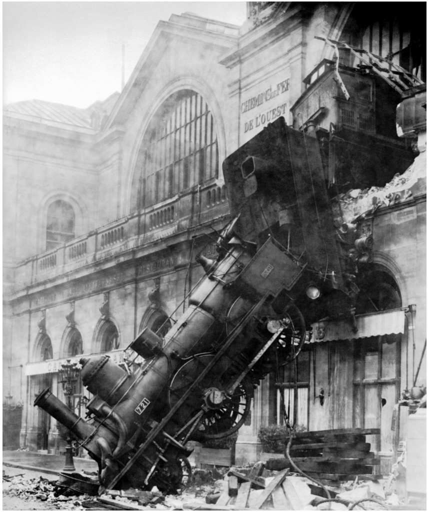
Machine Learning 1.11: Graphical Models

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

UNIVERSITY OF BATH

Derailment

- Imagine we want to predict when a train might derail due to "overspeed on sharp curves".



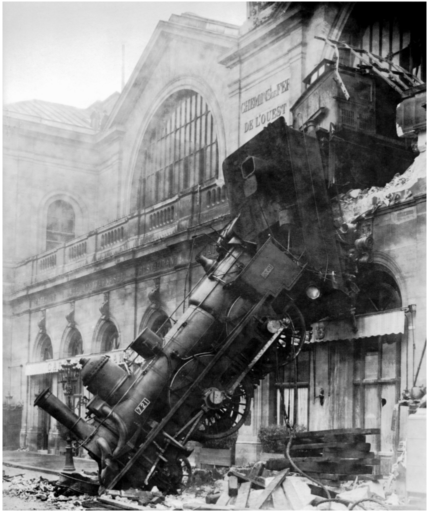
Derailment

- Imagine we want to predict when a train might derail due to "overspeed on sharp curves".

- Collect data:

Driver on duty (hours)	6	9	0	8	1	10
Lateness (%)	13	0	1	0	0	0
Speed Limit (km/h)	30	30	30	10	20	10
Radius (meters)	150	180	180	40	100	40
Train Age (years)	8	31	12	33	0	30
Track Age (years)	2	9	3	0	6	0
Crashed	0	0	0	1	1	1

(actually a simulation)



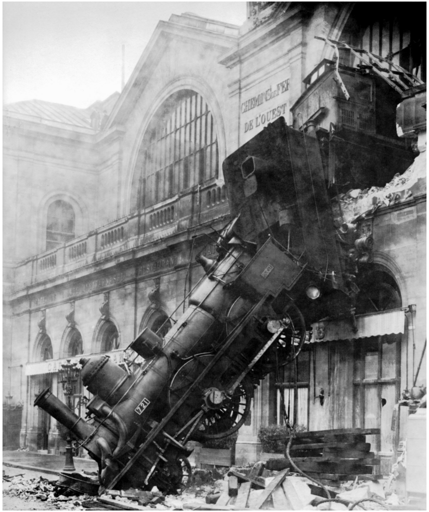
Derailment

- Imagine we want to predict when a train might derail due to "overspeed on sharp curves".
- Collect data:

Driver on duty (hours)	6	9	0	8	1	10
Lateness (%)	13	0	1	0	0	0
Speed Limit (km/h)	30	30	30	10	20	10
Radius (meters)	150	180	180	40	100	40
Train Age (years)	8	31	12	33	0	30
Track Age (years)	2	9	3	0	6	0
Crashed	0	0	0	1	1	1

(actually a simulation)

- But we also know **structure**:
  - First 3 parameters allow us to predict speed.
  - Speed plus second three allow us to predict a crash.



Structure

Instead of learning

$$P(\texttt{crashed}|\texttt{6 features})$$

learn

$$P(\texttt{crashed}|\texttt{3 features}, \texttt{speed})P(\texttt{speed}|\texttt{3 features})$$

Why is this useful?

Structure

Instead of learning

$$P(\texttt{crashed}|\texttt{6 features})$$

learn

$$P(\texttt{crashed}|\texttt{3 features}, \texttt{speed})P(\texttt{speed}|\texttt{3 features})$$

Why is this useful?

- Lower dimensional $\implies$ curse of dimensionality reduced.
- Can be thought of as an "explicit manifold".

Structure

Instead of learning

$$P(\texttt{crashed}|\texttt{6 features})$$

learn

$$P(\texttt{crashed}|\texttt{3 features}, \texttt{speed})P(\texttt{speed}|\texttt{3 features})$$

Why is this useful?

- Lower dimensional $\implies$ curse of dimensionality reduced.
- Can be thought of as an "explicit manifold".
- Improved performance.
- More data may be available for predicting speed.

Structure

Instead of learning

$$P(\texttt{crashed}|\texttt{6 features})$$

learn

$$P(\texttt{crashed}|\texttt{3 features}, \texttt{speed})P(\texttt{speed}|\texttt{3 features})$$

Why is this useful?

- Lower dimensional $\implies$ curse of dimensionality reduced.
- Can be thought of as an "explicit manifold".
- Improved performance.
- More data may be available for predicting speed.

Do we need to actually know speed?

Structure

Instead of learning

$$P(\texttt{crashed}|\texttt{6 features})$$

learn

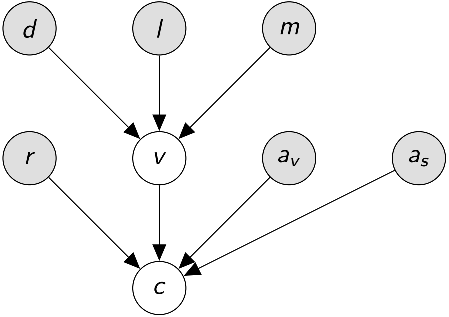
$$P(\texttt{crashed}|\texttt{3 features}, \texttt{speed})P(\texttt{speed}|\texttt{3 features})$$

Why is this useful?

- Lower dimensional $\implies$ curse of dimensionality reduced.
- Can be thought of as an "explicit manifold".
- Improved performance.
- More data may be available for predicting speed.

Do we need to actually know speed? No (but easier, so yes for now)

Visualising Dependency



d = Driver on duty (hours)
l = Lateness (%)
m = Speed Limit (km/h)

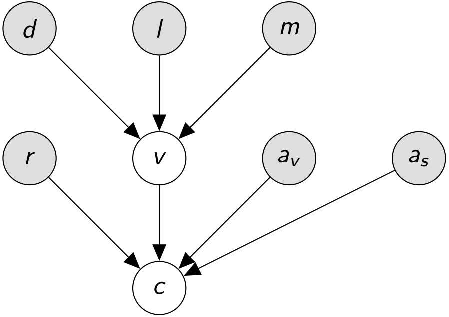
v = Speed (km/h)

r = Radius (meters)
a_v = Train Age (years)
a_s = Track Age (years)

c = Crashed

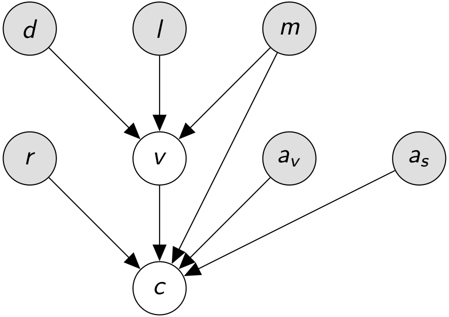
This is called a **graphical model**
(Specifically, it is a **Bayesian Network**)

Graphical Models



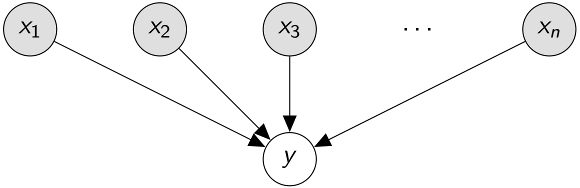
- Structure = conditional independence.
- If we know v (speed) then m (max speed) gives us **no extra information** about c (crash).

Graphical Models



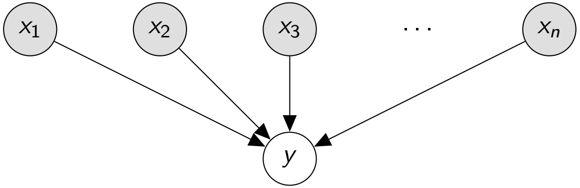
- Structure $=$ conditional independence.
- If we know v (speed) then m (max speed) gives us **no extra information** about c (crash).
- Structure is in the **omitted edges**.
- Connect m to c and above no longer true.

This is Pointless



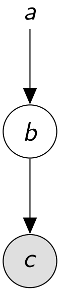
Why?

This is Pointless



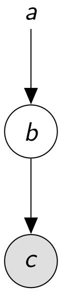
Why?
Everything connected $\therefore$ no conditional independence $\therefore$ no added structure
(Graphical model for classification/regression)

Parts



- Circle means **random variable** (b, c).
- No circle means fixed (hyper-)**parameter** (a).

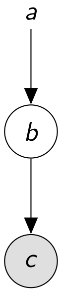
Parts



- Circle means **random variable** (b, c).
- No circle means fixed (hyper-)**parameter** (a).

- Shaded means **observed** (c).
- Unshaded means **latent** (b).

Parts

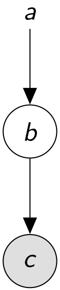


- Circle means **random variable** (b, c).
- No circle means fixed (hyper-)**parameter** (a).

- Shaded means **observed** (c).
- Unshaded means **latent** (b).

- More RVs are usually observed during training (often only model for test given).

Parts



- Circle means **random variable** (b, c).
- No circle means fixed (hyper-)**parameter** (a).

- Shaded means **observed** (c).
- Unshaded means **latent** (b).

- More RVs are usually observed during training (often only model for test given).

- Two kinds of latent variable:
  - **Hidden** – Something we know exists, but can't measure.
  - **Hypothetical** – Might not actually exist.

UNIVERSITY OF BATH

Three Representations

- **Bayesian network**
- Factor graphs
- Markov network

Bayesian Network

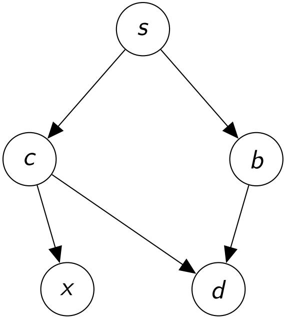
- Used for train example.
- Also called **Bayes network** or **Belief network**
- Despite name not always Bayesian!

Bayesian Network

- Used for train example.
- Also called **Bayes network** or **Belief network**
- Despite name not always Bayesian!

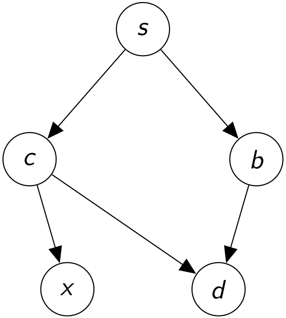
- Intuitive.
- Basis of "expert systems" (which don't really work).

Expert Systems I

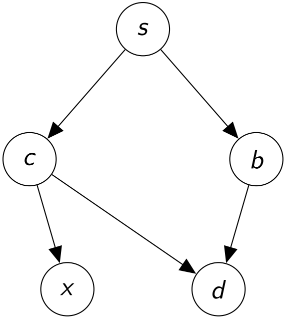


All RVs binary:

- $s =$ Smokes?
- $c =$ Has lung cancer?
- $b =$ Has bronchitis?
- $x =$ Shadow on x-ray?
- $d =$ Has dyspnoea? (difficulty breathing)



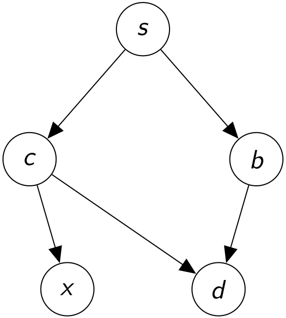
To equation:



To equation:

- Conditional probability for every RV:
  $= P(s|.)P(c|.)P(b|.)P(x|.)P(d|.)$

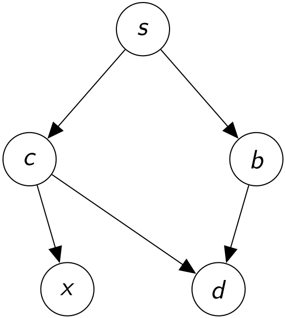
Expert Systems II



To equation:

- Conditional probability for every RV:
  $= P(s|.)P(c|.)P(b|.)P(x|.)P(d|.)$
- Conditional on parent RVs only:
  $= P(s)P(c|s)P(b|s)P(x|c,s)P(d|c,b)$

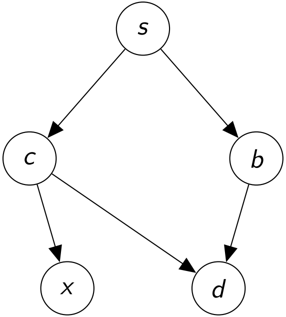
Expert Systems II



To equation:

- Conditional probability for every RV:
  $= P(s|.)P(c|.)P(b|.)P(x|.)P(d|.)$
- Conditional on parent RVs only:
  $= P(s)P(c|s)P(b|s)P(x|c,s)P(d|c,b)$
- Equal to joint distribution over all RVs
  $= P(s,c,b,x,d)$

Expert Systems III



Parameters – binary RVs ∴

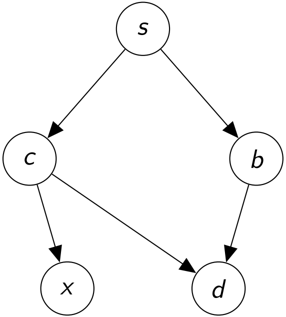
- $P(s = \texttt{false}) = 0.5$
- $P(s = \texttt{true}) = 0.5$

- $P(c = \texttt{false}|s = \texttt{false}) = 0.99$
- $P(c = \texttt{false}|s = \texttt{true}) = 0.9$
- $P(c = \texttt{true}|s = \texttt{false}) = 0.01$
- $P(c = \texttt{true}|s = \texttt{true}) = 0.1$

$\vdots$

(22 all in)

Expert Systems IV

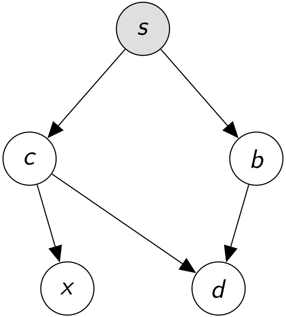


Can calculate any probability for any combination of known/unknown information.

1. Patient walks in, everything unknown.

$$P(c) = 0.055, \quad P(b) = 0.45$$

Expert Systems IV



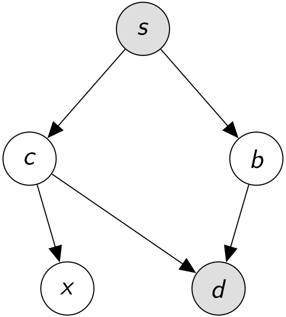
Can calculate any probability for any combination of known/unknown information.

1. Patient walks in, everything unknown.
2. Doctor asks if they smoke (true).

$$P(c) = 0.1, \quad P(b) = 0.6$$

$$P(s = \texttt{true}) = 1$$

Expert Systems IV



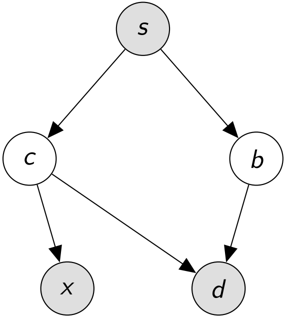
Can calculate any probability for any combination of known/unknown information.

1. Patient walks in, everything unknown.
2. Doctor asks if they smoke (true).
3. Doctor asks if they have difficulty breathing (false).

$$P(c) = 0.037, \quad P(b) = 0.242$$

$$P(s = \texttt{true}) = 1$$
$$P(d = \texttt{false}) = 1$$

Expert Systems IV



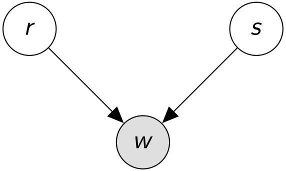
Can calculate any probability for any combination of known/unknown information.

1. Patient walks in, everything unknown.
2. Doctor asks if they smoke (true).
3. Doctor asks if they have difficulty breathing (false).
4. Doctor sends them for an x-ray (true).

$P(c) = 0.430, \quad P(b) = 0.158$

$P(s = \texttt{true}) = 1$
$P(d = \texttt{false}) = 1$
$P(x = \texttt{true}) = 1$

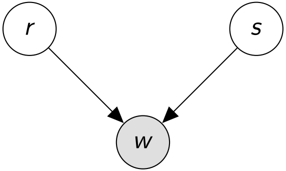
Explaining Away



Binary RVs:

- r = It's been raining.
- s = Sprinklers have been on.
- w = Grass is wet.

Explaining Away

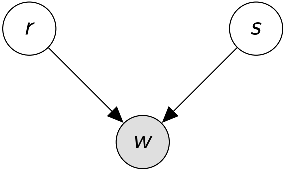


Binary RVs:

- r = It's been raining.
- s = Sprinklers have been on.
- w = Grass is wet.

Observe grass is wet, calculate:
$P(r) = 0.38, \quad P(s) = 0.76$

Explaining Away



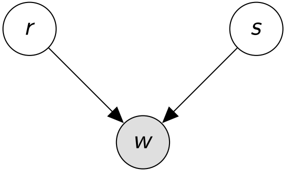
Binary RVs:

- $r =$ It's been raining.
- $s =$ Sprinklers have been on.
- $w =$ Grass is wet.

Observe grass is wet, calculate:
$P(r) = 0.38, \quad P(s) = 0.76$

Find out sprinkler on, calculate:
$P(r) = 0.2, \quad P(s) = 1.0$

Explaining Away



Binary RVs:

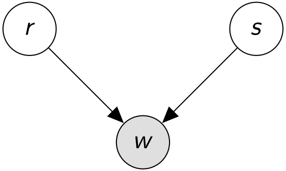
- $r =$ It's been raining.
- $s =$ Sprinklers have been on.
- $w =$ Grass is wet.

Observe grass is wet, calculate:
$P(r) = 0.38, \quad P(s) = 0.76$

Find out sprinkler on, calculate:
$P(r) = 0.2, \quad P(s) = 1.0$

Note how $P(r)$ drops?
Have explained wet grass $\therefore$ rain less likely.
Called **explaining away**.

Explaining Away



Note: Observing a parent does not make the children conditionally independent!

Binary RVs:

- r = It's been raining.
- s = Sprinklers have been on.
- w = Grass is wet.

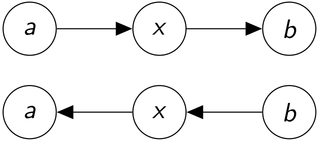
Observe grass is wet, calculate:
$P(r) = 0.38, \quad P(s) = 0.76$

Find out sprinkler on, calculate:
$P(r) = 0.2, \quad P(s) = 1.0$

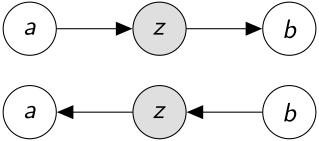
d-separation I

- d = directional. Opposite is d-connected.
- Defines if two nodes, a and b are conditionally independent, given observed nodes, z.
- Must be checked for all connecting paths, ignoring directions.

d-connected (dependent):



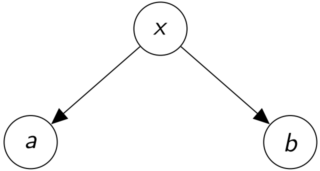
d-seperated (conditionally independent):



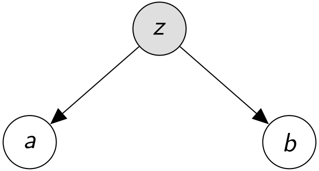
d-separation II

- d $=$ directional. Opposite is d-connected.
- Defines if two nodes, a and b are conditionally independent, given observed nodes, z.
- Must be checked for all connecting paths, ignoring directions.

d-connected (dependent):



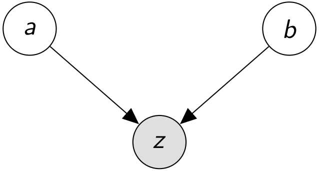
d-seperated (conditionally independent):



d-separation III

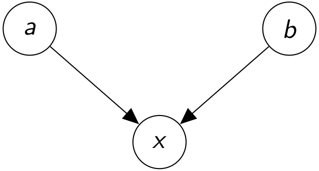
- d = directional. Opposite is d-connected.
- Defines if two nodes, a and b are conditionally independent, given observed nodes, z.
- Must be checked for all connecting paths, ignoring directions.

d-connected (dependent):



This is backwards!

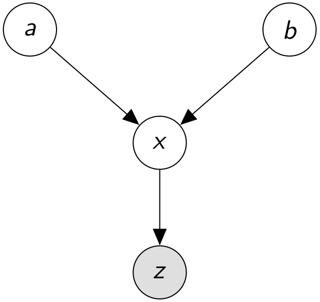
d-seperated (conditionally independent):



d-separation IV

- d = directional. Opposite is d-connected.
- Defines if two nodes, a and b are conditionally independent, given observed nodes, z.
- Must be checked for all connecting paths, ignoring directions.

d-connected (dependent):



Can think in terms of information "bouncing" off of forks and the children of forks.

Three Representations

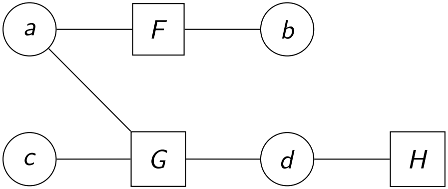
- Bayesian network
- **Factor graphs**
- Markov network

Factor Graphs

- Product of arbitrary functions.
- No requirement to be probabilistic or conditional.
- First used for "*Boolean satisfiability problem*" (SAT).

Factor Graphs

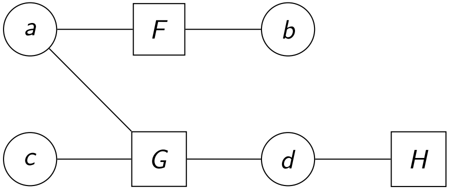
- Product of arbitrary functions.
- No requirement to be probabilistic or conditional.
- First used for "*Boolean satisfiability problem*" (SAT).



$$= F(a,b)G(a,c,d)H(d)$$

Factor Graphs

- Product of arbitrary functions.
- No requirement to be probabilistic or conditional.
- First used for "*Boolean satisfiability problem*" (SAT).



$$= F(a, b)G(a, c, d)H(d)$$

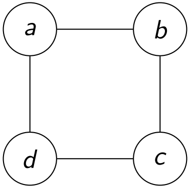
- Circles are RVs as before.
- Squares are functions, linked to their dependent variables.
- Often represents unnormalised probability.

Three Representations

- Bayesian network
- Factor graphs
- **Markov network**

Markov Network

- If a factor graph only has functions of one and two variables don't draw the boxes!



$$= F(a, b)G(b, c)H(c, d)I(d, a)A(a)B(b)C(c)D(d)$$

- First four: Pairwise terms
- Second four: Unary terms

Relationships

- All can be converted to factor graphs.
  - Convert to equation.
  - Add factors, connect to relevant RVs.

- Not necessarily true other way around.
- Can combine representations; risk of confusion.

Limits

RVs can represent anything:

- Single value
- Vector
- Matrix
- 3D model
- Graphical model!

Limits

RVs can represent anything:

- Single value
- Vector
- Matrix
- 3D model
- Graphical model!

Functions can be anything:

- Standard distributions
- Logistic regression
- Random forest
- Neural network
- Graphical model!

Note how graphical model is in both lists.

Further Advantages

- Support missing information.
- Can answer many "questions" (next lecture).

- Can use negative log likelihood "costs".
- Does not have to be properly probabilistic (though preferred).

Causality

- Graphical models are usually **causal**.
- Not necessary (or always possible), but easier to think about.

Causality

- Graphical models are usually **causal**.
- Not necessary (or always possible), but easier to think about.

- You can learn causality... but it's a research problem.
- Often ambiguous, and subject to unobserved information breaking your conclusions.
- Leads to a mathematical treatment of the scientific method.

Algorithms for Graphical Models

- Next lecture:
  - Dynamic programming.
  - Belief propagation.

- Next semester:
  - Gibbs sampling and other "Markov Chain Monte Carlo" (MCMC) methods.
  - Variational models.
  - Graph cuts (maybe).

(main ones; there are many more!)

Summary

- Graphical models represent problem structure (or other reasonable assumptions).
- Very flexible.

Further Reading

- "Information Theory, Inference, and Learning Algorithms" by MacKay, Chapter 16 – Viterbi algorithm.
- "Causality", by Judea Pearl – the only book you will ever want on causality; but only if you really, really want to know more (extremely hard stuff).
- Nothing to do with lecture:
  Kaggle article on data science statistics (biased sample)
  `https://www.kaggle.com/ash316/novice-to-grandmaster`