Introduction to Machine Learning: CS 436/580L

Clustering

Instructor: Arti Ramesh

Binghamton University

Machine Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Parametric

Non-parametric

Y Continuous

Y Discrete

Gaussians

Learned in closed form

Linear Functions

1. Learned in closed form
2. Using gradient descent

Decision Trees

Greedy search; pruning

Probability of class | features

1. Learn P(Y), P(X|Y); apply Bayes
2. Learn P(Y|X) w/ gradient descent

Non-probabilistic

Linear: perceptron gradient descent

Nonlinear: neural net: backprop

Support vector machines

2

Administrivia

- Quiz next Tuesday, Dec 5th, 1:15 pm
- HW 5 will be released later today

Outline

- K-means & Agglomerative Clustering
- Agglomerative Clustering
- Expectation Maximization (EM)

4

Overview of Learning

Type of Supervision

(eg, Experience, Feedback)

What is Being Learned?

	Labeled Examples	Reward	Nothing
Discrete Function	Classification		Clustering
Continuous Function	Regression		
Policy	Apprenticeship Learning	Reinforcement Learning	

5

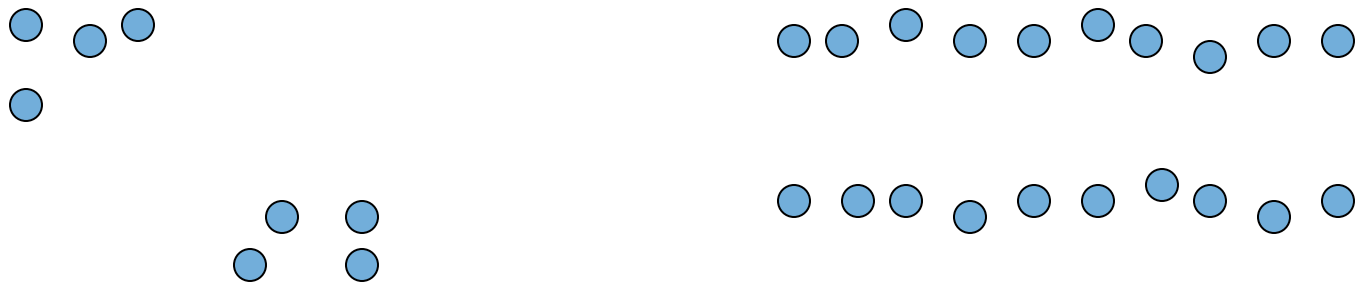
Clustering

Clustering systems:

- Unsupervised learning
- Requires data, but no labels
- Detect patterns e.g. in
  - Group emails or search results
  - Customer shopping patterns
  - Program executions (intrusion detection)
- Useful when don't know what you're looking for
- But: often get gibberish

Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



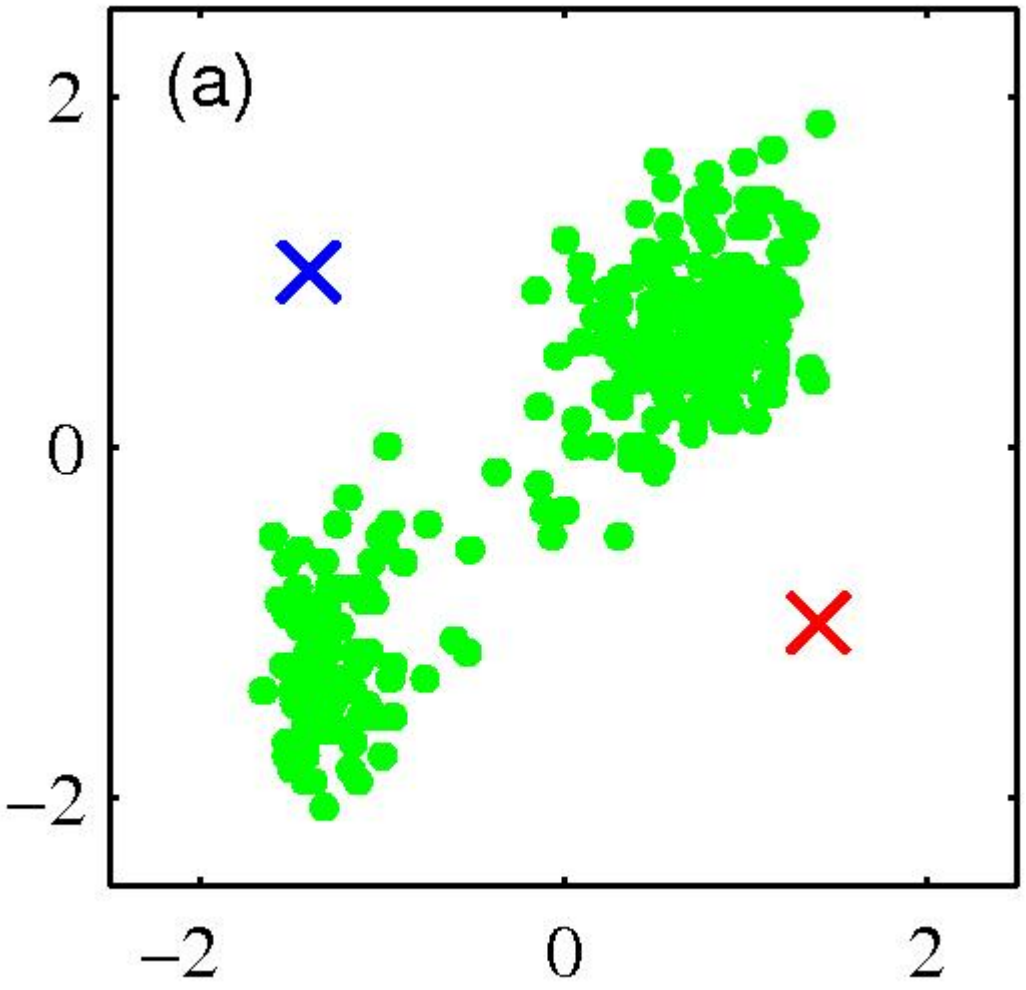
- What could "similar" mean?
  - One option: small (squared) Euclidean distance

$$\text{dist}(x, y) = (x - y)^{\mathsf{T}}(x - y) = \sum_i (x_i - y_i)^2$$

K-Means: Algorithm

- An iterative clustering algorithm
  - Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest cluster center
    - Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change

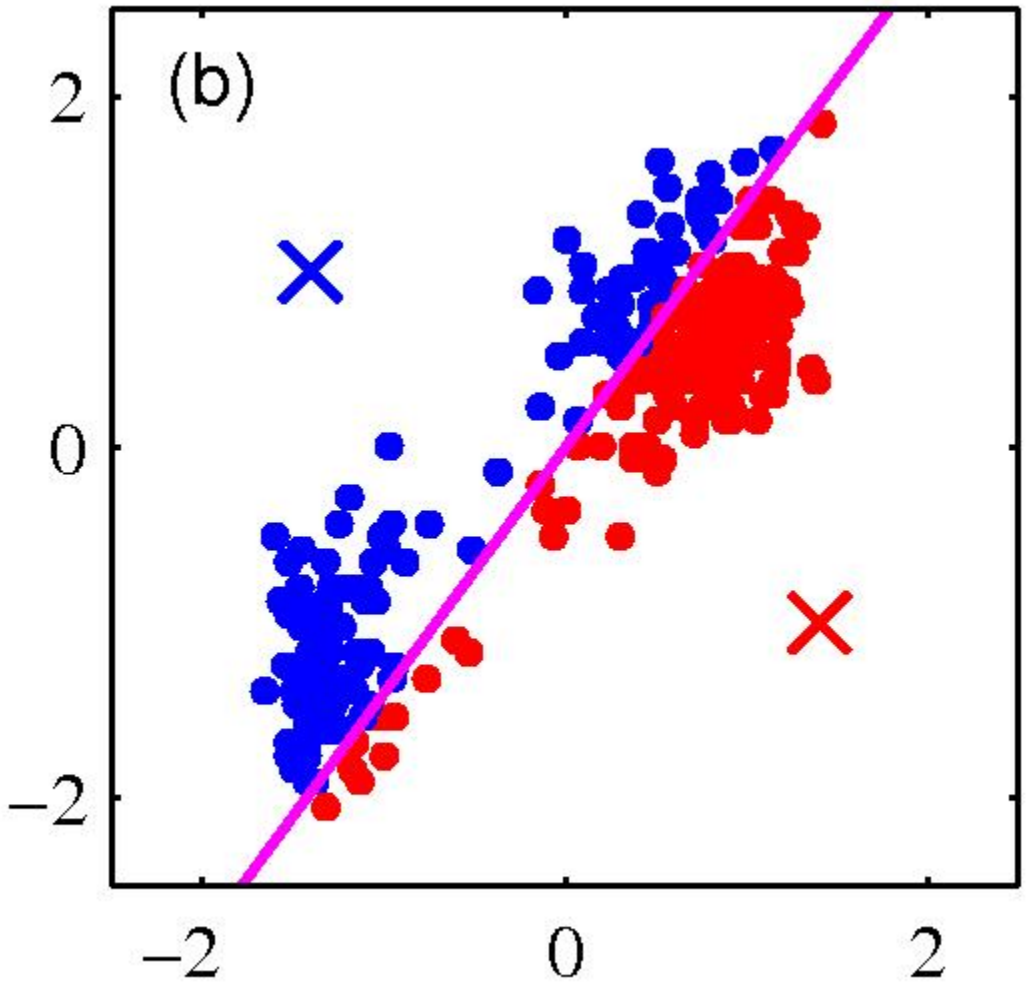
K-means clustering: Example



- Pick K random points as cluster centers (means)

9

K-means clustering: Example

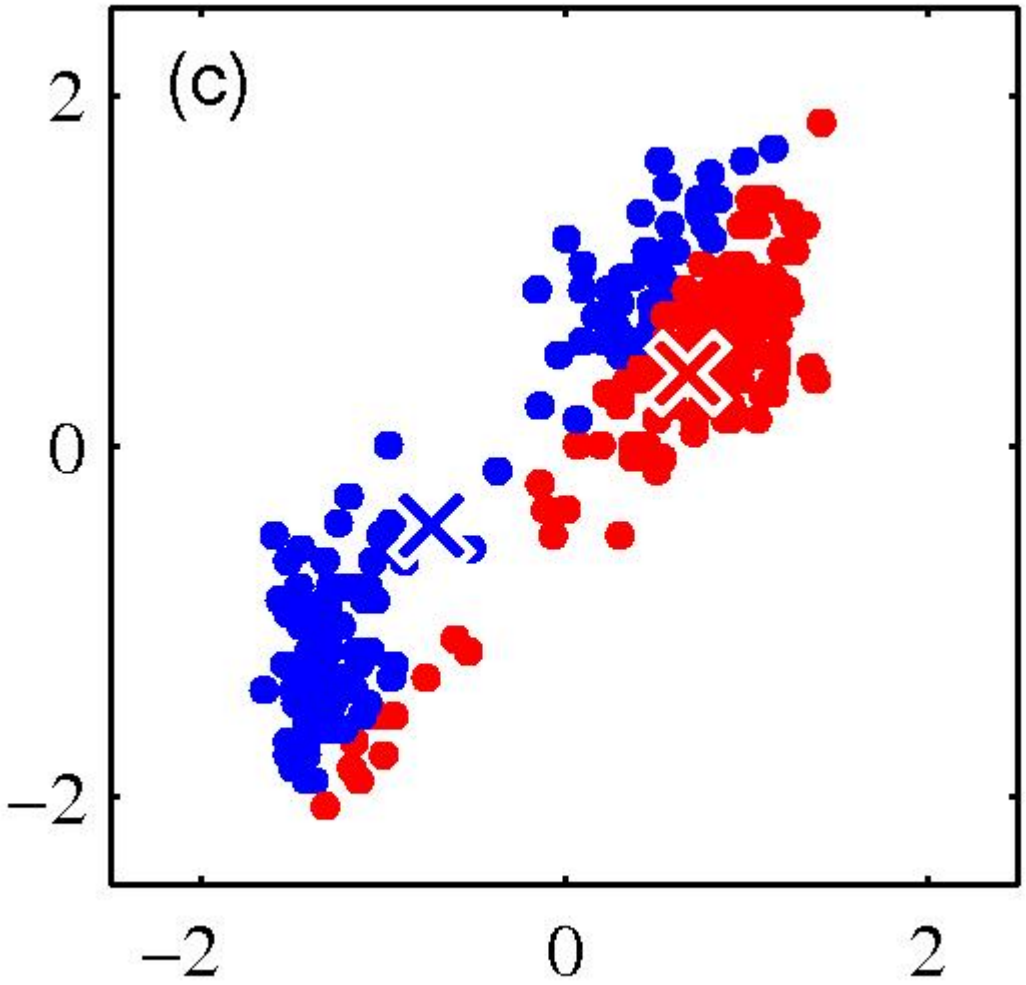


Iterative Step 1

- Assign data instances to closest cluster center

10

K-means clustering: Example

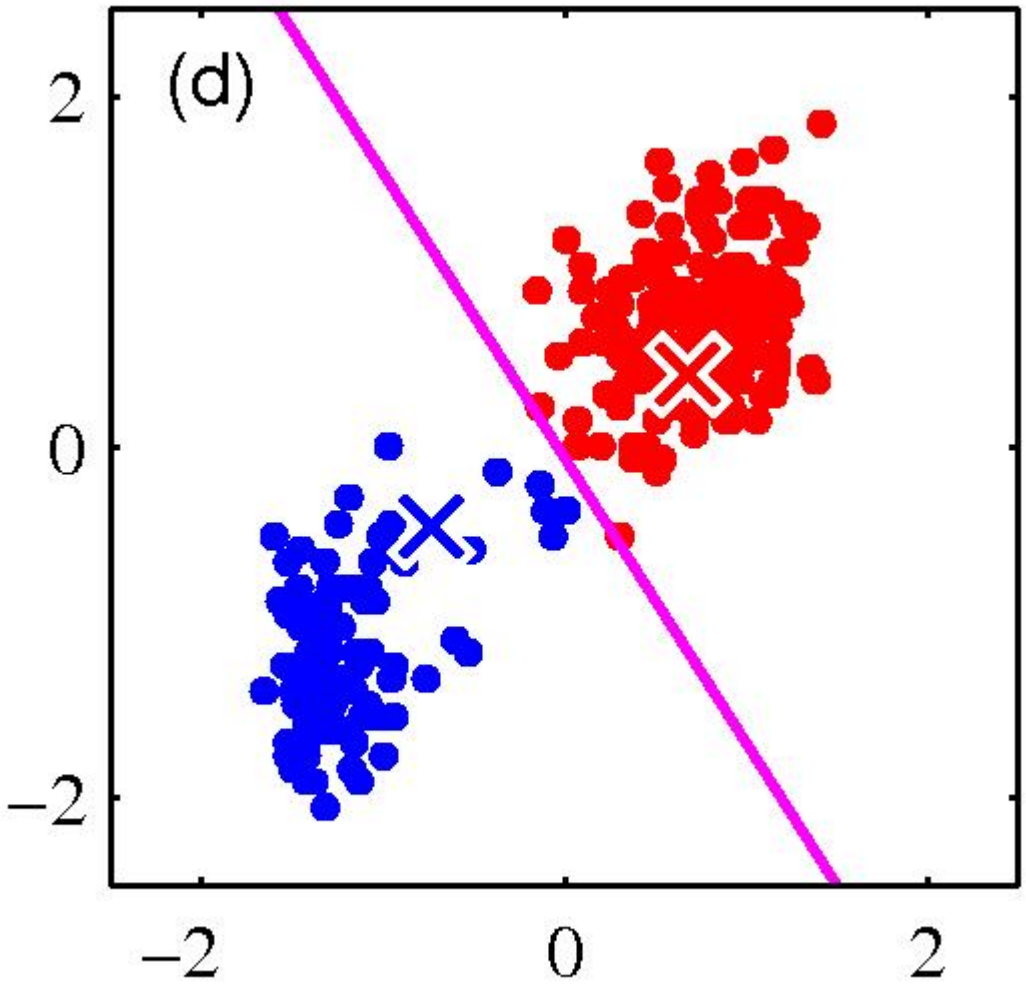


Iterative Step 2

- Change the cluster center to the average of the assigned points

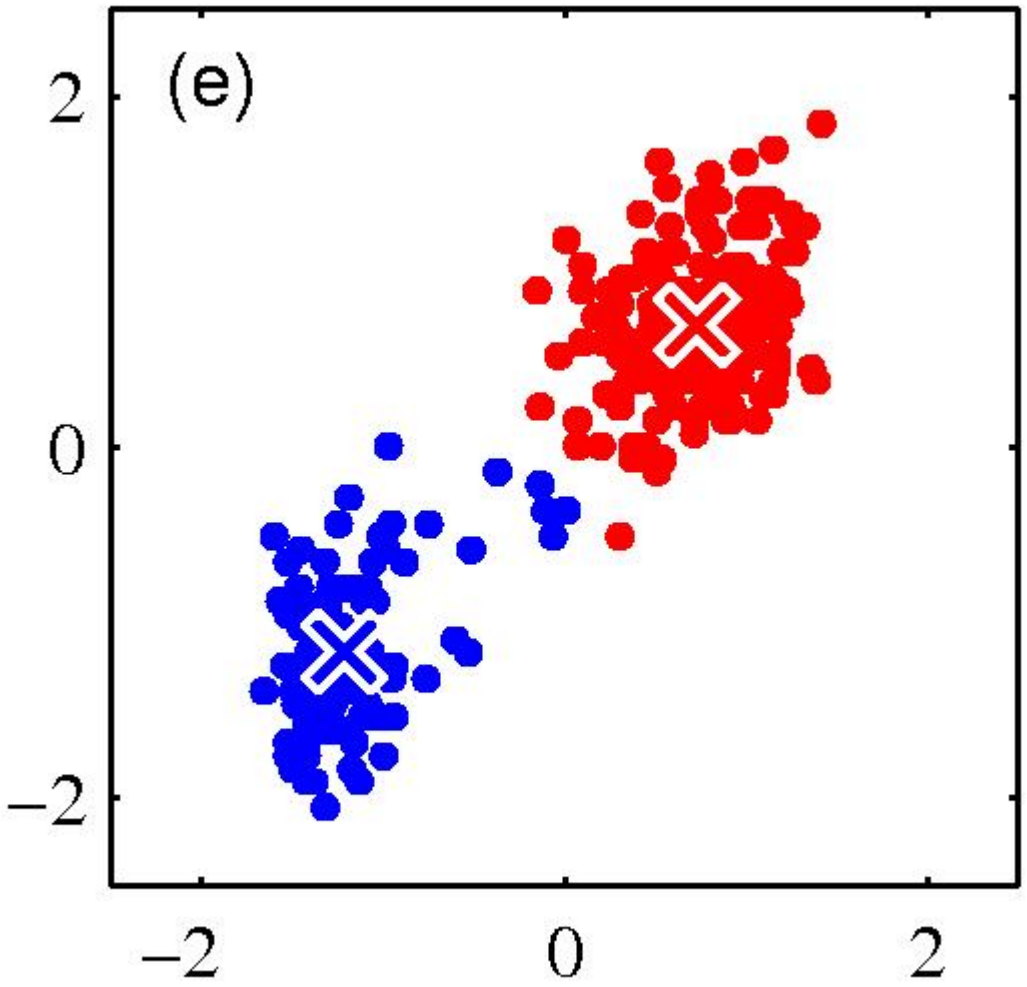
11

K-means clustering: Example



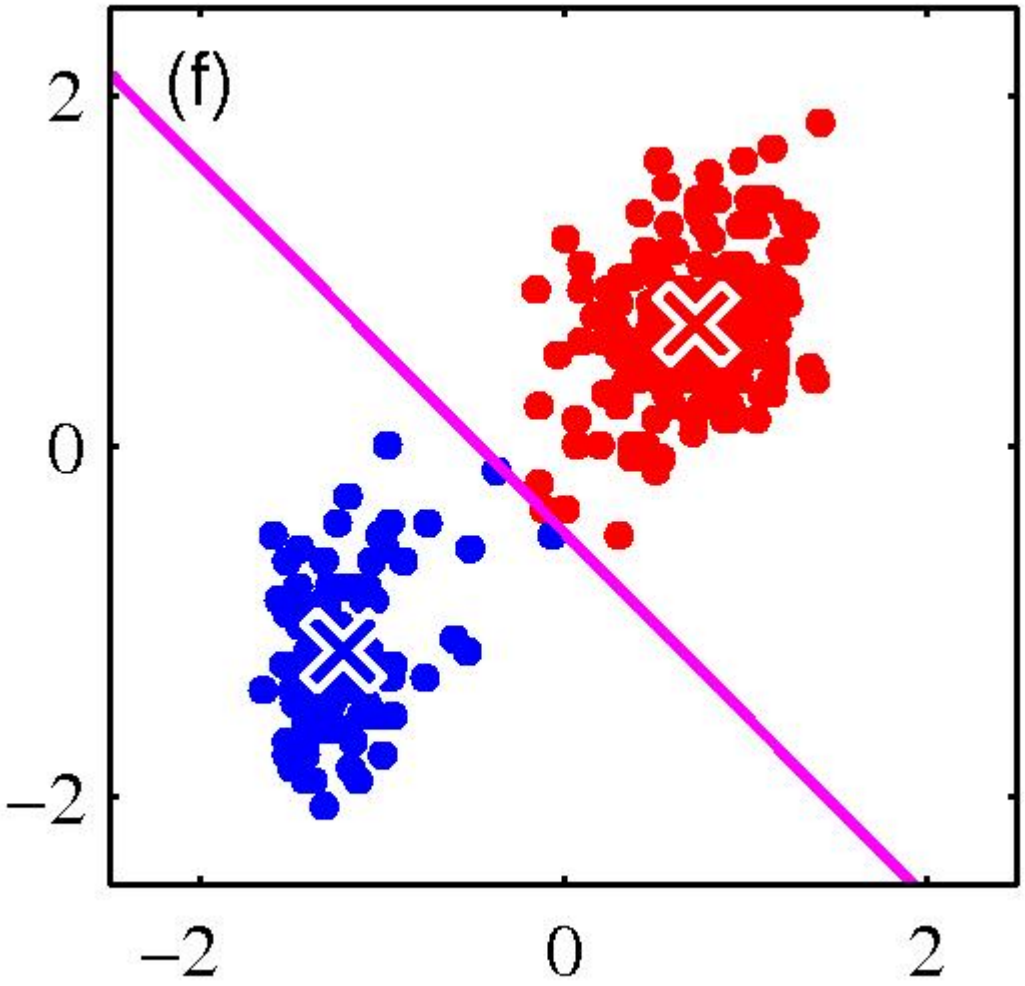
12

K-means clustering: Example



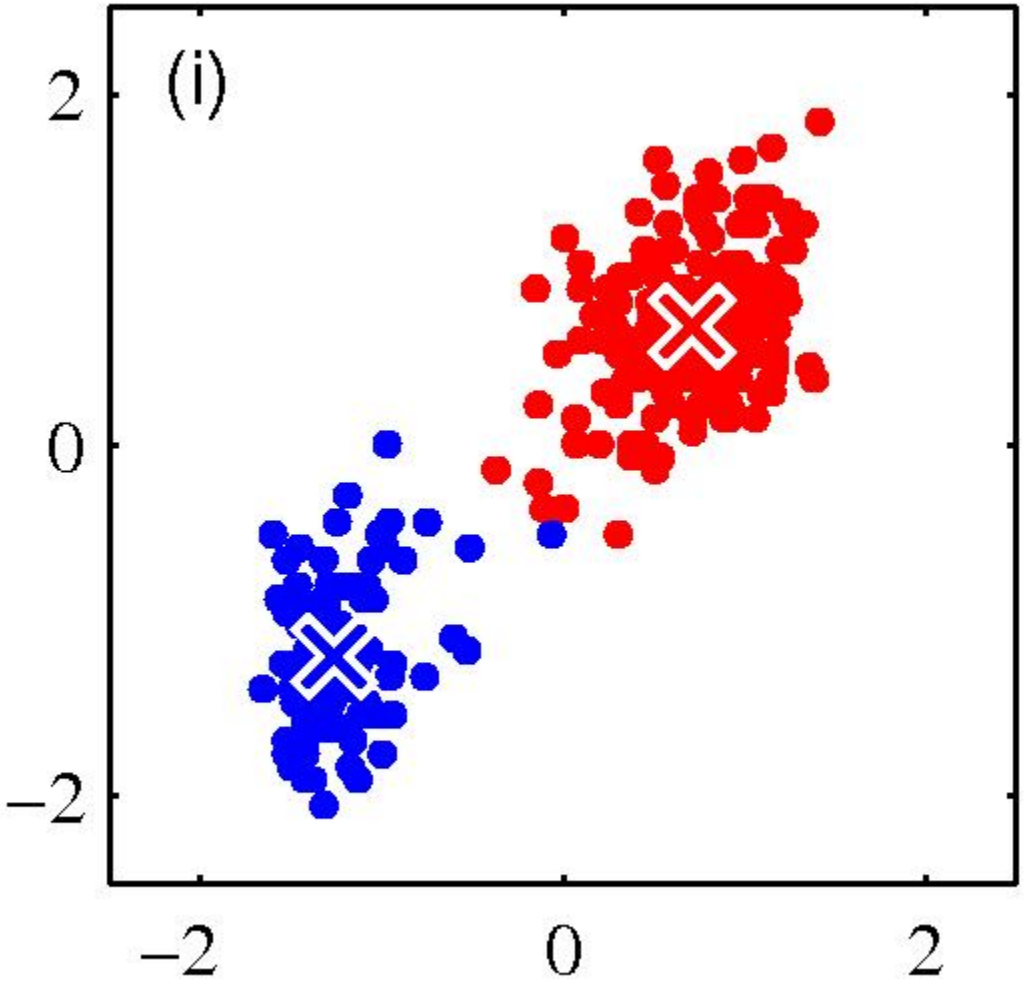
13

K-means clustering: Example



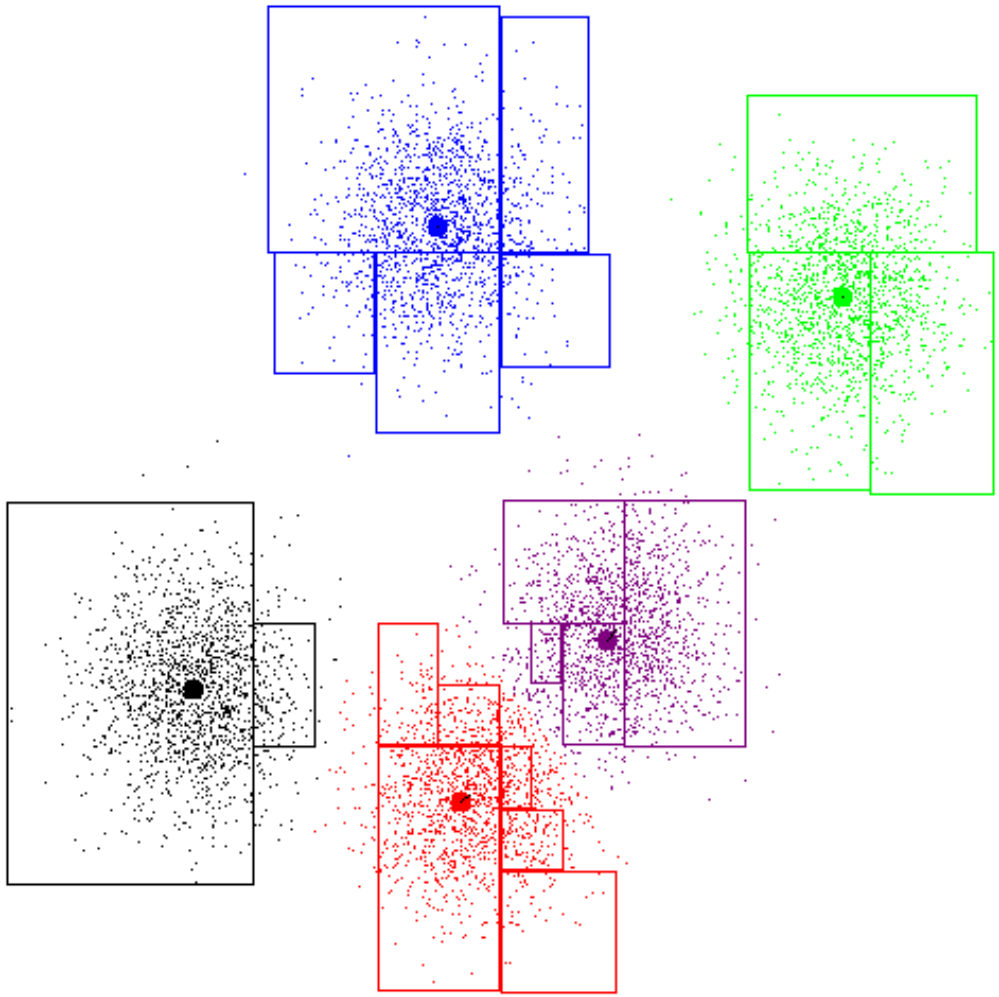
14

K-means clustering: Example



15

K-Means Example



Example: K-Means for Segmentation

K=2

Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.

Original

Example: K-Means for Segmentation

K=2

K=3

Original

Example: K-Means for Segmentation

K=2

K=3

K=10

Original

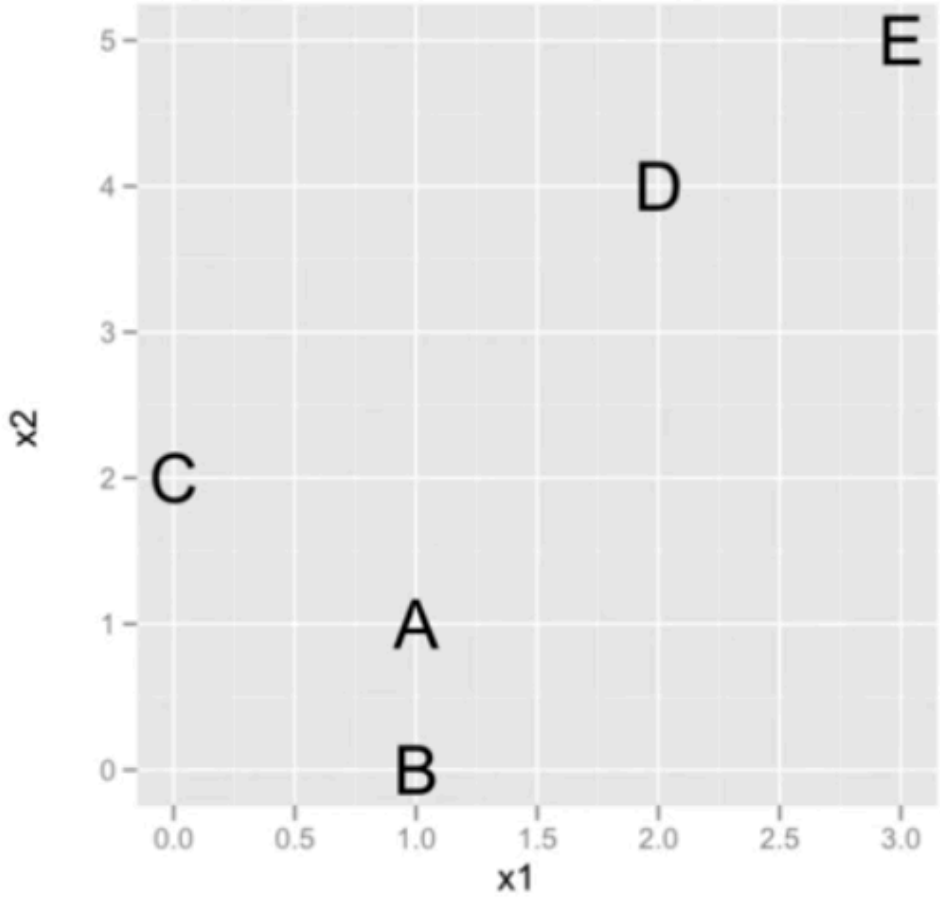
4%

8%

17%

K-Means Example

i	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5



Number of clusters: 2; A and C are initial cluster centers

K-Means Example

$\bar{X}_1^0$ ●

$\bar{X}_2^0$ ●

i	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

i	1	2
A	0	1.4
B	1	2.2
C	1.4	0
D	3.2	2.8
E	4.5	4.2

K-Means Example

i	1	2	Cluster
A	0	1.4	1
B	1	2.2	1
C	1.4	0	2
D	3.2	2.8	2
E	4.5	4.2	2

i	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

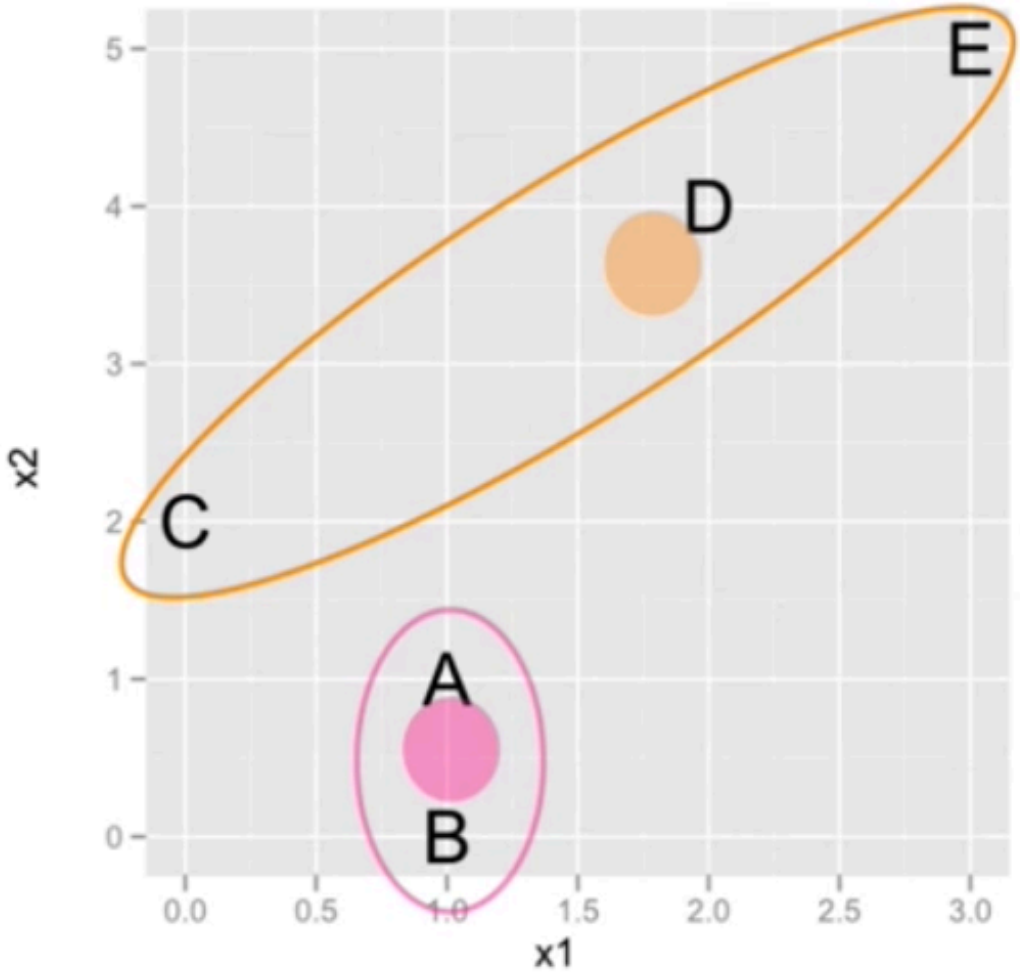
$\bar{X}_1^1$

$\bar{X}_2^1$

$\bar{X}_1^1 = (1, 0.5)$

$\bar{X}_2^1 = (1.7, 3.7)$

K-Means Example



$\bar{X}_1^1 = (1, 0.5)$

$\bar{X}_2^1 = (1.7, 3.7)$

K-Means Example

i	X_1	X_2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

$\bar{X}_1^1 = (1, 0.5)$

$\bar{X}_2^1 = (1.7, 3.7)$

i	1	2
A	0.5	2.7
B	0.5	3.7
C	1.8	2.4
D	3.6	0.5
E	4.9	1.9

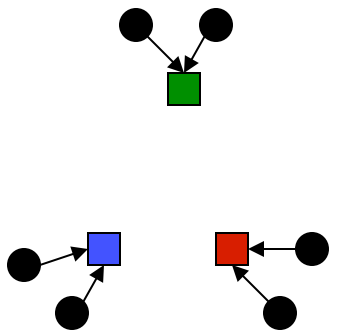
K-Means as Optimization

- Consider the total distance to the means:

$$\phi(\{x_i\}, \{a_i\}, \{c_k\}) = \sum_i \text{dist}(x_i, c_{a_i})$$

points assignments means

- Two stages each iteration:
  - Update assignments: fix means c, change assignments a
  - Update means: fix assignments a, change means c

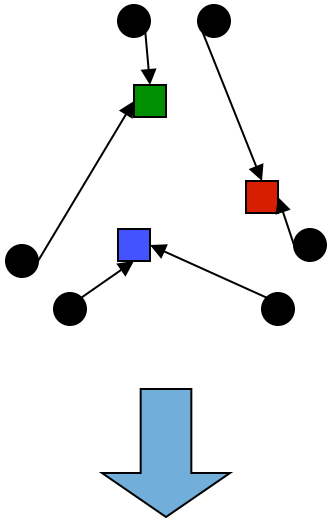


- Co-ordinate Descent
- Will it converge?
  - Yes!, if you can argue that each update can't increase Φ

Phase I: Update Assignments

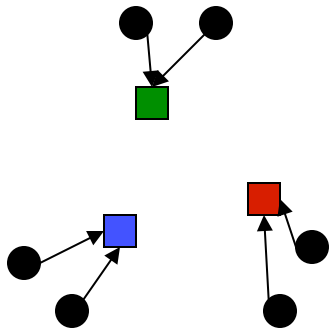
- For each point, re-assign to closest mean:

$$a_i = \underset{k}{\operatorname{argmin}} \operatorname{dist}(x_i, c_k)$$



- Can only decrease total distance phi!

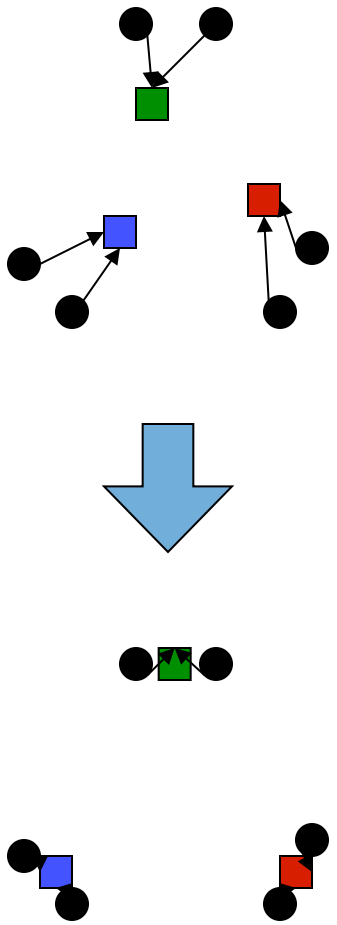
$$\phi(\{x_i\}, \{a_i\}, \{c_k\}) = \sum_i \operatorname{dist}(x_i, c_{a_i})$$



Phase II: Update Means

- Move each mean to the average of its assigned points:

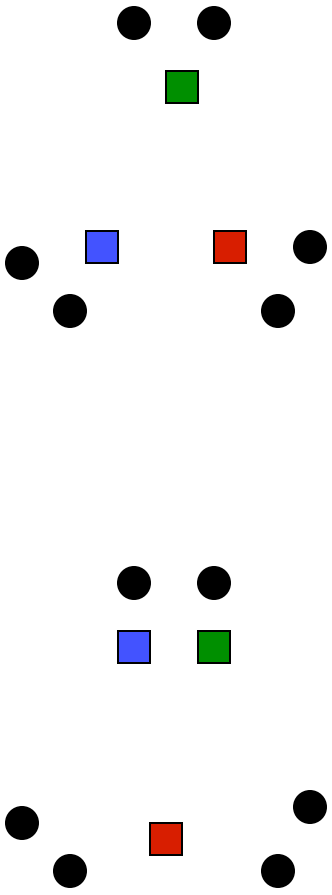
$$c_k = \frac{1}{|\{i : a_i = k\}|} \sum_{i:a_i=k} x_i$$



- Also can only decrease total distance… (Why?)
- Fun fact: the point y with minimum squared Euclidean distance to a set of points {x} is their mean

Initialization

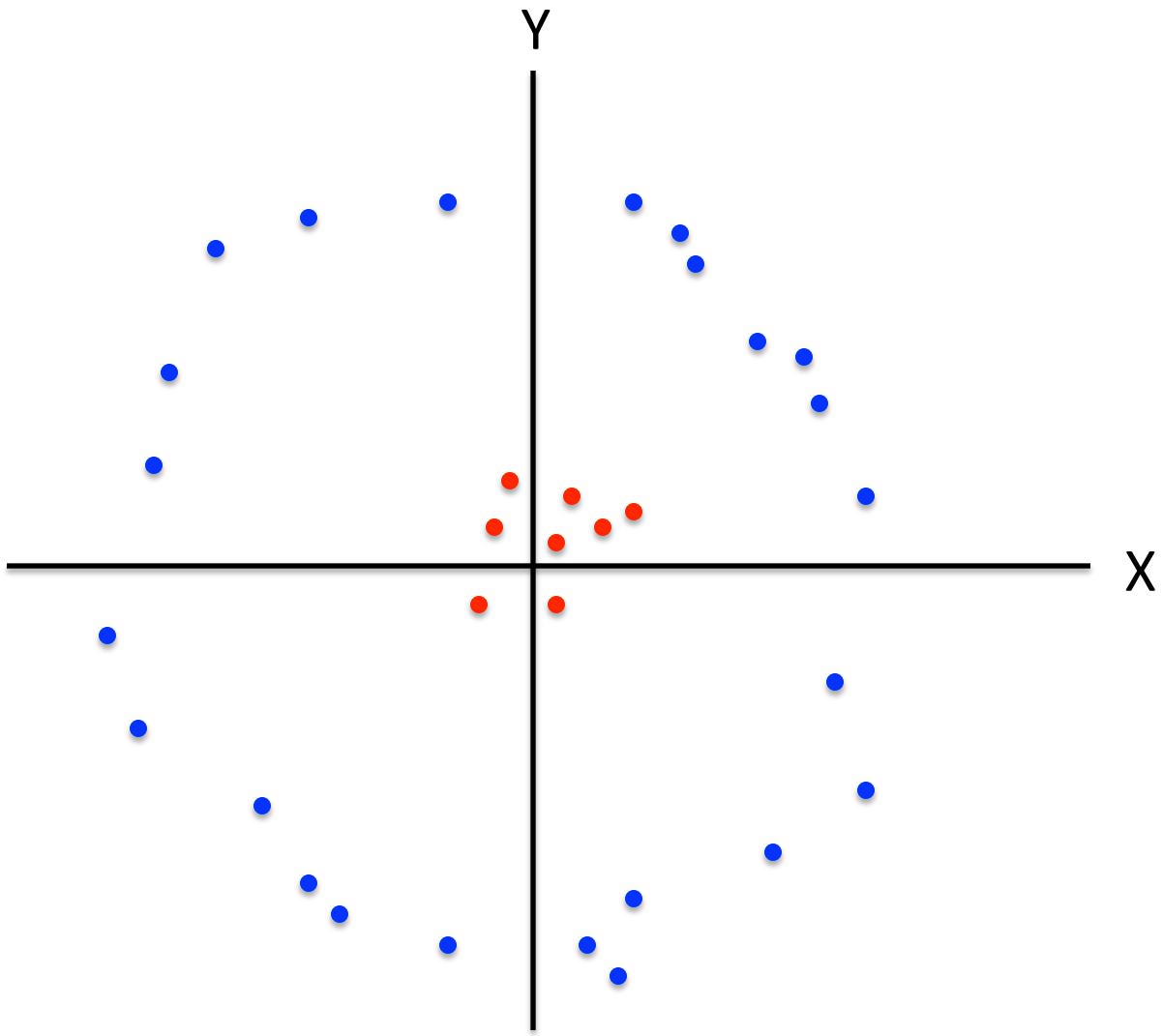
- K-means is non-deterministic
  - Requires initial means
  - It does matter what you pick!
  - What can go wrong?
  - Various schemes for preventing this kind of thing: variance-based split / merge, initialization heuristics



K-Means Getting Stuck

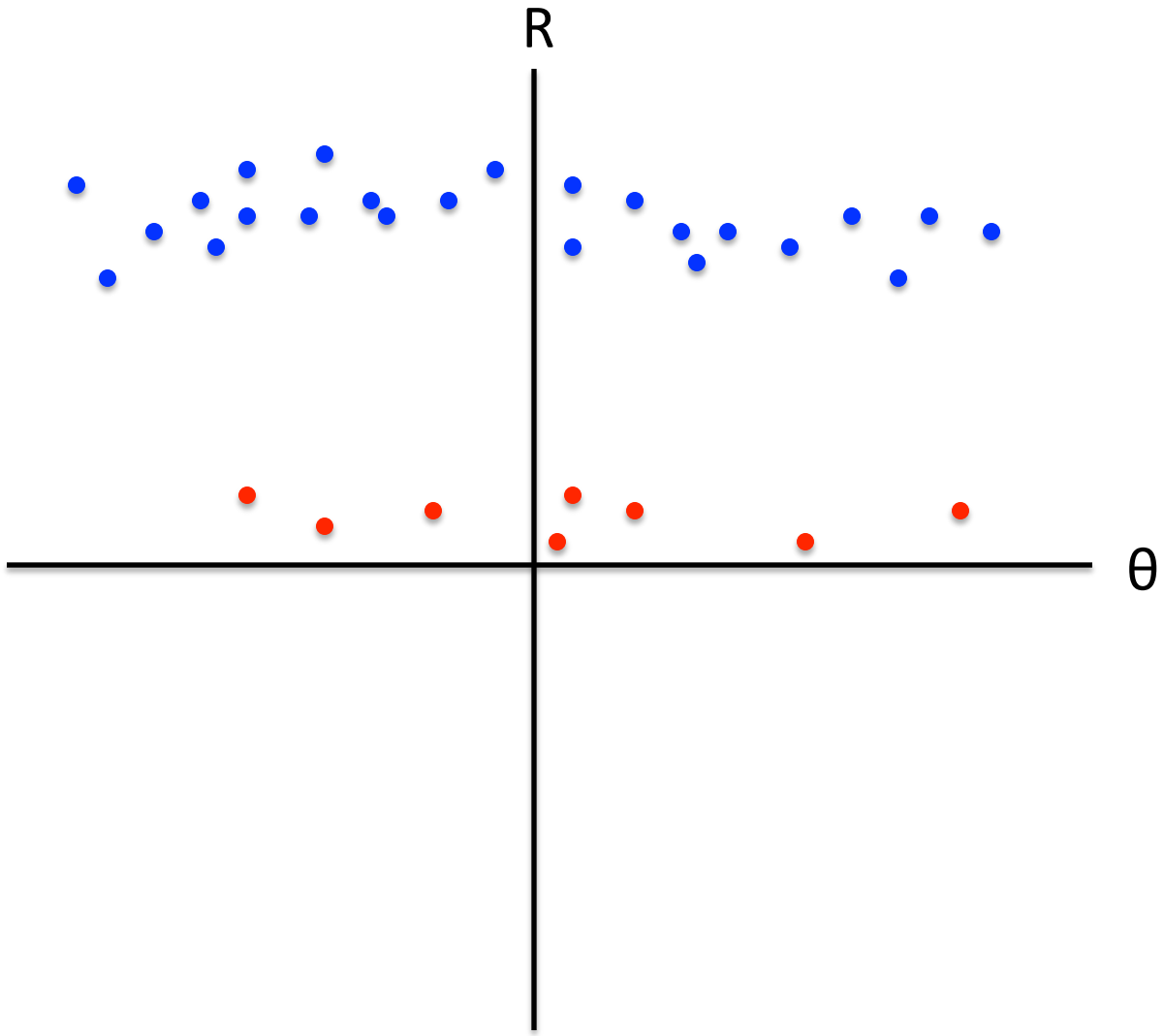
A local optimum:

Another Example



30

Another Example



31

K Means in Practice

- Avoiding getting stuck
  - Random restarts
  - Take restart with best objective value
- Better initialization
  - Kmeans++

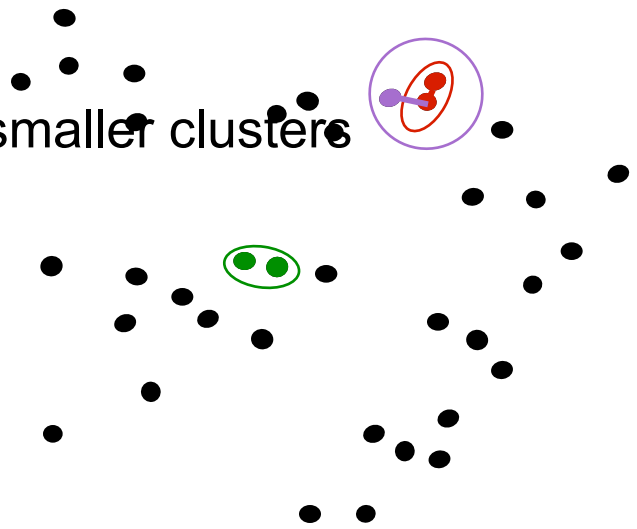
  1. Choose first centroid to be a random datapoint x
  2. For each datapoint, compute distance to nearest centroid so far
  3. Choose next center randomly among the datapoints, but weight the choice by the squared distance to the nearest centroid
  4. Repeat steps 2 and 3 until all centroids are chosen

K-Means Questions

- Will K-means converge?
  - To a global optimum?
- Will it always find the true patterns in the data?
  - If the patterns are very very clear?
- Runtime?
- Do people ever use it?
- How many clusters to pick?

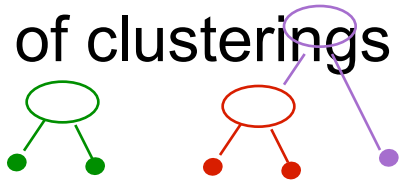
Agglomerative Clustering

- Agglomerative clustering:
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters



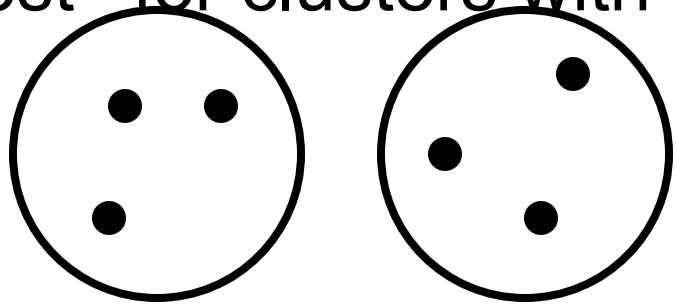
- Algorithm:
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there's only one cluster left

- Produces not one clustering, but a family of clusterings represented by a dendrogram



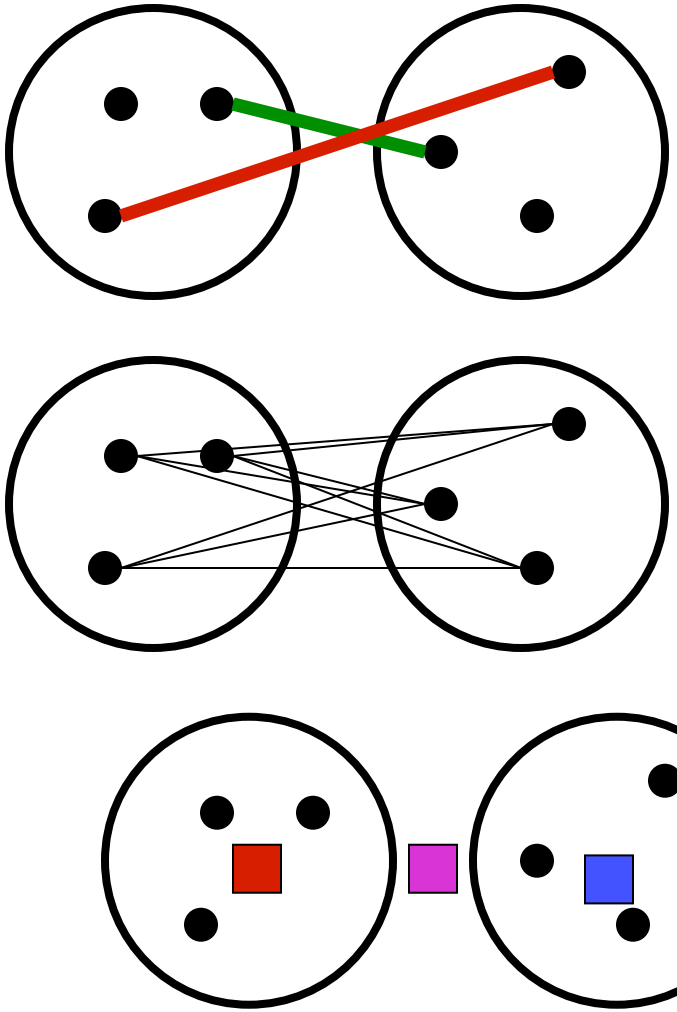
Agglomerative Clustering

- How should we define “closest” for clusters with multiple elements?

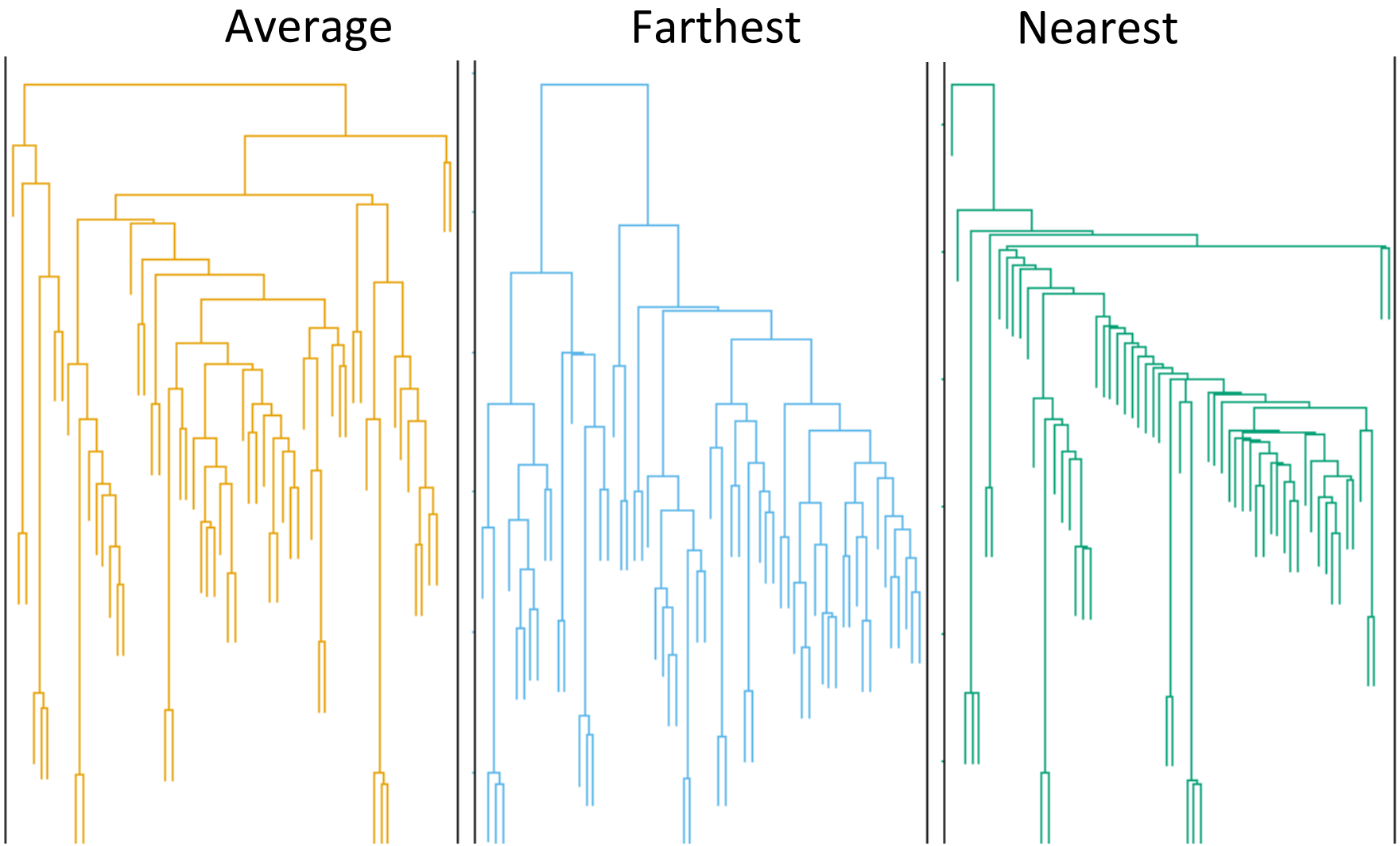


Agglomerative Clustering

- How should we define "closest" for clusters with multiple elements?
- Many options:
  - Closest pair (single-link clustering)
  - Farthest pair (complete-link clustering)
  - Average of all pairs
  - Ward's method
  - (min variance, like k-means)
    - Find pair of clusters that leads to minimum increase in total within cluster variance after merging



Clustering Behavior



Mouse tumor data from [Hastie]

37

Agglomerative Clustering Questions

- Will agglomerative clustering converge?
  - To a global optimum?
- Will it always find the true patterns in the data?
- Do people ever use it?
- How many clusters to pick?

Mixture Models

- Assume data is generated using the following procedure:
  - Pick one of the k components according to $P(z_k)$. This generates a hidden class label z_k.
  - Generate a data point by sampling from $P(x \mid z_k)$.
- Results in probability distribution of a single point

$$p(x^{(i)}) = \sum_{k=1}^{K} P(z_k) p(x^{(i)} | z_k)$$

Where, $P(x \mid z_k)$ is any distribution (gaussian, poisson, exponential, etc)

Gaussian Mixture Model

- Most common mixture model is a Gaussian Mixture Model:

$$p(x|z_k) = \mathcal{N}(\mu_k, \Sigma_k)$$

Clustering with GMMs

- Want to find $P(z_k)$ and $P(x \mid z_k)$ to learn the underlying model and find clusters
- Want to compute $P(z_k \mid x)$ for each point in training set to assign them to clusters
- Can we use maximum likelihood to infer both model and assignments?
  - Requires solving non-linear system of equations
  - No analytic solution

Problem: Missing labels

- If we knew the assignments, we could learn model parameters easily.
  - Parameter estimation!
- If we knew the model parameters, we could estimate the most likely assignments easily.
  - Classification!

Solution: The EM Algorithm

- We deal with missing parameters and labels by alternating between two steps:
  - Expectation:

    Fix model and estimate the missing labels
  - Maximization:

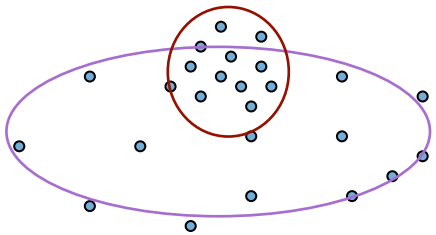
    Fix missing labels (usually a distribution of missing labels) and find the model that maximizes the expected log-likelihood of the data

EM: Soft Clustering

- Clustering typically assumes that each instance is given a “hard” assignment to exactly one cluster.
- Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.
  - Problematic because data points that lie roughly midway between cluster centers are assigned to one cluster
- *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters.

44

Probabilistic Clustering



- Try a probabilistic model!
  - allows overlaps, clusters of different size, etc.
- Can tell a *generative story* for data
  - P(X|Y) P(Y)
- Challenge: we need to estimate model parameters without labeled Ys

Y	X_1	X_2
??	0.1	2.1
??	0.5	-1.1
??	0.0	3.0
??	-0.1	-2.0
??	0.2	1.5
…	…	…

Finite Mixture Models

- Given a dataset: $D = \{\underline{x}_1, \ldots, \underline{x}_N\}$ ($\underline{x}_i$ is a d-dimensional vector)
- **Mixture model**: $\Theta = \{\alpha_1, \ldots, \alpha_K, \theta_1, \ldots, \theta_K\}$

$$p(\underline{x}|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k)$$

The $p_k(\underline{x}|z_k, \theta_k)$ are *mixture components*, $1 \leq k \leq K$

$z = (z_1, \ldots, z_K)$ is a vector of K binary indicator variables

Mixture Weights. $\alpha_k = p(z_k)$ $\sum_{k=1}^{K} \alpha_k = 1.$

46

Finite Mixture Model: Probabilistic View

the "membership weight" of data point $\underline{x}_i$ in cluster k, given parameters Θ

$$w_{ik} = p(z_{ik} = 1 | \underline{x}_i, \Theta) = \frac{p_k(\underline{x}_i | z_k, \theta_k) \cdot \alpha_k}{\sum_{m=1}^{K} p_m(\underline{x}_i | z_m, \theta_m) \cdot \alpha_m}$$

- The membership weight express our uncertainty about which of the "K" components generated the vector $\underline{x}_i$.

47

Gaussian Mixture Models (GMMs)

- We can define a GMM by making each "k-th" component a Gaussian density with parameters:

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t \Sigma_k^{-1}(\underline{x}-\underline{\mu}_k)}$$

$$\theta_k = \{\underline{\mu}_k, \Sigma_k\}$$

Question: How to learn these parameters from data?

48

EM algorithm: Key Idea

- Start with random parameters
- Find a class for each example (E-step)
  - Since we are using probabilistic classification, each example will be given a vector of probabilities
- Now we have a supervised learning problem. Estimate the parameters of the model using the maximum likelihood method (M-step)
- Iterate between the E-step and M-step until convergence

EM: Two Easy Steps

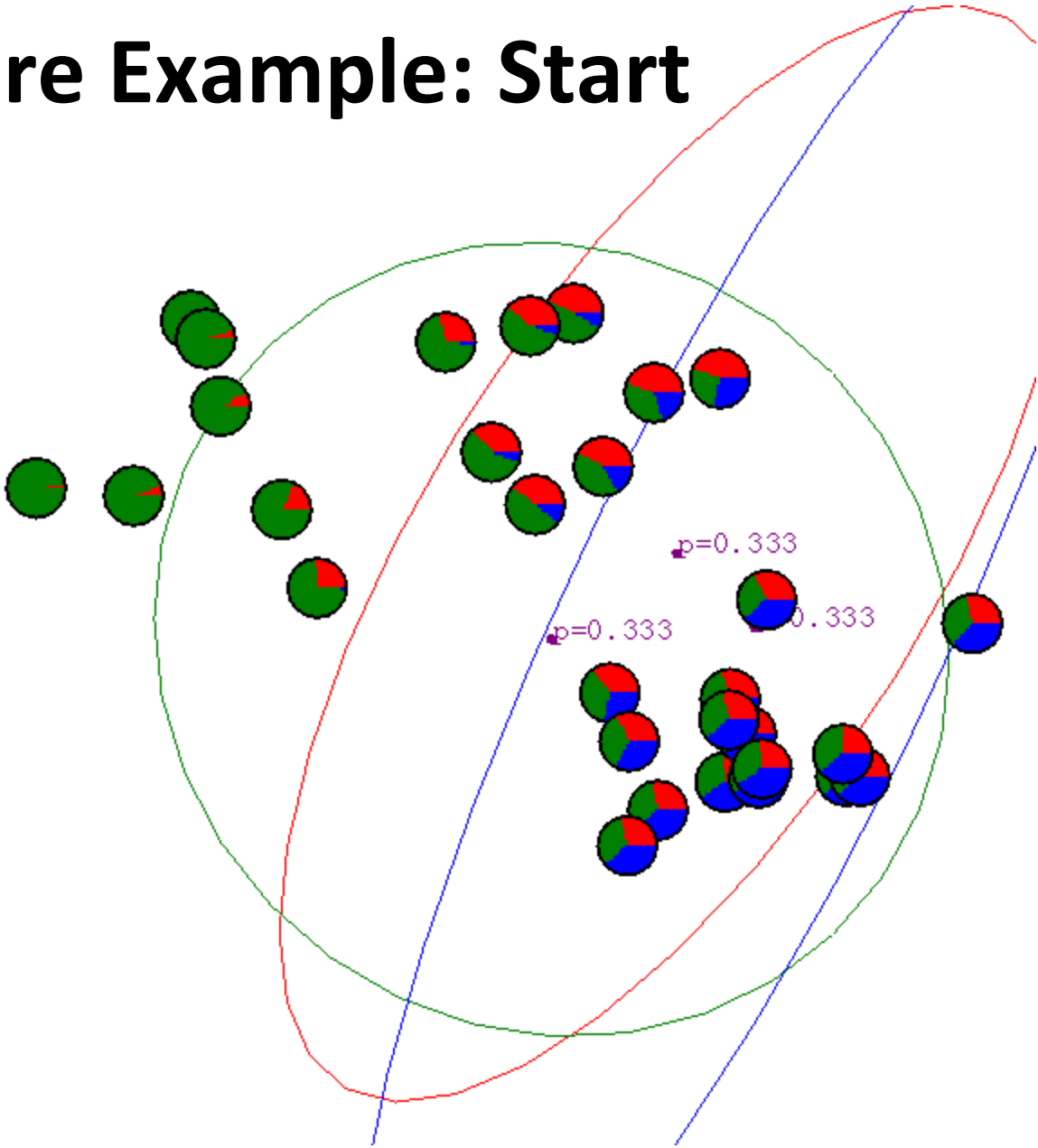
- E-step: (Yields a N x K matrix)
  - Compute w_{ik} for all data points indexed by "i" and all mixture components indexed by "k."
- M-step:
  - Use the membership weights and data to compute the new parameters

$$N_k = \sum_{i=1}^{N} w_{ik} \qquad \alpha_k^{new} = \frac{N_k}{N}$$

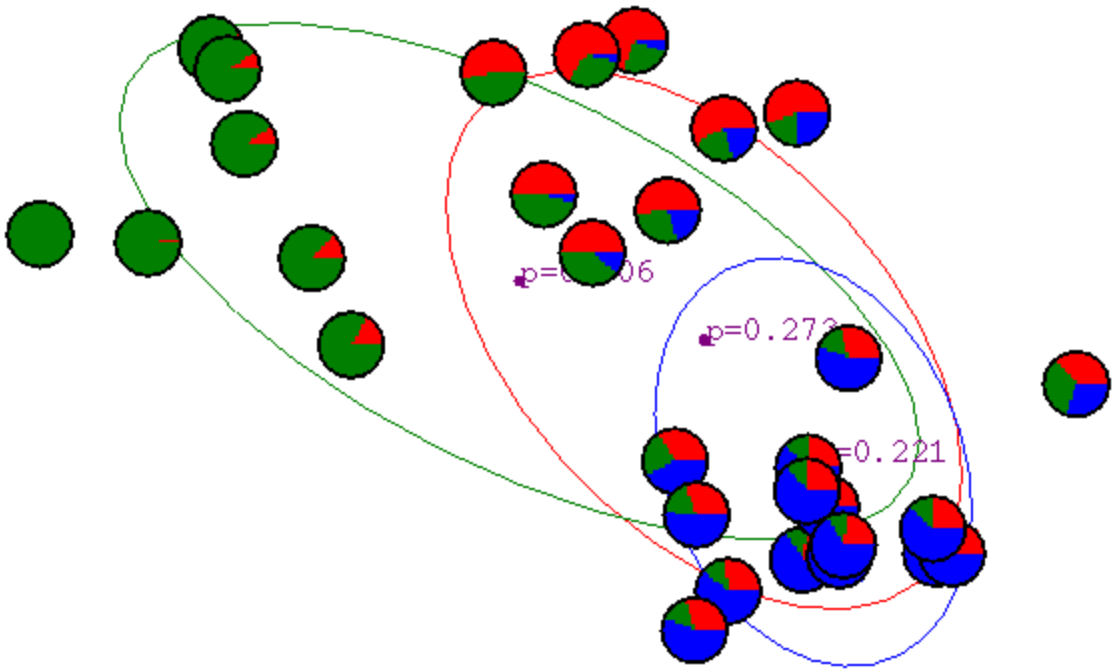
$$\underline{\mu}_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot \underline{x}_i$$

$$\Sigma_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot (\underline{x}_i - \underline{\mu}_k^{new})(\underline{x}_i - \underline{\mu}_k^{new})^t$$

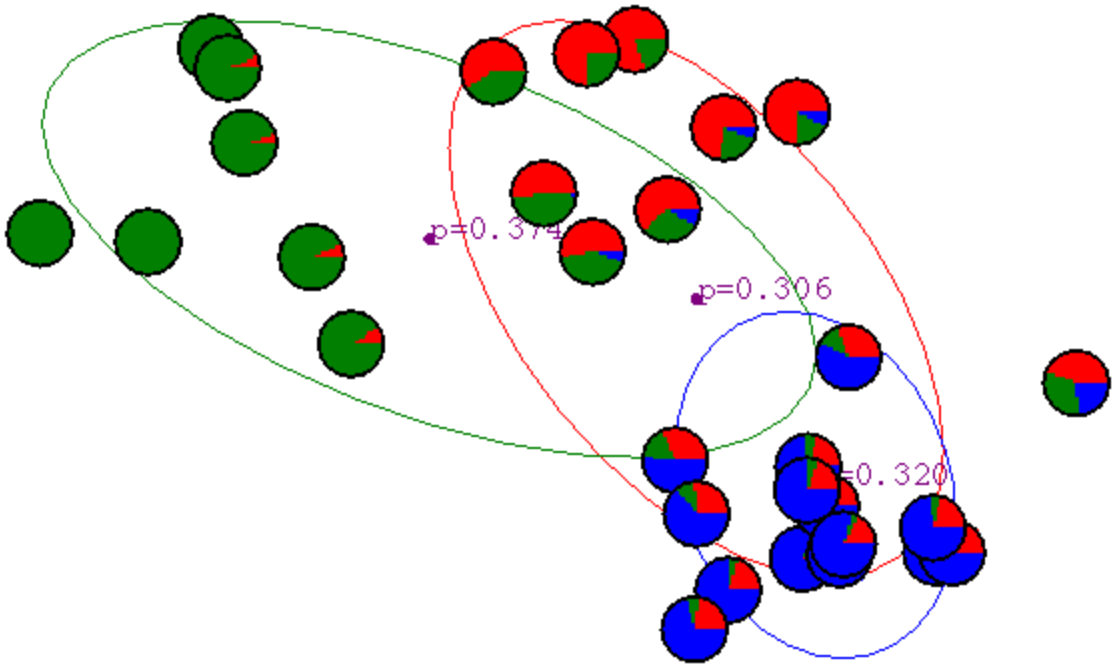
Gaussian Mixture Example: Start



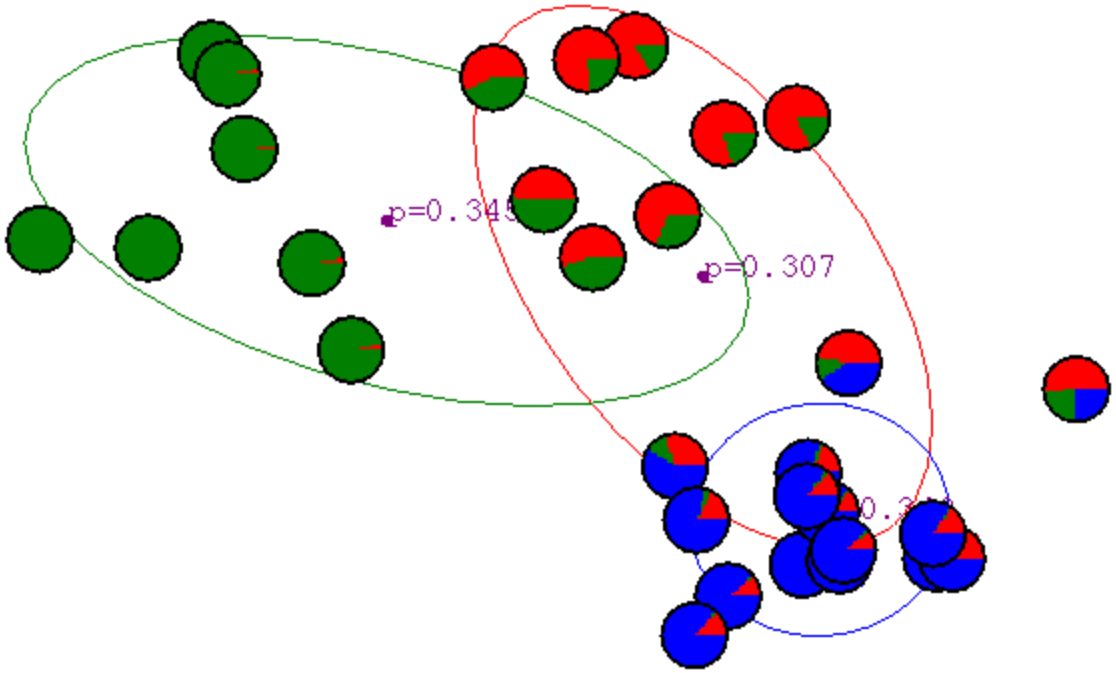
After first iteration



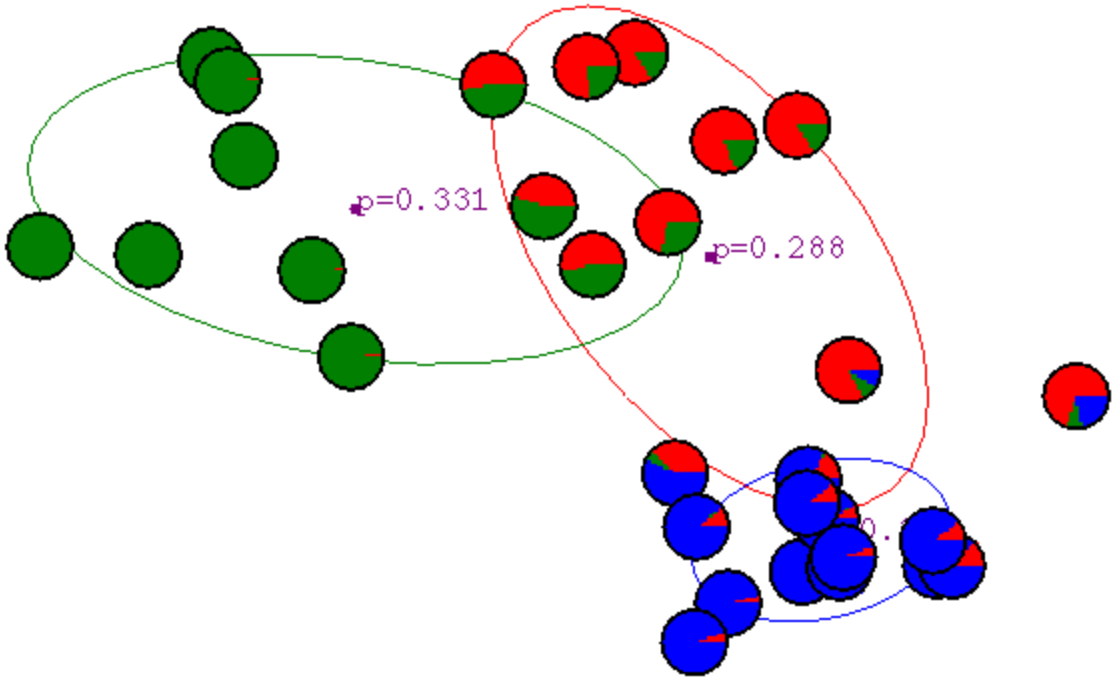
After 2nd iteration



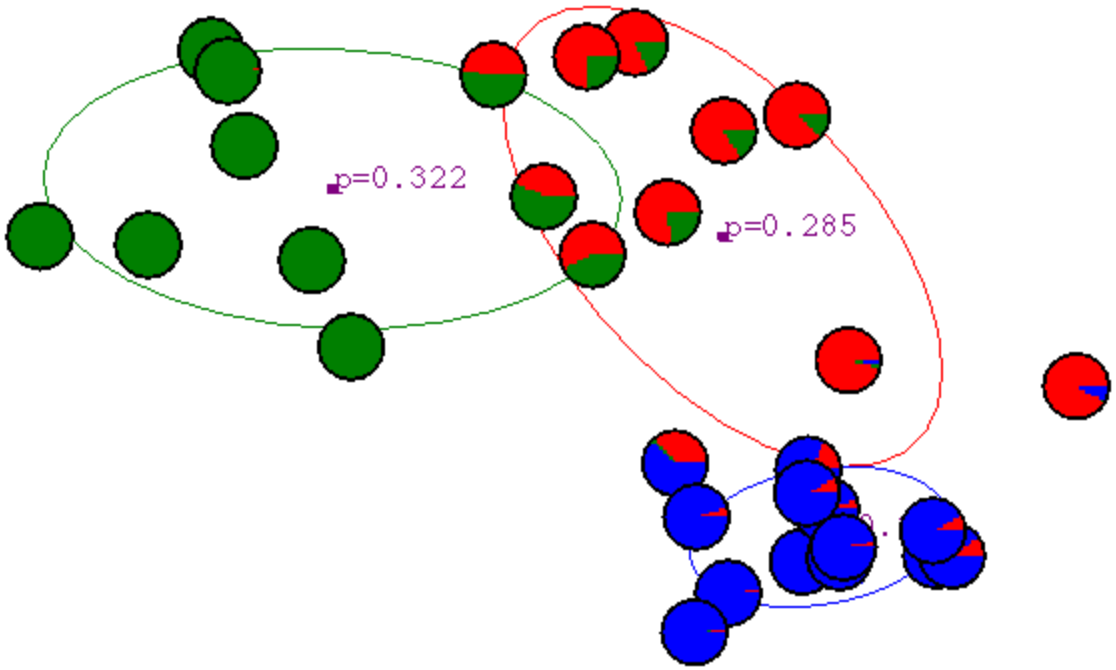
After 3rd iteration



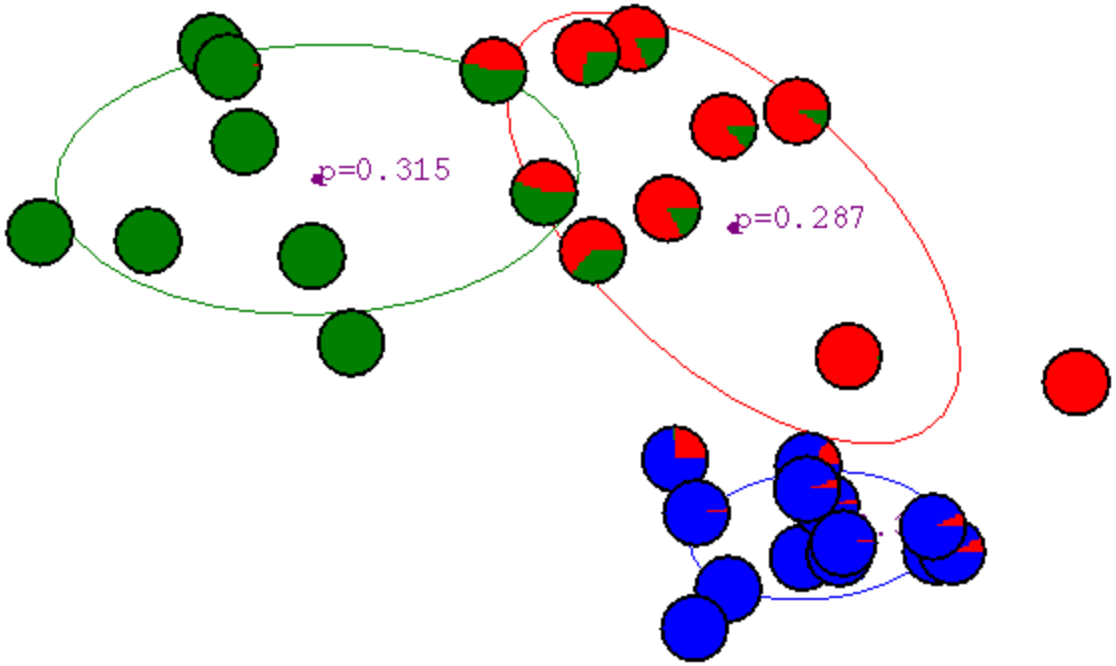
After 4th iteration



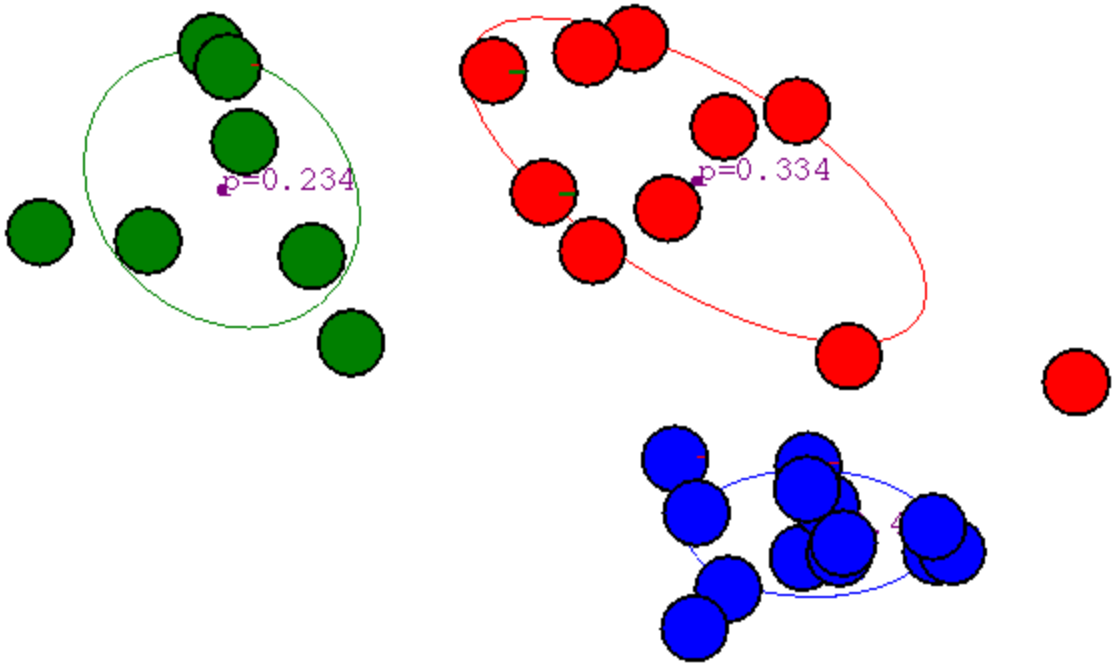
After 5th iteration



After 6th iteration



After 20th iteration



Properties of EM

- EM converges to a local minima
  - This is because each iteration improves the log-likelihood
  - Proof same as K-means
    - E-step can never decrease likelihood
    - M-step can never decrease likelihood
- If we make hard assignments instead of soft ones. Algorithm is equivalent to K-means!

What you should know

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- Know what agglomerative clustering is
- EM for mixture of Gaussians:
- Remember, E.M. can get stuck in local minima,
  - And empirically it ***DOES!***

EM Example

Assume that we have two coins, C1 and C2

Assume the bias of C1 is θ_1

(i.e., probability of getting heads with C1)

Assume the bias of C2 is θ_2

(i.e., probability of getting heads with C2)

We want to find θ_1, θ_2 by performing a number of trials

(i.e., coin tosses)

EM Example

First experiment

- We choose 5 times one of the coins.
- We toss the chosen coin 10 times

$$\theta_1 = \frac{number\ of\ heads\ using\ C1}{total\ number\ of\ flips\ using\ C1}$$

$$\theta_2 = \frac{number\ of\ heads\ using\ C2}{total\ number\ of\ flips\ using\ C2}$$

EM Example

H T T T H H T H T H

H H H H T H H H H H

H T H H H H H T H H

H T H T T T H H T T

T H H H T H H H T H

Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

$$\theta_1 = \frac{24}{24+6} = 0.8$$

$$\theta_2 = \frac{9}{9+11} = 0.45$$

EM Example

Assume a more challenging problem

H T T T H H T H T H

H H H H T H H H H H

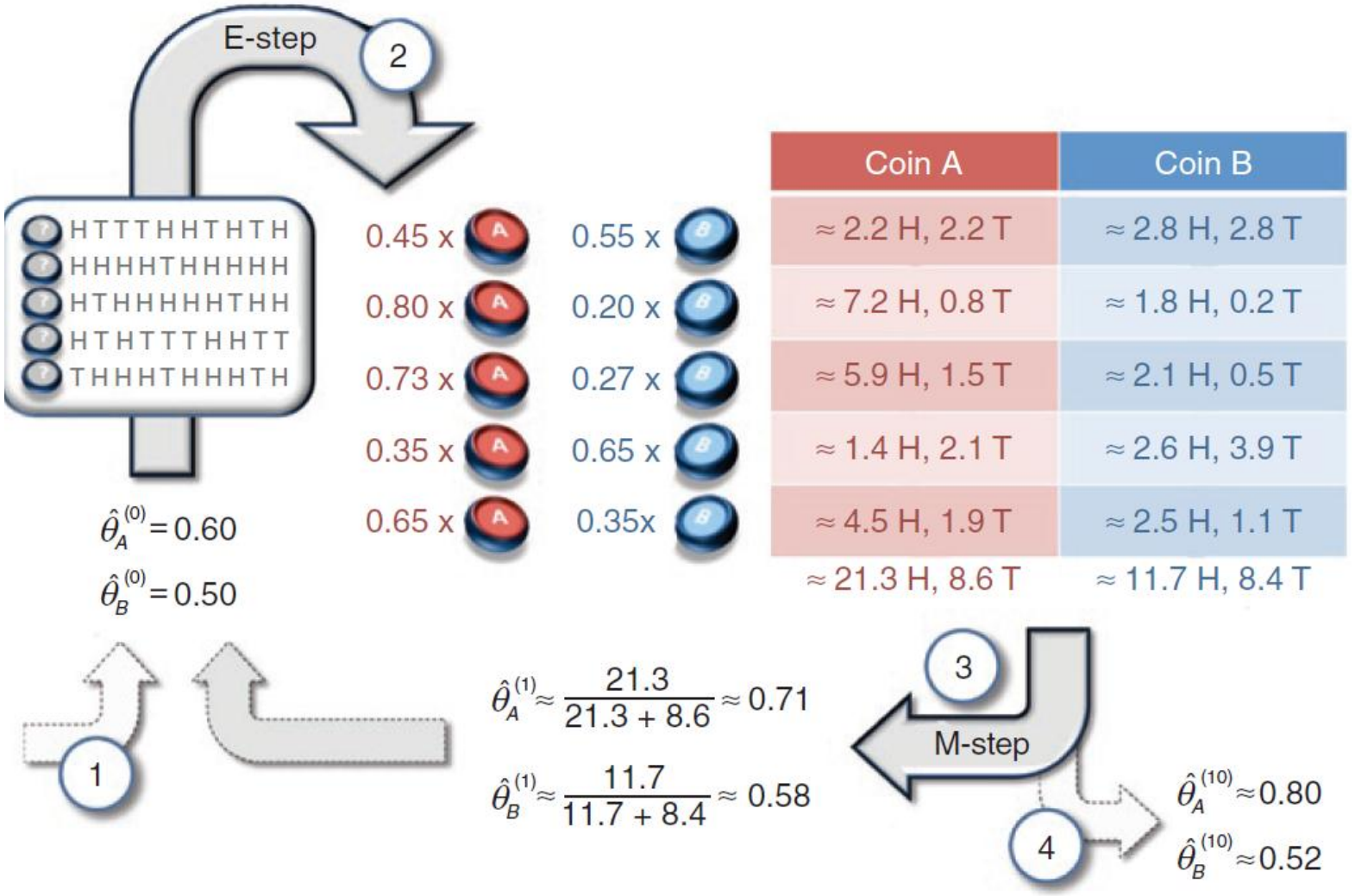
H T H H H H H T H H

H T H T T T H H T T

T H H H T H H H T H

- We do not know the identities of the coins used for each set of tosses (we treat them as hidden variables).

EM: The Intuition



EM: The Math

$$p(X_1, X_2, \cdots, X_5, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5 | \theta) \qquad \boldsymbol{z}_i = \begin{bmatrix} z_{i1} \\ z_{i2} \end{bmatrix} \in \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

$$= p(\{x_1{}^1, \cdots, x_1{}^{10}\}, \cdots, \{x_5{}^1, \cdots, x_5{}^{10}\}, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5 | \theta)$$

$$= p(\{x_1{}^1, \cdots, x_1{}^{10}\}, \cdots, \{x_5{}^1, \cdots, x_5{}^{10}\} | \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5, \theta) \, p(\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5)$$

$$= \prod_{i=1}^{5} p(\{x_i{}^1, \cdots, x_i{}^{10}\} | \boldsymbol{z}_i, \theta) \prod_{i=1}^{5} p(\boldsymbol{z}_i)$$

$$p(\boldsymbol{z}_i) = \prod_{k=1}^{2} \pi_\kappa{}^{z_{ik}} \quad \pi_\kappa \textit{ is the probability of selecting coin } k \in \{1,2\}$$

$$p(\{x_i{}^1, \cdots, x_i{}^{10}\} | \boldsymbol{z}_i, \theta) = \prod_{j=1}^{10} p(x_i{}^j | \boldsymbol{z}_i, \theta)$$

EM: The Math

$x_i{}^j = 1$ *If* j *toss of* i *run is head*

$x_i{}^j = 0$ *If* j *toss of* i *run is head*

$$p\left(x_i{}^j \middle| \boldsymbol{z}_i, \theta\right) = \prod_{k=1}^{2} \left[\theta_k{}^{x_i{}^j} (1-\theta_k)^{1-x_i{}^j}\right]^{z_{ik}}$$

then

$$p\left(X_1, X_2, \cdots, X_5, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5 | \theta\right)$$

$$= \prod_{i=1}^{5} \prod_{j=1}^{10} \prod_{k=1}^{2} \left[\theta_k{}^{x_i{}^j} (1-\theta_k)^{1-x_i{}^j}\right]^{z_{ik}} \prod_{i=1}^{5} \prod_{k=1}^{2} \pi_\kappa{}^{z_{ik}}$$

EM: Computing the Expectation

$$\ln p\left(X_1, X_2, \cdots, X_5, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5 | \theta\right)$$

$$= \sum_{i=1}^{5} \sum_{j=1}^{10} \sum_{k=1}^{2} z_{ik} \ln \theta_k{}^{x_i{}^j} (1 - \theta_k)^{1 - x_i{}^j} + \sum_{i=1}^{5} \sum_{k=1}^{2} z_{ik} \ln \pi_\kappa$$

Taking the expectation of the above

$$E_{p(Z|X)}\left[\ln p\left(X_1, X_2, \cdots, X_5, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5 | \theta\right)\right]$$

$$= \sum_{i=1}^{5} \sum_{j=1}^{10} \sum_{k=1}^{2} E_{p(Z|X)}[z_{ik}] \ln \theta_k{}^{x_i{}^j} (1 - \theta_k)^{1 - x_i{}^j} + \sum_{i=1}^{5} \sum_{k=1}^{2} E_{p(Z|X)}[z_{ik}] \ln \pi_\kappa$$

EM: Expectation Step

$$p(\boldsymbol{Z}|\boldsymbol{X},\theta) = p\left(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_5|X_1,X_2,\cdots,X_5,\theta\right)$$

$$E_{p(\boldsymbol{Z}|\boldsymbol{X})}[z_{ik}] = \sum_{\boldsymbol{z}_1}\cdots\sum_{\boldsymbol{z}_5} z_{ik}\, p(\boldsymbol{Z}|\boldsymbol{X},\theta) = \sum_{\boldsymbol{z}_i} z_{ik} p(\boldsymbol{z}_i|\{{x_i}^1,\cdots,{x_i}^{10}\})$$

$$p(\boldsymbol{z}_i|\{{x_i}^1,\cdots,{x_i}^{10}\}) = \frac{p(\{{x_i}^1,\cdots,{x_i}^{10}\}|\boldsymbol{z}_i,\theta)p(\boldsymbol{z}_i)}{p(\{{x_i}^1,\cdots,{x_i}^{10}\}|\theta)}$$

$$= \frac{\prod_{j=1}^{10}\prod_{k=1}^{2}\left[{\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}\right]^{z_{ik}}{\pi_\kappa}^{z_{ik}}}{\sum_{\boldsymbol{z}_i}\prod_{j=1}^{10}\prod_{k=1}^{2}\left[{\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}\right]^{z_{ik}}{\pi_\kappa}^{z_{ik}}}$$

EM: Expectation Step

$$E_{p(\boldsymbol{Z}|\boldsymbol{X})}[z_{ik}] = \sum_{\boldsymbol{z}_i} z_{ik} \frac{\prod_{j=1}^{10}\prod_{k=1}^{2}\left[{\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}\right]^{z_{ik}}{\pi_\kappa}^{z_{ik}}}{\sum_{\boldsymbol{z}_i}\prod_{j=1}^{10}\prod_{k=1}^{2}\left[{\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}\right]^{z_{ik}}{\pi_\kappa}^{z_{ik}}}$$

$$= \frac{\sum_{\boldsymbol{z}_i} z_{ik}\prod_{j=1}^{10}\prod_{k=1}^{2}\left[{\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}\right]^{z_{ik}}{\pi_\kappa}^{z_{ik}}}{\sum_{\boldsymbol{z}_i}\prod_{j=1}^{10}\prod_{k=1}^{2}\left[{\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}\right]^{z_{ik}}{\pi_\kappa}^{z_{ik}}}$$

EM: Expectation Step

Expectation (E) Step (fix θ):

$$\pi_1 = \frac{1}{2} \quad \pi_2 = \frac{1}{2}$$

$$E_{p(\boldsymbol{Z}|\boldsymbol{X})}[z_{ik}] = \frac{\prod_{j=1}^{10} {\theta_k}^{{x_i}^j}(1-\theta_k)^{1-{x_i}^j}}{\prod_{j=1}^{10} {\theta_1}^{{x_i}^j}(1-\theta_1)^{1-{x_i}^j} + \prod_{j=1}^{10} {\theta_2}^{{x_i}^j}(1-\theta_2)^{1-{x_i}^j}}$$

EM: Maximization Step

Maximization Step (fix $E_{p(\boldsymbol{Z}|\boldsymbol{X})}[z_{ik}]$):

$$\max L(\theta) = E_{p(Z|X)}\left[\ln p\left(X_1, X_2, \cdots, X_5, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_5 | \theta\right)\right]$$

$$= \sum_{i=1}^{5}\sum_{j=1}^{10}\sum_{k=1}^{2} E_{p(Z|X)}[z_{ik}] \ln \theta_k{}^{x_i{}^j}(1-\theta_k)^{1-x_i{}^j}_{5\quad 2} + \sum_{i=1}^{5}\sum_{k=1}^{2} E_{p(Z|X)}[z_{ik}] \ln \pi_\kappa$$

$$= \sum_{i=1}^{5}\sum_{j=1}^{10}\sum_{k=1}^{2} E_{p(Z|X)}[z_{ik}]\left(x_i{}^j \ln\theta_k + \left(1 - x_i{}^j\right)\ln(1-\theta_k)\right) + const$$

EM: Maximization Step

$$\frac{dL(\theta)}{d\theta_1} = \sum_{i=1}^{5}\sum_{j=1}^{10} E_{p(Z|X)}[z_{i1}]\left(x_i{}^j \frac{1}{\theta_1} - \left(1 - x_i{}^j\right)\frac{1}{1-\theta_1}\right) = 0$$

$$\sum_{i=1}^{5}\sum_{j=1}^{10} E_{p(Z|X)}[z_{i1}]\left(x_i{}^j(1-\theta_1) - \left(1 - x_i{}^j\right)\theta_1\right) = 0$$

$$\theta_1 = \frac{\sum_{i=1}^{5}\sum_{j=1}^{10} E_{p(Z|X)}[z_{i1}]x_i{}^j}{\sum_{i=1}^{5} 10\, E_{p(Z|X)}[z_{i1}]} \qquad \theta_2 = \frac{\sum_{i=1}^{5}\sum_{j=1}^{10} E_{p(Z|X)}[z_{i2}]x_i{}^j}{\sum_{i=1}^{5} 10\, E_{p(Z|X)}[z_{i2}]}$$

PrACTICAL ADVICE:
How to use machine learning for solving real world problems!

Instructor: Arti Ramesh

Steps in Supervised Learning

1. Determine the representation for “x,f(x)” and determine what “x” to use

   Feature Engineering

2. Gather a training set (not all data is kosher)

   Data Cleaning

3. Select a suitable evaluation method
4. Find a suitable learning algorithm among a plethora of available choices

Feature Engineering is the Key

- Most effort in ML projects is constructing features
- Black art: Intuition, creativity required
  - Understand properties of the task at hand
  - How the features interact with or limit the algorithm you are using.
- ML is an iterative process
  - Try different types of features, experiment with each and then decide which feature set/algorithm combination to use

What features will you use?

- Examples
  - Spam Filtering
  - Mapping images to names
- Feature Combination
  - Linear models cannot handle some dependencies between features (e.g. XOR)
  - Feature combinations might work better.
  - Quick growth of the number of features.

Evaluation

- Accuracy
  - Fraction of the examples that are correctly classified by the learner
- Precision, Recall and F-score (Next slide)
- Squared error (Regression problems)
- Likelihood
- Posterior probability
- Cost / Utility
- Etc.

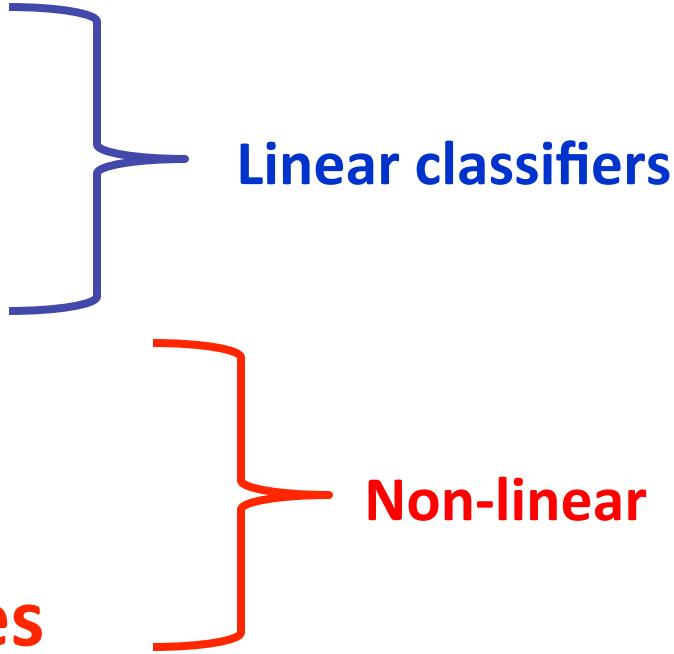
Precision, Recall and F-1 score

- Precision (P) = ; Recall (R)=
- F1-score =
  - Harmonic mean of precision and recall

	Actual=True	**Actual=False**
Predicted=True	***tp*** (correct result)	***fp*** (unexpected result)
Predicted=False	***fn*** (missing result)	***tn*** (correct absence of result)

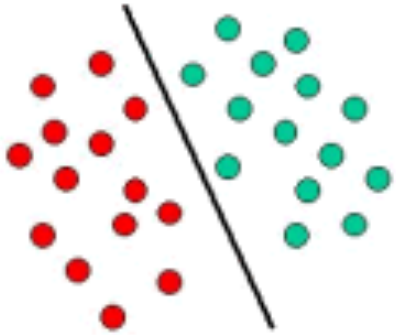
What algorithms (Classifiers/learners) to use

- Naïve Bayes
- Logistic Regression
- Linear SVMs
- Decision Trees
- Neural Networks
- Support Vector Machines
- K-nearest neighbors (Non-parametric)
- Bagging (Meta); Boosting (Meta)



Classifiers: Bias versus Variance

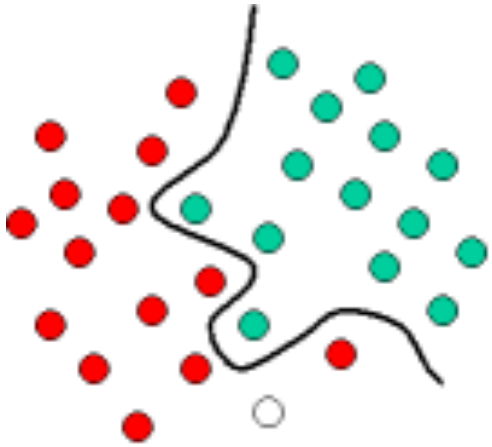
High Bias, low variance



Linear Classifier
- Naïve Bayes
- Logistic Regression
- Perceptron

Simpler

Low Bias, High variance



Non-Linear Classifier
- Support vector machine (Kernels)
- Neural networks
- Decision Trees

Complex

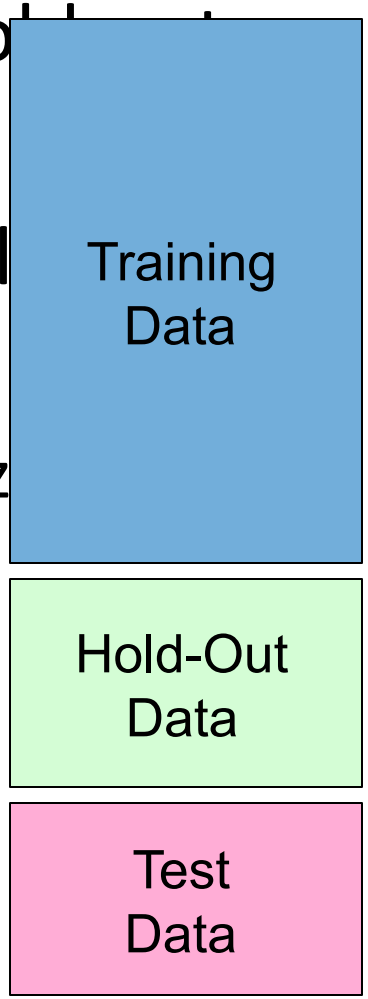
Learning = Representation + Evaluation + Optimization

- Thousands of learning algorithms
- Combinations of just three elements

Representation	Evaluation	Optimization
Instances	Accuracy	Greedy search
Hyperplanes	Precision/Recall	Branch & bound
Decision trees	Squared error	Gradient descent
Sets of rules	Likelihood	Quasi-Newton
Neural networks	Posterior prob.	Linear progr.
Graphical models	Margin	Quadratic progr.
Etc.	Etc.	Etc.

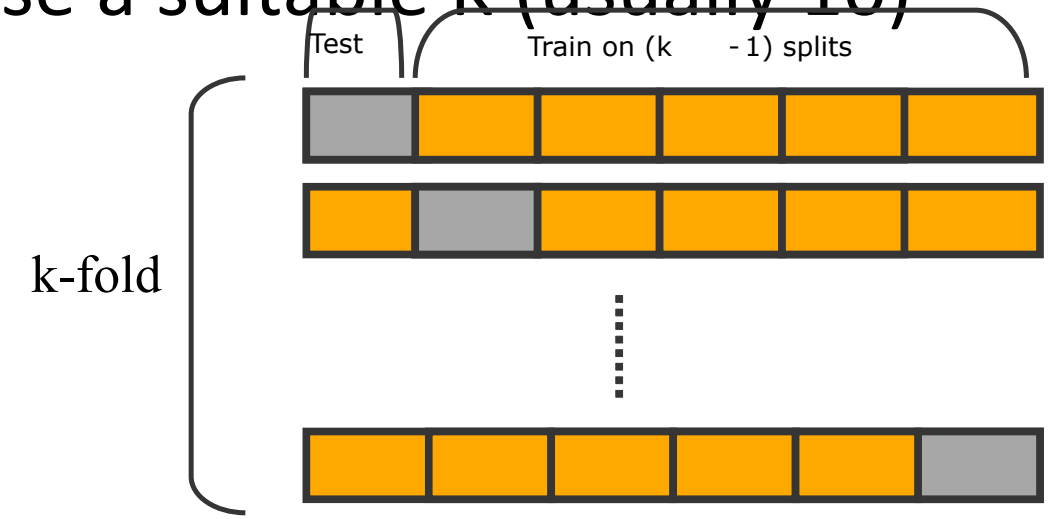
It's Generalization that Counts

- Divide data into training, test and ho validation set
- Algorithm must work on test exampl seen before
  - Training examples can just be memoriz
- Don't tune parameters on test data
- Use cross-validation



K-Fold Cross Validation

- Choose a suitable K (usually 10)



Data Alone Is Not Enough

- Classes of unseen examples are arbitrary
- So learner must make assumptions
- “No free lunch” theorems
- Luckily, real world is not random
- Induction is knowledge lever

Overfitting Has Many Faces

- **Classifier A is better than B on the training set but the reverse is true on the test set!!**
- The biggest problem in machine learning
- Bias and variance (Simple vs. Complex)
  - Can learn a simpler linear function vs. can learn any function
- Less powerful learners can be better
- Solutions: Cross-validation; Regularization

Intuition Fails In High Dimensions

- Curse of dimensionality
- Sparseness worsens exponentially with number of features
- Irrelevant features ruin similarity
- In high dimensions all examples look alike
- 3D intuitions do not apply in high dimensions
- Blessing of non-uniformity

Theoretical Guarantees Are Not What They Seem

- Bounds on number of examples needed to ensure good generalization
- Extremely loose
- Low training error ≠> Low test error
- Asymptotic guarantees may be misleading
- Theory is useful for algorithm design, not evaluation

More Data Beats a Cleverer Algorithm

- Easiest way to improve: More data
- Then
  - Data is bottleneck
- Now:
  - Scalability is bottleneck
- ML algorithms more similar than they appear
- Clever algorithms require more effort but can pay off in the end
- Biggest bottleneck is human time

Learn Many Models, Not Just One

- Three stages of machine learning
  1. Try variations of one algorithm, chose one
  2. Try variations of many algorithms, choose one
  3. Combine many algorithms, variations
- Ensemble techniques
  - Bagging
  - Boosting
  - Stacking
  - Etc.

Simplicity Does Not Imply Accuracy

- Occam's razor
- Common misconception:
  Simpler classifiers are more accurate
- Contradicts "no free lunch" theorems
- Counterexamples: ensembles, SVMs, etc.
- Can make preferred hypotheses shorter

Representable Does Not Imply Learnable

- Standard claim: “My language can represent/approximate any function”
- No excuse for ignoring others
- Causes of non-learnability
  - Not enough data
  - Not enough components
  - Not enough search
- Some representations exponentially more compact than others

Advanced topics

Supervised Learning and its Generalizations

- Supervised Learning
  - Desired output is simple. (e.g., purchase an item or not; the person has the disease or not; etc.)
- Structured Prediction: is a Generalization
  - Desired output is complex.

Structured Prediction: Examples

- Parsing: given an input sequence, build a tree whose leaves are the elements in the sequence and whose structure obeys some grammar.
- Collective classification: given a graph defined by a set of vertices and edges, produce a labeling of the vertices.
  - Labeling web pages given link information

Models and Algorithms for Structured Prediction

- Probabilistic Graphical Models
  - Compact representation of joint distribution
  - Principled way of dealing with uncertainty
  - Take advantage of conditional independence
- Markov logic and statistical relational models
  - Model both relational structure and uncertainty
    - One example related with another example
- Considerable machine learning expertise required here! (not yet a blackbox)