

Bayesian networks

Introduction and basis

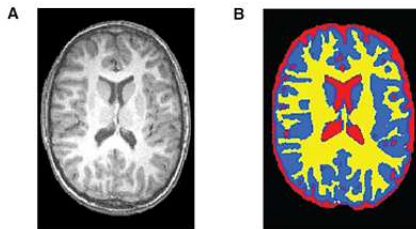
Helge Langseth and Thomas Dyhre Nielsen*

* Some of the introductory slides are stolen from Manfred Jaeger, Aalborg University.



Simultaneous Localization and Mapping: learn a map of the environment and locate current position

Example 2: Image Segmentation



(source: <http://pubs.niaaa.nih.gov/publications/arh313/243-246.htm>)

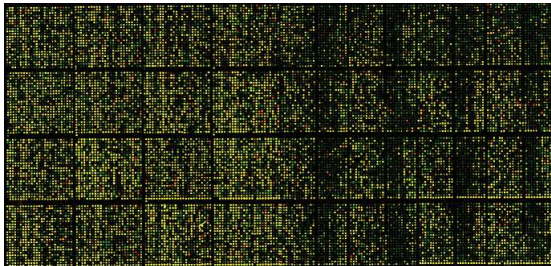
- Divide image into small number of regions representing structurally similar areas

Given a collection of texts:



Goal: automatically learn semantic descriptors for documents and words, that support document clustering, text understanding, information retrieval ...

Micro-array gene expression data:



Which genes are expressed under which conditions? Which are co-regulated, or functionally dependent?

Common ground in 4 examples:

- Use a **probabilistic model**, typically learned from data (using **statistical learning techniques**)
- Apply probabilistic inference algorithms to use models for prediction (**classification, regression**), structure analysis (**clustering, segmentation**)

Advantages of probabilistic/statistical methods:

- Principled quantification of prediction uncertainties
- Robust and principled techniques to deal with incomplete information, missing data.

Need: probabilistic models that ...

- can represent models for high-dimensional state spaces
- support efficient learning and inference techniques

Probabilistic Graphical Models ...

- support a structured specification of high-dimensional distributions in terms of low-dimensional factors
- structured representation can be exploited for efficient learning and inference algorithms (sometimes ...)
- graphical representation gives human-friendly design and description possibilities

Day 1: Bayesian networks – Definition and inference

- Definition of Bayesian networks: Syntax and semantics
- Exact inference
- Approximate inference using MCMC

Day 2: Variational inference – Introduction and basis

- Approximate inference through the *Kullback-Leibler divergence*
- *Variational Bayes*
- The *mean-field* approach to Variational Bayes

Day 3: Variational Bayes – cont'd

- Solving the VB equations
- Introducing Exponential families

Day 4: Scalable Variational Bayes

- Variational message passing
- Stochastic gradient ascent
- Stochastic variational inference

Day 5: Current approaches and extensions

- Variational Auto Encoders
- Black Box variational inference
- Probabilistic Programming Languages

I will assume that these topics are (fairly) well known:

- Probabilities, $P(X = x)$; conditional probabilities, $P(X = x | Y = y)$.
- Independence, $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}$; conditional independence $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$.
- “Standard” probability calculus:

Product: $P(x, y) = P(x | y) \cdot P(y) = P(x) \cdot P(y | x)$.

Sum-rule: $P(x \vee y) = P(x) + P(y) - P(x \wedge y)$.

Total probability: $P(Y = y) = \sum x' P(y | X = x') \cdot P(X = x')$.

Bayes rule: $P(x | y) = P(x) \cdot P(y | x) / P(y)$.

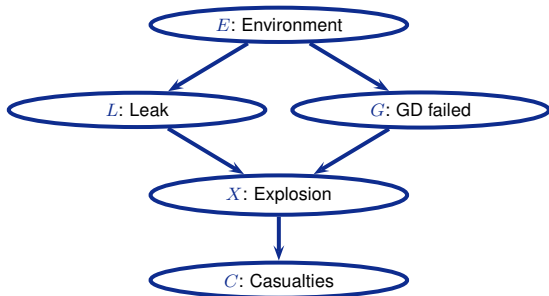
I will assume that these topics are (somewhat) known:

- Bayesian network syntax and semantics.
- Exact inference in Bayesian networks.
- Approximate inference using MCMC.

Implementation tasks:

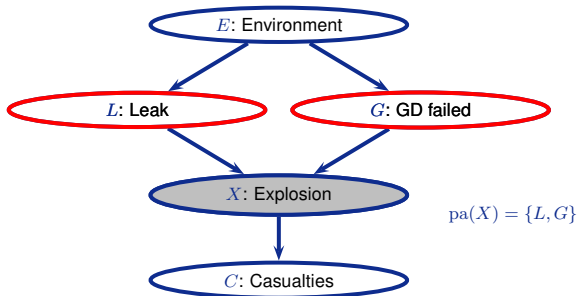
- We will get some implementation tasks as we move along.
- You will be supplied partly running Python-code (in the form of Jupyter notebooks).
- You will need to have **Python 3.x** on your computer (<https://www.python.org/downloads/>), and a set of packages (numpy, scipy, matplotlib, jupyter).

A simple example: “Explosion”



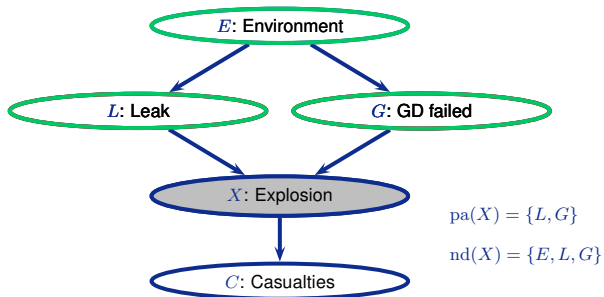
$$P(E, L, G, X, C)$$

A simple example: “Explosion”



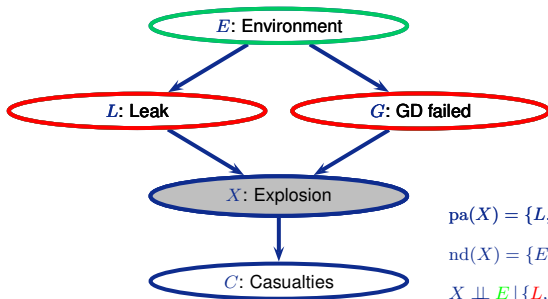
$$P(E, L, G, X, C)$$

A simple example: “Explosion”



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

A simple example: “Explosion”



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

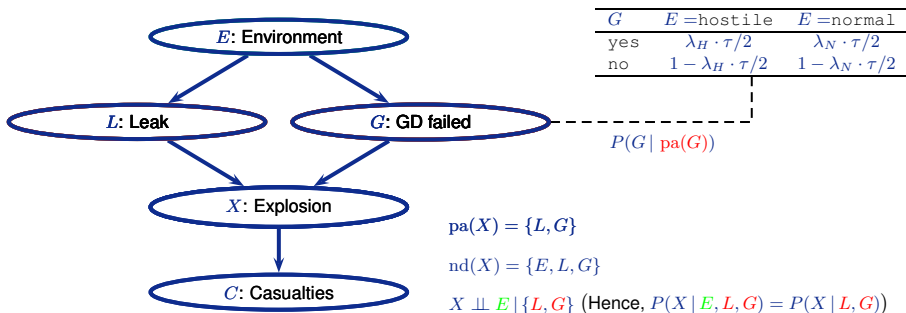
$$\text{nd}(X) = \{E, L, G\}$$

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

Other d-sep. rules: Jensen&Nielsen (07)

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

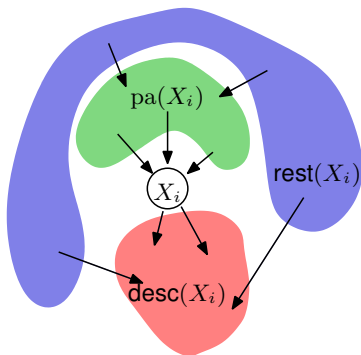
A simple example: “Explosion”



Other d-sep. rules: Jensen&Nielsen (07)

$$\begin{aligned}
 P(E, L, G, X, C) &= P(E) \cdot P(L \mid E) \cdot P(G \mid E, L) \cdot P(X \mid E, L, G) \cdot P(C \mid E, L, G, X) \\
 &= P(E) \cdot P(L \mid E) \cdot P(G \mid E) \cdot P(X \mid L, G) \cdot P(C \mid X)
 \end{aligned}$$

Markov properties \Leftrightarrow Factorisation property



In the distribution P defined by the BN the following independence relation holds:

$$P(X_i \mid \text{pa}(X_i), \text{rest}(X_i)) = P(X_i \mid \text{pa}(X_i))$$

“ X_i is independent of its non-descendants given its parents”

Bayesian network syntax

Let X_1, \dots, X_n be a collection of random variables. A Bayesian network over X_1, \dots, X_n consists of

- a directed acyclic graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ whose nodes \mathcal{V} are the variables X_1, \dots, X_n
- a set of local conditional distributions, $\mathcal{P} = \{p(X_i \mid \text{pa}(X_i)), X_i \in \mathcal{V}\}$, where $\text{pa}(X_i)$ are the parents of X_i in \mathcal{G} as defined by the edges \mathcal{E} .

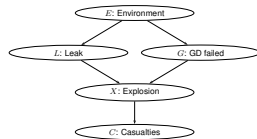
Bayesian network semantics

A Bayesian network \mathcal{N} with nodes X_1, \dots, X_n defines a joint distribution

$$p_{\mathcal{N}}(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i \mid \text{pa}(X_i))$$

Probability propagation

$$\left. \begin{array}{l} \text{Bayesian network} \\ + \\ \text{Evidence: } \mathbf{X}_E = \mathbf{x}_e \end{array} \right\} \Rightarrow P(\mathbf{X}_Q | \mathbf{x}_e)?$$

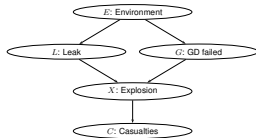


Example: Calculate $P(X = x | G = g)$

$$\begin{aligned} P(x, g) &= \sum_e \sum_l \sum_c P(E = e, L = l, g, X = x, C = c) \\ &= \sum_e \sum_l \sum_c P(e) \cdot P(l | e) \cdot P(g | e) \cdot P(x | l, g) \cdot P(c | x) \\ &= \sum_e P(e) \cdot P(g | e) \sum_l P(l | e) \cdot P(x | l, g) \sum_c P(c | x) \end{aligned}$$

Probability propagation

$$\left. \begin{array}{l} \text{Bayesian network} \\ + \\ \text{Evidence: } \mathbf{X}_E = \mathbf{x}_e \end{array} \right\} \Rightarrow P(\mathbf{X}_Q | \mathbf{x}_e)?$$

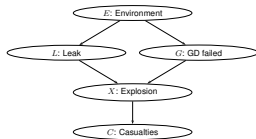


Example: Calculate $P(X = x | G = g)$

$$P(x, g) = \sum_e P(e) \cdot P(g | e) \sum_l P(l | e) \cdot P(x | l, g)$$
$$P(X = x | G = g) = \frac{P(x, g)}{\sum_{x'} P(X = x', g)} \propto P(x, g)$$

Probability propagation

$$\left. \begin{array}{l} \text{Bayesian network} \\ + \\ \text{Evidence: } \mathbf{X}_E = \mathbf{x}_e \end{array} \right\} \Rightarrow P(\mathbf{X}_Q | \mathbf{x}_e)?$$



Operations to calculate $P(\mathbf{X}_Q | \mathbf{x}_e)$

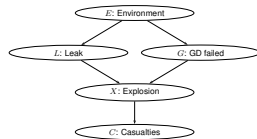
Restriction: Restrict domain of a potential (e.g., $P(g|E)$ from $P(G|E)$)

Combination: Multiplication of potentials (e.g., $P(l|e) \cdot P(X|l,g)$)

Marginalisation: Sum/integrate out a variable from a potential, e.g., the operation $\sum_l P(l|e) \cdot P(x|l,g)$, which removes L from the potential over $\{L, X, E, G\}$ and results in $P(x|e,g)$ over $\{X, E, G\}$.

Probability propagation

$$\left. \begin{array}{l} \text{Bayesian network} \\ + \\ \text{Evidence: } \mathbf{X}_E = \mathbf{x}_e \end{array} \right\} \Rightarrow P(\mathbf{X}_Q | \mathbf{x}_e)?$$



Requirements for efficient calculation of $P(\mathbf{X}_Q | \mathbf{x}_e)$

- Constraints wrt. structure:
 - Size of combined potentials
- Constraints wrt. distributions:
 - Ability to **perform** operations
 - Ability to **represent results** of operations

Bayesian network inference

Inference in the Bayesian network amounts to calculating $p(\mathbf{Z} = \mathbf{z} \mid \mathbf{X} = \mathbf{x})$, where

- $\mathbf{X} \subset \mathcal{V}$ are the observed variables, currently taking the configuration $\mathbf{X} = \mathbf{x}$.
- $\mathbf{Z} \subseteq \mathcal{V} \setminus \mathbf{X}$ are our variables of interest.

Inference is therefore the tool to answer any probabilistic query we may have in our domain, given a partial (or empty) observation from the domain

- *“What is the chance that Almería will move back to La Liga before 2020, given that they didn’t win during the first four games of this campaign?”*
- *“What is the probability of Google going bankrupt before you are done with your education?”*

Exact inference

Computation of *conditional distributions*

Given $\mathbf{X} = X_1, \dots, X_k \subset \mathcal{V}$, $x_i \in \text{dom}(X_i)$, $\mathbf{Z} = Z_1, \dots, Z_l \subset \mathcal{V}$, compute

$$p(\mathbf{Z} \mid \mathbf{X} = \mathbf{x})$$

Especially: *Single variable posterior distributions*: for each $Z \notin \mathbf{X}$ compute $p(Z \mid \mathbf{X} = \mathbf{x})$.

Problem reduction

This problem can be reduced to the computation of partial distributions, i.e., functions

$$p(\mathbf{Z}, \mathbf{X} = \mathbf{x})$$

(function defined on $\text{dom}(\mathbf{Z})$) because

$$p(\mathbf{Z} \mid \mathbf{X} = \mathbf{x}) = p(\mathbf{Z}, \mathbf{X} = \mathbf{x}) / p(\mathbf{X} = \mathbf{x})$$

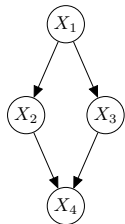
Direct approach: Denote $\mathbf{V}(= V_1, \dots, V_m) := \mathcal{V} \setminus (\mathbf{X} \cup \mathbf{Z})$, and let \mathbf{v} range over $\text{dom}(\mathbf{V})$. Then for $\mathbf{z} \in \text{dom}(\mathbf{Z})$:

$$\begin{aligned} p(\mathbf{Z} = \mathbf{z}, \mathbf{X} = \mathbf{x}) &= \sum_{\mathbf{v}} p(\mathbf{Z} = \mathbf{z}, \mathbf{X} = \mathbf{x}, \mathbf{V} = \mathbf{v}) \\ &= \sum_{\mathbf{v}} \prod_i p(Y_i \mid \text{pa}(Y_i))(\mathbf{z}, \mathbf{x}, \mathbf{v}) \\ &= \sum_{v_1 \in \text{dom}(V_1)} \cdots \sum_{v_m \in \text{dom}(V_m)} \prod_i p(Y_i \mid \text{pa}(Y_i))(\mathbf{z}, \mathbf{x}, \mathbf{v}) \end{aligned}$$

“Algorithm”: Sum out the v_j one by one, move factors $p(Y_i \mid \text{pa}(Y_i))(\mathbf{z}, \mathbf{x}, \mathbf{v})$ that do not depend on current v_j (because $V_j \notin \{Y_i\} \cup \text{pa}(Y_i)$) out of the sum.

Advantage: Can be used to compute conditional distributions for arbitrary set \mathbf{Z} of query variables.

Example



	X ₁	
	t	f
	.5	.5

	X ₂		
X ₁	t	f	
t	.7	.3	
f	.1	.9	

	X ₃		
X ₁	t	f	
t	.7	.3	
f	.2	.8	

		X ₄	
X ₂	X ₃	t	f
t	t	.9	.1
t	f	.7	.3
f	t	.8	.2
f	f	.4	.6

$$\begin{aligned}
 p(X_2 = x_2, X_4 = f) &= \sum_{x_1, x_3 \in \{t, f\}} p(X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4 = f) \\
 &= \sum_{x_1, x_3} \left[p(X_1 = x_1) p(X_2 = x_2 \mid X_1 = x_1) p(X_3 = x_3 \mid X_1 = x_1) \right. \\
 &\quad \left. p(X_4 = f \mid X_2 = x_2, X_3 = x_3) \right] \\
 &= \dots
 \end{aligned}$$

$$\begin{aligned}
 & p(X_2 = x_2, X_4 = f) \\
 &= \sum_{x_1} p(X_1 = x_1) p(X_2 = x_2 \mid X_1 = x_1) \left[\sum_{x_3} p(X_3 = x_3 \mid X_1 = x_1) \right. \\
 &\quad \left. p(X_4 = f \mid X_2 = x_2, X_3 = x_3) \right] \\
 &= \sum_{x_1} p(X_1 = x_1) p(X_2 = x_2 \mid X_1 = x_1) F_1(X_1 = x_1, X_2 = x_2) = F_2(X_2 = x_2)
 \end{aligned}$$

where

		X_3	
X_1		t	f
t		.7	.3
f		.2	.8

		X_4	
X_2	X_3	t	f
t	t	.9	.1
t	f	.7	.3
f	t	.8	.2
f	f	.4	.6

 \mapsto

x_1	x_2	$F_1(X_1, X_2)$
t	t	.7·.1 + .3·.3 = .16
t	f	.7·.2 + .3·.6 = .32
f	t	.2·.1 + .8·.3 = .26
f	f	.2·.2 + .8·.6 = .52

and

	X_1
	t f
	.5 .5

X_1	X_2	
	t f	
t	.7 .3	
f	.1 .9	

x_1	x_2	$F_1(X_1, X_2)$
t	t	.16
\vdots	\vdots	\vdots

 \mapsto

x_2	$F_2(X_2)$
t	...
f	...

- Variable elimination is exponential in maximal number of arguments of factors $p(\dots | \dots)$ resp. $F_j(\dots)$ that appear in the summation process.
- This number depends strongly on the network structure
- ... and can also depend strongly on the order in which we sum out the variables!

Approximate inference using sampling

Problem Structure

Input: Evidence $\mathbf{X} = \mathbf{x}$, random variable Z , value $z \in \text{dom}(Z)$.

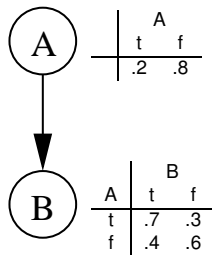
Output: Find approximation q for $p := P(Z = z \mid \mathbf{X} = \mathbf{x})$.

Grand plan

- Somehow sample instantiations from the domain \mathbf{Y} ; $\mathbf{X} \cup Z \subseteq \mathbf{Y}$.
- Somehow use the samples $\{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ to find the approximation q .

Observation: can use Bayesian network as random generator that produces full instantiations $\mathbf{Y} = \mathbf{y}$ according to distribution $P(\mathbf{Y})$.

Example:



- Generate random numbers r_1, r_2 uniformly from $[0,1]$.
- Set $A = t$ if $r_1 \leq .2$ and $A = f$ else.
- Depending on the value of A and r_2 set B to t or f .

Generation of one random instantiation: linear in size of network.

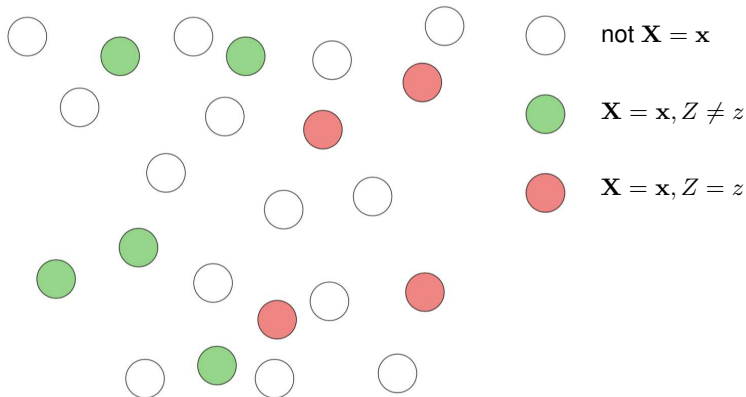
- Given a sample $\mathbf{y}_1, \dots, \mathbf{y}_N$ of complete instantiations generated (independently) by the sampling algorithm, approximate $P(\mathbf{X} = \mathbf{x})$ as

$$q^* := \frac{1}{N} |\{i \in 1, \dots, N \mid \mathbf{X} = \mathbf{x} \text{ in } \mathbf{y}_i\}|$$

- Similarly, the sample provides an estimate for $P(Z = z, \mathbf{X} = \mathbf{x})$.
- Put together, we can estimate

$$P(Z = z \mid \mathbf{X} = \mathbf{x}) = P(Z = z, \mathbf{X} = \mathbf{x}) / P(\mathbf{X} = \mathbf{x}).$$

Forward Sampling: Illustration

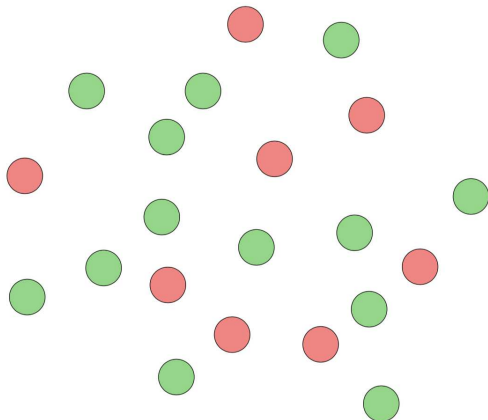


Approximation for $P(Z = z | X = x)$:

$$\frac{\# \text{ (red circle)}}{\# \text{ (green circle)} \cup \# \text{ (red circle)}}$$

Problem of forward sampling: samples with $\mathbf{X} \neq \mathbf{x}$ are useless!

Goal: find algorithm that samples according to $P(\mathbf{Z} \mid \mathbf{X} = \mathbf{z})$:



- **Principle:** obtain new sample from previous sample by randomly changing the value of only one selected variable.
- **Notation:** Let $\mathbf{Y} = (\mathbf{Z}, \mathbf{X})$ denote all variables in the domain, where $\mathbf{X} = \mathbf{x}$ is observed.

Gibbs sampling

$\mathbf{z}_0 :=$ arbitrary instantiation of \mathbf{Z} .

$\mathbf{y}_0 := (\mathbf{z}_0, \mathbf{x})$.

$t := 1$.

repeat forever

 choose $Z_k \in \mathbf{Z}$

 set $y_{t,j} := y_{t-1,j}$ for all Y_j except the chosen Z_k .

 generate randomly $z_{t,k}$ according to $P\left(Z_k \mid \mathbf{Y} \setminus \{Z_k\} = \mathbf{y}_t^{\downarrow \mathbf{Y} \setminus \{Z_k\}}\right)$

 Store the sampled value in Z_k 's location in \mathbf{y}_t .

$t := t + 1$.

$$\begin{aligned}
& P(Z_k = z_k \mid \mathbf{Y} \setminus \{Z_k\} = \mathbf{y}_t^{\downarrow \mathbf{Y} \setminus \{Z_k\}}) \\
& \propto P(Z_k = z_k, \mathbf{Y} \setminus \{Z_k\} = \mathbf{y}_t^{\downarrow \mathbf{Y} \setminus \{Z_k\}}) \\
& \propto P(Z_k = z_k \mid \text{pa}(Z_k) = \mathbf{y}_t^{\downarrow \text{pa}(Z_k)}). \\
& \prod_{i: Y_i \in \text{ch}(Z_k)} P(Y_i = y_{t,i} \mid \text{pa}(Y_i) \setminus \{Z_k\} = \mathbf{y}_t^{\downarrow \text{pa}(Y_i) \setminus \{Z_k\}}, Z_k = z_k) \quad (*)
\end{aligned}$$

where \propto means: equals up to a constant that does not depend on z_k .

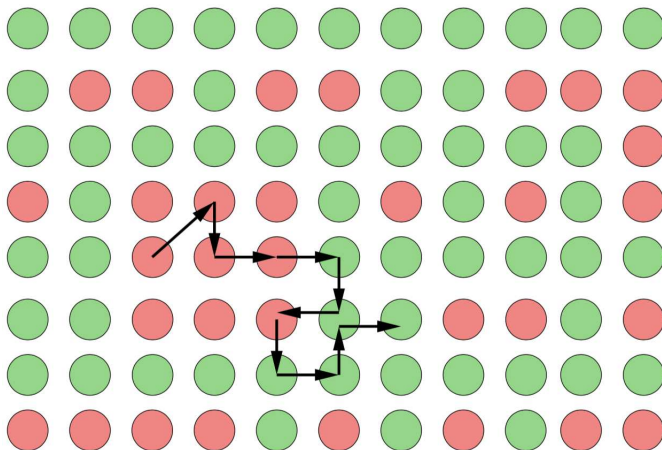
\rightsquigarrow **Note:** To sample a value we only need to consider the **Markov blanket** for Z_k !

To sample $Z_{t,k}$

- Normalize (*) to obtain $P(Z_k \mid \mathbf{Y} \setminus \{Z_k\} = \mathbf{y}_t^{\downarrow \mathbf{Y} \setminus \{Z_k\}})$
 - $P(Z_k = z_k \mid \mathbf{Y} \setminus \{Z_k\} = \mathbf{y}_t^{\downarrow \mathbf{Y} \setminus \{Z_k\}})$ is called the *full conditional* for Z_k .
- Sample value $z_{t,k}$ according to the resulting distribution

Gibbs Random Walk

The process of Gibbs sampling can be understood as a *random walk* in the space of all instantiations with $\mathbf{X} = \mathbf{x}$:



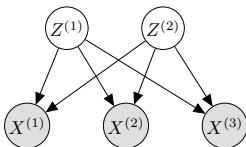
Reachable in one step: instantiations that differ from current one by value assignment to at most one variable (assume randomized choice of variable Z_k).

Code Task: Exact and approximate inference

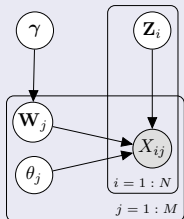
- Something nice.

A more elaborate example: Factor analysis

- *Factor analysis* is a statistical model used to summarize a high-dimensional observation \mathbf{X} of correlated variables by a smaller set \mathbf{Z} of *factors* that a priori are assumed independent.
- **Example:** \mathbf{X} is a set of scores a subject gets from some intelligence-test, \mathbf{Z} models different types of intelligence (e.g., sense of logics, verbal skills, ...).



Mathematical formulation:

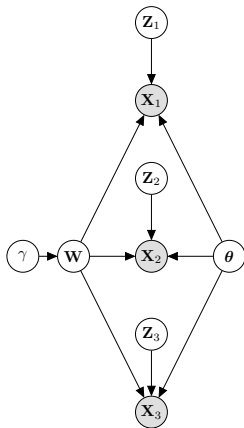


- $\mathbf{Z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- $X_{i,j} \mid \{\mathbf{z}_i, \mathbf{w}_j, \theta_j\} \sim \mathcal{N}(\mathbf{w}_j^\top \mathbf{z}_i, 1/\theta_j)$.
- **Bayesian setting:** Add priors for \mathbf{W}_j 's and θ .

Relevant questions given a dataset $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$:

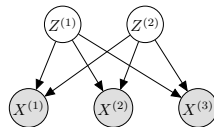
- Learning: $p(\mathbf{w}, \theta, \gamma \mid \mathcal{D})$.
- “Understanding” a new example \mathbf{x}^* : $p(\mathbf{z} \mid \mathbf{X} = \mathbf{x}^*, \mathcal{D})$.
- ...

Unfolded model



FA model “unfolded” for three data instances (X_1, X_2, X_3)

Recall local model



Observations

Inspecting the independence properties of unfolded model we see that the

- number of variables (W and θ) in “separating factor” are manageable.
- posterior cannot be calculated in closed-form because the priors (assumed a priori independent) are not conjugate. (More on this later.)

↪ Approximate inference required.

Full conditional for $p(\mathbf{w}_j \mid \mathbf{w}_{-j}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z}, \gamma)$

Let $\mathbf{w}_{-j}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z}, \gamma$ be a configuration over all variables except \mathbf{w}_j . Then

$$p(\mathbf{w}_j \mid \mathbf{w}_{-j}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z}, \gamma) \propto p(\mathbf{w}_j \mid \gamma) \prod_{i=1}^N p(x_{ij} \mid \mathbf{w}_j, \mathbf{z}_i, \boldsymbol{\theta}_j)$$

With a bit of pencil pushing we find that:

$$p(\mathbf{w}_j \mid \mathbf{w}_{-j}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z}, \gamma) = \mathcal{N}(\mathbf{w}_j \mid \boldsymbol{\mu}, \mathbf{Q}^{-1}),$$

where

- $\mathbf{Q} \leftarrow \gamma \mathbf{I} + \theta_j \sum_{i=1}^N \mathbf{z}_i \mathbf{z}_i^\top$
- $\boldsymbol{\mu} \leftarrow \mathbf{Q}^{-1} \theta_j \sum_{i=1}^N x_{ij} \mathbf{z}_i$

Full conditional for $p(\gamma \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z})$

$$p(\gamma \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z}) \propto p(\gamma) \prod_{j=1}^M p(\mathbf{w}_j \mid \gamma)$$

We find that

$$p(\gamma \mid \mathbf{w}, \boldsymbol{\theta}, \mathbf{x}, \mathbf{z}) = \text{Gamma}(\gamma \mid \text{shape}, \text{rate}),$$

where

- $\text{shape} \leftarrow \text{prior_shape} + \frac{M \cdot D}{2}$
- $\text{rate} \leftarrow \text{prior_rate} + \frac{1}{2} \sum_{j=1}^M \mathbf{w}_j^T \mathbf{w}_j$