Inf2b Learning and Data
Lecture 3: Clustering and data visualisation

Hiroshi Shimodaira
(Credit: Iain Murray and Steve Renals)
Centre for Speech Technology Research (CSTR)
School of Informatics
University of Edinburgh
http://www.inf.ed.ac.uk/teaching/courses/inf2b/
https://piazza.com/ed.ac.uk/spring2018/inf2b08009learning
Office hours: Wednesdays at 14:00-15:00 in IF-3.04
Jan-Mar 2018

Today’s Schedule

1. What is clustering
2. K-means clustering
3. Hierarchical clustering
4. Example – unmanned ground vehicle navigation
5. Dimensionality reduction with PCA and data visualisation
6. Summary

Visualisation of film review users

MovieLens data set
(http://grouplens.org/datasets/movielens/)
C ≈1000 users, M ≈1700 movies

2D plot of users based on rating similarity

Application of clustering

- Face clustering
doi: 10.1109/CVPR.2013.450
LHI-Animal-Face dataset
- Image segmentation
http://dx.doi.org/10.1093/bioinformatics/btr246
- Document clustering
- Thesaurus generation
- Temporal Clustering of Human Behaviour
http://www.f-zhou.com/tc.html

A two-dimensional space

Manderins

http://homepages.inf.ed.ac.uk/imurray2/teaching/oranges_and_lemons/
Clustering and data visualisation

If centres moved,
Pick
Assign each point to its nearest centre

Evaluation of clustering
One way to measure the quality of a k-means clustering solution is by a sum-squared error function, i.e. the sum of squared distances of each point from its cluster centre.

Let \( z_{kn} = 1 \) if the point \( x_n \) belongs to cluster \( k \) and \( z_{kn} = 0 \) otherwise. Then:
\[
E = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn} \| x_n - m_k \|^2
\]
where \( m_k \) is the centre of cluster \( k \).

Sum-squared error is related to the variance — thus performing k-means clustering to minimise \( E \) is sometimes called minimum variance clustering.

This is a within-cluster error function — it does not include a between clusters term

\* In the unlikely event of a tie, break tie in some way. For example, assign to the centre with smallest index in memory.

K-means clustering
A simple algorithm to find clusters:

- Pick \( K \) random points as cluster centre positions
- Assign each point to its nearest centre
- Move each centre to mean of its assigned points
- If centres moved, goto 2.

K-means is an optimisation algorithm for \( \mathcal{L} \).
Local optima are found, i.e. there is no guarantee of finding global optimum. Running multiple times and using the solution with best \( \mathcal{L} \) is common.
How to decide $K$?

- The sum-squared error decreases as $K$ increases ($E \to 0$ as $K \to N$)
- We need another measure?!

![Graph showing sum-squared error vs $K$](image)

Failures of K-means (e.g. 1)

Large clouds pull small clusters off-centre

Failures of K-means (e.g. 2)

Distance needs to be measured sensibly.

Hierarchical clustering

Form a ‘dendrogram’ / binary tree with data at leaves

Bottom-up / Agglomerative:
- Repeatedly merge closest groups of points
- Often works well. Expensive: $O(N^3)$

Top-down / Divisive:
- Recursively split groups into two (e.g. with k-means)
- Early choices might be bad.
- Much cheaper! $\sim O(N^2)$ or $O(N^2 \log N)$

More detail: Pattern Classification (2nd ed.), Duda, Hart, Stork. §10.9

Stanley

Stanford Racing Team; DARPA 2005 challenge

http://robots.stanford.edu/talks/stanley/

Inside Stanley

Perception and intelligence

It would look pretty stupid to run off the road, just because the trip planner said so.
Stanley used a Gaussian mixture model. “Souped up k-means.”
The cluster just in front is road (unless we already failed).

- High-dimensional data are difficult to understand and visualise.
- Consider dimensionality reduction for visualisation.

**Principal Component Analysis (PCA)**

**Optimal projection of 2D data onto 1D**

- Mapping 2D to 1D: $y_n = u^T x_n = u_1 x_{n1} + u_2 x_{n2}$
- Optimal mapping: $\text{max} \ Var(y)$
- $\text{Var}(y) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{y})^2$
- cf. least squares fitting (linear regression)

**Principal Component Analysis (PCA) (cont.)**

- Let $v = p_2$, i.e. the second largest eigen value, $\lambda_2$
- Map $x_n$ onto the axis by $v$:
  $z_n = v^T x_n = v_1 x_{n1} + \cdots + v_D x_{nD}$
- point $(y_{n1}, z_{n1})^{T}$ in $\mathbb{R}^2$ is the projection of $x_n \in \mathbb{R}^D$ on the 2D plane spanned by $u$ and $v$.
- $\text{Var}(y) = \lambda_1$, $\text{Var}(z) = \lambda_2$
- Can be generalised to mapping from $\mathbb{R}^D$ to $\mathbb{R}^\ell$ using $\{p_1, \ldots, p_\ell\}$, where $\ell < D$.
- NB: Dimensionality reduction may involve loss of information. Some information will be lost if
  $\frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{D} \lambda_i} < 1$

**Covariance matrix**

$\Sigma = \begin{bmatrix} s_{11} & \cdots & s_{1D} \\ \vdots & \ddots & \vdots \\ s_{D1} & \cdots & s_{DD} \end{bmatrix}$ is D-by-D symmetric matrix

- In scalar representation:
  $s_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)(x_{nj} - \bar{x}_j)$, $\bar{x}_i = \frac{1}{N} \sum_{n=1}^{N} x_{ni}$
- Relation with Pearson’s correlation coefficient:
  $r_{ij} = \frac{1}{\sqrt{s_{ii} s_{jj}}} \frac{1}{N-1} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)(x_{nj} - \bar{x}_j)$
  $= \frac{1}{\sqrt{s_{ii} s_{jj}}} \frac{1}{N-1} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)(x_{nj} - \bar{x}_j)$
  $= \frac{s_{ij}}{\sqrt{s_{ii} s_{jj}}} \quad \text{cf. } \cos \theta = \frac{1}{\sqrt{s_{ii} s_{jj}}} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)(x_{nj} - \bar{x}_j)$
- Covariance matrix $\Sigma$ is a symmetric matrix

**PCA on the film review toy data**

<table>
<thead>
<tr>
<th>Movie</th>
<th>Action</th>
<th>Drama</th>
<th>Horror</th>
<th>Romance</th>
<th>Sci-Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>McCarthy</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>M-Storm</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Pong</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Travers</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Turan</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

**Covariance matrix**

$\Sigma = \begin{bmatrix} 2.66 & -1.07 & 5.53 & -4.67 & -0.67 \\ 0.001 & 0.97 & 0.02 & 1.27 & 0.12 \\ 0.256 & 1.99 & 0.66 & 0.44 & 0.504 \\ 0.031 & 1.77 & 0.67 & 0.20 & 1.87 \\ -4.67 & 5.37 & 0.67 & 10.97 & 5.37 \\ -1.20 & 0.65 & 0.20 & 1.90 & 0.65 \\ -0.80 & 0.94 & 0.10 & 0.70 & 1.07 \\ -0.67 & 1.27 & 0.87 & 6.67 & 0.92 \end{bmatrix}, \quad P = \begin{bmatrix} -0.941 & 0.345 & -0.269 & -0.160 & 0.053 & -0.12 \\ 0.265 & 0.917 & 0.058 & 0.448 & -0.504 & 1.190 \\ 0.001 & 0.778 & -0.503 & 0.028 & -0.390 & 1.383 \\ -0.275 & 0.154 & -0.098 & -0.160 & 0.025 & 0.483 \\ -0.481 & 0.081 & 0.094 & 0.720 & 0.743 & 0.005 \\ 0.001 & 0.461 & 0.676 & 0.106 & -0.047 & 0.375 \end{bmatrix}$

$\Sigma = \begin{bmatrix} 20.78 & 1.25 & 4.67 & 0.80 & 1.07 \\ 0.10 & 0.33 & 0.67 & 1.20 & 0.67 \\ 0.91 & 0.67 & 0.67 & 1.00 & 0.67 \\ 0.01 & 0.33 & 0.67 & 1.20 & 0.67 \\ 1.20 & 0.67 & 0.67 & 1.00 & 0.67 \\ 0.01 & 0.33 & 0.67 & 1.20 & 0.67 \end{bmatrix}$

$\Sigma = \begin{bmatrix} 20.78 & 1.25 & 4.67 & 0.80 & 1.07 \\ 0.10 & 0.33 & 0.67 & 1.20 & 0.67 \\ 0.91 & 0.67 & 0.67 & 1.00 & 0.67 \\ 0.01 & 0.33 & 0.67 & 1.20 & 0.67 \\ 1.20 & 0.67 & 0.67 & 1.00 & 0.67 \end{bmatrix}$

$\Sigma = \begin{bmatrix} 20.78 & 1.25 & 4.67 & 0.80 & 1.07 \\ 0.10 & 0.33 & 0.67 & 1.20 & 0.67 \\ 0.91 & 0.67 & 0.67 & 1.00 & 0.67 \\ 0.01 & 0.33 & 0.67 & 1.20 & 0.67 \\ 1.20 & 0.67 & 0.67 & 1.00 & 0.67 \end{bmatrix}$

$\Sigma = \begin{bmatrix} 20.78 & 1.25 & 4.67 & 0.80 & 1.07 \\ 0.10 & 0.33 & 0.67 & 1.20 & 0.67 \\ 0.91 & 0.67 & 0.67 & 1.00 & 0.67 \\ 0.01 & 0.33 & 0.67 & 1.20 & 0.67 \\ 1.20 & 0.67 & 0.67 & 1.00 & 0.67 \end{bmatrix}$
PCA on the film review toy data (cont.)

Dimensionality reduction $D \rightarrow \ell$ by PCA

\[
\begin{pmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_\ell
\end{pmatrix}
= \begin{pmatrix}
  p_1^T \\
  p_2^T \\
  \vdots \\
  p_\ell^T
\end{pmatrix}
\begin{pmatrix}
  x \\
  x \\
  \vdots \\
  x
\end{pmatrix}
\]

where $\{p_i\}_{i=1}^\ell$ are the eigen vectors for the $\ell$ largest eigen values of $S$. The above can be rewritten as

\[
y = A^T x \quad \text{linear transformation from } R^D \text{ to } R^\ell
\]

\[
y = (y_1, \ldots, y_\ell)^T : \ell\text{-dimensional vector}
\]

\[
A = (p_1, \ldots, p_\ell) : D \times \ell \text{ matrix}
\]

In many applications, we normalise data before PCA, e.g. $y = A^T (x - \bar{x})$.

Summary
- **Clustering**
  - $K$-means for minimising ‘cluster variance’
  - Review notes, not just slides
  - [other methods exist: hierarchical, top-down and bottom-up]

- **Unsupervised learning**
  - Spot structure in unlabelled data
  - Combine with knowledge of task

- **Principal Component Analysis (PCA)**
  - Find principal component axes for dimensionality reduction and visualisation

- Try implementing the algorithm! (Lab 3 this week)

Quizes
- **Q1:** Find computational complexity of $k$-means algorithm
- **Q2:** For $k$-means clustering, discuss possible methods for mitigating the local minimum problem.
- **Q3:** Discuss possible problems with $k$-means clustering and solutions when the variances of data (i.e. $s_i, i=1, \ldots, D$) are much different from each other.
- **Q4:** For $k$-means clustering, show $E(t+1) \leq E(t)$, (NE)
- **Q5:** At page 37, show $y = u^T x$.
- **Q6:** At page 39, show $\text{Var} (y) = \lambda_1$, where $\lambda_1$ is the largest eigen value of $S$. (NE)
- **Q7:** The first principal component axis is sometimes confused with the line of least squares fitting (or regression line). Explain the difference.