DATA MINING & MACHINE LEARNING (I)

Thiago Marzagão

limitações do k-means

- Requer que nós mesmos decidamos k.

limitações do k-means

- Requer que nós mesmos decidamos k.
- Força *toda* amostra a pertencer a um cluster.

limitações do k-means

- Requer que nós mesmos decidamos k.
- Força *toda* amostra a pertencer a um cluster.
- Não funciona bem com clusters de tamanhos muito diferentes

limitações do k-means

- Requer que nós mesmos decidamos k.
- Força *toda* amostra a pertencer a um cluster.
- Não funciona bem com clusters de tamanhos muito diferentes
- Não funciona bem com determinados tipos de clusters (alongados, irregulares, variância desigual).

outros algoritmos de clusterização

A comparison of the clustering algorithms in scikit-learn

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

http://scikit-learn.org/stable/modules/clustering.html

outros algoritmos de clusterização

A comparison of the clustering algorithms in scikit-learn

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

http://scikit-learn.org/stable/modules/clustering.html

DBSCAN

- DBSCAN tenta encontrar regiões de alta densidade.

DBSCAN

- DBSCAN tenta encontrar regiões de alta densidade.
- Alta densidade = muitas amostras, próximas umas das outras.

DBSCAN

- DBSCAN tenta encontrar regiões de alta densidade.
- Alta densidade = muitas amostras, próximas umas das outras.
- Encontra o k p/ nós: não precisamos saber o k a priori.

DBSCAN

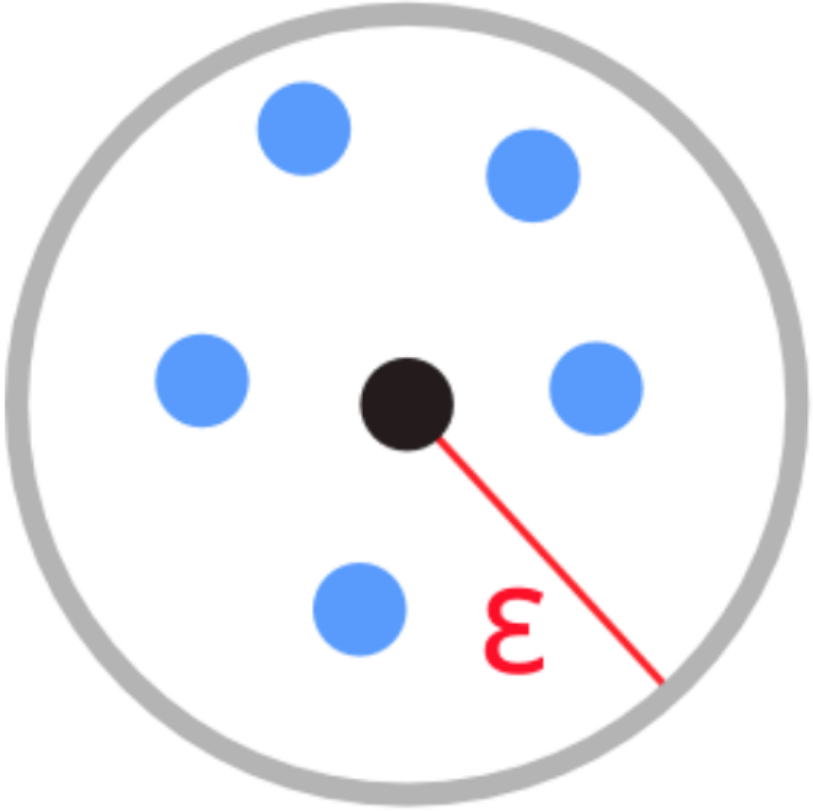
- DBSCAN tenta encontrar regiões de alta densidade.
- Alta densidade $=$ muitas amostras, próximas umas das outras.
- Encontra o k p/ nós: não precisamos saber o k a priori.
- Permite que amostras não pertençam a nenhum cluster (vai ser útil na próxima aula - detecção de anomalias).

DBSCAN

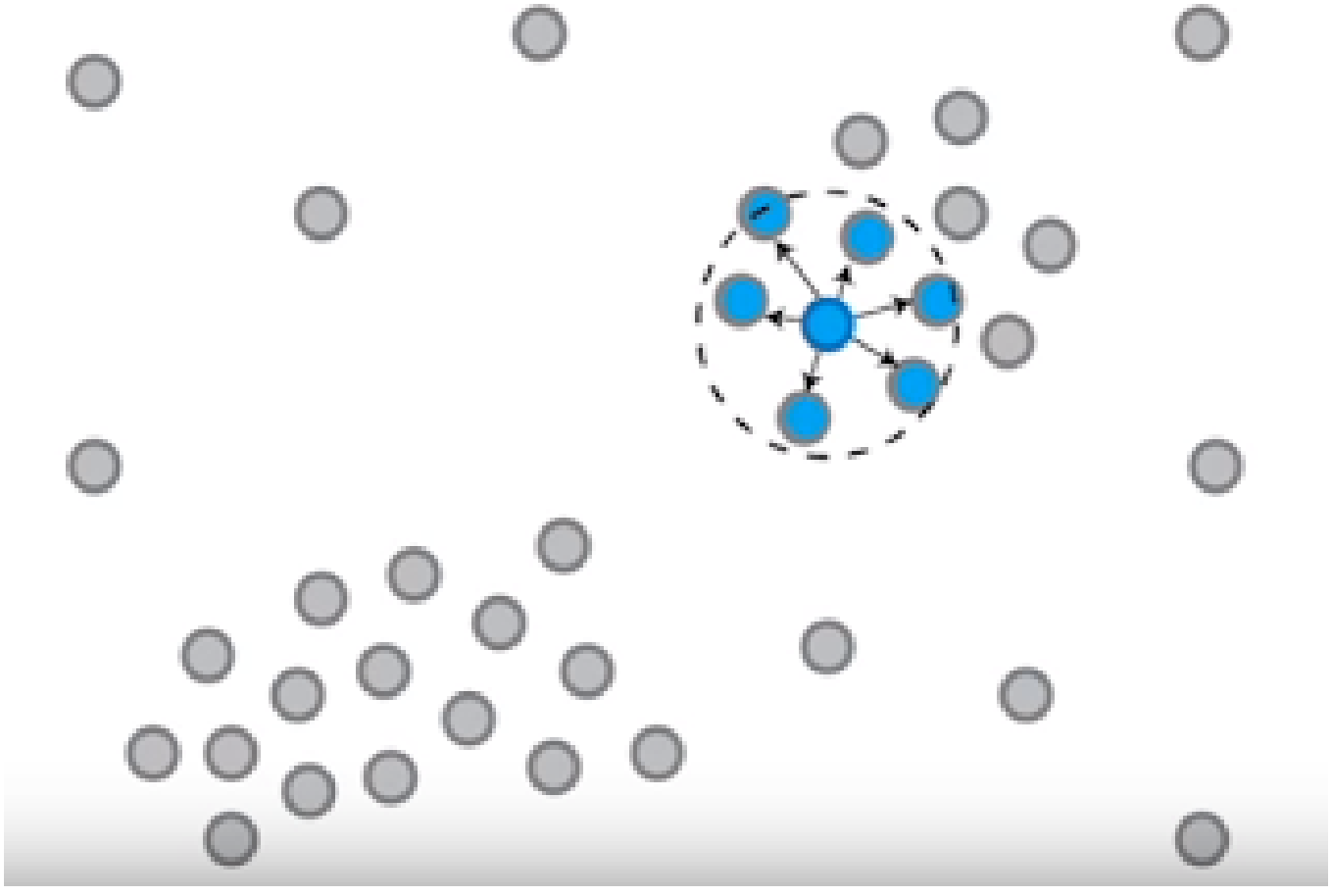
- DBSCAN tenta encontrar regiões de alta densidade.
- Alta densidade $=$ muitas amostras, próximas umas das outras.
- Encontra o k p/ nós: não precisamos saber o k a priori.
- Permite que amostras não pertençam a nenhum cluster (vai ser útil na próxima aula - detecção de anomalias).
- Não parte das mesmas premissas que o k-means (é mais flexível - permite clusters de tamanhos muito diferentes, por exemplo).

DBSCAN

- 2 parâmetros:
- ε
- $MinPts$

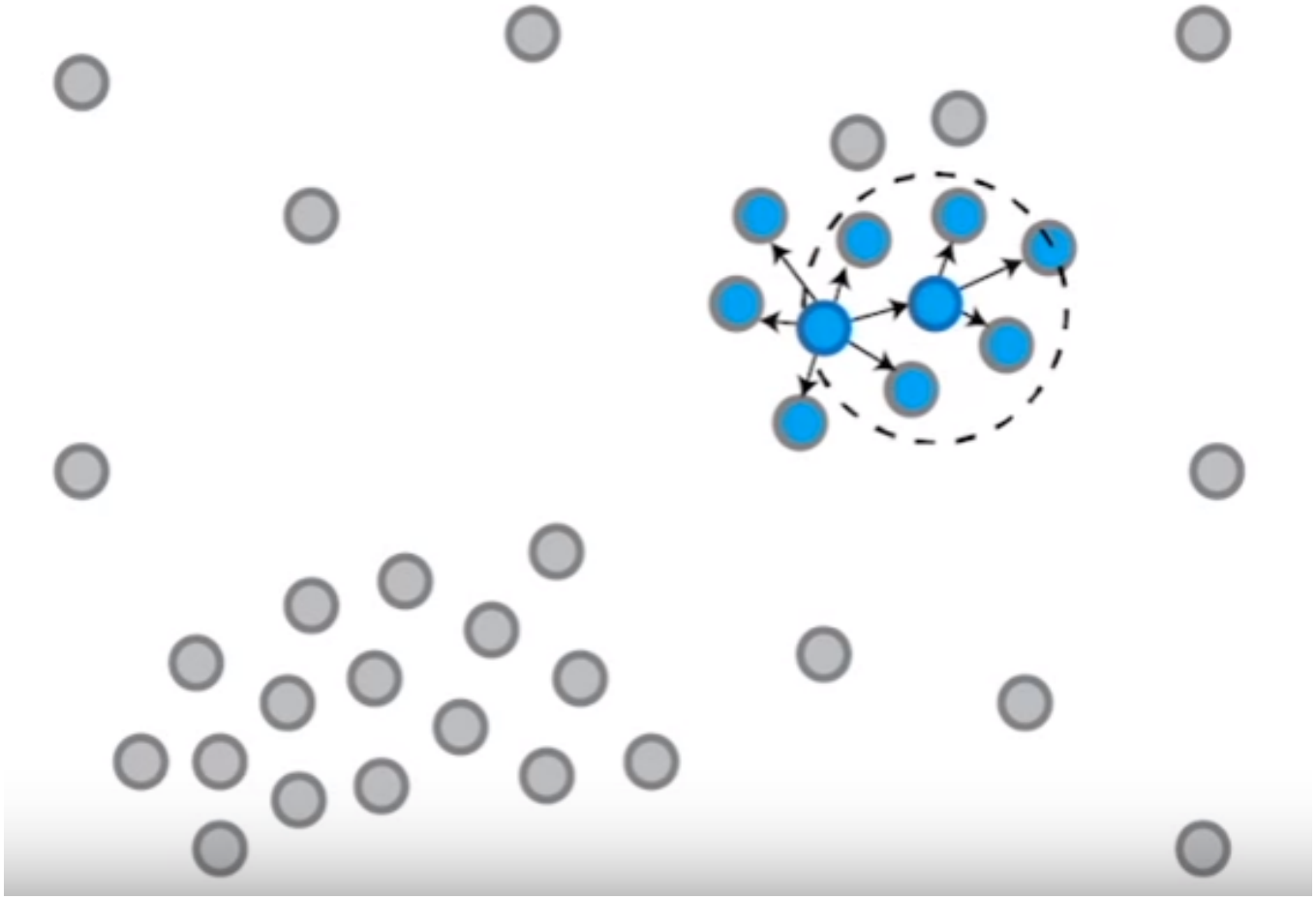


DBSCAN



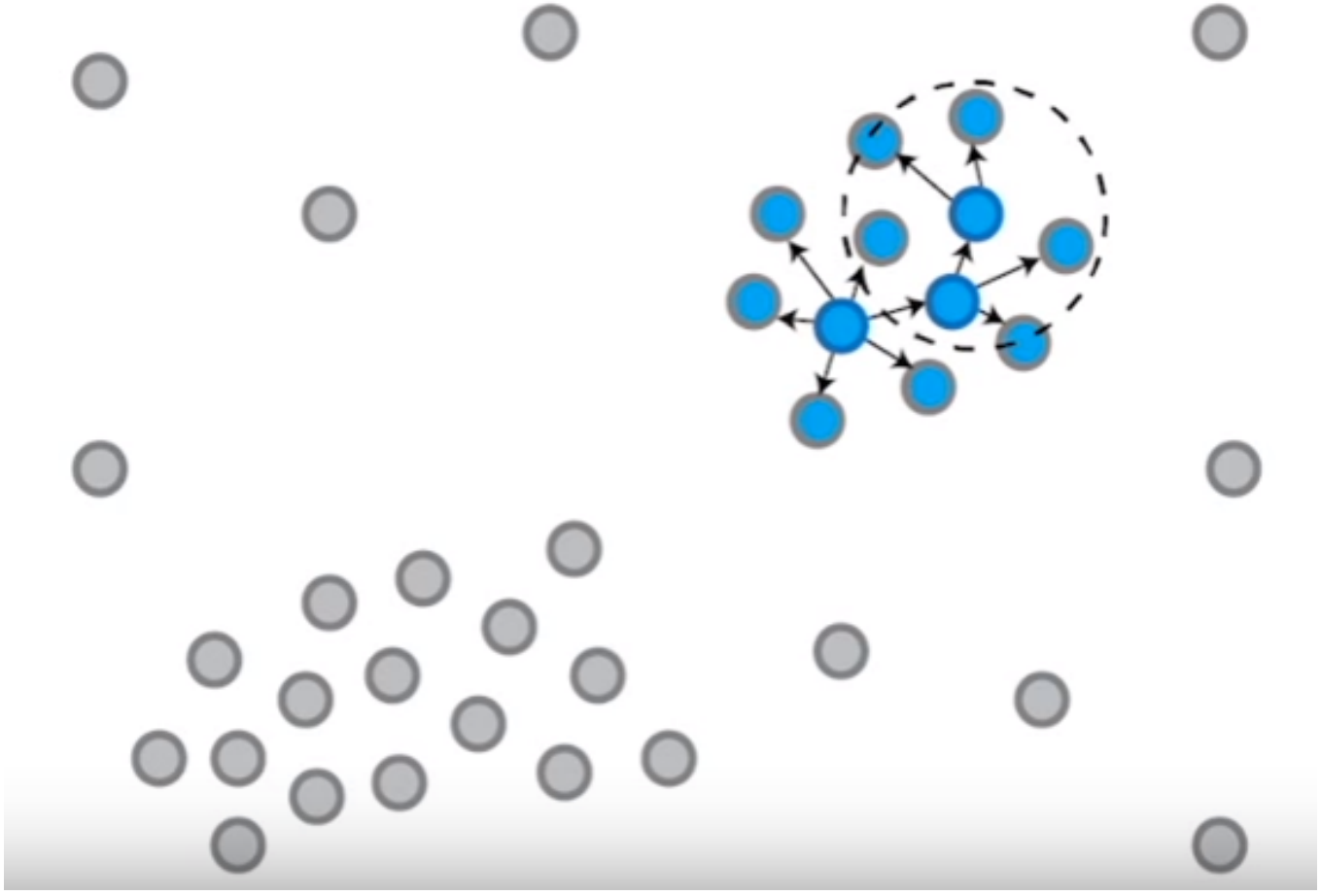
https://practice2code.blogspot.com.br/2017/07/dbscan-clustering-algorithm.html

DBSCAN



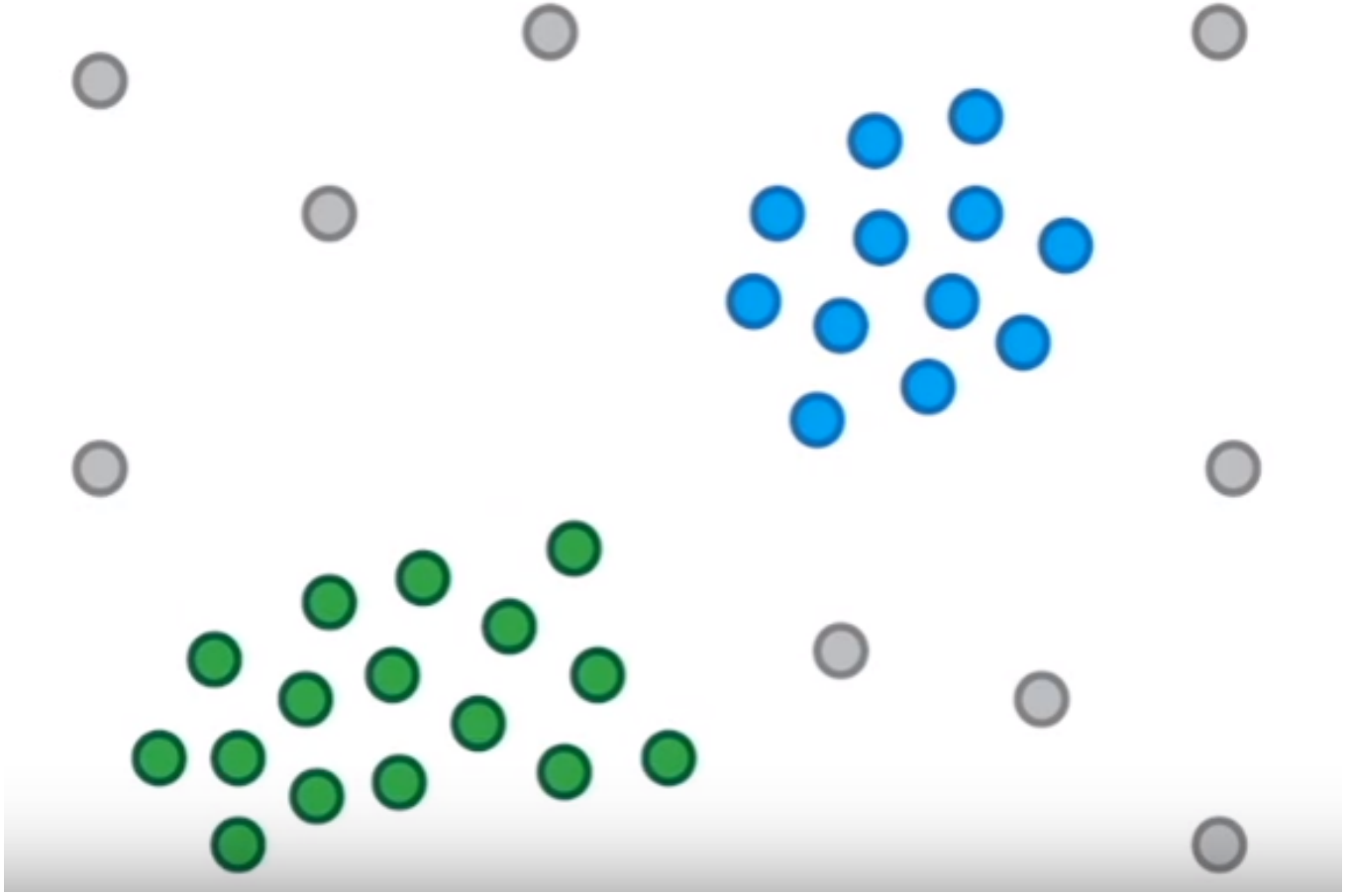
https://practice2code.blogspot.com.br/2017/07/dbscan-clustering-algorithm.html

DBSCAN



https://practice2code.blogspot.com.br/2017/07/dbscan-clustering-algorithm.html

DBSCAN



https://practice2code.blogspot.com.br/2017/07/dbscan-clustering-algorithm.html

DBSCAN

- 3 tipos de amostras:

DBSCAN

- 3 tipos de amostras:
- $core$

DBSCAN

- 3 tipos de amostras:
- $core$
  - Tem ao menos $MinPts$ amostras num raio de ε ao redor de si.

DBSCAN

- 3 tipos de amostras:
- $core$
  - Tem ao menos $MinPts$ amostras num raio de ε ao redor de si.
- $border$

DBSCAN

- 3 tipos de amostras:
- $core$
  - Tem ao menos $MinPts$ amostras num raio de ε ao redor de si.
- $border$
  - Não tem ao menos $MinPts$ amostras num raio de ε ao redor de si, mas tem uma amostra $core$ num raio de ε ao redor de si.

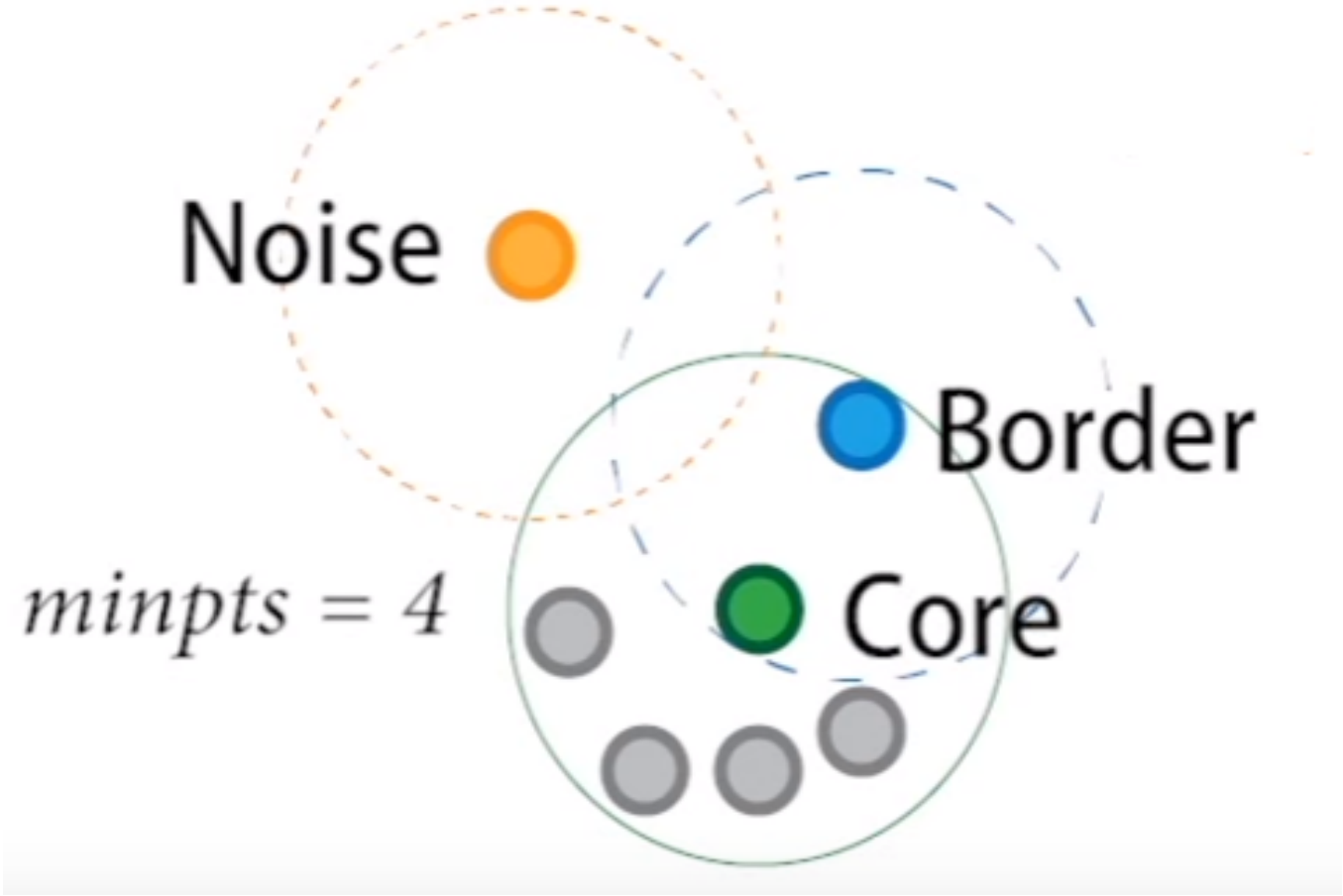
DBSCAN

- 3 tipos de amostras:
- $core$
  - Tem ao menos $MinPts$ amostras num raio de ε ao redor de si.
- $border$
  - Não tem ao menos $MinPts$ amostras num raio de ε ao redor de si, mas tem uma amostra $core$ num raio de ε ao redor de si.
- $noise$

DBSCAN

- 3 tipos de amostras:
- $core$
  - Tem ao menos $MinPts$ amostras num raio de ε ao redor de si.
- $border$
  - Não tem ao menos $MinPts$ amostras num raio de ε ao redor de si, mas tem uma amostra $core$ num raio de ε ao redor de si.
- $noise$
  - Todas as demais amostras.

DBSCAN



https://practice2code.blogspot.com.br/2017/07/dbscan-clustering-algorithm.html

DBSCAN

- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

DBSCAN

- Quanto maior o ε (p/ um dado $MinPts$), maiores os círculos e portanto mais amostras tendem a ser $core$.

DBSCAN

- Quanto maior o ε (p/ um dado $MinPts$), maiores os círculos e portanto mais amostras tendem a ser $core$.
- Quanto menor o $MinPts$ (p/ um dado ε), mais amostras tendem a cumprir o requisito de ter ao menos $MinPts$ num raio de ε ao seu redor e portanto mais amostras tendem a ser $core$.