Machine Learning 10-601, 10-301

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

April 21, 2021

Today:

- Learning graphical models
  1. EM: learning from partially observed data
  2. Mixture models, clustering
  3. Structure learning

Readings:

- Bishop chapter 9-9.2 mixture models
- Kevin Murphy chapter 11.4 (optional)

Bishop: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

EM : Learning from Partially Observed Training Data

EM algorithm

S	A	K	E	H
1	0	1	1	0
0	1	?	0	1
0	1	1	0	0
0	?	0	1	0
1	1	1	1	1

- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values

EM algorithm:

- Iterate until convergence:
  - **E step**: use current Bayes net parameters θ to estimate unobserved Z values

S	A	K	E	H
1	0	1	1	0
0	1	?	0	1
0	1	1	0	0
0	?	0	1	0
1	1	1	1	1

Pr	S	A	K	E	H
1.0	1	0	1	1	0
0.6	0	1	1	0	1
0.4	0	1	0	0	1
1.0	0	1	1	0	0
0.2	0	1	0	1	0
0.8	0	0	0	1	0
1.0	1	1	1	1	1

  - **M step**: use estimated values of Z to retrain Bayes net params θ

$$\theta_{K=1|S=i,A=j} = P(K=1|S=i,A=j) = \frac{\sum_{m=1}^{M} P_\theta(s_m=i, a_m=j, k_m=1)}{\sum_{m=1}^{M} P_\theta(s_m=i, a_m=j)}$$

m^{th} training example

EM algorithm

S	A	K	E	H
1	0	1	1	0
0	1	?	0	1
0	1	1	0	0
0	?	0	1	0
1	1	1	1	1

- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values

EM algorithm:

- Iterate until convergence:
  - **E step**: use current Bayes net paran

wait – how do we compute these probabilities??

S	A	K	E	H
1	0	1	1	0
0	1	?	0	1
0	1	1	0	0
0	?	0	1	0
1	1	1	1	1

Pr	S	A	K	E	H
1.0	1	0	1	1	0
0.6	0	1	1	0	1
0.4	0	1	0	0	1
1.0	0	1	1	0	0
0.2	0	1	0	1	0
0.8	0	0	0	1	0
1.0	1	1	1	1	1

  - **M step**: use estimated values of Z to retrain Bayes net params θ

$$\theta_{K=1|S=i,A=j} = P(K=1|S=i,A=j) = \frac{\sum_{m=1}^{M} P_\theta(s_m=i, a_m=j, k_m=1)}{\sum_{m=1}^{M} P_\theta(s_m=i, a_m=j)}$$

m^{th} training example

Only One Unobserved Variable:

Parents <S, A>	P(K=1|S,A)	P(K=0|S,A)
<1, 1>	0.9	0.1
<1, 0>	0.8	0.2
<0, 1>	0.7	0.3
<0, 0>	0.1	0.9

Know

How do we calculate P(K=1 | S=s, A=a, E=e, H=h) ?

$$P(K=1|S=s,A=a,E=e,H=h) = \frac{P(S=s,A=a,K=1,E=e,H=h)}{P(S=s,A=a,E=e,H=h)}$$

$$= \frac{P(S=s,A=a,K=1,E=e,H=h)}{P(S=s,A=a,K=1,E=e,H=h)+P(S=s,A=a,K=0,E=e,H=h)}$$

where:

$$P(S=s,A=a,K=k,E=e,H=h) = P(S=s)P(A=a)P(K=k|S=s,A=a)P(E=e|K=k)P(H=1|K=k)$$

Efficient: O(2n) for n Boolean variables.

EM algorithm

S	A	K	E	H
1	0	1	1	0
0	1	?	0	1
0	1	1	0	0
0	?	0	1	0
1	1	1	1	1

- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values

EM algorithm:

- Iterate until convergence:
  - **E step**: use current Bayes net parameters θ to estimate unobserved Z values

S	A	K	E	H
1	0	1	1	0
0	1	?	0	1
0	1	1	0	0
0	?	0	1	0
1	1	1	1	1

Pr	S	A	K	E	H
1.0	1	0	1	1	0
0.6	0	1	1	0	1
0.4	0	1	0	0	1
1.0	0	1	1	0	0
0.2	0	1	0	1	0
0.8	0	0	0	1	0
1.0	1	1	1	1	1

  - **M step**: use estimated values of Z to retrain Bayes net params θ

$$\theta_{K=1|S=i,A=j} = P(K=1|S=i,A=j) = \frac{\sum_{m=1}^{M} P_\theta(s_m=i, a_m=j, k_m=1)}{\sum_{m=1}^{M} P_\theta(s_m=i, a_m=j)}$$

m^{th} training example

EM Algorithm - Precisely

EM is a general procedure for learning from partly observed data

Given observed training feature values X, unobserved Z, from all examples

Iterate until convergence:

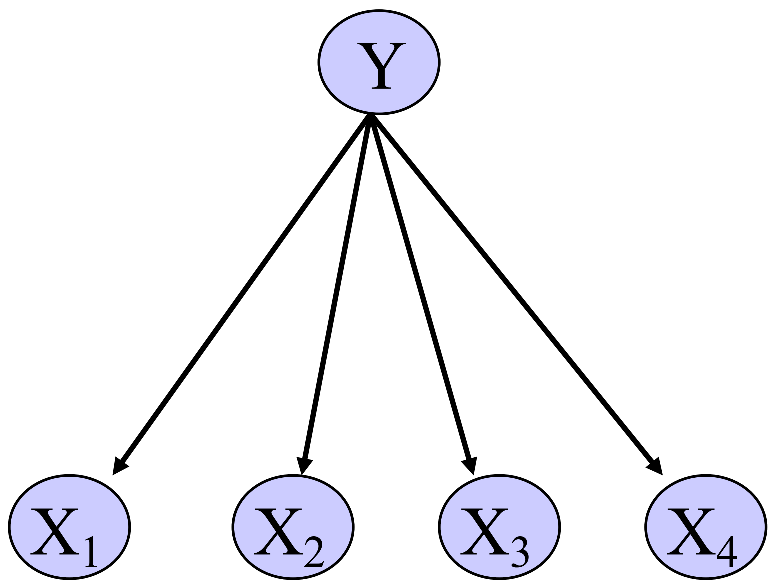
- E Step: Use X and current θ to calculate $P(Z|X,\theta)$
- M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X,Z|\theta')]$$

Guaranteed to find θ that is local maximum of $E_{P(Z|X,\theta)}[\log P(X,Z|\theta')]$

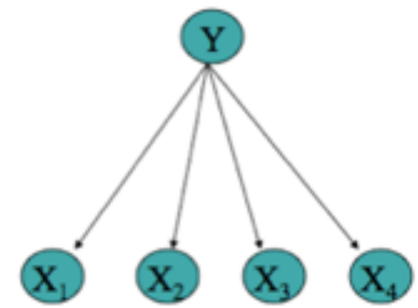
Using Unlabeled Data to Help Train Naïve Bayes Classifier

Learn P(Y|X)



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

EM for semi-supervised Naïve Bayes



Given observed set X, unobserved set Y of values
(only missing values are labels Y for some examples)

E step: Calculate for each training example, k

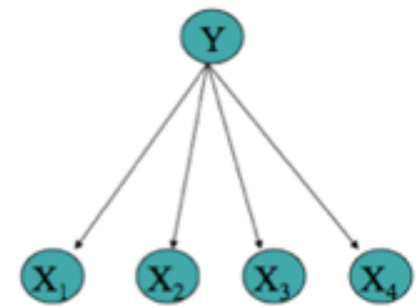
the expected value of each unobserved value of variable Y

$$E_{P(Y|X_1 \ldots X_N)}[y(k)] = P(y(k)=1|x_1(k), \ldots x_N(k); \theta) = \frac{P(y(k)=1)\prod_i P(x_i(k)|y(k)=1)}{\sum_{j=0}^{1} P(y(k)=j)\prod_i P(x_i(k)|y(k)=j)}$$

k^{th} training example

M step: Calculate estimates similar to MLE, but replacing each count by its expected count

EM for semi-supervised Naïve Bayes



Given observed set X, unobserved set Y of values
(only missing values are labels Y for some examples)

E step: Calculate for each training example, k

the expected value of each unobserved value of variable Y

$$E_{P(Y|X_1 \dots X_N)}[y(k)] = P(y(k) = 1|x_1(k), \dots x_N(k); \theta) = \frac{P(y(k)=1) \prod_i P(x_i(k)|y(k)=1)}{\sum_{j=0}^{1} P(y(k)=j) \prod_i P(x_i(k)|y(k)=j)}$$

k^{th} training example

M step: Calcu
replac

Why is expected value of Boolean-valued Y just P(Y=1)?

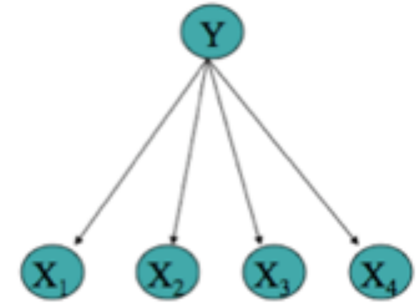
Answer: the definition of expected value:

$$E_{P(Y)}[Y] = \sum_{y \in \{0,1\}} P(Y = y) \; y$$

$$= [P(Y=1) \; 1] + [P(Y=0) \; 0]$$

$$= \; P(Y=1)$$

EM for semi-supervised Naïve Bayes



Given observed set X, unobserved set Y of values
(only missing values are labels Y for some examples)

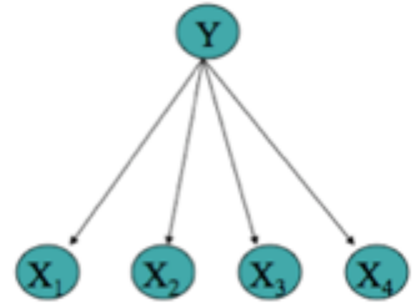
E step: Calculate for each training example, k

the expected value of each unobserved value of variable Y

$$E_{P(Y|X_1 \ldots X_N)}[y(k)] = P(y(k) = 1 | x_1(k), \ldots x_N(k); \theta) = \frac{P(y(k) = 1) \prod_i P(x_i(k) | y(k) = 1)}{\sum_{j=0}^{1} P(y(k) = j) \prod_i P(x_i(k) | y(k) = j)}$$

k^{th} training example

M step: Calculate estimates similar to MLE, but
replacing each count by its <u>expected count</u>



Given observed set X, unobserved set Y of values (only missing values are labels Y for some examples)

E step: Calculate for each training example, k

the expected value of each unobserved variable Y

$$E_{P(Y|X_1 \ldots X_N)}[y(k)] = P(y(k) = 1|x_1(k), \ldots x_N(k); \theta) = \frac{P(y(k)=1) \prod_i P(x_i(k)|y(k)=1)}{\sum_{j=0}^{1} P(y(k)=j) \prod_i P(x_i(k)|y(k)=j)}$$

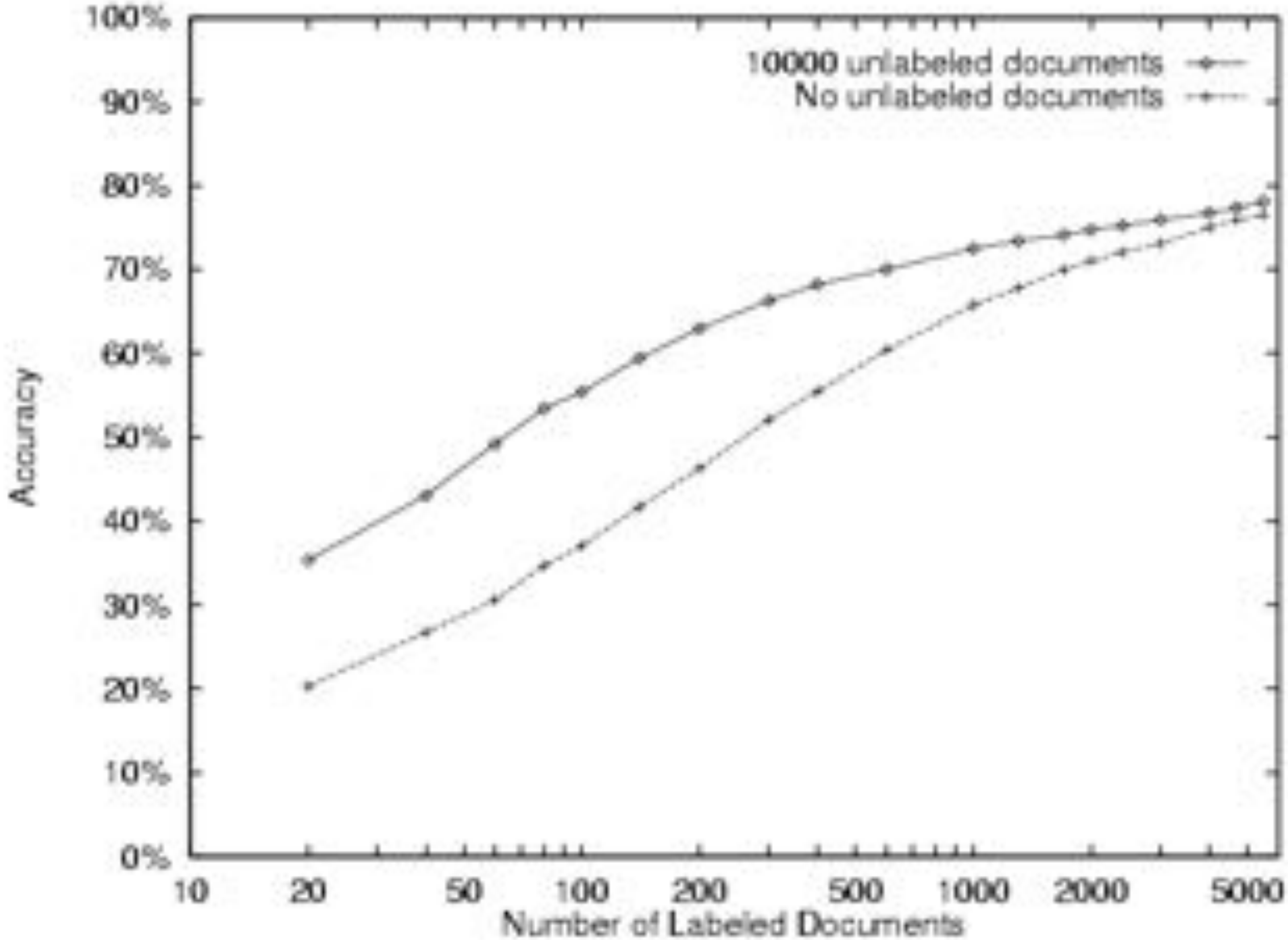
k^{th} training example

M step: Calculate estimates similar to MLE, but replacing each count by its <u>expected count</u>

$$\theta_{ij|m} = \hat{P}(X_i = j|Y = m) = \frac{\sum_k P(y(k) = m|x_1(k) \ldots x_N(k)) \; \delta(x_i(k) = j)}{\sum_k P(y(k) = m|x_1(k) \ldots x_N(k))}$$

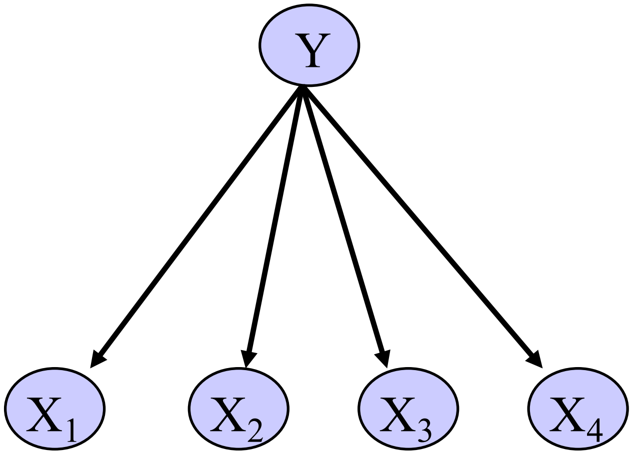
MLE would be: $\hat{P}(X_i = j|Y = m) = \frac{\sum_k \delta((y(k) = m) \wedge (x_i(k) = j))}{\sum_k \delta(y(k) = m)}$

20 Newsgroups



Question: What if our data provides <u>no</u> Y labels,
but we believe P(Y,X1,X2,X3,X4) is
defined by this Naïve Bayes net structure?

Can we still use EM to learn P(Y,X1,X2,X3,X4)?



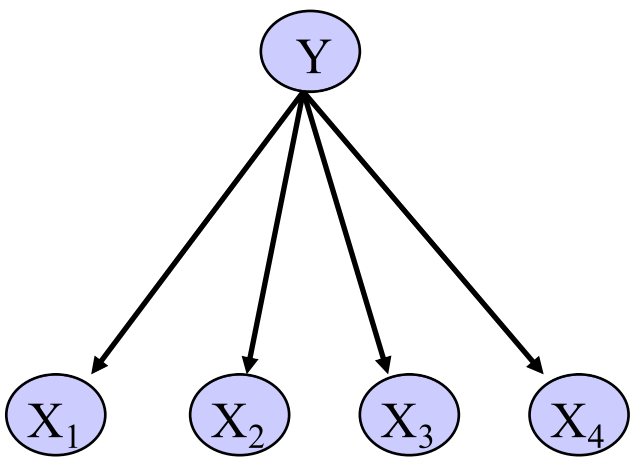
Y	X1	X2	X3	X4
?	0	0	1	1
?	0	1	0	0
?	0	0	1	0
?	0	1	1	0
?	0	1	0	1

Question: What if our data provides <u>no</u> Y labels, but we believe P(Y,X1,X2,X3,X4) is defined by this Naïve Bayes net structure?

→ Unsupervised clustering

→ Y is the unobserved indicator of which cluster each X belongs to. P(Y=1|X), P(Y=0|X) indicate the prob. that X belongs to each cluster

→ Or, if we want to consider more clusters, we define Y to have more values (i.e., Y in {0,1,2,…,N})



Unobserved cluster label to be learned

Y	X1	X2	X3	X4
?	0	0	1	1
?	0	1	0	0
?	0	0	1	0
?	0	1	1	0
?	0	1	0	1

Question: What if our data provides no Y labels, but we believe P(Y,X1,X2,X3,X4) is defined by this Naïve Bayes net structure?

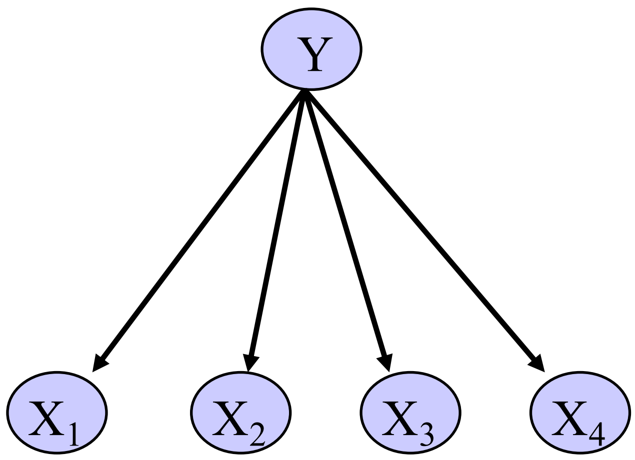
→ Unsupervised clustering

→ Y is the unobserved indicator of which cluster each X belongs to. P(Y=1|X), P(Y=0|X) indicate the prob. that X belongs to each cluster

Suppose we assume P(X1,X2,X3,X4) is a mixture of two distributions (two clusters). Then:

P(X1,X2,X3,X4) =
P(Y=1) P(X1,X2,X3,X4 | Y=1)
+ P(Y=0) P(X1,X2,X3,X4 | Y=0)

Unobserved cluster label to be learned



Y	X1	X2	X3	X4
?	0	0	1	1
?	0	1	0	0
?	0	0	1	0
?	0	1	1	0
?	0	1	0	1

Question: What if our data provides no Y labels, but we believe P(Y,X1,X2,X3,X4) is defined by this Naïve Bayes net structure?

→ Unsupervised clustering

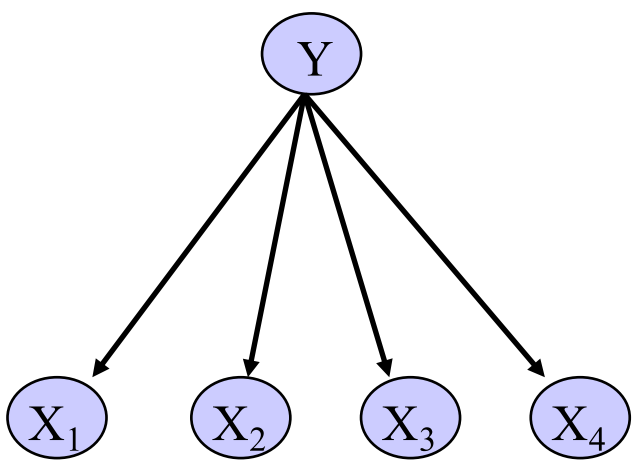
→ Y is the unobserved indicator of which cluster each X belongs to. P(Y=1|X), P(Y=0|X) indicate the prob. that X belongs to each cluster

Suppose we assume P(X1,X2,X3,X4) is a mixture of two distributions (two clusters). Then:

$$P(X1,X2,X3,X4) = P(Y=1)\, P(X1,X2,X3,X4 \mid Y=1) + P(Y=0)\, P(X1,X2,X3,X4 \mid Y=0)$$

This form is called a "mixture distribution"

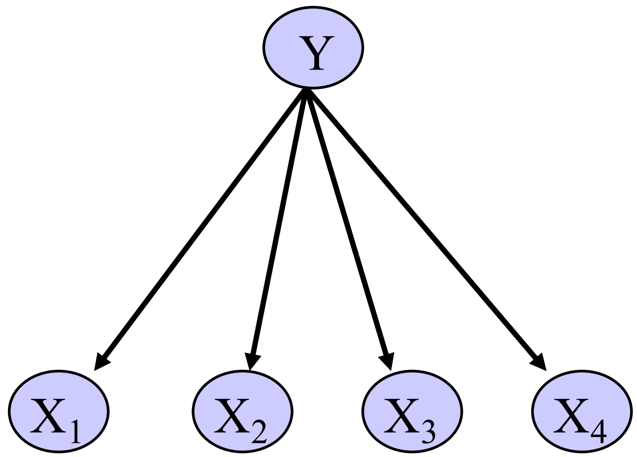
Unobserved cluster label to be learned



Y	X1	X2	X3	X4
?	0	0	1	1
?	0	1	0	0
?	0	0	1	0
?	0	1	1	0
?	0	1	0	1

Question: What if our data provides no Y labels, but we believe P(Y,X1,X2,X3,X4) is defined by this Naïve Bayes net structure?

→ Unsupervised clustering : EM



Y	X1	X2	X3	X4
?	0	0	1	1
?	0	1	0	0
?	0	0	1	0
?	0	1	1	0
?	0	1	0	1

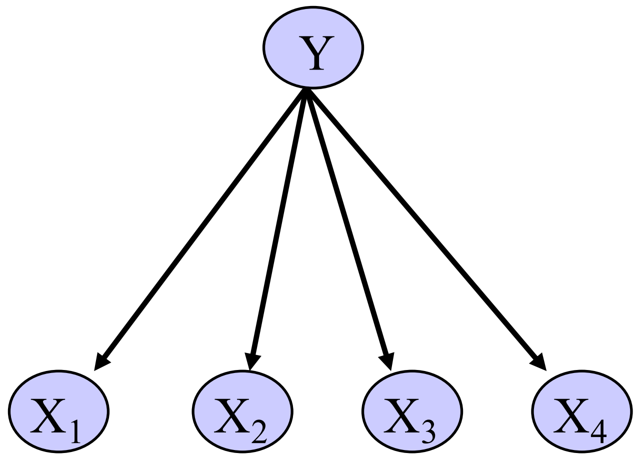
EM →

Learned probabilistic cluster label

Pr	Y	X1	X2	X3	X4
0.8	1	0	0	1	1
0.2	0	0	0	1	1
0.3	1	0	1	0	0
0.7	0	0	1	0	0
0.4	1	0	0	1	0
0.6	0	0	0	1	0
0.7	1	0	1	1	0
0.3	0	0	1	1	0
0.6	1	0	1	0	1
0.4	0	0	1	0	1

Question: What if our data provides <u>no</u> Y labels,
but we believe P(Y,X1,X2,X3,X4) is
defined by this Naïve Bayes net structure?

→ Unsupervised clustering : EM

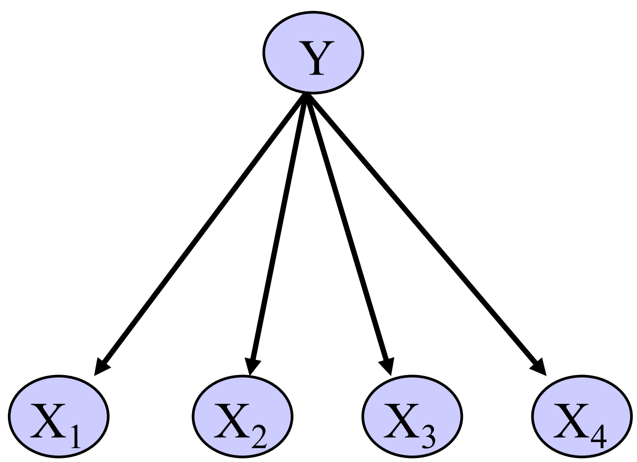


Y	X1	X2	X3	X4
?	0.1	7.2	3.1	1.4
?	9.9	2.1	5.0	0.2
?	8.0	0.7	5.1	0.9
?	1.1	6.2	2.9	2.1
?	1.4	8.3	2.7	1.8

← What if real-valued X_i's?

Question: What if our data provides <u>no</u> Y labels,
but we believe P(Y,X1,X2,X3,X4) is
defined by this Naïve Bayes net structure?

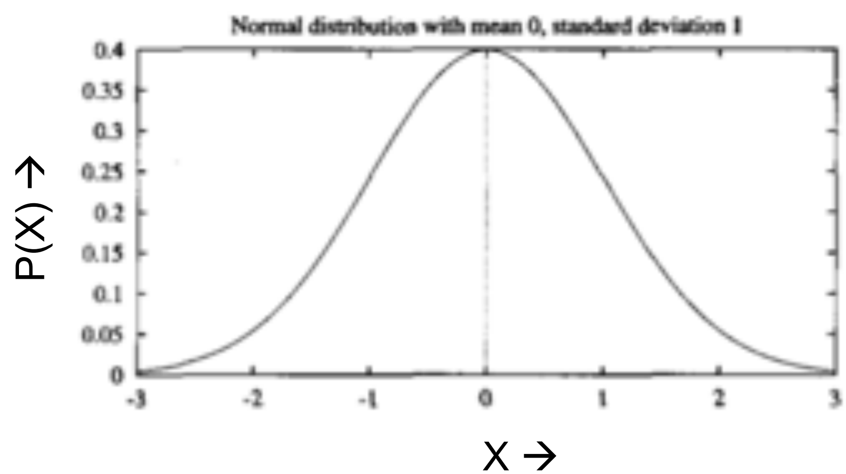
→ Unsupervised clustering : EM



Y	X1	X2	X3	X4
?	0.1	7.2	3.1	1.4
?	9.9	2.1	5.0	0.2
?	8.0	0.7	5.1	0.9
?	1.1	6.2	2.9	2.1
?	1.4	8.3	2.7	1.8

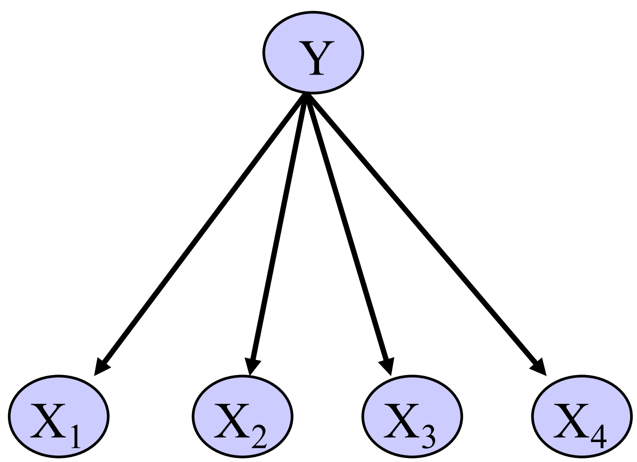
What if real-valued X_i's?
Need different form of $P(X_i|Y)$
e.g., Gaussian

$$P(X_i = x|Y = y) = \frac{1}{\sqrt{2\pi\sigma_{iy}^2}} \exp\left(-\frac{1}{2\sigma_{iy}^2}(x - \mu_{iy})^2\right)$$



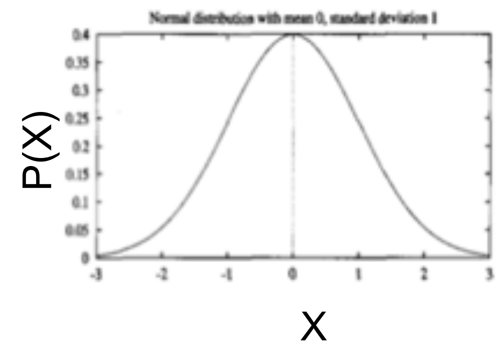
Question: What if our data provides <u>no</u> Y labels,
but we believe P(Y,X1,X2,X3,X4) is
defined by this Naïve Bayes net structure?

→ Unsupervised clustering : EM



Y	X1	X2	X3	X4
?	0.1	7.2	3.1	1.4
?	9.9	2.1	5.0	0.2
?	8.0	0.7	5.1	0.9
?	1.1	6.2	2.9	2.1
?	1.4	8.3	2.7	1.8

EM
→



Pr	Y	X1	X2	X3	X4
0.8	1	0.1	7.2	3.1	1.4
0.2	0	0.1	7.2	3.1	1.4
0.3	1	9.9	2.1	5.0	0.2
0.7	0	9.9	2.1	5.0	0.2
0.4	1	8.0	0.7	5.1	0.9
0.6	0	8.0	0.7	5.1	0.9
0.7	1	1.1	6.2	2.9	2.1
0.3	0	1.1	6.2	2.9	2.1
0.6	1	1.4	8.3	2.7	1.8
0.4	0	1.4	8.3	2.7	1.8

EM for Mixture of Gaussians Clustering

Let's simplify to make this easier:

1. assume $X=\langle X_1 \ldots X_n \rangle$, and the X_i are conditionally independent given Z. (the Naïve Bayes assumption).

$$P(X|Z=j) = \prod_i N(X_i|\mu_{ji}, \sigma_{ji})$$

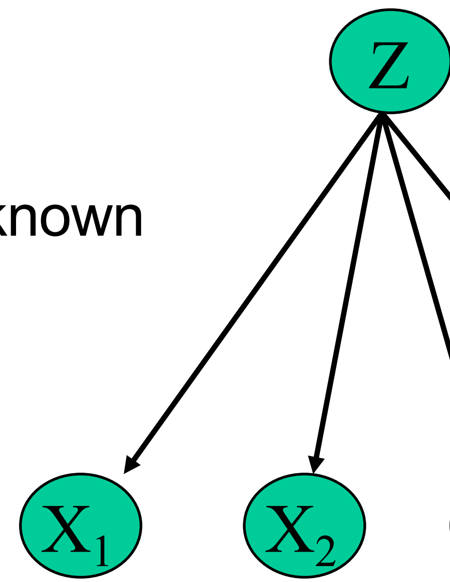
2. assume only 2 clusters (Z in {0,1}), and $\forall i, j, \sigma_{ji} = \sigma$

$$P(\mathbf{X}) = \sum_{j=1}^{2} P(Z=j|\pi) \prod_i N(x_i|\mu_{ji}, \sigma)$$

3. Assume σ known, $\pi_1 \ldots \pi_K,$ $\mu_{1i} \ldots \mu_{Ki}$ unknown

Observed: $X=\langle X_1 \ldots X_n \rangle$

Unobserved: Z

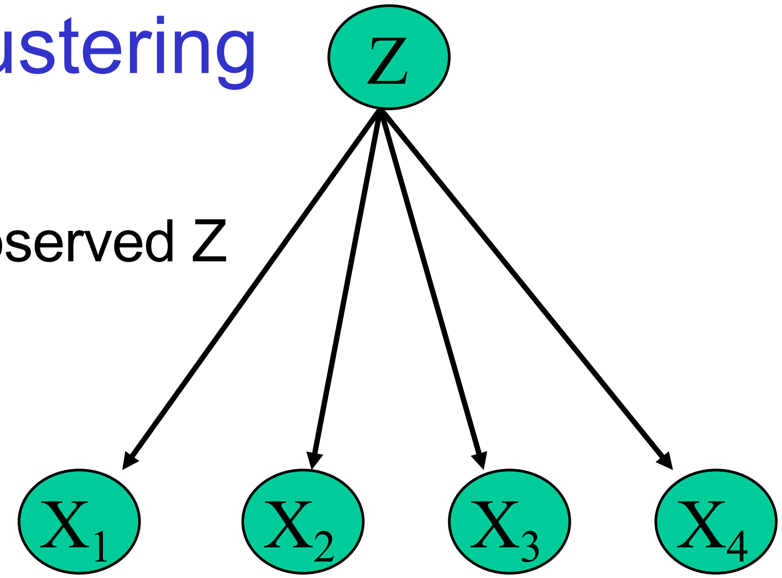


$$\theta = \langle \pi, \mu_{ji} \rangle$$

$$\pi = P(Z=1)$$

$$\mu_{i,j} = \mu_{X_i|Z=j}$$

EM for Gaussian mixture model clustering



Given observed real-valued variables X_i, unobserved Z

where $\theta = \langle \pi, \mu_{ji} \rangle$

$$\pi \equiv P(Z = 1)$$

$$\mu_{ji} \equiv \text{mean of Gaussian for } P(X_i | Z = j)$$

Iterate until convergence:

- E Step: For each observed example X(n), calculate P(Z(n) | X(n), θ)

$$P(z(n) = k \mid x(n), \theta) = \frac{[\prod_i N(x_i(n) | \mu_{k,i}, \sigma)] \quad (\pi^k (1 - \pi)^{(1-k)})}{\sum_{j=0}^{1} [\prod_i N(x_i(n) | \mu_{j,i}, \sigma)] \quad (\pi^j (1 - \pi)^{(1-j)}))}$$

- M Step: Update

$$\pi \leftarrow \frac{1}{N} \sum_{n=1}^{N} E[z(n)] \qquad \mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j | x(n), \theta) \quad x_i(n)}{\sum_{n=1}^{N} P(z(n) = j | x(n), \theta)}$$

Observed data X_1, X_2, unknown cluster assignment Z

Goal: Learn mixture distribution, interpreting Z as cluster label

Learn $P(X_1, X_2 | \theta) =$

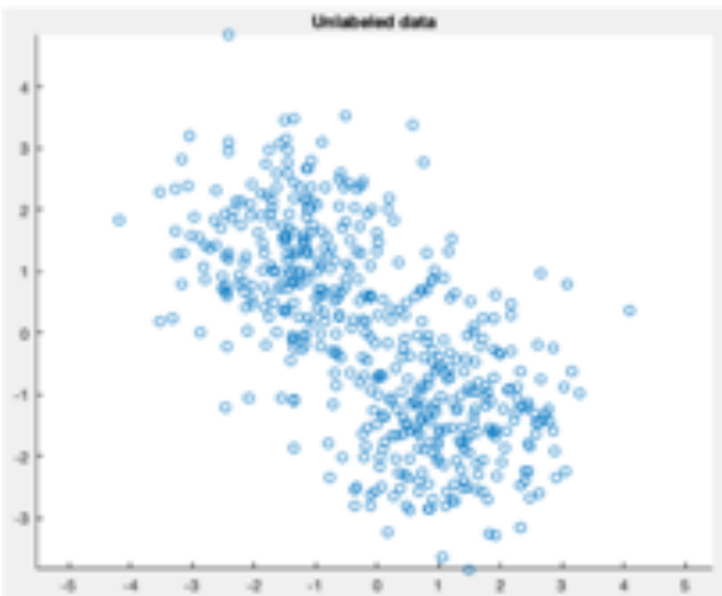
$$P(Z=1|\theta)\, P(X_1, X_2| Z=1,\theta) + P(Z=0|\theta)\, P(X_1, X_2| Z=0,\theta)$$

Z	X1	X2
?	0.9	-1.3
?	-1.5	1.2
?	-0.4	-0.6
…	…	…

EM Algorithm

1. Choose any initial θ
2. Iterate until convergence:
   - E Step: Use X and current θ to calculate P(Z|X,θ)
   - M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$$



Observed data X_1, X_2, unknown cluster assignment Z

Goal: Learn mixture distribution, interpreting Z as cluster label

Learn $P(X_1, X_2 | \theta) =$

$$P(Z=1 | \theta)\, P(X_1, X_2 | Z=1,\theta)$$
$$+\ P(Z=0 | \theta)\, P(X_1, X_2 | Z=0,\theta)$$

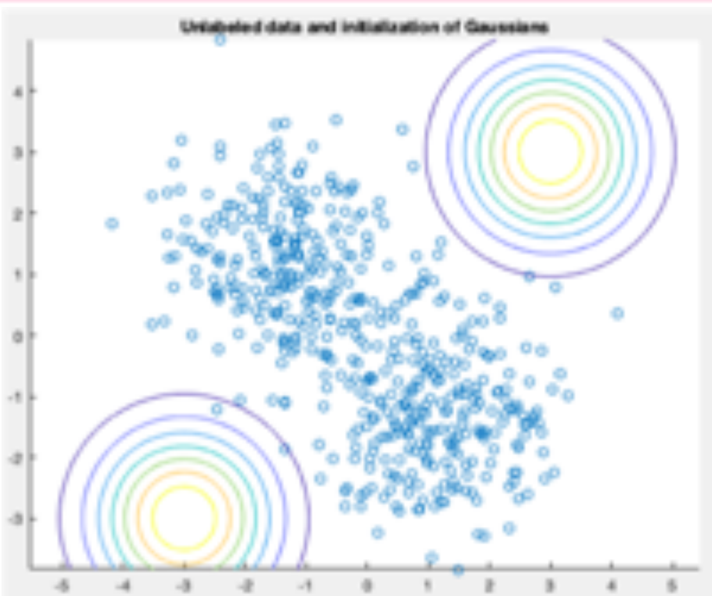
Z	X1	X2
?	0.9	-1.3
?	-1.5	1.2
?	-0.4	-0.6
…	…	…

EM Algorithm

1. Choose any initial θ

2. Iterate until convergence:

- E Step: Use X and current θ to calculate $P(Z|X,\theta)$
- M Step: Replace current θ by
  $$\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$$



Observed data X_1, X_2, unknown cluster assignment Z

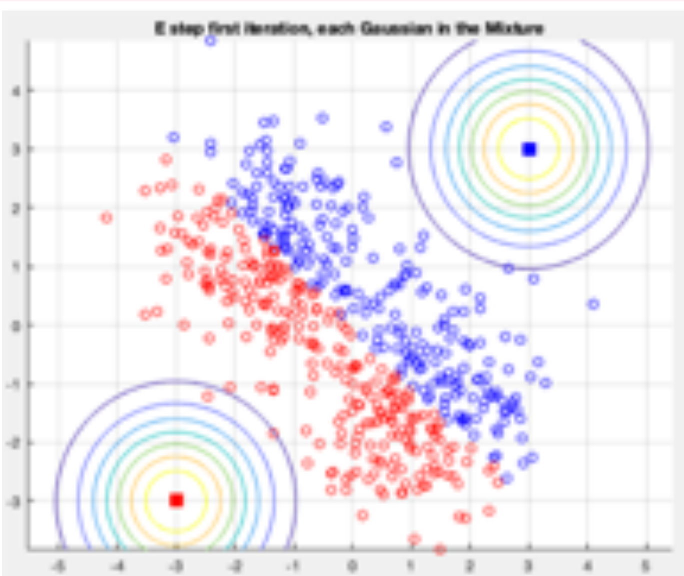
Goal: Learn mixture distribution, interpreting Z as cluster label

Learn $P(X_1, X_2 | \theta) =$

$$P(Z=1|\theta)\, P(X_1, X_2 | Z=1,\theta) + P(Z=0|\theta)\, P(X_1, X_2 | Z=0,\theta)$$

EM Algorithm

1. Choose any initial θ
2. Iterate until convergence:
   - E Step: Use X and current θ to calculate P(Z|X,θ)
   - M Step: Replace current θ by
     $$\theta \leftarrow \arg\max_{\theta'} \; E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$$



Z	X1	X2
?	0.9	-1.3
?	-1.5	1.2
?	-0.4	-0.6
…	…	…

E-Step

Probability	Z	X1	X2
0.8	1	0.9	-1.3
0.2	0	0.9	-1.3
0.3	1	-1.5	1.2
0.7	0	-1.5	1.2
0.6	1	-0.4	-0.6
0.4	0	-0.4	-0.6
…	…	…	…

Observed data X_1, X_2, unknown cluster assignment Z

Goal: Learn mixture distribution, interpreting Z as cluster label

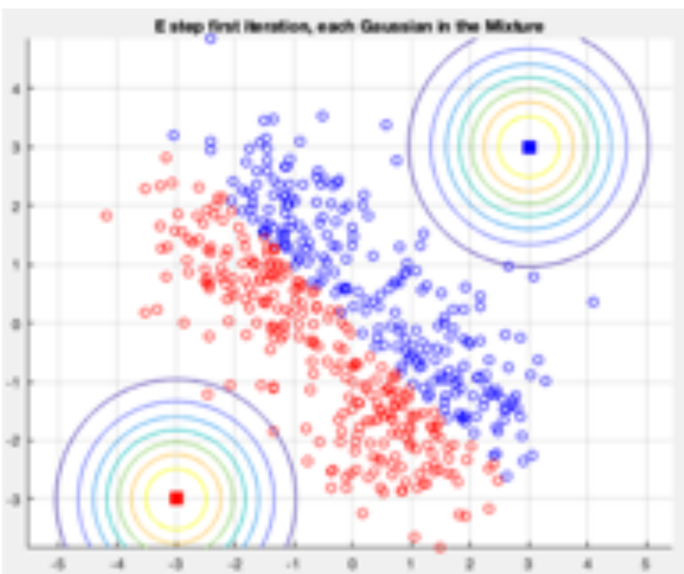
Learn $P(X_1, X_2 | \theta) =$
$P(Z=1 | \theta)\, P(X_1, X_2 | Z=1,\theta)$
$+\ P(Z=0 | \theta)\, P(X_1, X_2 | Z=0,\theta)$

Probability	Z	X1	X2
0.8	1	0.9	-1.3
0.2	0	0.9	-1.3
0.3	1	-1.5	1.2
0.7	0	-1.5	1.2
0.6	1	-0.4	-0.6
0.4	0	-0.4	-0.6
…	…	…	…

EM Algorithm

1. Choose any initial θ
2. Iterate until convergence:
   - E Step: Use X and current θ to calculate $P(Z|X,\theta)$
   - M Step: Replace current θ by
     $$\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$$

M-Step $\theta_{Z=1}$



$$\theta_{Z=1} \equiv P(Z=1) \leftarrow \frac{1}{N}\sum_{n=1}^{N} P_{Z|X,\theta}(Z_n = 1)$$

$$= \frac{0.8 + 0.3 + 0.6 + \ldots}{3 + \ldots}$$

Observed data X_1, X_2, unknown cluster assignment Z

Goal: Learn mixture distribution, interpreting Z as cluster label

Learn $P(X_1, X_2 | \theta) =$

$P(Z=1 | \theta)\, P(X_1, X_2 | Z=1, \theta)$

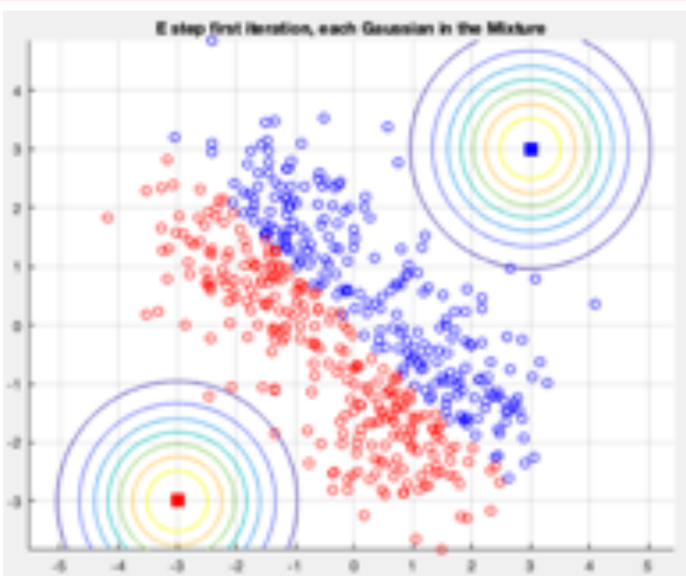
$+\ P(Z=0 | \theta)\, P(X_1, X_2 | Z=0, \theta)$

Probability	Z	X1	X2
0.8	1	0.9	-1.3
0.2	0	0.9	-1.3
0.3	1	-1.5	1.2
0.7	0	-1.5	1.2
0.6	1	-0.4	-0.6
0.4	0	-0.4	-0.6
…	…	…	…

EM Algorithm

1. Choose any initial θ
2. Iterate until convergence:
   - E Step: Use X and current θ to calculate $P(Z|X,\theta)$
   - M Step: Replace current θ by $\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

M-Step $\theta_{Z=1}$



$$\theta_{Z=1} \equiv P(Z=1) \leftarrow \frac{1}{N}\sum_{n=1}^{N} P_{Z|X,\theta}(Z_n = 1)$$

$$= \frac{0.8 + 0.3 + 0.6 + \ldots}{3 + \ldots}$$

note if Z observed, we would have

$$\theta_{Z=1} \equiv P(Z=1) \leftarrow \frac{1}{N}\sum_{n=1}^{N} Z$$

Observed data X_1, X_2, unknown cluster assignment Z

Goal: Learn mixture distribution, interpreting Z as cluster label

Learn $P(X_1, X_2 | \theta) =$

$$P(Z=1|\theta)\, P(X_1, X_2 | Z=1,\theta)$$
$$+\ P(Z=0|\theta)\, P(X_1, X_2 | Z=0,\theta)$$

Probability	Z	X1	X2
0.8	1	0.9	-1.3
0.2	0	0.9	-1.3
0.3	1	-1.5	1.2
0.7	0	-1.5	1.2
0.6	1	-0.4	-0.6
0.4	0	-0.4	-0.6
…	…	…	…

EM Algorithm

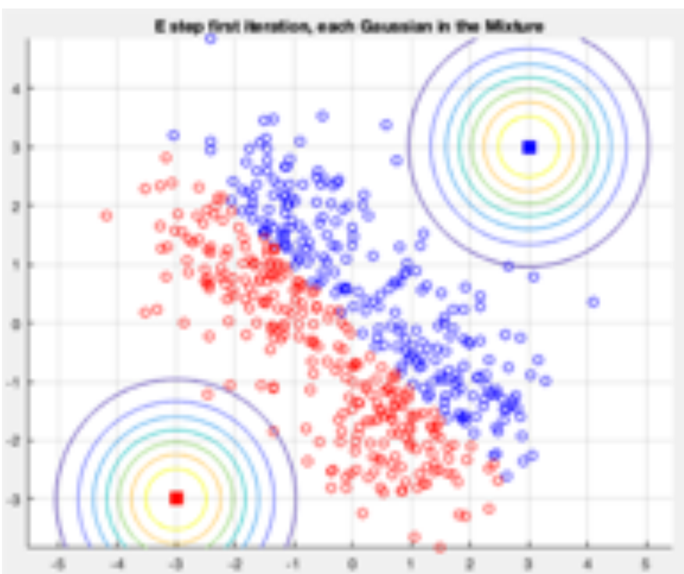
1. Choose any initial θ
2. Iterate until convergence:
   - E Step: Use X and current θ to calculate $P(Z|X,\theta)$
   - M Step: Replace current θ by
     $$\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$$

M-Step: $\mu_{X_i|Z=j}$

$$\mu_{X_i|Z=j} \leftarrow \frac{\sum_{n=1}^{N} P(Z_n = j) X_{i,n}}{\sum_{n=1}^{N} P(Z_n = j)}$$

e.g.,

$$\mu_{X_2|Z=1} = \frac{0.8(-1.3) + 0.3(1.2) + 0.6(-0.6) + \ldots}{0.8 + 0.3 + 0.6 + \ldots}$$



Observed data X_1, X_2, unknown cluster assignment Z

Goal: Learn mixture distribution, interpreting Z as cluster label

Learn $P(X_1, X_2 | \theta) =$

$$P(Z=1|\theta)\, P(X_1, X_2 | Z=1,\theta) + P(Z=0|\theta)\, P(X_1, X_2 | Z=0,\theta)$$

Probability	Z	X1	X2
0.8	1	0.9	-1.3
0.2	0	0.9	-1.3
0.3	1	-1.5	1.2
0.7	0	-1.5	1.2
0.6	1	-0.4	-0.6
0.4	0	-0.4	-0.6
…	…	…	…

EM Algorithm

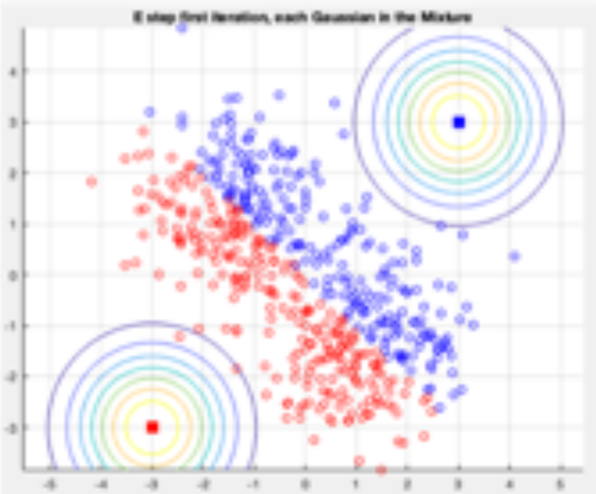
1. Choose any initial θ
2. Iterate until convergence:
   - E Step: Use X and current θ to calculate $P(Z|X,\theta)$
   - M Step: Replace current θ by
     $\theta \leftarrow \arg\max_{\theta'} E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

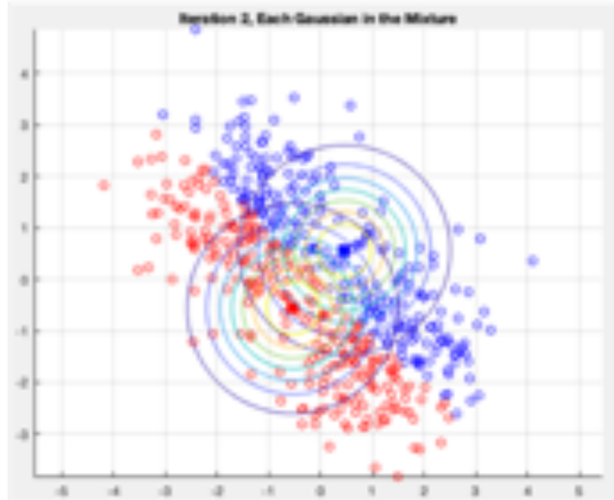
M-Step $\mu_{X_i|Z=j}$

$$\mu_{X_i|Z=j} \leftarrow \frac{\sum_{n=1}^{N} P(Z_n = j) X_{i,n}}{\sum_{n=1}^{N} P(Z_n = j)}$$

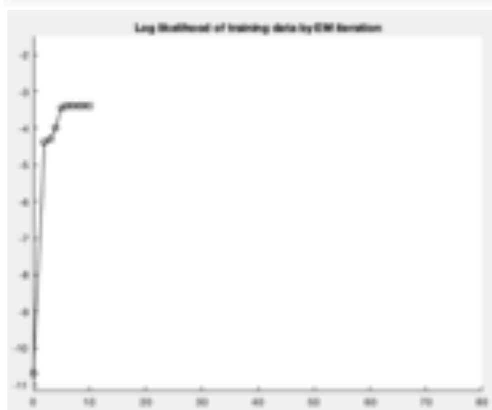
e.g.,

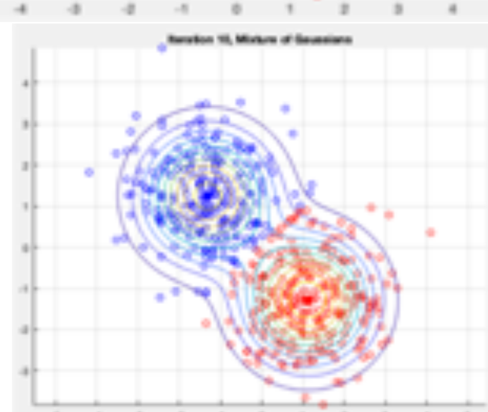
$$\mu_{X_2|Z=1} = \frac{0.8(-1.3) + 0.3(1.2) + 0.6(-0.6) + \dots}{0.8 + 0.3 + 0.6 + \dots}$$



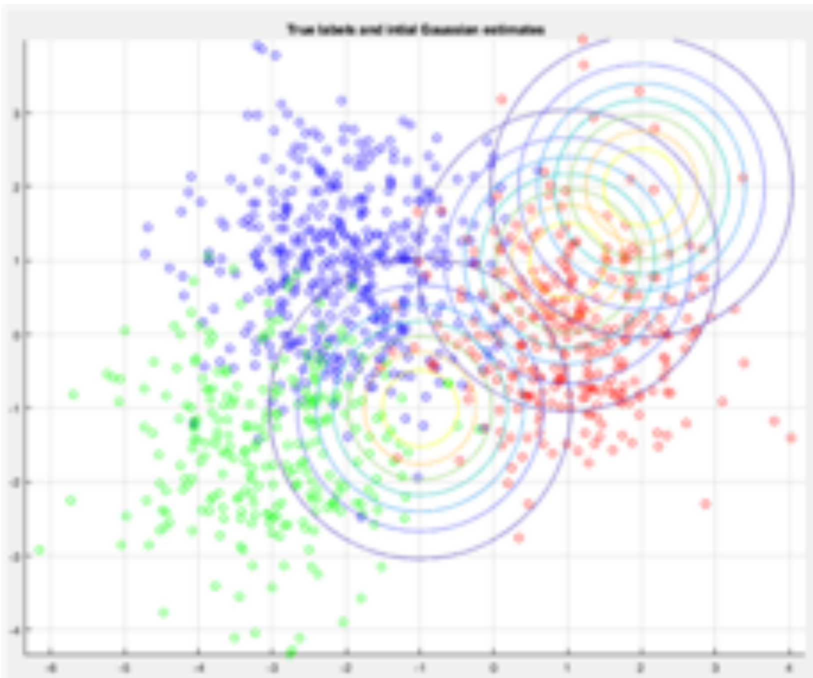


Final P(Z)=[0.4893 0.5107]

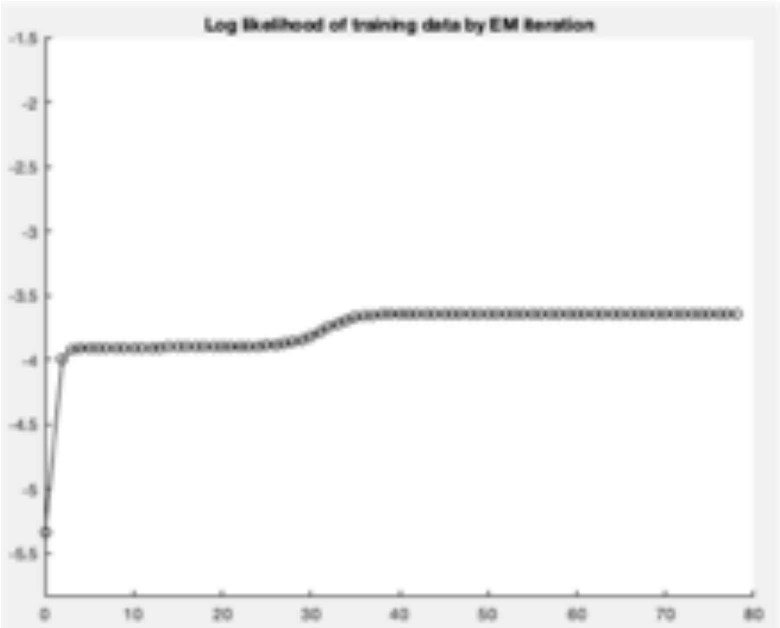




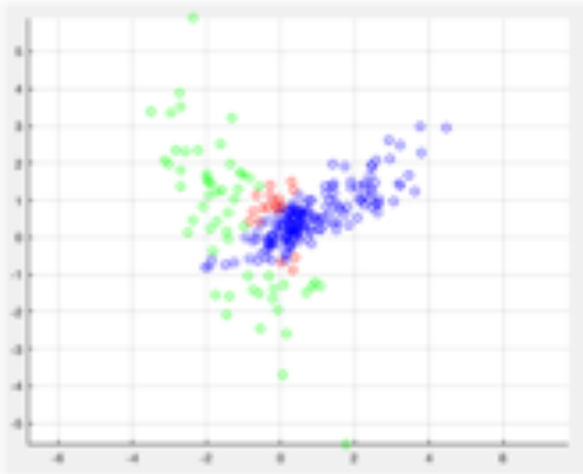
Example: Mixture of Three (Spherical) Gaussians



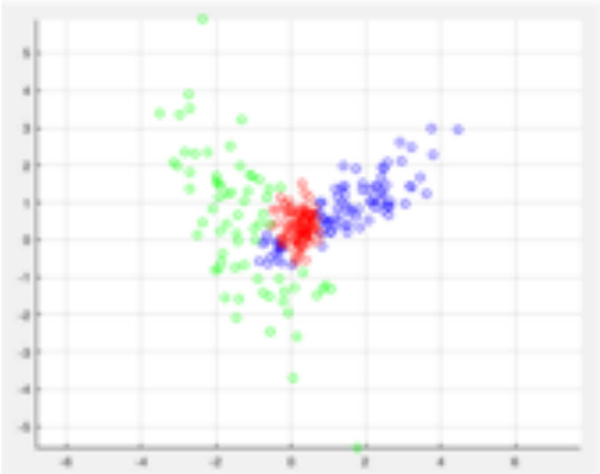
EM →



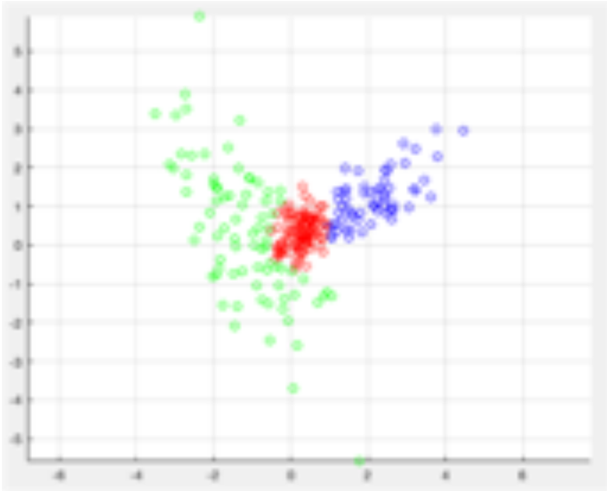
EM assuming mixture of 3 Gaussian components : no conditional indep assumptions, so non-spherical Gaussians



10 iterations

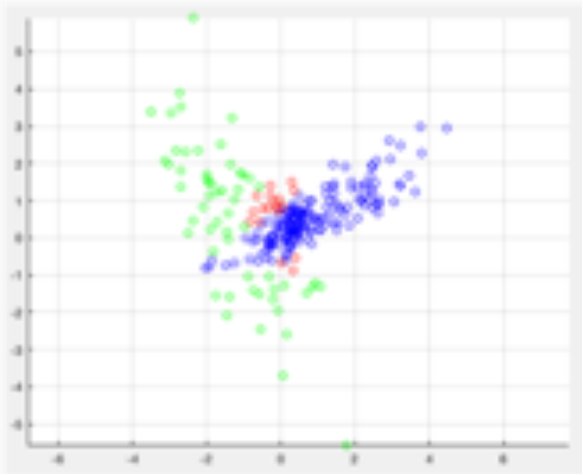


20 iterations

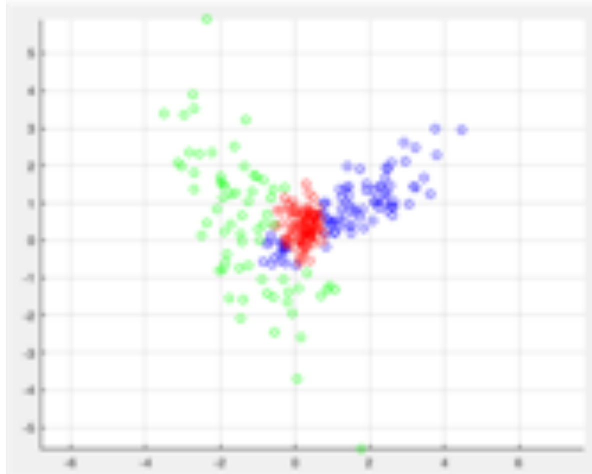


60 iterations

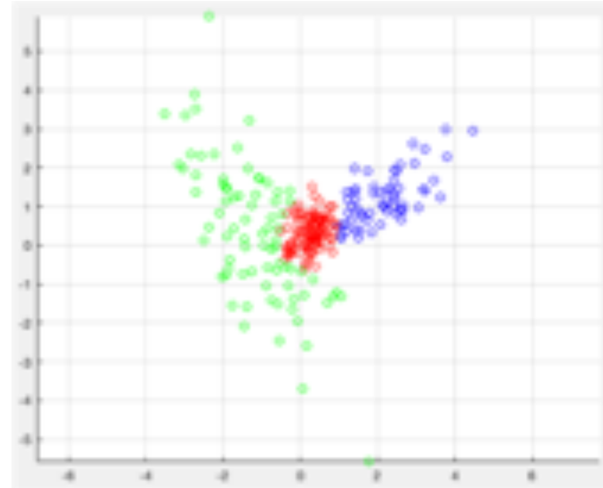
EM assuming mixture of 3 Gaussian components : no conditional indep assumptions, so non-spherical Gaussians



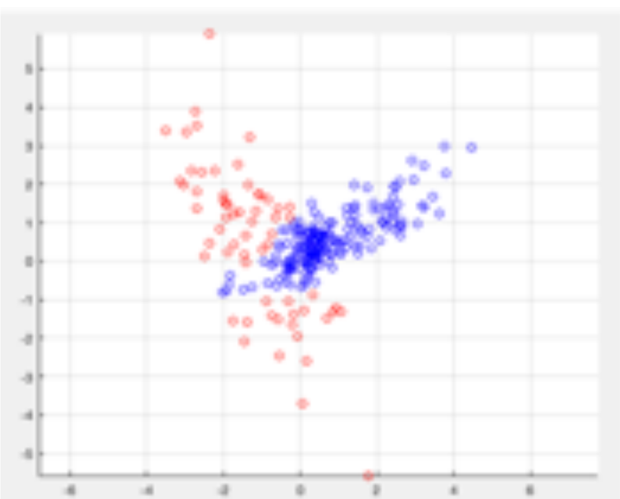
10 iterations



20 iterations

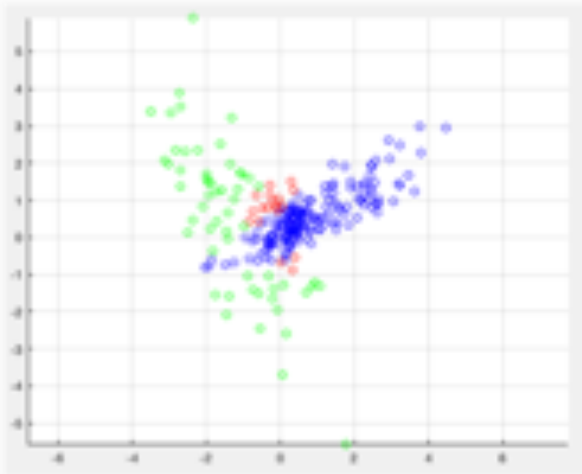


60 iterations

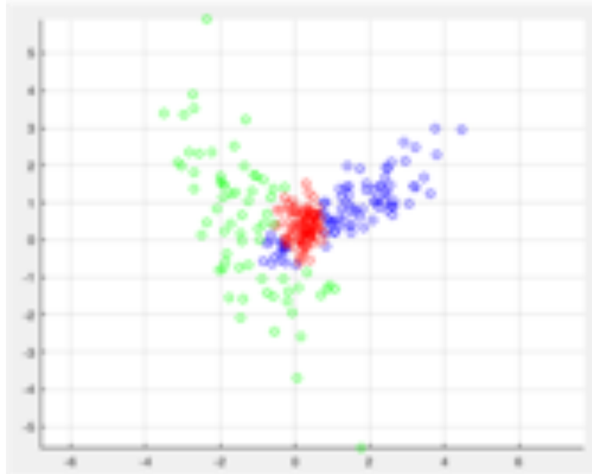


2 components

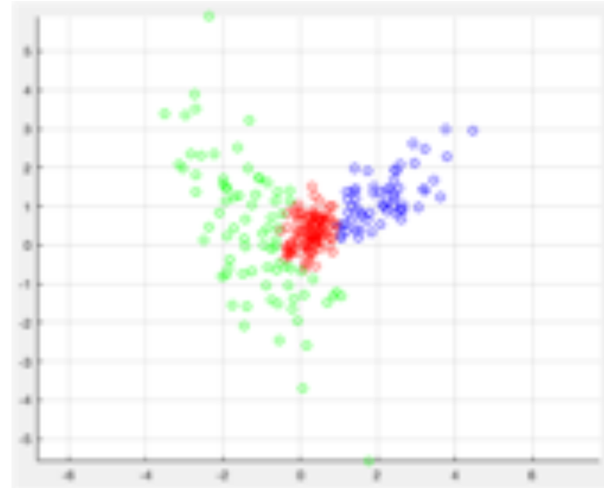
EM assuming mixture of 3 Gaussian components : no conditional indep assumptions, so non-spherical Gaussians



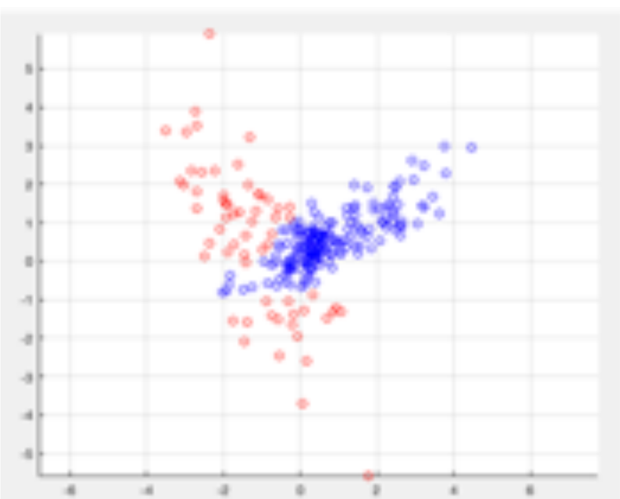
10 iterations



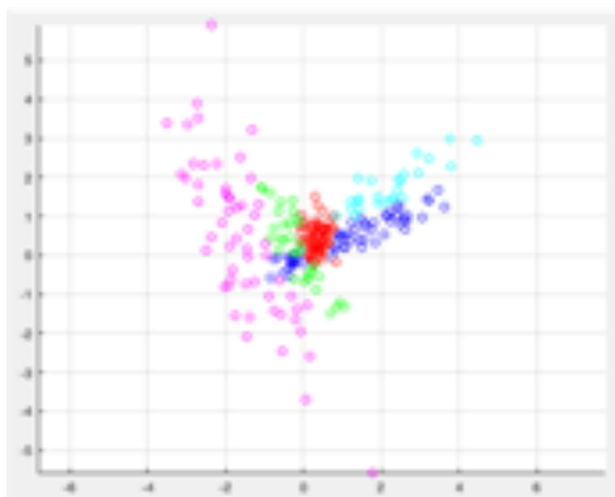
20 iterations



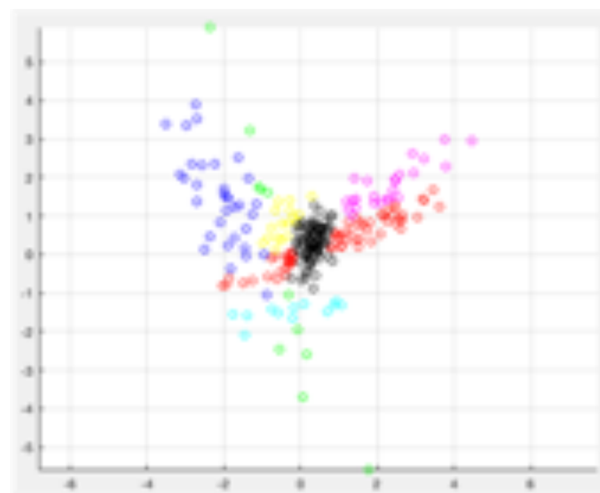
60 iterations



2 components

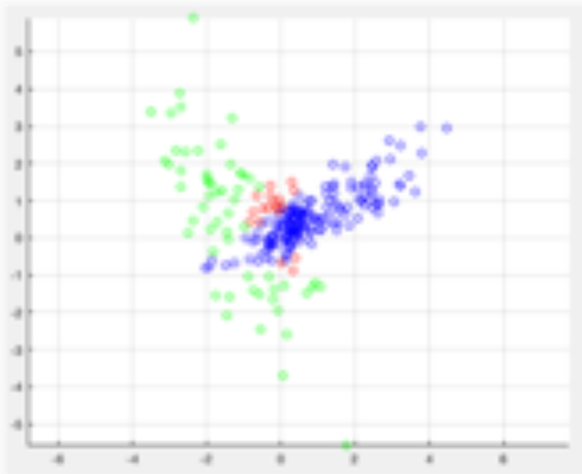


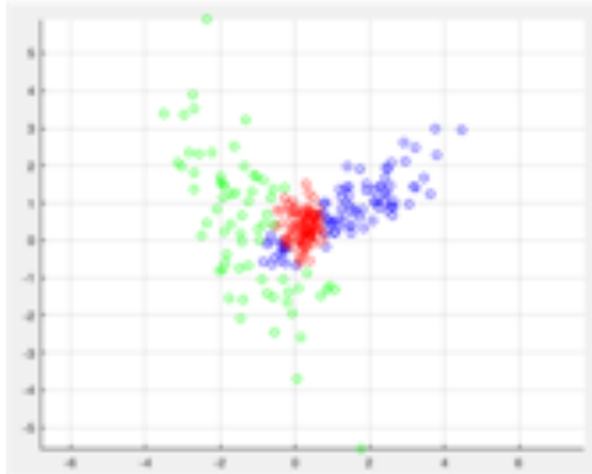
6 components

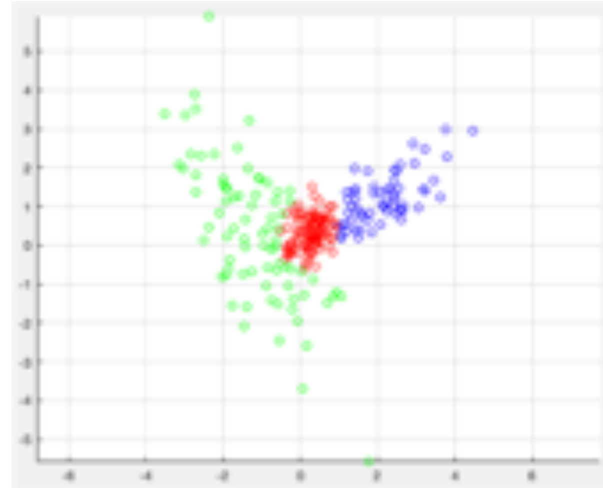


10 components

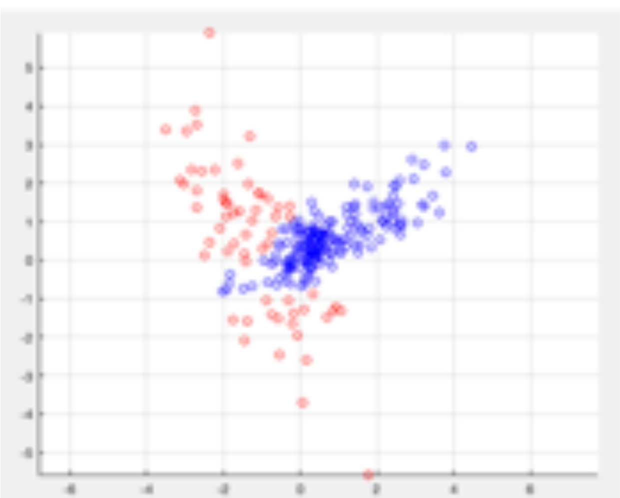
EM assuming mixture of 3 Gaussian components : no conditional indep assumptions, so non-spherical Gaussians

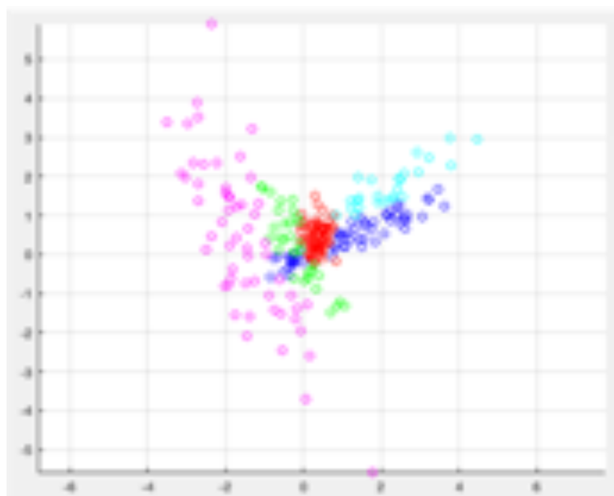

10 iterations

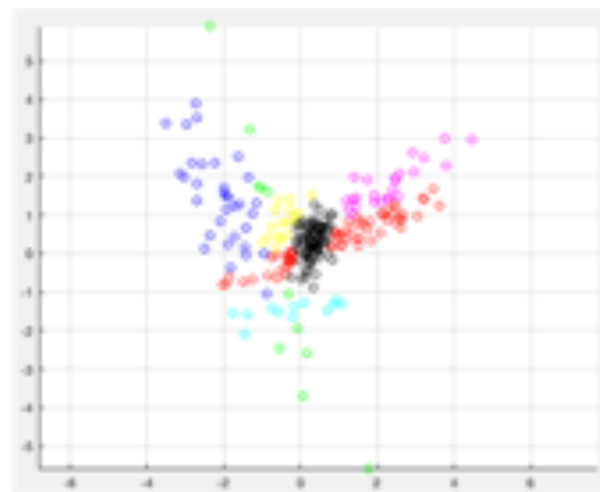

20 iterations


60 iterations

How should we choose the number of clusters?


2 components


6 components


10 components

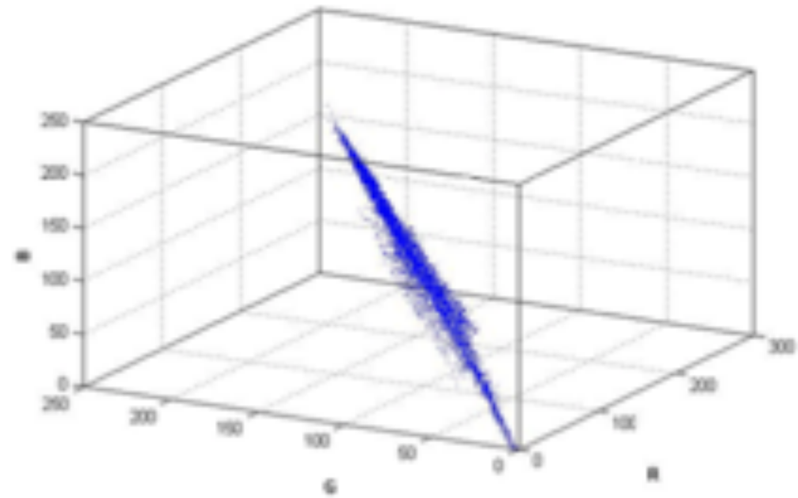
How to choose number k of clusters?

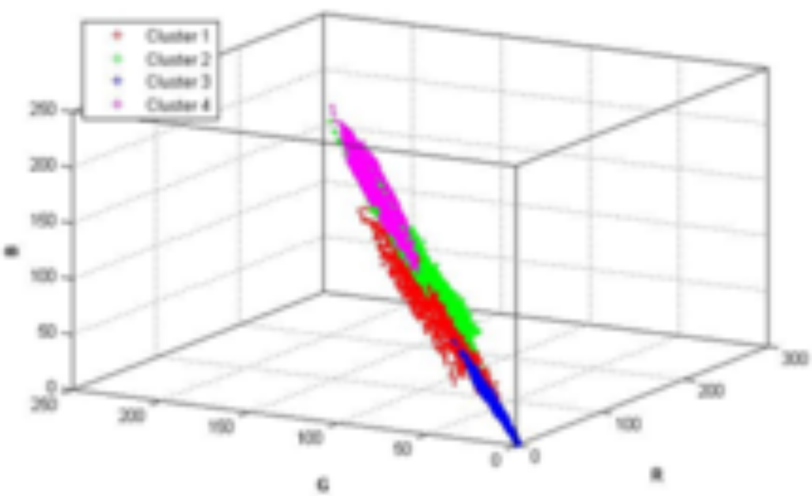
- We can try multiple values of k, evaluating each by the data likelihood P(Data | k component mixture model)
- Note if we do this on the training data, the k that maximizes
  P(trainData | k component mixture model)
  will be k = number of training examples!
- Use held-out test data to chose k
  P(testData | k component mixture model)

Applications of GMM in computer vision

1- Image segmentation:

$$X = (R, G, B)^T$$





[courtesy Mohand Saïd Allili]

What you should know about EM mixture model clustering

- Another application of EM to learn from partially observed data
- Unobserved variable: cluster label
- Based on Bayes net that models mixture distribution
- Can use this for both discrete-valued, real-valued X_i
- Doesn't answer the question of *how many* clusters to assume
  - But cross validation can reveal which choice is best on held-out data

Learning Bayes Net Structure

How can we learn Bayes Net graph structure?

In general case, open problem

- can require lots of data (else high risk of overfitting)
- can use Bayesian priors, or other kinds of prior assumptions about graph structure to constrain search

One key result:

- Chow-Liu algorithm: finds "best" tree-structured network
- What's best?
  - suppose P(**X**) is true distribution, T(**X**) is distribution of our tree-structured network, where $\mathbf{X} = \langle X_1, \ldots X_n \rangle$
  - Chow-Liu minimizes Kullback-Leibler divergence:

$$KL(P(\mathbf{X}) \mid\mid T(\mathbf{X})) \equiv \sum_k P(\mathbf{X}=k) \log \frac{P(\mathbf{X}=k)}{T(\mathbf{X}=k)}$$

Kullback-Leibler Divergence

- KL(P(X) || T(X)) is a measure of the difference between probability distributions P(X) and T(X)

$$KL(P(\mathbf{X}) \,||\, T(\mathbf{X})) \equiv \sum_k P(\mathbf{X}=k) \log \frac{P(\mathbf{X}=k)}{T(\mathbf{X}=k)}$$

- It is assymetric, always greater or equal to 0
- It is 0 iff P(X)=T(X)

Chow-Liu Algorithm

Key result: To minimize KL(P || T) over possible tree networks T approximating true P, it suffices to find the tree network T that maximizes the sum of mutual informations over its edges

Mutual information for an edge between variable A and B:

$$I(A, B) = \sum_a \sum_b P(a, b) \log \frac{P(a, b)}{P(a)P(b)}$$

This works because for tree networks with nodes $\mathbf{X} \equiv \langle X_1 \ldots X_n \rangle$

$$\begin{aligned} KL(P(\mathbf{X}) \mid\mid T(\mathbf{X})) &\equiv \sum_k P(\mathbf{X} = k) \log \frac{P(\mathbf{X} = k)}{T(\mathbf{X} = k)} \\ &= -\sum_i I(X_i, Pa(X_i)) + \sum_i H(X_i) - H(X_1 \ldots X_n) \end{aligned}$$

Chow-Liu Algorithm

1. for each pair of variables A,B, use training data to estimate P(A,B), P(A), and P(B)

2. for each pair A, B calculate mutual information

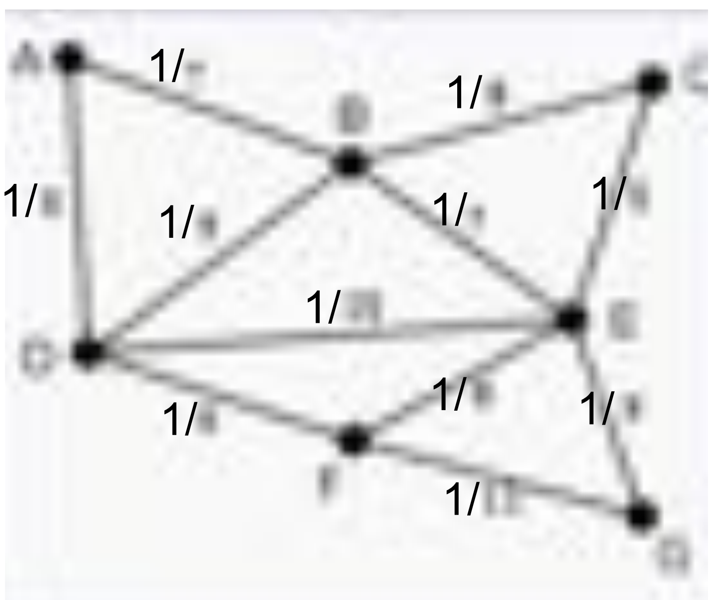
$$I(A,B) = \sum_a \sum_b P(a,b) \log \frac{P(a,b)}{P(a)P(b)}$$

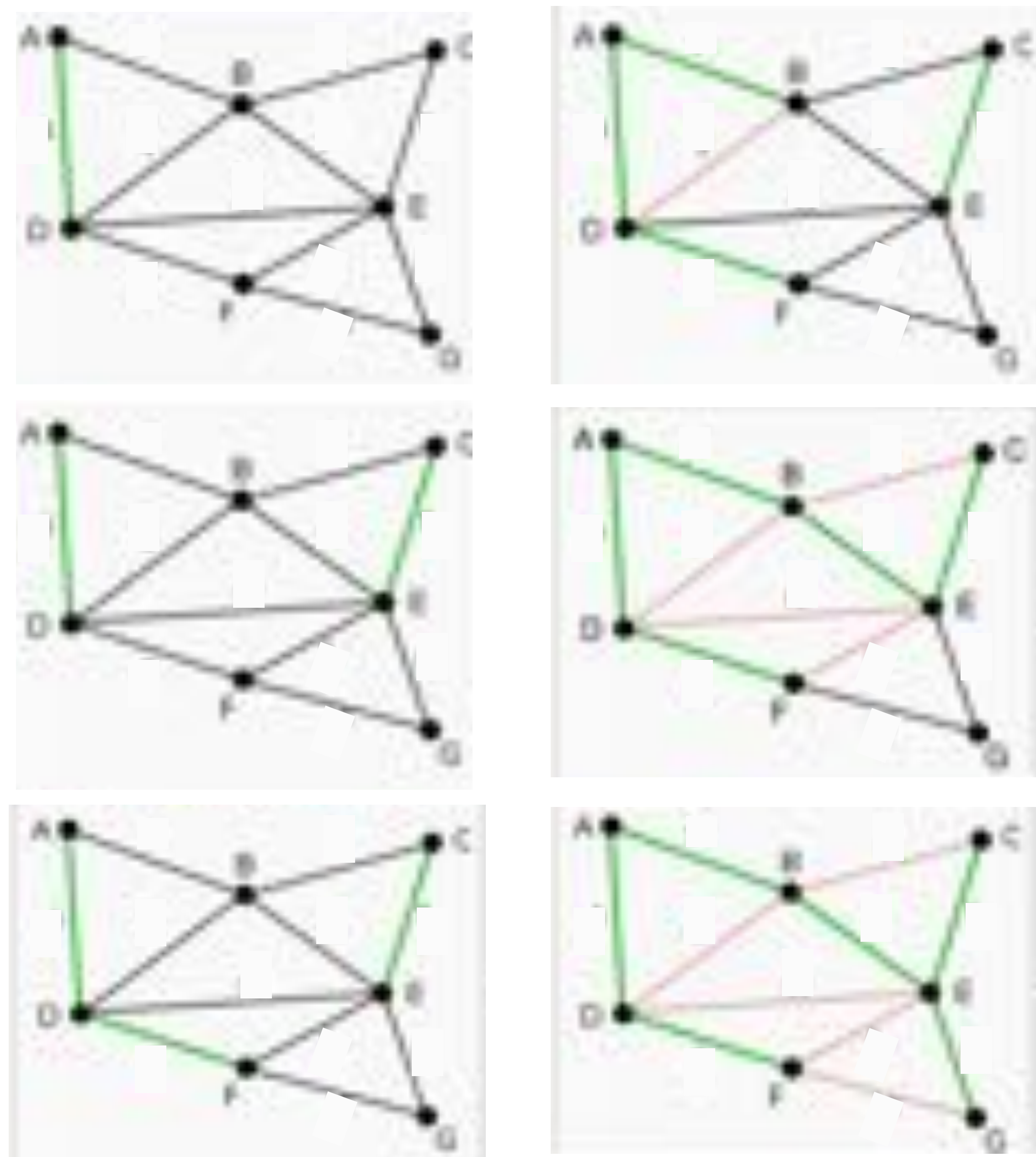
3. calculate the maximum spanning tree over the set of variables, using edge weights *I(A,B)*
   (given N vars, this costs only $O(N^2)$ time)

4. add arrows to edges to form a directed-acyclic graph

5. learn the CPD's for this graph

Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree





[courtesy A. Singh, C. Guestrin]

Bayes Nets – What You Should Know

- Representation
  - Bayes nets represent joint distribution as a DAG + Conditional Distributions
  - D-separation lets us decode conditional independence assumptions
- Inference
  - NP-hard in general
  - For some graphs, closed form inference is feasible
  - Approximate methods too, e.g., Monte Carlo methods, …
- Learning
  - Easy for known graph, fully observed data (MLE's, MAP est.)
  - EM for partly observed data, known graph
  - Learning graph structure: Chow-Liu for tree-structured networks
  - Hardest when graph unknown, data incompletely observed