10-301/601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

K-Means
+
Societal Impacts of ML

Henry Chai & Matt Gormley
Lecture 26
Dec. 5, 2022

Reminders

- **Homework 9: Learning Paradigms**
  - Out: Fri, Dec. 2
  - Due: Fri, Dec. 9 at 11:59pm
    (only two grace/late days permitted)

Crowdsourcing Exam Questions

In-Class Exercise

1. Select one of lecture-level learning objectives
   http://mlcourse.org/slides/10601-objectives.pdf
2. Write a question that assesses that objective
3. Adjust to avoid 'trivia style' question

Answer Here:

CLUSTERING

Clustering, Informal Goals

Goal: Automatically partition unlabeled data into groups of similar data points.

Question: When and why would we want to do this?

Useful for:

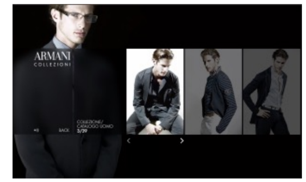
- Automatically organizing data.
- Understanding hidden structure in data.
- Preprocessing for further analysis.
  - Representing high-dimensional data in a low-dimensional space (e.g., for visualization purposes).

Slide courtesy of Nina Balcan

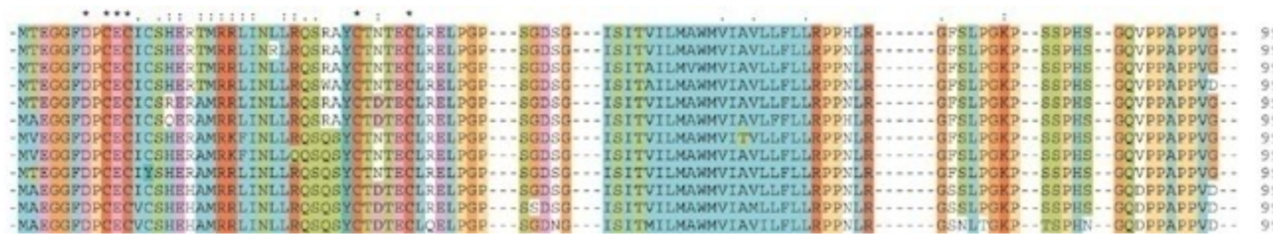
Applications (Clustering comes up everywhere…)

- Cluster news articles or web pages or search results by topic.

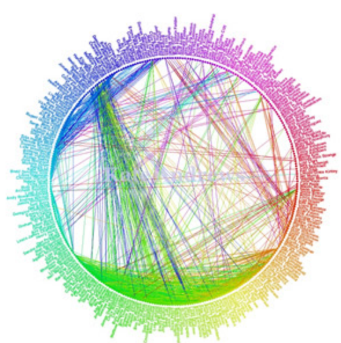


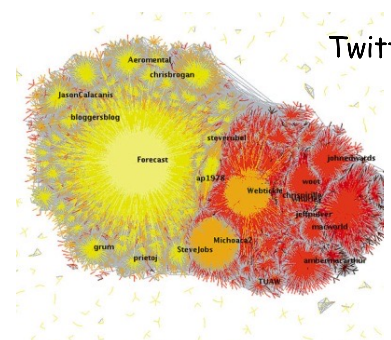
- Cluster protein sequences by function or genes according to expression profile.

- Cluster users of social networks by interest (community detection).



Facebook network

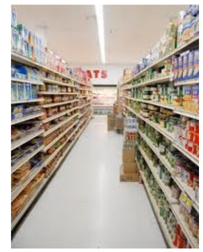


Twitter Network

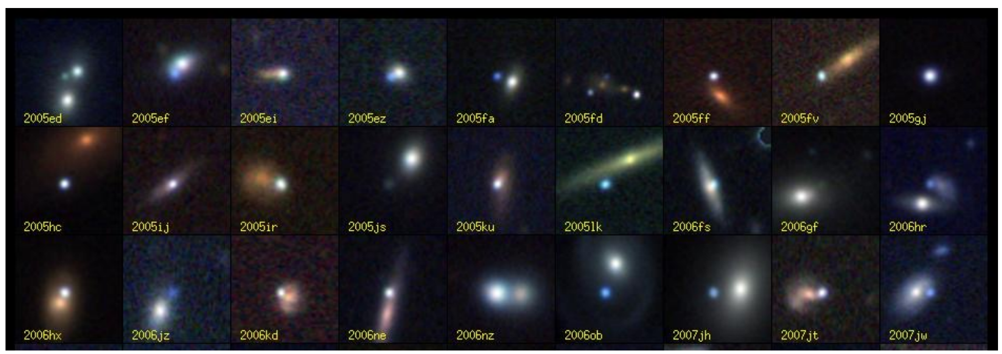
Slide courtesy of Nina Balcan

Applications (Clustering comes up everywhere…)

- Cluster customers according to purchase history.



- Cluster galaxies or nearby stars (e.g. Sloan Digital Sky Survey)

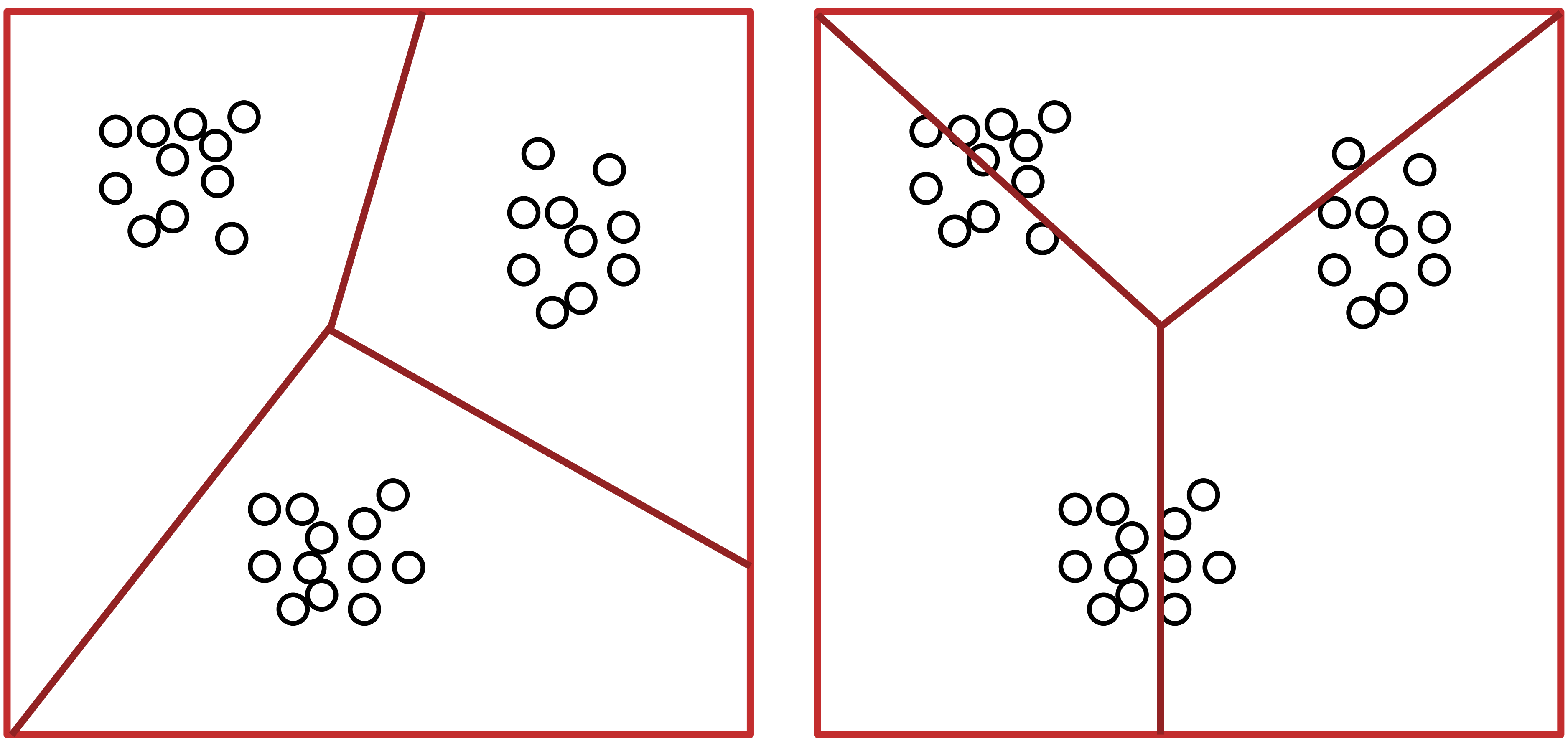


- And many many more applications….

Slide courtesy of Nina Balcan

Clustering

Question: Which of these partitions is "better"?



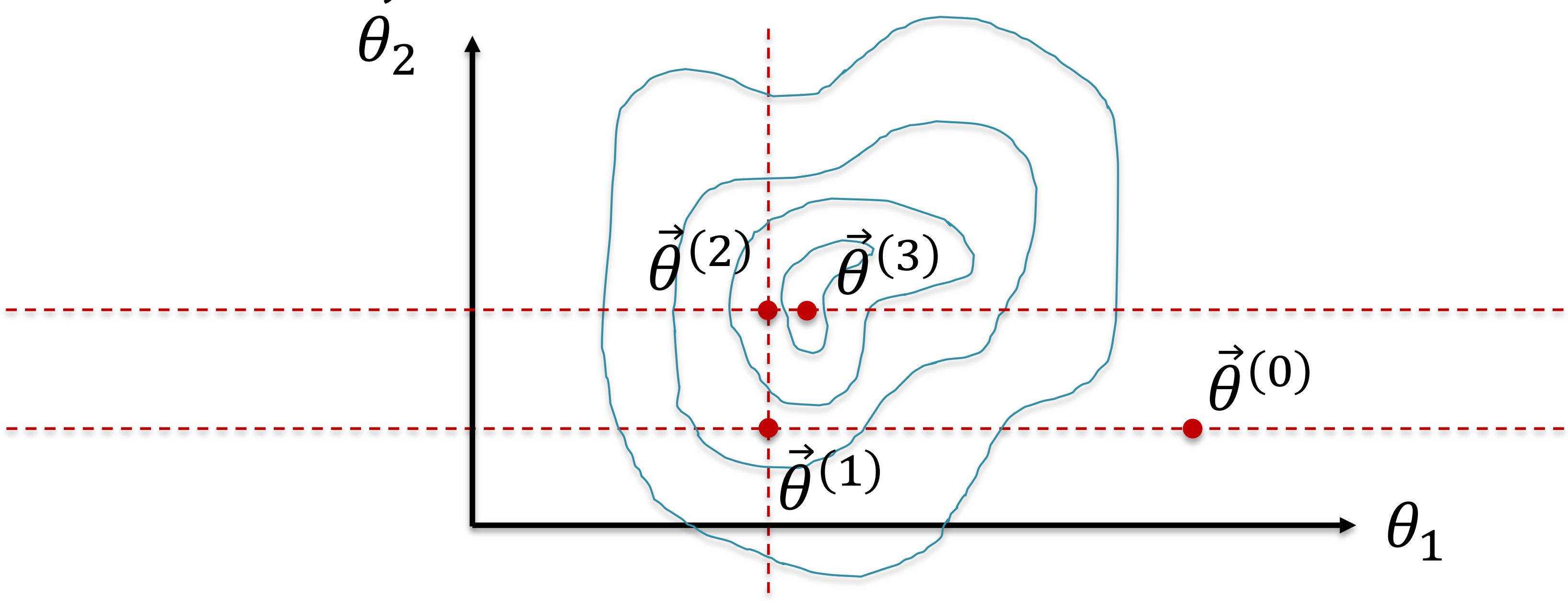
OPTIMIZATION BACKGROUND

Coordinate Descent

- Goal: minimize some objective

$$\vec{\theta}^* = \underset{\vec{\theta}}{\operatorname{argmin}} J(\vec{\theta})$$

- Idea: iteratively pick one variable and minimize the objective w.r.t. just that one variable, *keeping all the others fixed.*



Block Coordinate Descent

- Goal: minimize some objective (with 2 blocks)

$$\vec{\alpha}^*, \vec{\beta}^* = \underset{\vec{\alpha}, \vec{\beta}}{\operatorname{argmin}} J(\vec{\alpha}, \vec{\beta})$$

- Idea: iteratively pick one *block* of variables ($\vec{\alpha}$ or $\vec{\beta}$) and minimize the objective w.r.t. that block, keeping the other(s) fixed.

while not converged:

$$\vec{\alpha} = \underset{\vec{\alpha}}{\operatorname{argmin}} J(\vec{\alpha}, \vec{\beta})$$

$$\vec{\beta} = \underset{\vec{\beta}}{\operatorname{argmin}} J(\vec{\alpha}, \vec{\beta})$$

K-MEANS

K-Means Algorithm (Derivation)

Recipe for K-Means Derivation:

1) Define a Model.
2) Choose an objective function.
3) Optimize it!

K-Means Algorithm (Derivation)

- Input: unlabeled data $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N},\ \mathbf{x}^{(i)} \in \mathbb{R}^M$
- Goal: Find an assignment of points to clusters
- Model Paramters:
  - cluster centers: $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K],\ \mathbf{c}_j \in \mathbb{R}^M$
  - cluster assignments: $\mathbf{z} = [z^{(1)}, z^{(2)}, \ldots, z^{(N)}],\ z^{(i)} \in \{1, \ldots, K\}$
- Decision Rule: assign each point $\mathbf{x}^{(i)}$ to its nearest cluster center $\mathbf{c}_j$

K-Means Algorithm (Derivation)

- Input: unlabeled data $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}, \ \mathbf{x}^{(i)} \in \mathbb{R}^M$
- Goal: Find an assignment of points to clusters
- Model Paramters:
  - cluster centers: $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K], \ \mathbf{c}_j \in \mathbb{R}^M$
  - cluster assignments: $\mathbf{z} = [z^{(1)}, z^{(2)}, \ldots, z^{(N)}], \ z^{(i)} \in \{1, \ldots, K\}$
- Decision Rule: assign each point $\mathbf{x}^{(i)}$ to its nearest cluster center $\mathbf{c}_j$
- Objective:

$$\hat{\mathbf{C}} = \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{N} \min_j ||\mathbf{x}^{(i)} - \mathbf{c}_j||_2^2$$

K-Means Algorithm (Derivation)

- Input: unlabeled data $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N, \;\; \mathbf{x}^{(i)} \in \mathbb{R}^M$
- Goal: Find an assignment of points to clusters
- Model Paramters:
  - cluster centers: $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K], \;\; \mathbf{c}_j \in \mathbb{R}^M$
  - cluster assignments: $\mathbf{z} = [z^{(1)}, z^{(2)}, \dots, z^{(N)}], \;\; z^{(i)} \in \{1, \dots, K\}$
- Decision Rule: assign each point $\mathbf{x}^{(i)}$ to its nearest cluster center $\mathbf{c}_j$
- Objective:

$$\hat{\mathbf{C}} = \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{j} ||\mathbf{x}^{(i)} - \mathbf{c}_j||_2^2$$

$$= \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{z^{(i)}} ||\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}||_2^2$$

K-Means Algorithm (Derivation)

- Input: unlabeled data $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N, \ \ \mathbf{x}^{(i)} \in \mathbb{R}^M$
- Goal: Find an assignment of points to clusters
- Model Paramters:
  - cluster centers: $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K], \ \ \mathbf{c}_j \in \mathbb{R}^M$
  - cluster assignments: $\mathbf{z} = [z^{(1)}, z^{(2)}, \ldots, z^{(N)}], \ \ z^{(i)} \in \{1, \ldots, K\}$
- Decision Rule: assign each point $\mathbf{x}^{(i)}$ to its nearest cluster center $\mathbf{c}_j$
- Objective:

$$\hat{\mathbf{C}} = \underset{\mathbf{C}}{\text{argmin}} \sum_{i=1}^N \min_j ||\mathbf{x}^{(i)} - \mathbf{c}_j||_2^2$$

$$= \underset{\mathbf{C}}{\text{argmin}} \sum_{i=1}^N \min_{z^{(i)}} ||\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}||_2^2$$

$$\hat{\mathbf{C}}, \hat{\mathbf{z}} = \underset{\mathbf{C},\mathbf{z}}{\text{argmin}} \sum_{i=1}^N ||\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}||_2^2$$

K-Means Algorithm (Derivation)

- Input: unlabeled data $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}, \ \mathbf{x}^{(i)} \in \mathbb{R}^M$
- Goal: Find an assignment of points to clusters
- Model Paramters:
  - cluster centers: $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K], \ \mathbf{c}_j \in \mathbb{R}^M$
  - cluster assignments: $\mathbf{z} = [z^{(1)}, z^{(2)}, \dots, z^{(N)}], \ z^{(i)} \in \{1, \dots, K\}$
- Decision Rule: assign each point $\mathbf{x}^{(i)}$ to its nearest cluster center $\mathbf{c}_j$
- Objective:

$$\hat{\mathbf{C}} = \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{j} ||\mathbf{x}^{(i)} - \mathbf{c}_j||_2^2$$

$$= \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{i=1}^{N} \min_{z^{(i)}} ||\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}||_2^2$$

$$\hat{\mathbf{C}}, \hat{\mathbf{z}} = \underset{\mathbf{C}, \mathbf{z}}{\operatorname{argmin}} \sum_{i=1}^{N} ||\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}||_2^2$$

$$= \underset{\mathbf{C}, \mathbf{z}}{\operatorname{argmin}} J(\mathbf{C}, \mathbf{z})$$

Now apply
Block Coordinate Descent!

K-Means Algorithm

1) **Given** unlabeled feature vectors
   $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$
2) **Initialize** cluster centers $c = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$
3) **Repeat** until convergence:
   a) $\mathbf{z} \leftarrow \text{argmin}_{\mathbf{z}} J(\mathbf{C}, \mathbf{z})$
   (pick each *cluster assignment* to minimize distance)
   b) $\mathbf{C} \leftarrow \text{argmin}_{\mathbf{C}} J(\mathbf{C}, \mathbf{z})$
   (pick each *cluster center* to minimize distance)

This is an application of
Block Coordinate Descent!
The only remaining step is to figure out
what the argmins boil down to…

K-Means Algorithm

1) **Given** unlabeled feature vectors
$D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(N)}\}$
2) **Initialize** cluster centers $c = \{\mathbf{c}_1,\ldots, \mathbf{c}_K\}$
3) **Repeat** until convergence:
   a) for i in {1,…, N}
   $z^{(i)} \leftarrow \textbf{argmin}_j \, (\| \mathbf{x}^{(i)} - \mathbf{c}_j \|_2)^2$
   b) for j in {1,…,K}
   $\mathbf{c}_j \leftarrow \textbf{argmin}_{\mathbf{c}_j} \sum_{i:z^{(i)} = j} (\| \mathbf{x}^{(i)} - \mathbf{c}_j \|_2)^2$

The minimization over cluster assignments decomposes, so that we can find each $z^{(i)}$ independently of the others

Likewise, the minimization over cluster centers decomposes, so we can find each $\mathbf{c}_j$ independently

K-Means Algorithm

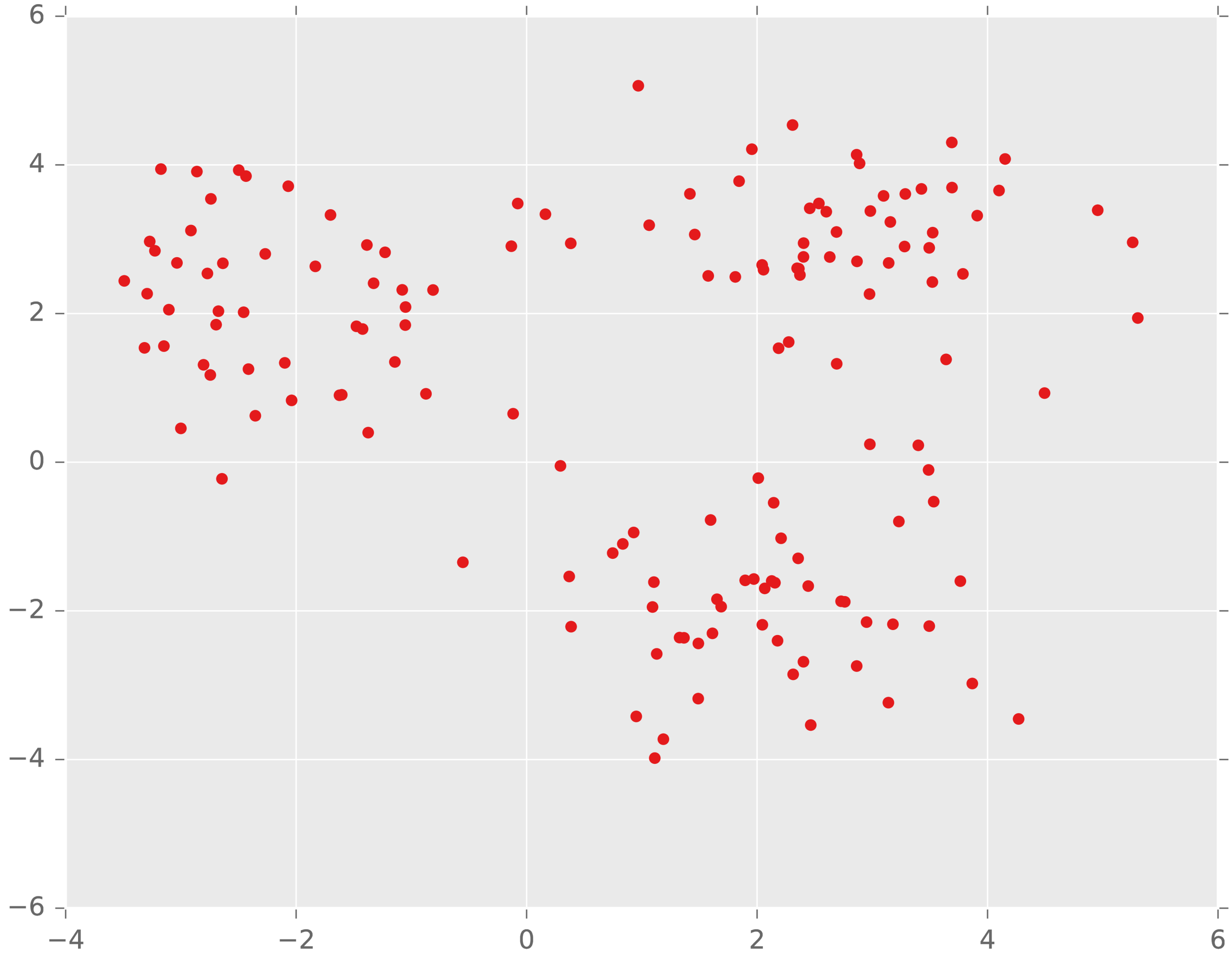
1) **Given** unlabeled feature vectors
   $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$
2) **Initialize** cluster centers $c = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$
3) **Repeat** until convergence:
   a) for i in $\{1, \ldots, N\}$
      $z^{(i)} \leftarrow$ **index** j of cluster center **nearest** to $\mathbf{x}^{(i)}$
   b) for j in $\{1, \ldots, K\}$
      $\mathbf{c}_j \leftarrow$ **mean** of **all** points assigned to cluster j

K=3 cluster centers

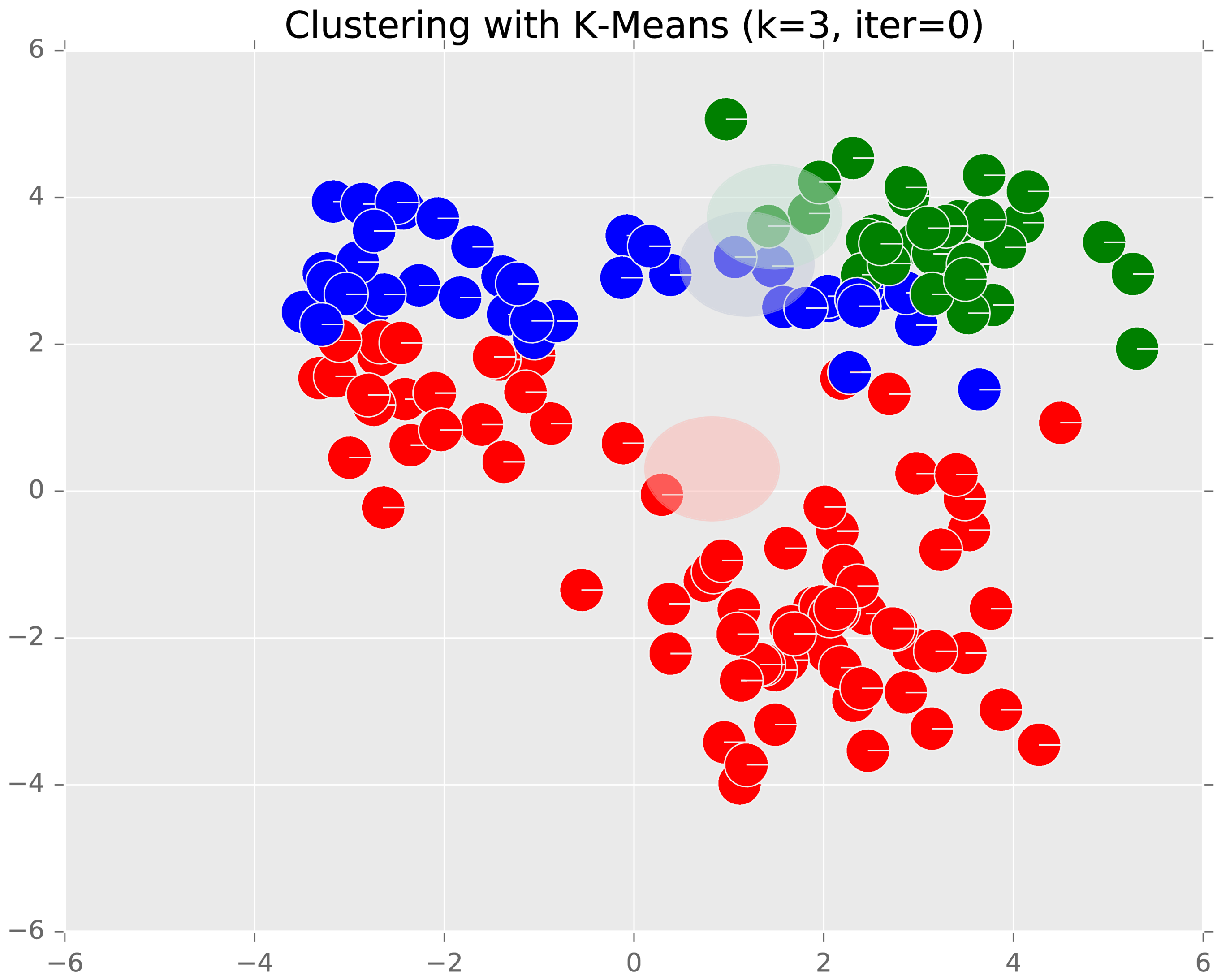
K-MEANS EXAMPLE

Example: K-Means

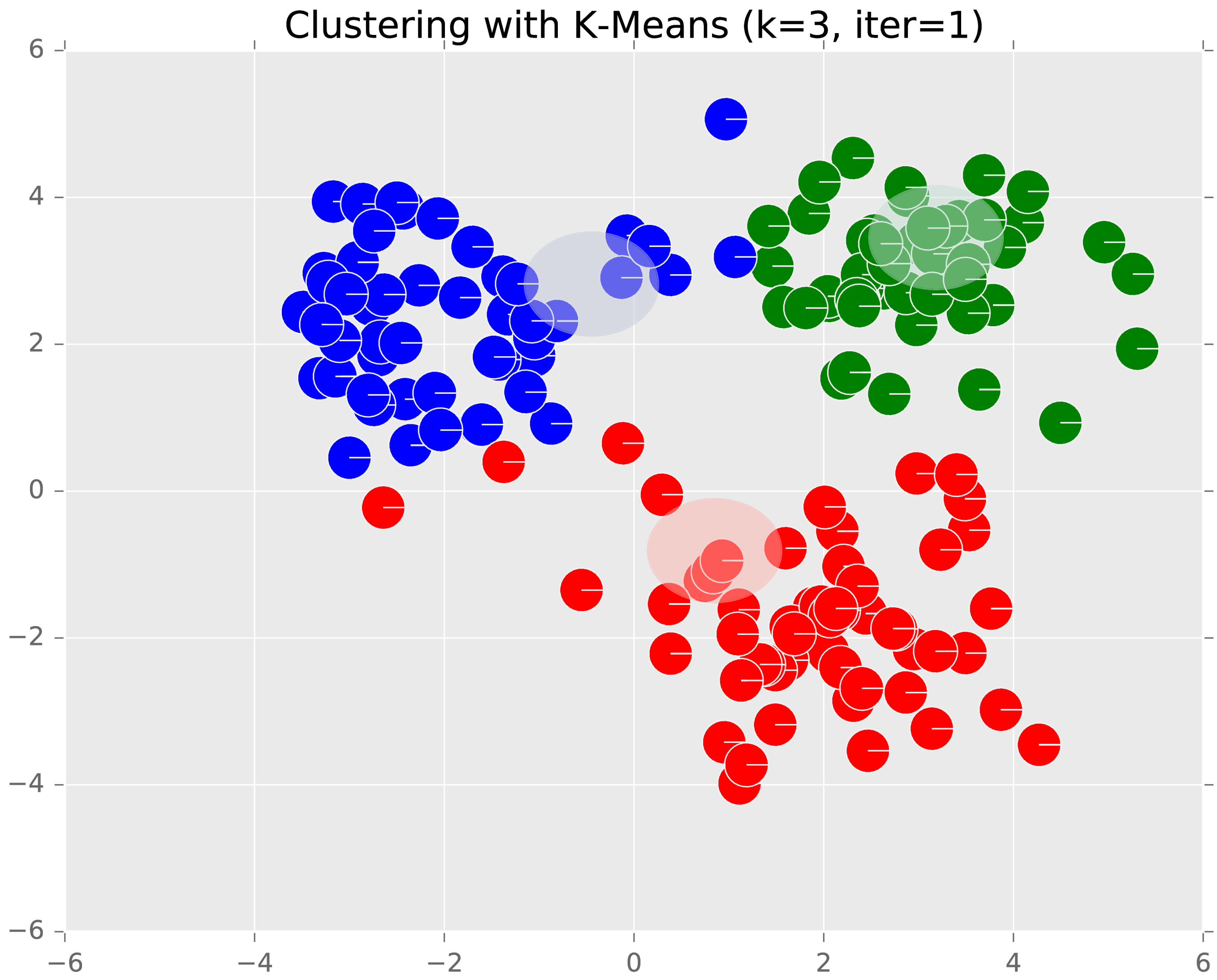
Example: K-Means



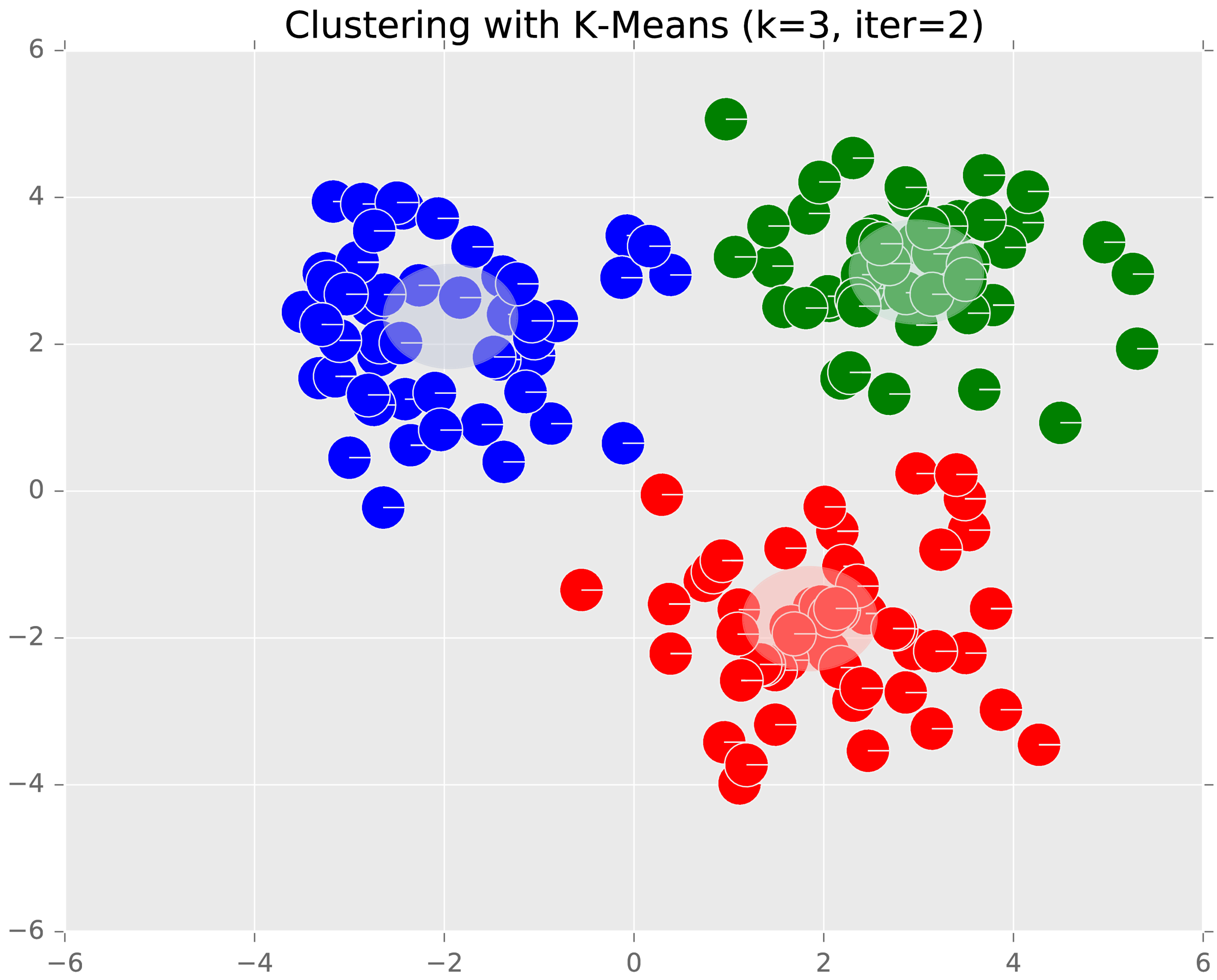
Example: K-Means



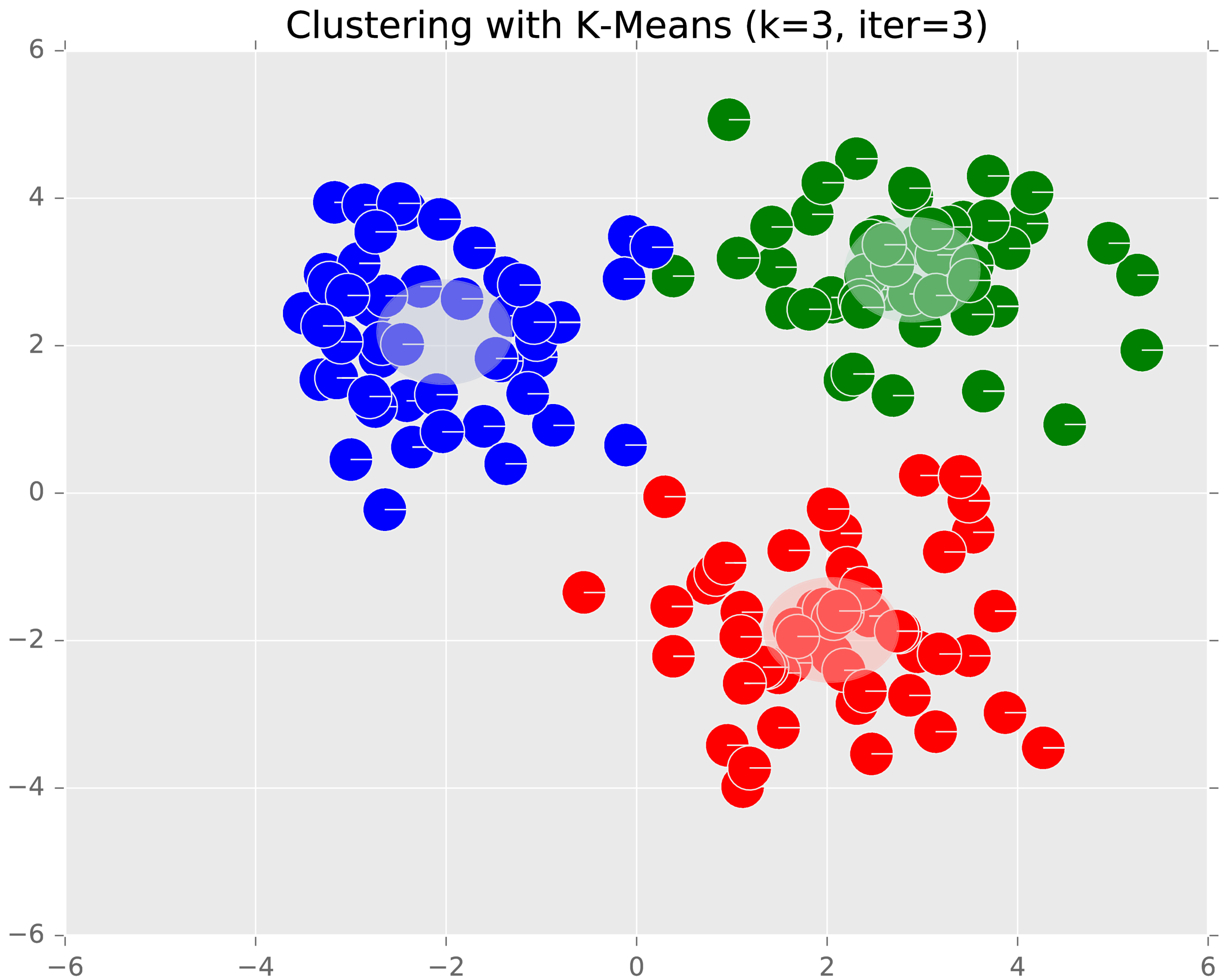
Example: K-Means



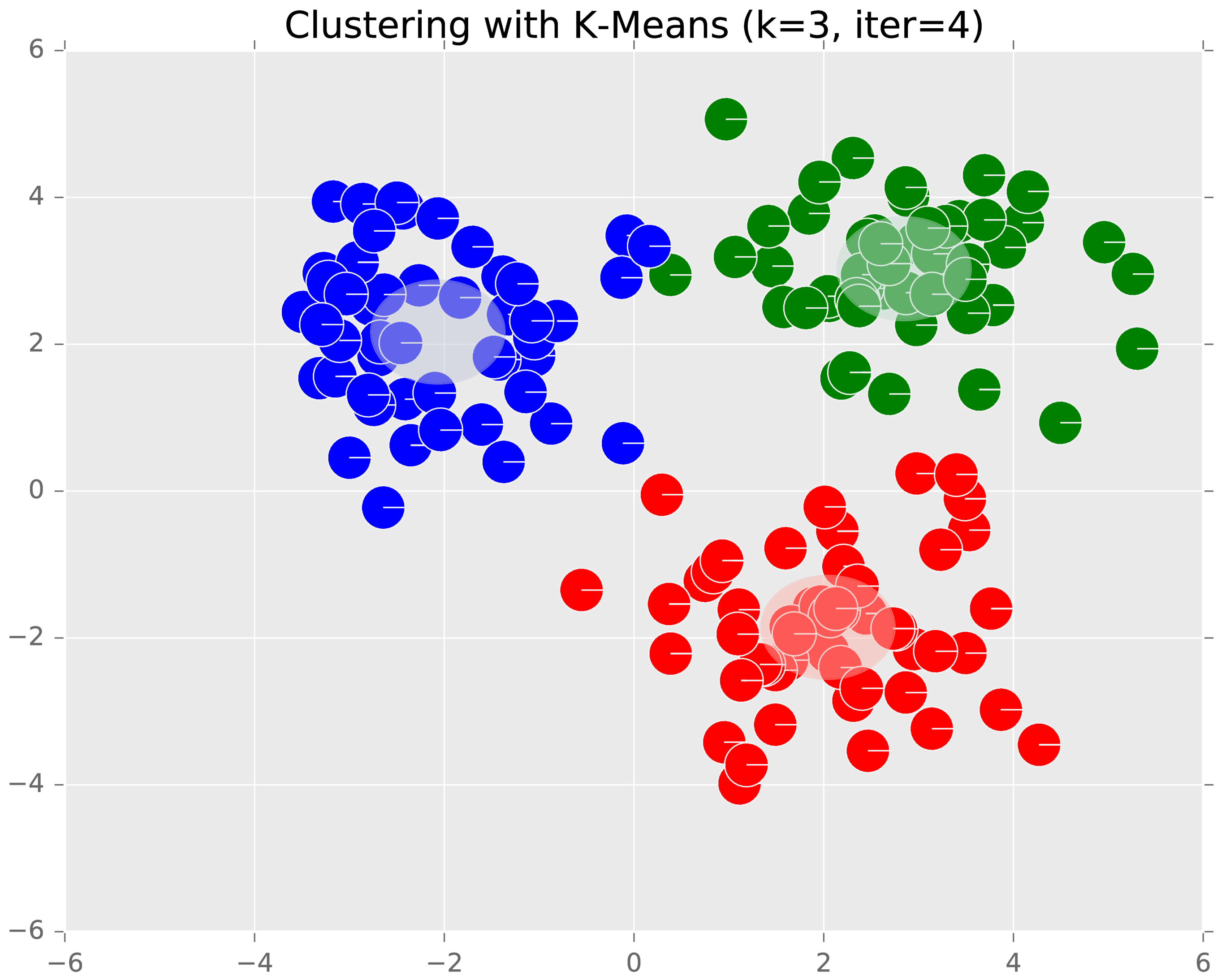
Example: K-Means



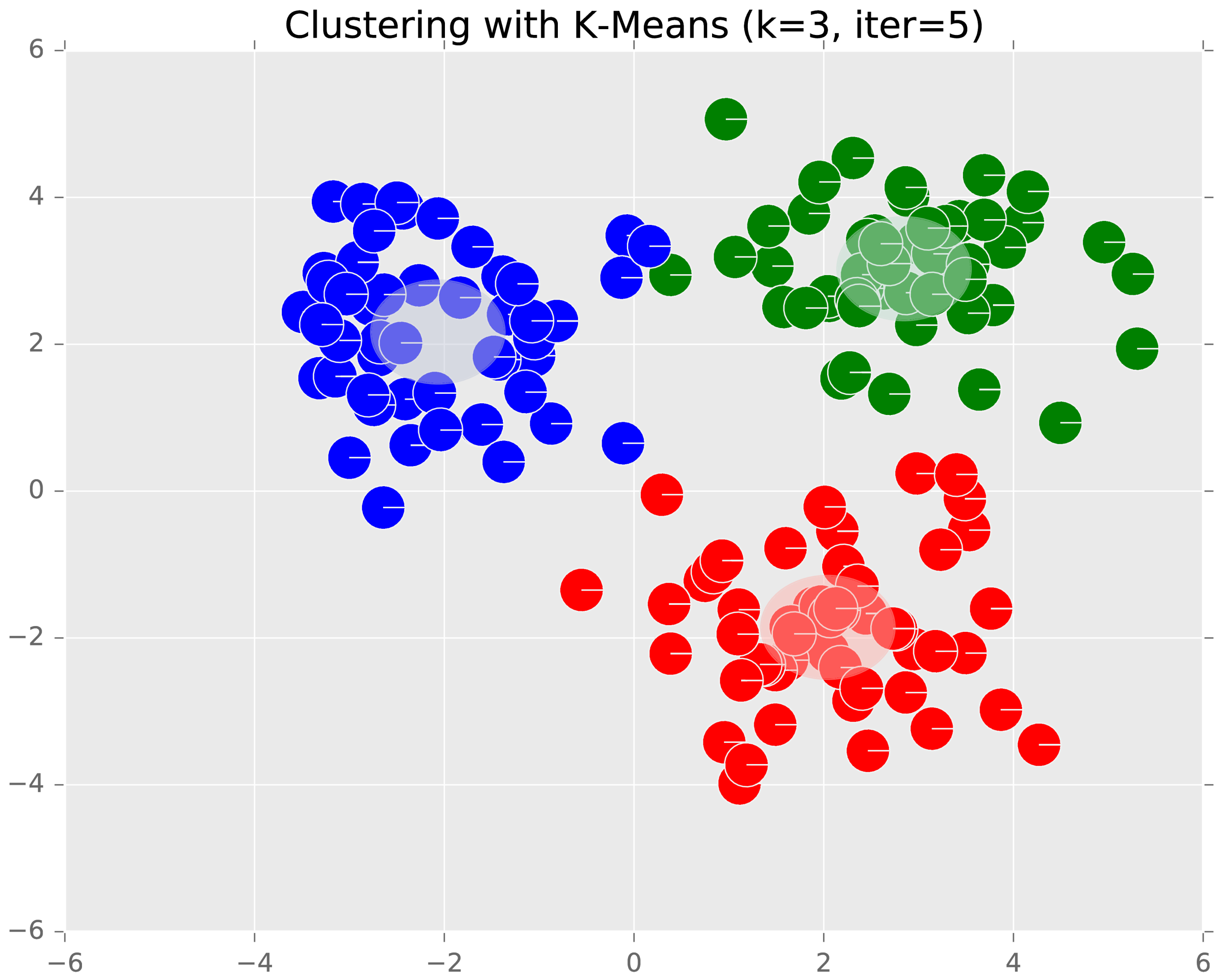
Example: K-Means



Example: K-Means



Example: K-Means

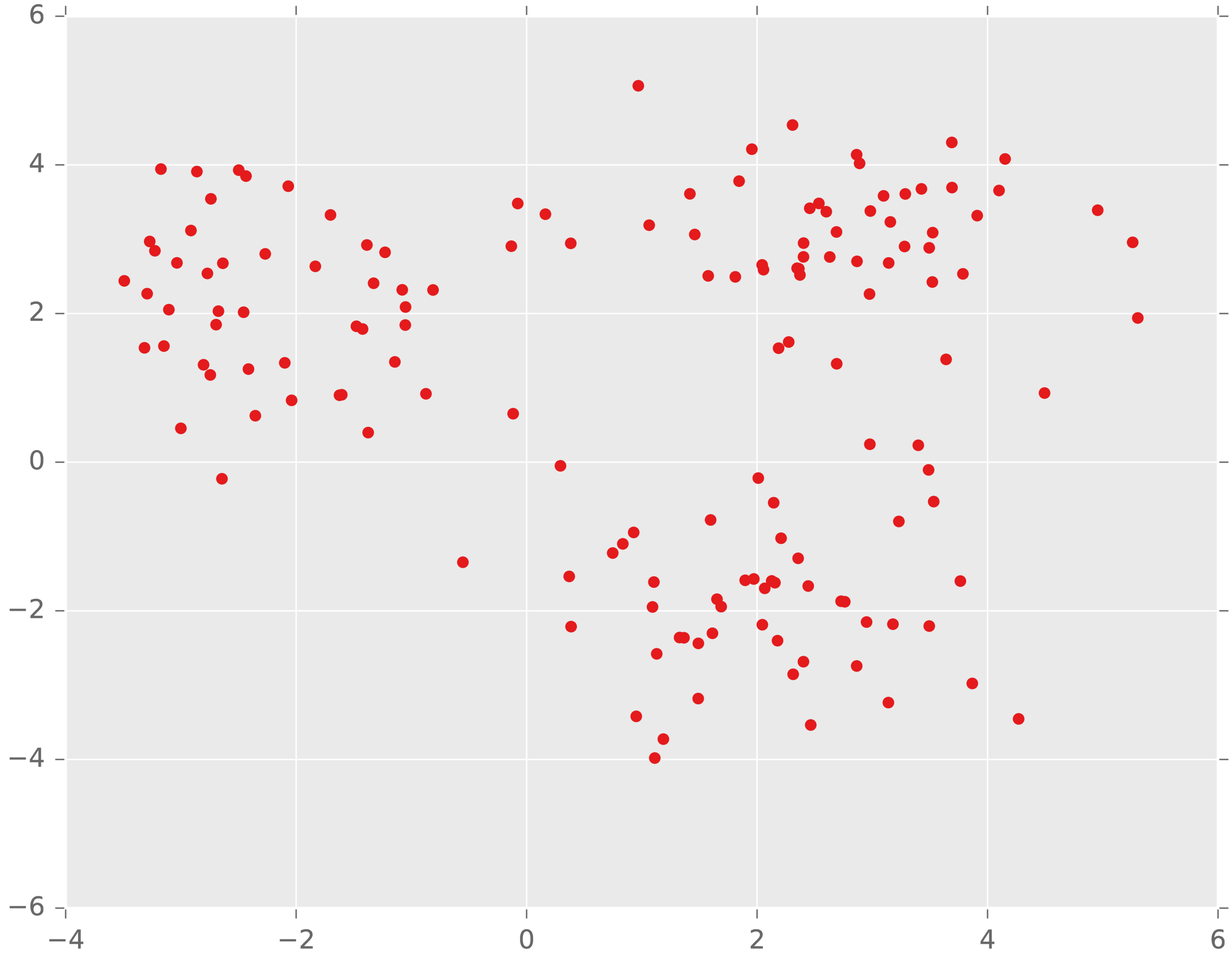


K=2 cluster centers

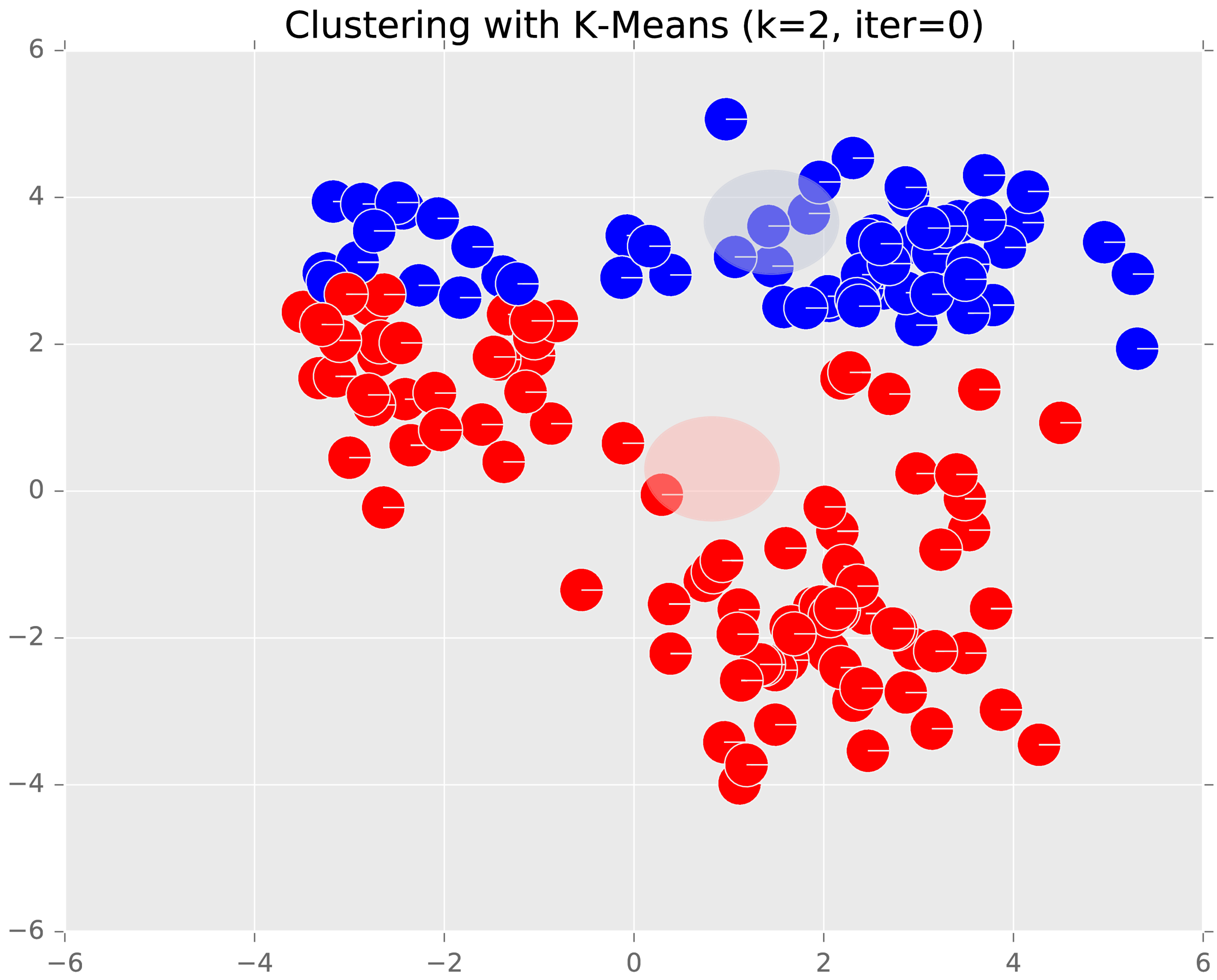
K-MEANS EXAMPLE

Example: K-Means

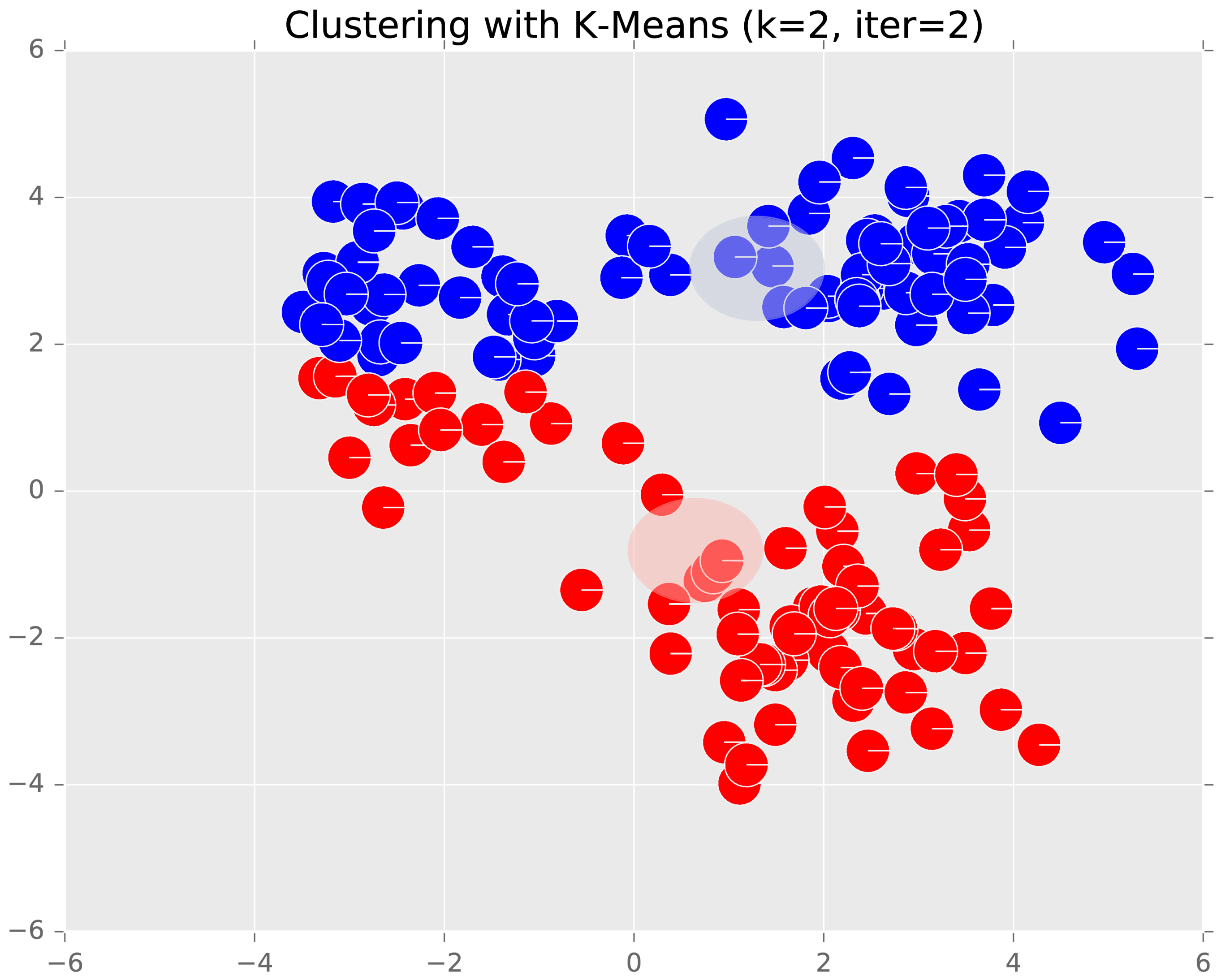
Example: K-Means



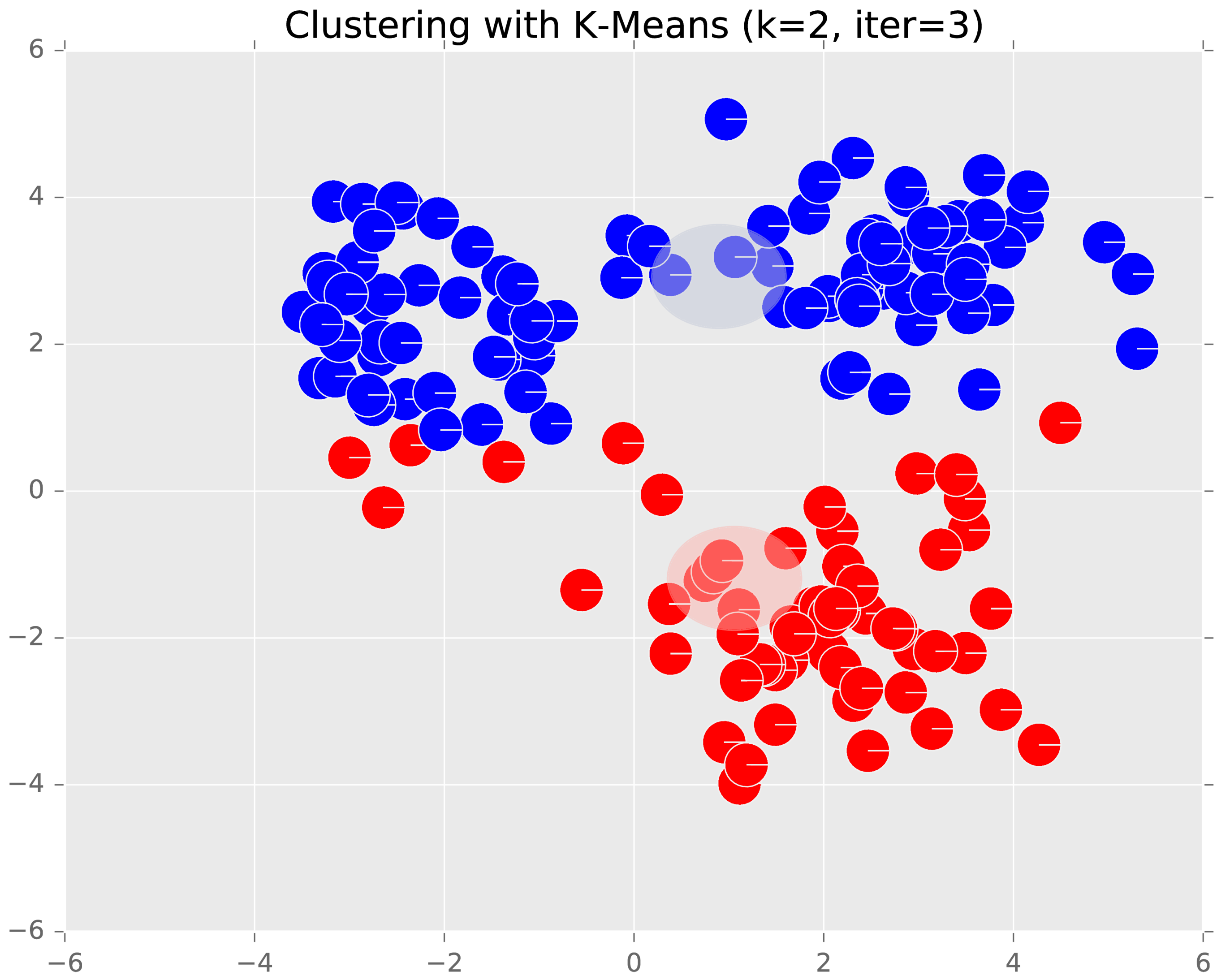
Example: K-Means



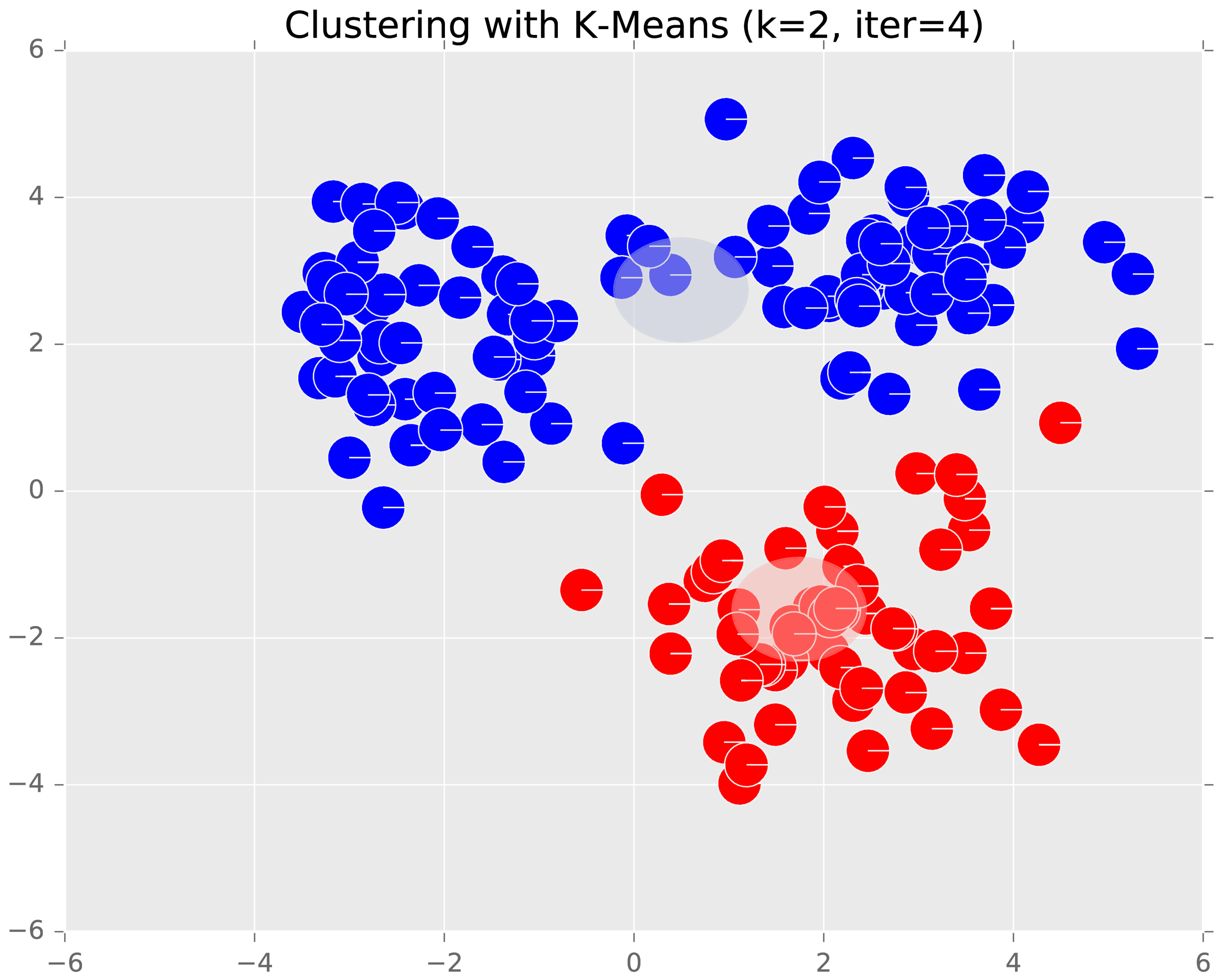
Example: K-Means



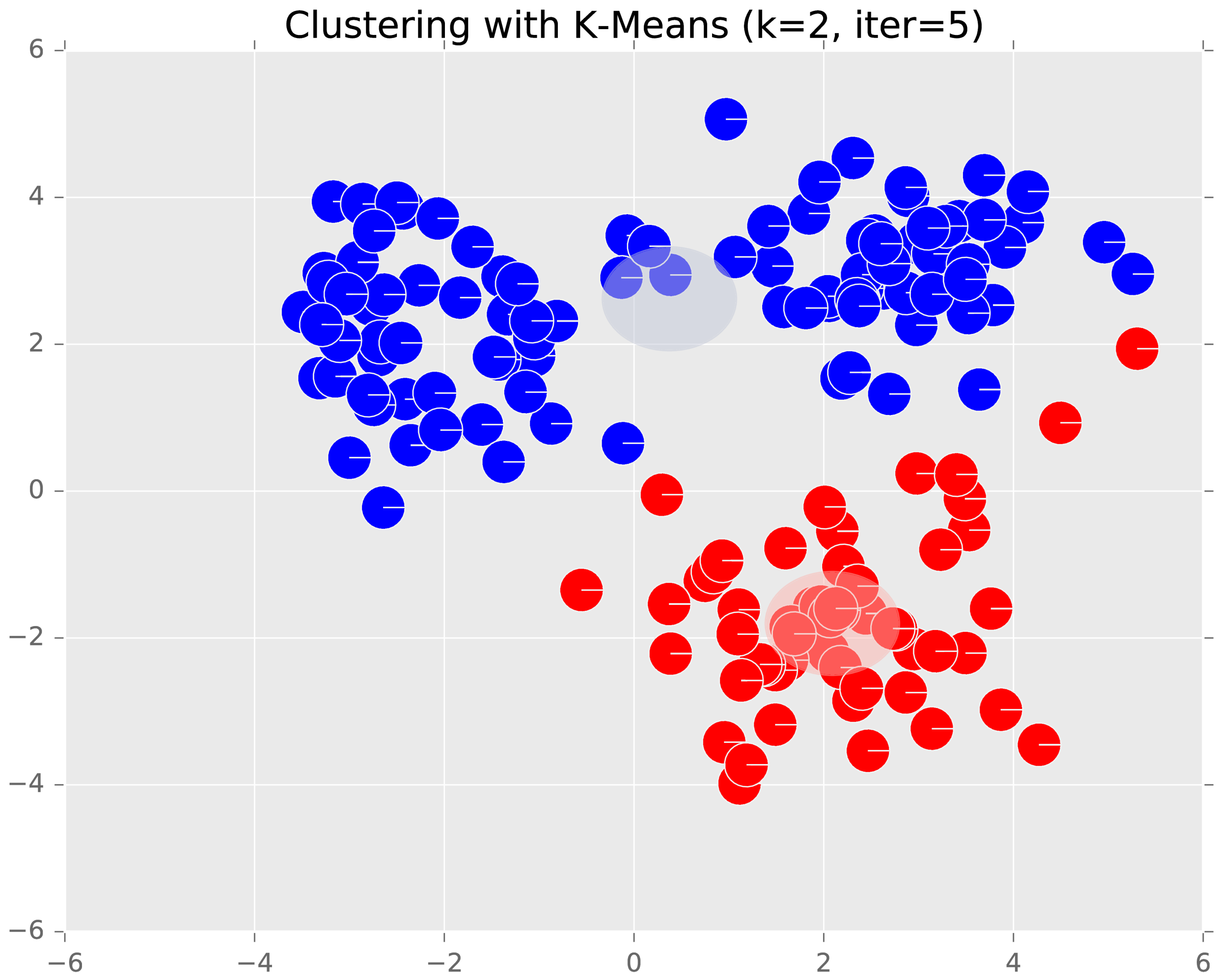
Example: K-Means



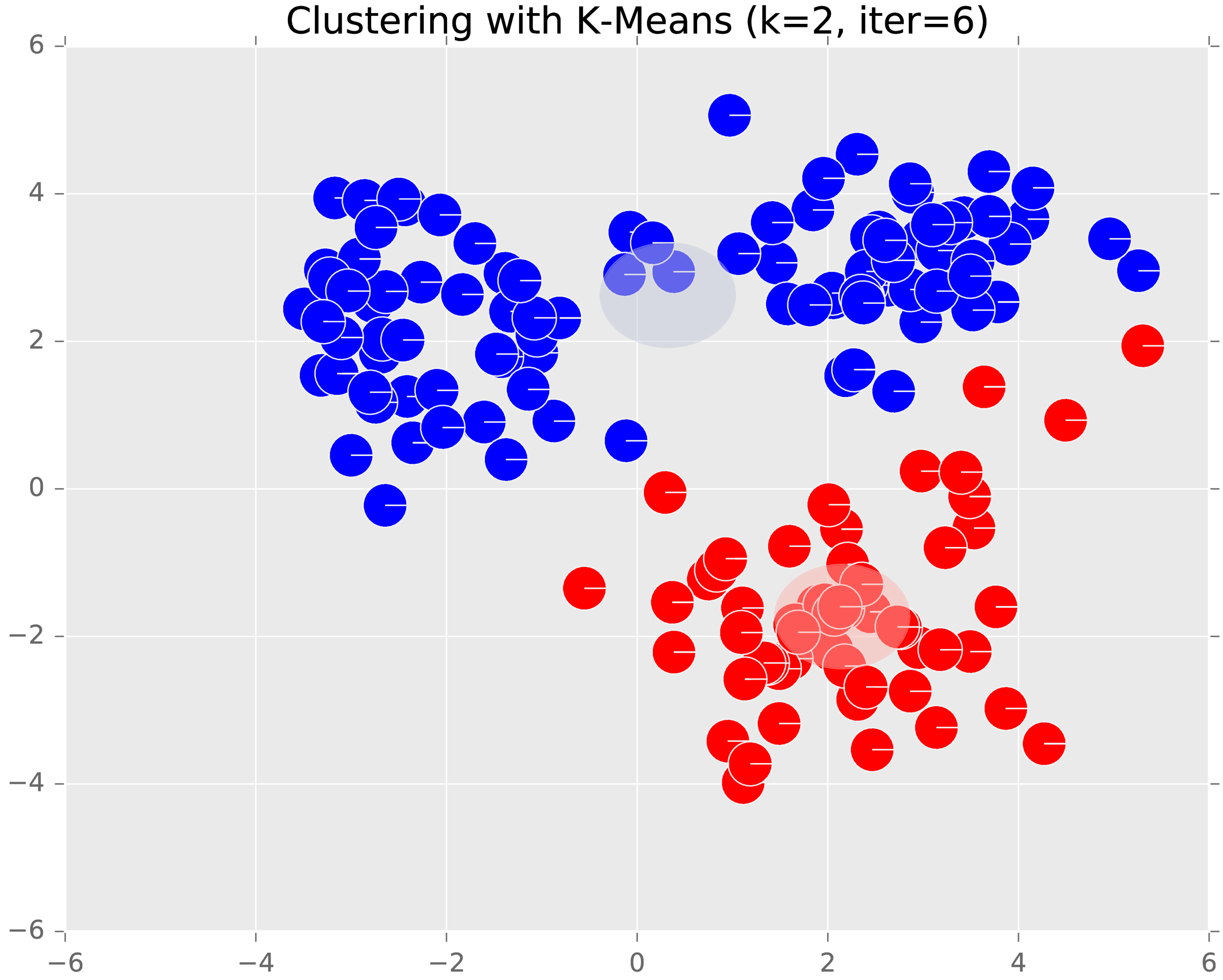
Example: K-Means



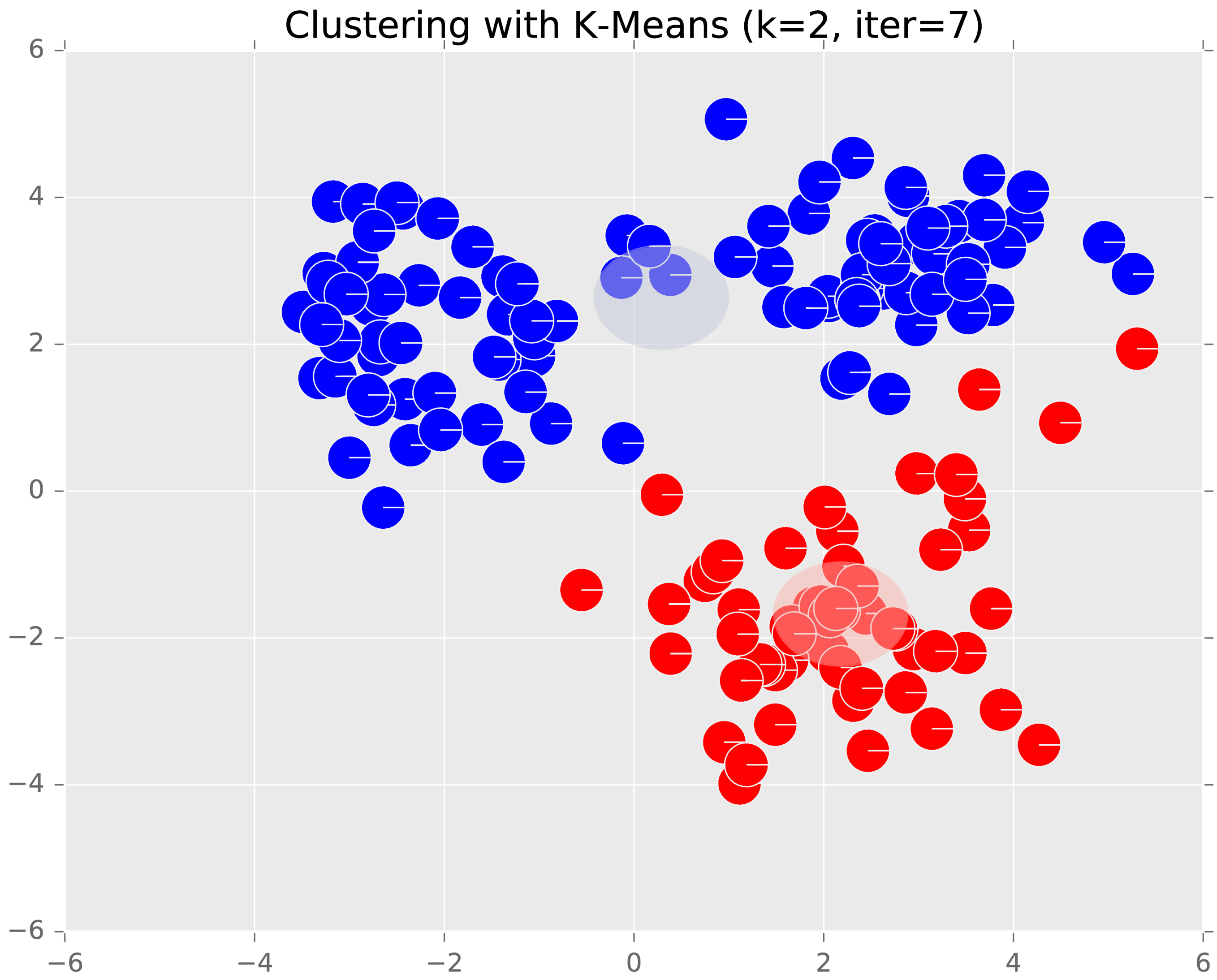
Example: K-Means



Example: K-Means



Example: K-Means



INITIALIZING K-MEANS

Initialization of K-Means

K-Means Algorithm

1) **Given** unlabeled feature vectors
   $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$
2) **Initialize** cluster centers $c = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$
3) **Rep t**
   a) for i in
   b) for j in

Remaining Question:
How should we initialize the cluster centers?

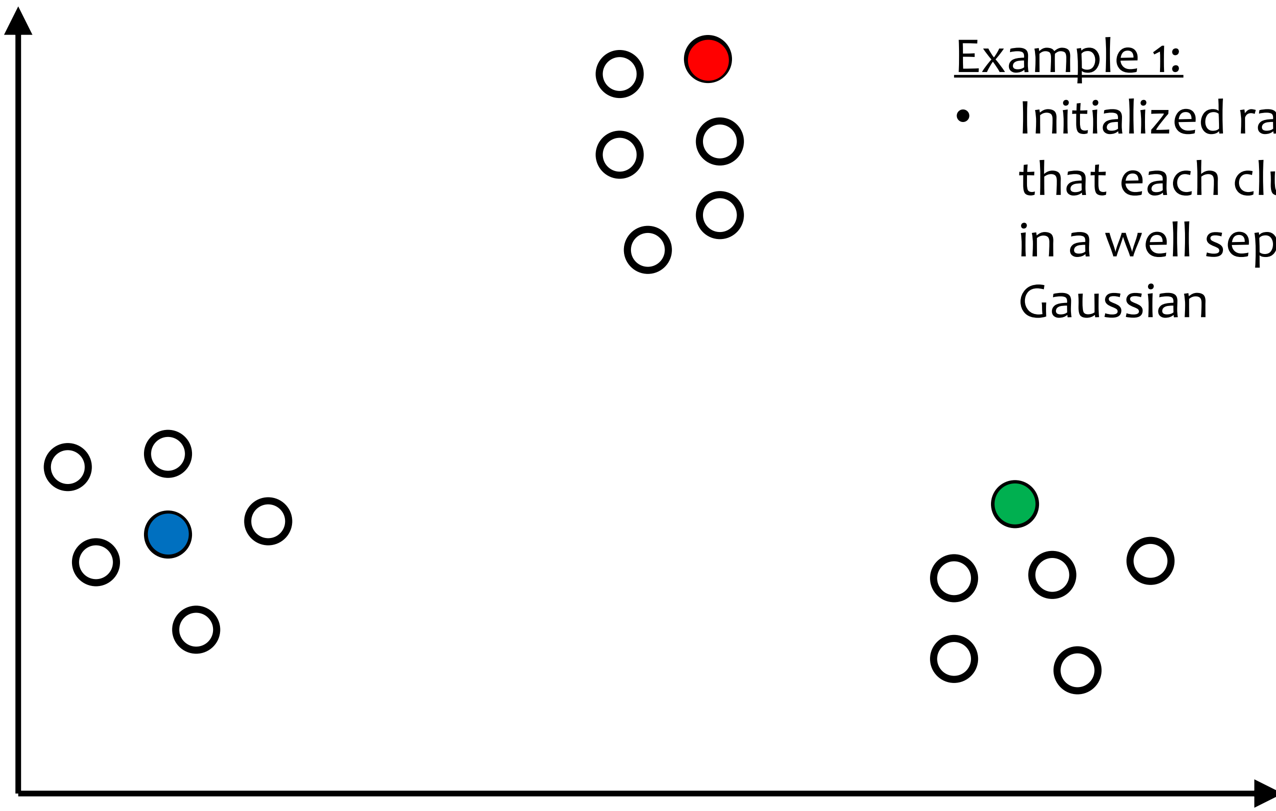
Three Solutions:

1. Random centers (picked from the data points)
2. Furthest point heuristic
3. K-Means++

Initialization for K-Means

Algorithm #1: Random Initialization
Select each cluster center uniformly at random from the data points in the training data

Observations:
Even when data comes from well-separated Gaussians…
- …sometimes works great!
- …sometimes get stuck in poor local optima.

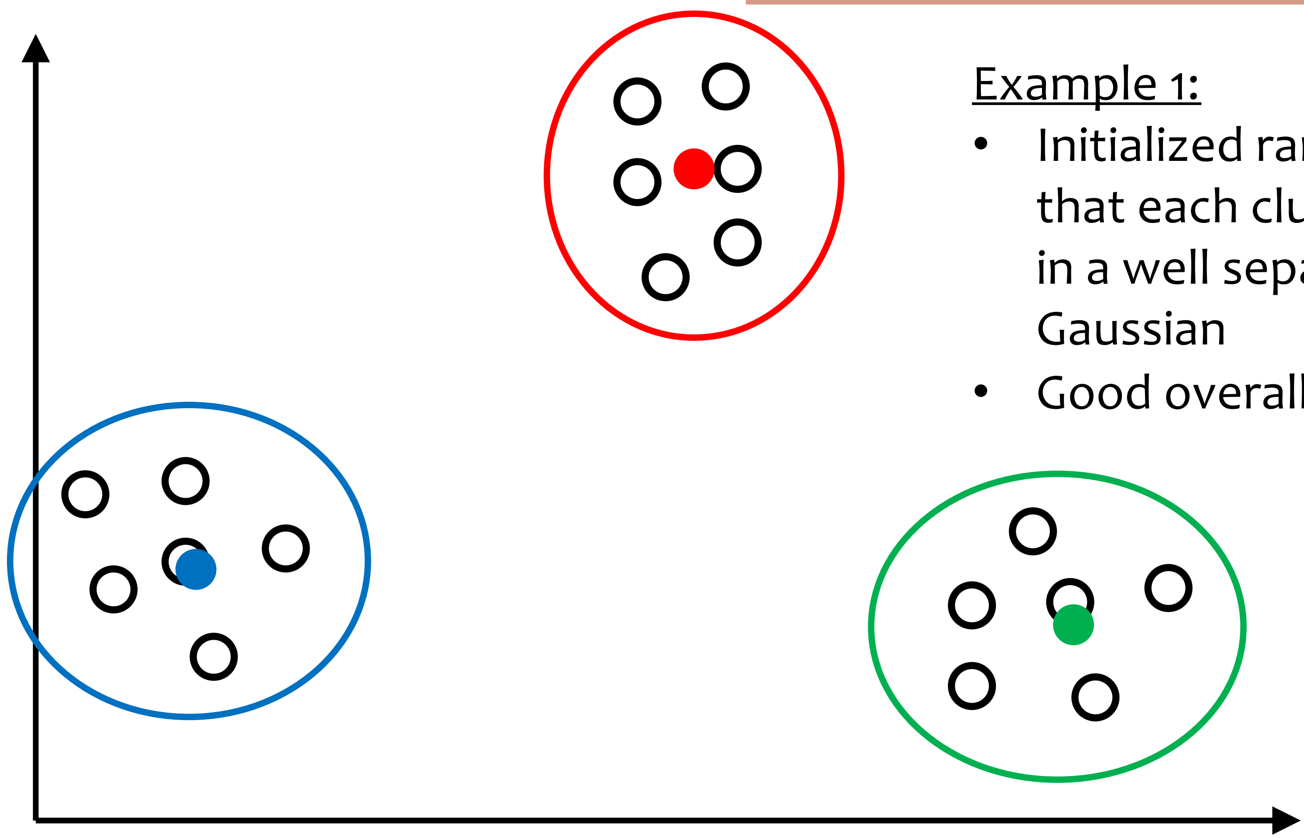


Example 1:
- Initialized randomly such that each cluster center is in a well separated Gaussian

Initialization for K-Means

<u>Algorithm #1: Random Initialization</u>
Select each cluster center uniformly at random from the data points in the training data

<u>Observations:</u>
Even when data comes from well-separated Gaussians…
- …sometimes works great!
- …sometimes get stuck in poor local optima.



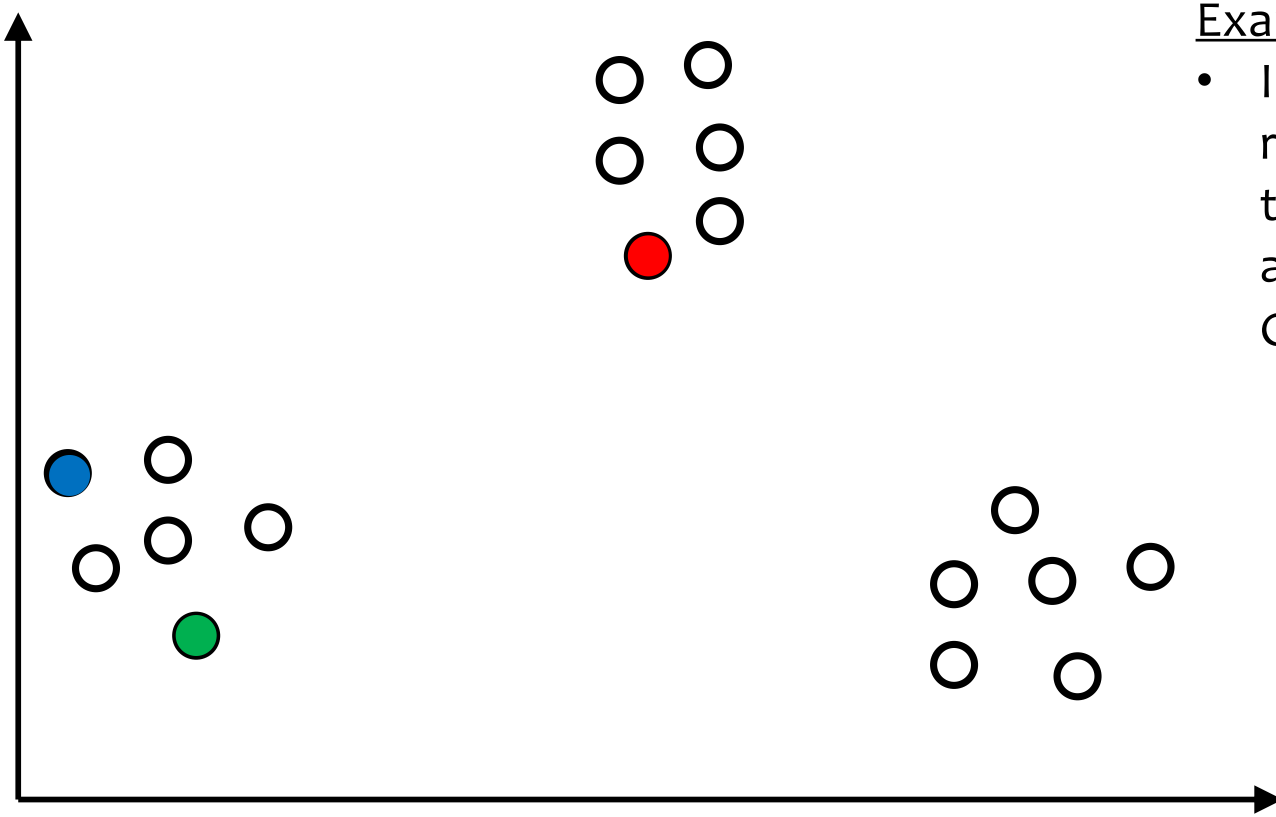
<u>Example 1:</u>
- Initialized randomly such that each cluster center is in a well separated Gaussian
- Good overall performance

Initialization for K-Means

<u>Algorithm #1: Random Initialization</u>
Select each cluster center uniformly at random from the data points in the training data

<u>Observations:</u>
Even when data comes from well-separated Gaussians…

- …sometimes works great!
- …sometimes get stuck in poor local optima.



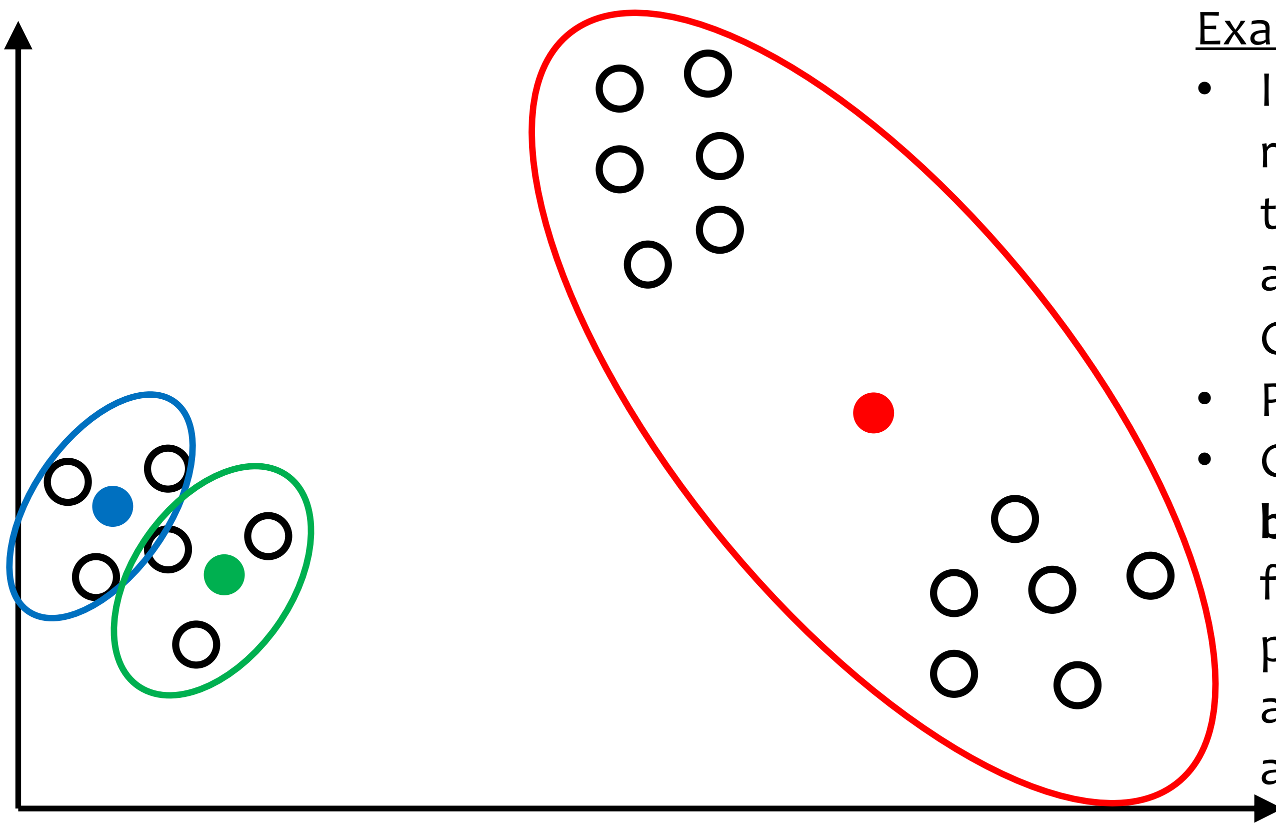
<u>Example 2:</u>

- Initialized randomly such that two centers are in the same Gaussian cluster

Initialization for K-Means

<u>Algorithm #1: Random Initialization</u>
Select each cluster center uniformly at random from the data points in the training data

<u>Observations:</u>
Even when data comes from well-separated Gaussians…
- …sometimes works great!
- …sometimes get stuck in poor local optima.



<u>Example 2:</u>
- Initialized randomly such that two centers are in the same Gaussian cluster
- Poor performance
- Can be **arbitrarily bad** (imagine the final red cluster points moving arbitrarily far away!)

Initialization for K-Means

K-Mean Performance (with Random Initialization)

If we do **random initialization**, as k increases, it becomes more likely we won't have perfectly picked one center per Gaussian in our initialization (so K-Means will output a bad solution).

- For k equal-sized Gaussians,

$$\Pr[\text{each initial center is in a different Gaussian}] \approx \frac{k!}{k^k} \approx \frac{1}{e^k}$$

- Becomes unlikely as k gets large.

Slide courtesy of Nina Balcan

Initialization for K-Means

<u>Algorithm #2: Furthest Point Heuristic</u>

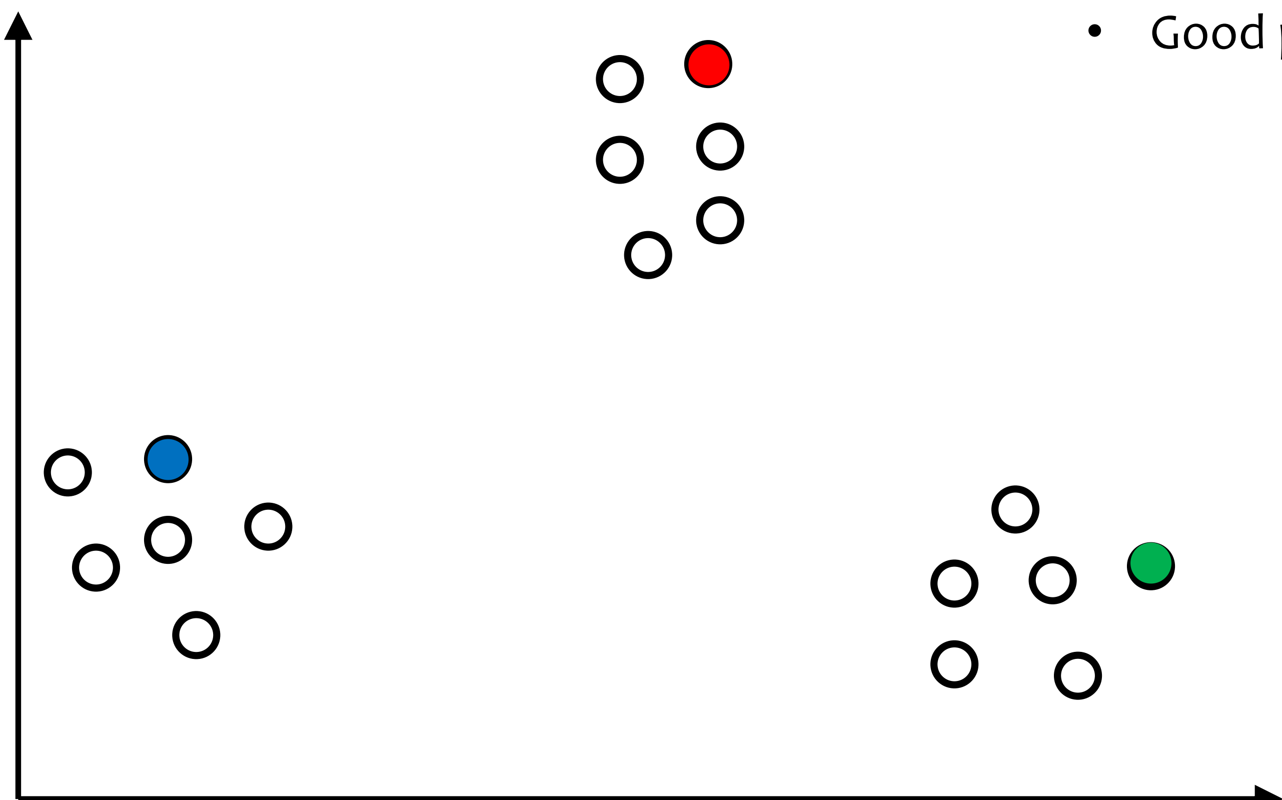
1. Pick the first cluster center c_1 **randomly**
2. Pick each subsequent center c_j so that it is **as far as possible** from the previously chosen centers $c_1, c_2, \ldots, c_{j-1}$

<u>Observations:</u>

- Solves the problem with Gaussian data
- But outliers pose a new problem!

<u>Example 1:</u>

- No outliers
- Good performance



Initialization for K-Means

Algorithm #2: Furthest Point Heuristic

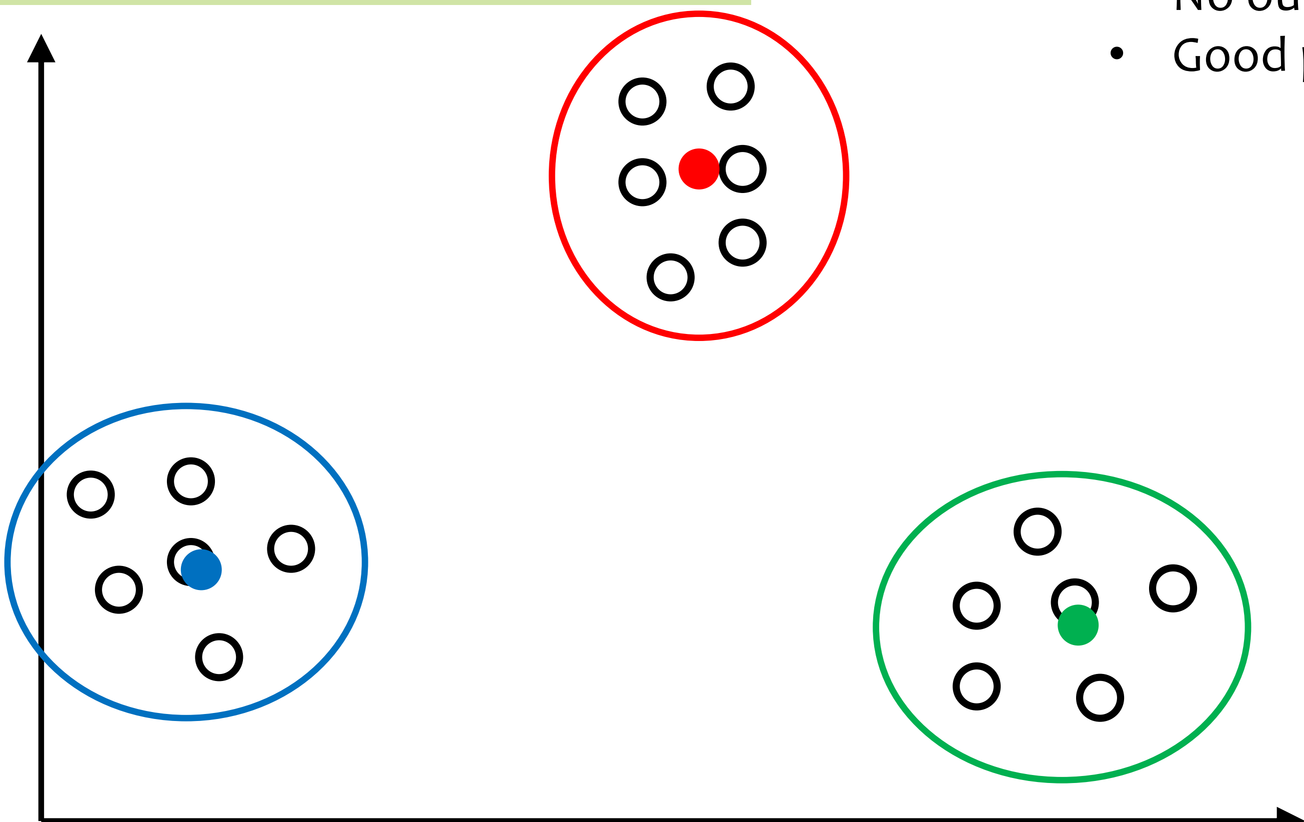
1. Pick the first cluster center $\mathbf{c}_1$ **randomly**
2. Pick each subsequent center $\mathbf{c}_j$ so that it is **as far as possible** from the previously chosen centers $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{j-1}$

Observations:

- Solves the problem with Gaussian data
- But outliers pose a new problem!

Example 1:

- No outliers
- Good performance



Initialization for K-Means

Algorithm #2: Furthest Point Heuristic

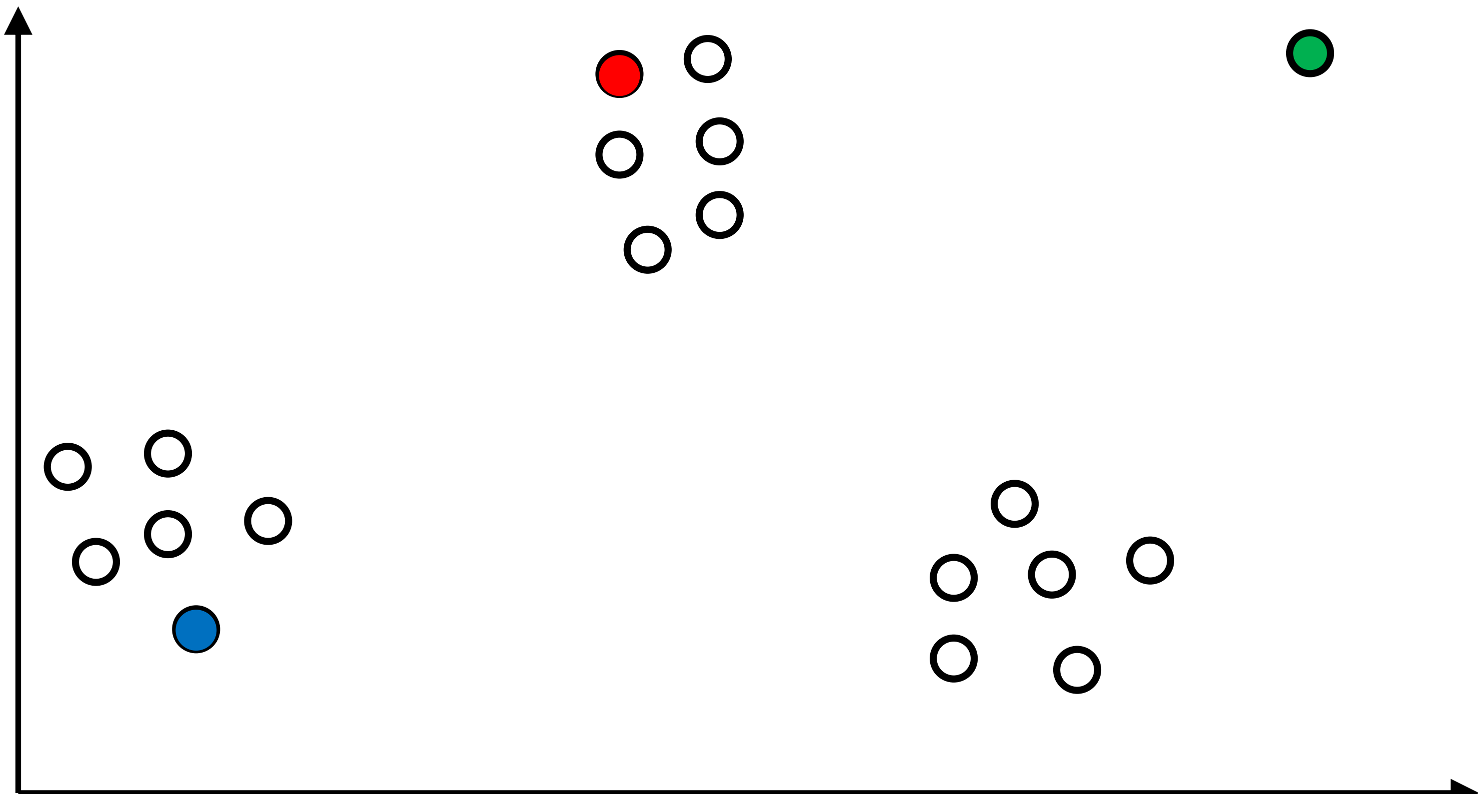
1. Pick the first cluster center $\mathbf{c}_1$ **randomly**
2. Pick each subsequent center $\mathbf{c}_j$ so that it is **as far as possible** from the previously chosen centers $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{j-1}$

Observations:

- Solves the problem with Gaussian data
- But outliers pose a new problem!

Example 2:

- One outlier throws off the algorithm
- Poor performance



Initialization for K-Means

<u>Algorithm #2: Furthest Point Heuristic</u>

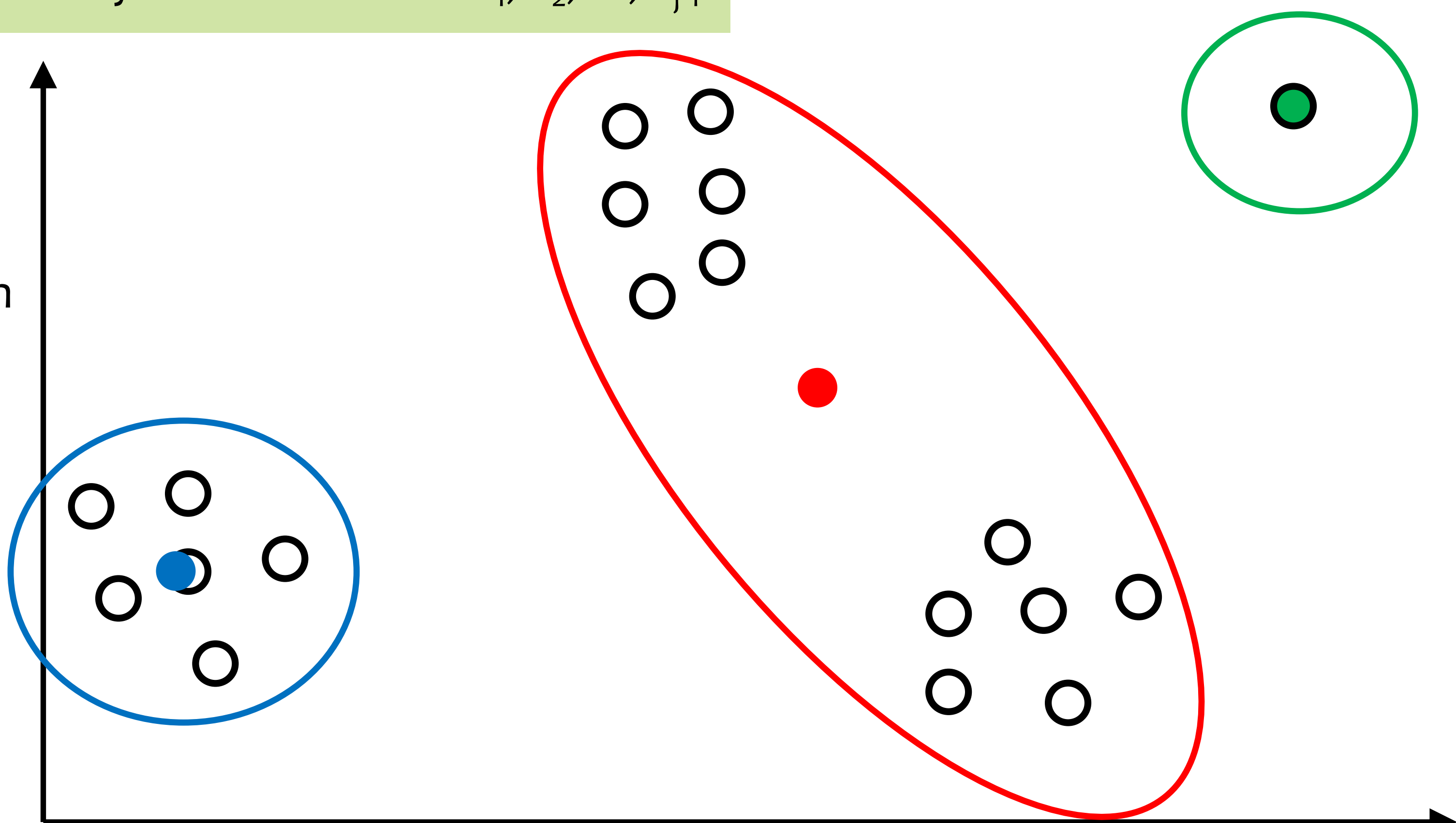
1. Pick the first cluster center $\mathbf{c}_1$ **randomly**
2. Pick each subsequent center $\mathbf{c}_j$ so that it is **as far as possible** from the previously chosen centers $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{j-1}$

<u>Observations:</u>

- Solves the problem with Gaussian data
- But outliers pose a new problem!

<u>Example 2:</u>

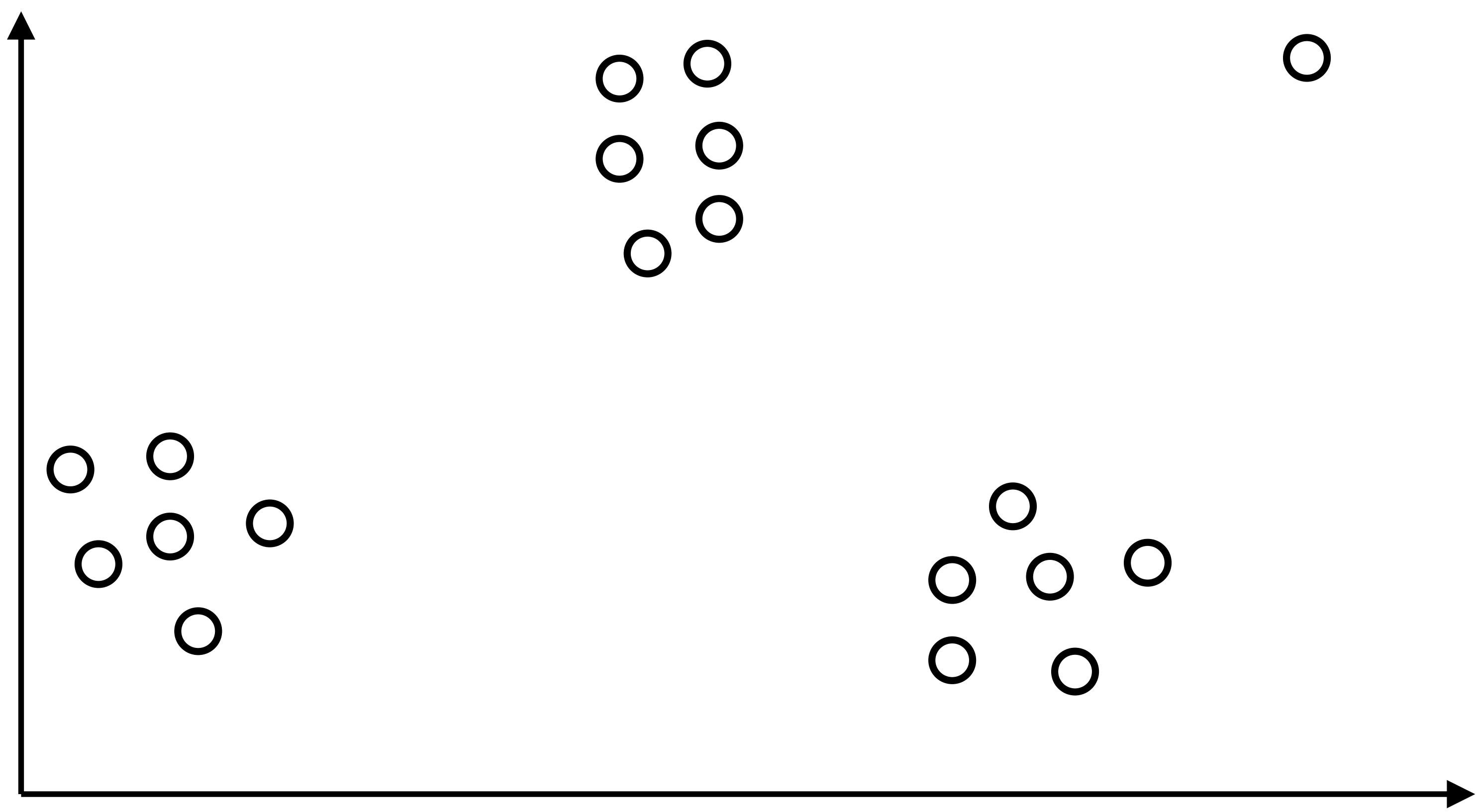
- One outlier throws off the algorithm
- Poor performance



Initialization for K-Means

Algorithm #3: K-Means++

- Let $\mathrm{D}(\mathbf{x})$ be the distance between a point x and its nearest center. Chose the next center proportional to $\mathrm{D}^2(\mathbf{x})$.



Initialization for K-M

i	D(x)	$D^2(x)$	$P(c_2 = x^{(i)})$
1	3	9	9/137
2	2	4	4/137
…			
7	4	16	16/137
…			
N	3	9	9/137
	Sum:	137	1.0

<u>Algorithm #3: K-Means++</u>

- Let $D(\mathbf{x})$ be the distance between a point x and its nearest center. Chose the next center proportional to $D^2(\mathbf{x})$.

- Choose $\mathbf{c_1}$ at random.
- For $j = 2, \ldots, K$
  - Pick $\mathbf{c_j}$ among $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}$ according to the distribution

$$\mathbf{P(c_j = x^{(i)}) \propto \min_{j'<j} \left\|x^{(i)} - c_{j'}\right\|^2} \quad D^2(\mathbf{x^i})$$

Theorem: K-Means++ always attains an $O(\log k)$ approximation to optimal K-Means solution in expectation.

Slide adapted from Nina Balcan

Initialization for K-M

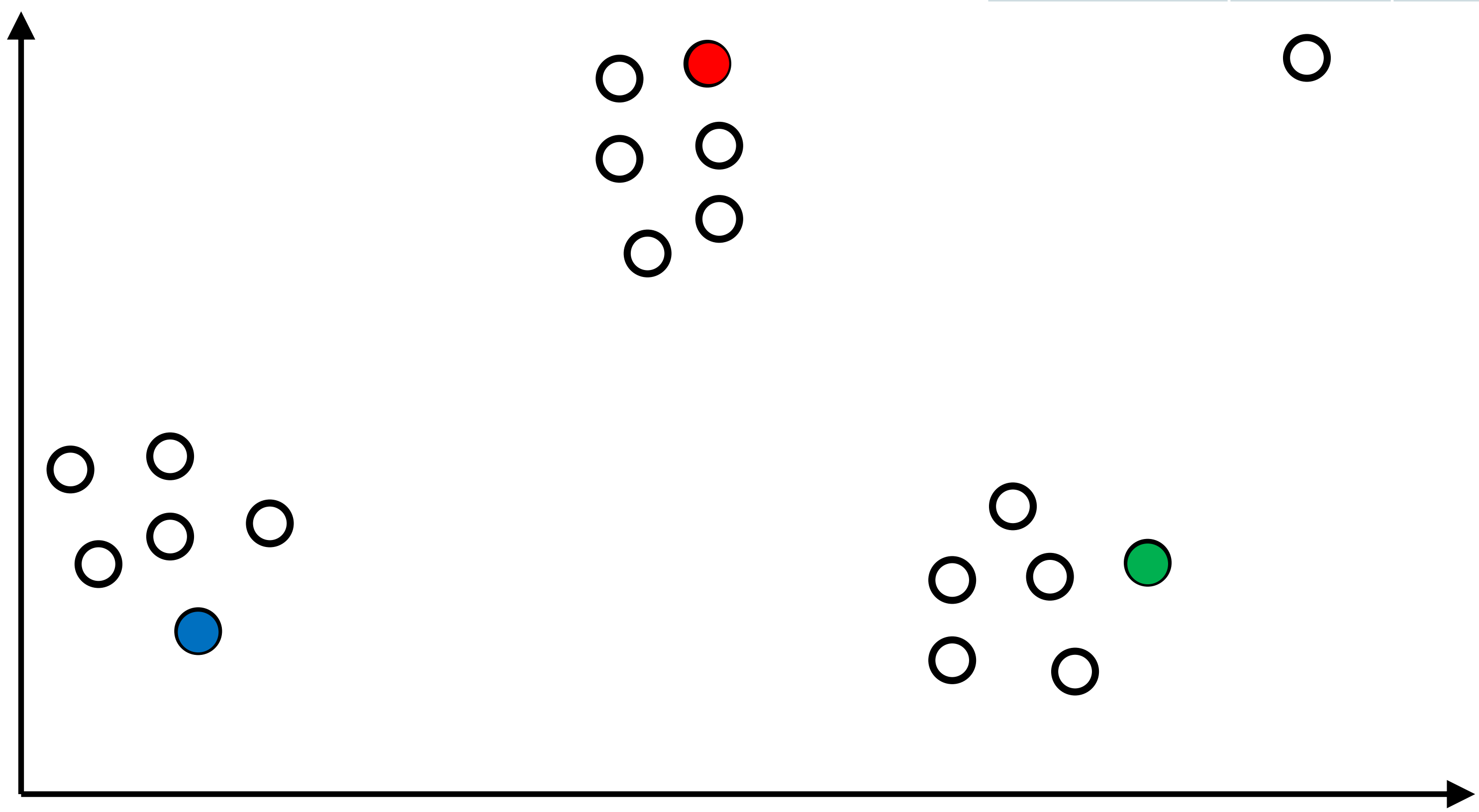
i	D(x)	$D^2(x)$	$P(c_2 = x^{(i)})$
1	3	9	9/137
2	2	4	4/137
…			
7	4	16	16/137
…			
N	3	9	9/137
	Sum:	137	1.0

<u>Algorithm #3: K-Means++</u>

- Let $D(\mathbf{x})$ be the distance between a point x and its nearest center. Chose the next center proportional to $D^2(\mathbf{x})$.

<u>Example 1:</u>

- One outlier
- Good performance



Initialization for K-M

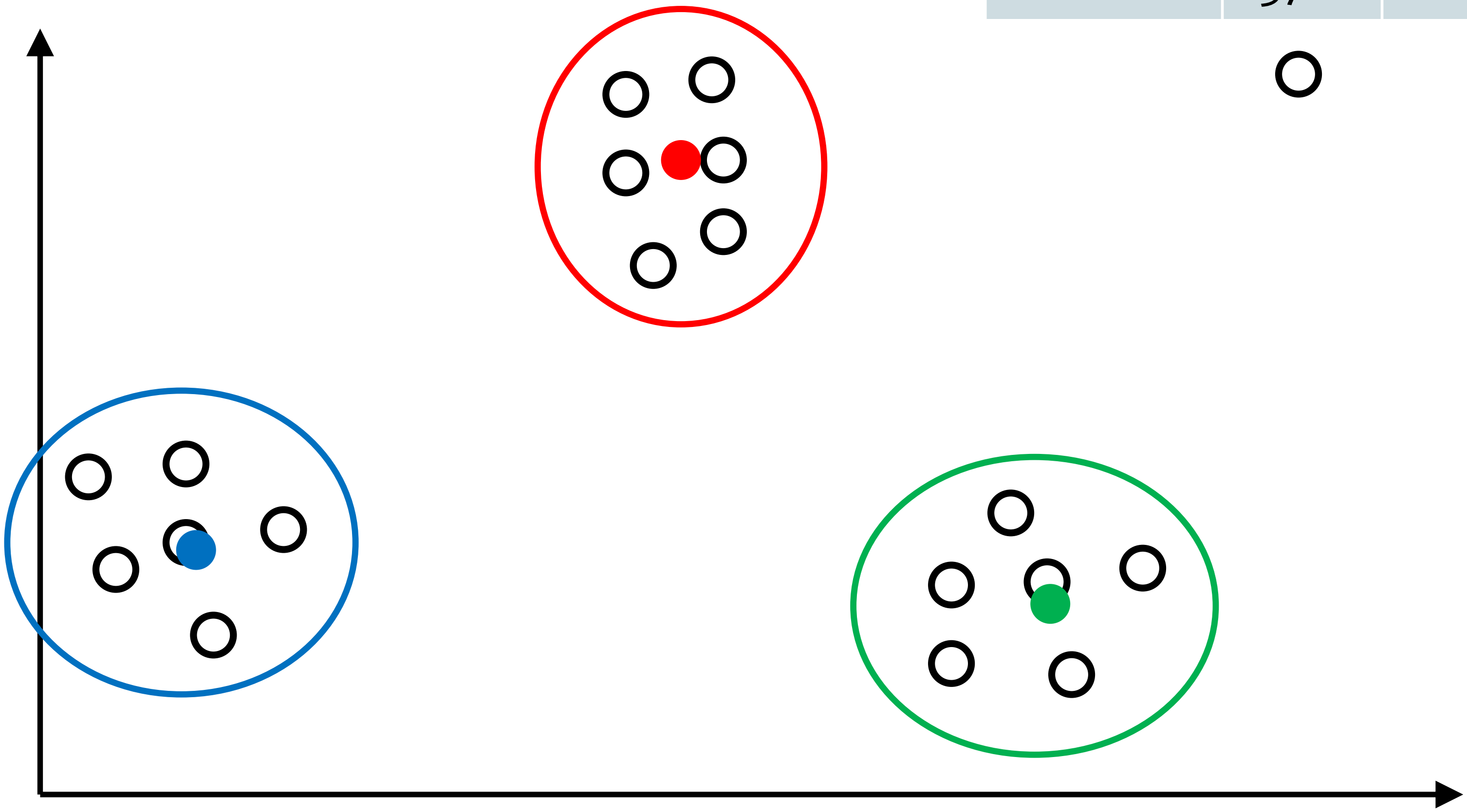
i	D(x)	$D^2(x)$	$P(c_2 = x^{(i)})$
1	3	9	9/137
2	2	4	4/137
…			
7	4	16	16/137
…			
N	3	9	9/137
	Sum:	137	1.0

<u>Algorithm #3: K-Means++</u>

- Let $D(\mathbf{x})$ be the distance between a point x and its nearest center. Chose the next center proportional to $D^2(\mathbf{x})$.

<u>Example 1:</u>

- One outlier
- Good performance



Initialization for K-Means

Algorithm #3: K-Means++

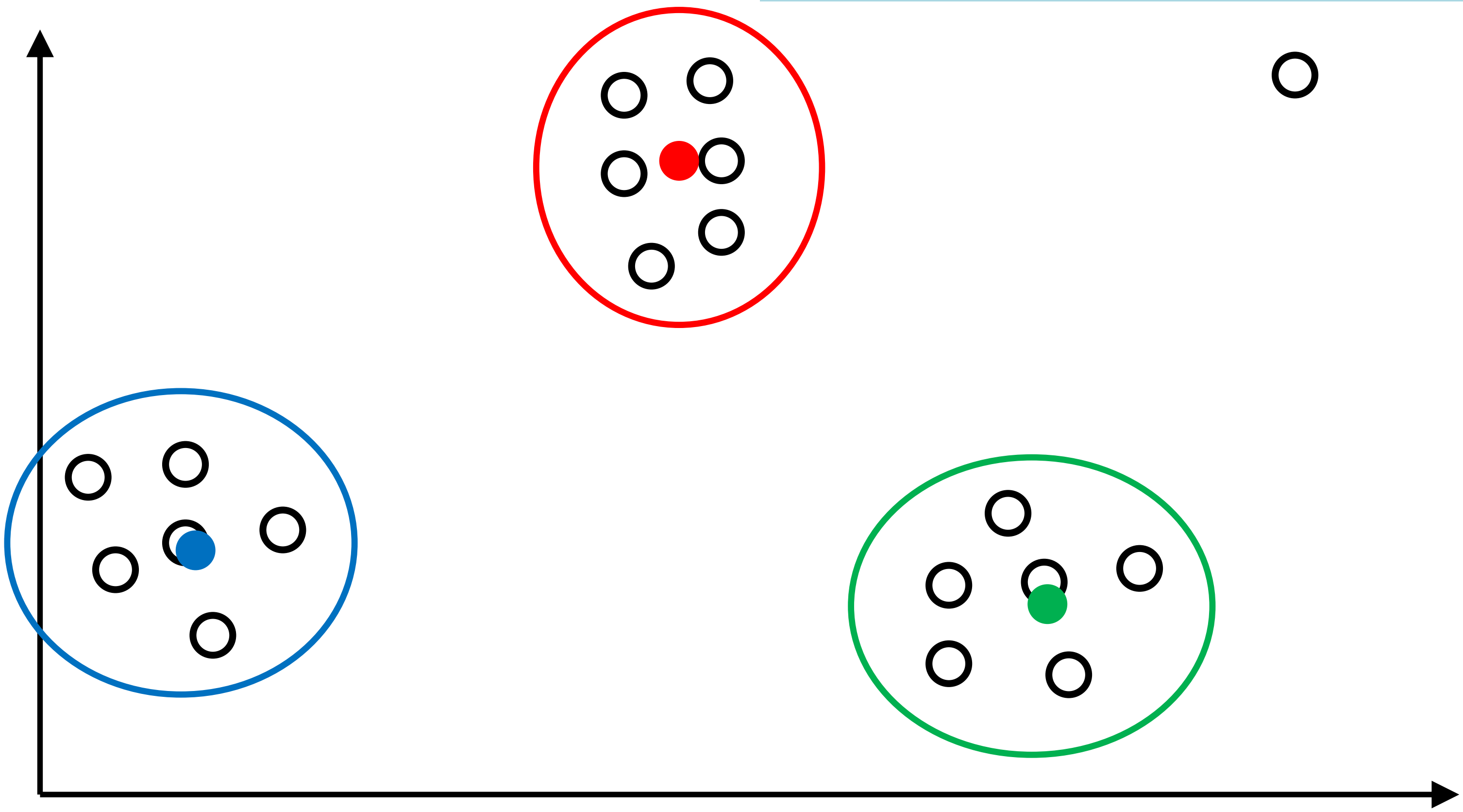
- Let $\mathrm{D}(\mathbf{x})$ be the distance between a point x and its nearest center. Chose the next center proportional to $\mathrm{D}^2(\mathbf{x})$.

Observations:

- Interpolates between random and farthest point initialization
- Solves the problem with Gaussian data
- **And** solves the outlier problem

Example 1:

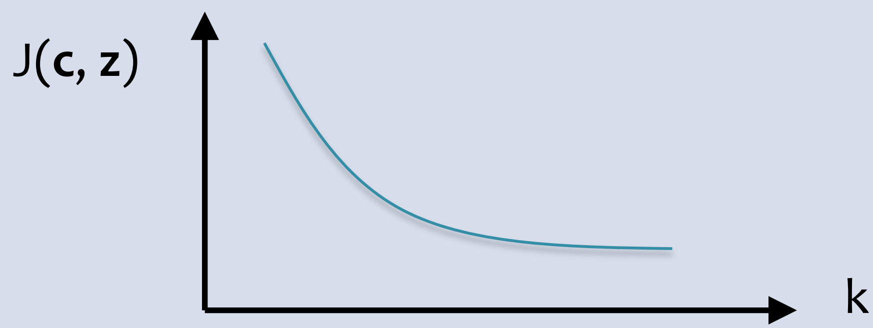
- One outlier
- Good performance



Q&A

Q: In k-Means, since we don't have a validation set, how do we pick k?

A: Look at the training objective function as a function of k and pick the value at the "elbo" of the curve.



Q: What if our random initialization for k-Means gives us poor performance?

A: Do **random restarts**: that is, run k-means from scratch, say, 10 times and pick the run that gives the lowest training objective function value.

The objective function is **nonconvex**, so we're just looking for the best local minimum.

Learning Objectives

K-Means

You should be able to…

1. Distinguish between coordinate descent and block coordinate descent
2. Define an objective function that gives rise to a "good" clustering
3. Apply block coordinate descent to an objective function preferring each point to be close to its nearest objective function to obtain the K-Means algorithm
4. Implement the K-Means algorithm
5. Connect the non-convexity of the K-Means objective function with the (possibly) poor performance of random initialization

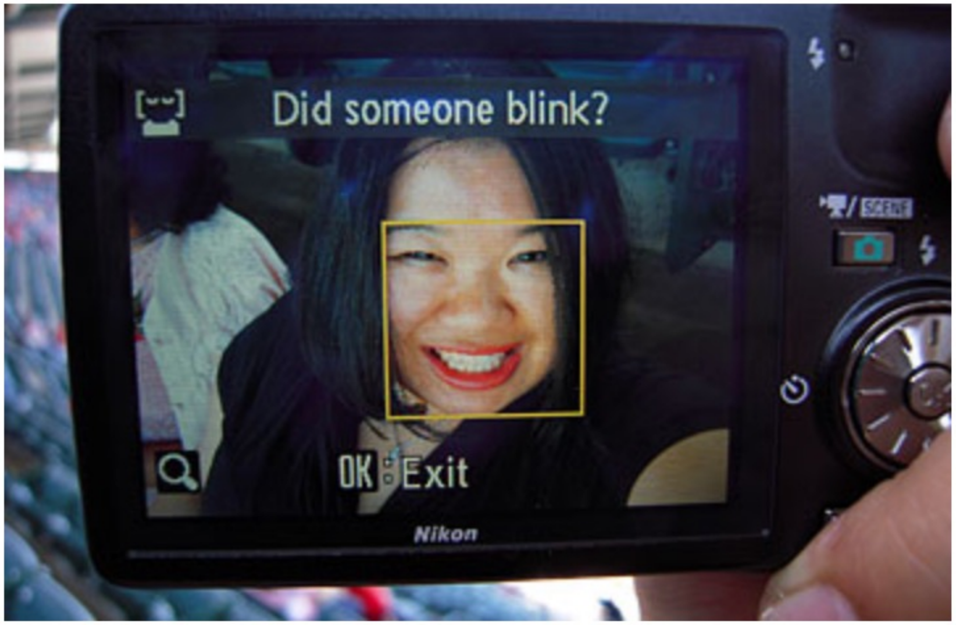
FAIRNESS IN ML

Are Face-Detection Cameras Racist?

By Adam Rose | Friday, Jan. 22, 2010

Tweet Share Read Later

When Joz Wang and her brother bought their mom a Nikon Coolpix S630 digital camera for Mother's Day last year, they discovered what seemed to be a malfunction. Every time they took a portrait of each other smiling, a message flashed across the screen asking, "Did someone blink?" No one had. "I thought the camera was broken!" Wang, 33, recalls. But when her brother posed with his eyes open so wide that he looked "bug-eyed," the messages stopped.



Joz Wang

Wang, a Taiwanese-American strategy consultant who goes by the Web handle "jozjozjoz," thought it was funny that the camera had difficulties figuring out when her family had their eyes open. So she

Source: http://content.time.com/time/business/article/0,8599,1954643,00.html

IS THE IPHONE X RACIST? APPLE REFUNDS DEVICE THAT CAN'T TELL CHINESE PEOPLE APART, WOMAN CLAIMS

BY **CHRISTINA ZHAO** ON 12/18/17 AT 12:24 PM EST

"A Chinese woman [surname Yan] was offered two refunds from Apple for her new iPhone X… [it] was unable to tell her and her other Chinese colleague apart."

"Thinking that a faulty camera was to blame, the store operator gave [Yan] a refund, which she used to purchase another iPhone X. But the new phone turned out to have the same problem, prompting the store worker to offer her another refund … It is unclear whether she purchased a third phone"

Source: https://www.newsweek.com/iphone-x-racist-apple-refunds-device-cant-tell-chinese-people-apart-woman-751263

"As facial recognition systems become more common, Amazon has emerged as a frontrunner in the field, courting customers around the US, including police departments and Immigration and Customs Enforcement (ICE)."

Gender and racial bias found in Amazon's facial recognition technology (again)

Research shows that Amazon's tech has a harder time identifying gender in darker-skinned and female faces

By James Vincent | Jan 25, 2019, 9:45am EST

Source: https://www.theverge.com/2019/1/25/18197137/amazon-rekognition-facial-recognition-bias-race-gender

Healthcare risk algorithm had 'significant racial bias'

It reportedly underestimated health needs for black patients.

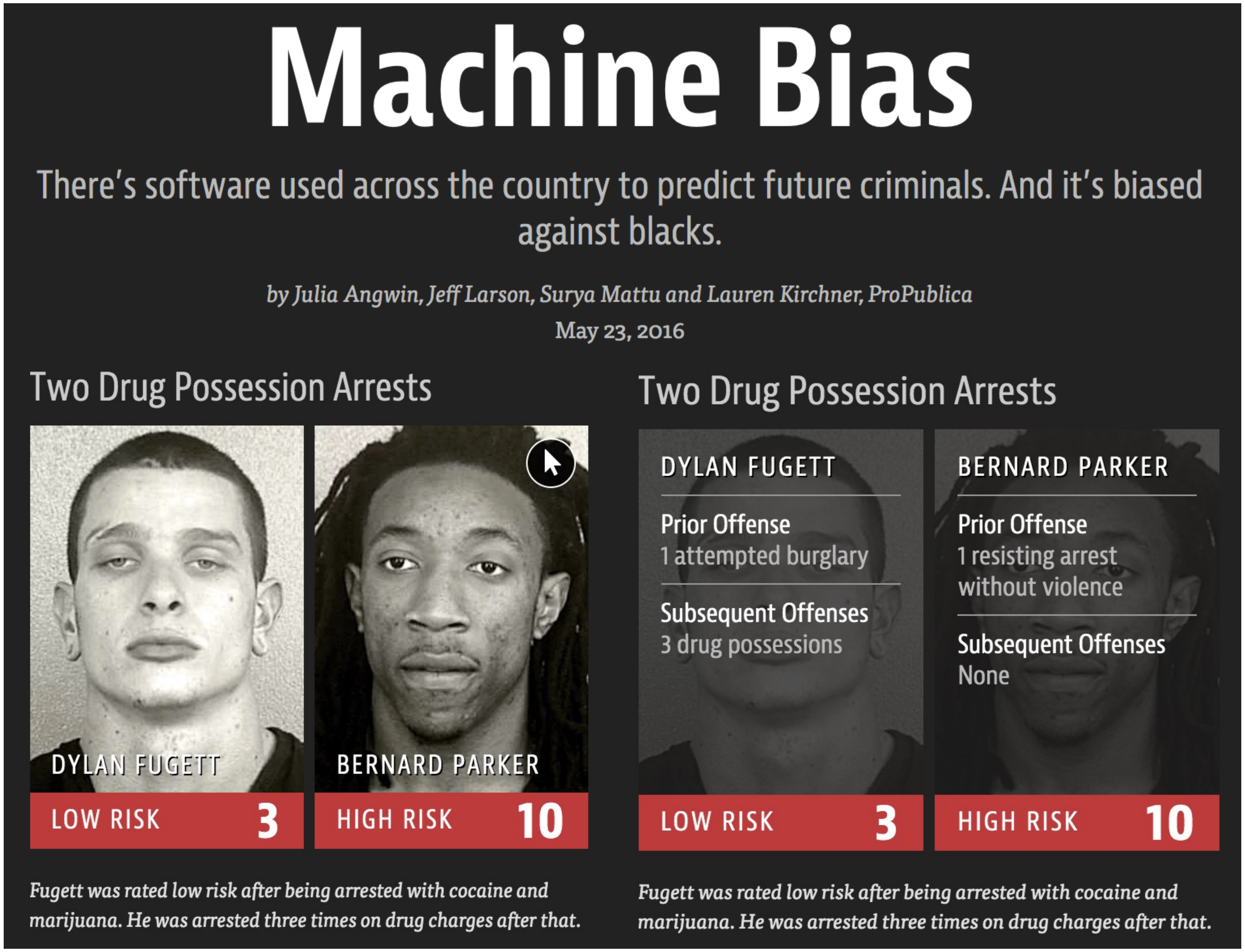
Jon Fingas, @jonfingas
10.26.19 in Medicine

"While it [the algorithm] didn't directly consider ethnicity, its emphasis on medical costs as bellwethers for health led to the code routinely underestimating the needs of black patients. A sicker black person would receive the same risk score as a healthier white person simply because of how much they could spend."

Source: https://science.sciencemag.org/content/366/6464/447

Word embeddings and analogies

- https://lamyiowce.github.io/word2viz/



Source: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Different Types of Errors

	True label	Predicted label
True positive (TP)	$+1$	$+1$
False positive (FP)	-1	$+1$
True negative (TN)	-1	-1
False negative (FN)	$+1$	-1

How We Analyzed the COMPAS Recidivism Algorithm

by Jeff Larson, Surya Mattu, Lauren Kirchner and Julia Angwin

May 23, 2016

All Defendants	Low	High
Survived	2681	1282
Recidivated	1216	2035
FP rate: 32.35		
FN rate: 37.40		

Black Defendants	Low	High
Survived	990	805
Recidivated	532	1369
FP rate: 44.85		
FN rate: 27.99		

White Defendants	Low	High
Survived	1139	349
Recidivated	461	505
FP rate: 23.45		
FN rate: 47.72		

This is one possible definition of unfairness.

We'll explore a few others and see how they relate to one another.

Source: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

CMU

Running Example

- Suppose you're an admissions officer for CMU, deciding which applicants to admit to your program
- $\boldsymbol{x}$ are the features of an applicant (e.g., standardized test scores, GPA)
- a is a protected attribute (e.g., gender), usually categorical i.e. $a \in \{a_1, \dots, a_C\}$
- $h(\boldsymbol{x}, a)$ is your model's prediction, which usually corresponds to some decision or action (e.g., $+1 =$ admit to CMU)
- y is the true, underlying target variable, usually thought of as some latent or hidden state (e.g., $+1 =$ this applicant would be "successful" at CMU)

Three Criteria for Fairness

- **Independence**: $h(\boldsymbol{x}, a) \perp a$
  - Probability of being accepted is the same for all genders
- **Separation**: $h(\boldsymbol{x}, a) \perp a \mid y$
  - All "good" applicants are accepted with the same probability, regardless of gender
  - Same for all "bad" applicants
- **Sufficiency**: $y \perp a \mid h(\boldsymbol{x}, a)$
  - For the purposes of predicting y, the information contained in $h(\boldsymbol{x}, a)$ is "sufficient", a becomes irrelevant

Achieving Fairness

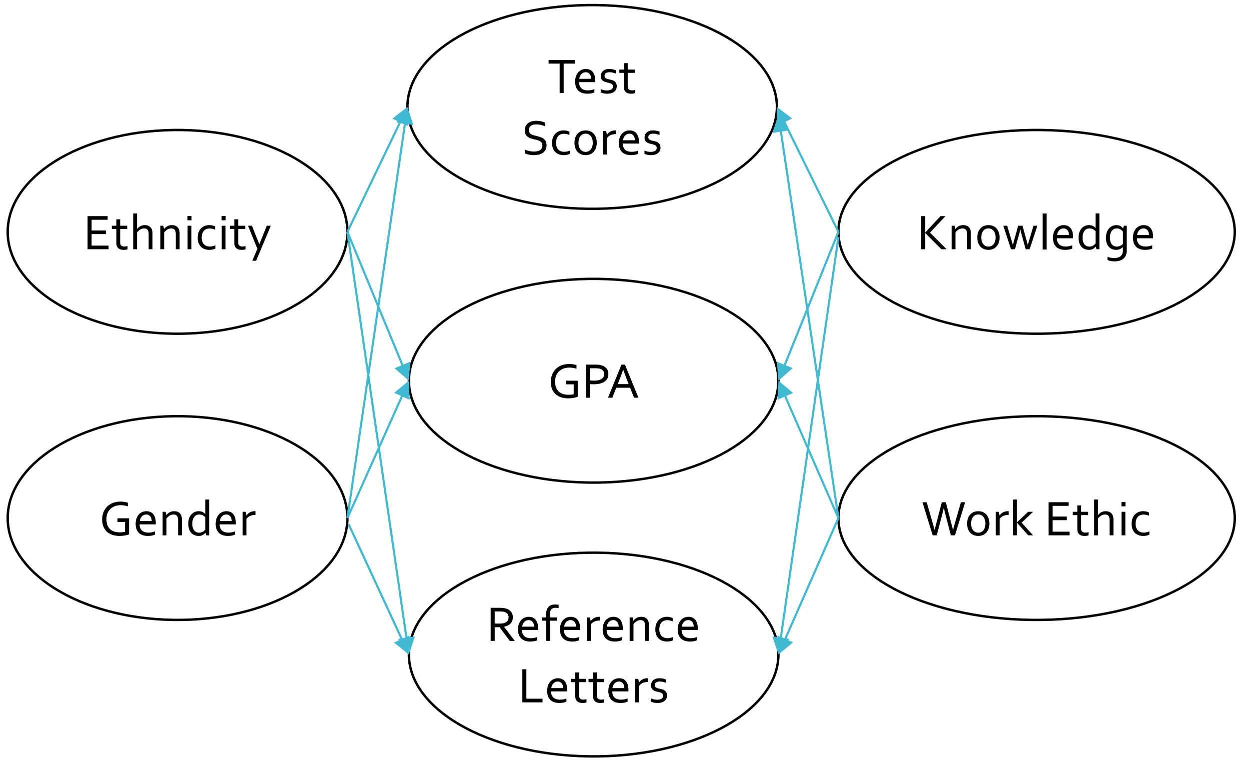
- Pre-processing data
- Additional constraints during training
- Post-processing predictions

Three Criteria for Fairness

- **Independence**: $h(\boldsymbol{x}, a) \perp a$
  - Probability of being accepted is the same for all genders
- **Separation**: $h(\boldsymbol{x}, a) \perp a \mid y$
  - All "good" applicants are accepted with the same probability, regardless of gender
  - Same for all "bad" applicants
- **Sufficiency**: $y \perp a \mid h(\boldsymbol{x}, a)$
  - For the purposes of predicting y, the information contained in $h(\boldsymbol{x}, a)$ is "sufficient", a becomes irrelevant
- Any two of these criteria are mutually exclusive in the general case!

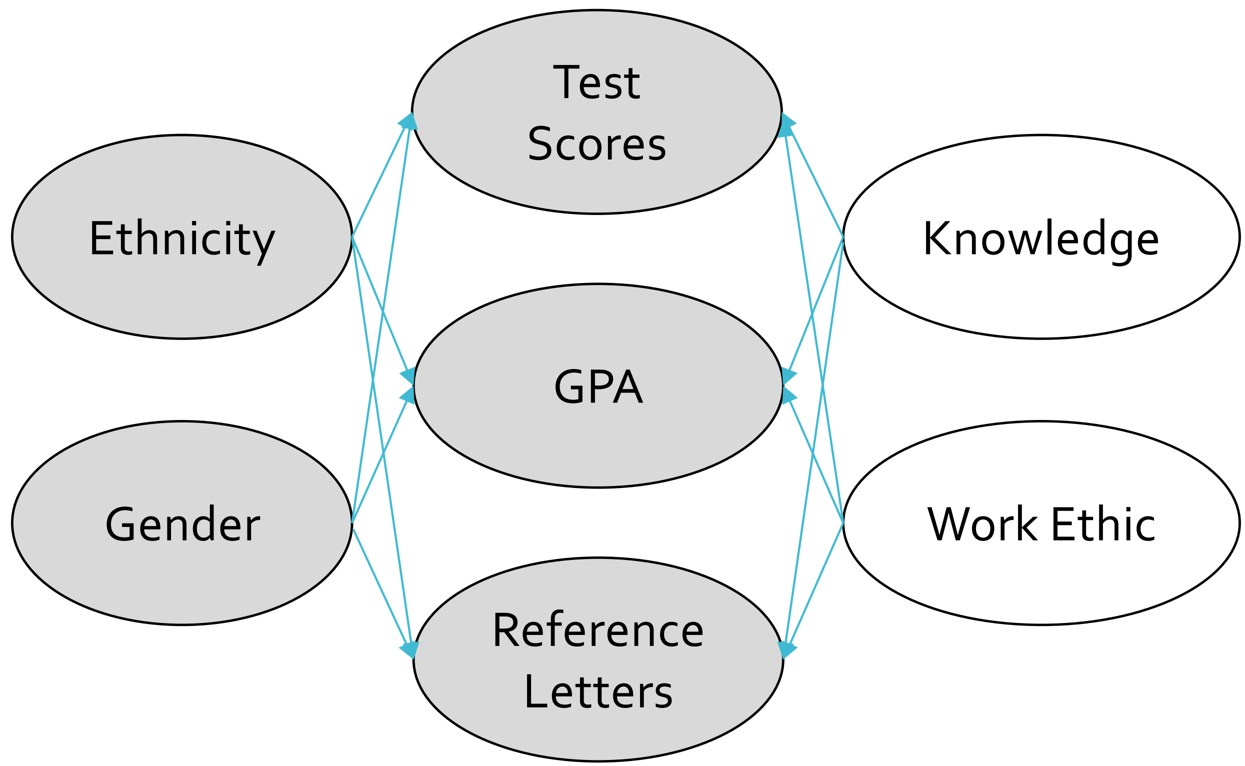
A Fourth Criterion for Fairness

- ~~Causality~~ Bayesian networks to the rescue!



A Fourth Criterion for Fairness

- ~~Causality~~ Bayesian networks to the rescue!



- Counterfactual fairness: how would an applicant's probability of acceptance change if they were a different gender?

Source: Counterfactual fairness, Kusner et al., https://papers.nips.cc/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf