BINUS UNIVERSITY

Course : COMP6577 – Machine Learning

Effective Period : February 2020

Clustering: DBSCAN & Gaussian Mixture Model

Session 23 & 24

People
Innovation
Excellence

Learning Outcome

- LO3: Student be able to experiment classification and clustering algorithm from given dataset

Outline

- DBSCAN
- Gaussian Mixture Models
- Gaussian Mixture Modeling and Clustering
- Case study

People
Innovation
Excellence

Background

- Clustering algorithms are attractive for the task of class identification. However, the application to large spatial databases rises the following requirements for clustering algorithms:
  - Minimal requirements of domain knowledge to determine the input parameters, because appropriate values are often not known in advance when dealing with large databases
  - Discovery of clusters with arbitrary shape, because the shape of clusters in spatial databases may be spherical, drawn-out, linear, elongated etc.
  - Good efficiency on large databases, i.e. on databases of significantly more than just a few thousand objects.
- The well-known clustering algorithms offer no solution to the combination of these requirements.

BINUS UNIVERSITY

People
Innovation
Excellence

DBSCAN

- One of clustering algorithm DBSCAN requires only one input parameter and supports the user in determining an appropriate value for it. It discovers clusters of arbitrary shape.
- DBSCAN is efficient even for large spatial databases.
- DBSCAN (Density Based Spatial Clustering of Applications with Noise) is designed to discover the clusters and the noise in a spatial database according to definition:
  - (cluster) Let *D* be a database of points. cluster *C* wrt. Eps and MinPts is a non-empty subset of D satisfying the following conditions:
    1) ∀ p, q: if p ∈ C and q is density-reachable from p wrt. Eps and MinPts, then q ∈ C. (Maximality)
    2) V p, q ∈ C: p is density-connected to q wrt. EPS and MinPts. (Connectivity)

DBSCAN

  - (noise) Let C_t ….. C_k be the clusters of the database D wrt. parameters Eps_i and $MinPts_i$, i = 1 ….. k. Then we define the noise as the set of points in the database D not belonging to any cluster Ci , i.e. noise = $\{p \in D \mid \forall i: p \notin C_i)$.
- DBSCAN uses global values for Eps and MinPts, i.e. the same values for all clusters. The density parameters of the "thinnest" cluster are good candidates for these global parameter values specifying the lowest density which is not considered to be noise.

People
Innovation
Excellence

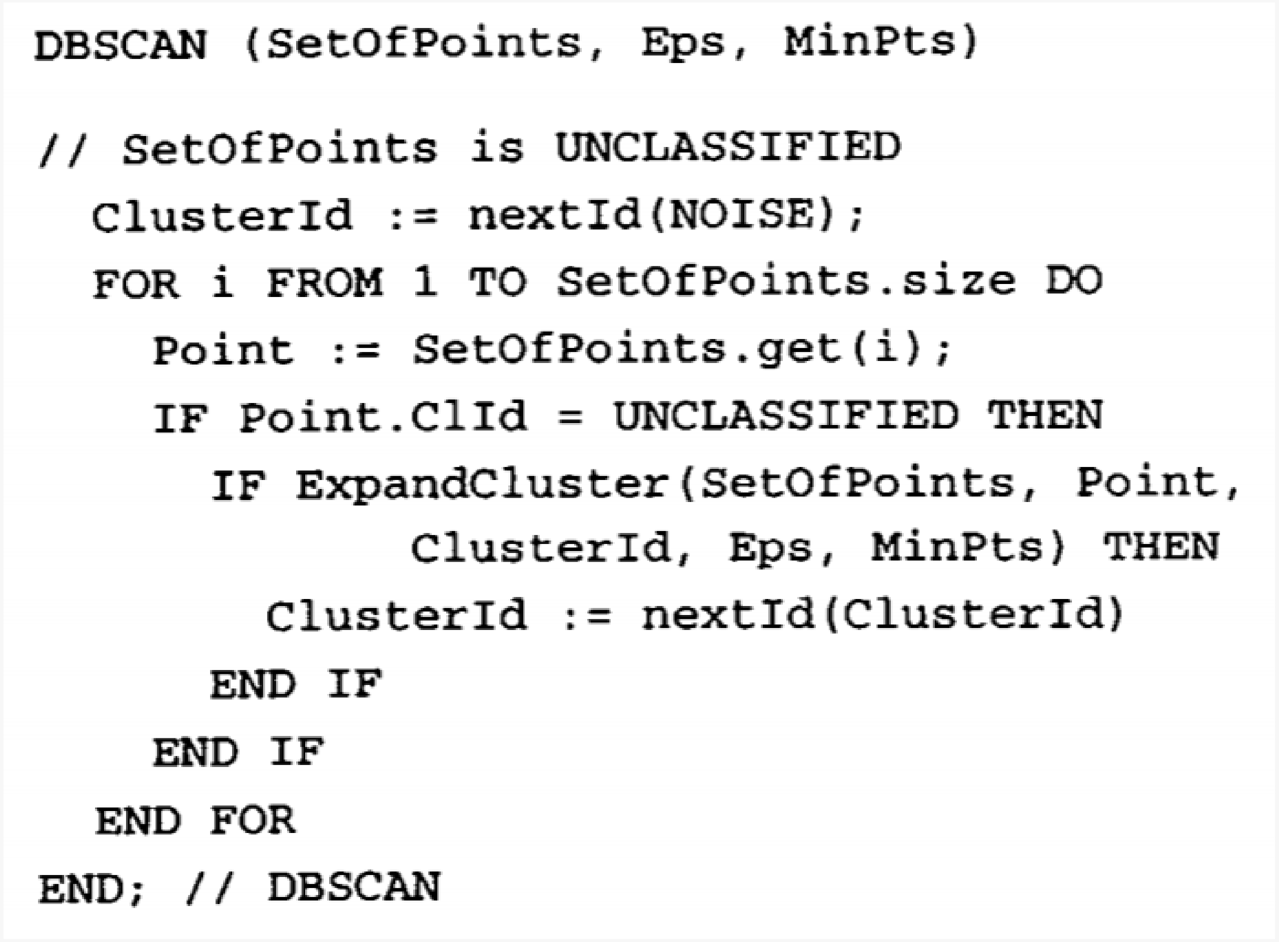
Parameters

- ***eps***: *specifies how close points should be to each other to be considered a part of a cluster. It means that if the distance between two points is lower or equal to this value (eps), these points are considered neighbors.*

- ***minPoints***: *the minimum number of points to form a dense region. For example, if we set the minPoints parameter as 5, then we need at least 5 points to form a dense region.*

People
Innovation
Excellence

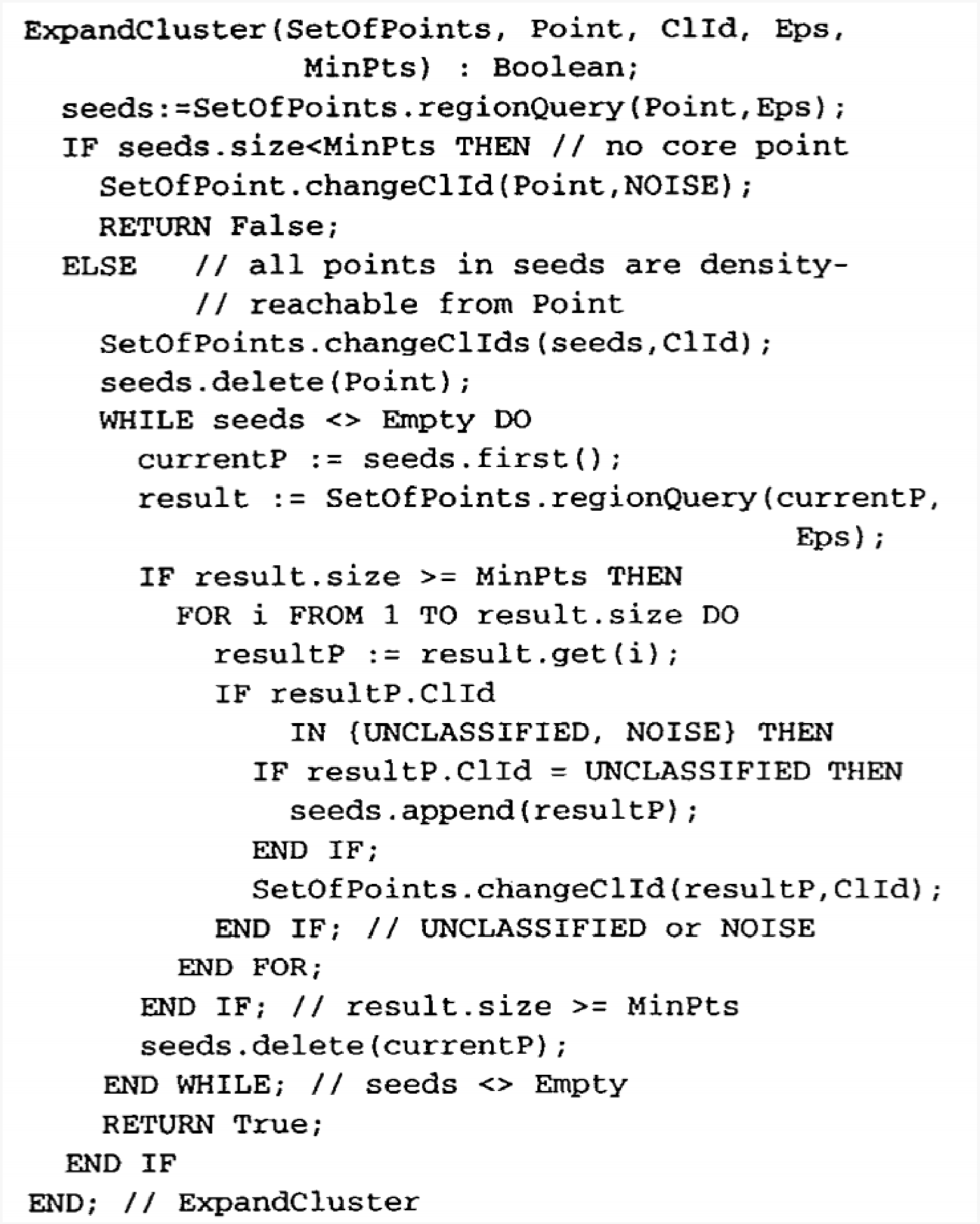
DBSCAN Algorithm



```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point,
                ClusterId, Eps, MinPts) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // DBSCAN
```

DBSCAN Algorithm



```
ExpandCluster(SetOfPoints, Point, ClId, Eps,
                  MinPts) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE   // all points in seeds are density-
         // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
                                                Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
              IN {UNCLASSIFIED, NOISE} THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
            SetOfPoints.changeClId(resultP,ClId);
          END IF; // UNCLASSIFIED or NOISE
        END FOR;
      END IF; // result.size >= MinPts
      seeds.delete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

People
Innovation
Excellence

Gaussian Mixture Models

- Mixture modeling provides the freedom to model the unknown pdf, p(x), as a linear combination of different distributions, that is,

$$p(\mathbf{x}) = \sum_{k=1}^{K} P_k p(\mathbf{x}|k)$$

where P_k is the parameter weighting the specific contributing pdf, p(x|k).

- To guarantee that p(x) is a pdf, the weighting parameters must be nonnegative and add t($\sum_{k=1}^{K} P_k = 1$)

- Mixture modeling is a typical task involving hidden variables; that is, the labels, k, of the pdf from which an obtained observation has originated. In practice, each p(x|k) is chosen from a known pdf family, parameterized via a set of pa

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k p(\boldsymbol{x}|k; \boldsymbol{\xi}_k).$$

- and the task is to estimate (Pk, ξ k), k = 1, 2, ... , K, based on a set of observations x_n, n = 1, 2, ... , N. The set of observations, X = {x_n, n = 1, ... , N}, forms the incomplete set while the complete set {X , K} comprises the samples (x_n, k_n), n = 1, ... , N, with k_n being the label of the distribution (pdf) from which x_n was drawn. Parameter estimation for such a problem naturally lends itself to be treated via the EM algorithm.

People
Innovation
Excellence

Gaussian mixtures

- Let $p(\boldsymbol{x}|k;\boldsymbol{\xi}_k) = p(\boldsymbol{x}|k;\boldsymbol{\mu}_k,\Sigma_k) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_k,\Sigma_k\right)$

  where for simplicity we will assume that $\Sigma_k = \sigma_k^2 I$, k = 1, ... , *K*. We will further assume the observations to be i.i.d. For such a modeling, the following hold true:

✓ The log-likelihood of the complete data set is given by

$$\ln p\left(\mathcal{X},\mathcal{K};\Xi,\boldsymbol{P}\right) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n,k_n;\boldsymbol{\xi}_{k_n}) = \sum_{n=1}^{N} \ln\left(p(\boldsymbol{x}_n|k_n;\boldsymbol{\xi}_{k_n})P_{k_n}\right)$$

We have used the notation,

$$\Xi = [\boldsymbol{\xi}_1^T,\ldots,\boldsymbol{\xi}_K^T]^T,\quad \boldsymbol{P} = [P_1,P_2,\ldots,P_K]^T,\quad \text{and}\quad \boldsymbol{\xi}_k = [\boldsymbol{\mu}_k^T,\sigma_k^2]^T$$

- ✓ The posterior probabilities of the hidden discrete variables are given b

$$P(k|\boldsymbol{x};\,\Xi,\boldsymbol{P}) = \frac{p(\boldsymbol{x}|k;\boldsymbol{\xi}_k)P_k}{p(\boldsymbol{x};\,\Xi,\boldsymbol{P})}$$

where

$$p(\boldsymbol{x};\,\Xi,\boldsymbol{P}) = \sum_{k=1}^{K} P_k p(\boldsymbol{x}|k;\boldsymbol{\xi}_k)$$

We have now all the ingredients required by the EM algorithm. St $\Xi^{(0)}$ g fron $\boldsymbol{P}^{(0)}$ and , the (j + 1) iteration comprises the following steps:

- E-step: compute

$$\boxed{P(k|\boldsymbol{x}_n; \Xi^{(j)}, \boldsymbol{P}^{(j)}) = \frac{p(\boldsymbol{x}_n|k; \boldsymbol{\xi}_k^{(j)})P_k^{(j)}}{\sum_{k=1}^{K} P_k^{(j)} p(\boldsymbol{x}_n|k; \boldsymbol{\xi}_k^{(j)})}, \ n = 1, 2, \ldots, N,}$$

- which in turn defines

$$\begin{aligned}\mathcal{Q}(\Xi, \boldsymbol{P}; \Xi^{(j)}, \boldsymbol{P}^{(j)}) &= \sum_{n=1}^{N} \mathbb{E}\left[\ln\left(p(\boldsymbol{x}_n|\mathrm{k}_n; \boldsymbol{\xi}_{\mathrm{k}_n})P_{\mathrm{k}_n}\right)\right] \\ &:= \sum_{n=1}^{N}\sum_{k=1}^{K} P(k|\boldsymbol{x}_n; \Xi^{(j)}, \boldsymbol{P}^{(j)})\Big(\ln P_k - \frac{l}{2}\ln\sigma_k^2 \\ &\quad - \frac{1}{2\sigma_k^2}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2\Big) + C,\end{aligned}$$

- where C includes all the terms corresponding to the normalization constant. Note that we have finally relaxed the notation from k_n to k, because we sum up over all k, which does not depend on n.

- M-step: Maximization of $Q(\boldsymbol{\Xi}, \boldsymbol{P}; \boldsymbol{\Xi}^{(j)}, \boldsymbol{P}^{(j)})$ with respect to all the involved parameters results in the following set of recursions:

Set for notational convenience $\gamma_{kn} := P(k|\boldsymbol{x}_n; \boldsymbol{\Xi}^{(j)}, \boldsymbol{P}^{(j)})$

Then

$$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn} \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_{kn}},$$

$$\sigma_k^{2(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)}\|^2}{l \sum_{n=1}^{N} \gamma_{kn}},$$

$$P_k^{(j+1)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{kn}.$$

Iterations continue until a convergence criterion is met

Gaussian Mixture Modeling and Clustering

- A clustering is a specific allocation of the points to clusters. In general, assigning points to clusters according to an optimality criterion is an NP-hard task. Thus, in general, any clustering algorithm provides a suboptimal solution.
- Gaussian mixture modeling is among the popular clustering algorithms. The main assumption is that the points, which belong to the same cluster, are distributed according to the same Gaussian distribution (this is how similarity is defined in this case), of unknown mean and covariance matrix.
- Each mixture component defines a different cluster. Thus, the goal is to run the EM algorithm over the available data points to provide, after convergence, the posterior probabilities $P(k|x_n)$, $k = 1, 2, \dots, K$, $n = 1, 2, \dots, N$, where each k cor
cluster k a

$$\text{assign } \boldsymbol{x}_n \text{ to cluster } k = \arg\min_i P(i|\boldsymbol{x}_n), \;\; i = 1, 2, \dots, K.$$

Gaussian Mixture Modeling and Clustering

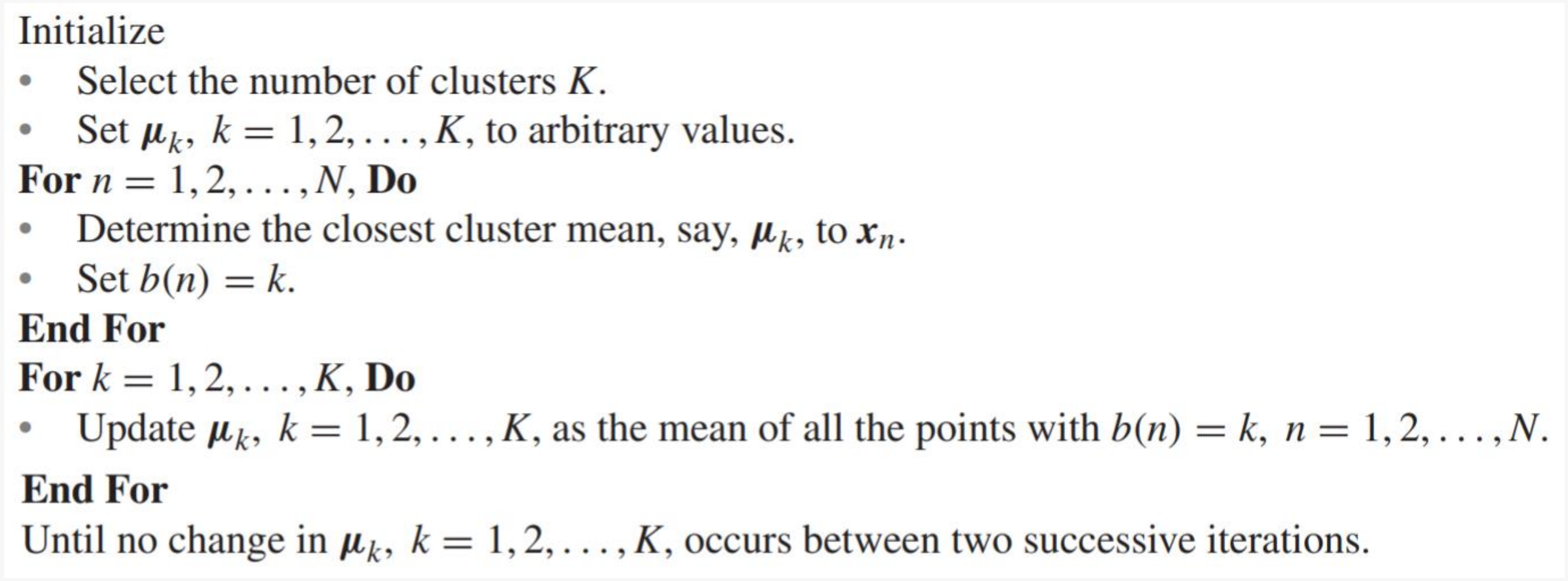
- The EM algorithm for clustering can be considered to be a refined version of a more primitive scheme, known as the **k-means** or **isodata algorithm**.
- In the EM algorithm, the posterior probability of each point, x_n, with respect to each one of the clusters, *k*, is computed recursively. Moreover, the mean value μ_k, of the points associated with cluster k, is computed as a weighted average of all the training points.
- In contrast, in the k-means algorithm, at each iteration the posterior probability, gets a binary value in {1, 0}; for each point, xn, the Euclidean distance from all the currently available estimates of the mean values is computed, and the posterior probability is estimated according to the followi

$$P(k|\boldsymbol{x}_n) = \begin{cases} 1, & \text{if } ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 < ||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2,\ j \neq k, \\ 0, & \text{otherwise.} \end{cases}$$

Gaussian Mixture Modeling and Clustering

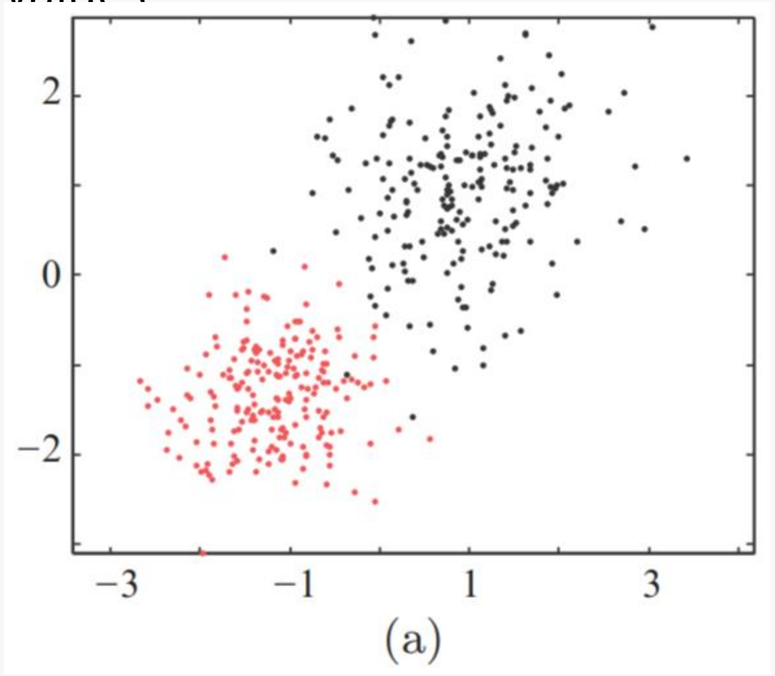
- The k-means algorithm is not concerned about covariance matrices. Despite its simplicity, it is not an exaggeration to say that it is the most well-known clustering algorithm, and a number of theoretical papers and improved versions have been proposed over the years.

The k-means or isodata clustering algorithm



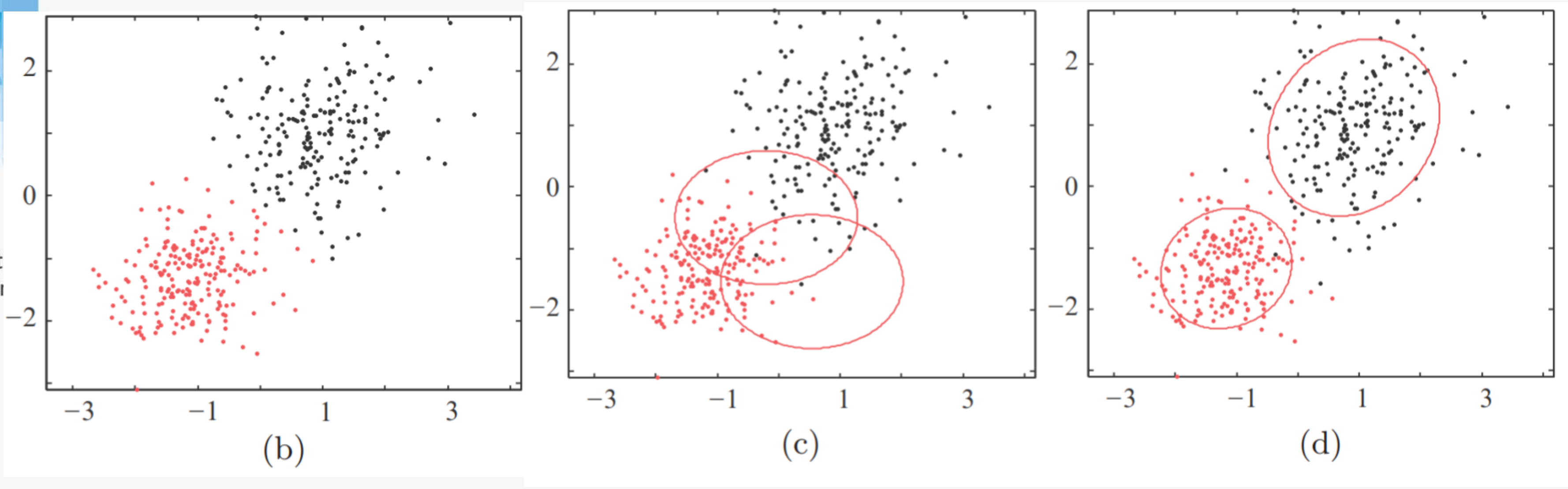
```
Initialize
•   Select the number of clusters $K$.
•   Set $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, to arbitrary values.
For $n = 1, 2, \ldots, N$, Do
•   Determine the closest cluster mean, say, $\boldsymbol{\mu}_k$, to $\boldsymbol{x}_n$.
•   Set $b(n) = k$.
End For
For $k = 1, 2, \ldots, K$, Do
•   Update $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, as the mean of all the points with $b(n) = k,\ n = 1, 2, \ldots, N$.
End For
Until no change in $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, occurs between two successive iterations.
```

- The following figure shows the data points generated by two Gaussians; 200 points from each one. The points are shown by red and gray colors, depending on the Gaussian that generated them. For both, the EM and the k-means algorithms, the correct number of clusters (K = 2) was given. The k-means was initialized with zero mean values.



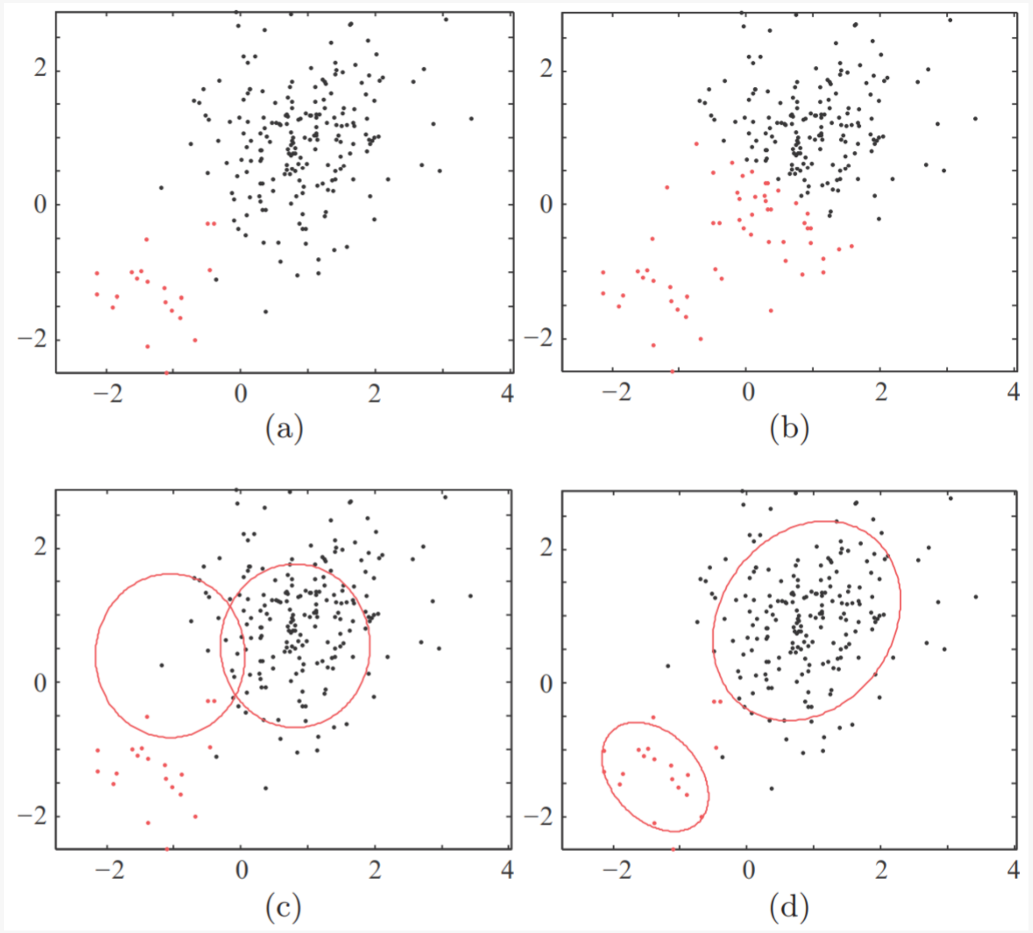
(a)

- Figure (b) shows the clusters formed by the k-means and in figure (d) the clusters formed by the EM algorithm. Figure (c) shows the Gaussians that were used for the initialization of the EM.



BINUS UNIVERSITY

- The figure shows the respective sequence of figures, which corresponds to points obtained by the same Gaussians; however, now, there is an imbalance to the number of the points, where only 20 points spring from the first one and 200 points from the second.



People
Innovation
Excellence

- Observe that the k-means has a problem in recovering the true clustering structure; it attempts to make the two clusters more equally sized. A number of techniques and versions of the basic k-means scheme have been proposed to overcome its drawbacks.
- Finally, it must be stressed that both, the EM and the k-means algorithms, will always recover as many clusters as the user-defined input variable, K, dictates.
- In the case of the EM algorithm, this drawback is overcome when the variational EM algorithm is used.

Case Study

Given data of Singapore Airbnb which can be downloaded in this link

https://www.kaggle.com/jojoker/singapore-airbnb

- Do clustering for the Singapore Airbnb data above using the method that you have learnt.

End of Session 23 & 24

References

- Sergios Theodoridis. (2015). *Machine Learning: a Bayesian and Optimization Perspective*. Jonathan Simpson. ISBN: 978-0-12-801522-3. Chapter 12.
- http://www.cs.csi.cuny.edu/~gu/teaching/courses/csc76010/slides/Clustering%20Algorithm%20by%20Vishal.pdf
- https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf
- https://www.kaggle.com/jojoker/singapore-airbnb