GENERAL ASSEMBLY

DBSCAN

Joseph Nelson, Mark Mummert, DC DSI

AGENDA

- Review K-means
- Define DBSCAN
- Introduce Algorithm
- Identify strengths and weaknesses
- Implement in code

BY THE END OF THIS LESSON STUDENTS WILL BE ABLE TO...

- Explain the DBSCAN algorithm
- Identify its strengths and weaknesses
- Implement in sklearn

REVIEW

- What is clustering?

What is Clustering?

- Clustering is an unsupervised learning technique we employ to group 'similar' data points together
- With unsupervised learning there is no clear objective - no right answer. Just observations with features.

What is Clustering?

- K-means review:
- Volunteer to explain the algorithm

- What are some problems we observed with this yesterday?

- DBSCAN solves all of our problems!*

- *Not really

WHAT IS DBSCAN?

- Density-based Spatial Clustering of Applications with Noise
- For DBSCAN, clusters of **high density** are separated by clusters of **low density**.
- Most widely used and applicable clustering algorithm
- Takes predefined input and can discover clusters of any shape, not just the sphere-like clusters that k-means often computes.

WHAT IS DBSCAN?

- DBSCAN is a **density based clustering algorithm**, meaning that the algorithm finds clusters by seeking areas of the dataset that have a higher density of points than the rest of the dataset.

- Requires two input parameters:

epsilon: the minimum distance between two points for them to be considered a cluster

Minimum points: minimum number of points necessary to form a cluster

The original paper on DBSCAN:
https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

Algorithm

Suppose we have a dataset of n-dimensional data points.

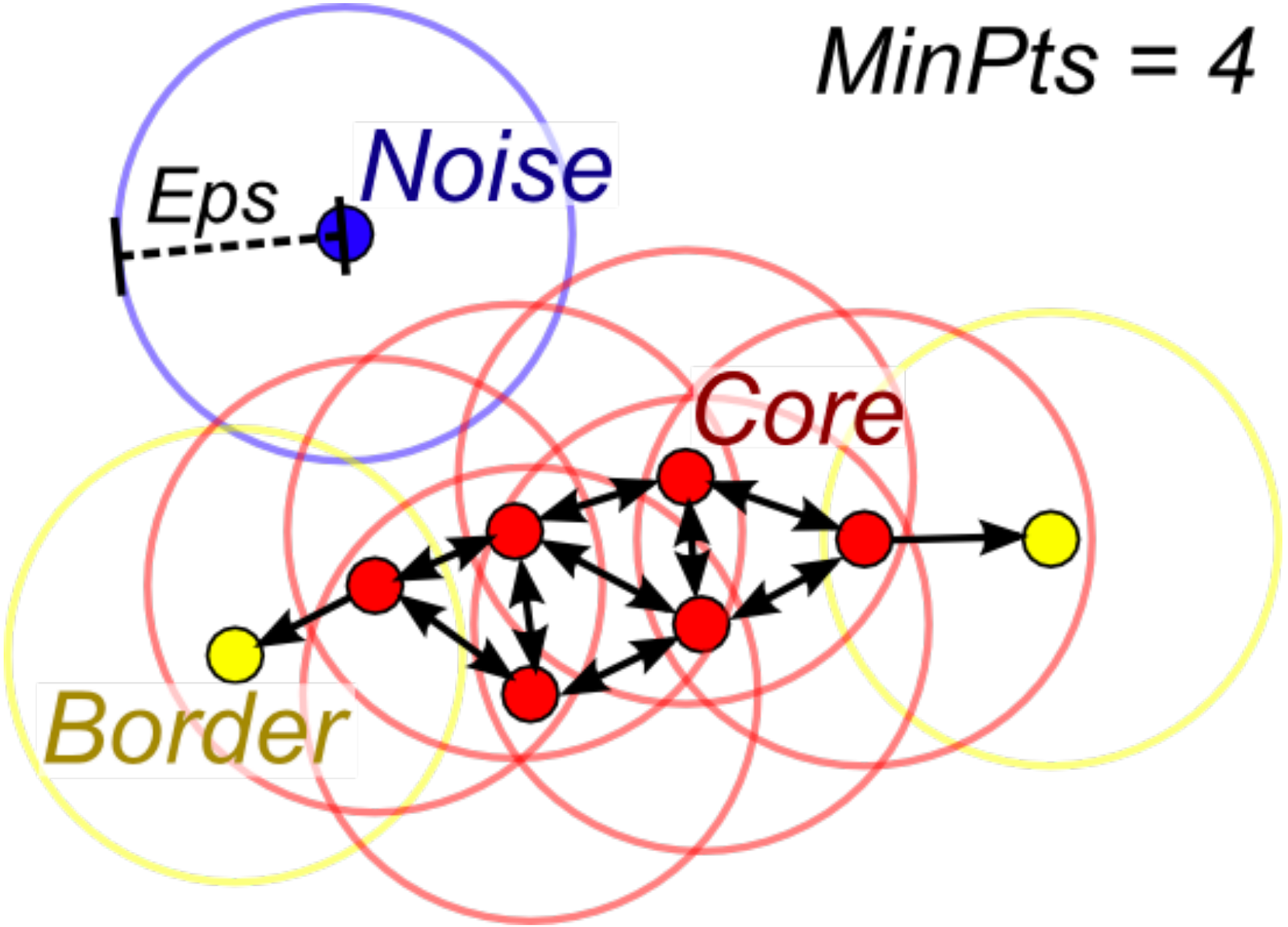
For each point in the dataset we make an n-dimensional sphere of radius *epsilon* around the point and count the number of data points within the sphere.

If the number of points within the sphere are more than *min_points* then we mark the center of the sphere to be belonging to a cluster (including the starting point).

We also mark the points inside the sphere to be belonging to the same cluster. We then recursively expand the cluster by applying the same criteria to the points inside the sphere, except the center.

If the number of points inside the sphere are less than *min_points*, we ignore it and proceed to the next point in the dataset.

VISUALIZED



Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

Let’s check out everyone’s favorite clustering visualizer….

- Algorithm visualization:
- http://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

- What is an optimal epsilon for the ’pimpled smiley’ distribution of points?
  Min number of points?

Comparing k-means and hierarchical

- Whereas k-means can be thought of as a "general" clustering approach, DBSCAN performs especially well with unevenly distributed, non-linear clusters.

- DBSCAN is density based, which means that it determines clusters based on the number of points in a certain area

- By choosing too few points for DBSCAN, i.e. less than two, we'll effectively get a straight line if we connect the points, just like linkage clustering.

Comparing k-means and hierarchical

- ADVANTAGES
- DISADVANTAGES

Comparing k-means and hierarchical

- ADVANTAGES –
- Clusters can be any size or shape
- No need to choose number of clusters
- Robust to outliers

- DISADVANTAGES –
- More parameters to tune
- Doesn’t work with clusters of varying density within the same data set
- Not every point is assigned to a cluster

- Things to be aware of when choosing parameters:

A low min points will build more clusters from noise

Epsilon will depend on how you scale your data

Some sources recommend a min points of 1 + the number of dimension of your data.

Coding Implementation

To the repo!