Clustering Part I

Professor Marie Roch
San Diego State University

Why Cluster

- Group Similarity
- Compression algorithms
- Soundscape analysis
Clustering Methods

- K-means/Vector Quantization
- Gaussian Mixed Models

CAUTION!

Ecologist/Marine Biologist/Acoustician/Statistician crossing

I used to think correlation implied causation.  Then I took a statistics class.  Now I don't.  Sounds like the class helped.  Well, maybe.

xkcd.com
Some Background: Similarity

• How similar are two vectors?

• Distance metric (distortion)
  \[ d(x, y) = \begin{cases} 
    0 & x = y \\ 
    > 0 & x \neq y 
  \end{cases} \]

  – Satisfies triangle \( \neq \) : \( d(x, y) + d(y, z) \geq d(x, z) \)
  – Symmetric: \( d(x, y) = d(y, x) \)

Distance Between Points

• One Dimension \((x_1, y_1)\)
• Two Dimensions \((x_1, y_1, x_2, y_2)\)
• N Dimensions?
Euclidean distance/distortion

Straight line distance (squared) between two points

\[ d^2(x, y) = \sum_{i=1}^{D} (x_i - y_i)^2 \]

as a vector operation:

\[ d^2(x, y) = (x-y)^T (x - y) \]

Does Euclidean distance always make sense?
Distortion - Mahalanobis

- Mahalanobis distortion
  - Accounts for the variance and covariance ($\Sigma$)
  - Removes assumption of equal scaling

$$d_{Mahalanobis}(\bar{x}, \bar{y}) = (\bar{x} - \bar{y})' \Sigma^{-1} (\bar{x} - \bar{y})$$

$k$-means clustering
also known as vector quantization

- Let’s assume that we know there are $k$ clusters.
- How do we find them?
$k$-means clustering

- Find vectors representative of clusters.
- Representative vectors sometimes called codewords and the collection a codebook.

\[ R^2 \text{ partition induced by } k\text{-means} \]

vector to be quantized

mean vectors (codewords)

decision boundaries

Huang et al., p 165
**k-means/Vector Quantization clustering**

Select \( k \) vectors at random as initial centers from training sample \( X \)

\[
done = \text{false}; \\
\text{old\_distortion} = \infty \\
\text{while not done} \\
\quad \text{Compute } d(x_i, c_j) \text{ for each training vector and center} \\
\quad \text{Partition training vectors according to } c_j \text{ which produced smallest distortion} \\
\quad \text{Compute new centers by taking the mean (centroid) of each partition} \\
\quad \text{distortion} = \text{compute avg. minimum distortion for all training vectors} \\
\quad \text{done} = \text{distortion} / \text{old\_distortion} > \text{threshold} \\
\quad \text{old\_distortion} = \text{distortion}
\]

---

**Quantizing**

Quantization finds the closest codeword in codebook \( c \):

\[
q(\bar{x}, c) = \bar{c}_i \leftrightarrow i = \arg\min_{1 \leq k \leq K} d(\bar{x}, \bar{c}_k)
\]

Sometimes we want the distortion to the closest codeword:

\[
\text{distortion}(\bar{x}, c) = \min_{1 \leq k \leq K} d(\bar{x}, \bar{c}_k)
\]
Using Vector Quantization/k-means

• Unsupervised classifier
  – Centroids represent the distribution of items
  – Used in discrete hidden Markov models
• Supervised classifiers
  – Construct codebooks for each class
  – Find class with minimum distortion

A Supervised VQ classifier

• Training
  For each class \( \omega_i \) in \( \Omega \) construct a codebook.
• Testing
  Given a set of test vectors \( X = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_T\} \)
  Find codebook with smallest distortion for the vectors.
VQ Classification

\[ \text{MinDistortion} = \infty \]
for \( cidx = 1 \) to \( |\Omega| \)
\[ \text{SumDistortion} = 0 \]
for \( vidx = 1 \) to \( T \)
\[ \text{SumDistortion} = \text{SumDistortion} + \text{distortion}(\bar{x}_{vidx}, \text{book}_{cidx}) \]
if \( \text{SumDistortion} < \text{MinDistortion} \)
\[ \text{MinDistortion} = \text{SumDistortion} \]
\[ \text{MinIdx} = cidx \]

Decide that \( X \) belongs to class \( \omega_{\text{MinIdx}} \)

Note: Frequently, the average distortion is computed.

Clustering Part II

Professor Marie Roch
San Diego State University

J&M: 9.4.2 to end of 9.4
supplemental readings from
What happen when distributions overlap?

- We need a different way to cluster our data
- Soft methodology- probabilities not certainties

Multidimensional Gaussians

\[ f_i(x) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)} \]

where \( d = \# \) dimensions, \( i = \) classifier index

Maximum likelihood estimators are simply sample mean & variance
Gaussian Intuitions: Size of $\Sigma$

- $\mu = [0\ 0]$  
- $\Sigma = I$ 
- $\Sigma = 0.6I$ 
- $\Sigma = 2I$

As $\Sigma$ becomes larger, Gaussian becomes more spread out; as $\Sigma$ becomes smaller, Gaussian more compressed.

Gaussian Mixture Models (GMMs)

- Another unsupervised learner
- Similar to $k$-means, multiple GMMs can be used for supervised classification
- $N$ normal distributions represent data’s distribution.
  - Means: Similar to the codewords
  - $P(x \mid \mu, \Sigma)$ measures similarity
GMM: Parameters

- Parameters for each of K mixtures k=1...K:
  - $\mu_k$ mean
  - $\sum_k$ variance-covariance matrix
  - $c_k$ mixture weight where $\sum_{k=1}^{K} c_k = 1$

  - Use $\Phi_k$ to denote $(c_k, \mu_k, \sum_k)$ and $\Phi$ to denote the entire set of parameters (Note: Authors do not include $c_k$ in $\Phi_k$)

Sample 16 mixture model
Based upon R² cepstral speech data

Surface plot of pdfs
Equal likelihood contour lines
Unsupervised partitions

16 mixtures

Huang et al. p. 173

GMM: Evaluation probability

- Probability of an observation:

\[
\Pr(\bar{x} | \Phi) = \sum_{k=1}^{K} c_k N_k (\bar{x} | \mu_k, \Sigma_k)
\]

\[
= \sum_{k=1}^{K} c_k \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2} (\bar{x} - \mu_k)^T \Sigma_k^{-1} (\bar{x} - \mu_k)}
\]
GMM: Mixture weights

• Interpretation of mixture weights $c_i$

  – Prior probability that observation $x$ comes from mixture $i$.
  ... or ...
  – Scaling of mixtures such that all mixtures together form a pdf.

GMM Parameter estimation:
The EM algorithm

• Suppose that we wish to maximize a parameter set $\phi$ given $Y=y$, but $\phi$ also depends upon random variable $X$.

• That is, if we had $X=x$, we could select $\phi$ to maximize:

$$P(X = x, Y = y \mid \phi)$$
The EM Algorithm

- X=x is unavailable, and we will refer to it as hidden.
- Outline of the EM algorithm:
  1. Use an initial estimate of φ to determine E[X] taking into account Y=y.
  2. Use Y=y and E[X] to determine a new φ.
  3. If converged, stop, otherwise goto 1.
- Convergence is guaranteed

GMM: Estimation

- Application of the EM algorithm
- Expectation step
  - Determine how well each mixture models each observation
    - Determine probability of observation with respect to mixture in question.
    - Divide by probability of seeing the observation (sum across mixtures).
GMM: Expectation step

- How well does mixture $k$ model $y_i$?

$$\gamma_k^i = \frac{P(y_i | \text{mixture } k)}{P(y_i | \text{all mixtures})}$$

$$= \frac{P(y_i | \Phi_k)}{P(y_i | \Phi)}$$

$$= \frac{c_k P(y_i | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} c_j P(y_i | \mu_j, \Sigma_j)}$$

GMM: Expectation step

- Also need to determine how well each mixture represents the training data
  - Sum the $\gamma$’s for each mixture over all observations.
  - This is not a probability. Used as a normalizing factor in the maximization step.

$$\gamma_k = \sum_{i=1}^{N} \gamma_k^i$$
GMM: Maximization step

• $\hat{c}_k = \frac{\text{How well mixture } k \text{ represents training data}}{\text{How well all mixtures represent training data}}$
  
  $= \frac{\gamma_k}{\sum_{j=1}^{K} \gamma_j}$
  
  $= \frac{\gamma_k}{N}$

GMM: Maximization step

• Why does $\sum_{j=1}^{K} \gamma_j = N$?
  – $\gamma_k$ represents contribution of $k^{\text{th}}$ mixture to probability measured for each observation.
  – The total contribution to the probability for a single observation must be one. (all probability must be accounted for).
  – As there are $N$ observations, the sum is $N$. 
GMM: Maximization step

\[ \hat{\mu}_k = \frac{\sum_{i=1}^{N} (\text{Contribution mixture } k \text{ across } y's) y_i}{\sum_{i=1}^{N} y_i^t y_i} \]

\[ = \frac{\sum_{i=1}^{N} c_k P(y_i \mid \mu_k, \Sigma_k) y_i}{\sum_{i=1}^{N} c_k P(y_i \mid \Phi) P(y_i \mid \Phi)} \]

Assumes \( y_i \) is a column vector.

Use \((y_i - \mu_k)(y_i - \mu_k)'\) if \( y_i \) is a row vector.
GMM

- Convergence is typically fast 5-15 iterations
- Initialization
  - Train a $K$ codeword VQ codebook
  - For each codeword $k$, let $X^{(k)}$ denote the training vectors associated with it. Then:
    \[
    c_k = \frac{|X^{(k)}|}{\sum_{j=1}^{K} |X^{(j)}|} \quad \mu_k = mean\left(|X^{(k)}|\right) = \text{codeword}_k
    \]
    \[
    \Sigma_k = \text{cov}\left(|X^{(k)}|\right)
    \]

**Old Faithful Data**

https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
Application of GMMs

- Similar to $k$-means/VQ, we can use in unsupervised/supervised manner

- Supervised training of models is similar to our VQ classifier.
  - Joint likelihood instead of average distortion
  - Look for max likelihood instead of min distortion

GMM Classification

$$\text{MaxLikelihood} = -\infty$$
for $gidx = 1$ to $|\Omega|$ 

$$\text{JointLikelihood} = 0$$
for $vidx = 1$ to $T$

% Assume vectors independent

$$\text{JointLikelihood} = \text{JointLikelihood} + \log \left( P(\tilde{x}_{vidx} | \phi_{gidx}) \right)$$

if JointLikelihood > MaxLikelihood 

$$\text{MaxLikelihood} = \text{JointLikelihood}$$

$$\text{MaxIdx} = gidx$$

Decide that $X$ belongs to class $\omega_{\text{MaxIdx}}$
Questions?

Practicalities

• Consider a set of independent observations

\[ P(X_{1...T}) = \prod_{i=1}^{T} P(X_i) \]

and suppose that the \( P(X_i) \) was bounded by .1

\[ \log P(X_{1...T}) = \sum_{i=1}^{T} \log (P(X_i)) \]

logarithms prevent underflow
Variance-Covariance Matrices

• It is common to assume that $\Sigma^{-1}$ is diagonal.

• Why?

Precomputation

$$f_i(\bar{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\bar{x} - \mu)^T \Sigma^{-1} (\bar{x} - \mu)}$$

constant $k$ independent of $i$

$$f_i(\bar{x}) = ke^{-\frac{1}{2}(\bar{x} - \mu)^T \Sigma^{-1} (\bar{x} - \mu)}$$

define $\Delta = (\bar{x} - \mu)$

$$\log f_i(\bar{x}) = \log k + \frac{1}{2} \Delta^T \Sigma^{-1} \Delta$$

$\Sigma^{-1}$ can be precomputed, and if it is diagonal $\Sigma^{-1}(j, j) = 1/\Sigma_i(j, j)$

and $\Delta^T \Sigma^{-1} \Delta$ is just the sum of the products $\sum_{j=1}^d \Delta_i \Sigma^{-1}(j, j) \Delta_j$