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/spring2017/infr08009learning

Jan-Mar 2017
Today’s Schedule

1. What is clustering
2. K-means clustering
3. Hierarchical clustering
4. Example – unmanned ground vehicle navigation
5. Dimensionality reduction and data visualisation
6. Appendix
Clustering: partition a data set into meaningful or useful groups, based on distances between data points

Clustering is an unsupervised process — the data items do not have class labels

Why cluster?

Interpreting data  Analyse and describe a situation by automatically dividing a data set into groupings

Compressing data  Represent data vectors by their cluster index — vector quantisation
“Human brains are good at finding regularities in data. One way of expressing regularity is to put a set of objects into groups that are similar to each other. For example, biologists have found that most objects in the natural world fall into one of two categories: things that are brown and run away, and things that are green and don’t run away. The first group they call animals, and the second, plants.”

**Recommended reading:** David MacKay textbook, p284–
Visualisation of film review users

**MovieLens data set**
(http://grouplens.org/datasets/movielens/)

$C \approx 1000$ users, $M \approx 1700$ 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

http://homepages.inf.ed.ac.uk/imurray2/teaching/oranges_and_lemons/
The Unsupervised data

width/cm
height/cm

Inf2b Learning and Data: Lecture 3  Clustering and data visualisation
Navel oranges

![Graph showing heights and widths of navel oranges. The x-axis represents width in cm, ranging from 6 to 10, and the y-axis represents height in cm, ranging from 4 to 10. The points are clustered into groups, possibly indicating different types or qualities of oranges.]
Spanish jumbo oranges

![Graph showing height and width of oranges.]
Belsan lemons

Inf2b Learning and Data: Lecture 3  Clustering and data visualisation
Some other lemons

![Graph showing height vs width of lemons]
“Selected seconds” oranges

![Graph showing data points for height and width in centimeters.](image)
A simple algorithm to find clusters:

1. Pick $K$ random points as cluster centre positions
2. Assign each point to its nearest centre*
3. Move each centre to mean of its assigned points
4. If centres moved, goto 2.

* 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
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:

\[
E = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} z_{kn} \| x_n - m_k \|^2
\]

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.
Theory of $K$-means clustering

- If assignments don’t change, algorithm terminates.

- **Can assignments cycle, never terminating?**

- **Convergence proof technique:** find a *Lyapunov function* $\mathcal{L}$, that is bounded below and cannot increase. $\mathcal{L} = \text{sum of square distances between points and centres}$

  NB: $E^{(t+1)} \leq E^{(t)}$

- **$K$-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?!
Failures of $K$-means (e.g. 1)

Large clouds pull small clusters off-centre
Distance needs to be measured sensibly.
$K$-means clustering is not the only method for clustering data

See:
http://en.wikipedia.org/wiki/Cluster_analysis
Hierarchical clustering (NE)

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
Bottom-up clustering of the lemon/orange data

Hierarchical clustering (centroid–distance)
Stanford Racing Team; DARPA 2005 challenge

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

It would look pretty stupid to run off the road, just because the trip planner said so.
How to stay on the road?

Classifying road seems hard. Colours and textures change: road appearance in one place may match ditches elsewhere.
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 of data for visualisation.

Project each sample in 3D onto a 2D plane.
Orthogonal projection of data onto an axis

\[ y = \|x\| \cos \theta = u^T x \]
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: \( \max_u \text{Var} (y) \)

\[
\text{Var} (y) = \frac{1}{N-1} \sum_{n=1}^{N} (y_n - \bar{y})^2
\]

cf. least squares fitting (linear regression)
Principal Component Analysis (PCA)

- Mapping $D$-dimensional data to a principal component axis $\mathbf{u} = (u_1, \ldots, u_D)^T$ that maximises $\text{Var}(y)$:
  
  $$y_n = \mathbf{u}^T \mathbf{x}_n = u_1 x_{n1} + \cdots + u_D x_{nD}$$

  NB: $||\mathbf{u}|| = 1$

- $\mathbf{u}$ is given as the eigen vector with the largest eigen value of the covariance matrix, $\mathbf{S}$:
  
  $$\mathbf{S} = \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T, \quad \bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$$

- Eigen values $\lambda_i$ and eigen vectors $\mathbf{p}_i$ of $\mathbf{S}$:
  
  $$\mathbf{S} \mathbf{p}_i = \lambda_i \mathbf{p}_i, \quad i = 1, \ldots, D$$

  If $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$, then $\mathbf{u} = \mathbf{p}_1$

  NB: $\mathbf{p}_i^T \mathbf{p}_j = 0$, i.e. $\mathbf{p}_i \perp \mathbf{p}_j$ for $i \neq j$

  $\mathbf{p}_i$ is assumed to have been normalised so that $||\mathbf{p}_i|| = 1$. 

Let $\mathbf{v} = \mathbf{p}_2$, i.e. the eigen vector for the second largest eigenvalues, $\lambda_2$

Map $\mathbf{x}_n$ on to the axis by $\mathbf{v}$:

$$z_n = \mathbf{v}^T \mathbf{x}_n = v_1 x_{n1} + \cdots + v_D x_{nD}$$

Point $(y_n, z_n)^T$ in $\mathcal{R}^2$ is the projection of $\mathbf{x}_n \in \mathcal{R}^D$ on the 2D plane spanned by $\mathbf{u}$ and $\mathbf{v}$.

Can be generalised to a mapping from $\mathcal{R}^D$ to $\mathcal{R}^\ell$ using $\{\mathbf{p}_1, \ldots, \mathbf{p}_\ell\}$, where $\ell < D$.

NB: Dimensionality reduction may involve loss of information. Some information will be lost if $\sum_{i=1}^\ell \lambda_i < \sum_{i=1}^D \lambda_i$. 
Covariance matrix

\[ S = \begin{pmatrix} s_{11} & \cdots & s_{1D} \\ \vdots & \ddots & \vdots \\ s_{D1} & \cdots & s_{DD} \end{pmatrix} \quad \cdots \ D\text{-by-}D \text{ symmetric matrix} \]

- In scalar representation:
  \[ s_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)(x_{nj} - \bar{x}_j), \quad \bar{x}_i = \frac{1}{N} \sum_{n=1}^{N} x_{ni} \]

- Relation with Pearson’s correlation coefficient:
  \[ r_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} \left( \frac{x_{ni} - \bar{x}_i}{s_i} \right) \left( \frac{x_{nj} - \bar{x}_j}{s_j} \right) \]
  \[ = \frac{1}{s_is_j} \frac{1}{N-1} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)(x_{nj} - \bar{x}_j) \]
  \[ = \frac{s_{ij}}{\sqrt{s_{ii}s_{jj}}} \quad \text{cf:} \quad s_i = \sqrt{s_{ii}} = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (x_{ni} - \bar{x}_i)^2} \]
PCA on the film review toy data

<table>
<thead>
<tr>
<th></th>
<th>Australia</th>
<th>Body of Lies</th>
<th>Burn After</th>
<th>Hancock</th>
<th>Milk</th>
<th>Rev Road</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denby</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>McCarthy</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>M'stern</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Puig</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Travers</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Turan</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

\[
S = \begin{pmatrix}
2.66 & -1.07 & 0.53 & -4.67 & -1.20 & -0.67 \\
-1.07 & 1.47 & 1.07 & 3.27 & 0.60 & 1.27 \\
0.53 & 1.07 & 3.47 & 0.67 & 0.20 & 1.87 \\
-4.67 & 3.27 & 0.67 & 10.97 & 2.30 & 3.67 \\
-1.20 & 0.60 & 0.20 & 2.30 & 1.10 & 0.60 \\
-0.67 & 1.27 & 1.87 & 3.67 & 0.60 & 3.07
\end{pmatrix}
\]

\[
P = \begin{pmatrix}
-0.341 & 0.345 & 0.326 & -0.180 & 0.603 & -0.512 \\
0.255 & 0.151 & -0.240 & -0.548 & 0.496 & 0.554 \\
0.101 & 0.786 & -0.503 & 0.028 & -0.280 & -0.198 \\
0.827 & -0.154 & 0.096 & -0.182 & 0.025 & -0.450 \\
0.181 & -0.065 & -0.341 & 0.733 & 0.556 & 0.015 \\
0.304 & 0.461 & 0.676 & 0.309 & -0.047 & 0.375
\end{pmatrix}
\]

\[
Q = \begin{pmatrix}
15.8 & 0 & 0 & 0 & 0 & 0 \\
0 & 4.85 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.13 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.634 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.288 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

where \( P = (p_1, \ldots, p_6) \) and \( (Q)_{ii} = \lambda_i \) for \( i = 1, \ldots, 6 \).
PCA on the film review toy data (cont.)

![PCA plot](chart)

- 1st principal component
- 2nd principal component

- Denby
- McCarthy
- Morgenstern
- Puig
- Travers
- Turan

**Inf2b Learning and Data: Lecture 3**
**Clustering and data visualisation**
Dimensionality reduction $D \to \ell$ by PCA

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

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

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

$y = (y_1, \ldots, y_\ell)^T$ : $\ell$-dimensional vector

$A = (p_1, \ldots, p_\ell)$ : $D \times \ell$ 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 4 in Week 4, Lab 6 in Week 6)
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.
The fix: clusters have shapes as well as centres:

Assume each point is from one of $K$ Gaussian distributions

Just like $K$-means, but:

Assign points to Gaussian assigning highest probability.

Update cluster with mean and variance of points it owns.

Fancier (usual) version: points have soft assignments in proportion to their probability under each cluster.
Each cluster $k \in \{1 \ldots K\}$ has fitted a model $P(x \mid c = k)$. 

$$P(c = k \mid x) = \frac{P(x \mid c = k) P(c = k)}{P(x)} \propto P(x \mid c = k) P(c = k)$$
The model is called a mixture of Gaussians
The algorithm is called EM (Expectation Maximisation)*
EM maximises $P(\text{data}|\text{fitted model})$
Does EM converge?

* EM is a general method to maximise likelihoods of probabilistic models with latent variables, e.g. cluster assignments.
The clustering on the right has much higher probability than the $k$-means solution on the left.