CSE 332
Introduction to Visualization

Cluster Analysis

Klaus Mueller

Computer Science Department
Stony Brook University and SUNY Korea

Lecture	Topic	Projects
1	Intro, schedule, and logistics	
2	Applications of visual analytics, data types	
3	Data sources and preparation	Project 1 out
4	Data reduction, similarity & distance, data augmentation	
5	Dimension reduction	
6	Introduction to D3	
7	Visual communication using infographics	
8	Visual perception and cognition	Project 2 out
9	Visual design and aesthetic	
10	Cluster analysis	
11	High-dimensional data, dimensionality reduction	
12	Principal component analysis (PCA)	
13	Visualization of spatial data: volume visualization intro	Project 3 out
14	Introduction to GPU programming	
15	Visualization of spatial data: raycasting, transfer functions	
16	Illumination and isosurface rendering	
17	Midterm	
18	Scientific visualization	
19	Non-photorealistic and illustrative rendering	Project 4 out
20	Midterm discussion	
21	Principles of interaction	
22	Visual analytics and the visual sense making process	
23	Visualization of graphs and hierarchies	
24	Visualization of time-varying and streaming data	Project 5 out
25	Maps	
26	Memorable visualizations, visual embellishments	
27	Evaluation and user studies	
28	Narrative visualization, storytelling, data journalism, XAI	

Finding The Needle – Cluster Analysis

Data summarization

- data reduction
- cluster centers, shapes, and statistics

Customer segmentation

- collaborative filtering

Social network analysis

- find similar groups of friends (communities)

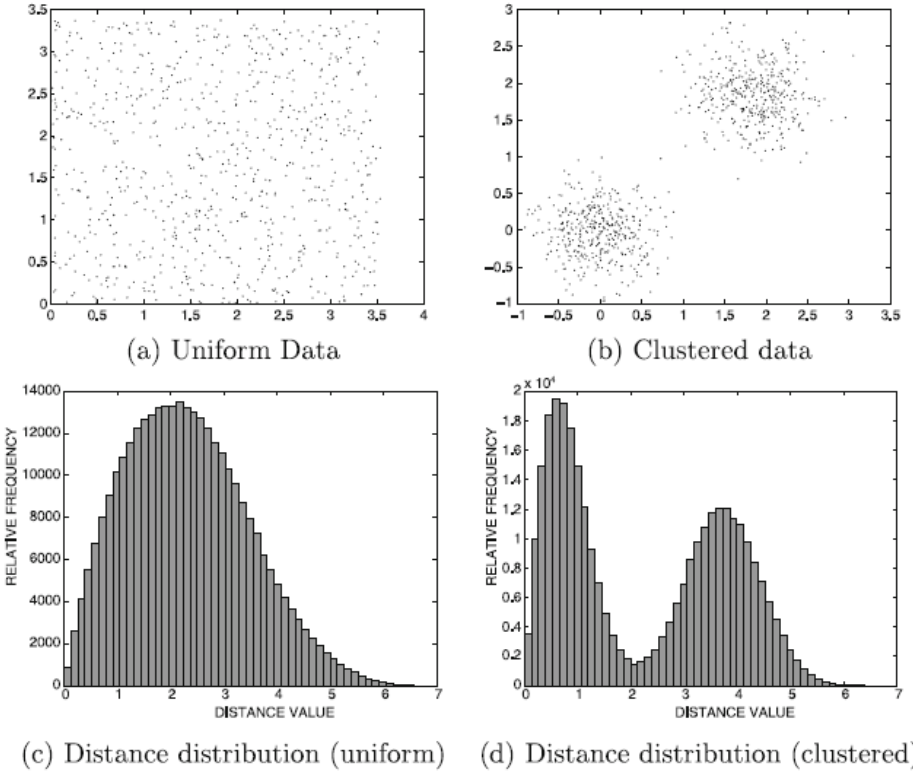
Precursor to other analysis

- use as a preprocessing step for classification and outlier detection

Attribute Selection

With 1,000s of attributes (dimensions) which ones are relevant and which one are not?

avoid

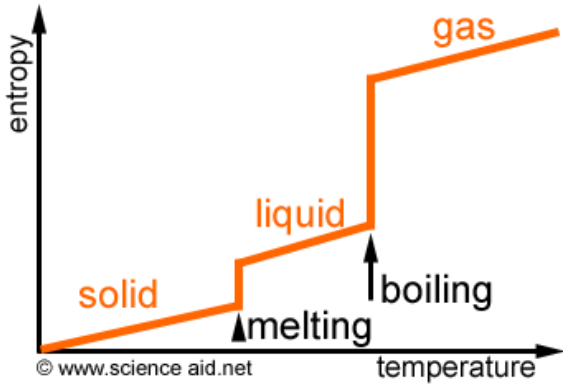


keep

Attribute Selection

How to measure attribute “worthiness”

- use entropy



Entropy

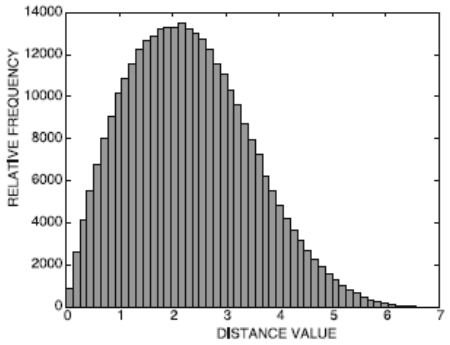
- originates in thermodynamics
- measures lack of order or predictability

Entropy in statistics and information theory

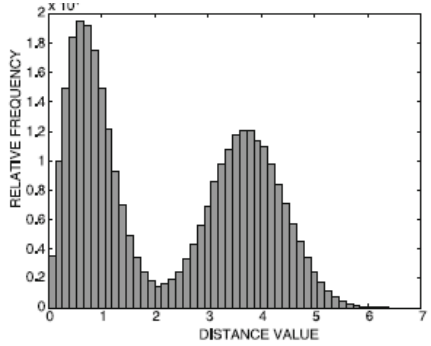
- has a value of 1 for uniform distributions (not predictable)
- knowing the value has a lot of information (high surprise)
- a value of 0 for a constant value (fully predicable)
- knowing the value has zero information (low surprise)

ENTROPY

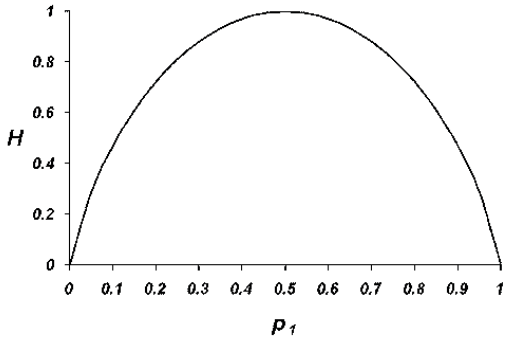
Assume m bins, $1 \leq i \leq m$: $E = -\sum_{i=1}^{m}[p_i \log(p_i) + (1-p_i)\log(1-p_i)].$



(c) Distance distribution (uniform)



(d) Distance distribution (clustered)



E high

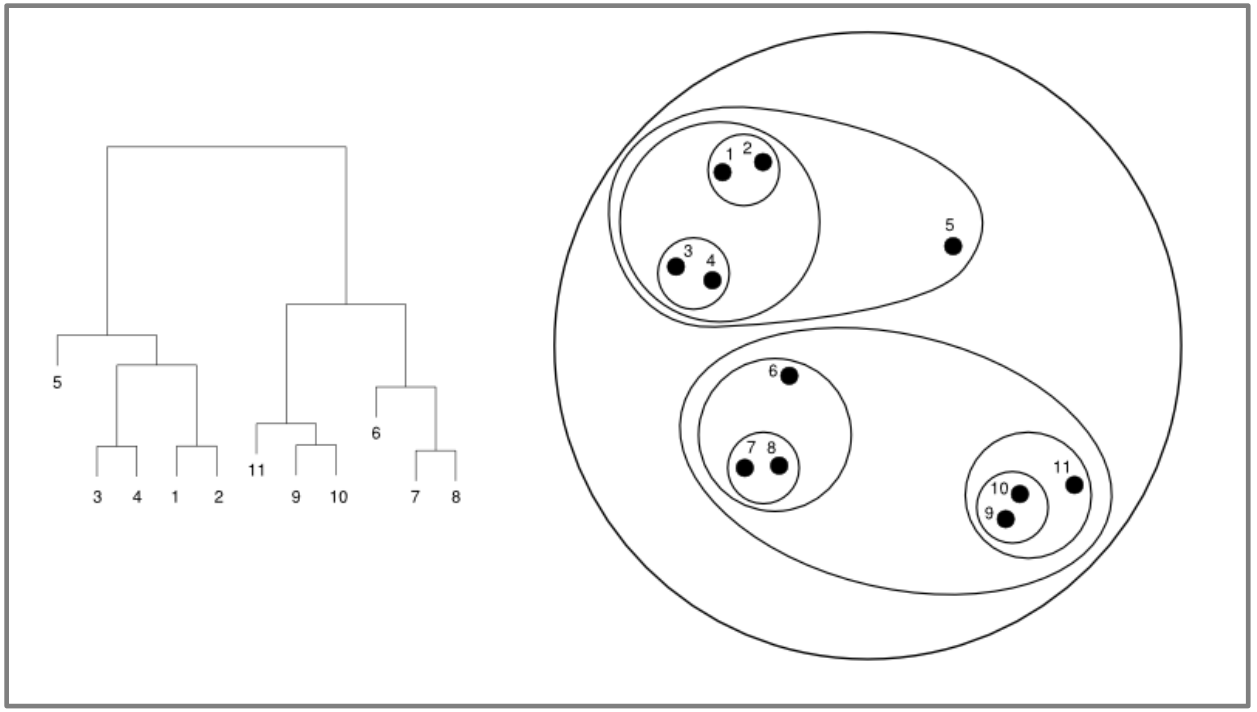
E low

Binary source (e.g. coin)

Algorithm:

- start with all attributes and compute distance entropy
- greedily eliminate attributes that reduce the entropy the most
- stop when entropy no longer reduces or even increases

Hierarchical Clustering

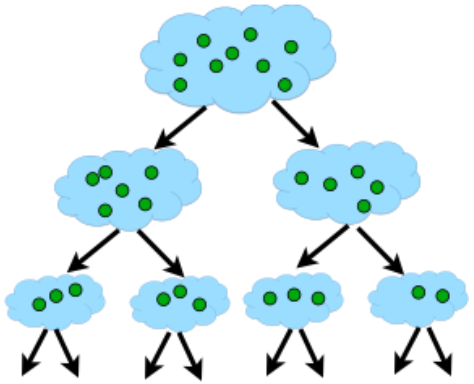


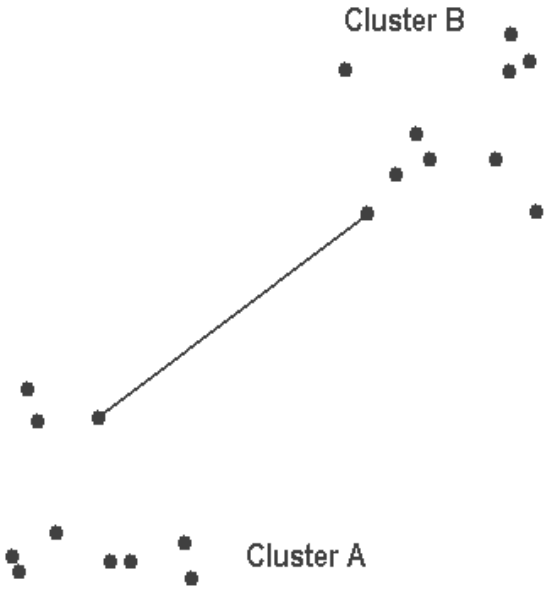
Two options:

- top down (divisive)
- bottom up (agglomerative)

Bottom-Up Agglomerative Methods

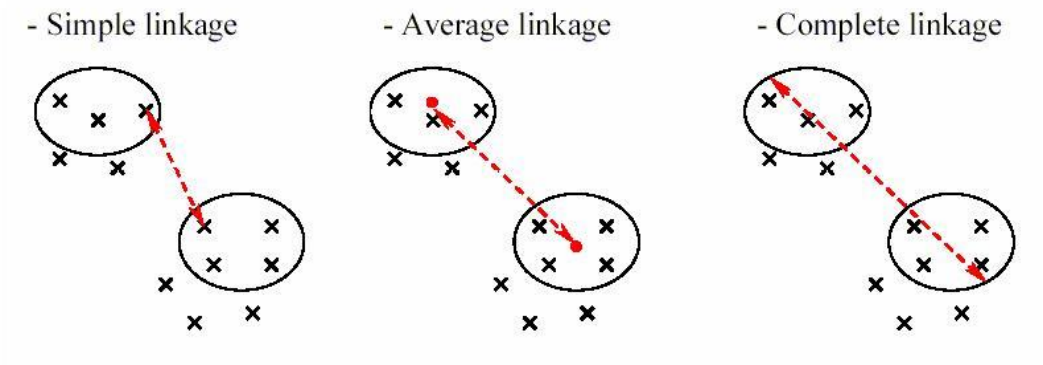
Algorithm *AgglomerativeMerge*(Data: $\mathcal{D}$)
begin
  Initialize $n \times n$ distance matrix M using $\mathcal{D}$;
  repeat
    Pick closest pair of clusters i and j using M;
    Merge clusters i and j;
    Delete rows/columns i and j from M and create
      a new row and column for newly merged cluster;
    Update the entries of new row and column of M;
  until termination criterion;
  return current merged cluster set;
end





How to merge?

Merge Criteria



Single linkage

- distance = minimum distance between all $m_i \cdot m_j$ pairs of objects
- joins the closest pair

Worst (complete) linkage

- distance = maximum distance between all $m_i \cdot m_j$ pairs of objects
- joins the pair furthest apart

Group-average linkage

- distance = average distance between all object pairs in the groups

Other methods:

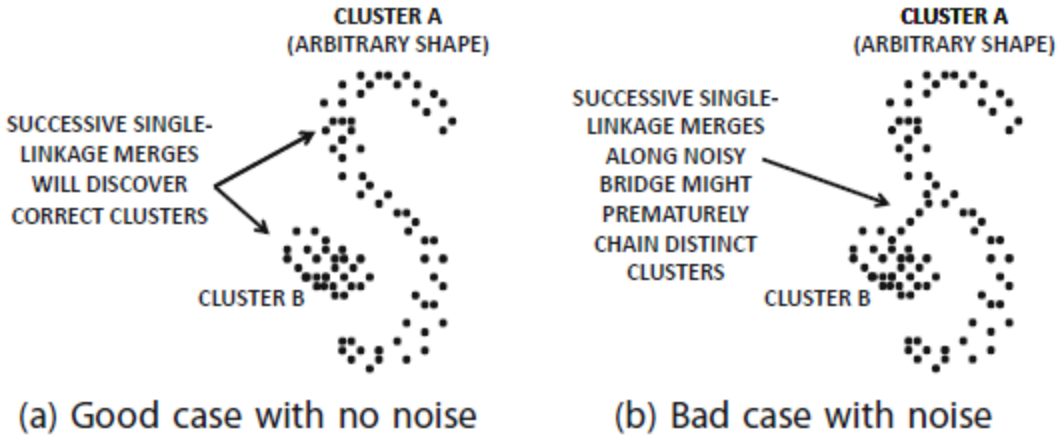
- closest centroid, variance-minimization, Ward's method

Comparison

Centroid-based methods tend to merge large clusters

Single linkage method can merge chains of closely related points to discover clusters of arbitrary shape

- but can also (inappropriately) merge two unrelated clusters, when the chaining is caused by noisy points between two clusters



(a) Good case with no noise

(b) Bad case with noise

Comparison

Complete (worst-case) linkage method tends to create spherical clusters with similar diameter

- will break up the larger clusters into smaller spheres
- also gives too much importance to data points at the noisy fringes of a cluster

The group average, variance, and Ward's methods are more robust to noise due to the use of multiple linkages in the distance computation

Hierarchical methods are sensitive to a small number of mistakes made during the merging process

- can be due to noise
- no way to undo these mistakes

DBSCAN

Highly-cited density-based hierarchical clustering algorithm (Ester et al. 1996)

- clusters are defined as density-connected sets
- epsilon-distance neighbor criterion (Eps)

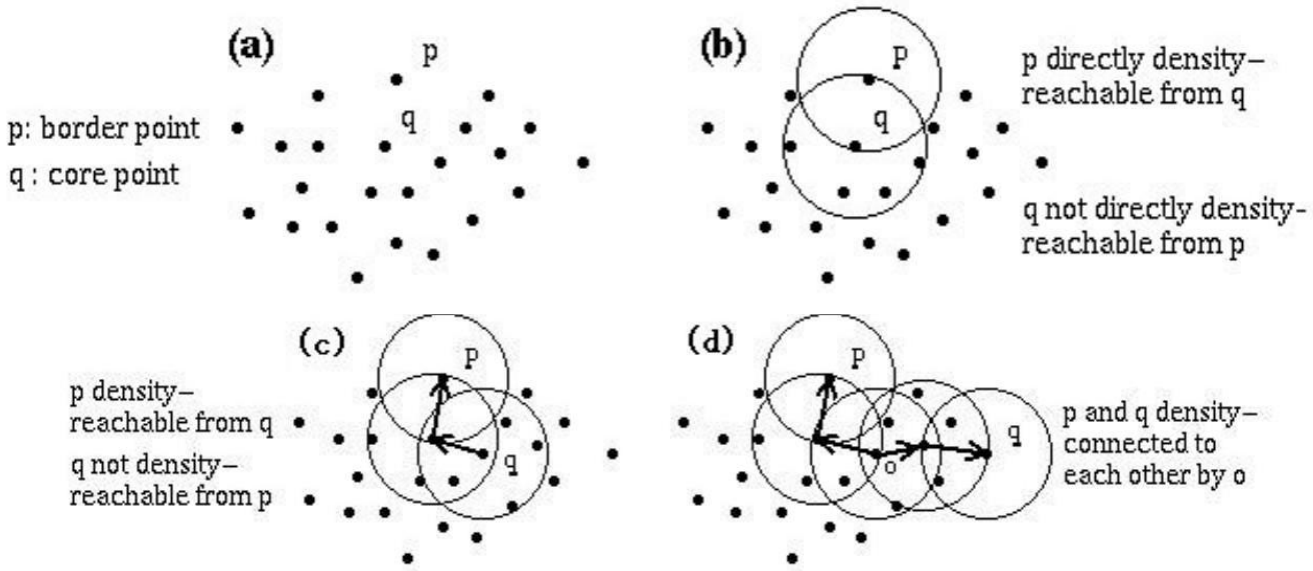
  $N_{Eps}(p) = \{q \in D \mid dist(p,q) \leq Eps\}$
- minimum point cluster membership and core point (MinPts)

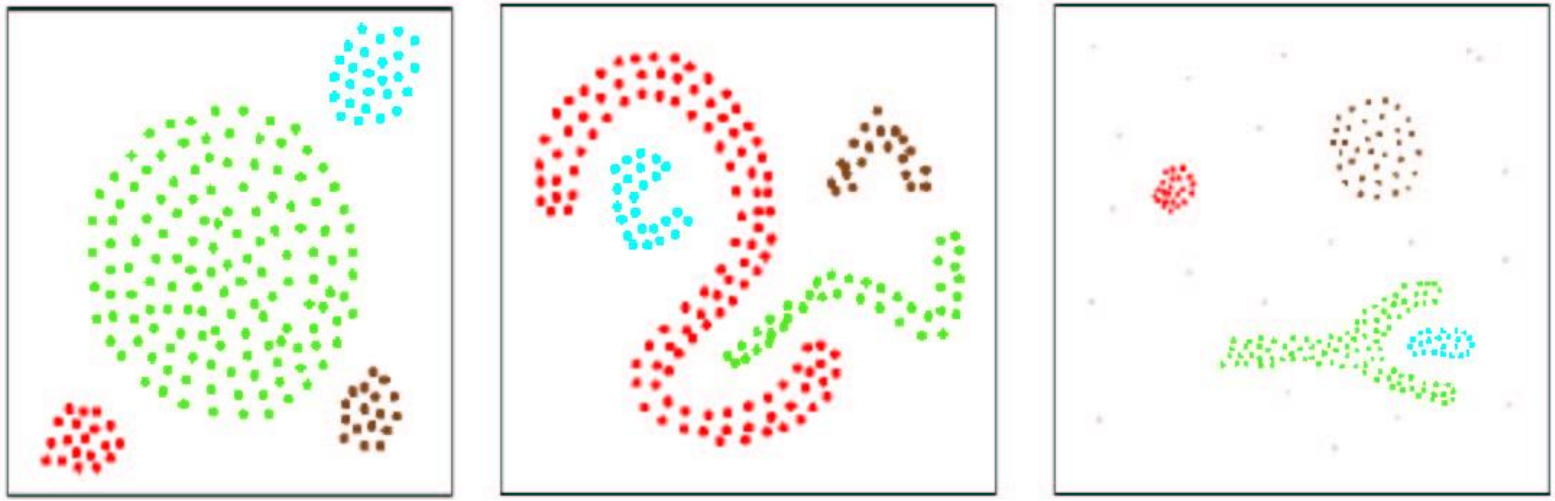
  $|N_{Eps}(q)| \geq MinPts$
- notions of density-connected & density-reachable (direct, indirect)
- a point p is directly density-reachable from a point q wrt. Eps, MinPts if

  $p \in N_{Eps}(q)$ and

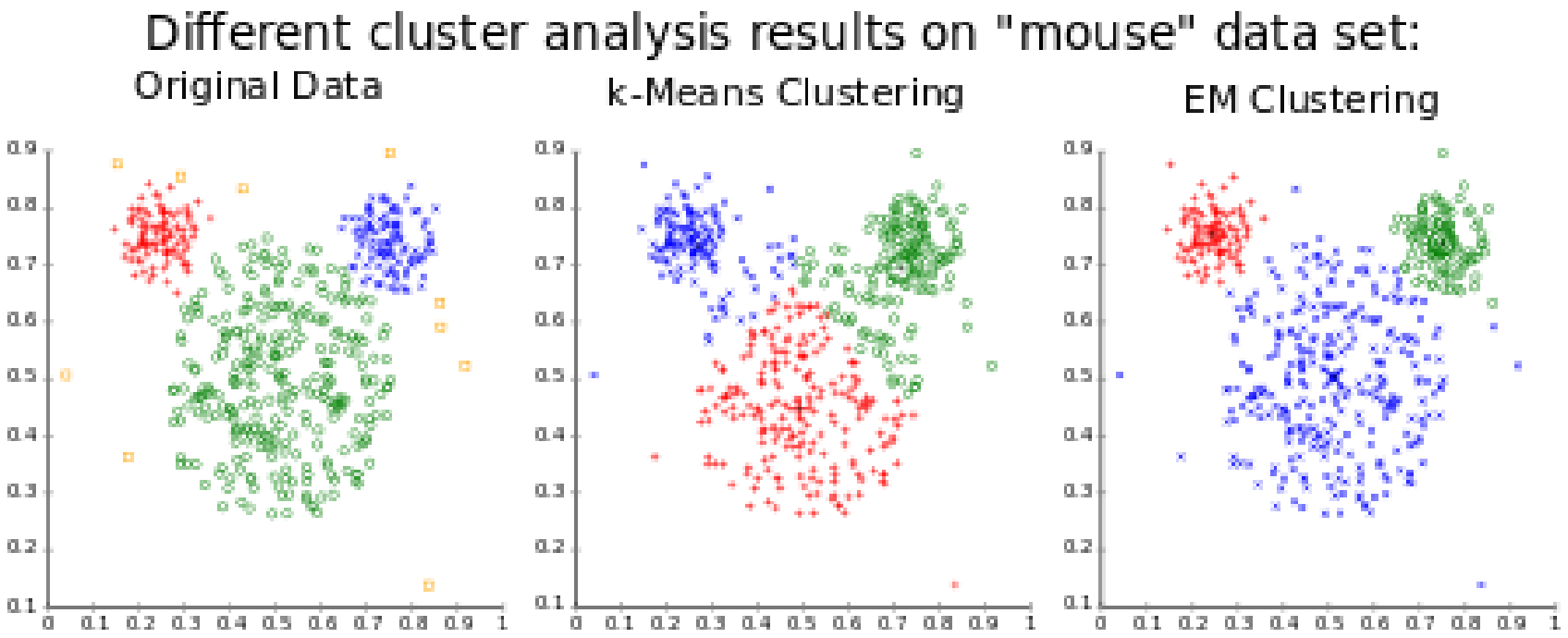
  $|N_{Eps}(q)| \geq MinPts$ (core point condition)

DBSCAN


(a)
p
q
p: border point
q : core point
(b)
p
q
p directly density–reachable from q
q not directly density-reachable from p
(c)
p
q
p density–reachable from q
q not density–reachable from p
(d)
p
q
o
p and q density–connected to each other by o



Probabilistic Extension to K-Means



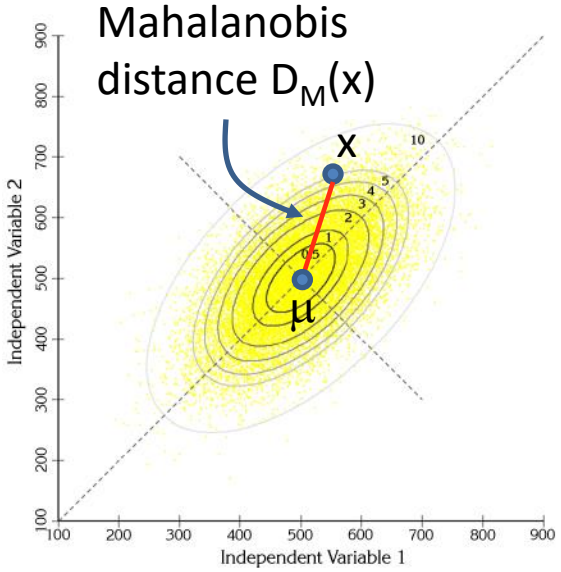
Mahalanobis Distance

The distance between a point P and a distribution D

- measures how many standard deviations P is away from the mean of D
- S is the covariance matrix of the distribution D
- the Mahanalobis distance D_M of a point x to a cluster center μ is

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1} (x-\mu)}.$$

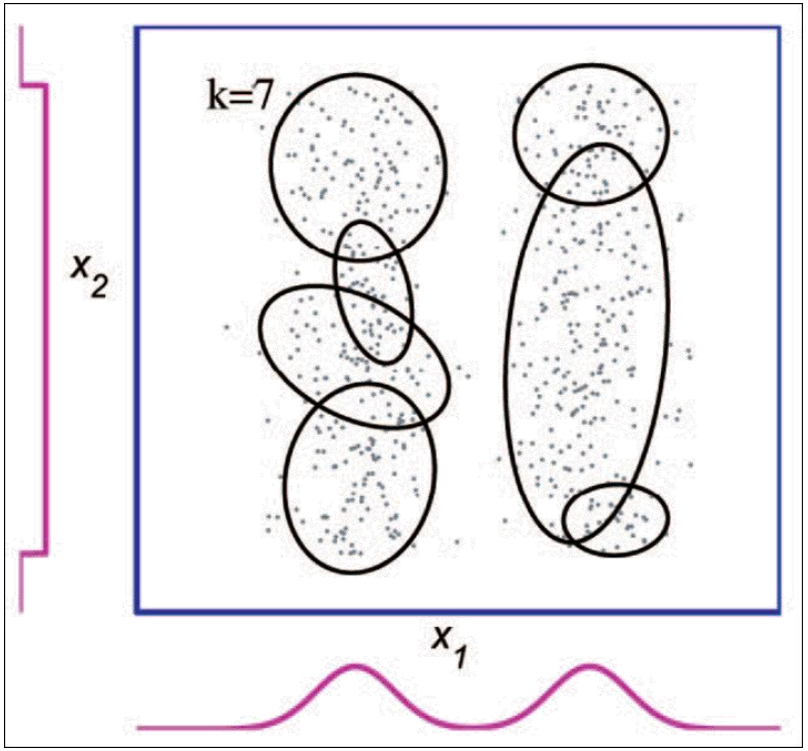
- x and μ are N-dimensional vectors
- S is a N×N matrix
- the outcome $D_M(x)$ is a single-dimensional number



Probabilistic Clustering

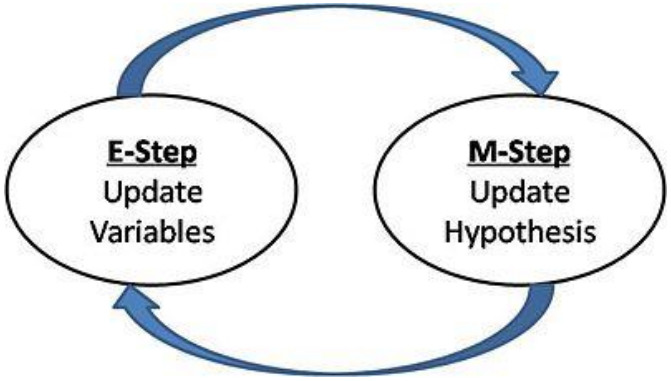
Better match for point distributions

- overlapping clusters are now possible
- better match with real world?
- Gaussian mixtures



Need a probabilistic algorithm

- Expectation-Maximization



EM Algorithm (Mixture Model)

probability that d_i is in class c_j
(Mahanalobis distance of d_i to c_j)

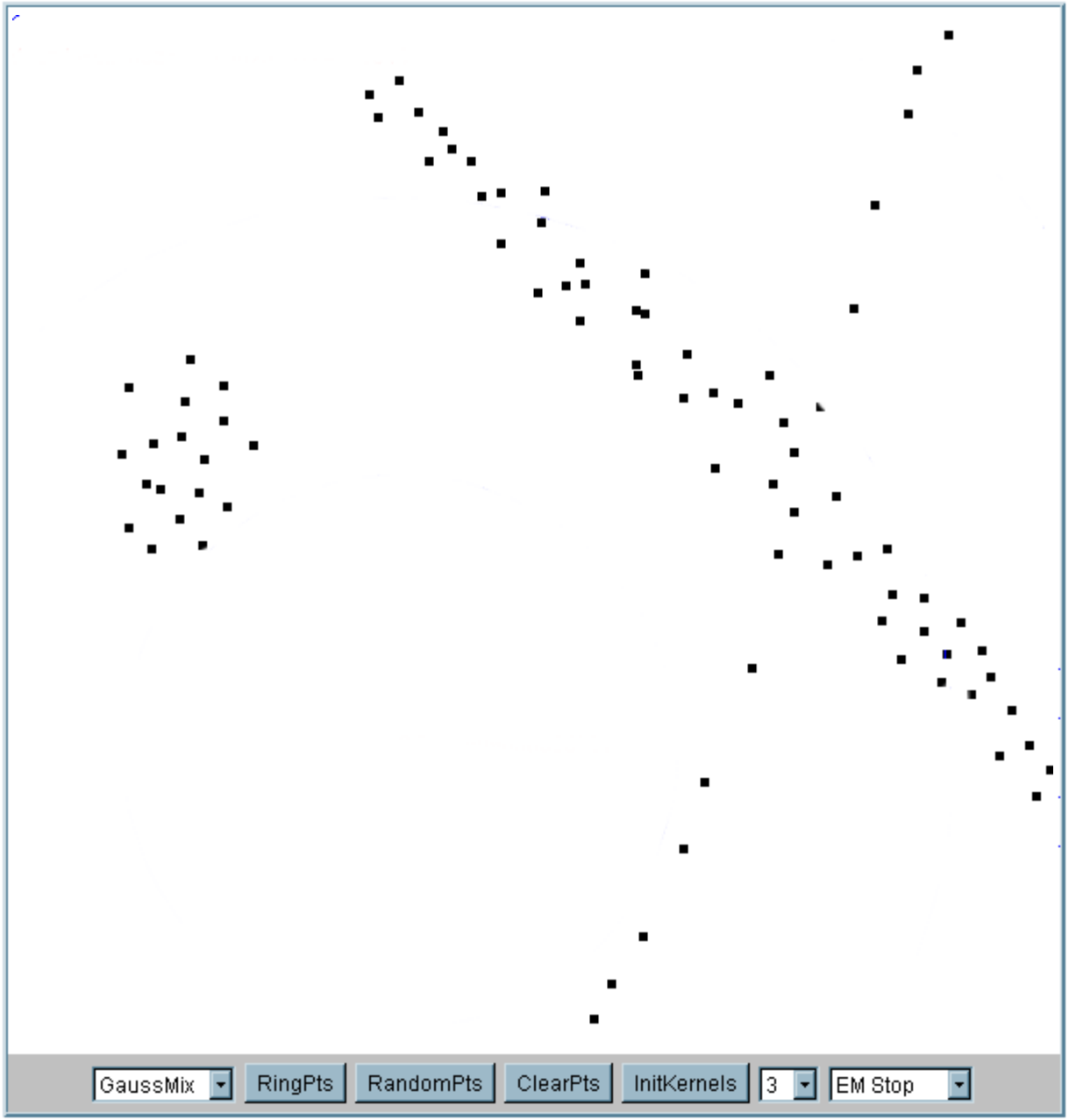
- Initialize K cluster centers
- Iterate between two steps
  - **E**xpectation step: assign n points to m clusters/classes

$$P(d_i \in c_k) = w_k \Pr(d_i \mid c_k) \Big/ \sum_j w_j \Pr(d_i \mid c_j)$$

$$w_k = \frac{\sum_i \Pr(d_i \in c_k)}{n}$$ = probability of class c_k

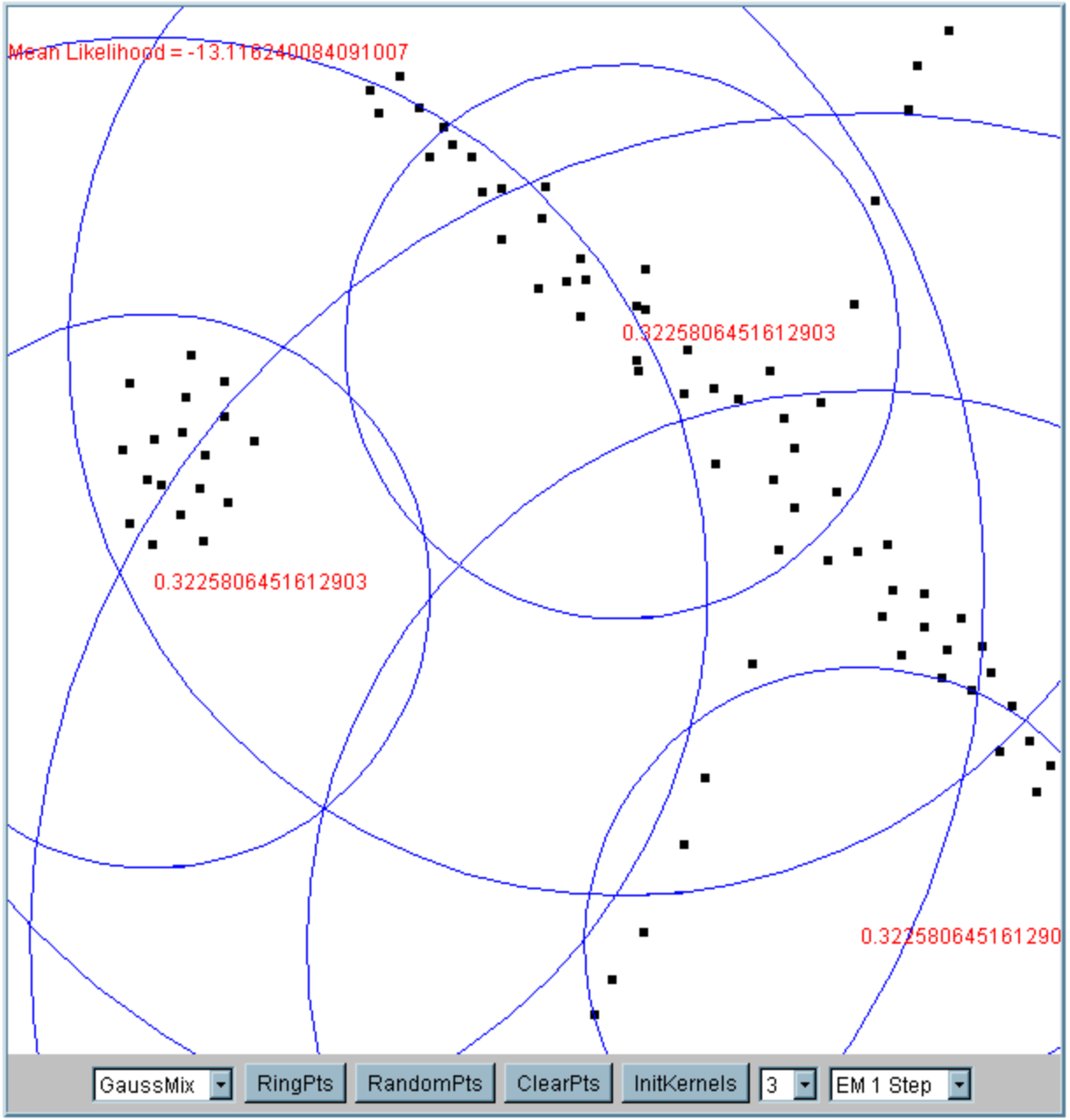
  - **M**aximation step: estimate model parameters

$$\mu_k = \frac{1}{n} \sum_{i=1}^{n} \frac{d_i P(d_i \in c_k)}{\sum_k P(d_i \in c_j)}$$

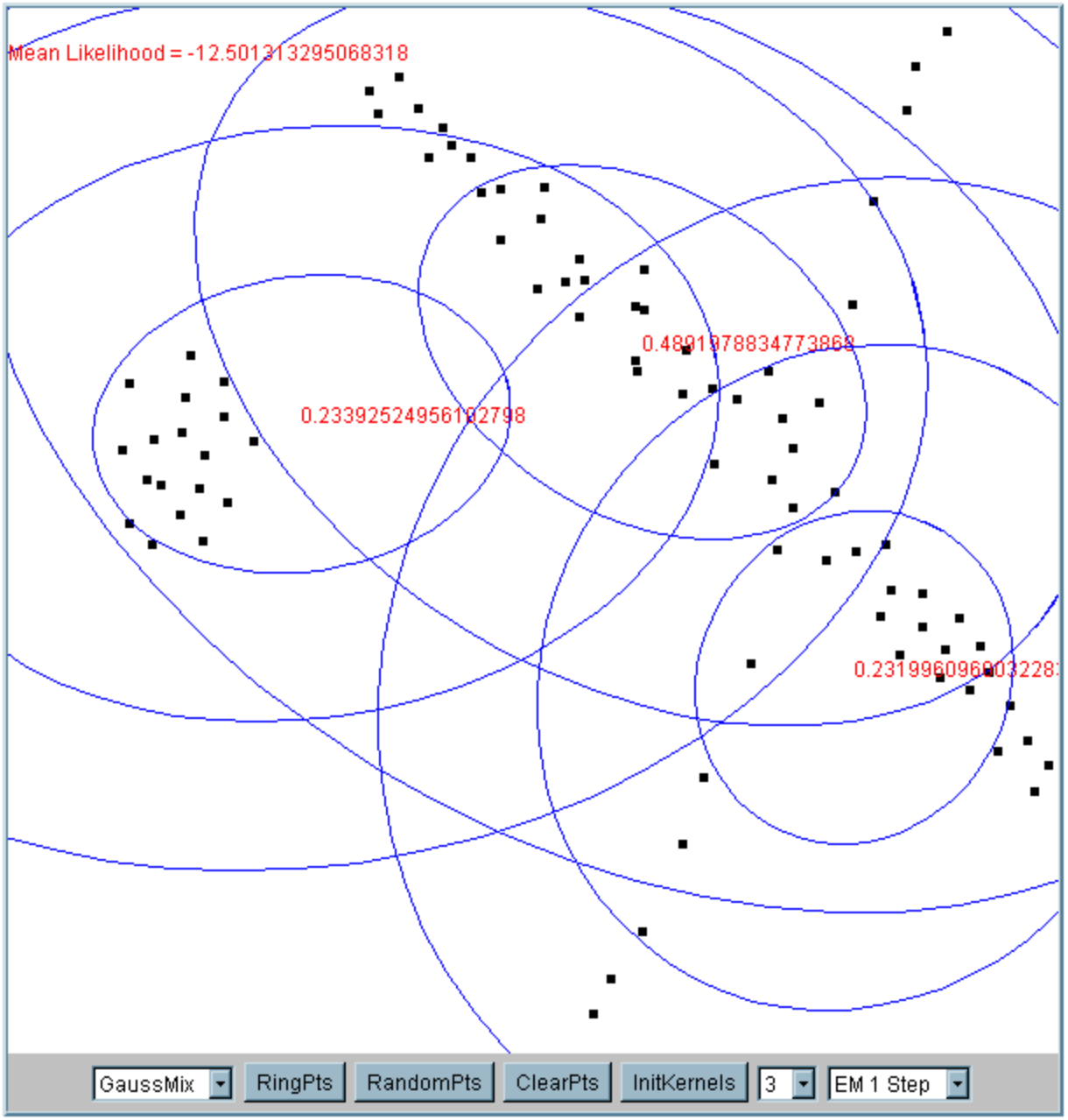

GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM Stop

Iteration 1

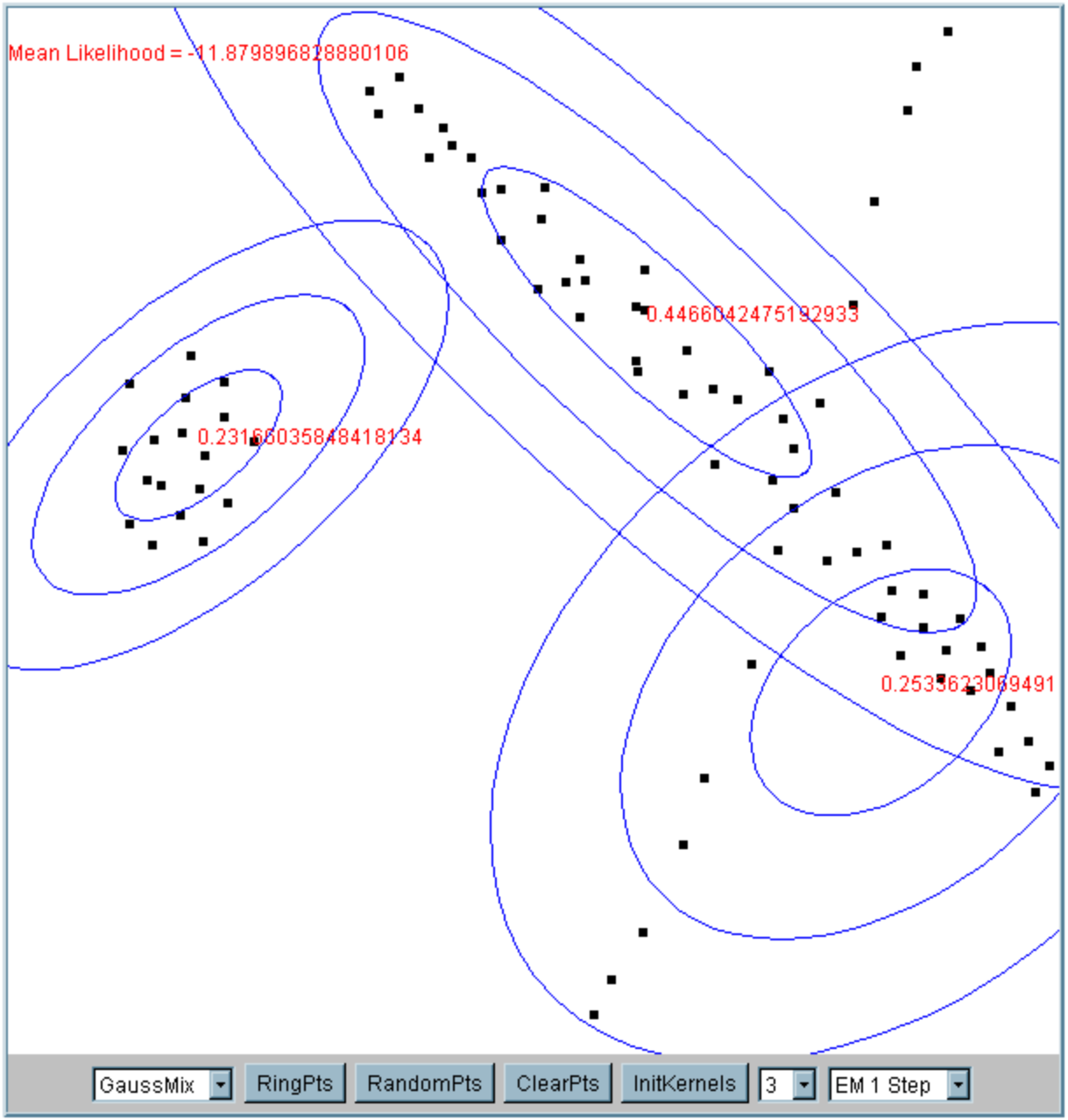
The cluster means are randomly assigned



Iteration 2


Mean Likelihood = -12.501313295068318
0.48918788834773868
0.233925249561102798
0.2319960960032281
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 5


Mean Likelihood = -11.879896828880106
0.4466042475192933
0.2316603584841813 4
0.2535623069491
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 25