Applied Data Science

Xinyu Zhong
Queens' College

January 23, 2024

Contents

Abstract

Non-examinable: Linear discriminant analysis,SVMs, Boosting and bagging, random forests, hierarchical divisive clustering, kernel PCA, ISOMAP, t-SNE, Hartigan Wong (from k-means),self-organising maps, HCS algorithm (from graph clustering), local outlier factor.

1 Understanding the structure of Data

Simpson's paradox:

Simpson's paradox is observed in probability and statistics; a trend appears in several groups of data but disappears or reverses when the groups are combined.

1.1 Adjusting data without tampering with signal

1.1.1 Expression ranges and One-hot encoding

For categorical data, we assign value by using one-hot encode which are the n unit vectors, which are equal distance from each other, from n-dimensional space to represent n categories

1.1.2 Standardisation vs scaling

There are different ways of scaling, the most straightforward one is linear scaling, others include log scaling. Note that outliers will seriously affect scaling, it can be done by cap our value, however, it also means that we throw away data points Note that we lose certain information when standardise, i.e. for Z

$$Z = \frac{x_i - \mu}{\sigma}$$

you will lose the standard deviation. Also, μ and σ can be influenced by outliers. Median and MAD(median absolute deviation)are not affected by outliers

$$Z_{\text{med}} = \frac{x_i - \text{ median}}{MAD} \tag{1}$$

$$MAD = \text{ median } (|\, x_i - \text{ median } |) \tag{2}$$

1.1.3 Near zero variance

Near zero variance means that the variable has little variance i.e. almost constant

1.1.4 Multi-collinearity

Multi-collinearity is a concept where independent variables are highly correlated. i.e. correlation coefficient = 1. We use PCA analysis to reduce the dimension of the highly correlated variable space

1.1.5 Missing Data

There are three types of missing data:

1. Missing completely at random (MCAR): the reason for any data being missing data is independent from all variables, so introduces no bias
2. 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
3. Missing not at random (MNAR): the value of the missing data is related to the reason for it being missing

1.1.6 Imputation

Imputation is the process of replacing missing data with substituted values. Common imputation methods include:

- Static imputation: replace missing data with a constant
- Generative imputation: use a model to predict the missing data
- Omission: remove the data point with missing data
- Multiple imputation: generate multiple imputed datasets and combine them

1.1.7 Dimensionality reduction

Dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables. It can be done by:

1.2 Engineering Model Robustness

- we draw the samples independently and identically (iid) at random from the distribution (there is no underlying structure that is present in the data)
- the sets are disjunct partitions of the original distribution (no intersection between training set and test set) the size of the validation and test sets should be comparable (if not identical)
- The validation set should be large enough to detect differences between models
- accuracy is not the only metric to measure the performance of the model
- on the test set, the error between the prediction and the actual label is the test error
- the objective function of the algorithm minimizes the test errors by parameter tuning
- Models are further evaluated for Bias and Variance (assessment of overfitting/underfitting)

1.2.1 Validation set

- K-fold validation is used to tune the hyperparameters of the model
- Leave-one-out cross-validation (LOOCV) is a special case of k-fold validation, where k = n
- Data is split into training, validation and test sets

1.2.2 Confusion matrix

- True positive (TP): correctly predicted positive
- False positive (FP): incorrectly predicted positive
- True negative (TN): correctly predicted negative
- False negative (FN): incorrectly predicted negative

1.2.3 Unbalanced data

Unbalance data should be fixed by up-weighting or down-sampling
Nyquist–Shannon sampling theorem:

> The theorem states that a function $x(t)$ that contains no frequencies higher than B hertz is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2B}$ seconds apart.

Kullback-Leiber divergence (per classes or using a binning approach for continuous data) :

2 Supervised Learning: Regression

We assume the model

$$Y = \beta_0 + \beta_1 X + \epsilon$$

where β_0 and β_1 are two unknown constants that represent the intercept and the slope; β_0 and β_1 are also known as coefficients or parameters and ϵ is the error term.

Given the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model coefficients, we predict the output, $\hat{y}$ using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

where $\hat{y}$ indicates the prediction of Y on the basis of $X = x$. The hat symbol denotes an estimated value. By minimising the sum of squared errors (SSE), we can find the estimated coefficients:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

and

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

2.1 Standard errors

The standard error of an estimator reflects how it varies under repeated subsampling.

$$SE\left(\hat{\beta}_1\right)^2 = \frac{\sigma^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

and

$$SE\left(\hat{\beta}_0\right)^2 = \sigma^2 \left\{ \frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \right\}$$

where $\sigma^2 = \mathrm{Var}(\epsilon)$.

2.2 Hypothesis Testing

Standard errors can be used to perform hypothesis testing on the coefficients. The most common hypothesis test involves testing the null hypothesis of: H_0 : There is no relationship between X and Y H_1 : There is some relationship between X and Y Or, more formally:

$$\begin{aligned} H_0 : &\quad \beta_1 = 0 \\ H_1 : &\quad \beta_1 \neq 0 \end{aligned}$$

To test the H_0 we compute a t-statistic:

$$t = \frac{\hat{\beta}_1 - 0}{SE\left(\hat{\beta}_1\right)}$$

This will be a t-distribution with $n - 2$ degrees of freedom. Using R, we can compute the probability of observing any value $\geq |t|$. We call this probability **p**-value.

2.3 Accuracy

Residual standard error:

$$RSE = \sqrt{\frac{1}{n-2}RSS}$$

where the residual sum of squares is $RSS = \sum_{i=1}^{n} (y_i - \hat{y_1})^2$
R^2,fraction of variance explained is

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

where $TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$ is the total sum of squares.

2.4 Correlation

$$\text{Cor}(X, Y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

and R^2 is $R^2 = \text{Cor}(X, Y)^2$.

2.5 Type of Loss Functions

- Mean Absolute Error (MAE) (L1 Loss)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- Mean Squared Error (MSE) (L2 Loss)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Mean Biased Error (MBE)

$$\text{MBE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)$$

  It is lesser used as the positive and negative errors cancel each other out.

- Hubber Loss (L1-L2 Loss)

$$L_\delta == \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{if } |y - f(x)| < \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases}$$

Note that L1 loss is more robust to outliers than L2 loss, as L2 loss squares the error term. However, it means that L1 is less stable Another consideration when choosing the loss function is the differentiability of the loss function.

2.6 Bias and Variance Trade-off

Bias: the model's error rate on the training set (rephrased, the difference between the average prediction and the correct value we are predicting).

A model with high bias is oversimplified (insufficient information acquired from the training data).

Variance: the model's error rate on the validation (or test) set, in addition to the bias

A model with high variance captures the signal and the noise in the training data and fails to generalise well on (unseen) test data.

2.7 Bias/variance decomposition

On a model M built on an output variable y, and X predictors

$$y = f(X) + \epsilon$$

Where ϵ is the error term, expected to be normally distributed with mean 0 . The expected error on Y is

$$\text{Err}(Y) = E\left[(Y - f(\hat{X}))^2\right]$$

The $\text{Err}(Y)$ can be further decomposed as:

$$\text{Err}(Y) = E[f(\hat{X}) - f(X)]^2 + E\left[(f(\hat{X}) - E[f(\hat{X})])^2\right] + \sigma_\epsilon^2$$

$\text{Err}(Y)$ is the sum of squared bias, variance and irreducible error.

2.8 Multiple (linear) regression. Model selection

How to choose which subsets to use, for n features, there are 2^n subsets, which is not feasible to try all of them.

2.9 Model selection.

Forward Selection

1. Start with the null model, a model that contains an intercept but no predictors
2. Fit p simple linear regressions, each with only one feature and add to the null model the variable that results in the lowest RSS.
3. Add to that model the variable that results in the lowest RSS amongst all two-variable models.
4. Continue until some stopping rule is satisfied, for example when all remaining variables have a p-value above some threshold

backward selection

1. Start with all variables in the model; fit the model
2. Remove the variable with the largest p-value i.e. the variable that is the least statistically significant
3. The new $p-1$-variable model is fit, and the variable with the largest p-value is removed.
4. Continue until a stopping rule is reached e.g. when all remaining variables have a significant p-value above some threshold

2.10 Regularisation

Regularisation is the process of adjusting an algorithm to prefer a smaller model, to avoid overfitting. This is done by modifiying the loss function to include a penalty for large weights.

Regularization: modifying the loss function to penalize large weights.

$$\hat{Y} = W \cdot X$$

The 'size' [L_2 norm] of weights:

$$\|w\| = \sqrt{\sum_{i=1}^{n} w_i^2}$$

- Lasso Regression (L1 Regularisation)

$$\text{Loss} = \text{MSE} + \alpha \sum_{i=1}^{n} |w_i|$$

- Ridge Regression (L2 Regularisation)

$$\text{Loss} = \text{MSE} + \alpha \sum_{i=1}^{n} w_i^2$$

- Elastic Net Regression (L1-L2 Regularisation)

$$\text{Loss} = \text{MSE} + \alpha \sum_{i=1}^{n} \left(\rho w_i^2 + (1-\rho)\,|w_i|\right)$$

The advantage of Lasso is that it can shrink some coefficients to zero, which is useful for feature selection. However, it is not differentiable at zero, which makes it difficult to optimise. Ridge regression is differentiable everywhere, but it cannot shrink coefficients to zero. Elastic net regression is a compromise between Lasso and Ridge regression.

3 K nearest neighbour

K nearest neighbour works by finding the k nearest neighbours of a data point, and make prediction based on the labels of the k nearest neighbours. The distance between two data points can be defined using different distance metrics, such as Euclidean distance, Manhattan distance, etc.

- KNN is a non-parametric method, which means that it does not make any assumptions about the underlying data distribution.
- KNN distance can be defined with Euclidean distance, Manhattan distance, etc.

$$d_{eucleadain}\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2}$$

$$d_{manhatten}\left(x_i, x_j\right) = \sum_{k=1}^{p} |x_{ik} - x_{jk}|$$

- Lower K means a more complex model (Prone to overfitting)
- Higher K means a less complex model (Prone to under fitting). In the most extreme case, K = N, then your model would predict the same outcome for all data points.

3.1 Pros and cons

4 Logistic Regression

4.1 Basics of Logistic Regression

- Input: continuous or categorical
- Output: categorical

- Logistic function:
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$
- The fitting method is maximum likelihood
- The likelihood function is:
$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'}))$$
- We can manipulate the logistic function to get the logit function:
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$
- The logit function is a linear function of X
- In the case of multiple predictors, the logit function is:
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$
where p is the number of predictors.
- In the case of multiple classes, we can use one-vs-all method to classify the data.
- It is done by select a class as the baseline, and compare the probability of the other classes with the baseline class.
$$P(Y=j|X=x_0) = \frac{e^{\beta_{0j}+\beta_{1j}x_0}}{1+\sum_{l=1}^{K-1} e^{\beta_{0l}+\beta_{1l}x_0}}$$
$$P(Y=K|X=x_0) = \frac{1}{1+\sum_{l=1}^{K-1} e^{\beta_{0l}+\beta_{1l}x_0}}$$
and here K is the baseline class.

5 Generative vs Discriminative Models

Generative models model the joint probability distribution $P(X,Y)$, while discriminative models model the conditional probability distribution $P(Y|X)$.

5.1 Example of Generative Model

- Naive Bayes
- Linear Discriminant Analysis (non-examinable)

5.2 Example of Discriminative Model

- Logistic Regression
- Support Vector Machine (non-examinable)

6 Support Vector Machine(non-examinable)

7 PCA

PCA is a dimensionality reduction technique. It is used to reduce the dimension of the data while preserving the most important information.

7.1 Maximal Variance

1. We want to choose a vector w so that it is as informative as possible. i.e. it maximizes the variance of the projection of data onto w.
2. Suppose that
$$a_i = w^T x_i$$
where a_i is the projection of x_i onto w.
3. The mean of the projection is
$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{1}{n}\sum_{i=1}^{n} w^T x_i = w^T \bar{x}$$
4. The variance of the projection is
$$var(a) = \frac{1}{n-1}\sum_{i=1}^{n} (a_i - \bar{a})^2 = \frac{1}{n-1}\sum_{i=1}^{n} \left(w^T x_i - w^T \bar{x}\right)^2 = \frac{1}{n-1}\sum_{i=1}^{n} w^T (x_i - \bar{x})(x_i - \bar{x})^T w = w^T Q w$$
The variance in quardratic form. Here Q is the covariance matrix of x by definition.
5. It is then maximised using Lagrange multiplier
$$\max_w (L = w^T Q w - \gamma\left(w^T w - 1\right))$$
and takes derivative with respect to w and set it to 0.
6. The optimal solution is
$$Qw = \gamma w$$
i.e. w is the eigenvector of Q, with eigenvalue γ.
7. L then is:
$$L = w^T \gamma w - \gamma\left(w^T w - 1\right) = \gamma$$
8. which is maximised when γ maximised, is the largest eigenvalue of Q.

7.2 Total Variance

- The total variance, V is the sum of the variance of each projection.
- The ith eigenvalue accounts for λ_i/V of the total variance, where λ_i is the ith eigenvalue and V is the total variance.

The total variance is the sum of the variance of each projection.

7.3 Information loss

Dimensionality reduction is a trade-off between information loss and computational efficiency.

7.4 Singular value decomposition

??

8 Non-linear dimensionality reduction

- Kernel PCA
- Multidimensional scaling
- Isomap
- t-SNE and UMAP

8.1 Multidimensional scaling

The ideal of Multidimensional scaling is that it maps high dimensional data into low-dimensional space in a way that preserves the distances between points as much as possible.

- The distance o_{ij}

  $$o_{ij} = \|x_i - x_j\|^2$$

  where x_i is the ith data point. Distance such as Euclidean distance, Manhattan distance, etc. can be used.
- The distance d_{ij} is a two dimensional distance between the ith and jth data point in the low dimensional space.
- The aim is to find the y_i such that d_ij:

  $$d_{ij} = \|y_i - y_j\|^2$$

  are as close to o_{ij} as possible.
- The closeness is measured by the stress function:

  $$\text{stress}, \mathrm{S}^2 = \frac{\sum_{i<j} (d_{ij} - o_{ij})^2}{\sum_{i<j} o_{ij}^2}$$

- The stress function is minimised using gradient descent, or other optimisation methods.
- MDS is equivalent to PCA when the distance is Euclidean distance and stress is defined as above.
- Different stress term can be used to emphasis different aspects of the data.

9 Decision Tree

A decision tree is a non-parametric method, therefore it does not require data pre-processing.

9.1 Recursive binary partitions

Decision Tree is a tree based method that segment the feature space into simple regions (rectangles in high dimenstional space) It makes prediction using average (mean, mode) of the most common class.

9.2 Characteristic of a decision tree

- branches and internal nodes: represent the decision rules
- leaf nodes (terminal nodes): represent the outcome.
- Decision tree can be trained without rescaling or standardization

9.3 How to grow a tree

How to grow the tree?

1. Partition the predictor space (i.e. all possible values of $X_1, X_2, \ldots, X_p$) into J distinct, non-overlapping regions, labelled $R_1, \ldots, R_J$
2. Each observation in a given region R_f is given the same predicted value (or class), which is the mean (or mode) of all response variables in that region

Greedy approach - regression

1. At each step, we care about the best partition at the moment without caring about future partitions.
2. We stop at each leaf as fewer than a small fixed number of observations
3. For one of the predictors X_j with a cutoff threshold c, we define two regions:

$$R_1(j,c) = \{X \mid X_j < c\} \text{ and } R_2(j,c) = \{X \mid X_j \geq c\}$$

4. An observation x_i is in region $R_1(j,c)$ if $x_i < c$ and in region $R_2(j,c)$ if $x_{ij} \geq c$
5. For the first iteration, $R_1(j,c)$ and $R_2(j,c)$ partitions the entire predictor space
6. We choose j and c to minimise the residual sum of squares, RSS:

$$\sum_{i:x_i \in R_1(j,c)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,c)} (y_i - \hat{y}_{R_2})^2$$

7. We repeat the process for $R_1(j,c)$ and $R_2(j,c)$ until a stopping criterion is reacheds

Classification Similarly to regression, classification uses Classification error, Gini index and Cross-entropy to measure the quality of a split, instead of RSS. We use $\hat{p}_{mk}$ to denote the proportion of training observations in the region $R_m(j,c)$are from the kth class.

- Classification error:

$$E_m(j,c) = 1 - \max_k(\hat{p}_{mk})$$

- Gini index:

$$G_m(j,c) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Cross-entropy:

$$H_m(j,c) = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

The total classification error or Gini index is the weighted average of the classification error or Gini index for each region.

$$\mathrm{G}(i,c) = \sum_{m=1}^{J} \frac{1}{n_m} G_m$$

9.3.1 Example: K= 2 classes

Plot G and E against p, where p is the proportion of class 1 in the region. G and E maximised at p = 0.5.

1. After each partition, new E,
$$E = E_1 \times \frac{n_1}{n} + E_2 \times \frac{n_2}{n}$$
Or equivalently, G:
$$G = G_1 \times \frac{n_1}{n} + G_2 \times \frac{n_2}{n}$$
We choose the feature that minimises E or G the most to partition first.

9.3.2 Entropy and Information gain

Entropy is a measure of the uncertainty of a random variable, it is defined as:

$$H(X) = -\sum_{i=1}^{n} p\left(x_i\right) \log p\left(x_i\right)$$

where $p(x_i)$ is the probability of the ith outcome. The information gain is the difference between the entropy before the split and the weighted average of the entropy after the split.

$$IG(Y, X_j = x) = H(Y) - H(Y|X_j = x)$$

where S is the set of examples, A is the attribute to be tested, S_v is the subset of S for which attribute A has value v.

$$IG(Y, X_j) = H(Y) - \sum_x p(X_j = x) H(Y|X_j = x)$$

Here Y is the class label, X_j is the jth feature.

9.4 Pros and cons

- Pros: Tree are interpretable, you can easily read off why model gave a particualr prediction
- Pros: Trees can handle both numerical and categorical data.
- Pros: Trees can handle missing data, using surrogate variables.
- Cons: Trees are unstable, a small change in the data can lead to a large change in the structure of the tree. This can be solved by using ensemble methods.

9.5 Size of tree

- Small tree are has the risk of under-fitting, i.e. high bias, low variance
- Large tree has the risk of over-fitting, i.e. low bias, high variance

Interpretation of bias and variance in decision tree: bias comes from the simplifying assumptions made to the model, variance comes from the sensitivity to small changes in the training set.

9.5.1 Pruning

Pruning is a method to reduce the size of the tree. It is a method to avoid overfitting.

1. Start with a large tree T_0 and with complexity hyperparameter α.

2. Find the subtree $T \subset T_0$ such that it minimises the cost function:

$$sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

where $|T|$ is the number of terminal nodes of Tm and R_m is the rectangle (region) corresponding to the mth terminal node.

3. Use cross-validation to choose α. We can do it bottom up and top-down.
4. $\alpha = 0$ means no pruning, $\alpha = \infty$ means the 1 leaf.

9.6 Bagging

Bagging is the process of creating multiple models from different subsets of the training data, of the same size, by bootstraping. The final prediction is averaged across (majority vote or average across) the predictions of all of the sub-models.

9.7 Random Forest

Random forest is an ensemble method that combines multiple decision trees. It is a bagging method with a random selection of features. The final prediction is averaged across the predictions of all of the sub-models.

9.8 Boosting

Boosting sequentially builds a model by training a decision tree on the residuals of the previous model. The final prediction is the sum of the predictions of all of the sub-models. The parameters include the learning rate, which is the shrinkage factor of the model for the residuals.

10 K-means

K-mean is a clustering method, which is an unsupervised learning method. It is used to group data points into clusters, where the data points in the same cluster are similar to each other, and the data points in different clusters are dissimilar to each other.

K-mean clustering are widely used in pattern recognition, computer vision, etc.

K-mean is similar to k-nearest neighbour.

K-mean is a NP-hard problem, which means that it is computationally expensive to find the optimal solution. Therefore, we use a heuristic approach to find a sub-optimal solution.

10.1 Algorithm

10.1.1 Lloyd's algorithm (naive k-means)

Steps of Lloyd's algorithm:

1. Initialisation: Randomly assign each data point to one of the k clusters
2. Assign each data point to the closest centroid
3. Compute the centroid of each cluster
4. Repeat steps 2 and 3 until the centroids do not change

This algorithm should reduce the within-cluster sum of squares (WCSS) at each iteration.

10.1.2 K-means alternatives

- K-median: use the median instead of the mean to compute the centroid
- K-medoids: use the most central point in the cluster as the centroid
- Manhattan distance: use Manhattan distance instead of Euclidean distance to compute the distance between a data point and a centroid (i.e. $d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$)

10.1.3 Initialisation

There are serveral ways to initialise the centroids:

- Random partition: Randomly assign each data point to one of the k clusters
- Forgy: Randomly select k data points as the initial centroids
- K-means++: select the first centroid randomly, then select the next centroid from the remaining data points with a probability proportional to the distance from the previous centroid

10.2 Fuzzy k-means

1. Multiple cluster assignment: x_i has cluster assignment w_{ik}, with $w_{ik}^{-1} = \sum_{j=1}^{K} \left(\frac{\|x_i - c_k\|}{\|x_i - c_j\|} \right)^{\frac{2}{m-1}}$
2. Centroid update - New cluster centroids are $c_k = \frac{\sum_{i=1}^{n} w_{ik}^m x_i}{\sum_{i=1}^{n} w_{ik}^m}$ - This minimises the weighted mean squared error $E = \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik}^m \|x_i - c_k\|^2$ - m is a fuzziness hyper parameter

11 Clustering

11.1 Clustering evaluation

1. Internal evaluation: Optimising a function over clusters doesn't necessarily mean that the clusters are good
2. A internal criterion involves measuring the similarity between data observations within the same cluster and the dissimilarity between data observations from different clusters
3. External evaluation: Requires true label which is not always available

11.1.1 Silhouette score

- $$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j),$$
- b_i is the minimum (amongst all other clusters) average distance to the points in cluster C_j to which i does not belong:
$$b_i = \min_{j \neq i} \frac{1}{|C_j|} \sum_{k \in C_j} d(i, k)$$
where $d(i, j)$ is the distance between ith and jth data point. C_i is the cluster that the ith data point belongs to. $|C_i|$ is the number of data points in cluster C_i.
- The silhouette score is then:
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

The overall silhouette score is the average of the silhouette score of all data points.

$$s = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

- The silhouette score is between -1 and 1. The higher the score, the better the clustering.

11.1.2 Other evaluation metrics

- Elbow method: plot the within-cluster sum of squares (WCSS) against the number of clusters. The optimal number of clusters is the number of clusters after which the WCSS does not decrease significantly.
- Information criteria: AIC, BIC, etc.
- Gap statistic: compare the within-cluster sum of squares (WCSS) of the data to the WCSS of the data generated from a uniform distribution.
-

11.2 Issues with clustering

- clustering is very sensitive to choices that you make, i.e. hyperparameters, choice of distance matrix.
- calculate pairwise distance is computationally expensive
- interpretation of clusters is subjective
- hard clustering is not robust to outliers, i.e. since all data points are assigned to a cluster, outliers can affect the cluster centroid.
- very unstable to small changes in the data
- in high dimensional space, the distance between data points becomes less meaningful, i.e. the curse of dimensionality. There is not much difference in Euclidean space between two data points.

11.3 Types of clustering

- Hard vs soft clustering
- Hierarchical vs non-Hierarchical clustering
- Agglomerative vs partitioning/divisive
- Centroid vs distribution-based vs density vs graph-based vs spectral. Examples include k-means, Gaussian mixture models, DBSCAN, etc.

11.4 Various types 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

12 GMM

12.1 Self-organising maps(non-examinable)

Self organising map is a dimensionality reduction technique. It distort the data space to a 2D space, while preseving the grid structure.

12.2 Generative Modelling

We assume that the data is generated from a mixture of Gaussian distribution. Each point is generated from one or more of the Gaussian distribution.

12.2.1 Latent variable

We introduce a latent variable z to indicate which Gaussian distribution the data point is generated from.

$$p(x, z) = p(x|z)p(z)$$

In more general case with θ

$$p(x, z|\theta) = p(x|z, \theta)p(z|\theta)$$

Given that we do not know z, we can marginalise it out:

$$p(x|\theta) = \sum_z p(x, z|\theta) = \sum_z p(x|z, \theta)p(z|\theta)$$

In the case that z is discrete, we can use the sum. Assuming z is discrete, and z can be 1,2,3 .. K, we can write:

$$p(x|\theta) = \sum_{k=1}^{K} p(x|z = k, \theta)p(z = k|\theta)$$

12.3 Gaussian Mixture Model

Gaussian mixture model, is defined as:

$$p(x|\theta) = \sum_{k=1}^{K} \phi_k f\left(x|\mu_k, \Sigma_k\right)$$

where ϕ_k is the mixing coefficient, μ_k is the mean, Σ_k is the covariance matrix, and f is the Gaussian distribution. The full parameter space is then:

$$\theta = \{\phi_1, \ldots, \phi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K\}$$

Note that $\sum_{k=1}^{K} \phi_k = 1$. This is related to the latent variable model by:

$$\phi_k = p(z = k|\theta)$$

12.3.1 Expectation Maximisation

We want to find the maximum likelihood estimate of θ. However, there is latent variable z that we do not know. Therefore, we use the expectation maximisation algorithm to find the maximum likelihood estimate of θ.

1. Expectation: compute the posterior probability of z given x and θ:

$$q_{ik} = \frac{\phi_k f\left(x_i|\mu_k, \Sigma_k\right)}{\sum_{j=1}^{K} \phi_j f\left(x_i|\mu_j, \Sigma_j\right)}$$

2. Maximisation: maximise the expected log likelihood with respect to θ:

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{K} q_{ik}^{(t)} \log\left(\frac{p\left(x_i, z_i = k \mid \theta\right)}{q_{ik}^{(t)}}\right)$$

Here we used the Jensen's inequality:

$$\sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} q_{ik} \frac{p\left(x_i, z_i = k \mid \theta\right)}{q_{ik}}\right) \geq \sum_{i=1}^{n} \sum_{k=1}^{K} q_{ik} \log\left(\frac{p\left(x_i, z_i = k \mid \theta\right)}{q_{ik}}\right)$$

13 Spectral Clustering

Spectral clustering is a graph-based clustering method

- Diagonal matrix D is the degree matrix, where

$$D_{ii} = deg(i) = \sum_{j=1}^{n} W_{ij}$$

  which is the value of number of edges connected to node i.

- Adjacency matrix A is the matrix of connections between nodes, where

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge between node } i \text{ and node } j \\ 0, & \text{otherwise} \end{cases}$$

- The Laplacian matrix L is defined as:

$$L = D - A$$

- L is positive semi-definite, and has n non-negative real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. ...

14 DBSCAN

DBSCAN stands for density-based spatial clustering of applications with noise.

- 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
- All other points are outliers

14.1 Pros and cons

- Pros: DBSCAN can find clusters of any shape, and it can also identify outliers.
- Pros: DBSCAN is a non-parametric method, which means that it does not require data pre-processing.
- Pros: DBSCAN can also identify outliers.s
- Cons: DBSCAN is very sensitive to the choice of ϵ and $minPts$.
- Cons: DBSCAN is very sensitive to the choice of distance metric.
-

It is a density-based clustering method, which means that it can find clusters of any shape, and it can also identify outliers. It is also a non-parametric method, which means that it does not require data pre-processing. It has two parameters: ϵ and $minPts$. ϵ is the radius of the circle to be created around each data point to check the density . $minPts$ is the minimum number of points in a neighbourhood to be considered as a core point.