10-708 Probabilistic Graphical Models

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Markov Properties
+
Factor Graphs

Matt Gormley
Lecture 8
Feb. 10, 2021

Q&A

Q: When should I prefer a directed graphical model to an undirected graphical model?

A: As we'll see today, the primary differences between them are:

1. the conditional independence assumptions they define
2. the normalization assumptions they make (Bayes Nets are locally normalized)

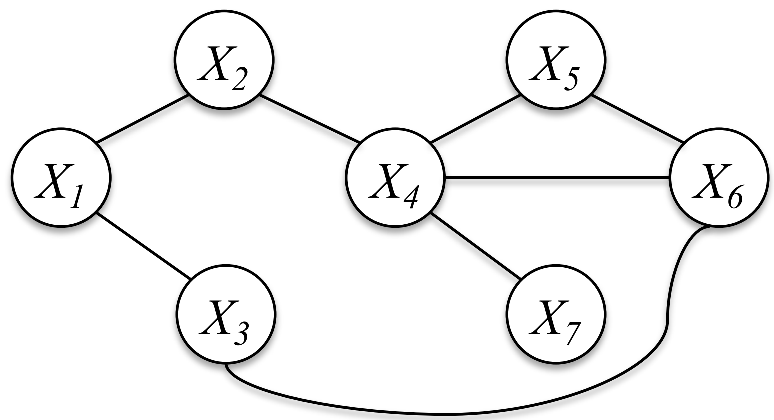
(That said, we'll also tie them together via a single framework: factor graphs.)

There are also some practical differences (e.g. ease of learning) that result from the locally vs. globally normalized difference.

GLOBAL / LOCAL / PAIRWISE MARKOV PROPERTIES

UGMs: Markov Properties

Suppose you wanted to list out all the conditional independencies for a given undirected graphical model…



UGMs: Markov Properties

Given an undirected graph G…

- *Def:* the **global Markov properties** are
  $$\mathbb{I}(G) = \{\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_C : \mathsf{sep}_G(\mathbf{X}_A, \mathbf{X}_B | \mathbf{X}_C)\}$$
  where X_A, X_B, X_C are sets of variables
- *Def*: the **pairwise Markov properties** are
  $$\mathbb{I}_p(G) = \{X_i \perp\!\!\!\perp X_j | \mathcal{X} - \{X_i, X_j\} : (X_i, X_j) \notin E(G)\}$$
- *Def*: the **local Markov properties** are
  $$\mathbb{I}_l(G) = \{X_i \perp\!\!\!\perp \mathcal{X} - \{X_i\} - MB_G(X_i) | MB_G(X_i)\}$$
  where $MB_G(X_i)$ returns the markov blanket of X_i

UGMs: Markov Properties

- *Proposition*: Any distribution that factors according to G satisfies the global Markov properties associated with G, i.e. $I(G) \subseteq I(P)$, where…
  - G is an undirected graph
  - $I(P)$ is the set of all conditional independencies satisfied by P
  - $I(G)$ is the set of global Markov properties
- *Proof*: (see whiteboard)

UGMs: Markov Properties

What about the converse of our proposition above? Not quite… but it does hold for positive distributions.

- *Theorem (Hammersley-Clifford)*: Any **positive** distribution satisfies the global Markov properties associated with G, i.e. $I(G) \subseteq I(P)$,also factors according to G where…
  - G is an undirected graph
  - I(P) is the set of all conditional independencies satisfied by P
  - I(G) is the set of global Markov properties
- *Proof*: (see Pradeep's lecture notes)

UGMs: Markov Properties

This is all about independencies, but what about **dependencies**?

- *Not true (too strong)*: Suppose X_i and X_j are not separated given $\mathbf{X_C}$ in UG G. Then there **does not exist** a distribution P that factors according to G where X_i and X_j are independent given $\mathbf{X_C}$

- *Theorem (slightly weaker)*: Suppose X_i and X_j are not separated given $\mathbf{X_C}$ in UG G. Then there exists a distribution P that factors according to G where X_i and X_j are dependent given $\mathbf{X_C}$

UGMs: Markov Properties

This is all about independencies, but what about **dependencies**?

- *Not true (too strong)*: Suppose X_i and X_j are not separated given $\mathbf{X_C}$ in UG G. Then there **does not exist** a distribution P that factors according to G where X_i and X_j are independent given $\mathbf{X_C}$

- *Theorem (slightly weaker)*: Suppose X_i and X_j are not separated given $\mathbf{X_C}$ in UG G. Then there exists a distribution P that factors according to G where X_i and X_j are dependent given $\mathbf{X_C}$

UGMs: Markov Properties

Given an undirected graph G…

- *Def:* the **global Markov properties** are
  $\mathbb{I}(G) = \{\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_C : \text{sep}_G(\mathbf{X}_A, \mathbf{X}_B | \mathbf{X}_C)\}$
  where X_A, X_B, X_C are sets of variables
- *Def*: the **pairwise Markov properties** are
  $\mathbb{I}_p(G) = \{X_i \perp\!\!\!\perp X_j | \mathcal{X} - \{X_i, X_j\} : (X_i, X_j) \notin E(G)\}$
- *Def*: the **local Markov properties** are
  $\mathbb{I}_l(G) = \{X_i \perp\!\!\!\perp \mathcal{X} - \{X_i\} - MB_G(X_i) | MB_G(X_i)\}$
  where $MB_G(X_i)$ returns the markov blanket of X_i

Proposition:

$$\mathbb{I}_p(G) \subseteq \mathbb{I}_l(G) \subseteq \mathbb{I}(G)$$

Proof: …left as an exercise…

UGMs: Markov Properties

If we restrict to positive distributions we can make a stronger statement:

Proposition 9 *For any positive distribution P, the following statements are equivalent:*

1. *P satisfies cond. independencies in $\mathbb{I}_p(G)$*
2. *P satisfies cond. independencies in $\mathbb{I}_\ell(G)$*
3. *P satisfies cond. independencies in $I(G)$*

DGMs: Markov Properties

For directed graphical models,

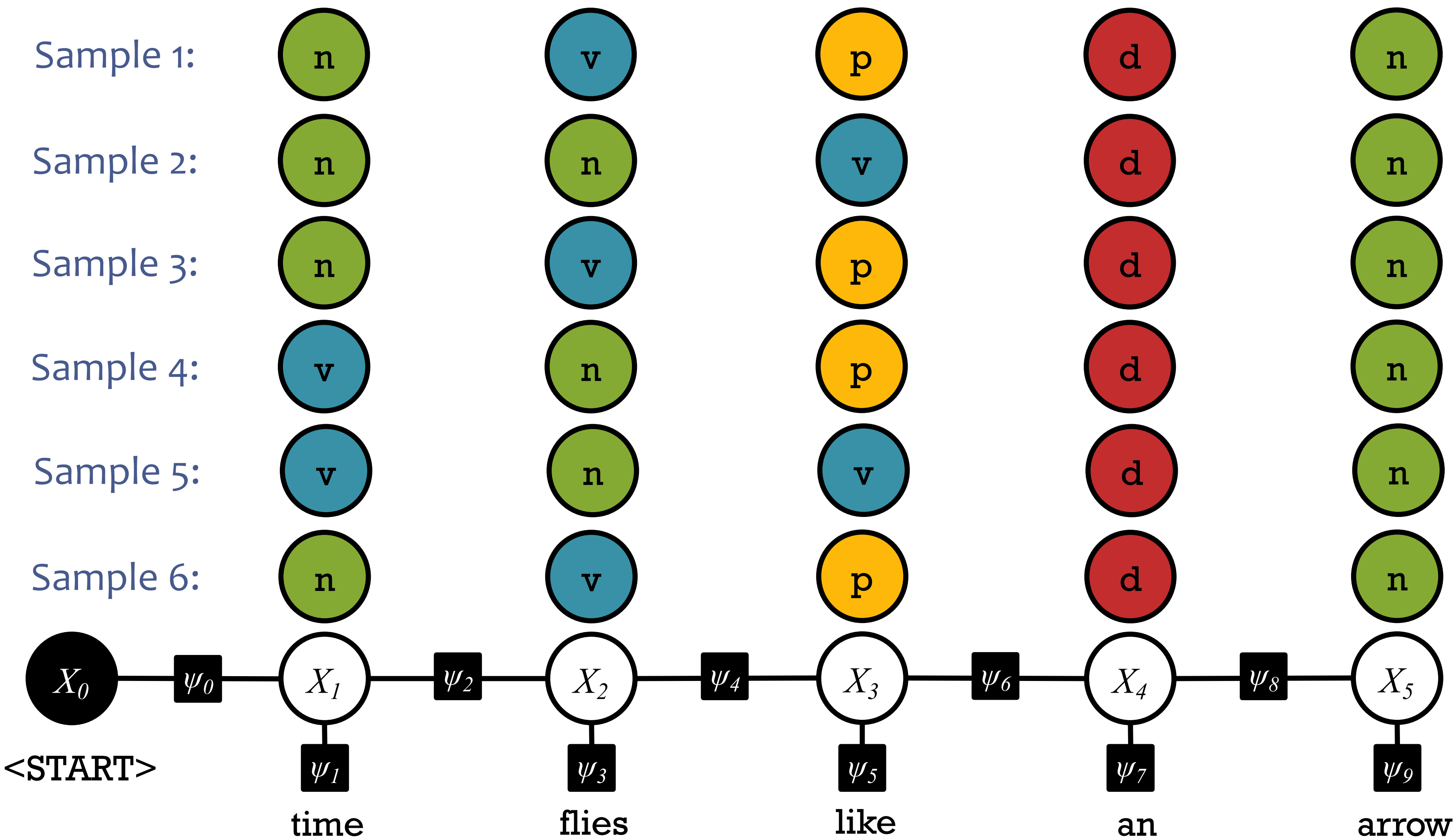
- we can also spell out global, local, and pairwise Markov properties
- they derive from d-separation, an interesting notion of locality, and the Markov blanket
- similar theorems to UGM

Representation of both directed and undirected graphical models

FACTOR GRAPHS

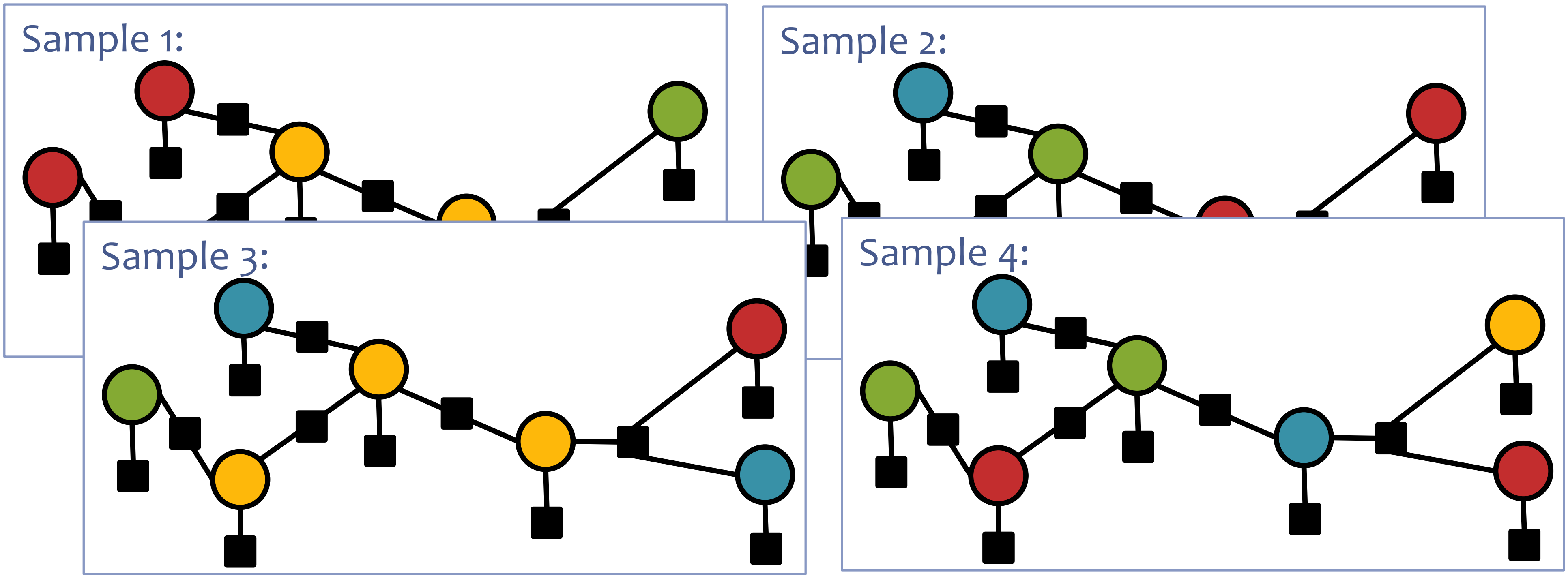
Sampling from a Joint Distribution

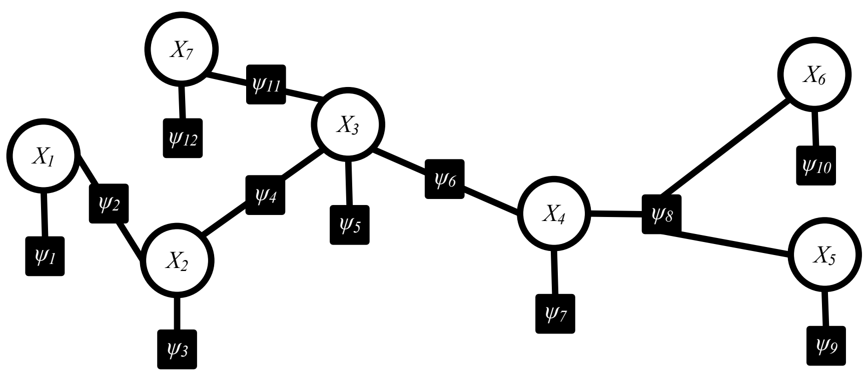
A **joint distribution** defines a probability $p(\boldsymbol{x})$ for each assignment of values $\boldsymbol{x}$ to variables $\boldsymbol{X}$. This gives the **proportion** of samples that will equal $\boldsymbol{x}$.



Sampling from a Joint Distribution

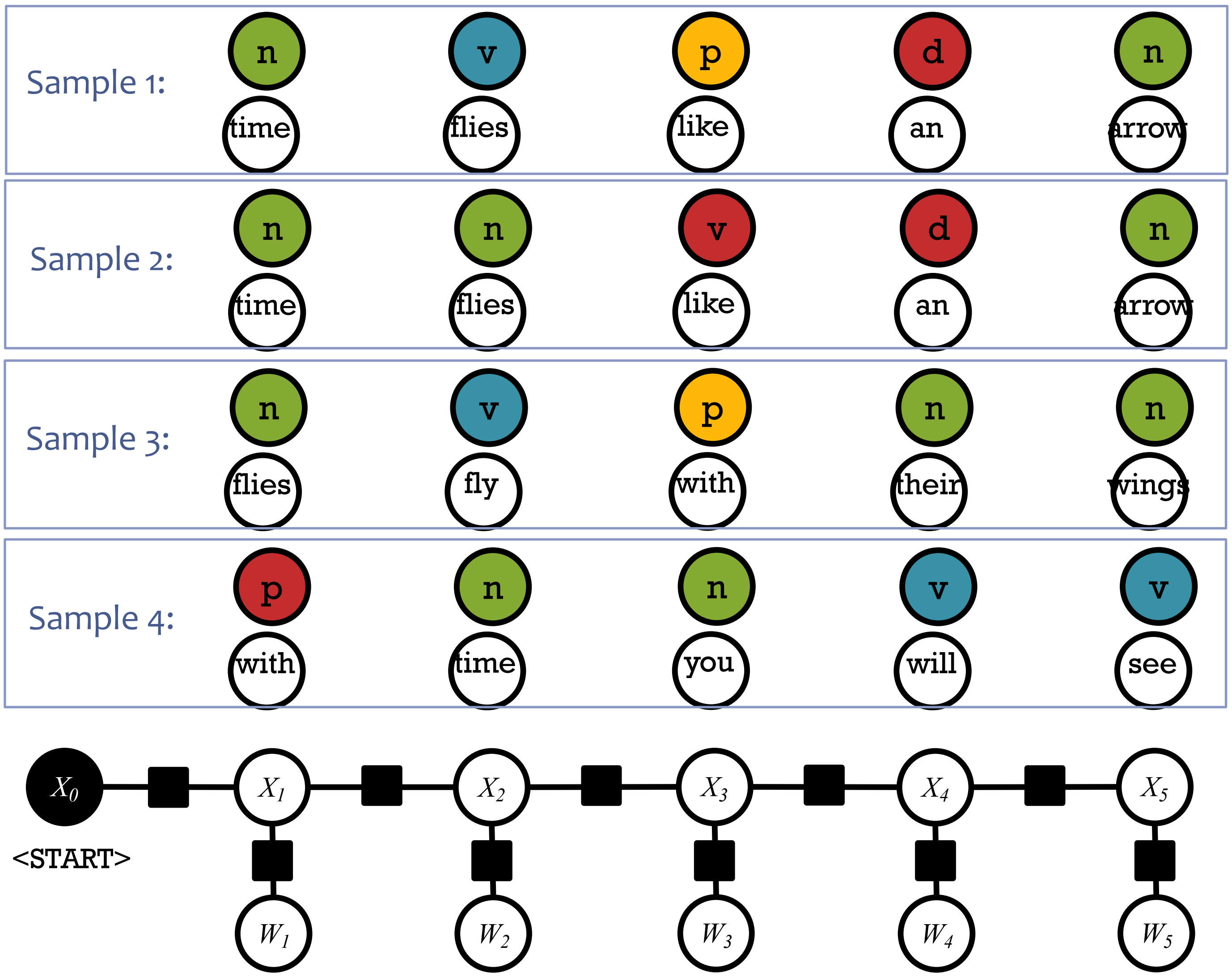
A **joint distribution** defines a probability $p(\boldsymbol{x})$ for each assignment of values $\boldsymbol{x}$ to variables $\boldsymbol{X}$. This gives the **proportion** of samples that will equal $\boldsymbol{x}$.





Sampling from a Joint Distribution

A **joint distribution** defines a probability $p(\boldsymbol{x})$ for each assignment of values $\boldsymbol{x}$ to variables $\boldsymbol{X}$. This gives the **proportion** of samples that will equal $\boldsymbol{x}$.



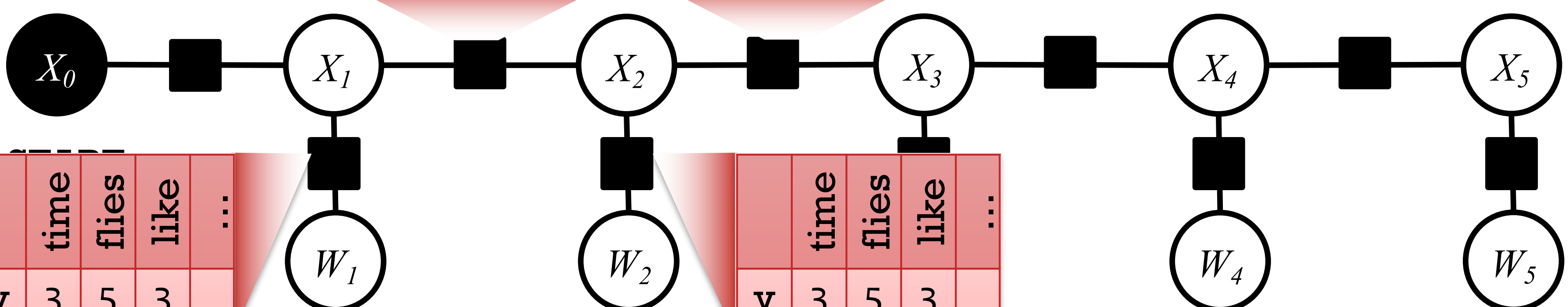
Factors have local opinions (≥ 0)

Each black box looks at *some* of the tags X_i and words W_i

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

Note: We chose to reuse the same factors at different positions in the sentence.



	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

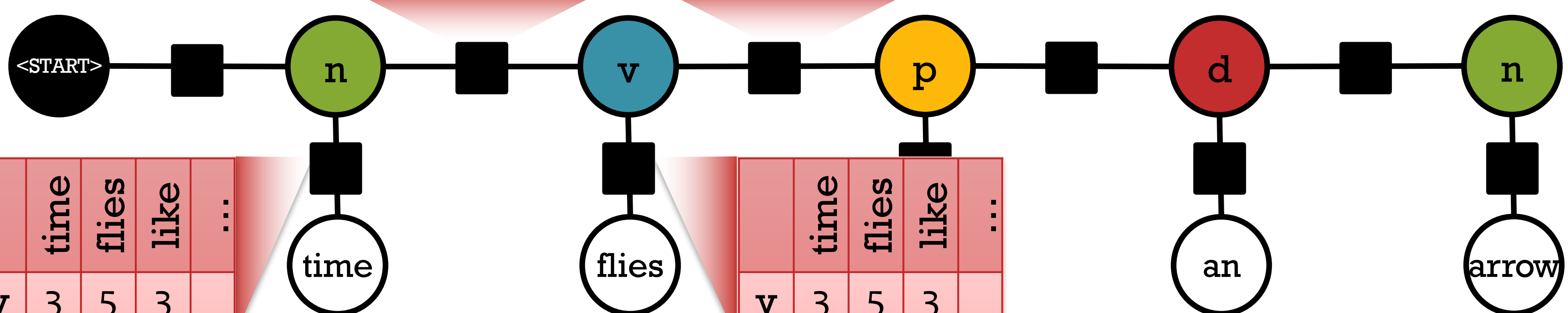
Factors have local opinions (≥ 0)

Each black box looks at *some* of the tags X_i and words W_i

$$p(\text{n, v, p, d, n, time, flies, like, an, arrow}) \quad = \quad ?$$

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0



	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

Global probability = product of local opinions

Each black box looks at *some* of the tags X_i and words W_i

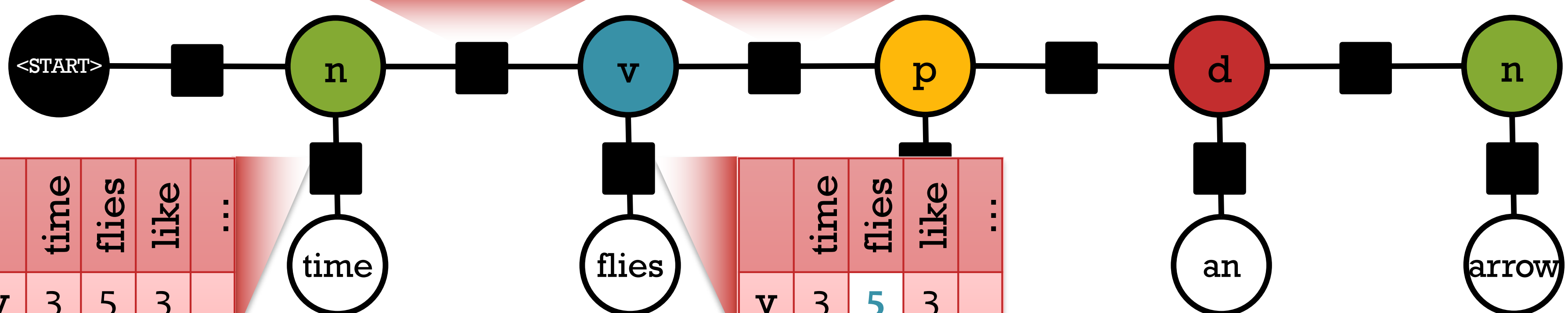
$$p(\text{n, v, p, d, n, time, flies, like, an, arrow}) = \frac{1}{Z}(4 * 8 * 5 * 3 * \dots)$$

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

Uh-oh! The probabilities of the various assignments sum up to Z > 1.

So divide them all by Z.



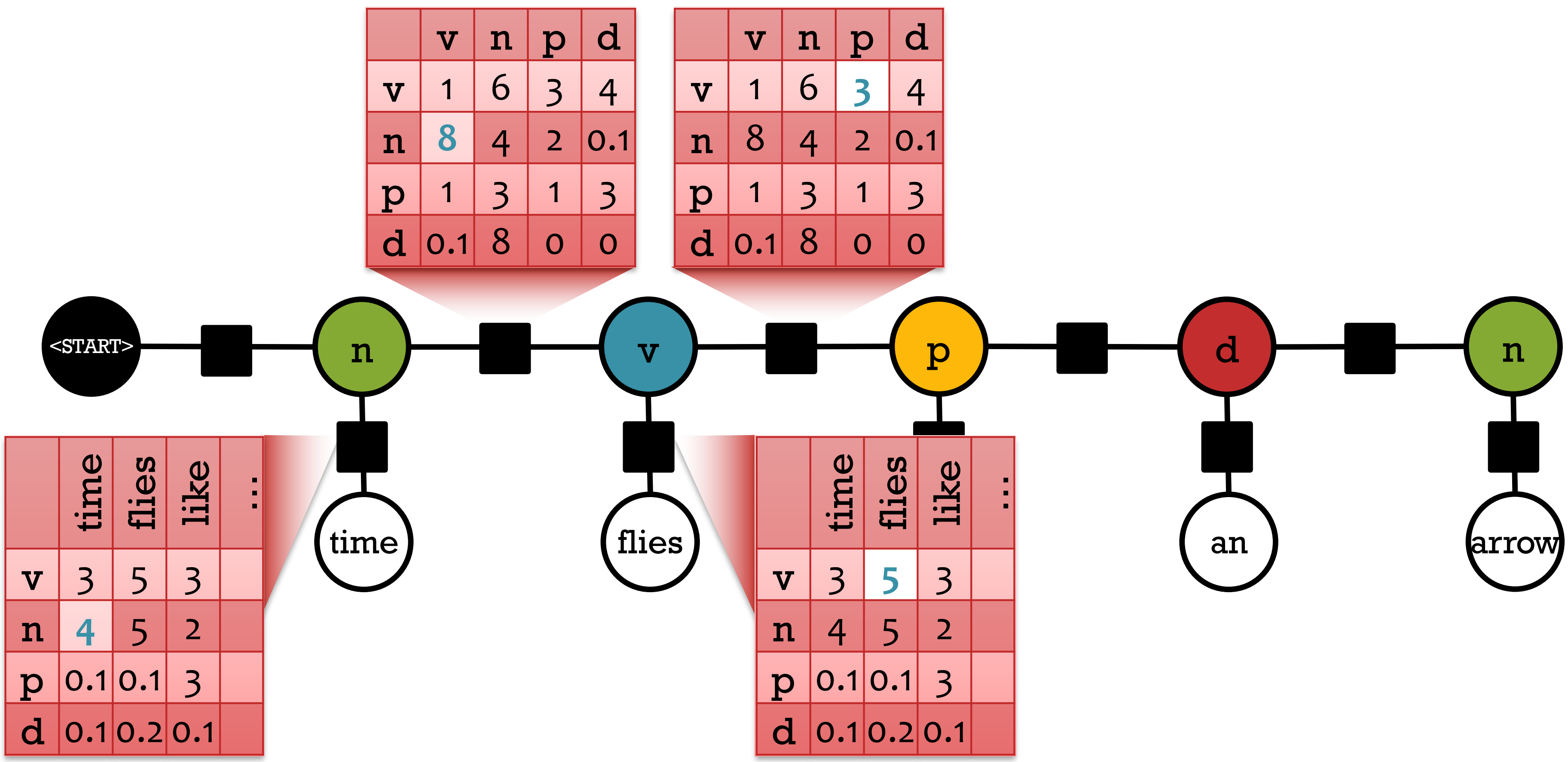
	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

Markov Random Field (MRF)

Joint distribution over tags X_i <u>and</u> words W_i
The individual factors aren't *necessarily* probabilities.

$$p(\text{n, v, p, d, n, time, flies, like, an, arrow}) = \frac{1}{Z}(4 * 8 * 5 * 3 * \ldots)$$



	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

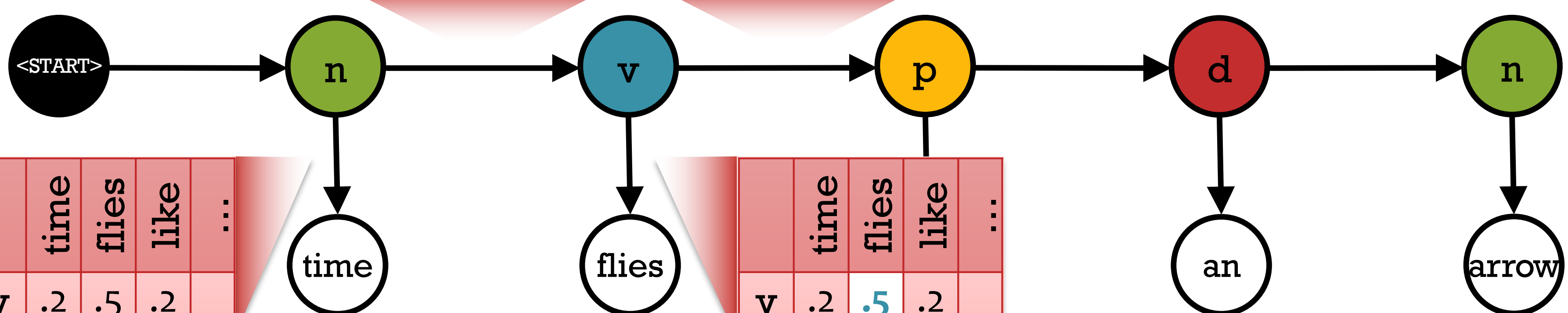
Bayesian Networks

But sometimes we *choose* to make them probabilities.
Constrain each row of a factor to sum to one. Now $Z = 1$.

$$p(\text{n, v, p, d, n, time, flies, like, an, arrow}) = \cancel{\frac{1}{Z}}(.3 * .8 * .2 * .5 * \ldots)$$

	v	n	p	d
v	.1	.4	.2	.3
n	.8	.1	.1	0
p	.2	.3	.2	.3
d	.2	.8	0	0

	v	n	p	d
v	.1	.4	.2	.3
n	.8	.1	.1	0
p	.2	.3	.2	.3
d	.2	.8	0	0



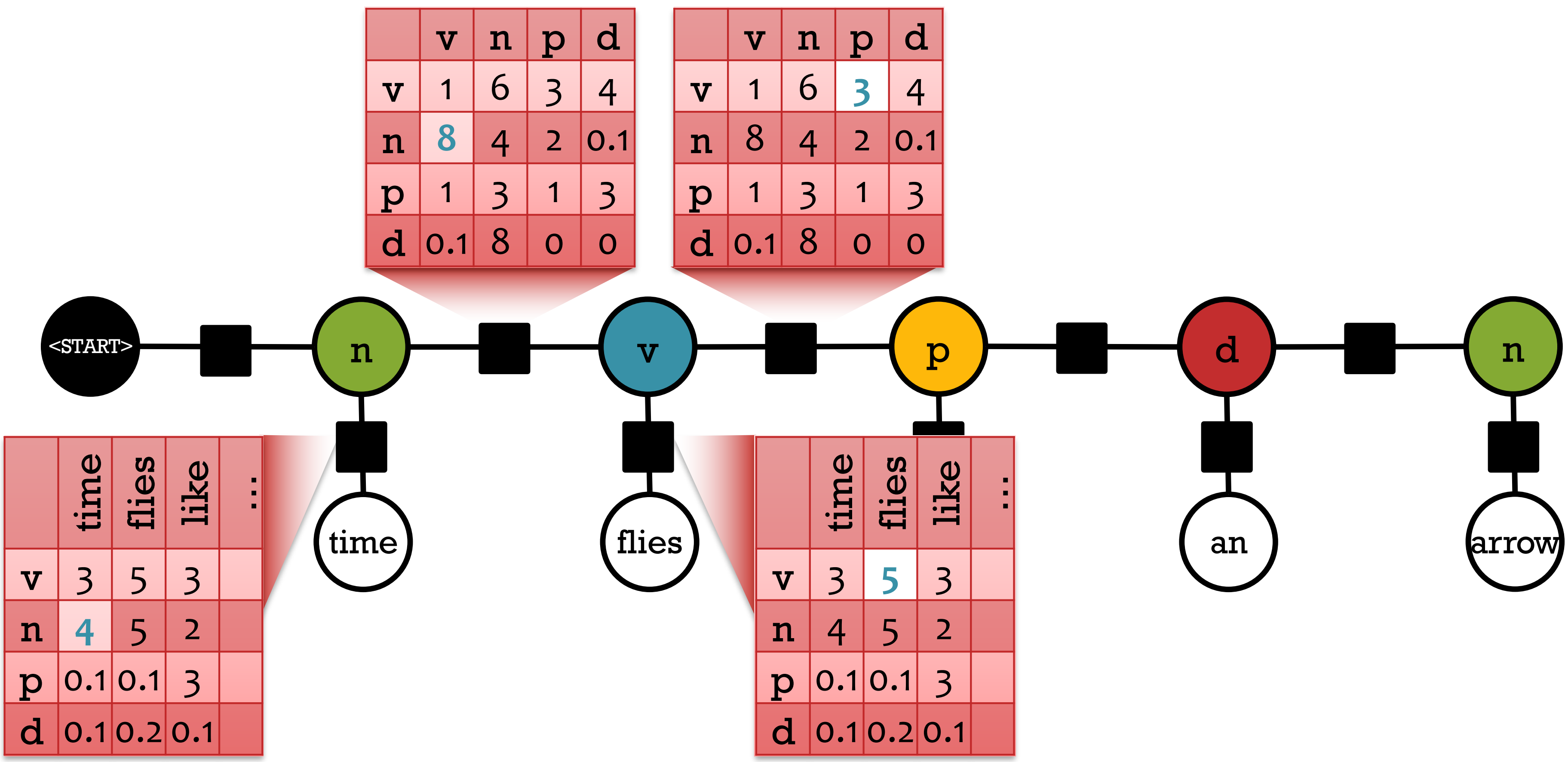
	time	flies	like	…
v	.2	.5	.2	
n	.3	.4	.2	
p	.1	.1	.3	
d	.1	.2	.1	

	time	flies	like	…
v	.2	.5	.2	
n	.3	.4	.2	
p	.1	.1	.3	
d	.1	.2	.1	

Markov Random Field (MRF)

Joint distribution over tags X_i and words W_i

$$p(\text{n, v, p, d, n, time, flies, like, an, arrow}) = \frac{1}{Z}(4 * 8 * 5 * 3 * \ldots)$$



	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

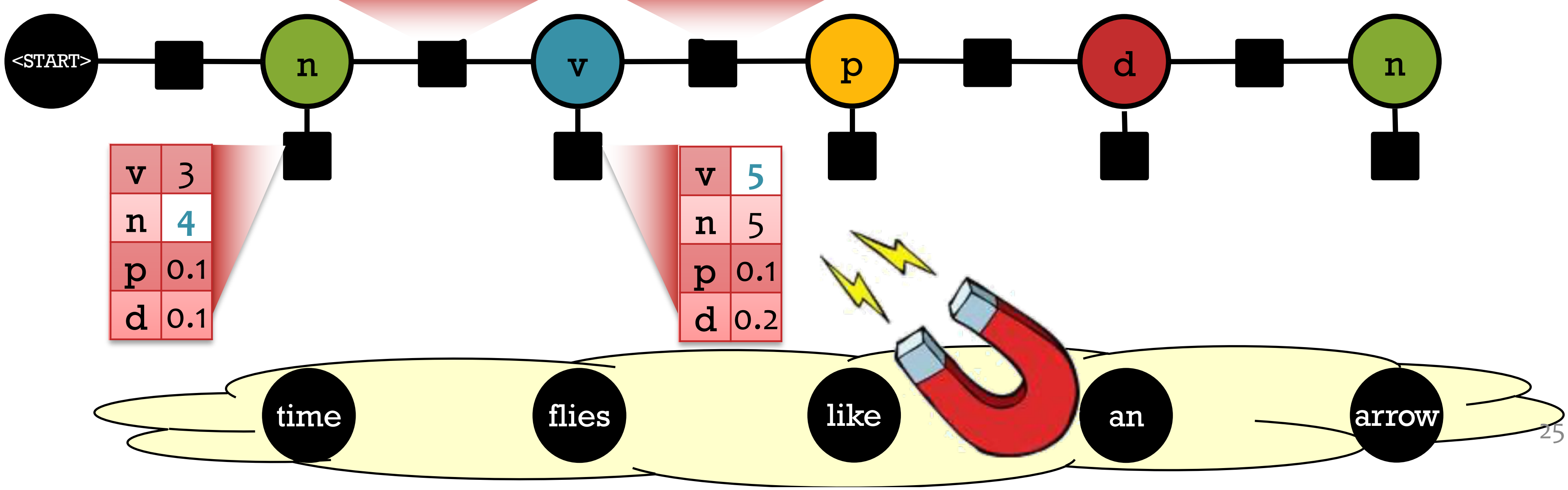
Conditional Random Field (CRF)

Conditional distribution over tags X_i given words w_i.
The factors and Z are now specific to the sentence w.

$$p(\text{n, v, p, d, n} \mid \text{time, flies, like, an, arrow}) \quad = \frac{1}{Z}(4 * 8 * 5 * 3 * \ldots)$$

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0

	v	n	p	d
v	1	6	3	4
n	8	4	2	0.1
p	1	3	1	3
d	0.1	8	0	0



v	3
n	4
p	0.1
d	0.1

v	5
n	5
p	0.1
d	0.2

How General Are Factor Graphs?

- Factor graphs can be used to describe
  - Markov Random Fields (undirected graphical models)
    - i.e., log-linear models over a tuple of variables
  - Conditional Random Fields
  - Bayesian Networks (directed graphical models)

- *Inference* treats all of these interchangeably.
  - Convert your model to a factor graph first.
  - Pearl (1988) gave key strategies for *exact* inference:
    - **Belief propagation**, for inference on *acyclic* graphs
    - **Junction tree algorithm**, for making *any* graph acyclic (by merging variables and factors: blows up the runtime)

Factor Graph Notation

- Variables:

$$\mathcal{X} = \{X_1, \ldots, X_i, \ldots, X_n\}$$

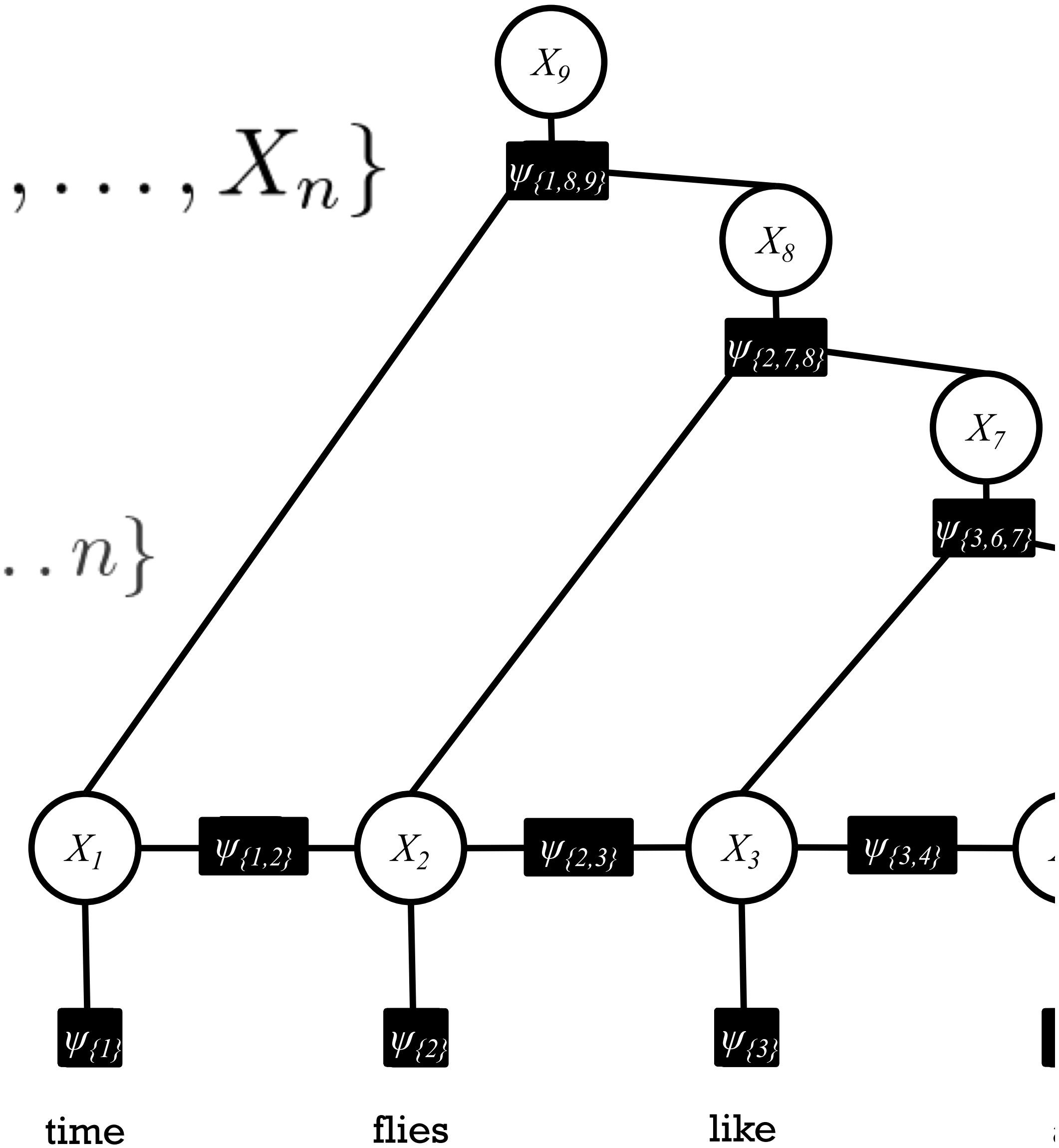
- Factors:

$$\psi_\alpha, \psi_\beta, \psi_\gamma, \ldots$$

where $\alpha, \beta, \gamma, \ldots \subseteq \{1, \ldots n\}$

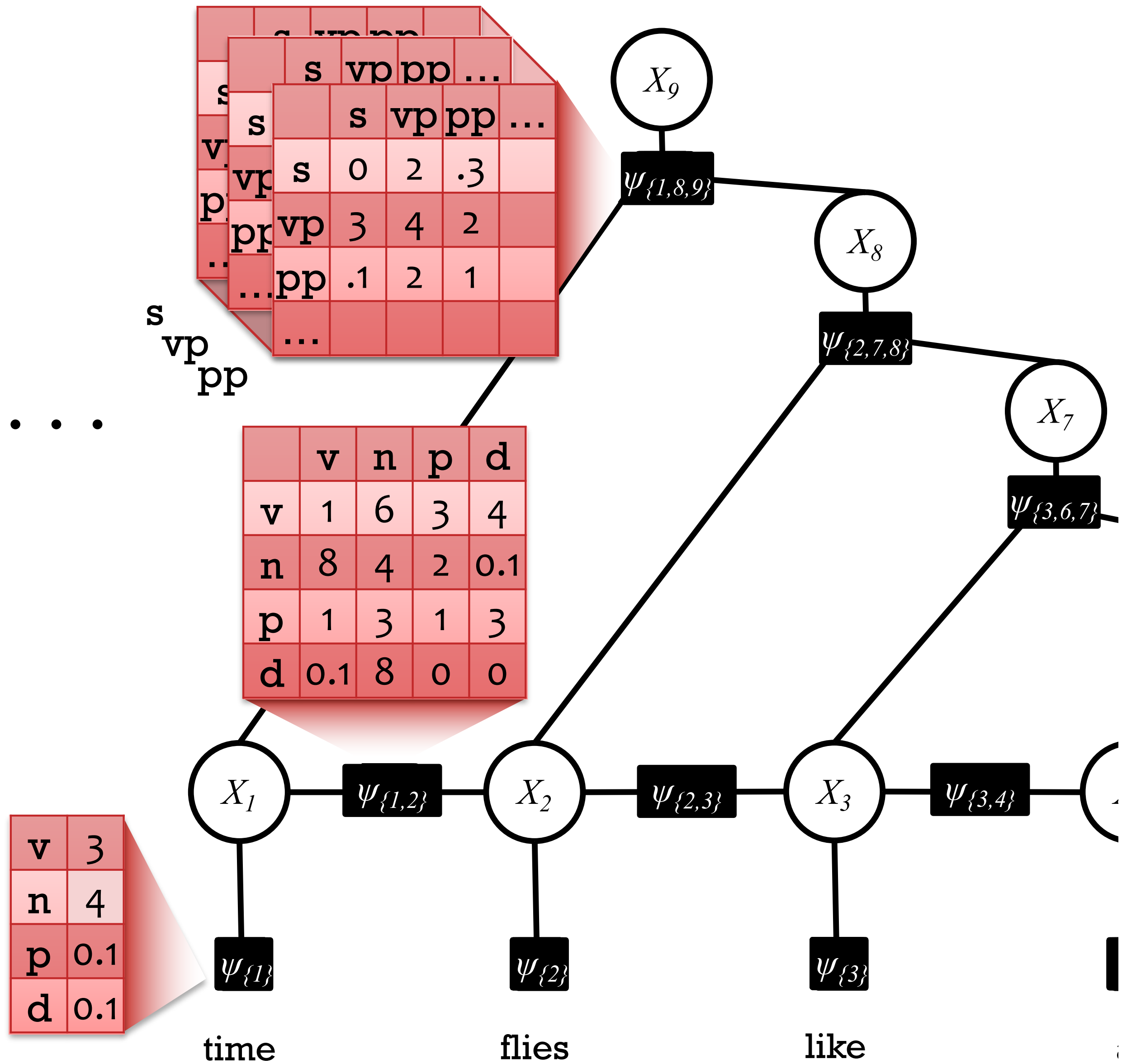
Joint Distribution

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_\alpha \psi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}})$$



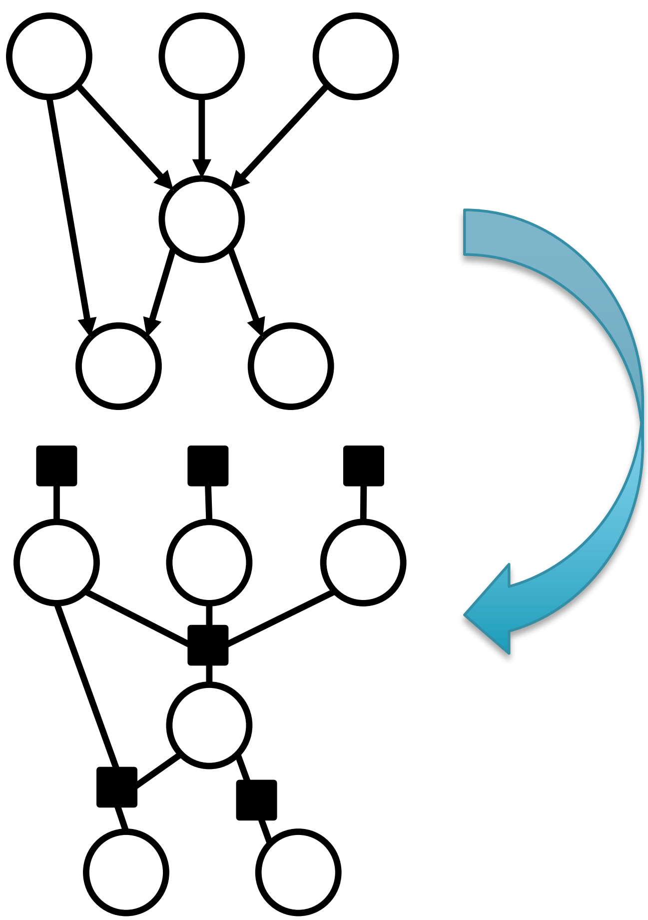
Factors are Tensors

- Factors:
  $$\psi_\alpha, \psi_\beta, \psi_\gamma, \ldots$$

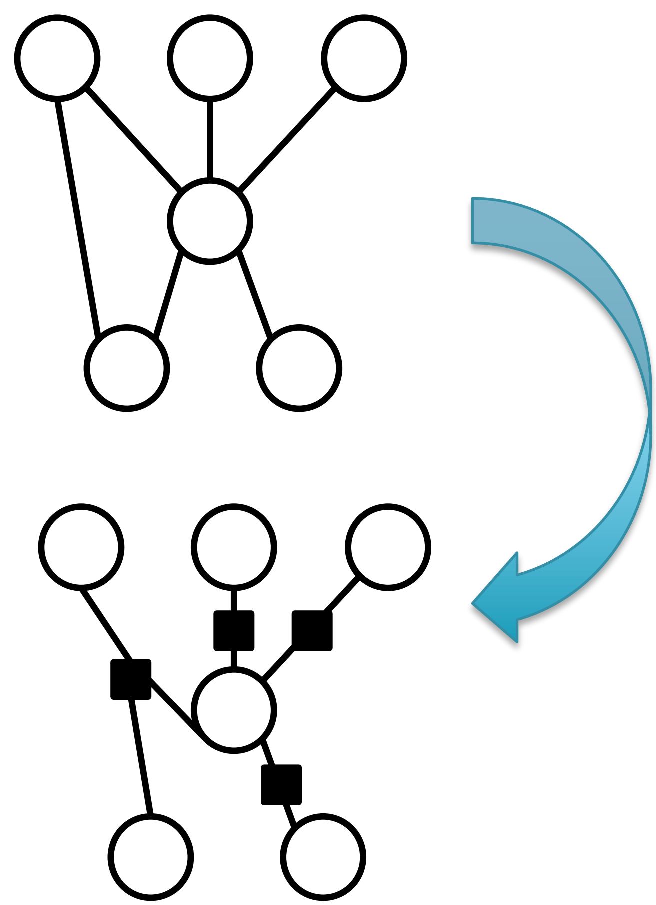


Converting to Factor Graphs

Each conditional and marginal distribution in a **directed GM** becomes a factor



Each maximal clique in an **undirected GM** becomes a factor



Equivalence of directed and undirected trees

- Any undirected tree can be converted to a directed tree by choosing a root node and directing all edges away from it
- A directed tree and the corresponding undirected tree make the same conditional independence assertions
- Parameterizations are essentially the same.
  - Undirected tree:
  - Directed tree:
  - Equivalence:

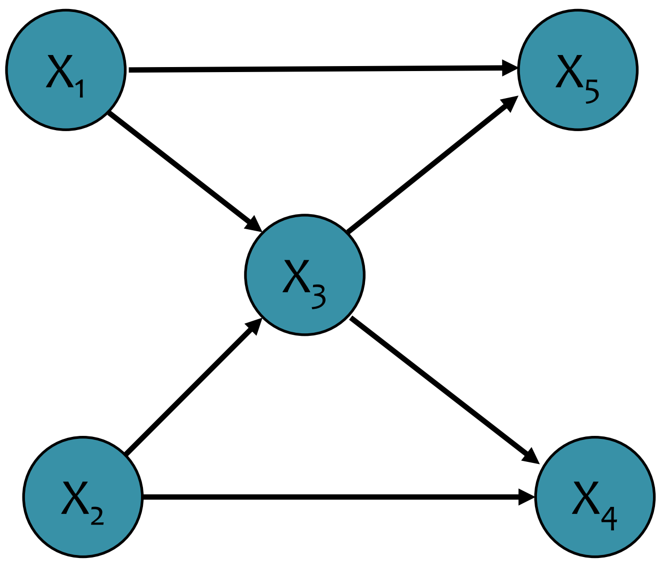
$$p(x) = \frac{1}{Z}\left(\prod_{i\in V}\psi(x_i)\prod_{(i,j)\in E}\psi(x_i,x_j)\right)$$

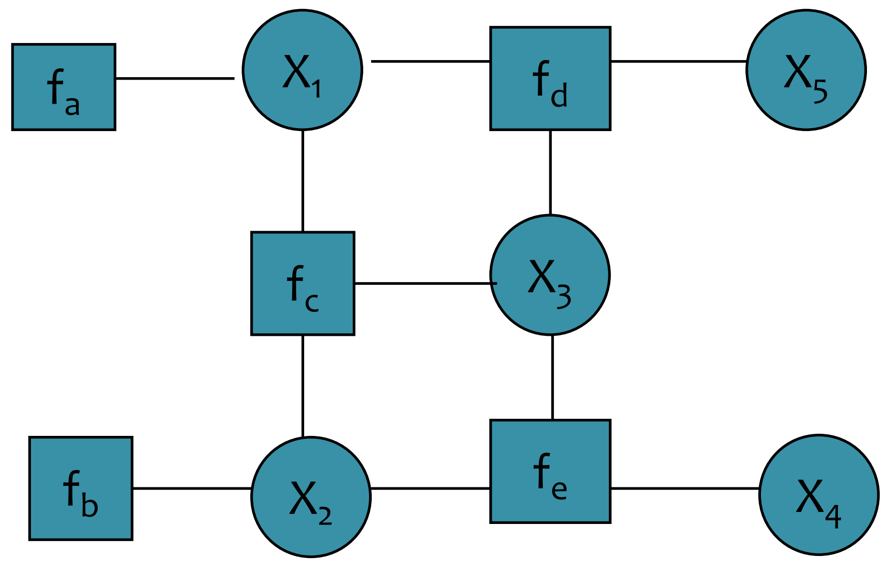
$$p(x) = p(x_r)\prod_{(i,j)\in E}p(x_j|x_i)$$

$$\psi(x_r) = p(x_r); \quad \psi(x_i,x_j) = p(x_j|x_i);$$
$$Z = 1, \quad \psi(x_i) = 1$$

Factor Graph Examples

- Example 1



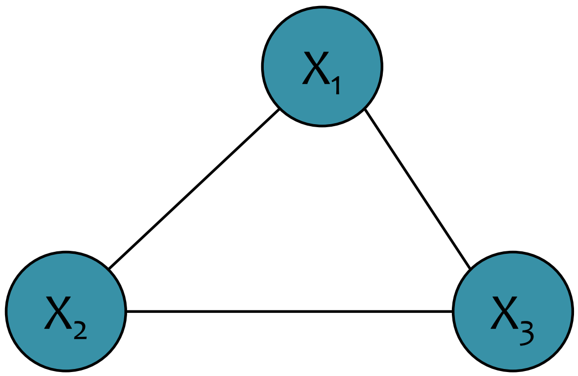


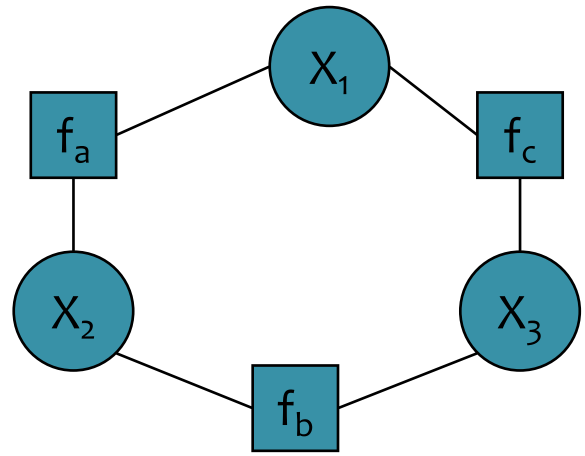
$P(X_1)\ \ P(X_2)\ \ P(X_3|X_1,X_2)\ \ P(X_5|X_1,X_3)\ \ P(X_4|X_2,X_3)$

$f_a(X_1)\ \ f_b(X_2)\ \ f_c(X_3,X_1,X_2)\ \ f_d(X_5,X_1,X_3)\ \ f_e(X_4,X_2,X_3)$

Factor Graph Examples

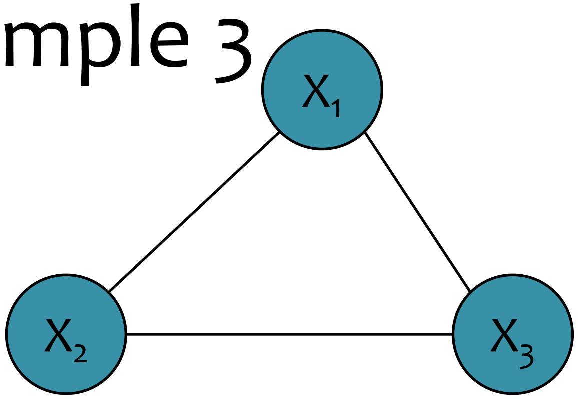
- Example 2

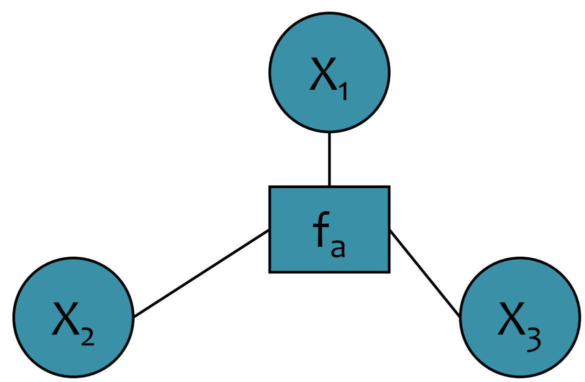




$\psi(x_1,x_2,x_3) = f_a(x_1,x_2)f_b(x_2,x_3)f_c(x_3,x_1)$

- Example 3

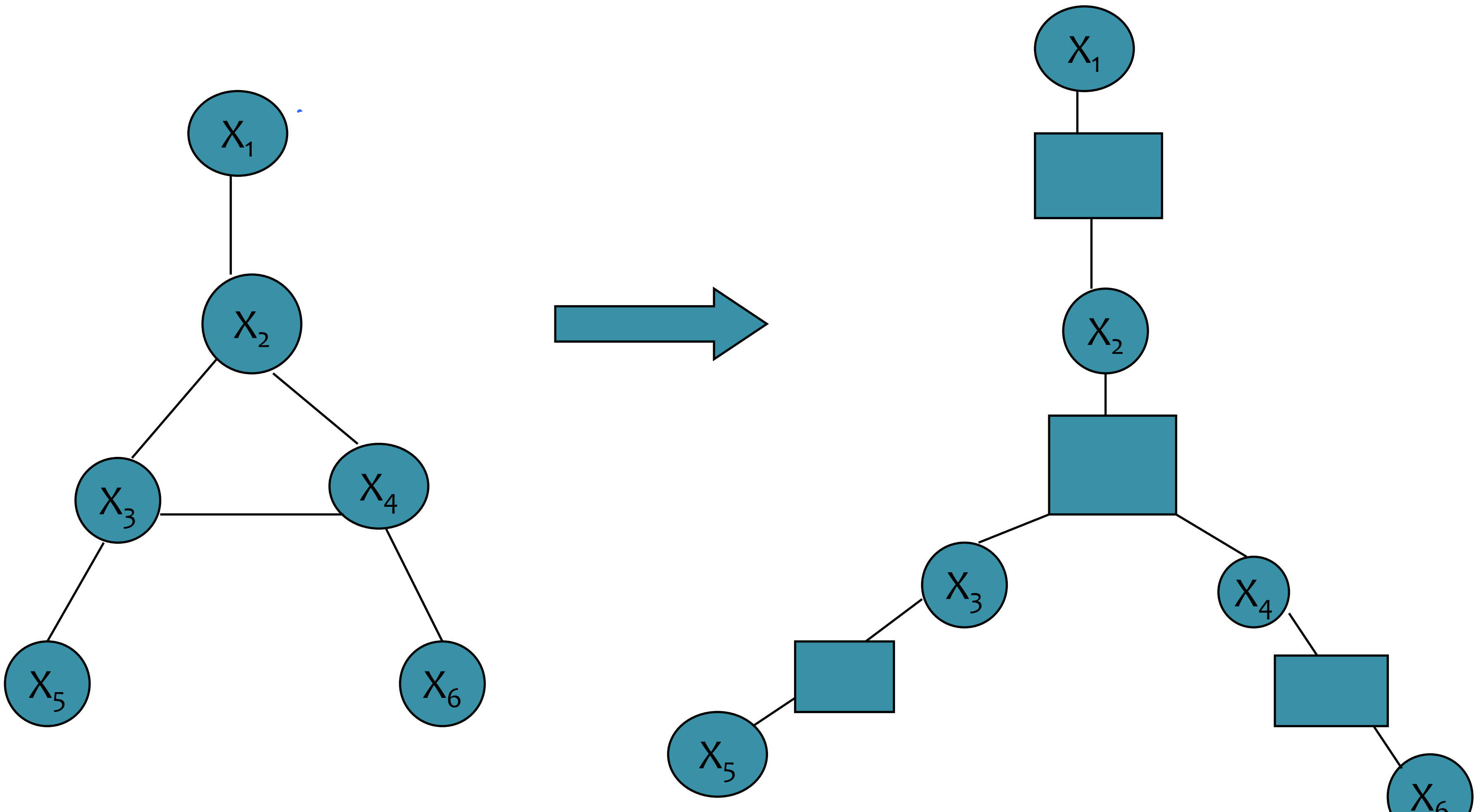




$\psi(x_1,x_2,x_3) = f_a(x_1,x_2,x_3)$

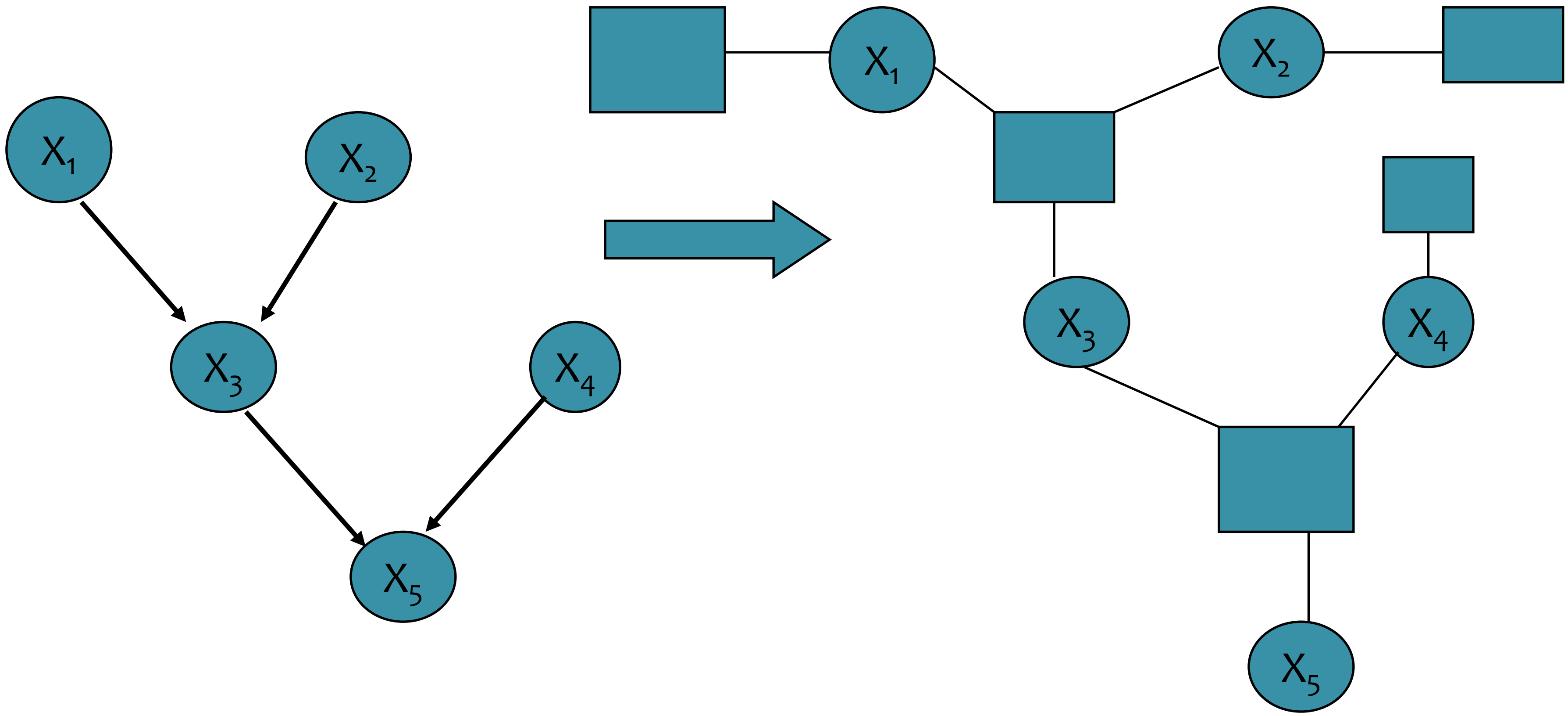
Tree-like Undirected GMs to Factor Trees

- Example 4

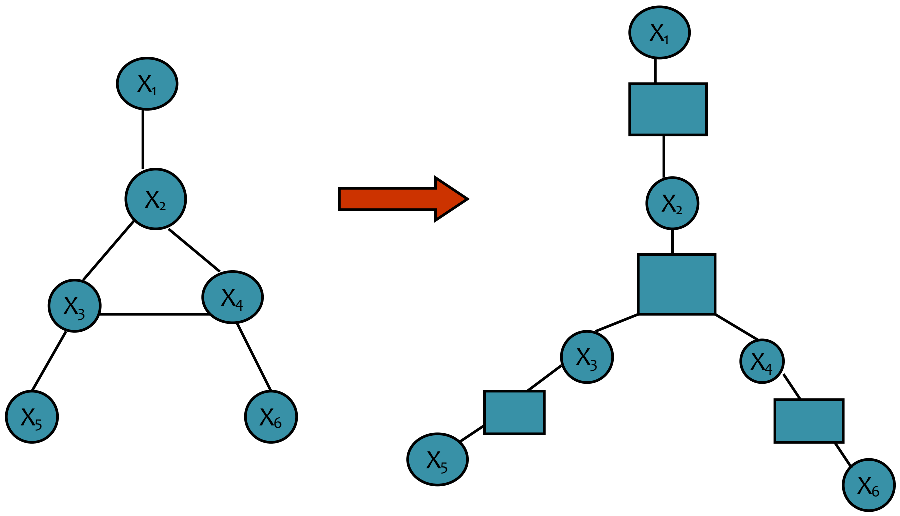


Poly-trees to Factor trees

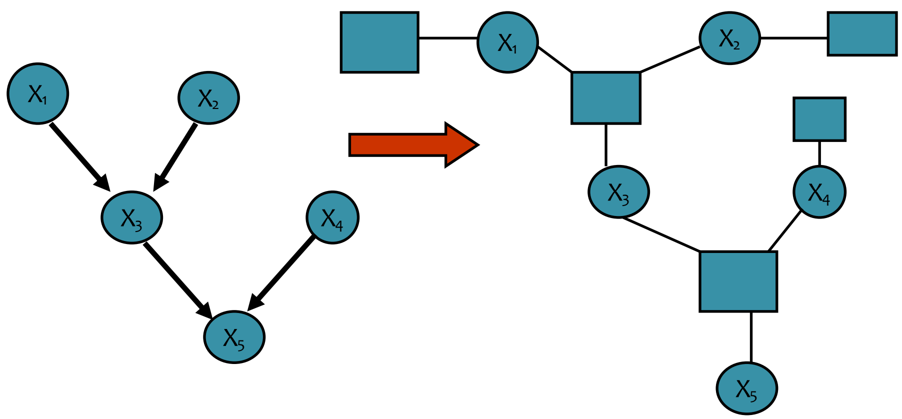
- Example 5



Why factor graphs?



- Because FG turns tree-like graphs to factor trees,
- Trees are a data-structure that guarantees correctness of BP !



MRF VS. CRF

MRF vs. CRF

Markov Random Field (MRF):

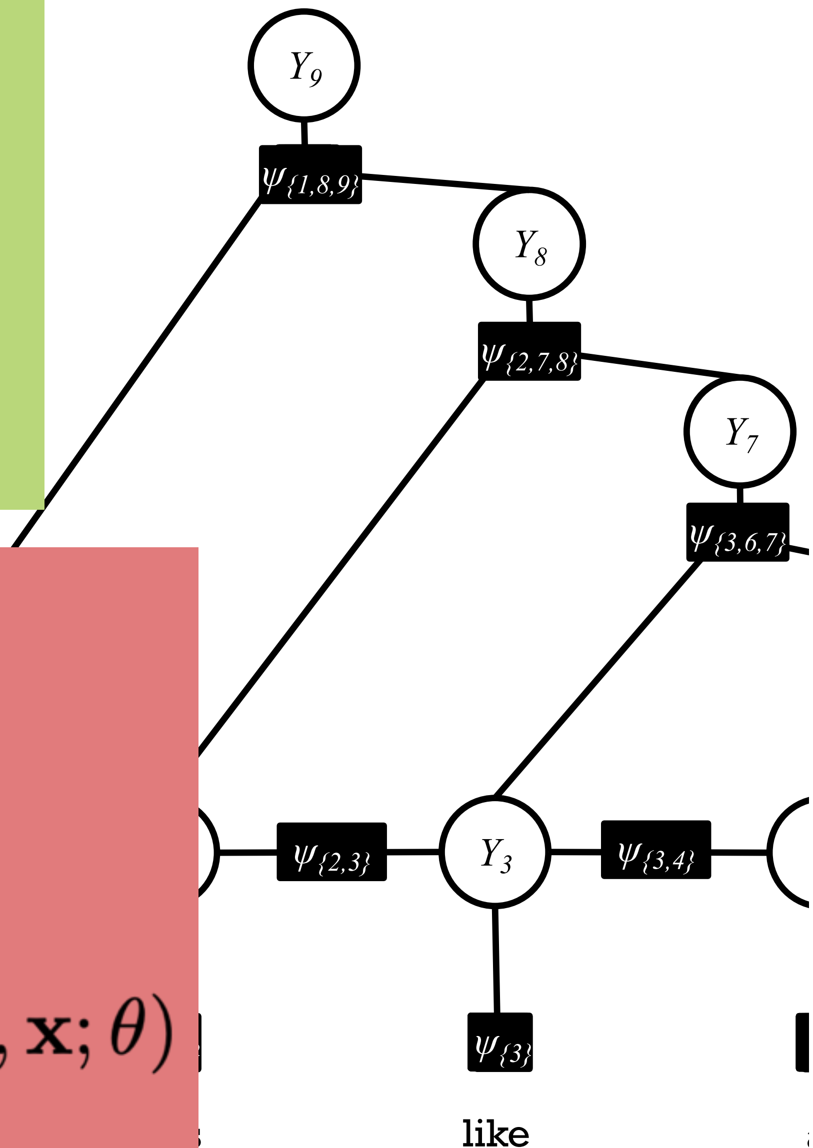
- just a distribution over variables **y**
- partition function Z is just a function of the parameters

$$p_{\boldsymbol{\theta}}(\mathbf{y}) = \frac{1}{Z(\theta)} \prod_{\alpha} \psi_{\alpha}(\mathbf{y}_{\alpha}; \theta)$$

Conditional Random Field (CRF):

- conditions on some additional observed variables **x**
- partition function Z is a function of **x** as well

$$p_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x}; \theta)} \prod_{\alpha} \psi_{\alpha}(\mathbf{y}_{\alpha}, \mathbf{x}; \theta)$$

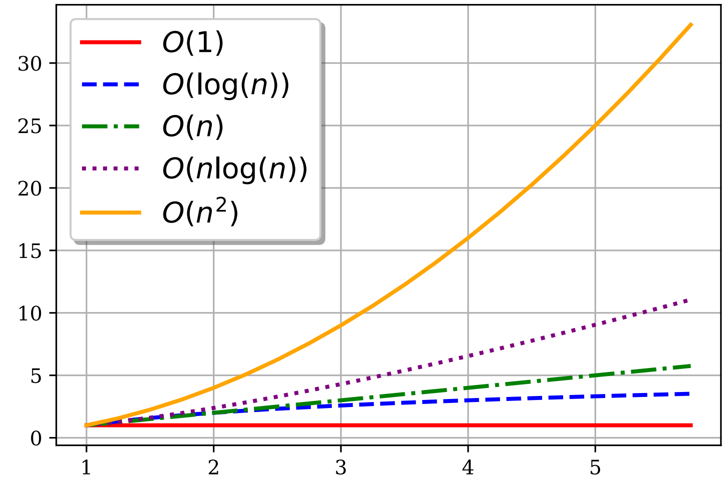


COMPUTATIONAL COMPLEXITY

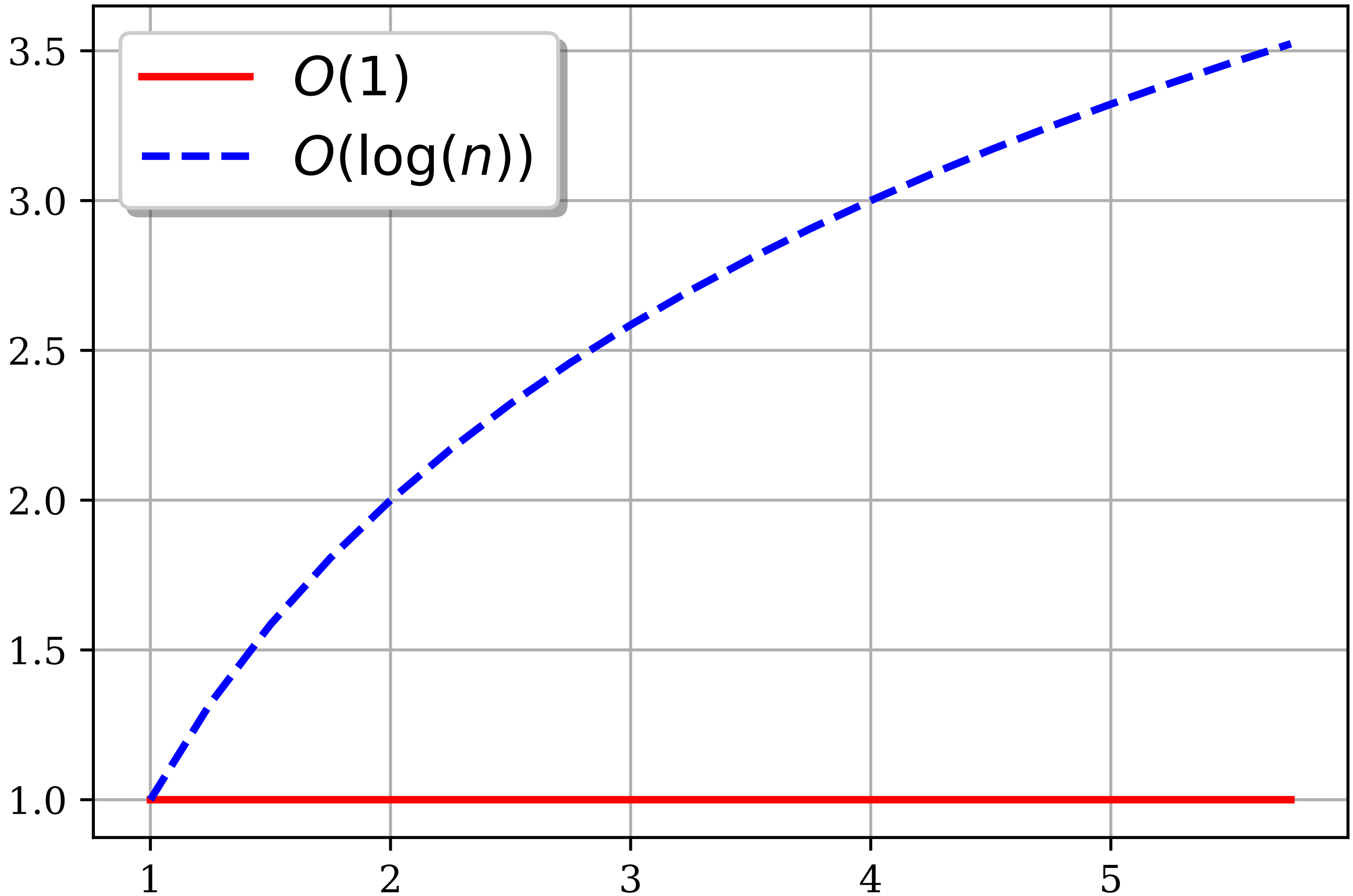
Analysis of Algorithms

Key Questions:

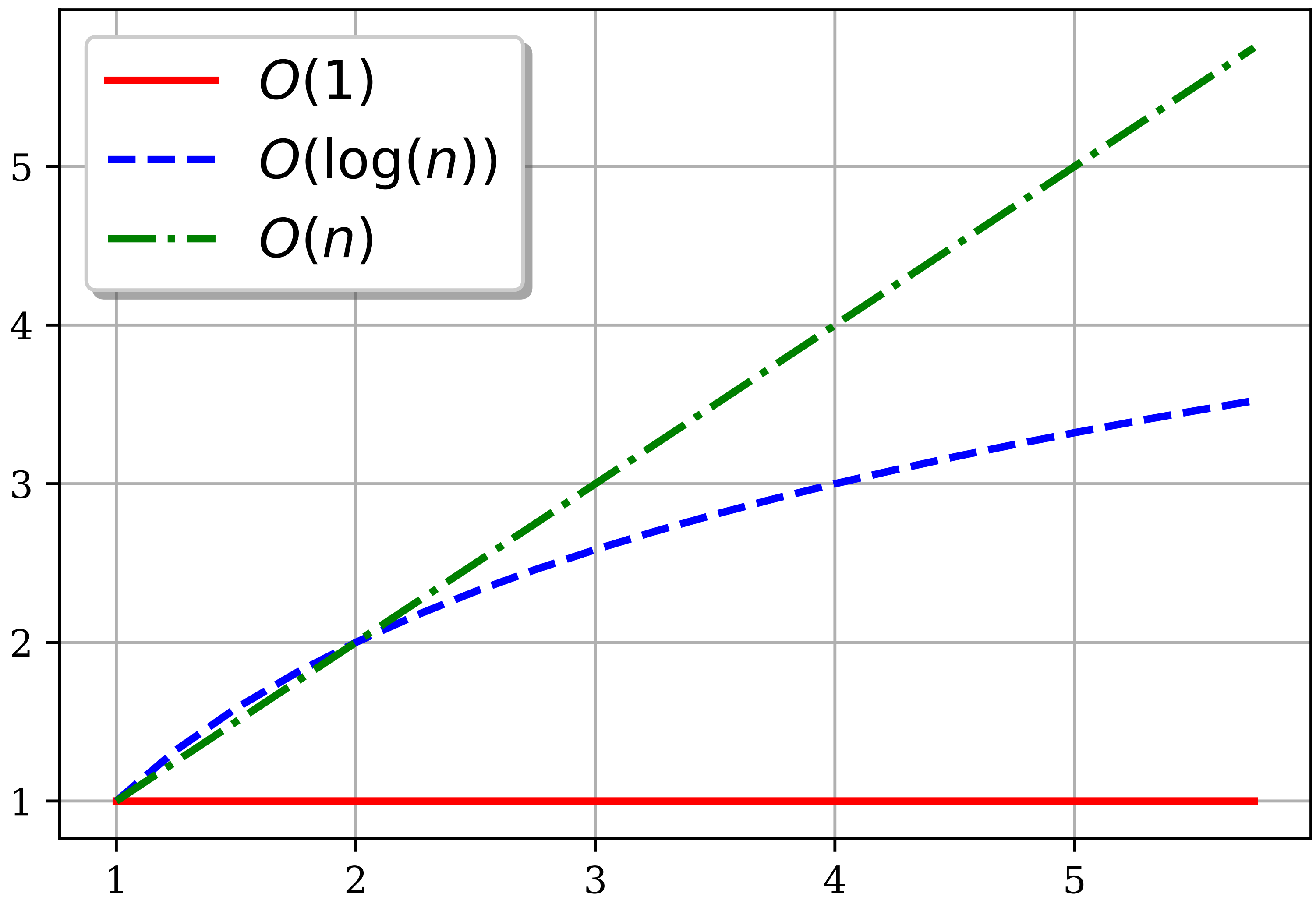
1. Given a single algorithm, will it complete on a given input in a reasonable amount of time/space?
2. Given two algorithms which one is better?



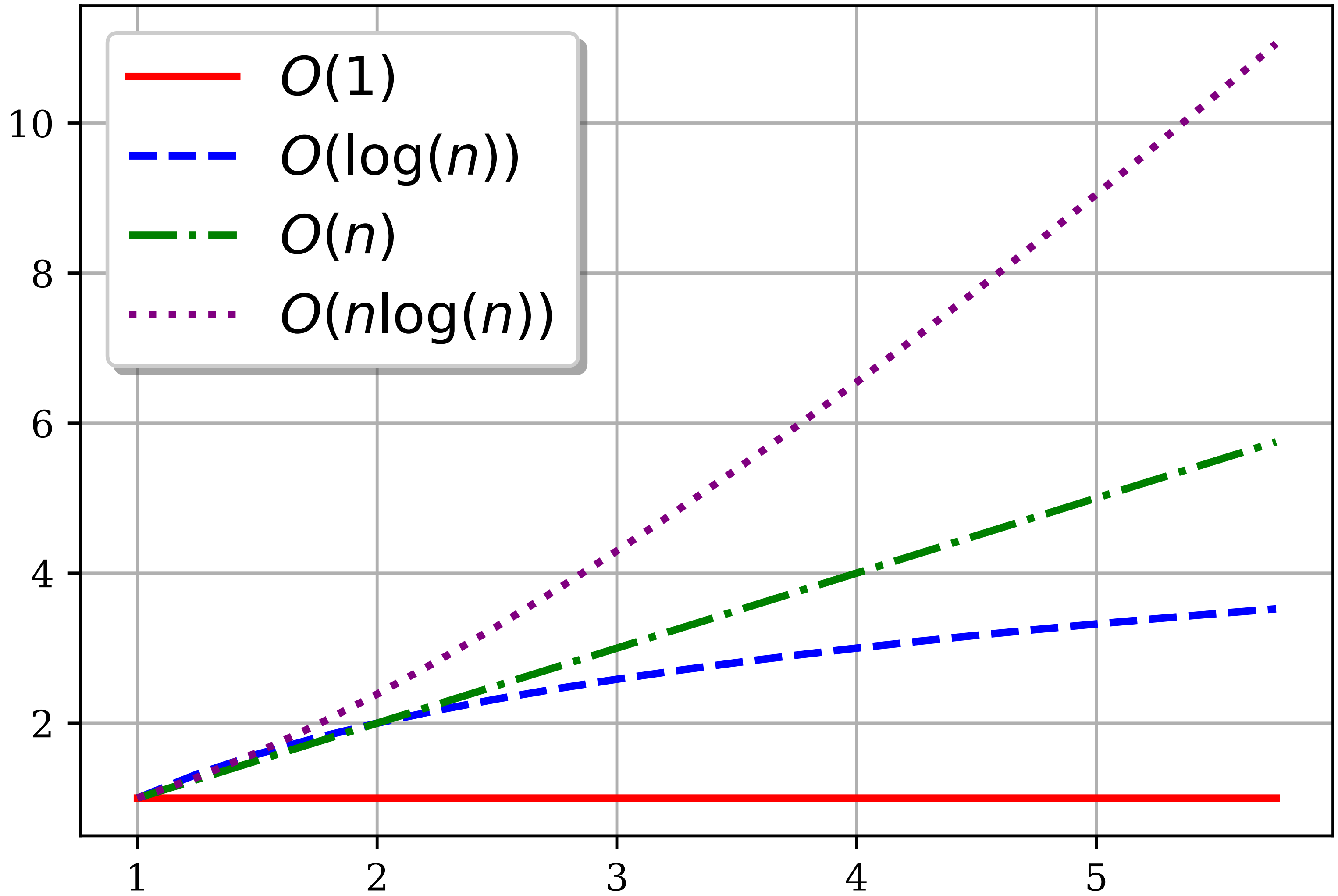
Comparing Algorithm Runtimes



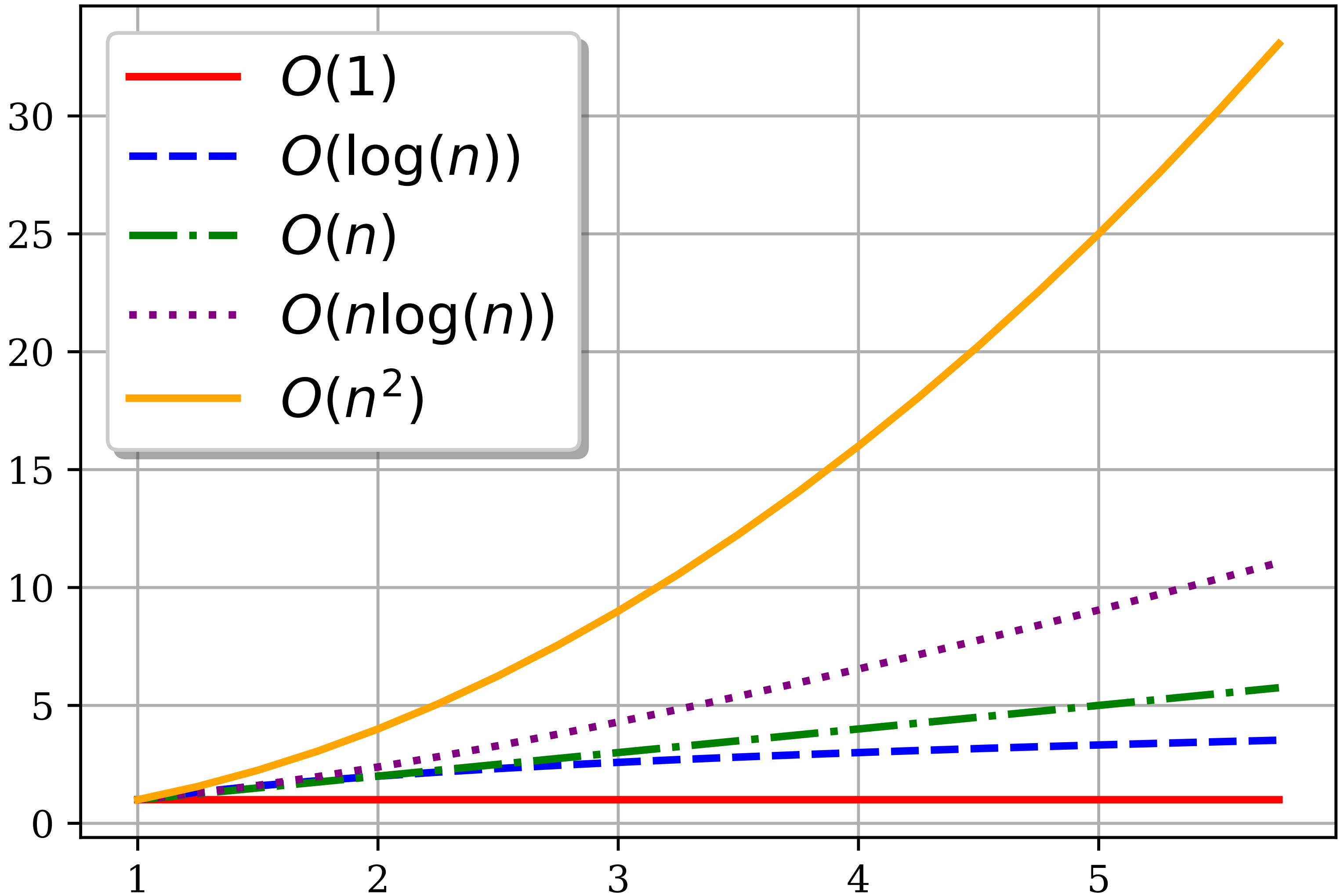
Comparing Algorithm Runtimes



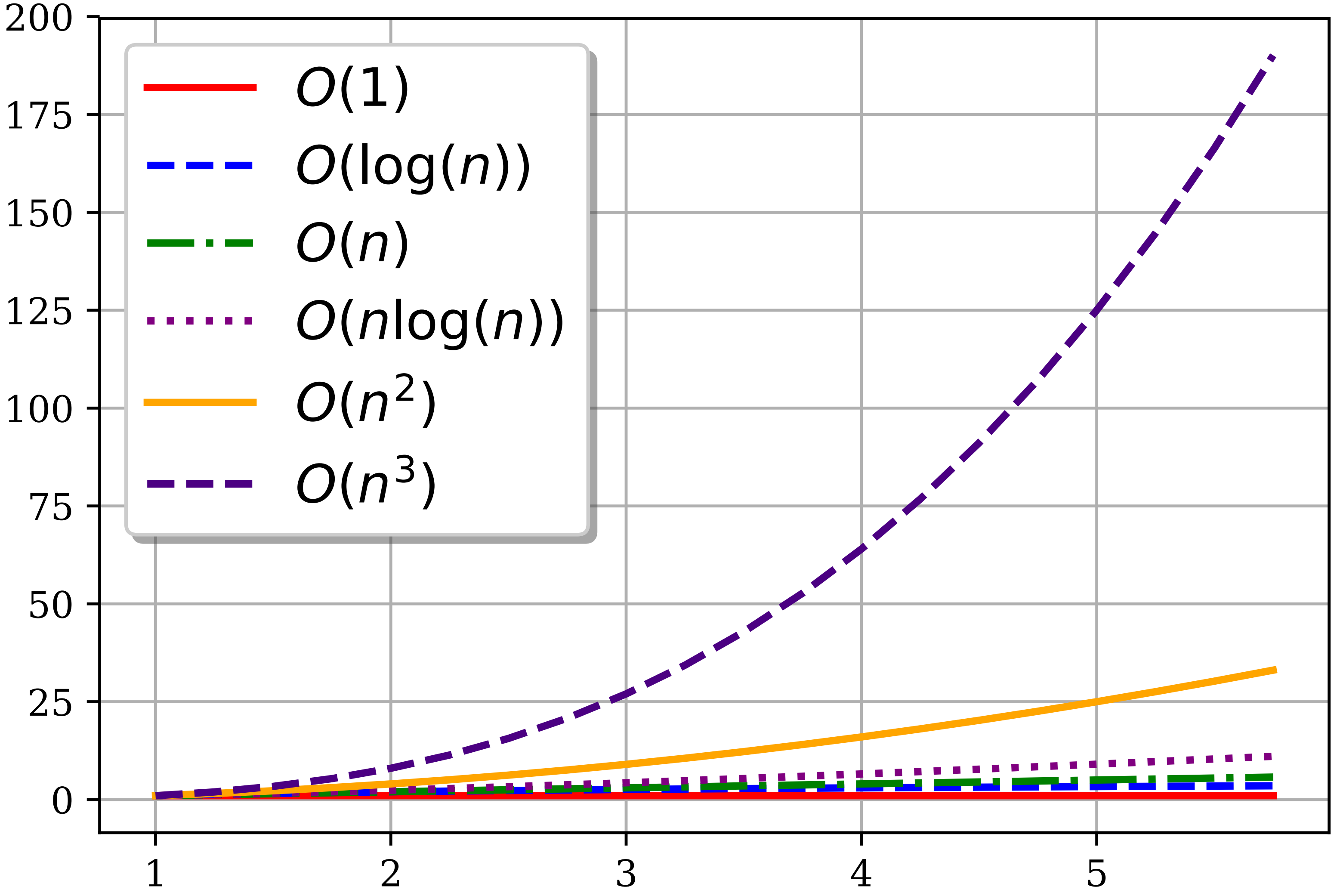
Comparing Algorithm Runtimes



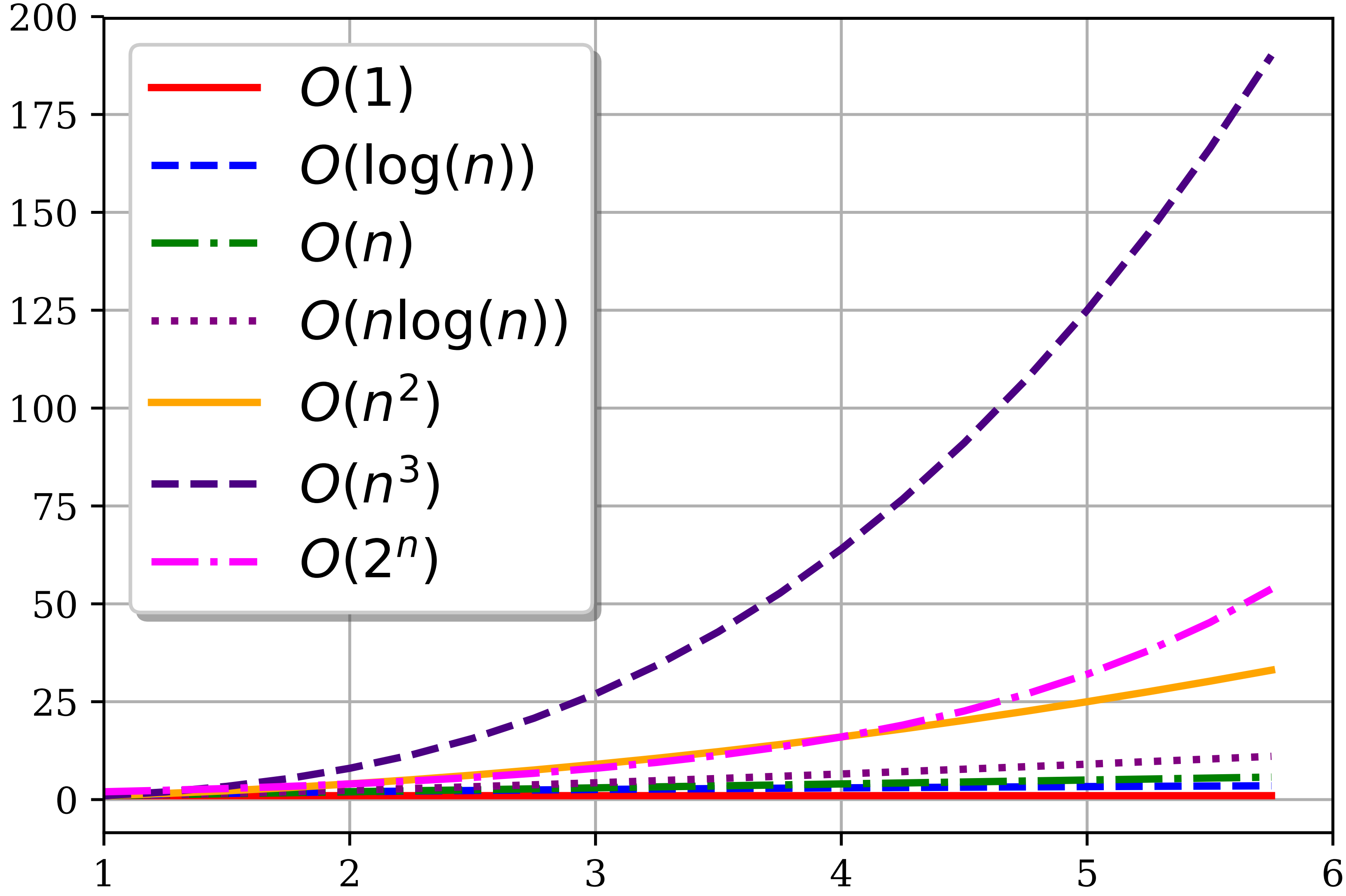
Comparing Algorithm Runtimes



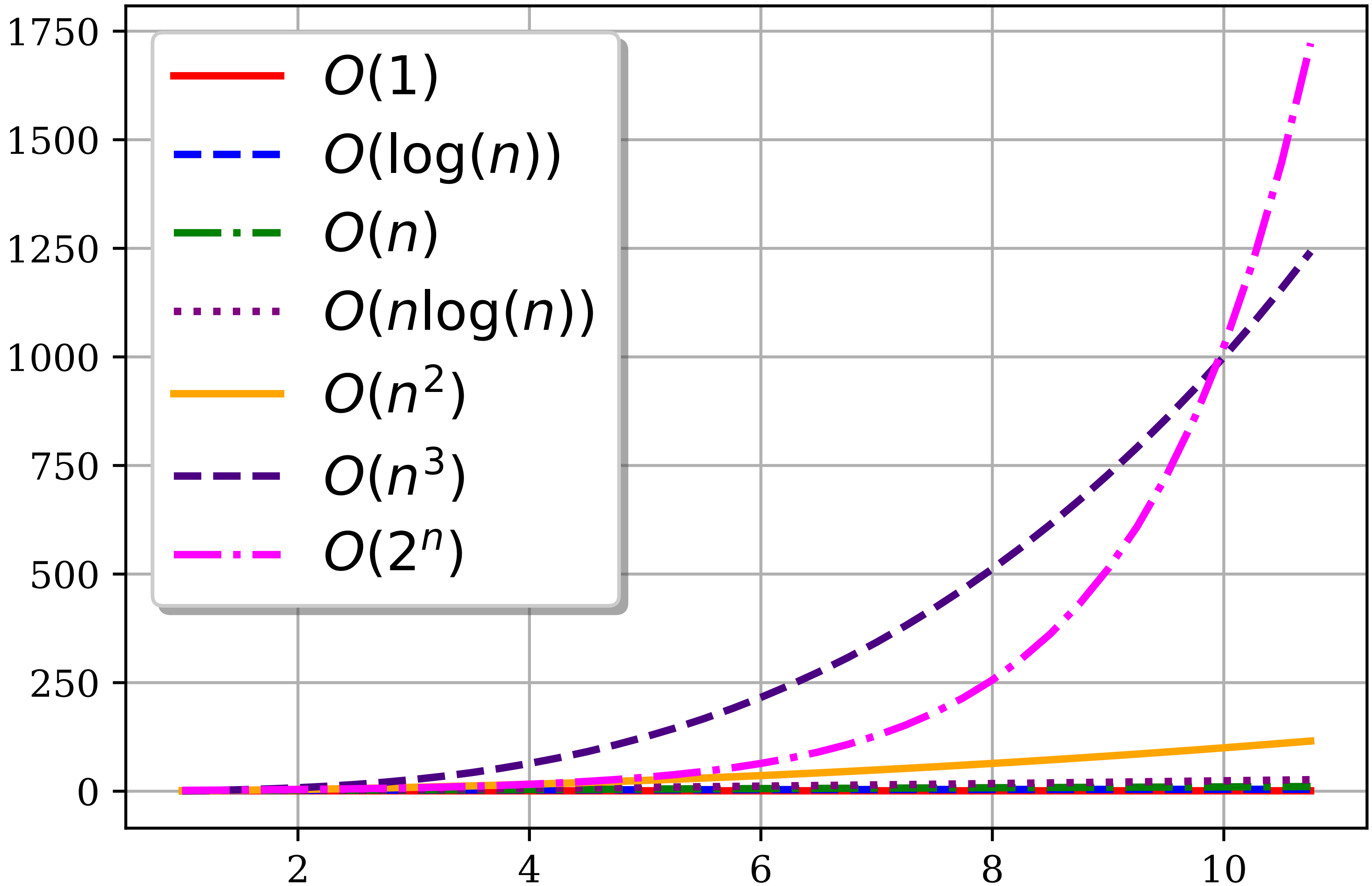
Comparing Algorithm Runtimes



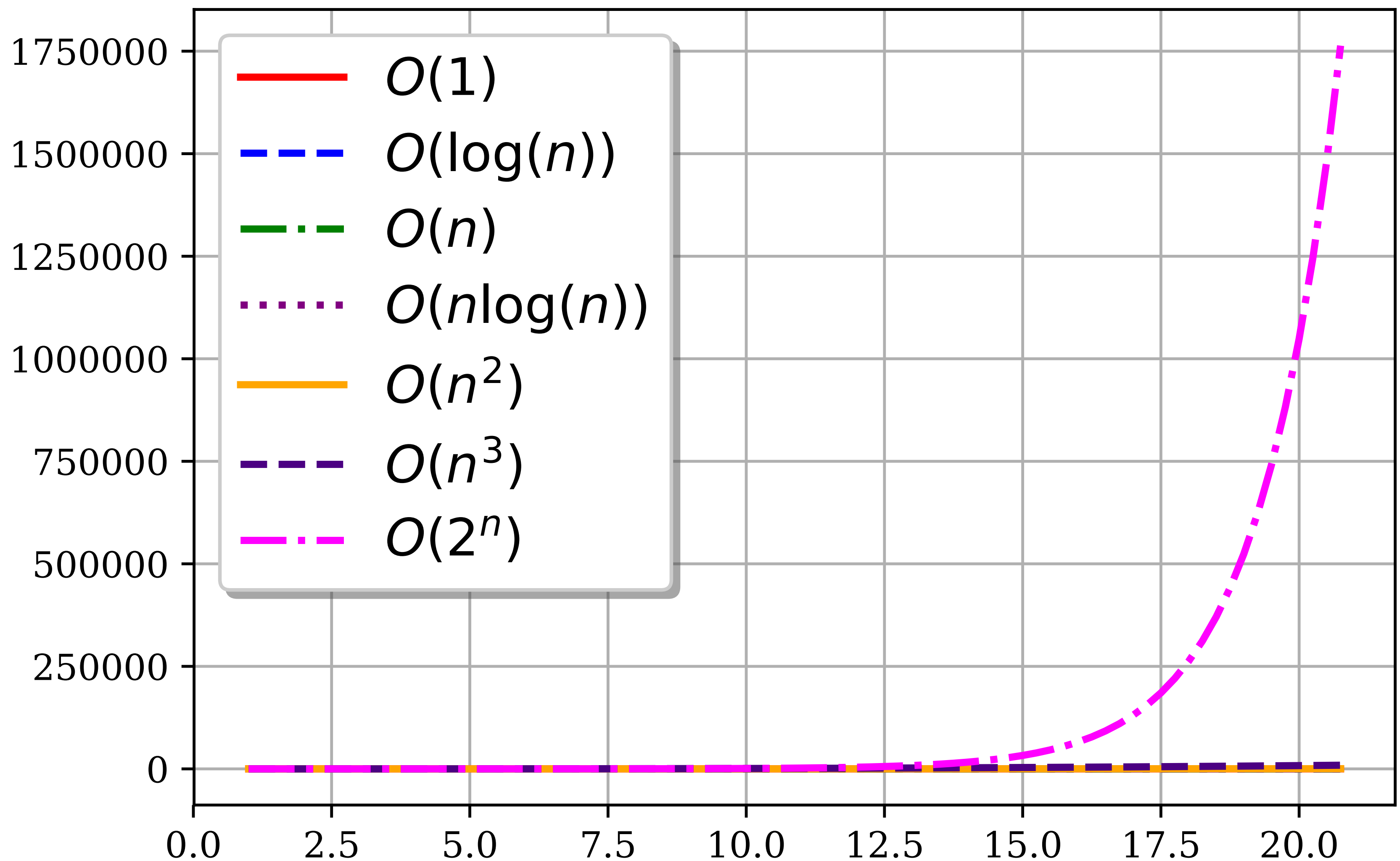
Comparing Algorithm Runtimes



Comparing Algorithm Runtimes



Comparing Algorithm Runtimes



Comparing Algorithm Runtimes

Computational Complexity	Name
$O(1)$	constant
$O(\log(n))$	logarithmic
$O(n)$	linear
$O(n \log(n))$	"n log n"
$O(n^2)$	quadratic
$O(n^3)$	cubic
$O(2^n)$	exponential
$O(n!)$	factorial
$O(n^n)$	superexponential

Complexity Classes

- An algorithm runs in **polynomial time** if its runtime is a polynomial function of the input size (e.g. $O(n^k)$ for some fixed constant k)
- The **class P** consists of all problems that can be solved in polynomial time

- A problem for which the answer is binary (e.g. yes/no) is called a **decision problem**
- The **class NP** contains all decision problems where 'yes' answers can be verified (proved) in polynomial time
- A problem is **NP-Hard** if given an O(1) oracle to solve it, every problem in NP can be solved in polynomial time (e.g. by reduction)
- A problem is **NP-Complete** if it belongs to both the classes NP and NP-Hard

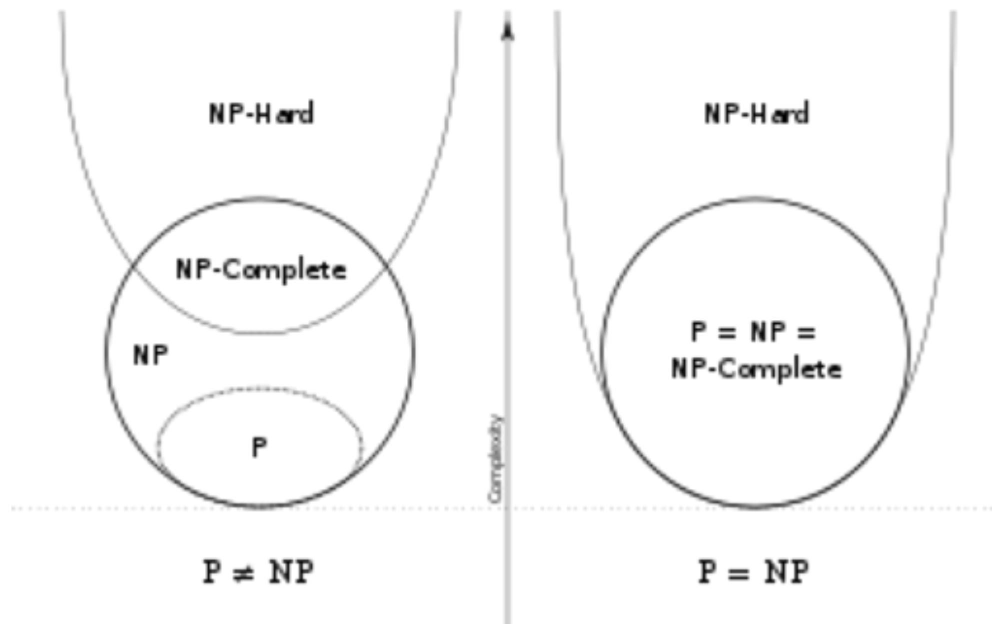


Figure from https://en.wikipedia.org/wiki/NP-completeness

Complexity Classes

- A problem for which the answer is a nonnegative integer is called a **counting problem**
- The **class #P** contains the counting problems that align to decision problems in NP
  - really this is the class of problems that count the number of accepting paths in a Turing machine that is nondeterministic and runs in polynomial time
- A problem is **#P-Hard** if given an O(1) oracle to solve it, every problem in #P can be solved in polynomial time (e.g. by reduction)
- A problem is **#P-Complete** if it belongs to both the classes #P and #P-Hard
- There are no known polytime algorithms for solving #P-Complete problems. If we found one it would imply that P = NP.

Examples of #P-Hard problems

- #SAT, i.e. how many satisfying solutions for a given SAT problem?
- How many solutions for a given DNF formula?
- How many solutions for a 2-SAT problem?
- How many perfect matchings for a bipartite graph?
- How many graph colorings (with k colors) for a given graph G?

Examples from https://en.wikipedia.org/wiki/%E2%99%AFP-complete

EXACT INFERENCE

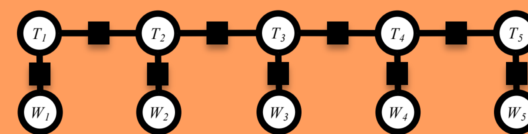
Exact Inference

1. Data

$$\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$$

2. Model

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$



3. Objective

$$\ell(\theta; \mathcal{D}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}^{(n)} \mid \boldsymbol{\theta})$$

5. Inference

1. Marginal Inference

$$p(\boldsymbol{x}_C) = \sum_{\boldsymbol{x}' : \boldsymbol{x}'_C = \boldsymbol{x}_C} p(\boldsymbol{x}' \mid \boldsymbol{\theta})$$

2. Partition Function

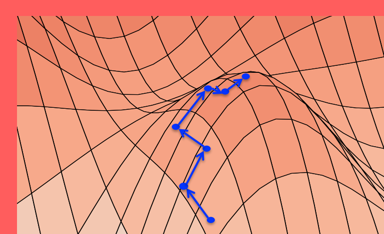
$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

3. MAP Inference

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\text{argmax}}\ p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

4. Learning

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}}\, \ell(\boldsymbol{\theta}; \mathcal{D})$$



5. Inference

Three Tasks:

1. Marginal Inference (#P-Hard)
Compute marginals of variables and cliques

$$p(x_i) = \sum_{\boldsymbol{x}' : x'_i = x_i} p(\boldsymbol{x}' \mid \boldsymbol{\theta}) \qquad \Bigg| \qquad p(\boldsymbol{x}_C) = \sum_{\boldsymbol{x}' : \boldsymbol{x}'_C = \boldsymbol{x}_C} p(\boldsymbol{x}' \mid \boldsymbol{\theta})$$

2. Partition Function (#P-Hard)
Compute the normalization constant

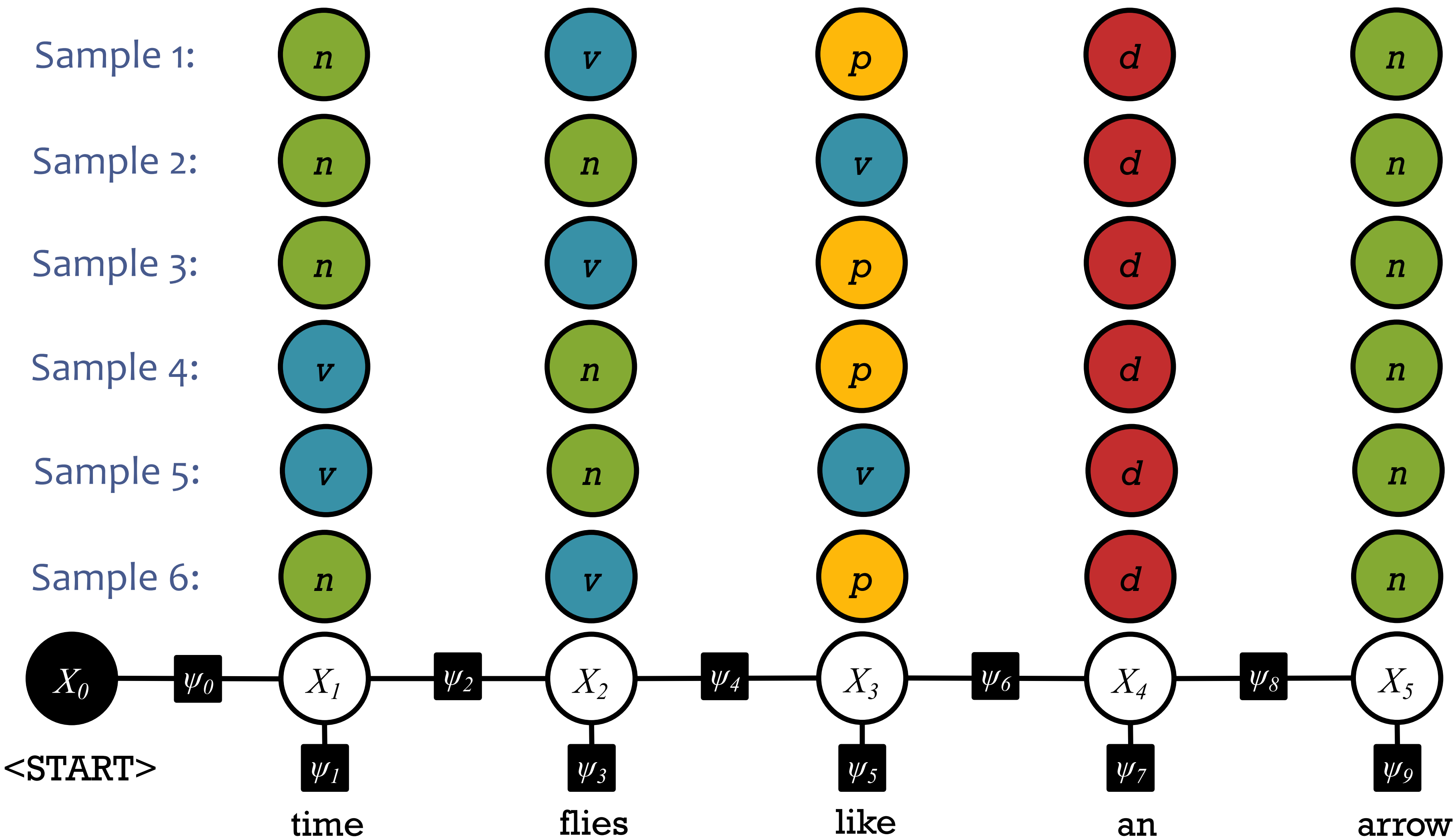
$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

3. MAP Inference (NP-Hard)
Compute variable assignment with highest probability

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \; p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

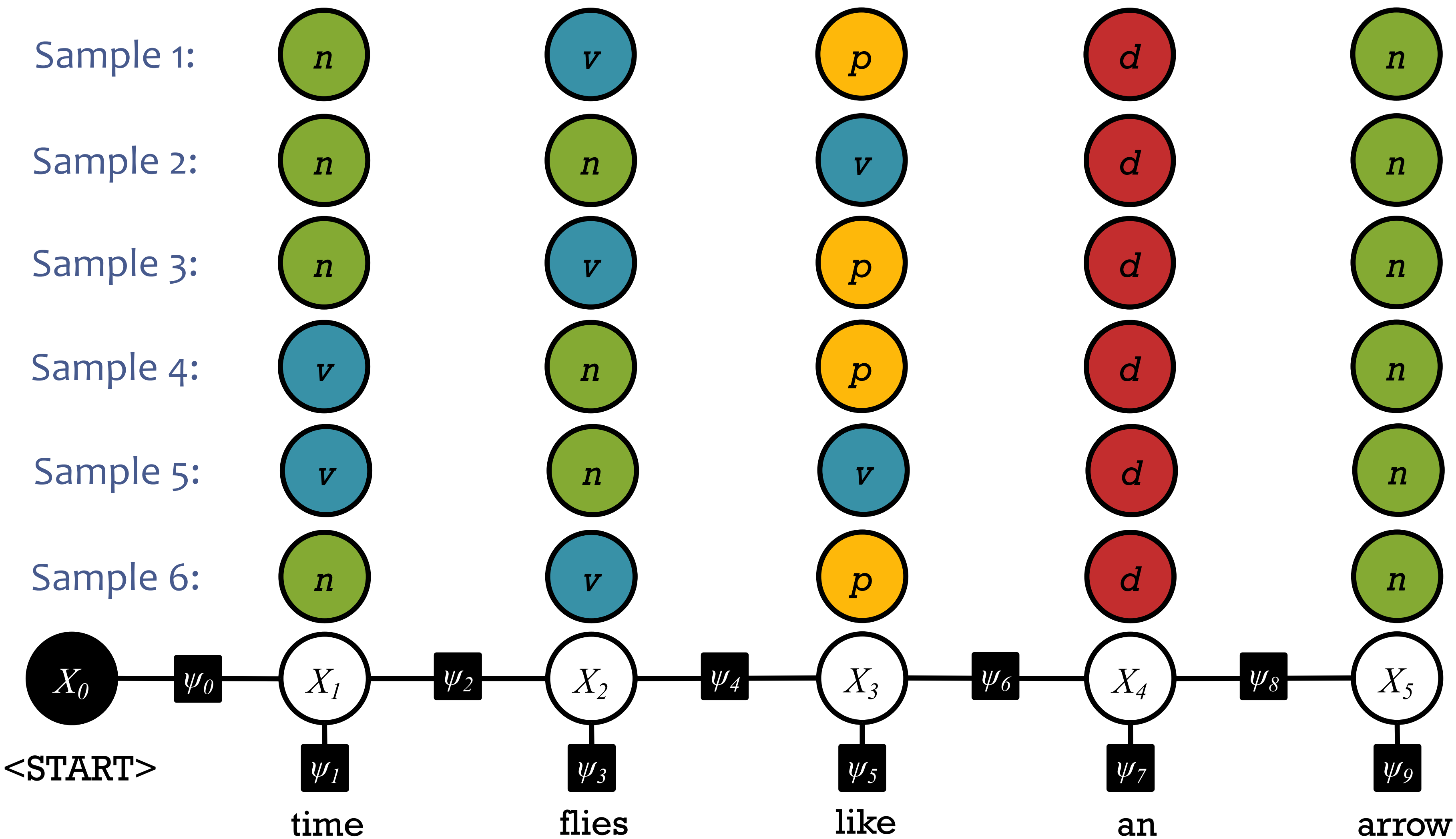
Marginals by Sampling on Factor Graph

Suppose we took many samples from the distribution over taggings: $p(\boldsymbol{x}) = \frac{1}{Z} \prod_{\alpha} \psi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}})$



Marginals by Sampling on Factor Graph

The marginal $p(X_i = x_i)$ gives the probability that variable X_i takes value x_i in a random sample



Marginals by Sampling on Factor Graph