F20DL and F21DL: Part 2, Machine Learning Lecture 2 Bayesian learning and Bayes Nets

Katya Komendantskaya

Last time

... we discussed

- Possible worlds
- Unconditional and Conditional Probabilities
- Bayes Rules
- Examples of reasoning with the above

Today:

- Formalise the remaining theory (the notion of a random variable)
- Introduce Bayes Nets

Quick warm up

What are we learning in Bayesian Learning?

Quick warm up

What are the possible worlds here (in last lecture terminology)?

Picture	Cell 33	Cell 42	Cell 48	Cell 58	Face expression
P1	White	Black	White	White	Happy
P2	Black	Black	White	White	Happy
P3	White	White	White	Black	Sad
P4	White	White	Black	White	Sad
P5	Black	White	Black	Black	Happy
P6	White	White	Black	Black	Sad
P7	Black	White	White	Black	Sad
P8	Black	White	Black	Black	Sad
P9	White	Black	Black	Black	Sad
P10	White	Black	White	Black	Sad

Possible World Semantics, Revisited

- A possible world specifies an assignment of one value to each random variable.

Picture / World	Cell 33	Cell 42	Cell 48	Cell 58	Face expression
P1	White	Black	White	White	Happy

Possible World Semantics, Revisited

- A possible world specifies an assignment of one value to each random variable.

Picture / World	Cell 33	Cell 42	Cell 48	Cell 58	Face expression
P1	White	Black	White	White	Happy

- A random variable is a function from possible worlds into a set of values (the range of the random variable).

Picture/World	Cell 33
P1 $\longrightarrow$	White
P2 $\longrightarrow$	Black
P3 $\longrightarrow$	White
P4 $\longrightarrow$	White
P5 $\longrightarrow$	White
P6 $\longrightarrow$	White
P7 $\longrightarrow$	Black
P8 $\longrightarrow$	Black
P9 $\longrightarrow$	Black
P10 $\longrightarrow$	White

Random Variables

- The domain (or *range*) of a variable X, written $dom(X)$, is the set of values X can take.

Example

$dom(Cell33) = \{Black, White\}$

Random Variables

- The domain (or *range*) of a variable X, written $dom(X)$, is the set of values X can take.

Example

$dom(Cell33) = \{Black, White\}$

- A tuple of random variables $\langle X_1, \ldots, X_n \rangle$ is a complex random variable with domain $dom(X_1) \times \cdots \times dom(X_n)$. Often the tuple is written as $X_1, \ldots, X_n$.

Example

$\langle Cell33, Cell42, Cell48, Cell58 \rangle$

Random Variables

- The domain (or *range*) of a variable X, written $dom(X)$, is the set of values X can take.

Example

$dom(Cell33) = \{Black, White\}$

- A tuple of random variables $\langle X_1, \ldots, X_n \rangle$ is a complex random variable with domain $dom(X_1) \times \cdots \times dom(X_n)$. Often the tuple is written as $X_1, \ldots, X_n$.

Example

$\langle Cell33, Cell42, Cell48, Cell58 \rangle$

- Assignment $X = x$ means variable X has value x.

Example

$Cell33 = Black$

Possible World Semantics, Revisited

- $\omega \models X = x$
  means variable X is assigned value x in world ω.
  ... the rest of the theory introduced yesterday extends accordingly.

Possible World Semantics, Revisited

- $\omega \models X = x$
  means variable X is assigned value x in world ω.
  ... the rest of the theory introduced yesterday extends accordingly.

Example

$\mathit{Picture2} \models (\mathit{Cell33} = \mathit{Black})$...

Probability Distributions

- A probability distribution on a random variable X is a function $dom(X) \to [0,1]$ such that

$$x \mapsto P(X = x).$$

This is written as $P(X)$.

Example

We will be talking about distribution $P(Cell58)$ as a function from $\{Black, White\}$ to $[0,1]$:
$Black \longrightarrow 0{,}6$
$White \longrightarrow 0{,}4$

From Random variables to Propositions:

This also includes the case where we have tuples of variables.

- A proposition is a Boolean formula made from assignments of values to variables.

Example

$Cell33 = Black \wedge Cell42 = White$

- A probability distribution $P(X, Y)$ over X and Y assigns a value $P(X = x \wedge Y = y)$ for each $x \in dom(X)$ and $y \in dom(Y)$.

From Random variables to Propositions:

This also includes the case where we have tuples of variables.

- A proposition is a Boolean formula made from assignments of values to variables.

Example

$Cell33 = Black \land Cell42 = White$

- A probability distribution $P(X, Y)$ over X and Y assigns a value $P(X = x \land Y = y)$ for each $x \in dom(X)$ and $y \in dom(Y)$.
- E.g., $P(X, Y, Z)$ means $P(\langle X, Y, Z \rangle)$.
- When $dom(X)$ is infinite sometimes we need a probability density function... (like e.g. normal (Gaussian) distribution).

Chain Rule

Conditional Probabilities can be used to decompose conjunctions: Since $P(a|b) = \frac{P(a \wedge b)}{P(b)}$ we can also have $P(b \wedge a) = P(a|b)P(b)$!!!

Chain Rule

Conditional Probabilities can be used to decompose conjunctions: Since $P(a|b) = \frac{P(a \wedge b)}{P(b)}$ we can also have $P(b \wedge a) = P(a|b)P(b)$!!!

For any propositions $f_1, \ldots f_n$,

$$P(f_1 \wedge f_2 \wedge \ldots \wedge f_n)$$

$$=$$

Chain Rule

Conditional Probabilities can be used to decompose conjunctions: Since $P(a|b) = \frac{P(a \wedge b)}{P(b)}$ we can also have $P(b \wedge a) = P(a|b)P(b)$!!!

For any propositions $f_1, \ldots f_n$,

$$\begin{aligned} & P(f_1 \wedge f_2 \wedge \ldots \wedge f_n) \\ = \quad & P(f_n | f_1 \wedge \cdots \wedge f_{n-1}) \times \\ & P(f_1 \wedge \cdots \wedge f_{n-1}) \\ = \quad & \end{aligned}$$

Chain Rule

Conditional Probabilities can be used to decompose conjunctions: Since $P(a|b) = \frac{P(a \wedge b)}{P(b)}$ we can also have $P(b \wedge a) = P(a|b)P(b)$!!!

For any propositions $f_1, \ldots f_n$,

$$\begin{aligned}
&P(f_1 \wedge f_2 \wedge \ldots \wedge f_n) \\
&\quad = \quad P(f_n|f_1 \wedge \cdots \wedge f_{n-1}) \times \\
&\qquad\quad\; P(f_1 \wedge \cdots \wedge f_{n-1}) \\
&\quad = \quad P(f_n|f_1 \wedge \cdots \wedge f_{n-1}) \times \\
&\qquad\quad\; P(f_{n-1}|f_1 \wedge \cdots \wedge f_{n-2}) \times \\
&\qquad\quad\; P(f_1 \wedge \cdots \wedge f_{n-2}) \\
&\quad =
\end{aligned}$$

Chain Rule

Conditional Probabilities can be used to decompose conjunctions: Since $P(a|b) = \frac{P(a \wedge b)}{P(b)}$ we can also have $P(b \wedge a) = P(a|b)P(b)$!!!

For any propositions $f_1, \ldots f_n$,

$$
\begin{aligned}
& P(f_1 \wedge f_2 \wedge \ldots \wedge f_n) \\
= \quad & P(f_n|f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& P(f_1 \wedge \cdots \wedge f_{n-1}) \\
= \quad & P(f_n|f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& P(f_{n-1}|f_1 \wedge \cdots \wedge f_{n-2}) \times \\
& P(f_1 \wedge \cdots \wedge f_{n-2}) \\
= \quad & P(f_n|f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& P(f_{n-1}|f_1 \wedge \cdots \wedge f_{n-2}) \\
& \times \cdots \times P(f_3|f_1 \wedge f_2) \times P(f_2|f_1) \times P(f_1) \\
= \quad & \prod_{i=1}^{n} P(f_i|f_1 \wedge \cdots \wedge f_{i-1})
\end{aligned}
$$

Conditional independence

Random variable X is independent of random variable Y given random variable Z if,

$$P(X|Y,Z) = P(X|Z)$$

Conditional independence

Random variable X is independent of random variable Y given random variable Z if,

$$P(X|Y,Z) = P(X|Z)$$

i.e. for all $x_i \in dom(X)$, $y_j \in dom(Y)$, $y_k \in dom(Y)$ and $z_m \in dom(Z)$,

$$\begin{aligned} & P(X = x_i | Y = y_j \wedge Z = z_m) \\ = \quad & P(X = x_i | Y = y_k \wedge Z = z_m) \\ = \quad & P(X = x_i | Z = z_m). \end{aligned}$$

That is, knowledge of Y's value doesn't affect the belief in the value of X, given a value of Z.

Four Equivalent statements

1. X is conditionally independent of Y given Z
2. Y is conditionally independent of X given Z
3. $$P(X|Y,Z) = P(X|Z)$$
4. $$P(X,Y|Z) = P(X|Z)P(Y|Z)$$

Variables X and Y are unconditionally independent if

$$P(X,Y) = P(X)P(Y)$$

(i.e. they are independent given no observations)

The Chain rule and the above facts underlie Bayes Nets (also known as Belief networks).

Bayes Nets – main idea

- Exploiting conditional probabilities on all domains of all variables is computationally expensive

Bayes Nets – main idea

- Exploiting conditional probabilities on all domains of all variables is computationally expensive
- Given a random variable X, a small set of variables may exist that directly affect X

Bayes Nets – main idea

- Exploiting conditional probabilities on all domains of all variables is computationally expensive
- Given a random variable X, a small set of variables may exist that directly affect X
- ...and X is conditionally independent of the rest of variables, given values for the directly affecting variables

Bayes Nets – main idea

- Exploiting conditional probabilities on all domains of all variables is computationally expensive
- Given a random variable X, a small set of variables may exist that directly affect X
- ...and X is conditionally independent of the rest of variables, given values for the directly affecting variables
- The set of locally affecting variables is called the Markov blanket

Bayes net explores this locality

Bayes networks

1. Totally order the variables of interest: $X_1, \ldots, X_n$

Bayes networks

1. Totally order the variables of interest: $X_1, \ldots, X_n$
2. Theorem of probability theory (chain rule):
   $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$

Bayes networks

1. Totally order the variables of interest: $X_1, \ldots, X_n$
2. Theorem of probability theory (chain rule):
   $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$
3. The parents $parents(X_i)$ of X_i are those predecessors of X_i that render X_i independent of the other predecessors. That is,

Bayes networks

1. Totally order the variables of interest: $X_1, \ldots, X_n$
2. Theorem of probability theory (chain rule):
   $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1})$
3. The parents $parents(X_i)$ of X_i are those predecessors of X_i that render X_i independent of the other predecessors. That is,
   $parents(X_i) \subseteq X_1, \ldots, X_{i-1}$ and
   $P(X_i|parents(X_i)) = P(X_i|X_1, \ldots, X_{i-1})$
4. So taking (2) and (3) together,
   $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i))$

Bayes networks

1. Totally order the variables of interest: $X_1, \ldots, X_n$
2. Theorem of probability theory (chain rule):
   $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$
3. The parents $parents(X_i)$ of X_i are those predecessors of X_i that render X_i independent of the other predecessors. That is,
   $parents(X_i) \subseteq X_1, \ldots, X_{i-1}$ and
   $P(X_i | parents(X_i)) = P(X_i | X_1, \ldots, X_{i-1})$
4. So taking (2) and (3) together,
   $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i))$

The probability over all variables $P(X_1, \ldots, X_n)$ is called the joint probability distribution.
A Bayes net defines a factorisation of the joint probability distribution, where conditional probabilities form factors that multiply together.

Bayes networks

1. Totally order the variables of interest: $X_1, \dots, X_n$
2. Theorem of probability theory (chain rule):
   $P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \dots, X_{i-1})$
3. The parents $parents(X_i)$ of X_i are those predecessors of X_i that render X_i independent of the other predecessors. That is,
   $parents(X_i) \subseteq X_1, \dots, X_{i-1}$ and
   $P(X_i | parents(X_i)) = P(X_i | X_1, \dots, X_{i-1})$
4. So taking (2) and (3) together,
   $P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i))$

The probability over all variables $P(X_1, \dots, X_n)$ is called the joint probability distribution.
A Bayes net defines a factorisation of the joint probability distribution, where conditional probabilities form factors that multiply together.

5. A Bayes net can be visualised as a graph: the nodes are random variables; there is an arc from the parents of each node into that node.

Example: fire alarm belief network

Variables:

- Fire: there is a fire in the building
- Tampering: someone has been tampering with the fire alarm
- Smoke: what appears to be smoke is coming from a window
- Alarm: the fire alarm goes off
- Leaving: people are leaving the building *en masse*.
- Report: a colleague says that people are leaving the building *en masse*. (A noisy sensor for leaving.)

Example: fire alarm belief network

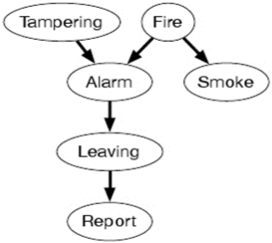
Variables:

- Fire: there is a fire in the building
- Tampering: someone has been tampering with the fire alarm
- Smoke: what appears to be smoke is coming from a window
- Alarm: the fire alarm goes off
- Leaving: people are leaving the building *en masse*.
- Report: a colleague says that people are leaving the building *en masse*. (A noisy sensor for leaving.)

- Variables: Fire, Tampering, Smoke, Alarm, Leaving, Report
- Domains: true, false
- assignment of values to variables: Fire = true
- We can define probability distributions from these domains to $[0, 1]$.

We may imagine the following network:

$P(tampering = true) = 0{,}02$
$P(fire = true) = 0{,}01$
$P(alarm|fire = true \wedge tampering = true) = 0{,}5$
$P(alarm|fire = true \wedge tampering = false) = 0{,}99$
$P(alarm|fire = false \wedge tampering = true) = 0{,}85$
$P(alarm|fire = false \wedge tampering = false) = 0{,}0001$
$P(smoke|fire = true) = 0{,}9$
$P(smoke|fire = false) = 0{,}01$
$P(leaving|alarm = true) = 0{,}88$
$P(leaving|alarm = false) = 0{,}001$
$P(report|leaving = true) = 0{,}75$
$P(report|leaving = false) = 0{,}01$



The network represents factorisation:

$P(tampering, fire, alarm, smoke, leaving, report) = P(tampering) * P(fire) * P(alarm|tampering \wedge fire) * P(smoke|fire) * P(leaving|alarm) * P(report|leaving)$

Now lets come back to our test exercises

- What kind of factorisation will your tables imply?
- How to work it out?

Check out pp. 90-94 of Witten et al. (2001) Data Mining book (esp. the **Table of Probabilities**); – practice this in Test 1, Part 2.

Edition 2017 – pp. 96-100

When you complete the table for our small facial recognition set, this table will give factorisation for:

$P(Cell33, Cel42, Cel48, Cell58, Emotion) =$
$P(Emotion) * P(Cell33|Emotion) * P(Cell42|Emotion) *$
$P(Cell48|Emotion) * P(Cell58|Emotion).$

Summary: Components of a belief network

A belief network consists of:

- a directed acyclic graph with nodes labeled with random variables
- a domain for each random variable
- a set of conditional probability tables for each variable given its parents (including prior probabilities for nodes with no parents).

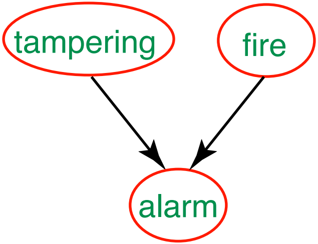
Summary: Components of a belief network

A belief network consists of:

- a directed acyclic graph with nodes labeled with random variables
- a domain for each random variable
- a set of conditional probability tables for each variable given its parents (including prior probabilities for nodes with no parents).

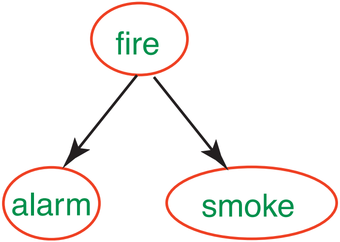
- The parents of a node n are those variables on which n directly depends.
- A belief network is a graphical representation of dependence and independence:
  - A variable is independent of its non-descendants given its parents.

Common descendants



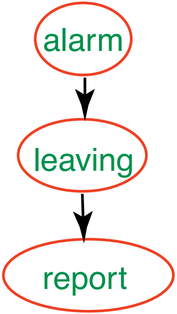
- *tampering* and *fire* are independent
- *tampering* and *fire* are dependent given *alarm*
- Intuitively, *tampering* can explain away *fire*

Common ancestors



- *alarm* and *smoke* are dependent
- *alarm* and *smoke* are independent given *fire*
- Intuitively, *fire* can explain *alarm* and *smoke*; learning one can affect the other by changing your belief in *fire*.

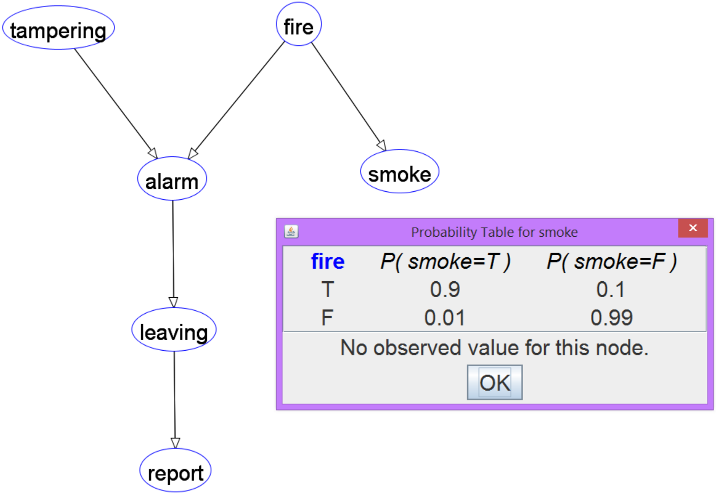
Chain



- *alarm* and *report* are dependent
- *alarm* and *report* are independent given *leaving*
- Intuitively, the only way that the *alarm* affects *report* is by affecting *leaving*.

Using belief networks

- Record and Compute conditional probabilities



fire	P(smoke=T)	P(smoke=F)
T	0.9	0.1
F	0.01	0.99

Using belief networks

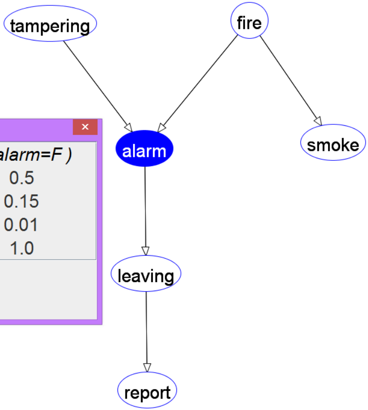
- Record and Compute conditional probabilities

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.

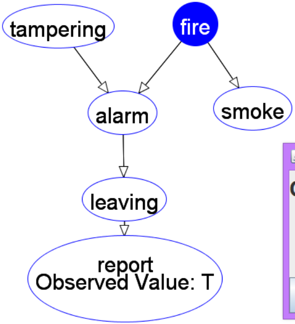
OK

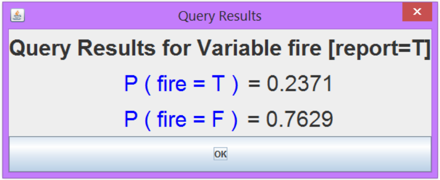


Using belief networks

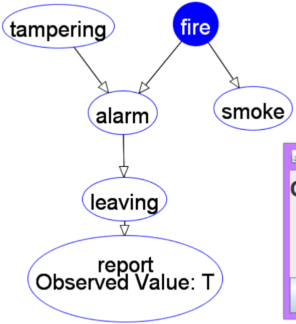
- Record and Compute conditional probabilities
- Make observations and compute posterior, or Bayesian, probabilities

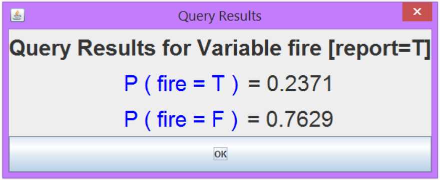
Demo on the board: posterior probabilities





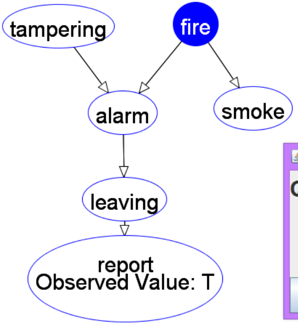
Demo on the board: posterior probabilities

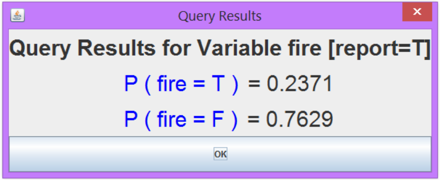




We revised our belief: the prior probability of "Fire" was 0,01!

Demo on the board: posterior probabilities

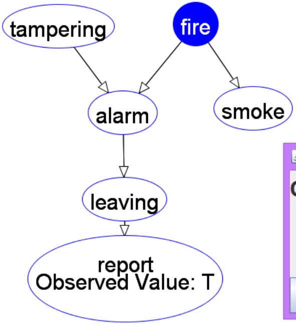


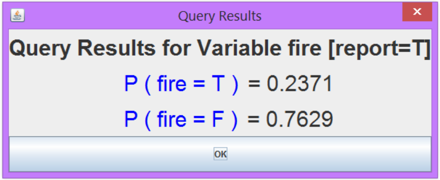


We revised our belief: the prior probability of “Fire” was 0,01!

It is in essence our last lecture Bayes learning exercise, but with more complex connections among variables.

Demo on the board: posterior probabilities





We revised our belief: the prior probability of "Fire" was 0,01!
It is in essence our last lecture Bayes learning exercise, but with more complex connections among variables. Now lets learn the formal side.

Using belief networks

- Record and Compute conditional probabilities
- Make observations and compute posterior, or Bayesian, probabilities
- If you observe variable $\overline{Y}$, the variables whose posterior probability is different from their prior are:
  - The ancestors of $\overline{Y}$ and
  - their descendants.

Lets define an algorithm to do that:

Variable Elimination Algorithm

Factors

A factor is a representation of a function from a tuple of random variables into a number.
We will write factor f on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$.

Factors

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

We will write factor f on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$.

We can assign some or all of the variables of a factor:

- $f(X_1{=}v_1, X_2, \ldots, X_j)$, where $v_1 \in dom(X_1)$, is a factor on $X_2, \ldots, X_j$.
- $f(X_1{=}v_1, X_2{=}v_2, \ldots, X_j{=}v_j)$ is a number that is the value of f when each X_i has value v_i.

The former is also written as $f(X_1, X_2, \ldots, X_j)_{X_1=v_1}$, etc.

Example factors

$r(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$r(X{=}t, Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

Example factors

$r(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$r(X{=}t, Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$r(X{=}t, Y, Z{=}f)$:

Example factors

$r(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$r(X{=}t, Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$r(X{=}t, Y, Z{=}f)$:

Y	val
t	
f	

$r(X{=}t, Y{=}f, Z{=}f) =$

Example factors

$r(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$r(X=t, Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$r(X=t, Y, Z=f)$:

Y	val
t	0.9
f	0.8

$r(X=t, Y=f, Z=f) =$ 0,8

Multiplying factors

The product of factor $f_1(\overline{X}, \overline{Y})$ and $f_2(\overline{Y}, \overline{Z})$, where $\overline{Y}$ are the variables in common, is the factor $(f_1 \times f_2)(\overline{X}, \overline{Y}, \overline{Z})$ defined by:

$$(f_1 \times f_2)(\overline{X}, \overline{Y}, \overline{Z}) \quad = \quad f_1(\overline{X}, \overline{Y}) f_2(\overline{Y}, \overline{Z}).$$

Multiplying factors example

f_1:

A	B	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

B	C	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1 \times f_2$:

A	B	C	val
t	t	t	0.03
t	t	f	
t	f	t	
t	f	f	
f	t	t	
f	t	f	
f	f	t	
f	f	f	

Multiplying factors example

f_1:

A	B	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

B	C	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1 \times f_2$:

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

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:

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

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	

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

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 $\sum_Z f(Z)$.

Variable elimination algorithm

The task:

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

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

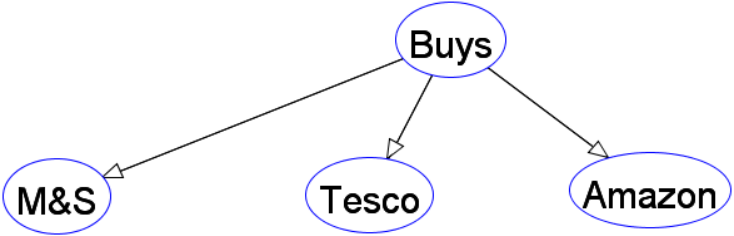
1. Construct a factor for each conditional probability.
2. Set the observed variables to their observed values.
3. Sum out each of the other variables (the $\{Z_1, \ldots, Z_k\}$) according to some elimination ordering.
4. Multiply the remaining factors. Normalize by dividing the resulting factor $f(Z)$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

In Test 1, Part 2 you will be tracing execution of this algorithm, only item (3) will be redundant, since you will have all variables observed, and there will be no $\{Z_1, \ldots, Z_k\}$ to sum out.

Naive Bayes Net...

is just a special kind of a Bayes Net

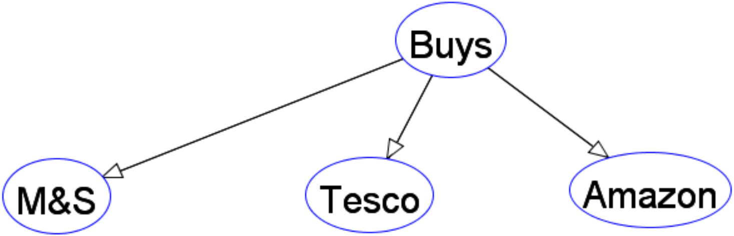
- has the single node – the class – on which all variables depend
- All other variables are independent of each other given the class
- Looks as follows (much simplified version of my shopping Example from Lecture 1)



Naive Bayes Net...

is just a special kind of a Bayes Net

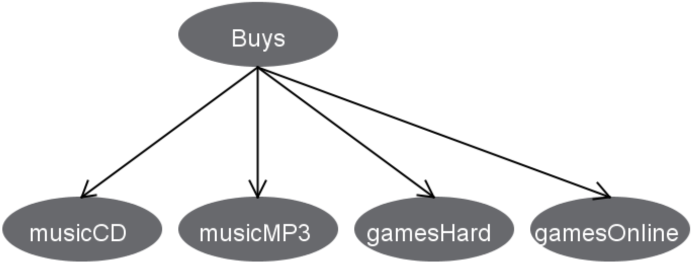
- has the single node – the class – on which all variables depend
- All other variables are independent of each other given the class
- Looks as follows (much simplified version of my shopping Example from Lecture 1)



- Your Test1 data set fits naturally to this scheme: You have one class and four variables given by column names.

Last lecture example

Loading the data set of last lecture example to Weka generates the following Naive Bayes Net:



How about other network architectures?

A research question

What other architectures may suit for classifying your data set? Try generating those manually and automatically in Weka. Consult pages 261 – 270 of the textbook Data Mining (Witten et al.) (2011 Edition). (pp. 339 – 349 in 2017 Edition)

Test1, Part2, Probabilities from Data sets

- In DATA MINING textbook (2011, Witten et al), read §4.2 p. 90-94 (in 2017 edition, p. 96-100). It shows how to convert a data set into Bayes factors, which constitute a Bayes net
- Take the (small) facial recognition set from last lecture
- For it, produce the same Table with Counts and Probabilities as on p.91: it defines a Naive Bayes net for the data set. In your table, the order of columns should be: Cell33, Cell42, Cell48, Cell58, Emotion.
- Take a test example (observation of a new picture)

  $$(Cell33 = White), (Cell42 = Black), (Cell48 = Black), (Cell58 = White)$$

- Use your Table with Counts and Probabilities and the formulae given on pp.92-93 to compute the likelihood of a face being "Happy" and "Sad" given the observation.
- Use the computed likelihoods to compute probabilities for

  $$P(Happy|Cell33 = White, Cell42 = Black, Cell48 = Black, Cell58 = White)$$

  and

  $$P(Sad|Cell33 = White, Cell42 = Black, Cell48 = Black, Cell58 = White)$$

Test 1, Part 3 – Bayes nets in Weka

1. Load the given data set to Weka, run the Naive Bayes classifier on it. Use the option "Use training set"
   Compare the table that Weka gives as a result with your table with counts and probabilities. Analyse the differences: Are there any? what are they? and why they occur?
2. Using the same data set, run the BayesNet algorithm, with the settings:
   K2 algorithm is used to learn the network architecture. In the algorithm, the maximum number of parents is set to 1.
   `SimpleEstimator` should be used for estimating the conditional probability tables of a Bayes network once the structure has been learned.
   These settings are described by the following Weka Command:
   ```
   weka.classifiers.bayes.BayesNet -D -Q
   weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES
   -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A
   0.5
   ```
   Visualise the resulting Bayes network, be ready to answer questions about its architecture.

Test 1, Part 3 – Bayes nets in Weka

3 Using the same data set, run the BayesNet algorithm, with the settings:
TAN algorithm is used to learn the network architecture.
SimpleEstimator should be used for estimating the conditional probability tables of a Bayes network once the structure has been learned.
These settings are described by the following Weka Command:
```
weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.TAN -- -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A
0.5
```
Visualise the resulting Bayes network, be ready to answer questions about its architecture.

4 Using the Bayes Net created by the last algorithm (TAN), run a test to find a prediction for the emotion of a picture where

$$Cell33 = White, Cell42 = Black, Cell48 = Black, Cell58 = White.$$

Compare this with Naive Bayes probabilities that you computed using your table with counts and probabilities.

Test 1 overview

- Part 1, Q1-4: pen and pencil computations of conditional and Bayesian probabilities. (Simple)
- Part 2, Q5-10: pen and pencil: inferring Bayes net factorisation from a Data set table, using Variable Elimination algorithm to compute posterior probabilities after the observation is made. (Harder, mainly needs care and attention to detail)
- Part 3,Q11-14: Weka: load the data set, practice Bayesian learning with Weka's algorithms. (Mixed difficulty)

Essential reading

- Check relevant chapters in the recommended textbook (2011, Witten et al):
  1. Probabilities and Bayes factors from Data sets: §4.2 pp. 90-94 (in 2017 edition, p. 96-100)
  2. Bayes Nets: §9.2 pp.261-273 (in 2017 edition, p. 339-349)
  3. How to tune and use Weka for Bayesian learning: §11.4 pp.451-454. (in 2017 edition, §2.4.1 in (`https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf`))
- After that, you will be ready to start first half of the Coursework 2 (the part on Bayes nets).

Further reading

... One topic round the corner that we will not consider in the lectures, but you may study yourself:

- Markov Chains
  - Similarly to Naive Bayes – just a special case of a Bayes net
  - So, we almost know about Markov Chains, already
  - Good time to look them up!