NSR: Related Areas and Models

Lecture 4
Overview of this Lecture

• Introduction: The need to address uncertainty

• Probabilistic Graphical Models (PGMs):
  – Bayesian Networks
  – Markov Random Fields
  – Markov Logic Networks
  – Markov Chains

• Probabilistic Soft Logic
Intro: The Need to Address Uncertainty

- **Data management** is at the core of many, many applications related to intelligent systems.

- **Uncertainty** is everywhere in data management:
  - Natural / Inherent uncertainty: some domains are uncertain “in nature” (e.g., weather forecasting, markets, etc.).
  - Uncertainty arising from automated processing of data (e.g., data integration, data cleaning, etc.).
  - Uncertainty arising from inconsistency and/or incompleteness (over and underspecification, resp.).
Intro: The Need to Address Uncertainty

Another way of looking at the sources of uncertainty:

• **Laziness:** It’s just too difficult to get everything right in modeling a domain, so we decide to incorporate uncertainty.

• **Theoretical ignorance:** There’s just not enough information to model the domain correctly.

• **Practical ignorance:** Even if theoretically all necessary knowledge is available, specific data points are missing that are required to make decisions.

Probabilistic Uncertainty

• Arguably the most successful approach to addressing uncertainty is via **probabilistic** models.

• There are other kinds of models that are useful in certain applications:
  - Possibilistic / Fuzzy values
  - Dempster-Shafer Theory

• Probabilities are typically **interpreted** in one of two ways:
  - Frequentist, a.k.a. “Objective”
  - Bayesian, a.k.a “Subjective”
Probabilistic *Graphical* Models

- Probabilistic Graphical Models (PGMs) are graph-based structures that are used to represent knowledge about an uncertain domain.

- **Representation:**
  - **Nodes:** Denote random variables, which can be either discrete or continuous.
  - **Edges:**
    - Can be either directed or undirected.
    - Encode probabilistic dependency between two variables; lack of an edge denotes conditional independence.
We will cover four kinds of PGMs:

- Bayesian Networks (BNs)
- Markov Networks / Markov Random Fields (MRFs)
- Markov Logic Networks (MLNs)
- Markov Chains (MCs)

**Note:** For ease of presentation, here we focus on discrete models, but there are continuous versions of each of these as well.
Model 1: Bayesian Networks

A BN is a directed acyclic graph where:

• each node represents a discrete random variable;

• if there is an edge between nodes $X$ and $Y$, we say that $X$ is a parent of $Y$; this represents direct dependence between $X$ and $Y$;

• each node is assigned a conditional probabilistic distribution $P(X_i|\text{Parents}(X_i))$ that quantifies the effect that parent nodes have on variable $X_i$;

• each variable is independent of its non-descendants in the graph, given the state of its parents;

• the absence of an edge between two nodes represents conditional independence between the corresponding variables.
A Classic Example (by J. Pearl)

• Suppose we have an anti-burglary alarm installed at home. It’s fairly reliable in detecting burglars, but it sometimes also responds to small earthquakes.

• We have two neighbors, John and Mary, who have agreed to call us at work if they hear the alarm:
  • John *always calls* when he hears the alarm, but sometimes he *mistakes* the sound of the phone with that of the alarm.
  • Mary, on the other hand, enjoys listening to loud music and sometimes *doesn’t hear* the alarm at all.

• Given evidence regarding who called (and who didn’t), we would like to estimate the probability that there is a burglary:
  
  *We’re at work, John calls saying that he hears the alarm but Mary doesn’t; is there a burglary?*

• **Variables:** Burglary, Earthquake, Alarm, JohnCalls, MaryCalls
Example

Here’s one way to model this as a BN:

The **joint probability distribution** is factorized as follows:

\[ P(E, B, A, J, M) = P(E) \cdot P(B) \cdot P(A|B, E) \cdot P(J|A) \cdot P(M|A) \]

Source: CIS520 UPenn (https://alliance.seas.upenn.edu/~cis520/dynamic/2017/wiki/)
**BNs: Problems**

**Property:** Given any probability distribution, there exists a BN that encodes it (*though unwieldy structures might be necessary*).

The most common problems to solve given a BN are:

- **PE** (Probability of “evidence” / Inference): compute the probability that a subset of the variables have given values (\#P-complete).

- **MAP** (Marginal a posteriori probability): given evidence $e$ and variable $X_i$, compute $Pr(X_i = x_i \mid e)$ (PP-complete in its decision version).

- **MPE** (Most probable explanation): given evidence, find the assignment for the rest of the variables that has the greatest probability (NP-complete in its decision version).

Although all of these problems are intractable in their general case, there are special cases with associated PTIME algorithms (exact and/or approximate).
Model 2: Markov Random Fields (MRFs)

A Markov Random Field (MRF) is an **undirected** graph where:

• each node represents a discrete random variable;

• edges correspond to a notion of direct probabilistic interaction, which is parameterized with **potential functions** (there is one potential function per **maximal clique**);

• **potentials**: non-negative real-valued functions of the values of each variable in each clique (referred to as the **state** of the clique);

• a node is **conditionally independent** of the rest of the nodes in the graph given the values of its immediate neighbors (referred to as the node’s Markov blanket).
Example

Variables:
- Sunny (the day is sunny)
- Hot (it’s hot outside)
- Beach (we go to the beach)
- Walk (we go on a walk)

The \textbf{joint distribution} of \( X = \{X_1, X_2, ..., X_n\} \) is defined as follows:

\[
P(X = x) = \frac{1}{Z} \prod_i \phi_i(x_{\{i\}})
\]

where \( \phi_i \) are potential functions and \( x_{\{i\}} \) is the state of the \( i \)-th maximal clique.

\( Z \) is a \textbf{normalizing constant}, which ensures that the sum of probabilities is equal to 1:

\[
Z = \sum_{x \in X} \prod_i \phi_i(x_{\{i\}})
\]
Example

We can calculate the probability that the day is sunny and hot, and that we go to the beach and not on a walk as follows:

\[
P(s \land h \land b \land \neg w) = \frac{1}{Z} (2 \times 3 \times 1.7) = \frac{10.2}{Z} = \frac{10.2}{82.82} \approx 0.1231
\]

Source: [Oliveira, 2009]
MRFs: Features

- **Issue**: Expressing the value of each state of each clique is *exponential* in the size of the cliques.

- We can obtain a more compact representation using functions called *features*.

- For this, we will use an equivalent formulation called *log-linear*.

- Features are introduced as follows:

\[
P(X = x) = \frac{1}{Z} e^{\sum_i w_i f_i(x)}
\]

where \( i \) varies over the set of cliques. Normalization is analogous:

\[
Z = \sum_{x \in X} e^{\sum_i w_i f_i(x)}
\]
MRFs: Features

• Now, features $f_i(x)$ (also real-valued functions over states) replace potential functions.

• Each $f_i(x)$ also has associated a weight $w_i$.

• In this lecture we consider binary features: $f_i(x) \in \{0,1\}$.

• The most direct translation to the original form is then:

  There is a feature corresponding to each possible state $x_{\{i\}}$ of each clique, with weight $\ln \phi_i(x_{\{i\}})$. 
Example

Coming back to our example, we can define a simple feature for the clique \{Sunny, Walk\} as follows:

\[
F(S,W) = \begin{cases} 
1 & \text{if Sunny=true} \land \text{Walk=false} \\
0 & \text{otherwise}
\end{cases}
\]

Source: [Oliveira, 2009]
Model 3: Markov Logic Networks

An MLN is a finite set of pairs $(F_i, w_i)$, where:

• $F_i$ is a FOL formula

• $w_i$ is a real number (the weight of the formula)

Together with a finite set of constants $C = \{c_1, c_2, \ldots, c_n\}$, it defines an $MRF M_{L,C}$ as follows:

• $M_{L,C}$ contains a binary node for each possible ground instance of an atom in $L$. The node’s value is 1 if the atom is true, and 0 otherwise.

• $M_{L,C}$ contains a feature for each ground instance of a formula $F_i$ in $L$. The value of the feature is 1 if it is true, and 0 otherwise, and its weight is the value $w_i$ associated with $F_i$ in $L$. 
MLNs

From this definition we can derive that:

• **Ground atoms** generate **nodes** in the network.

• There is an **edge** between nodes if and only if the ground atoms **appear together** in at least one ground instance of a ground formula in $L$.

• Formulas generate **cliques** in the network.
Example

- Consider the MLN defined by the following pairs:

  - $(\forall x \ Sm(x) \Rightarrow Ca(x), 1.5) \sim$ Smoking leads to cancer

  - $(\forall x \forall y \ Fr(x, y) \Rightarrow (Sm(x) \Leftrightarrow Sm(y)), 1.1) \sim$ If two people are friends, either they both smoke or they both don’t.

  Assume we have constants: $\{Anna, Bob\}$.

- $M_{L, C}$ can now be used to infer:

  - the probability that Anna and Bob are friends given their smoking habits,

  - the probability that Bob has cancer given his friendship with Anna,

  - if Anna has cancer, etc.
Example

The following graph corresponds to the induced MRF:

Formulas:

\[ \forall x \ Sm(x) \Rightarrow Ca(x), \]
\[ \forall x \ \forall y \ Fr(x,y) \Rightarrow (Sm(x) \Leftrightarrow Sm(y)) \]

The probability distribution represented by the MLN is the following:

\[ P(X = x) = \frac{1}{Z} e^{\Sigma_i w_i n_i(x)} \]

where \( n_i(x) \) is the number of ground instances of \( F_i \) that are satisfied by \( x \), and \( Z \) is the normalizing constant.
A Full Example

• Let’s define an MLN with the following pairs:

\[ \psi_1: (p(X) \Rightarrow q(X), 1.2) \]
\[ \psi_2: (p(X) \Rightarrow r(X), 2) \]
\[ \psi_3: (s(X) \Rightarrow r(X), 3) \]

and the singleton set of constants \( \{c\} \).

• Set of ground atoms \( \{p(c), q(c), r(c), s(c)\} \), graph:
A Full Example

We then have $2^4 = 16$ possible assignments of values to the MRF variables. Probabilities are as follows:

<table>
<thead>
<tr>
<th>$\lambda_i$</th>
<th>$p(c)$</th>
<th>$q(e)$</th>
<th>$r(c)$</th>
<th>$s(c)$</th>
<th>Satisfies</th>
<th>Potential</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>2</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>$\psi_1,\psi_2$</td>
<td>$1.2 + 2 = 3.2$</td>
<td>$e^{3.2}/Z \approx 0.006$</td>
</tr>
<tr>
<td>3</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>4</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>5</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>$\psi_1,\psi_2$</td>
<td>$1.2 + 2 = 3.2$</td>
<td>$e^{3.2}/Z \approx 0.006$</td>
</tr>
<tr>
<td>6</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>$\psi_1,\psi_2$</td>
<td>$1.2 + 2 = 3.2$</td>
<td>$e^{3.2}/Z \approx 0.006$</td>
</tr>
<tr>
<td>7</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>8</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>9</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>10</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>11</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>$\psi_2,\psi_3$</td>
<td>$2 + 3 = 5$</td>
<td>$e^{5}/Z \approx 0.038$</td>
</tr>
<tr>
<td>12</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>$\psi_2,\psi_3$</td>
<td>$2 + 3 = 5$</td>
<td>$e^{5}/Z \approx 0.038$</td>
</tr>
<tr>
<td>13</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>$\psi_1,\psi_3$</td>
<td>$1.2 + 3 = 4.2$</td>
<td>$e^{4.2}/Z \approx 0.017$</td>
</tr>
<tr>
<td>14</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>$\psi_1$</td>
<td>$1.2$</td>
<td>$e^{1.2}/Z \approx 0$</td>
</tr>
<tr>
<td>15</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
<tr>
<td>16</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>$\psi_1,\psi_2,\psi_3$</td>
<td>$1.2 + 2 + 3 = 6.2$</td>
<td>$e^{6.2}/Z \approx 0.127$</td>
</tr>
</tbody>
</table>
A Full Example

• The normalization factor \( Z \) is calculated:

\[
Z = 7e^{6.2} + 3e^{3.2} + 2e^0 + 2e^5 + e^{4.2} + e^{1.2} \approx 3,891.673
\]

• If we wish to calculate the probability of formula \( p(c) \land q(c) \), we sum the probabilities of all worlds that satisfy it, in this case worlds 13, 14, 15, and 16:

\[
\frac{e^{4.2} + e^{1.2} + e^{6.2} + e^{6.2}}{Z} \approx \frac{1,055.5}{3,891.673} \approx 0.271
\]
Remarks

• Hammersley–Clifford Theorem (1971):
  – Intuitively, this result relates (for strictly positive distributions) conditional independence with joint distribution factorization.
  – Much more on this here (outside the scope of this lecture): https://folk.idi.ntnu.no/helgel/thesis/forelesning.pdf

• Probabilistic independence is simple to determine: given by graph separation:
  – In particular, this means that the Markov blanket of a variable is simply comprised by its set of immediate neighbors.
  – Therefore, a variable is independent of the rest of the graph given its neighbors.

• Exact computation is #P-hard

• BNs can be converted into MRFs, which is useful for approximate query answering in certain cases.
Model 4: Markov Chains

A Markov Chain (MC) is a **stochastic process** of the form: \( \{X_n\}_{n \in \mathbb{N} \cup \{0\}} \):

- Set of random variables that represent the **evolution** of a system of random values **through time**.

- It satisfies the **Markov property**:

  given \( n \in \mathbb{N} \cup \{0\} \) and states \( x_0, x_1, ..., x_n, x_{n+1} \), we have:

\[
P(X_{n+1} = x_{n+1} \mid X_n = x_n, ..., X_0 = x_0) = P(X_{n+1} = x_{n+1} \mid X_n = x_n)
\]
Markov Chains

• The Markov property states that the conditional probability distribution of future states only depends on the current state.

• MCs can be represented as sequences of graphs where the edges of graph $n$ are labeled with the probability of going from one state at moment $n$ with other states at moment $n + 1$:

$$P(X_{n+1} = x \mid X_n = x_n).$$
Markov Chains

• The same information can be represented with a transition matrix from time $n$ to $n + 1$:

$$
\begin{pmatrix}
  p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\
  p_{21} & p_{22} & \cdots & p_{2j} & \cdots \\
  \vdots & \vdots & \ddots & \vdots & \ddots \\
  p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots \\
  \vdots & \vdots & \ddots & \vdots & \ddots
\end{pmatrix}
$$

where $p_{ij} = P(X_{n+1} = x_j \mid X_n = x_i)$.

• Given that the probabilities of transitioning from state $i$ to the rest of the states must sum to 1, we have $\sum_j p_{ij} = 1$. 

Example

Suppose we classify jobs into low, middle, and high wage, and suppose further that the most influential factor for the kind of job a person will have next is the job they have now.

We have:

- **State 1**: Person has a low wage job
- **State 2**: Person has a medium wage job
- **State 3**: Person has a high wage job

<table>
<thead>
<tr>
<th>Current job</th>
<th>State</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.65</td>
<td>0.28</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.15</td>
<td>0.67</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.12</td>
<td>0.36</td>
<td>0.52</td>
<td></td>
</tr>
</tbody>
</table>
Example

• This information may be expressed via a transition diagram:

• Or a transition matrix:

\[
\begin{bmatrix}
1 & 2 & 3 \\
1 & 0.65 & 0.28 & 0.07 \\
2 & 0.15 & 0.67 & 0.18 \\
3 & 0.12 & 0.36 & 0.52 \\
\end{bmatrix} = P
\]

Example

Now, we would like to know the probabilities of changing wage class within the next two jobs.

For example, if a person is currently in State 3, what is the probability that they will be in state 2 two jobs from now?

\[
\text{Probability: } 0.0336 + 0.2412 + 0.1872 = 0.462
\]

Source: http://www.math.bas.bg/~jeni/markov123.pdf
Example

Another way of going about this calculation is to take the element \((P^2)_{32}:

\[
\begin{pmatrix}
0.65 & 0.28 & 0.07 \\
0.15 & 0.67 & 0.18 \\
0.12 & 0.36 & 0.52
\end{pmatrix}
\begin{pmatrix}
0.65 & 0.28 & 0.07 \\
0.15 & 0.67 & 0.18 \\
0.12 & 0.36 & 0.52
\end{pmatrix}
= \\
\begin{pmatrix}
* & * & * \\
* & * & * \\
* & * & *
\end{pmatrix}
= \\
\begin{pmatrix}
* \\
* \\
* 
\end{pmatrix}
= \\
\begin{pmatrix}
* \\
* \\
0.12 \times 0.28 + 0.36 \times 0.67 + 0.52 \times 0.36 = 0.462 \\
*
\end{pmatrix}
Markov Chains

• We say that an MC is regular if some power of its transition function contains all positive elements.

• If an MC with matrix $P$ is regular, then there exists a unique vector $V$ such that for any vector of probabilities $\nu$ and large values of $n$ we have:

$$\nu \cdot P^n \approx V$$

• Vector $V$ is called the stationary distribution of the MC.

• To find $V$ we can solve the equation $V \cdot P = V$, using the fact that entries must sum to 1.

• Powers $P^n$ get iteratively closer to the matrix with rows that correspond to the stationary distribution $V$. 
Example: Equation Method

We want to find $V = (v_1, v_2, v_3)$ such that $V \cdot P = V$; that is,

$$
(v_1 \quad v_2 \quad v_3) \cdot 
\begin{pmatrix}
0.65 & 0.28 & 0.07 \\
0.15 & 0.67 & 0.18 \\
0.12 & 0.36 & 0.52
\end{pmatrix} = (v_1 \quad v_2 \quad v_3)
$$

Solving the system of equations:

\[
\begin{align*}
-0.35v_1 + 0.15v_2 + 0.12v_3 &= 0 \\
0.28v_1 - 0.33v_2 + 0.36v_3 &= 0 \\
0.07v_1 + 0.18v_2 - 0.48v_3 &= 0 \\
v_1 + v_2 + v_3 &= 1
\end{align*}
\]

we get $V = \left(\frac{104}{363}, \frac{532}{1089}, \frac{245}{1089}\right) \approx (0.2865, 0.4885, 0.2250)$
Example: Matrix Method

If we calculate several powers of the matrix, we can observe that they approach the matrix whose rows have the values of the stationary distribution:

\[ P^4 = \begin{pmatrix} 0.34 & 0.47 & 0.20 \\ 0.27 & 0.50 & 0.23 \\ 0.26 & 0.29 & 0.25 \end{pmatrix} \]

\[ P^{10} = \begin{pmatrix} 0.29 & 0.49 & 0.22 \\ 0.29 & 0.49 & 0.23 \\ 0.29 & 0.49 & 0.23 \end{pmatrix} \]

\[ P^{16} = \begin{pmatrix} 0.29 & 0.49 & 0.22 \\ 0.29 & 0.49 & 0.22 \\ 0.29 & 0.49 & 0.22 \end{pmatrix} \]
Visualization Tool

Markov Chains explained visually:

https://setosa.io/ev/markov-chains/

With two states (A and B) in our state space, there are 4 possible transitions (not 2, because a state can transition back into itself). If we're at 'A' we could transition to 'B' or stay at 'A'. If we're at 'B' we could transition to 'A' or stay at 'B'. In this two state diagram, the probability of transitioning from any state to any other state is 0.5.

Of course, real modelers don’t always draw out Markov chain diagrams. Instead they use a “transition matrix” to tally the transition probabilities. Every state in the state space is included once as a row and again as a column, and each cell in the matrix tells you the probability of transitioning from its row's state to its column's state. So, in the matrix, the cells do the same job that the arrows do in the diagram.
A Key Application of MCs

• Markov chains are the basis of many algorithms based on **sampling** from a probability distribution.

• In particular, they are at the core of a family of methods called **Markov Chain Monte Carlo**, named after the Monte-Carlo Casino in Monaco.
Markov Chain Monte Carlo (MCMC)

- We wish to approximate a distribution $P(X|x)$
- The method is based on the idea of simulating an MC 
  $\{X_i\}_{i \in \mathbb{N} \cup \{0\}}$ with stationary distribution $P(X|x)$:
  
  1) We start at a random state $X_0$.
  2) Generate the next state iteratively by taking samples of the value of one of the variables $X_i$ conditioned on the current values of the variables in $X_i$’s Markov blanket.
  3) The stationary distribution can be approximated by executing Step 2 a sufficient number of times (this number is typically referred to as the MC’s mixing time).

- Note that each sample of the value of $X_i$ depends only on its predecessor $X_{i-1}$.
- The probability of a query can also be calculated via this same kind of random walk.
Example

**Query:** \( P(\text{Beach} = \text{true} \mid \text{Sunny} = \text{true}, \text{Hot} = \text{true}) \)

- Let’s take as first state: \{s, h, b, \neg w\}
- For the next state, we sample from \textit{Walk} given its \textit{Markov blanket}, i.e., \( P(\text{Walk} \mid \text{Sunny} = \text{true}) \)
- If the sample is \( \text{Walk} = \text{true} \), the next state is: \{s, h, b, w\}
- Now, suppose that after 10 steps we visited 9 in which \( \text{Beach} = \text{true} \) and 1 in which \( \text{Beach} = \text{false} \), we have that:

\[
\begin{align*}
P(\text{B} = \text{true} \mid S = \text{true}, H = \text{true}) &= 0.9 \\
P(\text{B} = \text{false} \mid S = \text{true}, H = \text{true}) &= 0.1
\end{align*}
\]

\textbf{Exact value} is obtained by calculating:

\[
\frac{8.67+10.2}{8.67+10.2+0.544+0.64} = 0.94
\]

(using values of potentials from Slide 14)

Source: [Oliveira, 2009]
Probabilistic Soft Logic (PSL)

• In the same spirit as MLNs, PSL is a declarative language to specify PGMs.

• Main features:
  – Logical atoms with soft truth values in \([0,1]\) – this means that continuous models are required.
  – Dependencies encoded via weighted first order rules
  – Support for similarity functions and aggregation
  – Linear (in)equality constraints
  – Efficient MPE inference via continuous convex optimization

This section is based on: Kimmig, Bach, Broecheler, Huang, Getoor (2012): “A Short Introduction to Probabilistic Soft Logic”. NIPS Workshop on Probabilistic Programming.
PSL Application: Voter Opinion Modeling

Given a social network labeled with different relationships between nodes and how some of them voted, can we infer anything about how the others voted?

Source: [Kimmig et al., 2012]
PSL Programs

Ground atoms correspond to random variables:

- friend(carla, emma),
- friend(bob, dan),
- spouse(carla, dan), ...

Soft truth value assignments:

- friend(carla, emma) = 0.9
- friend(bob, dan) = 0.4

Weighted Rules:

- Local rule: 0.3: lives(A, S) \land majority(S, P) \rightarrow prefers(A, P)
- Propagation rule: 0.8: spouse(B, A) \land prefers(B, P) \rightarrow prefers(A, P)
- Similarity rule: similarAge(B, A) \land prefers(B, P) \rightarrow prefers(A, P)

Source: [Kimmig et al., 2012]
Partial functions:

- \( \text{prefers}(A, \text{dem}) + \text{prefers}(A, \text{rep}) \leq 1.0 \)

Sets:

- \( 0.4: \text{prefers}(A, P) \rightarrow \text{prefersAvg} (\{A.\text{friend}\}, P) \)
  - \( A.\text{friend} \): All \( X \) s.t. \( \text{friend}(A, X) \)
  - Truth value of \( \text{prefersAvg} \): average truth value of all atoms of the form \( \text{prefers}(X, P) \)

Source: [Kimmig et al., 2012]
PSL: Probabilistic Model

Ground rule’s distance from satisfaction given I

\[ f(I) = \frac{1}{Z} \exp \left( - \sum_{r \in P} \sum_{g \in G(r)} w_r (d_g(I))^k \right) \]

\( \in \{1, 2\} \)

Interpretation

Rule’s weight

Normalization constant

\[ Z = \int_{J \in \mathcal{I}} \exp \left( - \sum_{r \in P} \sum_{g \in G(r)} w_r (d_g(J))^k \right) \]

Source: [Kimmig et al., 2012]
PSL: Probabilistic Model

Distance to satisfaction:

\[ d_r(I) = \max\{0, I(\text{body}) - I(\text{head})\} \]

• Intuitively, “if body then head” is satisfied iff the truth value of the body is less than or equal to the truth value of the head.

• This is a generalization of classical logical implication.

In order to compute values of conjunctions and disjunctions, PSL uses the Lukasiewicz infinite value logic operators:

\[
\begin{align*}
I(v_1 \land v_2) &= \max\{0, I(v_1) + I(v_2) - 1\} \\
I(v_1 \lor v_2) &= \min\{1, I(v_1) + I(v_2)\} \\
I(\neg v) &= 1 - I(v)
\end{align*}
\]
PSL: Probabilistic Model

PSL programs ground out to special kinds of MRFs called **Hinge-loss MRFs**.

- **Nodes** are continuous variables in $[0,1]$.
- **Potentials** are hinge-loss functions.
- **Log-concave**: This means that a best interpretation can be found tractably.
- Details are out of scope; interested students are referred to:


Source: https://math.stackexchange.com/questions/782586/how-do-you-minimize-hinge-loss