Clustering
K-means

Machine Learning – CSEP546

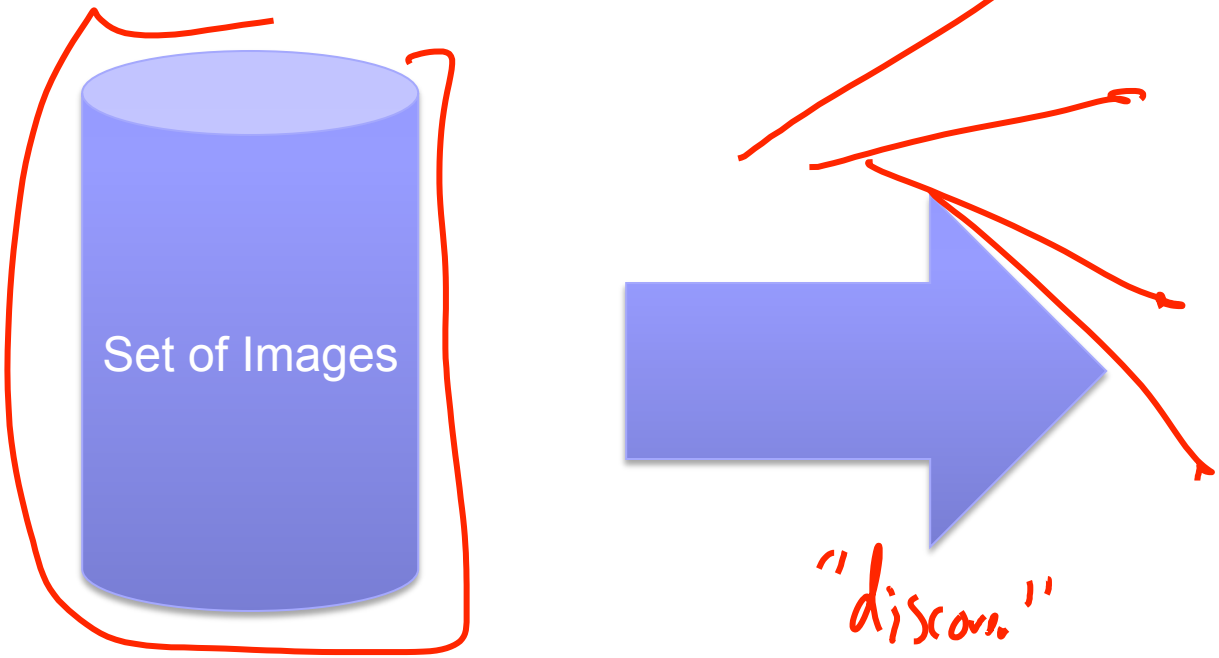
Carlos Guestrin

University of Washington

February 18, 2014

Clustering images

Unsupervised => no given labels

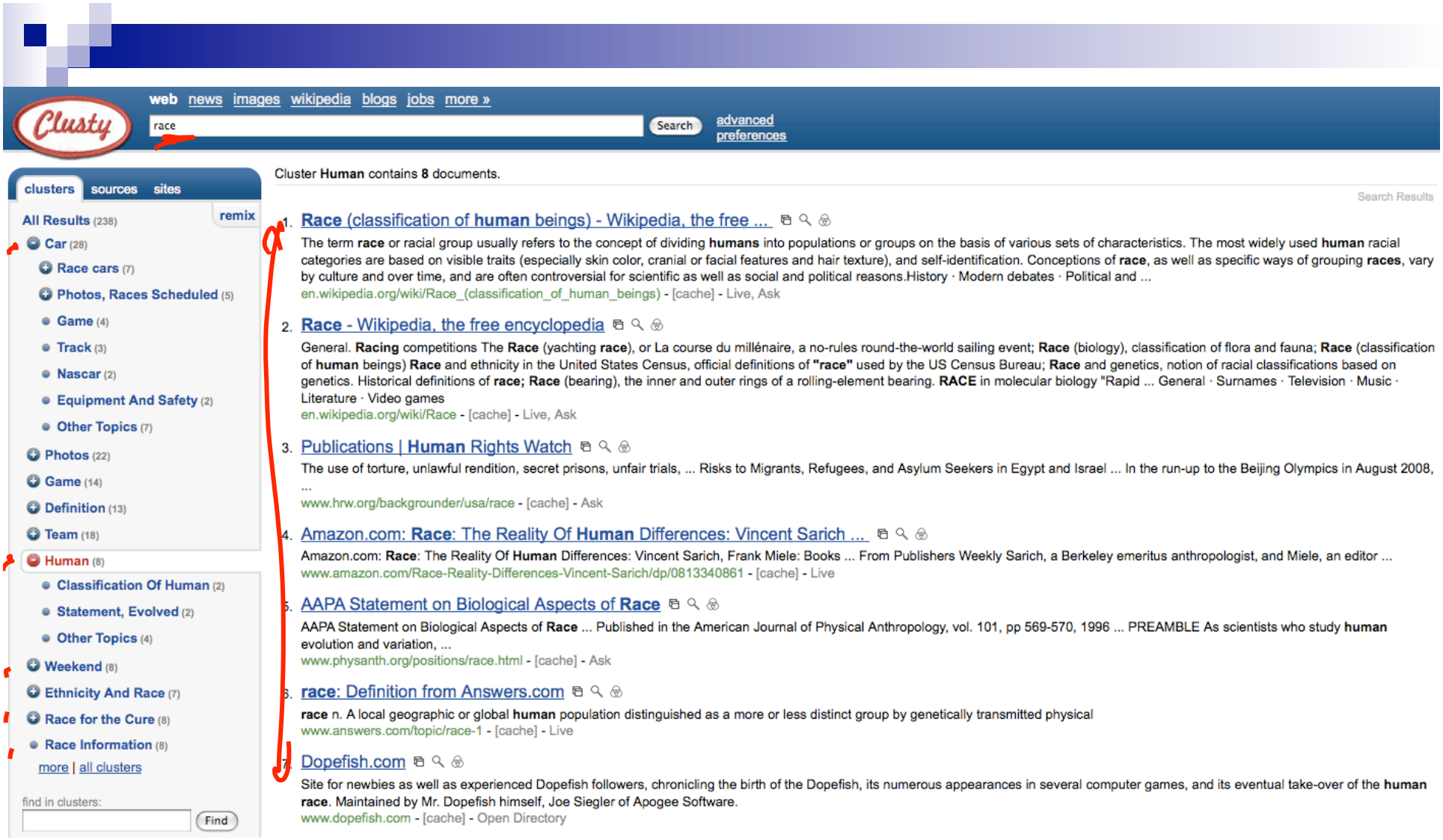


"discover"

groups of image

[Goldberger et al.]

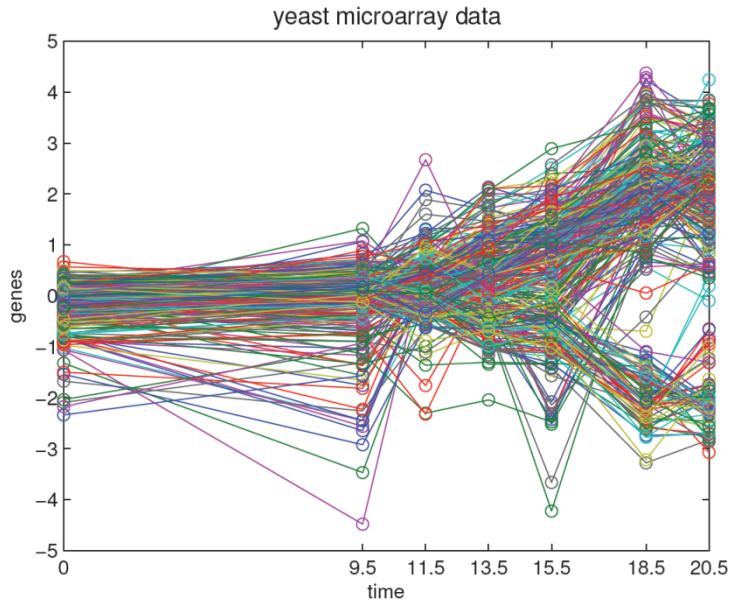
Clustering web search results

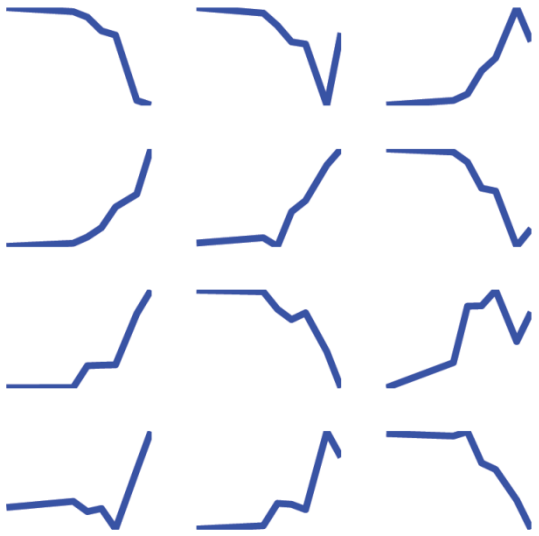


Example

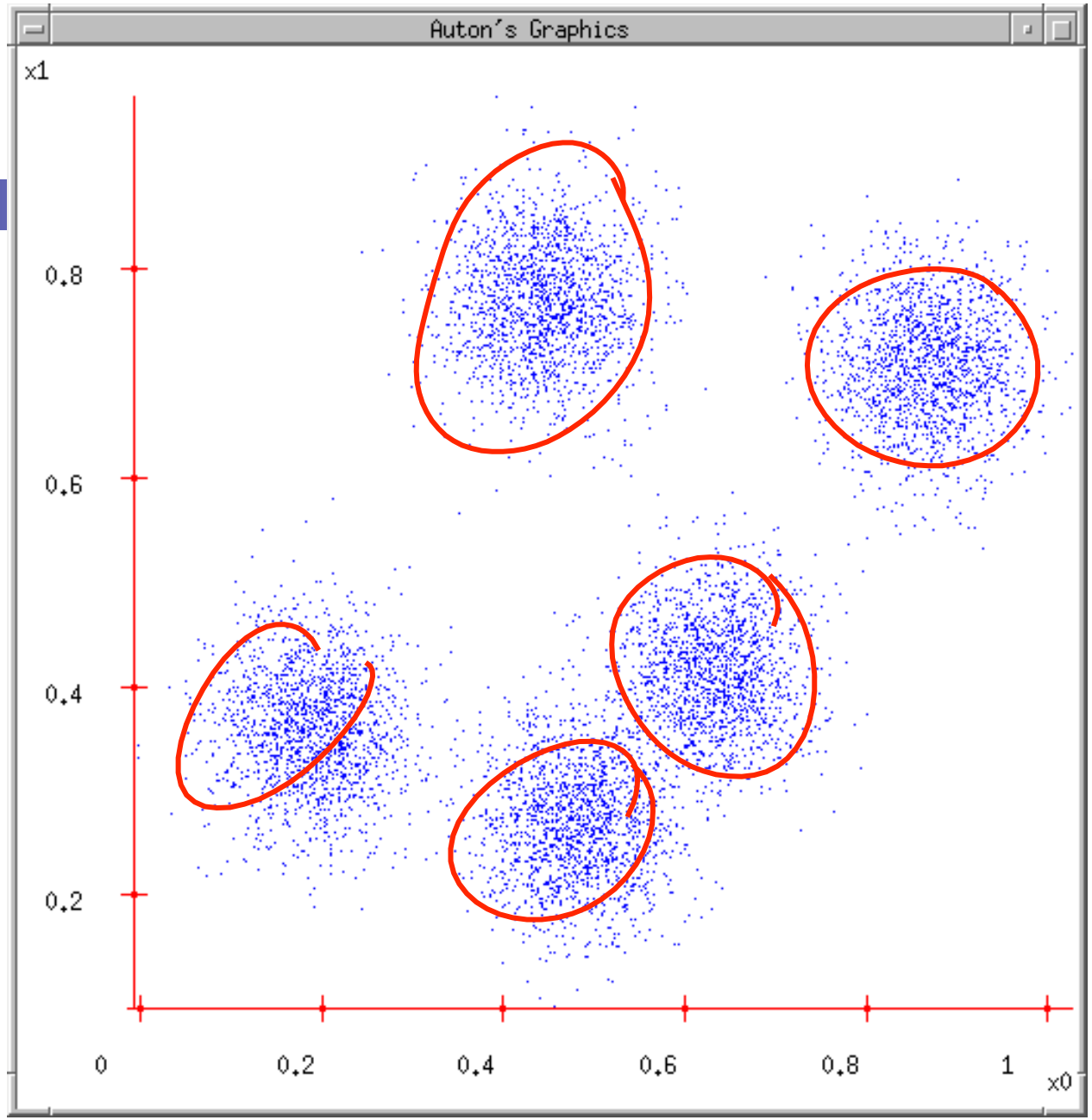
(Taken from Kevin Murphy's ML textbook)

- Data: gene expression levels
- Goal: cluster genes with similar expression trajectories



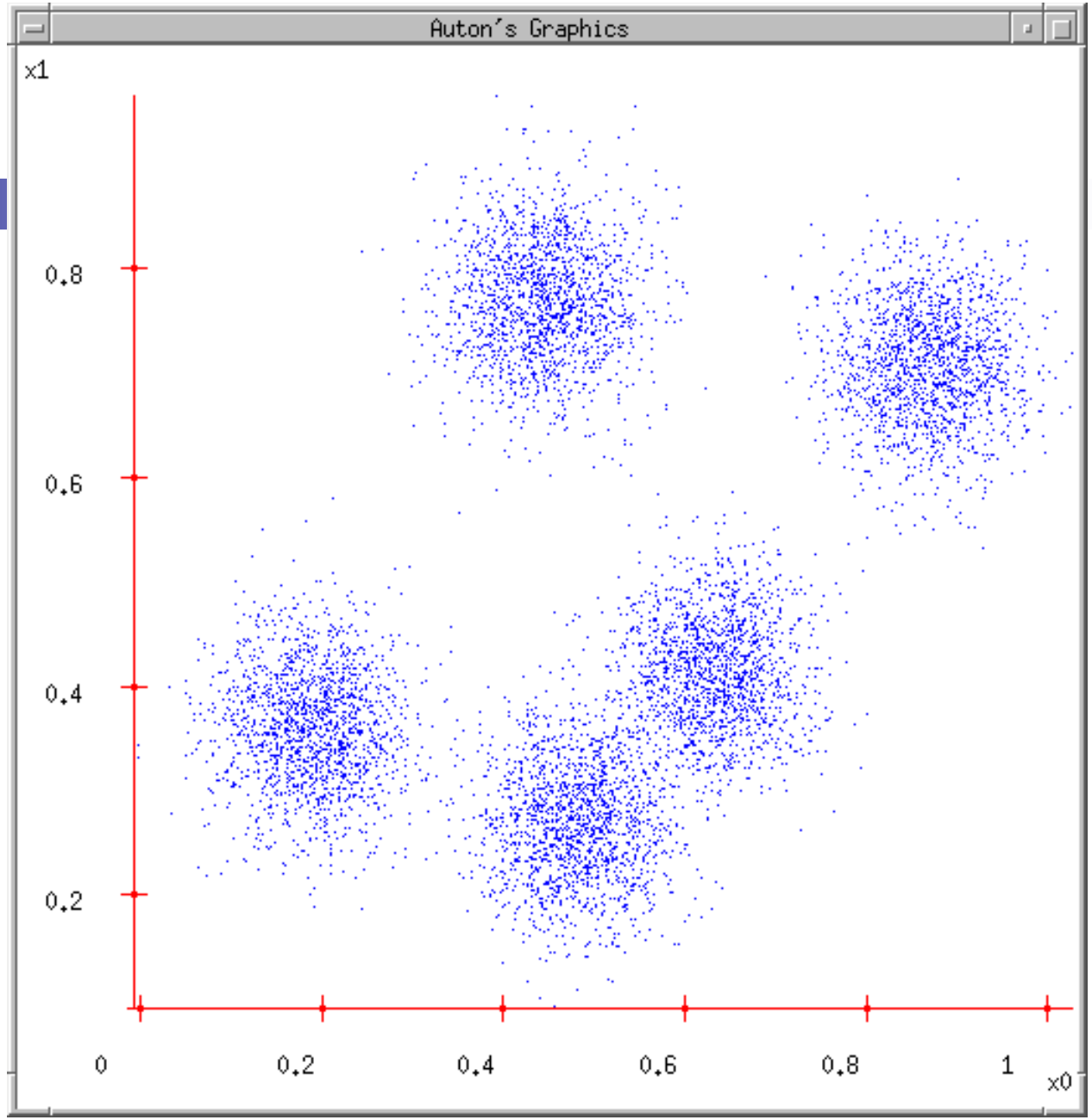


Some Data



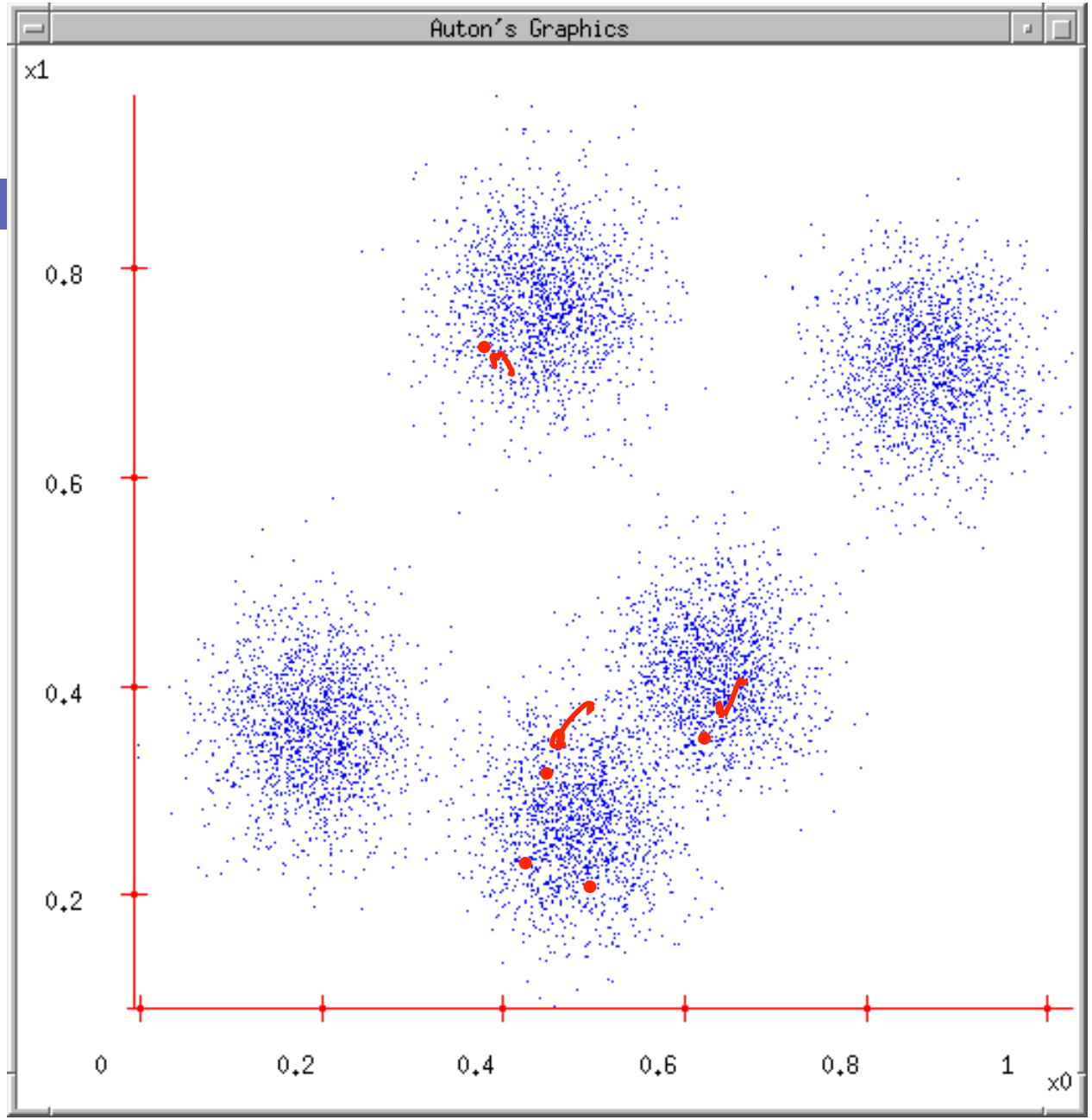
K-means

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



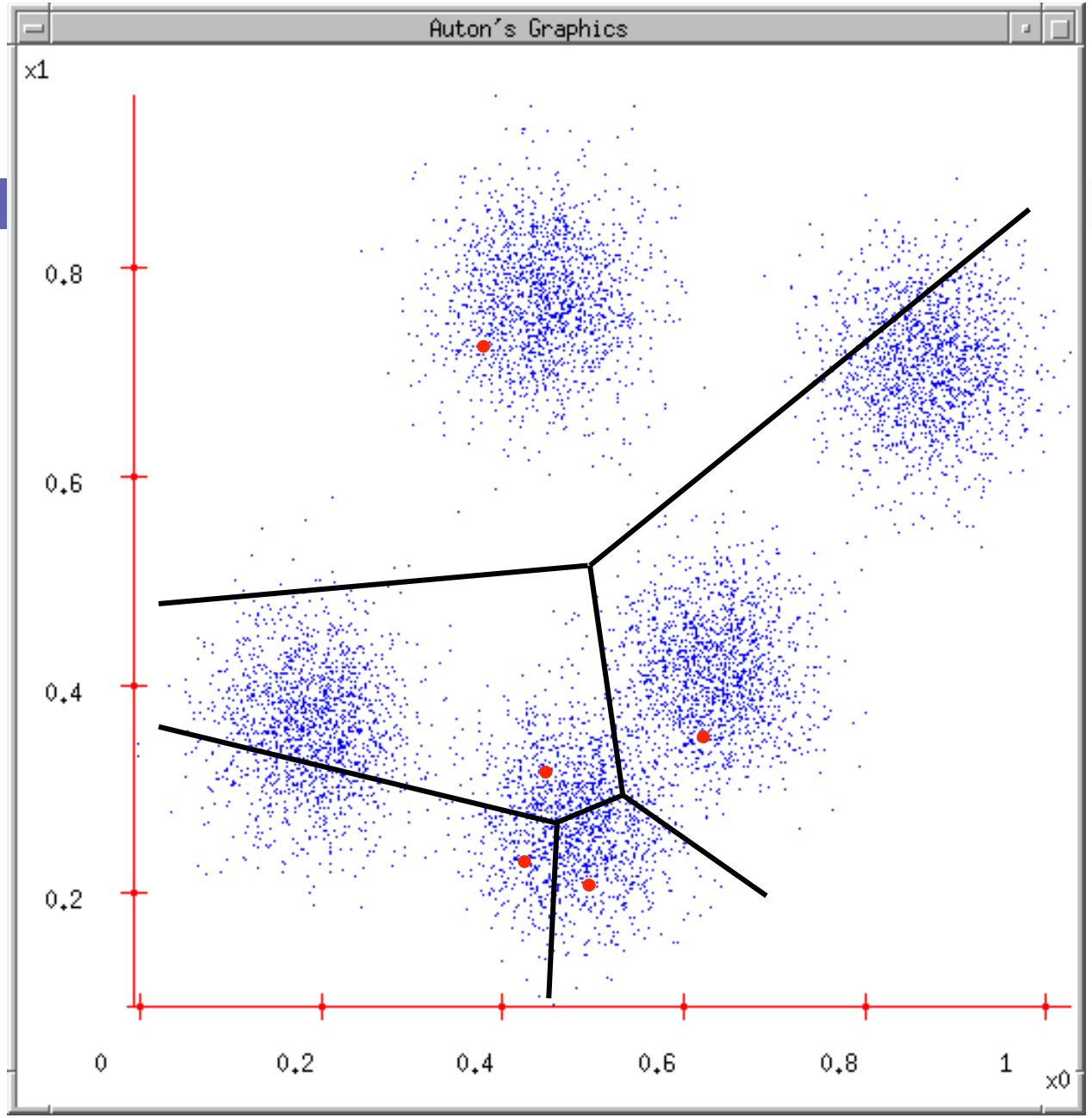
K-means

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



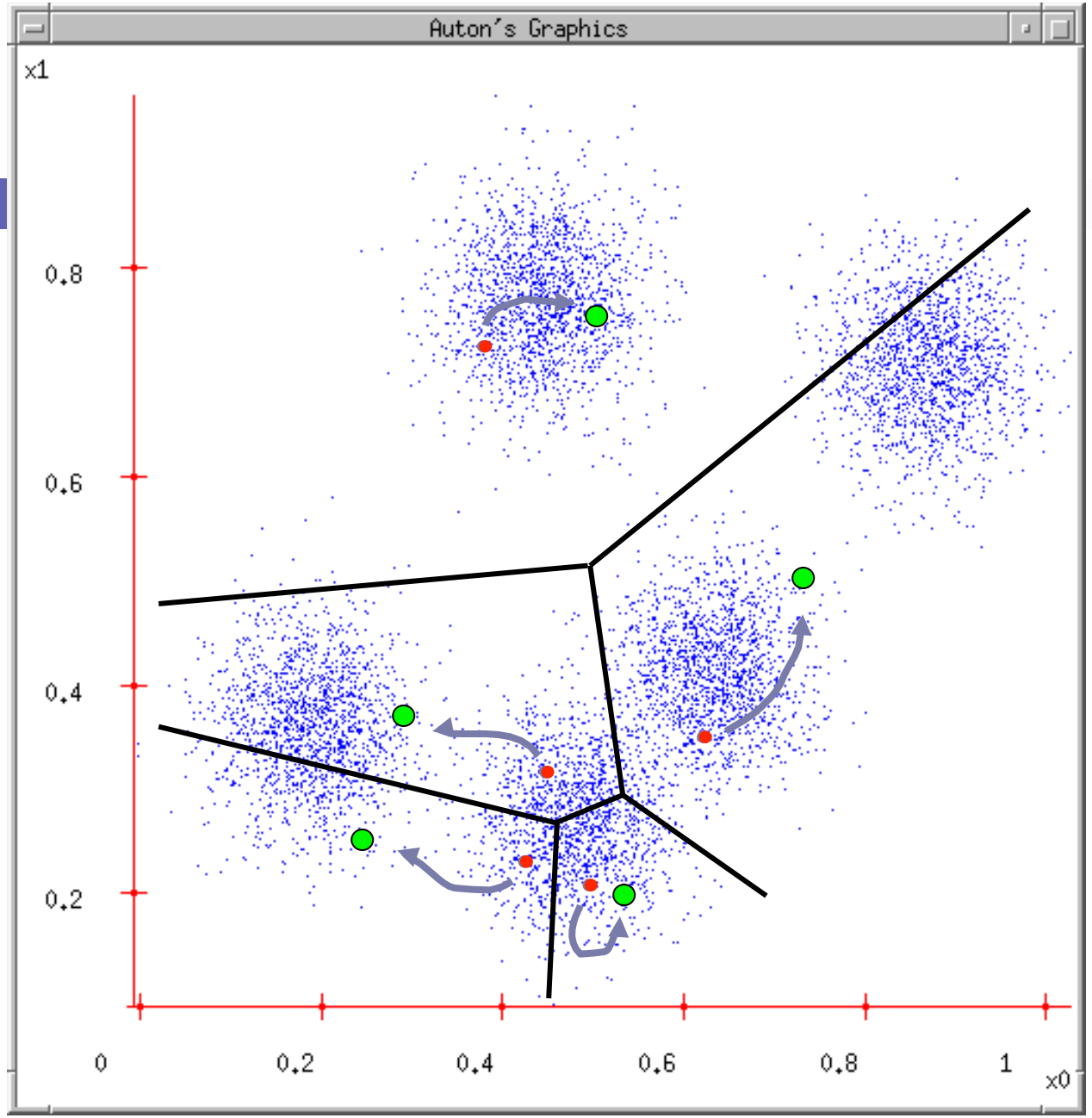
K-means

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)



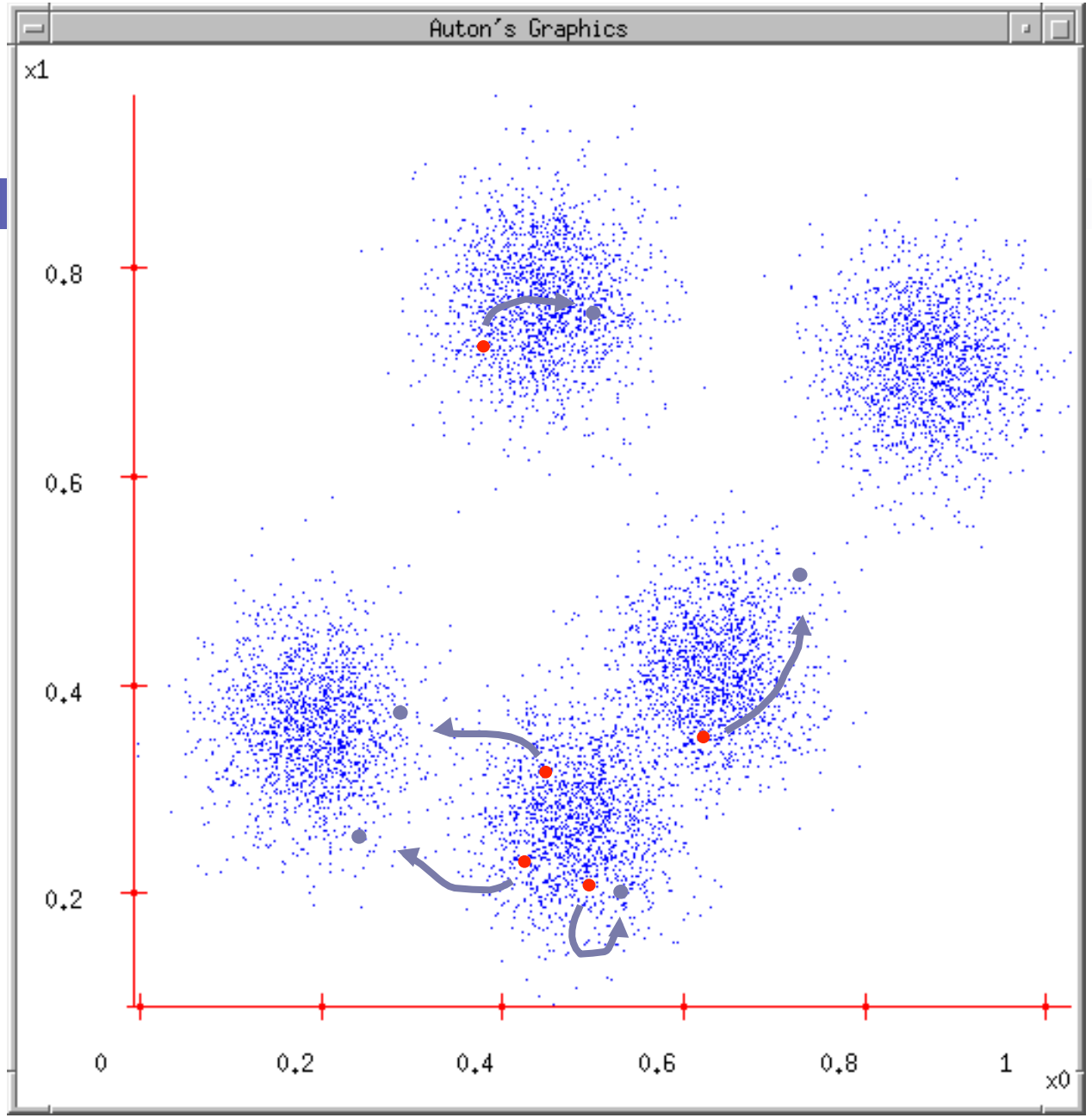
K-means

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
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

- **Classify**: Assign each point j∈{1,…N} to nearest center:
  - $C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$

- **Recenter**: μ_i becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C(j)=i} ||\mu - x_j||^2$
  - Equivalent to μ_i ← average of its points!

What is K-means optimizing?

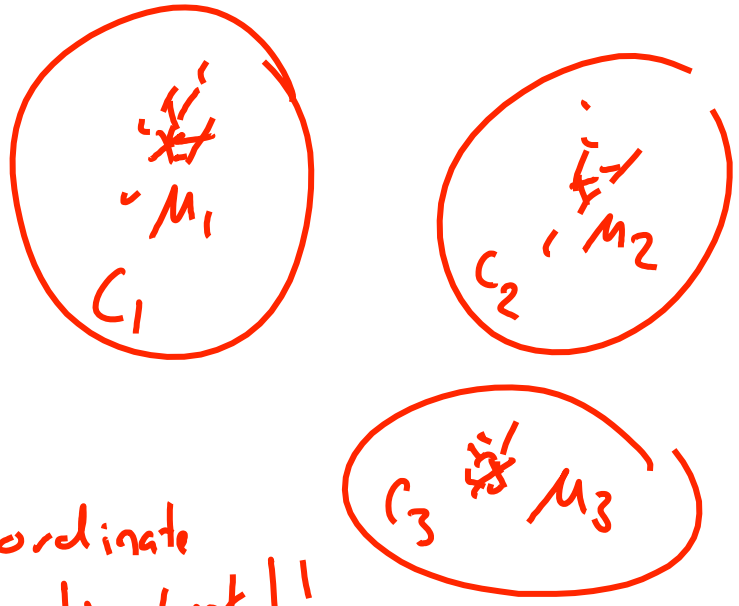
or loss function

- Potential function $F(\mu,C)$ of centers μ and point allocations C:
  - $$F(\mu, C) = \sum_{j=1}^{N} ||\mu_{C(j)} - x_j||^2$$

the center assigned to data point

- Optimal K-means:
  - $\min_\mu \min_C F(\mu,C)$

NP-hard, but we'll do coordinate descent & hope for the best!!



Does K-means converge??? Part 1

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Fix μ, optimize C

$\min_{C(1)} \min_{C(2)} \dots \min_{C(N)} \sum_{j=1}^{N} ||\mu_{C(j)} - x_j||^2$ ← indep optimizations per data point

$\Rightarrow$ For each j: $\min_{C(j)} ||\mu_{C(j)} - x_j||^2$ ← "classify" step

Does K-means converge??? Part 2

- Optimize potential function:

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Fix C, optimize μ

$$\min_{\mu_1} \min_{\mu_2} \dots \min_{\mu_k} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$ ← indep opt per cluster center

⇒ For each i:

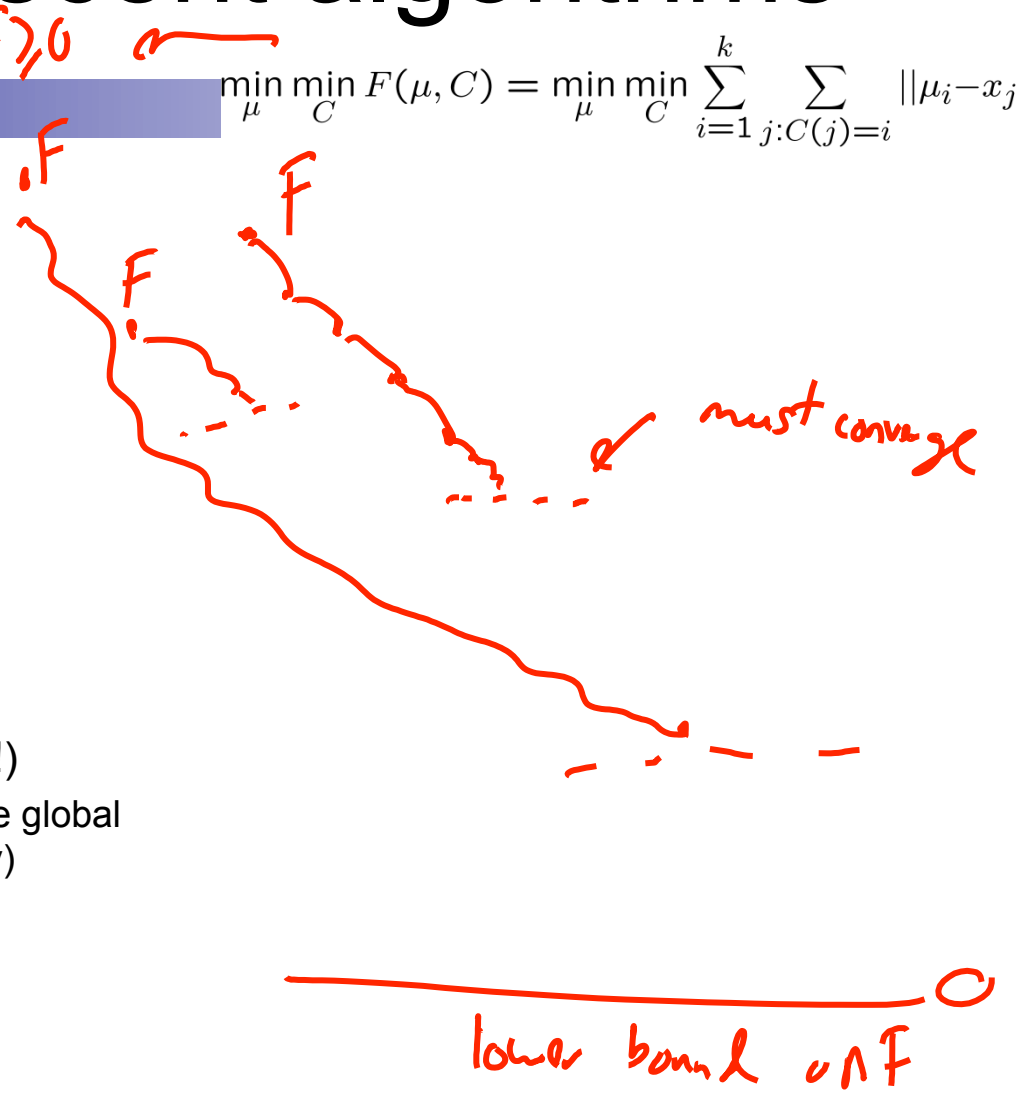
$$\min_{\mu_i} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$ ← "recenter"

⇒ For Euclidian norm, just average

Coordinate descent algorithms

$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Want: $\min_a \min_b F(a,b)$
- Coordinate descent:
  - fix a, minimize b
  - fix b, minimize a
  - repeat
- Converges!!!
  - if F is bounded
  - to a (often good) local optimum
    - as we saw in applet (play with it!)
      - (For LASSO it converged to the global optimum, because of convexity)



- K-means is a coordinate descent algorithm!

Mixtures of Gaussians

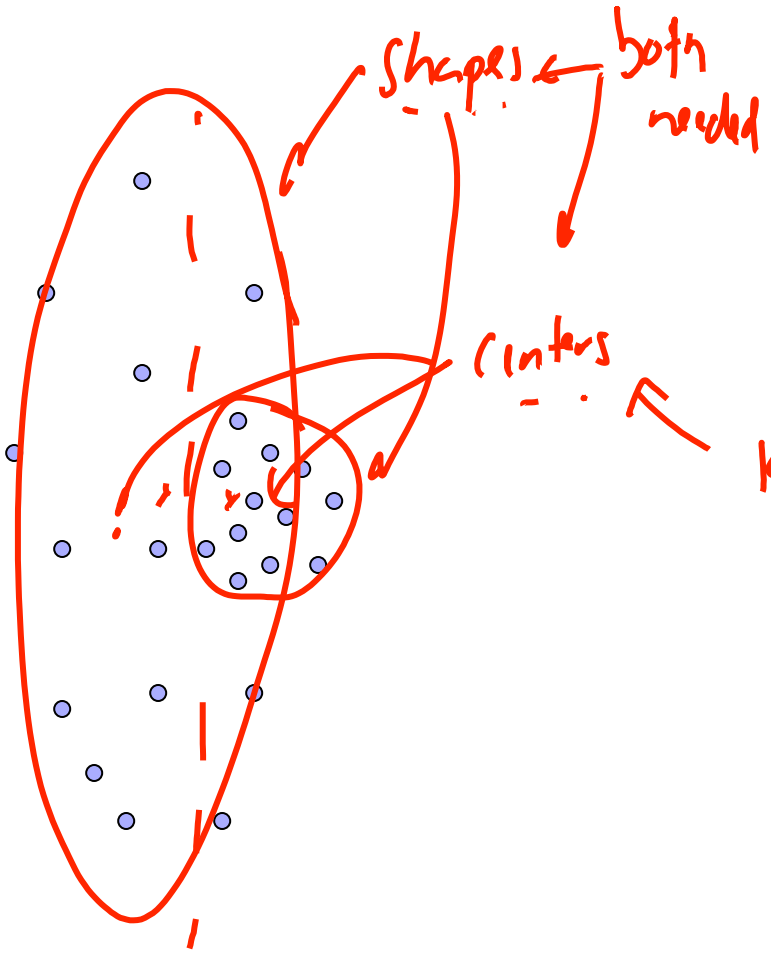
Machine Learning – CSEP546

Carlos Guestrin

University of Washington

February 18, 2014

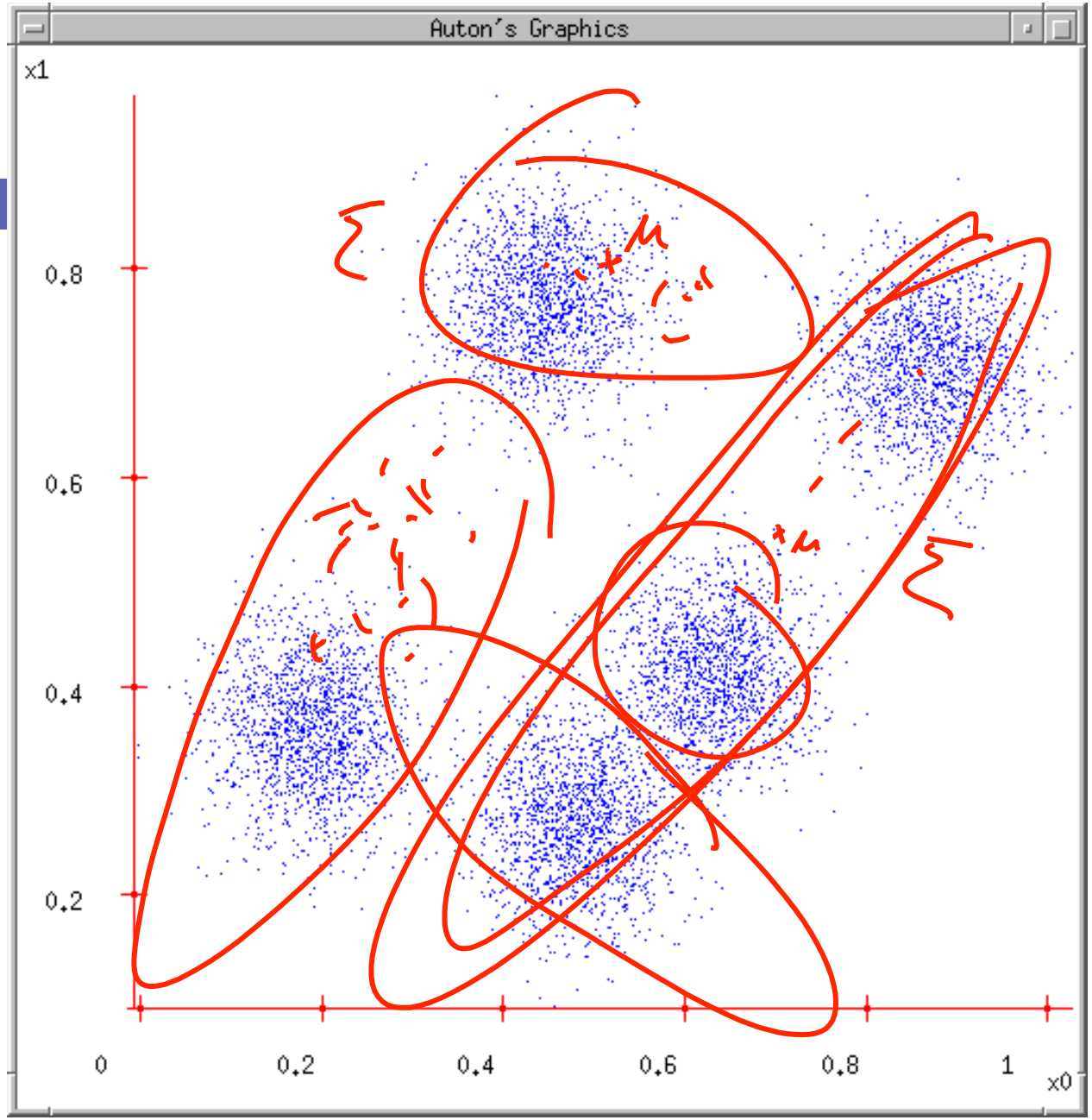
(One) bad case for k-means



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

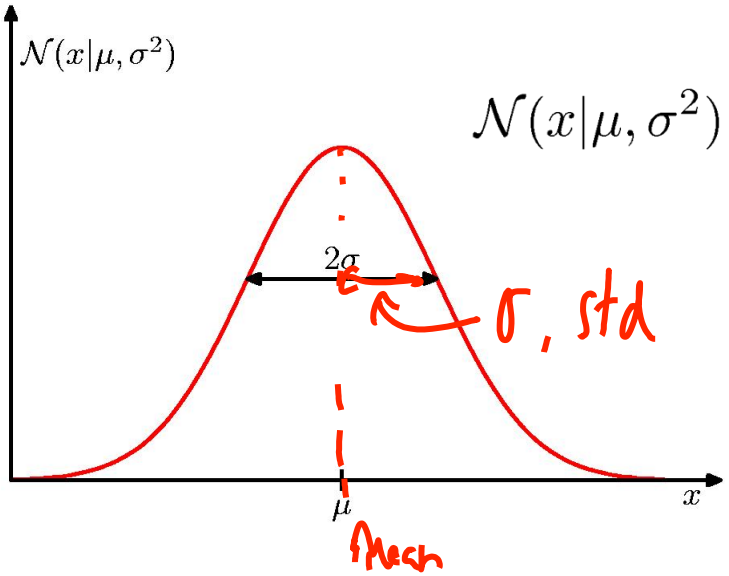
centers are not enough

Non-spherical data

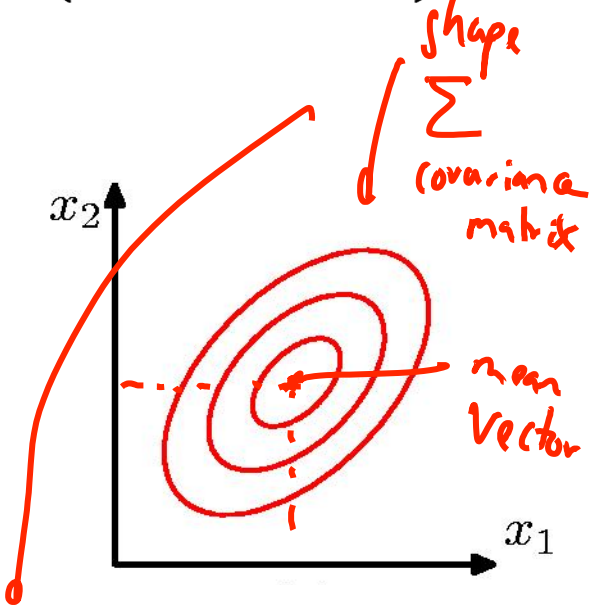


Quick Review of Gaussians

- Univariate and multivariate Gaussians

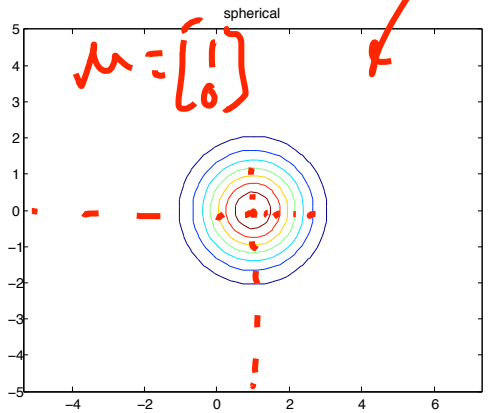


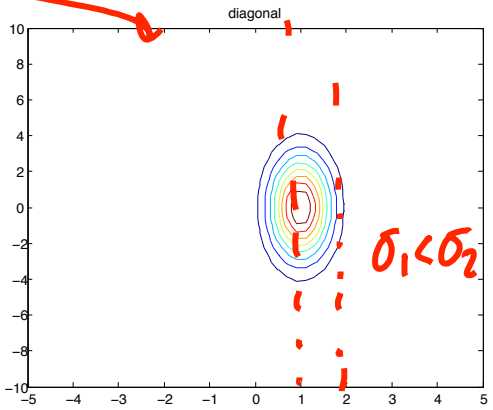
$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

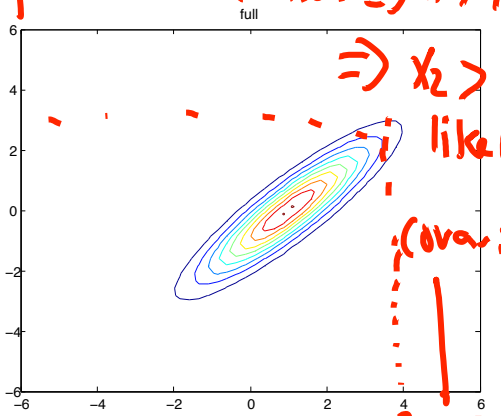


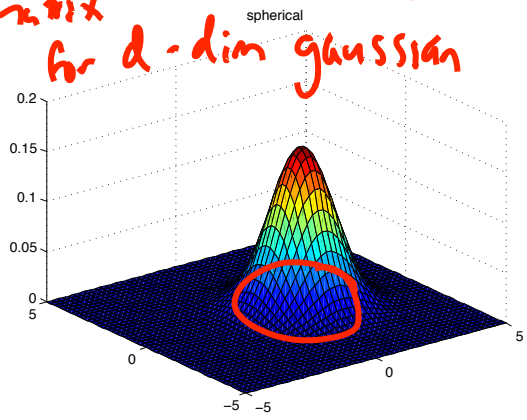
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

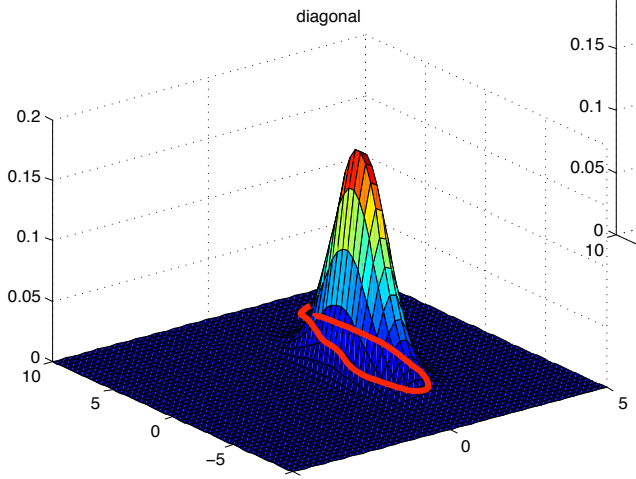
Two-Dimensional Gaussians

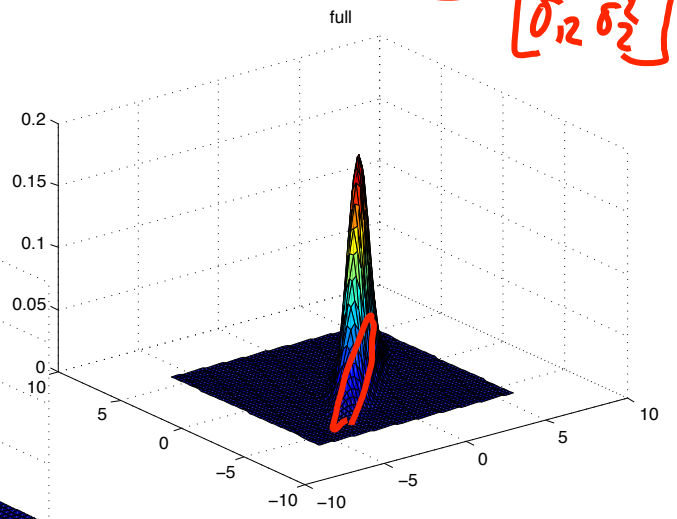






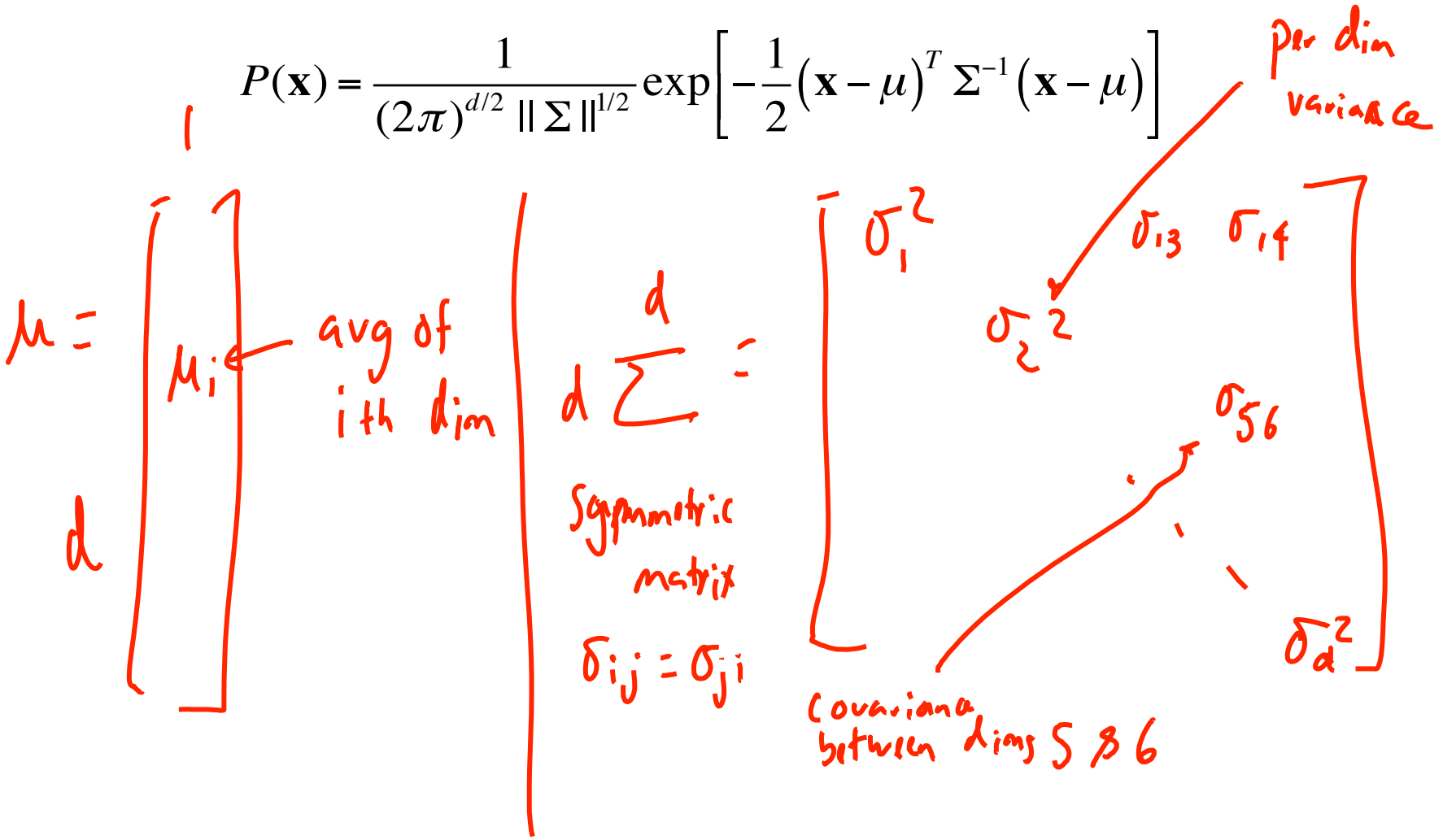






Gaussians in *d* Dimensions

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \|\Sigma\|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right]$$



Learning Gaussians

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \|\Sigma\|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right]$$

- Given data: $x^1, x^2, \ldots x^n$
- MLE for mean:

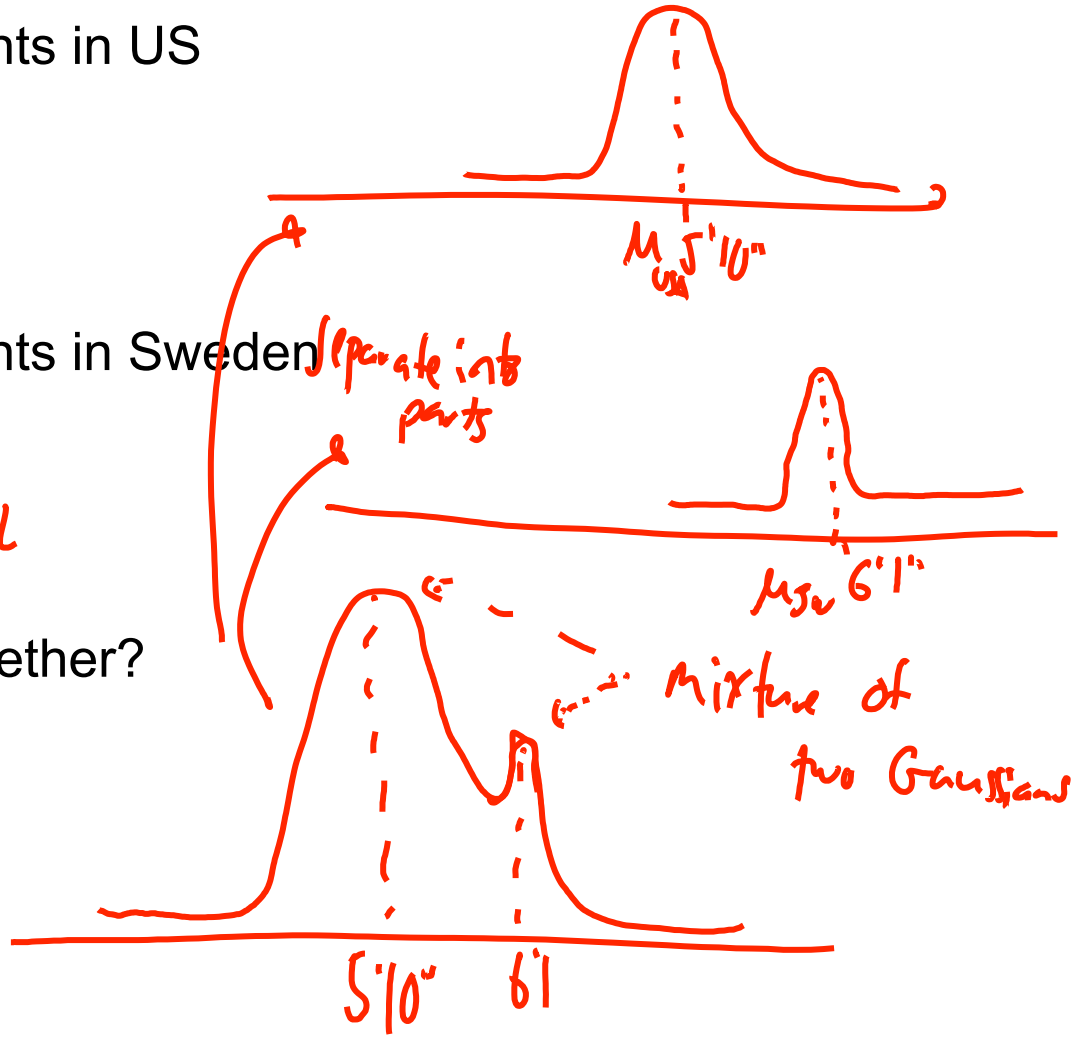
$$\mu = \frac{\sum_i x^i}{n}$$

- MLE for covariance:

$$\sigma_{ij} = \frac{1}{n} \sum_{u=1}^{n} \sum_{v=1}^{n} (\mu_i - x_i^u)(\mu_j - x_j^v)$$

When the world is not Gaussian

- Distribution of male heights in US
- Distribution of male heights in Sweden
- What if we mix these together?



GOAL

pop USA: 300M

pop Sweden: 10M

Gaussian Mixture Model

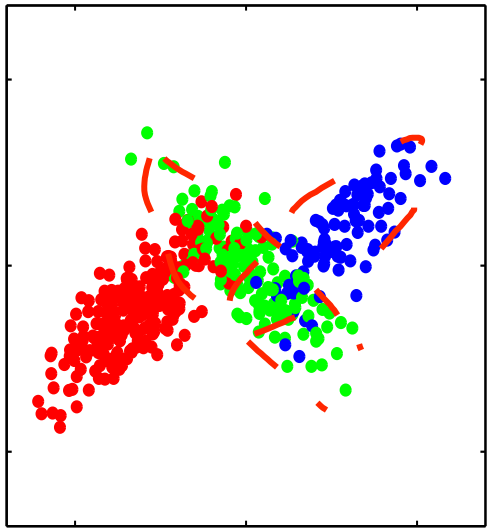
- Most commonly used mixture model
- Observations: $x^1, \ldots x^n$

  k · Gaussians
- Parameters:

  $\mu_i, \Sigma_i \quad i = 1 \ldots k$

  $z^j \in \{1, 2, \ldots k\}$
- Cluster indicator: z^j ← which Gaussian a point comes from
- Per-cluster likelihood:

  Learn a Gaussian per cluster, if we had z^j



- Ex. z^i = country of origin, x^i = height of ith person
  - k^{th} mixture component = distribution of heights in country *k*

Generative Model

$\pi_{USA} = \frac{300M}{310M}$

$\pi_{Sweden} = \frac{10M}{310M}$

- We can think of *sampling* observations from the model

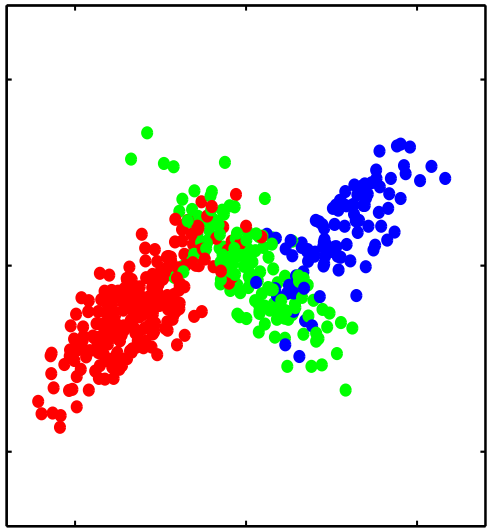
- For each observation *i*,
  - Sample a cluster assignment

z^i sampled with prob $\pi_1, \pi_2, \dots, \pi_k$

  - Sample the observation from the selected Gaussian

$x^i \sim N(\mu_i, \Sigma_i)$

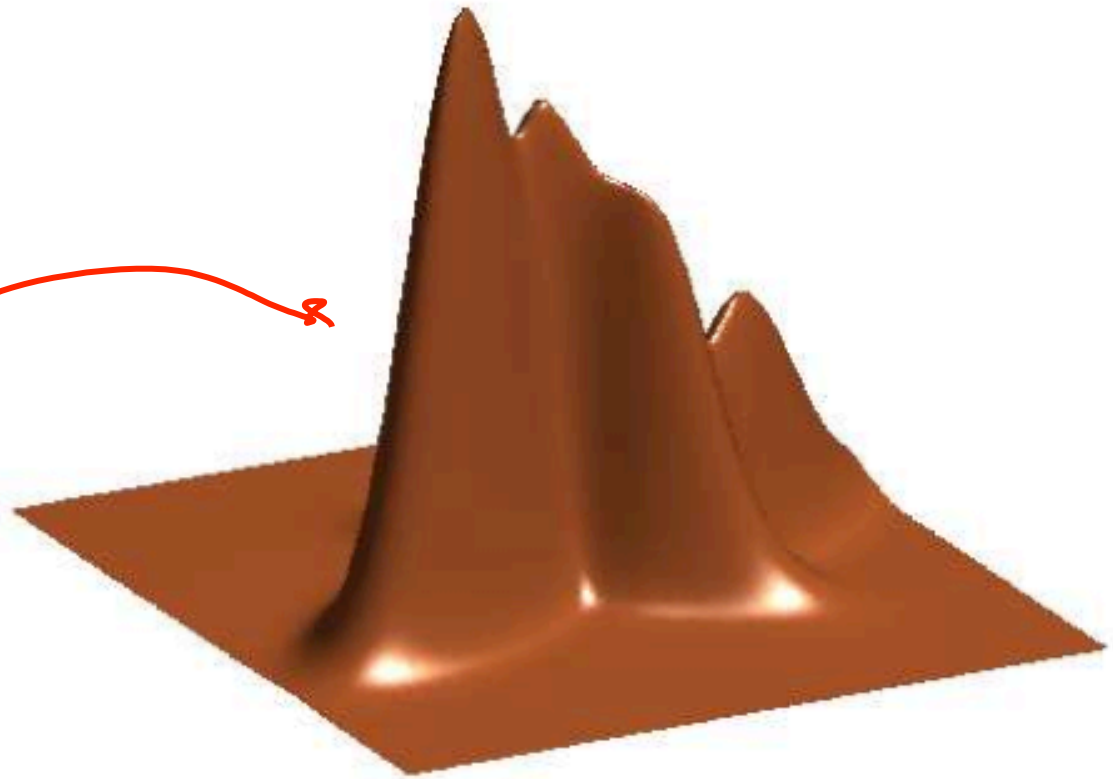
$z^i = \text{Sweden}, \quad x^i \sim N(6'1'', (3'')^2)$



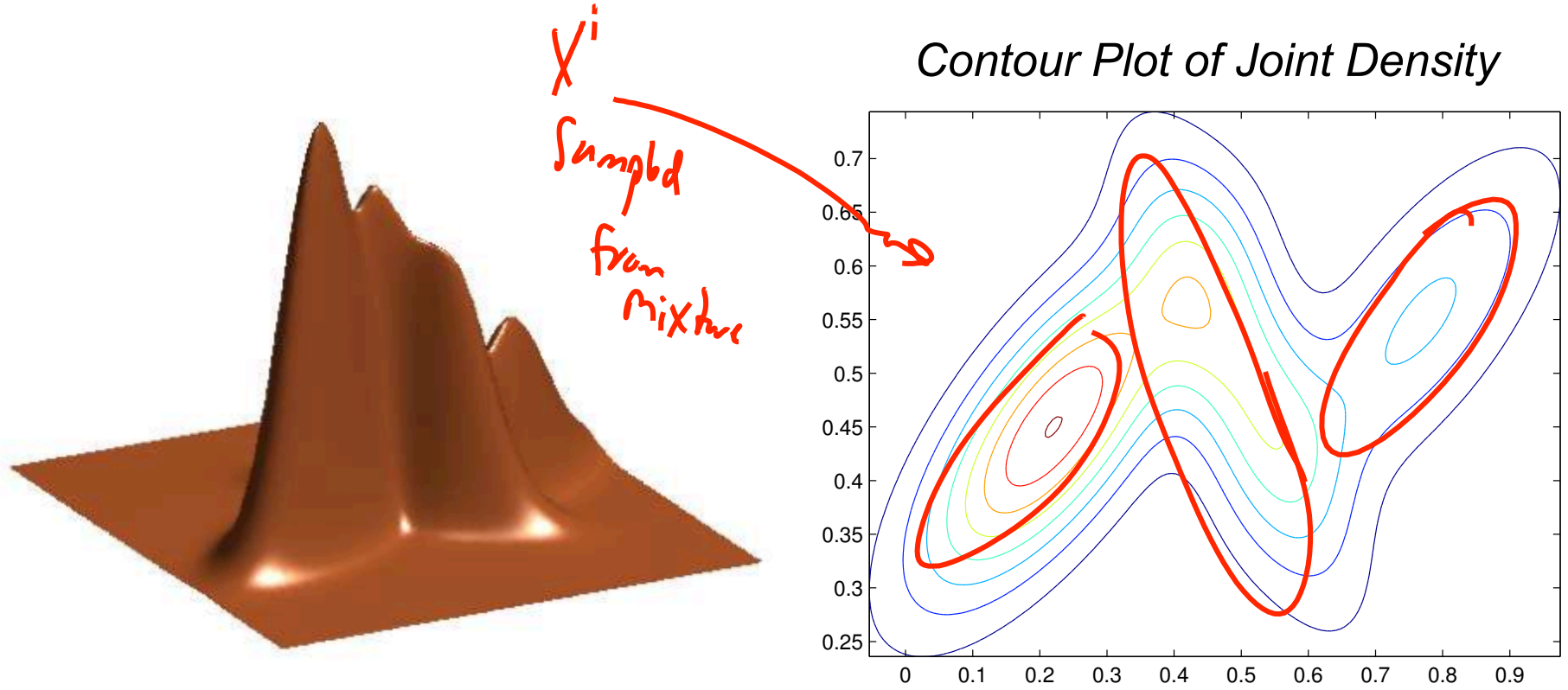
Density Estimation

- Estimate a density based on $x^1, \ldots, x^N$

e.g., fit mixture of Gaussians from points



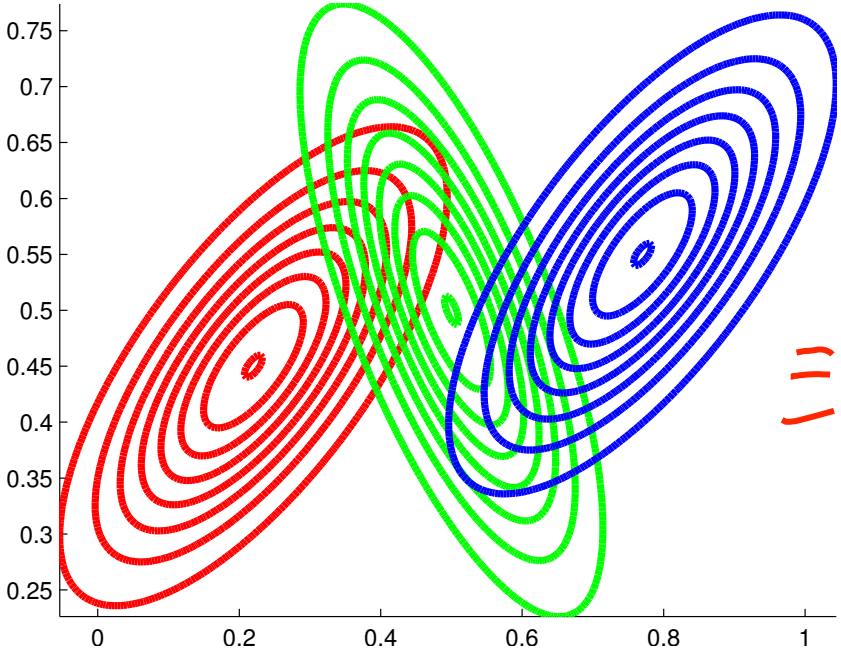
Density Estimation



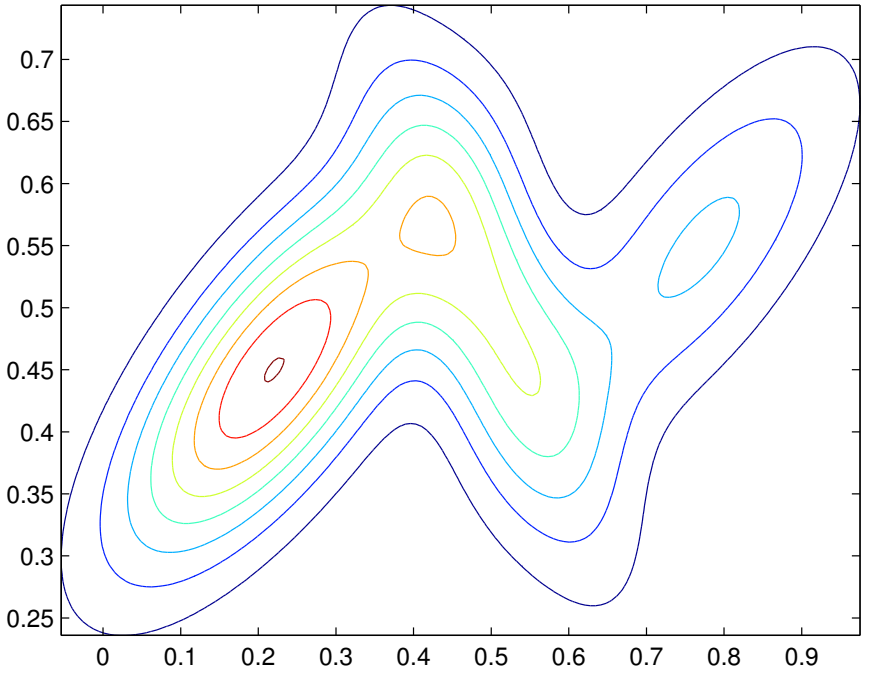
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



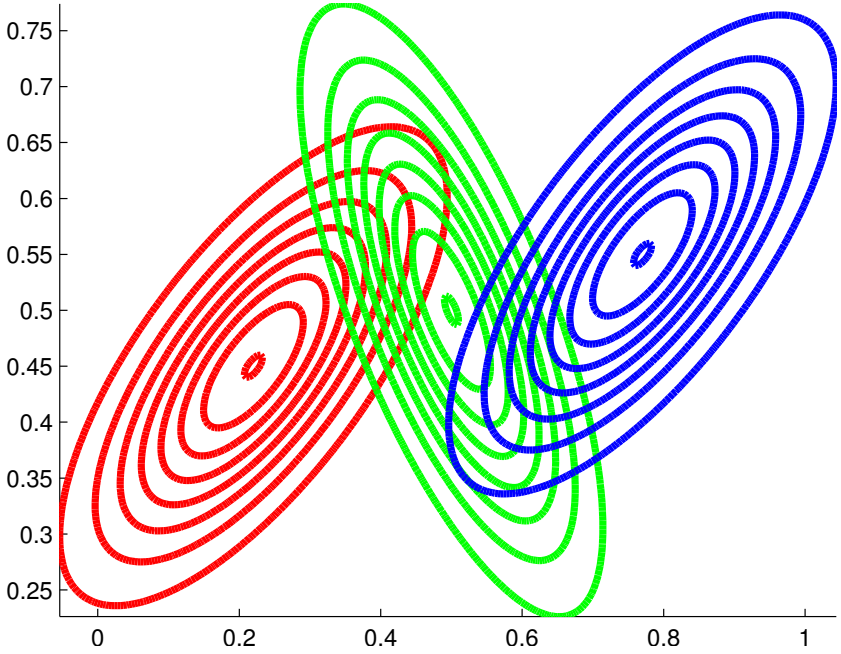
Contour Plot of Joint Density



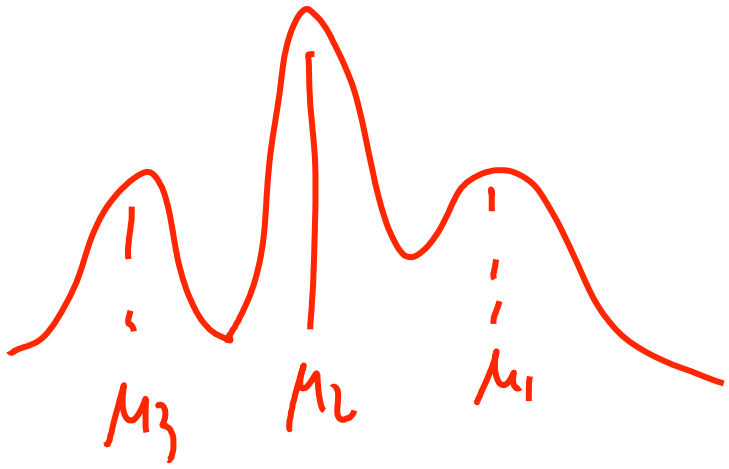
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



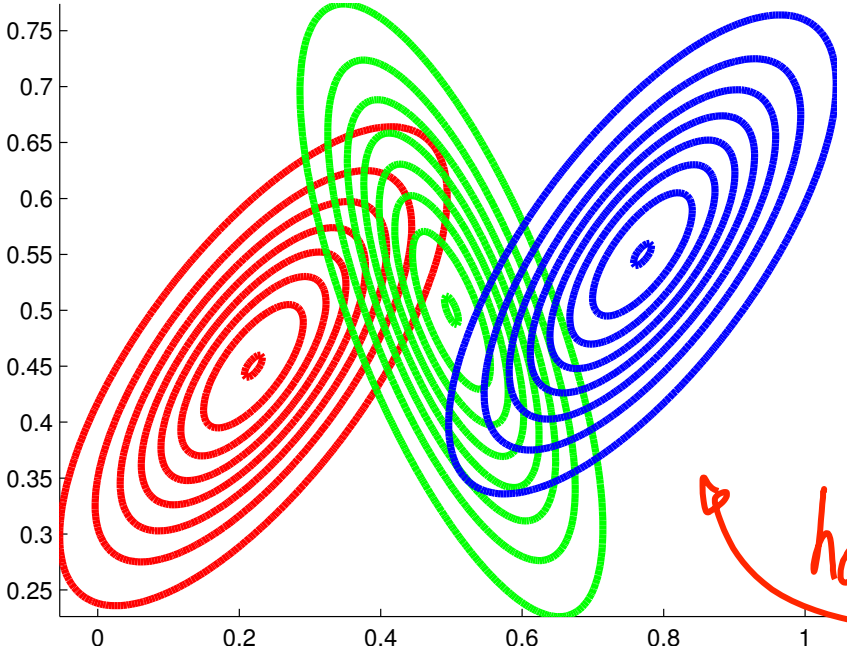
$$p(x^i|\pi,\mu,\Sigma) = \sum_{j=1}^{K} \pi_j \, N(\mu_j, \Sigma_j)$$



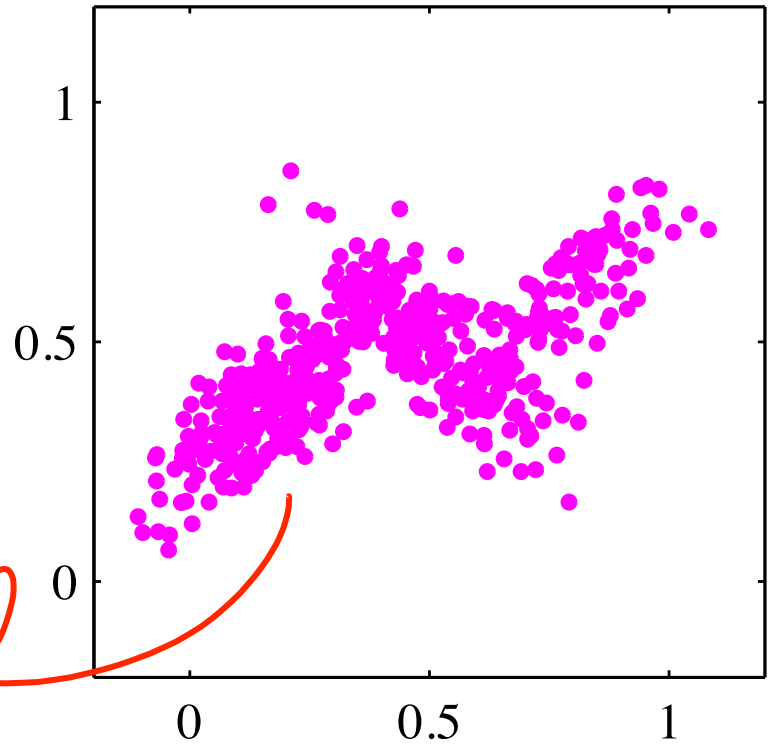
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



Our actual observations

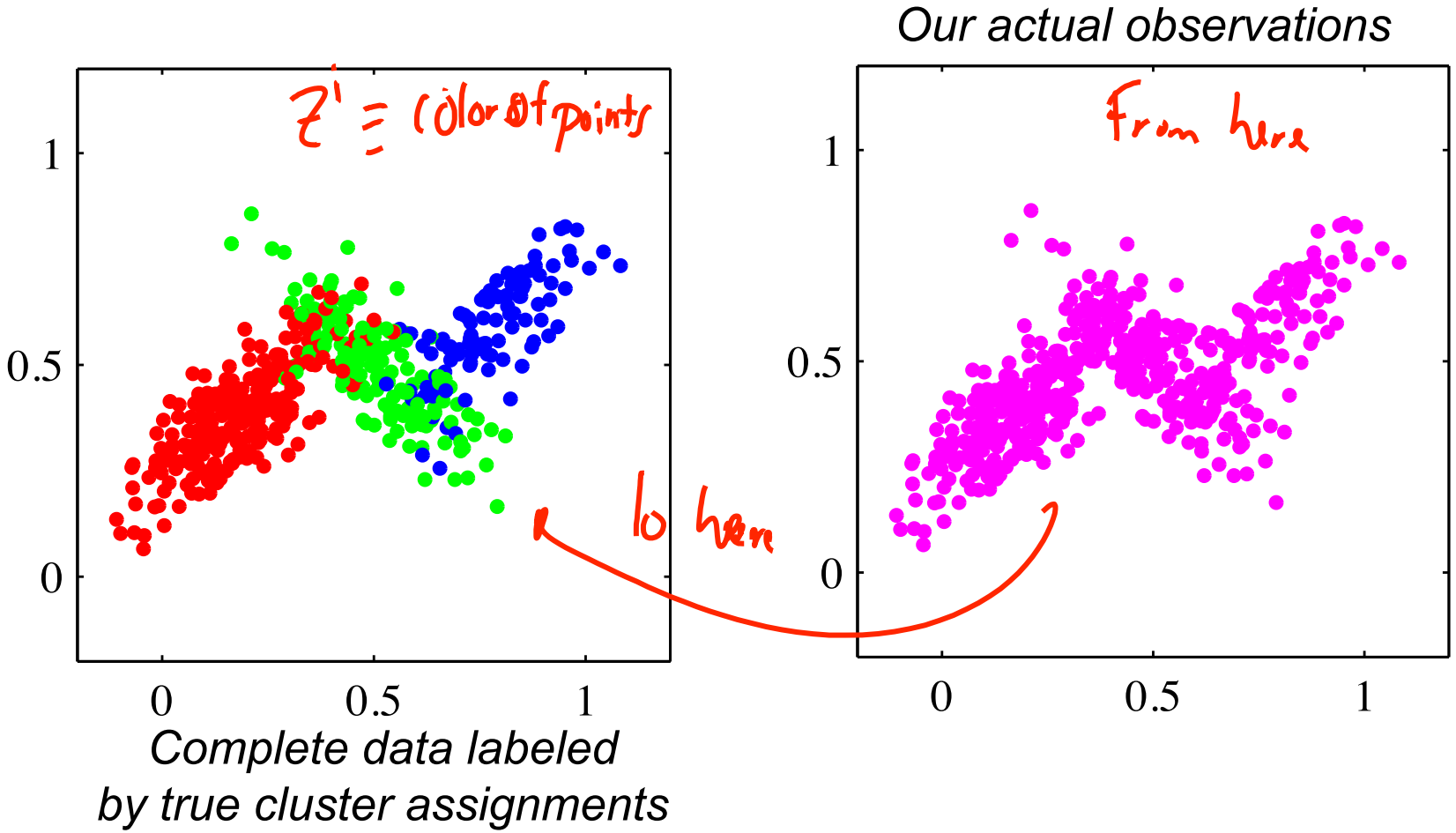


how??

C. Bishop, Pattern Recognition & Machine Learning

Clustering our Observations

- Imagine we have an assignment of each x^i to a Gaussian

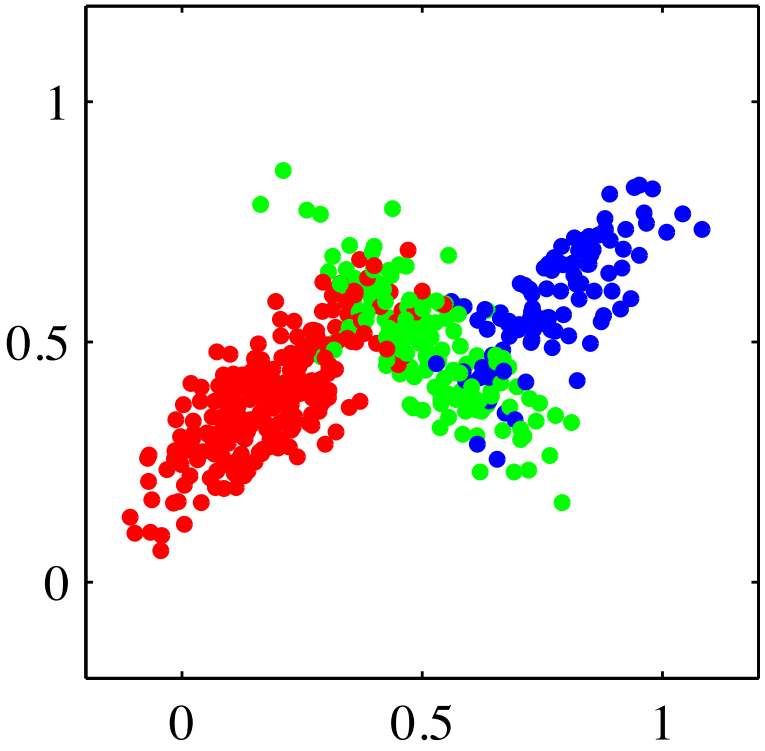


Complete data labeled by true cluster assignments

C. Bishop, Pattern Recognition & Machine Learning

Clustering our Observations

- Imagine we have an assignment of each x^i to a Gaussian



Complete data labeled by true cluster assignments

- Introduce latent cluster indicator variable z^i

$z^i \leftarrow$ color of each point

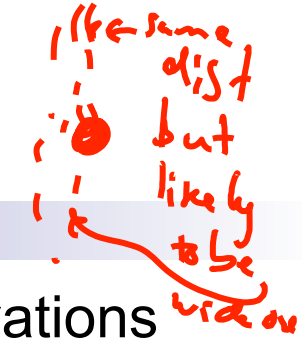
- Then we have

$$p(x^i | z^i, \pi, \mu, \Sigma) = N(\mu_{blue}, \Sigma_{blue})$$

if we had z^i, estimating $\mu_{blue}, \Sigma_{blue}$ would be easy

C. Bishop, Pattern Recognition & Machine Learning

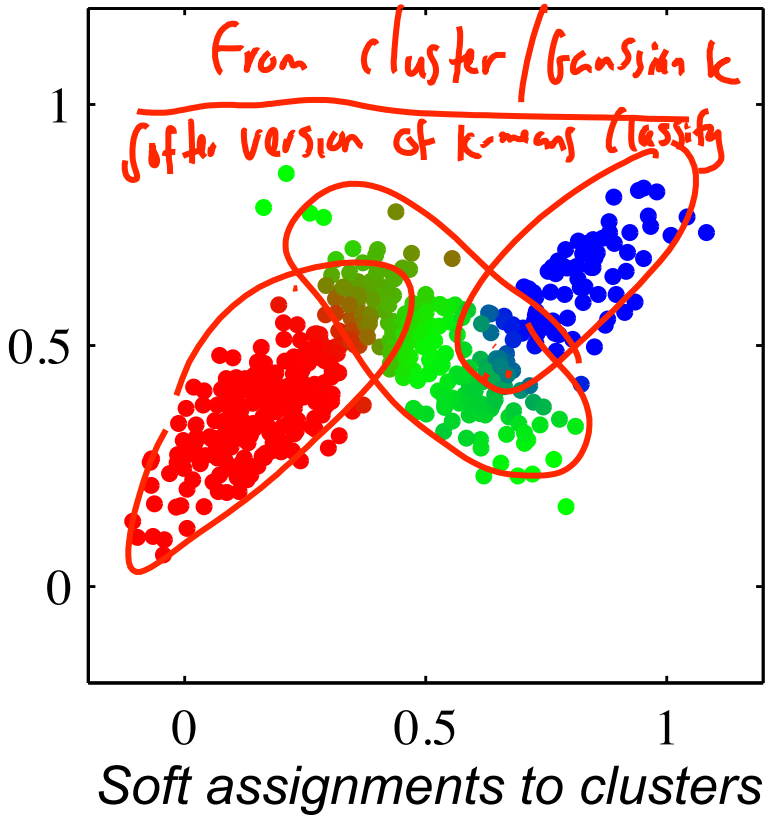
Clustering our Observations



- We must infer the cluster assignments from the observations

r_{ik} = how much point i comes from cluster/Gaussian k — "responsibilities"

soft version of k-means classify



Soft assignments to clusters

- Posterior probabilities of assignments to each cluster *given* model parameters:

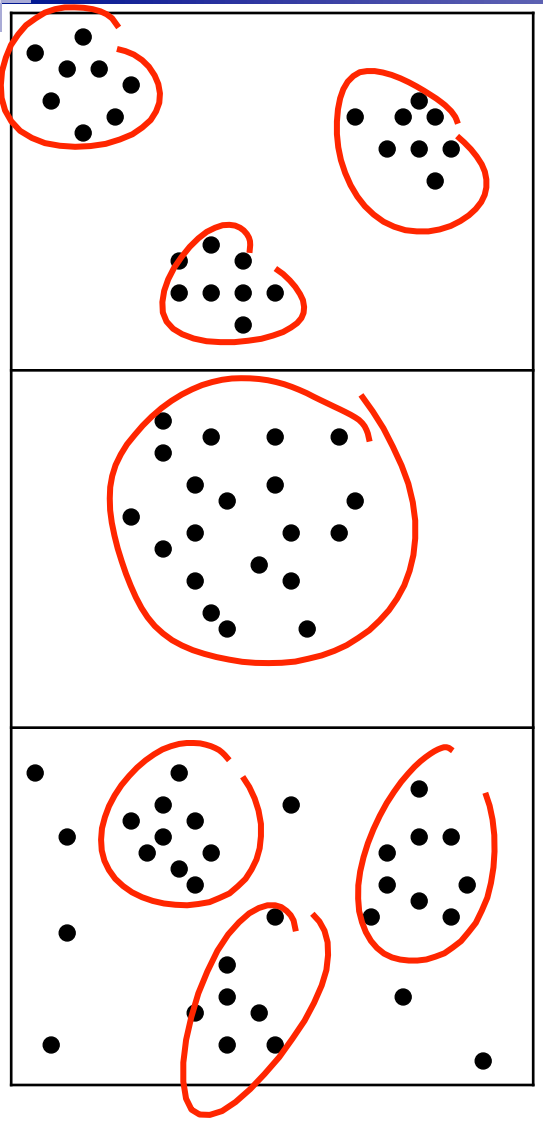
$$r_{ij} = p(z^i = j | x^i, \pi, \mu, \Sigma) =$$

$$= \frac{\pi_j \, N(\mu_j, \Sigma_j)}{\sum_{l=1}^{K} \pi_l \, N(\mu_l, \Sigma_l)}$$

normalize: $\sum_j r_{ij} = 1$

C. Bishop, Pattern Recognition & Machine Learning

Unsupervised Learning: not as hard as it looks



Sometimes easy

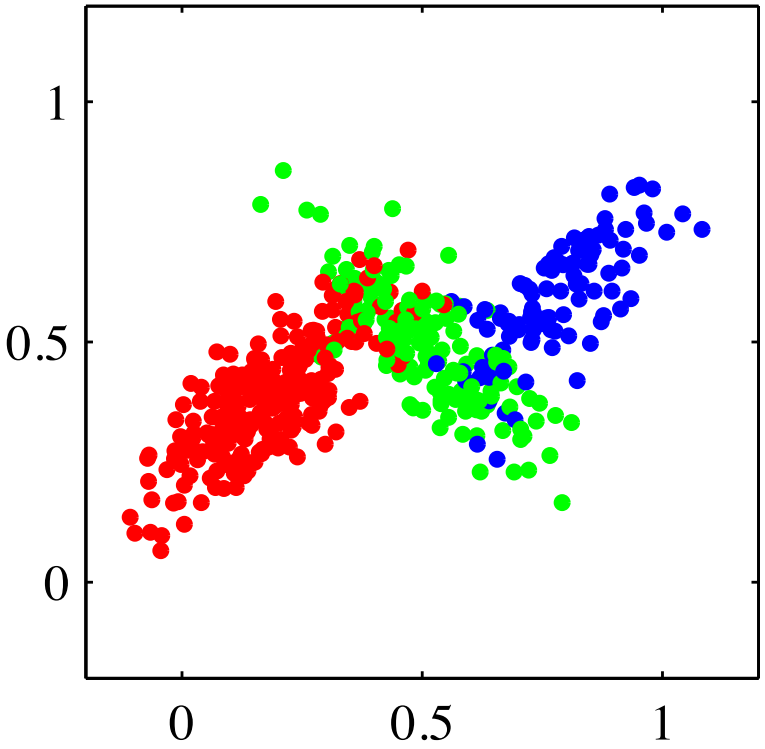
Sometimes impossible

and sometimes in between

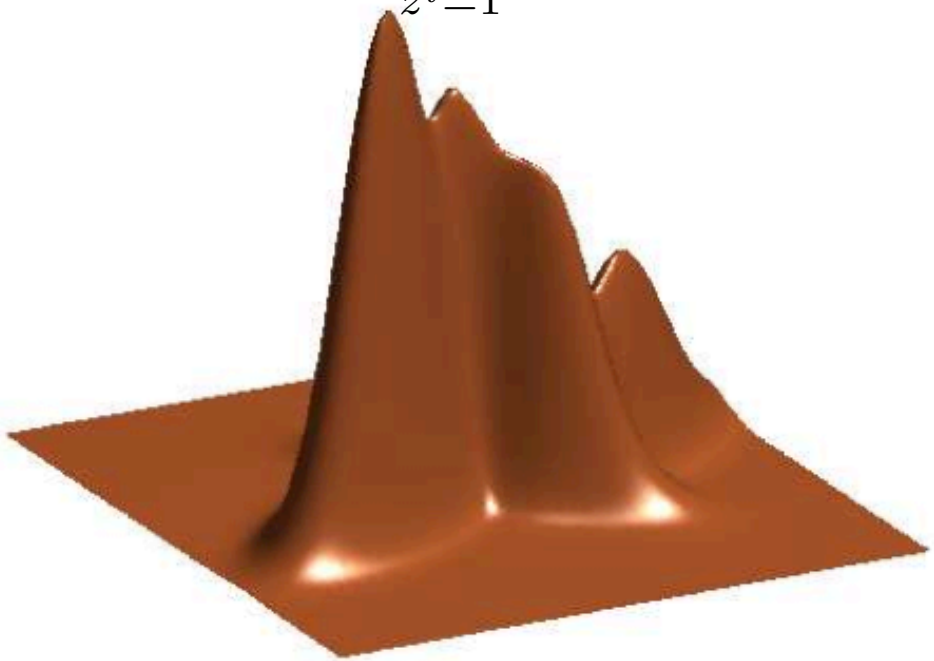
Summary of GMM Concept

- Estimate a density based on $x^1,\ldots,x^N$

$$p(x^i|\pi,\mu,\Sigma) = \sum_{z^i=1}^{K} \pi_{z^i}\mathcal{N}(x^i|\mu_{z^i},\Sigma_{z^i})$$



Complete data labeled by true cluster assignments



Surface Plot of Joint Density, Marginalizing Cluster Assignments

Summary of GMM Components

- Observations $x^i \in \mathbb{R}^d, \quad i = 1, 2, \ldots, N$
- Hidden cluster labels $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, 2, \ldots, N$
- Hidden mixture means $\mu_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, K$
- Hidden mixture covariances $\Sigma_k \in \mathbb{R}^{d \times d}, \quad k = 1, 2, \ldots, K$
- Hidden mixture probabilities $\pi_k, \quad \sum_{k=1}^{K} \pi_k = 1$

Gaussian mixture marginal and conditional likelihood **:**

$$p(x^i|\pi, \mu, \Sigma) = \sum_{z^i=1}^{K} \pi_{z^i} \; p(x^i|z^i, \mu, \Sigma)$$

$$p(x^i|z^i, \mu, \Sigma) = \mathcal{N}(x^i|\mu_{z^i}, \Sigma_{z^i})$$

Application to Document Modeling

Machine Learning – CSEP546

Carlos Guestrin

University of Washington

February 18, 2014

Cluster Documents

- Cluster documents based on topic

Document Representation

- Bag of words model

ignore order of words

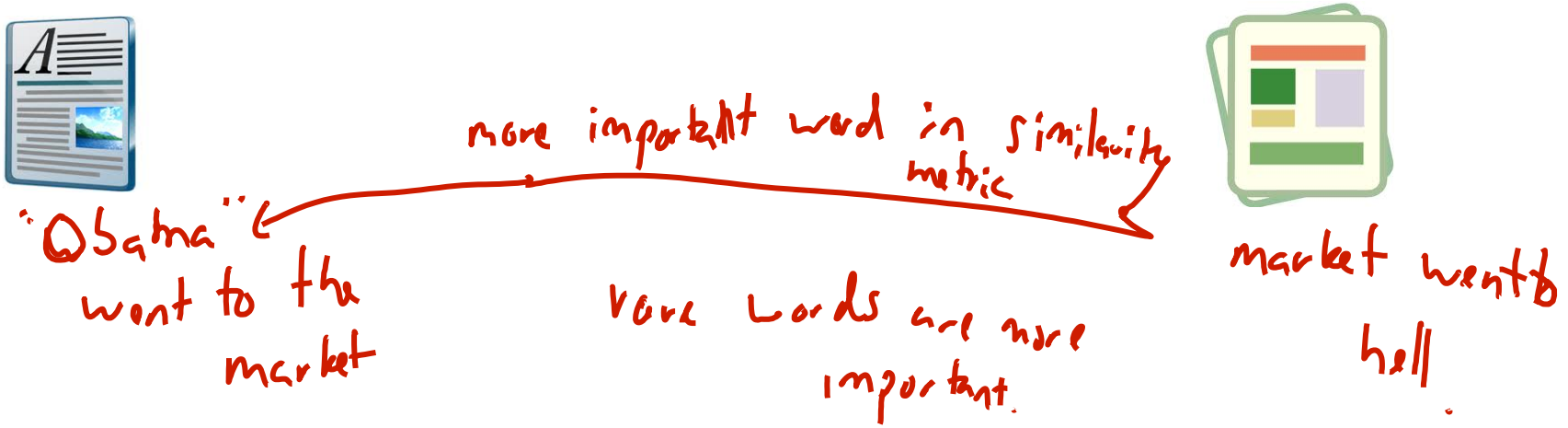
word count

$$X = \begin{bmatrix} wc_1 \\ wc_2 \\ \vdots \\ wc_d \end{bmatrix}$$

$d = |V|$ size of vocab

Issues with Document Representation

- Words counts are ***bad*** for standard similarity metrics



- Term Frequency – Inverse Document Frequency (tf-idf)
  - Increase importance of rare words

TF-IDF

- Term frequency:

$$\mathrm{tf}(t,d) =$$

  - Could also use $\{0,1\}, 1+\log \mathrm{tf}(t,d), \ldots$.

- Inverse document frequency:

$$\mathrm{idf}(t,D) = \log \frac{|D|}{1+|\{d \in D : t \in d\}|}$$

- tf-idf:

$$\mathrm{tfidf}(t,d,D) =$$

  - High for document d with high frequency of term t (high "term frequency") and few documents containing term t in the corpus (high "inverse doc frequency")

A Generative Model

- Documents: $d_1 \ldots d_n$
- Associated topics: $\pi_1, \ldots, \pi_K$ ← sample topic
- Parameters simple mixture of Gaussians: j

$$x^{di} = \left[\text{tfidf} \right] \sim N(\mu_j, \Sigma_j)$$

tfidf ≥ 0

Gaussians $\in \mathbb{R}$

more typically we use dist. positive countable

What you get from mixture model for documents

- Words give topic: $P(word \mid topic)$ ← topic model

topic = 5 (sports)

→ $P(ball \mid topic=5)$ is high ; $P(Comp.\ Scientist \mid topic=5)$ low

- Topic proportions:

$\pi_1, \pi_2 \ldots \pi_k$ ← likely doc → topic 1, 2, ⋮, k

- Topic distribution of each document:

Results from Wikipedia data using similar model (LDA)

Expectation Maximization

Machine Learning – CSEP546

Carlos Guestrin

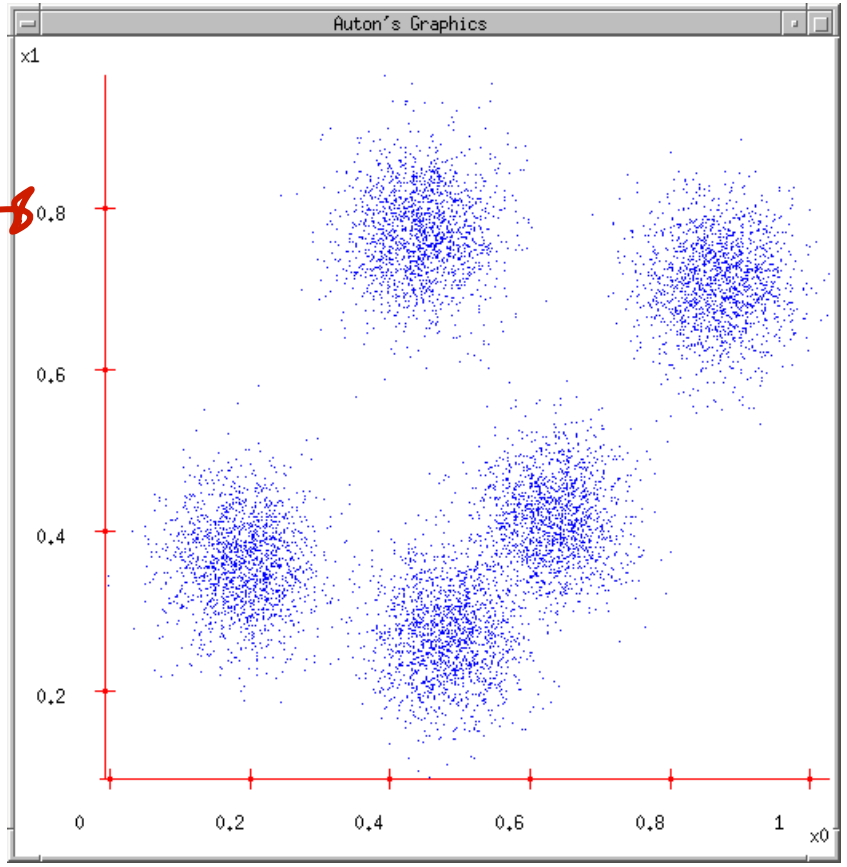
University of Washington

February 18, 2014

Next… back to Density Estimation

What if we want to do density estimation with multimodal or clumpy data?

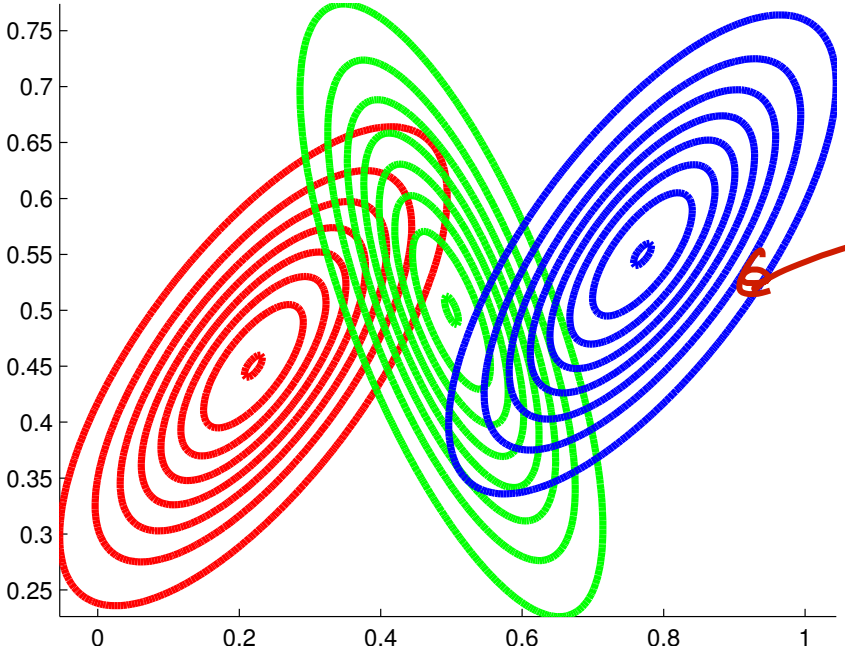
Goal: fit MoG model



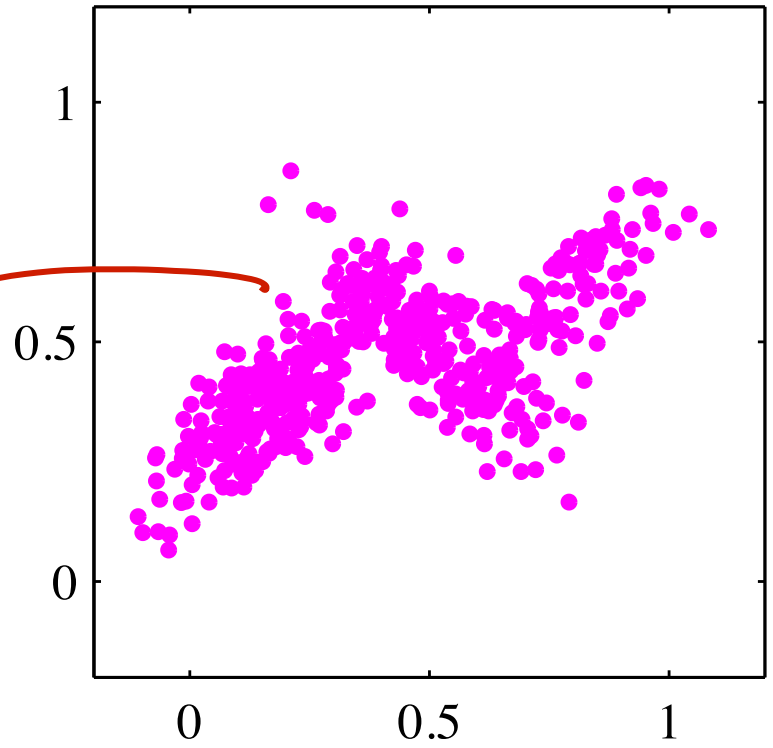
Learning Model Parameters

- Want to learn model parameters

Mixture of 3 Gaussians



Our actual observations



C. Bishop, Pattern Recognition & Machine Learning

ML Estimate of Mixture Model Params

Don't observe z^i ← cluster assignment

- Log likelihood

max $\theta = \{\mu, \Sigma\}$

$$L_x(\theta) \triangleq \log p(\{x^i\} \mid \theta) = \sum_{i=1}^{n} \log \sum_{z^i} p(x^i, z^i \mid \theta)$$

z^i ← avg unknown values

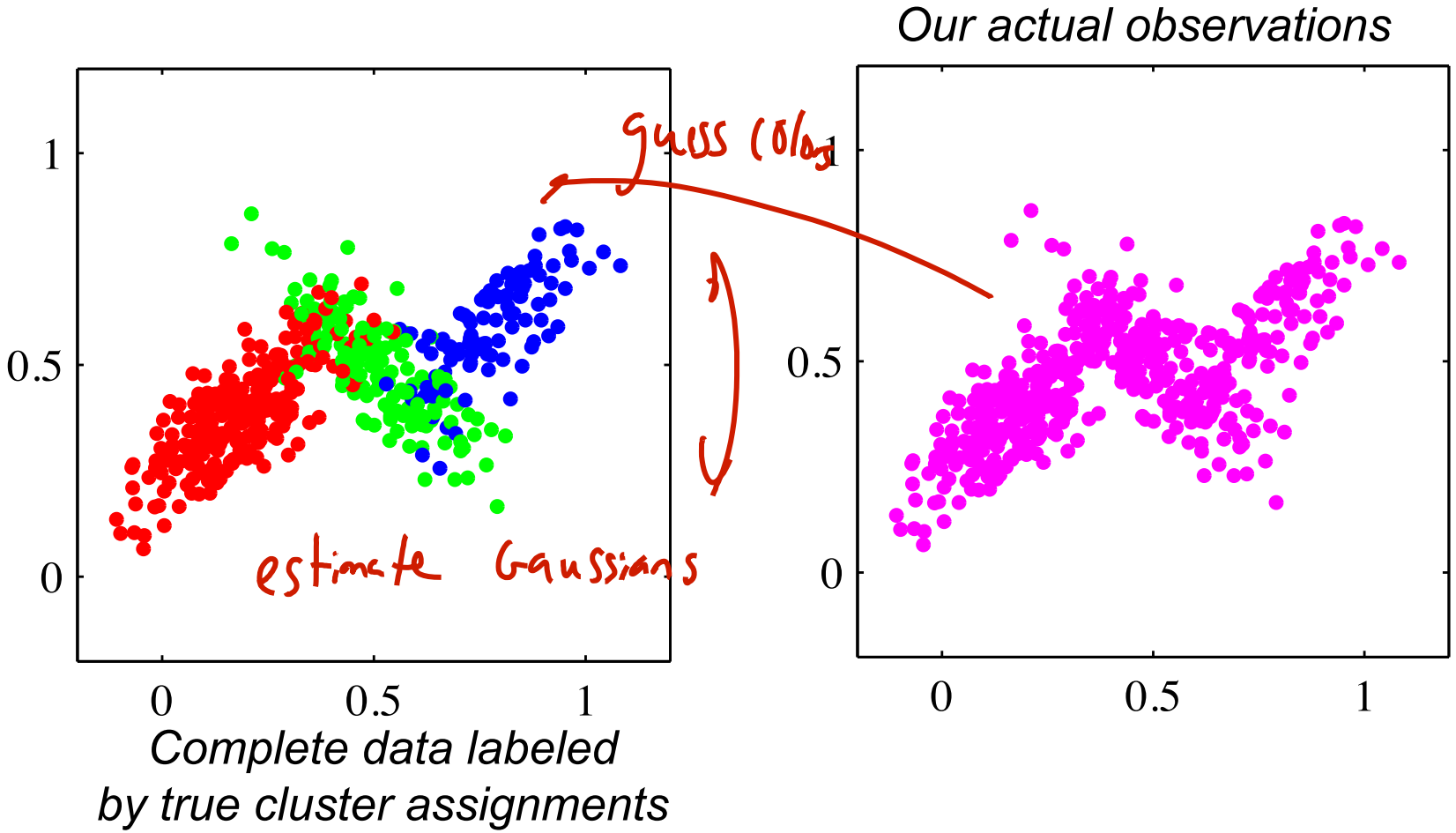
- Want ML estimate

$$\hat{\theta}^{ML} = \arg\max_{\theta \{\mu, \Sigma\}} L_x(\theta)$$

- Neither convex nor concave and local optima

Complete Data

- Imagine we have an assignment of each x^i to a cluster



C. Bishop, Pattern Recognition & Machine Learning

If "complete" data were observed…

- Assume class labels z^i were observed in addition to x^i

$$L_{x,z}(\theta) = \sum_i \log p(x^i, z^i \mid \theta)$$

Don't know z^i, estimate using responsibilities r_{ij}

- Compute ML estimates
  - Separates over clusters *k*!

$\pi_j = \frac{\text{Count}(z^i = j)}{N}$

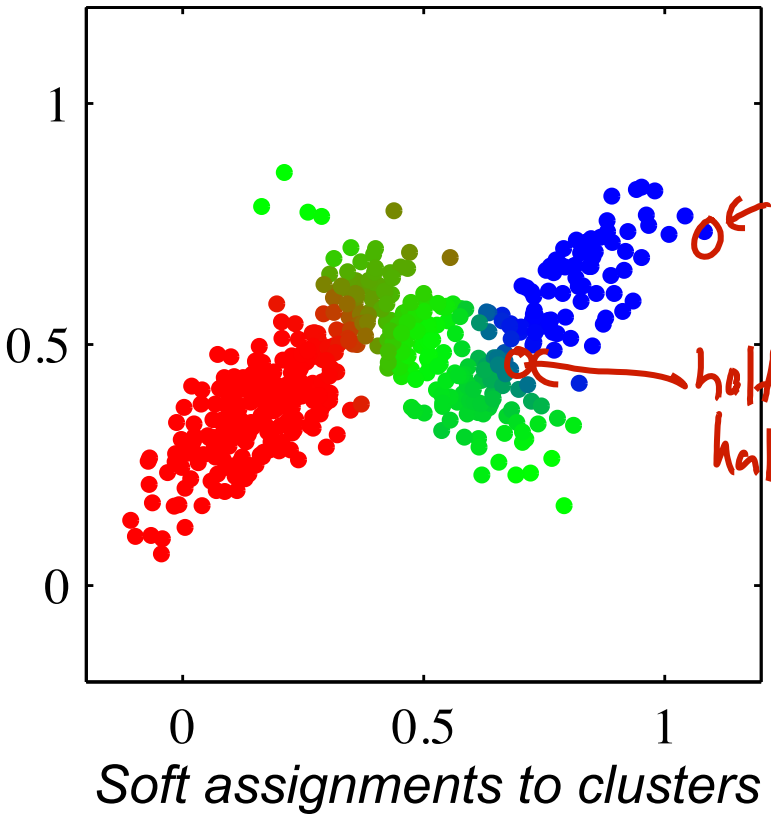
- Example: mixture of Gaussians (MoG) $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$

Take points in cluster j $(z^i = j)$

μ_j = avg of points Σ_j is covariance

Cluster Responsibilities

- We must infer the cluster assignments from the observations



Soft assignments to clusters

Given π, μ, Σ

- Posterior probabilities of assignments to each cluster *given* model parameters:

$$r_{ik} = p(z^i = k \mid x^i, \pi, \phi) = \frac{\pi_k \, \mathcal{N}(x^i; \mu_k, \Sigma_k)}{\sum_l \pi_l \, \mathcal{N}(x^i; \mu_l, \Sigma_l)}$$

Iterative Algorithm

- Motivates a coordinate ascent-like algorithm:
  1. Infer missing values z^i given estimate of parameters $\hat{\theta}$
  2. Optimize parameters to produce new $\hat{\theta}$ given "filled in" data z^i
  3. Repeat

- Example: MoG
  1. Infer "responsibilities"

$$r_{ik}^{(t)} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) =$$

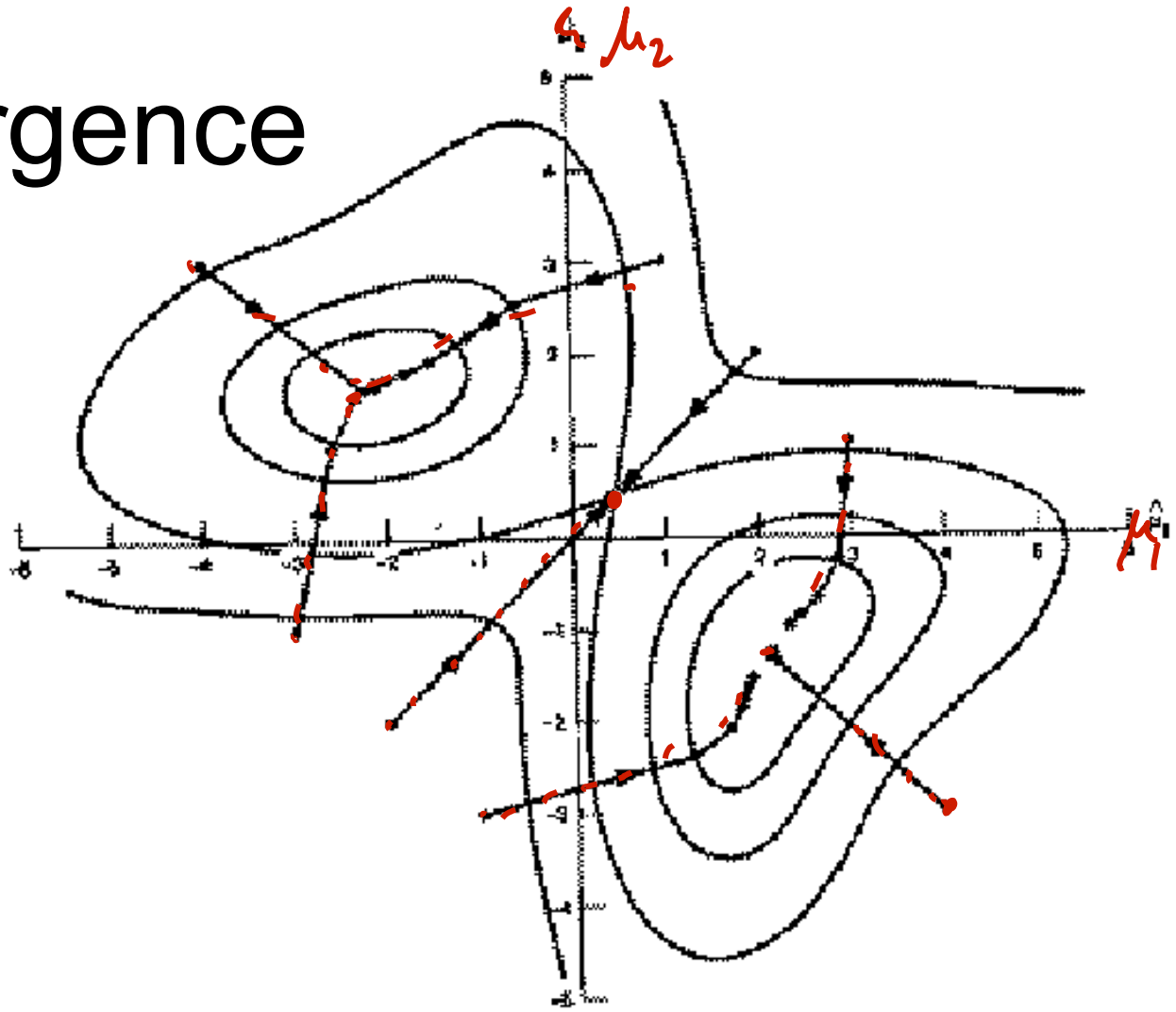
  2. Optimize parameters

$$\max \text{ w.r.t. } \pi_k :$$

$$\max \text{ w.r.t. } \mu_k :$$

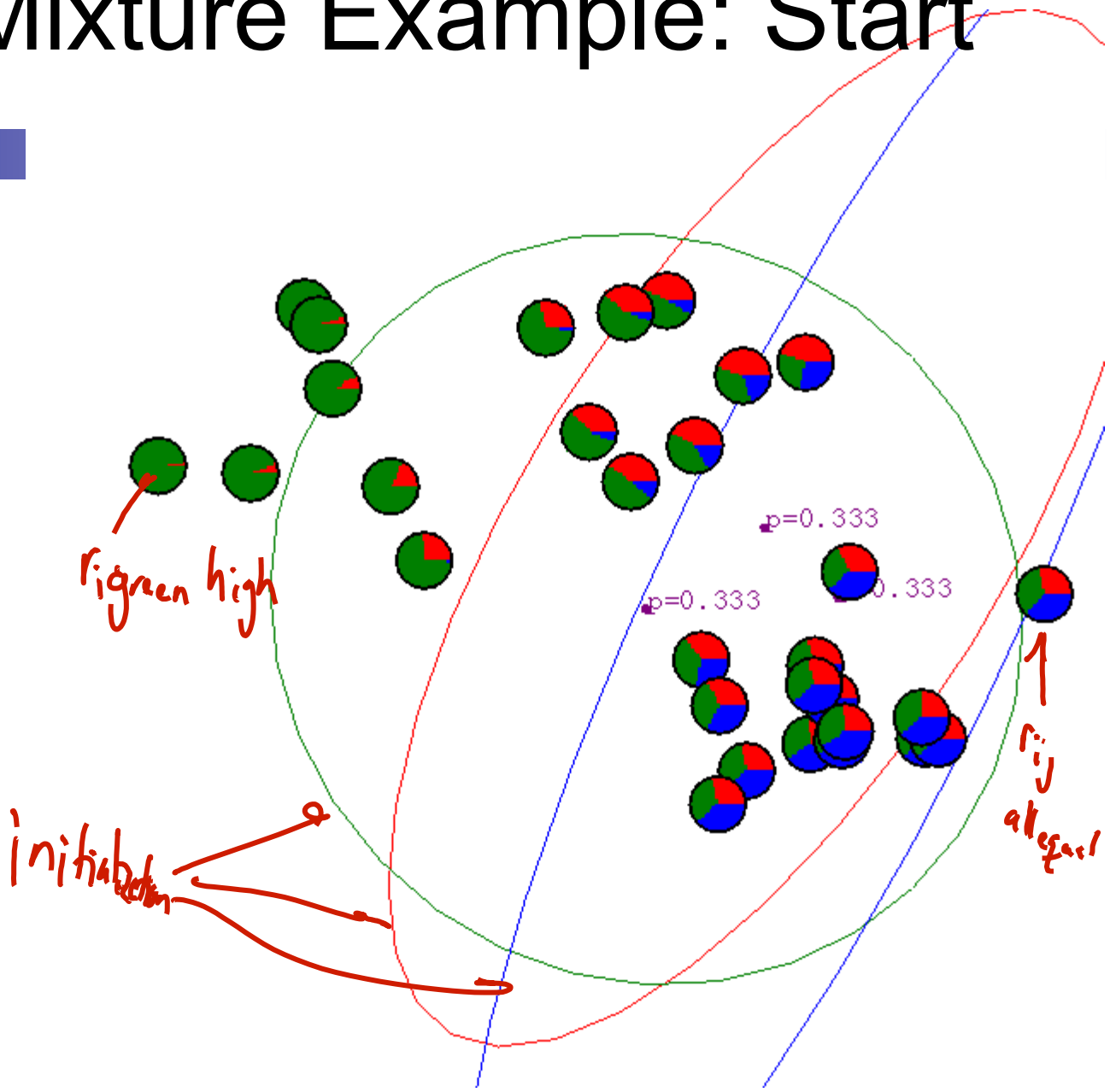
E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ➔ convergence to a local optimum guaranteed

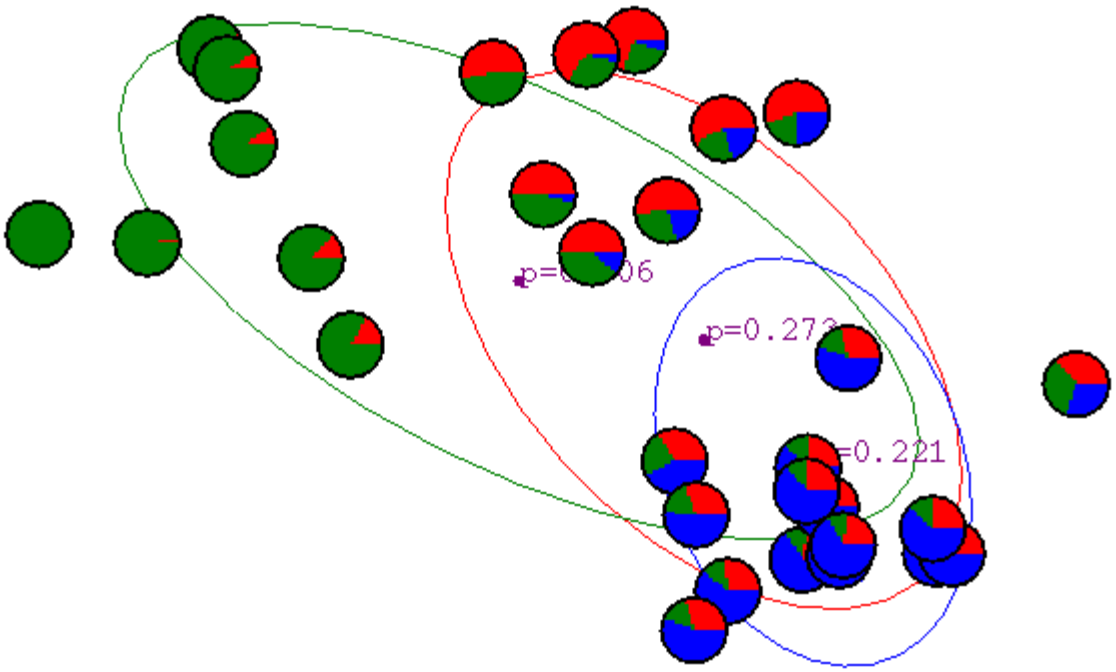


- This algorithm is REALLY USED. And in high dimensional state spaces, too. E.G. Vector Quantization for Speech Data

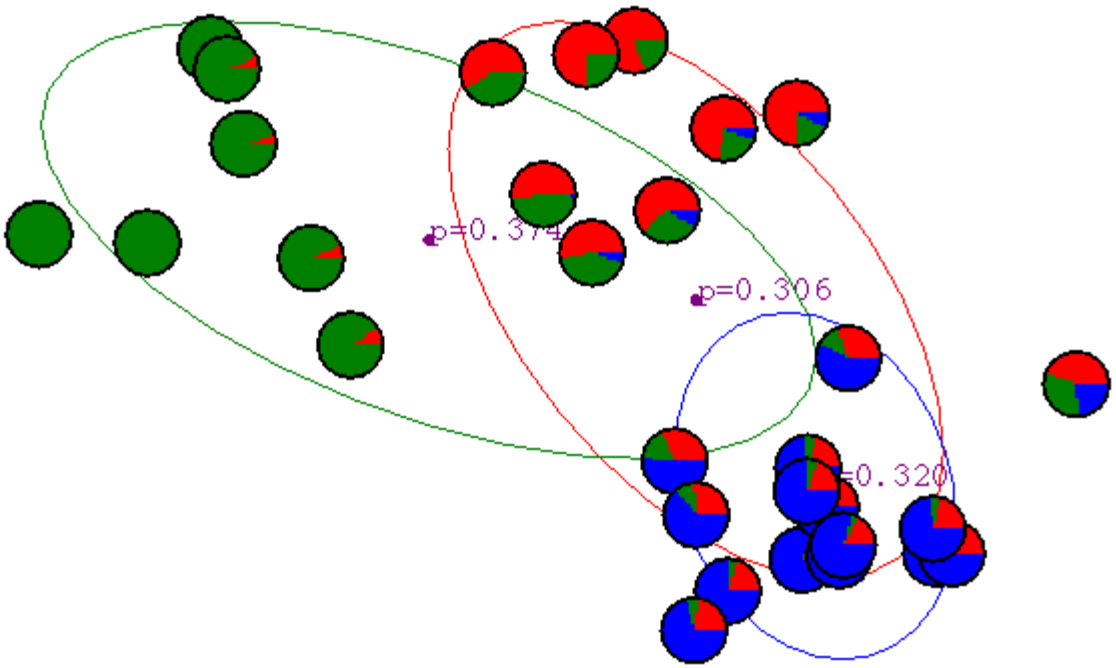
Gaussian Mixture Example: Start



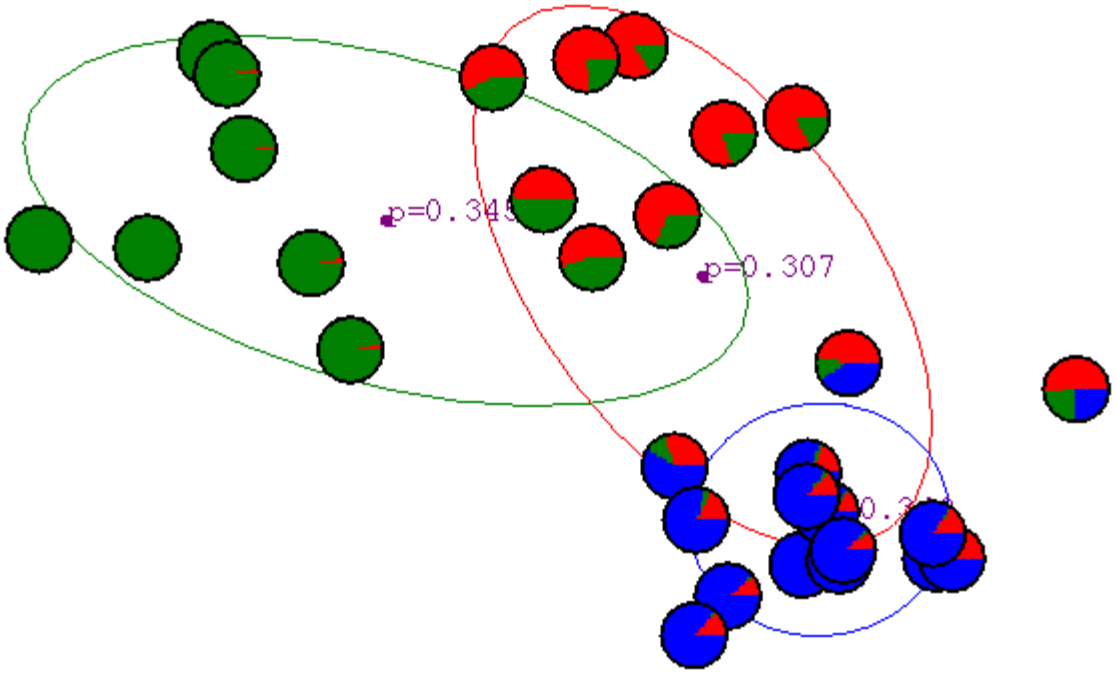
After first iteration



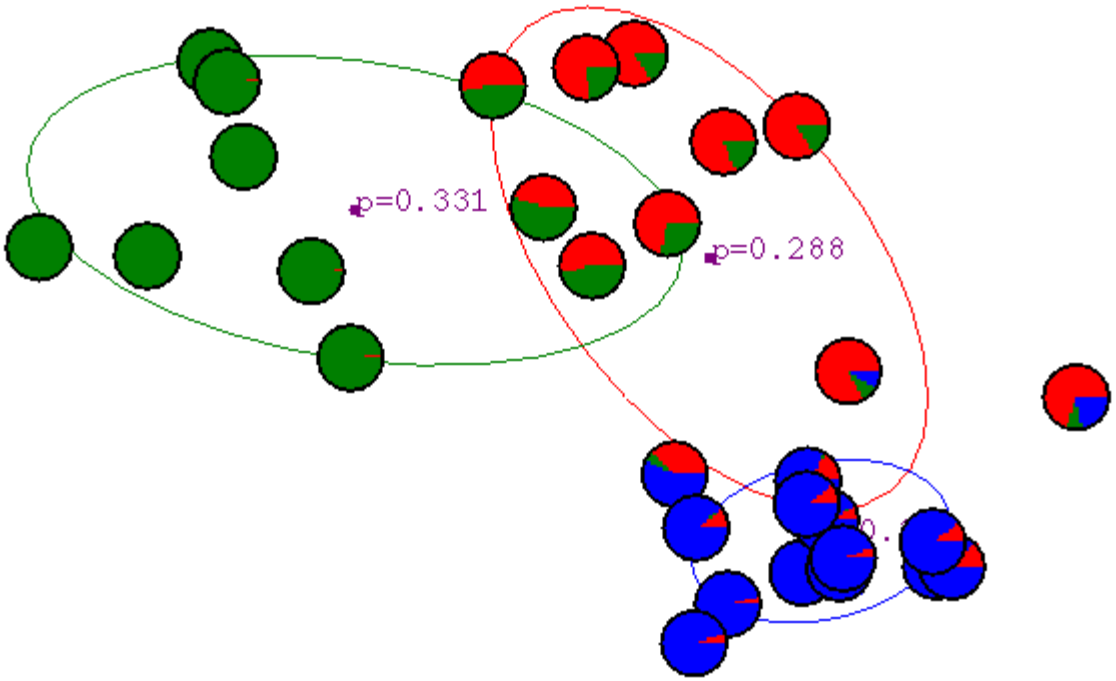
After 2nd iteration



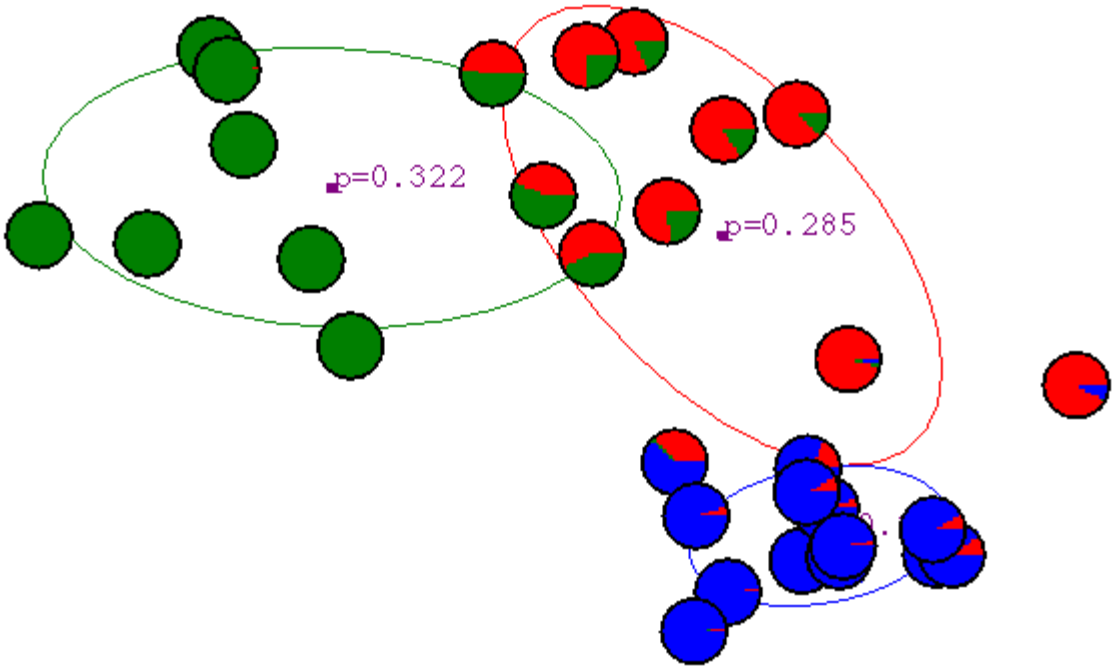
After 3rd iteration



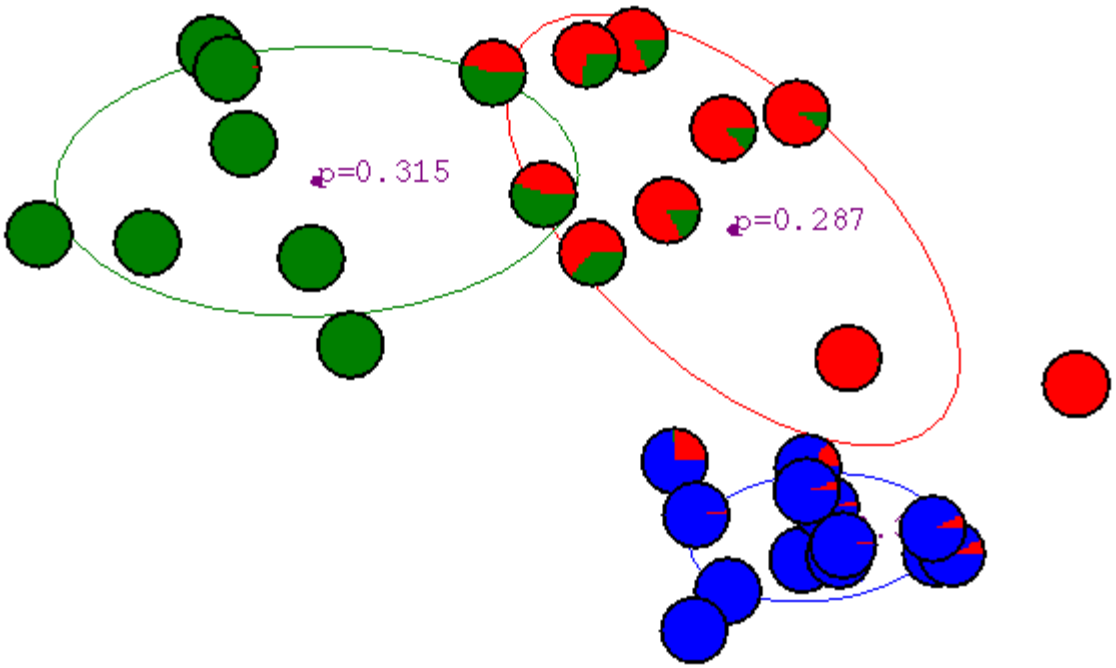
After 4th iteration



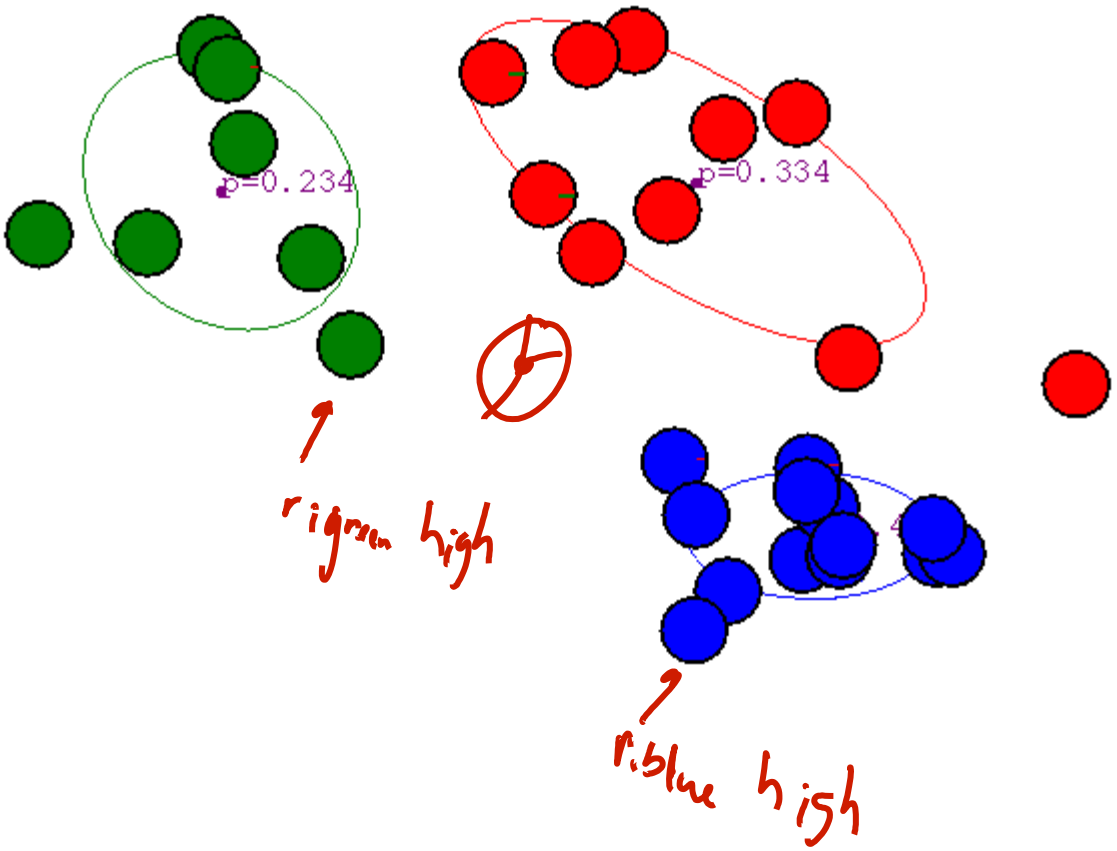
After 5th iteration



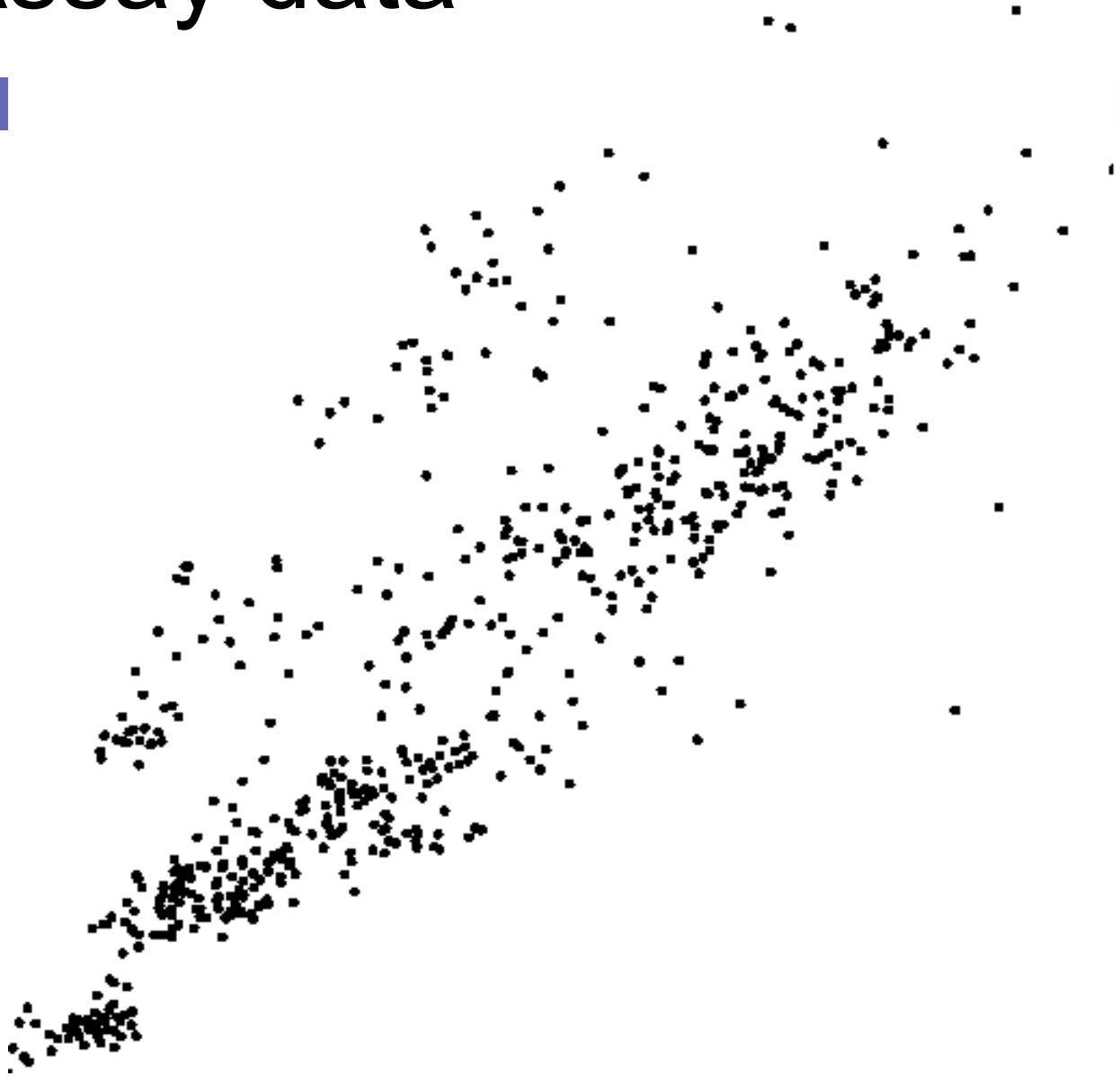
After 6th iteration



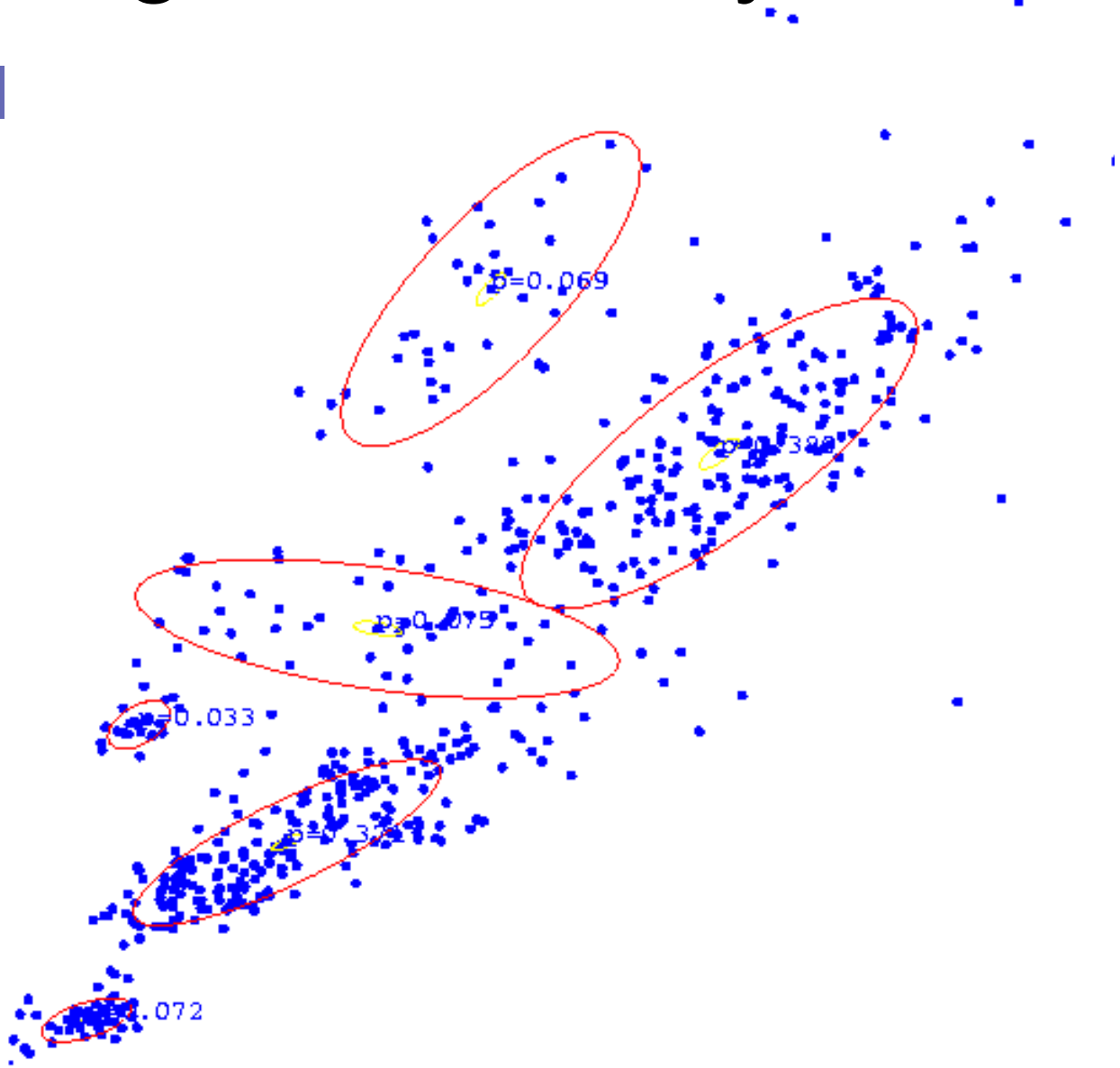
After 20th iteration



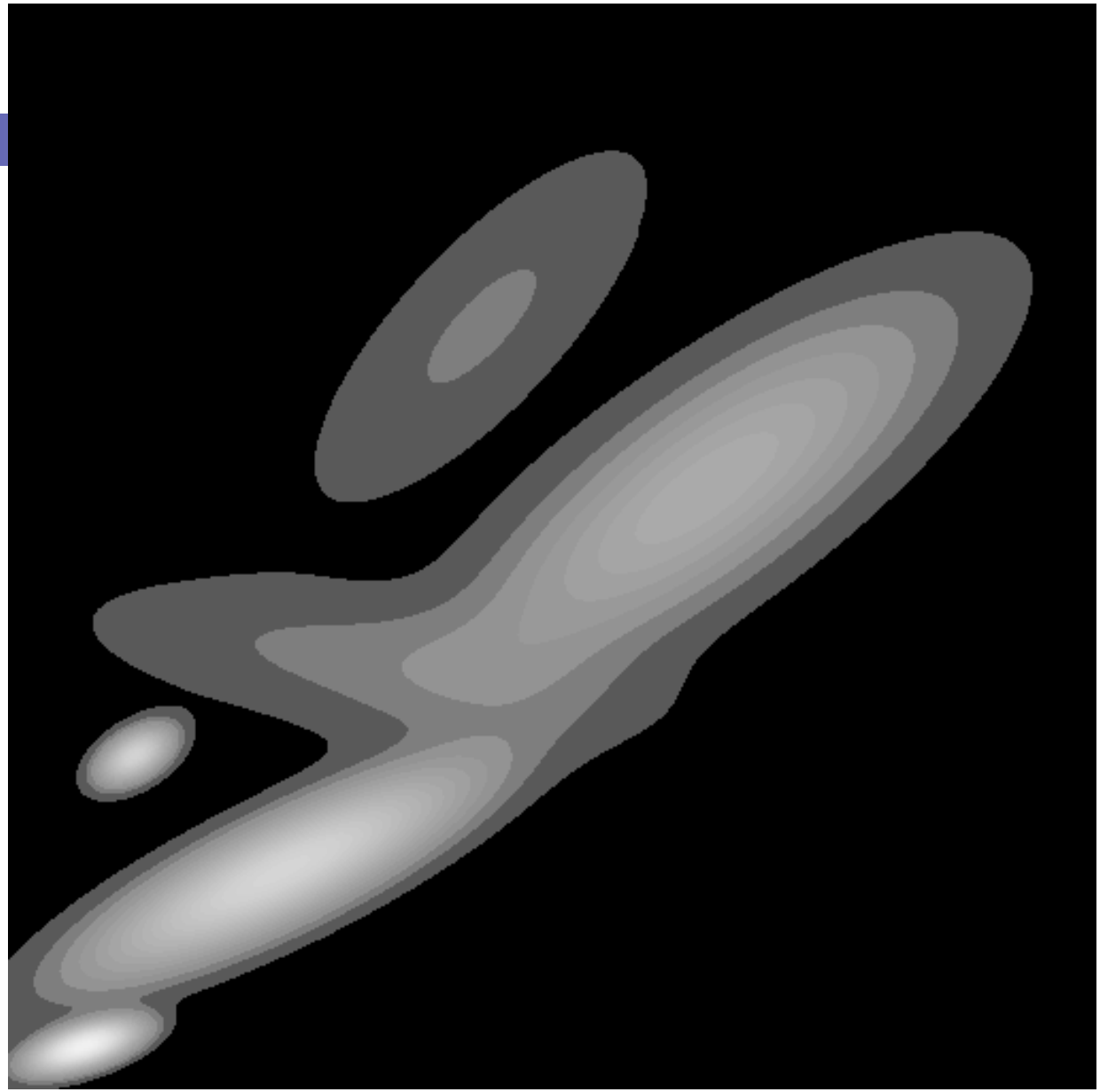
Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator



Initialization

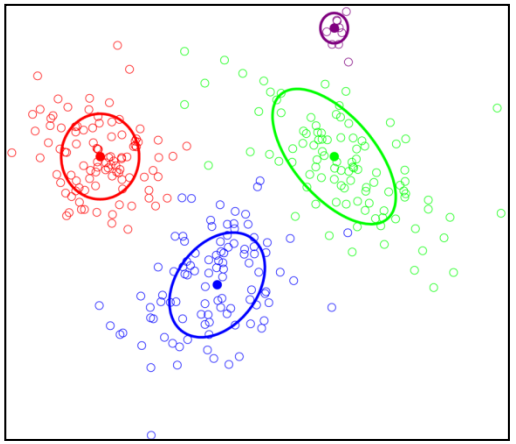
- In mixture model case where $y^i = \{z^i, x^i\}$ there are many ways to initialize the EM algorithm

- Examples:
  - Choose K observations at random to define each cluster. Assign other observations to the nearest "centriod" to form initial parameter estimates
  - Pick the centers sequentially to provide good coverage of data
  - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed

- Can be quite important to convergence rates in practice

Label switching

- Color = label does not matter
- Can switch labels and likelihood is unchanged

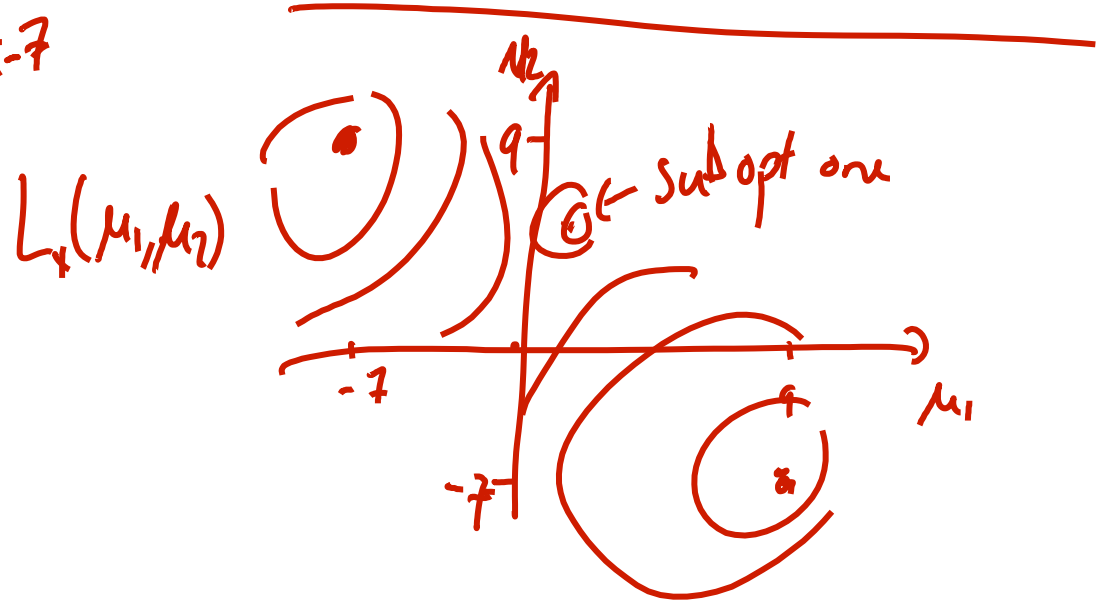


$\mu_1 = -7$

$\mu_2 = 9$

$\mu_1 = 9$

$\mu_2 = -7$



What you should know

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- EM for mixture of Gaussians:
  - How to "learn" maximum likelihood parameters (locally max. like.) in the case of unlabeled data
- Remember, E.M. can get stuck in local minima, and empirically it DOES
- EM is coordinate ascent