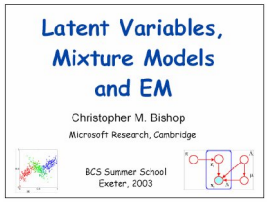
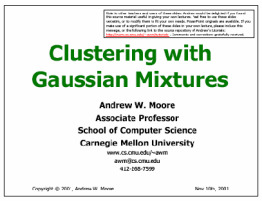
Gaussian Mixture Models (GMM) and the K-Means Algorithm

Source Material for Lecture

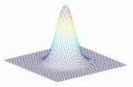


http://research.microsoft.com/~cmbishop/talks.htm



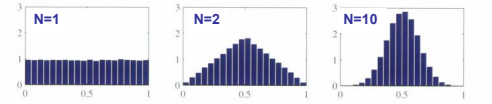
http://www.autonlab.org/tutorials/gmm.html

Gaussian Distribution aka Multivariate Normal Distribution.



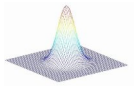
Very important in math and science due to the Central Limit Theorem: the distribution of the sum/mean of a set of iid random variables tends towards Gaussian as the number N of variables increases.

Example: sample mean of a set of N iid uniform(0,1) random variables:



Example: binomial distribution as a function of m (number of heads) of N iid binary Bernoulli trials becomes more-and-more Gaussian-like for large N

Gaussian Distribution aka Multivariate Normal Distribution.



1 dimensional case

mean

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

variance

D dimensional case

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

mean vector

covariance matrix

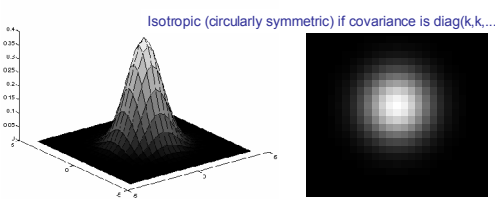
Review: The Gaussian Distribution

- Multivariate Gaussian

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

mean covariance

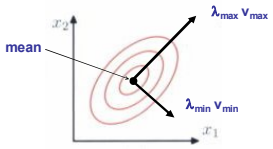
Isotropic (circularly symmetric) if covariance is diag(k,k,...,k)



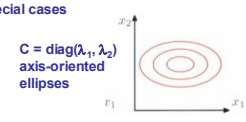
Gaussian

consider 2D case: constant-probability curves are ellipses

* centered at mean location u = (x0,y0)
* oriented and elongated according to eigenvalues λ and unit eigenvectors v of 2x2 symmetric pos.def. covariance matrix C



special cases

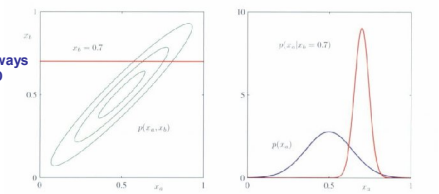


Gaussian

Gaussians are cool. Among other things, they have some amazing "self-replicating" properties (my word, not Bishop's)

For example, all marginals of a Gaussian are Gaussian.
Also, all conditionals of a Gaussian are Gaussian.
The combination of a Gaussian prior and a Gaussian likelihood using Bayes rule yields a Gaussian posterior.
And, for what it's worth (cause it is not in this chapter):
The sum of Gaussian random variables is Gaussian
Affine transforms of Gaussian r.v.s yield Gaussian r.v.s

examples: two ways of reducing a 2D Gaussian to 1D



blue curve : marginal (integrate down columns)
red curve : conditional (take values along red line and renormalize)

Likelihood Function

- Data set

$$D = \{\mathbf{x}_n\} \quad n = 1, \ldots, N$$

- Assume observed data points generated independently

$$p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Viewed as a function of the parameters, this is known as the *likelihood function*

Maximum Likelihood

- Set the parameters by maximizing the likelihood function
- Equivalently maximize the log likelihood

$$\ln p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

Maximum Likelihood Solution

- Maximizing w.r.t. the mean gives the *sample mean*

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

- Maximizing w.r.t covariance gives the *sample covariance*

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathsf{T}}$$

WILL DERIVE THIS ON THE BOARD FOR 1D CASE

Tasks 1-3 of incremental homework

1. Given a d-dimensional mean vector v and dxd covariance matrix C (symmetric, pos def), generate N random sample points distributed according to a Gaussian with mean v and covariance C. Recall that we discussed in class how to do this by first using randn in matlab to generate variables x with zero mean and covariance I, and then doing a linear transformation y=Ax + b to transform them into new variables y with the desired mean and covariance.

2. For a set of 2D points generated as above, make a 2D plot of them in a graphics display, and then draw overlaid on that an elliptical contour associated with a Gaussian of mean v and covariance C to verify that your transformed points form a cluster with the correct center location and elliptical spread. We discussed doing this by first generating points on a 2D circle with radius 1.5 and then doing a linear transformation Ax+b to transform those points into new points lying on an elliptical boundary.

Note: as a sanity check, try all three cases we talked about in class. That is, try to generate and draw data/contours from: 1) circular Gaussian, 2) elliptical Gaussian with major axis lying along either the x or y coordinate axis, and 3) rotated elliptical Gaussian (major axis is oriented diagonally).

Just added!

3. Given a set of N sample points from a Gaussian distribution with d-dimensional mean vector v and dxd covariance matrix C, compute the maximum likelihood estimates for v and C using the formulas from our class notes. If you generate the sample points using the routine in part 1 and some given mean v0 and covariance C0, and you then estimate the sample mean v and C using the maximum likelihood routine, you might expect v to be close to v0 (distance-wise), and C to be similar to C0 (similar axes and eigenvalues).

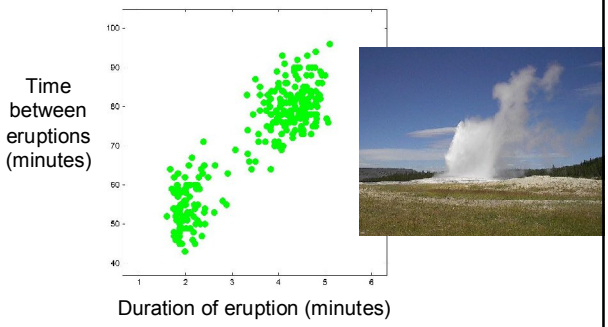
Comments

Gaussians are well understood and easy to estimate

However, they are unimodal, thus cannot be used to represent inherently multimodal datasets

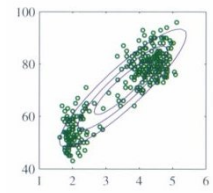
Fitting a single Gaussian to a multimodal dataset is likely to give a mean value in an area with low probability, and to overestimate the covariance.

Old Faithful Data Set



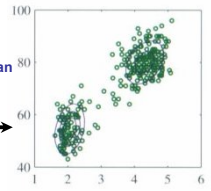
Multi-Modal Data

Major problem with the Gaussian is that it can only describe a distribution with one mode (one "bump")

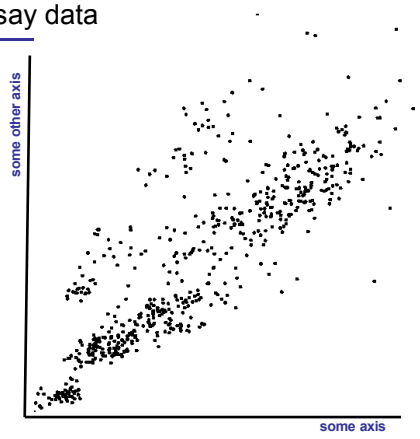


Bad description of bimodal data using the inherently unimodal Gaussian

However, if we are willing to use more than one Gaussian, we can fit one to each mode or cluster of the data.



Some Bio Assay data



Sometimes it may not even be easy to tell how many "bumps" there are…

Idea: Use a Mixture of Gaussians

- Linear super-position of Gaussians

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

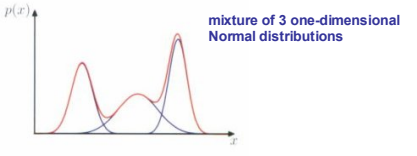
- Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

- Can interpret the mixing coefficients as prior probabilities

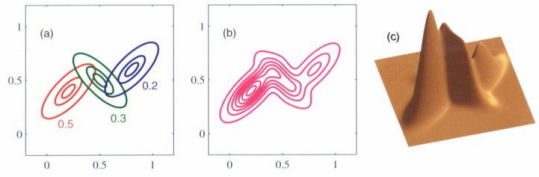
$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k)$$

Mixture of Gaussians

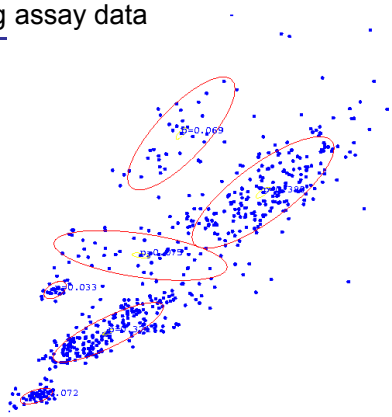


mixture of 3 one-dimensional Normal distributions

mixture of 3 two-dimensional Gaussians



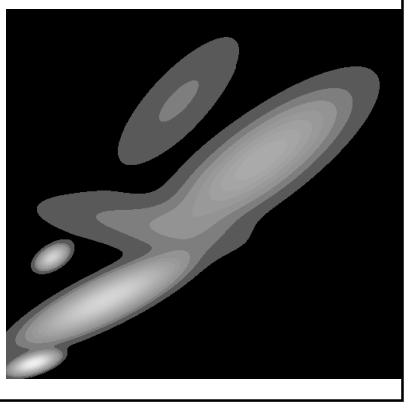
GMM describing assay data



GMM density function

Note: now we have a continuous estimate of the density, so can estimate a value at any point.

Also, could draw constant-probability contours if we wanted to.



Sampling from a Gaussian Mixture

What is the underlying process?

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

procedure to generate a mixture of gaussians

for i=1:N

generate a uniform U(0,1) random number to determine which of K components to draw a sample from (based on probabilities pi_k

generate a sample from a Gaussian N(mu_k, Sigma_k)

end

Sampling from a Gaussian Mixture

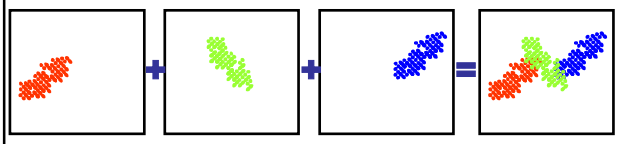
equivalent procedure to generate a mixture of gaussians:

for k=1:K

compute number of samples n_k = round(N * pi_k) to draw from the k-th component Gaussian

generate n_k samples from Gaussian N(mu_k, Sigma_k)

end



Task 4 of incremental homework

4. Given the specification of a mixture of Gaussians distribution with K Gaussian components, generate N sample points from that distribution. The values that are given to you are the mixing weights p1, p2, …, pK, the mean vector of each component v1, v2, …, vK, and the covariance matrix of each component C1, C2, …, CK. Recall from our class the generative process for a mixture of Gaussians, which can easily be turned into an algorithm: for n=1 to N, generate sample point xn by first choosing which of the K components it should come from (with the help of a uniform random number generator), and then using the routine from part 1 to generate a sample point from that component.

Fitting the Gaussian Mixture

- We wish to invert this process – given the data set, find the corresponding parameters:
  - mixing coefficients
  - means
  - covariances
- If we knew which component generated each data point, the maximum likelihood solution would involve fitting each component to the corresponding cluster
- Problem: the data set is unlabelled
- We shall refer to the labels as *latent* (= hidden) variables

Maximum Likelihood for the GMM

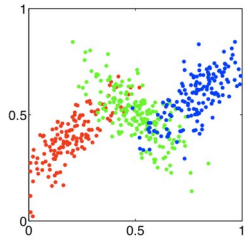
- The log likelihood function takes the form

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

- Note: sum over components appears *inside* the log
- There is no closed form solution for maximum likelihood

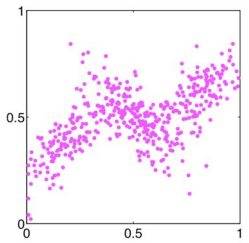
- However, with labeled data, the story is different

Labeled vs Unlabeled Data



labeled

Easy to estimate params
(do each color separately)



unlabeled

Hard to estimate params
(we need to assign colors)

Side-Trip : Clustering using K-means

K-means is a well-known method of clustering data.

Determines location of clusters (cluster centers), as well as which data points are "owned" by which cluster.

Motivation: K-means may give us some insight into how to label data points by which cluster they come from (i.e. determine ownership or membership)

K-means and Hierarchical Clustering

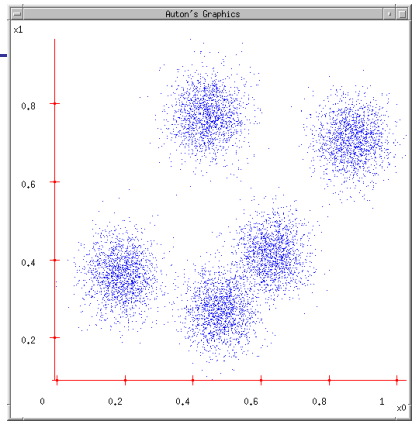
Note to other teachers and users of these slides. Andrew would be delighted if you found this source material useful in giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. PowerPoint originals are available. If you make use of a significant portion of these slides in your own lecture, please include this message, or the following link to the source repository of Andrew's tutorials: http://www.cs.cmu.edu/~awm/tutorials . Comments and corrections gratefully received.

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Some Data

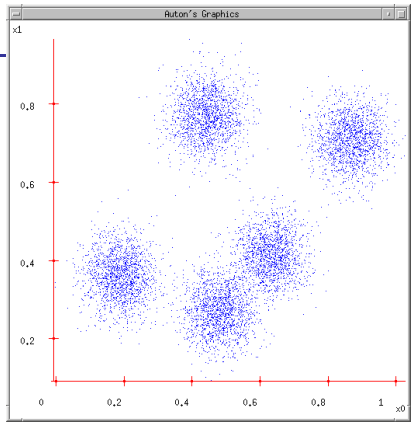
This could easily be modeled by a Gaussian Mixture (with 5 components)

But let's look at an satisfying, friendly and infinitely popular alternative…



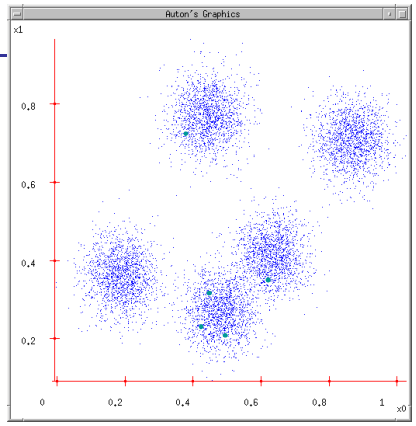
K-means

1. ~~Ask user how many clusters they'd like.~~ *(e.g. k=5)*



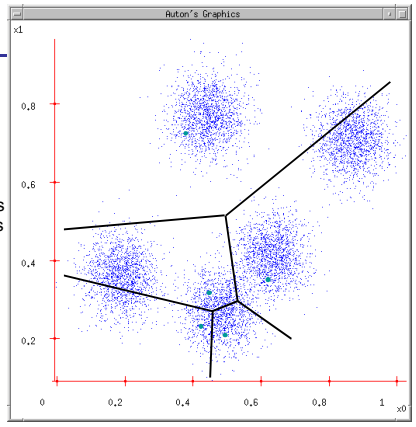
K-means

1. ~~Ask user how many clusters they'd like.~~ *(e.g. k=5)*
2. Randomly guess k cluster Center locations



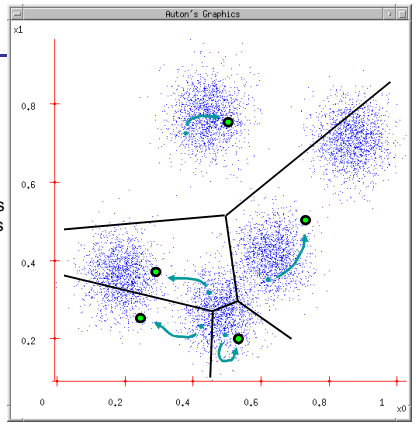
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



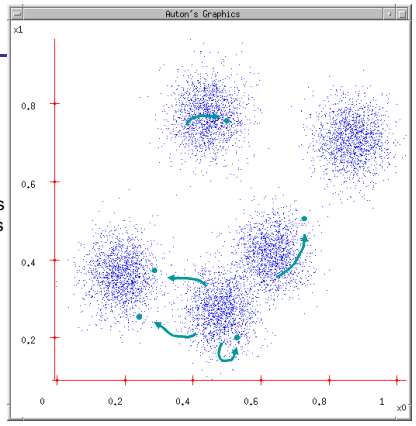
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means

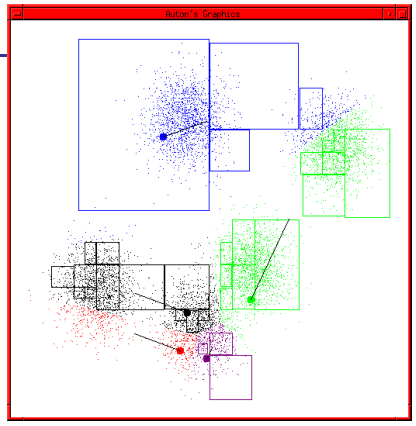
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



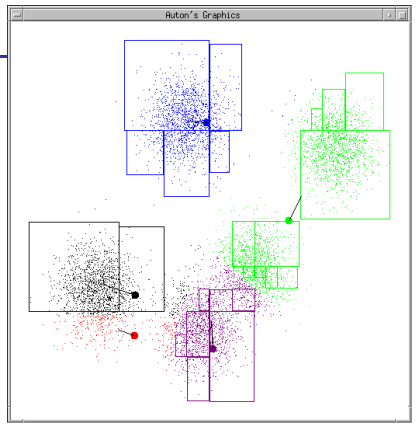
K-means Start

Example generated by Dan Pelleg's super-duper fast K-means system:

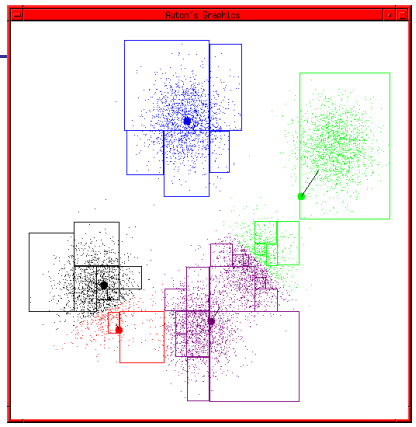
Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999, (KDD99) (available on www.autonlab.org/pap.html*)*



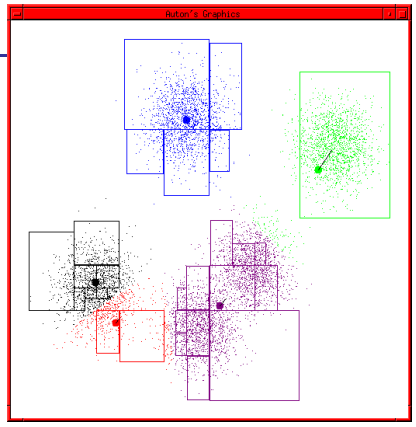
K-means continues…



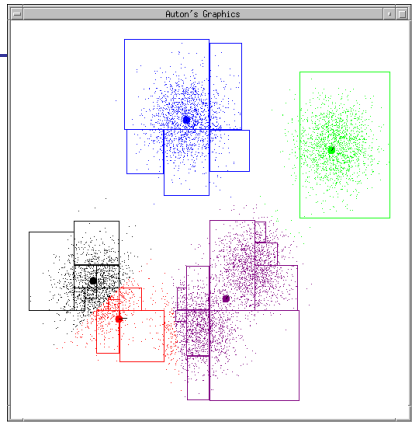
K-means continues…



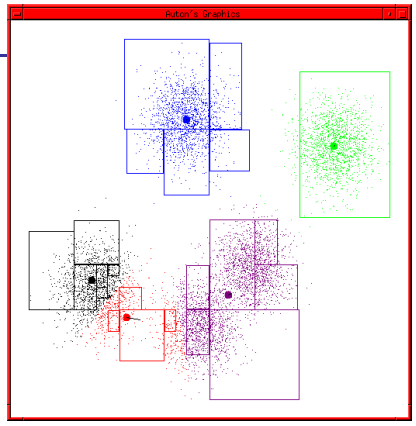
K-means continues…



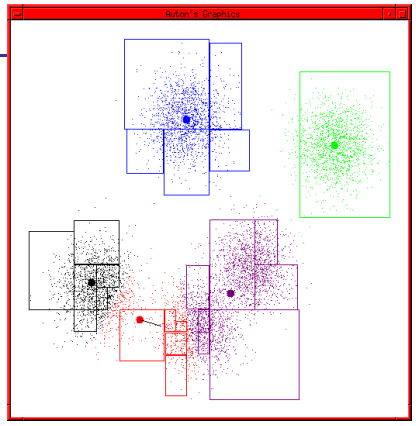
K-means continues…



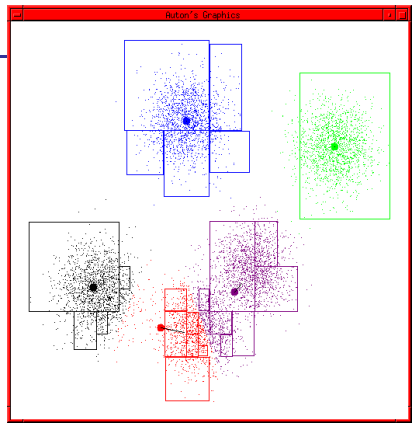
K-means continues…



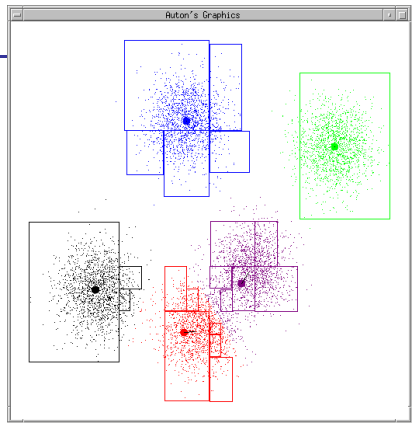
K-means continues…



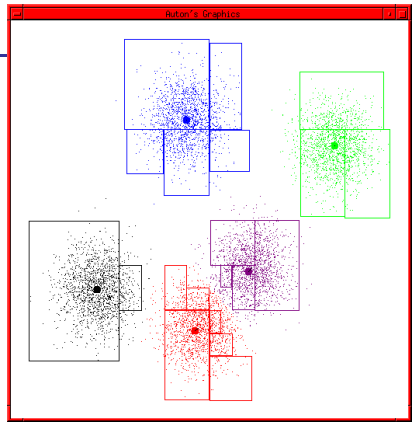
K-means continues…



K-means continues…



K-means terminates



Common uses of K-means

- Often used as an exploratory data analysis tool
- In one-dimension, a good way to quantize real-valued variables into k non-uniform buckets
- Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization)
- Also used for choosing color palettes on old fashioned graphical display devices!
- Used to initialize clusters for the EM algorithm!!!

Comments

We can model and visualize multimodal datasets by using multiple unimodal (Gaussian-like) clusters.

K-means gives us a way of partitioning points into N clusters. Once we know which points go to which cluster, we can estimate a Gaussian mean and covariance for that cluster.

We have introduced the idea of writing what you want to do as a function to be optimized (maximized or minimized). e.g. maximum likelihood estimation to fit parameters of a Gaussian.

Motivation for Next Time

- want to do MLE of mixture of Gaussian parameters
- But this is hard, because of the summation in the mixture of Gaussian equation (can't take the log of a sum).
- If we knew which point contribute to which Gaussian component, the problem would be a lot easier (we can rewrite so that the summation goes away)
- So... let's guess which point goes with which component, and proceed with the estimation.
- We were unlikely to guess right the first time, but based on our initial estimation of parameters, we can now make a better guess at pairing points with components.
- Iterate
- This is the basic idea underlying the EM algorithm.