Methods for Unsupervised learning

Chapter 14

Outline

- cluster analysis: k-means, vector quantization, k-medoids
- mixtures, soft clustering, EM algorithm
- choosing number of clusters- issues and the Gap test, silhouette statistic
- hierarchical clustering- agglomerative, divisive (TSVQ)
The Supervised Learning Problem

Starting point:

- Outcome measurement $y$ (also called dependent variable, response, target)
- Vector of $p$ predictor measurements $x$ (also called inputs, regressors, covariates, features, independent variables)
- In the regression problem, $y$ is quantitative (e.g. price, blood pressure)
- In classification, $y$ takes values in a finite, unordered set (survived/died, digit 0-9, type of land use)
- We have training data $(x_1, y_1), \ldots (x_N, y_N)$. These are observations (examples, instances) of these measurements.
Objectives

On the basis of the training data we would like to:

- Accurately predict unseen test cases
- Understand which inputs affect the outcome, and how
- Assess the quality of our predictions and inferences
Examples

- Predict whether someone will have a heart attack on the basis of demographic, diet and clinical measurements
- Determine whether an incoming email is “spam”, based on frequencies of key words in the message
- Identify the numbers in a handwritten zip code, from a digitized image
- Estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum
Table 1: Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between spam and email.

<table>
<thead>
<tr>
<th></th>
<th>george</th>
<th>you</th>
<th>your</th>
<th>hp</th>
<th>free</th>
<th>hpl</th>
<th>!</th>
<th>our</th>
<th>re</th>
<th>edu</th>
<th>remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>spam</td>
<td>0.00</td>
<td>2.26</td>
<td>1.38</td>
<td>0.02</td>
<td>0.52</td>
<td>0.01</td>
<td>0.51</td>
<td>0.51</td>
<td>0.13</td>
<td>0.01</td>
<td>0.28</td>
</tr>
<tr>
<td>email</td>
<td>1.27</td>
<td>1.27</td>
<td>0.44</td>
<td>0.90</td>
<td>0.07</td>
<td>0.43</td>
<td>0.11</td>
<td>0.18</td>
<td>0.42</td>
<td>0.29</td>
<td>0.01</td>
</tr>
</tbody>
</table>
### Handwritten Digit Identification

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
A sample of segmented and normalized handwritten digits, scanned from zip-codes on envelopes. Each image has $16 \times 16$ pixels of greyscale values ranging from $0 - 255$. 
Unsupervised learning

- Here we observe the features $X_1, X_2, \ldots X_p$, but no outcome measure $Y$. Denote the data by $x_i, i = 1, 2, \ldots N$ where $x_i = (x_{i1}, x_{i2}, \ldots x_{ip})$.

- Objectives are not well-defined:
  
  Do the observations fall into natural clusters?
  
  Which joint values of the features occur the most often in the data?
  
  Are the features independent, or is there covariation?

- Unsupervised learning is more difficult, but of increasing importance. It is easier to collect data on features, without an outcome variable.
Examples

- Market basket analysis - which products do customers tend to buy together? e.g. milk, eggs, bread

- Clustering for classification - handwritten zip code problem - can we find prototype digits for 1, 2, etc, to use for classification

- DNA Microarray data - which samples cluster together? Which genes cluster together?

- Recommender systems - Netflix competition - predict ratings of unseen movies from a sparse set of ratings of other movies
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
</tr>
</tbody>
</table>

Unsupervised learning: 10
Simulated data in the plane, clustered into three classes (represented by red, blue and green) by the K-means clustering algorithm
• (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.

• 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.
Successive iterations of the $K$-means clustering algorithm for the simulated data.
Issues

• Non-quantitative feature, 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! This is unlike the supervised learning problem, where increased model complexity eventually leads to overfitting, and an increase in test error.
More details of K-means

K-means clustering algorithm

0. Start with initial guesses for cluster centers (centroids)

1. For each data point, find closest cluster center (partitioning step)

2. Replace each centroid by average of data points in its partition

3. Iterate 1+2 until convergence

(See Fig 14.4, 14.6)
Write $x_i = (x_{i1}, ..., x_{ip})$: If centroids are $m_1, m_2, ..., m_k$, and partitions are $c_1, c_2, ..., c_k$, then one can show that K-means converges to a *local* minimum of

$$
\sum_{k=1}^{K} \sum_{i \in c_k} ||x_i - m_k||^2 \quad \text{Euclidean distance}
$$

(within cluster sum of squares)

**In practice:**

- Try many random starting centroids (observations) and choose solution with smallest of squares
Stepping back

- All clustering algorithms start with a dissimilarity measure for $j^{th}$ feature

$$d_j(x_{ij}, x_{i'j})$$ and define

$$D(x_i, x_{i'}) = \sum_{j=1}^{P} d_j(x_{ij}, x_{i'j})$$

Usually $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$
Other possibilities:

- **Correlation**

  
  \[ \rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}} \]

  \[ \bar{x}_i = \text{mean of observation } i \]

- **If observations are standardized:**

  \[ x_{ij} \leftarrow \frac{x_{ij} - \bar{x}_i}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2}} \]

  then \[ 2(1 - \rho(x_i, x_{i'})) = \sum_j (x_{ij} - x_{i,j'})^2 \]

  So clustering via correlation \(\equiv\) clustering via Euclidean distance with standardized features
Partitioning (Clustering) Algorithms

- Group assignment function ("encoder") $C(i)$
  
  $C : 1, 2, ...N \rightarrow (1, 2, ...K)$

- **Criterion**: choose $C$ to minimize

  $$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})$$

  (within cluster scatter)
Fact:

- $K$-means minimizes $W(C)$ when $D = \|x_i - x_{i'}\|^2$

\[
W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \|x_i - x_{i'}\|^2
\]

\[
= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2
\]

- $K$-means solves the *enlarged* problem:

\[
\min_{C, m_1 \ldots m_k} \sum_k \sum_{C(i)=k} \|x_i - m_k\|^2
\]

to find the (near-optimal) assignment function $C$
Microarray cancer data

**Figure 1**. A color-coded expression matrix of 599 (rows) and 400 (columns) features of breast cancer data. Each column is a different sample of 153 rows of features. The display is a heat map, where red indicates high expression level and green indicates low expression level. Features are grouped by their similarity. The rows and columns are displayed in a randomly chosen order.
Human tumor data: number of cancer cases of each type, in each of the three clusters from $K$-means clustering.

<table>
<thead>
<tr>
<th></th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3</td>
</tr>
<tr>
<td>Bre</td>
<td>3 2 2</td>
</tr>
<tr>
<td>CNS</td>
<td>5 0 0</td>
</tr>
<tr>
<td>Col</td>
<td>0 0 7</td>
</tr>
<tr>
<td>K562</td>
<td>0 2 0</td>
</tr>
<tr>
<td>Leuk</td>
<td>0 6 0</td>
</tr>
<tr>
<td>MCF7</td>
<td>0 2 0</td>
</tr>
<tr>
<td>Mela</td>
<td>1 7 0</td>
</tr>
<tr>
<td>NSCLC</td>
<td>7 2 0</td>
</tr>
<tr>
<td>Ova</td>
<td>6 0 0</td>
</tr>
<tr>
<td>Pros</td>
<td>2 0 0</td>
</tr>
<tr>
<td>Renal</td>
<td>9 0 0</td>
</tr>
<tr>
<td>Unk</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>
Initialization via the k-means++ procedure

Arthur and Vassilvitskii (2007) SIAM

1. Choose one center uniformly at random from among the data points.

2. For each data point $x$, compute $D(x)$, the distance between $x$ and the nearest center that has already been chosen.

3. Choose one new data point at random as a new center, using a weighted probability distribution where a point $x$ is chosen with probability proportional to $D(x)^2$.

4. Repeat Steps 2 and 3 until $k$ centers have been chosen.

5. Now that the initial centers have been chosen, proceed using standard k-means clustering.

The resulting algorithm is faster than random-start k-means and is guaranteed to find a solution with an approximation ratio within a factor of $O(\log k)$ relative to the optimal k-means solution.
Let $g_i$ be the cluster labels, $x_0$ a test point, and $\bar{x}_k$, $k = 1, \ldots, K$ be the cluster centroids. We can write

$$\|x_0 - \bar{x}_k\|^2 = \langle x_0, x_0 \rangle - \frac{2}{N_k} \sum_{g_i = k} \langle x_0, x_i \rangle + \frac{1}{N_k^2} \sum_{g_i = k} \sum_{g_i' = k} \langle x_i, x_i' \rangle, \quad (1)$$

Hence if we start with inner products, we can compute the distance of the test point to each of the centroids, and perform K-means clustering.
Suppose instead we had available only the pairwise (squared) Euclidean distances between observations,

$$\Delta_{ii'}^2 = ||x_i - x_{i'}||^2,$$

(2)

We convert the pairwise distances to centered inner-products, and then proceed as before. We write

$$\Delta_{ii'}^2 = ||x_i - \bar{x}||^2 + ||x_{i'} - \bar{x}||^2 - 2\langle x_i - \bar{x}, x_{i'} - \bar{x} \rangle.$$

(3)

Defining $B = \{-\Delta_{ii'}/2\}$, we double center $B$:

$$\tilde{K} = (I - M)B(I - M);$$

(4)

($M = 11^T/n$) it is easy to check that $\tilde{K}_{ii'} = \langle x_i - \bar{x}, x_{i'} - \bar{x} \rangle$, the centered inner-product matrix.
Total within-cluster sum of squares for $K$-means clustering applied to the human tumor microarray data.
The Gap test

Tibshirani, Walther & Hastie (2001)

- Let $W_k$ be the total within cluster sum of squares with $k$ centroids

- $\text{Gap}(k) = E[\log(W_k)] - \log(W_k)$ where expectation is over samples drawn uniformly from smallest box containing the data

- We choose $k$ to maximize $\text{Gap}(k)$.

- There are many other competing methods. An important advantage of the Gap test is that it can choose $k = 1$. 

Left panel: observed (green) and expected (blue) values of $\log W_K$ for the simulated data. Right panel: Gap curve, equal to the difference between the observed and expected values of $\log W_K$. The Gap estimate $K^*$ is the smallest $K$ producing a gap within one standard deviation of the maximum; here $K^* = 2$. 
Gap test- microarray data

- Graph 1: Log error vs. number of clusters
  - Error values: 11.5, 12.5, 13.5
  - Clusters: 2, 4, 6, 8, 10

- Graph 2: Gap vs. number of clusters
  - Gap values: 0.7, 0.8, 0.9, 1.0
  - Clusters: 2, 4, 6, 8, 10
Kaufman & Rousseeuw (1990) proposed the Silhouette statistic, for assessing clusters and estimating the optimal number.

For observation $i$, let $a(i)$ be the average distance to other points in its cluster, and $b(i)$ the average distance to points in the nearest cluster besides its own (nearest is defined by the cluster minimizing this average distance). Then the Silhouette statistic is defined by

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$  \hspace{1cm} (5)
A point is well clustered if $s(i)$ is large. The authors propose to choose the optimal number of clusters $\hat{k}$ as the value maximizing the average $s(i)$ over the dataset. Note that $s(i)$ is not defined for $k = 1$ cluster.

Simulation studies in Tibshirani et al. (2001) show that gap and silhouette perform similarly when true number of clusters $> 1$; Silhouette not defined when $k = 1$ but gap continues to work well.

But gap (and most methods) work badly in high dimensional problems.
Some simulation experiments

Six scenarios:

1. One cluster in 2 dimensions
2. One cluster in 10 dimensions
3. Three clusters in 2 dimensions
4. Four clusters in 3 dimensions
5. Four clusters in 10 dimensions
6. Two elongated clusters, stretching out on main diagonal of a 3-dimensional cube.
1. $CH(k) = \frac{B(k) / (k-1)}{W(k) / (n-k)}$ (\(W(k)\) and \(B(k)\) are within and between sums of squares.)

2. $KL(k) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right|$ where
\[
DIFF(k) = (k - 1)^2/pW_{k-1} - k^2/pW_k
\]

3. $H(k) = \left\{ \frac{W(k)}{W(k + 1) - 1} \right\} / (n - k - 1)$ (Hartigan)

4. Silhouette

5. Gap

6. Gap in PC orientation
<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>KL</td>
<td>0*</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>12</td>
<td>5</td>
<td>13</td>
<td>5</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>Hartigan</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>48</td>
</tr>
<tr>
<td>Silhouette</td>
<td>0*</td>
<td>18</td>
<td>22</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap</td>
<td>42*</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/pc</td>
<td>44*</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Null model in 2 dimensions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0*</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KL</td>
<td>0*</td>
<td>29</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hartigan</td>
<td>0*</td>
<td>0</td>
<td>1</td>
<td>20</td>
<td>21</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Silhouette</td>
<td>0*</td>
<td>49</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/unif</td>
<td>49*</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/pc</td>
<td>50*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Null model in 10D</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0</td>
<td>0</td>
<td>50*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KL</td>
<td>0</td>
<td>0</td>
<td>39*</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hartigan</td>
<td>0</td>
<td>0</td>
<td>1*</td>
<td>8</td>
<td>19</td>
<td>13</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Silhouette</td>
<td>0</td>
<td>0</td>
<td>50*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/unif</td>
<td>1</td>
<td>0</td>
<td>49*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/pc</td>
<td>2</td>
<td>0</td>
<td>48*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Three clusters</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Estimate of number of clusters $\hat{k}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Random 4 clusters in 3D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>42*</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>35*</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hartigan</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>3*</td>
<td>9</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Silhouette</td>
<td>0</td>
<td>20</td>
<td>15</td>
<td>15*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/unif</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>47*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/pc</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>42*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Random 4 clusters in 10D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>44*</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>KL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>45*</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hartigan</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>48*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Silhouette</td>
<td>0</td>
<td>13</td>
<td>20</td>
<td>16*</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/unif</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50*</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/pc</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>46*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Two elongated clusters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>16</td>
<td>27</td>
</tr>
<tr>
<td>KL</td>
<td>0</td>
<td>50*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hartigan</td>
<td>0</td>
<td>0*</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>35</td>
</tr>
<tr>
<td>Gap/unif</td>
<td>0</td>
<td>0*</td>
<td>17</td>
<td>16</td>
<td>2</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gap/pc</td>
<td>0</td>
<td>50*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Vector Quantization

- VQ is k-means clustering, applied to vectors arising from the blocks of an image
K means clustering (encoder)

codebook (centroids) + cluster assignments

transmission

decoder

reconstructed image
Sir Ronald A. Fisher (1890-1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a $1024 \times 1024$ grayscale image at 8 bits per pixel. The center image is the result of $2 \times 2$ block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel.
k-medoid clustering

- 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.
Example: Country Dissimilarities

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

Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.
<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>5.58</td>
<td></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>7.00</td>
<td>6.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUB</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>
<td></td>
</tr>
<tr>
<td>EGY</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>
<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>
Survey of country dissimilarities. 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.
Hierarchical clustering

- This approach produces a hierarchical clustering tree, or dendrogram.

- **divisive** (top down) hierarchical clustering uses a repeated 2-means procedure. It is also known as TVSQ (tree-structured vector quantization)

- More often hierarchical clustering is done **bottom-up** (agglomerative);

- starting with single observations, the “closest” pair are joined togethers. This process is continued; along the way, we consider joining observations with clusters, and clusters with clusters.

- different metrics can be used (see next slide)

- Important advantage- all numbers of clusters are available from a single dendrogram
Dendrograms from agglomerative hierarchical clustering of human tumor microarray data.

- **single linkage**: minimum pairwise distance
- **complete linkage**: maximum pairwise distance
- **average linkage**: average pairwise distance
Issues

- Attractive property: allows one to see all numbers of clusters at once.

- Can use any distance measure- Euclidean distance and correlation are the most popular

- Algorithm is greedy, clusters are not “optimal” in any sense

- Choice of features to use in clustering can make a big difference in the result. Eg choice of genes when clustering samples in microarray data

- Clusters cannot overlap- when clustering genes, this can be a drawback
Eisen’s Cluster program

- widely used at Stanford and in biology community- in Stanford Microarray Database (SMD)

- uses Euclidean distance or correlation between centroids, rather than the usual average pairwise distance. Michael Eisen (Stanford postdoc) did some experiments and felt it “worked better”. Can give noticeably different results than standard metric

- “centroid” option in R hclust program; can yield “inversions” in dendrogram.
Microarray data- results
Node ordering problem

- heatmaps are re-ordered according to order of leaves in dendrogram
- but dendrogram does not yield a unique order- can flip any of the $n$ internal nodes—$>2^{n-1}$ orderings
- have to use an additional criterion, eg in R package hclust, tightest cluster is put on left
- in Cluster program in SMD (Stanford Microarray Database), each pair of nodes is flipped so that closest pair are beside each other.
Unsupervised via supervised

- General idea: generate a reference sample to compare with the training sample.

- Use a two-class classifier to try to discriminate the training sample from the reference sample.

- Use the result of this classification to discover interesting properties of the training sample, e.g. clusters, or areas of high data density.
Details

- We have \( N \) observations on features \( X_1, X_2, \ldots X_p \). Call this dataset class 1, with corresponding value of a target \( Y=1 \)
- create an artificial “background” sample of size \( N \) with values for features \( X_1, X_2, \ldots X_p \). Call this class 0 and set \( Y = 0 \) for this sample.
- fit a classifier to the \( 2N \) observations \((Y, X_1, X_2, \ldots X_p)\), and hence find regions in feature space where probability of class 1 is high. These are regions of high relative density of the original data vs background sample
- How to generate the background sample? 1) uniform over box containing original data \( \Rightarrow \) we find regions of high density relative to uniform 2) product of data marginals \( \Rightarrow \) we find regions of high density relative to independence
Density estimation via classification. Left panel: Training set of 200 data points. Right panel: Training set plus 200 reference data points, generated uniformly over the rectangle containing the training data. The training sample was labeled as class 1, and the reference sample class 0, and a semiparametric logistic regression model was fit to the data. Some contours for \( \hat{g}(x) \) are shown.
CART: a better choice of classifier

- Classification and regression tree (CART) method fits a binary tree to the data- see chapter 9 of text
- at each node a split is made on one feature
- Hence use of CART in this setting provides info like “there are many observations with $x_1 > 4$ and $x_3 < 2$”, or “there are few observations with $x_2 < 1$ and $x_5 < 2$”
Example

- 1000 data points in 10 dimensions.
- 100 points are uniform on \([-1, 1]^{10}\).
- 900 points are uniform on \([0, 1]^{3}\) (first 3 features) and uniform on \([-1, 1]^{7}\) (last 7 features).
- We generated 1000 reference data points on \([-1, 1]^{10}\).
Results

\[ xx.3 < 0.000703851 \]
\[ xx.1 < -0.00635388 \]
\[ xx.2 < -0.0446066 \]
References
