Data Science

Machine Learning Techniques

Lesson 7: Performance Metrics, Imbalanced Data and Clustering Analysis

Hang Zhang, Ph.D.

November 19th, 2018

Lecture Outline

- Capstone Project
- Performance Metrics of Models
- Imbalanced Data
- Clustering Analysis

Where Are You?

Team Name	Public Ranking	Best Score	# of Submissions
Datas R Us	626	32.265	2
DS420_PandaPlayers	406 (412) 	1.958 (3.386)	3
DS420_astroclass_capstone (Merged with datarangers)	303	1.425	1
DS420_TYF			
DS420-GRJT	295 (470) 	1.420 (29.219)	4(1)
DS420_Galileo's_Gala	612	24.950	3
DS420_Galaxy	502	2.492	2

As of 23:50, November 18th, 2018

Performance Metrics in Classification

		Actual	
		Positive	Negative
Predicted	Pos	n_{11}	n_{12}
	Neg	n_{21}	n_{22}

- Type I error (False positive):

$$\Pr(\hat{y} = Pos \mid y = Neg) = n_{12} / (n_{12} + n_{22}) = n_{12} / n_{\bullet 2}$$

- Type II error(False negative):

$$\Pr(\hat{y} = Neg \mid y = Pos) = n_{21} / (n_{11} + n_{21}) = n_{21} / n_{\bullet 1}$$

- Accuracy: $(n_{11} + n_{22}) / (n_{11} + n_{22} + n_{12} + n_{21})$
- Sensitivity (True Positive Rate, Recall): Among the (n_{11}+n_{21}) true positive cases, the percentage that is predicted as positive:

$$\Pr(\hat{y} = Pos \mid y = Pos) = n_{11} / (n_{11} + n_{21}) = n_{11} / n_{\bullet 1} = 1 - Type\ II\ \text{Error}$$

- Precision: Among the (n_{11}+n_{12}) predicted positive cases, the percentage that is actually positive:

$$\Pr(y = Pos \mid \hat{y} = Pos) = n_{11} / (n_{11} + n_{12}) = n_{11} / n_{1\bullet}$$

F-Score

- Sometimes we want a single number that informs us of the quality of the solution. A popular way to combine precision (P) and recall (R) into a single number is by taking their harmonic mean. This is known as the balanced f-measure:

$$F = \frac{2 \times P \times R}{P + R}$$

Note…

One thing to keep in mind is that precision and recall (and hence f-measure) ***depend crucially on which class is considered*** the thing you wish to find. In particular, if you take a binary data set and flip what it means to be a positive or negative example, you will end up with completely different precision and recall values.

It is ***not the case that precision on the flipped task is equal to recall on the original task*** (nor vice versa). Consequently, f-measure is also not the same. For some tasks where you are less sure about what you want, report two sets of precision/recall/f-measure numbers, which vary based on which class is considered the thing to spot.

Example of Flipping Positive and Negative, and the Impacts on Performance Metrics

		Actual	
		Positive	Negative
Predicted	Pos	n_{11}	n_{12}
	Neg	n_{21}	n_{22}

		Actual	
		Negative	Positive
Predicted	Neg	n_{11}	n_{12}
	Pos	n_{21}	n_{22}

- Accuracy: $(n_{11}+n_{22})/(n_{11}+n_{22}+n_{12}+n_{21})$
- Sensitivity (True Positive Rate, Recall): Among the $(n_{11}+n_{21})$ true positive cases, the percentage that is predicted as positive:

$$\Pr(\hat{y}=Pos \mid y=Pos)=n_{11}/(n_{11}+n_{21})$$

- Precision: Among the $(n_{11}+n_{12})$ predicted positive cases, the percentage that is actually positive:

$$\Pr(y=Pos \mid \hat{y}=Pos)=n_{11}/(n_{11}+n_{12})$$

- Recall:

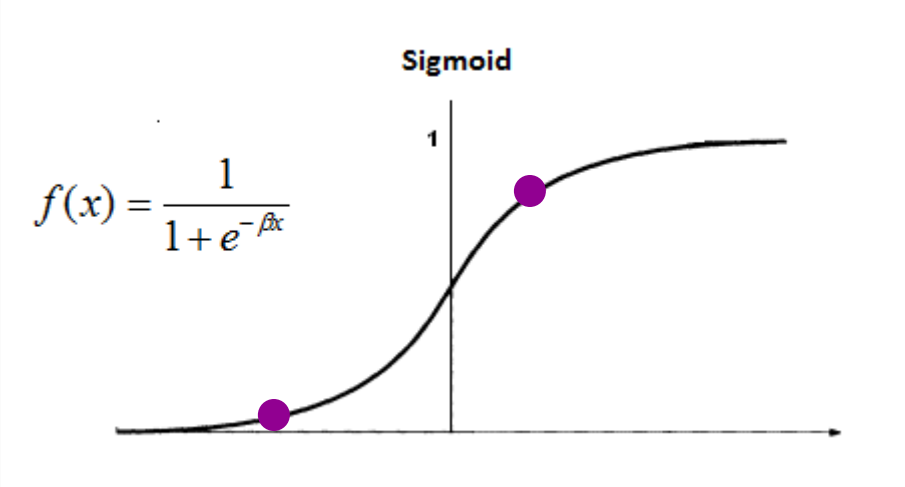
$$\Pr(\hat{y}=Pos \mid y=Pos)=n_{22}/(n_{12}+n_{22})$$

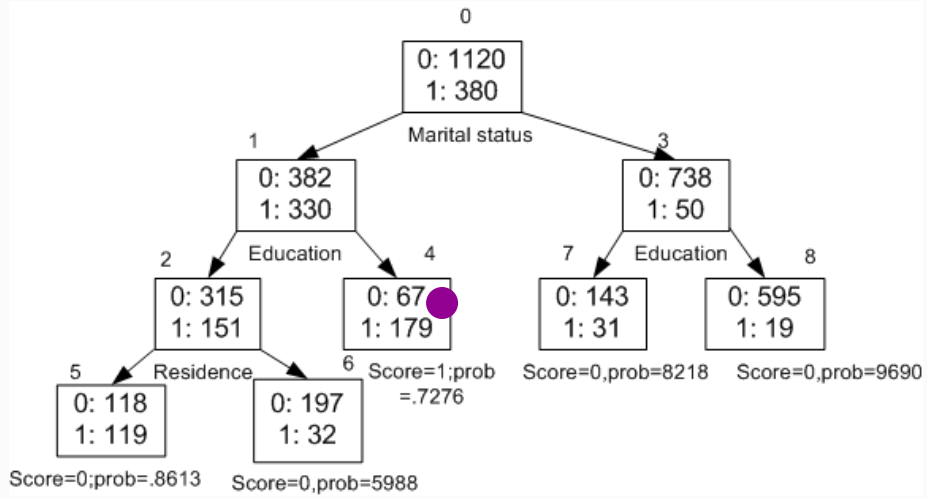
- Precision:

$$\Pr(y=Pos \mid \hat{y}=Pos)=n_{22}/(n_{21}+n_{22})$$

Accuracy, Recall, and Precision All Depend on Threshold

- Usually, classification model always outputs probability that an observation belongs to a class





- Whether we assign label 0 or 1 depends on whether the probability is greater than a threshold
  - In many algorithms, 0.5 is the default threshold

Let's See How the Threshold Impacts the Performance Metrics

Education	Age	Employer Sector	Prob(Salary>65k)	Actual	Threshold
High-School	50	Government	0.38	0	0.5
Bachelor	45	Private	0.69	1	
Associate	30	Non-profit	0.61	0	
Bachelor	36	Private	0.73	1	
Master	42	Private	0.82	1	
PhD	48	Government	0.7	1	
Master	25	Private	0.56	1	
Associate	20	Non-profit	0.48	0	
Bachelor	37	Private	0.92	0	
PhD	51	Government	0.79	1	

		Actual	
		Positive	Negative
Predict	Positive	6	2
	Negative	0	2

Recall=	100
Precision=	75

Education	Age	Employer Sector	Prob(Salary>65k)	Actual	Threshold
High-School	50	Government	0.38	0	0.8
Bachelor	45	Private	0.69	1	
Associate	30	Non-profit	0.61	0	
Bachelor	36	Private	0.73	1	
Master	42	Private	0.82	1	
PhD	48	Government	0.7	1	
Master	25	Private	0.56	1	
Associate	20	Non-profit	0.48	0	
Bachelor	37	Private	0.92	0	
PhD	51	Government	0.79	1	

		Actual	
		Positive	Negative
Predict	Positive	1	1
	Negative	5	3

Recall=	16.66667
Precision=	50

- Different threshold, you may get different performance metrics
- Is there a performance metrics that is independent with the threshold?

In-class Lab:

- Performance metrics.xlsx

ROC Curve

		Actual	
		Positive	Negative
Predicted	Pos	n_{11}	n_{12}
	Neg	n_{21}	n_{22}

- Two performance metrics are used to plot ROC curve:
  - True Positive (1-Type II error): $\Pr(\hat{y} = Pos \mid y = Pos) = n_{11}/(n_{11}+n_{21}) = n_{11}/n_{\bullet 1} = 1 - Type\ II\ \text{Error}$
  - False Positive (Type I error): $\Pr(\hat{y} = Pos \mid y = Neg) = n_{12}/(n_{12}+n_{22}) = n_{12}/n_{\bullet 2}$
  - If threshold = 0, all cases are classified as Positive, Type I error = 1 (all negative cases are classified as Positive), True Positive Rate = 1 (all positive classified as positive)
  - If threshold = 1, all cases are classified as Negative, Type I error = 0 (all negative cases are classified as Negative), True Positive Rate = 0 (All Positive cases are classified as Negative)
- By changing the threshold between 0 and 1, we get a curve connecting [0,0] and [1,1]
- This curve is called ROC curve (Receive Operating Characteristic).

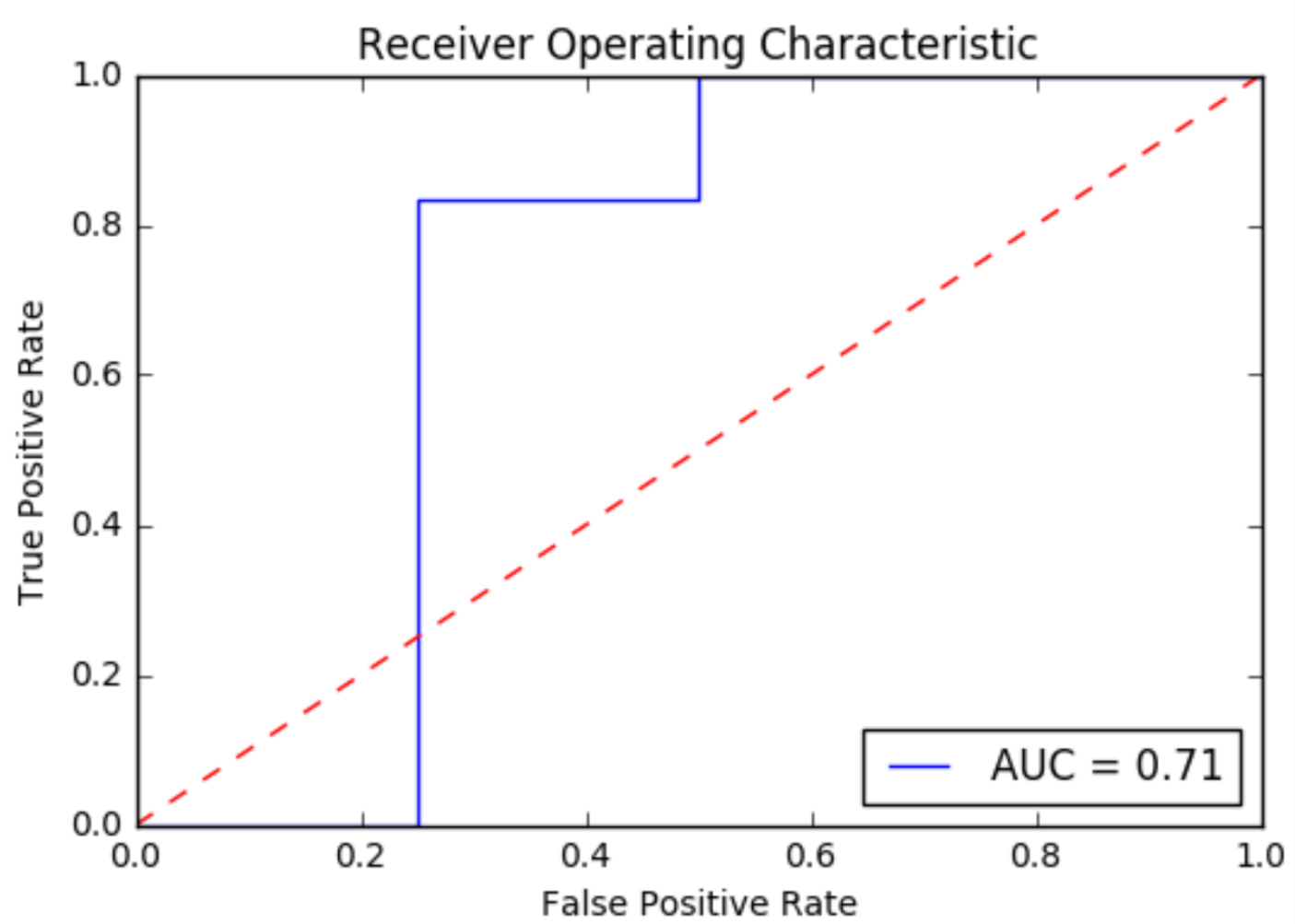
Example

Education	Age	Employer Sector	Prob(Salary>65k)	Actual	Threshold
High-School	50	Government	0.38	0	0.5
Bachelor	45	Private	0.69	1	
Associate	30	Non-profit	0.61	0	
Bachelor	36	Private	0.73	1	
Master	42	Private	0.82	1	
PhD	48	Government	0.7	1	
Master	25	Private	0.56	1	
Associate	20	Non-profit	0.48	0	
Bachelor	37	Private	0.92	0	
PhD	51	Government	0.79	1	

		Actual	
		Positive	Negative
Predict	Positive	6	2
	Negative	0	2

Recall=	100
Precision=	75

```
import matplotlib.pyplot as plt
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

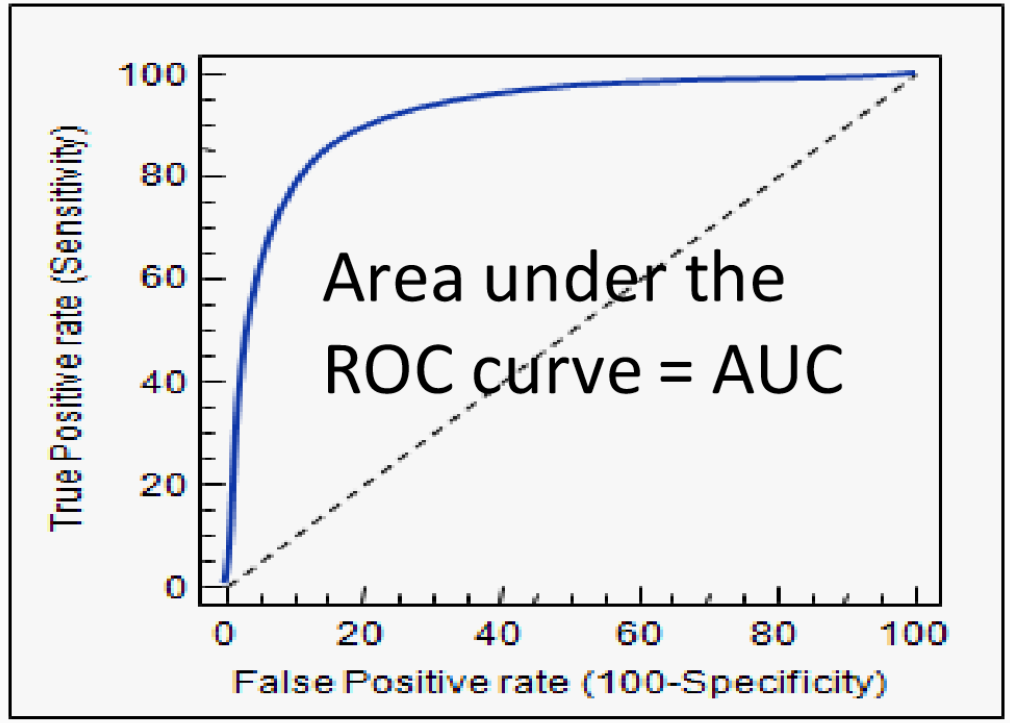


Area Under the ROC Curve

- Area under the ROC curve (AUC) is a measure of the model performance

$$0.5\ (random\ model) < AUC < 1\ (perfect\ model)$$

- Larger the AUC, better is the model



What Is the Statistical Meaning of AUC?

$$\Pr(\hat{y}_i > \hat{y}_j \mid y_i > y_j), \forall i \neq j$$

- Example:

```
import numpy as np
from sklearn import metrics

truth = np.array([0,1,0,1])
prediction = np.array([0.2, 0.5, 0.6, 0.45])

fpr, tpr, thresholds = metrics.roc_curve(truth, prediction, pos_label=1)
auc = metrics.auc(fpr, tpr)
print("Y=" + ",".join(map(str, truth)) + ", Y_hat=%s, AUC=%.2f"%(",".join(map(str, prediction)), auc))

prediction = np.array([0.2, 0.5, 0.6, 0.70])
fpr, tpr, thresholds = metrics.roc_curve(truth, prediction, pos_label=1)
auc = metrics.auc(fpr, tpr)
print("Y=" + ",".join(map(str, truth)) + ", Y_hat=%s, AUC=%.2f"%(",".join(map(str, prediction)), auc))
```

```
Y=0,1,0,1, Y_hat=0.2,0.5,0.6,0.45, AUC=0.50
Y=0,1,0,1, Y_hat=0.2,0.5,0.6,0.7, AUC=0.75
```

yi > yj	$\hat{y}_i > \hat{y}_j$
y2 > y1	Yes
y2 > y3	No
y4 > y1	Yes
y4 > y3	No
AUC	0.5

yi > yj	$\hat{y}_i > \hat{y}_j$
y2 > y1	Yes
y2 > y3	No
y4 > y1	Yes
y4 > y3	Yes
AUC	0.75

Change the last prediction to 0.90 does not change AUC

When to Use AUC as the Performance Metrics?

- When ranking matters
  - Recommendation systems: positions of the recommended items significantly impact the click through rate. So, we need to have the items that are most likely clicked by users returned in higher positions in the results
  - Marketing applications: You have 1M customers, but your marketing campaign only has budget to reach out to 1k customers. You need to rank your customers based on their predicted response probability.
- When training data is extremely skewed:
  - 0.5 is not a reasonable threshold any more.
  - If it is not easy to choose a threshold such that the cost of Type I and Type II errors can be minimized, start with AUC as your performance metrics

However

- Most of the machine learning models are implemented to minimize RMSE-like loss function
  - One model with smaller RMSE does not ensure it will have higher AUC
  - You should tune model parameters to optimize AUC

- Some machine learning algorithms are implemented to optimize AUC
  - Choose these algorithms if you really want to optimize AUC during the model training process
  - SVM^{perf} provides option to optimize AUC during model training process https://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html

Performance Metrics

Receiver operating characteristic (ROC) curve, AUC metric…

- 1.0: perfect prediction *!Alert: Too Good To Be True*
- 0.9: excellent prediction
- 0.8: good prediction
- 0.7: mediocre prediction
- 0.6: poor prediction
- 0.5: random prediction
- <0.5: something wrong!

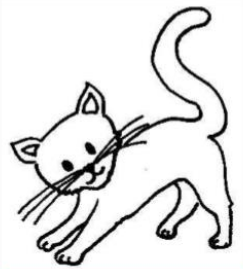
In-class Lab

- ROC and AUC

Dealing with Class Imbalance

Distributions Matter…

Because the Internet is all about cute kittens



Resulting in highly skewed distribution in training set…

The Class Imbalance Problem I

- Data sets are said to be balanced if there are, approximately, as many positive examples of the concept as there are negative ones.
- Many domains that do not have a balanced data set.

Examples:

- Helicopter Gearbox Fault Monitoring
- Discrimination between Earthquakes and Nuclear Explosions
- Document Filtering
- Detection of Oil Spills
- Detection of Fraudulent Telephone Calls

The Class Imbalance Problem II

- The problem with class imbalances is that standard learners are often biased towards the majority class.
- That is because these classifiers attempt to reduce global quantities such as the error rate, not taking the data distribution into consideration.
- As a result examples from the overwhelming class are well-classified whereas examples from the minority class tend to be misclassified.

$$LOSS = \sum_{i=1}^{n} (y_i - f_\theta(\mathbf{x}_i))^2 = \sum_{i=1}^{n_{pos}} (y_i - f_\theta(\mathbf{x}_i))^2 + \sum_{i=1}^{n_{neg}} (y_i - f_\theta(\mathbf{x}_i))^2$$

  - If $n_{neg} >> n_{pos}$, the LOSS function benefits more on making the negative cases accurate, than on making the positive cases accurate

Some Generalities

- Evaluating performance of a model on a class imbalance problem is not done appropriately with standard accuracy/error rate.
  - ROC Analysis is typically used, instead.

- There are three main ways to deal with class imbalances: re-sampling, re-weighing, and one-class learning

SMOTE: A State-of-the-Art Resampling Approach

- SMOTE stands for Synthetic Minority Oversampling Technique.
  - Technique designed by Chawla, Hall, & Kegelmeyer in 2002.

- It combines Informed **oversampling** of the **minority class** with **random undersampling** of the **majority class**.

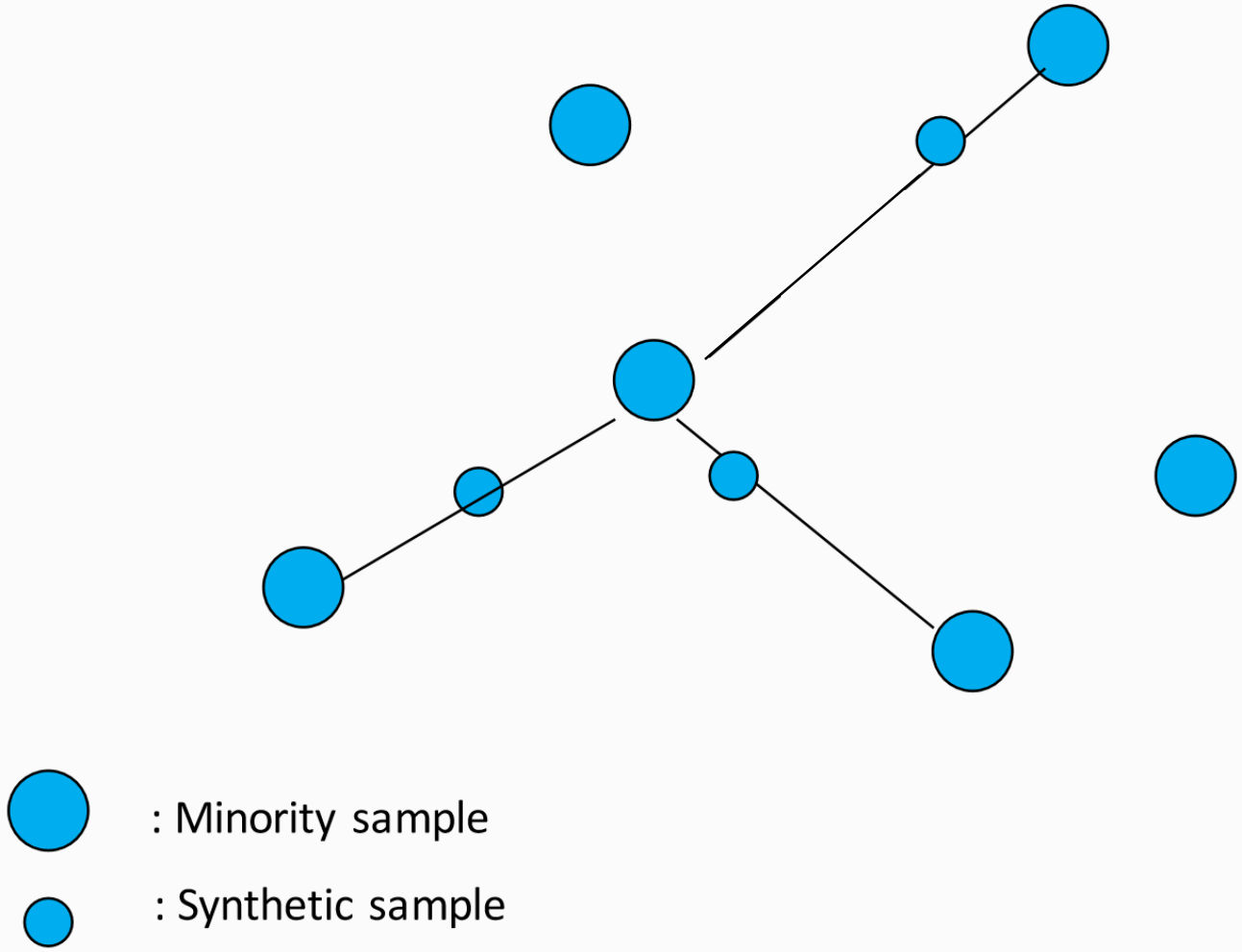
SMOTE's Informed Oversampling Procedure

For each minority Sample

- Find its k-nearest minority neighbors
- Randomly select j of these neighbors
- Randomly generate synthetic samples along the lines joining the minority sample and its j selected neighbors

  (j depends on the amount of oversampling desired)

SMOTE's Informed Oversampling Procedure I



SMOTE's Shortcomings

- Overgeneralization
  - SMOTE's procedure can be dangerous since it blindly generalizes the minority area without regard to the majority class.
  - This strategy is problematic in the case of highly skewed class distributions since, in such cases, the minority class is very sparse with respect to the majority class, thus resulting in a greater chance of class mixture.

In-class Lab

- Imbalanced Data

Clustering Analysis: General Concepts

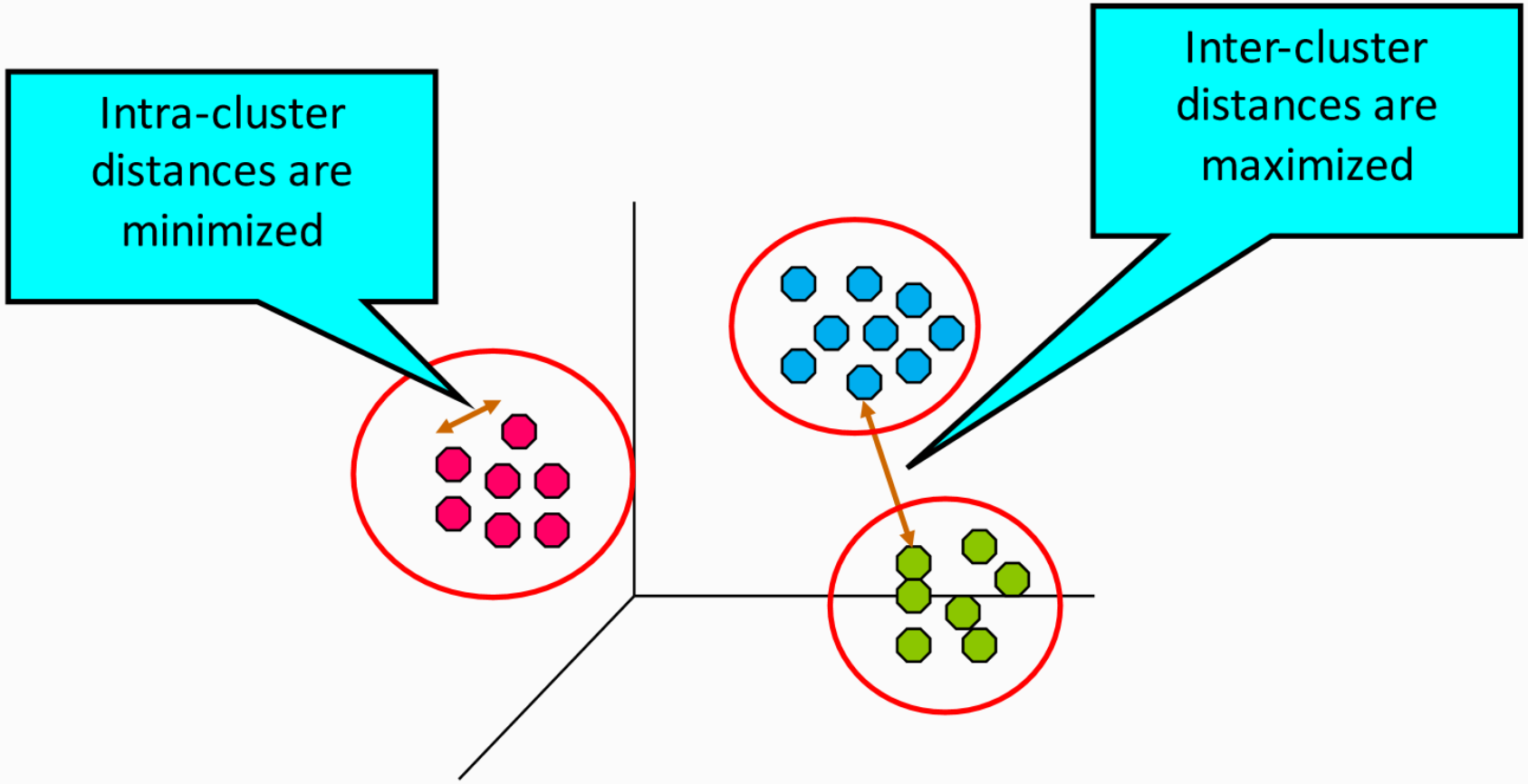
Some Definitions

- Cluster: a group of observations that are similar with each other

- Clustering Analysis: Find groups of objects (observations), such that observations within each cluster are similar with each other, and observations in different clusters are dissimilar with each other
  - Intra-cluster **similarity** is higher than inter-cluster **similarity**

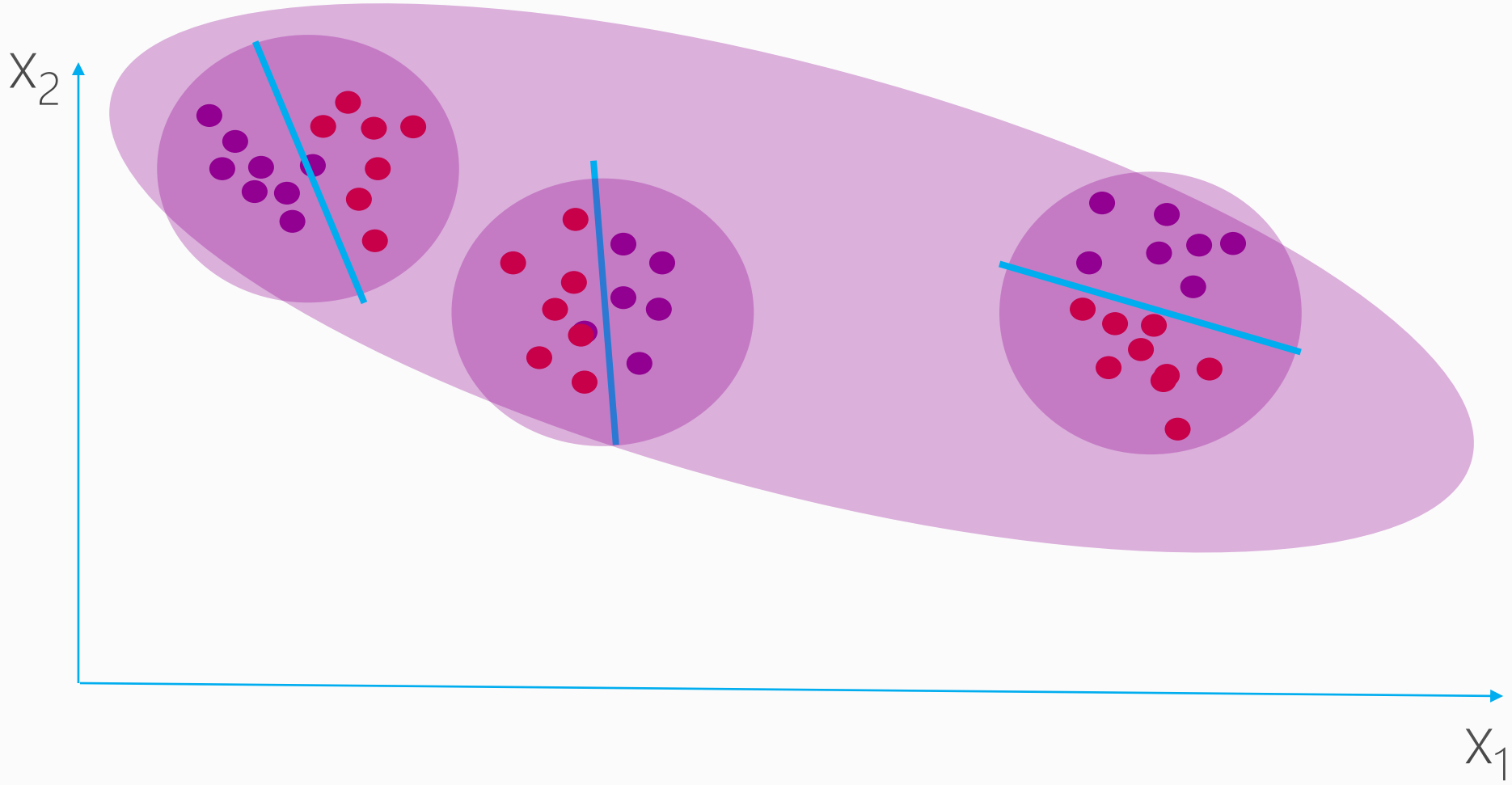
- Unsupervised Machine Learning: there is no labels telling you which observations should be in the same cluster. You need to allocate observations into clusters based on the features that describe the objects

A Toy Example



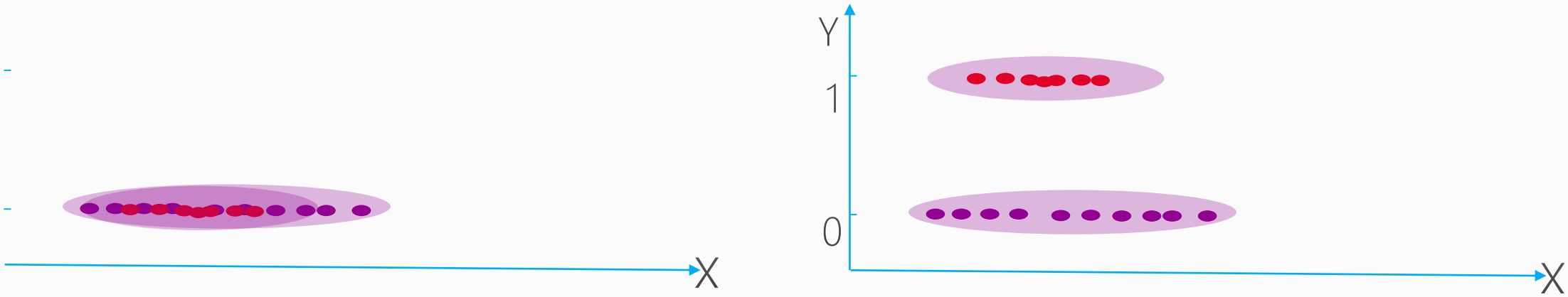
Unsupervised Machine Learning

- Might be useful for supervised machine learning

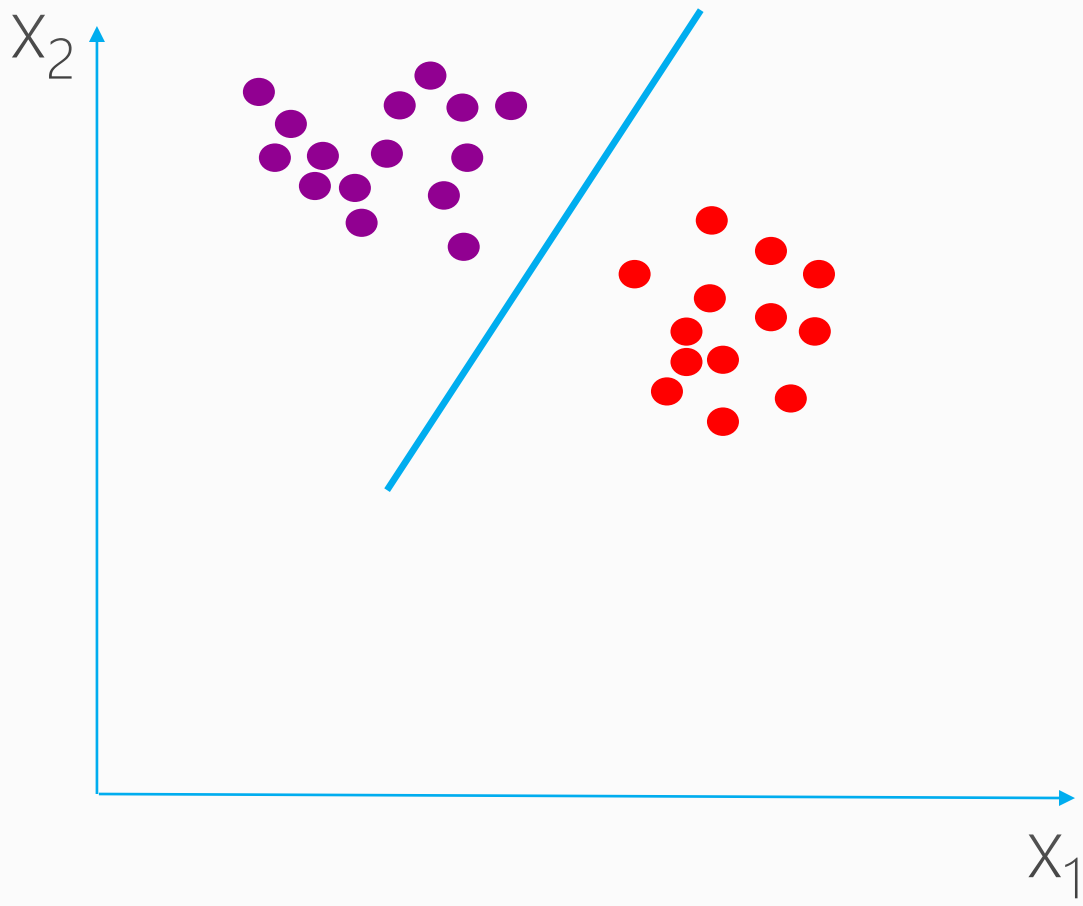


Clustering Analysis for Supervised Machine Learning (Classification) Tasks

- Do not include the label column in your clustering analysis
  - Since the observations are always clustered at the dimension of the label column
  - You will always see clustering pattern in this dimension



Can Be Used to Assess the Difficulty of your Classification Task



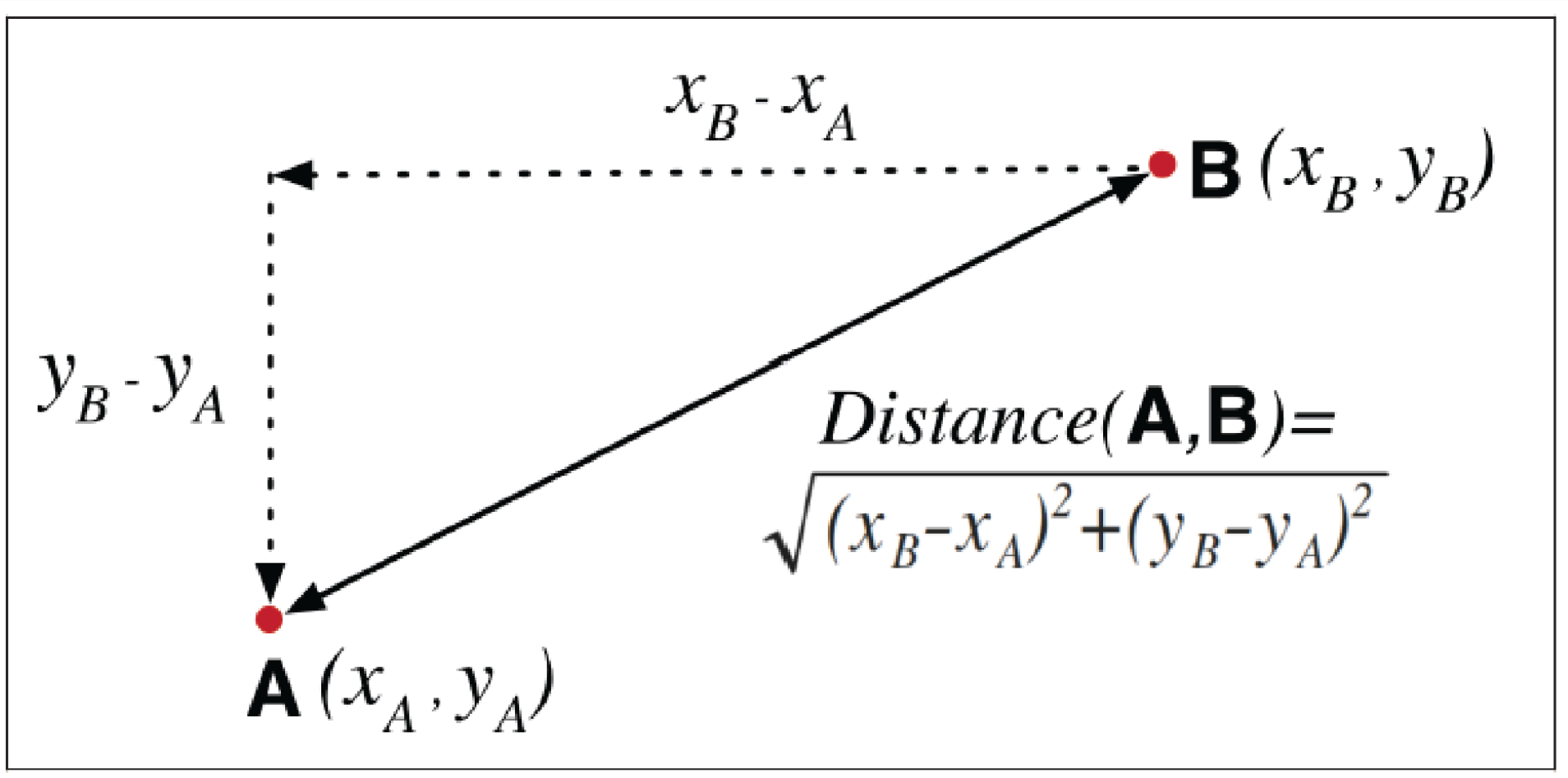
A dual assessment is the relevance of the data (feature) set with your classification task

How do we define “similarity”?

- Goal is to group together “similar” data – but what does this mean?
- No single answer, it depends on what we want to find or emphasize in the data; this is one reason why clustering is an “art”
- The similarity measure is often more important than the clustering algorithm used – don’t overlook this choice!

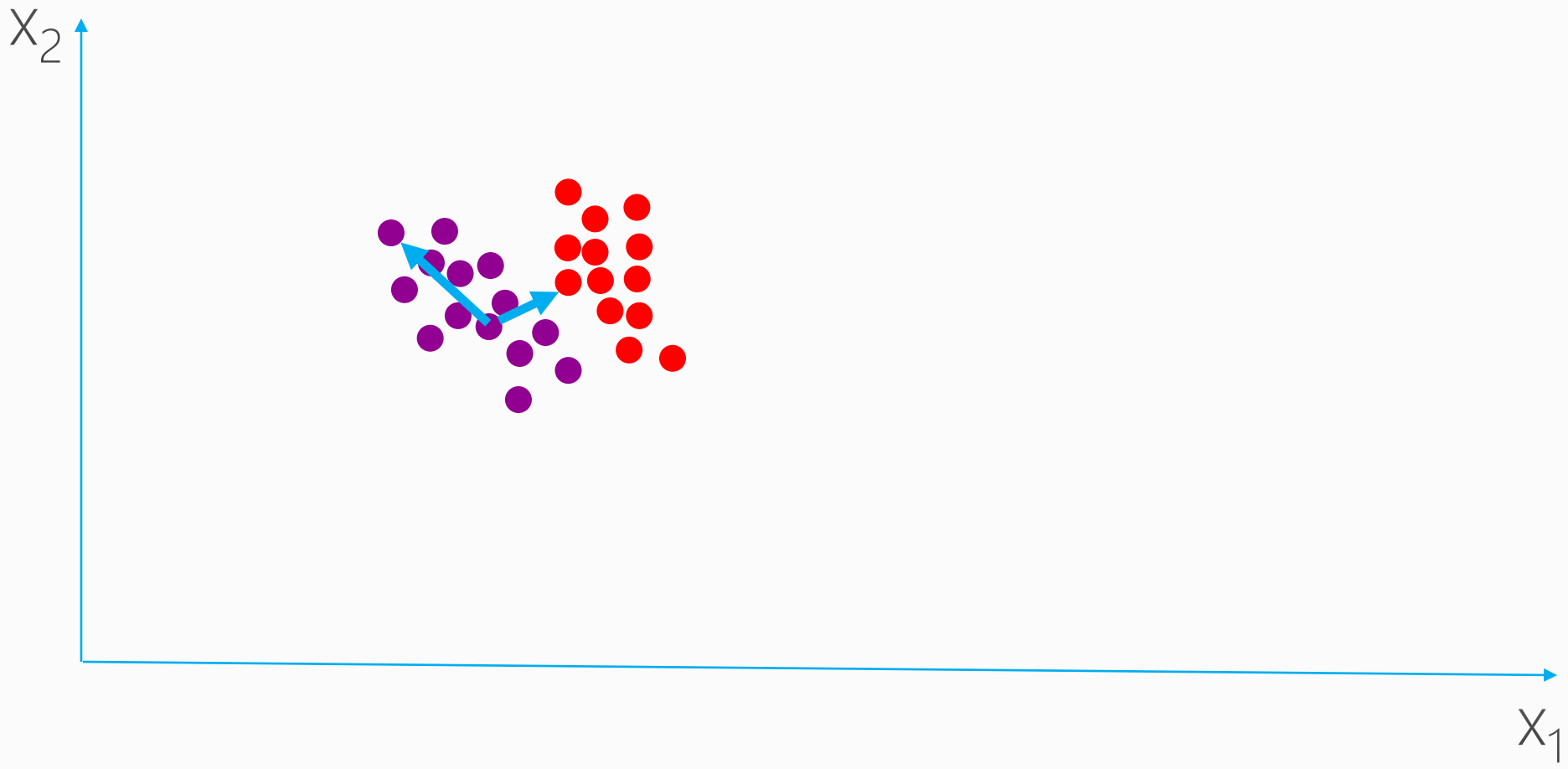
Similarity Measures

- Euclidean Distance



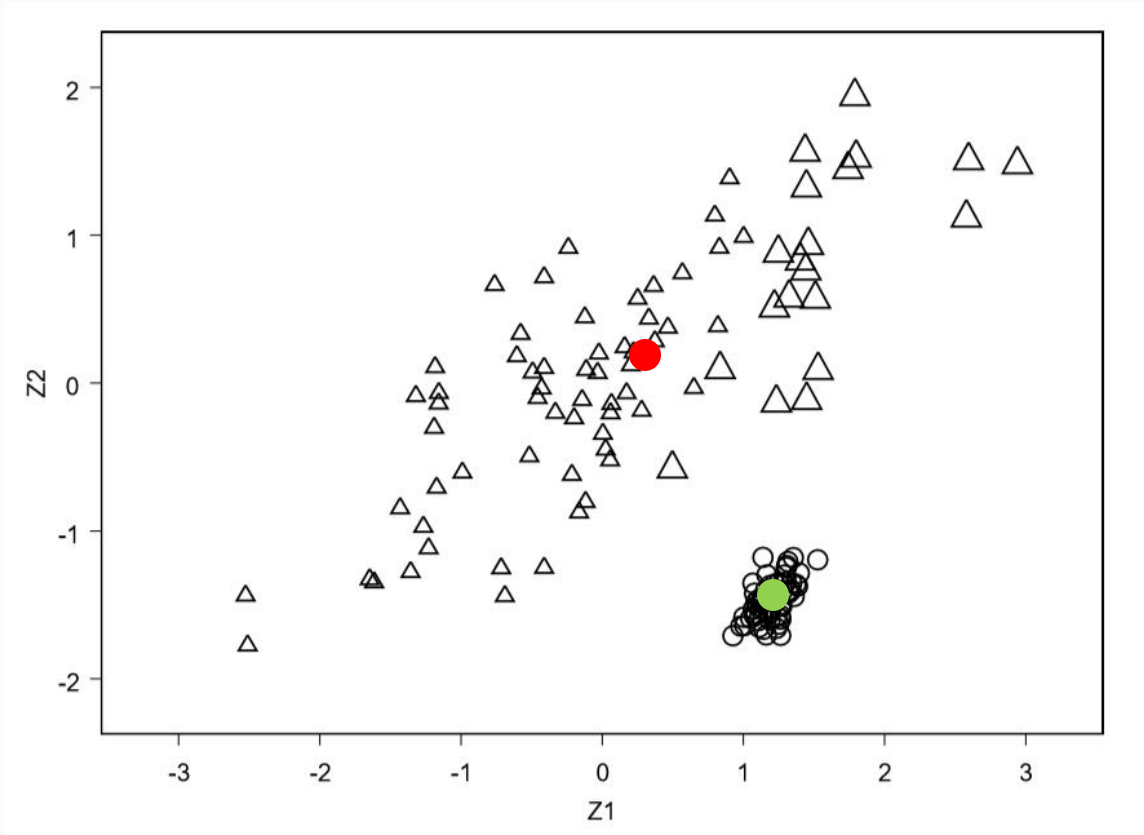
Problem of Euclidean Distance as Similarity Measurement

- It assumes that clusters are spheres



Mahalanobis Distance

$$D(\mathbf{x}_i, \mathbf{c}_k) = \{(\mathbf{x}_i - \mathbf{c}_k)' \mathbf{S}^{-1}{}_k (\mathbf{x}_i - \mathbf{c}_k)\}^{1/2}$$



Other Similarity Measures

Manhattan Distance

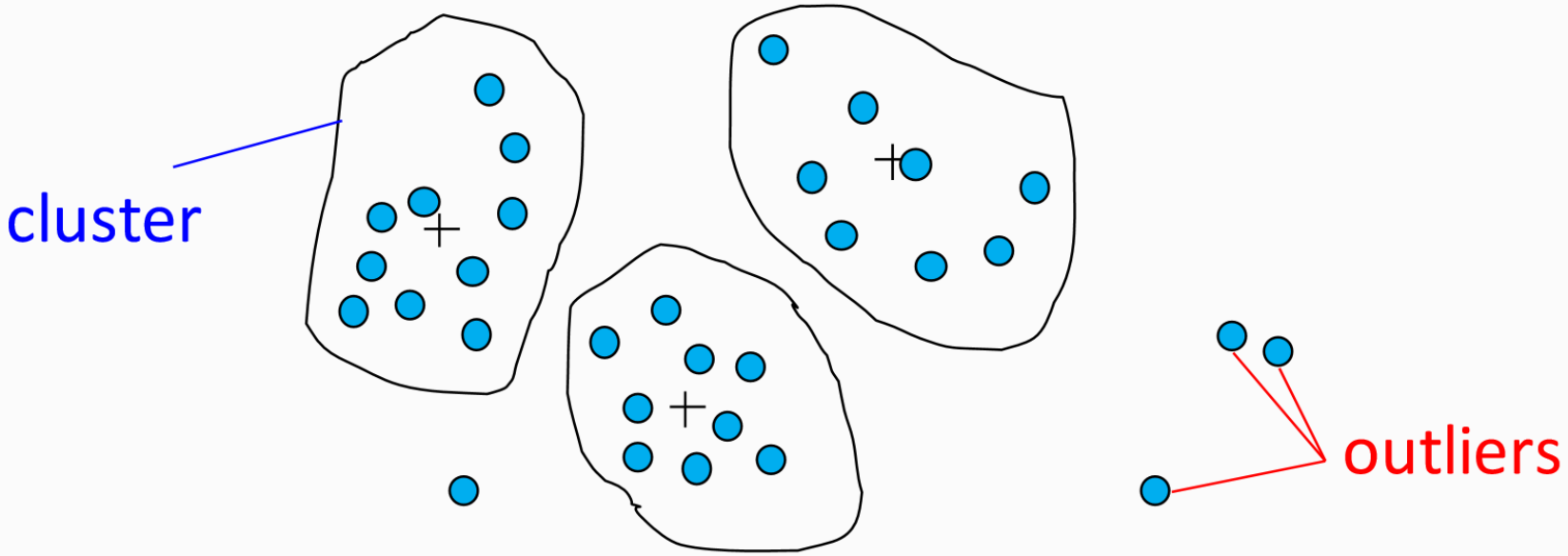
$$d_{\text{Manhattan}}(\mathbf{X},\mathbf{Y}) = \| \mathbf{X} - \mathbf{Y} \|_1 = | x_1 - y_1 | + | x_2 - y_2 | + \cdots$$

Jaccard Distance

$$d_{\text{Jaccard}}(X, Y) = 1 - \frac{| X \cap Y |}{| X \cup Y |}$$

Outliers

- **Outliers** are **objects that do not belong to any cluster** or form clusters of very small cardinality



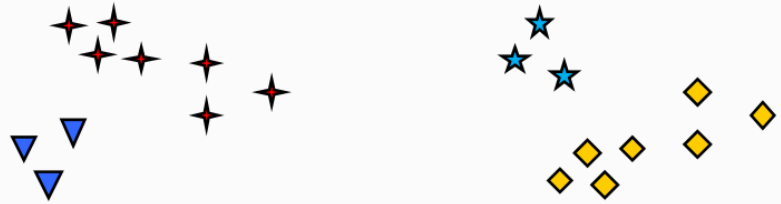
In some applications we are interested in discovering outliers, not clusters (**outlier analysis**)

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

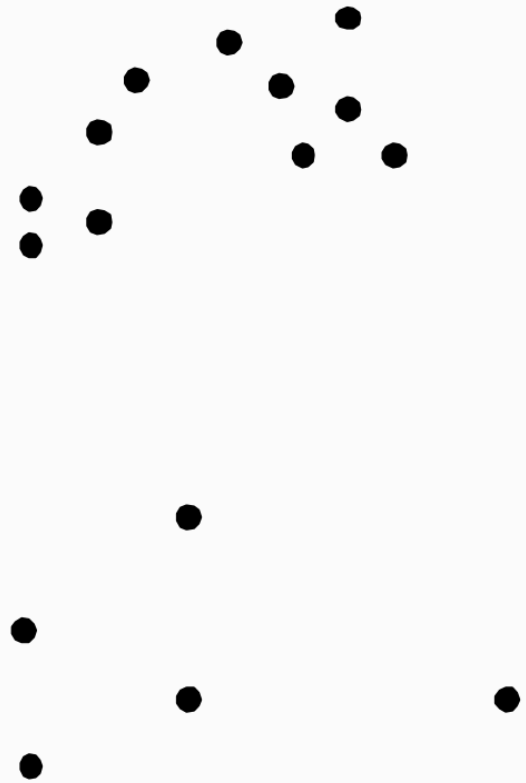


Four Clusters

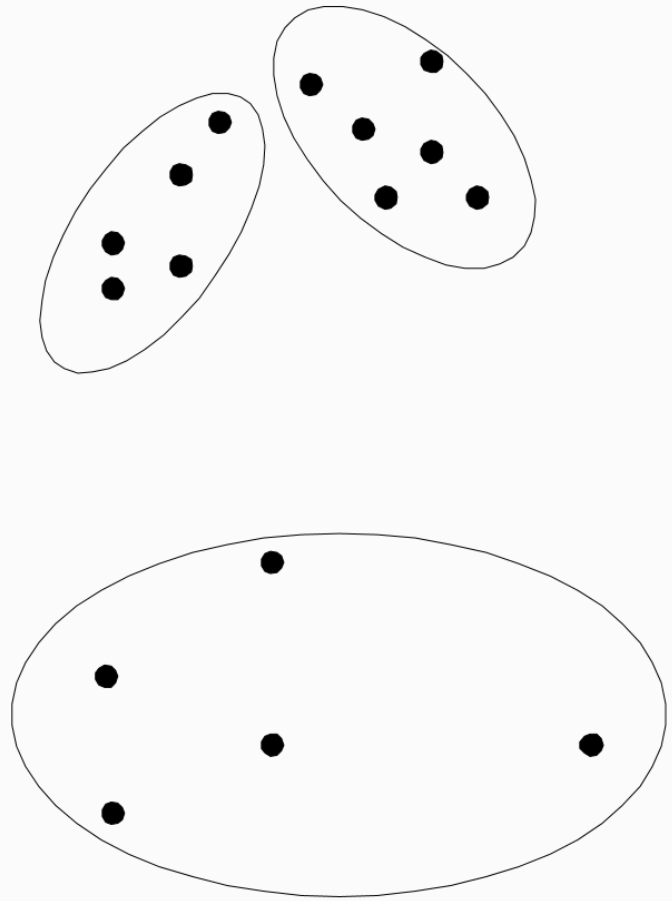
Types of Clusterings

- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

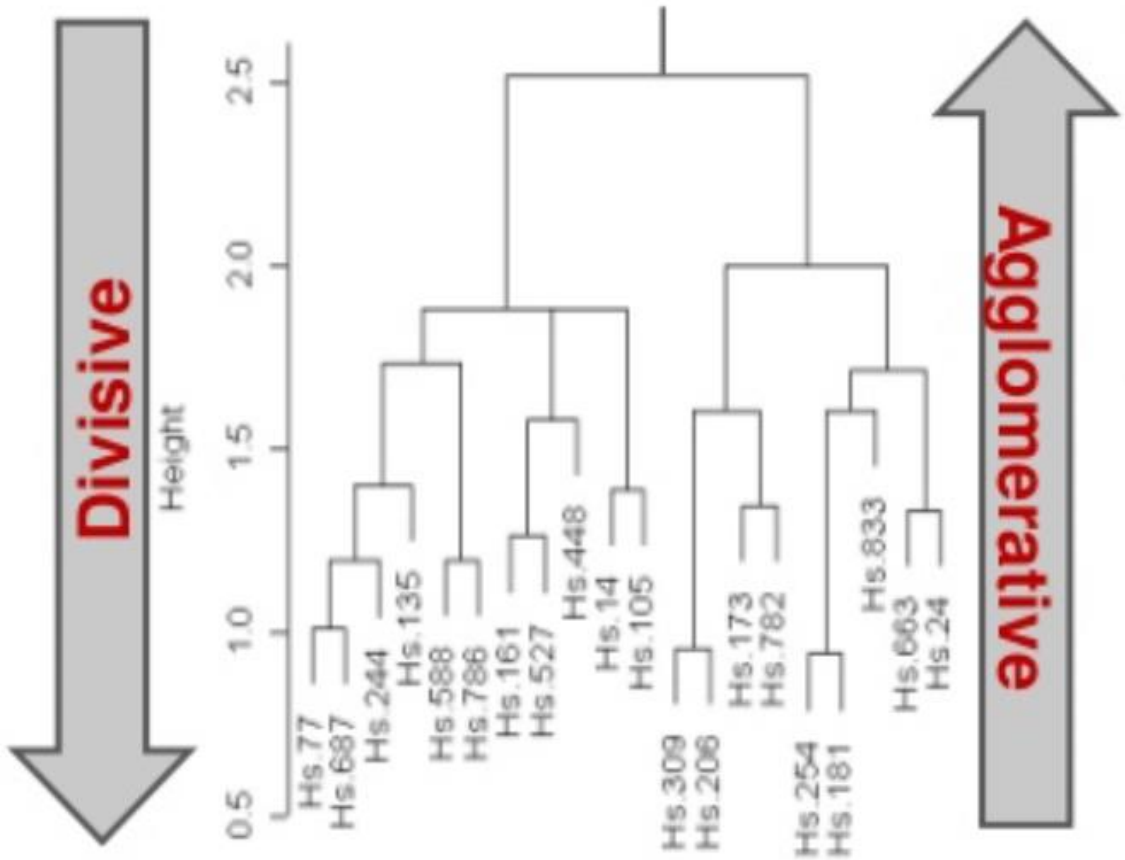


Original Points



A Partitional Clustering

HIERARCHICAL CLUSTERING TREE, DENDOGRAM



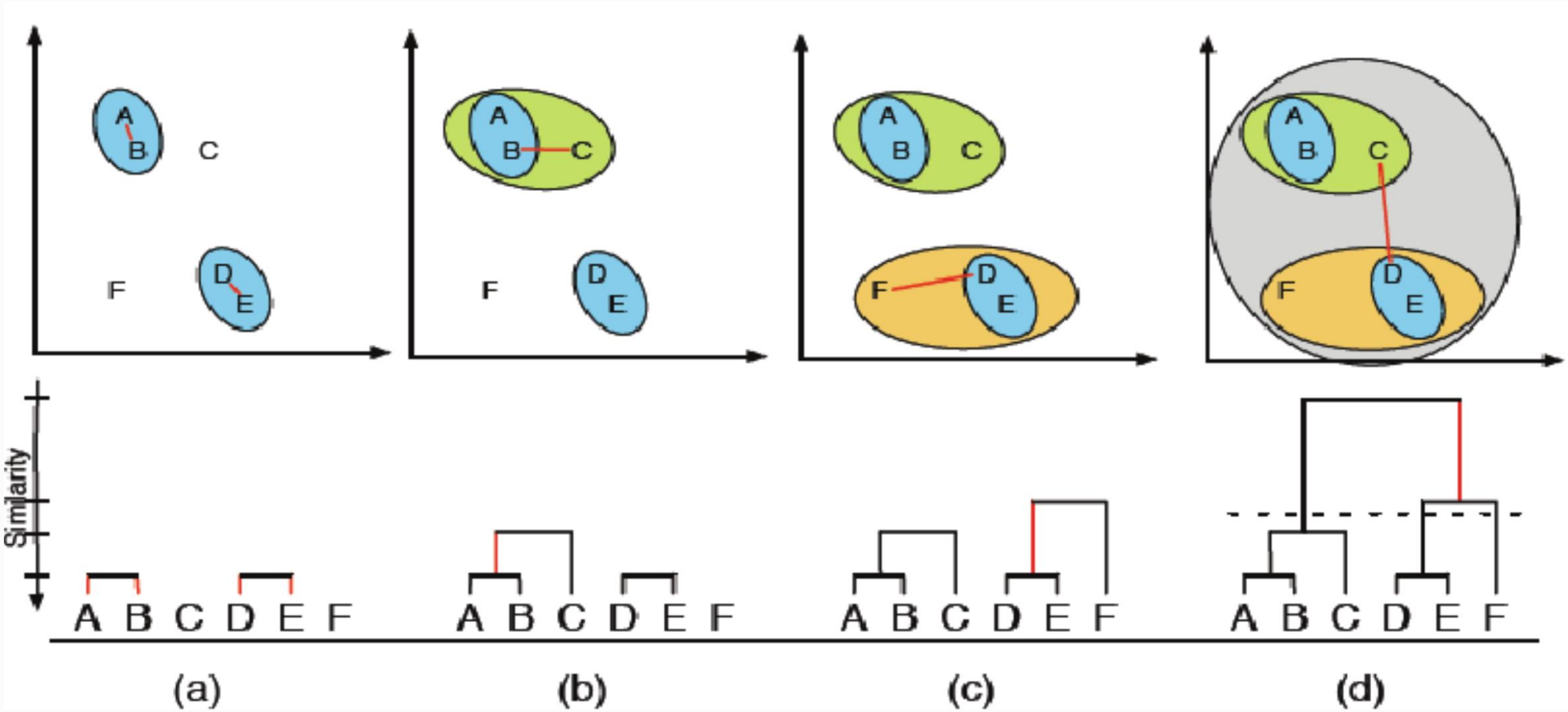
The hierarchy of clustering is given as a clustering tree or dendrogram

- leaves of the tree represent the individual objects
- internal nodes of the tree represent the clusters

Two main types of hierarchical clustering

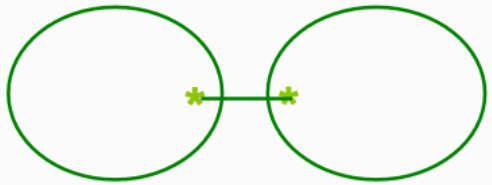
- **agglomerative (bottom-up)**
  - place each object in its own cluster (a singleton)
  - merge in each step the two most similar clusters until there is only one cluster left or the termination condition is satisfied
- **divisive (top-down)**
  - start with one big cluster containing all the objects
  - divide the most distinctive cluster into smaller clusters and proceed until there are *n* clusters or the termination condition is satisfied

Example of Agglomerative Hierarchical Clustering

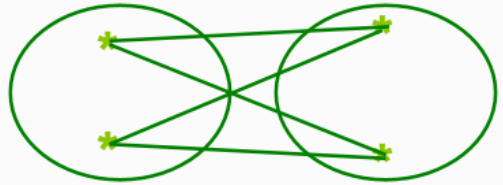


How to Determine Which Two Clusters to Merge in Agglomerative Hierarchical Clustering Analysis?

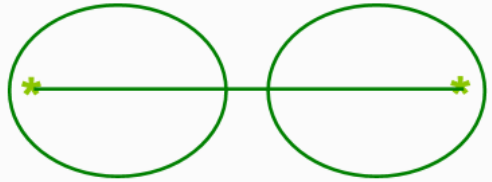
- Each cluster (before merge) might have multiple objects
- We need a measure to describe the similarity between two clusters



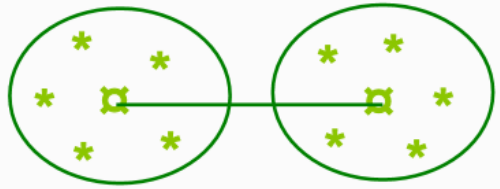
Single Linkage:
Minimum distance



Average Linkage:
Average distance



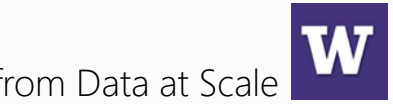
Complete Linkage:
Maximum distance



Centroid method:
Distance between centres

- Single linkage tends to create bigger clusters than complete linkage.
- Centroid method is the most popular linkage in use
- Trial and fail to see which measure gives you the most desired clusters

K-Means Clustering



K-means Clustering

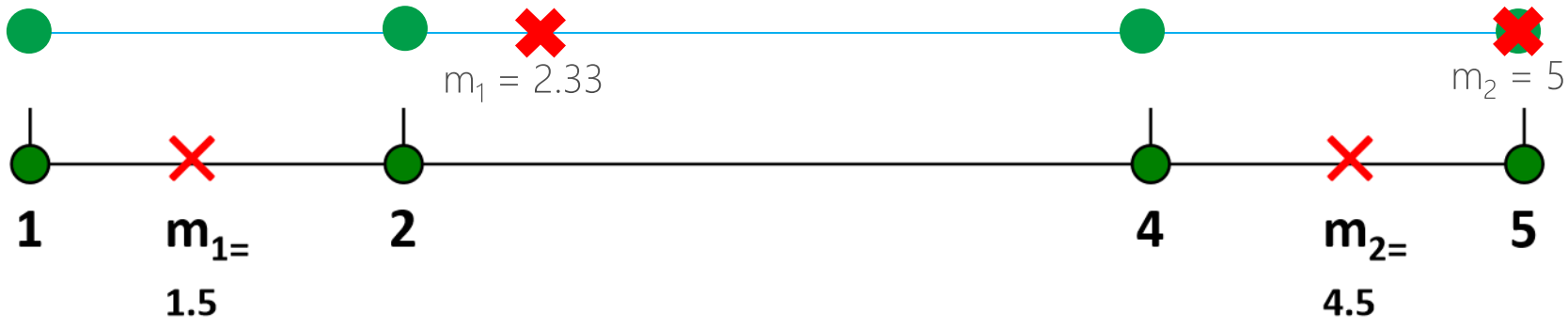
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Performance Metrics of K-Means Clustering: SSE

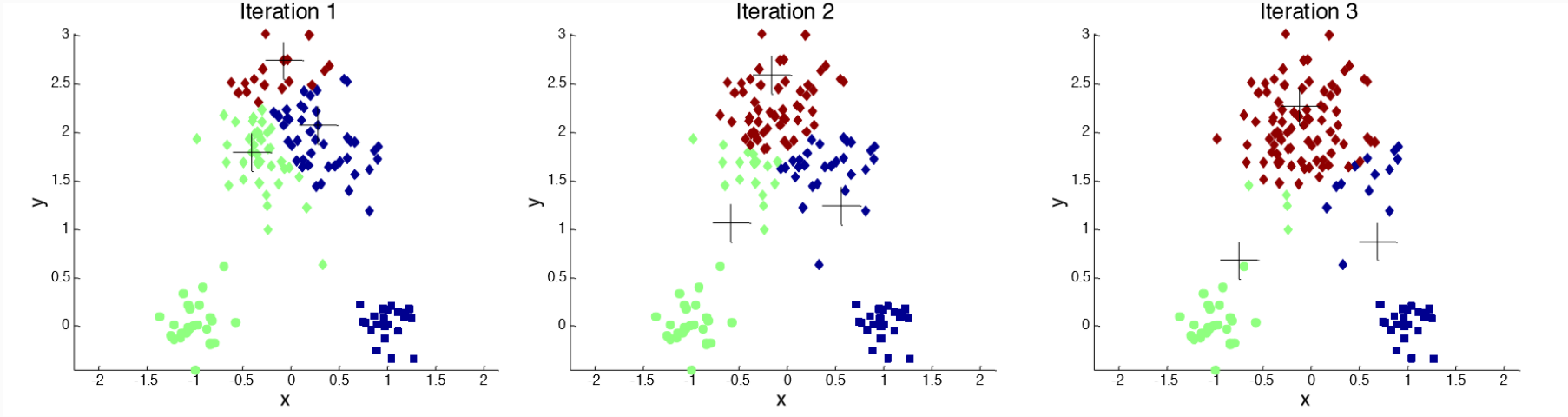
- Suppose the centroid of cluster C_j is m_j
- For each object x in C_j, compute the squared error between x and the centroid m_j
- Sum up the error of all the objects

$$SSE = \sum_{j} \sum_{x \in C_j} (x - m_j)^2$$

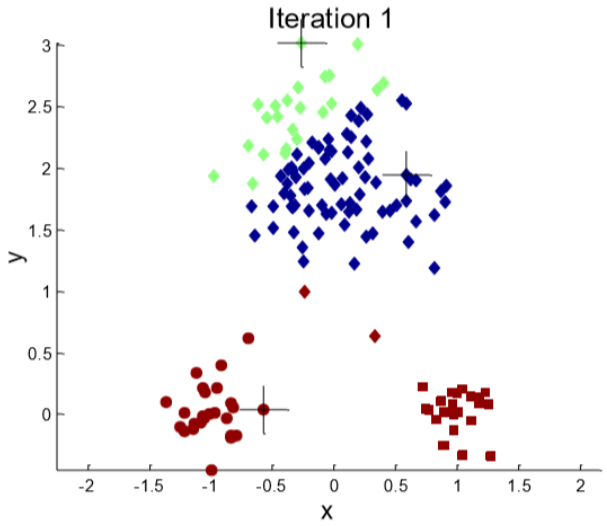


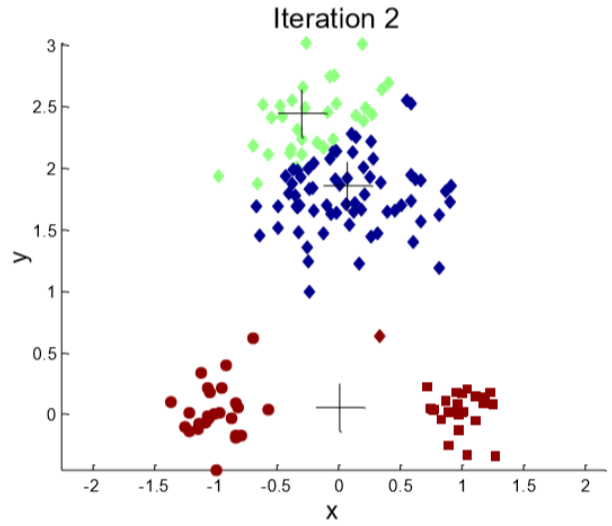
SSE = $(1-2.33)^2+(2-2.33)^2+(4-2.33)^2+(5-5)^2 = 4.67$

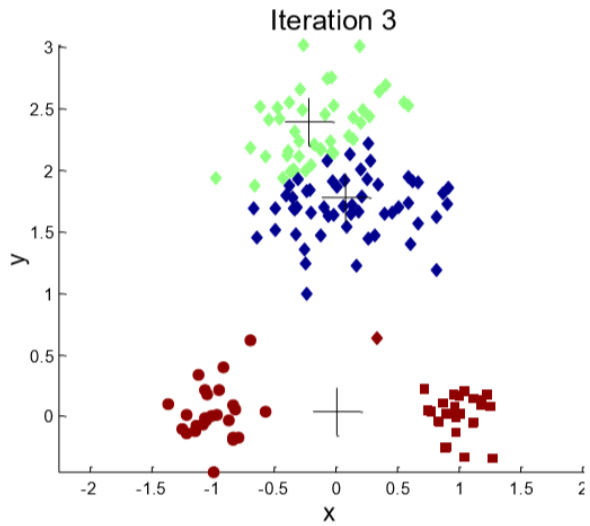
Importance of Choosing Initial Centroids

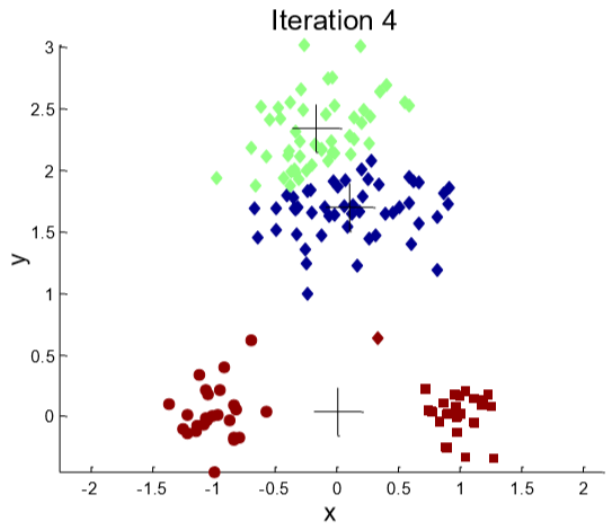


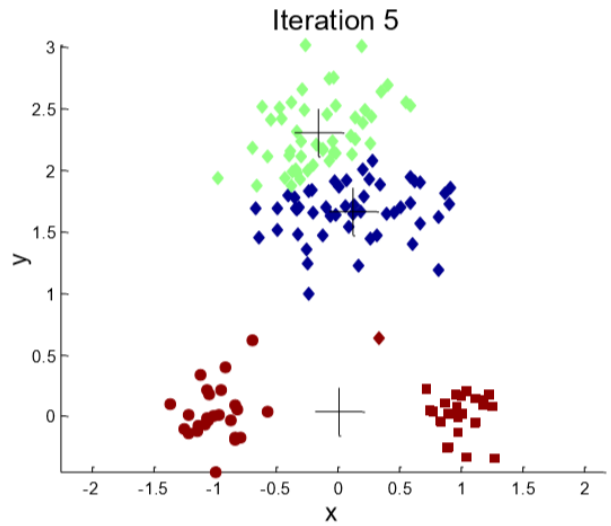
Importance of Choosing Initial Centroids











Solutions to Centroid Initialization Problem

In order to solve the centroid initialization problem, usually we do k-means for multiple times with fixed k

- Each time calculate SSE
- Choose the run with the minimal SSE as the final clustering result

Data Preprocessing in K-means Clustering

- K-means clustering requires all variables are numerical
- Numerical variables need to be scaled to remove the impact of scales of different variables
- Non-numerical variables?
  - Ordinal non-numerical variable, reasonable to represent the values as 1, 2, 3, …. For instance, education middle school, high school, college, masters, Ph.D. can be replaced by 1, 2, 3, 4, and 5
  - Other non-numerical variable, one-hot encoding.

K-Means Clustering: How to Determine K?

- Is minimizing SSE a good way to choose K?
  - If you make each observation as a single cluster, SSE=0

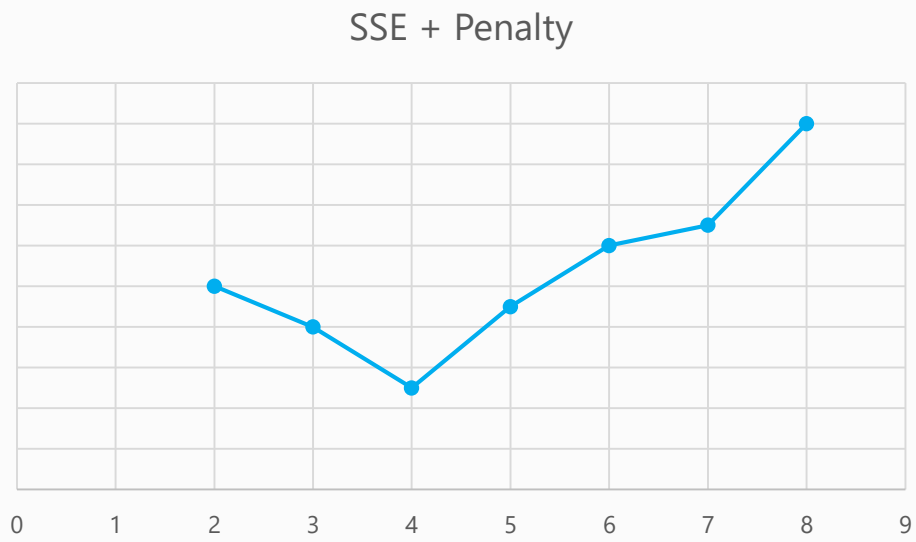
$$SSE = \sum_{j} \sum_{x \in C_j} (x - m_j)^2$$

- Consider regularization:
  - We can choose to minimize

$$\sum_{j=1}^{k} \sum_{x \in C_j} (x - m_j)^2 + \lambda \times N_k$$

for k = 1, 2, ..., K, where K is a reasonably maximal possible number of clusters,

N_k : number of independent parameters to be estimated in the k models, assuming that each cluster is generated by an underlying multivariate normal distributed model



- Still an on-going research topic, and very subjective

In-class Lab

- Clustering Analysis