Outline of the lecture

This lecture discusses classification trees and how to incorporate them into an ensemble (random forest). It discusses:

- Classification trees
- Random forests
- Object detection
- Kinect
A generic data point is denoted by a vector \( \mathbf{v} = (x_1, x_2, \cdots, x_d) \)

\[ S_j = S_j^L \cup S_j^R \]

[Criminisi et al, 2011]
Use information gain to decide splits

\[ I_j = H(S_j) - \sum_{i \in \{L,R\}} \frac{|S_j^i|}{|S_j|} H(S_j^i) \]

[Crinini et al, 2011]
Advanced: Gaussian information gain to decide splits

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

[Criminisi et al, 2011]
Each split node $j$ is associated with a binary split function

$$h(v, \theta_j) \in \{0, 1\},$$

$$I_j = H(S_j) - \sum_{i \in \{L,R\}} \frac{|S_j^i|}{|S_j|} H(S_j^i)$$

[Criminisi et al, 2011]
Alternative node decisions

\[ \mathbf{v} = (x_1, x_2) \in \mathbb{R}^2 \]
Random Forests for classification or regression

1. For $b = 1$ to $B$:

   (a) Draw a bootstrap sample $Z^*$ of size $N$ from the training data.

   (b) 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.

       i. Select $m$ variables at random from the $p$ variables.
       ii. Pick the best variable/split-point among the $m$.
       iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_{1}^{B}$.

[From the book of Hastie, Friedman and Tibshirani]
Randomization

Randomized node optimization. If $\mathcal{T}$ is the entire set of all possible parameters $\theta$ then when training the $j^{th}$ node we only make available a small subset $\mathcal{T}_j \subset \mathcal{T}$ of such values.

$$\theta_j^* = \arg \max_{\theta_j \in \mathcal{T}_j} I_j.$$
Building a forest (ensemble)

In a forest with $T$ trees we have $t \in \{1, \cdots, T\}$. All trees are trained independently (and possibly in parallel). During testing, each test point $\mathbf{v}$ is simultaneously pushed through all trees (starting at the root) until it reaches the corresponding leaves.

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

[Criminisi et al, 2011]
Effect of forest size
Effect of more classes and noise

[Crinini et al, 2011]
Effect of tree depth (D)

[Criminisi et al, 2011]
Effect of bagging

Randomized node optimization (RNO)

Bagging (50%) and RNO

[Criminisi et al, 2011]
Application to face detection

Training Data
- 5000 faces
  - All frontal
- 300 million non faces
  - 94000 non-face images

[Viola and Jones, 2001]
Object detection

Idea: Extract simple features from all 24 by 24 pixel patches $x_i$. E.g., the value of a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions. Then compare the level of activation (value of the feature $f$) with respect to a threshold (theta).

\[ h_t(x_i) = \begin{cases} 
1 & \text{if } f_t(x_i) > \theta_t \\
0 & \text{otherwise}
\end{cases} \]

\[ h(x) = \begin{cases} 
1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \\
0 & \text{otherwise}
\end{cases} \]
Object detection
Random Forests and the Kinect

[Jamie Shotton et al 2011]
Random Forests and the Kinect

**Lesson 1:** Use computer graphics to generate plenty of data.

[Jamie Shotton et al 2011]
Random Forests and the Kinect

Lesson 2: Use simple depth features within random forests algorithm.

For each pixel $x$, compute the feature:

$$f_{\theta}(I, x) = d_I(x + \frac{u}{d_I(x)}) - d_I(x + \frac{v}{d_I(x)})$$

$d_I(x)$ is the depth at pixel $x$ in image $I$

Parameters $\theta = (u; v)$ describe offsets $u$ and $v$.

The normalization of the offsets ensures the features are depth invariant: At a given point on the body, a fixed world space offset will result whether the pixel is close or far from the camera.

[Jamie Shotton et al 2011]
Tree algorithm

1. Randomly propose a set of splitting candidates $\phi = (\theta, \tau)$ (feature parameters $\theta$ and thresholds $\tau$).

2. Partition the set of examples $Q = \{(I, x)\}$ into left and right subsets by each $\phi$:

   $Q_1(\phi) = \{(I, x) \mid f_\theta(I, x) < \tau\}$

   $Q_r(\phi) = Q \setminus Q_1(\phi)$

3. Compute the $\phi$ giving the largest gain in information:

   $\phi^* = \arg\max_\phi G(\phi)$

   $G(\phi) = H(Q) - \sum_{s \in \{1, r\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi))$

4. If the largest gain $G(\phi^*)$ is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets $Q_1(\phi^*)$ and $Q_r(\phi^*)$.

[Jamie Shotton et al 2011]
Performance on train and test data

[Jamie Shotton et al 2011]
Applications: Interfaces
Next lecture

There is no next lecture. Congrats!