# Approximate Inference and MAP Inference

Wilker Aziz w.aziz@uva.nl

Fall 2025

*https://probabll.github.io*

# Outline and goals

Module 4 introduces *a key algorithm for Approximate Inference* known as Gibbs sampling (Chapter 12) as well as a modification to sum-product VE which addresses a different type of inference task known as max-product or MAP inference (Chapter 13).

ILOs After this module the student

- can use Gibbs sampling to perform approximate inference (marginals, conditionals and expected values);
- can use VE for max-product/MAP inference.

Textbook for this course: Koller and Friedman [1].

# Overview of Module 4

HC5a: approximate inference

LC5: Gibbs sampling and max-product VE in code.

HC5b: max-product VE

WC5: exercises.

# Table of contents

1. Sampling-Based Inference

2. Max-Product (or MAP) Inference

# Sampling-Based Inference

# Challenge

Inference tasks typically involve computing

- marginals $P_\Phi(Y) = \sum_M P_\Phi(M, Y)$
- conditionals $P_\Phi(Y|E = e) = \frac{P_\Phi(Y, E=e)}{P_\Phi(E=e)}$
- a combination of both $P_\Phi(Q|E = e) = \frac{\sum_M P_\Phi(Q, M, E=e)}{P_\Phi(E=e)}$
- expectations $\mathbb{E}[f(X)] = \sum_X P_\Phi(X) f(X)$

We have developed marginal/conditional inference via VE, which in many cases can be sufficiently tractable.

What about those cases when even VE is too expensive? Or how about computing the expectation of a complex function of the rvs?

We now develop a rather different approach, based on simulation.

# Sampling-Based Estimation

We have a dataset of ‘outcomes’ $\mathcal{D} = \{x[1], \ldots, x[M]\}$ sampled IID from a distribution $P(X)$.

IID means independently and identically distributed.

Suppose $\text{Val}(X) = \{0, 1\}$. If $P(X = 1) = \theta$, then the ‘sample mean’

$$T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^{M} x[m] \tag{1}$$

is an *unbiased estimator* of $\theta$.

More generally, we can use $\mathcal{D}$ to estimate the expected value of any function $f$ of the rv:

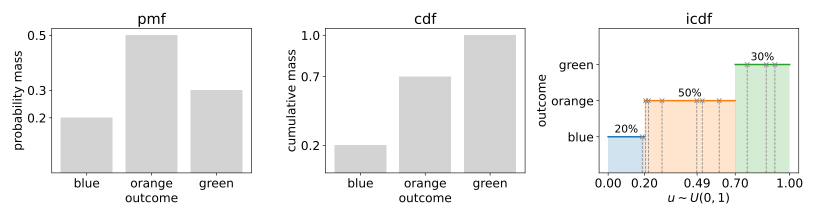
$$\mathbb{E}[f(X)] \approx \frac{1}{M} \sum_{m=1}^{M} f(x[m]) \tag{2}$$

# Sampling for Discrete Distributions

Consider the case where $\mathsf{Val}(X) = \{x^1, \ldots, x^K\}$ and $P(X = x^k) = \theta_k$.

By definition, we know that $0 \leq \theta_k \leq 1$ and we know that $\sum_{k=1}^{K} \theta_k = 1$. Hence, let's order the cumulative sums over $\theta_k$ on the real line.

Example with $K = 3$ and $\boldsymbol{\theta} = (0.2, 0.5, 0.3)^\top$



Let: probability mass function (pmf). Centre: cumulative distribution function (cdf). Right: inverse cdf (icdf).

# ICDF Sampling for Discrete Distributions [Box 12.A]

If $F^{-1}$ is the icdf of the rv $X$ with distribution $P$, then $F^{-1}(U)$ for $U \sim \mathcal{U}(0,1)$ is distributed by $P$.

In other words, transforming samples from a uniform random number generator via the icdf of an rv $X$ gives us a sample from the distribution $P(X)$.

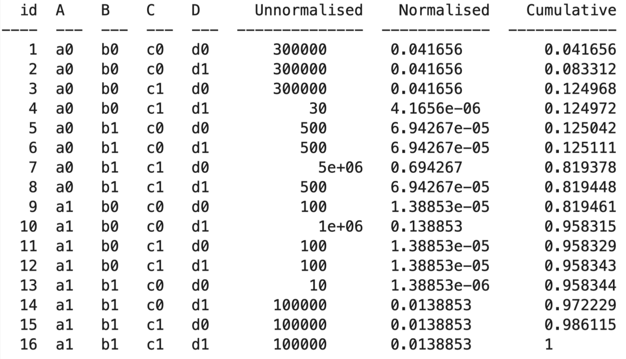
This scales (at worst) linearly with the cardinality of $X$:

- compute cumulative sums $c_k = \sum_{i \leq k} \theta_i$; define $c_0 = 0$.
- draw uniform number $u$ in the interval $[0,1]$;
- find $k$ such that $c_{k-1} < u \leq c_k$, return $x^k$.

# Sampling from PGMs Naively but Exactly via the ICDF Method

A PGM over a collection of discrete variables $X$ represents a discrete distribution $P_\Phi(X)$. Hence, in principle, it is possible to build the ICDF associated with $P_\Phi(X)$, and obtain samples via the ICDF method.

Misconception example:



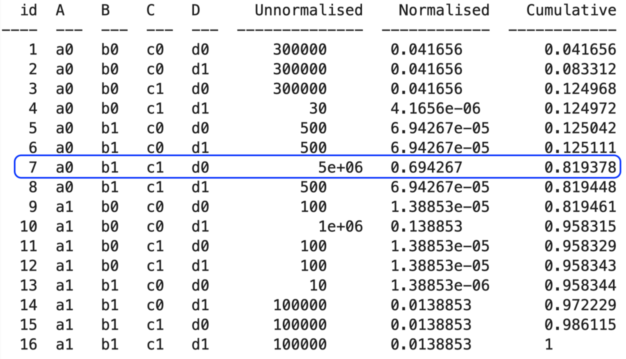
```
  id  A    B    C    D      Unnormalised    Normalised    Cumulative
----  ---  ---  ---  ---  --------------  ------------  ------------
   1  a0   b0   c0   d0       300000       0.041656          0.041656
   2  a0   b0   c0   d1       300000       0.041656          0.083312
   3  a0   b0   c1   d0       300000       0.041656          0.124968
   4  a0   b0   c1   d1           30       4.1656e-06        0.124972
   5  a0   b1   c0   d0          500       6.94267e-05       0.125042
   6  a0   b1   c0   d1          500       6.94267e-05       0.125111
   7  a0   b1   c1   d0            5e+06   0.694267          0.819378
   8  a0   b1   c1   d1          500       6.94267e-05       0.819448
   9  a1   b0   c0   d0          100       1.38853e-05       0.819461
  10  a1   b0   c0   d1            1e+06   0.138853          0.958315
  11  a1   b0   c1   d0          100       1.38853e-05       0.958329
  12  a1   b0   c1   d1          100       1.38853e-05       0.958343
  13  a1   b1   c0   d0           10       1.38853e-06       0.958344
  14  a1   b1   c0   d1       100000       0.0138853         0.972229
  15  a1   b1   c1   d0       100000       0.0138853         0.986115
  16  a1   b1   c1   d1       100000       0.0138853         1
```

What assignment will the ICDF method return if **uniform**$(0,1)$ gives us 0.514?

# Sampling from PGMs Naively but Exactly via the ICDF Method

A PGM over a collection of discrete variables $X$ represents a discrete distribution $P_\Phi(X)$. Hence, in principle, it is possible to build the ICDF associated with $P_\Phi(X)$, and obtain samples via the ICDF method.

Misconception example:



```
  id  A    B    C    D      Unnormalised    Normalised     Cumulative
----  ---  ---  ---  ---  --------------  ------------  ------------
   1  a0   b0   c0   d0         300000      0.041656         0.041656
   2  a0   b0   c0   d1         300000      0.041656         0.083312
   3  a0   b0   c1   d0         300000      0.041656         0.124968
   4  a0   b0   c1   d1             30      4.1656e-06       0.124972
   5  a0   b1   c0   d0            500      6.94267e-05      0.125042
   6  a0   b1   c0   d1            500      6.94267e-05      0.125111
   7  a0   b1   c1   d0              5e+06  0.694267         0.819378
   8  a0   b1   c1   d1            500      6.94267e-05      0.819448
   9  a1   b0   c0   d0            100      1.38853e-05      0.819461
  10  a1   b0   c0   d1              1e+06  0.138853         0.958315
  11  a1   b0   c1   d0            100      1.38853e-05      0.958329
  12  a1   b0   c1   d1            100      1.38853e-05      0.958343
  13  a1   b1   c0   d0             10      1.38853e-06      0.958344
  14  a1   b1   c0   d1         100000      0.0138853        0.972229
  15  a1   b1   c1   d0         100000      0.0138853        0.986115
  16  a1   b1   c1   d1         100000      0.0138853        1
```

What assignment will the ICDF method return if **uniform**$(0,1)$ gives us 0.514?

The assignment shown in the 7th row.

# Limitations

**Named-entity recognition (NER).** In 'Apple 's Jobs died 56 in Palo Alto on October 5 , 2011 .' some words refer to *entities (or abstract concepts)* in the real world (e.g. Apple Inc. is an organisation, Steve Jobs was one of its co-founders, Palo Alto is a place, October 5, 2011 was a specific day).

# Limitations

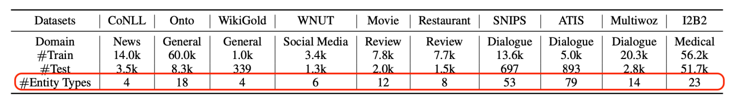
**Named-entity recognition (NER).** In ‘Apple ’s Jobs died 56 in Palo Alto on October 5 , 2011 .’ some words refer to *entities (or abstract concepts)* in the real world (e.g. Apple Inc. is an organisation, Steve Jobs was one of its co-founders, Palo Alto is a place, October 5, 2011 was a specific day).

The typical model to uncover this kind of structure is a special type of MN known as *conditional random field* (CRF). This CRF has an rv per ‘token’ (roughly, non-blank character sequences separated by blank characters) connected in a chain, each rv captures whether the token is part of a ‘named-entity’ of a certain type or not.

# Limitations

**Named-entity recognition (NER).** In 'Apple 's Jobs died 56 in Palo Alto on October 5 , 2011 .' some words refer to *entities (or abstract concepts)* in the real world (e.g. Apple Inc. is an organisation, Steve Jobs was one of its co-founders, Palo Alto is a place, October 5, 2011 was a specific day).

The typical model to uncover this kind of structure is a special type of MN known as *conditional random field* (CRF). This CRF has an rv per 'token' (roughly, non-blank character sequences separated by blank characters) connected in a chain, each rv captures whether the token is part of a 'named-entity' of a certain type or not.



| Datasets | CoNLL | Onto | WikiGold | WNUT | Movie | Restaurant | SNIPS | ATIS | Multiwoz | I2B2 |
|---|---|---|---|---|---|---|---|---|---|---|
| Domain<br>#Train<br>#Test | News<br>14.0k<br>3.5k | General<br>60.0k<br>8.3k | General<br>1.0k<br>339 | Social Media<br>3.4k<br>1.3k | Review<br>7.8k<br>2.0k | Review<br>7.7k<br>1.5k | Dialogue<br>13.6k<br>697 | Dialogue<br>5.0k<br>893 | Dialogue<br>20.3k<br>2.8k | Medical<br>56.2k<br>51.7k |
| #Entity Types | 4 | 18 | 4 | 6 | 12 | 8 | 53 | 79 | 14 | 23 |

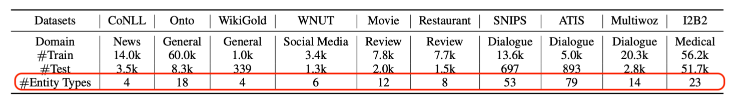
Number of entity types in different datasets.

How large is a tabular view of the joint CRF distribution for a sentence with 20 words in the *Movie* dataset?

# Limitations

**Named-entity recognition (NER).** In ‘Apple ’s Jobs died 56 in Palo Alto on October 5 , 2011 .’ some words refer to *entities (or abstract concepts)* in the real world (e.g. Apple Inc. is an organisation, Steve Jobs was one of its co-founders, Palo Alto is a place, October 5, 2011 was a specific day).

The typical model to uncover this kind of structure is a special type of MN known as *conditional random field* (CRF). This CRF has an rv per ‘token’ (roughly, non-blank character sequences separated by blank characters) connected in a chain, each rv captures whether the token is part of a ‘named-entity’ of a certain type or not.



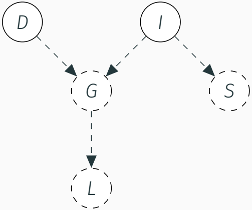
| Datasets | CoNLL | Onto | WikiGold | WNUT | Movie | Restaurant | SNIPS | ATIS | Multiwoz | I2B2 |
|---|---|---|---|---|---|---|---|---|---|---|
| Domain | News | General | General | Social Media | Review | Review | Dialogue | Dialogue | Dialogue | Medical |
| #Train | 14.0k | 60.0k | 1.0k | 3.4k | 7.8k | 7.7k | 13.6k | 5.0k | 20.3k | 56.2k |
| #Test | 3.5k | 8.3k | 339 | 1.3k | 2.0k | 1.5k | 697 | 893 | 2.8k | 51.7k |
| #Entity Types | 4 | 18 | 4 | 6 | 12 | 8 | 53 | 79 | 14 | 23 |

Number of entity types in different datasets.

How large is a tabular view of the joint CRF distribution for a sentence with 20 words in the *Movie* dataset? $\propto 12^{20}$ — for perspective, the number of microseconds in a year is in the order of $10^{13}$.

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

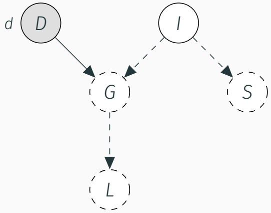
A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]



| | |
|---|---|
| Solid | *can* be sampled. |
| Dashed | **cannot** be sampled. |

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]

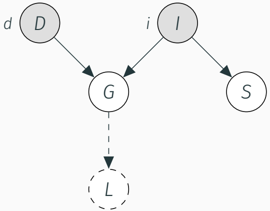


| | |
|---|---|
| Solid | *can* be sampled. |
| Dashed | **cannot** be sampled. |

1. with probability $P(D = d)$ draw $d \in \mathrm{Val}(D)$;

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]

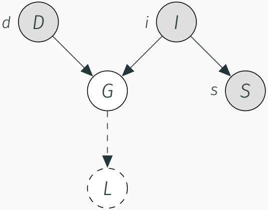


| | |
|---|---|
| Solid | *can* be sampled. |
| Dashed | **cannot** be sampled. |

1. with probability $P(D = d)$ draw $d \in \mathsf{Val}(D)$;
2. with probability $P(I = i)$ draw $i \in \mathsf{Val}(I)$;

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]

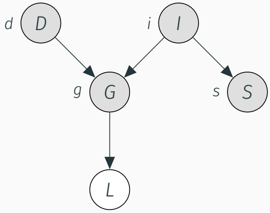


Solid *can* be sampled.
Dashed **cannot** be sampled.

1. with probability $P(D = d)$ draw $d \in \mathsf{Val}(D)$;
2. with probability $P(I = i)$ draw $i \in \mathsf{Val}(I)$;
3. with probability $P(S = s|I = i)$ draw $s \in \mathsf{Val}(S)$;

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]

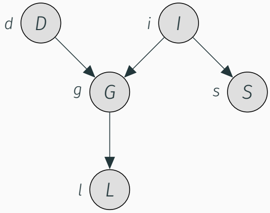


Solid *can* be sampled.
Dashed **cannot** be sampled.

1. with probability $P(D = d)$ draw $d \in \mathrm{Val}(D)$;
2. with probability $P(I = i)$ draw $i \in \mathrm{Val}(I)$;
3. with probability $P(S = s|I = i)$ draw $s \in \mathrm{Val}(S)$;
4. with probability $P(G = g|D = d, I = i)$ draw $g \in \mathrm{Val}(G)$;

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]

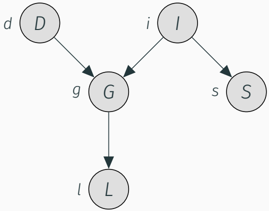


| | |
|---|---|
| Solid | *can* be sampled. |
| Dashed | **cannot** be sampled. |

1. with probability $P(D = d)$ draw $d \in \mathsf{Val}(D)$;
2. with probability $P(I = i)$ draw $i \in \mathsf{Val}(I)$;
3. with probability $P(S = s|I = i)$ draw $s \in \mathsf{Val}(S)$;
4. with probability $P(G = g|D = d, I = i)$ draw $g \in \mathsf{Val}(G)$;
5. with probability $P(L = l|G = g)$ draw $l \in \mathsf{Val}(L)$;

# Exploiting the BN Structure: Forward Sampling [Section 12.1]

A BN is parameterised by a hierarchically organised collection of CPDs. We can traverse the BN in a topological order sampling from each CPD (e.g., via the ICDF method). As we sample a node, its assignment becomes available to condition the sampling of its directed descendants. [Algorithm 12.1]

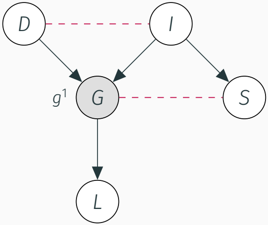


Solid *can* be sampled.
Dashed **cannot** be sampled.

1. with probability $P(D = d)$ draw $d \in \mathsf{Val}(D)$;
2. with probability $P(I = i)$ draw $i \in \mathsf{Val}(I)$;
3. with probability $P(S = s|I = i)$ draw $s \in \mathsf{Val}(S)$;
4. with probability $P(G = g|D = d, I = i)$ draw $g \in \mathsf{Val}(G)$;
5. with probability $P(L = l|G = g)$ draw $l \in \mathsf{Val}(L)$;

The product of probabilities in steps (1)–(5) is precisely the BN probability for the outcome $(d, i, s, g, l) \in \mathsf{Val}(D, I, S, G, L)$.

# Forward Sampling – Limitations



Cannot handle evidence (since conditioning tends to create undirected dependencies).

No equivalent for MNs (dependencies are undirected by design).

Some special variants of VE can be designed to support something like forward sampling for certain BNs/MNs, but a general solution calls for a different approach altogether.

# Markov Chain Monte Carlo (MCMC) [Section 12.3]

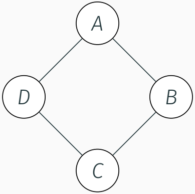
MCMC methods construct a *sequence* of samples.

This sequence is constructed so that, although the first sample may be generated from an arbitrary distribution, successive samples are generated from distributions that get closer and closer to the desired distribution.

MCMC methods apply equally well to directed and undirected models with and without evidence.

# Gibbs Sampling (Intuition) 1/5

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



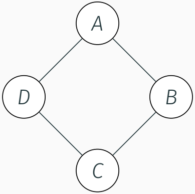
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $A$ given $(B = b, C = c, D = d)$.

1. express the distribution $P_{\Phi}(A|B = b, C = c, D = d) \propto$

# Gibbs Sampling (Intuition) 1/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



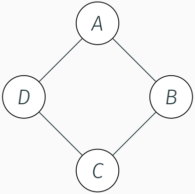
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $A$ given $(B = b, C = c, D = d)$.

1. express the distribution $P_{\Phi}(A|B = b, C = c, D = d) \propto \phi_1[b](A)\phi_4[d](A)$;

# Gibbs Sampling (Intuition) 1/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

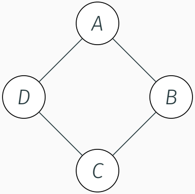
Now, let's resample $A$ given $(B = b, C = c, D = d)$.

1. express the distribution $P_{\Phi}(A|B = b, C = c, D = d) \propto \phi_1[b](A)\phi_4[d](A)$;
2. sample $a'$ using the ICDF method.

Intermediate assignment: $(A = a', B = b, C = c, D = d)$

# Gibbs Sampling (Intuition) 2/5

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



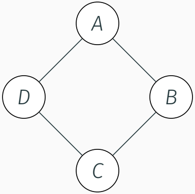
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $B$ given $(A = a', C = c, D = d)$.

1. express the distribution $P_{\mathbf{\Phi}}(B|A = a', C = c, D = d) \propto$

# Gibbs Sampling (Intuition) 2/5

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



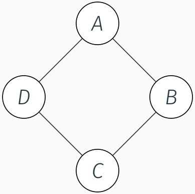
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $B$ given $(A = a', C = c, D = d)$.

1. express the distribution $P_{\mathbf{\Phi}}(B|A = a', C = c, D = d) \propto \phi_1[a'](B)\phi_2[c](B)$;

# Gibbs Sampling (Intuition) 2/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

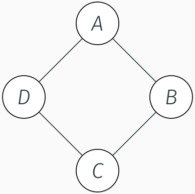
Now, let's resample $B$ given $(A = a', C = c, D = d)$.

1. express the distribution $P_{\Phi}(B|A = a', C = c, D = d) \propto \phi_1[a'](B)\phi_2[c](B)$;
2. sample $b'$ using the ICDF method.

Intermediate assignment: $(A = a', B = b', C = c, D = d)$

# Gibbs Sampling (Intuition) 3/5

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



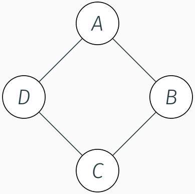
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $C$ given $(A = a', B = b', D = d)$.

1. express the distribution $P_{\Phi}(C|A = a', B = b', D = d) \propto$

# Gibbs Sampling (Intuition) 3/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



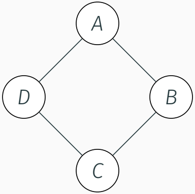
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $C$ given $(A = a', B = b', D = d)$.

1. express the distribution $P_{\Phi}(C|A = a', B = b', D = d) \propto \phi_2[b'](C)\phi_3[d](C)$;

# Gibbs Sampling (Intuition) 3/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

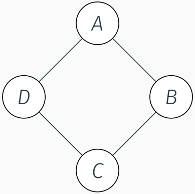
Now, let's resample $C$ given $(A = a', B = b', D = d)$.

1. express the distribution $P_{\Phi}(C|A = a', B = b', D = d) \propto \phi_2[b'](C)\phi_3[d](C)$;
2. sample $c'$ using the ICDF method.

Intermediate assignment: $(A = a', B = b', C = c', D = d)$

# Gibbs Sampling (Intuition) 4/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



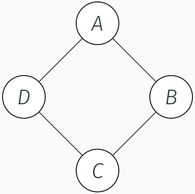
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $D$ given $(A = a', B = b', C = c')$.

1. express the distribution $P_{\Phi}(D|A = a', B = b', C = c') \propto$

# Gibbs Sampling (Intuition) 4/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



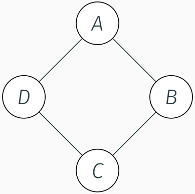
| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $D$ given $(A = a', B = b', C = c')$.

1. express the distribution $P_{\Phi}(D|A = a', B = b', C = c') \propto \phi_3[c'](D)\phi_4[a'](D)$;

# Gibbs Sampling (Intuition) 4/5

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.



| $\Phi$ | Scope |
|---|---|
| $\phi_1$ | A, B |
| $\phi_2$ | B, C |
| $\phi_3$ | C, D |
| $\phi_4$ | D, A |

Now, let's resample $D$ given $(A = a', B = b', C = c')$.

1. express the distribution $P_{\mathbf{\Phi}}(D|A = a', B = b', C = c') \propto \phi_3[c'](D)\phi_4[a'](D)$;
2. sample $d'$ using the ICDF method.

Final assignment: $(A = a', B = b', C = c', D = d')$

# Gibbs Sampling (Intuition) 5/5 [Section 12.3.1]

Let's start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.

| | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| **Origin** $(x)$ | $a$ | $b$ | $c$ | $d$ |
| | $a'$ | $b$ | $c$ | $d$ |
| | $a'$ | $b'$ | $c$ | $d$ |
| | $a'$ | $b'$ | $c'$ | $d$ |
| | $a'$ | $b'$ | $c'$ | $d'$ |
| **Destination** $(x')$ | $a'$ | $b'$ | $c'$ | $d'$ |

The first $(x)$ and last $(x')$ assignments are called **states**. The intermediate assignments are part of the processing of **transitioning** from the origin state to the destination state.

Transitioning from state $x$ to $x'$ involves resampling each node in turn conditioned on the node's Markov blanket.

# Gibbs Sampling (Intuition) 5/5 [Section 12.3.1]

Let’s start from an assignment of the Misconception example. For example: $(A = a, B = b, C = c, D = d)$.

| | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| **Origin** $(x)$ | $a$ | $b$ | $c$ | $d$ |
| | $a'$ | $b$ | $c$ | $d$ |
| | $a'$ | $b'$ | $c$ | $d$ |
| | $a'$ | $b'$ | $c'$ | $d$ |
| | $a'$ | $b'$ | $c'$ | $d'$ |
| **Destination** $(x')$ | $a'$ | $b'$ | $c'$ | $d'$ |

The first $(x)$ and last $(x')$ assignments are called **states**. The intermediate assignments are part of the processing of **transitioning** from the origin state to the destination state.

Transitioning from state $x$ to $x'$ involves resampling each node in turn conditioned on the node’s Markov blanket.

We sample *locally* like we did in forward sampling. But this is only possible because we had a state $x$ to condition on. The distribution of the destination state is not $P_{\Phi}$ but it is related to it in a way we shall discuss.

# Markov Chain 1/2 [Section 12.3.2]

A Markov chain defines a transition model $\mathcal{T}$ in a state space.

We can draw *trajectories* through this space by ‘jumping’ probabilistically from state to state.

# Markov Chain 1/2 [Section 12.3.2]

A Markov chain defines a transition model $\mathcal{T}$ in a state space.

We can draw *trajectories* through this space by ‘jumping’ probabilistically from state to state.

In our example/intuition using the Misconception example

- the state space was the set of all joint assignments of our rvs;

# Markov Chain 1/2 [Section 12.3.2]

A Markov chain defines a transition model $\mathcal{T}$ in a state space.

We can draw *trajectories* through this space by 'jumping' probabilistically from state to state.

In our example/intuition using the Misconception example

- the state space was the set of all joint assignments of our rvs;
- we jumped from an assignment $(A = a, B = b, C = c, D = d)$, the origin state $x$, to an assignment $(A = a', B = b', C = c', D = d')$, the destination state $x'$, by resampling each rv in turn while holding the others fixed at their (intermediate) values;

# Markov Chain 1/2 [Section 12.3.2]

A Markov chain defines a transition model $\mathcal{T}$ in a state space.

We can draw *trajectories* through this space by ‘jumping’ probabilistically from state to state.

In our example/intuition using the Misconception example

- the state space was the set of all joint assignments of our rvs;
- we jumped from an assignment $(A = a, B = b, C = c, D = d)$, the origin state $x$, to an assignment $(A = a', B = b', C = c', D = d')$, the destination state $x'$, by resampling each rv in turn while holding the others fixed at their (intermediate) values;
- in effect, we jumped from $x$ to $x'$ with probability proportional to

$$P_{\Phi}(A = a'|B = b, C = c, D = d) \times P_{\Phi}(B = b'|A = a', C = c, D = d)$$
$$\times P_{\Phi}(C = c'|A = a', B = b', D = d) \times P_{\Phi}(D = d'|A = a', B = b', C = c')$$

# Markov Chain 2/2 [Section 12.3.2]

From a state $x$ we move to a state $x'$ with probability $\mathcal{T}(x \rightarrow x')$.

The **transition model is probabilistic** in the sense that $\sum_{x'} \mathcal{T}(x \rightarrow x') = 1$, and this holds no matter the origin state $x$.

Grasshopper example (Fig. 12.3 of textbook):

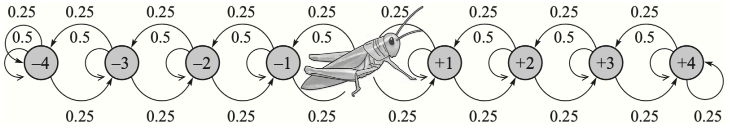


**Figure 12.3 The** Grasshopper **Markov chain**

The ‘state’ is the position of a grasshopper on a finite and discretised line segment; the transition probabilities are shown in the figure.

# Temporal Dynamics 1/2

The Markov chain specifies a random sampling process that defines a random sequence of states $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$

As the transition model is random, the state of the process at step $t$ can be viewed as a random variable $X^{(t)}$: [Algorithm 12.5]

- draw an initial state from some known distribution $P^{(0)}(X^{(0)})$;
- from there on, for any step $t$ at state $x^{(t)}$, draw the next state $x^{(t+1)}$ from the transition model with probability $\mathcal{T}(x^{(t)} \to x^{(t+1)})$.

By construction then, the probability that the chain is at some state $x'$ at step $t+1$ is:

$$P^{(t+1)}(X^{(t+1)} = x') = \sum_{x} P^{(t)}(X^{(t)} = x)\mathcal{T}(x \to x')$$

# Temporal Dynamics 2/2

The probability that the chain is at some state $x'$ at step $t+1$ is:

$$P^{(t+1)}(X^{(t+1)} = x') = \sum_{x} P^{(t)}(X^{(t)} = x)\mathcal{T}(x \to x')$$

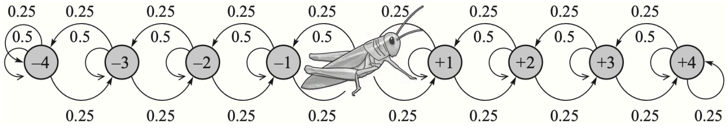


**Figure 12.3** **The** Grasshopper **Markov chain**

| | -2 | -1 | 0 | 1 | 2 |
|---|---|---|---|---|---|
| $P^{(0)}$ | 0 | 0 | 1 | 0 | 0 |
| $P^{(1)}$ | 0 | .25 | .5 | .25 | 0 |
| $P^{(2)}$ | $.25^2$<br>$= .0625$ | $2(.25 \times .5)$<br>$= .25$ | $.5^2 + 2(.25 \times .5)$<br>$= .375$ | $2(.25 \times .5)$<br>$= .25$ | $.25^2$<br>$= .0625$ |

# Stationary Distribution 1/2 [Section 12.3.2.3]

Over time, $P^{(t+1)}(X^{(t+1)} = x') \approx P^{(t)}(X^{(t)} = x') \approx \pi(X = x')$. That is, the process **converges** to the so-called stationary distribution $\pi(X)$.

This distribution is what we call an *equilibrium* relative to the transition model.

Intuitively this equilibrium means that *the probability of being in a state is the same as the probability of transitioning into it from a randomly sampled predecessor.*

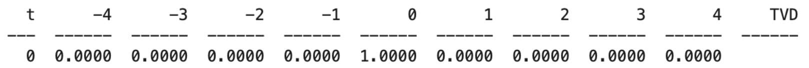
Formally, $\pi(X)$ is a stationary distribution for a Markov chain $T$ if it satisfies:

$$\pi(X = x') = \sum_{x} \pi(X = x)\mathcal{T}(x \to x') \tag{3}$$

Another name for $\pi(X)$ is the *invariant distribution.*

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

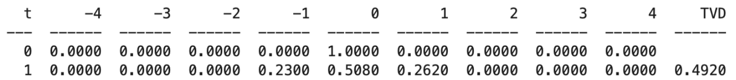


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
```

$\mathrm{TVD}(p,q) = {}^{1}\!/_{2} \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

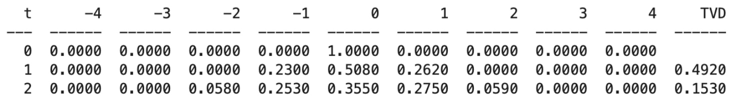


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
```

$\mathrm{TVD}(p, q) = 1/2 \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

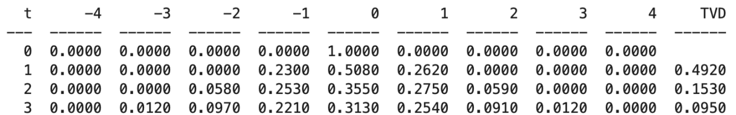


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
  2  0.0000  0.0000  0.0580  0.2530  0.3550  0.2750  0.0590  0.0000  0.0000  0.1530
```

$\mathrm{TVD}(p,q) = {}^{1}\!/_{2} \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

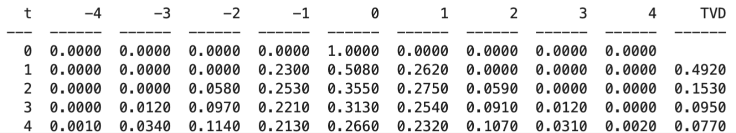


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
  2  0.0000  0.0000  0.0580  0.2530  0.3550  0.2750  0.0590  0.0000  0.0000  0.1530
  3  0.0000  0.0120  0.0970  0.2210  0.3130  0.2540  0.0910  0.0120  0.0000  0.0950
```

$\mathrm{TVD}(p, q) = {}^1\!/_2 \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

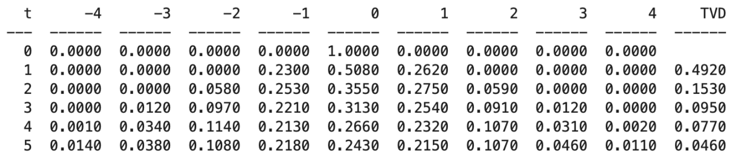


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
  2  0.0000  0.0000  0.0580  0.2530  0.3550  0.2750  0.0590  0.0000  0.0000  0.1530
  3  0.0000  0.0120  0.0970  0.2210  0.3130  0.2540  0.0910  0.0120  0.0000  0.0950
  4  0.0010  0.0340  0.1140  0.2130  0.2660  0.2320  0.1070  0.0310  0.0020  0.0770
```

$\mathrm{TVD}(p, q) = {}^1\!/_2 \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

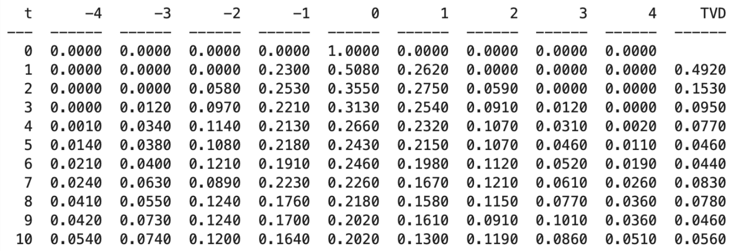


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
  2  0.0000  0.0000  0.0580  0.2530  0.3550  0.2750  0.0590  0.0000  0.0000  0.1530
  3  0.0000  0.0120  0.0970  0.2210  0.3130  0.2540  0.0910  0.0120  0.0000  0.0950
  4  0.0010  0.0340  0.1140  0.2130  0.2660  0.2320  0.1070  0.0310  0.0020  0.0770
  5  0.0140  0.0380  0.1080  0.2180  0.2430  0.2150  0.1070  0.0460  0.0110  0.0460
```

$\mathrm{TVD}(p, q) = 1/2 \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).

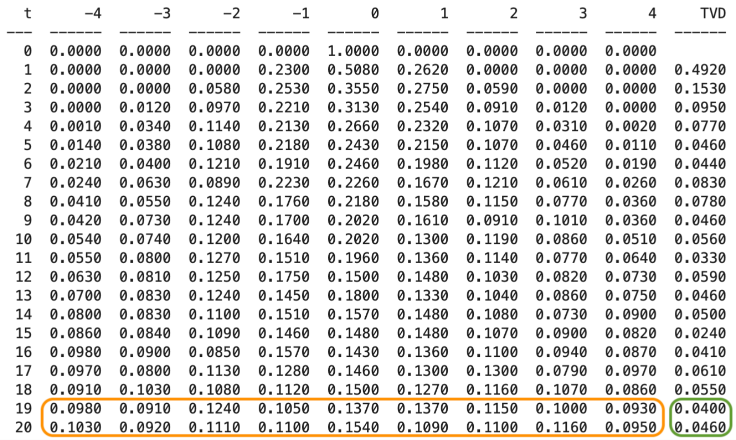


```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
  2  0.0000  0.0000  0.0580  0.2530  0.3550  0.2750  0.0590  0.0000  0.0000  0.1530
  3  0.0000  0.0120  0.0970  0.2210  0.3130  0.2540  0.0910  0.0120  0.0000  0.0950
  4  0.0010  0.0340  0.1140  0.2130  0.2660  0.2320  0.1070  0.0310  0.0020  0.0770
  5  0.0140  0.0380  0.1080  0.2180  0.2430  0.2150  0.1070  0.0460  0.0110  0.0460
  6  0.0210  0.0400  0.1210  0.1910  0.2460  0.1980  0.1120  0.0520  0.0190  0.0440
  7  0.0240  0.0630  0.0890  0.2230  0.2260  0.1670  0.1210  0.0610  0.0260  0.0830
  8  0.0410  0.0550  0.1240  0.1760  0.2180  0.1580  0.1150  0.0770  0.0360  0.0780
  9  0.0420  0.0730  0.1240  0.1700  0.2020  0.1610  0.0910  0.1010  0.0360  0.0460
 10  0.0540  0.0740  0.1200  0.1640  0.2020  0.1300  0.1190  0.0860  0.0510  0.0560
```

$\mathrm{TVD}(p, q) = 1/2 \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Stationary Distribution 2/2 [Section 12.3.2.3]

Grasshopper Markov chain: stationary distribution simulated using 1000 sampled trajectories (code on GitHub).



```
  t      -4      -3      -2      -1       0       1       2       3       4     TVD
---  ------  ------  ------  ------  ------  ------  ------  ------  ------  ------
  0  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
  1  0.0000  0.0000  0.0000  0.2300  0.5080  0.2620  0.0000  0.0000  0.0000  0.4920
  2  0.0000  0.0000  0.0580  0.2530  0.3550  0.2750  0.0590  0.0000  0.0000  0.1530
  3  0.0000  0.0120  0.0970  0.2210  0.3130  0.2540  0.0910  0.0120  0.0000  0.0950
  4  0.0010  0.0340  0.1140  0.2130  0.2660  0.2320  0.1070  0.0310  0.0020  0.0770
  5  0.0140  0.0380  0.1080  0.2180  0.2430  0.2150  0.1070  0.0460  0.0110  0.0460
  6  0.0210  0.0400  0.1210  0.1910  0.2460  0.1980  0.1120  0.0520  0.0190  0.0440
  7  0.0240  0.0630  0.0890  0.2230  0.2260  0.1670  0.1210  0.0610  0.0260  0.0830
  8  0.0410  0.0550  0.1240  0.1760  0.2180  0.1580  0.1150  0.0770  0.0360  0.0780
  9  0.0420  0.0730  0.1240  0.1700  0.2020  0.1610  0.0910  0.1010  0.0360  0.0460
 10  0.0540  0.0740  0.1200  0.1640  0.2020  0.1300  0.1190  0.0860  0.0510  0.0560
 11  0.0550  0.0800  0.1270  0.1510  0.1960  0.1360  0.1140  0.0770  0.0640  0.0330
 12  0.0630  0.0810  0.1250  0.1750  0.1500  0.1480  0.1030  0.0820  0.0730  0.0590
 13  0.0700  0.0830  0.1240  0.1450  0.1800  0.1330  0.1040  0.0860  0.0750  0.0460
 14  0.0800  0.0830  0.1100  0.1510  0.1570  0.1480  0.1080  0.0730  0.0900  0.0500
 15  0.0860  0.0840  0.1090  0.1460  0.1480  0.1480  0.1070  0.0900  0.0820  0.0240
 16  0.0980  0.0900  0.0850  0.1570  0.1430  0.1360  0.1100  0.0940  0.0870  0.0410
 17  0.0970  0.0800  0.1130  0.1280  0.1460  0.1300  0.1300  0.0790  0.0970  0.0610
 18  0.0910  0.1030  0.1080  0.1120  0.1500  0.1270  0.1160  0.1070  0.0860  0.0550
 19  0.0980  0.0910  0.1240  0.1050  0.1370  0.1370  0.1150  0.1000  0.0930  0.0400
 20  0.1030  0.0920  0.1110  0.1100  0.1540  0.1090  0.1100  0.1160  0.0950  0.0460
```

$\mathrm{TVD}(p, q) = {}^{1}\!/\!{}_{2} \sum_x |p(x) - q(x)|$ is a notion of distance between two distributions.

# Which Markov Chains do Eventually Converge?

A Markov chain $\mathcal{T}$ may have 1 stationary distribution, none, or many.

A **regular Markov chain** is one such that:

- there exists some finite number of steps $k$ such that for every $x, x' \in \mathrm{Val}(X)$, the probability of getting from $x$ to $x'$ in exactly $k$ steps is $> 0$.

Regular Markov chains have unique stationary distributions (regularity is sufficient but not necessary).

Two simple conditions that are *sufficient* for regularity:

- it is possible to get from any sate to any state using a positive probability path in the state graph;
- for each state $x$, there is a positive probability of transitioning from $x$ to $x$ in one step (self-loop).

# Gibbs Sampler for PGMs

For some distribution $P_\Phi(\boldsymbol{X})$ factorised by a set of factors $\Phi$, the Gibbs sampler is based on a Markov chain prescribed by a product of 'local transition models', each defined as

$$\mathcal{T}_i((\boldsymbol{x}_{-i}, x_i) \to (\boldsymbol{x}_{-i}, x_i')) = P_\Phi(X_i = x_i' | \boldsymbol{X}_{-i} = \boldsymbol{x}_{-i})$$

For any $X_i$, $\mathrm{MB}(X_i) \subseteq \boldsymbol{X}_i$, hence when we condition on $\boldsymbol{X}_{-i} = \boldsymbol{x}_{-i}$, we condition on the Markov blanket of $X_i$, fully separating it from the other rvs, which typically makes sampling from $P_\Phi(X_i | \boldsymbol{X}_{-i} = \boldsymbol{x}_{-i})$ tractable via the ICDF method.

Intuitively, one rv at a time, we forget its value and resample it given the current assignment of the other rvs. The order we resample the variables is not important, so long as we resample them all.

The stationary distribution of this chain is $P_\Phi(\boldsymbol{X})$.

The set of rvs $\boldsymbol{X}_{-i}$ is $\boldsymbol{X} \setminus \{X_i\}$ similarly $\boldsymbol{x}_i$ denotes the outcomes of all RVs but $X_i$.

# Gibbs Sampler with Evidence

For some distribution $P_\Phi(X)$ factorised by a set of factors $\Phi$, and evidence $\boldsymbol{E} = \boldsymbol{e}$, we first reduce the factors using $\boldsymbol{E} = \boldsymbol{e}$ and proceed as with standard Gibbs.

Use $Y = X \setminus E$ to denote the unassigned rvs.

The stationary distribution of the chain is $P_\Phi(Y|E = e)$.

# Is the Gibbs Sampler Regular?

If all factors in $\Phi$ are *strictly positive*, then the Gibbs-sampling Markov chain is regular for $P_{\Phi}$.

So, if we expect Gibbs sampling to be necessary for our model, we should be careful to parameterise the model with strictly positive factors (this will be a practical consideration when we discuss *learning* PGMs).

# Mixing [Section 12.3.5.1]

How long until the Markov chain has converged?

# Mixing [Section 12.3.5.1]

### How long until the Markov chain has converged?

The time $T$ at which $P^{(T)}$ is finally close enough to the stationary distribution, expressed in relation to a convenient notion of distance between distributions (e.g., TVD or KL), is the so-called mixing time.

# Mixing [Section 12.3.5.1]

How long until the Markov chain has converged?

The time $T$ at which $P^{(T)}$ is finally close enough to the stationary distribution, expressed in relation to a convenient notion of distance between distributions (e.g., TVD or KL), is the so-called **mixing time**.

**It is only after this time that we begin to collect samples from the stationary distribution**.

Intuitively: $T$ is the number of steps it takes for the chain to finally overcome the consequences of our starting from some arbitrary choice of $P^{(0)}$.

# Burn-in Time

In practice, **we often discard ('burn') the first $T$ steps of a sampled trajectory**, since at early steps the Markov chain is unlikely to have converged to the stationary distribution.

Ideally, the burn-in time would coincide with the *mixing time*.

But, unfortunately, there is no way to deduce the mixing time, and, in practice, mixing times can be unfeasibly long.

# MCMC in Practice 1/3

Hyperparameters

- Burn-in time?
- How many samples after burn-in?
- Should we take all samples after burn-in (after all, they are correlated, not IID)?
- In what order should we resample the rvs in a joint assignment?
- Where should we get $\boldsymbol{x}^{(0)}$ from?

These are important questions for which we only have heuristic answers.

# MCMC in Practice 2/3

Burn-in time?

As large as we can afford; but we can approximately test for mixing.

# MCMC in Practice 2/3

### Burn-in time?

As large as we can afford; but we can approximately test for mixing.

### How many samples after burn-in?

As many as we can afford. Typically, we choose this hyperparameter on the basis of an estimate performance for the task of interest (e.g., using cross-validation or some heldout dataset).

# MCMC in Practice 2/3

Burn-in time?

As large as we can afford; but we can approximately test for mixing.

How many samples after burn-in?

As many as we can afford. Typically, we choose this hyperparameter on the basis of an estimate performance for the task of interest (e.g., using cross-validation or some heldout dataset).

Should we take all samples after burn-in (after all, they are correlated, not IID)?

Typically, yes. Only if we have a strong reason to prefer using a subset of the samples (e.g., the downstream task scales poorly as a function of sample size), should we 'thin' the chain (collect every $k$th sample).

# MCMC in Practice 3/3

In what order should we resample the rvs in a joint assignment?

Preferably a random order. We should resample each and every rv (given its MB), but, if the MN is really large and there are too many rvs, it is possible to resample only 1 per step, so long as it is chosen uniformly at random. This has the effect of slowying down mixing, and should be avoided if possible.

# MCMC in Practice 3/3

**In what order should we resample the rvs in a joint assignment?**

Preferably a random order. We should resample each and every rv (given its MB), but, if the MN is really large and there are too many rvs, it is possible to resample only 1 per step, so long as it is chosen uniformly at random. This has the effect of slowying down mixing, and should be avoided if possible.

**Where should we get $x^{(0)}$ from?**

In theory, anywhere will do. But preferably $x^{(0)}$ should be close to a realistic sample (e.g., a data sample, or a good heuristic).

# Diagnosing *lack of* Convergence/Mixing

In general, we cannot ascertain convergence to the stationary distribution.

That's because if $P^{(t)} \approx P^{(t+1)}$, it is possible that

- this is the stationary distribution ✓
- the sampler is stuck in a region that's hard to escape from, so consecutive $P^{(t)}, P^{(t+1)}$ look similar ✗

# Diagnosing *lack of* Convergence/Mixing

In general, we cannot ascertain convergence to the stationary distribution.

That's because if $P^{(t)} \approx P^{(t+1)}$, it is possible that

- this is the stationary distribution ✓
- the sampler is stuck in a region that's hard to escape from, so consecutive $P^{(t)}, P^{(t+1)}$ look similar ✗

But we may be able to detect **lack of convergence**,

- since before convergence necessarily $P^{(t)} \not\approx P^{(t+1)}$

Think of this as something like empirically testing a theory: we can falsify the theory (observe a situation that contradicts the theory), but observing that it has not yet been contradicted does not amount to a proof [2].

# Diagnosing *lack of* Convergence/Mixing

In general, we cannot ascertain convergence to the stationary distribution.

That's because if $P^{(t)} \approx P^{(t+1)}$, it is possible that

- this is the stationary distribution ✓
- the sampler is stuck in a region that's hard to escape from, so consecutive $P^{(t)}, P^{(t+1)}$ look similar ✗

But we may be able to detect **lack of convergence**,

- since before convergence necessarily $P^{(t)} \not\approx P^{(t+1)}$

Think of this as something like empirically testing a theory: we can falsify the theory (observe a situation that contradicts the theory), but observing that it has not yet been contradicted does not amount to a proof [2].

# Disagreement between Independent Chains [Section 12.3.5.2]

The most common strategy is to simulate independent chains, each starting from a different initial state.

1. Extend the chains by $C$ steps each.
2. Then, look for evidence of *lack of* convergence.
   - If we find, go back to (1) and simulate $C$ extra steps for each chain.
   - Else, bookmark the current step as the *burn-in* time, and go to (3).
3. Extend the chains by $M$ steps each, for whatever sample-size $M$ we are interested in.

The last $M$ samples of each chain are **assumed** to come from the stationary distribution. Not because we have proof of convergence, but because we could not convince ourselves of the contrary.

A typical strategy for (2) is then to quantify disagreement between independent chains.

# $\hat{R}$ (pronounced ‘r-hat’) [Box 12.B]

Suppose $\mathbf{X}_k$ is the sequence of $M$ samples in the $k$th chain (out of $K$ chains).

Suppose we compute a *statistic* $f(\mathbf{X}_k[m])$ of each sample. For example, a common statistic is the logarithm of the sample’s unnormalised probability under the model.

A measure of disagreement is based on the $B$ (between-chains) and $W$ (within-chain) variances:

$$\bar{f}_k = \frac{1}{M}\sum_{m=1}^{M} f(\mathbf{X}_k[m]) \qquad \bar{f} = \frac{1}{K}\sum_{k=1}^{K}\bar{f}_k$$

$$B = \frac{M}{K-1}\sum_{k=1}^{K}(\bar{f}_k - \bar{f})^2 \qquad W = \frac{1}{K}\frac{1}{M-1}\sum_{k=1}^{K}\sum_{m=1}^{M}(f(\mathbf{X}_k[m] - \bar{f}_k)^2)$$

For $V = \frac{M-1}{M}W + \frac{1}{M}B$, $\hat{R} = \sqrt{V/W}$. If the chain have not all converged, this estimate will be high. If $\hat{R}$ is close to 1, either they have all converged, or the starting points were not sufficiently dispersed (roughly, the sampler is stuck in a region that’s difficult to escape).

# Summary

- Sampling is an alternative to exact inference, whereby we *estimate* key quantities using ‘samples’ (aka ‘particles’).
- It is difficult to sample ‘exactly’ from a PGM, because exact samplers typically require a tabular view of the PGM’s distribution.
- The hierarchical organisation of CPDs in BNs enable a tractable form of sampling, so long as we do not have evidence.
- In general, we need MCMC methods built on Markov chains that converge to the desired distribution.
- Gibbs sampling, a form of MCMC, is scalable because it resamples one rv at a time, given its Markov blanket.
- It is difficult to decided when a Markov chain has converged, we use heuristics that at best detect lack of convergence.
- At any rate, MCMC methods are highly scalable and allow us to approximate inference for most PGMs, however large and complex they may be.

# What Next?

LC5: Gibbs sampling in code.

HC5b: MAP inference.

WC5: exercises.

# Max-Product (or MAP) Inference

# To Be Continued...

# References

[1] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[2] K. R. Popper. *The Logic of Scientific Discovery*. Hutchinson, London, 1934.