

---

# Machine Learning

## Clustering

# Class So Far

---

- Representation
  - Decision Trees, Instances
- Evaluation
- Optimization
  - Hill Climbing, Genetic Algorithms
- Homework #3: Feature Design and “what ML tech do I use when”
  - plus some exercises on clustering, GAs, instance-based algorithms

# First, some epistemology

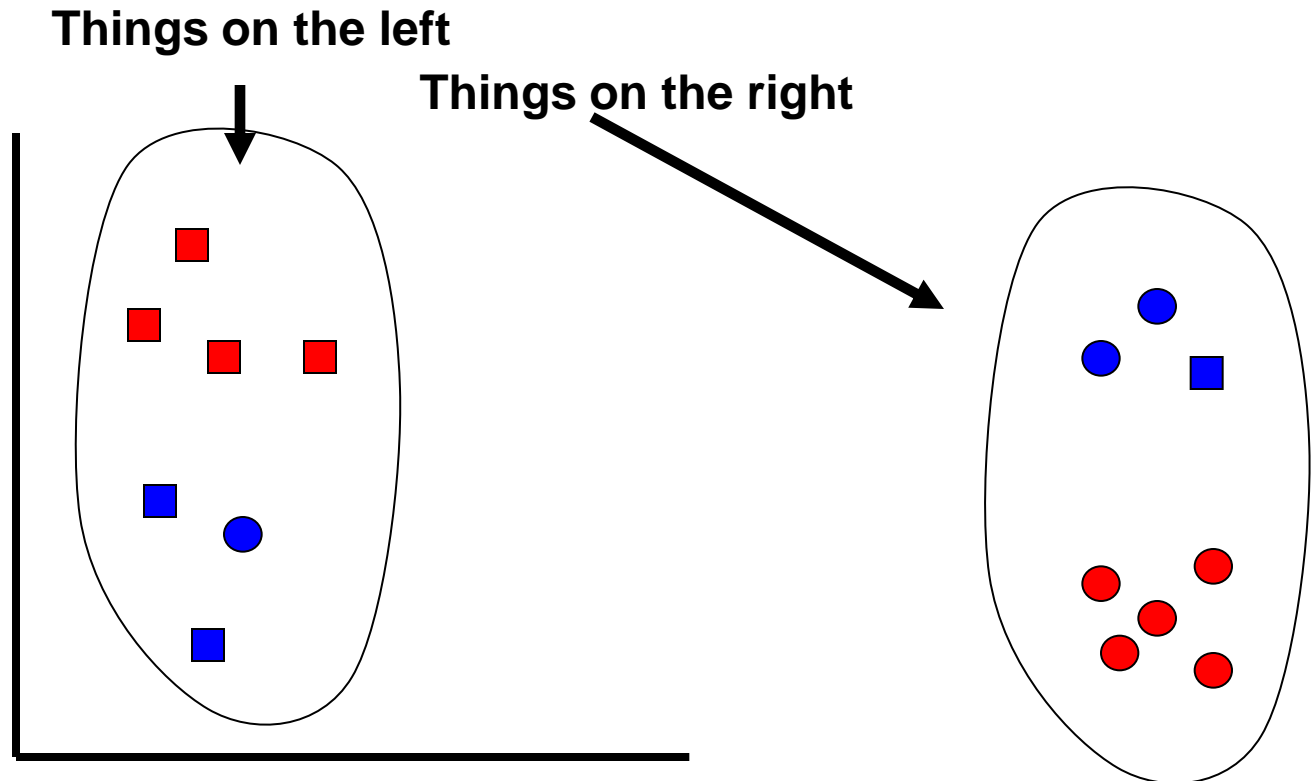
---

- There are known knowns. These are things we know that we know.
  - Databases!
- There are known unknowns. That is to say, there are things that we know we don't know.
  - Supervised Machine Learning
- But there are also unknown unknowns. There are things we don't know we don't know
  - Unsupervised Machine Learning (Clustering)

# Clustering

---

- Grouping data into (hopefully useful) sets.



# Clustering

---

- Unsupervised Learning
  - No labels
- Why do clustering?
  - Hypothesis Generation/Data Understanding
    - Clusters might suggest natural groups.
  - Visualization
  - Data pre-processing, e.g.:
    - Converting continuous attributes to nominal
    - For *efficiency*
      - Text Classification (e.g., search engines, Google Sets)

# Some definitions

---

- Let  $X$  be the dataset:

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

- An ***m-clustering*** of  $X$  is a partition of  $X$  into  $m$  sets (clusters)  $C_1, \dots, C_m$  such that:

1. Clusters are non - empty:  $C_i \neq \{\}, 1 \leq i \leq m$

2. Clusters cover all of  $X$ :  $\bigcup_{i=1}^m C_i = X$

3. Clusters do not overlap:  $C_i \cap C_j = \{\}, \text{if } j \neq i$

# How many possible clusters? (Stirling numbers)

---

Size of dataset  
↓

$$S(n, m) = \frac{1}{m!} \sum_{i=0}^m (-1)^{m-i} \binom{m}{i} i^n$$

↑  
Number of clusters

$$S(15, 3) = 2,375,101$$

$$S(20, 4) = 45,232,115,901$$

$$S(100, 5) \approx 10^{68}$$

# What does this mean?

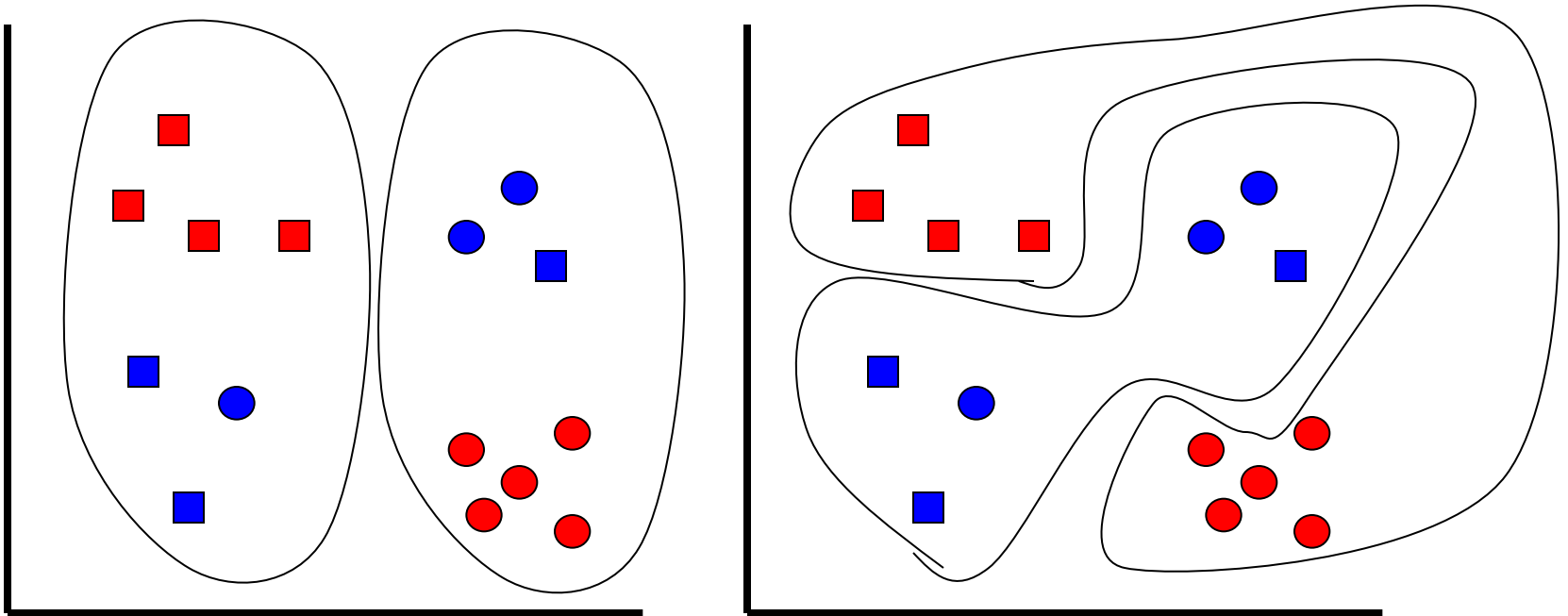
---

- We can't try all possible clusterings.
- Clustering algorithms look at a small fraction of all partitions of the data.
- The exact partitions tried depend on the kind of clustering used.

# Who is right?

---

- Different techniques cluster the same data set DIFFERENTLY.
- Who is right? Is there a “right” clustering?



# Steps in Clustering

---

- Select Features
- Define a Proximity Measure
- Choose a Clustering Algorithm
- Validate the Results
- Interpret the Results

# Kinds of Clustering

---

- Sequential
  - Fast
  - Results depend on data order
- Cost Optimization
  - Fixed number of clusters (typically)
  - Probabilistic models
- Hierarchical
  - Start with many clusters
  - join clusters at each step

# A Sequential Clustering Method

---

- Basic Sequential Algorithmic Scheme (BSAS)
  - S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, London England, 1999
- Assumption: The number of clusters is not known in advance.
- Let:
  - $d(x,C)$  be the *distance* between feature vector  $x$  and cluster  $C$ .
  - $\Theta$  be the *threshold of dissimilarity*
  - $q$  be the *maximum number of clusters*

# BSAS Pseudo Code

---

$m = 1$

$C_1 = \{x_1\}$

For  $i = 2$  to  $n$

Find  $C_k : d(x_i, C_k) = \min_{\forall j} d(x_i, C_j)$

If  $(d(x_i, C_k) > \Theta)$  and  $(m < q)$

$m = m + 1$

$C_m = \{x_i\}$

Else

$C_k = C_k \cup \{x_i\}$

End

End

# **Where is the cluster, exactly?**

---

$d(x, C)$  = distance from  $x$  to  $C$

How to compute?

# BSAS Characteristics

---

## Advantages

Fast! Only examine each data point once  
(takes  $O(nq)$ )

Number of clusters tuned from data

## Disadvantages

Must set  $q$ ,  $\Theta$

Sensitive to initial conditions

# Kinds of Clustering

---

- Sequential
  - Fast
  - Results depend on data order
- Cost Optimization
  - Fixed number of clusters (typically)
  - Often probabilistic models
- Hierarchical
  - Start with many clusters
  - join clusters at each step

# A Cost-optimization method

---

- K-means clustering

- J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*", Berkeley, University of California Press, 1:281-297

- A greedy algorithm

- Partitions  $n$  samples into  $k$  clusters

- minimizes the sum of the squared distances to the cluster centers

# The K-means algorithm

---

- Place  $K$  points into the space represented by the objects that are being clustered. These points represent initial group centroids (means).
- Assign each object to the group that has the closest centroid (mean).
- When all objects have been assigned, recalculate the positions of the  $K$  centroids (means).
- Repeat Steps 2 and 3 until the centroids no longer move.

# K-means clustering

---

- The way to initialize the mean values is not specified.
  - Randomly choose  $k$  samples?
- Results depend on the initial means
  - Try multiple starting points?
- Assumes  $K$  is known.
  - How do we choose this?

# Mixture Models

$$P(x) = \sum_{i=1}^{n_c} P(c_i)P(x|c_i)$$

**Objective function:** Log likelihood of data

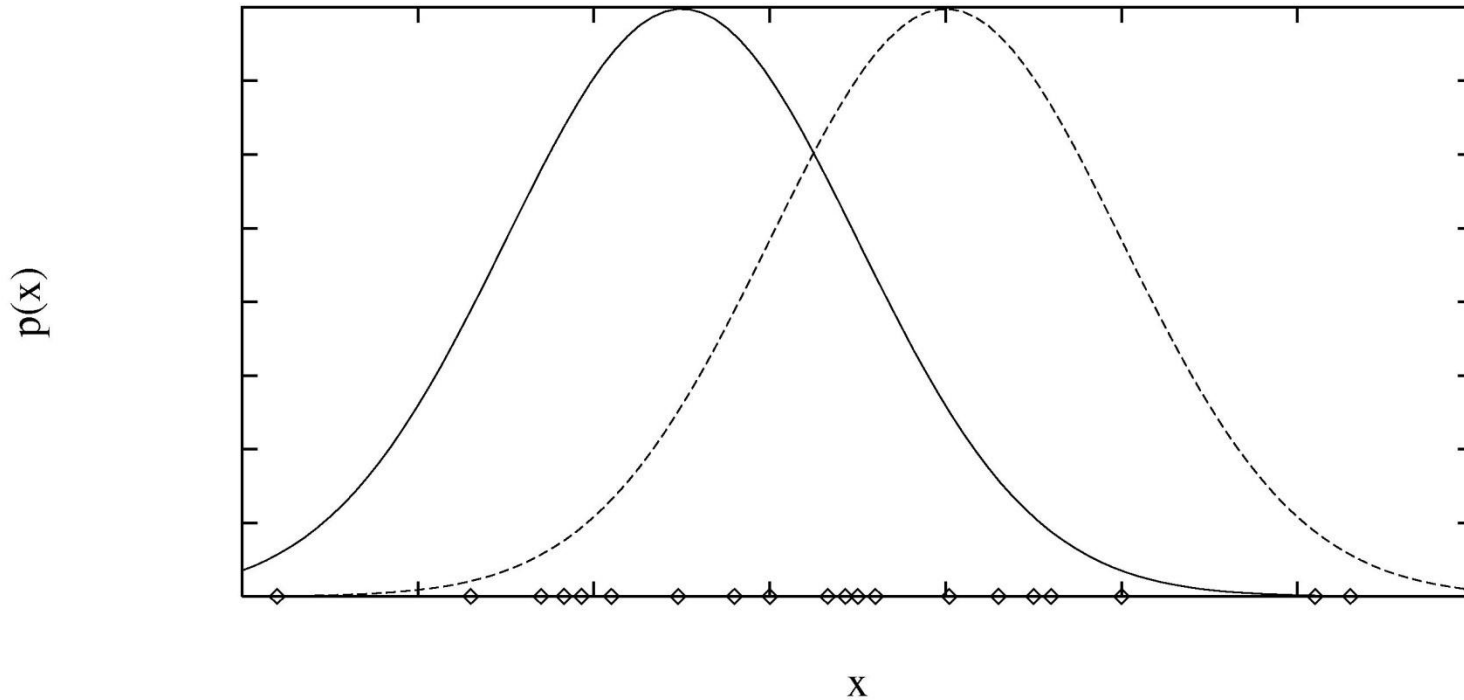
**Naive Bayes:**  $P(x|c_i) = \prod_{j=1}^{n_d} P(x_j|c_i)$

**AutoClass:** Naive Bayes with various  $x_j$  models

**Mixture of Gaussians:**  $P(x|c_i) =$  Multivariate Gaussian

**In general:**  $P(x|c_i)$  can be any distribution

# Mixtures of Gaussians



$$P(x|\mu_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{1}{2} \left( \frac{x - \mu_i}{\sigma} \right)^2 \right]$$

# The EM Algorithm

Initialize parameters ignoring missing information

Repeat until convergence:

**E step:** Compute expected values of unobserved variables, assuming current parameter values

**M step:** Compute new parameter values to maximize probability of data (observed & estimated)

(Also: Initialize expected values ignoring missing info)

# EM for Mixtures of Gaussians

**Initialization:** Choose means at random, etc.

**E step:** For all examples  $x_k$ :

$$P(\mu_i | x_k) = \frac{P(\mu_i)P(x_k | \mu_i)}{P(x_k)} = \frac{P(\mu_i)P(x_k | \mu_i)}{\sum_{i'} P(\mu_{i'})P(x_k | \mu_{i'})}$$

**M step:** For all components  $c_i$ :

$$\begin{aligned} P(c_i) &= \frac{1}{n_e} \sum_{k=1}^{n_e} P(\mu_i | x_k) \\ \mu_i &= \frac{\sum_{k=1}^{n_e} x_k P(\mu_i | x_k)}{\sum_{k=1}^{n_e} P(\mu_i | x_k)} \\ \sigma_i^2 &= \frac{\sum_{k=1}^{n_e} (x_k - \mu_i)^2 P(\mu_i | x_k)}{\sum_{k=1}^{n_e} P(\mu_i | x_k)} \end{aligned}$$

## Mixtures of Gaussians (cont.)

- K-means clustering  $\prec$  EM for mixtures of Gaussians
- Mixtures of Gaussians  $\prec$  Bayes nets
- Also good for estimating joint distribution of continuous variables

# Mixture Models for Documents

---

- Learn simultaneously  $P(w \mid \text{topic})$ ,  $P(\text{topic} \mid \text{doc})$

“Arts”

“Budgets”

“Children”

“Education”

NEW

MILLION

CHILDREN

SCHOOL

FILM

TAX

WOMEN

STUDENTS

SHOW

PROGRAM

PEOPLE

SCHOOLS

MUSIC

BUDGET

CHILD

EDUCATION

MOVIE

BILLION

YEARS

TEACHERS

PLAY

FEDERAL

FAMILIES

HIGH

MUSICAL

YEAR

WORK

PUBLIC

BEST

SPENDING

PARENTS

TEACHER

ACTOR

NEW

SAYS

BENNETT

FIRST

STATE

FAMILY

MANIGAT

YORK

PLAN

WELFARE

NAMPHY

OPERA

MONEY

MEN

STATE

THEATER

PROGRAMS

PERCENT

PRESIDENT

ACTRESS

GOVERNMENT

CARE

ELEMENTARY

LOVE

CONGRESS

LIFE

HAITI

# Kinds of Clustering

---

- Sequential
  - Fast
  - Results depend on data order
- Cost Optimization
  - Fixed number of clusters (typically)
  - Probabilistic models
- Hierarchical
  - Start with many clusters
  - join clusters at each step

# Greedy Hierarchical Clustering

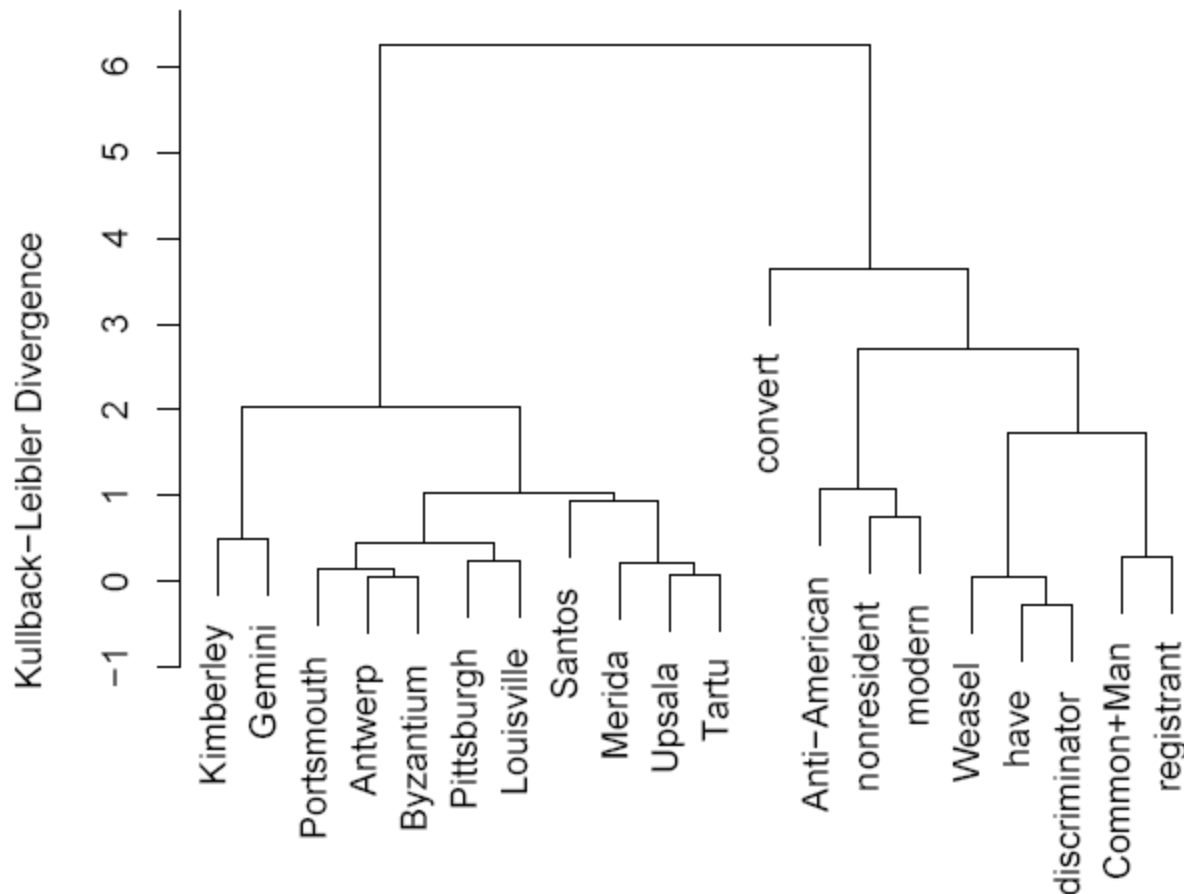
---

- Initialize one cluster for each data point
- Until *done*
  - Merge the two *nearest* clusters

# Hierarchical Clustering on Strings

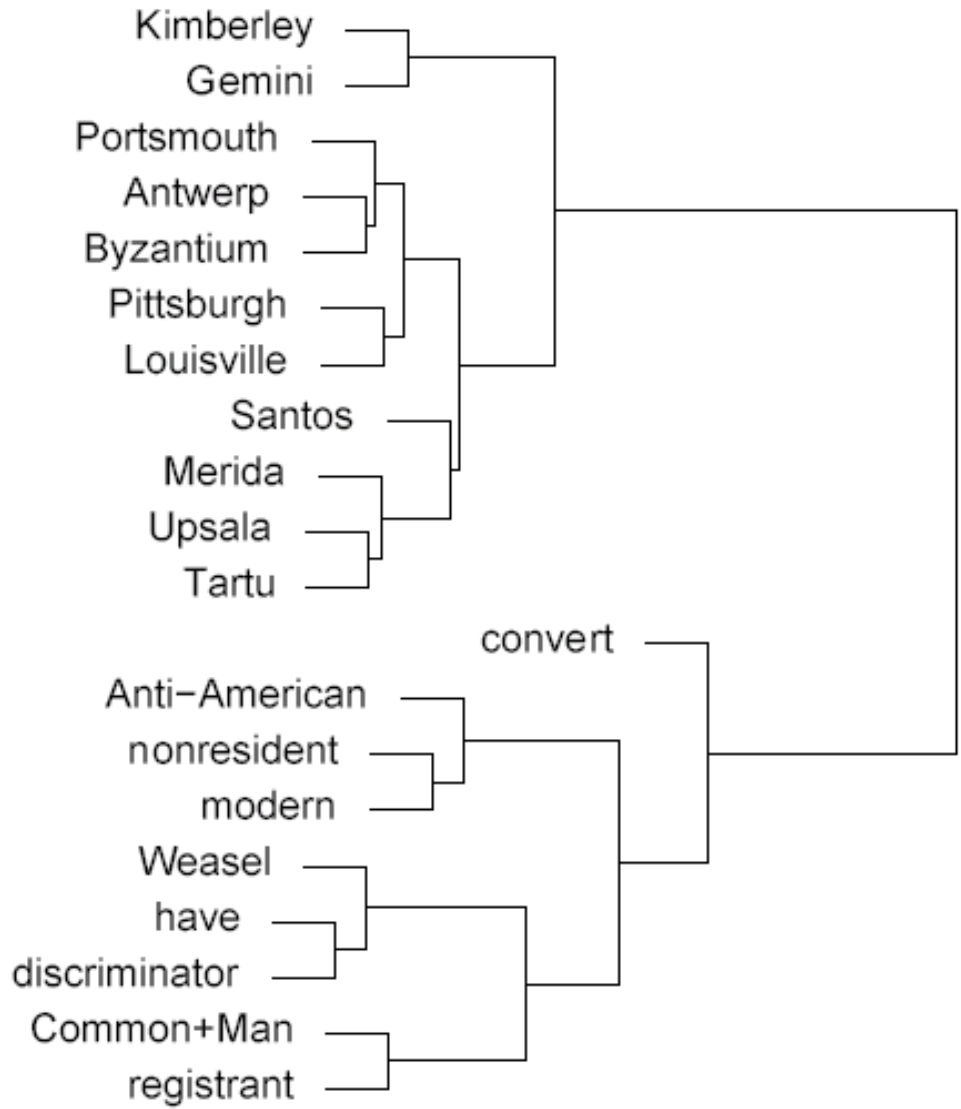
- Features = *contexts* in which strings appear

10 cities and 10 people



Kullback-Leibler Divergence

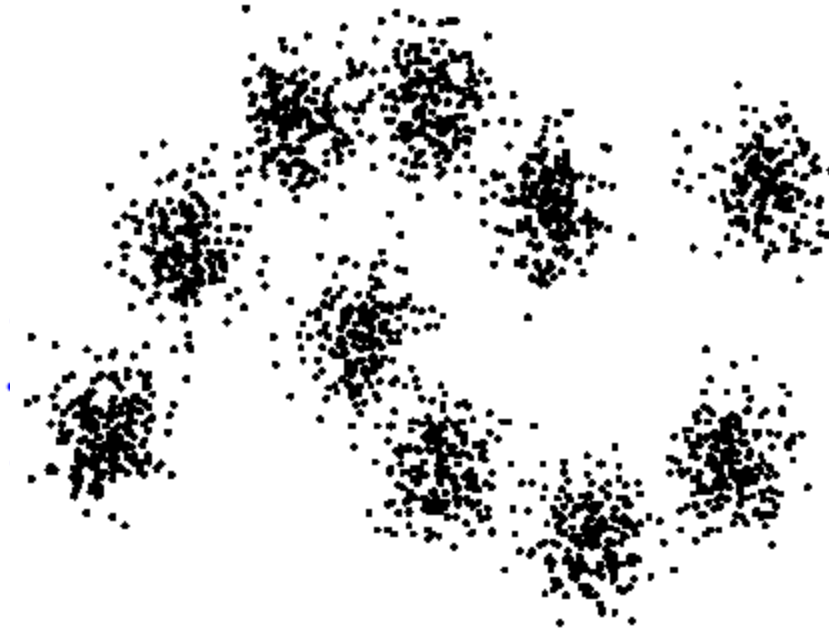
-1 0 1 2 3 4 5 6



10 cities and 10 people

# Classic Example: Half Moons

---



From Batra et al., <http://www.cs.cmu.edu/~rahuls/pub/bmvc2008-clustering-rahuls.pdf>

# Summary

---

- Algorithms:
  - Sequential clustering
    - Requires key distance threshold, sensitive to data order
  - K-means clustering
    - Requires # of clusters, sensitive to initial conditions
    - Special case of mixture modeling
  - Greedy agglomerative clustering
    - Naively takes  $O(n^2)$  runtime
    - Hard to tell when you're "done"