Unsupervised learning

- Goal:
  - Are there clusters in the data?

  - Many dimensions – total mess

How to define a cluster

- Define metric

$$dist(x, y) = \sum_{i=1}^{n} (x_i - y_i)^2$$

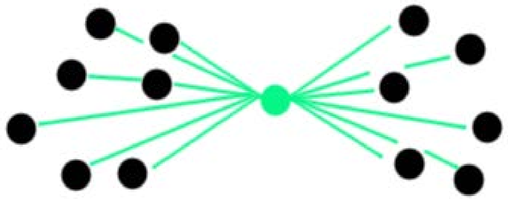
- What formal definition – let`s define `energy` as

$$V(k, \mu_1, \ldots, \mu_k) = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, \mu_i)$$

- The less – the better

K=1

K=2



total length of Green >> total length of Red

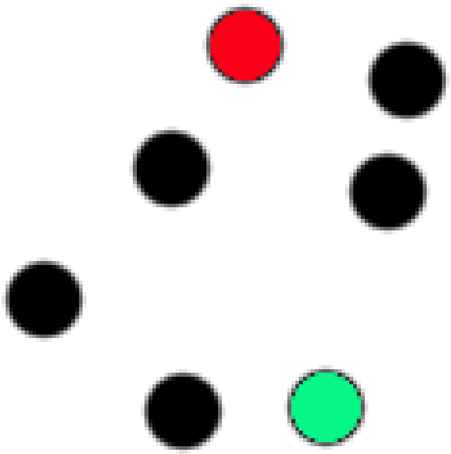
Task for a fixed k

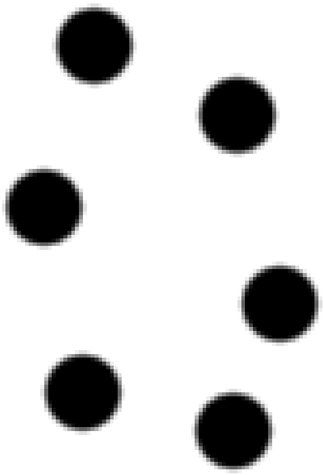
Find clusters that minimize the value of V

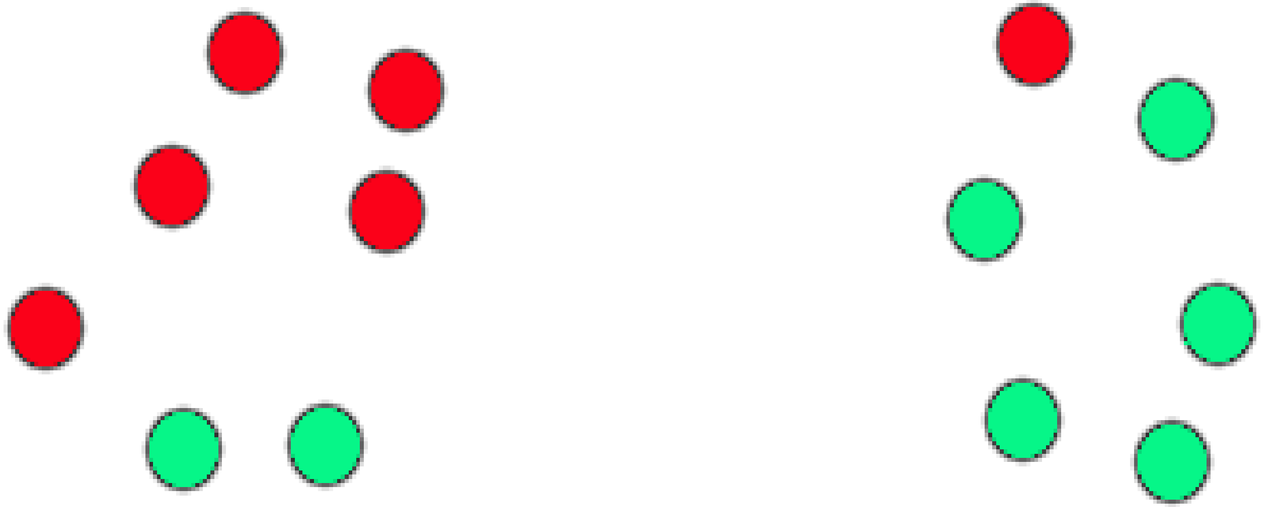
- It’s np-hard in general case
- Lloyd`s algorithm converges to local minimum

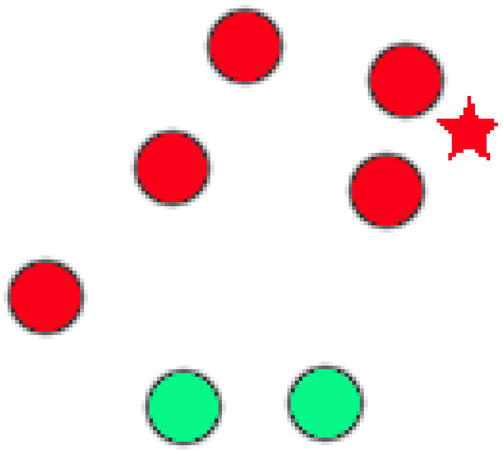
Lloyd’s algorithm

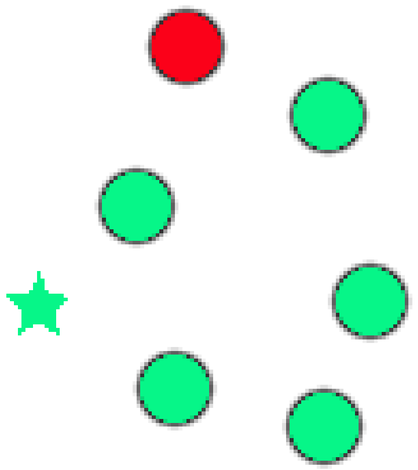
- EM version
  - Fix some value of k (?)
  - Take initial k centers (?)
  - Assign all points to clusters
  - Recalculate centers
  - Iterate previous two steps

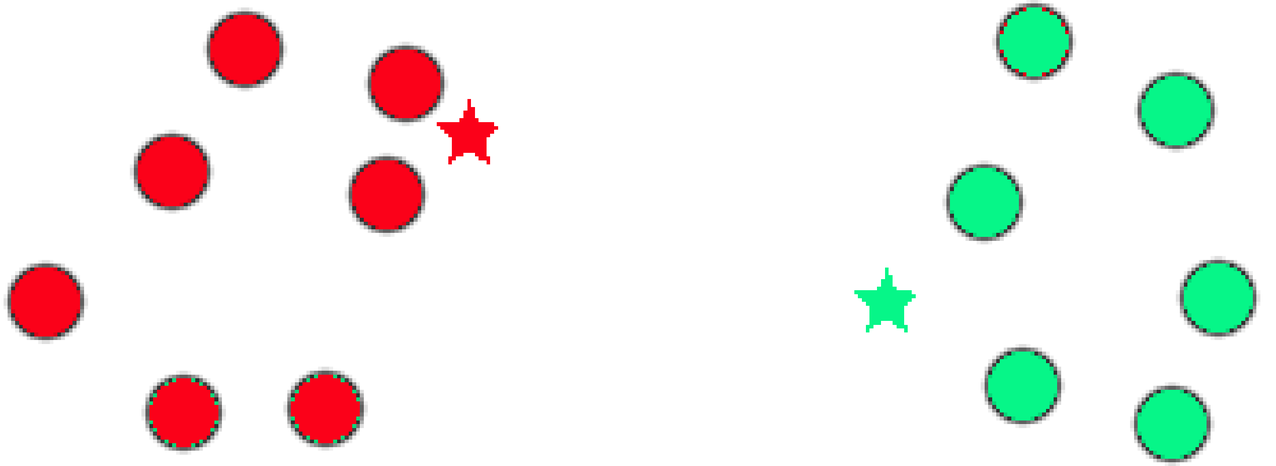


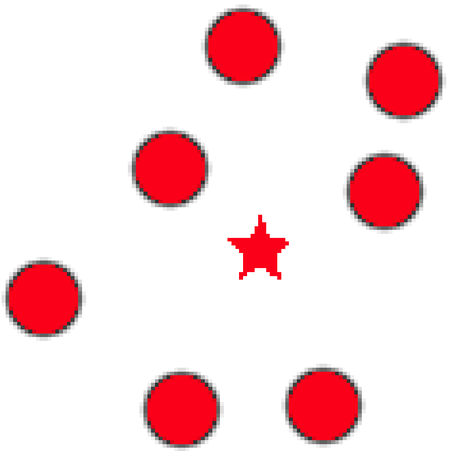


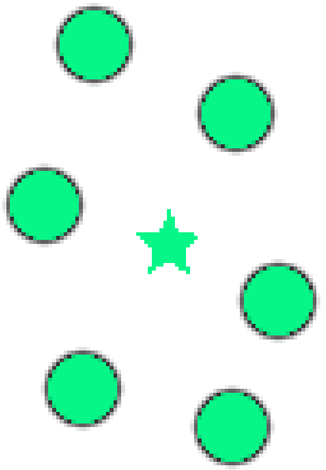












Notes

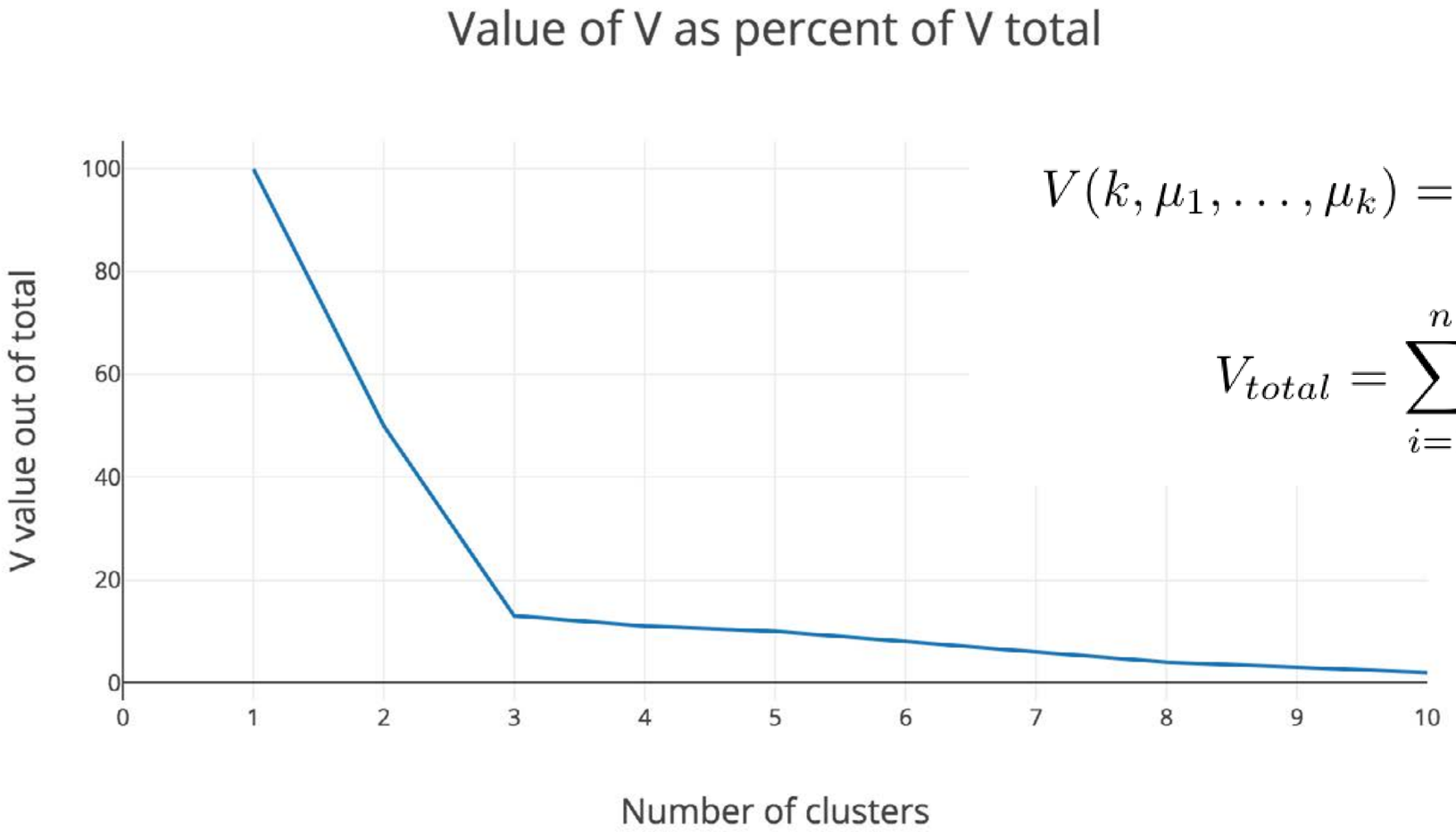
- Scaling again is important
  - Always use it
- Initialization is crucial
  - Never do random
  - Good heuristics – K-means++

K-means++

- Intuition: close points work poorly
- The further the point from already chosen – the better
  - Take 1st at random
  - For each point update D(x):=distance to closest cluster
  - Choose next center with probability proportional to D(x)^2

- K-means ‖ implemented in Spark MLlib
- First, generate a set of candidates for centers
  - Done in parallel
- Apply K-means++ only on those candidates
  - Don’t need much resources

How to define the number of clusters?



$$V(k, \mu_1, \ldots, \mu_k) = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, \mu_i)$$

$$V_{total} = \sum_{i=1}^{n} dist(x_i, \bar{x})$$

Recap

- What the clusters are
- How to formalize clustering
- K-means
  - K-means++ and K-means ||
  - Elbow method to define `k`