CS57300
PURDUE UNIVERSITY

NOVEMBER 15, 2021

DATA MINING

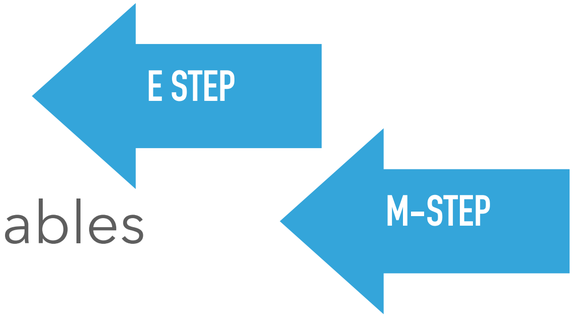
ANNOUCENMENT

- Midterm grades are out!
- Assignment 5 is out!
  - Due date: November 28, 11:59pm

HOW TO LEARN GMMS?

EXPECTATION-MAXIMIZATION (EM) ALGORITHM

- Popular algorithm for parameter estimation in data with hidden/unobserved values
  - Hidden variables=cluster membership
- Basic idea
  - Initialize parameters
  - Predict values for hidden variables given current parameters
  - Estimate parameters given current prediction for hidden variables
  - Repeat



EM FOR GMM

- Suppose we make a guess for the parameters values
- Use these to evaluate posterior probs for cluster memberships (using Bayes rule)

$$\Gamma(x_n) = [\gamma_1(x_n), ..., \gamma_K(x_n)]$$

E-STEP

- Now compute the log-likelihood using predicted cluster memberships

$$log\ p(x, z|\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_i(x_n) \left[log\ w_k + log\ N(x_n|\mu_k, \Sigma_k) \right]$$

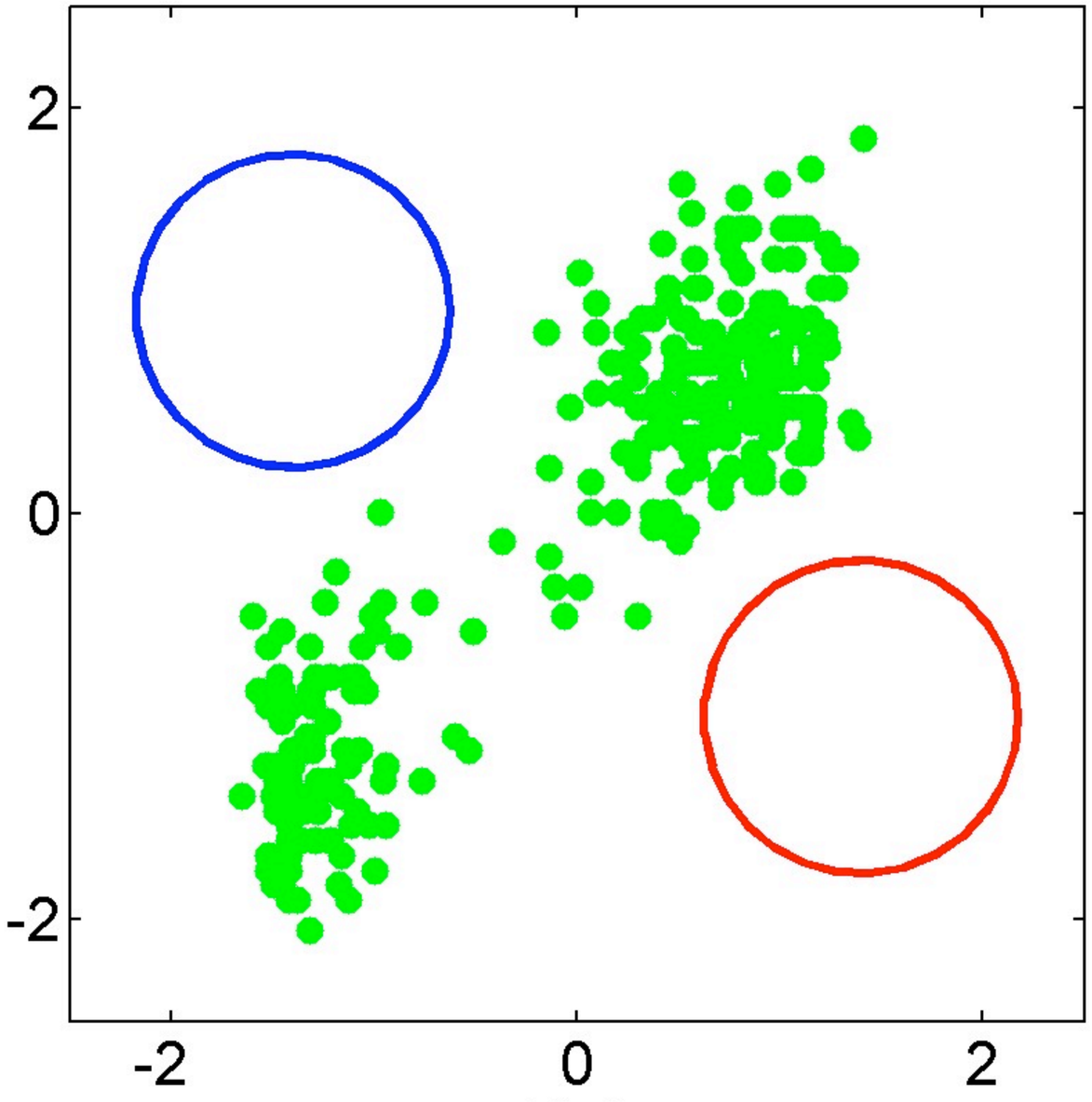
- Use expected complete likelihood to determine MLE for parameters (w_k,μ_k,Σ_k)

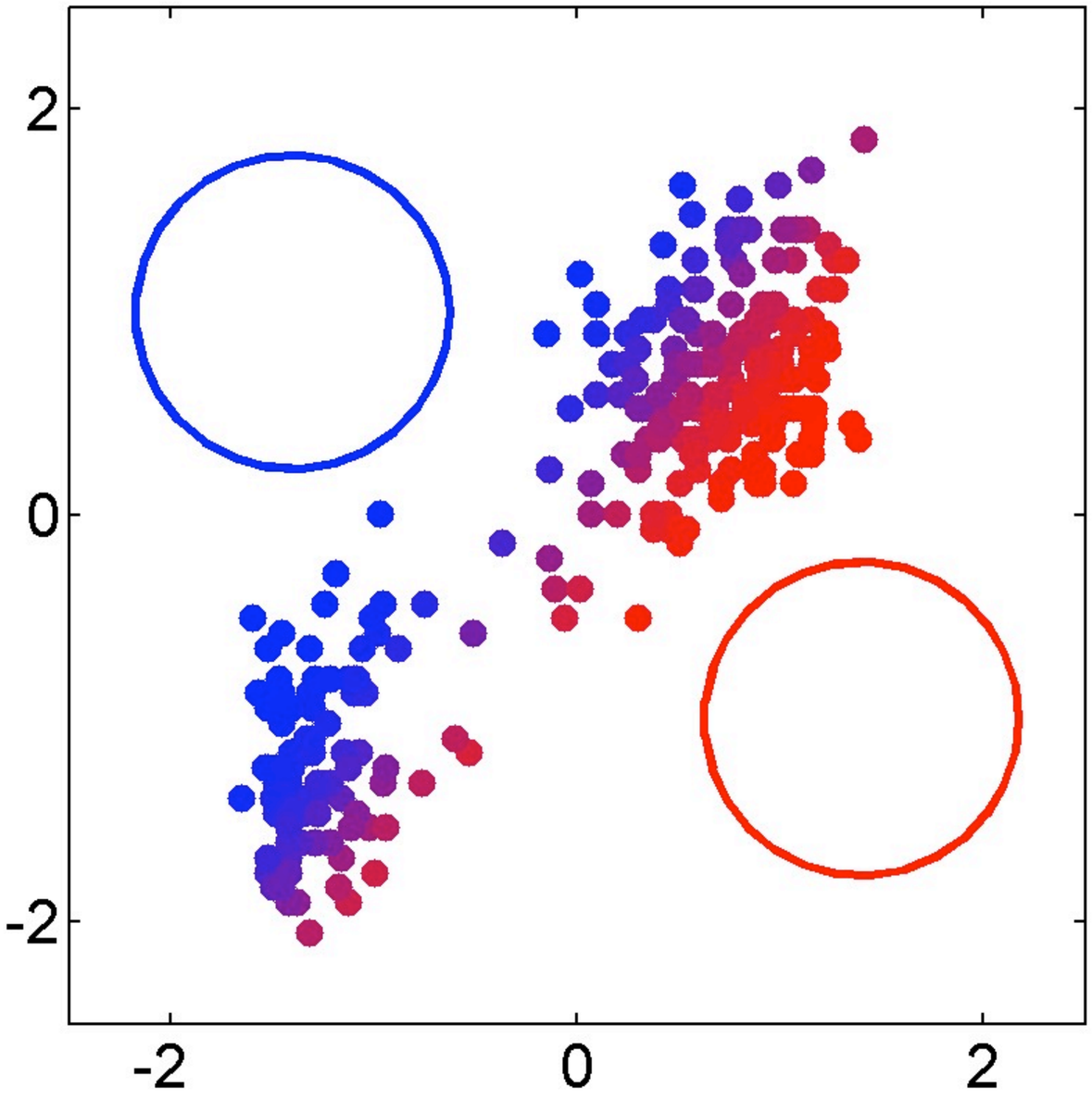
M-STEP

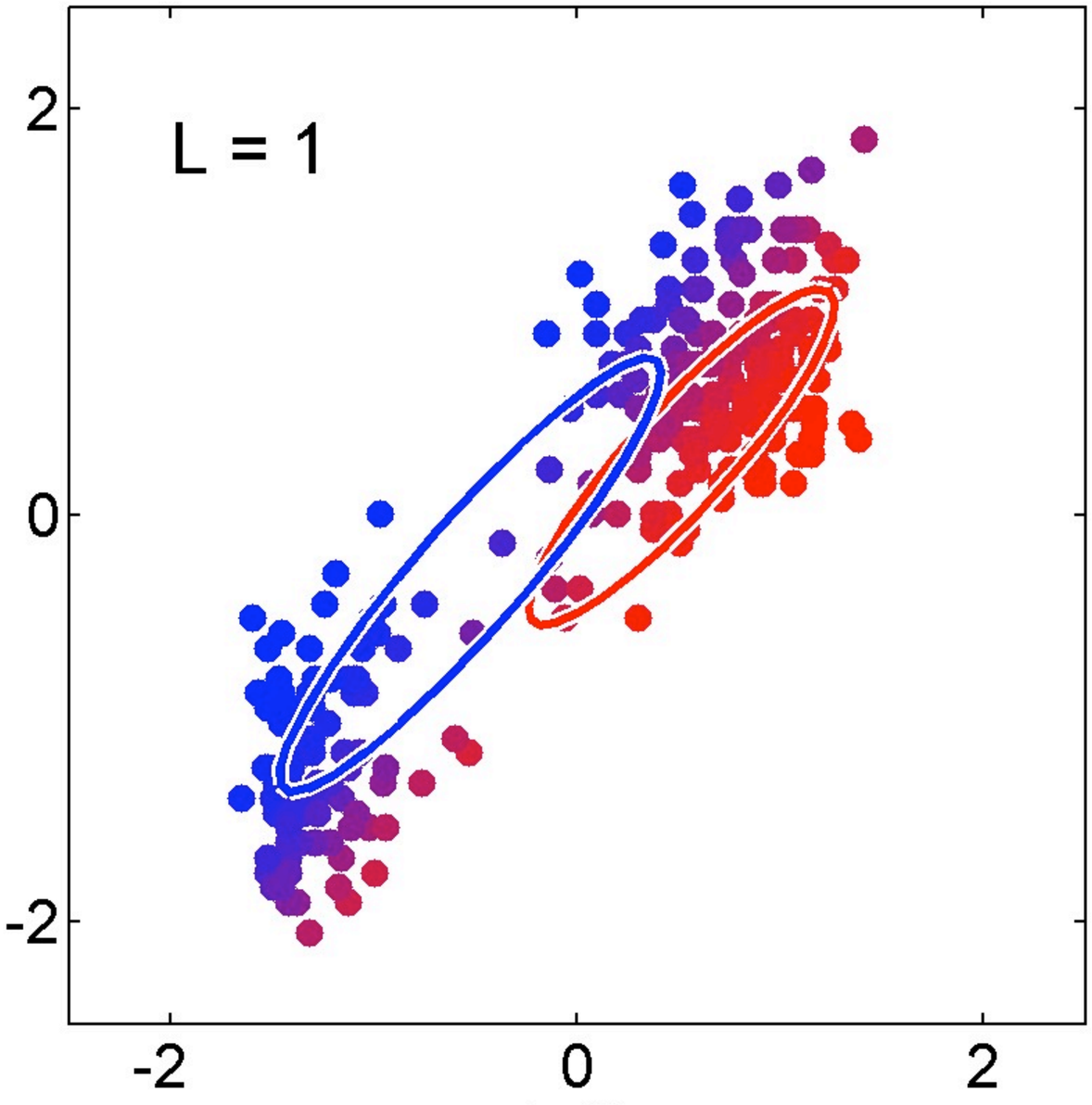
MORE ON EM

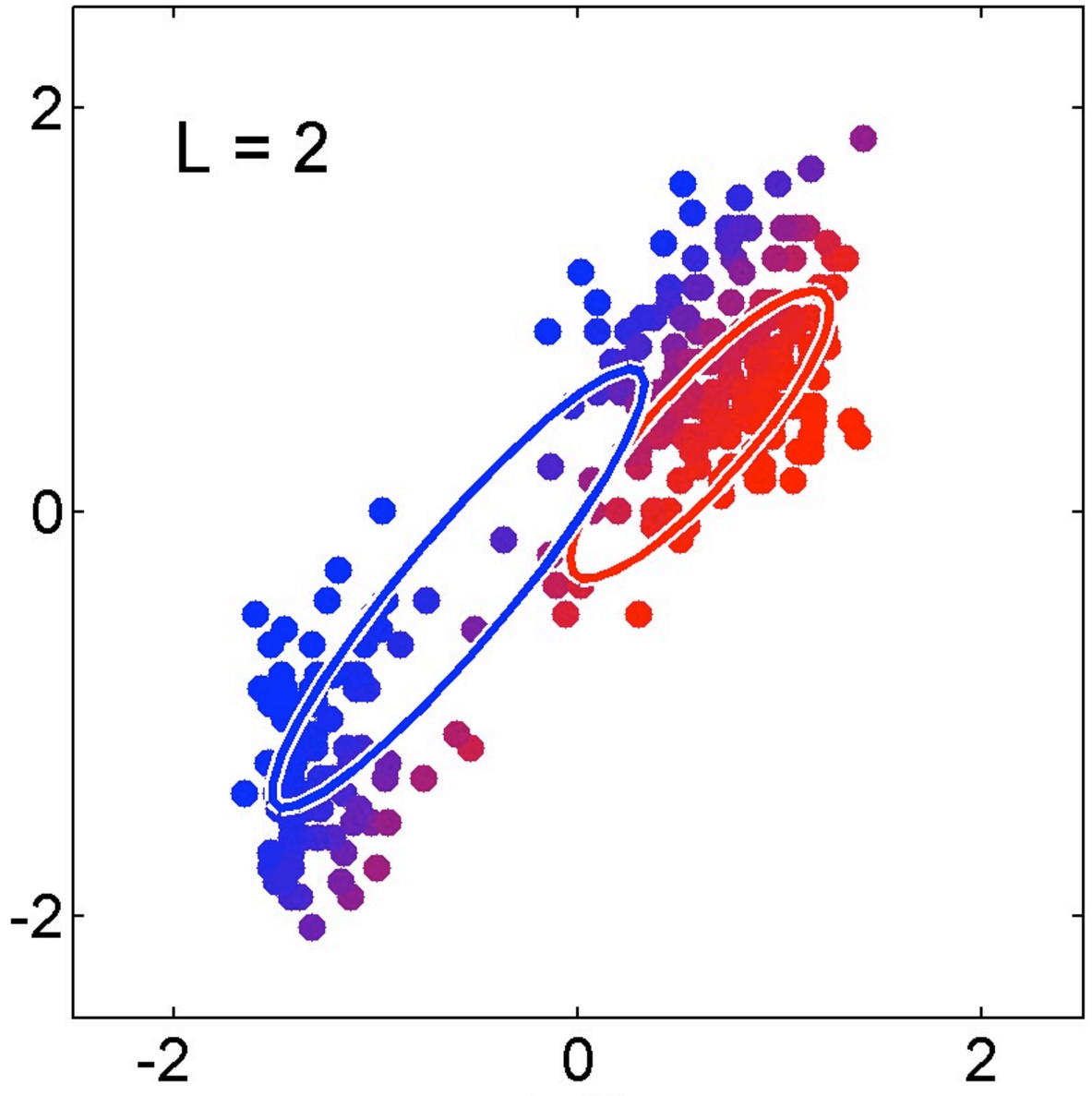
- Often both the E and the M step can be solved in closed form
- Neither the E step nor the M step can decrease the log-likelihood
- Algorithm is guaranteed to converge to a local maximum of the likelihood
- Must specify initialization and stopping criteria

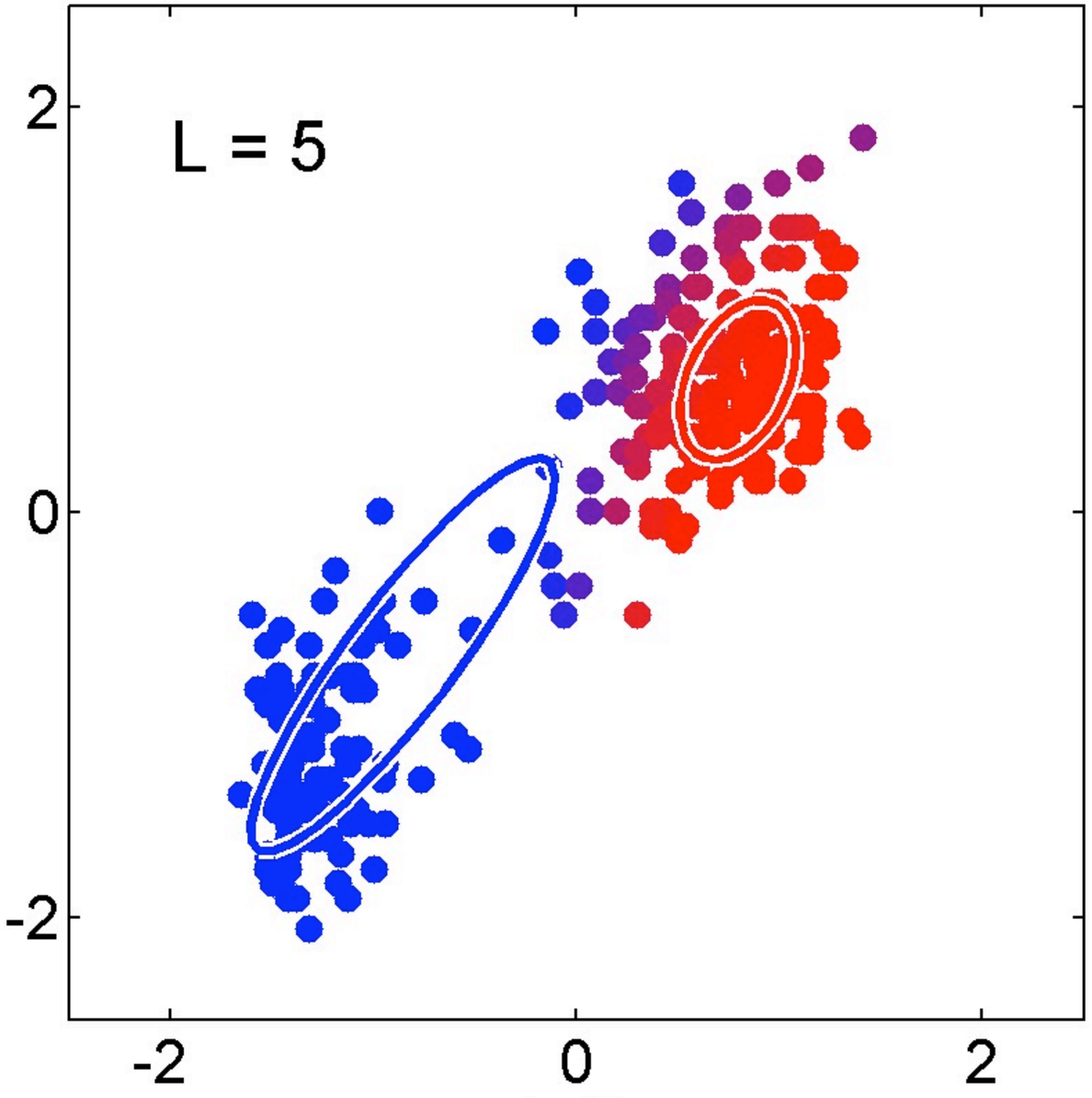
GMM EXAMPLE

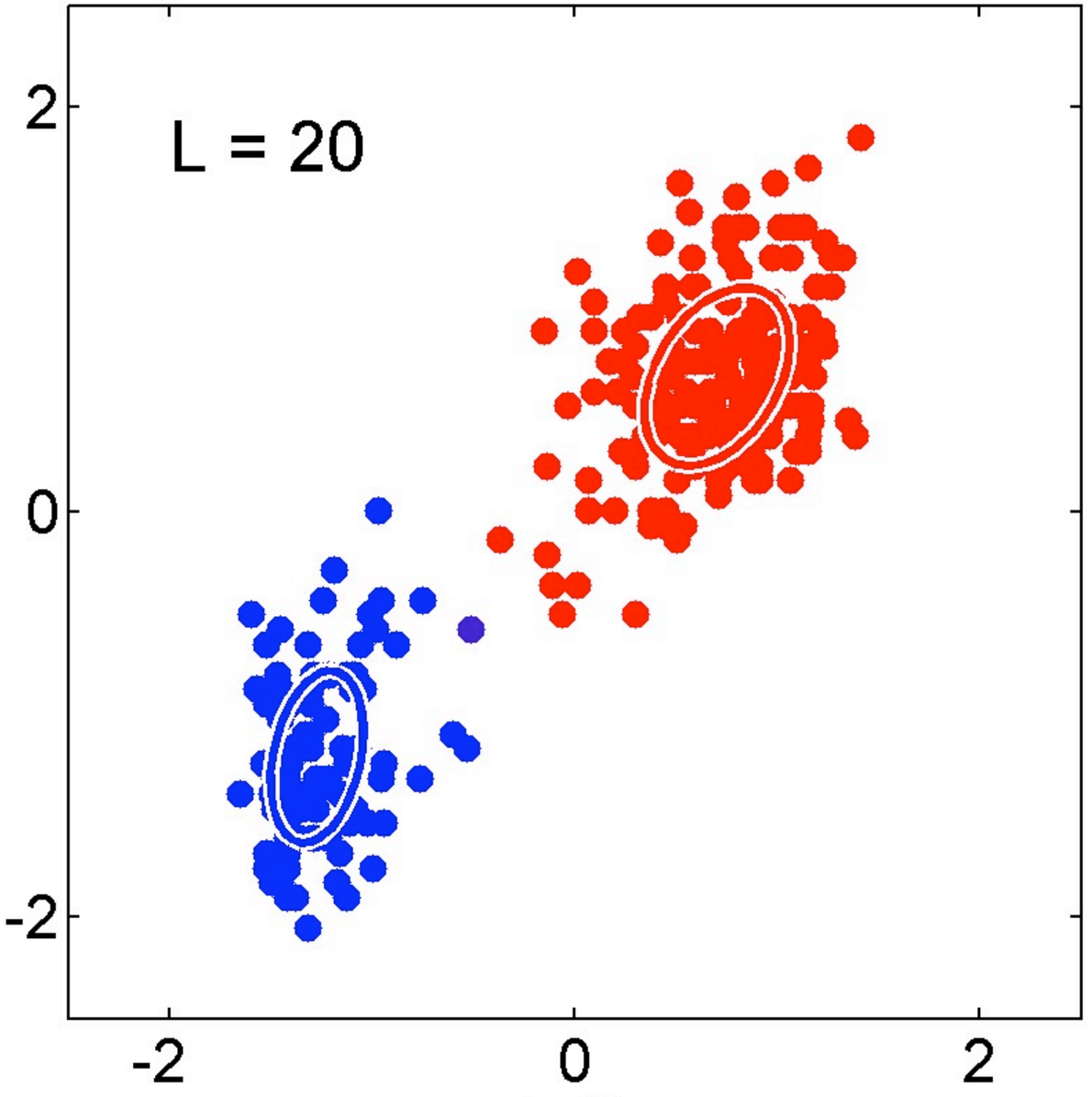

2
0
-2
-2
0
2


2
0
-2
-2
0
2


L = 1
2
0
-2
-2
0
2


L = 2
2
0
-2
-2
0
2


L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

PROBABILISTIC CLUSTERING

- Model provides full distributional description for each component
  - May be able to interpret differences in the distributions
- Soft clustering (compared to k-mean hard clustering)
  - Given the model, each point has a k-component vector of membership probabilities
- Key cost: assumption of parametric model

MIXTURE MODELS

- Knowledge representation?
  - **Parametric model**
    parameters = mixture coefficient and component parameters
- Score function?
  - **Likelihood**
- Search?
  - **Expectation maximization**
    iteratively find parameters that maximize likelihood and predicts cluster memberships
- Optimal? Exhaustive?

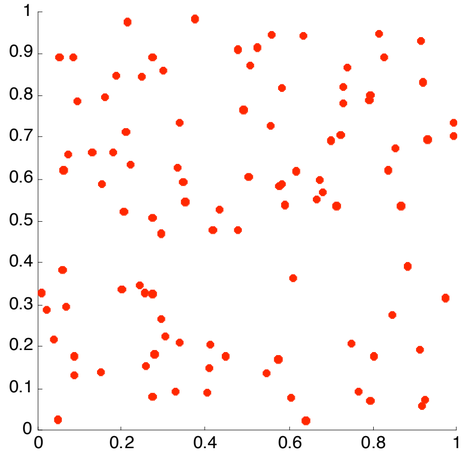
DESCRIPTIVE MODELING: EVALUATION

CLUSTER VALIDITY

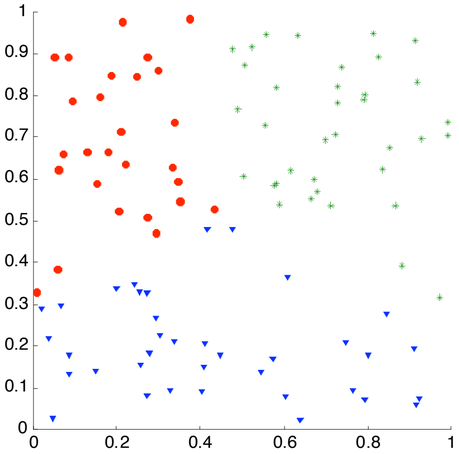
- For prediction tasks there are a variety of external evaluation metrics
  - Accuracy, precision, recall, area under ROC, etc.
- For cluster analysis the external evaluation should evaluate the “goodness” of the resulting clusters
- Why do we want external validation?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters

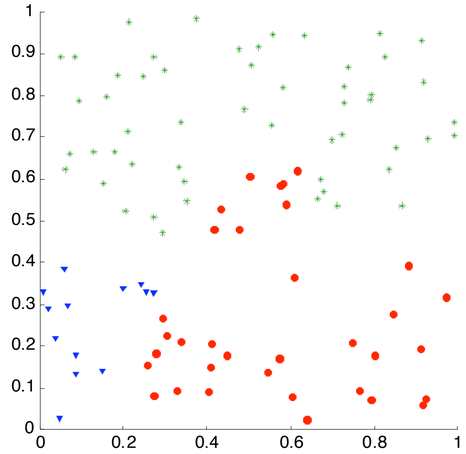
RANDOM DATA: CLUSTERING STILL RETURNS RESULTS

Random Points



K-means





Complete Link

EVALUATION APPROACHES

- Determine the clustering tendency of the data
- Evaluate the quality of clustering results
  - Evaluate the clusters using known class labels
  - Evaluate how well the clusters “fit” the data
  - Determine which of two different clustering results is better
  - Determine the “correct” number of clusters

CLUSTERING TENDENCY

- Evaluate whether a dataset has clusters before clustering
- Most common approach (for low-dimensional Euclidean data)
  - Use a statistical test for spatial randomness
  - Hopkins statistic: sample *p* points from dataset, generate *p* random points in same space

$$H = \frac{\sum_{i=1}^{p} u_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

u_i: distance from random point to NN in data

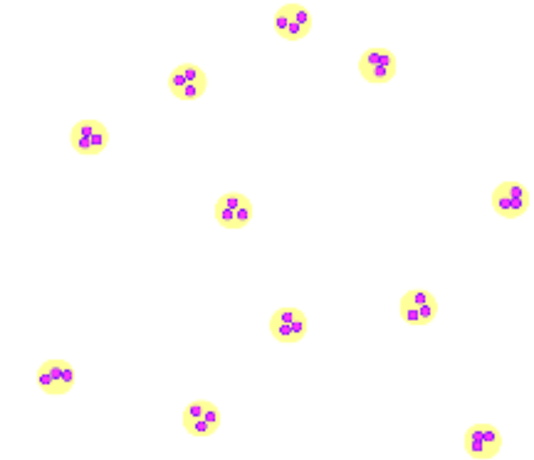
w_i: distance from sample point to NN in data

HOPKINS STATISTIC

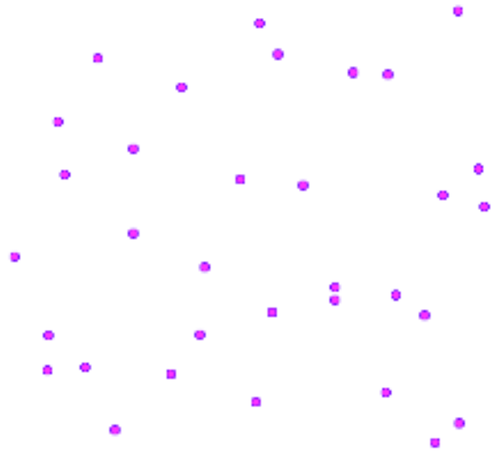
$$H = \frac{\sum_{i=1}^{p} u_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

u_i: distance from random point to NN in data

w_i: distance from sample point to NN in data



H close to 1, clustered!



H close to 0.5, random data!

TYPES OF CLUSTERING EVALUATION MEASURES

- **Supervised**
  - Measures the extent to which clusters match external class label values
- **Unsupervised**
  - Measures goodness of fit without class labels

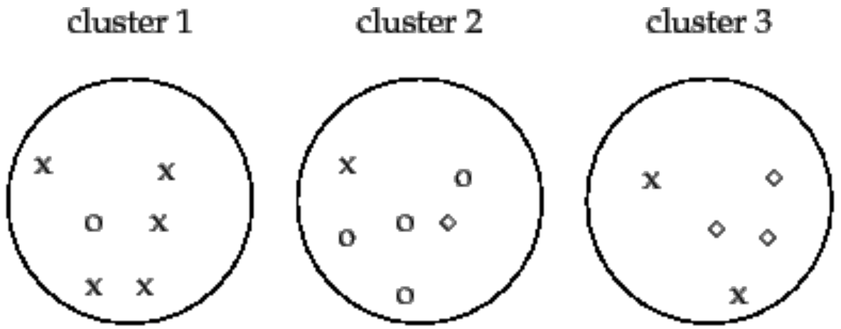
SUPERVISED: EVALUATING CLUSTER QUALITY WITH LABELS

- If you have class labels why cluster?
  - Usually labels come from small hand-labeled dataset for evaluation
  - But have remaining large dataset to cluster automatically
  - May want to assess how close clusterings correspond to classes but still allow for more variation in the clusters

CLASSIFICATION-ORIENTED

- **Purity**: a measure of the degree to which a cluster/group (G_i) contains objects of one particular class (C_j)
  - A cluster G_i will be labeled as the majority class among all objects in G_i

$$purity(C, G) = \frac{1}{N} \sum_{i=1}^{K} max_j |x \in G_i, x \in C_j|$$



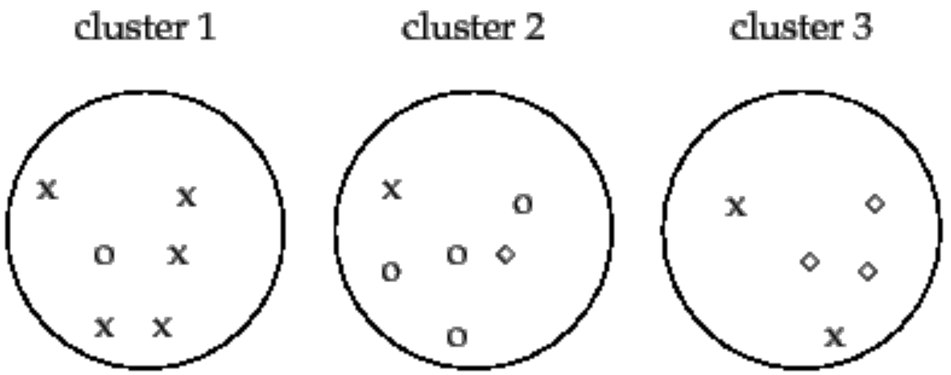
  - High purity is easy to achieve when the number of clusters is large

CLASSIFICATION-ORIENTED

- **Entropy**: the degree to which each cluster (G) consists of objects of a single class (C)
  - For each cluster G_i compute the probability of class *j* (within the cluster)

$$entropy(C, G) = \sum_{i=1}^{K} - \sum_{j=1}^{C} p_{ij} log(p_{ij})$$

How does score for this clustering change?



CLASSIFICATION-ORIENTED

- **Normalized mutual information gain:**
  - Measures the amount of information by which our knowledge about the classes (C) increases when the clusters (G) are identified

$$NMI(C,G) = \frac{I(C,G)}{H(C)+H(G)}$$

$$= \frac{\sum_c \sum_g p(c,g) log \frac{p(c,g)}{p(c)p(g)}.}{-\sum_c p(c) log\, p(c) - \sum_g p(g) log\, p(g)}$$

  - NMI score is between 0 (min) and 1 (max).
  - Denominator (normalization) adjusts for problem that entropy tends to increase with the number of clusters