CPSC 340: Machine Learning and Data Mining

K-Means Clustering

Fall 2015
Admin

• Assignment 1 solutions posted after class.
  – Tutorials for Assignment 2 on Monday.
Random Forests

• Random forests are one of the best ‘out of the box’ classifiers.

• Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.
Random Forests

• Random forests are one of the best ‘out of the box’ classifiers.
• Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.
Classifying Cancer Types

• “I collected gene expression data for 1000 different types of cancer cells, can you tell me the different classes of cancer?”

• We are not given the class labels $y$, but want meaningful labels.
• An example of unsupervised learning.

Unsupervised Learning

- **Supervised learning:**
  - We have features $x_i$ and class labels $y_i$.
  - Write a program that produces $y_i$ from $x_i$.

- **Unsupervised learning:**
  - We only have $x_i$ values, but no explicit target labels.
  - You want to do ‘something’ with them.

- **Some unsupervised learning tasks:**
  - Outlier detection: Is this a ‘normal’ $x_i$?
  - Data visualization: What does the high-dimensional $X$ look like?
  - Association rules: Which $x_{ij}$ occur together?
  - Latent-factors: What ‘parts’ are the $x_i$ made from?
  - Ranking: Which are the most important $x_i$?
  - Clustering: What types of $x_i$ are there?
Clustering

• **Clustering:**
  – Input: set of objects described by features $x_i$.
  – Output: an assignment of objects to ‘groups’.

• Unlike classification, we are not given the ‘groups’.
  – Algorithm must discover groups.

• Example of groups we might discover in e-mail spam:
  – ‘Lucky winner’ group.
  – ‘Weight loss’ group.
  – ‘Nigerian prince’ group.
Clustering Example
Clustering Example
Data Clustering

• General goal of clustering algorithms:
  – Objects in the same group should be ‘similar’.
  – Objects in different groups should be ‘different’.

• But the ‘best’ clustering is hard to define:
  – We don’t have a test error.
  – Generally, there is no ‘best’ method in unsupervised learning.
  – Means there are lots of methods: we’ll focus on important/representative ones.

• Why cluster?
  – You could want to know what the groups are.
  – You could want a ‘prototype’ example for each group.
  – You could want to find the group for a new example x.
  – You could want to find objects related to a new example x.
Clustering of Epstein-Barr Virus
Other Clustering Applications

• NASA: what types of stars are there?
• Biology: are there sub-species?
• Documents: what kinds of documents are on my HD?
• Commercial: what kinds of customers do I have?
• Clothing: what sizes of clothing should I make?
K-Means

• Most popular clustering method is k-means.
• Input:
  – The number of clusters ‘k’.
  – Initial guesses of the ‘mean’ of each cluster.
• Algorithm:
  – Assign each $x_i$ to its closest mean.
  – Update the means based on the assignment.
  – Repeat until convergence.
K-Means Example

Start with ‘k’ initial ‘means’
(usually, random data points)
K-Means Example

Assign each object to the closest mean.
K-Means Example

Update the mean of each group.
K-Means Example

Assign each object to the closest mean.
K-Means Example

Update the mean of each group.
K-Means Example

Assign each object to the closest mean.
Update the mean of each group.
K-Means Example

Assign each object to the closest mean.
K-Means Example

Update the mean of each group.

Stop if no objects change groups.
Cost of K-means

• The bottleneck is calculating distance from $x_i$ to mean $c$:

$$
\| x_i - \mu_c \| = \sqrt{\sum_{j=1}^{d} (x_{ij} - \mu_{cj})^2}
$$

• Each time we do this costs $O(d)$ to go through all features.

• For each of the ‘$n$’ objects, we compute the distance to ‘$k$’ clusters.

• **Total cost of assigning objects to clusters is $O(ndk)$**.
  – Fast if $k$ is not too large.

• Updating means is cheaper: $O(nd)$. 
K-Means Issues

• Guaranteed to converge when using Euclidean distance.
• Clustering a new object:
  – Assign to the nearest mean.
• Assumes you know ‘k’.
• Each object is assigned to one (and only one) cluster:
  – No possibility to leave objects unassigned.
• It may converge to sub-optimal local solution...
K-Means Clustering with Different Initialization
K-Means Initialization

• Classic approach to dealing with sensitivity to initialization:
  – Try several different random starting points, choose the ‘best’.

• Newer approach: **K-Means++**
  – Choose a random data point as the first mean.
  – Compute the distance of every point to the closest mean.
  – Sample the next proportional to these distances squared.

• K-Means++ tends to give means that are far apart.
  – Can prove it yields an approximation to optimal K-means clustering.
Vector Quantization

• K-means originally comes from signal processing.
• Designed for vector quantization:
  – Replace ‘vectors’ (objects) with a set of ‘prototypes’ (means).

• Example: Facebook places:
Vector Quantization: Image Colors

- Usual RGB representation of a pixel’s color: three 8-bit numbers.
  - For example, [241 13 50] = □.
  - Can apply k-means to find set of prototype colours.

Original: (24-bits/pixel)  
K-Means Quantized: (6-bits/pixel)
Vector Quantization: Image Colors

• Usual RGB representation of a pixel’s color: three 8-bit numbers.
  – For example, [241 13 50] = 🟠.
  – Can apply k-means to find set of prototype colours.

Original: (24-bits/pixel)  K-Means Quantized: (3-bits/pixel)
Vector Quantization: Image Colors

• Usual RGB representation of a pixel’s color: three 8-bit numbers.
  – For example, [241 13 50] = □.
  – Can apply k-means to find set of prototype colours.

Original: (24-bits/pixel)  K-Means Quantized: (2-bits/pixel)
Vector Quantization: Image Colors

- Usual RGB representation of a pixel’s color: three 8-bit numbers.
  - For example, \([241, 13, 50]\) = 🟢.
  - Can apply k-means to find set of prototype colours.
What is K-Means Doing?

• We can interpret K-Means as trying to minimize an objective:
  – Sum of distances from each object $x_i$ to its center:

$$f(\mu_1, \mu_2, ..., \mu_k, c(1), c(2), ..., c(n)) = \sum_{i=1}^{d} \| x_i - \mu_{c(i)} \|$$

• We alternate between:
  – Updating cluster assignments $c(i)$.
  – Updating means $\mu_c$.

• Convergence follows because
  – Each step does not increase the objective.
  – There are a finite number of assignments to $k$ clusters.
K-Medoids

- With other distances, k-means may not converge.
- However, changing objective function gives convergent algorithms.
- E.g., we can use the L1-norm: \[ \| x_i - m_c \|_1 = \sum_{j=1}^{d} | x_{ij} - m_{cj} | \]
- A ‘k-medoids’ algorithm based on the L1-norm optimizes:
  \[ f(m_1, m_2, \ldots, m_K; c(1), c(2), \ldots, c(n)) = \sum_{i=1}^{d} \| x_i - m_{c(i)} \|_1 \]
  - Cluster assignment based on the L1-norm.
  - Update ‘medoids’ by setting them to the median.
- This approach is more robust to outliers.
Summary

• **Unsupervised learning**: fitting data without explicit labels.
• **Clustering**: finding ‘groups’ of related objects.
• **K-means**: simple iterative clustering strategy.
• **Vector quantization**: replacing measurements with ‘prototypes’.
• **K-medoids**: generalization to other distance functions.

• Next time:
  – Non-parametric clustering.