Analysis about Decision Tree Learning

Hypothesis Space Search by ID3

- Hypothesis space is complete (cf. Concept Learning)
- Entropy-driven search is robust to noise - always provides a solution that fits to training data
- Follows the Occam's Razor principle - "prefer shortest tree"

Occam's Razor

Why prefer short hypotheses?

Argument in favor:

- Fewer short hyps. than long hyps.

$\rightarrow$ a short hyp that fits data unlikely to be coincidence (cf. Copernicus)

$\rightarrow$ a long hyp that fits data might be coincidence

Analysis about Decision Tree Learning

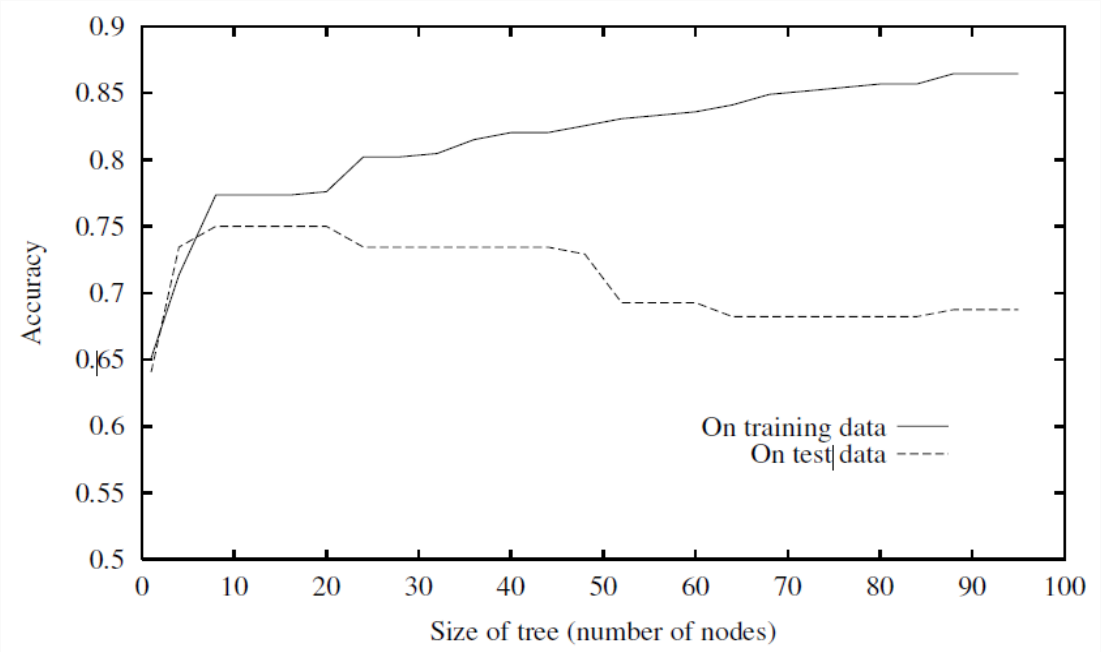
Overfitting

Overfitting

- The most informative attributes typically near the root, *but*

Overfitting

- The most informative attributes typically near the root, *but*
- all attributes tested toward the leaves $\Rightarrow$ easily overfits



Overfitting

Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution $\mathcal{D}$ of data: $error_{\mathcal{D}}(h)$

Overfitting

Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

Avoiding Overfitting

- Stop training with some criterion (i.e. few samples left)
- Grow a full tree, then post-prune
- Use a separate validation set to stop training

Avoiding Overfitting - A Post-pruning Example

1. Convert tree to equivalent set of rules
2. Prune each rule independently of others
3. Sort final rules into desired sequence for use

IF $(Outlook = Sunny) \wedge (Humidity = High)$
THEN $PlayTennis = No$

...

Perhaps most frequently used method (e.g., *C4.5*)

Analysis about Decision Tree Learning

Different types of variables

Continuous Valued Attributes

Create a discrete attribute to test continuous

- $Temperature = 82.5$
- $(Temperature > 72.3) = t, f$

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

Attributes with Many Values

Problem:

- If attribute has many values, *Gain* will select it
- Imagine using $Date = Jun\_3\_1996$ as attribute

One approach: use *GainRatio* instead

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_i is subset of S for which A has value v_i

Attributes with Costs

Consider

- medical diagnosis, *BloodTest* has cost \$150

How to learn a consistent tree with low expected cost?

One approach: replace gain by

- Tan and Schlimmer (1990)

$$\frac{Gain^2(S, A)}{Cost(A)}.$$

- Nunez (1988)

$$\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$$

  where $w \in [0, 1]$ determines importance of cost

Unknown Attribute Values

What if some examples missing values of A?

Use training example anyway, sort through tree

- If node n tests A, assign most common value of A among other examples sorted to node n
- assign most common value of A among other examples with same target value
- assign probability p_i to each possible value v_i of A
    - assign fraction p_i of example to each descendant in tree

Classify new examples in same fashion

Decision Trees – Pros & Cons

- Pros
  - Powerful hypothesis space (cf. concept learning)
  - Robust to noise (probabilistic)
  - Easy to interpret and add heuristics
  - (adaptable computing time in testing and fits nicely to GPU)
- Cons
  - Local minimum (and sensitive to training data) and severe ovefitting (high bias)

Advanced Topic: Randomisation

Randomisation

There are many ways to add randomisation to the process of learning:

- Data: Bootstrapping & bagging & cross-validation
- Features: Random splitting & sub-spaces (feature selection)
- Initialisation: Random initialisation
- Learning: Evolutionary & genetic algorithms

Power of randomisation

Randomisation is a meta-technique which can be used along with any machine learning method – often randomisation is the easier way as compared to detail analysis of method properties and sensitivity to training data, and their adjustment

Advanced Topic: Randomisation

Randomisation & Decision trees

Randomisation & Decision Trees = Random Forests

- Bagging (bootstrap aggregation) is a technique for reducing the variance of an estimated prediction function.
- Bagging seems to work especially well for high-variance, low-bias, procedures, such as trees.
- For regression:
  - We fit the same tree many times for bootstrap-sampled versions of training data and average over all trees.
- For Classification:
  - A committee of trees each casts a vote and majority wins.
- *Random forests* is a substantial modification of bagging that builds on a large collection of *de-correlated trees*.

Random Forest for Regression or Classification

1: **for** b = 1 to B **do**
2: Draw a bootstrap sample **Z** of size N from the training data.
3: Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
4: Select m variables at random from the p variables.
5: Pick the best variable among the m.
6: Split the node into daughter nodes.
7: **end for**
8: Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression:	$\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.
Classification:	Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{rf}^B(x) = \text{majority vote}\left\{\hat{C}_b(x)\right\}_1^B$.

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

J. Shotton et al. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011

Video: kinect_cvpr2011_bestpaper.mp4

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

J. Shotton et al. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011

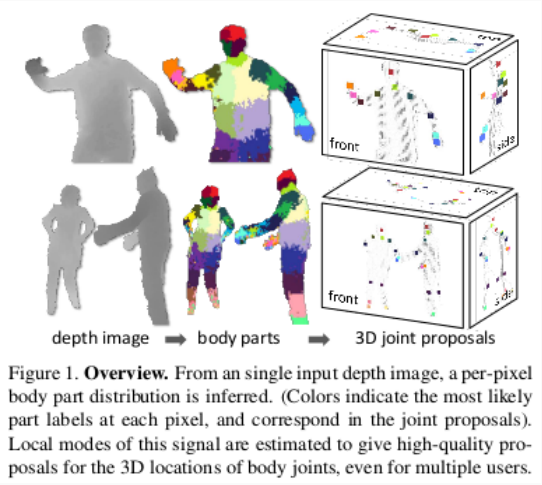


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

Video: kinect_cvpr2011_bestpaper.mp4

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

J. Shotton et al. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011

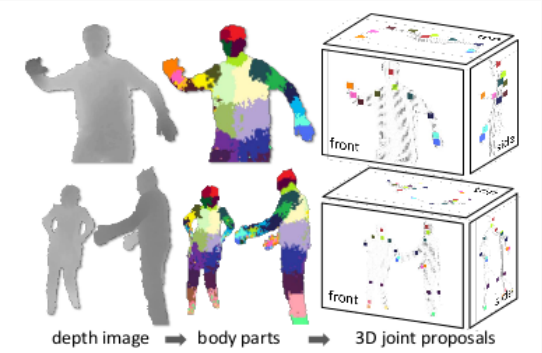


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

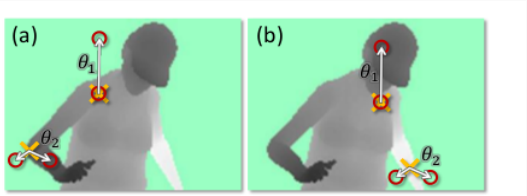


Figure 3. **Depth image features.** The yellow crosses indicates the pixel $\mathbf{x}$ being classified. The red circles indicate the offset pixels as defined in Eq. 1. In (a), the two example features give a large depth difference response. In (b), the same two features at new image locations give a much smaller response.

Video: kinect_cvpr2011_bestpaper.mp4

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

J. Shotton et al. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011

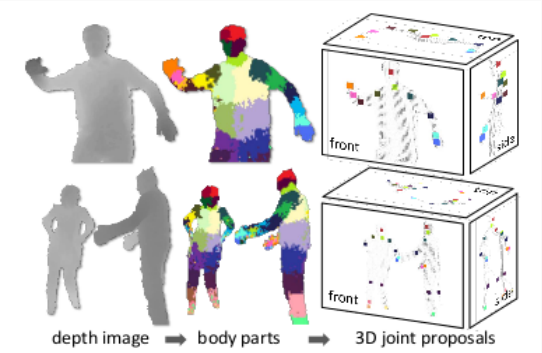


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.

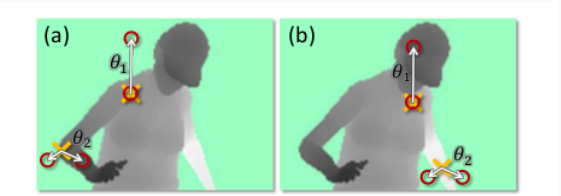


Figure 3. **Depth image features.** The yellow crosses indicates the pixel $\mathbf{x}$ being classified. The red circles indicate the offset pixels as defined in Eq. 1. In (a), the two example features give a large depth difference response. In (b), the same two features at new image locations give a much smaller response.

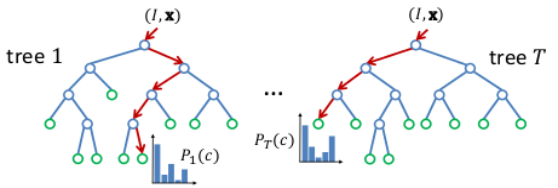


Figure 4. **Randomized Decision Forests.** A forest is an ensemble of trees. Each tree consists of split nodes (blue) and leaf nodes (green). The red arrows indicate the different paths that might be taken by different trees for a particular input.

Video: kinect_cvpr2011_bestpaper.mp4

Summary

Summary

- Decision tree model
- Entropy and information gain
- The principles of the ID3 algorithm
- Problem of overfitting
- Randomisation and the principles behing the random forest algorithm