CPSC 340 Tutorial 4

Tanner Johnson and Clement Fung

University of British Columbia

October 2, 2017

Overview

Decision Tree

- Decision Tree
- Decision Tree Learner

Random Forest

- Random Tree Learner
- Random Forest
- Bagging

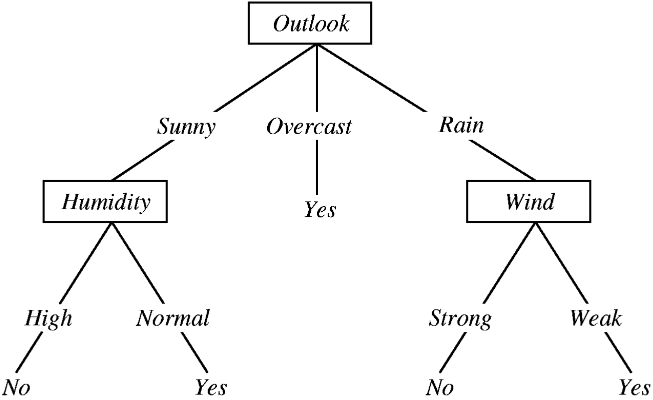
Clustering

- K-means
- Vector Quantization

A2 Code Walkthrough

Decision Tree

- Make a series of yes/no* questions to classify



Decision Tree Learner

- Start with a single node containing all the training data
- Recursively call the following until a certain depth is obtained
- Calculate Entropy/Information of the node*
- For each attribute (column / dimension)
    - For each unique value val
        - Split each point on the the rule $X > val$ where X is training data
        - Calc Information gain*
        - If best split seen so far, keep it
- If no split produces an information gain, do not split the node
- Build split and prediction functions

Random Tree Learner

- Same as previous slide
- But only iterate over $\sqrt{d}$ of the attributes (randomly chosen)
- Since we plan to train several trees, this helps decrease the correlation between each tree. (See next slide)

Random Forest

- Train several random trees (”forest”)
- To classify an input, use each of the random trees to classify
- The overall classification of the forest is the mode of all the random trees classification. *
- As long as the the tree outputs are not correlated, this method will decrease over fitting
- Bagging can also help decrease over fitting.

Bagging (Bootstrap Aggregation)

- Train each tree on a subset that is sampled uniformly and with replacement from the training data
- This reduces variance
- Some implementations weight each of the training points, and increase the weight of points that are misclassified

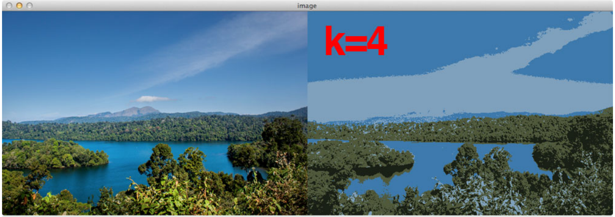
K-means Clustering

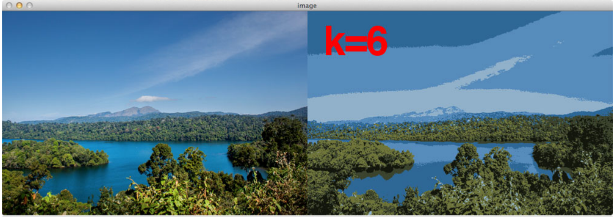
- An unsupervised clustering method
- Input: Dataset, Number of clusters
- Assigns each datapoint to a cluster
- Algorithm:
  - 1. Initialize k cluster centres
  - 2. Assign each point to its nearest cluster centre
  - 3. Move each cluster centre to the mean of the points assigned to it
  - Repeat 2-3 until no points change clusters
- Demo

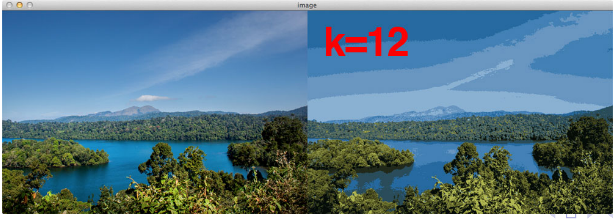
Vector Quantization (using images as motivating example)

- Currently, we can store images as 3 RGB values [0-255]
- Each pixel of an image takes 24 bits.
- If we used only 4 common RGB colours, each pixel would need 2 bits instead. 12x less.

Example







Vector Quantization

- We find these 4 colours using k-means clustering.
- Each point is in 3 dimensional space, and the resulting cluster means are the colours chosen.
- In the quantized space, each pixel value is replaced with the mean of the cluster it belongs to.
- Bigger k gives a better image, but at a lower compression rate.
- Need to store the "lookup table" for the common colours.

Assignment 2 Code

Let's walk through the A2 code.