Clusters probabilísticos

Dr. Raimundo Sánchez

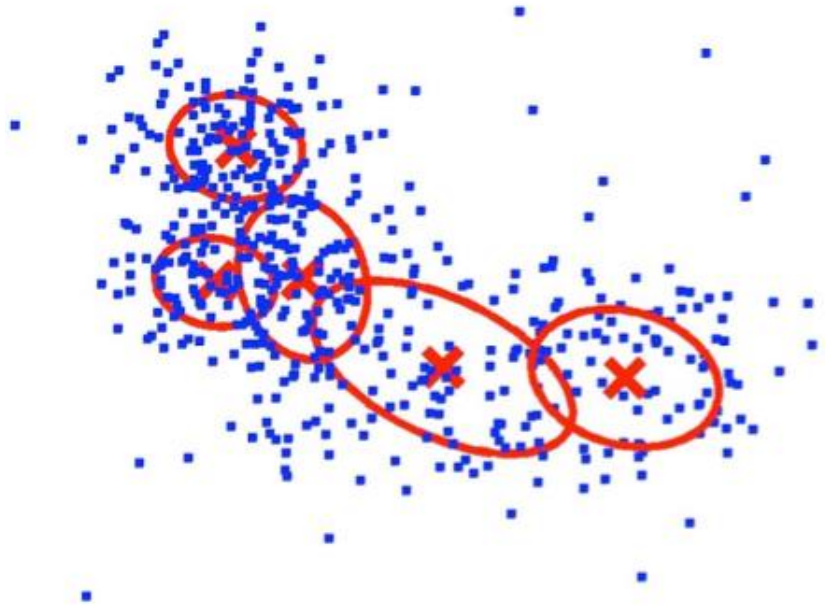
raimundo.sanchez@uai.cl

@raimun2

Métodos probabilísticos

Los métodos probabilísticos proporcionan una descripción distribucional completa para cada componente, generando clústeres flexibles.

Es decir, dado un modelo, cada punto tiene un vector de K probabilidades de pertenencia



$$f(x) = \sum_{k=1}^{K} w_k f_k(x;\theta)$$

probability of observing x

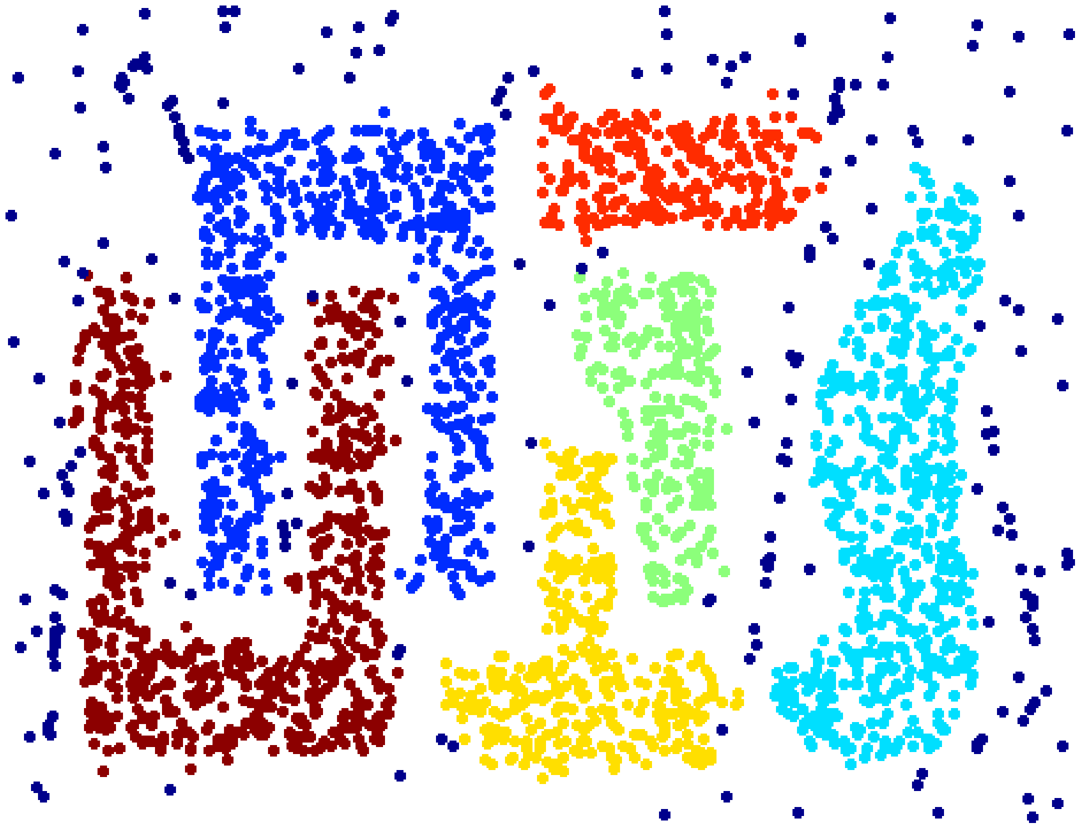
likelihood of point belonging to cluster k

likelihood of x being generated from cluster k

Métodos basados en densidad:
DBSCAN

DBSCAN

DBSCAN es un algoritmo de clusters de densidad, donde dado un conjunto de puntos en algún espacio, agrupa los puntos que están empaquetados de cerca.



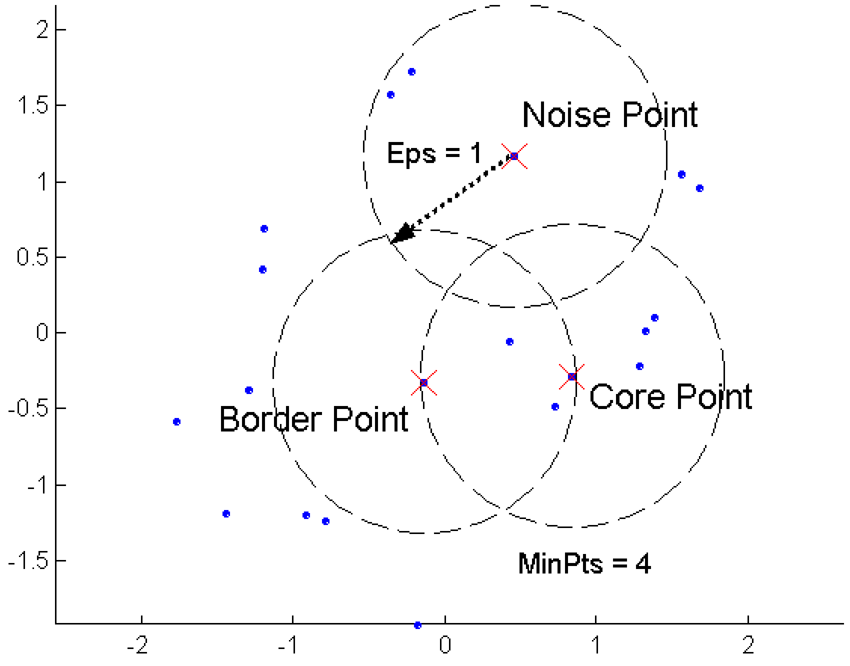
Definiciones

Densidad es el número de puntos dentro de un determinado **radio (Eps)**

Un **punto central** es un punto que tiene el mismo **número de puntos (MinPts)** dentro de una o mas esferas definida por Eps (incluido él mismo).

Un **punto fronterizo** tiene menos puntos que MinPts dentro de Eps, pero está cerca de un punto central

Un punto de ruido es cualquier punto que no sea un punto central o un punto fronterizo.

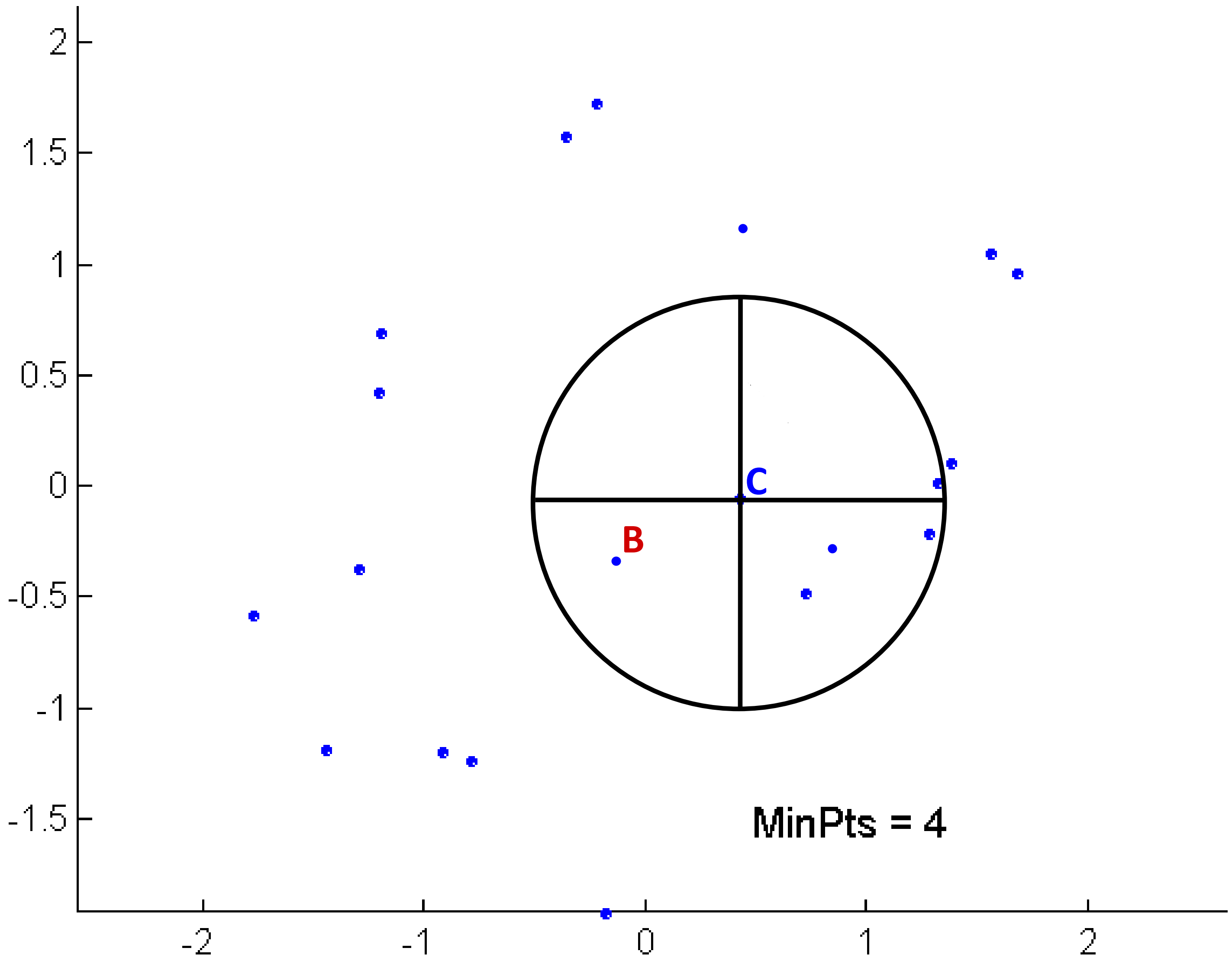


Algoritmo

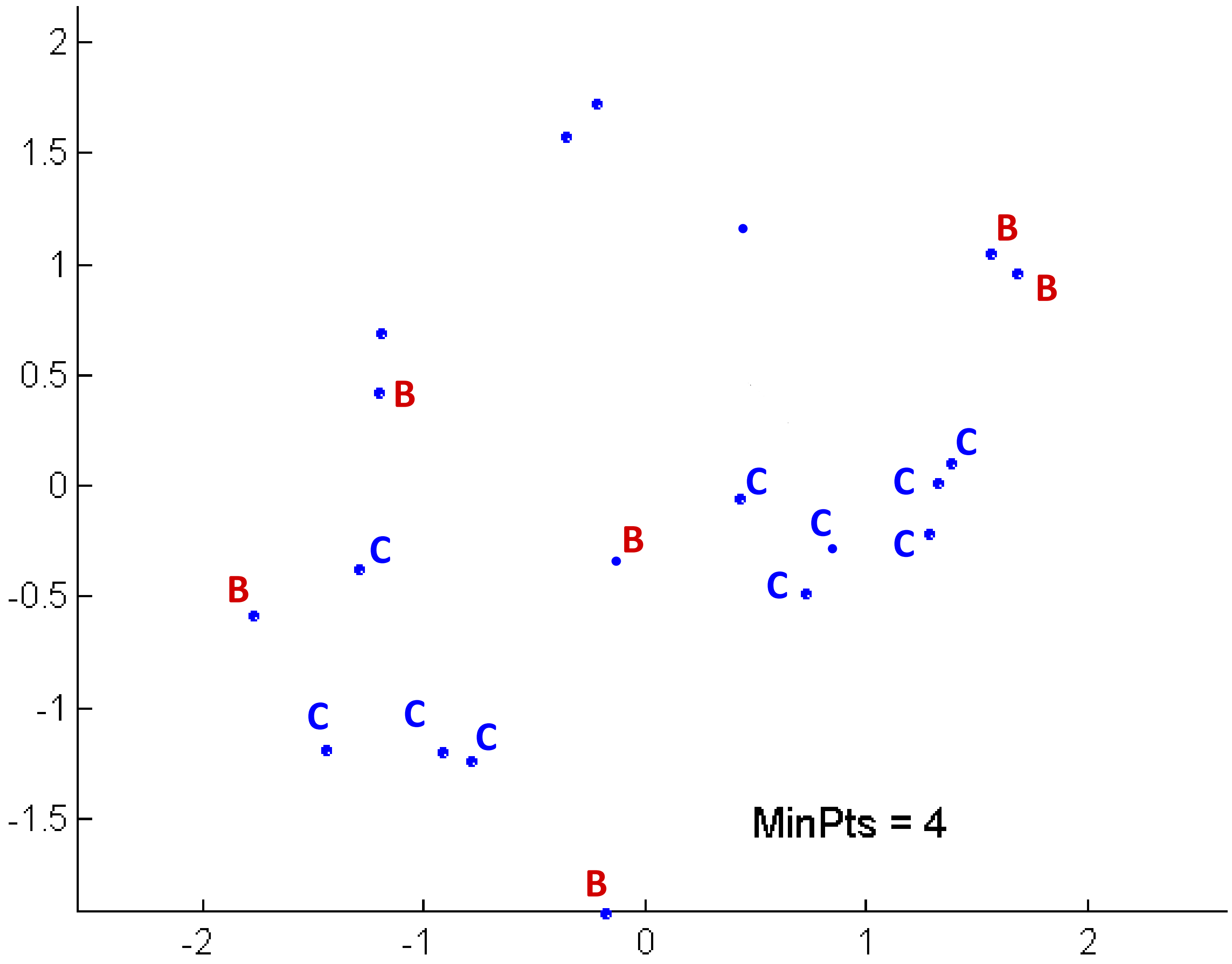
- Definir **Eps** y **MinPts**
- Determinar puntos **centro**, **frontera**, y **ruido**
- Eliminar puntos de ruido
- Aplicar el siguiente algoritmo de clusters

$current\_cluster\_label \leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
    Label the current core point with cluster label $current\_cluster\_label$
  end if
  for all points in the Eps-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label $current\_cluster\_label$
    end if
  end for
end for

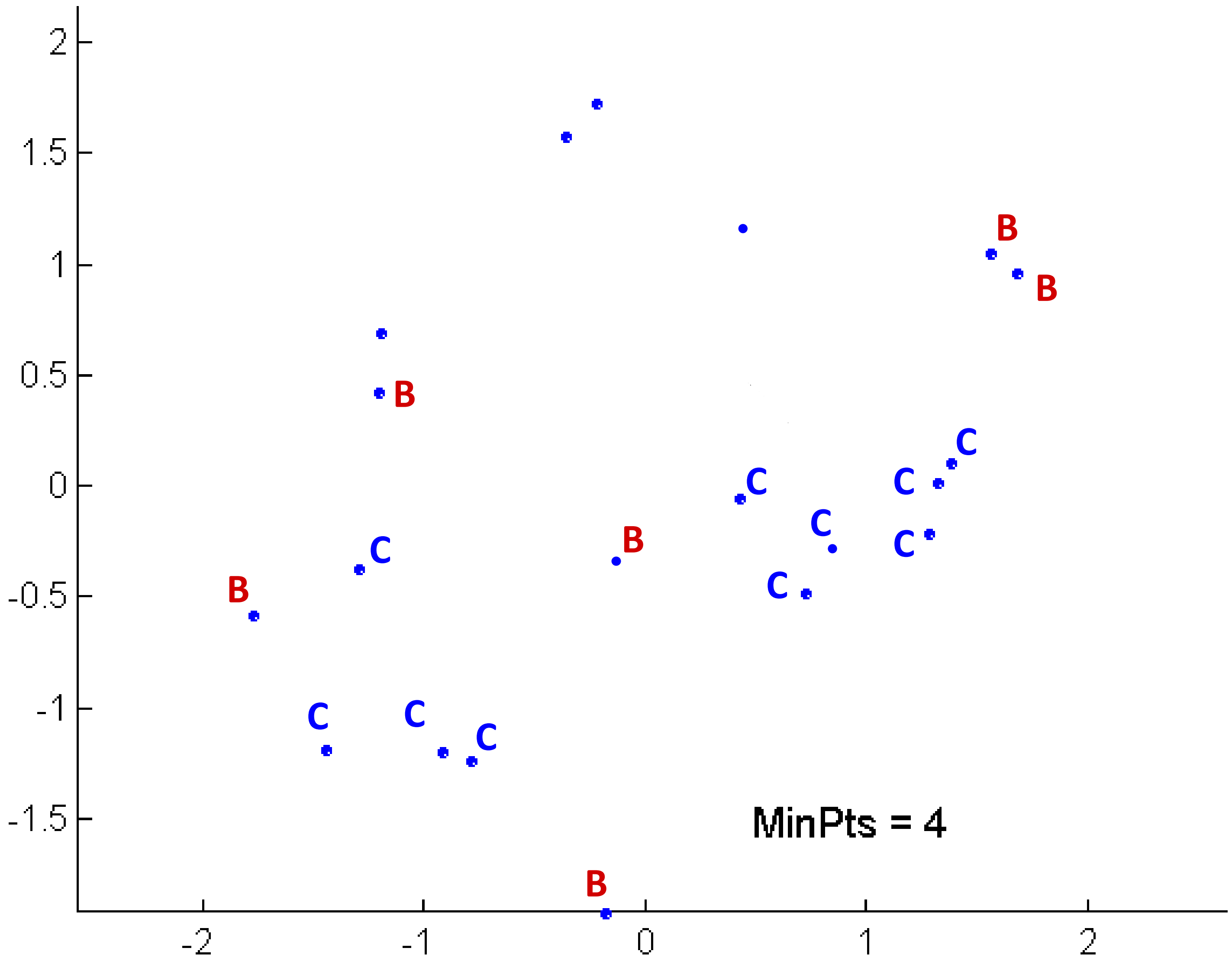
Ejemplo



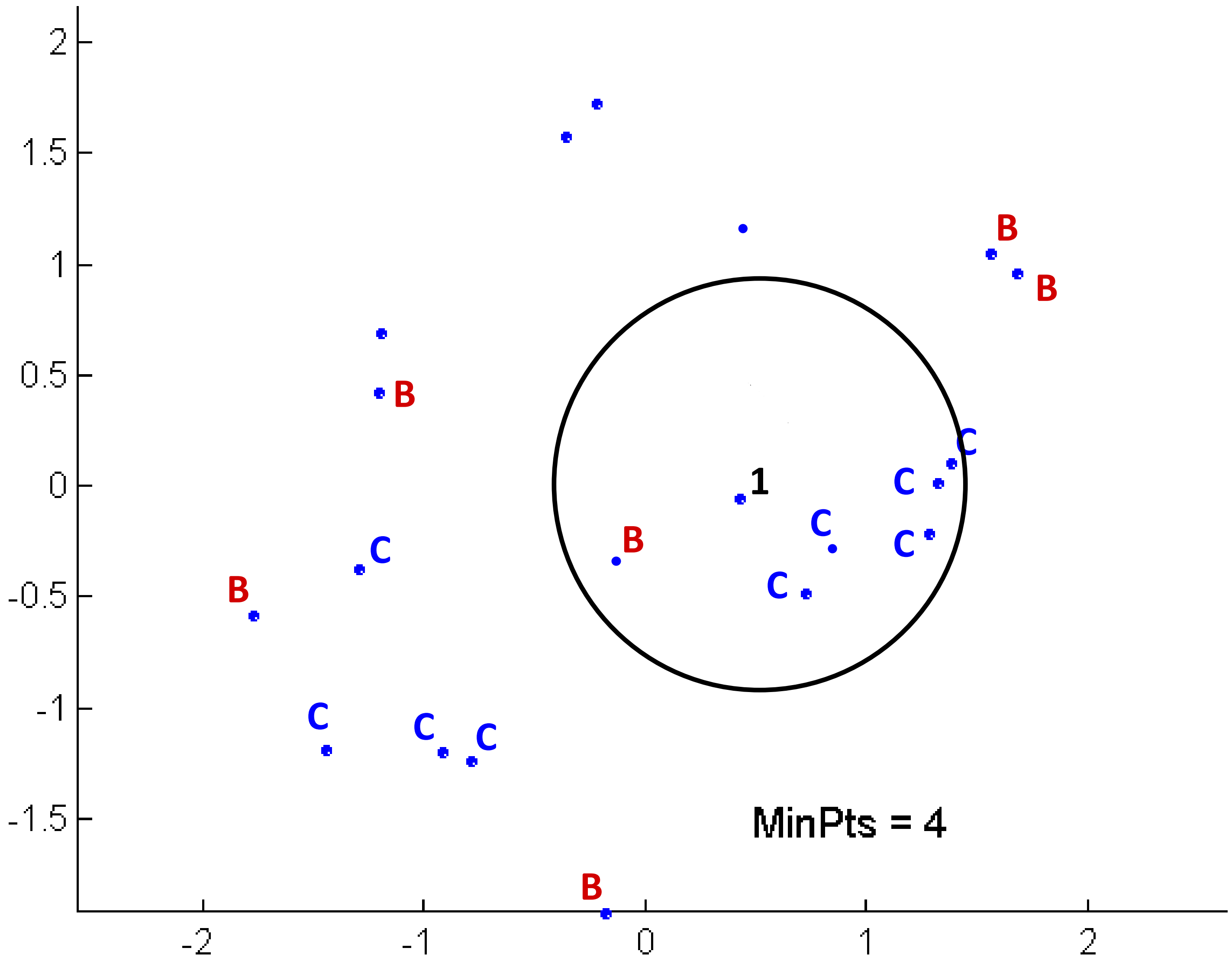
Ejemplo



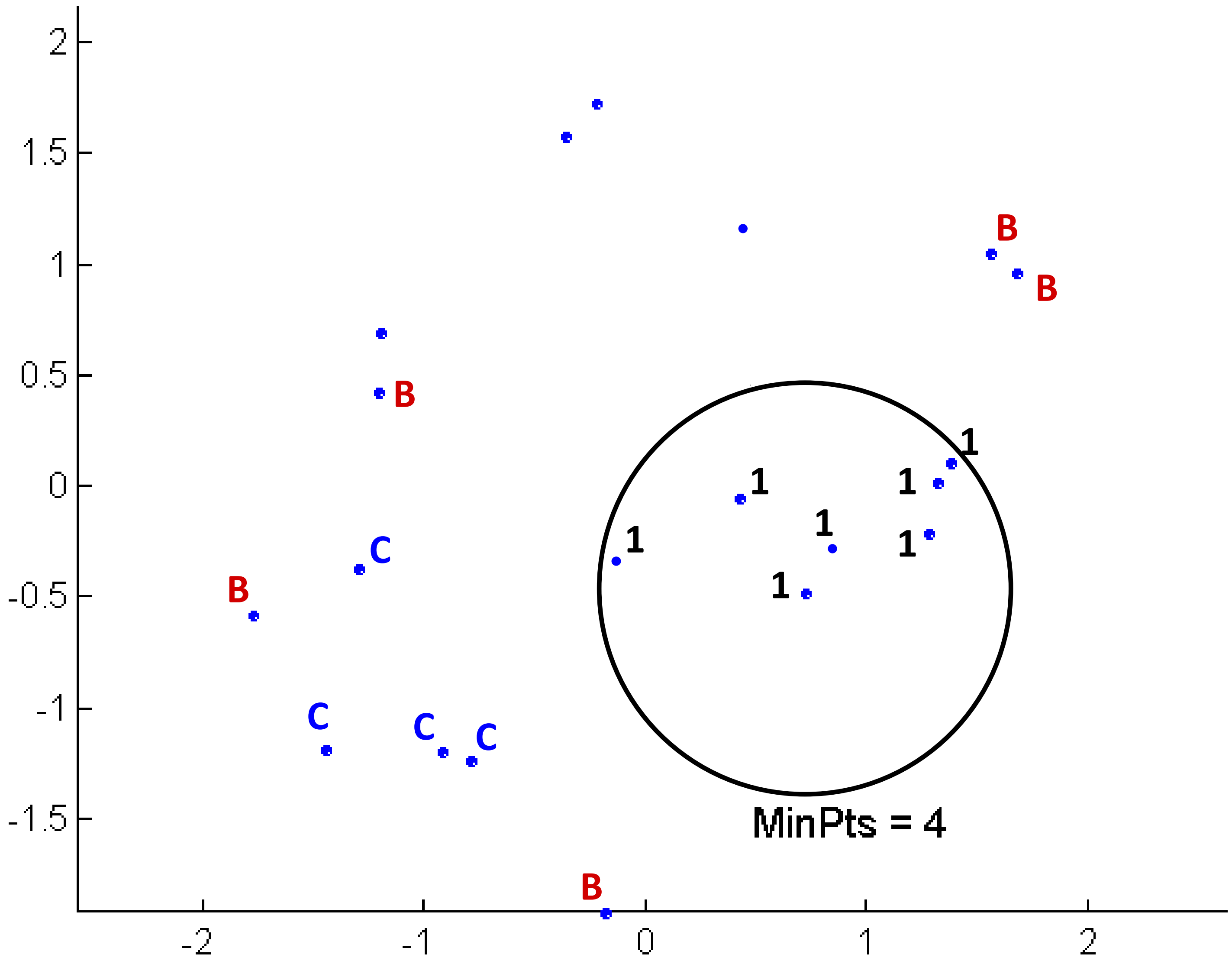
Ejemplo



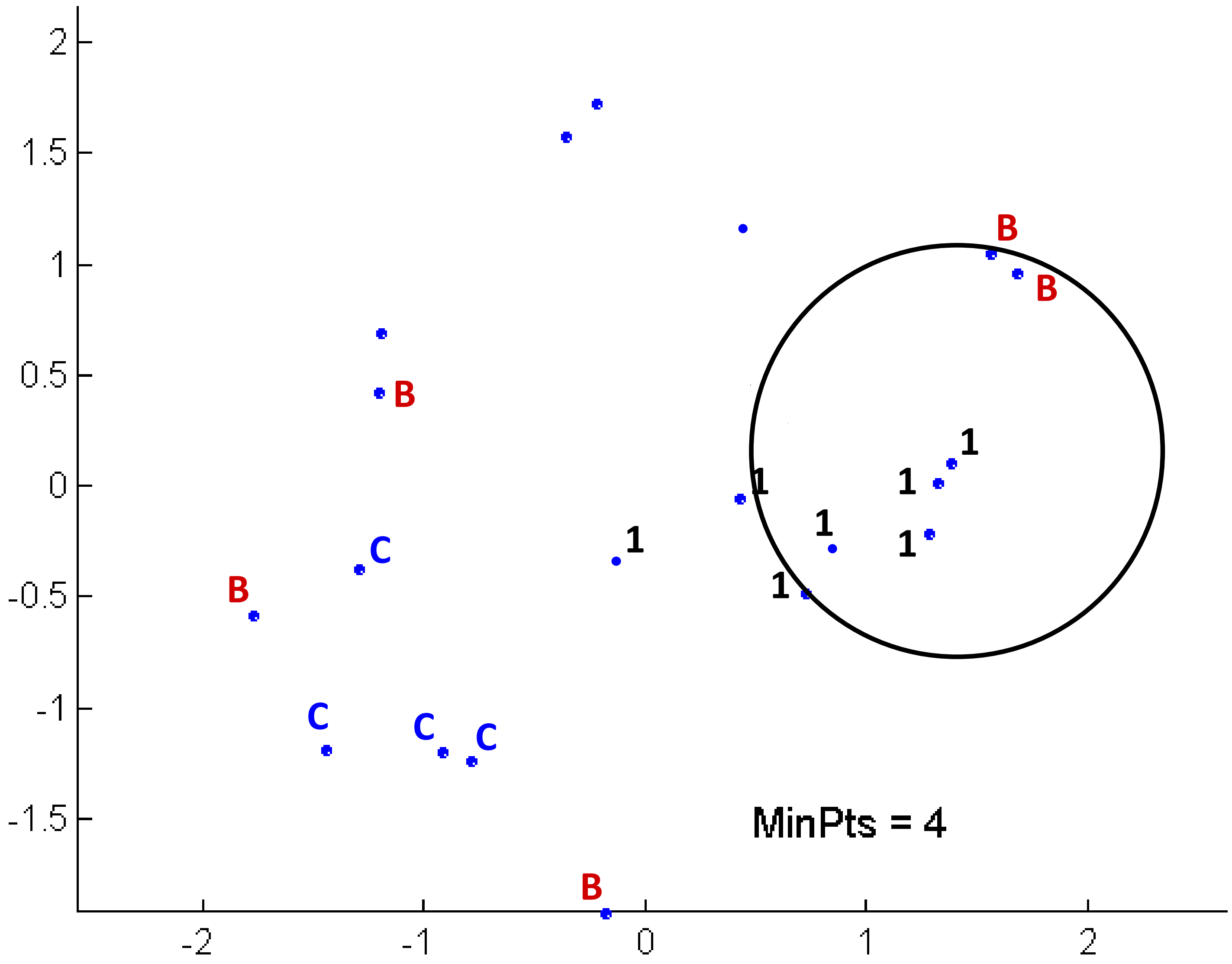
Ejemplo



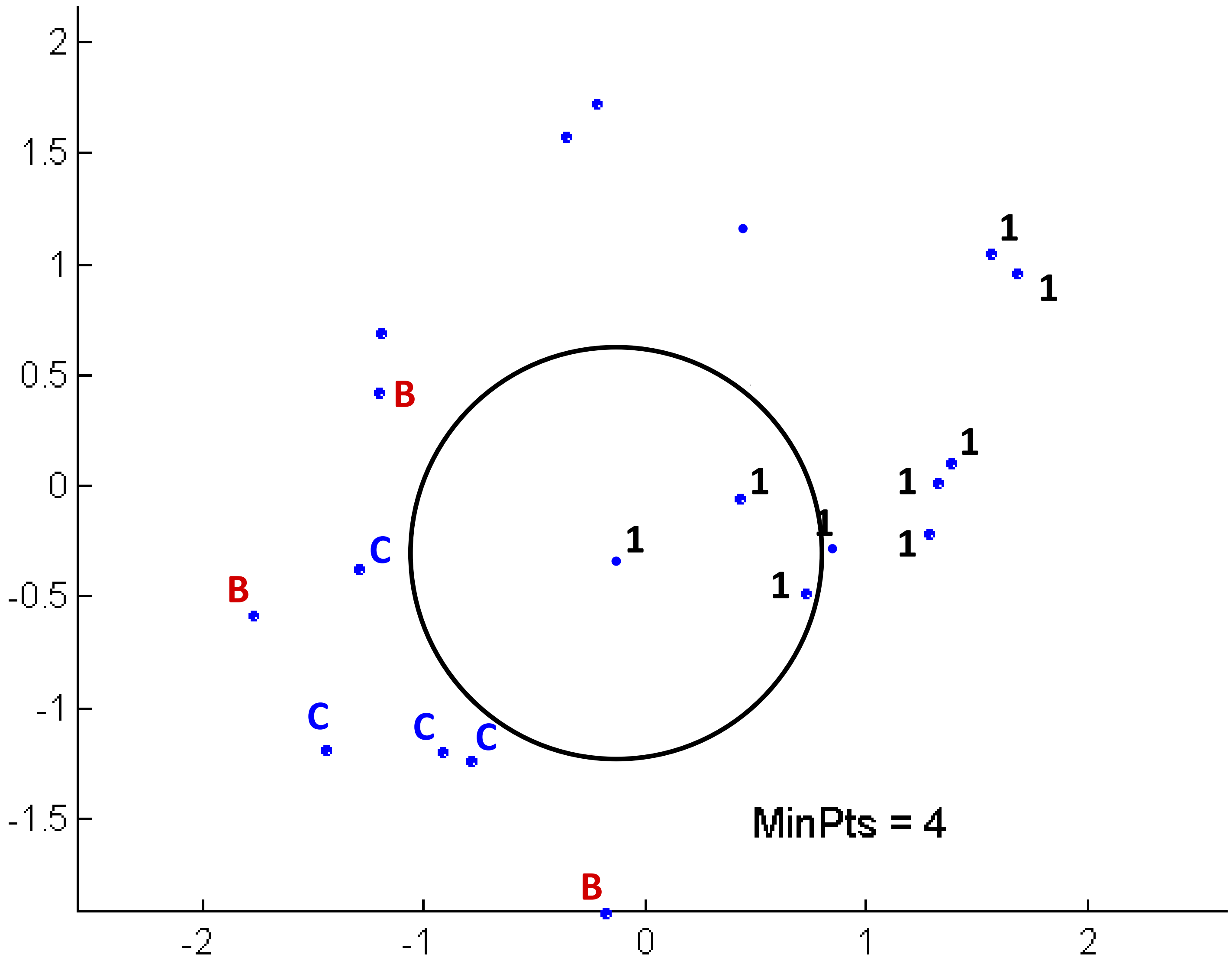
Ejemplo



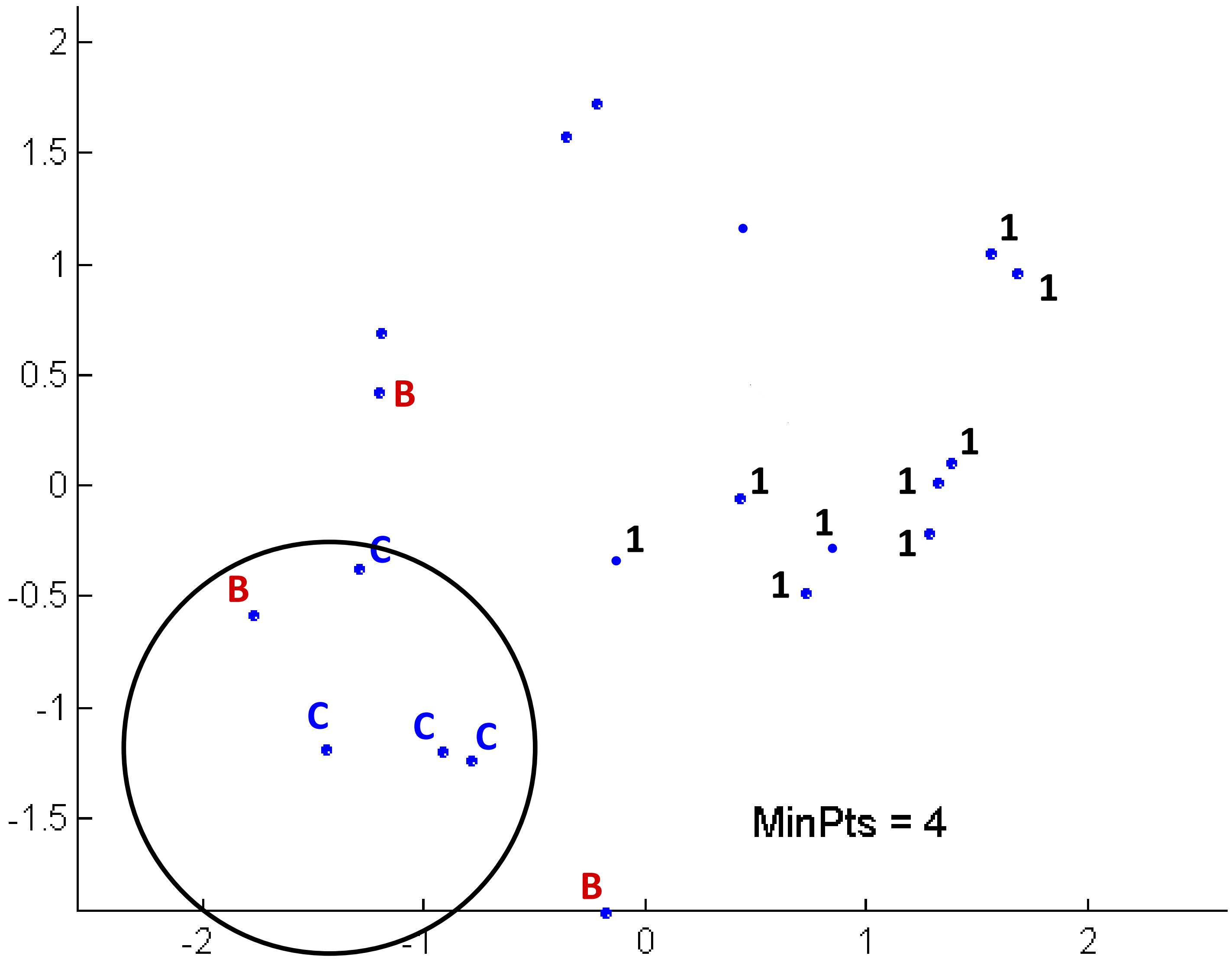
Ejemplo



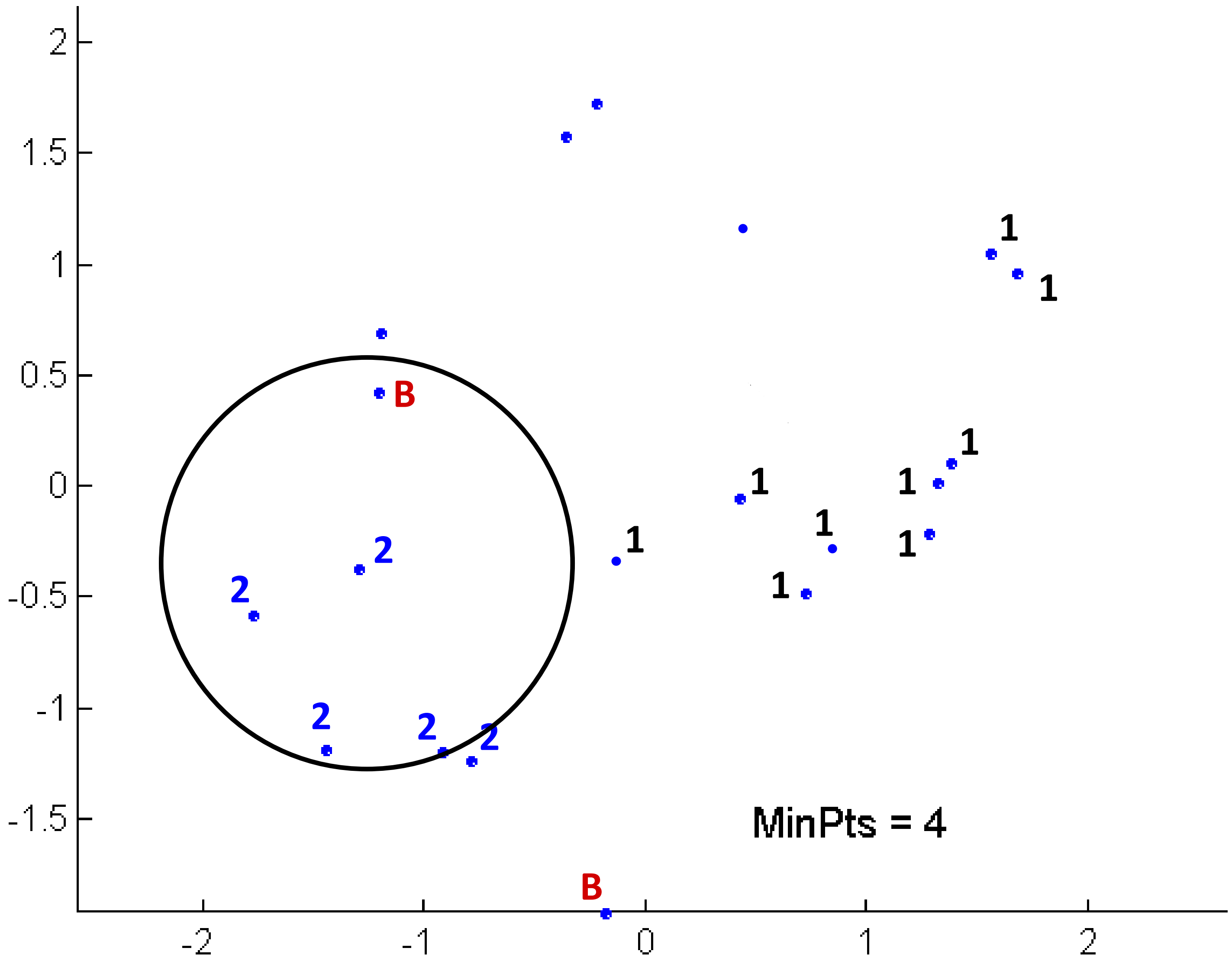
Ejemplo



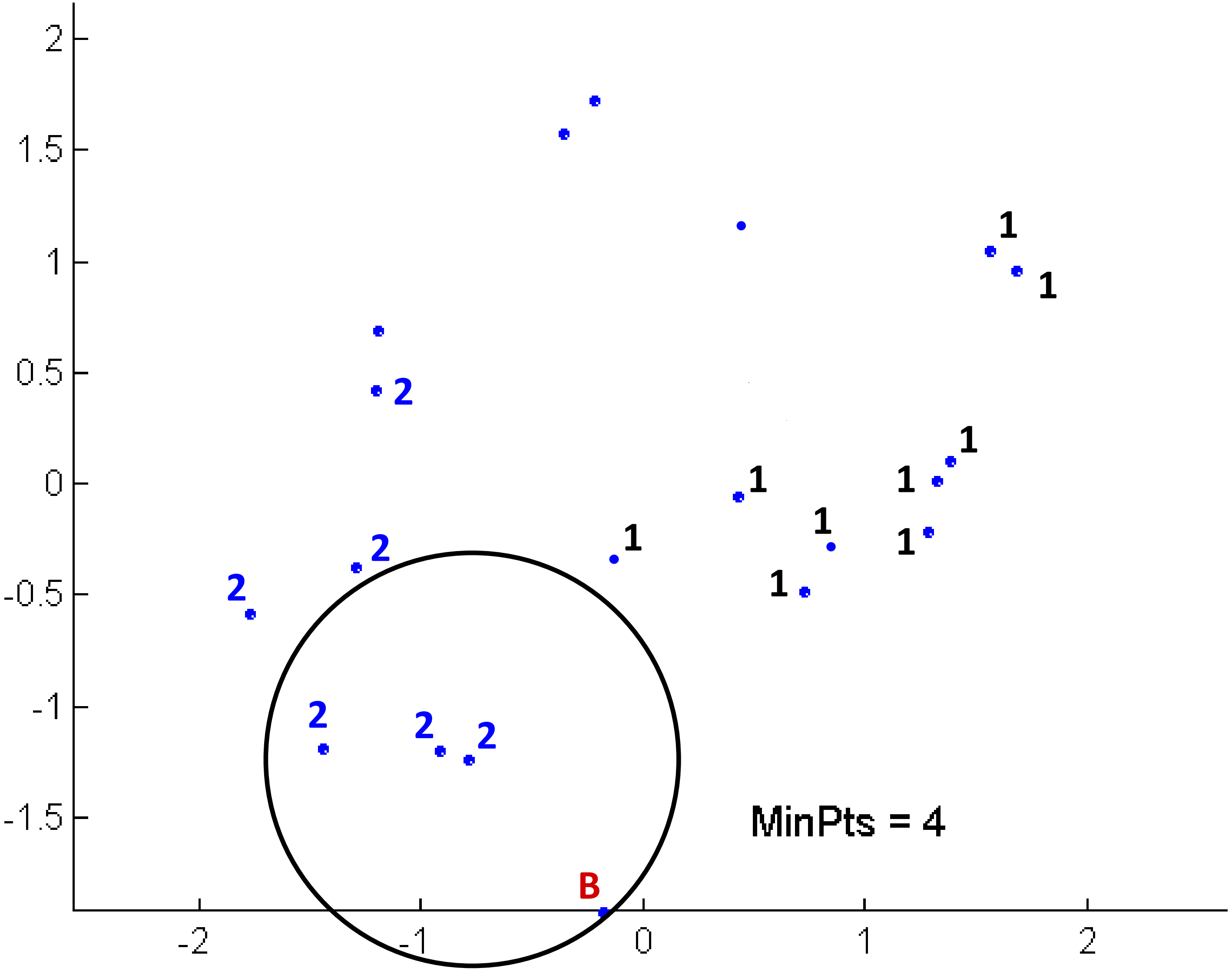
Ejemplo



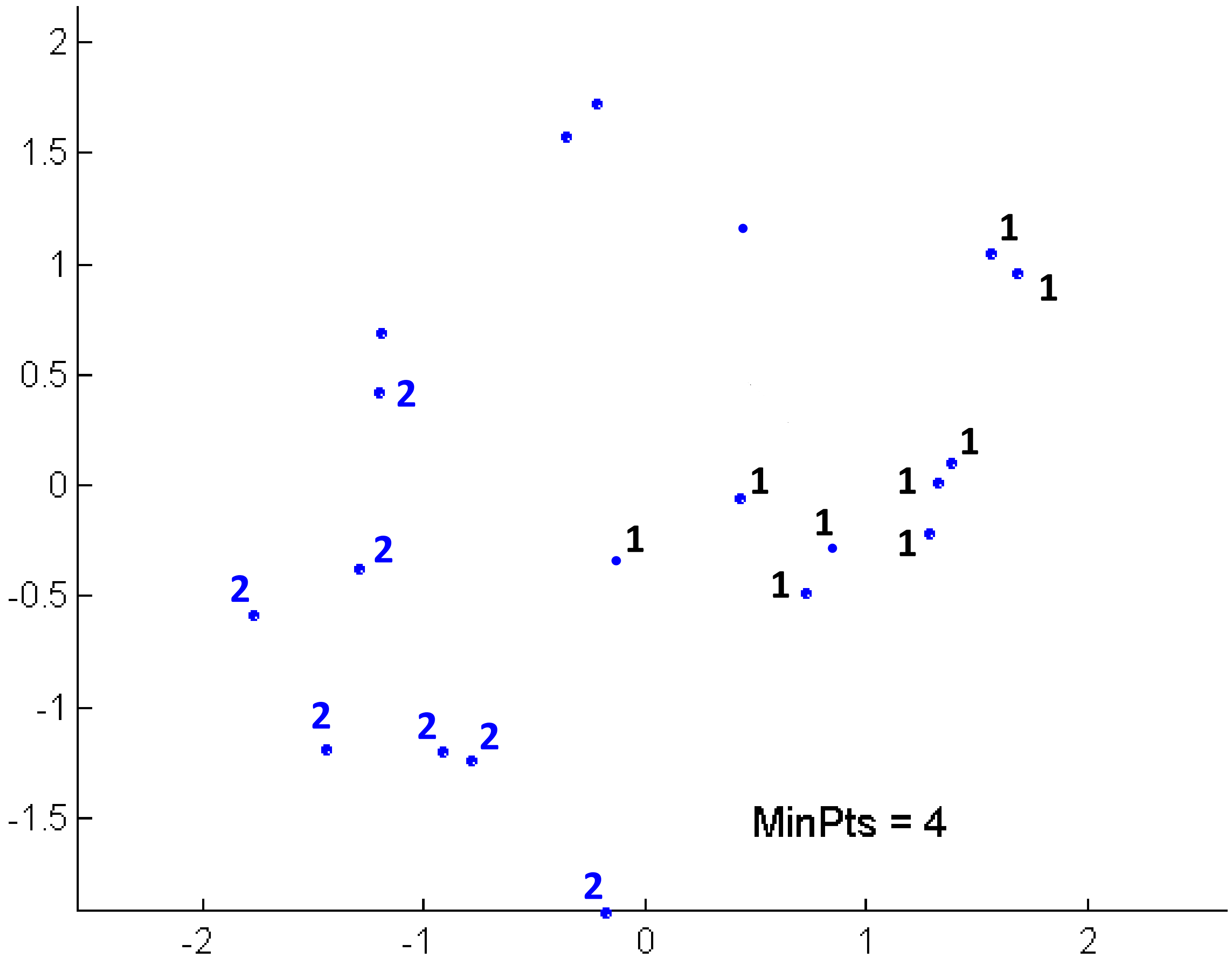
Ejemplo



Ejemplo



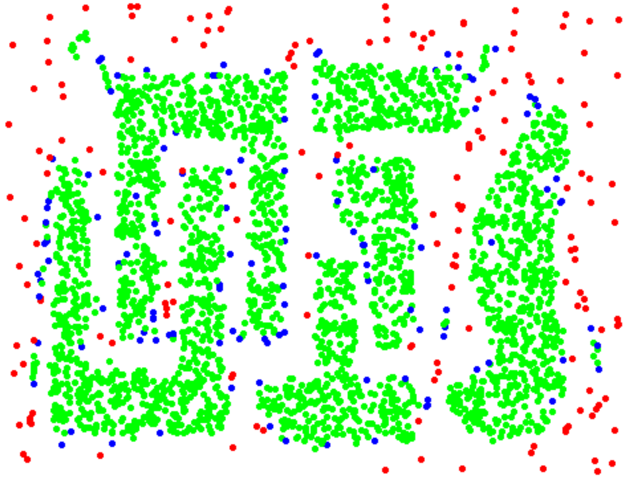
Ejemplo



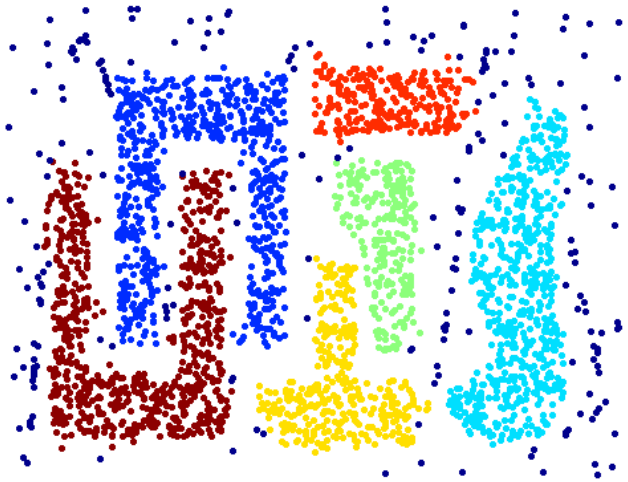
Ejemplo

Eps = 10, MinPts = 4

Puntos originales



Tipos de puntos:
centro, frontera y
ruido

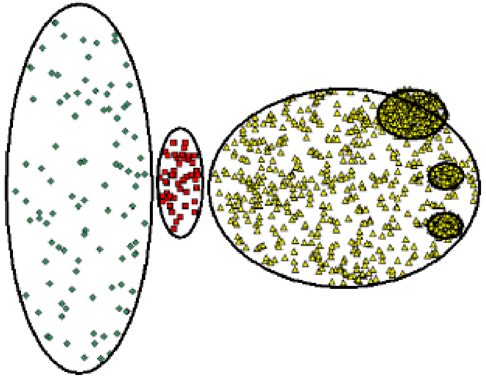


Clusters

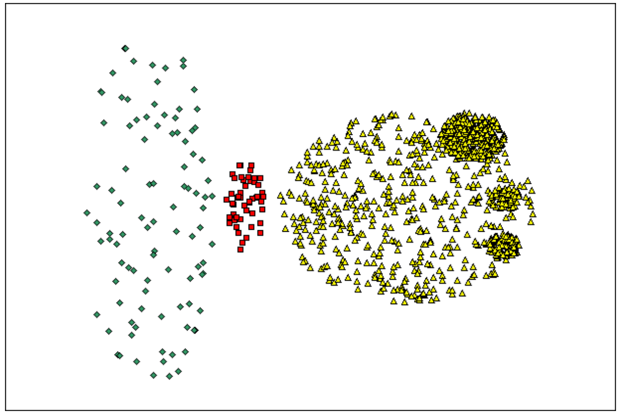
Fortalezas y debilidades

La complejidad del algoritmo es O (n^2), debido a la distancia entre todos los puntos.

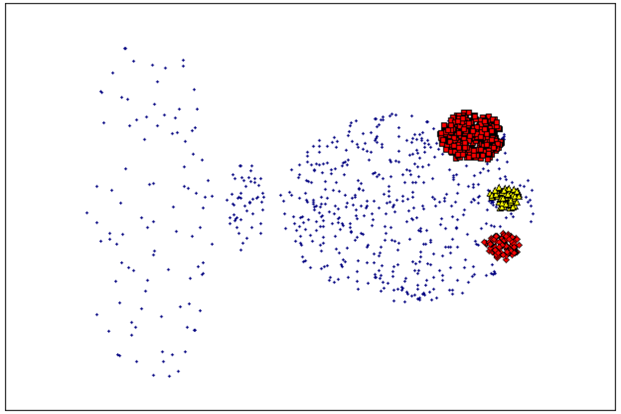
Fortalezas:
Resistente al ruido
Puede manejar grupos de diferentes formas y tamaños.



Debilidades:
Varianza de densidades
Datos de alta dimensión



MinPts = 4
Eps = 9,75



MinPts = 4
Eps = 9,92

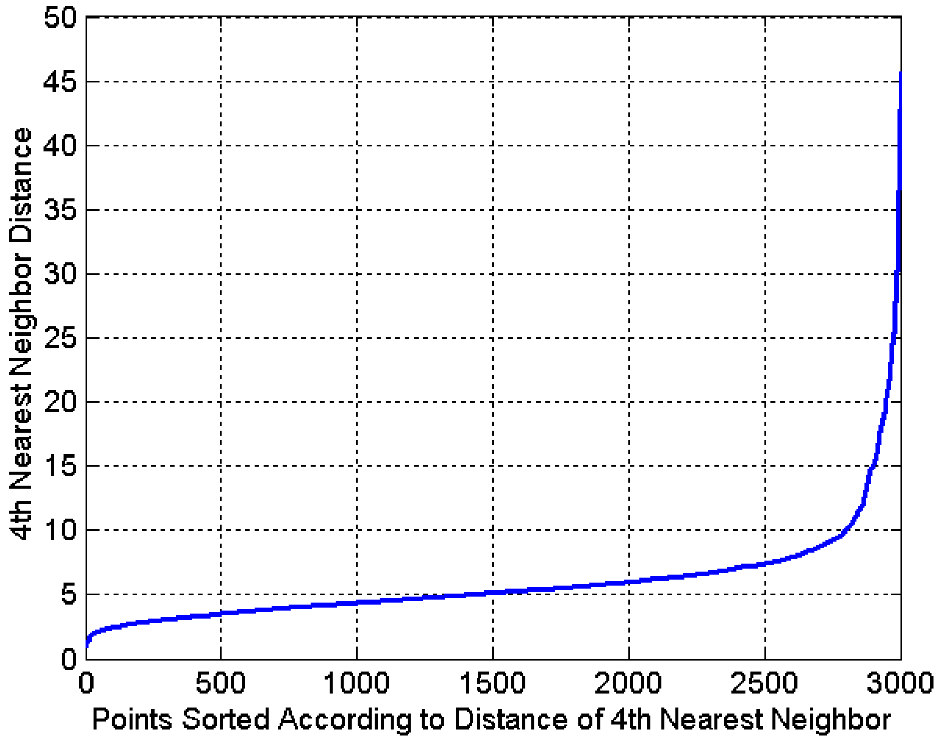
Ajustes de parámetros

Cómo seleccionar **EPS** y **MinPts**?

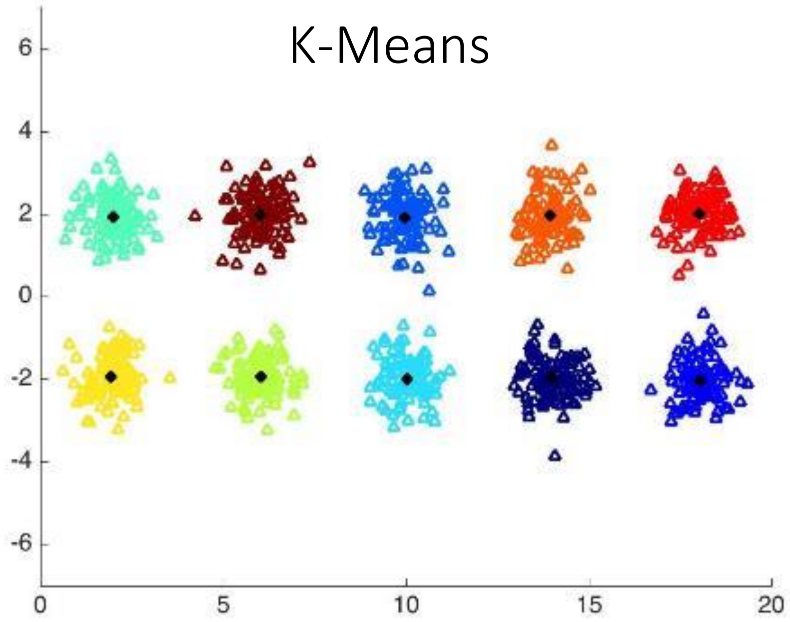
La idea es que para los puntos de un grupo, sus k vecinos más cercanos están aproximadamente a la misma distancia.

Los puntos de ruido tienen al k vecino más cercano a mayor distancia.

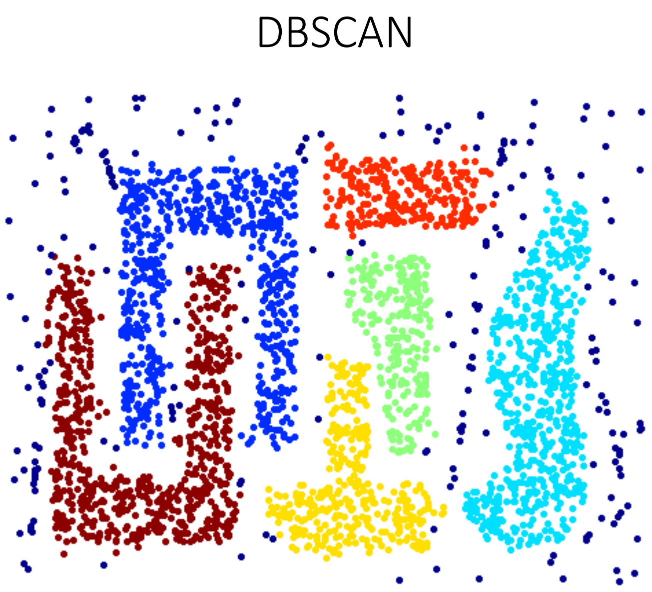
Trace la distancia ordenada de cada punto a su k vecino más cercano,
y seleccione **EPS** cerca del crecimiento exponencial.



K-medias y DBSCAN



- Eficiente O(K*n*i)
- Encuentra clusters esféricos
- Sensible a condiciones iniciales y k
- **Solo se puede usar cuando el "mean" esta definido (variables continuas)**
- Susceptible a los outliers
- **Existe variaciones (k-modes, k-medioids)**
- No es bueno con clusters de diferente tamaño / densidad

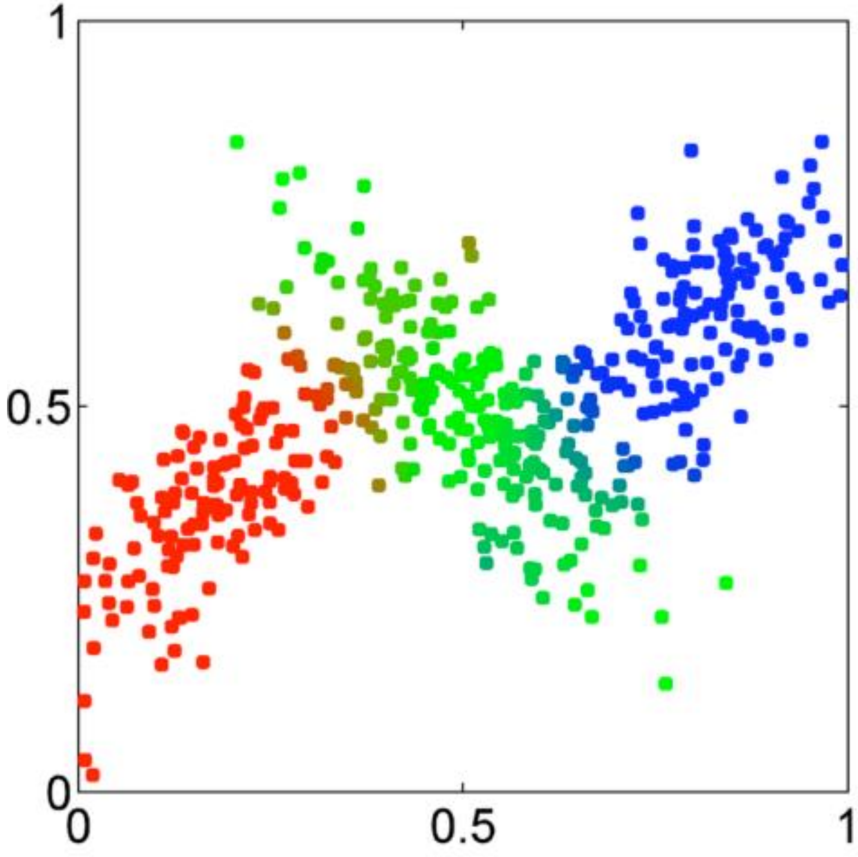


- Menos eficiente O(n^2)
- **Resistente al ruido**
- **Puede buscar clusters de distinto tamaño y forma**
- No es bueno en espacios con mucha variación de densidad entre puntos
- **No es tan bueno con altas dimensionalidades**

Soft clustering

Definición

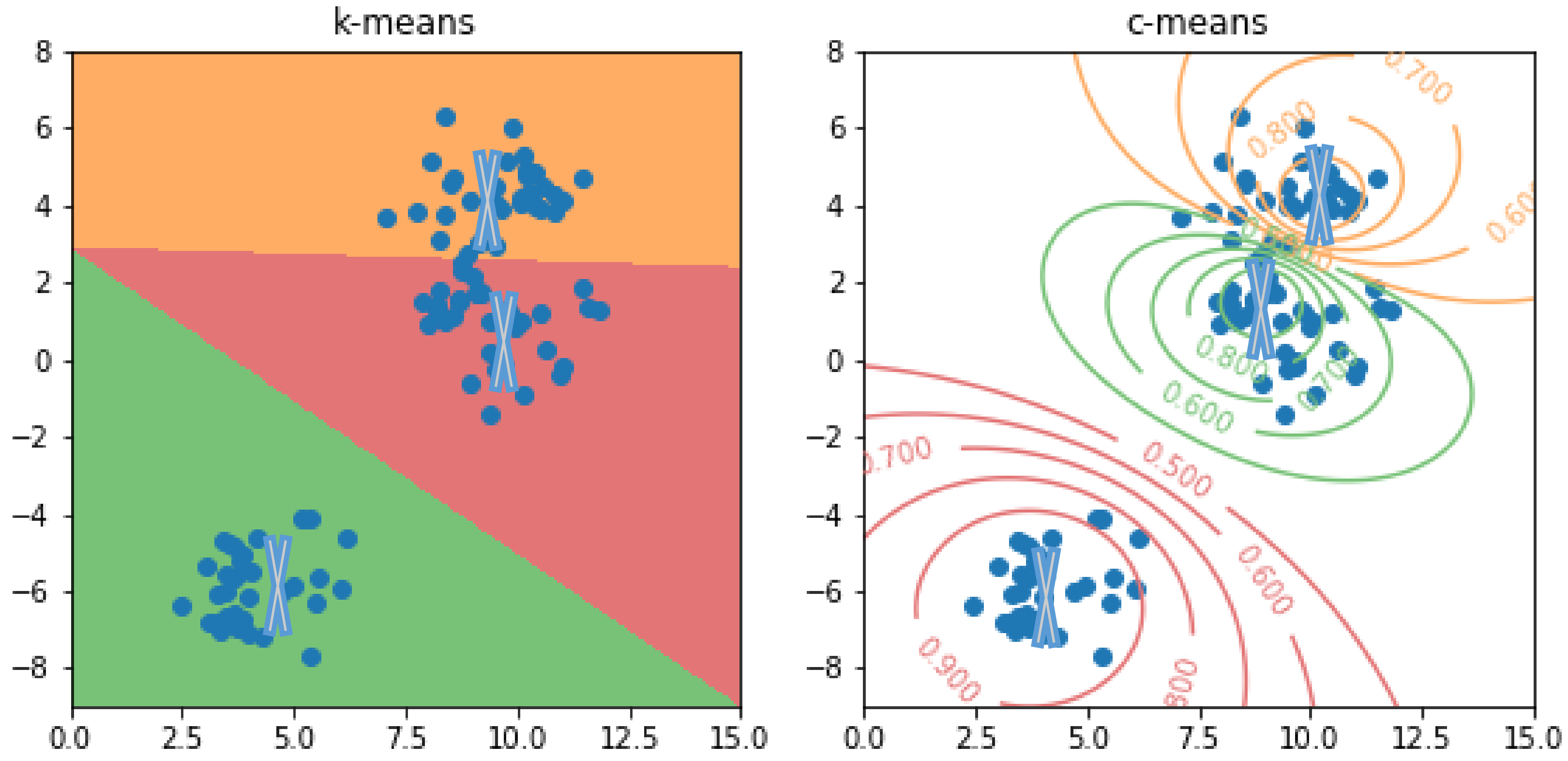
En los algoritmos de clústeres difusos, cada punto de datos puede tener una pertenencia o probabilidad de asignación distinto de cero a muchos (normalmente todos) clústeres.



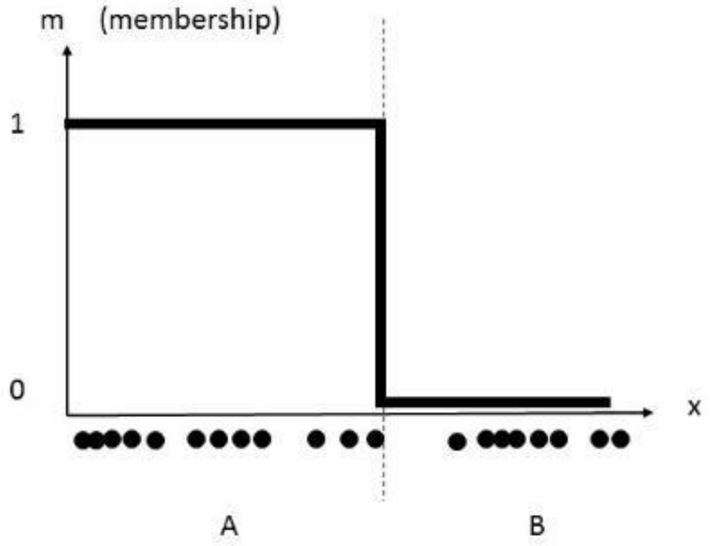
Fuzzy C-Means

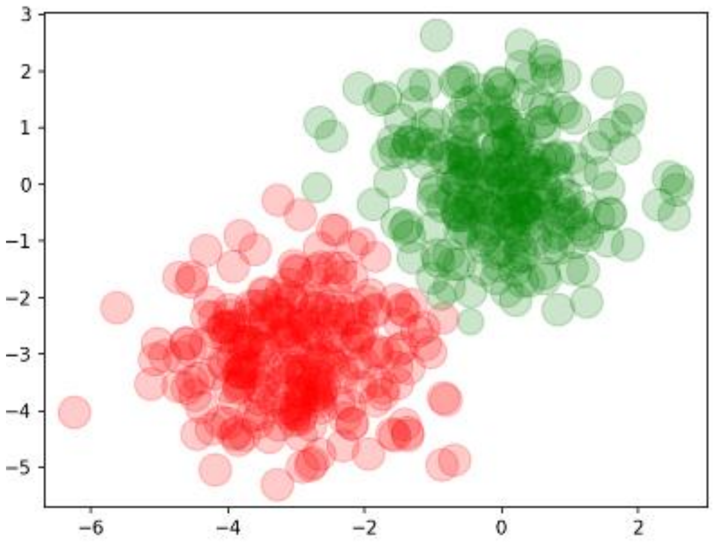
Fuzzy C-means

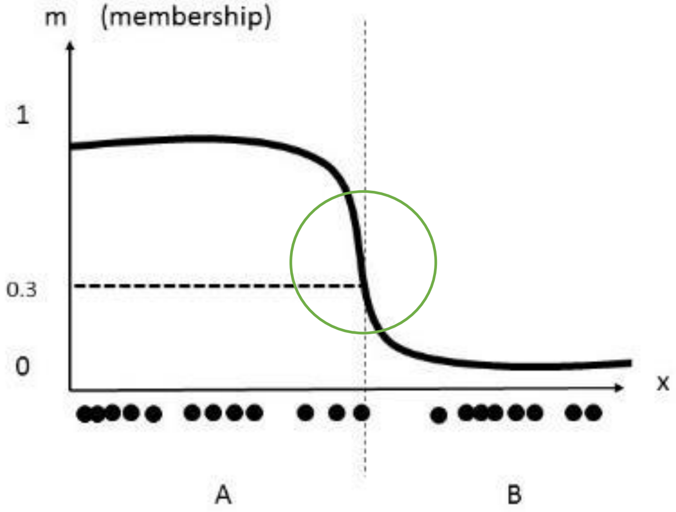
Permite que cada punto de datos pertenezca a más de un clúster. Dado el número de clúster K (determinado por el usuario), cada clúster está asociado a un centroide y cada punto se asigna a uno o varios clústeres en función de una medida de similitud.

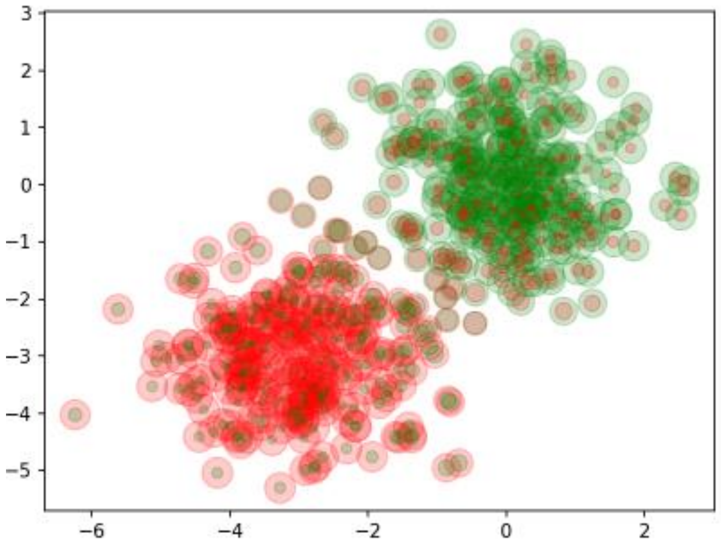


Pertenencia difusa









Algoritmo

- **Paso 1: Inicialización aleatoria de U de tal manera que:**

Número de clústeres, definido por el usuario

Número de entidades

$$\sum_{j=1}^{K} u_{ij} = 1, \quad \forall i = 1, \ldots, n$$

Pertenencia de punto i a cluster j

Matriz de pertenencia

K clusters X points	K j=1	K j=2	K j=3
X i=1	0,5	0,2	0,3
X i=2	1	0	0
X i=3	0,3	0,3	0,4
X i=4	0,1	0,1	0,8
X i=5	0	0,5	0,5

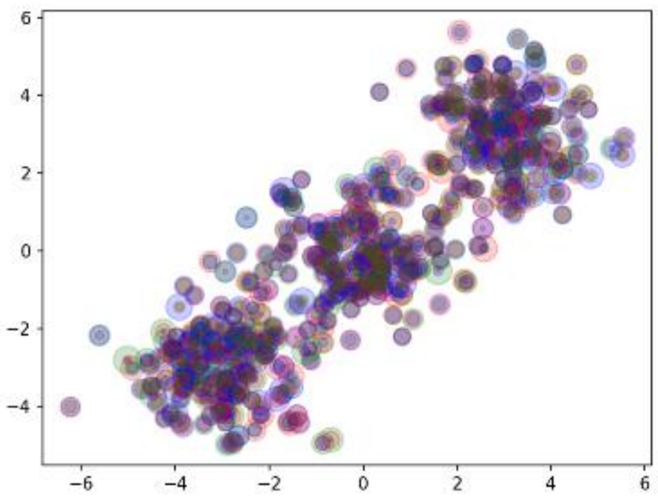
- **Paso 2: Seleccione K centroides basado en:**

m: Fuzzifier [1, ∞)

$$r_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} x(i)}{\sum_{i=1}^{n} u_{ij}^{m}}$$

X(i): Punto de datos i

Centro de cluser j



- **Paso 3: Calcule la función de costo wc(C).**

Si la función de costo obtiene un valor más alto o está por debajo de un umbral específico, DETENGA EL ALGORITMO.

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} u_{ik}^{m} d(x(i), r_k)$$

- **Paso 4: Vuelva a calcular la matriz de pertenencia utilizando:**

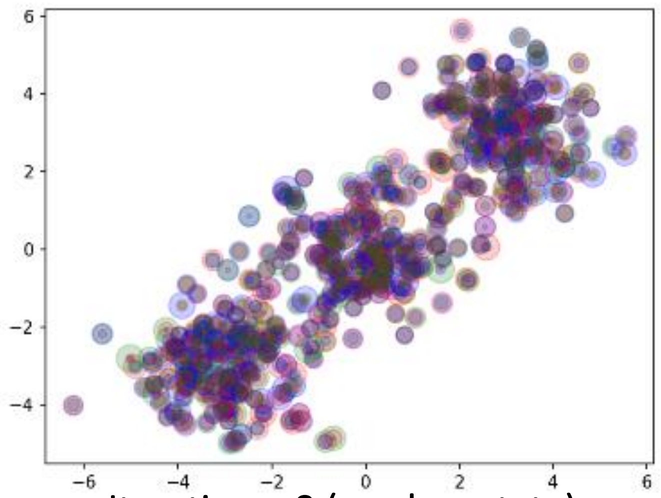
$$u_{ij} = \frac{\frac{1}{d(r_j, x(i))}^{\frac{1}{m-1}}}{\sum_{k=1}^{K} \left(\frac{1}{d(r_k, x(i))}^{\frac{1}{m-1}} \right)}$$

- **Paso 5: Volver al Paso 2**

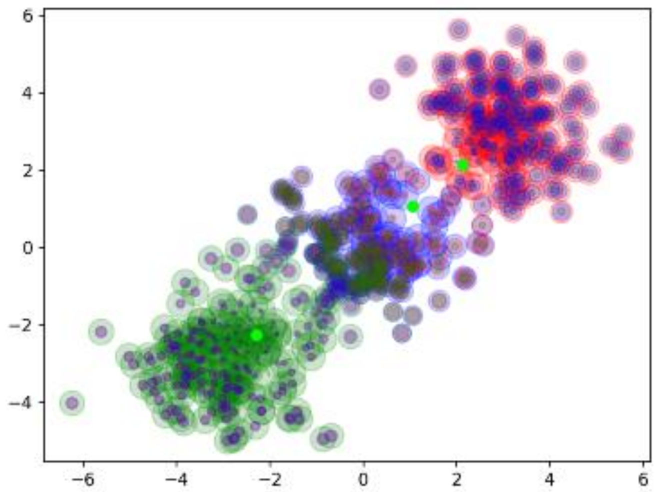
Matriz de pertenencia

K clusters X points	K j=1	K j=2	K j=3
X i=1	0,5 / 0,9	0,2	0,3 / 0,5
X i=2	0,3 / 1,3	0,3	0,4
X i=3	0,3 / 0,1	0,3 / 0,1	0,4 / 0,8
X i=4	0,1	0,1	0,8
X i=5	0 / 1	0,5	0,5

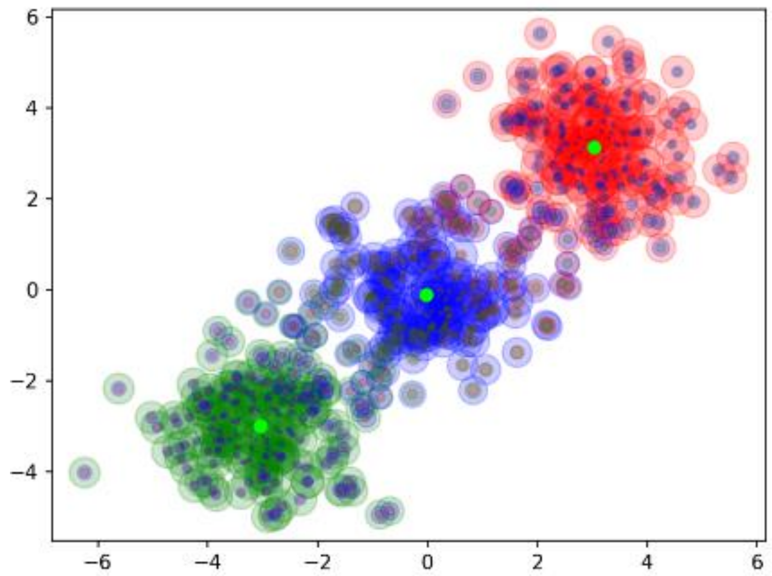
FCM: K=3 M=2



Iteration = 0 (random state)



Iteration = 4



Iteration = 10

Ajuste de parámetros

K y M se definen por el usuario

Función objetivo: generalización de la distancia dentro del clúster (cohesión)

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i)\in C_k} u_{ik}^{m} d(x(i), r_k)$$

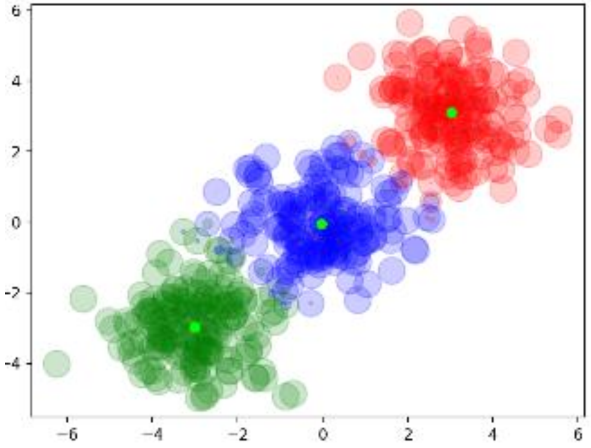
- **U es una matriz de pertenencia / peso con valores entre 0 y 1, de tal forma que**

$$\sum_{j=1}^{K} u_{ij} = 1, \quad \forall i = 1, \ldots, n$$

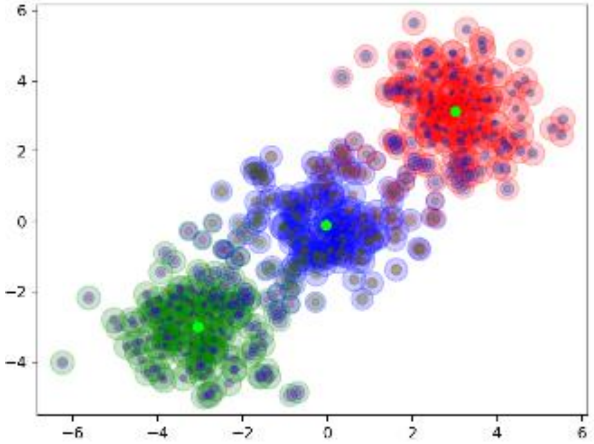
- m ∈ [1, ∞) es el fuzzificador que determina el nivel de difusividad del cluster.
- Un m grande da como resultado valores de pertenencia uij más pequeños, y por lo tanto, clústeres mas difusos. En el límite m=1, todos los uij convergen un 0 o 1, lo que implica un particionamiento dur equivalente a k-medias
- El estado inicial es aleatorio

Fuzzificador M

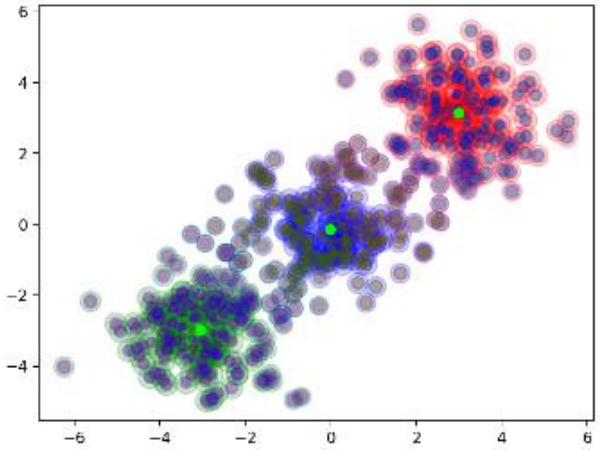
FCM: K=3



M=1.1



M=2



M=3

Coeficiente de partición difusa (FPC)

El coeficiente de partición difuso evalúa la variabilidad de las asignaciones.

$$F(U) = \frac{tr(U * U^T)}{n}$$

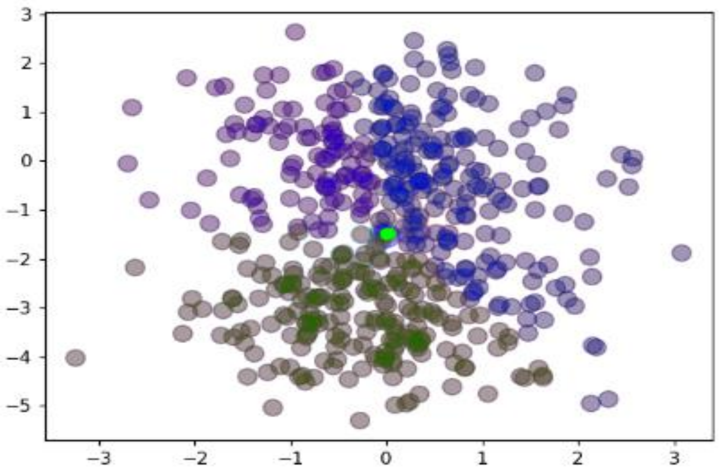
donde U es la matriz de pertenencia, * es el operador de multiplicación entre matrices, y tr( ) es la traza de la matriz, es decir, la suma de los valores diagonales.

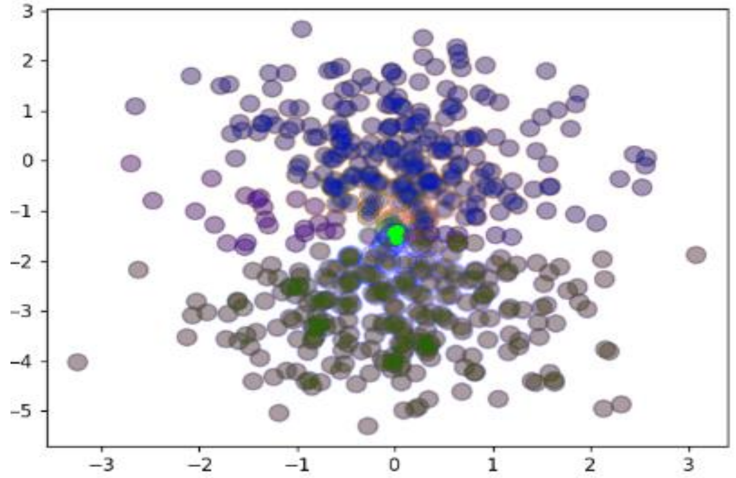
El coeficiente de partición difusa varía entre 0 y 1 donde un valor cercano a 1 implica menor variabilidad en la matriz de pertenencia, que se asocia a una mejor clusterización de los datos.

Coeficiente de partición difusa (FPC)

Puede utilizar FPC para buscar los mejores valores para los parámetros, como en k-medias

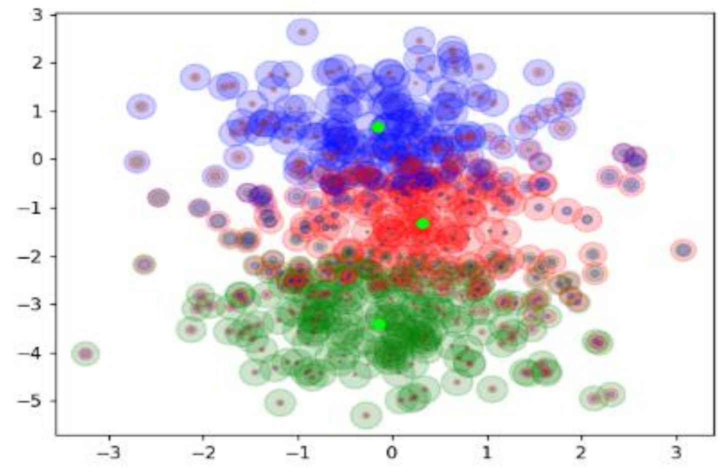
FPC=0.11166

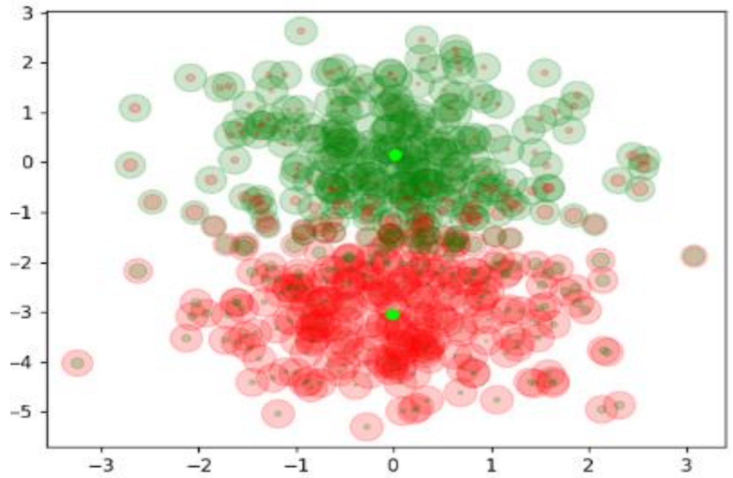




FPC=0.117

FPC=0.27





FPC=0.45

GMM

Definición

Un modelo de mezcla gaussiana (GMM) asume que los datos se generaron a partir de una mezcla de K gaussianos multidimensionales, donde cada componente tiene parámetros: Nk(μk,∑k).

K es definido por el usuario.

Mean vector (like mean in 1D)

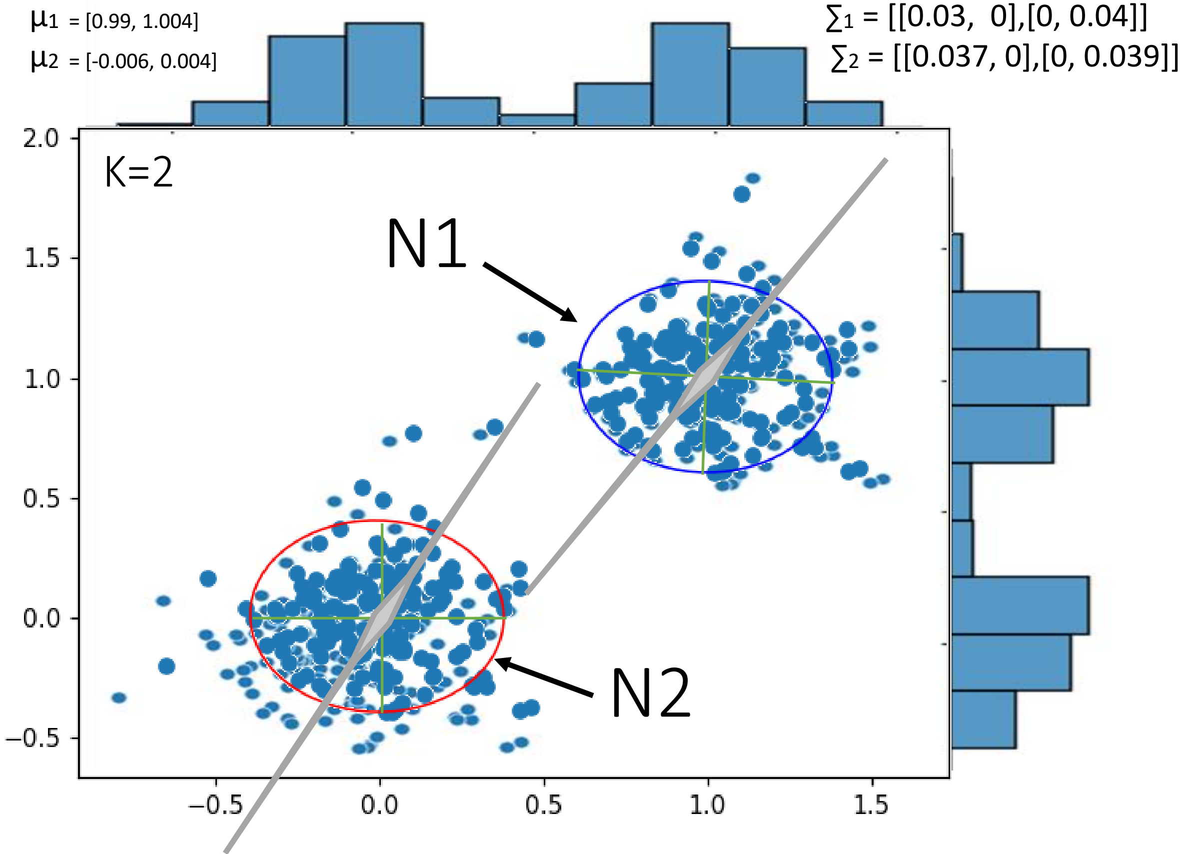
μ_1 = [0.99, 1.004]

μ_2 = [-0.006, 0.004]

Covariance Matrix (Like SD in 1D

$\sum_1$ = [[0.03, 0],[0, 0.04]]

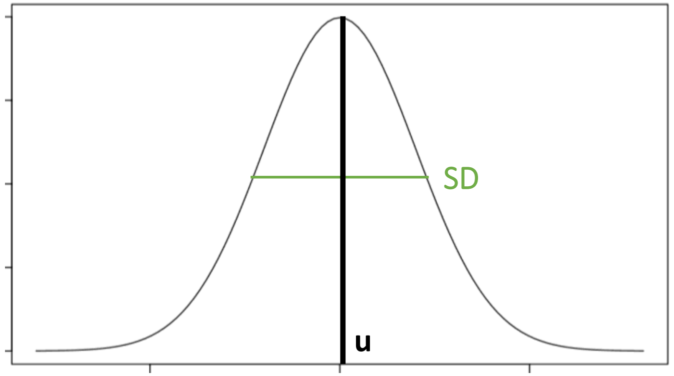
$\sum_2$ = [[0.037, 0],[0, 0.039]]



Gaussiano multivariado

$$X \sim \mathcal{N}(\mu, \sigma^2). \qquad \mu = \frac{\sum X}{N}$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \qquad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$



Un Gaussiano multidimensional, para datos con dimensiones p se especifica de la siguiente manera:

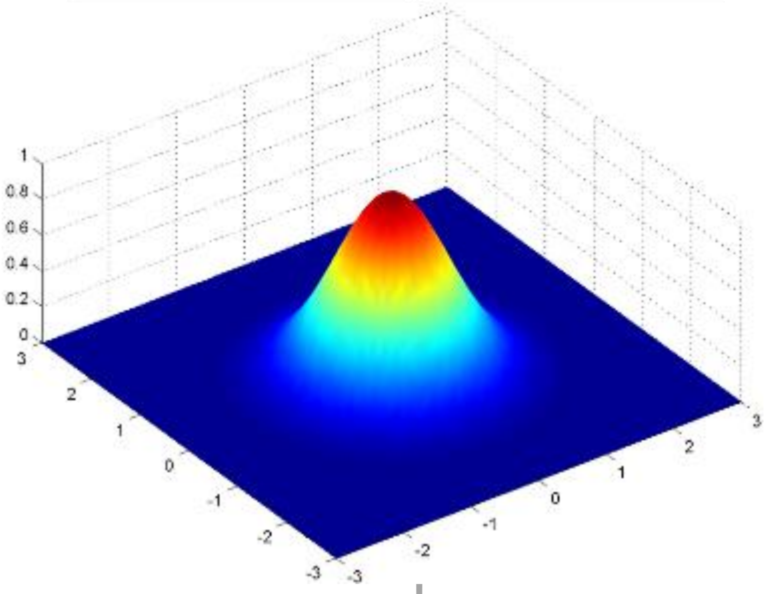
Dónde:

$$x \sim N(\mu, \Sigma)$$

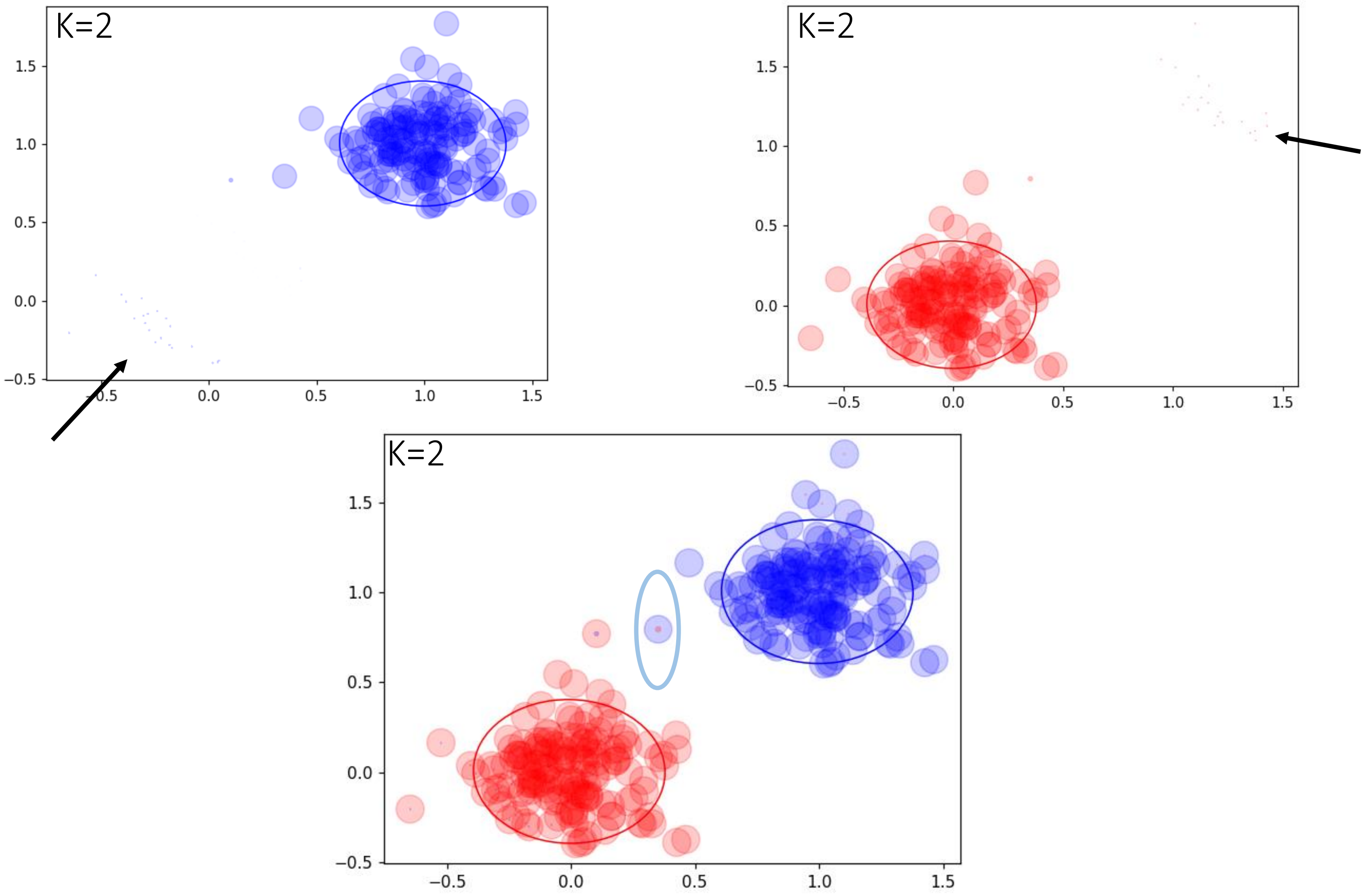
$$\mu = (E[X_1], \ldots, E[X_p])$$

$$\Sigma = \begin{bmatrix} Var(X_1) & \cdots & Cov(X_1, X_p) \\ \cdots & \cdots & \cdots \\ Cov(X_1, X_p) & \cdots & Var(X_p) \end{bmatrix}$$

$$p(\mathbf{x}) = p(x_1, \cdots, x_p) = \frac{1}{\sqrt{(2\pi)^p|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right)$$



GMM



GMM

Prob over total (simple graph)

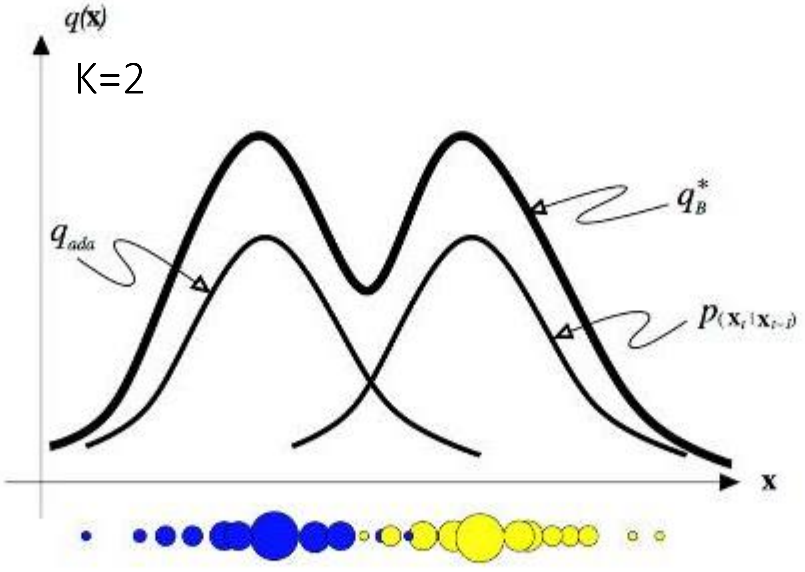
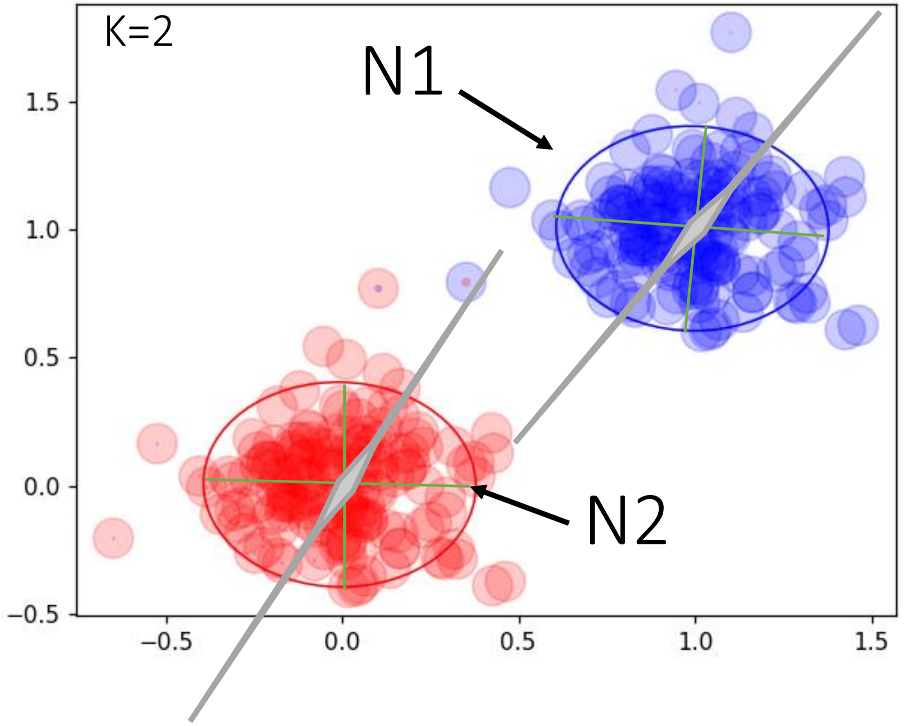
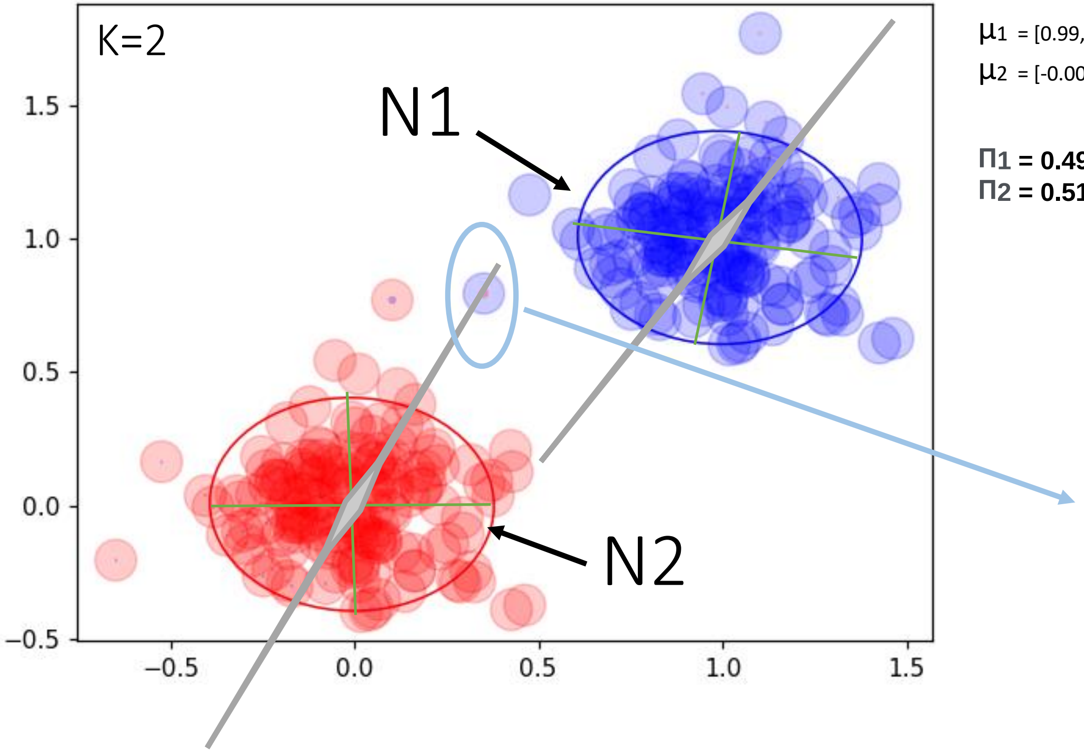


Fig. 5. Mixture of Gaussians for the proposal distribution.

Model GMM (K component)



GMM



μ_1 = [0.99, 1.004]
μ_2 = [-0.006, 0.004]

Σ_1 = [[0.03, 0],[0, 0.04]]
Σ_2 = [[0.037, 0],[0, 0.039]]

Π_1 = 0.49
Π_2 = 0.51

Cada punto tiene una probabilidad de pertenecer a N1 y N2

p(x) = p(N1)p(x|N1) + p(N2)p(x|N2)

p(N1)p(x|N1) << p(N2)p(x|N2)

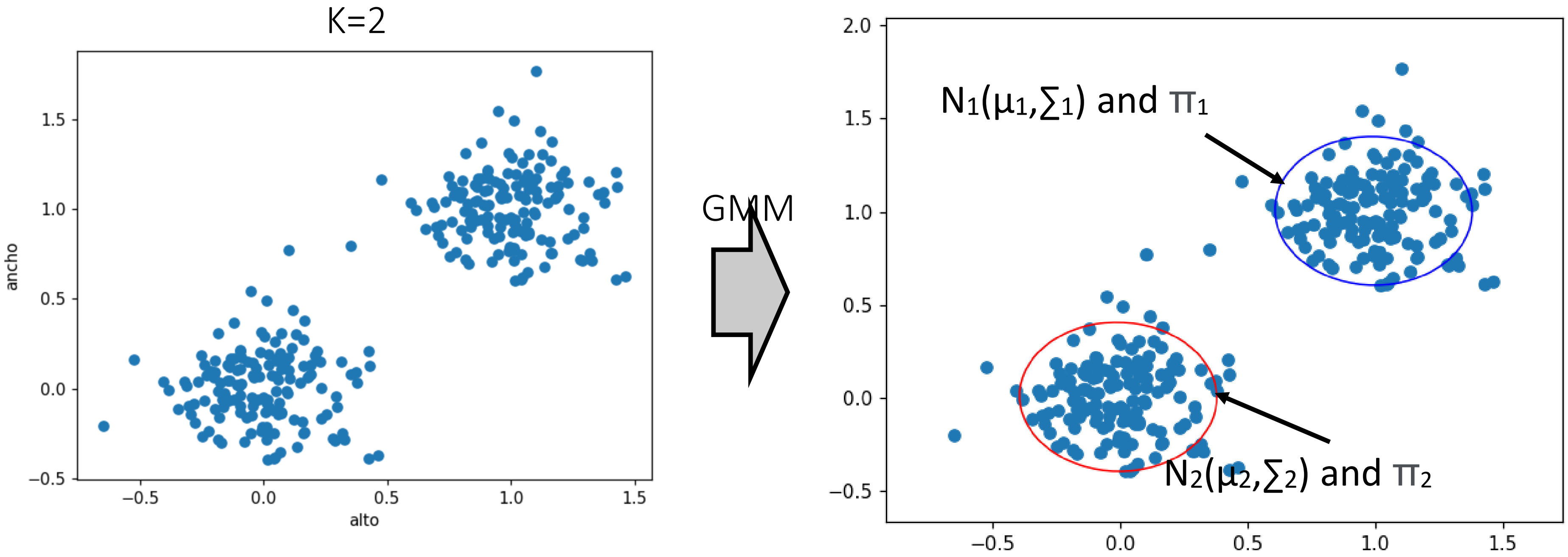
$$p(x) = \sum_{k=1}^{K} p(k)\overbrace{p(x|k)}^{\text{P de x dado k}} = \sum_{k=1}^{K} p(k)p(x|x \sim N(\overbrace{\mu_k}^{\text{Media de k}}, \underbrace{\Sigma_k}_{\text{Matriz de covarianza de k}})) = \sum_{k=1}^{K} \pi_k \overbrace{N(\mu_k, \Sigma_k)}^{\text{Gauss Dist}}$$

P de k sobre el total de K

Peso de k

Algoritmo

Dado un K, ¿cómo encontramos el mejor valor para cada μk, ∑k y πk?



MAX

$$p(x) = \sum_{k=1}^{K} p(k)p(x|k) = \sum_{k=1}^{K} p(k)p(x|x \sim N(\mu_k, \Sigma_k)) = \sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)$$

Algoritmo

- Maximiza la verosimilitud de todo el conjunto de datos:

$$\overbrace{P(X|\pi, \mu, \Sigma)}^{\text{MAX LIKELIHOOD}} = \prod_{i=1}^{N} \left[\sum_{k=1}^{K} \pi_k \overbrace{N(x_i | \mu_k, \Sigma_k)}^{\text{P of } i \text{ in Gauss Dist } k} \right]$$

DataSet X with N points

Each point *i*

Each cluster *k*

Weight of *k*

- La suma de todos los pesos tiene que ser 1:

$$\sum_{k=1}^{K} \pi_k = 1$$

Tenemos que ajustar µk, ∑k y πk para maximizar la probabilidad = Derivar verosimilitud con respecto a µk, ∑k y πk

Algoritmo

- Defina algunas cantidades auxiliares:

$$Z_{nk} = \begin{cases} 1 & \text{if } x_n \text{ is in cluster } k \\ 0 & \text{otherwise.} \end{cases}$$

$$\gamma(Z_{nk}) = P(Z_{nk} = 1|X_n) = \frac{P(Z_k=1)P(X_n|Z_{nk}=1)}{\sum_{j=1}^{K} P(Z_{nj}=1)P(X_n|Z_{nj}=1)} = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$$

(Bayes thm.)

$$N_k = \sum_{n=1}^{N} \gamma(Z_{nk})$$

La derivada de: $P(X|\pi,\mu,\Sigma) = \prod_{i=1}^{N} \left[\sum_{k=1}^{K} \pi_k N(x_i|\mu_k,\Sigma_k)\right]$ respecto con μk, ∑k y πk

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk}) x_n$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk})(x_n - N_k)(x_n - N_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Gamma => Mu, Sigma, Pi

Mu, Sigma, Pi => Gamma

Dependencia circular=> algoritmo

Resumen algoritmo

- Maximización de verosimilitud (EM)

  - Paso 0) Seleccione K

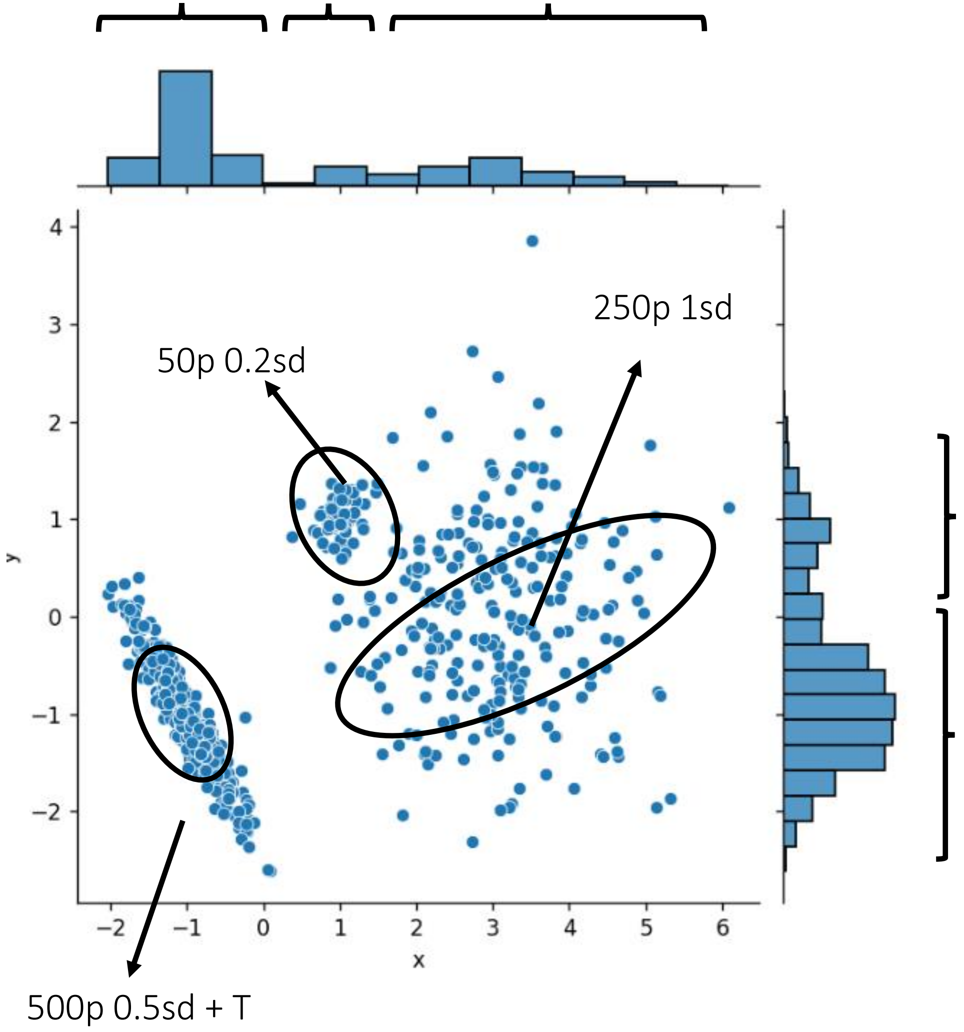
  - Paso 1) Inicializar μk, $\sum k$ y πk con valores aleatorios (inicializar Q veces, utilizar una función de puntuación, tomar la mejor condición inicial)

  - Paso 2) Calcular gamma (paso de expectativa)

  - Paso 3) Recalcular μk, $\sum k$ y πk con Gamma (Paso de Maximización)

  - Paso 4) Compruebe la convergencia de los parámetros, si la convergencia no se satisface vaya al paso 2

Ejemplo



Datos generados artificialmente

Pistas:

Ocupe 3 blobs **con diferente numero de puntos:** 500, 250 y 50

Los 3 blobs tienen dista densidad (SD): 0.5, 1 y 0.2

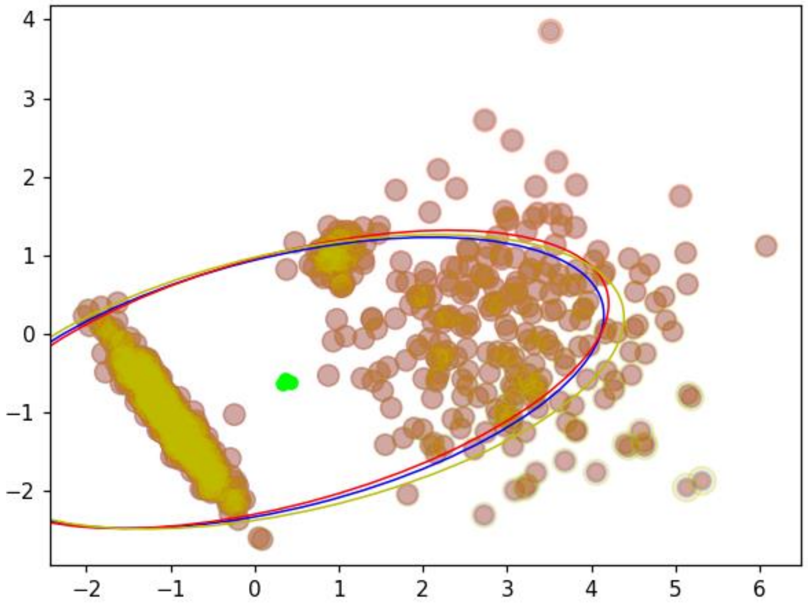
2 de ellos son esféricos, 1 de ellos a sufrido una transformación para alargarlo.

¿Cómo deberían ser los pesos de π?

Supongamos que no sabemos nada de lo anterior y usemos GMM

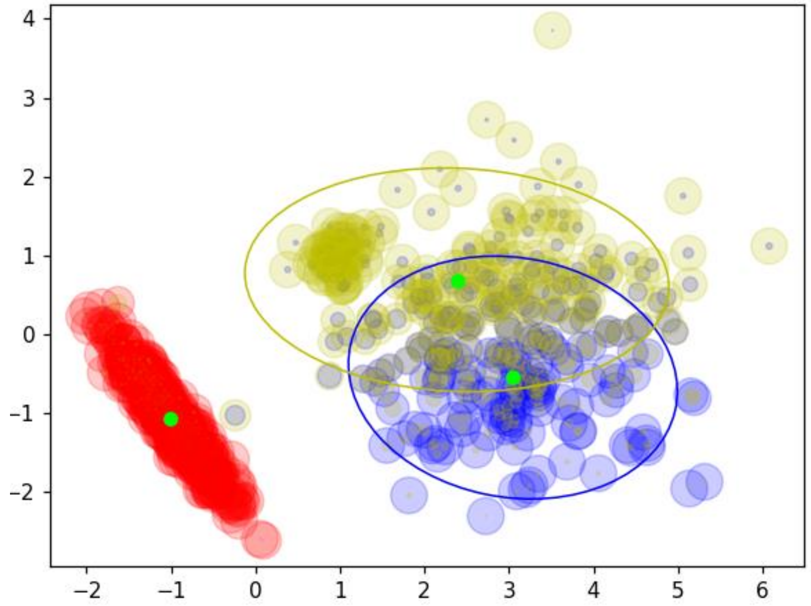
Ejemplo

K=3, 100 inicios random



Iteración=1

π1=0.333 π2=0.329 π3=0.337

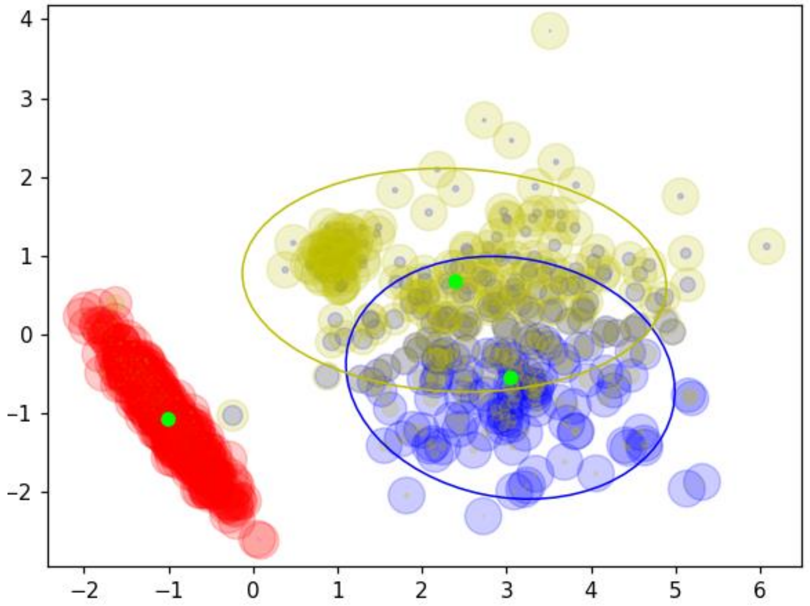


Iteración=5

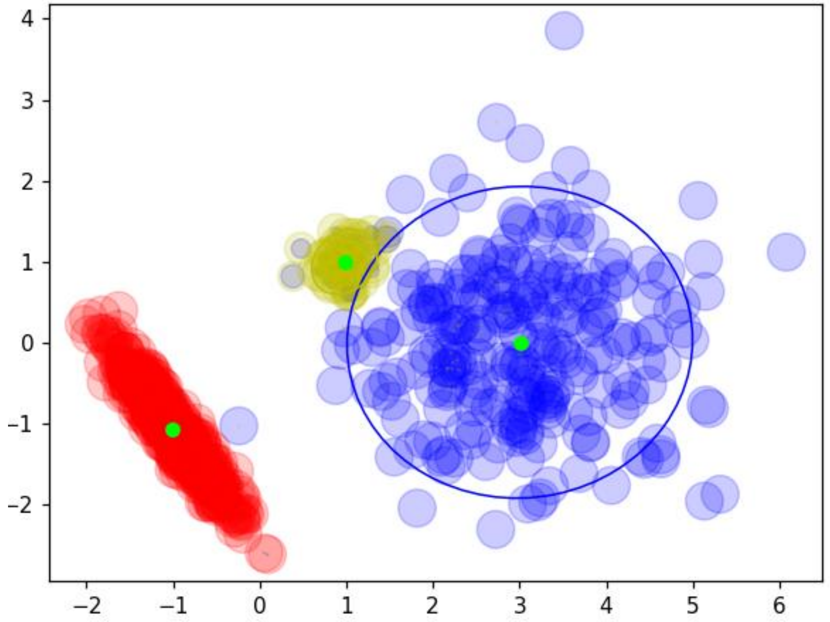
π1= 0.15 π2=0.62 π3=0.21

Ejemplo

K=3, 100 inicios random



Iteración=5



Iteración=10

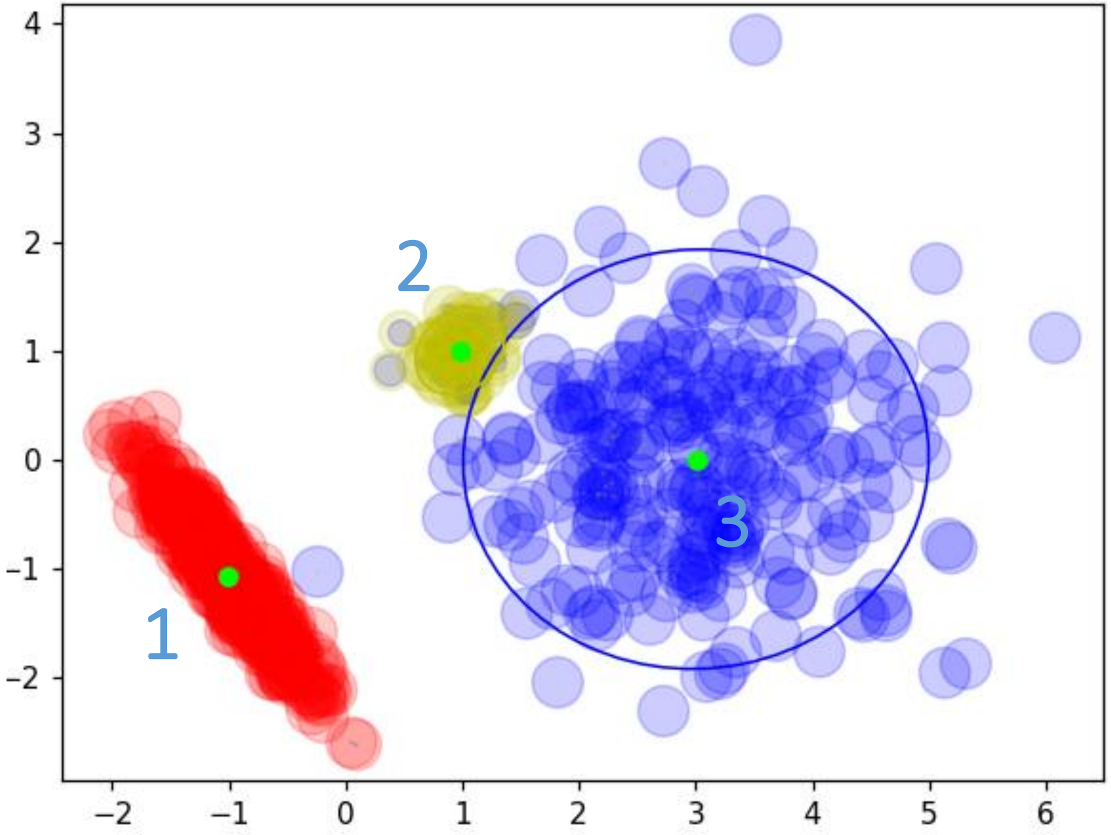
π1= 0.15 π2=0.62 π3=0.21

π1= 0.31 π2=0.62 π3=0.06

Ejemplo

Vista por distribuciones aisladas

K=3, 100 inicios random



Iteración=10

π_1= 0.62
μ_1= [-1.003, -1.065]
Σ_1= [[0.13, -0.18], [-0.18, 0.27]]

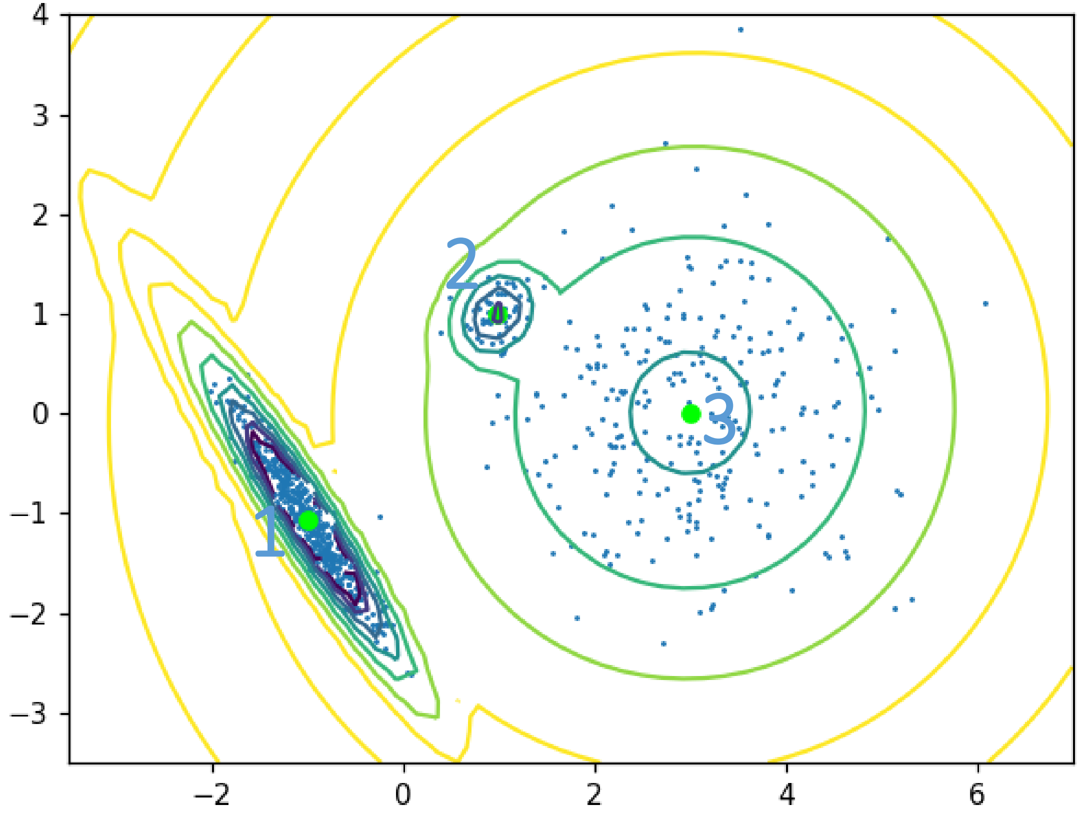
π_2=0.06
μ_2= [0.978, 0.999]
Σ_2= [[0.03, 0.008], [0.008, 0.039]]

π_3=0.31
μ_3= [3.001, 0.007]
Σ_3= [[0.99, 0.015], [0.015, 0.92]]

Ejemplo

Vista de la probabilidad total de cada punto

K=3, 100 inicios random



Iteración=10

$\pi1= 0.62$
$\mu1= [-1.003, -1.065]$
$\sum1= [[0.13, -0.18], [-0.18, 0.27]]$

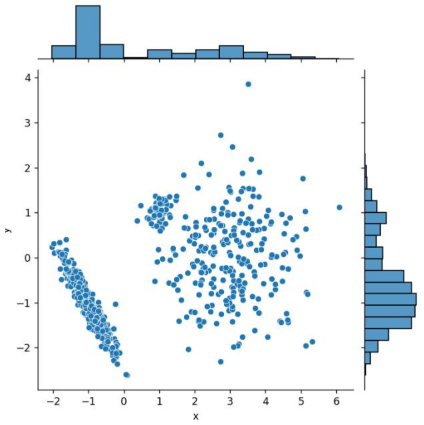
$\pi2=0.06$
$\mu2= [0.978, 0.999]$
$\sum2= [[0.03, 0.008], [0.008, 0.039]]$

$\pi3=0.31$
$\mu3= [3.001, 0.007]$
$\sum3= [[0.99, 0.015], [0.015, 0.92]]$

Selección de K

- Si elegimos K para maximizar la probabilidad, cuando K aumenta el valor de la probabilidad máxima no puede disminuir.
- Por lo tanto, los modelos más complejos siempre mejorarán la probabilidad.
- Es necesario penalizar la complejidad del modelo.
- Necesitamos un equilibrio entre lo bien que encaja el modelo y los datos y la simplicidad del modelo
  - Score(θ,M) = error(M) + penalización (M)
    La penalización puede depender del número de parámetros del modelo (p) y del número de puntos de datos (n).
  - El error se basa generalmente en la probabilidad de los datos dados el modelo (L).
  - AIC Akaike information criterion $Score_{AIC} = -2 \log L + 2p$
  - BIC: Bayesian information criterion $Score_{BIC} = -2 \log L + p \log n$

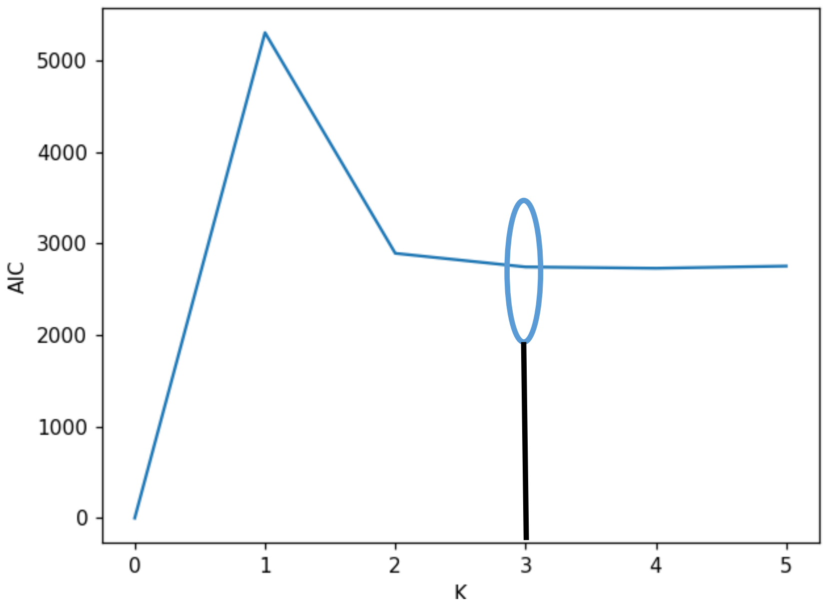
Selección de K

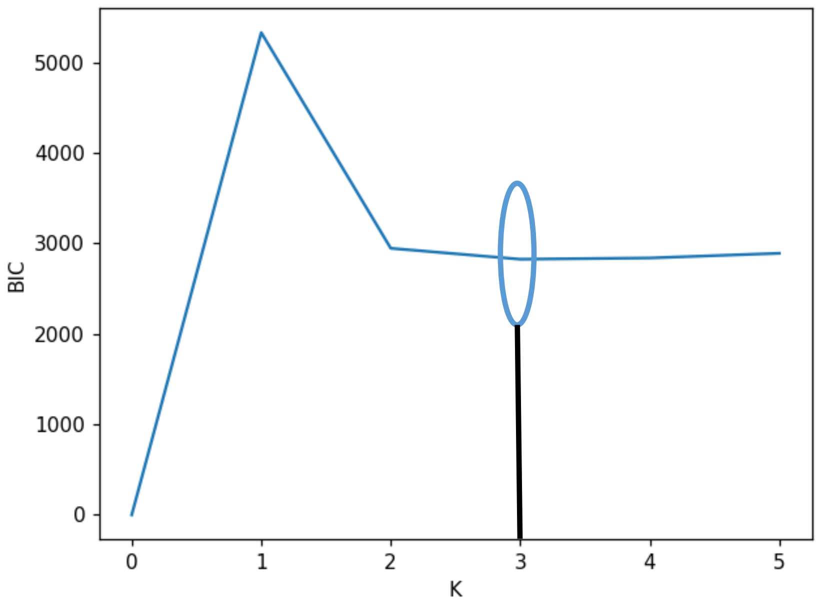


AIC and BIC

100 inicios aleatorios

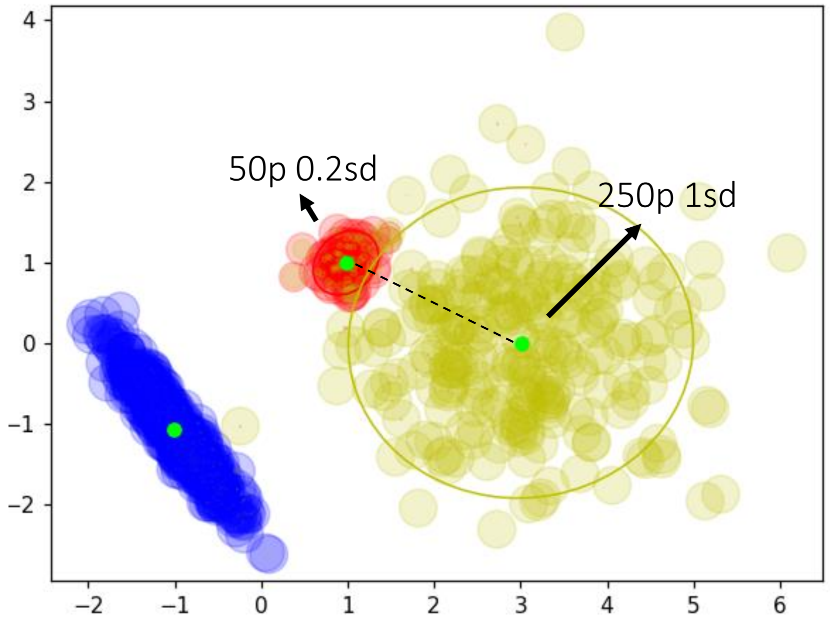
Hay poca mejora entre K=2 y K=3 debido a la distribución de los datos.





Ejemplo

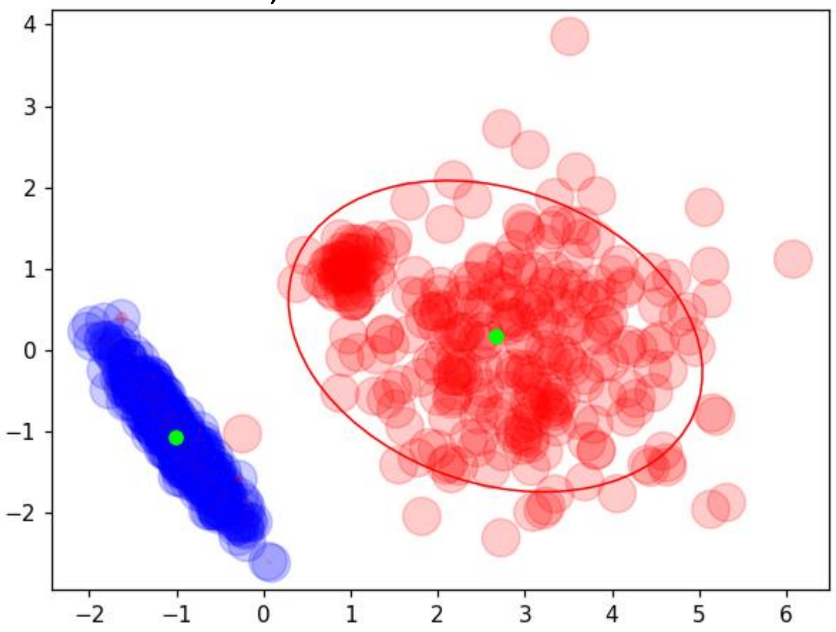
K=3, 100 inicios random



AIC=2745

BIC=2825

K=2, 100 inicios random



AIC=2894

BIC=2945

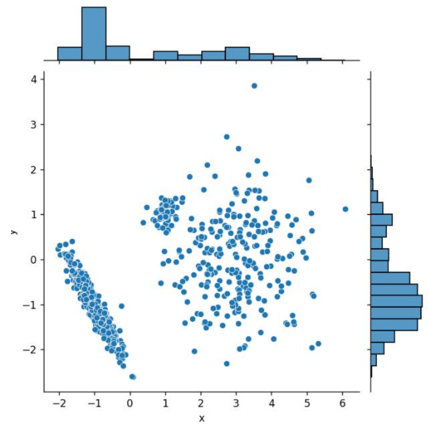
Un cluster puede asimilar a otro muy cercano y que posea pocos datos.
Hay que tener cuidado al analizar los scores. Como también generar suficientes condiciones iniciales y usar suficientes iteraciones.

Condiciones iniciales

Opciones para definir la condición inicial de μk $\sum k$ y πk para un cierto K

1. El usuario define todos o algunos de ello
2. Utilice otro método, como los medios K, para buscar en el centroide de los clústeres preliminares (μk)
3. Q iteraciones aleatorias iniciales. Puntúa cada uno de ellos y toma el mejor (AIC, BIC o verosimilitud).

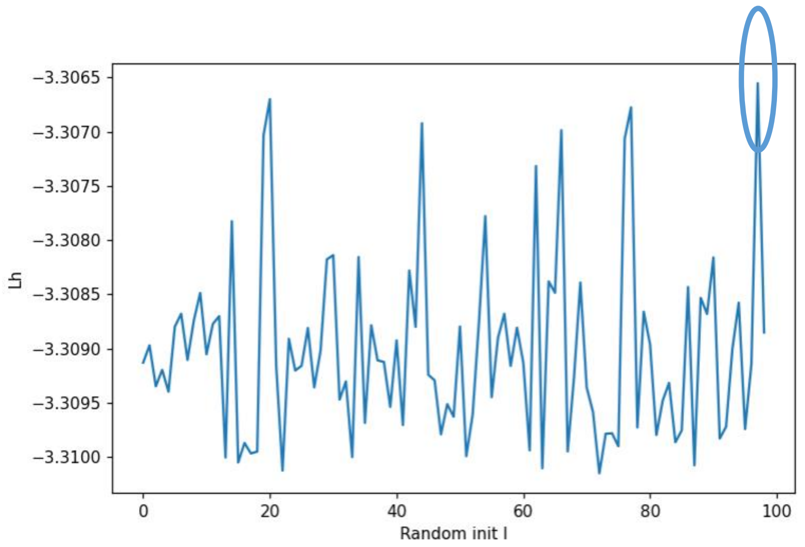
Condiciones iniciales

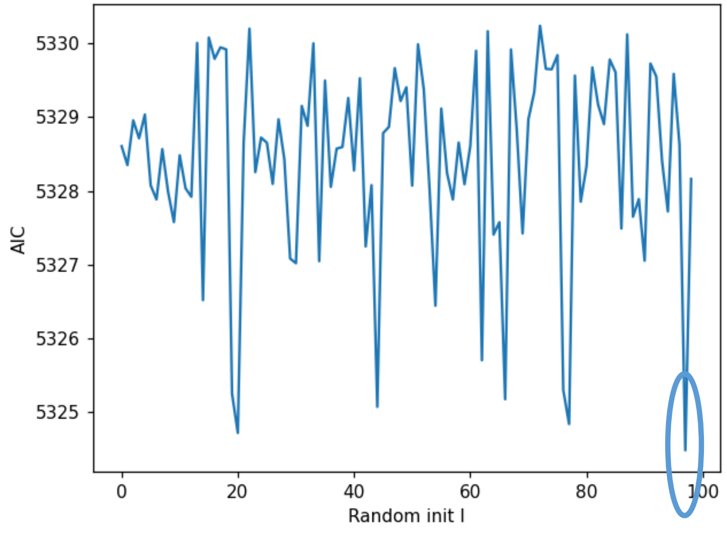


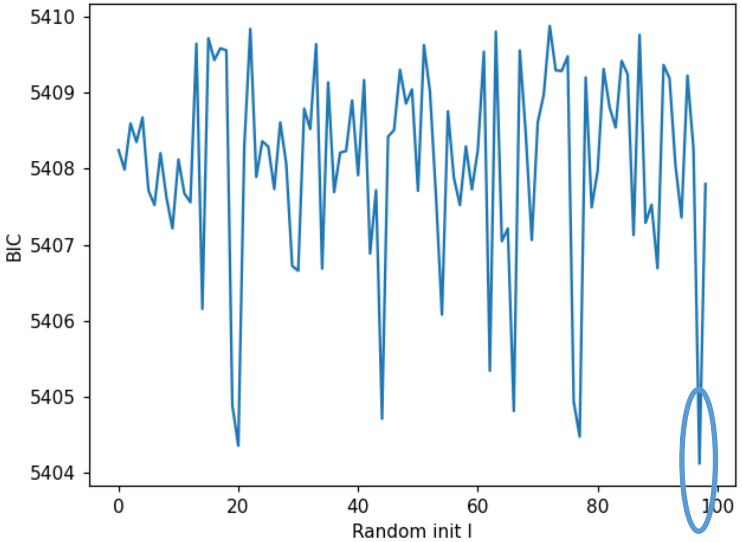
AIC, BIC and
Likelihood Score

K=3

100 random init

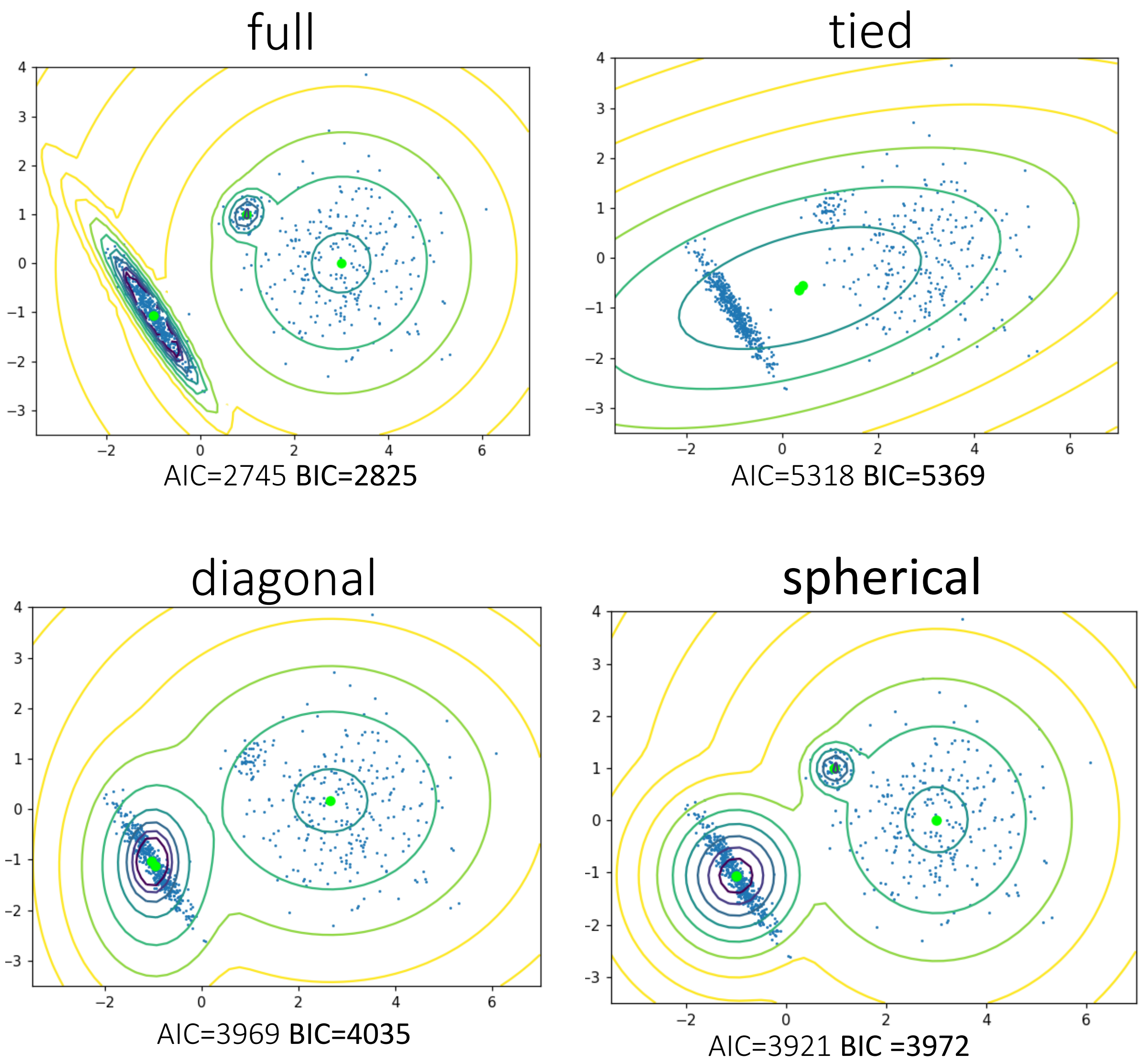






En este caso, la iteración i = 97; tiene las mejores condiciones iniciales

Matrices de covarianzas



- EII: equal volume, round shape (spherical covariance)
- VII: varying volume, round shape (spherical covariance)
- EEI: equal volume, equal shape, axis parallel orientation (diagonal covariance)
- VEI: varying volume, equal shape, axis parallel orientation (diagonal covariance)
- EVI: equal volume, varying shape, axis parallel orientation (diagonal covariance)
- VVI: varying volume, varying shape, equal orientation (diagonal covariance)
- EEE: equal volume, equal shape, equal orientation (ellipsoidal covariance)
- EEV: equal volume, equal shape, varying orientation (ellipsoidal covariance)
- VEV: varying volume, equal shape, varying orientation (ellipsoidal covariance)
- VVV: varying volume, varying shape, varying orientation (ellipsoidal covariance)

Clusters probabilísticos

Dr. Raimundo Sánchez

raimundo.sanchez@uai.cl

@raimun2