

IIT KHARAGPUR

IIT GANDHINAGAR


NPTEL ONLINE CERTIFICATION COURSES

Scalable Data Science

Lecture 23: Clustering

Sourangshu Bhattacharya

Computer Science and Engineering

IIT KHARAGPUR

In this Lecture:

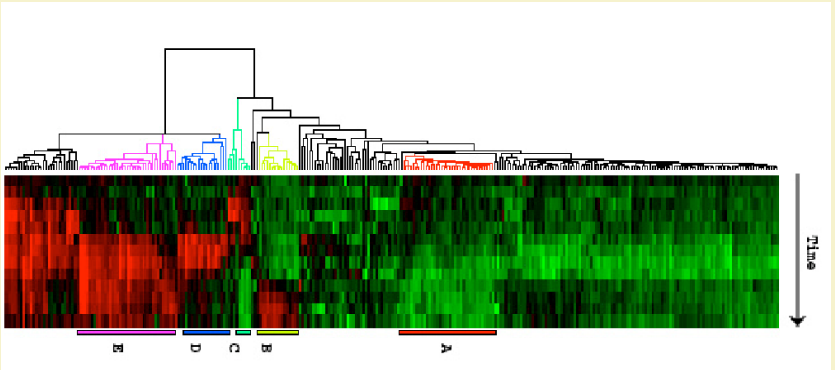
- K – means clustering and applications
- Lloyd's algorithm, EM and Limitations.
- K – means ++
- Scalable k – means ++

K – means clustering and applications

Clustering

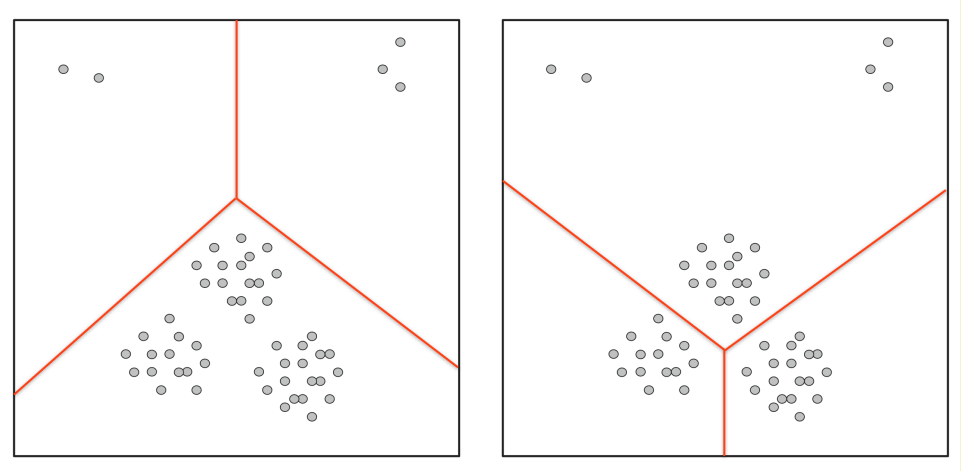
- Unsupervised learning
  - When your data doesn't have labels
- Useful for
  - Detecting patterns e.g. in image data, customer shopping results, anomalies…
  - For optimizing, e.g. distributing data across various machines, cleaning up search results, facility allocation for city planning…
  - when you "don't know" what is it exactly that we are looking for

[Image segmentation via clustering, James Hayes]



Clustering: basic idea

- Grouping objects into small number of meaningful groups
  - How to define similarity / distance between objects?
  - What is meaningful?
  - How many groups?
  - Typically there is no supervision



[Picture from Ackerman talk]

Developing framework: object representation

- First develop a mathematical representation of points
  - Object representation: E.g. vectors, set, sequences... when we want to represent the objects in isolation
    - Ex: Document → set / vector, image → vector , DNA → sequences
  - Interaction representation : as networks, when we are representing only the interaction between objects
    - Ex. Social / road / network,

Clustering framework: distance function

- In the object representation we need an appropriate distance function
  - Lp norms for vectors
  - Jaccard distance for sets
  - Edit distance for sequences
  - Divergences for probability distributions…

- Typically, nice to have the metric properties
  - $d(x,x) = 0,\ d(x,y) \geq 0$
  - $d(x,y) = d(y,x)$
  - $d(x,y) + d(y,z) \geq d(x,z)$

- Also nice if it is easy to calculate "average"

$$\min_x \sum_i d(p_i, x)$$

Distance function: lp norms

- L2 norm/Euclidean distance

$$D(x, y) = \sqrt[2]{\sum_{i=1}^{m} (x_i - y_i)^2}$$

- L1 norm
- L-infinity norm
- Easy to calculate averages.
- Also related is cosine distance

Objective function

- Specifying number of clusters
  - K-means / K-median
- Specifying cluster separation / quality
  - e.g. radius of cluster, Dunn's index,..
- Graph based measures
- Working w/o an objective function
  - Hierarchical clustering schemes

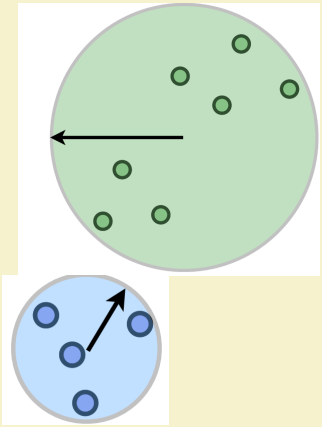
K-means

- Distance function is typically L2
- C = $\{c_1, c_2, \ldots, c_k\}$, cost(C)= $\sum_x \min_{c_x} d(x, c_x)^2$
- Find C to optimize the above cost
  - Leads to a natural partitioning of the data
- Large amount of work, both from theory & data mining community
  - Great example of divergence between theory and practice and how that prompted new research directions for both

k-means objective: alternate view

- Define “best” k-clustering of the data by
  - minimizing the “radius” of the each cluster

minimize $\sum_i radius(C_i)$



  - minimizing the variance of each cluster
    - The mean $c_i = \frac{1}{|C_i|}\sum_{x \in C_i} x$ is the “expected” location of a point
    - Hence variance of $C_i = \sum_{x \in C_i} ||x - c_i||^2$

[slide from Sergei V. & Suresh V.]

Lloyd's algorithm, EM and Limitations.

The canonical algorithm: Lloyd's algorithm

- Iterative algorithm
- Iterate
  - Find current centers of partitions
  - Assign points to nearest centers
  - Recalculate centers

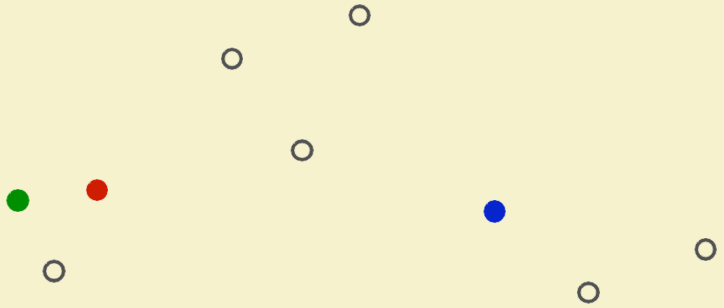
Lloyd's algorithm

- Iterative algorithm
- Iterate
  - Find current centers of partitions
  - Assign points to nearest centers
  - Recalculate centers

- Stopping criteria
  - when no (or small #) points change cluster
  - when cluster centers don't shift much
  - ….

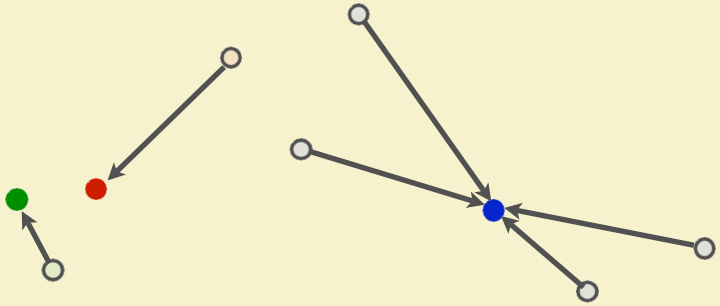
Lloyd's Method: k-means

Initialize with random clusters



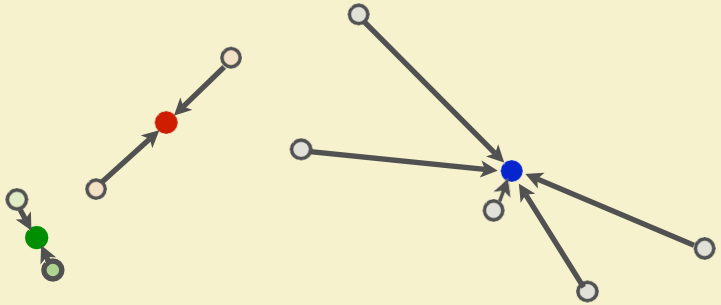
Lloyd's Method: k-means

Assign each point to nearest center



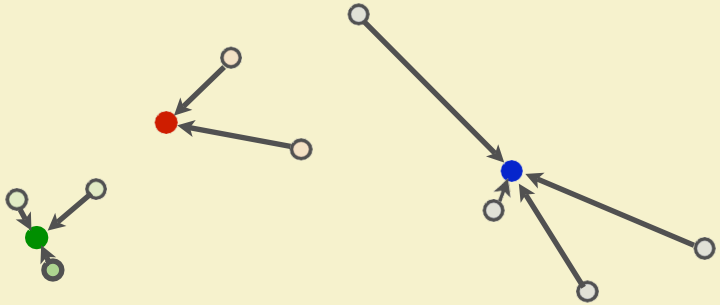
Lloyd's Method: k-means

Recompute optimum centers (means)



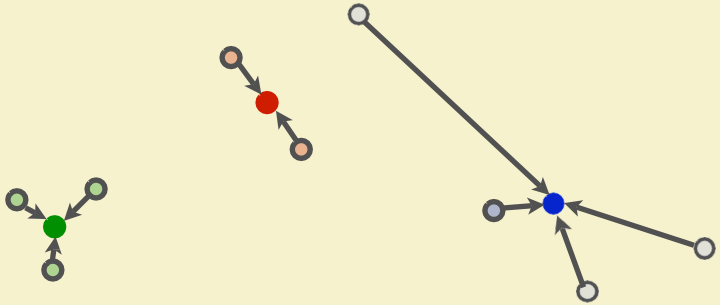
Lloyd's Method: k-means

Repeat: Assign points to nearest center



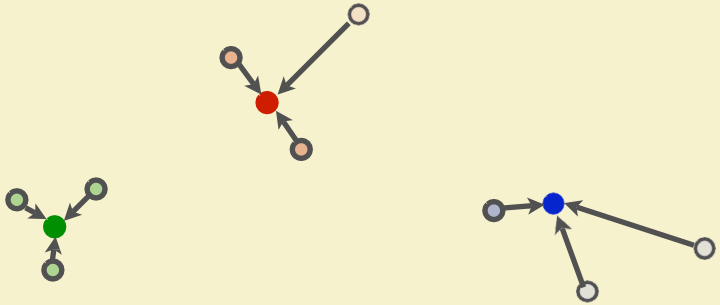
Lloyd's Method: k-means

Repeat: Recompute centers



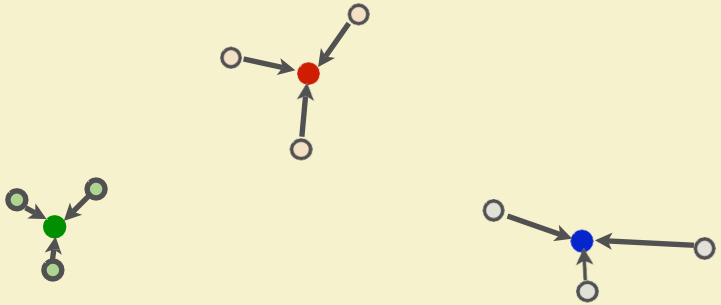
Lloyd's Method: k-means

Repeat...



Lloyd's Method: k-means

Repeat...Until clustering does not change



Lloyd's algorithm: analysis

- *k* centers, *N* points, *d* dimensions
- Time taken to calculate new cluster assignments : *O(k N d)*
- Time taken to calculate new centers : *O(Nd)*
- Number of iterations?

Lloyd's algorithm: convergence?

- For any current clustering, consider the objective function

$$\text{cost(C)} = \sum_x \min_{c_x} d(x, c_x)^2$$

Lloyd's algorithm: convergence?

- For any current clustering, consider the objective function

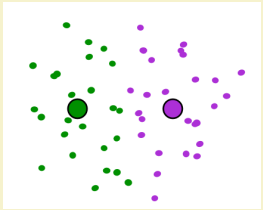
$$\text{cost(C)} = \sum_x \min_{c_x} d(x, c_x)^2$$

- At every step of the algorithm, this potentially decreases

Convergence

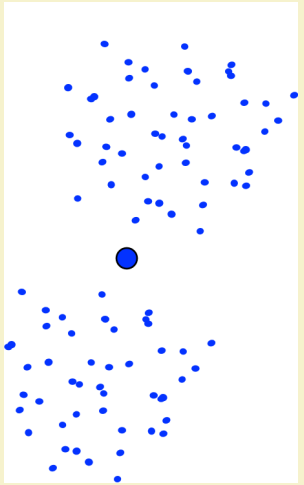
- It is known that in some datasets, Lloyd's algorithm can take exponential $(2^{\sqrt{n}})$ number of steps
  - These tend to be unrealistic

- Bigger problem is where it converges to--- depends on initialization



Should have put single cluster here

and two here

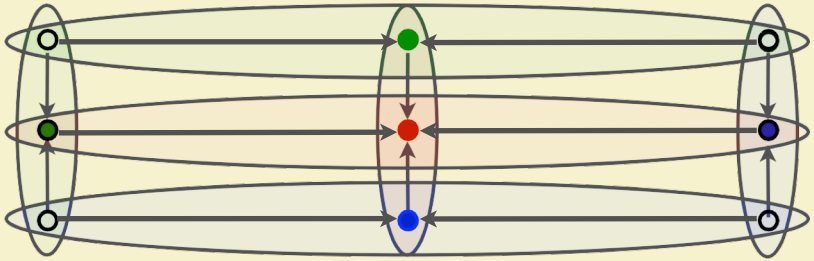


[Example from Sontag]

Convergence Analysis

Lloyd's Algorithm can be thought as a generalization of EM –algorithm for estimating mixtures of Gaussian distribution.

Finds a local optimum



That is potentially arbitrarily worse than optimal solution

K-means ++

Challenge

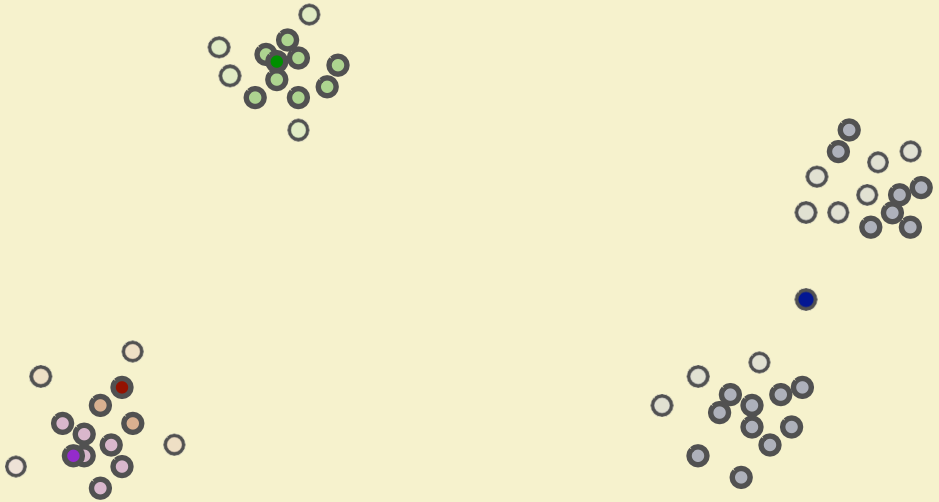
Develop an approximation algorithm for k-means clustering that is competitive with the k-means method in speed and solution quality.

Easiest line of attack: focus on the initial center positions.

Classical k-means: pick k points at random.

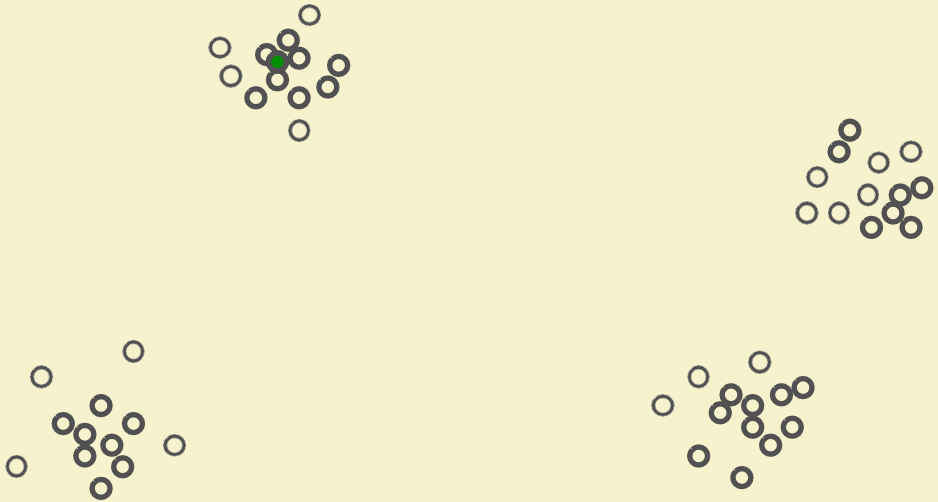
k-means on Gaussians

k-means on Gaussians



Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

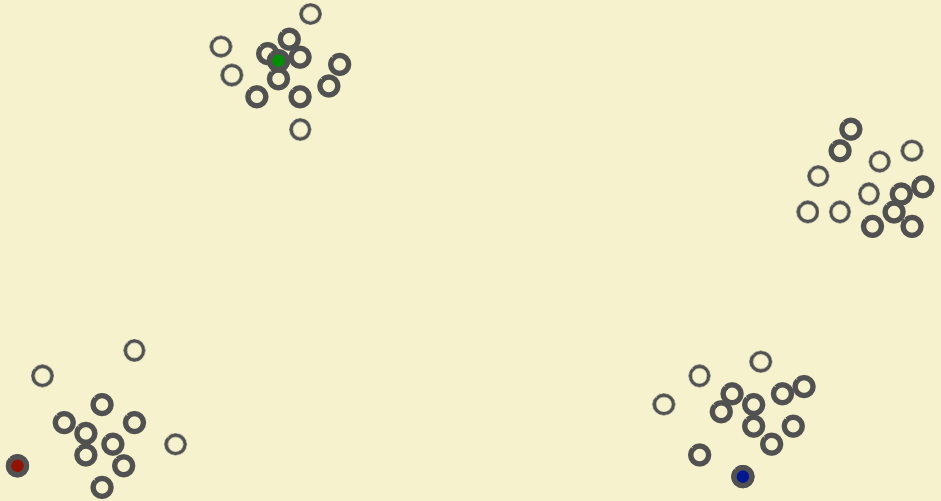


Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

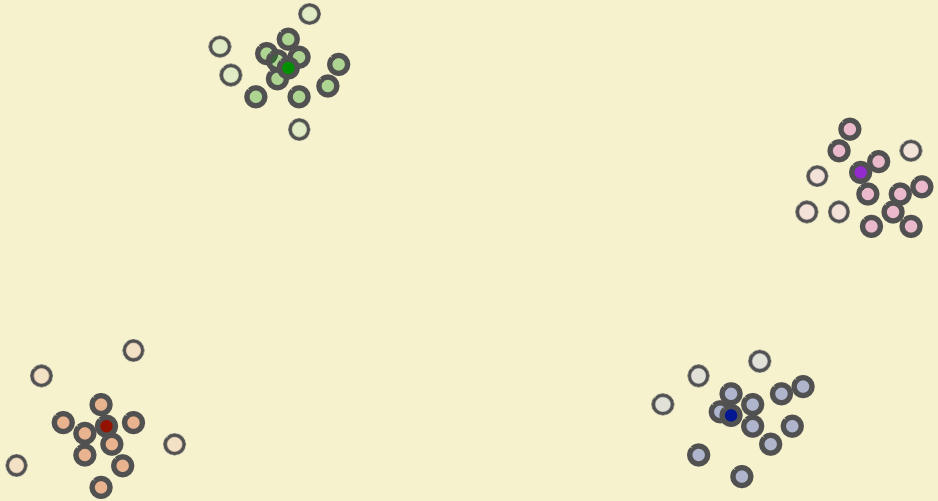


Easy Fix

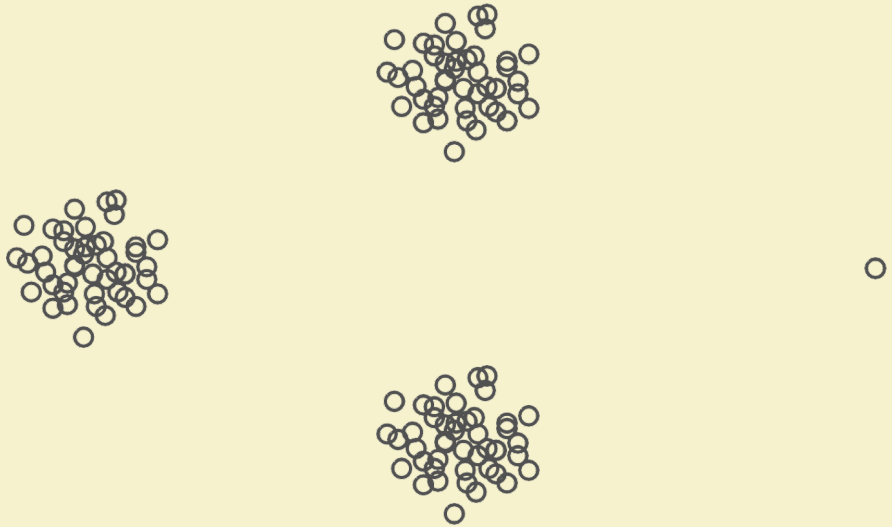
Select centers using a furthest point algorithm (2-approximation to k-Center clustering).

Easy Fix

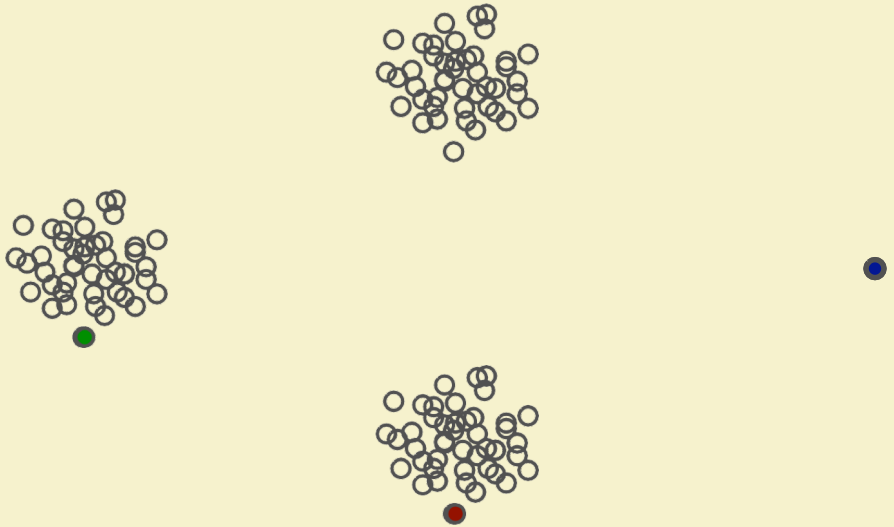
Select centers using a furthest point algorithm (2-approximation to k-Center clustering).



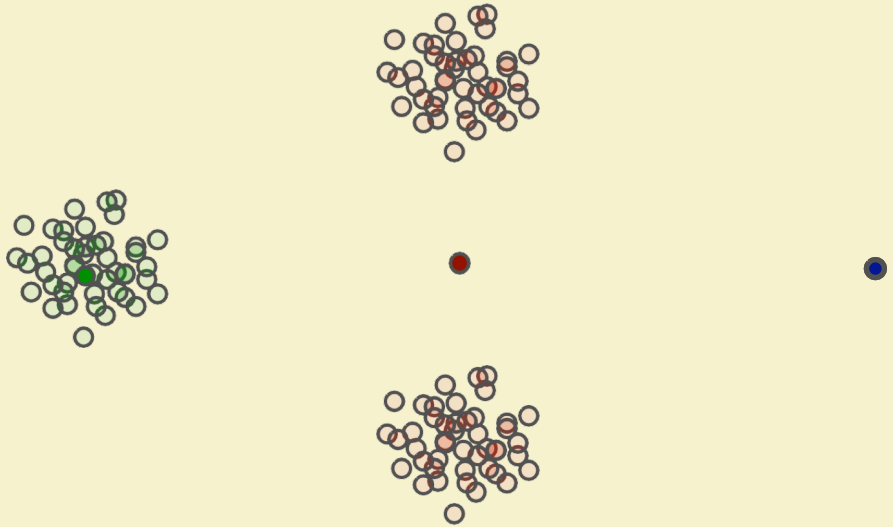
Sensitive to Outliers



Sensitive to Outliers



Sensitive to Outliers



k-means++

Interpolate between the two methods:

Let $D(x)$ be the distance between x and the nearest cluster center. Sample proportionally to $(D(x))^{\alpha} = D^{\alpha}(x)$

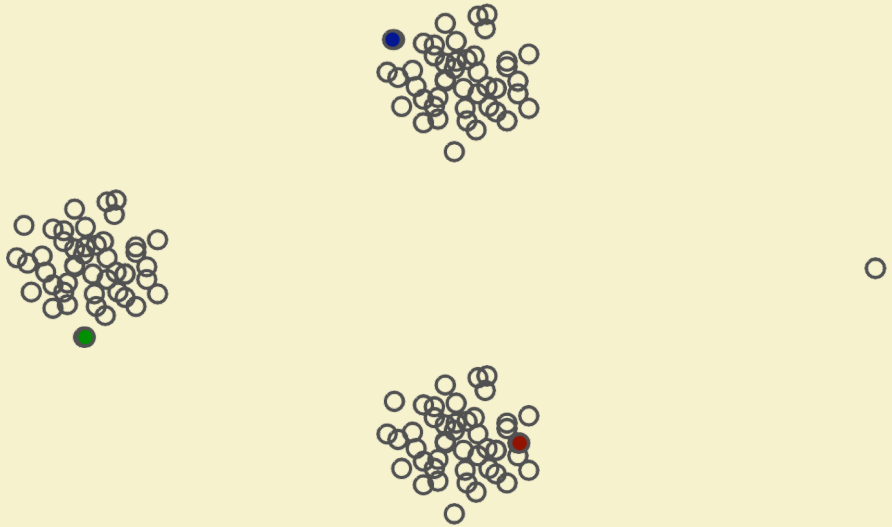
Original Lloyd's: $\alpha = 0$

Furthest Point: $\alpha = \infty$

k-means++: $\alpha = 2$

Contribution of *x* to the overall error

k-Means++



k-Means++

Theorem *Let $\mathcal{C}$ be an arbitrary clustering. Choose $u > 0$ "uncovered" clusters from $\mathcal{C}_{\text{OPT}}$, and let $\mathcal{X}_u$ denote the set of points in these clusters. Also let $\mathcal{X}_c = \mathcal{X} - \mathcal{X}_u$. Now suppose we add $t \leq u$ random centers to $\mathcal{C}$, chosen with D^2 weighting. Let $\mathcal{C}'$ denote the the resulting clustering, and let ϕ' denote the corresponding potential. Then,*

$$E[\phi'] \leq \left(\phi(\mathcal{X}_c) + 8\phi_{\text{OPT}}(\mathcal{X}_u)\right) \cdot (1 + H_t) + \frac{u-t}{u} \cdot \phi(\mathcal{X}_u).$$

Here, H_t denotes the harmonic sum, $1 + \frac{1}{2} + \cdots + \frac{1}{t}$.

k-Means++

Theorem: k-means++ is $\Theta(\log k)$ approximate in expectation.

K-MEANS ||

What's wrong with K-means++?

- Needs K passes over the data
- In large data applications, not only the data is massive, but also K is typically large (e.g., easily 1000).
- Does not scale!

Intuition for a solution

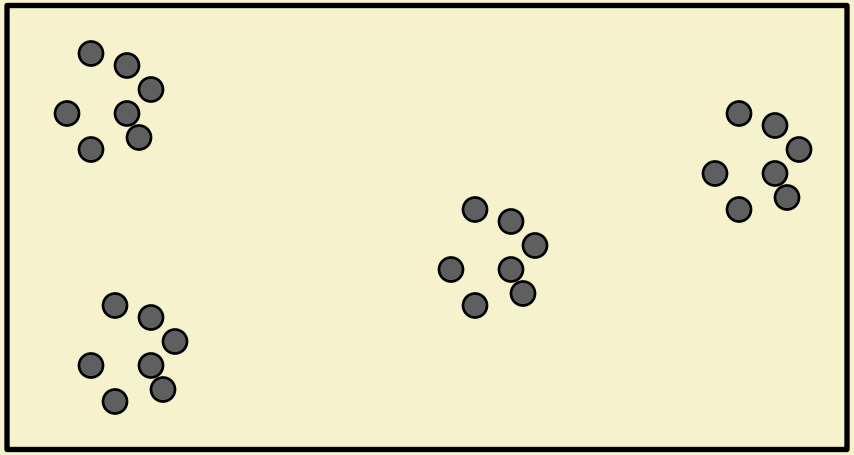
- K-means++ samples one point per iteration and updates its distribution
- What if we **oversample** by sampling each point independently with a larger probability?
- Intuitively equivalent to updating the distribution much less frequently
  - Coarser sampling
- Turns out to be sufficient: K-means||

K-means|| [Bahmani et al. '12]

- Choose $l>1$ [Think $l=\Theta(k)$]
- Initialize C to an arbitrary set of points
- For R iterations do:
  - Sample each point x in X independently with probability $p_x = ld^2(x,C)/\varphi_X(C)$.
  - Add all the sampled points to C
- Cluster the (weighted) points in C to find the final k centers

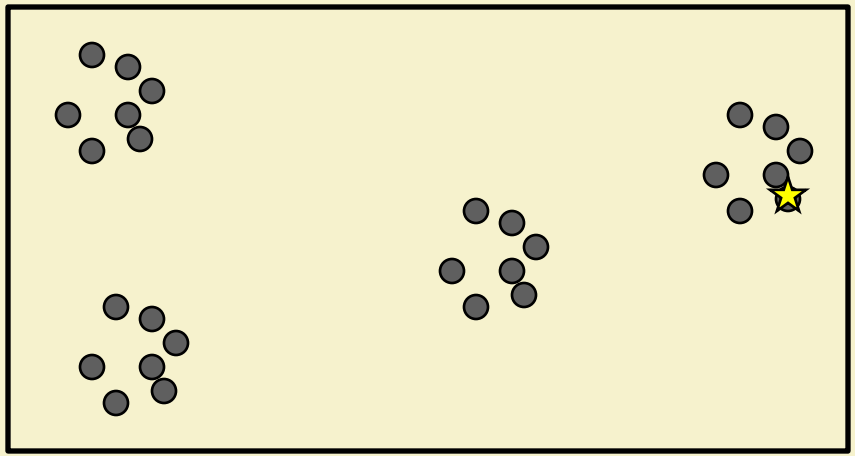
K-means| | Initialization

K=4,
Oversampling factor =3



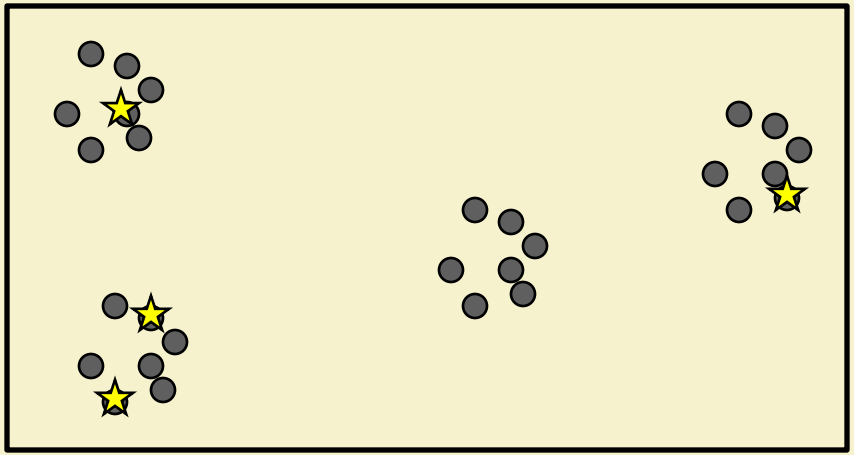
K-means| | Initialization

K=4,
Oversampling factor =3



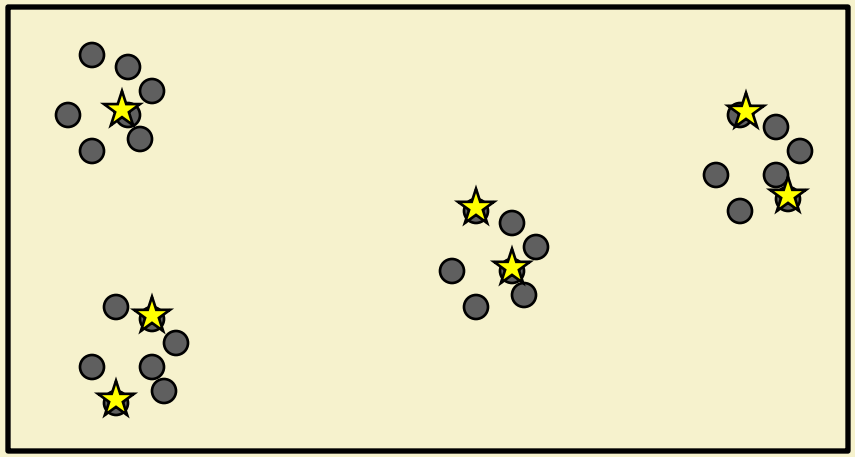
K-means| | Initialization

K=4,
Oversampling factor =3



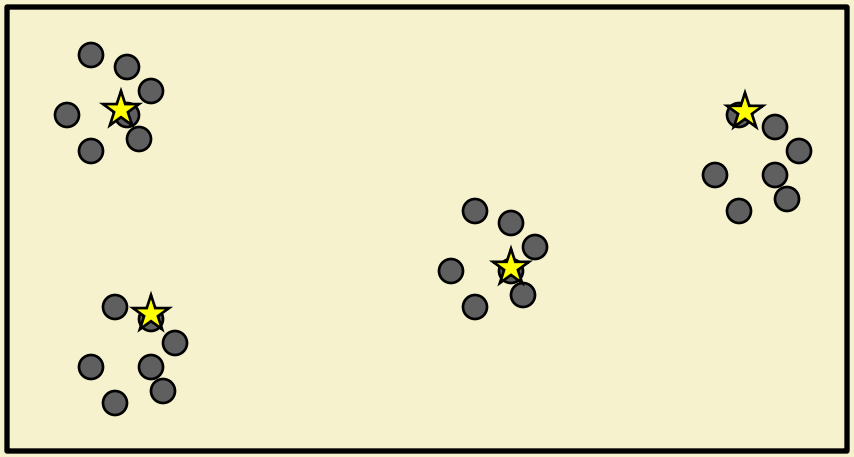
K-means| | Initialization

K=4,
Oversampling factor =3



K-means | | Initialization

K=4,
Oversampling factor =3



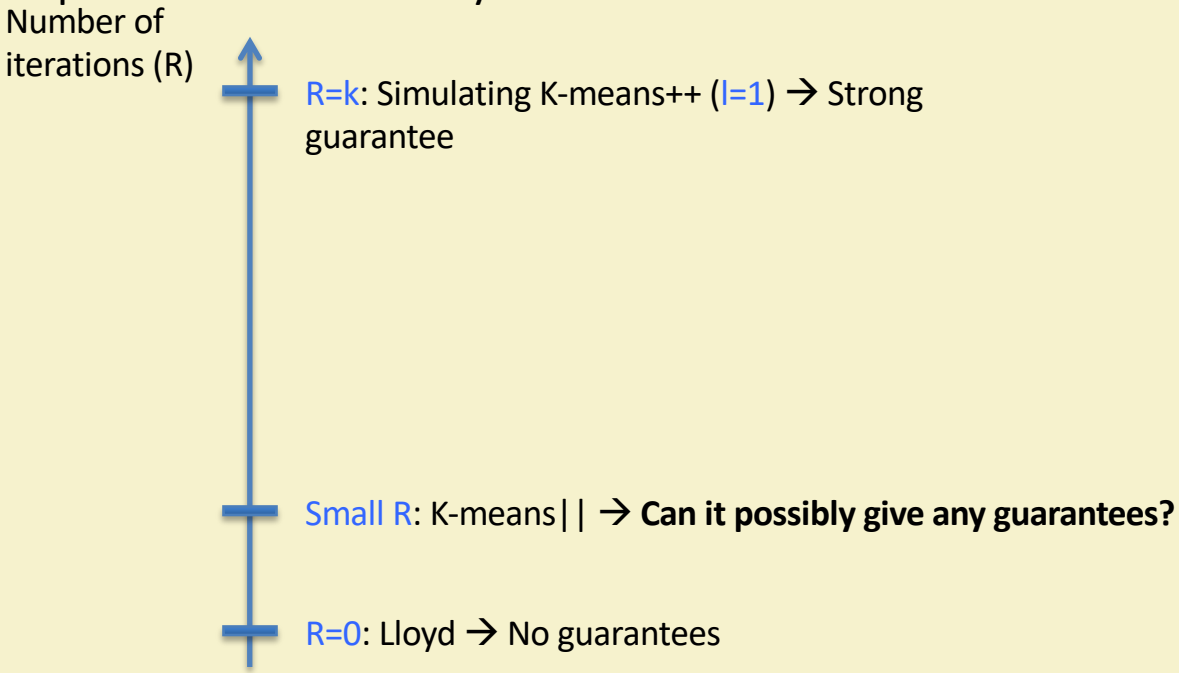
Cluster the intermediate centers

K-means|| [Bahmani et al. '12]

- Choose $l>1$ [Think $l=\Theta(k)$]
- Initialize C to an arbitrary set of points
- For R iterations do:
  - Sample each point x in X independently with probability $p_x = l d^2(x,C)/\varphi_X(C)$.
  - Add all the sampled points to C
- Cluster the (weighted) points in C to find the final k centers

K-means| |: Intuition

- An interpolation between Lloyd and K-means++



Theorem

- **Theorem:** If φ and φ’ are the costs of the clustering at the beginning and end of an iteration, and OPT is the cost of the optimum clustering:

$$E[\varphi'] \leq O(OPT) + \frac{k}{el}\varphi$$

- **Corollary:**
  - Let ψ= cost of initial clustering
  - K-means|| produces a constant-factor approximation to OPT, using only O(log (ψ/OPT)) iterations

Experimental Results: Quality

	Clustering Cost Right After Initialization	Clustering Cost After Lloyd Convergence
Random	NA	22,000
K-means++	430	65
K-means||	16	14

GAUSSMIXTURE: 10,000 points in 15 dimensions
K=50
Costs scaled down by 10^4

- K-means|| much harder than K-means++ to get confused with noisy outliers

Experimental Results: Convergence

- K-means|| reduces number of Lloyd iterations even more than K-means++

	Number of Lloyd Iterations till Convergence
Random	167
K-means++	42
K-means||	28

SPAM: 4,601 points in 58 dimensions
K=50

References:

- David Arthur , Sergei Vassilvitskii, **k-means++: the advantages of careful seeding**, *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, p.1027-1035, January 07-09, 2007, New Orleans, Louisiana.*

- Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. **Scalable k-means++**. *Proceedings of VLDB Endowment 5, 7 (March 2012), 622-633.*

Thank You!!