Mixture Models and EM:
Model-Based Clustering

CSE 446: Machine Learning
Slides by Emily Fox
University of Washington
May 22, 2019

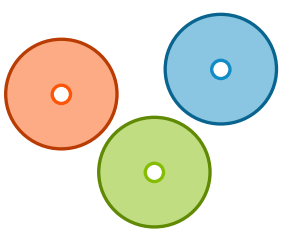
Limitations of k-means

Assigns observations to closest cluster center

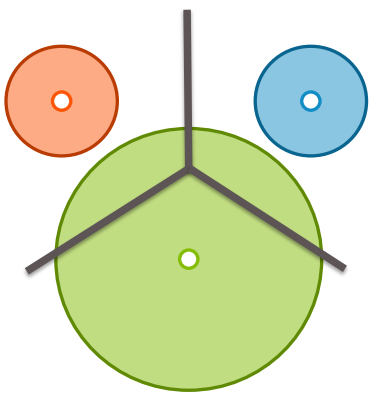
$$z_i \leftarrow \arg\min_j ||\mu_j - \mathbf{x}_i||_2^2$$

Only center matters
Not cluster shapes

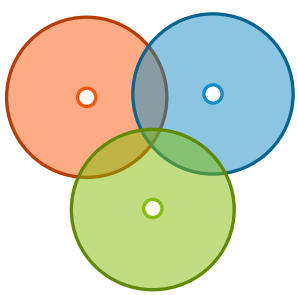
Equivalent to assuming
spherically symmetric clusters



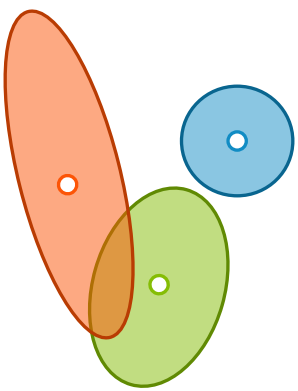
Failure modes of k-means



disparate cluster sizes



overlapping clusters

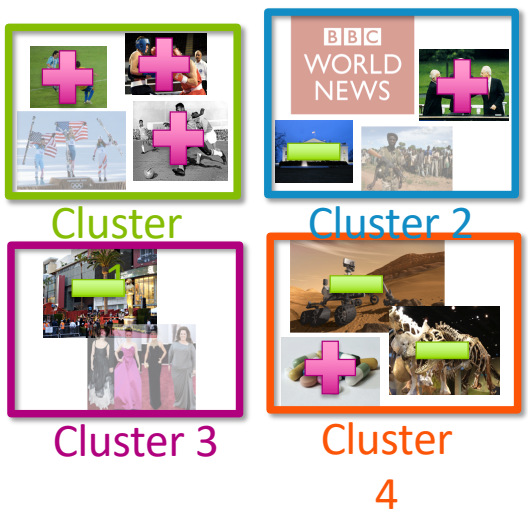


different shaped/oriented clusters

Motivates probabilistic model: Mixture model

- Take uncertainty in assignment into account
  e.g., when clustering documents, might want to say 54% chance document is world news, 45% science, 1% sports, and 0% entertainment

- Accounts for cluster shapes not just centers

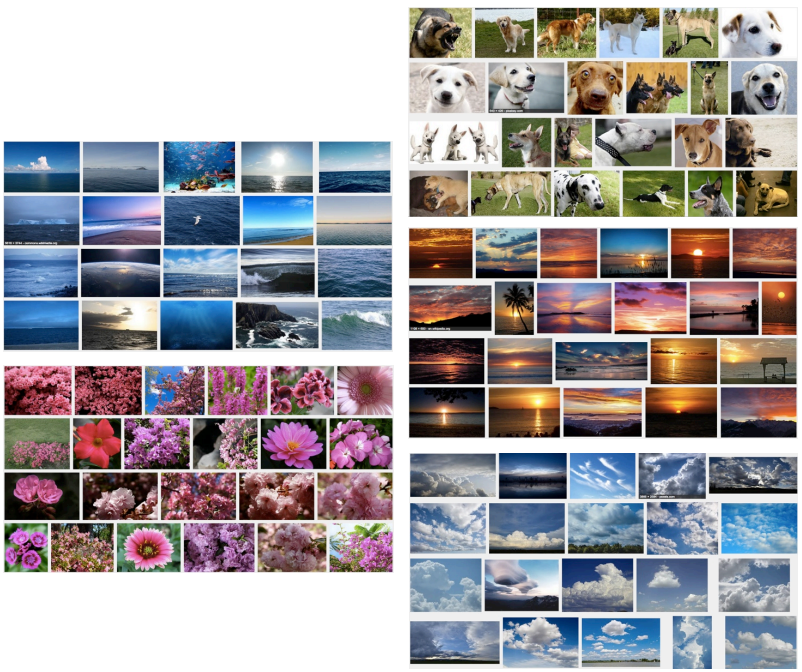


Mixture models

Motivating application: Clustering images

Discover groups of similar images

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

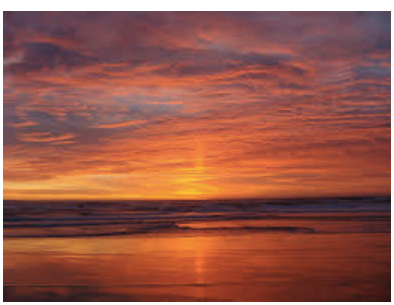


Simple image representation

Consider average red, green, blue pixel intensities

average RGB value over all pixels in entire image.

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



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

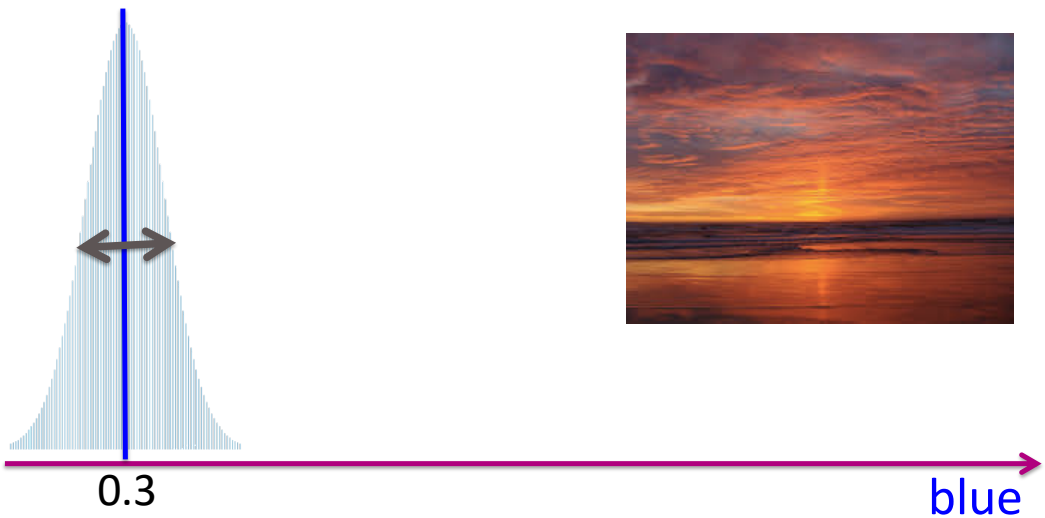
[R = 0.02, G = 0.95, B = 0.4]

Distribution over all cloud images

Let's look at just the blue dimension

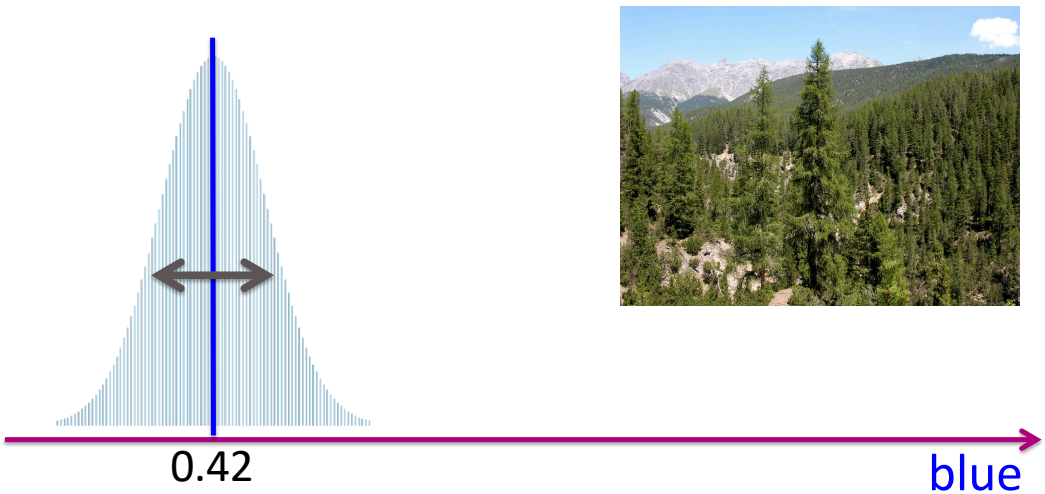
Distribution over all sunset images

Let's look at just the blue dimension

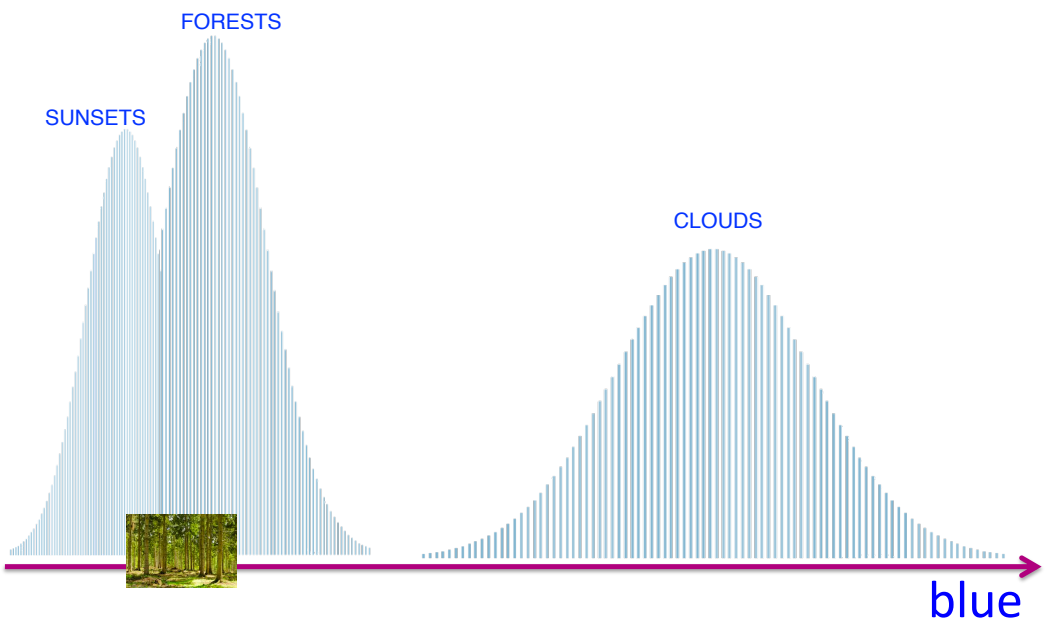


Distribution over all forest images

Let's look at just the blue dimension



Distribution over all images



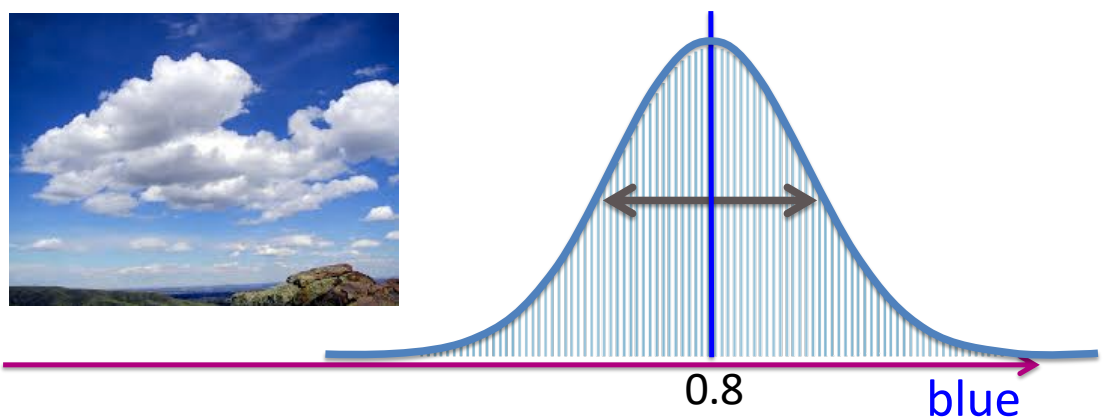
Can be distinguished along other dim

Now look at the red dimension

Background: Gaussian distributions

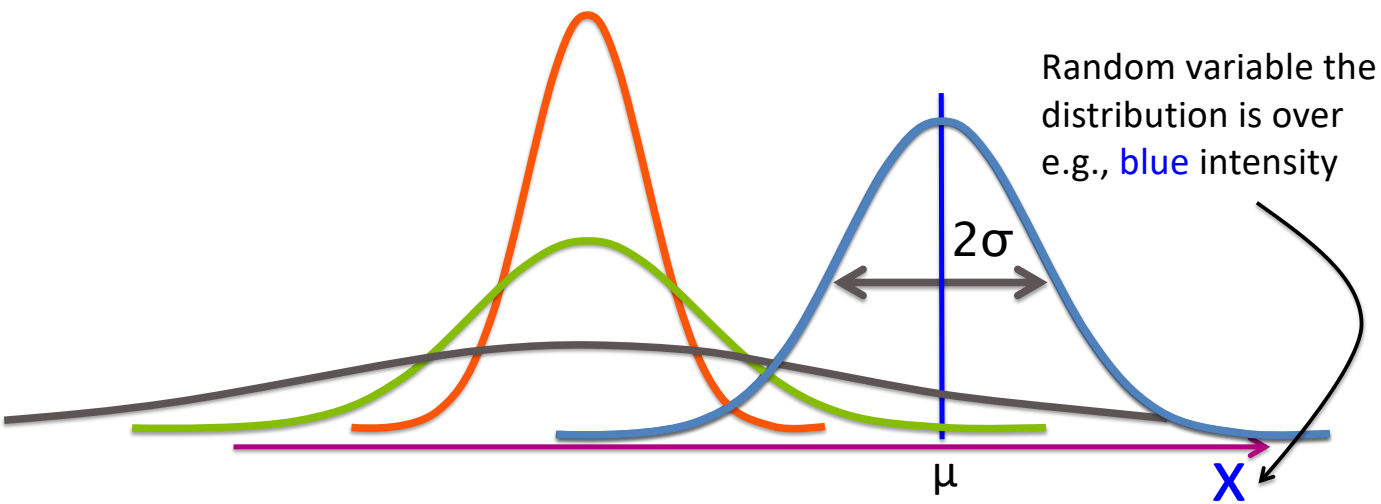
Model for a given image type

For each dim 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 st. dev. σ)

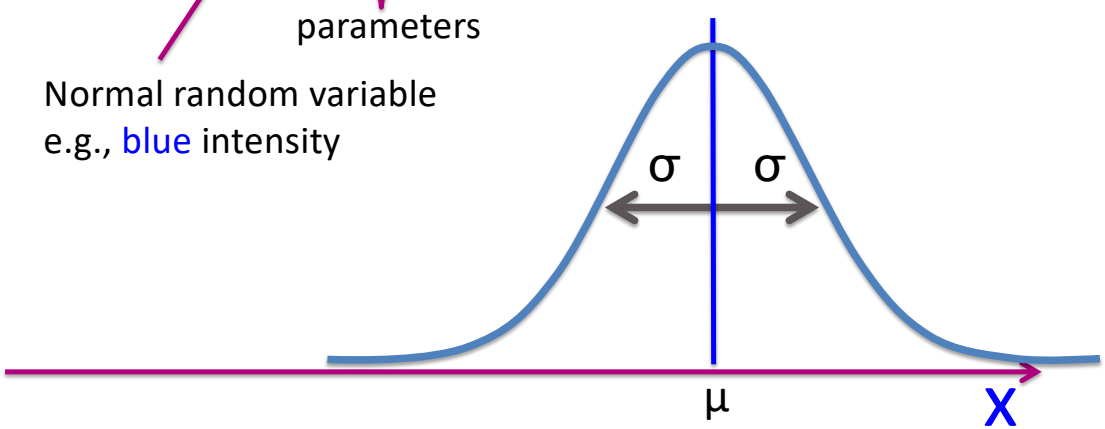


Notation a 1D Gaussian distribution

Probability that X takes on the value x. $\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$

parameters

Normal random variable
e.g., blue intensity



Multivariate Gaussian density

quadratic form
Note that covariance matrix is always positive definite
(unless degenerate; i.e. one of the variables has no variance)

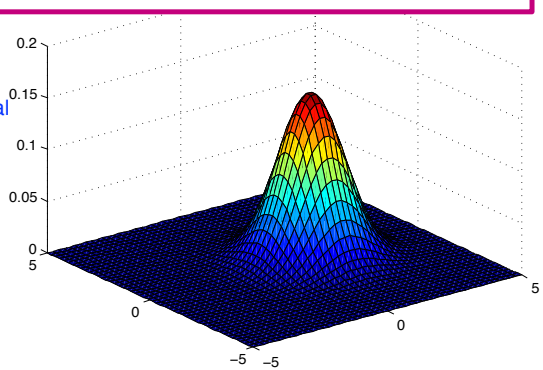
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

determinant
= (2pi)^d | SIgma|

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

parameters

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



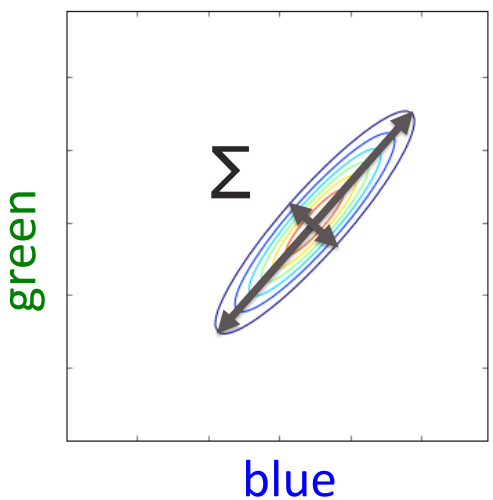
Multivariate Gaussian

Fully specified by mean μ and covariance Σ

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

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

covariance determines orientation + spread



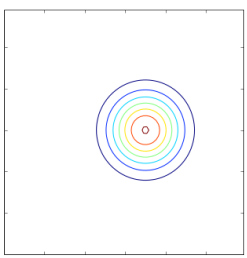
Spread in this direction is sigma_blue^2

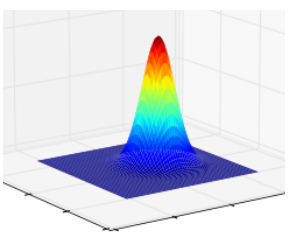
Covariance structure

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}=0, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right\}$$

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

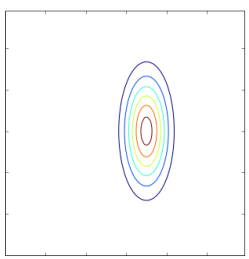


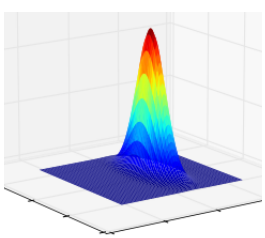


Covariance structure

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}=0,\boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}}\exp\left\{-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right\}$$

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



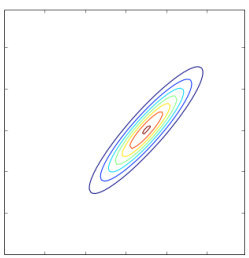


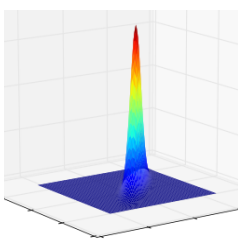
x1/ sigma_1^2 + x2/sigma_2^2 = C —> level sets are ellipses!

Covariance Structure

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}=0,\boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}}\exp\left\{-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right\}$$

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





eigenvector decomposition

Sigma = QDQ^T —> Sigma^{-1} = QD^-1Q^T

x^TQD^-1Q^Tx

Let y = Q^Tx —> projection of x into the basis provided by Q

= y^TD^-1y
= y1 / sigma_1^2 + y2 / sigma_2^2 —> ellipse in the new coordinate system specified by columns of Q (the eigenvectors)

Important facts

- **Affine Property**

$$\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$A\mathbf{X} + \mathbf{b} \sim N(A\boldsymbol{\mu} + \mathbf{b}, A\boldsymbol{\Sigma}A^T)$$

- **Constructing:** $X_1, \ldots, X_d \sim N(0,1)$ independent. Then $\mathbf{X} \sim N(0, I)$. Then

$$A\mathbf{X} + \boldsymbol{\mu} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$\boldsymbol{\Sigma} = AA^T$$

- **Sphering:**

If $\boldsymbol{\Sigma}$ is psd, symmetric, then

$$\mathbf{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \rightarrow \quad A^{-1}(\mathbf{Y} - \boldsymbol{\mu}) \sim N(0, I)$$
$$\boldsymbol{\Sigma} = AA^T$$

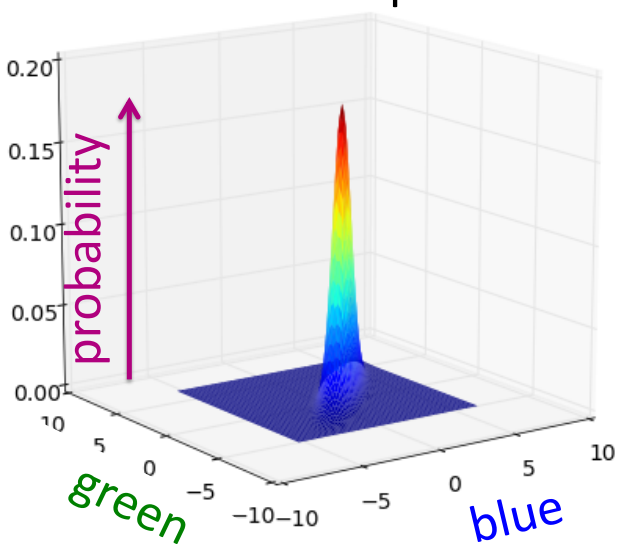
More intuition

1. Start with $\mathbf{X} \sim N(0, I)$
2. (Scaling step) $D^{1/2}\mathbf{X} \sim N(0, D)$.
3. (Rotation) $UD^{1/2}\mathbf{X} \sim N(0, \mathbf{\Sigma})$ where $\mathbf{\Sigma} = UDU^T$.
4. (Translation) $UD^{1/2}\mathbf{X} + \boldsymbol{\mu} \sim N(\boldsymbol{\mu}, \mathbf{\Sigma})$

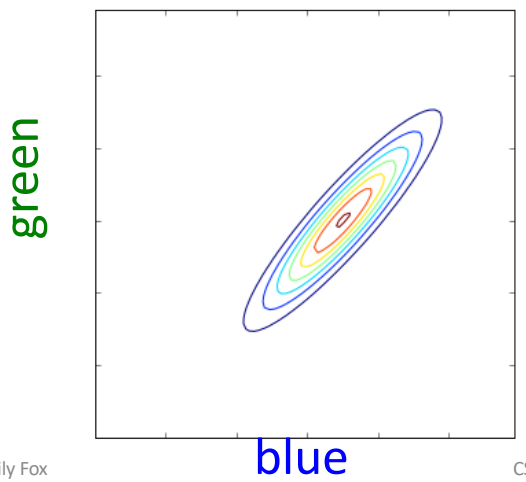
Multivariate Gaussians

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

3D mesh plot



Contour plot



Summary

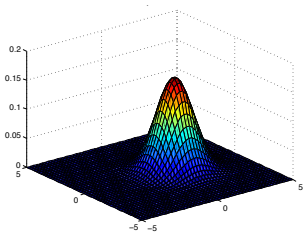
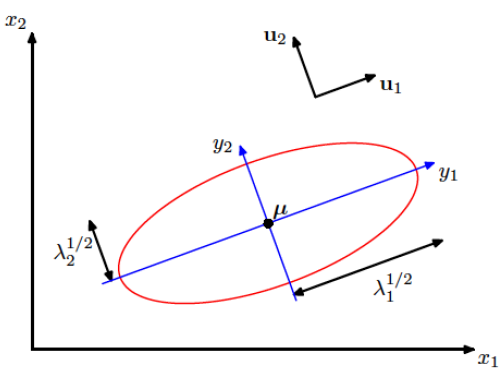
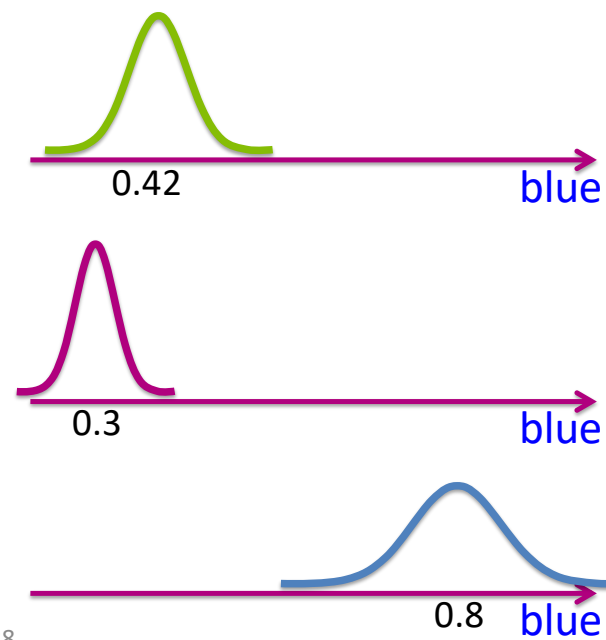


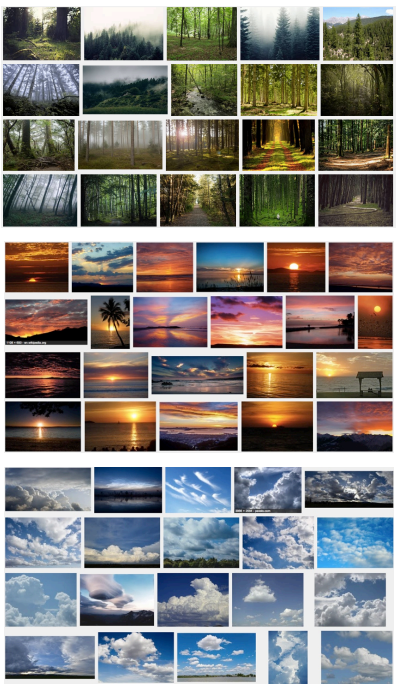
Figure 2.7 The red curve shows the elliptical surface of constant probability density for a Gaussian in a two-dimensional space $\mathbf{x} = (x_1, x_2)$ on which the density is $\exp(-1/2)$ of its value at $\mathbf{x} = \boldsymbol{\mu}$. The major axes of the ellipse are defined by the eigenvectors $\mathbf{u}_i$ of the covariance matrix, with corresponding eigenvalues λ_i.



Mixture Model (to be used for clustering)

Model as Gaussian per category/cluster



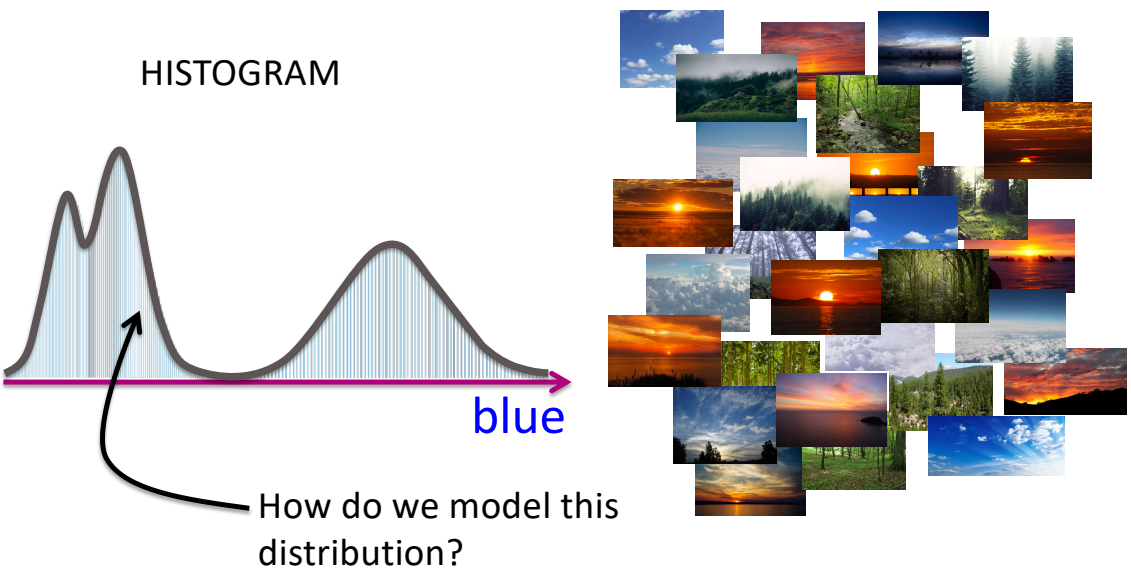


Forests

Sunsets

Clouds

Jumble of unlabeled images



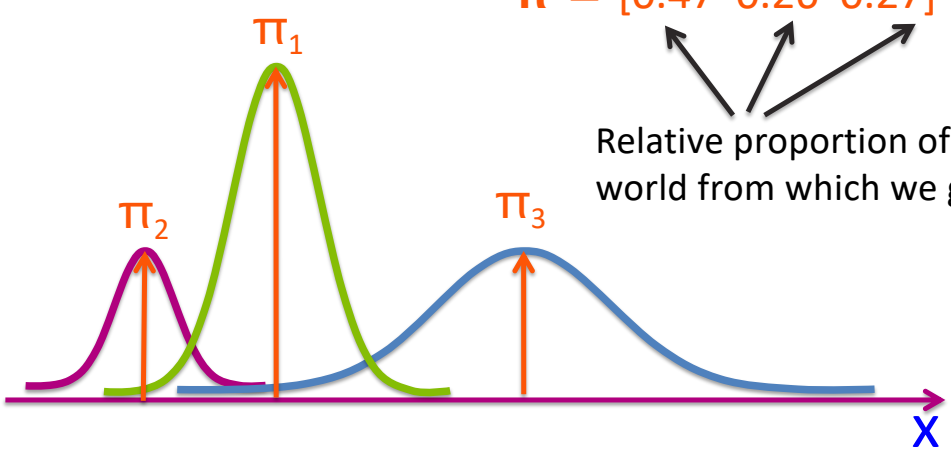
Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

π_1 π_2 π_3

$\boldsymbol{\pi} = [0.47\ \ 0.26\ \ 0.27]$

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



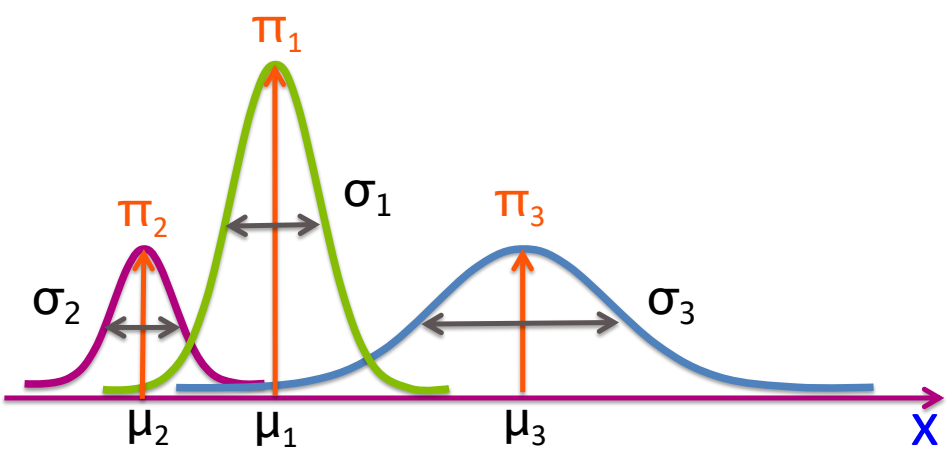
$$0 \le \pi_k \le 1$$

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

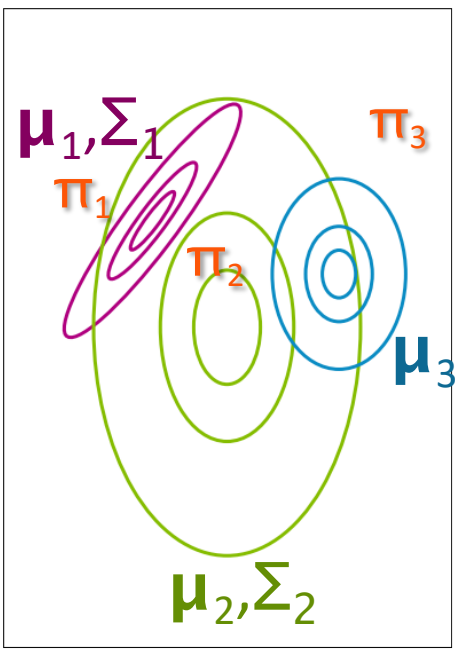
This is how we assume our data was generated. There are k classes that each make up some proportion pi_i of all the data points. We randomly choose one of these k classes based on these proportions/probabilities, and sample a single point from that distribution. WE DONT KNOW THESE DISTRIBUTIONS; use data to try to infer their parameters

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:

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

Mixture model

- K clusters, defined by the following parameters

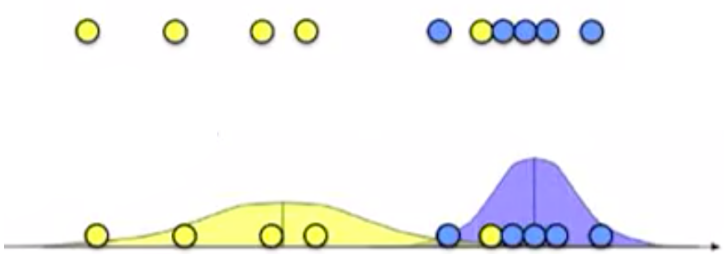
$$\Theta = \{\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{K} \qquad \sum_{j=1}^{K} \pi_j = 1.$$

- Problem: Assume that the data comes from such a distribution, and recover the parameters of the distribution.
- Determine, for each point, the likelihood of it belonging to cluster j, for each j.

K=2 1-D Gaussians, with unknown mean and variance

- Easy if know the source of each data point.

Could approximate the mean and varaince for each of the k classes and compute the parameters from there.



- What if we don't know the source?

Images dues to Victor Lavrenko

To understand better, introduce additional "latent" variables

- K clusters, defined by the following parameters

$$\Theta = \{\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{k} \qquad \sum_{j=1}^{k} \pi_j = 1.$$

one hot encoding of the classes

- For each point $\mathbf{x}$, let $\mathbf{z} = (z_1, \ldots, z_K)$

indicate which cluster it was chosen from. These are called "latent variables".

$$p(z_k = 1) = \pi_k \qquad p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

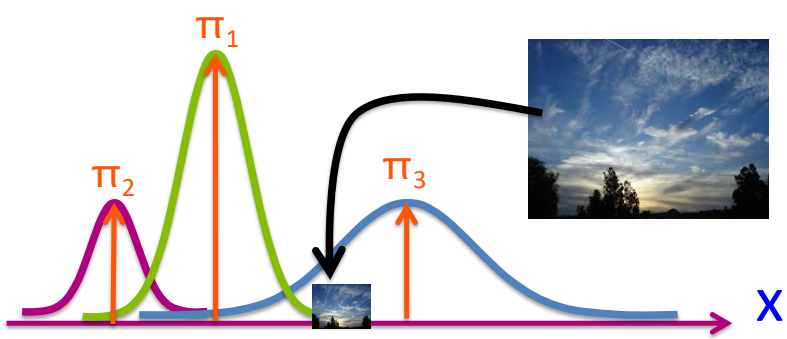
With z's in hand, we can compute many relevant quantities

- Conditional distribution of **x** given **z**

- Therefore:

- Conditional probability

of **z** given **x**



Mixture model cont.

- Conditional distribution of $\mathbf{x}$

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Therefore:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- We will also be interested in the conditional probability of **z** given **x**

$$\begin{aligned} p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \end{aligned}$$

And we can now try to calculate MLE

- Given dataset from a mixture model, find parameters

$$\Theta = \{\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{k}$$

that maximize loglikelihood.

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

MLE

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Gradient = 0 gives the following conditions:

Probability that datapoint x_i comes from the kth cluster

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik}\, \mathbf{x}_i$$

$$p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

expected number of points in cluster k

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

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik}\, (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

$$\pi_k = \frac{N_k}{N}$$

N = total number of points in dataset

Does not give us a closed form ☹

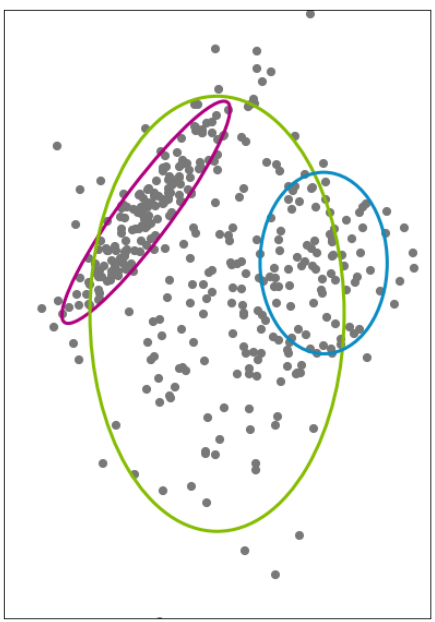
Expectation Maximization Algorithm

Algorithm for solving the Gaussian Mixture Model

Two step approach based on following observation

- If we knew the $\mathbf{z}_i$'s, we could estimate all the parameters.
- If we knew all the parameters we could estimate the $\mathbf{z}_i$'s (or more precisely, the chance each point came from each cluster)
- EM is an iterative algorithm that alternates between these two steps.

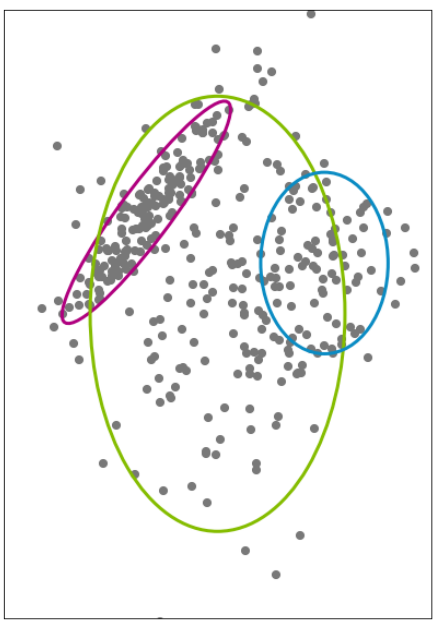
E step: estimate responsibilities



Using our current best guess of the parameters

Compute $r_{ik} = \Pr(z_{ik} = 1 \mid \mathbf{x}_i, \text{params}\{\pi_k, \boldsymbol{\mu}_k, \Sigma_k\})$ (responsibilities)

Responsibilities in equations



Using whatever estimation of the parameters we currently have

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

probability each point is in each cluster

M step:

Given the soft assignments r_{ij}, estimate parameters

Estimating cluster parameters from soft assignments r_{ij}

Estimating cluster parameters from assignments r_{ij}

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
$x_4[1]$	$x_4[2]$	$x_4[3]$	1
$x_5[1]$	$x_5[2]$	$x_5[3]$	2
$x_6[1]$	$x_6[2]$	$x_6[3]$	2

Suppose that magically r_{ik}'s are all 0, 1, i.e. hard assignments

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

Mean/covariance MLE

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

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i \text{ in } k} x_i$$

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

Cluster proportion MLE

R	G	B	Cluster
$x_4[1]$	$x_4[2]$	$x_4[3]$	1

R	G	B	Cluster
$x_5[1]$	$x_5[2]$	$x_5[3]$	2
$x_6[1]$	$x_6[2]$	$x_6[3]$	2

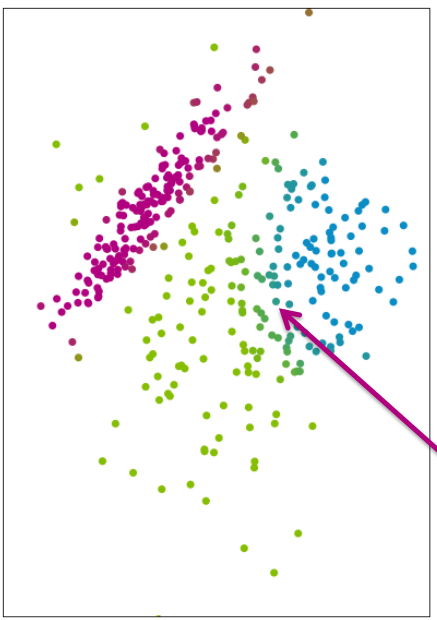
R	G	B	Cluster
$x_1[1]$	$x_1[2]$	$x_1[3]$	3
$x_2[1]$	$x_2[2]$	$x_2[3]$	3
$x_3[1]$	$x_3[2]$	$x_3[3]$	3

obs in cluster k

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

total # of obs

Estimating cluster parameters from soft assignments



Instead of having a full observation x_i in cluster k, just allocate a portion r_{ik}

x_i divided across all clusters, as determined by r_{ik}

Maximum likelihood estimation from soft assignments

R	G	B	r_{i1}	r_{i2}	r_{i3}
$x_1[1]$	$x_1[2]$	$x_1[3]$	0.30	0.18	0.52
$x_2[1]$	$x_2[2]$	$x_2[3]$	0.01	0.26	0.73
$x_3[1]$	$x_3[2]$	$x_3[3]$	0.002	0.008	0.99
$x_4[1]$	$x_4[2]$	$x_4[3]$	0.75	0.10	0.15
$x_5[1]$	$x_5[2]$	$x_5[3]$	0.05	0.93	0.02
$x_6[1]$	$x_6[2]$	$x_6[3]$	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

expected number of points in BLUE cluster

Maximum likelihood estimation from soft assignments

Use these probabilities to update parameters of each class
- estimate mean (weighted by prob point is in the cluster)
- estimate variance (weighted by prob point is in cluster)

R	G	B	Cluster 1 weights
$x_1[1]$	$x_1[2]$	$x_1[3]$	0.30
$x_2[1]$			
$x_3[1]$			
$x_4[1]$			
$x_5[1]$			
$x_6[1]$			

R	G	B	Cluster 2 weights
$x_1[1]$	$x_1[2]$	$x_1[3]$	0.18
$x_2[1]$			
$x_3[1]$			
$x_4[1]$			
$x_5[1]$			
$x_6[1]$			

R	G	B	Cluster 3 weights
$x_1[1]$	$x_1[2]$	$x_1[3]$	0.52
$x_2[1]$	$x_2[2]$	$x_2[3]$	0.73
$x_3[1]$	$x_3[2]$	$x_3[3]$	0.99
$x_4[1]$	$x_4[2]$	$x_4[3]$	0.15
$x_5[1]$	$x_5[2]$	$x_5[3]$	0.02
$x_6[1]$	$x_6[2]$	$x_6[3]$	0.01

Cluster-specific location/shape MLE

R	G	B	Cluster 1 weights
$x_1[1]$	$x_1[2]$	$x_1[3]$	0.30
$x_2[1]$	$x_2[2]$	$x_2[3]$	0.01
$x_3[1]$	$x_3[2]$	$x_3[3]$	0.002
$x_4[1]$	$x_4[2]$	$x_4[3]$	0.75
$x_5[1]$	$x_5[2]$	$x_5[3]$	0.05
$x_6[1]$	$x_6[2]$	$x_6[3]$	0.13

1.242

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

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

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

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

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

M step summary

Compute cluster parameter estimates from soft assignments

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

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

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

Expectation maximization (EM)

Expectation maximization (EM): An iterative algorithm

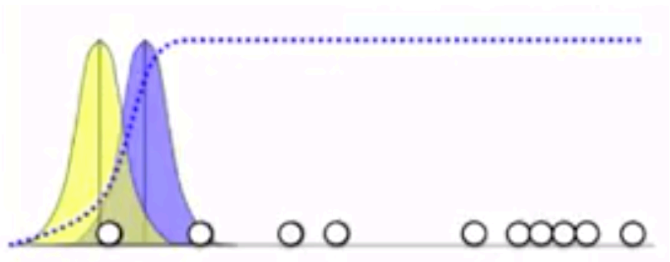
Motivates an iterative algorithm:

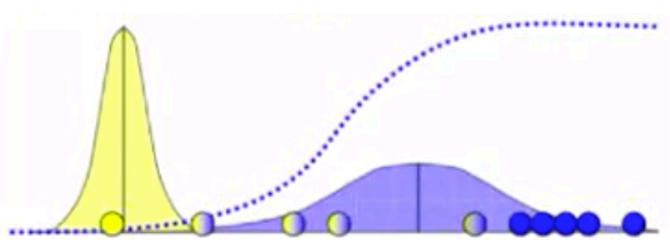
1. E-step: estimate cluster responsibilities given current parameter estimates

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

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

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

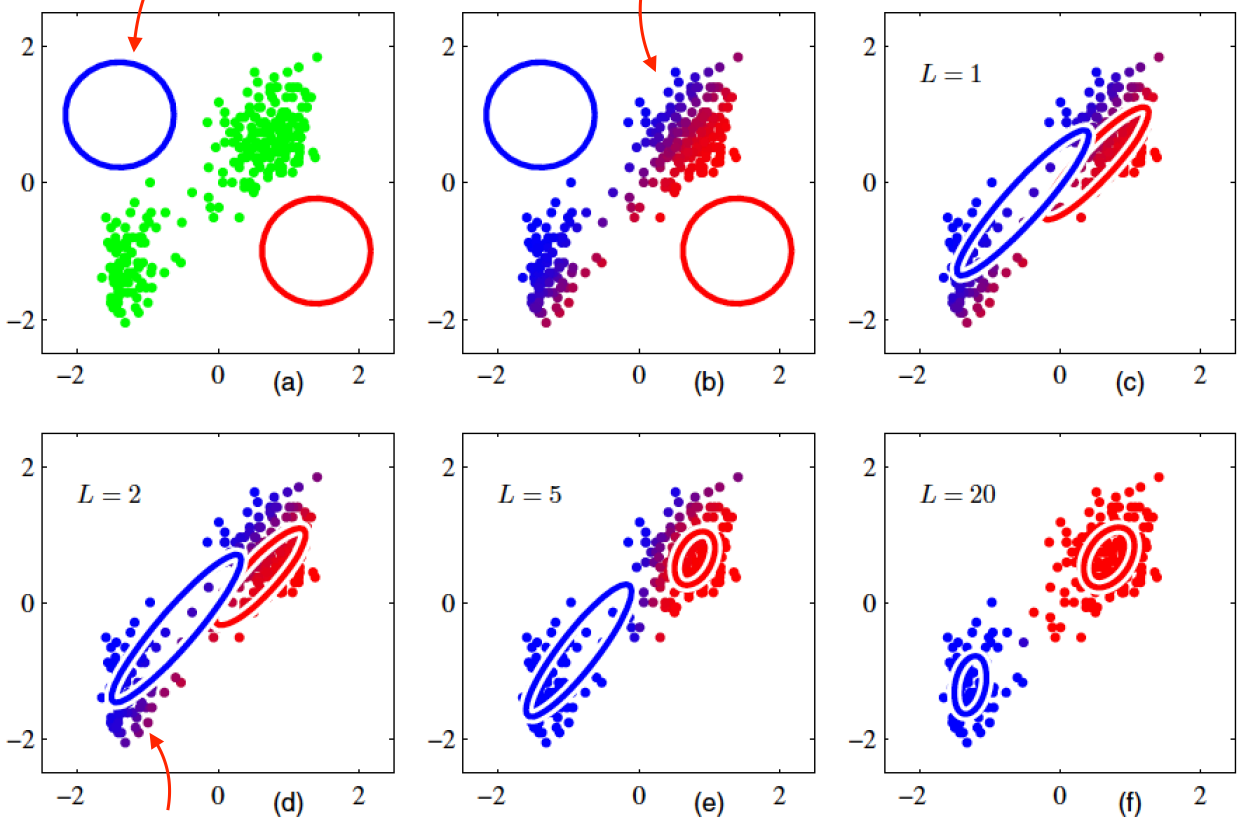




initial BLUE distribution

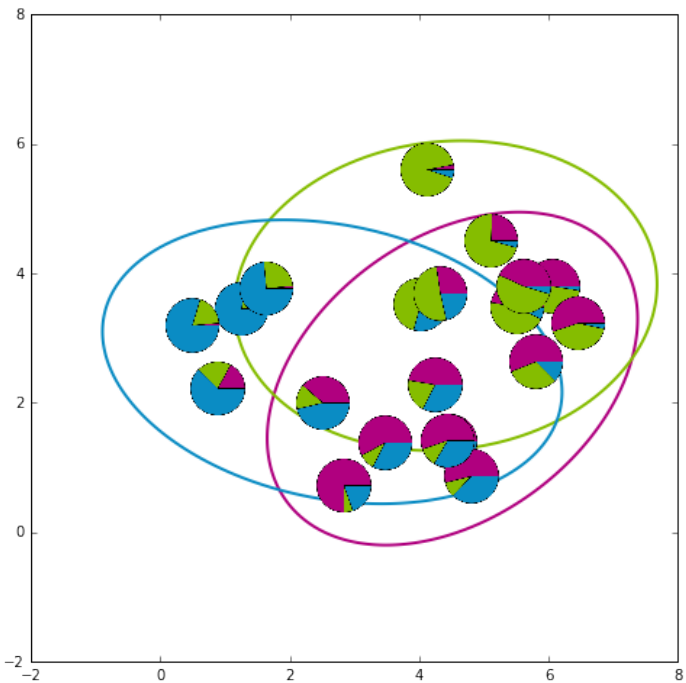
probability each datapoint is in either the blue or red cluster

adjust predictions of parameters/distributions

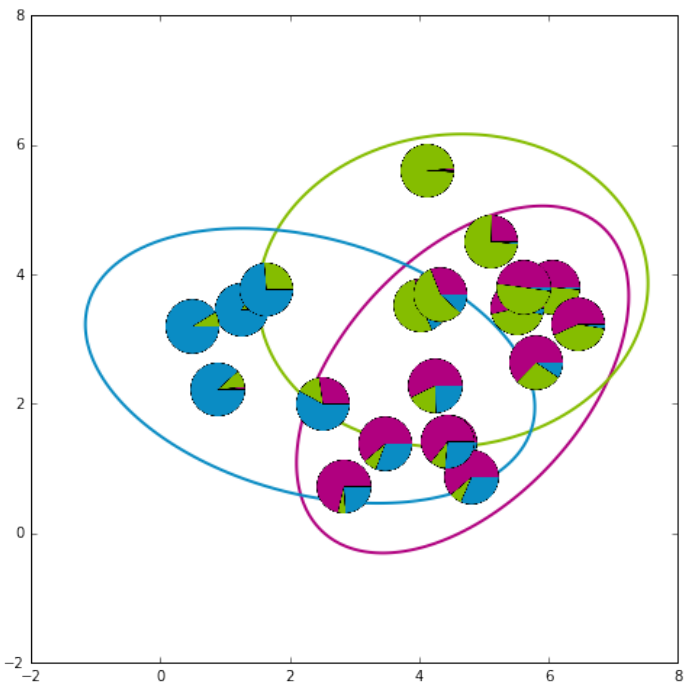


probability each datapoint is in either the blue or red cluster

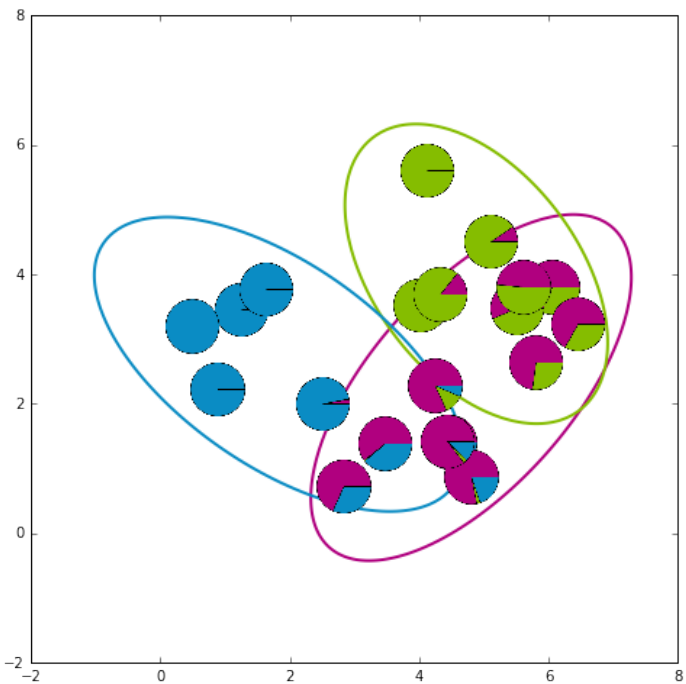
EM for mixtures of Gaussians in pictures – initialization



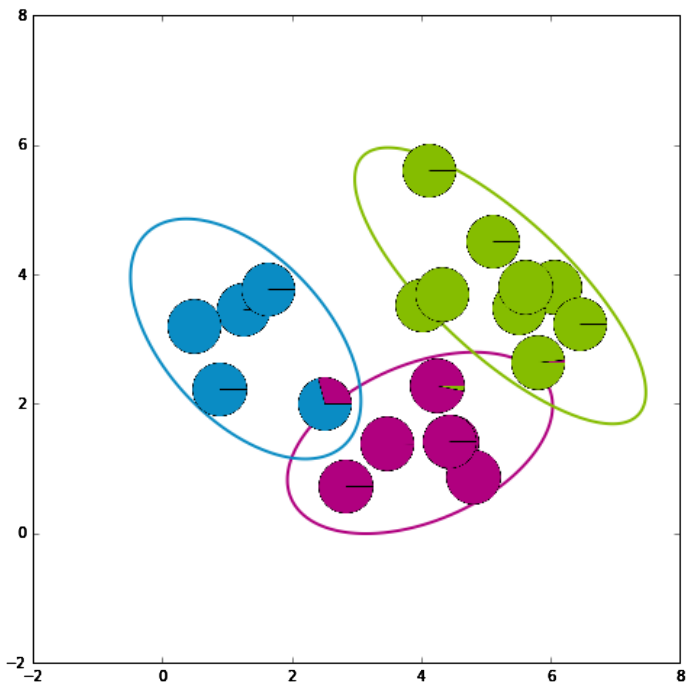
EM for mixtures of Gaussians in pictures – after 1st iteration



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



EM for mixtures of Gaussians in pictures – converged solution



The nitty gritty of EM

Convergence and initialization of EM

Convergence of EM

Not guaranteed to end up at a optimal solution; could get stuck in a local maximum

- EM is a coordinate-ascent algorithm
  - Can equate E-and M-steps with alternating maximizations of the objective function
- Convergence to a local maximum.

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

For convergence, check whether the distributions are moving

Initialization

success of algorithm dependent on initial conditions; possibly use K means to determine initial locations of distributions

- Many ways to initialize the EM algorithm
- Important for convergence rates & quality of local maximum found
- Examples:
  - Choose K observations at random to define K "centroids". Assign other observations to nearest centroid to form initial parameter estimates.
  - Initialize from k-means solution

Potential of vanilla EM to overfit

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 ∞ !

Simple regularization of M-step for mixtures of Gaussians

Simple fix: Don't let variances → 0!

Add small amount to diagonal of covariance estimate

Summary

What you can do now…

- Understand Gaussian mixture models (and multivariate Gaussians)
- Estimate soft assignments (responsibilities) given mixture model parameters
- Solve maximum likelihood parameter estimation using soft assignments (weighted data)
- Implement an EM algorithm for inferring soft assignments and cluster parameters
  - Determine an initialization strategy
  - Implement a variant that helps avoid overfitting issues
- Compare and contrast with k-means
  - Soft vs. hard assignments