Intelligent Agents: Web-mining Agents

Probabilistic Graphical Models

Continuous Space

Tanya Braun
Probabilistic Graphical Models (PGMs)

1. Recap: **Propositional modelling**
   - Factor model, Bayesian network, Markov network
   - Semantics, inference tasks + algorithms + complexity

2. **Probabilistic relational models** (PRMs)
   - Parameterised models, Markov logic networks
   - Semantics, inference tasks

3. **Lifted inference**
   - LVE, LJT, FOKC
   - Theoretical analysis

4. **Lifted learning**
   - Recap: propositional learning
   - From ground to lifted models
   - Direct lifted learning

5. **Approximate Inference: Sampling**
   - Importance sampling
   - MCMC methods

6. **Sequential models & inference**
   - Dynamic PRMs
   - Semantics, inference tasks + algorithms + complexity, learning

7. **Decision making**
   - (Dynamic) Decision PRMs
   - Semantics, inference tasks + algorithms, learning

8. **Continuous Space**
   - Gaussian distributions and Bayesian networks
   - Probabilistic soft logic
Models with Continuous Variables

• Discretisation of continuous variables
  • Discrete model again
  • Own set of problems
    • Hard to find good discretisation
    • High granularity might be necessary
      → large ranges → large factors
    • Lose characteristics of variable
      • Not each value necessarily associated with a probability
      • Nearby values have similar probabilities → hard to capture in a discrete distribution (no notion of closeness between range values)

• Therefore, use models with continuous variables
Outline: 8. Continuous Space

A. **Basics**
   - Continuous variables, probability density function, cumulative probability distribution
   - Joint distribution, marginal density, conditional density

B. **Gaussian models**
   - (Multivariate) Gaussian distribution
   - (Parameterised) Gaussian Bayesian networks

C. **Probabilistic Soft Logic (PSL)**
   - Modelling, semantics, inference task
Probability Density Function

• Continuous randvar $R$
  • Range $\mathcal{R}(R) = [0,1]$

• Function $p : \mathbb{R} \rightarrow \mathbb{R}$ is a probability density function (PDF) for $R$ if it is a non-negative, integrable function s.t.
  \[
  \int_{\mathcal{R}(R)} p(r)dr = 1
  \]

• For any $a$ (and $b$) in event space
  \[
  P(R \leq a) = \int_{-\infty}^{a} p(r)dr \quad \quad P(a \leq R \leq b) = \int_{a}^{b} p(r)dr
  \]
  • Function $P$ is a cumulative distribution for $R$
  • Intuitively, value of $p(r)$ at point $r$ is the incremental amount that $r$ adds to the cumulative distribution during integration
• Continuous randvar $R$ has a uniform distribution over $[a, b]$, denoted $R \sim \text{Unif}[a, b]$, if it has the PDF

$$p(r) = \begin{cases} \frac{1}{b-a} & b \geq r \geq a \\ 0 & \text{otherwise} \end{cases}$$

• Density can be larger than 1 if $b - a < 1$

• Can be legal if the total area under the pdf is 1
Joint/Multivariate Distribution

• Let $P$ be a joint distribution over continuous randvars $R_1, \ldots, R_n$

• Function $p(r_1, \ldots, r_n)$ is a joint density function of $R_1, \ldots, R_n$ if
  • $p(r_1, \ldots, r_n) \geq 0$ for all values $r_1, \ldots, r_n$ of $R_1, \ldots, R_n$
  • $p$ is an integrable function
  • For any choice $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$,

$$P(a_1 \leq R_1 \leq b_1, \ldots, a_n \leq R_n \leq b_n) = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} p(r_1, \ldots, r_n) dr_1 \cdots r_n$$
Marginal Density

- Given a joint density, integrate out the non-query randvars
  - E.g., given \( p(r, s) \) a joint density for randvars \( R, S \), then

  \[
p(r) = \int_{-\infty}^{+\infty} p(r, s) \, ds
  \]

- Shorthand notations
  - \( p_R = p(r) \) marginal density
  - \( p_{R,S} = p(r, s) \) joint density
Conditional Density Function

• Discrete case: \( P(S|R = r) = \frac{P(S,R=r)}{P(R=r)} \)
  • Problem in continuous case: \( P(R = r) = 0 \)
    \( \rightarrow P(S|R = r) \) undefined

• To avoid problem, condition on event
  \( r - \epsilon \leq R \leq r + \epsilon \) and consider limit when \( \epsilon \to 0 \)

\[
P(S|r) = \lim_{\epsilon \to 0} P(S|r - \epsilon \leq R \leq r + \epsilon)
\]

• If a continuous joint density \( p(r,s) \) exists, derive form of this expression:

\[
p(s|r) = \frac{p(r,s)}{p(r)}
\]
  • If \( p(r) = 0 \), conditional density undefined
  • Chain rule and Bayes’ rule hold as well:

\[
p(r,s) = p(r)p(s|r) \quad p(s|r) = \frac{p(s)p(r|s)}{p(r)}
\]
Outline: 8. Continuous Space

A. Basics
   • Continuous variables, probability density function, cumulative probability distribution
   • Joint distribution, marginal density, conditional density

B. Gaussian models
   • (Multivariate) Gaussian distribution
   • (Parameterised) Gaussian Bayesian networks

C. Probabilistic Soft Logic (PSL)
   • Modelling, semantics, inference task
Models with Continuous Variables

• Problem: Space of possible parameterisation essentially unbounded

• Special case: (Multivariate) Gaussian distributions
  • Two parameters per variable: mean, variance
  • Strong assumptions, e.g.,
    • Exponential decay away from its mean
    • Linearity of interactions between randvars
      → Assumptions often invalid but still work as a good approximation for many real-world distributions
  • Many generalisations exist which use Gaussians as a foundation
    • Non-linear interactions
    • Mixture of Gaussians
PDFs: Gaussian/Normal Distribution

- Continuous random variable $R$ has a Gaussian distribution with mean $\mu$ and variance $\sigma^2$, denoted $R \sim \mathcal{N}(\mu, \sigma^2)$, if it has the PDF

$$p(r) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(r-\mu)^2}{2\sigma^2}}$$

- Expected value and variance of $R$ given by $\mu$ and $\sigma^2$
- Standard deviation: $\sigma$

Standard Gaussian $R \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$:

$$p(r) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(r)^2}{2}}$$
Multivariate Gaussian

- Univariate Gaussian: two parameters
  - Mean $\mu$ and variance $\sigma^2$
- Multivariate Gaussian distribution over continuous random variables $R_1, \ldots, R_n$ characterised by
  - $n$-dimensional mean vector $\mu$
  - Symmetric $n \times n$ covariance matrix $\Sigma$
  - I.e., $\mathcal{N}(\mu; \Sigma)$
- Density function defined as

$$p(r) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp \left[ -\frac{1}{2} (r - \mu)^T \Sigma^{-1} (r - \mu) \right]$$

- $r = (r_1, \ldots, r_n)^T$
- $|\Sigma|$ determinant of $\Sigma$
- To induce a well-defined density, $\Sigma$ must be positive-definite
  - For any $r \in \mathbb{R}^n$ s.t. $r \neq 0 : r^T \Sigma r > 0$
  - Guaranteed to be non-singular $\rightarrow$ non-zero determinant.

Standard multivariate Gaussian $R_1, \ldots, R_n$ with
- $\mu = 0$ (all-zero vector)
- $\Sigma = I$ (identity matrix)
Example

• Joint Standard Gaussian distribution over two randvars $R_1, R_2$, i.e.,
  $\mu = (0 \ 0)^T, \Sigma = I_2$

• Joint Gaussian distribution over three randvars $R_1, R_2, R_3$
  • Mean vector, covariance matrix:
    $\mu = \begin{pmatrix} 1 \\ -3 \\ 4 \end{pmatrix}$
    $\Sigma = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & -5 \\ -2 & -5 & 8 \end{pmatrix}$
  • Covariances $Cov[R_1; R_3]$ and $Cov[R_2; R_3]$ negative, i.e., $R_3$ negatively correlated with $R_1$ (and $R_2$)
    • When $R_1$ ($R_2$) goes up, $R_3$ goes down
Marginalisation

• Trivial with covariance matrix:
  • Compute pairwise covariances, i.e., generating the distribution in its covariance form
  • Given covariance form $\Sigma$: Read off from $\mu, \Sigma$

• Assume a joint Gaussian distribution over $\{R, T\}$ where $R \in \mathbb{R}^n$ and $T \in \mathbb{R}^m$
  • One can decompose mean and covariance:
    $$p(r, t) = \mathcal{N}\left((\mu_R, \mu_T); \begin{bmatrix} \Sigma_{RR} & \Sigma_{RT} \\ \Sigma_{TR} & \Sigma_{TT} \end{bmatrix}\right)$$
  • where
    • $\mu_R \in \mathbb{R}^n$, $\mu_T \in \mathbb{R}^m$,
    • $\Sigma_{RR}$ an $n \times n$ matrix, $\Sigma_{RT}$ an $n \times m$ matrix,
    $\Sigma_{TR} = \Sigma_{RT}^T$ an $m \times n$ matrix, $\Sigma_{TT}$ a $m \times m$ matrix

• Then, marginal distribution over $T$ given by Gaussian distribution of $\mathcal{N}(\mu_T; \Sigma_{TT})$
Example

• Given joint Gaussian distribution over three randvars $R_1, R_2, R_3$
  
  • Mean vector, covariance matrix:

$$
\begin{align*}
\mu &= \begin{pmatrix} 1 \\ -3 \\ 4 \end{pmatrix} \\
\Sigma &= \begin{pmatrix} 4 & 2 & -2 \\ 2 & 5 & -5 \\ -2 & -5 & 8 \end{pmatrix}
\end{align*}
$$

• $p(R_1, R_2)$ given by Gaussian distribution with

$$
\begin{align*}
\mu &= \begin{pmatrix} 1 \\ -3 \end{pmatrix} \\
\Sigma &= \begin{pmatrix} 4 & 2 \\ 2 & 5 \end{pmatrix}
\end{align*}
$$
Dual: Information/Precision Form

- Rewrite \( \exp \left[ -\frac{1}{2} (r - \mu)^T \Sigma^{-1} (r - \mu) \right] \) by setting \( \Gamma = \Sigma^{-1} \) and multiplying out:
  \[
  -\frac{1}{2} (r - \mu)^T \Gamma (r - \mu) = -\frac{1}{2} [r^T \Gamma r - 2r^T \Gamma \mu + \mu^T \Gamma \mu]
  \]

- \( \mu^T \Gamma \mu \) is constant over the different \( r \), therefore,
  \[
p(r) \propto \exp \left( -\frac{1}{2} [r^T \Gamma r - 2r^T \Gamma \mu] \right)
  = \exp \left[ -\frac{1}{2} r^T \Gamma r + \Gamma \mu^T \right]
  = \exp \left[ -\frac{1}{2} r^T \Gamma r + (\Gamma \mu)^T r \right]
  \]

- \( \Gamma \mu \) called potential vector

\[
\begin{align*}
r^T \Gamma \mu &= (r^T \Gamma \mu)^T \quad \triangleright A^T = A \\
&= (\Gamma \mu^T r^T)^T \quad \triangleright (AB)^T = B^T A^T \\
&= (\Gamma \mu)^T r \quad \triangleright A^T = A \\
&= (\Gamma \mu)^T r \quad \triangleright k^T = k, k \text{ a scalar}
\end{align*}
\]
Dual: Information/Precision Form

• For a decomposition \( \{R, T\} \) where \( R \in \mathbb{R}^n \) and \( T \in \mathbb{R}^m \):

\[
\Gamma = \Sigma^{-1} = \begin{bmatrix} \Sigma_{RR} & \Sigma_{RT} \\ \Sigma_{TR} & \Sigma_{TT} \end{bmatrix}^{-1} = \begin{bmatrix} \Gamma_{RR} & \Gamma_{RT} \\ \Gamma_{TR} & \Gamma_{TT} \end{bmatrix}
\]

• Getting to \( \Sigma \)
  • \( \Sigma_{RR} = \left( \Gamma_{RR} - \Gamma_{RT} \Gamma_{TT}^{-1} \Gamma_{TR} \right)^{-1} \)
  • \( \Sigma_{TT} = \left( \Gamma_{TT} - \Gamma_{TR} \Gamma_{RR}^{-1} \Gamma_{RT} \right)^{-1} \)
  • \( \Sigma_{RT} = -\Gamma_{RR}^{-1} \Gamma_{RT} \left( \Gamma_{TT} - \Gamma_{TR} \Gamma_{RR}^{-1} \Gamma_{RT} \right)^{-1} = \Sigma_{TR}^T \)
  • \( \Sigma_{TR} = -\Gamma_{TT}^{-1} \Gamma_{TR} \left( \Gamma_{RR} - \Gamma_{RT} \Gamma_{TT}^{-1} \Gamma_{TR} \right)^{-1} = \Sigma_{RT}^T \)

• Getting to \( \Gamma \)
  • \( \Gamma_{RR} = \left( \Sigma_{RR} - \Sigma_{RT} \Sigma_{TT}^{-1} \Sigma_{TR} \right)^{-1} \)
  • \( \Gamma_{TT} = \left( \Sigma_{TT} - \Sigma_{TR} \Sigma_{RR}^{-1} \Sigma_{RT} \right)^{-1} \)
  • \( \Gamma_{RT} = -\Sigma_{RR}^{-1} \Sigma_{RT} \left( \Sigma_{TT} - \Sigma_{TR} \Sigma_{RR}^{-1} \Sigma_{RT} \right)^{-1} = \Gamma_{TR}^T \)
  • \( \Gamma_{TR} = -\Sigma_{TT}^{-1} \Sigma_{TR} \left( \Sigma_{RR} - \Sigma_{RT} \Sigma_{TT}^{-1} \Sigma_{TR} \right)^{-1} = \Gamma_{RT}^T \)
Conditioning

- Conditioning a Gaussian on observations $E = e$ easy to perform in the information form by setting $E$ to $e$ in one of the following

\[ p(r) \propto \exp \left[ -\frac{1}{2} (r - \mu)^T \Gamma (r - \mu) \right] \]

\[ \propto \exp \left[ -\frac{1}{2} r^T \Gamma r + (\Gamma \mu)^T r \right] \]

- Assuming a decomposition into $R$ and $E$, i.e.,

\[ p(r, e) = \mathcal{N} \left( \begin{pmatrix} \mu_R \\ \mu_E \end{pmatrix} ; \begin{bmatrix} \Sigma_{RR} & \Sigma_{RE} \\ \Sigma_{ER} & \Sigma_{EE} \end{bmatrix} \right) \]

\[ \propto \exp \left[ -\frac{1}{2} \begin{pmatrix} r \\ e - \mu_E \end{pmatrix}^T \begin{bmatrix} \Gamma_{RR} & \Gamma_{RT} \\ \Gamma_{TR} & \Gamma_{TT} \end{bmatrix} \begin{pmatrix} r \\ e - \mu_E \end{pmatrix} \right] \]
In the exponential function:

\[
-\frac{1}{2} \left( (r - \mu'_R) - (\mu'_E) \right)^T \begin{bmatrix} \Gamma_{RR} & \Gamma_{RE} \\ \Gamma_{ER} & \Gamma_{EE} \end{bmatrix} \left( (r - \mu'_R) - (\mu'_E) \right)
\]

\[
= -\frac{1}{2} (r - \mu'_R)^T \Gamma_{RR} (r - \mu'_R) - \frac{1}{2} 2(r - \mu'_R)^T \Gamma_{RE} (e - \mu'_E) - \frac{1}{2} (e - \mu'_E)^T \Gamma_{EE} (e - \mu'_E)
\]

\[
\propto -\frac{1}{2} (r - \mu'_R)^T \Gamma_{RR} (r - \mu'_R) - (r - \mu'_R)^T \Gamma_{RE} (e - \mu'_E)
\]

Does not depend on \( r \)

Use \(-A\) to get expression into the form \((r - \mu)^T \Gamma (r - \mu)\) by factoring out \( \Gamma_{RR} \)

\[A = \frac{1}{2} (e - \mu'_E) \Gamma_{ER} \Gamma_{RR}^{-1} \Gamma_{RR} \Gamma_{RE}^{-1} \Gamma_{RE} (e - \mu'_E)\]

\[
\exp \left[ -\frac{1}{2} \left( (r - \mu'_R + \Gamma_{RR}^{-1} \Gamma_{ER} (e - \mu'_E))^T \Gamma_{RR} (r - \mu'_R + \Gamma_{RR}^{-1} \Gamma_{RE} (e - \mu'_E)) \right) \right]
\]

\[
\propto \exp \left[ -\frac{1}{2} \left( (r - \mu'_R + \Gamma_{RR}^{-1} \Gamma_{ER} (e - \mu'_E))^T \Gamma_{RR} (r - \mu'_R + \Gamma_{RR}^{-1} \Gamma_{RE} (e - \mu'_E)) \right) \right]
\]

\[\mu^* = \mu'_R - \Gamma_{RR}^{-1} \Gamma_{ER} (e - \mu'_E)\]

\[\Sigma^* = \Gamma_{RR}\]
Conditioning

• Conditioning a Gaussian on observations $E = e$ with remaining randvars $R$

• Result:

$$R|E = e \sim \mathcal{N}(\mu^*, \Sigma^*)$$

- Information form:
  $$\begin{align*}
  \mu^* &= \mu_R - \Gamma_R^{-1}\Gamma_{ER}(e - \mu_E) \\
  \Sigma^* &= \Gamma_{RR}
  \end{align*}$$

- Covariance form:
  $$\begin{align*}
  \mu^* &= \mu_R + \Sigma_{RE}\Sigma_{EE}^{-1}(e - \mu_E) \\
  \Sigma^* &= \Sigma_{RR} - \Sigma_{RE}\Sigma_{EE}^{-1}\Sigma_{ER}
  \end{align*}$$

• Mean moved from $\mu_R$ according to correlation and variance on observations $\Sigma_{RE}\Sigma_{EE}^{-1}(e - \mu_E)$

• Covariance does not depend on observations $e$
Query Answering: Summary

• For marginalisation, read off parameters in covariance form
  • Marginal query for \( \mathbf{T} : \mathcal{N}(\mathbf{\mu}_T; \Sigma_{TT}) \)

• For conditioning, one needs to invert the covariance matrix to obtain the information form
  • Conditioning on \( \mathbf{E} = \mathbf{e} : \mathbf{R}|\mathbf{E} = \mathbf{e} \sim \mathcal{N}(\mathbf{\mu}^*, \Sigma^*) \)
    • In covariance form
      • \( \mathbf{\mu}^* = \mathbf{\mu}_R + \Sigma_{RE}\Sigma_{EE}^{-1}(\mathbf{e} - \mathbf{\mu}_E) \)
      • \( \Sigma^* = \Sigma_{RR} - \Sigma_{RE}\Sigma_{EE}^{-1}\Sigma_{ER} \)

• Matrix inversion can be very costly!
Linear Gaussian Model

• Let $S$ be a continuous randvar with continuous parents $R_1, \ldots, R_k$

• $S$ has a **linear Gaussian model** if there are parameters $\beta_0, \ldots, \beta_k$ and $\sigma^2$ such that

$$p(S|r_1, \ldots, r_k) = \mathcal{N}(\beta_0 + \beta_1 r_1 + \cdots + \beta_k r_k; \sigma^2)$$

$$= \mathcal{N}(\beta_0 + \beta^T r; \sigma^2)$$

• $p(S|r_1, \ldots, r_k)$ a **conditional probability distribution (CPD)**

• Interpretations
  
  • $\beta_0$ is an initial mean $\mu_0$ that is moved according to the influences by the parents
  
  • $S$ is a linear function of $R_1, \ldots, R_k$ with the addition of Gaussian noise: $S = \beta_0 + \beta_1 r_1 + \cdots + \beta_k r_k + \epsilon$
    
    • $\epsilon$ a Gaussian randvar with mean 0 and variance $\sigma^2$, representing the noise in the system
  
• Does not allow $\sigma^2$ to depend on parent values
  
• But can be a useful approximation
Independencies in Gaussians

• Let randvars $R_1, \ldots, R_n$ have a joint distribution $\mathcal{N}(\mu; \Sigma)$

• Then, $R_i, R_j$ independent iff $\Sigma_{ij} = 0$
  • Joint distribution needs to be Gaussian for this equivalence to hold
    • If the distribution is not Gaussian, $\Sigma_{ij} = 0$ might be the case and there still might be a dependence between $R_i, R_j$

• Conditional independence can be read off in the inverse of the covariance matrix, $\Sigma^{-1}$
  • Given a Gaussian distribution $p(r_1, \ldots, r_n) = \mathcal{N}(\mu; \Sigma)$
  • Then, $\Sigma_{ij}^{-1} = 0$ iff $p \models \left( R_i \perp R_j | \{R_1, \ldots, R_n\} \setminus \{R_i, R_j\} \right)$
Example

• Joint Standard Gaussian distribution over two randvars $R_1, R_2$, i.e.,
  - $\mu = (0 \ 0)^T, \Sigma = I_2$
  - $R_1, R_2$ independent as $\Sigma_{ij} = \Sigma_{ji} = 0$

• Gaussian for $R_1, R_2, R_3$ from before
  - Covariance and inverse covariance matrix:
    \[
    \Sigma = \begin{pmatrix}
    4 & 2 & -2 \\
    2 & 5 & -5 \\
    -2 & -5 & 8
    \end{pmatrix}
    \Sigma^{-1} = \begin{pmatrix}
    0.3125 & -0.125 & 0 \\
    -0.125 & 0.5833 & 0.3333 \\
    0 & 0.3333 & 0.3333
    \end{pmatrix}
    \]
  - $R_1, R_3$ conditionally independent given $R_2$
  - $\Sigma_{13}^{-1} = 0$ iff
  \[
  p = (R_1 \perp R_3 | \{R_1, R_2, R_3\} \setminus \{R_1, R_3\}) = (R_1 \perp R_3 | R_2)
  \]
Gaussian Bayesian Network (GBN)

- Factorisation of a joint distribution into factors also possible with linear Gaussians as local CPDs
- A BN is a directed acyclic graph $G$ whose nodes are discrete randvars $\{R_1, \ldots, R_n\}$ and whose full joint $P_G$ factorises according to the local CPTs, i.e.,
  \[ P_G = \prod_i P(R_i | \text{parents}(R_i)) \]

- Gaussian BN is a BN where
  - $R_i$ are continuous randvars
  - All CPDs are linear Gaussians
  - E.g., $T_1 \rightarrow T_2 \rightarrow T_3$ (also depicted right)
    - $p(T_1) = \mathcal{N}(1; 4)$
    - $p(T_2 | T_1) = \mathcal{N}(-3.5 + 0.5 \cdot T_1; 4)$
    - $p(T_3 | T_2) = \mathcal{N}(1 + (-1) \cdot T_2; 3)$
Connection to Multivariate Gaussian

• Linear GBN an alternative representation to multivariate Gaussian distribution
  • A linear Gaussian BN always defines a joint multivariate Gaussian distribution

• Let $S$ be a linear Gaussian of its parents $R_1, \ldots, R_k$
  • $\mathcal{N}(\beta_0 + \beta^T r; \sigma^2) = \mathcal{N}(\beta_0 + \beta_1 r_1 + \cdots + \beta_k r_k; \sigma^2)$
  • $R_1, \ldots, R_k$ jointly Gaussian with $\mathcal{N}(\mu; \Sigma)$

• Distribution of $S$ is a Gaussian $p(S) = \mathcal{N}(\mu_S; \sigma_S^2)$ with
  \[
  \mu_S = \beta_0 + \beta^T r \\
  \sigma_S^2 = \sigma^2 + \beta^T \Sigma \beta
  \]

• Joint distribution over $\{R_1, \ldots, R_k, S\}$ is a Gaussian with
  
  \[
  \text{Cov}[R_i; S] = \sum_{j=1}^{k} \beta_j \Sigma_{ij}
  \]
General Procedure for Conversion

• Let \((R_1, ..., R_n)\) be the randvars of a GBN
  • Each \(R_i\) is a Gaussian \(\mathcal{N}(\beta_0 + \beta^T \mathbf{r}; \sigma^2)\) conditional on its parents \(\text{parents}(R_i)\)
  • \((R_1, ..., R_n)\) follows a topological ordering \(\theta\) s.t. \(\forall R_j \in \{R_1, ..., R_n\} : \forall R_i \in \text{parents}(R_j) : R_i \prec_\theta R_j\)
  • Build a matrix \(B^{n \times n}\) that has a non-zero entry \(\beta_{ij}\) if there exists a parent-child relation \(R_i \rightarrow R_j\) with \(\beta_{ij}\) being the factor for \(R_i\) in the \(\beta\) of \(R_j\)
    \[
    B = \begin{pmatrix}
    0 & \beta_{12} & \cdots & \beta_{1n} \\
    0 & 0 & \cdots & \beta_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & 0
    \end{pmatrix}
    \]
    • \(i\) chooses the row(s), \(j\) chooses the column(s)
  • \(B\) is upper-triangular because no loops allowed in BNs
    • Including self-loops \(\rightarrow \beta_{ii} = 0\) as well
General Procedure for Conversion

• Joint distribution $p(r_1, ..., r_n)$ given by $\mathcal{N} (\mu, \Sigma)$
  • Means
    \[
    \mu = \left( \mu_1, \beta_{0,2} + \beta_2^T r, ..., \beta_{0,n} + \beta_n^T r \right)^T
    \]
  • Covariance (recursive rules): $j \in \{2, ..., n\}, i = 1 ... j - 1$
    \[
    \begin{aligned}
    \Sigma_{11} &\leftarrow \sigma_1^2 \\
    \Sigma_{ij} &\leftarrow \Sigma_{ii} B_{ij} \\
    \Sigma_{ji} &\leftarrow \Sigma_{ij}^T \\
    \Sigma_{jj} &\leftarrow \sigma_j^2 + \Sigma_{ji} B_{ij}
    \end{aligned}
    \]
  • First index chooses the row(s), second index chooses the column(s)

given $B$  \quad \text{filling } \Sigma \text{ layer-wise:}

\[
B = \begin{pmatrix}
0 & \beta_{12} & \cdots & \beta_{1n} \\
0 & 0 & \cdots & \beta_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0
\end{pmatrix} \quad \Sigma = \begin{pmatrix}
\Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1n} \\
\Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{n1} & \Sigma_{n2} & \cdots & \Sigma_{nn}
\end{pmatrix}
\]
GBN: Conversion Example

- **GBN**
  
  \[ p(T_1) = \mathcal{N}(1; 4) \]
  
  \[ p(T_2|T_1) = \mathcal{N}(-3.5 + 0.5 \cdot T_1; 4) \]
  
  \[ p(T_3|T_2) = \mathcal{N}(1 + (-1) \cdot T_2; 3) \]

- **Goal: Joint distribution**
  
  \[ p(t_1, t_2, t_3) = \mathcal{N}(\mu; \Sigma) \]

  \[ \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} \]

  \[ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix} \]

- **Matrix B**
  
  \[ B_{12}: T_1 \to T_2, \beta_1 = 0.5 \]
  
  \[ B_{23}: T_2 \to T_3, \beta_1 = -1 \]
  
  - Rest: zeroes
  - Result:
    
    \[ B = \begin{pmatrix} 0 & 0.5 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} \]

- **Means**
  
  \[ \mu_1 = 1 \]
  
  \[ \mu_2 = -3.5 + 0.5 \cdot \mu_1 = -3 \]
  
  \[ \mu_3 = 1 + (-1) \cdot \mu_2 = 4 \]
  
  - Result:
    
    \[ \mu = \begin{pmatrix} 1 \\ -3 \\ 4 \end{pmatrix} \]
GBN: Conversion Example

- Filling $\Sigma$:
  \[
  \Sigma = \begin{pmatrix}
  \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\
  \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\
  \Sigma_{31} & \Sigma_{32} & \Sigma_{33}
  \end{pmatrix}
  \]

- Need $B$ and the recursive rules

- $j = 2, i = 1$

  - $\Sigma_{12} = \Sigma_{11} B_{12} = 4 \cdot 0.5 = 2$
  
  - $\Sigma_{21} = \Sigma_{12}^T = 2^T = 2$
  
  - $\Sigma_{22} = \sigma_2^2 + \Sigma_{21} B_{12} = 4 + 2 \cdot 0.5 = 5$

- First index: row(s)
- Second index: column(s)
GBN: Conversion Example

• Remaining goal:
  \[ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix} \]

• Need \( B \) and the recursive rules
  \[ B = \begin{pmatrix} 0 & 0.5 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix} \]

• \( j = 3, i = 12 \)
  \[ \Sigma_{(12)3} = \Sigma_{(12)(12)}B_{(12)3} \]
  \[ = \begin{pmatrix} 4 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \end{pmatrix} \]
  \[ = \begin{pmatrix} -2 \\ -5 \end{pmatrix} \]

• \( \Sigma_{3(12)} \)
  \[ = \Sigma_{(12)3}^T \]
  \[ = \begin{pmatrix} -2 \\ -5 \end{pmatrix} \]
  \[ = \begin{pmatrix} -2 & -5 \end{pmatrix} \]

• \( \Sigma_{33} \)
  \[ = \sigma_3^2 + \Sigma_{3(12)}B_{(12)3} \]
  \[ = 3 + (-2 - 5) \begin{pmatrix} 0 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ -2 \end{pmatrix} \]
  \[ = 3 + 5 = 8 \]
Inference in GBNs

• Inference in linear Gaussians with Variable Elimination
  • Representation through linear Gaussian CPDs instead of CPTs/factors
  • Modified operations for multiply/sum-out

• Message passing formulation
  • Approximate belief propagation

• Sampling in the continuous space
  • Rejection sampling, importance sampling, MCMC methods for GBNs

• Actually using the full joint
  • Marginalisation, conditioning as sketched in Basics

See Ch. 14 of PGM book for further information
Lifting the Full Joint

• Lifting conversion approach by Shachter and Kenley for parameterised GBNs
  • GBN with PRVs $A_1, \ldots, A_m$ as nodes
    • PDF for each $A_i$ applies to each $R \in \text{gr}(A_i)$
    • $m \ll n$, $n = |\bigcup_i \text{gr}(A_i)|$
    • Semantics: grounding and forming full joint $p(\bigcup_i \text{gr}(A_i))$
    • Simple case for GBNs (general case under review):
      For all parent-child relations $R(X) \rightarrow S(Y)$, it holds that $X \cap Y = \emptyset$
      • Each child instance has the same parent instances as its siblings

\[ p(R(X)) = \mathcal{N}(1; 4) \]
\[ p(S(Y)|R(X)) = \mathcal{N}(-3.5 + 0.5 \cdot T_1; 4) \]
\[ p(T(Z)|S(Y)) = \mathcal{N}(1 + (-1) \cdot T_2; 3) \]

Lifting the Full Joint: Simple Case

• With PRVs, matrix $B$ and covariance matrix have liftable blocks for each PRV
  • Given the case of no overlaps in logvars: $B$

\[
\begin{pmatrix}
0 & \cdots & 0 & \beta_{r_1s_1} & \cdots & \beta_{r_1s_m} & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & \beta_{r_n s_1} & \cdots & \beta_{r_n s_m} & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & \beta_{s_1 t_1} & \cdots & \beta_{s_1 t_l} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 0 & \cdots & 0 & \beta_{s_m t_1} & \cdots & \beta_{s_m t_l} \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0
\end{pmatrix}
\]
Lifting the Full Joint: Simple Case

• With PRVs, matrix $B$ and covariance matrix have liftable blocks for each PRV

• Given the case of no overlaps in logvars: $B$

Each $R(x_i)$ has the same influence on each $S(y_j)$. Given $P(s(Y)|r(X)) = \mathcal{N}(\beta_0 + \beta_1 r(X); \sigma^2)$, 
\[ \beta_{r_isj} = \beta_1 \]
for all $i \in \{1, ..., n\}, j \in \{1, ..., m\}$.
The same holds for $S(y_j)$ and $T(z_k)$ (as well as $R(x_i)$ and $T(z_k)$, which has $\beta_{r_itk} = 0$).
Lifting the Full Joint: Simple Case

- With PRVs, matrix $B$ and covariance matrix have liftable blocks for each PRV
- Given the case of no overlaps in logvars: $\Sigma$

$$
\begin{align*}
R(X) = & \begin{pmatrix}
\Sigma_{r_1r_1} & \cdots & \Sigma_{r_1r_n} \\
\vdots & \ddots & \vdots \\
\Sigma_{r_nr_1} & \cdots & \Sigma_{r_nr_n}
\end{pmatrix} \\
S(Y) = & \begin{pmatrix}
\Sigma_{s_1r_1} & \cdots & \Sigma_{s_1r_n} \\
\vdots & \ddots & \vdots \\
\Sigma_{s_mr_1} & \cdots & \Sigma_{s_mr_n}
\end{pmatrix} \\
T(Z) = & \begin{pmatrix}
\Sigma_{t_1r_1} & \cdots & \Sigma_{t_1r_n} \\
\vdots & \ddots & \vdots \\
\Sigma_{t_lr_1} & \cdots & \Sigma_{t_lr_n}
\end{pmatrix}
\end{align*}
$$
Lifting the Full Joint: Simple Case

\[ \Sigma_{r_1 r_1} = \sigma_2^2 \]
\[ \Sigma_{r_1 r_2} = \sigma_2^2 \bar{B}_{r_1 r_2} = \sigma_2^2 \bar{B}_{11} = \sigma_2^2 \cdot 0 = 0 \]
\[ \Sigma_{r_2 r_1} = 0 \]
\[ \Sigma_{r_2 r_2} = \sigma_2^2 + \Sigma_{r_1 r_2} \bar{B}_{r_1 r_2} = \sigma_2^2 + 0 = \sigma_2^2 \]

\[ R(X) \]
\[
\begin{pmatrix}
\Sigma_{r_1 r_1} & \cdots & \Sigma_{r_1 r_n} & \cdots & \Sigma_{r_1 s_1} & \cdots & \Sigma_{r_1 s_m} & \cdots & \Sigma_{r_1 t_1} & \cdots & \Sigma_{r_1 t_l}

\vdots & \ddots & \vdots

\Sigma_{r_n r_1} & \cdots & \Sigma_{r_n r_n} & \cdots & \Sigma_{r_n s_1} & \cdots & \Sigma_{r_n s_m} & \cdots & \Sigma_{r_n t_1} & \cdots & \Sigma_{r_n t_l}

\Sigma_{s_1 r_1} & \cdots & \Sigma_{s_1 r_n} & \cdots & \Sigma_{s_1 s_1} & \cdots & \Sigma_{s_1 s_m} & \cdots & \Sigma_{s_1 t_1} & \cdots & \Sigma_{s_1 t_l}

\vdots & \ddots & \vdots

\Sigma_{s_m r_1} & \cdots & \Sigma_{s_m r_n} & \cdots & \Sigma_{s_m s_1} & \cdots & \Sigma_{s_m s_m} & \cdots & \Sigma_{s_m t_1} & \cdots & \Sigma_{s_m t_l}

\vdots & \ddots & \vdots

\Sigma_{t_1 r_1} & \cdots & \Sigma_{t_1 r_n} & \cdots & \Sigma_{t_1 s_1} & \cdots & \Sigma_{t_1 s_m} & \cdots & \Sigma_{t_1 t_1} & \cdots & \Sigma_{t_1 t_l}

\vdots & \ddots & \vdots

\Sigma_{t_l r_1} & \cdots & \Sigma_{t_l r_n} & \cdots & \Sigma_{t_l s_1} & \cdots & \Sigma_{t_l s_m} & \cdots & \Sigma_{t_l t_1} & \cdots & \Sigma_{t_l t_l}
\end{pmatrix}
\]

\[ \Sigma_{11} \leftarrow \sigma^2_1 \]
\[ \Sigma_{ij} \leftarrow \Sigma_{ii} \bar{B}_{ij} \]
\[ \Sigma_{ji} \leftarrow \Sigma_{ij}^T \]
\[ \Sigma_{jj} \leftarrow \sigma^2_j + \Sigma_{ji} \bar{B}_{ij} \]

\[
\begin{pmatrix}
0 & \beta_{rs} & 0 \\
0 & 0 & \beta_{st} \\
0 & 0 & 0
\end{pmatrix}
\]

\[ \bar{B} \]

on-diagional: \[ \sigma_2^2 \]

off-diagonals: 0
Lifting the Full Joint: Simple Case

\[
\Sigma(r_1...r_n)s_1
= \Sigma(r_1...r_n)(r_1...r_n)B(r_1...r_n)s_1
= (\begin{pmatrix}
\sigma_R^2 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma_R^2
\end{pmatrix})(\begin{pmatrix}
\beta_{rs}
\vdots
\vdots
\beta_{rs}
\end{pmatrix})
= (\begin{pmatrix}
\sigma_R^2 \beta_{rs} \\
\vdots \\
\sigma_R^2 \beta_{rs}
\end{pmatrix})
= (4 \cdot 0.5) = (2)
\]

\[
\Sigma_{s1s1}
= \sigma_S^2(y) + \Sigma_{s1}(r_1...r_n)B(r_1...r_n)s_1
= \sigma_S^2(y) + (\sigma_R^2(\beta_{rs}) \cdots \sigma_R^2(\beta_{rs}))
= \sigma_S^2(y) + n\sigma_R^2(\beta_{rs})^2
= 4 + n \cdot 4 \cdot 0.5^2 = 4 + n
\]
Lifting the Full Joint: Simple Case

\[
\Sigma_{(r_1...r_n s_1) s_2} = \Sigma_{(r_1...r_n s_1)(r_1...r_n s_1)} B_{(r_1...r_n s_1) s_2} \left( \begin{array}{ccc}
\sigma_R^2(X) & ... & 0 \\
... & ... & ... \\
0 & ... & \sigma_R^2(X) \\
\sigma_R^2(X) \beta_{rs} & ... & \sigma_R^2(X) \beta_{rs} \\
... & ... & ... \\
n \sigma_R^2(X) \beta_{rs} & ... & \sigma_R^2(X) \beta_{rs} \\
\end{array} \right) \left( \begin{array}{c}
\beta_{rs} \\
... \\
\beta_{rs} \\
\end{array} \right)
\]

\[
= \left( \begin{array}{ccc}
\sigma_R^2(X) & ... & 0 \\
... & ... & ... \\
0 & ... & \sigma_R^2(X) \\
\sigma_R^2(X) \beta_{rs} & ... & \sigma_R^2(X) \beta_{rs} \\
... & ... & ... \\
n \sigma_R^2(X) \beta_{rs} & ... & \sigma_R^2(X) \beta_{rs} \\
\end{array} \right) = \left( \begin{array}{c}
4 \cdot 0.5 \\
... \\
4 \cdot 0.5 \\
n \cdot 4 \cdot 0.5^2 \\
\end{array} \right) = \left( \begin{array}{c}
2 \\
... \\
2 \\
n \\
\end{array} \right)
\]

\[
\Sigma_{s_2 s_2} = \sigma_s^2 + \Sigma_{s_2 (r_1...r_n s_1)} B_{(r_1...r_n s_1) s_2} = \sigma_s^2 + (\sigma_R^2(X) \beta_{rs} \ldots \sigma_R^2(X) \beta_{rs} \ldots n \sigma_R^2(X) \beta_{rs}^2) \left( \begin{array}{c}
\beta_{rs} \\
... \\
\beta_{rs} \\
\end{array} \right)
\]

\[
= \sigma_s^2 + n \sigma_R^2(X) \beta_{rs}^2 = 4 + n
\]
Lifting the Full Joint: Simple Case

\[
\begin{bmatrix}
\sigma^2_R(x) & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma^2_R(x)
\end{bmatrix}
\quad \text{on-diagonal: } \sigma^2_R(x)
\]

\[
\begin{bmatrix}
\sigma^2_R(x)\beta_{rs} & \cdots & \sigma^2_R(x)\beta_{rs} \\
\vdots & \ddots & \vdots \\
\sigma^2_R(x)\beta_{rs} & \cdots & \sigma^2_R(x)\beta_{rs}
\end{bmatrix}
\quad \text{all: } \sigma^2_R(x)\beta_{rs}
\]

\[
\begin{bmatrix}
\sigma^2_S(y) + n\sigma^2_R(x)\beta_{rs}^2 & \cdots & n\sigma^2_R(x)\beta_{rs}^2 \\
\vdots & \ddots & \vdots \\
n\sigma^2_R(x)\beta_{rs}^2 & \cdots & \sigma^2_S(y) + n\sigma^2_R(x)\beta_{rs}^2
\end{bmatrix}
\quad \text{on-diagonal: } \sigma^2_S(y) + n\sigma^2_R(x)\beta_{rs}^2
\]

\[
\begin{bmatrix}
\sigma^2_S(y) + n\sigma^2_R(x)\beta_{rs}^2 & \cdots & n\sigma^2_R(x)\beta_{rs}^2 \\
\vdots & \ddots & \vdots \\
n\sigma^2_R(x)\beta_{rs}^2 & \cdots & \sigma^2_S(y) + n\sigma^2_R(x)\beta_{rs}^2
\end{bmatrix}
\quad \text{off-diagonal: } n\sigma^2_R(x)\beta_{rs}^2
\]

\[
\begin{bmatrix}
\Sigma r_1 r_1 & \cdots & \Sigma r_1 r_n \\
\vdots & \ddots & \vdots \\
\Sigma s_1 r_1 & \cdots & \Sigma s_1 r_n
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 s_1 & \cdots & \Sigma r_1 s_m \\
\vdots & \ddots & \vdots \\
\Sigma s_1 s_1 & \cdots & \Sigma s_1 s_m
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 t_1 & \cdots & \Sigma r_1 t_l \\
\vdots & \ddots & \vdots \\
\Sigma s_1 t_1 & \cdots & \Sigma s_1 t_l
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Sigma r_1 r_1 & \cdots & \Sigma r_1 r_n \\
\vdots & \ddots & \vdots \\
\Sigma s_1 r_1 & \cdots & \Sigma s_1 r_n
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 s_1 & \cdots & \Sigma r_1 s_m \\
\vdots & \ddots & \vdots \\
\Sigma s_1 s_1 & \cdots & \Sigma s_1 s_m
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 t_1 & \cdots & \Sigma r_1 t_l \\
\vdots & \ddots & \vdots \\
\Sigma s_1 t_1 & \cdots & \Sigma s_1 t_l
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Sigma r_1 r_1 & \cdots & \Sigma r_1 r_n \\
\vdots & \ddots & \vdots \\
\Sigma s_1 r_1 & \cdots & \Sigma s_1 r_n
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 s_1 & \cdots & \Sigma r_1 s_m \\
\vdots & \ddots & \vdots \\
\Sigma s_1 s_1 & \cdots & \Sigma s_1 s_m
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 t_1 & \cdots & \Sigma r_1 t_l \\
\vdots & \ddots & \vdots \\
\Sigma s_1 t_1 & \cdots & \Sigma s_1 t_l
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Sigma r_1 r_1 & \cdots & \Sigma r_1 r_n \\
\vdots & \ddots & \vdots \\
\Sigma s_1 r_1 & \cdots & \Sigma s_1 r_n
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 s_1 & \cdots & \Sigma r_1 s_m \\
\vdots & \ddots & \vdots \\
\Sigma s_1 s_1 & \cdots & \Sigma s_1 s_m
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 t_1 & \cdots & \Sigma r_1 t_l \\
\vdots & \ddots & \vdots \\
\Sigma s_1 t_1 & \cdots & \Sigma s_1 t_l
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Sigma r_1 r_1 & \cdots & \Sigma r_1 r_n \\
\vdots & \ddots & \vdots \\
\Sigma s_1 r_1 & \cdots & \Sigma s_1 r_n
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 s_1 & \cdots & \Sigma r_1 s_m \\
\vdots & \ddots & \vdots \\
\Sigma s_1 s_1 & \cdots & \Sigma s_1 s_m
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 t_1 & \cdots & \Sigma r_1 t_l \\
\vdots & \ddots & \vdots \\
\Sigma s_1 t_1 & \cdots & \Sigma s_1 t_l
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Sigma r_1 r_1 & \cdots & \Sigma r_1 r_n \\
\vdots & \ddots & \vdots \\
\Sigma s_1 r_1 & \cdots & \Sigma s_1 r_n
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 s_1 & \cdots & \Sigma r_1 s_m \\
\vdots & \ddots & \vdots \\
\Sigma s_1 s_1 & \cdots & \Sigma s_1 s_m
\end{bmatrix}
\begin{bmatrix}
\Sigma r_1 t_1 & \cdots & \Sigma r_1 t_l \\
\vdots & \ddots & \vdots \\
\Sigma s_1 t_1 & \cdots & \Sigma s_1 t_l
\end{bmatrix}
\]
### Lifting the Full Joint: Simple Case

**R(X)**

<table>
<thead>
<tr>
<th>( T(Z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m\sigma^2_{R(X)}\beta_{rs}\beta_{st} )</td>
</tr>
</tbody>
</table>

\[ \vdots \quad \vdots \quad \vdots \]

\[ \rightarrow \text{all: } m\sigma^2_{R(X)}\beta_{rs}\beta_{st} = m \cdot 4 \cdot 0.5 \cdot (-1) = -2m \]

**S(Y)**

<table>
<thead>
<tr>
<th>( T(Z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (\sigma^2_{S(Y)} + mn\sigma^2_{R(X)}\beta_{rs}^2)\beta_{st} )</td>
</tr>
</tbody>
</table>

\[ \vdots \quad \vdots \quad \vdots \]

\[ \rightarrow \text{all: } (\sigma^2_{S(Y)} + mn\sigma^2_{R(X)}\beta_{rs}^2)\beta_{st} = -4 - mn \]

**T(Z)**

| \( \sigma^2_{T(Z)} + m\beta_{st}^2(\sigma^2_{S(Y)} + mn\sigma^2_{R(X)}\beta_{rs}^2) \) | \( \cdots \) | \( m\beta_{st}^2(\sigma^2_{S(Y)} + mn\sigma^2_{R(X)}\beta_{rs}^2) \) |

\[ \vdots \quad \vdots \quad \vdots \]

\[ \rightarrow \text{on-diagonal: } \sigma^2_{T(Z)} + m\beta_{st}^2(\sigma^2_{S(Y)} + mn\sigma^2_{R(X)}\beta_{rs}^2) = 3 + 4m + m^2n \]

\[ \rightarrow \text{off-diagonal: } m\beta_{st}^2(\sigma^2_{S(Y)} + mn\sigma^2_{R(X)}\beta_{rs}^2) = 4m + m^2n \]
Lifted Joint

• Only two structures required for covariance matrix
  - A matrix
    \[
    R(X) = \begin{pmatrix}
    0 & \sigma_{R(X)}^2 \beta_{rs} \\
    \sigma_{R(X)}^2 \beta_{rs} & n \sigma_{R(X)}^2 \beta_{rs}^2 \\
    m \sigma_{R(X)}^2 \beta_{rs} \beta_{st} & (\sigma_{S(Y)}^2 + mn \sigma_{R(X)}^2 \beta_{rs}^2) \beta_{st} \\
    -2m & -4 - mn & 4m + m^2 n
    \end{pmatrix}
    \]
  - A vector for on-diagonal covariance entries
    - Individual variances
      - Have to be stored anyway

Lifted Joint

• Only two structures required for covariance matrix
• Depend only on the number of PRVs, not the domain sizes!

Lifted Query Answering

• Marginal queries
  • Read off values in (lifted) covariance representation

• Conditional queries $R|E = e \sim \mathcal{N}(\mu^*, \Sigma^*)$
  • $\mu^* = \mu_R + \Sigma_{RE} \Sigma_{EE}^{-1}(e - \mu_E)$
  • $\Sigma^* = \Sigma_{RR} - \Sigma_{RE} \Sigma_{EE}^{-1} \Sigma_{ER}$
  • Matrix multiplication, inversion required
    • Possible to compute them in a lifted manner due to block structure
      • Proof in paper by Hartwig and Möller (2020)

• Evidence is ground
  • Probably no symmetries in observations with real numbers as range values
    → unlikely to get identical observations
    • Fig.: 50% of ground instances get random values assigned as evidence
Interim Summary

• Linear Gaussian models
  • Linear dependency between child and parent randvars
  • Full joint given by vector of means and covariance matrix
    • Information form as inverse of covariance form
  • Query answering
    • Marginal using covariance matrix
    • Conditional using information form

• Gaussian BNs
  • Explicitly encode independencies in network structure
    • Conditional linear Gaussian
  • GBN = multivariate Gaussian distribution
  • Lifting for PRVs without an overlap in logvars between parent and child
Hybrid Models

• Models that contain discrete ($D_i$ in fig.) and continuous randvars ($X_i$ in fig.)

• Some general results
  • Even representing the correct marginal distribution in a hybrid network can require space that is exponential in the size of the network
  • Query answering problem is NP-hard even if the GBN is a polytree where all discrete randvars are Boolean-valued and where every continuous randvar has at most one discrete ancestor
    • There are not even approximate algorithms to solve the problem in polynomial time with a useful error bound without further restrictions

Figures from PGM book by Koller and Friedman, p. 615+616.
Outline: 8. Continuous Space

A. Basics
   • Continuous variables, probability density function, cumulative probability distribution
   • Joint distribution, marginal density, conditional density

B. Gaussian models
   • (Multivariate) Gaussian distribution
   • (Parameterised) Gaussian Bayesian networks

C. Probabilistic Soft Logic (PSL)
   • Modelling, semantics, inference task
Probabilistic Soft Logic (PSL)

- Logic-based approach
- Probabilistic programming language
  - Predicate = relationship or property
  - Atom = continuous randvar
  - Rule = dependency or constraint
  - Set = define aggregates
- PSL program = rules + input database
- Implementation: https://psl.linqs.org

Syntax & Semantics

• Let $R$ be a set of weighted logical rules, each $R_j$ has the form

$$w_j : \bigwedge_{i \in I_j^-} x_i \Rightarrow \bigvee_{i \in I_j^+} x_i$$

• $w_j \geq 0$

• Sets $I_j^-, I_j^+$ index conjuncted/disjuncted literals

• Equivalent clausal form:

$$\left( \bigvee_{i \in I_j^+} x_i \right) \lor \left( \bigvee_{i \in I_j^-} \neg x_i \right)$$

• Probability distribution (compare: MLNs)

$$P(x) \propto \exp \left( \sum_{R_j \in R} w_j \left( \bigvee_{i \in I_j^+} x_i \right) \lor \left( \bigvee_{i \in I_j^-} \neg x_i \right) \right)$$
MPE Inference

• MPE: Find the most probable assignment to the unobserved randvars
  • I.e., given a model ground over an input database,
    \[ \arg\max_x \sum_{R_j \in R} w_j \left( \bigvee_{i \in I_j^+} x_i \right) \bigvee \left( \bigvee_{i \in I_j^-} \neg x_i \right) \]
  • Combinatorial, NP-hard

• Approximation:
  View as optimising rounding probabilities
Expected Score

• Expected score of a clause is the weight times the probability that at least one literal is true:

\[ w_j \left( 1 - \prod_{i \in I_j^+} (1 - p_i) \prod_{i \in I_j^-} p_i \right) \]

• Then, expected total score is

\[ \hat{W} = \sum_{R_j \in R} w_j \left( 1 - \prod_{i \in I_j^+} (1 - p_i) \prod_{i \in I_j^-} p_i \right) \]

• But, \( \text{argmax} \hat{W} \) highly non-convex due to product \( p \)

\[ \rightarrow \text{or-semantics} \rightarrow \text{trick:} \]

Instead of computing \( P(A \lor B) \)
\[ = P(A) + P(B) - P(A \land B) \]
compute
\[ P(\neg \neg (A \lor B)) \]
\[ = 1 - P(\neg A \land \neg B) \]
Approximate Inference

• Instead: Optimise a linear program that bounds expected score

\[
\sum_{R_j \in R} w_j \left( 1 - \prod_{i \in I^+_j} (1 - p_i) \prod_{i \in I^-_j} p_i \right) \geq \left( 1 - \frac{1}{e} \right) \sum_{R_j \in R} w_j \min \left\{ \sum_{i \in I^+_j} p_i + \sum_{i \in I^-_j} (1 - p_i), 1 \right\}
\]

• Can give \(1 - \frac{1}{e}\)-optimal discrete solution
Scalable Approximate Inference

- Linear programming algorithms do not scale well to big probabilistic models

- Instead of solving the problem as one big optimisation, decompose the problem based on its graphical structure
  - Compare: cliques/clusters
Consensus Optimisation

• Decompose problem and solve sub-problems independently (in parallel), then merge results
  • Sub-problems are ground rules
  • Auxiliary variables enforce consensus across sub-problems

• Framework:
  Alternating direction method of multipliers (ADMM) (Boyd, 2011)
  • Guaranteed to converge for convex problems
  • Inference with ADMM fast, scalable, straightforward to implement (Bach et al, 2017)

Local Consistency Relaxation

- Relax search over consistent marginals to simpler set

\[
\arg\max_{\mu \in [0,1]^n} \sum_{R_j \in R} w_j \min \left\{ \sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i), 1 \right\}
\]
Continuous Variables & Similarity

• Continuous values interpreted as similarities
  • E.g., multiple ontologies → alignment
    • Match/Don’t match → similar to what extent?

⇒ Soft logic
Soft Logic

- Logical operators defined for continuous values in the [0,1] interval
  - Interpret as similarities or degree of truth

- Łukasiewicz logic
  - $p \land q = \max\{p + q - 1, 0\}$
  - $p \lor q = \min\{p + q, 1\}$
  - $\neg p = 1 - p$

- PSL: Use Łukasiewicz logic to interpret rules
  - Hinge-loss MNs (or Markov random fields as called in the publications by the PSL team) formalise this
Hinge-loss MNs

• Relaxed, logic-based MNs can reason about both discrete and continuous graph data scalably and accurately
  • General objective
    \[
    \arg\max_{y \in [0,1]^n} \sum_{j=1}^{m} w_j \min \left\{ \sum_{i \in I_j^+} y_i + \sum_{i \in I_j^-} (1 - y_i), 1 \right\}
    \]
    \[
    = \arg\min_{y \in [0,1]^n} \sum_{j=1}^{m} w_j \max \left\{ 1 - \sum_{i \in I_j^+} y_i - \sum_{i \in I_j^-} (1 - y_i), 0 \right\}
    \]
  • Notion of distance to satisfaction

Distance to Satisfaction

\[
\arg\min_{y \in [0,1]^n} \sum_{j=1}^{m} w_j \max \left\{ 1 - \sum_{i \in I_j^+} y_i - \sum_{i \in I_j^-} (1 - y_i), 0 \right\}
\]

• Maximum value of any unweighted term is 1
  • Term is satisfied

• Unsatisfied term → distance to satisfaction
  • How far it is from achieving its maximum value
  • Each unweighted objective term measures how far the linear constraint is away from being satisfied:

\[
1 - \sum_{i \in I_j^+} y_i - \sum_{i \in I_j^-} (1 - y_i) \leq 0
\]
Relaxed Linear Constraints

• Instead of requiring logical clauses, each term can be defined using any function \( \ell_j(y) \) linear in \( y \)

\[
\arg\min_{y \in [0,1]^n} \sum_{j=1}^m w_j \max\{\ell_j(y), 0\}
\]

• Each term represents the distance to satisfaction of a linear constraint \( \ell_j(y) \leq 0 \)
  • Can use logical clauses or something else based on domain knowledge
  • Also called hinge losses
  • Sometimes \( \max\{\ell_j(y), 0\} \) gets squared to better trade off conflicting objective terms

• Weight indicates how important it is to satisfy a constraint relative to others by scaling the distance to satisfaction
Hinge-loss MNs

- Let \( y = (y_1, \ldots, y_n) \) be a vector of \( n \) randvars and \( x = (x_1, \ldots, x_{n'}) \) be a vector of \( n' \) randvars with joint range \( D = [0,1]^{n+n'} \)

- Let \( \phi = (\phi_1, \ldots, \phi_m) \) be a vector of \( m \) continuous potentials of the form
  \[
  \phi_j(y, x) = \left( \max\{\ell_j(y, x), 0\} \right)^{p_j}
  \]
  - \( \ell_j(y, x) \) linear function of \( y, x \)
  - \( p_j \in \{1,2\} \)

- For \( (y, x) \in D \) and given a vector of \( m \) weights \( w = (w_1, \ldots, w_m) \), constrained hinge-loss energy function \( f_w \) is defined as
  \[
  f_w(y, x) = \sum_{j=1}^{m} w_j \phi_j(y, x)
  \]
Hinge-loss MNs

- Let $c = (c_1, ..., c_r)$ be a vector of linear constraint functions which further restrict the domain $D$ to $D'$
- **Hinge-loss MN** over randvars $y$ and conditioned on randvars $x$ is a PDF defined as follows
  - if $(y, x) \notin D'$, then $P(y|x) = 0$
  - if $(y, x) \in D'$, then
    $$P(y|x) = \frac{1}{Z(w, x)} \exp(-f_w(y, x))$$
    - where
      $$Z(w, x) = \int_{y | (y, x) \in D'} \exp(-f_w(y, x)) \, dy$$
- Define hinge-loss MNs using PSL
Application: E.g., Entity Resolution

• Goal: Identify references that denote the same person

• Use model to express dependencies
  • “If A=B and B=C, then A and C must also denote the same person”
  • “If two people have similar names, they are probably the same”
  • “If two people have similar friends, they are probably the same”
Interim Summary

- PSL
  - Logic programming language
  - Approximations
    - Linear program that bounds MPE solution from below
    - Decomposition of PGM to optimise set of subproblems (consensus optimisation)
    - Local consistency relaxation
  - Soft logic: Łukasiewicz logic
    - Interpret continuous values as similarities/degree of truth
- Hinge-loss MNs
  - Notion of distance to satisfaction
  - Define using PSL
Outline: 8. Continuous Space

A. Basics
   • Continuous variables, probability density function, cumulative probability distribution
   • Joint distribution, marginal density, conditional density

B. Gaussian models
   • (Multivariate) Gaussian distribution
   • (Parameterised) Gaussian Bayesian networks

C. Probabilistic Soft Logic (PSL)
   • Modelling, semantics, inference task

The End