Mixture Models:
Model-Based Clustering

Emily Fox & Carlos Guestrin

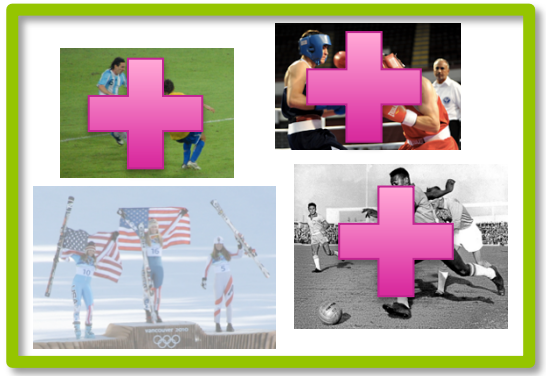
Machine Learning Specialization

University of Washington

Why a probabilistic approach?

Learn user preferences

Set of clustered documents read by user



Cluster 1

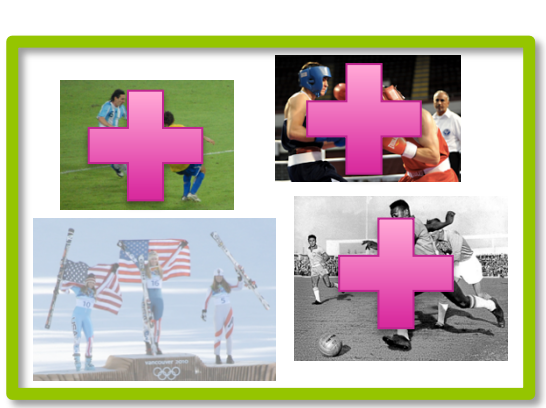
Cluster 2

Cluster 3

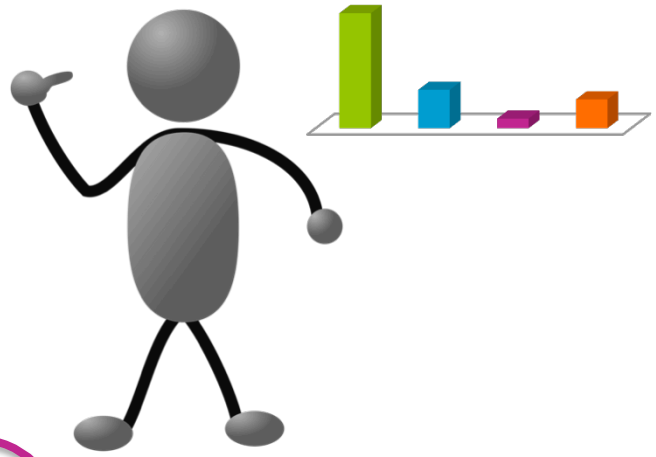
Cluster 4

Use feedback to learn user preferences over topics

Uncertainty in cluster assignments



Cluster 1

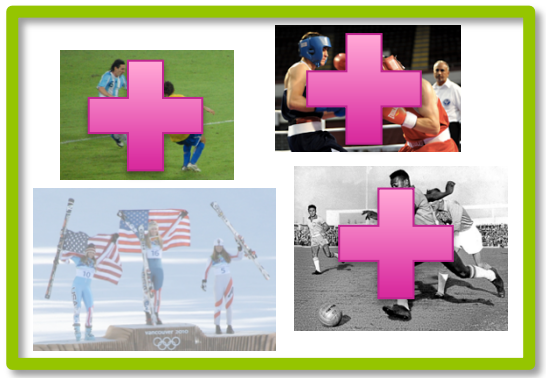


Cluster 3

Cluster 4

Slightly closer to Cluster 4 than Cluster 2, but count fully for Cluster 4?

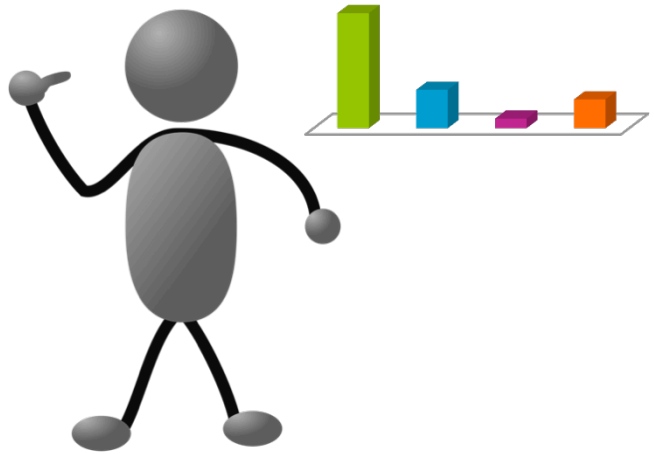
Uncertainty in cluster assignments



Cluster 1

Cluster 3

Cluster 4



Hard assignments don't tell full story

Other limitations of k-means

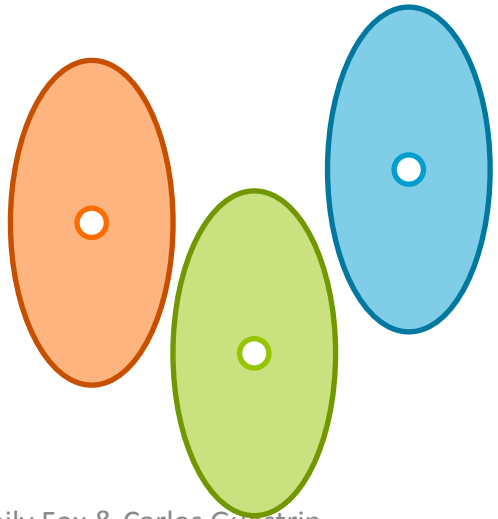
Assign observations to closest cluster center

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

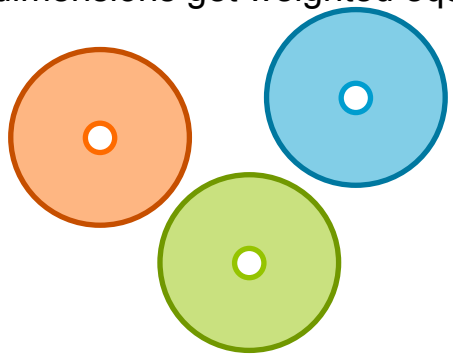
Only center matters

Can use weighted Euclidean, but requires *known* weights

Still assumes all clusters have the same axis-aligned ellipses

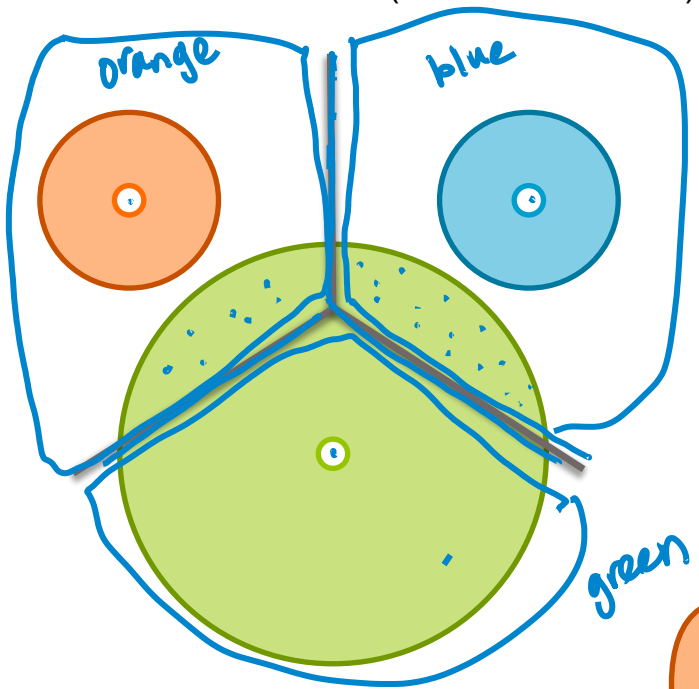


Equivalent to assuming spherically symmetric clusters
(all dimensions get weighted equally)



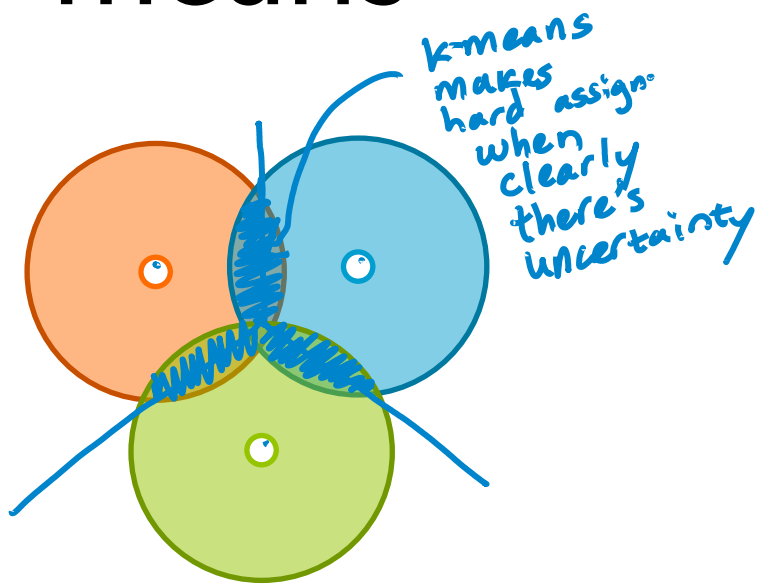
Failure modes of k-means

Voronoi tessellation (the 3 lines in blue)

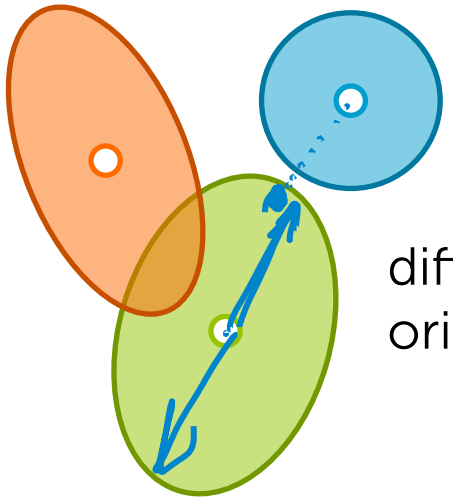


disparate cluster sizes

Gree cluster is big, not in terms of the number of data points, but in terms of the spread of points within cluster.



overlapping clusters



different shaped/ oriented clusters

Motivates probabilistic model: **mixture model**

- 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

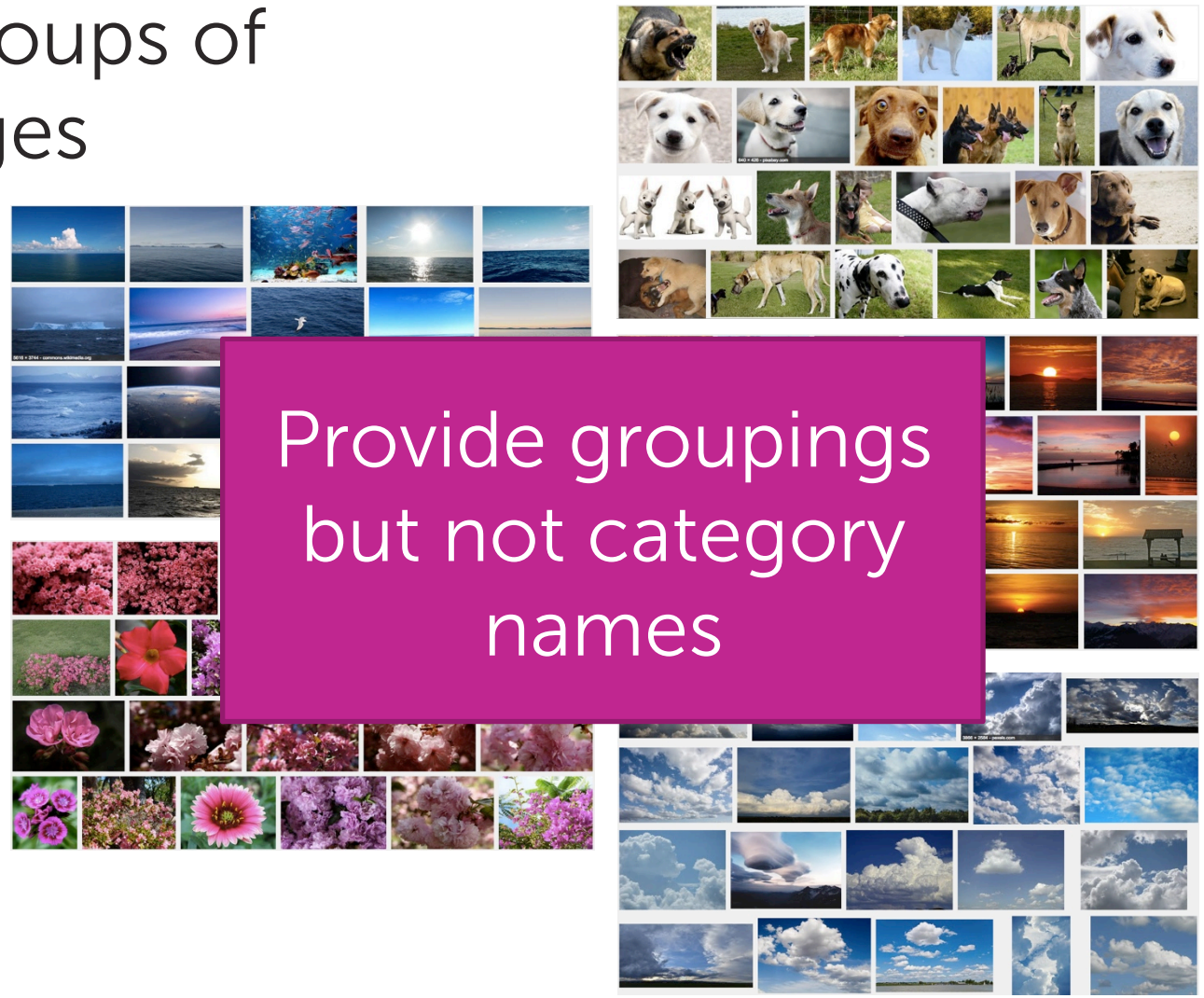
- Enables learning weightings of dimensions
  - e.g., how much to weight each word in the vocabulary when computing cluster assignment

Mixture models

Motivating application: Clustering images

Discover groups of similar images

- Ocean
- Pink flower
- Dog
- Sunset
- Clouds
- …



Provide groupings but not category names

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]

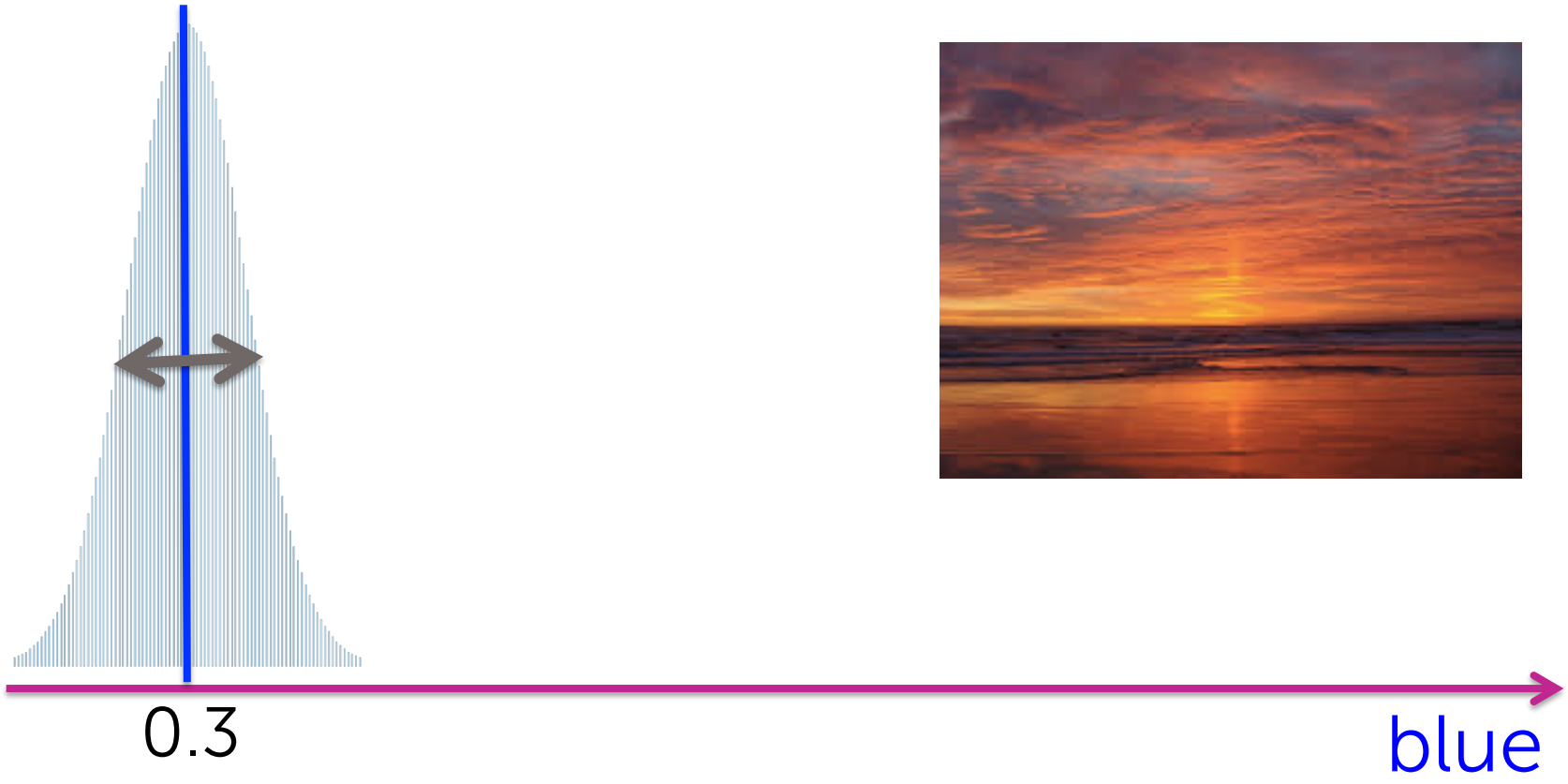
[R = 0.02, G = 0.95, B = 0.4]

Distribution over all cloud images

Let's look at just the blue dimension

Distribution over all sunset images

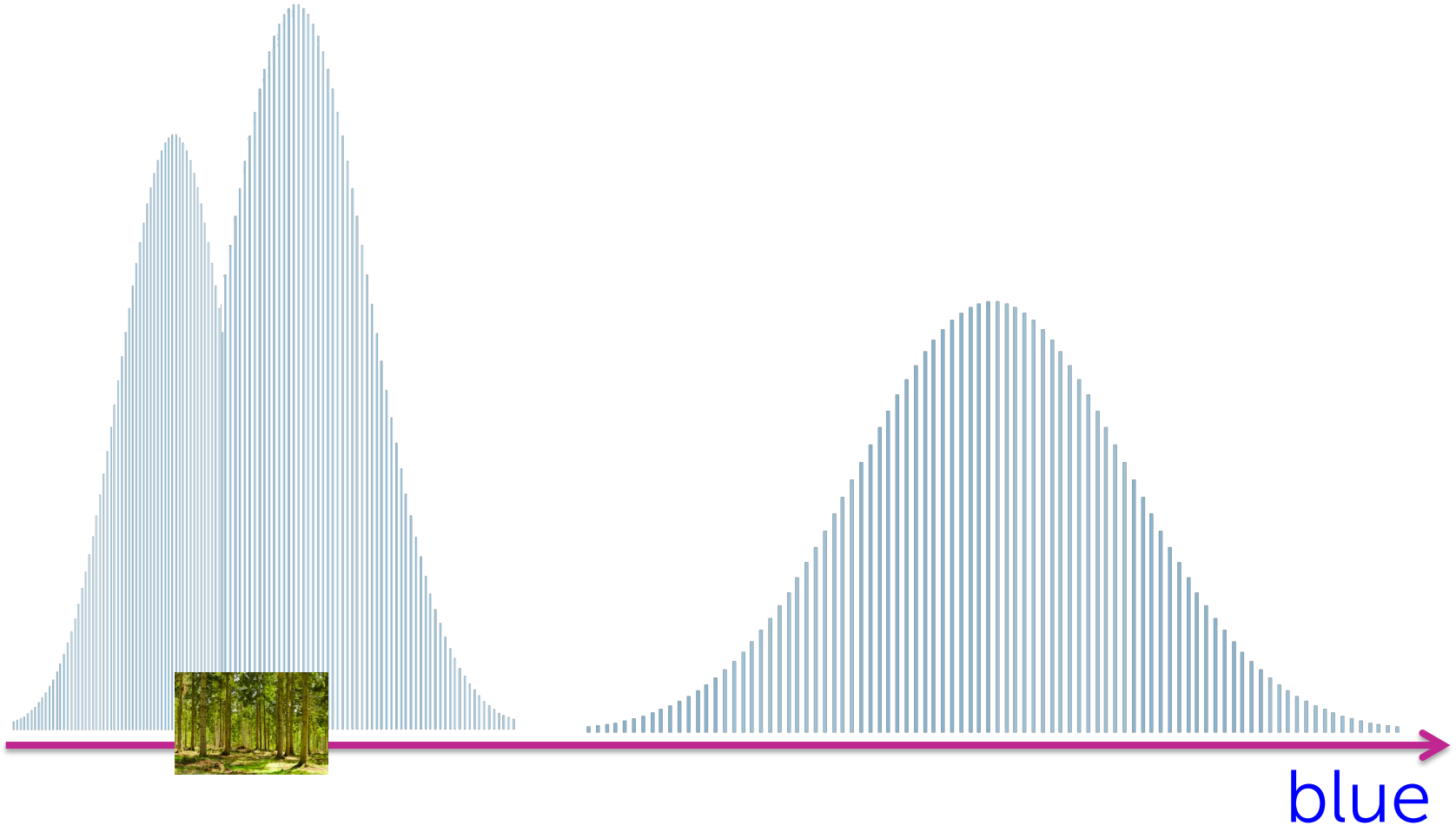
Let's look at just the blue dimension



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

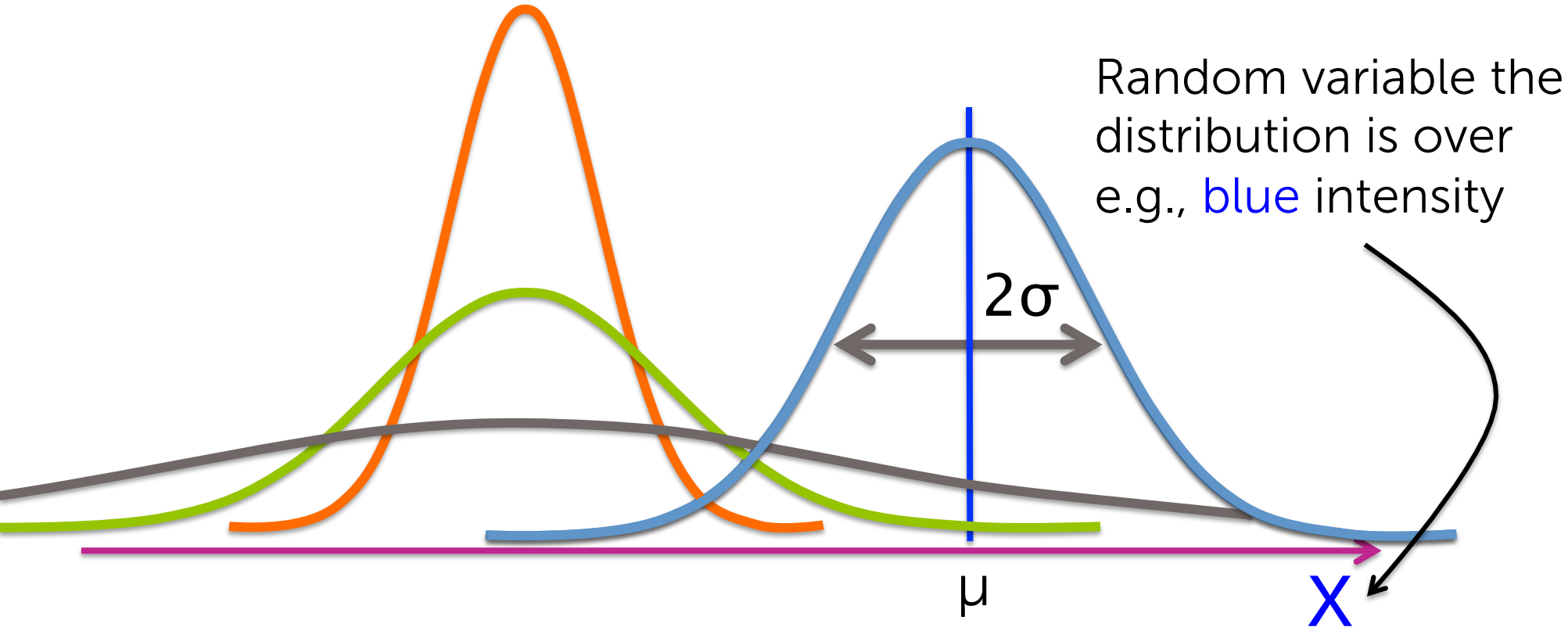
Background: Gaussian distributions

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)



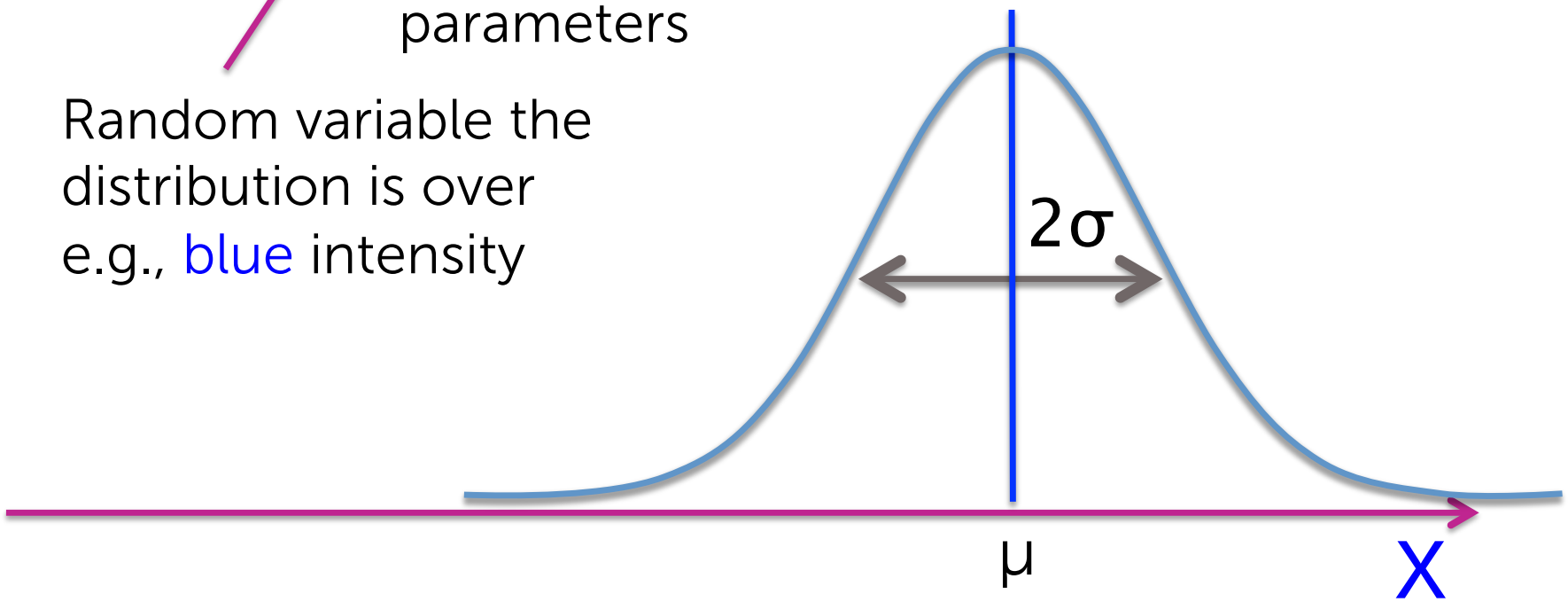
Notating a 1D Gaussian distribution

Capital N: Sometimes Gaussian distribution is referred to as the normal distribution.

$$N(x \mid \mu, \sigma^2)$$

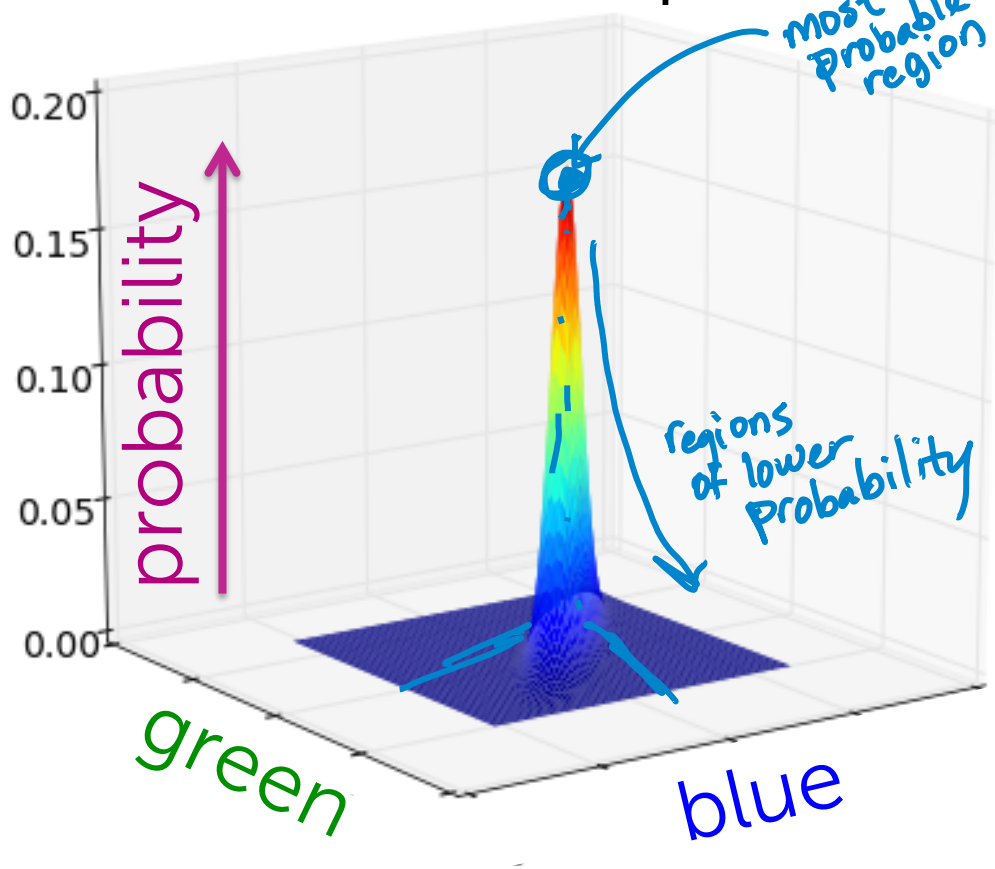
parameters

Random variable the distribution is over
e.g., blue intensity

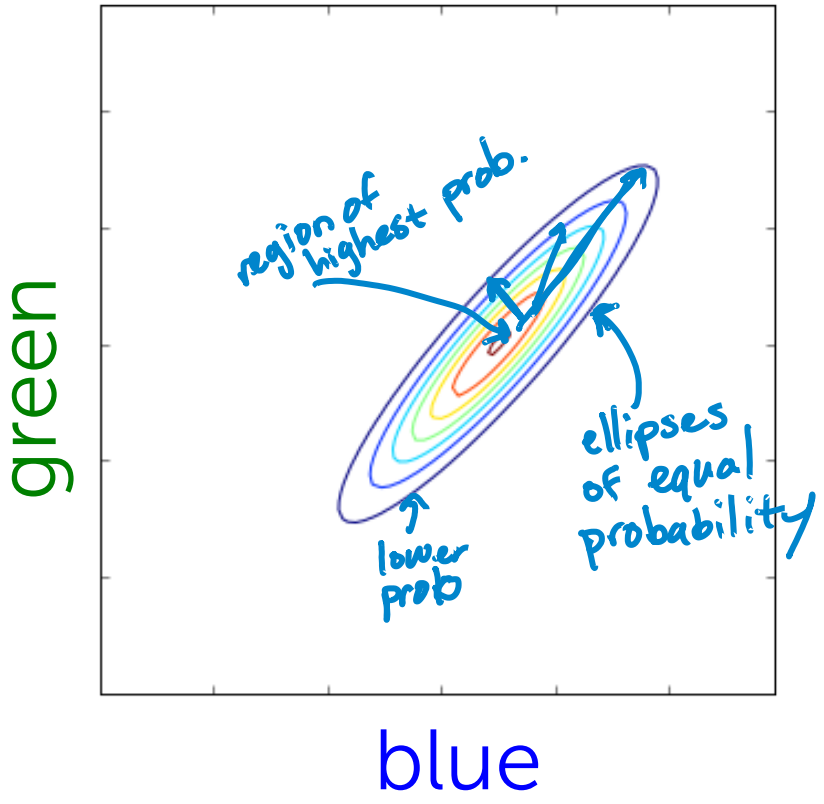


2D Gaussians – Bird's eye view

3D mesh plot



Contour plot

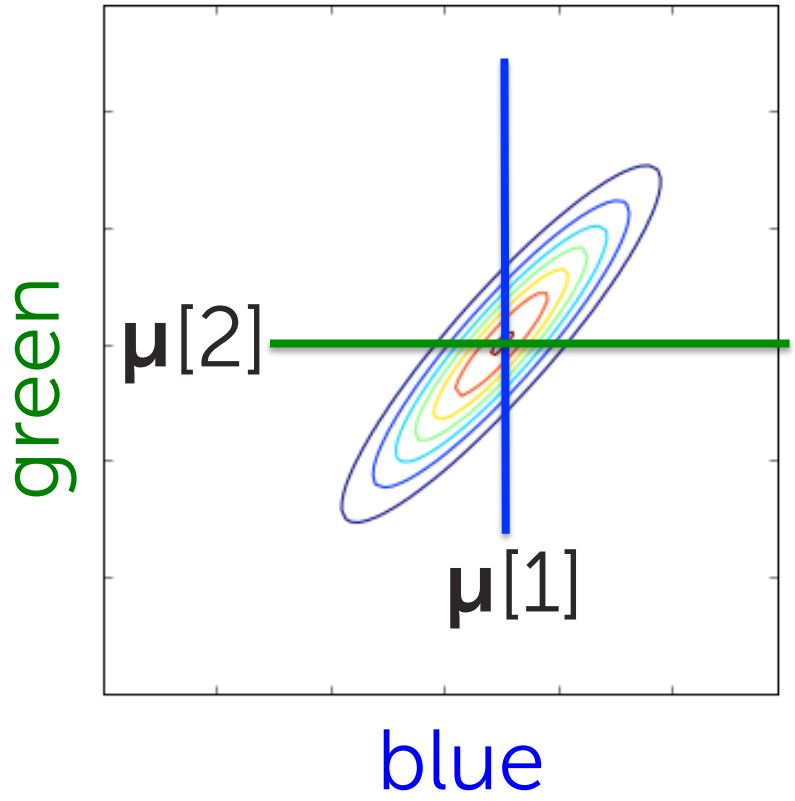


2D Gaussians – Parameters

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

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

mean centers the distribution in 2D



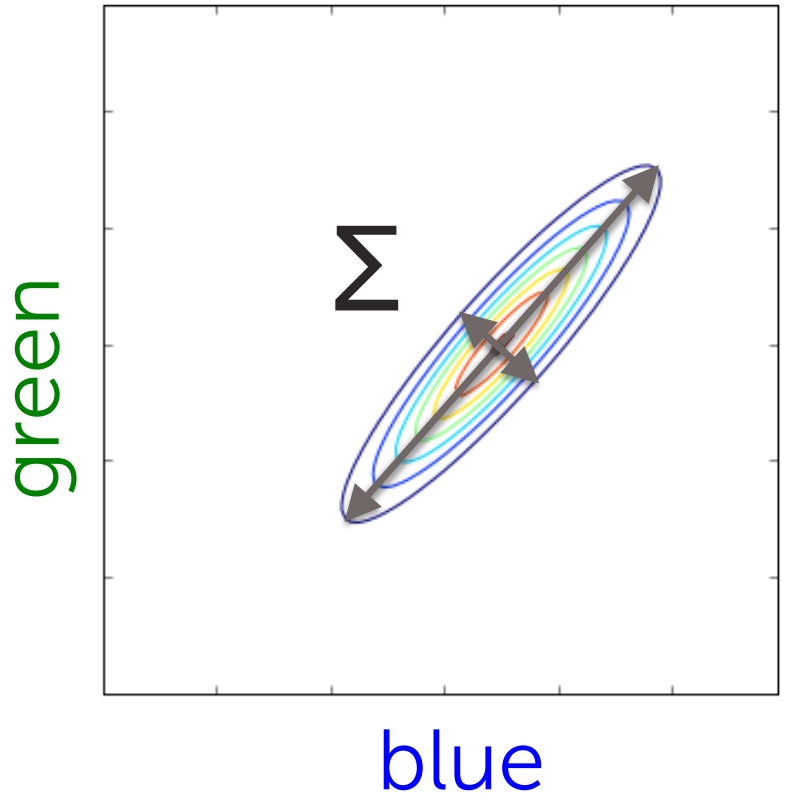
2D Gaussians – Parameters

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

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

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

covariance determines orientation + spread

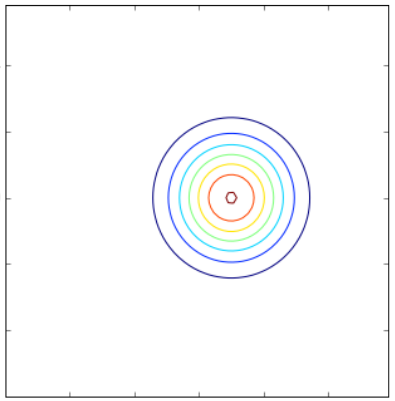


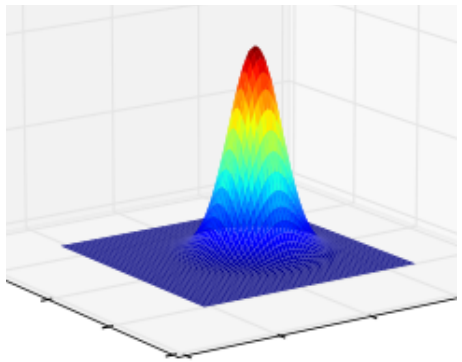
In the covariance matrix upper sigma, sigma blue squared is the variance along just the blue intensity direction or dimension of the observation vector. Similarly, sigma green squared. Sigma blue,green and sigma green.blue actually have the same value due to symmetric matrix. These 2 off-diagonal terms specify the correlation structure of this distribution (orientation of ellipses).

Covariance structures

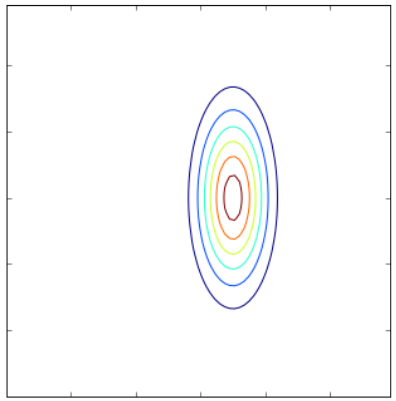
No correlation between variables

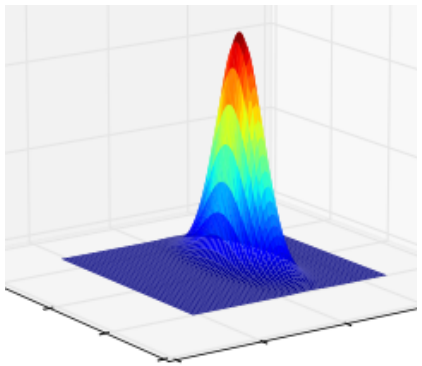
$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$





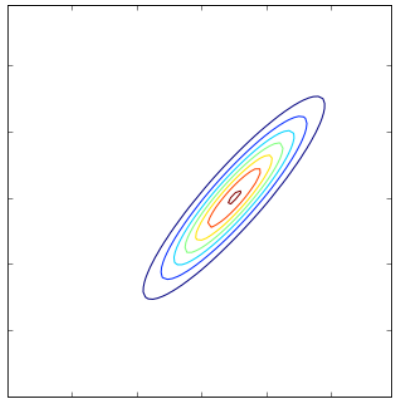
$$\Sigma = \begin{pmatrix} \sigma_B{}^2 & 0 \\ 0 & \sigma_G{}^2 \end{pmatrix}$$

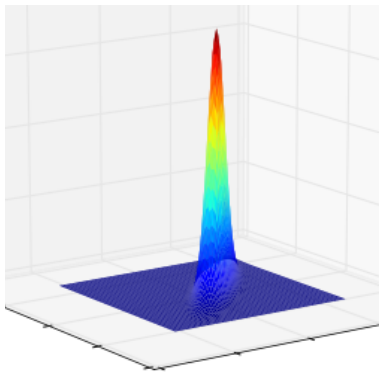




positively correlated random variables (if blue intensity is high, very likely green intensity is also hgih).

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





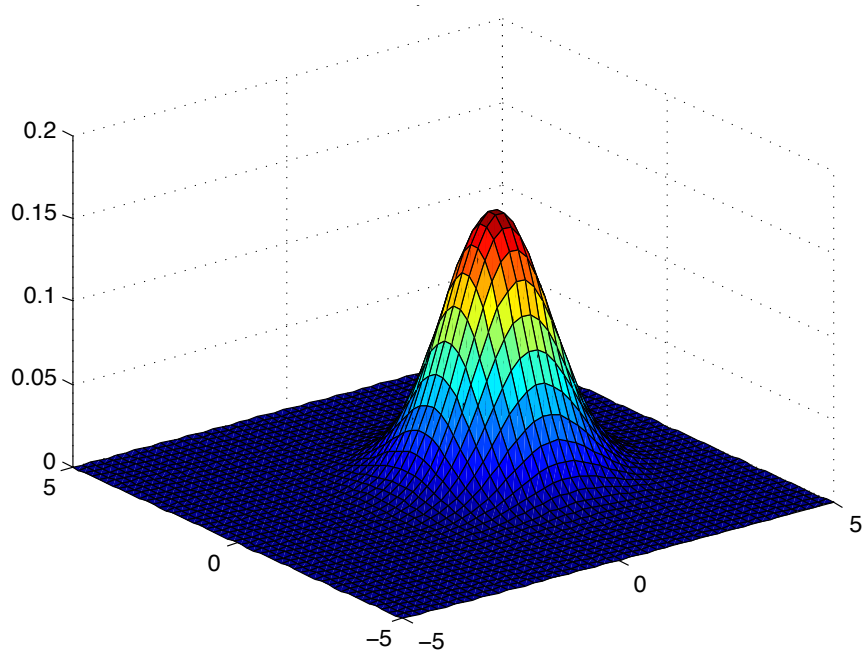
Notating a multivariate Gaussian

If x has d different dimensions, the covariance matrix will be a d*d matrix. There are going to be d variance parameters along the diagonal. Those off-diagonals are going to capture this correlation structure amongst these d different random variables. The matrix is also called positive semidefinite. There are some constraints on what the element of this matrix can be.

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

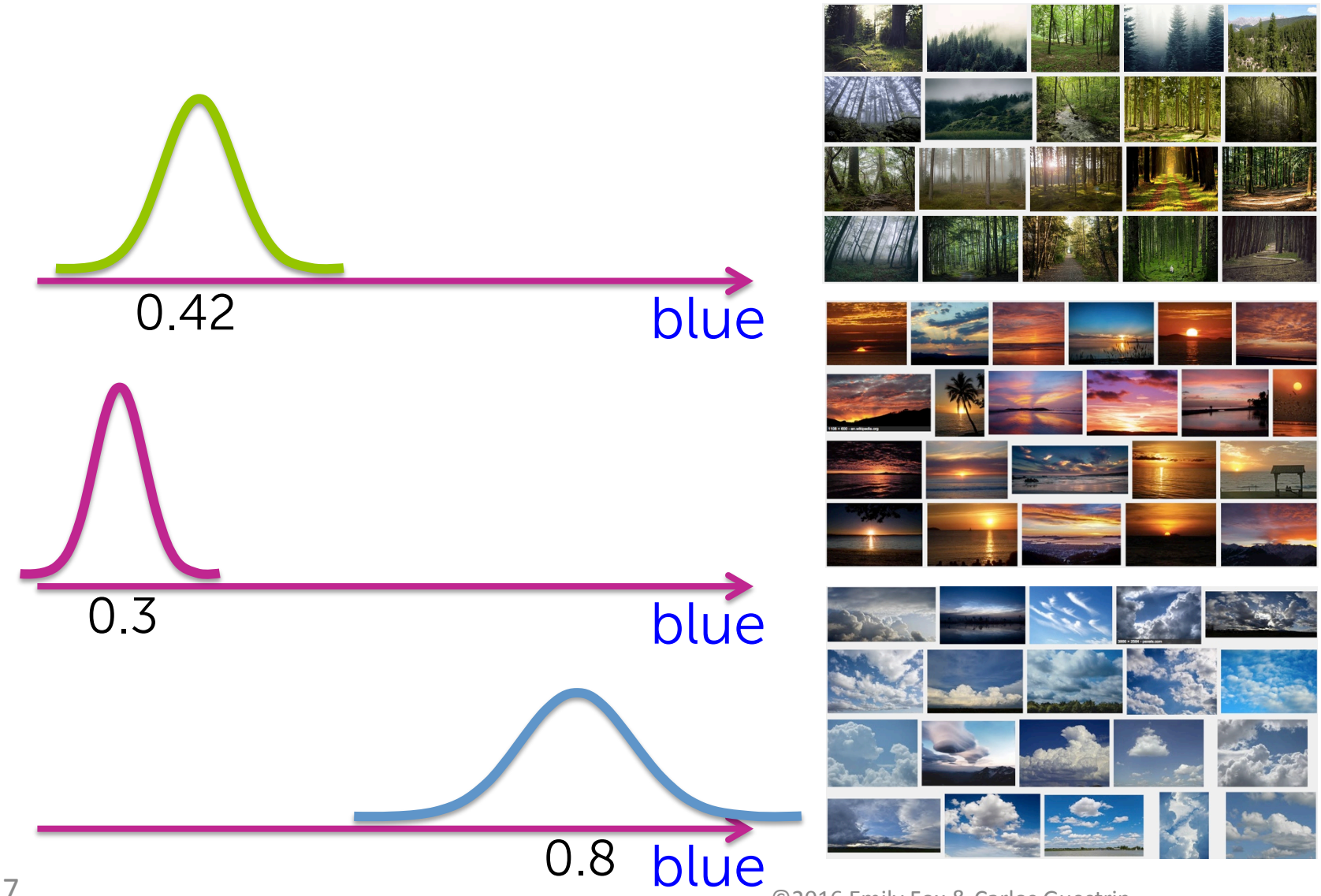
parameters

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

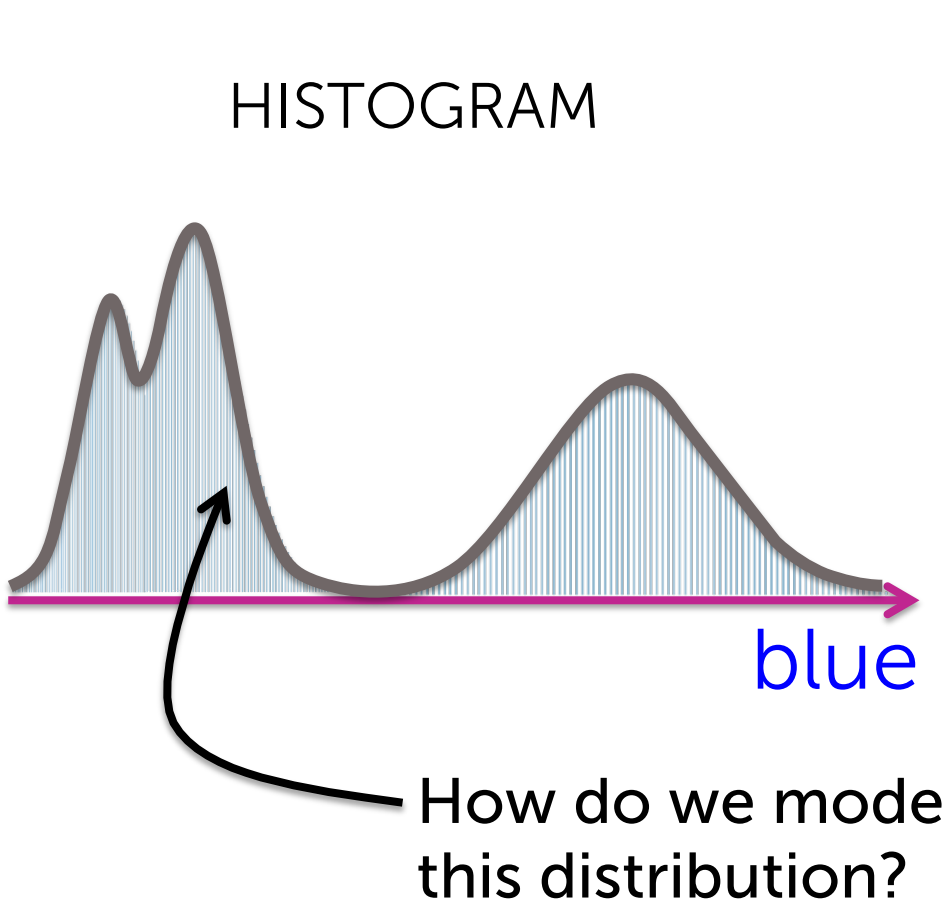


Mixture of Gaussians

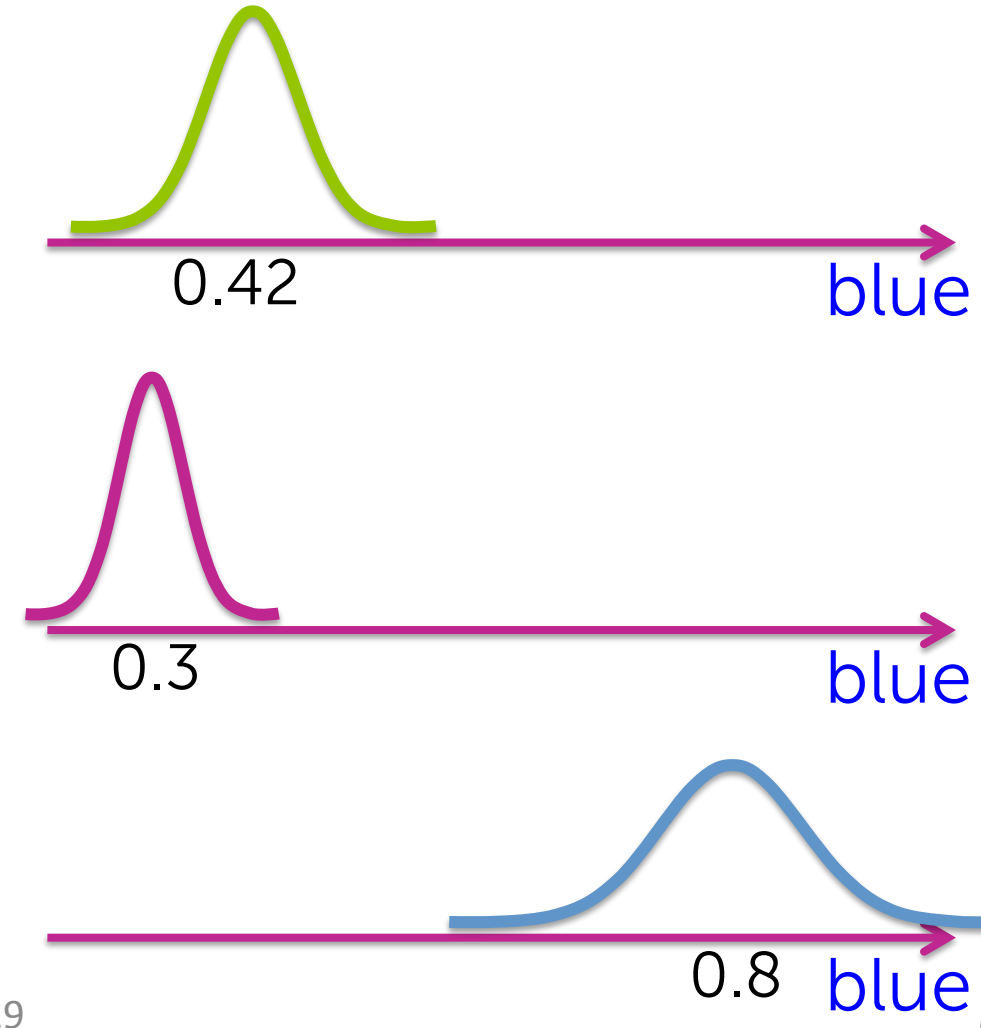
Model as Gaussian per category/cluster



Jumble of unlabeled images



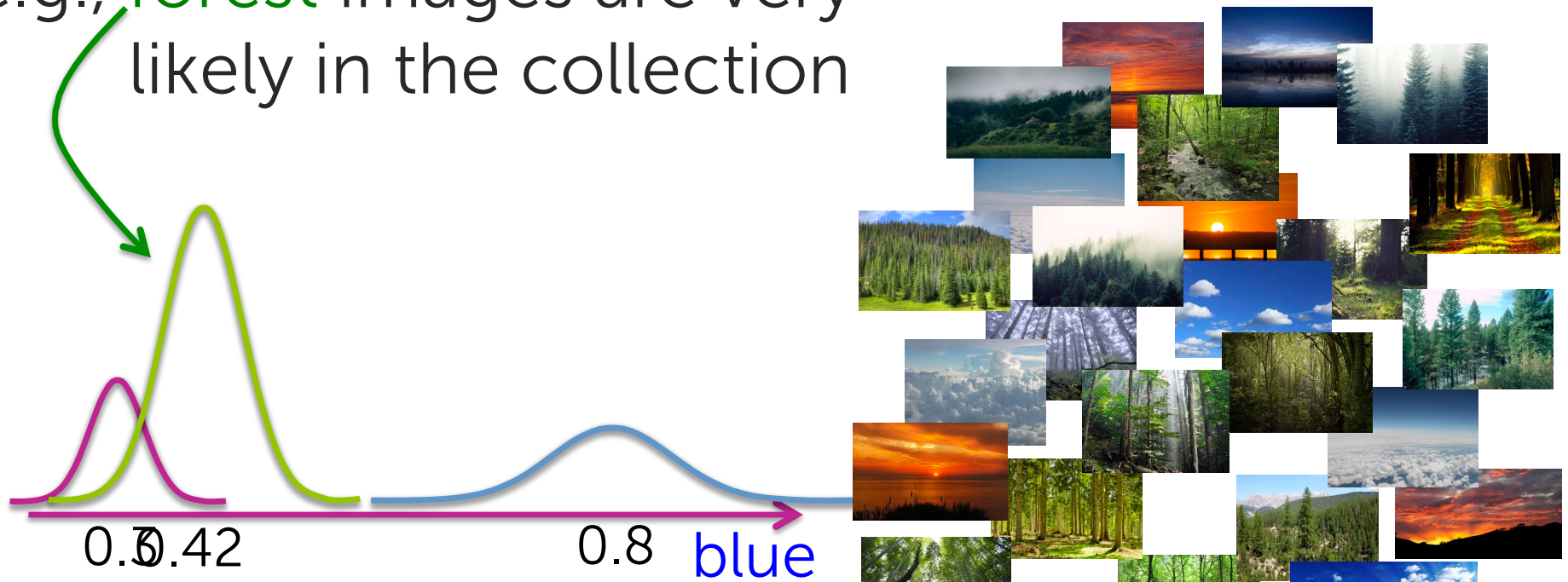
Model of jumble of unlabeled images



What if image types not equally represented?

The simple average assumes that each one of these categories appears in equal proportion in the data set, i.e., equal numbers of sunset, cloud and forest images. But what if there are many more forest images than others ?

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

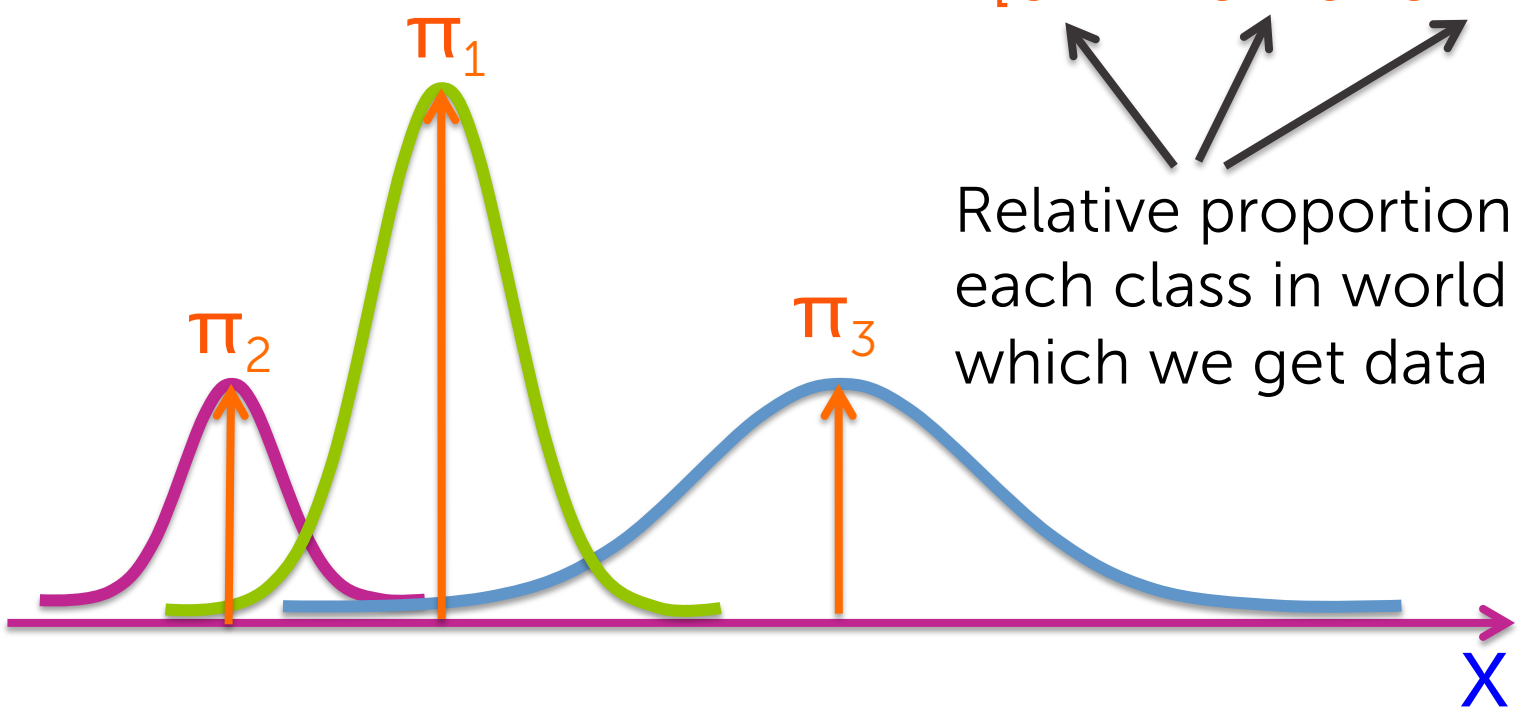


Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

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

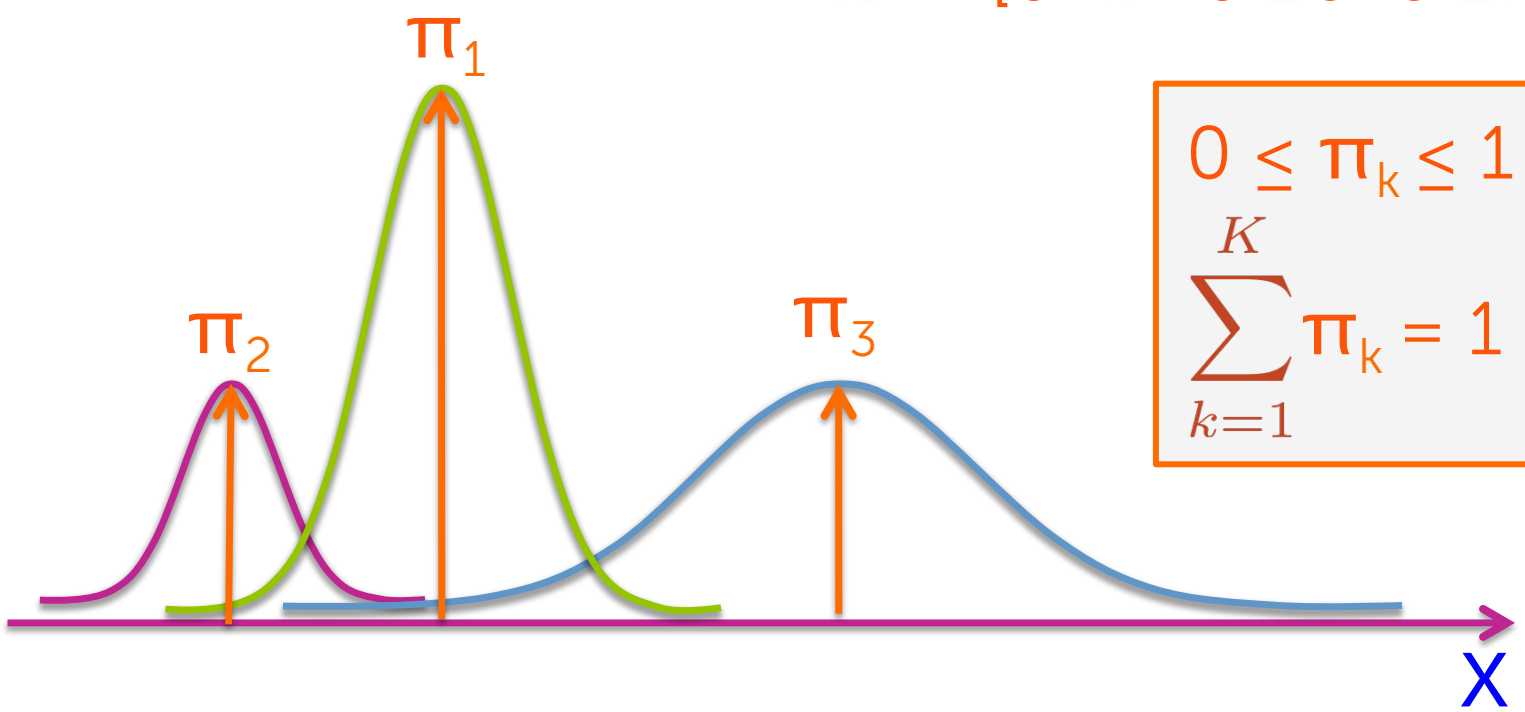
Relative proportion of each class in world from which we get data



Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

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



$$0 \leq \pi_k \leq 1$$

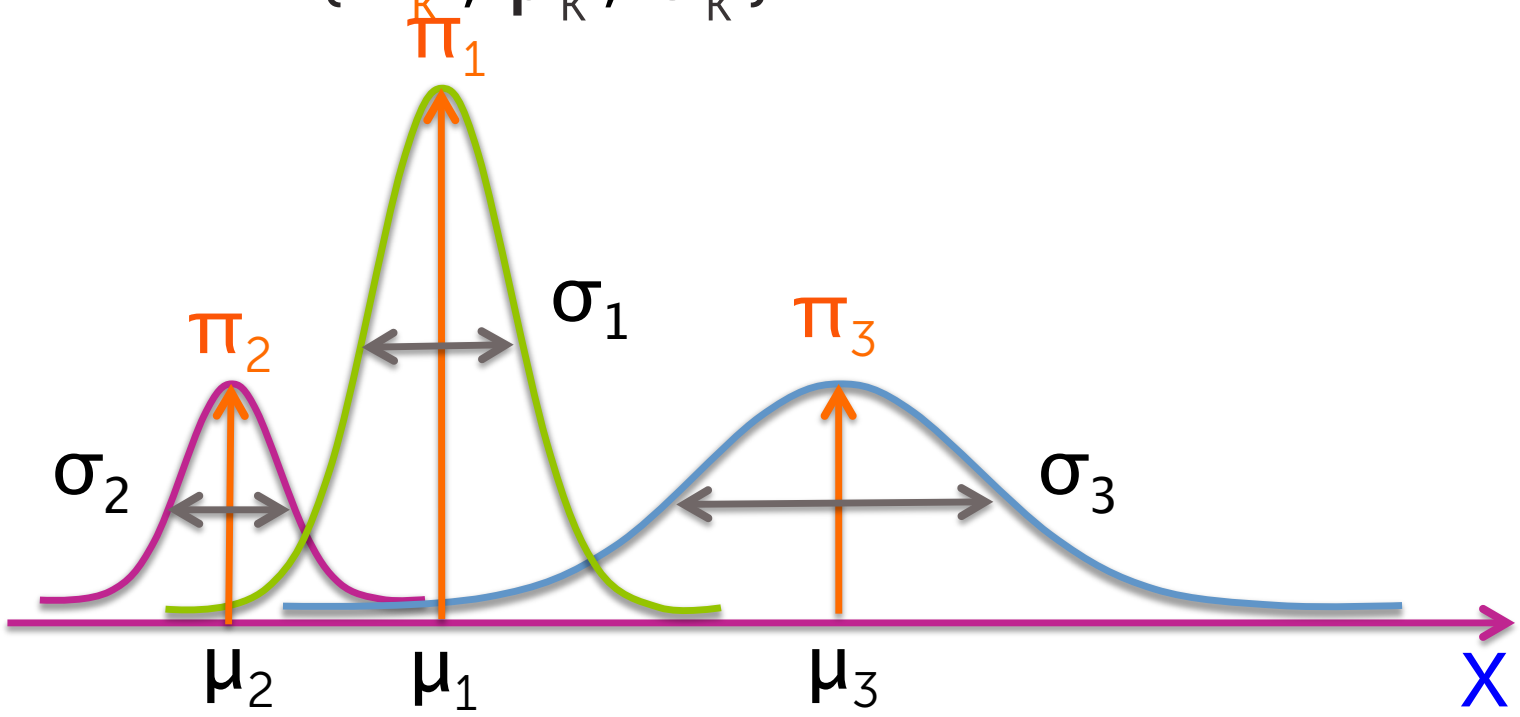
$$\sum_{k=1}^{K} \pi_k = 1$$

Convex combination

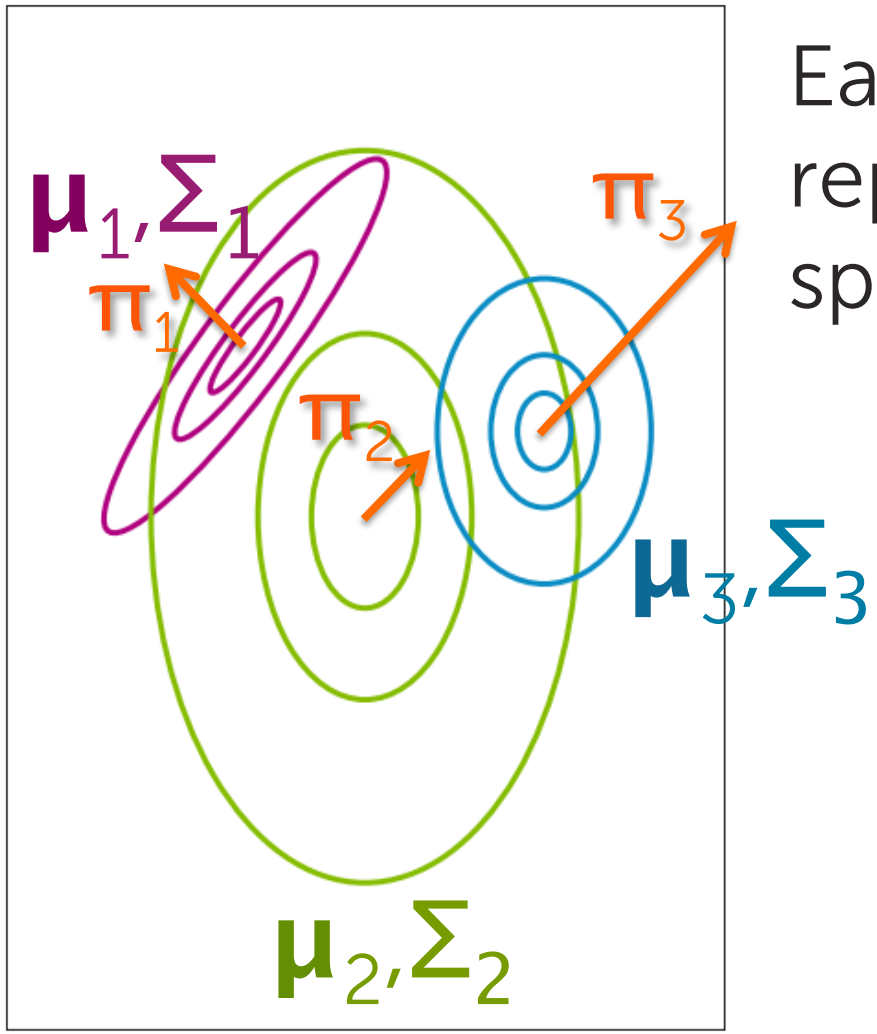
Mixture of Gaussians (1D)

Each mixture component represents a unique cluster specified by:

$$\{\pi_k, \mu_k, \sigma_k^2\}$$



Mixture of Gaussians (general)



Each mixture component represents a unique cluster specified by:

$$\{\pi_k , \boldsymbol{\mu}_k , \Sigma_k\}$$

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)

cluster assignment for obs. x_i

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

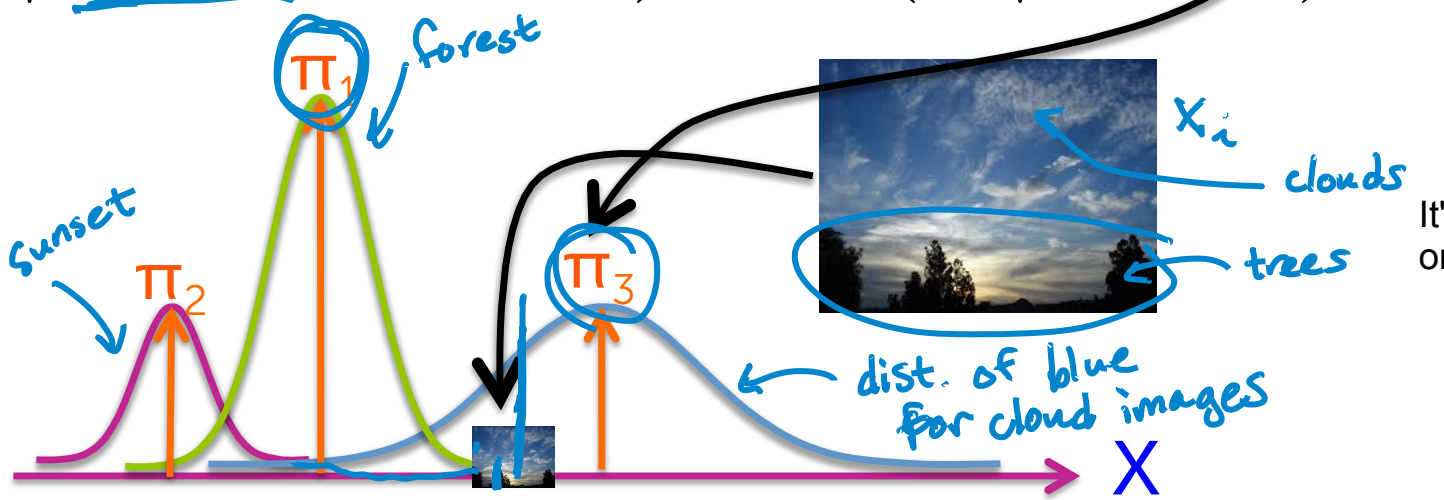
prior term

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")

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

likelihood term

$[RGB]_i$



It'd be natural to have uncertainty on the assignment of this image.

Document clustering

Discover groups of related documents

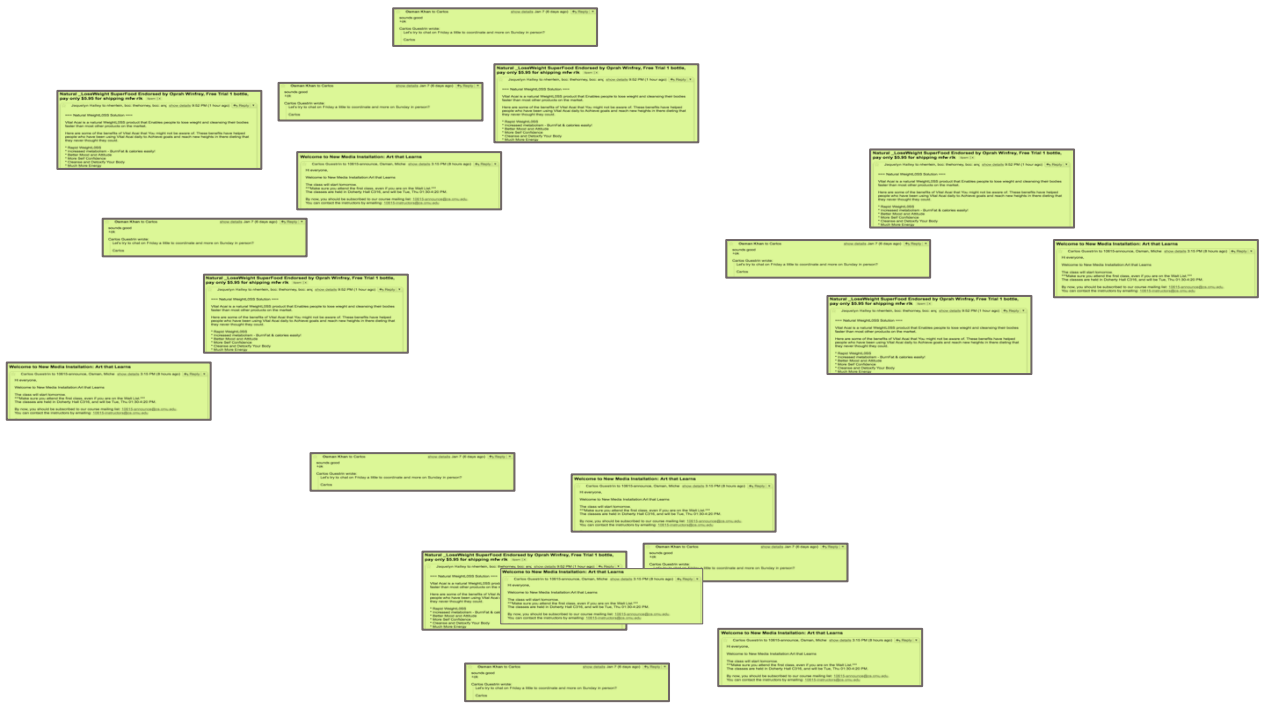
Document representation

$\mathbf{x}_i$ = tf-idf vector

Mixture of Gaussians for clustering documents

Space of all documents

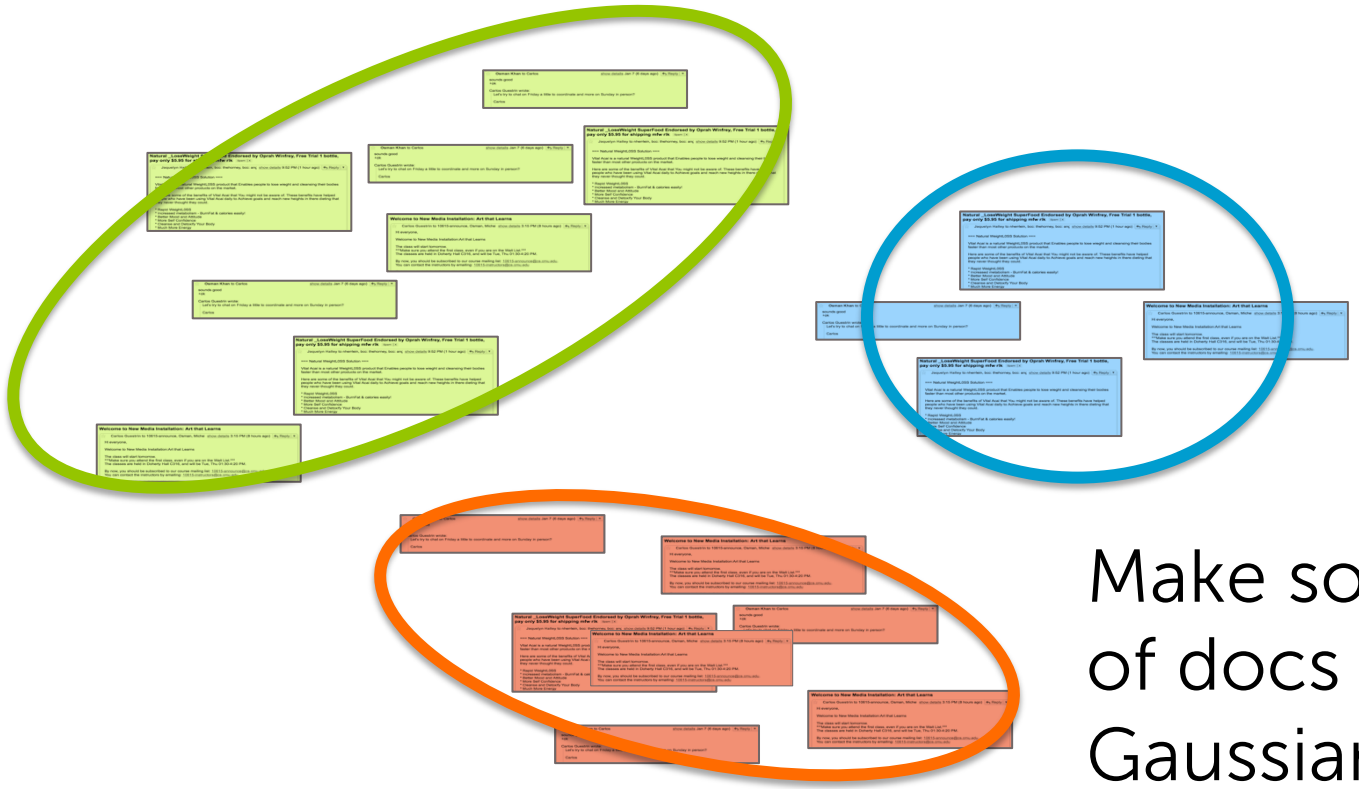
(really lives in $\mathbf{R}^V$ for vocab size V)



Mixture of Gaussians for clustering documents

Space of all documents

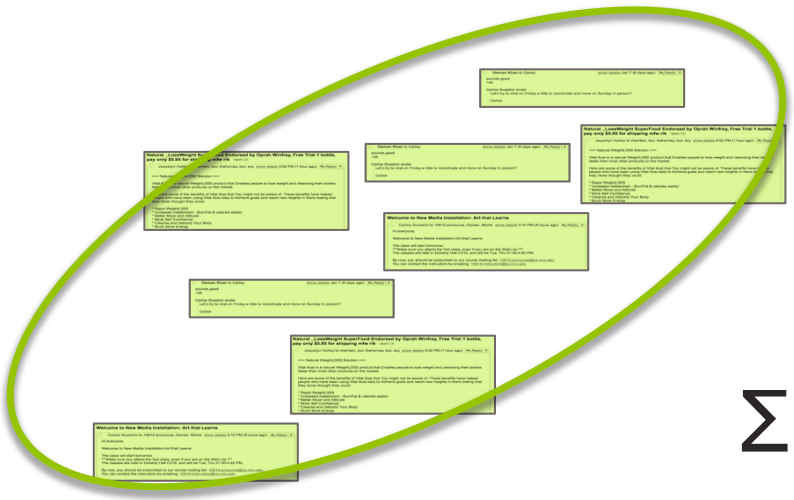
(really lives in $\mathbf{R}^V$ for vocab size V)



Make soft assignments of docs to each Gaussian

Counting parameters

Each cluster has $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$



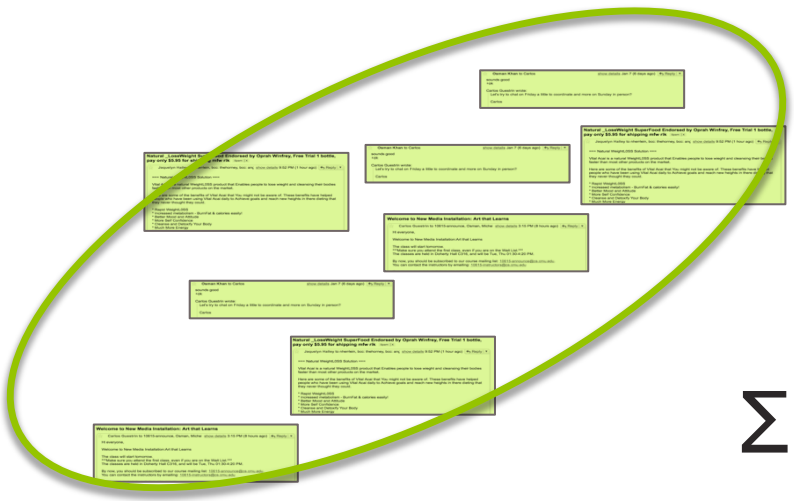
In 2D:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} \\ \sigma_{2,1} & \sigma_2^2 \end{pmatrix}$$

3 unique parameters since sigma 1,2 equals sigmal 2,1.

Counting parameters

Each cluster has $\{\pi_k, \mu_k, \Sigma_k\}$

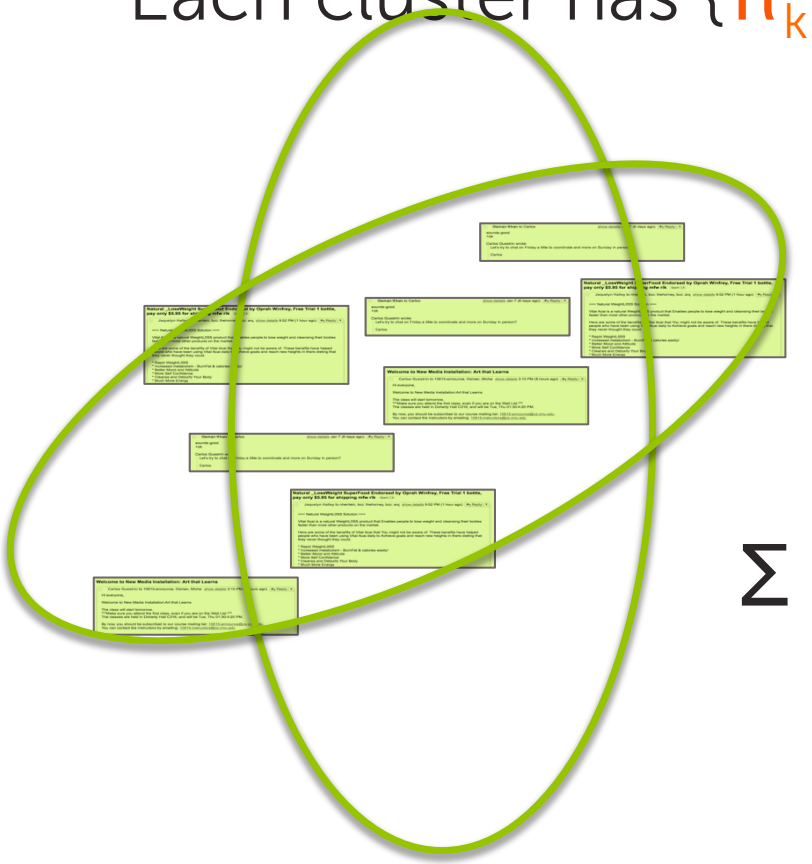


In V (vocab size) dims:

$$\Sigma = \begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \frac{V(V+1)}{2} \text{ unique parameters} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

Restricting to diagonal covariance

Each cluster has {π_k , $\boldsymbol{\mu}_k$, Σ_k diagonal }

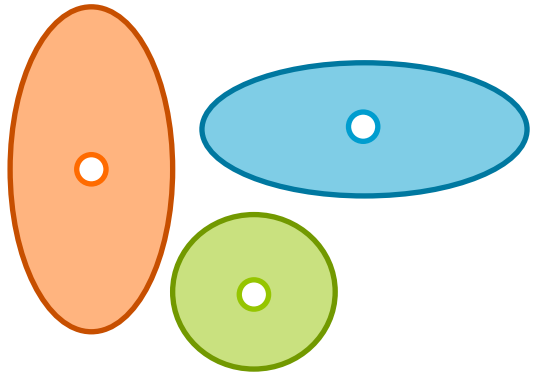


V params

$$\Sigma = \begin{pmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \sigma_3^2 & \\ 0 & & & \ddots \\ & & & \sigma_V^2 \end{pmatrix}$$

Instead of having an arbitrarily aligned ellipse, we're going to constrain ourselves just to looking at axis aligned ellipses.

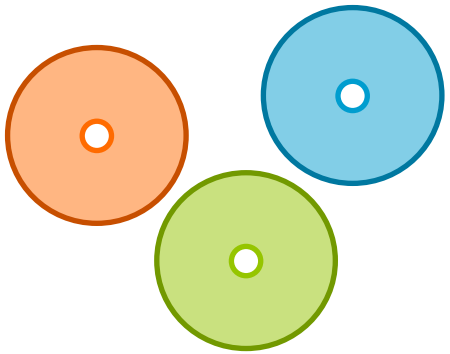
Restrictive assumption, but…

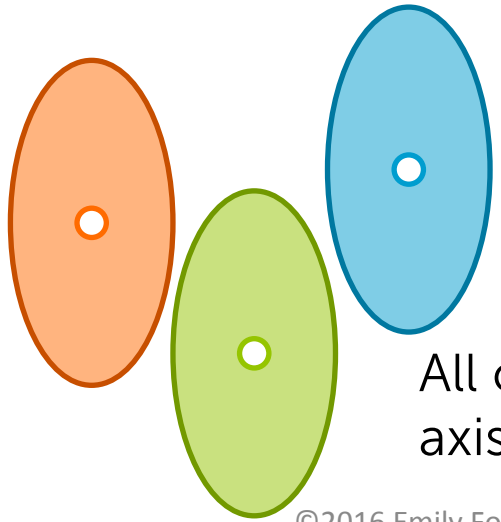


- Can **learn** weights on dimensions (e.g., weights on words in vocab)
- Can learn **cluster-specific** weights on dimensions

Still more flexible than k-means

Spherically symmetric clusters





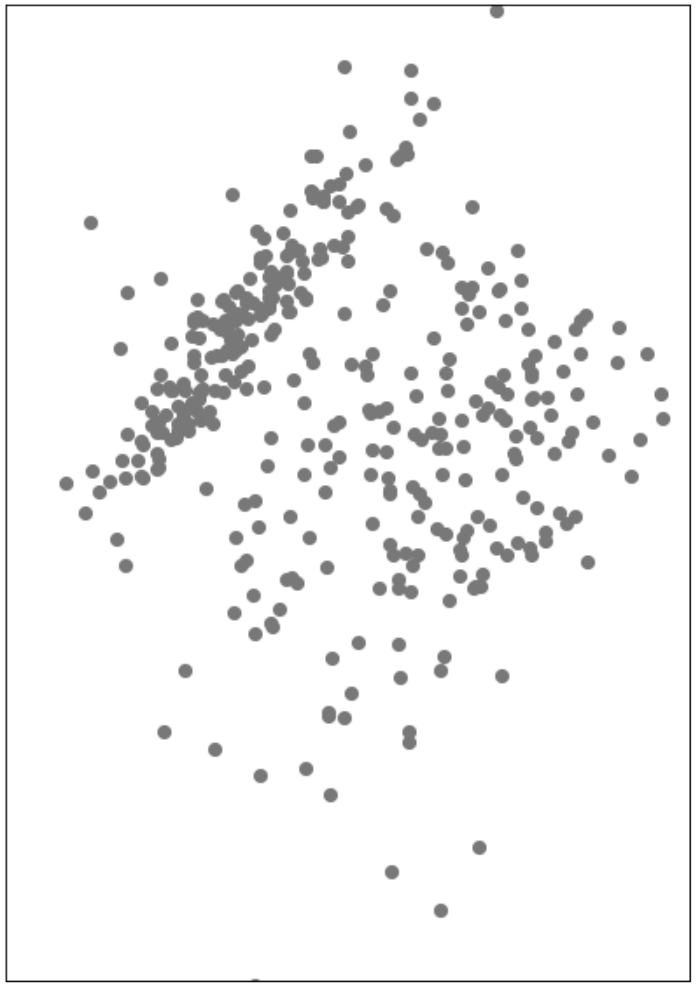
Specify weights…

All clusters have same axis-aligned ellipses

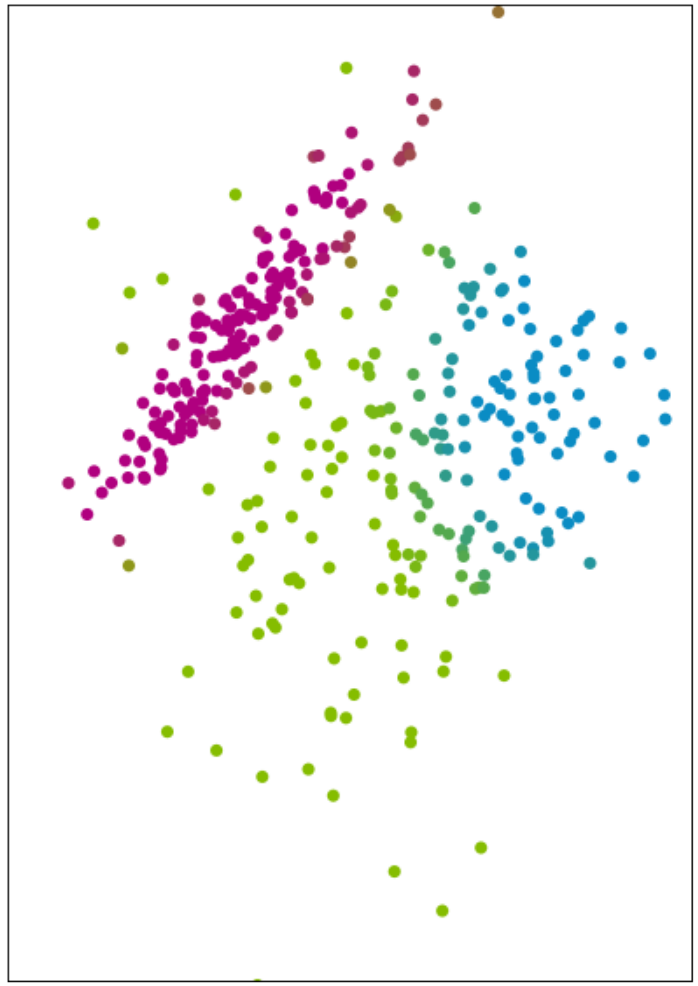
Inferring soft assignments with expectation maximization (EM)

Inferring cluster labels

Data

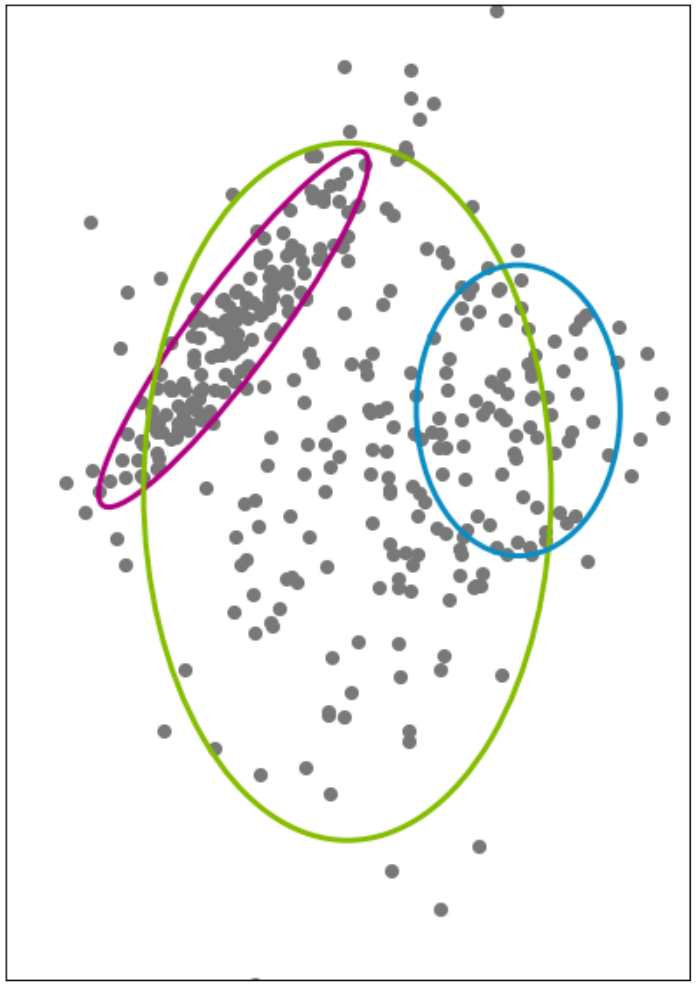


Desired soft assignments



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

Compute responsibilities



$r_i = [r_{i1}\ r_{i2}\ \ldots\ r_{iK}]$ # clusters

Responsibility cluster k takes for observation i

fixed values defining the distribution

"given"

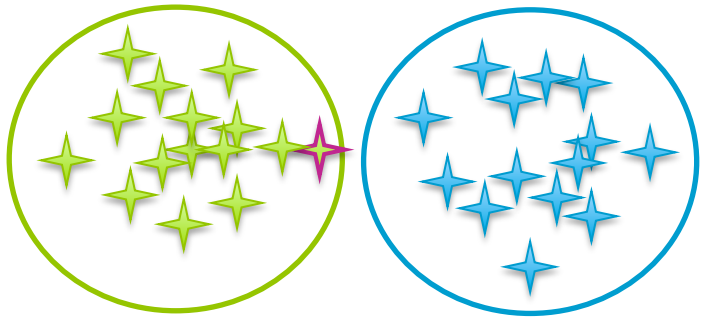
$$r_{ik} = p(z_i = k \mid \{\pi_j, \mu_j, \Sigma_j\}_{j=1}^{K}, x_i)$$

random variable

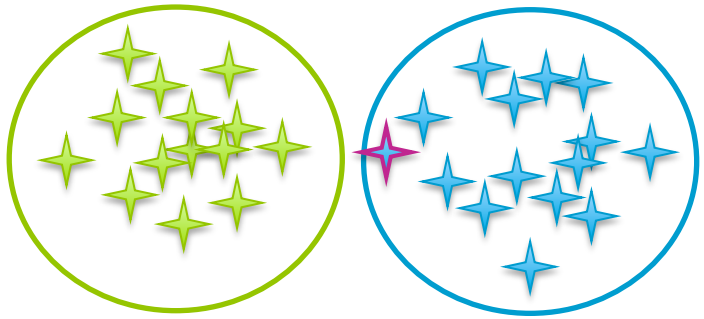
probability of assignment to cluster k

given model parameters and observed value

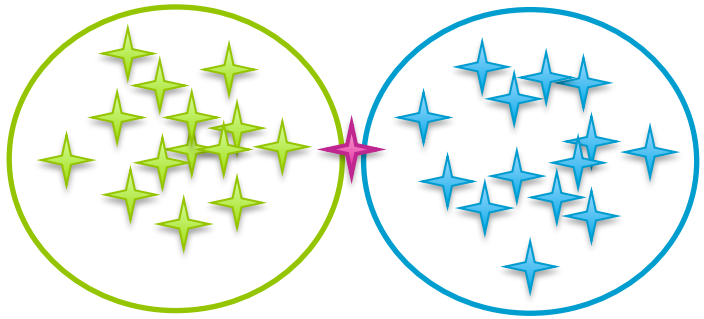
Responsibilities in pictures



Green cluster takes more responsibility



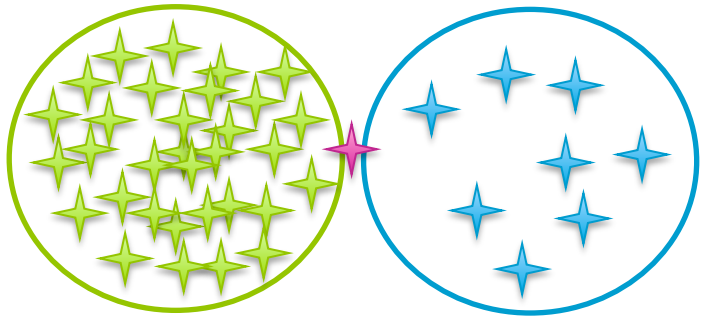
Blue cluster takes more responsibility



Uncertain… split responsibility

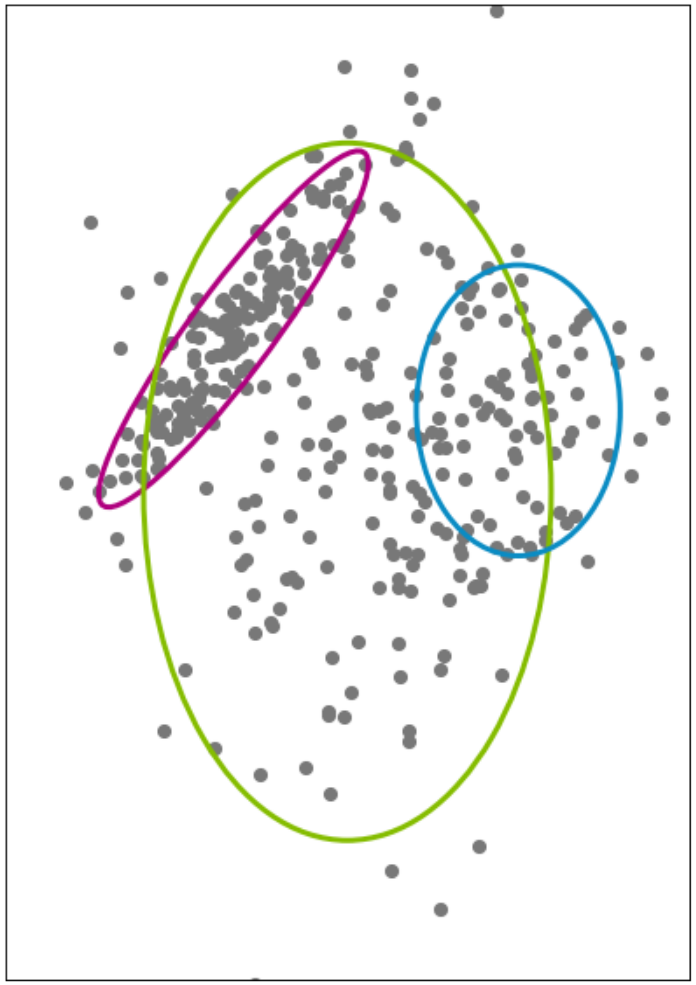
Responsibilities in pictures

Need to weight by cluster probabilities,
not just cluster shapes



Still uncertain,
but green cluster seems
more probable…
takes more responsibility

Responsibilities in equations

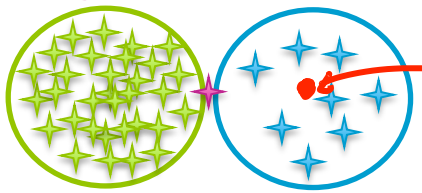


Responsibility cluster k takes for observation i

$$r_{ik} = \pi_k \; N(x_i \mid \mu_k, \Sigma_k)$$

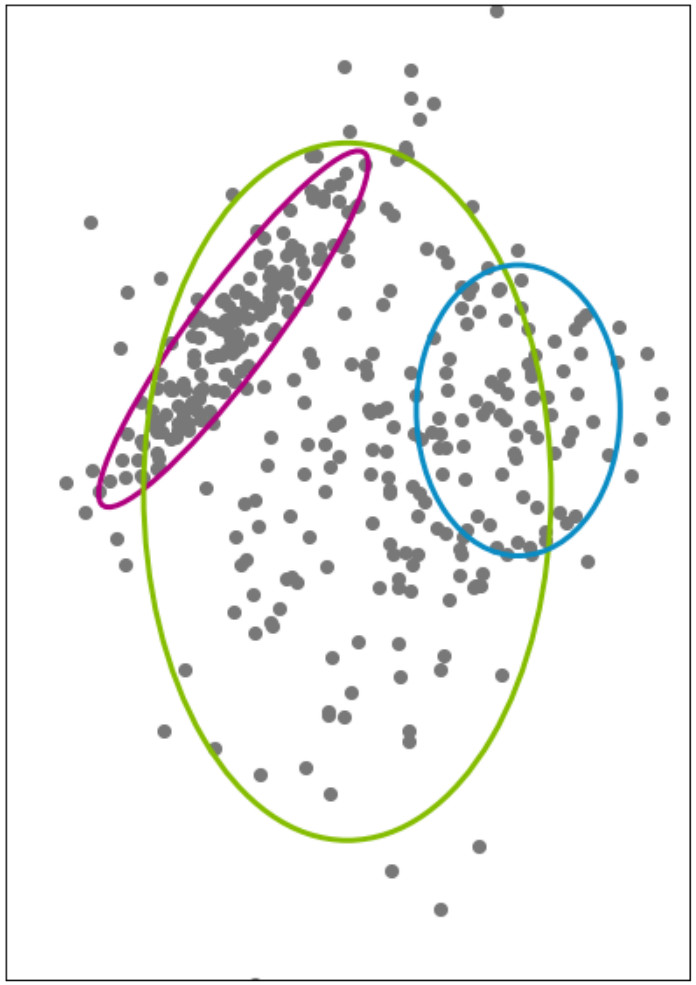
Initial probability of being from cluster k

How likely is the observed value $\mathbf{x}_i$ under this cluster assignment?



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

Responsibilities in equations



Responsibility cluster k takes for observation i

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

Normalized over all possible cluster assignments

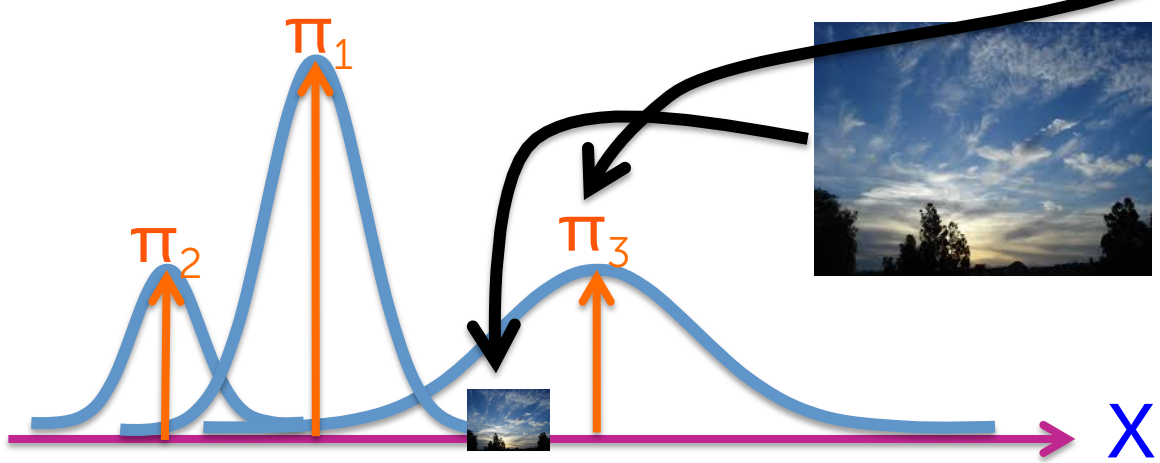
Recall: According to the model…

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

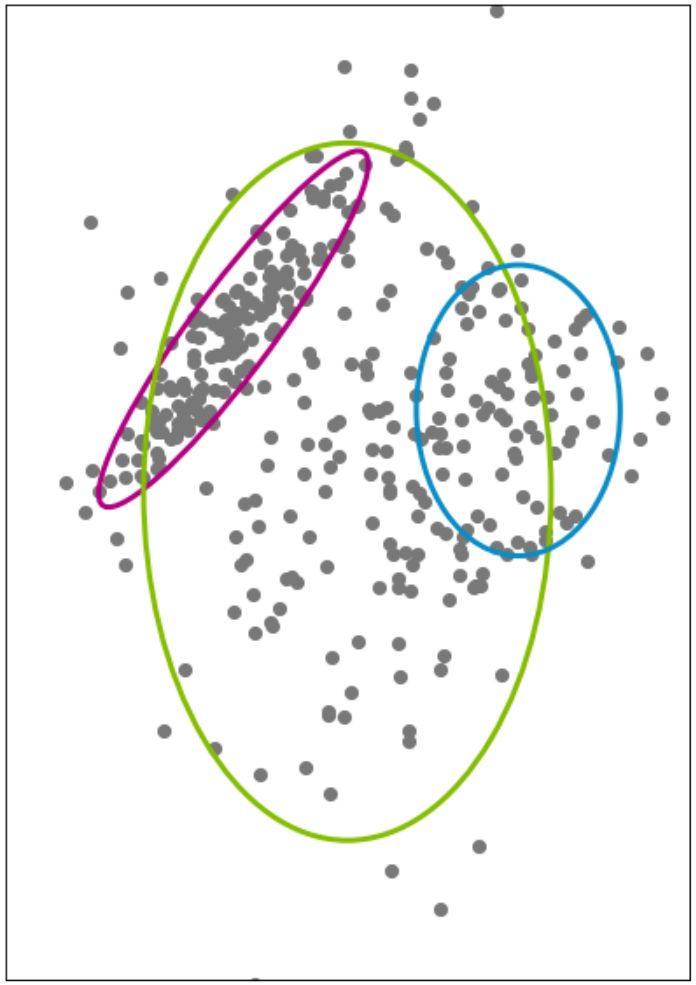
$$p(z_i = k) = \pi_k$$

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")

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



Part 1 summary



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

But, we don't know these!

Responsibility calculation as and application of Bayes' rule

An application of Bayes' rule

Responsibility cluster k takes for observation i

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

params

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

$p(z_i = k \mid \text{params})$

$p(x_i \mid z_i = k, \text{params})$

An application of Bayes' rule

Responsibility cluster k takes for observation i

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

A params B

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

$p(z_i = j \mid \text{params})$

$p(x_i \mid z_i = j, \text{params})$

An application of Bayes' rule

$$r_{ik} = p(A|B,\text{params})$$

$$= \frac{p(A|\text{params})p(B|A,\text{params})}{\sum_C p(C|\text{params})p(B|C,\text{params})}$$

by definition equivalent to $p(B,C|\text{params})$

events B and C

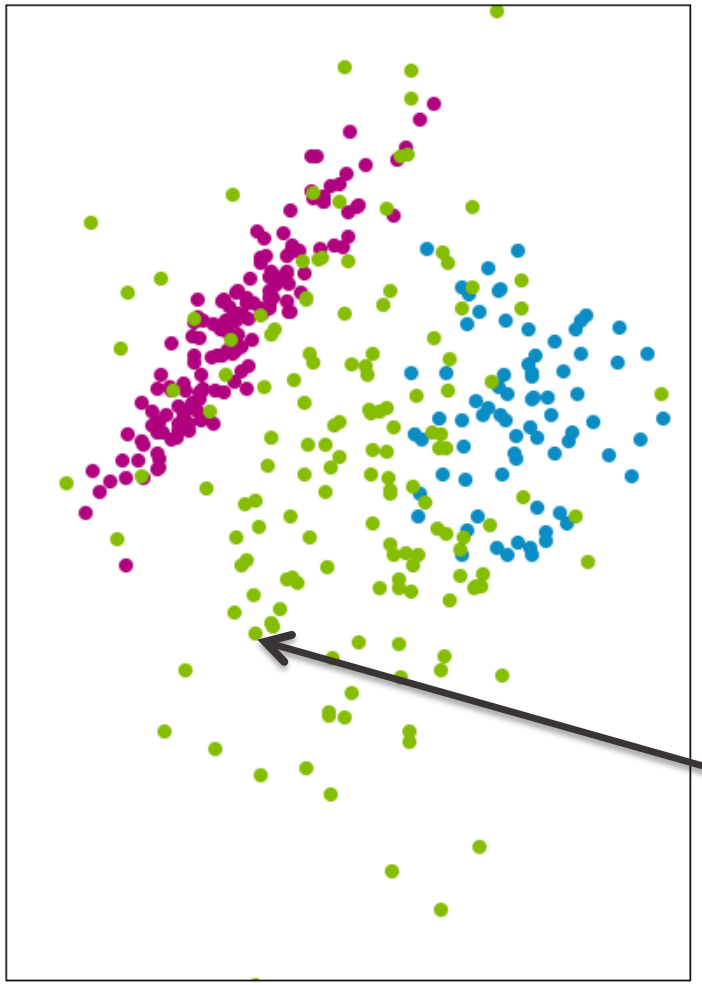
$$\sum_C p(B,C|\text{params}) = p(B|\text{params})$$

$$= \frac{p(A|\text{params})p(B|A,\text{params})}{p(B|\text{params})}$$

more general form of $p(A|B) = \frac{p(A)p(B|A)}{p(B)}$

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?

Only observations that are assigned to a given cluster inform the parameters of that cluster.

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 $\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\}$ given data assigned to cluster k

maximum likelihood estimation (MLE)

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

Mean/covariance MLE

The hat (^) above each variable denotes estimate, specifically maximum likelihood estimate.

sum these vectors

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

divide by 3 (the total # of obs.)

denotes "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

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

The MLE is also given by what's called sample covariance estimate.

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

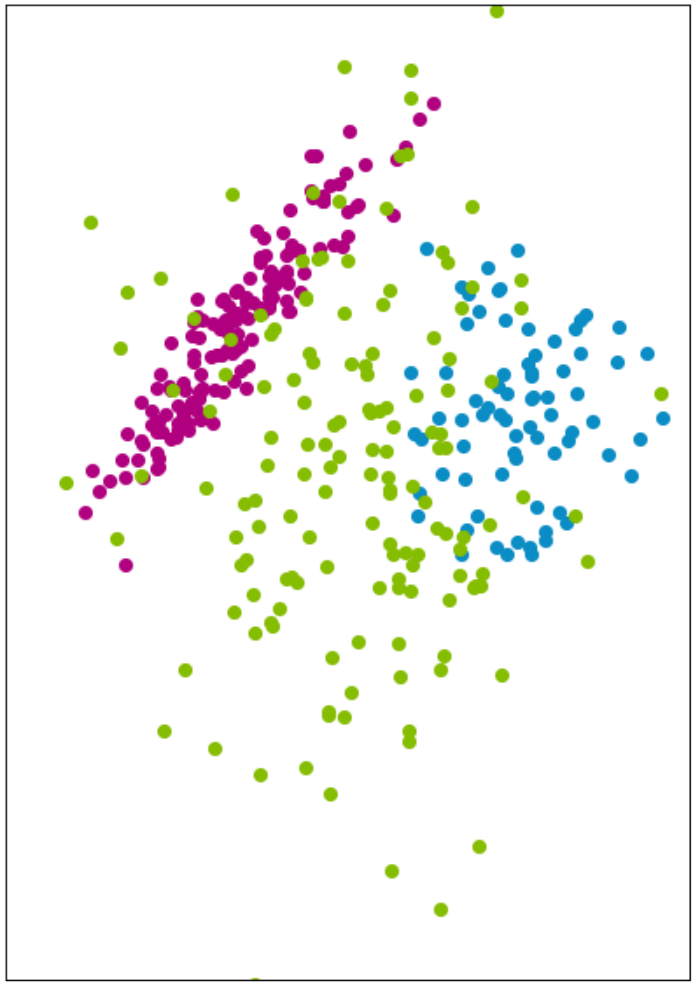
obs in cluster k

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

total # of obs

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

Part 2a summary



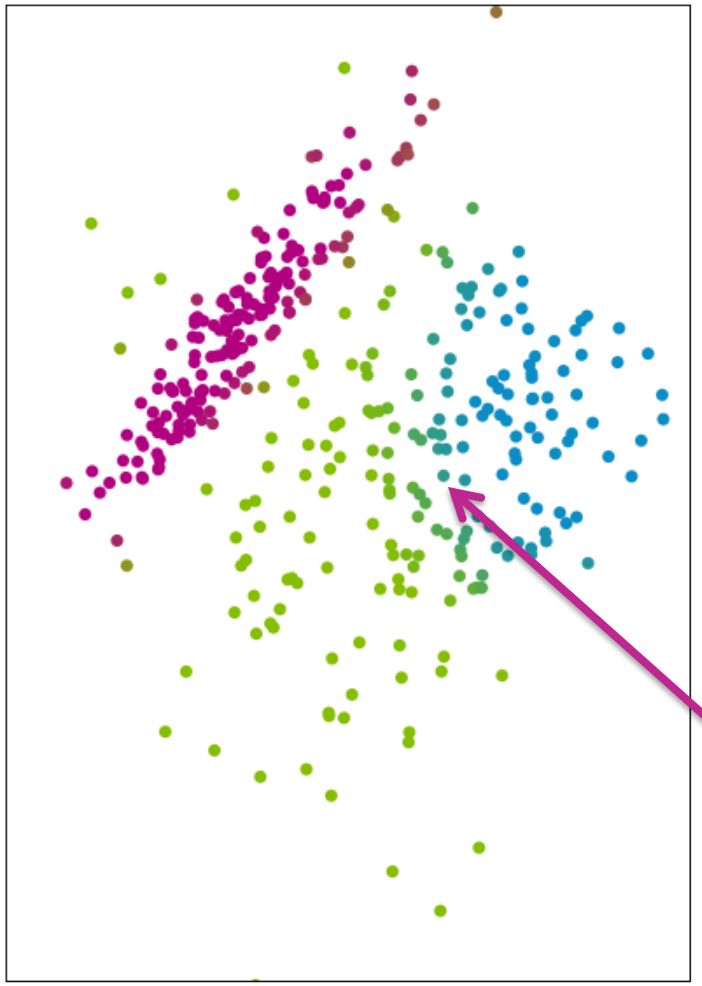
needed to compute soft assignments

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	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

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
$\mathbf{x}_2[1]$			
$\mathbf{x}_3[1]$			
$\mathbf{x}_4[1]$			
$\mathbf{x}_5[1]$			
$\mathbf{x}_6[1]$			

R	G	B	Cluster 2 weights
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	0.18
$\mathbf{x}_2[1]$			
$\mathbf{x}_3[1]$			
$\mathbf{x}_4[1]$			
$\mathbf{x}_5[1]$			
$\mathbf{x}_6[1]$			

R	G	B	Cluster 3 weights
$\mathbf{x}_1[1]$	$\mathbf{x}_1[2]$	$\mathbf{x}_1[3]$	0.52
$\mathbf{x}_2[1]$	$\mathbf{x}_2[2]$	$\mathbf{x}_2[3]$	0.73
$\mathbf{x}_3[1]$	$\mathbf{x}_3[2]$	$\mathbf{x}_3[3]$	0.99
$\mathbf{x}_4[1]$	$\mathbf{x}_4[2]$	$\mathbf{x}_4[3]$	0.15
$\mathbf{x}_5[1]$	$\mathbf{x}_5[2]$	$\mathbf{x}_5[3]$	0.02
$\mathbf{x}_6[1]$	$\mathbf{x}_6[2]$	$\mathbf{x}_6[3]$	0.01

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

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

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

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

Total weight in cluster k = effective # obs

Compute cluster parameter estimates with weights on each row operation

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

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

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

Total weight in cluster k = effective # obs

Estimate cluster proportions from relative weights

Total weight in cluster:

1.242	2.8	2.42

Total weight in dataset: 6

datapoints N

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:

1	2	3

Equating the estimates…

Blue hand written notes in this page talks about how soft assignment equations also apply to hard assignment case.

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

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

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

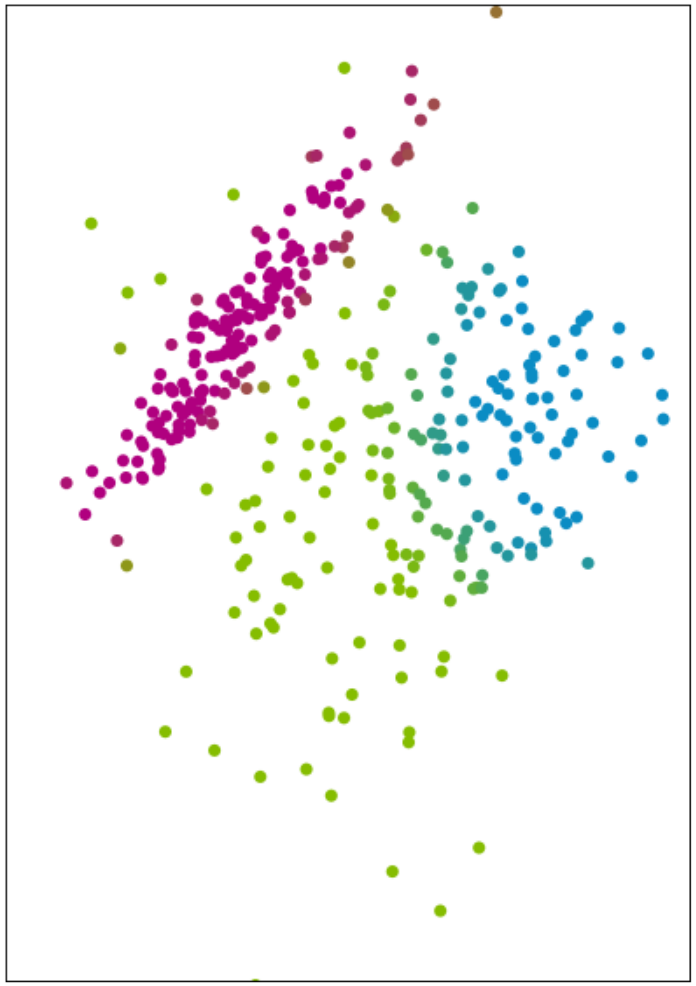
only add x_i if i in k ($r_{ik}=1$) $\rightarrow \frac{1}{N_k} \sum_{i \text{ in } k} x_i$ ✓

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

same as above

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

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:

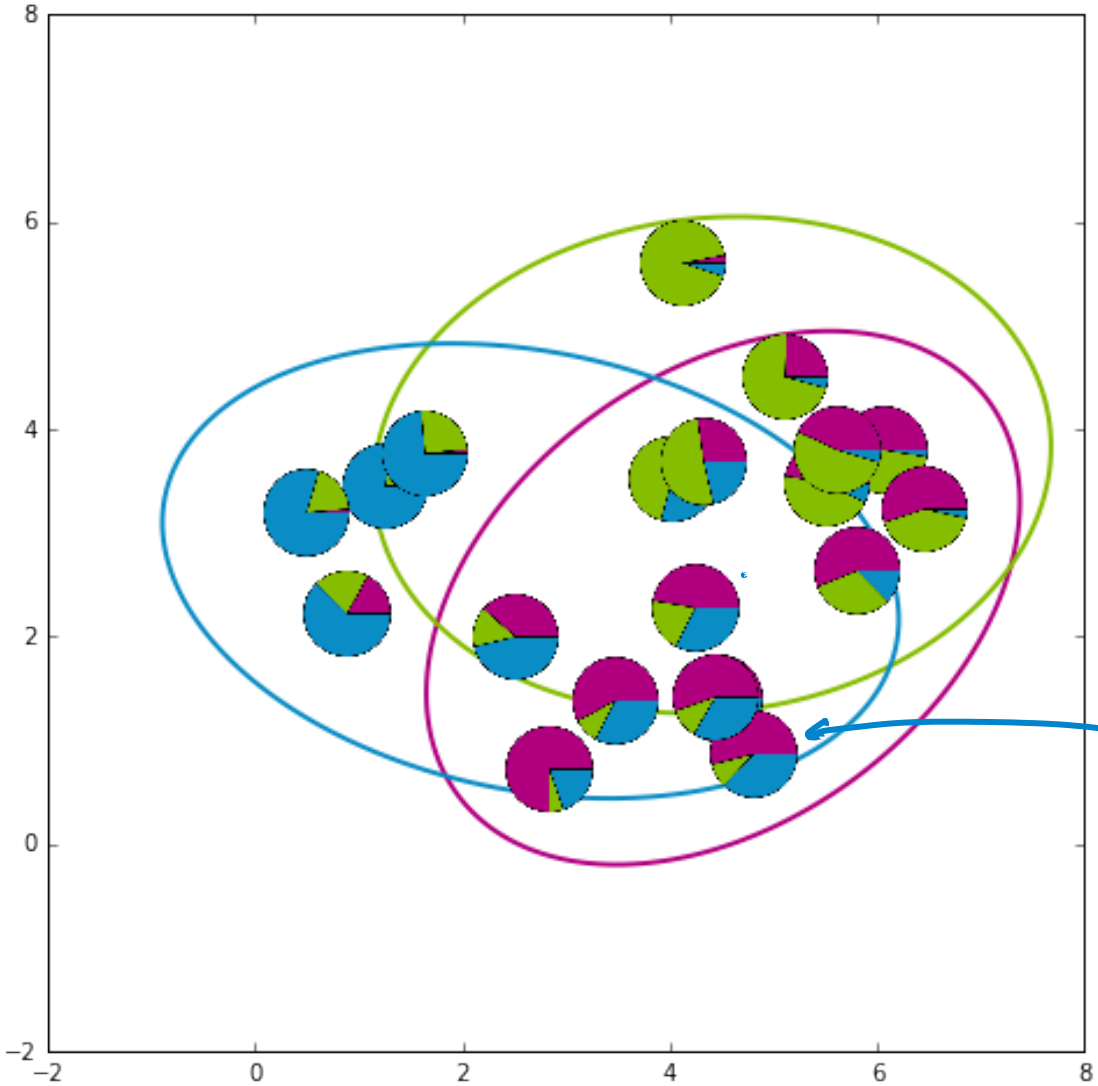
1. **E-step:** estimate cluster responsibilities given current parameter estimates

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

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

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

EM for mixtures of Gaussians in pictures – initialization



Initialize set of model parameters

iter counter

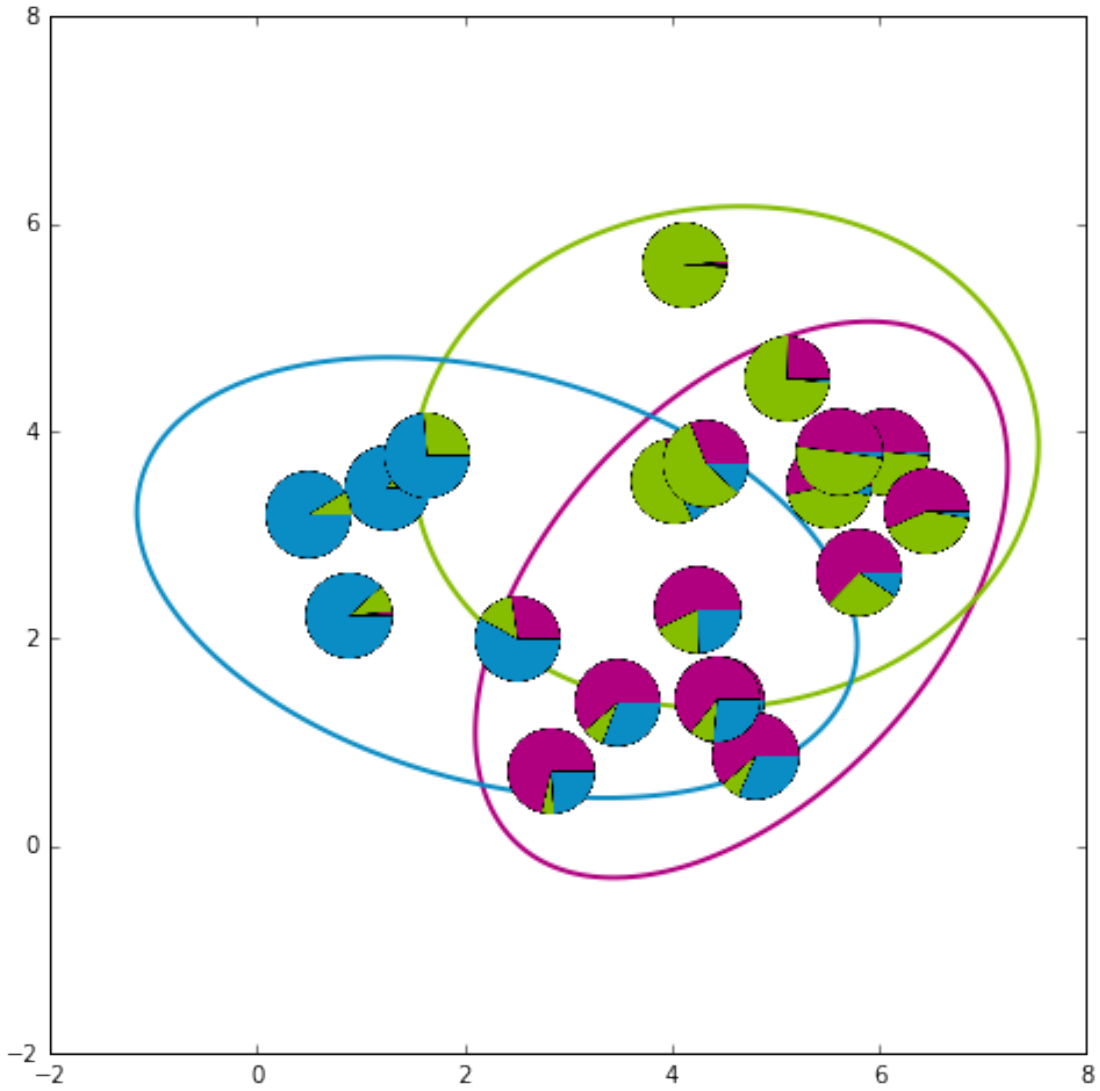
$\{\pi_k^{(0)}, \hat{\mu}_k^{(0)}, \hat{\Sigma}_k^{(0)}\}$

Then compute

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

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

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

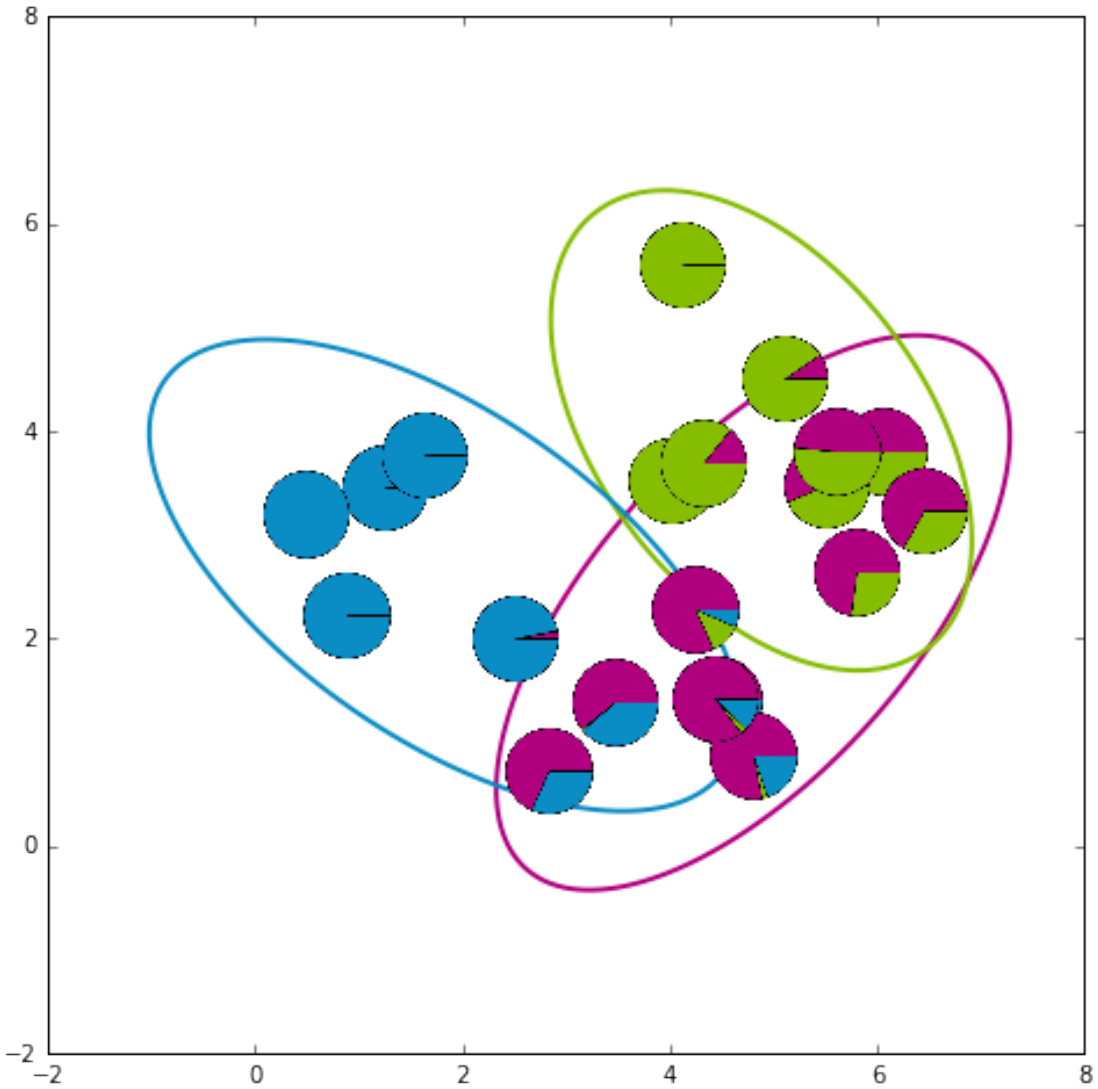


Maximize likelihood given soft assign. $r_{ik}^{(1)}$

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

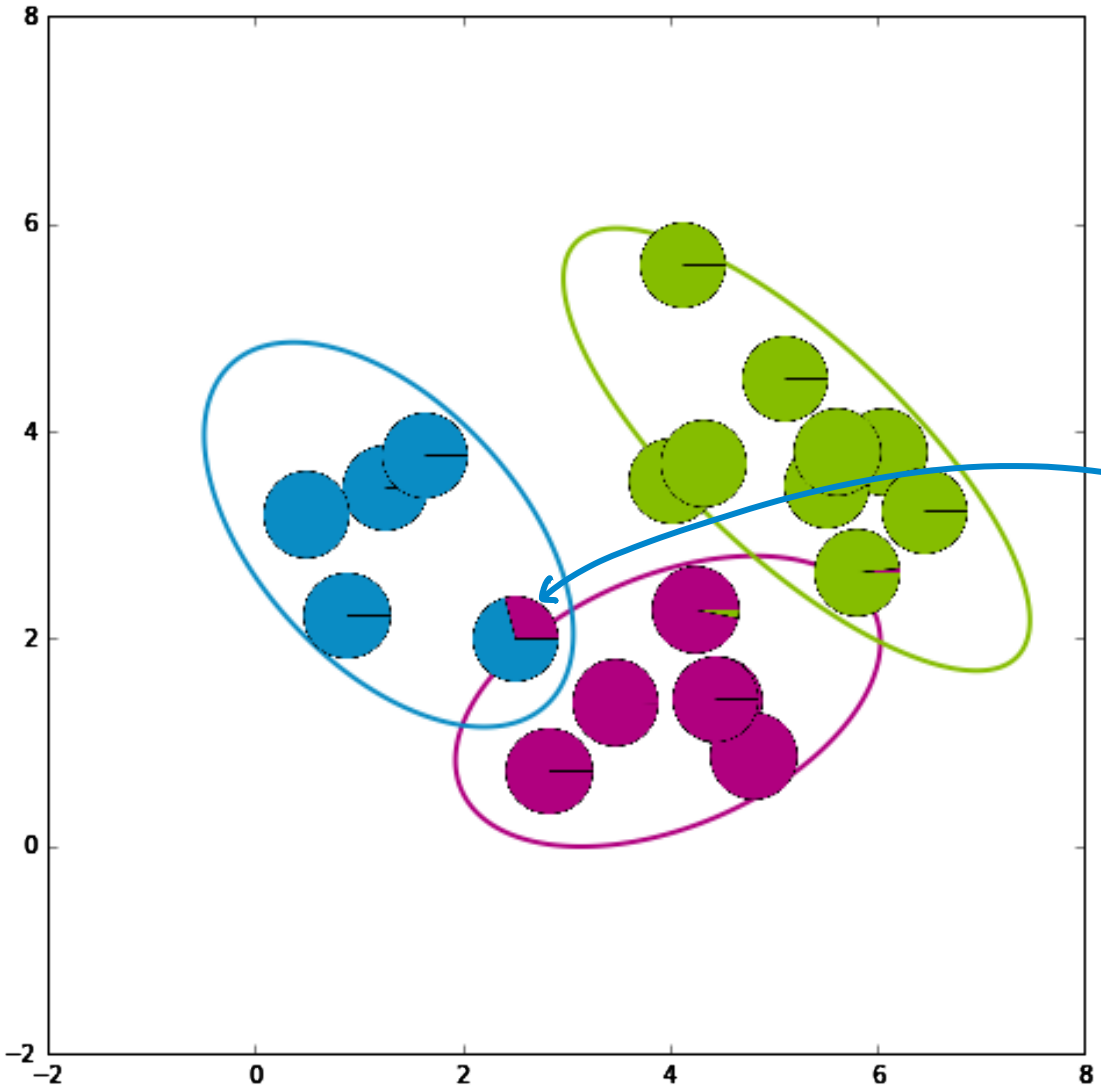
Then recompute responsibilities $\hat{r}_{ik}^{(2)}$

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



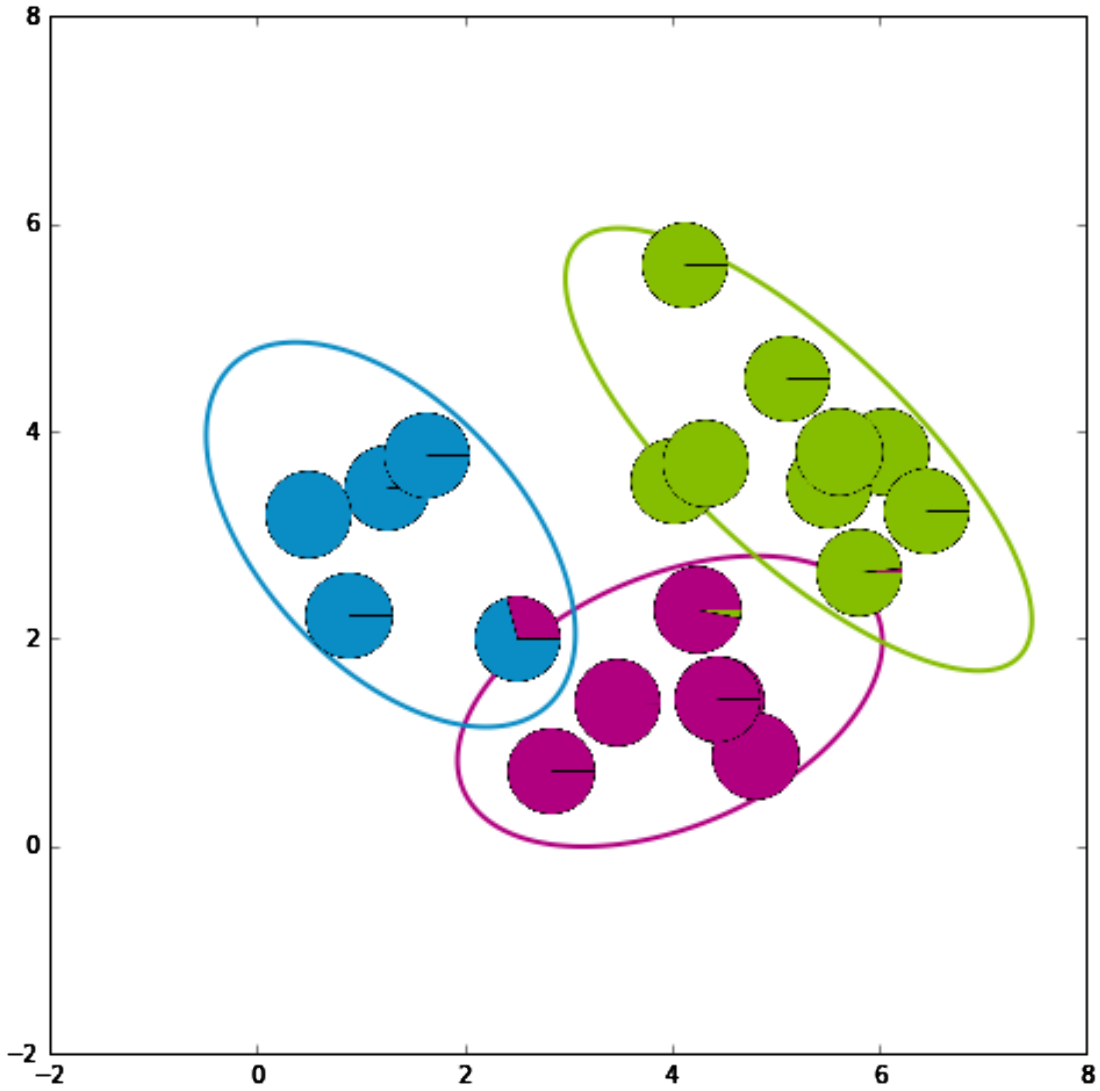
rinse + repeat until convergence

EM for mixtures of Gaussians in pictures – converged solution



clearly see uncertainty in assignment of obs. to blue or fuchsia cluster, even in final assignments.

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
- Convergences to a local mode

- We will assess via (log) likelihood of data under current parameter and responsibility estimates

Initialization

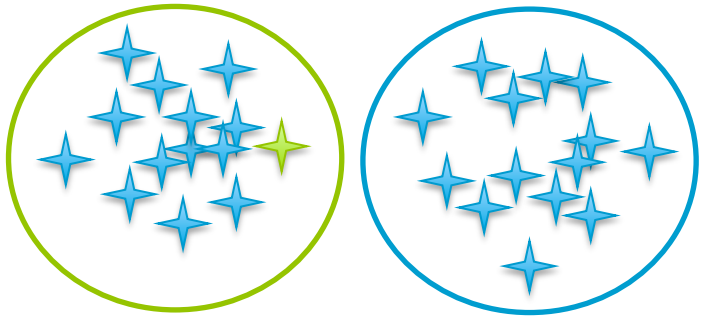
- Many ways to initialize the EM algorithm
- Important for convergence rates and quality of local mode found
- Examples:
  - Choose K observations at random to define K "centroids". Assign other observations to nearest centriod to form initial parameter estimates. (hard assignment)
  - Pick centers sequentially 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 ∞ !

<= All driven by the fact that the likelihood of the green observation, under a Gaussian that's absolutely certain that the only value it can take is the value that was observed, is infinite. That's going to completely dominate the overall likelihood of this set of assignment.

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 $\boldsymbol{\mu}_k[w] = \mathbf{x}_i[w]$ and $\sigma^2_{w,k} = 0$

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 → 0!

Add small amount to diagonal of covariance estimate

Alternatively, take Bayesian approach and place prior on parameters.

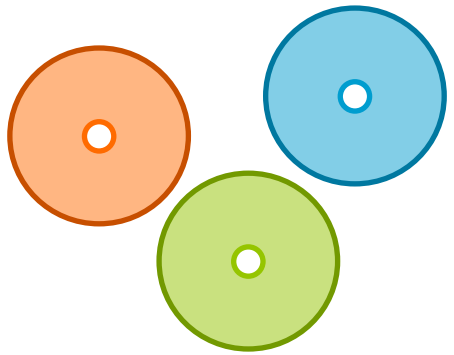
Similar idea, but all parameter estimates are "smoothed" via cluster pseudo-observations.

Relationship to k-means

Consider Gaussian mixture model with

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

Spherically symmetric clusters



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

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

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

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

Infinitesimally small variance EM = k-means

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

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

Infinitesimally small

Decision based on distance to nearest cluster center

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

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

Summary for mixture models and the EM algorithm

What you can do now…

- Interpret a probabilistic model-based approach to clustering using mixture models
- Describe model parameters
- Motivate the utility of soft assignments and describe what they represent
- Discuss issues related to how the number of parameters grow with the number of dimensions
  - Interpret diagonal covariance versions of mixtures of Gaussians
- Compare and contrast mixtures of Gaussians and k-means
- Implement an EM algorithm for inferring soft assignments and cluster parameters
  - Determine an initialization strategy
  - Implement a variant that helps avoid overfitting issues