Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 18

Oct, 21, 2015

Slide Sources
Raymond J. Mooney University of Texas at Austin

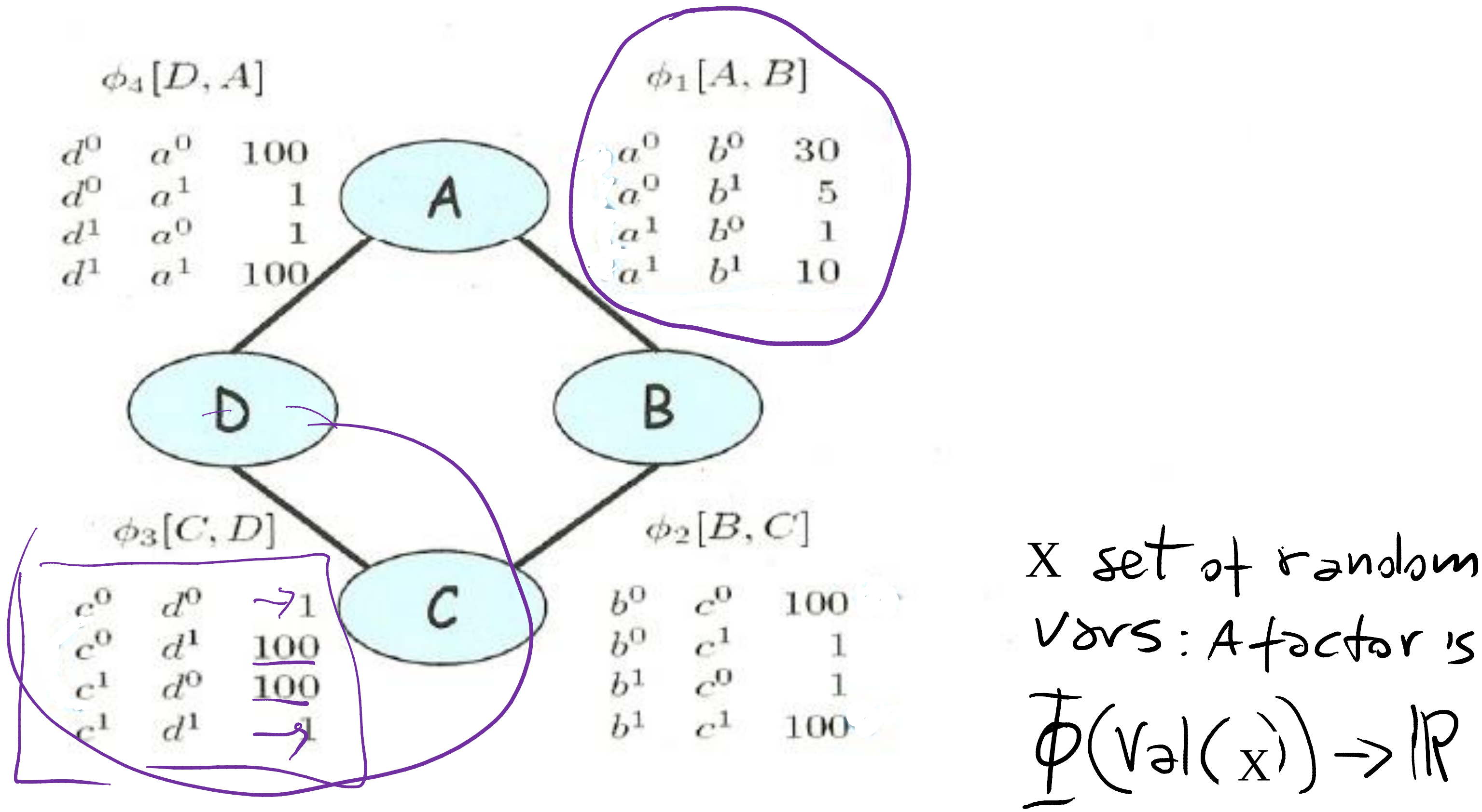
D. Koller, Stanford CS - Probabilistic Graphical Models

Lecture Overview

<u>Probabilistic Graphical models</u>

- Recap Markov Networks
- Applications of Markov Networks
- Inference in Markov Networks (Exact and Approx.)
- Conditional Random Fields

Parameterization of Markov Networks



Factors define the local interactions (like CPTs in Bnets)

What about the global model? What do you do with Bnets?

How do we combine local models?

As in BNets by multiplying them!

$$\tilde{P}(A,B,C,D) = \phi_1(A,B) \times \phi_2(B,C) \times \phi_3(C,D) \times \phi_4(A,D)$$

$$P(A,B,C,D) = \frac{1}{Z}\tilde{P}(A,B,C,D)$$

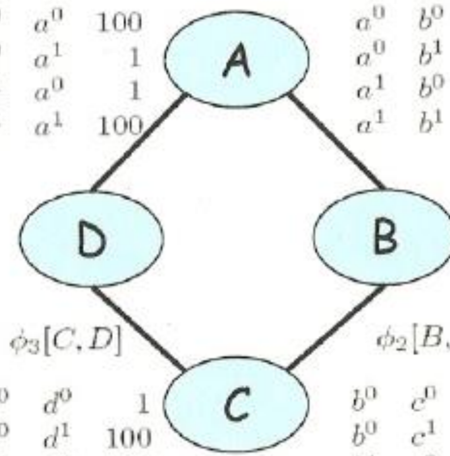
Assignment				Unnormalized	Normalized
a^0	b^0	c^0	d^0	300000	.04
a^0	b^0	c^0	d^1	300000	.04
a^0	b^0	c^1	d^0	300000	.04
a^0	b^0	c^1	d^1	30	4.1×10^{-6}
a^0	b^1	c^0	d^0	500	
a^0	b^1	c^0	d^1	500	
a^0	b^1	c^1	d^0	5000000	.69
a^0	b^1	c^1	d^1	500	
a^1	b^0	c^0	d^0	100	
a^1	b^0	c^0	d^1	1000000	
a^1	b^0	c^1	d^0	100	
a^1	b^0	c^1	d^1	100	
a^1	b^1	c^0	d^0	10	
a^1	b^1	c^0	d^1	100000	
a^1	b^1	c^1	d^0	100000	
a^1	b^1	c^1	d^1	100000	

$\phi_4[D,A]$

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

$\phi_1[A,B]$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10



$\phi_3[C,D]$

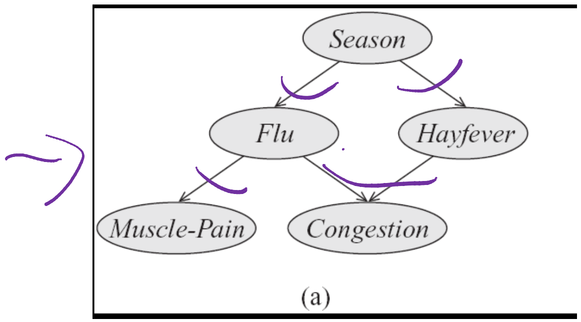
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

$\phi_2[B,C]$

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

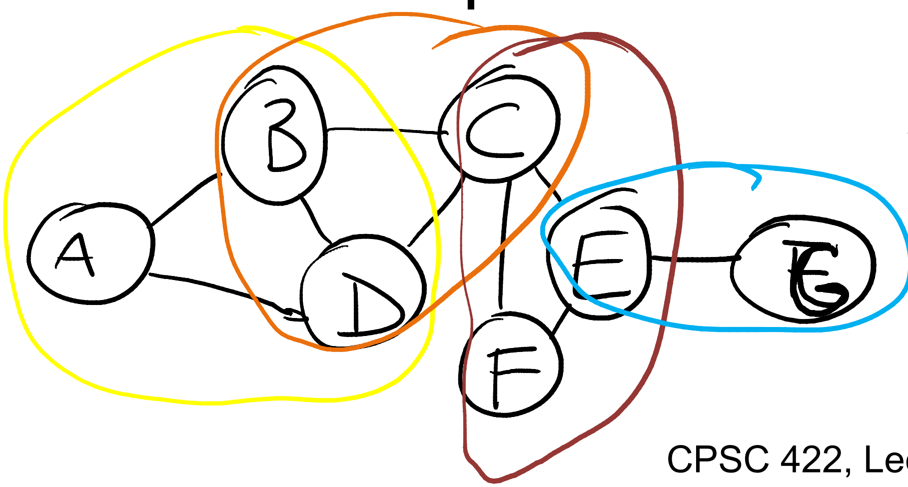
Step Back…. From structure to factors/potentials

In a Bnet the joint is factorized….



(a)

In a Markov Network you have one factor for each maximal clique



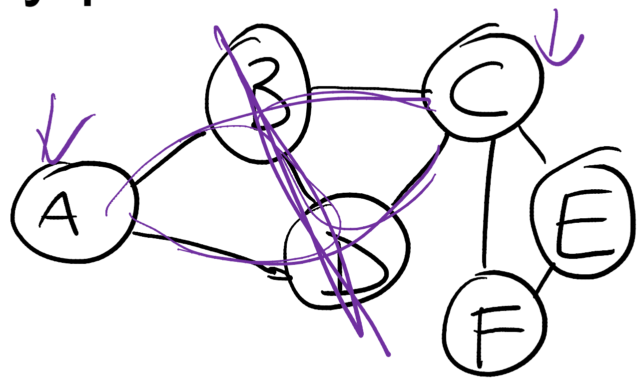
$\Phi_1(ABD)$

$\Phi_2(BDC)$ $\Phi_4(EG)$

$\Phi_3(CEF)$

General definitions

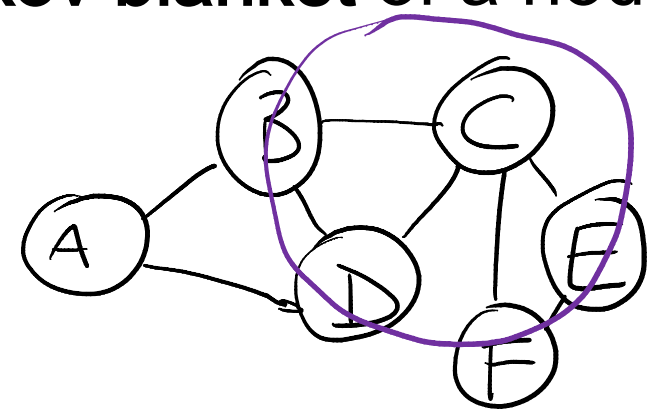
Two nodes in a Markov network are **independent** if and only if every path between them is cut off by evidence



eg for A C

So the **markov blanket** of a node is…?

eg for C



Lecture Overview

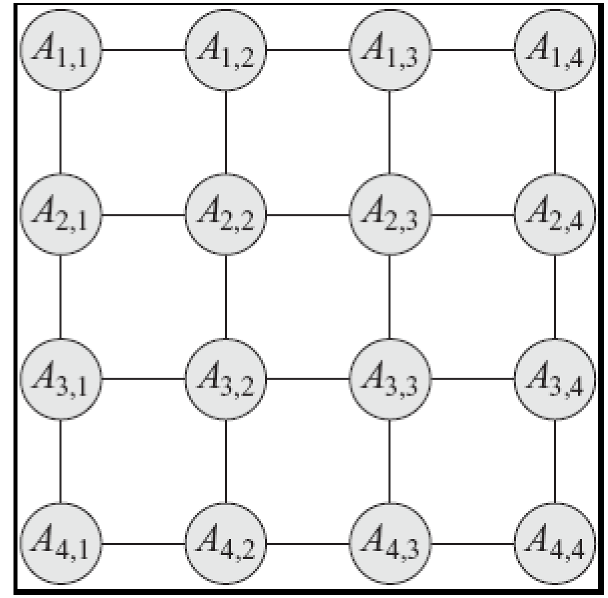
Probabilistic Graphical models

- Recap Markov Networks
- Applications of Markov Networks
- Inference in Markov Networks (Exact and Approx.)
- Conditional Random Fields

Markov Networks Applications (1): Computer Vision

Called **Markov Random Fields**

- Stereo Reconstruction
- Image Segmentation
- Object recognition



Typically **pairwise MRF**

- Each *vars* correspond to a *pixel* (or *superpixel*)
- Edges (factors) correspond to interactions between adjacent pixels in the image
  - E.g., in segmentation: from generically penalize discontinuities, to road under car

Image segmentation

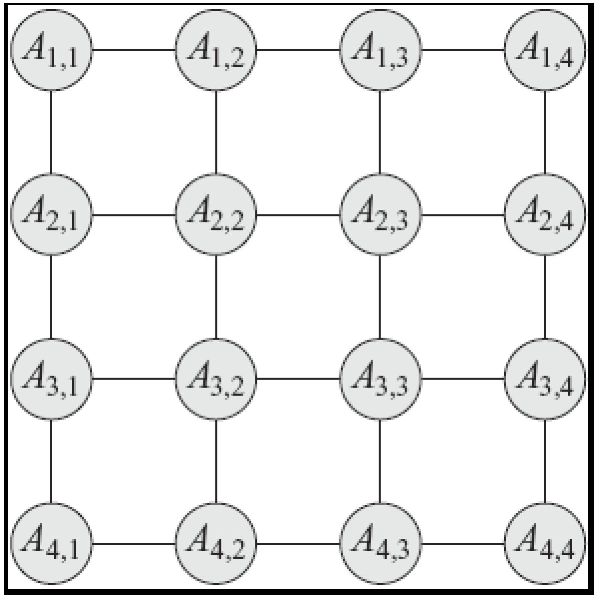
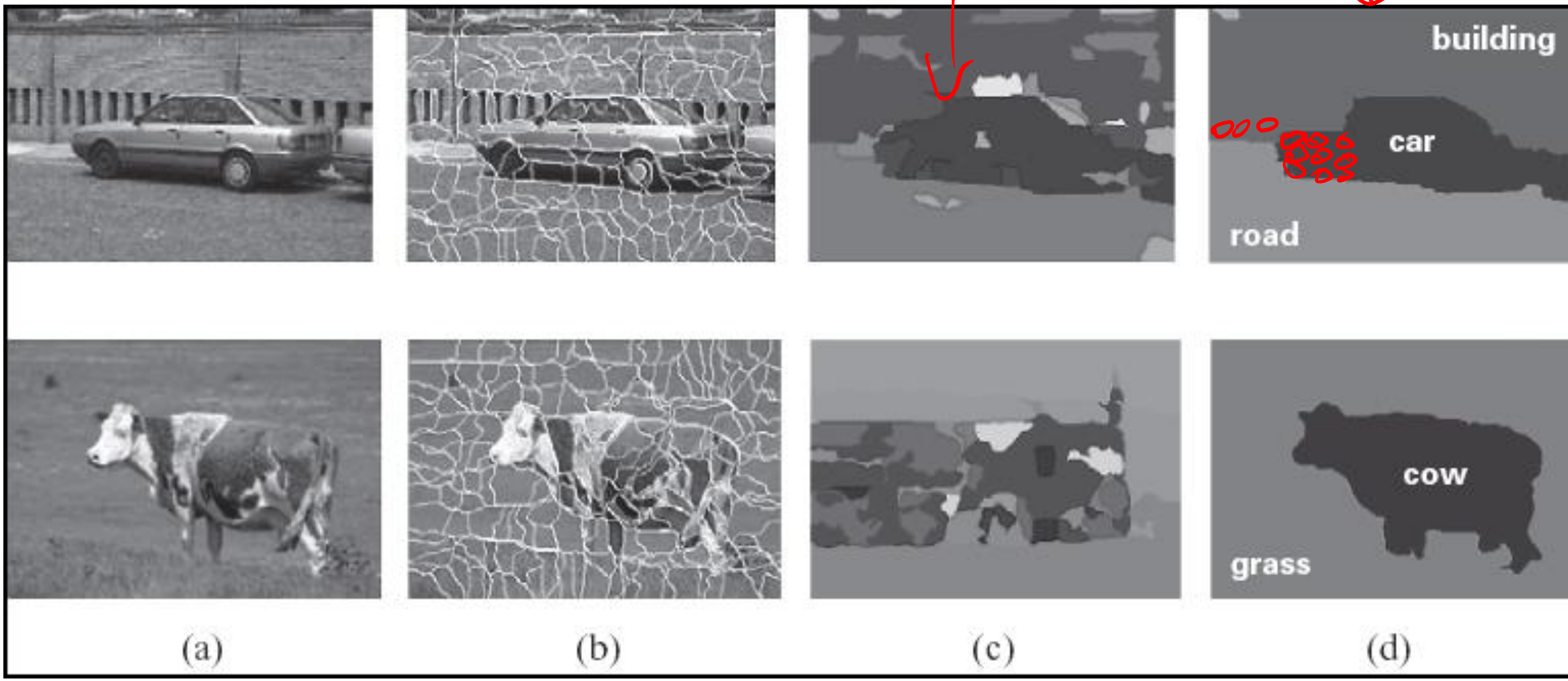


Image segmentation

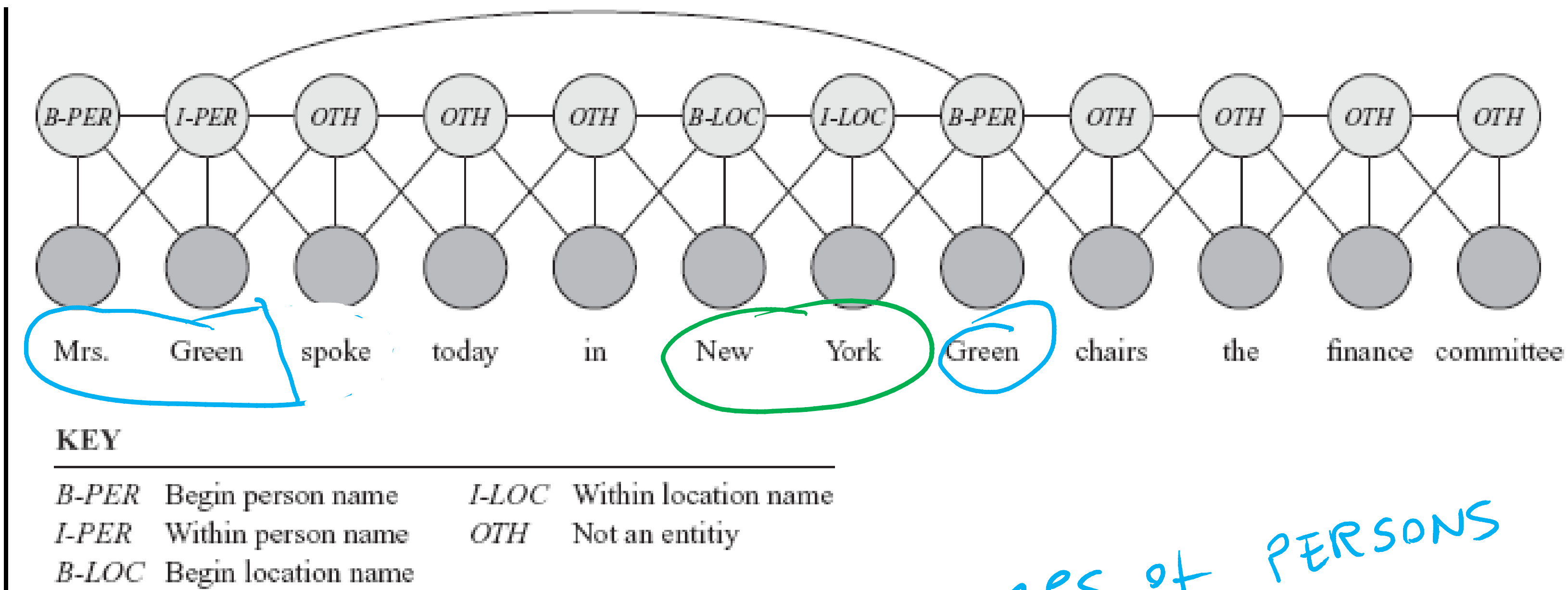


classifying each superpixel independently

with a Markov Random Field!

Markov Networks Applications (2): Sequence Labeling in NLP and BioInformatics

Conditional random fields



recognize names of PERSONS
LOCATIONS etc
NAMED ENTITIES

Lecture Overview

Probabilistic Graphical models

- Recap Markov Networks
- Applications of Markov Networks
- Inference in Markov Networks (Exact and Approx.)
- Conditional Random Fields

Variable elimination algorithm for Bnets

To compute $P(Z| Y_1=v_1 ,\ldots ,Y_j=v_j)$:

1. Construct a factor for each conditional probability.
2. Set the observed variables to their observed values.
3. Given an elimination ordering, simplify/decompose sum of products
4. Perform products and sum out Z_i
5. Multiply the remaining factors Z
6. Normalize: divide the resulting factor $f(Z)$ by $\sum_Z f(Z)$.

Variable elimination algorithm for Markov Networks…..

same!

Gibbs sampling for Markov Networks

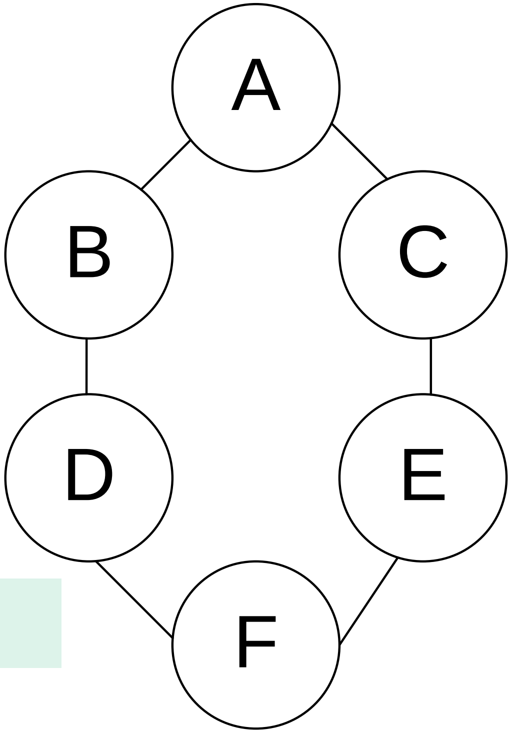
iclicker

Example: P(D | C=0)

Note: never change evidence!

Resample non-evidence variables in a pre-defined order or a random order

Suppose we begin with A



What do we need to sample?

A. **P(A |** B=0)

B. **P(A |** B=0, C=0)

C. **P(** B=0, C=0**| A)**

A	B	C	D	E	F
1	0	0	1	1	0

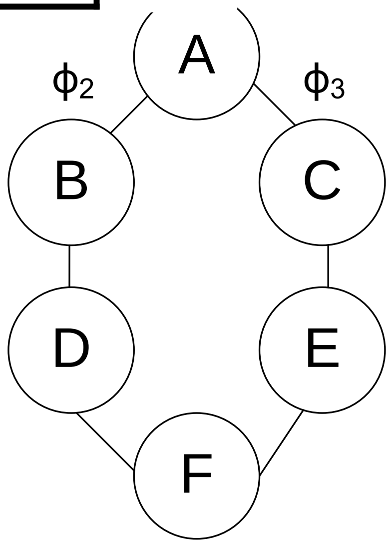
Example: Gibbs sampling

Resample probability distribution of P(A|BC)

A	B	C	D	E	F
1	0	0	1	1	0
?	0	0	1	1	0

	A=1	A=0
B=1	1	5
B=0	4.3	0.2

	A=1	A=0
C=1	1	2
C=0	3	4



$$\Phi_2 \times \Phi_3 =$$

A=1	A=0
12.9	0.8

A=1	A=0
0.95	0.05

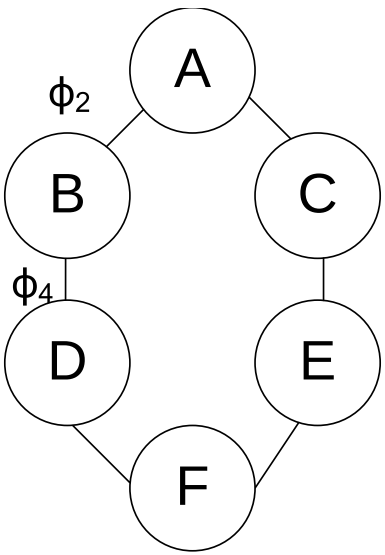
Example: Gibbs sampling

Resample probability distribution of B given A D

	A=1	A=0
B=1	1	5
B=0	4.3	0.2

A	B	C	D	E	F
1	0	0	1	1	0
1	0	0	1	1	0
1	?	0	1	1	0

	D=1	D=0
B=1	1	2
B=0	2	1



$$\phi_2 \times \phi_4 =$$

B=1	B=0
1	8.6

B=1	B=0
0.11	0.89

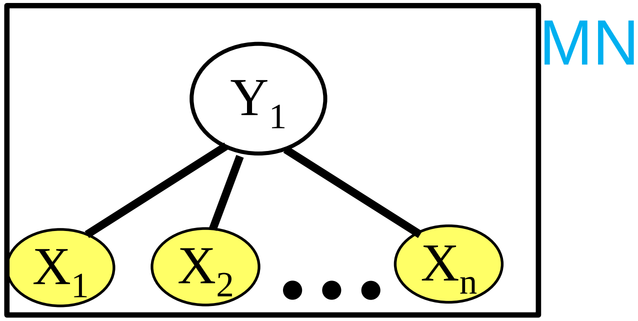
Lecture Overview

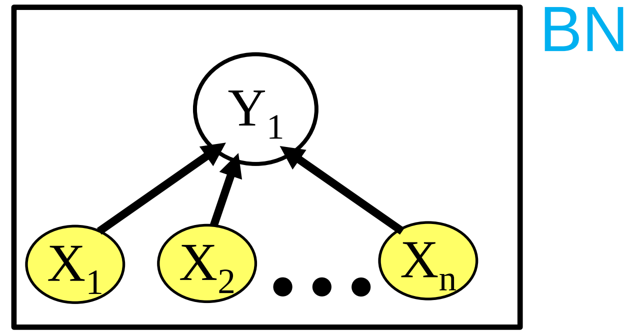
Probabilistic Graphical models

- Recap Markov Networks
- Applications of Markov Networks
- Inference in Markov Networks (Exact and Approx.)
- Conditional Random Fields

We want to model $P(Y_1 | X_1 .. X_n)$

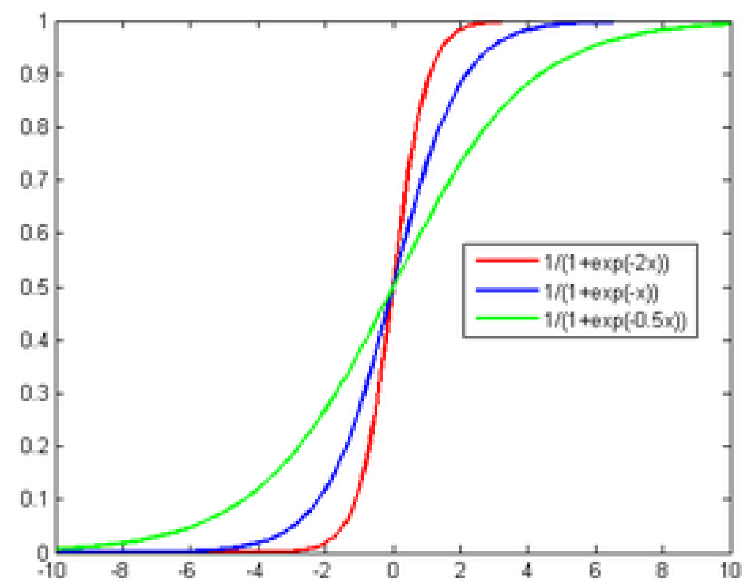
… where all the X_i are always observed





- Which model is simpler, MN or BN?

- Naturally aggregates the influence of different parents



Conditional Random Fields (CRFs)

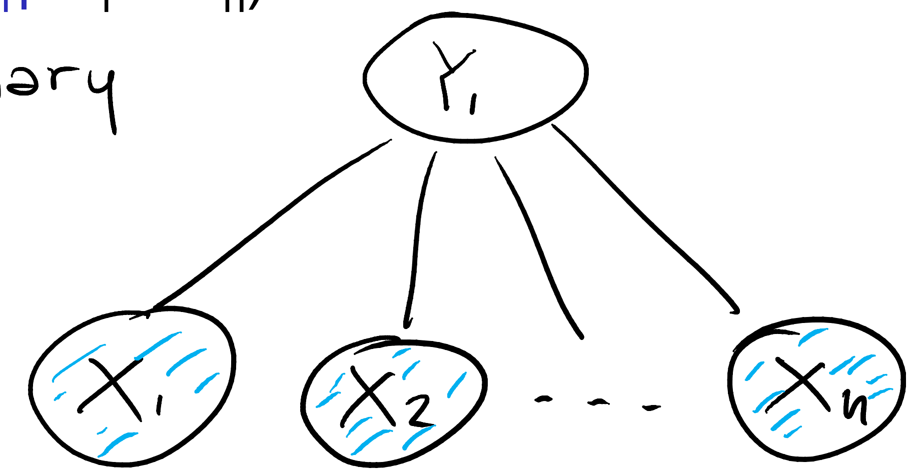
- Model $P(Y_1 .. Y_k | X_1 .. X_n)$
- Special case of Markov Networks where all the X_i are always observed

  - Simple case $P(Y_1 | X_1 \ldots X_n)$

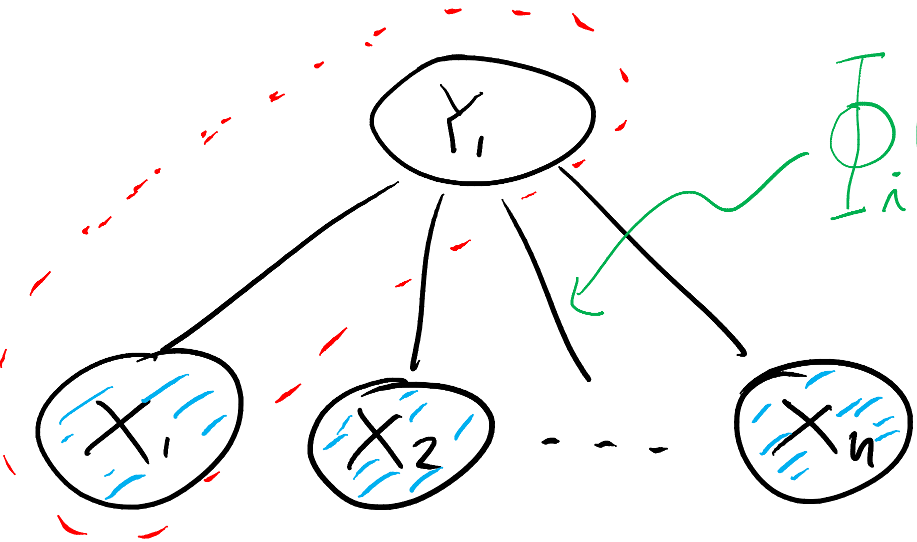
all vars are binary

$Y_1 = \{0, 1\}$

$\forall i \; X_i = \{0, 1\}$



What are the Parameters?



$$\Phi_i(X_i, Y_1) = \exp\{w_i * \mathbb{1}\{X_i = 1, Y_1 = 1\}\}$$

one such factor for each clique

also $\Phi_0(Y_1) = \exp\{w_0 \mathbb{1}\{Y = 1\}\}$

Example $w_2 = 1.5$ $\Phi_2(X_2, Y_1)$

X_2	Y_1	Φ_2
1	1	$e^{1.5}$
0	1	1
1	0	1
0	0	1

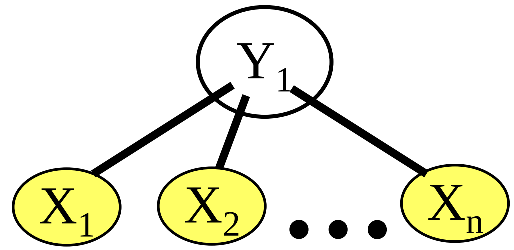
Example $w_0 = .4$

Y_1	Φ_0
0	1
1	$e^{.4}$

Let's derive the probabilities we need

$$\phi_i(X_i, Y_1) = \exp\{w_i * 1\{X_i = 1, Y_1 = 1\}\}$$

$$\phi_0(Y_1) = \exp\{w_0 * 1\{Y_1 = 1\}\}$$



$$\tilde{P}(Y_1 = 1, x_1, x_2, \ldots, x_n) = \Phi_0(Y_1) * \prod_{i=1}^{n} \Phi_i(x_i, Y_1)$$

A. $e^{\sum_1^n w_i}$

B. $e^{w_0 + \sum_1^n w_i * x_i}$

D $e^{w_0 + \sum_1^n w_i}$

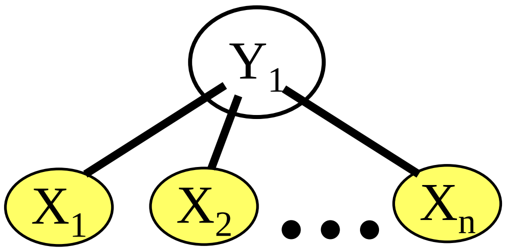
C. $e^{w_0 + \sum_1^n x_i}$

iclicker.

Let's derive the probabilities we need

$$\phi_i(X_i, Y_1) = \exp\{w_i \mathbb{1}\{X_i = 1, Y_1 = 1\}\}$$

$$\phi_0(Y_1) = \exp\{w_0 \mathbb{1}\{Y_1 = 1\}\}$$



$$\tilde{P}(Y_1 = 1, x_1, x_2, \ldots, x_n) = \Phi_0(Y_1) * \prod_{i=1}^{n} \Phi_i(x_i, Y_1)$$

example

$$P(Y_1 = 1, X_1 = 0, X_2 = 1, X_3 = 1)$$

$$e^{w_0 * 1} * e^{w_1 * 0} * e^{w_2 * 1} * e^{w_3 * 1}$$

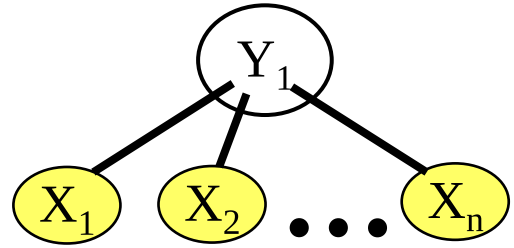
$$e^{w_0} * e^{w_1 * x_1} * e^{w_2 * x_2} * e^{w_3 * x_3} =$$

$$= e^{w_0 + \sum w_i x_i}$$

Let's derive the probabilities we need

$$\phi_i(X_i, Y_1) = \exp\{w_i \mathbb{1}\{X_i = 1, Y_1 = 1\}\}$$

$$\phi_0(Y_1) = \exp\{w_0 \mathbb{1}\{Y_1 = 1\}\}$$



$$\tilde{P}(Y_1 = 0, x_1, x_2, \ldots, x_n) = \Phi_0(Y_1) * \prod_{i=1}^{n} \Phi_i(x_i, Y_1)$$

A. 1

B. e^{w_0}

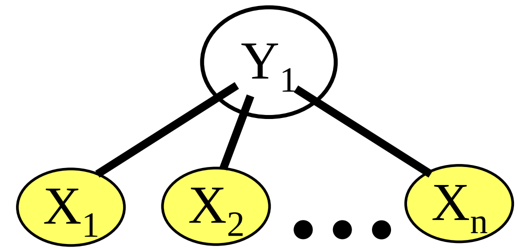
C. 0

D. $e^{\sum_{1}^{n} w_i}$

Let's derive the probabilities we need

(a) $\tilde{P}(Y_1 = 1, x_1, \ldots, x_n) = \exp(w_0 + \sum_{i=1}^{n} w_i x_i)$

(b) $\tilde{P}(Y_1 = 0, x_1, \ldots, x_n) = 1$



$$P(Y_1 = 1 \mid x_1, \ldots, x_n) = \frac{\tilde{P}(Y_1 = 1, x_1, \ldots, x_n)}{P(x_1, \ldots, x_n)}$$

← sum of (a) and (b)

$$= \frac{\exp(w_0 + \sum w_i x_i)}{1 + \exp(w_0 + \sum w_i x_i)}$$

z

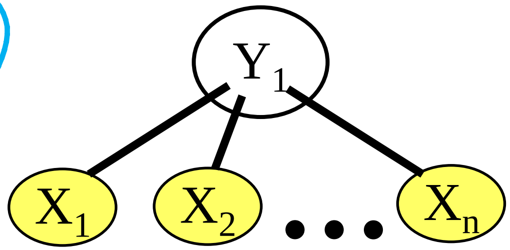
$$\frac{e^{z}}{1 + e^{z}} \quad \text{or} \quad \frac{1}{e^{-z} + 1}$$

sigmoid function

Let's derive the probabilities we need

(a) $\tilde{P}(Y_1 = 1, x_1, \ldots, x_n) = \exp\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$ ← z

(b) $\tilde{P}(Y_1 = 0, x_1, \ldots, x_n) = 1$



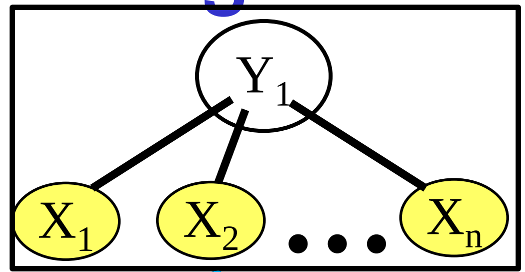
$$P(Y_1 = 1 \mid x_1, \ldots, x_n) = \frac{\tilde{P}(Y_1=1, x_1, \ldots, x_n)}{P(x_1, \ldots, x_n)}$$

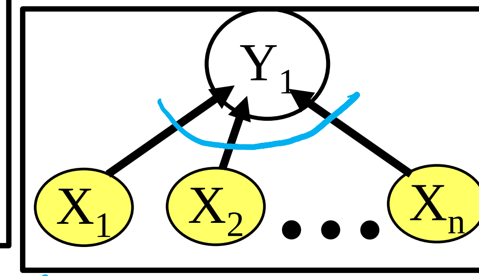
← sum of (a) and (b)

$$= \frac{e^{z}}{1 + e^{z}} \cdot \frac{e^{-z}}{e^{-z}}$$

$$\frac{1}{e^{-z} + 1}$$

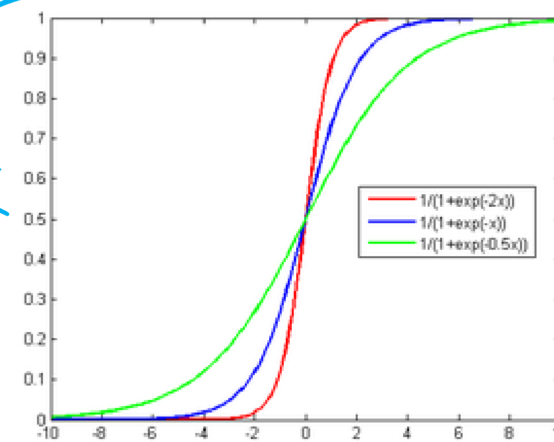
Sigmoid Function used in Logistic Regression

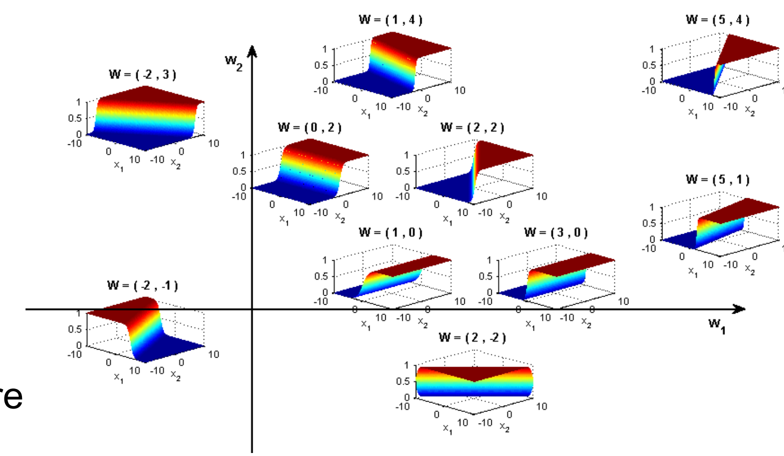




- Great practical interest
- Number of param w_i is linear instead of exponential in the number of parents
- Natural model for many real-world applications
- Naturally aggregates the influence of different parents

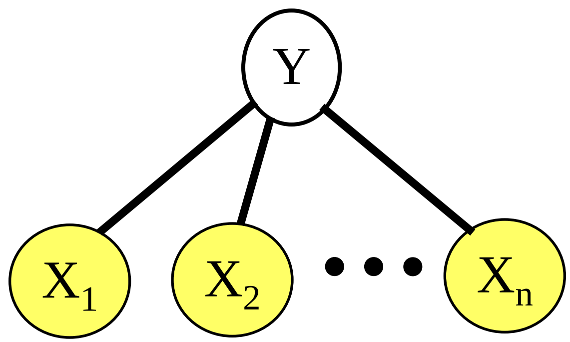
$\frac{1}{1+e^{-x}}$





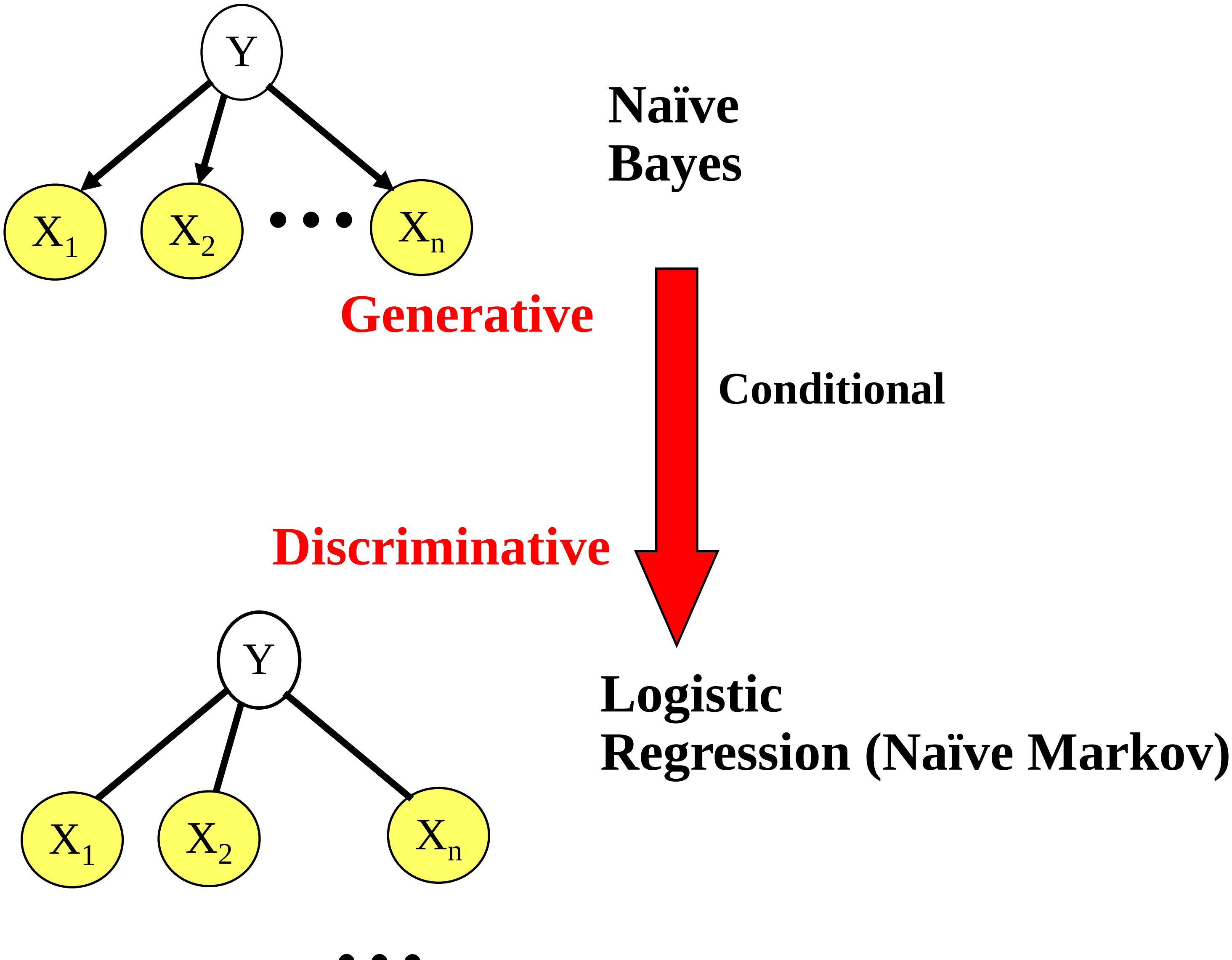
Logistic Regression as a Markov Net (CRF)

Logistic regression is a simple Markov Net (a CRF) *aka* **naïve markov model**



- But only models the **conditional distribution**, P(Y | $\mathbf{X}$) and not the full joint P(X, Y)

Naïve Bayes vs. Logistic Regression



Learning Goals for today's class

➢You can:

- Perform Exact and Approx. Inference in Markov Networks
- Describe a few applications of Markov Networks
- Describe a natural parameterization for a Naïve Markov model (which is a simple CRF)
- Derive how P(Y|X) can be computed for a Naïve Markov model
- Explain the discriminative vs. generative distinction and its implications

Next class Fri   Linear-chain CRFs

To Do   Revise generative temporal models (HMM)

Midterm, Mon, Oct 26,
we will start at 9am sharp

How to prepare….

- Work on **practice material** posted on Connect
- **Learning Goals** (look at the end of the slides for each lecture – or complete list on Connect)
- Go to **Office Hours** (Ted is offering and extra slot on Fri check Piazza
- Revise all the **clicker questions** and **practice exercises**

Midterm, Mon, Oct 26,
we will start at 9am sharp

How to prepare….

- Keep Working on **assignment-2** !
- Go to **Office Hours**
- **Learning Goals** (look at the end of the slides for each lecture – will post complete list)
- Revise all the **clicker questions** and **practice exercises**
- Will post **more practice material** today

Generative vs. Discriminative Models

Generative models (like Naïve Bayes): *not* directly designed to maximize performance on classification. They model the *joint distribution* $P(X, Y)$.

Classification is then done using Bayesian inference

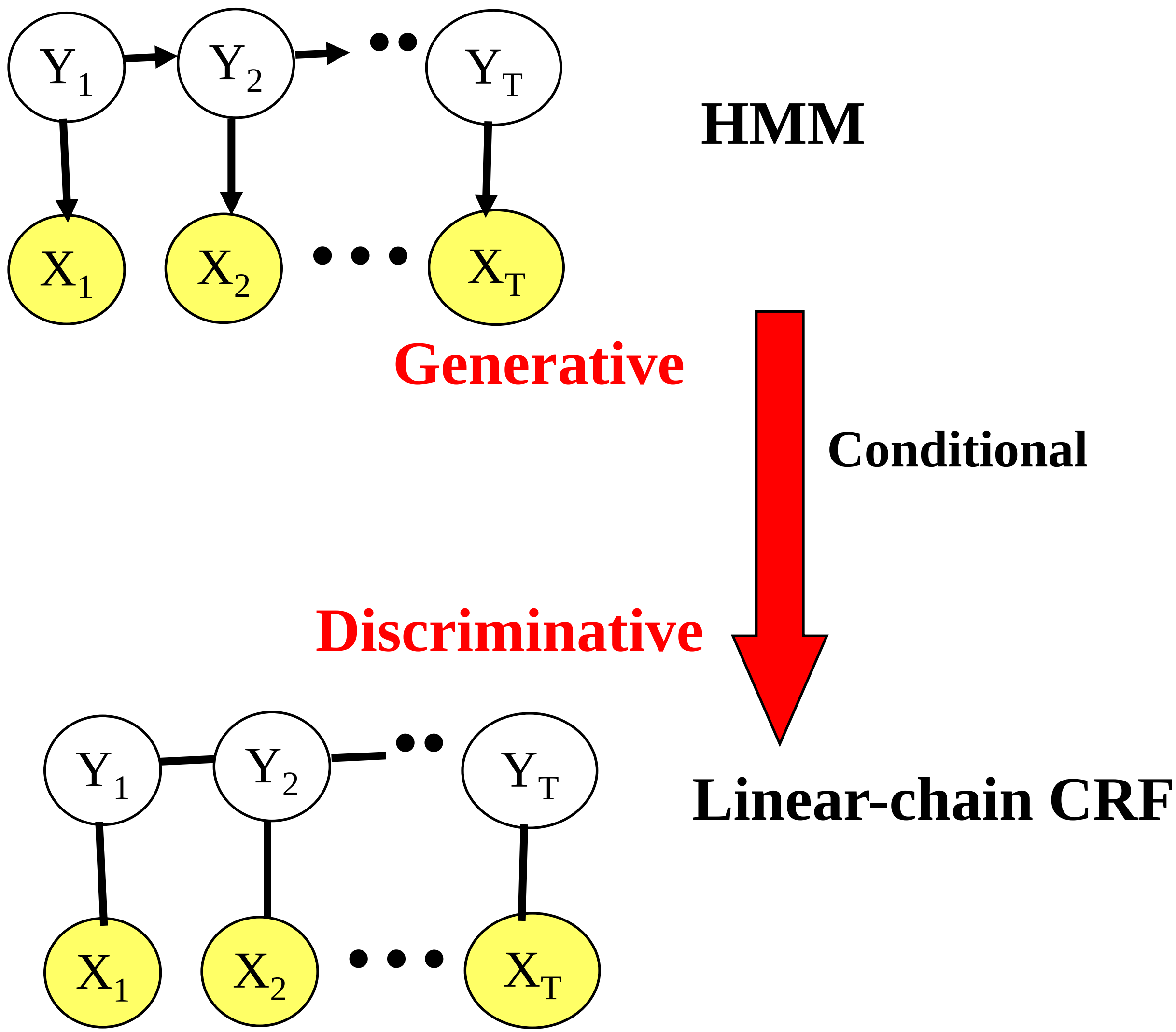
But a generative model can also be used to perform any other inference task, e.g. $P(X_1 \mid X_2, \ldots X_n,)$

- "Jack of all trades, master of none."

Discriminative models (like CRFs): specifically designed and trained to maximize performance of classification. They only model the *conditional distribution* $P(Y \mid X)$.

By focusing on modeling the conditional distribution, they generally perform better on classification than generative models when given a reasonable amount of training data.

On Fri: Sequence Labeling



Lecture Overview

- Indicator function
- P(X,Y) vs. P(X|Y) and Naïve Bayes
- Model P(Y|X) explicitly with Markov Networks
  - Parameterization
  - Inference
- Generative vs. Discriminative models

P(**X**,**Y**) vs. P(**Y**|**X**)

Assume that you always observe a set of variables
$\mathbf{X} = \{X_1 \ldots X_n\}$
and you want to predict one or more variables
$\mathbf{Y} = \{Y_1 \ldots Y_m\}$

You can model P(**X**,**Y**) and then infer P(**Y**|**X**)

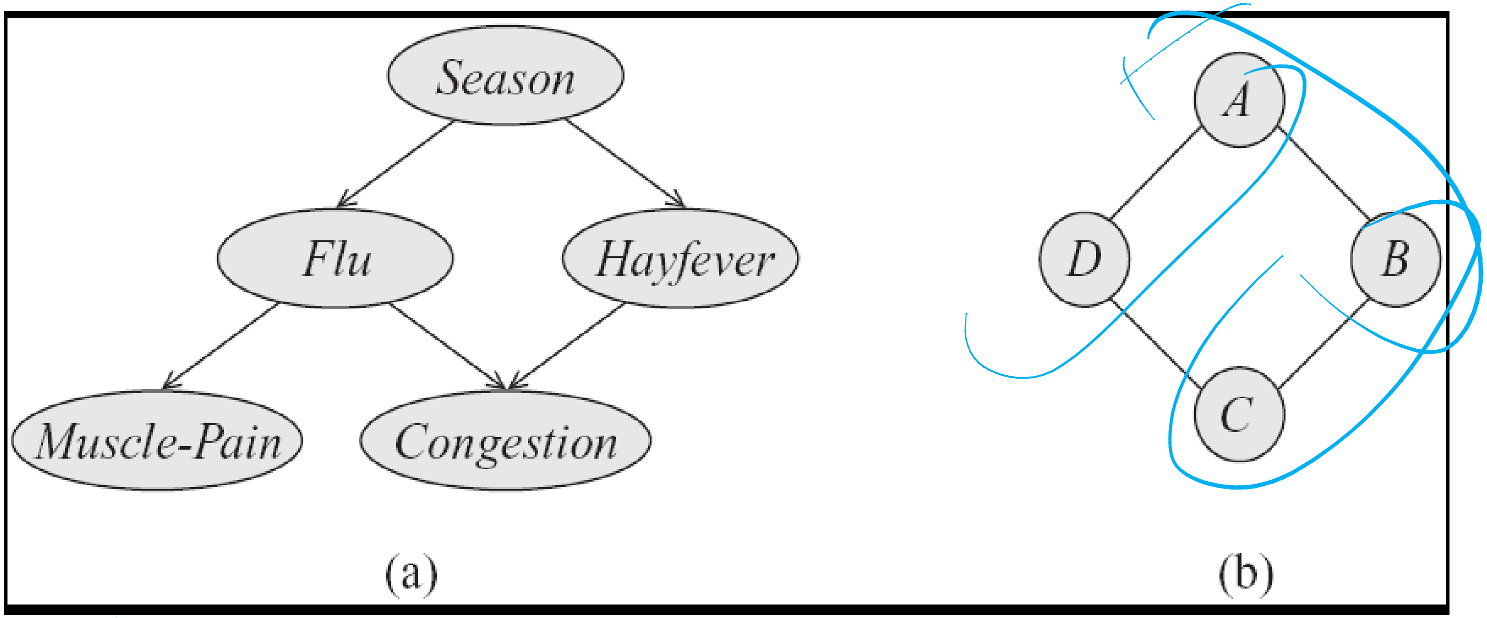
P(X,Y) vs. P(Y|X)

With a **Bnet** we can represent a joint as the product of Conditional Probabilities

With a **Markov Network** we can represent a joint a the product of **Factors**

We will see that **Markov Network** are also suitable for representing the conditional prob. P(Y|X) directly

Directed vs. Undirected



$$P(S,F,H,M,C) = P(S) * P(F|S) * P(H|S) * P(M|F) * P(C|FH)$$

$$P(ABCD) = \frac{1}{Z} \Phi_1(AB) * \Phi_2(BC) * \Phi_3(CD) * \Phi_4(AD)$$

Factorization

Naïve Bayesian Classifier P(Y,**X**)

A very simple and successful Bnets that allow to classify entities in a set of classes Y_1, given a set of features ($X_1 \ldots X_n$)

Example:

- Determine whether an email is spam (only two classes *spam*=T and *spam*=F)
- Useful attributes of an email ?

words contained in the email

Assumptions

- The value of each attribute depends on the classification
- **(Naïve)** The attributes are independent of each other given the classification

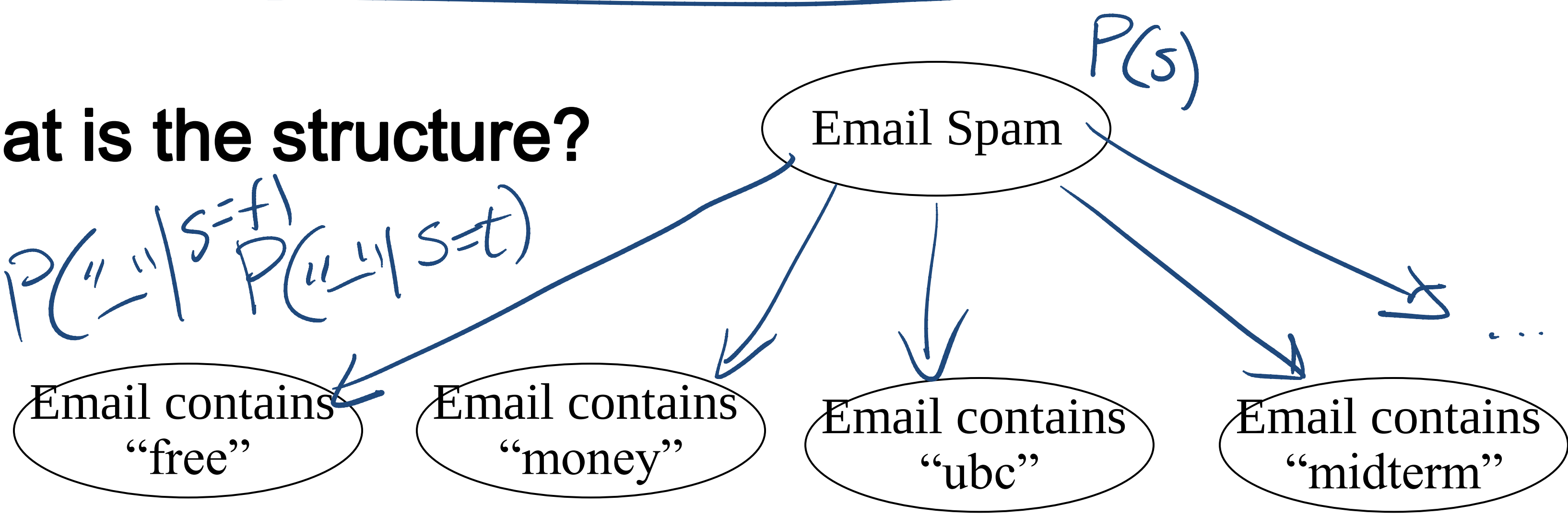
P("bank" | "account" , spam=T) = P("bank" | spam=T)

Naïve Bayesian Classifier for Email Spam

The corresponding Bnet represent : P(Y1, X1…Xn)

- **What is the structure?**

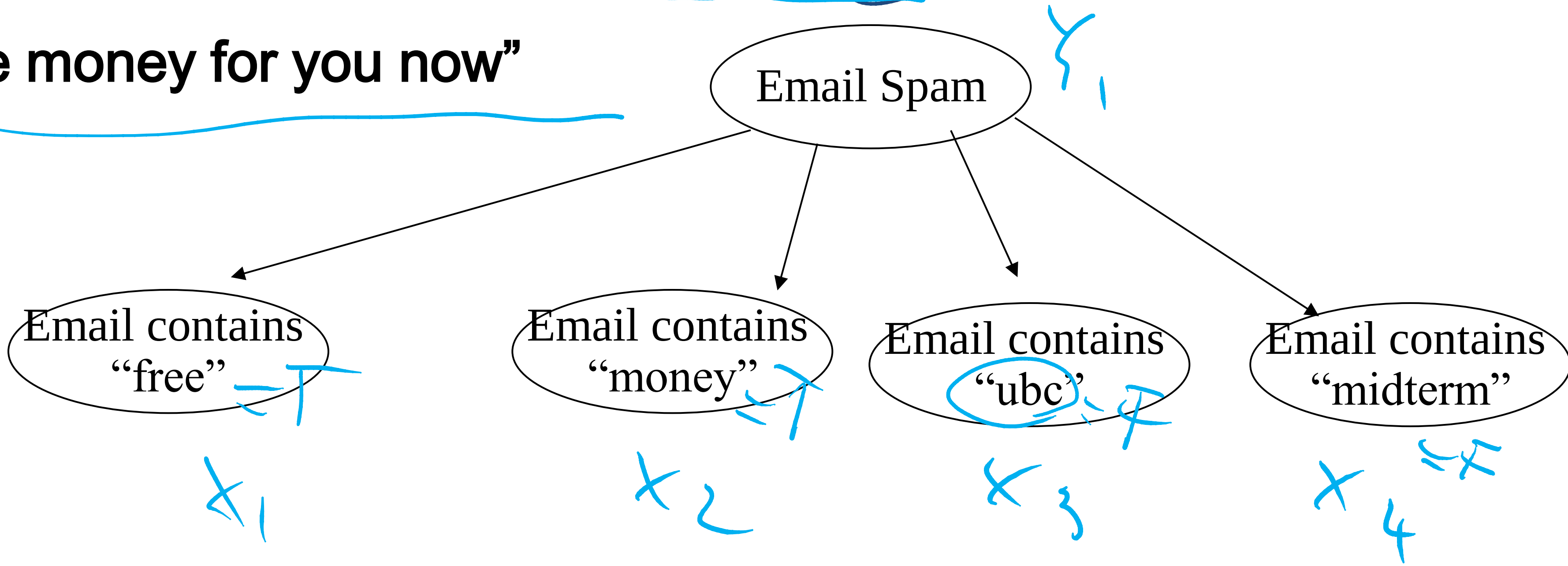
words



NB Classifier for Email Spam: Usage

Can we derive : $P(Y_1 | X_1 \ldots X_n)$ for any $x_1 \ldots x_n$

"free money for you now"

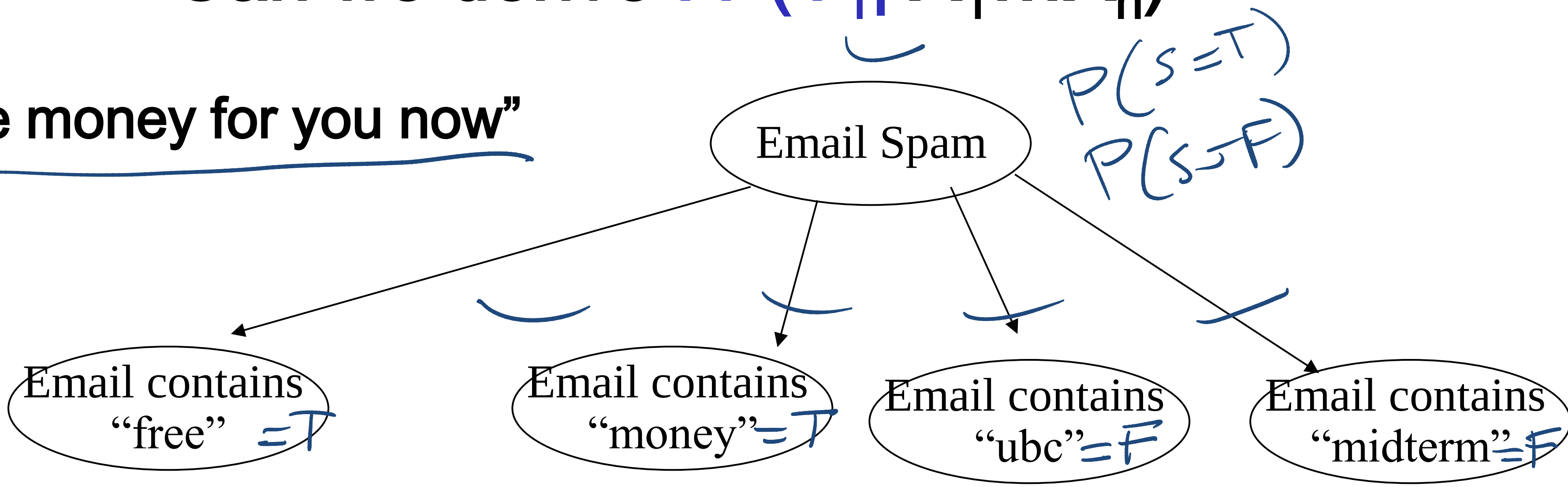


But you can also perform any other inference…
e.g., $P(X_1 | X_3 \quad)$

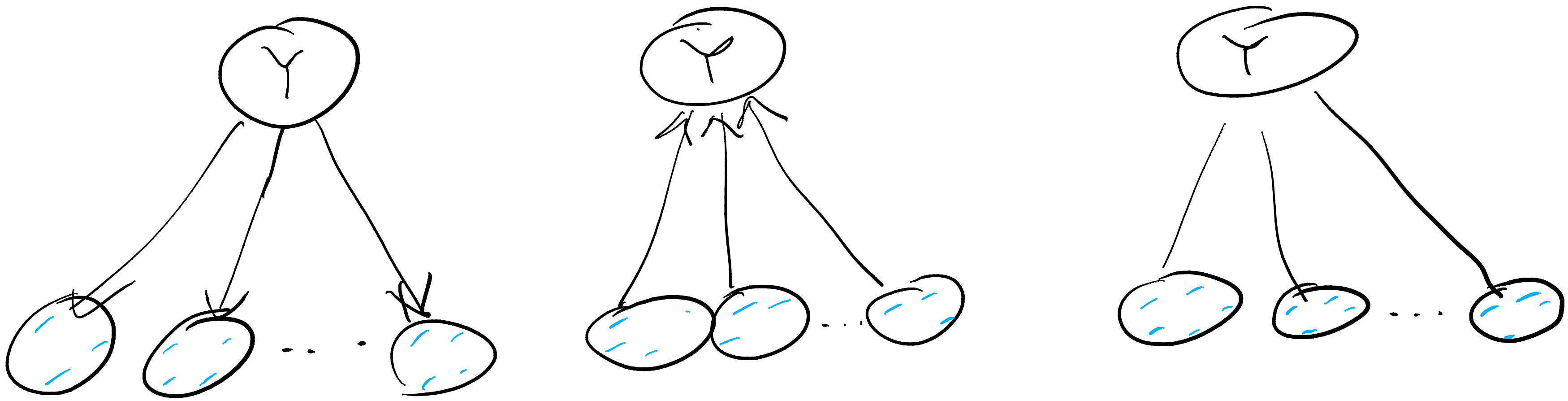
NB Classifier for Email Spam: Usage

Can we derive : $P(Y_1 | X_1 \dots X_n)$

"free money for you now"



But you can perform also any other inference e.g., $P(X_1 | X_3 \quad)$



Y
Y
Y