Binary classification

Person	Height(m)	Weight(kg)	Male?	Prediction
P1	1.79	75	Yes	Yes
P2	1.64	54	No	No
P3	1.70	63	Yes	No
P4	1.88	78	Yes	Yes
P5	175	70	Yes	No
P6	1.65	52	No	Yes

evaluate
each sample

Performance evaluation

Confusion matrix
for classification

		Predicted class	
Actual class		Class=Yes	Class=No
	Class=Yes	a(TP)	b(FN)
	Class=No	c(FP)	d(TN)

Evaluation metrics:
Accuracy
Precision
Recall
F1-score
Balanced accuracy

Confusion matrix
for clustering

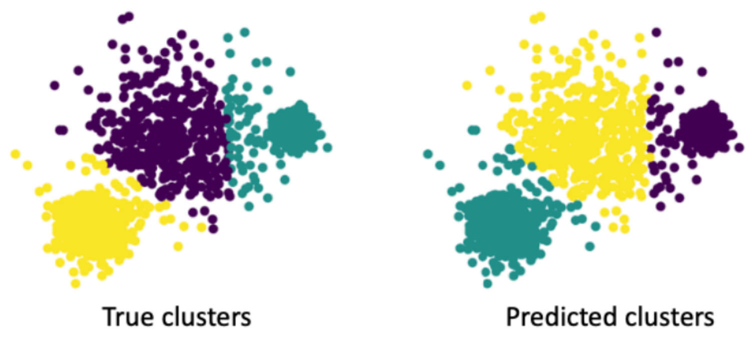
		Predicted clusters	
Actual clusters		The same	Not the same
	The same	a(TP)	b(FN)
	Not the same	c(FP)	d(TN)

Evaluation metrics:
Rand index, R

evaluate
each pair
of samples

Clustering

messy classification, but good clustering



decompose
multi-class
classification into
a collection of
binary problems

aggregate
performance of
each binary
classification

Aggregation

Class	Accuracy	Cells
1	0.9	150
2	0.95	50
3	0.85	100
4	0.8	40
5	0.7	20
6	0.2	10

$$Macro - average = \frac{0.9 + 0.95 + \cdots + 0.7 + 0.2}{6} = 0.73$$

$$Micro - average = \frac{0.9 * 150 + \cdots + 0.2 * 10}{150 + \cdots + 10} = 0.85$$

evaluate a model based
on its performance on
each validation subset
in training data

compare models
with different
hyper-parameters

Comparison

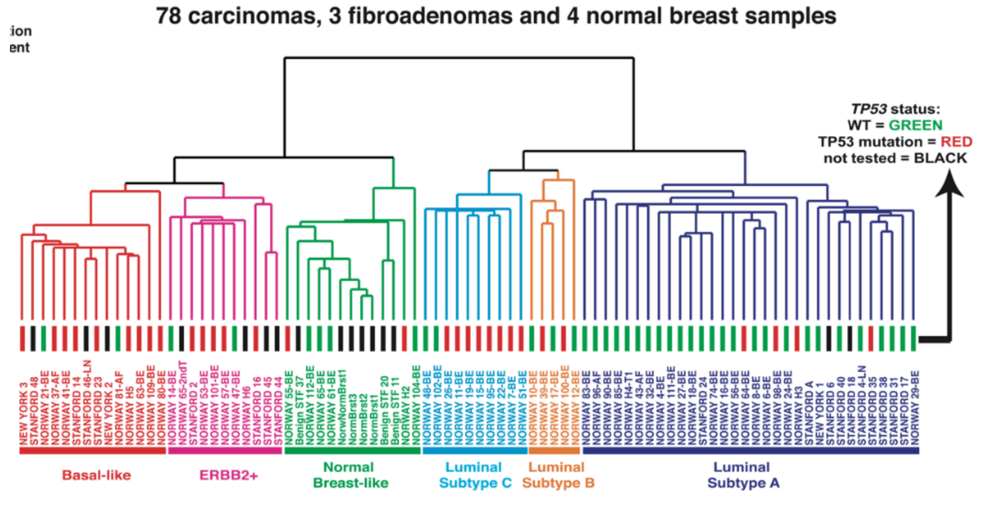
Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M

KNN with K=3
P1: M
P2: M
P3: M
P4: M
Accuracy=0.75

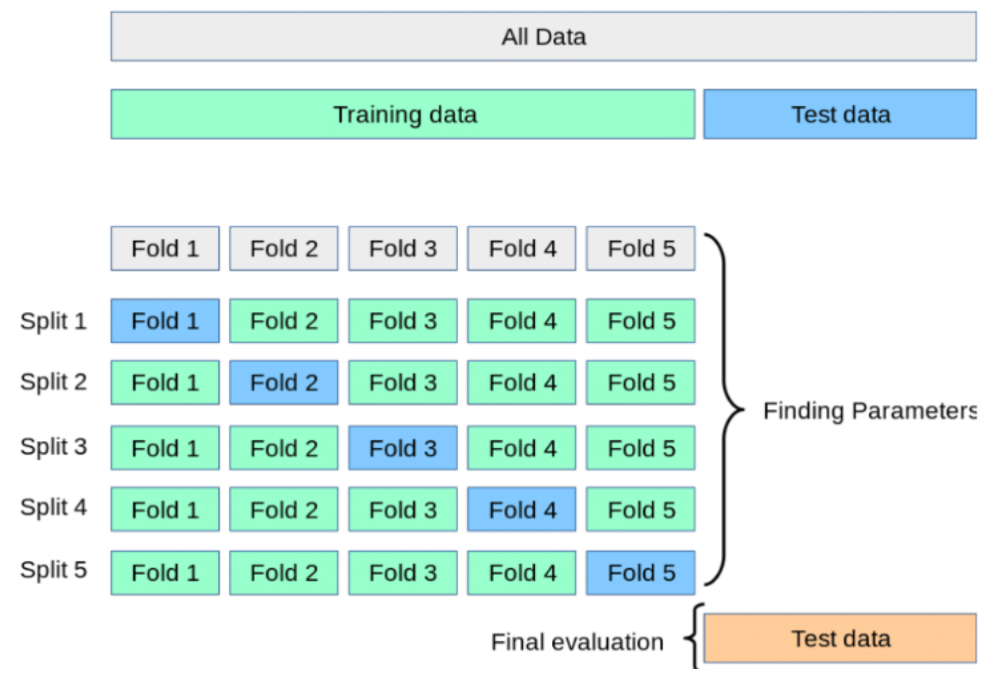
KNN with K=1
P1: P4—M
P2: P3—M
P3: P2—F
P4: P1—M
Accuracy=0.5

Multi-class classification

78 carcinomas, 3 fibroadenomas and 4 normal breast samples



n-fold cross-validation



BMEG3105 Data analytics for personalized genomics and precision medicine
Lecturer: Yu LI (李煜) from CSE
Wednesday, 5 October 2022
Lecture 9: Clustering and classification performance evaluation

In this scribing, the main points are usually highlighted with colors. The paragraphs and sentences are to provide further clarification and elaboration about the concepts. You may skip them if you are already familiar with the details of those concepts.

Review of last lecture

- logistic regression
- loss function
- gradient descent algorithm

Today's content - performance evaluation

Central question: How good is this classification/clustering model?

>Purposes of performance evaluation

During classification and clustering, there are a variety of choices and methods for different steps. Examples of these choices include, but not limited to, normalization methods (min-max or z-score normalization), definition of distance between samples (correlation coefficient or Euclidean distance), the value of K in the K-nearest neighbors algorithm. All these decisions will result in a different model that may perform differently even with the same data. Therefore, performance evaluation is essential in order to:

- Comparing different models/methods quantitatively
- Identify the strengths and weaknesses of a model/method
- Select the most suitable model/method for the question of interested

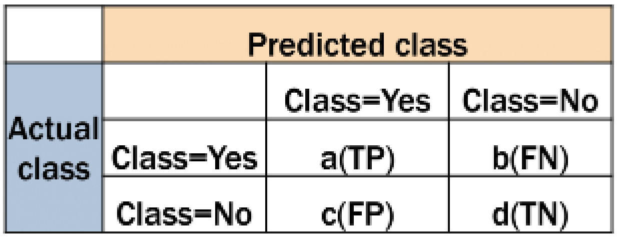
>Outline

1. Binary classification - confusion matrix and metrics
2. Multi-class classification - an extension from binary classification evaluation
3. Cross-fold validation - model evaluation by training data only
4. Clustering - each pair of samples rather than each individual sample

1. Binary classification - confusion matrix and metrics

The ultimate goal of classification is to assign a correct label to a sample according to its class. Assuming that ground truths of a binary-outcomes classification are known, a confusion matrix can be drawn based on the actual and predicted classes of samples. Several useful metrics can be obtained from the confusion matrix for evaluation.

>Confusion matrix



		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	a(TP)	b(FN)
	Class=No	c(FP)	d(TN)

- TP: true positive, samples that are actually “Yes” and are also predicted as “Yes”
- FN: false negative, samples that are actually “Yes” and are predicted as “No”
- FP: false positive, samples that are actually “No” and are predicted as “Yes”
- TN: true negative, samples that are actually “No” and are also predicted as “No”

Remark: “Yes” and “No” refer to the two distinct classes in the binary outcomes

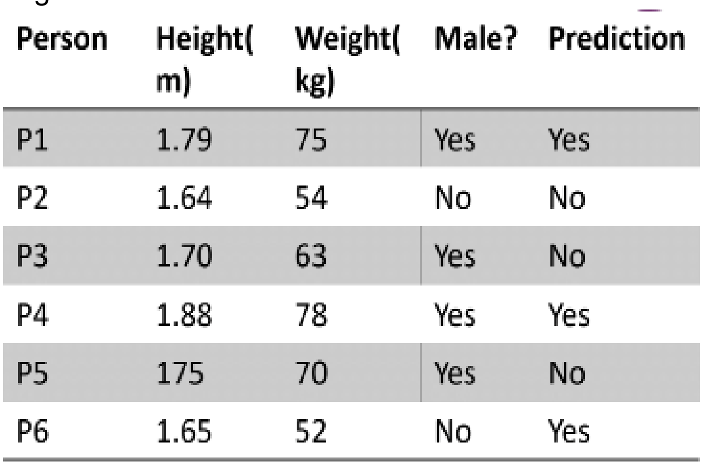
>Metrics derived from confusion matrix

Names of metrics	Mathematical definitions	Physical meanings
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Among all predictions, how many of them are correct?
Precision (= positive predictive value)	$\frac{TP}{TP+FP}$	Among all the predicted positive (“Yes”) samples, how many of them are actually positive (“Yes”)?
Recall (= sensitivity or true positive rate)	$\frac{TP}{TP+FN}$	Among all the actual positive (“Yes”) samples, how many of them are predicted positive (“Yes”)?
F1-score	$\frac{2*precision*recall}{precision+recall}$	Harmonic mean of precision and recall (i.e. weighted average of precision and recall with both treated as of same importance)
Specificity (= true negative rate)	$\frac{TN}{TN+FP}$	Among all the actual negative (“No”) samples, how many of them are predicted negative (“No”)?
Balanced accuracy	$0.5*\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$	Arithmetic mean of true positive rate and true negative rate, for imbalanced data

Remark: Specificity is not included in class, I just add here for the sake of completeness

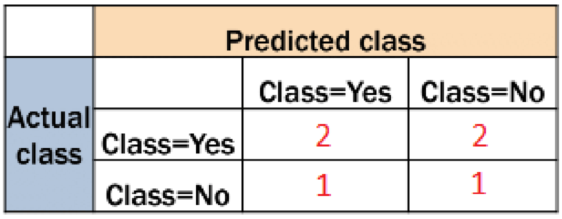
>An example

Original data matrix



Person	Height(m)	Weight(kg)	Male?	Prediction
P1	1.79	75	Yes	Yes
P2	1.64	54	No	No
P3	1.70	63	Yes	No
P4	1.88	78	Yes	Yes
P5	175	70	Yes	No
P6	1.65	52	No	Yes

Confusion matrix and performance evaluation



		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	2	2
	Class=No	1	1

Accuracy = (2+1)/(2+1+1+2) = 0.5
Precision = (2)/(2+1) = 0.667
Recall = (2)/(2+2) = 0.5
F1-score = (2*0.667*0.5)/(0.667+0.5) = 0.571
Specificity = (1)/(1+1) = 0.5
Balanced accuracy = 0.5*[2/(2+2)+1/(1+1)] = 0.5

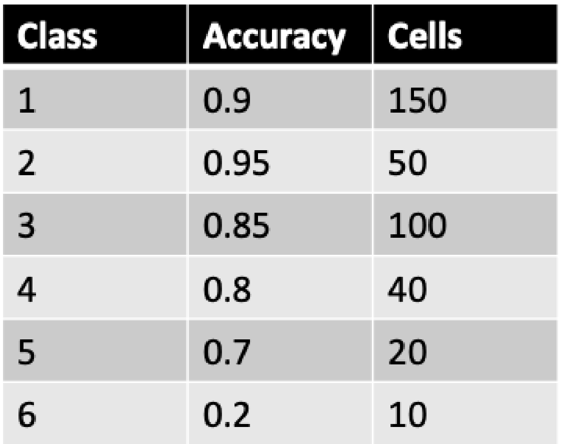
>Pitfalls when analyzing these metrics

- These metrics (even balanced accuracy) can be misleading when data is imbalanced, it is always better to look at confusion matrix directly in such case
- Model with higher accuracy does not necessarily be better, depending on the context. For example, for rare cancer pre-screening, a model with ~100% recall but only 50% accuracy is better than a model with ~100% accuracy but only 50% recall because the potential consequence for any missing positive case can be serious

2. Multi-class classification - an extension from binary classification evaluation

Often, there will be more than 2 classes for a classification problem. For KNN, no change is required for the algorithm to predict multi-class. For logistic regression, a logistic regression model will be built for each class. The class with the highest value is assigned to the data during prediction. The performance evaluation metrics (e.g. accuracy, precision, recall, F1 score…) can then extend its application to multi-class classification by considering each class as a binary classification problem. In other words, a multi-class classification is transformed into a collection of binary classifications. The accuracies for all of the classes are aggregated to form a single value representing the multi-class classification performance.

>An example (accuracy is computed for each class)



Class	Accuracy	Cells
1	0.9	150
2	0.95	50
3	0.85	100
4	0.8	40
5	0.7	20
6	0.2	10

Classification results

$$Macro-average = \frac{0.9 + 0.95 + \cdots + 0.7 + 0.2}{6} = 0.73$$

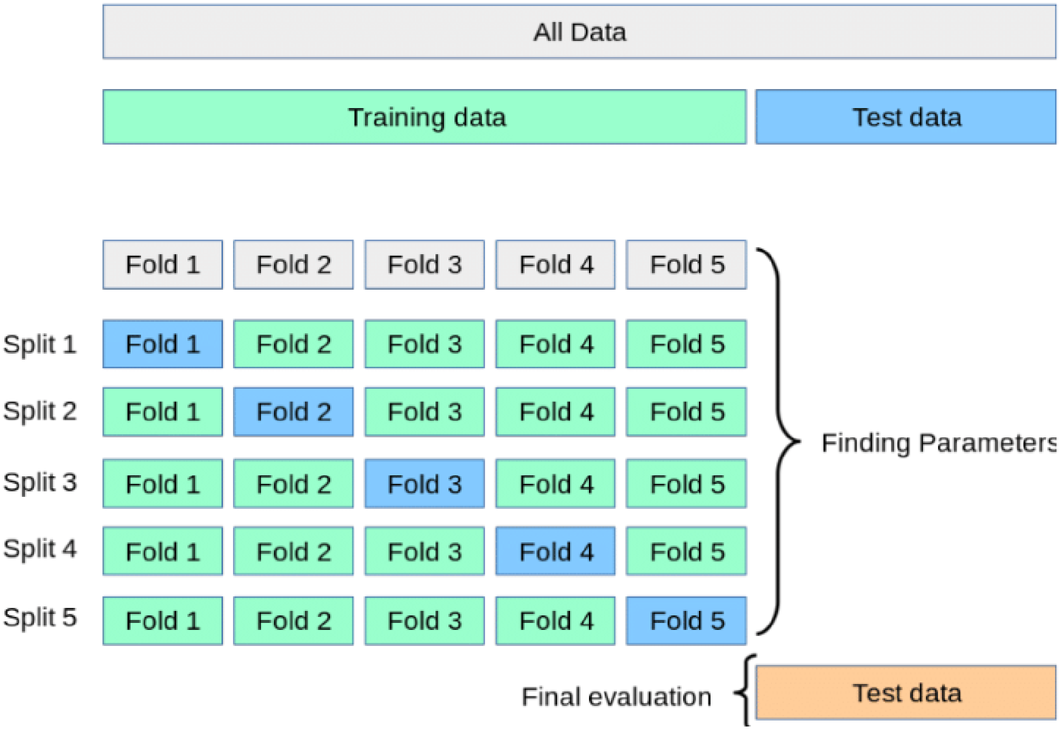
$$Micro-average = \frac{0.9 * 150 + \cdots + 0.2 * 10}{150 + \cdots + 10} = 0.85$$

Aggregated accuracy

In this example, the cells are classified into 6 different classes. By decomposing the multi-class classification into a collection of binary classifications, the accuracy for each class can be computed. These accuracies can be further aggregated into one value to represent performance of multi-class classification. The aggregated accuracy can be a macro-average or micro-average. In macro-average, each class is treated as equally important, which allows the performance of small classes to show up. In micro-average, the accuracy of each class is weighted by the class size, which highlights the performance of classes with more samples.

3. Cross-fold validation/n-fold cross-validation - model evaluation by training data only

The main purpose of n-fold cross-validation is to evaluate the performance of a model/method using the training data only because the ground truths of testing data may not be known. n-fold cross-validation involves partitioning the training data into n complementary subsets and evaluates the classification performance on each subset. It is a systemic way to determine hyper-parameters so that the best method/model for classification can be selected. Leave-one-out cross-validation is a special form of n-fold cross-validation with n=N, where N is the size of training data.



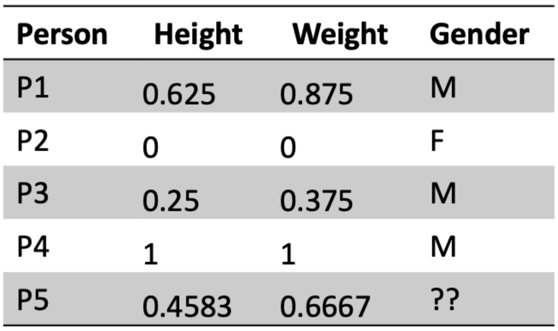
Schematic diagram of 5-fold cross-validation

>Procedures

- Partition training data into n disjoint subsets randomly
- Repeat the following processes for each subset
    - Choose one subset to be the validation data
    - Predict the class of validation data using other non-chosen subsets
    - Compute accuracy on validation data by comparing prediction and truth
- Final accuracy = mean of the accuracies obtained from all iterations

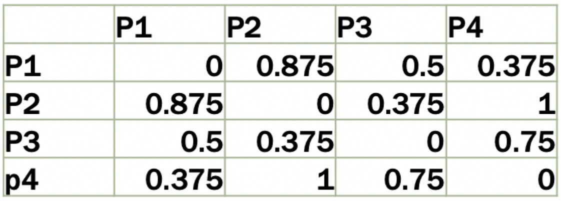
>An example (leave-one-out cross-validation to determine whether K=1 or K=3 is better)

Original data (only P1-P4 are training data)



Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

Distance matrix (supremum distance)



	P1	P2	P3	P4
P1	0	0.875	0.5	0.375
P2	0.875	0	0.375	1
P3	0.5	0.375	0	0.75
p4	0.375	1	0.75	0

When K=1

Validation data	Neighbor	Prediction	Actual
P1	P4	M	M
P2	P3	M	F
P3	P2	F	M
P4	P1	M	M

Accuracy = 2/4 = 0.5

When K=3

Validation data	Neighbors	Prediction	Actual
P1	P2,P3,P4	M	M
P2	P1,P3,P4	M	F
P3	P1,P2,P4	M	M
P4	P1,P2,P3	M	M

Accuracy = 3/4 = 0.75 > 0.5

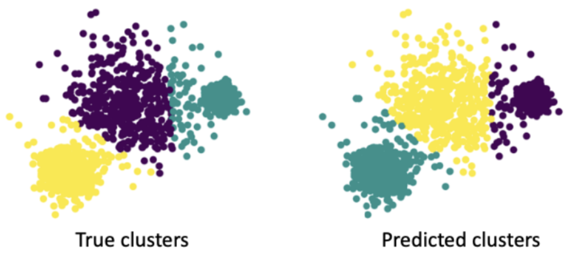
So K=3 is a better hyper-parameter for this problem

>Potential usages

- Estimate the performance of a model on the testing data
- Choose the K for KNN
- Select a better classification method from KNN and logistic regression

Not suitable for training the weights for logistic regression (because the weights are not hyper-parameters defined by users, they should be obtained through gradient descent algorithm)

<u>4. Clustering</u> - each pair of samples rather than each individual sample



messy classification, but good clustering

True clusters

Predicted clusters

Clustering is different from classification in the way that no label is available to verify the correctness of each sample. The performance of clustering should be judged by considering the relations among different samples. As long as two similar samples are grouped in the same cluster, the clustering algorithm has performed correctly, and vice versa. To evaluate the performance of a clustering model, all pairs of samples have to be examined. There will be a total of nC2 pairs of samples if the sample size is n. For each pair of samples, their actual clusters are compared to their predicted clusters. Similar to binary-outcomes classification, a confusion matrix can be constructed to summarize the clustering performance. Rand index (homologous to accuracy in classification) can be computed from the confusion matrix.

<u>>Confusion matrix</u>

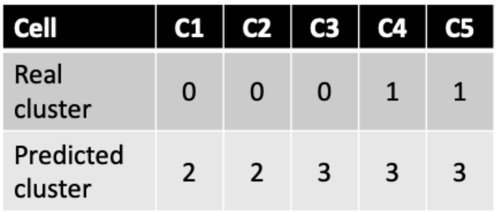
	Predicted clusters		
Actual clusters		**The same**	**Not the same**
	The same	**a(TP)**	**b(FN)**
	Not the same	**c(FP)**	**d(TN)**

- a, TP: true positive, the pairs of samples that originate from the same cluster and are also predicted to one cluster
- b, FN: false negative, the pairs of samples that originate from the same cluster and are predicted to different clusters
- c, FP: false positive, the pairs of samples that originate from different clusters and are predicted to one cluster
- d, TN: true negative, the pairs of samples that originate from different clusters and are also predicted to different clusters

<u>>Rand index, R</u> (equivalent to accuracy in binary classification)

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{Number\ of\ all\ the\ pair\ combinations}$$

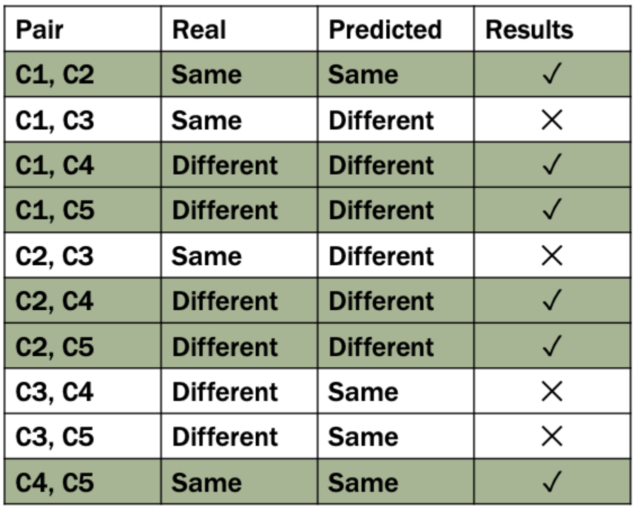
>An example



Cell	C1	C2	C3	C4	C5
Real cluster	0	0	0	1	1
Predicted cluster	2	2	3	3	3

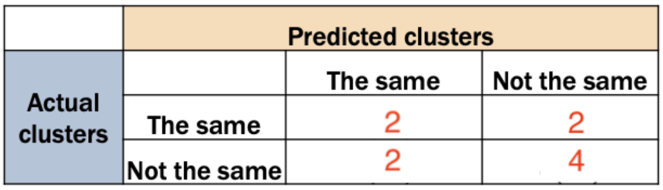
Clustering result

In the above data, five cells are grouped into two clusters using a certain clustering algorithm. To assess the performance of this clustering, the relations for each pair of samples are explored by comparing the real clusters and the predicted clusters of each pair of samples. A confusion matrix and rand index are then obtained.



Pair	Real	Predicted	Results
C1, C2	Same	Same	✓
C1, C3	Same	Different	×
C1, C4	Different	Different	✓
C1, C5	Different	Different	✓
C2, C3	Same	Different	×
C2, C4	Different	Different	✓
C2, C5	Different	Different	✓
C3, C4	Different	Same	×
C3, C5	Different	Same	×
C4, C5	Same	Same	✓

Evaluation on each pair of samples



		Predicted clusters	
		The same	Not the same
Actual clusters	The same	2	2
	Not the same	2	4

Confusion matrix

Rand index, R = (2+4) / (2+2+2+4) = 0.6

Tool for performance evaluation

Python package: *Scikit-learn*
https://scikit-learn.org/stable/modules/clustering.html#clustering- performance-evaluation
https://scikit- learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
https://scikit-learn.org/stable/modules/cross_validation.html

Potential project - 2,3

Project title: *Data preprocessing for the gene expression matrix*

- Data collecting and merging (if needed)
- Exploration
- Visualization
- Data cleaning
- Dimension reduction (next lecture)
- Get distance matrix
- Perform classification/clustering
- Performance evaluation

Resource and uncovered topics

- Introduction to data mining: Chapter 4.5 & 4.6 & 5.7 & 5.8 & 8.5
- Bootstrap
- Overfitting and generalization
- Other clustering and classification methods
- Comparison between different methods
- Clustering
- Classification