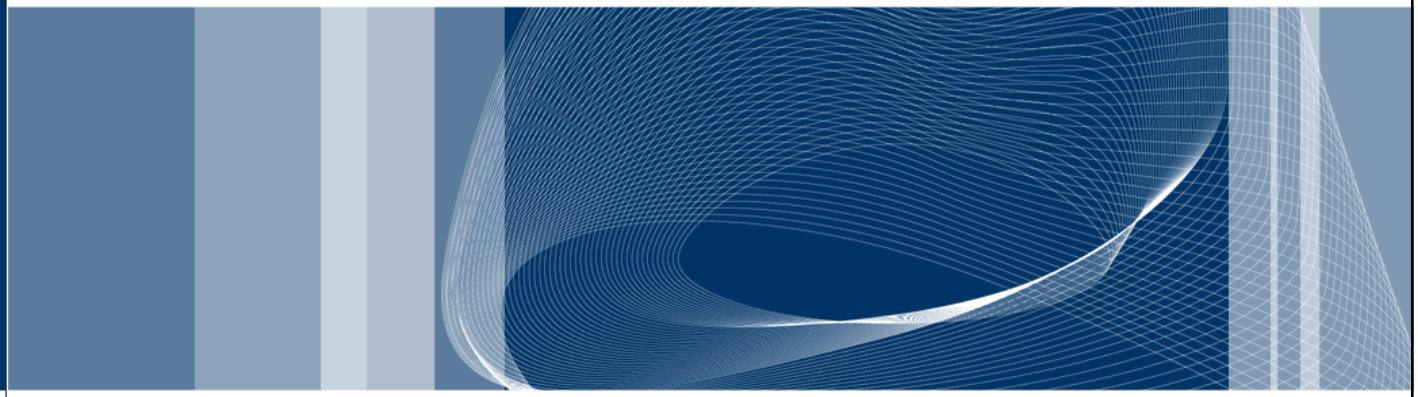
POLITECNICO DI MILANO



Density Based Clustering

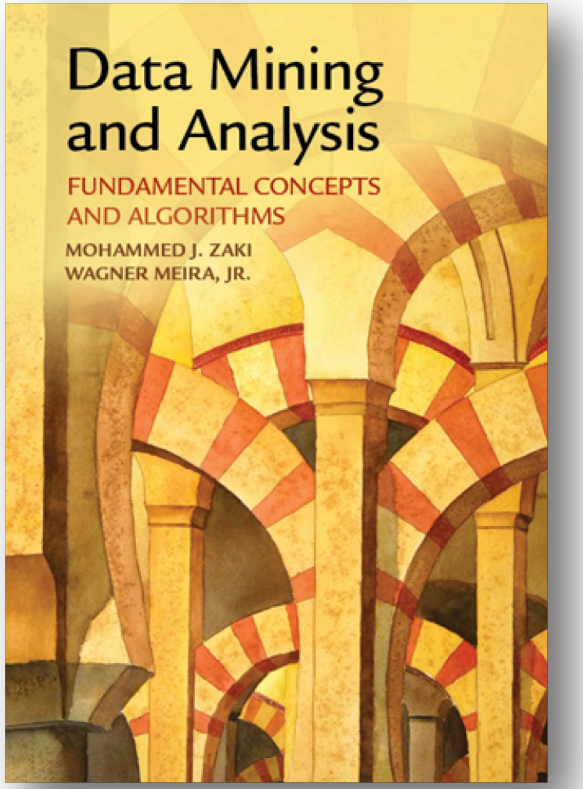
Data Science for Mobility

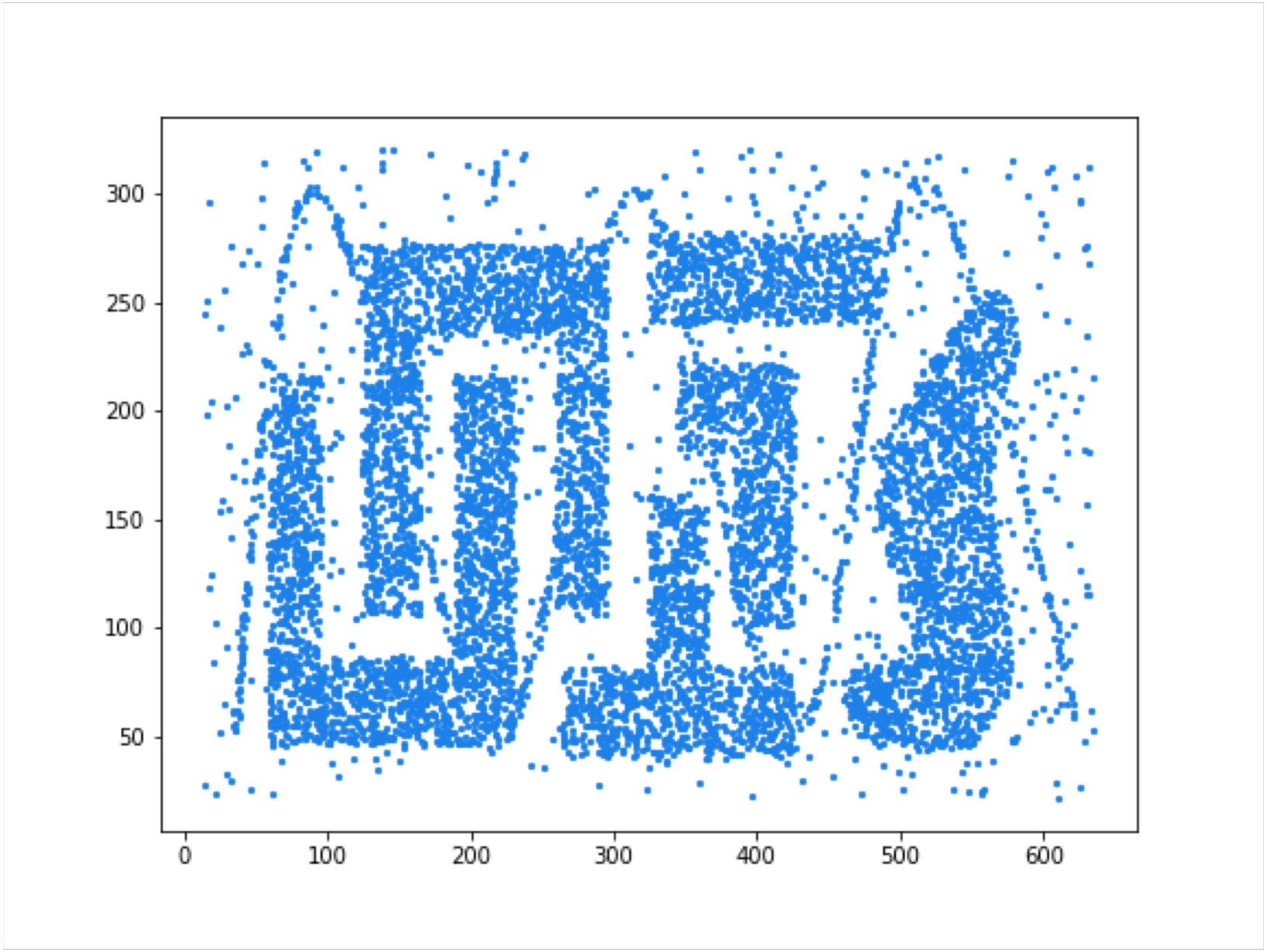
Prof. Pier Luca Lanzi

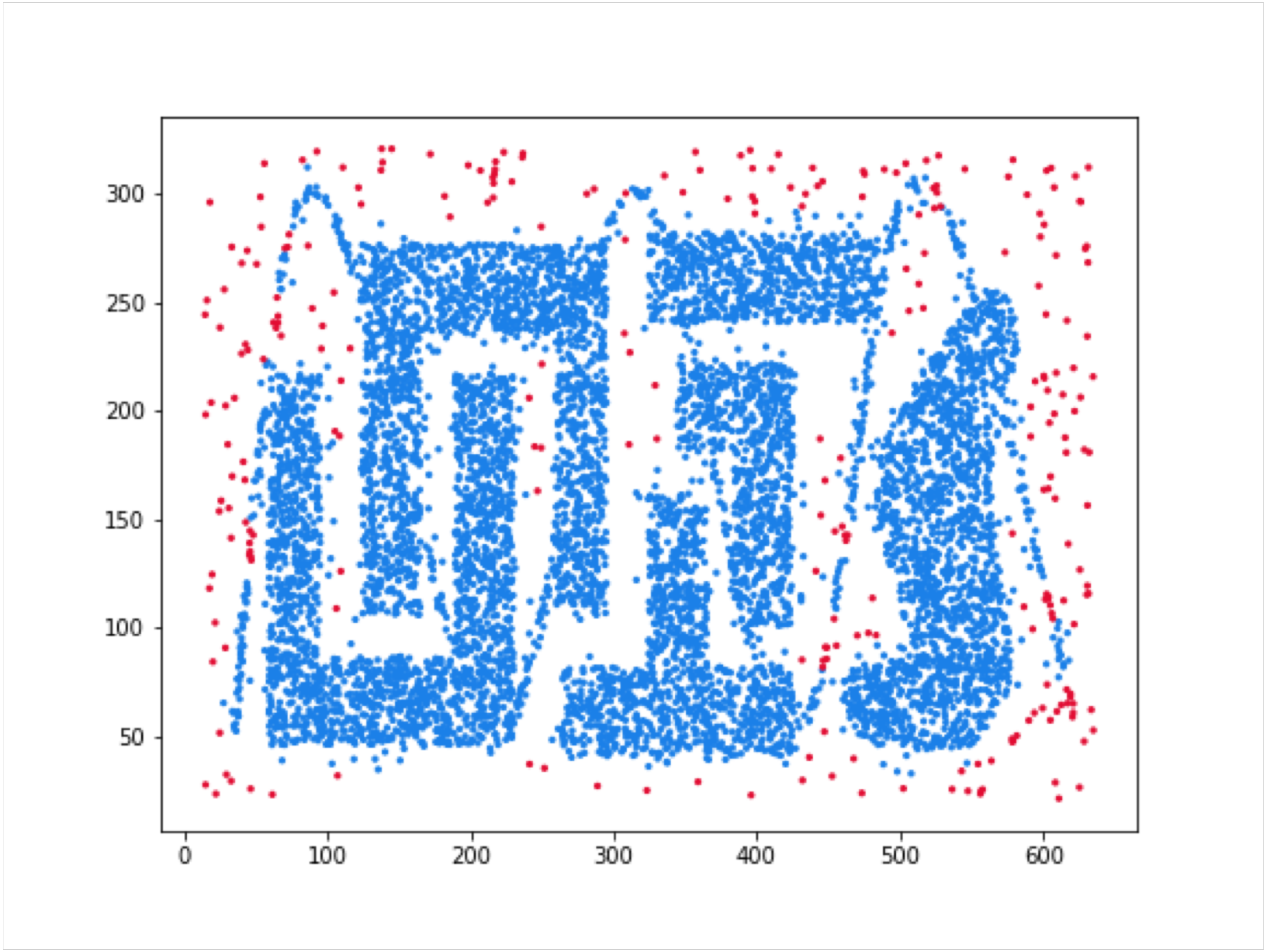
Readings

- "Data Mining and Analysis" by Zaki & Meira
  - Chapter 15

- http://www.dataminingbook.info




300
250
200
150
100
50
0
100
200
300
400
500
600


300
250
200
150
100
50
0
100
200
300
400
500
600

What is density-based clustering?

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

DBSCAN: Basic Concepts

- The neighborhood within a radius ε of a given object is called the ε-neighborhood of the object

$$N_\epsilon(x) = \{y | \delta(x, y) \leq \epsilon\}$$

- Core Object
  - If the ε-neighborhood of an object contains at least **minpts** objects, then the object is a core object

- Directly density reachable
  - An object x is directly density-reachable from object y if x is within the ε-neighborhood of y and y is a core object

DBSCAN: Basic Concepts

- Density Reachable
  - An object x is density-reachable from object y if there is a chain of objects $x_1, \ldots, x_n$ where $x_1=x$ and $x_n=y$ such that x_{i+1} is directly density reachable from x_i
- Density Connected
  - An object p is density-connected to q with respect to ε and MinPts if there is an object o such that both p and q are density reachable from o
- Density-Based Cluster
  - A density-based cluster is defined as a maximal set of density connected points.

DBSCAN: Basic Concepts

- Density corresponds to have at least minpts points within a specified radius ε
- A border point has fewer than minpt within ε, but is in the neighborhood of a core point
- A noise point is any point that is not a core point nor a border point

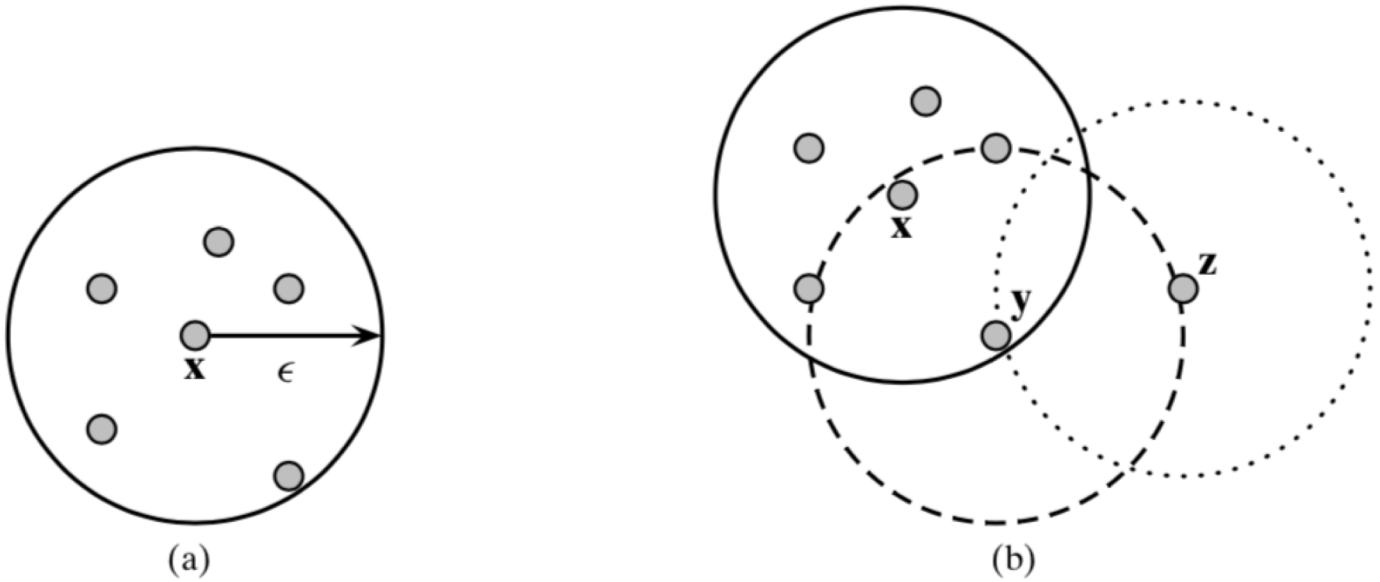


Figure 15.2. (a) Neighborhood of a point. (b) Core, border, and noise points.

Core, border and noise points when minpts is 6

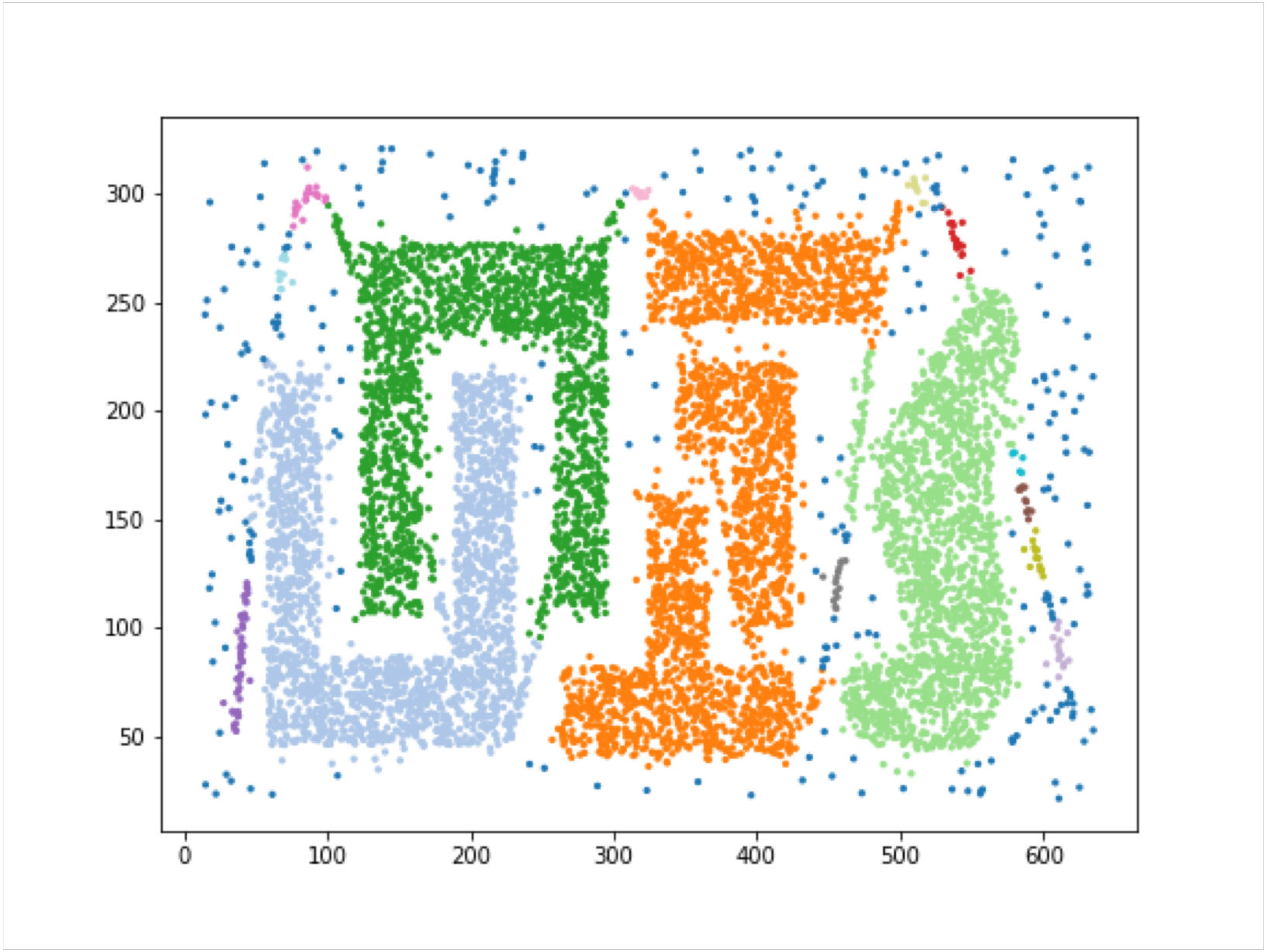
ALGORITHM 15.1. Density-based Clustering Algorithm

DBSCAN (D, ϵ, $minpts$):

```
1  Core ← ∅
2  foreach x_i ∈ D do // Find the core points
3      Compute N_ε(x_i)
4      id(x_i) ← ∅ // cluster id for x_i
5      if N_ε(x_i) ≥ minpts then Core ← Core ∪ {x_i}
6  k ← 0 // cluster id
7  foreach x_i ∈ Core, such that id(x_i) = ∅ do
8      k ← k + 1
9      id(x_i) ← k // assign x_i to cluster id k
10     DENSITYCONNECTED (x_i, k)
11 C ← {C_i}_{i=1}^k, where C_i ← {x ∈ D | id(x) = i}
12 Noise ← {x ∈ D | id(x) = ∅}
13 Border ← D \ {Core ∪ Noise}
14 return C, Core, Border, Noise
```

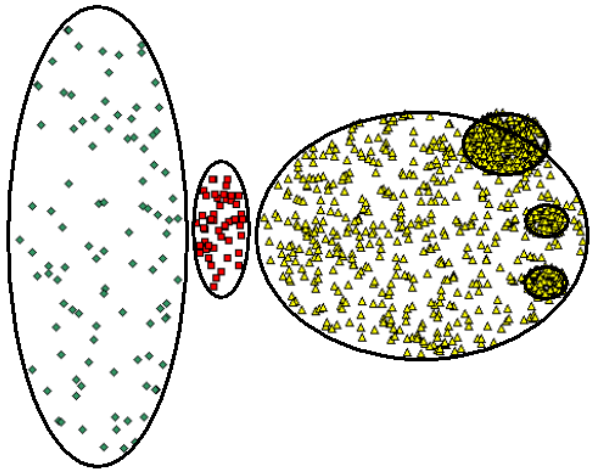
DENSITYCONNECTED (x, k):

```
15 foreach y ∈ N_ε(x) do
16     id(y) ← k // assign y to cluster id k
17     if y ∈ Core then DENSITYCONNECTED (y, k)
```

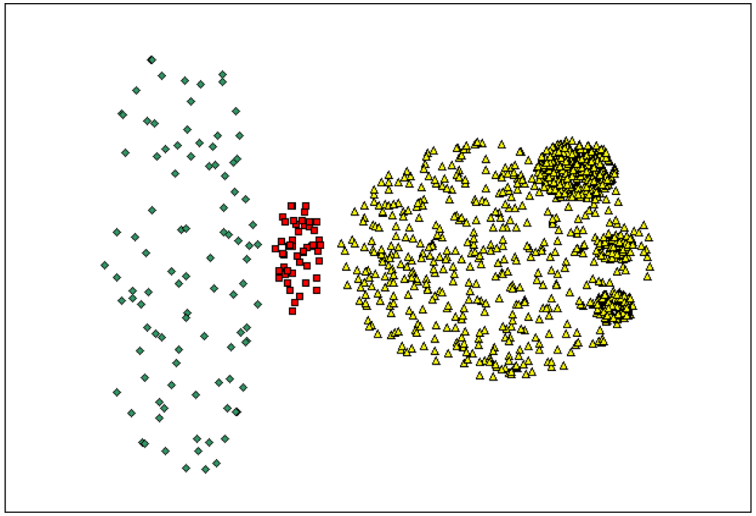

300
250
200
150
100
50
0
100
200
300
400
500
600

When DBSCAN May Fail?

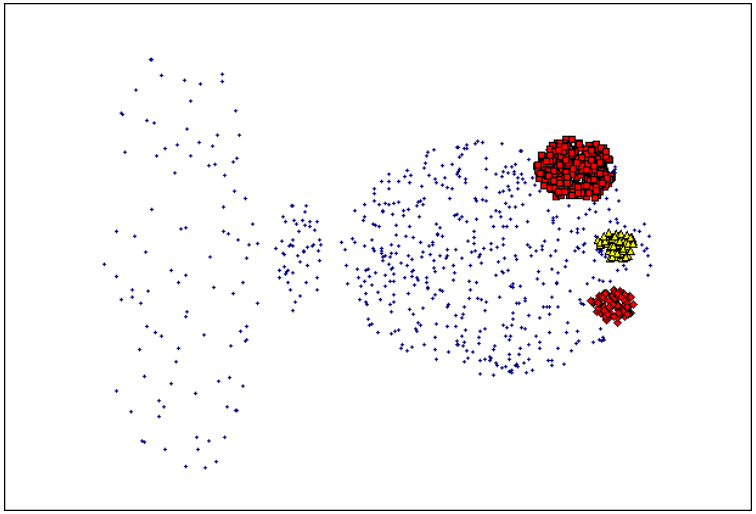
- Varying densities
- High-dimensional data



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

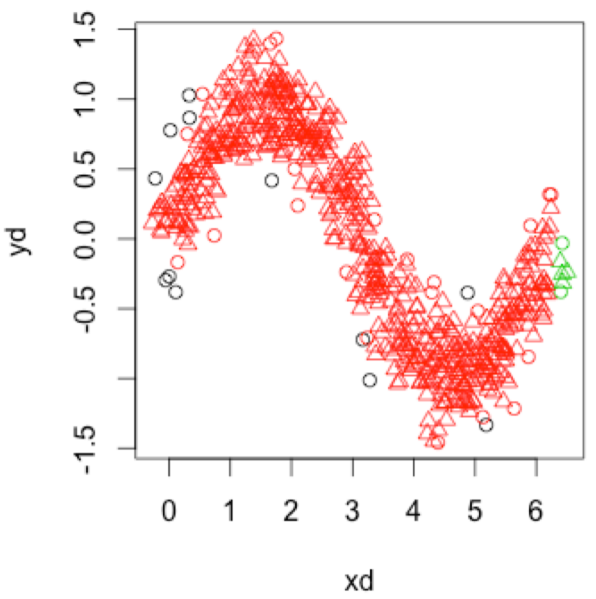
Examples using R

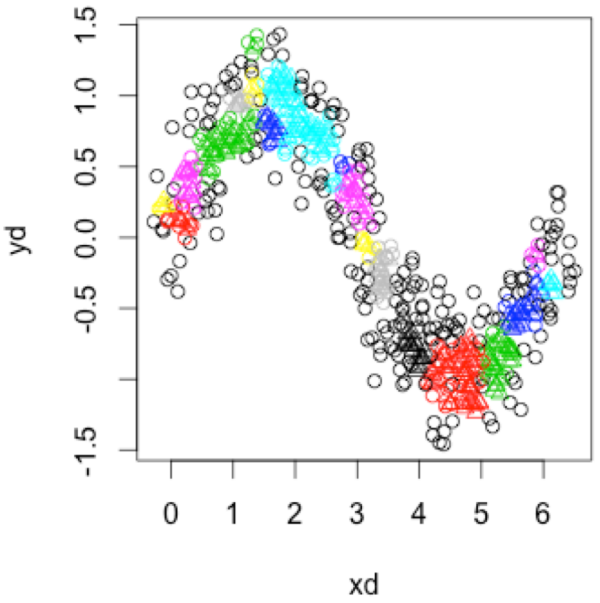
Density-Based Clustering in R


```
library(fpc)

set.seed(665544)
n <- 600
x <- cbind(runif(10, 0, 10)+rnorm(n, sd=0.2), runif(10, 0,
10)+rnorm(n,sd=0.2))

par(bg="grey40")
ds <- dbscan(x, 0.2, showplot=1)
```

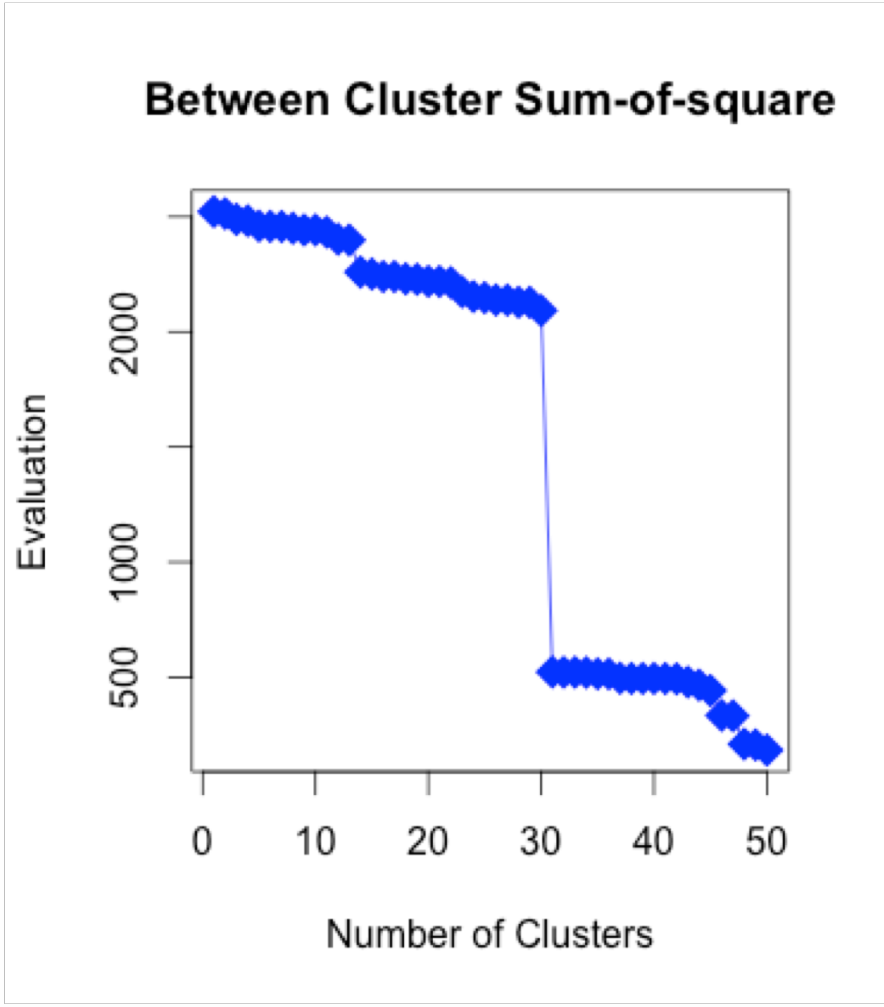
Density-Based Clustering in R

```
library(fpc)

set.seed(665544)

x <- seq(0,6.28,0.1)
y <- sin(x)

xd <- x+rnorm(630,sd=0.2)
yd <- y+rnorm(630,sd=0.2)
plot(xd,yd)

par(bg="grey40")
d <- cbind(xd,yd)

# this works nicely since the epsilon is
# the same size of the standard deviation (0.2)
# used to generate the data
ds <- dbscan(d, 0.2, showplot=1)

# this does not work so nicely
ds <- dbscan(d, 0.1, showplot=1)
```

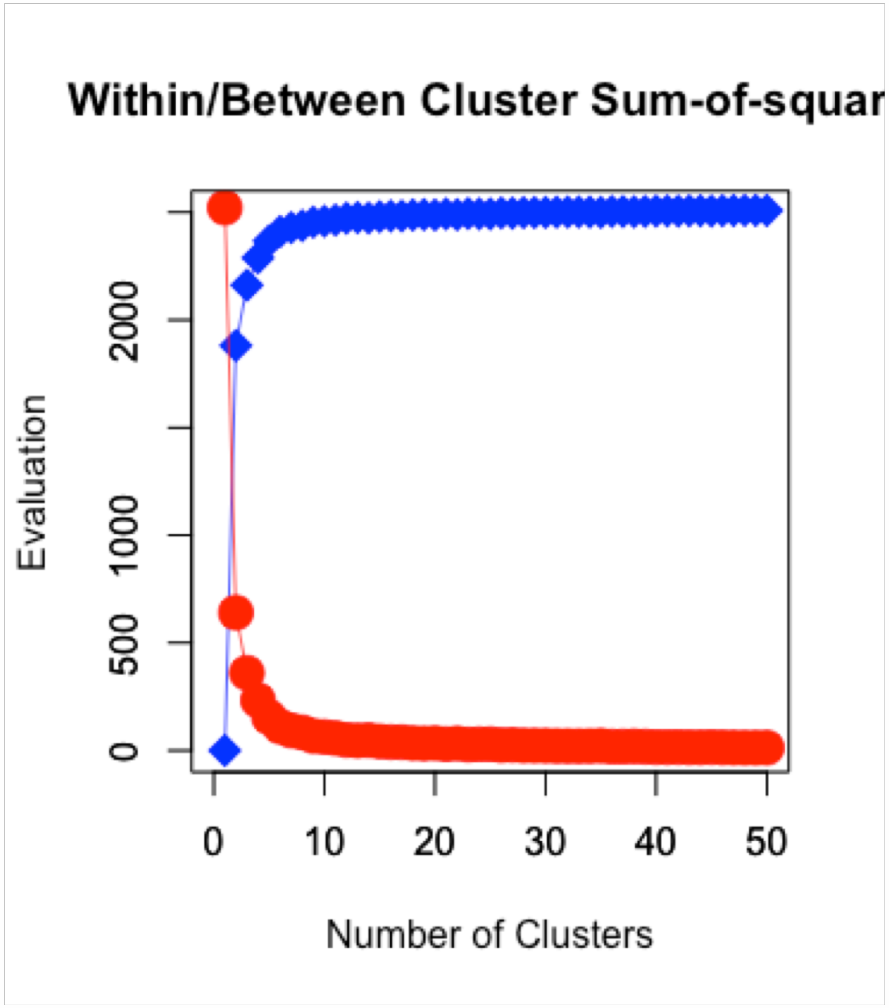




Clustering Comparisons on Sin Data

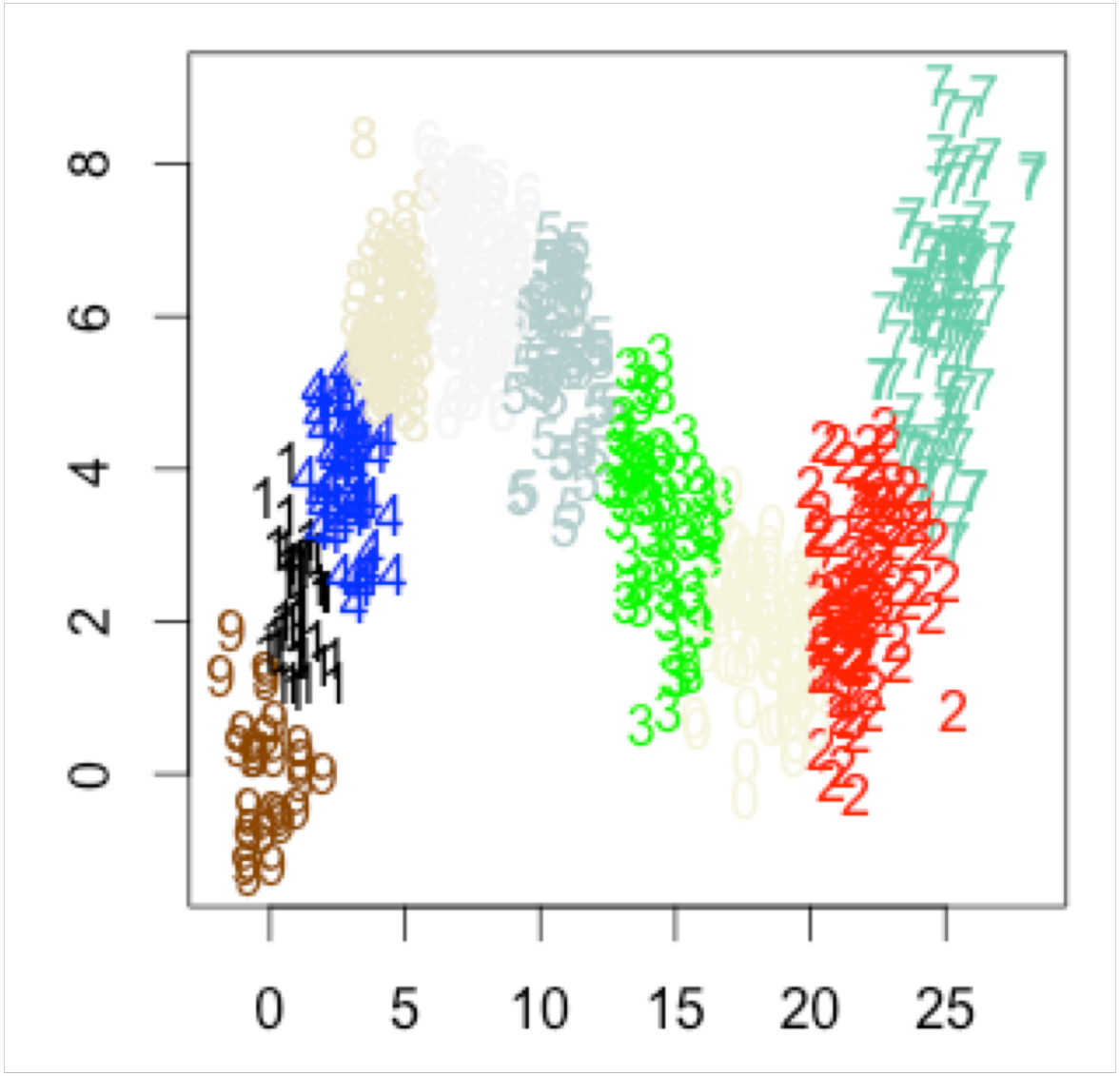


hierarchical clustering



kmeans clustering

Clustering Comparisons on Sin Data (k-means with 10 clusters)



Run the python notebooks and the KNIME workflows for this lecture