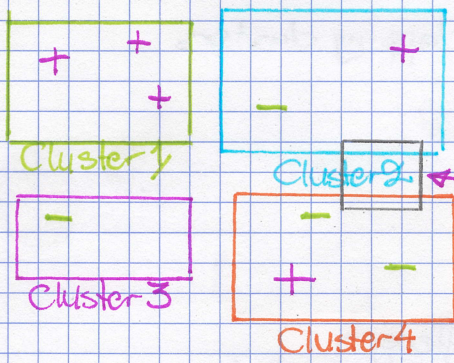
MIXTURE MODELS

Model-Based Clustering

Why a probabilistic approach?

Learn user preferences

Set of clustered documents read by user



Use feedback to learn user preferences over topics

Uncertainty in cluster assignments

Slightly closer to Cluster4 than Cluster2, but count fully for Cluster4?

Hard assignments don't tell full story

Other limitations to k-means

Assign observations to closest cluster center

$$z_i \leftarrow \arg\min_j \|\mu_j - x_i\|_2^2$$

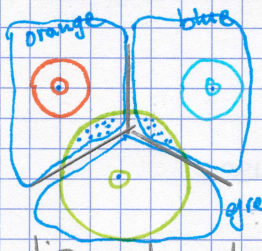
Only center matters

Can use weighted Euclidean, but requires known weights

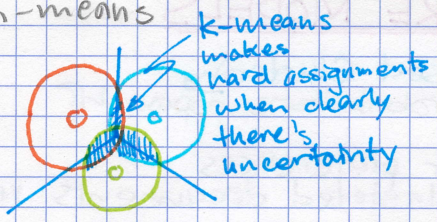
Equivalent to assuming spherically symmetric clusters

still assumes all clusters have the same axis-aligned ellipses

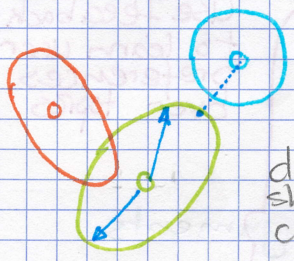
Failure modes of k-means



disparate cluster sizes



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

Aggregating over unknown classes in an image database

Simple image representation

Consider average red, green, blue pixel intensities

Image #1

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

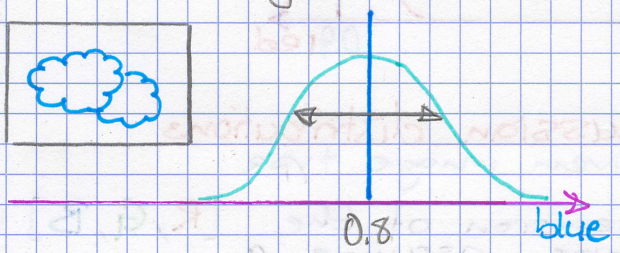
Image #2

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

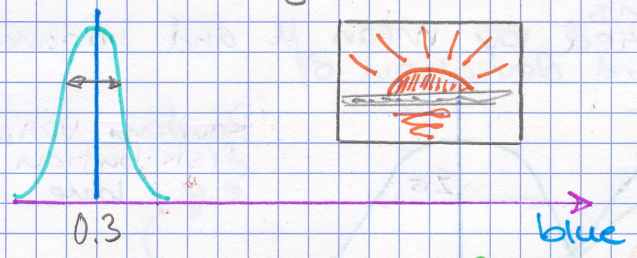
Image #3

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

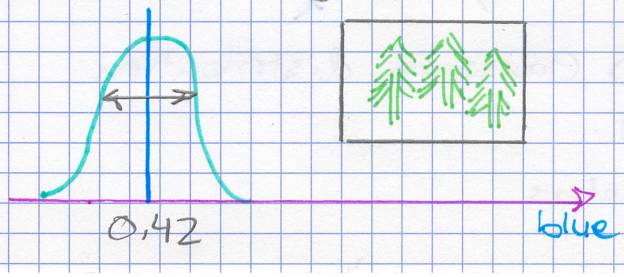
Distribution over all cloud images
Let's look at just the blue dimension



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

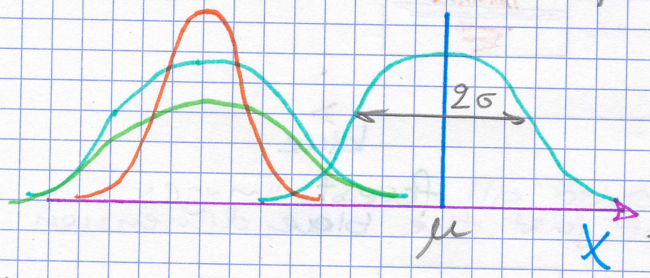
Can be distinguished along other dim
Now look at the red dimension

Univariate Gaussian distributions

Model for a given image type

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

1-D Gaussians
Fully specified by mean μ and variance σ^2 (or standard deviation σ)



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

Notating a 1-D Gaussian distribution

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

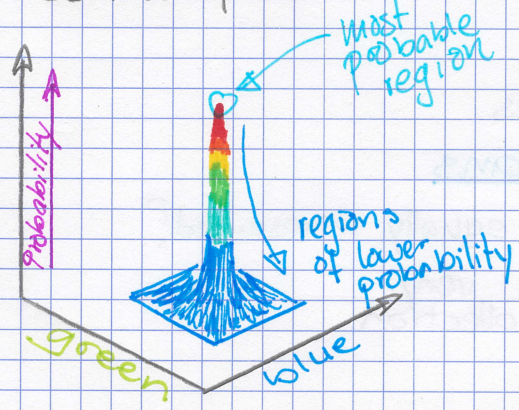
Random variable the distribution is over e.g., blue intensity

parameters

Bivariate and multivariate Gaussians

2D Gaussians

3D mesh plot



2D Gaussians — Parameters

Fully specified by mean μ and covariance Σ

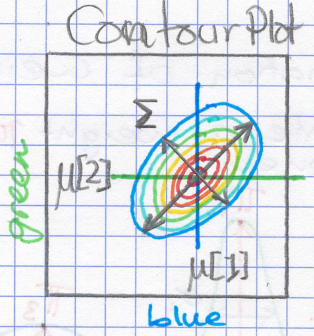
$$\mu = [\mu_{blue}, \mu_{green}]$$

mean centers the distribution in 2D

$$\Sigma = \begin{bmatrix} \sigma^2_{blue} & \sigma_{blue,green} \\ \sigma_{green,blue} & \sigma^2_{green} \end{bmatrix}$$

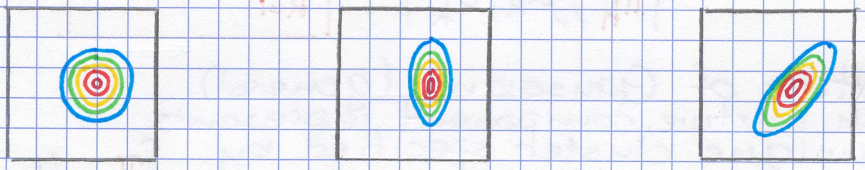
covariance determines orientation + spread

Contour Plot



Covariance structures

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma^2_B & 0 \\ 0 & \sigma^2_G \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma^2_B & \sigma_{B,G} \\ \sigma_{G,B} & \sigma^2_G \end{bmatrix}$$



Notating a multivariate Gaussian

$N(x \mid \mu, \Sigma)$

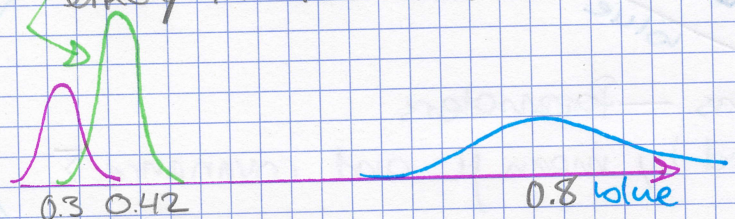
parameters

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

Mixture of Gaussians

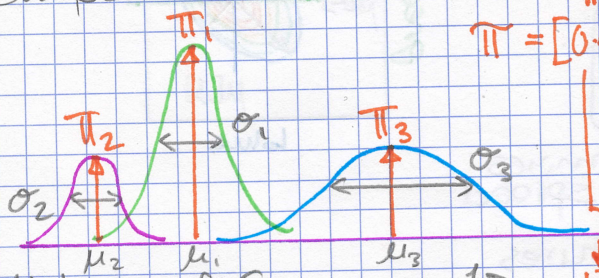
What if image types not equally represented?

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



Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component



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

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

$0 \leq \pi_k \leq 1$

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

- Mixture of Gaussians 1D
  Each mixture component represents a unique cluster by:

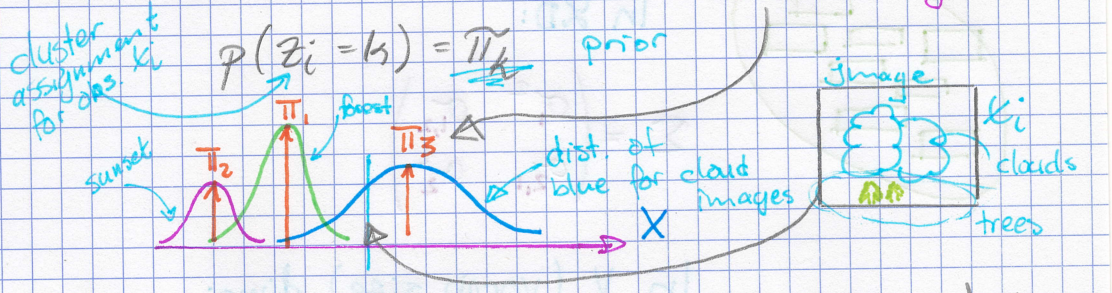
  $\{\pi_k, \mu_k, \sigma_k^2\}$

- Mixture of Gaussians (general)
  Each mixture component represents a unique cluster specified by: $\{\pi_k, \mu_k, \Sigma_k\}$

Interpreting the mixture of Gaussian terms

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)



Given observation x_i is from cluster k, what's the likelihood of seeing x_i? (e.g., just look at distribution for "clouds")

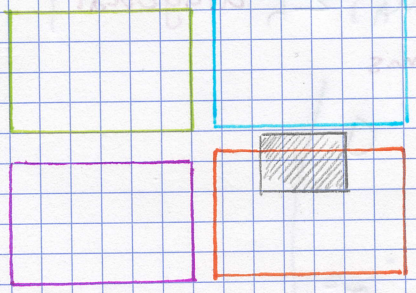
$$p(x_i \mid z_i = k, \mu_k, \Sigma_k) = \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$$

likelihood

$[RGB]_i$

Scaling mixtures of Gaussians for document clustering

Discover groups of related documents

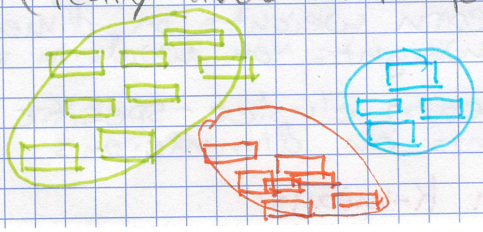


Document representation

$x_i =$ [| | | tf-idf vector | |]

Mixture of Gaussians for clustering documents

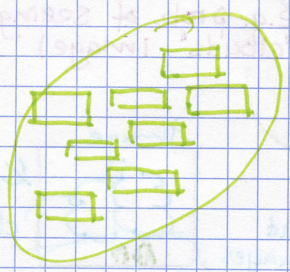
Space of all documents (really lives in $\mathbb{R}^V$ for vocab size V)



Make soft assignments of docs to each Gaussian

Counting parameters

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



In 2D:

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

In V (vocab size) dims:

$$\Sigma = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & & \cdot \\ \cdot & & \cdot & & & & & & \cdot \\ \cdot & & & \frac{V(V+1)}{2} & & & & & \cdot \\ \cdot & & & & \cdot & & & & \cdot \\ \cdot & & & & & \cdot & & & \cdot \\ \cdot & & & & & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Restricting to diagonal covariance

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

V params

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

Restrictive assumption, but...

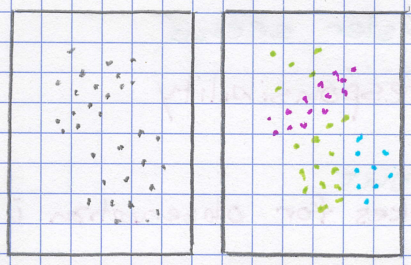
- Can learn weights on dimensions (e.g., weights on words in vocab)
- Can learn cluster-specific weights on dimensions

Still more flexible than K-means

Computing soft assignments from known cluster parameters

Inferring cluster labels

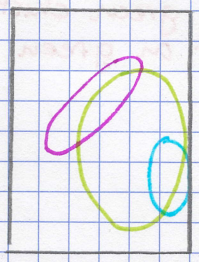
Data | Desired soft assignments



Part 1

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

Compute responsibilities



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

Responsibility cluster k takes for observation i

fixed values defining the distribution

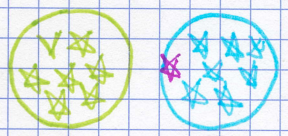
$$r_{ik} = p(z_i = k \mid \{\pi_j, \mu_j, \Sigma_j\}_{j=1}^{K}, x_i)$$

("given")

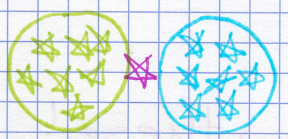
- r_{ik}: random variable
- $p(z_i = k \mid \ldots)$: probability of assignment to cluster k
- $\{\pi_j, \mu_j, \Sigma_j\}_{j=1}^{K}$: given model parameters
- x_i: observed value

Responsibilities in pictures

Green cluster takes more responsibility



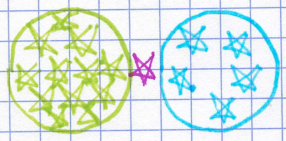
Blue cluster takes more responsibility



Uncertain... split responsibility

Responsibilities in pictures (cont'd)

Need to weight by cluster probabilities, not just cluster shapes



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

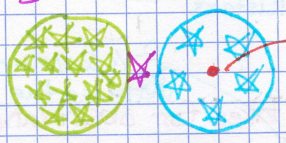
Responsibilities in equations

Responsibility cluster k takes for observation i

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

Initial probability of being from cluster k

How likely is the observed value x_i under this cluster assignment?



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

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

Normalized over all possible cluster assignments

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!

Estimating cluster parameters from known cluster assignments

Part 2a

Imagine we knew the cluster (hard) assignments z_i

Data table decoupling over clusters

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

Maximum likelihood estimation

Sum these vectors

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

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

divide by 3 (the total # of obs.)

maximum likelihood estimation (MLE)

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

$\hat{\mu}_k$ denotes "estimate"

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

N_k = # of obs. in cluster

average data points in cluster k

Mean / covariance MLE

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

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

Cluster proportion MLE

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!

~~Estimating cluster assignment parameters from soft assignments~~

Part 2b:

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

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

Maximum likelihood estimation from soft assignments

Just like M boosting with weighted observations

	R	G	B	r_{i1}	r_{i2}	r_{i3}
x_1	$x_1[1]$	$x_1[2]$	$x_1[3]$	0.30	0.18	0.52
x_2	$x_2[1]$	$x_2[2]$	$x_2[3]$	0.01	0.26	0.73
x_3	$x_3[1]$	$x_3[2]$	$x_3[3]$	0.002	0.008	0.99
x_4	$x_4[1]$	$x_4[2]$	$x_4[3]$	0.75	0.10	0.15
x_5	$x_5[1]$	$x_5[2]$	$x_5[3]$	0.05	0.93	0.02
x_6	$x_6[1]$	$x_6[2]$	$x_6[3]$	0.13	0.86	0.01
Total weight in cluster (effective # of obs.)				1.242	2.8	2.42

52% chance this obs is in cluster 3

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

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

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

Compute cluster parameter estimates with weights on each row operation

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

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	

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

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

Total weight in cluster k = effective # obs.

Estimate cluster proportions from relative weights

6 ↔ # datapoints N

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

hard assignments have:

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

Equating the estimates...

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

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

$= N_k$

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

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

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

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

EM iterates in equations and pictures

Expectation maximization (EM)

An iterative algorithm

Motivates an iterative algorithm:

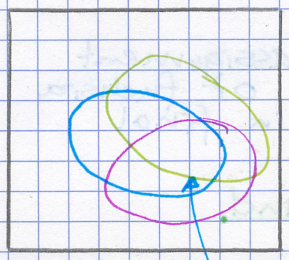
1. E-step: estimate cluster responsibilities given current parameter estimates

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

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

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

Then compute

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

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

fuchsia, blue, green

— after 1st iteration

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

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

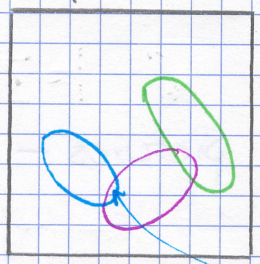
Then recompute responsibilities

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

— after 2nd iteration

rinse + repeat until convergence

EM for mixtures of Gaussians in pictures (cont'd) — converged solution



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

Convergence, initialization, and overfitting of EM

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
- Convergence to a local mode
- We will assess via (log) likelihood of data under current parameter and responsibility estimates

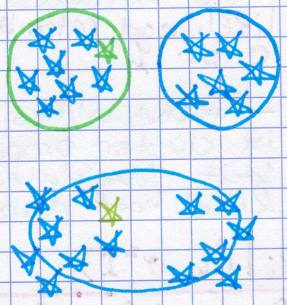
Initialization

- 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 examples:
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 $\mu_{k}[w] = x_i[w]$ 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

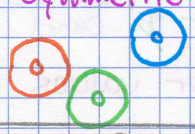
Add small amount to diagonal of covariance estimate

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

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



- Spherical clusters with equal variances, so relative likelihoods just function of distance to cluster center

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

- As variances $\to 0$, likelihood ratio becomes 0 or 1

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

- 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

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

Infinitesimally small

Decision based on distance to nearest cluster center

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

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