Carnegie Mellon University
Heinz College

95-865 Unstructured Data Analytics

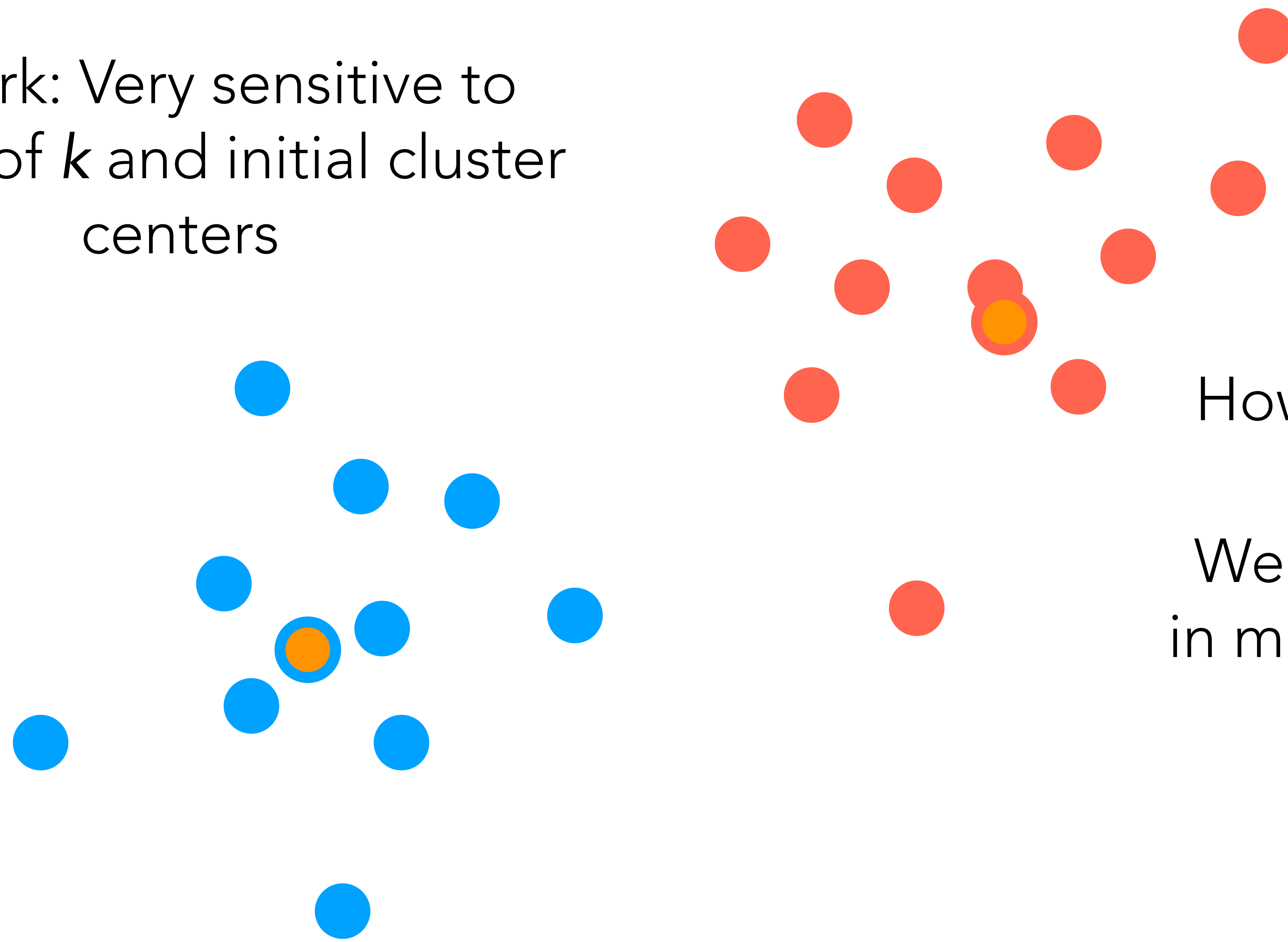
Lecture 7: Clustering

Slides by George H. Chen

(Flashback) *k*-means

Final output: cluster centers, cluster assignment for every point

Remark: Very sensitive to choice of *k* and initial cluster centers



How to guess *k*?

We'll discuss this in more detail next lecture

Suggested way to pick initial cluster centers: "*k*-means++" method

(rough intuition: incrementally add centers; favor adding center far away from centers chosen so far)

(Flashback)

When does *k*-means work well?

k-means is related to a generative model, which will help us understand when *k*-means is expected to work well

Example: Generative Model

Think of flipping a coin

each outcome: heads or tails

Each outcome doesn't depend on any of the previous outcomes

Example: Generative Model

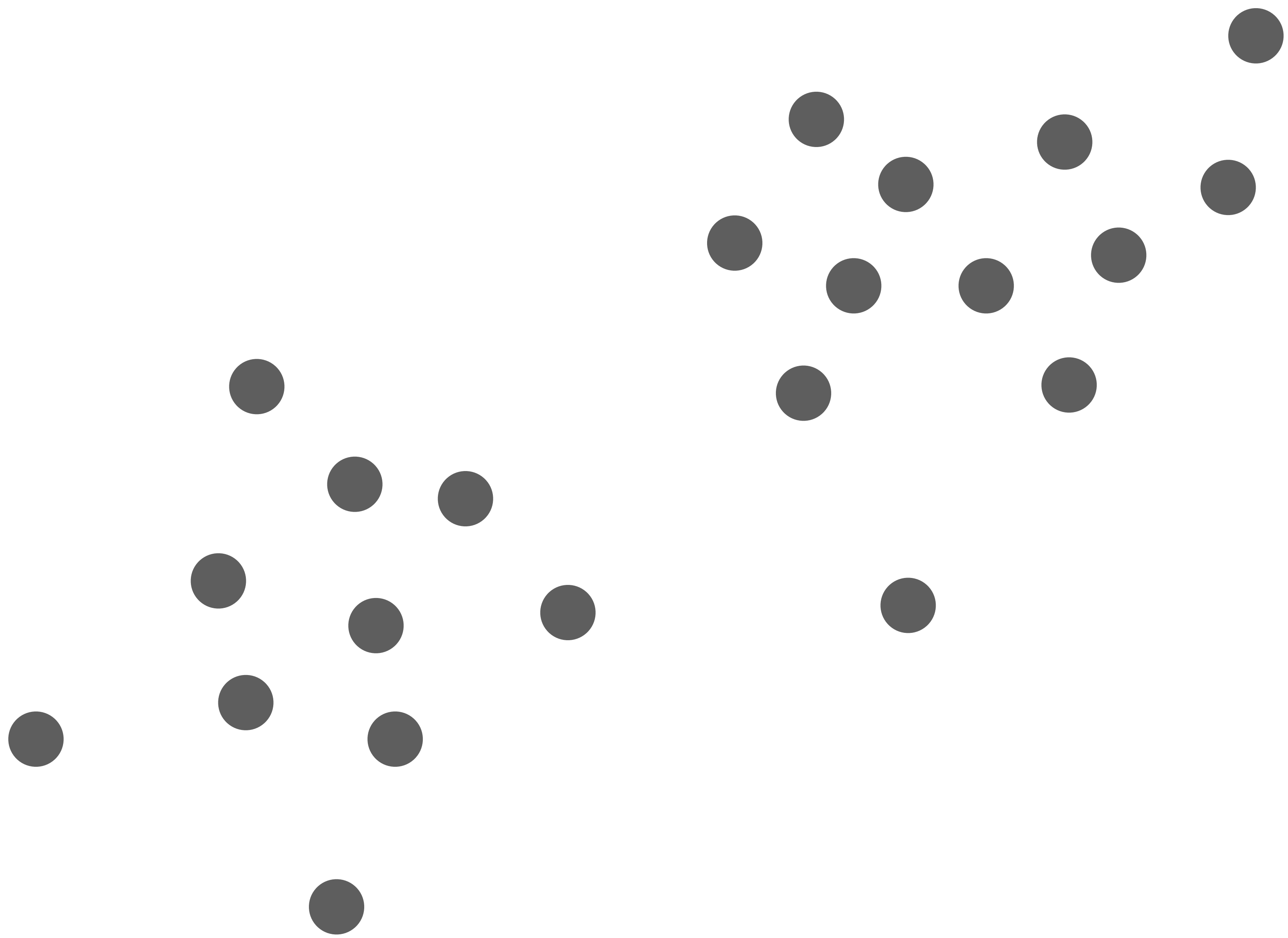
pressing a button

Think of ~~flipping a coin~~

each outcome: 2D point

Each outcome doesn't depend on any of the previous outcomes

Gaussian Mixture Model (GMM)



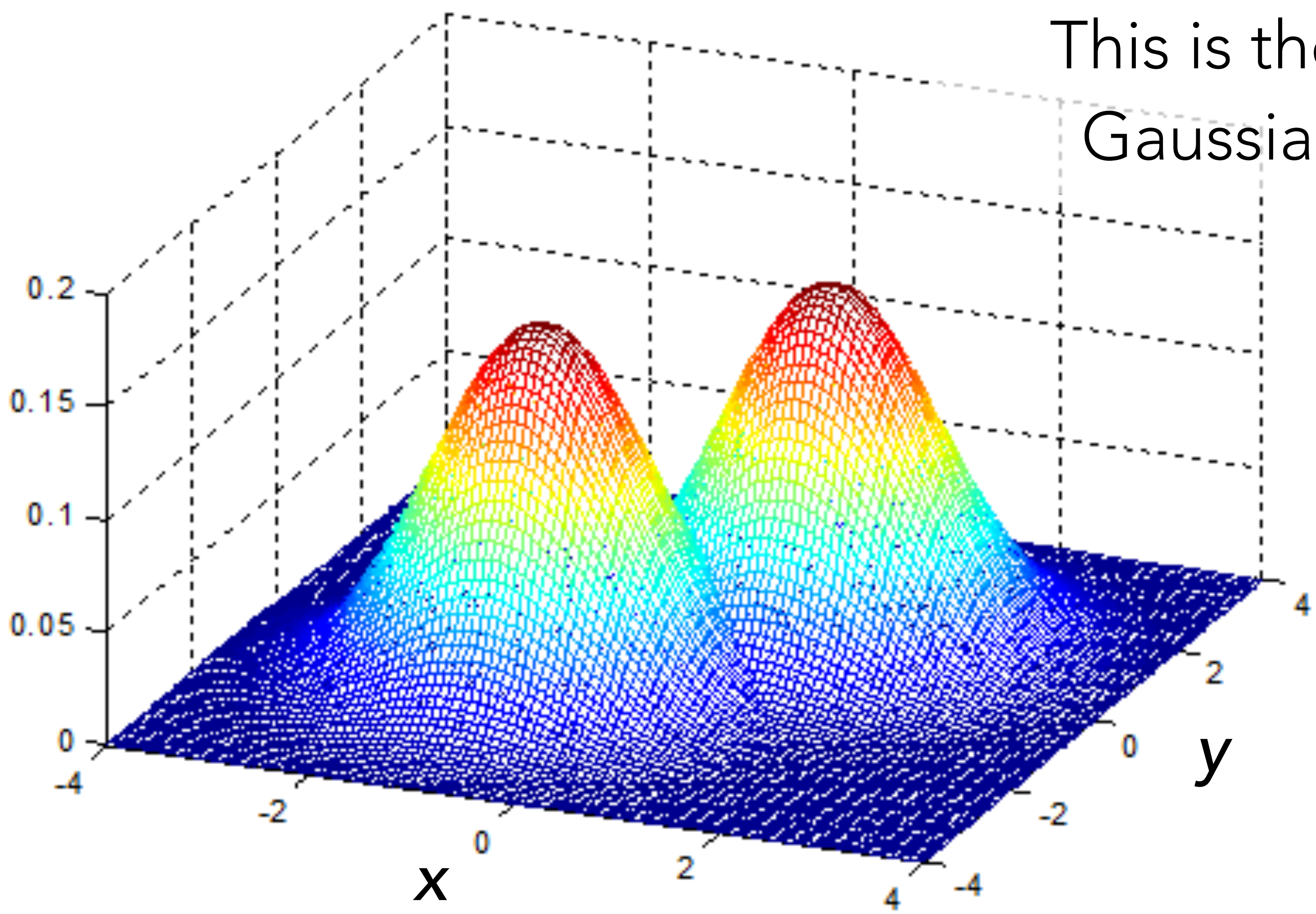
Each point sampled independently from same distribution
(but what is this distribution?)

Example: GMM for 2D Data

Every point sampled independently from probability distribution below:

This is the sum of two 2D Gaussian distributions!

how probable point generated at (x, y) is



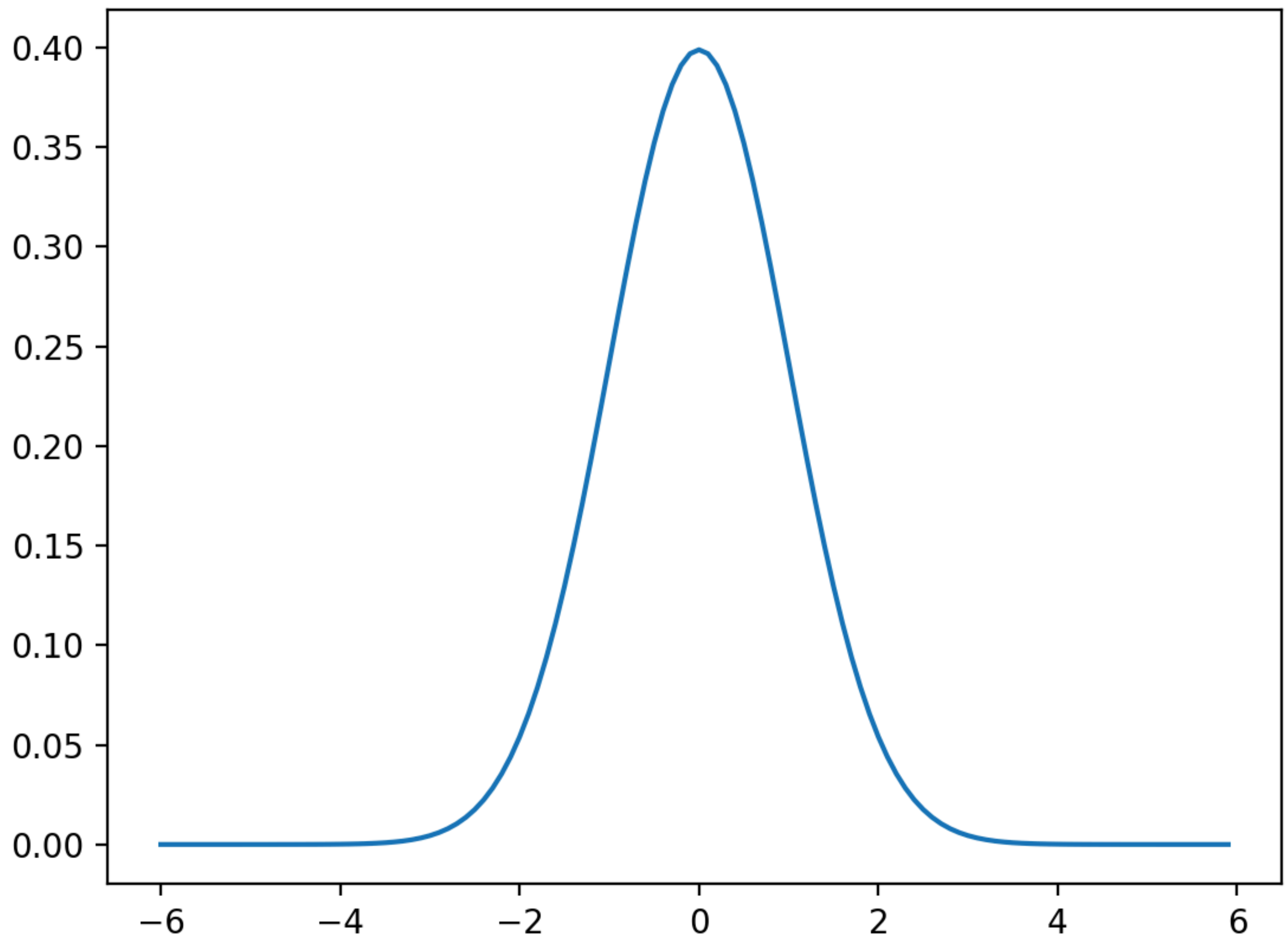
Red = more likely

Blue = less likely

Example of a 2D probability distribution

Image source: https://www.intechopen.com/source/html/17742/media/image25.png

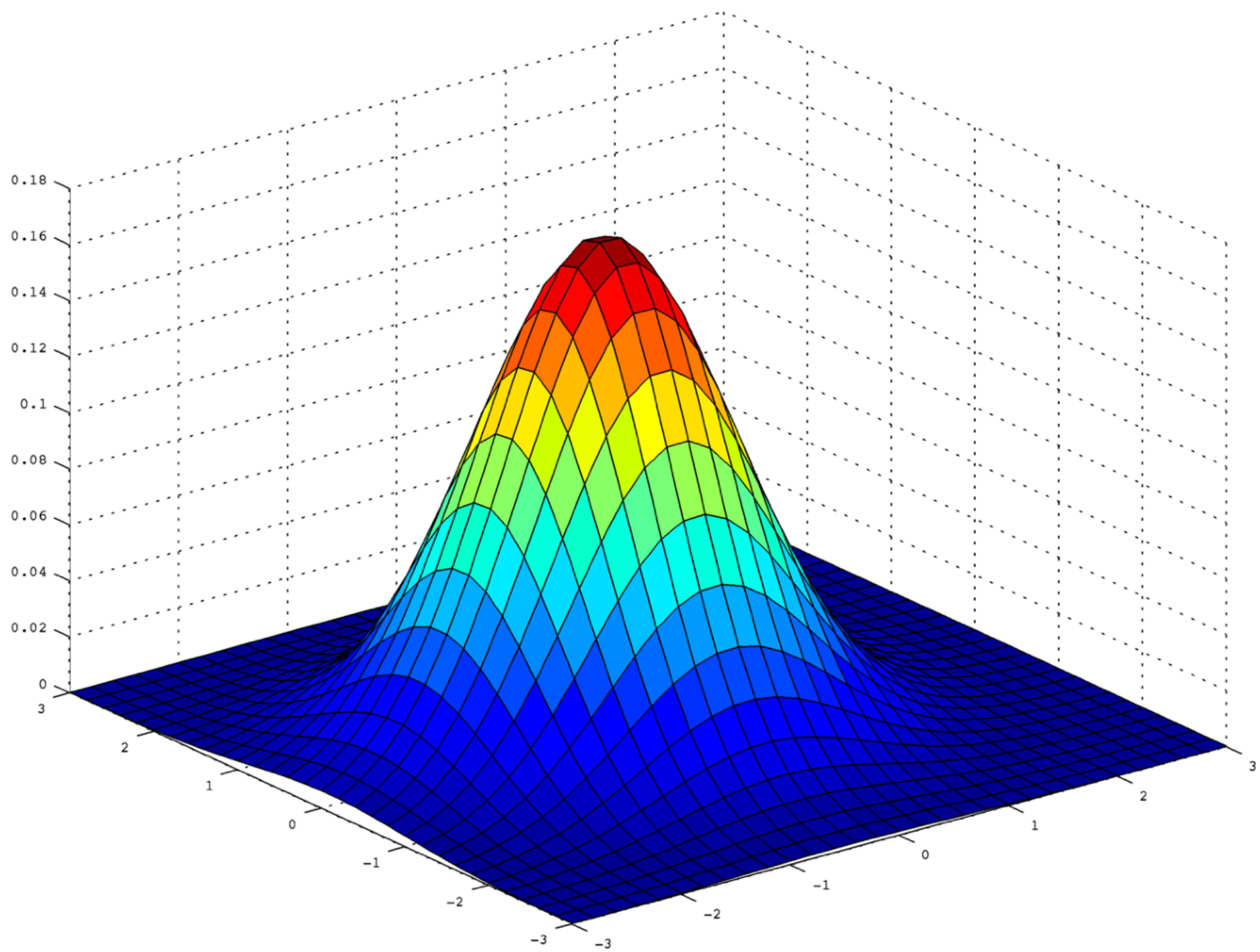
Quick Reminder: 1D Gaussian



This is a 1D Gaussian distribution

Image source: https://matthew-brett.github.io/teaching//smoothing_intro-3.hires.png

2D Gaussian



This is a 2D Gaussian distribution

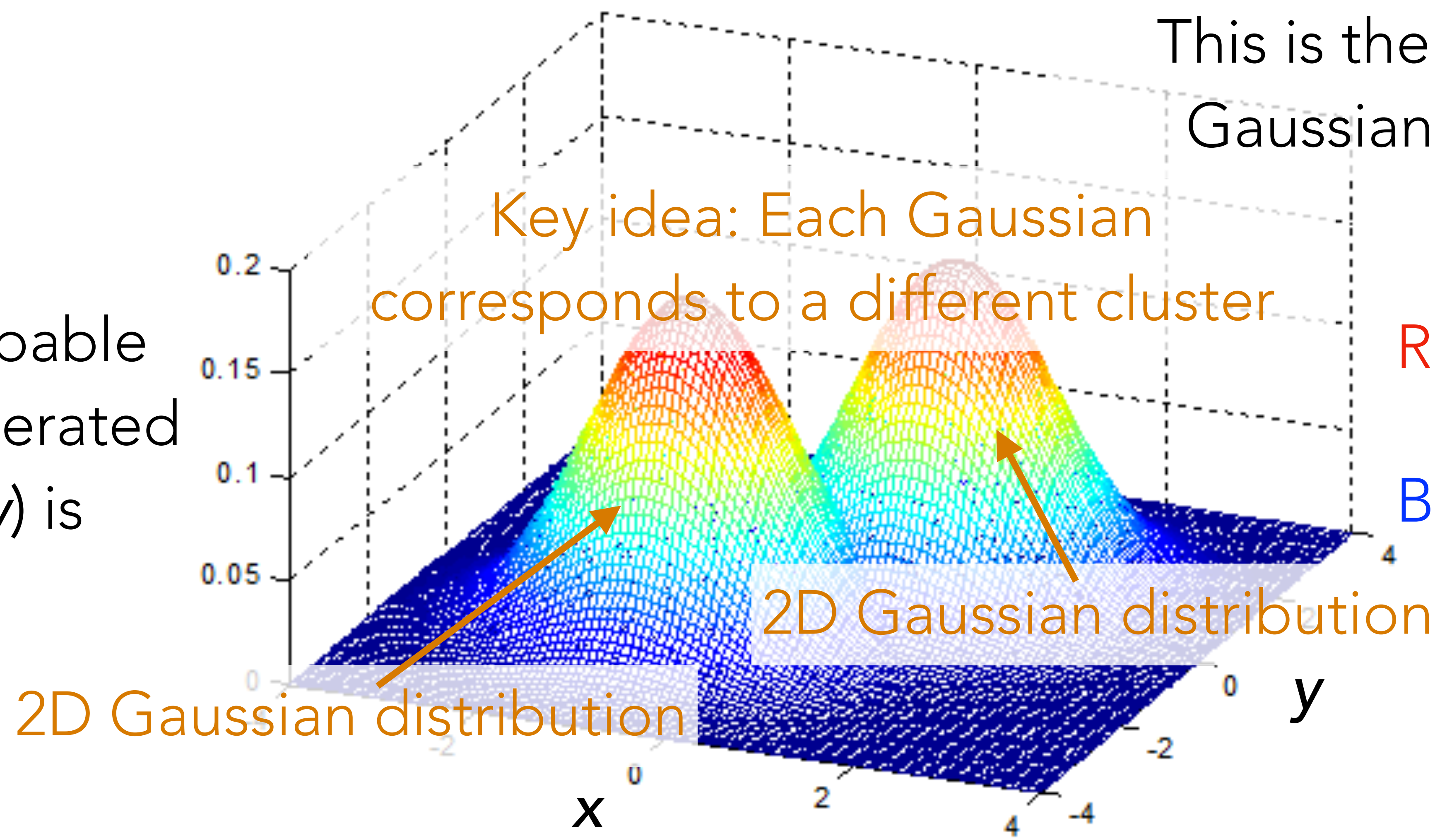
Image source: https://i.stack.imgur.com/OIWce.png

Example: GMM for 2D Data

Every point sampled independently from probability distribution below:

This is the sum of two 2D Gaussian distributions!

how probable point generated at (x, y) is



Red = more likely

Blue = less likely

Example of a 2D probability distribution

Image source: https://www.intechopen.com/source/html/17742/media/image25.png

GMM: The General Case

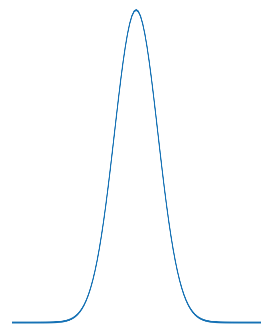
A GMM is the sum of k different d-dimensional Gaussian distributions so that the overall probability distribution looks like k mountains

(We've been looking at $d = 2$)

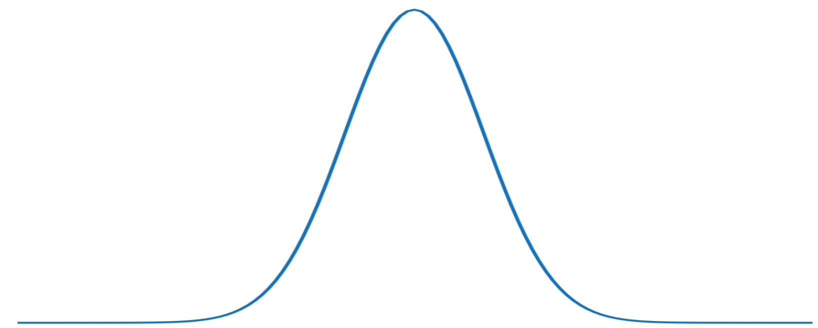
- Each mountain corresponds to a different cluster
- Different mountains can have different peak heights
- One missing thing we haven't discussed yet:
  different mountains can have different shapes

2D Gaussian Shape

In 1D, you can have a skinny Gaussian or a wide Gaussian



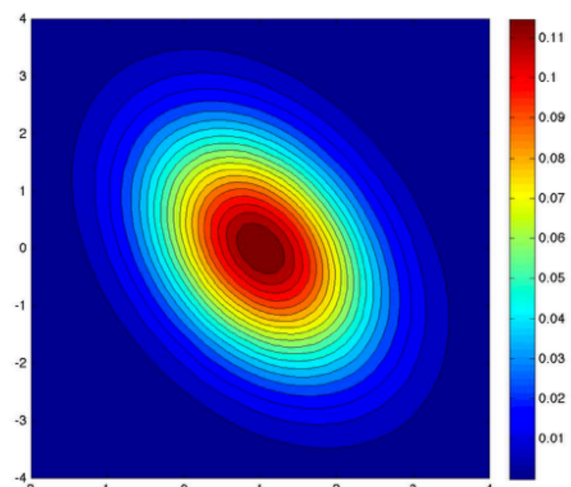
Less uncertainty



More uncertainty

In 2D, you can more generally have ellipse-shaped Gaussians

Ellipse enables encoding relationship between variables



Can't have arbitrary shapes

Top-down view of an example 2D Gaussian distribution

Image source: https://www.cs.colorado.edu/~mozer/Teaching/syllabi/ProbabilisticModels2013/homework/assign5/a52dgauss.jpg

GMM: The General Case

A GMM is the sum of k different d-dimensional Gaussian distributions so that the overall probability distribution looks like k mountains

(We've been looking at $d = 2$)

- Each mountain corresponds to a different cluster
- Different mountains can have different peak heights
- Different mountains can have different ellipse shapes (captures correlation/"covariance" information)

Example: 1D GMM with 2 Clusters

Cluster 1

Probability of generating a point from cluster 1 = 0.5

Gaussian mean = −5

Gaussian variance = 1

Cluster 2

Probability of generating a point from cluster 2 = 0.5

Gaussian mean = 5

Gaussian variance = 1

What do you think this looks like?

Example: 1D GMM with 2 Clusters

<u>Cluster 1</u>

Probability of generating a point from cluster 1 = 0.5

Gaussian mean = –5

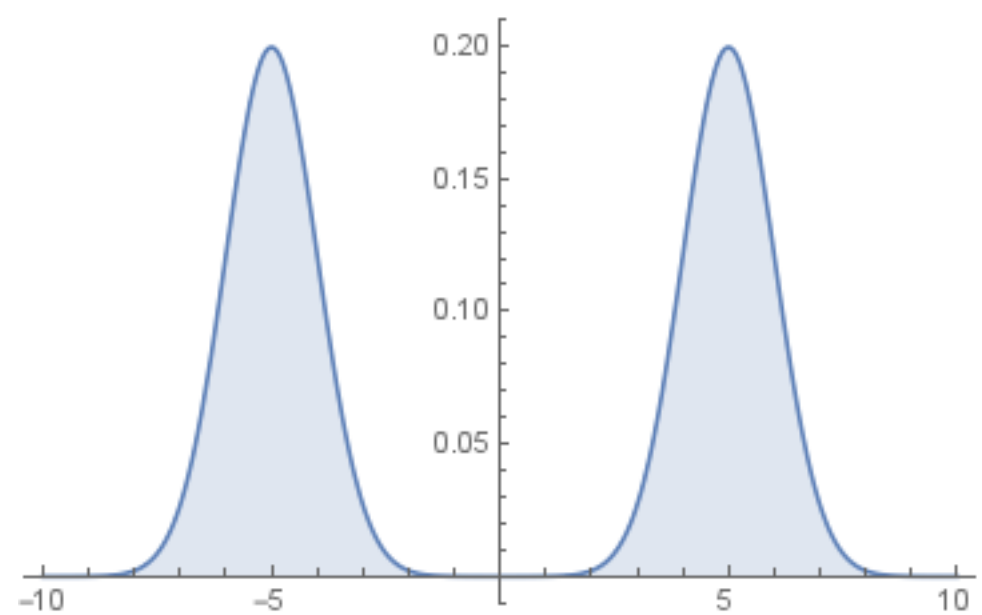
Gaussian variance = 1

<u>Cluster 2</u>

Probability of generating a point from cluster 2 = 0.5

Gaussian mean = 5

Gaussian variance = 1



Example: 1D GMM with 2 Clusters

Cluster 1

Probability of generating a point from cluster 1 = **0.7**

Gaussian mean = –5

Gaussian variance = 1

Cluster 2

Probability of generating a point from cluster 2 = **0.3**

Gaussian mean = 5

Gaussian variance = 1

What do you think this looks like?

Example: 1D GMM with 2 Clusters

<u>Cluster 1</u>

Probability of generating a point from cluster 1 = 0.7

Gaussian mean = −5

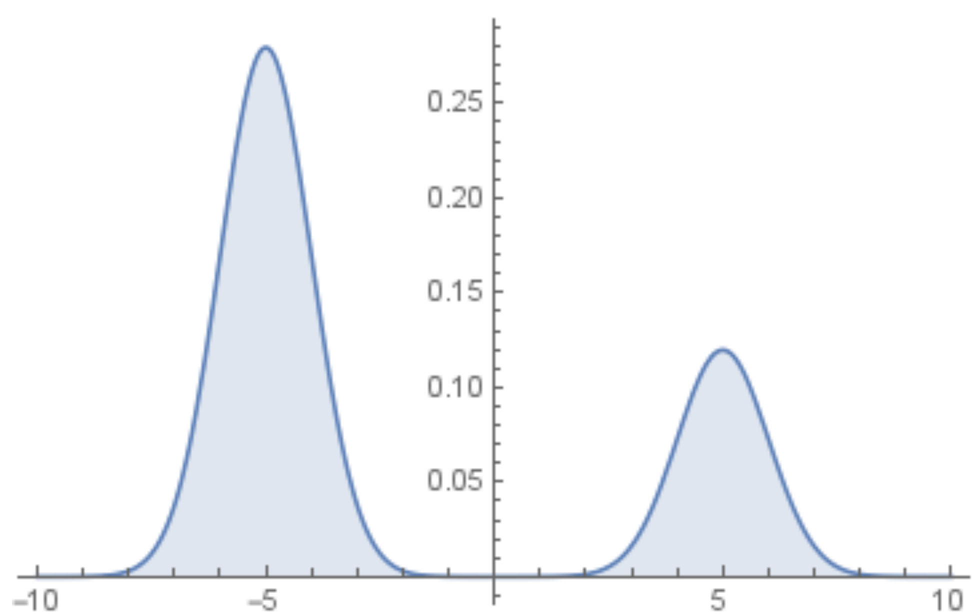
Gaussian variance = 1

<u>Cluster 2</u>

Probability of generating a point from cluster 2 = 0.3

Gaussian mean = 5

Gaussian variance = 1



Example: 1D GMM with 2 Clusters

<u>Cluster 1</u>

Probability of generating a point from cluster 1 = 0.7

Gaussian mean = −5

Gaussian variance = 1

<u>Cluster 2</u>

Probability of generating a point from cluster 2 = 0.3

Gaussian mean = 5

Gaussian variance = 1

How to generate 1D points from this GMM:

1. Flip biased coin (side 1 has probability 0.7, side 2 has probability 0.3)

   Let Z be the side that we got (it is either 1 or 2)

2. If Z = 1: sample 1 point from Gaussian mean −5, variance 1

   If Z = 2: sample 1 point from Gaussian mean 5, variance 1

Example: 1D GMM with 2 Clusters

<u>Cluster 1</u>

Probability of generating a point from cluster 1 = 0.7

Gaussian mean = −5

Gaussian variance = 1

<u>Cluster 2</u>

Probability of generating a point from cluster 2 = 0.3

Gaussian mean = 5

Gaussian variance = 1

How to generate 1D points from this GMM:

1. Flip biased coin (side 1 has probability 0.7, side 2 has probability 0.3)

   Let Z be the side that we got (it is either 1 or 2)

2. Sample 1 point from the Gaussian from cluster Z

Example: 1D GMM with 2 Clusters

Cluster 1

Probability of generating a point from cluster 1 = π_1

Gaussian mean = μ_1

Gaussian variance = σ_1^2

Cluster 2

Probability of generating a point from cluster 2 = π_2

Gaussian mean = μ_2

Gaussian variance = σ_2^2

How to generate 1D points from this GMM:

1. Flip biased coin (side 1 has probability π_1, side 2 has probability π_2)

   Let *Z* be the side that we got (it is either 1 or 2)

2. Sample 1 point from the Gaussian from cluster *Z*

Example: 1D GMM with *k* Clusters

<u>Cluster 1</u>

Probability of generating a point from cluster 1 = π_1

Gaussian mean = μ_1

Gaussian variance = σ_1^2

$\cdots$

<u>Cluster *k*</u>

Probability of generating a point from cluster *k* = π_k

Gaussian mean = μ_k

Gaussian variance = σ_k^2

How to generate 1D points from this GMM:

1. Flip biased coin (side 1 has probability π_1, …, side *k* has probability π_k)

   Let *Z* be the side that we got (it is some value 1, …, *k*)

2. Sample 1 point from the Gaussian from cluster *Z*

Example: 2D GMM with *k* Clusters

Cluster 1		Cluster *k*
Probability of generating a point from cluster 1 = π_1	…	Probability of generating a point from cluster ***k*** = π_k
Gaussian mean = μ_1		Gaussian mean = μ_k
Gaussian covariance = Σ_1		Gaussian covariance = Σ_k

2-dim.

2-by-2 matrices

How to generate 2D points from this GMM:

1. Flip biased coin (side 1 has probability π_1, …, side *k* has probability π_k)

   Let *Z* be the side that we got (it is some value 1, …, *k*)

2. Sample 1 point from the Gaussian from cluster *Z*

Example: GMM with *k* Clusters

<u>Cluster 1</u>

Probability of generating a point from cluster 1 = π_1

Gaussian mean = μ_1

Gaussian covariance = Σ_1

$\cdots$

<u>Cluster *k*</u>

Probability of generating a point from cluster *k* = π_k

Gaussian mean = μ_k ← *d*-dim.

Gaussian covariance = Σ_k

← *d*-by-*d* matrices

How to generate points from this GMM:

1. Flip biased coin (side 1 has probability π_1, …, side *k* has probability π_k)

   Let *Z* be the side that we got (it is some value 1, …, *k*)

2. Sample 1 point from the Gaussian from cluster *Z*

High-Level Idea of GMM

- Generative model that gives a *hypothesized* way in which data points are generated

In reality, data are unlikely generated the same way!

In reality, data points might not even be independent!

“All models are wrong, but some are useful.”

–George Box

High-Level Idea of GMM

- Generative model that gives a *hypothesized* way in which data points are generated

In reality, data are unlikely generated the same way!

In reality, data points might not even be independent!

- Learning ("fitting") the parameters of a GMM
  - Input: *d*-dimensional data points, your guess for *k*
  - Output: $\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k$
- *After* learning a GMM:
  - For *any* *d*-dimensional data point, can figure out probability of it belonging to each of the clusters

How do you turn this into a cluster assignment?

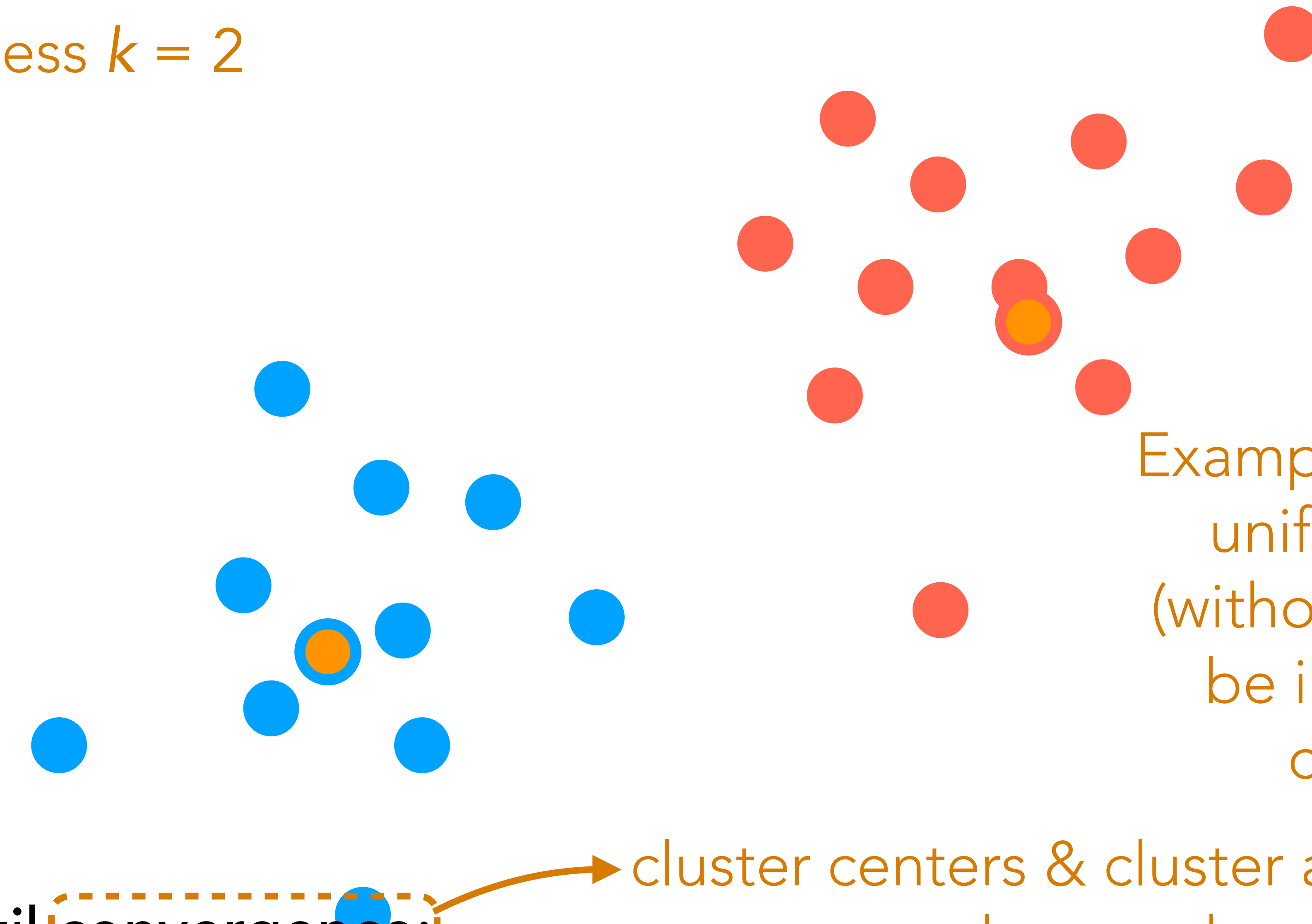
of clusters k-means

Distance function: Euclidean

Step 0: Guess k

We'll guess k = 2

Step 1: Guess where cluster centers are



Example: choose k points uniformly at random (without replacement) to be initial guesses for cluster centers

Repeat until convergence:

cluster centers & cluster assignments no longer change

Step 2: Assign each point to belong to the closest cluster

Step 3: Update cluster means (to be the center of mass per cluster)

k-means

Step 0: Guess *k*

Step 1: Guess where cluster centers are

Repeat until convergence:

Step 2: Assign each point to belong to the closest cluster

Step 3: Update cluster means (to be the center of mass per cluster)

(Rough Intuition) Learning a GMM

Step 0: Guess *k*

Step 1: Guess cluster probabilities, means, and covariances

(often done using *k*-means)

Repeat until convergence:

Step 2: Compute probability of each point being in each of the *k* clusters

Step 3: Update cluster probabilities, means, and covariances accounting for probabilities of each point belonging to each of the clusters

This algorithm is called the **Expectation-Maximization** (EM) algorithm for GMMs (and approximately does maximum likelihood)

(Note: EM by itself is a general algorithm not just for GMMs)

Clustering

Demo