Machine Learning Department
School of Computer Science
Carnegie Mellon University

Application: Factor Graphs

Matt Gormley
Lecture 10
Nov. 26, 2018

Reminders

- Homework C: Data Structures
  - Out: Mon, Nov. 26
  - Due: Mon, Dec. 3 at 11:59pm
- Quiz B: Computation; Programming & Efficiency
  - Wed, Dec. 5, in-class
  - Covers Lectures 7 – 12

Q&A

APPLICATION: EXACT INFERENCE IN GRAPHICAL MODELS

MOTIVATION: STRUCTURED PREDICTION

Structured Prediction

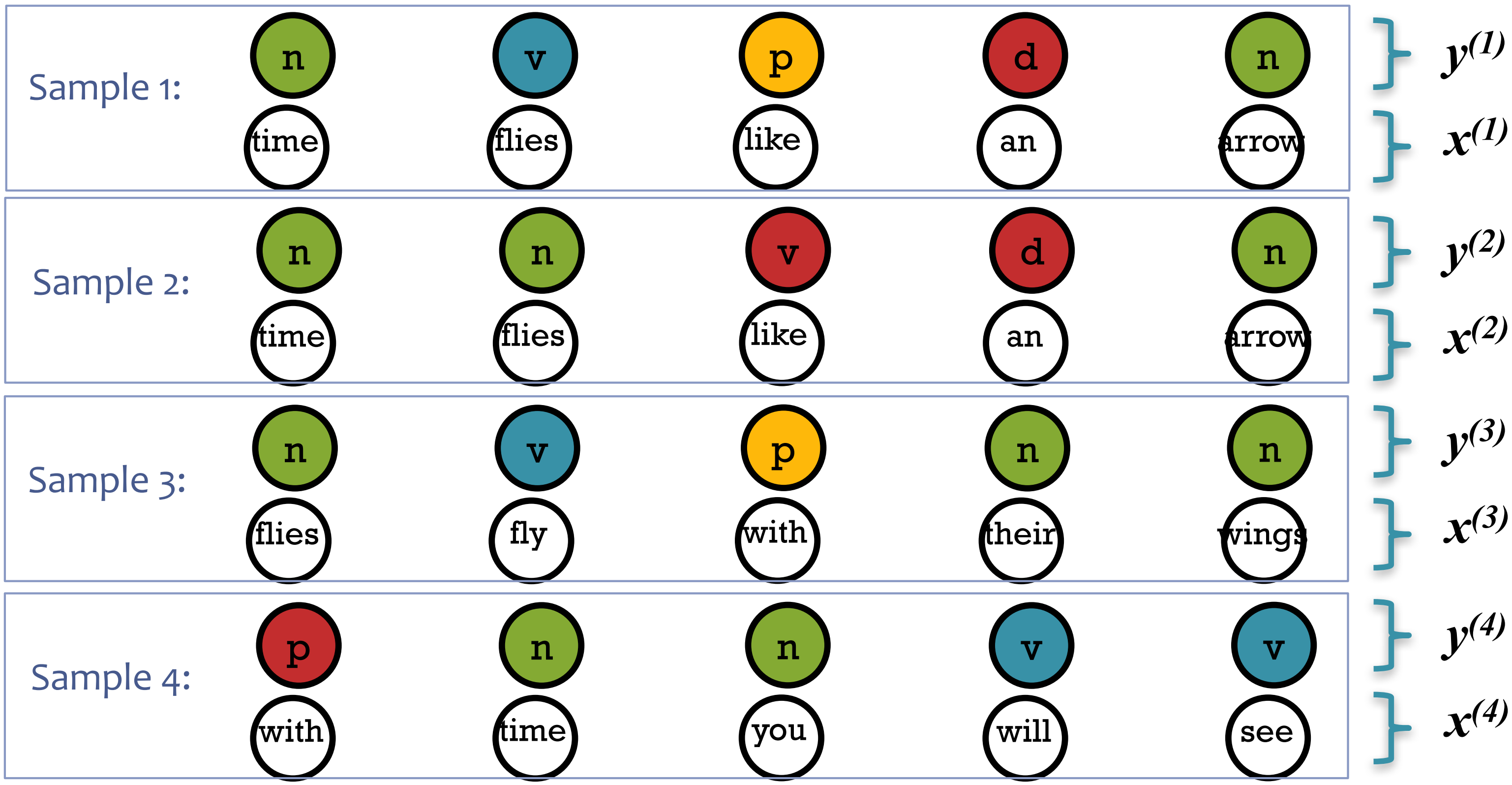
- Most of the models we've seen so far were for **classification**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \dots, x_K)$
  - Predict a (binary) **label:** y
- Many real-world problems require **structured prediction**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \dots, x_K)$
  - Predict a **structure:** $\boldsymbol{y} = (y_1, y_2, \dots, y_J)$
- Some *classification* problems benefit from **latent structure**

Structured Prediction Examples

- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Word alignment
  - Congressional voting
- **Examples of latent structure**
  - Object recognition

Dataset for Supervised Part-of-Speech (POS) Tagging

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



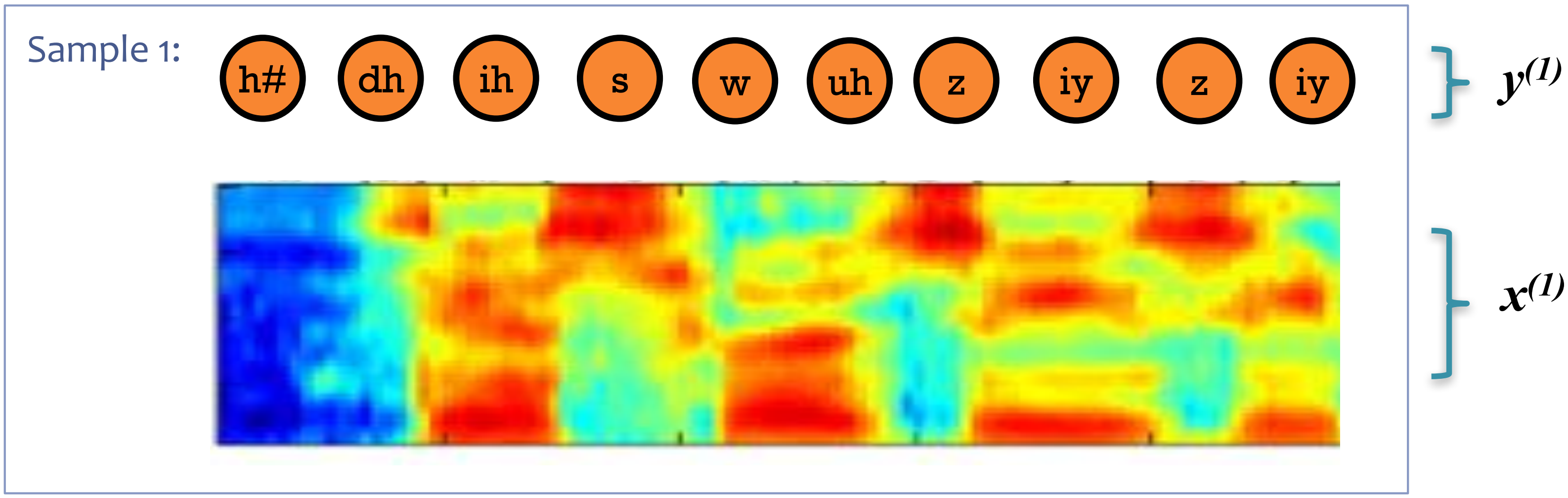
Dataset for Supervised Handwriting Recognition

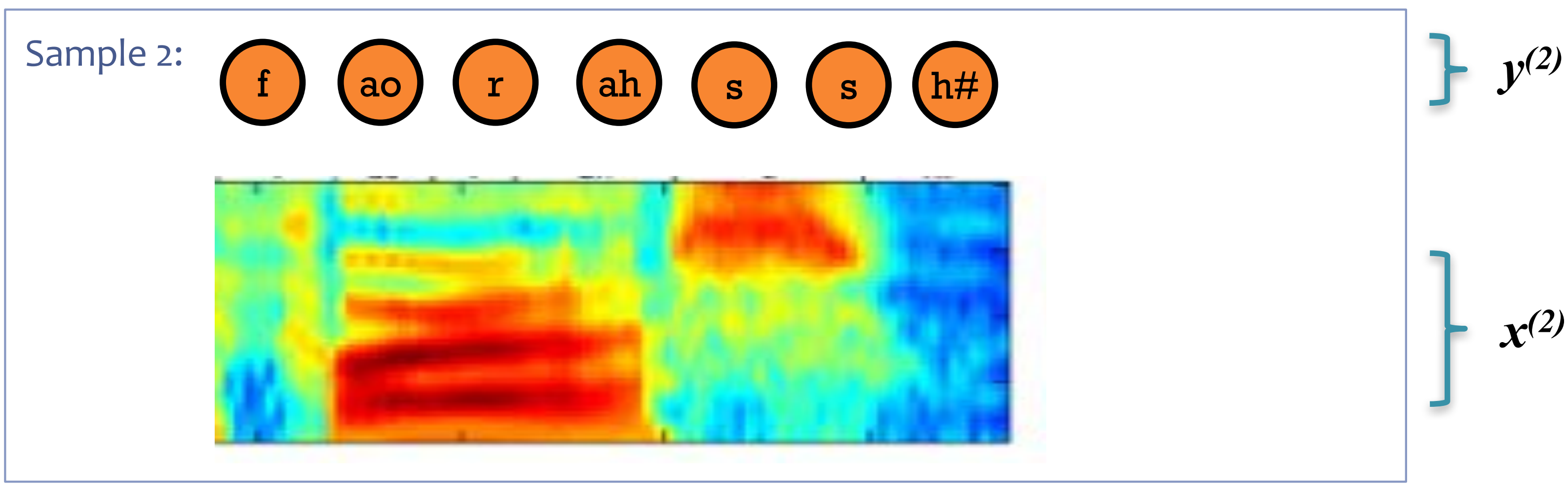
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Figures from (Chatzis & Demiris, 2013)

Dataset for Supervised Phoneme (Speech) Recognition

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



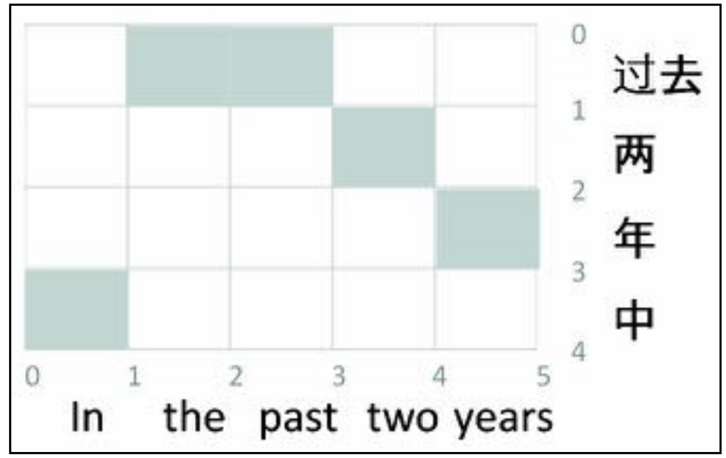


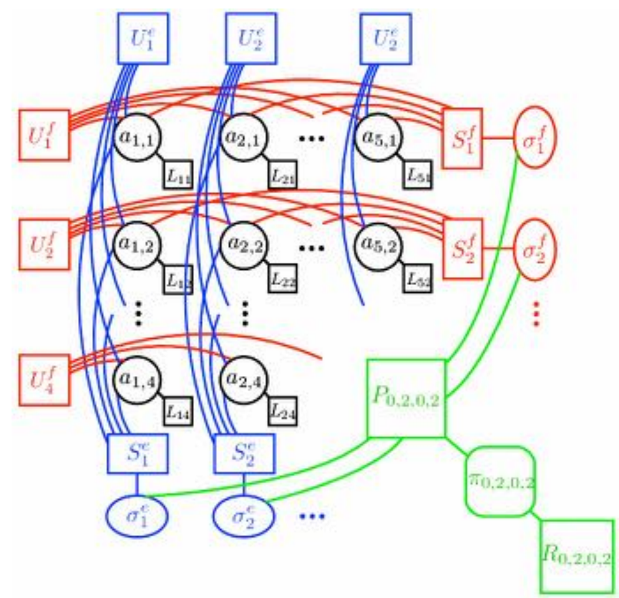
Figures from (Jansen & Niyogi, 2013)

Application:

Word Alignment / Phrase Extraction

- **Variables (boolean):**
  - For each (Chinese phrase, English phrase) pair, are they linked?
- **Interactions:**
  - Word fertilities
  - Few “jumps” (discontinuities)
  - Syntactic reorderings
  - “ITG contraint” on alignment
  - Phrases are disjoint (?)

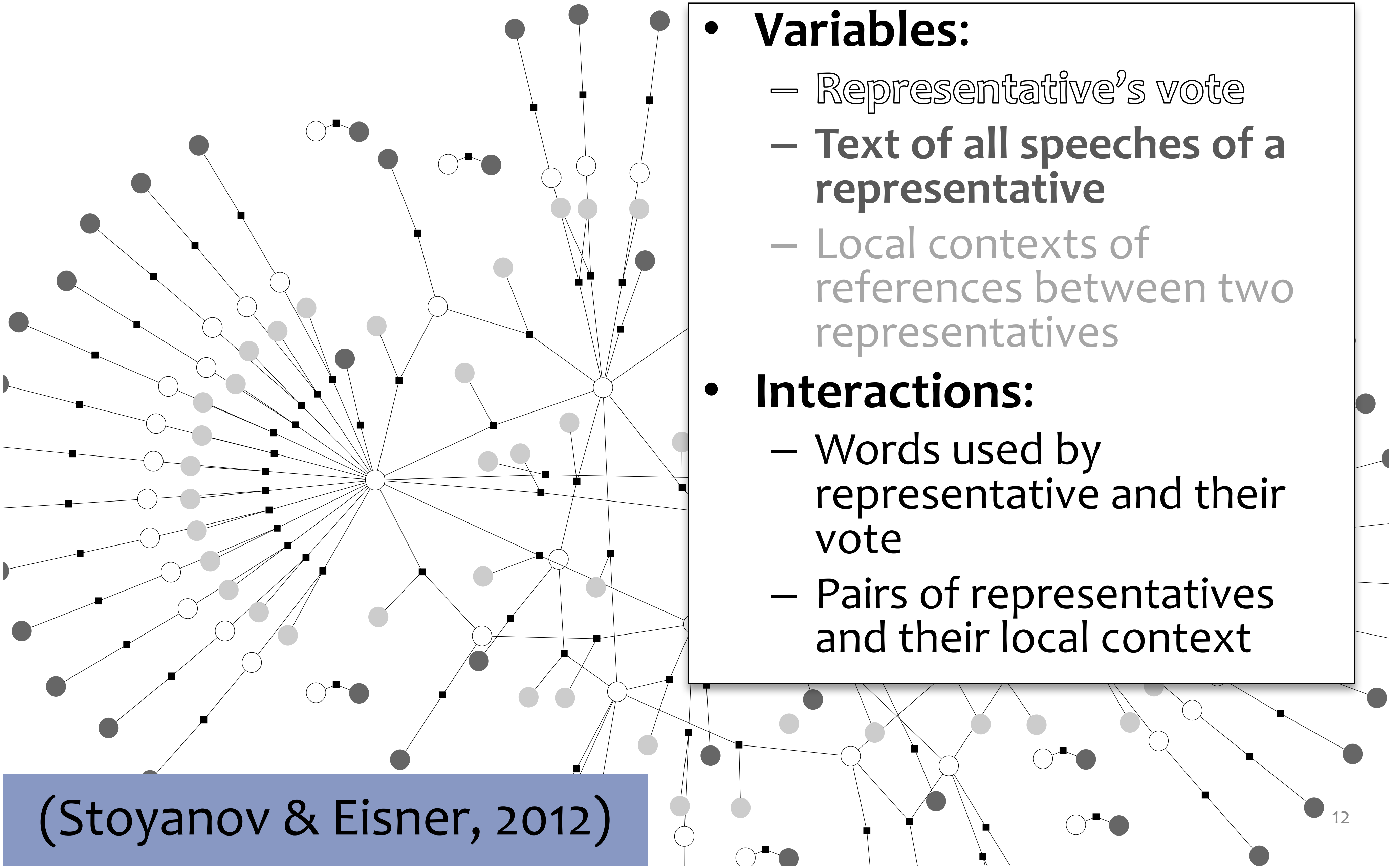




(Burkett & Klein, 2012)

Application:

Congressional Voting



- **Variables:**
  - Representative's vote
  - **Text of all speeches of a representative**
  - Local contexts of references between two representatives
- **Interactions:**
  - Words used by representative and their vote
  - Pairs of representatives and their local context

(Stoyanov & Eisner, 2012)

Structured Prediction Examples

- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Word alignment
  - Congressional voting
- **Examples of latent structure**
  - Object recognition

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

$\boldsymbol{x}^{(1)}$

pigeon $\}\ y^{(1)}$

$\boldsymbol{x}^{(2)}$

rhinoceros $\}\ y^{(2)}$

$\boldsymbol{x}^{(3)}$

leopard $\}\ y^{(3)}$

$\boldsymbol{x}^{(4)}$

llama $\}\ y^{(4)}$

Case Study: Object Recognition

Data consists of images x and labels y.

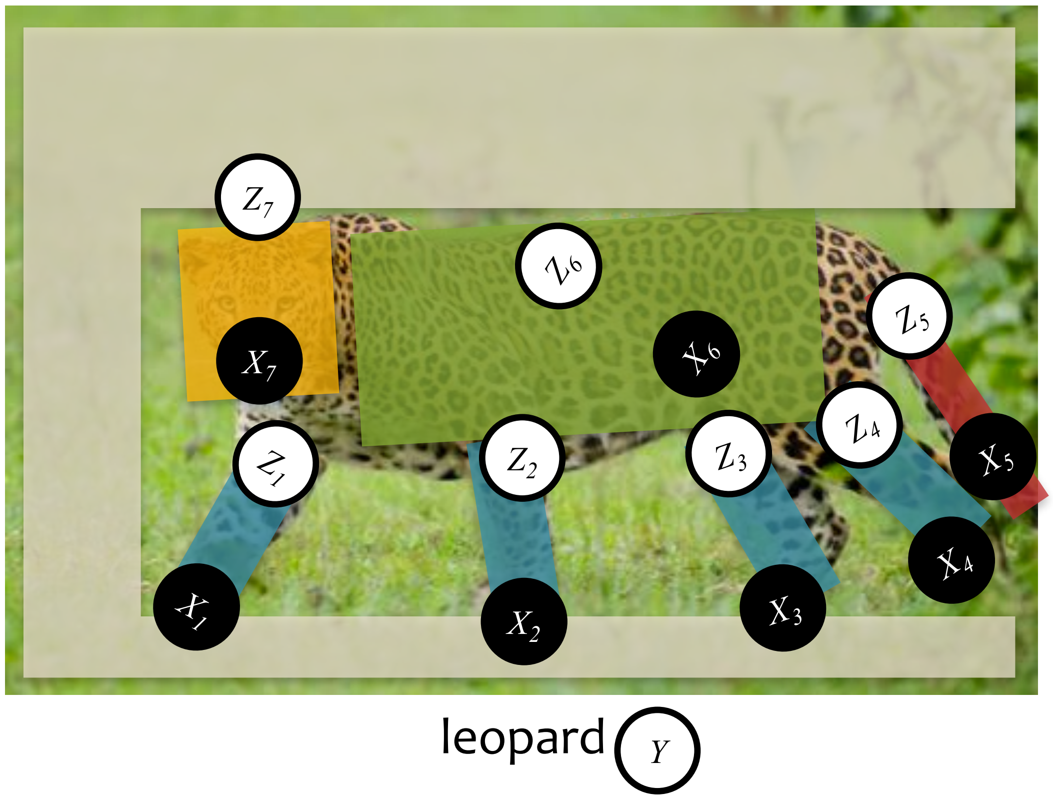
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

leopard

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

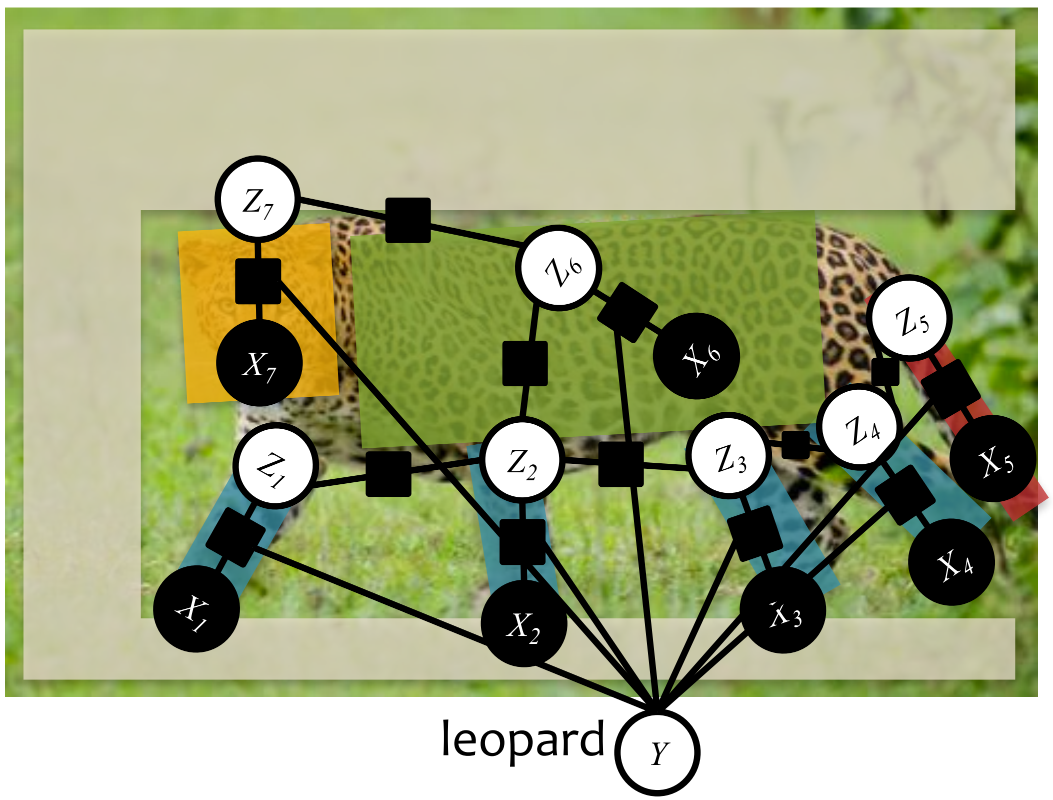
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Structured Prediction

Preview of challenges to come…

- Consider the task of finding the **most probable assignment** to the output

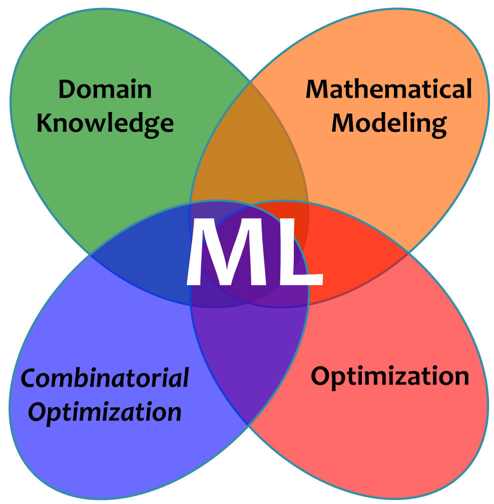
Classification	Structured Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y\|\mathbf{x})$	$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, p(\mathbf{y}\|\mathbf{x})$
where $y \in \{+1, -1\}$	where $\mathbf{y} \in \mathcal{Y}$
	and $\|\mathcal{Y}\|$ is very large

Machine Learning

The **data** inspires the structures we want to predict

Our **model** defines a score for each structure

It also tells us what to optimize



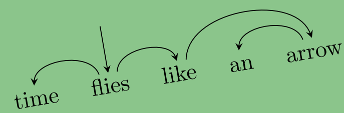
Inference finds {best structure, marginals, partition function} for a new observation

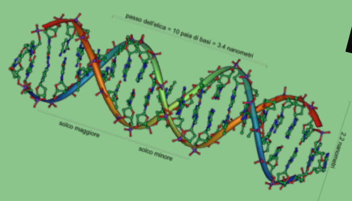
Learning tunes the parameters of the model

(**Inference** is usually called as a subroutine in learning)

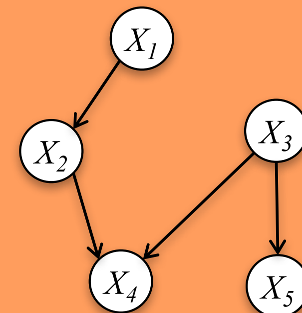
Machine Learning

Data

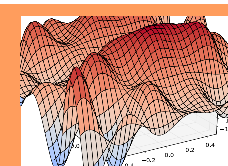




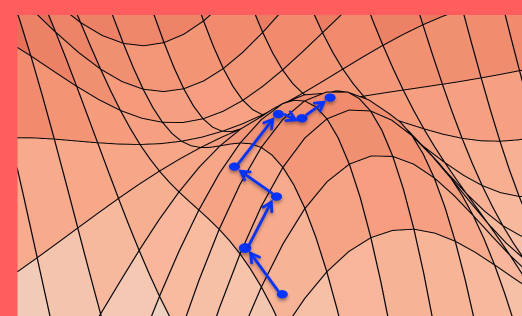
Model



Objective



Learning



Inference

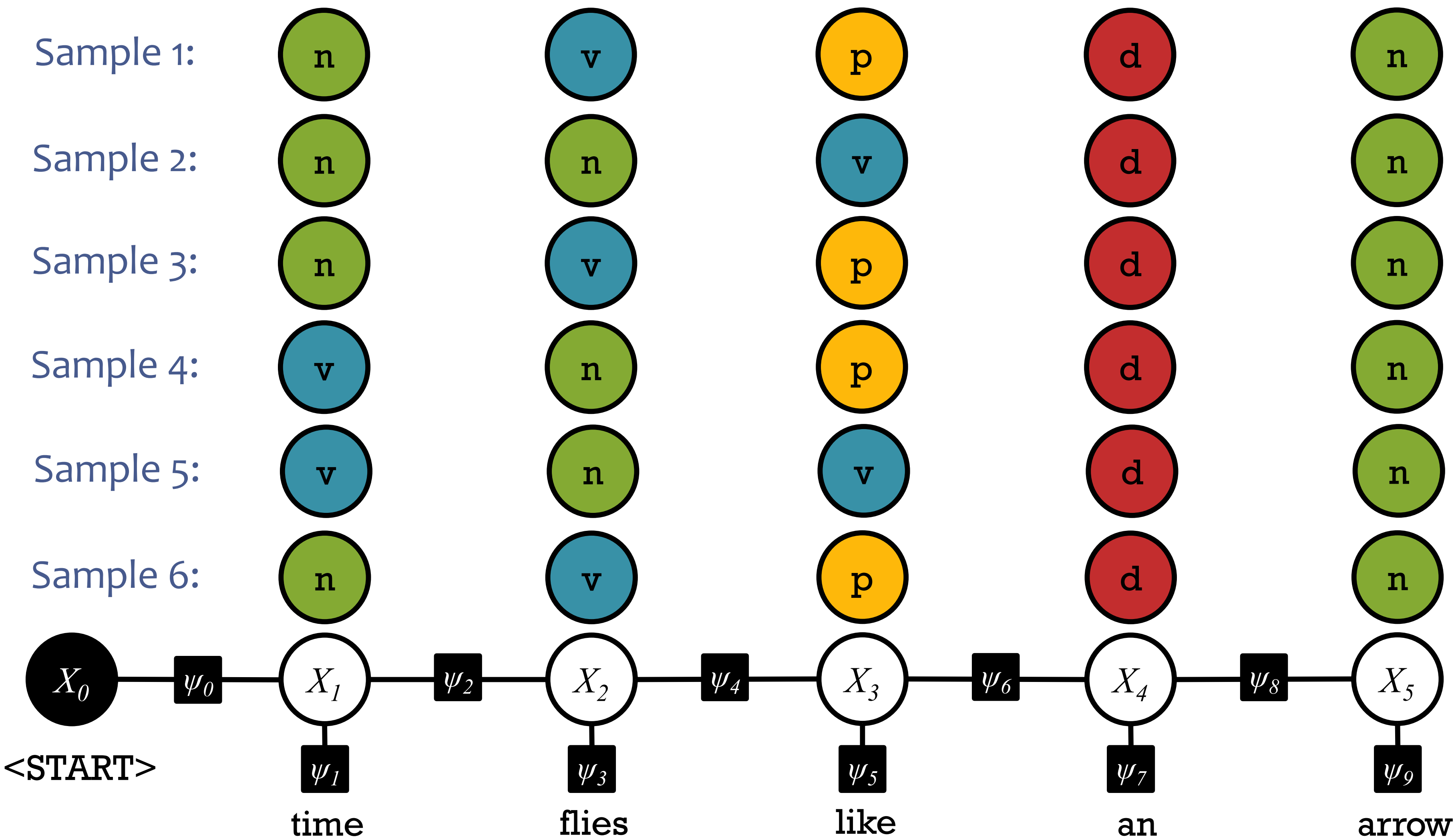
(**Inference** is usually called as a subroutine in learning)

Representation of both directed and undirected graphical models

FACTOR GRAPHS

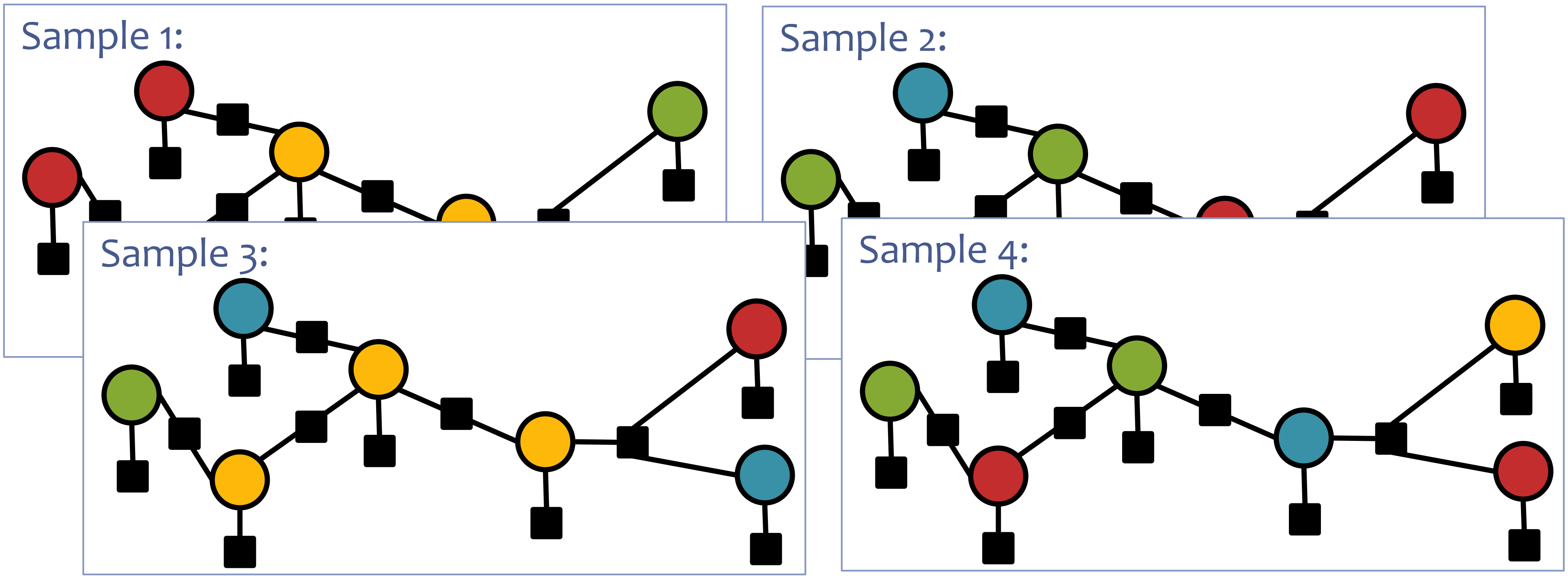
Sampling from a Joint Distribution

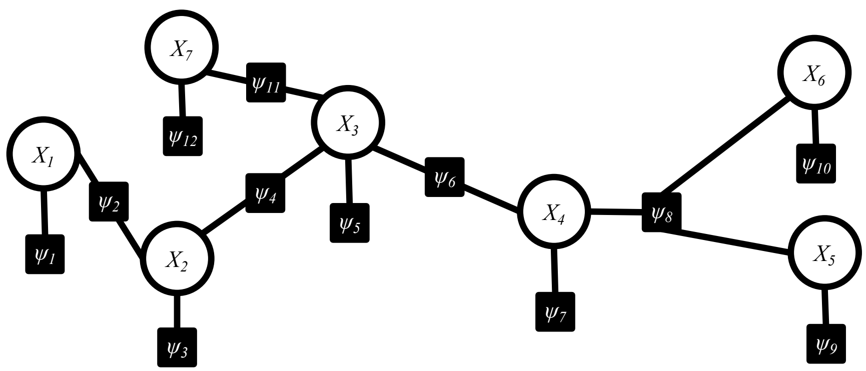
A **joint distribution** defines a probability $p(\boldsymbol{x})$ for each assignment of values $\boldsymbol{x}$ to variables $\boldsymbol{X}$. This gives the **proportion** of samples that will equal $\boldsymbol{x}$.



Sampling from a Joint Distribution

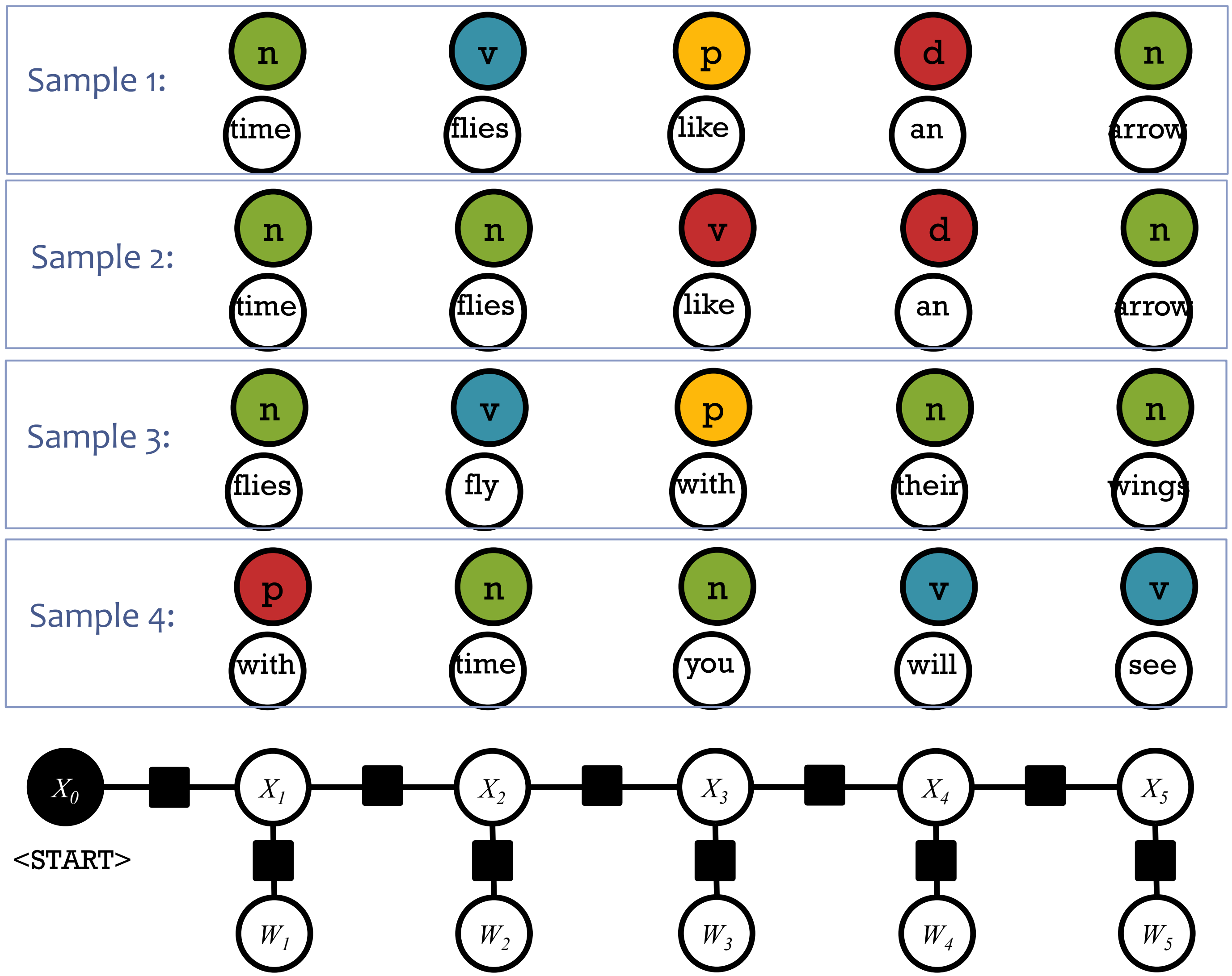
A **joint distribution** defines a probability $p(\boldsymbol{x})$ for each assignment of values $\boldsymbol{x}$ to variables $\boldsymbol{X}$. This gives the **proportion** of samples that will equal $\boldsymbol{x}$.





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 $\boldsymbol{x}$.



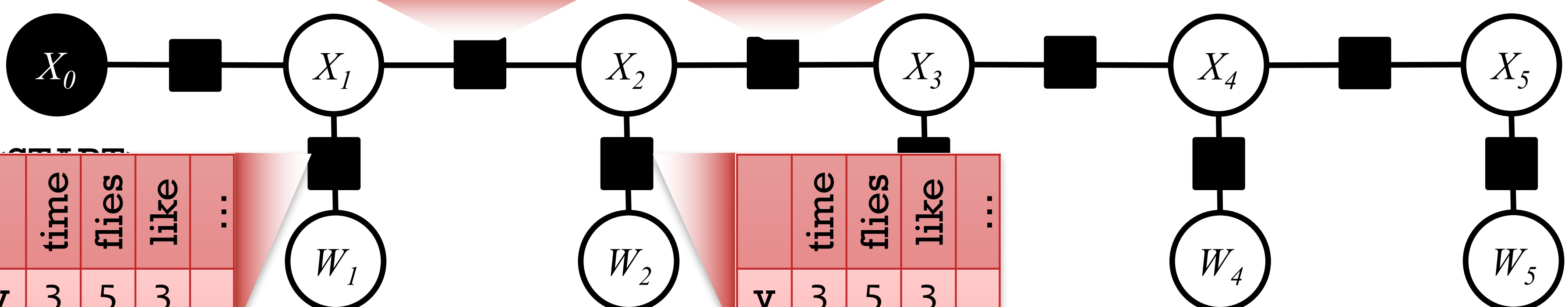
Factors have local opinions (≥ 0)

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

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

	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	

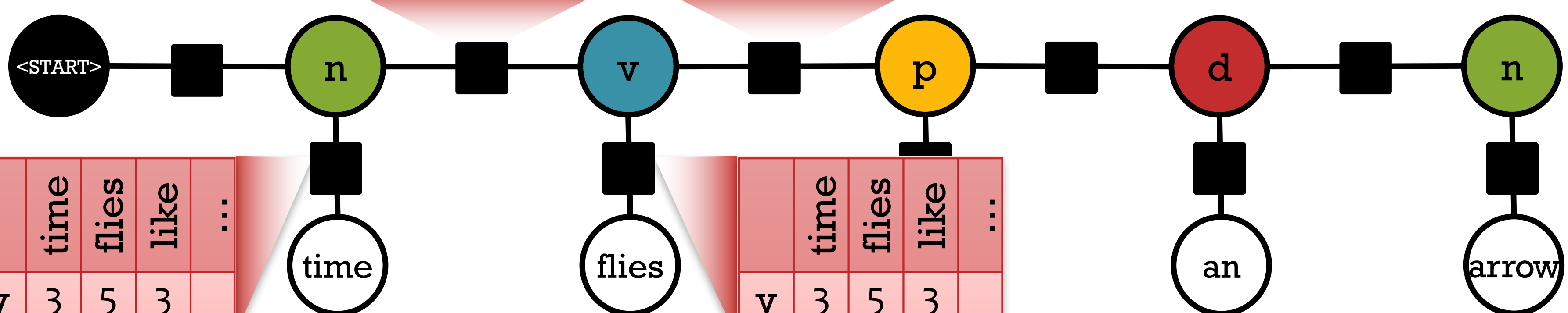
Factors have local opinions (≥ 0)

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

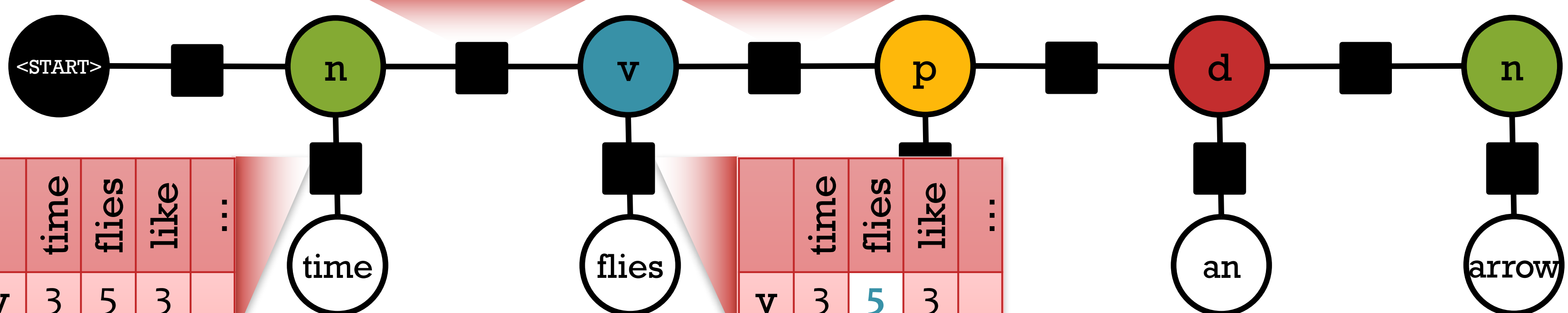
Each black box looks at *some* of the tags X_i and words W_i

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

	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

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	

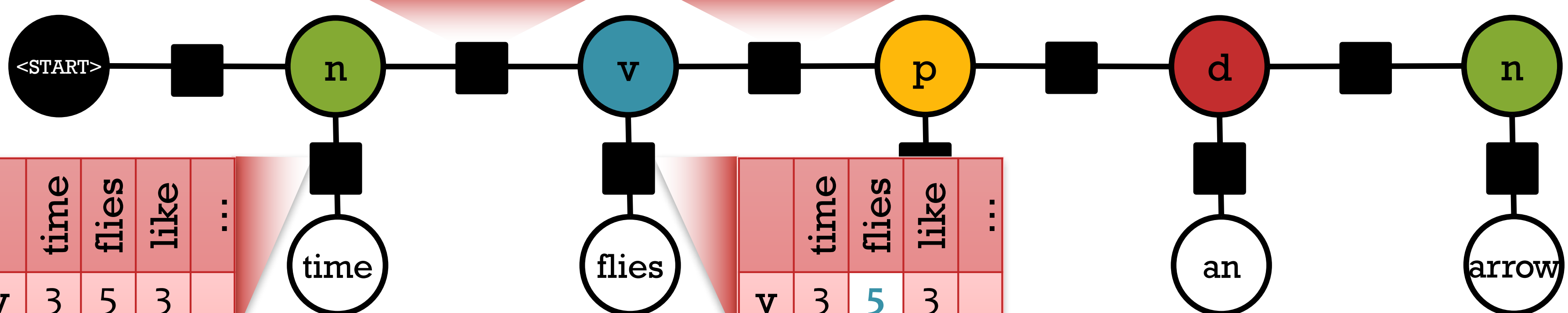
Markov Random Field (MRF)

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

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

	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



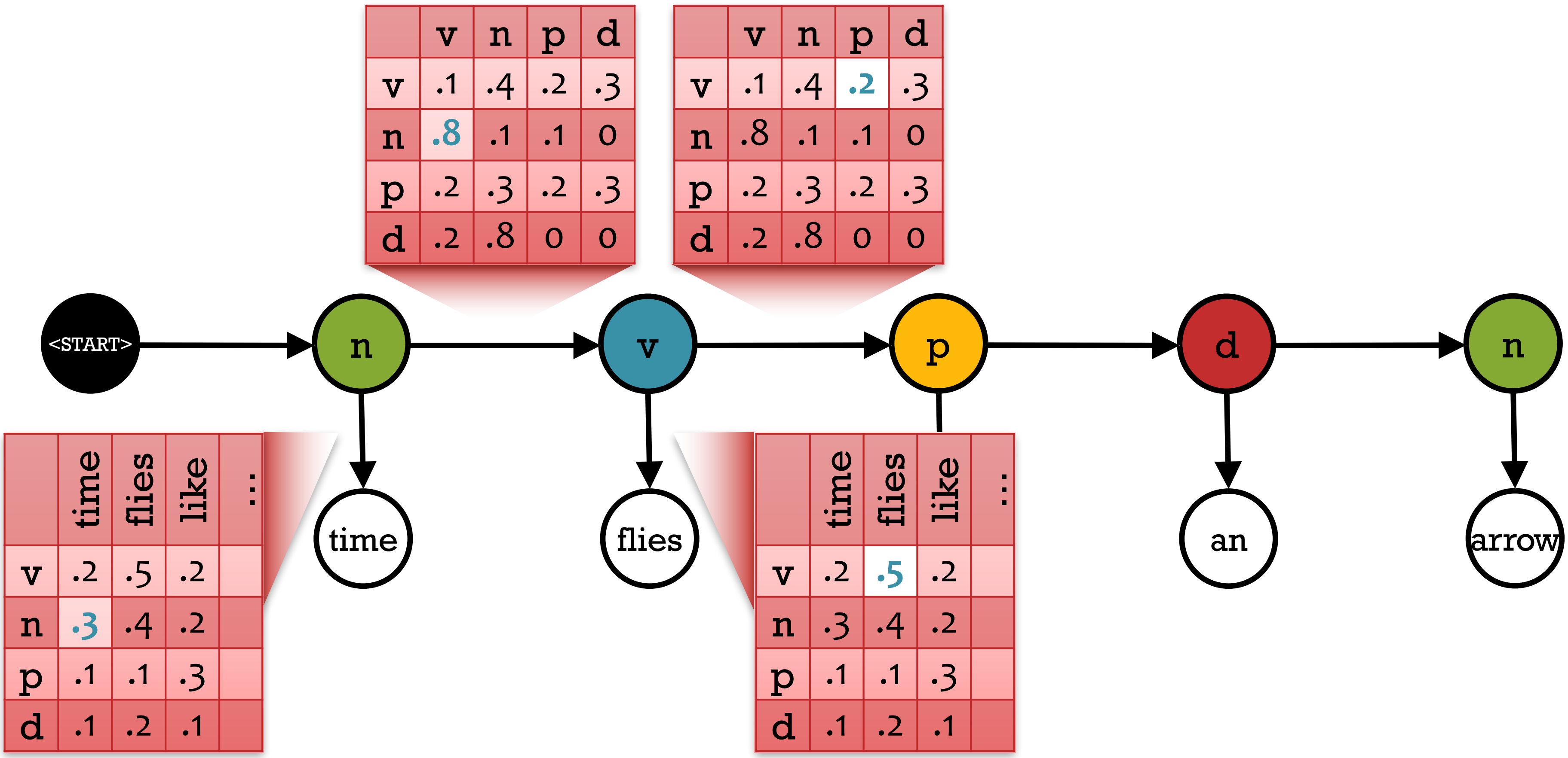
	time	flies	like	…
v	3	5	3	
n	4	5	2	
p	0.1	0.1	3	
d	0.1	0.2	0.1	

	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	

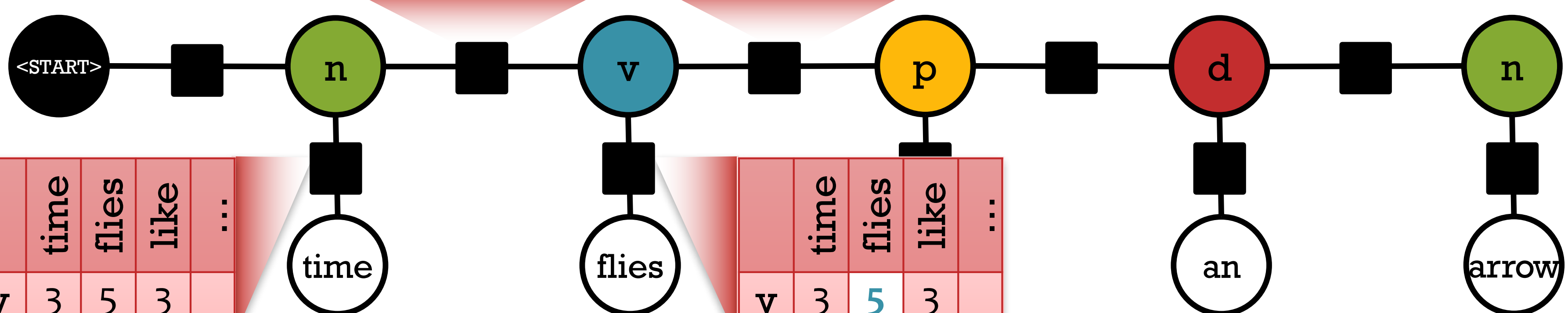
Markov Random Field (MRF)

Joint distribution over tags X_i <u>and</u> 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.

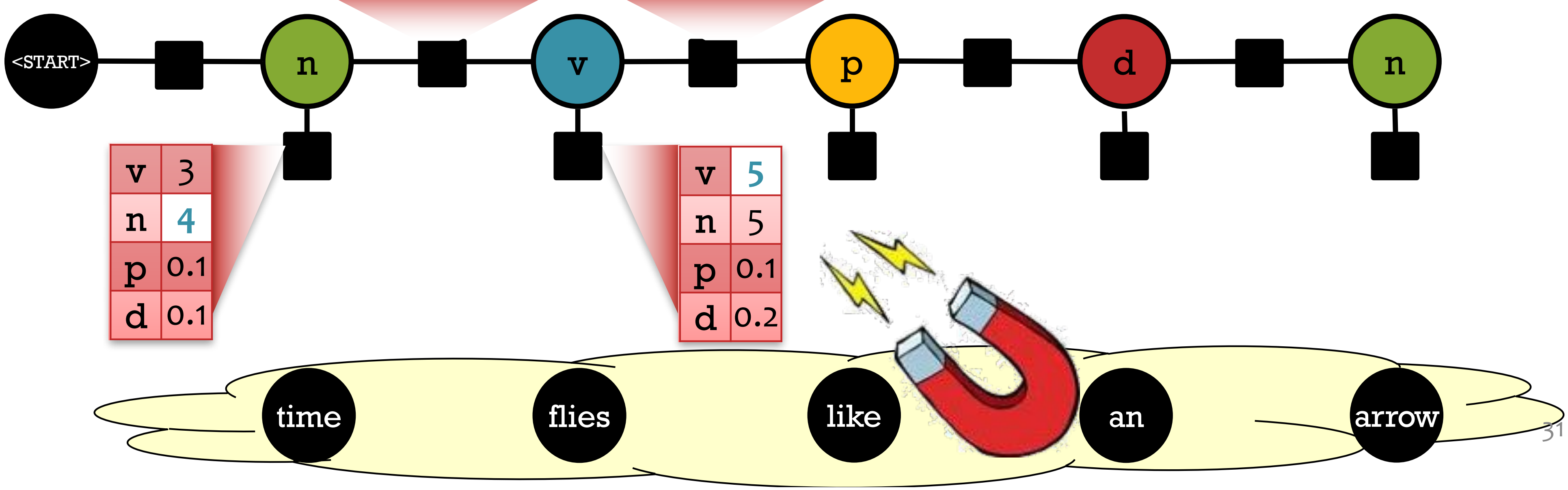
$$p(\text{n, v, p, d, n} \mid \text{time, flies, like, an, arrow}) = \frac{1}{Z}(4 * 8 * 5 * 3 * \ldots)$$

	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:

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

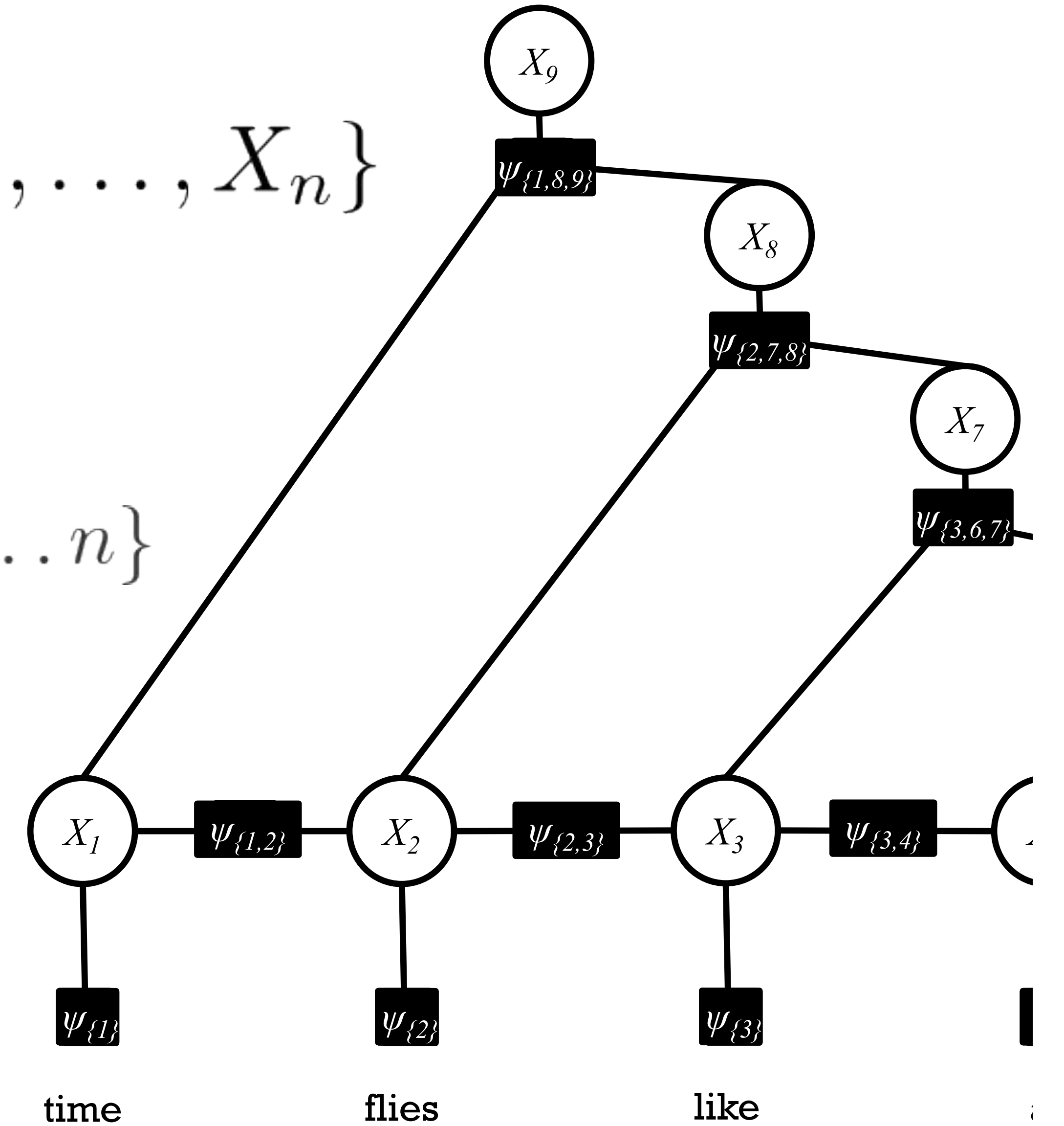
- Factors:

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

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

Joint Distribution

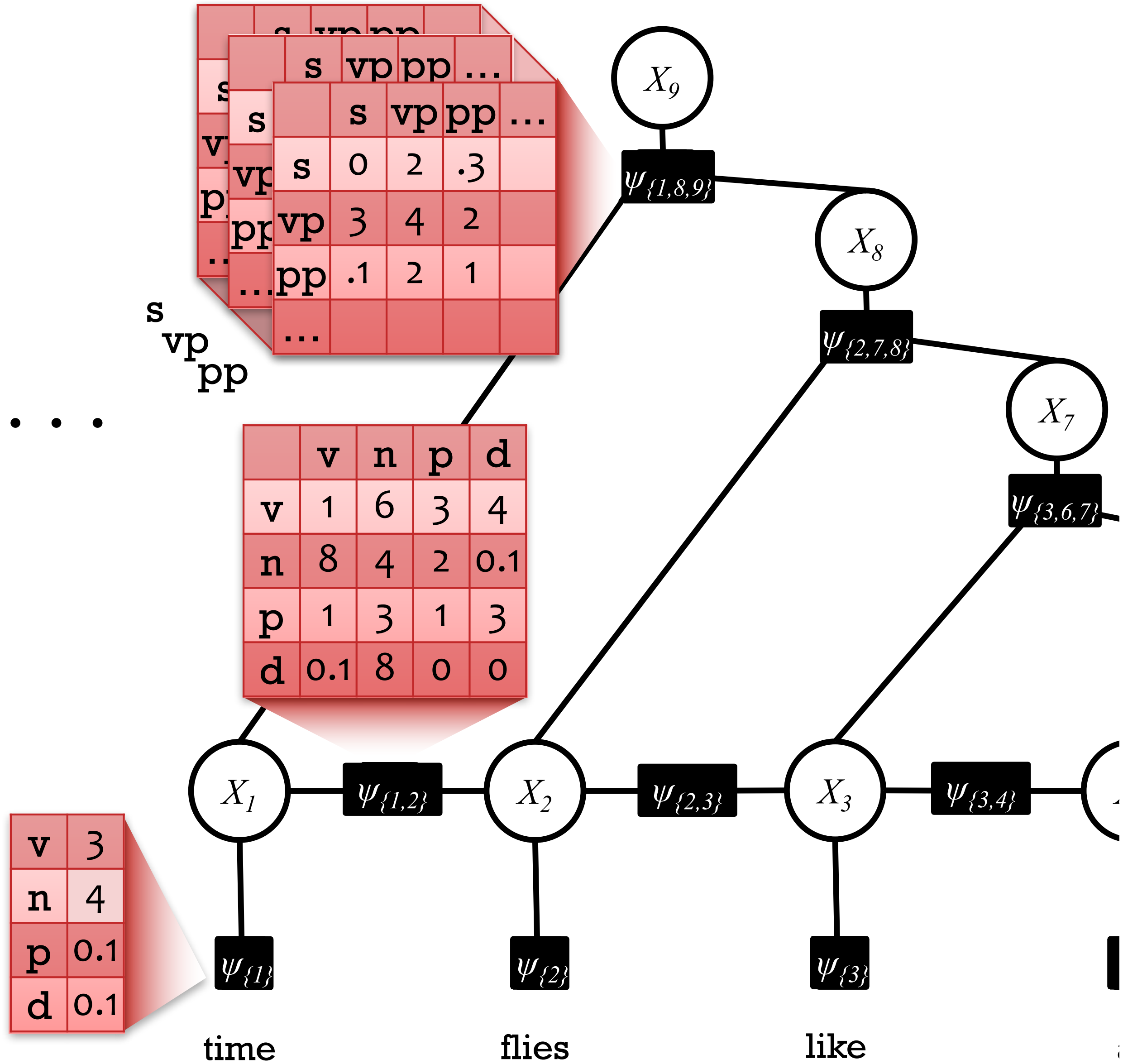
$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_\alpha \psi_\alpha(\boldsymbol{x}_{\boldsymbol{\alpha}})$$



Factors are Tensors

- Factors:

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



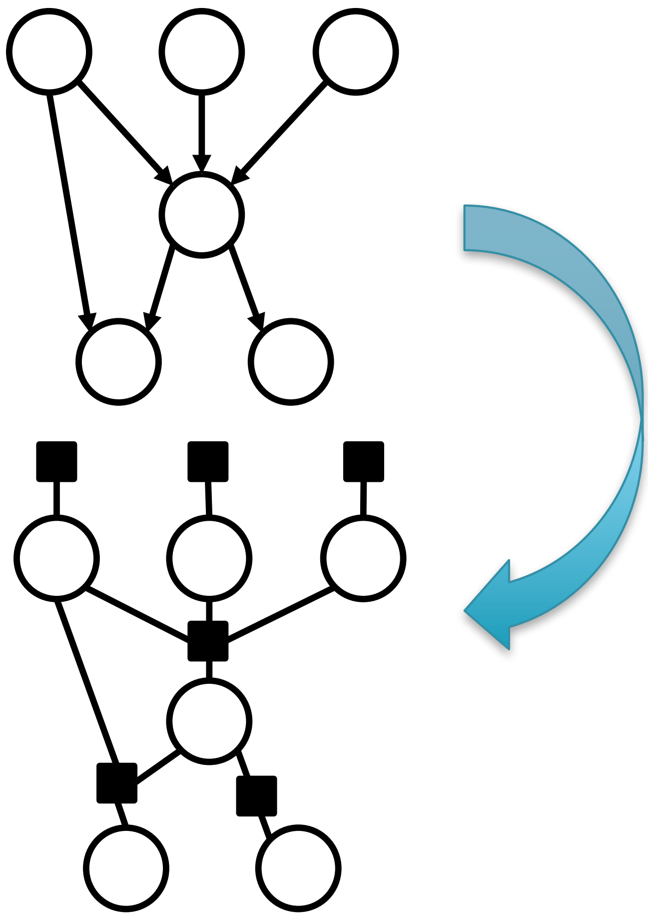
	s	vp	pp	…
s	0	2	.3	
vp	3	4	2	
pp	.1	2	1	
…				

	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

Converting to Factor Graphs

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



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