Scalable Uncertainty Management

06 – Markov Logic

Rainer Gemulla

July 13, 2012

Overview

In this lecture

- Statistical relational learning (SRL)
- Introduction to probabilistic graphical models (PGM)
- Basics of undirected models (called *Markov networks*)
- Markov logic as a template for undirected models
- Basics of inference in Markov logic networks

Not in this lecture

- Directed models (called *Bayesian networks*)
- Other SRL approaches (such as *probabilistic relational models*)
- High coverage and in-depth discussion of inference
- Learning Markov logic networks

Outline

Correlations in probabilistic databases

- Simple probabilistic models
  - Tuple-independent databases
  - Block-disjoint independent databases
  - Key/foreign key constraints, ...
- Correlations (mainly) through $\mathcal{RA}$ queries/views
  - Any discrete probability distribution can be modeled
  - Queries describe *precisely* how result is derived

Example (Nell)

NellExtraction

Subject	Pattern	Object	Source	$\mathbb{P}$
Sony	produces	Walkman	1	0.96
IBM	produces	PC	1	0.96
IBM	produces	PC	2	1
Microsoft	produces	MacOS	2	0.9
AlbertEinstein	bornIn	Ulm	1	0.9

NellSource

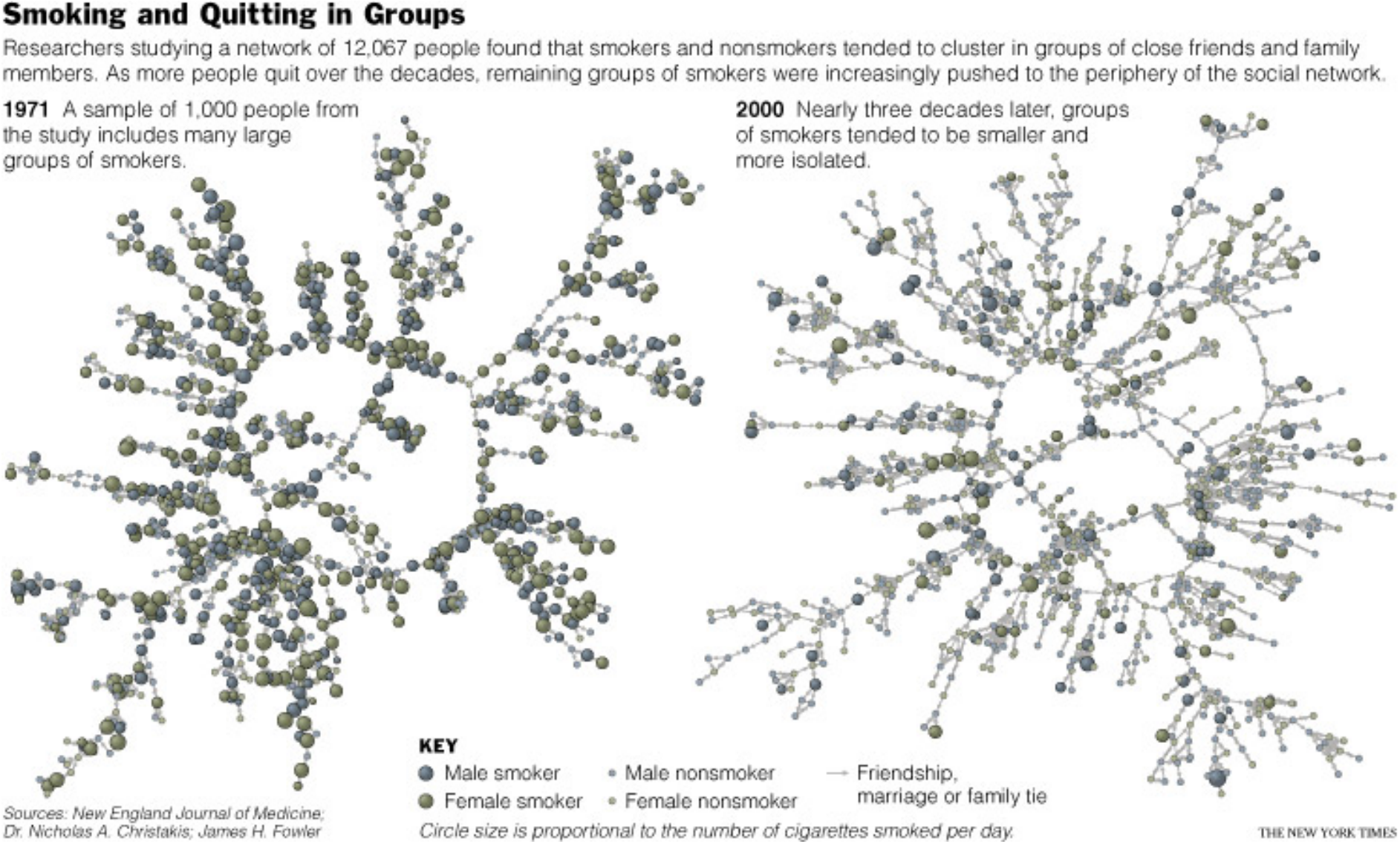
Source	$\mathbb{P}$
1	0.99
2	0.1

$$\text{Produces}(x, y) \leftarrow \text{NellExtraction}(x, \text{'produces'}, y, s), \text{NellSource}(s)$$

Produces

Subject	Object	$\mathbb{P}$
Sony	Walkman	0.9504
IBM	PC	0.95536
Microsoft	MacOS	0.09

Statistical relational learning (I)



Does John smoke?

Learn correlations from structured data, then apply to new data.

Statistical relational learning (II)

- Goal: Declarative modelling of correlations in structured data
- Idea: Use (subsets of) first-order logic
  - Very expressive formalism; lots of knowledge bases use it
  - Symmetry: $\forall x.\forall y.\text{Friends}(x,y) \iff \text{Friends}(y,x)$
  - Everybody has a friend: $\forall x.\exists y.\text{Friends}(x,y)$
  - Transitivity: $\forall x.\forall y.\forall z.\text{Friends}(x,y) \wedge \text{Friends}(y,z) \implies \text{Friends}(x,z)$
  - Smoking causes cancer: $\forall x.\text{Smokes}(x) \implies \text{Cancer}(x)$
  - Friends have similar smoking habits:
    $\forall x.\forall y.\text{Friends}(x,y) \implies (\text{Smokes}(x) \iff \text{Smokes}(y))$
- Problem: Real-world knowledge is incomplete, contradictory, complex
  $\rightarrow$ Above rules do *not* generally hold, but they are "likely" to hold!
- Approach: Combine first-order logic with probability theory
  - Expressiveness of first-order logic
  - Principled treatment of uncertainty using probability theory

There are many approaches of this kind. Our focus is on *Markov logic*, a recent and very successful language.

Markov logic networks

Definition

A *Markov logic network* is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and the weight w_i is a real number.

Example

$$1.5 \begin{cases} \text{Smoking causes cancer} \\ \forall x.\text{Smokes}(x) \implies \text{Cancer}(x) \end{cases}$$

$$1.1 \begin{cases} \text{Friends have similar smoking habits} \\ \forall x.\forall y.\text{Friends}(x,y) \implies (\text{Smokes}(x) \iff \text{Smokes}(y)) \end{cases}$$

- Formulas may or may not hold
- Weights express confidence
  - High positive weight $\rightarrow$ confident that formula holds
  - High negative weight $\rightarrow$ confident that formula does not hold
  - But careful: weights actually express confidence of certain "groundings" of a formula and *not* the formula as a whole (more later)
- Formulas may introduce complex correlations

Simple MLN for entity resolution

Which citations refer to the same publication?

author	Richardson, Matt and Domingos, Pedro	M. Richardson and P. Domingos	Domingos, Pedro and Richardson, Matthew
title	Markov Logic Networks	Markov logic networks	Markov Logic: A Unifying Framework for Statistical Relational Learning
year	2006	2006	2007

// predicates
HasToken(token, field, citation) *// e.g., HasToken('Logic', 'title', C1)*
SameField(field, citation, citation) *// Semantic equality of values in a field*
SameCitation(citation, citation) *// Semantic equality of citations*

// formulas
HasToken(+t, +f, c1) ^ HasToken(+t, +f, c2) => SameField(+f, c1, c2)
SameField(+f, c1, c2) => SameCitation(c1, c2)
SameCitation(c1, c2) ^ SameCitation(c2, c3) => SameCitation(c1, c3)

Rule weights are usually learned from data. The same rule may have different weights for different constants (indicated by "+").

Alchemy

- Alchemy is well-known software package for Markov logic
- Developed at University of Washington
- Supports a wide range of tasks
  - Structure learning
  - Weight learning
  - Probabilistic inference
- Has been used for wide range of applications
  - Information extraction
  - Social network modeling
  - Entity resolution
  - Collective classification
  - Link prediction
- Check out `http://alchemy.cs.washington.edu/`
  - Code
  - Real-world datasets
  - Real-world Markov logic networks
  - Literature

From Markov logic to graphical models (example)

Friends

Name1	Name2	Value
Anna	Bob	Yes
Bob	Anna	Yes
Anna	Anna	Yes
Bob	Bob	Yes

Smokes

Name	Value
Anna	Yes

Cancer

Name	Value
Anna	No

1.5 { Smoking causes cancer
$\forall x.\text{Smokes}(x) \implies \text{Cancer}(x)$

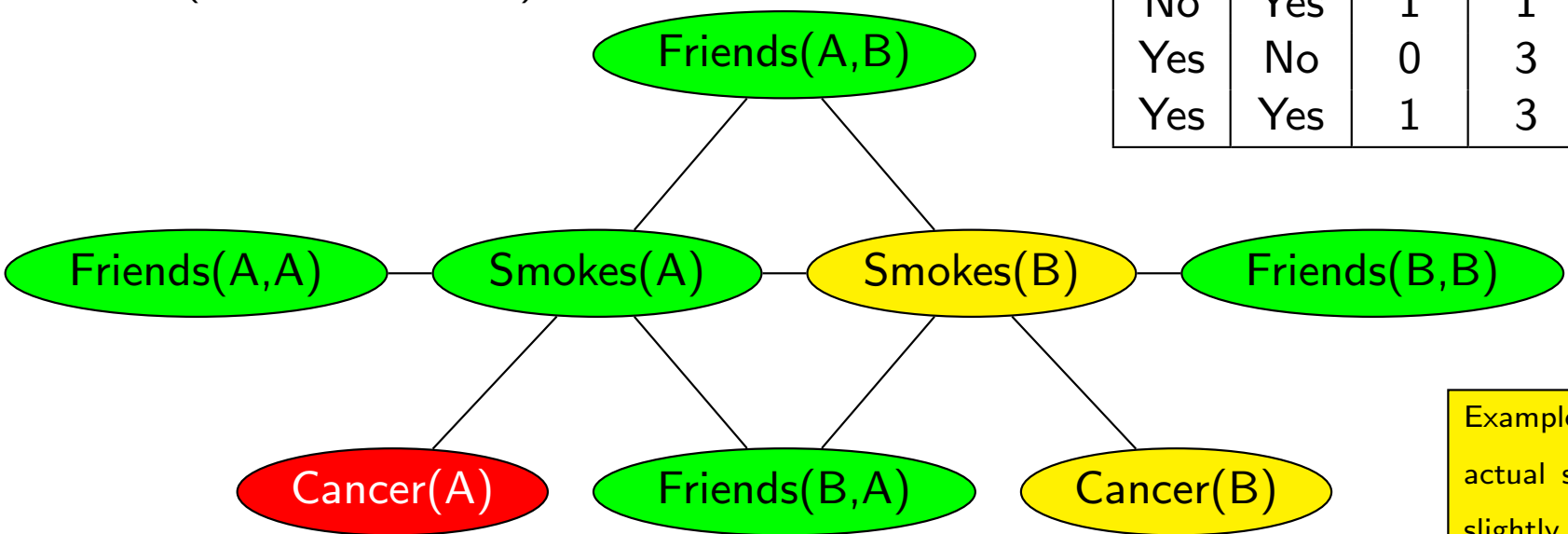
1.1 { Friends have similar smoking habits
$\forall x.\forall y.\text{Friends}(x,y) \implies (\text{Smokes}(x) \iff \text{Smokes}(y))$

Inference result

- $\mathbb{P}(\text{Bob smokes}) = 84.6\%$
- $\mathbb{P}(\text{Bob has cancer}) = 76.9\%$

Inference (conceptual)

S(B)	C(B)	#R1	#R2	$\sum w$	$\mathbb{P}$
No	No	1	1	2.6	7.7%
No	Yes	1	1	2.6	7.7%
Yes	No	0	3	3.3	15.4%
Yes	Yes	1	3	4.8	69.2%



Example is simplified; actual semantics are slightly different.

Probabilistic databases and graphical models

	Probabilistic databases	**Graphical models**
Probabilistic model	Simple (disjoint-independent tuples)	Complex (independencies given by graph)
Query	Complex (e.g., $\exists x.\exists y.R(x,y) \wedge S(x)$)	Simple (e.g., $\mathbb{P}(X_1, X_2 \mid Z_1, Z_2, Z_3)$)
Network	Dynamic (database + query)	Static (Bayesian or Markov network)
Complexity measured in size of	Database	Network
Complexity parameter	Query	Treewidth
System	Extension to RDBMS	Stand-alone

Hybrid approaches have many potential applications and are under active research.

Outline

Outline

Reasoning with uncertainty

- Goal: Automated reasoning system
  - Take all available information
    (e.g., patient information: symptoms, test results, personal data)
  - Reach conclusions
    (e.g., which diseases the patient has, which medication to give)
- Desiderata
  1. Separation of knowledge and reasoning
     - ⋆ Declarative, model-based representation of knowledge
     - ⋆ General suite of reasoning algorithms, applicable to many domains
  2. Principled treatment of uncertainty
     - ⋆ Partially observed data
     - ⋆ Noisy observations
     - ⋆ Non-deterministic relationships
- Lots of applications
  - medical diagnosis, fault diagnosis, analysis of genetic and genomic data, communication and coding, analysis of marketing data, speech recognition, *natural language understanding*, segmenting and denoising images, social network analysis, ...

Probabilistic models

- Multiple interrelated aspects may relate to the reasoning task
  - Possible diseases
  - Hundreds of symptoms and diagnostic tests
  - Personal characteristics

1. Characterize data by a set of random variables
   - Flu (yes / no)
   - Hayfever (yes / no)
   - Season (Spring / Sommer / Autumn / Winter)
   - Congestion (yes / no)
   - MusclePain (yes / no)

   $\rightarrow$ Variables and their domain are important design decision
2. Model dependencies by a *joint distribution*
   - Diseases, season, and symptoms are correlated
   - Probabilistic models construct joint probability space

     $\rightarrow 2 \cdot 2 \cdot 4 \cdot 2 \cdot 2$ outcomes (64 values, 63 non-redundant)
   - Given joint probability space, interesting questions can be answered

$$\mathbb{P}(\,\mathrm{Flu} \mid \mathrm{Season}{=}\mathrm{Spring}, \mathrm{Congestion}, \neg\mathrm{MusclePain}\,)$$

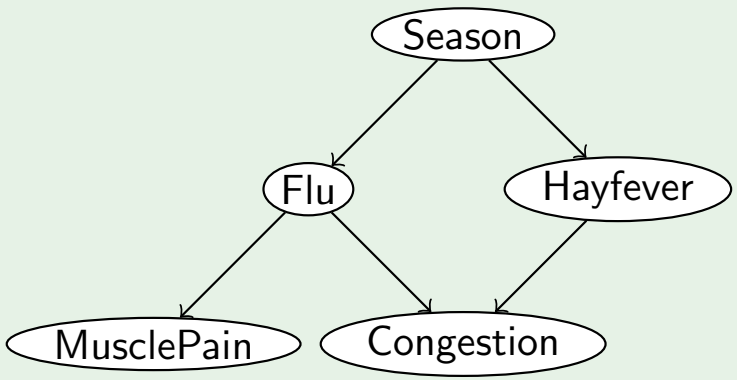
Specifying a joint distribution is infeasible in general!

Probabilistic graphical models

- A graph-based representation of *direct* probabilistic interactions
- A break-down of high-dimensional distributions into smaller *factors* (here: 63 vs. 17 non-redundant parameters)
- A compact representation of a set of (conditional) *independencies*

Example (directed graphical model)

Graph representation



Factorization

$$\mathbb{P}(S, F, H, M, C) = \mathbb{P}(S)\,\mathbb{P}(F \mid S)\,\mathbb{P}(H \mid S)\,\mathbb{P}(C \mid F, H)\,\mathbb{P}(M \mid F)$$

Independencies

$$(F \perp H \mid S), (C \perp S \mid F, H), (M \perp H, C, S \mid F)$$

Main components

1. Representation
   - Tractability
     - ★ Variables tend to interact *directly* only with very few others
     - ★ Natural and compact encoding as graphical model
   - Transparency
     - ★ Models can be understood/evaluated by human experts
2. Inference
   - Answer queries using the distribution as model of the world
   - Work on graph structure
     $\rightarrow$ orders of magnitude faster than working on joint probability
3. Learning
   - Learn a model from data that captures past experience to a good approximation
   - Human experts may provide rough guidance
   - Details filled in by fitting the model to the data
     $\rightarrow$ Often better reflection of domain than hand-constructed models, sometimes surprising insights

Graphical models exploit locality structure that appears in many distributions that arise in practice.

Outline

Notation

Let $\mathbf{X}$ and $\mathbf{Y}$ be sets of random variables with domain $\mathrm{Dom}(\mathbf{X})$ and $\mathrm{Dom}(\mathbf{Y})$. Let $\mathbf{x} \in \mathrm{Dom}(\mathbf{X})$ and $\mathbf{y} \in \mathrm{Dom}(\mathbf{Y})$.

Expression	Shortcut notation
$\mathbb{P}(\mathbf{X} = \mathbf{x})$	$\mathbb{P}(\mathbf{x})$
$\mathbb{P}(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y})$	$\mathbb{P}(\mathbf{x} \mid \mathbf{y})$
$\forall \mathbf{x}. \mathbb{P}(\mathbf{X} = \mathbf{x}) = f(\mathbf{x})$	$\mathbb{P}(\mathbf{X}) = f(\mathbf{X})$
$\forall \mathbf{x}. \forall \mathbf{y}. \mathbb{P}(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y}) = f(\mathbf{x}, \mathbf{y})$	$\mathbb{P}(\mathbf{X} \mid \mathbf{Y}) = f(\mathbf{X}, \mathbf{Y})$

- $\mathbb{P}(\mathbf{X})$ and $\mathbb{P}(\mathbf{X} \mid \mathbf{Y})$ are entire probability distributions
- Can be thought of as functions from $\mathrm{Dom}(\mathbf{X}) \to [0, 1]$ or $(\mathrm{Dom}(\mathbf{X}), \mathrm{Dom}(\mathbf{Y})) \to [0, 1]$, respectively
- $f_{\mathbf{y}}(\mathbf{X}) = \mathbb{P}(\mathbf{X} \mid \mathbf{y})$ is often referred to as *conditional probability distribution* (CPD)
- For discrete variables, may be represented as a table (CPT)

Conditional independence

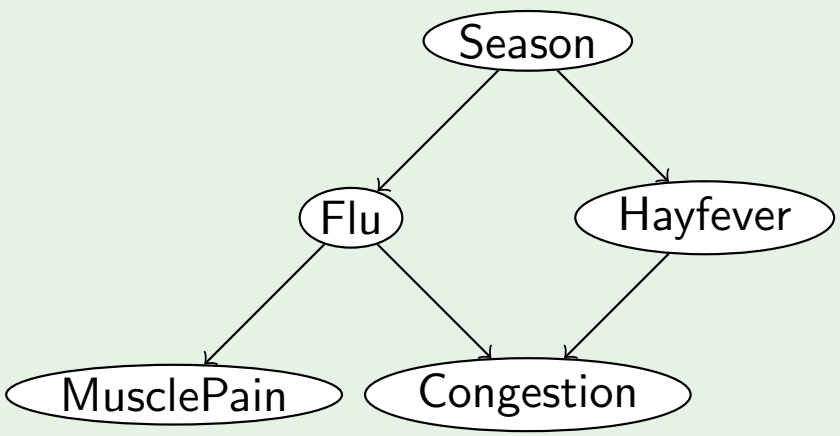
Definition

Let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be sets of random variables. $\mathbf{X}$ and $\mathbf{Y}$ are said to be *conditionally independent* given $\mathbf{Z}$ if and only if

$$\mathbb{P}(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z}) = \mathbb{P}(\mathbf{X} \mid \mathbf{Z})\,\mathbb{P}(\mathbf{Y} \mid \mathbf{Z}).$$

We write $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$ for this conditional independence statement. If $\mathbf{Z} = \emptyset$, we write $(\mathbf{X} \perp \mathbf{Y})$ for *marginal independence*.

Example



$$(F \perp H \mid S), (C \perp S \mid F, H)$$
$$(M \perp H, C, S \mid F)$$

$$\begin{aligned}&\mathbb{P}(S, F, H, M, C)\\ &= \mathbb{P}(S) \cdot \mathbb{P}(F \mid S) \cdot \mathbb{P}(H \mid S)\\ &\quad \cdot \mathbb{P}(C \mid F, H) \cdot \mathbb{P}(M \mid F)\end{aligned}$$

Properties of conditional independence

Theorem

In general, $(\mathbf{X} \perp \mathbf{Y})$ *does not imply nor is implied by* $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$

The following relationships hold:

$$
\begin{aligned}
(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) &\iff (\mathbf{Y} \perp \mathbf{X} \mid \mathbf{Z}) && (\textit{symmetry}) \\
(\mathbf{X} \perp \mathbf{Y}, \mathbf{W} \mid \mathbf{Z}) &\implies (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) && (\textit{decomposition}) \\
(\mathbf{X} \perp \mathbf{Y}, \mathbf{W} \mid \mathbf{Z}) &\implies (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}, \mathbf{W}) && (\textit{weak union}) \\
(\mathbf{X} \perp \mathbf{W} \mid \mathbf{Z}, \mathbf{Y}) \wedge (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) &\implies (\mathbf{X} \perp \mathbf{Y}, \mathbf{W} \mid \mathbf{Z}) && (\textit{contraction})
\end{aligned}
$$

For positive distributions and mutally disjoint sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$*:*

$$(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}, \mathbf{W}) \wedge (\mathbf{X} \perp \mathbf{W} \mid \mathbf{Z}, \mathbf{Y}) \implies (\mathbf{X} \perp \mathbf{Y}, \mathbf{W} \mid \mathbf{Z}) \quad (\textit{intersection})$$

Proof.

Discussed in exercise group. □

Querying a distribution (1)

Consider a joint distribution on a set of variables $\mathcal{X}$

- Let $\mathbf{E} \subseteq \mathcal{X}$ be a set of *evidence variables* that takes values $\mathbf{e}$
- Let $\mathbf{W} = \mathcal{X} - \mathbf{E}$ be the set of *latent variables*
- Let $\mathbf{Y} \subseteq \mathbf{W}$ be a set of *query variables*
- Let $\mathbf{Z} = \mathbf{W} - \mathbf{Y}$ be the set of *non-query variables*

Example

- $\mathcal{X} = \{\, \text{Season}, \text{Congestion}, \text{MusclePain}, \text{Flu}, \text{Hayfever} \,\}$
- $\mathbf{E} = \{\, \text{Season}, \text{Congestion}, \text{MusclePain} \,\}$
- $\mathbf{e} = \{\, \text{Spring}, \text{Yes}, \text{No} \,\}$
- $\mathbf{W} = \{\, \text{Flu}, \text{Hayfever} \,\}$
- $\mathbf{Y} = \{\, \text{Flu} \,\}$
- $\mathbf{Z} = \{\, \text{Hayfever} \,\}$

Querying a distribution (2)

1. Conditional probability query
   - Compute the *posterior distribution* of the query variables $\mathbb{P}(\mathbf{Y} \mid \mathbf{e})$
2. MAP query
   - Compute the most likely value of the latent variables $\text{MAP}(\mathbf{W} \mid \mathbf{e}) = \text{argmax}_{\mathbf{w}} \mathbb{P}(\mathbf{w} \mid \mathbf{e}) = \text{argmax}_{\mathbf{w}} \mathbb{P}(\mathbf{w}, \mathbf{e})$
3. Marginal MAP query
   - Compute the most likely value of the query variables $\text{MAP}(\mathbf{Y} \mid \mathbf{e}) = \text{argmax}_{\mathbf{y}} \mathbb{P}(\mathbf{y} \mid \mathbf{e}) = \text{argmax}_{\mathbf{y}} \sum_{\mathbf{z}} \mathbb{P}(\mathbf{y}, \mathbf{z}, \mathbf{e})$

Example

$\mathbb{P}(\mathbf{W} \mid \mathbf{e})$	Flu	¬Flu
Hayfever	5%	35%
¬Hayfever	40%	20%

1. $\mathbb{P}(\text{Flu} \mid \text{Spring}, \text{Congestion}, \neg\text{MusclePain}) \rightarrow$ Yes (45%), No (55%)
2. $\text{MAP}(\text{Flu}, \text{Hayfever} \mid \text{Spring}, \text{Congestion}, \neg\text{MusclePain}) \rightarrow$ Only flu
3. $\text{MAP}(\text{Flu} \mid \text{Spring}, \text{Congestion}, \neg\text{MusclePain}) \rightarrow$ No flu (!)

Querying graphical models

- Graphical models induce conditional independences
- Queries reason about dependencies between variables

Can we evaluate queries more efficiently given a graphical model and its associated independences?

Example

Independence properties help inference!

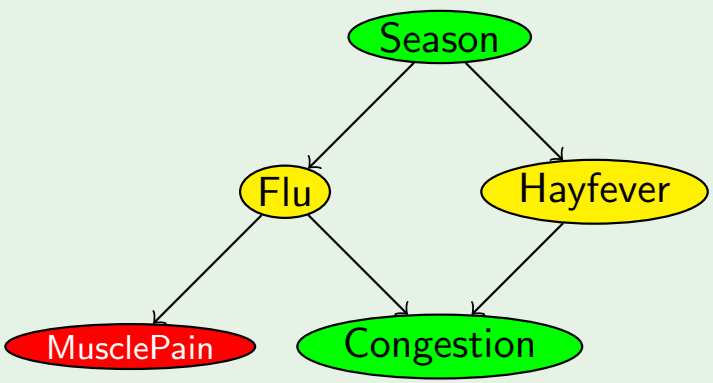


Table known to satisfy $(F \perp H \mid \mathbf{E})$

$\mathbb{P}(\mathbf{W} \mid \mathbf{e})$	Flu	¬Flu	
Hayfever	24%	16%	40%
¬Hayfever	36%	24%	60%
	60%	40%	

Thus, for example, monotonicity is now known to hold for MAP:

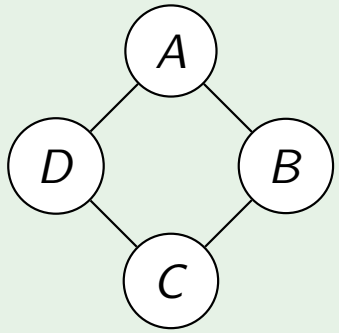
$$\text{MAP}(\text{Flu}, \text{Hayfever} \mid \mathbf{E}) = (\text{MAP}(\text{Flu} \mid \mathbf{E}), \text{MAP}(\text{Hayfever} \mid \mathbf{E}))$$

Outline

Misconception example

Example

- Alice, Bob, Charles, and Debbie study in pairs for the SUM exam



- Lecturer misspoke in class, giving rise to a possible misconception
- Some students figured out the problem, others did not

Which of the students has the misconception?

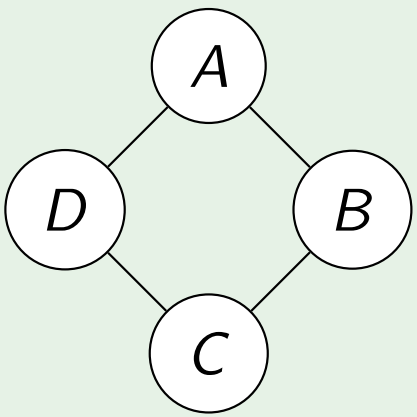
- If A does not have the misconception, he may help B and D
  $\rightarrow$ Students influence each other
- If A has the misconception, he may be helped by B and D
  $\rightarrow$ Influence has no natural "direction"
- A does not study with C $\rightarrow$ No *direct* influence between A and C

Markov network

Definition

A *Markov network* is an undirected graph $\mathcal{H} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X}$ is a set of random variables and $\mathcal{E} \subseteq \mathcal{X} \times \mathcal{X}$ is the set of edges.

Example



- $\mathcal{X} = \{ A, B, C, D \}$
- $\mathcal{E} = \{ (A, B), (B, C), (C, D), (D, A) \}$

We will see that Markov networks encode a set of conditional independence assumptions between its variables.

Local models

Definition

Let $\mathbf{D}$ be a set of random variables. A *factor* ϕ is a function from $\text{Dom}(\mathbf{D}) \to \mathbb{R}$. A factor is *nonnegative* if has range $\mathbb{R}^+$. The set $\mathbf{D}$ is called the *scope* of the factor and is denoted $\text{Scope}[\phi]$.

We restrict attention to nonnegative factors.

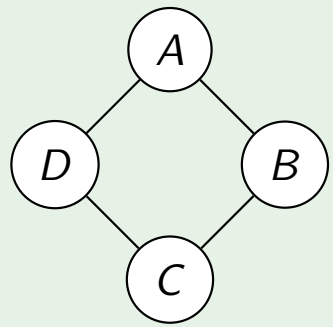
Example

A	B	ϕ_1
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

B	C	ϕ_2
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

C	D	ϕ_3
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

D	A	ϕ_4
d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100



- Factors describe "compatibility" between values (not normalized)
- ϕ_1: More "weight" when A and B agree than when they disagree
- ϕ_1: More weight when A and B are both right than when both are wrong
- ϕ_1: If they disagree, more weight when A is right than when B is right

Combining local models

Definition

Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ be three disjoint sets of random variables and let $\phi_1(\mathbf{X}, \mathbf{Y})$ and $\phi_2(\mathbf{Y}, \mathbf{Z})$ be two factors. The *factor product* $\psi = \phi_1 \times \phi_2$ is given by the factor $\psi : \mathrm{Dom}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \to \mathbb{R}$ with

$$\psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \phi_1(\mathbf{X}, \mathbf{Y}) \cdot \phi_2(\mathbf{Y}, \mathbf{Z}).$$

Example

A	B	ϕ_1
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

B	C	ϕ_2
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

A	B	C	ψ
a^0	b^0	c^0	3000
a^0	b^0	c^1	30
a^0	b^1	c^0	5
a^0	b^1	c^1	500
a^1	b^0	c^0	100
a^1	b^0	c^1	1
a^1	b^1	c^0	10
a^1	b^1	c^1	1000

Factor products combine local models by "joining" factors on the common part $\mathbf{Y}$.

Factor products and the product rule of probability

Recall the product rule of probability

$$\mathbb{P}(\mathbf{X}, \mathbf{Y}) = \mathbb{P}(\mathbf{Y})\,\mathbb{P}(\mathbf{X} \mid \mathbf{Y}).$$

Example

MusclePain

M	$\mathbb{P}$
Yes	0.1
No	0.9

Flu | MusclePain

M	F	$\mathbb{P}$
Yes	Yes	0.8
Yes	No	0.2
No	Yes	0.1
No	No	0.9

Flu, MusclePain

M	F	$\mathbb{P}$
Yes	Yes	0.08
Yes	No	0.02
No	Yes	0.09
No	No	0.81

- Set $\phi_1(\text{MusclePain}) = \mathbb{P}(\text{MusclePain})$
- Set $\phi_2(\text{MusclePain}, \text{Flu}) = \mathbb{P}(\text{Flu} \mid \text{MusclePain})$
- Set $\psi(\text{MusclePain}, \text{Flu}) = \mathbb{P}(\text{MusclePain}, \text{Flu})$
- Then $\psi = \phi_1 \times \phi_2$

Factor products generalize the product rule of probability.

Gibbs distribution

Definition

A distribution $\mathbb{P}_\Phi$ is a *Gibbs distribution* parameterized by a set of factors $\Phi = \{\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)\}$ if it is defined by

$$\mathbb{P}_\Phi(X_1, \ldots, X_n) = \frac{1}{Z}\tilde{\mathbb{P}}_\Phi(X_1, \ldots, X_n)$$

$$\tilde{\mathbb{P}}_\Phi(X_1, \ldots, X_n) = \phi_1(\mathbf{D}_1) \times \phi_2(\mathbf{D}_2) \times \cdots \times \phi_m(\mathbf{D}_m)$$

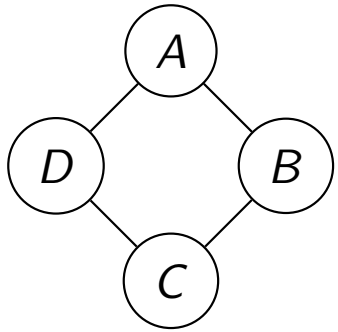
$$Z = \sum_{X_1, \ldots, X_n} \tilde{\mathbb{P}}_\Phi(X_1, \ldots, X_n)$$

Here, $\tilde{\mathbb{P}}_\Phi(X_1, \ldots, X_n)$ is an *unnormalized measure* and Z a normalizing constant called the *partitioning function*.

- Factors *contribute* to the overall joint distribution
- Overall dist. takes into consideration the contribution from *all* factors

A set of factors defines a Gibbs distribution, i.e., a joint probability distribution over all variables.

Gibbs distribution for Misconception example



A	B	ϕ_1
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

B	C	ϕ_2
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

C	D	ϕ_3
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

D	A	ϕ_4
d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

A	B	C	D	$\tilde{\mathbb{P}}$	$\mathbb{P}$
a_0	b_0	c_0	d_0	300,000	0.04
a_0	b_0	c_0	d_1	300,000	0.04
a_0	b_0	c_1	d_0	300,000	0.04
a_0	b_0	c_1	d_1	30	$4.1 \cdot 10^{-6}$
a_0	b_1	c_0	d_0	500	$6.9 \cdot 10^{-5}$
a_0	b_1	c_0	d_1	500	$6.9 \cdot 10^{-5}$
a_0	b_1	c_1	d_0	5,000,000	0.69
a_0	b_1	c_1	d_1	500	$6.9 \cdot 10^{-5}$
a_1	b_0	c_0	d_0	100	$1.4 \cdot 10^{-5}$
a_1	b_0	c_0	d_1	1,000,000	0.14
a_1	b_0	c_1	d_0	100	$1.4 \cdot 10^{-5}$
a_1	b_0	c_1	d_1	100	$1.4 \cdot 10^{-5}$
a_1	b_1	c_0	d_0	10	$1.4 \cdot 10^{-6}$
a_1	b_1	c_0	d_1	100,000	0.014
a_1	b_1	c_1	d_0	100,000	0.014
a_1	b_1	c_1	d_1	100,000	0.014

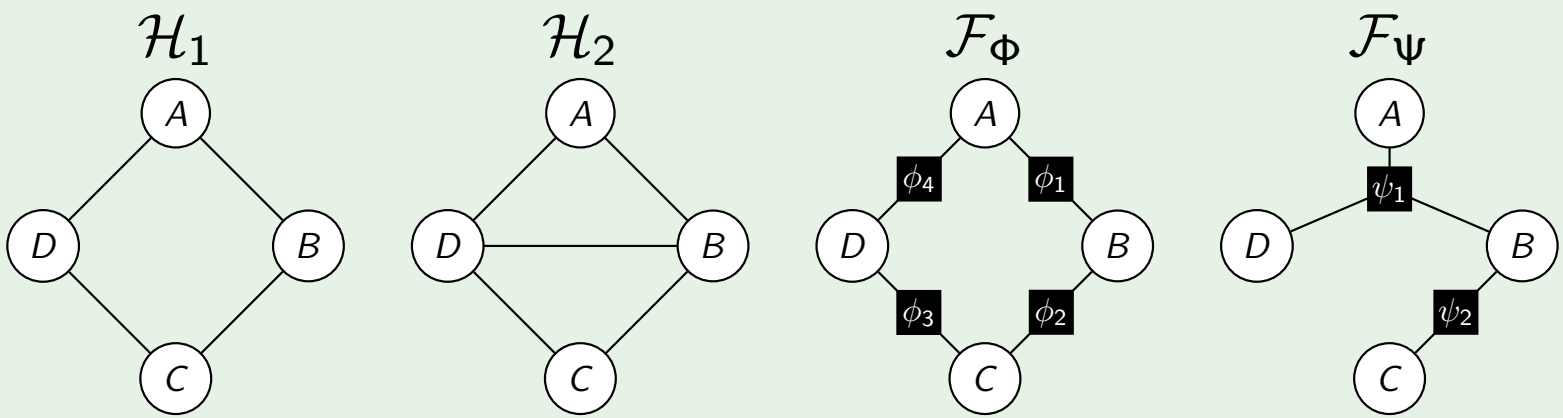
$Z = 7,201,840$

Factorization and factor graphs

Definition

A distribution $\mathbb{P}_\Phi$ with $\Phi = \{\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)\}$ *factorizes* over a Markov network $\mathcal{H}$ if each $\mathbf{D}_i$ is a complete subgraph of $\mathcal{H}$. The factors ϕ_i are often called *clique potentials*.

Example



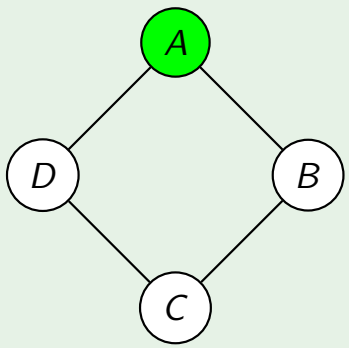
- $\Phi = \{\phi_1(A, B), \phi_2(B, C), \phi_3(C, D), \phi_4(D, A)\}$
- $\Psi = \{\psi_1(A, B, D), \psi_2(B, C)\}$
- $\mathbb{P}_\Phi$ factorizes over both $\mathcal{H}_1$ and $\mathcal{H}_2$
- $\mathbb{P}_\Psi$ factorizes over only $\mathcal{H}_2$

Active paths

Definition

Let X_1—. . .—X_k be a path in $\mathcal{H} = (\mathcal{X}, \mathcal{E})$. Let $\mathbf{Z} \subseteq \mathcal{X}$ be a set of observed variables. The path X_1—. . .—X_k is *active* given $\mathbf{Z}$ if $X_i \notin \mathbf{Z}$ for $1 \leq i \leq k$.

Example



All active paths given A:

- D–C
- C–B
- D–C–B

Some inactive paths given A:

- D–A–B
- C–D–A–B

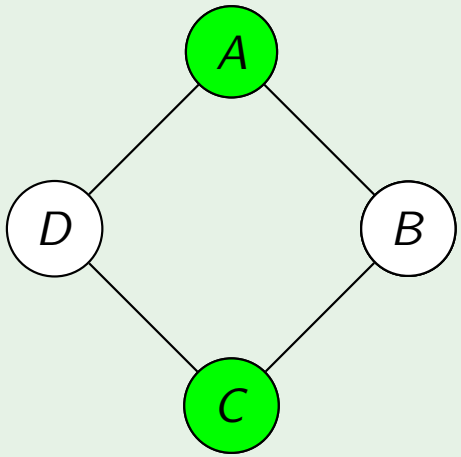
Separation and independencies for Markov networks

Definition

We say that a set of nodes $\mathbf{Z}$ *separates* $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{H}$, denoted $\text{sep}_{\mathcal{H}}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z})$, if there is no active path between any node in $\mathbf{X}$ and any node in $\mathbf{Y}$ given $\mathbf{Z}$. We associate with $\mathcal{H}$ the following set of independencies:

$$\mathcal{I}(\mathcal{H}) = \{ (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) : \text{sep}_{\mathcal{H}}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z}) \}$$

Example



- $\emptyset$ does not separate any nodes
- $\{ A \}$ does not separate any nodes
- $\{ A, C \}$ separates $\{ B \}$ and $\{ D \}$
- $\{ A, B, C \}$ does not separate any nodes

$$\mathcal{I}(\mathcal{H}) = \{ (B \perp D \mid A, C), (D \perp B \mid A, C) \\ (A \perp C \mid B, D), (C \perp A \mid B, D) \}$$

Relationship Gibbs distributions and Markov networks

Definition

- Let $\mathbb{P}$ be a probability distribution over $\mathbf{X}$. Define $\mathcal{I}(\mathbb{P})$ to be the set of independence assertions of the form $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z})$ that hold in $\mathbb{P}$.
- A Markov network $\mathcal{H}$ is an *I-map* for $\mathbb{P}$ if $\mathcal{I}(\mathcal{H}) \subseteq \mathcal{I}(\mathbb{P})$.

Theorem

Soundness ($\rightarrow$)

Let $\mathbb{P}$ be a distribution and $\mathcal{H}$ be a Markov network over $\mathcal{X}$. If $\mathbb{P}$ is a Gibbs distribution that factorizes over $\mathcal{H}$, then $\mathcal{H}$ is an I-map for $\mathbb{P}$.

Theorem (Hammersley-Clifford theorem)

Soundness ($\leftarrow$)

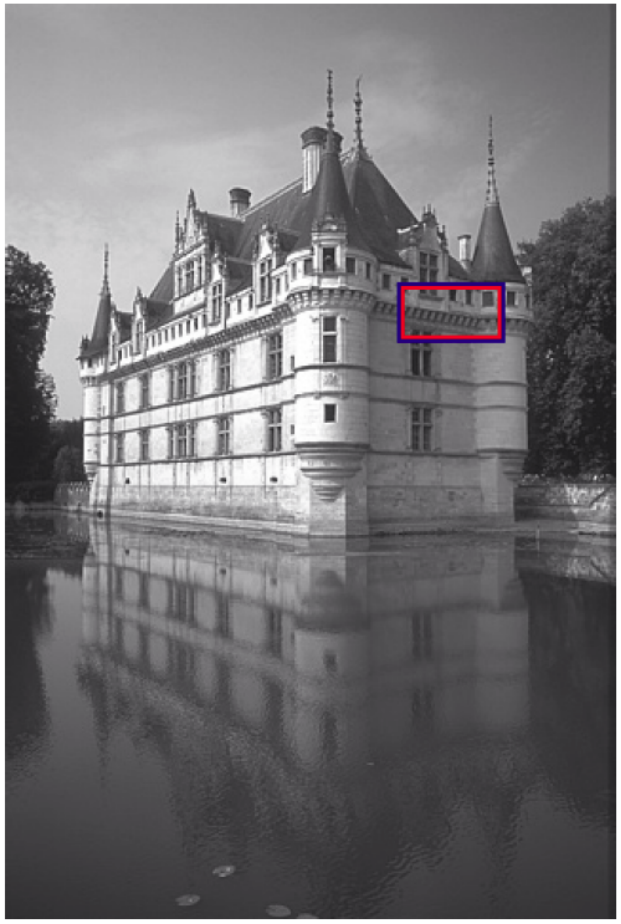
Let $\mathbb{P}$ be a positive distribution and $\mathcal{H}$ be a Markov network over $\mathcal{X}$. If $\mathcal{H}$ is an I-map for $\mathbb{P}$, then $\mathbb{P}$ is a Gibbs distribution that factorizes over $\mathcal{H}$.

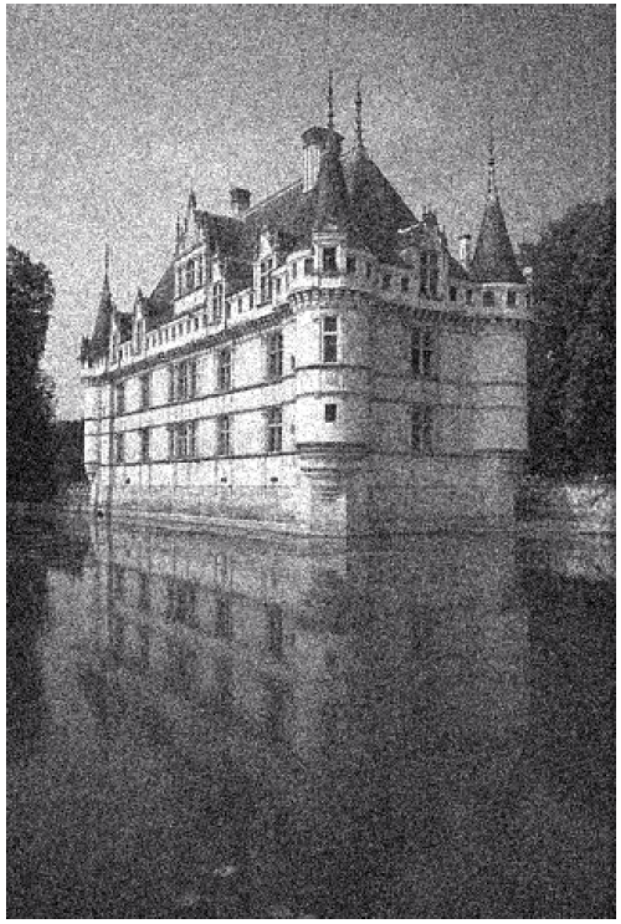
Theorem

Completeness

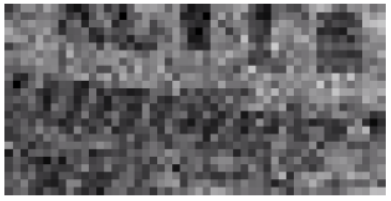
If X and Y are not separated given $\mathbf{Z}$ in $\mathcal{H}$, then X and Y are dependent for some distribution $\mathbb{P}$ that factorizes over $\mathcal{H}$.

Application: Image denoising





Original

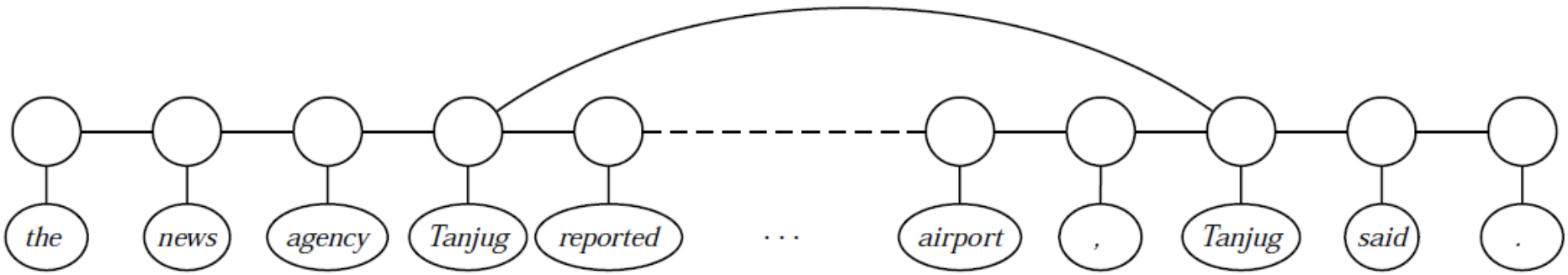


Noisy image

Denoised image

Application: Stanford Named Entity Recognizer

Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names.



- Local evidence often strong clue for label
- Long-range evidence (label consistency) helps when local evidence is insufficient

Outline

Outline

Semantics of Markov logic networks

Definition

A Markov logic network $L = \{ (F_i, w_i) \}$ is a template for constructing Markov networks. Given a set of constants C, a *ground Markov logic* $M_{L,C}$ specifies a distribution over the possible worlds as follows

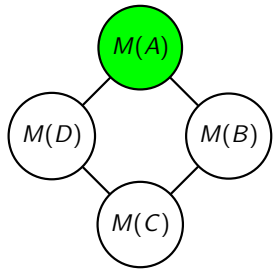
$$\mathbb{P}(\mathbf{X} = \mathbf{x}) \propto \exp\left[\sum_i w_i n_i(\mathbf{x})\right],$$

where $n_i(\mathbf{x})$ is the number of "true groundings" of formula F_i in the possible world $\mathbf{x}$.

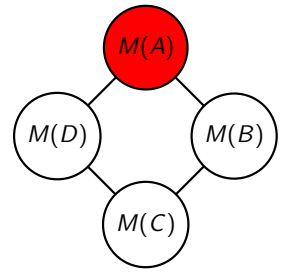
- A possible world $\mathbf{x}$ is likely if
  1. It satisfies many groundings with positive weight
  2. If satisfies few groundings with negative weight
  3. It satisfies groundings with high positive weight
  4. It does not satisfy groundings with high negative weight

How many true groundings does a formula have?

- $F_1 = M(A)$

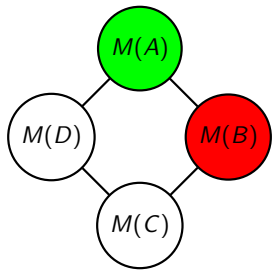


$n_1 = 1$

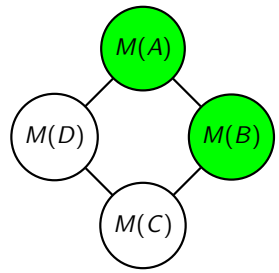


$n_1 = 0$

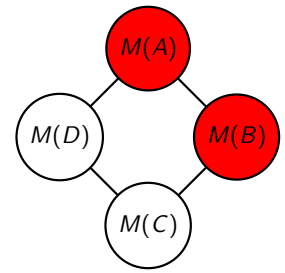
- $F_2 = M(A) \vee M(B)$



$n_2 = 1$



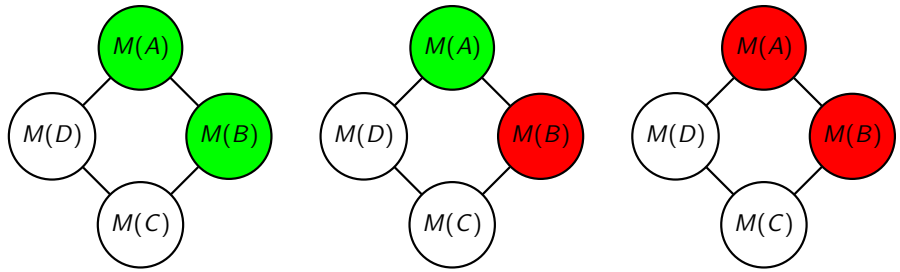
$n_2 = 1$



$n_2 = 0$

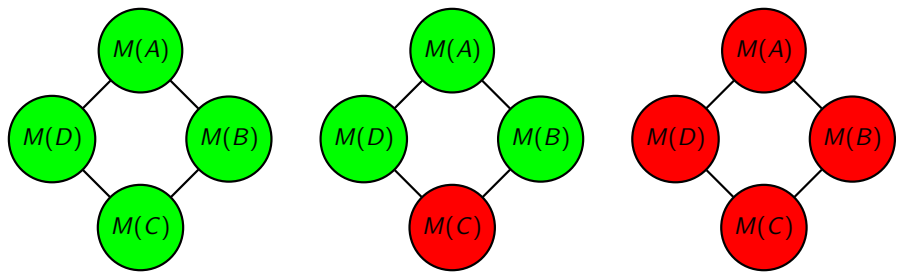
How many true groundings does a formula have? (2)

- $F_3 = M(A) \wedge M(B)$



FOL:	1	0	0	(strict)
MLN:	$n_3 = 2$	$n_3 = 1$	$n_3 = 0$	(smoothed)

- $F_4 = \forall x.M(x)$

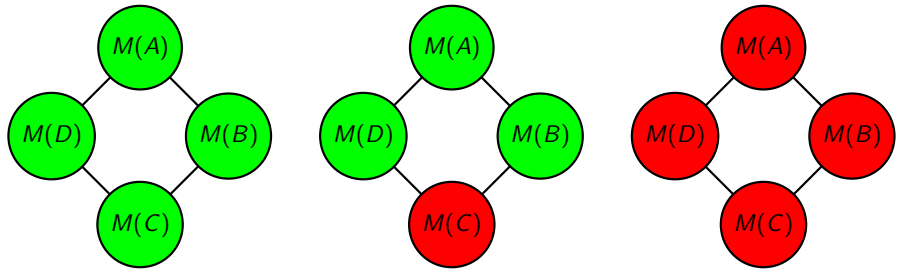


FOL:	1	0	0	(strict)
MLN:	$n_4 = 4$	$n_4 = 3$	$n_4 = 0$	(smoothed)

Conjunctions in FOL are sensitive to noise: If just one of the conjuncts is unsatisfied, the formula is also unsatisfied. MLNs count how many of the conjuncts are true and thus are less sensitive to noise.

How many true groundings does a formula have? (3)

- $F_5 = \exists x.M(x)$



FOL:	1	1	0	(strict)
MLN:	$n_4 = 1$	$n_4 = 1$	$n_4 = 0$	(strict)

Disjunctions in FOL are insensitive to noise, so we are fine.

Grounding a formula in Markov logic

Let F be a formula and $C = \{ c_1, \ldots, c_d \}$ be a set of constants.
Conceptually, we obtain the set $G(F)$ of ground formulas as follows:

1. Whenever a subformulas of form $\exists x.F'(x)$ occurs, replace by $(F'(c_1) \vee \cdots \vee F'(c_d))$
2. Convert the formula to form $\forall \mathbf{x}.F'(\mathbf{x})$, where F' is in conjunctive normal form and is quantifier-free, optionally simplify, denote result by $\mathrm{cnf}(F)$
3. For all $\mathbf{c} \in C^{|\mathbf{x}|}$, set $G(F, \mathbf{c}) = \{ G : G \text{ is a clause in } F'(\mathbf{c}) \}$
4. Set $G(F) = \left\{ G(F, \mathbf{c}) : \mathbf{c} \in C^{|\mathbf{x}|} \right\}$

Example

- $C = \{ A, B \}$
- $F_1 = \forall x.\mathrm{Smokes}(x) \implies \mathrm{Cancer}(x)$
  1. No existential quantifiers $\rightarrow$ nothing to do
  2. $\mathrm{cnf}(F_1) = \forall x.\neg \mathrm{S}(x) \vee \mathrm{C}(x)$
  3. $G(F_1, A) = \{ \neg \mathrm{S}(A) \vee \mathrm{C}(A) \}$
     $G(F_1, B) = \{ \neg \mathrm{S}(B) \vee \mathrm{C}(B) \}$
  4. $G(F_1) = \{ \{ \neg \mathrm{S}(A) \vee \mathrm{C}(A) \}, \{ \neg \mathrm{S}(B) \vee \mathrm{C}(B) \} \}$

Grounding a formula (example)

Example

- $C = \{ A, B \}$
- $F_2 = \forall x.\forall y.\text{Friends}(x, y) \implies (\text{Smokes}(x) \iff \text{Smokes}(y))$
  1. No existential quantifiers $\rightarrow$ nothing to do
  2. $\text{cnf}(F_2) = \forall x.\forall y.[\neg \text{F}(x, y) \vee \text{S}(x) \vee \neg \text{S}(y)] \wedge [\neg \text{F}(x, y) \vee \neg \text{S}(x) \vee \text{S}(y)]$
  3. $G(F_2, (A, A)) = \{ \neg \text{F}(A, A) \vee \text{S}(A) \vee \neg \text{S}(A), \neg \text{F}(A, A) \vee \neg \text{S}(A) \vee \text{S}(A) \}$
     $G(F_2, (A, B)) = \{ \neg \text{F}(A, B) \vee \text{S}(A) \vee \neg \text{S}(B), \neg \text{F}(A, B) \vee \neg \text{S}(A) \vee \text{S}(B) \}$
     $G(F_2, (B, A)) = \{ \neg \text{F}(B, A) \vee \text{S}(B) \vee \neg \text{S}(A), \neg \text{F}(B, A) \vee \neg \text{S}(B) \vee \text{S}(A) \}$
     $G(F_2, (B, B)) = \{ \neg \text{F}(B, B) \vee \text{S}(A) \vee \neg \text{S}(B), \neg \text{F}(B, B) \vee \neg \text{S}(A) \vee \text{S}(B) \}$
  4. $G(F_2) = \{\{ \neg \text{F}(A, A) \vee \text{S}(A) \vee \neg \text{S}(A), \neg \text{F}(A, A) \vee \neg \text{S}(A) \vee \text{S}(A) \},$
     $\{ \neg \text{F}(A, B) \vee \text{S}(A) \vee \neg \text{S}(B), \neg \text{F}(A, B) \vee \neg \text{S}(A) \vee \text{S}(B) \},$
     $\{ \neg \text{F}(B, A) \vee \text{S}(B) \vee \neg \text{S}(A), \neg \text{F}(B, A) \vee \neg \text{S}(B) \vee \text{S}(A) \},$
     $\{ \neg \text{F}(B, B) \vee \text{S}(A) \vee \neg \text{S}(B), \neg \text{F}(B, B) \vee \neg \text{S}(A) \vee \text{S}(B) \} \}$

Grounding a Markov logic network

Given an MLN $\{(F_i, w_i)\}$ and a set of constants C.

1. Create a Boolean variable $R(\mathbf{c})$ for each predicate that occurs in one of the formulas and each $\mathbf{c} \in C^m$, where m is the arity of the relation
2. For each formula F_i
   1. Ground F_i to obtain $G(F_i)$
   2. For each ground set of clauses $G(F_i, \mathbf{c}) \in G(F_i)$
      1. Split weight evenly among clauses: $w_i' = w_i / |G(F_i, \mathbf{c})|$
      2. For each clause F_{ij} in $G(F_i, \mathbf{c})$, create a factor

$$\phi(\mathbf{D}_{ij}) = w_i' f_{ij}(\mathbf{D}_{ij}),$$

where $\mathbf{D}_{ij}$ is the set of variables that occur in F_{ij}, and

$$f_{ij}(\mathbf{D}_{ij}) = \begin{cases} 1 & \text{if } j\text{-th clause in in } G(F_i, \mathbf{c}) \text{ is satisfied for assignment } \mathbf{D}_{ij} \\ 0 & \text{otherwise} \end{cases}$$

is an "indicator feature" with weight w_i'.

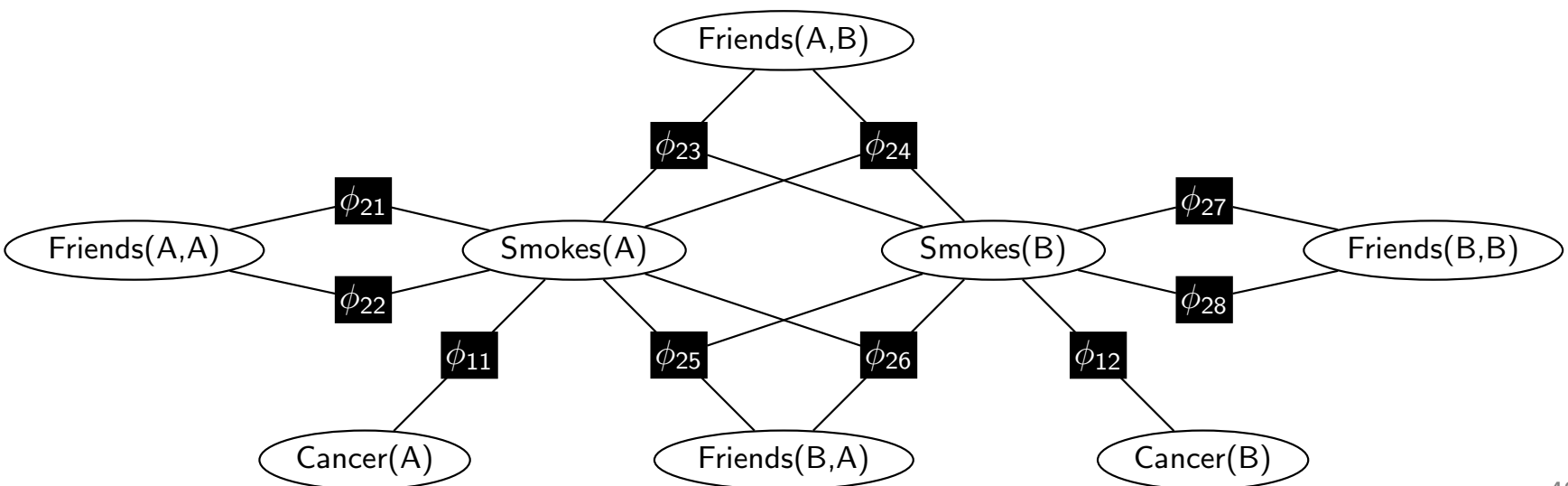
The weight of a ground CNF formula is split evenly among its clauses.

Grounding a Markov logic network (example)

F_1: 1.5 { Smoking causes cancer
$\forall x.\text{Smokes}(x) \implies \text{Cancer}(x)$

F_2: 1.1 { Friends have similar smoking habits
$\forall x.\forall y.\text{Friends}(x, y) \implies (\text{Smokes}(x) \iff \text{Smokes}(y))$

$$
\begin{aligned}
C &= \{A, B\} \\
G(F_1) &= \{\{\neg \mathrm{S}(A) \vee \mathrm{C}(A)\}, \\
&\quad\ \{\neg \mathrm{S}(B) \vee \mathrm{C}(B)\}\} \\
G(F_2) &= \{\{\neg \mathrm{F}(A,A) \vee \mathrm{S}(A) \vee \neg \mathrm{S}(A), \\
&\quad\ \neg \mathrm{F}(A,A) \vee \neg \mathrm{S}(A) \vee \mathrm{S}(A)\}, \\
&\quad\ \{\neg \mathrm{F}(A,B) \vee \mathrm{S}(A) \vee \neg \mathrm{S}(B), \\
&\quad\ \neg \mathrm{F}(A,B) \vee \neg \mathrm{S}(A) \vee \mathrm{S}(B)\}, \\
&\quad\ \{\neg \mathrm{F}(B,A) \vee \mathrm{S}(B) \vee \neg \mathrm{S}(A), \\
&\quad\ \neg \mathrm{F}(B,A) \vee \neg \mathrm{S}(B) \vee \mathrm{S}(A)\}, \\
&\quad\ \{\neg \mathrm{F}(B,B) \vee \mathrm{S}(A) \vee \neg \mathrm{S}(B), \\
&\quad\ \neg \mathrm{F}(B,B) \vee \neg \mathrm{S}(A) \vee \mathrm{S}(B)\}\}
\end{aligned}
\qquad
\begin{aligned}
f_{11}, w'_{11} &= 1.50 \\
f_{12}, w'_{12} &= 1.50 \\
f_{21}, w'_{21} &= 0.55 \\
f_{22}, w'_{22} &= 0.55 \\
f_{23}, w'_{23} &= 0.55 \\
f_{24}, w'_{24} &= 0.55 \\
f_{25}, w'_{25} &= 0.55 \\
f_{26}, w'_{26} &= 0.55 \\
f_{27}, w'_{27} &= 0.55 \\
f_{28}, w'_{28} &= 0.55
\end{aligned}
$$



Outline

Log-linear model

Definition

A positive distribution $\mathbb{P}$ is a log-linear model over a Markov network $\mathcal{H}$ if it is associated with

- a set of *features* $\mathcal{F} = \{ f_1(\mathbf{D}_1), \ldots, f_m(\mathbf{D}_m) \}$, where each $\mathbf{D}_i$ is a complete subgraph in $\mathcal{H}$
- a set of weights $w_1, \ldots, w_m$

such that

$$\mathbb{P}(X_1, \ldots, X_n) \propto \exp\left[\sum_{i=1}^{m} w_i f_i(\mathbf{D}_i)\right].$$

The terms $\epsilon_i(\mathbf{D}_i) = -w_i f_i(\mathbf{D}_i)$ are called *energy functions*.

$\log \mathbb{P}(X_1, \ldots, X_n)$ is a linear combination of the the features. The linearity allows us to detect and *eliminate* redundancy in the features (using standard linear algebra techniques).

From factors to features

Definition

Let $\mathbf{D}$ be a subset of variables. An *indicator feature* is a function $f(\mathbf{D}) : \mathbf{D} \to \{0, 1\}$.

Theorem

Every factor of a graphical model on discrete variables can be expressed in terms of a linear combination of weighted indicator features.

Proof (Boolean case).

Consider a factor $\phi(X_1, \ldots, X_k)$ on k Boolean variables. Let Θ be the set of all assignments of values to $X_1, \ldots, X_k$. Set

$$w_\theta = \ln \phi(X_1[\theta], \ldots, X_k[\theta]) \qquad \text{(constants)}$$

$$f_\theta(X_1, \ldots, X_k) = \begin{cases} 1 & \text{if } X_1 = X_1[\theta], \ldots, X_k = X_k[\theta] \\ 0 & \text{otherwise} \end{cases} \qquad \text{(indicator features)}$$

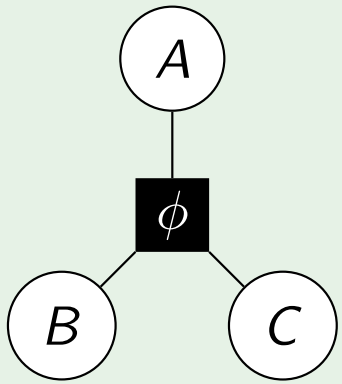
$$\ln \phi(X_1, \ldots, X_k) = \sum_{\theta \in \Theta} w_\theta f_\theta(X_1, \ldots, X_k) \qquad \text{(decomposition)}$$

□

From factors to features (example)

Example

Consider three friends with similiar interests and let A, B, C be Boolean variables that indicate whether each of the friends likes football.



ϕ

X_1	X_2	X_3	ϕ	w
F	F	F	10	2.3
F	F	T	1	0
⋮	⋮	⋮	⋮	⋮
T	F	T	1	0
T	T	F	1	0
T	T	T	10	2.3

We have

$$\ln \phi(A, B, C) = \sum_{\theta} w_\theta f_\theta(A, B, C) = 2.3 \cdot f_{FFF}(A, B, C) + 2.3 \cdot f_{TTT}(A, B, C).$$

Even more compact: $\ln \phi(A, B, C) = 2.3 \cdot I_{ABC \vee \neg A \neg B \neg C}$

From Gibbs distribution to log-linear models

Theorem

Every positive Gibbs distribution $\mathbb{P}$ over $\mathcal{H}$ on Boolean variables $X_1, \ldots, X_n$ has a log-linear model over $\mathcal{H}$ with only indicator features and vice versa.

Proof.

$$\begin{aligned}
\mathbb{P}(X_1, \ldots, X_n) &= \frac{1}{Z} \prod_{i=1}^{m} \phi_i(\mathbf{D}_i) \\
&= \frac{1}{Z} \exp\left[\sum_{i=1}^{m} \ln \phi_i(\mathbf{D}_i)\right] \\
&= \frac{1}{Z} \exp\left[\sum_{i=1}^{m} \sum_{\theta \in \Theta_{\mathbf{D}_i}} w_\theta f_\theta(\mathbf{D}_i)\right].
\end{aligned}$$

□

Markov logic networks are "templates" for constructing log-linear models. Any positive Gibbs distribution with finite-domain variables can be modeled.

Outline

Outline

Inference in probabilistic graphical models

- Recall the queries of interest
  1. Conditional probability query
  2. MAP query
  3. Marginal MAP query

Definition

Let $\mathbb{P}_\Phi$ be a Gibbs distribution over variables $\{X, X_1, \ldots, X_n\}$.

1. The $\mathbb{P}_\Phi$-*decision problem* asks whether $\mathbb{P}_\Phi(X = x) > 0$,
2. The $\mathbb{P}_\Phi$-*probability computation problem* asks for $\mathbb{P}_\Phi(X = x)$.

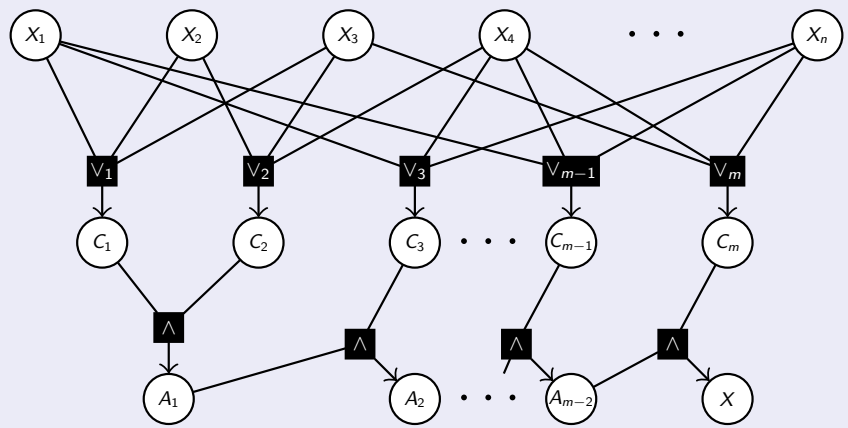
Complexity of inference in probabilistic graphical models

Theorem

The $\mathbb{P}_\Phi$-decision problem is NP-complete, $\mathbb{P}_\Phi$-probability computation is #P-hard.

Proof (by reduction from 3-SAT and #3-SAT).

Take a 3-SAT formula $\Psi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ over variables $\mathcal{X} = \{ X_1, X_2, \ldots, X_n \}$. Consider the following Gibbs distribution $\mathbb{P}_\Phi$ over Boolean variables:



Here, $\vee_i(C_i, \mathbf{X}_i) = 1$ if for assignment $\mathbf{X}_i$ the truth value of clause C_i equals variable C_i, else $\vee_i(C_i, \mathbf{X}_i) = 0$; similarly for $\wedge$-factors. $\mathbb{P}_\Phi$ can be computed in polynomial time in the size of Ψ. Assertion 1 follows since $\mathbb{P}_\Phi(X = \texttt{TRUE}) > 0$ if and only if Ψ is satisfiable. $\mathbb{P}_\Phi(X = \texttt{TRUE}) = \mathbb{P}(\Psi)$ where $\mathbb{P}(X_i = \texttt{TRUE}) = 1/2$ and the $\{ X_i \}$ are i.i.d. Assertion 2 follows since $\#\Psi = 2^n \mathbb{P}(\Psi) = 2^n \mathbb{P}_\Phi(X = \texttt{TRUE})$. □

Queries in Markov logic

- Standard PGM queries, e.g.,
  $\mathbb{P}(\,\text{Smokes(B)}, \text{Cancer(B)} \mid \text{Smokes(A)} \wedge \text{Friends(A,B)} \wedge \ldots\,)$
  $\rightarrow$ #P-hard
- More general queries of form "What is the probability that formula F_1 holds given that formula F_2 holds?", e.g.,
  $\mathbb{P}(\,\exists x.\text{Cancer(x)} \mid \forall x.\text{Smokes(x)}\,)$
- Let L be an MLN and C be a set of constants

$$\begin{aligned}\mathbb{P}(\,F_1 \mid F_2, L, C\,) &= \mathbb{P}(\,F_1 \mid F_2, M_{L,C}\,) \\ &= \frac{\mathbb{P}(\,F_1 \wedge F_2 \mid M_{L,C}\,)}{\mathbb{P}(\,F_2 \mid M_{L,C}\,)} \\ &= \frac{\sum_{\mathbf{x} \in \mathcal{X}_{F_1} \cap \mathcal{X}_{F_2}} \mathbb{P}(\,\mathcal{X} = \mathbf{x} \mid M_{L,C}\,)}{\sum_{\mathbf{x} \in \mathcal{X}_{F_2}} \mathbb{P}(\,\mathcal{X} = \mathbf{x} \mid M_{L,C}\,)},\end{aligned}$$

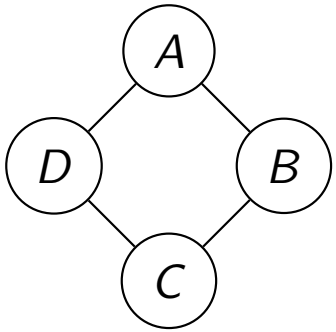
where $\mathcal{X}_F$ is the set of worlds in which F holds

We focus on standard PGM queries.

Outline

Naive approach

Exponential in number of variables!



A	B	ϕ_1
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

B	C	ϕ_2
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

C	D	ϕ_3
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

D	A	ϕ_4
d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

A	B	C	D	$\tilde{\mathbb{P}}$	$\mathbb{P}$
a_0	b_0	c_0	d_0	300,000	0.04
a_0	b_0	c_0	d_1	300,000	0.04
a_0	b_0	c_1	d_0	300,000	0.04
a_0	b_0	c_1	d_1	30	$4.1 \cdot 10^{-6}$
a_0	b_1	c_0	d_0	500	$6.9 \cdot 10^{-5}$
a_0	b_1	c_0	d_1	500	$6.9 \cdot 10^{-5}$
a_0	b_1	c_1	d_0	5,000,000	0.69
a_0	b_1	c_1	d_1	500	$6.9 \cdot 10^{-5}$
a_1	b_0	c_0	d_0	100	$1.4 \cdot 10^{-5}$
a_1	b_0	c_0	d_1	1,000,000	0.14
a_1	b_0	c_1	d_0	100	$1.4 \cdot 10^{-5}$
a_1	b_0	c_1	d_1	100	$1.4 \cdot 10^{-5}$
a_1	b_1	c_0	d_0	10	$1.4 \cdot 10^{-6}$
a_1	b_1	c_0	d_1	100,000	0.014
a_1	b_1	c_1	d_0	100,000	0.014
a_1	b_1	c_1	d_1	100,000	0.014

$Z = 7{,}201{,}840$

Grounding with evidence (1)

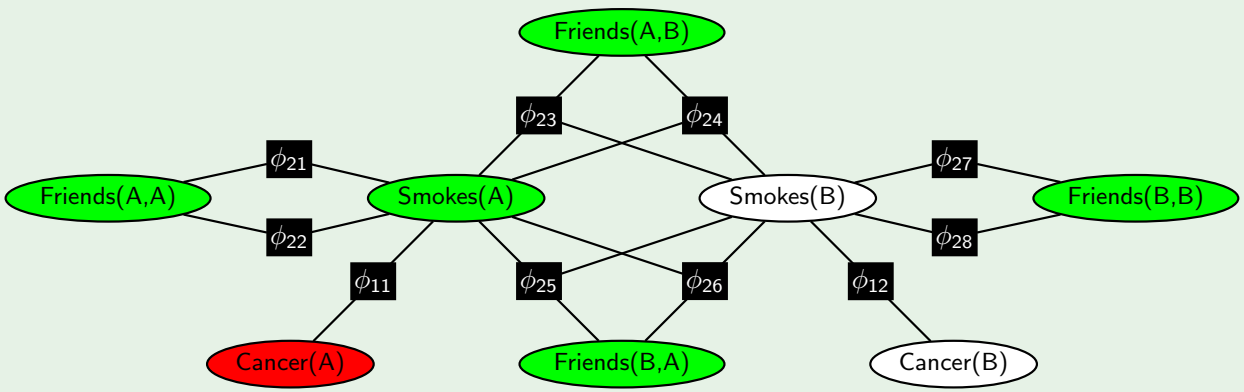
Denote by M the weighted ground clauses in a ground Markov logic network $M_{L,C}$. Given evidence $\mathbf{E}$, we can partition M into:

1. Clauses M_1 that involve only observed variables
2. Clauses M_2 that involve both observed and latent variables
3. Clauses M_3 that involve only latent variables

$$\log \mathbb{P}(\mathbf{W} \mid \mathbf{E}) = -\log Z + \sum_{\phi=(f,w)\in M} wf(\mathbf{W}_f, \mathbf{E}_f)$$

$$= -\log Z + \underbrace{\sum_{(f,w)\in M_1} wf(\mathbf{E}_f)}_{\text{Constant}} + \sum_{(f,w)\in M_2} wf(\mathbf{W}_f, \mathbf{E}_f) + \sum_{(f,w)\in M_3} wf(\mathbf{W}_f)$$

Example



$$M_1 = \{ \phi_{11}, \phi_{21}, \phi_{22} \}$$

$$M_2 = \{ \phi_{23}, \phi_{24}, \phi_{25}, \phi_{26}, \phi_{27}, \phi_{28} \}$$

$$M_3 = \{ \phi_{12} \}$$

Grounding with evidence (2)

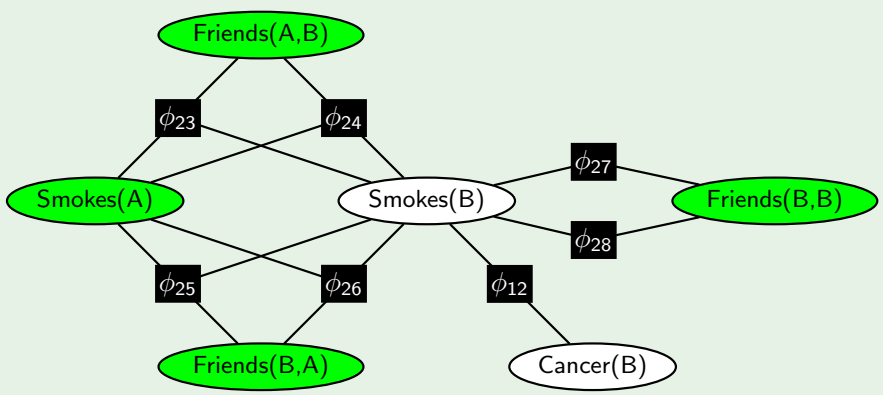
Denote by M the weighted ground clauses in a ground Markov logic network $M_{L,C}$. Given evidence $\mathbf{E}$, we can partition M into:

1. Clauses M_1 that involve only observed variables
2. Clauses M_2 that involve both observed and latent variables
3. Clauses M_3 that involve only latent variables

$$\log \mathbb{P}(\mathbf{W} \mid \mathbf{E}) = -\log Z + \sum_{\phi=(f,w)\in M} wf(\mathbf{W}_f, \mathbf{E}_f)$$

$$= -\log Z' + \underbrace{\sum_{(f,w)\in M_2} wf(\mathbf{W}_f, \mathbf{E}_f)}_{\text{Replace observed variables by their values}} + \sum_{(f,w)\in M_3} wf(\mathbf{W}_f)$$

Example



$M_2 = \{ \phi_{23}, \phi_{24}, \phi_{25}, \phi_{26}, \phi_{27}, \phi_{28} \}$

$M_3 = \{ \phi_{12} \}$

Grounding with evidence (3)

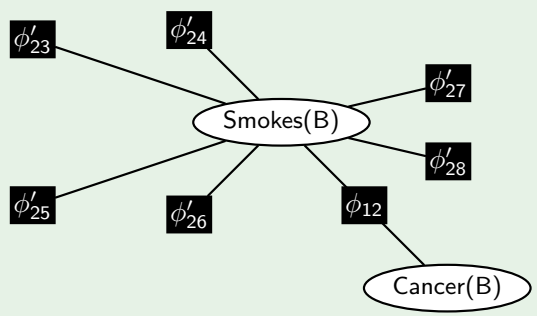
Denote by M the weighted ground clauses in a ground Markov logic network $M_{L,C}$. Given evidence $\mathbf{E}$, we can partition M into:

1. Clauses M_1 that involve only observed variables
2. Clauses M_2 that involve both observed and latent variables
3. Clauses M_3 that involve only latent variables

$$\begin{aligned}
\log \mathbb{P}(\mathbf{W} \mid \mathbf{E}) &= -\log Z + \sum_{\phi=(f,w)\in M} wf(\mathbf{W}_f, \mathbf{E}_f) \\
&= -\log Z' + \sum_{(f,w)\in M_2'} wf(\mathbf{W}_f) + \sum_{(f,w)\in M_3} wf(\mathbf{W}_f) \\
&= -\log Z' + \sum_{(f,w)\in M'} wf(\mathbf{W}_f)
\end{aligned}$$

No observed variables are left. Gives rise to efficient grounding methods.

Example



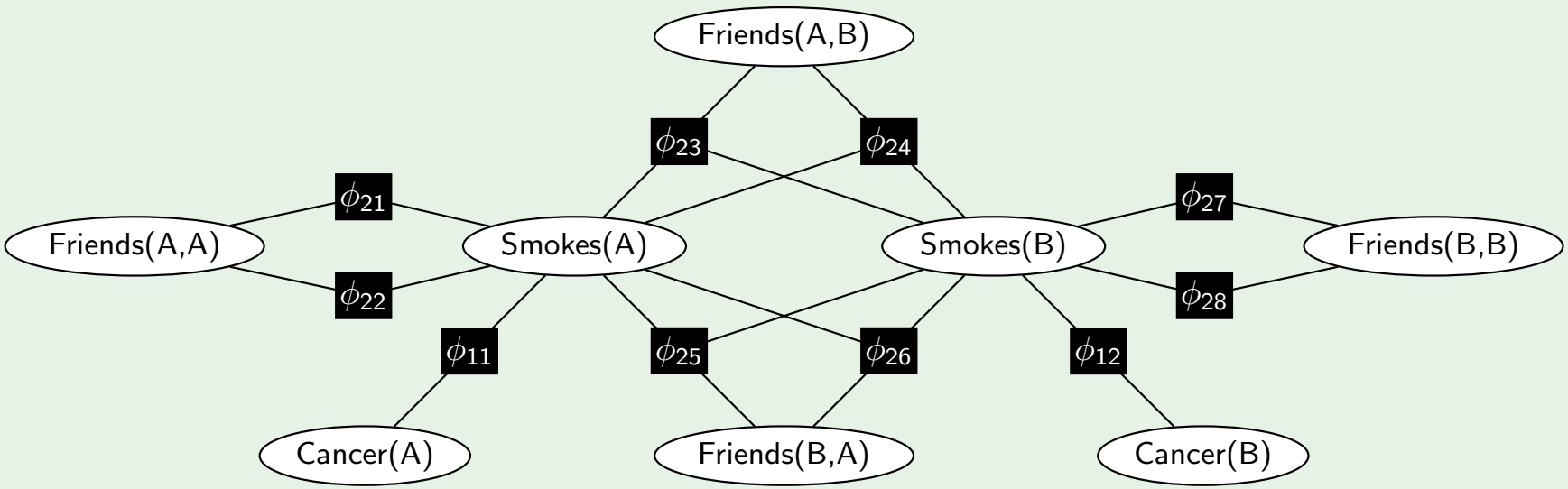
$$M_2' = \{ \phi'_{23}, \phi'_{24}, \phi'_{25}, \phi'_{26}, \phi'_{27}, \phi'_{28} \}$$
$$M_3 = \{ \phi_{12} \}$$
$$M' = M_2' \cup M_3$$

$$\phi_{24} = \neg F(A,B) \vee \neg S(A) \vee S(B)$$
$$\phi'_{24} = \texttt{FALSE} \vee \texttt{FALSE} \vee S(B)$$
$$= S(B)$$

MAP inference for MLNs (1)

Example

What is the most likely world for a given Markov logic network?



Corresponds to weighted CNF formula:

$$\Psi = f_{11} \wedge f_{12} \wedge f_{23} \wedge f_{24} \wedge f_{25} \wedge f_{26} \wedge f_{27} \wedge f_{28}$$

MAP inference for MLNs (2)

Definition

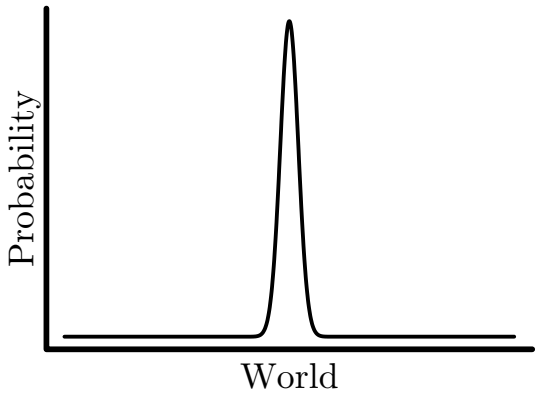
Consider a CNF formula F over variables $\mathcal{X}$, in which each of the clauses $f_1, \ldots, f_m$ is associated with a corresponding weight $w_1, \ldots, w_m$. The *Weighted MAX-SAT problem* is to find an assignment $\mathbf{x}^* \in \mathcal{X}_F$ that maximizes the sum of the weights of satisfied clauses, i.e., $\mathbf{x}^* = \text{argmax}_{\mathbf{x}} \sum_i w_i f_i$.

Consider the following transformation:

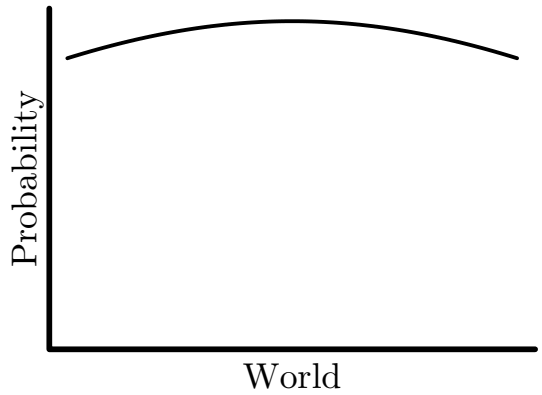
$$\begin{aligned}\operatorname*{argmax}_{\mathbf{x}} \mathbb{P}(\mathbf{x}) &= \operatorname*{argmax}_{\mathbf{x}} \left[\frac{1}{Z} \exp \sum_{(f,w)\in M_{L,C}} w f(\mathbf{x})\right] \\ &= \operatorname*{argmax}_{\mathbf{x}} \sum_{(f,w)\in \underbrace{M_{L,C}}_{F}} \underbrace{w}_{w_i} \underbrace{f(\mathbf{x})}_{f_i} = \mathbf{x}^*\end{aligned}$$

There are many algorithms and solvers for Weighted MAX-SAT, both exact and approximate. Specialized algorithms for MLNs do exist; they try to reduce grounding by computing $M_{L,C}$ only partially.

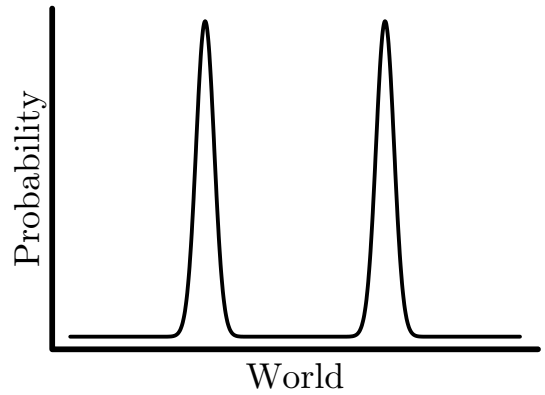
MAP inference for MLNs (3)



MAP world characterizes distribution well



MAP world not distinguished from other words

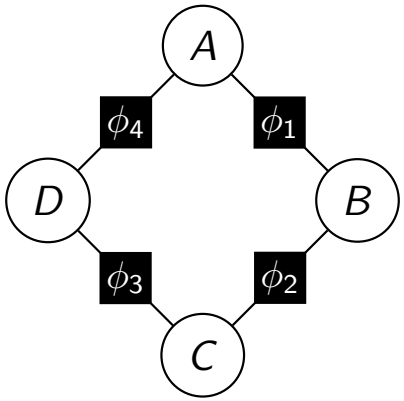


MAP world(s) characterize only a part of the distribution

MAP estimates provide the "most consistent" world, i.e., the world that satisfies most of the rules. This world *may or may not* characterize the entire distribution well.

Variable elimination (idea)

Goal: Eliminate non-query variables from the graph.



ϕ_1

A	B	ϕ
a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

ϕ_2

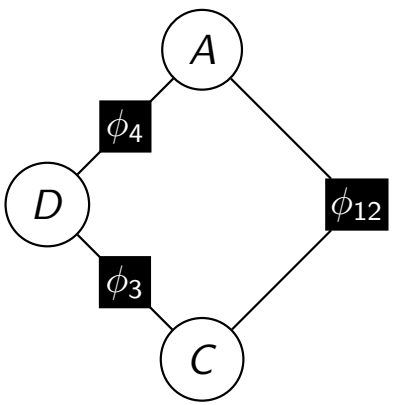
B	C	ϕ
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

ϕ_3

C	D	ϕ
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

ϕ_4

D	A	ϕ
d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100



$\phi_1 \times \phi_2$

A	B	C	ϕ
a^0	b^0	c^0	3000
a^0	b^0	c^1	30
a^0	b^1	c^0	5
a^0	b^1	c^1	500
a^1	b^0	c^0	100
a^1	b^0	c^1	1
a^1	b^1	c^0	10
a^1	b^1	c^1	1000

ϕ_{12}

A	C	ϕ
a^0	c^0	3005
a^0	c^1	530
a^1	c^0	110
a^1	c^1	1001

B has been eliminated ("marginalized out"). The resulting factor graph represents $\mathbb{P}(A, C, D)$.

Variable elimination (why it works)

Recall that

$$\mathbb{P}(A, B, C, D) = \frac{1}{Z}\phi_1(A, B) \times \phi_2(B, C) \times \phi_3(C, D) \times \phi_4(D, A)$$

and thus

$$\begin{aligned}
\mathbb{P}(A, C, D) &= \mathbb{P}\left(A, b^0, C, D\right) + \mathbb{P}\left(A, b^1, C, D\right) \\
&= \frac{1}{Z}[\phi_1(A, b^0) \times \phi_2(b^0, C) \times \phi_3(C, D) \times \phi_4(D, A) \\
&\qquad + \phi_1(A, b^1) \times \phi_2(b^1, C) \times \phi_3(C, D) \times \phi_4(D, A)] \\
&= \frac{1}{Z}\left[\left\{\sum_{b \in \{b^0, b^1\}} \phi_1(A, b) \times \phi_2(b, C)\right\} \times \phi_3(C, D) \times \phi_4(D, A)\right] \\
&= \frac{1}{Z}\left[\phi_{12}(A, C) \times \phi_3(C, D) \times \phi_4(D, A)\right]
\end{aligned}$$

Variable elimination (remarks)

- Also called sum-product variable elimination
- Whenever we eliminate a variable B
  - We remove all factors connected to B
  - We introduce a single factor that is connected to the neighbors of B
  - If B has k neighbors, the new factor has 2^k rows
    $\rightarrow$ Potentially exponential blow-up
- Computational cost
  - Dominated by sizes of intermediate factors
  - Depends strongly on elimination ordering
  - NP-hard to find optimal ordering
  - Lots of useful heuristics exist
  - "Conditioning" can be used to avoid large factors for increased processing time
- Similar observations give rise to other important algorithms, e.g., "message passing" in "clique trees"

Outline

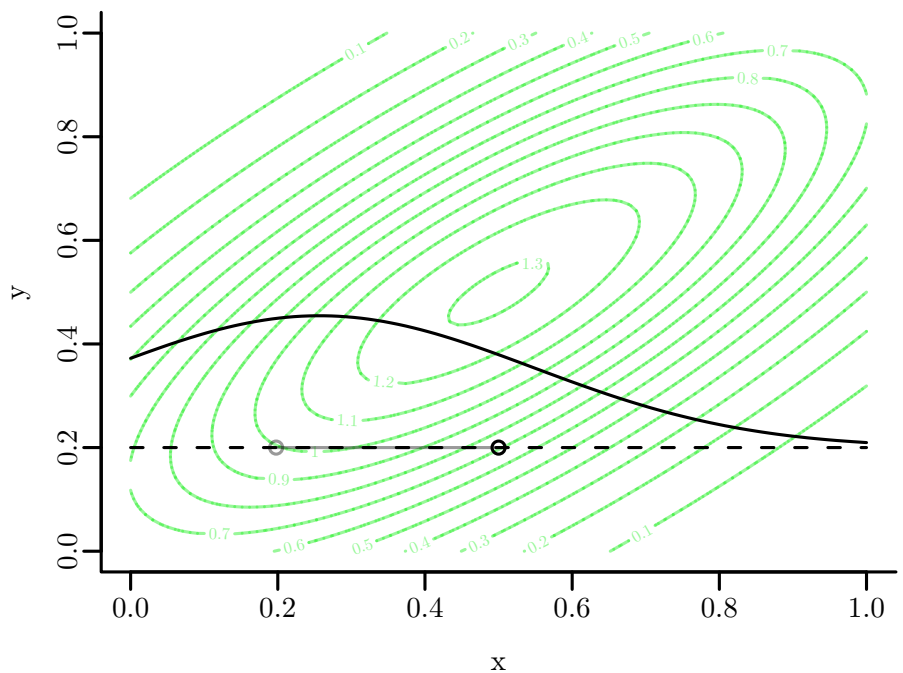
Sampling methods

- Also called *particle-based approximate inference*
- Idea: Obtain samples from the distribution underlying the graphical model
- If samples were independent, we could count how often each variables is true/false and apply the sampling theorem
- Sampling is much more difficult in Markov networks $\rightarrow$ samples are generally dependent
  - Goal is to minimize the dependencies
  - More samples needed than "implied" by the sampling theorem
  - If dependencies vanish between far-apart samples $\rightarrow$ correctness and convergence
- Many techniques
  - Forward sampling (for directed models)
  - Likelihood weighting
  - Importance sampling
  - Gibbs sampling
  - Other Markov Chain Monte Carlo (MCMC) methods
  - Collapsed particles

Gibbs sampling (idea)

Gibbs sampling is a simple algorithm to sample from $\mathbb{P}(X, Y)$. It is used when it is hard to sample from $\mathbb{P}(X, Y)$, but easy to sample from $\mathbb{P}(X \mid Y)$ and $\mathbb{P}(Y \mid X)$.

1. Pick an initial point (x_0, y_0)
2. For $n = 1, 2, \ldots$
   1. Generate $x_n \sim \mathbb{P}(X \mid Y = y_{n-1})$

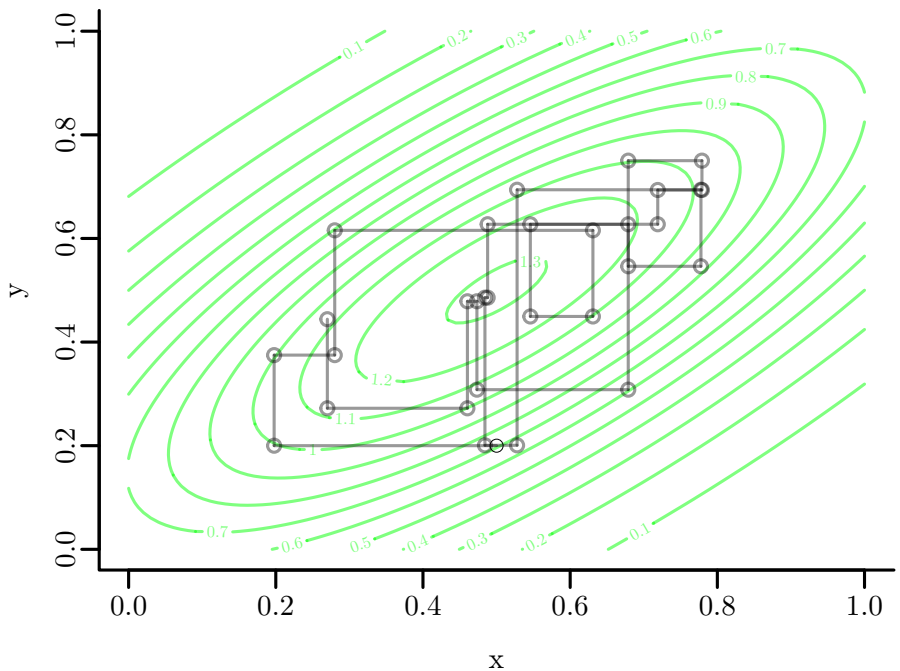


$n = 1$

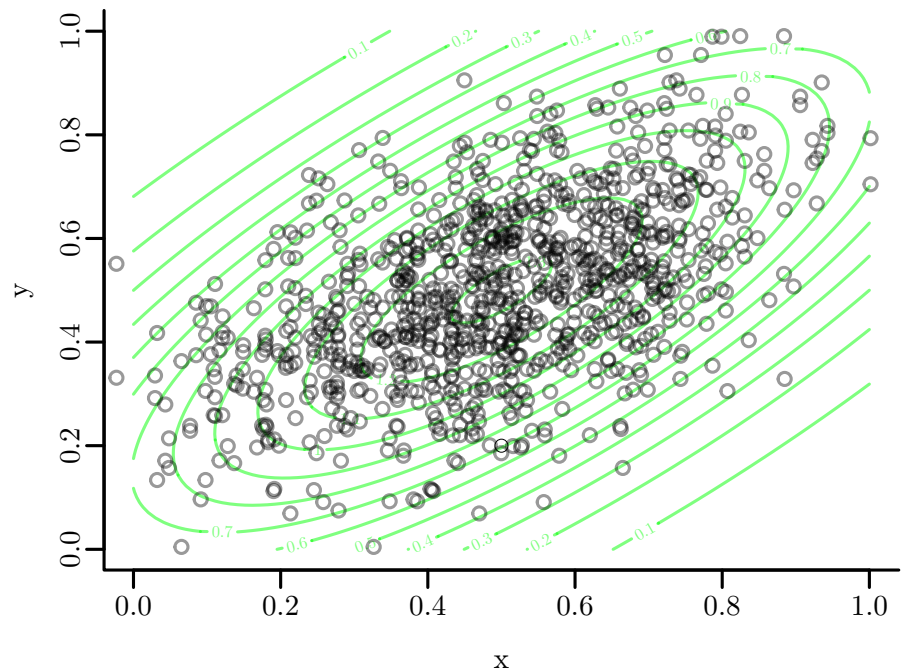
Gibbs sampling (idea)

Gibbs sampling is a simple algorithm to sample from $\mathbb{P}(X, Y)$. It is used when it is hard to sample from $\mathbb{P}(X, Y)$, but easy to sample from $\mathbb{P}(X \mid Y)$ and $\mathbb{P}(Y \mid X)$.

1. Pick an initial point (x_0, y_0)
2. For $n = 1, 2, \ldots$
   1. Generate $x_n \sim \mathbb{P}(X \mid Y = y_{n-1})$
   2. Generate $y_n \sim \mathbb{P}(Y \mid X = x_n)$



$n = 15$



$n = 500$

Gibbs sampling for Markov networks

Recall that

$$\mathbb{P}(A, B, C, D) = \frac{1}{Z}\phi_1(A, B) \times \phi_2(B, C) \times \phi_3(C, D) \times \phi_4(D, A).$$

Sampling from $\mathbb{P}(A, B, C, D)$ is hard but sampling from

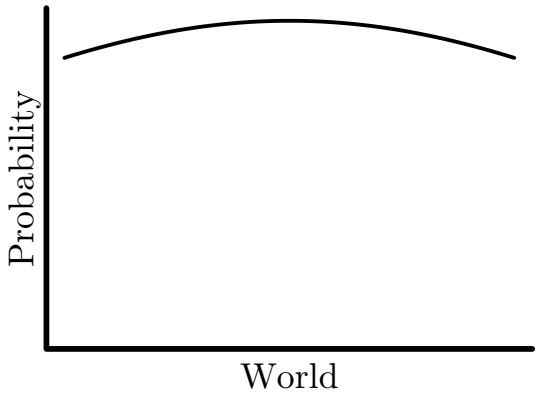
$$\begin{aligned}
\mathbb{P}(A \mid B, C, D) &= \frac{\mathbb{P}(A, B, C, D)}{\mathbb{P}(B, C, D)} \\
&= \frac{\frac{1}{Z}[\phi_1(A, B) \times \phi_2(B, C) \times \phi_3(C, D) \times \phi_4(D, A)]}{\frac{1}{Z}\sum_{a \in \{a^0, a^1\}}[\phi_1(a, B) \times \phi_2(B, C) \times \phi_3(C, D) \times \phi_4(D, a)]} \\
&= \frac{\phi_1(A, B) \times \phi_4(D, A)}{\sum_{a \in \{a^0, a^1\}} \phi_1(a, B) \times \phi_4(D, a)}
\end{aligned}$$

is easy. Only the factors connected to A remain.

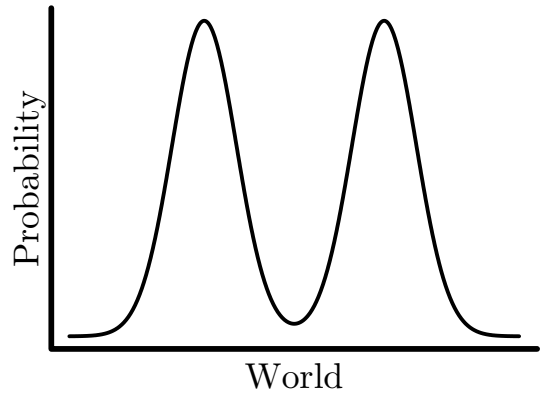
When resampling a variable A, we only have to look at the factors connected to A, and thus only the subset of variables connected to A. These variables are called the *Markov blanket* of A.

Gibbs sampling for Markov networks (remarks)

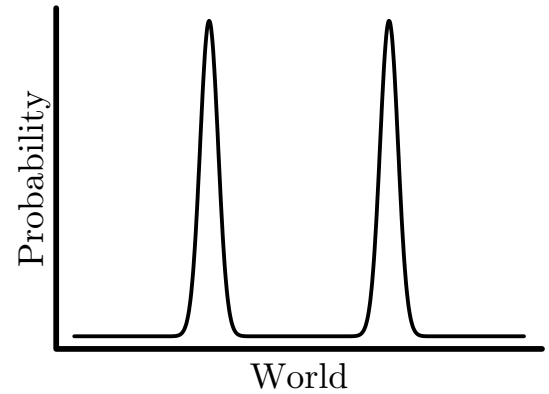
- Variables are picked according to a *schedule*
  $\rightarrow$ sequential, random, ...
- An instance of the more general class of MCMC methods
  - Markov chains describe how the sampling process moves through the set of worlds
  - Irreducible if all worlds can be reached from all other worlds
  - Convergence speed depends on how fast the sampling process moves (*mixing time*)



Gibbs sampling works well (fast mixing)



Gibbs sampling works reasonable (slow mixing)



Gibbs sampling does not work (not irreducible)

- MCMC methods can perform "bigger" steps than Gibbs sampling; they change multiple variables simultaneously

Outline

Lessons learned

- Probabilistic databases and graphical models focus on different aspects of probabilistic reasoning
- Probabilistic graphical models
  - Describe and reason about probability distributions and independencies
  - Exploit locality structure (conditional independence)
  - Main components: representation, inference, learning
- Markov logic
  - Combines first-order logic and probability theory
  - Set of formulas with weights
  - Template for generating undirected graphical models
- Inference
  - #P-hard in general
  - MAP inference on MLNs corresponds to Weighted MAX-SAT
  - Exact methods for probability computation (e.g., variable elimination) may work when graph has no dense regions
  - Approximate methods often based on MCMC sampling
  - Gibbs sampling is the simplest MCMC method; it changes one variable at a time

Suggested reading

- Daphne Koller, Nir Friedman
  Probabilistic Graphical Models: Principles and Techniques
  The MIT Press, 2009

- Matthew Richardson and Pedro Domingos
  Markov Logic Networks
  Machine Learning, 62(1-2), pp. 107–136, 2006

- Michael Mitzenmacher, Eli Upfal
  Probability and Computing: Randomized Algorithms and Probabilistic Analysis
  Cambridge University Press, 2005

- `http://alchemy.cs.washington.edu/`