Clustering
K-means

Machine Learning – CSE546

Carlos Guestrin

University of Washington

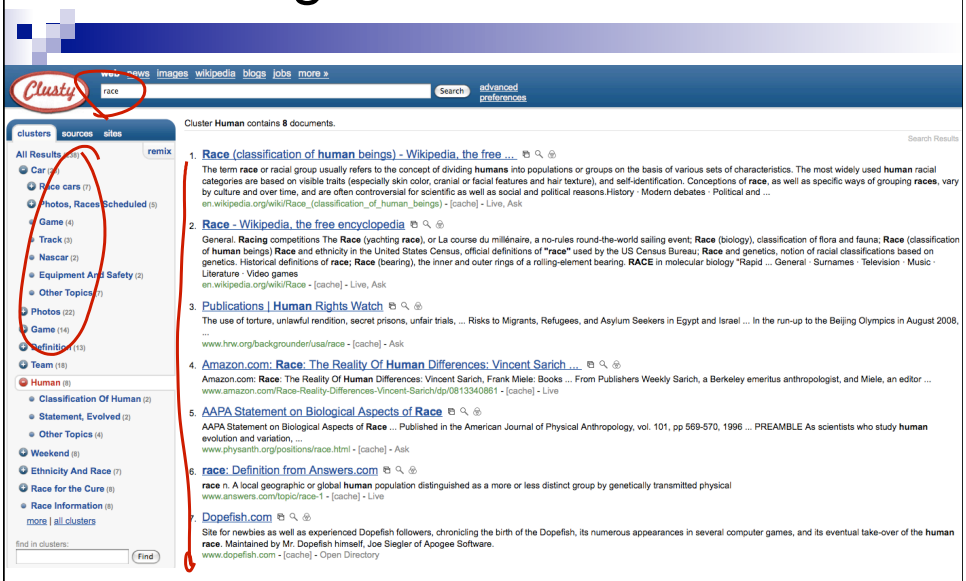
November 4, 2014

Clustering images

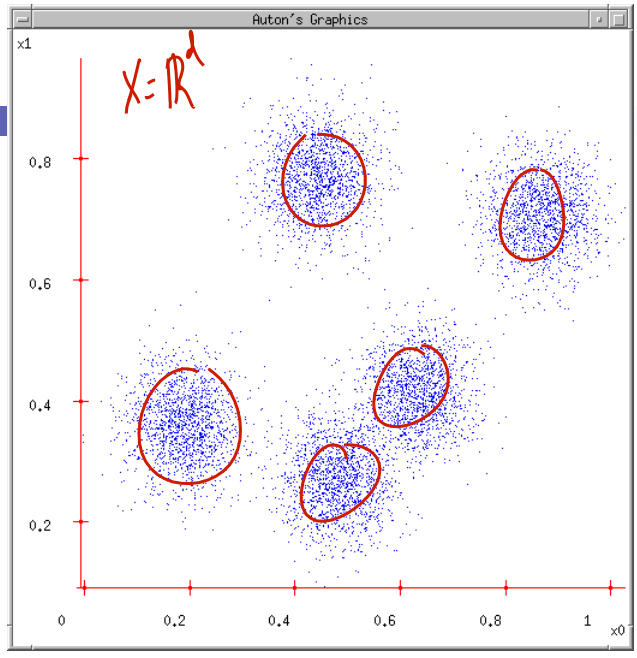
Set of Images

[Goldberger et al.]

Clustering web search results

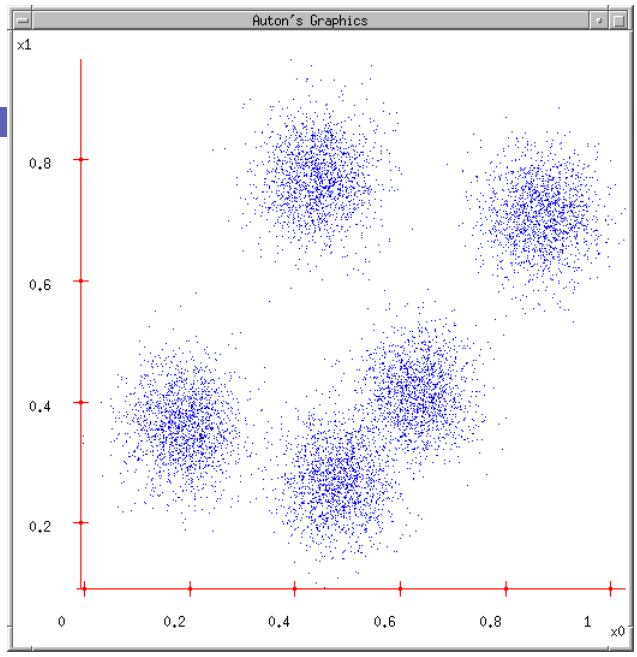


Some Data



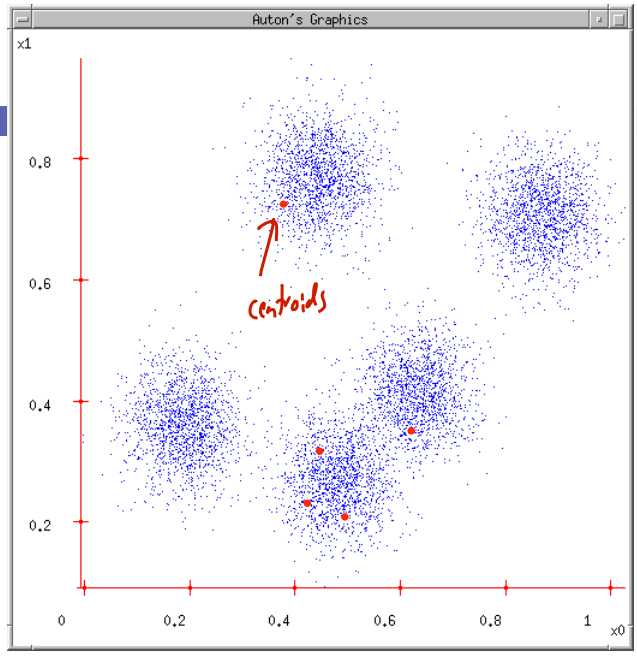
K-means

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



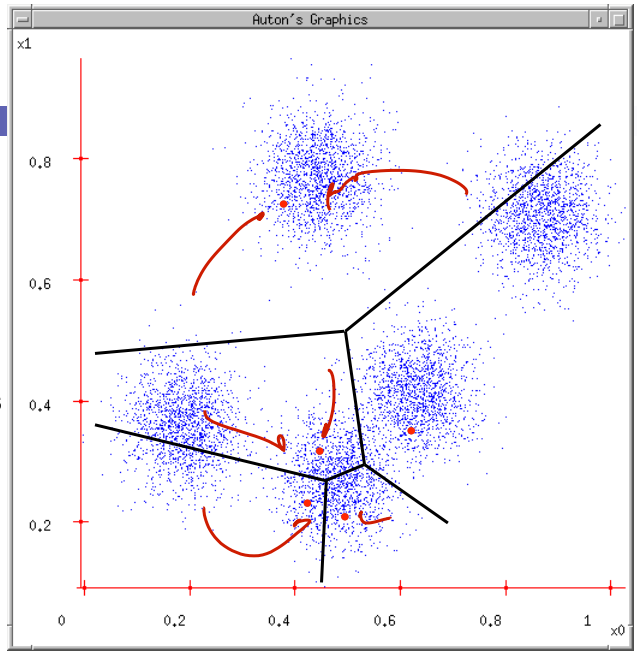
K-means

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



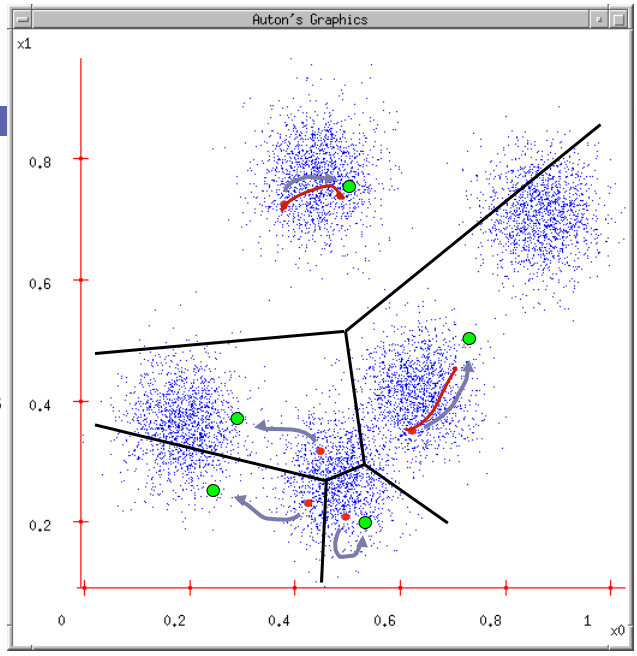
K-means

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)



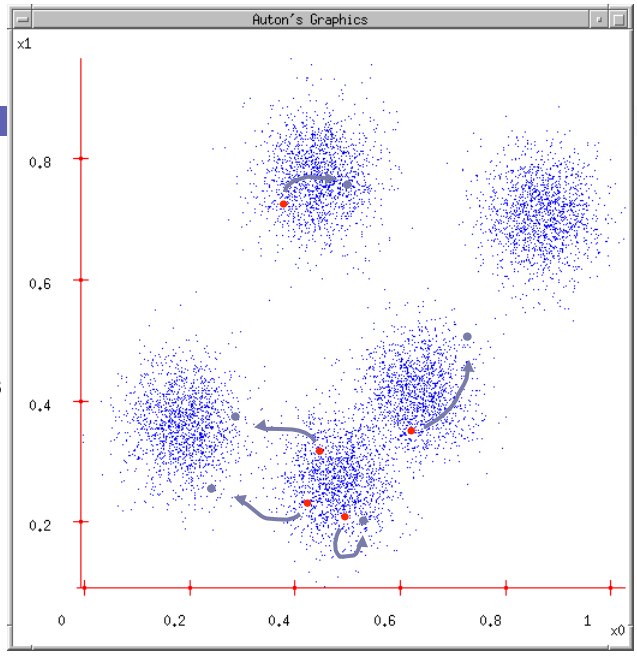
K-means

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

d-dim vectors

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)},\ldots, \mu_k^{(0)}$ iteration

or "smartly"

Repeat until convergence: no point changes cluster membership

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

fix μ, opt C

- **Recenter**: $\mu_i^{(t+1)}$ becomes centroid of its point:
  - $\mu_i^{(t+1)} \leftarrow \arg\min_\mu \sum_{j:C^{(t)}(j)=i} ||\mu - x_j||^2$ sum of points in cluster i

fix C, opt μ

$\mu_i^{(t+1)} = \frac{\sum_{j:C(j)=i} x_j}{|\{j: C(j)=i\}|}$

  - Equivalent to $\mu_i \leftarrow$ average of its points!

What is K-means optimizing?

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

centers ↑ ↑ allocations

Coordinate descent!!

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

↖ unfortunately, simultaneous optimization is very hard

Does K-means converge??? Part 1

- Optimize potential function:

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

↑ sum over clusters

- Fix $\mu^{(t)}$, optimize C

↑ sum over points in each cluster

$$\min_{C: C(1) \dots C(j) \dots C(N)} \sum_{j=1}^{N} ||\mu_{C(j)}^{(t)} - x_j||^2 = \min_{C(1)} \dots \min_{C(j)} \dots \min_{C(N)} \sum_{j=1}^{N} ||\mu_{C(j)}^{(t)} - x_j||^2$$

$$= \sum_{j=1}^{N} \min_{C(j)} ||\mu_{C(j)}^{(t)} - x_j||^2$$

indep. OPT problems

find closest center to x_j

⇒ "classification 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 $C^{(t)}$, optimize μ

$$\min_{\mu_1} \dots \min_{\mu_i} \dots \min_{\mu_k} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

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

recenter step

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)

random restarts help

$a, b \quad F(a,b)$
$\downarrow \quad \geq$
$a, b' \quad F(a,b')$
$\downarrow \quad \geq$
$a', b' \quad F(a',b')$

F bounded
$F \geq 0$

- K-means is a coordinate descent algorithm!

picking k:
- $F(\mu^*, C^*)$ decreases as k increases
=) can't use F to pick k
- other techniques, related to cross validation or regularization

Mixtures of Gaussians

Machine Learning – CSE546

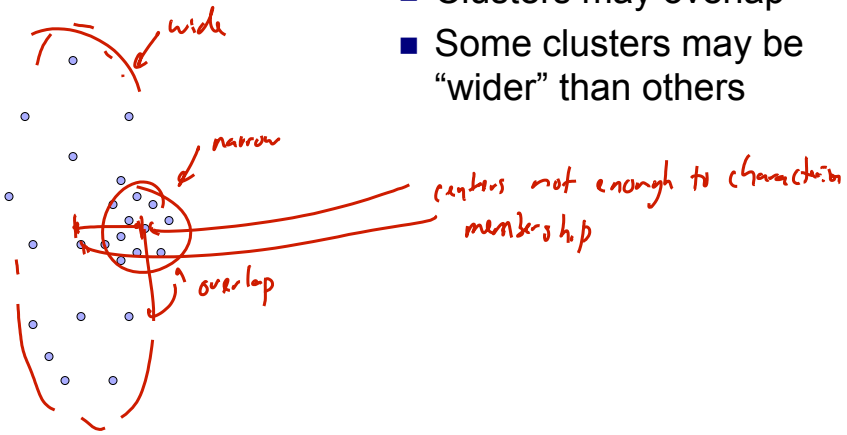
Carlos Guestrin

University of Washington

November 4, 2014

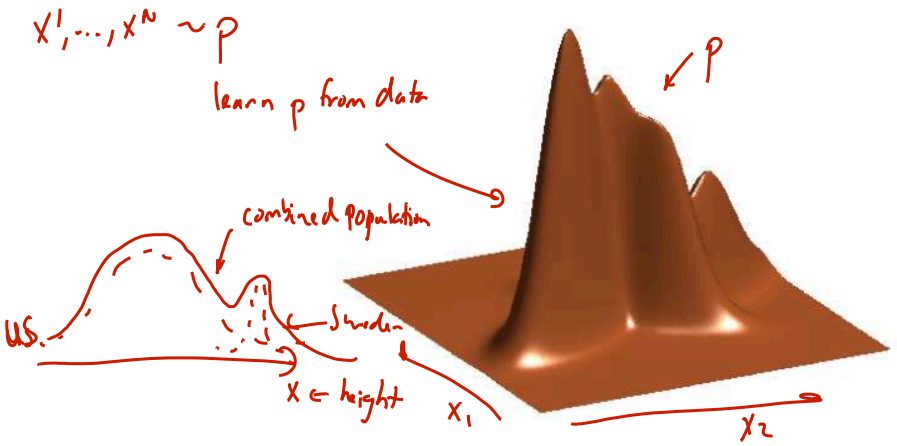
(One) bad case for k-means

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

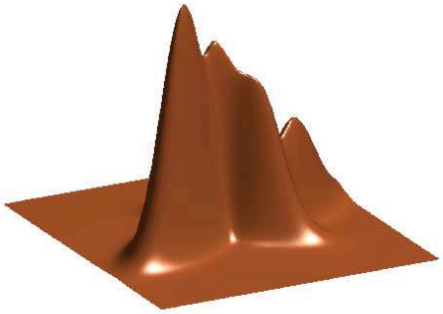


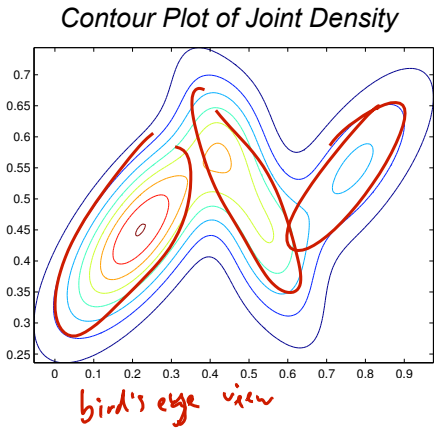
Density Estimation

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



Density Estimation

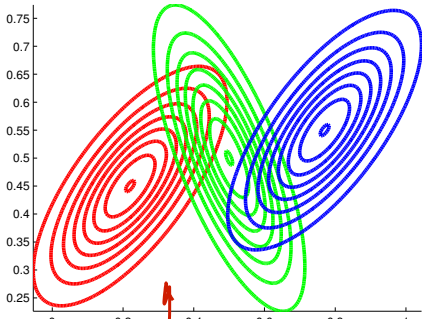




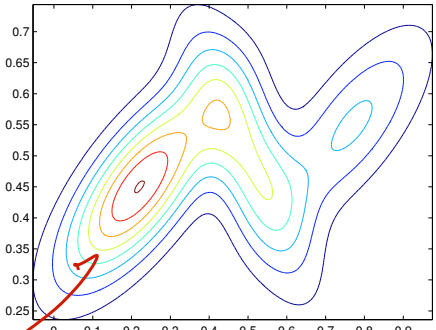
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



Contour Plot of Joint Density



original

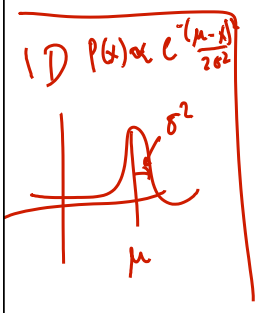
p sum with weights Π_i of each Gaussian

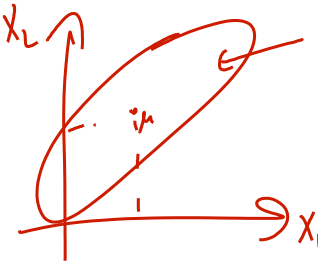
Gaussians in *d* Dimensions

mean vector

covariance matrix

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





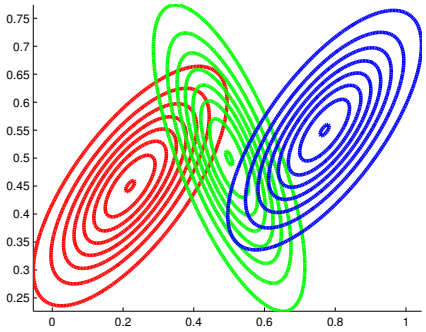
$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix}$$

$\sigma_{12} = \sigma_{21}$

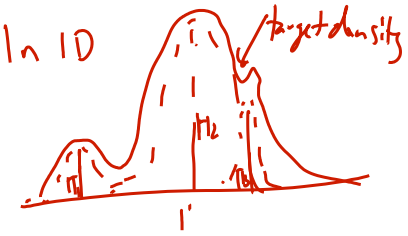
Density as Mixture of Gaussians

- Approximate density with a mixture of Gaussians

Mixture of 3 Gaussians



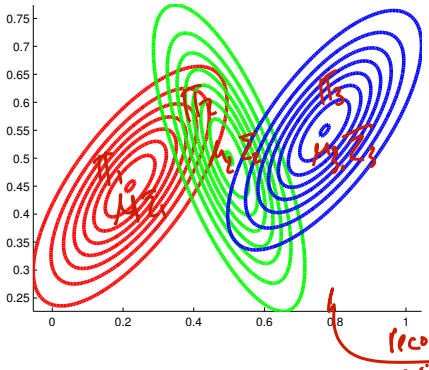
$$p(x^j|\pi,\mu,\Sigma) = \sum_{i=1}^{K} \pi_i \, N(x^j|\mu_i,\Sigma_i)$$



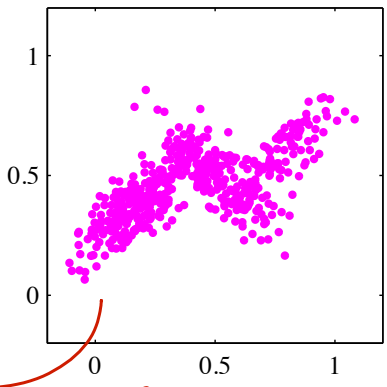
Density as Mixture of Gaussians

- Approximate with density with a mixture of Gaussians

Mixture of 3 Gaussians



Our actual observations



C. Bishop, Pattern Recognition & Machine Learning