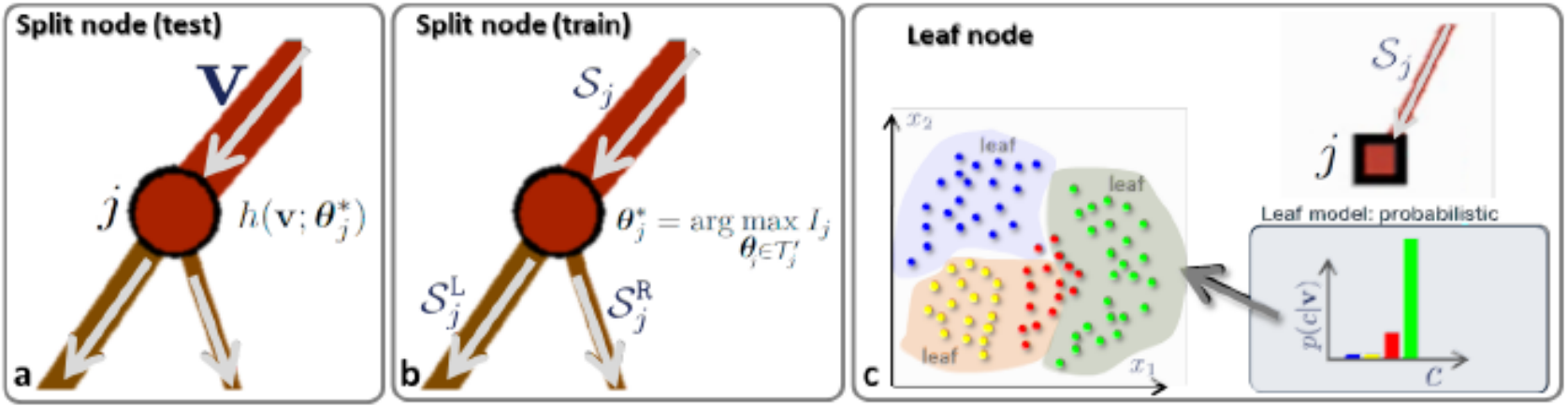


Fig. 2.5: **Split and leaf nodes.** (a) Split node (testing). A split node is associated with a weak learner (or split function, or test function). (b) Split node (training). Training the parameters $\boldsymbol{\theta}_j$ of node j involves optimizing a chosen objective function (maximizing the information gain I_j in this example). (c) A leaf node is associated with a predictor model. For example, in classification we may wish to estimate the conditional $p(c|\mathbf{v})$ with $c \in \{c_k\}$ indicating a class index.

- The weak learner model: linear or non-linear data separation

Each split node j is associated with a binary split function

$$h(\mathbf{v}, \boldsymbol{\theta}_j) \in \{0, 1\},$$

$\boldsymbol{\theta} = (\phi, \psi, \tau)$ where ψ defines the geometric primitive

f captures thresholds for the inequalities used in the binary test.

The filter function ϕ selects some features of choice out of the entire vector v.

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

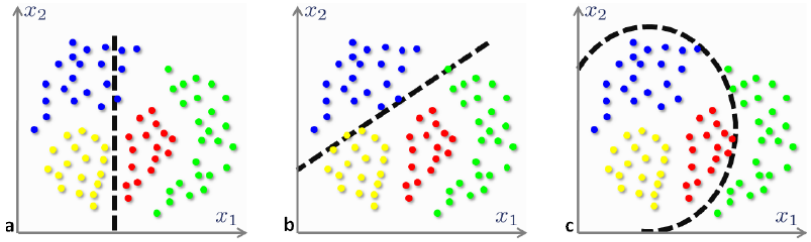


Fig. 2.6: **Example weak learners.** **(a)** Axis-aligned hyperplane. **(b)** General oriented hyperplane. **(c)** Quadratic (conic in 2D). For ease of visualization here we have $\mathbf{v} = (x_1\ x_2) \in \mathbb{R}^2$ and $\boldsymbol{\phi}(\mathbf{v}) = (x_1\ x_2\ 1)$ in homogeneous coordinates. In general data points $\mathbf{v}$ may have a much higher dimensionality and $\boldsymbol{\phi}$ still a dimensionality of ≤ 2.

- The training objective function

$$\boldsymbol{\theta}_j^* = \arg\max_{\boldsymbol{\theta}_j} I_j$$

with

$$I_j = I(\mathcal{S}_j, \mathcal{S}_j^{\mathrm{L}}, \mathcal{S}_j^{\mathrm{R}}, \boldsymbol{\theta}_j).$$

- The randomness model
  - random training data set sampling [11] (e.g. bagging), and
  - randomized node optimization [46]

    If T is the entire set of all possible parameters then when training the j_{th} node we only make available a small subset

The leaf prediction model

- The probabilistic leaf predictor model for the t[th] tree is then

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

The ensemble model

In a forest with T trees we have $t \in \{1, \cdots, T\}$.

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

Stopping criteria

- it is common to stop the tree when a maximum number of levels D has been reached. Alternatively, one can impose a minimum information gain.

- SVM→ in binary classification problems (only two target classes) it guarantees maximum-margin separation

- Boosting→ builds strong classifiers as linear combination of many weak classifiers

Do not extend naturally to multiple class problems

- **Problem statement.** The classification task may be summarized as follows:

> *Given a labelled training set learn a general mapping which associates previously unseen test data with their correct classes.*

- **The training objective function.**
  - Forest training happens by optimizing the parameters of the weak learner at each split node j via:

$$\boldsymbol{\theta}_j^* = \arg\max_{\boldsymbol{\theta}_j \in \mathcal{T}_j} I_j. \qquad I_j = H(\mathcal{S}_j) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i)$$

- **Randomness**.
  - Randomness is injected via randomized node optimization
  - For instance, before starting training node j we can randomly sample = 1000 parameter values out of possibly billions or even infinite possibilities.

- **The leaf and ensemble prediction models.**
  - probabilistic output as they return not just a single class point prediction but an entire class distribution.

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

- The effect of the forest size on generalization

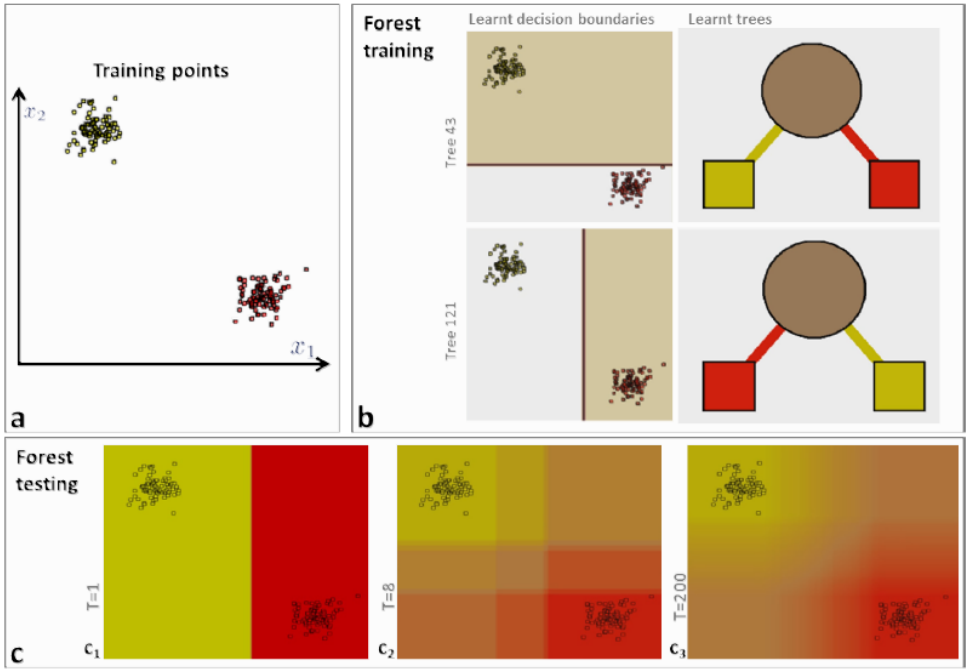


Fig. 3.3: **A first classification forest and the effect of forest size** T. **(a)** Training points belonging to two classes. **(b)** Different training trees produce different partitions and thus different leaf predictors. The colour of tree nodes and edges indicates the class probability of training points going through them. **(c)** In testing, increasing the forest size T produces smoother class posteriors. All experiments were run with $D = 2$ and axis-aligned weak learners. See text for details.

❑ Multiple classes and training noise

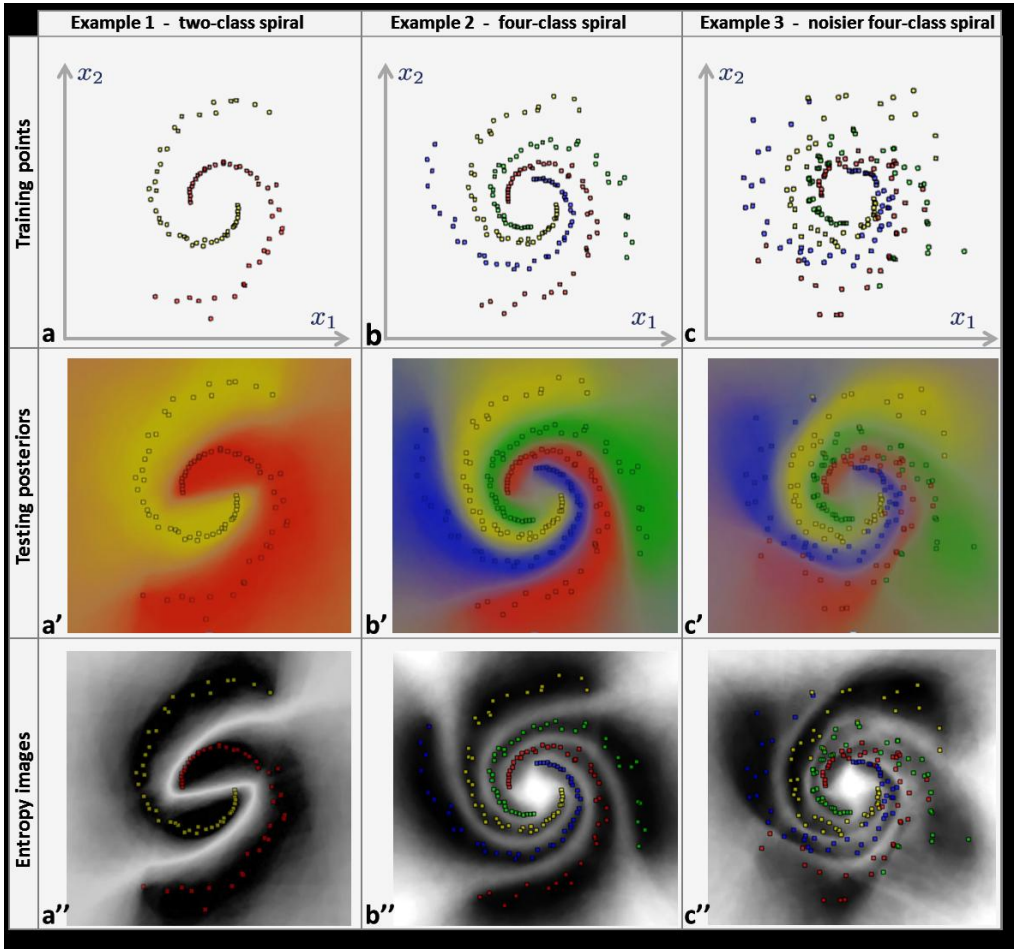
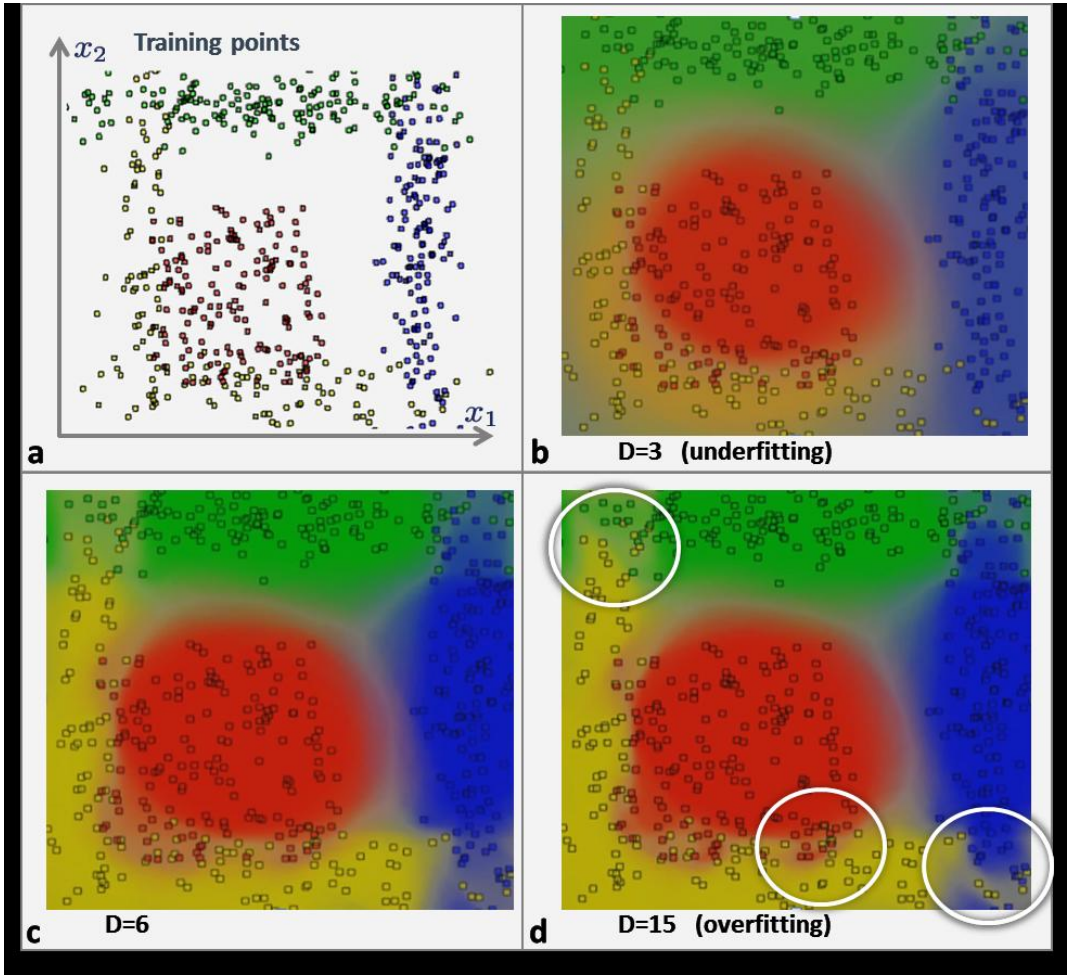


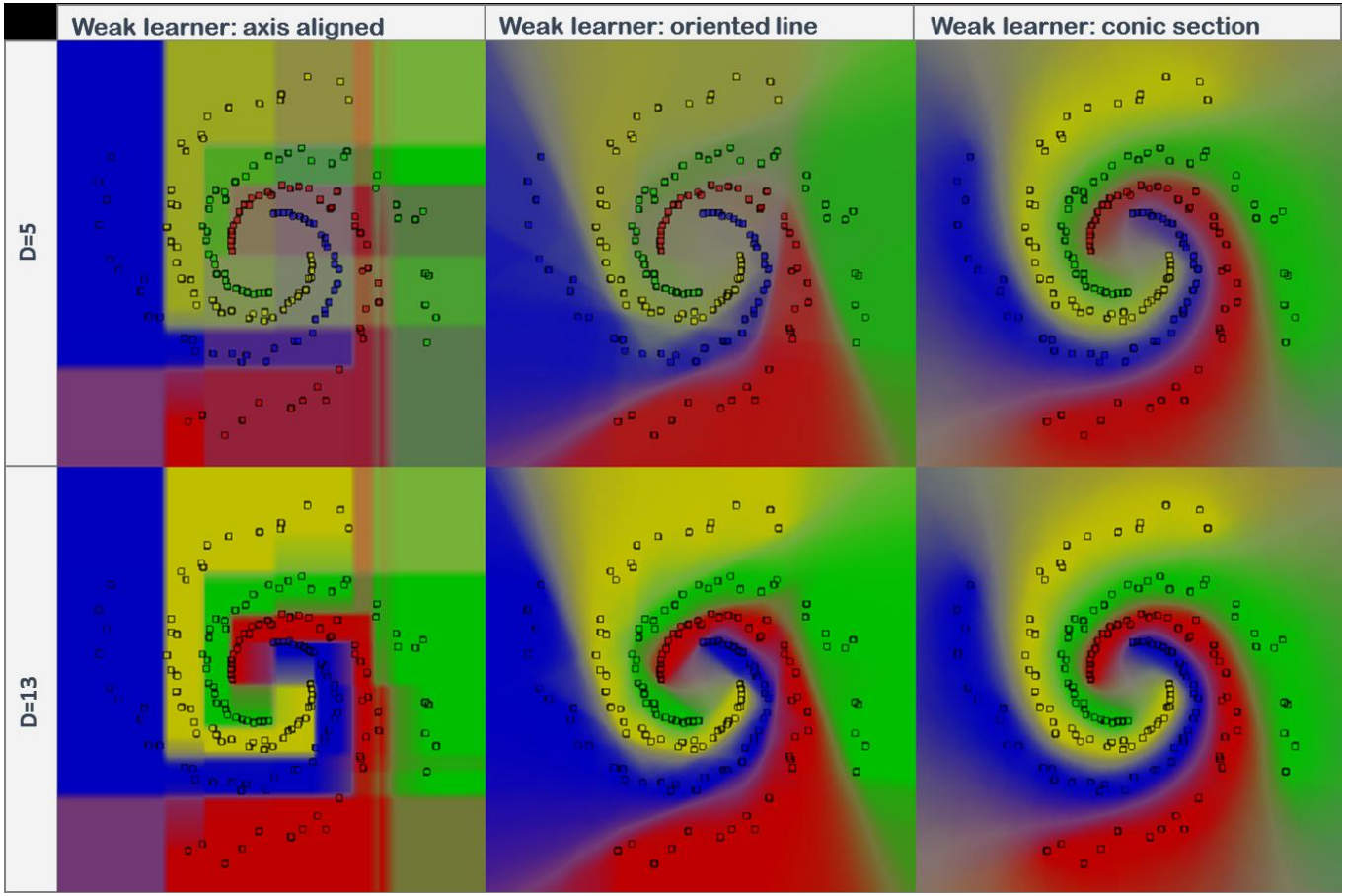
Fig. 3.4: The effect of multiple classes and noise in training data. (a,b,c) Training points for three different experiments: 2-class spiral, 4-class spiral and another 4-class spiral with noisier point positions, respectively. (a',b',c') Corresponding testing posteriors. (a'',b'',c'') Corresponding entropy images (brighter for larger entropy). The classification forest can handle both binary as well as multiclass problems. With larger training noise the classification uncertainty increases (less saturated colours in c' and less sharp entropy in c''). All experiments in this figure were run with T = 200, D = 6, and a conic-section weak-learner model

❑ "Sloppy" labels and the effect of the tree depth



The effect of tree depth. A four-class problem with both mixing of training labels and large gaps. (a) Training points. (b,c,d)
Testing posteriors for different tree depths. All experiments were run
with T = 200 and a conic weak-learner model. The tree depth is a
crucial parameter in avoiding under- or over-fitting.

❑ The effect of the weak learner



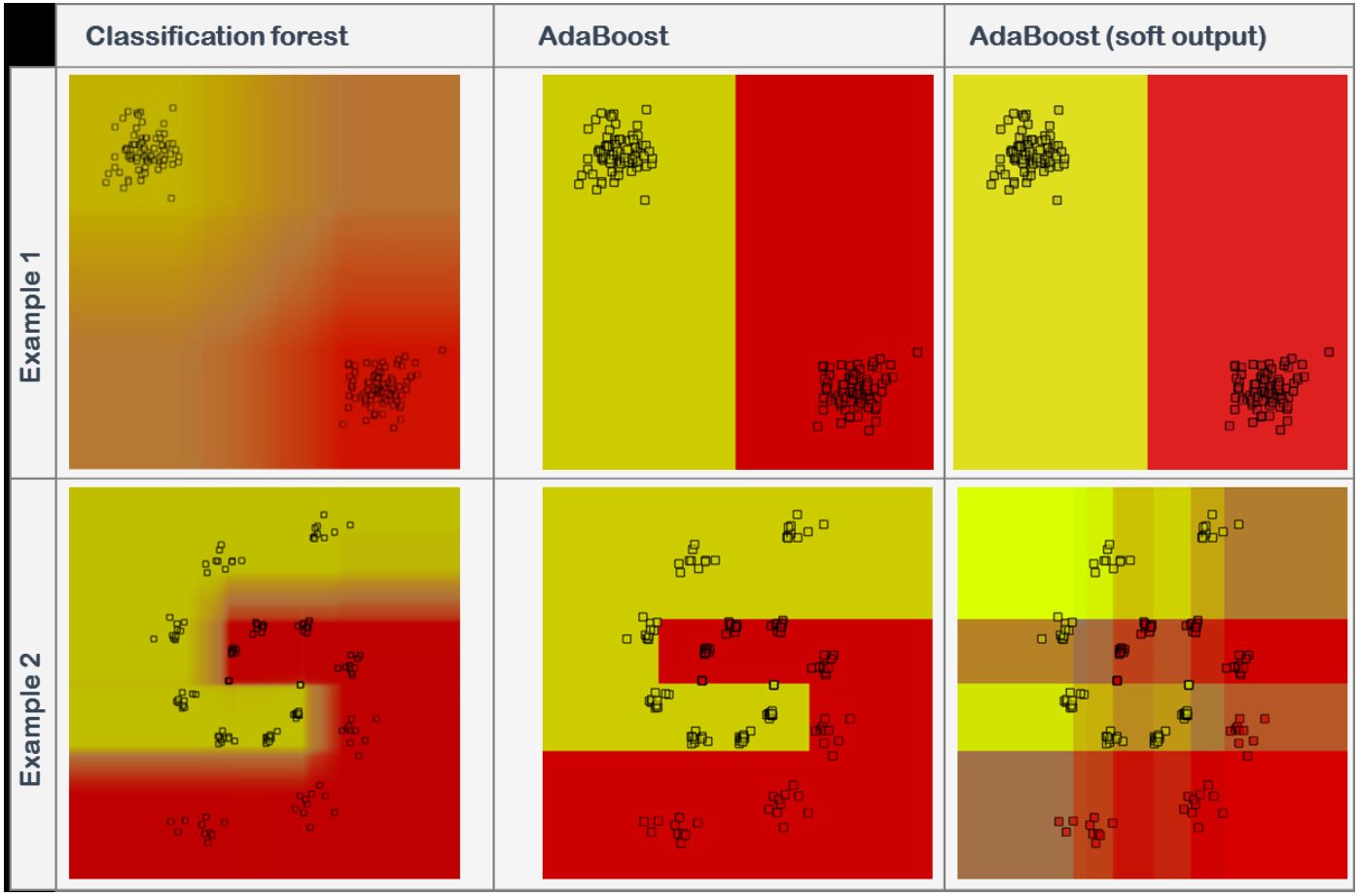
❑ The effect of randomness

Larger randomness yields a much lower overall confidence, especially noticeable in shallower trees.

- The hallmark of support vector machines is their ability to separate data belonging to different classes via a margin-maximizing surface.

- This important property is replicated in random classification forests under certain conditions.

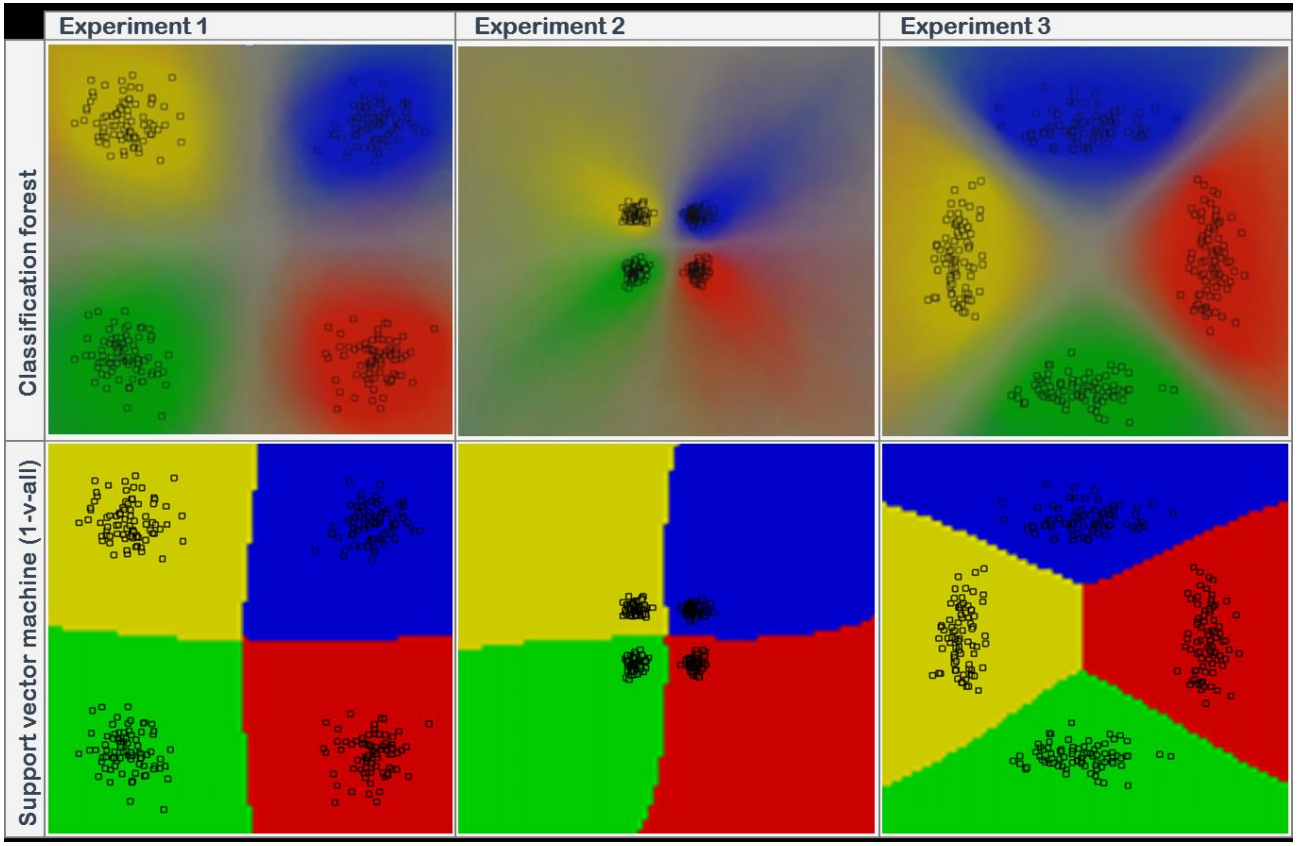
❑ Comparison with boosting



Adaboost→ overly confident.

RF→ supperior in terms of additional uncertainty

❑ Comparison with SVM



Both forests and SVMs achieve good separation results. However, forests also produce uncertainty information.

Probabilistic SVM counterparts such as the relevance vector machine do produce confidence output but at the expense of further computation.

- There are thirty one different body part classes

- The unit of computation is a single pixel in position $p \in R^2$ and with associated feature vector $v(p) \in R^d$

- Visual features are simple depth comparisons between pairs of pixel locations. So, for pixel p its feature vector $v = (x1; : : : ; xi; : : : ; xd) \in \mathbb{R}^d$ is a collection of depth differences:

$$x_i = J(\mathbf{p}) - J\left(\mathbf{p} + \frac{\mathbf{r}_i}{J(\mathbf{p})}\right) \qquad (3.2)$$

where $J(.)$ denotes a pixel depth in mm (distance from camera plane).

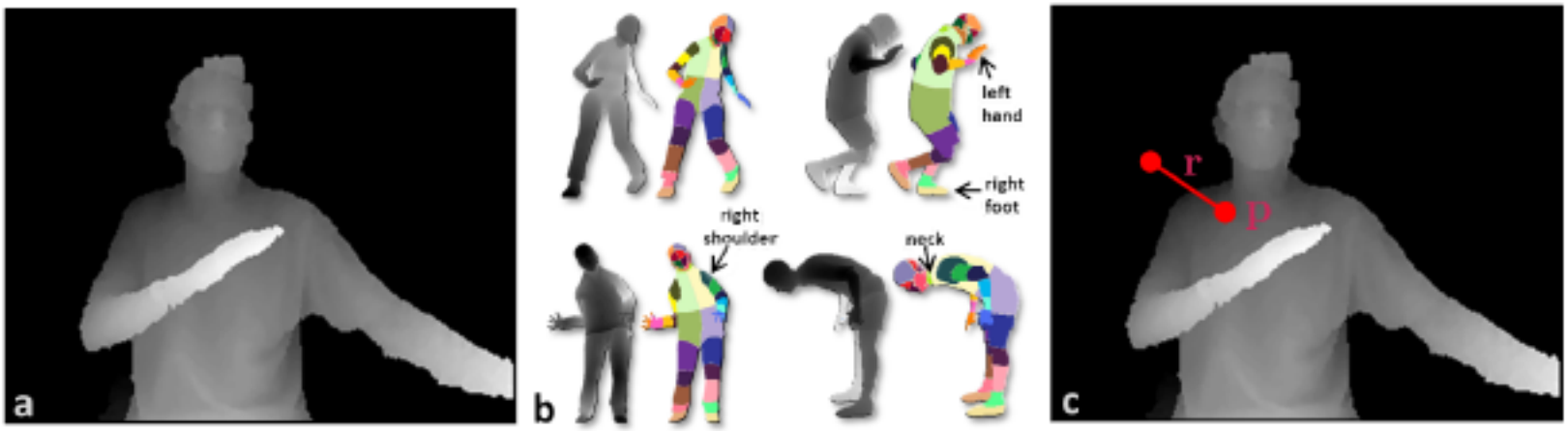


Fig. 3.15: **Classification forests in Microsoft Kinect for XBox 360.** **(a)** An input frame as acquired by the Kinect depth camera. **(b)** Synthetically generated ground-truth labeling of 31 different body parts [82]. **(c)** One of the many features of a "reference" point $\mathbf{p}$. Given $\mathbf{p}$ computing the feature amounts to looking up the depth at a "probe" position $\mathbf{p} + \mathbf{r}$ and comparing it with the depth of $\mathbf{p}$.

4. Regression forests.

- Regression forests are used for the non-linear regression of dependent variables given independent input.
- Both input and output may be multi-dimensional.
- The output can be a point estimate or a full probability density function.

4. Regression forests.
4.1 Nonlinear regression in the literature

- In geometric computer vision, a popular technique for achieving robust regression via randomization is RANSAC.
  - Disadvantage → output is non probabilistic
  - Regression forests may be thought of as an extension of RANSAC

- The success of support vector classification has encouraged the development of support vector regression (SVR)

4. Regression forests.
4.2 Specializing the decision forest model for regression

Given a labelled training set learn a general mapping which associates previously unseen independent test data with their correct continuous prediction.

4. Regression forests.
4.2 Specializing the decision forest model for regression

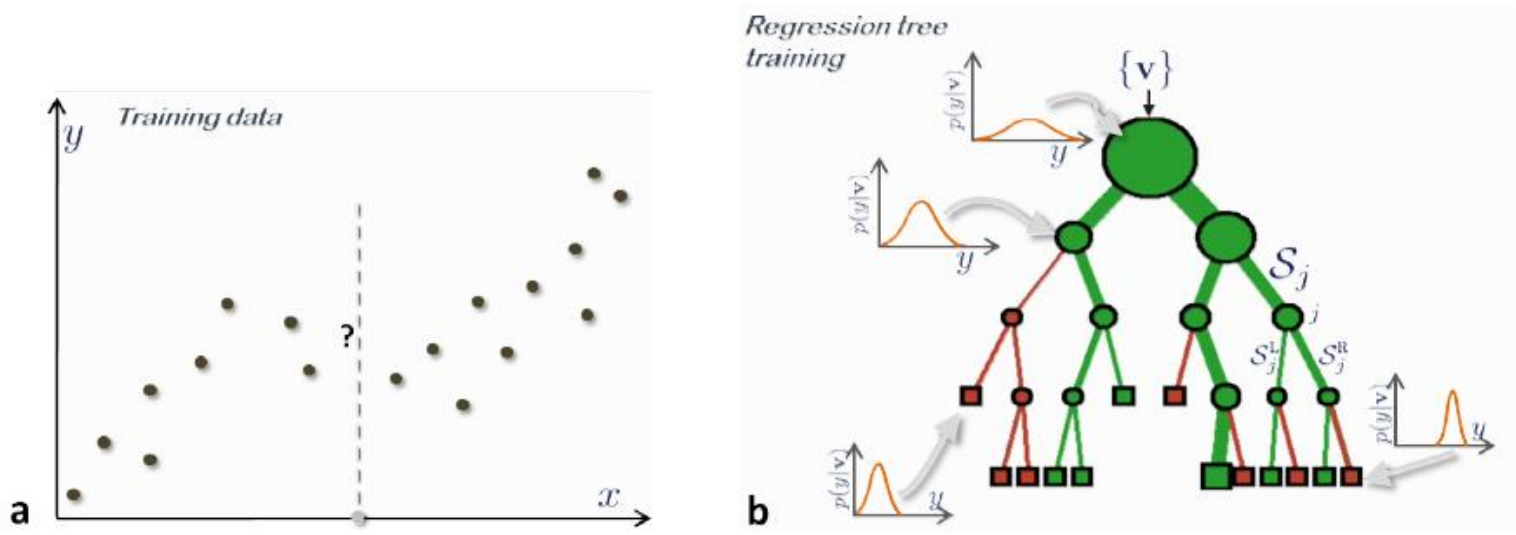


Fig. 4.1: Regression: training data and tree training. (a) Training data points are shown as dark circles. The associated ground truth label is denoted by their position along the y coordinate. The input feature space here is one-dimensional in this example (v = (x)). x is the independent input and y is the dependent variable. A previously unseen test input is indicated with a light gray circle. (b) A binary regression tree. During training a set of labelled training points {v} is used to optimize the parameters of the tree. In a regression tree the entropy of the continuous densities associated with different nodes decreases (their conffidence increases) when going from the root towards the leaves.

4. Regression forests.
4.2 Specializing the decision forest model for regression

- Given a multi-variate input v we wish to associate a continuous multi-variate label $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^n$.

- More generally, we wish to estimate the probability density function $p(\mathrm{y}|\mathrm{v})$.

4. Regression forests.
4.2 Specializing the decision forest model for regression

- The prediction model.
  - when the data reaches a terminal node then that leaf needs to make a prediction.

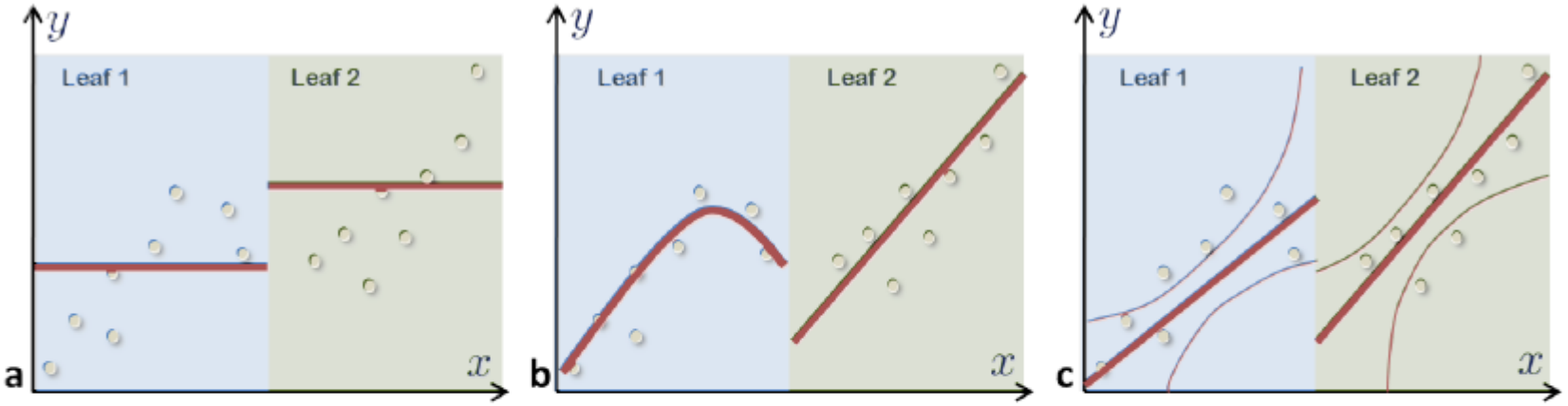


Fig. 4.2: **Example predictor models.** Different possible predictor models. **(a)** Constant. **(b)** Polynomial and linear. **(c)** Probabilistic-linear. The conditional distribution $p(y|x)$ is returned in the latter.

4. Regression forests.
4.2 Specializing the decision forest model for regression

- The ensemble model.

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

- Randomness model. (= classific.)
  - The amount of randomness is controlled during training by the parameter $\rho = |\mathcal{T}_j|$
- The training objective function.

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

Appendix A illustrates how information theoretical derivations lead to the following definition of information gain:

$$I_j = \sum_{\mathbf{v}\in\mathcal{S}_j} \log\left(|\Lambda_{\mathbf{y}}(\mathbf{v})|\right) - \sum_{\mathtt{i}\in\{\mathtt{L},\mathtt{R}\}} \left(\sum_{\mathbf{v}\in\mathcal{S}_j^{\mathtt{i}}} \log\left(|\Lambda_{\mathbf{y}}(\mathbf{v})|\right) \right) \qquad (4.2)$$

4. Regression forests.
4.2 Specializing the decision forest model for regression

- The weak learner model (≈ classific)

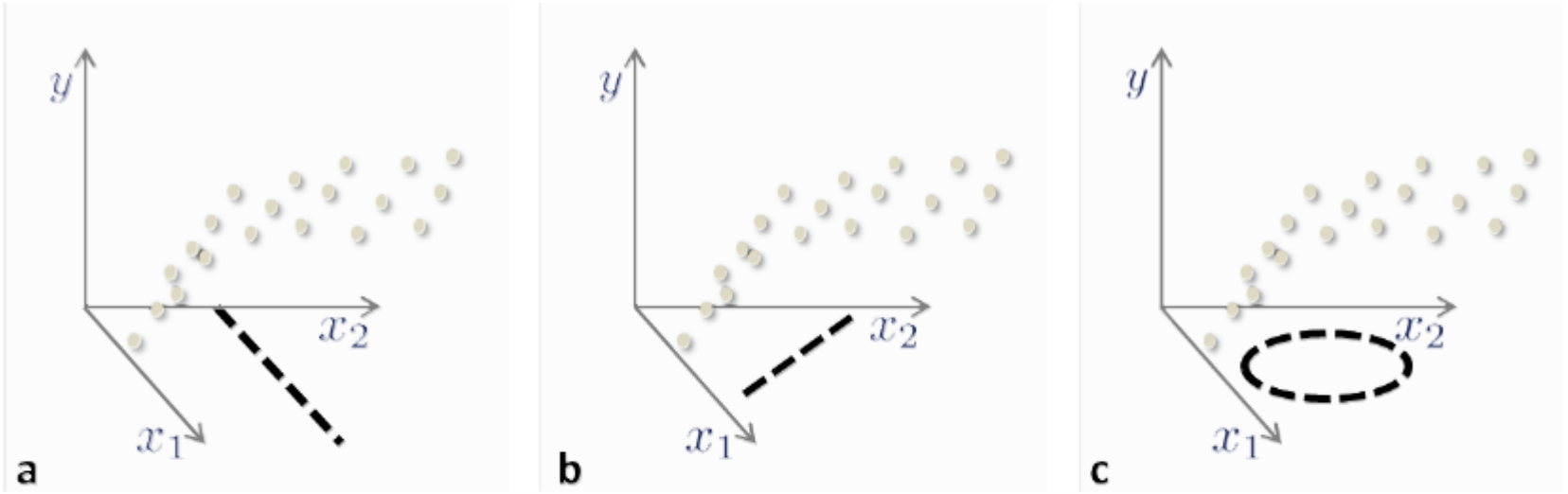


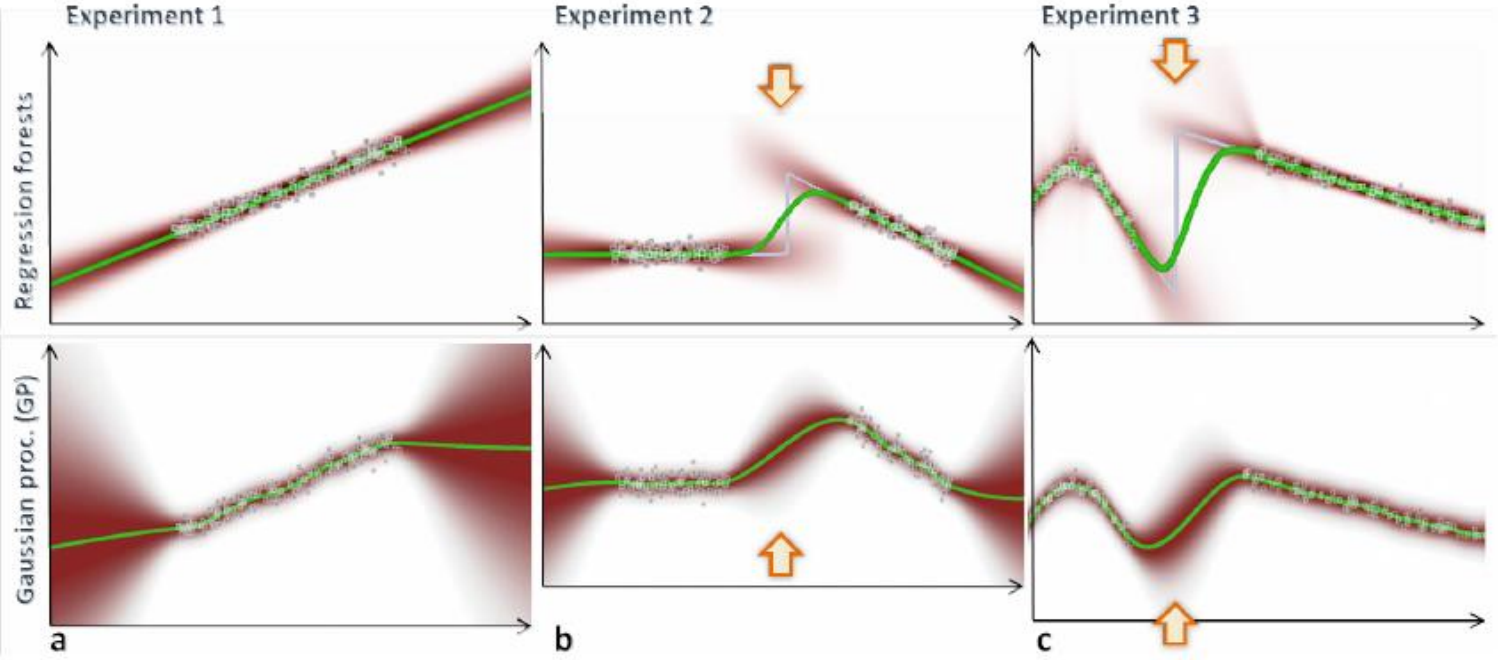
Fig. 4.5: **Example weak learners.** The (x_1, x_2) plane represents the $d-$dimensional input domain (independent). The y space represents the $n-$dimensional continuous output (dependent). The example types of weak learner are like in classification **(a)** Axis-aligned hyperplane. **(b)** General oriented hyperplane. **(c)** Quadratic (corresponding to a conic section in 2D). Further weak learners may be considered.

4. Regression forests.
4.3 Effect of model parameters

- The effect of the forest size
  - As the number of trees increases both the prediction mean and its uncertainty become smoother.

- The effect of the tree depth
  - Under and over-fitting

- Comparison with Gaussian processes
  - The hallmark of Gaussian processes is their ability to model uncertainty in regression problems.



Comparing regression forests with Gaussian processes.
(a,b,c) Three training datasets and the corresponding testing posteriors overlaid on top. In both the forest and the GP model uncertainties increase as we move away from training data. However, the actual shape of the posterior is different. (b,c) Large gaps in the training data are filled in both models with similarly smooth mean predictions (green curves). However, the regression forest manages to capture the bi-modal nature of the distributions, while the GP model produces intrinsically uni-modal Gaussian predictions.

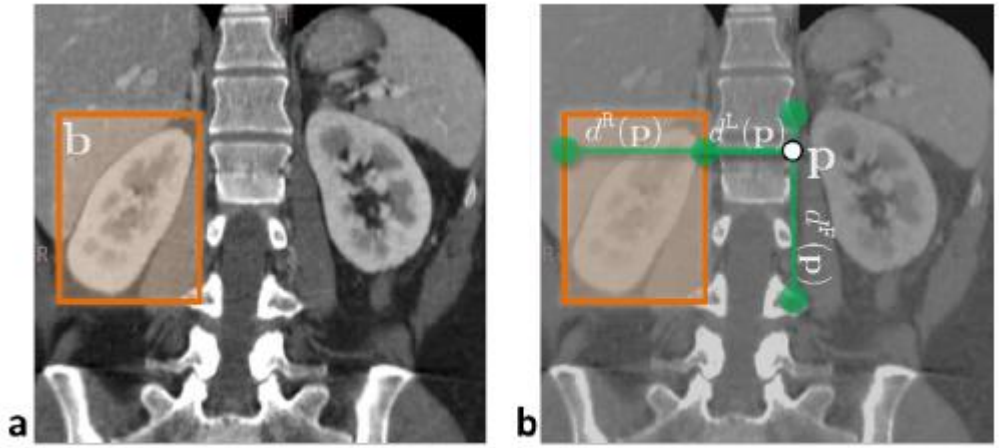
4. Regression forests.
4.5 Semantic parsing of 3D computed tomography scans

- Commercial product Microsoft Amalga Unfied Intelligence System.
- Detect the presence/absence of a certain anatomical structure
- The position of each voxel → p = (x y z).
- For each organ of interest we wish to estimate the position of a 3D axis-aligned bounding box

For a voxel $\mathbf{p}$ its feature vector $\mathbf{v}(\mathbf{p}) = (x_1, \ldots, x_i, \ldots, x_d) \in \mathbb{R}^d$ is a collection of differences:

$$x_i = \frac{1}{|\mathbf{B}_i|} \sum_{\mathbf{q} \in \mathbf{B}_i} J(\mathbf{q}). \qquad (4.5)$$

where J(p) denotes the density of the tissue in an element of volume at position p as measured by the CT scanner. B is the 3D feature box



information gain:

$$I_j = \log|\Lambda(\mathcal{S}_j)| - \sum_{i\in\{\mathrm{L},\mathrm{R}\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} \log|\Lambda(\mathcal{S}_j^i)| \qquad (4.6)$$

with $\Lambda(\mathcal{S}_j)$ the 6×6 covariance matrix of the relative displacement vector $\mathbf{d}(\mathbf{p})$ computed for all points $\mathbf{p} \in \mathcal{S}_j$. Note that here as a pre-