Computer-supported Decision Making

Including: Which movie should I watch tonight?

Helge Langseth

Joint work with Thomas D. Nielsen, Aalborg University

NTNU

Outline

A gas leak scenario

1. A gas leak (L) can lead to an explosion (X);
2. An explosion (X) can lead to one or more casualties (C);
3. Gas leaks (L) detected by a gas detector (G) are not harmful;
4. The environment (E) influences the gas leak frequency (L)
5. . . . and the reliability of the gas detector (G).

A gas leak scenario

1. A gas leak (L) can lead to an explosion (X);
2. An explosion (X) can lead to one or more casualties (C);
3. Gas leaks (L) detected by a gas detector (G) are not harmful;
4. The environment (E) influences the gas leak frequency (L)
5. . . . and the reliability of the gas detector (G).

Relevant questions we may pose:

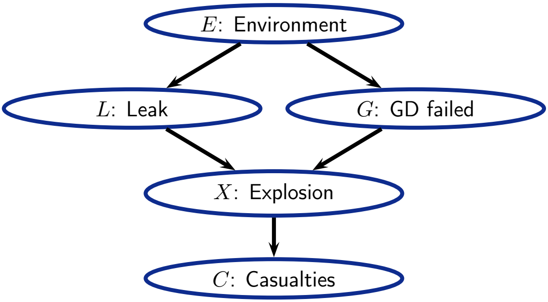
Deductive reasoning: *"To what extent will an improved gas detector reduce expected number of casualties/year?"*

Abductive reasoning: *"Having seen a series of explosions recently, can I say something about the environment?"*

General case: *"What is* $P(\text{Query variables}|\text{Observed variables})$*?"*

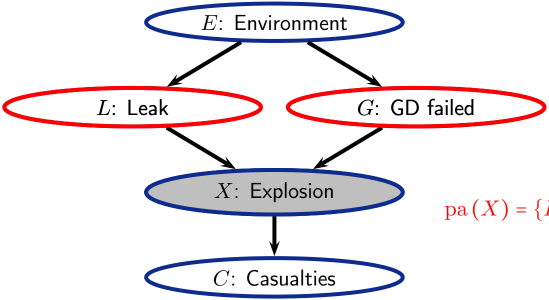
Desired: A modelling framework where (causal) knowledge can be encoded and relevant queries answered.

A simple BN example: "Explosion"



$$P(E, L, G, X, C)$$

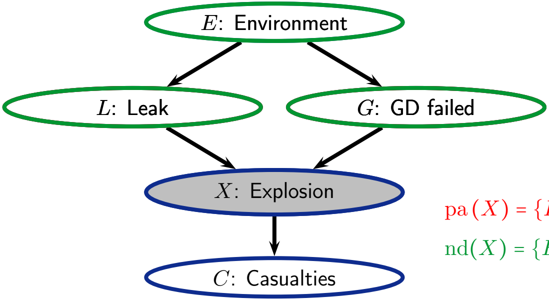
A simple BN example: "Explosion"



$\mathrm{pa}\,(X) = \{L, G\}$

$P(E, L, G, X, C)$

A simple BN example: "Explosion"

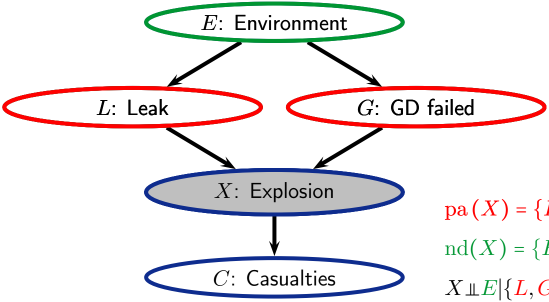


$\mathrm{pa}(X) = \{L, G\}$

$\mathrm{nd}(X) = \{E, L, G\}$

$$P(E, L, G, X, C)$$

A simple BN example: "Explosion"



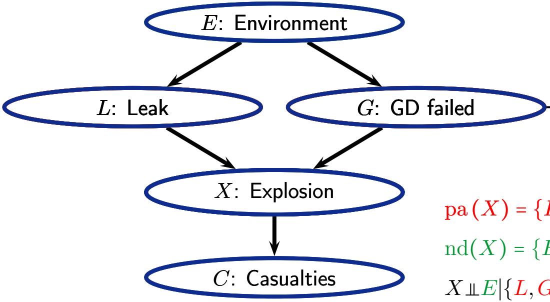
$\mathrm{pa}(X) = \{L, G\}$

$\mathrm{nd}(X) = \{E, L, G\}$

$X \perp\!\!\!\perp E \mid \{L, G\}$

$P(E, L, G, X, C)$

A simple BN example: "Explosion"



G	E =hostile	E =normal
yes	p_H	p_N
no	$1 - p_H$	$1 - p_N$

$P(G|\mathrm{pa}(G))$

$\mathrm{pa}(X) = \{L, G\}$

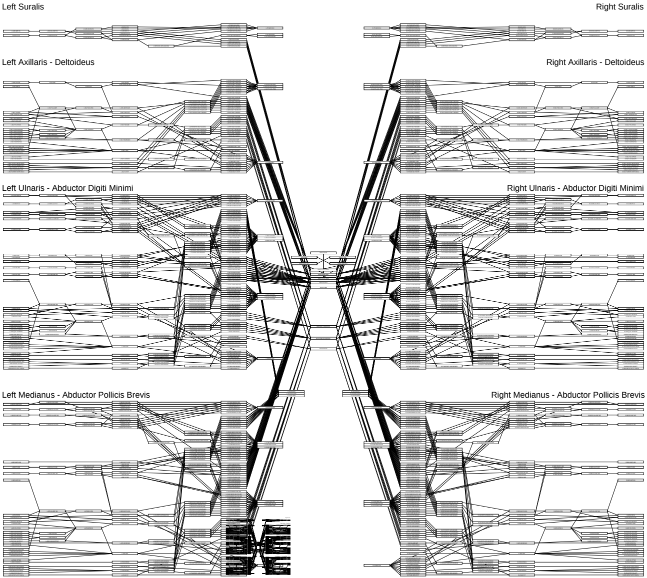
$\mathrm{nd}(X) = \{E, L, G\}$

$X \perp\!\!\!\perp E | \{L, G\}$ (Hence, $P(X|E, L, G) = P(X|L, G)$)

$$P(E, L, G, X, C) = P(E) \cdot P(L|E) \cdot P(G|E, L) \cdot P(X|E, L, G) \cdot P(C|E, L, G, X)$$
$$= P(E) \cdot P(L|E) \cdot P(G|E) \cdot P(X|L, G) \cdot P(C|X)$$

Fast inference algorithms utilise these independence properties.

MUNIN: An expert system for electromyography



Consider a text document represented by the variables

X_i: "Is word i of the vocabulary used in the document?"

We have n (no. terms in vocabulary) variables for each document.

Example:

- Vocabulary: {`text`, `image`, `Bayes`, `network`, `analysis` }
- Document: $\underbrace{\texttt{Bayes}}_{\text{word 3}}\texttt{ian}\ \underbrace{\texttt{text}}_{\text{word 1}}\ \underbrace{\texttt{analysis}}_{\text{word 5}}\ \underbrace{\texttt{rocks}}_{\text{N/A}}.$
- Representation: $(X_1, \ldots, X_5) = (1, 0, 1, 0, 1)$

Consider a text document represented by the variables

X_i: "Is word i of the vocabulary used in the document?"

We have n (no. terms in vocabulary) variables for each document.

Example:

- Vocabulary: {`text`, `image`, `Bayes`, `network`, `analysis` }
- Document: $\underbrace{\texttt{Bayes}}_{\text{word 3}}\texttt{ian}\ \underbrace{\texttt{text}}_{\text{word 1}}\ \underbrace{\texttt{analysis}}_{\text{word 5}}\ \underbrace{\texttt{rocks}}_{\text{N/A}}$.
- Representation: $(X_1, \ldots, X_5) = (1, 0, 1, 0, 1)$

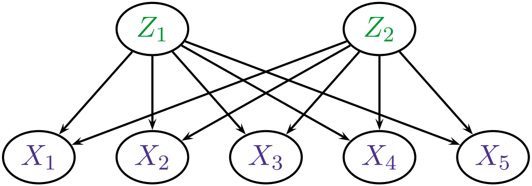
Question:

Can we automatically examine a number of documents and find their "meaning", thus get a representation better suited for analysis?

- Fewer, more cleverly defined features.
- New features capturing *semantics* ("meaning") of text, not *syntax* ("writing style").

One solution:

- Introduce $\boldsymbol{Z} = (Z_1, \ldots, Z_q)$, a vector of latent variables, representing a "compressed" representation (assuming $q \ll n$).
- Each Z_j defines a "topic" (aggregated meaning) of document; the presence of a topic influences the probability of seeing a specific word in the document.
- In this *factor analysis* model, the factor variables model the correlation among the attributes.



Example:

- The often-used `reuters-22173` dataset was analysed.
  - Vocabulary containing a subset of 500 words (chosen automatically).
  - Documents appeared on the Reuters newswire in 1987.
- Some of the discovered topics:
  - negotiation, agreement, deal, hope, meeting, international, . . .
  - USDA, agriculture, crop, export, wheat, tonnes, . . .
  - estimate, expect, statistics, reserve, fall, rise, season, qtr, . . .
  - . . .

Key conclusion:
Latent variables appear useful to detect/automatically learn and represent generalisations of high-dimensional data!

Collaborative filtering: To predict the utility of items for the active user based on a database of rating data.

Input data: Our data is a *sparse matrix* of *ratings*

	Star Wars	Platoon	π	Olsen banden 1	Festen	Das boot	...
User 1		5		1	4	5	...
User 2	1		5	2	5		...
User 3	5		5				...
User 4		2	1		3	2	...
User 5	3		4			5	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Notation: $R_{p,i}$ is the rating User p gives to Movie i, so $R_{2,1} = 1$.

Goal: Predict values for unobserved ratings given a database of ratings. E.g., how will User 5 rate Festen? (I.e., what is a good guess for $R_{5,5}$?)

What determines how a use will rate a movie?

- Let movie no. i be represented by $\boldsymbol{M}_i$, a point in $\mathbb{R}^q$:
  - Each of the q dimensions of $\boldsymbol{M}_i$ has some (implicit) meaning, e.g., degree of chick-movie, size of production, etc.
  - Finding a good encoding is done automatically during learning, but we can inspect the representation afterwards.
  - Movie representations spread around zero; we model them as standard Gaussians a priori.
  - Each user p has a preference for the q aspects of a movie. This is modelled by a q-dimensional vector $\boldsymbol{v}_p$.

What determines how a use will rate a movie?

- Let movie no. i be represented by $\boldsymbol{M}_i$, a point in $\mathbb{R}^q$.
- Let user no. p be represented by $\boldsymbol{U}_p$, a point in $\mathbb{R}^r$:
  - Each of the r dimensions of $\boldsymbol{U}_p$ has some (implicit) meaning.
  - User representations spread around zero; we model them as standard Gaussians a priori.
  - Each movie i has a connection to the r aspects of a user. This is modelled by a r-dimensional vector $\boldsymbol{w}_i$.

What determines how a use will rate a movie?

- Let movie no. i be represented by $\boldsymbol{M}_i$, a point in $\mathbb{R}^q$.
- Let user no. p be represented by $\boldsymbol{U}_p$, a point in $\mathbb{R}^r$.
- Let ψ_p be average rating for user p (modelling "grumpiness"), and ϕ_i be average rating for movie i after adjusting for which users have rated it (modelling "quality").

What determines how a use will rate a movie?

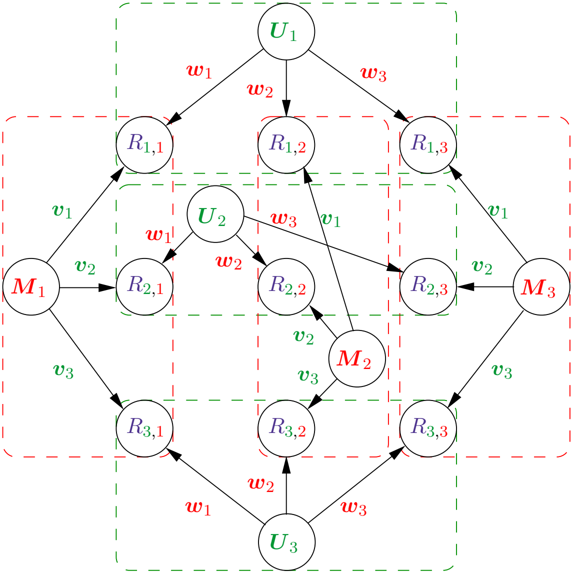
- Let movie no. i be represented by $\boldsymbol{M}_i$, a point in $\mathbb{R}^q$.
- Let user no. p be represented by $\boldsymbol{U}_p$, a point in $\mathbb{R}^r$.
- Let ψ_p be average rating for user p (modelling "grumpiness"), and ϕ_i be average rating for movie i after adjusting for which users have rated it (modelling "quality").

Rating model:

$$R_{p,i} | \{\boldsymbol{M}_i = \boldsymbol{m}_i, \boldsymbol{U}_p = \boldsymbol{u}_p\} = \boldsymbol{v}_p{}^{\mathsf{T}}\boldsymbol{m}_i + \boldsymbol{w}_i{}^{\mathsf{T}}\boldsymbol{u}_p + \phi_i + \psi_p + \epsilon$$

where ϵ is a noise term.

The model is a Bayesian Network



The 10 items closest to *Star Wars* and *Tree Colors: Blue*:

	Close to *Star Wars*
1.	The Empire Strikes Back
2.	*The Princess Bride*
3.	Star Trek II
4.	Return of the Jedi
5.	Raiders of the Lost Ark
6.	Star Trek IV
7.	*Private Parts*
8.	Star Trek VI
9.	Mystery Science Theatre 3000
10.	Men in Black

	Close to *Three Colors: Blue*
1.	Welcome to the Dollhouse
2.	Heavenly Creatures
3.	Three Colors: White
4.	Wings of Desire
5.	Everyone Says I Love You
6.	Muriel's Wedding
7.	Dead Man Walking
8.	The Nightmare Before Christmas
9.	Boogie Nights
10.	To Die For

NOTE!
Only the rating matrix is used to find these patterns!

About the model building:

- We chose $r = 1$ and $q = 3$ for this analysis
- `MovieLens` data – 100 000 ratings (943 users, 1682 movies)
- **Automatic (EM) learning** used to find all parameters

Understanding the Model – Quality parameter

The value of ψ_i may say something about the "movie quality":

Movies with highest ψ_i	
1.	The Shawshank Redemption ▲
2.	Schindler's List ▲
3.	Star Wars ▲
4.	Casablanca ▲
5.	The Usual Suspects ▲
6.	Rear Window ▲
7.	Raiders of the Lost Ark ▲
8.	The Silence of the Lambs ▲
9.	One Flew Over the Cuckoo's Nest ▲
10.	12 Angry Men ▲

Movies with highest average rating	
1.	Entertaining Angels: The Dorothy Day Story ▼
2.	Someone Elses America ▼
3.	Aiqing wansui ▼
4.	Santa with Muscles ▼
5.	The Saint of Fort Washington ▼
6.	Star Kid ▼
7.	Marlene Dietrich: Shadow and Light ▼
8.	Prefontaine ▼
9.	They Made Me a Criminal ▼
10.	A Great Day in Harlem ▼

▲ : **In Top 25** of IMDBs list of 250 best movies

▼ : **Not at all** in IMDBs list of 250 best movies

Conclusions

- Making decisions under uncertainty is a task we are faced with every day. Still, it is hard do automate this reasoning process in a computer. **Bayesian Networks** is one framework that shows promise in this regard.
- **Latent variable models** can effectively be used to synthesise/aggregate information from complex high-dimensional data.
- We have seen by example that the latent variables give a reasonable aggregation in the movie domain, as they offer a relevant **semantic interpretation**.
- Although we have only discussed qualitative properties of the model, it also gives **very good quantitative results** wrt. quality of recommendations: We obtain a *mean absolute error* of **0.69** on the `MovieLens` dataset; other systems typically in the area **0.73 – 0.74**.