Advanced Institute for Artificial Intelligence

Aprendizado Não-Supervisionado

Advanced Institute for Artificial Intelligence

https://advancedinstitute.ai

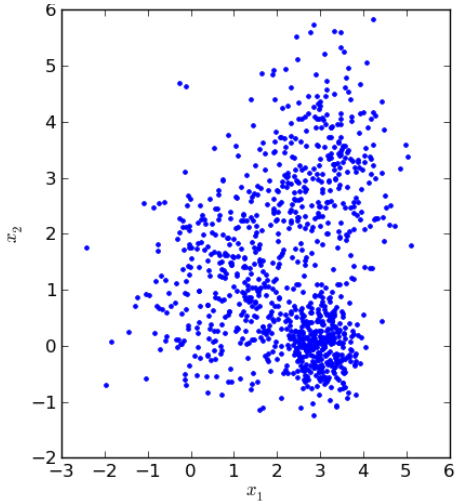
Aprendizado Não-Supervisionado (Clustering): Definição

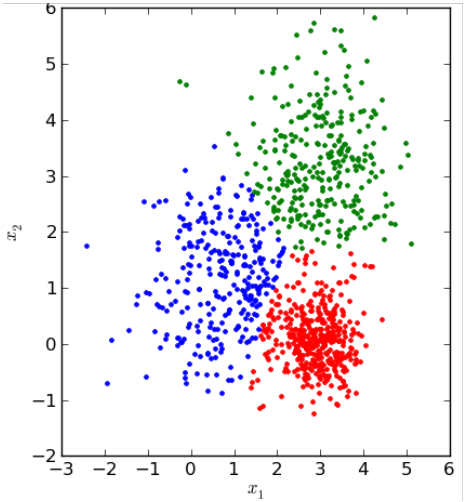
Agrupamento (Clustering): O que é?

- Em alguns domínios, gostaríamos de "classificar" exemplos em grupos semelhantes
- Porém, não possuímos rótulos de classes na base de treinamento

Exemplos

- **Agrupar** consumidores de uma plataforma web
- **Agrupar** documentos de acordo com seu tipo, tópico, etc
- **Identificar** anomalias e fraudes


6
5
4
3
2
1
0
−1
−2
x_2
−3
−2
−1
0
1
2
3
4
5
6
x_1


6
5
4
3
2
1
0
−1
−2
x_2
−3 −2 −1 0 1 2 3 4 5 6
x_1

Conceitos (Everitt, 1974)

- Um grupo (*cluster*) é um conjunto de entidades semelhantes entre si, enquanto entidades pertencentes a grupos diferentes não são semelhantes entre si
- A **distância** (diferença) entre quaisquer 2 instâncias em um mesmo grupo é menor que a distância entre qualquer dupla de instâncias pertencentes a grupos distintos

Mais Formalmente.....

Agrupamento em partições rígidas

- Agrupar objetos, designando instâncias $\boldsymbol{X}$ em grupos $\boldsymbol{C} = \{C_1, \ldots, C_k\}$, tal que:
- $C_1 \cup \cdots \cup C_k = \boldsymbol{X}$: Todos as instâncias devem ser designadas a um grupo
- $C_i \neq$: Todos os grupos possuem pelo menos 1 instância
- $C_i \cap C_j =$: Cada instância possue apenas um rótulo
- **Objetivo:** Encontrar mapeamento f:

$$f : \boldsymbol{X} \to \{1, \ldots, k\},$$

atribuindo cada instância a um grupo

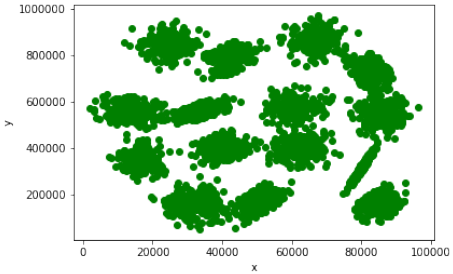
- Enumerar e avaliar todas as partições possíveis é inviável para qualquer aplicação não-trivial

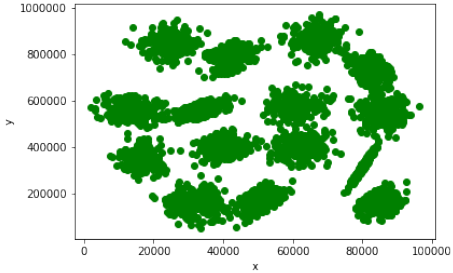
Como definir que instâncias são similares o suficiente para serem colocadas no mesmo grupo?

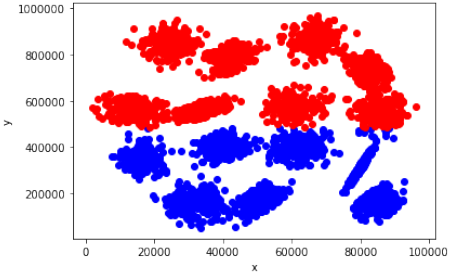
Diferenças podem ser subjetivas

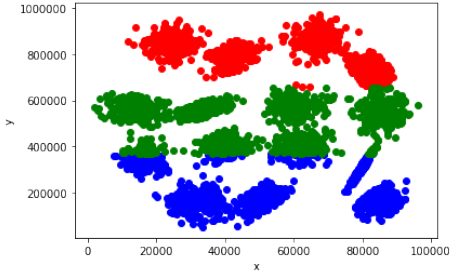
- Assim como na *classificação*, algoritmos de agrupamento se baseiam nos atributos para tentar formar agrupamentos bem definidos

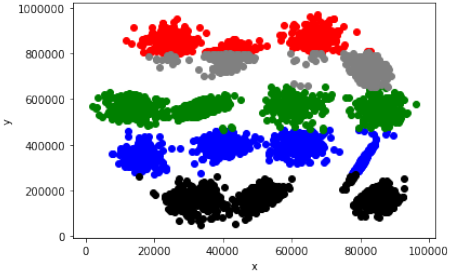
Avaliar qual seria o melhor agrupamento pode ser desafiador




1000000
800000
600000
400000
200000
y
0
20000
40000
60000
80000
100000
x


1000000
800000
600000
400000
200000
y
0
20000
40000
60000
80000
100000
x


1000000
800000
600000
400000
200000
y
0
20000
40000
60000
80000
100000
x


1000000
800000
600000
400000
200000
0
y
0
20000
40000
60000
80000
100000
x

- Assim como para classificação, cada algoritmo de agrupamento segue alguma estratégia e suposições para definir um agrupamento adequado

Advanced
Institute for
Artificial
Intelligence

Algoritmo K-Médias (*K-Means*)

K-Means

Separa os dados de treinamento em K grupos

Algoritmo amplamente utilizado na prática

- Simples
- Fácil de se interpretar
- É eficiente computacionalmente

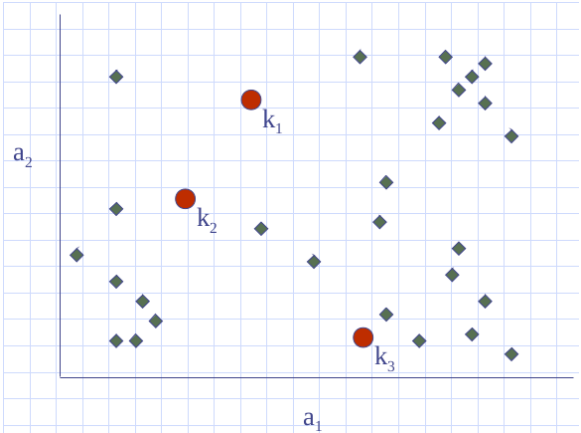
Algoritmo K-Means

Partindo de um número conhecido de grupos K:

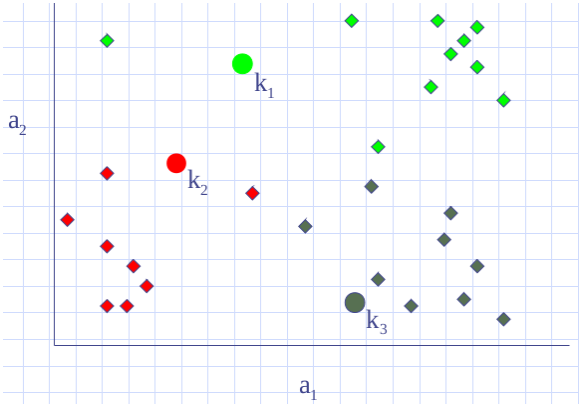
1. Selecione K *centróides* (instâncias que representarão cada grupo)
2. Atribua cada instância ao cluster que tiver o centróide mais próximo
3. Recomputa o centróide de acordo com a nova separação
4. Repita os 2 passos acima até a convergência

Exemplo Ilustrativo

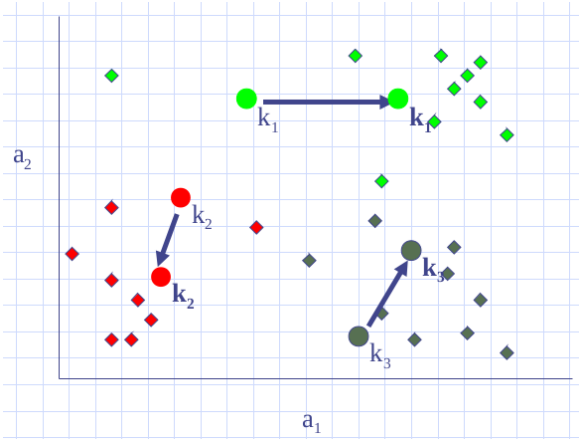
Com $K = 3$, iniciamos definindo 3 centróides



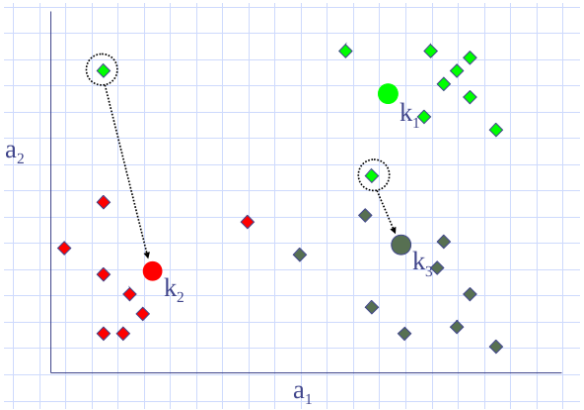
Rotular todos os exemplos de acordo com o centróide mais próximo



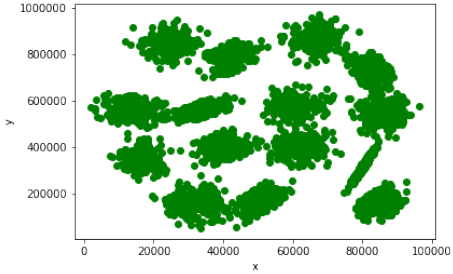
Atualizar todos os centróides

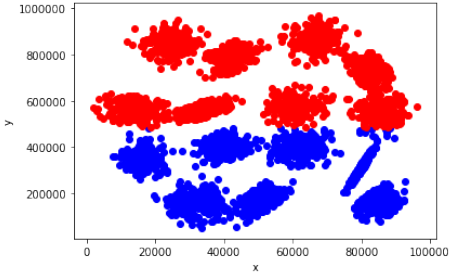


Atualizar grupos, e repetir até convergência



Exemplo com $K = 2$





Parâmetros

1. K
2. Métrica de distância
3. Inicialização dos centróides

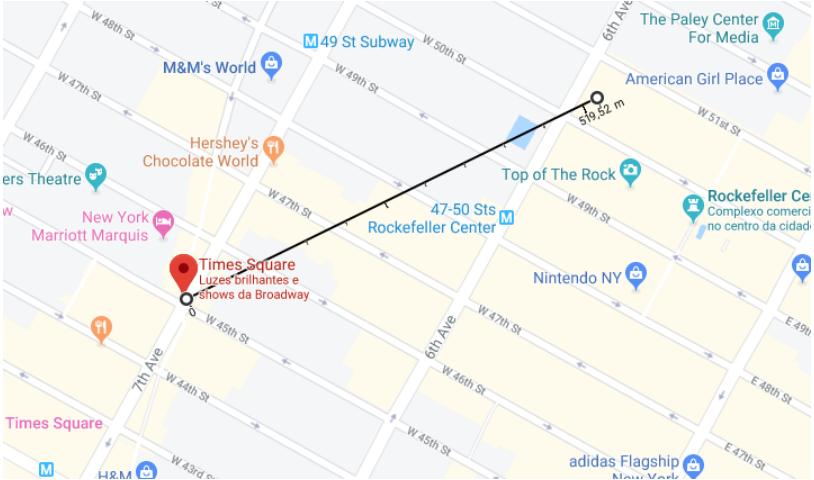
Distância Euclidiana

$$d_{euc}(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{m=1}^{M} (a_m - b_m)^2} \tag{1}$$

- Distância "comum" entre dois pontos

Exemplo Distância Euclidiana

Exemplo Distância Euclidiana



Distância Manhattan

$$d_{man}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{m=1}^{M} |a_m - b_m| \tag{2}$$

- Na analogia com distâncias espaciais, seria como "percorrer as ruas"

Exemplo Distância Manhattan

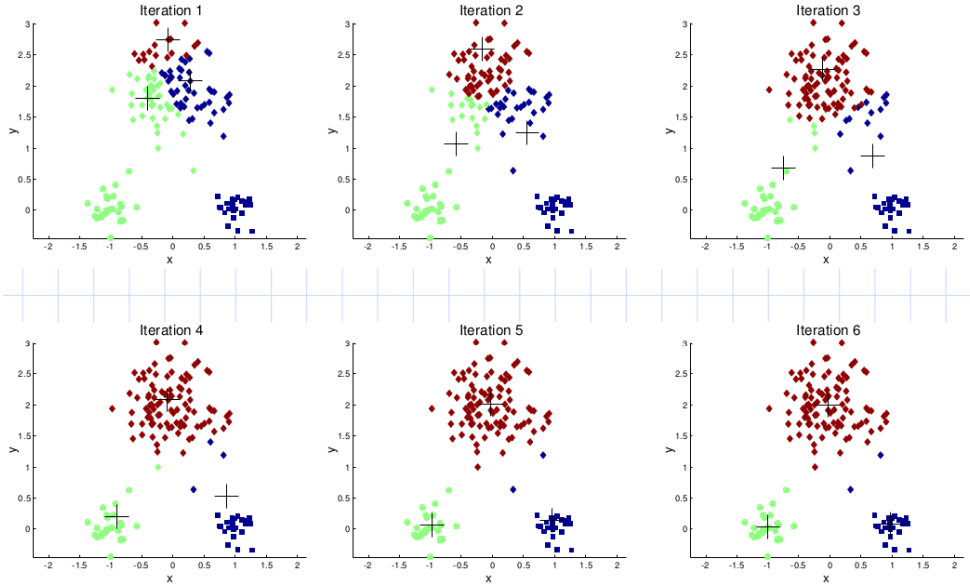
Comparação entre as distâncias

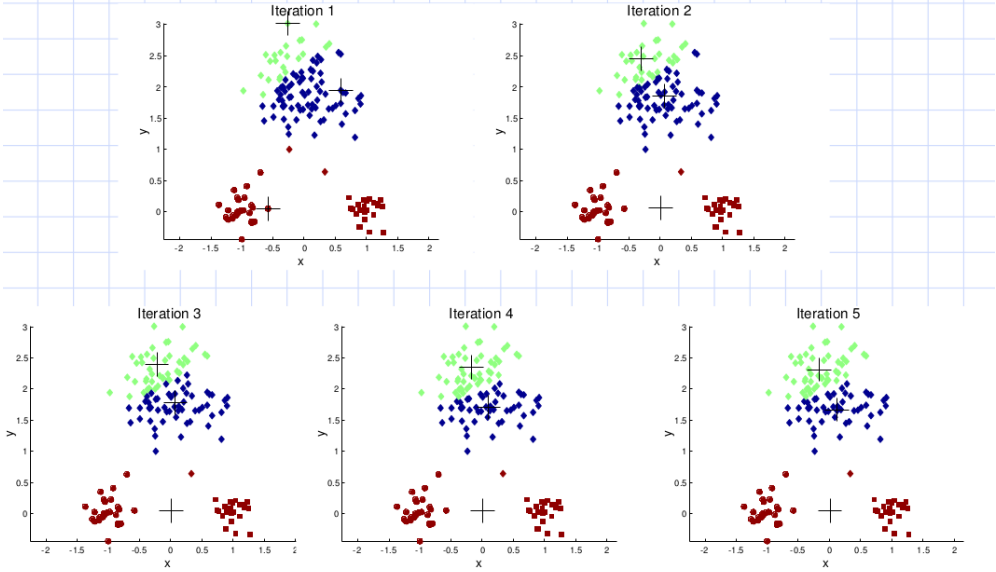
Euclidean

Manhattan

Inicialização dos centróides

- A inicialização dos centróides afeta no resultado final a na velocidade de convergência
- Portanto, a estratégia de inicialização afeta na efetividade do algoritmo

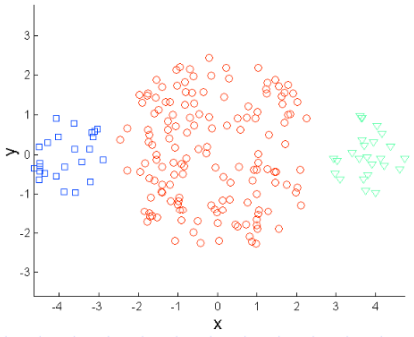

Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
x
y


Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
x
y

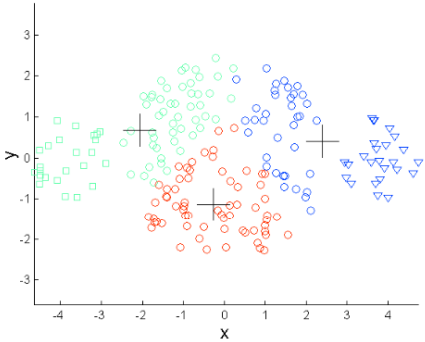
Como inicializar os centróides?

- Executar múltiplas vezes e selecionar os melhores clusters
- Seleção informada de centróides distantes entre si

Grupos de tamanhos diferentes

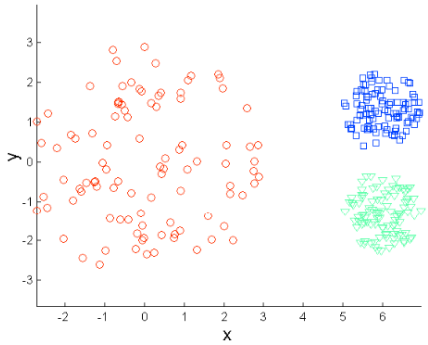


(a) Dados Gerados

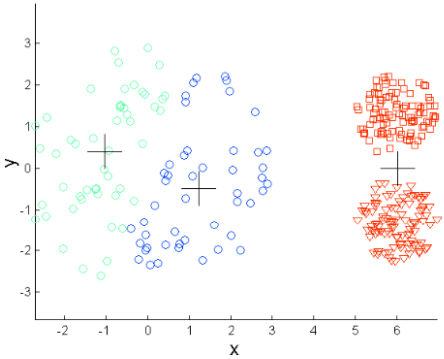


(b) $K = 3$

Grupos de densidades diferentes

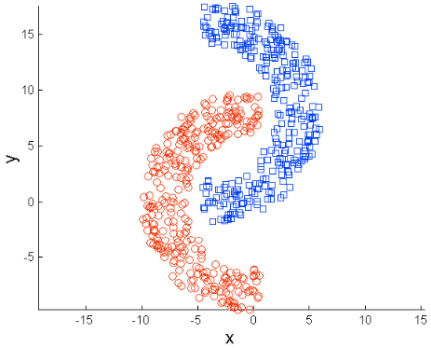


(c) Dados Gerados

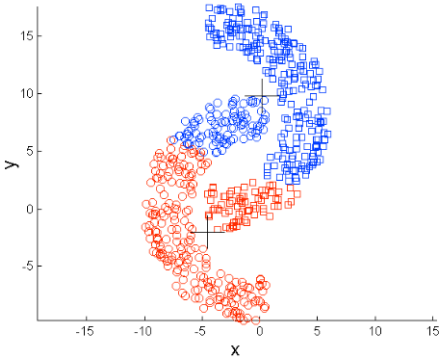


(d) $K = 3$

Formas não globulares

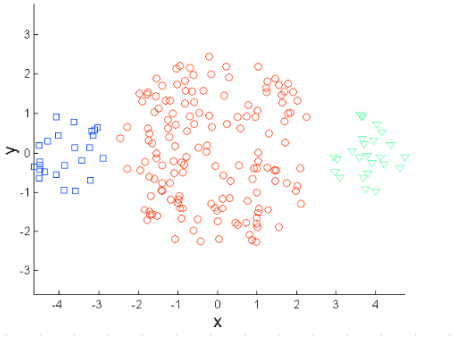


(e) Dados Gerados

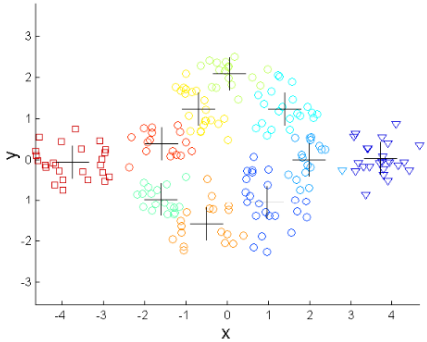


(f) $K = 2$

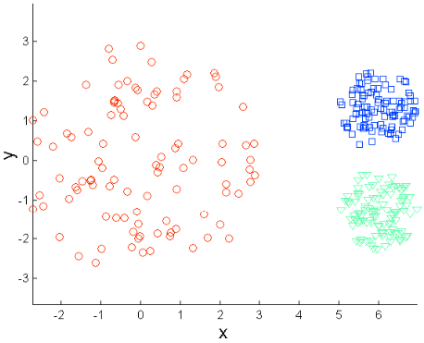
Aliviando esses problemas aumentando o parâmetro K



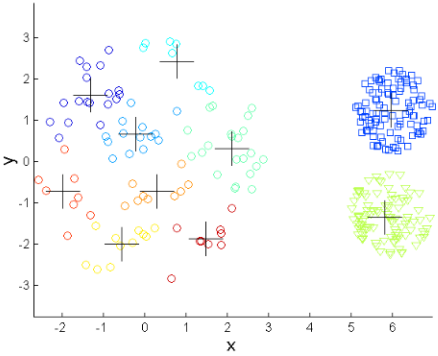
(g) Dados Gerados



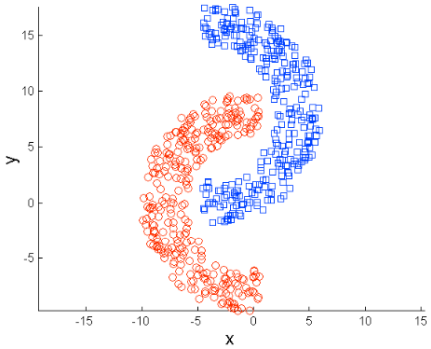
(h) Mais grupos



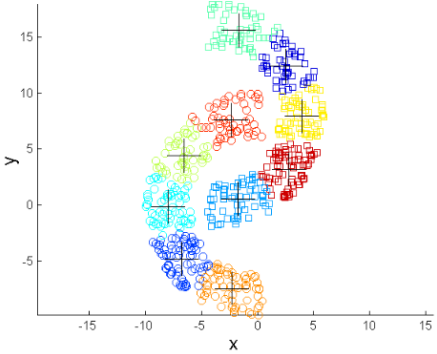
(i) Dados Gerados



(j) Mais grupos



(k) Dados Gerados



(l) Mais grupos

Pré e Pós-processamentos para o KMeans

Pré-processamento

- Normalização
- Eliminação de outliers

Pós-processamento

- Eliminar pequenos clusters
- Dividir grupos com instâncias muito distantes entre si
- Unir grupos com instâncias muito próximas

Advanced Institute for Artificial Intelligence

DBSCAN

DBSCAN

- O KMeans é um algoritmo baseado em protótipos
- Algoritmos baseados em densidade definem clusteres separando regiões com alta concentração de instâncias
- O *DBSCAN* é um dos algoritmos mais utilizados dentre os baseados em densidade

Separa as instâncias da base em 3 tipos

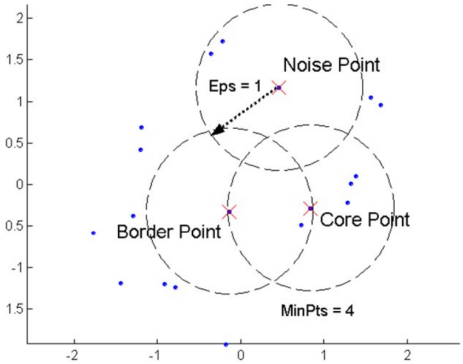
- **Core** - Pontos que estão no centro de uma concentração de instâncias (semelhante à ideia de centróide)
- **Border** - Pontos que fazem parte de um grupo mas não estão no centro dele, formando a "borda".
- **Noise** - Pontos isolados das outras instâncias

Algoritmo

1. Percorre e base de treinamento e rotula cada exemplo como **core**, **border**, ou **noise**.
2. Elimina todos os exemplos rotulados como **noise**
3. Insere uma aresta entre cada par de exemplos **core** próximos uns dos outros
4. Cada componente conexo resulta em um cluster
5. Cada **border** é atribuido ao **core** correspondente

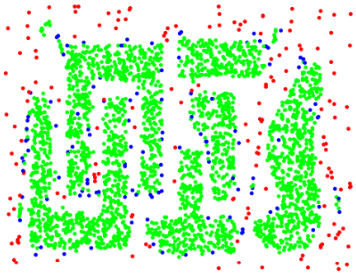
Como exatamente rotular cada instância? DBSCAN possui 2 parâmetros: eps e $minSamples$

- Todos as instâncias que possuem no mínimo $minSamples$ pontos a uma distância máxima de eps (vizinhança) são **core**
- Se um ponto possui menos que $minSamples$ vizinhos mas está na vizinhança de algum ponto core, ele é **border**
- Todos os pontos não classificados como core ou border são **noise**


Noise Point
Eps = 1
Border Point
Core Point
MinPts = 4
2
1.5
1
0.5
0
0.5
-1
1.5
-2
-1
0
1
2

Exemplo definição de pontos

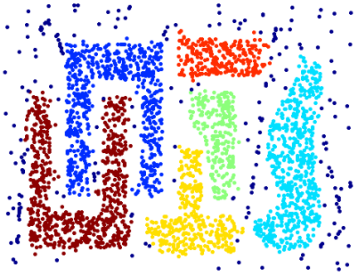
Original



Pontos: core, border e noise

Exemplo grupos

Original



Clusters

Vantagens do DBSCAN

- Resistente a ruído
- Consegue lidar com grupos de diferentes tamanhos e formas

Desvantagens do DBSCAN

- Não consegue capturar grupos com diferentes densidades