Decision Trees and Forests

Pascal Fua
IC-CVLab
What data science methods do you use at work?

- We will focus on methods that do not rely heavily on probabilities.
- We may cover some of the others at the end of the class time permitting.
Decision Tree

Tree structure

A decision tree

root node

internal (split) node

terminal (leaf) node

Is top part blue?

false

true

Is bottom part green?

false

true

Indoor

false

true

Outdoor
Training

• Training set \( v \)
• To each sample \( v \) is assigned a class \( c \).

\( \forall v \in v \)

\[ h(v, \theta) \in \{\text{True, False}\} \]

• Compute \( p_l(c) \), the proportion of samples in each class that lands in leaf \( l \).
• Let us assume that $v$ falls into leaf $l$.
• We take the probability of belonging to class $c$, $p(c | v)$, to be $p_l(c)$ if it lands in leaf $l$. 

$$p(c | v) = p_l(c)$$
Weak Learners

Weak learner examples

**Weak learner: axis aligned.**

\[ h(v, \theta) = [\tau_1 > \phi(v) \cdot \psi > \tau_2] \]

Feature response for 2D example.

With \( \psi = (1 0 \psi_3) \) or \( \psi = (0 1 \psi_3) \)

**Weak learner: oriented line.**

\[ h(v, \theta) = [\tau_1 > \phi(v) \cdot \psi > \tau_2] \]

Feature response for 2D example.

With \( \psi \in \mathbb{R}^3 \)

**Weak learner: conic section.**

\[ h(v, \theta) = \left[ \tau_1 > \phi^T(v) \psi \phi(v) > \tau_2 \right] \]

Feature response for 2D example.

With \( \psi \in \mathbb{R}^{3 \times 3} \) a matrix representing a conic.
Let $p^k$ be the proportion of data points in $\mathcal{S}$ that are assigned to class $k$. We can define

- the Gini index $Q(\mathcal{S}) = \sum_{k=1}^{K} p^k(1 - p^k)$,
- the entropy $Q(\mathcal{S}) = -\sum_{k=1}^{K} p^k \ln p^k$.

They both vanish when $\exists k, p^k = 1$.

They are maximized when all $p^k$ are equal.

$\longrightarrow$ Minimizing these measures favors leaves in which a large fraction of samples belong to the same class.
Maximizing Information Gain

At each node, pick the weak learner that delivers the highest information gain.
Problematic for AdaBoost ....

When using linear classifiers as weak learners.
… but not for Trees
Use multiple trees to increase robustness:

\[ p(c|v) = f(p_1(c|v), \ldots, p_T(c|v)) \]

- How many trees?
- How different should they be?
- How do we fuse their outputs?
Creating Multiple Trees

- The subsets are typically chosen randomly with replacement.
- This is known as bagging.
Fusing the Output

Naive Bayesian:

\[ p(c | v) \propto \prod_{t} p_t(c | v) \]

\[ L(c, v) = \frac{1}{T} \sum_{t} - \log(p_t(c | v)) \]

- Assumes the output of each tree is independent from each other.
- Valid assumption if the training subsets are disjoint.
- Justifiable assumption if the training database is large enough.
\[ L(c, v) = \frac{1}{T} \sum_{t} - \log(p_t(c|v)) \]
Weak classifiers at every level of the tree split the space.
Graphical Interpretation

Weak classifiers at every level of the tree split the space.
Graphical Interpretation

The splits are combined by the hierarchical nature of the tree.
Graphical Interpretation

The splits are combined by the hierarchical nature of the tree.
Graphical Interpretation
Graphical Interpretation

- Each tree produces its own partition of the space.
- These partitions are combined in a Naive Bayesian manner.
Relationship to Boosting

- **Boosted Cascades:**
  - Very unbalanced tree.
  - Good for unbalanced binary problems, such as sliding window object detection.

- **Randomized forests:**
  - Less deep, more balanced.
  - Ensemble of trees gives robustness.
  - Good for multi-class problems.
3D Pose Estimation

To track the car:
1. We track interest points in the image.
2. We infer their 3D position from the tracks.
Classification-Based Approach to Matching

- One class per keypoint.
- Train a decision forest to recognize them.
Simple Weak Learners

The nodes contain simple tests of the form “Is $l(m_1) > l(m_2)$?”

Posteriors can be learned from:

- Warped images
- Video sequences
Most smooth kernels work, even simple box filters.

128, 256, or 512 binary tests usually suffice.

Random arrangement of tests effective as long as they are evenly sampled.
Point Correspondences

--> Real-time on a 2008 cell phone.
Body Part Estimation

- depth sensor
- infrared emitter
- infrared camera
- RGB camera
Depth Image
Depth Sequence

Depth image. Side view Top view
Processing Pipeline

input depth image → body parts

BPC → Clustering

body joint hypotheses

front view
side view
top view

Kinect Skeletal Tracking

EPFL
Body Part Recognition

Input depth image

Training labelled data

Visual features

Visual feature: \( x(p, \Delta p) = J(p) - J(p + \frac{\Delta p}{J(p)}) \)

Weak classifier: \( h(p, \Delta p, \tau) = x(p, \Delta p) - \tau \)

- Very fast to compute.
- Real-time performance
Synthetic Training Data

Record mocap
100,000s of poses

Retarget to varied body shapes

Render (depth, body parts) pairs

Train invariance to:
Influence of Tree Depth

Input depth  

Ground truth parts  

Inferred parts  

Depth 18
Choosing the Tree Depth

Average per-class accuracy

Tree Depth

On synthetic test data.

900k training images.

15k training images.
Choosing the Number of Trees

Average per-class accuracy

Number of trees

1 tree 3 trees 6 trees

ground truth

inferred body parts (most likely)
Result

Input depth image with background removed.

Inferred body parts posterior

\[ p(c|v) \]
Decision Forests in Short

- They make it comparatively easy to interpret what is happening.
- Their behavior is easy to modify.
- They can be trained using moderate amounts of data.

--> Very useful in many practical applications.