Unsupervised Clustering

Mark Dredze

Machine Learning
CS 600.475

Today

- Focus on specific algorithms for clustering
- Gently transition into probabilistic methods

Classification

- We can't do classification anymore
  - No classes!
- But we still have a notion of groups
- Divide things into two piles
- Classification found patterns that explained a label
- We can find patterns that separate the data

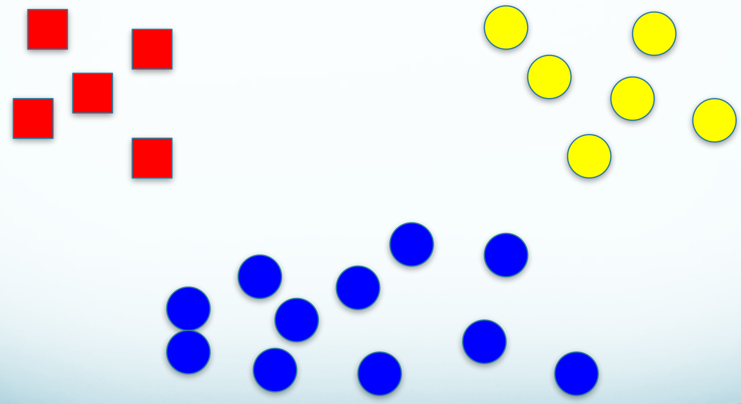
Clustering

- Sort the data into clusters (groups)
- Examples that are in the same group are similar
  - Ideally: clusters correspond to class labels
- We don't know what we will get
  - What does it mean for examples to be similar
  - Think of the problems with similarity in knn

Clustering

- Data $\{(x_i)\}_{i=1}^{N}$ $x_i \in \Re^M$
- Input: number of clusters k
- Algorithm: partition data into k clusters
  - Each x belongs to a cluster
- Cluster: a group of similar examples

Geometric Model



Solving Clustering

- How do we group examples into clusters?
- Same as before!
  - Design a model
  - Define a model objective to represent learning goal
  - Write procedure for maximizing objective
  - Compute model parameters using procedure

Defining Clusters

- A cluster is a group of similar examples
- Define cluster k by a prototype μ_k
- $r_{nk} \in \{0,1\}$, value of 1 means example n in cluster k
- $\mu_k = \frac{1}{\sum_{n=1}^{N} r_{nk}} \sum_{i=1}^{N} r_{nk} x_n$, mean of the examples in cluster k

Objective

- What are good clusters?
  - A good cluster is a group of points that is maximally similar
  - Objective: maximize the similarity of every cluster
- Objective (distortion measure)

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \left\| x_n - \mu_k \right\|^2$$

  - Each example measured as distance from prototype
    - Euclidean distance
  - Note: similar to sum of squares error

Learning

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \left\| x_n - \mu_k \right\|^2$$

- Notice there are two parameters: μ_k and r_{nk}
- Learning: select values for parameters that minimize objective

Learning

- Note that μ_k and r_{nk} are dependent on each other
  - If we knew μ_k we could set r_{nk}
    - Assign each point to closest cluster
  - If we knew r_{nk} we could set μ_k
    - Compute cluster means from examples in cluster
- Strategy: iterative procedure
  - Select μ_k that minimizes J with fixed r_{nk}
  - Select r_{nk} that minimizes J with fixed μ_k

Update Rules

- Take the derivative of J with respect to each parameter
  - Set to 0 and solve for the parameter

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \left\| x_n - \mu_j \right\|^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

Convergence

- Each update reduces the value of J
  - Therefore it will converge
- Note: J is non-convex
  - Resulting value may not be the best
  - Initialization values matter!

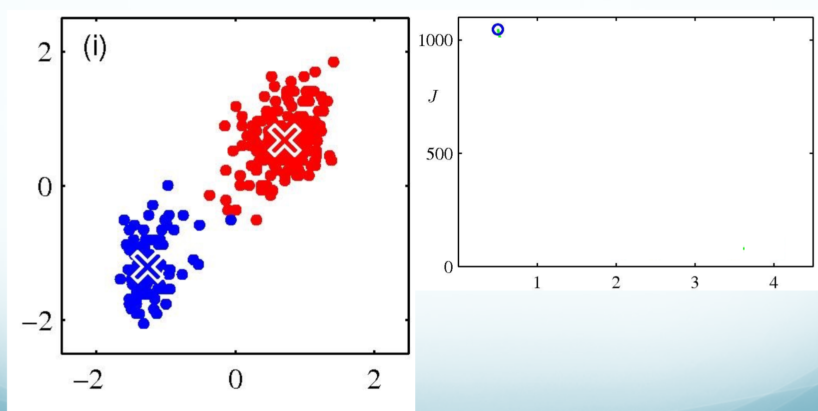
Algorithm: K-Means

- Given data $\{(x_i)\}_{i=1}^{N} \quad x_i \in \Re^M$
- Initialize μ_k
- Iteratively update until convergence:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

Illustration



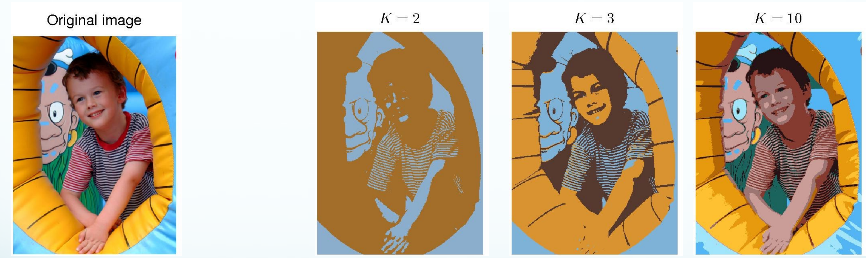
Differences from Classification

- No train and test data
  - We have no labels, so use all available data
- How do we choose k?
  - Requires human input
- Evaluation measure
  - What do we compare against?
    - Usually we have some labeled data for evaluation only

Image Compression and Segmentation

- Even this simple algorithm yields powerful results
- Image compression/segmentation
  - Each pixel is represented using RGB values
  - Cluster pixels using K-means
    - Note: ignores location of pixel in image

Image Segmentation



K-Means Issues

- Computational Complexity?
  - Re-assignment step:
    - Vector distance- M operations
    - Find best cluster for each example: K*N distances
    - Total: O(KNM)
  - Compute new means:
    - Each example added to cluster once- O(NM)
  - For I iterations, total is O(IKNM)
    - Linear in each variable
      - Still slow compared to some supervised methods
- Difficulty of finding optimal assignment?
  - NP-Hard in Euclidean space (certainly non-convex)
  - Solution: Random restarts

Problems with K-Means

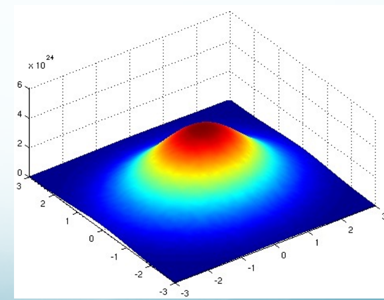
- Hard assignment
  - Examples go from one cluster to the other right away
- A smooth transition might be better
- How do we do smooth transitions?
- Probabilities!

Generative Clustering Model

- Let's come up with a generative story with clustering
- Assume we have K clusters
- Each cluster represented by a multi-variate Gaussian
- Generative process:
  - Select a cluster (a Gaussian distribution)
  - Generate an example by sampling from the Gaussian

Gaussian Mixtures

- Since we have multiple Gaussians generating points, we call the model Gaussian Mixture Model
- Why Gaussians?
  - Captures intuition about clusters
  - Examples are more likely to be near center of cluster



Gaussian Mixture Model

- .

Gaussian Mixture Model

- Cluster means, variances, coefficients

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

- Cluster Responsibilities

$$\gamma(z_k) = \frac{\pi_k N(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x \mid \mu_j, \Sigma_j)}$$

Algorithm: GMMs

- Given data $\{(x_i)\}_{i=1}^{N}$ $x_i \in \Re^M$
- Initialize μ_k Σ_k π_k
- Iteratively update until convergence:
  - $\gamma(z_k)$
  - μ_k Σ_k π_k

GMM Video

- http://www.clsp.jhu.edu/~damianos/DOC/movie_3gaussians.gif

Similarities

	K-Means	Gaussian Mixtures
Assign examples to clusters	r_{nk}	$\gamma(z_k)$
Compute new model parameters that maximize assignments	μ_k	μ_k Σ_k π_k

Same Algorithm

- The maximization algorithm for both models is the same!
- Iterate two steps
  - Compute the **expected** cluster assignments according to the current model
  - **Maximize** the model parameters according to the current cluster assignments
- Expectation Maximization Algorithm (EM)

Next Time

The EM Algorithm