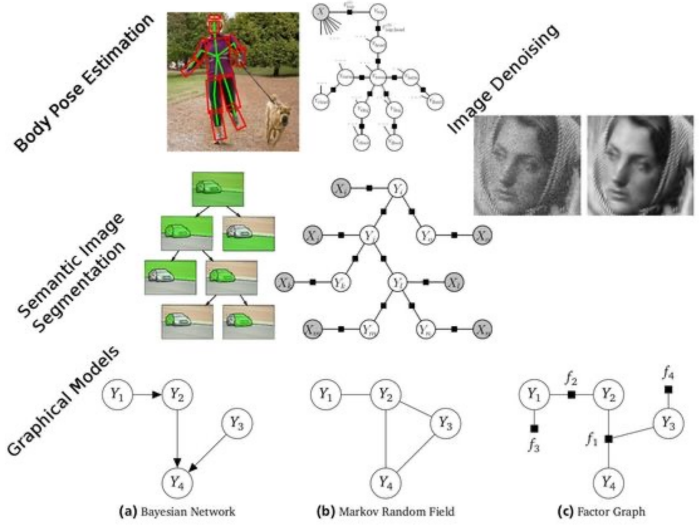
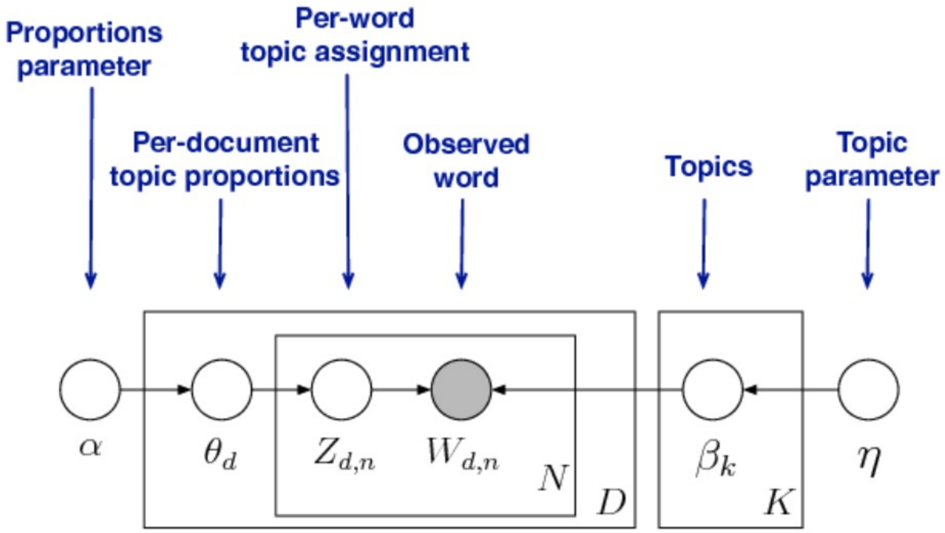
[illegible]G[illegible]; H[illegible]
C[illegible] F[illegible]

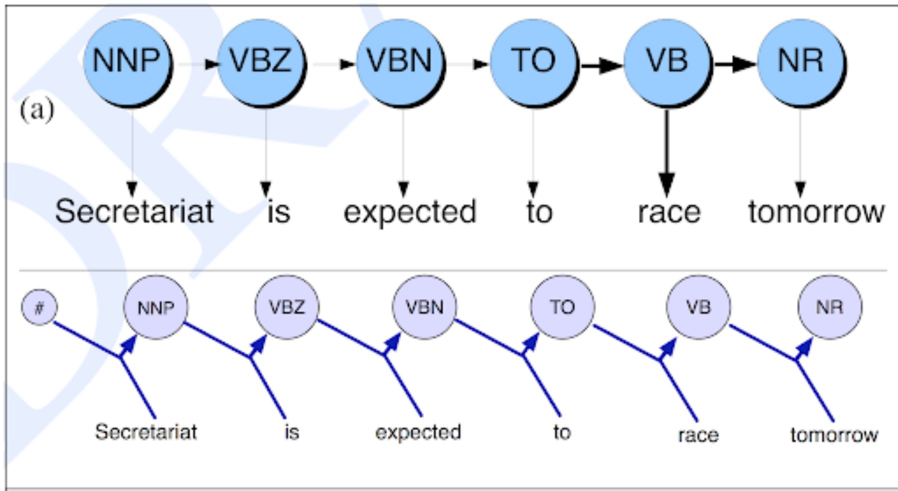
आई आई टी हैदराबाद
IIT Hyderabad

[illegible] G[illegible]

- [illegible]: [illegible] ([illegible], [illegible], [illegible])
  - They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models.
  - Insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph.

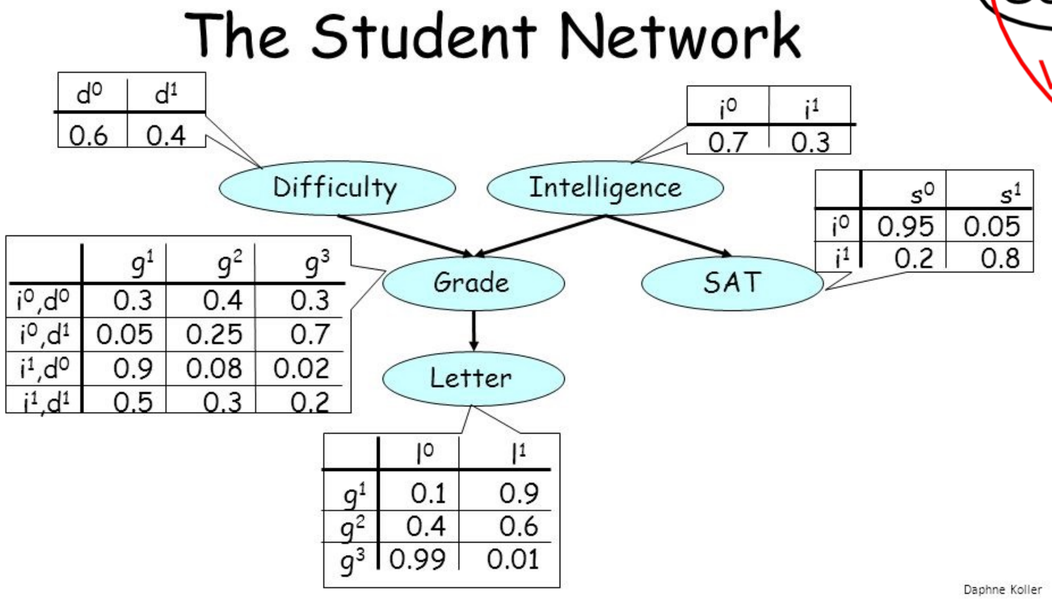






[illegible] G [illegible]

- A graph comprises nodes (also called vertices) connected by links (also known as edges or arcs). In a probabilistic graphical model, each node represents a random variable (or group of random variables), and the links express probabilistic relationships between these variables.
- Bayesian networks, also known as directed graphical models
- Markov random fields, also known as undirected graphical models



The Student Network

d^0	d^1
0.6	0.4

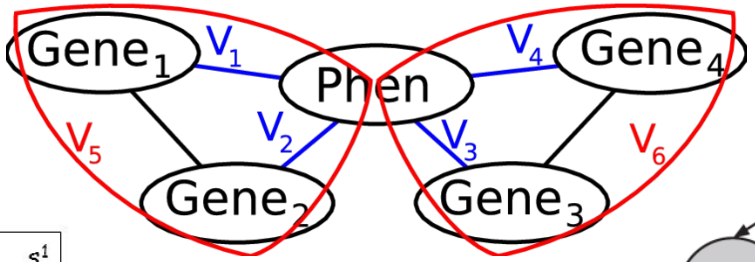
i^0	i^1
0.7	0.3

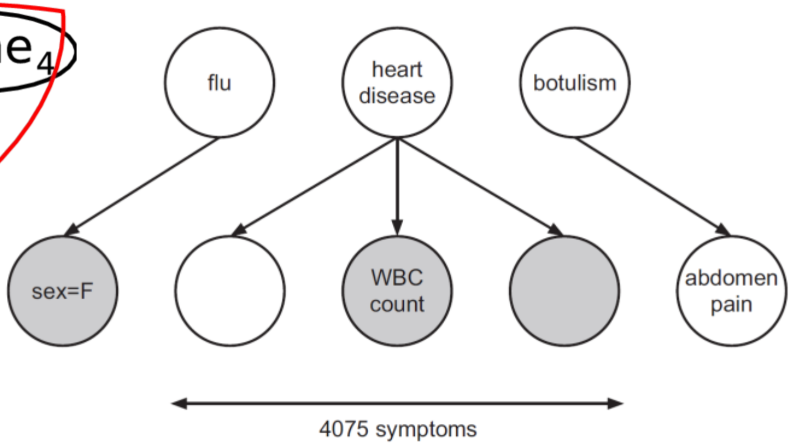
	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

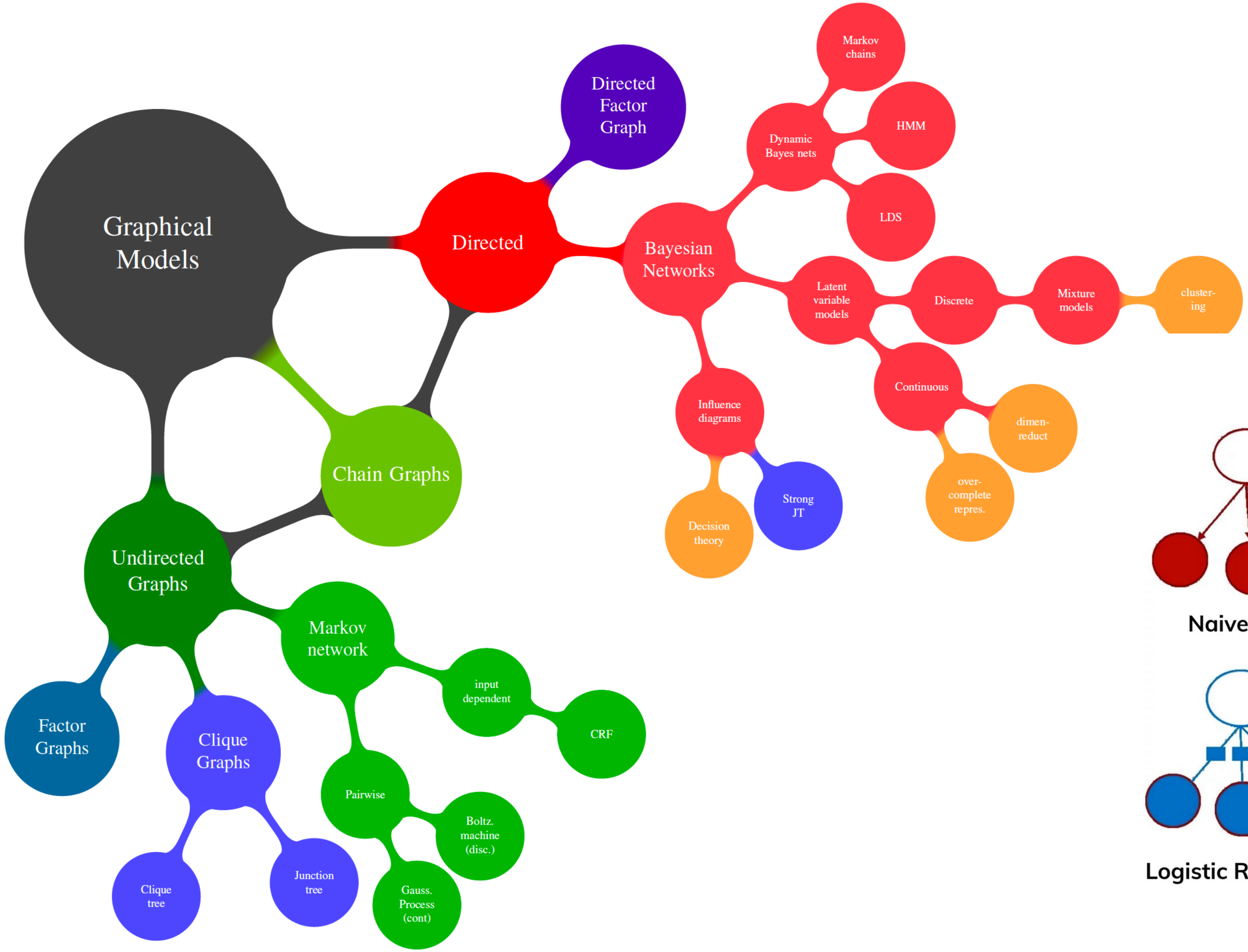
Daphne Koller

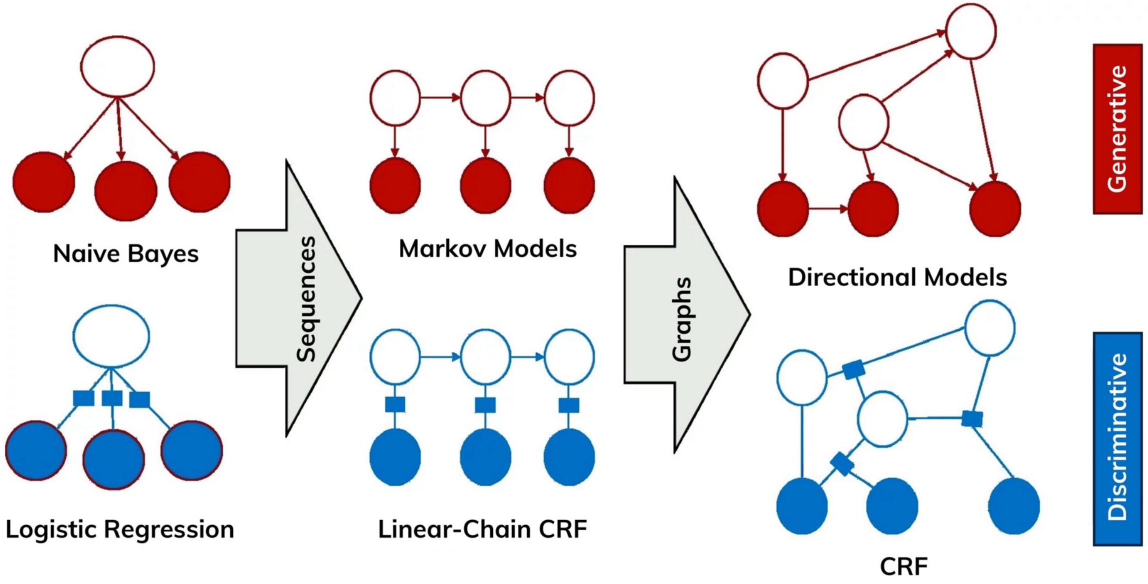




https://www.researchgate.net/publication/47498960_Markov_Logic_Networks_in_the_Analysis_of_Genetic_Data

[illegible]G [illegible] [illegible]





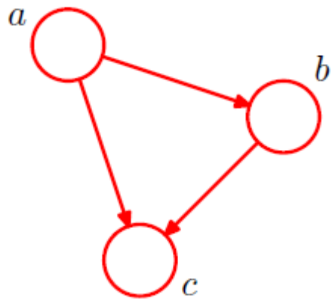
Adapted from C.Sutton, A.McCallum, “An introduction to Conditional Random Fields”

आई आई टी हैदराबाद
IIT Hyderabad

[illegible] G [illegible]

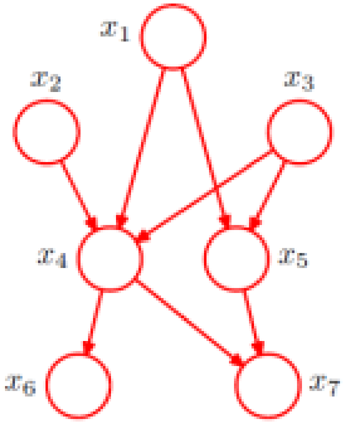
- Independently specifying all the entries of a table p(x1; : : : ; xN) over binary variables xi takes O(2^N) space
- Structure is also important for computational tractability of inferring quantities of interest.
- Given a distribution on N binary variables, p(x1; : : : ; xN), computing a marginal such as p(x1) requires summing over the 2^(N-1) states of the other variables.
- Belief networks (also called Bayes' networks or Bayesian belief networks) are a way to depict the independence
- assumptions made in a distribution

$$p(a, b, c) = p(c|a, b)p(b|a)p(a).$$



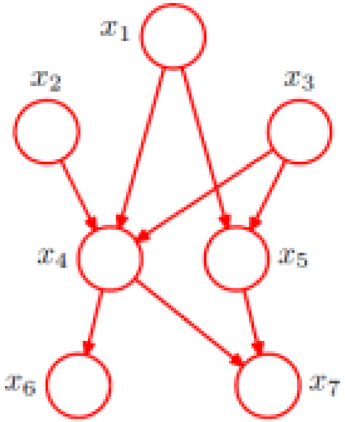
[illegible]G[illegible]

- Independently specifying all the entries of a table p(x1; : : : ; xN) over binary variables xi takes O(2^N) space
- Structure is also important for computational tractability of inferring quantities of interest.
- Given a distribution on N binary variables, p(x1; : : : ; xN), computing a marginal such as p(x1) requires summing over the 2^(N-1) states of the other variables.
- Belief networks (also called Bayes' networks or Bayesian belief networks) are a way to depict the independence
- assumptions made in a distribution



[illegible]G[illegible]

- Independently specifying all the entries of a table p(x1; …; xN) over binary variables xi takes O(2^N) space
- Structure is also important for computational tractability of inferring quantities of interest.
- Given a distribution on N binary variables, p(x1; … ; xN), computing a marginal such as p(x1) requires summing over the 2^(N-1) states of the other variables.
- Belief networks (also called Bayes' networks or Bayesian belief networks) are a way to depict the independence
- assumptions made in a distribution



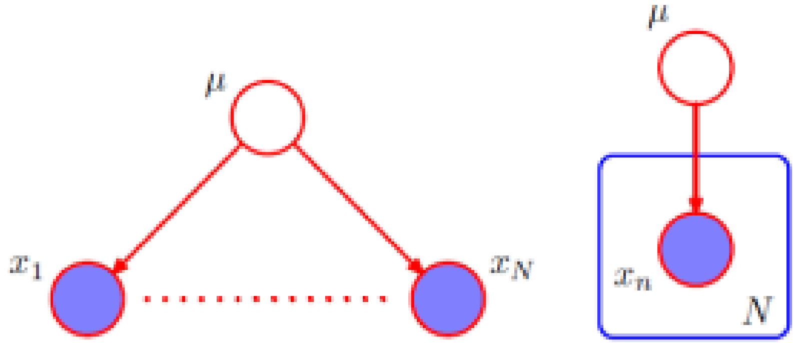
$$p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5).$$

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

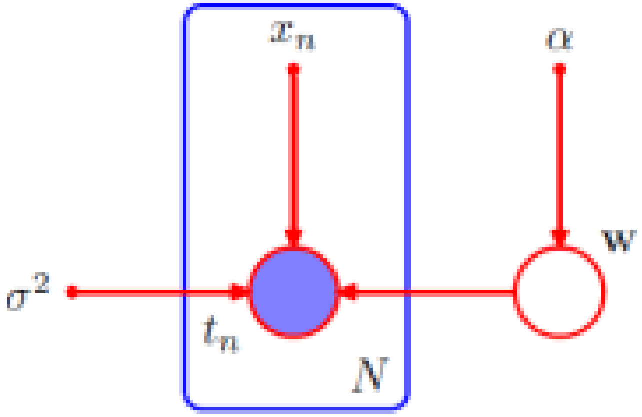
[illegible]

- **Plate Notation**
- Bayesian Linear Regression

$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu).$$



$$p(\mathbf{t}, \mathbf{w}|\mathbf{x}, \alpha, \sigma^2) = p(\mathbf{w}|\alpha) \prod_{n=1}^{N} p(t_n|\mathbf{w}, x_n, \sigma^2).$$



[illegible]G[illegible]

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

$$p(a|b,c) = p(a|c).$$

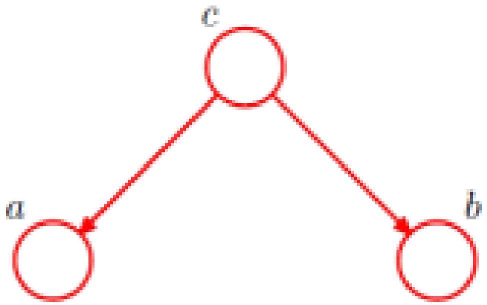
$$\begin{aligned} p(a,b|c) &= p(a|b,c)p(b|c) \\ &= p(a|c)p(b|c). \end{aligned}$$

$$p(a,b,c) = p(a|c)p(b|c)p(c).$$

- **Conditional Independence**
- *a* is conditionally independent of *b* given *c*.

joint distribution of *a* and *b* factorizes into the product of the marginal distribution of *a* and the marginal distribution of *b* (again both conditioned on *c*).

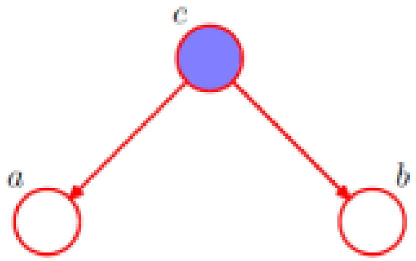
$$a \perp\!\!\!\perp b \mid c$$



[illegible]

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

- **Conditional Independence**



$$p(a, b, c) = p(a|c)p(b|c)p(c).$$

$$\begin{aligned} p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

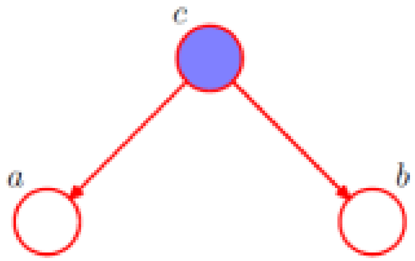
$$a \perp\!\!\!\perp b \mid c.$$

Is a and b unconditionally independent ?

[illegible]

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

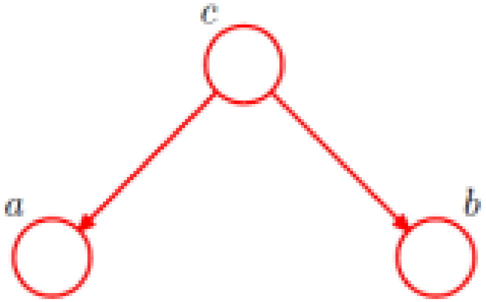
- **Conditional Independence**



$$p(a, b, c) = p(a|c)p(b|c)p(c).$$

$$\begin{aligned} p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

$$a \perp\!\!\!\perp b \mid c.$$



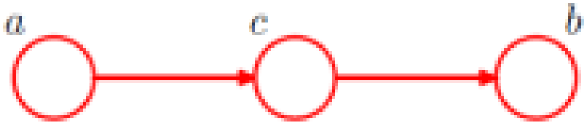
$$p(a, b) = \sum_{c} p(a|c)p(b|c)p(c).$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

[illegible]G[illegible]

- **Conditional Independence**

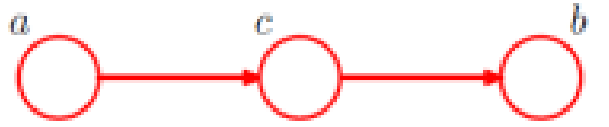
D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph



[illegible] G [illegible]

- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph



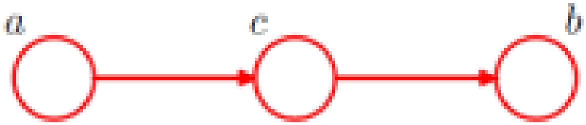
$$p(a, b, c) = p(a)p(c|a)p(b|c).$$

Is a independent of b ?
Is a independent of b conditioned on c ?

[illegible]

- **Conditional Independence**

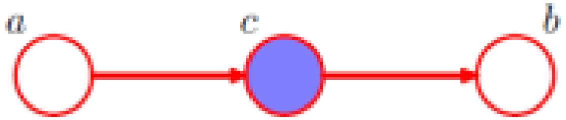
D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph



$$p(a,b,c) = p(a)p(c|a)p(b|c).$$

$$p(a,b) = p(a)\sum_c p(c|a)p(b|c) = p(a)p(b|a).$$

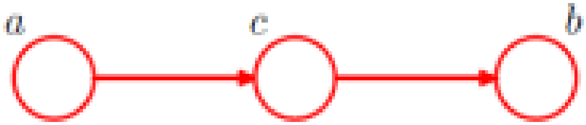
$$a \not\perp\!\!\!\perp b \mid \emptyset$$



[illegible]G[illegible]

- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph



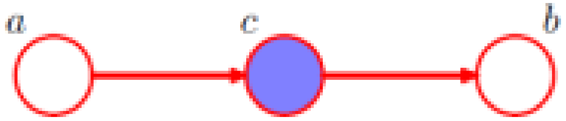
$$p(a,b,c) = p(a)p(c|a)p(b|c).$$

$$p(a,b) = p(a)\sum_c p(c|a)p(b|c) = p(a)p(b|a).$$

$$a \not\!\perp\!\!\!\perp b \mid \emptyset$$

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

$$a \perp\!\!\!\perp b \mid c.$$



[illegible]G[illegible]

- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

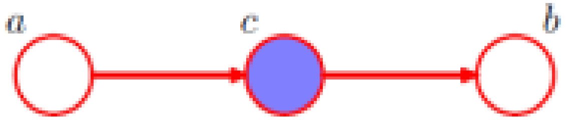
$$p(a, b) = p(a) \sum_{c} p(c|a)p(b|c) = p(a)p(b|a).$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

$$\begin{aligned} p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

$$a \perp\!\!\!\perp b \mid c.$$

The node *c* is said to be *head-to-tail* with respect to the path from node *a* to node *b*. Such a path connects nodes *a* and *b* and c blocks them renders them dependent.



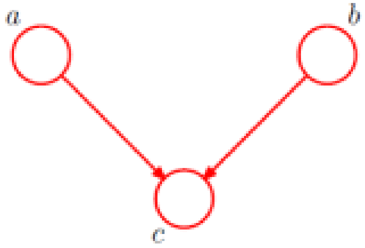
[illegible] G [illegible]

- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

Is a independent of b ?

Is a independent of b conditioned on c ?

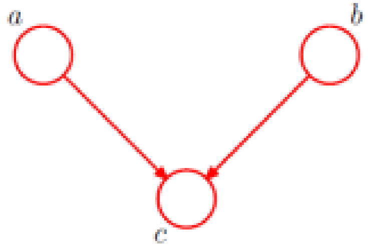


[illegible]G[illegible]

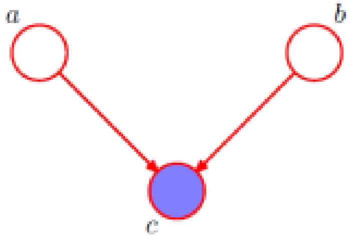
- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

$$p(a, b, c) = p(a)p(b)p(c|a, b).$$



$$p(a, b) = p(a)p(b) \qquad a \perp\!\!\!\perp b \mid \emptyset.$$

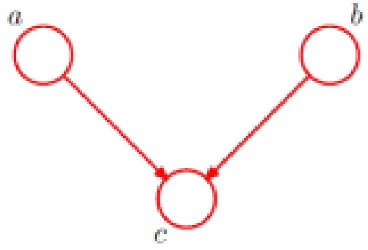


Is a independent of b conditioned on c ?

- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

$$p(a, b, c) = p(a)p(b)p(c|a, b).$$

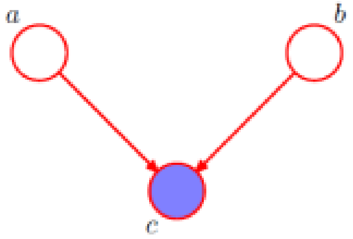


$$p(a, b) = p(a)p(b)$$

$$a \perp\!\!\!\perp b \mid \emptyset.$$

$$\begin{aligned} p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)p(b)p(c|a, b)}{p(c)} \end{aligned}$$

$$a \not\perp\!\!\!\perp b \mid c.$$



[illegible]G[illegible]

- **Conditional Independence**

D-separation : An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph

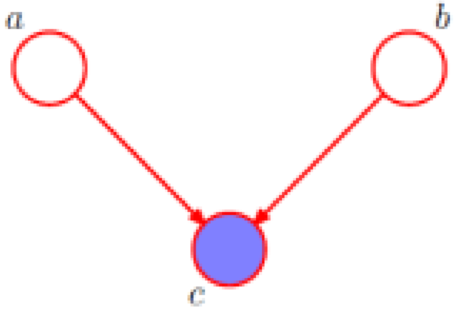
node *c* is *head-to-head* with respect to the path from *a* to *b* because it connects to the heads of the two arrows. When node *c* is unobserved, it ‘blocks’ the path, and the variables *a* and *b* are independent. However, conditioning on *c* ‘unblocks’ the path and renders *a* and *b* dependent.

$$p(a,b) = p(a)p(b)$$

$$a \perp\!\!\!\perp b \mid \emptyset.$$

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(b)p(c|a,b)}{p(c)} \end{aligned}$$

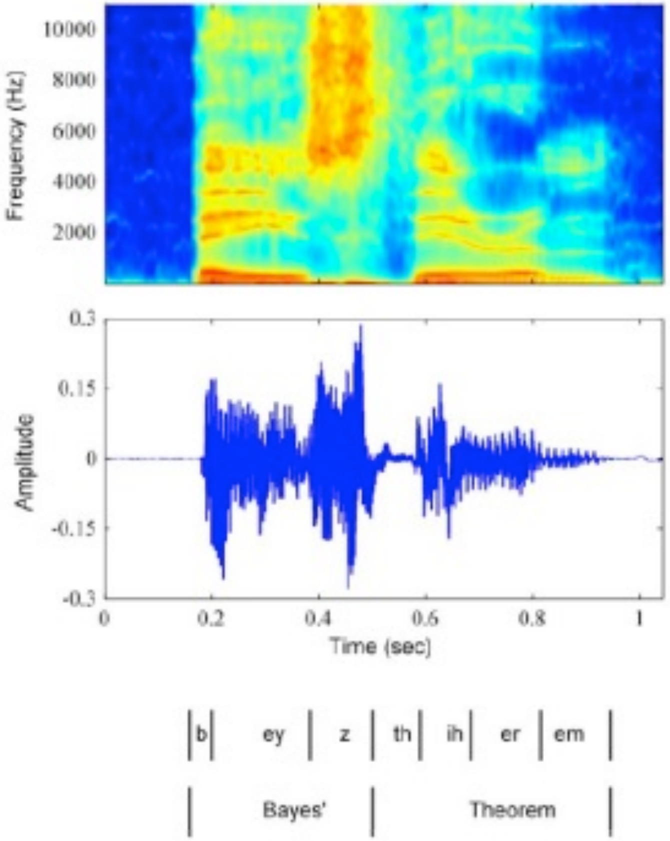
$$a \not\perp\!\!\!\perp b \mid c.$$

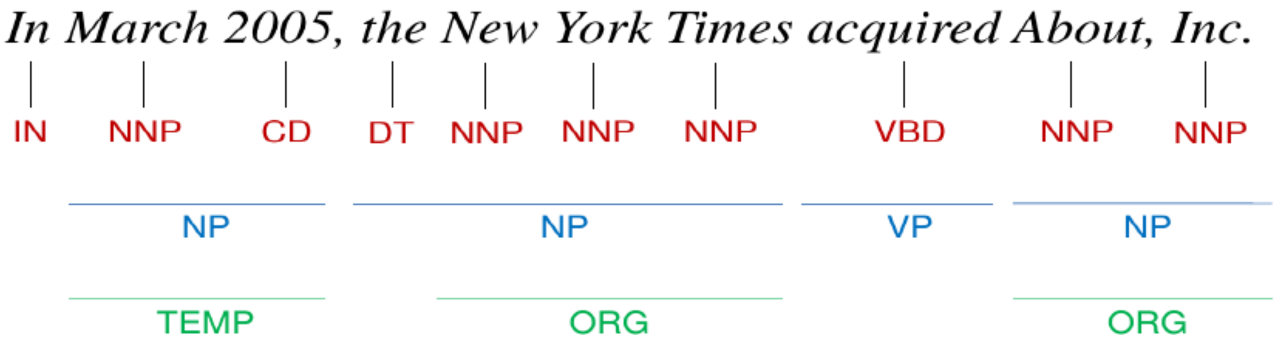


explaining away’.

[illegible]

- Sequential data : rainfall measurements on successive days at a particular location, or the daily values of a currency exchange rate (time series data) , sequence of nucleotide base pairs along a strand of DNA or the sequence of characters in an English sentence





In March 2005, the New York Times acquired About, Inc.

IN NNP CD DT NNP NNP NNP VBD NNP NNP

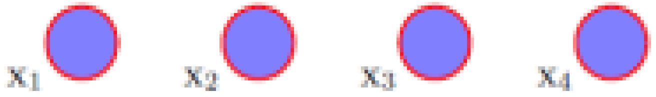
NP NP VP NP

TEMP ORG ORG

Markov Model

- binary variable denoting whether on a particular day it rained or not.

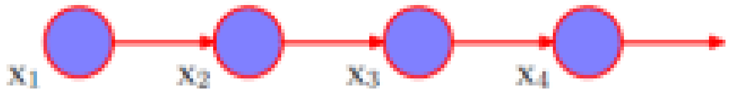
$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{x}_1, \dots, \mathbf{x}_{n-1}).$$



- Markov Model (first Order) : conditional distributions on the right-hand side is independent of all previous observations except the most recent

$$p(\mathbf{x}_n|\mathbf{x}_1, \dots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}).$$

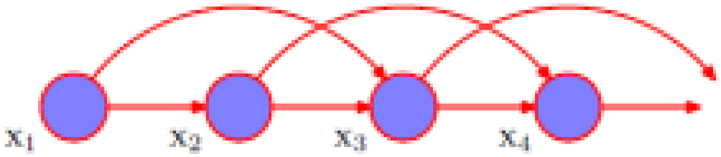


$K(K-1)$ parameters.

M^{th} order Markov chain,

$p(\mathbf{x}_n|\mathbf{x}_{n-M}, \dots, \mathbf{x}_{n-1})$.

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1) \prod_{n=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2}).$$



$K^{M-1}(K-1)$ parameters.

[illegible]G[illegible]

- Inference in Graphical Models
  - Message passing algorithms
  - Max-sum algorithm

$$\mathbf{x}^{\max} = \arg\max_{\mathbf{x}} p(\mathbf{x})$$

$$\max_{\mathbf{x}} p(\mathbf{x}) = \max_{x_1} \ldots \max_{x_M} p(\mathbf{x})$$

$$\begin{aligned}\max_{\mathbf{x}} p(\mathbf{x}) &= \frac{1}{Z} \max_{x_1} \cdots \max_{x_N} \left[\psi_{1,2}(x_1, x_2) \cdots \psi_{N-1,N}(x_{N-1}, x_N)\right] \\ &= \frac{1}{Z} \max_{x_1} \left[\psi_{1,2}(x_1, x_2) \left[\cdots \max_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right]\right].\end{aligned}$$

Sum product algorithm

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(\mathbf{x}).$$

$$p(\mathbf{x}) = \prod_{s \in \mathrm{ne}(x)} F_s(x, X_s)$$

$$p(x) = \prod_{s \in \mathrm{ne}(x)} \left[\sum_{X_s} F_s(x, X_s)\right]$$

Hidden Markov Model

- HMM is widely used in speech. recognition (Jelinek, 1997; Rabiner and Juang, 1993), natural language modelling (Manning and Sch¨utze, 1999), on-line handwriting recognition (Nag et al., 1986), and for the analysis of biological sequences such as proteins and DNA

- Standard classification problem assumes individual cases are disconnected and independent (i.i.d.: independently and identically distributed).
- Each token in a sequence is assigned a label. Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors (not i.i.d).

- A given sentence, *“Time flies like an arrow”*
- Represent the input sentence with a token vector $\boldsymbol{x}$

↑

(Bold italic)

(*NOTE: This does not present a feature vector*)

t	1	2	3	4	5	
$\boldsymbol{x}$	$Time$	$flies$	$like$	an	$arrow$	(T = 5)
	x_1	x_2	x_3	x_4	x_5	

- Predict part-of-speech (a vector $\boldsymbol{y}$) tags for the tokens $\boldsymbol{x}$

t	1	2	3	4	5
$\boldsymbol{x}$	$Time$	$flies$	$like$	an	$arrow$
	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{y}$	NN	VBZ	IN	DT	NN
	y_1	y_2	y_3	y_4	y_5

Predict

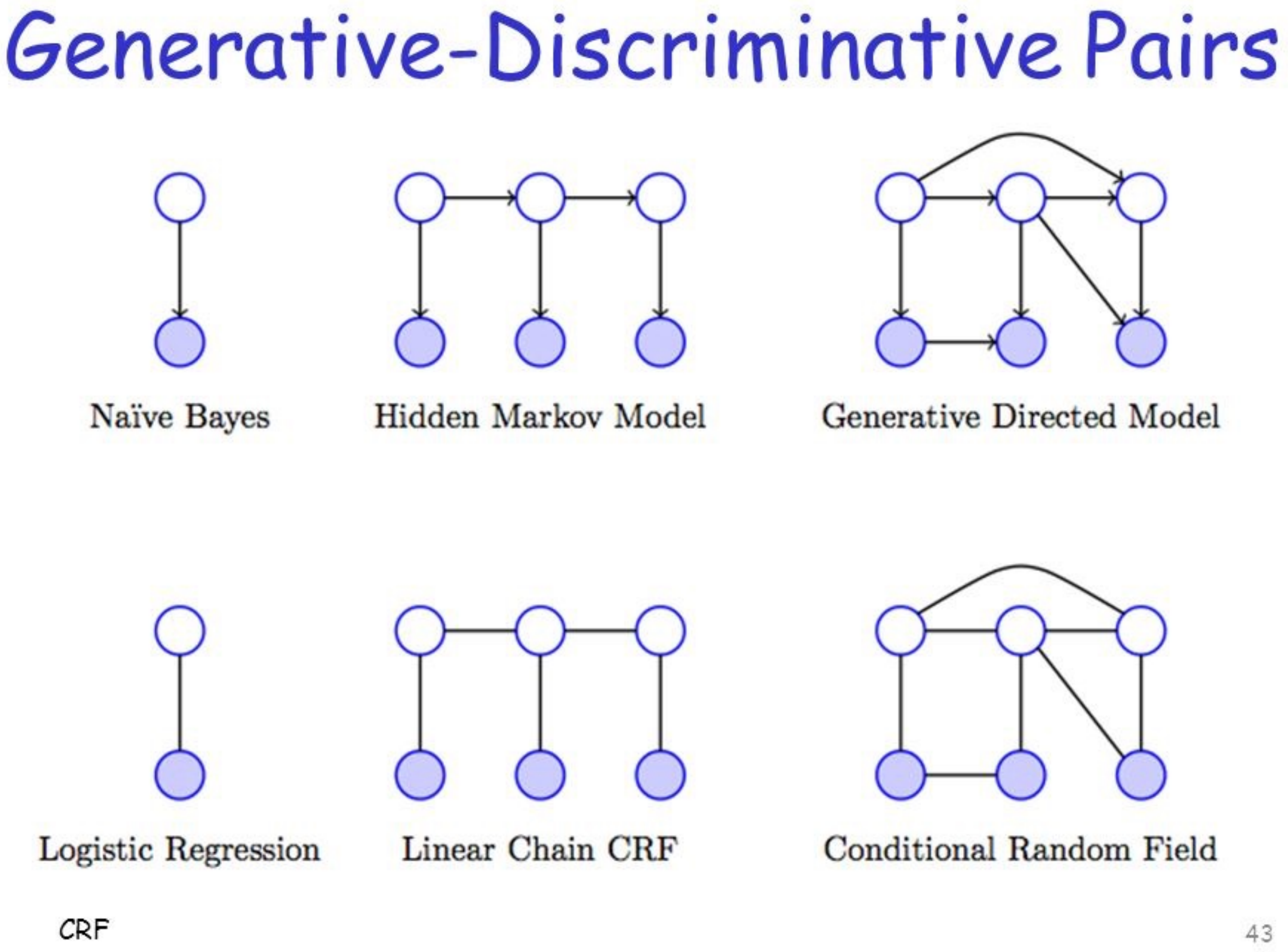
- *Modeling*: how to build (assume) $P(\boldsymbol{y}|\boldsymbol{x})$
  - Hidden Markov Model (HMM), Structured Perceptron, Conditional Random Fields (CRFs), etc

- *Training*: how to determine unknown parameters in the model so that they fit to a training data
  - Maximum Likelihood (ML), Maximum a Posteriori (MAP), etc
  - Gradient-based method, Stochastic Gradient Descent (SGD), etc

- *Inference*: how to compute $\operatorname{argmax} P(\boldsymbol{y}|\boldsymbol{x})$ efficiently
  - Viterbi algorithm

Probabilistic Sequence Models

- Probabilistic sequence models allow integrating uncertainty over multiple, interdependent classifications and collectively determine the most likely global assignment.
- Two standard models
  - Generative Model : Hidden Markov Model (HMM)
  - Discriminative Model : Conditional Random Field (CRF)



Generative-Discriminative Pairs

CRF

Hidden Markov Model

- $\boldsymbol{x}$: the sequence of tokens (words)
- $\boldsymbol{y}$: the sequence of POS tags

- Bayes' theorem:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})}{P(\boldsymbol{x})}$$

- Bayesian inference: decompose $P(\boldsymbol{y}|\boldsymbol{x})$ into two factors, $P(\boldsymbol{x}|\boldsymbol{y})$ and $P(\boldsymbol{y})$, which might be easier to model

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}) \underset{\substack{\uparrow \\ \text{Bayes'} \\ \text{theorem}}}{=} \operatorname*{argmax}_{\boldsymbol{y}} \frac{P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})}{P(\boldsymbol{x})} \underset{\substack{\uparrow \\ P(\boldsymbol{x}) \text{ is the same} \\ \text{for all } \boldsymbol{y}}}{=} \operatorname*{argmax}_{\boldsymbol{y}} P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})$$

HMM

- Two Markov assumptions to simplify $P(\boldsymbol{x}|\boldsymbol{y})$ and $P(\boldsymbol{y})$
  - A word appears depending only on its POS tag
    - Independently of other words around the word
    - Generated by emission probability distribution

$$P(\boldsymbol{x}|\boldsymbol{y}) \approx \prod_{t=1}^{T} P(x_t|y_t)$$

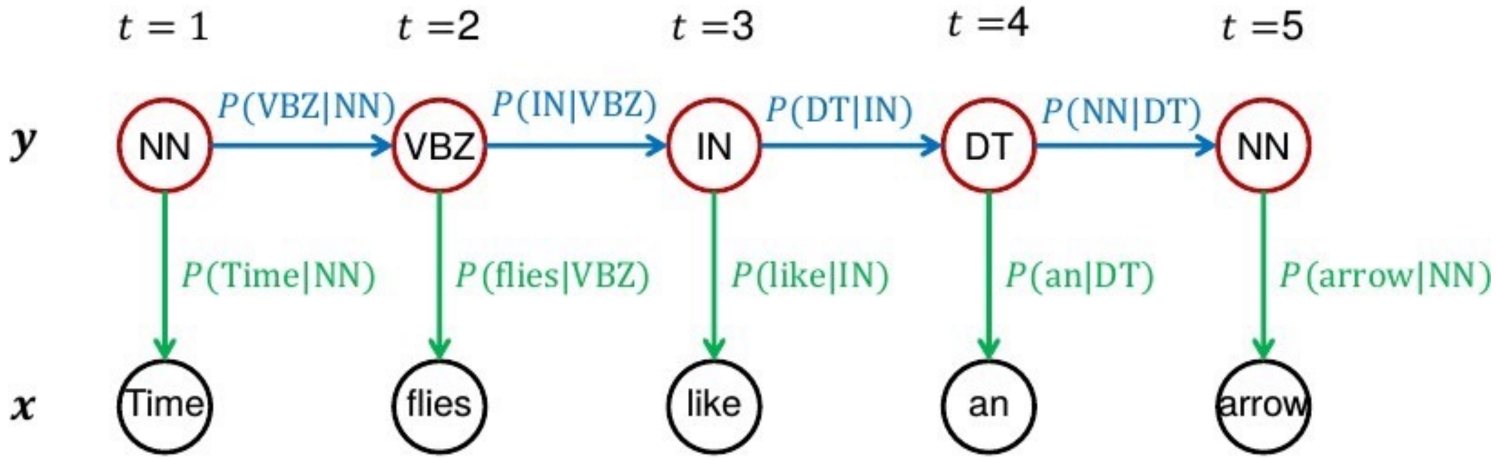
  - A POS tag is dependent only on the previous one
    - Rather than the entire tag sequence
    - Generated by transition probability distribution

$$P(\boldsymbol{y}) \approx \prod_{t=1}^{T} P(y_t|y_{t-1})$$

- Then, the most probable tag sequence $\widehat{\boldsymbol{y}}$ is computed by,

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} P(\boldsymbol{y}|\boldsymbol{x}) = \underset{\boldsymbol{y}}{\operatorname{argmax}} P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y}) \approx \underset{\boldsymbol{y}}{\operatorname{argmax}} \prod_{t=1}^{T} P(x_t|y_t)P(y_t|y_{t-1})$$

POS Tagging



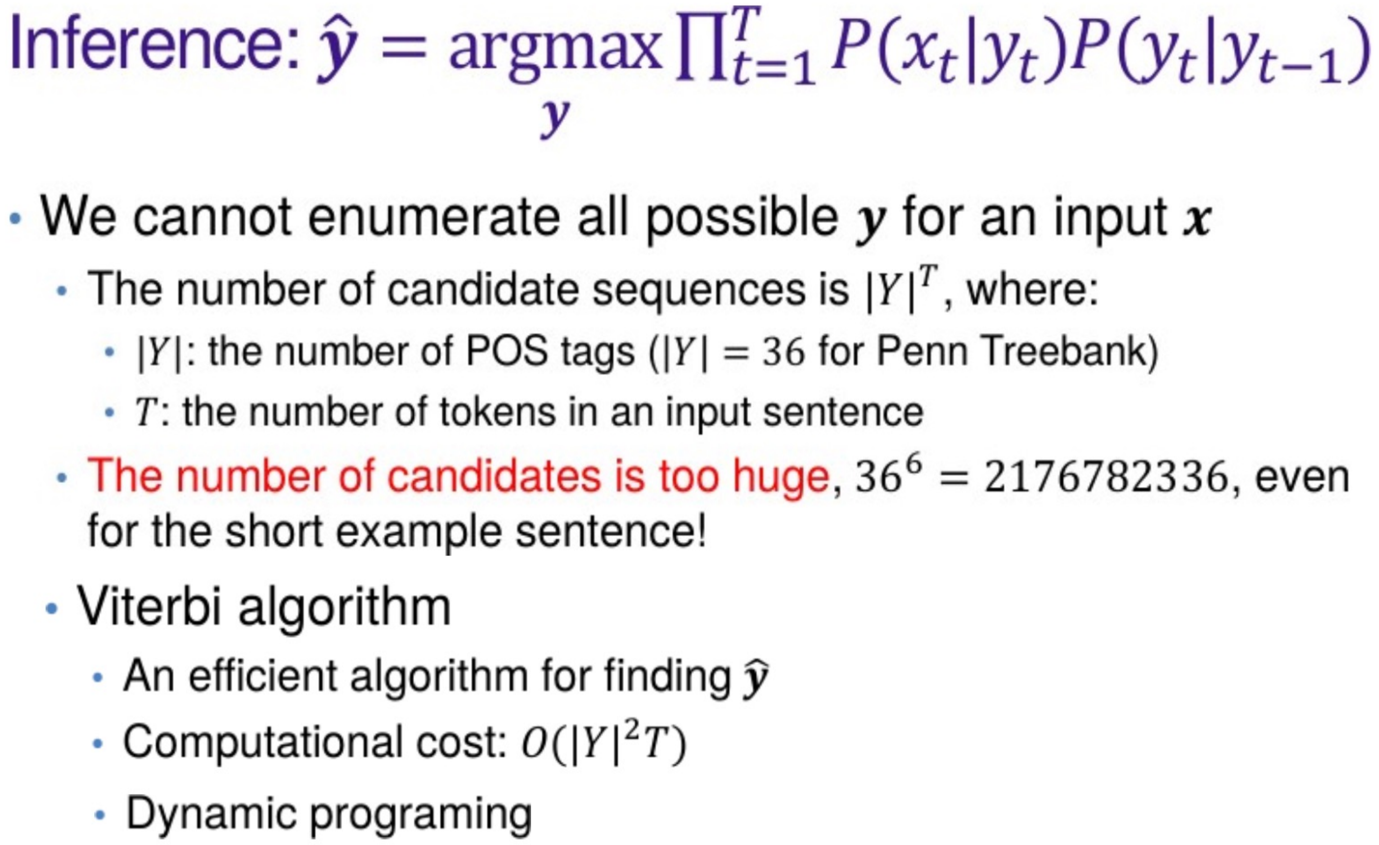
- We can compute $\phi(\boldsymbol{x}, \boldsymbol{y})$ if we decide an assignment of $\boldsymbol{y}$ for a given input $\boldsymbol{x}$: $\prod_{t=1}^{T} P(x_t|y_t)P(y_t|y_{t-1})$

$$P(x_t|y_t) = \frac{C(x_t, y_t)}{C(y_t)} = \frac{(\text{the number of times where } x_t \text{ is annotated as } y_t)}{(\text{the number of occurrences of tag } y_t)}$$

$$P(y_t|y_{t-1}) = \frac{C(y_t, y_{t-1})}{C(y_{t-1})} = \frac{(\text{the number of occurrences of tag } y_t \text{ followd by } y_{t-1})}{(\text{the number of occurrences of tag } y_{t-1})}$$

Viterbi Algorithm

- Given the observations, find the best possible tag sequence which generated it.



Inference: $\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} \prod_{t=1}^{T} P(x_t|y_t)P(y_t|y_{t-1})$

- We cannot enumerate all possible $\boldsymbol{y}$ for an input $\boldsymbol{x}$
  - The number of candidate sequences is $|Y|^T$, where:
    - $|Y|$: the number of POS tags ($|Y| = 36$ for Penn Treebank)
    - T: the number of tokens in an input sentence
  - The number of candidates is too huge, $36^6 = 2176782336$, even for the short example sentence!
- Viterbi algorithm
  - An efficient algorithm for finding $\hat{\boldsymbol{y}}$
  - Computational cost: $O(|Y|^2 T)$
  - Dynamic programing

Conditional Random Field

$$Y = \bar{y}_1^n = y_1 \ldots yn.$$

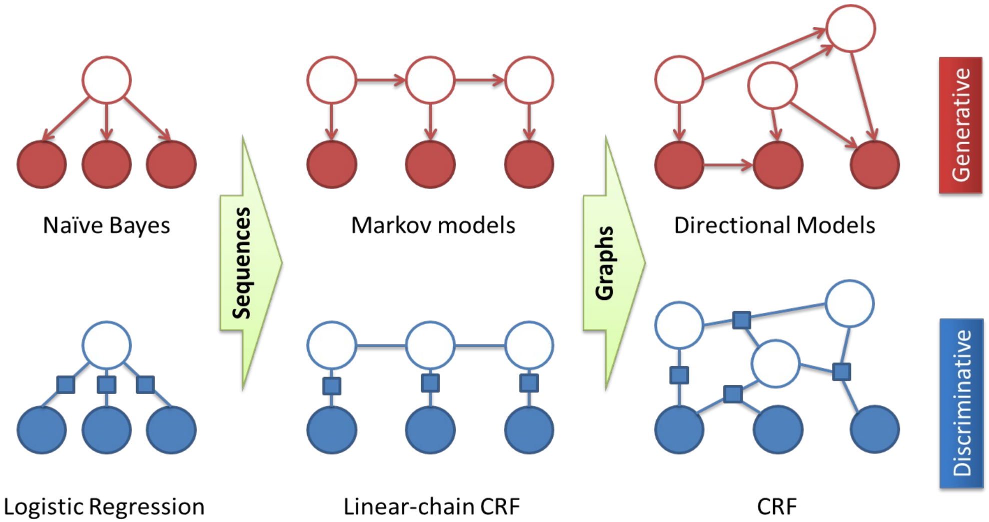
$$X = x_1^n = x_1 \ldots x_n$$

HMM

$$\begin{aligned}\hat{Y} &= \underset{Y}{\operatorname{argmax}}\, p(Y|X) \\ &= \underset{Y}{\operatorname{argmax}}\, p(X|Y)p(Y) \\ &= \underset{Y}{\operatorname{argmax}} \prod_i p(x_i|y_i) \prod_i p(y_i|y_{i-1})\end{aligned}$$

CRF

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}}\, P(Y|X)$$



Adapted from C. Sutton, A. McCallum, "An Introduction to Conditional Random Fields", ArXiv, November 2010

Conditional Random Field

- Conditional probability is defined,

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp((\boldsymbol{w} \cdot \boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})))}{\boxed{\sum_{\boldsymbol{y}} \exp((\boldsymbol{w} \cdot \boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})))}} \longleftarrow$$

Normalized by the sum of exp'd scores of all possible paths in the lattice

- The same inference algorithm (Viterbi)

- Input: sequence of tokens $\boldsymbol{x} = (x_1\ x_2\ \ldots\ x_T)$
- Output: sequence of POS tags $\hat{\boldsymbol{y}} = (\widehat{y_1}\ \widehat{y_2}\ \ldots\ \widehat{y_T})$
- Mapping to global feature vector: $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})$: $(\boldsymbol{x},\boldsymbol{y}) \to \mathcal{R}^m$

$$\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y}) = \sum_{t=1}^{T} \boxed{\{\boldsymbol{u}(x_t, y_t)\ ,\ \boldsymbol{b}(y_{t-1}, y_t)\}} \longleftarrow$$

Local feature vector (at t):

- Unigram feature vector
- Bigram feature vector

  - Each element of feature vector consists of a feature function, e.g.,
    - $u_{109}(x_t, y_t) = \{1 \text{ (if } x_t = \text{Brown } \textit{and } y_t = \text{Noun); } 0 \text{ (otherwise)}\}$
    - $b_2(y_{t-1}, y_t) = \{1 \text{ (if } y_{t-1} = \text{Noun } \textit{and } y_t = \text{Verb); } 0 \text{ (otherwise)}\}$

CRF : Training and Inference

$$\begin{aligned}
\hat{Y} &= \operatorname*{argmax}_{Y \in \mathcal{Y}} P(Y|X) \\
&= \operatorname*{argmax}_{Y \in \mathcal{Y}} \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right) \\
&= \operatorname*{argmax}_{Y \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)\right) \\
&= \operatorname*{argmax}_{Y \in \mathcal{Y}} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i) \\
&= \operatorname*{argmax}_{Y \in \mathcal{Y}} \sum_{i=1}^{n} \sum_{k=1}^{K} w_k f_k(y_{i-1}, y_i, X, i)
\end{aligned}$$

$$p(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)}$$

$$p(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)$$

- Viterbi algorithm : like the HMM, the linearchain CRF depends at each timestep on only one previous output token y[i-1].

[illegible]G [illegible]

- References
  - *Pattern Recognition and Machine Learning* by Bishop
  - *Probabilistic Machine Learning* by Kevin Murphy
  - *Speech and Language processing* Jurafsky and Martin
  - McCallum, A.: Efficiently inducing features of conditional random fields. In: *Proc. 19th Conference on Uncertainty in Artificial Intelligence*. (2003)