APPLIED MACHINE LEARNING

Methods for Clustering
What is clustering?

Clustering is a type of multivariate statistical analysis also known as cluster analysis, unsupervised classification analysis, or numerical taxonomy.

Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called clusters.

Cluster: a collection of data objects that are “similar” to one another and thus can be treated collectively as one group.
Classification versus Clustering

Supervised Classification = Classification
→ We know the class labels and the number of classes.

Unsupervised Classification = Clustering
→ We do not know the class labels and may not know the number of classes.
Classification versus Clustering

Unsupervised Classification = Clustering

→ Hard problem when no pair of objects have exactly the same feature.

→ Need to determine how similar two or more objects are to one another.
Semi-Supervised Clustering

Semi-supervised clustering is a mix between classification and clustering.

→ A subset of the data is labeled
→ The rest is unlabeled
→ Infer the labels for unlabeled data by grouping them with the most similar labeled datapoints
What is Good Clustering?

Which two subgroups of pictures are similar and why?
What is Good Clustering?

Which two subgroups of pictures are similar and why?
What is Good Clustering?

• A good clustering method will produce high quality clusters in which
  • The intra-class (that is, intra-cluster) similarity is high.
  • The inter-class similarity is low.

• The quality measure of a cluster depends on the similarity measure used!

• The quality of a clustering method is also measured by its ability to
  discover some or all of the hidden patterns.
How to cluster?

Priors:
• Data cluster within a circle (isotropic Gaussian distribution)
• There are 2 clusters

Outliers (noise)

Relevant Data
How to cluster?

Priors:
- Data is generated by a complex distribution
- There are 3 clusters
Clusters’ Types

Globular Clusters

Non-Globular Clusters
How to cluster?

What choice of model (circle, ellipse) for the cluster?
How many models?
K-Means clustering generates a number $K$ of disjoint clusters that minimize:

$$J(\mu^1, \ldots, \mu^K) = \sum_{k=1}^{K} \sum_{x^i \in c_k} \left| x^i - \mu^k \right|^2$$

- $x^i$ $i^{th}$ data point
- $\mu^k$ geometric centroid
- $c_k$ cluster label or number

What choice of model (circle, ellipse) for the cluster? **Circle**

How many models? **Fixed number: $K=2$**

Where to place them for optimal clustering?
Initialization: initialize at random the positions of the centers of the clusters

In mldemos; centroids are initialized on one datapoint with no overlap across centroids.
K-means Clustering

Assignment Step:
- Calculate the distance from each data point to each centroid.
- Assign the responsibility of each data point to its “closest” centroid.

If a tie happens (i.e. two centroids are equidistant to a data point, one assigns the data point to the smallest winning centroid).

**Formula:**

$$k_i = \arg\min_k \{ d(x^i, \mu^k) \}$$

Responsibility of cluster $k$ for point $x^i$

$$r_i^k = \begin{cases} 1 & \text{if } k_i = k \\ 0 & \text{otherwise} \end{cases}$$

$x^i$: $i^{th}$ data point

$\mu^k$: geometric centroid
K-means Clustering

Update step (M-Step):
Recompute the position of centroid based on the assignment of the points

\[ k_i = \arg \min_k \{ d(x^i, \mu^k) \} \]

Responsibility of cluster \( k \) for point \( x^i \)

\[ r_i^k = \begin{cases} 1 & \text{if } k_i = k \\ 0 & \text{otherwise} \end{cases} \]

\[ \mu^k = \frac{\sum_i r_i^k x^i}{\sum_i r_i^k} \]
K-means Clustering

\[ k_i = \arg \min_k \left\{ d \left( x^i, \mu^k \right) \right\} \]

Responsibility of cluster \( k \) for point \( x^i \)

\[ r_i^k = \begin{cases} 1 & \text{if } k_i = k \\ 0 & \text{otherwise} \end{cases} \]

\[ \mu^k = \frac{\sum_i r_i^k x^i}{\sum_i r_i^k} \]

Go back to the assignment step and repeat the update step.
K-means Clustering

Stopping Criterion: stop the process when the centers are stable.
K-means Clustering

K-means creates a hard partitioning of the dataset.
K-means Clustering

The algorithm of K-means is a simple version of Expectation-Maximization applied to a model composed of isotropic Gauss functions
K-means Clustering

Computing the distance to the k-th centroid is equivalent to computing the probability that the data point has been generated by the k-th model.

\[ d(x^i, \mu^k) \sim p(x^i; \mu^k) = e^{-(x^i - \mu^k)^2} \]

The likelihood of the k-th model is:

\[ L(X; \mu^k) = \prod_i e^{-(x^i - \mu^k)^2} \]

Assignment Step (E-step):
~ Compute expectation of the equivalent Gaussian model with unity variance and centered on the centroid
• Assign the responsibility of each data point to its “closest” centroid.
~ Bayes’ rule: If \( p(x^i; \mu^2) > p(x^i; \mu^1) \), then \( x^i \) belongs to cluster 2.
K-means Clustering

The new centroid is closer to the datapoints after the update step,

\[ \mu^k = \frac{\sum_i r_i^k x_i}{\sum_i r_i^k} \]

\[ \Rightarrow e^{-(x_i - \mu^k)^2} \]

\[ \Rightarrow \text{the likelihood of the } k\text{-th model increases.} \]

\[ L(X; \mu^k) = \prod_i e^{-(x_i - \mu^k)^2} \]

**Update Step (M-step):**

\~ Maximize expectation of the equivalent Gaussian model with unity variance and centered on the centroid
**K-means Clustering**

The new centroid is closer to the datapoints after the update step,

\[
\mu_k = \frac{\sum_i r_i^k x_i}{\sum_i r_i^k}
\]

\[\Rightarrow e^{-(x_i - \mu_k)^2}\]

\[\Rightarrow \text{the likelihood of the k-th model increases.}\]

**Update Step (M-step):**

~ Maximize expectation of the equivalent Gaussian model with unity variance and centered on the centroid
K-means Clustering: Algorithm

1. **Initialization:** Pick \( K \) arbitrary centroids and set their geometric means to random values (in mldemos; centroids are initialized on one datapoint with no overlap across centroids).

2. Calculate the distance from each data point to each centroid.

3. **Assignment Step:** Assign the responsibility of each data point to its “closest” centroid (E-step). If a tie happens (i.e. two centroids are equidistant to a data point, one assigns the data point to the smallest winning centroid).

\[
k_i = \arg \min_k \{d(x^i, \mu^k)\} \quad r_i^k = \begin{cases} 1 & \text{if } k_i = k \\ 0 & \text{otherwise} \end{cases}
\]

4. **Update Step:** Adjust the centroids to be the means of all data points assigned to them (M-step)

\[
\mu^k = \frac{\sum_i r_i^k x^i}{\sum_i r_i^k}
\]

5. Go back to step 2 and repeat the process until the clusters are stable.
K-means Clustering: Properties

- There are always K clusters.
- The clusters do not overlap.
- Each member of a cluster is closer to its cluster than to any other cluster.

The algorithm is guaranteed to converge in a finite number of iterations

But it converges to a local optimum!

It is hence very sensitive to initialization of the centroids.
K-means Clustering: Weaknesses

Unbalanced clusters:
K-means takes into account only the distance between the means and data points; it has no representation of the variance of the data within each cluster.

Elongated clusters:
K-means imposes a fixed shape for each cluster (sphere).
K-means Clustering: Weaknesses

Very sensitive to the choice of the number of clusters $K$ and the initialization. → MIdemos example
K-means Clustering: Properties

Advantages:

• Computationally faster than other clustering techniques.
• Produces tighter clusters, especially if the clusters are globular.
• Guaranteed to converge.

Drawbacks:

• Does not work well with non-globular clusters.

• Sensitivity to choice of initial partitions
  Different initial partitions can result in different final clusters.

• Assumes a fixed number $K$ of clusters.

→ It is, therefore, good practice to run the algorithm several times using different $K$ values, to determine the optimal number of clusters.
Soft K-means Clustering

Assignment Step (E-step):
• Calculate the distance from each data point \( x^i \) to each centroid.
• Assign the responsibility of each data point \( x^i \) to its “closest” centroid.

Each data point \( x^i \) is given a soft `degree of assignment' to each of the means \( \mu^k \).

\[
 r^k_i : \text{responsible of cluster } k \text{ for point } x^i
\]

\[
 r^k_i = \frac{e^{(-\beta \cdot d(\mu^k, x^i))}}{\sum_{k'} e^{(-\beta \cdot d(\mu^{k'}, x^i))}} \in [0, 1], \quad \beta \in \mathbb{R}_+
\]

Normalized over clusters: \( \sum_k r^k_i = 1 \)
**Update step** (M-Step):
Recompute the position of centroid based on the assignment of the points

The model parameters, i.e. the means, are adjusted to match the weighted sample means of the data points that they are responsible for.

The update algorithm of the soft K-means is identical to that of the hard K-means, aside from the fact that the responsibilities to a particular cluster are now real numbers varying between 0 and 1.

\[ r_{ik}^k: \text{responsibility of cluster } k \text{ for point } x^i \]
\[ r_{ik}^k = \frac{e^{-\beta \cdot d(\mu_k, x^i)}}{\sum_{k'} e^{-\beta \cdot d(\mu_{k'}, x^i)}} \in [0, 1], \quad \beta \in \mathbb{R}_+ \]

Normalized over clusters: \( \sum_k r_{ik}^k = 1 \)

\[ \mu_k = \frac{\sum_i r_{ik}^k \cdot x^i}{\sum_i r_{ik}^k} \]
Soft K-means Clustering

\( \beta \) is the stiffness
\[ \sigma \equiv \frac{1}{\sqrt{\beta}} \] measures the disparity across clusters

\[ r_i^k \]: responsibility of cluster \( k \) for point \( x_i \)
\[ r_i^k = \frac{e^{-\beta \cdot d(\mu_k, x_i)}}{\sum_{k'} e^{-\beta \cdot d(\mu_{k'}, x_i)}} \in [0,1], \quad \beta \in \mathbb{R}_+ \]

Normalized over clusters: \( \sum_k r_i^k = 1 \)

small \( \sigma \)
\sim large \( \beta \)

large \( \sigma \)
\sim small \( \beta \)
Soft K-means Clustering

Soft K-means algorithm with a small (left), medium (center) and large (right) $\beta$
Soft K-means Clustering

Iterations of the Soft K-means algorithm from the random initialization (left) to convergence (right). Computed with $\beta = 10$. 
Clustering with Mixture of Gaussians

Extension of soft K-means to fit data distribution with a mixture of isotropic Gaussian pdf.

Responsibility factor for each cluster $\rightarrow$ likelihood that each cluster best represents the distribution of the data

Optimize likelihood of each point to belong to the overall distribution.

Likelihood of overall distribution (with uniform prior for each Gaussian):

$$L(X; \mu^1, \sigma^1, \mu^2, \sigma^2) = L(X; \mu^1, \sigma^1) + L(X; \mu^2, \sigma^2)$$
Clustering with Mixture of Gaussians

Assignment Step (E-step):

\[ r_i^k : \text{responsibility of cluster } k \text{ for point } x^i \]

\[ r_i^k = \frac{\alpha_k p(x^i; \mu^k, \sigma^k)}{\sum_{k'} \alpha_k p(x^i; \mu^{k'}, \sigma^{k'})}, \quad \alpha_k \in [0,1] \]

\[ p(x^i; \mu^k, \sigma^k) \in [0,1] : \text{Gauss pdf evaluated at } x^i \]

Normalized over clusters: \[ \sum_k r_i^k = 1 \]

Relative importance of each of the K clusters (measure of number of datapoints in each cluster)

\[ \rightarrow \text{in GMM, we will see that this is a measure of the relative likelihood that the Gaussian } k \text{ (or cluster } k) \text{ generated the whole dataset.} \]
Clustering with Mixture of Gaussians

Update Step (M-Step):

- \( r_i^k \): responsibility of cluster \( k \) for point \( x^i \)

\[
r_i^k = \frac{\alpha_k p(x^i; \mu^k, \sigma^k)}{\sum_k \alpha_k p(x^i; \mu^k, \sigma^k)}, \quad \alpha_k \in [0, 1]
\]

- \( p(x^i; \mu^k, \sigma^k) \in [0, 1] \): Gauss pdf evaluated at \( x^i \)

Normalized over clusters: \( \sum_k r_i^k = 1 \)

Relative importance of each of the \( k \) cluster (measure of number of datapoints in each cluster)

\( \rightarrow \) In GMM, we will see that this is a measure of the likelihood that the Gaussian \( k \) (or cluster \( k \)) generated the whole dataset.

This fits only a mixture of \textit{spherical} Gaussians!
Clustering with Mixture of Gaussians

Update Step (M-Step):

\[
r^k_i : \text{responsibility of cluster } k \text{ for point } x^i
\]

\[
r^k_i = \frac{\alpha_k p(x^i; \mu^k, \sigma^k_j)}{\sum_{k'} \alpha_k p(x^i; \mu^{k'}, \sigma^{k'}_j)}, \quad \alpha_k \in [0, 1]
\]

\[
p(x^i; \mu^k, \sigma^k) \in [0, 1]: \text{Gauss pdf evaluated at } x^i
\]

Normalized over clusters: \(\sum_k r^k_i = 1\)

Can fit ellipsoids, but aligned with the axes of the original frame of reference

\[
(\sigma^k_j)^2 = \frac{\sum_i r^k_i (x^i_j - \mu^k_j)^2}{M \cdot \sum_i r^k_i}
\]

\(j=1 \text{ to } N \) variance parameters, \(N\): data dimensionality.
Clustering with Mixture of Gaussians

Assignment Step (E-step):

Likelihood of the mixture of Gaussians: \( L(X; \{ \mu^k, \Sigma^k \}_{k=1}^K) = \sum_{k=1}^K \alpha_k \cdot p(X; \mu^k, \Sigma^k) \)

with \( p(X; \mu^k, \Sigma^k) = \prod_{i=1}^M e^{-\frac{(x_i - \mu_k)^T (\Sigma_k)^{-1} (x_i - \mu_k)}} \) (likelihood of Gaussian \( k \))

\( \mu^k, \Sigma^k \): mean and covariance matrix of Gaussian \( k \)

Mixing Coefficients are normalized \( \sum_{k=1}^K \alpha_k = 1 \)

\( \alpha_k \sim \frac{1}{M} \sum_{i=1}^M \sum_{l} p(x_i; \mu^l, \Sigma^l) \)
Clustering with Mixtures of Gaussians

Clustering with Mixtures of Gaussians using spherical Gaussians (left) and non-spherical Gaussians (i.e. with full covariance matrix) (right).

Notice how the clusters become elongated along the direction of the clusters (the grey circles represent the first and second variances of the distributions).
Gaussian Mixture Modeling

The parameters of a GMM are the means, covariance matrices and prior pdf:

\[ \Theta = \left\{ \mu^1, \ldots, \mu^K, \sum^1, \ldots, \sum^K, \alpha_1, \ldots, \alpha_K \right\} \]

Estimation of all the parameters can be done through *Expectation-Maximization* (E-M). E-M tries to find the optimum of the likelihood of the model given the data, i.e.:

\[ \max_{\Theta} L(\Theta | X) = \max_{\Theta} p(X | \Theta) \]
Expectation-Maximization

One usually can safely assume that the datapoints are i.i.d. (identically and independently distributed).

\[
\max_{\Theta} \ p(X \mid \Theta) = \max_{\Theta} \ \prod_{i=1}^{M} \sum_{k=1}^{K} \alpha_k \cdot p(x^i; \mu^k, \Sigma^k)
\]

Computing the log of the likelihood yields the same optimum:

\[
\max_{\Theta} \ \log p(X \mid \Theta) = \max_{\Theta} \ \log p(X \mid \Theta)
\]

\[
\max_{\Theta} \ \log \prod_{i=1}^{M} \sum_{k=1}^{K} \alpha_k \cdot p(x^i; \mu^k, \Sigma^k) = \max_{\Theta} \ \sum_{i=1}^{M} \log \left( \sum_{k=1}^{K} \alpha_k \cdot p(x^i; \mu^k, \Sigma^k) \right)
\]

See derivation of E-M in the annexes posted on the website
Initialization:

The priors $\alpha_1, \ldots, \alpha_k$ can be uniform for starters.
The means $\mu^1, \ldots, \mu^k$ can be initialized with K-means.

Estimation Step (E-step):

At each step $t$, estimate, for each Gaussian $k$, the probability that this Gaussian is being responsible for generating each point of the dataset by computing:

$$p(k \mid \Theta^{(t)}) = \frac{1}{M} \sum_{i=1}^{M} \frac{p_k(x^i; \mu^{k(t)}, \Sigma^{k(t)}) \cdot \alpha_k^{(t)}}{\sum_j p_k(x^i; \mu^{j(t)}, \Sigma^{j(t)}) \cdot \alpha_j^{(t)}}$$
E-M Estimate for Gaussian Mixture Models

Update Step (M-step):

Recompute the means, covariances matrices and prior probabilities so as to maximize the log – likelihood of the current estimate : \( \log\left(L\left(\Theta^{(i)} \mid X\right)\right) \) and using current estimate of the probabilities : \( p\left(k \mid \Theta^{(i)}\right) \)

\[
\mu_{k(t+1)} = \frac{\sum_j p\left(k \mid x^j, \Theta^{(i)}\right) \cdot x^j}{\sum_j p\left(k \mid x^j, \Theta^{(i)}\right)}
\]

\[
\alpha_k^{(t+1)} = \frac{1}{M} \sum_j p\left(k \mid x^j, \Theta^{(i)}\right)
\]

\[
\sum_{k(t+1)} = \frac{\sum_j p\left(k \mid x^j, \Theta^{(i)}\right) \left(x^j - \mu_{k(t+1)}\right) \left(x^j - \mu_{k(t+1)}\right)^T}{\sum_i p\left(k \mid x^j, \Theta^{(i)}\right)}
\]
Gaussian Mixture Model
Gaussian Mixture Model

GMM using 4 Gaussians with random initialization
Gaussian Mixture Model

Expectation Maximization is very sensitive to initial conditions:

GMM using 4 Gaussians with new random initialization
Gaussian Mixture Model

Very sensitive to choice of number of Gaussians. Number of Gaussians can be optimized iteratively using AIC or BIC (see later slides):

Here, GMM using 8 Gaussians
Evaluation of Clustering Methods
Evaluation of Clustering Methods

**Clustering methods rely on hyper parameters**

- Number of clusters
- 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 (e.g. intra-cluster distance versus inter-cluster distances)

E.g.: **Residual Sum of Square** is an internal measure (available in mldemos); Gives the squared distance of each vector from its centroid summed over all vectors.

\[
\text{RSS} = \sum_{k=1}^{K} \sum_{x \in C_k} |x - \mu_k|^2
\]

→ Internal measures are problematic as the metric of similarity is often already optimized by clustering algorithm.
RSS for K-Means

The RSS measures keep decreasing as one adds more clusters.

It is zero (optimal) when one has as many clusters as datapoints!

It can still be used to determine when to stop adding clusters by looking at the slope of the decrease of the measure.

\[
\text{RSS} = \sum_{k=1}^{K} \sum_{x \in C_k} (x - \mu^k)^2
\]
K-means Clustering: Examples

Cluster Analysis of Hedge Funds (fonds speculatifs)
[N. Das, 9th Int. Conf. on Computing Economis and Finance, 2011]

No legal definition of Hedge funds - consists of a wide category of investment funds with high risk & high returns – variety of strategies for guiding the investment

Research Question: classify type of Hedge funds based on information provided to the client

Data Dimension (Features): such as: asset class, size of the hedge fund, incentive fee, risk-level, and liquidity of hedge funds
K-means Clustering: Examples

Cluster Analysis of Hedge Funds (fonds speculatifs)  
[N. Das, 9th Int. Conf. on Computing Econimis and Finance, 2011]

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

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

Optimal results are found with 7 clusters.

The clusters do not correspond to those found by one classical Hedge fund database (ZCM)

Criteria for ZCM classification encompass subjective assessment perhaps.
Other internal measures

K-Means, soft-K-Means and GMM have several hyperparameters:
(Fixed number of clusters, beta, number of Gaussian functions)

→ Measure to determine how well the choice of hyperparameters fit the dataset (maximum-likelihood measure)

\[ X : \text{dataset}; \ M : \text{number of datapoints}; \ B : \text{number of free parameters} \]

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

\( L \): maximum likelihood of the model given \( B \) parameters

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

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

K-Means, soft-K-Means and GMM have several hyperparameters:
(Fixed number of clusters, beta, number of Gaussian functions)

- Measure to determine how well the choice of hyperparameters fit the dataset (maximum-likelihood measure)

\[ X : \text{dataset}; \ M : \text{number of datapoints}; \ B : \text{number of free parameters} \]

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

\( L \): maximum likelihood of the model given \( B \) parameters

Choosing AIC versus BIC depends on the application:

\( \rightarrow \) 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.

Penalty for increase in computational costs

Compare methods in practical II next week
External measures of clustering

External measures assume that a subset of datapoints have class label \( \rightarrow \text{semi-supervised learning} \)

They measure how well these datapoints are clustered.

- Needs to have an idea of the number of existing classes and have labeled some datapoints
- Interesting only in cases when labeling is highly time-consuming when the data is very large (e.g. in speech recognition)
Semi-Supervised Learning

Clustering 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.

\[ M : \text{nm of labeled datapoints} \]
\[ C = \{c_i\} : \text{the set of classes} \]
\[ K : \text{nm of clusters}, \]
\[ n_{ik} : \text{nm of members of class } c_i \text{ and of cluster } k \]

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

\[ F_1(c_i, k) = \frac{2R(c_i, k)P(c_i, k)}{R(c_i, k) + P(c_i, k)} \]

\[ R(c_i, k) = \frac{n_{ik}}{|c_i|} \]
\[ P(c_i, k) = \frac{n_{ik}}{|K|} \]

Penalize fraction of labeled points in each class

Picks for each class the cluster with the maximal F1 measure

Recall: proportion of datapoints correctly classified/clusterized

Precision: proportion of datapoints of the same class in the cluster
Semi-Supervised Learning

$M$ : 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 $k$

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

$F_1(c_i, k) = \frac{2R(c_i, k)P(c_i, k)}{R(c_i, k) + P(c_i, k)}$

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

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

**Recall**: proportion of datapoints correctly classified/clusterized

**Precision**: proportion of datapoints of the same class in the cluster

\[
R(c_1, k = 1) = \frac{2}{2} = P(c_2, k = 2) = \frac{2}{2}
\]

\[
P(c_1, k = 1) = \frac{2}{6} < P(c_2, k = 2) = \frac{4}{6}
\]
Semi-Supervised Learning

\[ F(C, K) = \frac{2}{6} F(c_1, k = 1) + \frac{4}{6} F(c_2, k = 2) = 0.7 \]

Penalize fraction of labeled points in each class

Picks for each class the cluster with the maximal F1 measure

- \( M \): 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 \( k \)

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

\[
F_1(c_i, k) = \frac{2R(c_i, k)P(c_i, k)}{R(c_i, k) + P(c_i, k)}
\]

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

\[
P(c_i, k) = \frac{n_{ik}}{|k|}
\]
Semi-Supervised Learning

Semi-Supervised Clustering
(Here 50% labeled data-points yields perfect clustering)

Both BIC and F1 measure give the correct number of clusters.
BIC and F1 are used only to assess the result of the clustering not to run clustering.
Semi-Supervised Learning

Semi-Supervised Clustering
Label a fraction of datapoints and run clustering
(here 100% labelled datapoints yields perfect clustering)
Semi-Supervised Learning

When initialized differently, one obtains a different suboptimal clustering.

Semi-Supervised Clustering
Label a fraction of datapoints and run clustering
(here 100% labelled datapoints yields perfect clustering)
Summary

When to use what? E.g. k-means versus soft K-means versus GMM

When using any machine learning algorithm, you have to balance a number of factors:

- Computing time at training and at testing
- Number of open parameters
- Curse of dimensionality (order of growth with number of datapoints, dimension, etc)
- Robustness to initialization, optimality of solution