

GAUSSIAN MIXTURE MODELS

Scott O'Hara

Metrowest Developers Machine Learning Group

03/17/2021

REFERENCES

Machine Learning: Clustering & Retrieval

University of Washington, Profs. Emily Fox & Carlos Guestrin

<https://www.coursera.org/learn/ml-regression/home/welcome>

CS181 course at Harvard University:

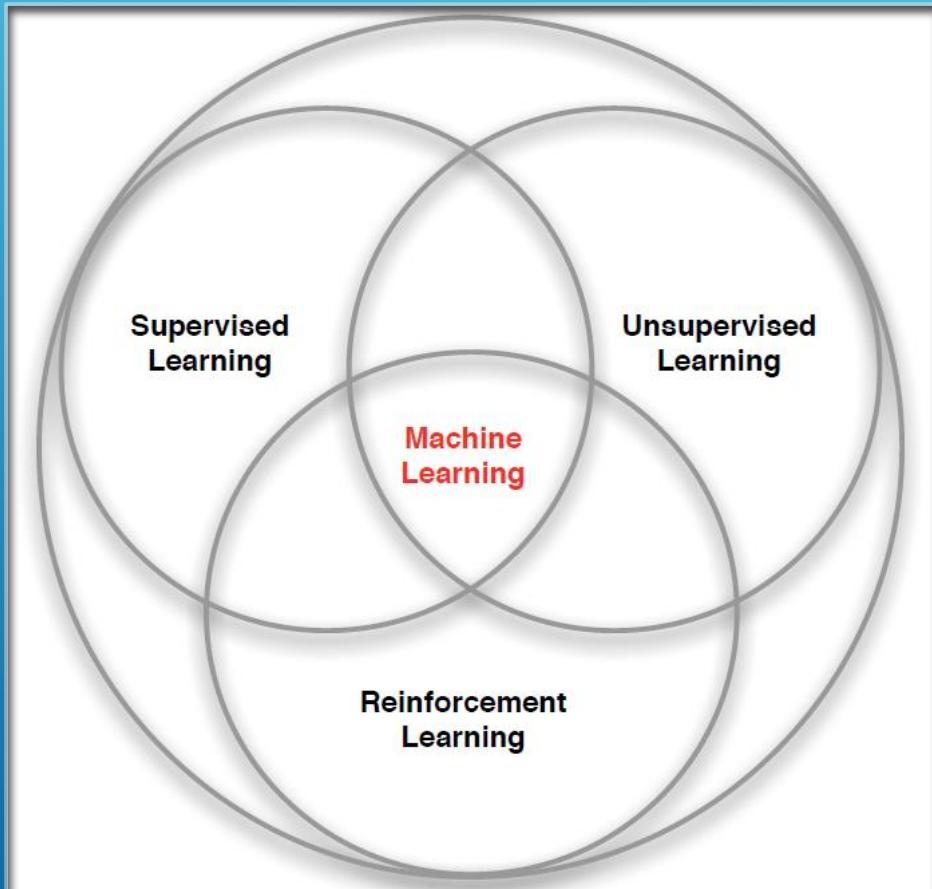
- ▶ *CS181 Intelligent Machines: Perception, Learning and Uncertainty, Sarah Finney, Spring 2009*
- ▶ *CS181 Intelligent Machines: Perception, Learning and Uncertainty, Prof. David C Brooks, Spring 2011*
- ▶ *CS181 – Machine Learning, Prof. David Parkes, Spring 2017.* <https://harvard-ml-courses.github.io/cs181-web-2017/>

Applied Machine Learning in Python

University of Michigan, Prof. Kevin Collins Thompson

<https://www.coursera.org/learn/python-machine-learning/home/welcome>

3 TYPES OF MACHINE LEARNING



Supervised Learning – Learn a function from labeled data that maps input attributes to an output label e.g., linear regression, decision trees, SVMs.

Unsupervised Learning – Learn patterns in unlabeled data e.g., principle component analysis or clustering algorithms such as K-means, HAC, or Gaussian mixture models.

Reinforcement Learning – An agent learns to maximize rewards while acting in an uncertain environment.

FOUR KINDS OF UNSUPERVISED LEARNING

References:

The Hundred-Page Machine Learning Book. Andriy Burkov.

Applied Machine Learning in Python. Coursera. University of Michigan, Prof. Kevin Collins Thompson

Cluster analysis,
https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1002271612 (last visited Jan. 27, 2021).

Dimensionality reduction,
https://en.wikipedia.org/w/index.php?title=Dimensionality_reduction&oldid=1002754996 (last visited Jan. 27, 2021).

1. Density Estimation

- Model the probability density function of the unknown probability distribution from which the dataset has been drawn.

2. Dimensionality Reduction

- Finds an approximate version of a dataset using fewer features while retaining some meaningful properties of the original data.

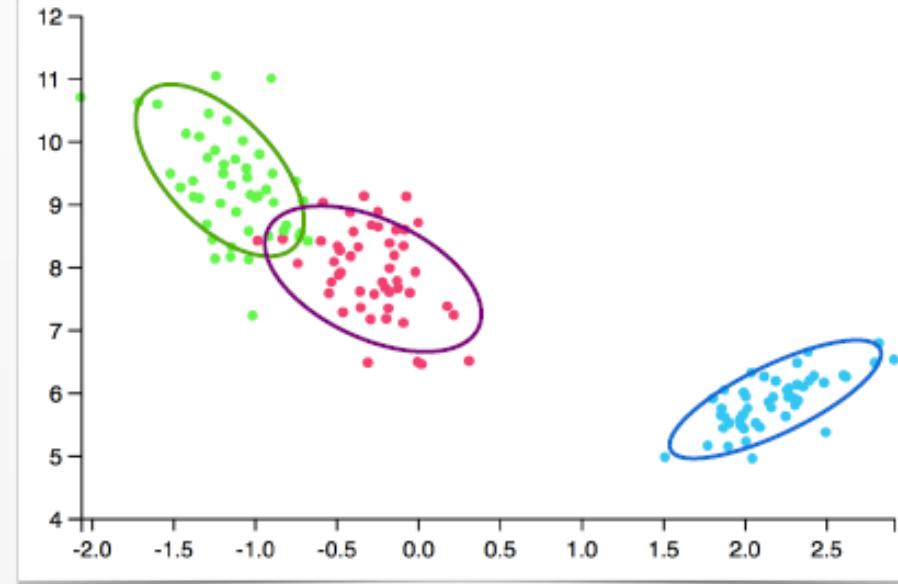
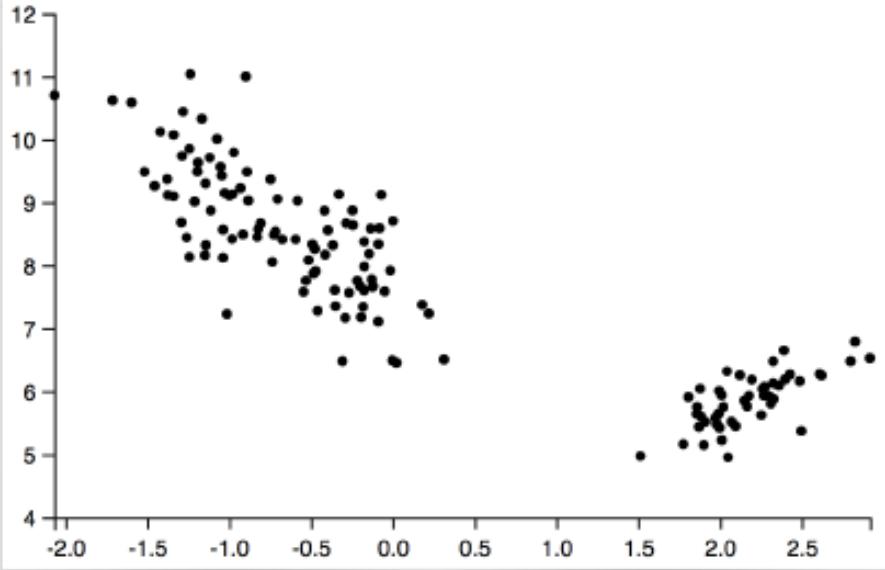
3. Outlier Detection

- Detect the examples in the dataset that are very different from what a typical example in the dataset looks like.

4. Clustering

- The task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more like each other than to those in other groups (clusters).

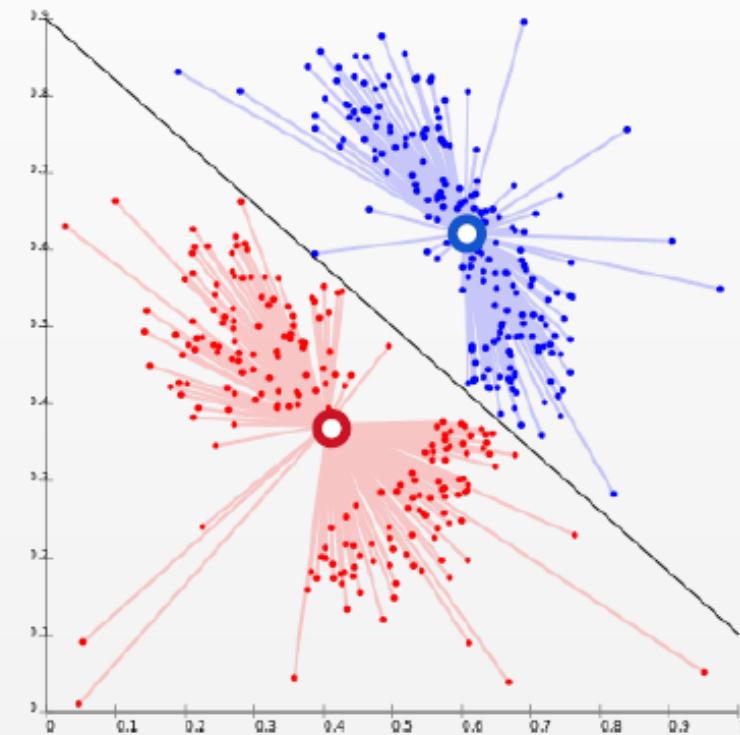
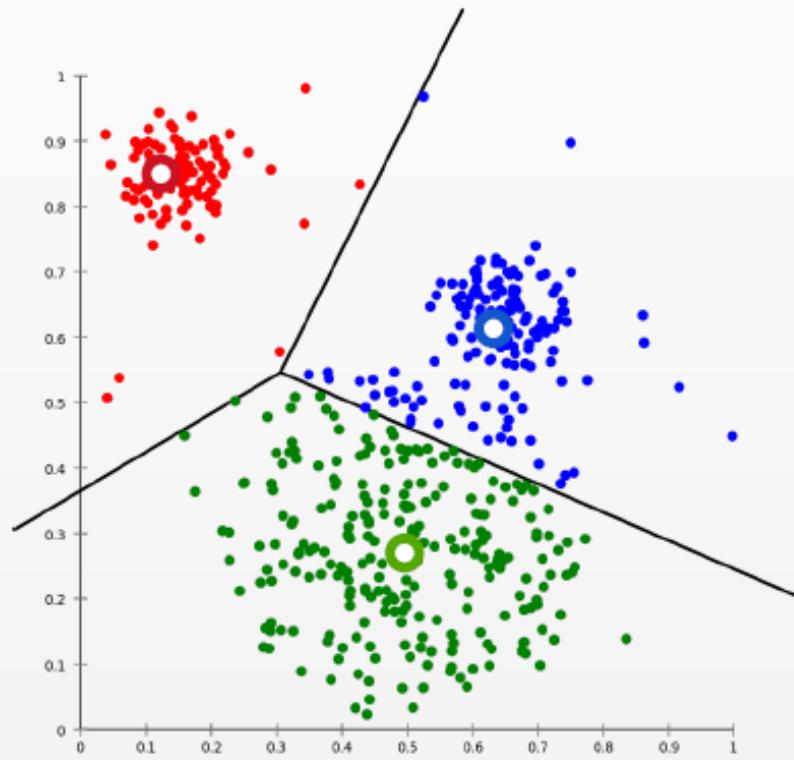
Clustering



- Unsupervised learning (no labels for training)
- Group data into similar classes that
 - Maximize *inter-cluster* similarity
 - Minimize *intra-cluster* similarity

Four Types of Clustering

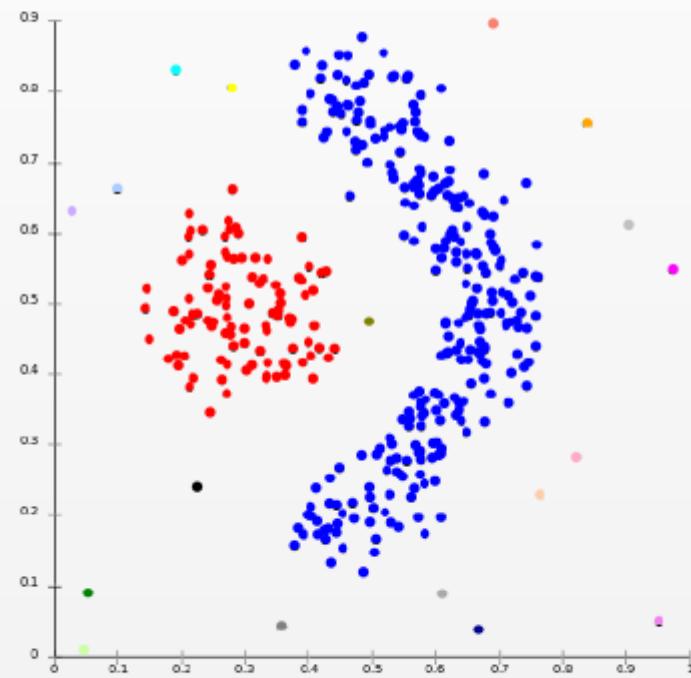
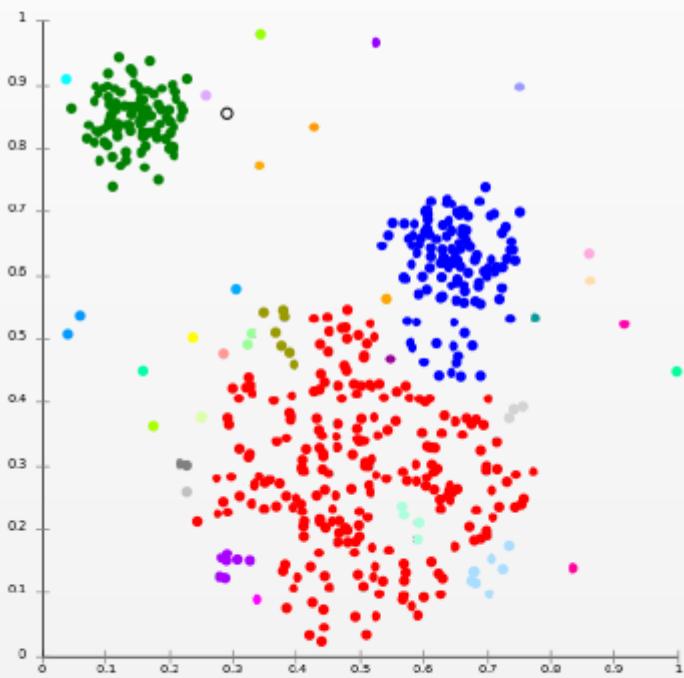
1. *Centroid-based (K-means, K-medoids)*



Notion of Clusters: Voronoi tessellation

Four Types of Clustering

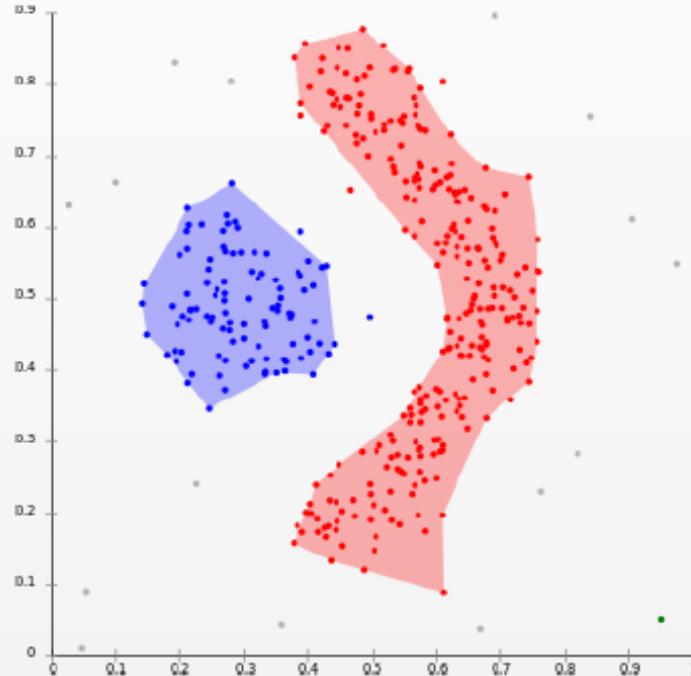
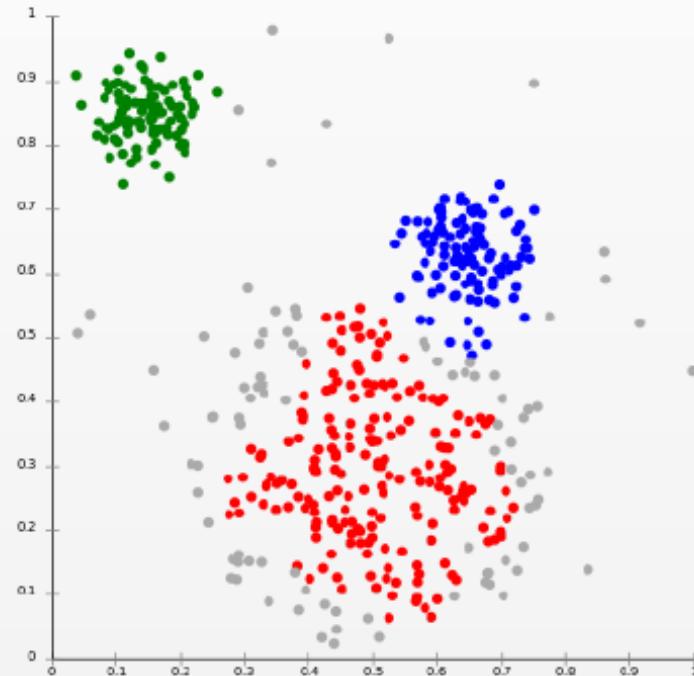
2. Connectivity-based (Hierarchical)



Notion of Clusters: Cut off dendrogram at some depth

Four Types of Clustering

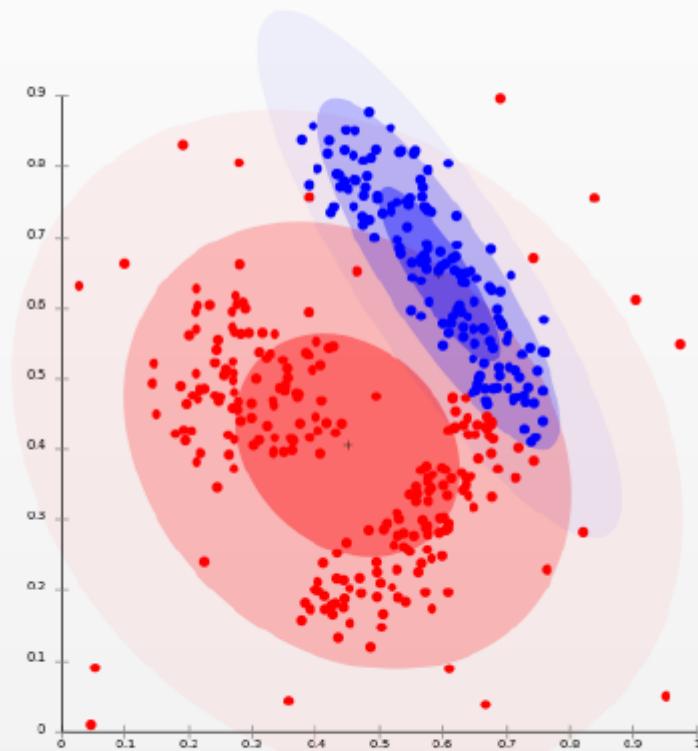
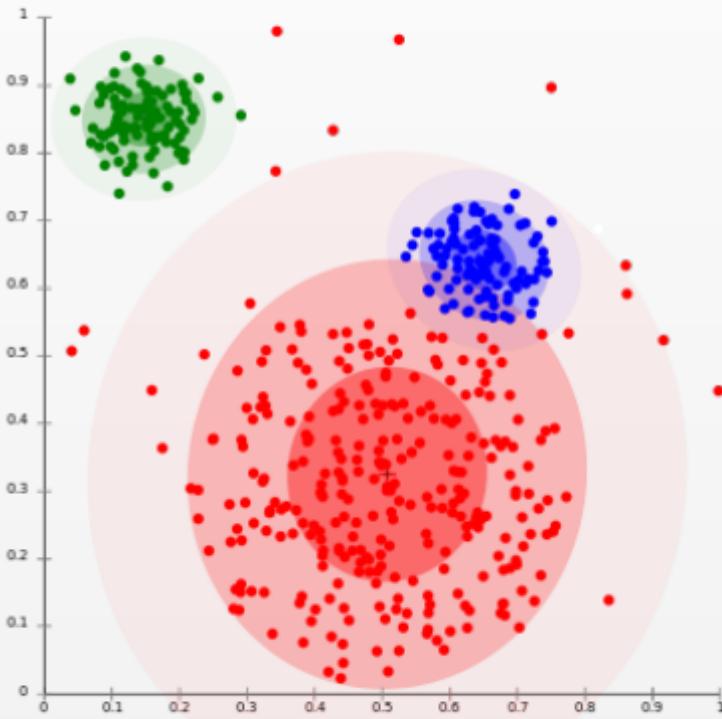
3. Density-based (*DBSCAN*, *OPTICS*)



Notion of Clusters: Connected regions of high density

Four Types of Clustering

4. Distribution-based (Mixture Models)



Notion of Clusters: Distributions on features

K-means Clustering

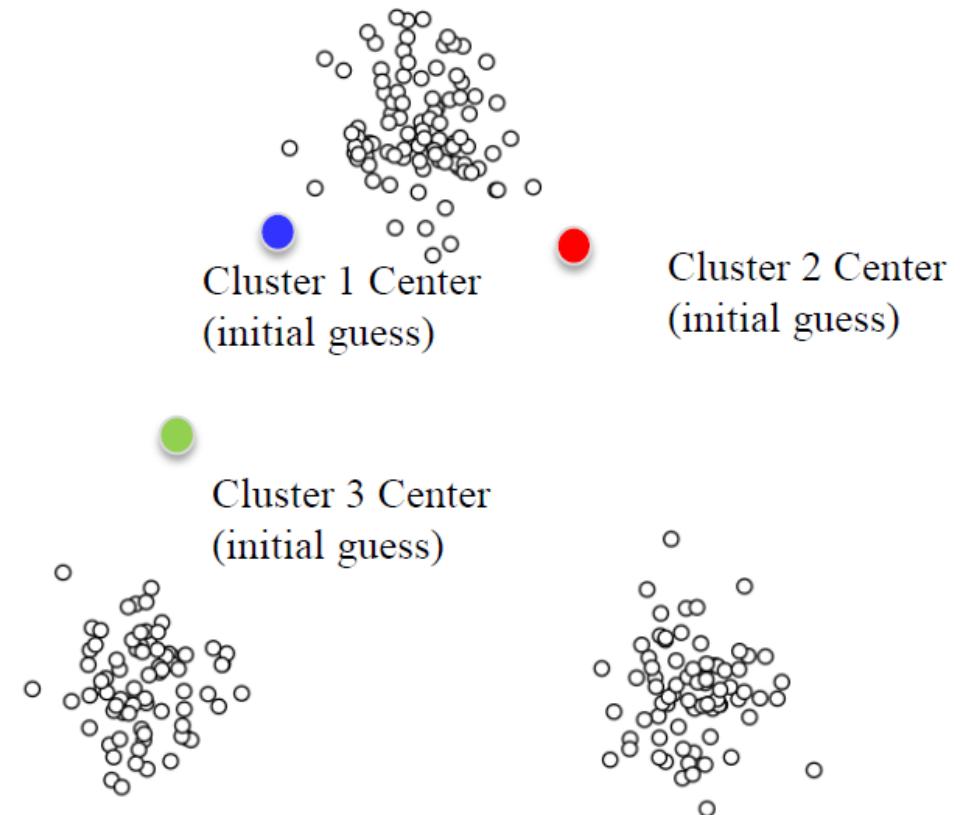
The k-means algorithm

Initialization Pick the number of clusters k you want to find. Then pick k *random* points to serve as an initial guess for the cluster centers.

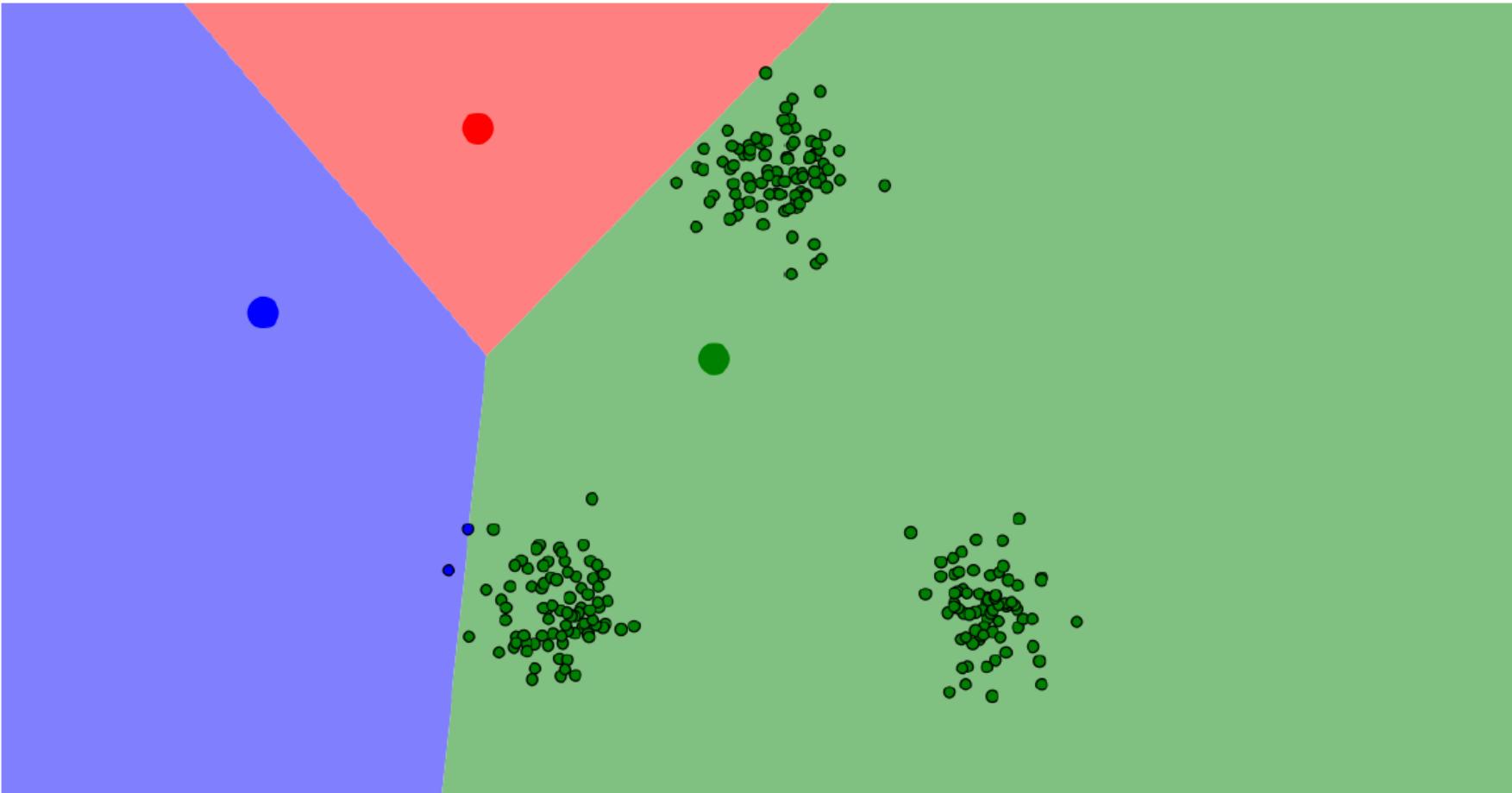
Step A Assign each data point to the nearest cluster center.

Step B Update each cluster center by replacing it with the mean of all points assigned to that cluster (in step A).

Repeat steps A and B until the centers converge to a stable solution.



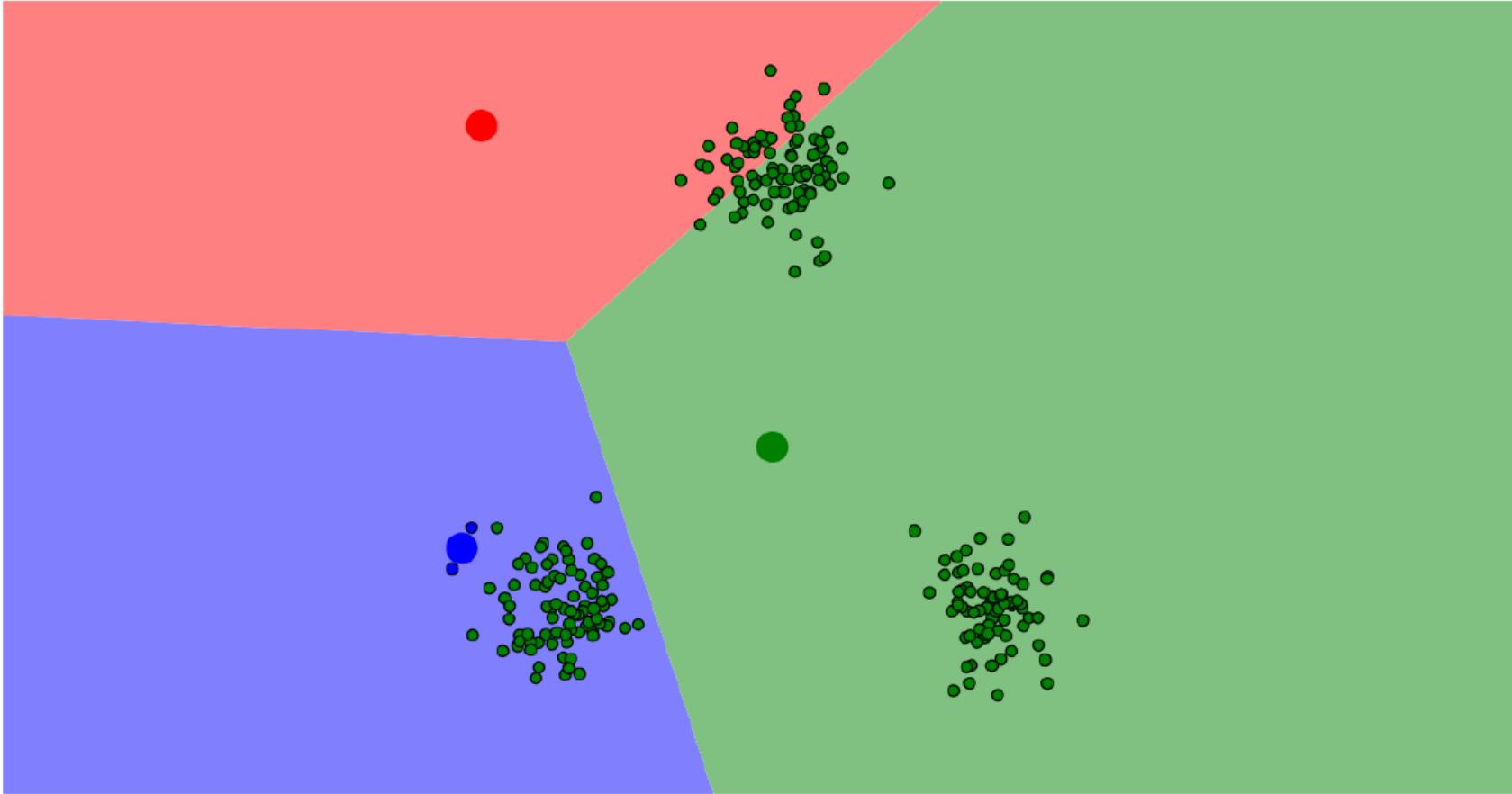
K-means Example: Step 1A



We want three clusters, so three centers are chosen randomly.

Data points are colored according to the closest center.

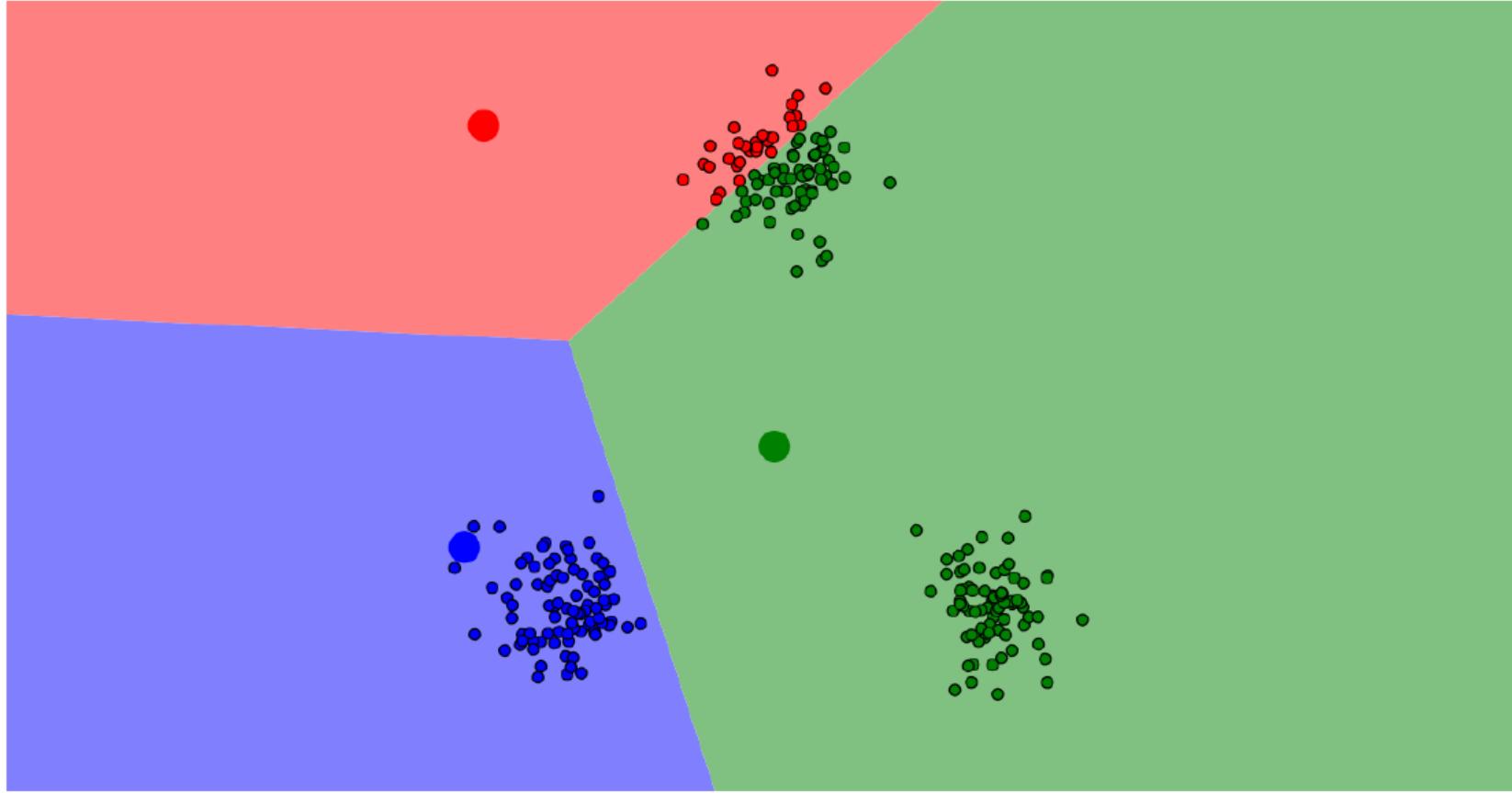
K-means Example: Step 1B



Each center is
then updated...

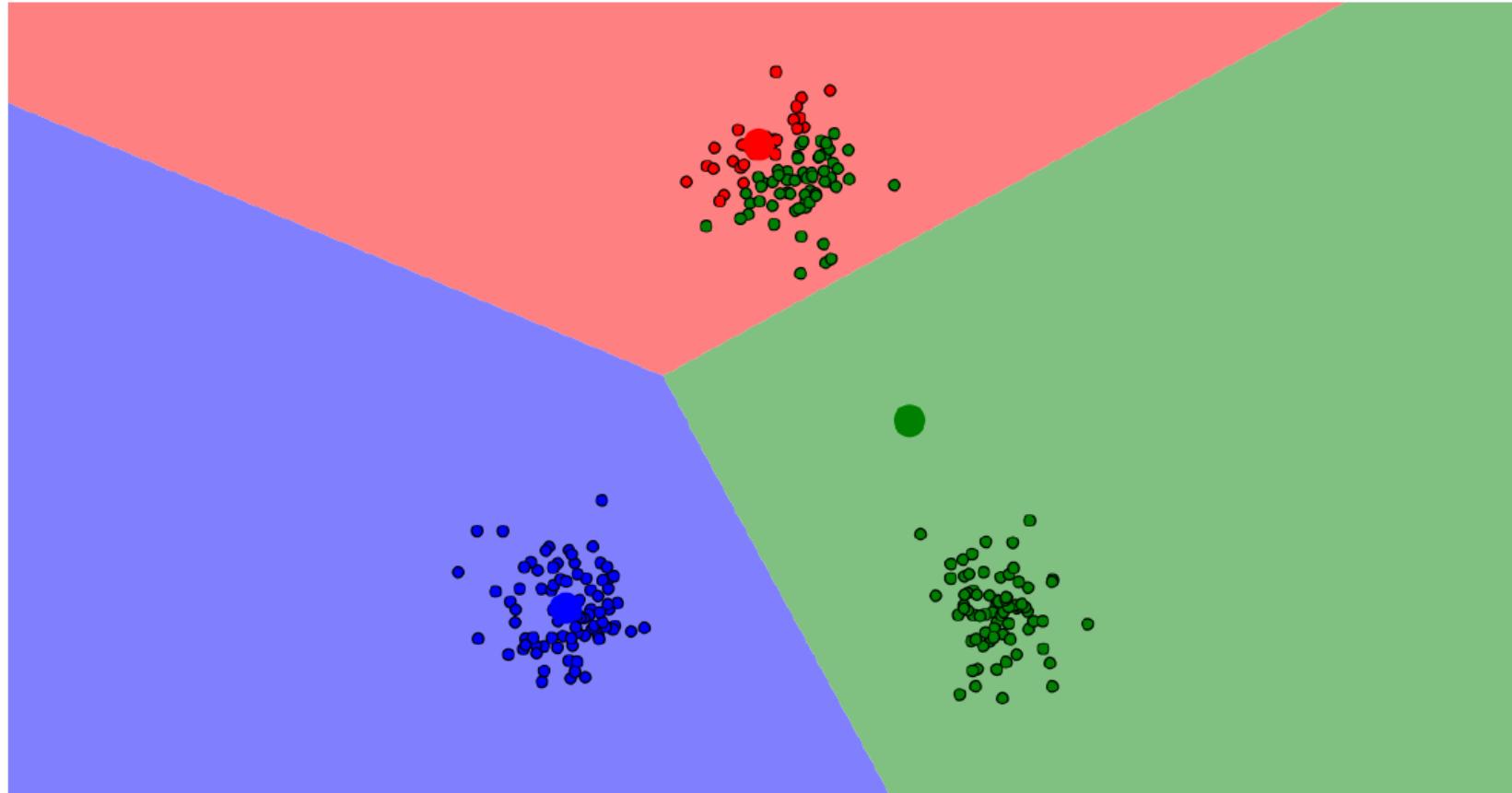
... using the mean
of all points
assigned to that
cluster.

K-means Example: Step 2A



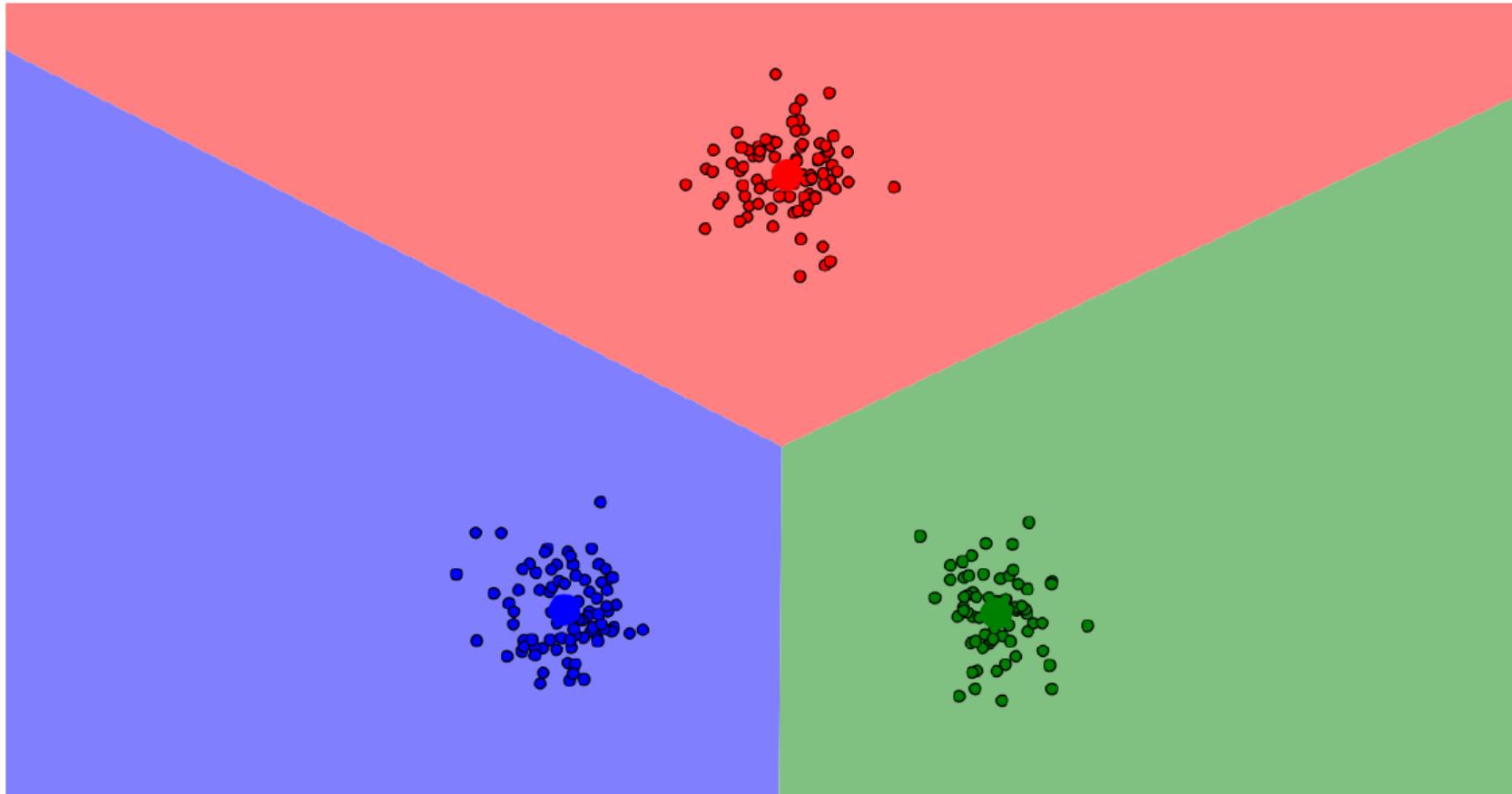
Data points are colored (again) according to the closest center.

K-means Example: Step 2B



Re-calculate all
cluster centers.

K-means Example: Converged



After repeating
these steps for
several more
iterations...

The centers
converge to a
stable solution!

These centers
define the final
clusters.

Other limitations of k-means

Assign observations to closest cluster center

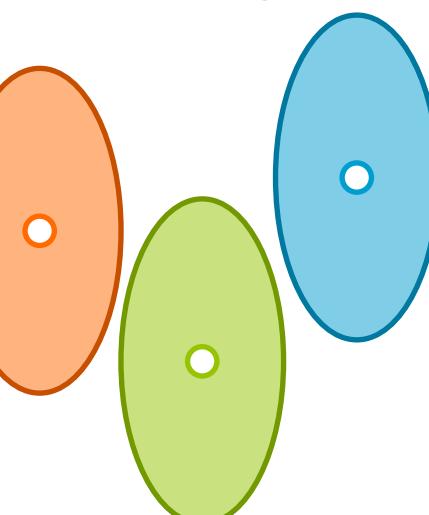
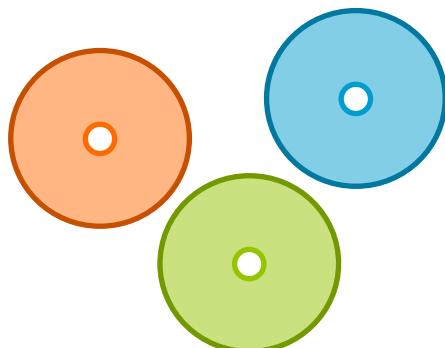
$$z_i \leftarrow \arg \min_j \| \mu_j - \mathbf{x}_i \|_2^2$$

Only center matters

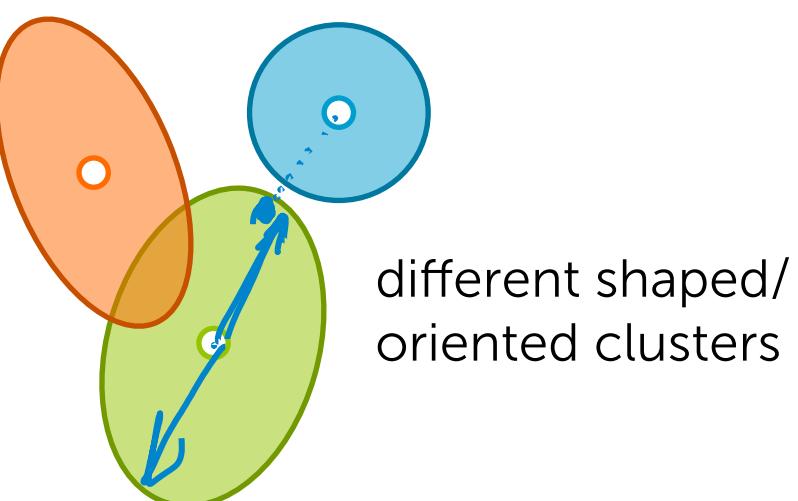
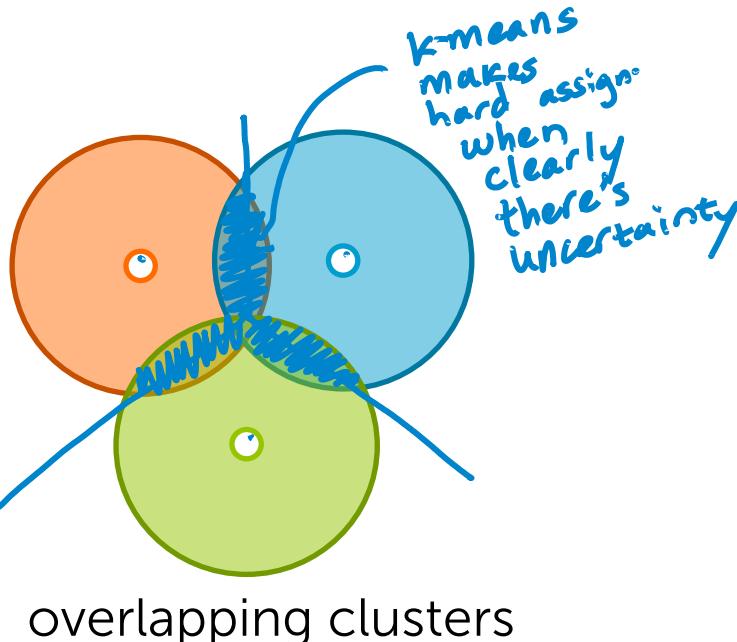
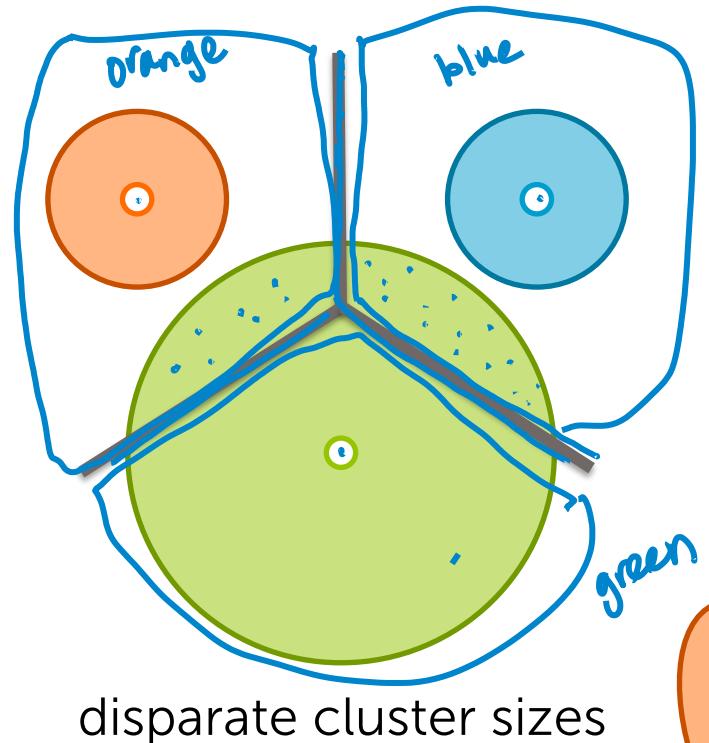
Can use weighted Euclidean,
but requires *known* weights

Still assumes all clusters have
the same axis-aligned ellipses

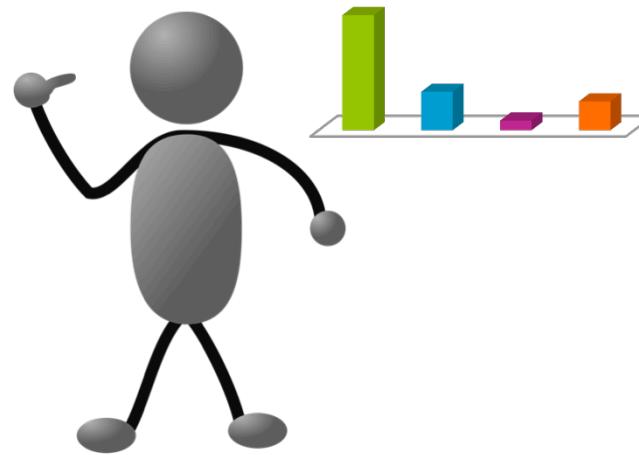
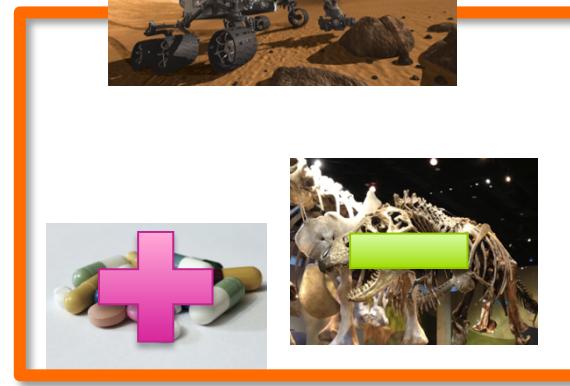
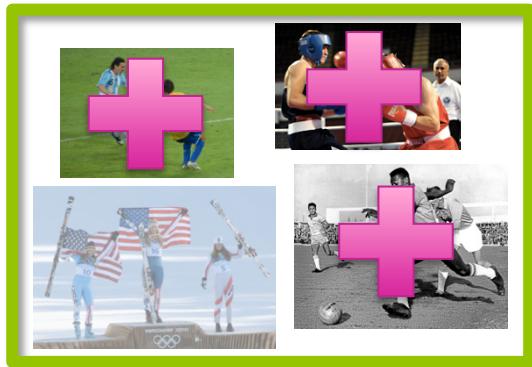
Equivalent to assuming
spherically symmetric clusters



Failure modes of k-means



Uncertainty in cluster assignments



Hard assignments
don't tell full story

Mixture models

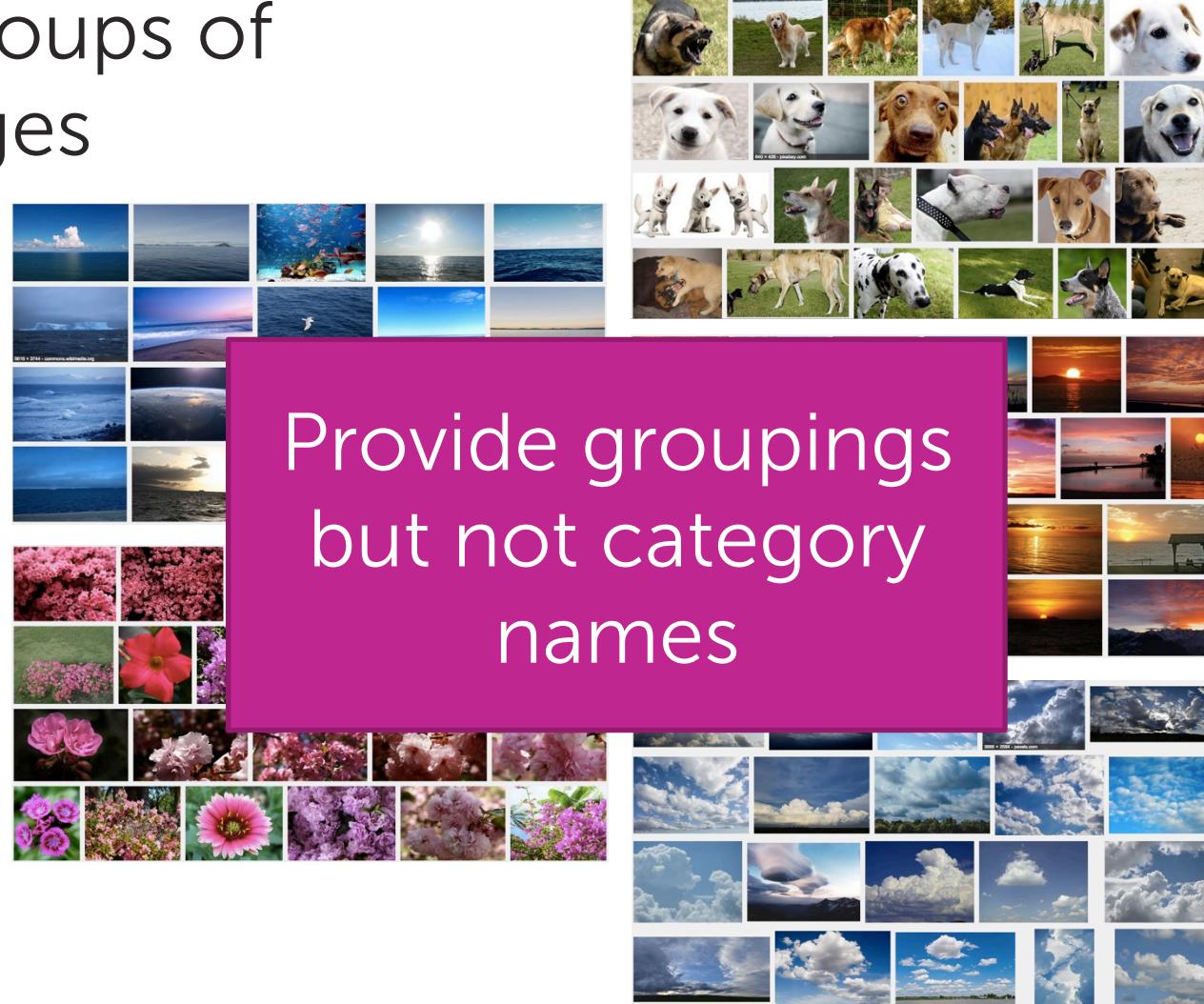
Mixture Models

- Provides **soft assignments** of observations to clusters (uncertainty in assignment)
 - e.g., 54% chance document is **world news**, 45% **science**, 1% **sports**, and 0% **entertainment**
- Accounts for cluster **shapes** not just **centers**
- Enables **learning weightings** of dimensions
 - e.g., how much to weight each word in the vocabulary when computing cluster assignment

Motivating application: Clustering images

Discover groups of similar images

- Ocean
- Pink flower
- Dog
- Sunset
- Clouds
- ...



Simple image representation

Consider average red, green, blue pixel intensities



[R = 0.05, G = 0.7, B = 0.9]



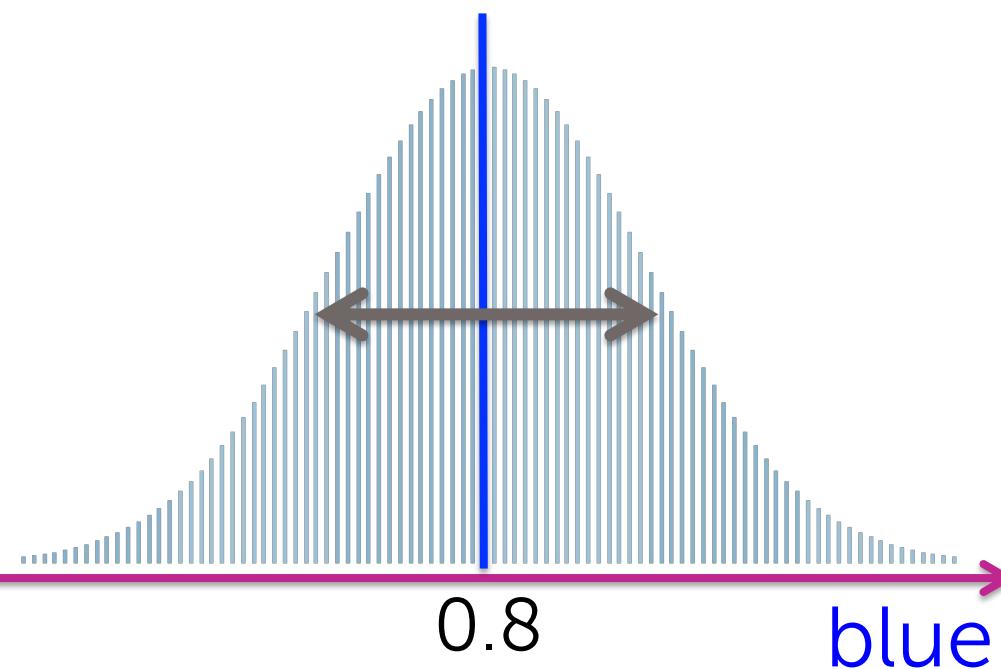
[R = 0.85, G = 0.05, B = 0.35]



[R = 0.02, G = 0.95, B = 0.4]

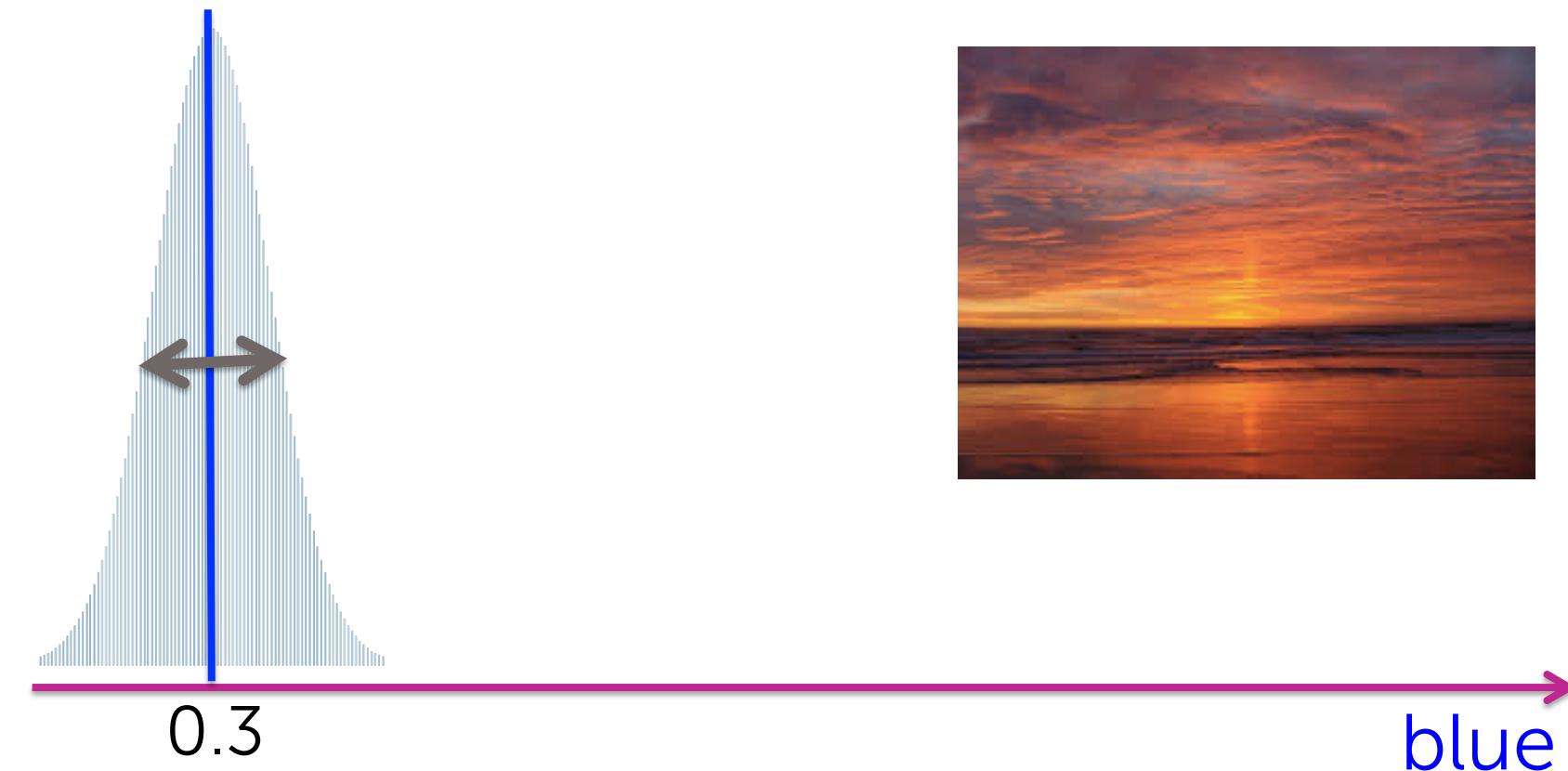
Distribution over all **cloud** images

Let's look at just the **blue** dimension



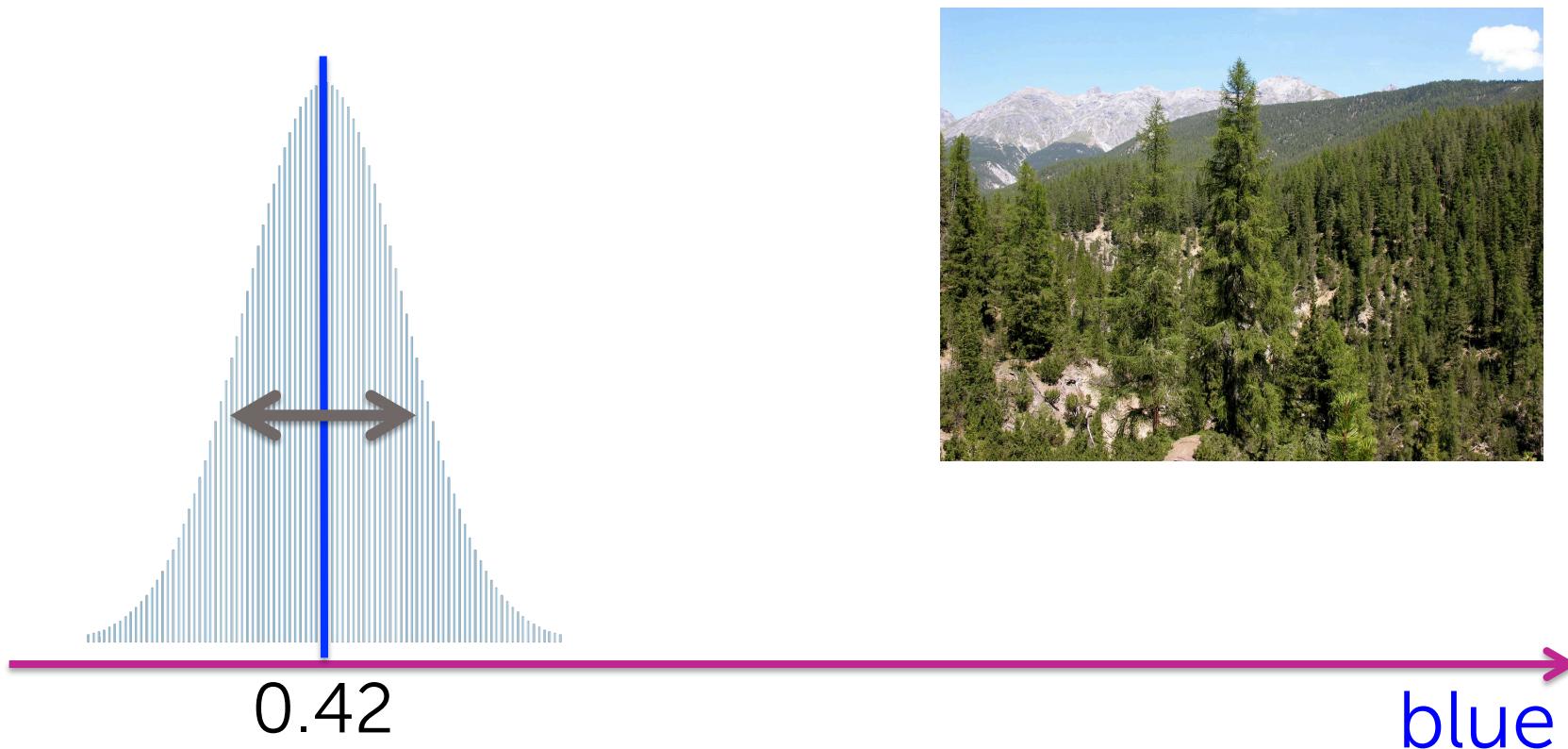
Distribution over all **sunset** images

Let's look at just the **blue** dimension

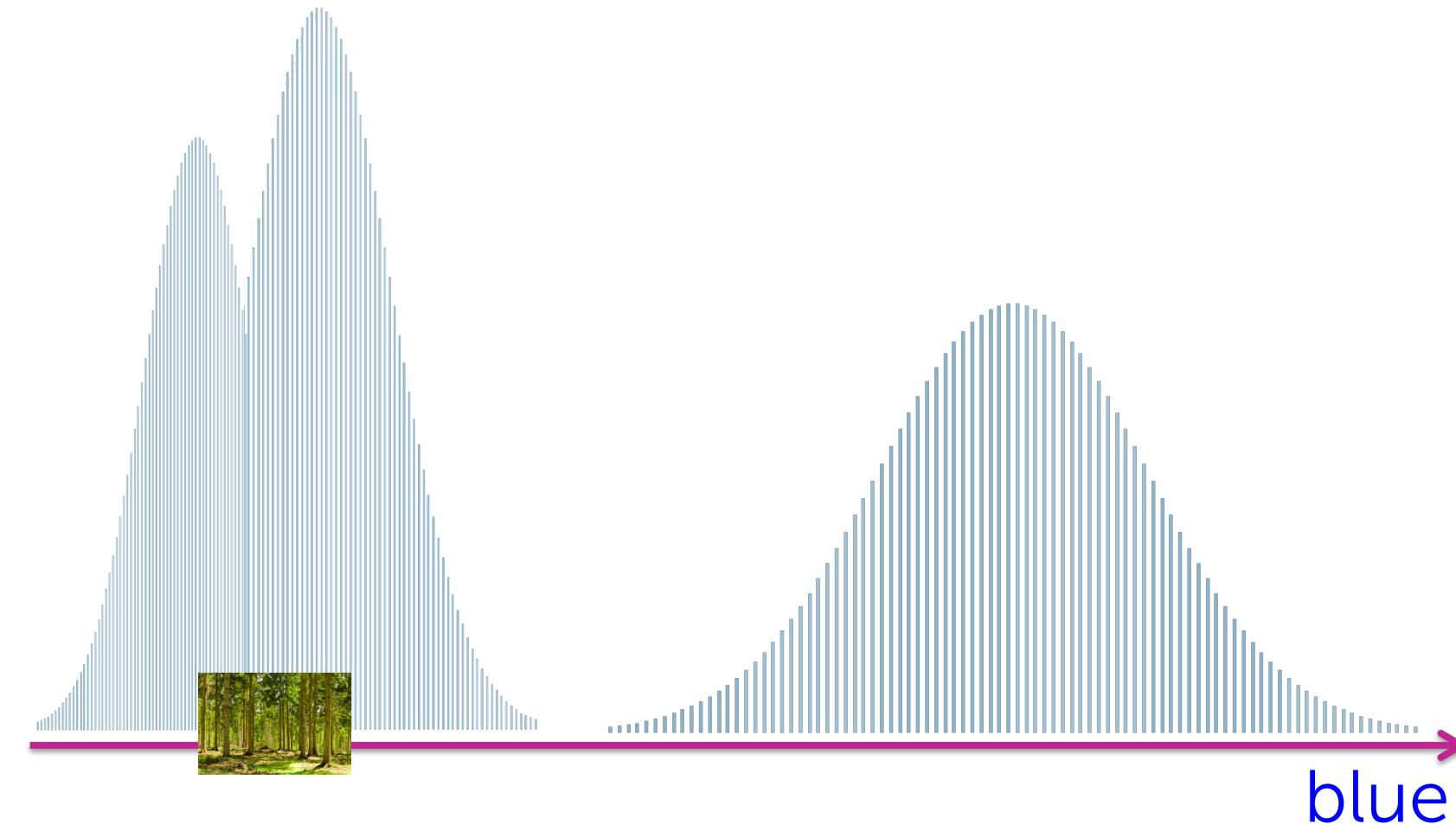


Distribution over all forest images

Let's look at just the **blue** dimension



Distribution over all images



Can be distinguished along other dim

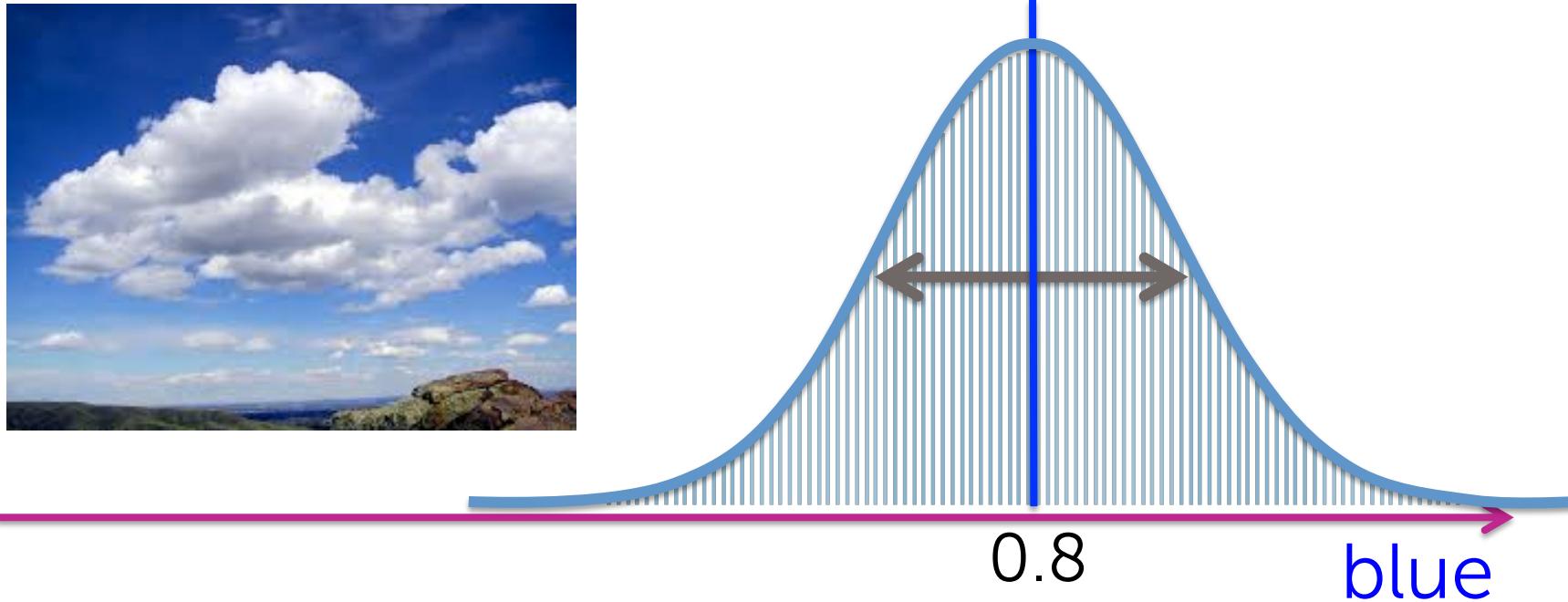
Now look at the **red** dimension



Background: Gaussian distributions

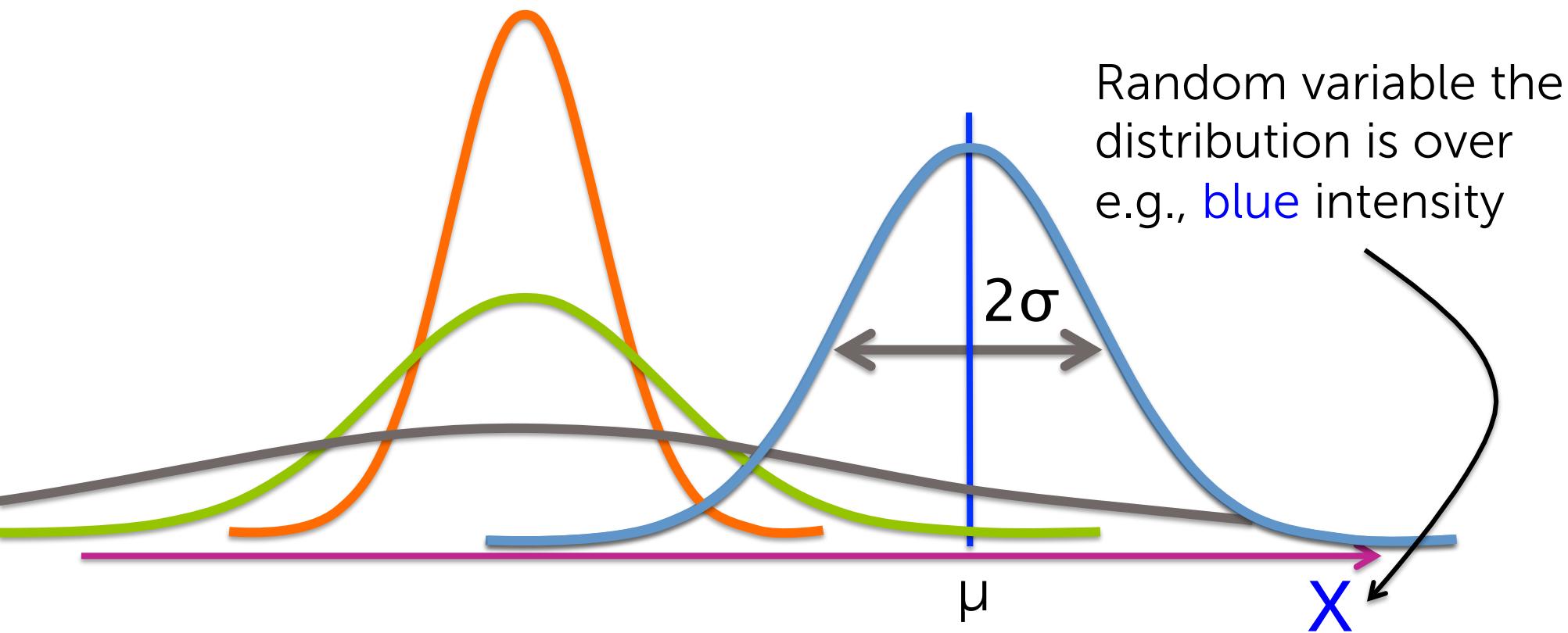
Model for a given image type

For **each dimension** of the [R, G, B] vector,
and **each image type**, assume a
Gaussian distribution over color intensity

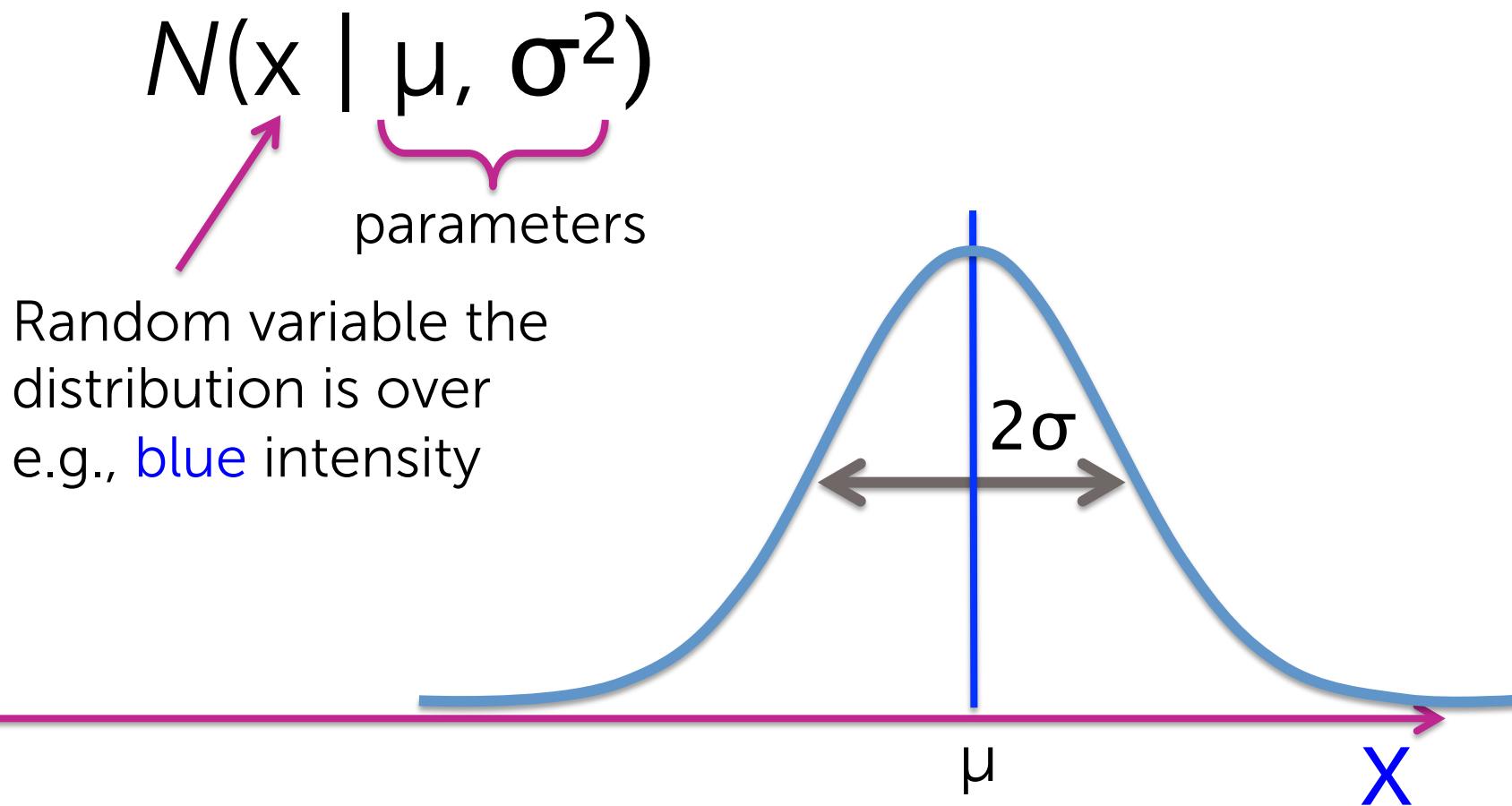


1D Gaussians

Fully specified by **mean** μ and **variance** σ^2
(or **standard deviation** σ)

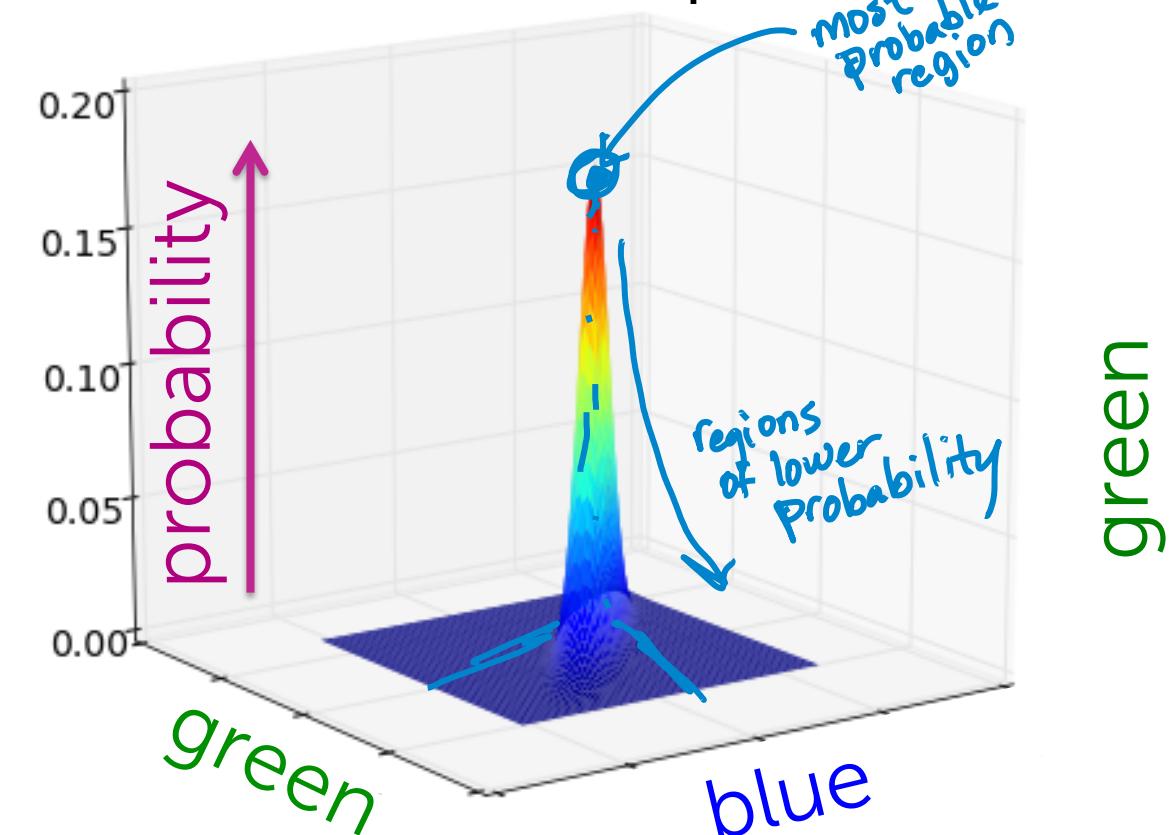


Notating a 1D Gaussian distribution

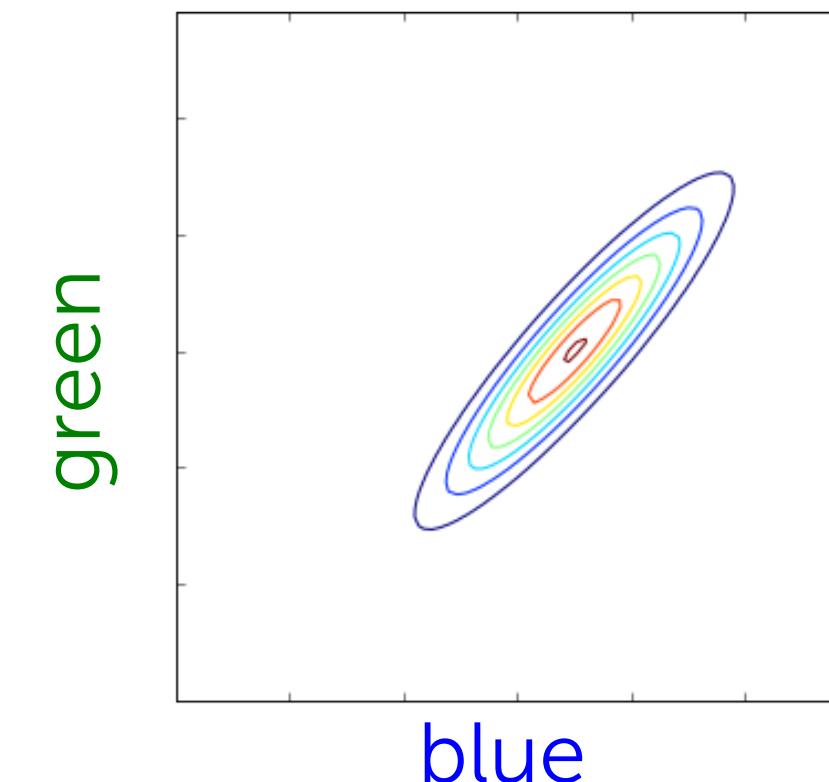


2D Gaussians – Bird's eye view

3D mesh plot



Contour plot

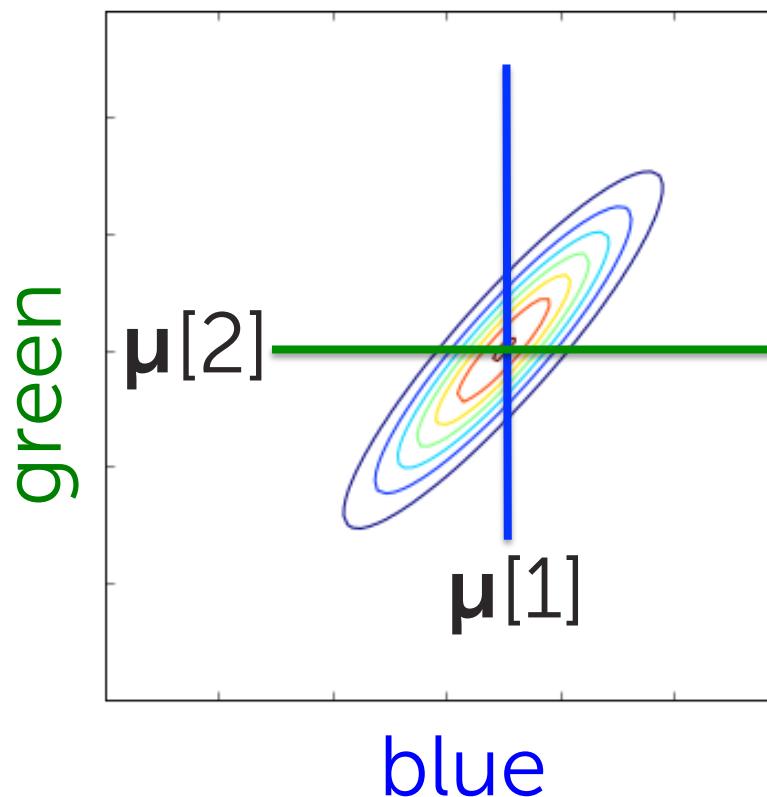


2D Gaussians – Parameters

Fully specified by **mean μ** and **covariance Σ**

$$\mu = [\mu_{\text{blue}}, \mu_{\text{green}}]$$

mean centers the distribution in 2D



2D Gaussians – Parameters

Fully specified by **mean** μ and **covariance** Σ

$$\mu = [\mu_{\text{blue}}, \mu_{\text{green}}]$$

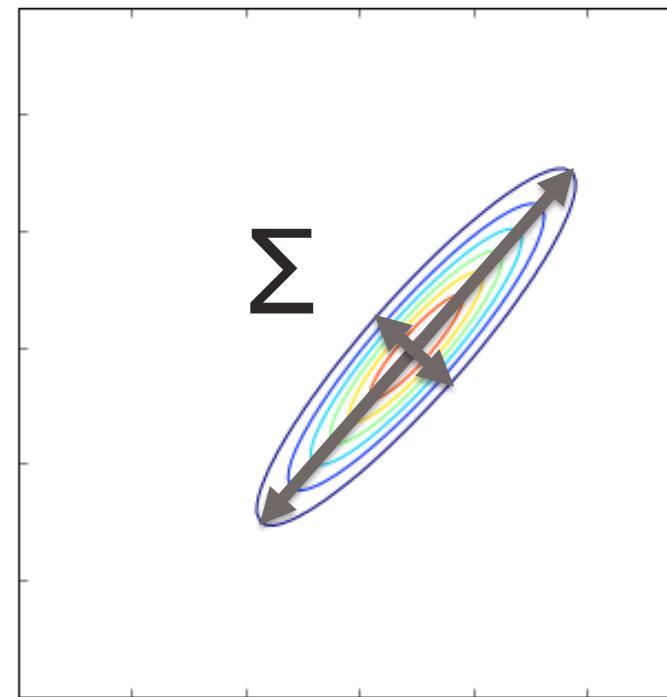
$$\Sigma = \begin{pmatrix} \sigma_{\text{blue}}^2 & \sigma_{\text{blue},\text{green}} \\ \sigma_{\text{green},\text{blue}} & \sigma_{\text{green}}^2 \end{pmatrix}$$

covariance determines
orientation + spread

green

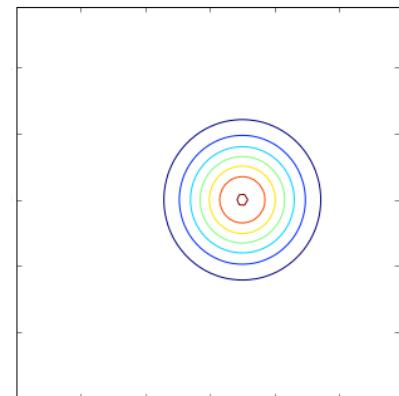
Σ

blue

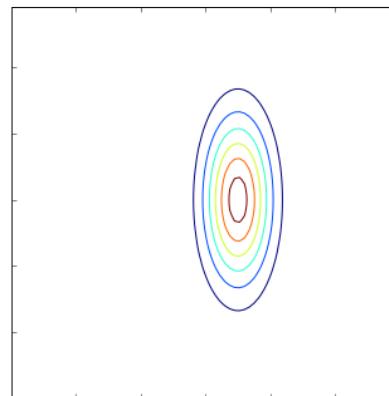


Covariance structures

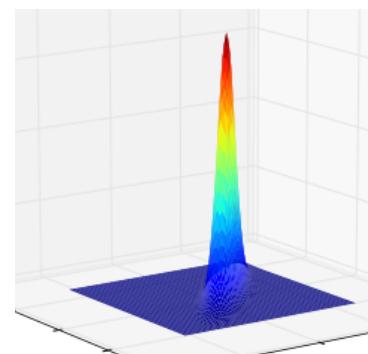
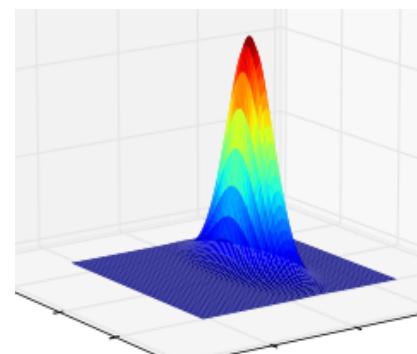
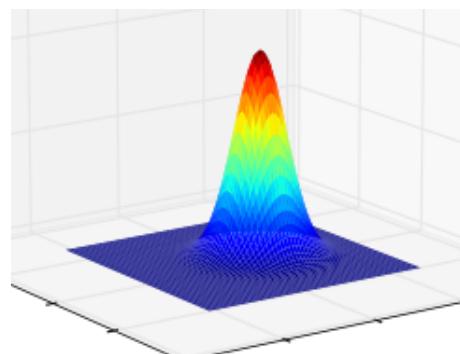
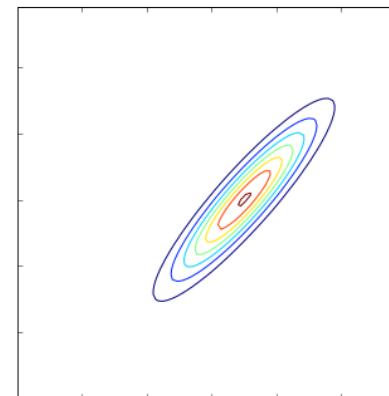
$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} \sigma_B^2 & 0 \\ 0 & \sigma_G^2 \end{pmatrix}$$



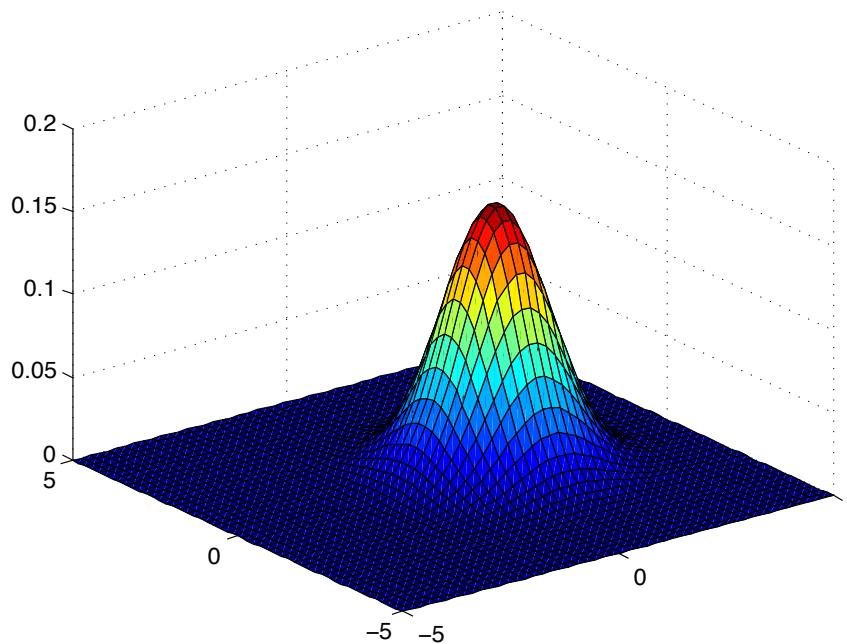
$$\Sigma = \begin{pmatrix} \sigma_B^2 & \sigma_{B,G} \\ \sigma_{G,B} & \sigma_G^2 \end{pmatrix}$$



Notating a multivariate Gaussian

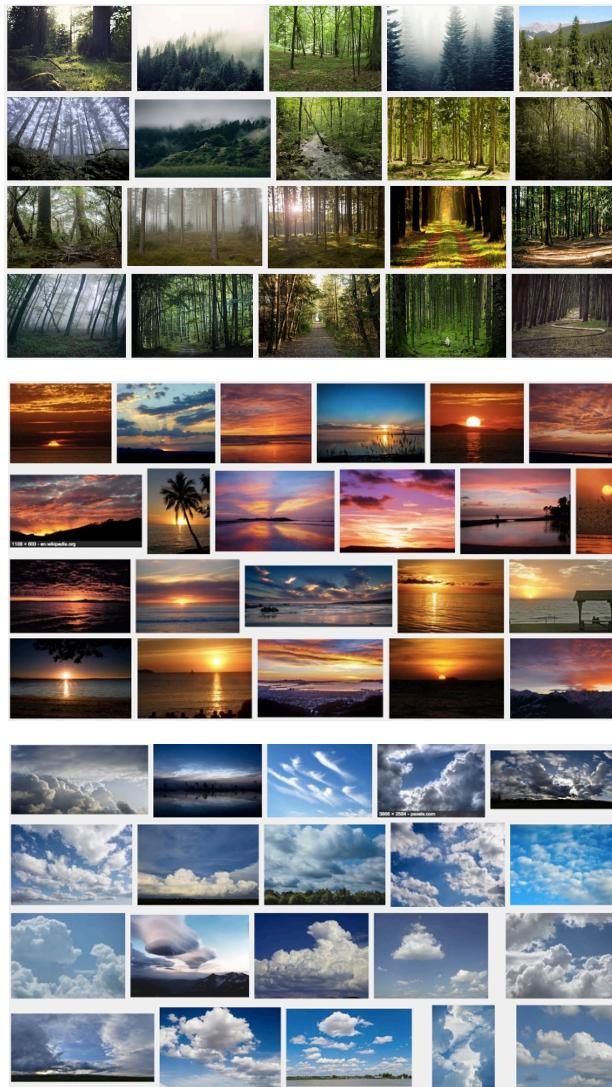
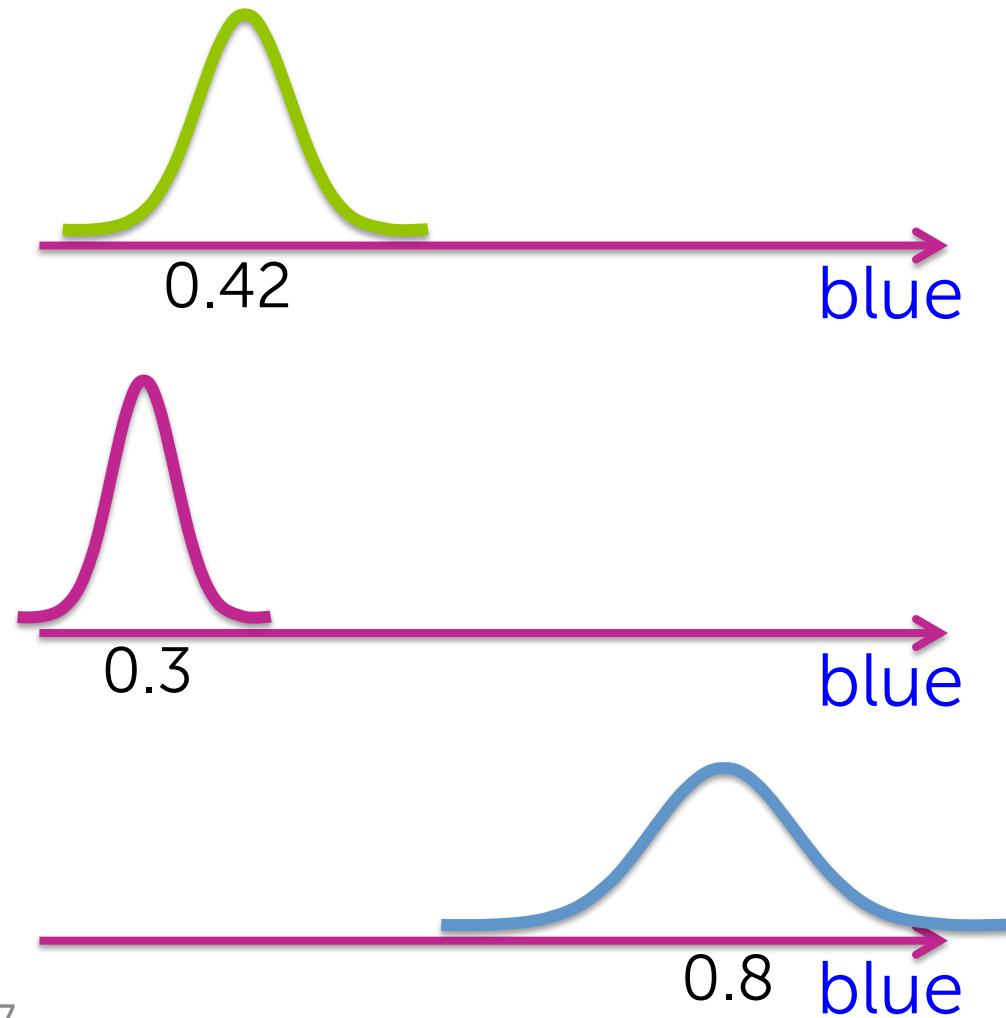
$$N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Random **vector**
e.g., [R, G, B] intensities



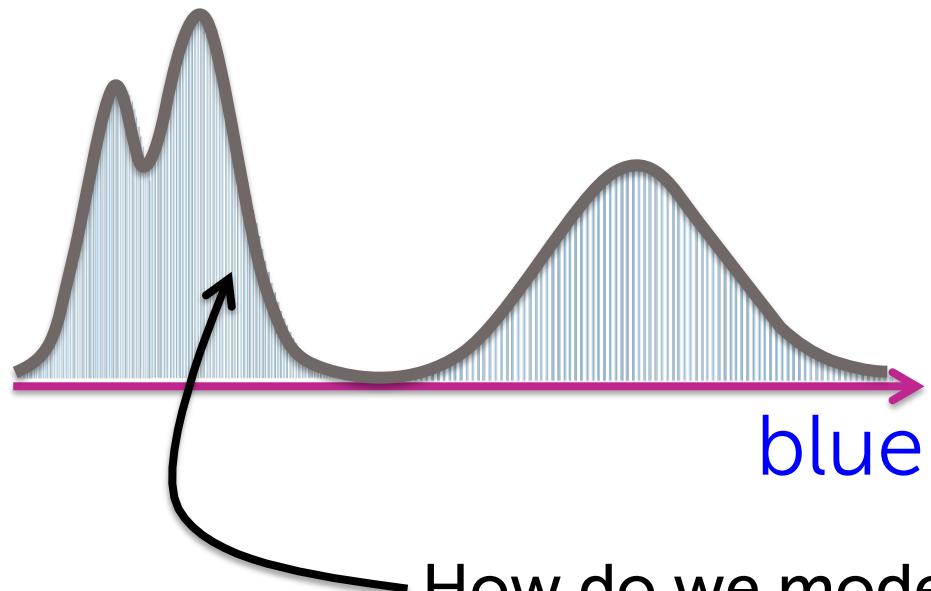
Mixture of Gaussians

Model as Gaussian per category/cluster



Jumble of unlabeled images

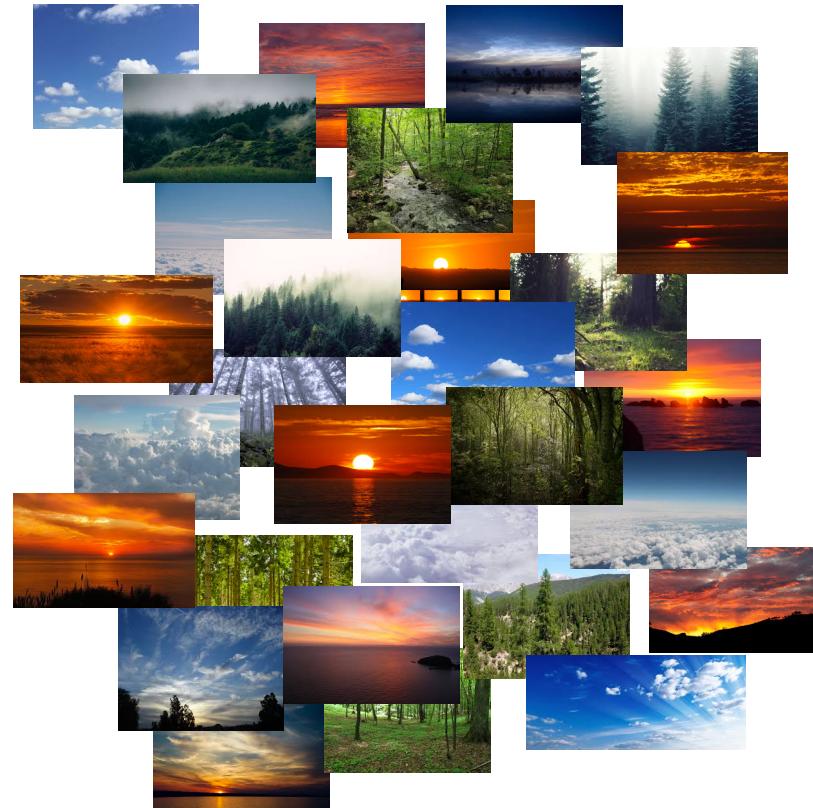
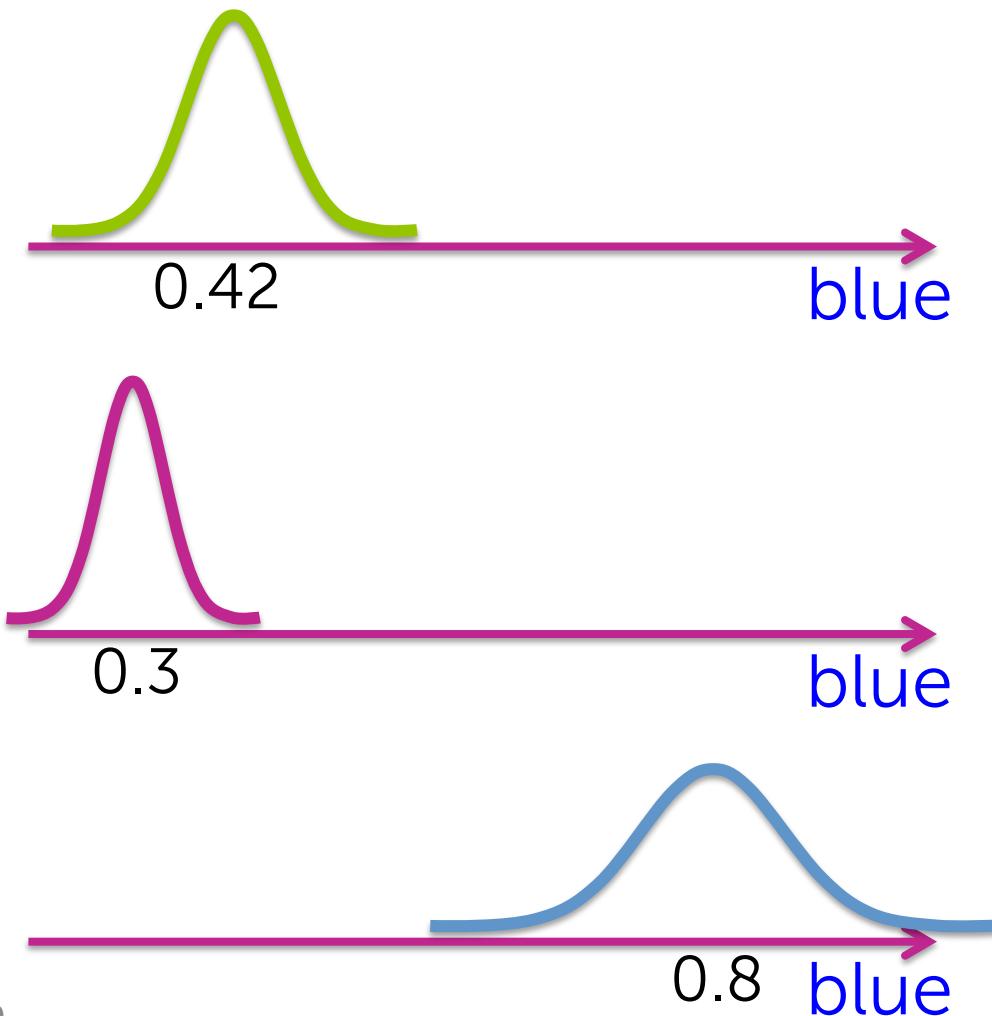
HISTOGRAM



How do we model
this distribution?

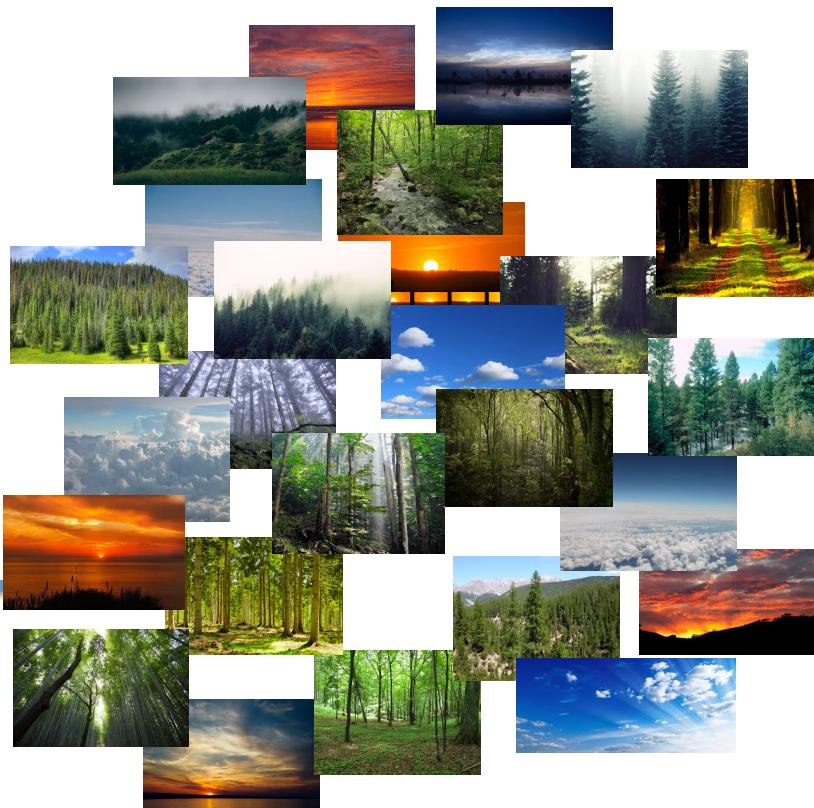
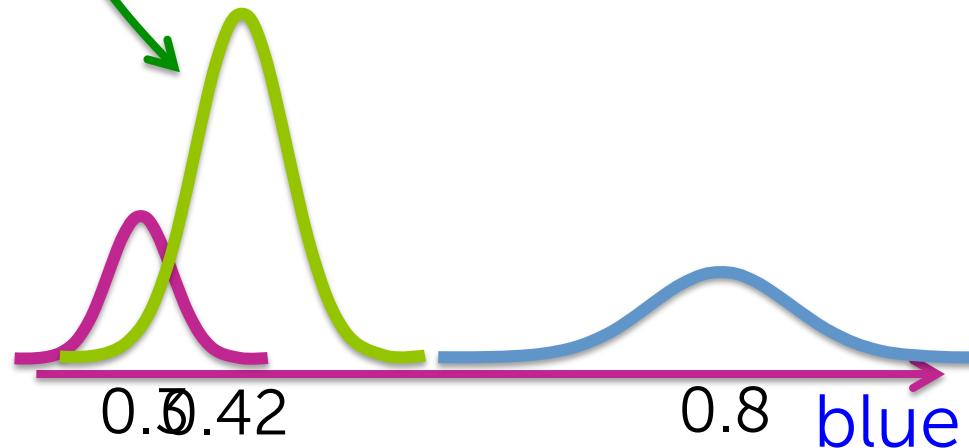


Model of jumble of unlabeled images



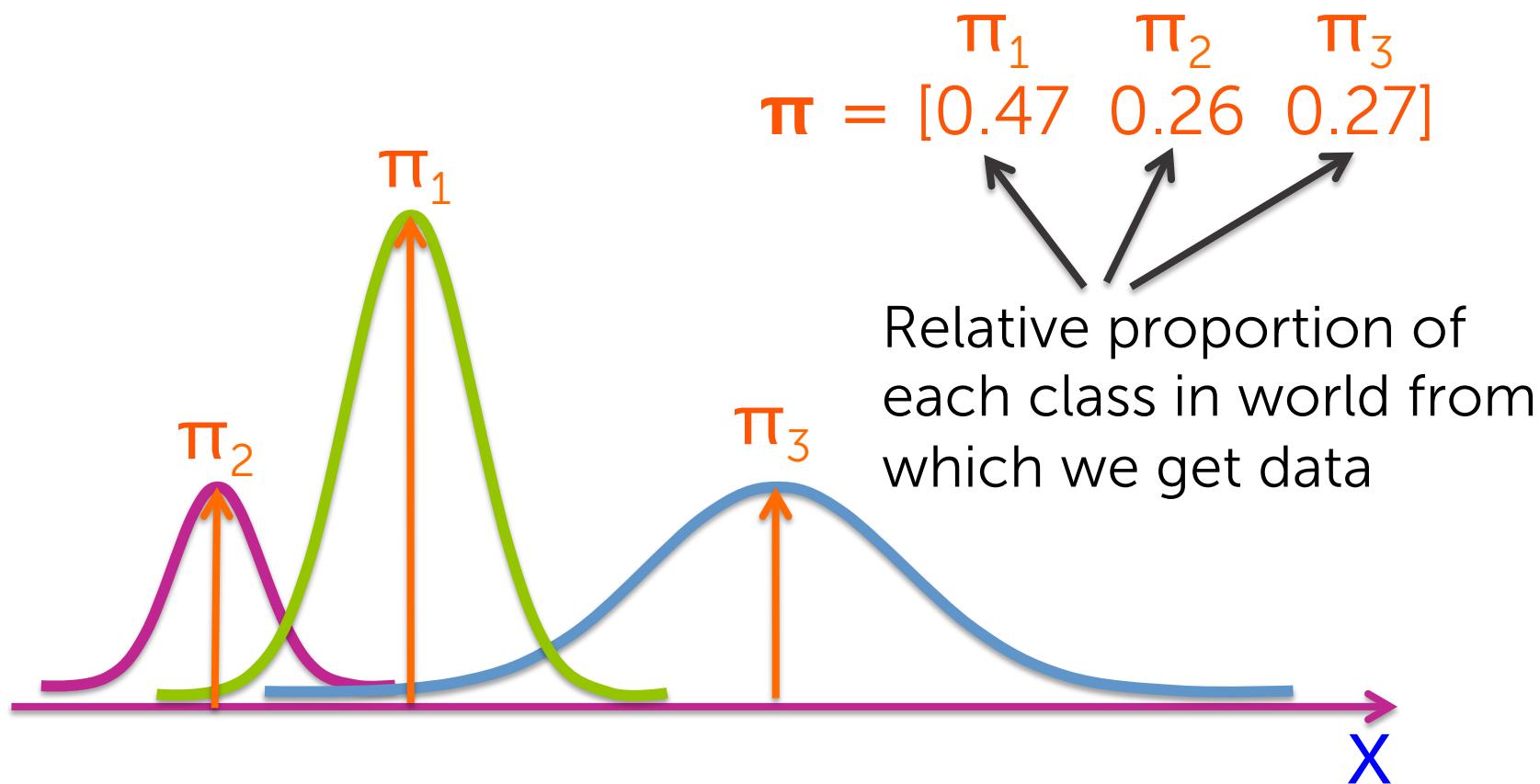
What if image types not equally represented?

e.g., forest images are very likely in the collection



Combination of weighted Gaussians

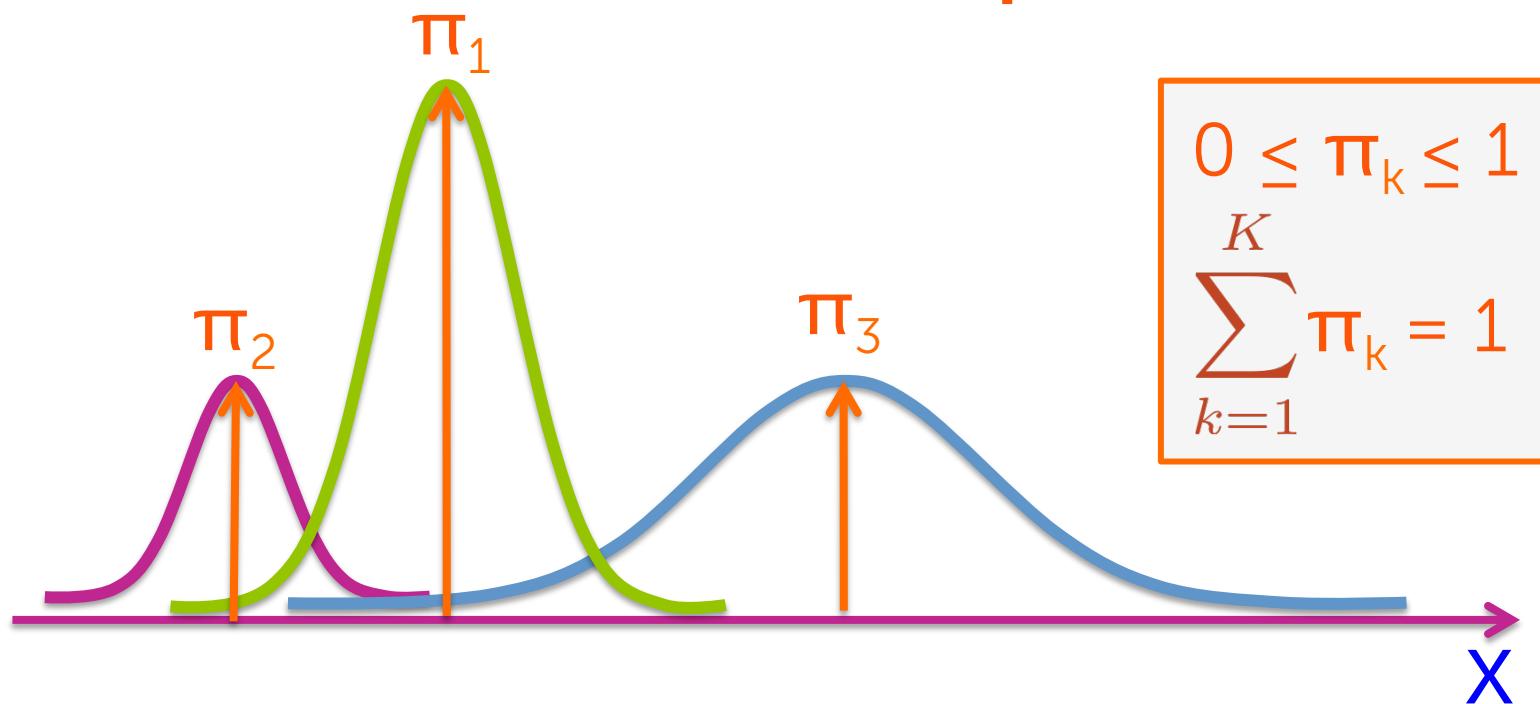
Associate a weight π_k with each Gaussian component



Combination of weighted Gaussians

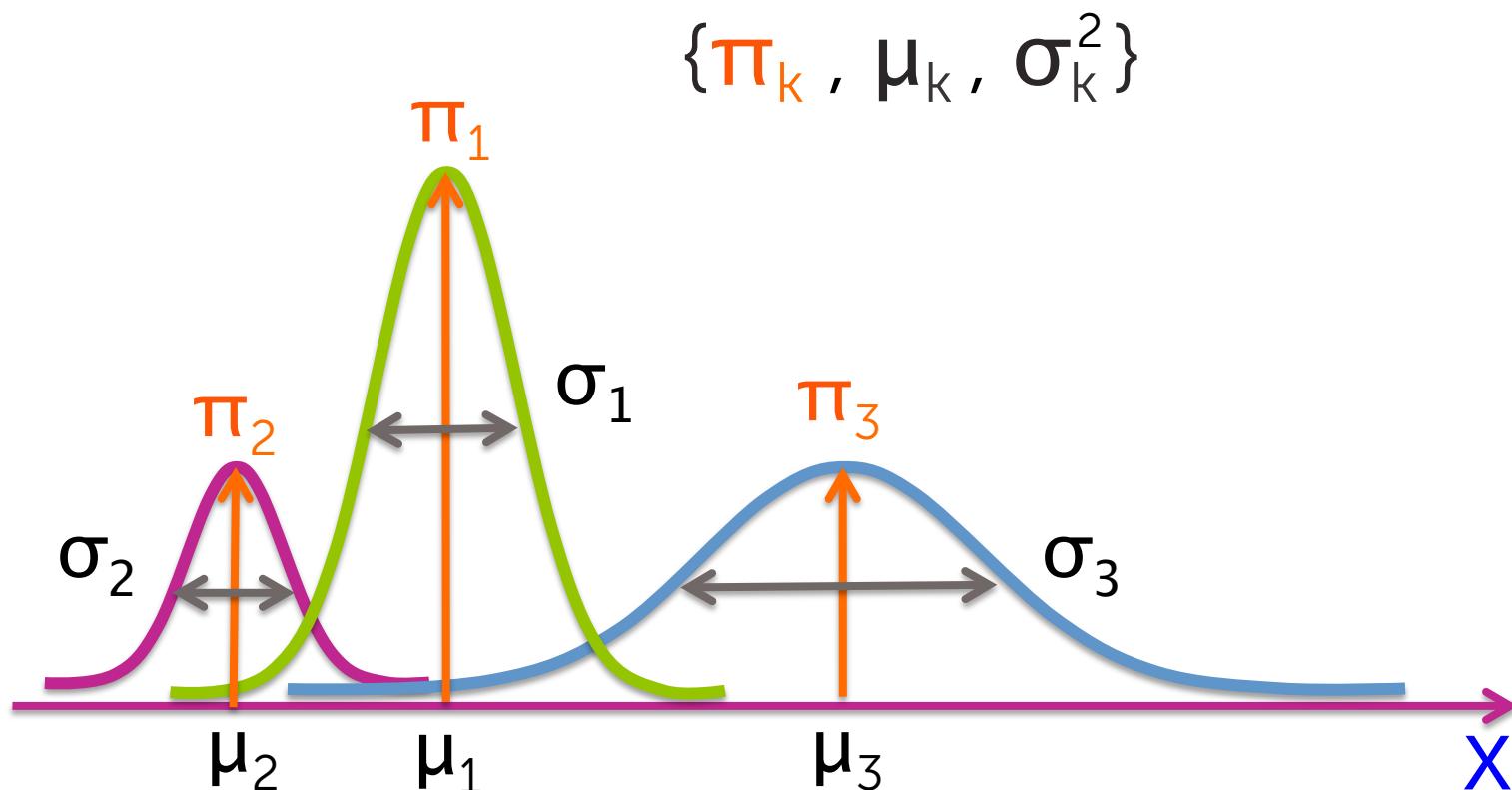
Associate a weight π_k with each Gaussian component

$$\boldsymbol{\pi} = [\pi_1 \quad \pi_2 \quad \pi_3] = [0.47 \quad 0.26 \quad 0.27]$$

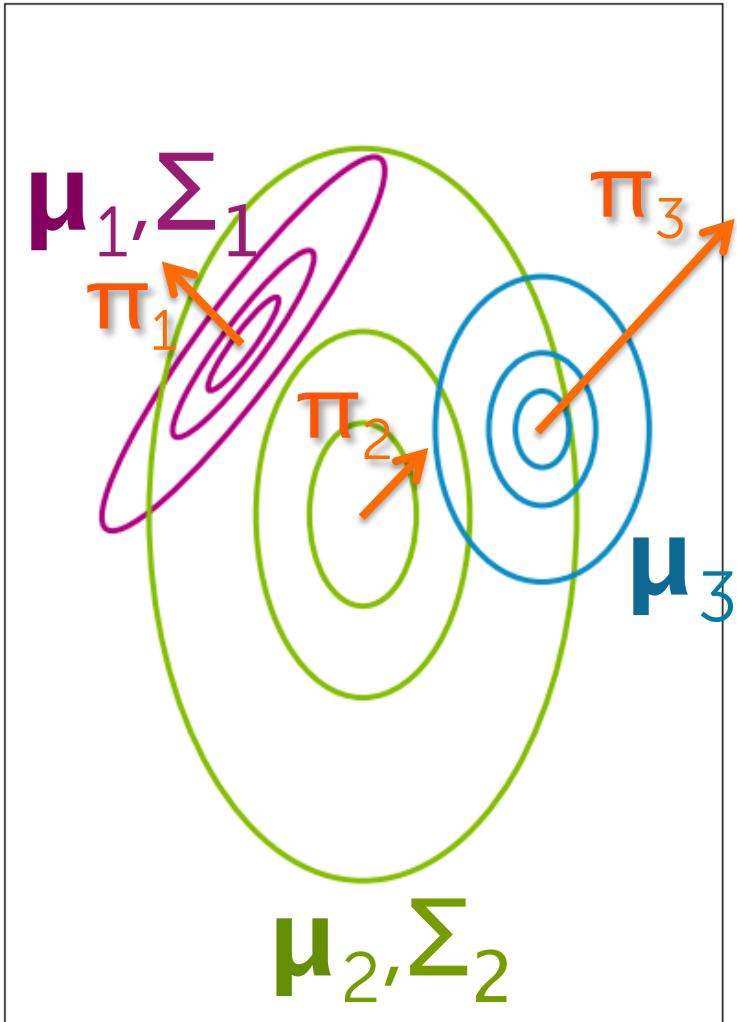


Mixture of Gaussians (1D)

Each mixture component represents a unique cluster specified by:



Mixture of Gaussians (general)



Each mixture component represents a unique cluster specified by:

$$\{\pi_k, \mu_k, \Sigma_k\}$$

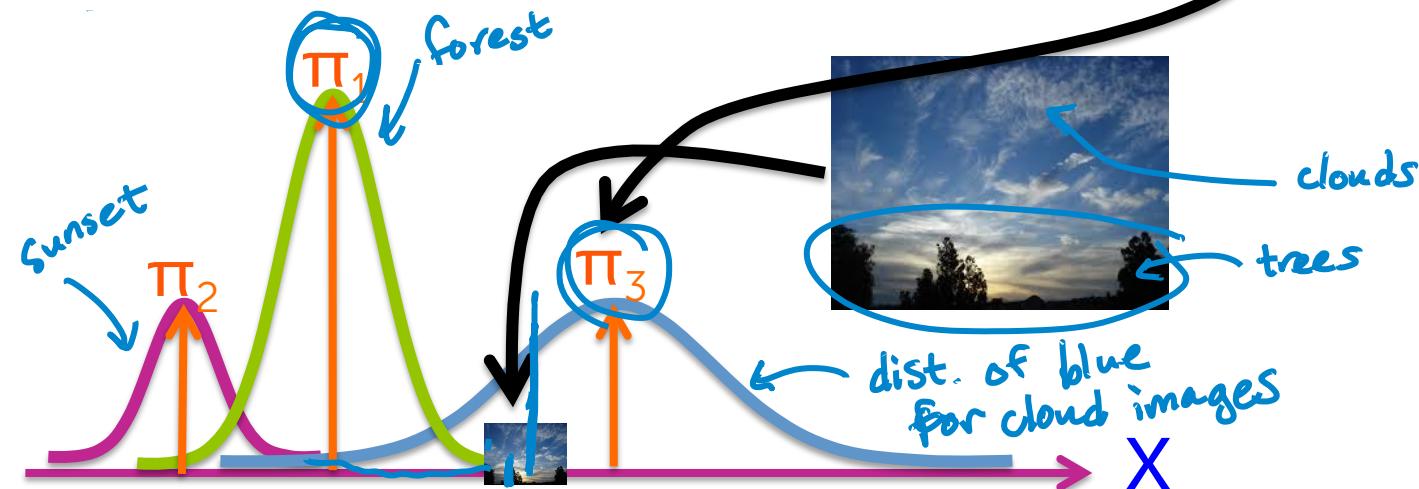
According to the model...

Without observing the image content, what's the probability it's from cluster k? (e.g., prob. of seeing "clouds" image)

$$p(z_i = k) = \pi_k$$

Given observation \mathbf{x}_i is from cluster k, what's the likelihood of seeing \mathbf{x}_i ? (e.g., just look at distribution for "clouds")

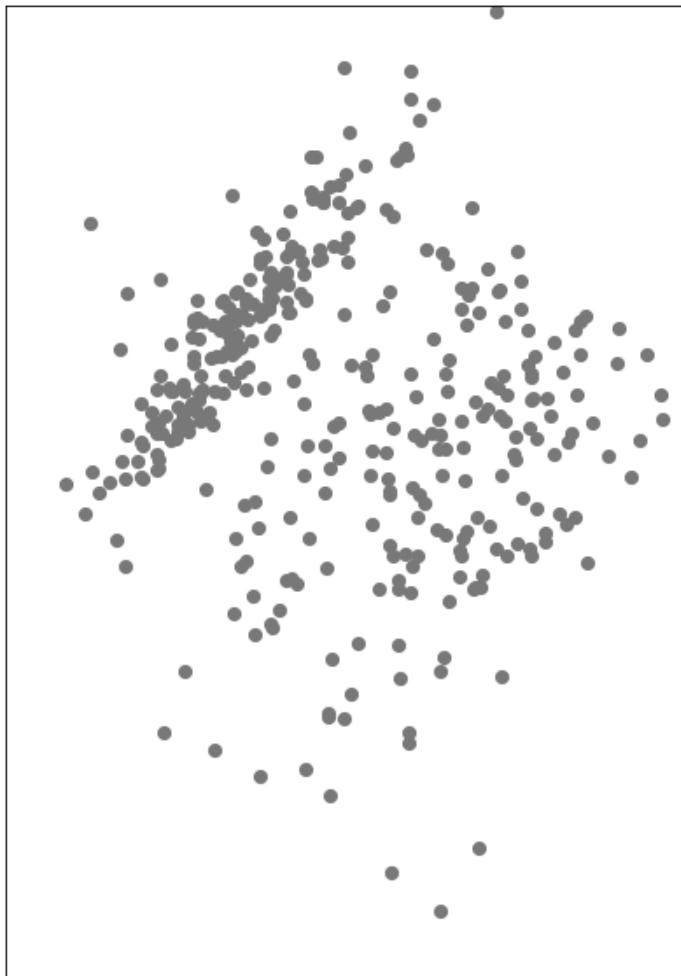
$$p(x \mid z_i = k, \mu_k, \Sigma_k) = N(x_i \mid \mu_k, \Sigma_k)$$



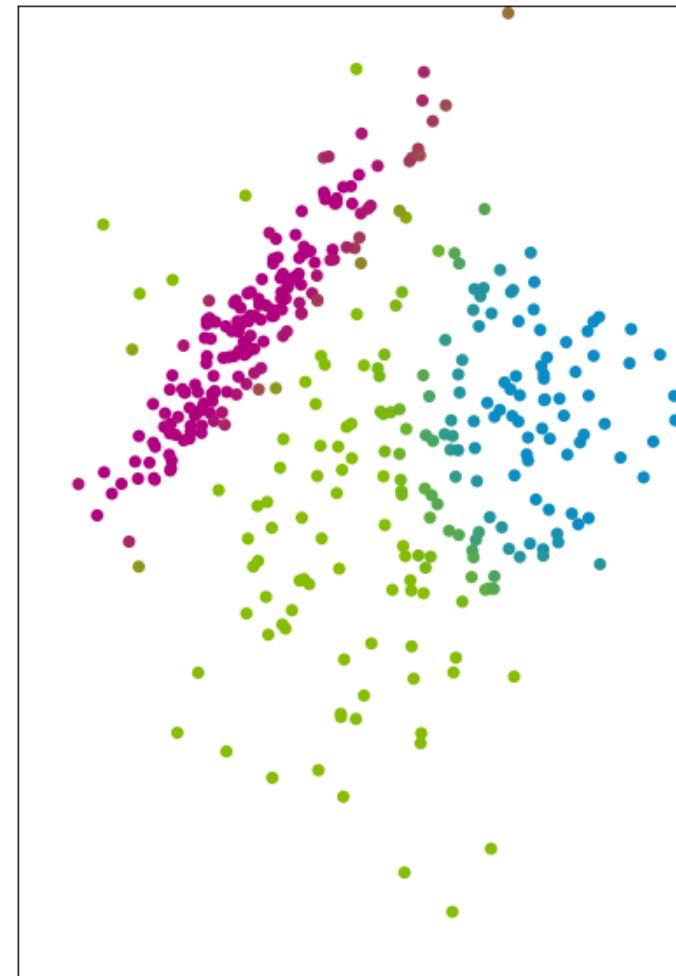
Inferring soft assignments with expectation maximization (EM)

Inferring cluster labels

Data

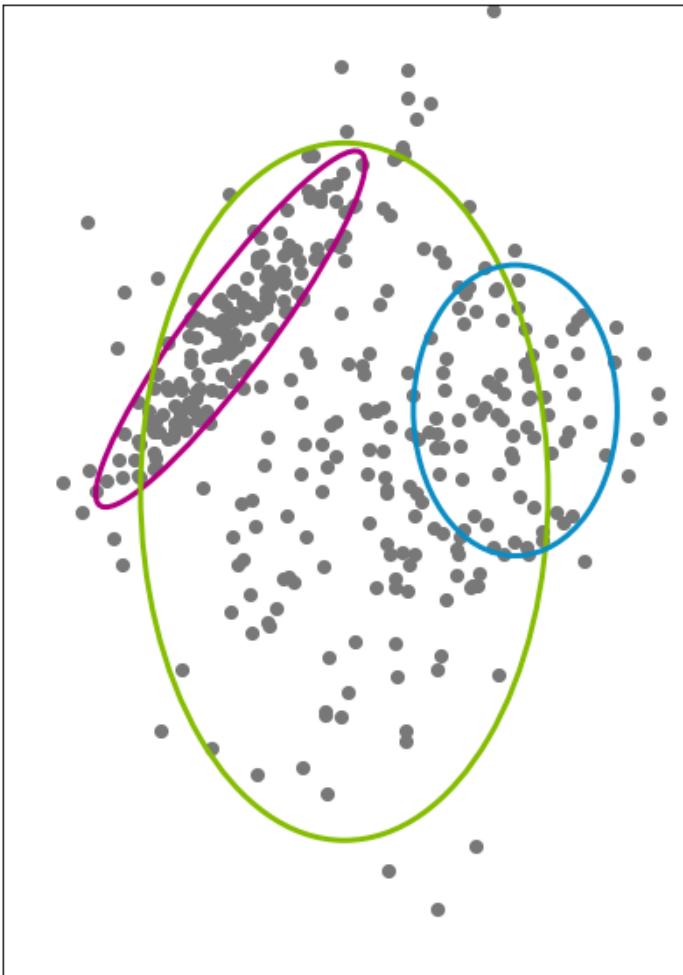


Desired soft assignments



Part 1:
What if we knew the cluster
parameters $\{\pi_k, \mu_k, \Sigma_k\}$?

Compute responsibilities



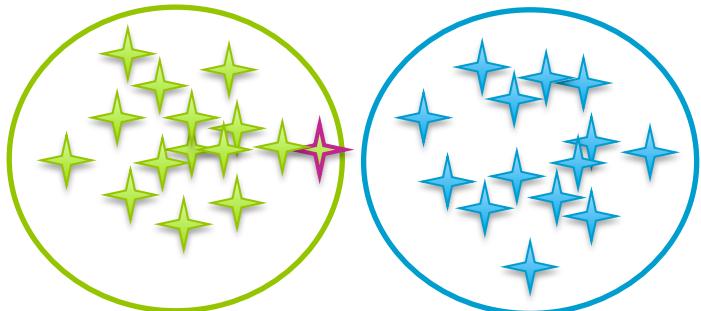
$$r_{ik} = p(z_i = k \mid \{\pi_j, \mu_j, \Sigma_j\}_{j=1}^K, x_i)$$

Responsibility cluster k takes for observation i

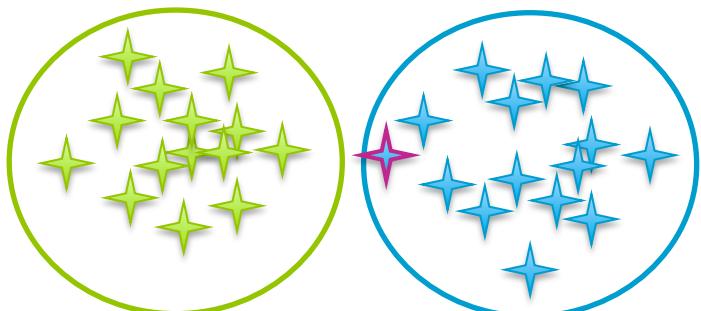
probability of assignment to cluster k

given model parameters and observed value

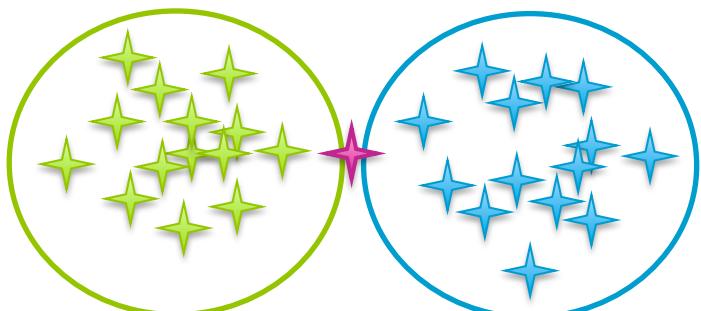
Responsibilities in pictures



Green cluster
takes more
responsibility



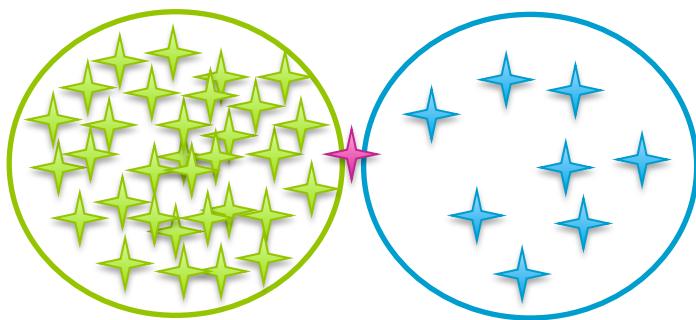
Blue cluster
takes more
responsibility



Uncertain...
split
responsibility

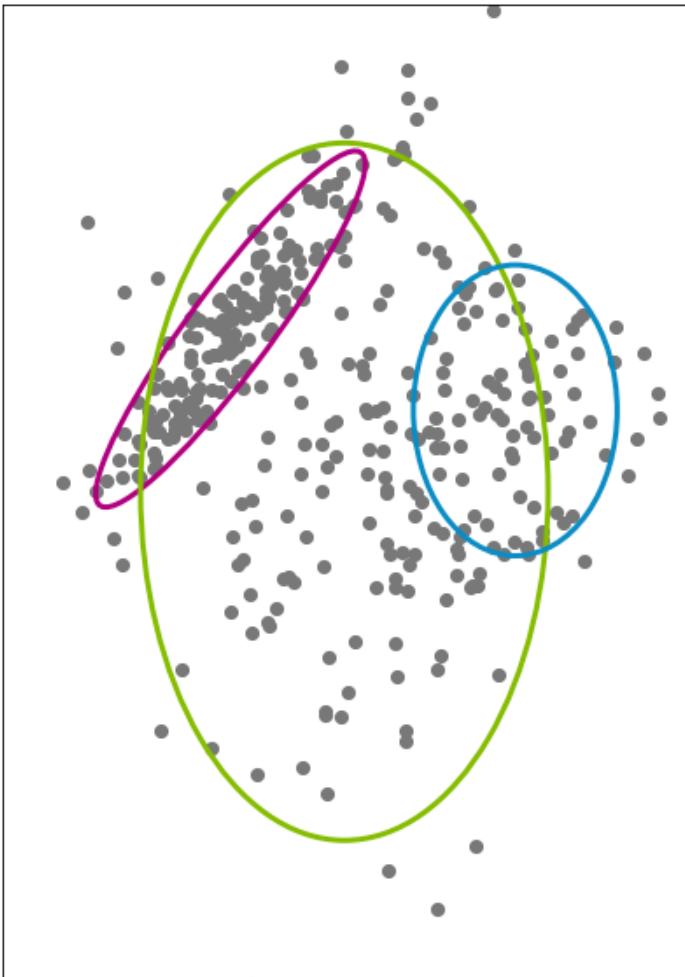
Responsibilities in pictures

Need to weight by cluster probabilities,
not just cluster shapes



Still **uncertain**,
but **green** cluster seems
more probable...
takes more responsibility

Responsibilities in equations



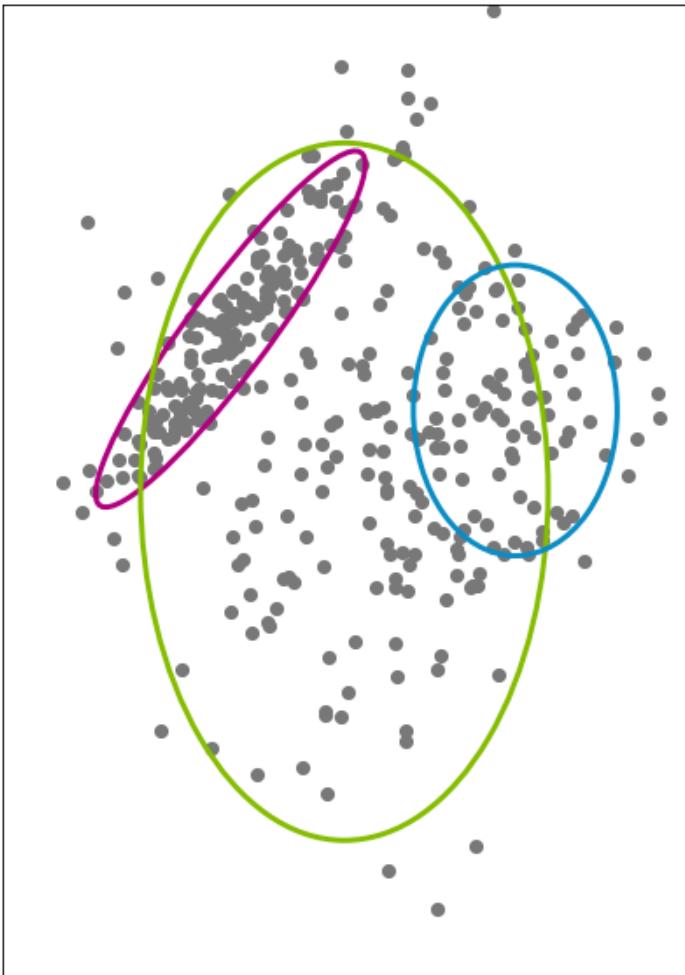
$$r_{ik} = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\text{How likely is the observed value } x_i \text{ under this cluster assignment?}}$$

Initial probability of being from cluster k

Responsibility cluster k takes for observation i

very unlikely under the green cluster, even though the prior on green is higher

Responsibilities in equations



Responsibility cluster k takes for observation i

$$r_{ik} = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_i | \mu_j, \Sigma_j)}$$

Normalized over all possible cluster assignments

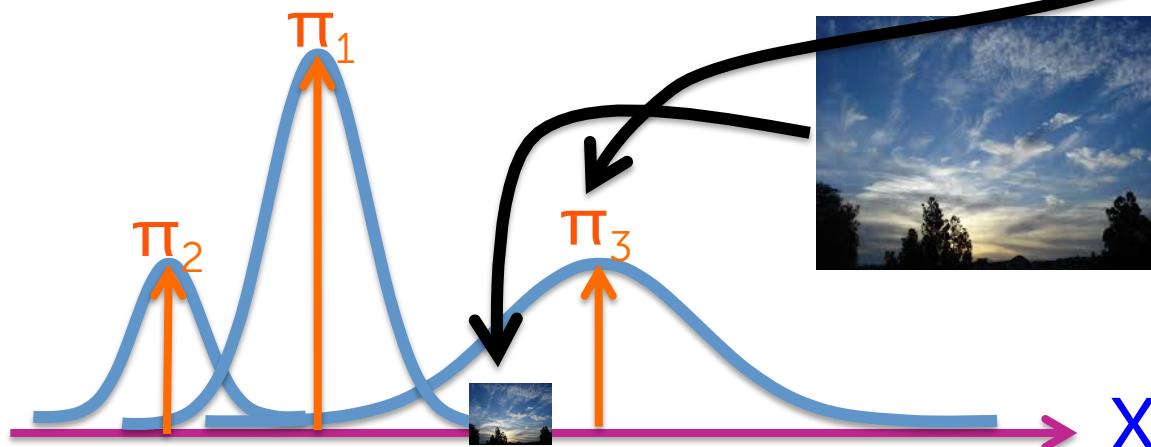
Recall: According to the model...

Without observing the image content, what's the probability it's from cluster k? (e.g., prob. of seeing "clouds" image)

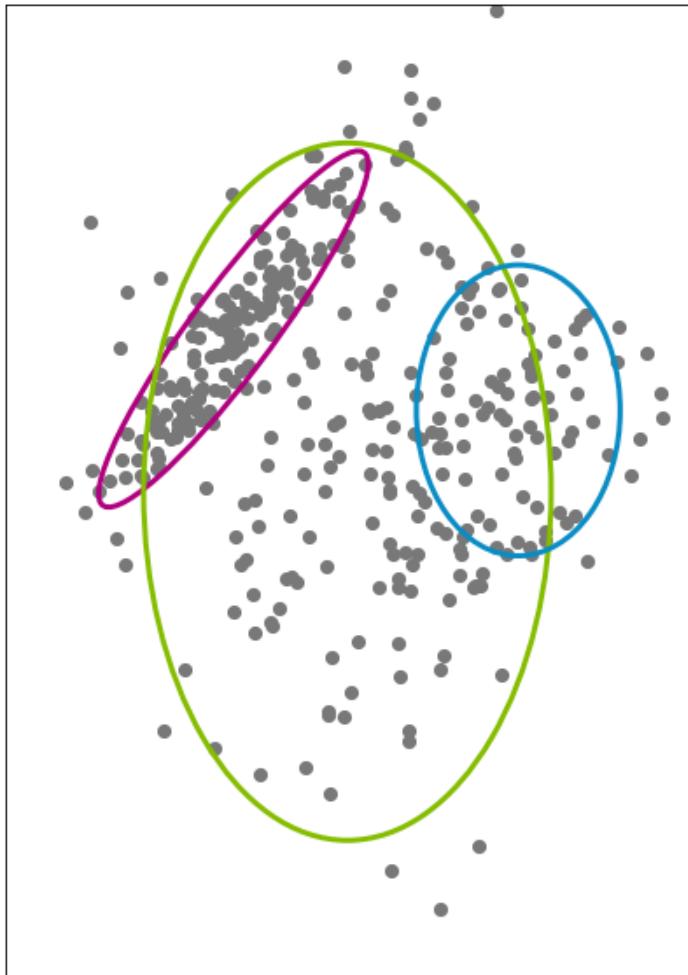
$$p(z_i = k) = \pi_k$$

Given observation \mathbf{x}_i is from cluster k, what's the likelihood of seeing \mathbf{x}_i ? (e.g., just look at distribution for "clouds")

$$p(x_i | z_i = k, \mu_k, \Sigma_k) = N(x_i | \mu_k, \Sigma_k)$$



Part 1 summary

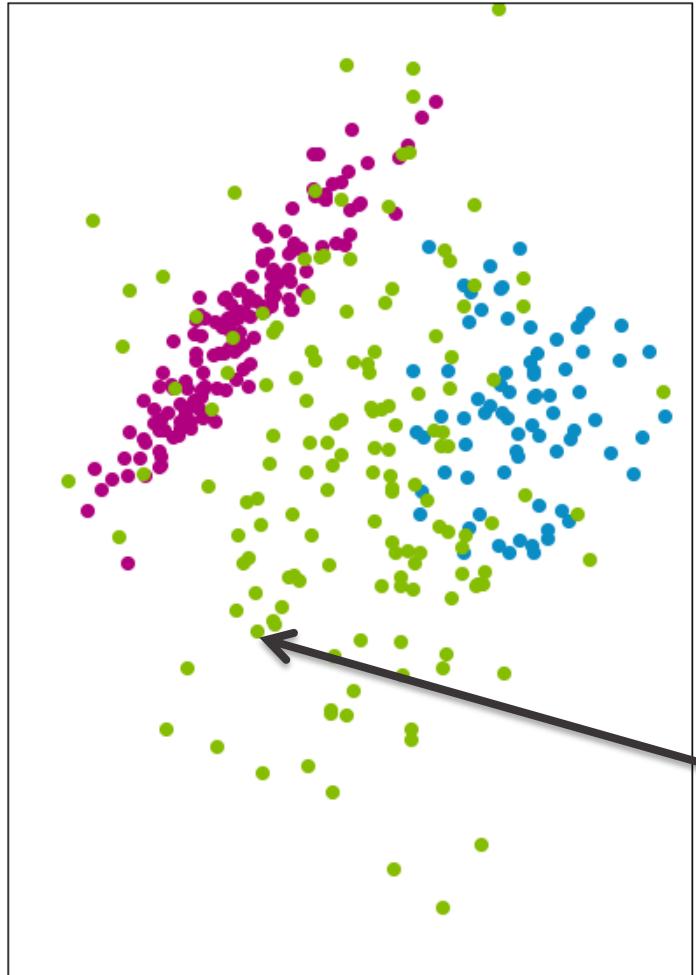


Desired soft assignments (**responsibilities**) are **easy** to compute when cluster parameters $\{\pi_k, \mu_k, \Sigma_k\}$ are known

But, we don't know these!

Part 2a:
Imagine we knew the cluster
(hard) assignments z_i

Estimating cluster parameters



Imagine we know the
cluster assignments

Estimation problem
decouples across
clusters

Is green point informative of
fuchsia cluster parameters?

NO!

Data table decoupling over clusters

R	G	B	Cluster
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	3
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	3
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	3
$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$	1
$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$	2
$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$	2

Maximum likelihood estimation

R	G	B	Cluster
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	3
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	3
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	3

Estimate $\{\pi_k, \mu_k, \Sigma_k\}$
given data assigned
to cluster k

maximum likelihood estimation
(MLE)

Find parameters that maximize the
score, or *likelihood*, of data

Mean/covariance MLE

Sum characteristics

R	G	B	Cluster
$x_1[1]$	$x_1[2]$	$x_1[3]$	3
$x_2[1]$	$x_2[2]$	$x_2[3]$	3
$x_3[1]$	$x_3[2]$	$x_3[3]$	3

divide by 3
(the total # of obs.)

"best estimate" $\hat{\mu}_k = \frac{1}{N_k} \sum_{i \text{ in } k} x_i$ ← average data points in cluster k

of obs. in cluster

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i \text{ in } k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

Scalar case: $\hat{\sigma}_k^2 = \frac{1}{N_k} \sum_{i \text{ in } k} (x_i - \hat{\mu}_k)^2$

Cluster proportion MLE

R	G	B	Cluster
$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$	1

R	G	B	Cluster
$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$	2
$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$	2

R	G	B	Cluster
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	3
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	3
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	3

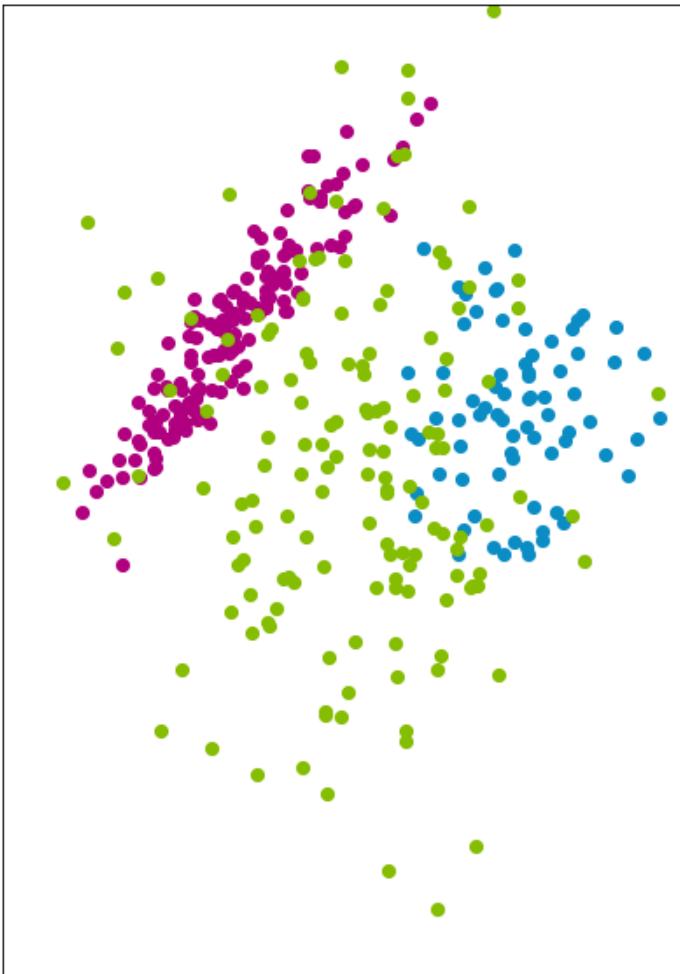
obs in cluster k

$$\hat{\pi}_k = \frac{N_k}{N}$$

total # of obs

True for general mixtures of i.i.d. data,
not just Gaussian clusters

Part 2a summary



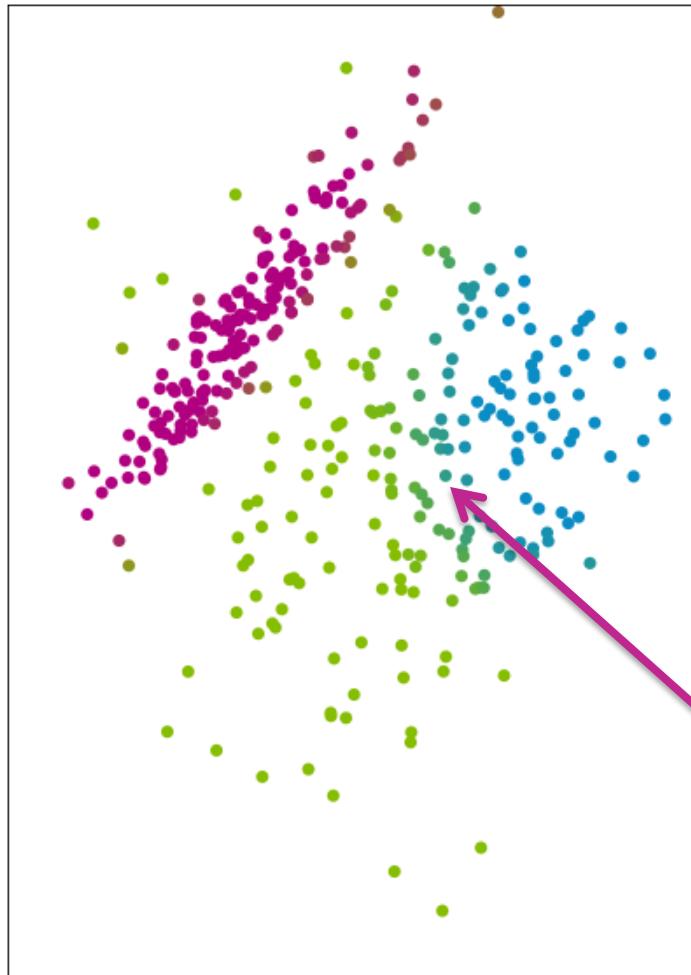
needed to compute soft assignments
Cluster parameters are simple
to compute if we know the
cluster assignments

But, we don't know these!

Part 2b:

What can we do with just soft assignments r_{ij} ?

Estimating cluster parameters from soft assignments



Instead of having a full observation \mathbf{x}_i in cluster k , just allocate a portion r_{ik}

\mathbf{x}_i divided across all clusters,
as determined by r_{ik}

Maximum likelihood estimation from soft assignments

Just like in boosting with weighted observations...

R	G	B	r_{i1}	r_{i2}	r_{i3}
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	0.30	0.18	0.52
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	0.01	0.26	0.73
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	0.002	0.008	0.99
$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$	0.75	0.10	0.15
$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$	0.05	0.93	0.02
$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$	0.13	0.86	0.01



52% chance
this obs is in
cluster 3

Total weight in cluster:
(effective # of obs)

1.242 2.8 2.42

Maximum likelihood estimation from soft assignments

R	G	B	Cluster 1 weights
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	0.30
$\mathbf{x}_2[1]$	R	G	Cluster 2 weights
$\mathbf{x}_3[1]$			
$\mathbf{x}_4[1]$	$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$
$\mathbf{x}_5[1]$	$\mathbf{x}_2[1]$	R	Cluster 3 weights
$\mathbf{x}_6[1]$	$\mathbf{x}_3[1]$	G	
$\mathbf{x}_4[1]$	$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$
$\mathbf{x}_5[1]$	$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$
$\mathbf{x}_6[1]$	$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$
	$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$
	$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$
	$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$

Cluster-specific location/shape MLE

R	G	B	Cluster 1 weights
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	0.30
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	0.01
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	0.002
$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$	0.75
$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$	0.05
$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$	0.13

Compute cluster parameter estimates
with weights on each row operation

$$\hat{\mu}_k = \frac{1}{N_k^{\text{soft}}} \sum_{i=1}^N r_{ik} x_i$$
$$\hat{\Sigma}_k = \frac{1}{N_k^{\text{soft}}} \sum_{i=1}^N r_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

$$N_k^{\text{soft}} = \sum_{i=1}^N r_{ik}$$

Total weight in cluster k
= effective # obs

MLE of cluster proportions $\hat{\pi}_k$

r_{i1}	r_{i2}	r_{i3}
0.30	0.18	0.52
0.01	0.26	0.73
0.002	0.008	0.99
0.75	0.10	0.15
0.05	0.93	0.02
0.13	0.86	0.01

Total weight
in cluster:

1.242 | 2.8 | 2.42

Total weight
in dataset:

6

datapoints N

$$\hat{\pi}_k = \frac{N_k^{\text{soft}}}{N}$$

$$N_k^{\text{soft}} = \sum_{i=1}^N r_{ik}$$

Total weight in cluster k
= effective # obs

Estimate cluster
proportions from
relative weights

Defaults to hard assignment case when r_{ij} in {0,1}

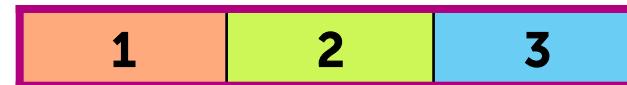
Hard assignments have:

$$r_{ik} = \begin{cases} 1 & i \text{ in } k \\ 0 & \text{otherwise} \end{cases}$$

R	G	B	r_{i1}	r_{i2}	r_{i3}
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	0	0	1
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	0	0	1
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	0	0	1
$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$	1	0	0
$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$	0	1	0
$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$	0	1	0

One-hot encoding of
cluster assignment

Total weight in cluster:



Equating the estimates...

$$\hat{\pi}_k = \frac{N_k^{\text{Soft}}}{N}$$

$$N_k^{\text{soft}} = \sum_{i=1}^N r_{ik}$$

if $r_{ik} > 0, 1$
just count
obs i in cluster
 k if $r_{ik} > 1$
 $= N_k$ ✓

$$\hat{\mu}_k = \frac{1}{N_k^{\text{soft}}} \sum_{i=1}^N r_{ik} x_i$$

only add x_i if i in k
($r_{ik} > 1$)

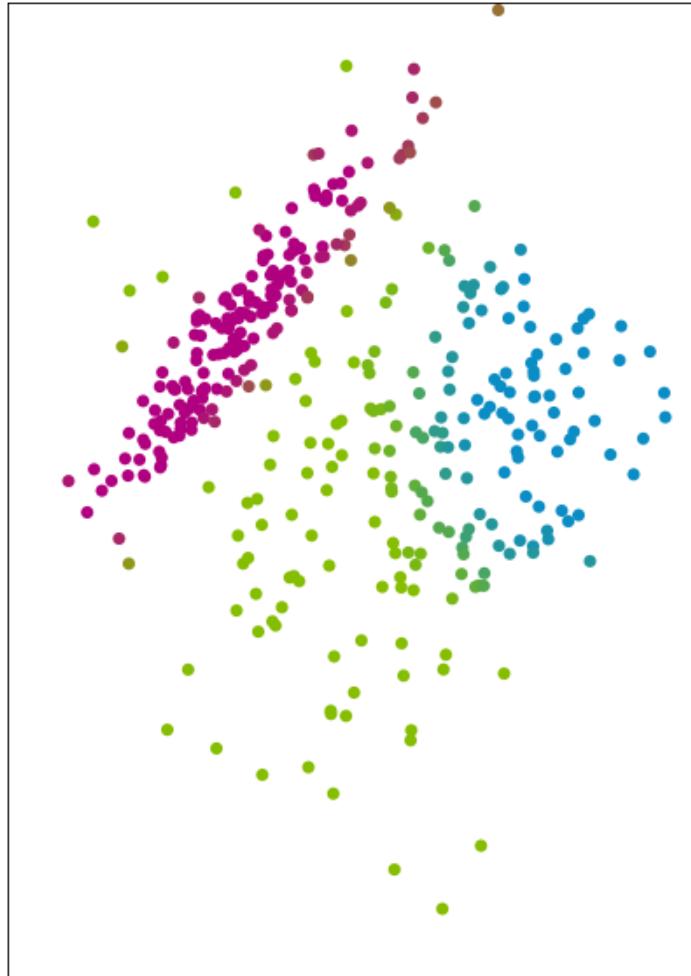
$$\rightarrow \frac{1}{N_k} \sum_{i \in k} x_i$$
 ✓

$$\hat{\Sigma}_k = \frac{1}{N_k^{\text{soft}}} \sum_{i=1}^N r_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

same as above

$$= \frac{1}{N_k} \sum_{i \in k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

Part 2b summary



Still straightforward
to compute cluster
parameter estimates
from soft assignments

Expectation maximization (EM)

Expectation maximization (EM): An iterative algorithm

Motivates an iterative algorithm:

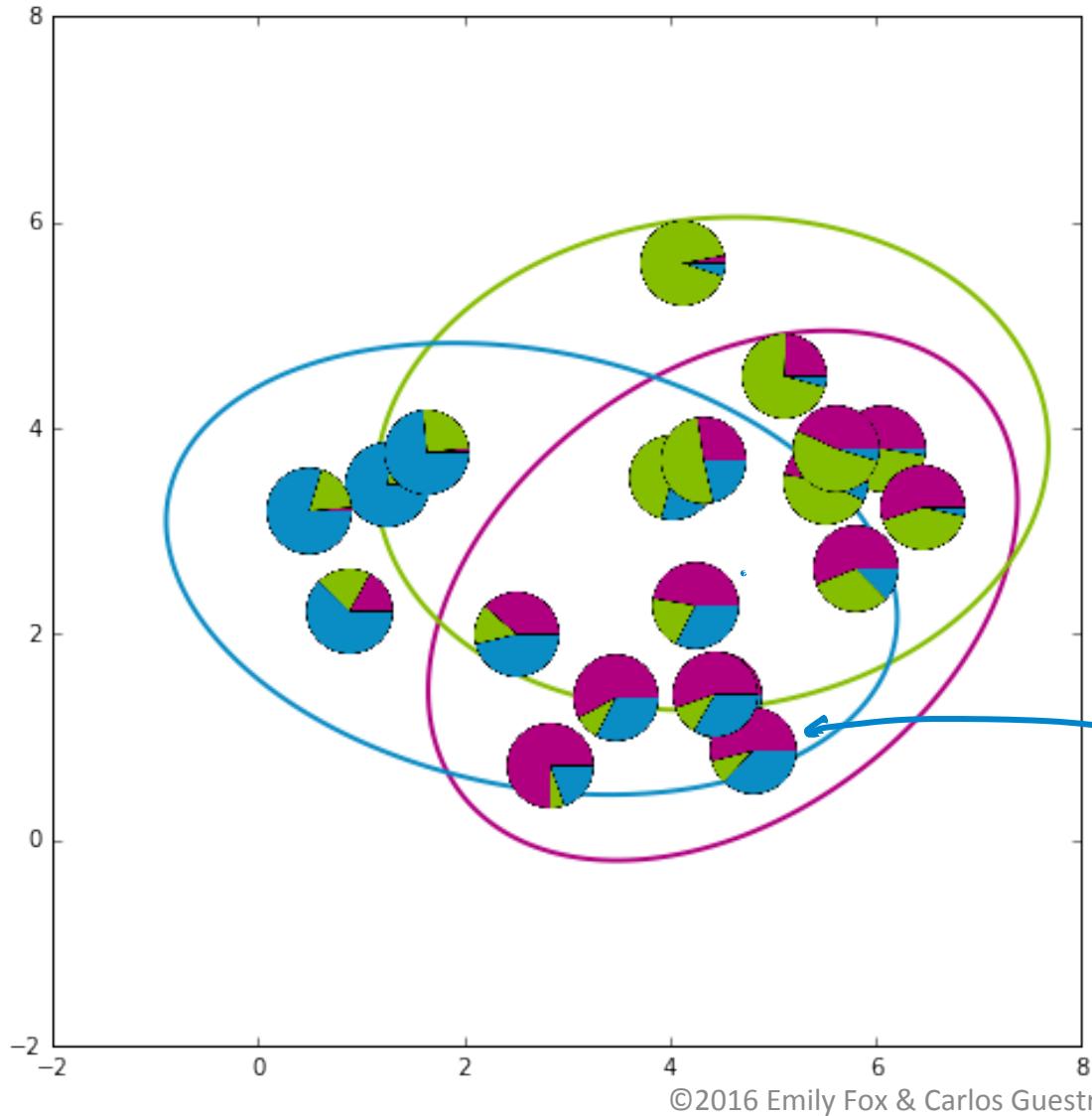
1. **E-step:** estimate cluster responsibilities
given current parameter estimates

$$\hat{r}_{ik} = \frac{\hat{\pi}_k N(x_i | \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^K \hat{\pi}_j N(x_i | \hat{\mu}_j, \hat{\Sigma}_j)}$$

2. **M-step:** maximize likelihood over
parameters given current responsibilities

$$\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k | \{\hat{r}_{ik}, x_i\}$$

EM for mixtures of Gaussians in pictures – initialization

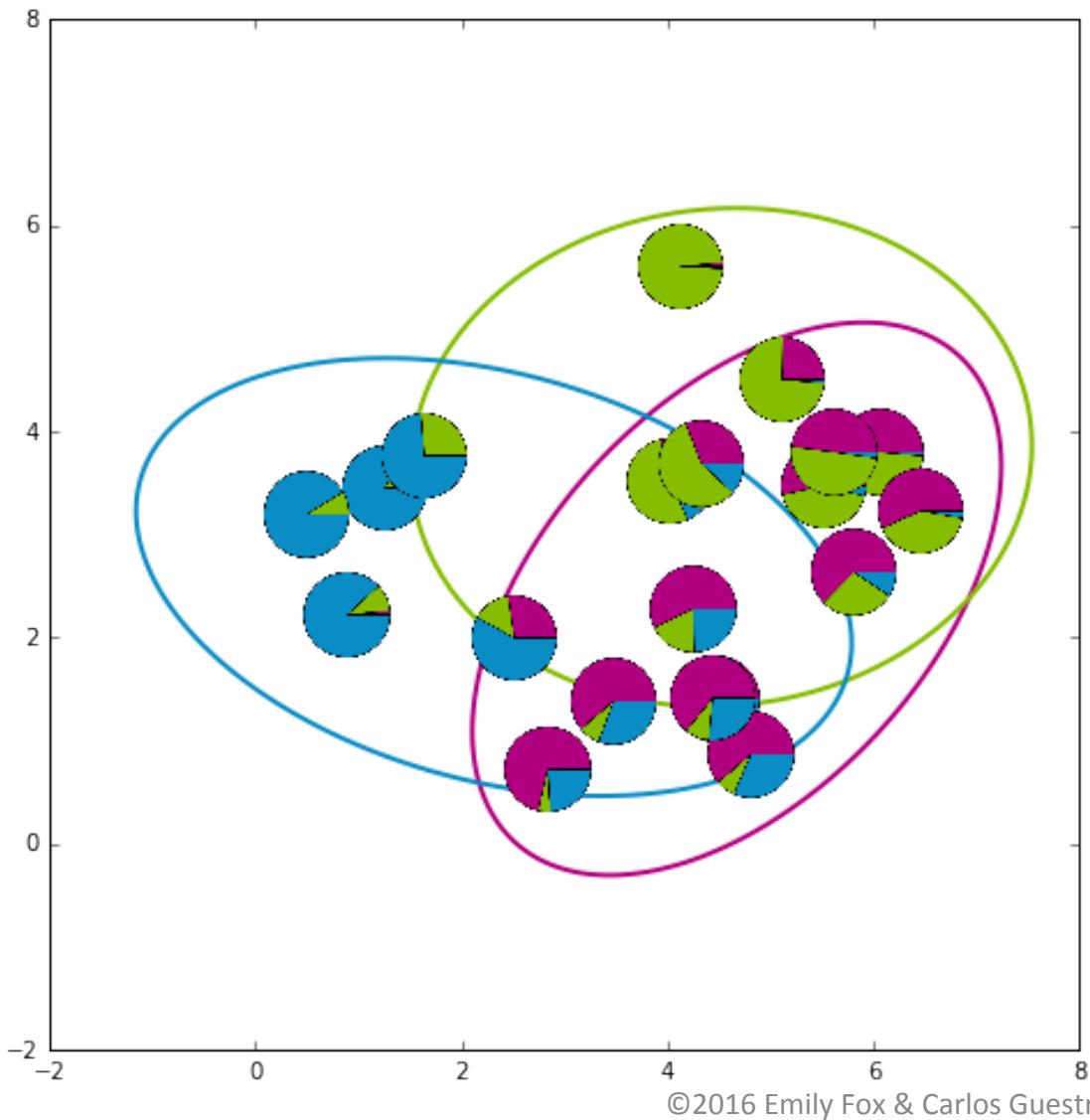


Initialize
iter counter
 $\{\pi_k^{(0)}, \mu_k^{(0)}, \hat{\Sigma}_k^{(0)}\}$

Then compute
 $\hat{r}_{ik}^{(1)}$

$$\hat{r}_i^{(1)} = [0.52 \quad 0.4 \quad 0.08] \quad \text{fuchsia} \quad \text{blue} \quad \text{green}$$

EM for mixtures of Gaussians in pictures – after 1st iteration



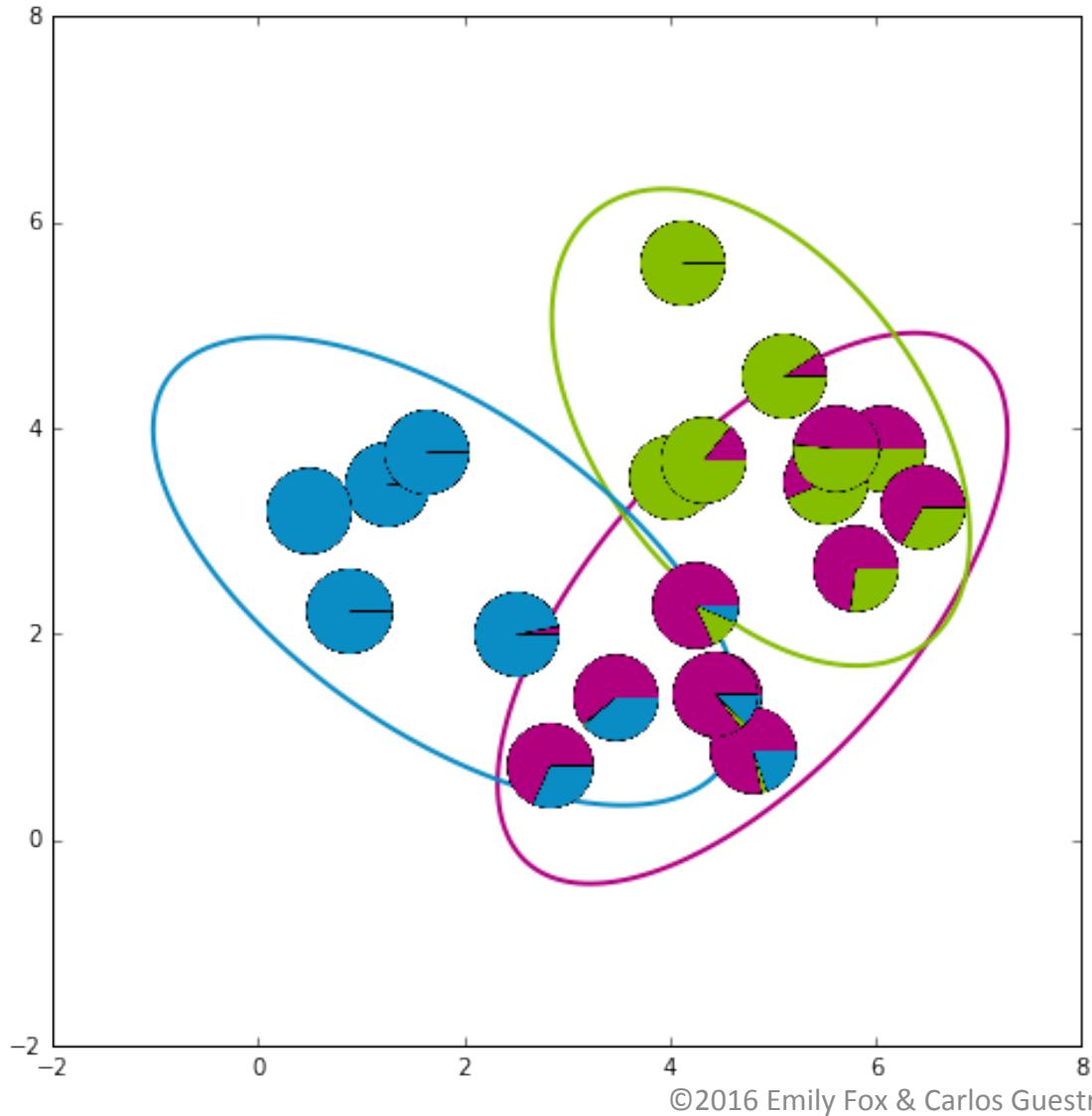
Maximize likelihood
given soft assign. $r_{ik}^{(1)}$

$$\rightarrow \{\hat{\pi}_k^{(1)}, \hat{\mu}_k^{(1)}, \hat{\Sigma}_k^{(1)}\}$$

Then recompute responsibilities

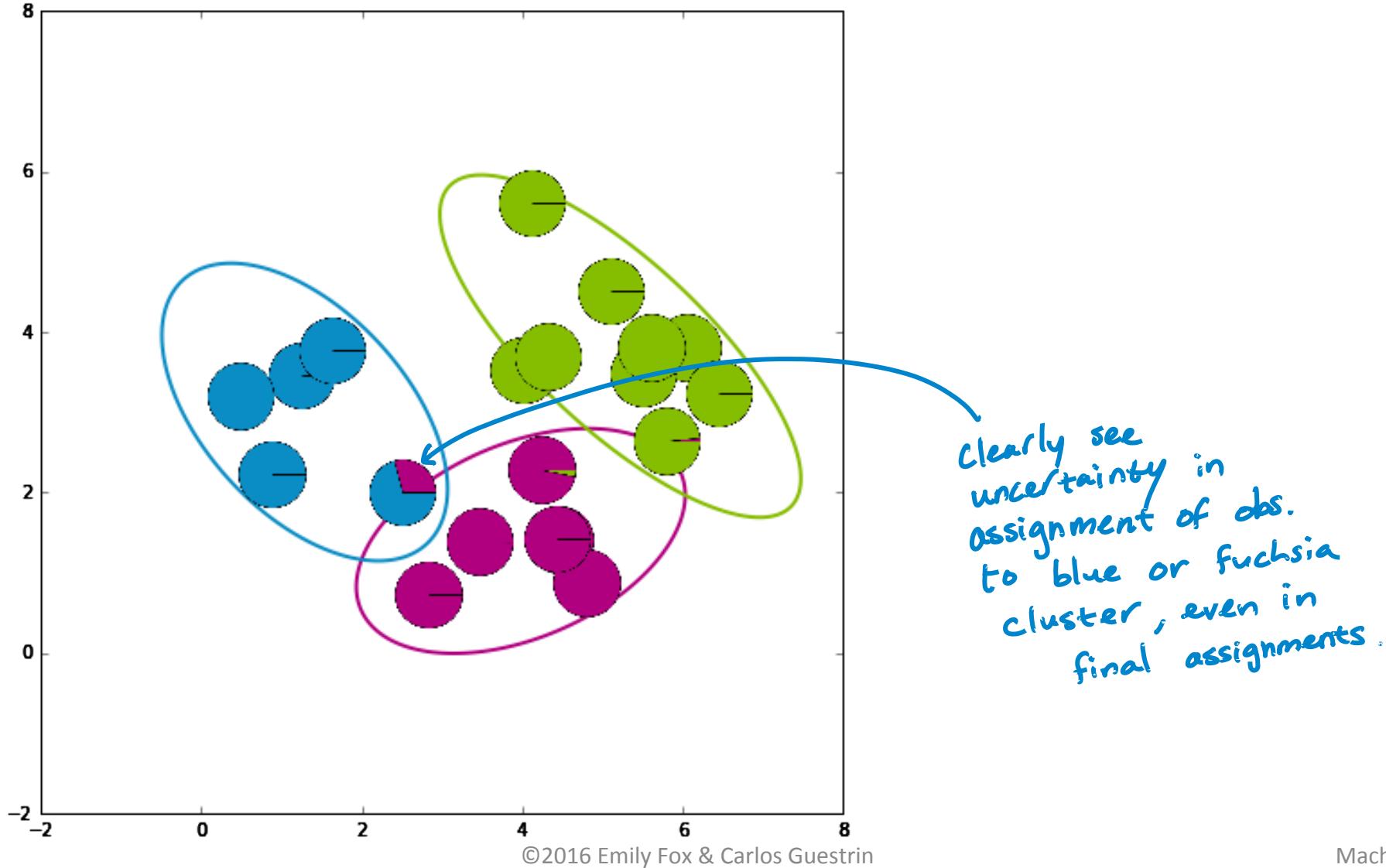
$$\hat{r}_{ik}^{(2)}$$

EM for mixtures of Gaussians in pictures – after 2nd iteration

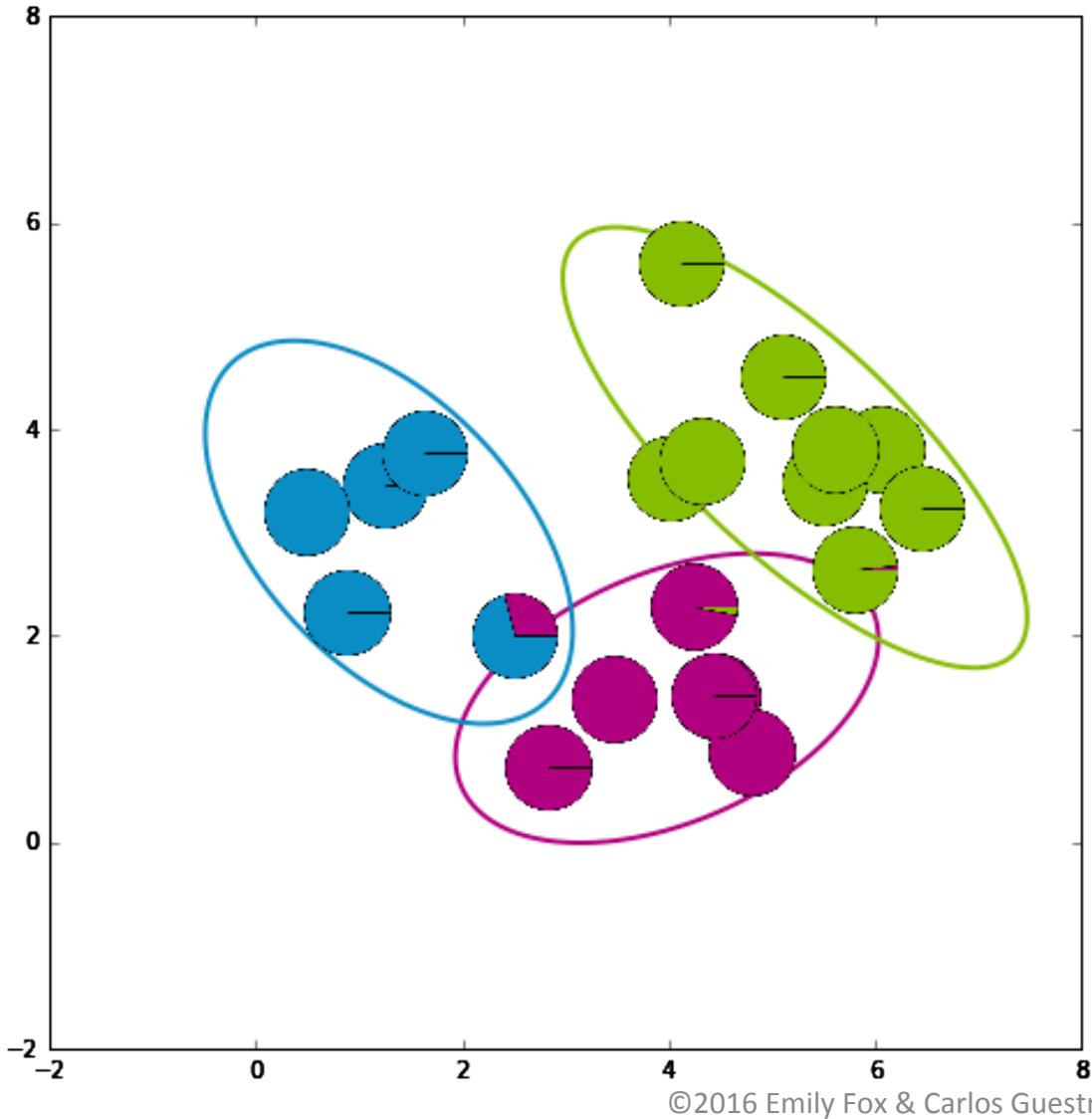


rinse
+
repeat
until convergence

EM for mixtures of Gaussians in pictures – converged solution



EM for mixtures of Gaussians in pictures - replay



The nitty gritty of EM

Convergence of EM

- EM is a coordinate-ascent algorithm
 - Can equate E-and M-steps with alternating maximizations of an objective function
- Converges to a local mode
- We will assess via (log) likelihood of data under current parameter and responsibility estimates

Initialization

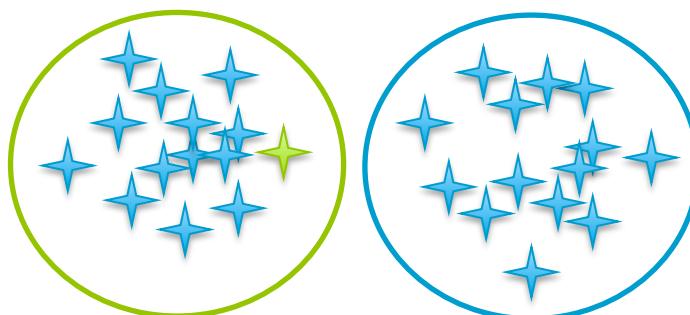
- Many ways to initialize the EM algorithm
- Important for convergence rates and quality of local mode found
- Examples:
 - Choose K observations at random to define K “centroids”. Assign other observations to nearest centroid to form initial parameter estimates.
 - Pick centers sequentially to provide good coverage of data like in k-means++
 - Initialize from k-means solution
 - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed

Overfitting of MLE

Maximizing likelihood can **overfit to data**

Imagine at K=2 example with one obs assigned to cluster 1 and others assigned to cluster 2

- What parameter values maximize likelihood?



Set center equal to point and shrink variance to 0

Likelihood goes to ∞ !

Overfitting in high dims

Doc-clustering example:

Imagine only 1 doc assigned to cluster k has word w
(or all docs in cluster agree on count of word w)

Likelihood maximized by setting $\mu_k[w] = \mathbf{x}_i[w]$ and $\sigma_{w,k}^2 = 0$

Likelihood of any doc with different count on word w being in cluster k is 0!

Simple regularization of M-step for mixtures of Gaussians

Simple fix: Don't let variances $\rightarrow 0!$

Add small amount to diagonal of covariance estimate

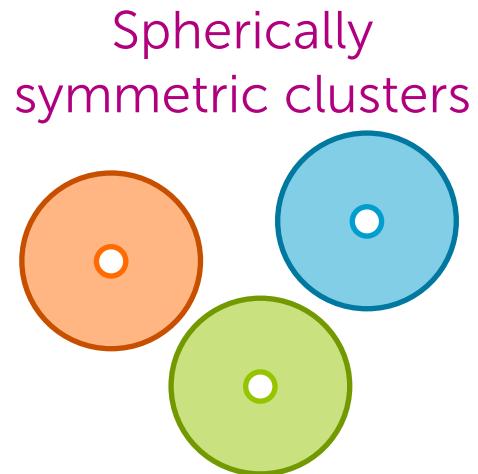
Alternatively, take Bayesian approach and place prior on parameters.

Similar idea, but all parameter estimates are “smoothed” via cluster pseudo-observations.

Relationship to k-means

Consider Gaussian mixture model with

$$\Sigma = \begin{pmatrix} \sigma^2 & & & \\ & \sigma^2 & & \\ & & \sigma^2 & \\ & \ddots & \ddots & \ddots & \sigma^2 \end{pmatrix}$$



and let the variance parameter $\sigma \rightarrow 0$

Datapoint gets fully assigned to nearest center, just as in k-means

- Spherical clusters with equal variances, so **relative likelihoods** just function of distance to cluster center
- As variances $\rightarrow 0$, **likelihood ratio becomes 0 or 1**
- Responsibilities weigh in cluster proportions, but dominated by likelihood disparity

$$\hat{r}_{ik} = \frac{\hat{\pi}_k N(x_i | \hat{\mu}_k, \sigma^2 I)}{\sum_{j=1}^K \hat{\pi}_j N(x_i | \hat{\mu}_j, \sigma^2 I)}$$

Infinitesimally small variance EM = k-means

1. **E-step:** estimate cluster responsibilities given current parameter estimates

$$\hat{r}_{ik} = \frac{\hat{\pi}_k N(x_i | \hat{\mu}_k, \sigma^2 I)}{\sum_{j=1}^K \hat{\pi}_j N(x_i | \hat{\mu}_j, \sigma^2 I)} \in \{0, 1\}$$

Infinitesimally small 

Decision based on distance to nearest cluster center 

2. **M-step:** maximize likelihood over parameters given current responsibilities (**hard assignments!**)

$$\hat{\pi}_k, \hat{\mu}_k \mid \{\hat{r}_{ik}, x_i\}$$

FUTURE TOPICS

- 1. Clustering Topics**
 - Hierarchical Agglomerative Clustering
 - HDBSCAN*
 - Cross-validation
 - Autoencoders
- 2. Density Estimation Topics**
 - Histograms
 - Kernel Density Estimation
- 3. Dimensionality Reduction**
 - Principal Component Analysis (PCA)
 - t-SNE
 - UMAP
 - Autoencoders
- 4. Outlier Detection Topics**
 - One-Class Classifier Learning
 - Autoencoders
- 5. Deep Neural Network Approaches**

References:

The Hundred-Page Machine Learning Book. Andriy Burkov.

Applied Machine Learning in Python. Coursera. University of Michigan, Prof. Kevin Collins Thompson

Cluster analysis,
https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1002271612 (last visited Jan. 27, 2021).

Dimensionality reduction,
https://en.wikipedia.org/w/index.php?title=Dimensionality_reduction&oldid=1002754996 (last visited Jan. 27, 2021).

GAUSSIAN MIXTURE MODELS

Scott O'Hara

Metrowest Developers Machine Learning Group

03/17/2021

NEW TALK

Reference: [Machine Learning: Clustering & Retrieval](#)
University of Washington, Profs. Emily Fox & Carlos Guestrin

Reference: [The Hundred-Page Machine Learning Book](#). Andriy Burkov.

Reference: [Applied Machine Learning in Python](#). Coursera.
University of Michigan, Prof. Kevin Collins Thompson