Understanding Independence: Common ancestors

- *alarm* and *smoke* are dependent.

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

Diagram:

```
fire
  /\  
alarm smoke
```
Understanding Independence: Common ancestors

- alarm and smoke are dependent

Diagram:
- fire
  - alarm
  - smoke
Intuitively, fire can explain alarm and smoke; learning one can affect the other by changing your belief in fire.

- alarm and smoke are dependent
- alarm and smoke are given fire
Understanding Independence: Common ancestors

- *alarm* and *smoke* are dependent
- *alarm* and *smoke* are independent given *fire*
Understanding Independence: 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.
Understanding Independence: Chain

- *alarm* and *report* are dependent

Intuitively, the only way that the *alarm* affects *report* is by affecting *leaving*.
Understanding Independence: Chain

- *alarm* and *report* are dependent

Intuitively, the only way that the *alarm* affects *report* is by affecting *leaving*. 

© D.L. Poole and A.K. Mackworth 2010-2020
Understanding Independence: Chain

- alarm and report are dependent
- alarm and report are dependent given leaving
Understanding Independence: Chain

- alarm and report are dependent
- alarm and report are independent given leaving
Understanding Independence: 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**.
tampering and fire are independent given alarm. Intuitively, tampering can explain away fire.
Understanding Independence: Common descendants

- tampering and fire are independent

- Intuitively, tampering can explain away fire
tampering and fire are independent

- tampering and fire are given alarm
Understanding Independence: Common descendants

- tampering and fire are independent
- tampering and fire are dependent given alarm
Understanding Independence: Common descendants

- tampering and fire are independent
- tampering and fire are dependent given alarm
- Intuitively, tampering can explain away fire
Understanding independence: example
1. On which given probabilities does $P(N)$ depend?
1. On which given probabilities does $P(N)$ depend?
2. If you were to observe a value for $B$, which variables’ probabilities will change?
Understanding independence: questions

1. On which given probabilities does $P(N)$ depend?
2. If you were to observe a value for $B$, which variables’ probabilities will change?
3. If you were to observe a value for $N$, which variables’ probabilities will change?
Understanding independence: questions

1. On which given probabilities does $P(N)$ depend?
2. If you were to observe a value for $B$, which variables’ probabilities will change?
3. If you were to observe a value for $N$, which variables’ probabilities will change?
4. Suppose you had observed a value for $M$; if you were to then observe a value for $N$, which variables’ probabilities will change?
Understanding independence: questions

1. On which given probabilities does $P(N)$ depend?
2. If you were to observe a value for $B$, which variables’ probabilities will change?
3. If you were to observe a value for $N$, which variables’ probabilities will change?
4. Suppose you had observed a value for $M$; if you were to then observe a value for $N$, which variables’ probabilities will change?
5. Suppose you had observed $B$ and $Q$; which variables’ probabilities will change when you observe $N$?
What variables are affected by observing?

If you observe variable(s) $\overline{Y}$, the variables whose posterior probability is different from their prior are:

- The ancestors of $\overline{Y}$ and
- their descendants.

Intuitively (if you have a causal belief network):

- You do abduction to possible causes and
- prediction from the causes.
d-separation

- A connection is a meeting of arcs in a belief network. A connection is open is defined as follows:
  - If there are arcs $A \rightarrow B$ and $B \rightarrow C$ such that $B \notin Z$, then the connection at $B$ between $A$ and $C$ is open.
  - If there are arcs $B \rightarrow A$ and $B \rightarrow C$ such that $B \notin Z$, then the connection at $B$ between $A$ and $C$ is open.
  - If there are arcs $A \rightarrow B$ and $C \rightarrow B$ such that $B$ (or a descendent of $B$) is in $Z$, then the connection at $B$ between $A$ and $C$ is open.

$x$ is d-connected from $y$ given $z$ if there is a path from $x$ to $y$, along open connections.

$x$ is d-separated from $y$ given $z$ if it is not d-connected.

$x$ is independent of $y$ given $z$ for all conditional probabilities iff $x$ is d-separated from $y$ given $z$.
A connection is a meeting of arcs in a belief network. A connection is open is defined as follows:

- If there are arcs \( A \rightarrow B \) and \( B \rightarrow C \) such that \( B \notin \overline{Z} \), then the connection at \( B \) between \( A \) and \( C \) is open.
- If there are arcs \( B \rightarrow A \) and \( B \rightarrow C \) such that \( B \notin \overline{Z} \), then the connection at \( B \) between \( A \) and \( C \) is open.
- If there are arcs \( A \rightarrow B \) and \( C \rightarrow B \) such that \( B \) (or a descendent of \( B \)) is in \( \overline{Z} \), then the connection at \( B \) between \( A \) and \( C \) is open.

\( X \) is d-connected from \( Y \) given \( \overline{Z} \) if there is a path from \( X \) to \( Y \), along open connections.

\( X \) is d-separated from \( Y \) given \( \overline{Z} \) if it is not d-connected.
A connection is a meeting of arcs in a belief network. A connection is open is defined as follows:

- If there are arcs $A \rightarrow B$ and $B \rightarrow C$ such that $B \notin \overline{Z}$, then the connection at $B$ between $A$ and $C$ is open.
- If there are arcs $B \rightarrow A$ and $B \rightarrow C$ such that $B \notin \overline{Z}$, then the connection at $B$ between $A$ and $C$ is open.
- If there are arcs $A \rightarrow B$ and $C \rightarrow B$ such that $B$ (or a descendent of $B$) is in $\overline{Z}$, then the connection at $B$ between $A$ and $C$ is open.

$X$ is d-connected from $Y$ given $\overline{Z}$ if there is a path from $X$ to $Y$, along open connections.

$X$ is d-separated from $Y$ given $\overline{Z}$ if it is not d-connected.

$\overline{X}$ is independent $\overline{Y}$ given $\overline{Z}$ for all conditional probabilities iff $\overline{X}$ is d-separated from $\overline{Y}$ given $\overline{Z}$.
A Markov random field is composed of

- of a set of random variables: $X = \{X_1, X_2, \ldots, X_n\}$ and
- a set of factors $\{f_1, \ldots, f_m\}$, where a factor is a non-negative function of a subset of the variables.
A Markov random field is composed of
- a set of random variables: \( X = \{ X_1, X_2, \ldots, X_n \} \) and
- a set of factors \( \{ f_1, \ldots, f_m \} \), where a factor is a non-negative function of a subset of the variables.

and defines a joint probability distribution:

\[
P(X = x) \propto \prod_k f_k(X_k = x_k).
\]

\( Z \) is a normalization constant known as the partition function.
A Markov random field is composed of

- of a set of random variables: \( X = \{ X_1, X_2, \ldots, X_n \} \) and
- a set of factors \( \{ f_1, \ldots, f_m \} \), where a factor is a non-negative function of a subset of the variables.

and defines a joint probability distribution:

\[
P(X = x) \propto \prod_k f_k(X_k = x_k).
\]

\[
P(X = x) = \frac{1}{Z} \prod_k f_k(X_k = x_k).
\]

\[
Z = \sum_x \prod_k f_k(X_k = x_k)
\]

where \( f_k(X_k) \) is a factor on \( X_k \subseteq X \), and \( x_k \) is \( x \) projected onto \( X_k \).

\( Z \) is a normalization constant known as the partition function.
A Markov network is a graphical representation of a Markov random field where the nodes are the random variables and there is an arc between any two variables that are in a factor together.
A **Markov network** is a graphical representation of a Markov random field where the nodes are the random variables and there is an arc between any two variables that are in a factor together.

A **factor graph** is a bipartite graph, which contains a variable node for each random variable and a factor node for each factor. There is an edge between a variable node and a factor node if the variable appears in the factor.
A **Markov network** is a graphical representation of a Markov random field where the nodes are the random variables and there is an arc between any two variables that are in a factor together.

A **factor graph** is a bipartite graph, which contains a variable node for each random variable and a factor node for each factor. There is an edge between a variable node and a factor node if the variable appears in the factor.

A **belief network** is a
A Markov network is a graphical representation of a Markov random field where the nodes are the random variables and there is an arc between any two variables that are in a factor together.

A factor graph is a bipartite graph, which contains a variable node for each random variable and a factor node for each factor. There is an edge between a variable node and a factor node if the variable appears in the factor.

A belief network is a type of Markov random field where the factors represent conditional probabilities, there is a factor for each variable, and directed graph is acyclic.
Independence in a Markov Network

- The **Markov blanket** of a variable \( X \) is the set of variables that are in factors with \( X \).
- A variable is independent of the other variables given its Markov blanket.
Independence in a Markov Network

- The **Markov blanket** of a variable $X$ is the set of variables that are in factors with $X$.
- A variable is independent of the other variables given its Markov blanket.
- $X$ is **connected** to $Y$ given $\overline{Z}$ if there is a path from $X$ to $Y$ in the Markov network, which does not contain an element of $Z$.
- $X$ is **separated** from $Y$ given $\overline{Z}$ if it is not connected.
Independence in a Markov Network

- The **Markov blanket** of a variable $X$ is the set of variables that are in factors with $X$.

- A variable is independent of the other variables given its Markov blanket.

- $X$ is **connected** to $Y$ given $Z$ if there is a path from $X$ to $Y$ in the Markov network, which does not contain an element of $Z$.

- $X$ is **separated** from $Y$ given $Z$ if it is not connected.

- A **positive** distribution is one that does not contain zero probabilities.

- $X$ is independent $Y$ given $Z$ for all positive distributions iff $X$ is separated from $Y$ given $Z$. 
The parameters of a graphical model are the numbers that define the model.

A belief network is a canonical representation: given the structure and the distribution, the parameters are uniquely determined.

A Markov random field is not a canonical representation. Many different parameterizations result in the same distribution.
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
- Rules
- Weighted Logical Formulae
- Noisy-or
- Logistic Function
- Neural network
### Tabular Representation

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>0.9</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>0.1</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>0.9</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>0.1</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>0.2</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>0.8</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>0.2</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>0.8</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>0.3</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>0.7</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>0.4</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>0.6</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>0.3</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>0.7</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>0.4</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>0.6</td>
</tr>
</tbody>
</table>

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

\[ P(d \mid A, B, C) \]

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

A
  true
    B
      true 0.9
      false 0.2
    false
      C
  false
    true 0.3
    false 0.4
```
Rule Representation

0.9 : \( d \leftarrow a \land b \)
0.2 : \( d \leftarrow a \land \neg b \)
0.3 : \( d \leftarrow \neg a \land c \)
0.4 : \( d \leftarrow \neg a \land \neg c \)
Weighted Logical Formulae

\[ d \leftrightarrow ((a \land b \land n_0) \lor (a \land \neg b \land n_1) \lor (\neg a \land c \land n_2) \lor (\neg a \land \neg c \land n_3)) \]

\( n_i \) are independent:

\[ P(n_0) = 0.9 \]
\[ P(n_1) = 0.2 \]
\[ P(n_2) = 0.3 \]
\[ P(n_3) = 0.4 \]
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 \( \rightarrow \) 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 \( \rightarrow \) noisy-or.

\[ V_1 V_2 \ldots V_k \]

\[ A_1 A_2 \ldots A_k \]

\( X \)

\( \rightarrow k + 1 \) parameters \( p_0 \ldots p_k \).

Invent Boolean variables \( A_0, A_1, \ldots, A_k \), with probabilities \( P(A_0) = p_0 \) and for \( i > 0 \)

\[
P(A_i = \text{true} \mid V_i = \text{true}) = p_i
\]

\[
P(A_i = \text{true} \mid V_i = \text{false}) = 0
\]

\[
P(X \mid A_0, A_1, \ldots, A_k) = \begin{cases} 
  1 & \text{if } \exists i \ A_i \text{ is true} \\
  0 & \text{if } \forall i \ A_i \text{ is false}
\end{cases}
\]
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.
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.

\[ wet \leftrightarrow wet\_from\_rain \lor wet\_from\_coffe \lor wet\_for\_other\_reasons \]
Logistic Functions

\[ P(h \mid e) = \frac{P(h \land e)}{P(e)} \]
P(h | e) = \frac{P(h \land e)}{P(e)}
= \frac{P(h \land e)}{P(h \land e) + P(\neg h \land e)}
Logistic Functions

\[ P(h \mid e) = \frac{P(h \land e)}{P(e)} \]
\[ = \frac{P(h \land e)}{P(h \land e) + P(\neg h \land e)} \]
\[ = \frac{1}{1 + \frac{P(\neg h \land e)}{P(h \land e)}} \]

sigmoid \( (\log \text{odds}(h \mid e)) \) = \frac{1}{1 + e^{-x}}
Logistic Functions

\[ P(h \mid e) = \frac{P(h \land e)}{P(e)} \]

\[ = \frac{P(h \land e)}{P(h \land e) + P(\neg h \land e)} \]

\[ = \frac{1}{1 + \frac{P(\neg h \land e)}{P(h \land e)}} \]

\[ = \frac{1}{1 + e^{-\log P(h \land e)/P(\neg h \land e)}} \]
Logistic Functions

\[ P(h \mid e) = \frac{P(h \land e)}{P(e)} \]

\[ = \frac{P(h \land e)}{P(h \land e) + P(\neg h \land e)} \]

\[ = \frac{1}{1 + \frac{P(\neg h \land e)}{P(h \land e)}} \]

\[ = \frac{1}{1 + e^{-\log \frac{P(h \land e)}{P(\neg h \land e)}}} \]

\[ = \text{sigmoid}(\log \text{odds}(h \mid e)) \]

\[ \text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \]

\[ \text{odds}(h \mid e) = \frac{P(h \land e)}{P(\neg h \land e)} \]
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 Representation of Conditional Probability

\[ P(d \mid A, B, C) = \text{sigmoid}(0.9^\dagger \times A \times B) \]
\[ + 0.2^\dagger \times A \times (1 - B) \]
\[ + 0.3^\dagger \times (1 - A) \times C \]
\[ + 0.4^\dagger \times (1 - A) \times (1 - C) \]

where 0.9^\dagger is \text{sigmoid}^{-1}(0.9).
Logistic Representation of Conditional Probability

\[
P(d \mid A, B, C) = \text{sigmoid}(0.9^\dagger \ast A \ast B \\
+ 0.2^\dagger \ast A \ast (1 - B) \\
+ 0.3^\dagger \ast (1 - A) \ast C \\
+ 0.4^\dagger \ast (1 - A) \ast (1 - C))
\]

where \(0.9^\dagger\) is \(\text{sigmoid}^{-1}(0.9)\).

\[
P(d \mid A, B, C) = \text{sigmoid}(0.4^\dagger \\
+ (0.2^\dagger - 0.4^\dagger) \ast A \\
+ (0.9^\dagger - 0.2^\dagger) \ast A \ast B \\
+ ...)
\]
• Build a neural network to predict $D$ from $A$, $B$, $C$
Neural Network

- Build a neural network to predict $D$ from $A$, $B$, $C$
- A neural network tries to predict expected values
Neural Network

- Build a neural network to predict $D$ from $A$, $B$, $C$
- A neural network tries to predict expected values
- The expected value of a variable with domain $\{0, 1\}$ is its probability.
Neural Network

- Build a neural network to predict $D$ from $A$, $B$, $C$
- A neural network tries to predict expected values
- The expected value of a variable with domain $\{0, 1\}$ is its probability.
  Typically use a sigmoid activation at the root, and optimize with log likelihood.
Neural Network

- Build a neural network to predict $D$ from $A$, $B$, $C$
- A neural network tries to predict expected values
- The expected value of a variable with domain $\{0, 1\}$ is its probability.
  Typically use a sigmoid activation at the root, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features
- For other discrete variables, the expected value is not the probability.
Neural Network

- Build a neural network to predict $D$ from $A$, $B$, $C$
- A neural network tries to predict expected values
- The expected value of a variable with domain $\{0, 1\}$ is its probability.
  Typically use a sigmoid activation at the root, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features
- For other discrete variables, the expected value is not the probability.
  We create a Boolean ($\{0, 1\}$) variable for each value — indicator variable $\equiv$ having an output for each value
Neural Network

- Build a neural network to predict $D$ from $A$, $B$, $C$
- A neural network tries to predict expected values
- The expected value of a variable with domain $\{0,1\}$ is its probability.
  Typically use a sigmoid activation at the root, and optimize with log likelihood.
  — such a neural network is a logistic regression model with learned features
- For other discrete variables, the expected value is not the probability.
  We create a Boolean ($\{0,1\}$) variable for each value — indicator variable $\equiv$ having an output for each value
- For other domains, a Bayesian neural network can represent the distribution over the outputs (not just a point prediction).