Statistics 202: Data Mining

K-means clustering
Based in part on slides from textbook, slides of Susan Holmes

©Jonathan Taylor

December 2, 2012
Outline

- $K$-means, $K$-medoids
- Choosing the number of clusters: Gap test, silhouette plot.
- Mixture modelling, EM algorithm.
**K-means**

*Figure*: Simulated data in the plane, clustered into three classes (represented by red, blue and green) by the $K$-means clustering algorithm. From *ESL*. 
$K$-means

Algorithm (Euclidean)

1. For each data point, the closest cluster center (in Euclidean distance) is identified;
2. Each cluster center is replaced by the coordinatewise average of all data points that are closest to it.
3. Steps 1. and 2. are alternated until convergence. Algorithm converges to a local minimum of the within-cluster sum of squares.

Typically one uses multiple runs from random starting guesses, and chooses the solution with lowest within cluster sum of squares.
**K-means**

**Non-Euclidean**

1. We can replace the Euclidean distance squared with some other dissimilarity measure $d$, this changes the assignment rule to minimizing $d$. is identified;

2. Each cluster center is replaced by the point that minimizes the sum of all pairwise $d$’s.

3. Steps 1. and 2. are alternated until convergence. Algorithm converges to a local minimum of the within-cluster sum of $d$’s.
Figure 14.6: Successive iterations of the K-means clustering algorithm for the simulated data of Figure 14.4.
$K$-means

**Figure:** Decrease in $W(C)$, the within cluster sum of squares.
Importance of Choosing Initial Centroids

Figure: Another example of the iterations of $K$-means
$K$-means

Two different $K$-means Clusterings

Original Points

Optimal Clustering

Sub-optimal Clustering
The Iris data (K-means)
Issues to consider

- Non-quantitative features, e.g. categorical variables, are typically coded by dummy variables, and then treated as quantitative.
- How many centroids $k$ do we use? As $k$ increases, both training and test error decrease!
- By test error, we mean the within-cluster sum of squares for data held-out when fitting the clusters . . .
- Possible to get empty clusters . . .
Choosing $K$

- Ideally, the within cluster sum of squares flattens out quickly and we might choose the value of $K$ at this “elbow”.
- We might also compare the observed within cluster sum of squares to a *null* model, like uniform on a box containing the data.
- This is the basis of the gap statistic.
**K-means**

![Graph showing log $W_K$ vs. Number of Clusters]

**Figure**: Blue curve is the $W_K$ for uniform, green curve is for data.
**K-means**

![Graph showing log W_k and Gap values for different numbers of clusters](image)

**Figure**: Largest gap is at 2, and the formal rule also takes into account the variability of estimating the gap.
$K$-medoid

**Algorithm**

- Same as $K$-means, except that centroid is estimated not by the average, but by the observation having minimum pairwise distance with the other cluster members.
- Advantage: centroid is one of the observations—useful, eg when features are 0 or 1. Also, one only needs pairwise distances for $K$-medoids rather than the raw observations.
- In R, the function `pam` implements this using Euclidean distance (not distance squared).
**Example: Country Dissimilarities**

This example comes from a study in which political science students were asked to provide pairwise dissimilarity measures for 12 countries.

<table>
<thead>
<tr>
<th></th>
<th>BEL</th>
<th>BRA</th>
<th>CHI</th>
<th>CUB</th>
<th>EGY</th>
<th>FRA</th>
<th>IND</th>
<th>ISR</th>
<th>USA</th>
<th>USS</th>
<th>YUG</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRA</td>
<td></td>
<td>5.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHI</td>
<td></td>
<td></td>
<td>7.00</td>
<td>6.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUB</td>
<td></td>
<td>7.08</td>
<td>7.00</td>
<td>3.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGY</td>
<td></td>
<td>4.83</td>
<td>5.08</td>
<td>8.17</td>
<td>5.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRA</td>
<td>2.17</td>
<td>5.75</td>
<td>6.67</td>
<td>6.92</td>
<td>4.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>6.42</td>
<td>5.00</td>
<td>5.58</td>
<td>6.00</td>
<td>4.67</td>
<td>6.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISR</td>
<td>3.42</td>
<td>5.50</td>
<td>6.42</td>
<td>6.42</td>
<td>5.00</td>
<td>3.92</td>
<td>6.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>2.50</td>
<td>4.92</td>
<td>6.25</td>
<td>7.33</td>
<td>4.50</td>
<td>2.25</td>
<td>6.33</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USS</td>
<td>6.08</td>
<td>6.67</td>
<td>4.25</td>
<td>2.67</td>
<td>6.00</td>
<td>6.17</td>
<td>6.17</td>
<td>6.92</td>
<td>6.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YUG</td>
<td>5.25</td>
<td>6.83</td>
<td>4.50</td>
<td>3.75</td>
<td>5.75</td>
<td>5.42</td>
<td>6.08</td>
<td>5.83</td>
<td>6.67</td>
<td>3.67</td>
<td></td>
</tr>
<tr>
<td>ZAI</td>
<td>4.75</td>
<td>3.00</td>
<td>6.08</td>
<td>6.67</td>
<td>5.00</td>
<td>5.58</td>
<td>4.83</td>
<td>6.17</td>
<td>5.67</td>
<td>6.50</td>
<td>6.92</td>
</tr>
</tbody>
</table>
Figure: Left panel: dissimilarities reordered and blocked according to 3-medoid clustering. Heat map is coded from most similar (dark red) to least similar (bright red). Right panel: two-dimensional multidimensional scaling plot, with 3-medoid clusters indicated by different colors.
The Iris data: $K$-medoid (PAM)
**K-medoid**

### Silhouette

- For each case $1 \leq i \leq n$, and set of cases $C$ and dissimilarity $d$ define

  $$
  \bar{d}(i, C) = \frac{1}{\#C} \sum_{j \in C} d(i, j).
  $$

- Each case $1 \leq i \leq n$ is assigned to a cluster $C_{l(i)}$. The silhouette width is defined for each case as

  $$
  \text{silhouette}(i) = \frac{\min_{j \neq l(i)} \bar{d}(i, C_j) - \bar{d}(i, C_{l(i)})}{\max(\bar{d}(i, C_{l(i)}), \min_{j \neq l(i)} \bar{d}(i, C_j))}.
  $$

- High values of silhouette indicate good clusterings.
- In R this is computable for `pam` objects.
The Iris data: silhouette plot for $K$-medoid

Silhouette plot of pam($x = \text{iris}[,-5]$, $k = 3$)

- 3 clusters $C_i$
- $n_i$ : $\text{avg}_{i∈C_j} s_i$

1: 50 | 0.80
2: 62 | 0.42
3: 38 | 0.45

Average silhouette width : 0.55
The Iris data: average silhouette width
Mixture modelling

A soft clustering algorithm

- Imagine we actually had labels $Y$ for the cases, then this would be a classification problem.
- For this classification problem, we might consider using a Gaussian discriminant model like LDA or QDA.
- We would then have to estimate $(\mu_j, \Sigma_j)$ within each “cluster.” This would be easy . . .
- The next model is based on this realization . . .
Mixture modelling

**EM algorithm**

- The abbreviation: $E=\text{expectation}$, $M=\text{maximization}$.  
- A special case of an *majorization-minimization* algorithm and widely used throughout statistics.
- Particularly useful for situations in which there might be some hidden data that would make the problem easy . . .
Mixture modelling

**EM algorithm**

- In this mixture model framework, we assume that the data were drawn from the same model as in QDA (or LDA).

\[ Y \sim \text{Multinomial}(1, \pi) \quad \text{(choose a label)} \]

\[ X \mid Y = \ell \sim \mathcal{N}(\mu_\ell, \Sigma_\ell) \]

- Only, we have lost our labels and only observe \( X_{n \times p} \).

- The goal is still the same, to estimate \( \pi, (\mu_\ell, \Sigma_\ell)_{1 \leq \ell \leq k}. \)
Mixture modelling

**EM algorithm**

- The algorithm keeps track of \((\mu_\ell, \Sigma_\ell)_{1 \leq \ell \leq k}\)
- It also tracks "guesses" at \(Y\) in the form of \(\Gamma_{n \times k}\).
- Alternates between "guessing" \(Y\) and estimating \(\pi, (\mu_\ell, \Sigma_\ell)_{1 \leq \ell \leq k}\).
Mixture modelling

EM algorithm

Initialize $\Gamma, \mu, \Sigma, \pi$.

Repeat For $1 \leq t \leq T$,

Estimate $\Gamma$ These are called the *responsibilities*

$$\hat{\gamma}^{(t+1)}_{i\ell} = \frac{\hat{\pi}^{(t)}_\ell \phi^{(t)}_\ell \hat{\Sigma}^{(t)}_\ell (X_i)}{\sum_{l=1}^K \hat{\pi}^{(t)}_l \phi^{(t)}_l \hat{\Sigma}^{(t)}_l (X_i)}$$

Estimate $\mu_\ell, 1 \leq k$

$$\hat{\mu}^{(t+1)}_\ell = \frac{\sum_{i=1}^n \hat{\gamma}^{(t+1)}_{i\ell} X_i}{\sum_{i=1}^n \hat{\gamma}^{(t+1)}_{i\ell}}$$

This is just weighted average with weights $\hat{\gamma}^{(t+1)}_\ell$. 
Mixture modelling

EM algorithm

**Estimate $\Sigma_\ell$, $1 \leq k$**

\[
\hat{\Sigma}^{(t+1)}_\ell = \frac{\sum_{i=1}^{n} \hat{\gamma}_{i\ell}^{(t+1)} (X_i - \hat{\mu}_\ell^{(t+1)}) (X_i - \hat{\mu}_\ell^{(t+1)})^T}{\sum_{i=1}^{n} \hat{\gamma}_{i\ell}^{(t+1)}}
\]

This is just a weighted estimate of the covariance matrix with weights $\hat{\gamma}_{i\ell}^{(t+1)}$.

**Estimate $\pi_\ell$**

\[
\hat{\pi}_\ell^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \hat{\gamma}_{i\ell}^{(t+1)}
\]
Mixture modelling

EM algorithm

- The quantities $\Gamma$ are not really parameters, they are “estimates” of the random labels $Y$ which were unobserved.
- If we had observed $Y$ then the rows of $\Gamma$ would be all zero except one entry, which would be 1.
- In this case, estimation of $\pi_\ell, \mu_\ell, \Sigma_\ell$ is just as it would have been in QDA . . .
- The EM simply replaces the unobserved $Y$ with a guess . . .
The Iris data: Gaussian mixture modelling
The Iris data (K-means)
The Iris data: silhouette plot for $K$-medoid

Silhouette plot of pam($x = \text{iris}[, -5], k = 3$)

$n = 150$

3 clusters $C_i$

$\bar{\text{avg}}_{i \in C_j} s_i$

1: 50 | 0.80

2: 62 | 0.42

3: 38 | 0.45

Average silhouette width: 0.55