APPLIED MACHINE LEARNING

Clustering

Part III – Evaluation Techniques

Evaluation of Clustering Methods

Clustering methods rely on hyper parameters

Number of clusters, elements in the cluster, distance metric

→ Need to determine the goodness of these choices

Clustering is unsupervised classification

→ Do not know the real number of clusters and the data labels
→ Difficult to evaluate these choices without *ground truth*

Evaluation of Clustering Methods

Two types of measures: Internal versus external measures

Internal measures rely on measures of similarity:

- (low) intra-cluster distance versus (high) inter-cluster distances
- Internal measures are problematic as the metric of similarity is often already optimized by the clustering algorithm.

External measures rely on ground truth (class labels):

- Given a (sub)-set of known class labels compute similarity of clusters to class labels.
- In real-world data, it's hard/infeasible to gather ground truth.

Evaluation of Clustering Methods

Two types of measures: Internal versus external measures

Internal measures rely on datapoints only and on a good choice of measure of similarity:

E.g. of internal measures RSS, BIC and AIC

External measures rely on ground truth (class labels):

- E.g. F1-

The RSS measure for clustering

Residual Sum of Square RSS is an internal measure

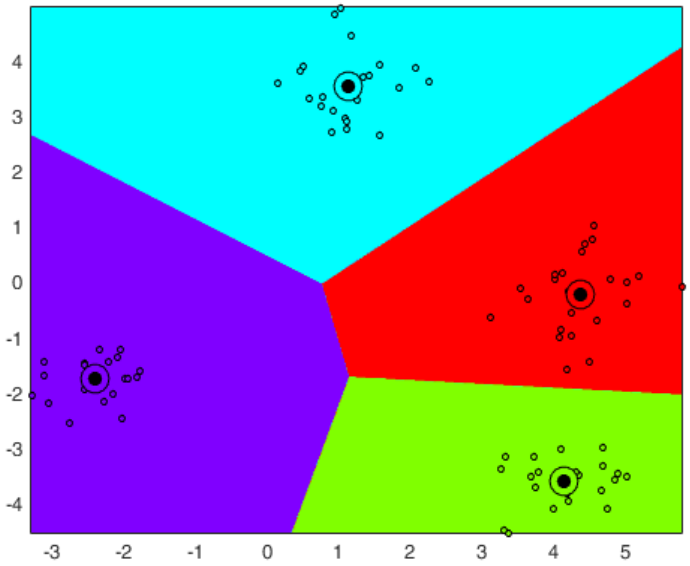
It computes the distance (in norm-2) of each datapoint from its centroid for all clusters.

$$\text{RSS}=\sum_{k=1}^{K}\sum_{x^i \in c_k} \left\| x^i - \mu^k \right\|_2$$

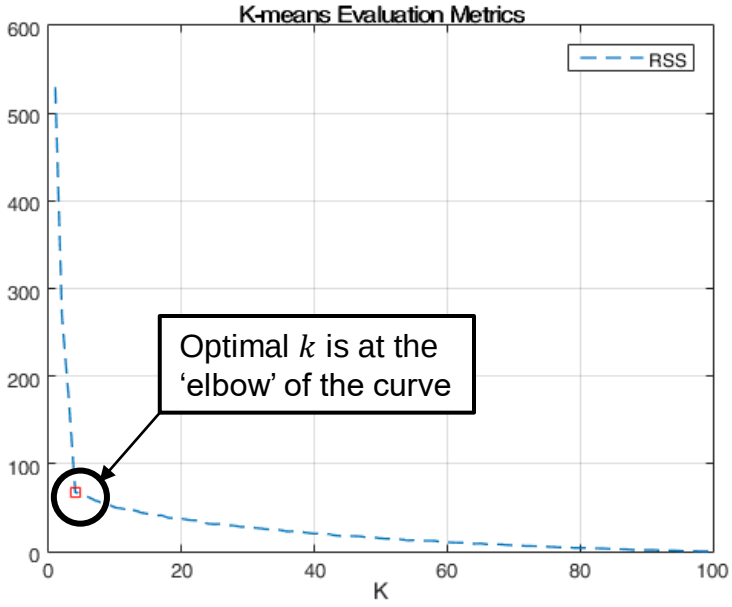
K-means Clustering: Examples

Procedure: Run K-means – increase monotonically number of clusters – take best run in each case;

Use RSS measure to measure improvement in clustering → determine a plateau



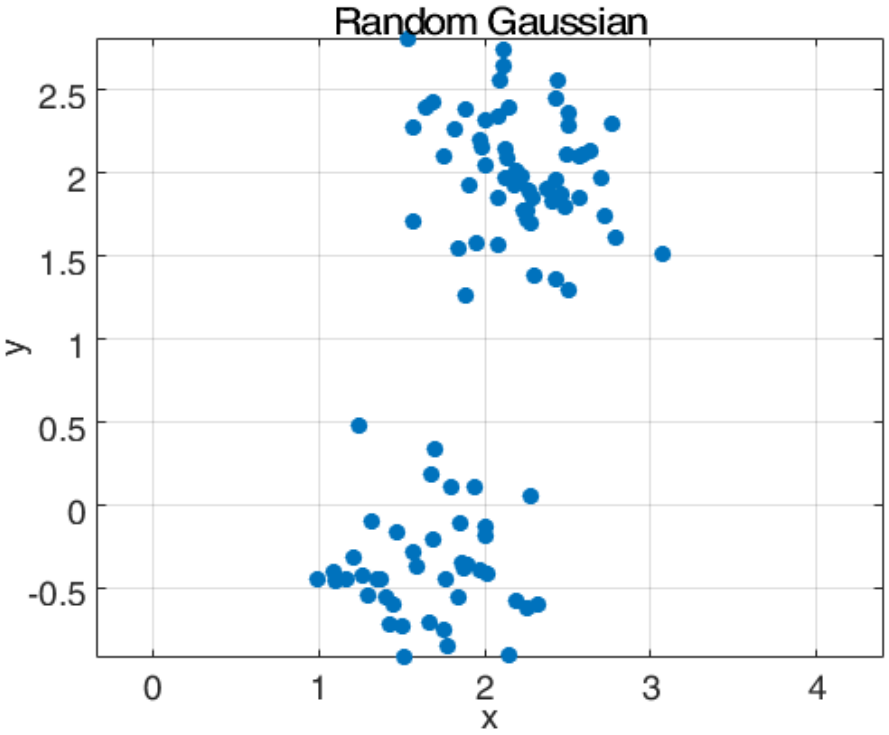
M: 100 *datapoints*
N: 2 *dimensions*



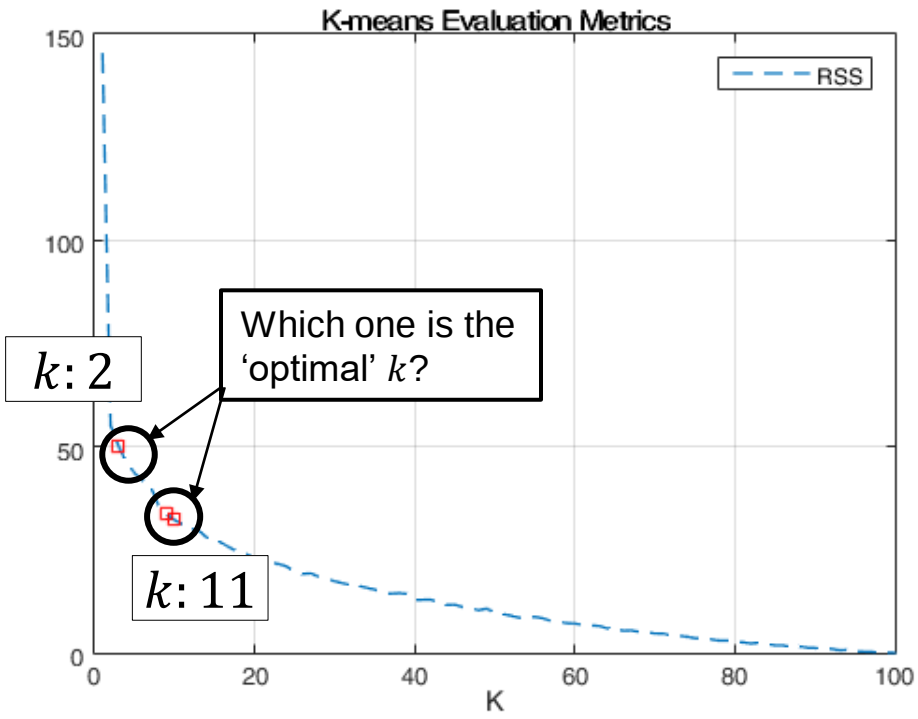
k: 4 *clusters*

K-means Clustering: Examples

The ‘elbow’ or ‘plateau’ method for choosing the optimal k from the RSS curve can be unreliable for certain datasets:



M: 100 *datapoints*
K: 3 *dimensions*



We don’t know! We need an additional penalty or criterion!

Other Metrics to Evaluate Clustering Methods

- Aikaike Information Criterion: $AIC = -2\ln L + 2B$
- Bayesian Information Criterion: $BIC = -2\ln L + B\ln(M)$

Penalty for increase in computational costs due to number of parameters and number of datapoints

L: maximum likelihood of the model

B: number of free parameters

M : number of datapoints

AIC and BIC determine how good the model fits the dataset (maximum-likelihood). The measure is balanced by how many parameters are needed to get a good fit.

As the number of datapoints (observations) increase, BIC assigns more weights to simpler models than AIC.

Low BIC implies either fewer explanatory variables, better fit, or both.

Choosing AIC versus BIC depends on the application:
Is the purpose of the analysis to make predictions, or to decide which model best represents reality? AIC may have better predictive ability than BIC, but BIC finds a computationally more efficient solution.

AIC for K-Means

For the particular case of K-means, we do not have a maximum likelihood estimate of the model:

$$AIC = -2\ln(L) + 2B$$

L : likelihood of model

B: number of free parameters

However, we can formulate a metric based on the *RSS* that penalizes for *model complexity* (# K-clusters), conceptually following *AIC*:

$$AIC_{RSS} = RSS + B$$

Number of free parameters *B=(K*N)*
K: # clusters
N: # dimensions

$$\text{RSS} = \sum_{k=1}^{K} \sum_{x \in C_k} \left| x - \mu^k \right|^2$$

Weighting Factor

BIC for K-Means

For the particular case of K-means, we do not have a maximum likelihood estimate of the model:

$$BIC = -2\ln(L) + \ln(M)B$$

However, we can formulate a metric based on the *RSS* that penalizes for *model complexity* (# K-clusters, # M-datapoints), conceptually following *BIC*:

$$BIC_{RSS} = RSS + \ln(M)\,B$$

Number of free parameters *B=(K*N)*
K: # clusters
N: # dimensions

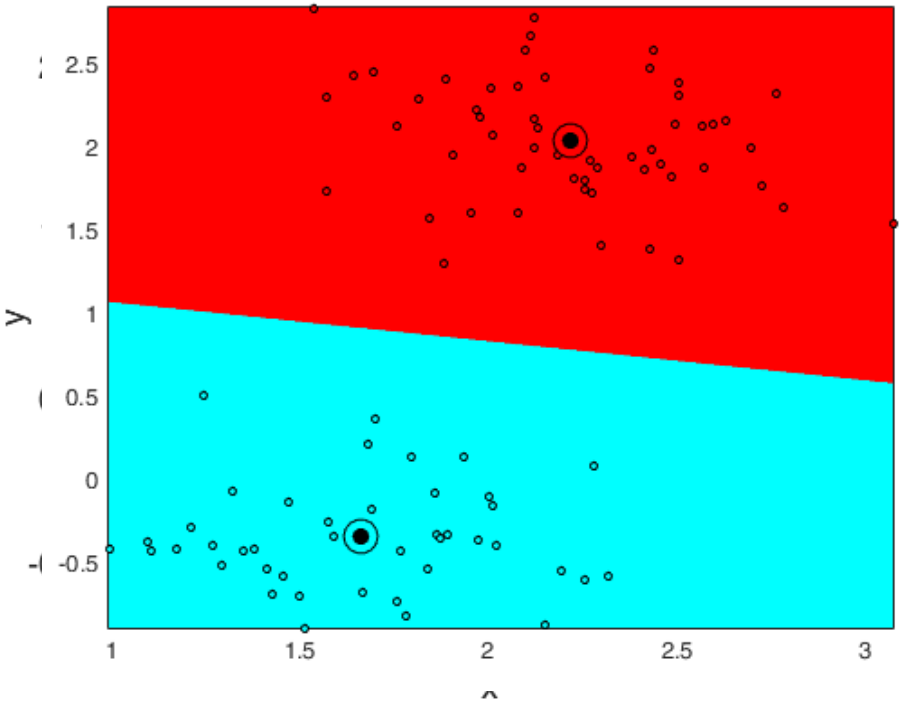
$$\text{RSS} = \sum_{k=1}^{K} \sum_{x \in C_k} \left| x - \mu^k \right|^2$$

Weighting factor penalizes wrt. # datapoints (i.e. computational complexity)

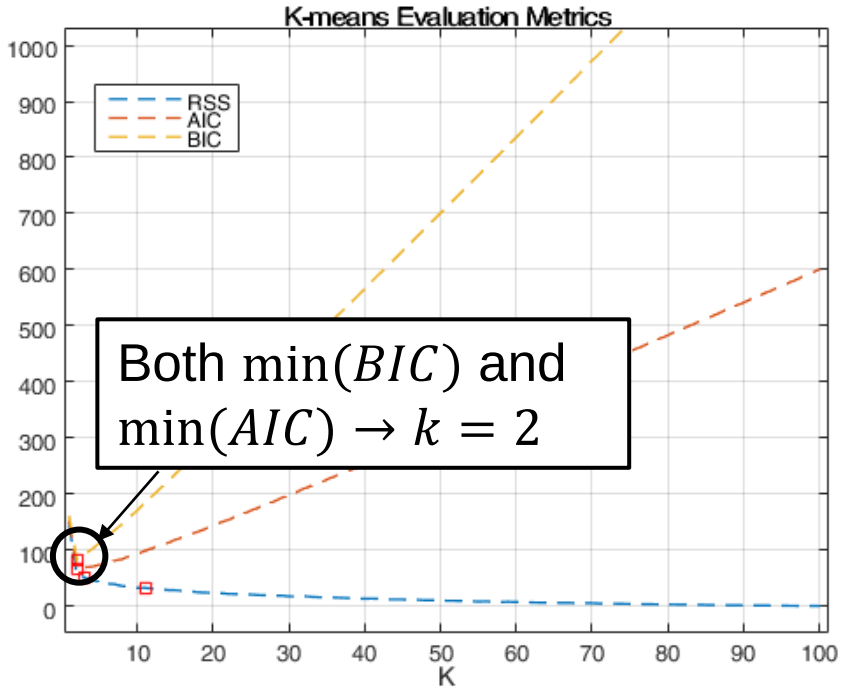
K-means Clustering: Examples

Procedure: Run K-means – increase monotonically number of clusters – run K-means with several initialization and take best run;

- use AIC/BIC curves to find the optimal k, which is $\min(AIC)$ or $\min(BIC)$



M: 100 *datapoints*
N: 3 *dimensions*

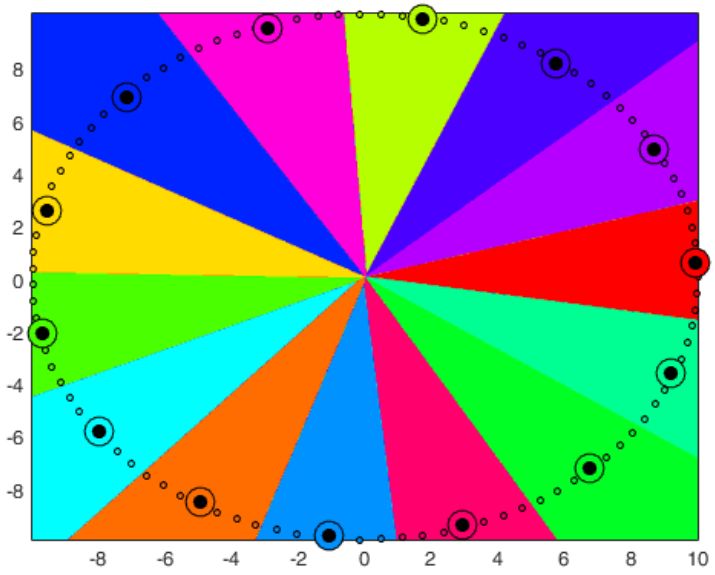


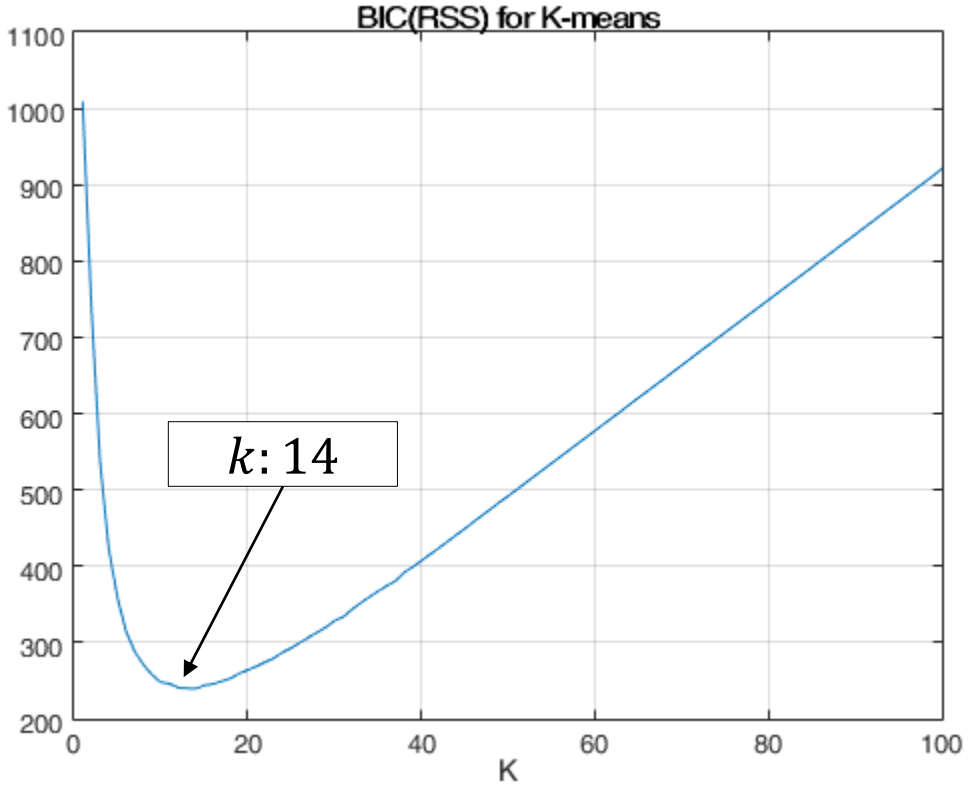
k: 2 *clusters*

BIC for K-Means

$$BIC_{RSS} = RSS + \ln(M)\,(K \cdot N)$$

M: 100 *datapoints*
N: 2 *dimensions*
K: 14 *clusters*

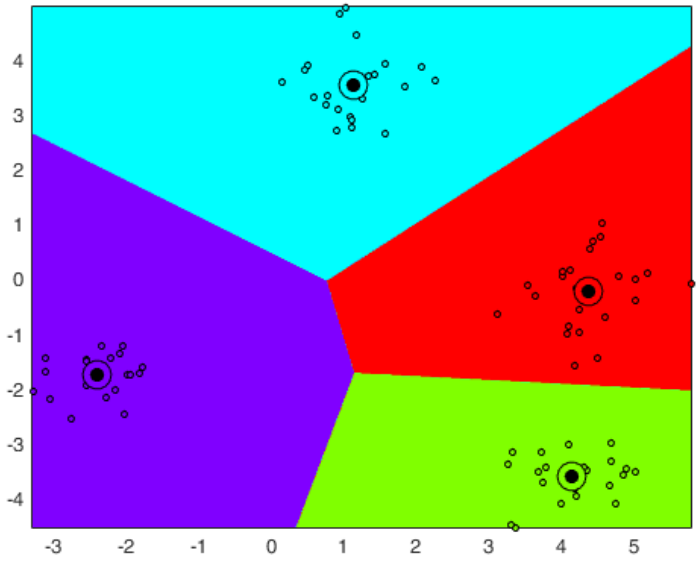


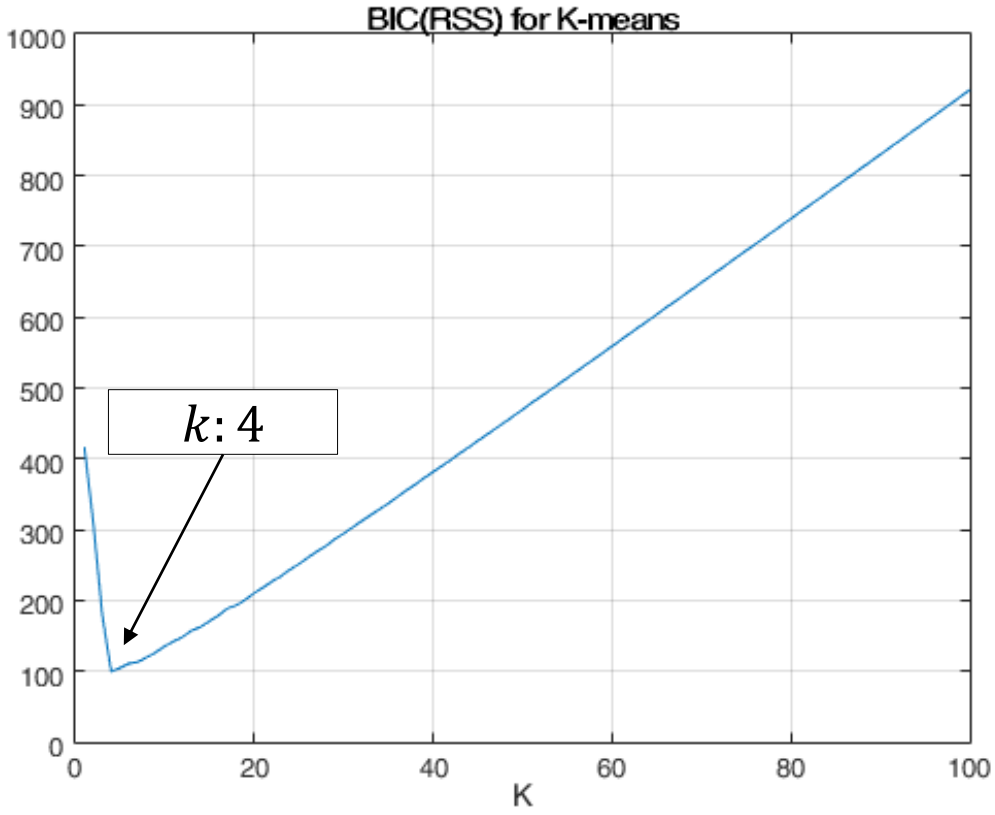


BIC for K-Means

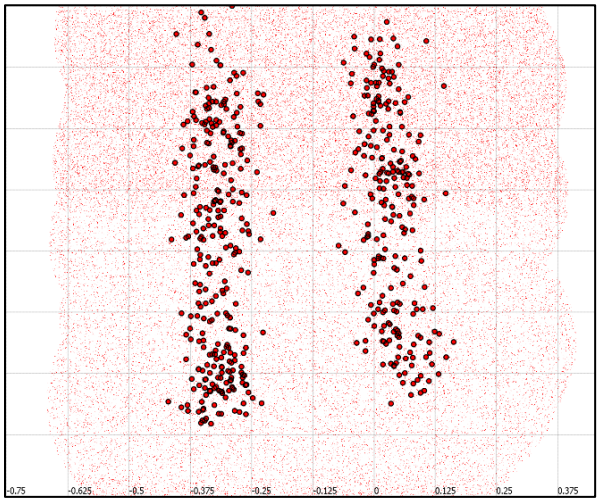
$$BIC_{RSS} = RSS + \ln(M)\,(K \cdot N)$$

M: 100 *datapoints*
N: 2 *dimensions*
K: 4 *clusters*

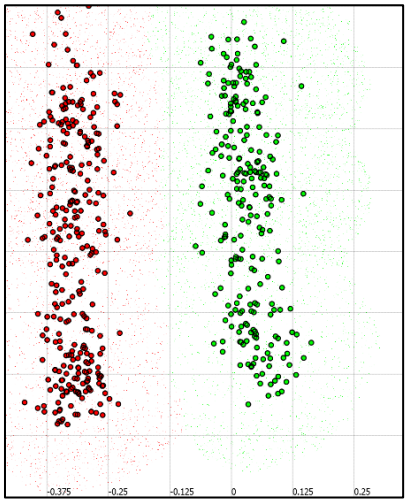




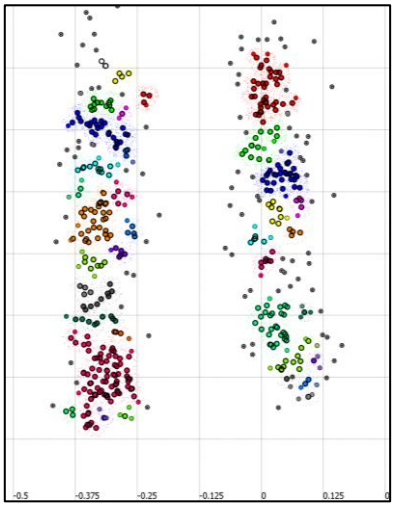
AIC / BIC for DBSCAN



DBSCAN large ε



DBSCAN medium ε



DBSCAN small ε

Compute centroid of each cluster and apply AIC/BIC for K−means

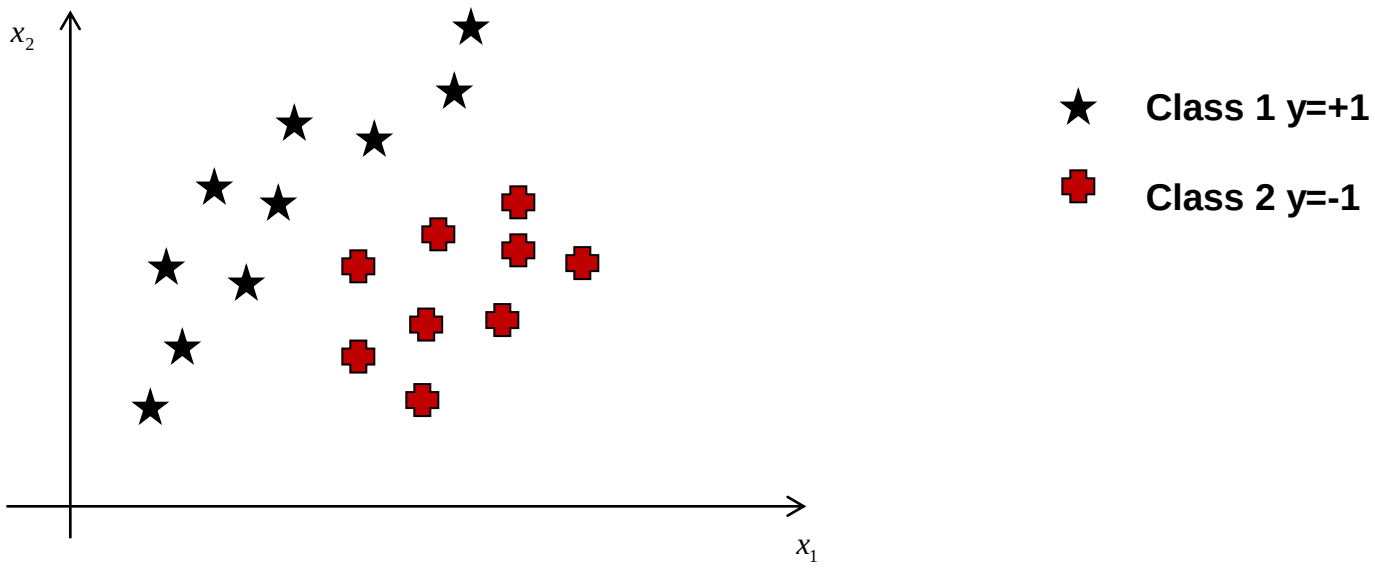
	DBSCAN large ε	DBSCAN medium ε	DBSCAN small ε
RSS	43	26	0.5
BIC	42	34	78
AIC	69	51	24

Semi-Supervised Clustering

Labels a subset of datapoints:

- Provides information about number of classes / clusters
- Provides indication of what is a member of the class

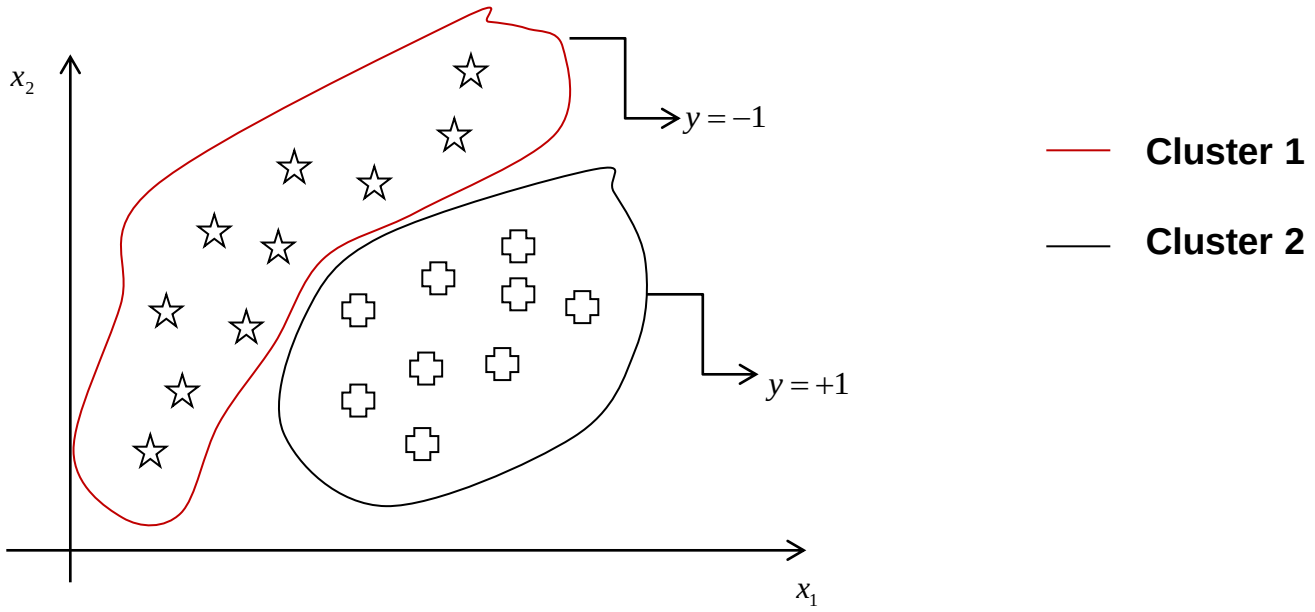
Semi-supervised clustering



All points are labelled : binary classification problem

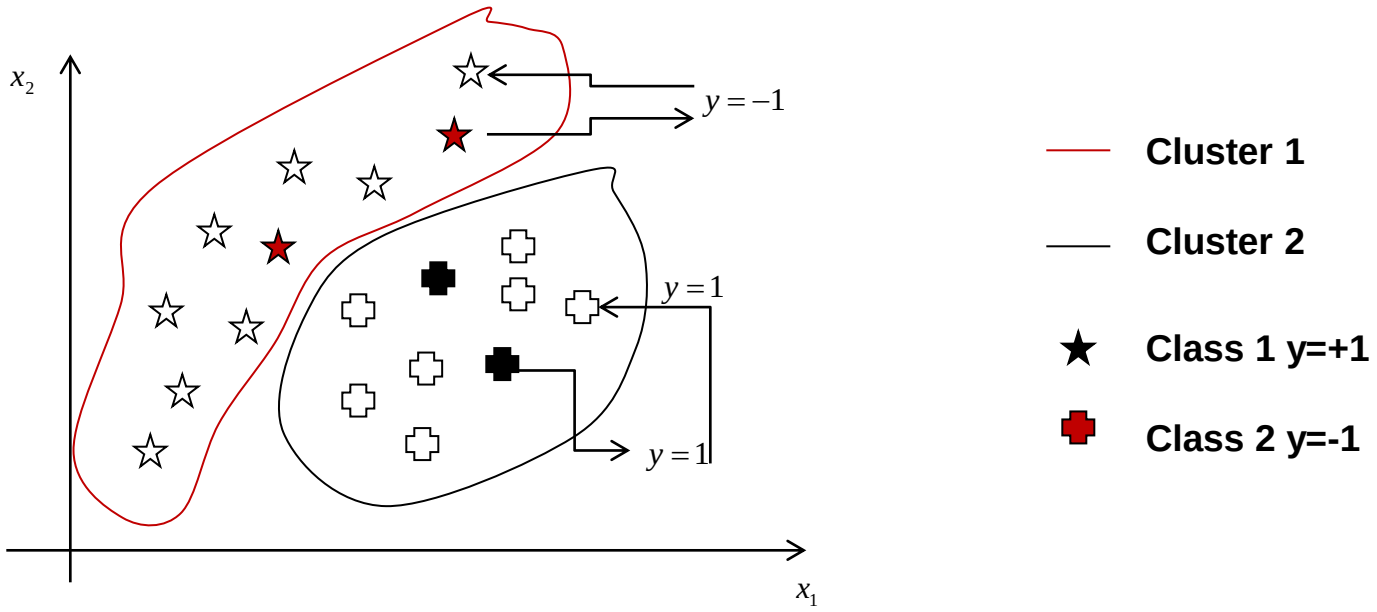
Each datapoint x has an associated label y=+/-1

Semi-supervised clustering



No points are labelled: class is inferred from cluster label

Semi-supervised clustering



- A subset of the date points are labelled.
- The rest of the points are unlabeled.
- The class label for the unlabeled points is inferred from the label of the points in the same cluster

Semi-Supervised Clustering

Clustering F1-Measure :

F_1 provides a measure of how good the clustering is:

$F_1 \in [0,1]$

$F_1 = 1$ is the optimum.

Tradeoff between clustering correctly all datapoints of the same class in the same cluster and making sure that each cluster contains points of only one class.

F1-measure

(careful: similar but not the same F-measure as the F-measure we will see for classification!)

Tradeoff between clustering correctly all datapoints of the same class in the same cluster and making sure that each cluster contains points of only one class.

M : nm of labeled datapoints

$C = \{c_i\}$: the set of classes

K : nm of clusters,

n_{ik} : nm of members of class c_i and of cluster k

$$R(c_i, k) = \frac{n_{ik}}{|c_i|}$$

Recall : proportion of datapoints in cluster k correctly classified as c_i

$$P(c_i, k) = \frac{n_{ik}}{|k|}$$

Precision : proportion of datapoints of the same class c_i in the cluster k

F1-measure

(careful: similar but not the same F-measure as the F-measure we will see for classification!)

Tradeoff between clustering correctly all datapoints of the same class in the same cluster and making sure that each cluster contains points of only one class.

For each cluster, compute: $F_1(c_i, k) = \frac{2R(c_i, k)P(c_i, k)}{R(c_i, k) + P(c_i, k)}$

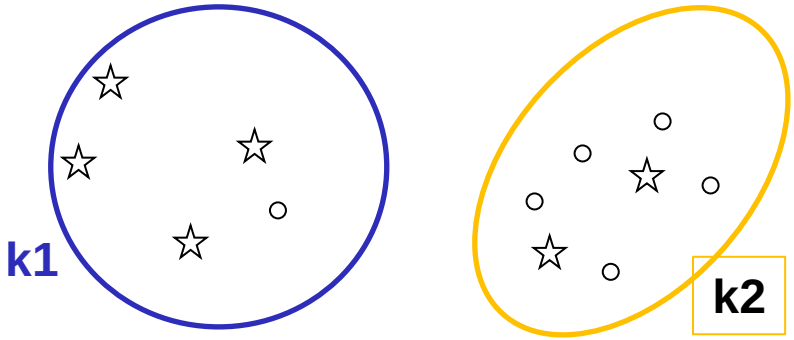
$$F_1(C, K) = \sum_{c_i \in C} \frac{|c_i|}{M} \max_k \{F_1(c_i, k)\}$$

Weights for the number of elements in each class

Picks for each class the cluster with the maximal F1 measure

Semi-Supervised Clustering

☆ Class 1
○ Class 2



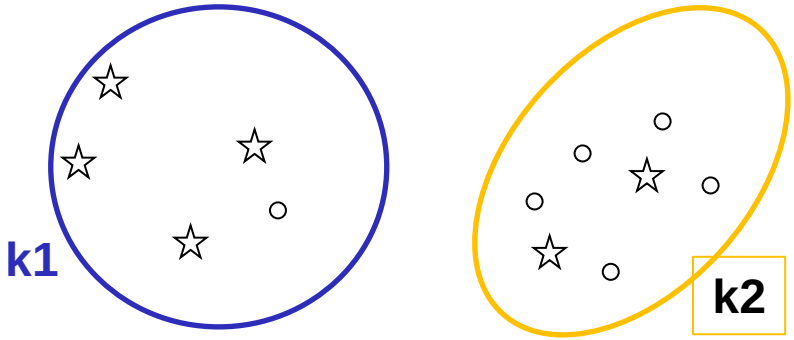
$$R(c_1, k1) = \frac{4}{6} ; R(c_1, k2) = \frac{2}{6}$$

$$P(c_1, k1) = \frac{4}{5} ; P(c_1, k2) = \frac{2}{7}$$

$$F(c_1, k1) = 0.72 > F(c_1, k2) = 0.30$$

Semi-Supervised Clustering

☆ Class 1
○ Class 2



$$R(c_2, k1) = \frac{1}{6}\ ; R(c_2, k2) = \frac{5}{6}$$

$$P(c_2, k1) = \frac{1}{5}\ ; P(c_2, k2) = \frac{5}{7}$$

$$F(c_2, k1) = 0.18 < F(c_2, k2) = 0.76$$

$$F_1(C, K) = \frac{6}{12} F(c_1, k1) + \frac{6}{12} F(c_2, k2)$$

$$F_1(C, K) = \mathbf{0.74}$$