Representations of Conditional Probabilities

There are many representations of conditional probabilities and factors:

Representations of Conditional Probabilities

There are many representations of conditional probabilities and factors:

- Tables
- Decision Trees
- Deterministic System with Noisy Inputs
  - Weighted Logical Formulae
  - Probabilistic Programs
- Noisy-or
- Logistic Function
- Neural Networks

Tabular Representation

$P(D \mid A, B, C)$:

A	B	C	D	Prob
true	true	true	true	0.9
true	true	true	false	0.1
true	true	false	true	0.9
true	true	false	false	0.1
true	false	true	true	0.2
true	false	true	false	0.8
true	false	false	true	0.2
true	false	false	false	0.8
false	true	true	true	0.3
false	true	true	false	0.7
false	true	false	true	0.4
false	true	false	false	0.6
false	false	true	true	0.3
false	false	true	false	0.7
false	false	false	true	0.4
false	false	false	false	0.6

Tabular Representation

$P(D \mid A, B, C)$:

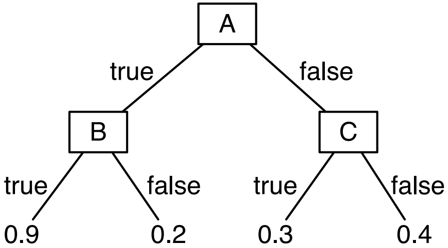
A	B	C	D	Prob
true	true	true	true	0.9
true	true	true	false	0.1
true	true	false	true	0.9
true	true	false	false	0.1
true	false	true	true	0.2
true	false	true	false	0.8
true	false	false	true	0.2
true	false	false	false	0.8
false	true	true	true	0.3
false	true	true	false	0.7
false	true	false	true	0.4
false	true	false	false	0.6
false	false	true	true	0.3
false	false	true	false	0.7
false	false	false	true	0.4
false	false	false	false	0.6

Python:

```
pd = [[[[0.6,0.4],[0.7,0.3]],[[0.6,0.4],[0.7,0.3]]],
      [[[0.8,0.2], [0.8,0.2]],[[0.1,0.9], [0.1,0.9]]]]
pd[1][0][1][0]
```

Decision Tree Representation

$$P(d \mid A, B, C)$$



Deterministic System with Noisy Inputs: Weighted Logical Formulae

$$d \leftrightarrow ((a \wedge b \wedge n_0) \\ \vee (a \wedge \neg b \wedge n_1) \\ \vee (\neg a \wedge c \wedge n_2) \\ \vee (\neg a \wedge \neg c \wedge n_3))$$

n_i are independent:

$$P(n_0) = 0.9$$
$$P(n_1) = 0.2$$
$$P(n_2) = 0.3$$
$$P(n_3) = 0.4$$

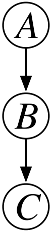
Deterministic System with Noisy Inputs: Probabilistic Programming

Probabilistic Program: a program with randomized inputs

```
if a:
     if b:
         d = flip(0.9)
     else:
         d = flip(0.2)
else:
    if c:
         d = flip(0.3)
     else:
         d = flip(0.4)
```

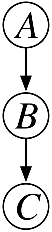
where $flip(p) = (random() < p)$

Representing Belief Networks



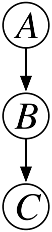
$$
\begin{aligned}
P(a) &= 0.1 \\
P(b \mid a) &= 0.8 \\
P(b \mid \neg a) &= 0.3 \\
P(c \mid b) &= 0.4 \\
P(c \mid \neg b) &= 0.75
\end{aligned}
$$

Representing Belief Networks



$$
\begin{aligned}
P(a) &= 0.1 \\
P(b \mid a) &= 0.8 \\
P(b \mid \neg a) &= 0.3 \\
P(c \mid b) &= 0.4 \\
P(c \mid \neg b) &= 0.75
\end{aligned}
$$

$P(a) = 0.1,$
$P(bifa) = 0.8,\ P(bifna) = 0.3,$
$P(cifb) = 0.4,\ P(cifnb) = 0.75.$
$b \leftrightarrow (a \wedge bifa) \vee (\neg a \wedge bifna)$
$c \leftrightarrow (b \wedge cifb) \vee (\neg b \wedge cifnbc)$



$$
\begin{aligned}
P(a) &= 0.1 \\
P(b \mid a) &= 0.8 \\
P(b \mid \neg a) &= 0.3 \\
P(c \mid b) &= 0.4 \\
P(c \mid \neg b) &= 0.75
\end{aligned}
$$

$P(a) = 0.1$,
$P(bifa) = 0.8$, $P(bifna) = 0.3$,
$P(cifb) = 0.4$, $P(cifnb) = 0.75$.
$b \leftrightarrow (a \wedge bifa) \vee (\neg a \wedge bifna)$
$c \leftrightarrow (b \wedge cifb) \vee (\neg b \wedge cifnbc)$

```
a = flip(0.1)
if a:
        b = flip(0.8)
else:
        b = flip(0.3)
if b:
        c = flip(0.4)
else:
        c = flip(0.75)
```

$flip(p) = (random() < p)$

Noisy-or

- The robot is wet if it gets wet from rain or coffee or sprinkler or another reason. They each have a probability of making the robot wet $\longrightarrow$ noisy-or.

Noisy-or

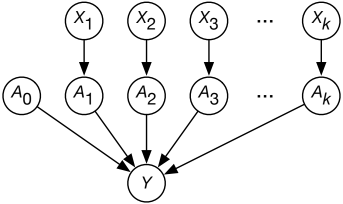
- The robot is wet if it gets wet from rain or coffee or sprinkler or another reason. They each have a probability of making the robot wet $\longrightarrow$ noisy-or.
- $P(Y \mid X_1, X_2, \ldots, X_k)$, with all variables Boolean, is defined using:

$$y \leftrightarrow n_0 \vee (n_1 \wedge x_1) \vee \cdots \vee (n_k \wedge x_k).$$

where n_i are unconditionally independent noise variables, with $P(n_i) = w_i$, and x_i means $X_i{=}true$.

Noisy-or — alternative definition

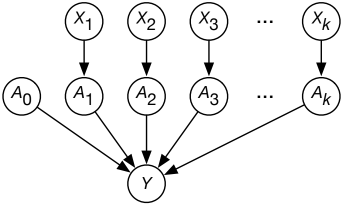
Noisy-or $P(Y \mid X_1, X_2, \ldots, X_k)$ can be defined using $k+1$ Boolean variables $A_0, A_1, \ldots, A_k$, where for each $i > 0$, A_i has X_i as its only parent.



$$P(A_0) = w_0$$

Noisy-or — alternative definition

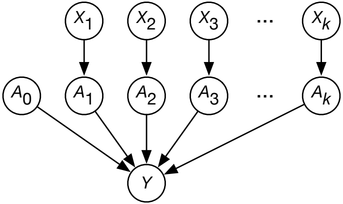
Noisy-or $P(Y \mid X_1, X_2, \ldots, X_k)$ can be defined using $k+1$ Boolean variables $A_0, A_1, \ldots, A_k$, where for each $i > 0$, A_i has X_i as its only parent.



$$P(A_0) = w_0$$
$$P(A_i{=}true \mid X_i{=}true) = w_i \text{ for } i > 0$$
$$P(A_i{=}true \mid X_i{=}false) = 0 \text{ for } i > 0$$

Noisy-or — alternative definition

Noisy-or $P(Y \mid X_1, X_2, \ldots, X_k)$ can be defined using $k+1$ Boolean variables $A_0, A_1, \ldots, A_k$, where for each $i > 0$, A_i has X_i as its only parent.



$$P(A_0) = w_0$$
$$P(A_i{=}true \mid X_i{=}true) = w_i \text{ for } i > 0$$
$$P(A_i{=}true \mid X_i{=}false) = 0 \text{ for } i > 0$$
$$P(Y \mid A_0, A_1, \ldots, A_k) = \begin{cases} 1 & \text{if } \exists i \; A_i \text{ is true} \\ 0 & \text{if } \forall i \; A_i \text{ is false} \end{cases}$$

Noisy-or: Example

- Suppose the robot could get wet from rain or coffee.
- There is a probability that it gets wet from rain if it rains, and a probability that it gets wet from coffee if it has coffee, and a probability that it gets wet for other reasons.

Noisy-or: Example

- Suppose the robot could get wet from rain or coffee.
- There is a probability that it gets wet from rain if it rains, and a probability that it gets wet from coffee if it has coffee, and a probability that it gets wet for other reasons.
- We could have:
  $P(wet\_from\_rain \mid rain) = 0.3$,
  $P(wet\_from\_coffee \mid coffee) = 0.2$
  $P(wet\_for\_other\_reasons) = 0.1$.

Noisy-or: Example

- Suppose the robot could get wet from rain or coffee.
- There is a probability that it gets wet from rain if it rains, and a probability that it gets wet from coffee if it has coffee, and a probability that it gets wet for other reasons.
- We could have:
  $P(wet\_from\_rain \mid rain) = 0.3$,
  $P(wet\_from\_coffee \mid coffee) = 0.2$
  $P(wet\_for\_other\_reasons) = 0.1$.
- The robot is wet if it wet from rain, wet from coffee, or wet for other reasons.

$$wet \leftrightarrow wet\_from\_rain \vee wet\_from\_coffe \vee wet\_for\_other\_reasons$$

Logistic Functions

$$P(h \mid e) = \frac{P(h \wedge e)}{P(e)}$$

Logistic Functions

$$
\begin{aligned}
P(h \mid e) &= \frac{P(h \wedge e)}{P(e)} \\
&= \frac{P(h \wedge e)}{P(h \wedge e) + P(\neg h \wedge e)}
\end{aligned}
$$

Logistic Functions

$$
\begin{aligned}
P(h \mid e) &= \frac{P(h \wedge e)}{P(e)} \\
&= \frac{P(h \wedge e)}{P(h \wedge e) + P(\neg h \wedge e)} \\
&= \frac{1}{1 + P(\neg h \wedge e)/P(h \wedge e)}
\end{aligned}
$$

Logistic Functions

$$
\begin{aligned}
P(h \mid e) &= \frac{P(h \wedge e)}{P(e)} \\
&= \frac{P(h \wedge e)}{P(h \wedge e) + P(\neg h \wedge e)} \\
&= \frac{1}{1 + P(\neg h \wedge e)/P(h \wedge e)} \\
&= \frac{1}{1 + e^{-\log P(h \wedge e)/P(\neg h \wedge e)}}
\end{aligned}
$$

Logistic Functions

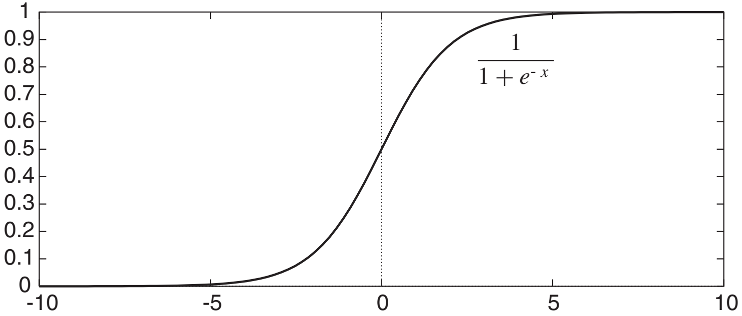
$$
\begin{aligned}
P(h \mid e) &= \frac{P(h \wedge e)}{P(e)} \\
&= \frac{P(h \wedge e)}{P(h \wedge e) + P(\neg h \wedge e)} \\
&= \frac{1}{1 + P(\neg h \wedge e)/P(h \wedge e)} \\
&= \frac{1}{1 + e^{-\log P(h \wedge e)/P(\neg h \wedge e)}} \\
&= sigmoid(\log odds(h \mid e))
\end{aligned}
$$

$$
sigmoid(x) = \frac{1}{1 + e^{-x}}
$$

$$
odds(h \mid e) = \frac{P(h \wedge e)}{P(\neg h \wedge e)}
$$

Logistic Functions

A conditional probability is the sigmoid of the log-odds.

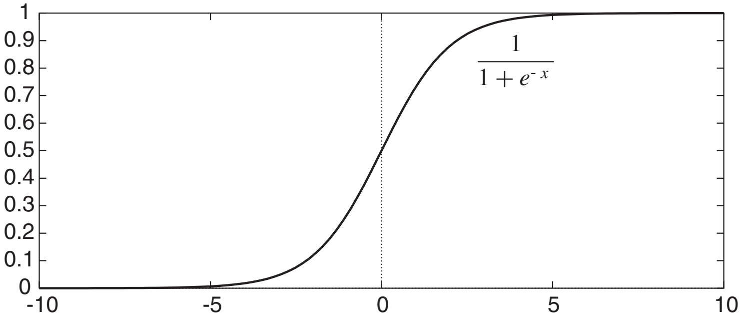


$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

A logistic function is the sigmoid of a linear function.

Logistic Functions

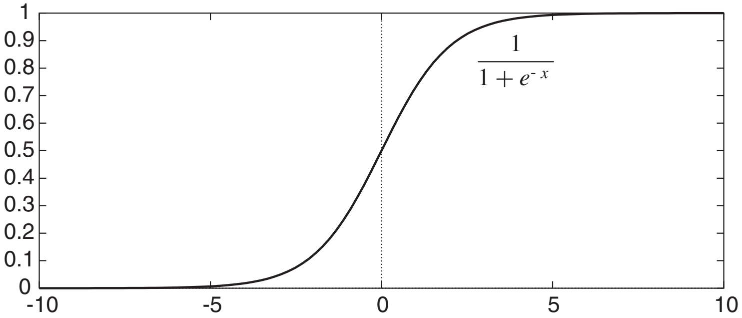
A conditional probability is the sigmoid of the log-odds.



$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

A logistic function is the sigmoid of a linear function.
If the odds form a product, the log odds give a sum.

A conditional probability is the sigmoid of the log-odds.



$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

A logistic function is the sigmoid of a linear function.
If the odds form a product, the log odds give a sum.
What independence is assumed?

Odds

- $odds(h \mid e) = \dfrac{P(h \wedge e)}{P(\neg h \wedge e)}$

Odds

- $odds(h \mid e) = \dfrac{P(h \wedge e)}{P(\neg h \wedge e)} = \dfrac{P(h \mid e)}{P(\neg h \mid e)}$

Odds

- $odds(h \mid e) = \frac{P(h \wedge e)}{P(\neg h \wedge e)} = \frac{P(h \mid e)}{P(\neg h \mid e)} = \frac{P(h \mid e)}{1 - P(h \mid e)}$

Odds

- $odds(h \mid e) = \dfrac{P(h \land e)}{P(\neg h \land e)} = \dfrac{P(h \mid e)}{P(\neg h \mid e)} = \dfrac{P(h \mid e)}{1 - P(h \mid e)}$
- Decomposing $P(h \land e)$ into $P(e \mid h) * P(h)$

$$odds(h \mid e) =$$

Odds

- $odds(h \mid e) = \dfrac{P(h \wedge e)}{P(\neg h \wedge e)} = \dfrac{P(h \mid e)}{P(\neg h \mid e)} = \dfrac{P(h \mid e)}{1 - P(h \mid e)}$
- Decomposing $P(h \wedge e)$ into $P(e \mid h) * P(h)$

$$odds(h \mid e) = \frac{P(e \mid h)}{P(e \mid \neg h)} * \frac{P(h)}{P(\neg h)}$$

- $\frac{P(h)}{P(\neg h)} = \frac{P(h)}{1 - P(h)}$ is the prior odds
- $\frac{P(e|h)}{P(e|\neg h)}$ is the likelihood ratio.

Odds

- $odds(h \mid e) = \frac{P(h \wedge e)}{P(\neg h \wedge e)} = \frac{P(h \mid e)}{P(\neg h \mid e)} = \frac{P(h \mid e)}{1 - P(h \mid e)}$
- Decomposing $P(h \wedge e)$ into $P(e \mid h) * P(h)$

$$odds(h \mid e) = \frac{P(e \mid h)}{P(e \mid \neg h)} * \frac{P(h)}{P(\neg h)}$$

- $\frac{P(h)}{P(\neg h)} = \frac{P(h)}{1-P(h)}$ is the prior odds
- $\frac{P(e|h)}{P(e|\neg h)}$ is the likelihood ratio.
- If $e = e_1 \wedge \cdots \wedge e_k$, and e_i & e_j are independent given h

$$\frac{P(e \mid h)}{P(e \mid \neg h)} =$$

Odds

- $odds(h \mid e) = \dfrac{P(h \wedge e)}{P(\neg h \wedge e)} = \dfrac{P(h \mid e)}{P(\neg h \mid e)} = \dfrac{P(h \mid e)}{1 - P(h \mid e)}$
- Decomposing $P(h \wedge e)$ into $P(e \mid h) * P(h)$

$$odds(h \mid e) = \frac{P(e \mid h)}{P(e \mid \neg h)} * \frac{P(h)}{P(\neg h)}$$

- $\frac{P(h)}{P(\neg h)} = \frac{P(h)}{1-P(h)}$ is the prior odds
- $\frac{P(e|h)}{P(e|\neg h)}$ is the likelihood ratio.
- If $e = e_1 \wedge \cdots \wedge e_k$, and e_i & e_j are independent given h

$$\frac{P(e \mid h)}{P(e \mid \neg h)} = \prod_{i=1}^{k} \frac{P(e_i \mid h)}{P(e_i \mid \neg h)}$$

Odds

- $odds(h \mid e) = \dfrac{P(h \wedge e)}{P(\neg h \wedge e)} = \dfrac{P(h \mid e)}{P(\neg h \mid e)} = \dfrac{P(h \mid e)}{1 - P(h \mid e)}$
- Decomposing $P(h \wedge e)$ into $P(e \mid h) * P(h)$

$$odds(h \mid e) = \frac{P(e \mid h)}{P(e \mid \neg h)} * \frac{P(h)}{P(\neg h)}$$

- $\frac{P(h)}{P(\neg h)} = \frac{P(h)}{1 - P(h)}$ is the prior odds
- $\frac{P(e|h)}{P(e|\neg h)}$ is the likelihood ratio.
- If $e = e_1 \wedge \cdots \wedge e_k$, and e_i & e_j are independent given h

$$\frac{P(e \mid h)}{P(e \mid \neg h)} = \prod_{i=1}^{k} \frac{P(e_i \mid h)}{P(e_i \mid \neg h)}$$

$$\log odds(h \mid e) =$$

Odds

- $odds(h \mid e) = \dfrac{P(h \wedge e)}{P(\neg h \wedge e)} = \dfrac{P(h \mid e)}{P(\neg h \mid e)} = \dfrac{P(h \mid e)}{1 - P(h \mid e)}$
- Decomposing $P(h \wedge e)$ into $P(e \mid h) * P(h)$

$$odds(h \mid e) = \frac{P(e \mid h)}{P(e \mid \neg h)} * \frac{P(h)}{P(\neg h)}$$

- $\frac{P(h)}{P(\neg h)} = \frac{P(h)}{1-P(h)}$ is the prior odds
- $\frac{P(e|h)}{P(e|\neg h)}$ is the likelihood ratio.
- If $e = e_1 \wedge \cdots \wedge e_k$, and e_i & e_j are independent given h

$$\frac{P(e \mid h)}{P(e \mid \neg h)} = \prod_{i=1}^{k} \frac{P(e_i \mid h)}{P(e_i \mid \neg h)}$$

$$\log odds(h \mid e) = \log \frac{P(h)}{P(\neg h)} + \sum_{i=1}^{k} \log \frac{P(e_i \mid h)}{P(e_i \mid \neg h)}$$

X_1	X_2	X_3	X_4	*Prob*
False	*False*	*False*	*False*	p_0=0.01
True	*False*	*False*	*False*	p_1=0.05
False	*True*	*False*	*False*	p_2=0.1
False	*False*	*True*	*False*	p_3=0.2
False	*False*	*False*	*True*	p_4=0.2

X_1	X_2	X_3	X_4	A	B
False	*False*	*True*	*True*	0.353535	0.860870
False	*True*	*False*	*True*	0.272727	0.733333
False	*True*	*True*	*True*	0.412305	0.985520
True	*False*	*False*	*True*	0.232323	0.565714
True	*False*	*True*	*True*	0.379655	0.969916
True	*True*	*False*	*False*	0.136364	0.366667
True	*True*	*False*	*True*	0.302112	0.934764
True	*True*	*True*	*True*	0.436050	0.997188

X_1	X_2	X_3	X_4	*Prob*
False	*False*	*False*	*False*	p_0=0.01
True	*False*	*False*	*False*	p_1=0.05
False	*True*	*False*	*False*	p_2=0.1
False	*False*	*True*	*False*	p_3=0.2
False	*False*	*False*	*True*	p_4=0.2

X_1	X_2	X_3	X_4	A	B
False	*False*	*True*	*True*	0.353535	0.860870
False	*True*	*False*	*True*	0.272727	0.733333
False	*True*	*True*	*True*	0.412305	0.985520
True	*False*	*False*	*True*	0.232323	0.565714
True	*False*	*True*	*True*	0.379655	0.969916
True	*True*	*False*	*False*	0.136364	0.366667
True	*True*	*False*	*True*	0.302112	0.934764
True	*True*	*True*	*True*	0.436050	0.997188

A is noisy-or. *B* is logistic regression.

X_1	X_2	X_3	X_4	*Prob*
False	*False*	*False*	*False*	p_0=0.01
True	*False*	*False*	*False*	p_1=0.05
False	*True*	*False*	*False*	p_2=0.1
False	*False*	*True*	*False*	p_3=0.2
False	*False*	*False*	*True*	p_4=0.2

X_1	X_2	X_3	X_4	A	B
False	*False*	*True*	*True*	0.353535	0.860870
False	*True*	*False*	*True*	0.272727	0.733333
False	*True*	*True*	*True*	0.412305	0.985520
True	*False*	*False*	*True*	0.232323	0.565714
True	*False*	*True*	*True*	0.379655	0.969916
True	*True*	*False*	*False*	0.136364	0.366667
True	*True*	*False*	*True*	0.302112	0.934764
True	*True*	*True*	*True*	0.436050	0.997188

A is noisy-or. *B* is logistic regression.

What if p_0 is increased to 0.05 with $p_1 \dots p_4$ fixed?

Logistic Representation of Conditional Probability

Can the running example be represented using a logistic function?

Logistic Representation of Conditional Probability

Can the running example be represented using a logistic function?

$$\begin{aligned} P(d \mid A, B, C) = \mathit{sigmoid}(&\mathit{logit}(0.9) * A * B \\ &+ \mathit{logit}(0.2) * A * (1 - B) \\ &+ \mathit{logit}(0.3) * (1 - A) * C \\ &+ \mathit{logit}(0.4) * (1 - A) * (1 - C)) \end{aligned}$$

where *logit* is the inverse is *sigmoid*.

Logistic Representation of Conditional Probability

Can the running example be represented using a logistic function?

$$\begin{aligned} P(d \mid A, B, C) = sigmoid(&logit(0.9) * A * B \\ &+ logit(0.2) * A * (1 - B) \\ &+ logit(0.3) * (1 - A) * C \\ &+ logit(0.4) * (1 - A) * (1 - C)) \end{aligned}$$

where *logit* is the inverse is *sigmoid*.

$$\begin{aligned} P(d \mid A, B, C) = sigmoid(&logit(0.4) \\ &+ (logit(0.2) - logit(0.4) * A \\ &+ (logit(0.9) - logit(0.2)) * A * B \\ &+ \ldots \end{aligned}$$

Logistic Representation of Conditional Probability

Can the running example be represented using a logistic function?

$$
\begin{aligned}
P(d \mid A, B, C) = \mathit{sigmoid}(&\mathit{logit}(0.9) * A * B \\
&+ \mathit{logit}(0.2) * A * (1 - B) \\
&+ \mathit{logit}(0.3) * (1 - A) * C \\
&+ \mathit{logit}(0.4) * (1 - A) * (1 - C))
\end{aligned}
$$

where *logit* is the inverse is *sigmoid*.

$$
\begin{aligned}
P(d \mid A, B, C) = \mathit{sigmoid}(&\mathit{logit}(0.4) \\
&+ (\mathit{logit}(0.2) - \mathit{logit}(0.4) * A \\
&+ (\mathit{logit}(0.9) - \mathit{logit}(0.2)) * A * B \\
&+ \dots
\end{aligned}
$$

– Allowing products in the features is canonical representation, which can represent any discrete conditional probability.
– This is the representation learned by gradient boosted trees.

Neural Network

- Build a neural network to predict D from A, B, C

Neural Network

- Build a neural network to predict D from A, B, C
- For Boolean outputs, typically use a sigmoid activation at the output, and optimize with log likelihood.

Neural Network

- Build a neural network to predict D from A, B, C
- For Boolean outputs, typically use a sigmoid activation at the output, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features

Neural Network

- Build a neural network to predict D from A, B, C
- For Boolean outputs, typically use a sigmoid activation at the output, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features
- For other discrete variables,

Neural Network

- Build a neural network to predict D from A, B, C
- For Boolean outputs, typically use a sigmoid activation at the output, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features
- For other discrete variables, softmax gives a probability distribution:

$$softmax((\alpha_1 \dots \alpha_k))_i = \frac{exp(\alpha_i)}{\sum_{j=1}^{k} exp(\alpha_j)}$$

Neural Network

- Build a neural network to predict D from A, B, C
- For Boolean outputs, typically use a sigmoid activation at the output, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features
- For other discrete variables, softmax gives a probability distribution:

$$softmax((\alpha_1 \dots \alpha_k))_i = \frac{exp(\alpha_i)}{\sum_{j=1}^{k} exp(\alpha_j)}$$

- For other domains, a Bayesian neural network represents the distribution over the outputs (not just a point prediction).

Sigmoid and Softmax

- How are sigmoid and softmax related?

Sigmoid and Softmax

- How are sigmoid and softmax related?

$$sigmoid(x) = \frac{1}{exp(-x) + 1}$$

Sigmoid and Softmax

- How are sigmoid and softmax related?

$$
\begin{aligned}
sigmoid(x) &= \frac{1}{exp(-x) + 1} \\
&= \frac{exp(x)}{exp(0) + exp(x)}
\end{aligned}
$$

Sigmoid and Softmax

- How are sigmoid and softmax related?

$$\begin{aligned} sigmoid(x) &= \frac{1}{exp(-x)+1} \\ &= \frac{exp(x)}{exp(0)+exp(x)} \\ &= softmax((0,x))_2 \end{aligned}$$

$(0, x)$ corresponds to the values $(false, true)$
$softmax((0, x))_2$ is the second component of the result

Sigmoid and Softmax

- How are sigmoid and softmax related?

$$\begin{aligned} sigmoid(x) &= \frac{1}{exp(-x)+1} \\ &= \frac{exp(x)}{exp(0)+exp(x)} \\ &= softmax((0,x))_2 \end{aligned}$$

  $(0, x)$ corresponds to the values $(false, true)$
  $softmax((0,x))_2$ is the second component of the result
- What happens in a softmax if a constant is added to each component?

Sigmoid and Softmax

- How are sigmoid and softmax related?

$$\begin{aligned} \mathit{sigmoid}(x) &= \frac{1}{\mathit{exp}(-x) + 1} \\ &= \frac{\mathit{exp}(x)}{\mathit{exp}(0) + \mathit{exp}(x)} \\ &= \mathit{softmax}((0, x))_2 \end{aligned}$$

  $(0, x)$ corresponds to the values $(\mathit{false}, \mathit{true})$
  $\mathit{softmax}((0, x))_2$ is the second component of the result
- What happens in a softmax if a constant is added to each component?
- How can you avoid overflow or underflow in softmax?
  ($\mathit{exp}(750)$ will overflow for most modern CPUs, and $\mathit{exp}(-750)$ results in zero.)

Sigmoid and Softmax

- How are sigmoid and softmax related?

$$sigmoid(x) = \frac{1}{exp(-x)+1} = \frac{exp(x)}{exp(0)+exp(x)} = softmax((0,x))_2$$

  $(0, x)$ corresponds to the values $(false, true)$
  $softmax((0,x))_2$ is the second component of the result
- What happens in a softmax if a constant is added to each component?
- How can you avoid overflow or underflow in softmax? ($exp(750)$ will overflow for most modern CPUs, and $exp(-750)$ results in zero.)
- Why not use same "trick" as sigmoid – setting one value to 0 – to reduce the number of parameters in softmax?