ECE 5984: Introduction to Machine Learning

Topics:

- Expectation Maximization
  - For GMMs
  - For General Latent Model Learning

Readings: Barber 20.1-20.3

Dhruv Batra

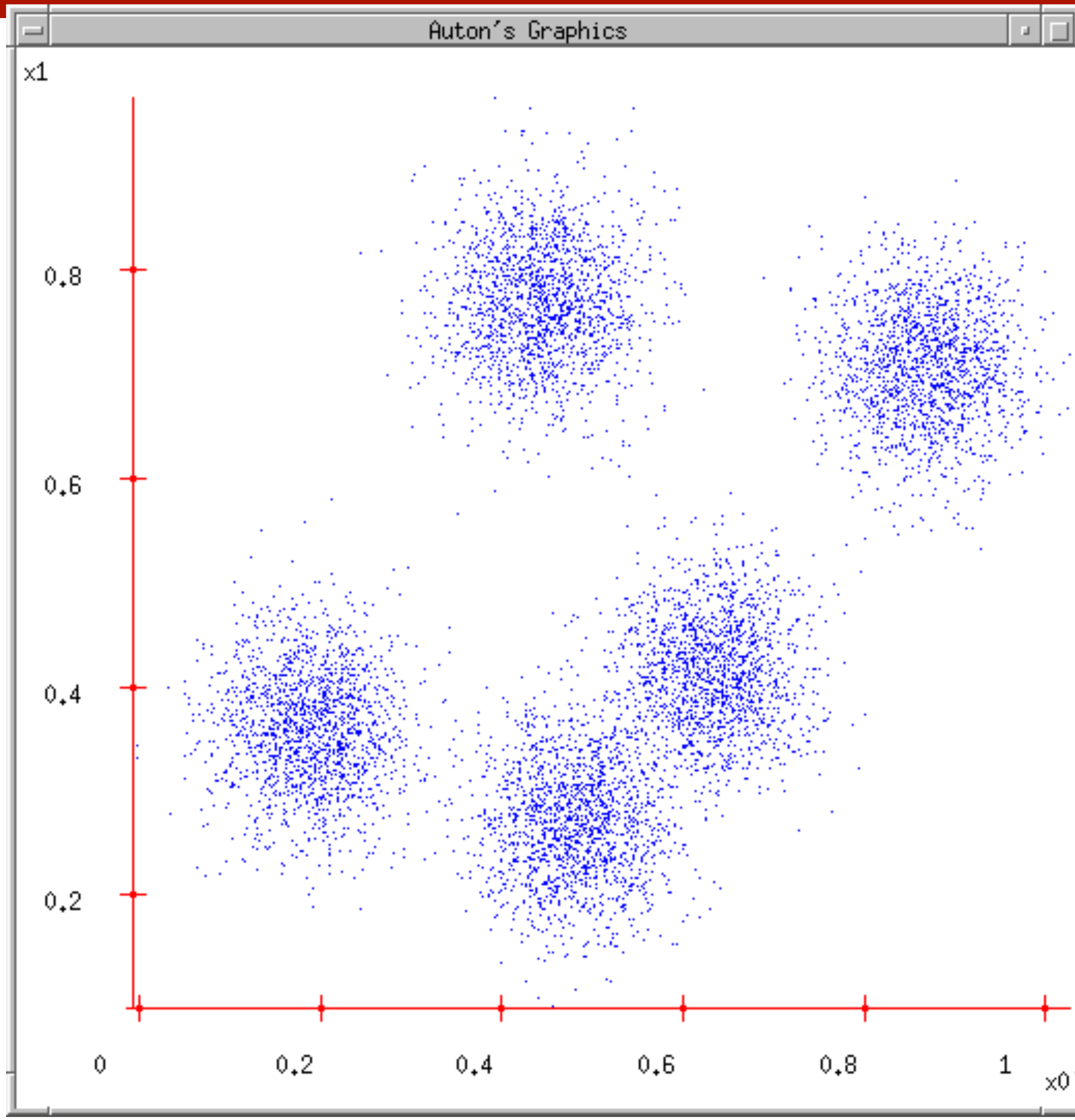
Virginia Tech

Administrativia

- Poster Presentation: Best Project Prize!
  - May 8 1:30-4:00pm
  - 310 Kelly Hall: ICTAS Building
  - Print poster (or bunch of slides)
  - Format:
    - Portrait
    - Eg. 2 feet (width) x 4 feet (height)
  - Less text, more pictures.

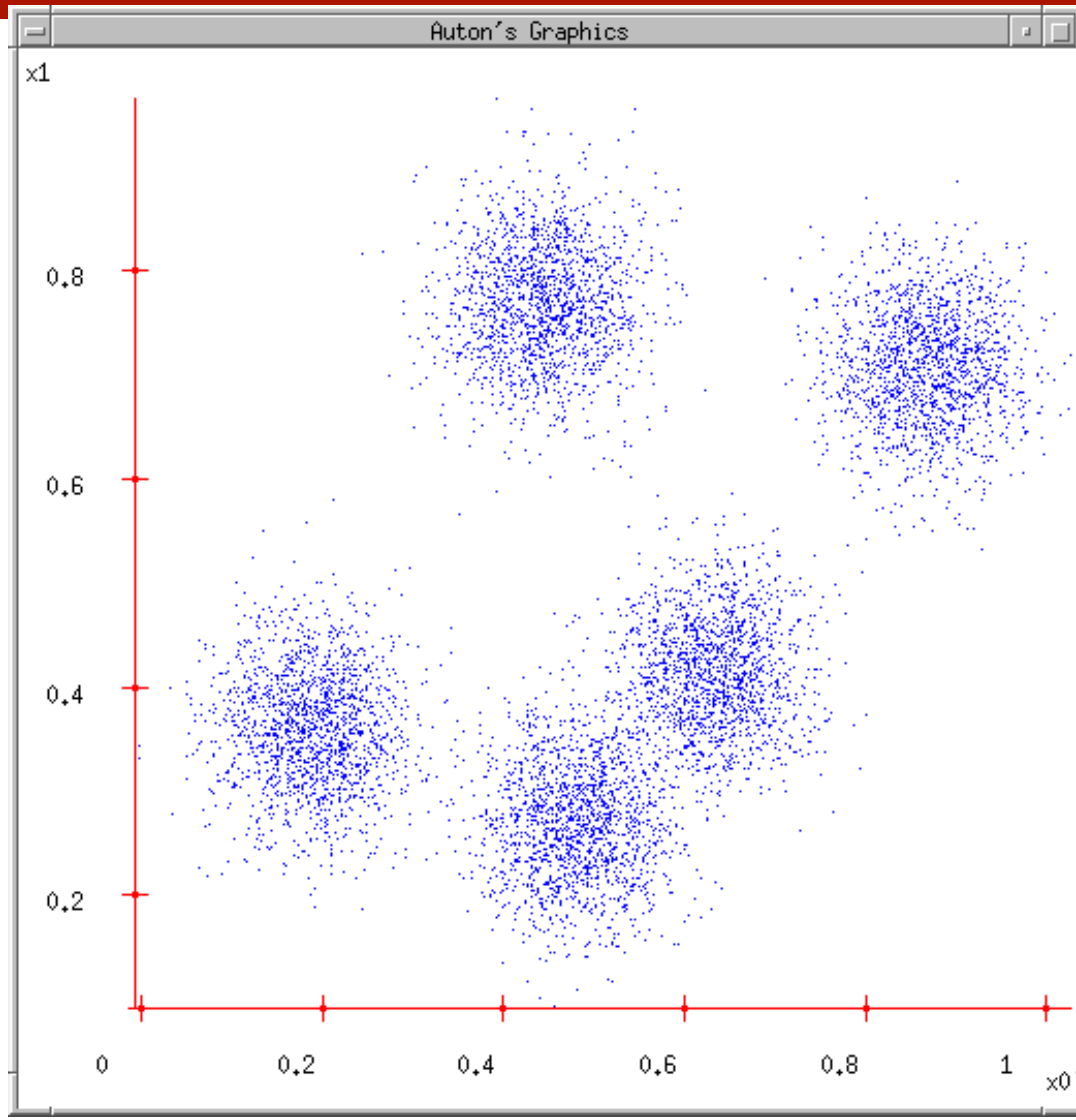
Recap of Last Time

Some Data



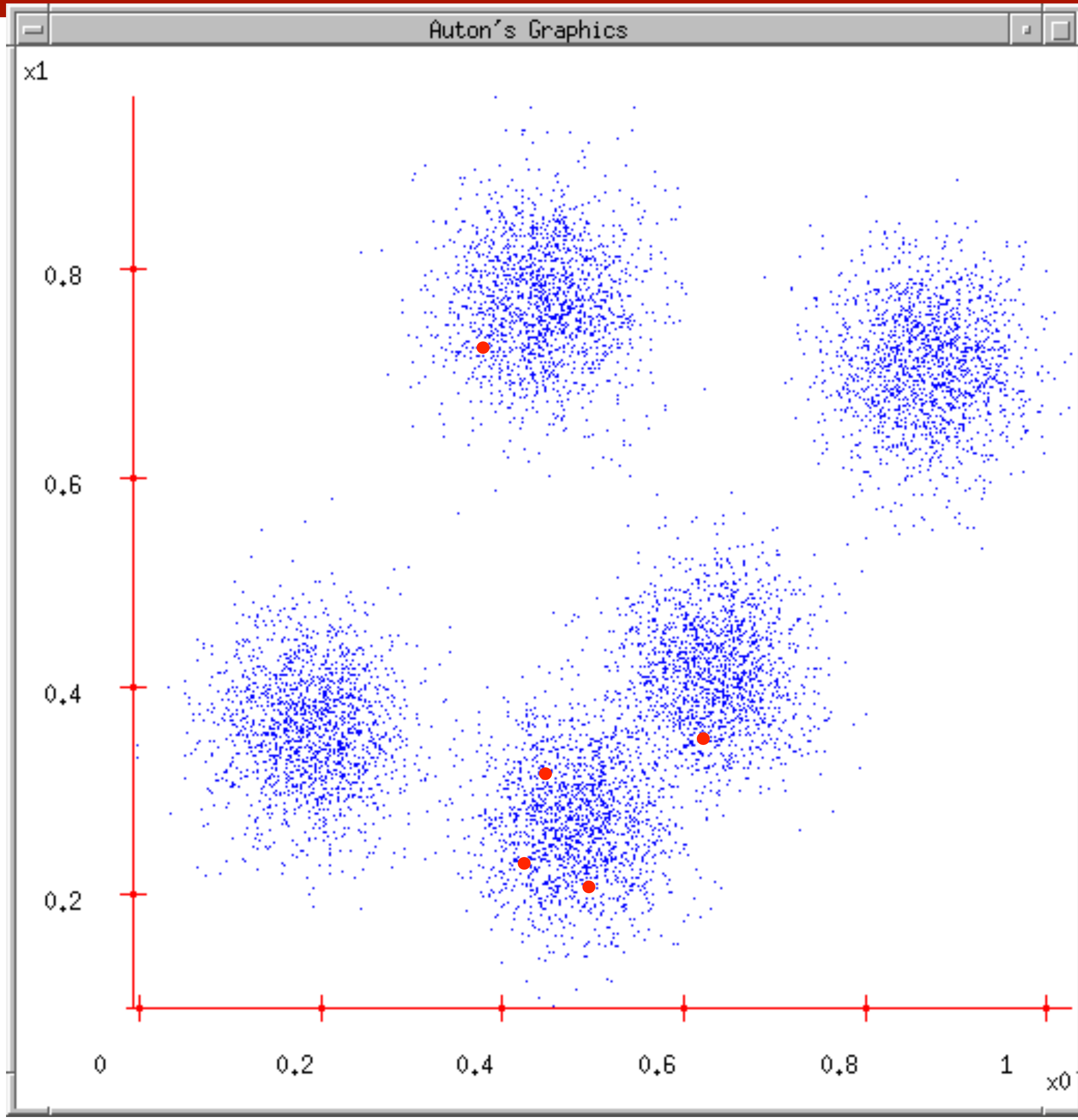
K-means

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



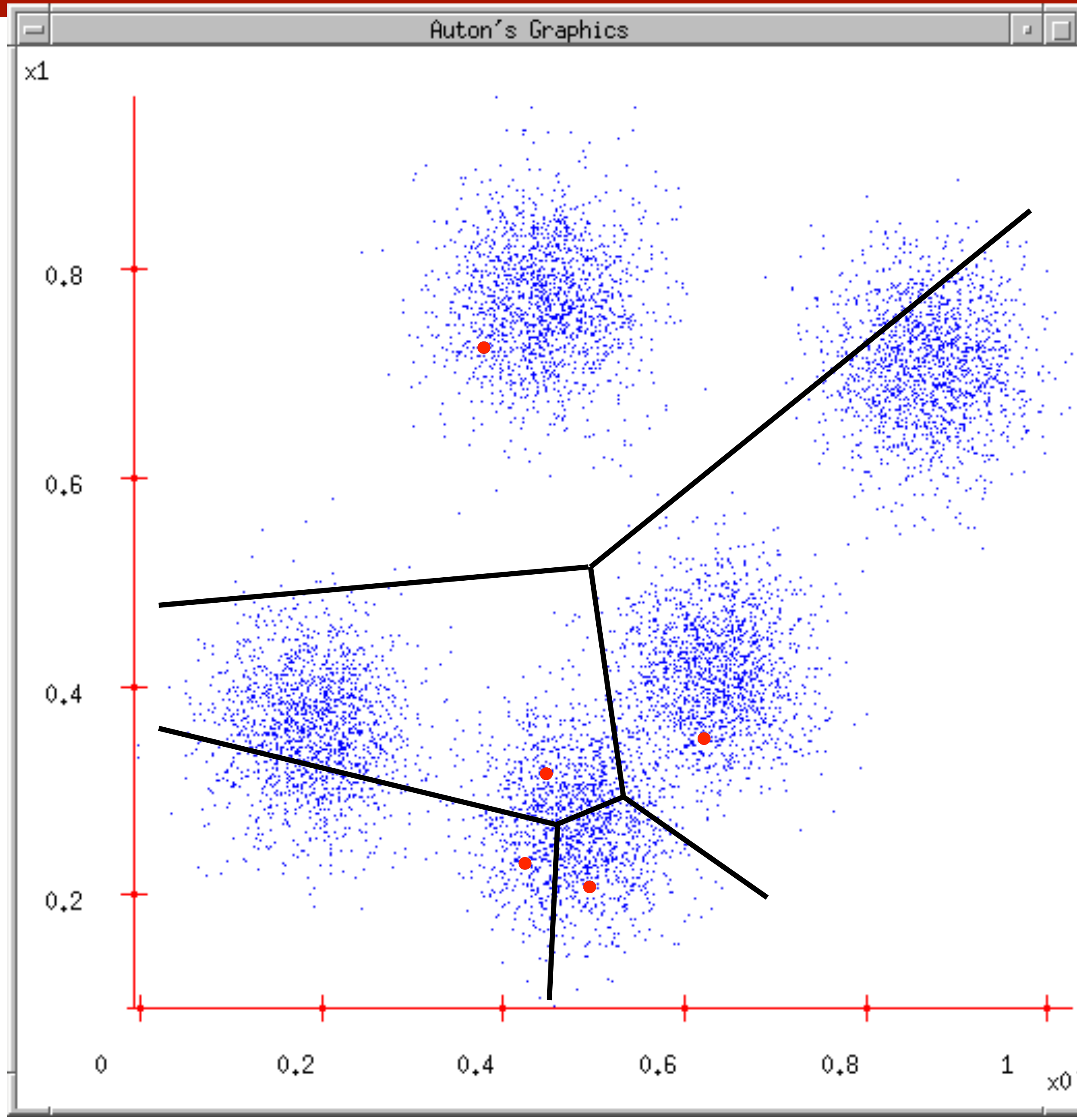
K-means

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



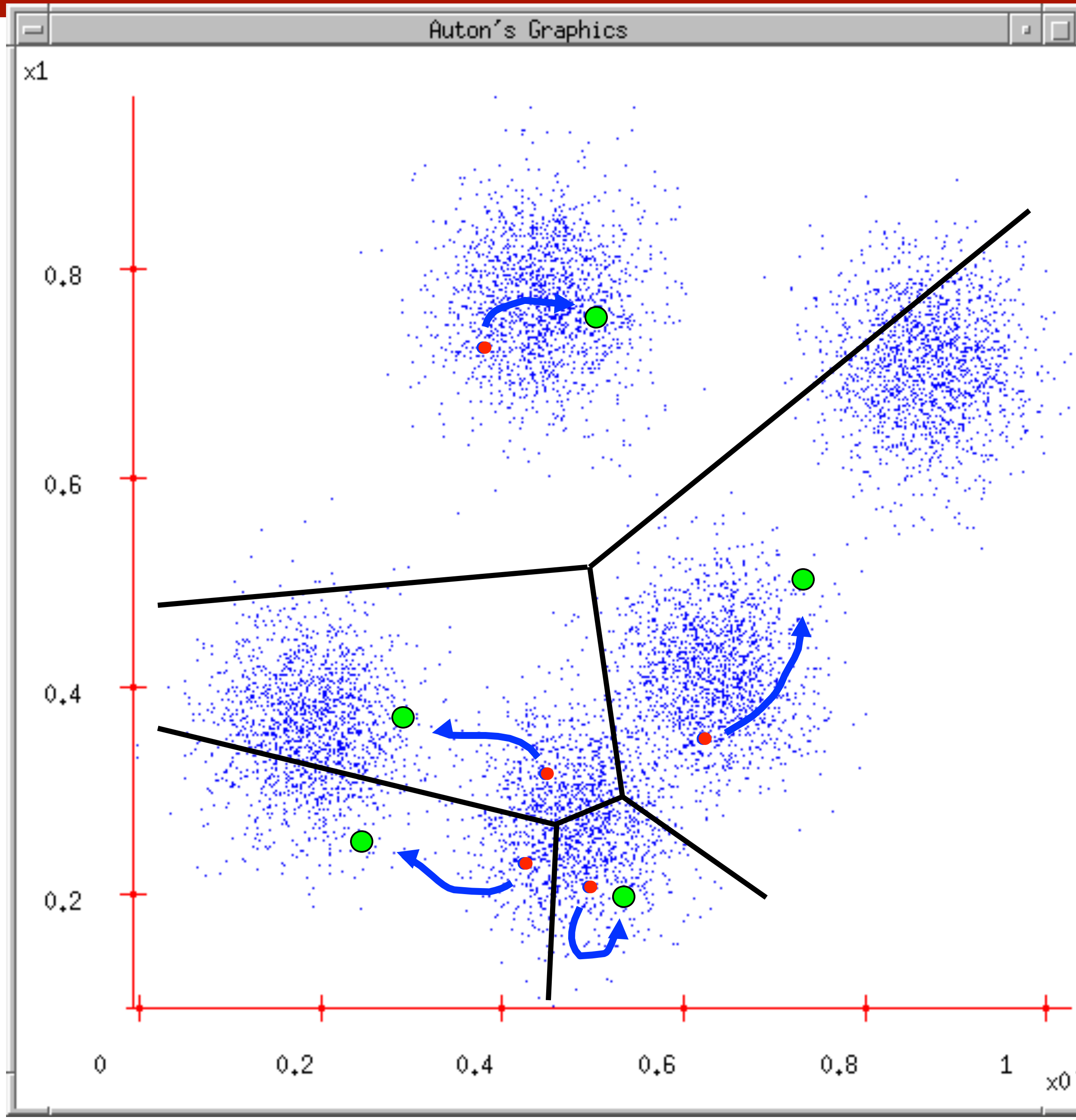
K-means

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)



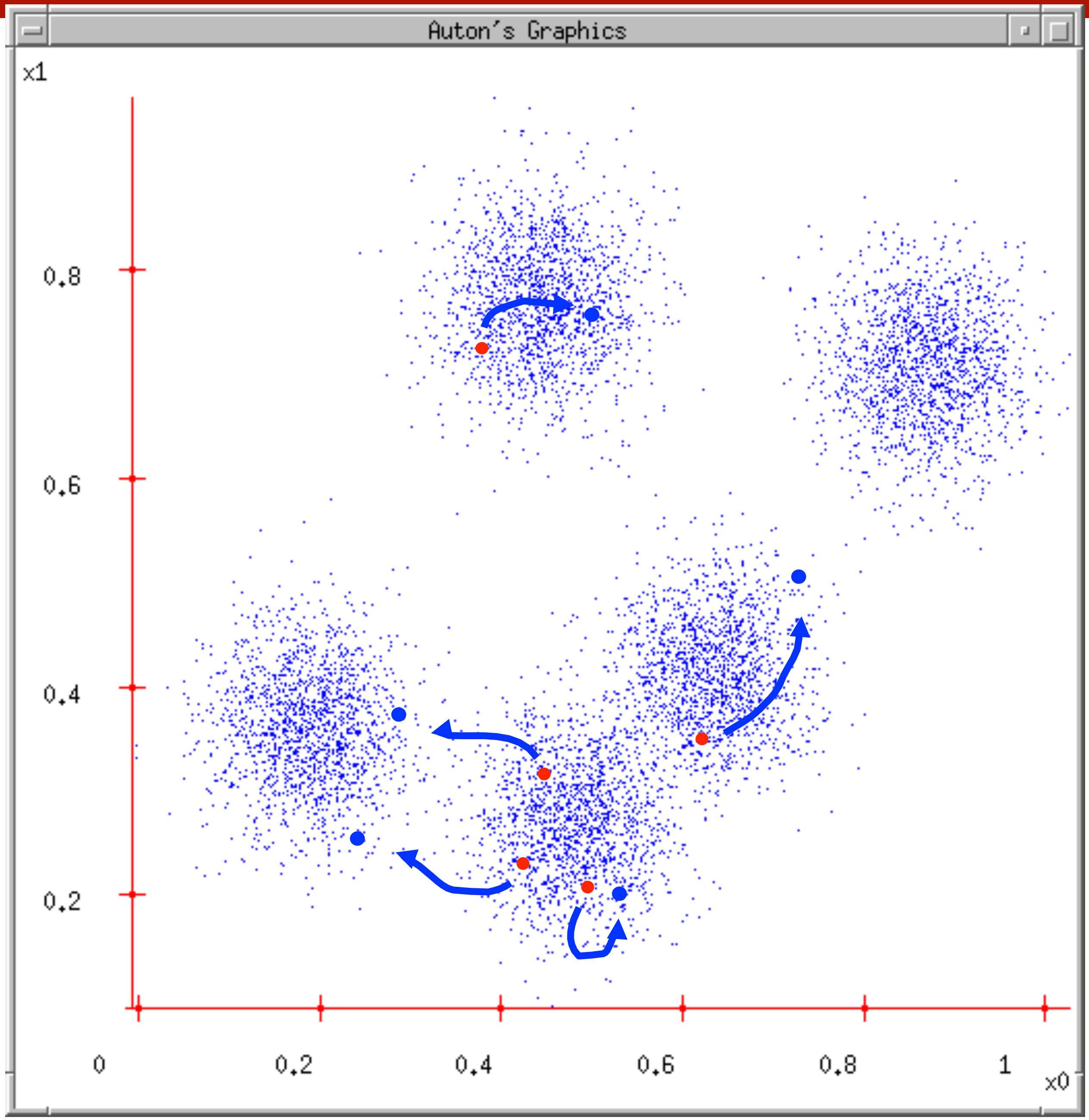
K-means

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

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$
- **Assign**:
  - Assign each point $i \in \{1, \ldots n\}$ to nearest center:
  - $C(i) \longleftarrow \underset{j}{\operatorname{argmin}} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$
- **Recenter**:
  - μ_j becomes centroid of its points

K-means as Co-ordinate Descent

- Optimize objective function:

$$\min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k} \min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_N} F(\boldsymbol{\mu}, \boldsymbol{a}) = \min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k} \min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_N} \sum_{i=1}^{N} \sum_{j=1}^{k} a_{ij} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$$

- Fix $\boldsymbol{\mu}$, optimize a (or C)

K-means as Co-ordinate Descent

- Optimize objective function:

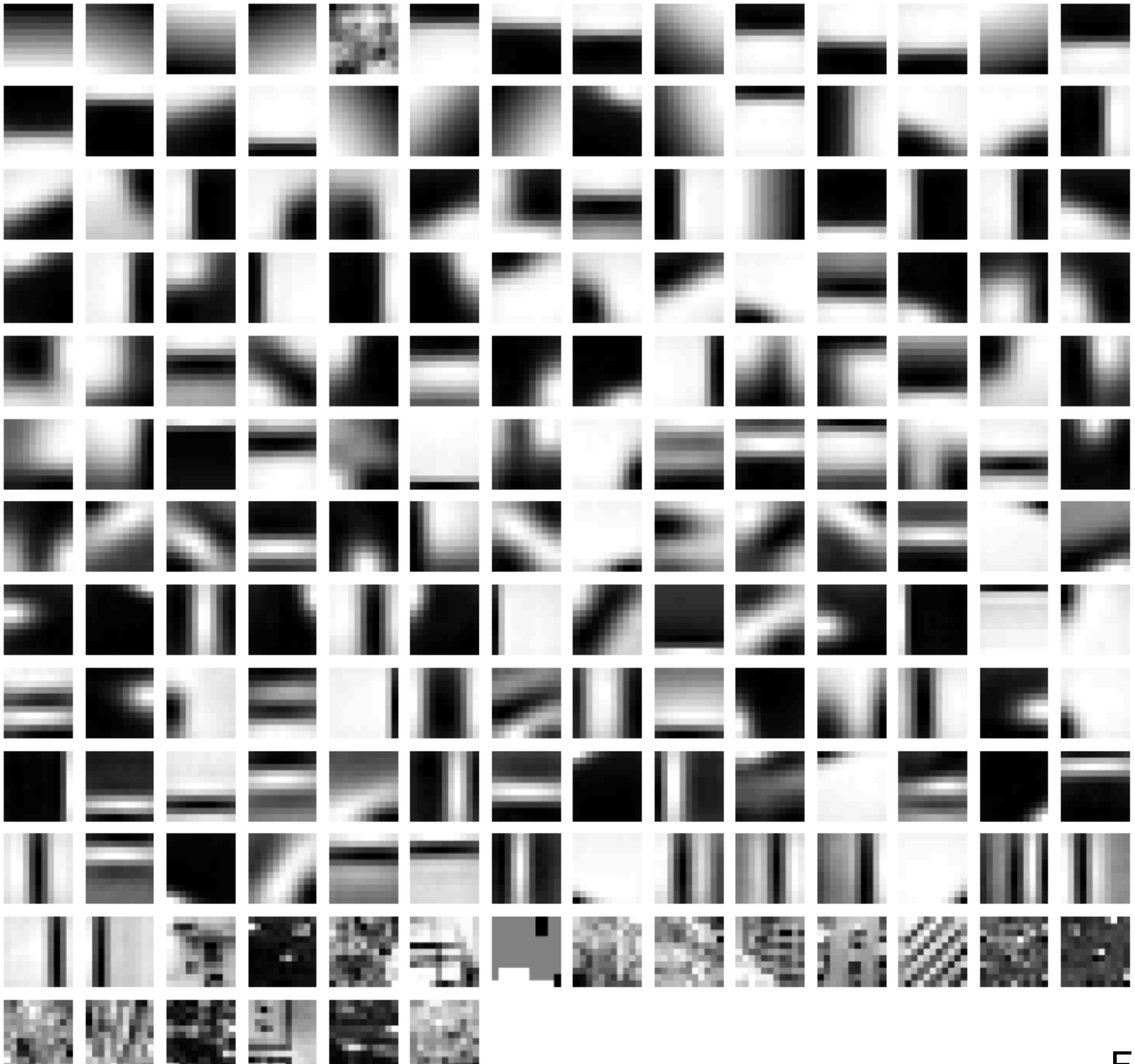
$$\min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k} \min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_N} F(\boldsymbol{\mu}, \boldsymbol{a}) = \min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k} \min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_N} \sum_{i=1}^{N} \sum_{j=1}^{k} a_{ij} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$$

- Fix a (or C), optimize μ

Object → Bag of ‘words’

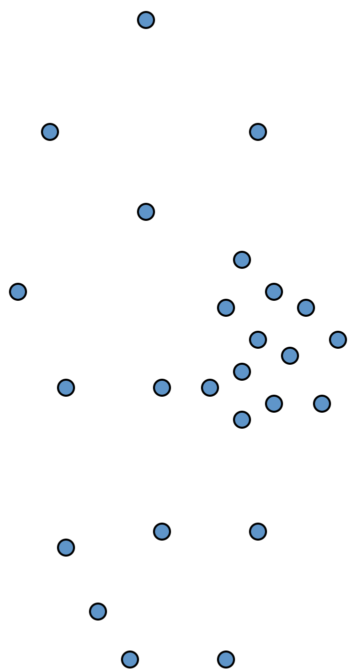
Fei-Fei Li

Clustered Image Patches



Fei-Fei et al. 2005

(One) bad case for k-means



- Clusters may overlap
- Some clusters may be "wider" than others

- GMM to the rescue!

GMM

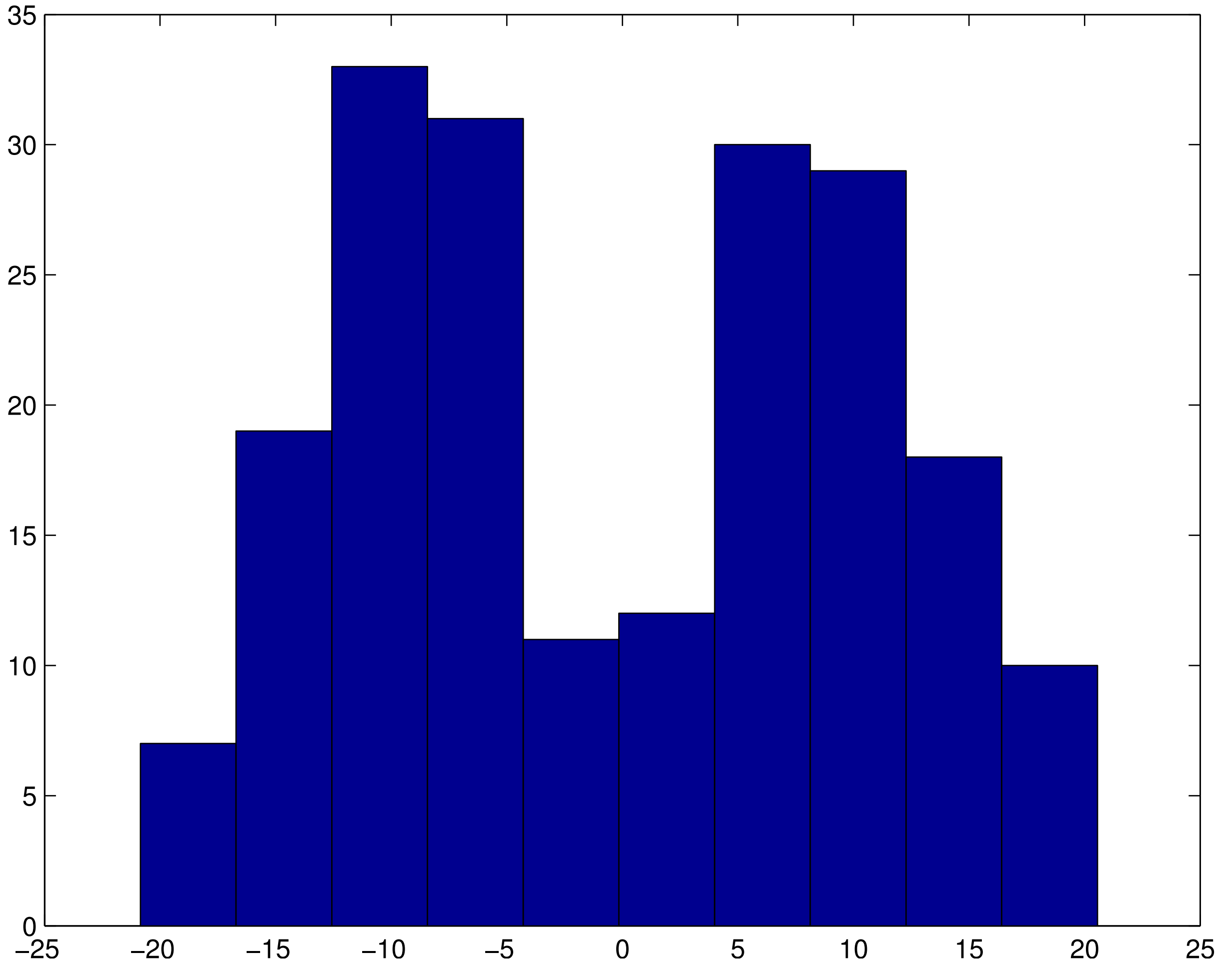
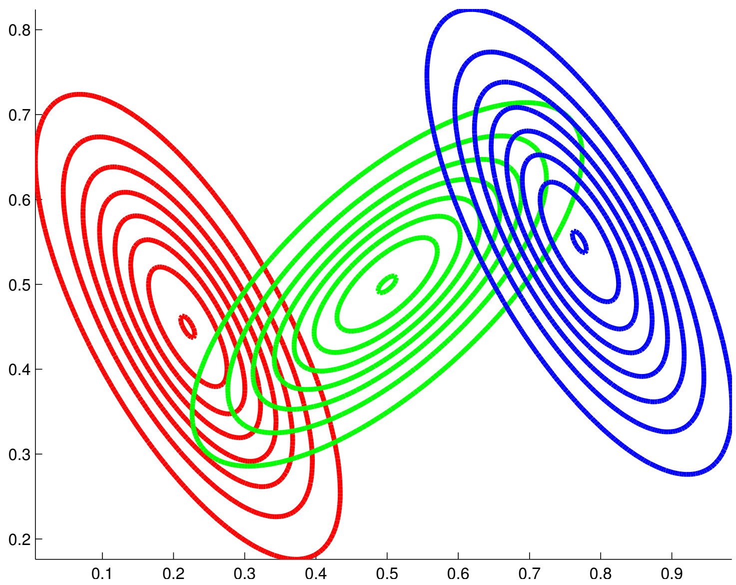
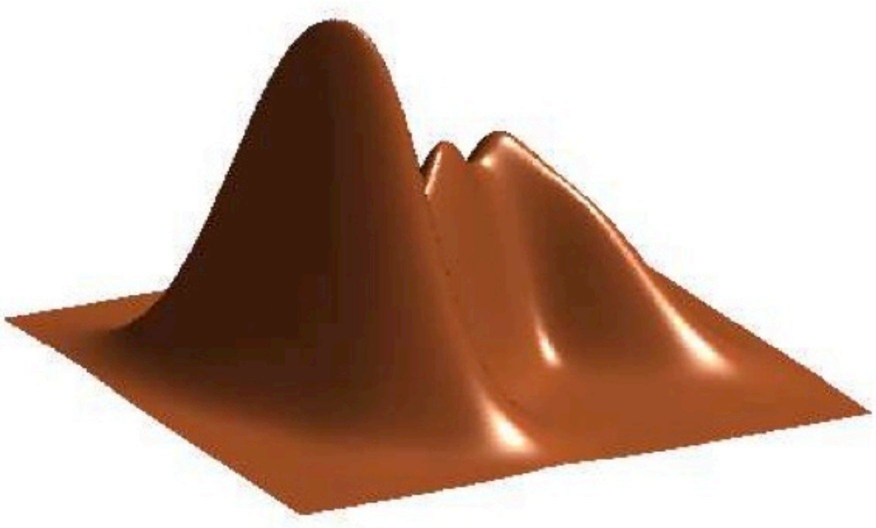


Figure Credit: Kevin Murphy

GMM





K-means vs GMM

- K-Means
  - http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

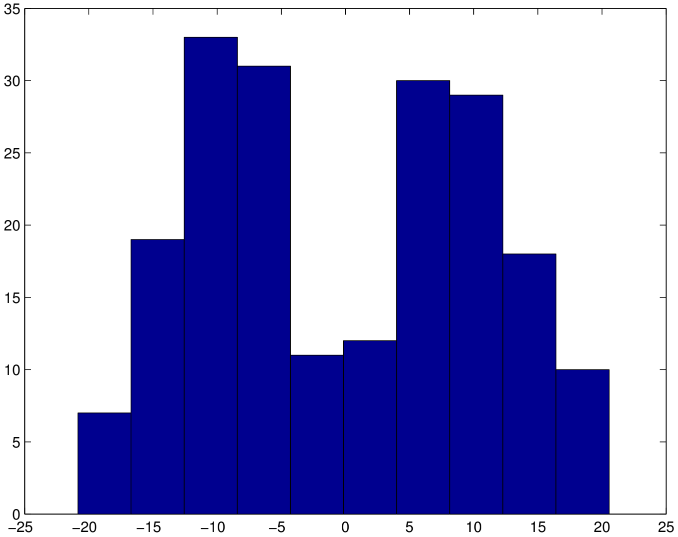
- GMM
  - http://www.socr.ucla.edu/applets.dir/mixtureem.html

Hidden Data Causes Problems #1

- Fully Observed (Log) Likelihood factorizes
- Marginal (Log) Likelihood doesn’t factorize
- All parameters coupled!

Hidden Data Causes Problems #2

- Identifiability



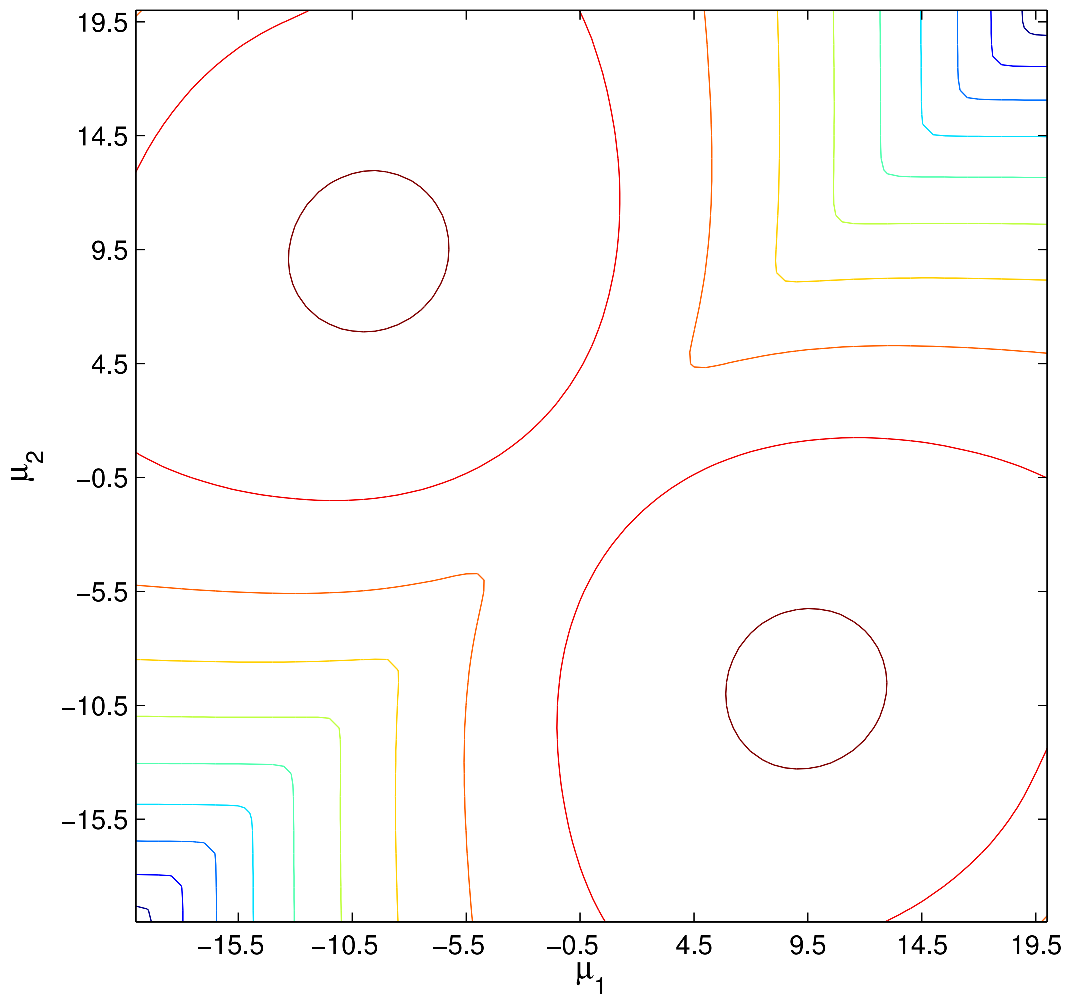
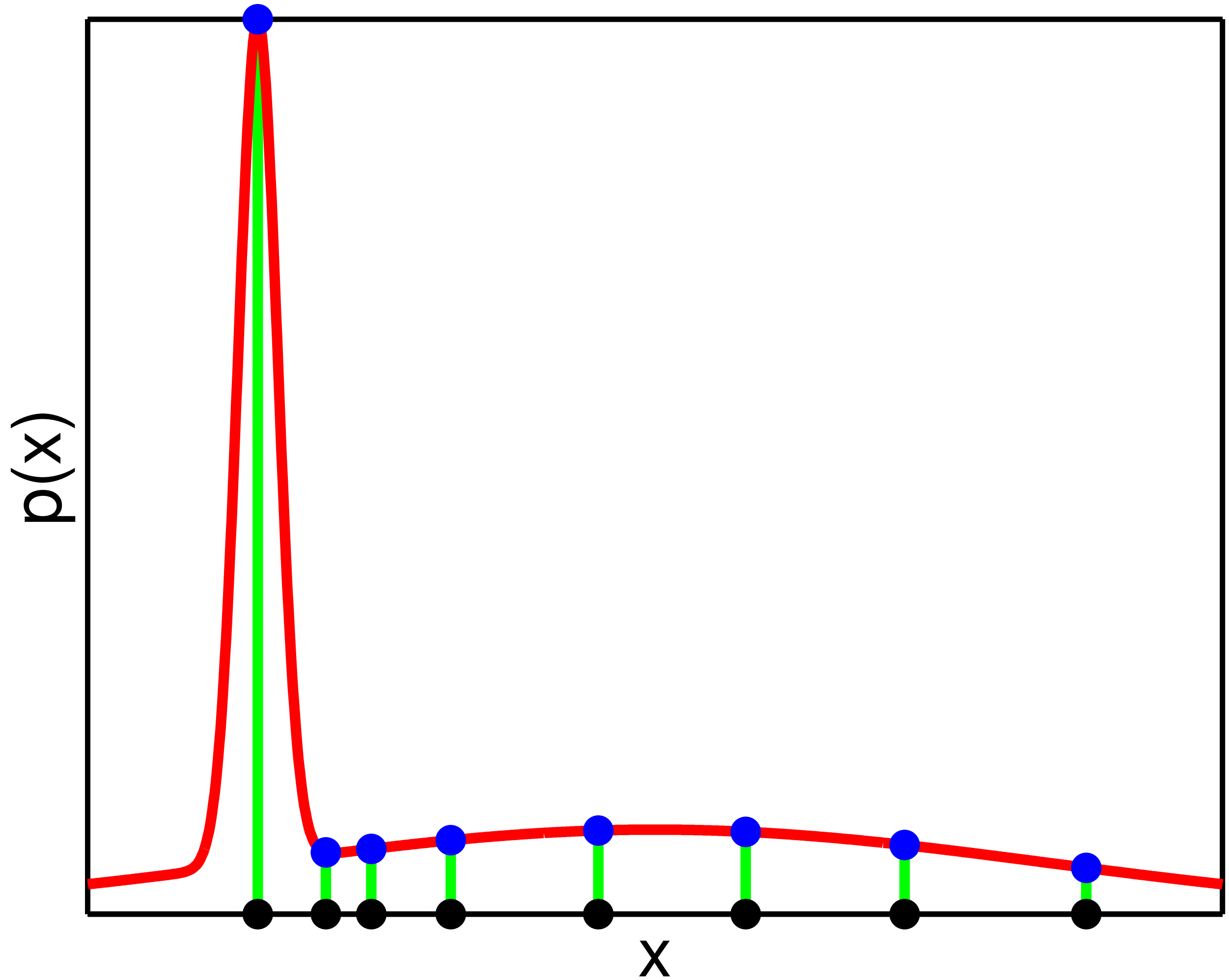


Figure Credit: Kevin Murphy

Hidden Data Causes Problems #3

- Likelihood has singularities if one Gaussian "collapses"



Special case: spherical Gaussians and hard assignments

- If P(X|Z=k) is spherical, with same σ for all classes:

$$P(\mathbf{x}_i \mid z = j) \propto \exp\left[-\frac{1}{2\sigma^2}\left\|\mathbf{x}_i - \mu_j\right\|^2\right]$$

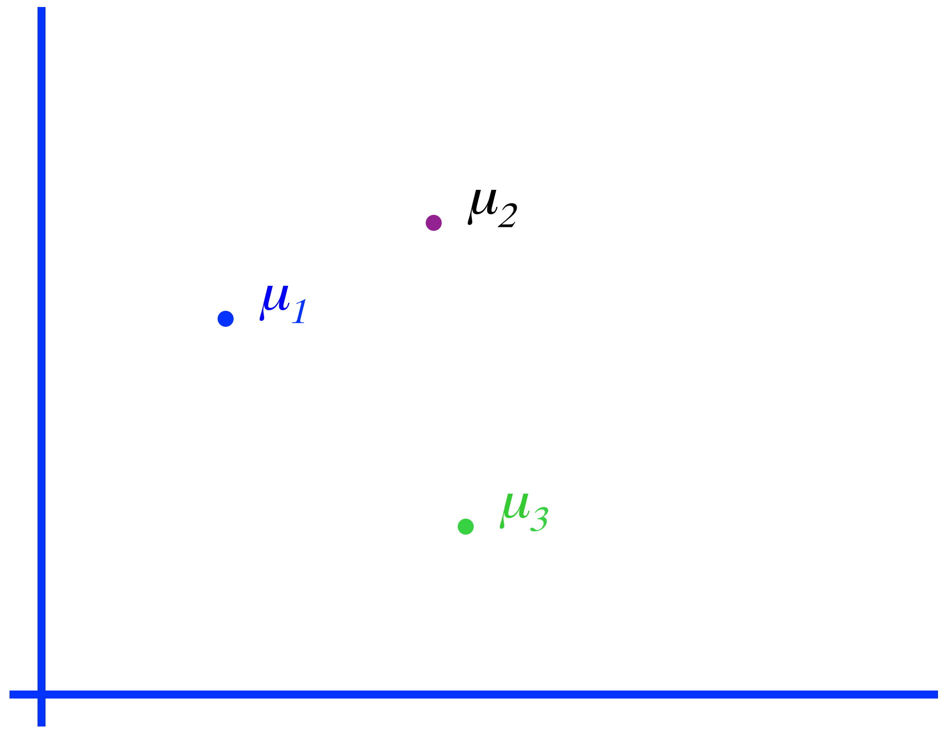
- If each x_i belongs to one class C(i) (hard assignment), marginal likelihood:

$$\prod_{i=1}^{N}\sum_{j=1}^{k} P(\mathbf{x}_i, y = j) \propto \prod_{i=1}^{N} \exp\left[-\frac{1}{2\sigma^2}\left\|\mathbf{x}_i - \mu_{C(i)}\right\|^2\right]$$

- M(M)LE same as K-means!!!

The K-means GMM assumption

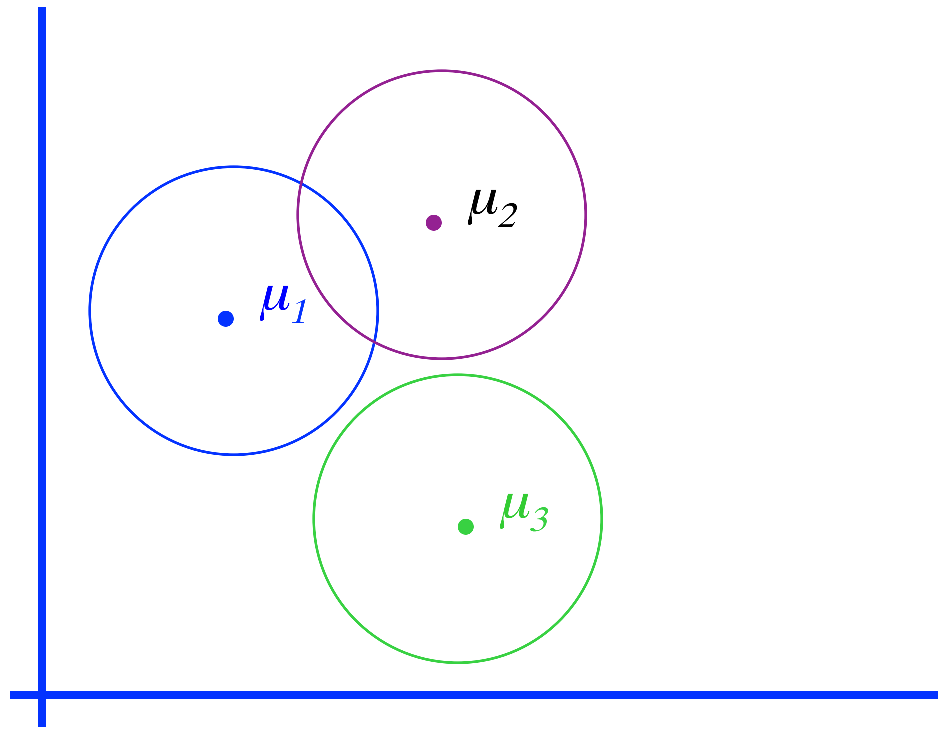
- There are k components
- Component *i* has an associated mean vector μ_ι



The K-means GMM assumption

- There are k components
- Component *i* has an associated mean vector μ_ι
- Each component generates data from a Gaussian with mean m_i and covariance matrix $\sigma^2 \boldsymbol{I}$

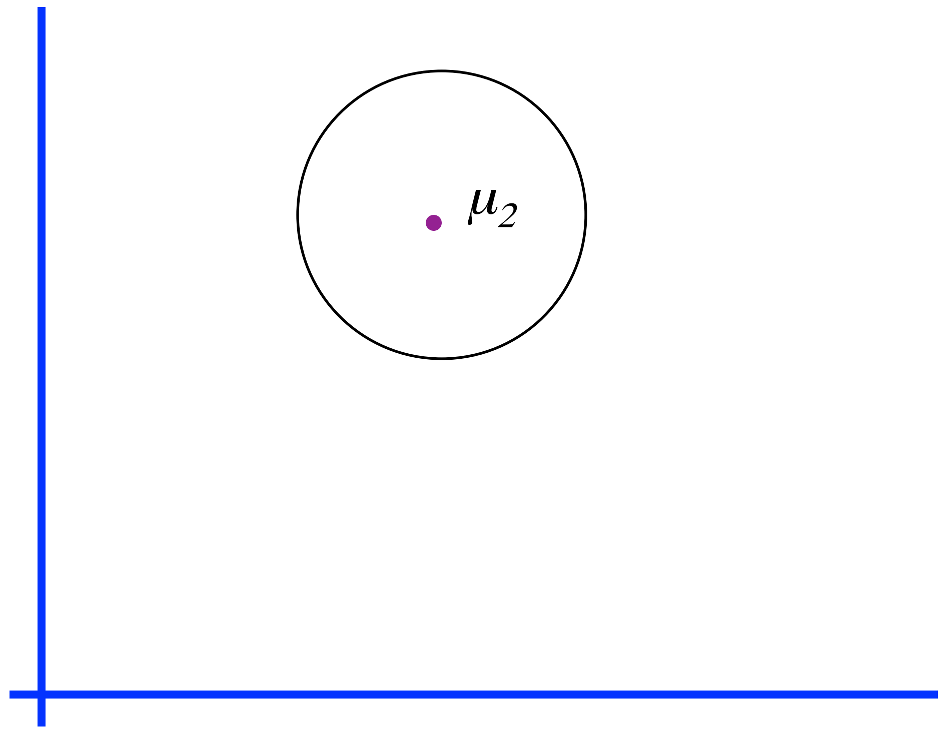
Each data point is generated according to the following recipe:



The K-means GMM assumption

- There are k components
- Component *i* has an associated mean vector μ_ι
- Each component generates data from a Gaussian with mean m_i and covariance matrix $\sigma^2 I$

Each data point is generated according to the following recipe:

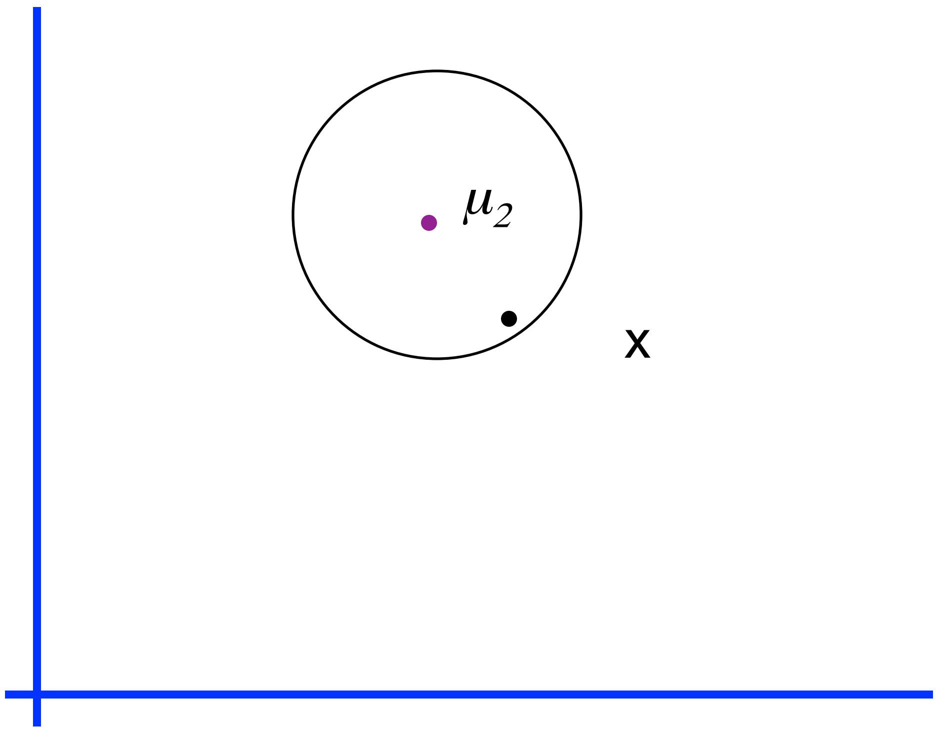


1. Pick a component at random: Choose component i with probability *P(y=i)*

The K-means GMM assumption

- There are k components
- Component *i* has an associated mean vector μ_ι
- Each component generates data from a Gaussian with mean m_i and covariance matrix $\sigma^2 I$

Each data point is generated according to the following recipe:

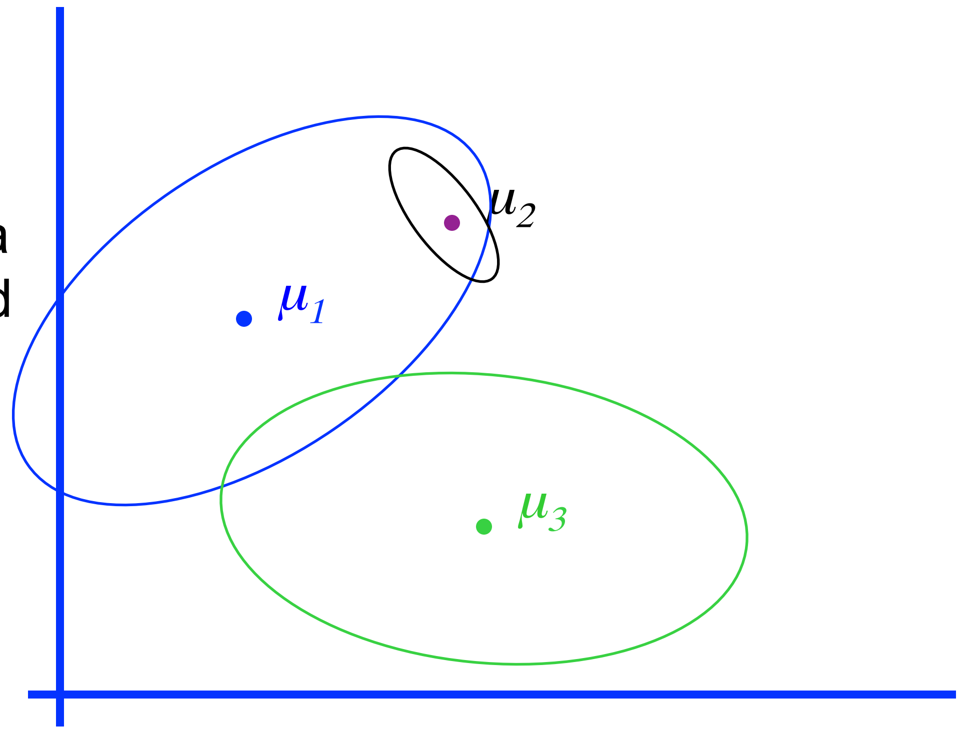


1. Pick a component at random: Choose component i with probability *P(y=i)*
2. Datapoint $\sim N(\mu_\iota, \sigma^2 I)$

The General GMM assumption

- There are k components
- Component *i* has an associated mean vector μ_ι
- Each component generates data from a Gaussian with mean m_i and covariance matrix Σ_i

Each data point is generated according to the following recipe:



1. Pick a component at random: Choose component i with probability *P(y=i)*
2. Datapoint $\sim \mathrm{N}(\mu_\iota, \Sigma_i)$

K-means vs GMM

- K-Means
  - http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

- GMM
  - http://www.socr.ucla.edu/applets.dir/mixtureem.html

EM

- Expectation Maximization [Dempster ‘77]

- Often looks like “soft” K-means

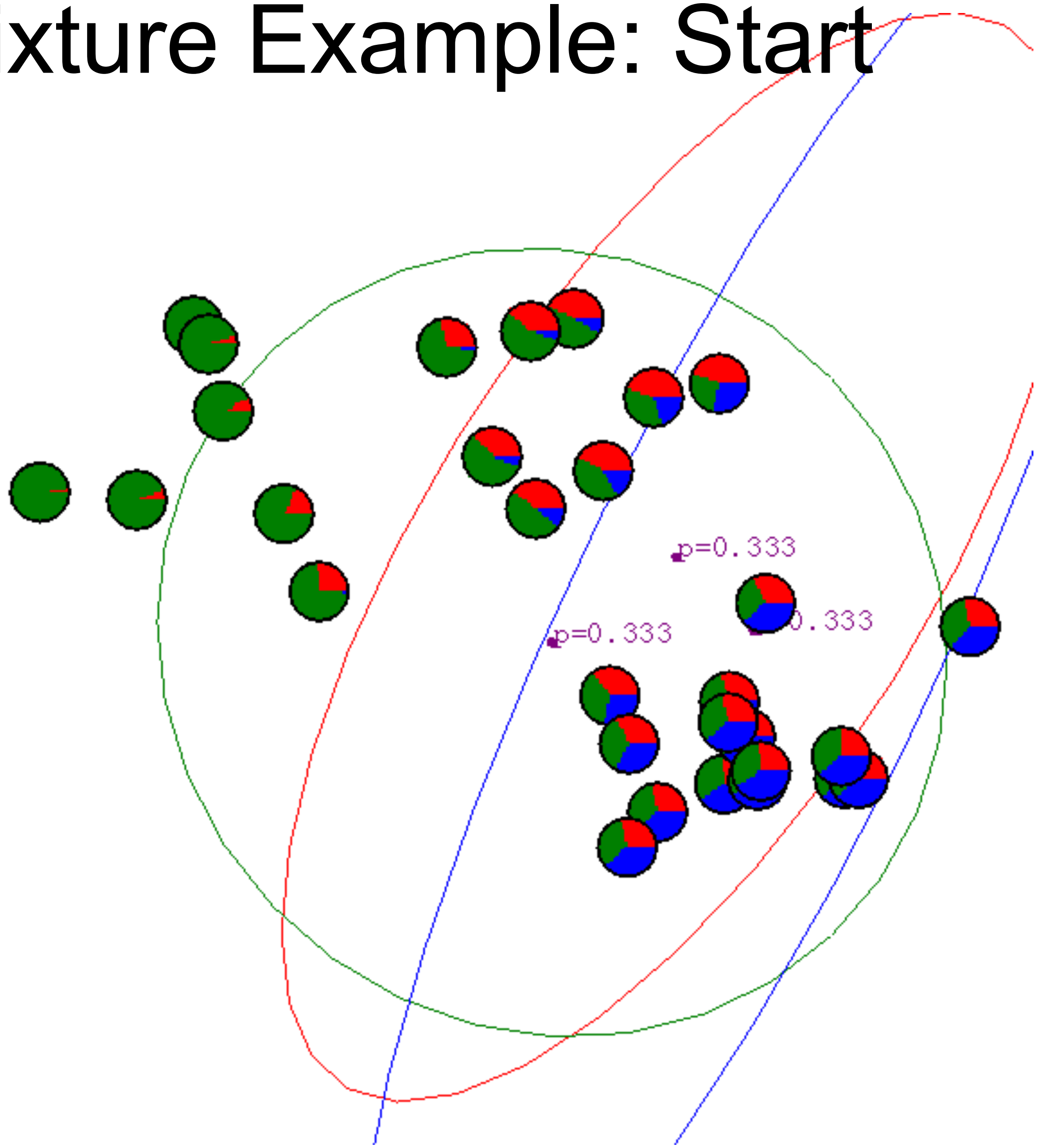
- Extremely general
- Extremely useful algorithm
  - Essentially THE goto algorithm for unsupervised learning

- Plan
  - EM for learning GMM parameters
  - EM for general unsupervised learning problems

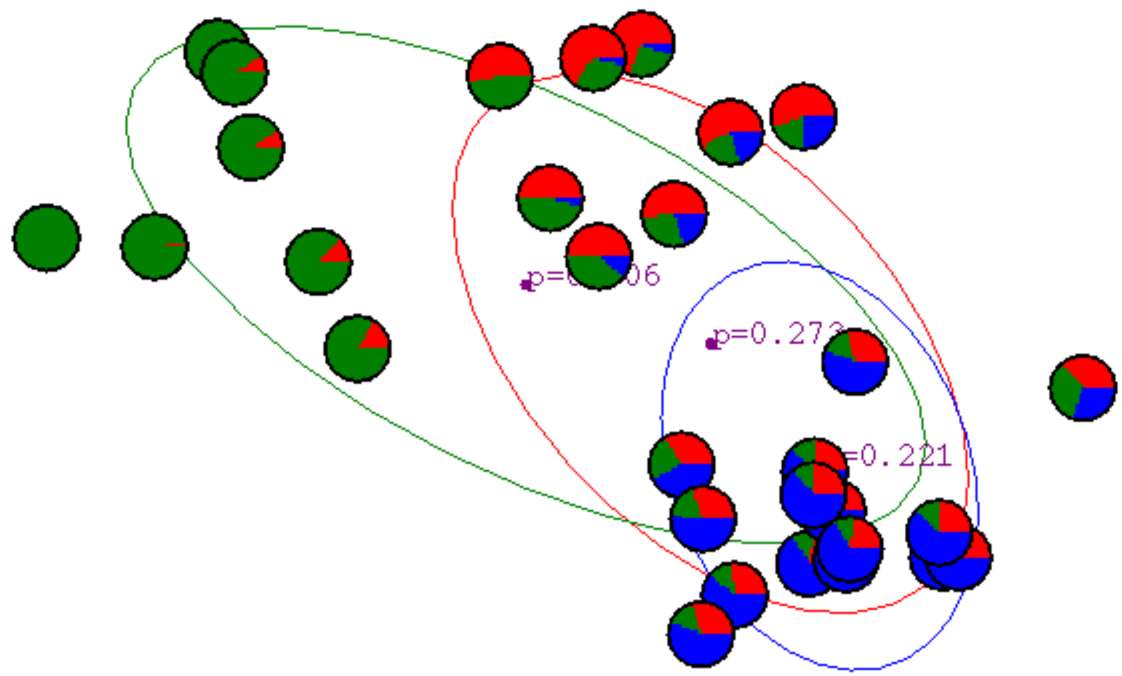
EM for Learning GMMs

- Simple Update Rules
  - E-Step: estimate $P(z_i = j \mid x_i)$
  - M-Step: maximize full likelihood weighted by posterior

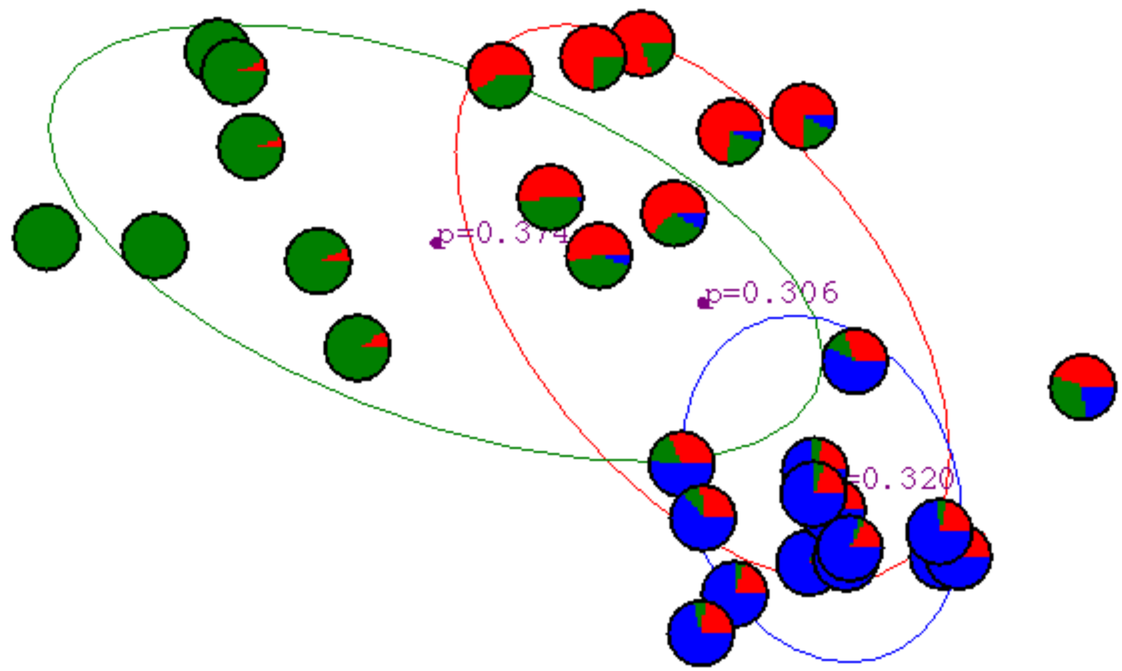
Gaussian Mixture Example: Start



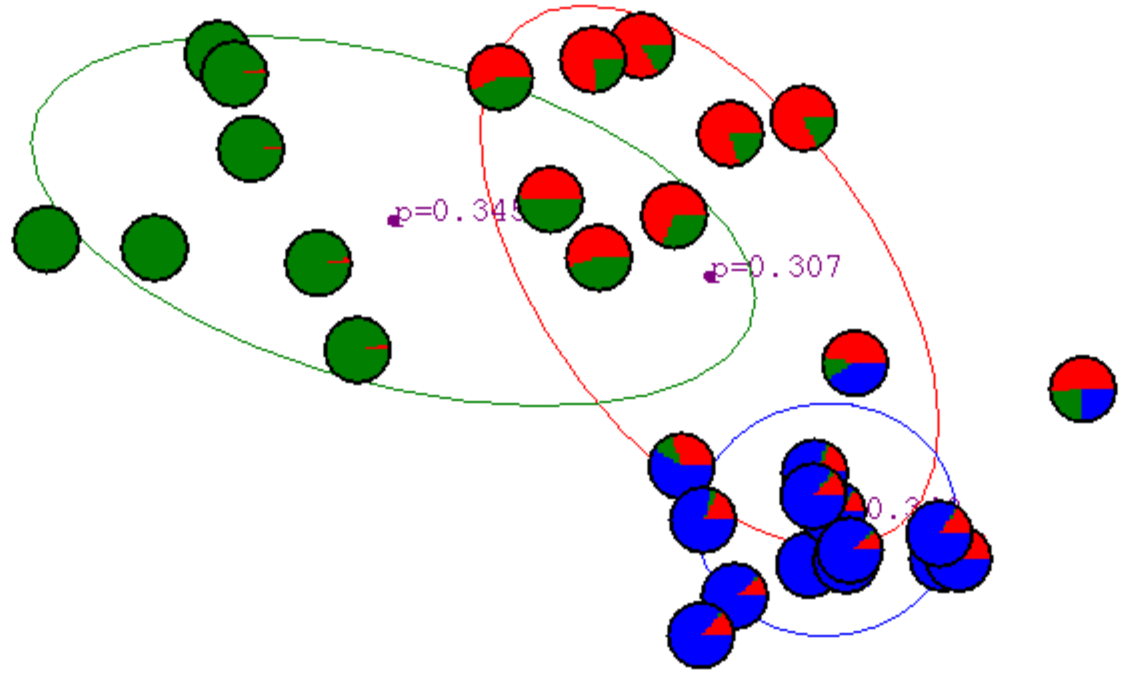
After 1st iteration



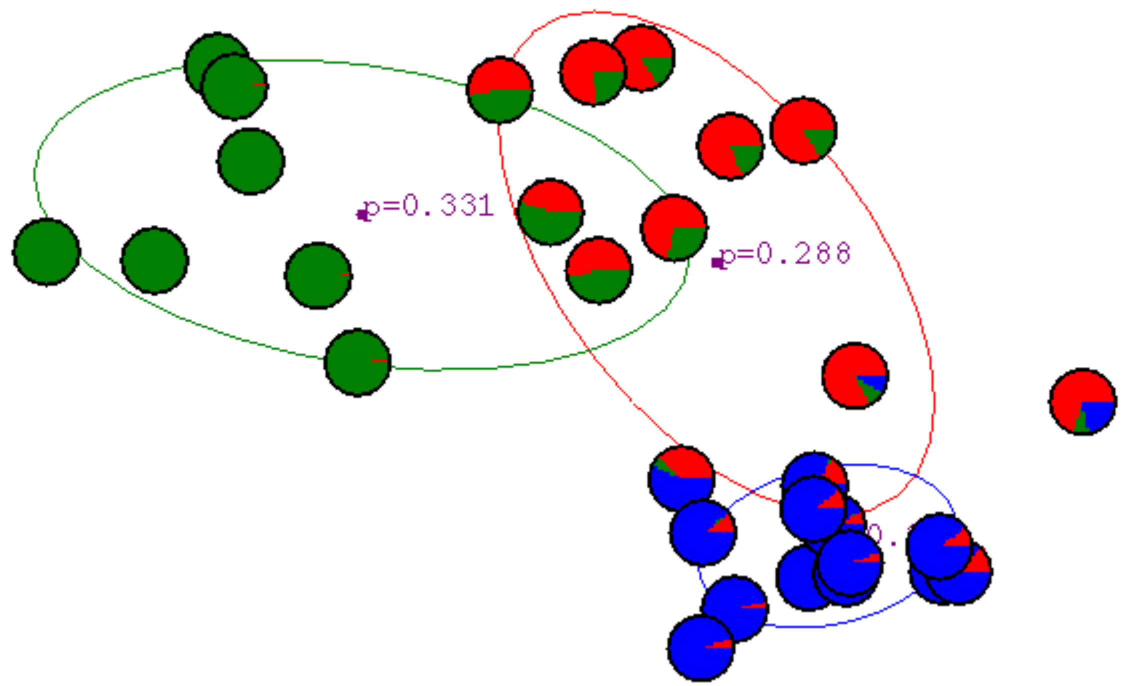
After 2nd iteration



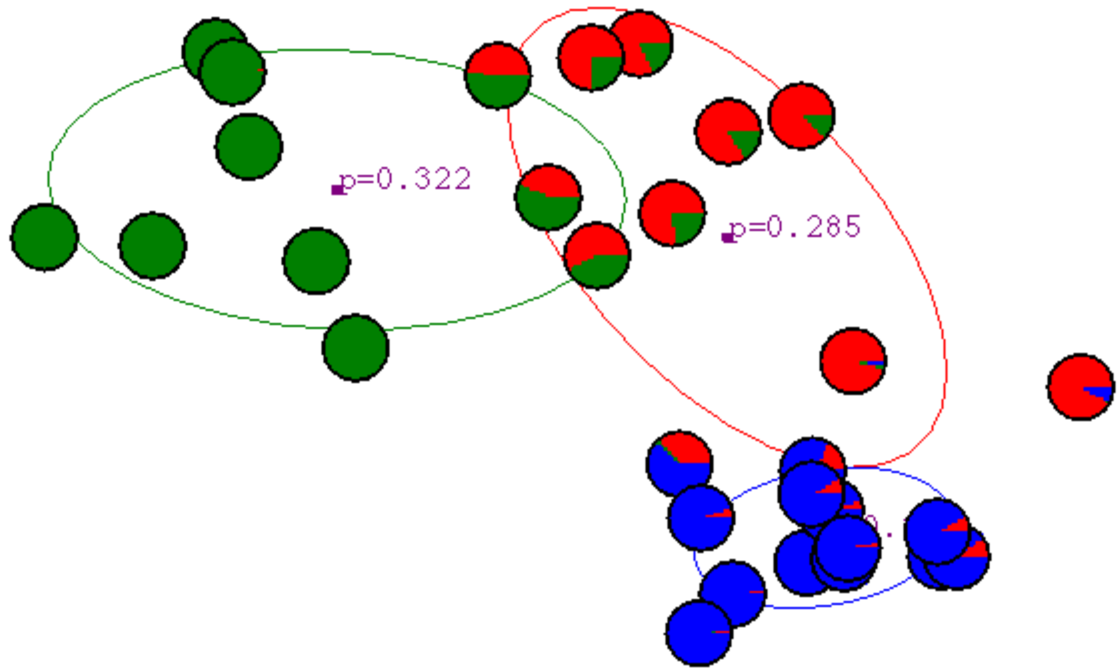
After 3rd iteration



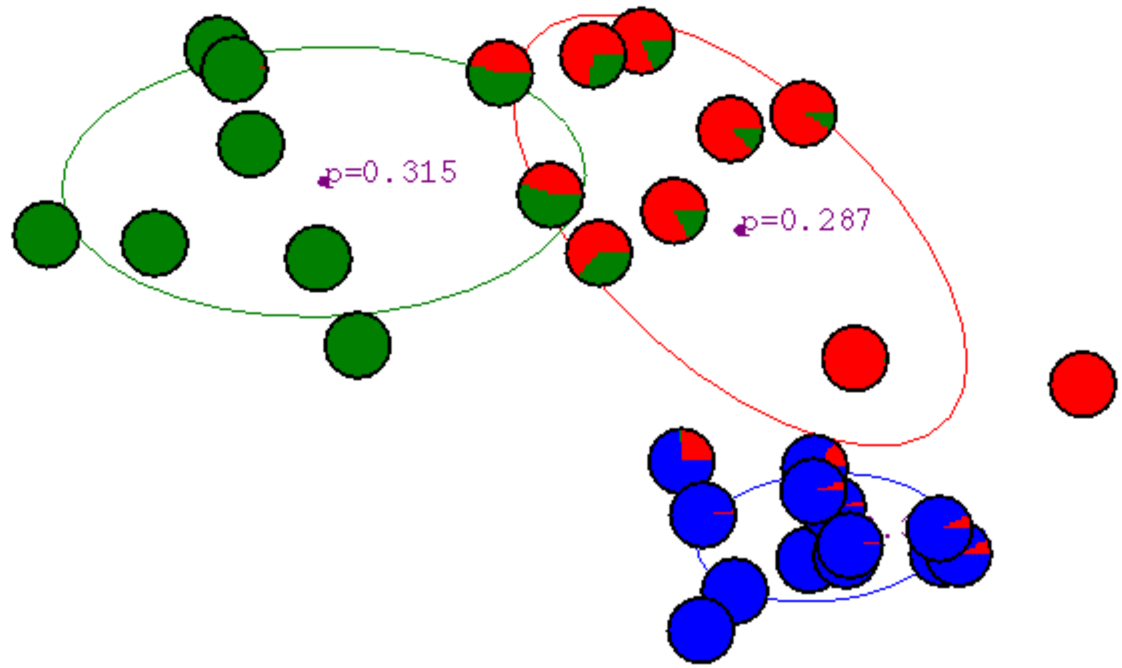
After 4th iteration



After 5th iteration



After 6th iteration



After 20th iteration