Assignment 2:
k-Means and DBSCAN

Erik Zeitler
Uppsala Database Laboratory

UPPSALA UNIVERSITET

Assignment 2 in a nutshell

- You will get
  - Two different data sets: `data/patterns.nt`, `data/patterns2.nt`
  - A k-means implementation
  - A DBSCAN implementation
- Study the data sets (visual inspection, queries etc)
  - How are they different?
- Find the best clustering on each data set, using both k-means and DBSCAN
  - k-means:
    - find and submit *k* and initial centroids
    - answer the questions
  - DBSCAN:
    - find and submit ε and *MinPts*
    - answer the questions

UPPSALA UNIVERSITET

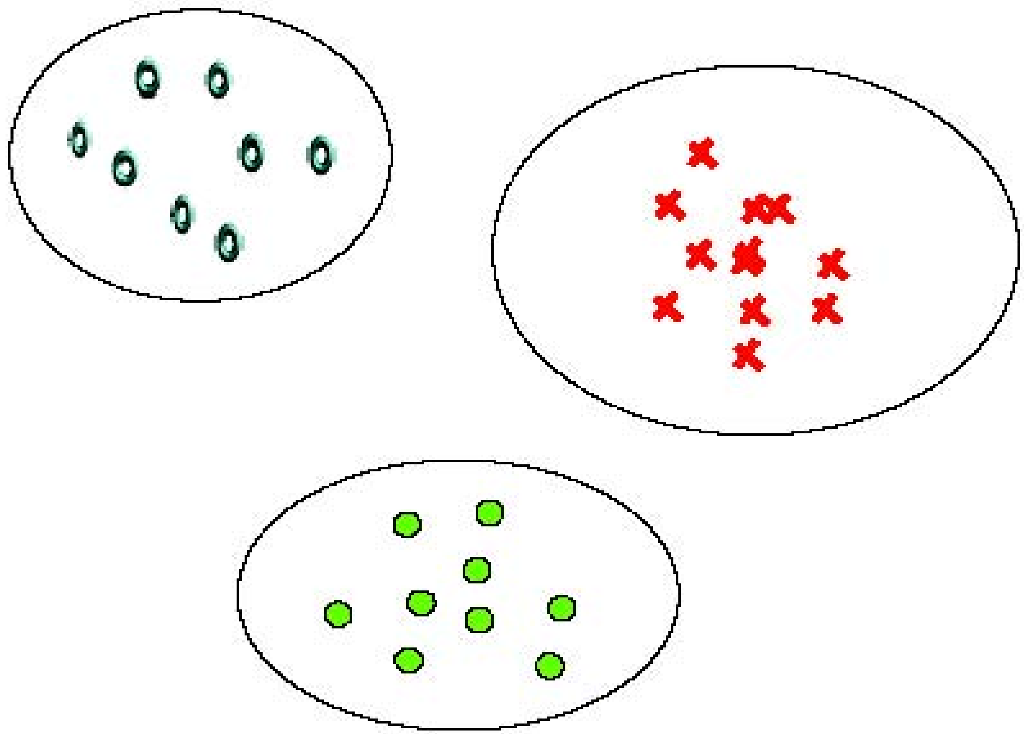
k-Means

- Input
  - M (set of points)
  - k (number of clusters)
- Output
  - $\mu_1, \ldots, \mu_k$ (set of k cluster centroids)
- k-Means is clustering M points into k clusters, by trying to minimize the squared error function

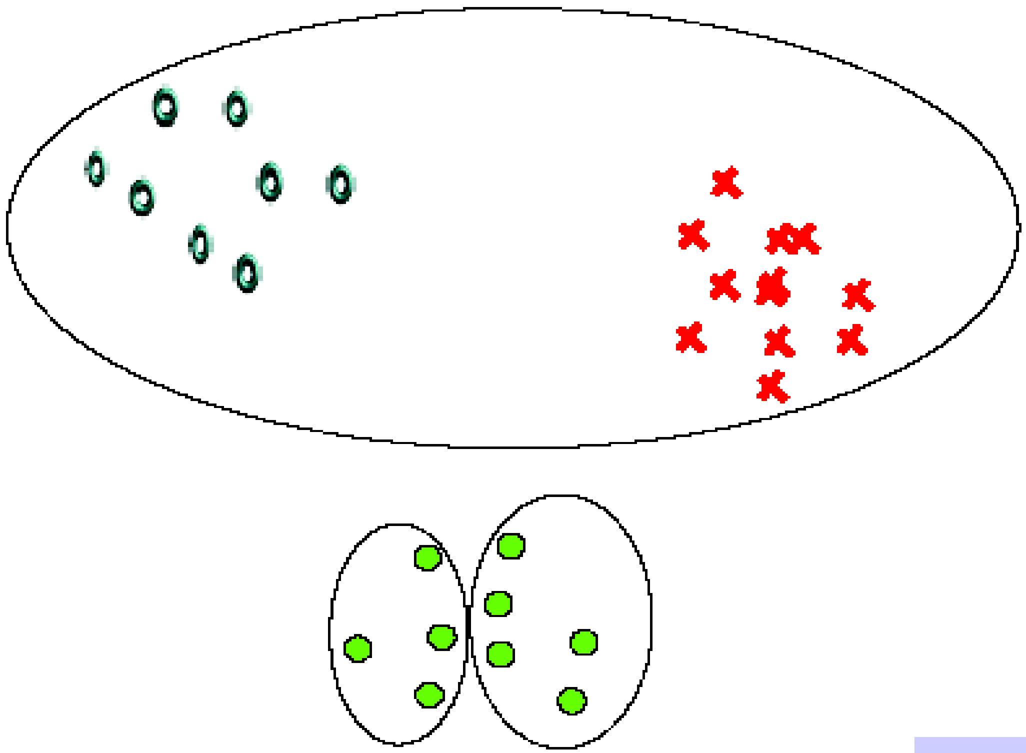
$$\sum_{i=1}^{k} \sum_{x_j \in S_i} \left(x_j - \mu_i\right)^2$$

μ_i is centroid of cluster S_i containing points $\{x_j\}$

K-Means on three clusters

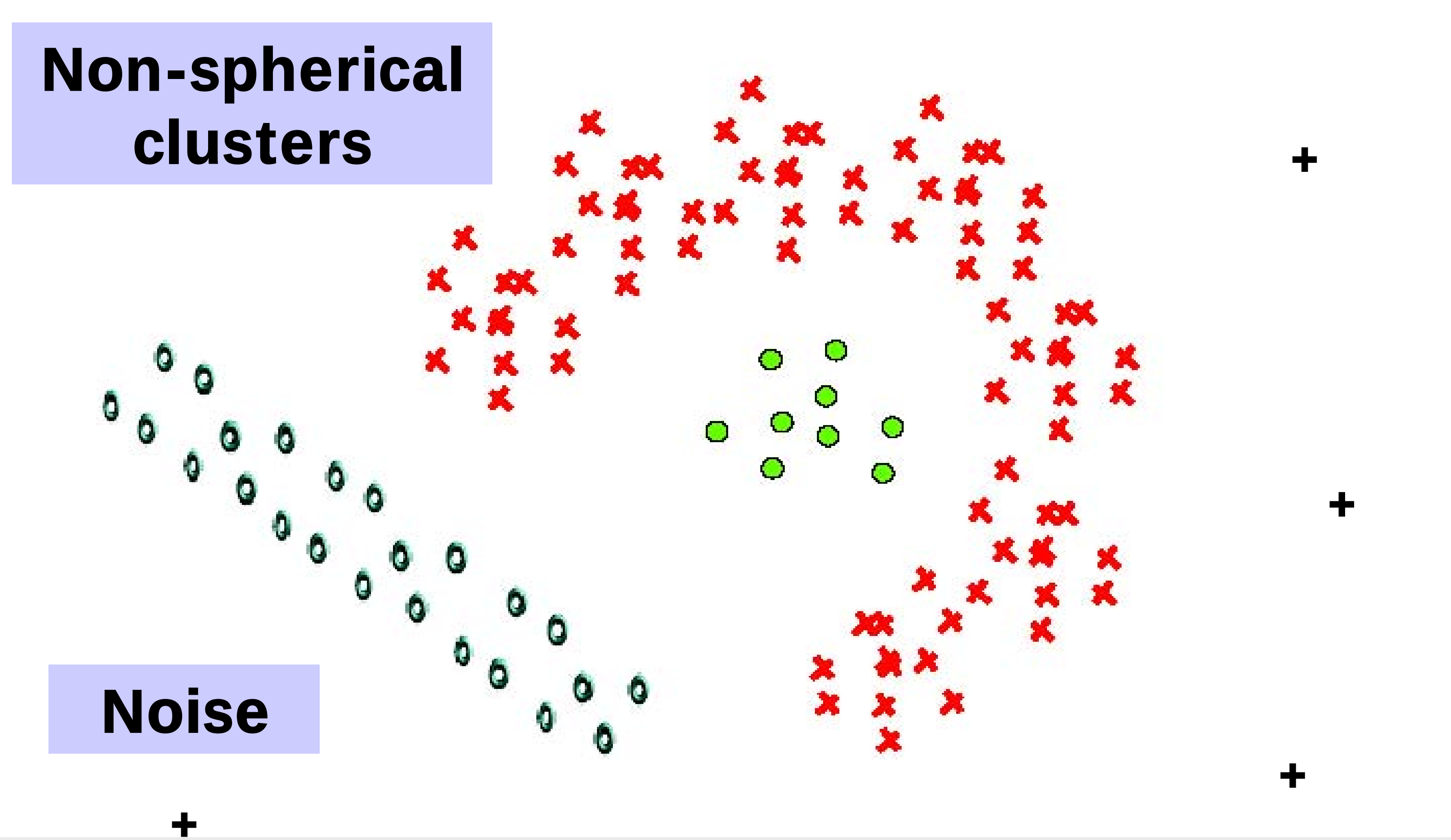


I'm feeling Unlucky



Bad initial points

Eat this!



Running k-Means

- How to choose k
  - study the data visually (pca + plot)
  - run k-Means for different k
    - compute SSE for each k
    - why not make a k-SSE-plot!
- How to choose initial centroids
  - select randomly among the data points
  - generate completely randomly
  - can we do something smarter?
- Submit your choice of initial centroids

UPPSALA UNIVERSITET

Questions

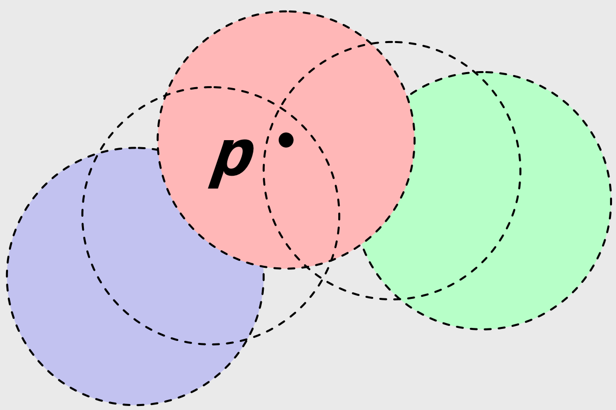
- How are the data sets different?
- Euclidean distance results in spherical clusters
  - What cluster shape does the Manhattan distance give?
  - Think of other distance measures too. What cluster shapes will those yield?
- Assuming that K-means converges

  in I iterations, with M points and X features for each point
  - give an approximation of the complexity expressed in K, I, M, and X
- Would normalization improve clustering quality?
  - Why? Why not?

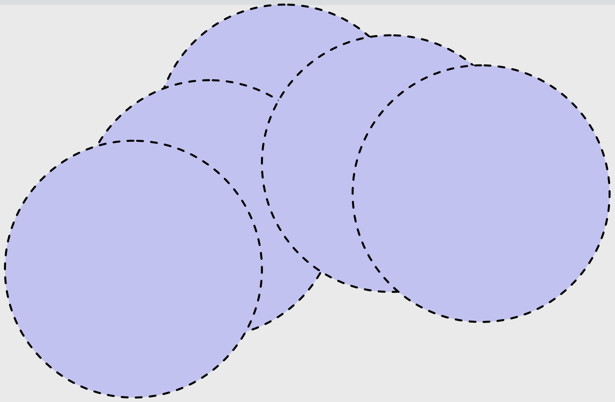
DBSCAN

```
i = 0
find the set of core points CP in M
do
  take a point p from CP
  find the set of points P which are
  density reachable to p
  if P = {}
     M = M \ {p}
  else
     Ci=P
     i=i+1
     M = M \ P
  end
while M ≠ {}
```

HOW?
Let's call this function **dr(p)**



Implement **dr**



```
create function dr(p) -> P

C = {p}
P = {p}

do
  remove a point p' from C
  find all points X that are ddr(p')
  C = C ∪ (X \ (P ∩ X))  % add newly discovered
                         % points to C
  P = P ∪ X
while C ≠ {}
result P
```

Running DBSCAN on you

- Are you a core point?
  - do you have at least *MinPts* neighbors within ε?
- For each core point
  - do you have another core point within ε?

    yes → merge

    (no → you are a separate cluster)

Running DBSCAN

- Guess ε using a k-dist plot
- Experiment with different ε and Minpts
- For each [ε, Minpts]:
  - Run DBSCAN
  - Count number of non-noise points:

```
count(select distinct
  ddr(datapoints(), #'datapoints',
  :eps, :minpts));
```

  - Count number of core points
  - Plot the core points
  - Plot the result of DBSCAN

UPPSALA UNIVERSITET

Questions

- For which points are density reachable symmetric, i.e.
  for which p, q: ***dr****(p, q)* and ***dr****(q, p)*?
- Express using only core objects and **ddr**, which objects will belong to a cluster