### 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

### Clustering

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.

### Recommended reading:

- David MacKay textbook, p284–
- [http://www.inference.phy.cam.ac.uk/mackay/itila/](http://www.inference.phy.cam.ac.uk/mackay/itila/)

### 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](http://dx.doi.org/10.1093/bioinformatics/btr246)
- Document clustering
- Thesaurus generation

### Visualisation of film review users


C ≈ 1000 users, M ≈ 1700 movies

2D plot of users based on rating similarity

### A two-dimensional space

- Oranges: height/cm, width/cm
- Lemons: height/cm, width/cm


### The Unsupervised data

- Manderins: height/cm, width/cm

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. Repeat until assignments don’t change

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.

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 $k$.

$E$ is common.

### Theory of $K$-means clustering

- If assignments don’t change, algorithm terminates.
- Can assignments cycle, never terminating?
- **Convergence proof technique**: find a Lyapunov function $L$, that is bounded below and cannot increase. $L = \text{sum of square distances between points and centres}$

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

- $K$-means is an optimisation algorithm for $L$. **Local optima** are found, i.e. there is no guarantee of finding global optimum. Running multiple times and using the solution with best $L$ 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 the sum-squared error decrease as $K$ increases.]

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

- Large clouds pull small clusters off-centre

![Plot showing large clouds pulling small clusters off-centre.]

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

- Distance needs to be measured sensibly.

![Plot showing differences in distances.]

Clustering clustering methods

- $K$-means clustering is not the only method for clustering data

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

Bottom-up clustering of the lemon/orange data

- ![Dendrogram representation.]

Stanley

Stanford Racing Team; DARPA 2005 challenge


Inside Stanley

Perception and intelligence

- It would look pretty stupid to run off the road, just because the trip planner said so.
Dimensionality reduction and data visualisation

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

Optimal projection of 2D data onto 1D

- Mapping D-dimensional data to a principal component axis \( u = (u_1, \ldots, u_D)^T \) that maximises \( \text{Var}(y) \):
  \[
y_i = u^T x_i = u_1 x_{i1} + \cdots + u_D x_{iD}.
\]
- \( u \) is given as the eigenvector with the largest eigenvalue of the covariance matrix, \( S \):
  \[
  S = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T, \quad \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n
  \]
- Eigen values \( \lambda_i \) and eigenvectors \( p_i \) of \( S \):
  \[
  S p_i = \lambda_i p_i, \quad i = 1, \ldots, D
  \]
- If \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_6 \), then \( u = p_1 \) and \( \text{Var}(y) = \lambda_1 \)

Principal Component Analysis (PCA)

- Let \( v = p_2 \), i.e. the eigenvector for the second largest eigen values, \( \lambda_2 \)
- Map \( x_n \) on to the axis by \( v \):
  \[
  z_{i2} = v^T x_i = v_1 x_{i1} + \cdots + v_D x_{iD}
  \]

Orthogonal projection of data onto an axis

- Mapping 2D to 1D: \( y_i = u^T x_i = u_1 x_{i1} + u_2 x_{i2} \)
- Optimal mapping: \( \text{max Var}(y) \)
  \[
  \text{Var}(y) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \bar{y})^2
  \]
- cf. least squares fitting (linear regression)

Covariance matrix

- In scalar representation:
  \[
  s_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} (x_{n,i} - \bar{x}_i)(x_{n,j} - \bar{x}_j)
  \]
- Relation with Pearson’s correlation coefficient:
  \[
  r_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} \frac{(x_{n,i} - \bar{x}_i)(x_{n,j} - \bar{x}_j)}{\sqrt{N-1} \sum_{n=1}^{N} (x_{n,i} - \bar{x}_i)^2}
  \]

Principal Component Analysis (PCA) (cont.)

- Point \( (y_2, z_2)^T \) in \( \mathbb{R}^2 \) is the projection of \( x_n \in \mathbb{R}^D \) on the 2D plane spanned by \( u \) and \( v \):
  \[
  \text{Var}(y) = \lambda_1, \quad \text{Var}(z) = \lambda_2
  \]
- Can be generalised a mapping from \( \mathbb{R}^D \) to \( \mathbb{R}^l \) using \( \{p_1, \ldots, p_l\} \), where \( l < D \).
- NB: Dimensionality reduction may involve loss of information. Some information will be lost if
  \[
  \sum_{i=1}^{l} \frac{\lambda_i}{\sum_{i=1}^{D} \lambda_i} < 1
  \]

PCA on the film review toy data

| Australia | Body of | Burn | Rev | Australia | Body of | Burn | Rev | Australia | Body of | Burn | Rev | Australia | Body of | Burn | Rev | Australia | Body of | Burn | Rev | Australia | Body of | Burn | Rev |
|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|-----------|---------|------|-----|
| Unitby    | 2       | 3    | 4    | 5        | 6       | 7    | 8    | Turan     | 2       | 3    | 4    | 5        | 6       | 7    | 8    | McCarthy  | 7       | 5    | 5    | 7        | 5       | 5    | 5    | 7        | 5       | 5    | 5    |
| McCarthy  | 7       | 5    | 5    | 7        | 5       | 5    | 5    | Turan     | 2       | 3    | 4    | 5        | 6       | 7    | 8    | McCarthy  | 7       | 5    | 5    | 7        | 5       | 5    | 5    | 7        | 5       | 5    | 5    |
| Unitby    | 2.66    | 1.07 | 0.57 | 1.60     | 0.86    | 0.45 | 0.34 | Unitby    | 0.94    | 0.34 | 0.26 | 0.180   | 0.63     | 0.512 |
| McCarthy  | 0.84    | 0.50 | 0.26 | 0.44     | 0.10    | 0.06 | 0.03 | McCarthy  | 1.00    | 0.10 | 0.08 | 0.02     | 0.08     | 0.564 |
| Australia | 0.82    | 0.54 | 0.10 | 0.10     | 0.55    | 0.33 | 0.30 | Australia | 0.67    | 0.41 | 0.25 | 0.12     | 0.25     | 0.493 |
| Body of   | 0.81    | 0.47 | 0.15 | 0.07     | 0.55    | 0.30 | 0.55 | Body of   | 1.18    | 0.50 | 0.25 | 0.12     | 0.25     | 0.493 |
| Burn      | 0.81    | 0.47 | 0.15 | 0.07     | 0.55    | 0.30 | 0.55 | Burn      | 1.18    | 0.50 | 0.25 | 0.12     | 0.25     | 0.493 |
| Rev       | 0.81    | 0.47 | 0.15 | 0.07     | 0.55    | 0.30 | 0.55 | Rev       | 1.18    | 0.50 | 0.25 | 0.12     | 0.25     | 0.493 |

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

Dimensionality reduction $D \rightarrow \ell$ by PCA

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

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

$$y = A^T x$$

- linear transformation from $R^D$ 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. $y = AT(x - \bar{x})$.

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 in Week 4)

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 eigenvalue 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.