ECE 5424: Introduction to Machine Learning

Topics:

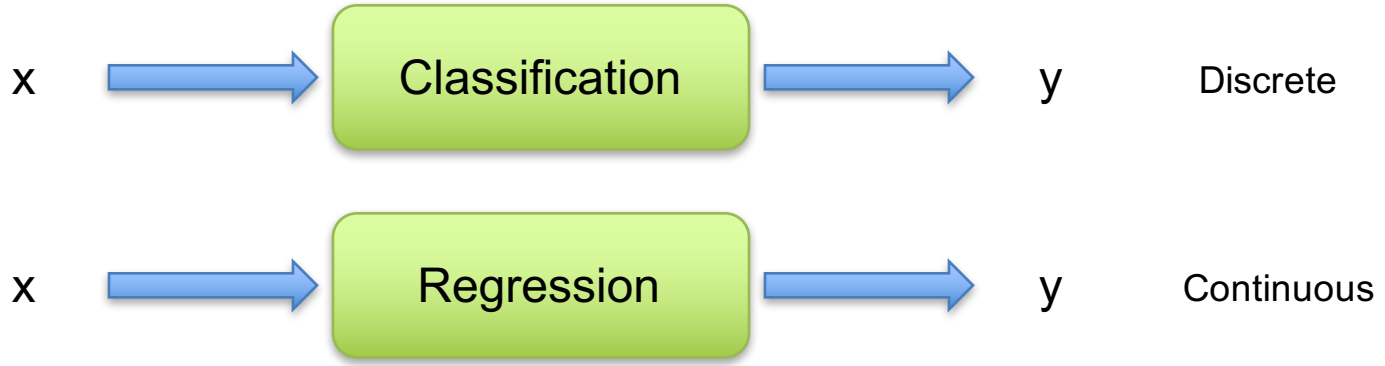
- Unsupervised Learning: Kmeans, GMM, EM

Readings: Barber 20.1-20.3

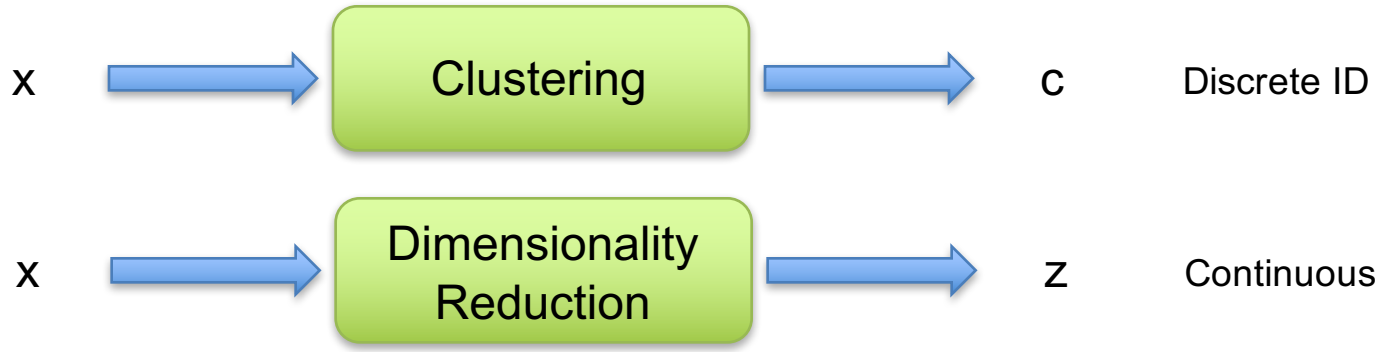
Stefan Lee
Virginia Tech

Tasks

Supervised Learning



Unsupervised Learning

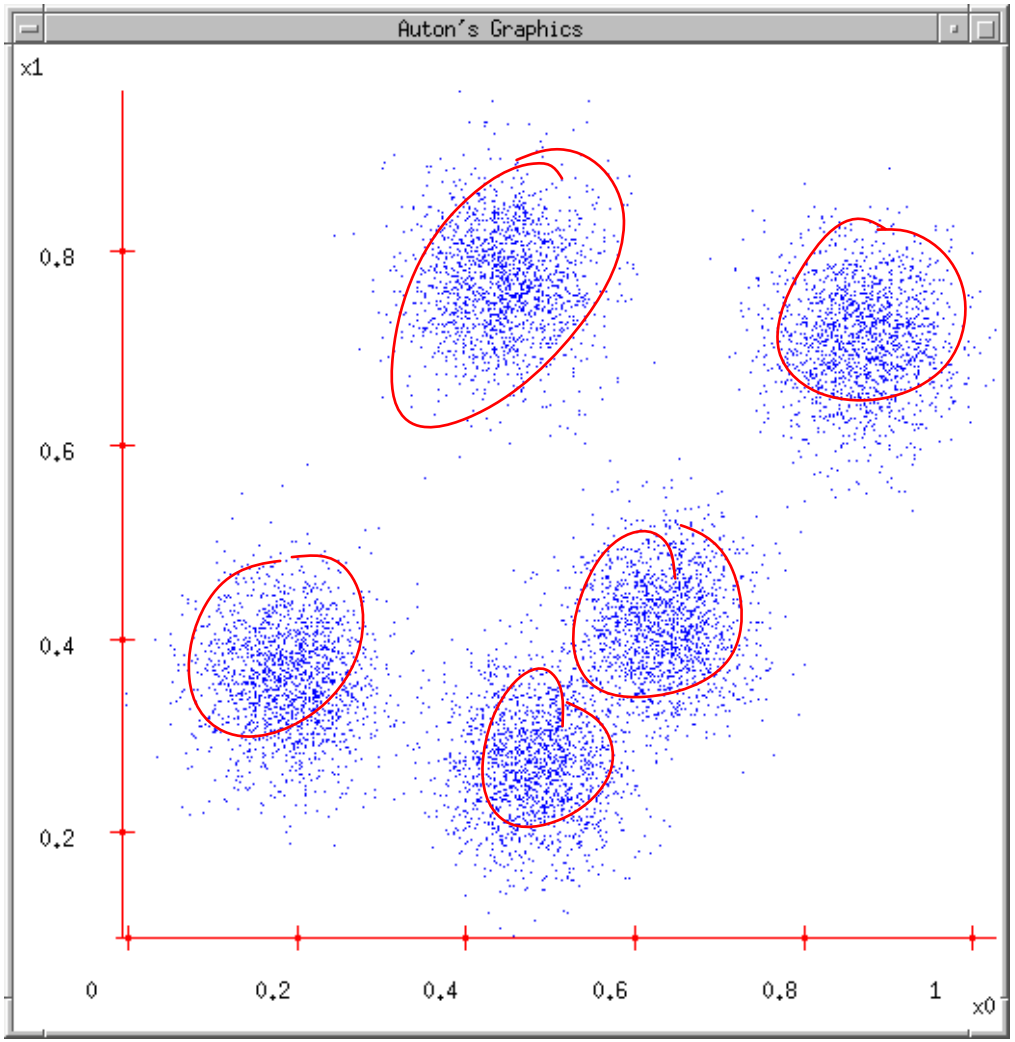


Unsupervised Learning

- Learning only with X
  - Y not present in training data

- Some example unsupervised learning problems:
  - Clustering / Factor Analysis
  - Dimensionality Reduction / Embeddings
  - Density Estimation with Mixture Models

New Topic: Clustering

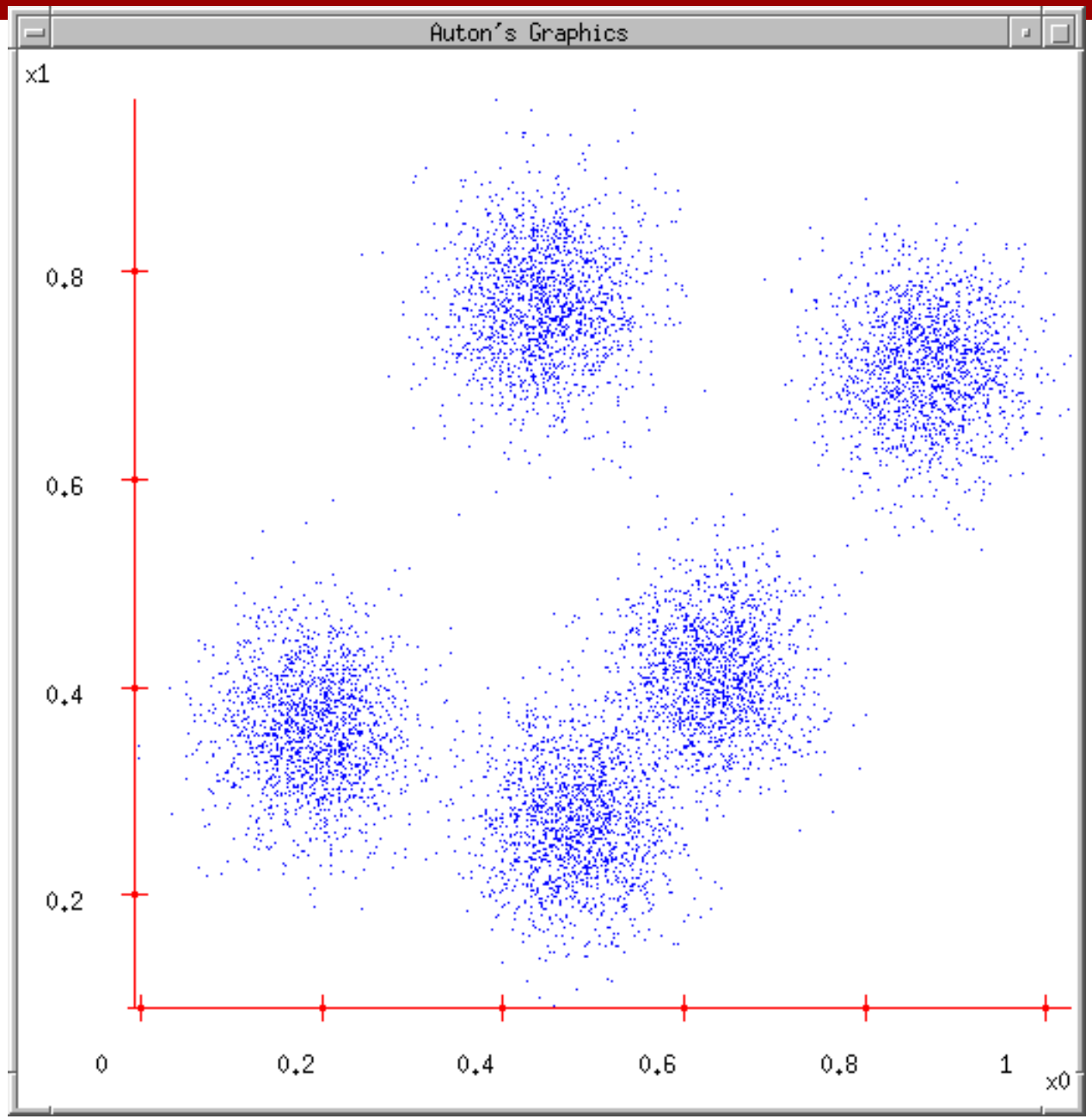


Slide Credit: Carlos Guestrin

Synonyms

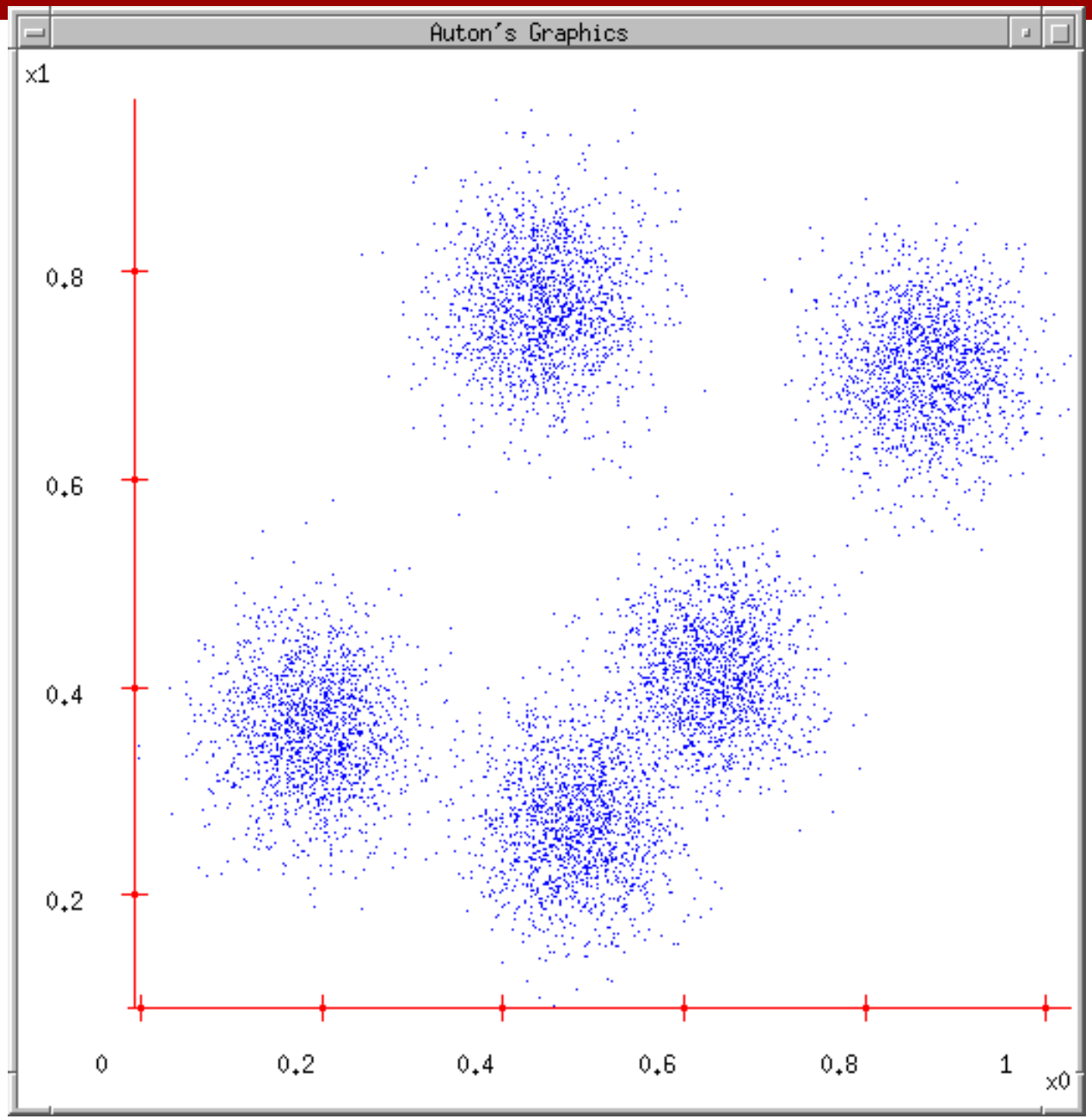
- Clustering
- Vector Quantization
- Latent Variable Models
- Hidden Variable Models
- Mixture Models
- Algorithms:
  - K-means
  - Expectation Maximization (EM)

Some Data



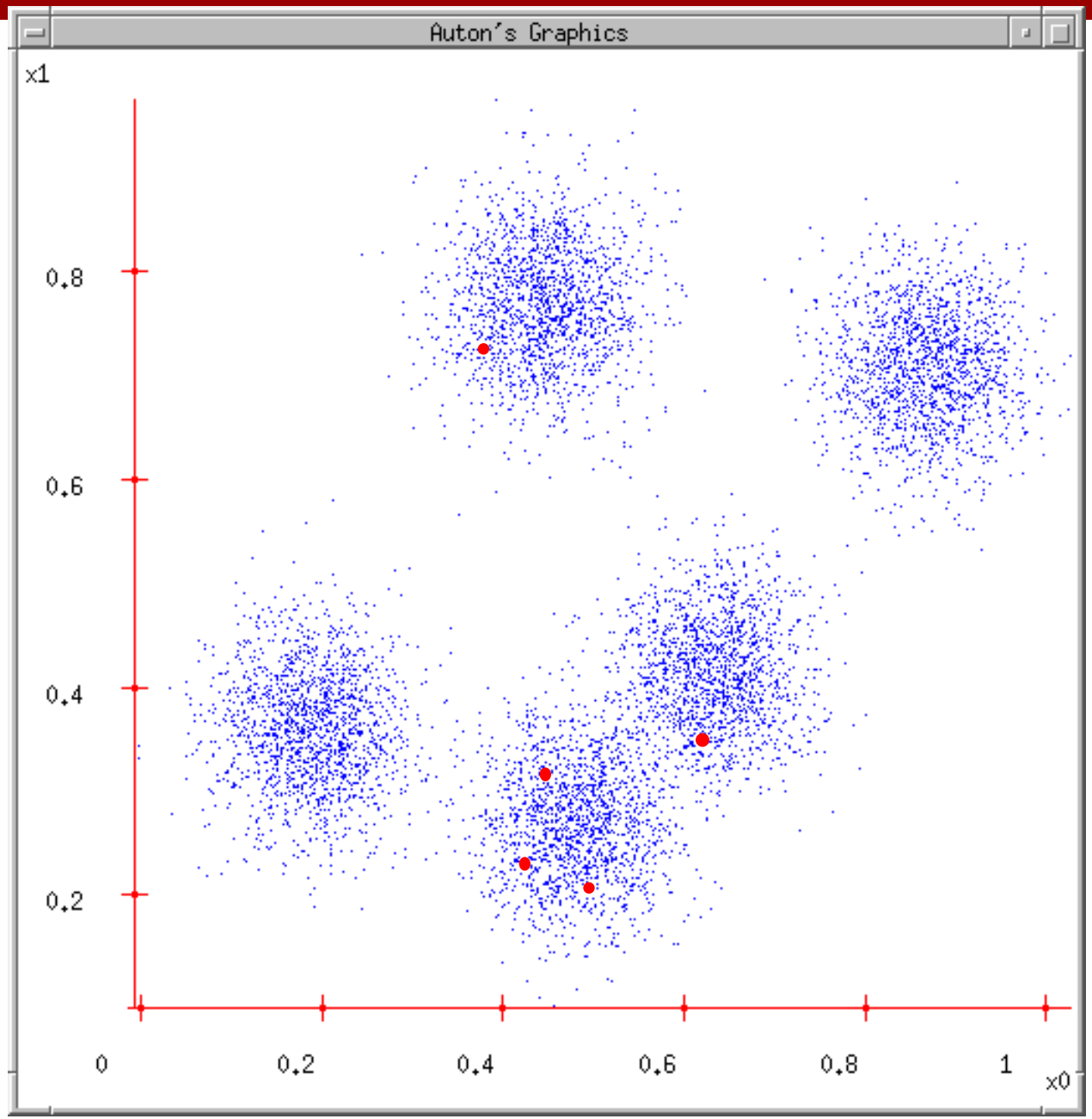
K-means

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



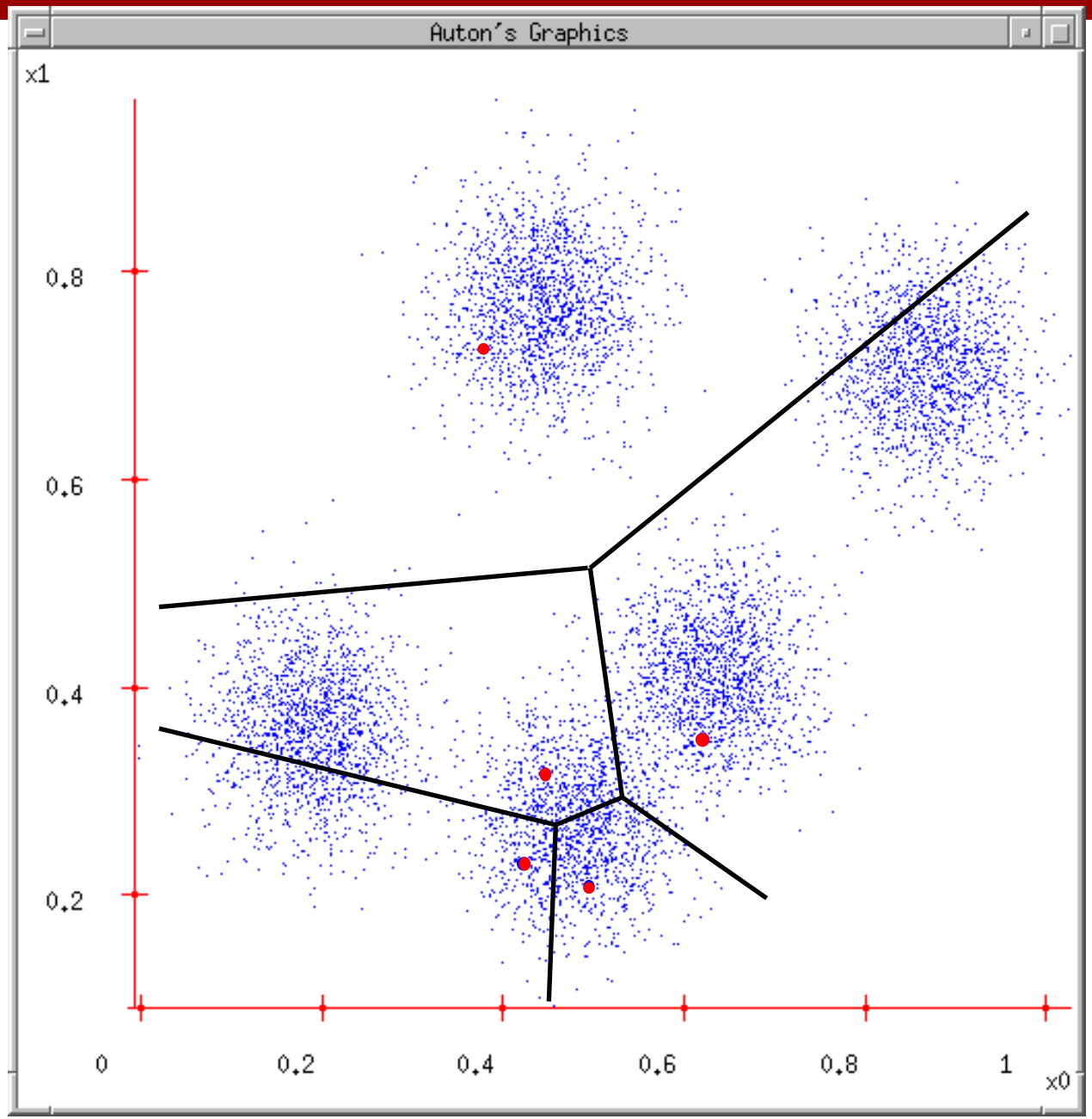
K-means

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



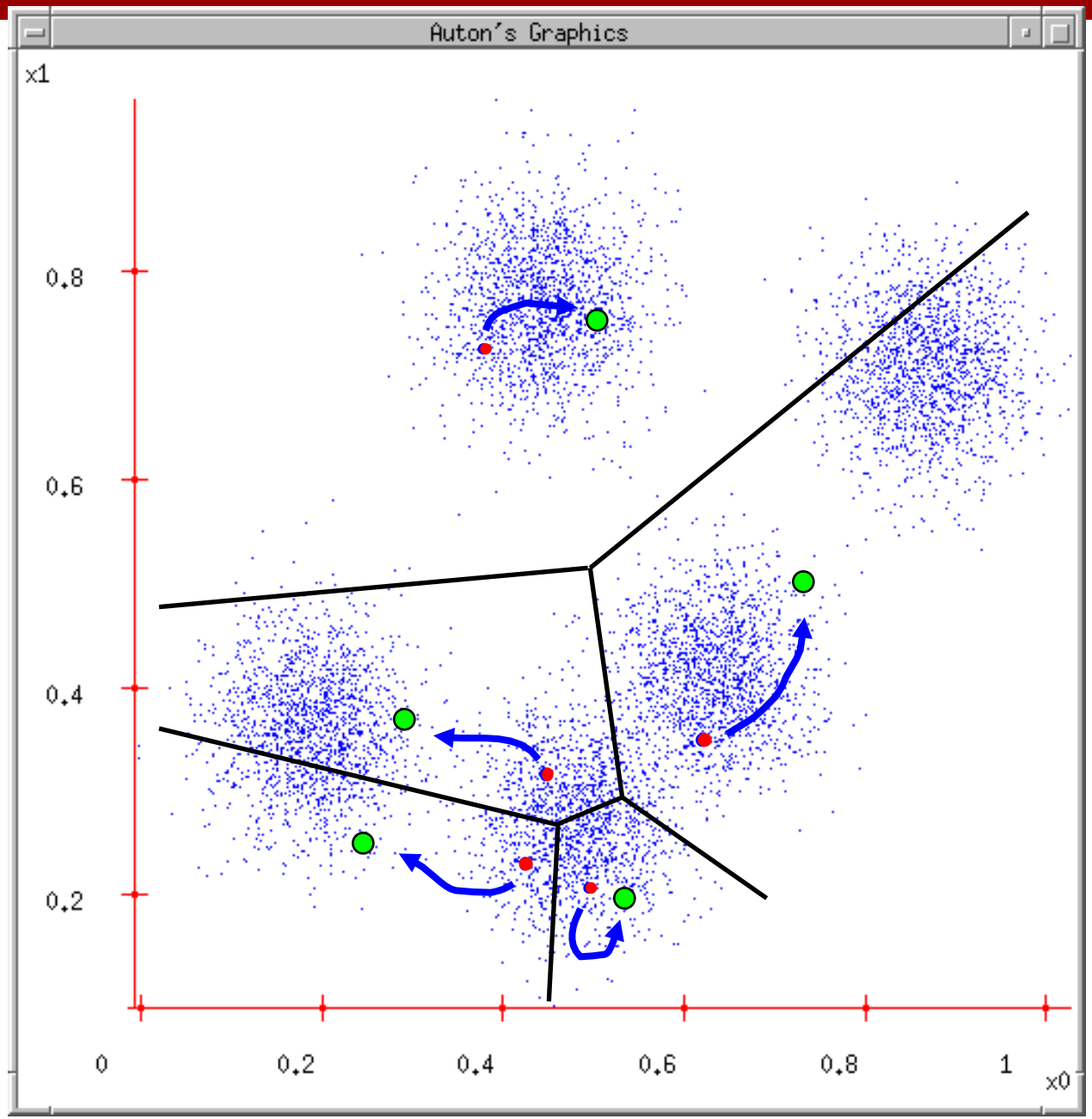
K-means

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)



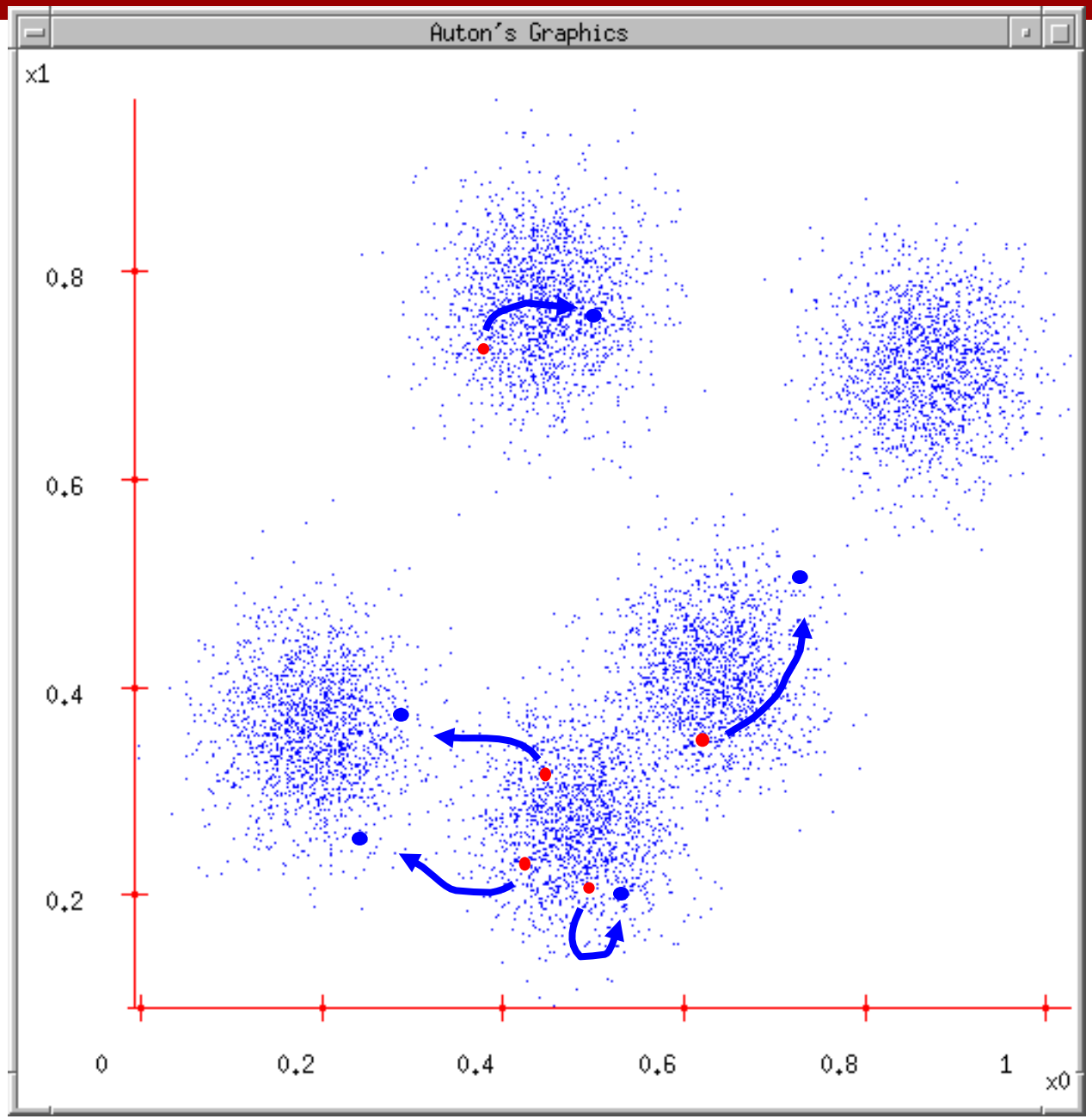
K-means

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. …Repeat until terminated!



K-means

- Randomly initialize *k* centers
  - $^{(0)} = {}_1^{(0)}, \ldots, {}_k^{(0)}$
- **Assign**:
  - Assign each point i {1,…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

- Demo
  - http://mlehman.github.io/kmeans-javascript/

What is K-means optimizing?

- Objective F(,C): function of centers and point allocations C:
  - $F(\boldsymbol{\mu}, C) = \sum_{i=1}^{N} ||\mathbf{x}_i - \boldsymbol{\mu}_{C(i)}||^2$
  - 1-of-k encoding $F(\boldsymbol{\mu}, \boldsymbol{a}) = \sum_{i=1}^{N} \sum_{j=1}^{k} a_{ij} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$
- Optimal K-means:
  - min $\min_a$ F(,a)

Coordinate descent algorithms

- 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!)

- K-means is a coordinate descent algorithm!

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 , 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

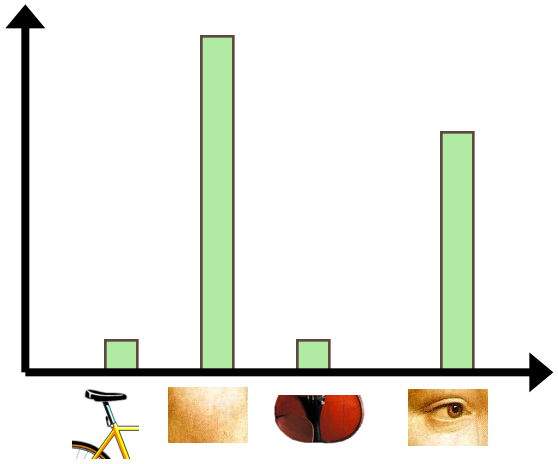
One important use of K-means

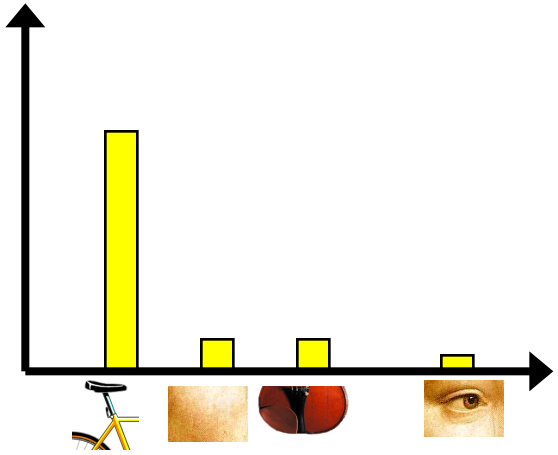
- Bag-of-word models in computer vision

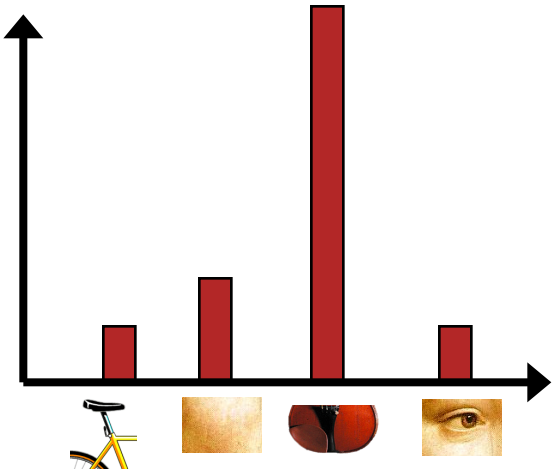
Bag of Words model

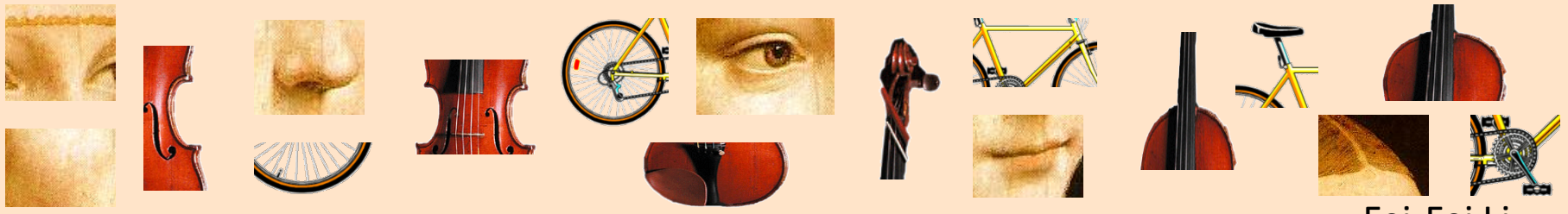
aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
...	
Zaire	0

Object → Bag of 'words'



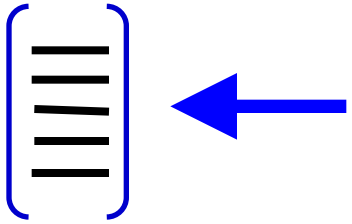






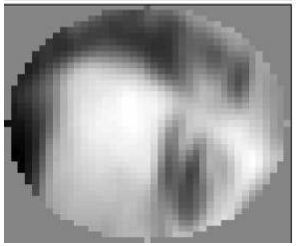
Fei-Fei Li

Interest Point Features

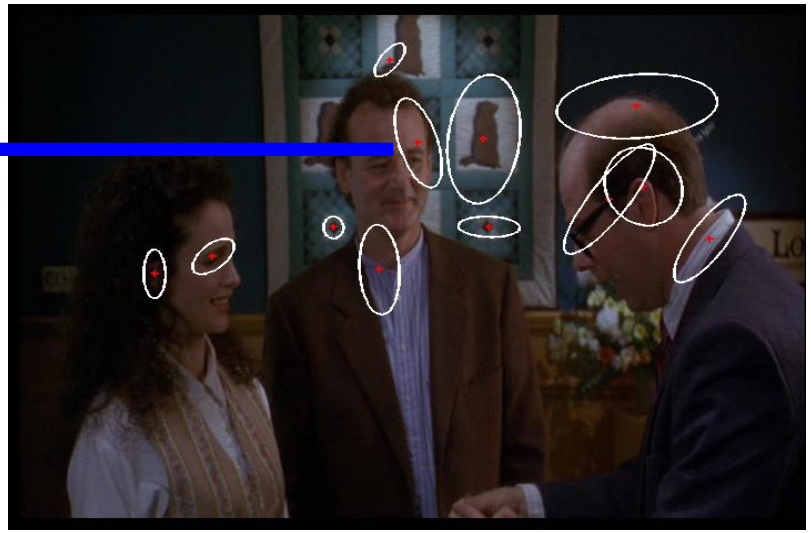


Compute SIFT descriptor

[Lowe'99]



Normalize patch



Detect patches

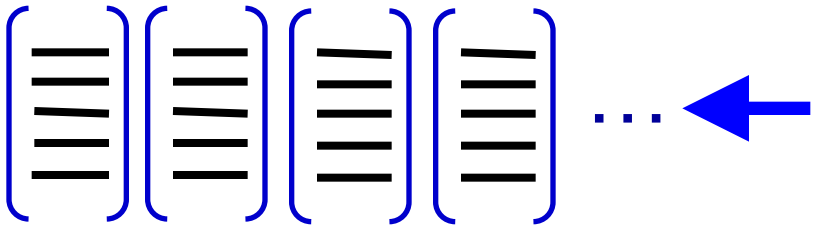
[Mikojaczyk and Schmid '02]

[Matas et al. '02]

[Sivic et al. '03]

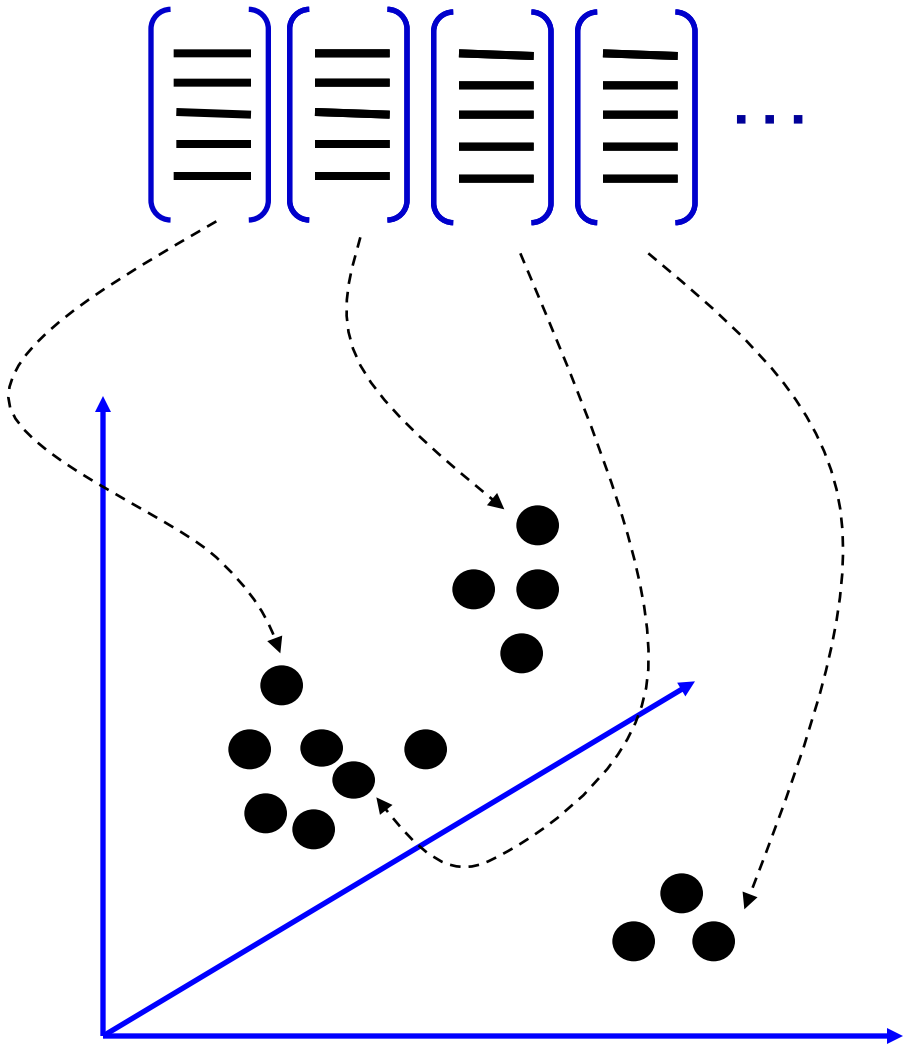
Slide credit: Josef Sivic

Patch Features



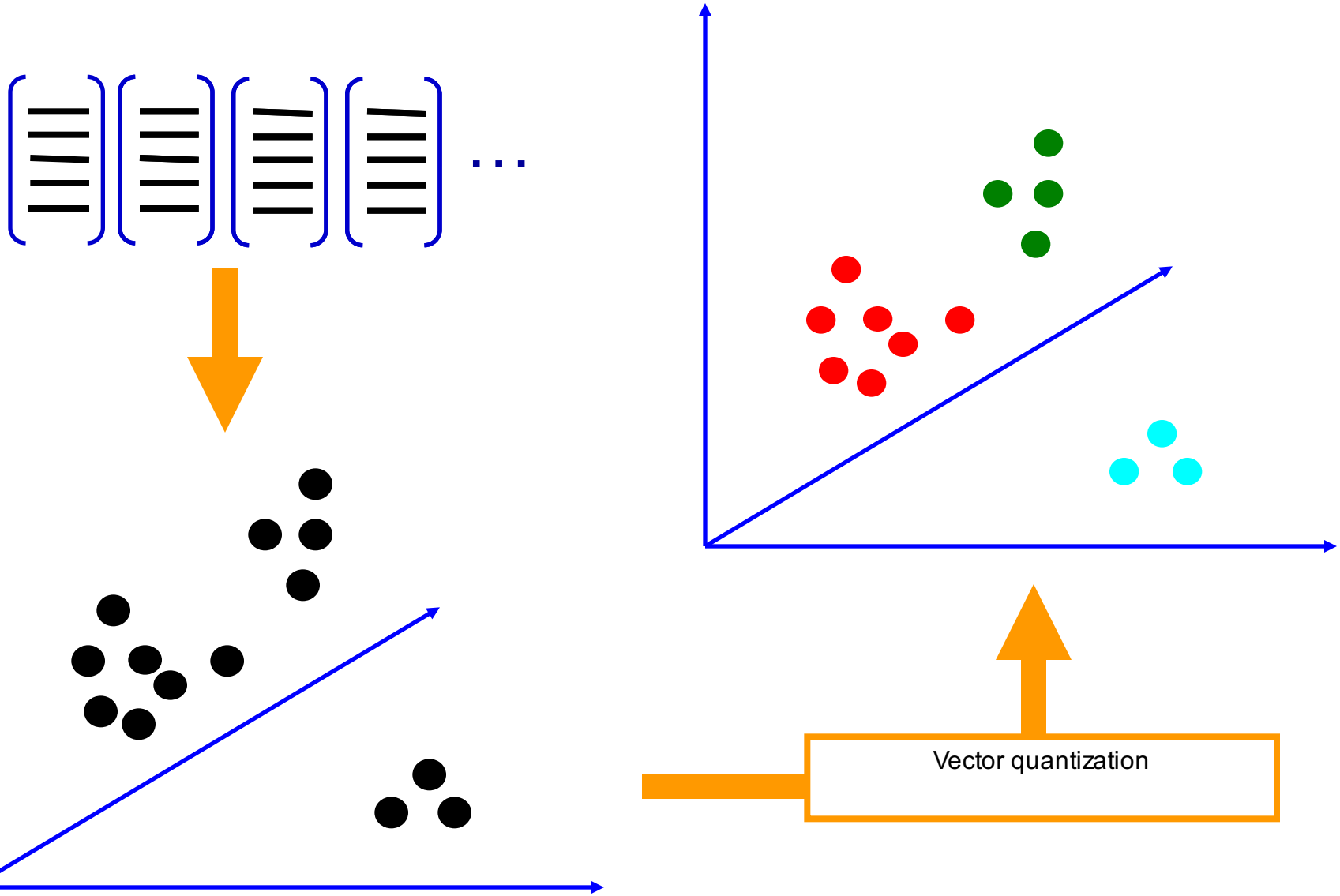
Slide credit: Josef Sivic

dictionary formation



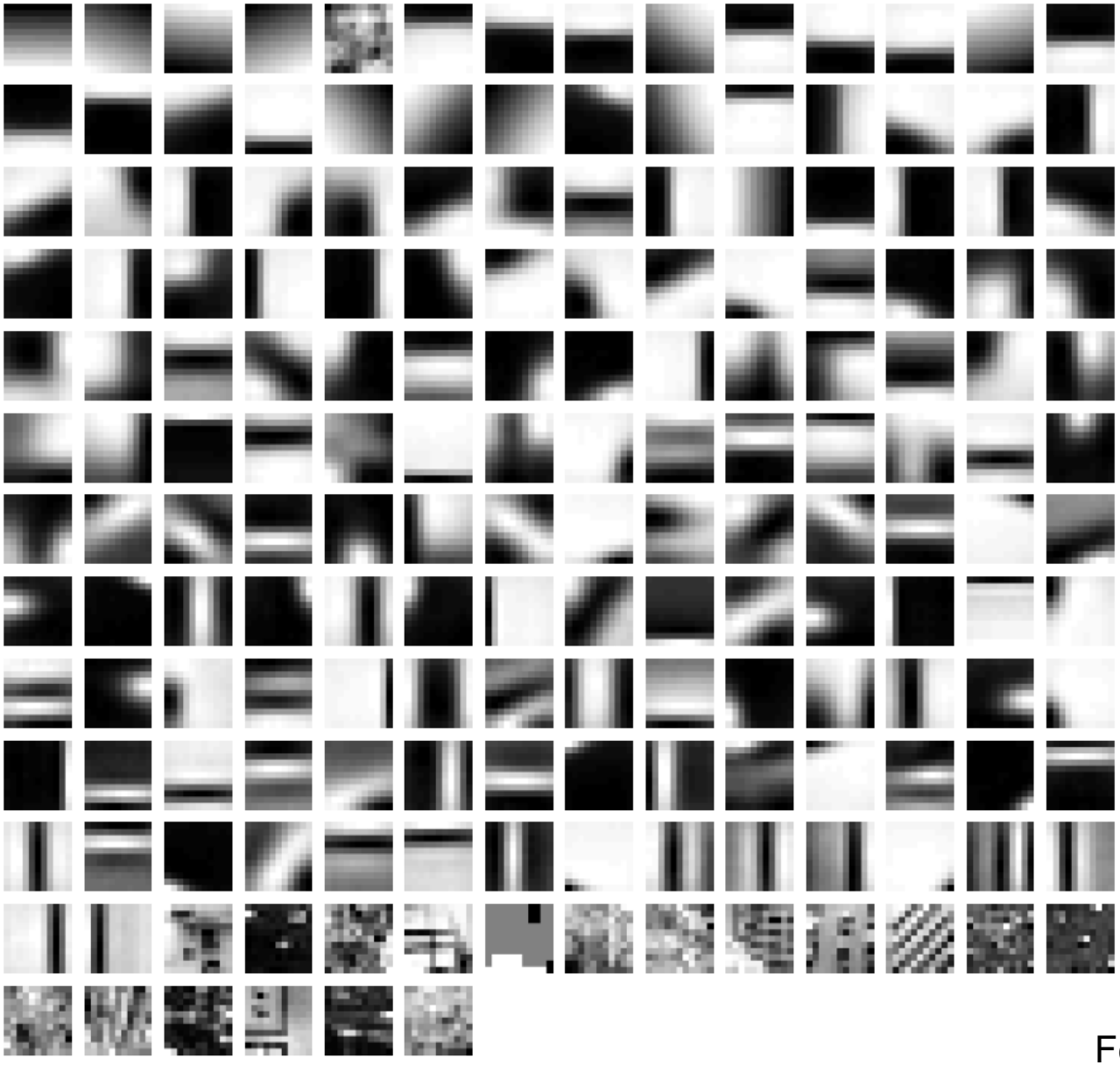
Slide credit: Josef Sivic

Clustering (usually k-means)



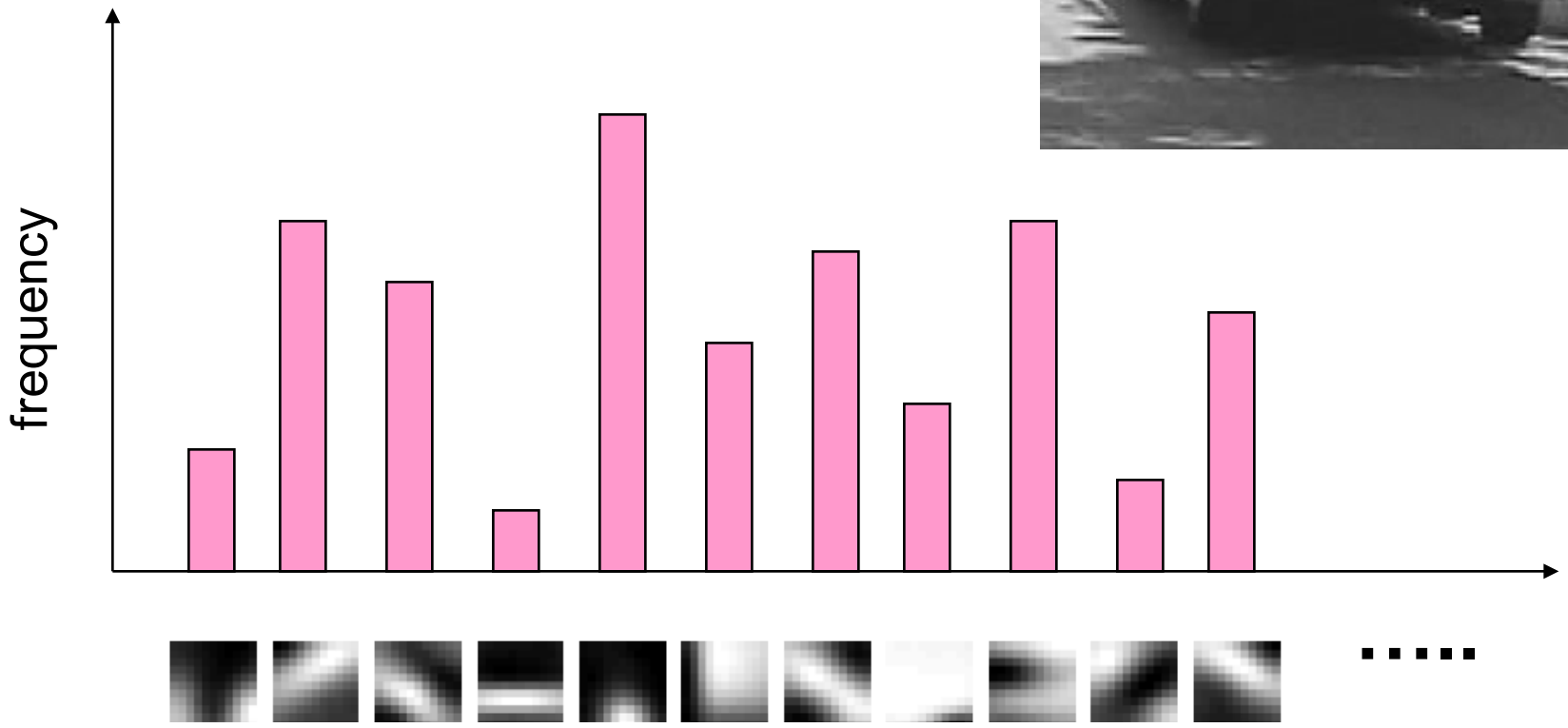
Slide credit: Josef Sivic

Clustered Image Patches

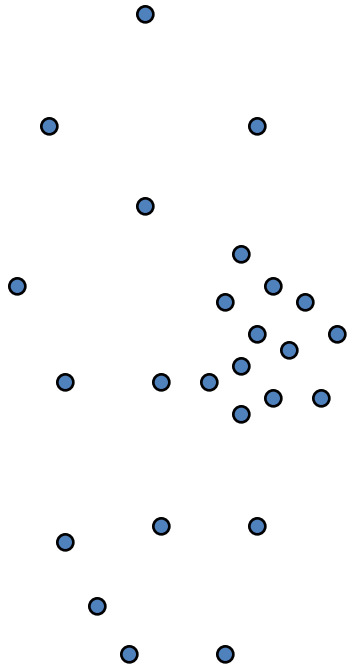


Fei-Fei et al. 2005

Image representation



(One) bad case for k-means



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

- GMM to the rescue!

GMM

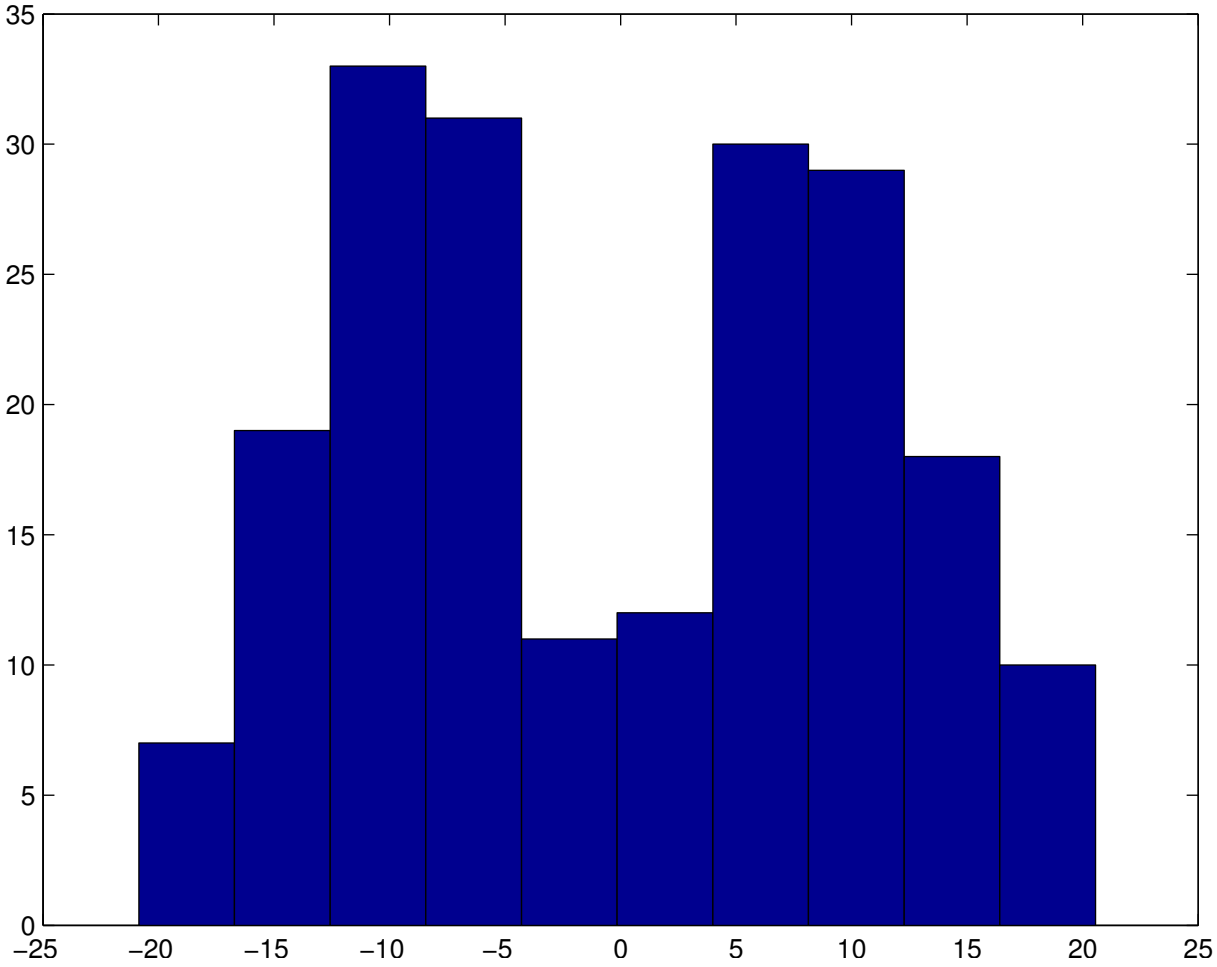
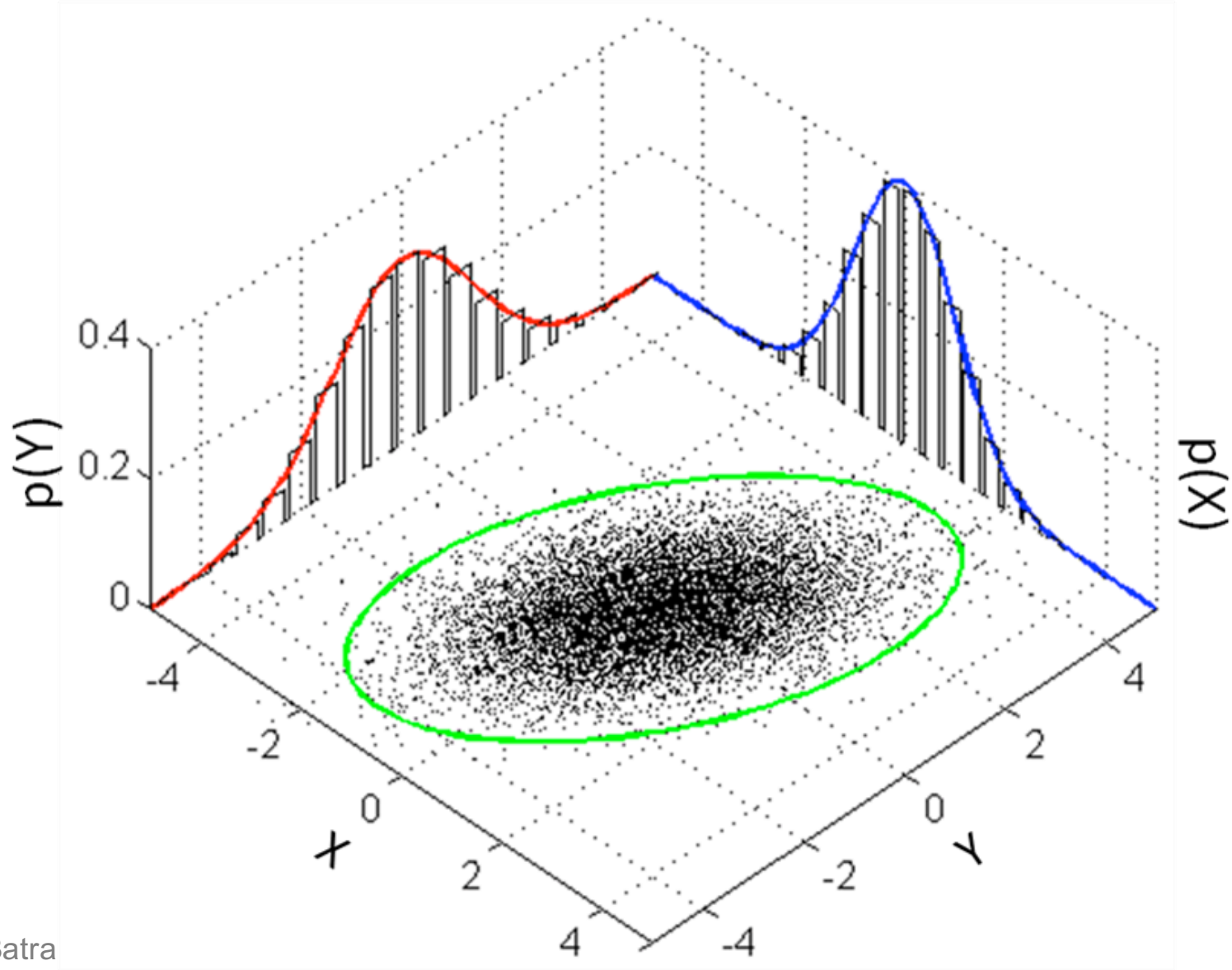
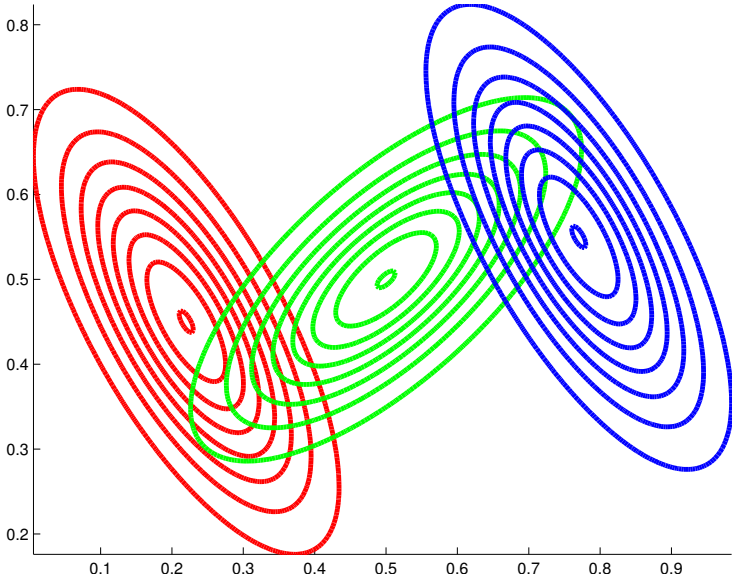


Figure Credit: Kevin Murphy

Recall Multi-variate Gaussians



GMM



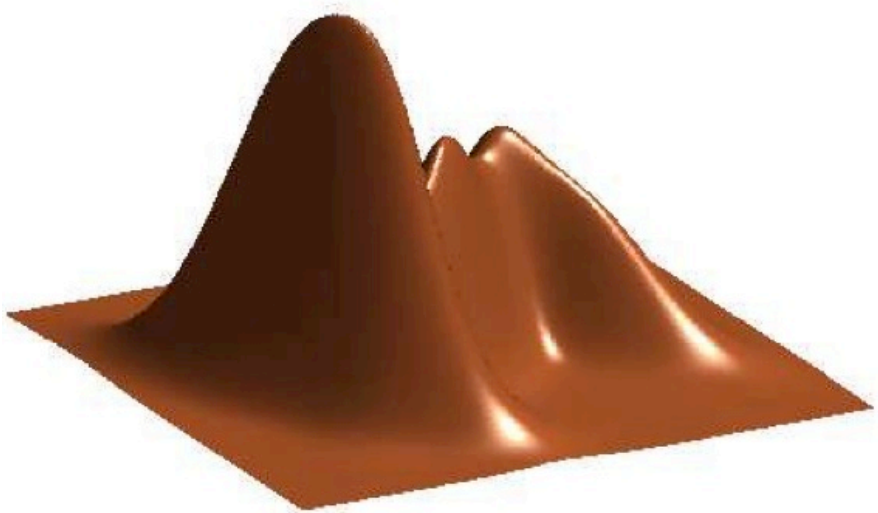


Figure Credit: Kevin Murphy

Hidden Data Causes Problems #1

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

GMM vs Gaussian Joint Bayes Classifier

- On Board
  - Observed Y vs Unobserved Z
  - Likelihood vs Marginal Likelihood

Hidden Data Causes Problems #2

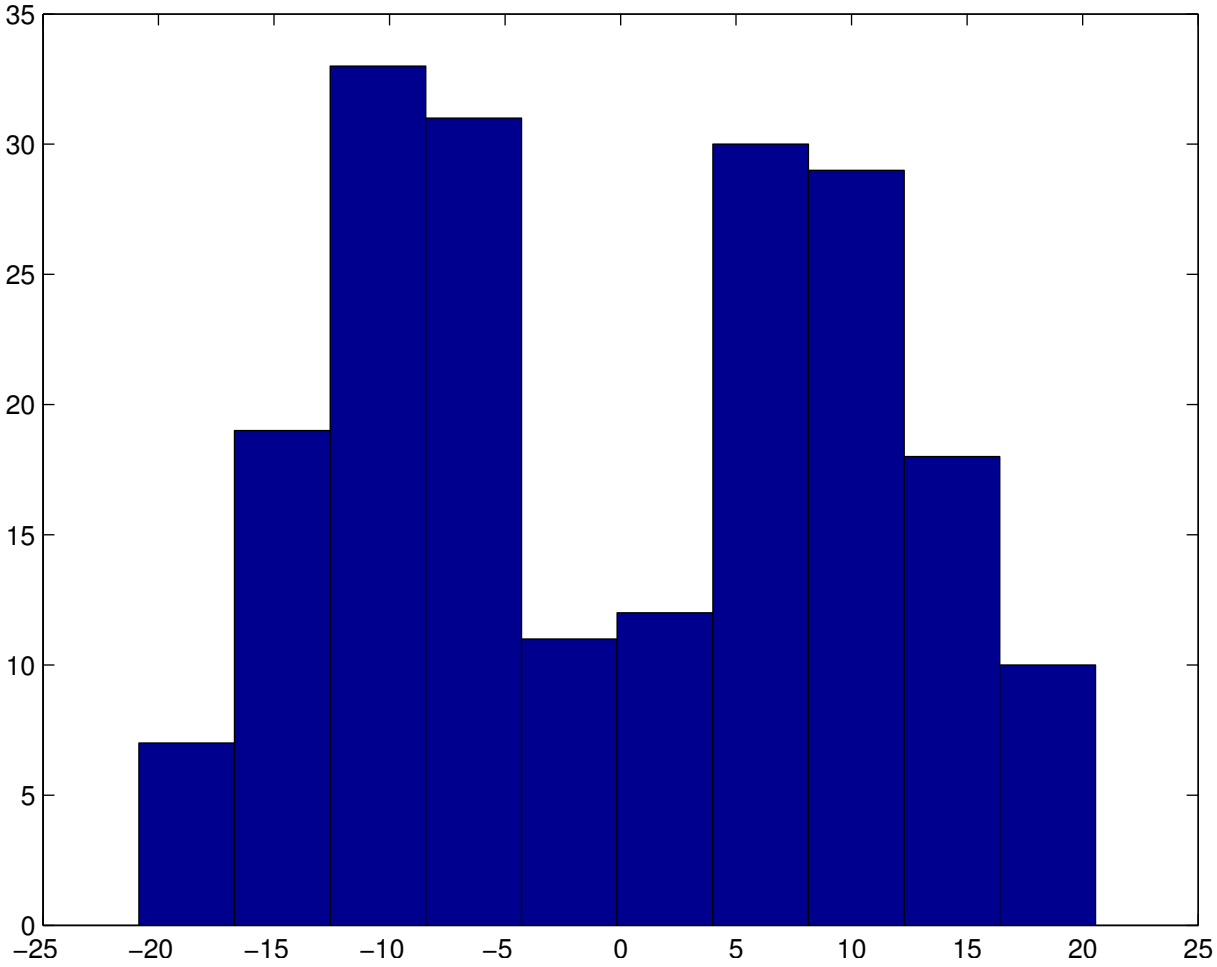
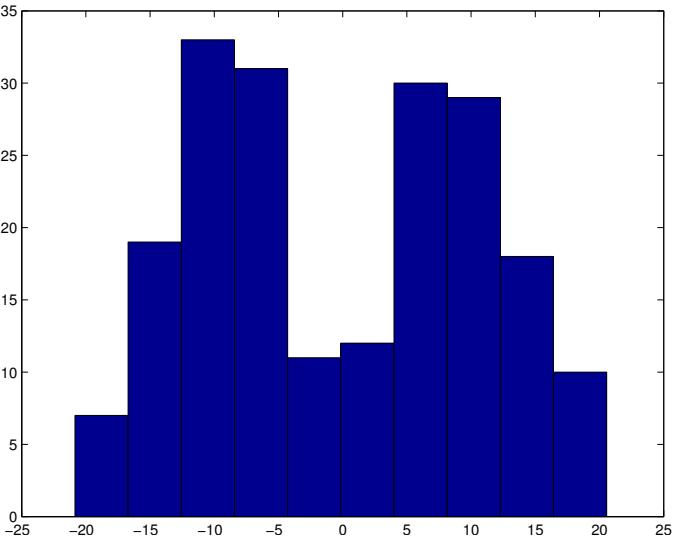


Figure Credit: Kevin Murphy

Hidden Data Causes Problems #2

- Identifiability



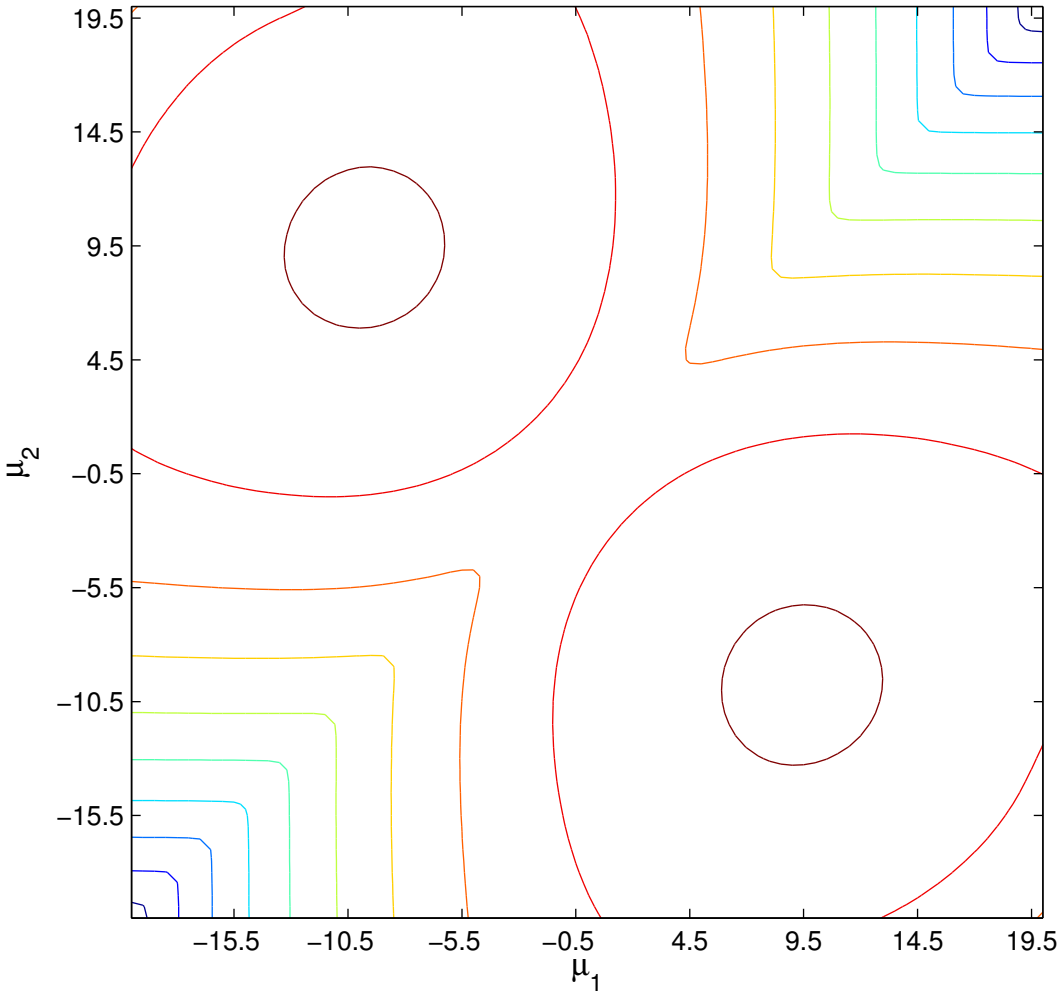
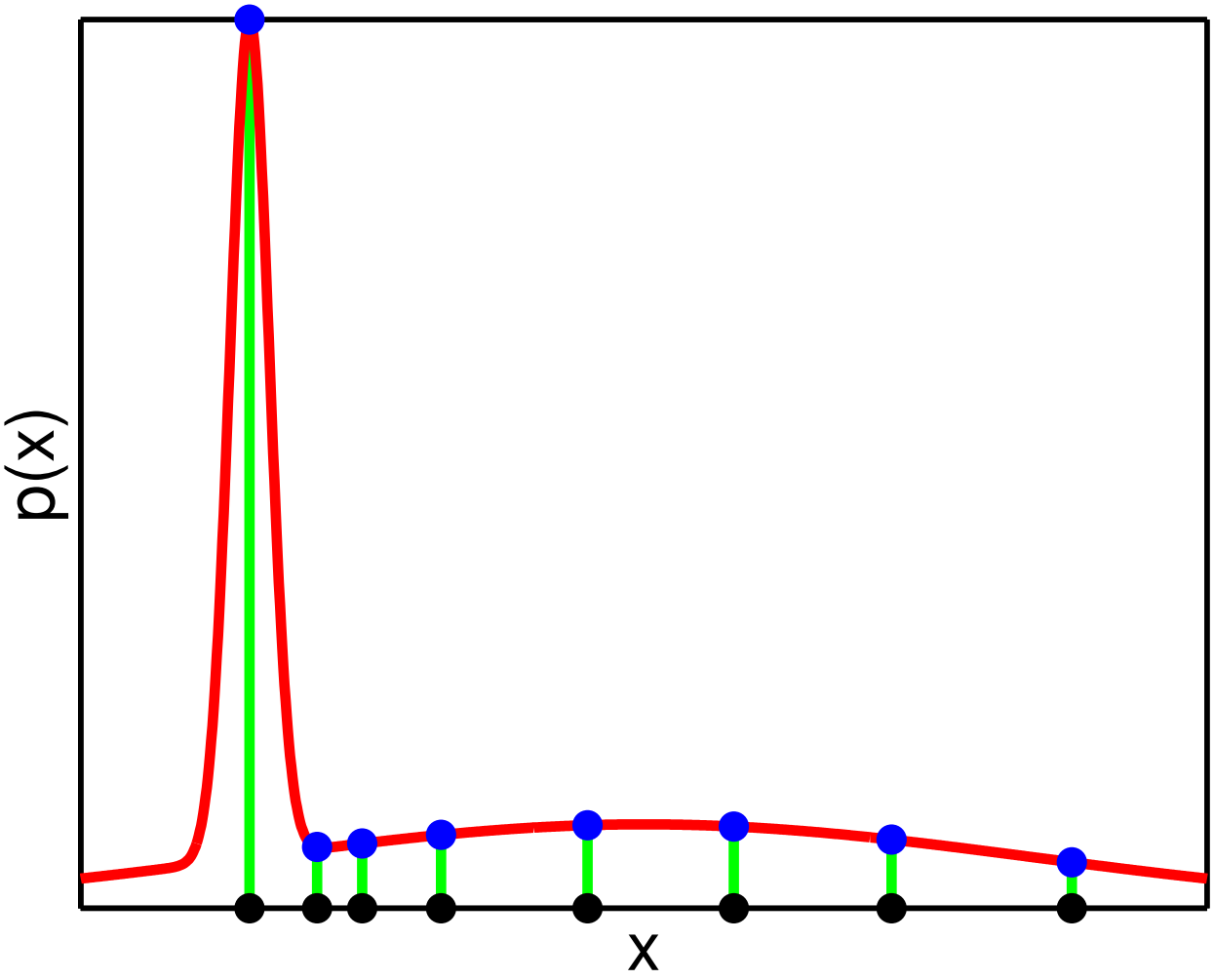


Figure Credit: Kevin Murphy

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 for all classes:

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

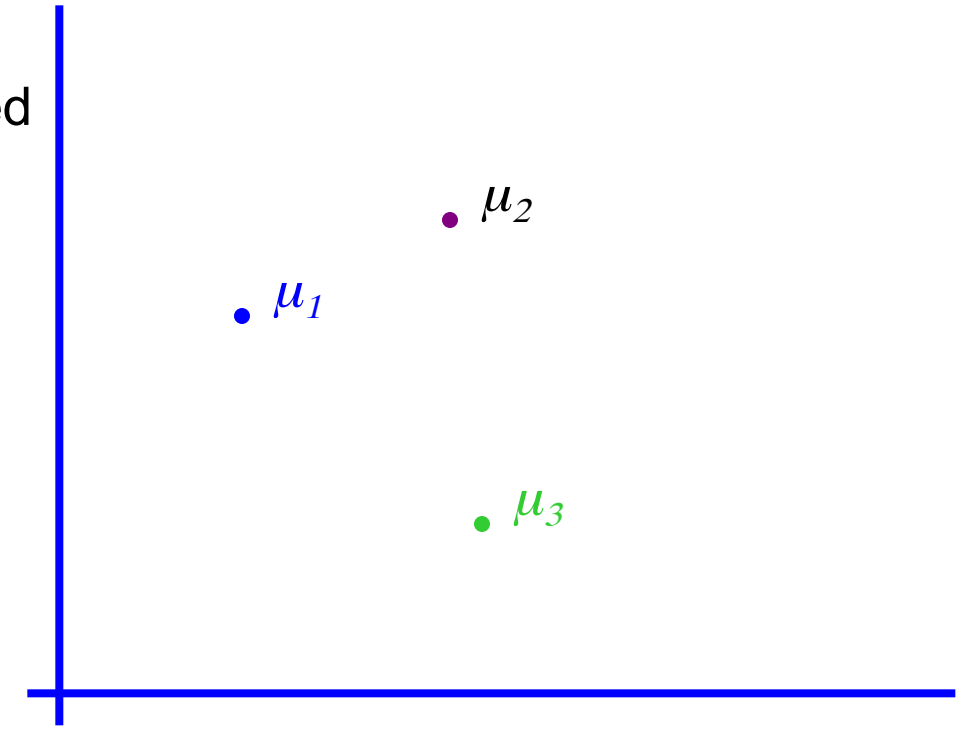
- If each x_i belongs to one class C(i) (hard assignment), marginal likelihood:

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

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

The K-means GMM assumption

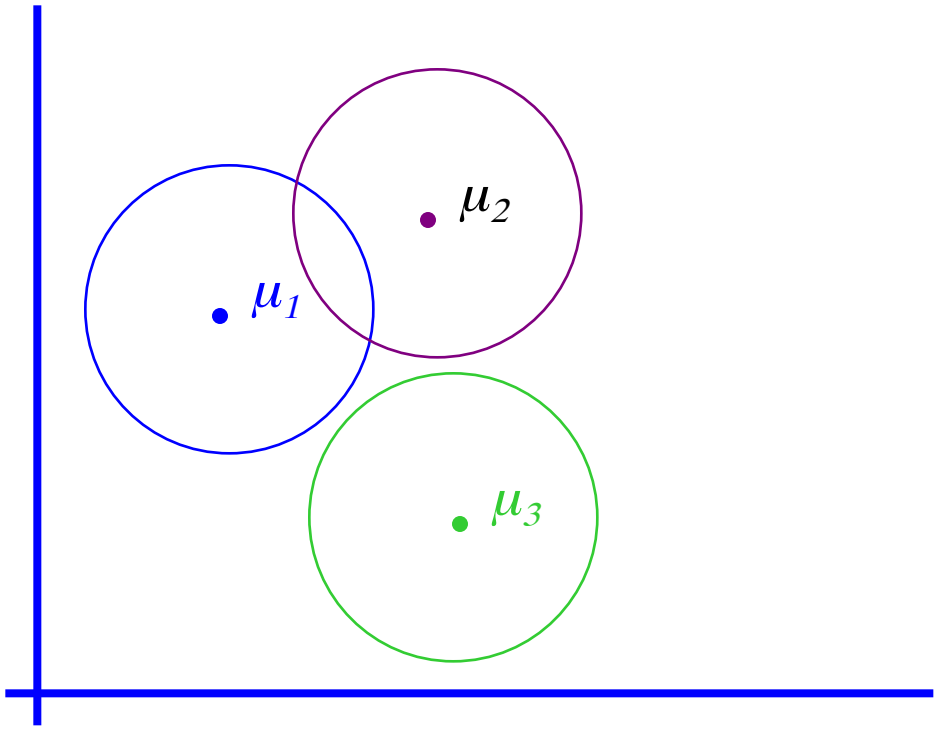
- There are k components
- Component *i* has an associated mean vector μ_i



The K-means GMM assumption

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

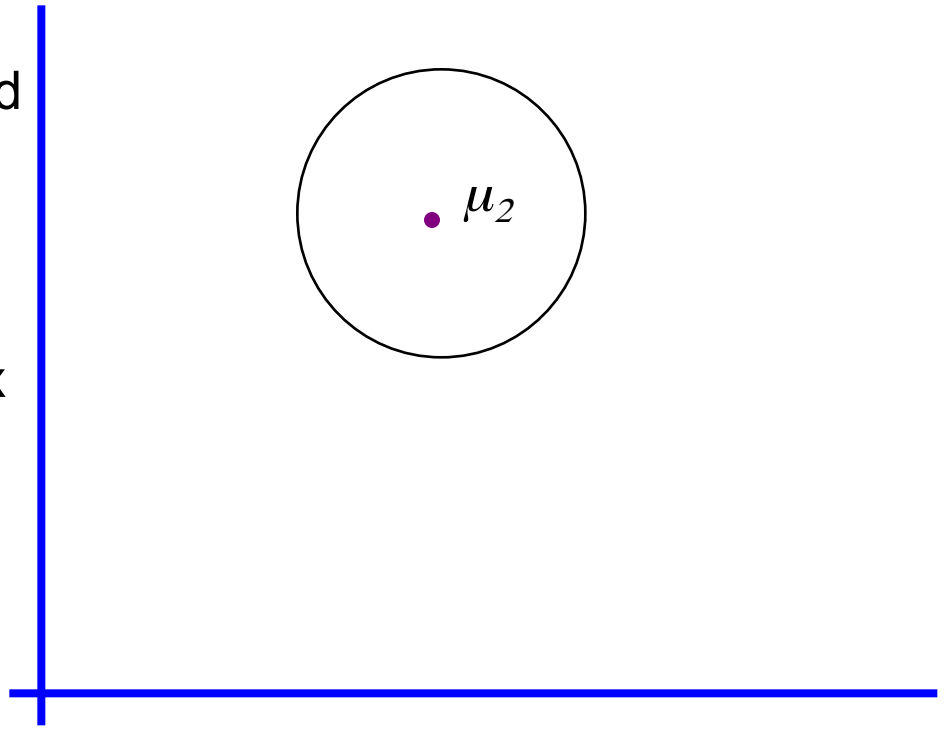


The K-means GMM assumption

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

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

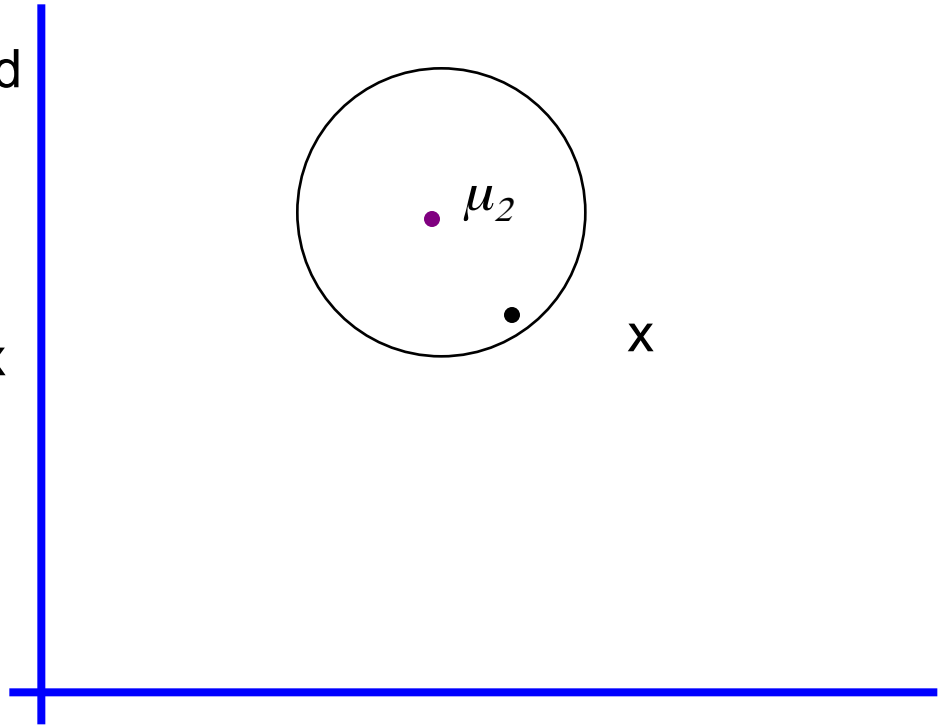


The K-means GMM assumption

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

1. Pick a component at random: Choose component i with probability *P(y=i)*
2. Datapoint ~ $\mathrm{N}(\mu_i, \sigma^2 \boldsymbol{I})$



The General GMM assumption

- There are k components
- Component *i* has an associated mean vector m_i
- 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 ~ N(m_i, Σ_i)