https://github.com/tedinburgh/ads2023

Cluster evaluation, issues and outliers

Tom Edinburgh
te269

Today: clustering and outliers

- Clustering evaluation (internal and external)
- Choosing the number of clusters
- Consensus clustering
- Outliers
- Imputation (briefly!)
- Questions: halfway through, at the end, or by email (te269)

Resources

- Slides adapted from:
  - Introduction to Statistical Learning with Python, Chapter 12

Internal vs external evaluation

- Validation of clustering is challenging
- There is no universal approach
- Internal evaluation: summarise the clusters using a quality score
- External evaluation: compare clusters to ‘ground truth’ class labels

Internal vs external evaluation

- Validation of clustering is challenging
- Internal evaluation: summarise the clusters using a quality score
  - Optimising a function over clusters doesn't necessarily say how useful the clustering is
- External evaluation: compare clusters to 'ground truth' class labels
  - Unlikely that class labels exists, but if they do then why cluster?
  - Class labels are one data partition, but is it necessarily the best clustering?
- Good clusters are subjective but some approaches help to identify bad clusters

Internal evaluation

- Evaluate the quality of the clustering based on the data that was clustered
- An internal criterion will typically involve measuring similarity between observations within a cluster and between observations between clusters
- This will bias towards clustering methods that use the same notion of similarity
- Some methods and some evaluation criteria make assumptions about how the data is clustered

- Can usually use these methods to identify the ‘optimal’ number of clusters

Silhouette coefficient

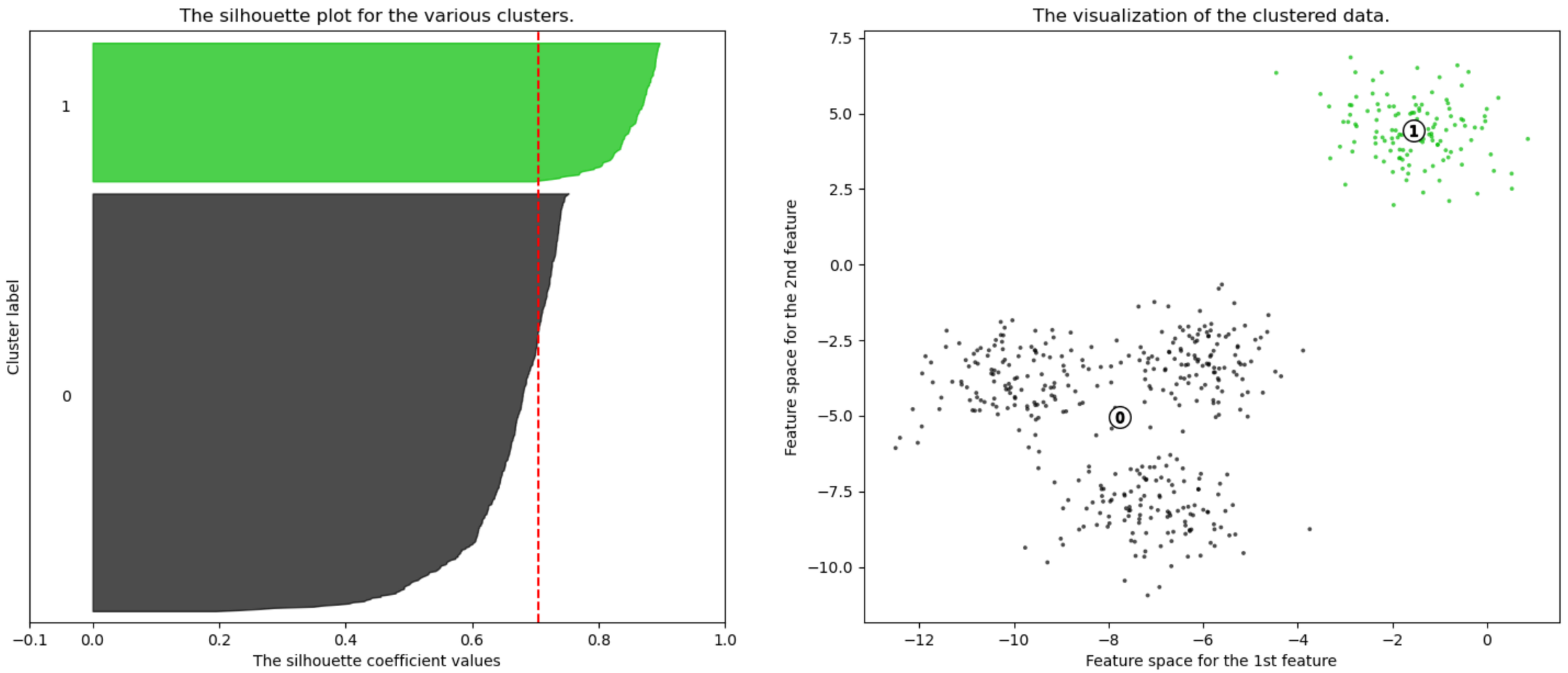
- Gives each observation a value between -1 and +1 for how similar it is to its own cluster (cohesion) compared to its next closest cluster (separation)

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i,\ i \neq j} d(i,j), \qquad b_i = \min_{j \neq i} \frac{1}{|C_j|} \sum_{k \in C_j} d(i,k)$$

- The silhouette value for the point indexed by i is $s_i = \begin{cases} 1 - a_i/b_i & a_i < b_i \\ 0 & a_i = b_i \\ -1 + b_i/a_i & a_i > b_i \end{cases}$

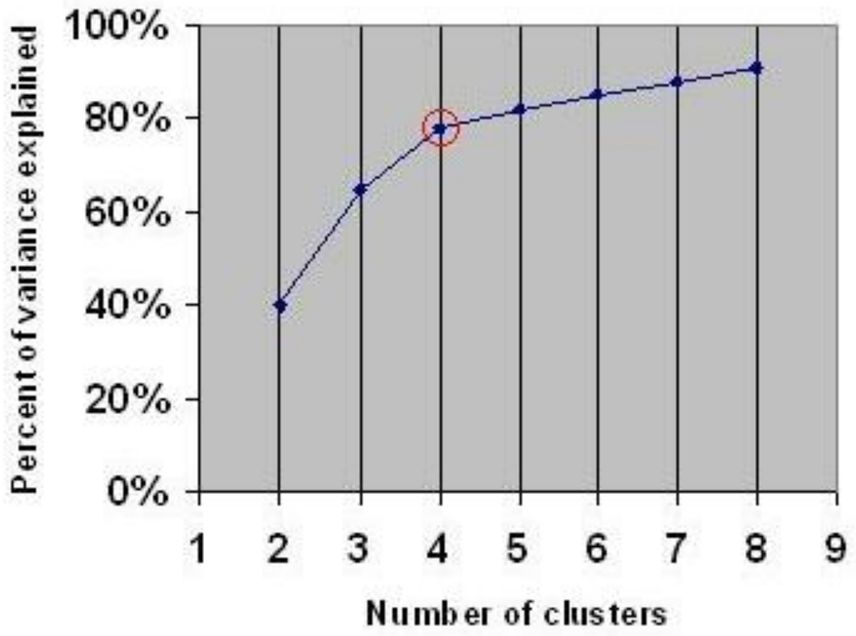
- The overall silhouette coefficient is the mean of s_i, which can be optimised over the number of clusters K

Silhouette coefficient



Elbow method

- Evaluate the **percentage of explained variance** (the ratio of between-cluster variance to the total variance) for clusterings with multiple values of K
- Choose the number of clusters so that adding an extra cluster doesn't explain much more of the total variance (this is often ambiguous though!)



Information criteria

- If there is a likelihood function for the clustering model, then we can use the Akaike information criterion or Bayesian information criterion
- This works for Gaussian mixture models
- $AIC = 2k - 2\log(\hat{L})$, where k is the number of estimated parameters

Gap statistics

- Compare the total within-cluster variation with expected within-cluster variation under a reference null data (matching characteristics of the data but from a distribution with no obvious clustering)
- Can be used with any clustering algorithm
- Minimise a quantity called the gap statistic over different values of k

J. R. Statist. Soc. B (2001)
63, *Part* 2, *pp.* 411–423

Estimating the number of clusters in a data set via the gap statistic

Robert Tibshirani, Guenther Walther and Trevor Hastie

Stanford University, USA

[Received February 2000. Final revision November 2000]

More options

- Dunn index is the ratio between the smallest between-cluster distance and the largest within-cluster distance, it looks for dense, well-separated clusters
- Some definitions from the graph-based clustering define similarity within and between clusters (using a distance metric), and can be used more widely

External evaluation

- Relies on external **benchmark** / **gold standard** class labels
- Classes may have internal structure (i.e. unknown clusters within the classes)
- Evaluating clustering becomes similar to evaluating (supervised) classification
- Reproducing known labels is not useful if the task is knowledge-discovery

Purity

- Similar to classification decision trees
- Assign each cluster C_k to the class label l_j most frequent in the cluster
- Average the number of correctly assigned observations
- purity$(C, L) = 1/n \sum_k \max_j | C_k \cap l_j |$
- This doesn't penalise the number of clusters, so can achieve a maximum value of 1 when each observation is in it's own cluster
- Also performs poorly for highly imbalanced datasets

Rand index and F-measure

- Positive/negative results defined over pairwise assignment (for a total of $n(n-1)/2$ pairs)
- Rand index is $\frac{TP + TN}{TP + TN + FP + FN}$
- F-measure is $\frac{(1+\beta)^2 TP}{(1+\beta)^2 TP + \beta^2 FN + FP}$

	i and j in same cluster	**i and j in different clusters**
i and j in same class	True positive	False negative
i and j in different classes	False positive	True negative

- Rand index weights false positives and false negatives equally, which is usually not ideal
- F-measure balances precision and recall

Issues with clustering

Issues with clustering

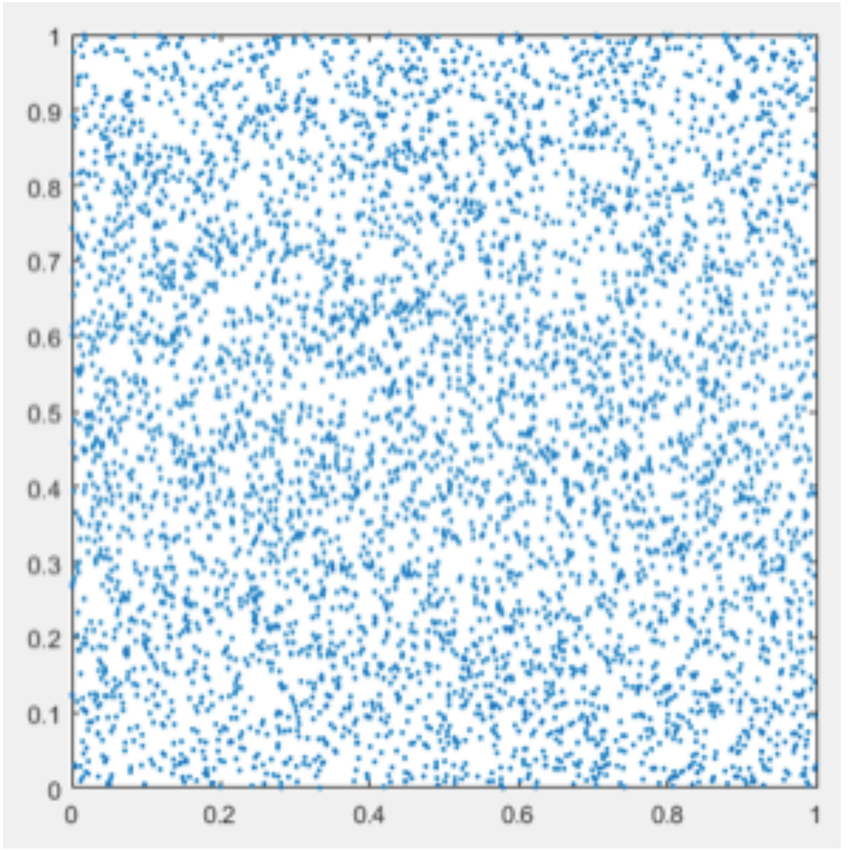
- Clustering can be very sensitive to choices that you make e.g. hyperparameters, choice of distance metric
- Calculating distances between pairwise observations is very expensive
- Interpretation of clustering algorithms can be subjective

Issues with clustering

- Clustering can be very sensitive to choices that you make e.g. hyperparameters, choice of distance metric
- Calculating distances between pairwise observations is very expensive
- Interpretation of clustering algorithms can be subjective
- Hard clustering algorithms force each observation to be part of a cluster, so the presence of outliers can distort the cluster
- Clustering is not very robust to perturbation, e.g. if you cluster n observations, the remove a subset of these completely at random and repeat the clustering, you may find a very different clustering!

Issues with clustering

- **Warning**: clustering methods can find structure where there isn't actually any. We should be wary of making strong conclusions about the output of clustering methods!
- Are you just finding clusters in noise?



Issues with clustering

- A big problem with high-dimensional data is the **curse of dimensionality**
- This refers to various issues with analysing data in high-dimensions, because the data becomes sparse
- There isn’t much difference in the Euclidean distances between pairs of points in a high-dimensional space
  - This relates to the fact that a hypersphere has a much smaller volume than a hypercube of the same radius

Consensus clustering

- Similar to ensembles in supervised learning
- The idea is to reduce variability in clusterings (e.g. from initialisation)
- Various methods for combining multiple runs of a clustering algorithm
- Monti consensus clustering: create matrices of the proportion of the runs that each pair of observations clustered together (for multiple datasets with perturbations to the data) and calculate the cumulative distribution function
- Cluster aggregation could involve collecting soft clusterings as posterior probability distributions and measuring the divergence from a reference distribution

Summary of clustering

Summary of clustering

- We want to find a ‘natural’ grouping of n observations into K clusters
- Observations in the same (/a different) cluster should be similar (/dissimilar)
- We might assume data was generated by separate processes and (without necessarily describing the processes), we want to identify observations from the same process

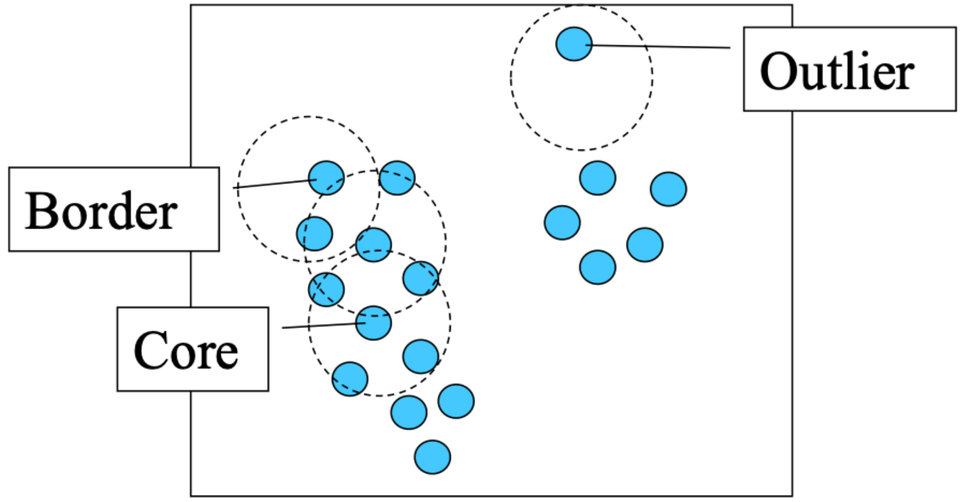
Summary of clustering

	Cluster types	Method	Fixed number of clusters
k-means	Hard	Partitioning	Yes
Fuzzy c-means	Soft	Partitioning	Yes
Hierarchical	Hard	Agglomerative	No
Gaussian mixture models	Soft	Partitioning	Yes
Density-based	Hard	Agglomerative	No
Graph-based	Hard	Partitioning	Yes
Spectral	Hard	Partitioning	Yes

Questions?

Recap: DBSCAN definitions

- Parameters: minPts and ϵ (the radius of a neighbourhood around each point)
- p is a **core point** if there are at least minPts within the ϵ neighbourhood around p
- q is **directly reachable from** p if q is within the ϵ neighbourhood around the core point p. If q is not a core point itself, it is a **border point**
- q is **reachable from** p if there is a path $p_1, \ldots, p_n$ of core points where
  - each p_{k+1} is directly reachable from p_k,
  - p_1 is directly reachable from p,
  - q is directly reachable from p_n
- All other points are **outliers**



Outliers

- What is an outlier?
- When might outliers be important?
- What is the difference between noise and outliers?

Outliers

- Outliers are data points that are considerably different from the remainder of the data
- Naturally occurring outliers do occur but are relatively rare
- They are usually either important or a nuisance (e.g. rare diseases, decimal errors)
- Label error, e.g. images of dogs but with a few cats included by accident
- Noise is generally not very interesting (not unusual values)

How can we identify outliers?

- Model-based:
  - Outliers are points that don’t fit the model very well or distort the model
  - Points far away from cluster centres or small clusters may be outliers
- Data-based:
  - Identify directly from the data without a model e.g. density-based

- What assumptions might we make about outliers?
- How do outliers relate to statistical significance/hypothesis testing?

Local outlier factor

- This is based on local density, similar concepts to to DBSCAN
- Identify points that have a much lower density, these are outliers
- LOF uses k-nearest neighbour distances rather than ϵ-neighbourhoods
- The reachability distance between points p and q is

  $\text{rd}_k(p, q) = \max(r_k(q), d(p, q))$, where r_k is the distance from q to its k^{th} -nearest neighbour
- The local reachability density of p is the average reachability distance of p from its neighbours (from, not to)

Local outlier factor

- The local outlier factor (LOF) compares the local reachability density (LRD) of p to the LRD of its k-nearest neighbours
- An LOF approximately 1 means a similar density to the neighbours
- An LOF < 1 means a higher density than the neighbours (an **inlier**)
- An LOF > 1 means a lower density than the neighbours (an **outlier**)

Missing data and imputation

Missing data and imputation

- How do you replace missing data (or ‘wrong’ data) with substituted values?
- There are various types of missing data:
  - Missing completely at random (MCAR): the reason for any data being missing data is independent from all variables, so introduces no bias
  - Missing at random (MAR): we can usually account for the bias, e.g. the missing data is related to a predictor variable that itself is fully recorded
  - Missing not at random (MNAR): the value of the missing data is related to the reason for it being missing

Missing data and imputation

- How do you replace missing data (or ‘wrong’ data) with substituted values?
- Imputation can be static (e.g. for time-series data, the last observation carried forward) or model-based
- If you have a generative model, you may be able sample from the model distribution to fill in the missing data (e.g. conditional on the available data)
- Other strategies for missing data include omission/partial deletion and using methods that are unaffected by the missing values
- Multiple imputation helps reduce variance introduced through imputation, by creating multiple datasets with different imputed values (similar to ensembles)

Questions?

- Feel free to email me at te269@cam.ac.uk

Example class 4

- Problem sheet is on ADS course GitLab page and Moodle
- k-means
- Vector quantisation using k-means

Coursework

- Assignment is on ADS course GitLab page
- Section A
  - Q1: PCA and k-means
  - Q2: Missing labels and duplicated observations
  - Q3: Missing data, imputation and outliers
- Section B
  - Q4: Random forests and supervised learning
  - Q5: Clustering (unsupervised learning)

Next time (Miles)

- Introduction to Neural Networks