F20DL and F21DL: Part 2: Machine Learning
Lecture 4. Unsupervised Learning: soft clustering

Katya Komendantskaya

Quick note on test examples

- You see last year lectures on Vision ahead of time
- After each lecture, I update the slides, – please use new slides

One important change happens to last year slides

- I change the data set used in the tests
- This is to avoid cases when last year students share last year answers with new students
- When you work on the test, please ensure that you use this year slides

Last lecture recap

- Yesterday, we considered an iconic clustering algorithm – **k-means**.
- Two practical issues remain:

Last lecture recap

- Yesterday, we considered an iconic clustering algorithm – **k-means**.
- Two practical issues remain:
    1. choice of the starting points: the first centroid rather than the first random cluster assignment
    2. the best choice of the number of clusters

Last lecture recap

- Yesterday, we considered an iconic clustering algorithm – **k-means**.
- Two practical issues remain:
  1. choice of the starting points: the first centroid rather than the first random cluster assignment
  2. the best choice of the number of clusters

Lets consider them by means of an example.

Practical task

- Take a given data set and load it to Weka;
- Run *k*-means algorithm, choose the number of clusters — 2
- Answer the following questions:
  - Q1 Did the result correspond to the classes in the initial data?
  - Q2 Which seed gave better accuracy?
  - Q3 Does increase in the number of clusters help?

Customer transactions

Trans.	Music on CD?	Music on MP3?	Board Games	On-line Games	Output
T1	No	Yes	No	Yes	Buys
T2	Yes	No	No	No	Cancels
T3	Yes	No	No	Yes	Buys
T4	Yes	No	Yes	No	Cancels
T5	No	Yes	No	No	Cancels
T6	No	Yes	Yes	No	Cancels
T7	No	No	No	Yes	Buys
T8	No	Yes	Yes	Yes	Cancels
T9	Yes	Yes	No	No	Cancels
T10	Yes	Yes	No	Yes	Buys

Answers in Weka

```
@relation transactions.nominal
@attribute musicCD yes, no
@attribute musicMP3 yes, no
@attribute gamesHard yes, no
@attribute gamesOnline yes, no
@attribute Buys buys, cancels
@data
no,yes,no,yes,buys
yes,no,no,no,cancels
yes,no,no,yes,buys
yes,no,yes,no,cancels
no,yes,no,no,cancels
no,yes,yes,no,cancels
no,no,no,yes,buys
no,yes,yes,yes,cancels
yes,yes,no,no,cancels

yes,yes,no,yes,buys
```

We tell Weka to ignore the class: "Classes to clusters evaluation" option.

Run of $k-means$

(I chose numClusters to be 2)

```
Final cluster centroids:
                                  Cluster#
Attribute        Full Data              0           1
                    (10.0)          (4.0)       (6.0)
==============================================
musicCD                yes            yes          no
musicMP3               yes            yes         yes
gamesHard               no            yes          no
gamesOnline            yes             no         yes
```

1. Did the result correspond to the classes in the initial data?

Run of $k - means$

```
Clustered Instances

0       4 ( 40%)
1       6 ( 60%)

Class attribute: Buys
Classes to Clusters:

 0 1  <-- assigned to cluster
 0 4 | buys
 4 2 | cancels

Cluster 0 <-- cancels
Cluster 1 <-- buys

Incorrectly clustered instances : 2.0  20      %
```

Q2: which seed is better?

- My previous run was for seed 10:

```
Initial starting points (random):

Cluster 0: yes,no,yes,no
Cluster 1: no,no,no,yes
```

Will try a few more...

Q2: which seed is better?

- My first run was for seed 10 (accuracy 80 %)

```
Initial starting points (random):

Cluster 0: yes,no,yes,no
Cluster 1: no,no,no,yes
```

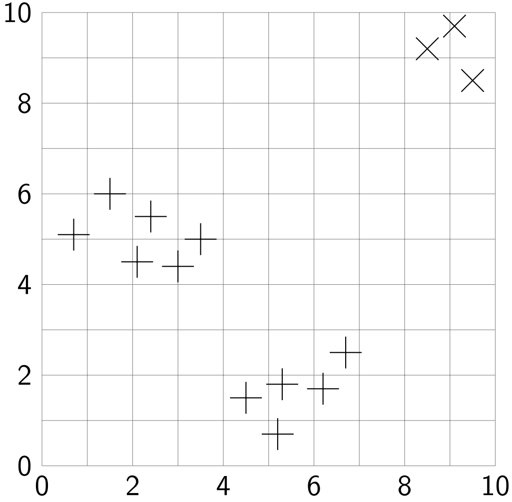
- Seed 5, accuracy 60 %

```
Initial starting points (random):

Cluster 0: no,yes,yes,yes
Cluster 1: yes,no,no,no
```

- Seed 12, accuracy 70 %

```
Initial starting points (random):

Cluster 0: no,no,no,yes
Cluster 1: no,yes,yes,no
```

Q3. Would increasing number of clusters help?



Q3. Would increasing number of clusters help?

```
Final cluster centroids:
                                Cluster#
Attribute      Full Data          0           1           2
                  (10.0)       (3.0)       (4.0)       (3.0)
=========================================================
musicCD              yes         yes          no         yes
musicMP3             yes          no         yes          no
gamesHard             no          no         yes          no
gamesOnline          yes         yes         yes          no

Class attribute: Buys
Classes to Clusters:

 0 1 2  <-- assigned to cluster
 3 1 0 | buys
 0 3 3 | cancels

Cluster 0 <-- buys
Cluster 1 <-- No class
Cluster 2 <-- cancels
```

Q3. Would increasing number of clusters help?

- Sometimes it does...
- For Purchase data set – no, it drops to 60 % at best... (e.g. with the seed 10, as we saw)

Test 2, Part 2

Answer the same questions Q1-Q3 playing with the small face emotion recognition set in Weka.

- Take the set with 7 instances and 3 features first

```
@data

 White , Black ,  White , Happy
 Black , Black ,  White , Happy
 White , White ,  Black , Sad
 White , White ,  White , Sad
 Black , White ,  Black , Happy
 White , Black ,  Black , Sad
```

- Take the full data set
- Same research question: how easy is it to get the right classes from data? how bad is influence of confusing features?

Plan for today

- We saw Bayesian Learning and k-means clustering

Plan for today

- We saw Bayesian Learning and k-means clustering
- ...Two iconic representatives of two main machine learning styles

Plan for today

- We saw Bayesian Learning and k-means clustering
- …Two iconic representatives of two main machine learning styles
- Machine Learning research is, in big part, about developing new combinations of learning algorithms

Plan for today

- We saw Bayesian Learning and k-means clustering
- …Two iconic representatives of two main machine learning styles
- Machine Learning research is, in big part, about developing new combinations of learning algorithms
- We will consider an example of one such hybrid machine learning method – EM algorithm – that combines clustering and Bayesian learning.

Plan for today

- We saw Bayesian Learning and *k*-means clustering
- …Two iconic representatives of two main machine learning styles
- Machine Learning research is, in big part, about developing new combinations of learning algorithms
- We will consider an example of one such hybrid machine learning method – EM algorithm – that combines clustering and Bayesian learning.
- It belongs to a subclass of clustering methods – soft clustering.

Plan for today

- We saw Bayesian Learning and k-means clustering
- ...Two iconic representatives of two main machine learning styles
- Machine Learning research is, in big part, about developing new combinations of learning algorithms
- We will consider an example of one such hybrid machine learning method – EM algorithm – that combines clustering and Bayesian learning.
- It belongs to a subclass of clustering methods – soft clustering.

Lets start with recapping a few important ideas from Bayes nets that we will need today

Recap of last week: Variable elimination algorithm

The task:

Given observation on variables $Y_1, \ldots, Y_j$, compute posterior probability of Z.

To compute $P(Z|Y_1 = v_1 \wedge \ldots \wedge Y_j = v_j)$:

1. Construct a factor for each conditional probability.
2. Set the observed variables to their observed values.
3. Sum out each of the other variables (the $\{Z_1, \ldots, Z_k\}$) according to some elimination ordering.
4. Multiply the remaining factors. Normalize by dividing the resulting factor $f(Z)$ by $\sum_Z f(Z)$.

It is exactly your Test1.2 exercise

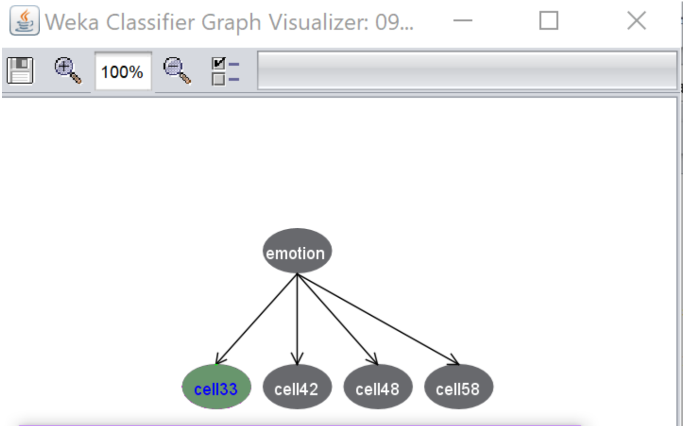
Recap of last week: Conditionals from probabilities

Table 4.2 The Weather Data, With Counts and Probabilities

Outlook			Temperature			Humidity			Windy			Play	
	Yes	***No***		***Yes***	***No***		***Yes***	***No***		***Yes***	***No***	***Yes***	***No***
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Reminder of Test1.2 exercise: Any data set gives rise to a table with conditionals. And any table with conditionals gives rise to...

Idea 1: Naive Bayes Net from Data

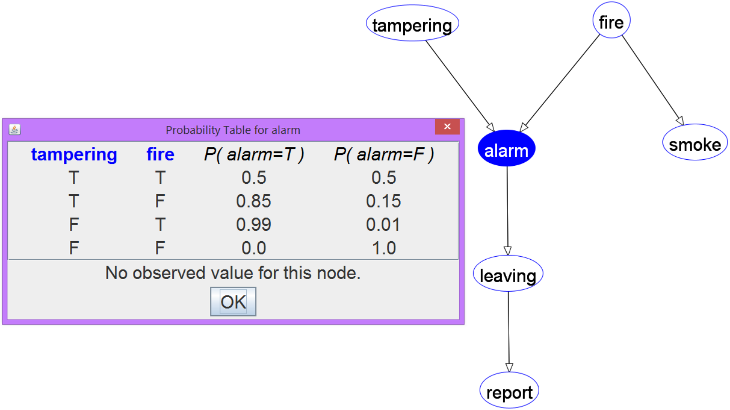


Probability Distribution Table For cell33

emotion	Black	White
Happy	0.625	0.375
Sad	0.312	0.688

Idea 2: Factors and Summing out variables

- A factor is a representation of a function from a tuple of random variables into a number.



Probability Table for alarm

tampering	fire	P(alarm=T)	P(alarm=F)
T	T	0.5	0.5
T	F	0.85	0.15
F	T	0.99	0.01
F	F	0.0	1.0

No observed value for this node.

OK

Most often, this function will be displayed as an array. The above table shows a factor $f(tampering, fire)$.

Idea 2: Factors and Summing out variables

We can sum out a variable, say X_1 with domain $\{v_1, \ldots, v_k\}$, from factor $f(X_1, \ldots, X_j)$, resulting in a factor on $X_2, \ldots, X_j$ defined by:

$$
\begin{aligned}
&(\sum_{X_1} f)(X_2, \ldots, X_j) \\
&= \quad f(X_1{=}v_1, \ldots, X_j) + \cdots + f(X_1{=}v_k, \ldots, X_j)
\end{aligned}
$$

Idea 2: Factors and Summing out a variable example

f_3:

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

$\sum_B f_3$:

A	C	val
t	t	0.57
t	f	
f	t	
f	f	

Idea 2: Factors and Summing out a variable example

f_3:

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

$\sum_B f_3$:

A	C	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

Stop and think: how big is the array $\sum_B f_3$? Do numbers sum up to 1? Why?

EM Algorithm: general idea

- Used for soft clustering — examples are probabilistically in classes.

EM Algorithm: general idea

- Used for soft clustering — examples are probabilistically in classes.
- Uses Naive Bayes classifier

EM Algorithm: general idea

- Used for soft clustering — examples are probabilistically in classes.
- Uses Naive Bayes classifier
- One class with k values $\{1, \ldots, k\}$, where k – is the user-defined number of clusters

EM Algorithm: general idea

- Used for soft clustering — examples are probabilistically in classes.
- Uses Naive Bayes classifier
- One class with k values $\{1, \ldots, k\}$, where k – is the user-defined number of clusters
- All other attributes/features are just children of the class node

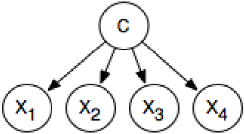
EM Algorithm: general idea

- Used for soft clustering — examples are probabilistically in classes.
- Uses Naive Bayes classifier
- One class with k values $\{1, \ldots, k\}$, where k – is the user-defined number of clusters
- All other attributes/features are just children of the class node
- Input features probabilistically depend on the class, but independent of each other

EM Algorithm: general idea

- Given: Model and data; incl k-valued random variable C
- Output: Produce probabilities needed for the classifier

Model



Data

X_1	X_2	X_3	X_4
t	f	t	t
f	t	t	f
f	f	t	t
	$\dots$		

$\Longrightarrow$ Probabilities

$P(C)$
$P(X_1|C)$
$P(X_2|C)$
$P(X_3|C)$
$P(X_4|C)$

EM Algorithm Overview

- Repeat the following two steps:
  - E-step: Augment your data with new variable C (the class whose values represent clusters); assign counters to each value of the class C
  - M-step: infer the maximum likelihood or maximum aposteriori probability from the data.
- Start either with made-up counts or made-up probabilities.
- EM will converge to a local maxima.

EM Algorithm : technical step 1

- Augment the data with:
  1. a class feature C, with k values
  2. The count column
- Map each original tuple into k tuples, one for each class
- The counts are assigned randomly, one for each class

EM Algorithm : technical step 1

- Augment the data with:
  1. a class feature C, with k values
  2. The count column
- Map each original tuple into k tuples, one for each class
- The counts are assigned randomly, one for each class

Suppose $k = 3$, and $dom(C) = \{1, 2, 3\}$.

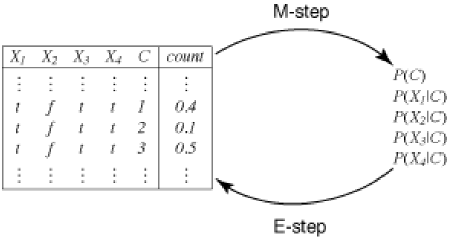
Initial Data

X_1	X_2	X_3	X_4
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t	f	t	t
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Augmented data

X_1	X_2	X_3	X_4	C	$Count$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t	f	t	t	1	0,4
t	f	t	t	2	0,1
t	f	t	t	3	0,5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

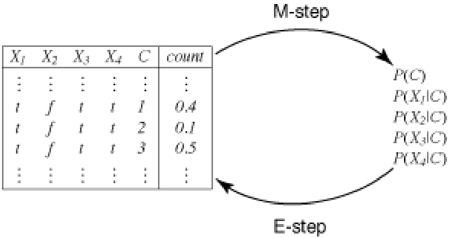
EM Algorithm



X_1	X_2	X_3	X_4	C	*count*
⋮	⋮	⋮	⋮	⋮	⋮
t	*f*	*t*	*t*	*1*	*0.4*
t	*f*	*t*	*t*	*2*	*0.1*
t	*f*	*t*	*t*	*3*	*0.5*
⋮	⋮	⋮	⋮	⋮	⋮

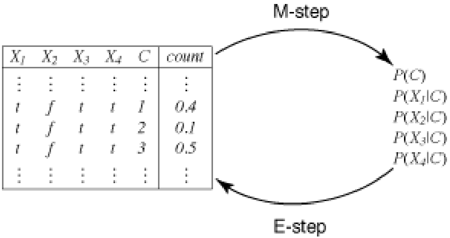
- Note: the “augmented data” is in fact a factorisation table for $X_1, \ldots X_n, C$.

EM Algorithm



- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

EM Algorithm

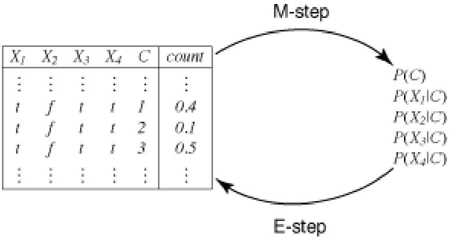


- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

M-step, by example

- $M_1[X_1, C] = \Sigma_{X_2,\ldots,X_4} A[X_1, \ldots, X_4, C]$ What will this array look like?

EM Algorithm

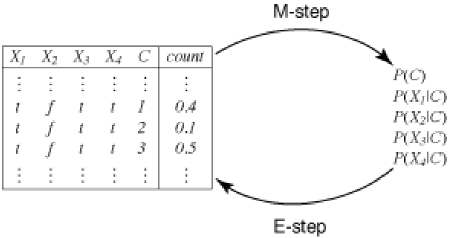


- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

M-step, by example

- $M_1[X_1, C] = \Sigma_{X_2,\ldots,X_4} A[X_1, \ldots, X_4, C]$ What will this array look like?
- Normalise each $M_i[X_i, C]$ to get probabilities: $P_i[X_i, C] = \frac{M_i[X_i, C]}{\sum_C M_i[X_i, C]}$.

EM Algorithm

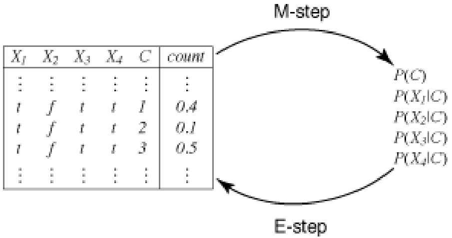


- Suppose $A[X_1, \dots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

M-step, by example

- $M_1[X_1, C] = \Sigma_{X_2,\dots,X_4} A[X_1, \dots, X_4, C]$ What will this array look like?
- Normalise each $M_i[X_i, C]$ to get probabilities: $P_i[X_i, C] = \frac{M_i[X_i, C]}{\sum_C M_i[X_i, C]}$.
  Stop and think: what are they?

EM Algorithm

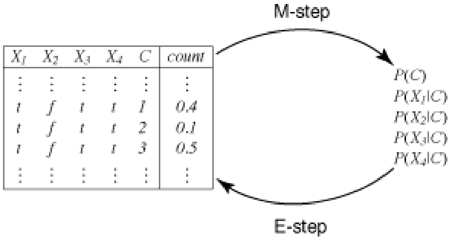


- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

M-step, by example

- $M_1[X_1, C] = \Sigma_{X_2,\ldots,X_4} A[X_1, \ldots, X_4, C]$ What will this array look like?
- Normalise each $M_i[X_i, C]$ to get probabilities: $P_i[X_i, C] = \frac{M_i[X_i, C]}{\sum_C M_i[X_i, C]}$.
  Stop and think: what are they?
- Finally, $P[C] = \frac{\Sigma_{X_1,\ldots,X_4} A[X_1,\ldots,X_4,C]}{s}$

EM Algorithm

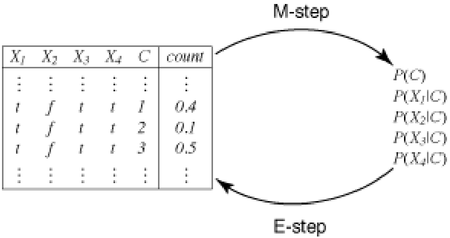


- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

M-step, by example

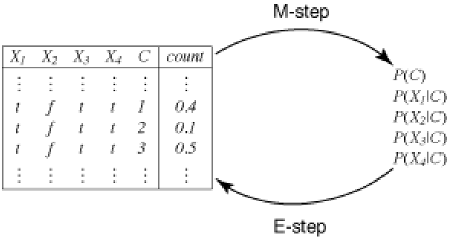
- $M_1[X_1, C] = \Sigma_{X_2,\ldots,X_4} A[X_1, \ldots, X_4, C]$ What will this array look like?
- Normalise each $M_i[X_i, C]$ to get probabilities: $P_i[X_i, C] = \frac{M_i[X_i,C]}{\sum_C M_i[X_i,C]}$. Stop and think: what are they?
- Finally, $P[C] = \frac{\Sigma_{X_1,\ldots,X_4} A[X_1,\ldots,X_4,C]}{s}$ What did we get now?

EM Algorithm



- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

EM Algorithm



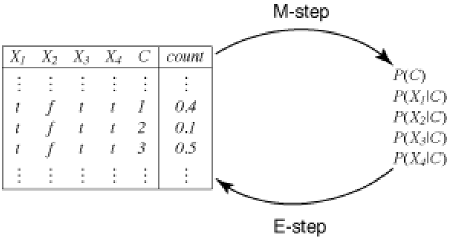
X_1	X_2	X_3	X_4	C	count
⋮	⋮	⋮	⋮	⋮	⋮
t	f	t	t	1	0.4
t	f	t	t	2	0.1
t	f	t	t	3	0.5
⋮	⋮	⋮	⋮	⋮	⋮

- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

E-step, by example

- Update the counts in A, based on posterior probabilities of C. E.g. replace 0,4 with:

EM Algorithm

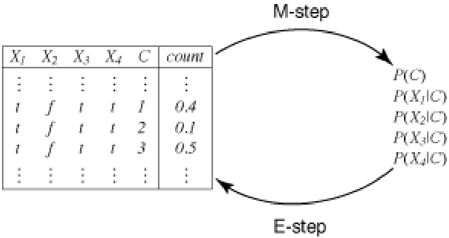


- Suppose $A[X_1, \dots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

E-step, by example

- Update the counts in A, based on posterior probabilities of C. E.g. replace 0,4 with:
- $P(C = 1|X_1 = t, X_2 = f, X_3 = t, X_4 = t)$

EM Algorithm

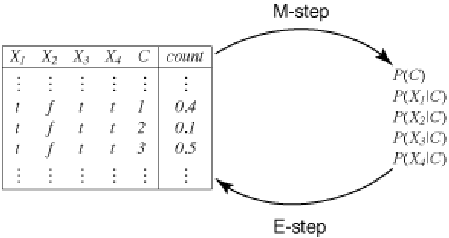


- Suppose $A[X_1, \ldots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

E-step, by example

- Update the counts in A, based on posterior probabilities of C. E.g. replace 0,4 with:
- $P(C = 1|X_1 = t, X_2 = f, X_3 = t, X_4 = t)$ where do we take this from?

EM Algorithm



- Suppose $A[X_1, \dots X_n, C]$ is augmented data and we have s examples
- $M_i[X_i, C]$ is the marginal probability $P(X_i, C)$ derived from A
- $P_i[X_i, C]$ is the conditional probability $P(X_i|C)$

E-step, by example

- Update the counts in A, based on posterior probabilities of C. E.g. replace 0,4 with:
- $P(C = 1|X_1 = t, X_2 = f, X_3 = t, X_4 = t)$ where do we take this from? $= \frac{P(X_1=t|C=1)P(X_2=f|C=1)P(X_3=t|C=1)P(X_4=t|C=1)P(C=1)}{\Sigma_{i=1}^{3} P(X_1=t|C=i)P(X_2=f|C=i)P(X_3=t|C=i)P(X_4=t|C=i)P(C=i)}$
- (remember similar likelihoods and normalisation in your test 1 exercises)

EM Algorithm – generally

- Repeat the following two steps:
  - E-step: update the augmented data based on the probability distribution. Suppose there are m copies of the tuple $< X_1 = v_1, \ldots, X_n = v_n >$ in the original data. In the augmented data, the count associated with class c, stored in $A[v_1, \ldots v_n, c]$ is updated to

$$m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$$

EM Algorithm – generally

- Repeat the following two steps:
  - E-step: update the augmented data based on the probability distribution. Suppose there are m copies of the tuple $< X_1 = v_1, \ldots, X_n = v_n >$ in the original data. In the augmented data, the count associated with class c, stored in $A[v_1, \ldots v_n, c]$ is updated to

    $$m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$$

  - M-step infers the maximum likelihood or maximum aposteriori probability from the data.

EM Algorithm – generally

- Repeat the following two steps:
  - E-step: update the augmented data based on the probability distribution. Suppose there are m copies of the tuple $< X_1 = v_1, \ldots, X_n = v_n >$ in the original data. In the augmented data, the count associated with class c, stored in $A[v_1, \ldots v_n, c]$ is updated to

$$m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$$

  - M-step infers the maximum likelihood or maximum aposteriori probability from the data.
- Similarity with k-means: E step assigns examples to classes, M steps determines what the classes predict.

EM Algorithm – generally

- Repeat the following two steps:
  - E-step: update the augmented data based on the probability distribution. Suppose there are m copies of the tuple $< X_1 = v_1, \ldots, X_n = v_n >$ in the original data. In the augmented data, the count associated with class c, stored in $A[v_1, \ldots v_n, c]$ is updated to

    $$m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$$

  - M-step infers the maximum likelihood or maximum aposteriori probability from the data.
- Similarity with k-means: E step assigns examples to classes, M steps determines what the classes predict.
- When to terminate?

EM Algorithm – generally

- Repeat the following two steps:
  - E-step: update the augmented data based on the probability distribution. Suppose there are m copies of the tuple $< X_1 = v_1, \ldots, X_n = v_n >$ in the original data. In the augmented data, the count associated with class c, stored in $A[v_1, \ldots v_n, c]$ is updated to

$$m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$$

  - M-step infers the maximum likelihood or maximum aposteriori probability from the data.
- Similarity with k-means: E step assigns examples to classes, M steps determines what the classes predict.
- When to terminate? When the changes to counters are "small enough".

EM Algorithm – a closer look

1. **Algorithm** EM (X,D,k)
2. **Inputs:** X - set of features $\{X_1, \ldots, X_n\}$
3. D data set of features $\{X_1, \ldots, X_n\}$
4. k number of clusters

EM Algorithm – a closer look

1. **Algorithm** EM (X,D,k)
2. **Inputs:** X - set of features $\{X_1, \ldots, X_n\}$
3. D data set of features $\{X_1, \ldots, X_n\}$
4. k number of clusters
5. **Output:** $P(C)$, $P(X_i|C)$ for each $i \in \{1 : n\}$, where $C = \{1, \ldots, k\}$

EM Algorithm – a closer look

1. **Algorithm** EM (X,D,k)
2. **Inputs:** X - set of features $\{X_1, \ldots, X_n\}$
3. D data set of features $\{X_1, \ldots, X_n\}$
4. k number of clusters
5. **Output:** $P(C)$, $P(X_i|C)$ for each $i \in \{1:n\}$, where $C = \{1, \ldots, k\}$
6. **Local:** real array $A[X_1, \ldots, X_n, C]$
7. real array $P[C]$
8. real arrays $M_i[X_i, C]$ for each $i \in \{1:n\}$
9. real arrays $P_i[X_i, C]$ for each $i \in \{1:n\}$
10. s := number of tuples in D

EM Algorithm – a closer look

1. **Algorithm** EM (X,D,k)
2. **Inputs:** X - set of features $\{X_1, \ldots, X_n\}$
3. D data set of features $\{X_1, \ldots, X_n\}$
4. k number of clusters
5. **Output:** $P(C)$, $P(X_i|C)$ for each $i \in \{1 : n\}$, where $C = \{1, \ldots, k\}$
6. **Local:** real array $A[X_1, \ldots, X_n, C]$
7. real array $P[C]$
8. real arrays $M_i[X_i, C]$ for each $i \in \{1 : n\}$
9. real arrays $P_i[X_i, C]$ for each $i \in \{1 : n\}$
10. s := number of tuples in D
11. Assign $P(C)$, $P(X_i|C)$ arbitrarily

EM Algorithm – a closer look

Algorithm EM (X,D,k)
Inputs: X - set of features $\{X_1, \ldots, X_n\}$
D data set of features $\{X_1, \ldots, X_n\}$
k number of clusters
Output: $P(C)$, $P(X_i|C)$ for each $i \in \{1 : n\}$, where $C = \{1, \ldots, k\}$
Local: real array $A[X_1, \ldots, X_n, C]$
real array $P[C]$
real arrays $M_i[X_i, C]$ for each $i \in \{1 : n\}$
real arrays $P_i[X_i, C]$ for each $i \in \{1 : n\}$
s := number of tuples in D
Assign $P(C)$, $P(X_i|C)$ arbitrarily
repeat

E step

    for each assignment $\langle X_1 = v_1, \ldots X_n = v_n\rangle \in D$ do
      let $m = |\langle X_1 = v_1, \ldots X_n = v_n\rangle \in D|$
      for each $c \in \{1 \ldots k\}$ do
        $A[v_1, \ldots, v_n, c] = m \times P(C = c|X_1 = v_1, \ldots, X_n = v_n)$

EM Algorithm – a closer look

Algorithm EM (X,D,k)
Inputs: X - set of features $\{X_1, \ldots, X_n\}$
D data set of features $\{X_1, \ldots, X_n\}$
k number of clusters
Output: $P(C)$, $P(X_i|C)$ for each $i \in \{1:n\}$, where $C = \{1, \ldots, k\}$
Local: real array $A[X_1, \ldots, X_n, C]$
real array $P[C]$
real arrays $M_i[X_i, C]$ for each $i \in \{1:n\}$
real arrays $P_i[X_i, C]$ for each $i \in \{1:n\}$
s := number of tuples in D
Assign $P(C)$, $P(X_i|C)$ arbitrarily
repeat

E step

for each assignment $\langle X_1 = v_1, \ldots X_n = v_n \rangle \in D$ do
  let $m = |\langle X_1 = v_1, \ldots X_n = v_n \rangle \in D|$
  for each $c \in \{1 \ldots k\}$ do
    $A[v_1, \ldots, v_n, c] = m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$

M step

for each $i \in \{1:n\}$ do
  $M_i[X_i, C] = \sum_{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n} A[X_1, \ldots, X_n, C]$
  $P_i[X_i, C] = \frac{M_i[X_i, C]}{\sum_C M_i[X_i, C]}$
$P[C] = \frac{\sum_{X_1, \ldots, X_n} A[X_1, \ldots, X_n, C]}{s}$

Final Example in Weka

Compare *k*-means and EM algorithm.

@relation transactions.nominal
@attribute musicCD yes, no
@attribute musicMP3 yes, no
@attribute gamesHard yes, no
@attribute gamesOnline yes, no
@attribute Buys buys, cancels
@data
no,yes,no,yes,buys
yes,no,no,no,cancels
yes,no,no,yes,buys
yes,no,yes,no,cancels
no,yes,no,no,cancels
no,yes,yes,no,cancels
no,no,no,yes,buys
no,yes,yes,yes,cancels
yes,yes,no,no,cancels

yes,yes,no,yes,buys

We tell Weka to ignore the class: use "Classes to clusters evaluation" option.

Run of $k - means$

(I chose numClusters to be 2)

```
Final cluster centroids:
                                 Cluster#
Attribute         Full Data             0            1
                     (10.0)         (4.0)        (6.0)
==============================================
musicCD                 yes           yes           no
musicMP3                yes           yes          yes
gamesHard                no           yes           no
gamesOnline             yes            no          yes
```

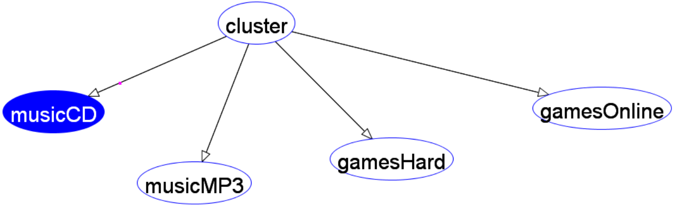
Run of $k - means$

```
Clustered Instances

0       4 ( 40%)
1       6 ( 60%)

Class attribute: Buys
Classes to Clusters:

 0 1  <-- assigned to cluster
 0 4 | buys
 4 2 | cancels

Cluster 0 <-- cancels
Cluster 1 <-- buys

Incorrectly clustered instances : 2.0  20      %
```

Run of EM algorithm

```
          Cluster
Attribute      0   1
            (0.6) (0.4)
==================
musicCD
  yes          2   5
  no           6   1
  [total]      8   6
musicMP3
  yes          6   2
  no           2   4
  [total]      8   6
gamesHard
  yes          3   2
  no           5   4
  [total]      8   6
gamesOnline
  yes          5   2
  no           3   4
  [total]      8   6
```

Output of EM to Bayes net



Probability Table for musicCD

cluster	P(musicCD=yes)	P(musicCD=no)
0	0.25	0.75
1	0.83	0.17

No observed value for this node.

OK

Run of EM algorithm

```
Class attribute: Buys
Classes to Clusters:

 0 1  <-- assigned to cluster
 3 1 | buys
 3 3 | cancels

Cluster 0 <-- buys
Cluster 1 <-- cancels

Incorrectly clustered instances : 4.0  40      %
```

Questions:

- Did it improve accuracy, compared to k-means?
- Did it (partially) recover the data table from Bayes Net exercise?

The EM Algorithm: conclusions

- a clever combination of ideas from k-means clustering and Bayesian learning
- Did not prove too useful on our toy examples
- In practice, may bring improvements, esp. when you are actually interested in probabilistic, rather than deterministic, results.
- Will it help on your big data set?

Test 2, Part 3

- Try running EM algorithm on the small (4 attributes $\times$ 10 instances) emotion recognition data set in Weka. Compare all settings and results with k-means clustering. Be ready to answer questions.
- Check relevant chapters on Clustering in the recommended textbook: Data Mining, by Witten et al. pp 273-294, pp.480-485. (in 2011 edition) (in 2017 edition – pp. 141 – 156)
- **EM algorithm:** pp. 285 – 288 (in 2017 Edition – 353-356)
- After that, you will be ready to complete Coursework 2, on Bayesian Learning and Clustering.

Exact Test structure:

1. Part 1 (Q1-6): Manual calculations on the set given in the 1st Clustering lecture
2. Part 2 (Q7-13): k-means algorithm on Weka
3. Part 3 (Q14-16): EM algorithm on Weka

Test 1 Part 2

1. Part 2: run k-means on Weka, with k=2, and Eucledian distance as options: (The Weka configuration: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-lastI 5 -num-slots 1)
2. Use 5 iterations, rather than default 500
3. Use "Classes to clusters evaluation"
4. Use the following data sets:
   - The full facial emotion recognition set (4 attributes + 1 class, 10 instances) given in Lecture 1, Week 6.
   - The reduced facial recognition set (used in this lecture and attached on Weka)
5. Run this algorithm for these 2 data sets and the seeds 1, 3, 5, 8, 10. How do seeds correspond to our exercises in the lecture?
6. In each experiment, record seeds, final centroids and accuracy

To prepare, do the following:

6 For EM algorithm on Weka:

- Take the full data setfrom Lecture 1
- Run EM algorithm with seeds 100, 10, 50 (The Weka configuration: weka.clusterers.EM -I 100 -N 2 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1) + seed
- Record the probability distributions (in tables) for these 3 EM clustering experiments

7 Be ready to answer questions about all listed Weka experiments.