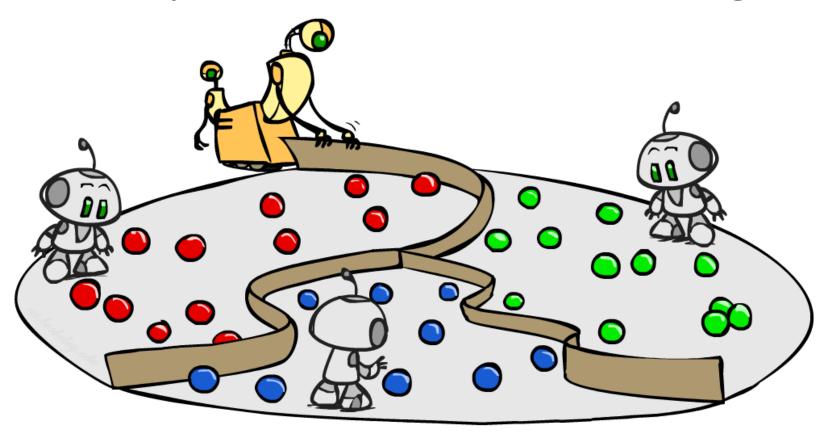
Unsupervised Machine Learning



AIMA Chapter 20

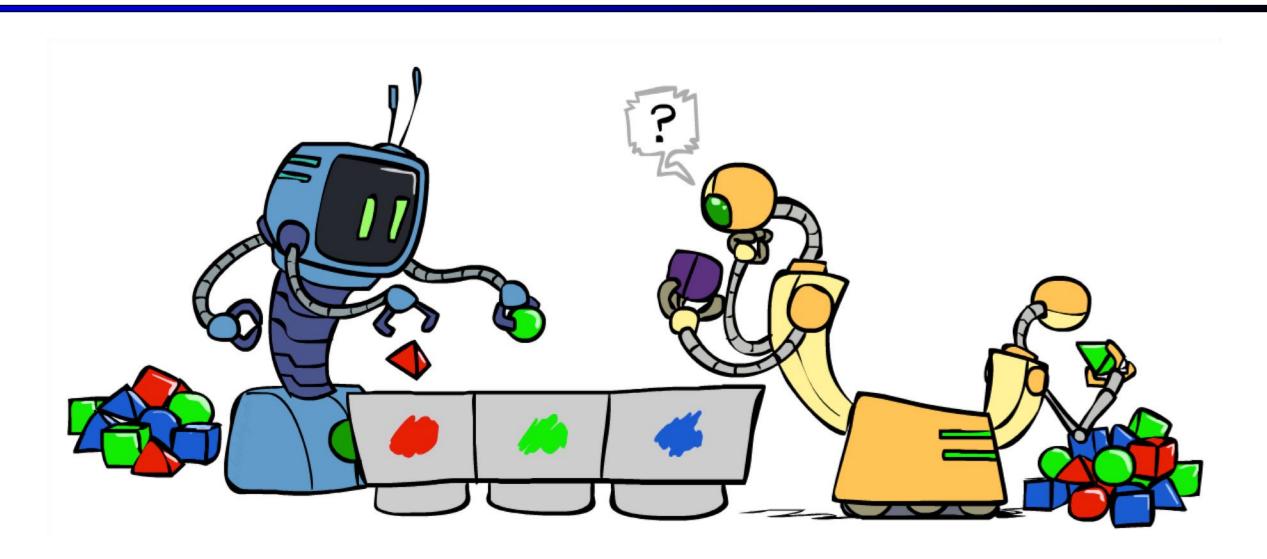
Types of Learning

- Supervised learning
 - Training data includes desired outputs
- Unsupervised learning



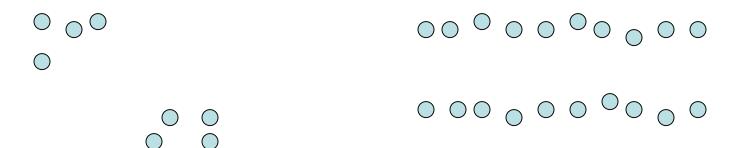
- Training data does not include desired outputs
- Semi-supervised learning
 - Training data includes a few desired outputs
- Reinforcement learning
 - Rewards from sequence of actions

Clustering



Clustering

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



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

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

Many other options, often domain specific

Clustering



- Group emails
- Group search results
- Find categories of customers
- Detect anomalous program executions

Story groupings: unsupervised clustering



World »

edit 🗵

Heavy Fighting Continues As Pakistan Army Battles Taliban

Voice of America - 10 hours ago

By Barry Newhouse Pakistan's military said its forces have killed 55 to 60 Taliban militants in the last 24 hours in heavy fighting in Taliban-held areas of the northwest. Pakistani troops battle Taliban militants for fourth day quardian.co.uk Army: 55 militants killed in Pakistan fighting. The Associated Press.

Christian Science Monitor - CNN International - Bloomberg - New York Times

all 3,824 news articles »



Sri Lanka admits bombing safe haven

quardian.co.uk - 3 hours ago

Sri Lanka has admitted bombing a "safe haven" created for up to 150000 civilians fleeing fighting between Tamil Tiger fighters and the army.

Chinese billions in Sri Lanka fund battle against Tamil Tigers Times Online Huge Humanitarian Operation Under Way in Sri Lanka Voice of America

BBC News - Reuters - AFP - Xinhua

all 2,492 news articles »



edit 🗵 Business »

Buffett Calls Investment Candidates' 2008 Performance Subpar

γ Hugh Son, Erik Holm and Andrew Frye May 2 (Bloomberg) -- Billionaire Warren Buffett said all of 🗽 candidates to replace him as chief investment officer of Berkshire Hathaway Inc. failed to beat the 38 percent decline of the Standard & Poor's 500 ...

uffett offers bleak outlook for US newspapers. Reuters

Buffer: Limit CEO pay through embarrassment MarketWatch

CNBC - The Associated Press - quardian.co.uk

all 1,454 news articles » Main

Chrysler's Fall May Help Administration Reshape GM

New York Times - 5 hours ago

🚧 to task force members, from left: Treasury's Ron Bloom and Gene Sperling, Labor's Edward Montgomery, and Steve Rattner. BY DAVID E. SANGER and BILL VLASIC WASHINGTON - Fresh from pushing Chrysler into bankruptcy, President Obama and his economic team ...

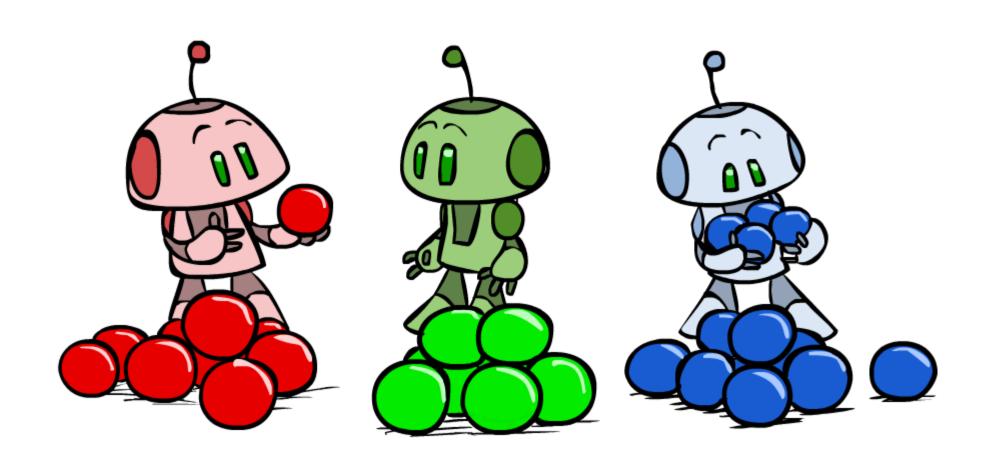


Comment by Gary Chaison Prof. of Industrial Relations, Clark University Bankruptcy reality sets in for Chrysler, workers | Detroit Free Press

Washington Post - Bloomberg - CNNMoney.com

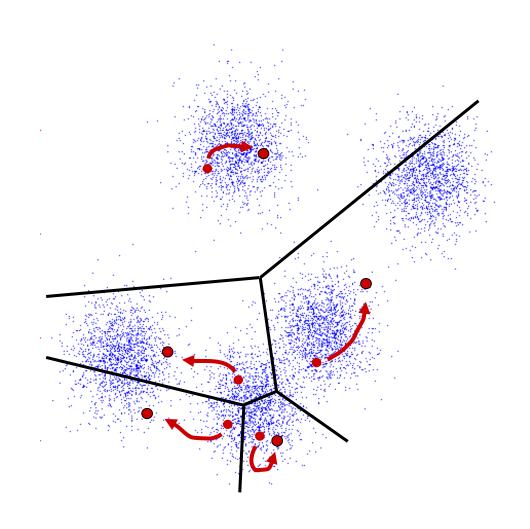
all 11,028 news articles .. OTC:FIATY - BIT:FR - GM

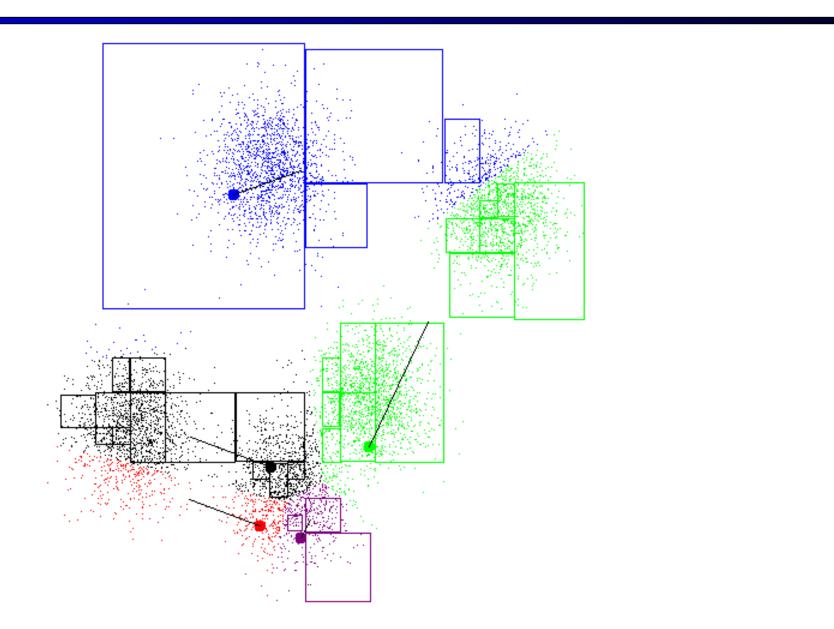
K-Means

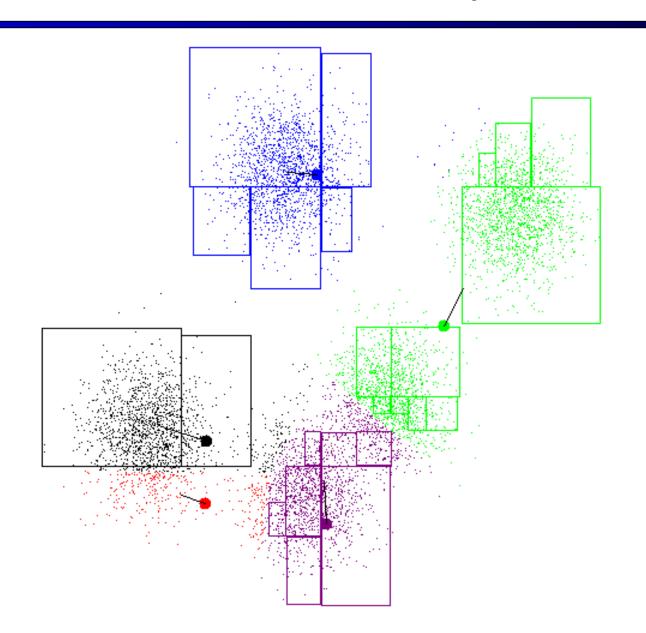


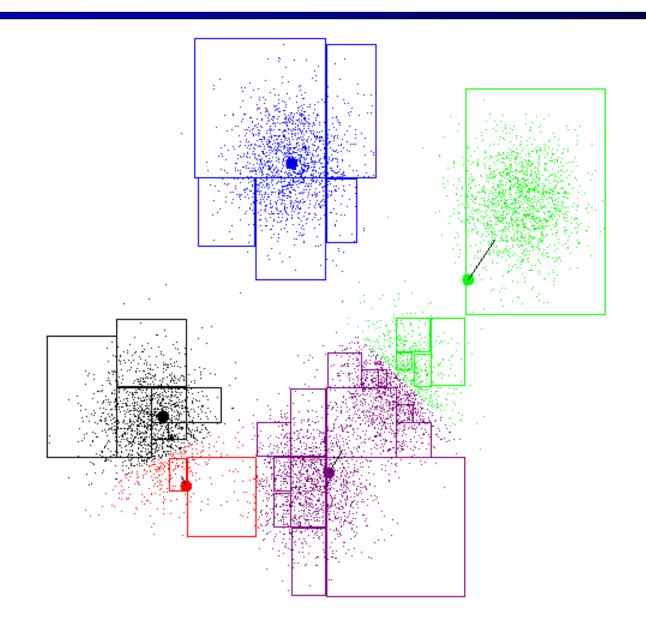
K-Means

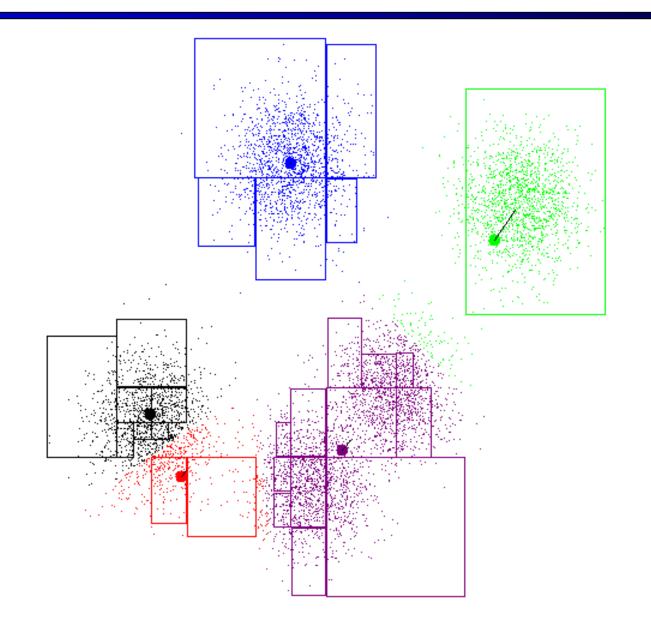
- An iterative clustering algorithm
 - Pick K random points as cluster centers (means)
 - Alternate:
 - Assign data instances to closest mean
 - Assign each mean to the average of its assigned points
 - Stop when no points' assignments change

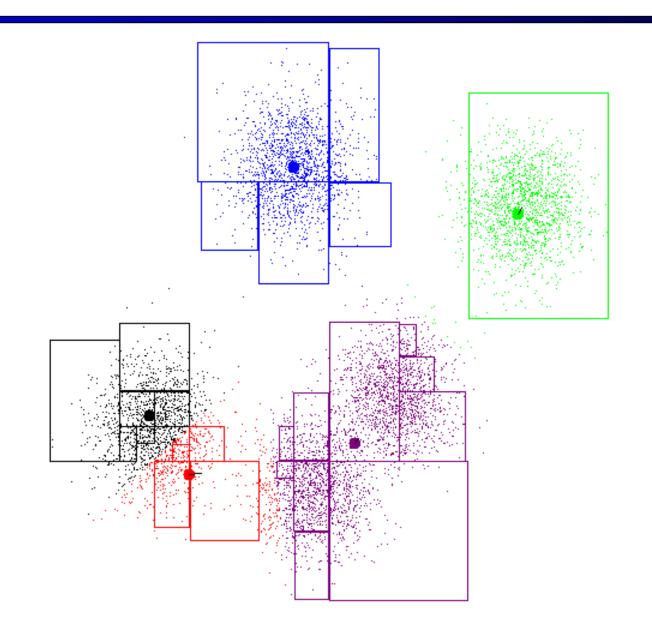


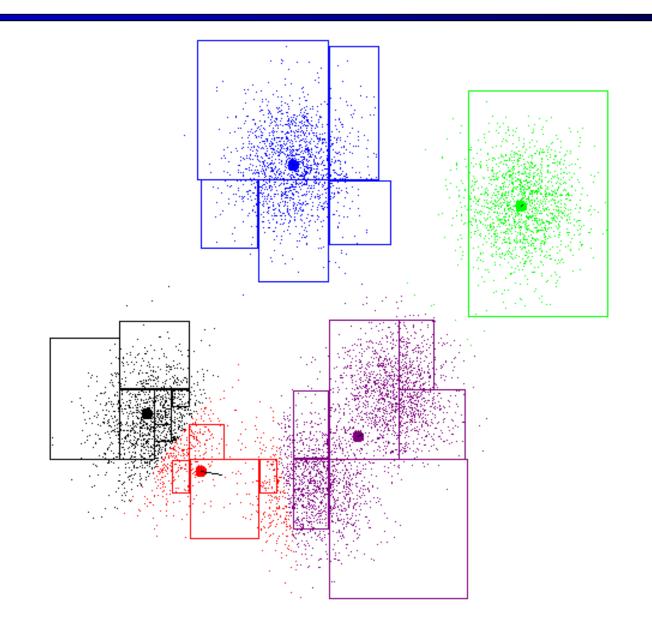


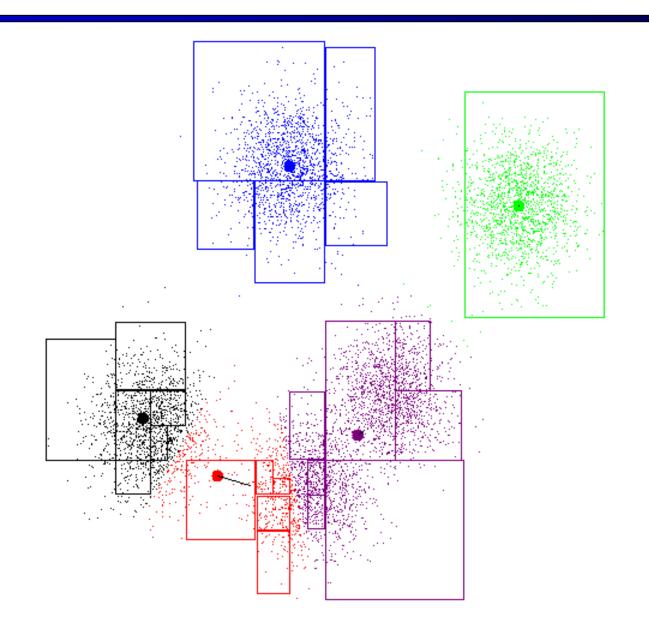


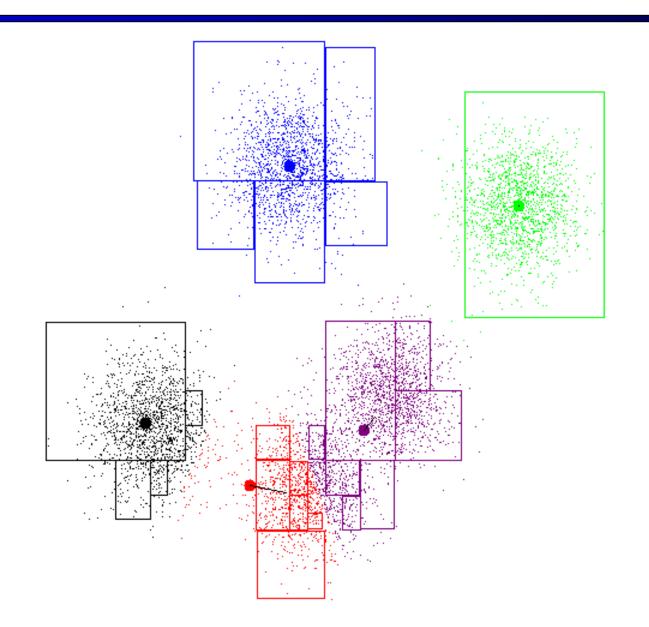


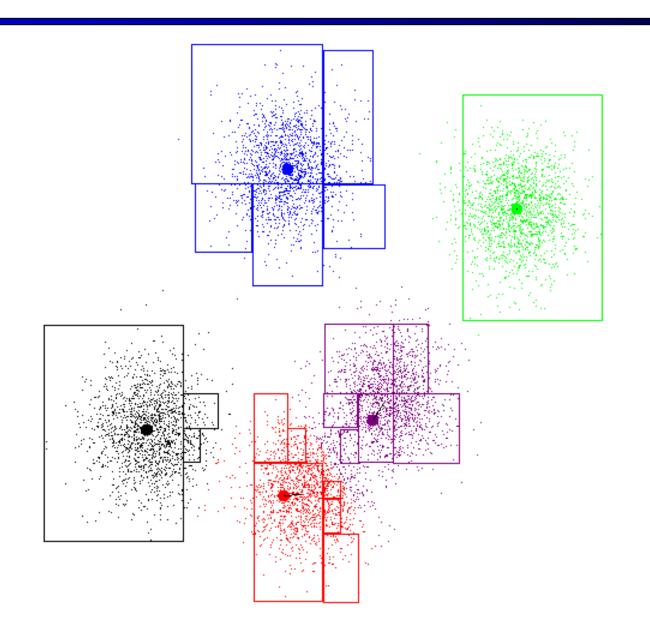


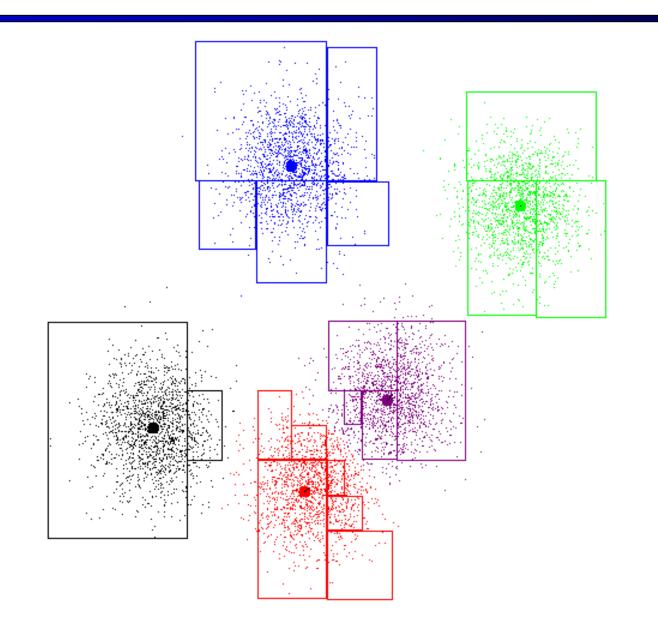






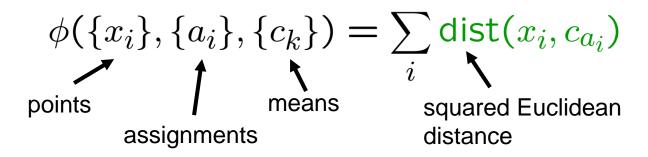


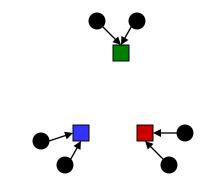




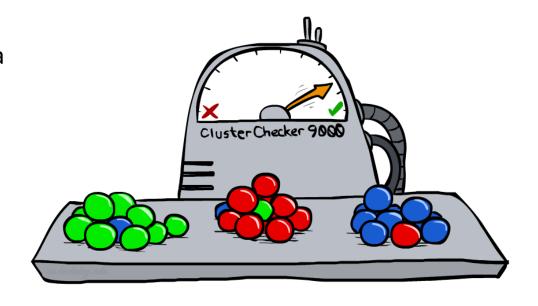
K-Means as Optimization

Consider the total distance to the means:





- Two stages each iteration:
 - Update assignments: fix means c, change assignments a
 - Update means: fix assignments a, change means c
- Each step cannot increase phi



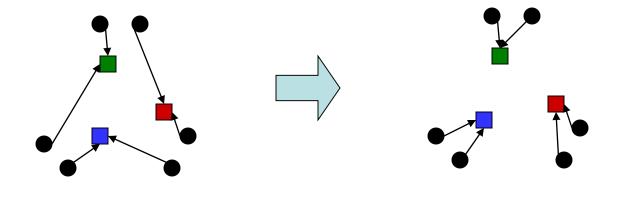
Phase I: Update Assignments

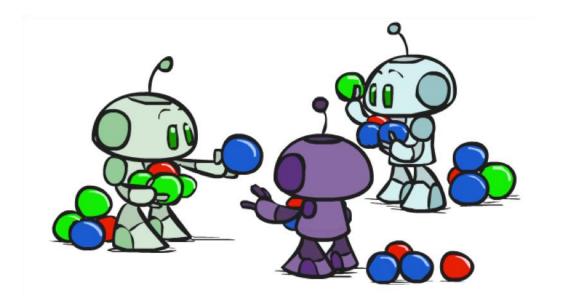
For each point, re-assign to closest mean:

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

Cannot increase total distance phi!

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



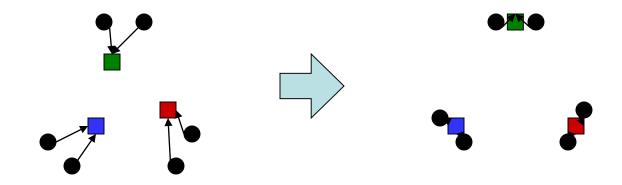


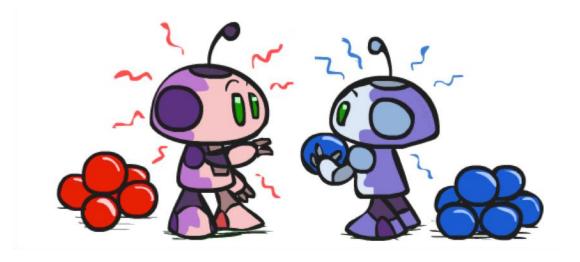
Phase II: Update Means

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

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

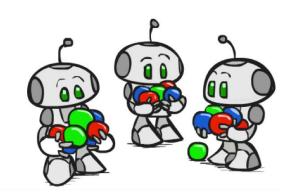
- Also cannot increase total distance
 - Fun fact: the point y with minimum squared Euclidean distance to a set of points {x} is their mean

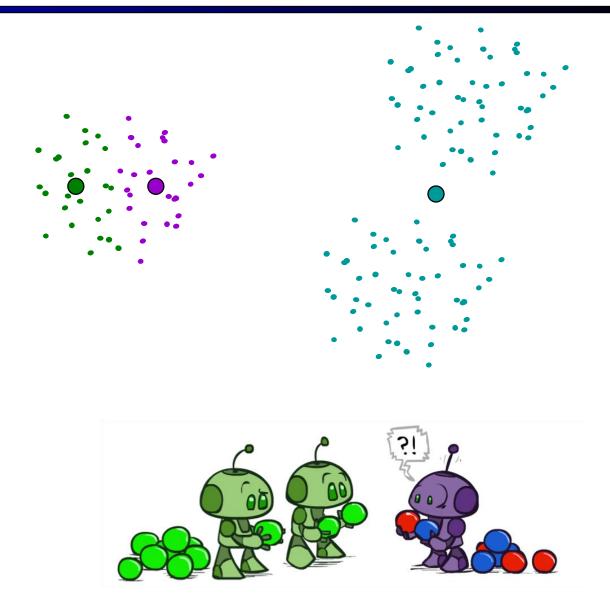




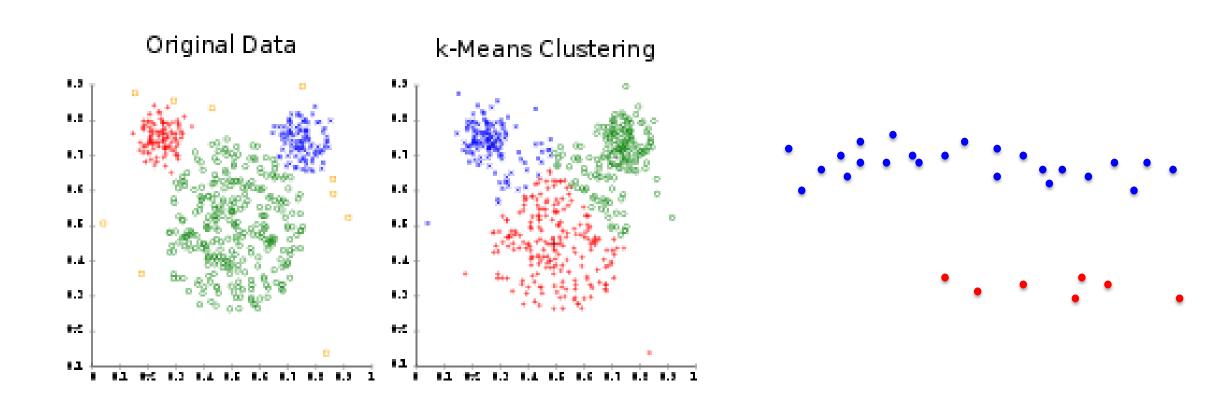
Initialization

- K-means is non-deterministic
 - Requires initial means
 - It does matter what you pick!
 - What can go wrong?
 - Local optima





Inductive Bias



Circular Clusters

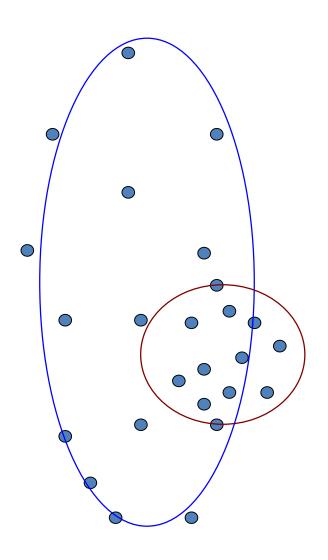
Equally Sized Clusters

Expectation-Maximization (EM)





Problems with k-means



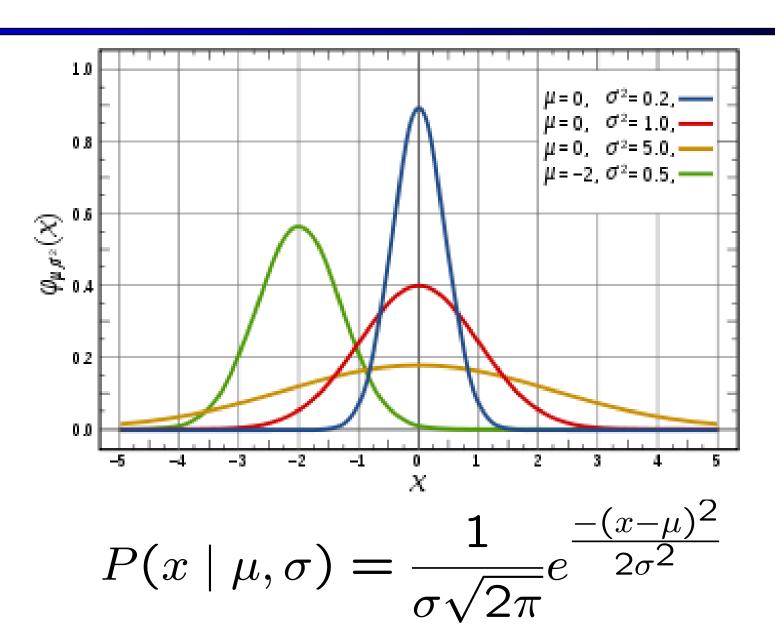
- Assigning data to closest centers
 - But some clusters may be "wider" than others
 - Distances can be deceiving!
- Hard Assignments
 - But clusters may overlap

Probabilistic Clustering

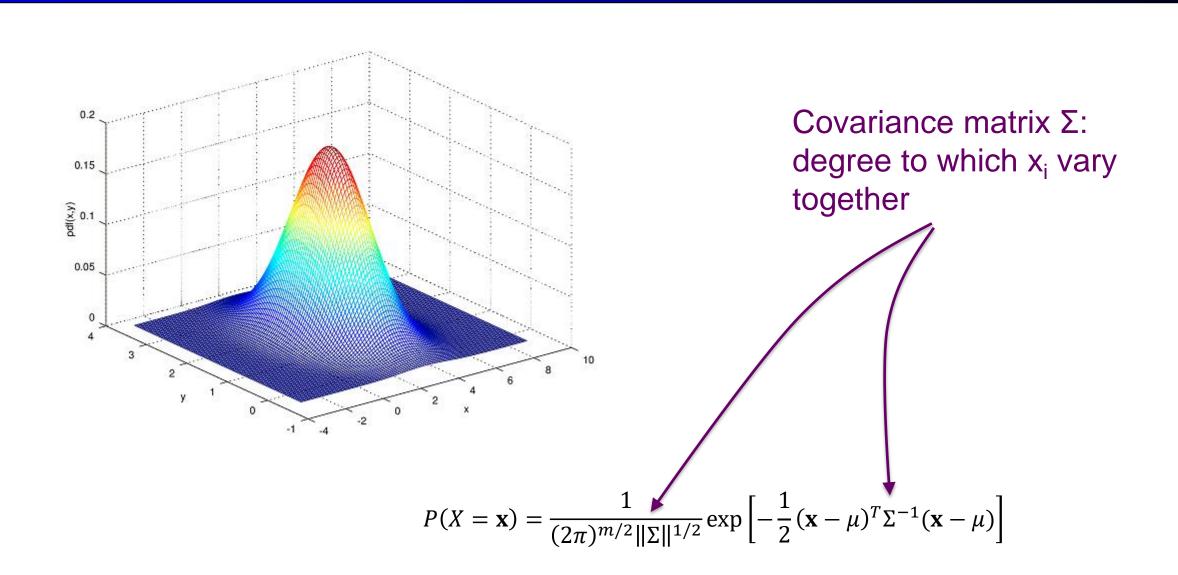
- Try a probabilistic model!
 - allows overlaps, clusters of different sizes/shapes, etc.

- Gaussian mixture model (GMM)
 - also called Mixture of Gaussians

Review: Gaussians

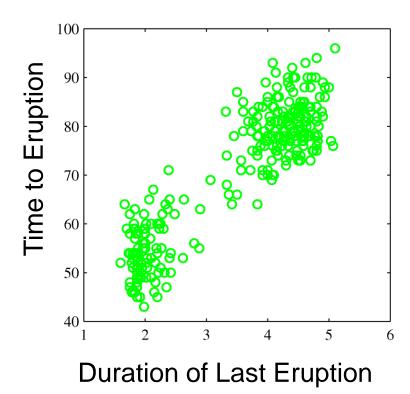


Multivariate Gaussians



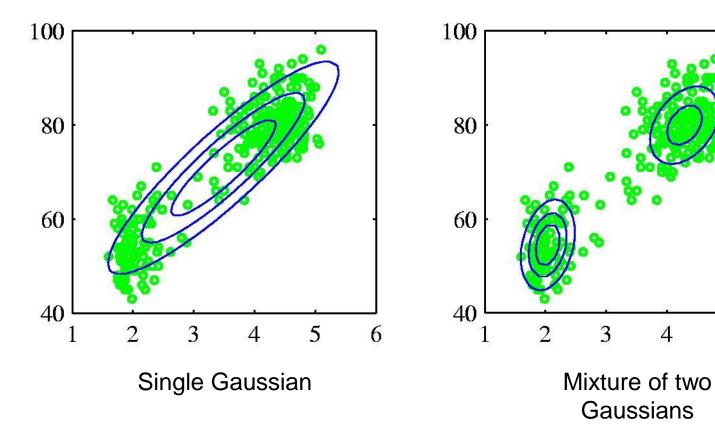
Mixtures of Gaussians

Old Faithful Data Set



Mixtures of Gaussians

Old Faithful Data Set

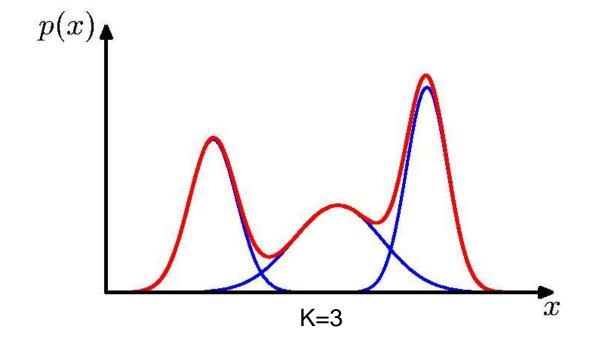


Mixtures of Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|oldsymbol{\mu}_k, oldsymbol{\Sigma}_k)$$
 Component

Mixing coefficient

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^K \pi_k = 1$$

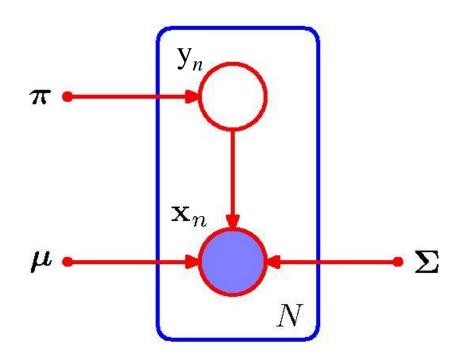


Gaussian mixture model

- P(Y): Distribution over k components (clusters)
- P(X|Y): Each component generates data from a **multivariate Gaussian** with mean μ_i and covariance matrix Σ_i

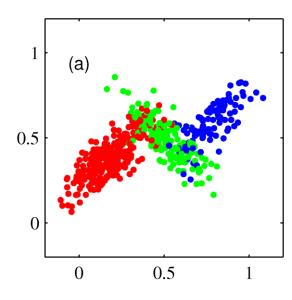
Each data point is sampled from a generative process:

- 1. Choose component i with probability P(y=i)
- 2. Generate data point from N(μ_i , Σ_i)



Supervised learning for GMM

- We observe both the data points and their labels (generated from which Gaussian components)
- How do we estimate parameters of GMM?



Supervised learning for GMM

- We observe both the data points and their labels (generated from which Gaussian components)
- How do we estimate parameters of GMM?
- Objective: maximize the likelihood

$$\prod_{j} P(y_{j} = i, \mathbf{x}_{j}) = \prod_{j} N(\mu_{i}, \Sigma_{i}) P(y_{j} = i)$$

- Closed form solution:
 - for component i, suppose we have n data points with label i

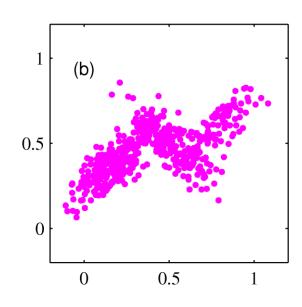
$$\mu_i = \frac{1}{n} \sum_{j=1}^n x_j \qquad \qquad \Sigma_i = \frac{1}{n} \sum_{j=1}^n \left(\mathbf{x}_j - \mu_i \right) \left(\mathbf{x}_j - \mu_i \right)^T$$

Unsupervised learning for GMM

- In clustering, we don't know the labels Y!
- Maximize marginal likelihood:

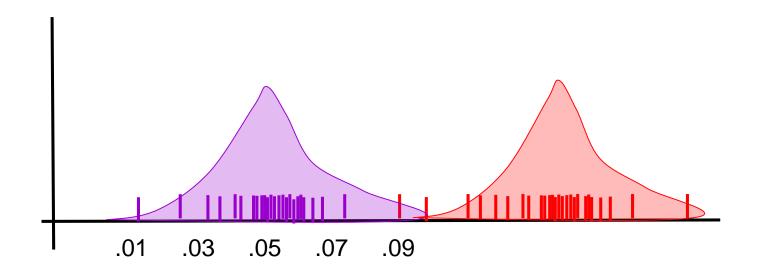
$$\prod_{j} P(\mathbf{x}_{j}) = \prod_{j} \sum_{i} P(y_{j} = i, \mathbf{x}_{j}) = \prod_{j} \sum_{i} N(\mu_{i}, \Sigma_{i}) P(y_{j} = i)$$

- How do we optimize it?
 - No closed form solution

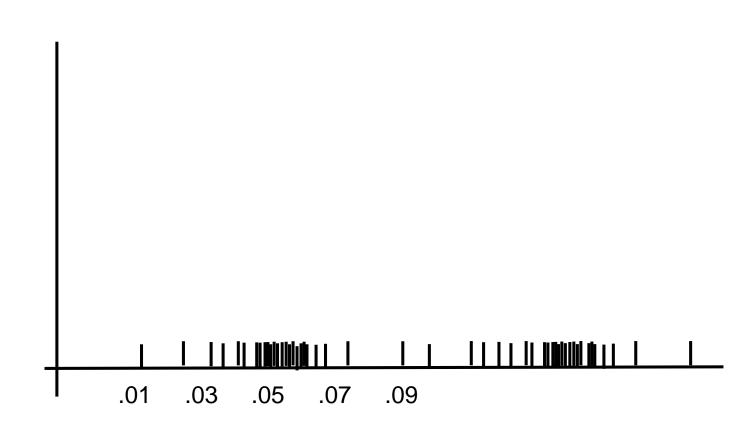


Simplest Example

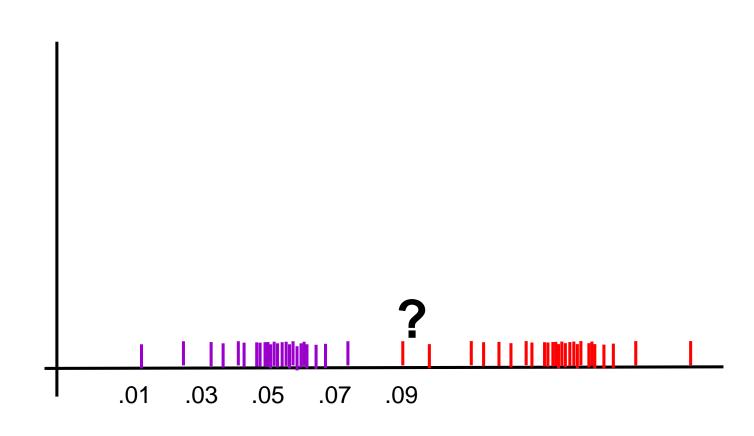
Mixture of two distributions



Input Looks Like

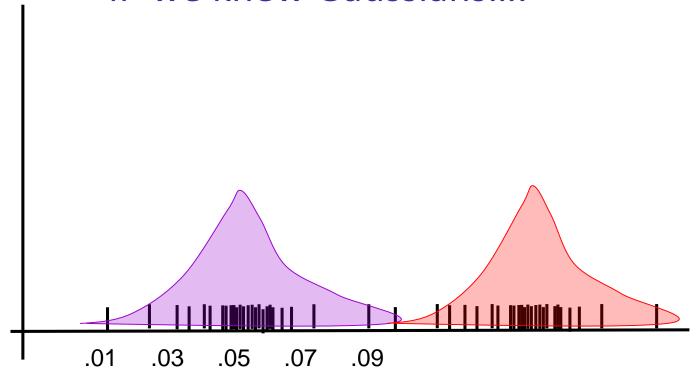


We Want to Predict

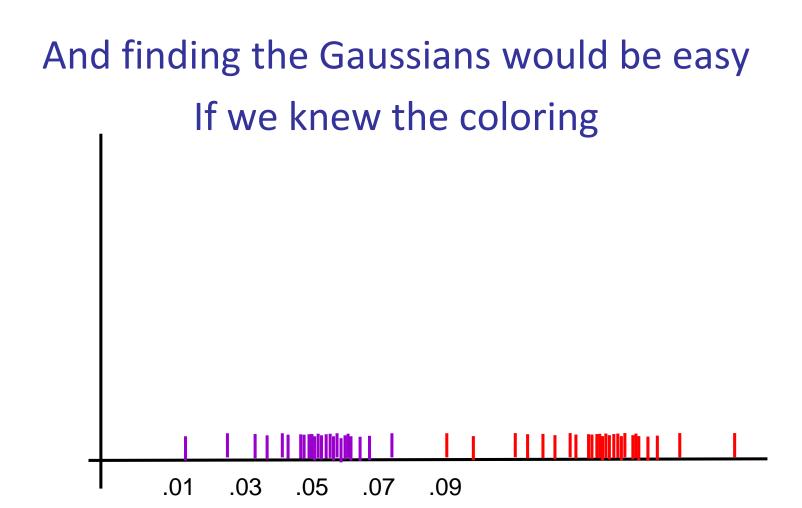


Chicken & Egg

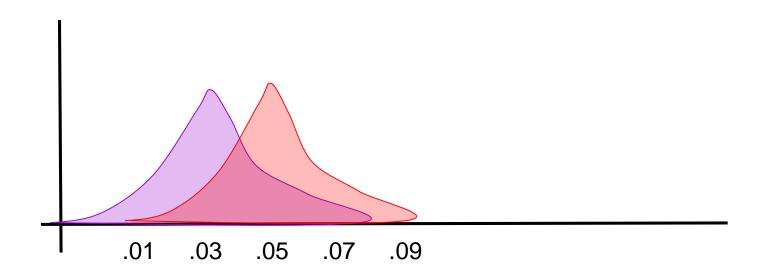
Note that coloring instances would be easy if we knew Gaussians....



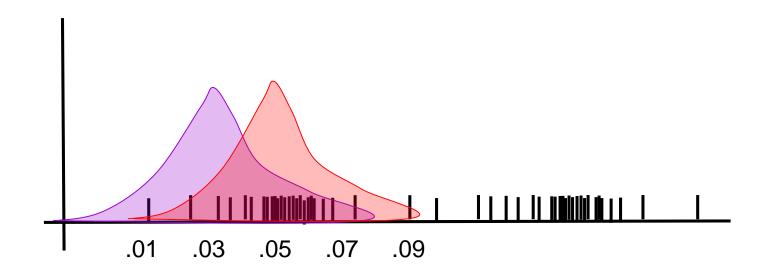
Chicken & Egg



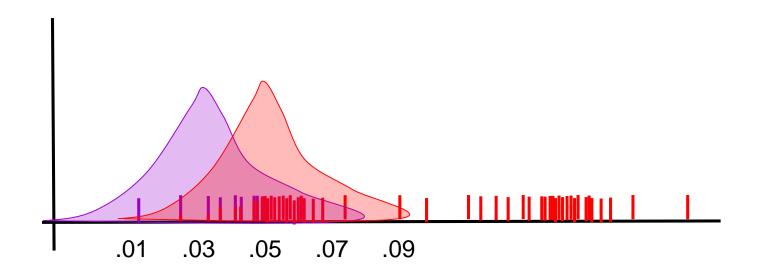
- Pretend we do know the parameters
 - Initialize randomly



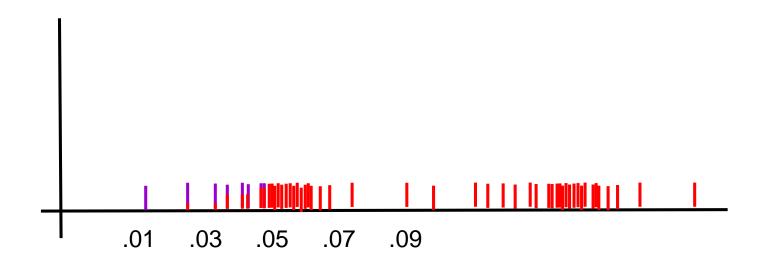
 [E step] Compute probability of each instance having each possible label



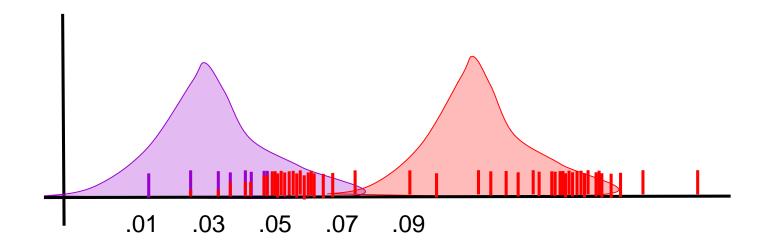
 [E step] Compute probability of each instance having each possible label



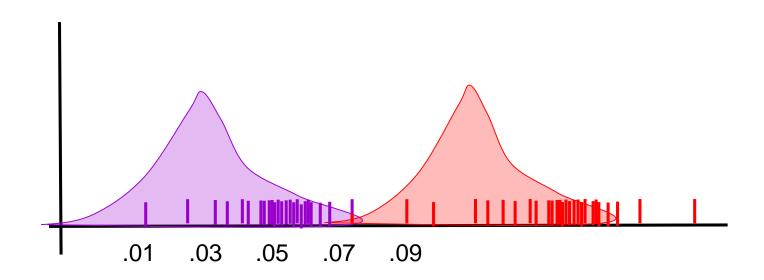
- [E step] Compute probability of each instance having each possible label
- [M step] Treating each instance as fractionally having both labels,
 compute the new parameter values



- [E step] Compute probability of each instance having each possible label
- [M step] Treating each instance as fractionally having both labels,
 compute the new parameter values

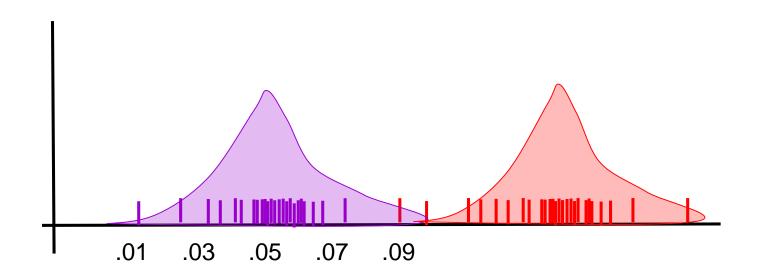


Repeat E-step



- Repeat E-step
- Repeat M-step

... until convergence



- Pick K random cluster models (Gaussians)
- Alternate:
 - Assign data instances proportionately to different models
 - Revise each cluster model based on its (proportionately) assigned points
- Stop when no changes

EM for GMM

Iterate: On the t'th iteration let our estimates be

$$\theta^{(t)} = \{ \mu_1^{(t)}, \mu_2^{(t)} \dots \mu_k^{(t)}, \sum_{i=1}^{t} (i), \sum_{i=1}^{t} (i), \sum_{i=1}^{t} (i), p_1^{(t)}, p_2^{(t)} \dots p_k^{(t)} \}$$

 $p_i^{(t)}$ is shorthand for estimate of P(y=i) on t'th iteration

E-step

Compute label distribution of each data point

$$P(y = i | x_j, \theta^{(t)}) \propto p_i^{(t)} P(x_j | \mu_i^{(t)}, \Sigma_i^{(t)})$$

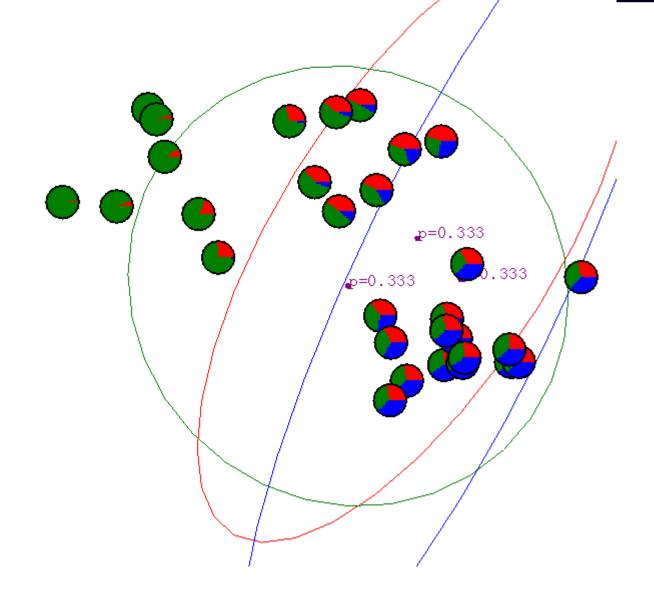
Just evaluate a Gaussian at x_j

M-step

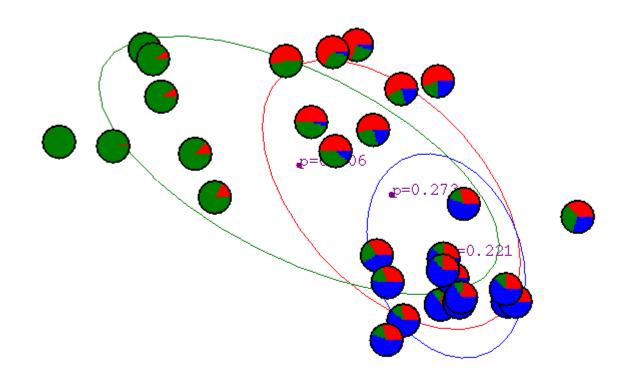
Compute weighted MLE of parameters given label distributions

$$\mu_{i}^{(t+1)} = \frac{\sum_{j} P\left(y = i \middle| x_{j}, \theta^{(t)}\right) x_{j}}{\sum_{j} P\left(y = i \middle| x_{j}, \theta^{(t)}\right)} \qquad \Sigma_{i}^{(t+1)} = \frac{\sum_{j} P\left(y = i \middle| x_{j}, \theta^{(t)}\right) \left[x_{j} - \mu_{i}^{(t+1)} \left[x_{j} - \mu_{i}^{(t+1)}\right]^{T}\right]}{\sum_{j} P\left(y = i \middle| x_{j}, \theta^{(t)}\right)} \qquad p_{i}^{(t+1)} = \frac{\sum_{j} P\left(y = i \middle| x_{j}, \theta^{(t)}\right)}{m} \qquad m = \text{\#training examples}$$

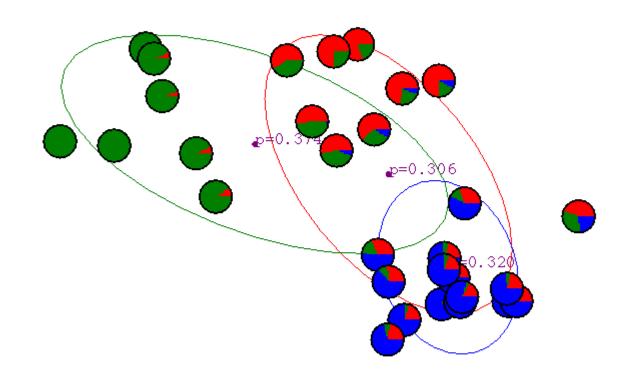
Gaussian Mixture Example: Start



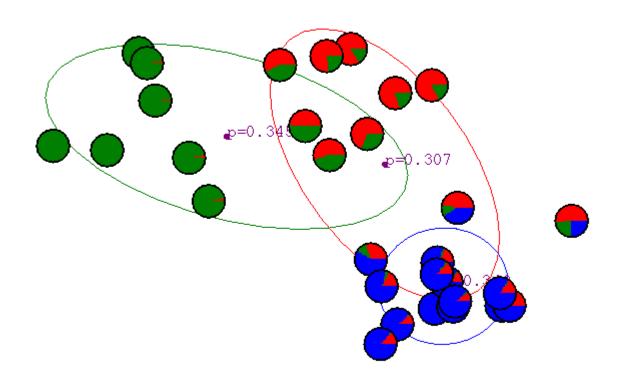
After first iteration



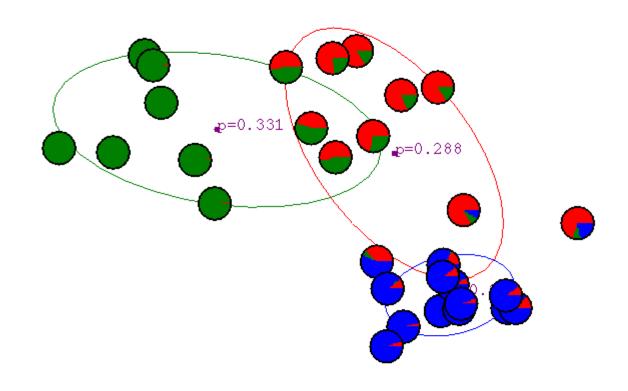
After 2nd iteration



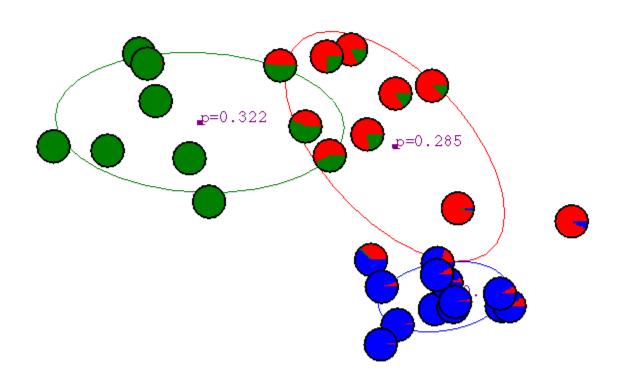
After 3rd iteration



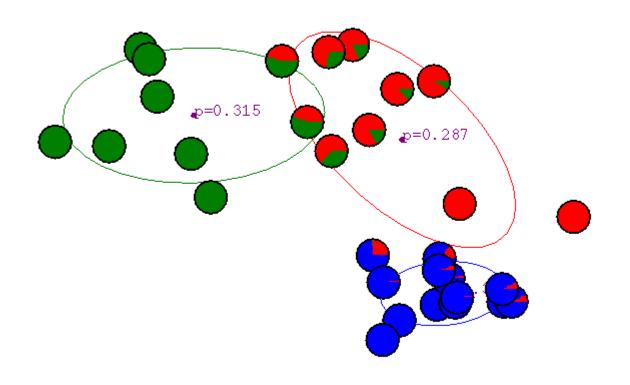
After 4th iteration



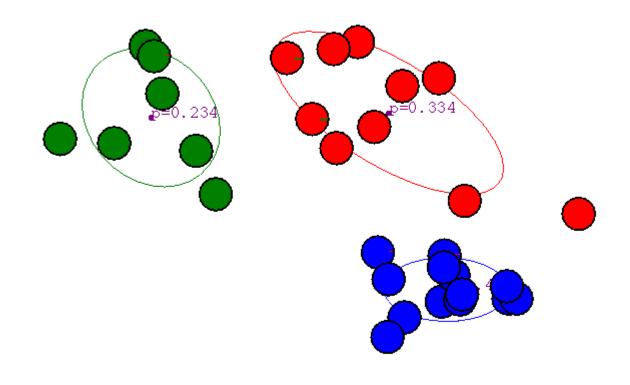
After 5th iteration



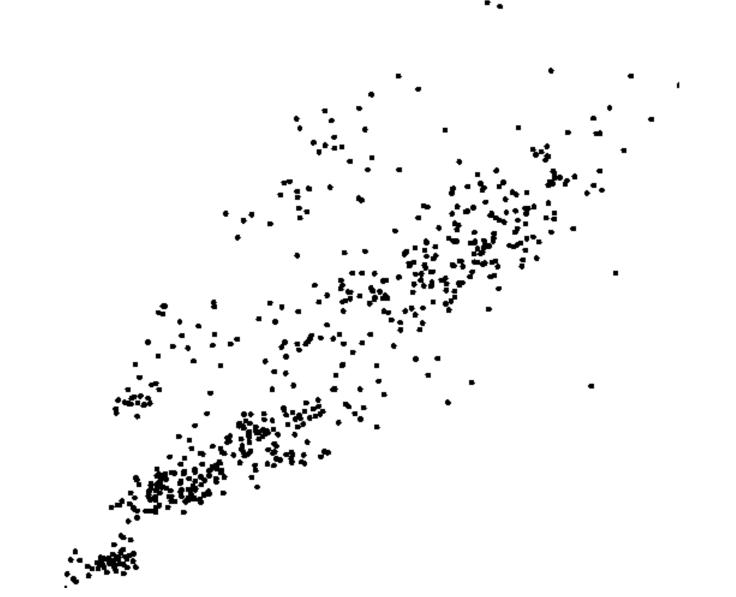
After 6th iteration



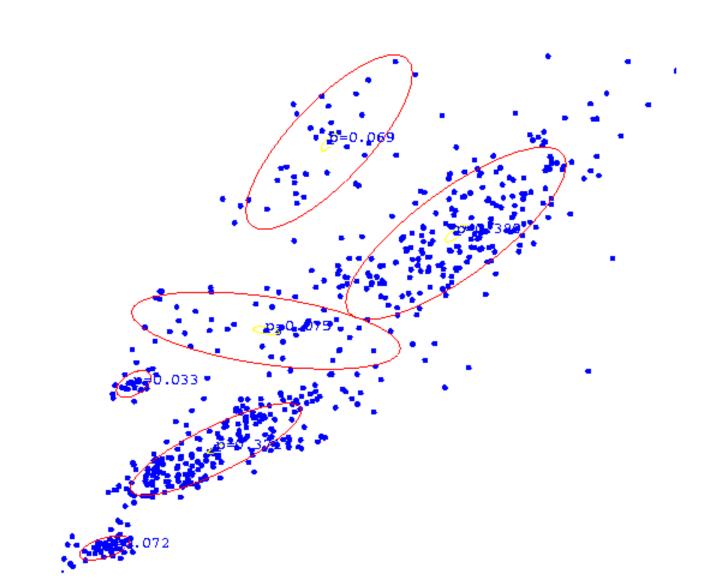
After 20th iteration



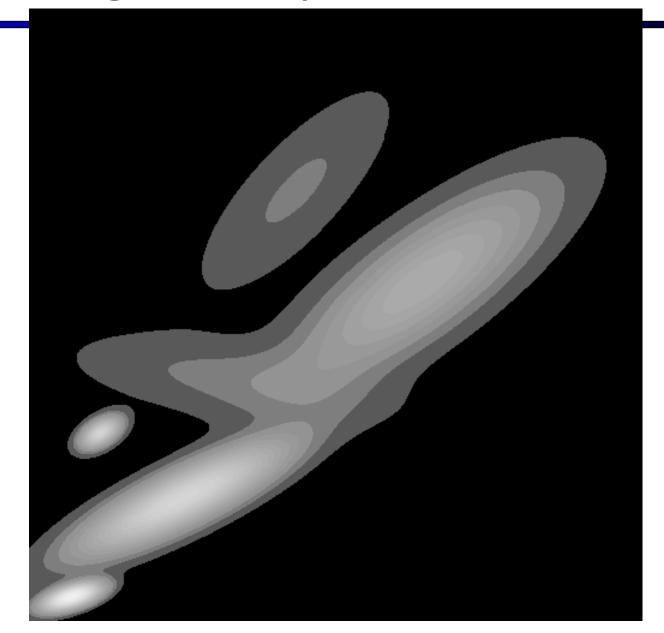
Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator



EM and K-means

- EM degrades to k-means if we assume
 - All the Gaussians are spherical and have identical weights and covariances
 - i.e., the only parameters are the means
 - The label distributions computed at E-step are point-estimations
 - i.e., hard-assignments of data points to Gaussians
 - Alternatively, assume the variances are close to zero

EM in General

- Can be used to learn any model with hidden variables (missing data)
- Alternate:
 - Compute distributions over hidden variables based on current parameter values
 - Compute new parameter values to maximize expected log likelihood based on distributions over hidden variables
- Stop when no changes

EM for HMMs

- [E step] Compute the distributions of hidden states given each training instance
 - Based on the forward and backward algorithms with fixed model parameters
- [M step] Update the parameters to maximize expected log likelihood based on distributions over hidden states
 - Simply count the transitions and emissions and then normalize
 - Counts are fractions (based on the probabilities of hidden states)
- Known as Baum–Welch algorithm

Math Behind EM

• EM is coordinate ascent on $F(\theta, Q)$

$$\ell(\theta:\mathcal{D}) \geq F(\theta,Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z},\mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$
 Jensen's inequality

- E-step fixes θ and optimizes Q
- M-step fixes Q and optimizes θ
- EM Converges
 - E-step doesn't decrease $F(\theta, Q)$
 - M-step doesn't either

Summary

Clustering

Group together similar instances

K-means

- Assign data instances to closest mean
- Assign each mean to the average of its assigned points

EM

- Assign data instances proportionately to different models
- Revise each model based on its (proportionately) assigned points