Introrudction to Probabilistic Graphical Model

Modeling, Inference, and Learning

Overview

- What’s Probabilistic Graphical Model for ?

- Tasks in Graphical Model:
  - Modeling
  - Learning
  - Inference

- Examples
  - Topic Model
  - Hidden Markov Model
  - Markov Random Field

Overview

- What's Probabilistic Graphical Model for ?
- Tasks in Graphical Model:
  - Modeling
  - Inference
  - Learning
- Examples
  - Topic Model
  - Hidden Markov Model
  - Markov Random Field

What’s PGM for ?

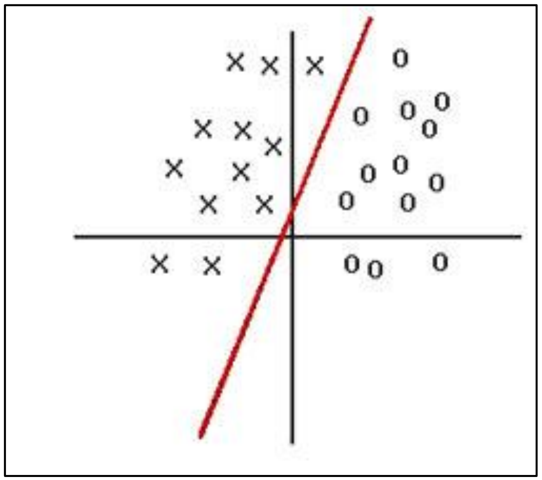
- Use “Probability” to “Model” dependencies among target of interest as a “Graph”. A unified framework for :
  - **Prediction** (Classification / Regression)
  - **Discovery** (Clustering / Pattern Recognition / System Modeling)
  - **State Tracking** (Localization/ Monitoring/ MotionTracking)
  - **Ranking** (Search Engine/ Recommendation for Text/ Image/ Item)

What’s PGM for ?

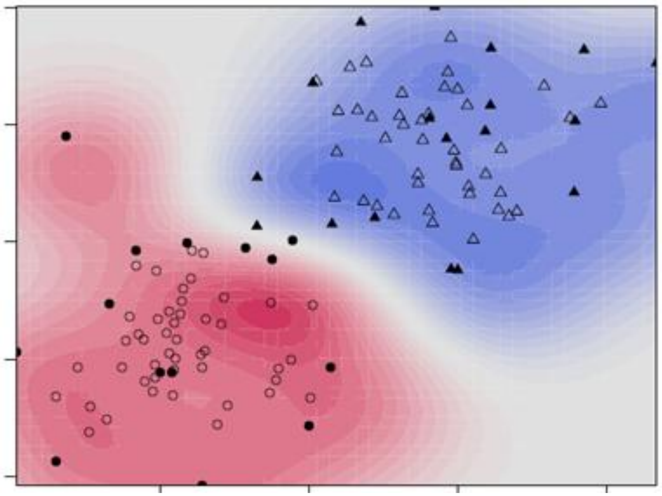
- Use “Probability” to “Model” dependencies among target of interest as a “Graph”. A unified framework for :
  - **Prediction** (Classification / Regression)
  - **Discovery** (Clustering / Pattern Recognition / System Modeling)
  - **State Tracking** (Localization/ Monitoring/ MotionTracking)
  - **Ranking** (Search Engine/ Recommendation for Text/ Image/ Item)

Prediction (Lectures Before …)

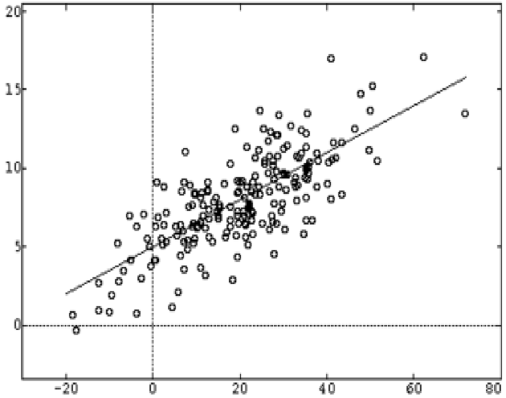
Variables of interest ?



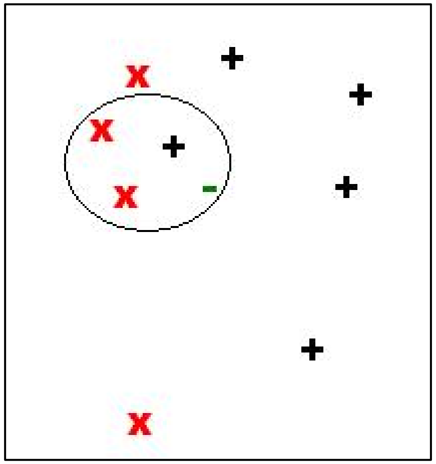
Perceptron



SVM



Linear Regression



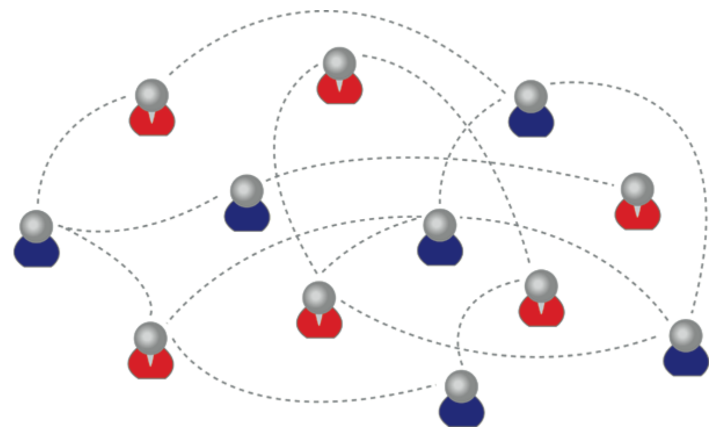
K - Nearest Neighbor

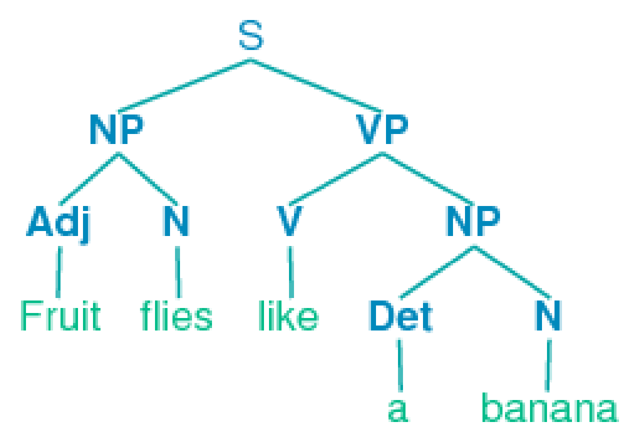
Prediction with PGM

One prediction are dependent on others.
(Collective Classification)

Data:

$(x_1,x_2,\dots.,x_d, y)$
$(x_1,x_2,\dots.,x_d, y)$
$(x_1,x_2,\dots.,x_d, y)$
…
$(x_1,x_2,\dots.,x_d, y)$



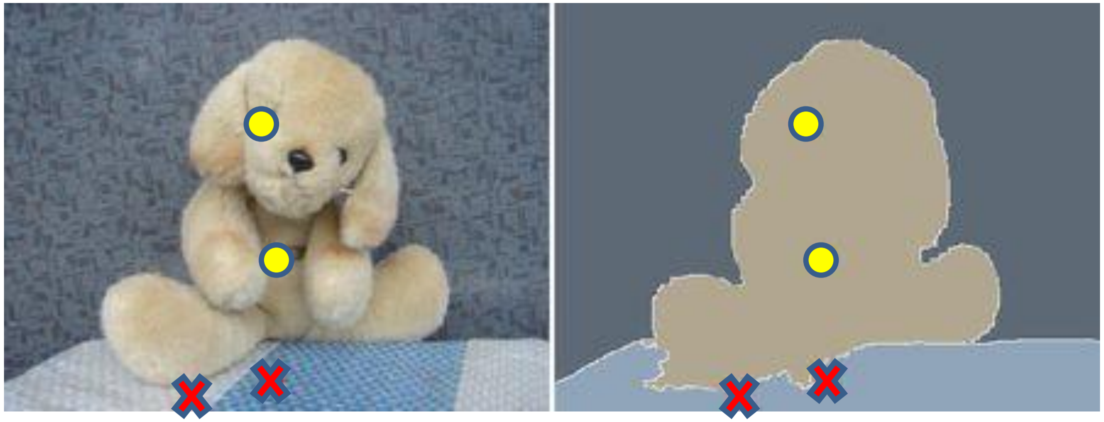


Prediction with PGM

Labels are missing for most of data.
(Semi-supervised Learning)

Data:

$(x_1, x_2, \ldots, x_d, y=0)$
$(x_1, x_2, \ldots, x_d, y=1)$
$(x_1, x_2, \ldots, x_d, y=?)$
…
$(x_1, x_2, \ldots, x_d, y=?)$



What's PGM for ?

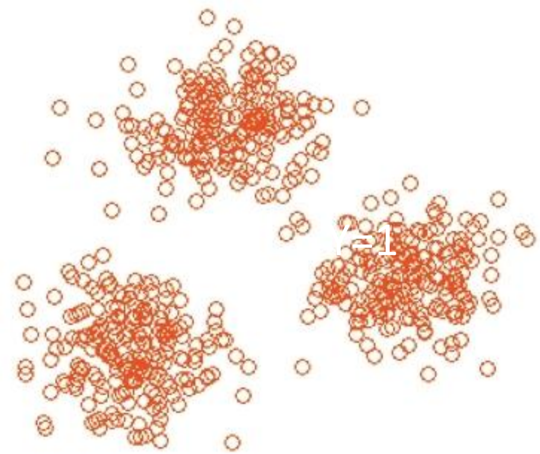
- Use "Probability" to "Model" dependencies among target of interest as a "Graph" . A unified framework for :
  - **Prediction** (Classification / Regression)
  - **Discovery** (Clustering / Pattern Recognition / System Modeling)
  - **State Tracking** (Localization/ Monitoring/ MotionTracking)
  - **Ranking** (Search Engine/ Recommendation for Text/ Image/ Item)

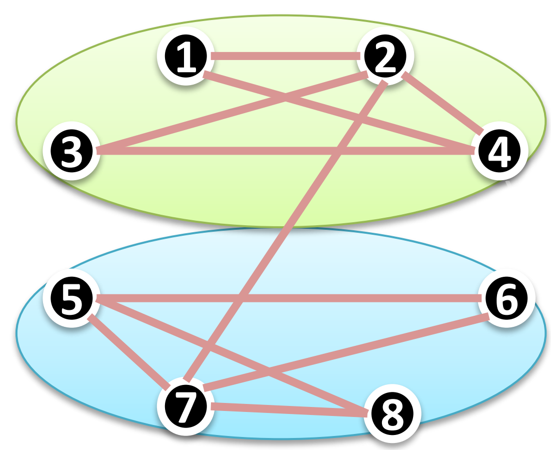
Discovery with PGM

We are interested about hidden variables.
(Unsupervised Learning)

Data:

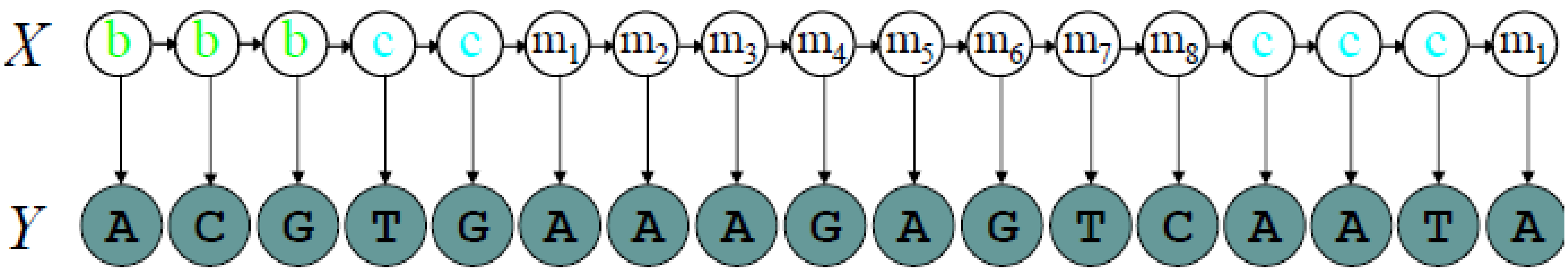
$(x_1, x_2, \ldots, x_d, y=?)$
$(x_1, x_2, \ldots, x_d, y=?)$
…
…
$(x_1, x_2, \ldots, x_d, y=?)$

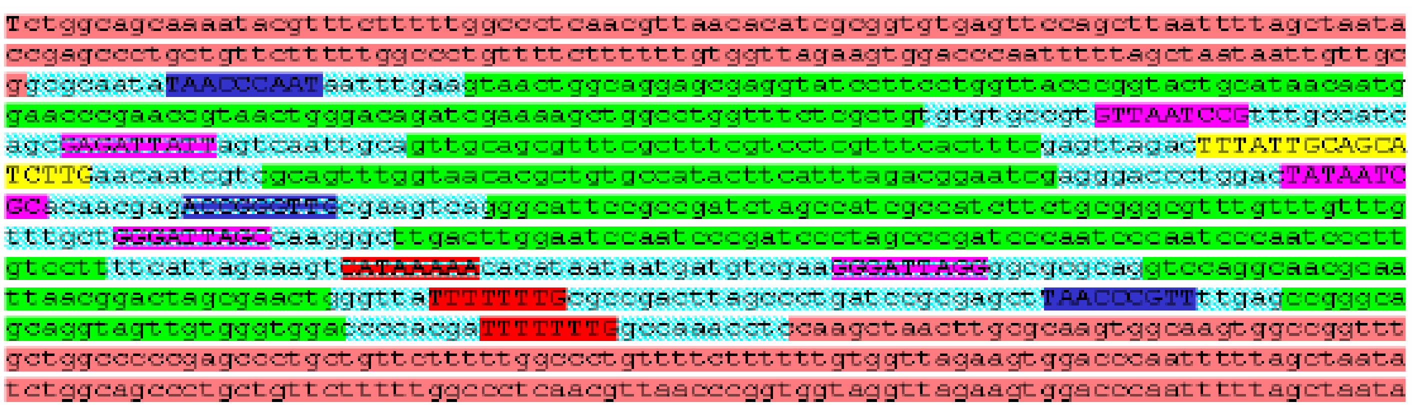




Discovery with PGM

Pattern Mining / Pattern Recognition.





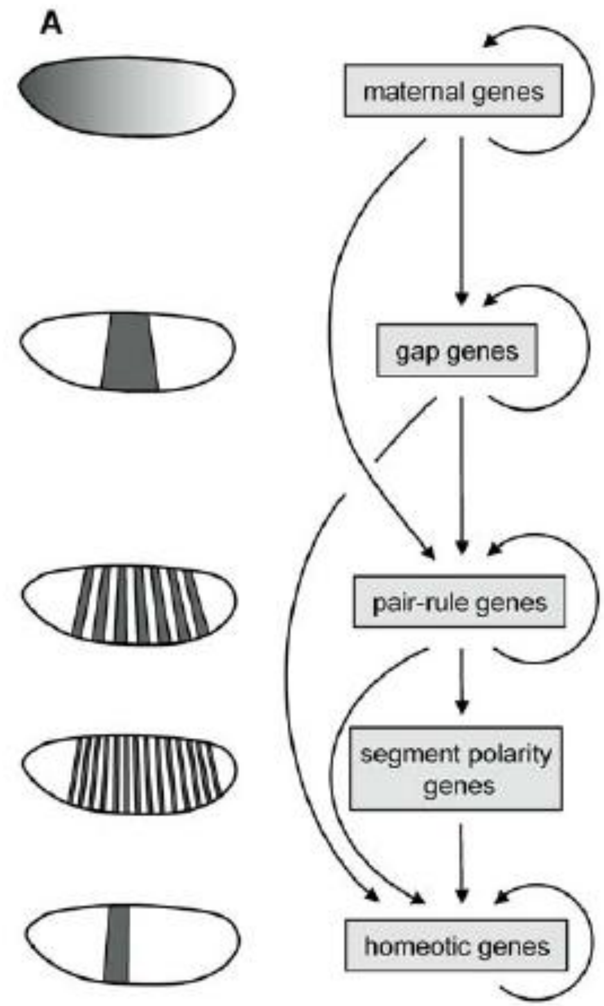
Discovery with PGM

Learn a whole picture of the problem domain.

Some domain knowledge in hand about the generating process.

Data:

$(x_1,x_2,\ldots,x_d)$
$(x_1,x_2,\ldots,x_d)$
…
…
$(x_1,x_2,\ldots,x_d)$



What’s PGM for ?

- Use “Probability” to “Model” dependencies among target of interest as a “Graph” . A unified framework for :
  - **Prediction** (Classification / Regression)
  - **Discovery** (Clustering / Pattern Recognition / System Modeling)
  - **State Tracking** (Localization/ Mapping/ MotionTracking)
  - **Ranking** (Search Engine/ Recommendation for Text/ Image/ Item)

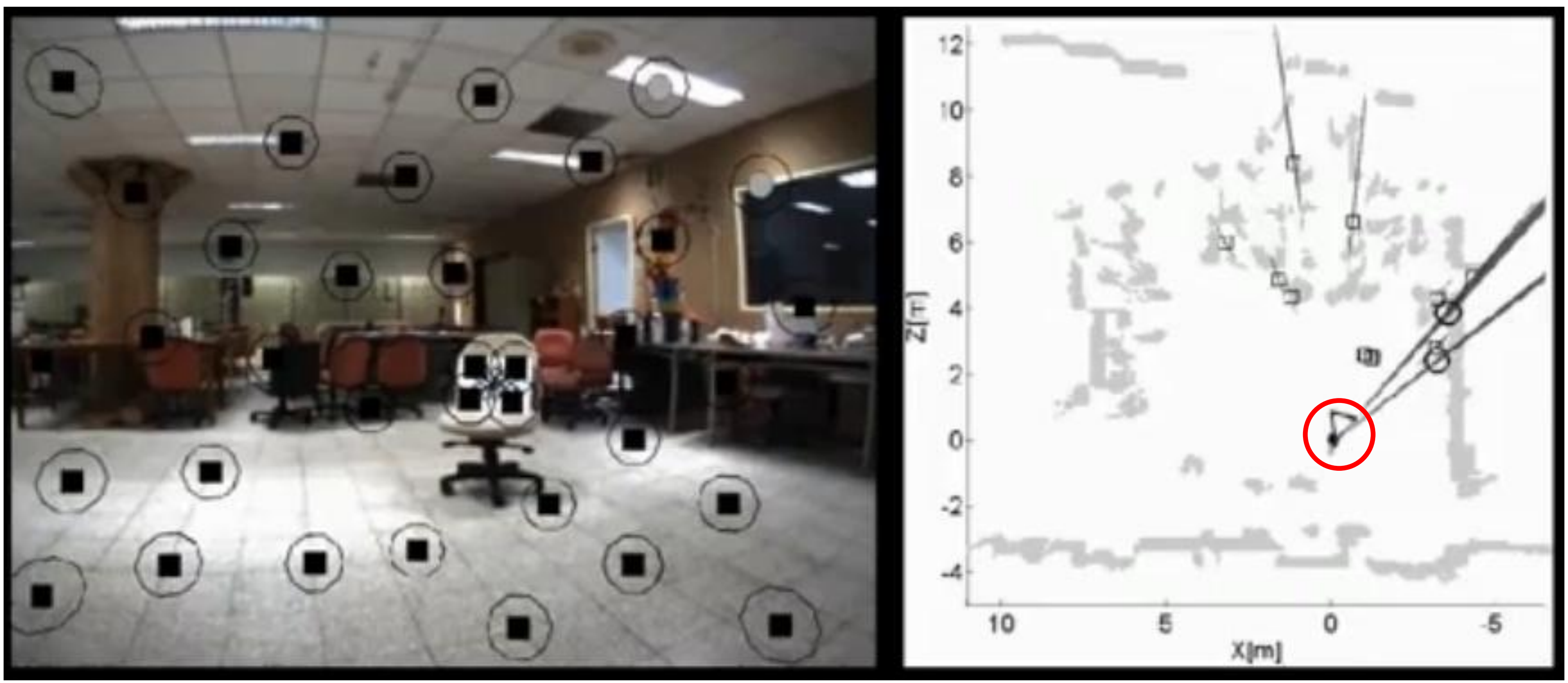
State Tracking with PGM

Given Observation from Sensors (ex. Camera, Laser, Sonar …….)

Localization : Locate sensor itself.

Mapping : Locate Feature Points of environment.

Tracking: Locate Moving Object in the environment.



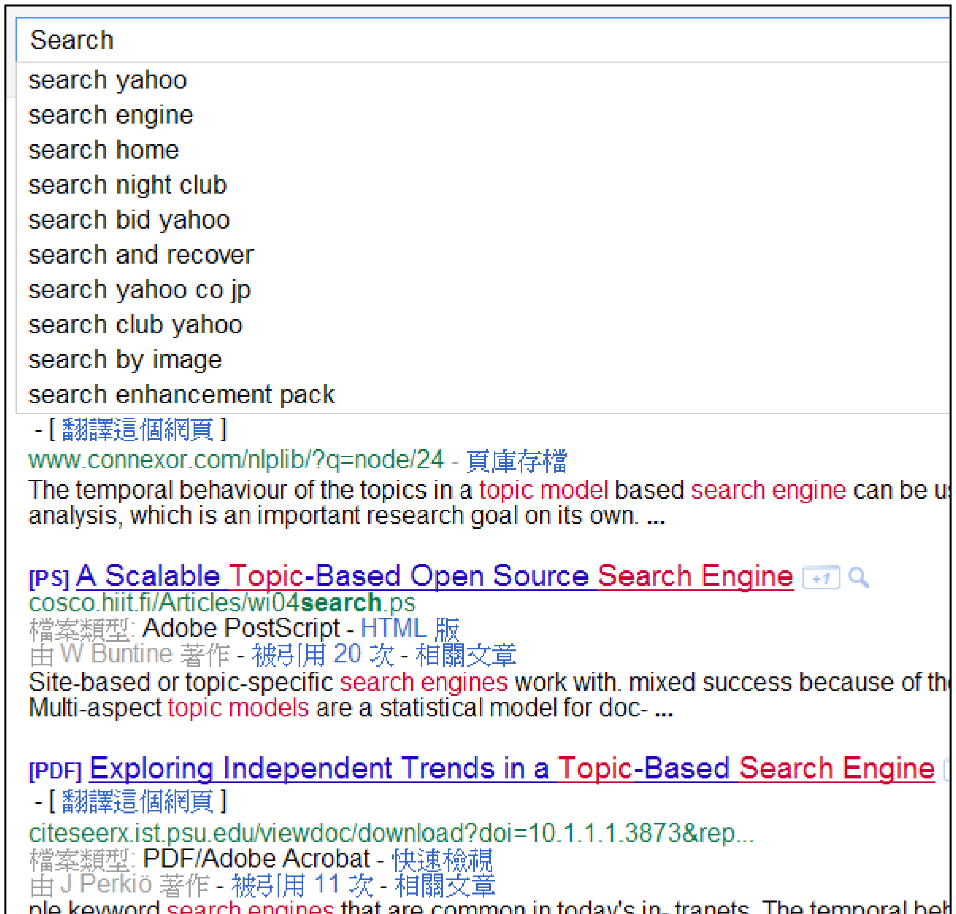
What’s PGM for ?

- Use “Probability” to “Model” dependencies among target of interest as a “Graph” . A unified framework for :
  - **Prediction** (Classification / Regression)
  - **Discovery** (Clustering / Pattern Recognition / System Modeling)
  - **State Tracking** (Localization/ Mapping/ MotionTracking)
  - **Ranking** (Search Engine/ Recommendation for Text/ Image/ Item)

Ranking with PGM

Who needs the scores ? → Search Engine / Recommendation.

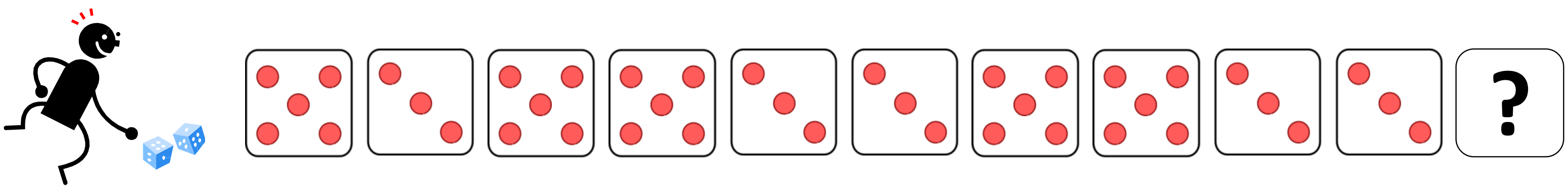
Variables of interest ?



Overview

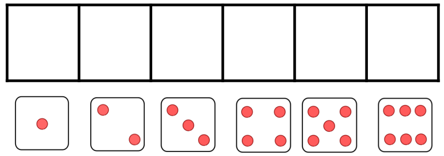
- What's Probabilistic Graphical Model for ?
- Tasks in Graphical Model:
  - Modeling
  - Learning
  - Inference
- Examples
  - Topic Model
  - Hidden Markov Model
  - Markov Random Field

A Simple Example

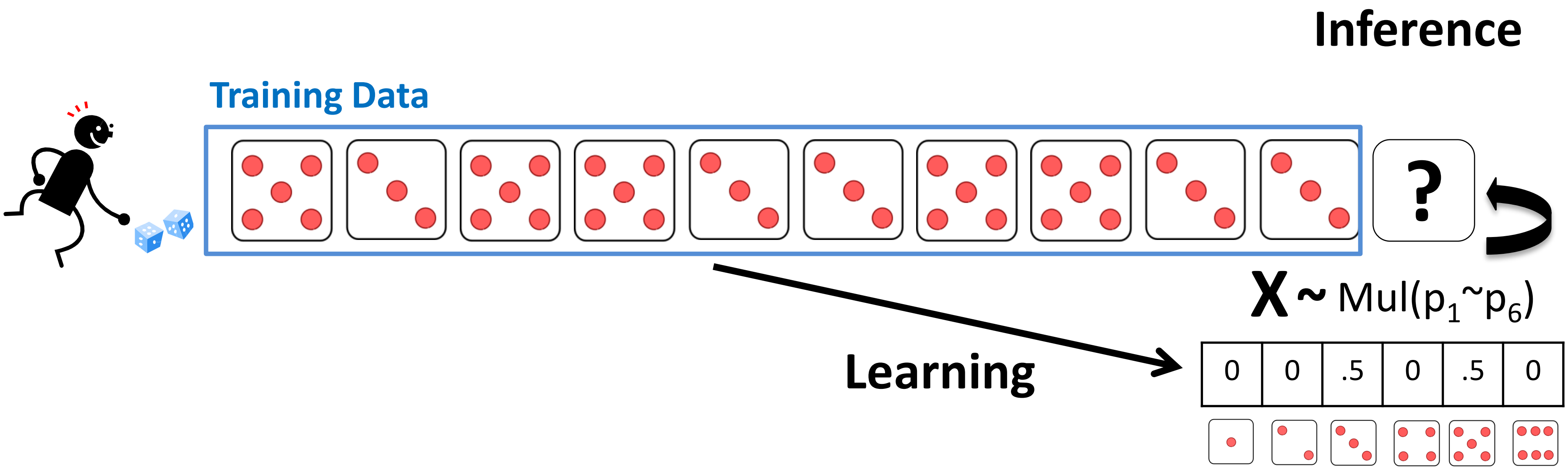


$X \sim \mathrm{Mul}(p_1 \sim p_6)$

Modeling

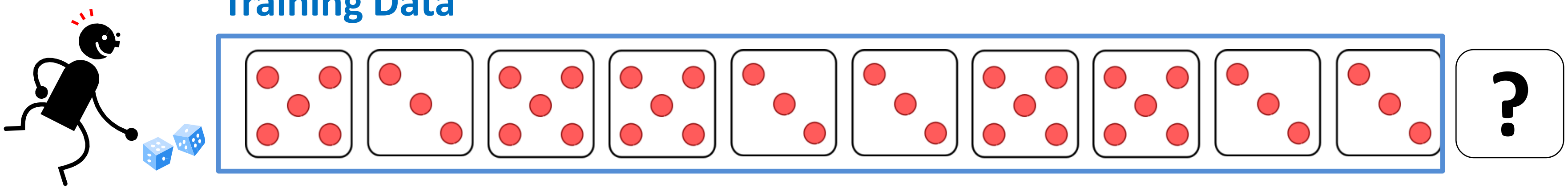


A Simple Example



A Simple Example

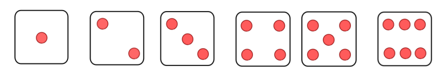
Training Data



$X \sim \mathrm{Mul}(p_1 \sim p_6)$

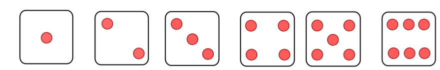
Is this the best model ? Why not

.1	.1	.3	.1	.3	.1



?

0	0	.5	0	.5	0



Maximum Likelihood Estimation (MLE) Criteria :

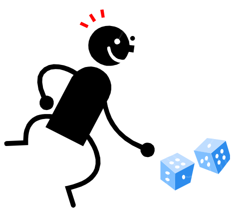
$$\text{Best Model} = \underset{\text{model}}{\text{argmax}}\ \text{Likelihood(Data)} = P(\text{Data}|\text{model}) = P(X_1) * P(X_2) * \ldots\ldots * P(X_{10})$$

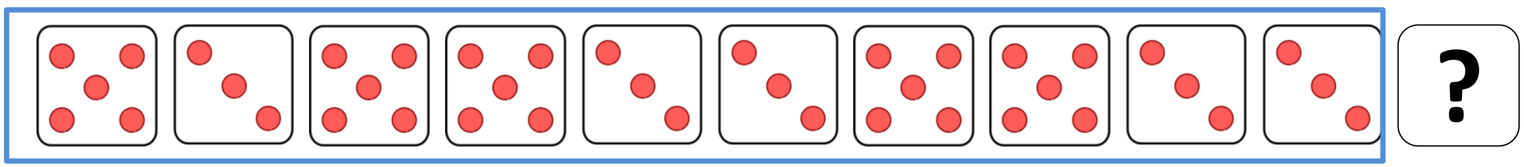
$$= (p_3)^5 * (p_5)^5 \quad \text{Sub. to } p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 1$$

$$\rightarrow\ p_3 = 5/(5+5)\ ,\ p_5 = 5/(5+5)$$

A Simple Example

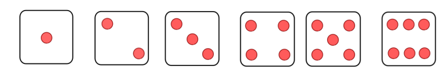
Training Data





$\mathbf{X} \sim \text{Mul}(p_1 \sim p_6)$

0	0	.5	0	.5	0



Is the "MLE" model best ?

Compute "**Likelihood**" on Testing Data.

Testing Data

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{X_5}$

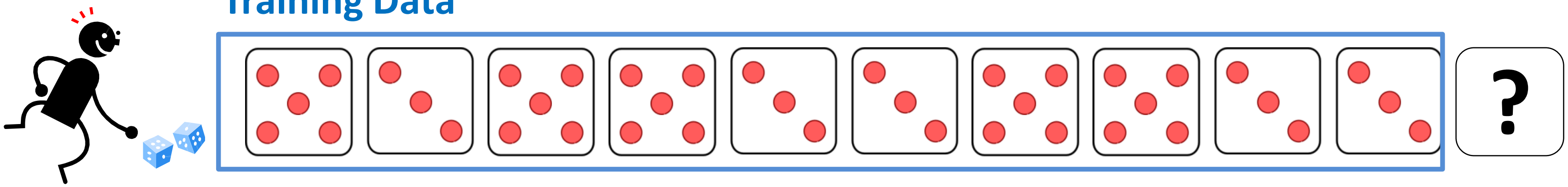
$$P(\text{ Data } | \text{ Your Model }) = P(X_1)^* \; P(X_2)^* \; P(X_3)^* \; P(X_4)^* \; P(X_5)$$

"**Likelihood**" tends to overflow so practically using "**Log Likelihood**":

$$\mathbf{ln[}\; P(\text{ Data } | \text{ Your Model }) \;\mathbf{]} = \mathbf{ln}P(X_1) + \mathbf{ln}P(X_2) + \mathbf{ln}P(X_3) + \mathbf{ln}P(X_4) + \mathbf{ln}P(X_5)$$

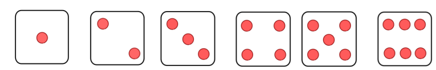
A Simple Example

Training Data



$\mathbf{X} \sim \text{Mul}(p_1 \sim p_6)$

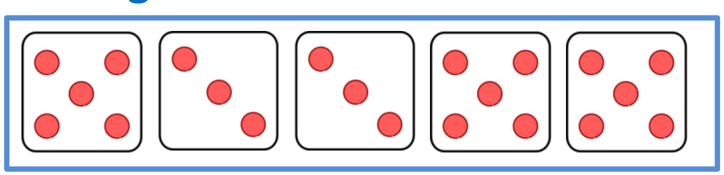
0	0	.5	0	.5	0



Is the “MLE” model best ?

Compute “**Likelihood**” on Testing Data.

Testing Data

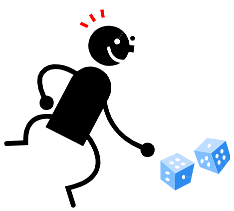


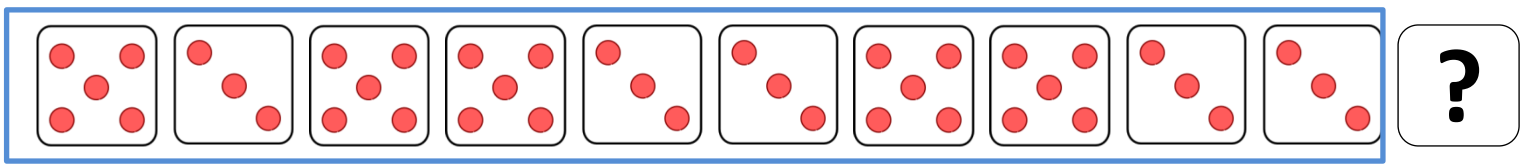
P(Data | Your Model) = .5* .5* .5* .5* .5 = 0.0312 **(1 is best)**

ln[P(Data | Your Model)] = ln(0.5) * 5 = -3.46 **(0 is best)**

A Simple Example

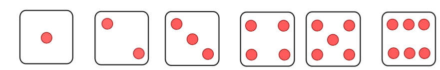
Training Data





$\mathbf{X} \sim \mathrm{Mul}(p_1 \sim p_6)$

0	0	.5	0	.5	0

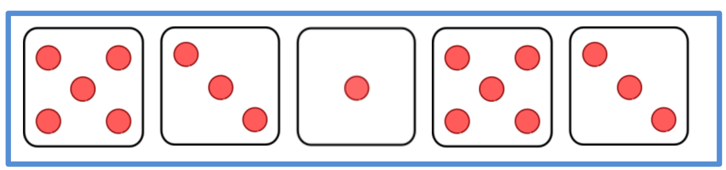


Overfit Training Data!!

Is the "MLE" model best ?

Compute "**Likelihood**" on Testing Data.

Testing Data

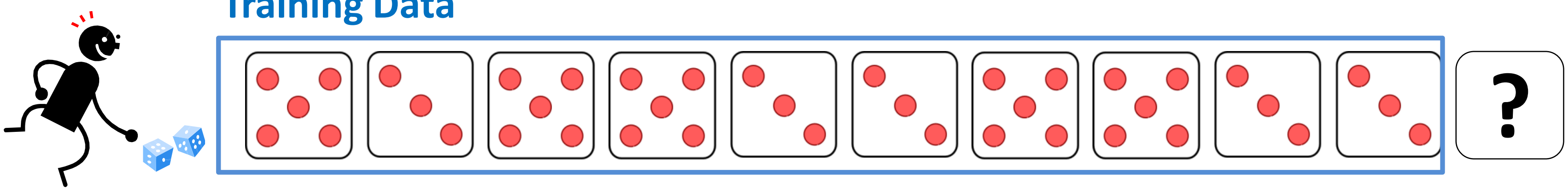


P(Data | Your Model) = .5* .5* .5* .5* .0 = 0

ln[P(Data | Your Model)] = ln(0.5) * 4 + ln(0)*1 = -∞

Bayesian Learning

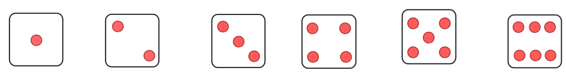
Training Data



$\mathbf{X} \sim \mathrm{Mul}(p_1 \sim p_6)$

Prior Knowledge ?

1/6	1/6	1/6	1/6	1/6	1/6



Prior = $P(p_1 \sim p_6)$ = const. * $(p_1)^1 * (p_2)^1 * (p_3)^1 * (p_4)^1 * (p_5)^1 * (p_6)^1$

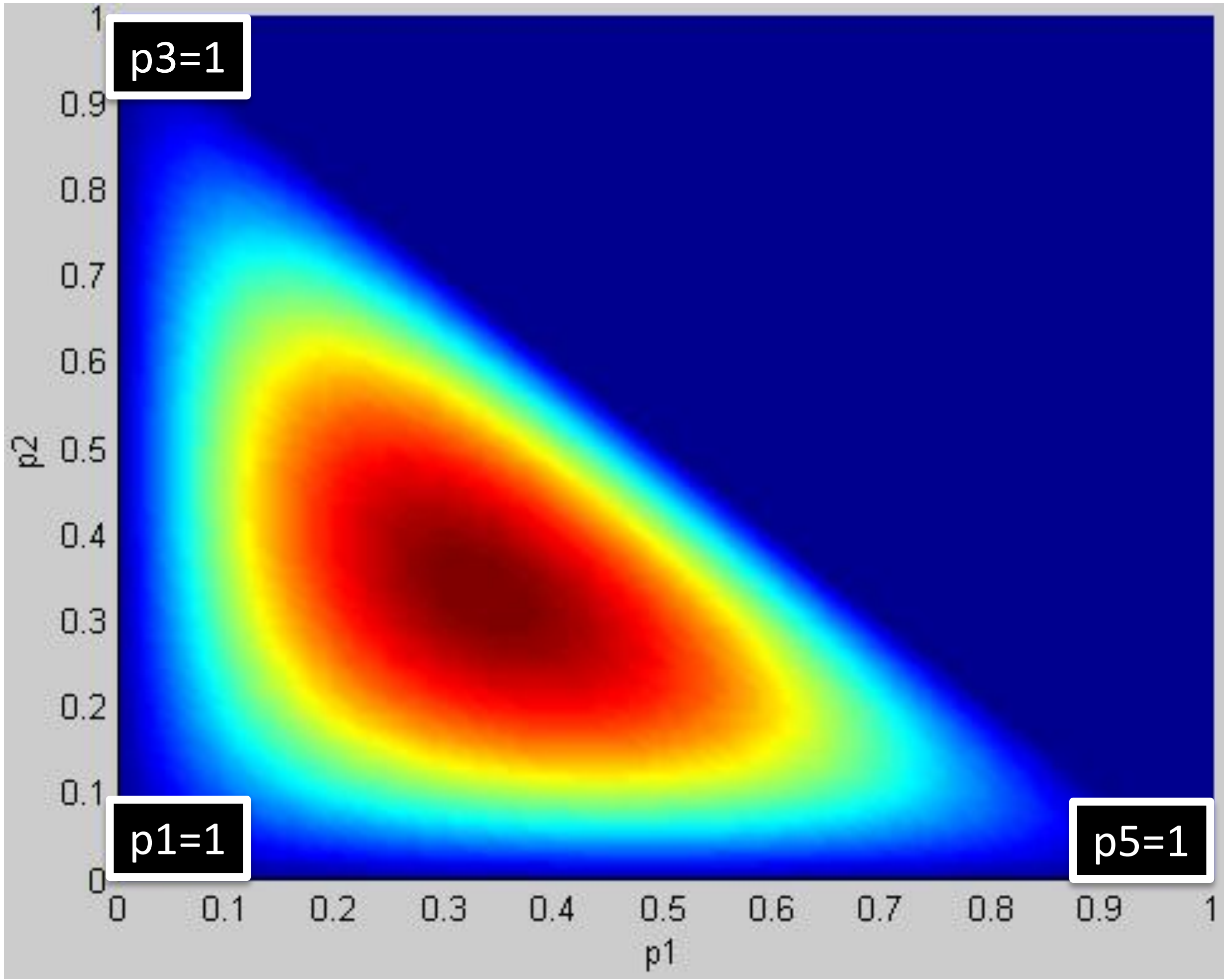
Likelihood = $P(\text{Data} | p_1 \sim p_6)$ = $(p_3)^5 * (p_5)^5$

$P(p_1 \sim p_6 | \text{Data})$ = const. * Piror * Likelihood

= const. * $(p_1)^1 * (p_2)^1 * (p_3)^{1+5} * (p_4)^1 * (p_5)^{1+5} * (p_6)^1$

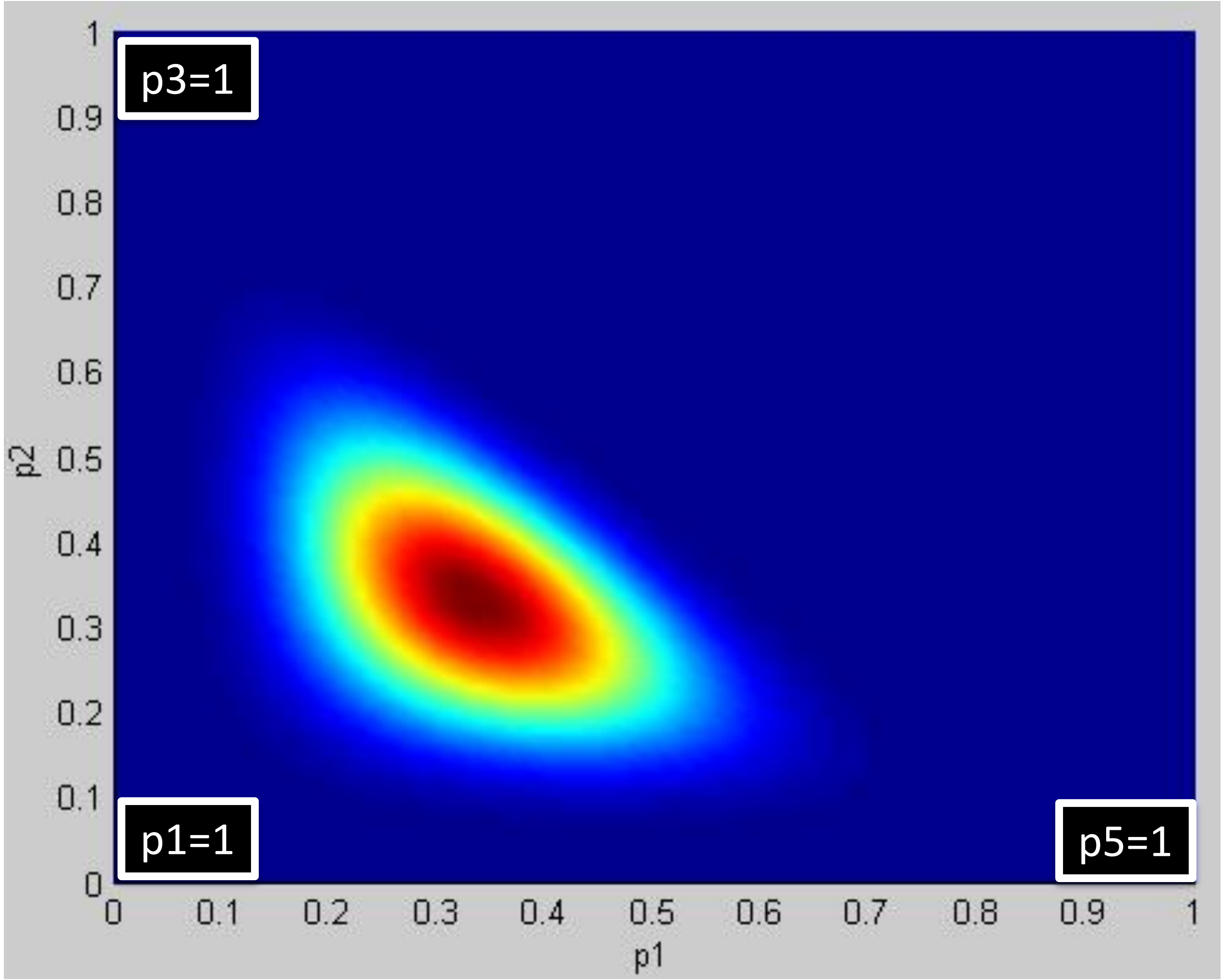
Dirichlet Distribution

$$\text{Prior} = P(p_1,p_3,p_5) = \text{const.} * (p_1)^1 * (p_3)^1 * (p_5)^1$$



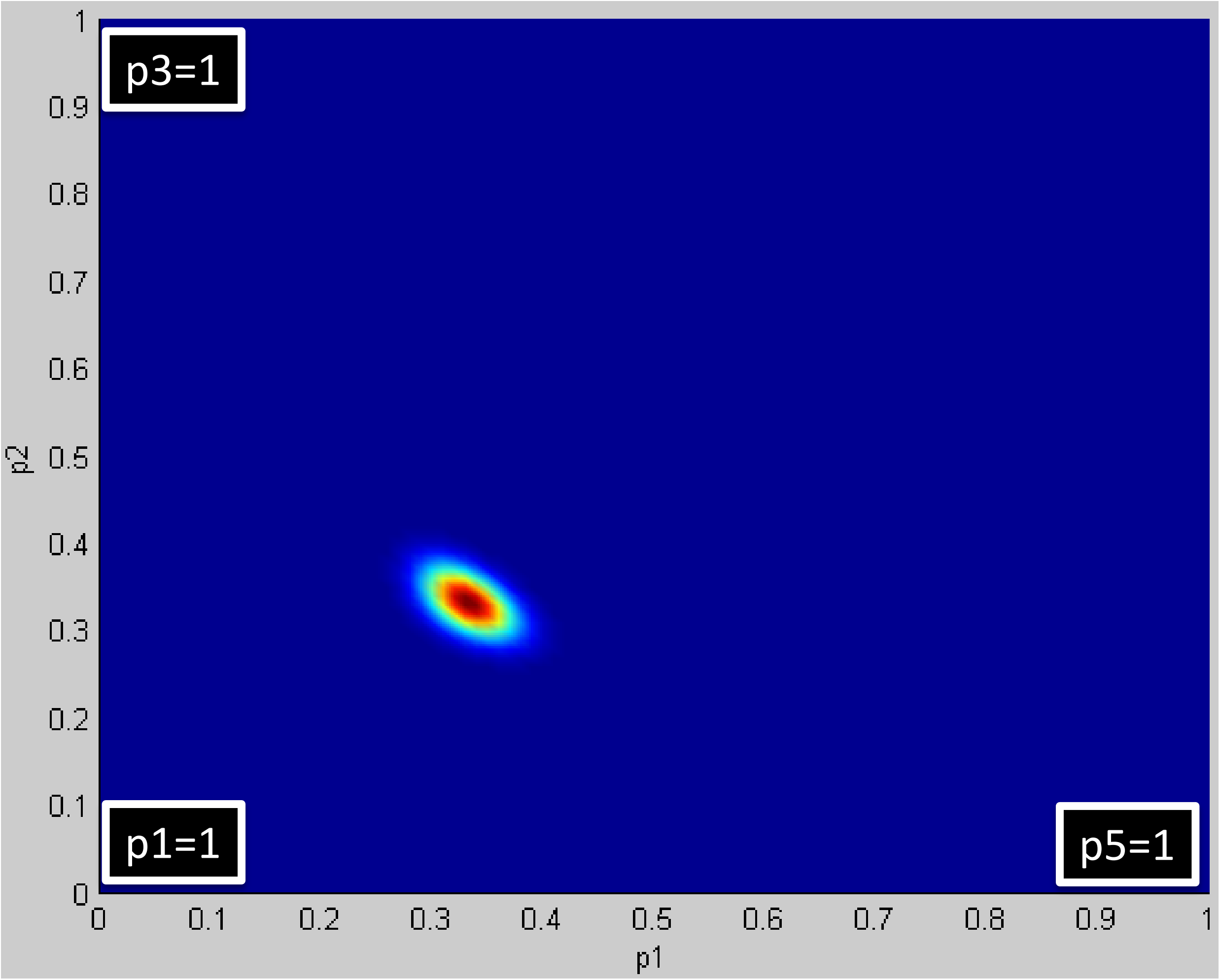
Dirichlet Distribution

Prior = $P(p_1,p_3,p_5) = \text{const.} * (p_1)^5 * (p_3)^5 * (p_5)^5$



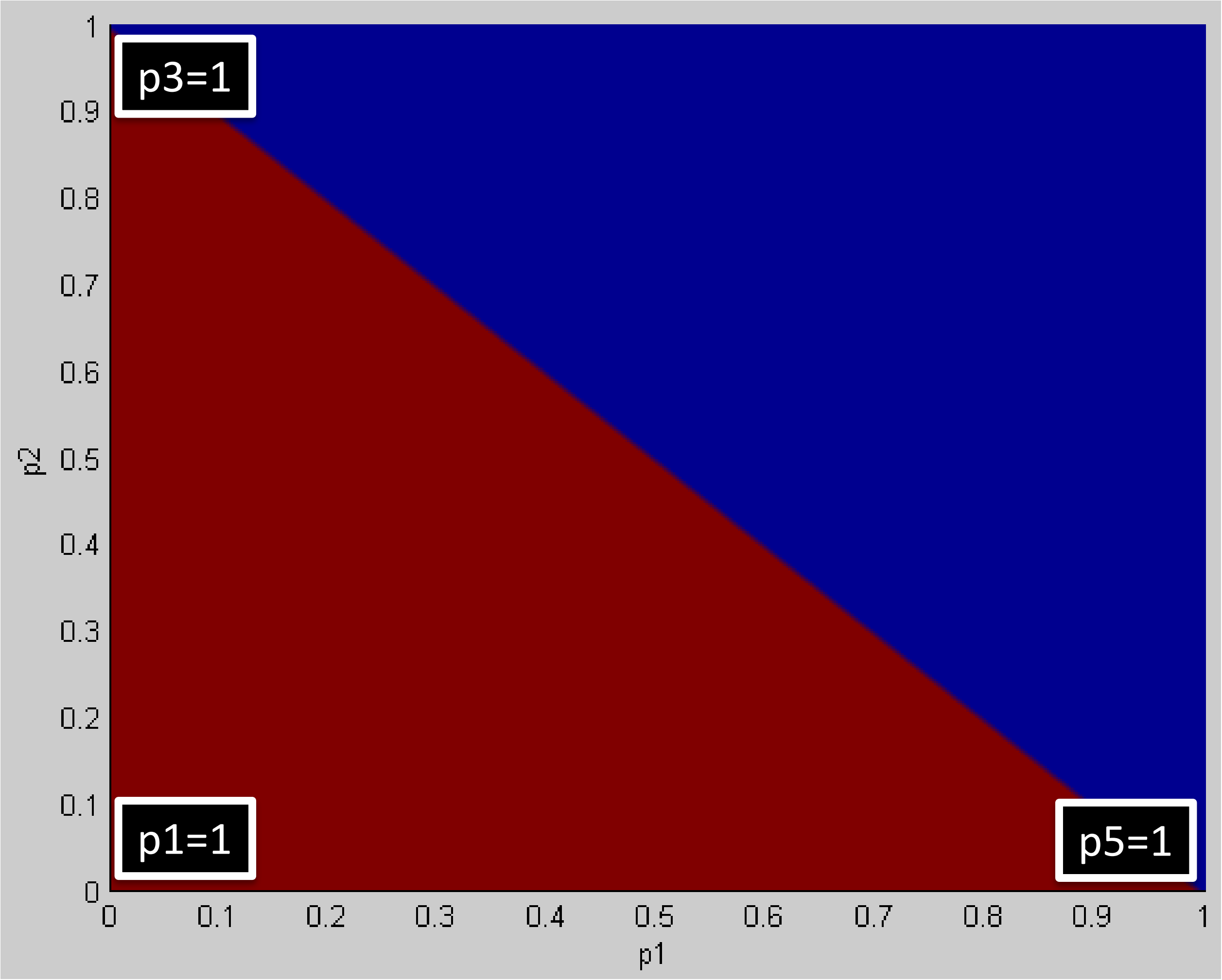
Dirichlet Distribution

$$\text{Prior} = P(p_1,p_3,p_5) = \text{const.} * (p_1)^{100} * (p_3)^{100} * (p_5)^{100}$$



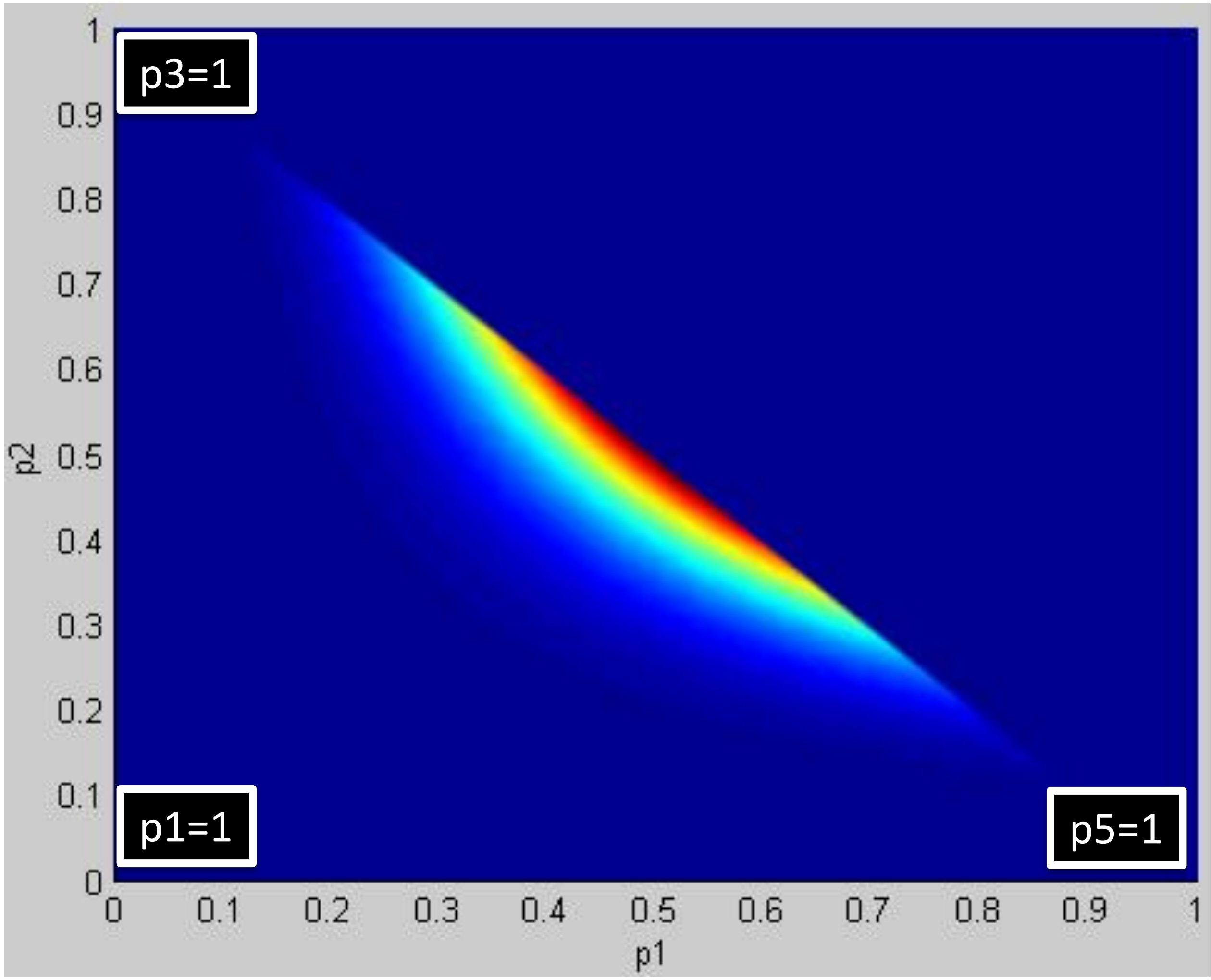
Dirichlet Distribution

Prior = $P(p_1,p_3,p_5) = const. * (p_1)^0 * (p_3)^0 * (p_5)^0$



Dirichlet Distribution

Likelihood = $P(p_1,p_3,p_5) = (p_1)^{\mathbf{0}} * (p_3)^{\mathbf{5}} * (p_5)^{\mathbf{5}}$

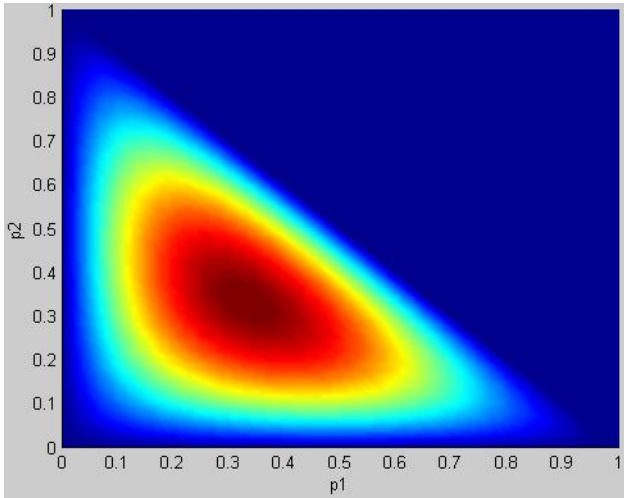


Dirichlet Distribution

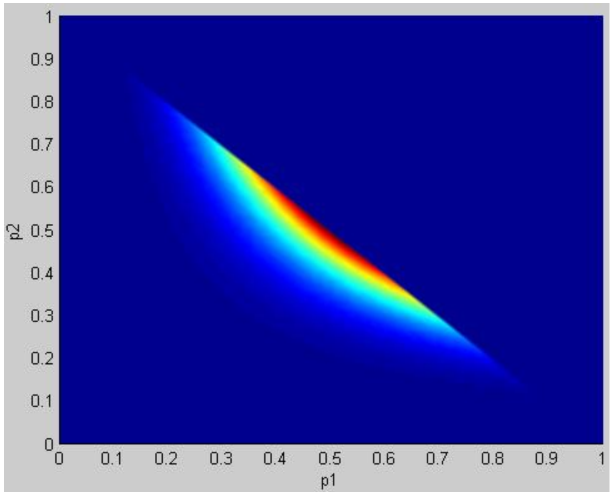
$$P(p_1,p_3,p_5|Data) = C*Prior*Likelihood = const.*\ (p_1)^{\mathbf{1}} * (p_3)^{1+\mathbf{5}} * (p_5)^{1+\mathbf{5}}$$

Observation: 5  5 

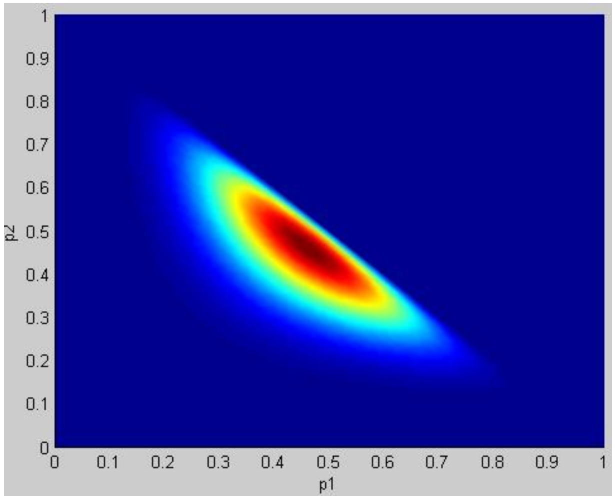
Prior



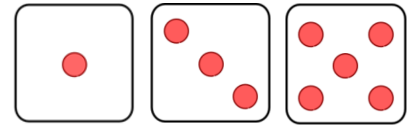
Likelihood



Posterior



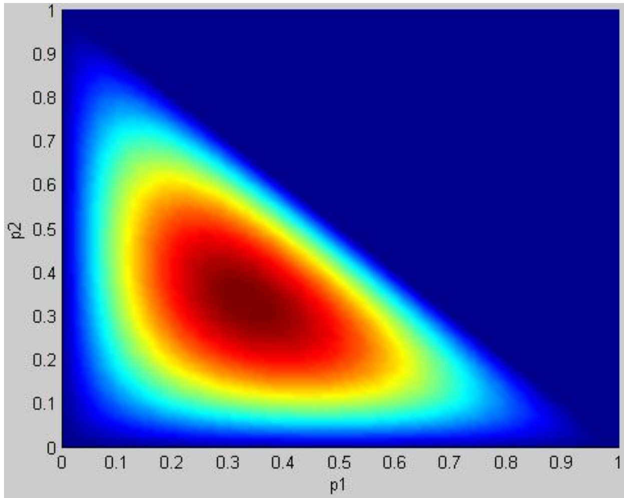
Dirichlet Distribution



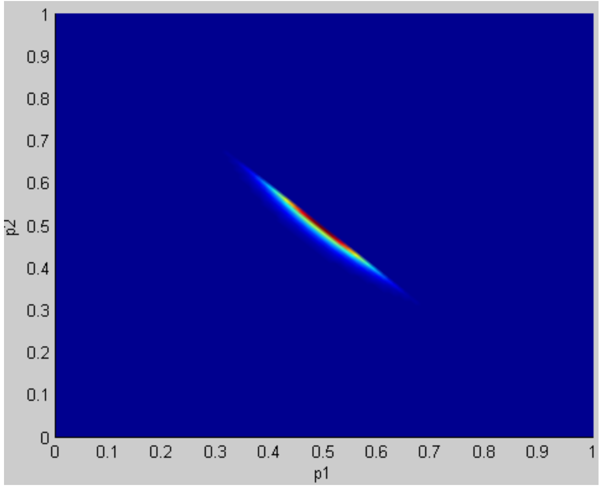
$$P(p_1,p_3,p_5|\text{Data}) = C*\text{Prior}*\text{Likelihood} = \text{const.}*\ (p_1)^{\mathbf{1}} * (p_3)^{1+\mathbf{30}} *(p_5)^{1+\mathbf{30}}$$

Observation: 30  30 

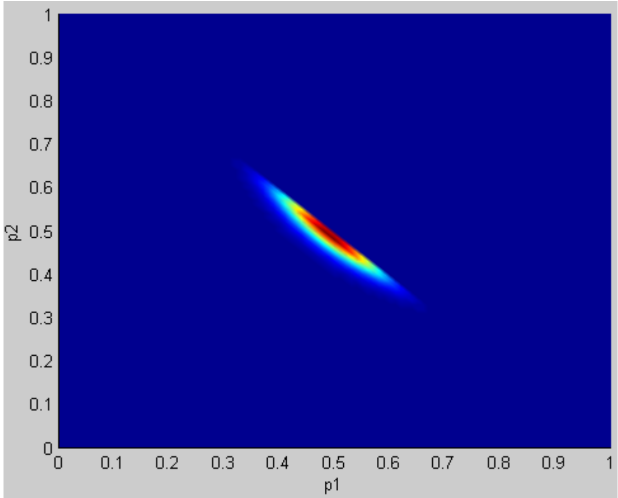
Prior



Likelihood



Posterior

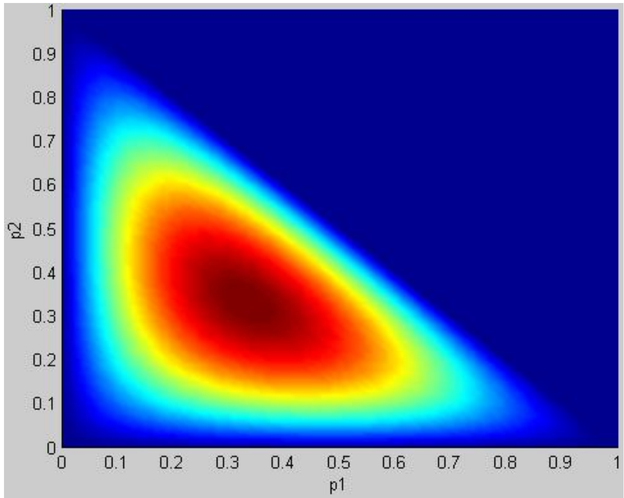


Dirichlet Distribution

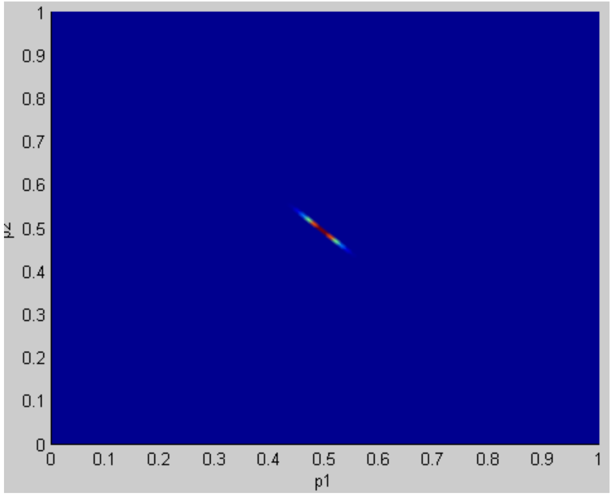
$$P(p_1,p_3,p_5 \mid Data) = C*Prior*Likelihood = const.* (p_1)^{\mathbf{1}} * (p_3)^{1+\mathbf{300}} *(p_5)^{1+\mathbf{300}}$$

Observation: 300  300 

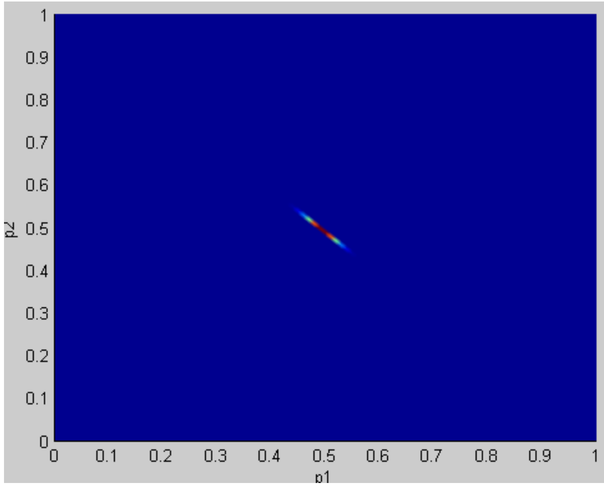
Prior



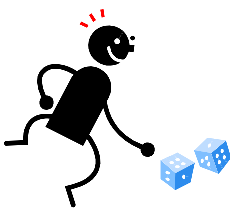
Likelihood



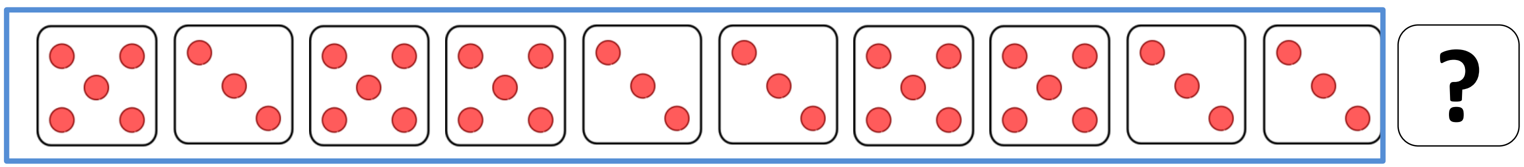
Posterior



Bayesian Learning



Training Data

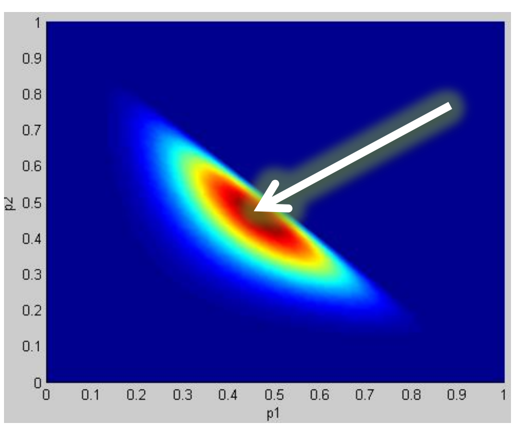


X ~ $\text{Mul}(p_1 \sim p_6)$

How to Predict with Posterior ?

$P(p_1 \sim p_6 | \text{Data})$ = const. * Piror * Likelihood

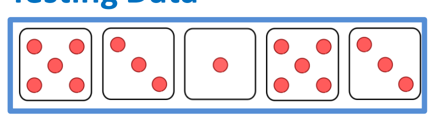
$= \text{const.} * (p_1)^1 * (p_2)^1 * (p_3)^{1+5} * (p_4)^1 * (p_5)^{1+5} * (p_6)^1$



1. Maximum Posterior (MAP) :

$$(p_1 \sim p_6) = \underset{p_1 \sim p_6}{\text{argmax}}\ P(p_1 \sim p_6 | \text{Data}) = \left(\frac{1}{16}, \frac{1}{16}, \frac{6}{16}, \frac{1}{16}, \frac{6}{16}, \frac{1}{16}\right)$$

Testing Data

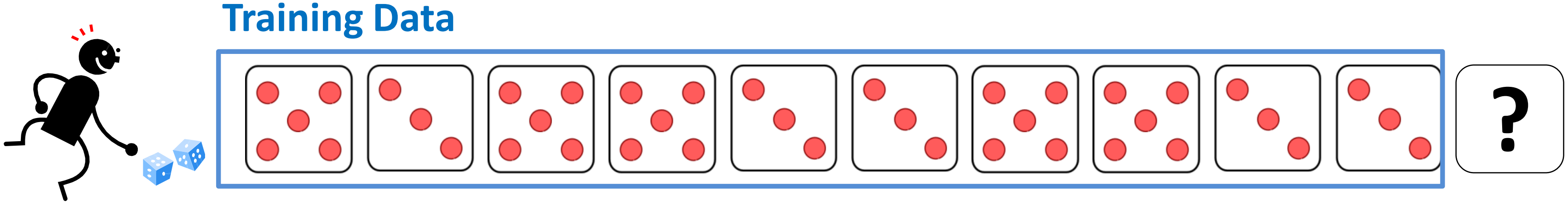


$$P(Data \mid p_1 \sim p_6) = \frac{6}{16} * \frac{6}{16} * \frac{1}{16} * \frac{6}{16} * \frac{6}{16} = 0.0012$$

$$\ln P(Data \mid p_1 \sim p_6) = -6.7$$ (much better)

Bayesian Learning

Training Data

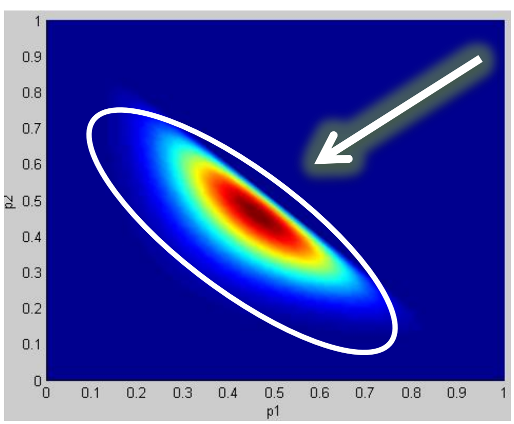


$\mathbf{X} \sim \mathrm{Mul}(p_1 \sim p_6)$

How to Predict with Posterior ?

$P(p_1 \sim p_6 | \text{Data}) = \text{const.} * \text{Piror} * \text{Likelihood}$

$= \text{const.} * (p_1)^1 * (p_2)^1 * (p_3)^{1+5} * (p_4)^1 * (p_5)^{1+5} * (p_6)^1$



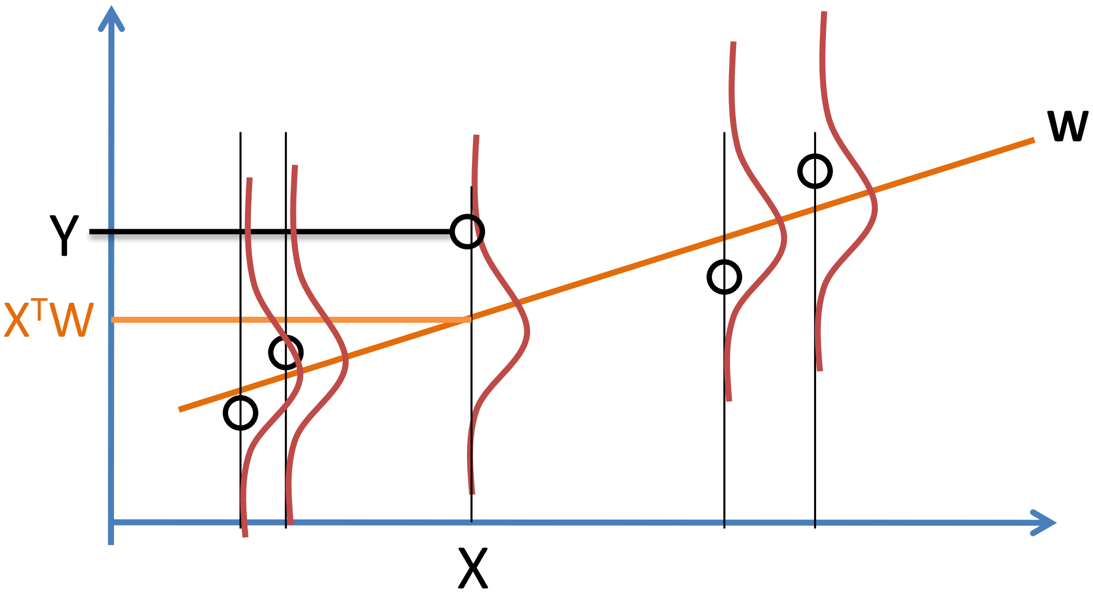
2. Evaluate Uncertainty :

$$Var_{P(p_1 \sim p_6|Data)}[\, p_1 \ldots p_6 \,] = (.003, .003, .014, .003, .014, .003)$$

$$Stderr_{P(p_1 \sim p_6|Data)}[\, p_1 \ldots p_6 \,] = (0.05, 0.05, 0.1, 0.05, 0.1, 0.05)$$

Bayesian Learning on Regression

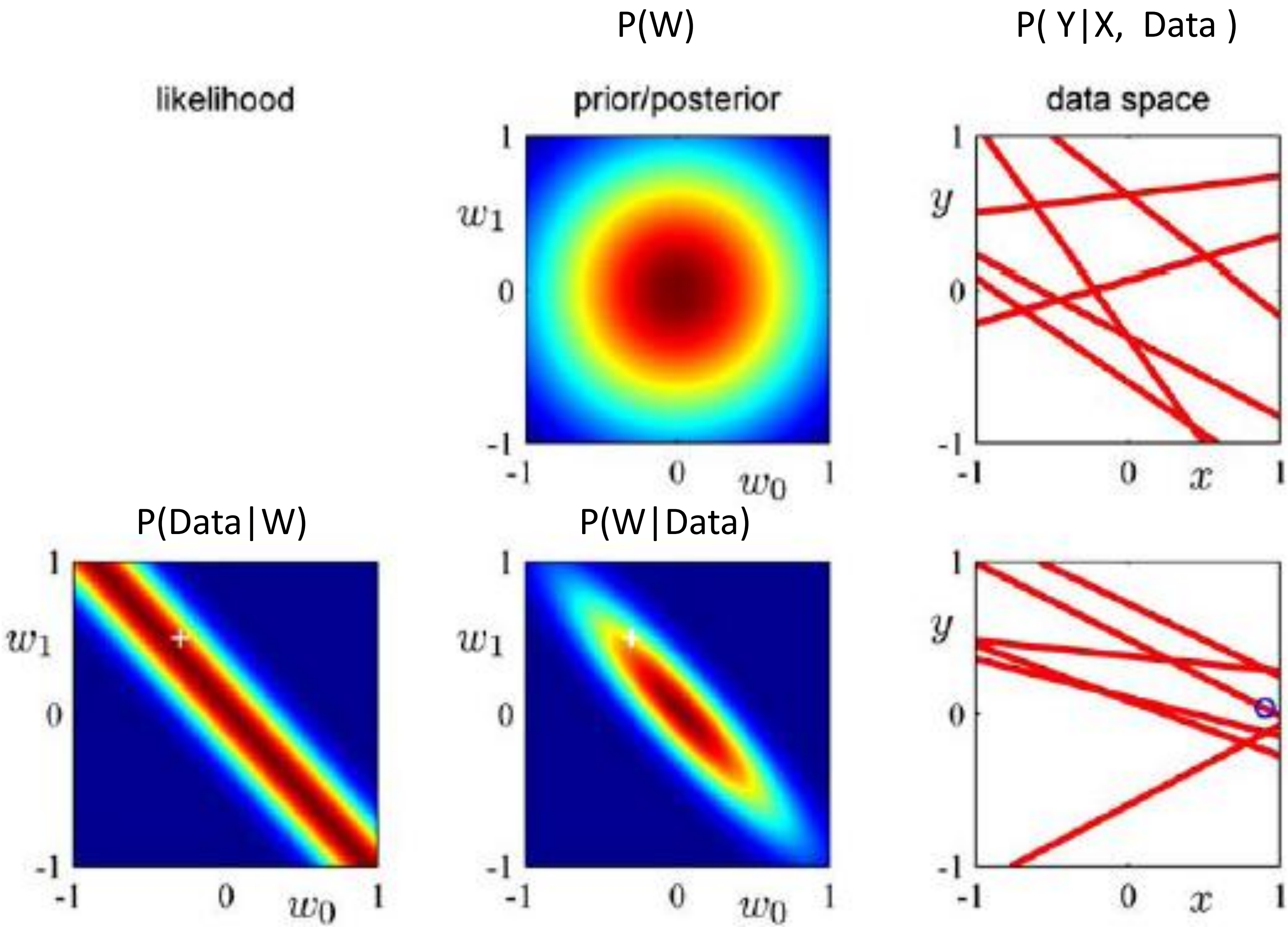
P(Y|X, W) = N(Y ; $X^T W$, σ^2)



$$Likelihood(W) = P(Data \mid W) = \prod_{n=1}^{N} P(Y_n \mid X_n, W)$$

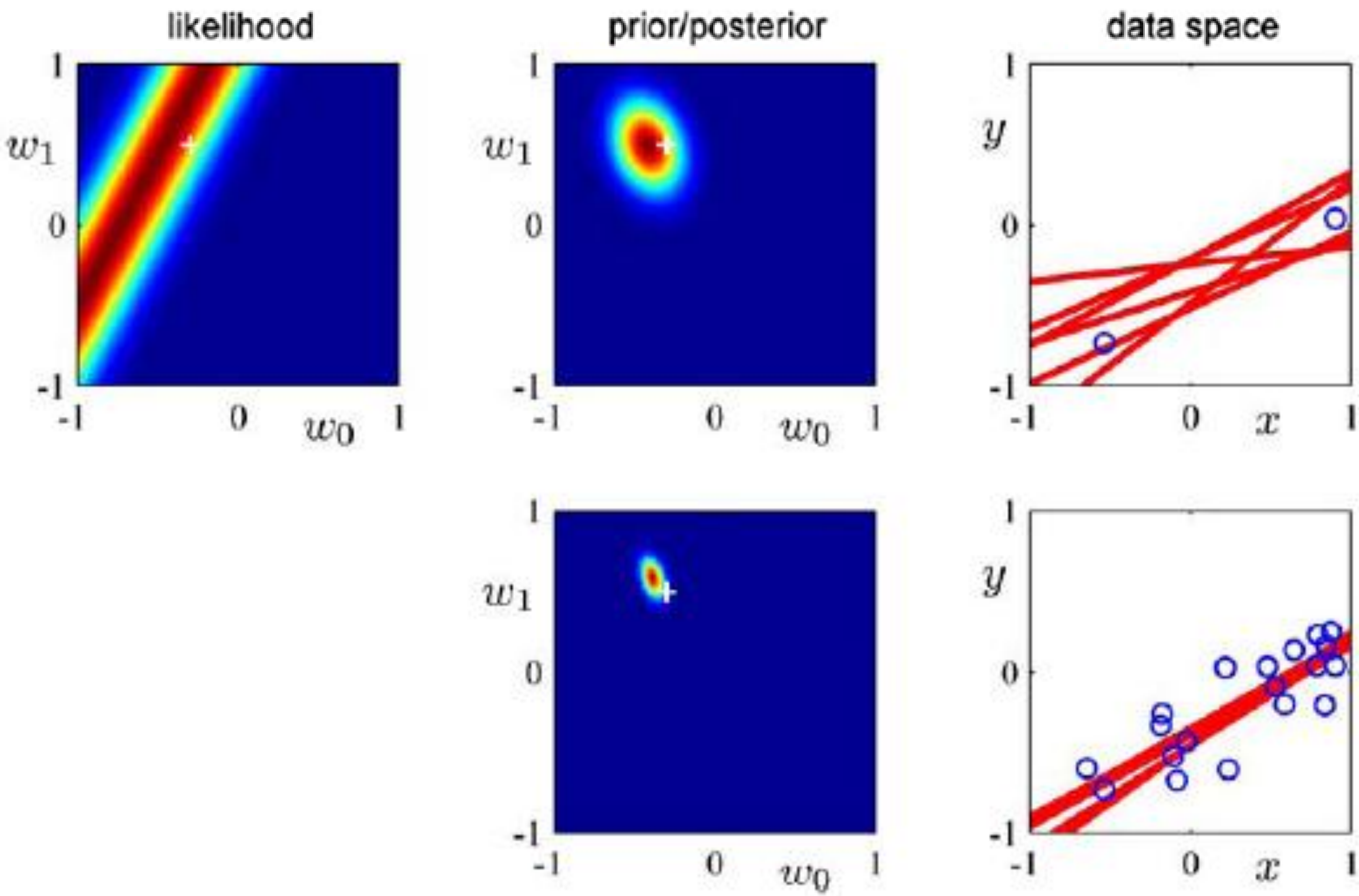
$$P(W \mid Data) = const. * Prior(W) * Likelihood(W)$$

Bayesian Learning on Regression



Bayesian Learning on Regression

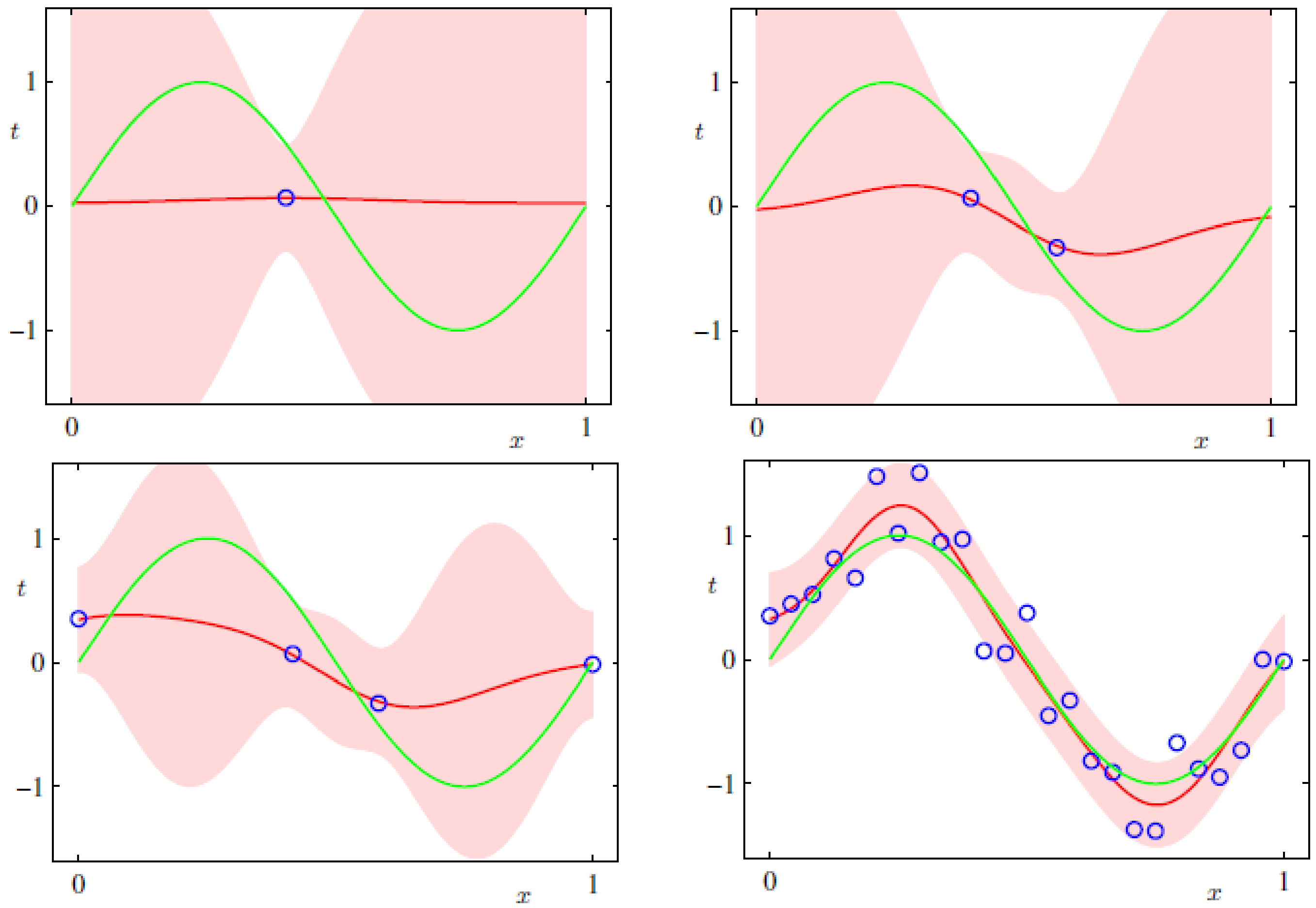
P(Data|W) P(W|Data) P(Y|X, Data)



Bayesian Inference on Regression

Predictive Distribution : P(Y|X , Data) = $\int$ P(Y|X, W) * P(W|Data) **dW**

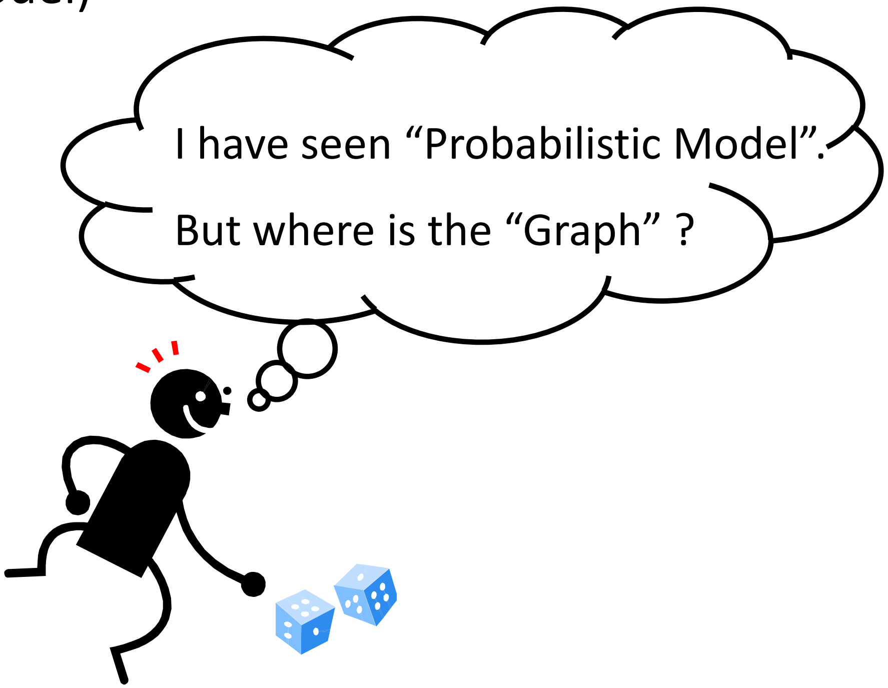
Modeling Uncertainty :



Overview

- What's Probabilistic Graphical Model for ?

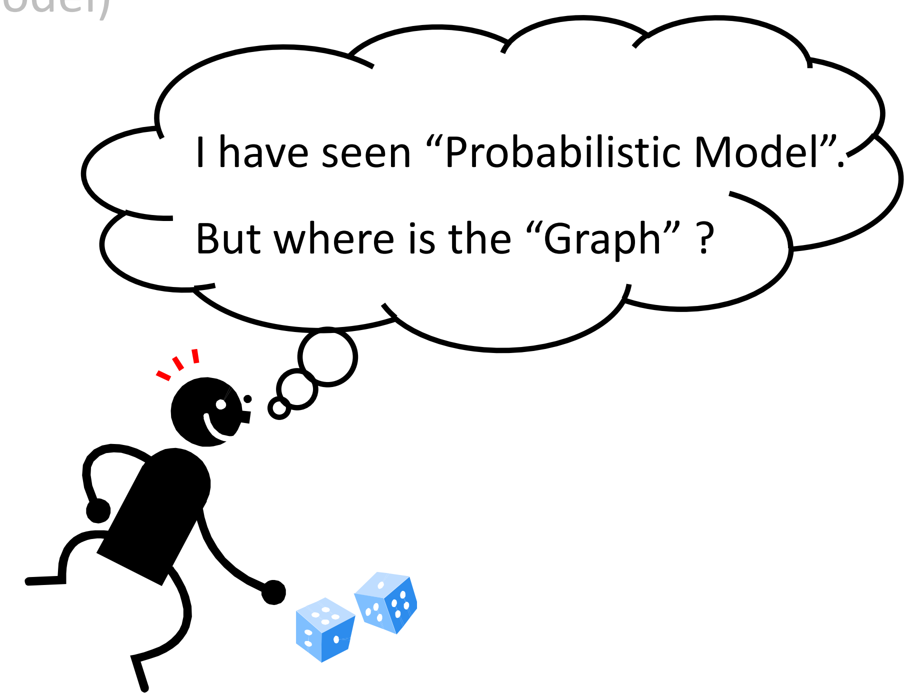
- Tasks in Graphical Model:
  - Modeling (Simple Probability Model)
  - Learning (MLE, MAP, Bayesian)
  - Inference (?)



- Examples
  - Topic Model
  - Hidden Markov Model
  - Markov Random Field

Overview

- What’s Probabilistic Graphical Model for ?
- Tasks in Graphical Model:
  - Modeling (Simple Probability Model)
  - Learning (MLE, MAP, Bayesian)
  - Inference (?)
- Examples
  - Topic Model
  - Hidden Markov Model
  - Markov Random Field



How to Model a Document of Text ?

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases.

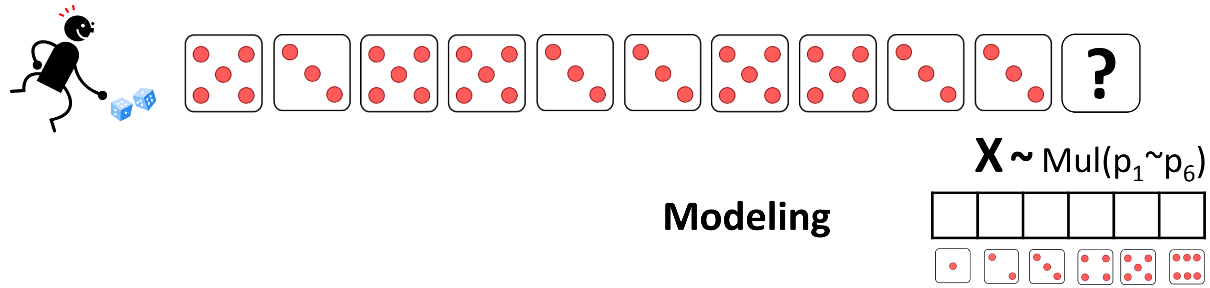
What you want to do ?

- Build Search Engine ?
- Natural Language Understanding ?
- Machine Translation ?

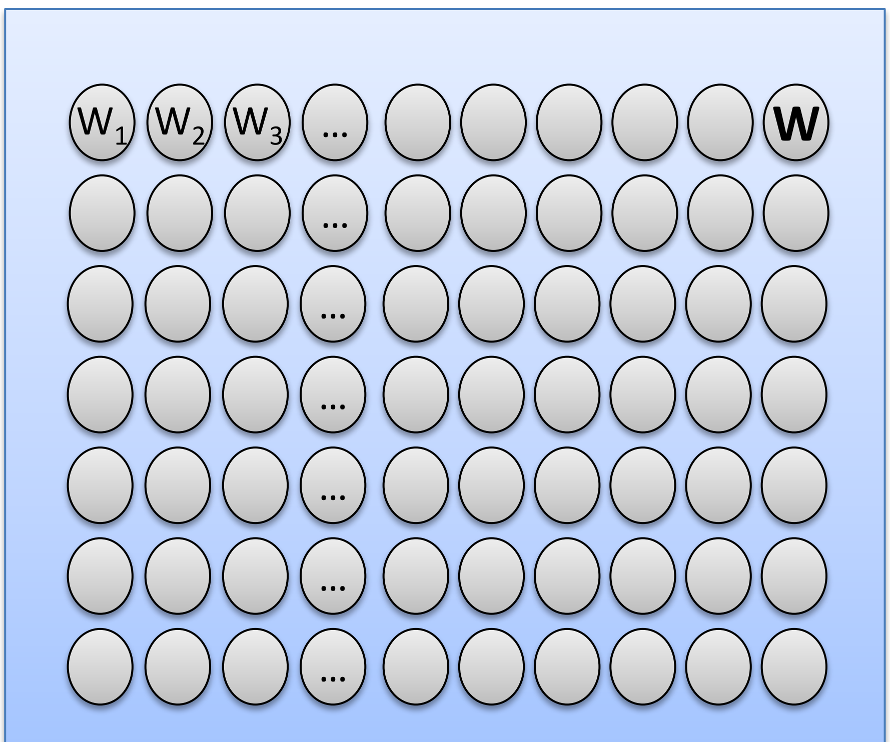
Bag of Words Model:

Not consider words position, just like a bag of words.

Model a Document of Text



Document



$$\mathbf{W} \sim \mathrm{Mul}(\, p_1 \sim p_{|\mathrm{Vocabulary}|} \,)$$

Model a Document of Text

Doc1	**Doc2**
dog,	learning,
puppy,	intelligence,
breed,	algorithm,

MLE from data ➔ **$p_W = 1/6$ for all w.**

What's the problem ?

➔ Likelihood = P(Data) = P(Doc1,Doc2) = $(1/6)^3 * (1/6)^3 = 2*10^{-5}$

Each Doc. has its own distribution of words ?!

Each Doc. has its own distribution of words ??

Abstract

We describe *latent Dirichlet allocation* (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. We present efficient approximate inference techniques based on variational methods and an EM algorithm for empirical Bayes … , text classification, and collaborativ… e probabilistic LSI model.

Documents on "the same Topic" has similar distribution of words.

ABSTRACT

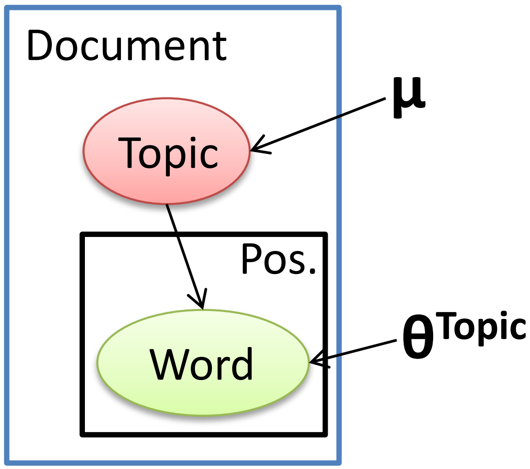
One essential issue of document clustering is to estimate the appropriate number of clusters for a document collection to which documents should be partitioned. In this paper, we propose a novel approach, namely DPMFS, to address this issue. The proposed approach is designed 1) to group documents into a set of clusters while the number of document clusters is determined by the Dirichlet process mixture model automatically; 2) to identify the discriminative words and separate them from irrelevant noise words via stochastic search variable selection technique. We explore the performance of our proposed approach on both a synthetic dataset and several realistic document datasets. The comparison between our proposed approach and stage-of-the-art document clustering approaches indicates that our approach is robust and effective for document clustering.

Abstract

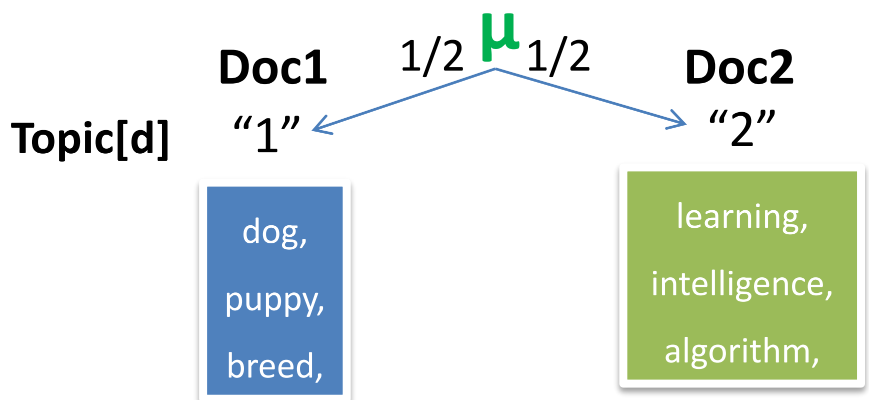
…s (DP) mixture models provide a flexible Bayesian framework for density estimation. Unfortunately, their flexibility comes at a cost: inference in DP mixture models is computationally expensive, even when conjugate distributions are used. In the common case when one seeks only a maximum a posteriori assignment of data points to clusters, we show that search algorithms provide a practical alternative to expensive MCMC and variational techniques. When a true posterior sample is desired, the solution found by search can serve as a good initializer for MCMC. Experimental results show that using these techniques is it possible to apply DP mixture models to very large data sets.

A Topic Model ---"Word" depends on "Topic"

Template Representation



Ground Representation



	dog	puppy	breed	learning	intelligence	algorithm
$\theta^{Topic\ 1}$	θ^1_{dog}	θ^1_{puppy}	θ^1_{breed}	$\theta^1_{learning}$	$\theta^1_{intelligence}$	$\theta^1_{algorithm}$
	1/3	1/3	1/3	0.0	0.0	0.0
$\theta^{Topic\ 2}$	θ^2_{dog}	θ^2_{puppy}	θ^2_{breed}	$\theta^2_{learning}$	$\theta^2_{intelligence}$	$\theta^2_{algorithm}$
	0.0	0.0	0.0	1/3	1/3	1/3

For all Documents **d**

1. draw Topic[**d**] ~ Multi (**μ**)

For all Position **w**

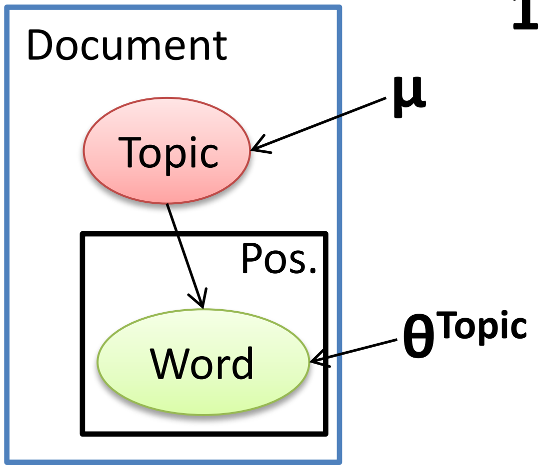
2. draw W[**d,w**]~ Multi ($\theta^{Topic[d]}$)

Likelihood = P(Data) = P(Doc1)*P(Doc2)

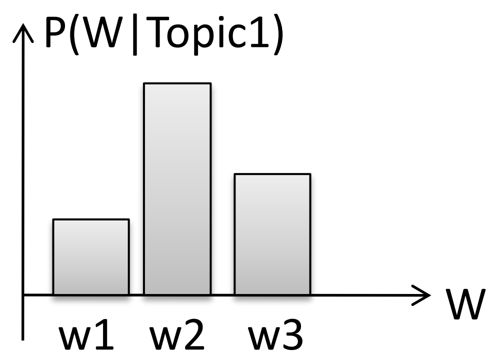
$= \{ P(T_1)P(W_1 \sim W_3 | T_1) \} * \{ P(T_2)P(W_1 \sim W_3 | T_1) \}$

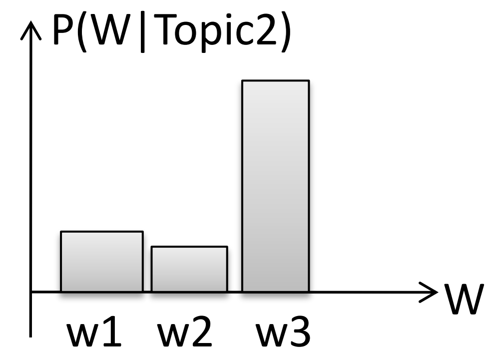
$= (1/2)\ (1/3)^3\ (1/2)\ (1/3)^3 = 3*10^{-3}$

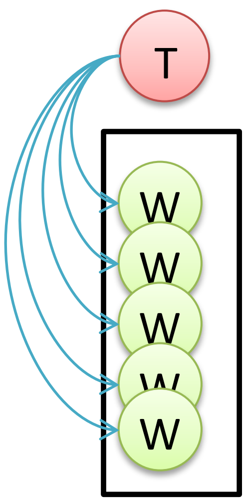
Learning with Hidden Variables

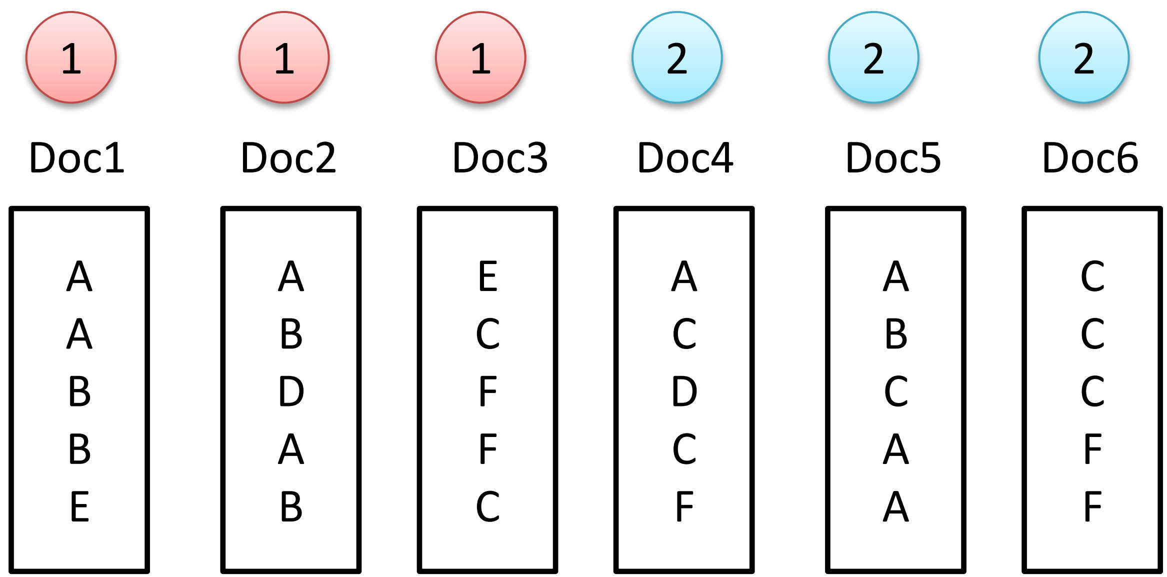


1. Given Topics, we can learn word's distribution.

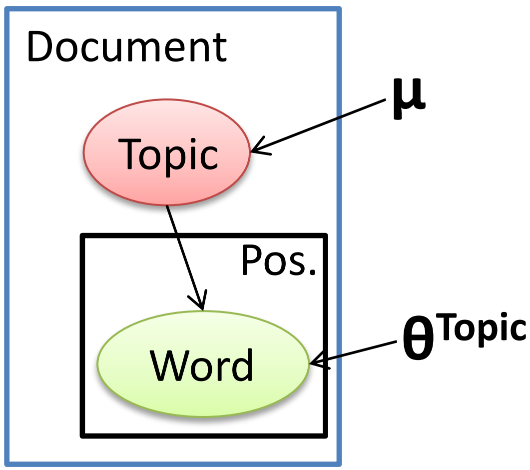






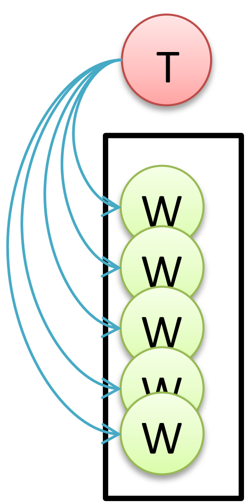


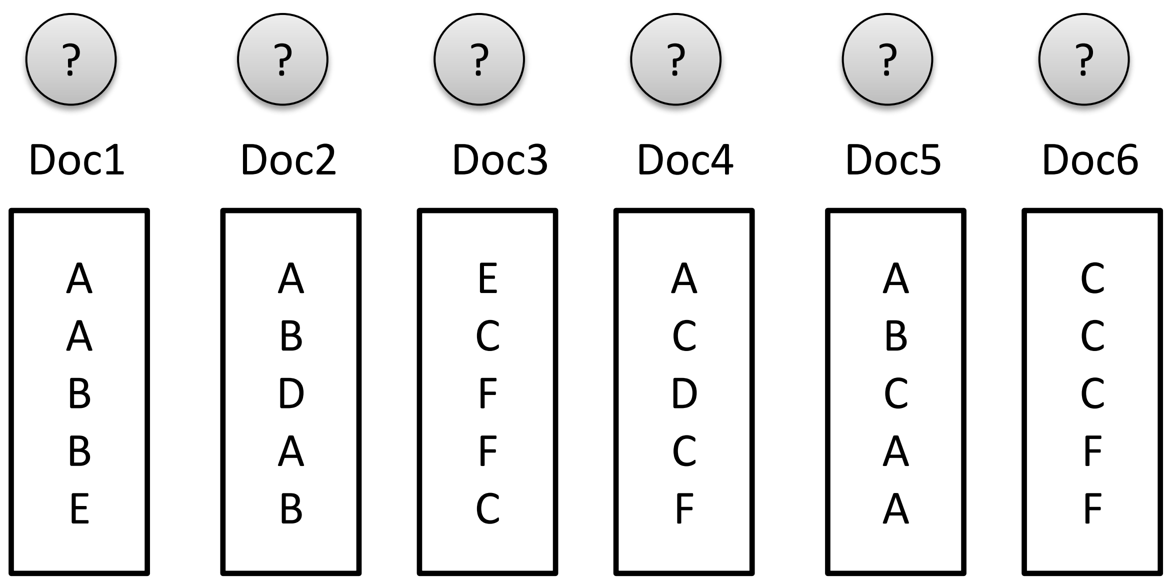
Learning with Hidden Variables



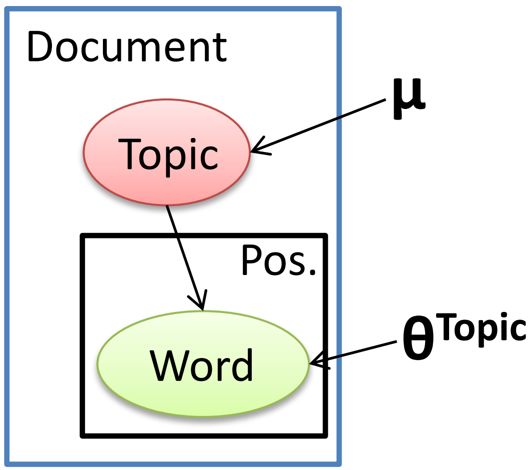
2. Given words distribution P(W|Topic), we can infer Topics.

$$P(T|W_1 \sim W_5)$$
$$= \text{const.} * P(T) * P(W_1|T)\, P(W_2|T) \ldots P(W_5|T)$$



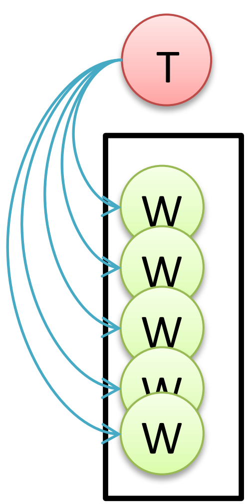


Learning with Hidden Variables



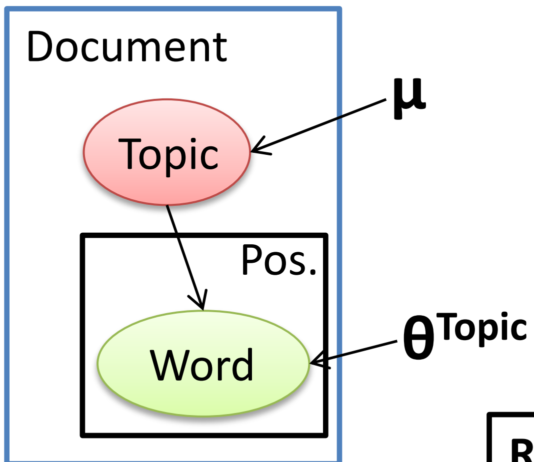
Both of them are unknown, how to Learn?

➔Using EM algorithm.
(here is simplified version.)



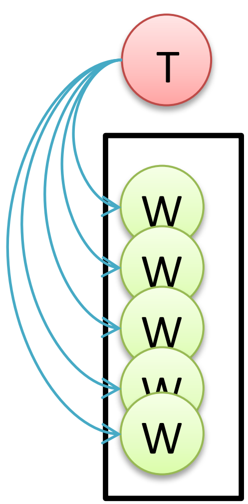
Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
A	A	E	A	A	C
A	B	C	C	B	C
B	D	F	D	C	C
B	A	F	C	A	F
E	B	C	F	A	F

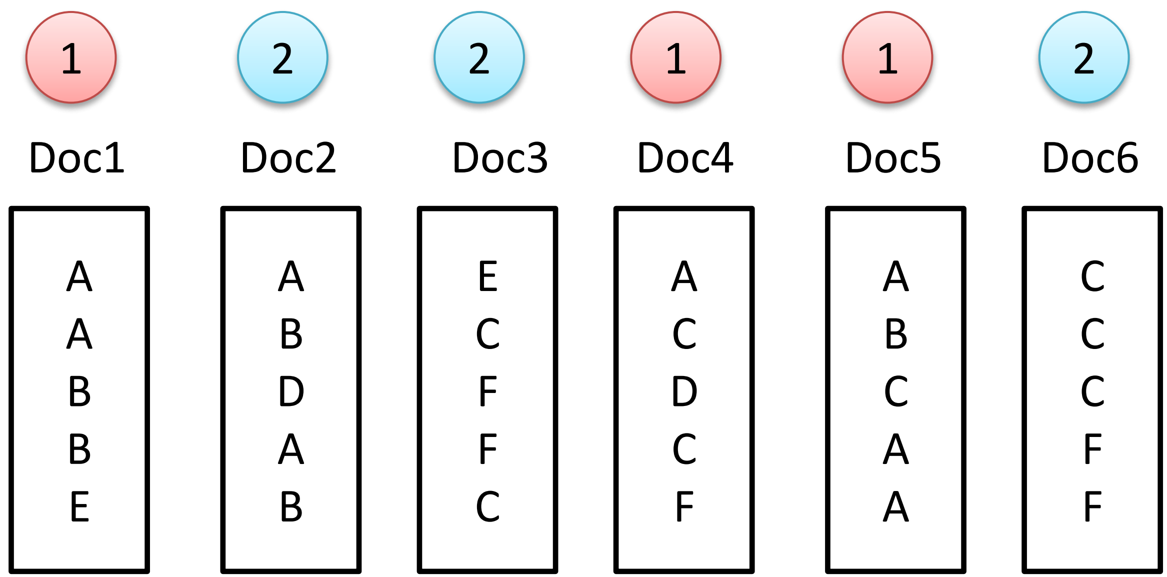
Learning with Hidden Variables



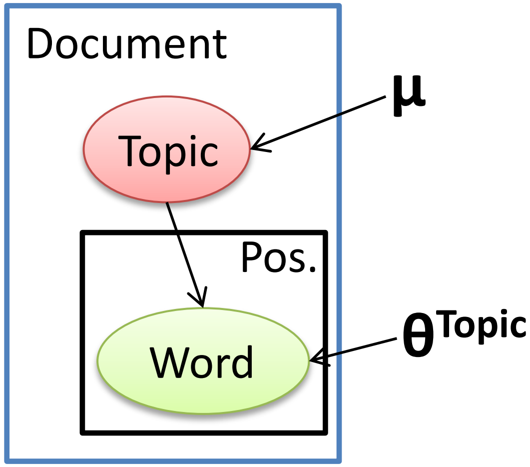
Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

Random Initalize :





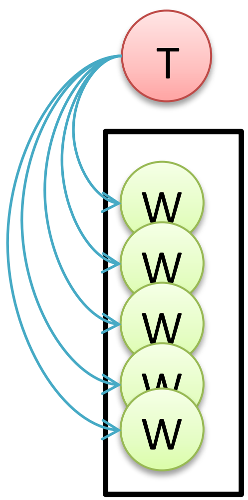
Learning with Hidden Variables

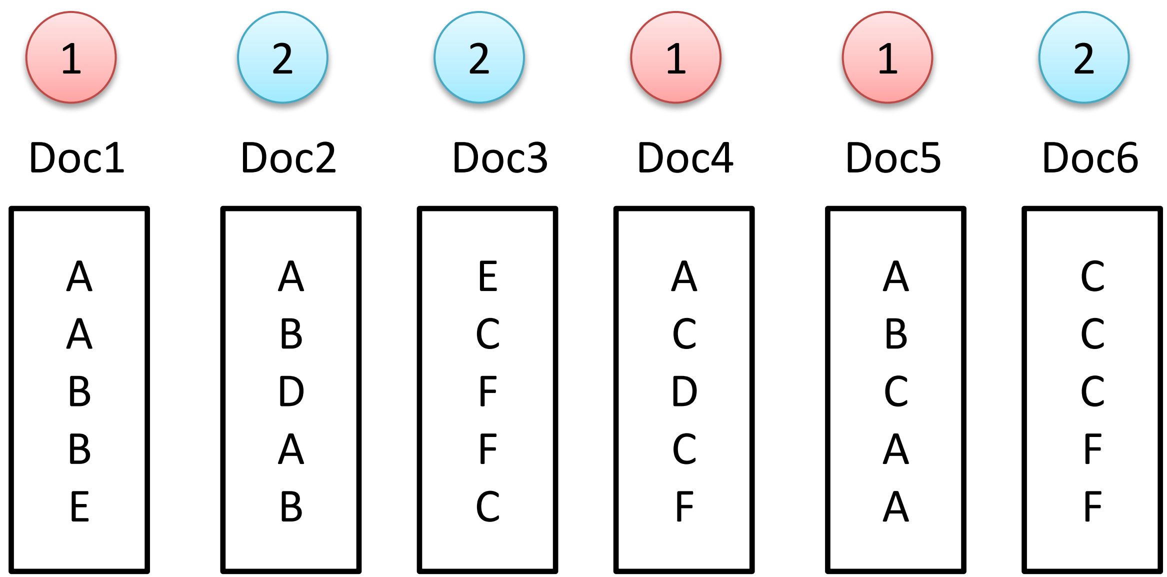


Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

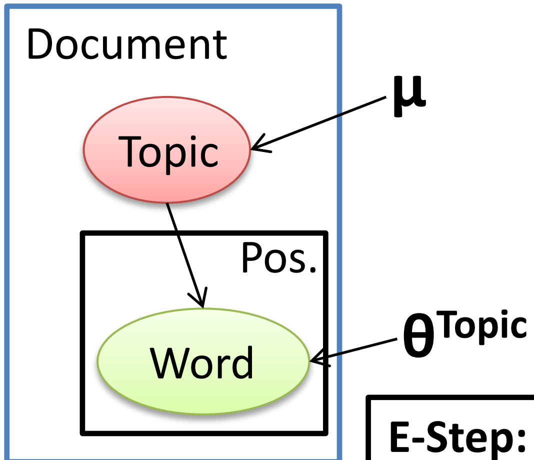
M-Step:

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15





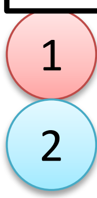
Learning with Hidden Variables



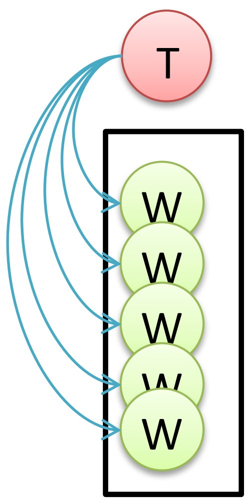
Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15

E-Step:

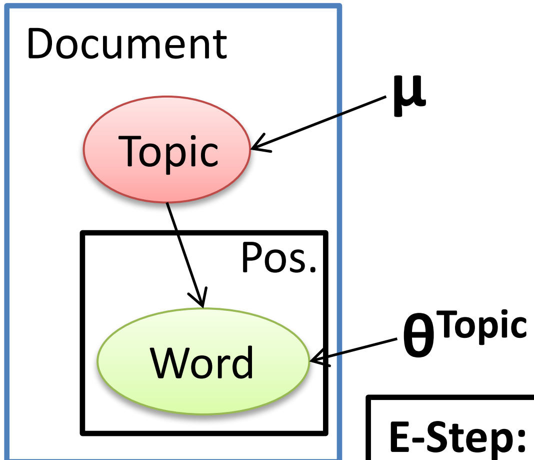


$$P(T|W_1\sim W_5) = const.* P(W_1\sim W_5|T) * P(T)$$ **1/2, 1/2**



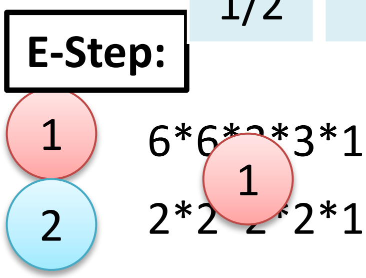
Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
A	A	E	A	A	C
A	B	C	C	B	C
B	D	F	D	C	C
B	A	F	C	A	F
E	B	C	F	A	F

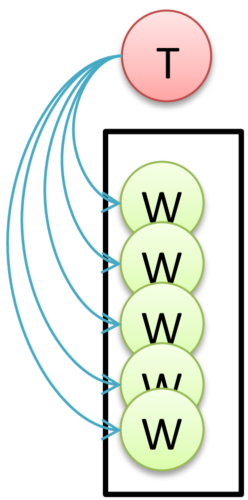
Learning with Hidden Variables

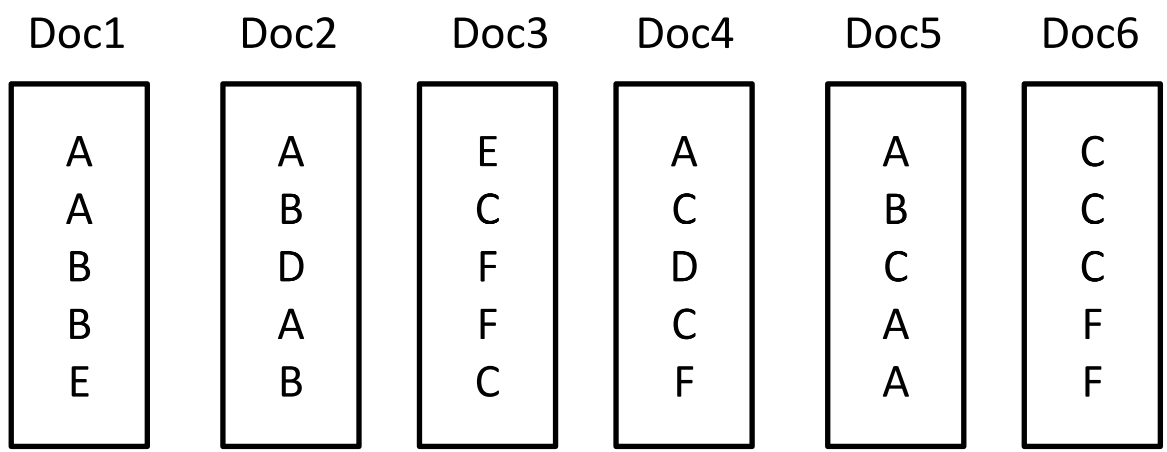


Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

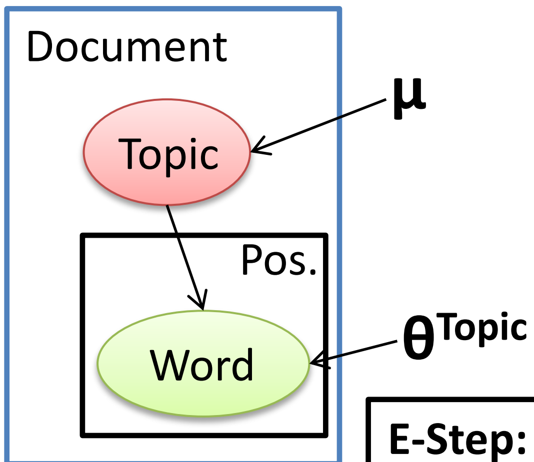
P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15







Learning with Hidden Variables



Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15

E-Step:

1

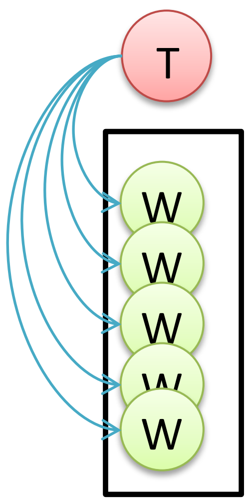
2

1

6*3*1*6*3

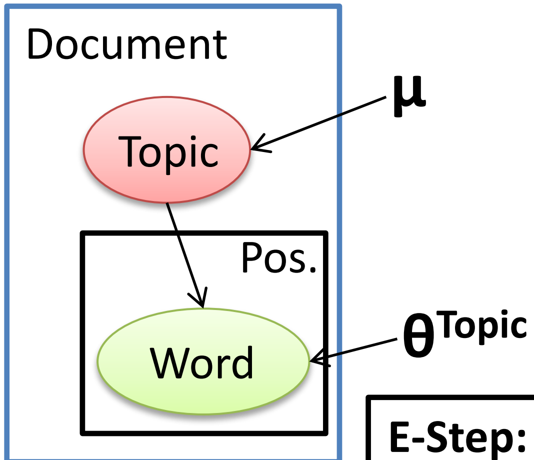
1

2*2*1*2*2



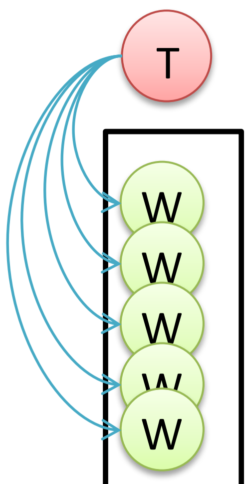
Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
A	A	E	A	A	C
A	B	C	C	B	C
B	D	F	D	C	C
B	A	F	C	A	F
E	B	C	F	A	F

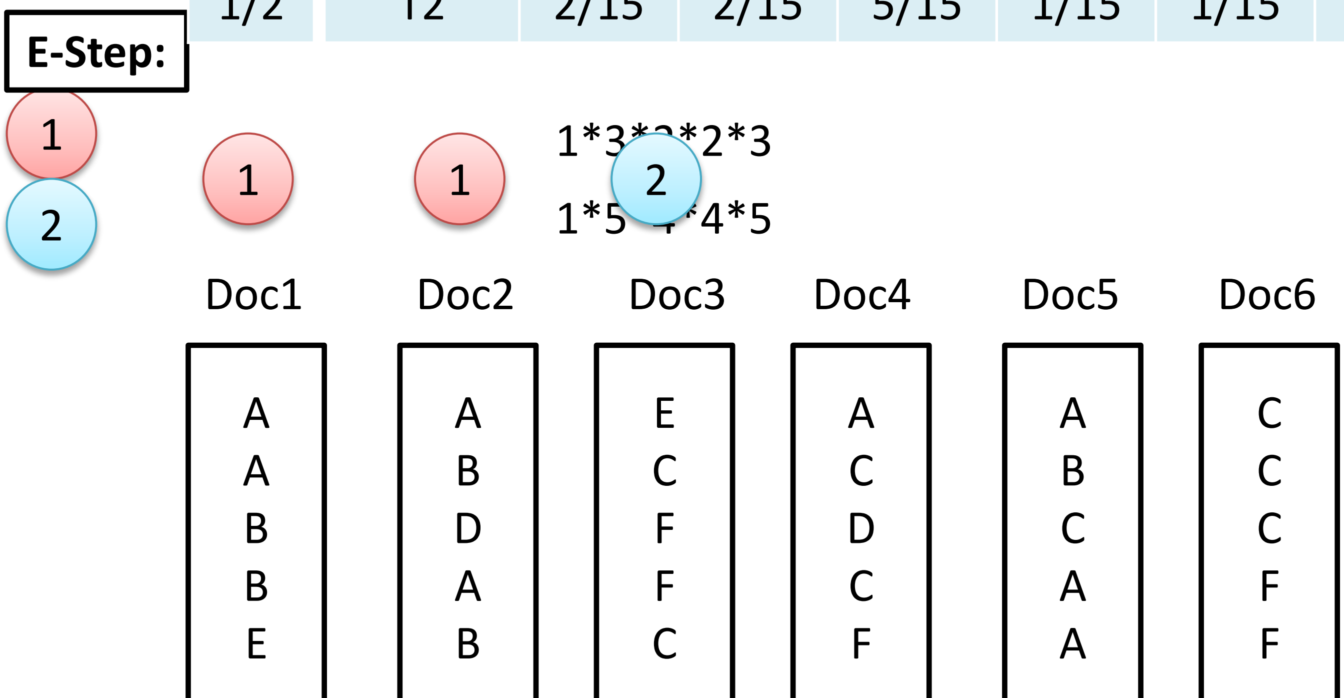
Learning with Hidden Variables



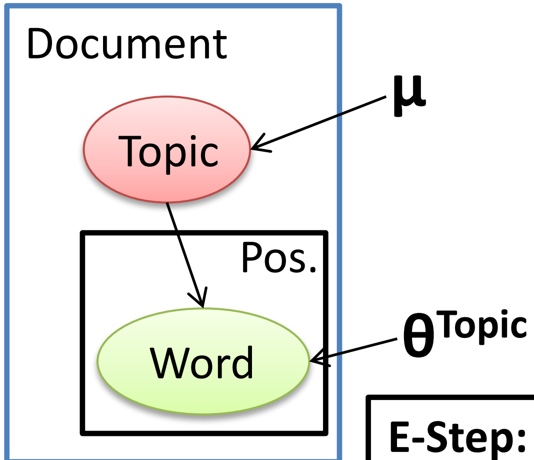
Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15





Learning with Hidden Variables



Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

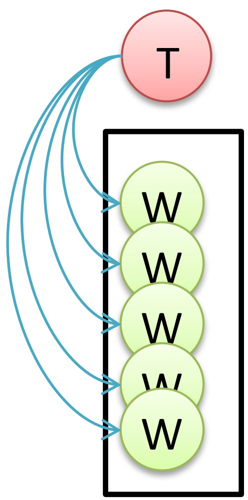
P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15

E-Step:

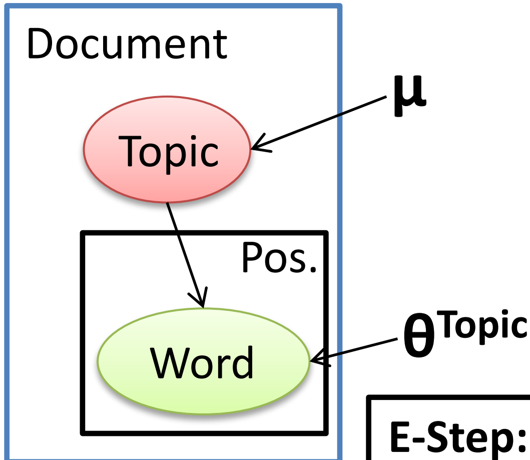
1
2

6*3*1*3*2
2*5*1*5*4

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
Topic	1	1	2	2		
	A	A	E	A	A	C
	A	B	C	C	B	C
	B	D	F	D	C	C
	B	A	F	C	A	F
	E	B	C	F	A	F



Learning with Hidden Variables



Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

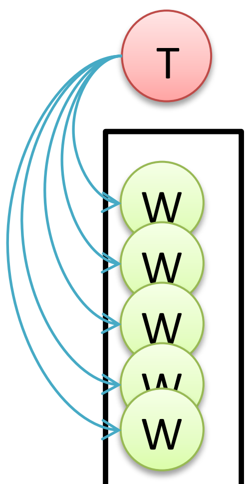
P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15

E-Step:

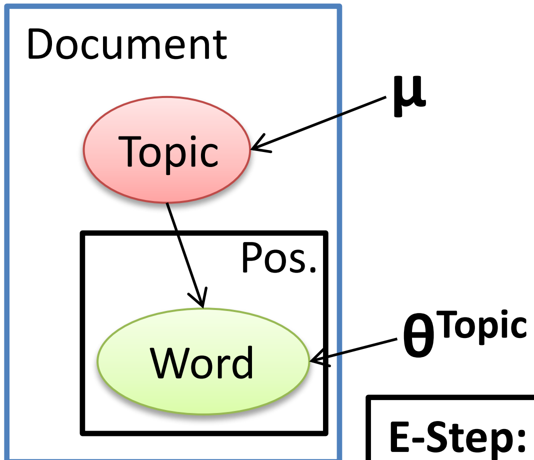
1
2

6*3*3*6*6
2*2*5*2*2

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
Topic	1	1	2	2	1	
	A	A	E	A	A	C
	A	B	C	C	B	C
	B	D	F	D	C	C
	B	A	F	C	A	F
	E	B	C	F	A	F



Learning with Hidden Variables



Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

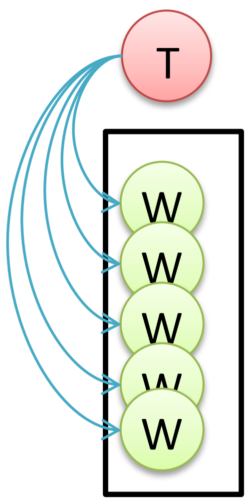
P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
1/2	T2	2/15	2/15	5/15	1/15	1/15	4/15

E-Step:

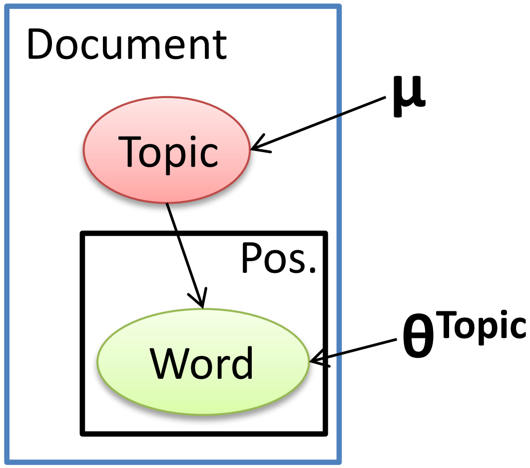
1
2

3*3*…*2*2
5*5…*4*4

1	1	2	2	1	2
Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
A	A	E	A	A	C
A	B	C	C	B	C
B	D	F	D	C	C
B	A	F	C	A	F
E	B	C	F	A	F



Learning with Hidden Variables

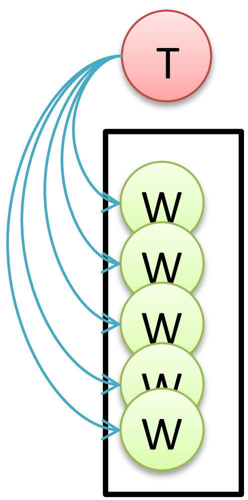


Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	6/15	3/15	3/15	1/15	1/15	2/15
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4/15

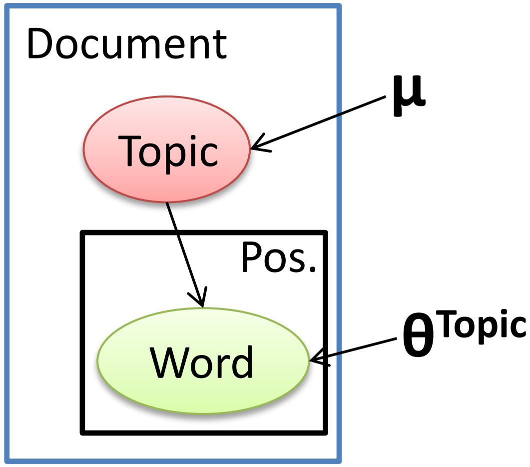
Not Converge:

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
	1	2	2	1	1	2
	1	1	2	2	1	2



Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
A	A	E	A	A	C
A	B	C	C	B	C
B	D	F	D	C	C
B	A	F	C	A	F
E	B	C	F	A	F

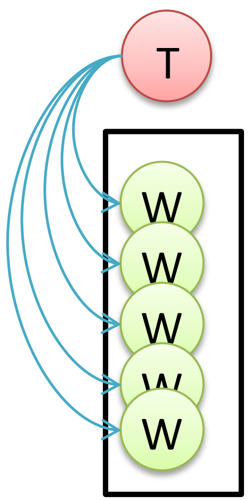
Learning with Hidden Variables

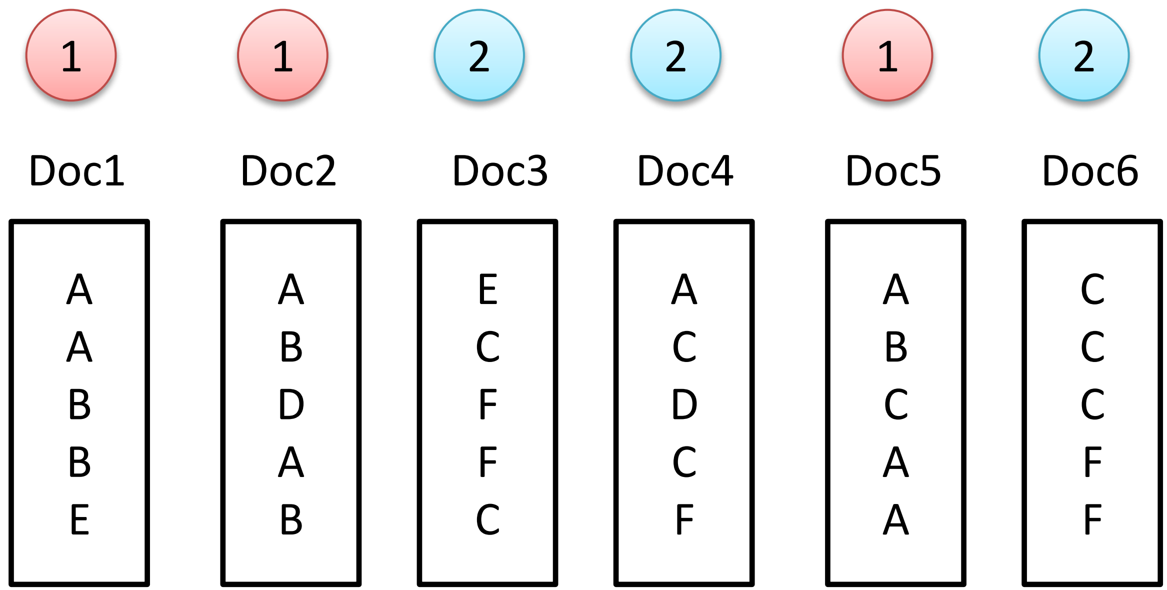


Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

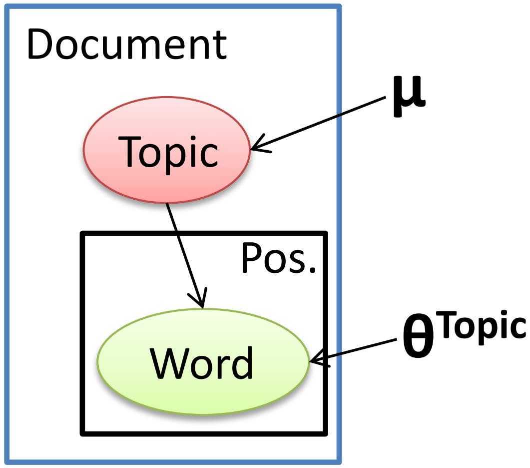
M-Step:

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	7/15	5/15	1/15	1/15	1/15	0/15
1/2	T2	1/15	0/15	7/15	1/15	1/15	5/15





Learning with Hidden Variables

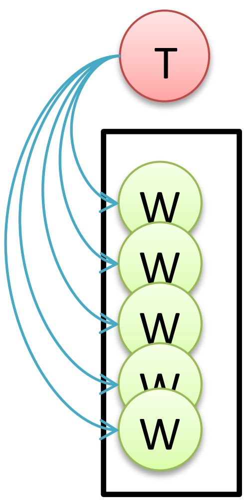


Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

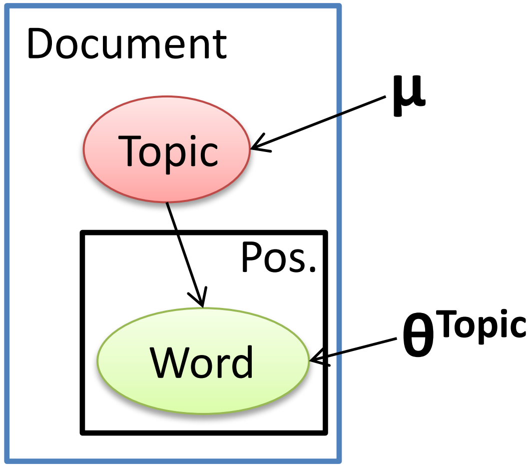
P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	7/15	5/15	1/15	1/15	1/15	0/15
1/2	T2	1/15	0/15	7/15	1/15	1/15	5/15

E-Step:

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
Topic	1	1	2	2	1	2
	A	A	E	A	A	C
	A	B	C	C	B	C
	B	D	F	D	C	C
	B	A	F	C	A	F
	E	B	C	F	A	F



Learning with Hidden Variables

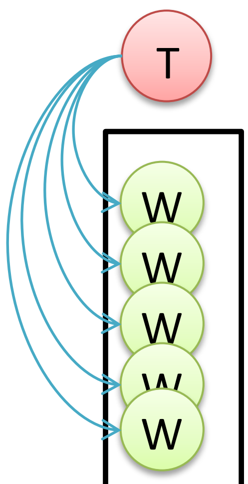


Topic is unknown (No one can tell us.) ,
how to learn word distribution ?

P(T)	P(W|T)	A	B	C	D	E	F
1/2	T1	7/15	5/15	1/15	1/15	1/15	0/15
							5/15

Converge

Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
1	1	2	2	1	2
1	1	2	2	1	2
A	A	E	A	A	C
A	B	C	C	B	C
B	D	F	D	C	C
B	A	F	C	A	F
E	B	C	F	A	F

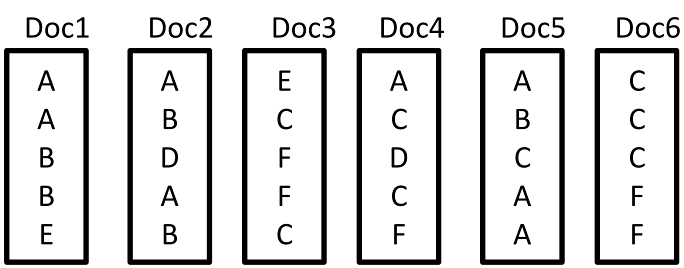


Learning with Hidden Variables

Learning & Inference Jointly.

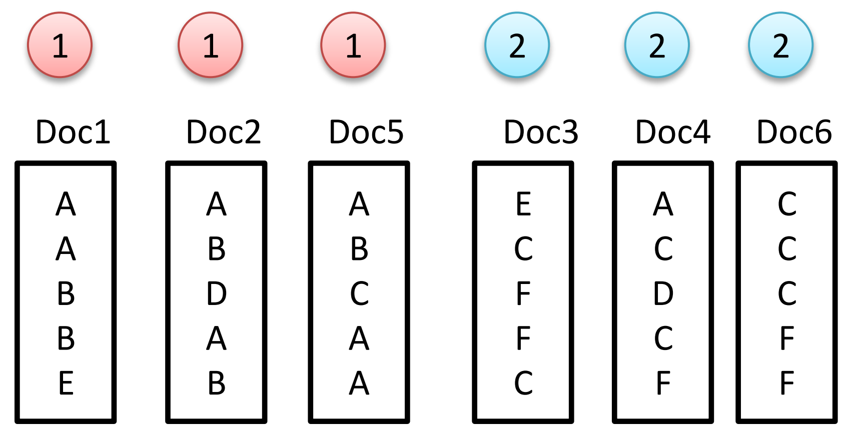
2 Tasks :

1. Infer Topic of documents.
2. Learn Topic's "word distribution" ?



P(T)
1/2
1/2

P(W|T)	A	B	C	D	E	F
T1	7/15	5/15	1/15	1/15	1/15	0/15
T2	1/15	0/15	7/15	1/15	1/15	5/15



Topic Model

Problem Solved ?

If.......

Topic1 Topic2

		word1	word2	word3	word4	word5	Word6
Topic1	Doc1	1	1	1	1	0	0
Topic2 ?	Doc2	0	0	0	0	1	1
Topic2	Doc3	0	0	1	1	1	1
?????????	Doc4	1	1	0	0	1	1
?????????	Doc5	0	0	1	1	0	0

Topic Model

Problem Solved ?

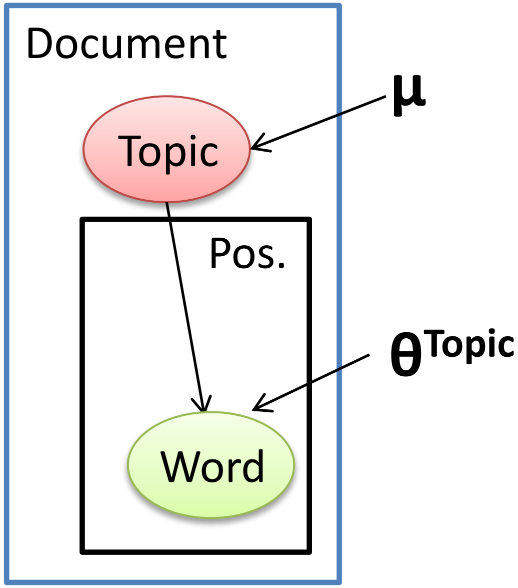
If…….

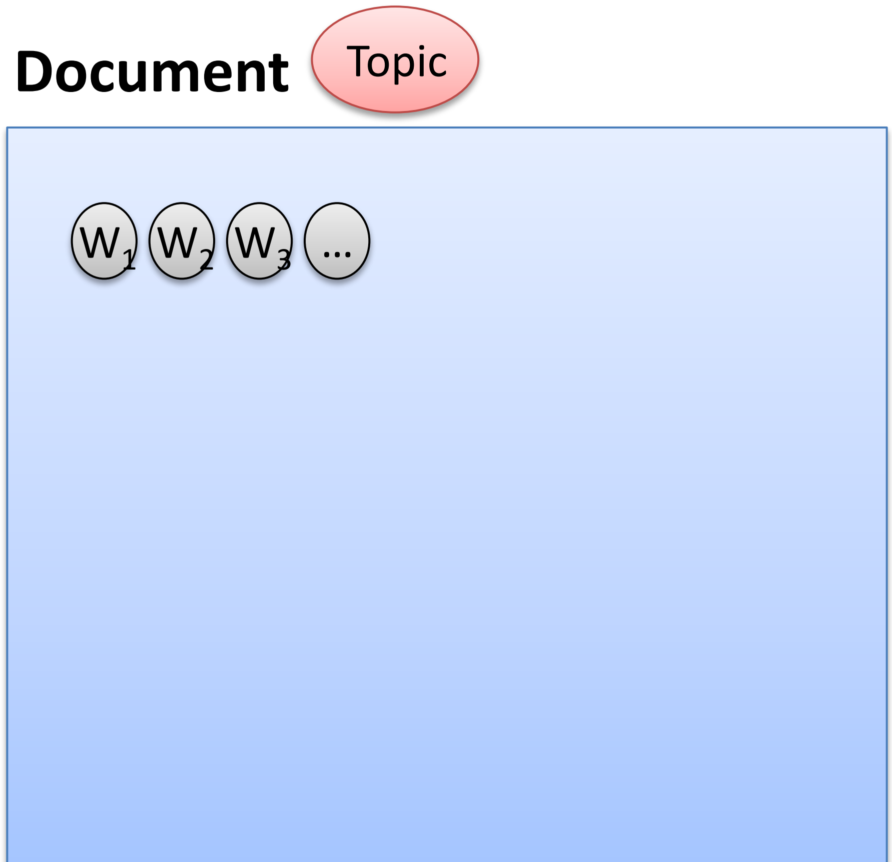
		Topic1		Topic2		Topic3	
		word1	**word2**	**word3**	**word4**	**word5**	**Word6**
Topic 1 or 2 ?	Doc1	1	1	1	1	0	0
Topic3	Doc2	0	0	0	0	1	1
Topic 2 or 3 ?	Doc3	0	0	1	1	1	1
Topic 1 or 3 ?	D[illegible]	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Topic2	Doc5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Obvious that
a doc. has **mix of Topics**
instead of **only 1 Topic**.

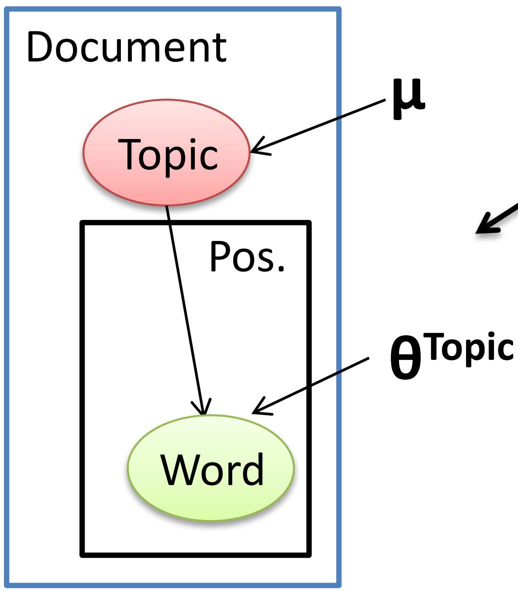
How to model each doc. as a “Mix of Topics”?

How to Model Mix of Topics ?



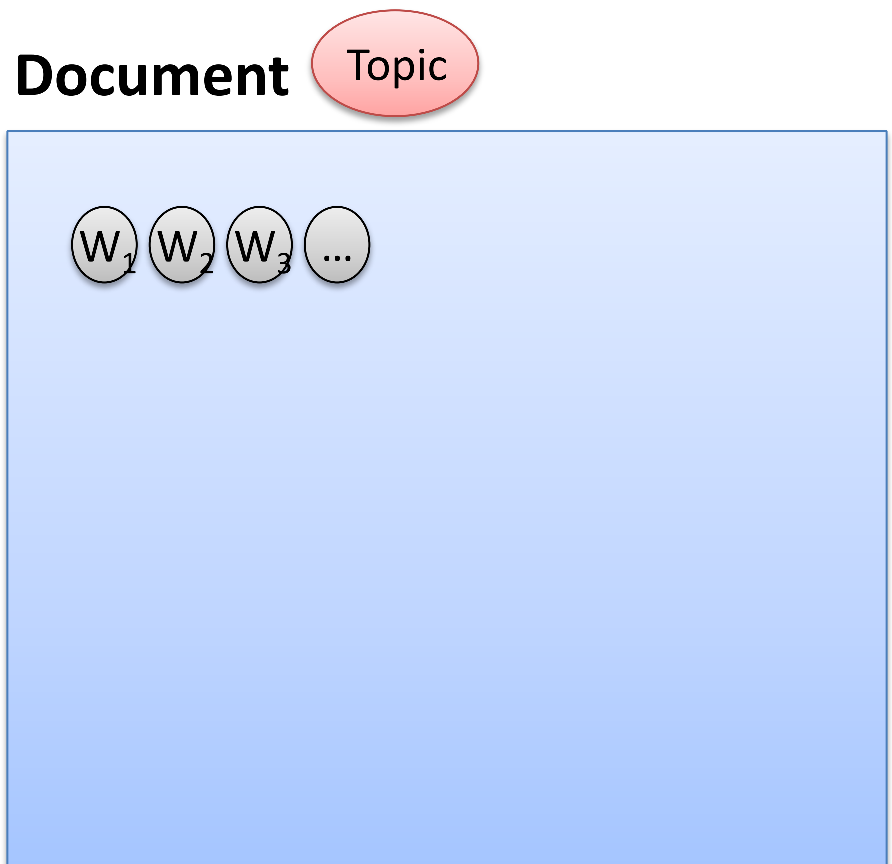


How to Model Mix of Topics ?

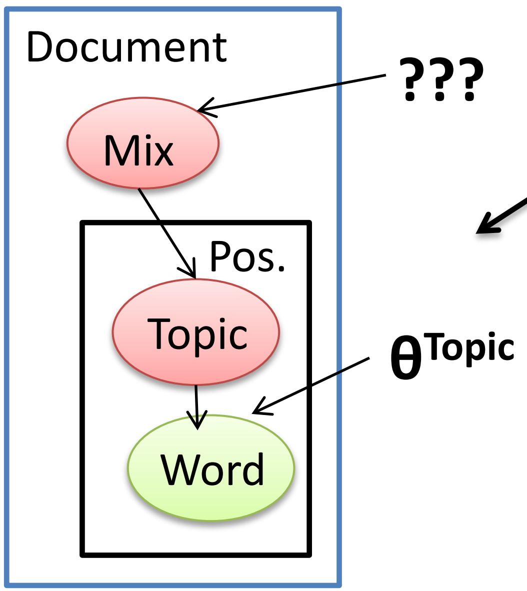


Relax “**one Topic per doc.**” assumption.
Instead, every doc. has a **“Mix”** of topics.

Mix = $(\mu_1, \mu_2, \mu_3, \ldots, \mu_K)$, $0 <= \mu_k <= 1$, $\Sigma_k \mu_k = 1$

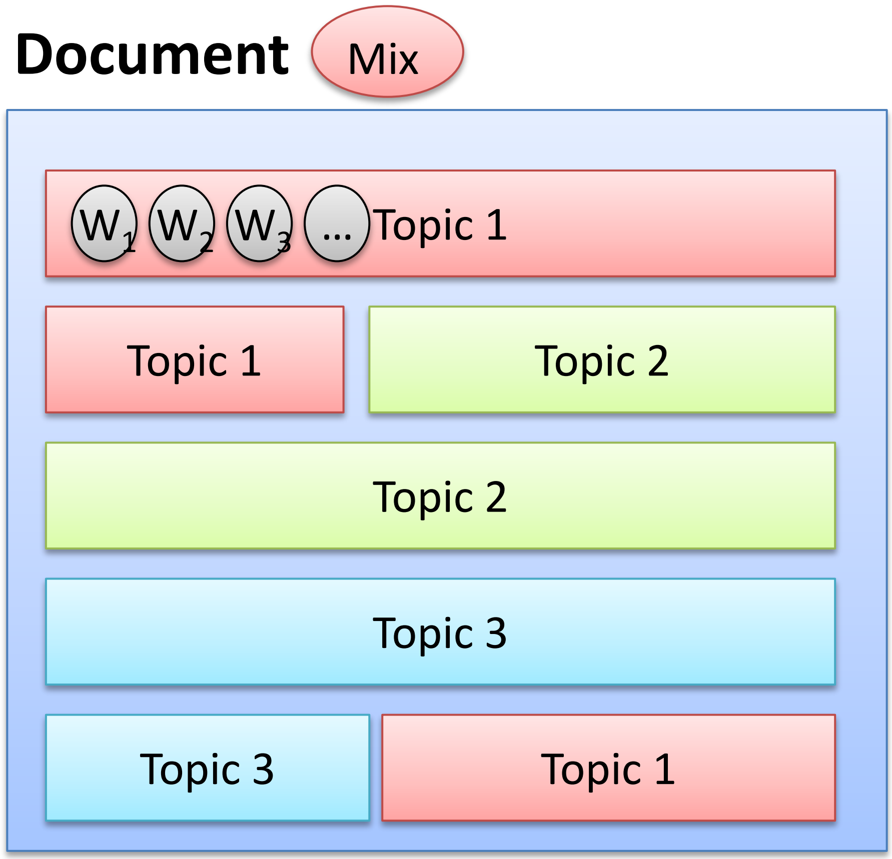


How to Model Mix of Topics ?

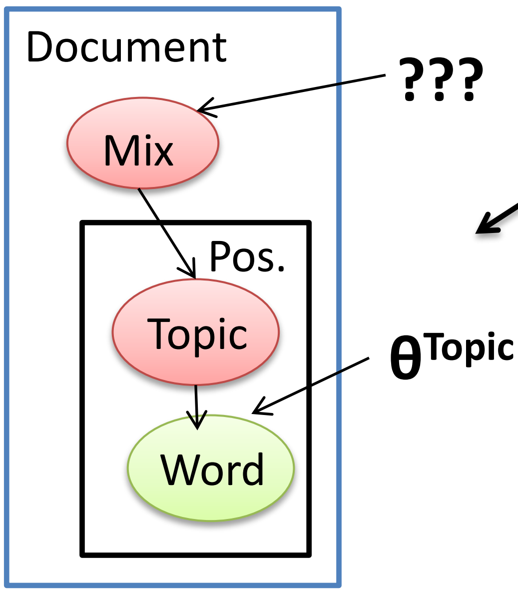


Relax "**one Topic per doc.**" assumption.
Instead, every doc. has a **"Mix"** of topics.

$Mix = (\mu_1, \mu_2, \mu_3, \ldots, \mu_K)$, $0 <= \mu_k <= 1$, $\Sigma_k \mu_k = 1$

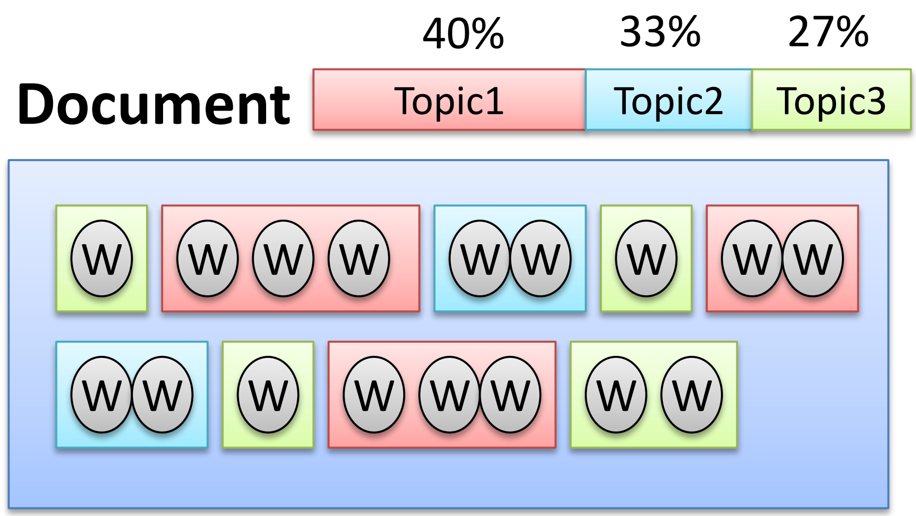


How to Model Mix of Topics ?



Relax "**one Topic per doc.**" assumption.
Instead, every doc. has a **"Mix"** of topics.

$\mathbf{Mix = (\mu_1, \mu_2, \mu_3, \ldots, \mu_K)\ ,\ 0 <= \mu_k <= 1,\ \ \Sigma_k\ \mu_k = 1}$



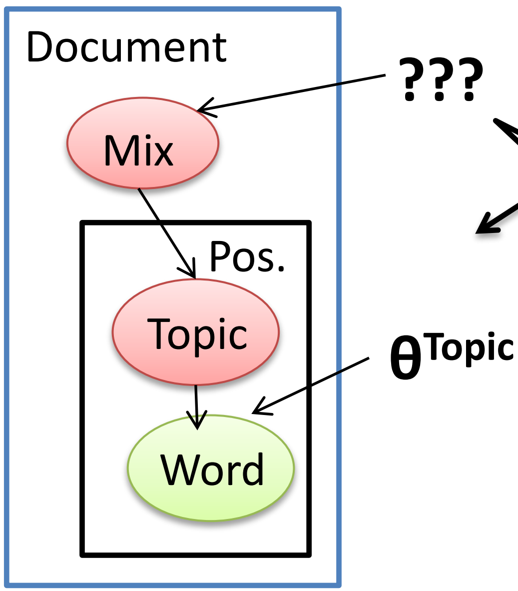
For all Documents **d**

1. draw Mix[**d**] ~ ?????

For all Position **p** in **d**

2. draw Topic[**d,p**]~ Multinomial(**Mix[d]**)
3. draw Word[**d,p**]~ Multinomial($\boldsymbol{\theta}^{\text{Topic}[\mathbf{d,p}]}$)

How to Model Mix of Topics ?



Relax "**one Topic per doc.**" assumption.
Instead, every doc. has a **"Mix"** of topics.

Mix = $(\mu_1, \mu_2, \mu_3, \ldots, \mu_K)$, $0 <= \mu_k <= 1$, $\Sigma_k \mu_k = 1$

Mix = $(\mu_1, \mu_2, \mu_3, \ldots, \mu_K)$ defines **a Multi(Mix) Dist. On Topics.**

While we need a "**Distribution on Mix**" ??

➔ Use "**Dirichlet distribution**".

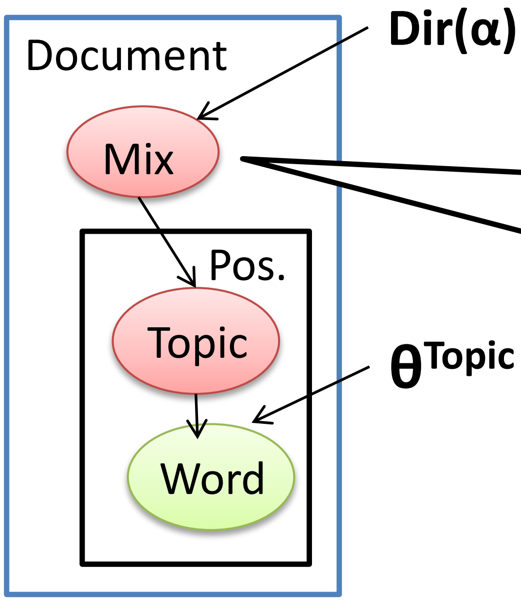
For all Documents **d**

1. draw Mix[**d**] ~ ?????

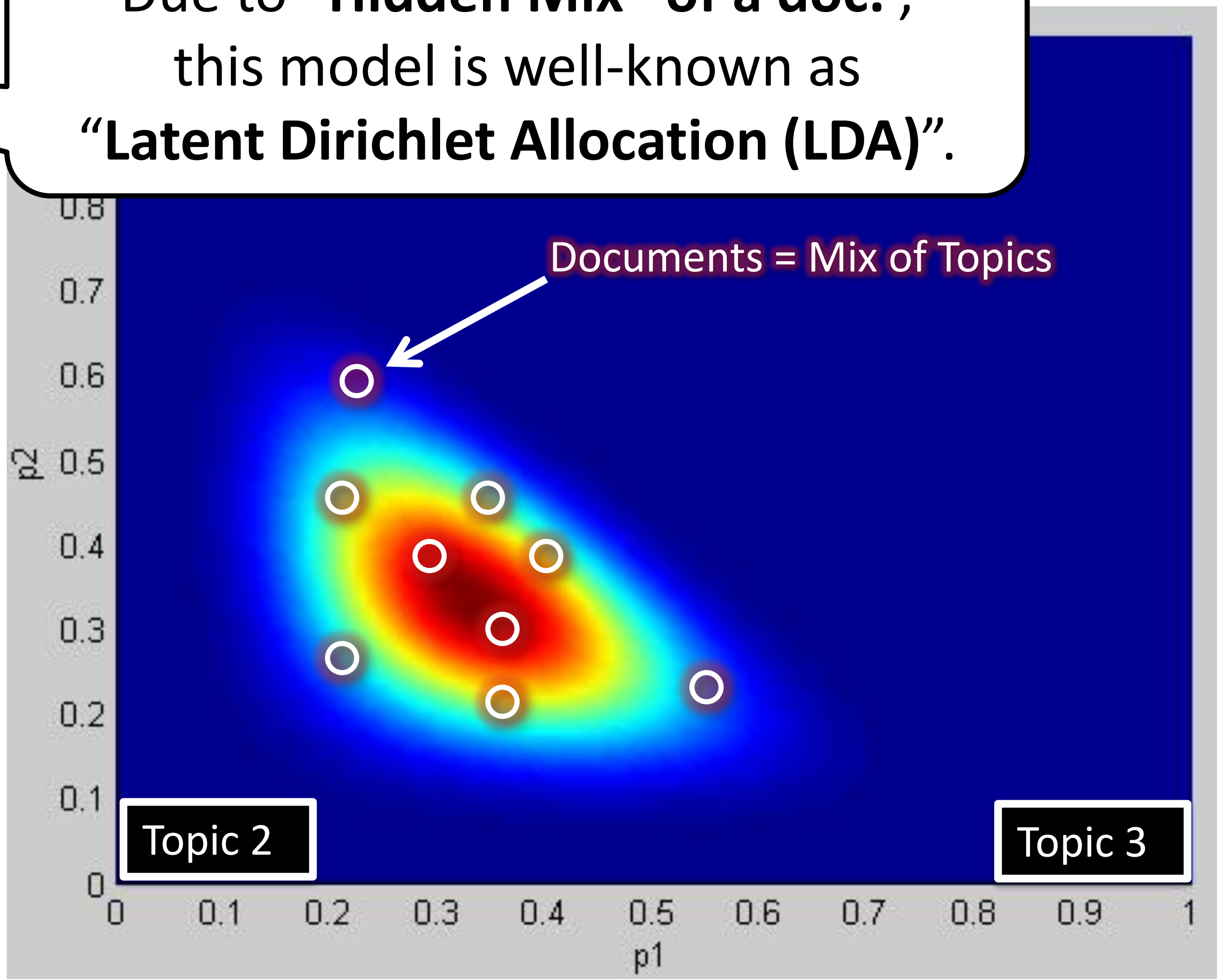
For all Position **p** in **d**

2. draw Topic[**d,p**]~ Multinomial(**Mix[d]**)
3. draw Word[**d,p**]~ Multinomial($\theta^{Topic[\mathbf{d,p}]}$)

Dirichlet Distribution



Due to **"Hidden Mix" of a doc.** ,
this model is well-known as
"**Latent Dirichlet Allocation (LDA)**".

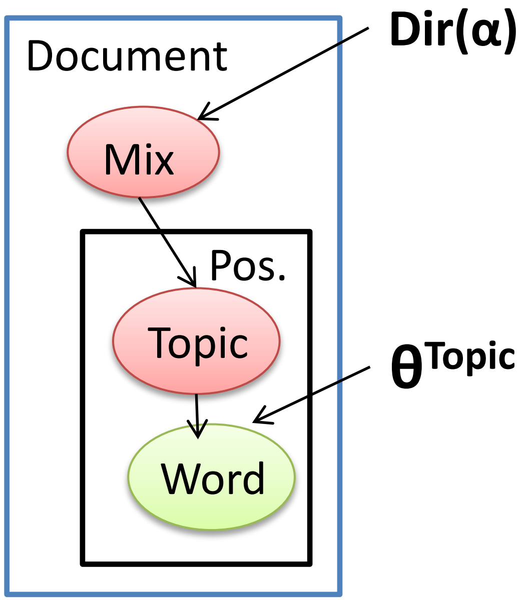


"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

From:
David M. Blei etal. ,
Latent Dirichlet Allocation, 2003

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

So, How to learn a LDA ?



The “E-Step” & “M-Step” is Much more difficult.

It’s out of scope. Please see :

David M. Blei etal. , Latent Dirichlet Allocation, 2003

Luckily, You don’t need to know exactly how to Learn & Infer.

There are tools online for Learning & Inference.

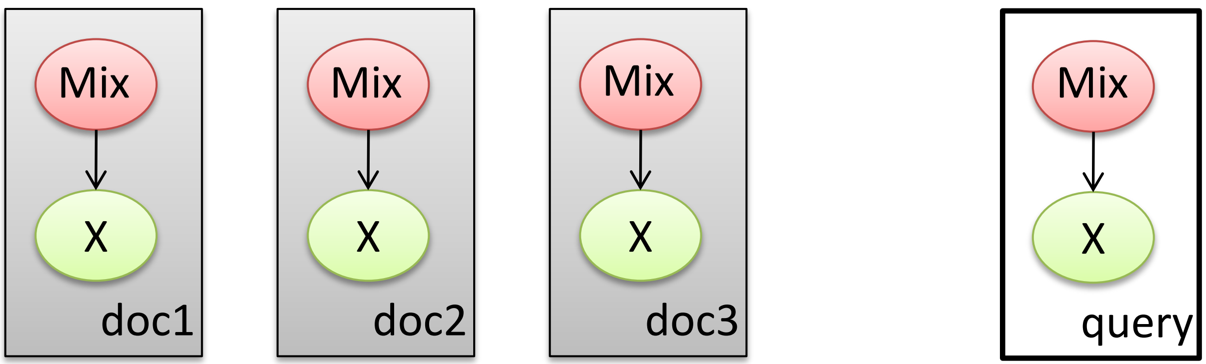
What important if knowing how to “design a model”.

Application for LDA : Search Engine

Retrieve in "Topic Level"
instead of "Word Level".

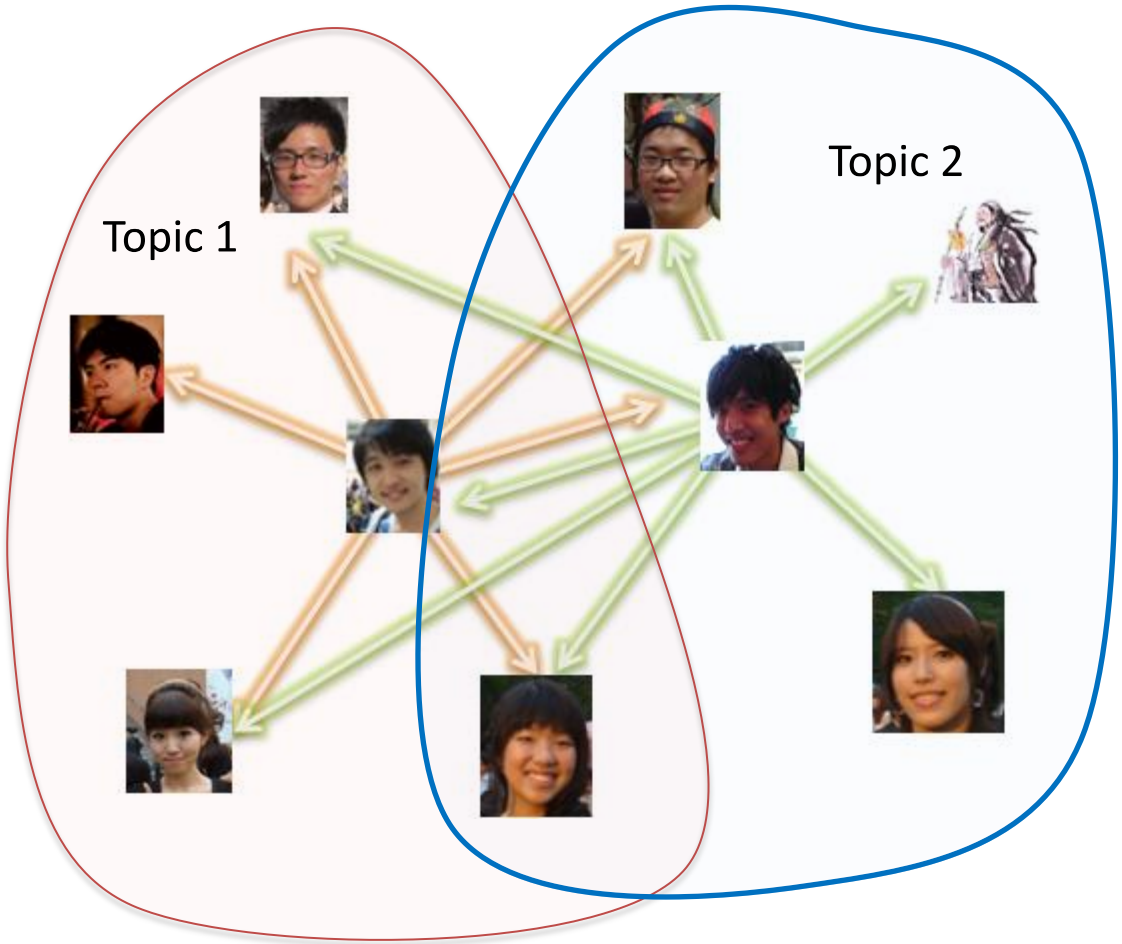
Given words in a document, we can infer its "topics".

Given words in a query, we can infer the "topics" user want.

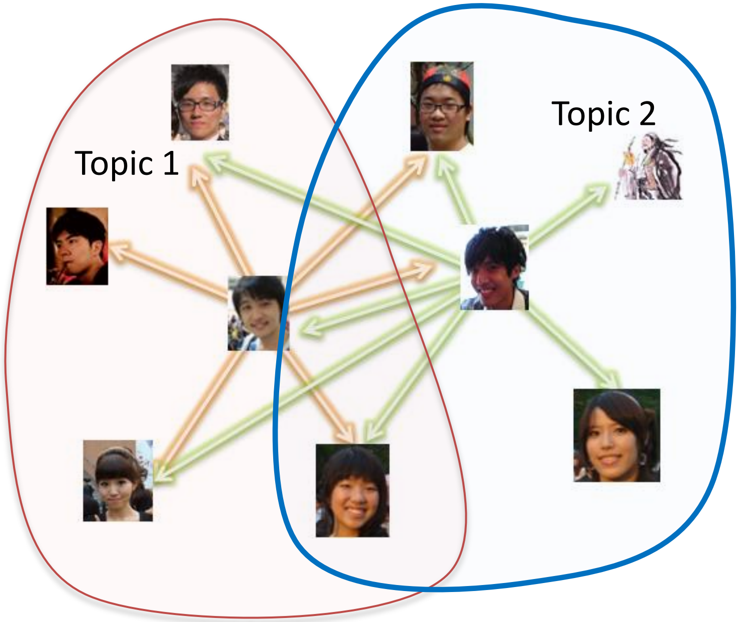


$$Score = \int P(Mix \mid Document) * P(Query \mid Mix)\ dMix$$

Application for LDA : Social Network



Application for LDA : Social Network



		1	1				1	1	1
	1		1						
	1	1							
					1	1	1		
				1		1	1		
				1	1		1		
	1			1	1	1		1	1
	1						1		1
	1						1	1	

Topic Model on Social Network

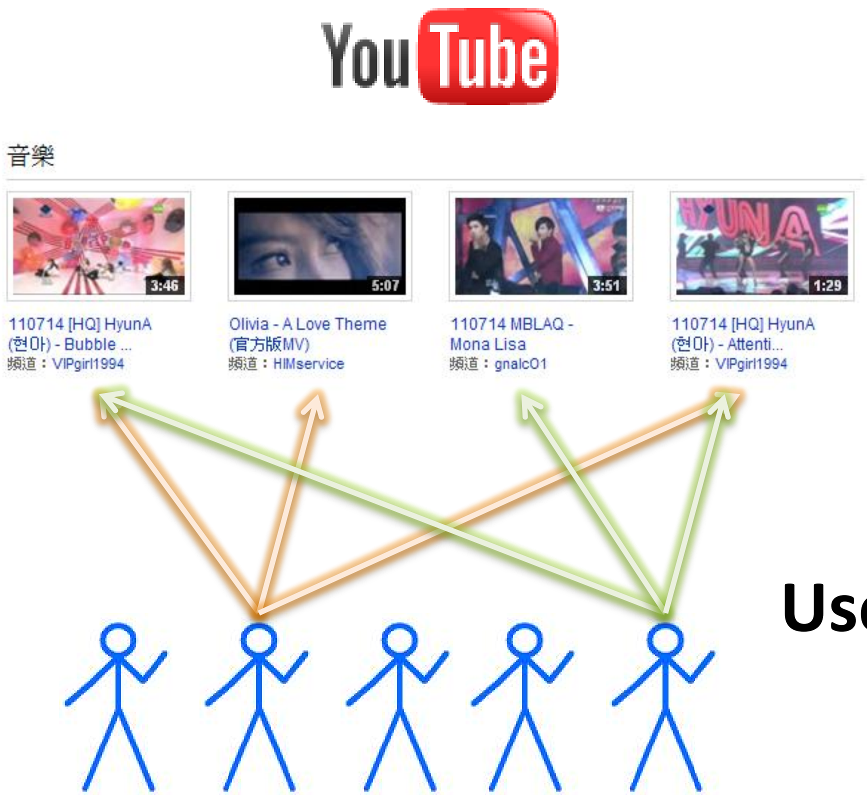
吉他社

熱舞社

資管系

	1	1				1	1	1
1		1						
1	1							
				1	1	1		
			1		1	1		
			1	1		1		
1			1	1	1		1	1
1						1		1
1						1	1	

Application for LDA : Recommendation

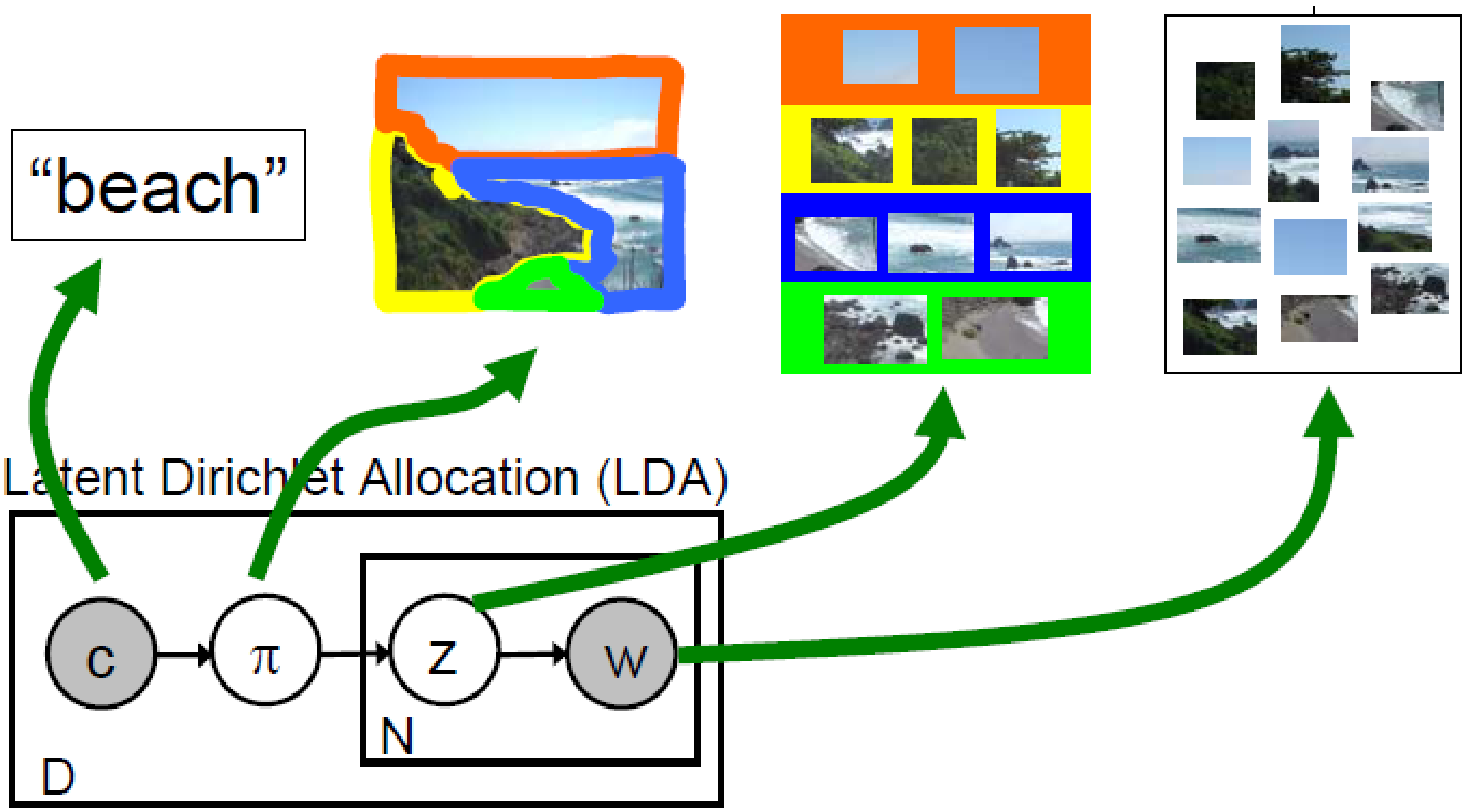


Movie

User

1	1	1					1	1
1	1	1						
1	1	1						
			1	1	1	1		
			1	1	1	1		
			1	1	1	1		
1	1	1	1	1	1	1	1	1
							1	1
							1	1
							1	1

Application for LDA : Image Categorize/Retrieval



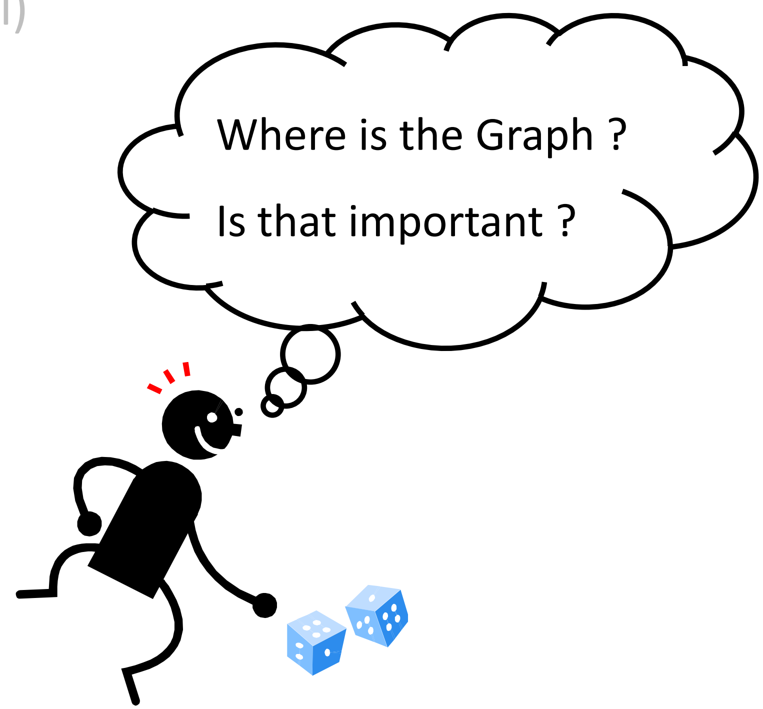
Fei-Fei et al. ICCV 2005

Overview

- What's Probabilistic Graphical Model for ?

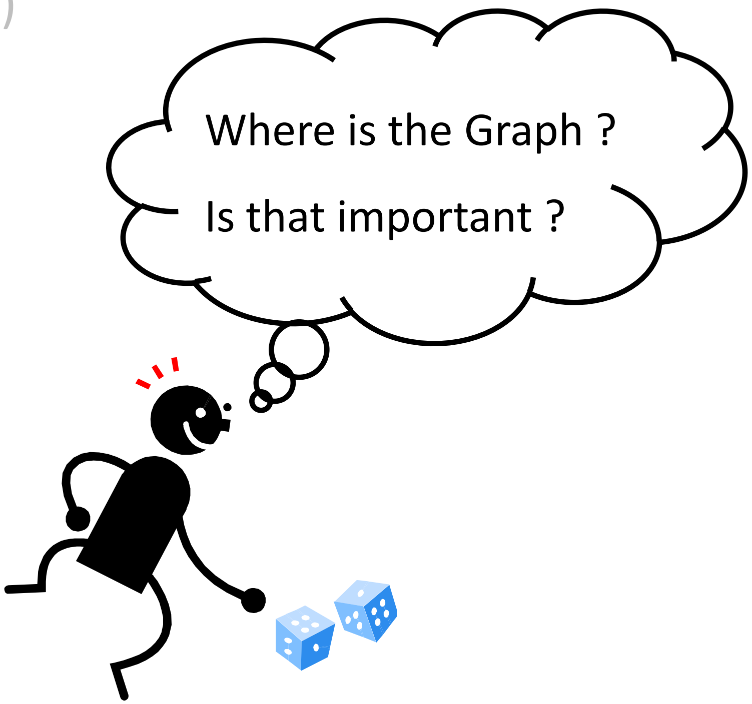
- Tasks in Graphical Model:
  - Modeling (Simple Probability Model)
  - Learning (MLE, MAP, Bayesian)
  - Inference (Bayes Rule ??)

- Examples
  - Topic Model (EM algorithm)
  - Hidden Markov Model
  - Markov Random Field



Overview

- What's Probabilistic Graphical Model for ?
- Tasks in Graphical Model:
  - Modeling (Simple Probability Model)
  - Learning (MLE, MAP, Bayesian)
  - Inference (Bayes Rule ??)
- Examples
  - Topic Model (EM algorithm)
  - Hidden Markov Model
  - Markov Random Field



Given a Problem: Binary Coding/Decoding

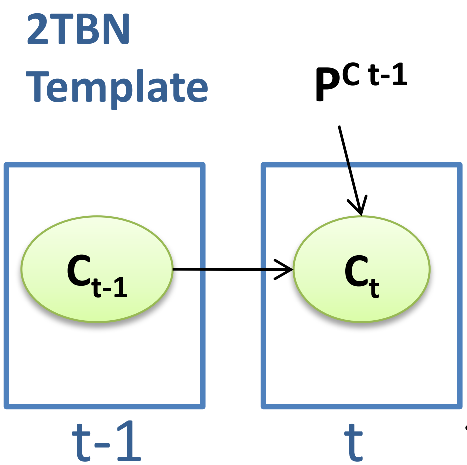
I		L	o	v	e		Y	o	u
001	00	0100	0100	0110	011	00	0111	0100	101

(Coding with different length, ex. Huffman Coding.)

Application Example :

1. Decode words from codes. (given coding pattern)
2. Learn Coding pattern from data.
3. Decide which coding method a data sequence uses.

A Naïve Model ---1^{order} Markov Chain



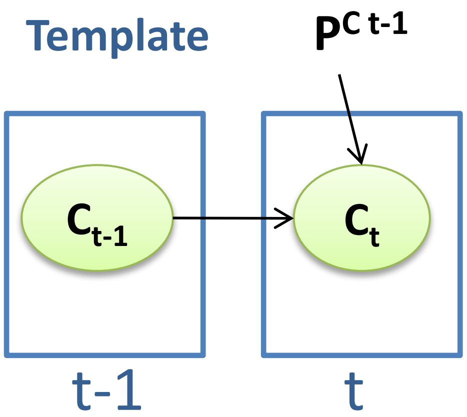
Design a model explaining this data:

010 1011 010 1001 010 1001 010 1011

A **2TBN template** specify how “before” affect “after”.

A Naïve Model ---1^{order} Markov Chain

2TBN Template



Design a model explaining this data:

010 1011 010 1001 010 1001 010 1011

Sample Generating Procedure:

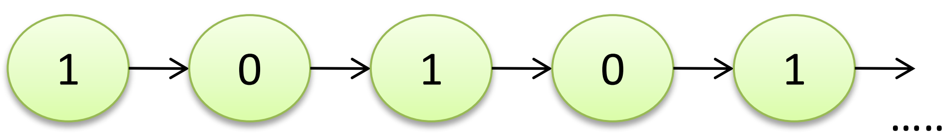
1. Draw C[t=1] ~ Bernoulli($\mathbf{p_0}$)

For t = 2 ~ T

2. Draw C[t] ~ Bernoulli($\mathbf{p}^{C[t-1]}$)

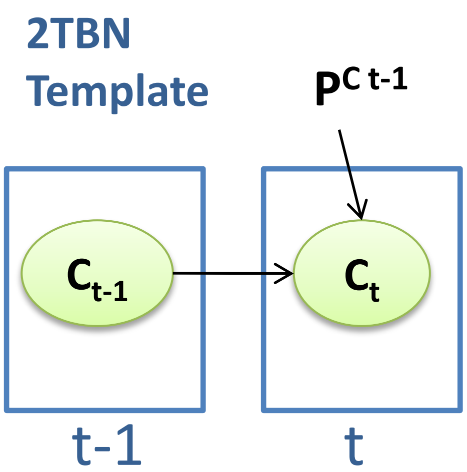
Ground Representation

C=0	C=1
0.2	0.8



$p^{C[t-1]}$	C_t=0	C_t=1
C_{t-1}=0	0.01	0.99
C_{t-1}=1	0.99	0.01

A Naïve Model ---1^{order} Markov Chain



Design a model explaining this data:

010 1011 010 1001 010 1001 010 1011

How to explain data with high likelihood ?

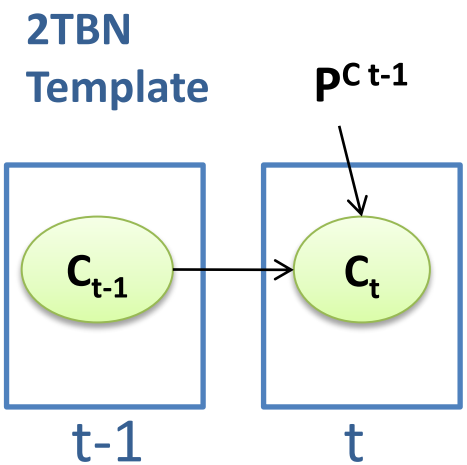
MLE estimate:

p_0

C=0	C=1
1	0

$p^{C[t-1]}$	C_t=0	C_t=1
C_{t-1}=0	$\frac{\#00}{\#0?}$	$\frac{\#01}{\#0?}$
C_{t-1}=1	$\frac{\#10}{\#1?}$	$\frac{\#11}{\#1?}$

A Naïve Model ---1^{order} Markov Chain



Design a model explaining this data:

010 1011 010 1001 010 1001 010 1011

How to explain data with high likelihood ?

MLE estimate:

p_0

C=0	C=1
1	0

$p^{C[t-1]}$	C_t=0	C_t=1
C_{t-1}=0	2/14	12/14
C_{t-1}=1	11/13	2/13

Likelihood = P(Data)

$= P(0)\, P(0|0)^2\, P(1|0)^{12}\, P(0|1)^{11}\, P(1|1)^2$

$= \quad 1*(2/14)^2\, (12/14)^{12}(11/13)^{11}(2/13)^2$

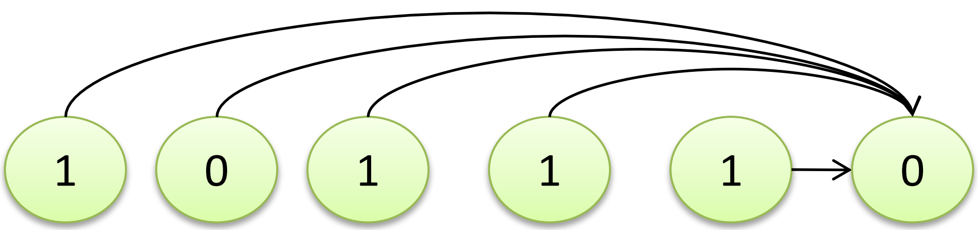
Can we do better ?

Pattern “010” “1011” ...
not explicitly handled.

How to Model a Pattern ?

Assume in data, “**101110**” is a **frequent pattern**.
(or we have known it is a coding for some word.)

A Naïve Approach : 5^{order} Markov Chian



C_{t-1}~C_{t-5}	C_t=0	C_t=1
00000	?	?
00001	?	?
……		
10111	High	Low
……		
11111	?	?

Table Size = #of params = 2^6

➔Intractability & Overfitting

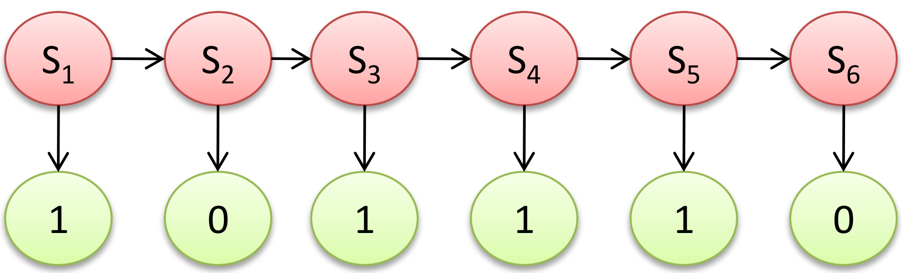
When handling a coding with
10 values = {1,2….,10} ➔ Size=10^6 !!!

How to Model a Pattern ?

Assume in data, “**101110**” is more **frequent** than usual.
(or we have known it is a coding for some word.)

Observation : a **pattern** can be produced with a **State Machine**

A **State Machine** is a special case of **1^{order} Markov Chain**:



1^{order} “Hidden” Markov Chain
Suffice to produce the pattern !

If there are **K patterns**, we can have **K State Machines** for them .

CPD of Red variable :

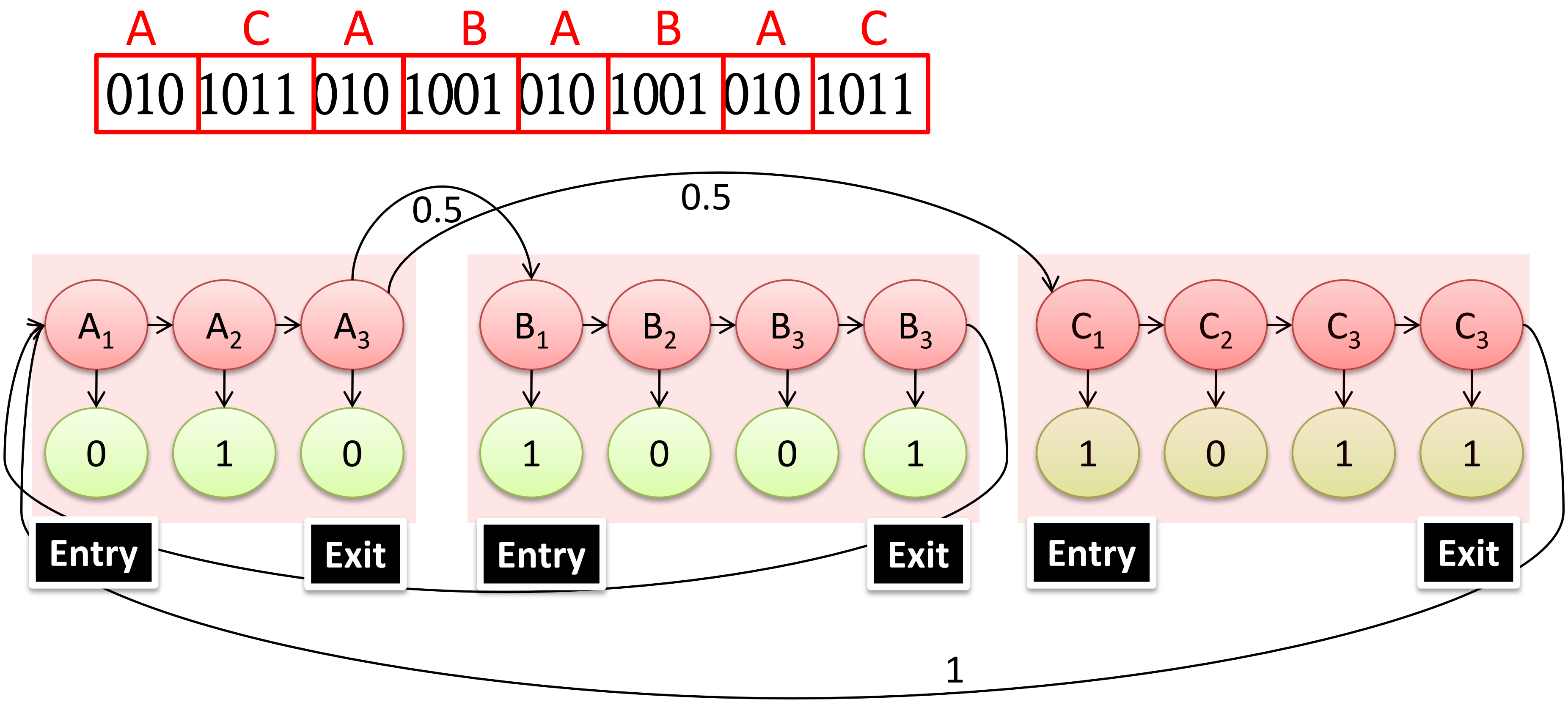
	S_1	S_2	S_3	S_4	S_5	S_6
S_1		1				
S_2			1			
S_3				1		
S_4					1	
S_5						1

CPD of Green variable :

	0	1
S_1	0	1
S_2	1	0
S_3	0	1
S_4	0	1
S_5	0	1
S_6	1	0

How to Model Multiple Patterns?

Design 3 state machines (A,B,and C) for the 3 patterns :



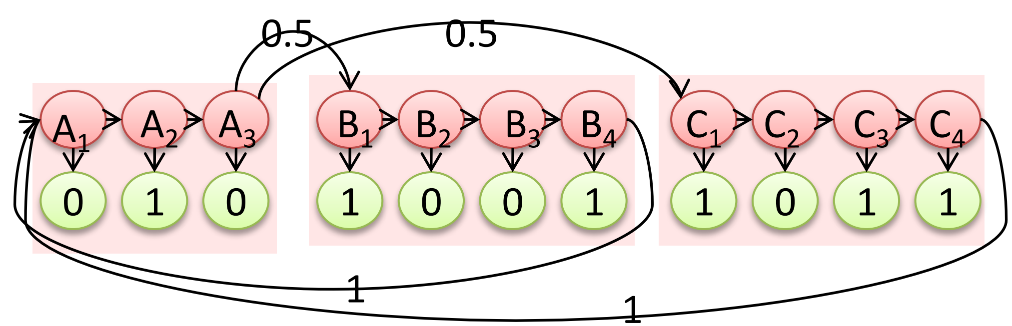
Consider “**Transition Probability**” among patterns.

How to Model Multiple Patterns?

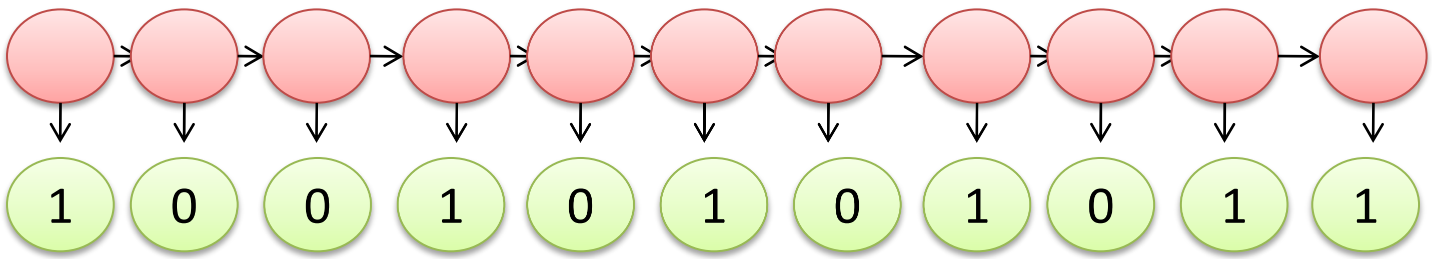
Transition Table of the “Hidden Markov Chain”

	A_1	A_2	A_3	B_1	B_2	B_3	B_4	C_1	C_2	C_3	C_4
A_1		1									
A_2			1								
A_3				.5				.5			
B_1					1						
B_2						1					
B_3							1				
B_4	1										
C_1									1		
C_2										1	
C_3											1
C_4	1										

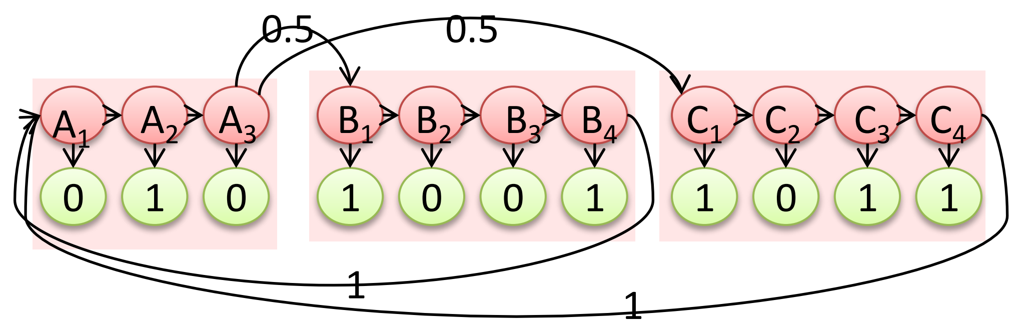
Transition Diagram of “Hidden Markov Chain”



How to decoding (Inference) ?

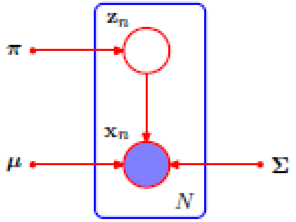


Transition Diagram of
"Hidden Markov Chain"



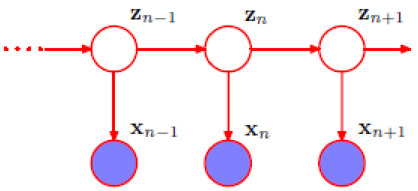
In terms of difficulty, there are 3 types of inference problem.

- Inference which is easily solved with Bayes rule.



- Inference which is tractable using some dynamic programming technique.

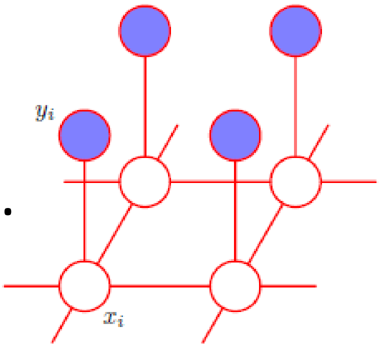
  (e.g. Variable Elimination or J-tree algorithm)



- Inference which is proved intractable

  & should be solved using some Approximate Method.

  (e.g. Approximation with Optimization or Sampling technique.)



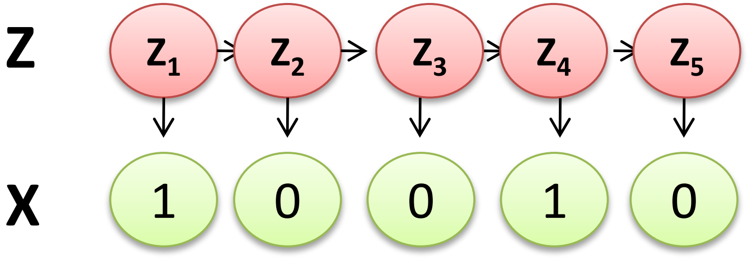
Most Probable Assignment

- Given Data = $\{X_1=x_1, \ldots, X_D=x_D\}$ and some other variables $Z=\{Z_1, \ldots, Z_k\}$ unspecified , Most Probable Assignment of Z is given by:

$$MPA(Z \mid X) = \arg\max_{Z} \; P(Z \mid X)$$

$$= \arg\max_{Z} \; \frac{P(X \mid Z)P(Z)}{\mathrm{P(X)}} = \arg\max_{Z} \; P(X \mid Z)P(Z)$$

$$= \arg\max_{Z_1 \sim Z_5} \; P(1 \mid Z_1) * P(0 \mid Z_2)P(Z_2 \mid Z_1) \ldots\ldots * P(0 \mid Z_5)P(Z_5 \mid Z_4)$$



$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_{A} F(A, B)$$

For every choice of B, we decide an $A^*(B)=\text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

F(A,B)	a1	a2	a3
b1	1	2	4
b2	3	5	7
b3	9	8	6

$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_A F(A,B)$$

For every choice of B, we decide an $A^*(B) = \text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

B	A(B)	M(B)
b1	a3	4
b2		
b3		

F(A,B)	a1	a2	a3
b1	1	2	4
b2	3	5	7
b3	9	8	6

$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_{A} F(A,B)$$

For every choice of B, we decide an $A^*(B)=\text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

B	A(B)	M(B)
b1	a3	4
b2	a3	7
b3		

F(A,B)	a1	a2	a3
b1	1	2	4
b2	3	5	7
b3	9	8	6

$M(B) = \max_A F(A,B)$: maxMarginal of B

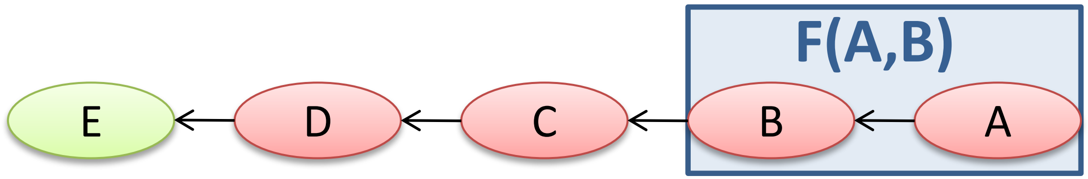
$$M(B) = \max_{A} F(A,B)$$

For every choice of B, we decide an $A^*(B) = \mathrm{argmax}_A\, F(A,B)$ with $M(B) = F(A^*(B), B)$.

B	A(B)	M(B)
b1	a3	4
b2	a3	7
b3	a1	9

F(A,B)	a1	a2	a3
b1	1	2	4
b2	3	5	7
b3	9	8	6

Most Probable Assignment



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

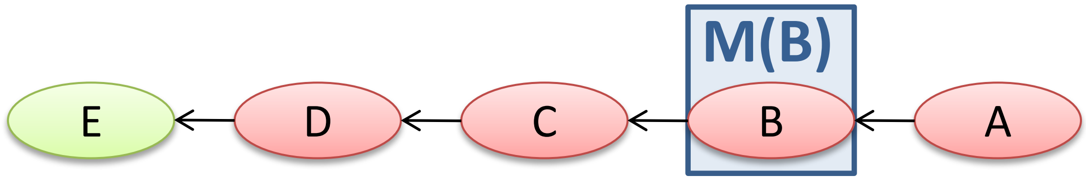
$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A \underline{P(B \mid A) P(A)}$$

F(A,B)

F(A,B)	a1	a2	a3
b1	…	…	…
b2	…	…	…
b3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

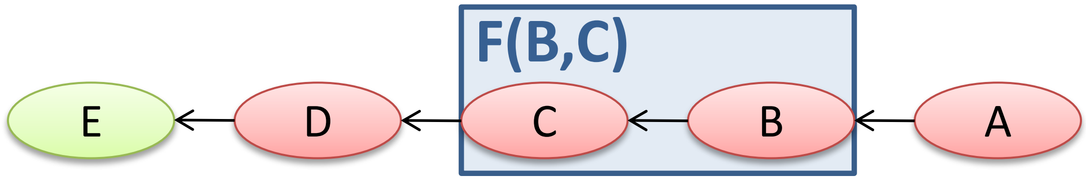
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \underline{\max_{A} P(B \mid A) P(A)}$$

M(B) = $\max_A$ F(A,B)

B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

F(A,B)	a1	a2	a3
b1	…	…	…
b2	…	…	…
b3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e,D,C,B,A)$$

$$=\max_{D}\max_{C}\max_{B}\max_{A} P(E=e\,|\,D)P(D\,|\,C)P(C\,|\,B)P(B\,|\,A)P(A)$$

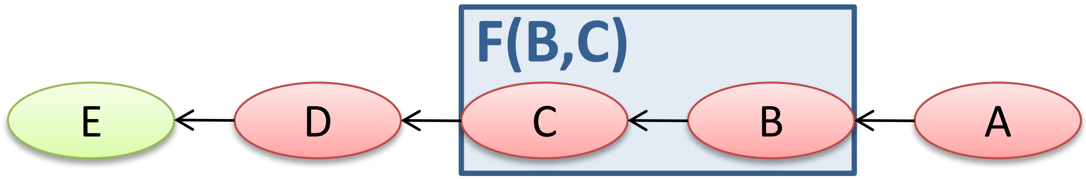
$$=\max_{D} P(E=e\,|\,D)\max_{C} P(D\,|\,C)\max_{B} P(C\,|\,B)\max_{A} P(B\,|\,A)P(A)$$

F(B,C)=P(C|B)M(B)

P(C|B)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

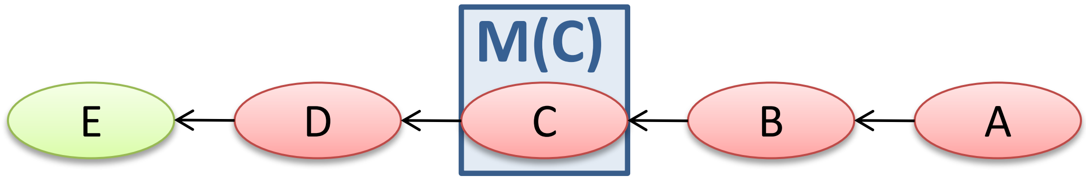
$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)$$

F(B,C)=P(C|B)M(B)

F(B,C)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

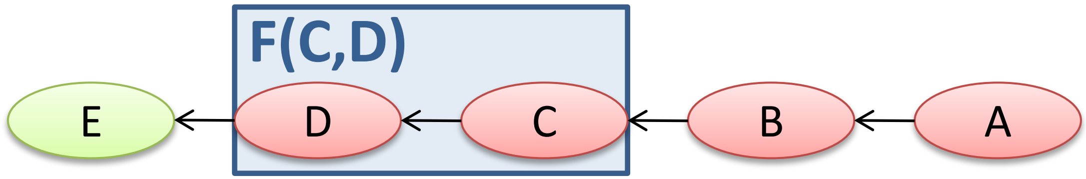
$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)$$

M(C) = max$_B$ F(B,C)

C	B*(C)	M(C)
c1	b3	...
c2	b1	...
c3	b2	...

F(B,C)	b1	b2	b3
c1	...	...	...
c2	...	...	...
c3	...	...	...

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

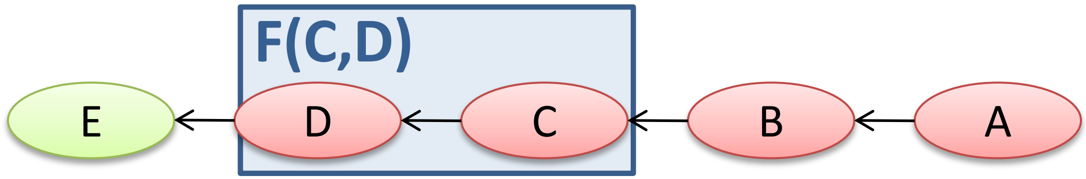
$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)$$

F(C,D) = P(D|C)M(C)

P(D|C)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

C	B*(C)	M(C)
c1	b3	…
c2	b1	…
c3	b2	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

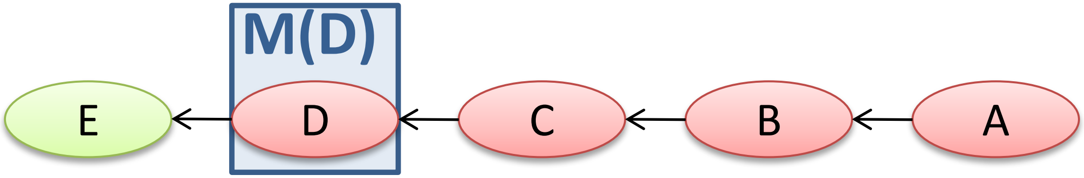
$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A) P(A)$$

F(C,D) = P(D|C)M(C)

F(C,D)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

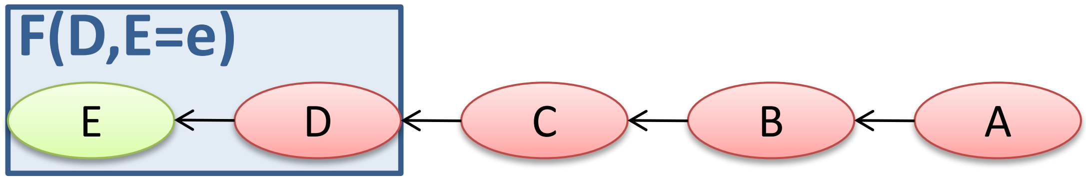
$$= \max_D P(E=e \mid D) \underline{\max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)}$$

M(D)=max$_C$ F(C,D)

D	C*(D)	M(D)
d1	c1	...
d2	c2	...
d3	c3	...

F(C,D)	c1	c2	c3
d1	...	...	...
d2	...	...	...
d3	...	...	...

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e, D, C, B, A)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

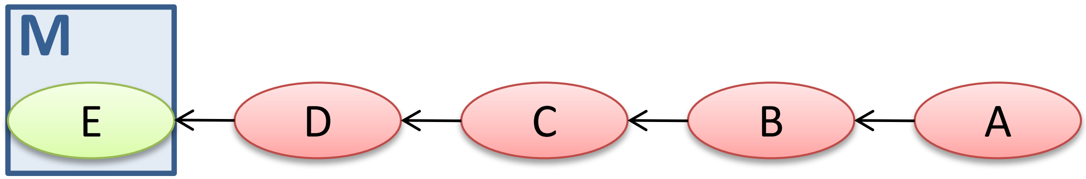
$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)$$

F(D)=P(E=e|D)M(D)

P(E=e|D)	d1	d2	d3
e	…	…	…

D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e,D,C,B,A)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

M= max$_D$ F(D)

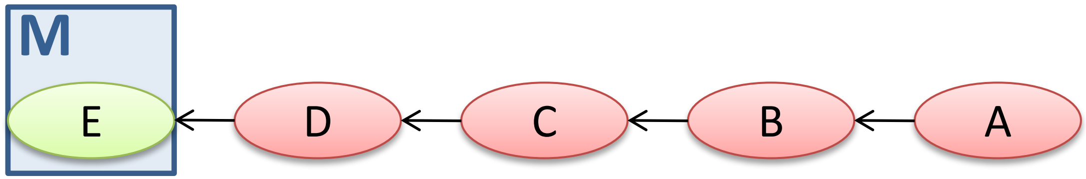
D*	M
d2	…

F(D,E=e)	d1	d2	d3
e	…	…	…

What we get ? ➔ **M = max$_{ABCD}$ P(A,B,C,D,E=e)**

What we want ? ➔ **(A*,B*,C*,D*) = argmax$_{ABCD}$ P(A,B,C,D,E=e)**

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(E=e,D,C,B,A)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

What we want ? ➔ (A*,B*,C*,D*) = argmax$_{ABCD}$ P(A,B,C,D,E=e)

(a1 , b1, c2, d2)

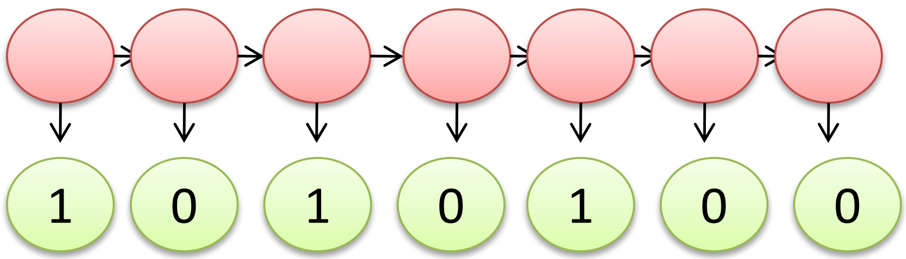
D*	M
d2	…

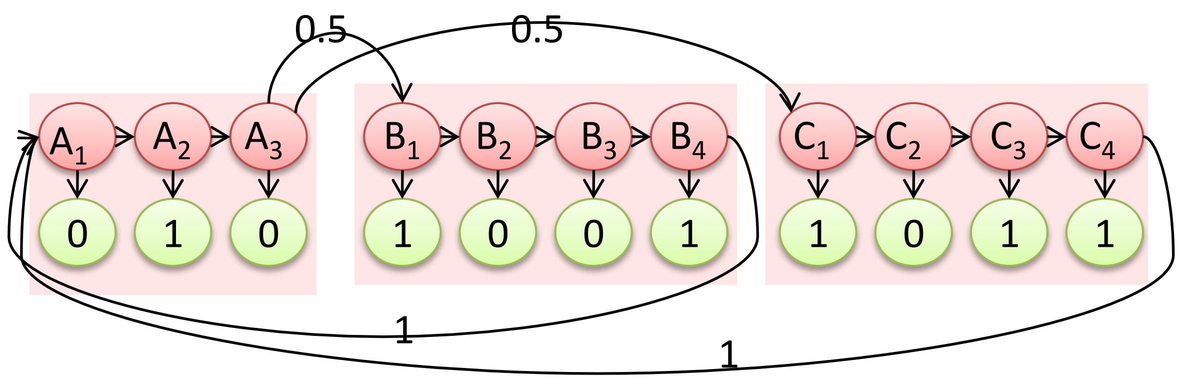
D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

C	B*(C)	M(C)
c1	b3	…
c2	b1	…
c3	b2	…

B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

How to decode (Inference) ?





How to decode (Inference) ?

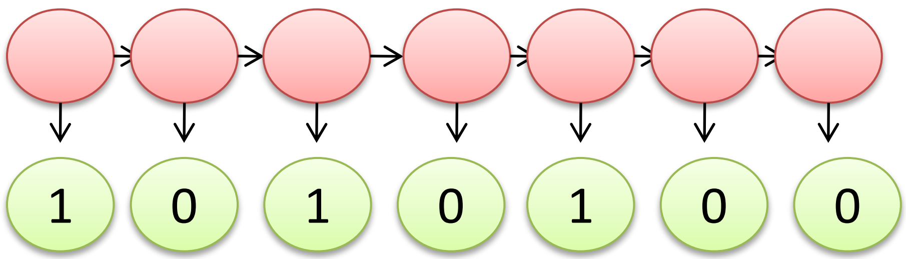
1 $C_4 \rightarrow A_1$

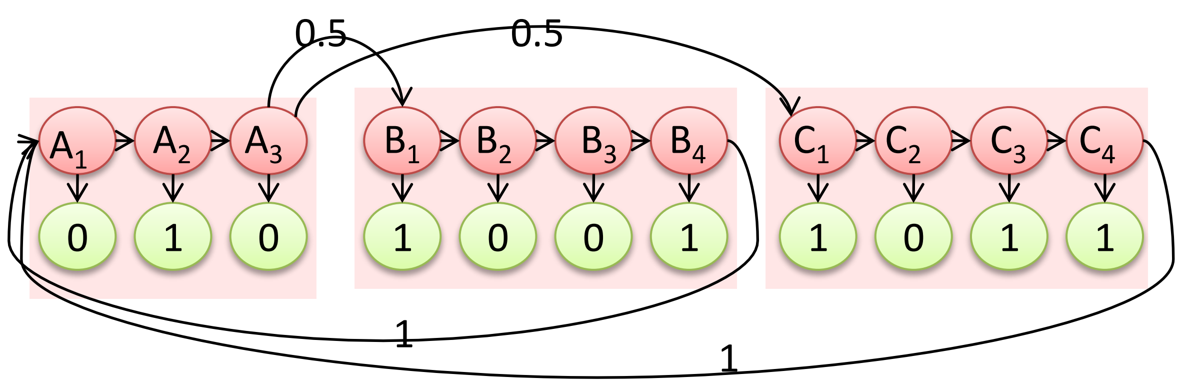
1 $C_1 \rightarrow C_2$

1 $B_4 \rightarrow A_1$

1 $B_1 \rightarrow B_2$

1 $A_2 \rightarrow A_3$





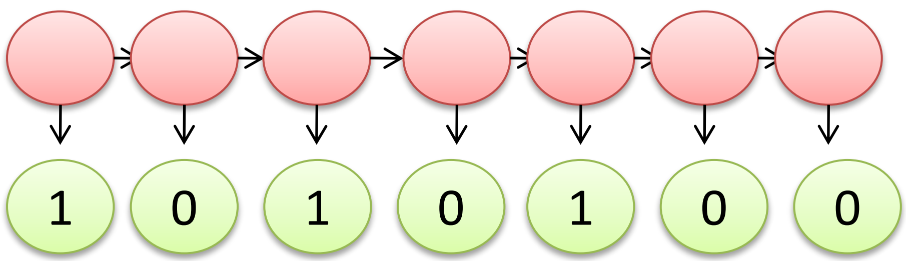
How to decode (Inference) ?

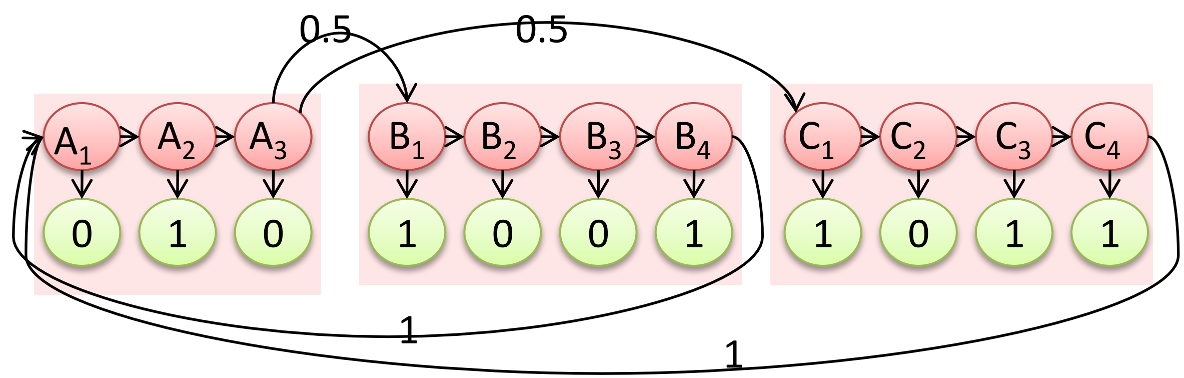
1 A_1

1 C_2

1 B_2

1 A_3





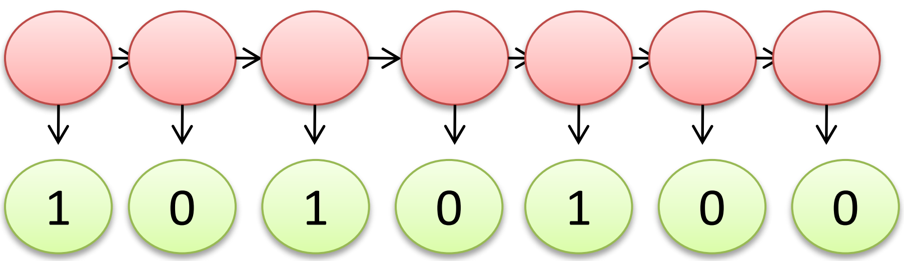
How to decode (Inference) ?

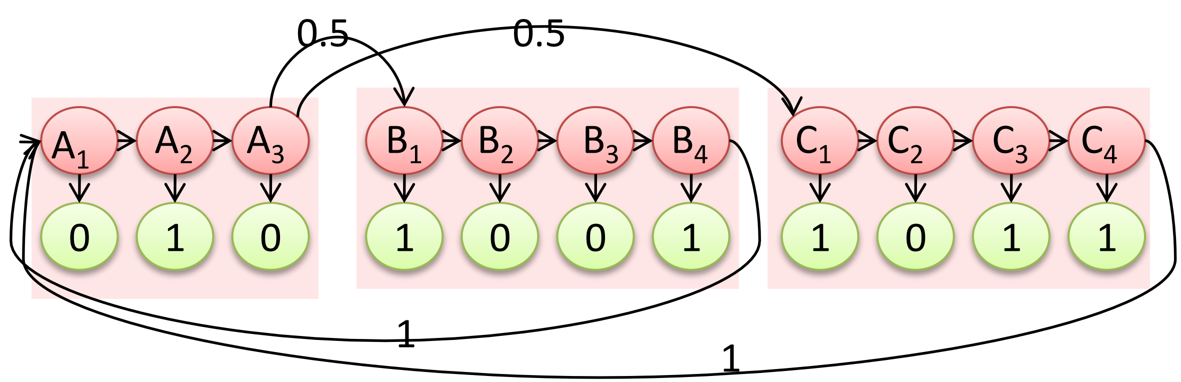
1 $A_1 \rightarrow A_2$

1 $C_2 \rightarrow C_3$

.5 B_2 C_1

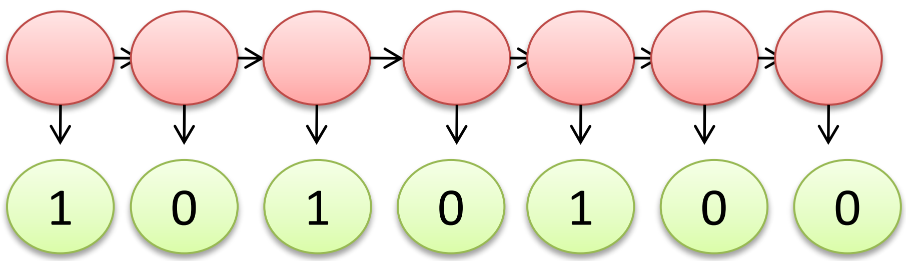
.5 $A_3 \rightarrow B_1$

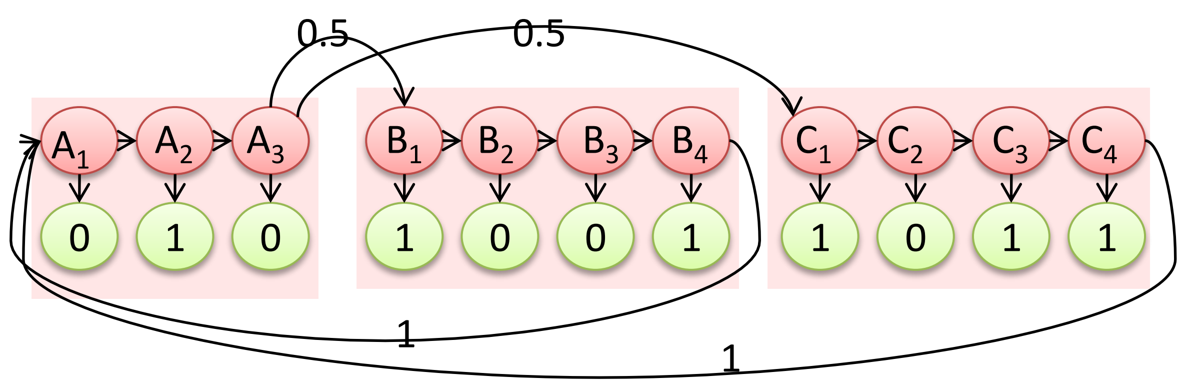




How to decode (Inference) ?

1	A_2
1	C_3
.5	C_1
.5	B_1





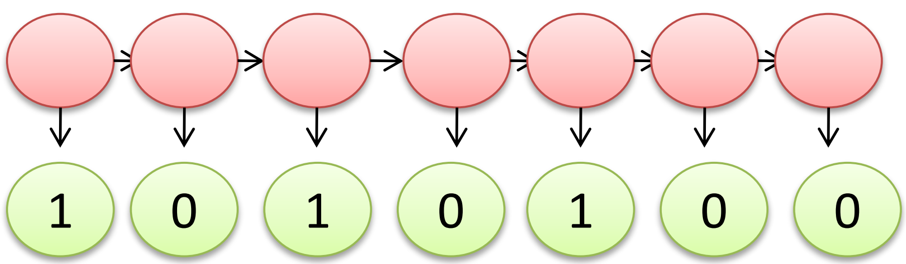
How to decode (Inference) ?

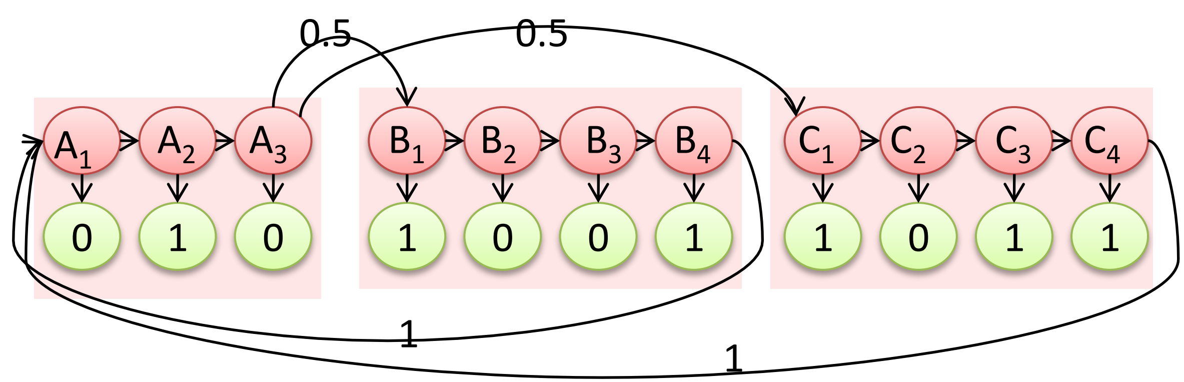
1 $A_2 \rightarrow A_3$

C_3

.5 $C_1 \rightarrow C_2$

.5 $B_1 \rightarrow B_2$



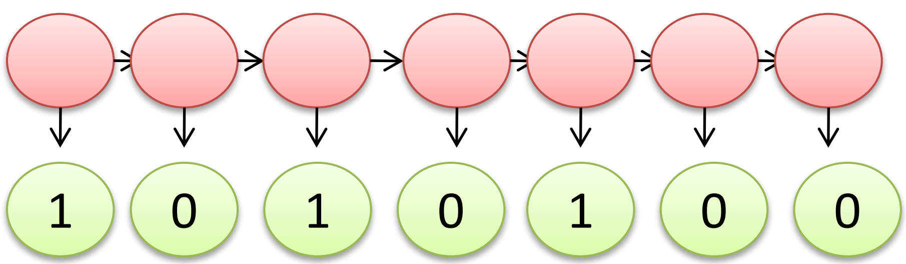


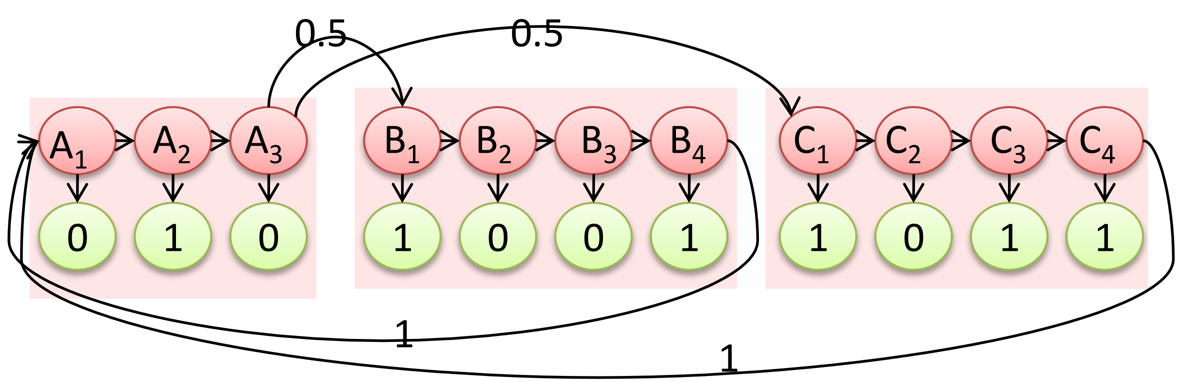
How to decode (Inference) ?

1 A_3

.5 C_2

.5 B_2





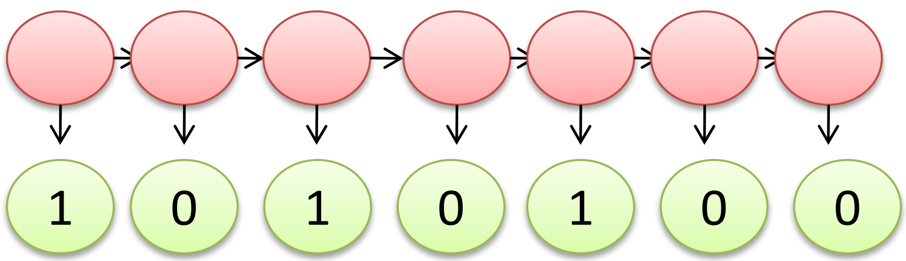
How to decode (Inference) ?

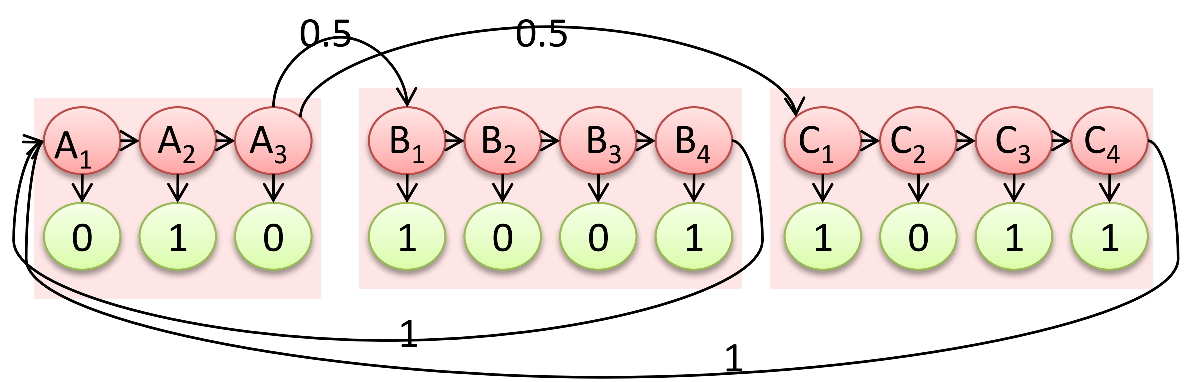
.5 $A_3 \rightarrow B_1$

.5 $\searrow C_1$

.5 $C_2 \rightarrow C_3$

B_2



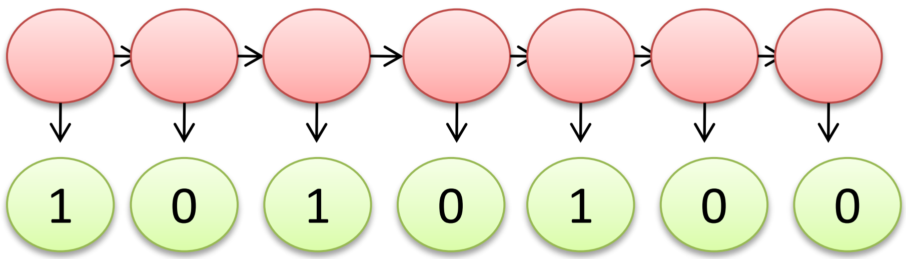


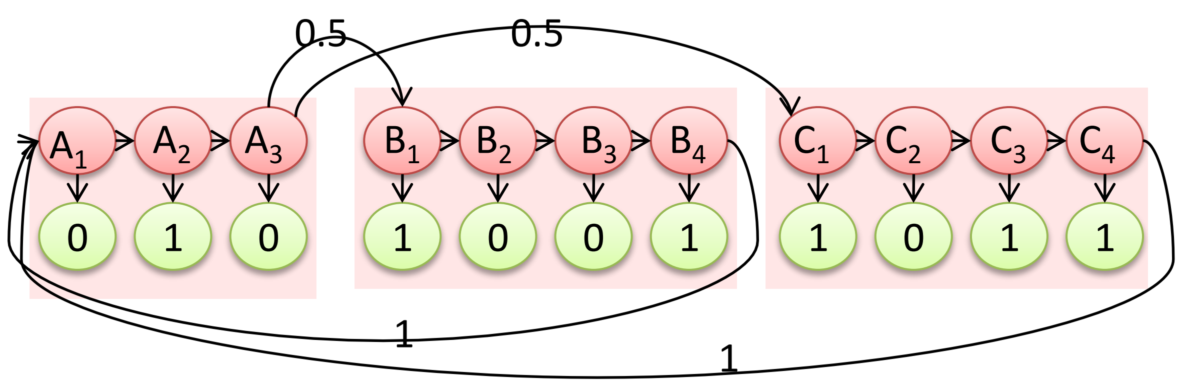
How to decode (Inference) ?

.5 B_1

.5 C_1

.5 C_3



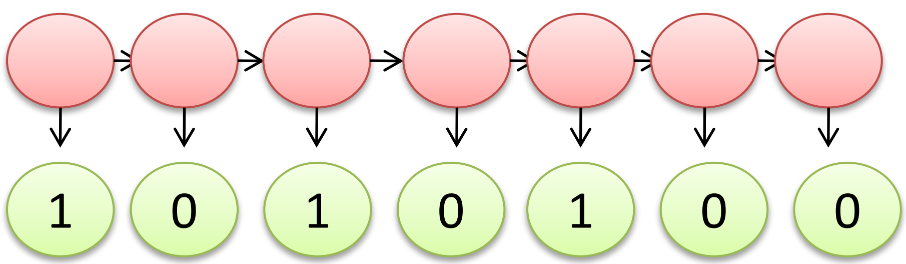


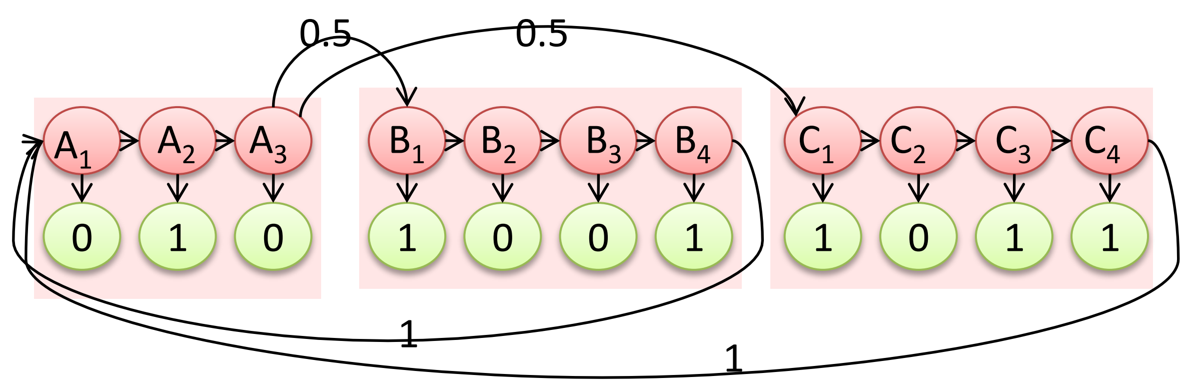
How to decode (Inference) ?

.5 $B_1 \rightarrow B_2$

.5 $C_1 \rightarrow C_2$

C_3





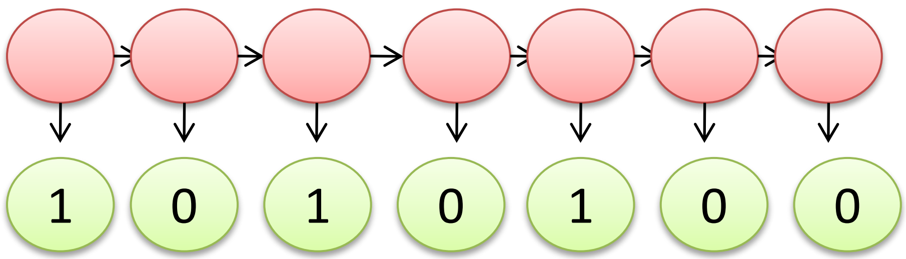
How to decode (Inference) ?

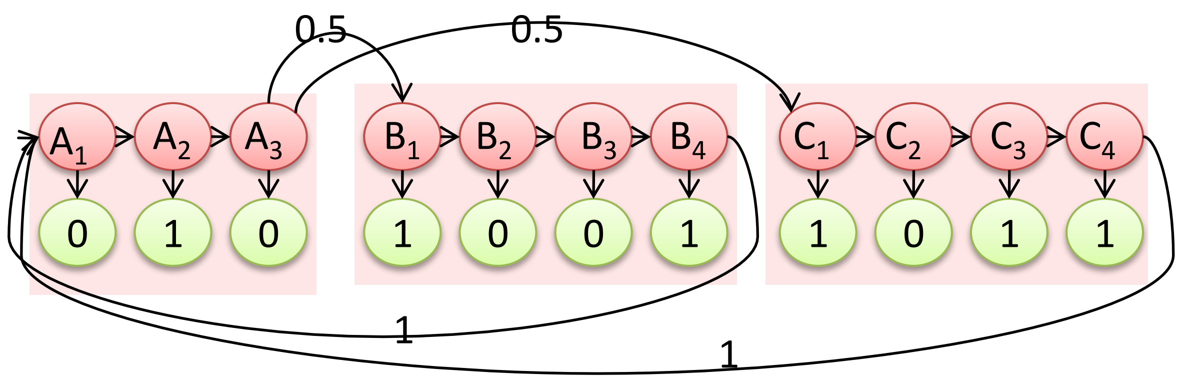
.5

.5

B_2

C_2





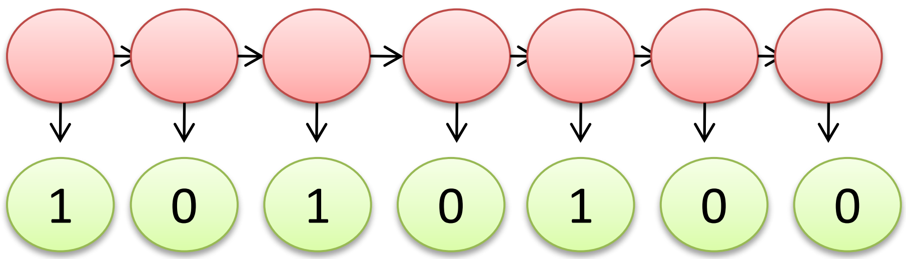
How to decode (Inference) ?

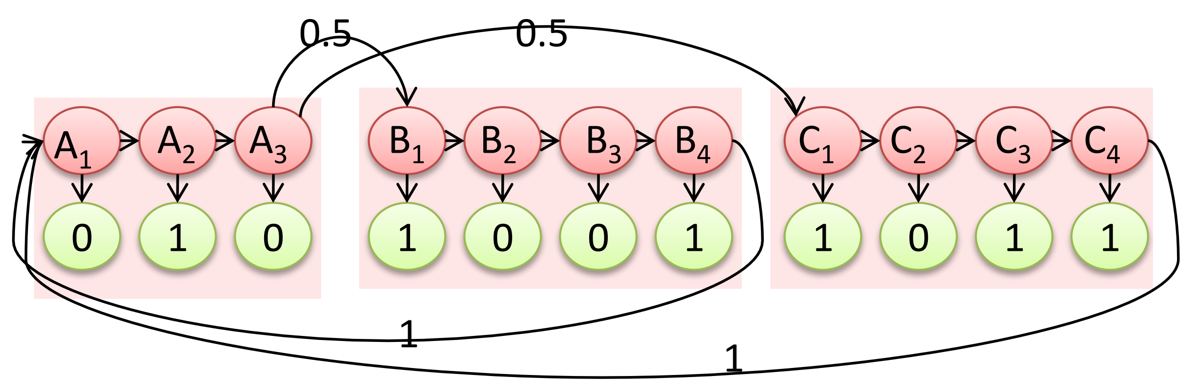
.5

$B_2 \rightarrow B_3$

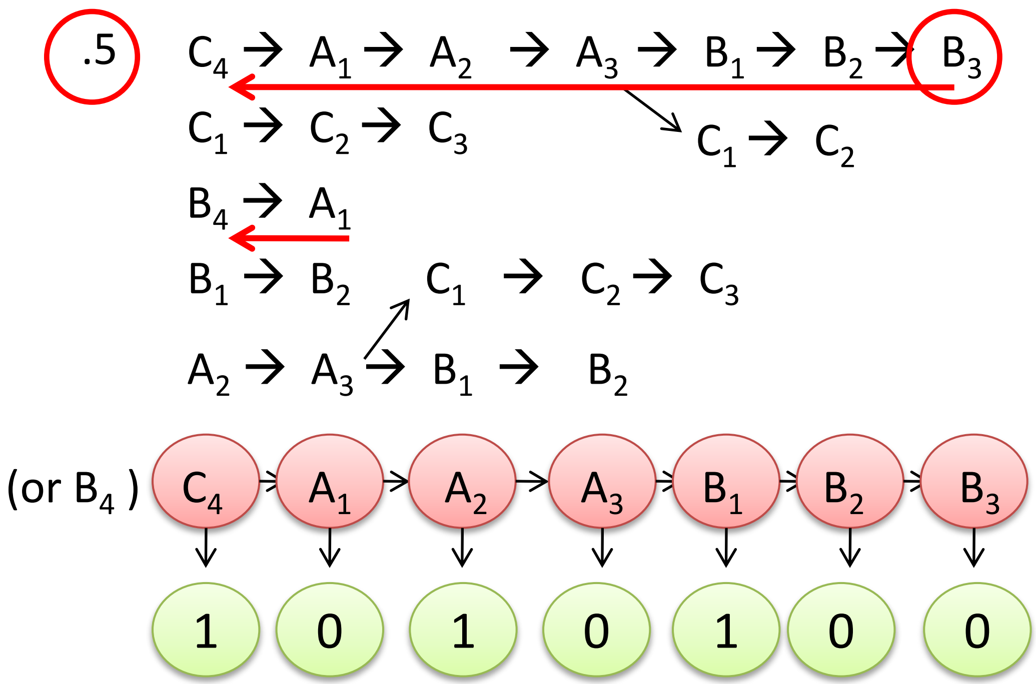
MPA Likelihood = $\max_Z P(X|Z)$

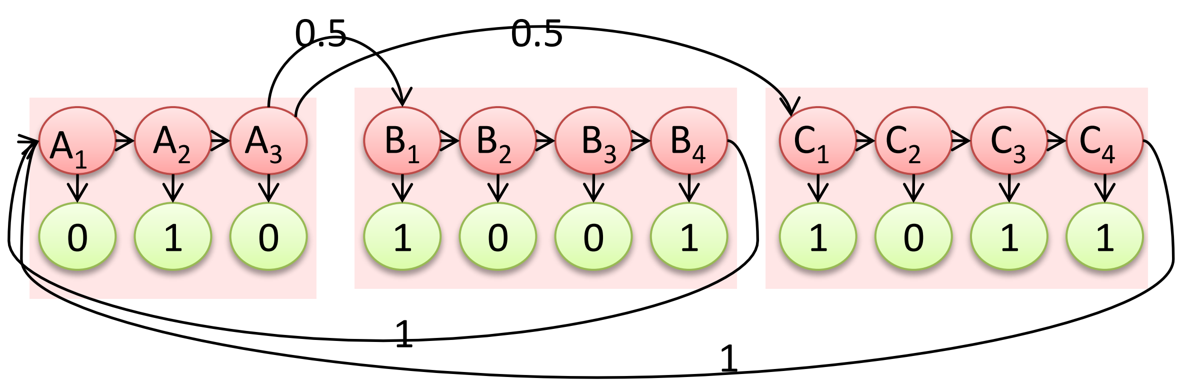
C_2





How to decode (Inference) ?





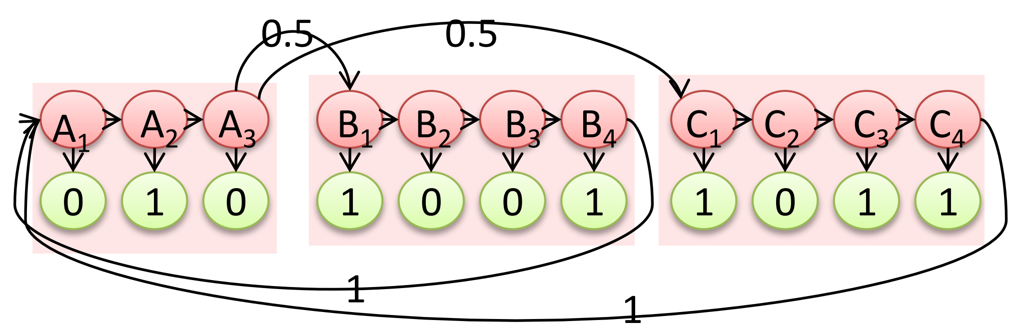
How to Model Multiple Patterns?

Transition Table of the "Hidden Markov Chain"

	A_1	A_2	A_3	B_1	B_2	B_3	B_4	C_1			
A1		1									
A2			1								
A3				.5				.5			
B1					1						
B2						1					
B3							1				
B4	1										
C1									1		
C2										1	
C3											1
C4	1										

Using single-layer **HMM** to implement, Transition Table is **O(|state|^2)** with most of entries "zero".

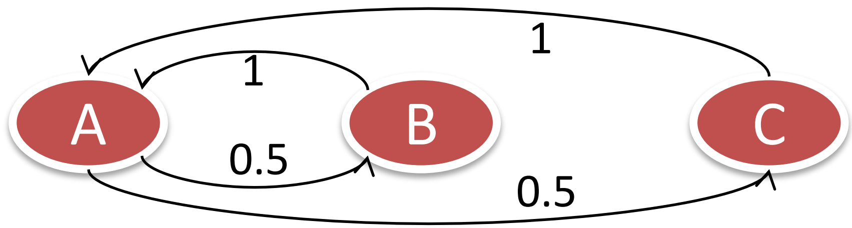
Transition Diagram of "Hidden Markov Chain"



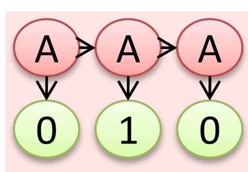
How to Model Multiple Patterns?

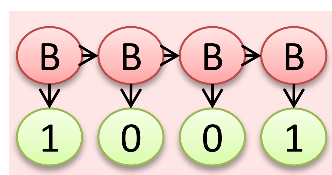
Exploit the Hierarchical Structure,
we can have more compact represent and separate the 2 layers.

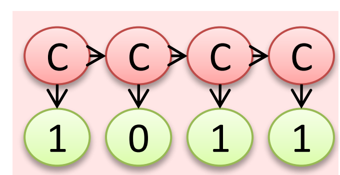
	A	B	C
A		.5	.5
B	1		
C	1		



	A1	A2	A3
A1		1	
A2			1







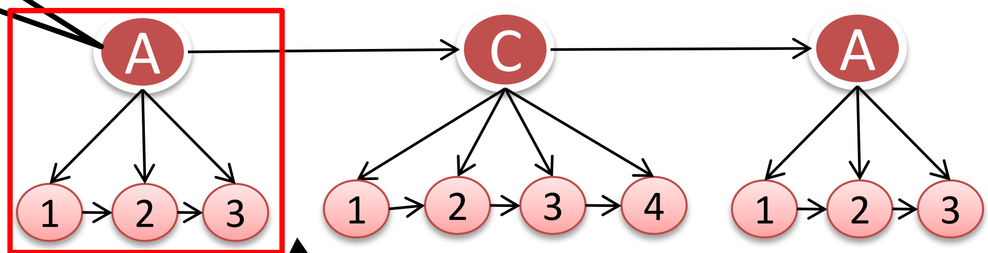
	C1	C2	C3	C4
C1		1		
C2			1	
C3				1

	B1	B2	B3	B4
B1		1		
B2			1	
B3				1

How to Model Multiple Patterns?

How can we know Here will be a “A” pattern **before decoding data** ???

Is this a Legal Graphical Model ?



	A	B	C
A		.5	.5
B	1		
C	1		

0 1 0 1 0 1 1 0 1 0

	A1	A2	A3
A1	[illegible]	[illegible]	[illegible]
A2	[illegible]	[illegible]	[illegible]

	C1	C2	C3	C4
[illegible]	[illegible]	[illegible]		
[illegible]	[illegible]	[illegible]	1	
[illegible]	[illegible]	[illegible]		1

	A1	A2	A3
A1		1	
A2			1

How can we know Here is no dependency **before decoding data** ?

How to encode structural uncertainty into some variables ?

No !!
Note **Structure of a GM** should be fixed in advance.
Uncertainty can only involve the **value of variable**.

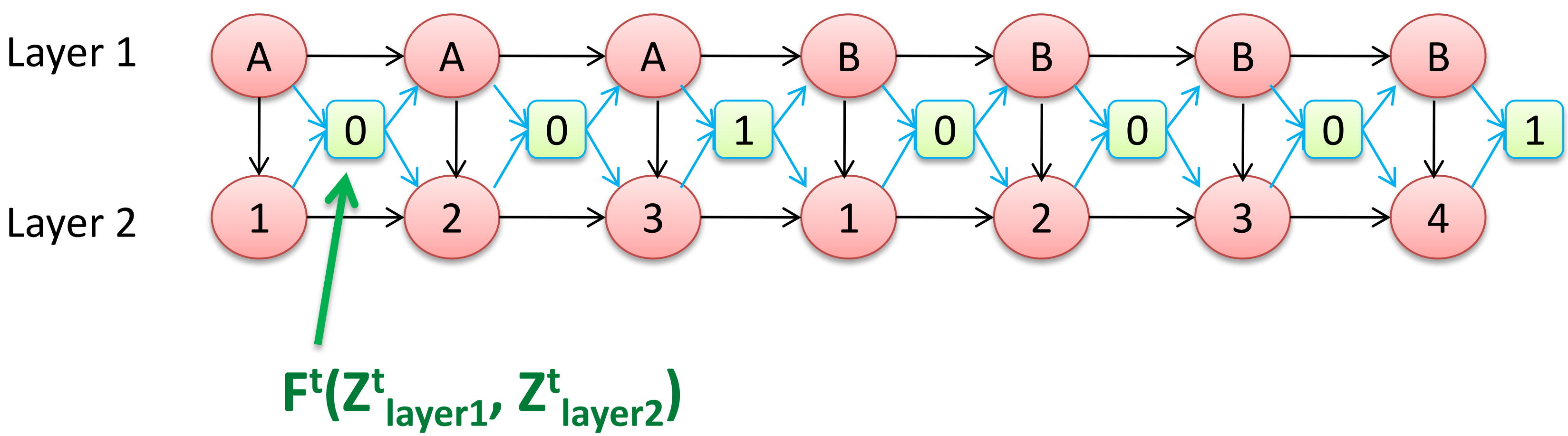
Solution of Hierarchical HMM

(2001, K.P.Murphy)

Introducing "Control Variable" = $F^t(Z^t_{layer1}, Z^t_{layer2}) = \begin{cases} 1, & \text{if } Z^t_{layer2} = \text{Exit State of } Z^t_{layer1} \\ 0, & \text{otherwise} \end{cases}$

$F^t = 0$ ➔ pattern not ending ➔ Layer 1 : $Z^{t+1} = Z^t$
Layer 2 : Transit according to State Machine

$F^t = 1$ ➔ pattern ending ➔ Layer 1 : Draw from $P(Z^{t+1}|Z^t)$
Layer 2 : Draw independently from previous.



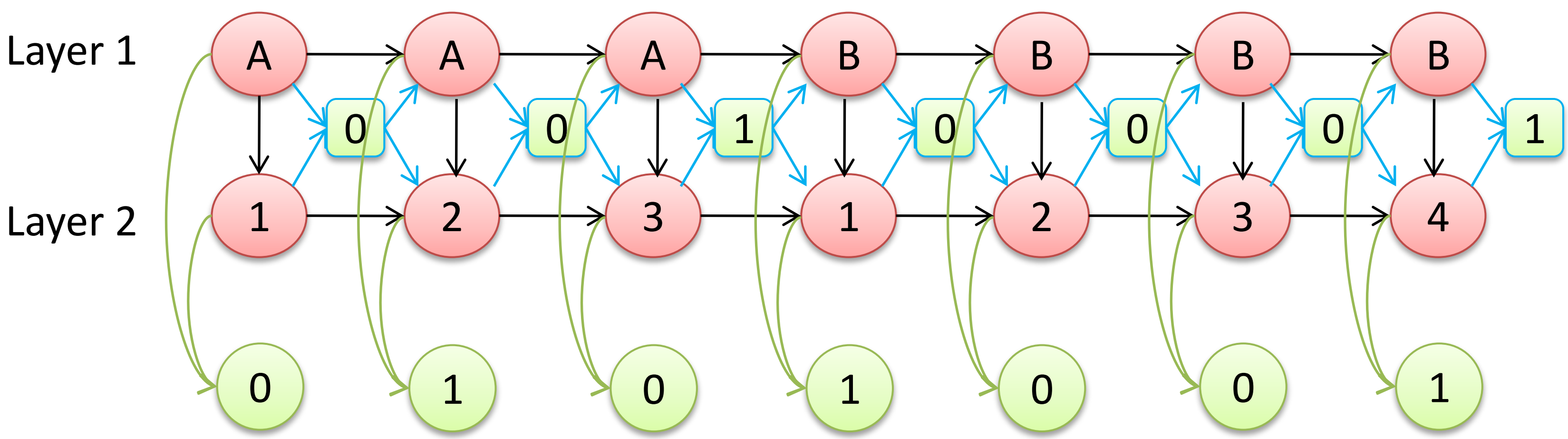
Solution of Hierarchical HMM

(2001, K.P.Murphy)

Introducing "Control Variable" = $F^t(Z^t{}_{layer1}, Z^t{}_{layer2}) = \begin{cases} 1, & if\ \ Z^t{}_{layer2} = Exit\ \ State \text{ of } \ Z^t{}_{layer1} \\ 0, & \text{otherwise} \end{cases}$

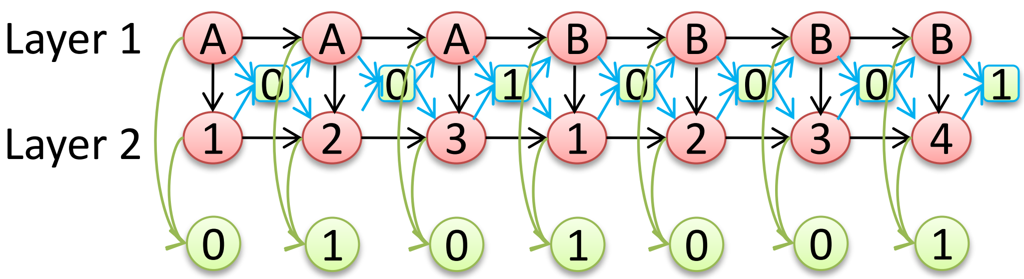
$F^t = 0$ ➔ pattern not ending ➔ Layer 1 : $Z^{t+1} = Z^t$
Layer 2 : Transit according to State Machine

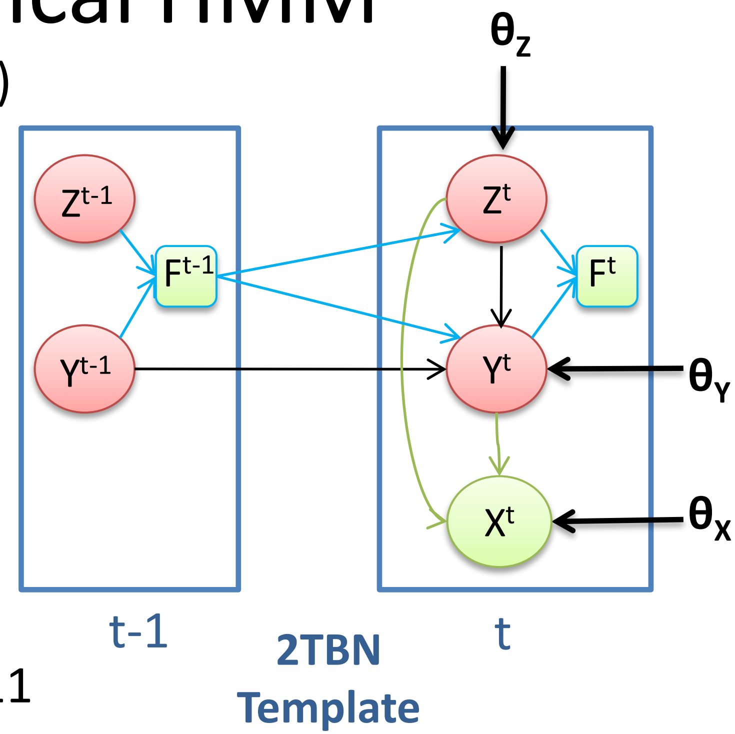
$F^t = 1$ ➔ pattern ending ➔ Layer 1 : Draw from $P(Z^{t+1} | Z^t)$
Layer 2 : Draw independently from previous.



Solution of Hierarchical HMM

(2001, K.P.Murphy)





Explain Data: 010 1011 010 1001 010 1001 010 1011

A C A B A B A C

Likelihood = P(A)*P(C|A) …… *P(C|A)

*P(transition in A|A)*P(transition in C|C) ……*P(transition in C|C)

*P(0 |A,1) P(1|A,2)P(0|A,3) * P(1|C,1) P(0|C,2)P(1|C,3) P(1|C,4)……

= P(A)*P(C|A) …… *P(C|A) = $(0.5)^4$

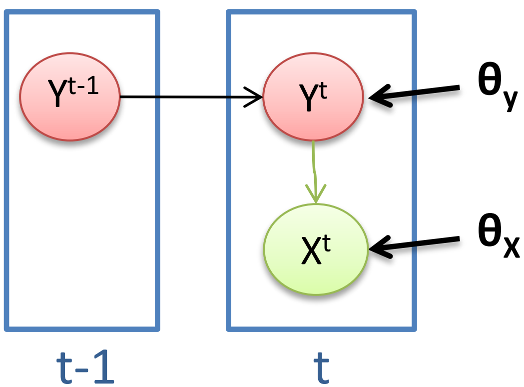
Deterministic Behavior

Much better than Naïve Model

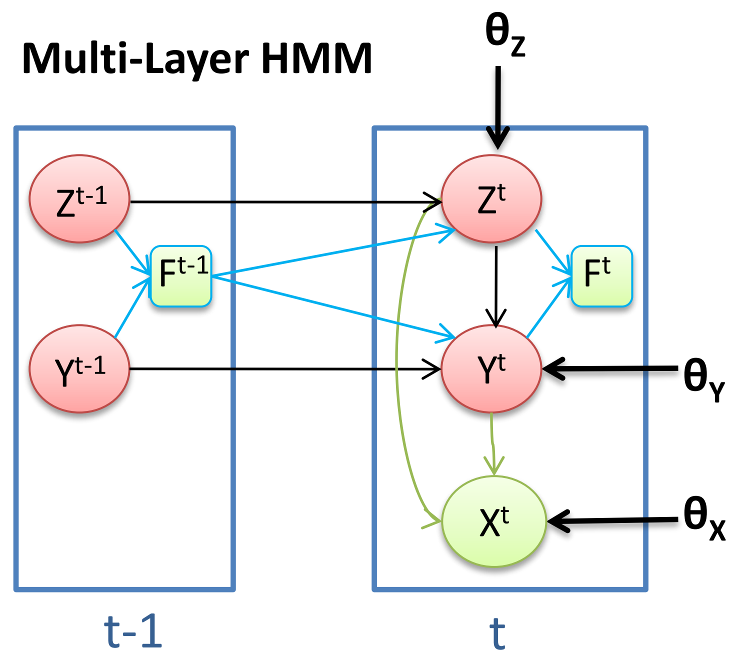
Comparison of Single & Multi-Layer HMM

Note: Both method can yield the same likelihood to explain data.

Single-Layer HMM



Multi-Layer HMM



Assume there are **K patterns**, where each pattern's **state machine has D states.**

Size of Transition Table:

$\theta_Y \rightarrow O((K*D)^2)$

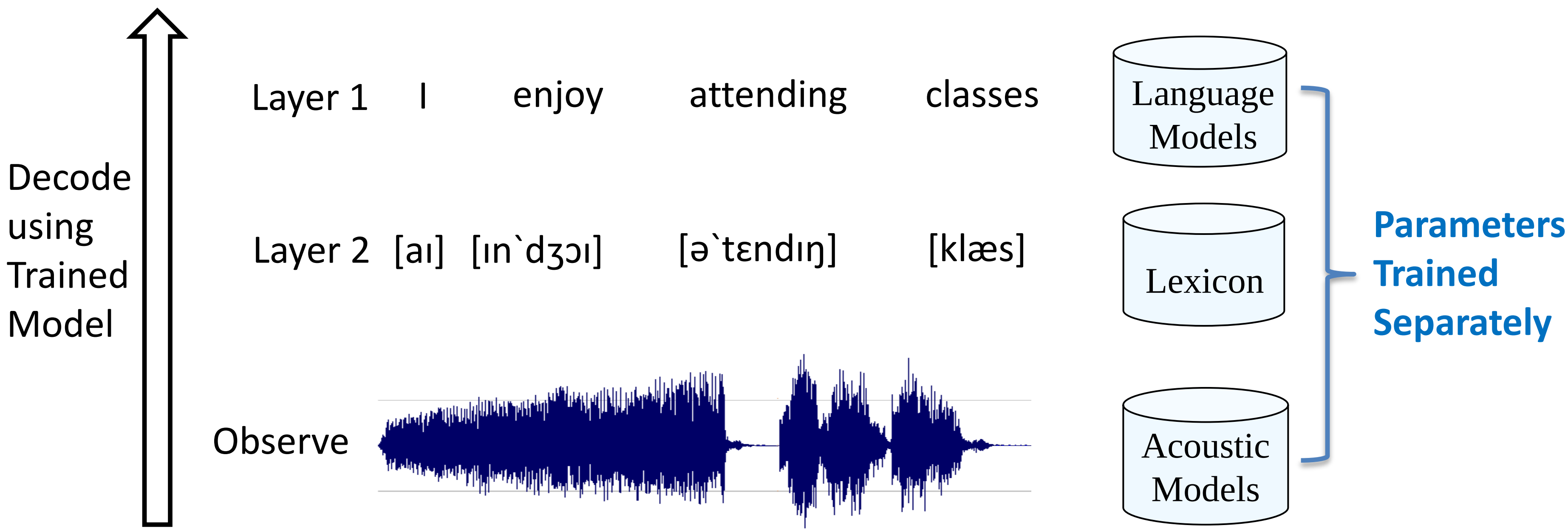
$\theta_Z \rightarrow O(K^2)$

$\theta_Y \rightarrow O(D^2)$

Application in Speech Recognition

(Speech Signal Processing 2010 Fall , 李琳山)

Hierarchical HMM is the foundation of Speech Recognition.

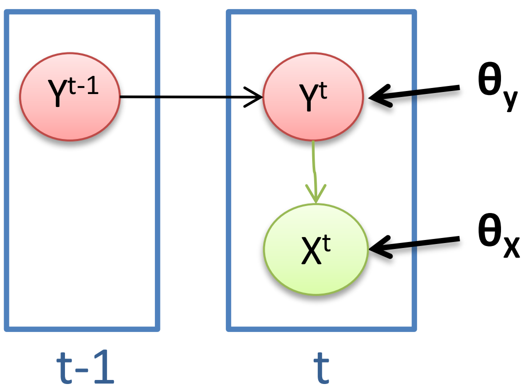


When **decoding low** layer**, high** layer's model is taken into consideration.
(ex. Some "**incorrect" Pronunciation** can be realized with knowledge of high-layer **Language Model**.)

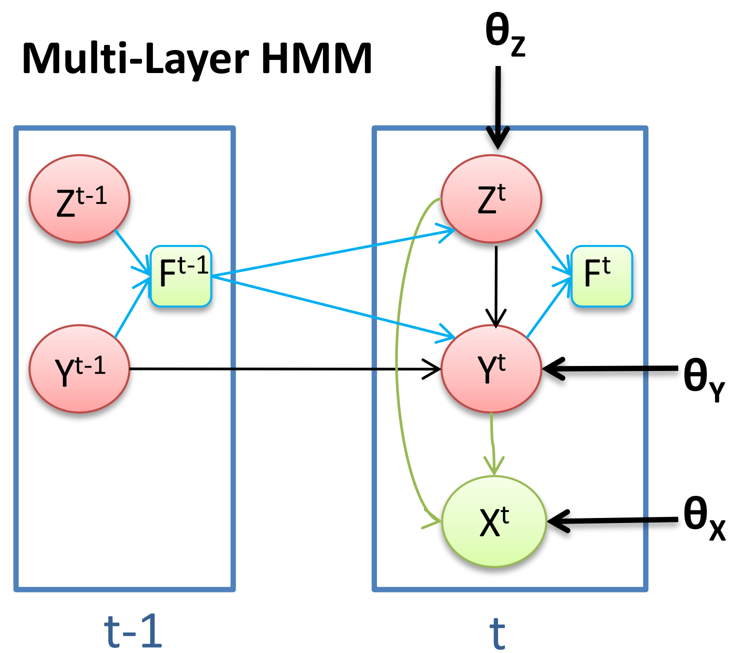
Comparison of Single & Multi-Layer HMM

Note: Both method can yield the same likelihood to explain data.

Single-Layer HMM



Multi-Layer HMM



1. We may want training **different layers separately**.

   (ex. high layer: Language model ;   low layer : Lexical Model)

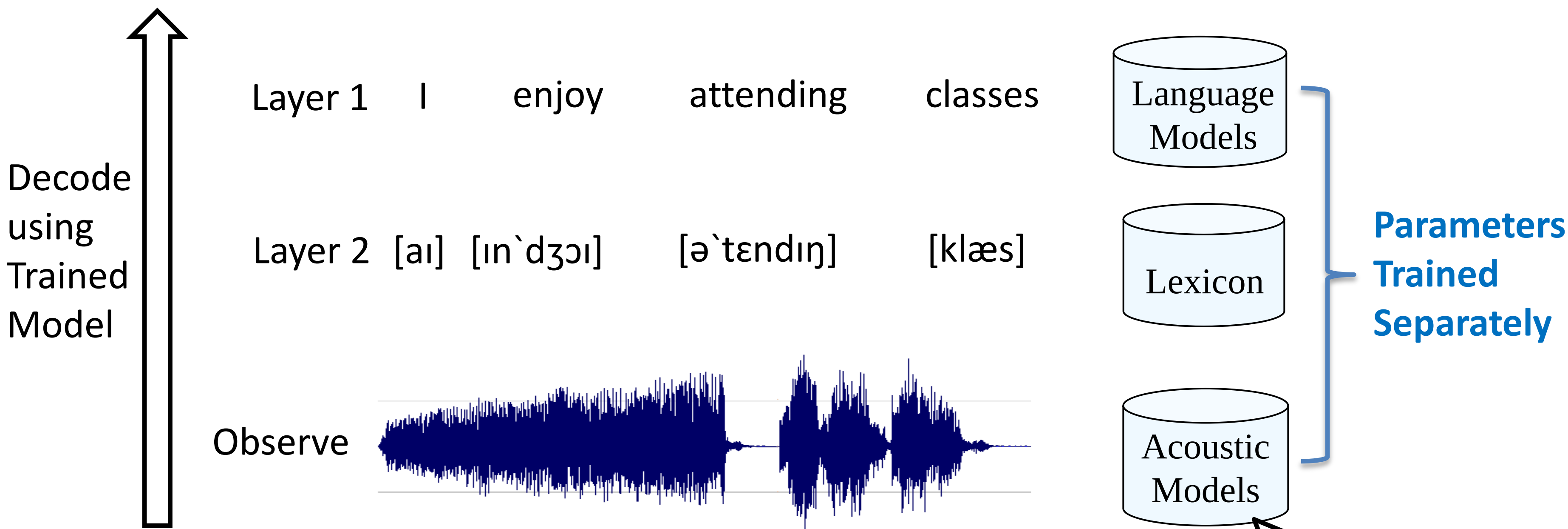
2. We may want introduce **prior on only some layers**. (see next)

See "2001, K.P.Murphy" for more advantages of this approach.

Application in Speech Recognition

(Speech Signal Processing 2010 Fall , 李琳山)

Hierarchical HMM is the foundation of Speech Recognition.



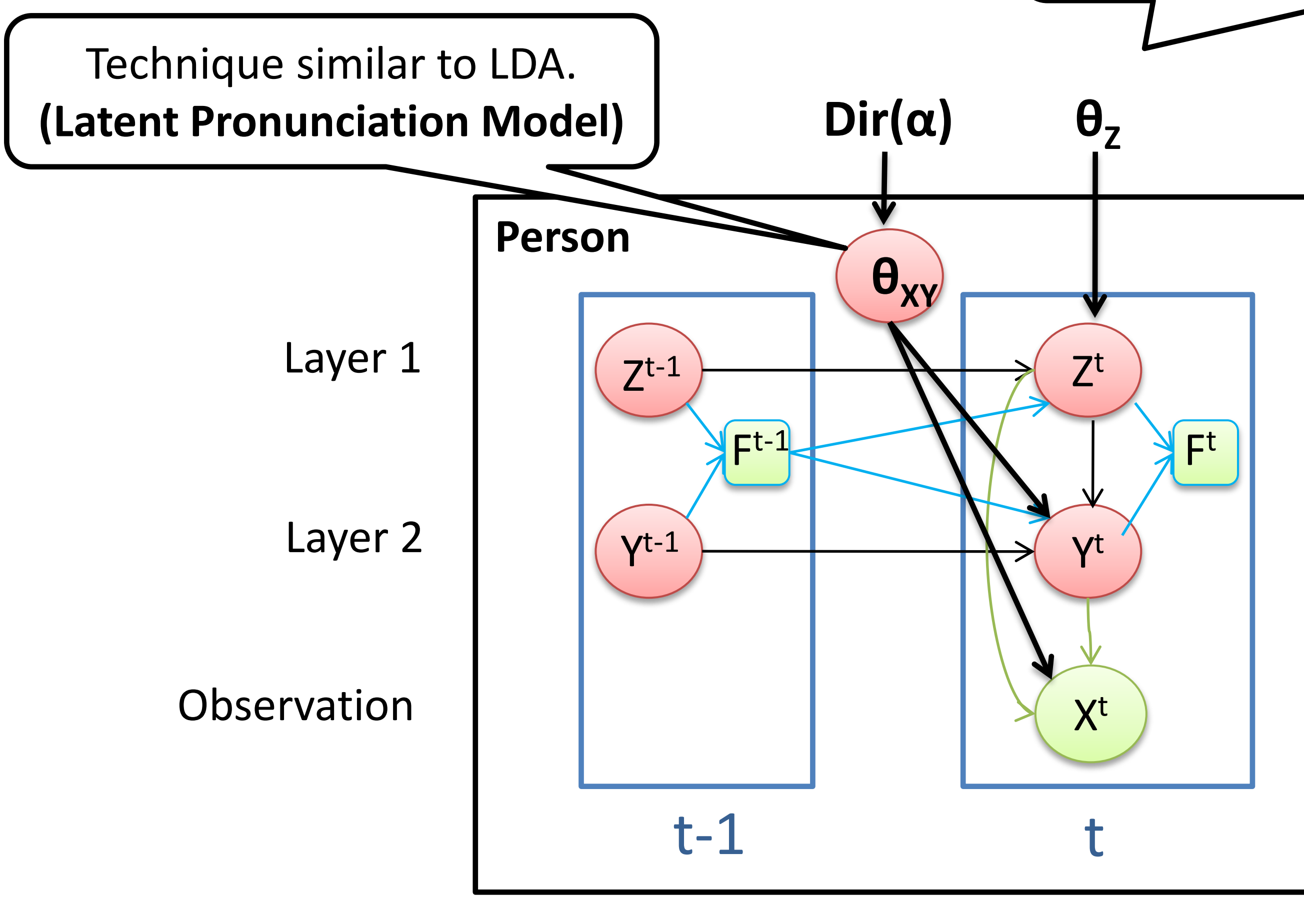
When **decoding low** layer**, high** layer's model is taken into consideration.
(ex. Some "**incorrect" Pronunciation** can be realized with knowledge of high-layer **Language Model**.)

How to handle different persons' pronunciation ??

Application in Speech Recognition

Transition Parameters : θ_Y , θ_Z

How to handle different persons' pronunciation ??



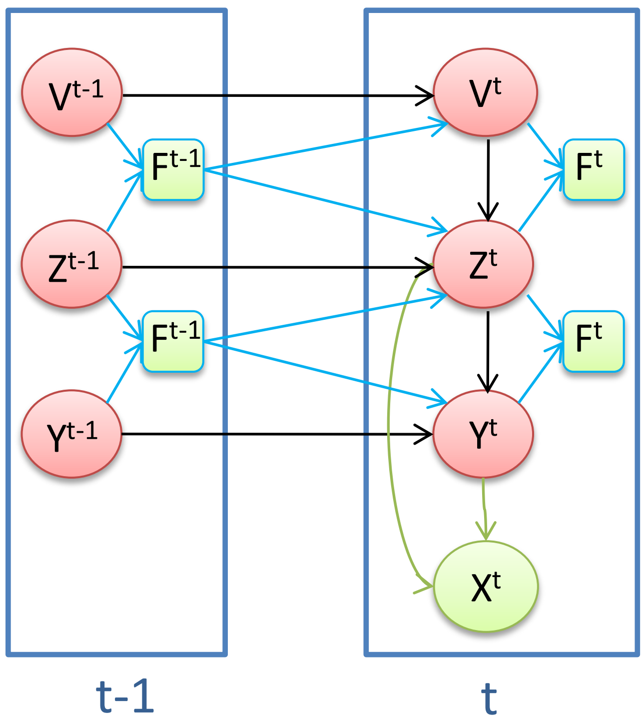
For every person, Introduce a **Pronunciation Pattern Variable θ_{XY} .**

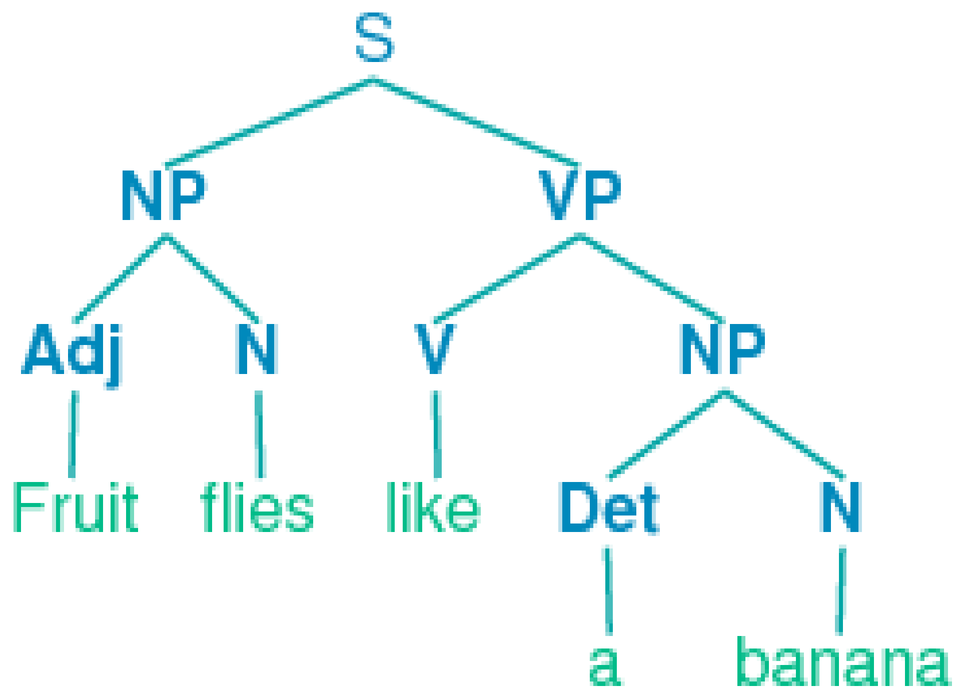
In the beginning, use "**prior of θ_{XY}** " to decode speech of a person.

As # of samples **from the person** grows, **new θ_{XY}** can be learned/inferred to adapt to that person.

Application : Natural Language Understanding

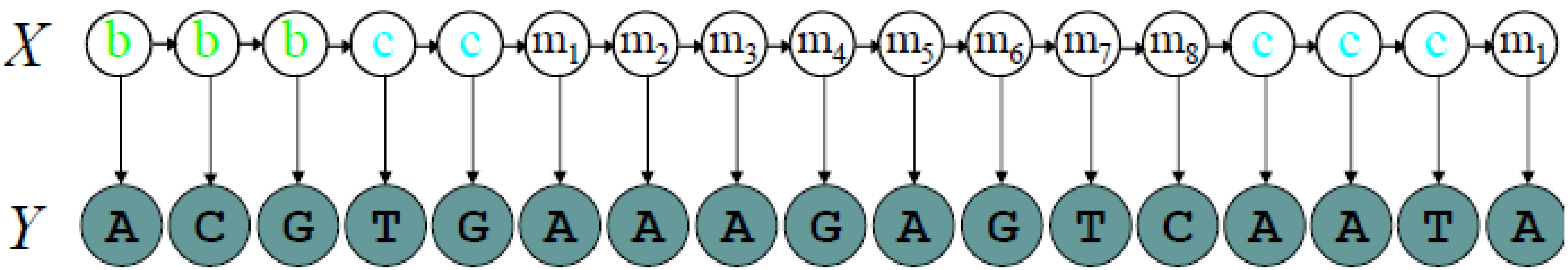
Multi-Layer HMM

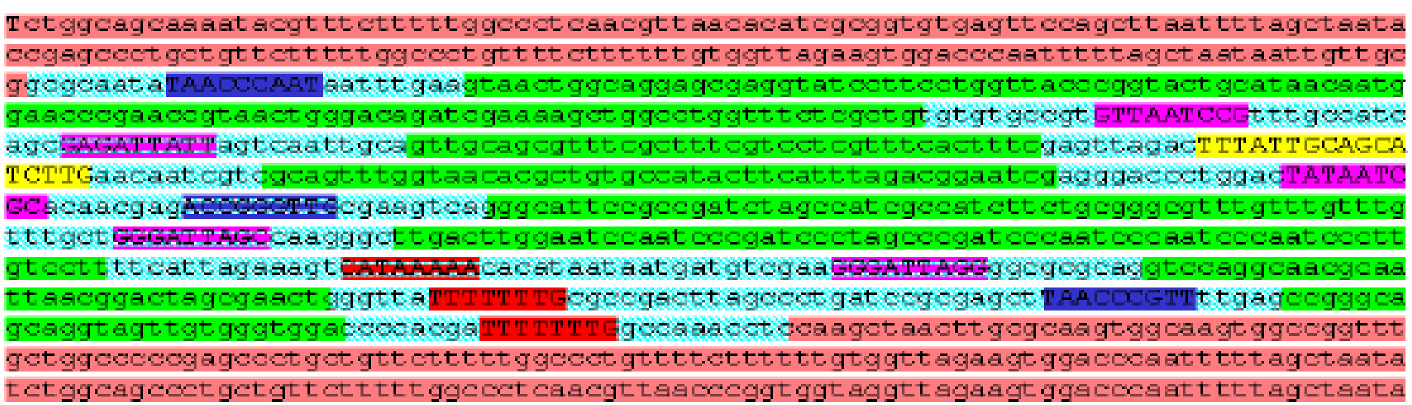




Gene Decoding

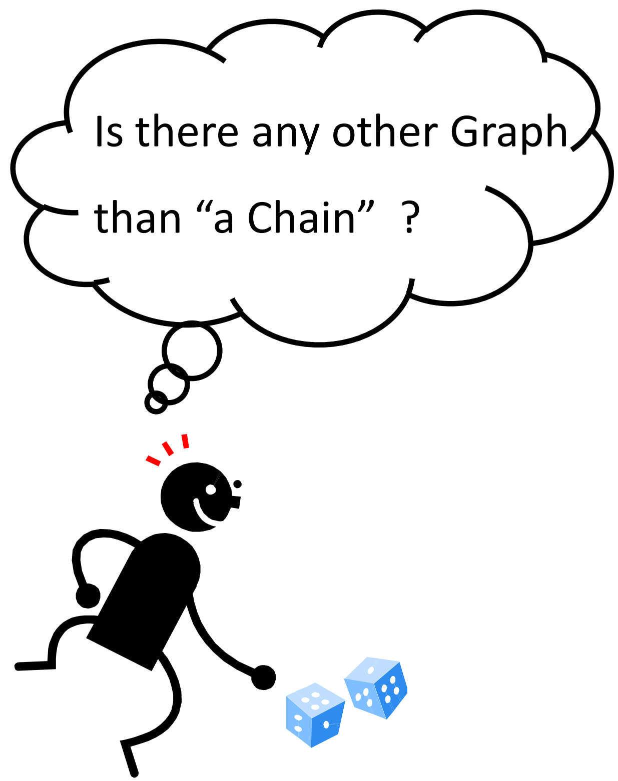
Pattern Mining / Pattern Recognition.





Overview

- What's Probabilistic Graphical Model for ?
- Tasks in Graphical Model:
  - Modeling (Simple Probability Model)
  - Learning (MLE, MAP, Bayesian)
  - Inference (Bayes Rule ??)
- Examples
  - Topic Model (EM algorithm)
  - Hidden Markov Model (Variable Elimination)
  - Markov Random Field



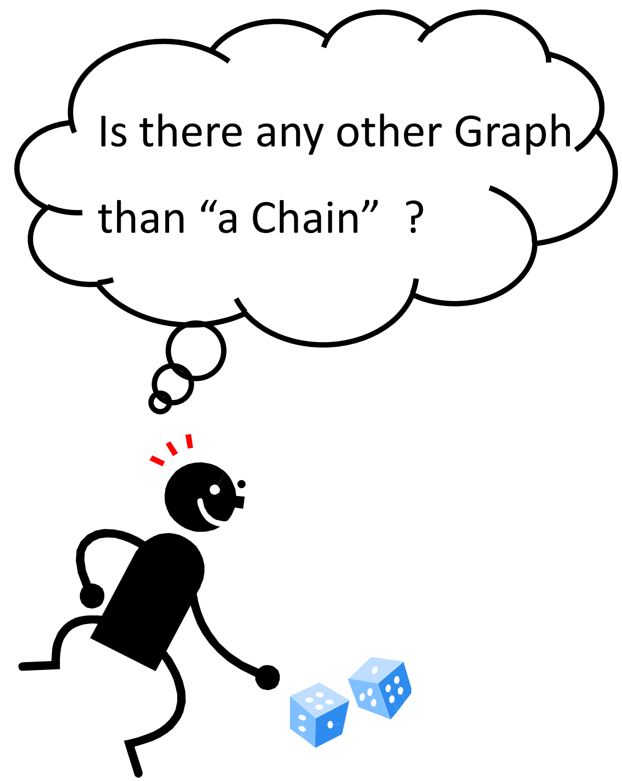
Overview

- What's Probabilistic Graphical Model for ?

- Tasks in Graphical Model:
  - Modeling (Simple Probability Model)
  - Learning (MLE, MAP, Bayesian)
  - Inference (Bayes Rule ??)

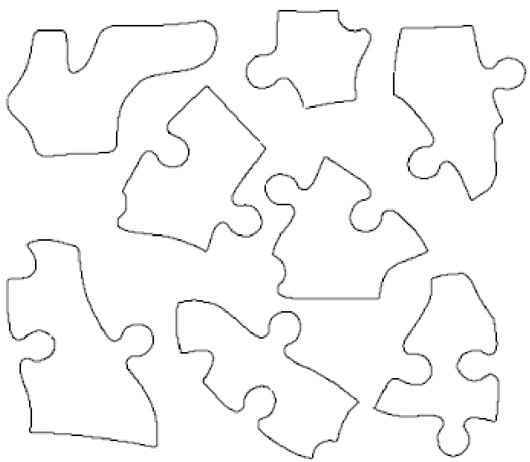
- Examples
  - Topic Model (EM algorithm)
  - Hidden Markov Model (Variable Elimination)
  - **Markov Random Field**

    (All of previous models are
    special cases of **Bayesian Network**.)

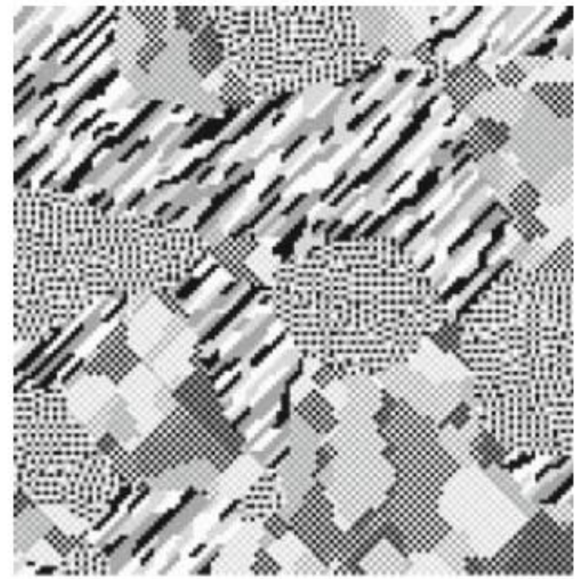


Given Domain Problem : Modeling Spatial Pattern

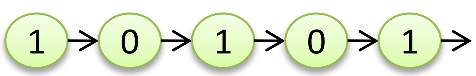
Distribution of Shape



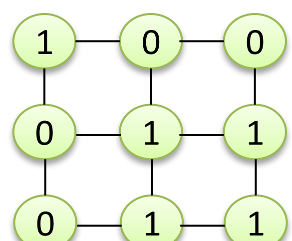
Distribution of Texture

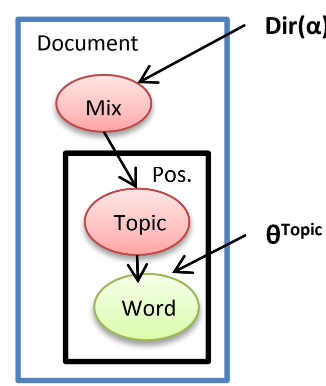


In sequential data, we model “**before as cause”** and “**after as effect** ”.



In spatial data, **who is the “cause” ?**





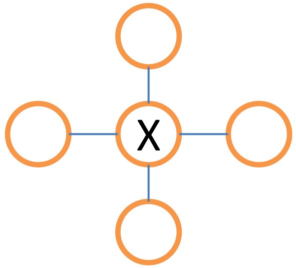
Modeling with Markov Random Field

Global Structure:

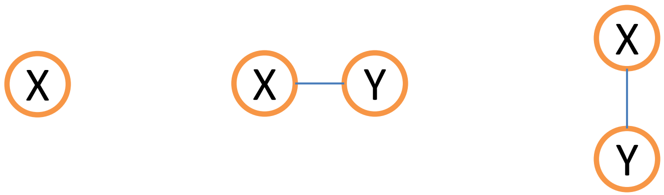
Define types of Node (variable) :

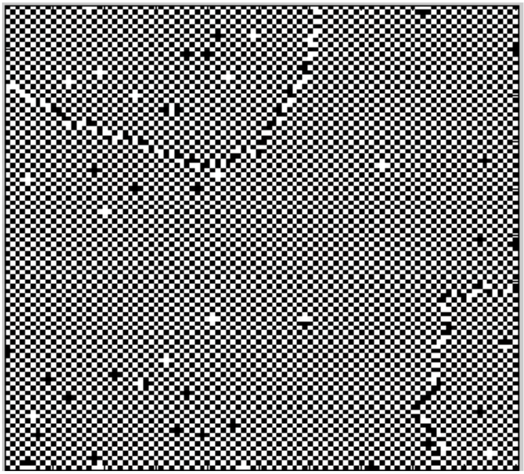
$$X \in Val(X) = \{0,1\}$$

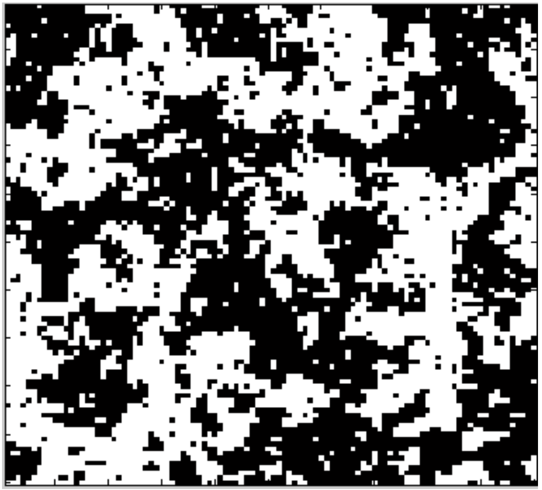
Without "cause & effect", MRF defines a "**Neighborhood**" a variable will interact with :



Along with Neighborhood, types of "**Cliques**" are defined :







Modeling with Markov Random Field

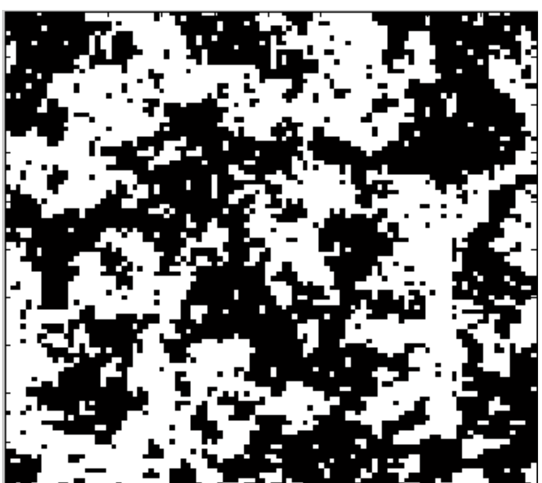
Local Structure :

Define **Potential function ϕ_C(variables in C)** for all **types of clique C** we care about. The Gibbs Distribution of the MRF is given by:

$$P(X) = \frac{1}{Z}\prod_{C \in clique} \phi(X_C), \text{ where } Z = \sum_{X \in Val(X)} P(X) \;\; \textit{is for normalize.}$$

ϕ_1(X,Y)	0	1
0	2	1
1	1	2

ϕ_2(X,Y)	0	1
0	2	1
1	1	2



ϕ_1(X,Y)	0	1
0	1	4
1	4	1

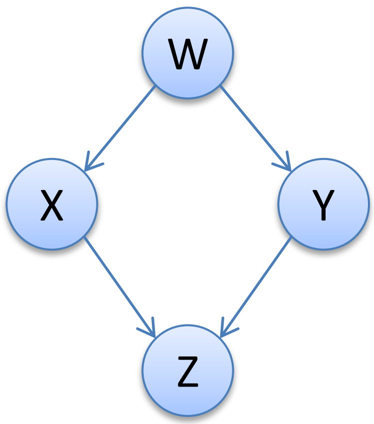
ϕ_2(X,Y)	0	1
0	1	4
1	4	1

ϕ_1(X,Y)	0	1
0	1	7
1	7	1

ϕ_2(X,Y)	0	1
0	7	1
1	1	7

How MRF Generate Samples

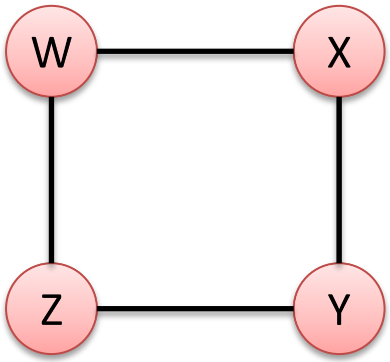
How BN generate samples ?



P(X|Pa(X)) is available given Pa(X).

➔ Sampling follows Topological Order.

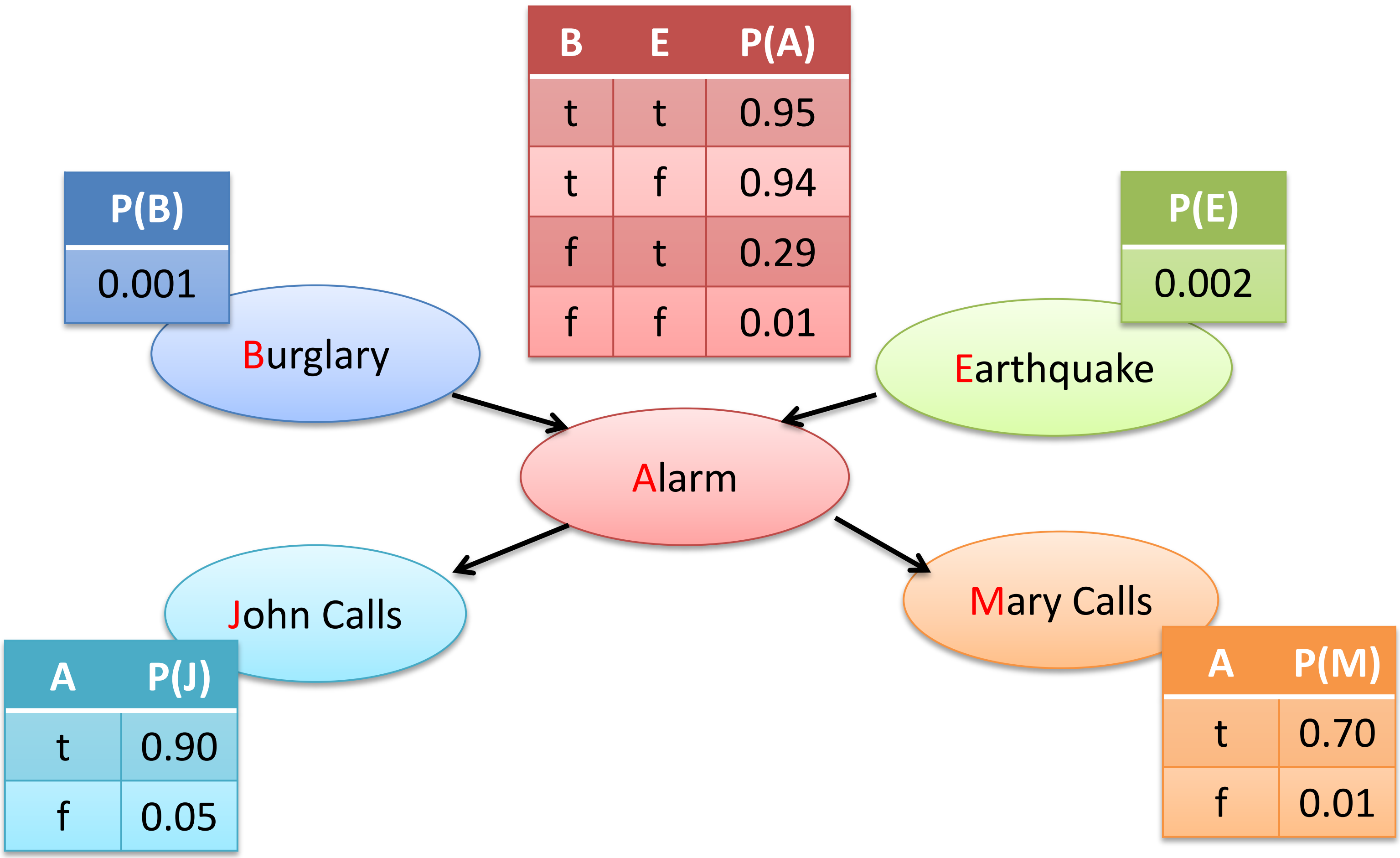
How MRF generate samples ?



P(X|N(X)) can be derived from Bayes Rule.

But how to find an order ?

How BN Generate Samples ?



P(B)
0.001

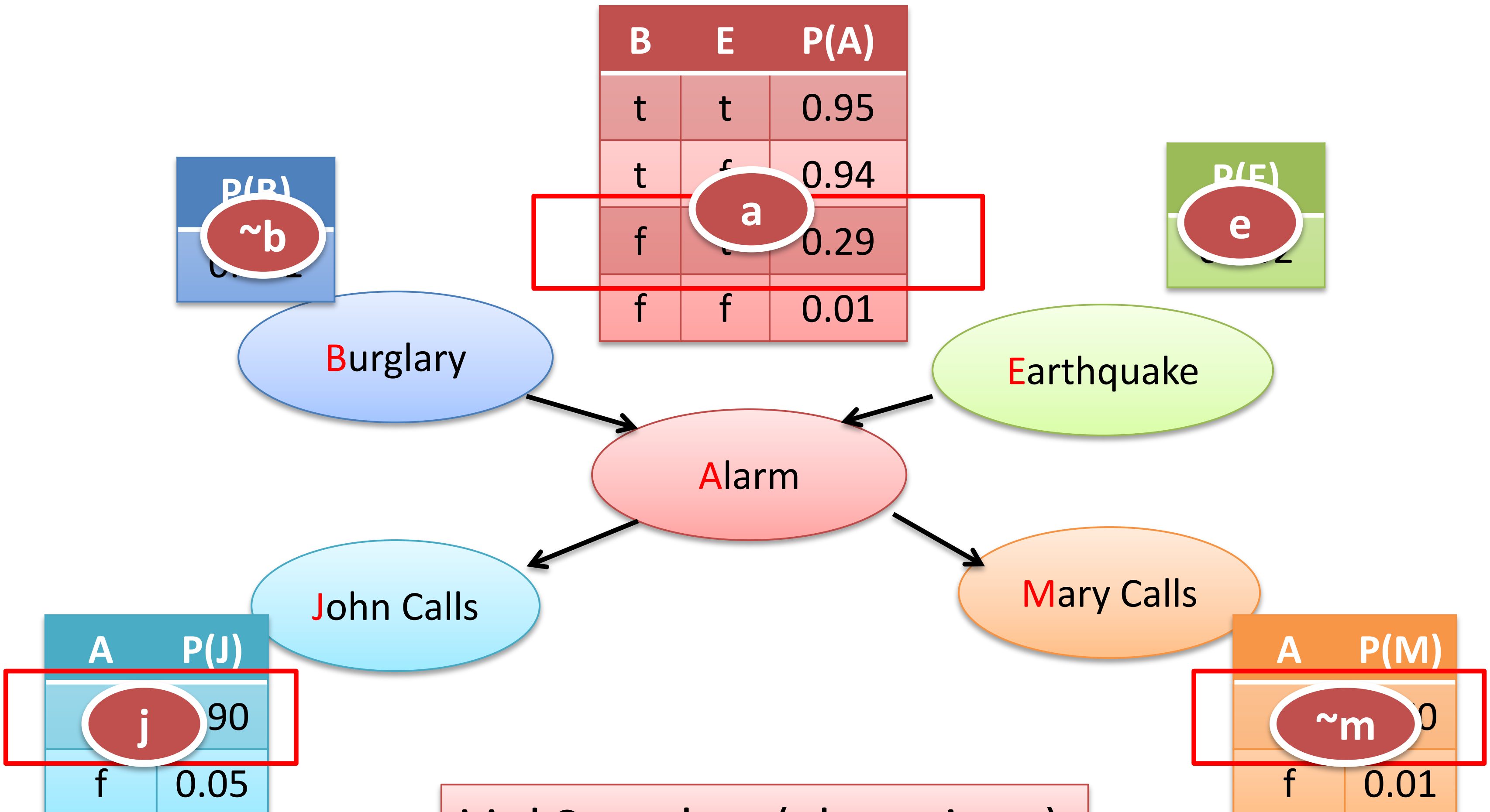
P(E)
0.002

B	E	P(A)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.01

A	P(J)
t	0.90
f	0.05

A	P(M)
t	0.70
f	0.01

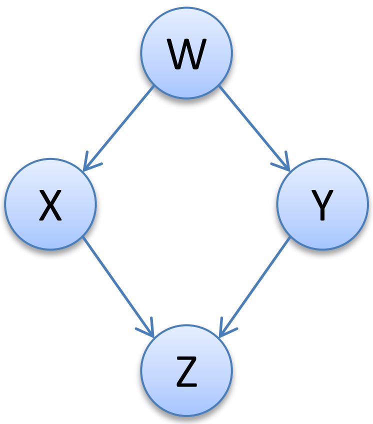
How BN Generate Samples ?



i.i.d Sample : (~b,e,a,j,~m)

How MRF Generate Samples

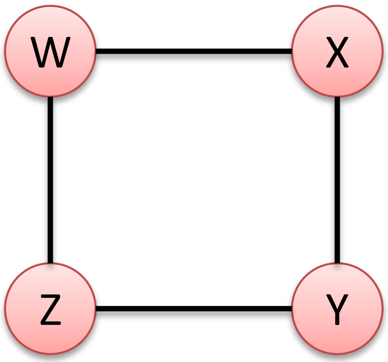
How BN generate samples ?



P(X|Pa(X)) is available given Pa(X).

➔ Sampling follows Topological Order.

How MRF generate samples ?



P(X|N(X)) can be derived from Bayes Rule.

But how to find an order ?

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

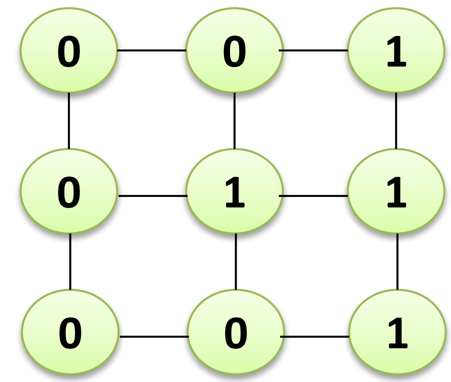
    for every variable X

        2. Draw X_t from P(X | $N(X)_{t-1}$).

    end

end

t=1



$$P(X=1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X=1, Y)}{\prod_{Y \in N(X)} \phi(X=1, Y) + \prod_{Y \in N(X)} \phi(X=0, Y)}$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.

for $t = 1 \sim M$

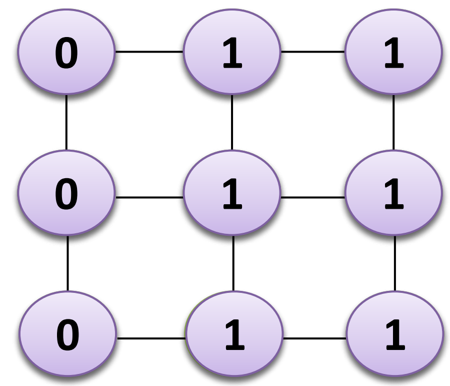
for every variable X

2. Draw X_t from $P(X \mid N(X)_{t-1})$.

end

end

t=2



$$P(X = 1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X = 1, Y)}{\prod_{Y \in N(X)} \phi(X = 1, Y) + \prod_{Y \in N(X)} \phi(X = 0, Y)}$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

For the central node:

$$P(X = 1 \mid N(X)) = \frac{1*9*9*1}{1*9*9*1 + 5*1*1*5} = 0.76$$

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

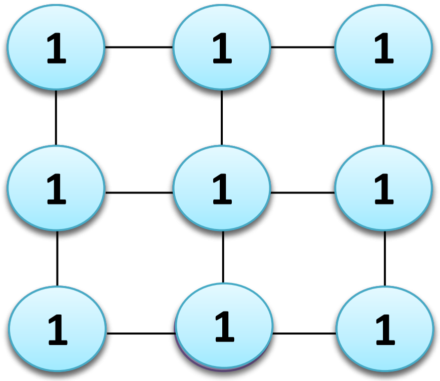
  for every variable X

    2. Draw X_t from P(X | N(X)$_{t-1}$).

  end

end

t=3



$$P(X=1|N(X)) = \frac{\prod_{Y\in N(X)} \phi(X=1,Y)}{\prod_{Y\in N(X)} \phi(X=1,Y) + \prod_{Y\in N(X)} \phi(X=0,Y)}$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

For the central node:

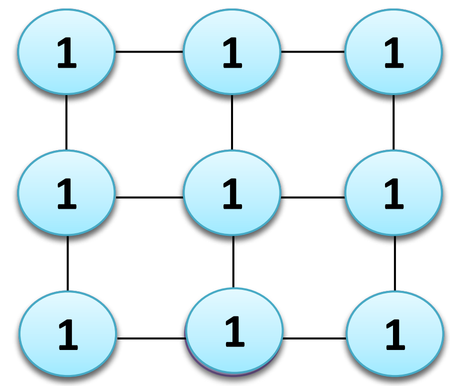
$$P(X=1|N(X)) = \frac{9*9*9*9}{9*9*9*9+1*1*1*1} = 0.99$$

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.
for $t = 1 \sim M$
    for every variable X
        2. Draw X_t from $P(X \mid N(X)_{t-1})$.
    end
end

t=3



When M is large enough, $X^{(M)}$ follows stationary dist. :

$$\pi_T(X) = P(X) = \frac{1}{Z}\prod_C \phi(X_C)$$

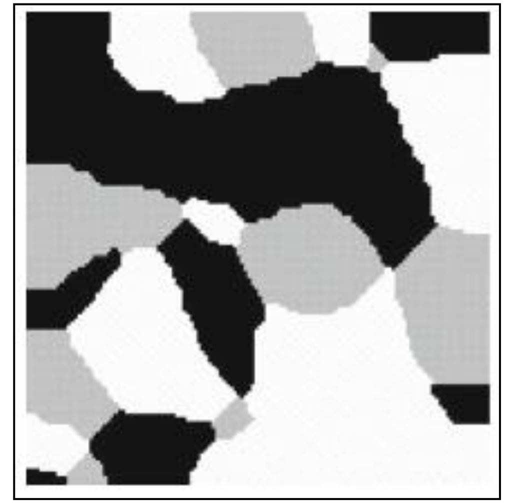
(Regularity: All entries in the Potential are positive.)

ϕ(X,Y)	0	1
0	5	1
1	1	9

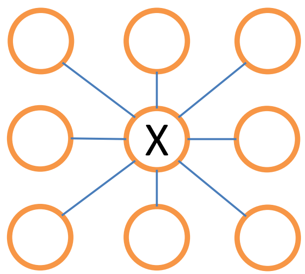
Modeling with Markov Random Field

To model texture with **multi-label**, we have: (ex.)

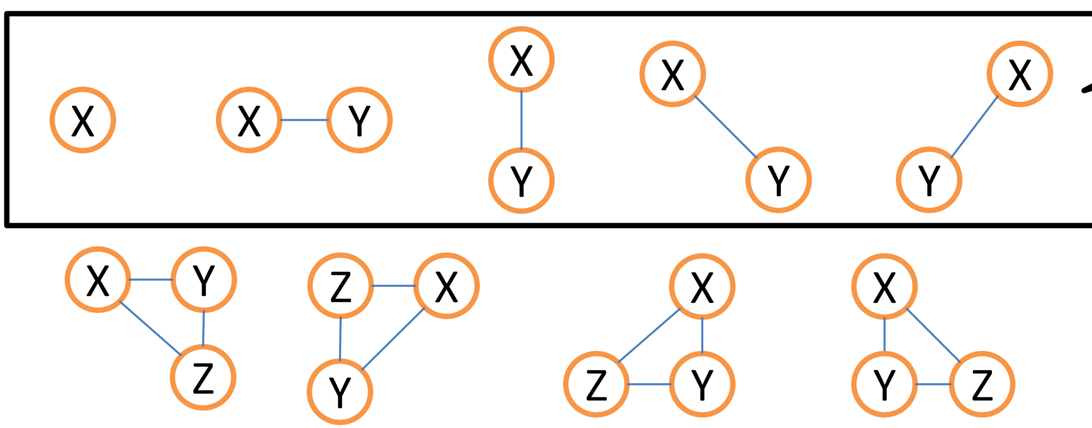
$$X \in Val(X) = \{0,1,2\}$$



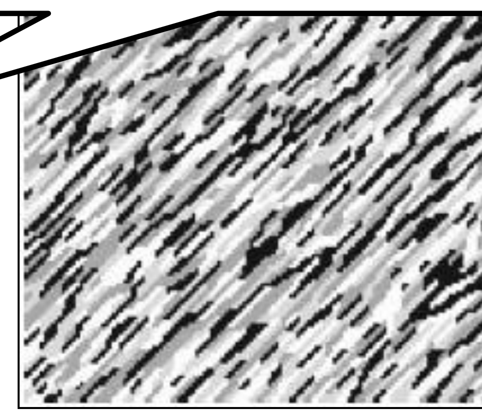
For **texture**, we extend **Neighborhood** to be:



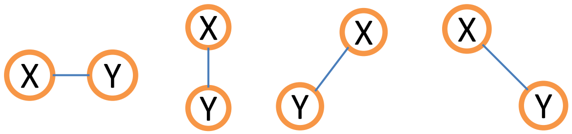
With Neighborhood, Possible "**Cliques**" are defined :



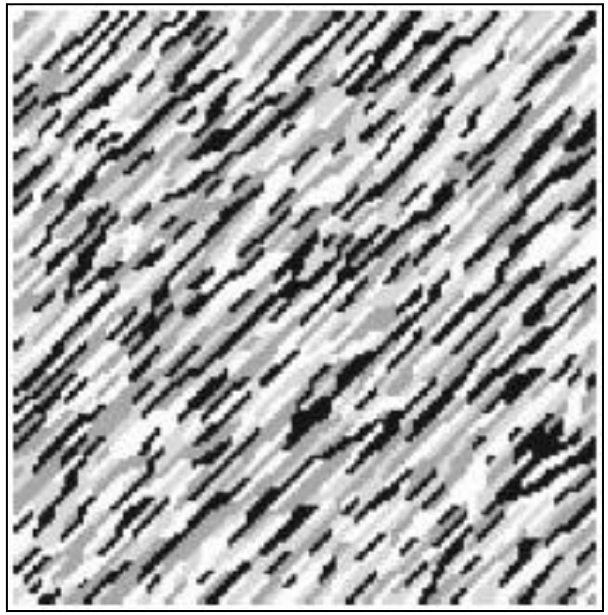
We may use only part of them.

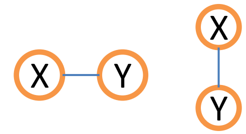


Modeling with Markov Random Field



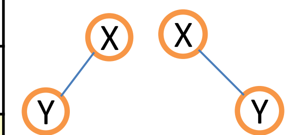
ϕ(X,Y)	0	1	2
0	7	1	1
1	1	7	1
2	1	1	7

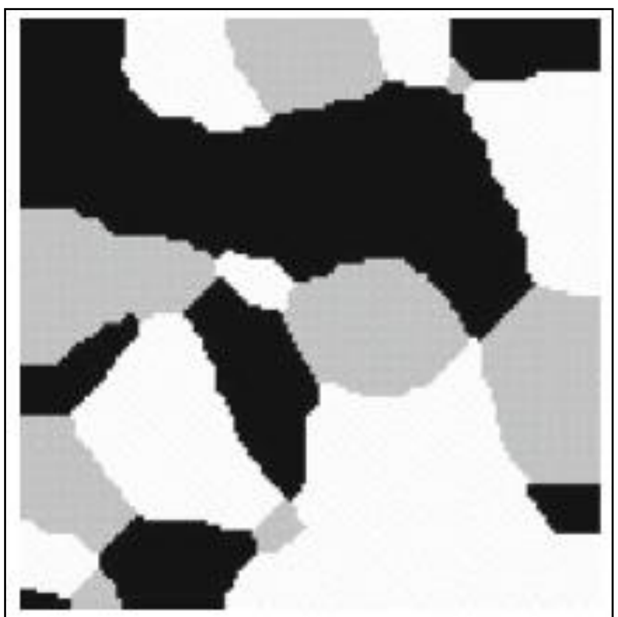




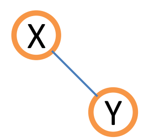
ϕ(X,Y)	0	1	2
0	1	7	7
1	7	1	7
2	7	7	1

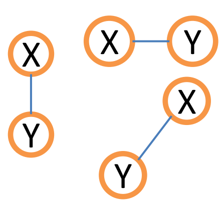
ϕ(X,Y)	0	1	2
0	9	1	1
1	1	9	1
2	1	1	9



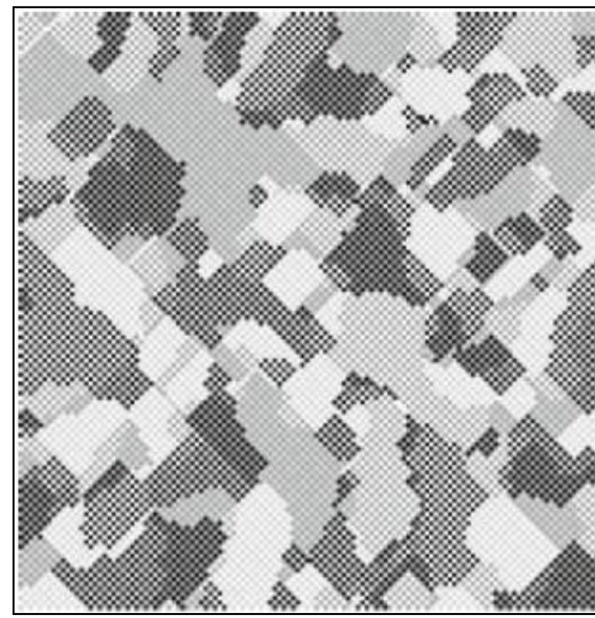


ϕ(X,Y)	0	1	2
0	1	7	7
1	7	1	7
2	7	7	1

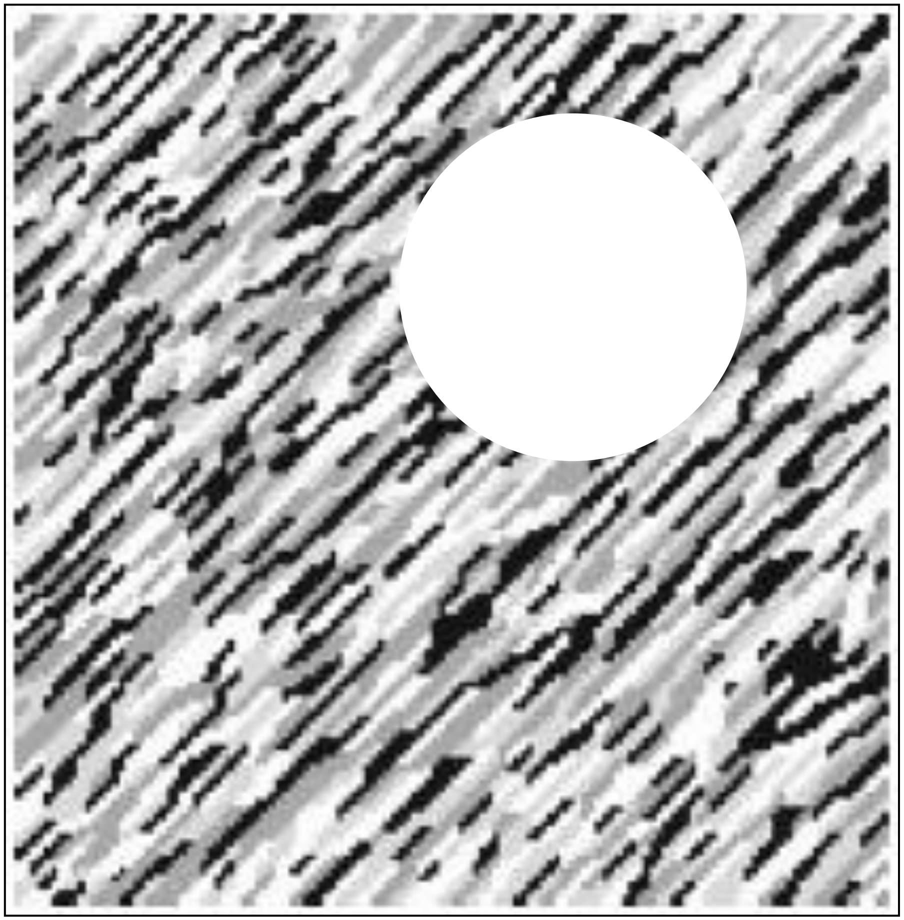




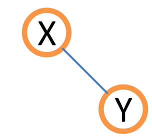
ϕ(X,Y)	0	1	2
0	7	1	1
1	1	7	1
2	1	1	7

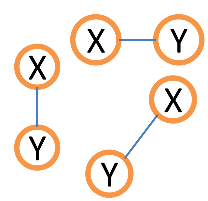


Infer The Lost Segment



ϕ(X,Y)	0	1	2
0	1	7	7
1	7	1	7
2	7	7	1

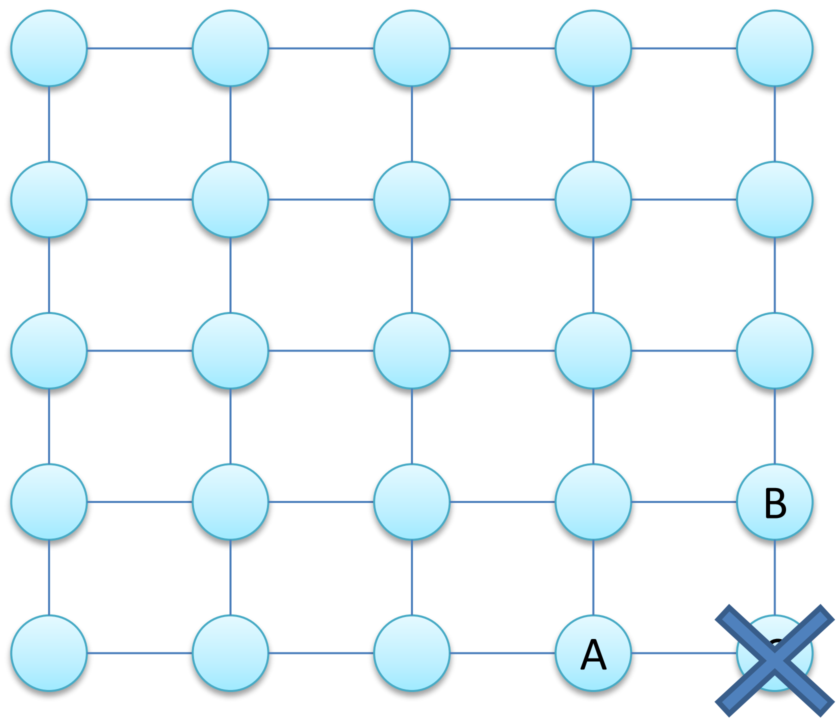




ϕ(X,Y)	0	1	2
0	7	1	1
1	1	7	1
2	1	1	7

Some Model are Intractable for Exact Inference

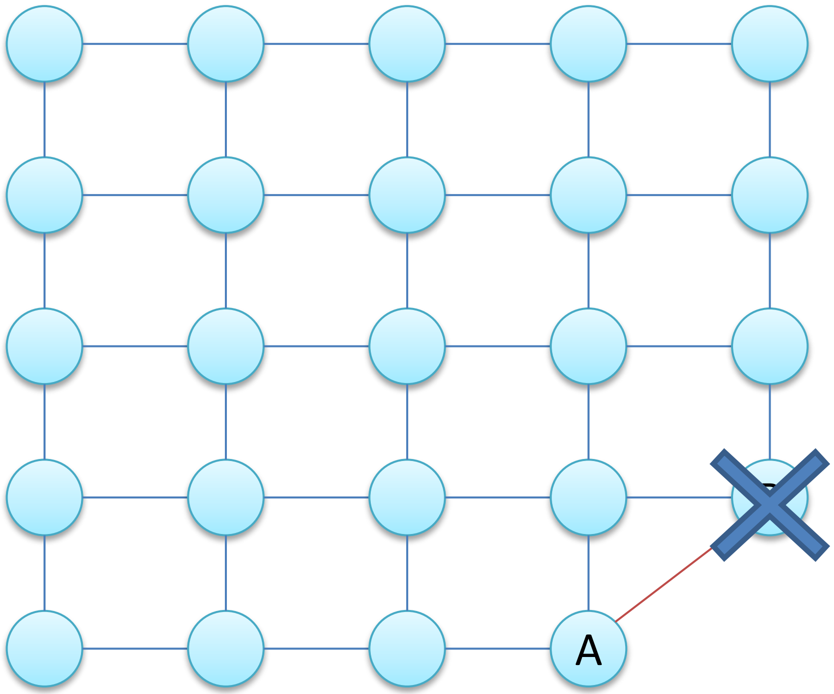
Example: A Grid MRF



$$\operatorname*{argmax}_{C} \; P(A,B)*P(B,C) = F(A,B)$$

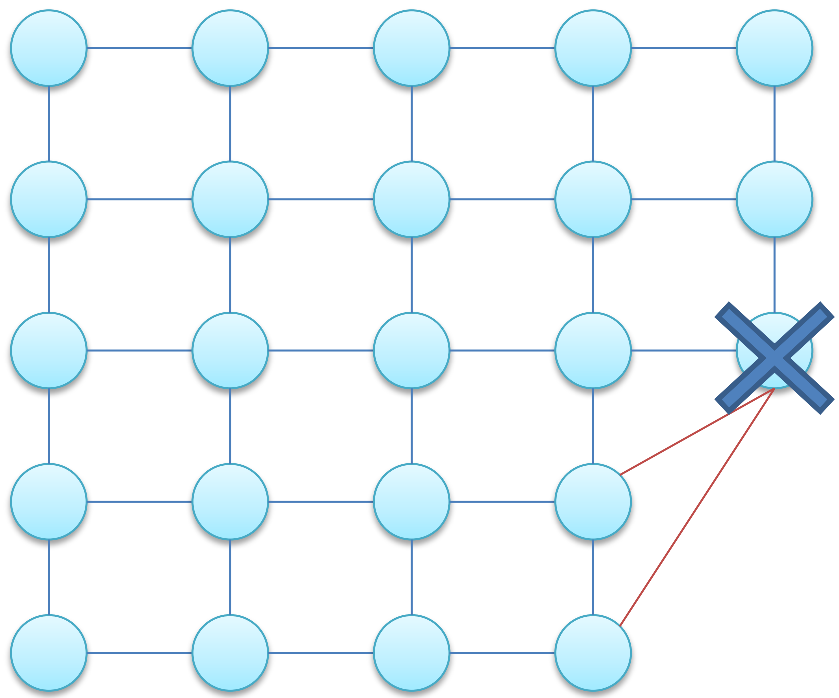
Some Model are Intractable for Exact Inference

Example: A Grid MRF



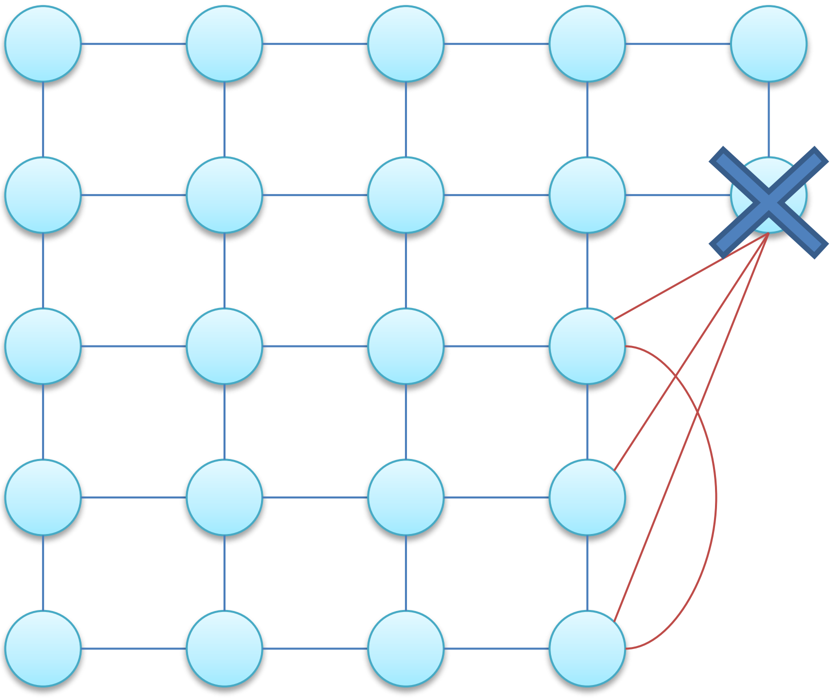
Some Model are Intractable for Exact Inference

Example: A Grid MRF



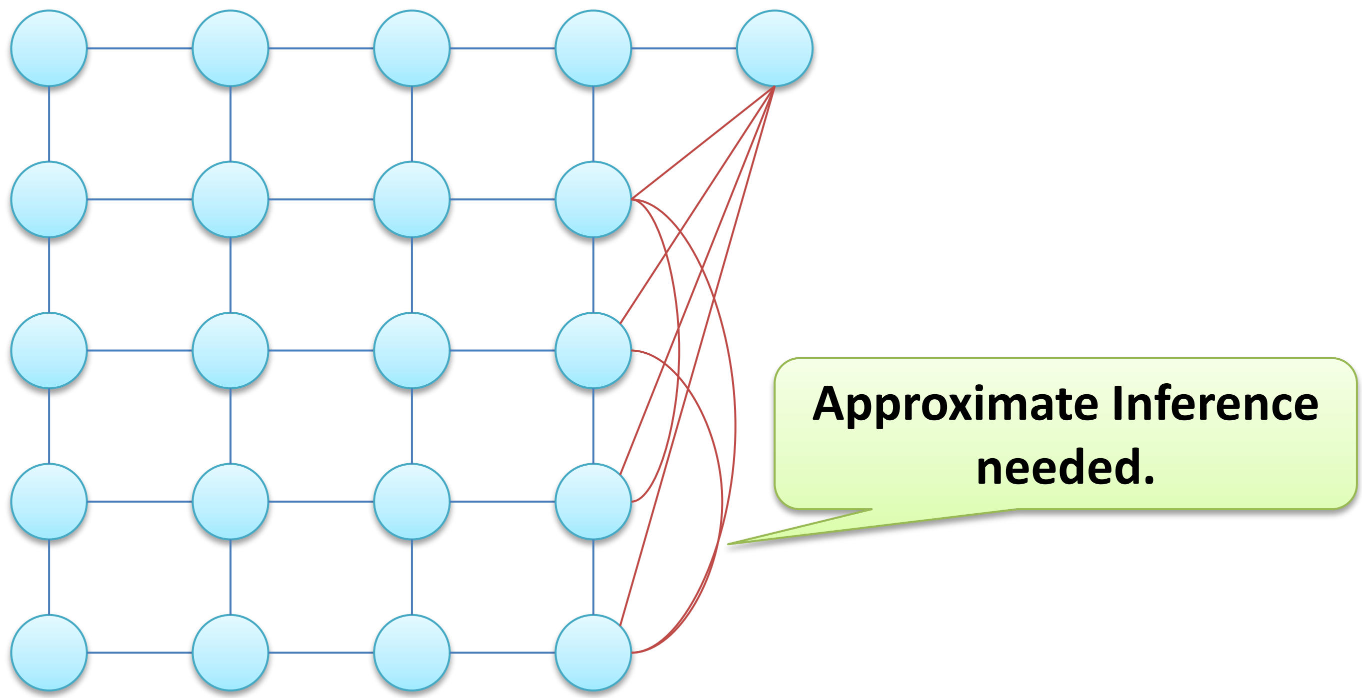
Some Model are Intractable for Exact Inference

Example: A Grid MRF



Some Model are Intractable for Exact Inference

Example: A Grid MRF



Generally, we will have clique of "size N" for a N*N grid, which is indeed intractable.

Gibbs Sampling for Inference

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

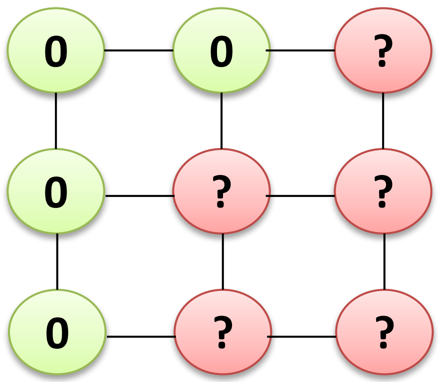
  for every variable X

    2. Draw X_t from $P(X \mid N(X)_{t-1})$.

  end

end

t=1



$$P(X = 1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X = 1, Y)}{\prod_{Y \in N(X)} \phi(X = 1, Y) + \prod_{Y \in N(X)} \phi(X = 0, Y)}$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

Gibbs Sampling for Inference

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

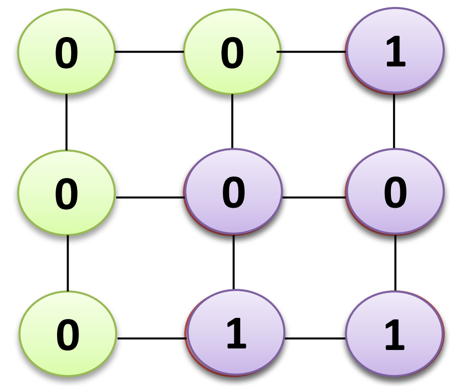
  for every variable X

   2. Draw X_t from $P(X \mid N(X)_{t-1})$.

  end

end

t=2



$$P(X=1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X=1,Y)}{\prod_{Y \in N(X)} \phi(X=1,Y) + \prod_{Y \in N(X)} \phi(X=0,Y)}$$

For the central node:

$$P(X=1 \mid N(X)) = \frac{1*9*1*1}{1*9*1*1+5*1*5*5} = 0.06$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

Gibbs Sampling for Inference

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

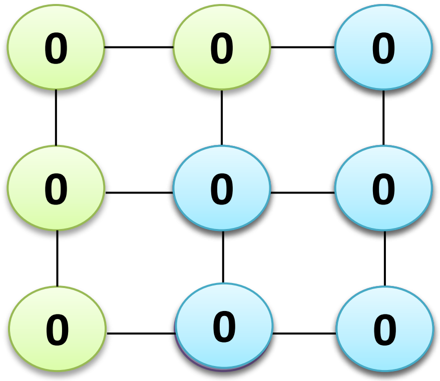
  for every variable X

    2. Draw X_t from $P(X \mid N(X)_{t-1})$.

  end

end

t=3



$$P(X = 1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X = 1, Y)}{\prod_{Y \in N(X)} \phi(X = 1, Y) + \prod_{Y \in N(X)} \phi(X = 0, Y)}$$

For the central node:

$$P(X = 1 \mid N(X)) = \frac{9*9*9*9}{9*9*9*9 + 1*1*1*1} = 0.99$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

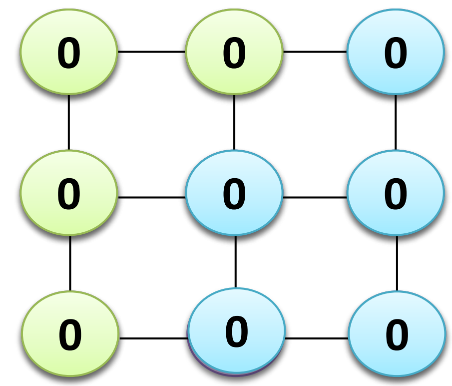
Gibbs Sampling for Inference

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M
    for every variable X
        2. Draw X_t from $P(X \mid N(X)_{t-1})$.
    end
end

t=3



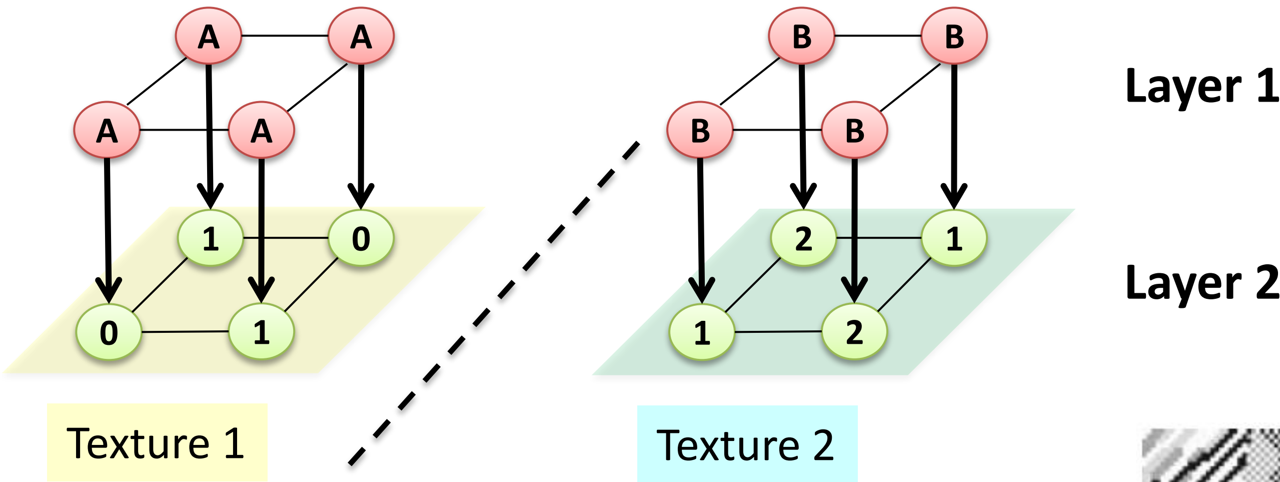
When M is large enough, $X^{(M)}$ follows stationary dist. :

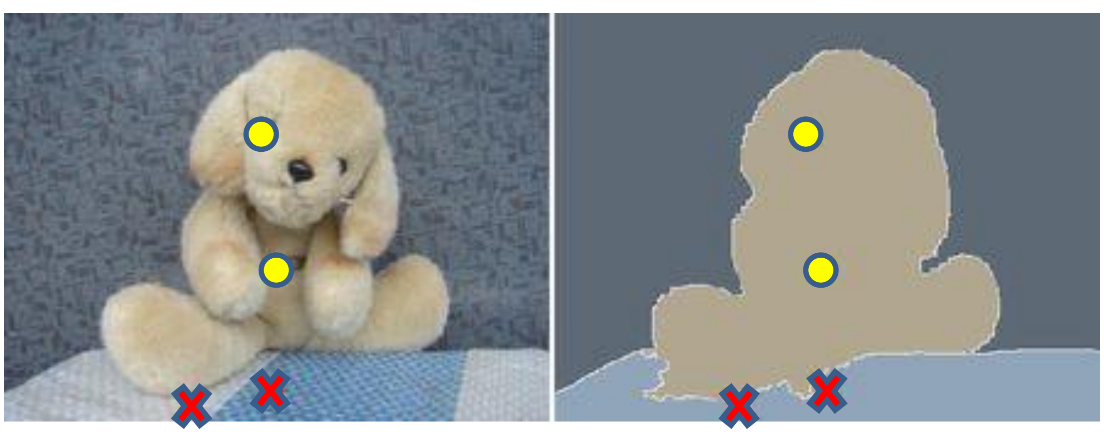
$$\pi_T(X) = P(X) = \frac{1}{Z}\prod_C \phi(X_C)$$

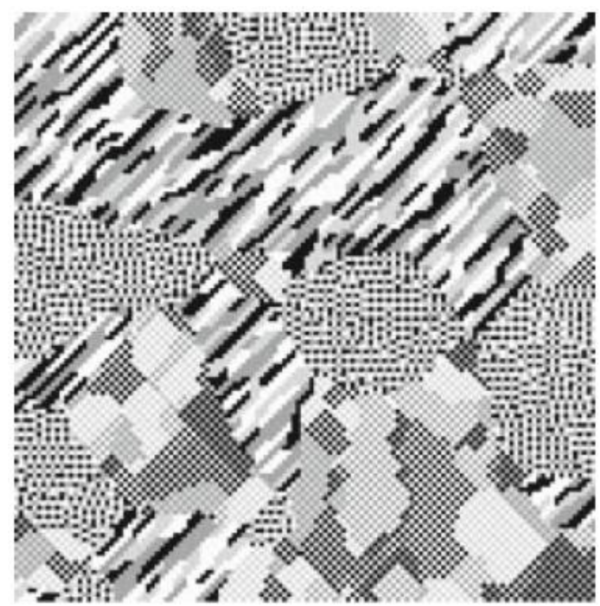
(Regularity: All entries in the Potential are positive.)

ϕ(X,Y)	0	1
0	5	1
1	1	9

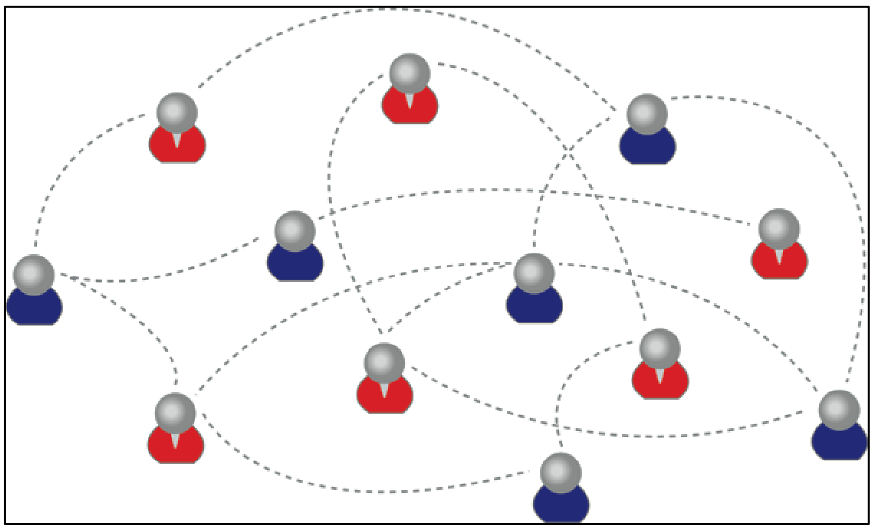
Application of MRF :
Joint Segmentation & Classification

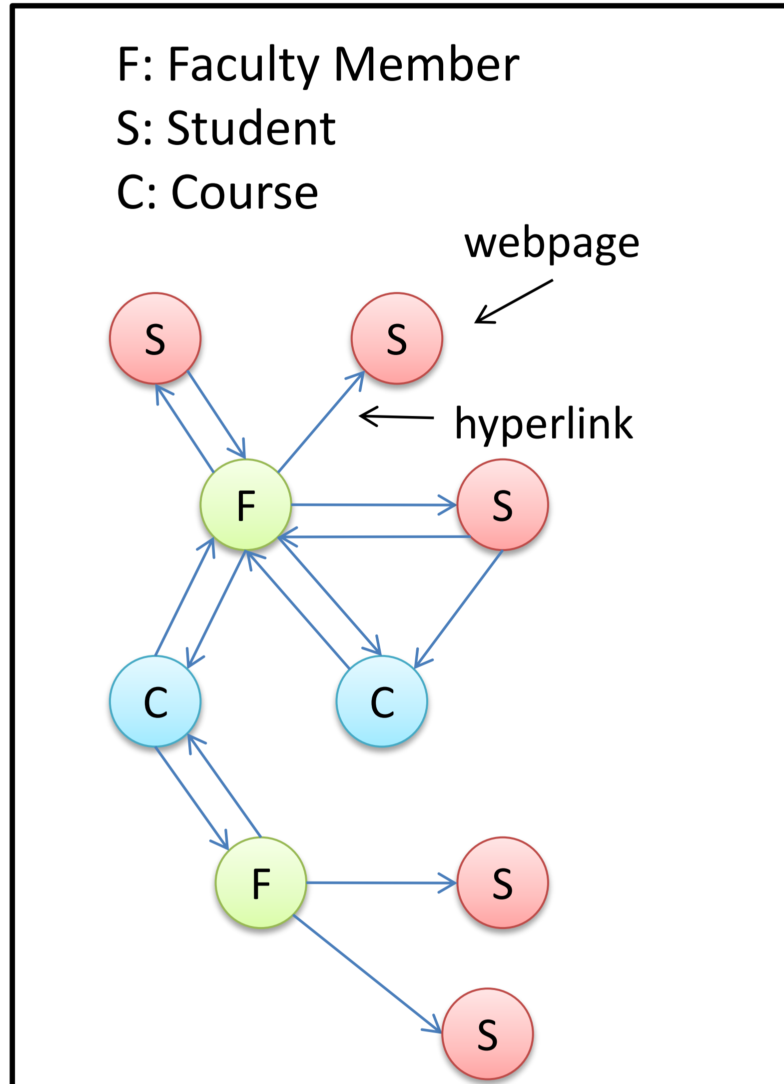




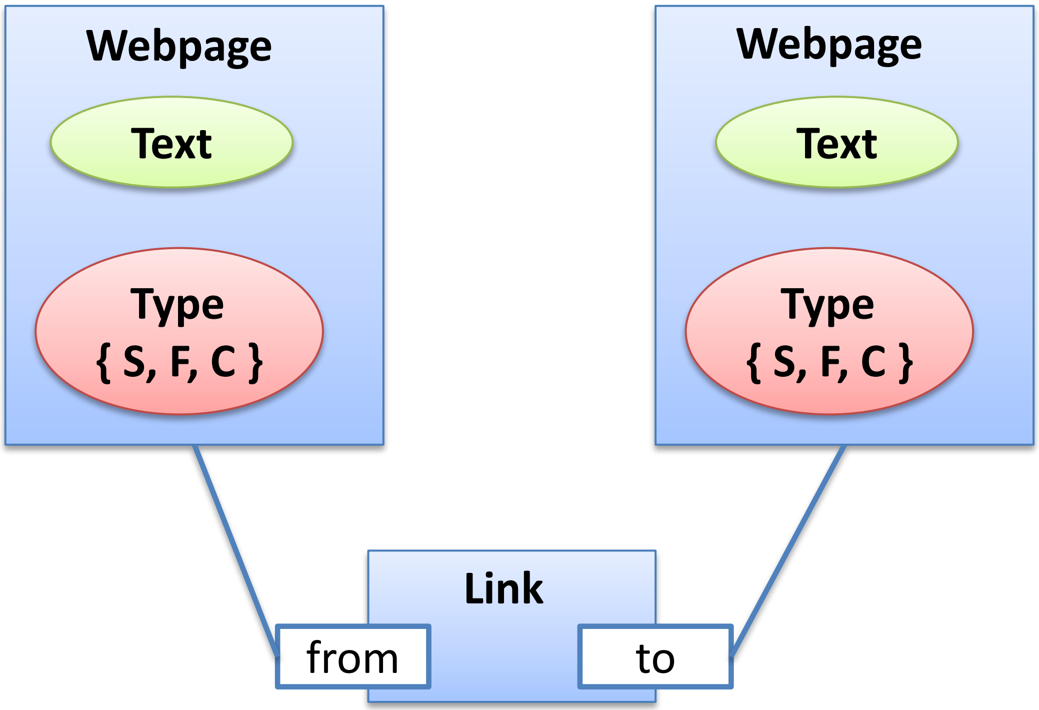


Application of MRF : Collective Classification

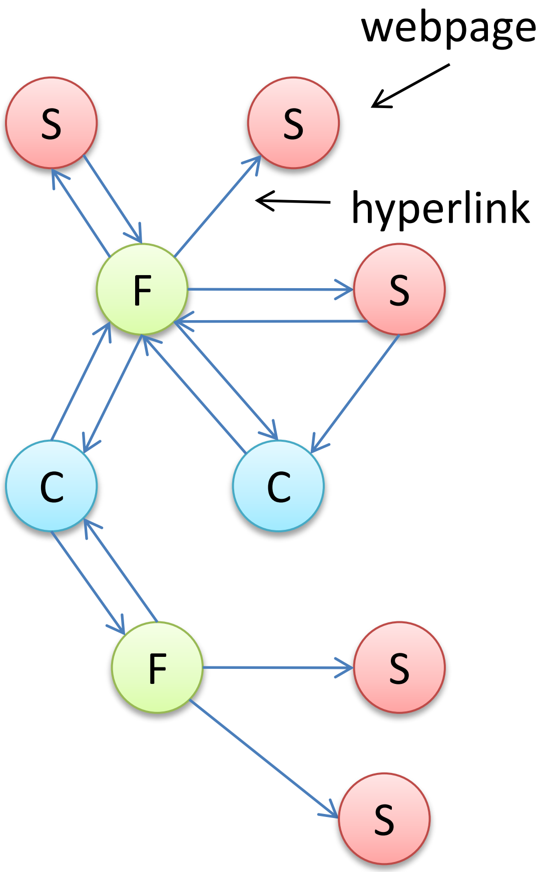




Collaborative Classification on Network



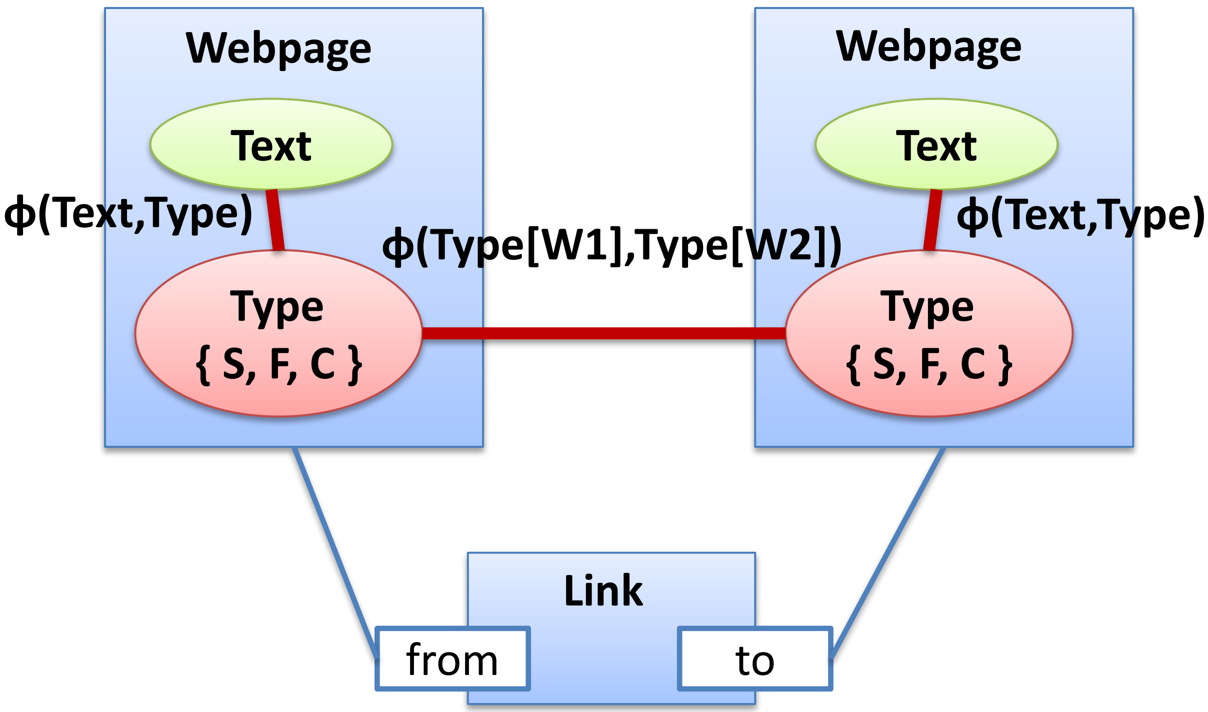
F: Faculty Member
S: Student
C: Course

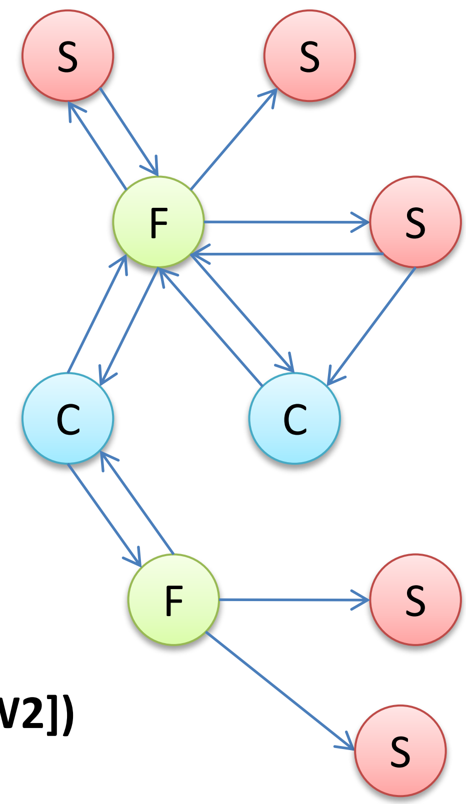


Define Global Dependency

We can define a MRF on the Schema:

F: Faculty Member
S: Student
C: Course





for each W1, W2, s.t. **Link(W1,W2)**, define **ϕ(Type[W1],Type[W2])**

Define Local Potential Function

ϕ(Type[w1], Type[w2])

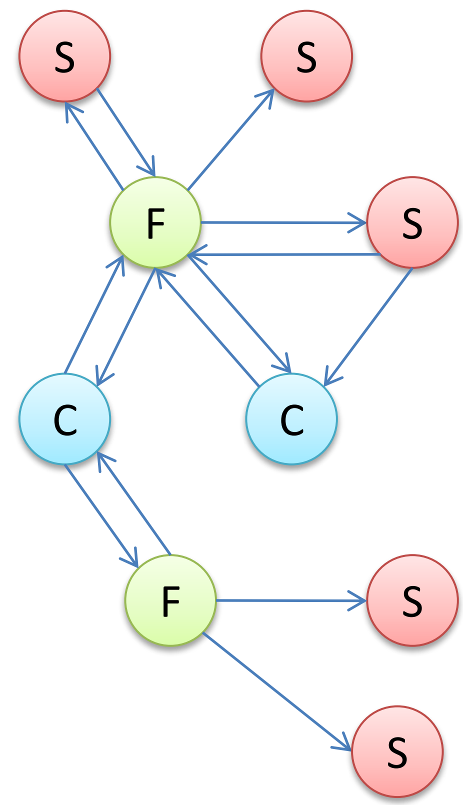
	F	**S**	**C**
F	0	3	5
S	2	0	1
C	5	0	0

F often links to S,C

C often links to F

$$P(X) = \frac{1}{Z} \prod_{C \in clique} \phi(X_C)$$

F: Faculty Member
S: Student
C: Course



Exact Inference
on Graphical Model

Reference:

Probabilistic Graphical Model Ch.9 , Ch. 10 (Koller & Friedman)
CMU, 10-708, Fall 2009 Probabilistic Graphical Models Lectures 8,9,10 (Eric Xing)

Probabilistic Inference

- A Graphical Model specifies a joint distribution $P_M(X)$ over a collection of variables X.

- How can we answer queries/questions about P(X) ?

  That is, how can we inference using P(X) ?

- Type of queries:

  - 1. Likelihood of evidence/assignments on variables
  - 2. Conditional Probability of some variables (given others).
  - 3. Most Probable Assignment for some variables (given others).

Query 1: Likelihood

- Given Evidence $E = \{X_1=x_1,\ldots, X_D=x_D\}$ specifying some variables' value and let $Z=\{Z_1, \ldots, Z_k\}$ be variables unspecified , the likelihood of a model M yielding this evidence can be computed by:

$$likelihood \;\; of \;\; E$$

$$= P_M(X) = \sum_{Z_1} \cdots \sum_{Z_K} P_M(Z_1,\ldots,Z_K, x_1,\ldots,x_D)$$

Naïve algorithm yield O($|Z|^K$) complexity…

Query 1: Likelihood

$$likelihood\ \ of\ E = \sum_{Z_1} \dots \sum_{Z_K} P_M(Z_1, \dots, Z_K, x_1, \dots, x_D)$$

- This measure is often used as criteria for Model Selection.

Ex. In speech recognition,

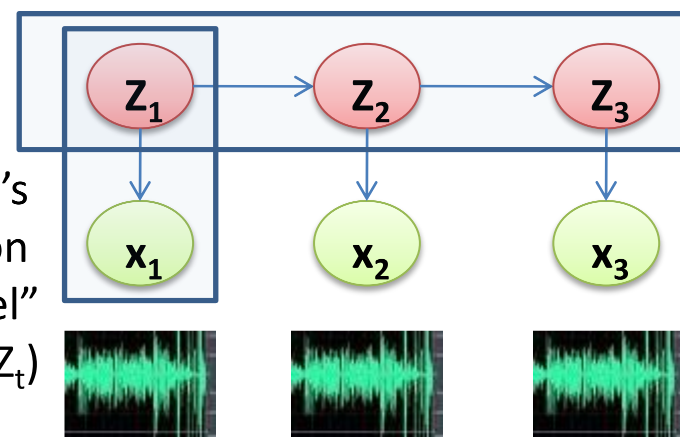
Z: words (unspecified) , X: wave sample (specified evidence E)

How likely a person M=(Language, Pronunciation) produce this wave sample can be Computed by:

$$likelihood\ \ of\ M\ \text{produce}\, \text{E}$$
$$= P_M(X)$$
$$= \sum_{Z_1} \sum_{Z_2} \sum_{Z_3} P_M(Z_1, Z_2, Z_3, x_1, x_2, x_3)$$

Summation over all possible words Z producing X

A person's "Language Model" $P(Z_{t+1}|Z_t)$



A person's "Pronunciation Model" $P(X_t|Z_t)$

Query 1: Likelihood

$$likelihood \ \ of \ \ E$$
$$= P_M(X) = \sum_{Z_1} \ldots \sum_{Z_K} P_M(Z_1, \ldots, Z_K, x_1, \ldots, x_D)$$

Taking special case E = empty , it can also be used to compute **Normalizing Const. = Z** in MRF as following :

$$(\ let \ \tilde{P}(Z_1 \ldots Z_K) = \prod_{clique C \ in \ M} \phi(C) \ \ be \ unnormalized \ dist. \ , \ \mathrm{P}(Z_1 \ldots Z_K) = \frac{1}{\mathrm{Z}} \tilde{P}(Z_1 \ldots Z_K) \ )$$

$$\sum_{Z_1} \ldots \sum_{Z_K} P(Z_1, \ldots, Z_K) \ = \ \frac{1}{Z} \sum_{Z_1} \ldots \sum_{Z_K} \tilde{P}(Z_1, \ldots, Z_K) \ = \ 1$$

$$==> \ \ Z = \sum_{Z_1} \ldots \sum_{Z_K} \tilde{P}(Z_1, \ldots, Z_K)$$

Query 2: Conditional (marginal) Probability

- Given Evidence E = $\{X_1=x_1,\ldots, X_D=x_D\}$ and some other variables $Z=\{Z_1, \ldots, Z_k\}$ unspecified , Conditional Probability of Z is given by:

$$P(Z \mid X) = \frac{P(Z, X)}{P(X)}, \ \textit{where } P(X) \text{ is } \textit{given} \text{ by Query 1}$$

- Sometimes we are interested in only some variables Y in Z, where Z = { Y ,W }, then conditional (marginal) prob. of Y is

$$P(Y \mid X) = \sum_{W} P(Z \mid X) = \sum_{W_1} \ldots \sum_{W_K} P(Y, W_1 \ldots W_K \mid X)$$

Naïve summation over uninterested variables W yield O($|W|^K$) complexity…

Query 2: Conditional (marginal) Probability

Ex. In speech recognition,

Z: words (unspecified) , X: wave sample (specified evidence E)

$$P(Z \mid X) = \frac{P(Z, X)}{P(X)}, \quad \textit{where } P(X) \text{ is } \textit{given} \text{ by Query 1}$$

A word sequence $Z_1 \ldots Z_K$'s prob. given the wave sample $X_1 \ldots X_K$

If we only care the 1st word, then:

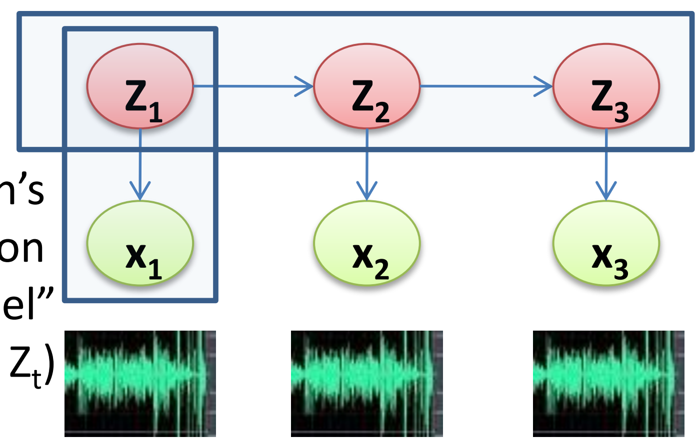
$$P(Z_1 \mid X) = \sum_{Z_2} \sum_{Z_3} P(Z_1, Z_2, Z_3 \mid X)$$

The 1st word Z_1's marginal distribution given the wave sample $X_1 \ldots X_K$

(naïve method is intractable for large K)

A person's "Language Model" $P(Z_{t+1} \mid Z_t)$

A person's "Pronunciation Model" $P(X_t \mid Z_t)$



Query 3: Most Probable Assignment

- Given Evidence $E = \{X_1=x_1,\ldots, X_D=x_D\}$ and some other variables $Z=\{Z_1, \ldots, Z_k\}$ unspecified , Most Probable Assignment of Z is given by:

$$MPA(Z \mid X) = \arg\max_{Z} \; P(Z \mid X)$$

$$= \arg\max_{Z} \; \frac{P(X \mid Z)P(Z)}{\mathrm{P(X)}} = \arg\max_{Z} \; P(X \mid Z)P(Z)$$

- MPA is also called "maximum a posteriori configuration" or "MAP inference".

Note:

1. Even if we have computed **Query 2 = P(Z|X)** , it's intractable to enumerate all possible Z to get **argmax_Z P(Z|X).**

2. MPA cares "**Joint Maximum**", not "**Marginal Maximum**".

$$\underset{Z}{\mathrm{argmax}} \; \mathrm{P(Z \mid X)} \neq \begin{cases} \arg\max_{Z1} \; P(Z_1 \mid X) \\ \ldots\ldots \\ \arg\max_{ZK} \; P(Z_K \mid X) \end{cases}$$

Query 3: Most Probable Assignment

We often just want to **"decode words "** from the wave sample,
That is, we care **$Z^* = \text{argmax}_Z\, P(Z|X)$** but not **$P(Z|X)$** itself.

Marginal Maximum:

$$\begin{cases} \arg\max_{Z1} P(Z_1 \mid X) \\ \arg\max_{Z2} P(Z_2 \mid X) \\ \arg\max_{Z3} P(Z_3 \mid X) \end{cases} \Rightarrow \text{may give } Z_1 = 'I', \quad Z_2 = 'comes', \quad Z_3 = 'front'$$

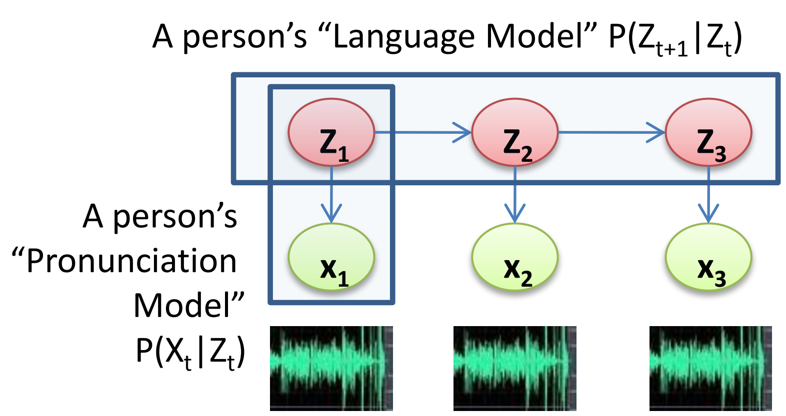
(inconsistent decoding)

Joint Maximum (MPA) :

$$\arg\max_{Z_1, Z_2, Z_3} P(Z_1, Z_2, Z_3 \mid X)$$

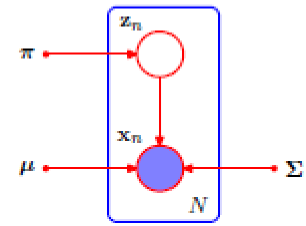
$$\Longrightarrow may\ \ give\ \ 'I'\ \ 'come'\ \ 'from'$$

(consistent decoding)



In terms of difficulty, there are 3 types of inference problem.

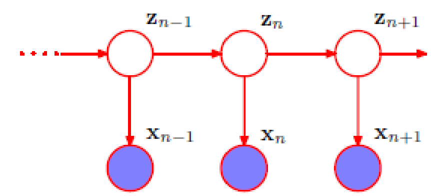
- Inference which is easily solved with Bayes rule.



Today's focus

- Inference which is tractable using some dynamic programming technique.

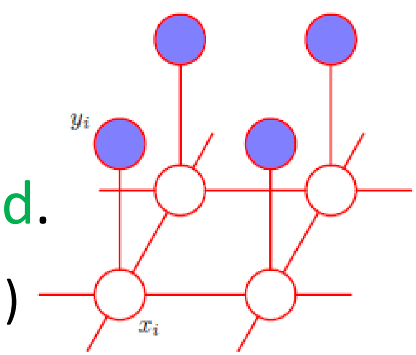
  (e.g. Variable Elimination or J-tree algorithm)



- Inference which is proved intractable

  & should be solved using some Approximate Method.

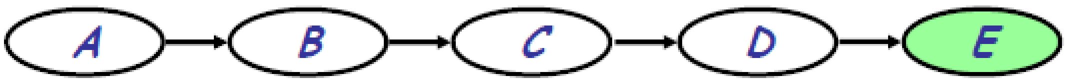
  (e.g. Approximation with Optimization or Sampling technique.)



Agenda

- Introduce the concept of "Variable Elimination" in special case of Tree-structured Factor Graph.
- Extend the idea of "VE" to general Factor graph with concept of "Clique Tree".
- See how to extend "VE" to "Most Probable Assignment" (MAP configuration) Problem.

Variable Elimination: Inference on a Chain



How to get P(E=e) ?

$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A,B,C,D,E=e)$$

By structure of the BN:

$$P(A,B,C,D,E)=P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

$$P(E)=\sum_D\sum_C\sum_B\sum_A P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

We can put summation as right as possible…

Variable Elimination: Inference on a Chain

How to get P(E=e) ?

$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A,B,C,D,E=e)$$

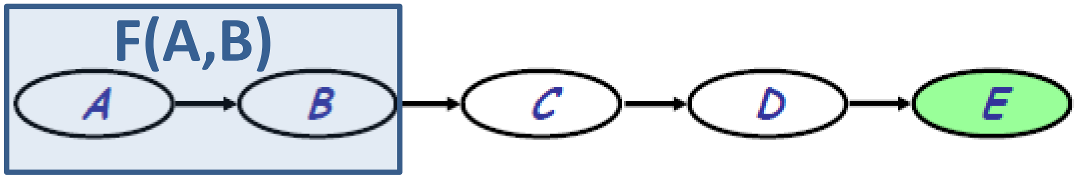
By structure of the BN:

$$P(A,B,C,D,E)=P(E\,|\,D)P(D\,|\,C)P(C\,|\,B)P(B\,|\,A)P(A)$$

$$P(E)=\sum_D\sum_C\sum_B\sum_A P(E\,|\,D)P(D\,|\,C)P(C\,|\,B)P(B\,|\,A)P(A)$$

$$=\sum_D P(E\,|\,D)\sum_C P(D\,|\,C)\sum_B P(C\,|\,B)\sum_A P(B\,|\,A)P(A)$$

Variable Elimination: Inference on a Chain



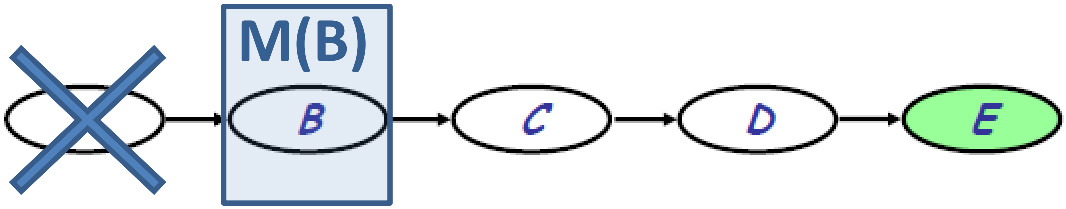
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B\,|\,A)P(C\,|\,B)P(D\,|\,C)P(E=e\,|\,D)$$

$$=\sum_D P(E\,|\,D)\sum_C P(D\,|\,C)\sum_B P(C\,|\,B)\sum_A \underline{P(B\,|\,A)P(A)}$$

F(A,B)

A Table size=|A||B|

Variable Elimination: Inference on a Chain



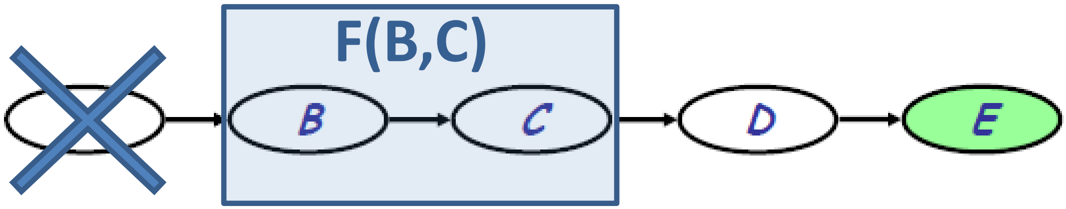
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B\mid A)P(C\mid B)P(D\mid C)P(E=e\mid D)$$

$$=\sum_D P(E\mid D)\sum_C P(D\mid C)\sum_B P(C\mid B)\sum_A P(B\mid A)P(A)$$

$\sum_A F(A,B)=M(\ B\ )$

Eliminate "A". A Table size=|B|.

Variable Elimination: Inference on a Chain



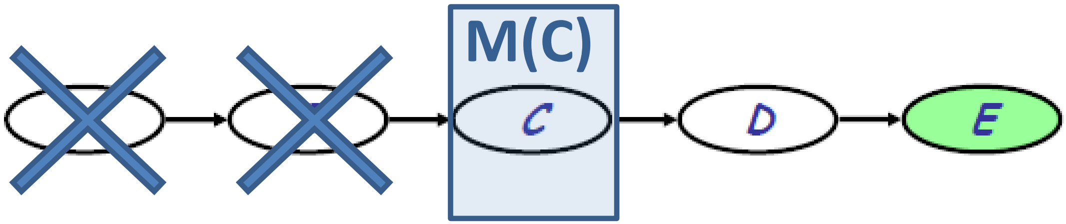
$$P(E=e) = \sum_A \sum_B \sum_C \sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$= \sum_D P(E|D) \sum_C P(D|C) \sum_B P(C|B) \sum_A P(B|A)P(A)$$

P(C|B)*M(B) = F(B,C)

A Table size=|B||C|.

Variable Elimination: Inference on a Chain



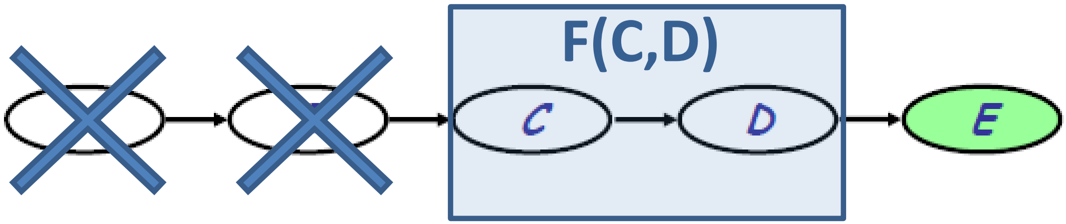
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A P(B|A)P(A)$$

$\sum_B$ F(B,C)=M(C)

Eliminate "B". A Table size=|C|.

Variable Elimination: Inference on a Chain



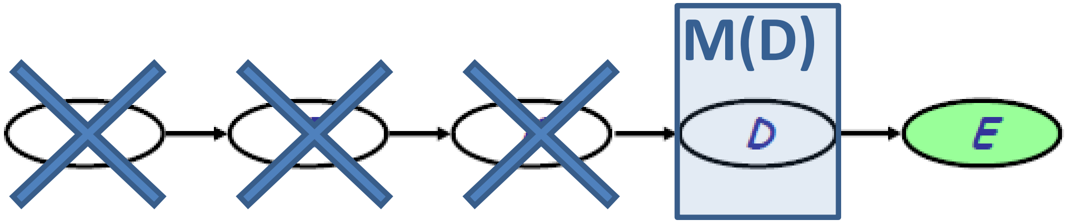
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B\,|\,A)P(C\,|\,B)P(D\,|\,C)P(E=e\,|\,D)$$

$$=\sum_D P(E\,|\,D)\sum_C P(D\,|\,C)\sum_B P(C\,|\,B)\sum_A P(B\,|\,A)P(A)$$

P(D|C)M(C) = F(C,D)

A Table size=|C||D|.

Variable Elimination: Inference on a Chain



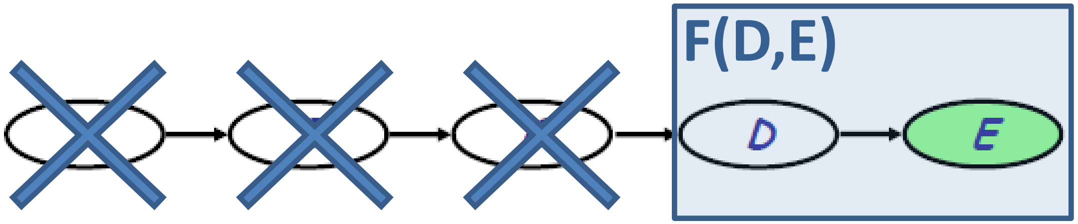
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A P(B|A)P(A)$$

$\sum_C F(C,D) = M(D)$

Eliminate "C". A Table size=|D|.

Variable Elimination: Inference on a Chain



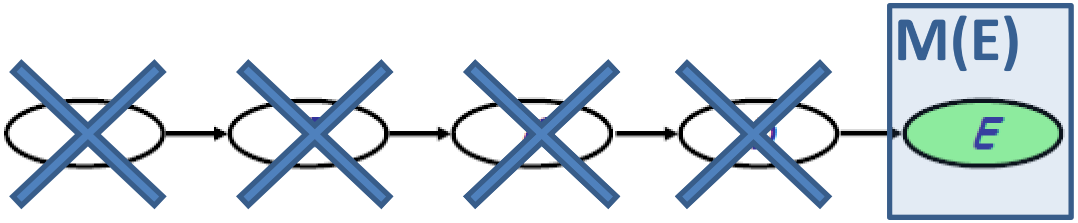
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B\mid A)P(C\mid B)P(D\mid C)P(E=e\mid D)$$

$$=\sum_D P(E\mid D)\sum_C P(D\mid C)\sum_B P(C\mid B)\sum_A P(B\mid A)P(A)$$

P(E|D)M(D)=F(D,E)

A Table size=|D||E|.

Variable Elimination: Inference on a Chain



$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A P(B|A)P(A)$$

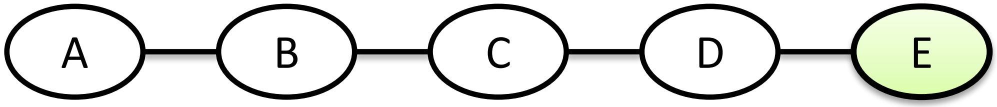
$\sum_D F(D,E)=M(E)=P(E)$

Eliminate D. Get the answer.

Both Time & Space Complexity are **O(|A||B|+|B||C|+|C||D|+|D||E|) → O(|Range|^2)**

Naïve method complexity = **O(|A||B||C||D||E|) → O(|Range|^N)**

Variable Elimination: Inference on a Chain



How about inference on Undirected Model (MRF) ?

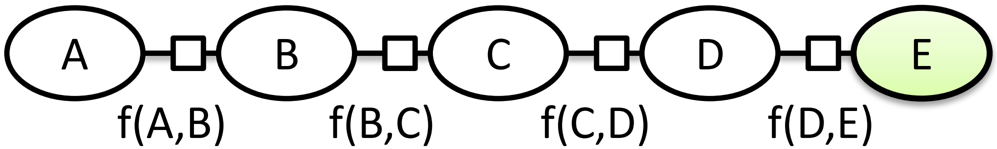
$$P(A,B,C,D,E) = \frac{1}{Z}\phi(E,D)\phi(D,C)\phi(C,B)\phi(B,A)$$

$$P(E) = \frac{1}{Z}\sum_D\sum_C\sum_B\sum_A \phi(E,D)\phi(D,C)\phi(C,B)\phi(B,A)$$

$$= \frac{1}{Z}\sum_D \phi(E,D)\sum_C \phi(D,C)\sum_B \phi(C,B)\sum_A \phi(B,A)$$

The same idea applies !!

Variable Elimination: Inference on a Chain



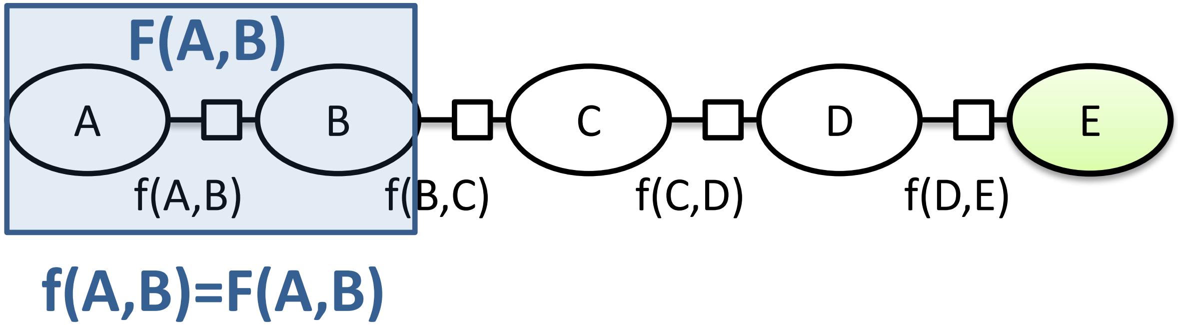
From now on, we won't distinguish between BN & MRF.
The same algorithm applies to them in a "Factor View".

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

$$P(A,B,C,D,E) = \mathbf{1}*P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

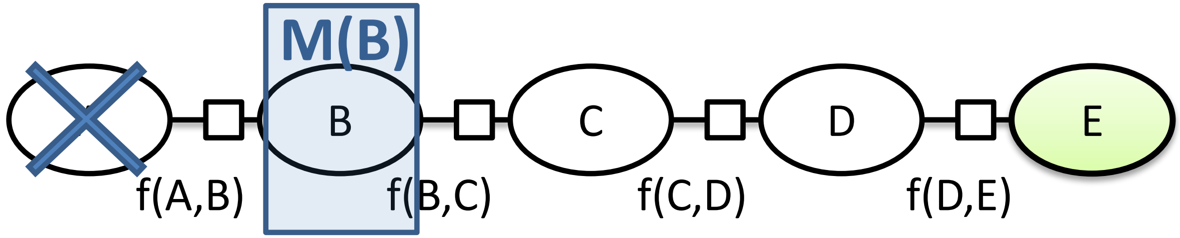
All viewed as: $\frac{1}{Z} f(E,D)\,f(D,C)\,f(C,B)\,f(B,A)$

Variable Elimination: Inference on a Chain



The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain

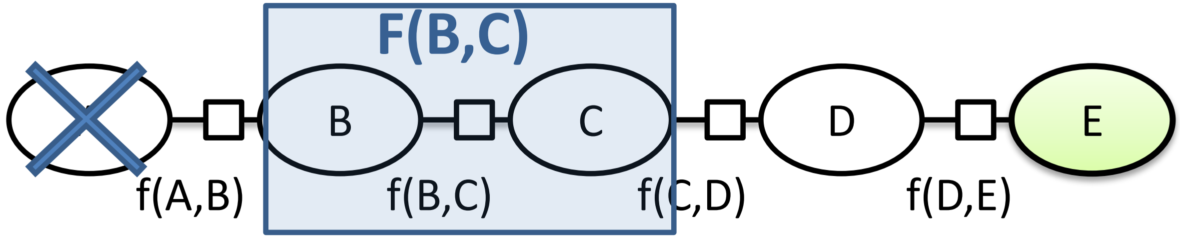


$\sum_A F(A,B) = M(B)$

Sum !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

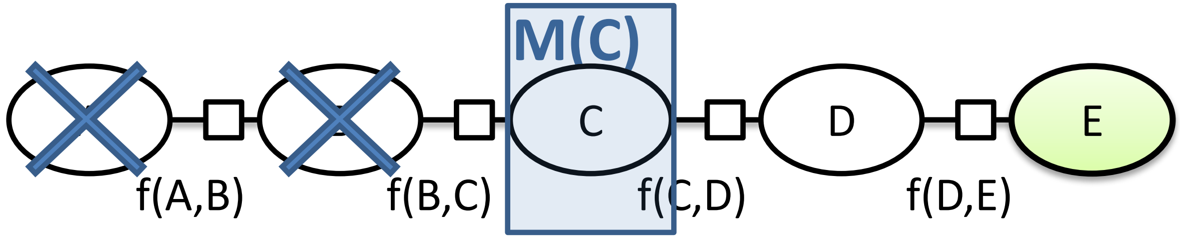


M(B)*f(B,C)= F(B,C)

Product !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

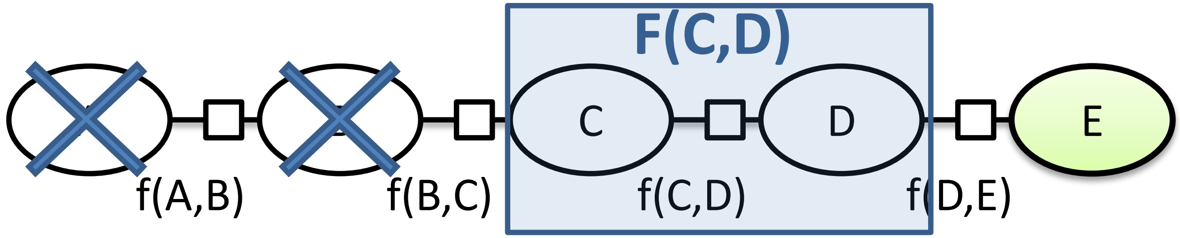


$\sum_B F(B,C)=M(C)$

Sum !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

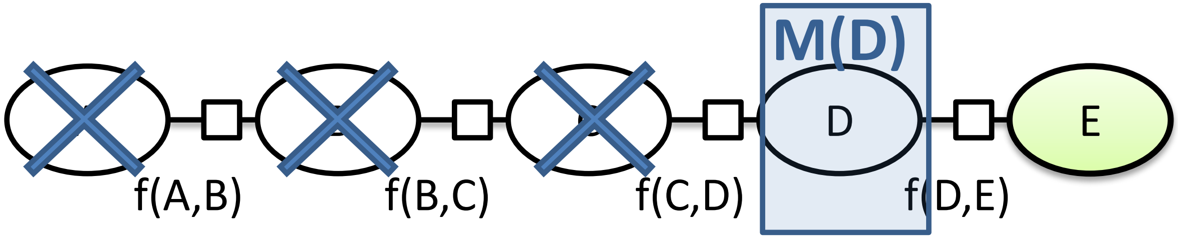


M(C)*f(C,D) = F(C,D)

Product !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

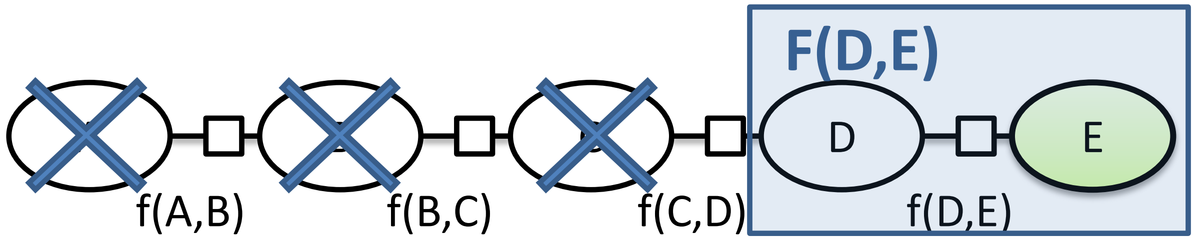


$\sum_C F(C,D) = M(D)$

Sum !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

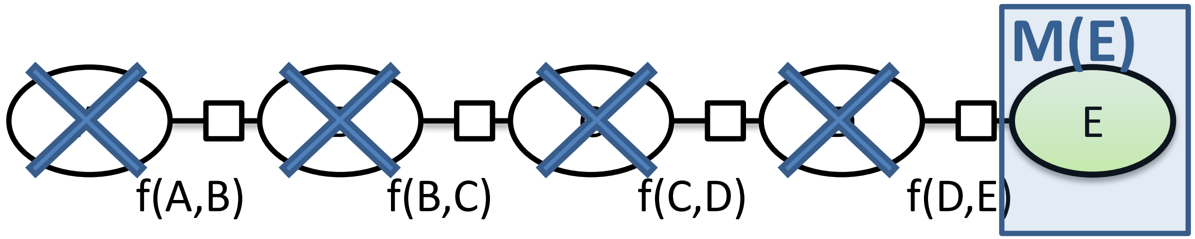


M(D)*f(D,E) = F(D,E)

Product !

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain



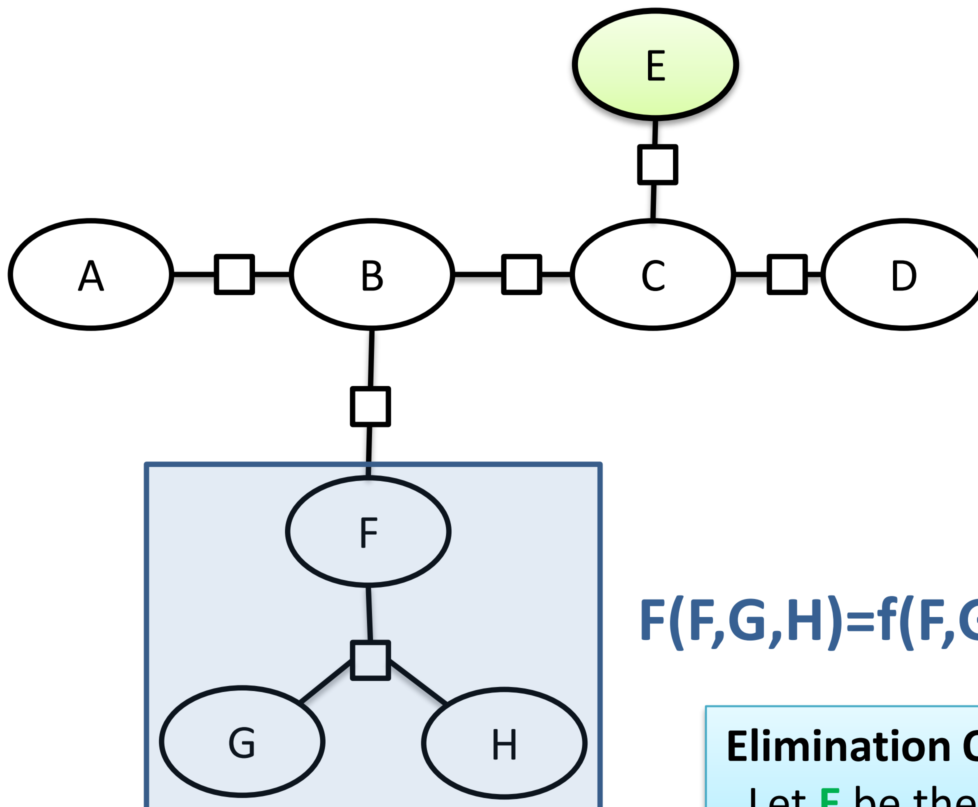
$\sum_D F(D,E) = M(E) = P(E)$

Sum !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.

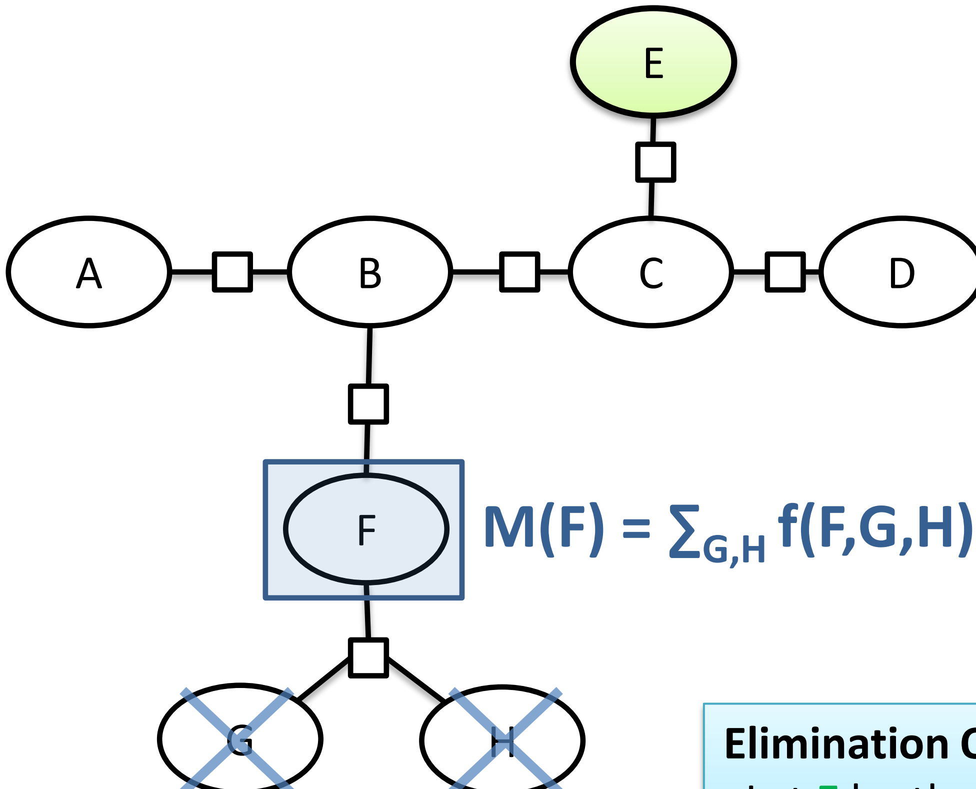


F(F,G,H)=f(F,G,H)

Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.

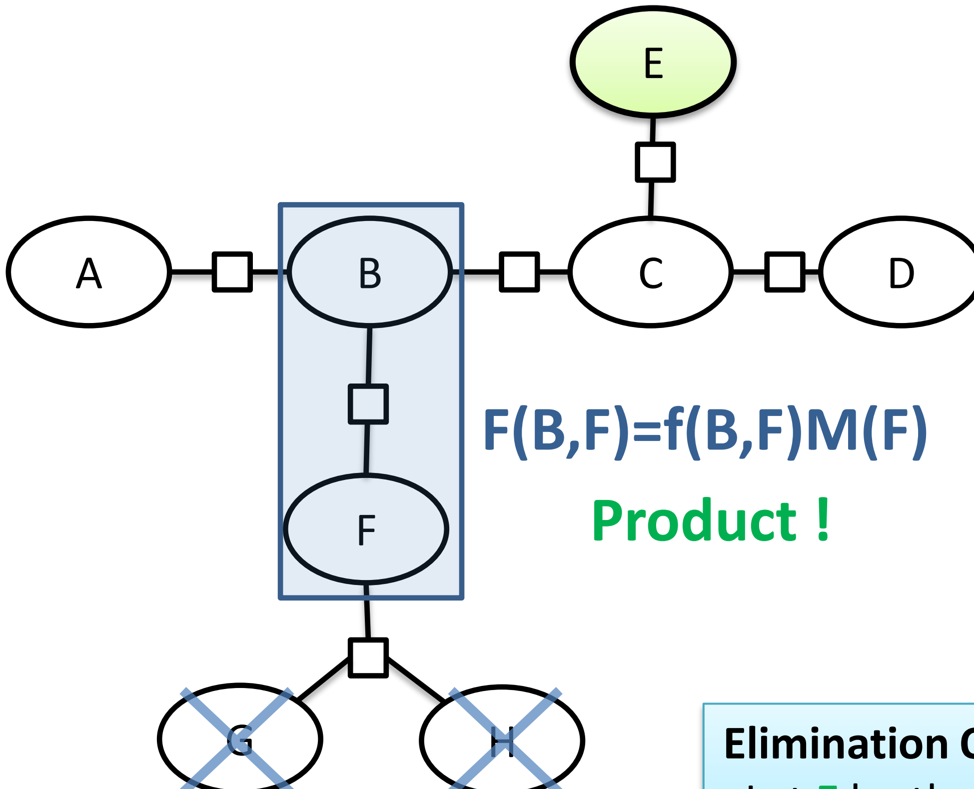


$M(F) = \sum_{G,H} f(F,G,H)$ **Sum !**

Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

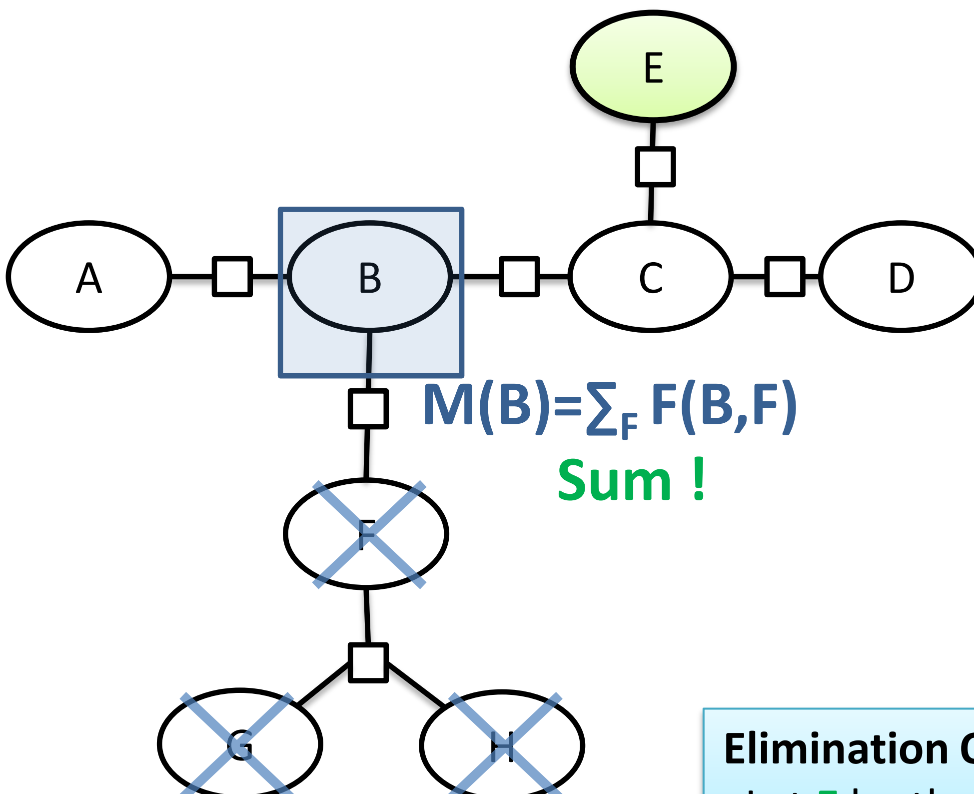
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

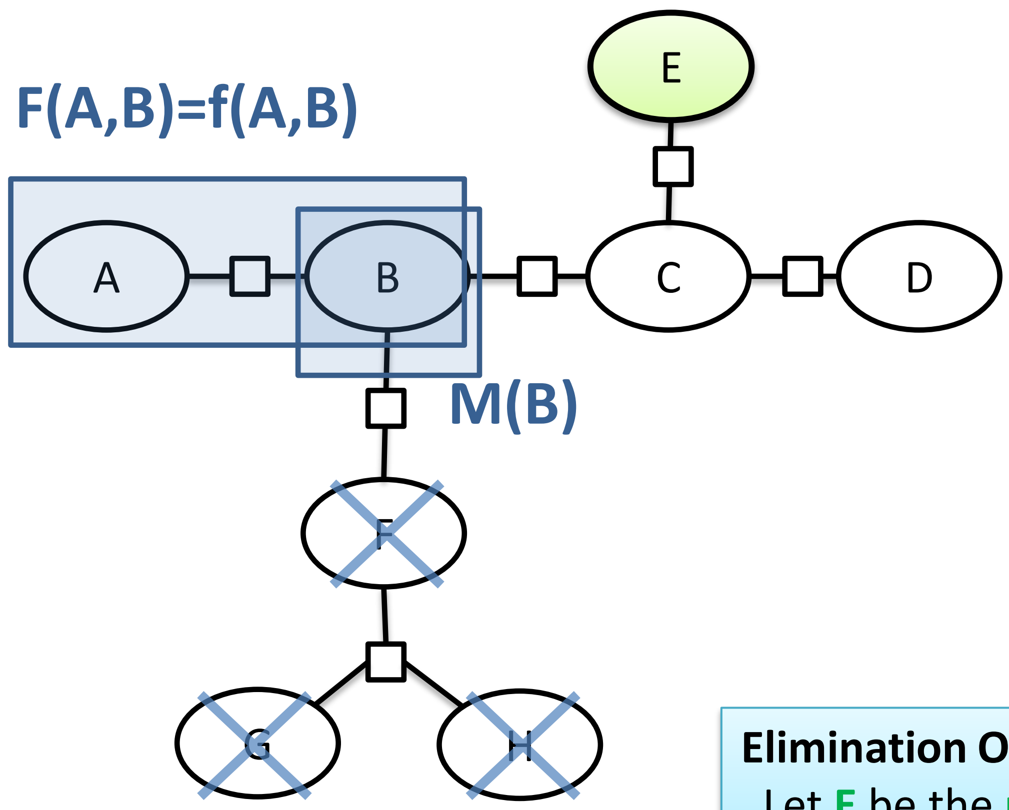
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

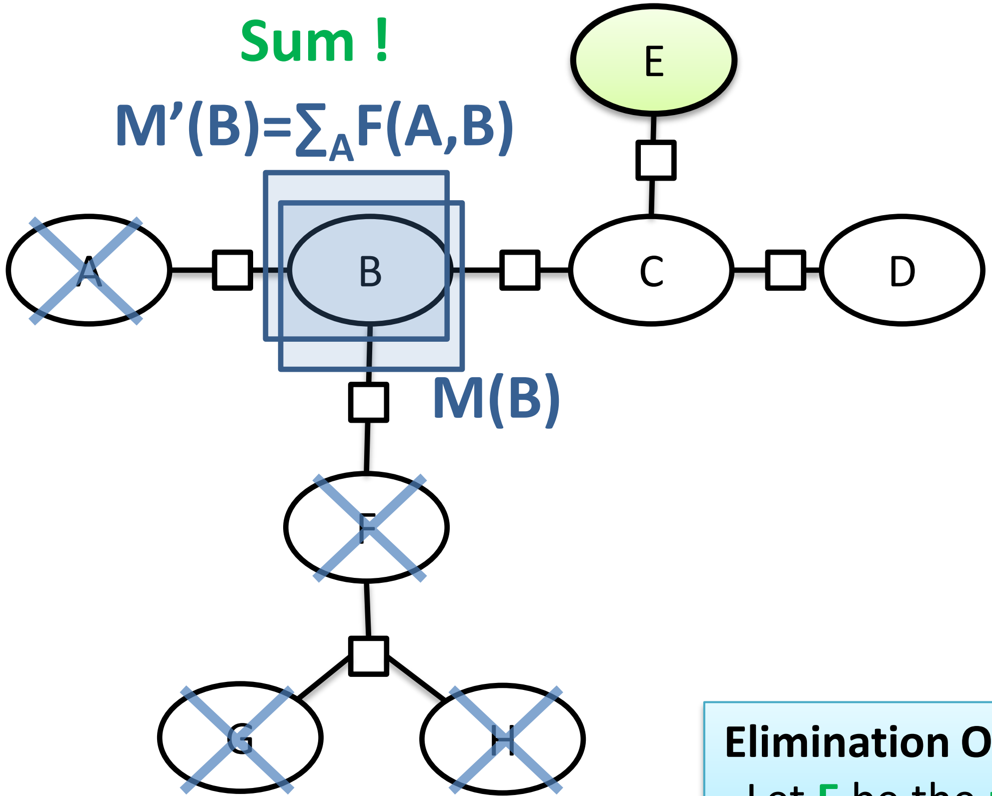
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

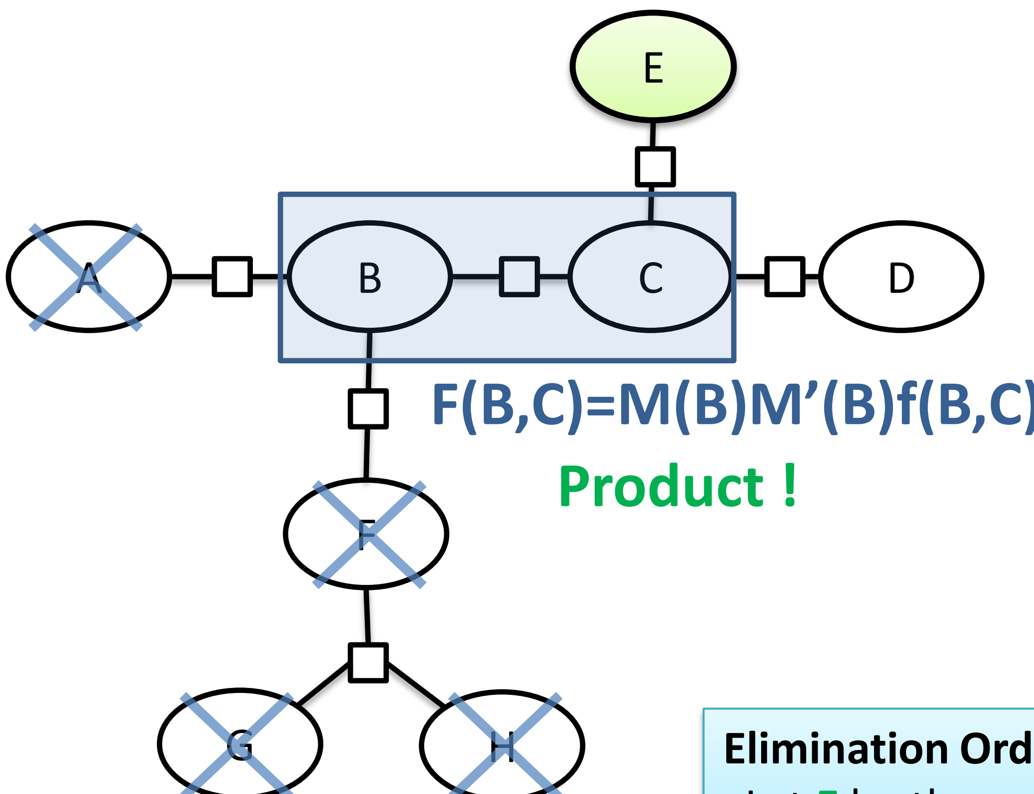
If the factor graph is a tree without cycle, then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

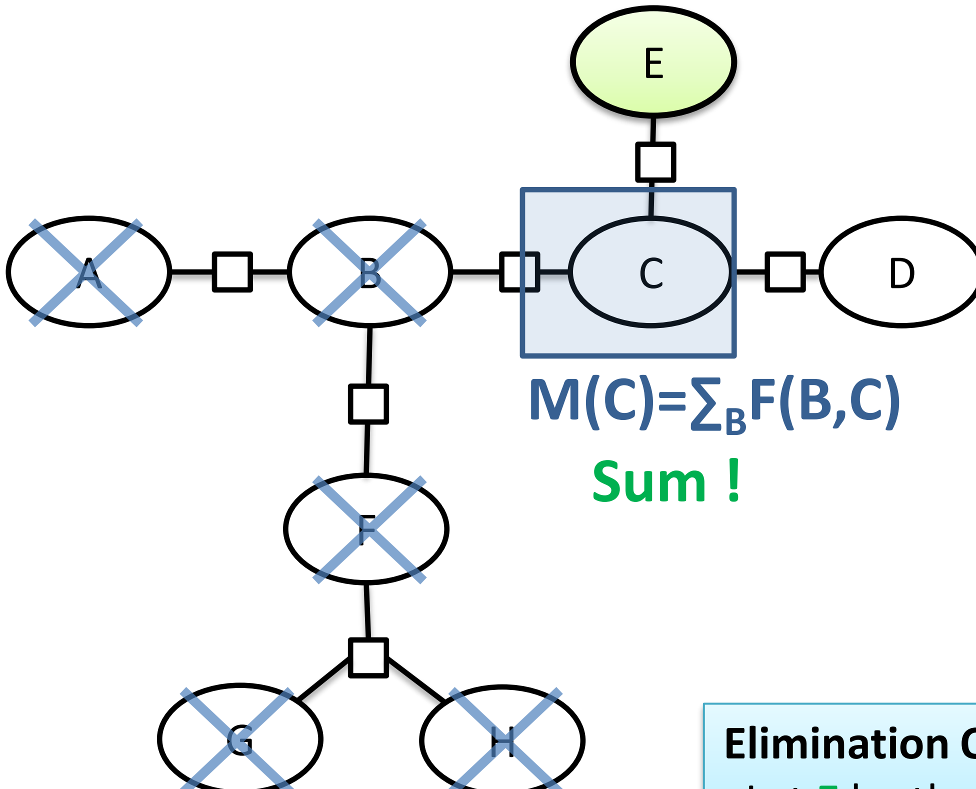
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



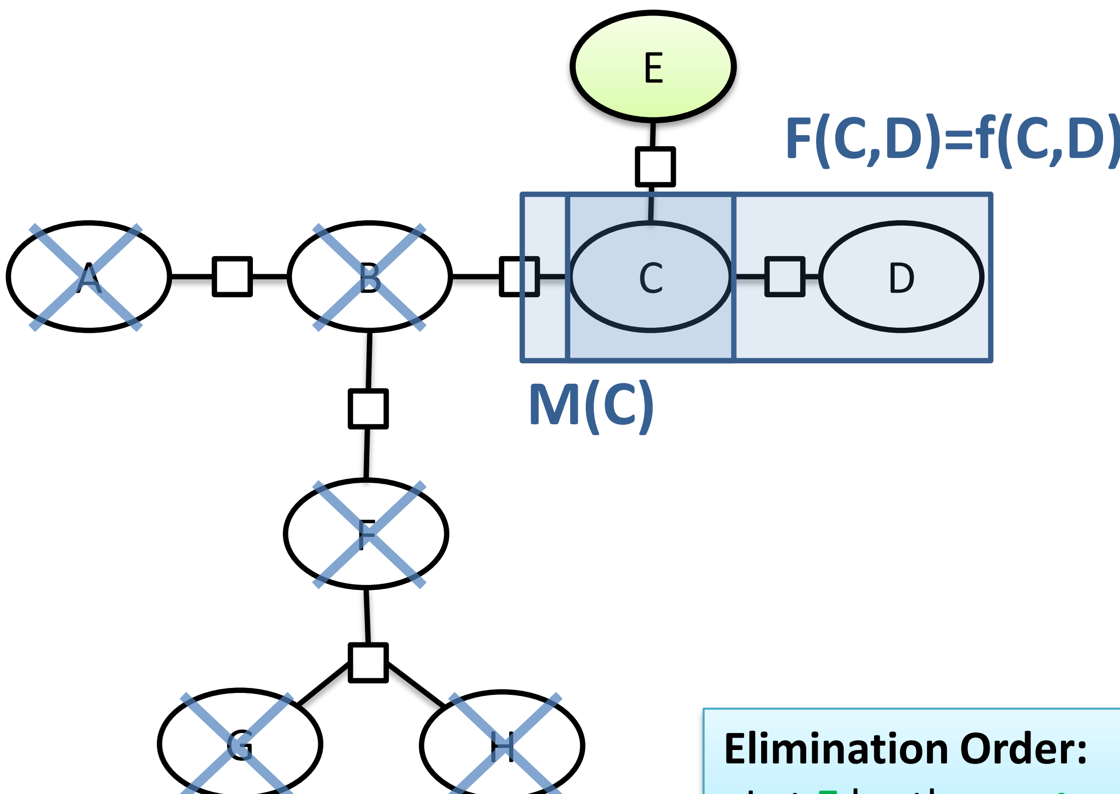
$M(C)=\sum_B F(B,C)$

Sum !

Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

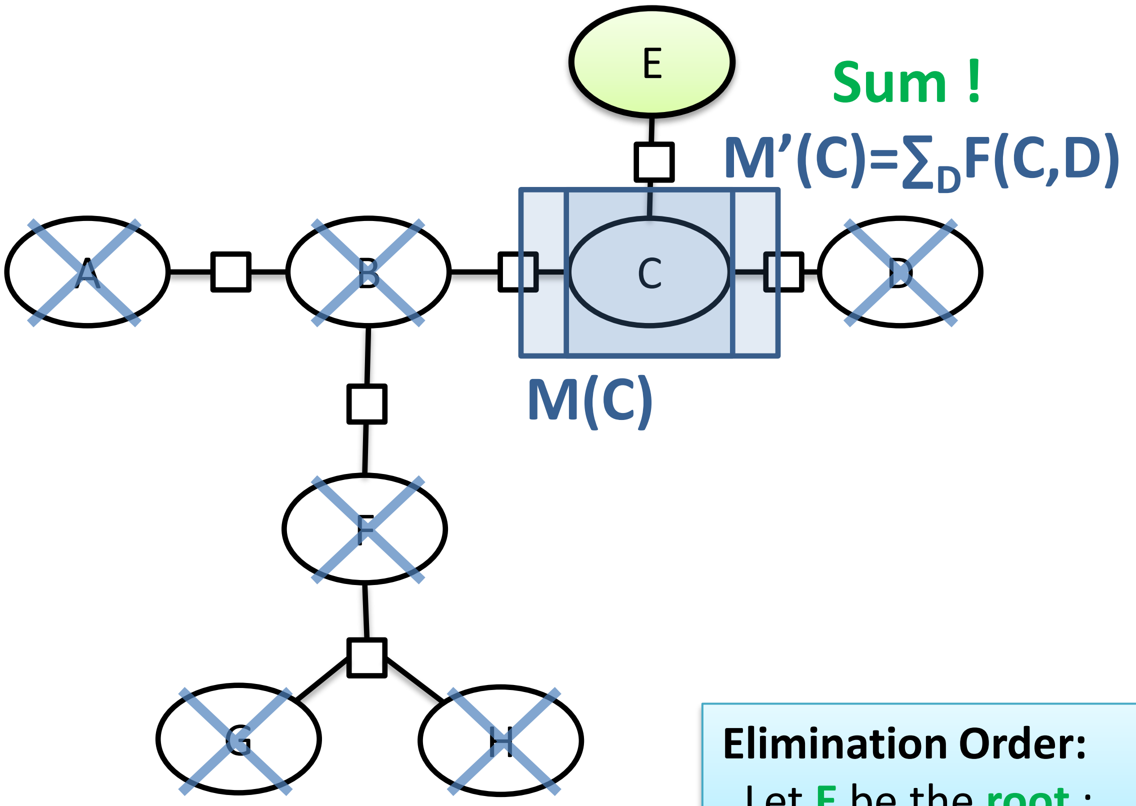
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

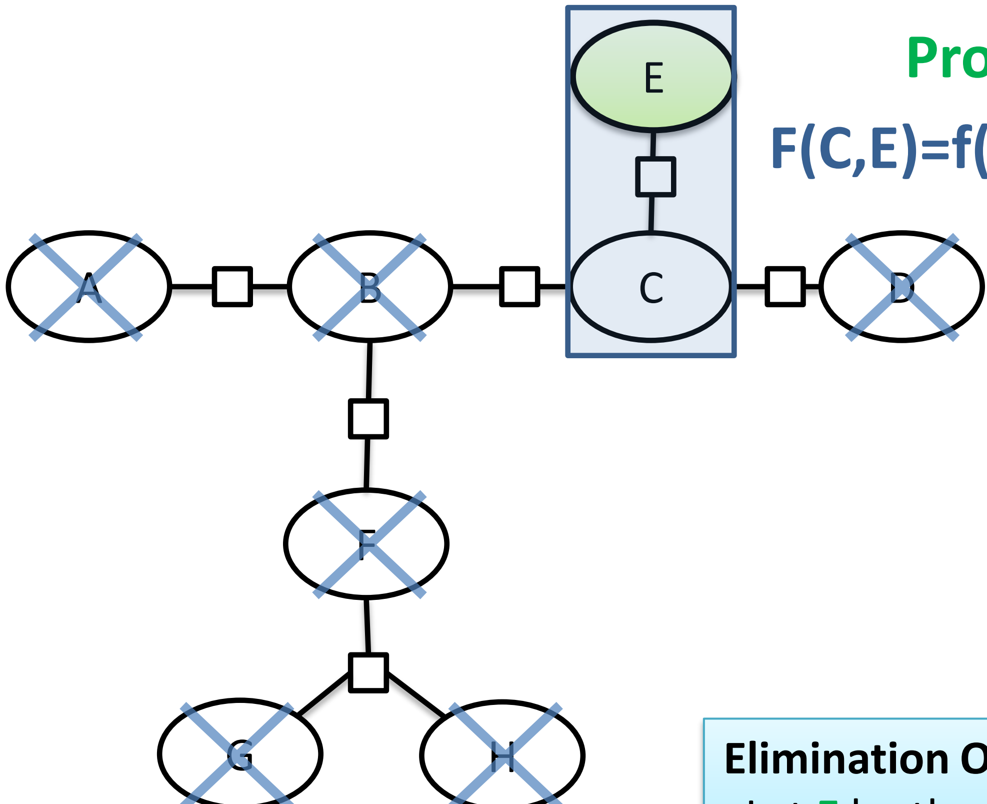
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle, then VE can be applied in a similar way.



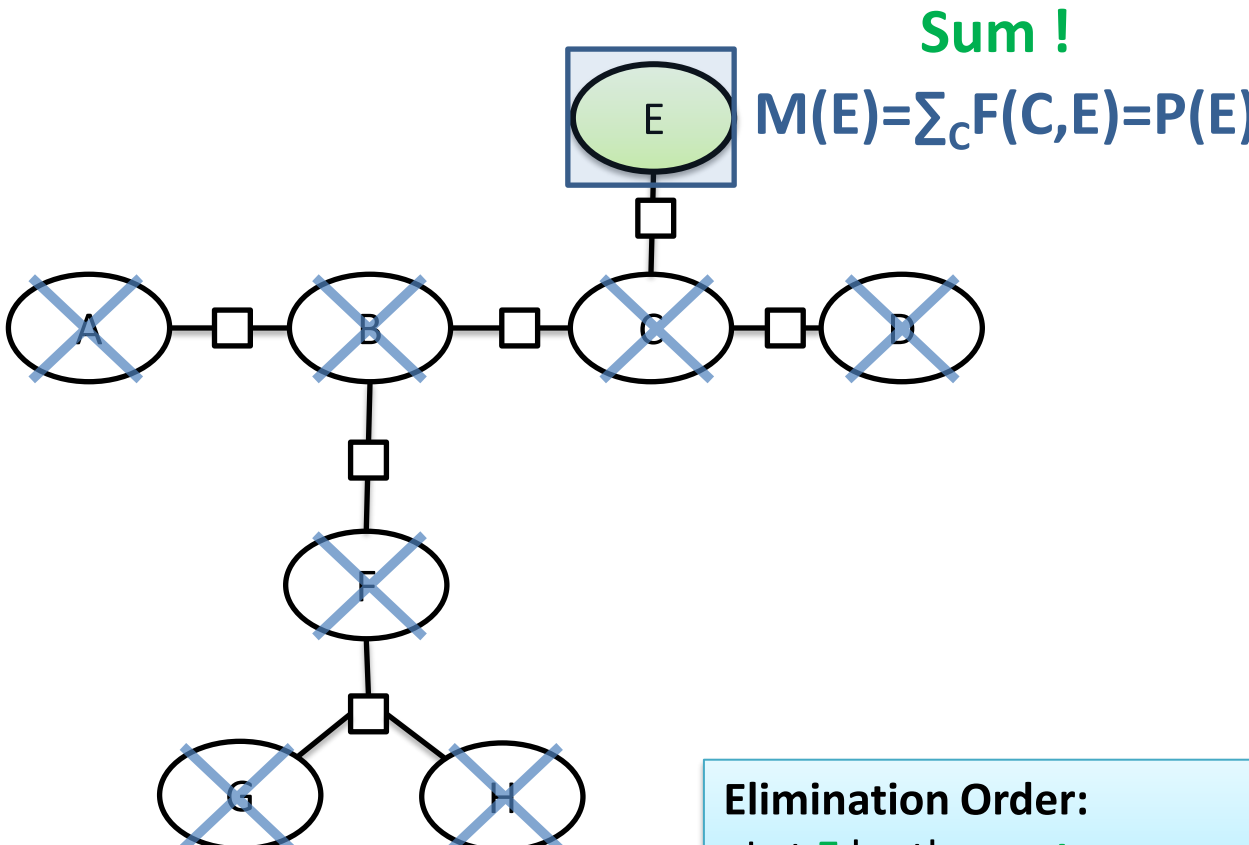
Product !

F(C,E)=f(C,E)M(C)M'(C)

Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle, then VE can be applied in a similar way.

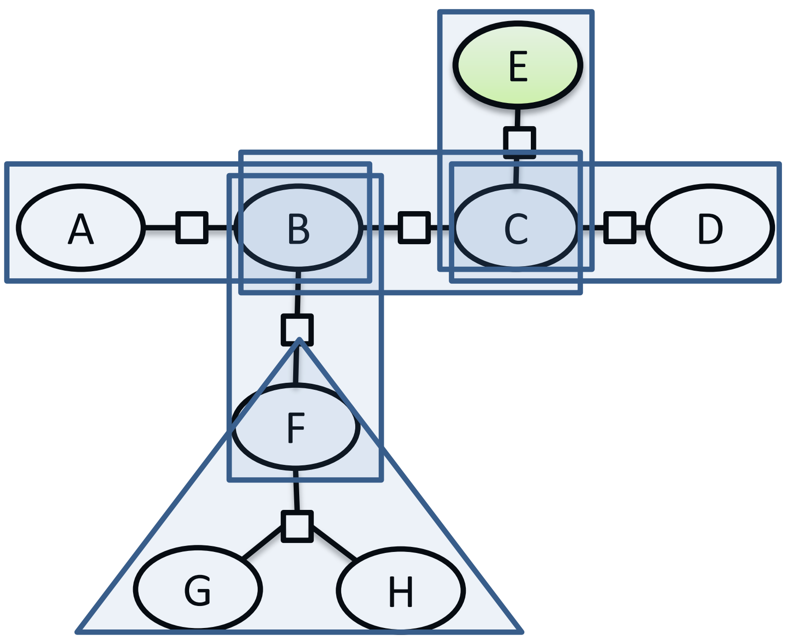


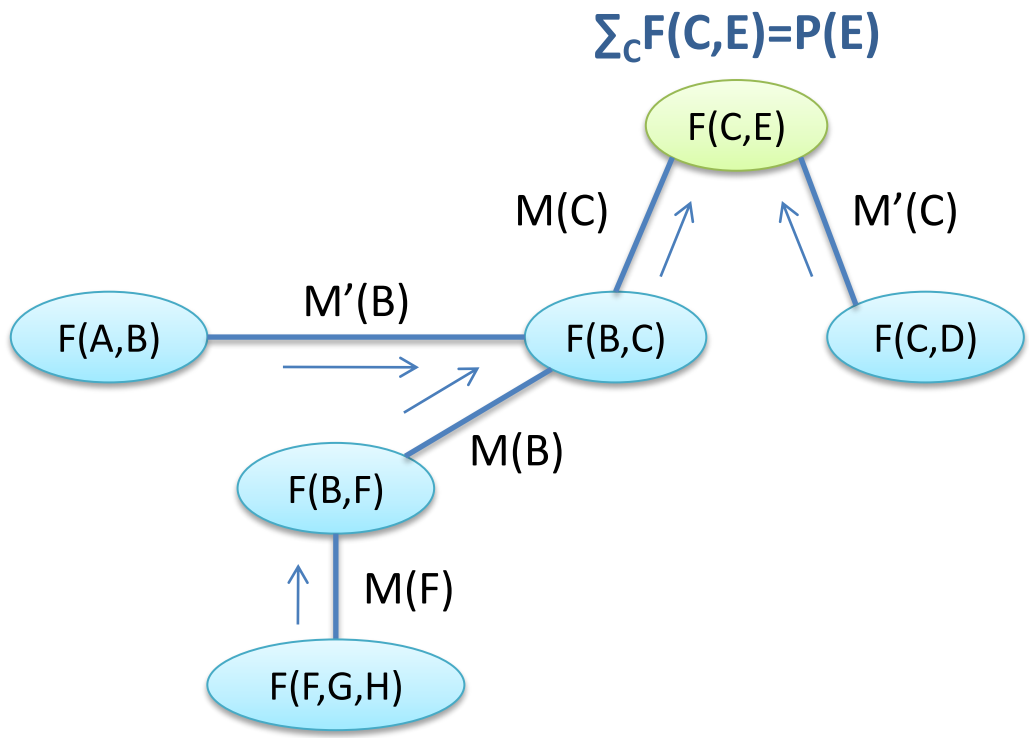
Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

Follow the Elimination Process, we can build a "Clique Tree". In which:

1. Every node is a F(.) before elimination.
2. Every edge is a "message" M(.) passed from F(.) to F(.).

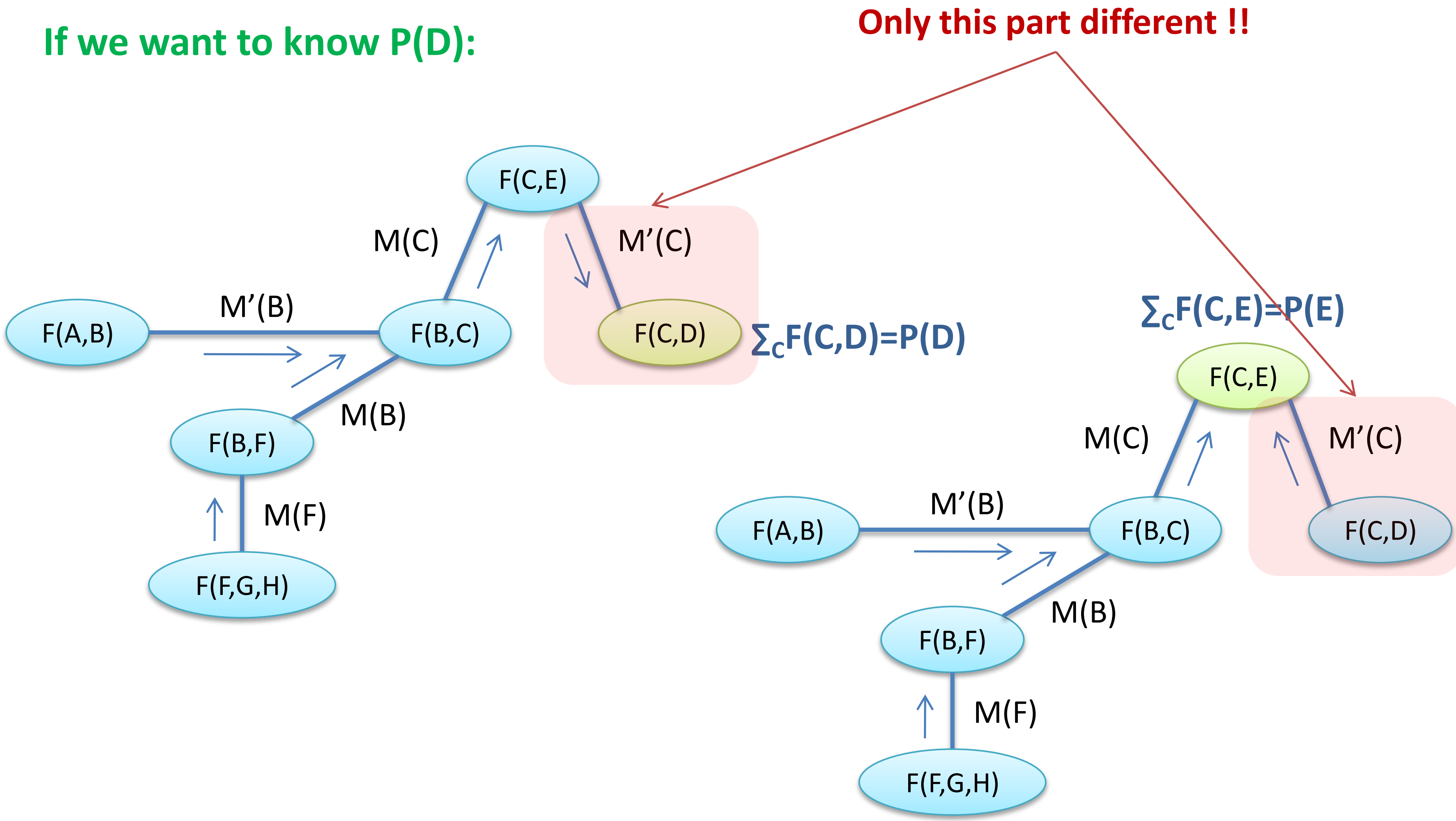




Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

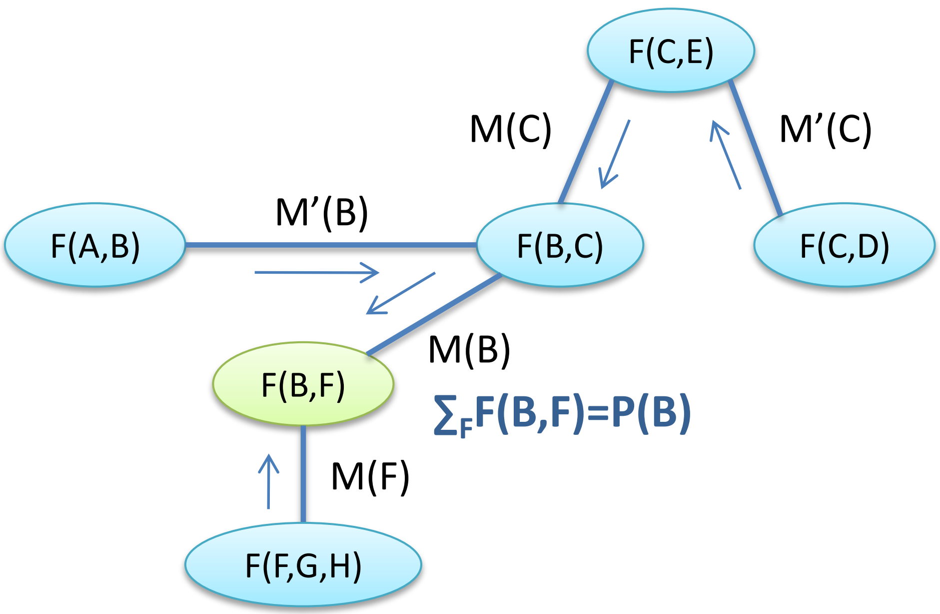
If we want to know P(D):

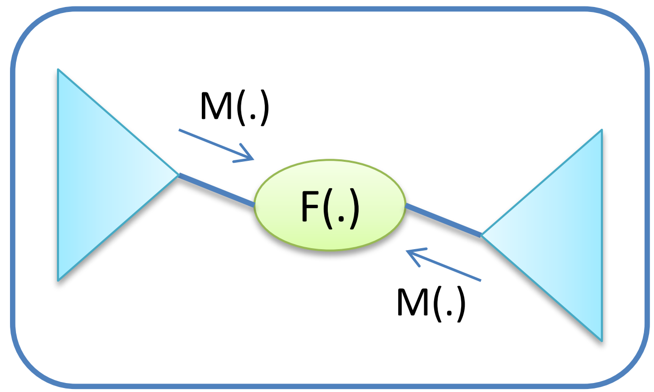


Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

If we want to know P(B):



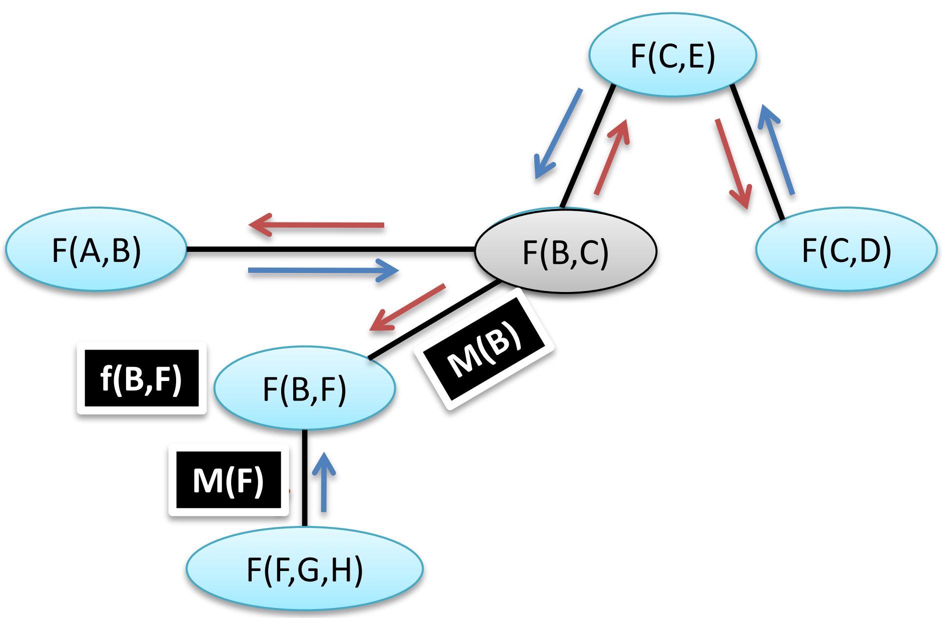


The queried node should get messages from all nodes on the tree to get the marginal distribution.

Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

To get marginal distribution of N nodes, we don't need run VE "N times", "2 times" are enough to get all possible messages.



$$P(B,F) = M(B) f(B,F) M(F)$$

1st pass:
Take a node as root.
Run Sum-Product from leaves to root.

2nd pass:
Run Sum-Product from root to leaves.

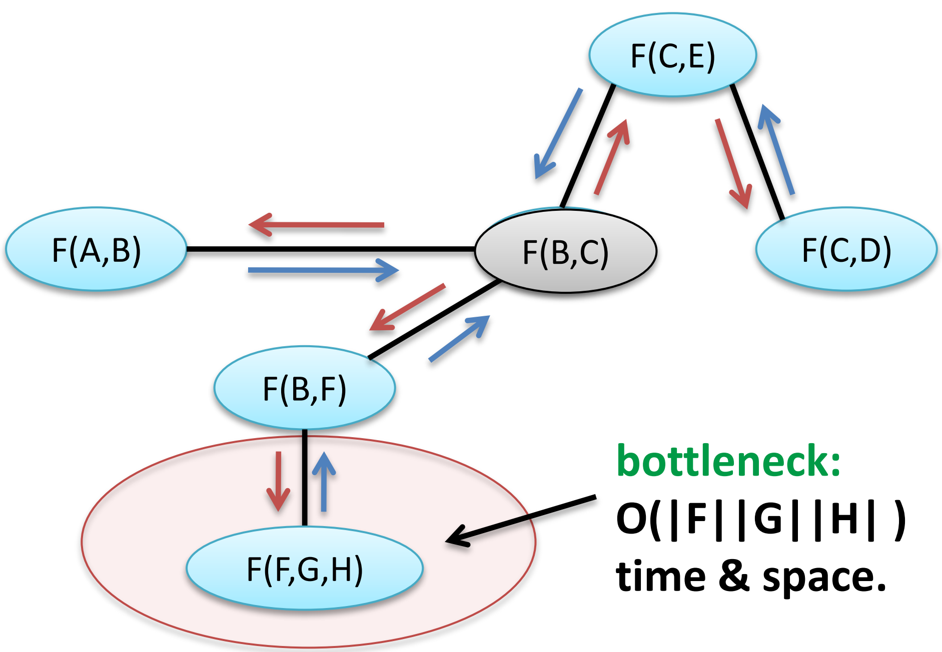
All marginal dist. can be derived from

1. Multiply all M(.) from neighbors by f(.) on this node.
2. Eliminate unwanted variables.

Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

To get marginal distribution of N nodes, we don't need run VE "N times", "2 times" are enough to get all possible messages.



Complexity :

Elimination on a node F(A,B,C) takes O(|A||B||C|) space & time.

So the algorithm's bottleneck is on elimination for the "**Largest Node**" on clique tree.

Agenda

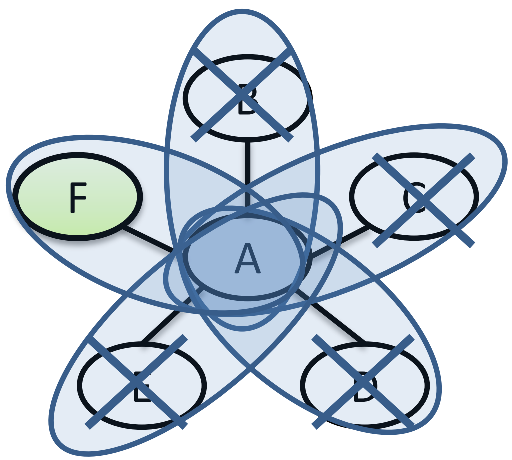
- Introduce the concept of "Variable Elimination" in special case of Tree-structured Factor Graph.
- **Extend the idea of "VE" to General Factor Graph with concept of "Clique Tree".**
- See how to extend "VE" to "Most Probable Assignment" (MAP configuration) Problem.

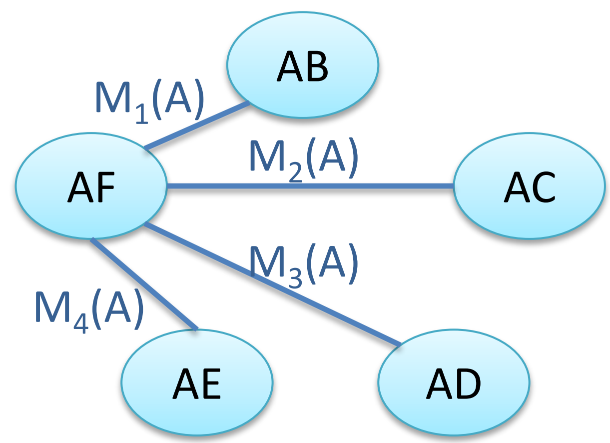
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 1: B C D E A





$$P(F)$$

$$= \frac{1}{Z} \sum_A \sum_E \sum_D \sum_C \sum_B f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

$$= \frac{1}{Z} \sum_A f(A,F) \sum_E f(A,E) \sum_D f(A,D) \sum_C f(A,C) \sum_B f(A,B)$$

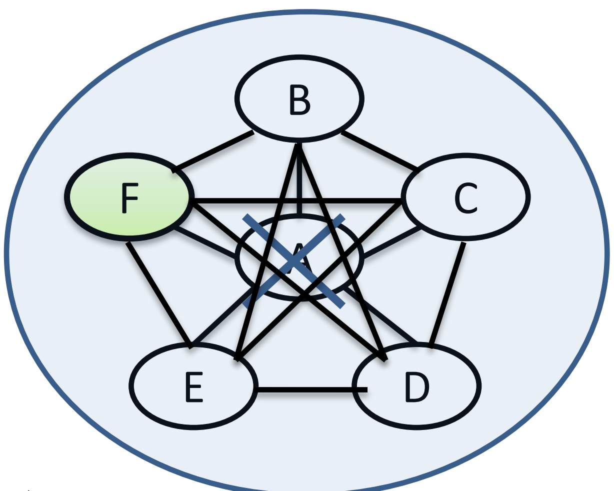
Maximum Node Size=2

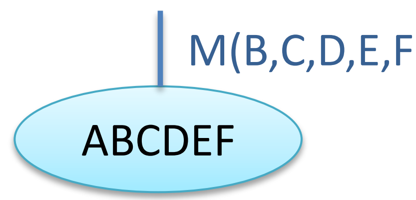
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





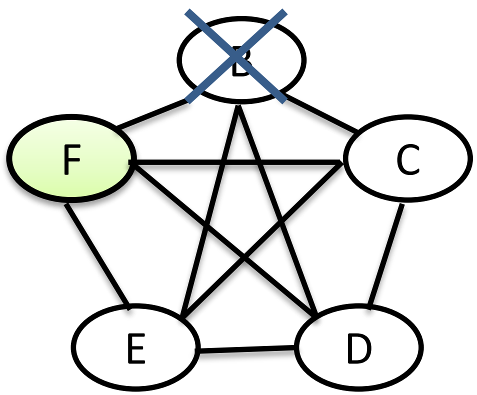
$$P(F) = \frac{1}{Z}\sum_{E}\sum_{D}\sum_{C}\sum_{B}\underbrace{\sum_{A} f(A,F)\,f(A,E)\,f(A,D)\,f(A,C)\,f(A,B)}_{M(B,C,D,E,F)}$$

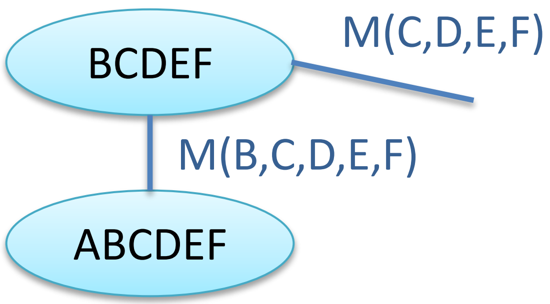
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





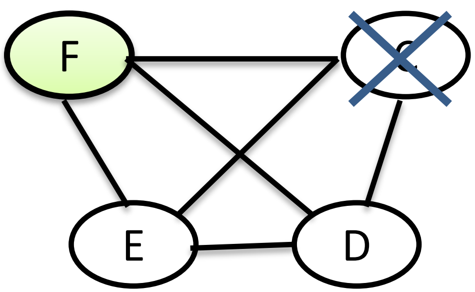
$$P(F) = \frac{1}{Z}\sum_E\sum_D\sum_C\sum_B\sum_A \underbrace{f(A,F)\,f(A,E)\,f(A,D)\,f(A,C)\,f(A,B)}_{M(C,D,E,F)}$$

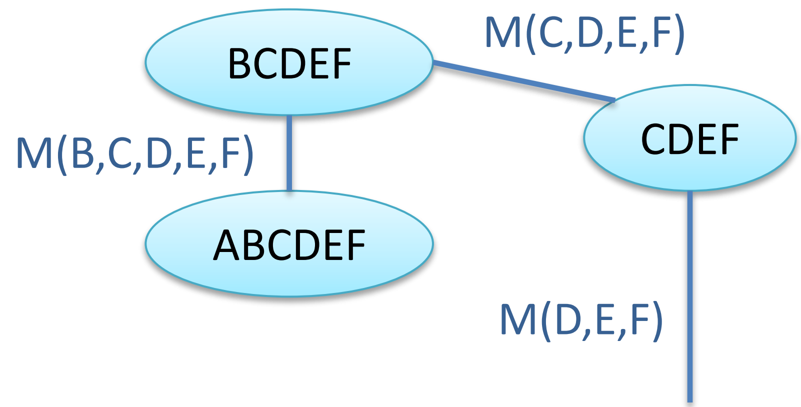
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F)$$

$$= \frac{1}{Z} \sum_E \sum_D \underline{\sum_C \sum_B \sum_A f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)}$$

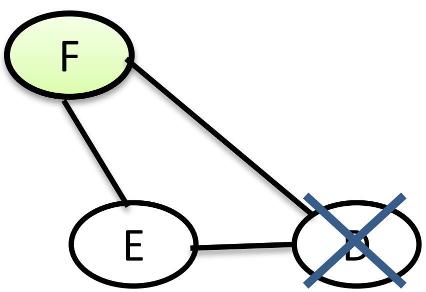
M(D,E,F)

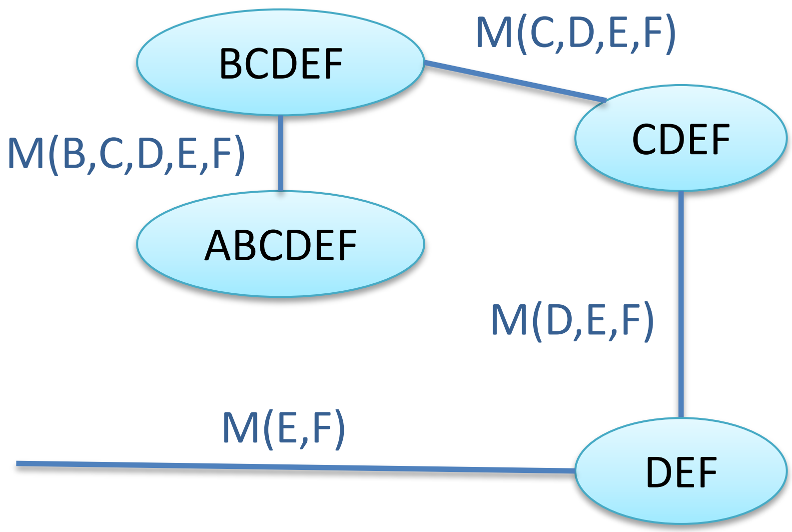
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F) = \frac{1}{Z}\underbrace{\sum_E\sum_D\sum_C\sum_B\sum_A f(A,F)\,f(A,E)\,f(A,D)\,f(A,C)\,f(A,B)}_{M(E,F)}$$

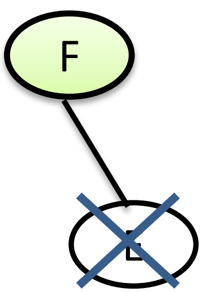
M(E,F)

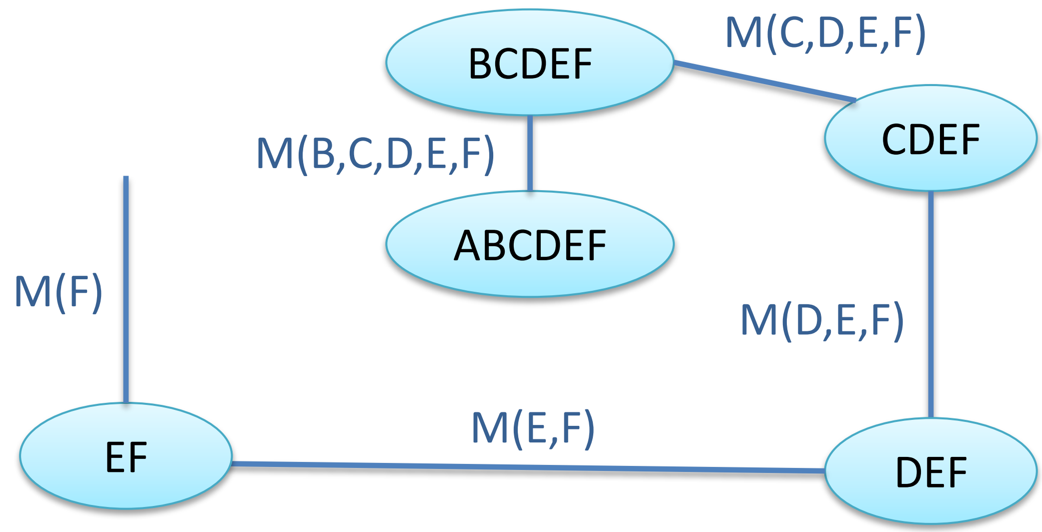
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





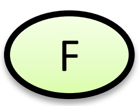
$$P(F) = \frac{1}{Z}\underbrace{\sum_E \sum_D \sum_C \sum_B \sum_A f(A,F)\, f(A,E)\, f(A,D)\, f(A,C)\, f(A,B)}_{M(F)}$$

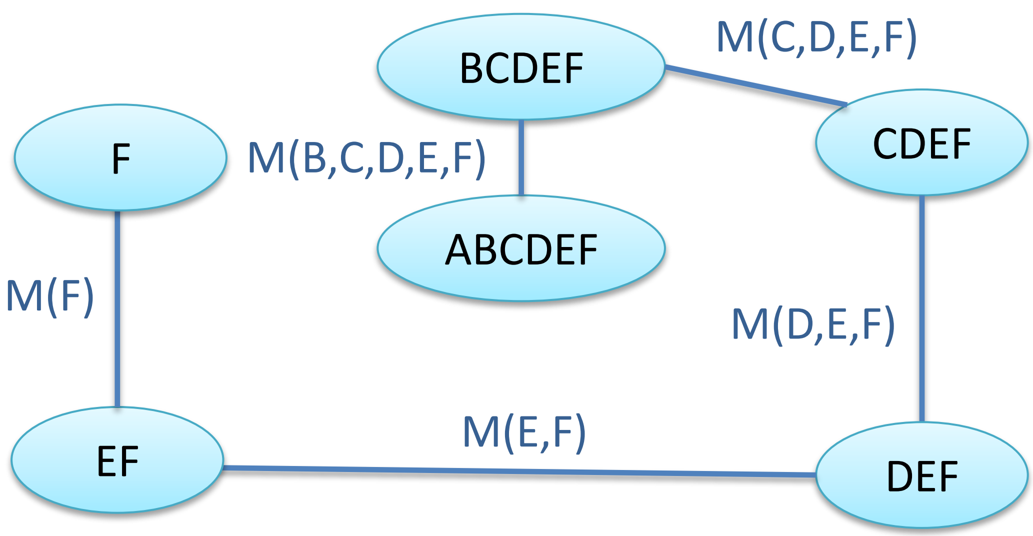
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F) = \frac{1}{Z}\sum_{E}\sum_{D}\sum_{C}\sum_{B}\sum_{A} f(A,F)\, f(A,E)\, f(A,D)\, f(A,C)\, f(A,B)$$

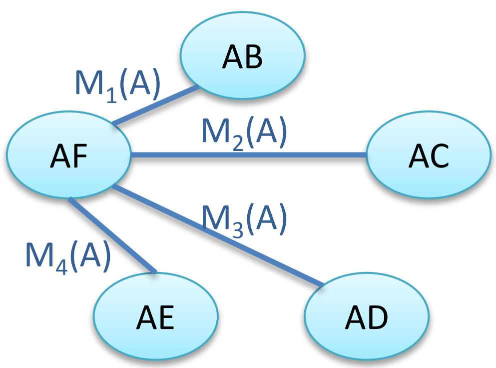
Maximum Node Size=6

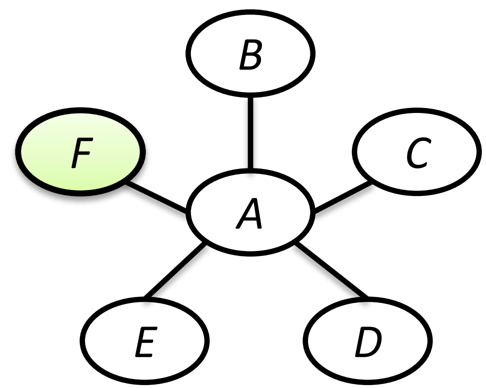
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 1: B C D E A





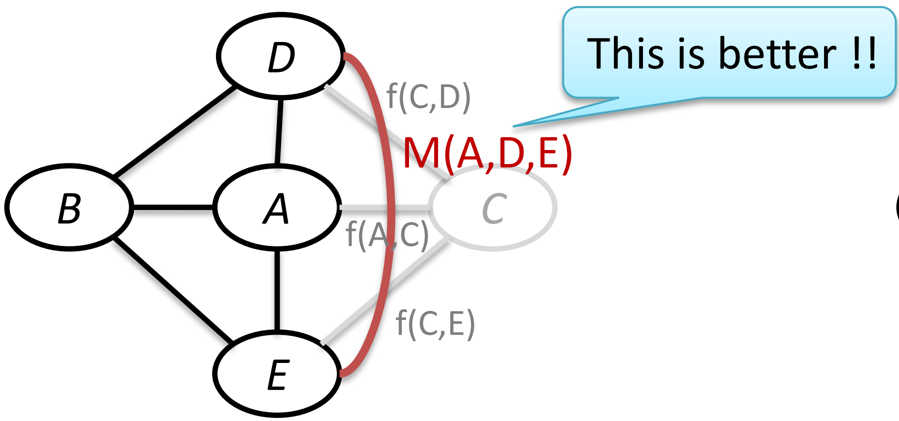
In "Tree" structure factor graph, the optimal "Elimination Order" is just "Elimination from leaves".

If factor graph is not Tree, what's the best elimination order ???

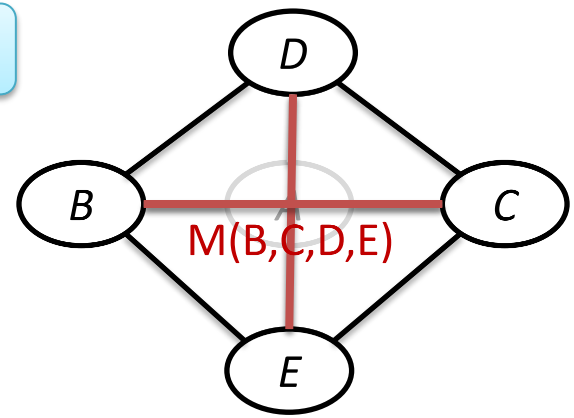
Variable Elimination: Inference on General Graph

When factor graph is not Tree, we want a Elimination Order "introducing as fewer edges as possible" (then we will have factor size smaller).

Eliminate "C" ➔ fill 1 edge



Eliminate "A" ➔ fill 2 edges



$$M(A,D,E) = \sum_C f(A,C)\, f(C,D)\, f(C,E)$$

Produce a factor of size 3

$$M(B,C,D,E) = \sum_A f(A,B)\, f(A,C)\, f(A,D)\, f(A,E)$$

Produce a factor of size 4

Variable Elimination: Inference on General Graph

Unfortunately, Finding Elimination order with "smallest maximum factor" is NP-hard.

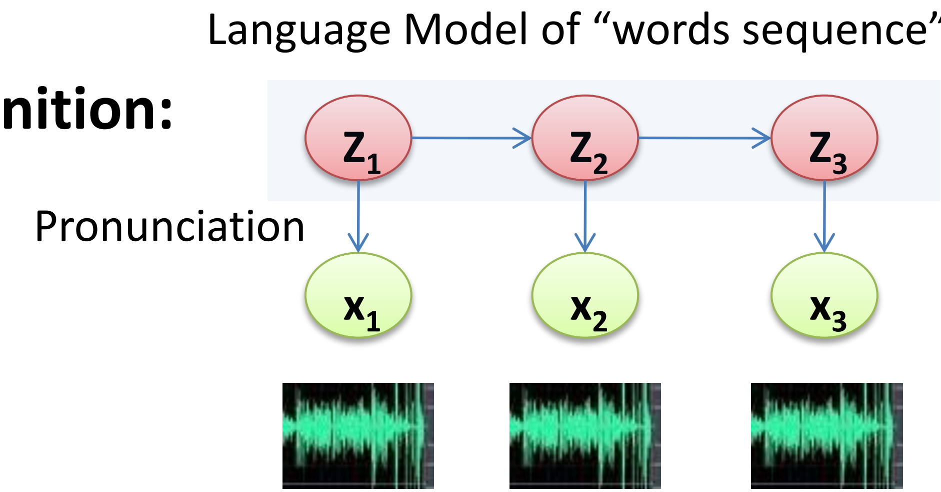
It's fortunate that greedy algorithm works quite well in practical, in which, we just search for the "least-cost" variable to eliminate:

1. If variables have same cardinality
   ➔ cost = (# of edges introduced by elimination).

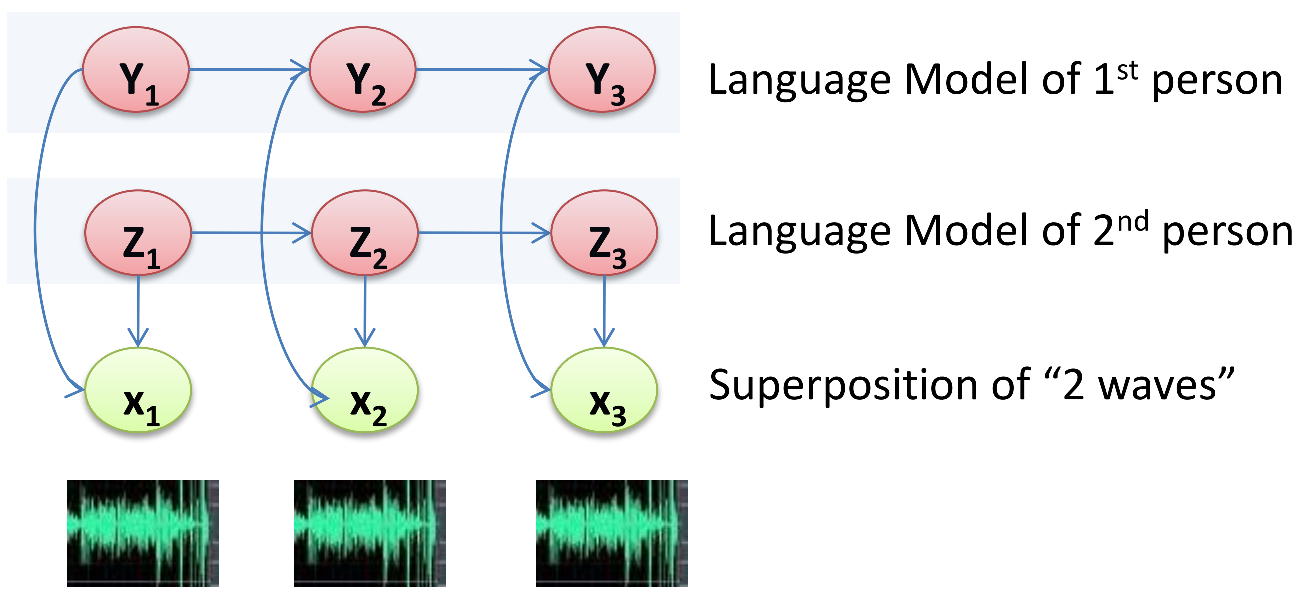
2. If variables have different cardinality
   ➔ cost = (# of edges)*(weight by cardinality of node involved)

Example: Factorial HMM

Speech Recognition:

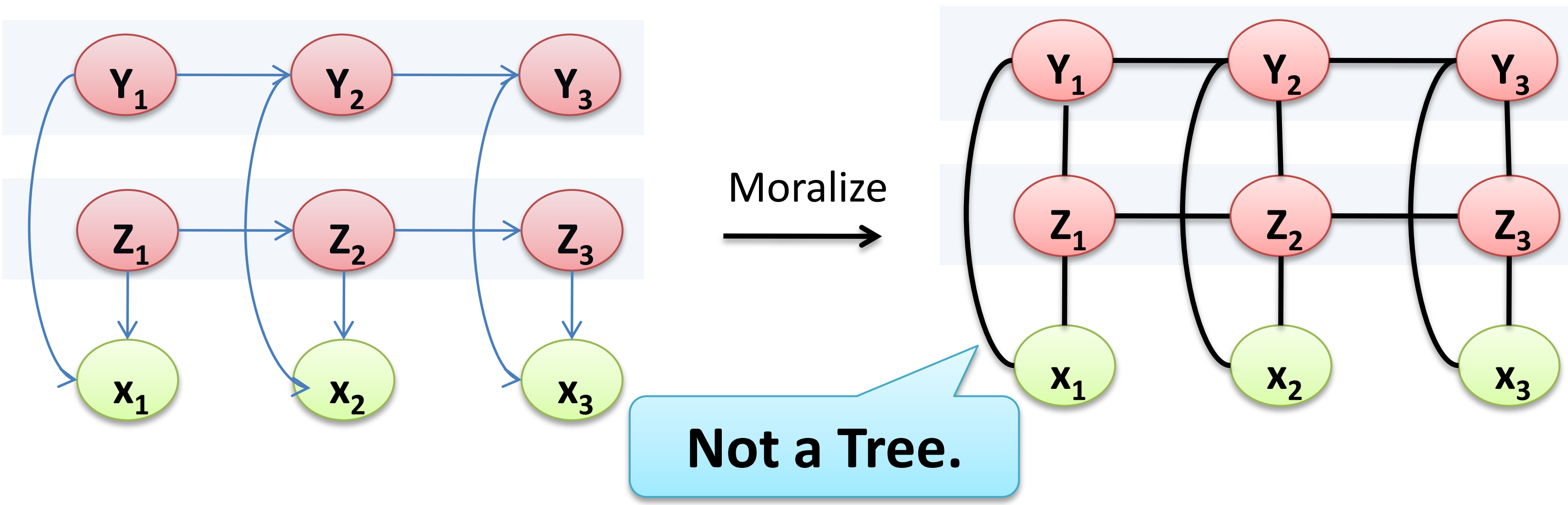


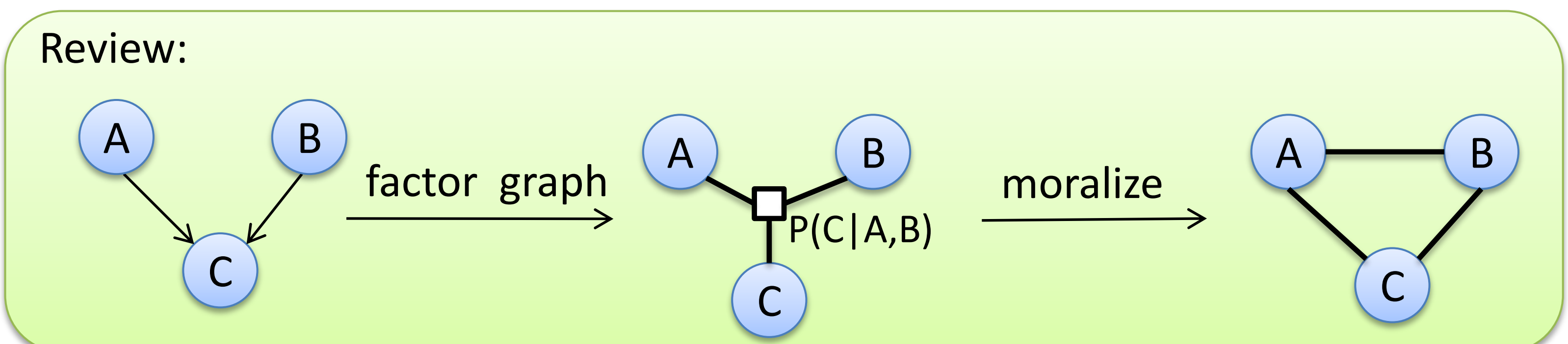
Decoding 2 person's speech from waves:



Example: Factorial HMM

Because a factor is a "clique" in undirected representation, we transform Factorial HMM into "undirected" before running VE.

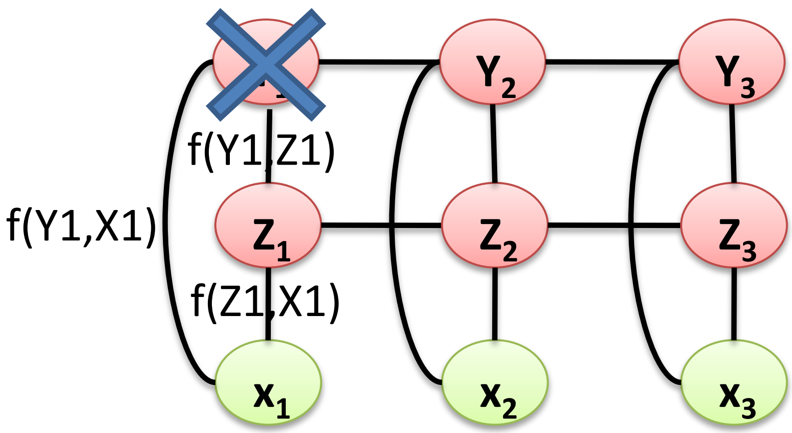




Example: Factorial HMM

Finding Elimination Order:

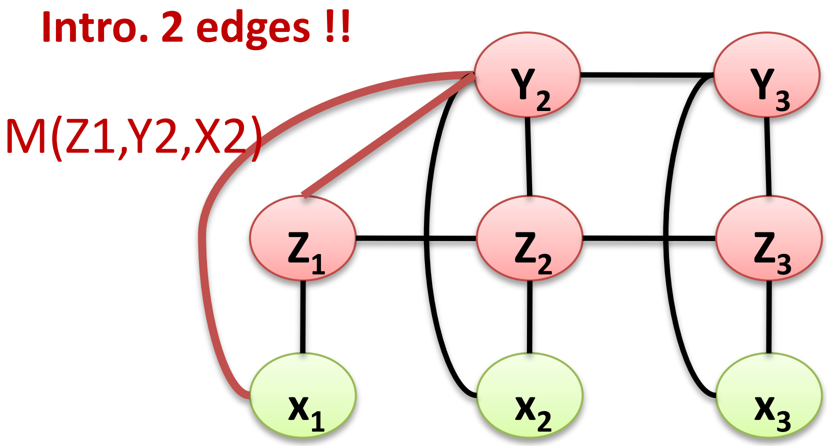
Find elimination adding as fewer edges as possible. (greedily)



$$M(Z_1, Y_2, X_1) = \sum_{Y_1} f(Y_1, Z_1) f(Y_1, Y_2) f(Y_1, X_1)$$

Example: Factorial HMM

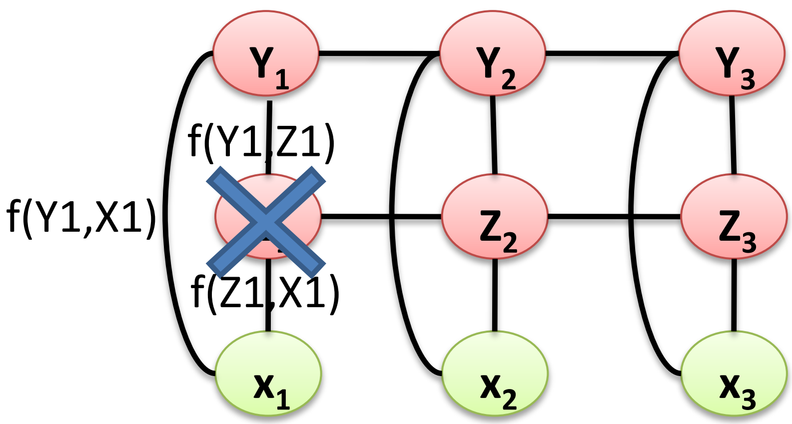
Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)



$$M(Z_1, Y_2, X_1) = \sum_{Y_1} f(Y_1, Z_1) f(Y_1, Y_2) f(Y_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

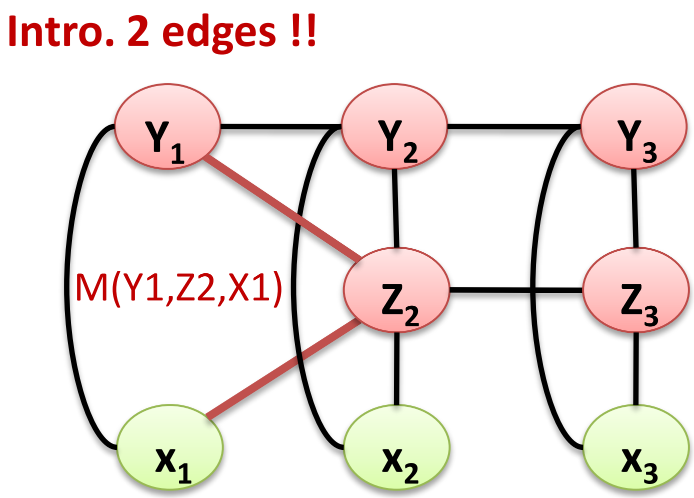


$$M(Y_1, Z_2, X_1) = \sum_{Z_1} f(Y_1, Z_1) f(Z_1, Z_2) f(Z_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)

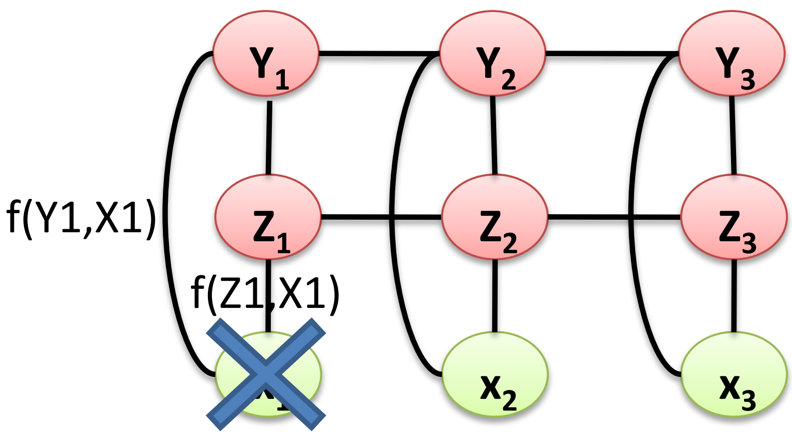


$$M(Y_1, Z_2, X_1) = \sum_{Z_1} f(Y_1, Z_1) f(Z_1, Z_2) f(Z_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



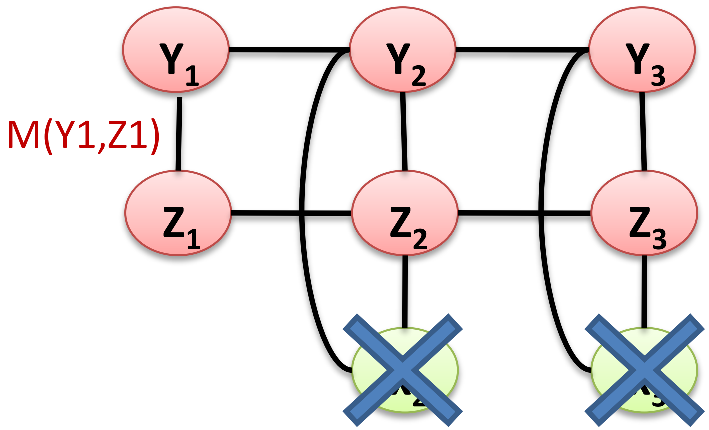
$$M(Y_1, Z_1) = \sum_{X_1} f(Y_1, X_1) f(Z_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)

Intro. no edges !! ➔ Let's eliminate !!

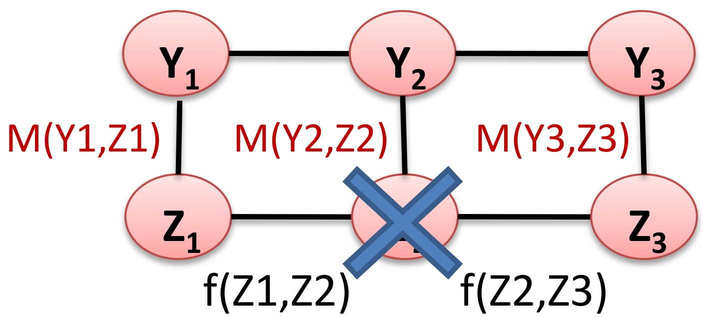


$$M(Y_1, Z_1) = \sum_{X_1} f(Y_1, X_1) f(Z_1, X_1)$$

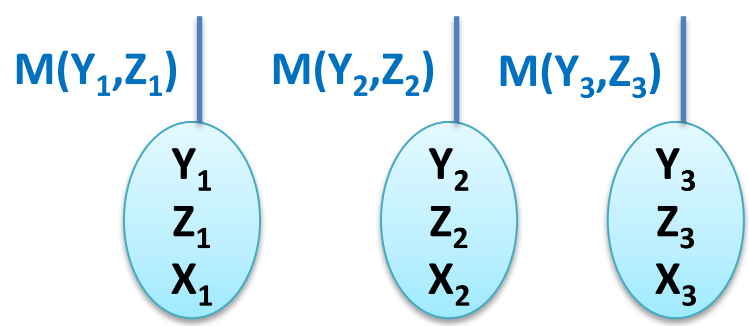
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



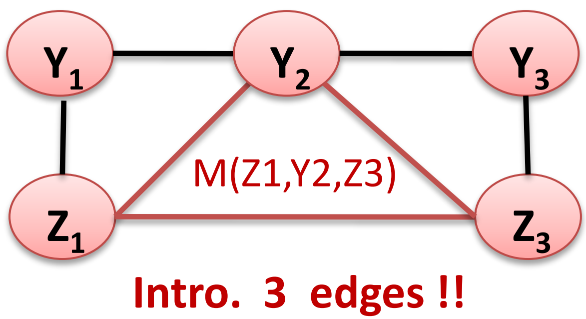
$$M(Z_1, Y_2, Z_3) = \sum_{Z_2} M(Y_2, Z_2)\, f(Z_1, Z_2)\, f(Z_2, Z_3)$$



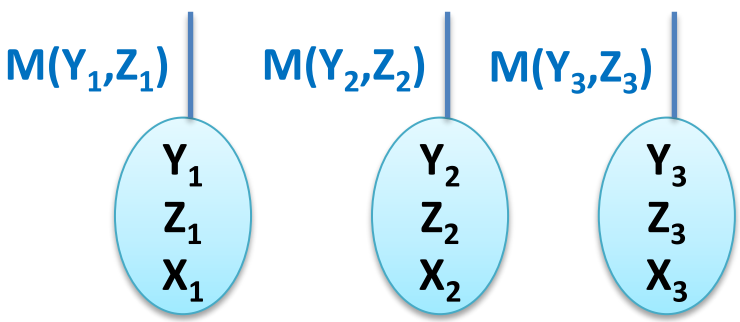
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)

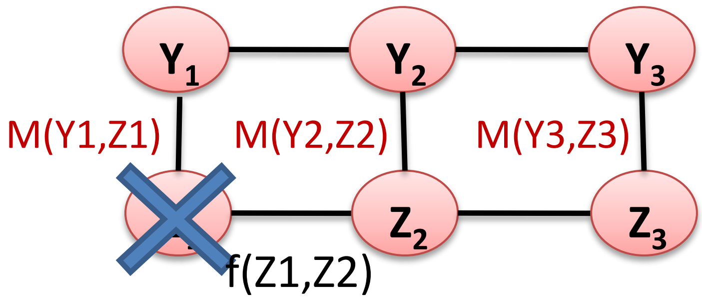


$$M(Z_1, Y_2, Z_3) = \sum_{Z_2} M(Y_2, Z_2) f(Z_1, Z_2) f(Z_2, Z_3)$$

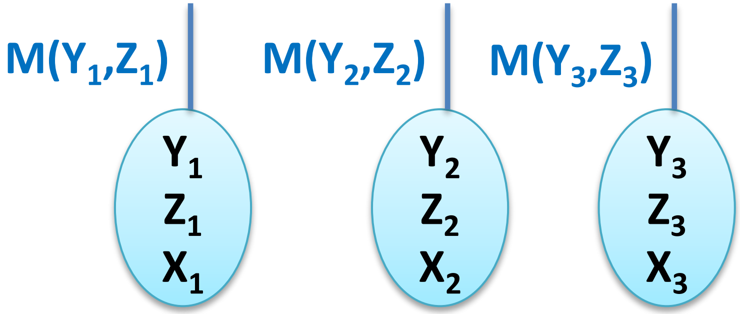


Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)



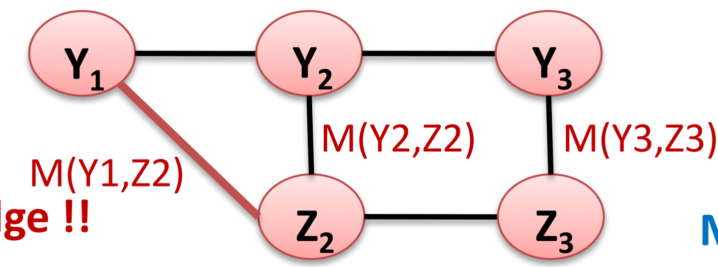
$$M(Y_1, Z_2) = \sum_{Z_1} M(Y_1, Z_1) f(Z_1, Z_2)$$



Example: Factorial HMM

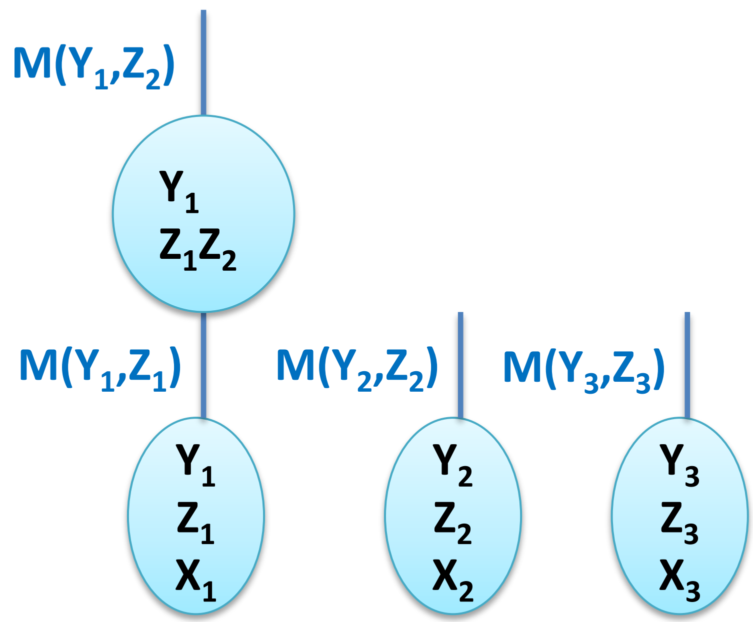
Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



Intro. 1 edge !!
(the best)

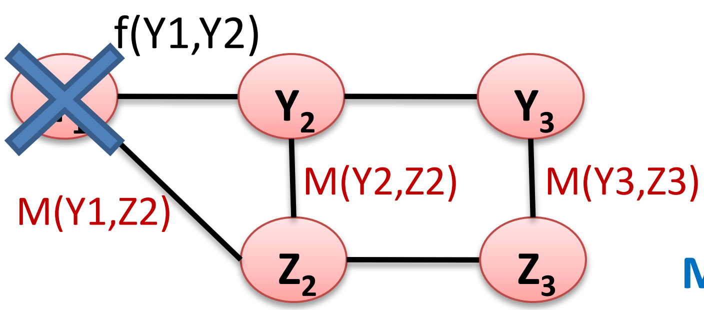
$$M(Y_1,Z_2)=\sum_{Z_1} M(Y_1,Z_1)\, f(Z_1,Z_2)$$



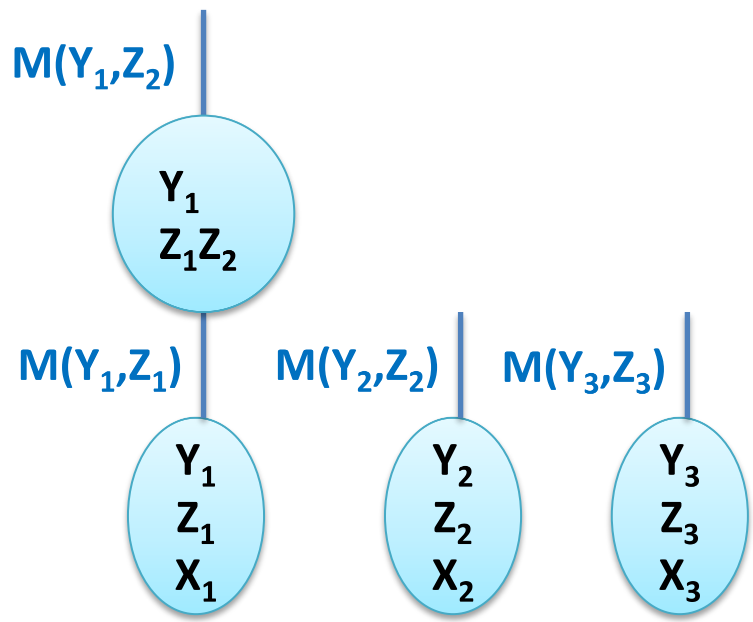
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



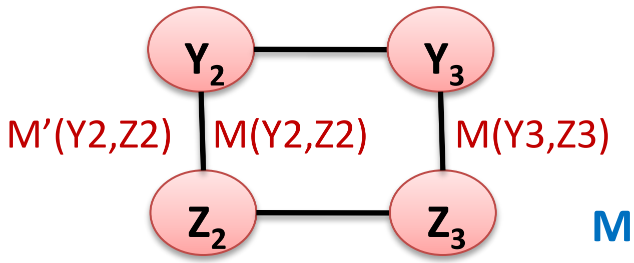
$$M'(Y_2, Z_2) = \sum_{Y_1} M(Y_1, Z_2)\, f(Y_1, Y_2)$$



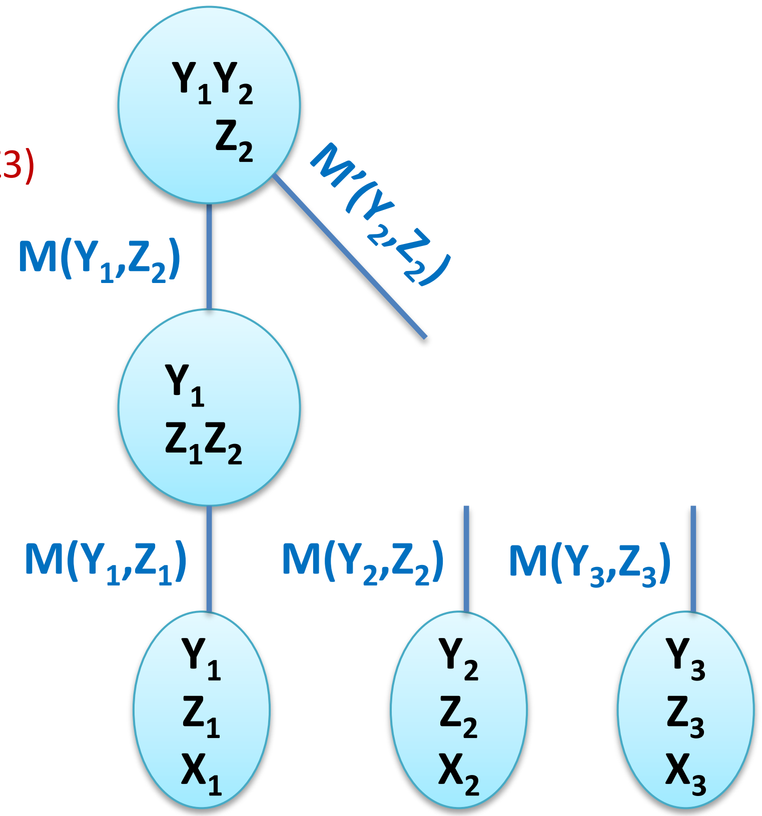
Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

Intro. 0 edge !!
(the best)



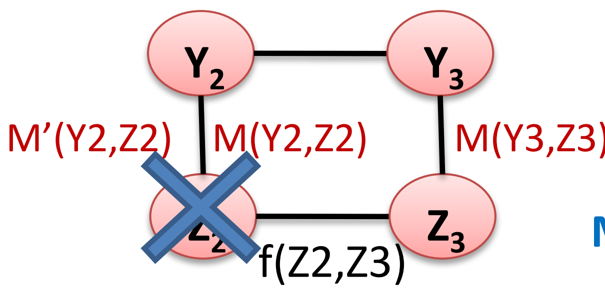
$$M'(Y_2, Z_2) = \sum_{Y_1} M(Y_1, Z_2)\, f(Y_1, Y_2)$$



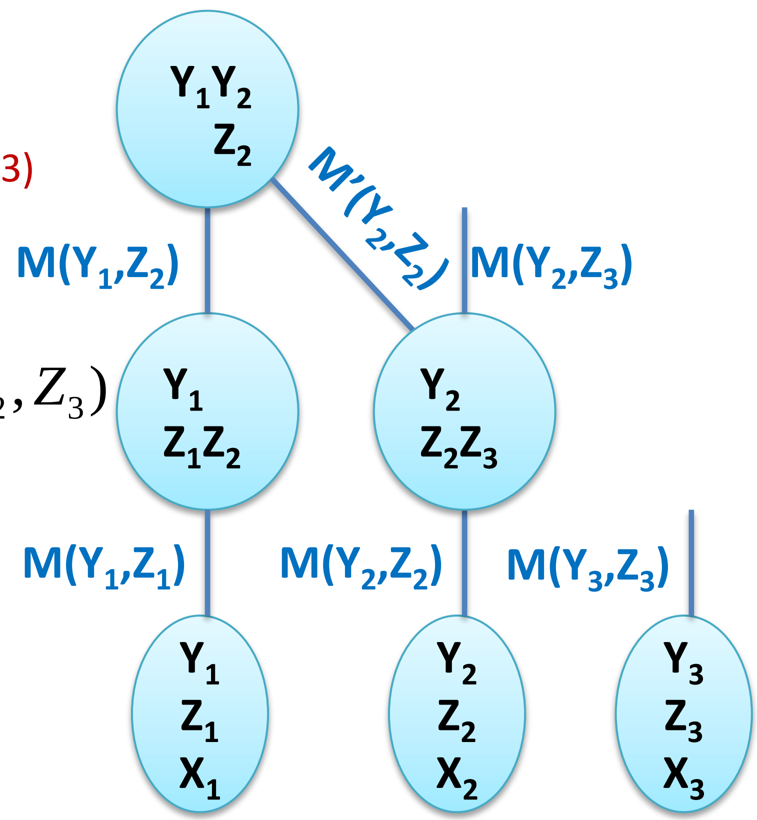
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



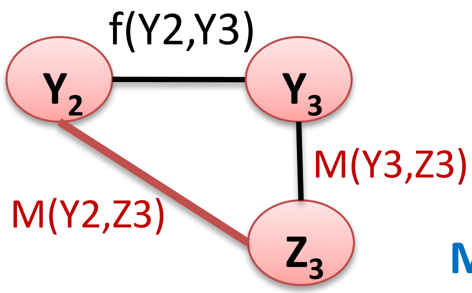
$$M(Y_2, Z_3) = \sum_{Z_2} M'(Y_2, Z_2) M(Y_2, Z_2) f(Z_2, Z_3)$$



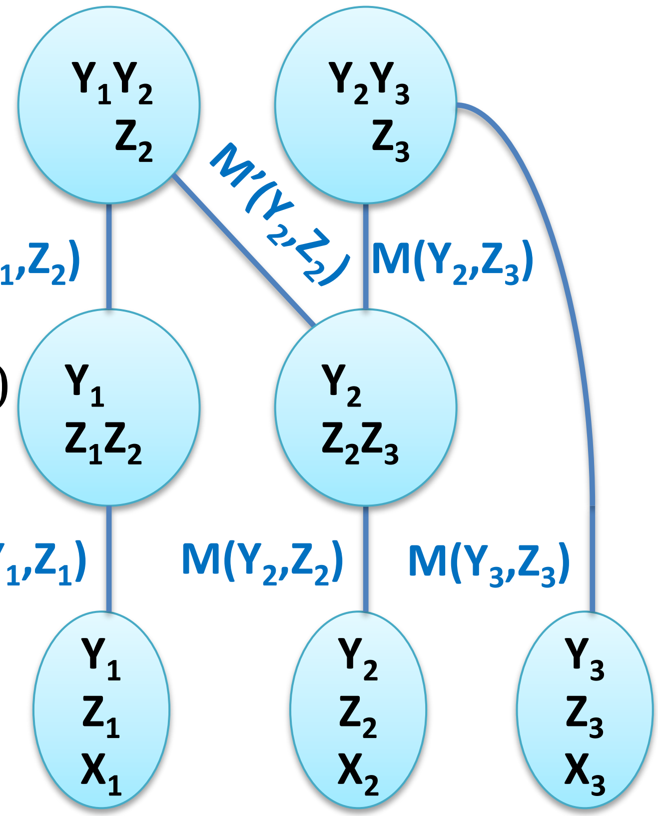
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



$$M(Y_2, Z_3) = \sum_{Z_2} M'(Y_2, Z_2) M(Y_2, Z_2) f(Z_2, Z_3)$$



Example: Factorial HMM

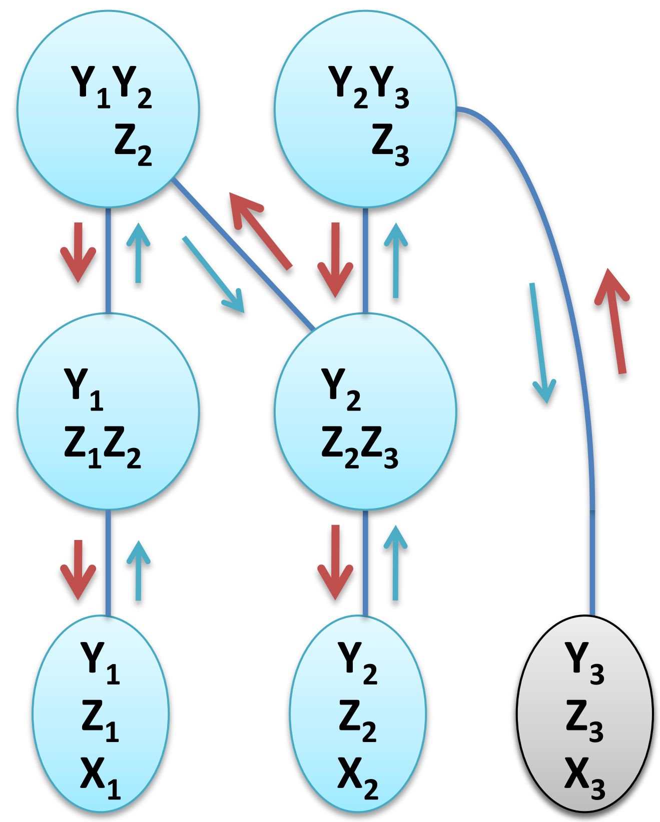
After building a clique tree, we can run "2 passes" on the tree to get all messages M(.) needed for computing marginal.

1st pass:
Take a node as root.
Run Sum-Product from leaves to root.

2nd pass:
Run Sum-Product from root to leaves.

All marginal dist. can be derived from

1. Multiply all M(.) from neighbors by f(.) on this node.
2. Eliminate unwanted variable.



Example: Factorial HMM

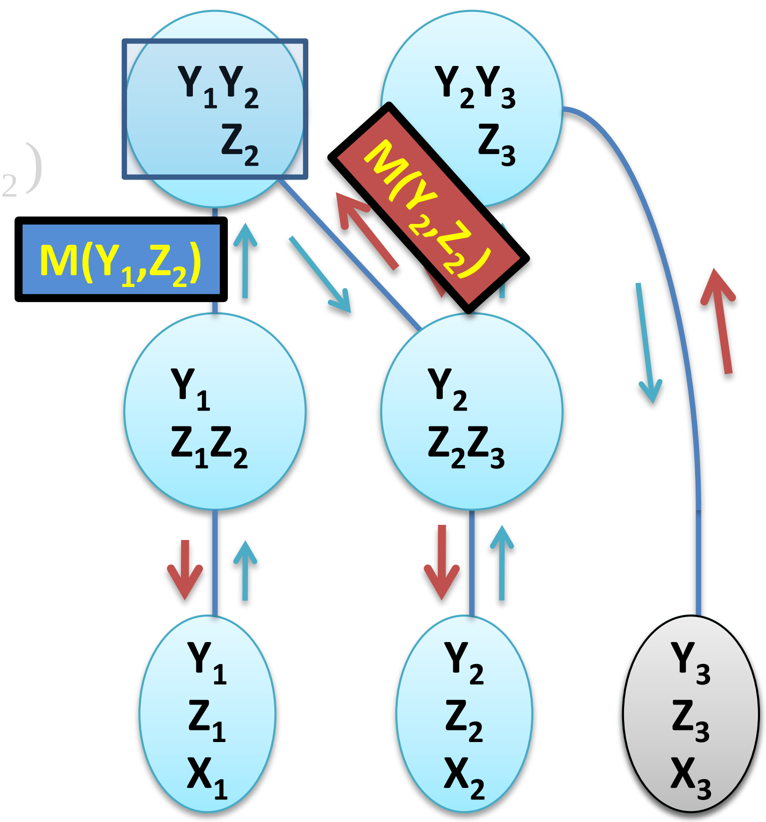
After building a clique tree, we can run "2 passes" on the tree to get all messages M(.) needed for computing marginal.

Assume we want : $P(Y_1,Y_2)$

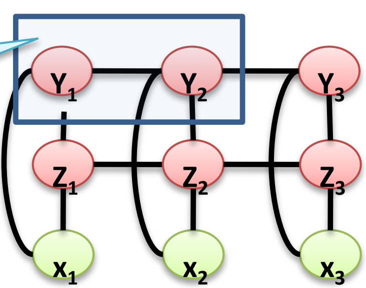
belief (or maginal dist.)

$$b(Y_1,Y_2,Z_2) = M(Y_1,Z_2)\,f(Y_1,Y_2,Z_2)\,M(Y_2,Z_2)$$

$$P(Y_1,Y_2) = \sum_{Z_2} b(Y_1,Y_2,Z_2)$$



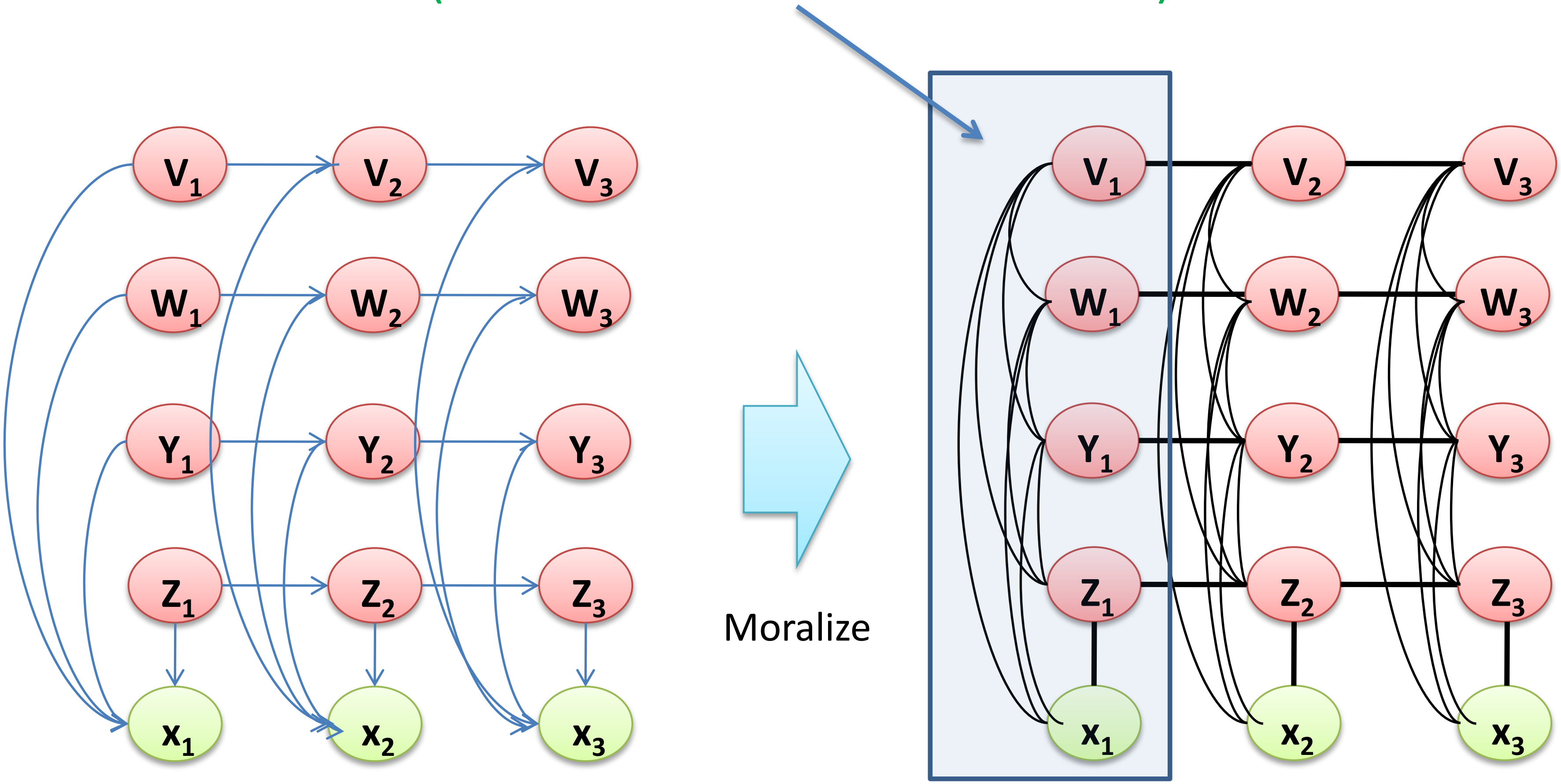
Y_1, Y_2 are **dependent.**



➔There **must be a node** in clique tree containing (Y_1,Y_2).

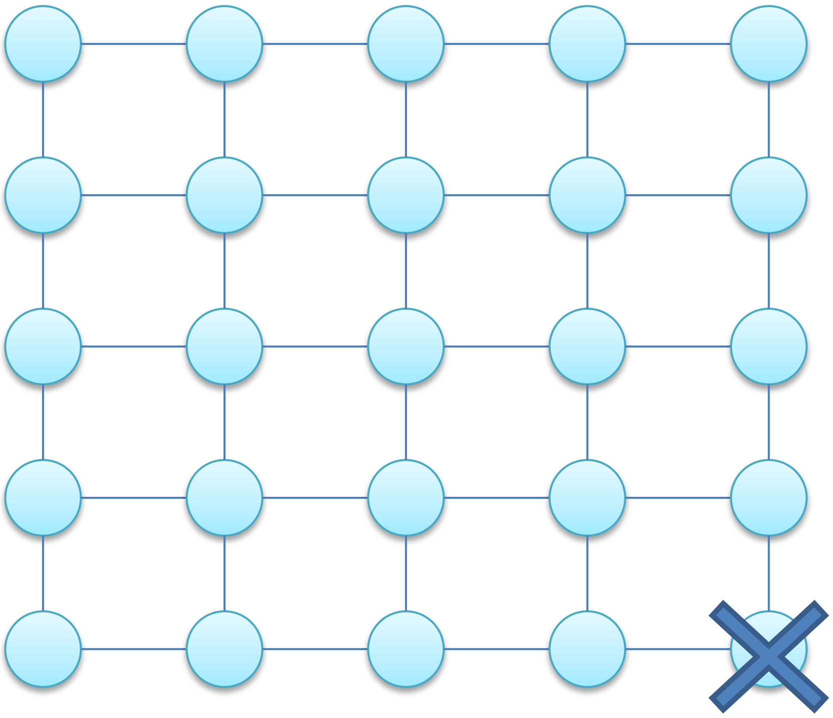
Example : General Factorial HMM

A clique size=5,
intractable most of times.
(No tractable elimination exist…)



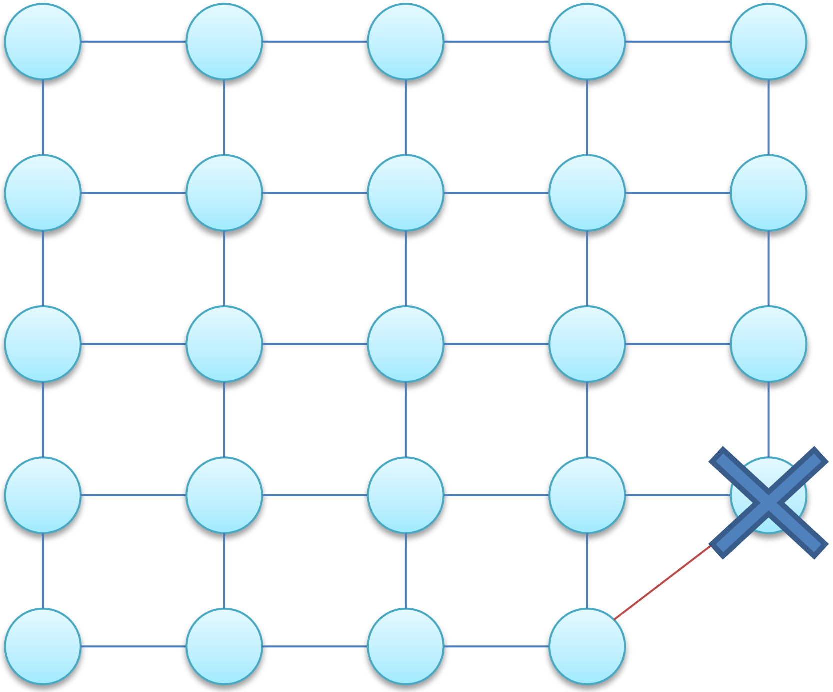
Some Model are Intractable for Exact Inference

Example: A Grid MRF



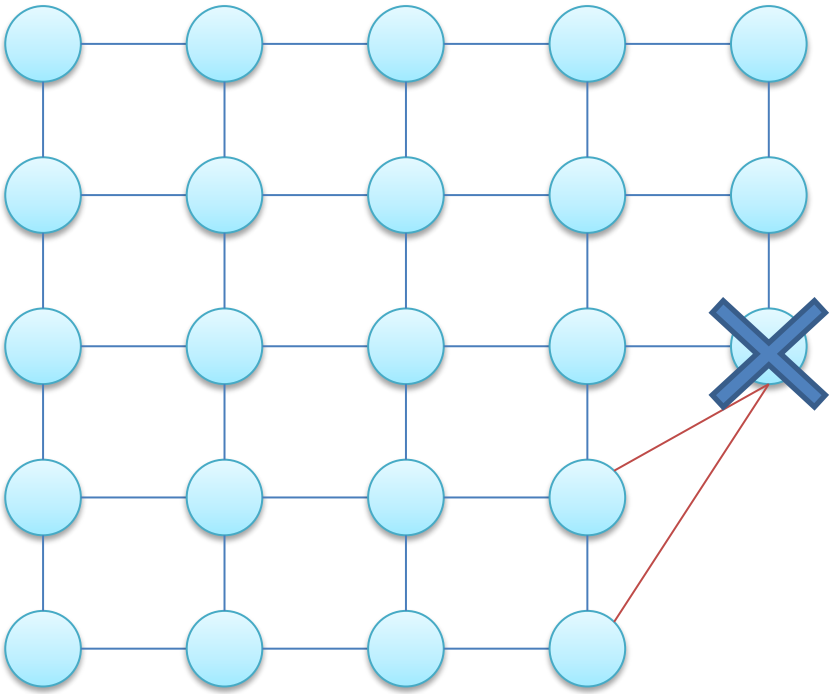
Some Model are Intractable for Exact Inference

Example: A Grid MRF



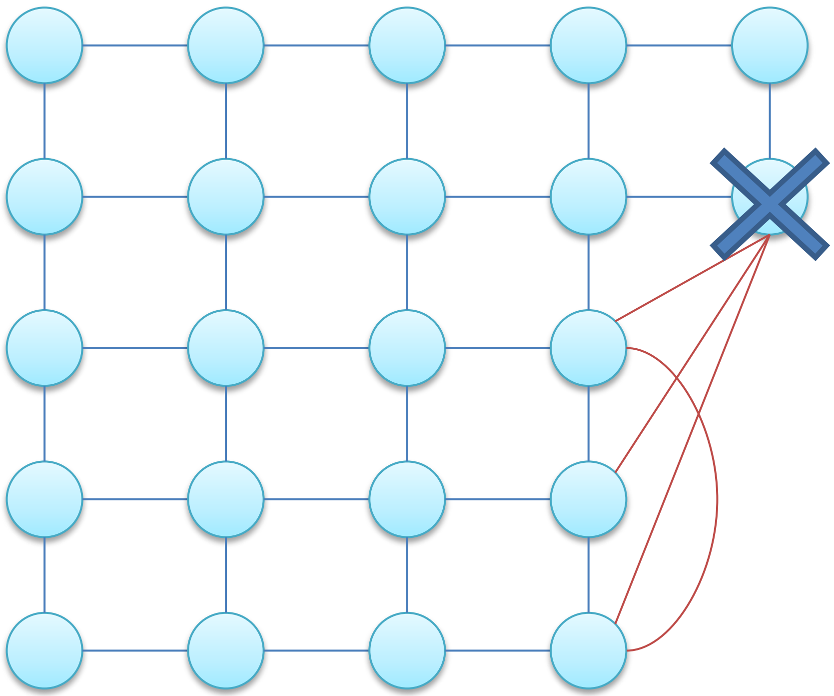
Some Model are Intractable for Exact Inference

Example: A Grid MRF



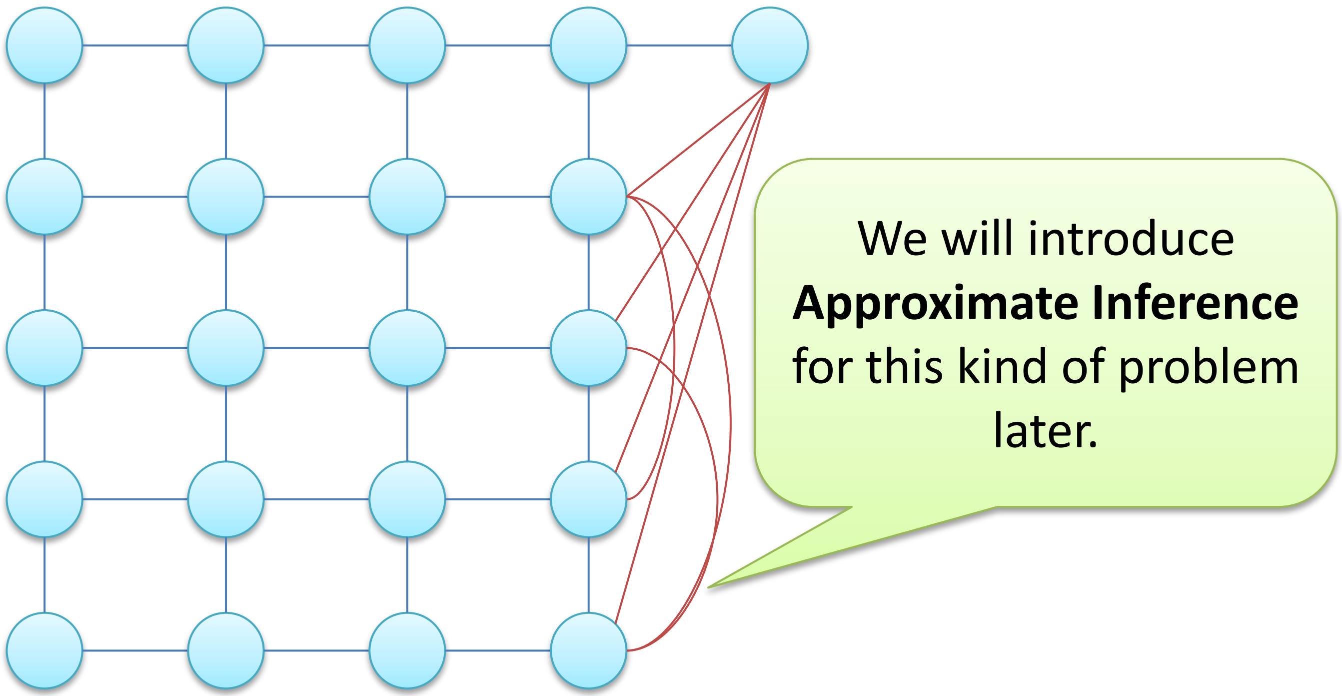
Some Model are Intractable for Exact Inference

Example: A Grid MRF



Some Model are Intractable for Exact Inference

Example: A Grid MRF

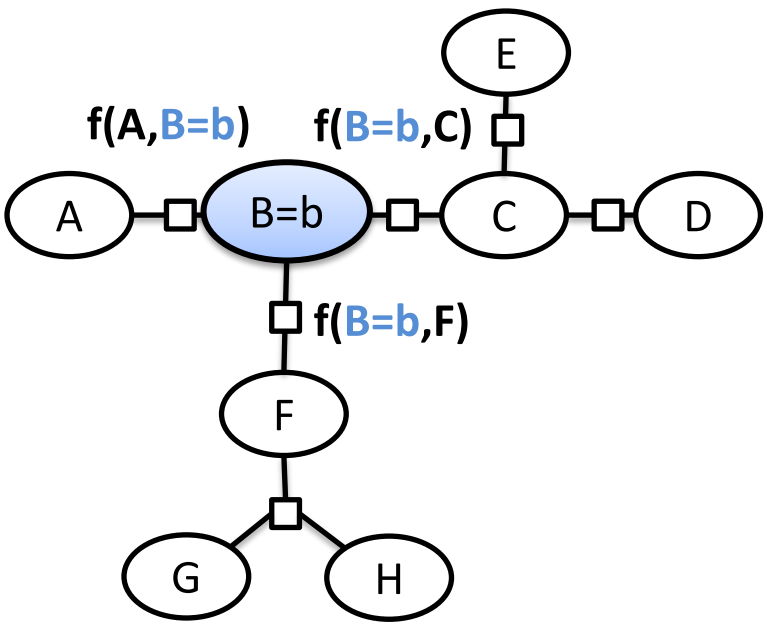


Generally, we will have clique of "size N" for a N*N grid, which is indeed intractable.

Variable Elimination: Dealing with Evidence

What if some variables $X=\{X_1 \ldots X_D\}$ are given in Evidence :

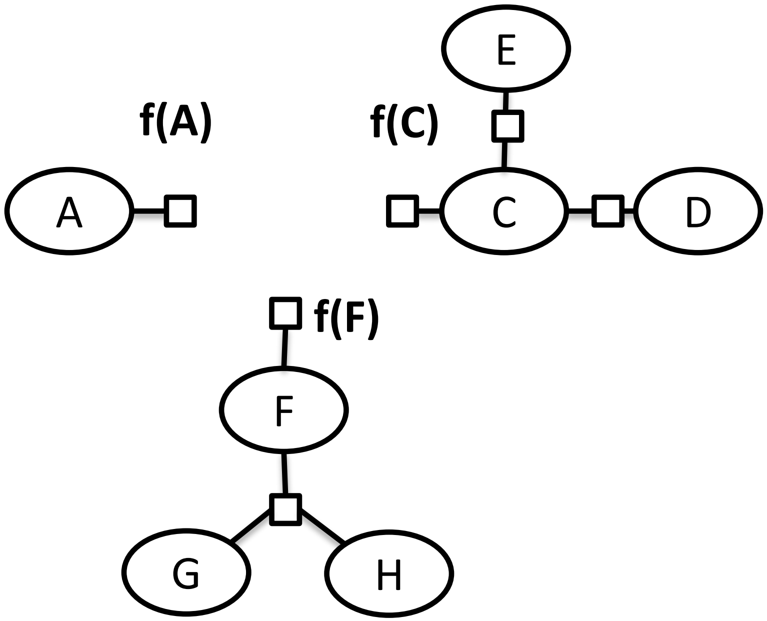
Given Evidence { B=b } :



Variable Elimination: Dealing with Evidence

What if some variables $X=\{X_1 \ldots X_D\}$ are given in Evidence :

Given Evidence { B=b } :



A model with evidence equivalent to another model without evidence.

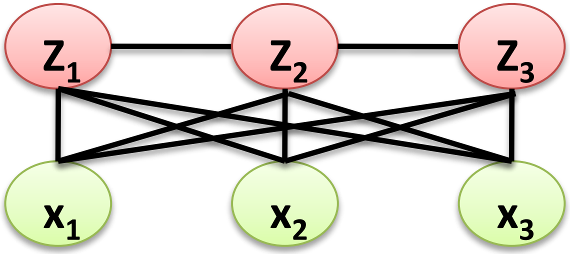
To infer $P_M(Z|X)$, we transform M to another model M' and infer $P_{M'}(Z)$.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(\cancel{X},Z)=\frac{1}{Z}f(Z_1,\cancel{X_1})f(Z_1,\cancel{X_2})f(Z_1,\cancel{X_3})$$
$$f(Z_2,\cancel{X_1})f(Z_2,\cancel{X_2})f(Z_2,\cancel{X_3})$$
$$f(Z_3,\cancel{X_1})f(Z_3,\cancel{X_2})f(Z_3,\cancel{X_3})$$
$$f(Z_1,Z_2)f(Z_2,Z_3)$$



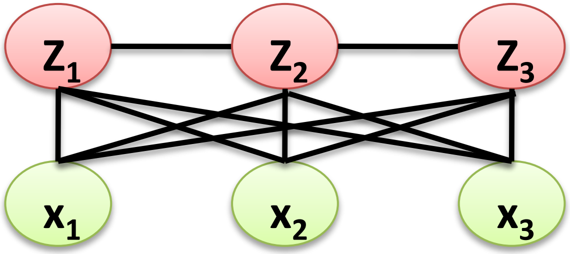
But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(X=x,Z)=\frac{1}{Z}\,f_x(Z_1)\,f_x{}'(Z_1)\,f_x{}''(Z_1)$$
$$f_x(Z_2)\,f_x{}'(Z_2)\,f_x{}''(Z_2)$$
$$f_x(Z_3)\,f_x{}'(Z_3)\,f_x{}''(Z_3)$$
$$f(Z_1,Z_2)\,f(Z_2,Z_3)$$



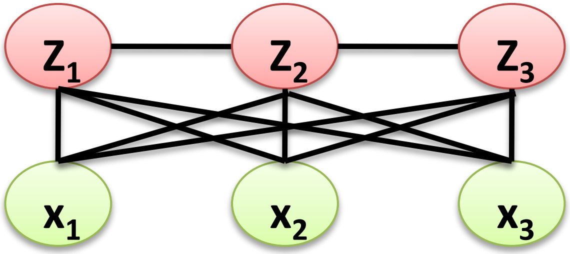
But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(X = x, Z) = \frac{1}{Z} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



But given X1~X3, we actually run inference on another model M'.

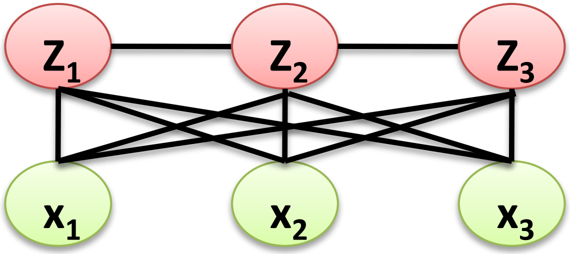
Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(Z \mid X = x) = \frac{P(X = x, Z)}{P(X = x)} = \frac{\frac{1}{Z} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}{\frac{1}{Z} \sum_{Z1} \sum_{Z2} \sum_{Z3} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}$$

$$= \frac{1}{Z'(x)} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



But given X1~X3, we actually run inference on another model M'.

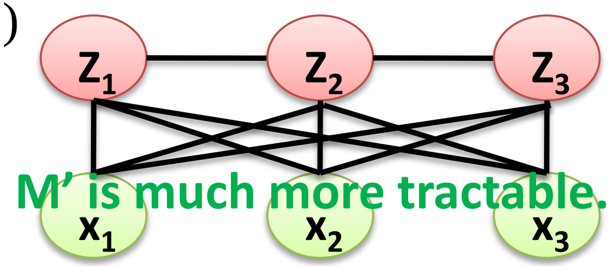
Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(Z \mid X = x) = \frac{P(X = x, Z)}{P(X = x)} = \frac{\cancel{\frac{1}{Z}} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}{\cancel{\frac{1}{Z}} \sum_{Z1} \sum_{Z2} \sum_{Z3} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}$$

$$= \frac{1}{Z'(x)} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



But given X1~X3, we actually run inference on another model M'.

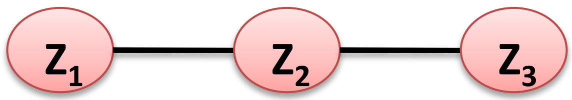
Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(Z \mid X = x) = \frac{P(X = x, Z)}{P(X = x)} = \frac{\cancel{\frac{1}{Z}} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}{\cancel{\frac{1}{Z}} \sum_{Z1} \sum_{Z2} \sum_{Z3} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}$$

$$= \frac{1}{Z'(x)} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



M' is much more tractable.

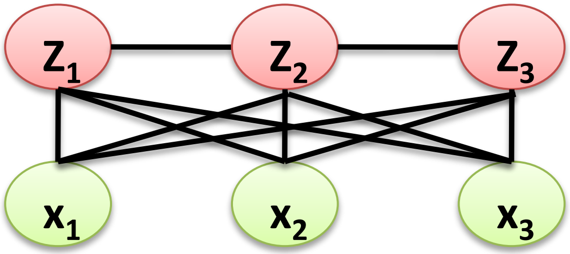
New normalize const. can be computed Using VE.

But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

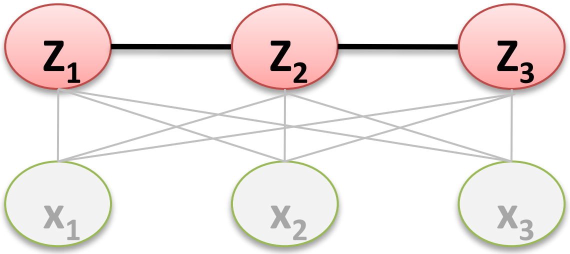
Even if P(Z|X) can be inferred efficiently, "learning P(X,Z)" is intractable.
One solution is model P(Z|X) directly , yielding "CRF" model.

$$P(X,Z) = \frac{1}{Z} f(Z_1,X_1) f(Z_1,X_2) f(Z_1,X_3)$$
$$f(Z_2,X_1) f(Z_2,X_2) f(Z_2,X_3)$$
$$f(Z_3,X_1) f(Z_3,X_2) f(Z_3,X_3)$$
$$f(Z_1,Z_2) f(Z_2,Z_3)$$



Intractable MRF model

$$P(Z|X) = \frac{1}{Z'(X)} f_X'(Z_1,Z_2) f_X'(Z_2,Z_3)$$



Tractable CRF Model

Agenda

- Introduce the concept of "Variable Elimination" in special case of Tree-structured Factor Graph.
- Extend the idea of "VE" to General Factor Graph with concept of "Clique Tree".
- **See how to extend "VE" to "Most Probable Assignment" (MAP configuration) Problem.**

Query 3: Most Probable Assignment

- Given Evidence E = {$X_1=x_1,\ldots, X_D=x_D$} and some other variables Z={$Z_1, \ldots, Z_k$} unspecified , Most Probable Assignment of Z is given by:

$$MPA(Z \mid X) = \arg\max_{Z} \; P(Z \mid X)$$

$$= \arg\max_{Z} \; \frac{P(X \mid Z)P(Z)}{\mathrm{P(X)}} = \arg\max_{Z} \; P(X \mid Z)P(Z)$$

$$\underset{Z}{\operatorname{argmax}} \; \mathrm{P(Z \mid X)} \neq \begin{cases} \arg\max_{Z1} P(Z_1 \mid X) \\ \ldots\ldots \\ \arg\max_{ZK} P(Z_K \mid X) \end{cases}$$

What's the different ?

MPA Goal:

$$\max_{Z} P(Z \mid X) = \max_{Z_1} \ldots \max_{Z_K} P(Z_1 \ldots Z_K \mid X)$$

Likelihood Goal::

(Solved using VE)

$$P(X) = \sum_{Z_1} \ldots \sum_{Z_K} P(Z_1 \ldots Z_K, X)$$

Exploring the similarity between "**max**" & "**∑**" is the key to solve MPA using VE.

What's the different ?

Review:

$$P(E=e)=\sum_D \sum_C \sum_B \sum_A P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$=\sum_D P(E \mid D)\sum_C P(D \mid C)\sum_B P(C \mid B)\underline{\sum_A P(B \mid A)P(A)}$$

M(B) : marginal of B

$$\max_{\text{A,B,C,D}} P(A,B,C,D,E=e)$$

$$=\max_D \max_C \max_B \max_A P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$=\max_D P(E=e \mid D)\max_C P(D \mid C)\max_B P(C \mid B)\underline{\max_A P(B \mid A)P(A)}$$

M'(B) : ???

$M(B) = max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_A F(A,B)$$

For every choice of B, we decide an $A^*(B) = \text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_{A} F(A,B)$$

For every choice of B, we decide an $A^*(B) = \text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

B	b1	b2	b3
A*(B)	a3		

B	b1	b2	b3
M(B)	4		

$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_A F(A,B)$$

For every choice of B, we decide an $A^*(B) = \text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

B	b1	b2	b3
A*(B)	a3	a3	

B	b1	b2	b3
M(B)	4	7	

M(B) = $\max_A$ F(A,B) : maxMarginal of B

$$M(B) = \max_{A} F(A,B)$$

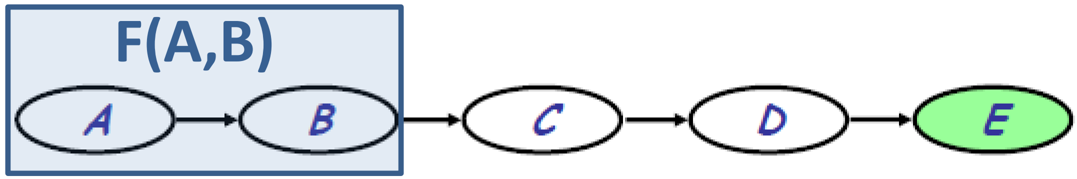
For every choice of B, we decide an A*(B)=argmax_A F(A,B) with M(B) = F(A*(B), B).

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

B	b1	b2	b3
A*(B)	a3	a3	a1

B	b1	b2	b3
M(B)	4	7	9

Most Probable Assignment on a Chain



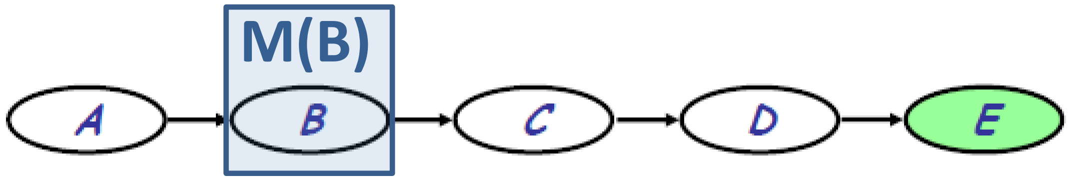
$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A \underbrace{P(B \mid A)P(A)}_{F(A,B)}$$

F(A,B)	a1	a2	a3
b1	...	...	...
b2	...	...	...
b3	...	...	...

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

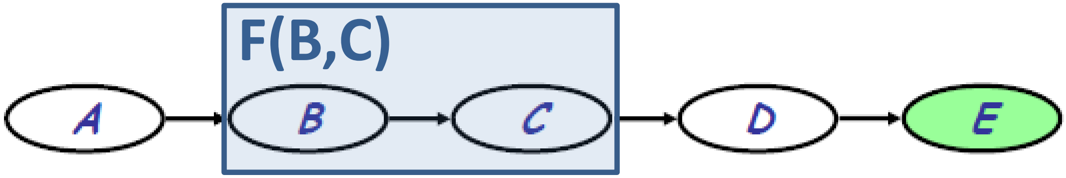
$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \underline{\max_A P(B \mid A)P(A)}$$

M(B) = max$_A$ F(A,B)

B	A*(B)	M(B)
b1	a1	...
b2	a3	...
b3	a2	...

F(A,B)	a1	a2	a3
b1	...	...	...
b2	...	...	...
b3	...	...	...

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A, B, C, D, E = e)$$

$$= \max_D \max_C \max_B \max_A P(E = e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

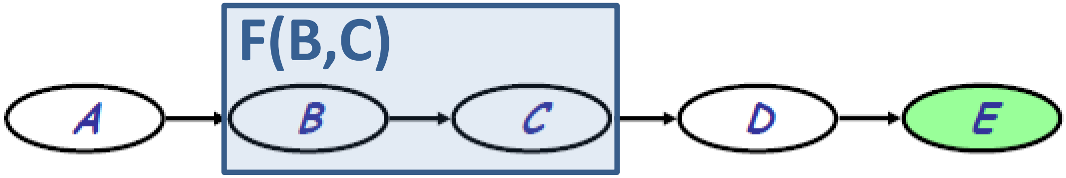
$$= \max_D P(E = e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \underline{\max_A P(B \mid A) P(A)}$$

F(B,C)=P(C|B)M(B)

P(C|B)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A, B, C, D, E = e)$$

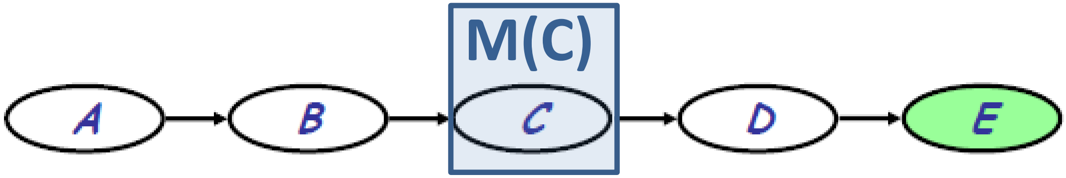
$$= \max_D \max_C \max_B \max_A P(E = e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

$$= \max_D P(E = e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)$$

F(B,C)=P(C|B)M(B)

F(B,C)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

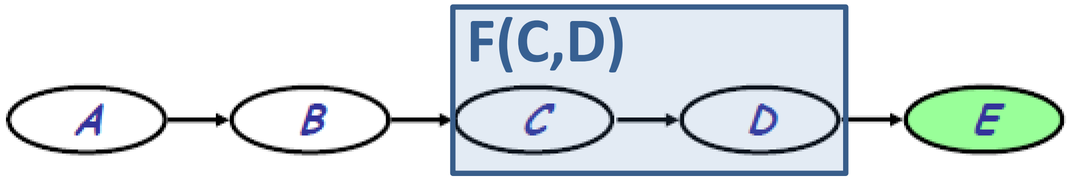
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A) P(A)$$

M(C) = $\max_B$ F(B,C)

C	B*(C)	M(C)
c1	b3	...
c2	b1	...
c3	b2	...

F(B,C)	b1	b2	b3
c1	...	...	...
c2	...	...	...
c3	...	...	...

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_D \max_C \max_B \max_A P(E=e\,|\,D)P(D\,|\,C)P(C\,|\,B)P(B\,|\,A)P(A)$$

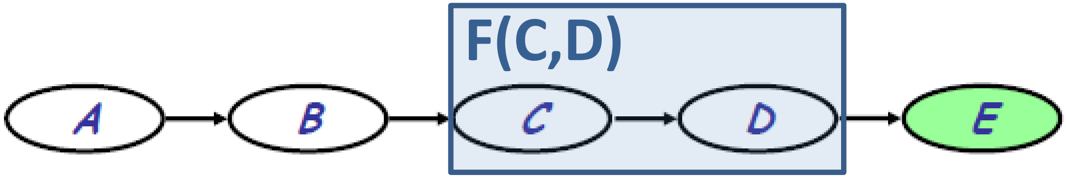
$$= \max_D P(E=e\,|\,D) \max_C P(D\,|\,C) \max_B P(C\,|\,B) \max_A P(B\,|\,A)P(A)$$

F(C,D) = P(D|C)M(C)

P(D|C)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

C	B*(C)	M(C)
c1	b3	…
c2	b1	…
c3	b2	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

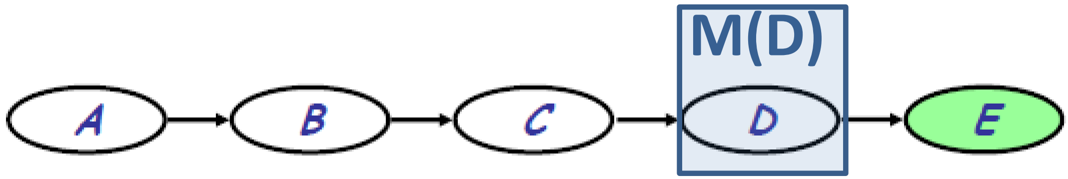
$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

F(C,D) = P(D|C)M(C)

F(C,D)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

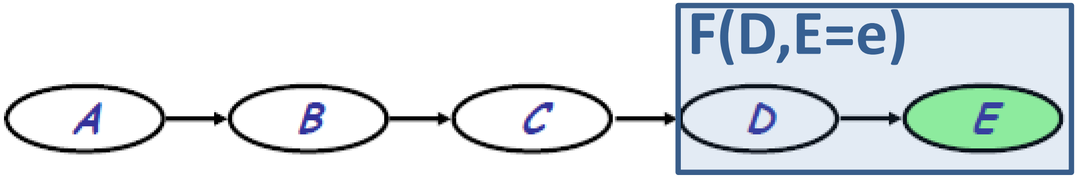
$$= \max_D P(E=e \mid D) \underline{\max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A)P(A)}$$

M(D)=max$_C$ F(C,D)

D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

F(C,D)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

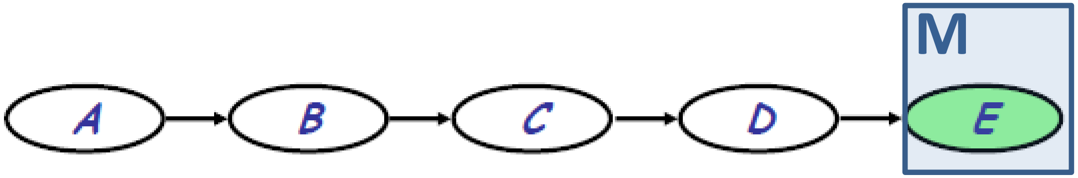
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A) P(A)$$

F(D)=P(E=e|D)M(D)

P(E=e|D)	d1	d2	d3
e	…	…	…

D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A) P(A)$$

M= $\max_D$ F(D)

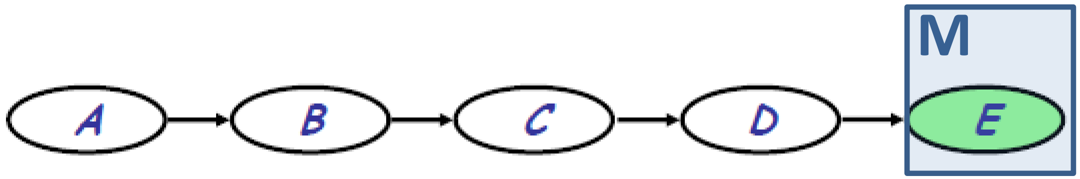
D*	M
d2	…

F(D,E=e)	d1	d2	d3
e	…	…	…

What we get ? ➔ M = $\max_{ABCD}$ P(A,B,C,D,E=e)

What we want ? ➔ (A*,B*,C*,D*) = argmax_{ABCD} P(A,B,C,D,E=e)

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_D \max_C \max_B \max_A P(E=e \mid D) P(D \mid C) P(C \mid B) P(B \mid A) P(A)$$

$$= \max_D P(E=e \mid D) \max_C P(D \mid C) \max_B P(C \mid B) \max_A P(B \mid A) P(A)$$

What we want ? ➔ (A*,B*,C*,D*) = argmax$_{ABCD}$ P(A,B,C,D,E=e)

(a1 , b1, c2, d2)

D*	M
d2	...

D	C*(D)	M(D)
d1	c1	...
d2	c2	...
d3	c3	...

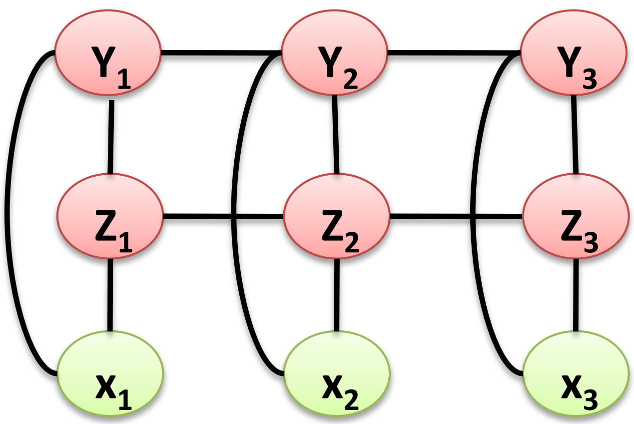
C	B*(C)	M(C)
c1	b3	...
c2	b1	...
c3	b2	...

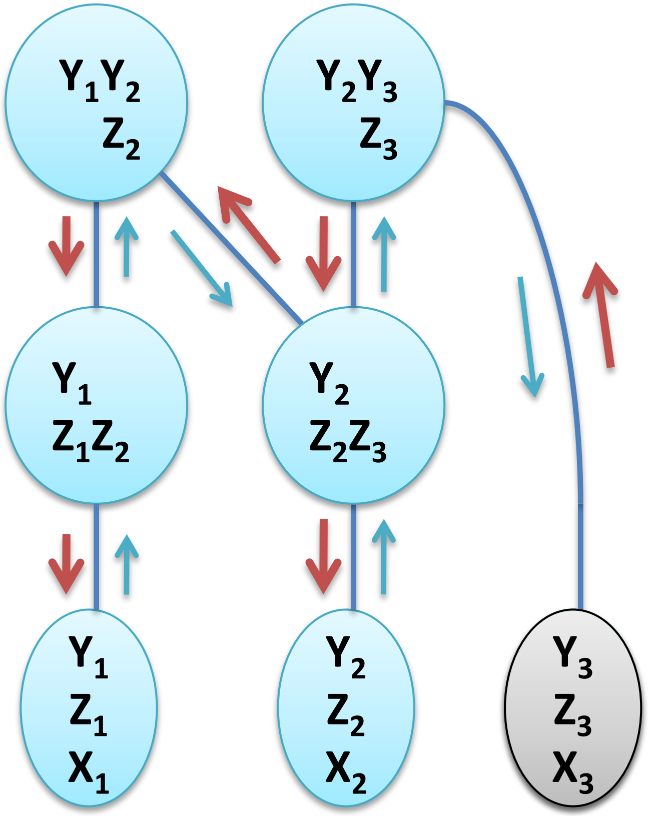
B	A*(B)	M(B)
b1	a1	...
b2	a3	...
b3	a2	...

Most Probable Assignment on general Graph

It's straight forward to generalize algorithm above to case of **general graph** with similarity of "∑" and "**max**".

(The difference is there must be a "traceback" procedure to find the "argmax" after we get "max".)





Summary

- To solve inference problems like "likelihood of X", "P(Z|X)", "Most Probable Assignment", we can use Variable Elimination (e.g. Sum-Product) algorithm
- In case of tree-structured factor graph, we just run "2 passes" of VE from leaves to a root & the reverse.
- In case of general-structured graph, we must find a "good" elimination order inducing smallest "maximum clique", which is often done with greedy method.
- When we know which variables will be given in advance, we can derive much easier model M' from original M with evidence, which is more tractable in Inference & Learning.

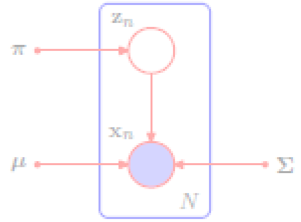
Deterministic (Variational) Approximate Inference

Reference:

Bayesian Reasoning and Machine Learning Ch. 28 (David Barber)
Probabilistic Graphical Model Ch. 11 (Koller & Friedman)
Pattern Recognition & Machine Learning Ch. 10. (Bishop)

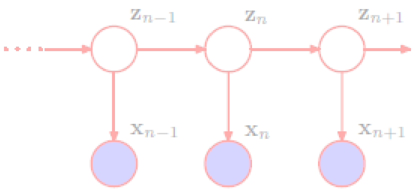
In terms of difficulty,
there are 3 types of inference problem.

- Inference which is easily solved with Bayes rule.



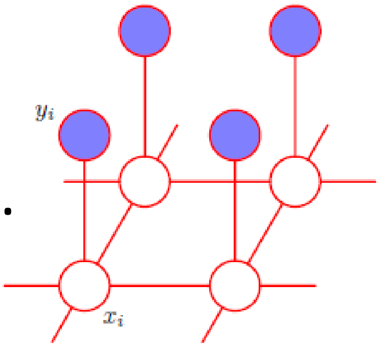
- Inference which is tractable using some dynamic programming technique.

  (e.g. Variable Elimination or J-tree algorithm)



Today's focus

- Inference which is proved intractable
  & should be solved using some Approximate Method.
  (e.g. Approximation with Optimization or Sampling technique.)

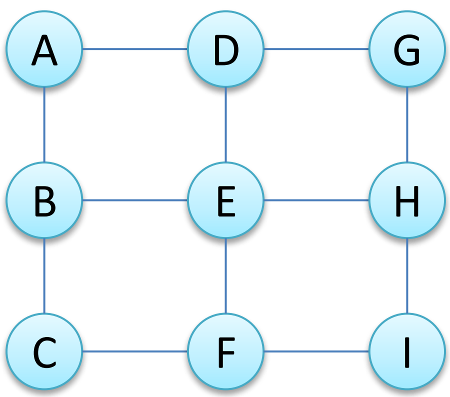


Agenda

- Principle of Variational Approximation
- Global Approximation
  (Mean Field Approximation)
- Message Approximation
  (Expectation Propagation)

Intractable Inference

Example: A N*N Grid MRF (N=3)



What we can solve: $\tilde{P}(X)$: *unnormalized distribution*

$$P(A,\ldots,I) = \frac{1}{Z}\tilde{P}(A,\ldots,I), \qquad \tilde{P}(A,\ldots,I) = \prod_{(X_1,X_2)} \phi(X_1, X_2)$$

(tractable)

What we cannot solve:

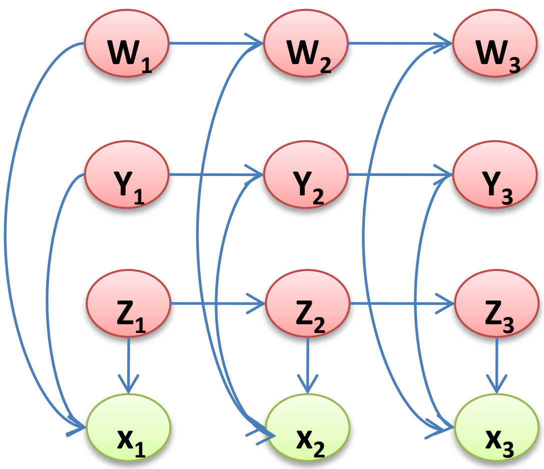
$$Z = \sum_{(A,\ldots,I)} \tilde{P}(A,\ldots,I)$$

$$\tilde{P}(A) = \sum_{(B,\ldots,I)} \tilde{P}(A,\ldots,I)$$

(intractable, as N increases)

Intractable Inference

Example:
N layers Factorial HMM



What we can solve:

$$P(W,Y,Z \mid X=x) = \frac{1}{Z(X=x)} P(W,Y,Z,X=x),$$

$$P(W,Y,Z,X=x) = \underline{P(W)*P(Y)*P(Z)*P(X=x|W,Y,Z)}$$

(easy)

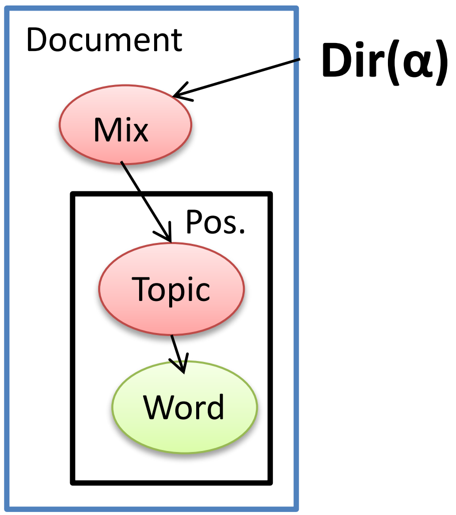
What we cannot solve: $\tilde{P}(X)$: *unnormalized distribution*

$$P(X=x) = \sum_{W,Y,Z} P(W,Y,Z,X=x)$$ **(hard)**

$$\tilde{P}(Z_1 = z \mid X=x) = \sum_{(W,Y,Z_2 \ldots Z_3)} P(W,Y,Z_1 \ldots Z_3, X=x)$$

Intractable Inference

Example:
Latent Topic Model



Some intractability comes not from "Structure", but from passing message between **different type of distribution**.

$$Let\ \ Mix = \theta = (\theta_1,\ldots,\theta_K),\ \ K = number\ of\ \ topics$$

$$P(Topic_1 = Z_1 \mid Mix = \theta) = \ \theta_{Z_1} \qquad M_{Topic1\to Mix}(\theta)\ =\ \sum_{Z_1=1}^{K} \theta_{Z_1} P(w_1 \mid Z_1)$$

$$P(Topic_2 = Z_2 \mid Mix = \theta) = \ \theta_{Z_2} \qquad M_{Topic2\to Mix}(\theta)\ =\ \sum_{Z_2=1}^{K} \theta_{Z_2} P(w_2 \mid Z_2)$$

No compact representation for message:

$$\tilde{P}(Mix = \theta \mid w) = P(Mix = \theta) * M_{Topic_1\to Mix}(\theta) * M_{Topic_2\to Mix}(\theta)$$

$$= \left(\frac{1}{const}\prod_{k=1}^{K}\theta_k^{\alpha-1}\right)\left(\sum_{Z_1}\theta_{Z_1}P(w_1 \mid Z_1)\right)\left(\sum_{Z_2}\theta_{Z_2}P(w_2 \mid Z_2)\right)$$

Summation is intractable. (Exponential to #variables)

$$\int_{\theta}\tilde{P}(Mix \mid w)\,d\theta = \frac{1}{const}\ \sum_{Z_1}\sum_{Z_2}\int_{\theta}\ \left(\prod_{k=1}^{K}\theta_k^{\alpha-1+1[k=Z_1]+1[k=Z_2]}\right)P(w_1 \mid Z_1)P(w_2 \mid Z_2)\ \ d\theta$$

Principle of Variational Approximation

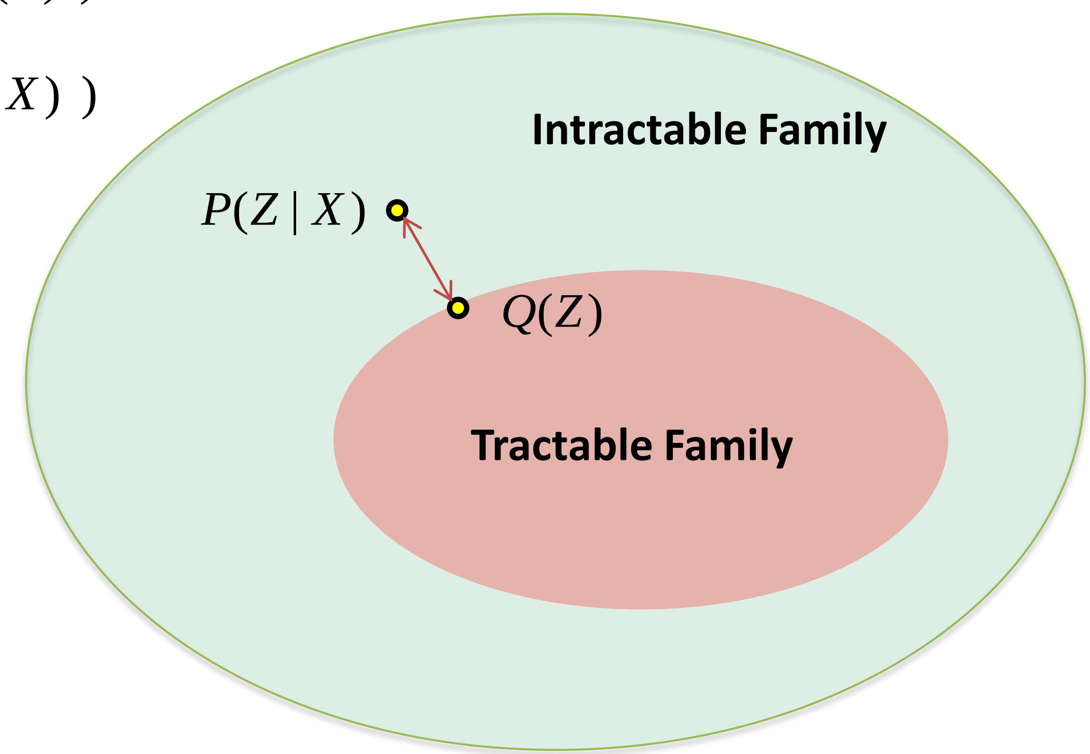
Let **X**: observation, **Z**: hidden variables.

Finds an approximate distribution **Q(Z)** from a "Tractable Family" that most similar to the target distribution **P(Z|X)** measured by some distance like KL divergence.

$$Q^*(Z) = \arg\min_{Q(Z)} \ \mathrm{KL}(\ P(Z \mid X) \,\|\, Q(Z) \ )$$

$$Q^*(Z) = \arg\min_{Q(Z)} \ \mathrm{KL}(\ Q(Z) \,\|\, P(Z \mid X) \ )$$

How can we optimize Q(Z) without computing P(Z|X) ?



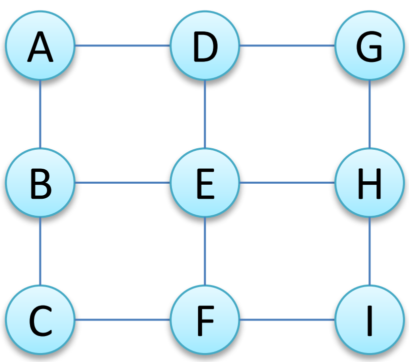
Agenda

- Principle of Variational Approximation
- Global Approximation
  (Mean Field Approximation)
- Message Approximation
  (Expectation Propagation)

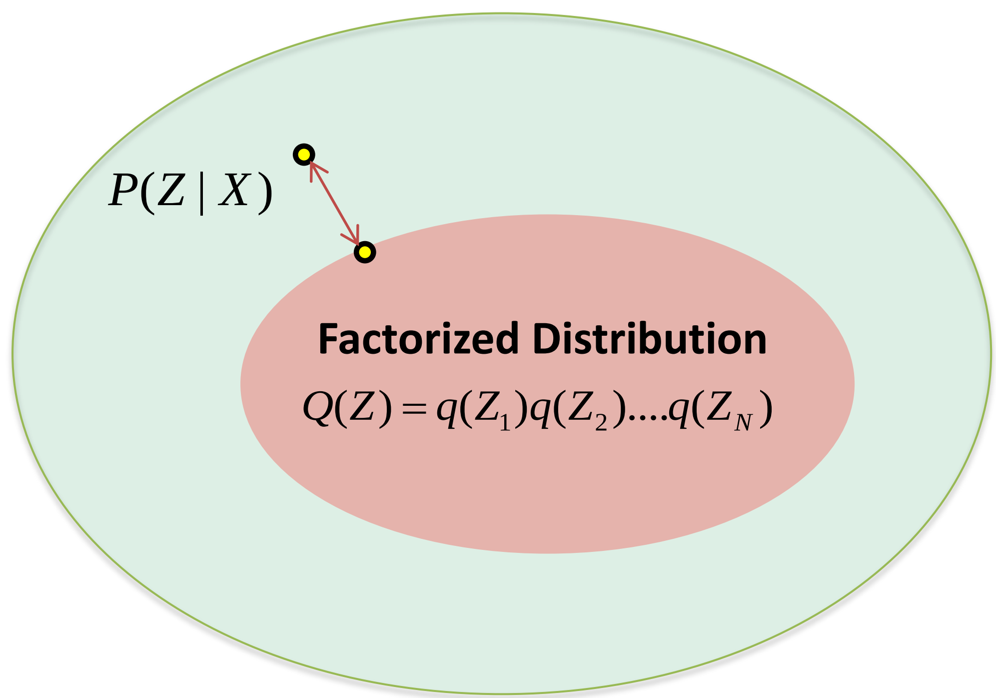
Global Approximation

One of the most popular tractable family is **Factorized Distribution**, which assumes the target (posterior) distribution **P(Z|X)** can be factorized into **$q(Z_1)*q(Z_2)...*q(Z_N)$**, that is, variables are independent to each other.

Example:



$$P(A,\ldots,I) = \frac{1}{Z}\tilde{P}(A,\ldots,I)$$
$$\approx q(A)q(B)\ldots\ldots q(I)$$



How to Find Q*(Z) ?

$$Q^*(Z) = \underset{Q(Z)\in TractableFamily}{\arg\min} \; \mathrm{KL}(\; Q(Z) \,\|\, P(Z\,|\,X)\;)$$

$$\mathrm{KL}(\; Q(Z) \,\|\, P(Z\,|\,X)\;) = E_{Q(Z)}[\log \frac{1}{P(Z\,|\,X)} - \log \frac{1}{Q(Z)}]$$

$$= E_{Q(Z)}[\log Q(Z) - \log P(Z\,|\,X)]$$

$$= E_{Q(Z)}[\log Q(Z) - \log P(Z, X) + \log P(X)]$$

$$= \underbrace{E_{Q(Z)}[\log Q(Z)] - E_{Q(Z)}[\log P(Z, X)]}_{\text{(Tractable if Q(Z) is tractable)}} + \underbrace{\log P(X)}_{\text{(intractable.... but Independent of Q(Z))}}$$

(Tractable if Q(Z) is tractable)

(intractable....
but Independent of Q(Z))

The resulting problem is equivalent to:

$$Q^*(Z) = \underset{Q(Z)\in TractableFamily}{\arg\max} \; \mathrm{E}_{\mathrm{Q(Z)}}[\log \frac{\mathrm{P(Z,X)}}{\mathrm{Q(Z)}}]$$

Find **Q(Z)** that put "similar weight" to **P(Z,X)** on which **Z=z** to happen.

How to Find $Q^*(Z)$?

$$Q^*(Z) = \arg\max_{Q(Z)=q(Z_1)q(Z_2)\dots q(Z_N)} \mathrm{E}_{Q(Z)}[\log \frac{\mathrm{P(Z,X)}}{\mathrm{Q(Z)}}]$$

Find **Q(Z)** that put "similar weight" to **P(Z,X)** on which **Z=z** to happen.

We maximize w.r.t. one $q(Z_n)$, while fixing all the other.

$$\max_{q(Z_1)} \mathrm{E}_{Q(Z)}[\log \mathrm{P(Z,X)}] - \mathrm{E}_{Q(Z)}[\log \mathrm{Q(Z)}]$$

$$= \mathrm{E}_{q(Z_1)}\left[\mathrm{E}_{q(Z_2)\dots q(Z_N)}[\log \mathrm{P(Z,X)}] \right] - \mathrm{E}_{q(Z_1)}[\log q(Z_1)] - \sum_{k\neq 1} \mathrm{E}_{q(Z_k)}[\log q(Z_k)]$$

Expectation over other variables denote as $\log \hat{P}(Z_1, X) + const.$

Independent to $q(Z_1)$

$$= \mathrm{E}_{q(Z_1)}\left[\log \frac{\hat{\mathrm{P}}(Z_1, X)}{q(Z_1)} \right] + \mathrm{const.}$$

$$-KL(\, q(Z_1) \,\|\, \hat{P}(Z_1, X)\,)$$

⇨

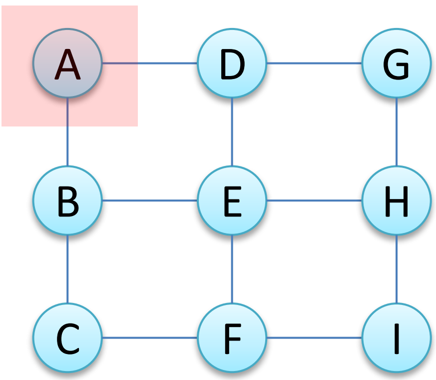
$$q^*(Z_1) = \hat{P}(Z_1, X) \;\Rightarrow\; \log q^*(Z_1) = \log \hat{P}(Z_1, X)$$

$$\log q^*(Z_1) = E_{q(Z_2)\dots q(Z_N)}[\log P(Z,X)] + const.$$

$$= \sum_{\substack{f \in factors \\ related\ to\ Z_1}} E[\ \log f(Z_{k_1} \dots Z_{k_m})\ ] + const.$$

How to Find Q*(Z) ?

Example:



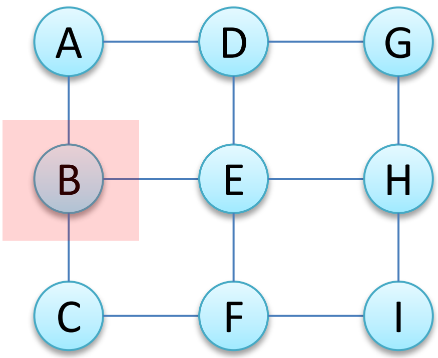
$$P(A,\ldots,I) = \frac{1}{Z}\tilde{P}(A,\ldots,I)$$
$$\approx q(A)q(B)\ldots\ldots q(I)$$

Given other **q(B)…q(I)** fixed, maximize w.r.t. **q(A)**:

$$\log \tilde{q}^*(A) = E_{q(B)\ldots q(I)}[\log \tilde{P}(A,\ldots,I)]$$
$$= \mathrm{E}_{q(B)}[\,\log\phi(A,B)\,] + \mathrm{E}_{q(D)}[\,\log\phi(A,D)\,] + \text{const.}$$

How to Find $Q^*(Z)$?

Example:



Given other **q(B)…q(I)** fixed, maximize w.r.t. **q(A)**:

$$\log \tilde{q}^*(A) = E_{q(B)\ldots q(I)}[\log \tilde{P}(A,\ldots,I)]$$
$$= \mathrm{E}_{q(B)}[\log \phi(A,B)] + \mathrm{E}_{q(D)}[\log \phi(A,D)]$$

$$\log \tilde{q}^*(B) = E_{q(A)q(C)\ldots q(I)}[\log \tilde{P}(A,\ldots,I)]$$
$$= \mathrm{E}_{q(A)}[\log \phi(A,B)] + \mathrm{E}_{q(C)}[\log \phi(B,C)] + \mathrm{E}_{q(E)}[\log \phi(B,E)] + \text{const.}$$

$$P(A,\ldots,I) = \frac{1}{Z}\tilde{P}(A,\ldots,I)$$
$$\approx q(A)q(B)\ldots\ldots q(I)$$

Iterate over all variables ….. until convergence !!

Guarantee convergence to stationary point of $\max_{Q(Z)} \mathrm{E}_{Q(Z)}[\log \frac{P(Z,X)}{Q(Z)}]$ **(Why?)**

(Every update **strictly increase** objective function, since **KL(q || p)=0 only if $q(z_k)=p(z_k)$**.
Since the maximum is bounded, we are guaranteed to convergence.)

Agenda

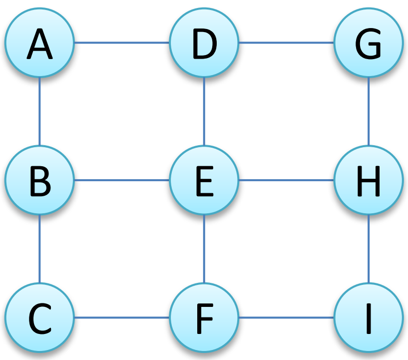
- Principle of Variational Approximation

- Global Approximation
  (Mean Field Approximation)

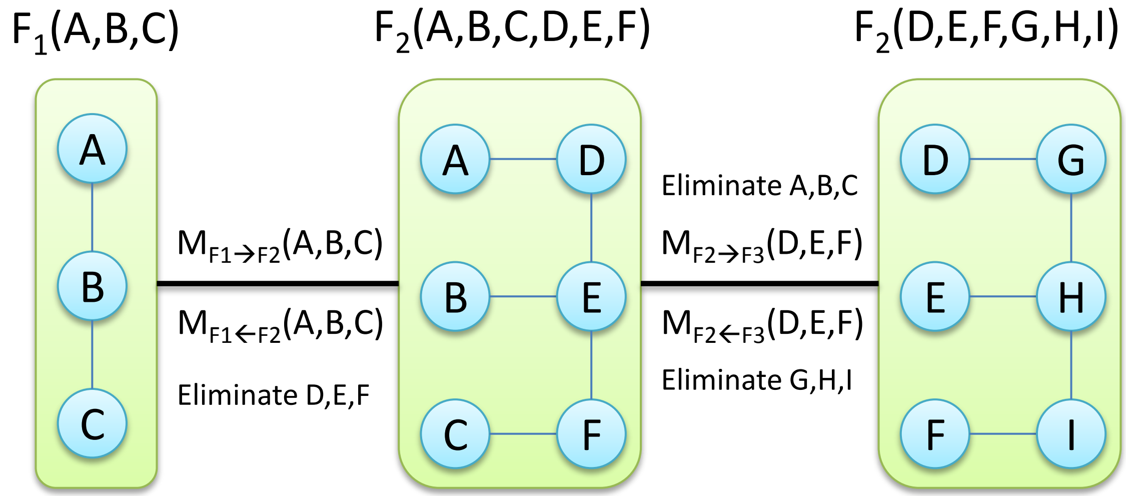
- Message Approximation
  (Expectation Propagation)

Message Approximation

Example: A N*N Grid MRF (N=3)



Variable Elimination ➔ Clique Tree



The Elimination:

$$M_{F_2 \to F_3}(D,E,F) = \sum_{A,B,C} M_{F_1 \to F_2}(A,B,C) F(A,B,C,D,E,F)$$

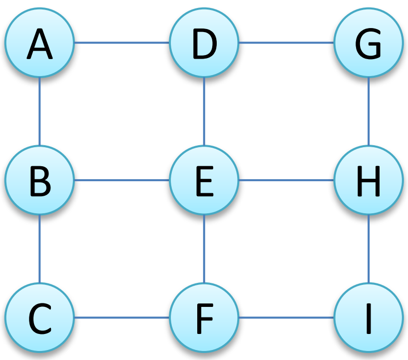
is intractable. (exponential in N)

However, can we approximate the message $\mathbf{M_{Fi \to Fj}(\ldots)}$ to make the elimination tractable ?

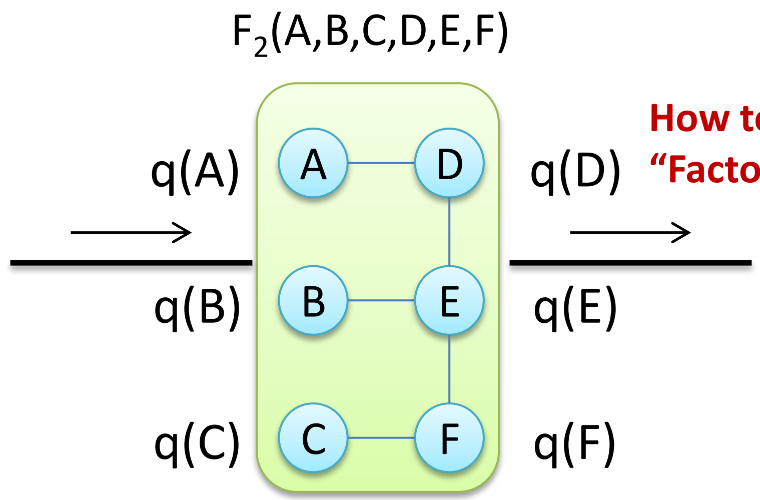
➔ **Assume it is factorized !!**

Message Approximation

Example: A N*N Grid MRF (N=3)



Variable Elimination ➔ Clique Tree



How to obtain the next "Factorized Message" ?

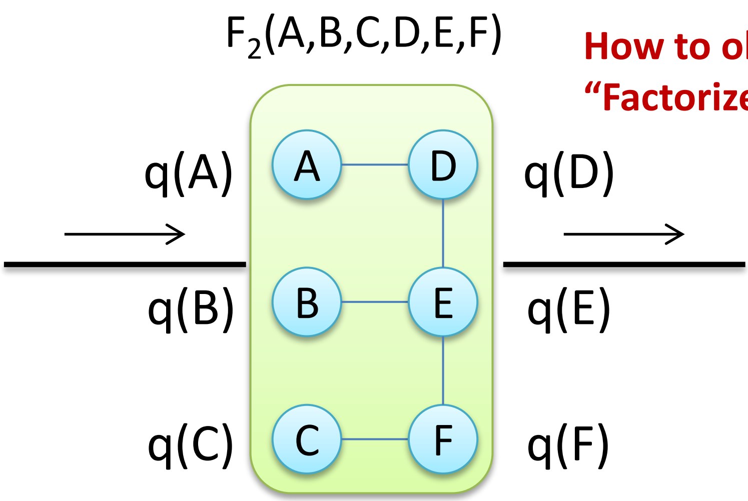
Approximate the message by a factorized distribution:

$$M_{F_1 \to F_2}(A,B,C) = q(A)q(B)q(C)$$

A, B, C not entangled !! $q(A)q(B)q(C)F_2(A,B,C,D,E,F)$ forms a tree.

➔ We can compute marginal by sum-product algorithm !!

How to obtain a Factorized Message ?



How to obtain the next "Factorized Message" ?

$$\min_{q(D)q(E)q(F)} \mathrm{KL}(\ q(D)q(E)q(F) \,\|\, M_{F_2 \to F_3}(\mathrm{D,E,F})\ )\ \textbf{?}$$

➔Need iteratively solve for **q(D), q(E), q(F)**.
(Is there more efficient way ?)

$$\min_{q(D)q(E)q(F)} \mathrm{KL}(\, M_{F_2 \to F_3}(\mathrm{D,E,F}) \,\|\, q(D)q(E)q(F)\,)\ \textbf{?}$$

$$\mathrm{KL}(\, M(\mathrm{D,E,F}) \,\|\, q(D)q(E)q(F)\,) = E_{M(\mathrm{D,E,F})}(\log \frac{M(D,E,F)}{q(D)q(E)q(F)})$$

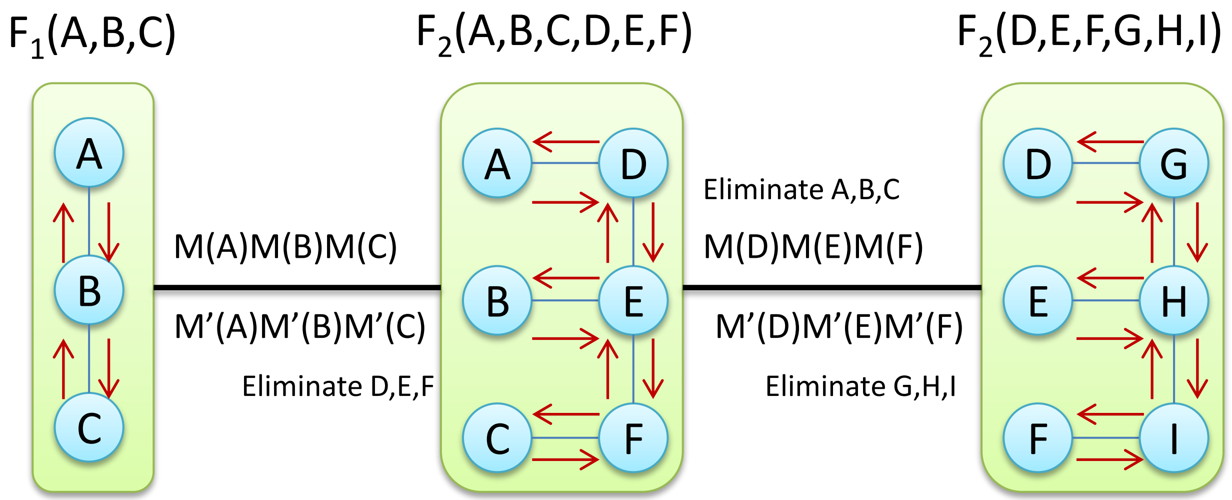
$$= E_{M(\mathrm{D,E,F})}(\log \frac{M(D,E,F)}{M(D)M(E)M(F)}) + E_{M(D)}[\log \frac{M(D)}{q(D)}] + E_{M(E)}[\log \frac{M(E)}{q(E)}] + E_{M(F)}[\log \frac{M(F)}{q(F)}]$$

$$= \underline{KL(M(D,E,F) \,\|\, M(D)M(E)M(F))} + \sum_{X \in \{D,E,F\}} \underline{KL(M(X) \,\|\, q(X))}$$

const.

Set q*(X) = M(X)
(set q(D), q(E), q(F) equal to the marginal.)

2-Layers Sum-Product Algorithm with Approximate Messages



Elimination is easy since **factors in every Clique form a "Tree"**.
Computing Marginal (ex. M(D), M(E), M(F)) can be done by **inner Sum-Product Algorithm**.

Approximate Message: Expectation Propagation

Previous example is a special case of "**Expectation Propagation**". General Expectation Propagation uses distribution come from **Log-linear model** (including **Gaussian**, **Multinomial**, **Poisson**, **Dirichlet** Distribution):

$$Q_\theta(\mathrm{X}) = \frac{1}{Z(\theta)} \exp\left\{ \theta^T \mathrm{f}(\mathrm{X}) \right\} \qquad Z(\theta) = \sum_X \exp\left\{ \theta^T \mathrm{f}(\mathrm{X}) \right\} \qquad \frac{\partial}{\partial \theta} \log Z(\theta) = E_{Q_\theta(\mathrm{X})}[f(X)]$$

$$\textit{where} \;\; f(X)^T = [f_1(X), f_2(X), \ldots, f_D(X)]^T \;\; \text{are sufficient statistics} (\textit{features}) \;\; \textit{derived from } X.$$

$$\min_\theta \; \mathrm{KL}(P(\mathrm{X}) \,\|\, Q_\theta(\mathrm{X})) = \underbrace{E_{P(\mathrm{X})}[\log P(X)]}_{\text{const.}} - E_{P(\mathrm{X})}[\log Q_\theta(X)]$$

$$\max_\theta \; E_{P(\mathrm{X})}[\log Q_\theta(X)] = E_{P(\mathrm{X})}[\theta^T f(X)] - \log Z(\theta)$$

$$\frac{\partial}{\partial \theta} E_{P(\mathrm{X})}[\theta^T f(X)] - \frac{\partial}{\partial \theta} \log Z(\theta) = 0 \;\; \Rightarrow \;\; E_{P(\mathrm{X})}[f(X)] = E_{Q_\theta(\mathrm{X})}[f(X)]$$

Moment Matching!!

Match the Expectation of feature to the original message.

Approximate Message: Expectation Propagation

Previous example is a special case of "**Expectation Propagation**". A more general version uses distribution come from **Log-linear model** (including **Gaussian**, **Multinomial**, **Poisson**, **Dirichlet** Distribution):

Moment Matching: $E_{P(\mathrm{X})}[f(X)] = E_{Q_\theta(\mathrm{X})}[f(X)]$

Example:

1. Q(X) is Multi(θ):

$$f_k(X) = 1[X = k]$$

$$E_{Q_\theta(X)}\left[1[X = k]\right] = Q_\theta(X = k)$$

Moment Matching:
Set equal Marginal Probability

$$Q_\theta(X = k) = \theta_k = P(X = k)$$

As previous MRF example.

2. Q(X) is Gaussian(μ,Σ):

$$f_1(X) = X \quad f_2(X) = XX^T$$

$$E_{Q_\theta(X)}[X] = \mu$$

$$E_{Q_\theta(X)}\left[XX^T\right] = \Sigma + \mu\mu^T$$

Moment Matching:
Set equal Mean, Variance.

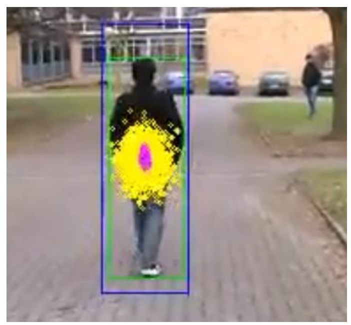
$$\mu = E_{P(X)}[X]$$

$$\Sigma = E_{P(X)}[XX^T] - \mu\mu^T$$
$$= Var_{P(X)}[X]$$

Example:
Use EP Handling Continuous / Discrete BN

When BN contains both Discrete / Continuous Variables, messages cannot have a compact representation…….

Gaussian



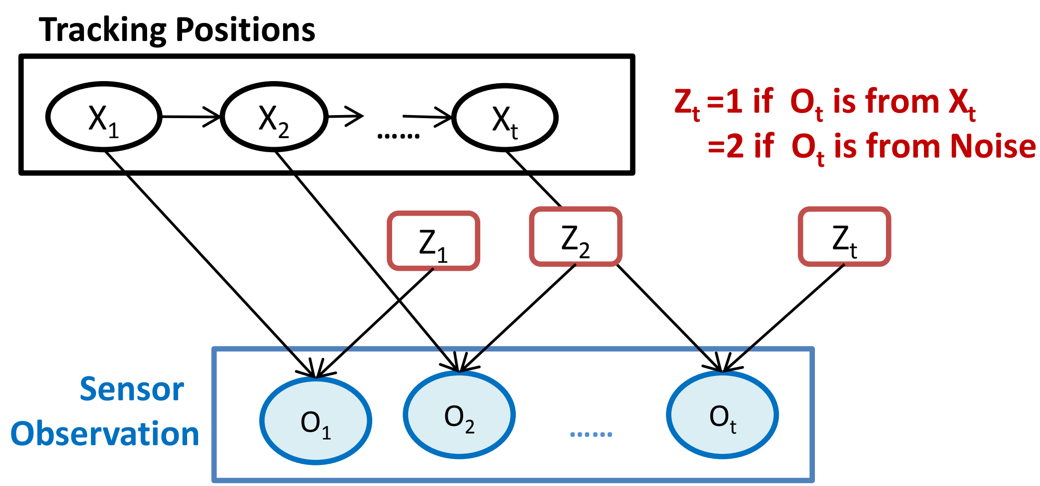
→

Mixture of Gaussian

→

Mixture of Mixture of Gaussian

(Exponentially Explode…)



Example:
Use EP Handling Continuous / Discrete BN

When BN contains both Discrete / Continuous Variables, messages cannot have a compact distribution…….

To prevent message grows to exponentially many mixtures of Gaussian….

$$M_{\rightarrow}(X_1) = \sum_{Z_1} P(Z_1)P(X_1)P(O_1 | Z_1, X_1)$$

N(X; μ_{Z1} , Σ_{Z1})

w_1*N(X; μ_1 , Σ_1) + w_2*N(X; μ_2 , Σ_2)

Approximate $M_{\rightarrow}(X_1)$ with **single Gaussian $Q(X_1)$** by "**Expectation Matching**":

$$\mu_Q = E_{M_{\rightarrow}(X_1)}[X_1] = w_1\mu_1 + w_2\mu_2$$

$$\Sigma_Q = Var_{M_{\rightarrow}(X_1)}[X_1]$$

$$= E_{P(Z_1)}[\ Var[X_1] \,|\, Z_1] + Var_{P(Z_1)}[\ E[X_1] \,|\, Z_1]$$

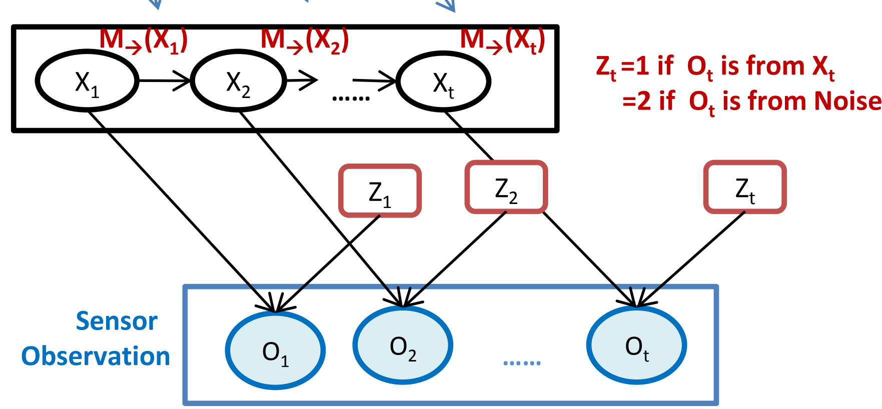
$$= E_{P(Z_1)}[\ \Sigma_{Z_1} \ | Z_1] + \ Var_{P(Z_1)}[\ \mu_{Z_1} \ | Z_1]$$

$$= w_1\Sigma_1 + w_1\Sigma_2$$

$$+ w_1(\mu_1 - \mu_Q)(\mu_1 - \mu_Q)^T$$

$$+ w_2(\mu_2 - \mu_Q)(\mu_2 - \mu_Q)^T$$

Approximate Message with N(μ_{Qt} , Σ_{Qt})



Agenda

- Principle of Variational Approximation

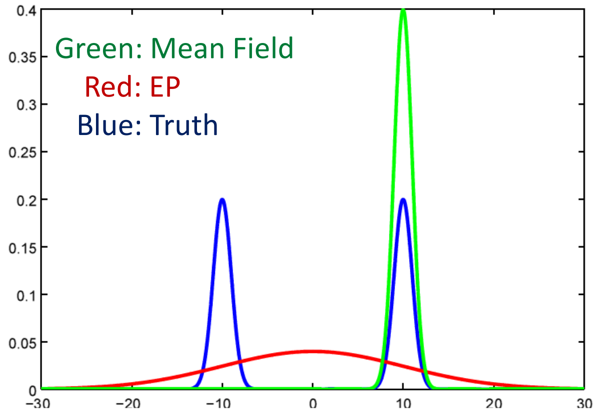
- Global Approximation
  (Mean Field Approximation)

- Message Approximation
  (Expectation Propagation)

- Comparison

Mean Field Approximation vs. Expectation Propagation

- Both of them find a tractable distribution (ex. Factorized distribution) Q(Z) to approximate the real distribution.
- Mean Field approximate joint posterior distribution P(Z|X), minimizing KL(Q||P). (Why not KL(P||Q) ? [1])
- Expectation Propagation approximate messages, minimizing KL(P||Q). (Why not KL(Q||P) ? [2])
- Expectation Propagation needs only one-pass Sum-Product, while Mean Field Approximation needs iterative maximization.
- min KL(Q||P) has more False Negative. (Why [3])
- min KL(P||Q) has more False Positive. (Why [4])



$$\frac{\partial}{\partial\theta}\log Z(\theta) = E_{Q_\theta(X)}[f(X)]$$

$$\frac{\partial}{\partial\theta}\log Z(\theta) = \frac{1}{Z(\theta)}\frac{\partial}{\partial\theta}\sum_X \exp\left\{\theta^T \mathrm{f}(X)\right\}$$

$$= \frac{1}{Z(\theta)}\sum_X \exp\left\{\theta^T \mathrm{f}(X)\right\} * \mathrm{f}(X) = \sum_X Q_\theta(X) f(X) = E_{Q_\theta(X)}[f(X)]$$

Back

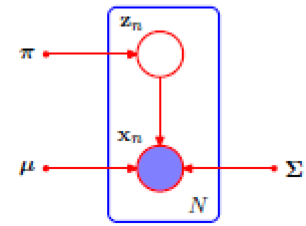
Particle-Based Approximate Inference on Graphical Model

Reference:

Probabilistic Graphical Model Ch. 12 (Koller & Friedman)
CMU, 10-708, Fall 2009 Probabilistic Graphical Models Lectures 18,19 (Eric Xing)
Pattern Recognition & Machine Learning Ch. 11. (Bishop)

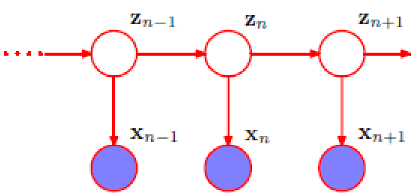
In terms of difficulty, there are 3 types of inference problem.

- Inference which is easily solved with Bayes rule.



- Inference which is tractable using some dynamic programming technique.

  (e.g. Variable Elimination or J-tree algorithm)

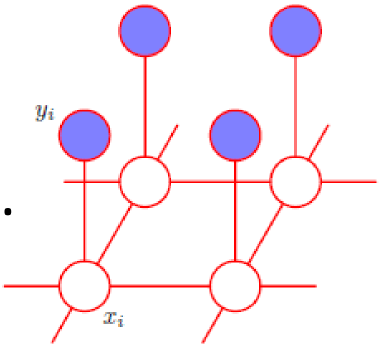


Today's focus

- Inference which is proved intractable

  & should be solved using some Approximate Method.

  (e.g. Approximation with Optimization or Sampling technique.)



Agenda

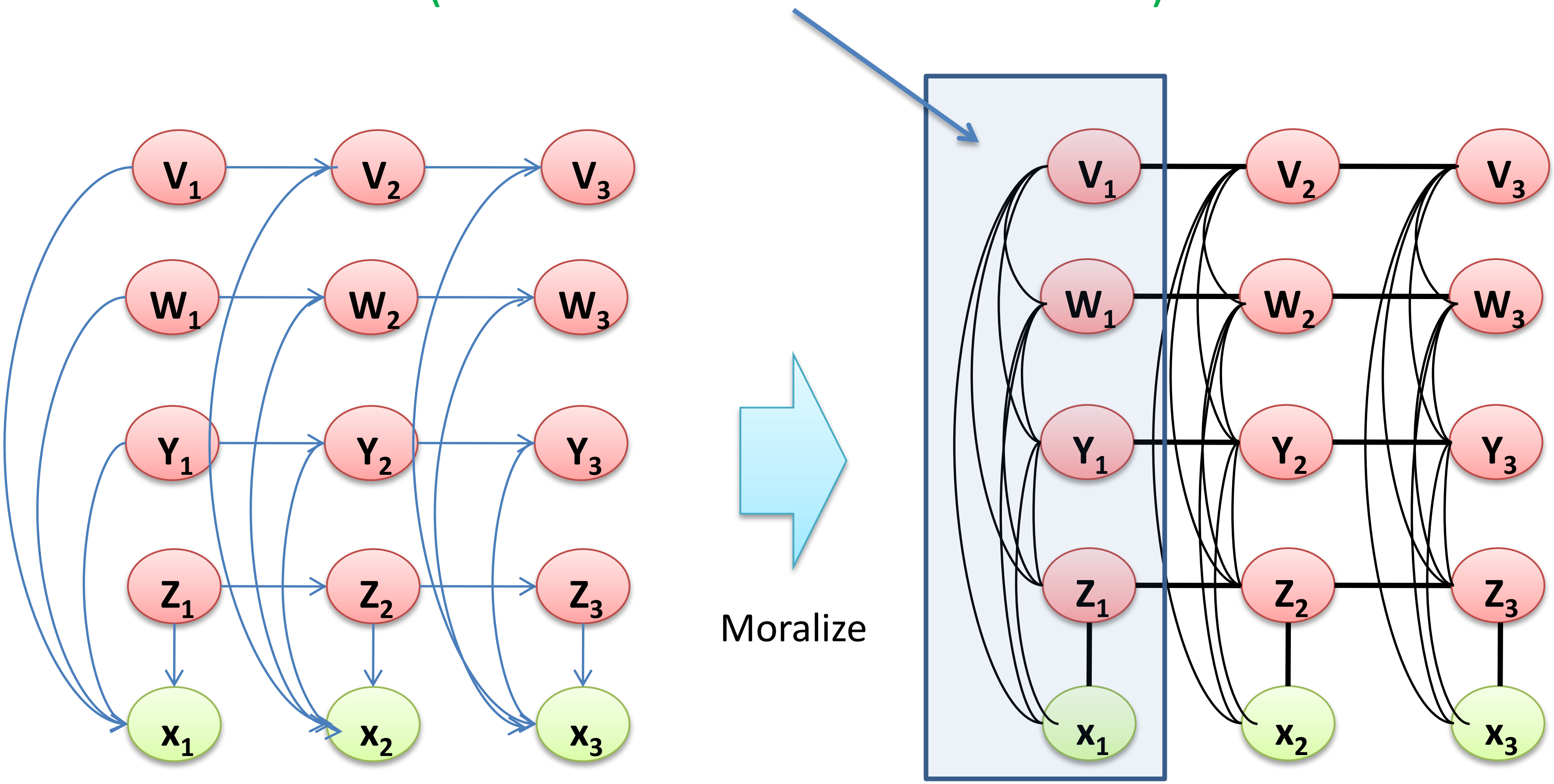
- When to use Particle-Based Approximate Inference ?
- Forward Sampling & Importance Sampling
- Markov Chain Monte Carlo (MCMC)
- Collapsed Particles

Agenda

- When to use Particle-Based Approximate Inference ?
- Forward Sampling & Importance Sampling
- Markov Chain Monte Carlo (MCMC)
- Collapsed Particles

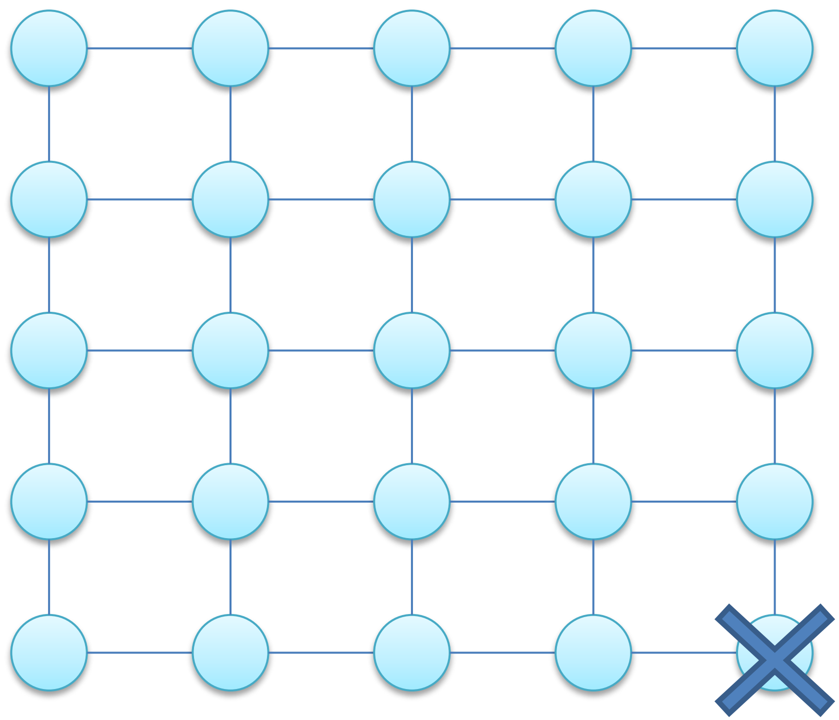
Example : General Factorial HMM

A clique size=5,
intractable most of times.
(No tractable elimination exist…)



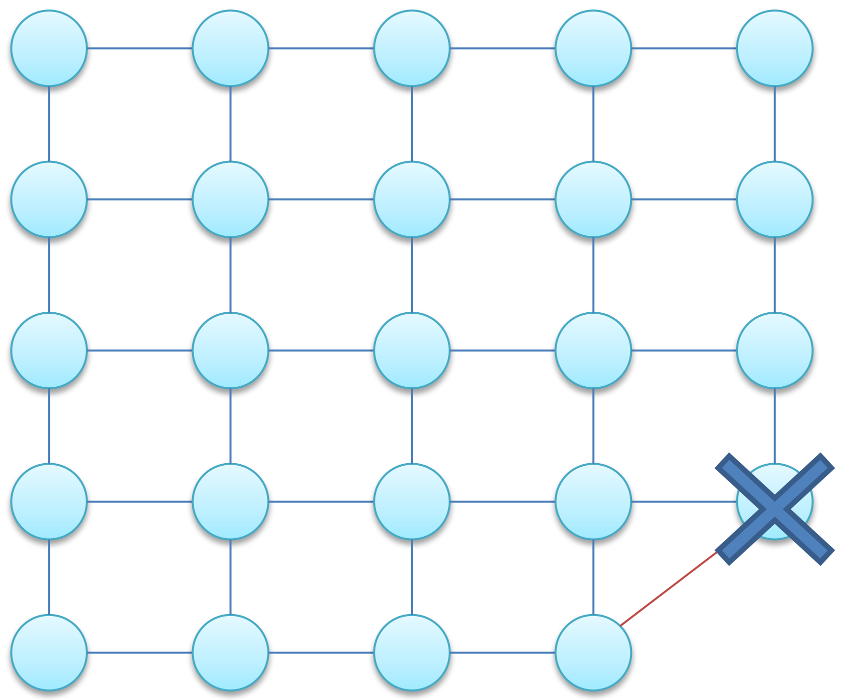
Some Model are Intractable for Exact Inference

Example: A Grid MRF



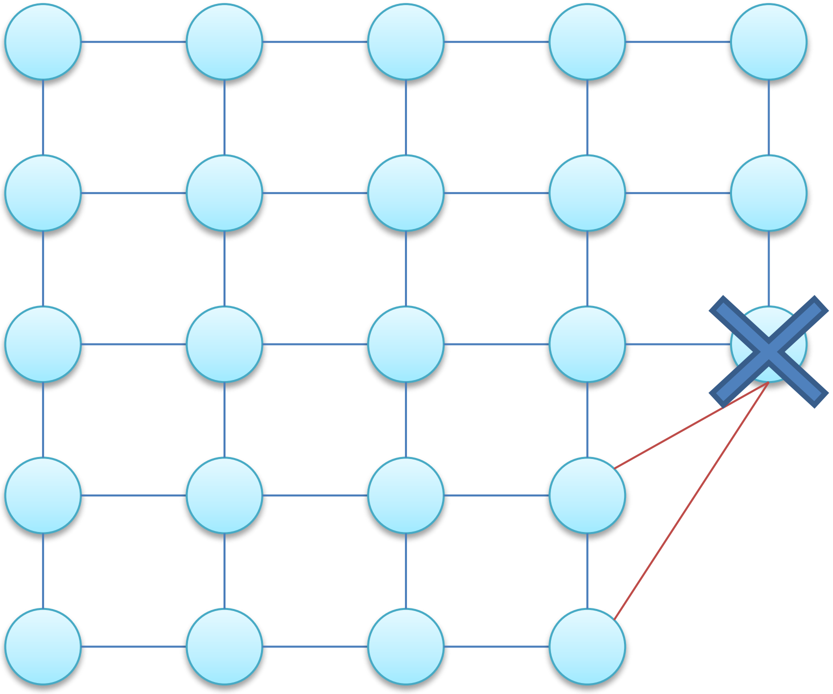
Some Model are Intractable for Exact Inference

Example: A Grid MRF



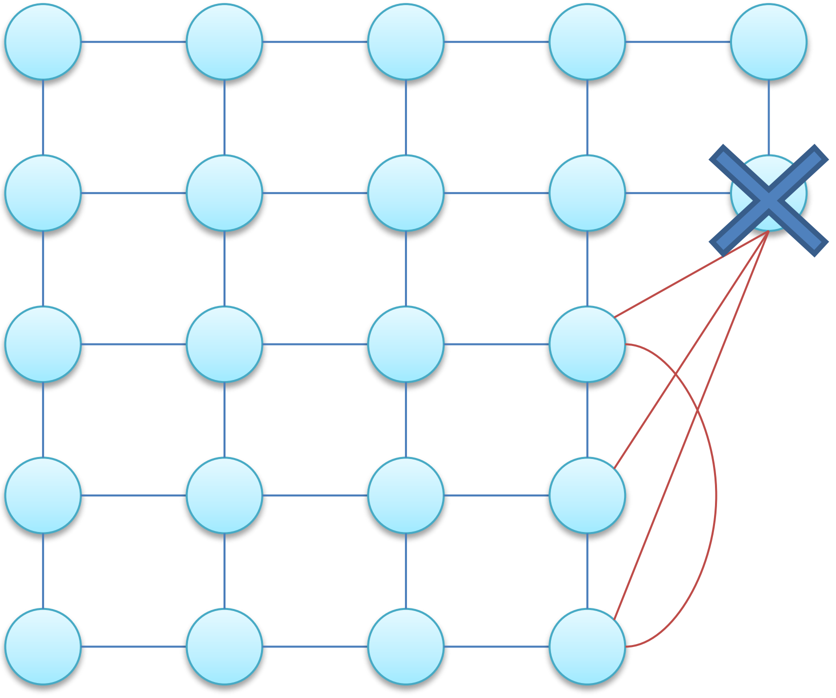
Some Model are Intractable for Exact Inference

Example: A Grid MRF



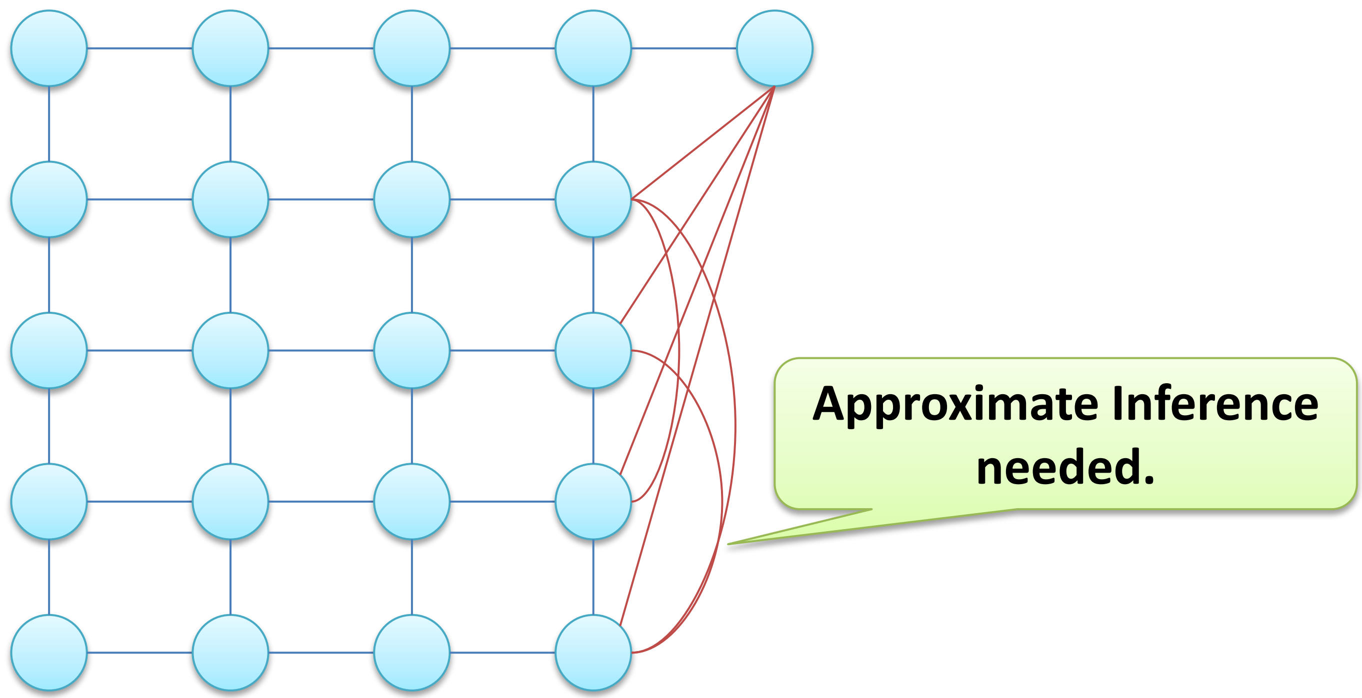
Some Model are Intractable for Exact Inference

Example: A Grid MRF



Some Model are Intractable for Exact Inference

Example: A Grid MRF



Generally, we will have clique of “size N” for a N*N grid, which is indeed intractable.

General idea of Particle-Based (Monte Carlo) Approximation

Most of Queries we want can be formed as:

Intractable when K→ ∞.

$$E_{P(X)}[f(X)] = \sum_{X_1} \dots \sum_{X_K} P(X_1 \dots X_K) * f(X_1 \dots X_K)$$

which is intractable most of time. Assume we can generate i.i.d. samples $X^{(1)} \dots X^{(n)}$ from P(X), we can approximate above using:

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f(X^{(n)})$$

It's a unbiased estimator whose variance converges to 0 when N→∞.

$$E[\hat{f}] = \frac{1}{N} E[\sum_{n=1}^{N} f(X^{(n)})] = E[f(X)]$$

$$Var[\hat{f}] = \frac{1}{N^2} Var[\sum_{n=1}^{N} f(X^{(n)})] = \frac{1}{N} Var[f(X)]$$

Var. not Related to dimension of X. Var→0 as N→∞

Which Problem can use Particle-Based (Monte Carlo) Approximation ?

- Type of queries:
  - 1. Likelihood of evidence/assignments on variables
  - 2. Conditional Probability of some variables (given others).
  - ~~3. Most Probable Assignment for some variables (given others).~~

Problem which can be written as following form:

$$E_{P(X)}[f(X)] = \sum_{X_1} \dots \sum_{X_K} P(X_1 \dots X_K) * f(X_1 \dots X_K)$$

Marginal Distribution (Monte Carlo)

To Compute Marginal Distribution on X_k

$$P(X_k = x_k)$$
$$= \sum_{X_{-k}} P(X_k = x_k, X_{-k}) = \sum_{X_k} \sum_{X_{-k}} P(X_k, X_{-k}) * 1\{X_k = x_k\}$$
$$= E_{P(X)}[\,1\{X_k = x_k\}\,]$$

Particle-Based Approximation:

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} 1\{X_k^{(n)} = x_k\}$$

(Just count the proportion of samples in which X_k=x_k)

Marginal Joint Distribution (Monte Carlo)

To Compute Marginal Distribution on (X_i , X_j)

$$P(X_i = x_i, X_j = x_j)$$
$$= \sum_{X_{-ij}} P(X_i = x_i, X_j = x_j, X_{-ij}) = \sum_{X_{-ij}} \sum_{Xi} \sum_{Xj} P(X_i, X_j, X_{-k}) * 1\{X_i = x_i \;\&\; X_j = x_j\}$$
$$= E_{P(X)}[\,1\{X_i = x_i \;\&\; X_j = x_j\}\,]$$

Particle-Based Approximation:

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} 1\{X_i^{(n)} = x_i, X_j^{(n)} = x_j\}$$

(Just count the proportion of samples in which X_i=x_i & X_j=x_j)

So What's the Problem ?

Note what we **can** do is:

"**Evaluate**" the probability/likelihood $P(X_1=x_1,\ldots, X_K=x_K)$.

What we **cannot** do is:

Summation / Integration in high-dim. space: $\sum_X P(X_1,\ldots, X_K)$.

What we **want to** do (for approximation) is:

"Draw" samples from $P(X_1,\ldots, X_K)$.

How to make better use of samples ?

How to know we've sampled enough ?

How to draw Samples from P(X) ?

- Forward Sampling
  draw from ancestor to descendant in BN.

- Rejection Sampling
  create samples using Forward Sampling, and reject those inconsistent with evidence.

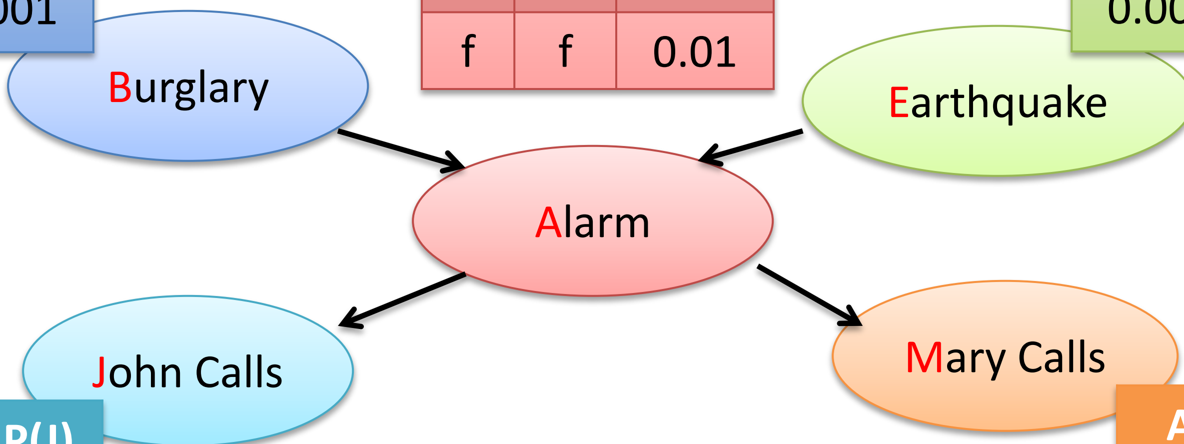
- Importance Sampling
  Sample from proposal dist. Q(X), but give large weight on sample with high likelihood in P(X).

- Markov Chain Monte Carlo
  Define a Transition Dist. $T(x \rightarrow x')$ s.t. samples can get closer and closer to P(X).

Agenda

- When to use Particle-Based Approximate Inference ?
- Forward Sampling & Importance Sampling
- Markov Chain Monte Carlo (MCMC)
- Collapsed Particles

Forward Sampling



P(B)
0.001

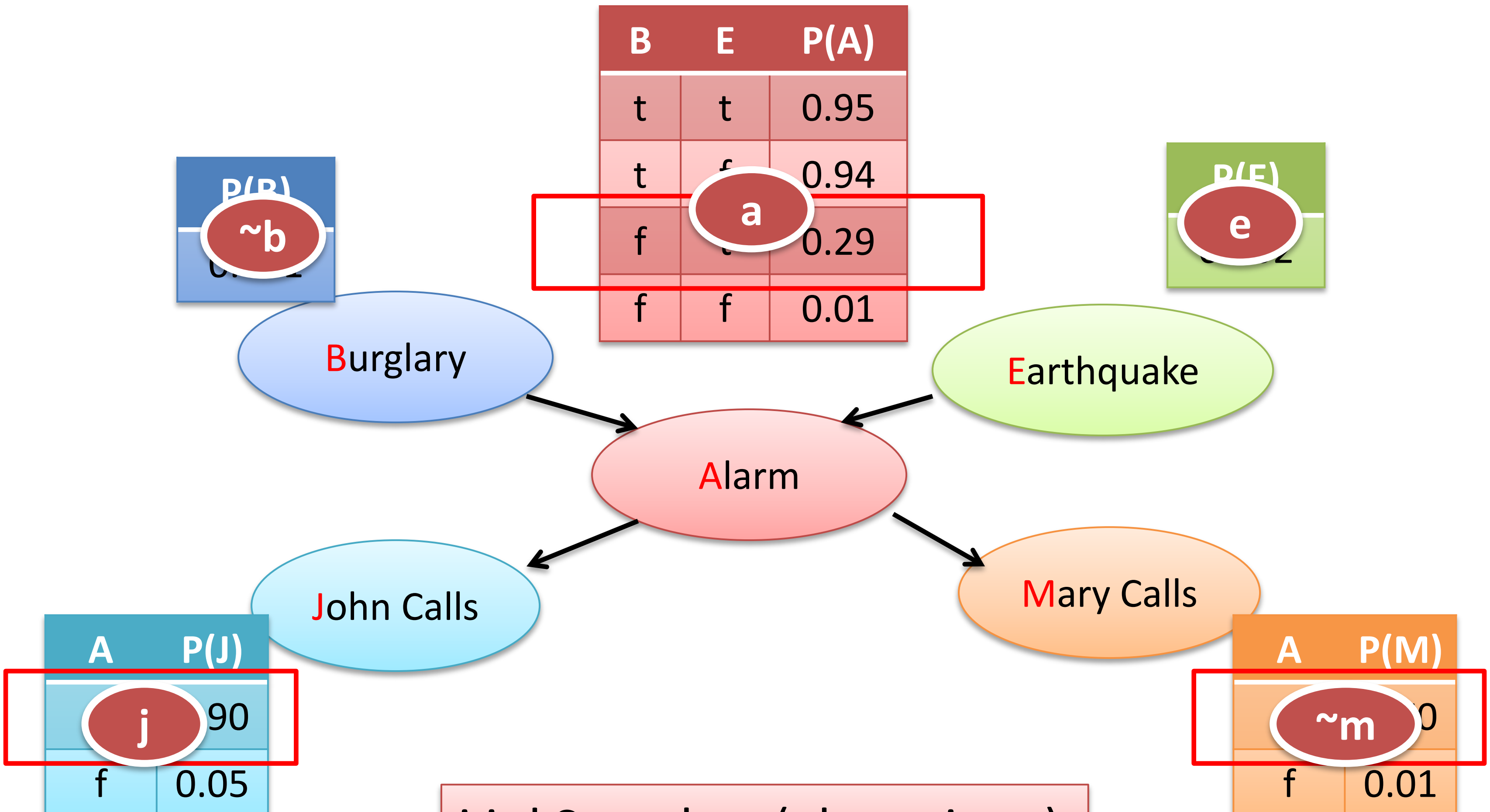
B	E	P(A)
t	t	0.95
t	f	0.94
f	t	0.29
f	f	0.01

P(E)
0.002

A	P(J)
t	0.90
f	0.05

A	P(M)
t	0.70
f	0.01

Forward Sampling



i.i.d Sample : (~b,e,a,j,~m)

Forward Sampling

Samples :
(~b,e,a,j,~m)
…
…
(~b,~e, a,~j, ~m)

← Particle-Based Represent
of the joint distribution P(B,E,A,J,M).

$$P(M=m)=\frac{1}{N}\sum_{n=1}^{N}1\{M^{(n)}=m\}$$

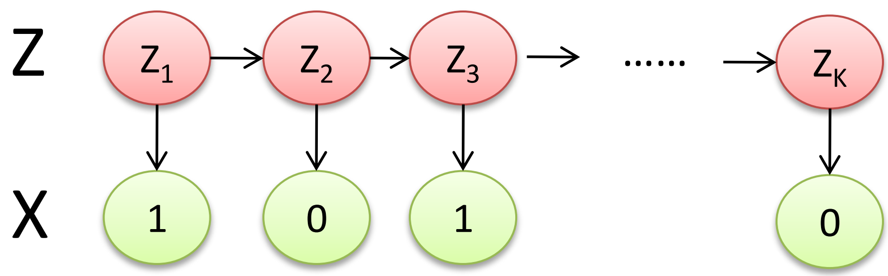
$$P(B=b,M=\sim m)=\frac{1}{N}\sum_{n=1}^{N}1\{B^{(n)}=b,M^{(n)}=\sim m\}$$

What if we want samples from P(**B, E, A** | J=j , M=~m) ?

1. Collect all samples in which J=j , M=~m .
2. Those samples form the particle-based representation of P(**B, E, A** | J=j, M=~m).

Disadvantage…….

Forward Sampling from P(Z|Data) ?



1. Forward Sampling N times.

2. Collect all samples **$(Z^{(n)},X^{(n)})$** in which **X_1=1, X_2=0, X_3=1, …… X_K=0.**

3. Those samples form the particle-based representation of **P(Z|X)**.

How many such samples can we get ??

➔ **N*P(Data) !!** (Less than 1 if N not large enough……)

Solutions…….

Importance Sampling to the Rescue

We need not draw from P(X) to compute $E_{P(X)}[f(X)]$:

$$E_{P(X)}[f(X)] = \sum_X P(X) * f(X)$$

$$= \sum_X Q(X) * (\frac{P(X)}{Q(X)} * f(X)) = E_{Q(X)}[\frac{P(X)}{Q(X)} * f(X)]$$

$$\hat{E}_{P(X)}[f(X)] = \frac{1}{N} \sum_{n=1}^{N} (\frac{P(X^{(n)})}{Q(X^{(n)})}) * f(X^{(n)})$$

That is, we can draw from an arbitrary distribution Q(X), but give larger weights on samples having higher probability under P(X).

Importance Sampling to the Rescue

Sometimes we can only evaluate an unnormalized distribution :

$$\widetilde{P}(X) \quad , where \ \frac{\widetilde{P}(X)}{Z} = P(X)$$

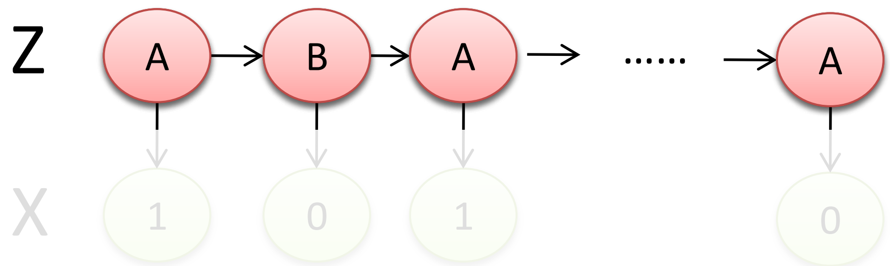
Then we can estimate Z as follows:

$$Z = \sum_X \widetilde{P}(X) = \sum_X Q(X)\frac{\widetilde{P}(X)}{Q(X)} = E_{Q(X)}[\frac{\widetilde{P}(X)}{Q(X)}] \qquad \hat{Z} = \frac{1}{N}\sum_{n=1}^{N}\frac{\widetilde{P}(X^{(n)})}{Q(X^{(n)})}$$

Note that we can compute $\hat{Z}$ only if we can evaluate a **normalized distribution** $Q(X)$, that is, we have Z_Q or $Q(X)$ is from a BN.

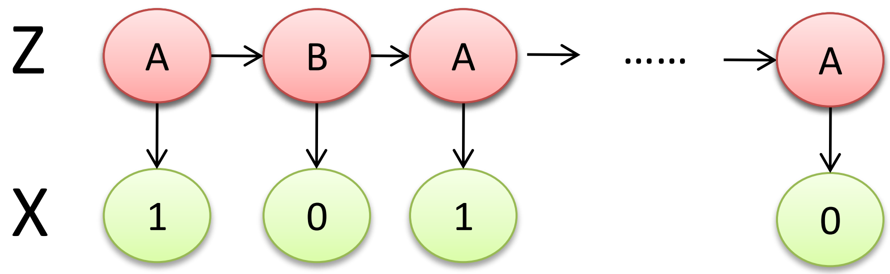
$$E_{P(X)}[f(X)] = \frac{1}{Z}E_{Q(X)}[\frac{\widetilde{P}(X)}{Q(X)} * f(X)] \qquad \hat{E}_{P(X)}[f(X)] = \frac{\hat{E}_{\widetilde{P}(X)}[f(X)]}{\hat{Z}} = \frac{\sum_{n=1}^{N}\frac{\widetilde{P}(X^{(n)})}{Q(X^{(n)})} * f(X^{(n)})}{\sum_{n=1}^{N}\frac{\widetilde{P}(X^{(n)})}{Q(X^{(n)})}}$$

Importance Sampling from P(Z|Data) ?



1. Sampling from P(Z), a normalized distribution obtained from BN truncating the part with evidence.

Importance Sampling from P(Z|Data) ?



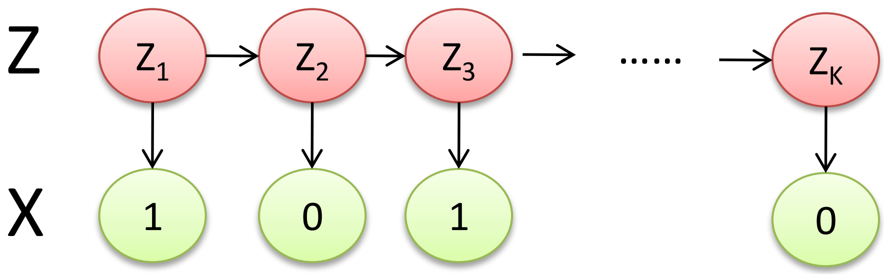
$$w^{(n)} = P(Data \mid Z)$$
$$= P(1 \mid A)P(0 \mid B)P(1 \mid A) \ldots P(0 \mid A)$$

1. Sampling from P(Z), a normalized distribution obtained from BN truncating the part with evidence.

2. Give each sample **(Z(n), X(n))** a weight:

$$w^{(n)} = \frac{\widetilde{P}(Z)}{Q(Z)} = \frac{P(Z)P(Data \mid Z)}{P(Z)} = P(Data \mid Z)$$

Importance Sampling from P(Z|Data) ?



(A,B,A,…,A)	w= 0.01
(A,B,A,…,B)	0.3
(B,B,B,…,A)	1.0

$N_{eff} = 1.31$

$P(Data)=N_{eff}/N = 1.31/3$

1. Sampling from P(Z), a normalized distribution obtained from BN truncating the part with evidence.

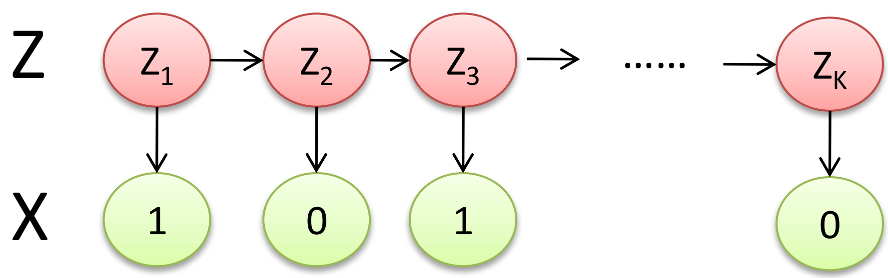
2. Give each sample **(Z(n), X(n))** a weight:

$$w^{(n)} = \frac{\tilde{P}(Z)}{Q(Z)} = \frac{P(Z)P(Data\,|\,Z)}{P(Z)} = P(Data\,|\,Z)$$

3. The effective number of samples is $N_{eff} = \sum_{n=1}^{N} w^{(n)}$

$$\left(\quad \hat{\mathrm{P}}(\mathrm{Data}) = \frac{1}{N}\sum_{n=1}^{N} w^{(n)} = \frac{1}{N}\sum_{n=1}^{N} P(Data\,|\,Z^{(n)}) \quad \right)$$

Importance Sampling from P(Z|Data) ?



(A,B,A,...,A)	w= 0.01
(A,B,A,...,B)	0.3
(B,B,B,...,A)	1.0

$N_{eff} = 1.31$

$P(Data)=N_{eff}/N = 1.31/3$

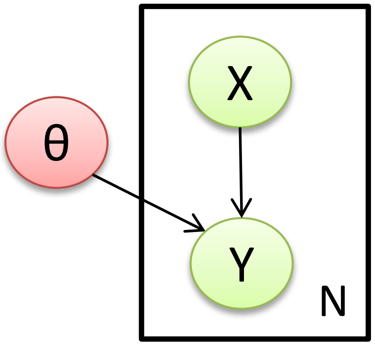
To get estimate of $P(Z_1|Data)$:

$$\hat{P}(Z_1 = B \mid Data) = \frac{0.01*0+0.3*0+1.0*1}{1.31} = 0.76$$

$$\hat{P}(Z_1 = A, Z_K = B \mid Data) = \frac{0.01*0+0.3*1+1.0*0}{1.31} = 0.23$$

Any joint dist. can be estimated. (No “out of clique” problem)

Bayesian Treatment with Importance Sampling



Ex. $P_\theta(Y=1|X) = \text{logistic}(\theta_1 * X + \theta_0)$

Often, Posterior on parameters θ :

$$P(\theta \mid Data) = \frac{P(Data \mid \theta)P(\theta)}{P(\text{Data})} = \frac{P(Data \mid \theta)P(\theta)}{\int_\theta P(Data \mid \theta)P(\theta)\ d\theta}$$

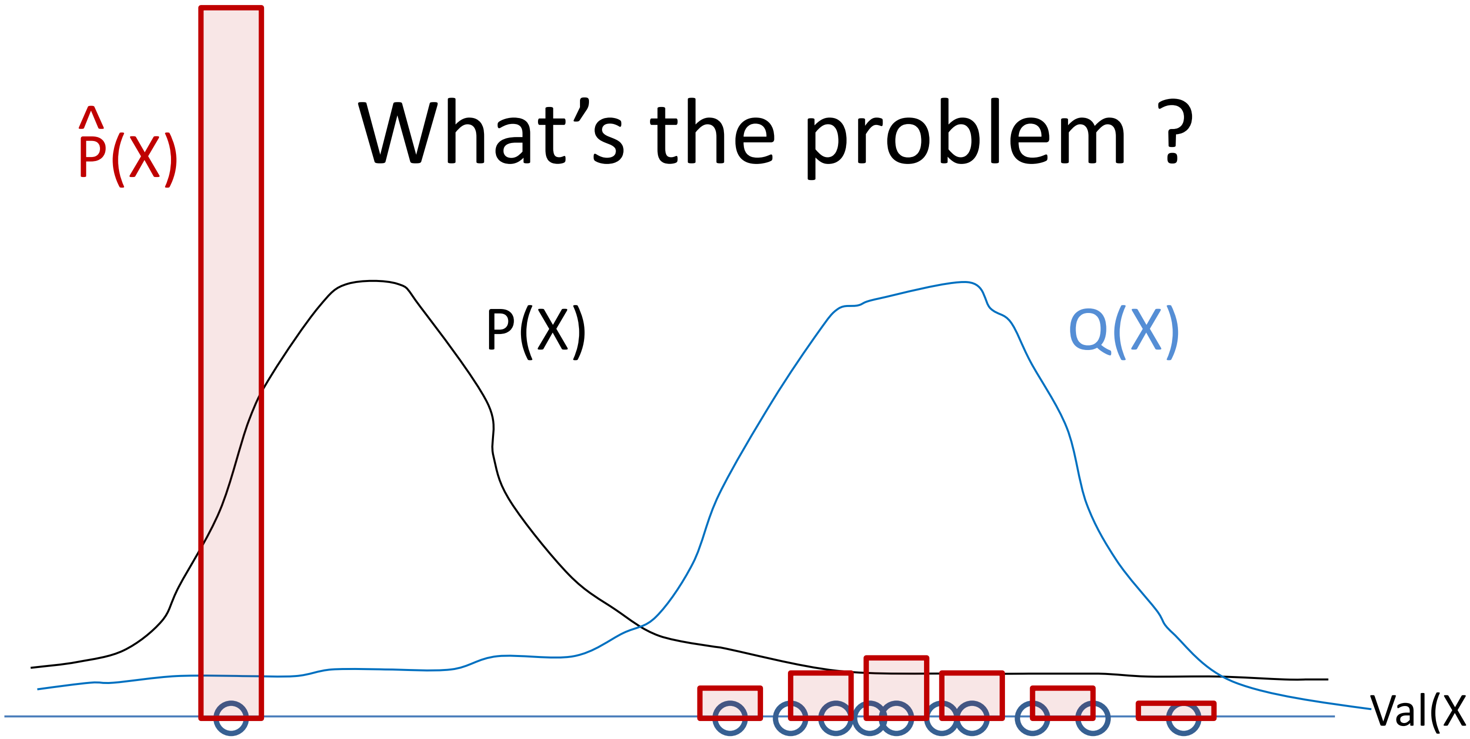
is **intractable** because many types of $P_\theta(\text{Data}|\theta)$ cannot be integrated analytically.

Approximate with:

$$\hat{\mathrm{P}}(\theta = a \mid \mathrm{Data}) = \frac{\sum_{n=1}^{N} P(Data \mid \theta^{(n)} = a)\ 1\{\theta^{(n)} = a\}}{\sum_{n=1}^{N} P(Data \mid \theta^{(n)})} = \frac{P(Data \mid \theta = a)\sum_{n=1}^{N} 1\{\theta^{(n)} = a\}}{\hat{P}(Data)}$$

We need not evaluate "the integration" to estimate P(θ|Data) using Importance Sampling.

What’s the problem ?



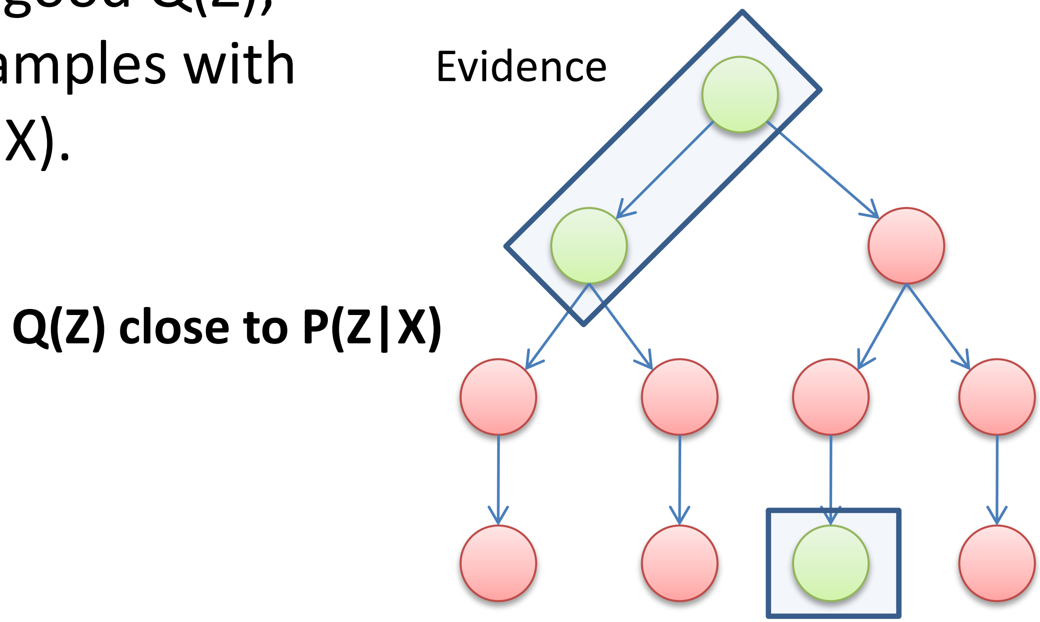
If P(X) and Q(X) not matched properly……

Only small number of samples will fall in the region with high P(X).

➔ Very large N needed to get a good picture of P(X).

How P(Z|X) and Q(Z) Match ?

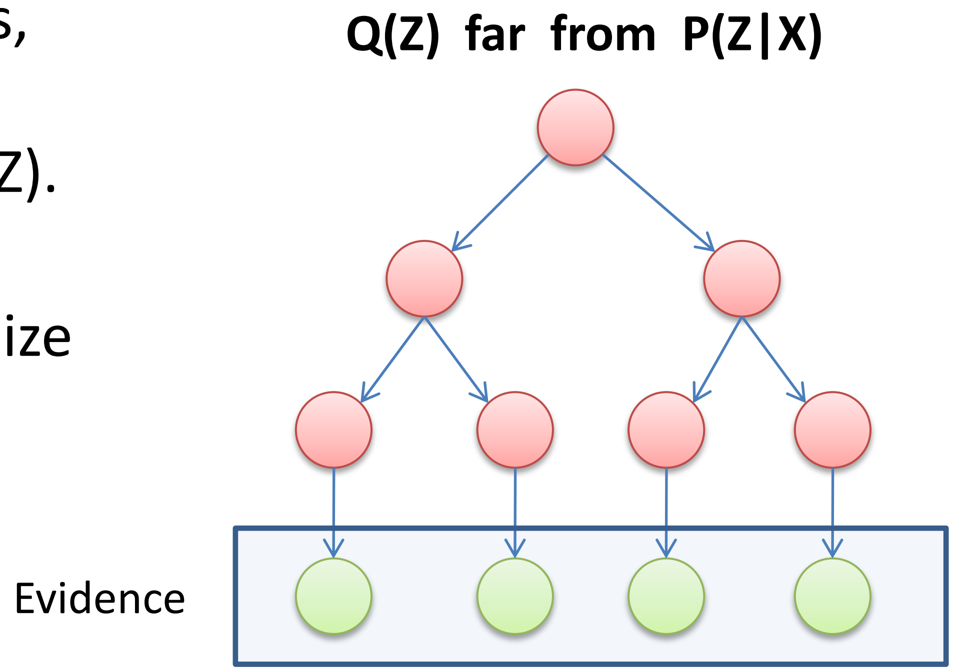
When evidence is close to root, forward sampling is a good Q(Z), which can generate samples with high likelihood in P(Z|X).



How P(Z|X) and Q(Z) Match ?

When evidence is on the leaves, forward sampling is a bad Q(Z), yields very low likelihood=P(X|Z).

So we need very large sample size to get a good picture of P(Z|X).

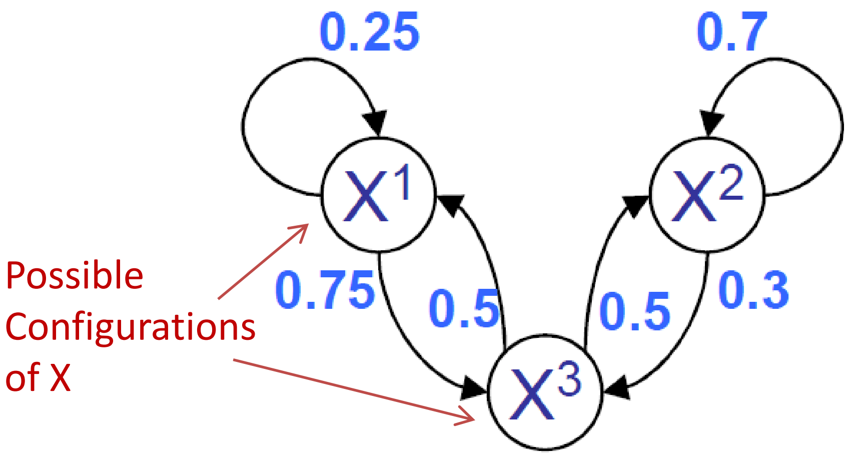


Can we **improve with time** to draw from a distribution more like the desired P(Z|X) ?
➔MCMC try to draw from a distribution closer and closer to P(Z|X).
(Apply **equally well in BN & MRF**.)

Agenda

- When to use Particle-Based Approximate Inference ?
- Forward Sampling & Importance Sampling
- Markov Chain Monte Carlo (MCMC)
- Collapsed Particles

What is Markov Chain (MC) ?



A set of Random Variables:
$\mathbf{X} = (X_1, \ldots. X_K)$

Variables change with Time:
$\mathbf{X}^{(t)} = (X_1^{(t)}, \ldots. X_K^{(t)})$

which take transition following:
$P(\mathbf{X}^{(t+1)} = x' \mid \mathbf{X}^{(t)} = x) = T(x \rightarrow x')$

There is a stationary distribution $\pi_T(X)$ for Transition T, in which:

$$\pi_T(X=x') = \sum_x \pi_T(X=x) * T(x \rightarrow x')$$

(After transition, still the same distribution over all possible configurations $X^1 \sim X^3$)

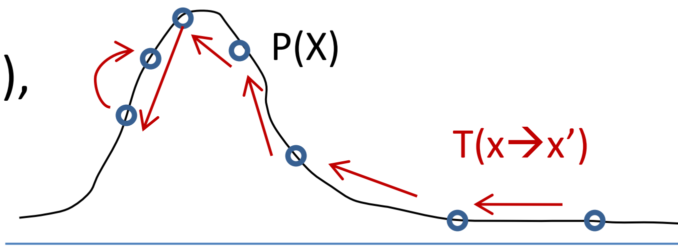
Ex. The MC (Markov Chain) above has only 1 variable X taking on values $\{x^1, x^2, x^3\}$,

There is a $\boldsymbol{\pi_T}$ s.t. $\pi_T * T = \begin{bmatrix} 0.2 & 0.5 & 0.3 \end{bmatrix} \begin{bmatrix} 0.25 & 0 & 0.75 \\ 0 & 0.7 & 0.3 \\ 0.5 & 0.5 & 0 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.5 & 0.3 \end{bmatrix} = \pi_T$

What is MCMC (Markov Chain Monte Carlo) ?

Importance Sampling is efficient only if **Q(X) matches P(X) well**.
Finding such Q(X) is difficult.

Instead, MCMC tries to find a transition dist. $T(x \rightarrow x')$,
s.t. **X tends to transit into states with high P(X)**,
and **finally follows stationary dist. $\pi_T = P(X)$.**

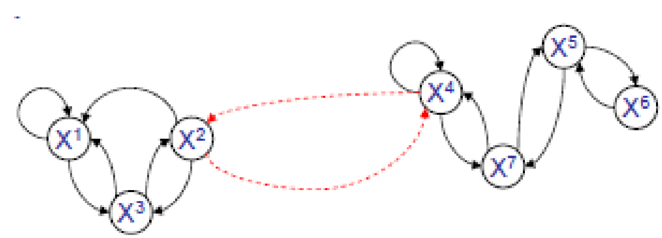


Setting $X^{(0)}$=any initial value, we samples $X^{(1)}, X^{(2)} \ldots\ldots X^{(M)}$ following $T(x \rightarrow x')$, and hope that $X^{(M)}$ follows stationary distribution **$\pi_T = P(X)$.**
If $X^{(M)}$ really does , we got a sample $X^{(M)}$ from P(X).

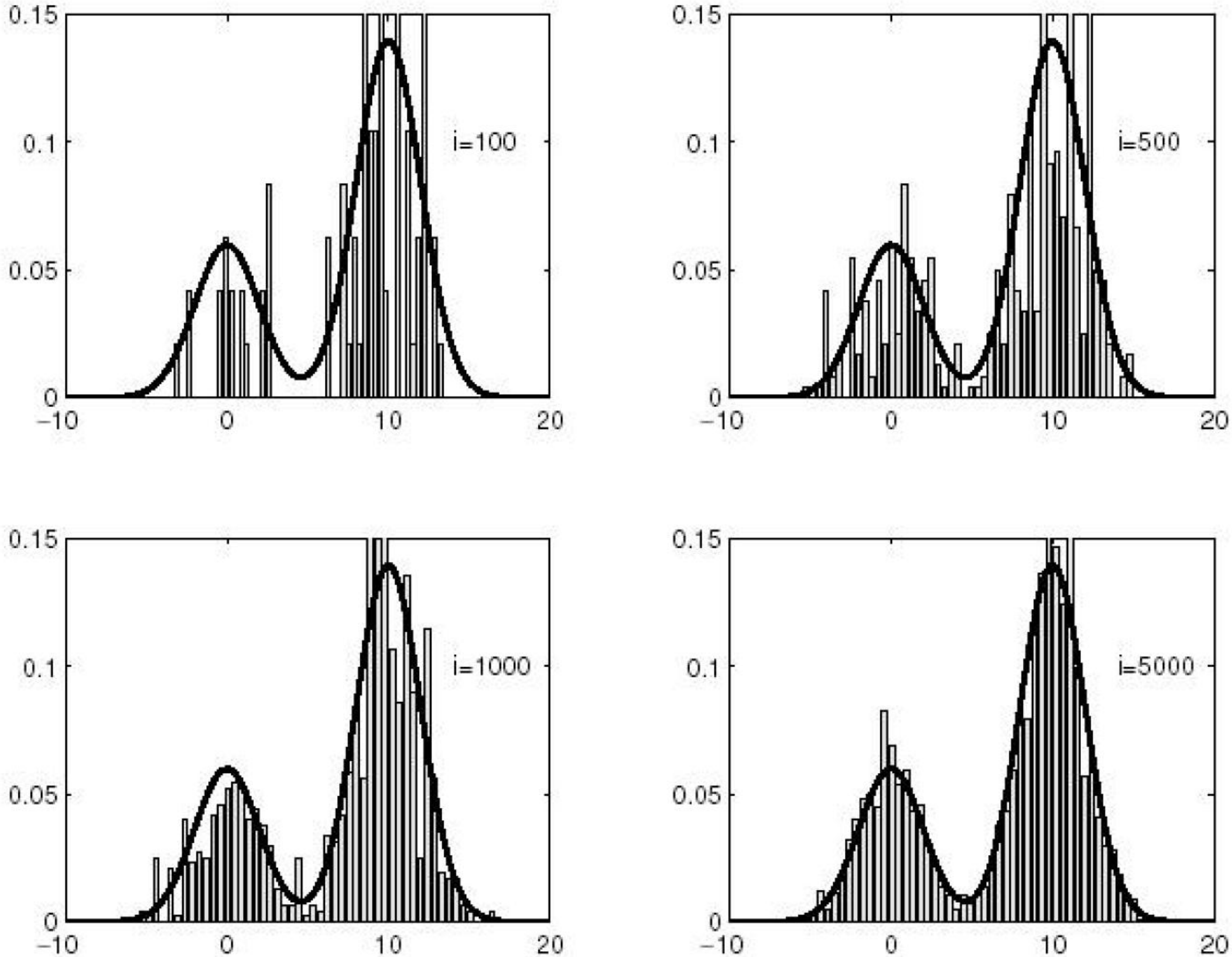
Why will the MC **converge to stationary distribution** ? there is a simple, useful **sufficient** condition:

"Regular " Markov Chain : (for finite state space)
Any state x can reach any other states x' with prob. > 0.
(all entries of Potential/CPD > 0)

→ $X^{(M)}$ follows a unique π_T as M large enough.



Example Result



How to define $T(x \rightarrow x')$? ---- Gibbs Sampling

Gibbs Sampling is the most popular one used in Graphical Model.
In graphical model :

> It is easy to draw sample from **"each individual variable given others $P(X_k | \mathbf{X}_{-k})$"**, while drawing from the **joint dist. of $(X_1, X_2, \ldots, X_K)$** is difficult.

So, we define $T(X \rightarrow X')$ in Gibbs-Sampling as :
Taking transition of $X_1 \sim X_K$ in turn with transition distribution :

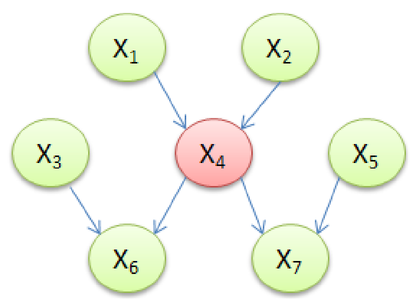
$$T_1(x_1 \rightarrow x_1'),\ T_2(x_2 \rightarrow x_2'),\ \ldots\ldots\ T_K(x_K \rightarrow x_K')$$

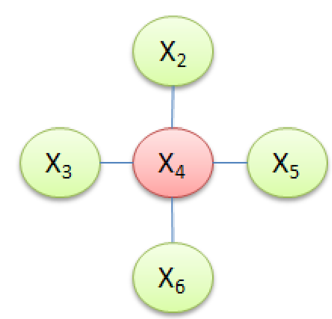
Where

$$T_k(x_k \rightarrow x_k') = P(X_k = x_k' | \mathbf{X}_{-k})$$ (Redraw $X_k \sim$ conditional dist. given all others.)

In a Graphical Model,

$$P(X_k = x_k' | \mathbf{X}_{-k}) = P(X_k = x_k' | \text{Markov Blanket}(X_k))$$





Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

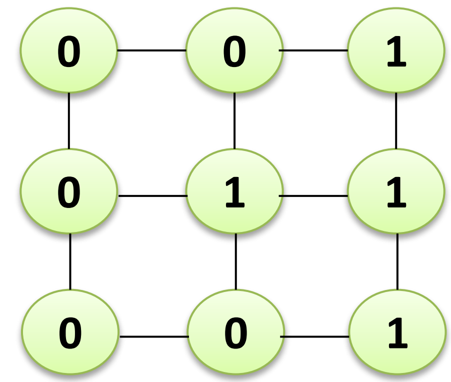
  for every variable X

    2. Draw X_t from $P(X \mid N(X)_{t-1})$.

  end

end

t=1



$$P(X = 1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X = 1, Y)}{\prod_{Y \in N(X)} \phi(X = 1, Y) + \prod_{Y \in N(X)} \phi(X = 0, Y)}$$

$\phi(X,Y)$	0	1
0	5	1
1	1	9

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

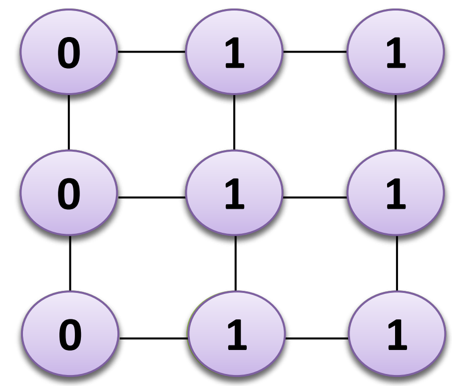
  for every variable X

    2. Draw X_t from $P(X \mid N(X)_{t-1})$.

  end

end

t=2



$$P(X=1 \mid N(X)) = \frac{\prod_{Y \in N(X)} \phi(X=1,Y)}{\prod_{Y \in N(X)} \phi(X=1,Y) + \prod_{Y \in N(X)} \phi(X=0,Y)}$$

For the central node:

$$P(X=1 \mid N(X)) = \frac{1*9*9*1}{1*9*9*1+5*1*1*5} = 0.76$$

ϕ(X,Y)	0	1
0	5	1
1	1	9

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.

for t = 1~M

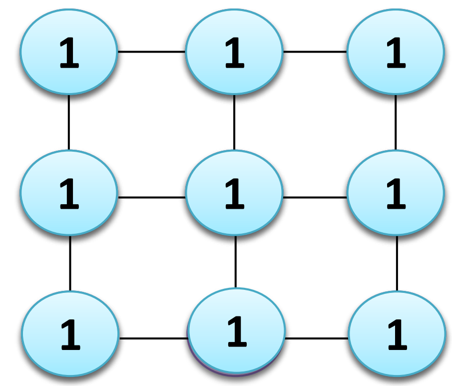
  for every variable X

    2. Draw X_t from P(X | $N(X)_{t-1}$).

  end

end

t=3



$$P(X=1|N(X)) = \frac{\prod_{Y \in N(X)} \phi(X=1,Y)}{\prod_{Y \in N(X)} \phi(X=1,Y) + \prod_{Y \in N(X)} \phi(X=0,Y)}$$

For the central node:

$$P(X=1|N(X)) = \frac{9*9*9*9}{9*9*9*9+1*1*1*1} = 0.99$$

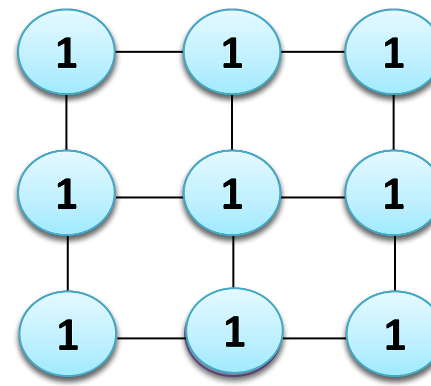
ϕ(X,Y)	0	1
0	5	1
1	1	9

Gibbs Sampling for MRF

Gibbs Sampling :

1. Initialize all variables randomly.
   for $t = 1$~M
   for every variable X
   2. Draw X_t from $P(X \mid N(X)_{t-1})$.
   end
   end

t=3



When M is large enough, $X^{(M)}$ follows stationary dist. :

$$\pi_T(X) = P(X) = \frac{1}{Z}\prod_C \phi(X_C)$$

(Regularity: All entries in the Potential are positive.)

ϕ(X,Y)	0	1
0	5	1
1	1	9

Why Gibbs Sampling has $\boldsymbol{\pi_T = P(X)}$?

To prove P(X) is the stationary distribution, we prove P(X) is invariant under $T_k(x_k \rightarrow x_k')$:

Assume $(X_1,\ldots,X_K)$ currently follows $P(X) = P(X_k|X_{-k}) * P(X_{-k})$,

1. After $T_k(x_k \rightarrow x_k')$, **$X_{-K}$ still follows $P(X_{-k})$** because they are unchanged.

2. After $T_k(x_k \rightarrow x_k') = P(X_k = x_k' | X_{-k})$ (new state indep. from current value x_k)

   ➔ **$X_k(t)$ still follows $P(X_k|X_{-k})$.**

So, after $T_1(x_1 \rightarrow x_1')$, ……, $T_1(x_K \rightarrow x_K')$, **$X=(X_1,\ldots,X_K)$ still follows P(X)**.

(**Uniqueness** & **Convergence** guaranteed from **Regularity** of MC.)

Gibbs Sampling not Always Work

When drawing from individual variable is not possible:

(We can evaluate P(Y|X) but not P(X|Y).)

Non-linear Dependency :

$$P(Y \mid X) = N(w_0 + w_1 X + w_2 X^2, \sigma^2)$$

$$P(Y \mid X) = \mathrm{logistic}\,(w_0 + w_1 X_1)$$

$$P(Y \mid X) = N\left(\sum_{n=1}^{N} \mathrm{K}(\mathrm{X}, \mathrm{X}^{(n)}),\ \sigma^2 \right)\ (\mathrm{kernel\ trick})$$

$$P(X \mid Y) = \frac{P(Y \mid X) P(X)}{\int_X P(Y \mid X) P(X)\, dX}$$

(Intractable Integration)

Large State Space : $(In\ Structure\ \ Learning,\ \mathrm{statespace} = G_1, G_2, G_3 \ldots.)$

$$P(G \mid Data) = \frac{P(Data \mid G) P(G)}{\sum_G P(Data \mid G) P(G)}$$

(Too large state space to do summation)

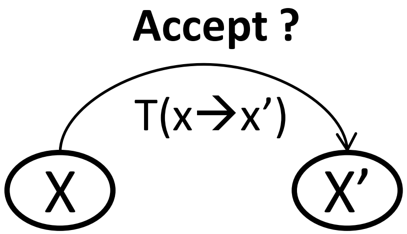
Other MCMC like **Metropolis-Hasting** needed. (see reference.)

Metropolis-Hasting ----MCMC

Metropolis-Hasting (M-H) is a general MCMC method to sample P(X|Y) whenever we can evaluate P(Y|X). (**evaluation of P(X|Y) not needed**)

In M-H, instead of drawing from P(X|Y), we draw from another **Proposal Dist. T(x→x')** based on current sample **x,** and **Accept the Proposal** with probability:

$$P(accept\ \ from\, x\, to\, x') = \begin{cases} 1 & ,\ if\ \ P(x')\mathrm{T}(\mathrm{x'} \rightarrow \mathrm{x}) > P(x)\mathrm{T}(\mathrm{x} \rightarrow \mathrm{x'}) \\ \dfrac{P(x')\mathrm{T}(\mathrm{x'} \rightarrow \mathrm{x})}{P(x)\mathrm{T}(\mathrm{x} \rightarrow \mathrm{x'})} & ,\ \ o.w. \end{cases}$$

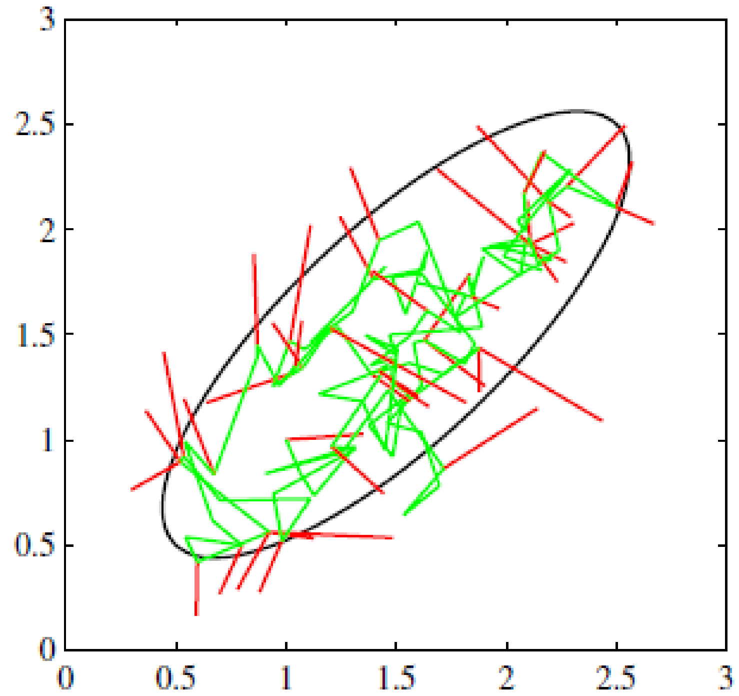


Example : $P(X) = N(\mu,\sigma^2)$

Proposal Dist. $T(x \rightarrow x') = N(x, 0.2^2)$

$$P(accept\ \ from\ x\ to\ x') = \begin{cases} 1\ , & if\ \ |x' - \mu| < |x - \mu| \\ \dfrac{N(x';\mu,\sigma^2)}{N(x;\mu,\sigma^2)}\ , & o.w. \end{cases}$$

$(\ T(x \rightarrow x') = T(x' \rightarrow x)\ \ this\ \ case.\ )$



(red: Reject)

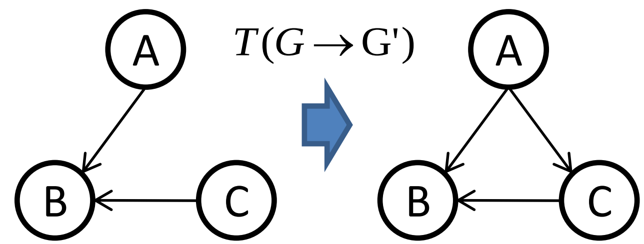
(green: Accept)

Example : Structure Posterior = P(G|Data)

Proposal Distribution:

$T(G \rightarrow G')$

$= \mathrm{P(add/remove\ a\ randomly\ chosen\ edge\ of\ G => G')}$



$$P(accept\ \ from\ G\ to\ G') = \begin{cases} 1\ \ ,\ \ if\ \ P(\mathrm{Data}\,|\,\mathrm{G'}) < P(\mathrm{Data}\,|\,\mathrm{G}) \\ \dfrac{P(\mathrm{Data}\,|\,\mathrm{G'})}{P(\mathrm{Data}\,|\,\mathrm{G})}\ ,\ \ o.w. \end{cases}$$

$(\ T(G \rightarrow G') = T(G' \rightarrow G)\ \ this\ \ case.\ )$

Why Metropolis-Hasting has $\pi_T = P(X)$?

Detailed-Balance Sufficient Condition:

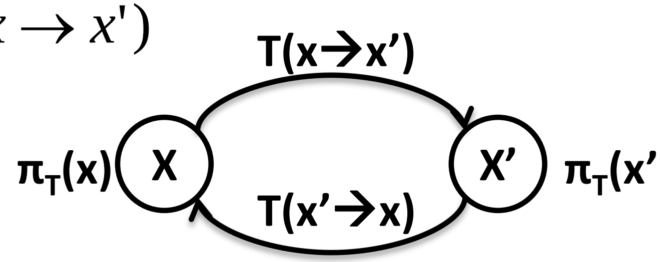
If $\pi_T(x')*T(x' \to x) = \pi_T(x)*T(x \to x')$, then $\pi_T(x)$ is **stationary under T.**

Given desired $\pi_T(x)=P(X)$, and a **Proposal dist.** $T(x \to x')$,
we can let **Detailed Balance** satisfied using **accept prob.** $A(x \to x')$ **:**

$$Assume\ P(x')T(x' \to x) < P(x)T(x \to x')\ ,\ then:$$

$$\text{We know } P(x')T(x' \to x)*1 = P(x)T(x \to x')*\frac{P(x')T(x' \to x)}{P(x)T(x \to x')}$$

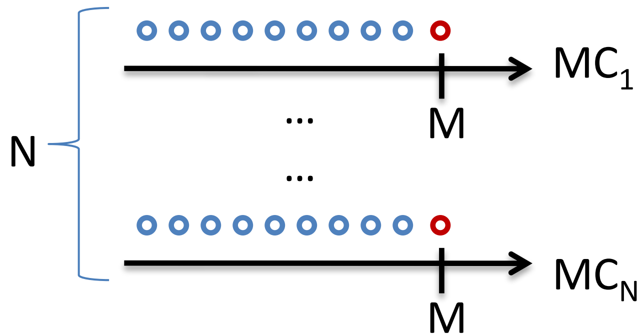
$$define\ \ A(x \to x') = \begin{cases} 1, & P(x')T(x' \to x) > P(x)T(x \to x') \\ \dfrac{P(x')T(x' \to x)}{P(x)T(x \to x')}, & o.w. \end{cases}$$



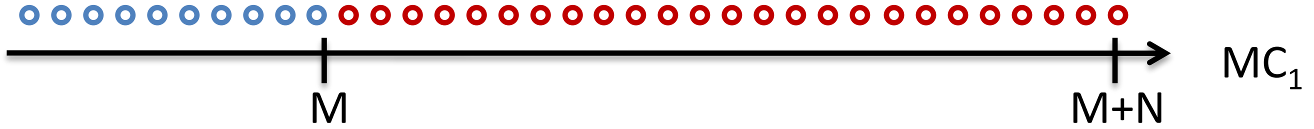
How to Collect Samples ?

Assume we want collecting N samples:

1. Run N times of MCMC and collect their M^{th} samples.



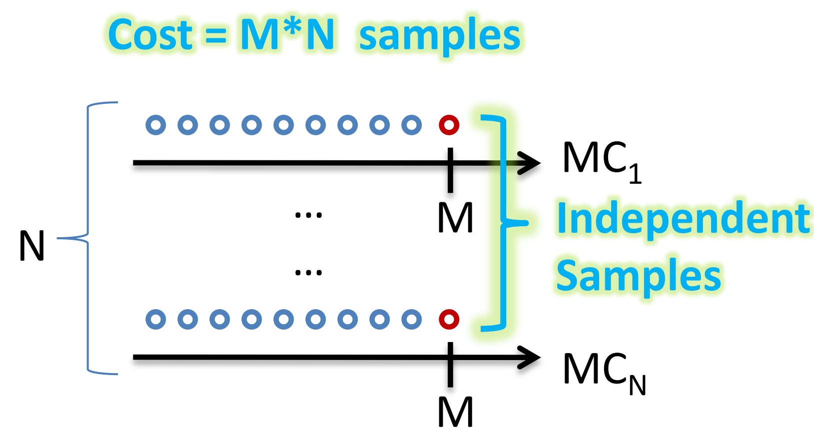
2. Run 1 time of MCMC and collect $(M+1)^{th}$ ~ $(M+N)^{th}$ samples.



What's the difference ??

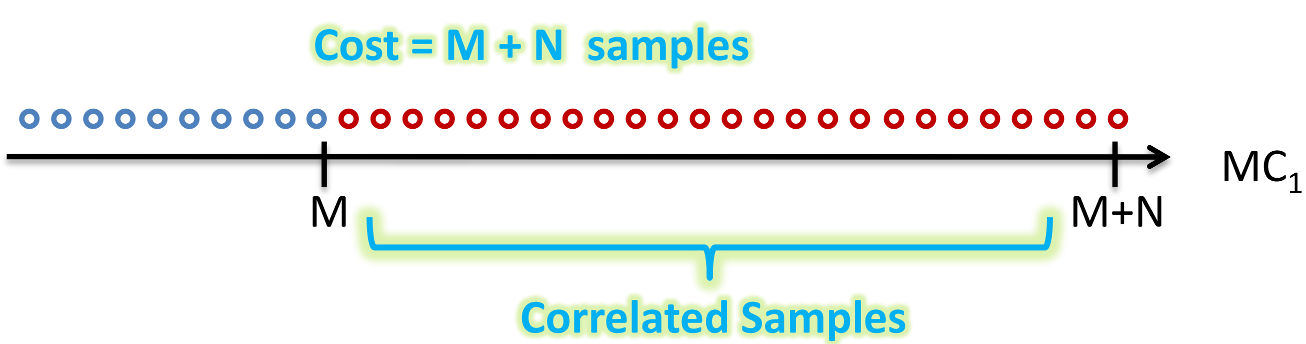
How to Collect Samples ?

Assume we want collecting N samples:

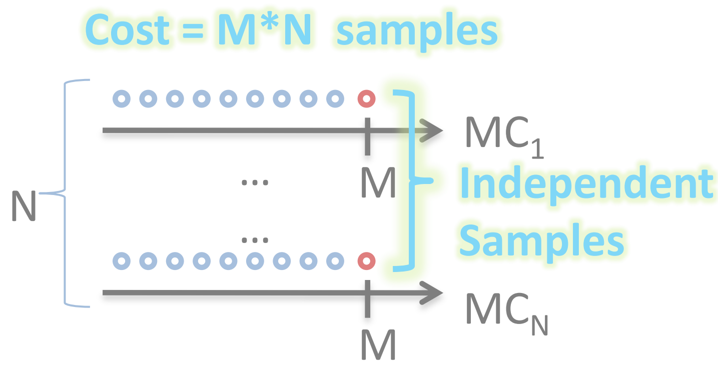


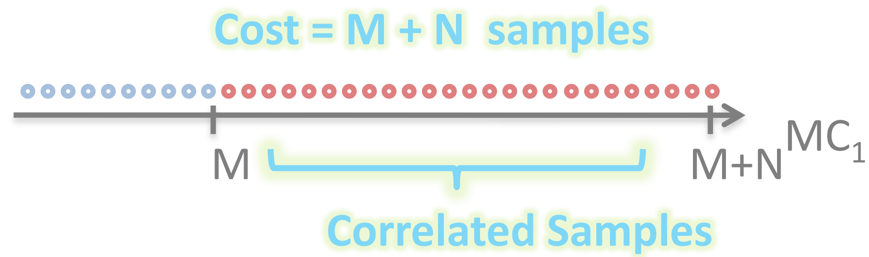
1. Run N times of MCMC and collect their M^{th} samples.

2. Run 1 time of MCMC and collect $(M+1)^{th}$ ~ $(M+N)^{th}$ samples.



Comparison





$$E[\hat{f}] = E[\frac{1}{N}\sum_{n=1}^{N} f(X^{(n)})] = \frac{1}{N} E[\sum_{n=1}^{N} f(X^{(n)})] = \frac{1}{N}\sum_{n=1}^{N} E[f(X^{(n)})] = E[f(X)]$$

No Independent Assumption Used ➔ **Unbiased Estimator in both cases.**

For simple analysis, Take N=2:

$$Var[\hat{f}] = Var[\frac{1}{2}(f(X^{(1)}) + f(X^{(2)})]$$

$$= \frac{1}{4}(Var[f(X^{(1)})] + Var[f(X^{(2)})]) = \frac{Var[f(X)]}{2}$$

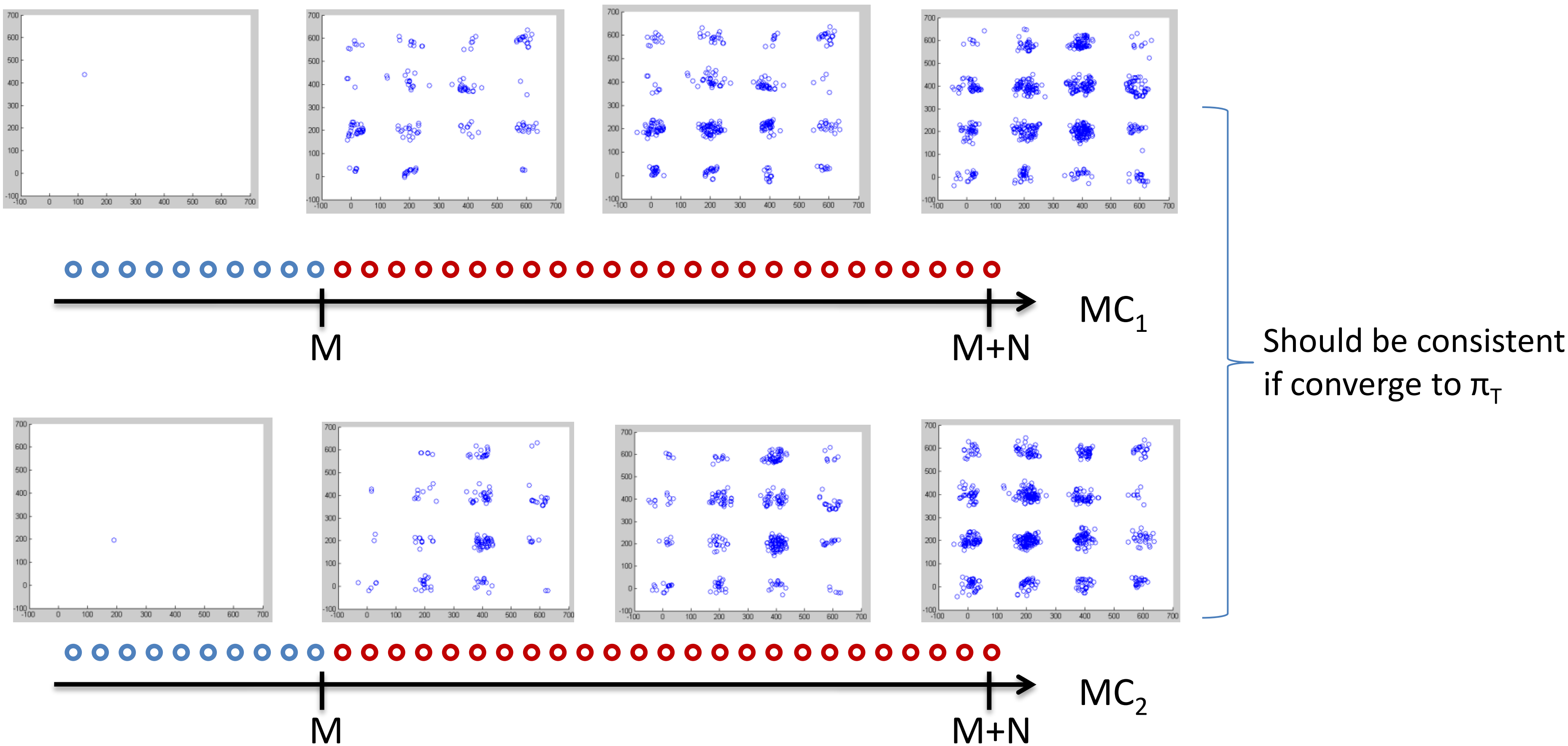
$$Var[\hat{f}] = Var[\frac{1}{2}(f(X^{(1)}) + f(X^{(2)})]$$

$$= \frac{1}{4}(Var[f(X^{(1)})] + Var[f(X^{(2)})] + 2Cov[f(X^{(1)}), f(X^{(2)})])$$

$$= \frac{Var[f(X)]}{2} + \rho_{f(X^{(1)}), f(X^{(2)})} * \frac{Var[f(X)]}{2} > \frac{Var[f(X)]}{2}$$

Practically, **many correlated samples** (right) outperforms **few independent samples** (left).

How to Check Convergence ?

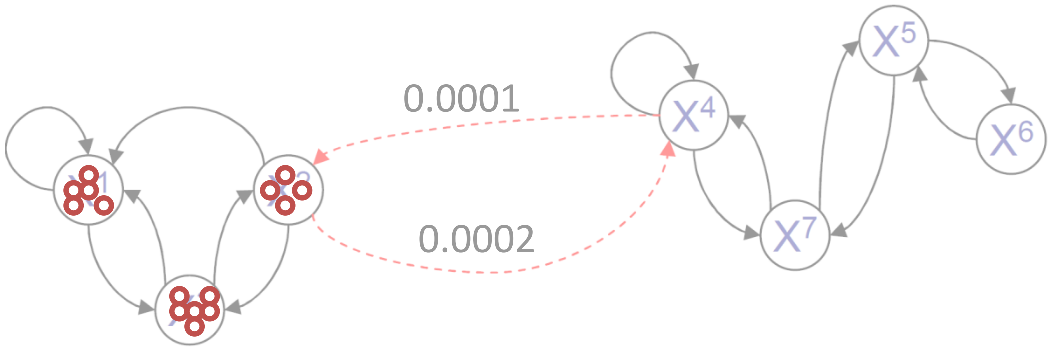


$$Check\ \text{Ratio} = \sqrt{\frac{\text{B}}{\text{W}}}\ \ close\ to\ 1\ enough. \quad \text{(assume K MCs, each with N samples.)} \qquad \bar{f} = \frac{1}{K}\sum_{k=1}^{K}\bar{f}_k$$

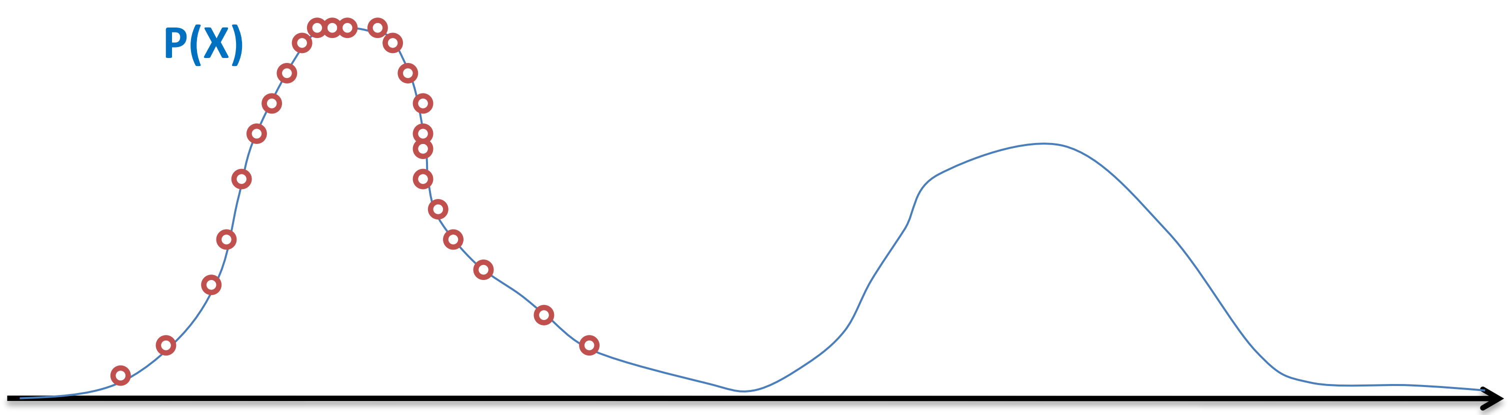
$$B = Var.\text{between}\ \ \text{MC} = \frac{\text{N}}{\text{K - 1}}\sum_{k=1}^{K}(\bar{f}_k - \bar{f})^2 \qquad \text{W} = Var.\text{within}\ MC = \frac{1}{K(N-1)}\sum_{k=1}^{K}\sum_{n=1}^{N}(f(X^{(k,n)}) - \bar{f}_k)^2$$

The Critical Problem of MCMC

When **ρ→1 , M→∞ , Var[.] not decreasing with N**
➔ MCMC cannot yield acceptable result in reasonable time.



Taking very large M to converge to $\boldsymbol{\pi_T}$.



How to Reduce Correlation (ρ) among Samples ?

Taking Large Step in Sample Space :

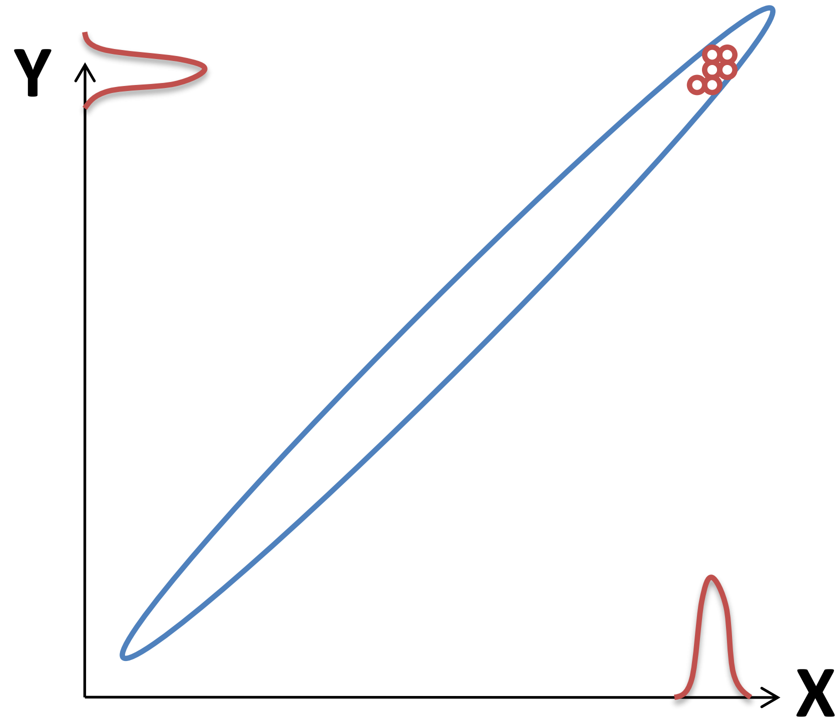
- Block Gibbs Sampling
- Collapsed-Particle Sampling

Problem of Gibbs Sampling

Correlation (ρ) between samples is high,
when correlation among variables X_1~X_K is high.

X **0**

Y	φ(Y,X)	0	1
0	0	99	1
	1	1	99



Taking very large M to converge to π_T .

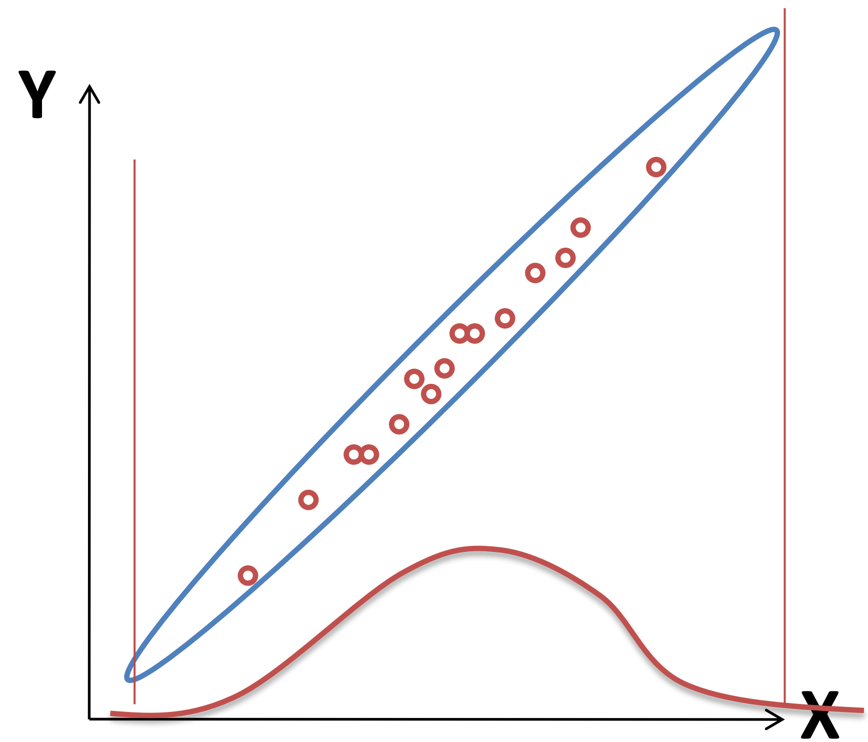
Block Gibbs Sampling

Draw "block" of variables jointly: P(X,Y)=P(X)P(Y|X)

Marginal

X	100	100

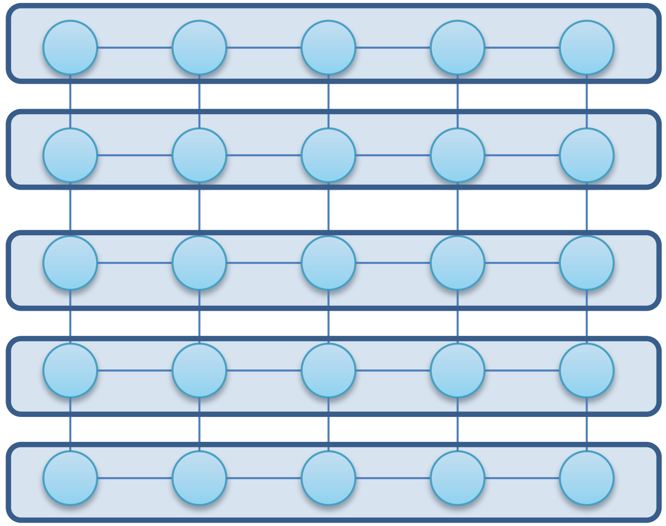
Y	φ(Y,X)	0	1
	0	99	1
	1	1	99

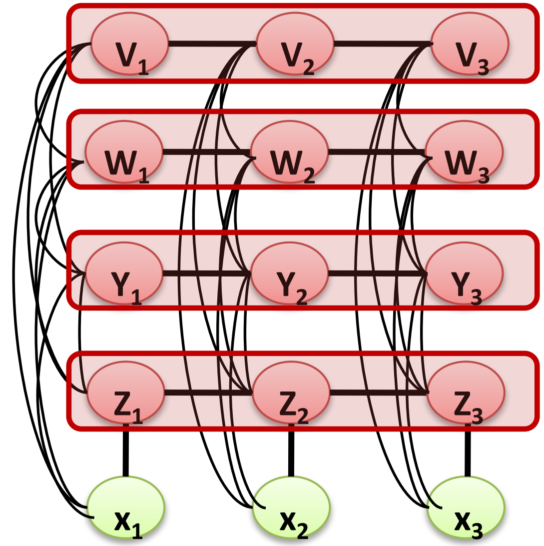


Converge to π_T much quickly.

Block Gibbs Sampling

Divide **X** into several "**tractable blocks**" $\mathbf{X_1}$, $\mathbf{X_2}$, …, $\mathbf{X_B}$.
Each block $\mathbf{X_b}$ can be **drawn jointly** given variables in other blocks.

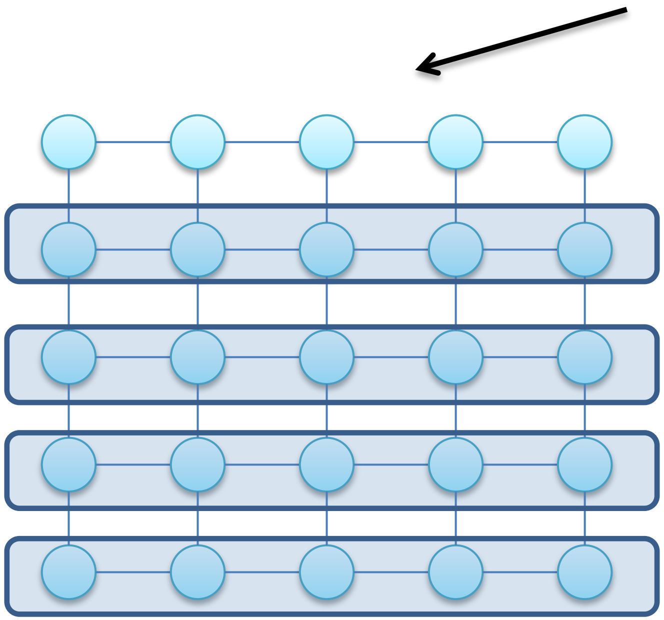


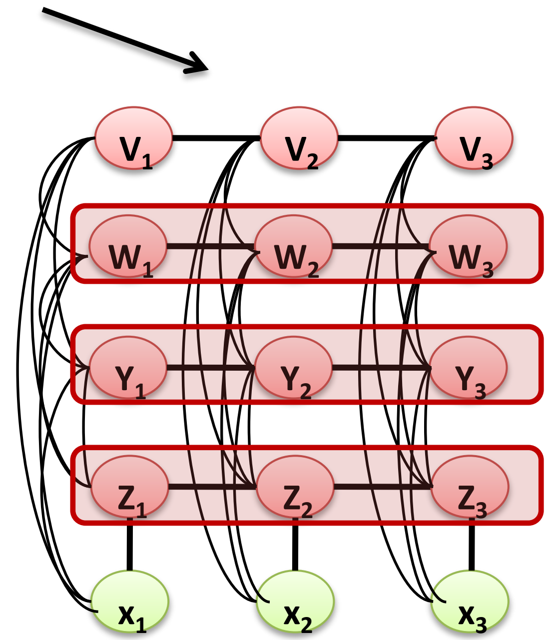


Block Gibbs Sampling

Divide **X** into several "**tractable blocks**" $\mathbf{X_1}$, $\mathbf{X_2}$, ..., $\mathbf{X_B}$.
Each block $\mathbf{X_b}$ can be **drawn jointly** given variables in other blocks.

Given samples on other blocks,
Drawing a block jointly from is tractable.

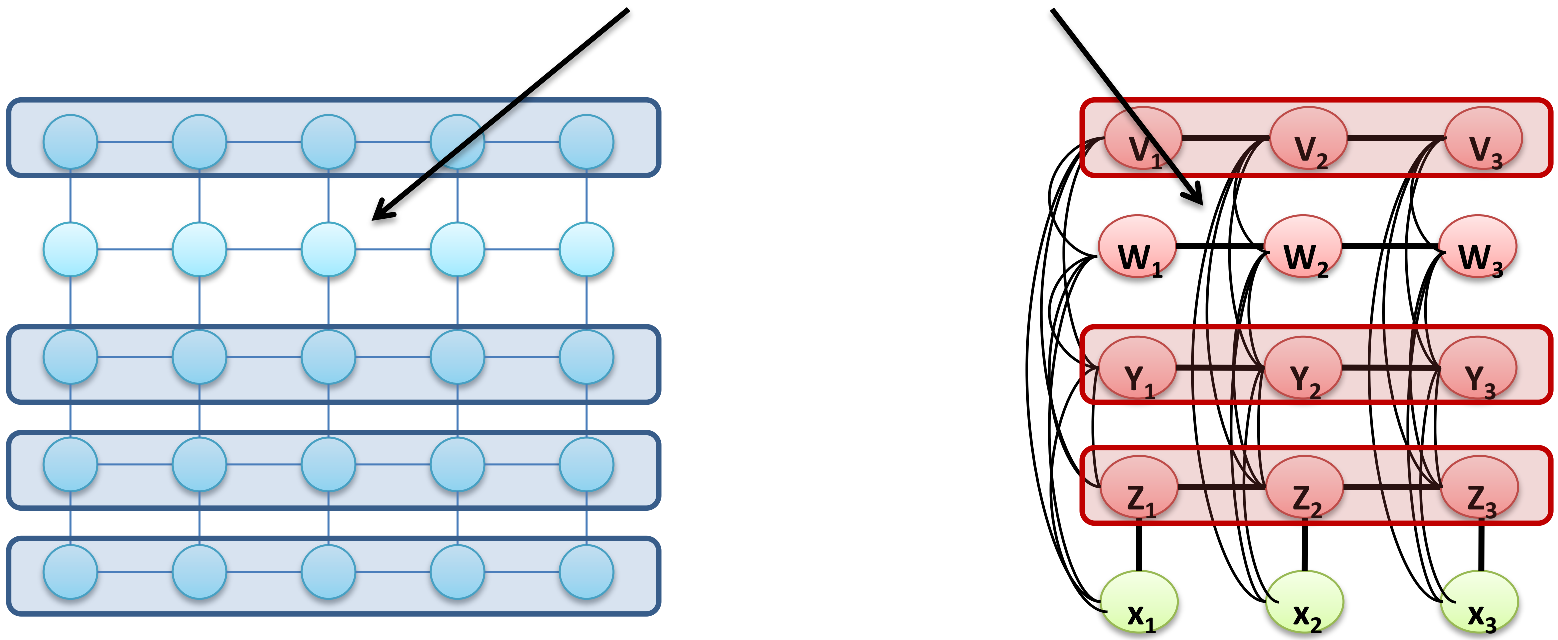




Block Gibbs Sampling

Divide **X** into several "**tractable blocks**" $\mathbf{X_1}$, $\mathbf{X_2}$, ..., $\mathbf{X_B}$.
Each block $\mathbf{X_b}$ can be **drawn jointly** given variables in other blocks.

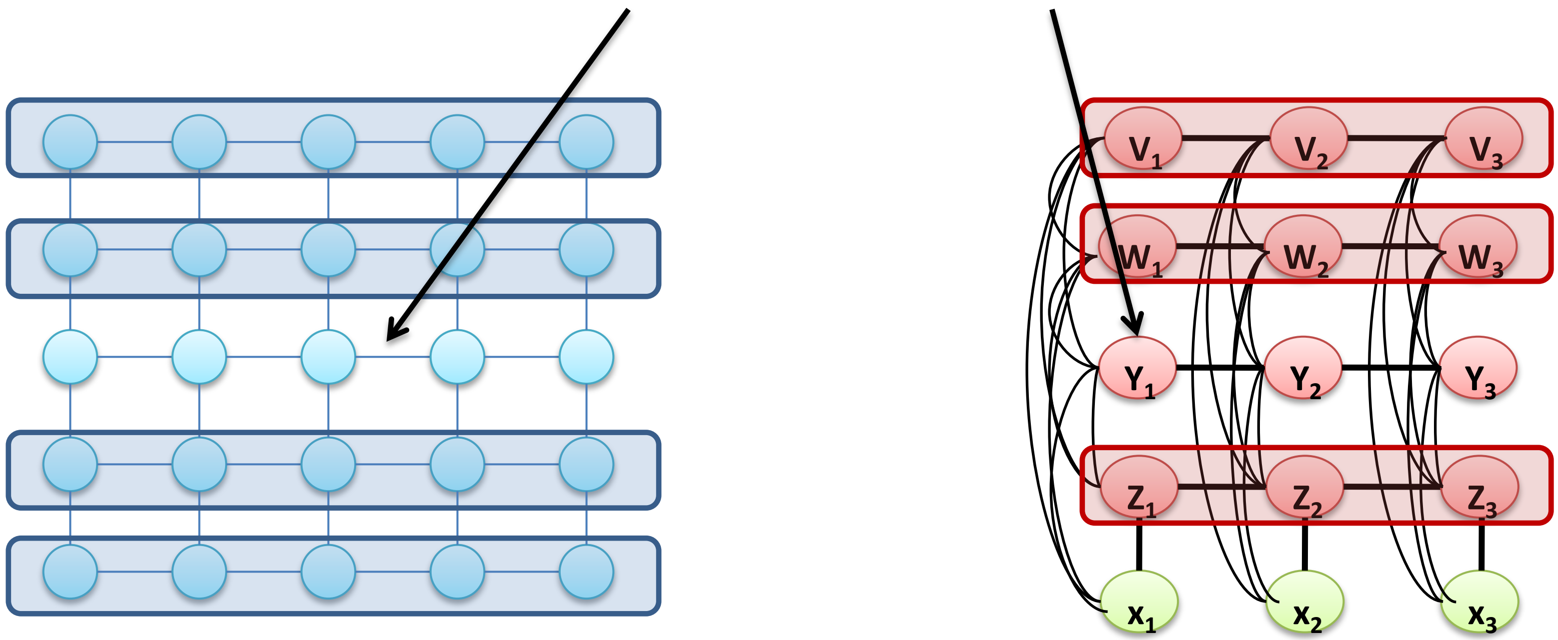
Given samples on other blocks,
Drawing a block jointly from is tractable.



Block Gibbs Sampling

Divide **X** into several “**tractable blocks**” $\mathbf{X_1}$, $\mathbf{X_2}$, …, $\mathbf{X_B}$.
Each block $\mathbf{X_b}$ can be **drawn jointly** given variables in other blocks.

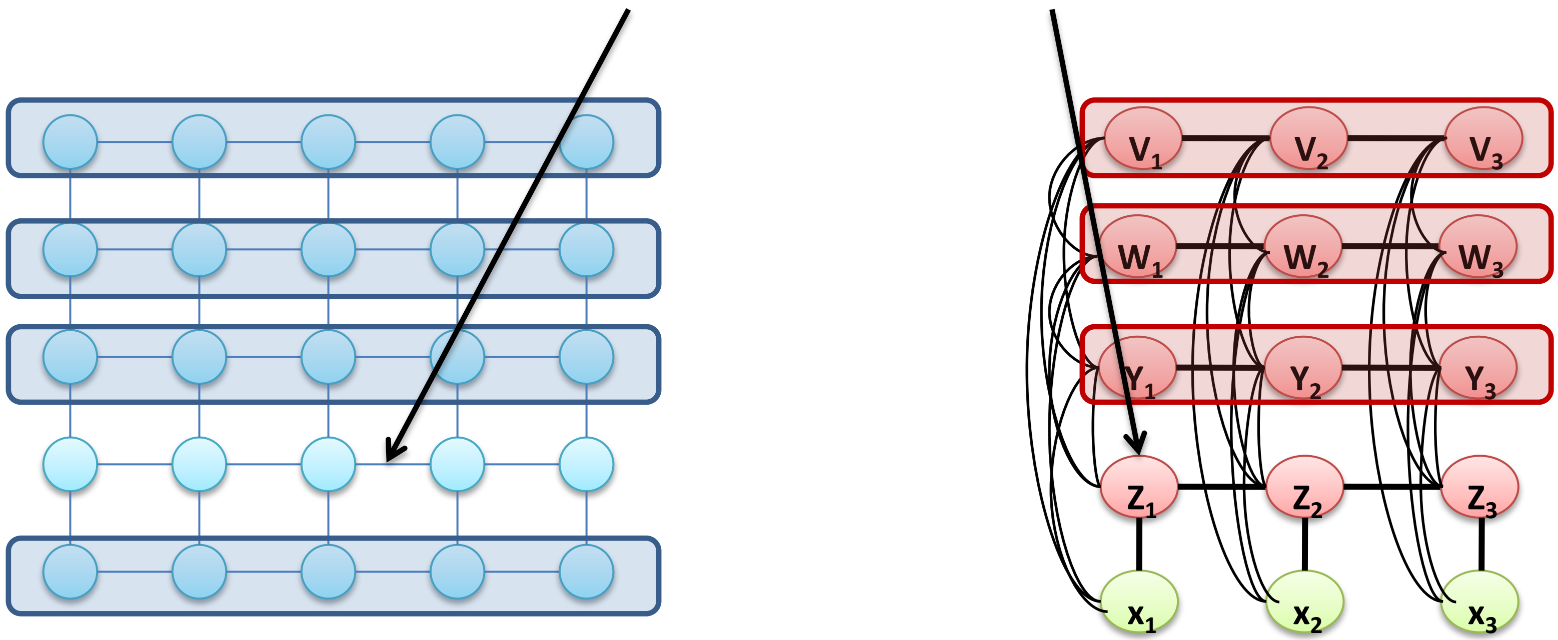
Given samples on other blocks,
Drawing a block jointly from is tractable.



Block Gibbs Sampling

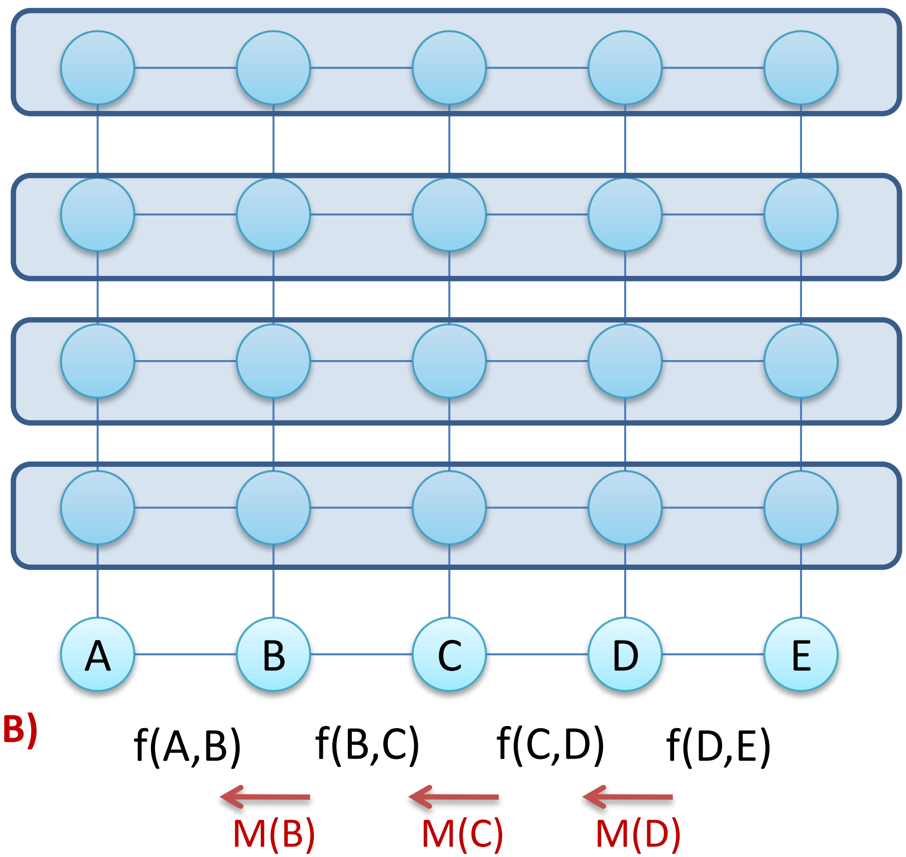
Divide **X** into several "**tractable blocks**" $\mathbf{X_1}$, $\mathbf{X_2}$, …, $\mathbf{X_B}$.
Each block $\mathbf{X_b}$ can be **drawn jointly** given variables in other blocks.

Given samples on other blocks,
Drawing a block jointly from is tractable.



Block Gibbs Sampling by VE

Drawing from a block X_b **jointly** may need 1 pass of VE.

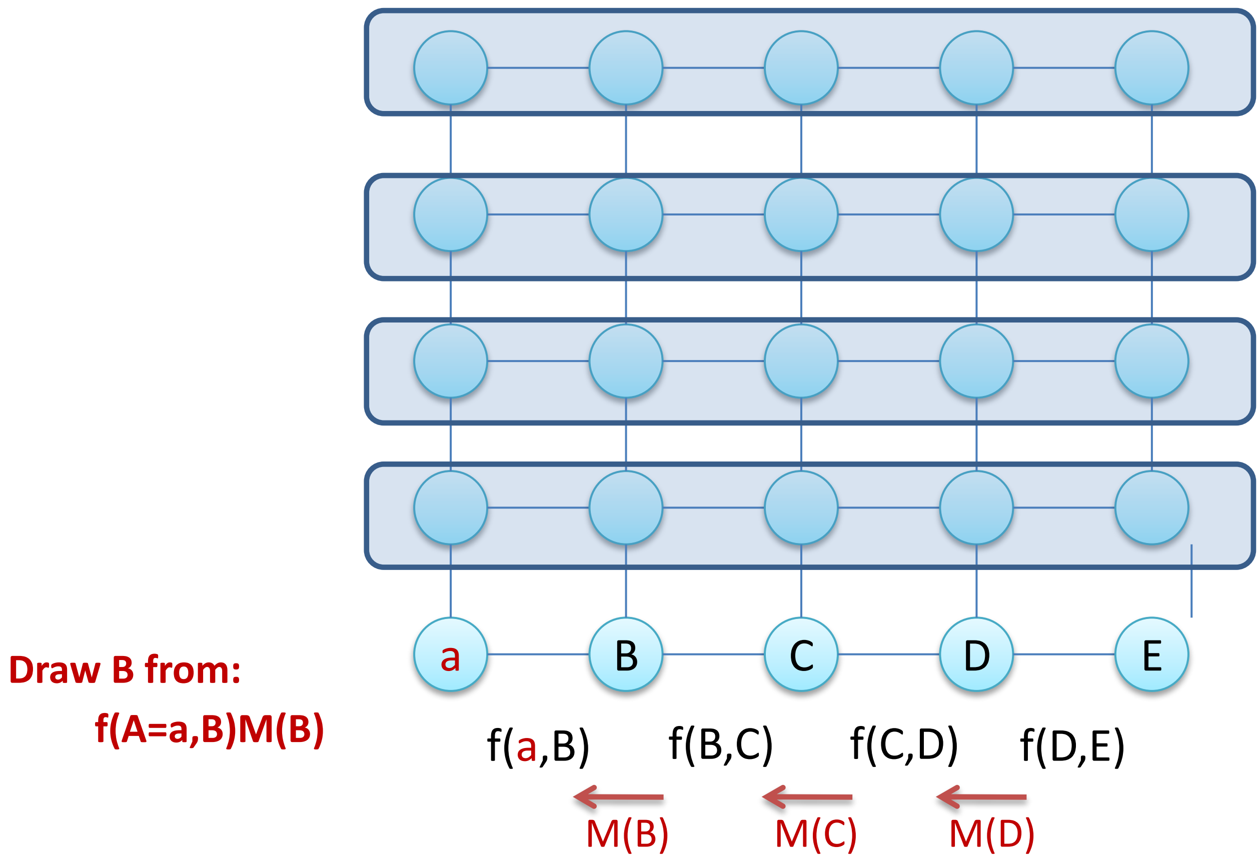


Draw A from:

$M(A)=\sum_B f(A,B)M(B)$

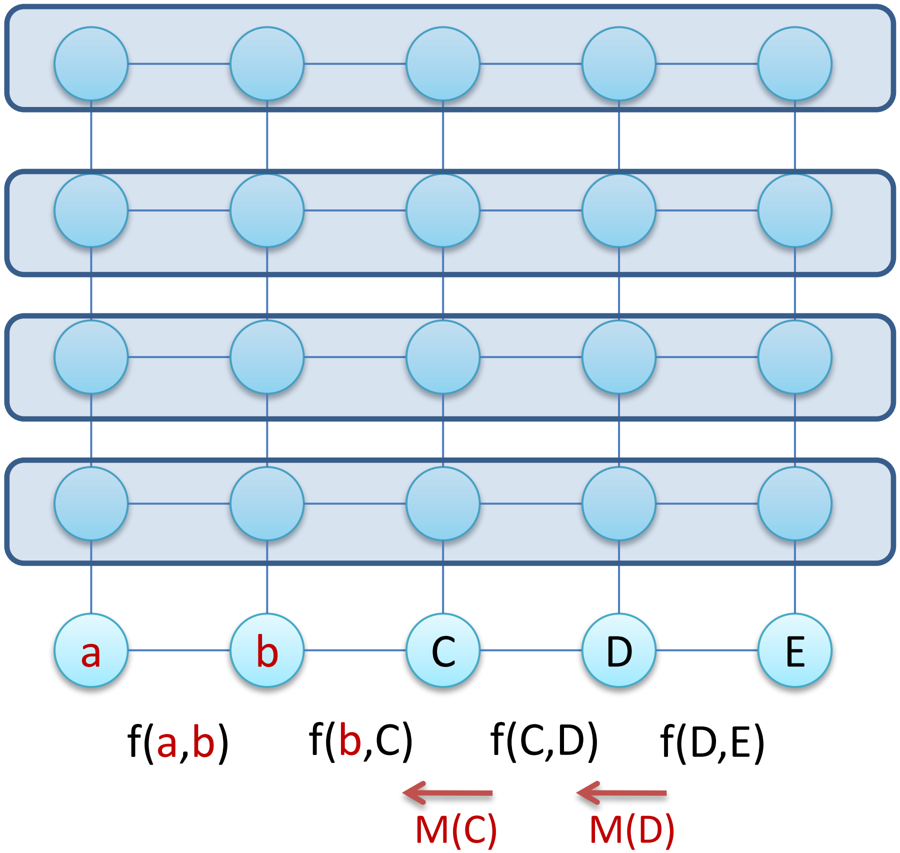
Block Gibbs Sampling by VE

Drawing from a block X_b **jointly** may need 1 pass of VE.



Block Gibbs Sampling by VE

Drawing from a block X_b **jointly** may need 1 pass of VE.

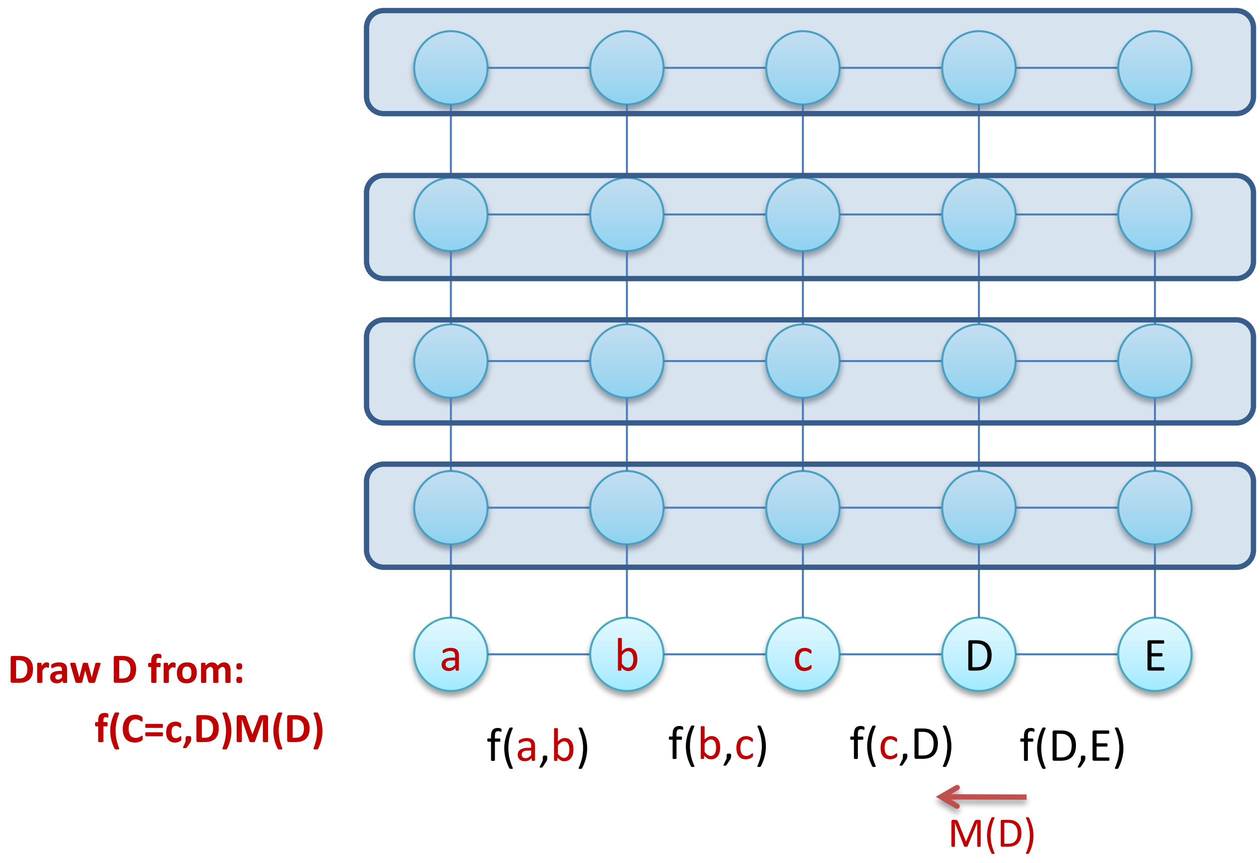


Draw C from:

f(B=b,C)M(C)

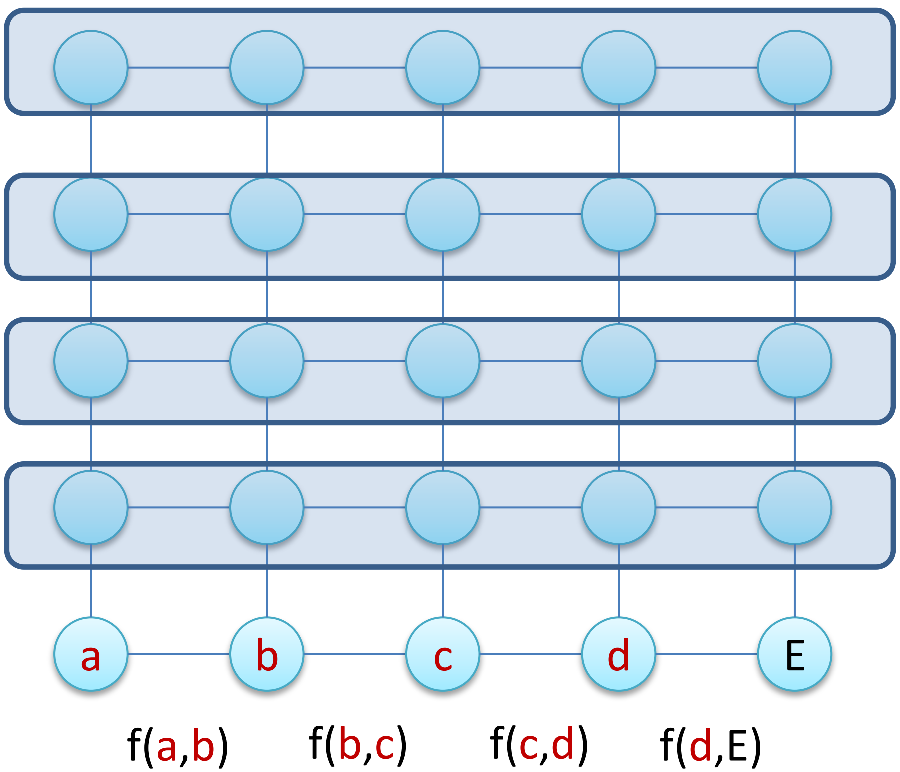
Block Gibbs Sampling by VE

Drawing from a block X_b **jointly** may need 1 pass of VE.



Block Gibbs Sampling by VE

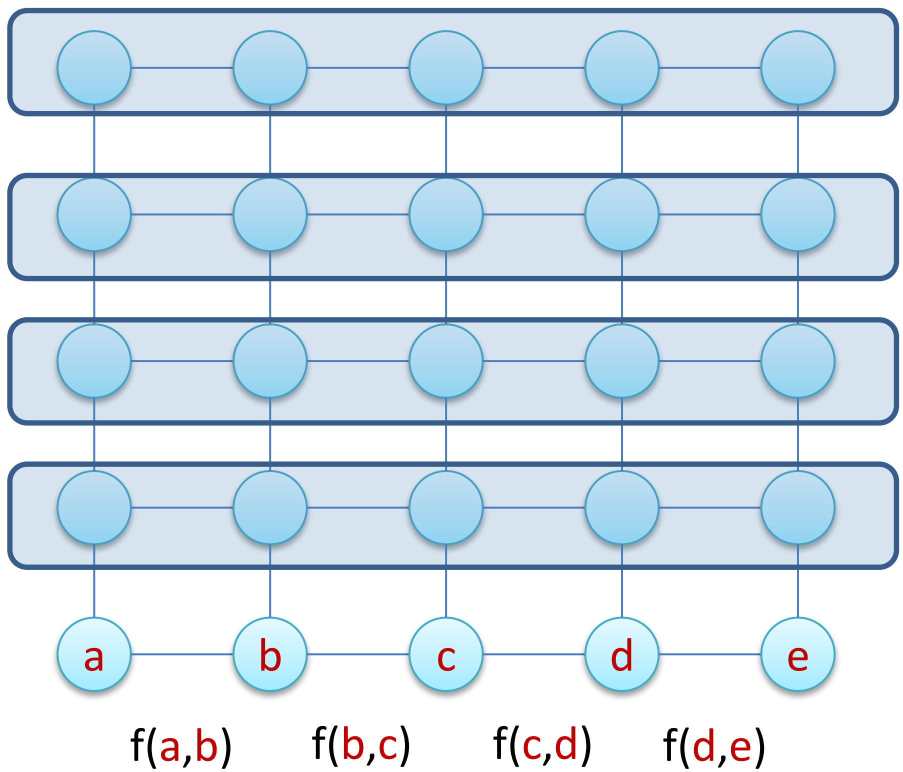
Drawing from a block $\mathbf{X_b}$ **jointly** may need 1 pass of VE.



Draw E from: f(D=d,E)

Block Gibbs Sampling by VE

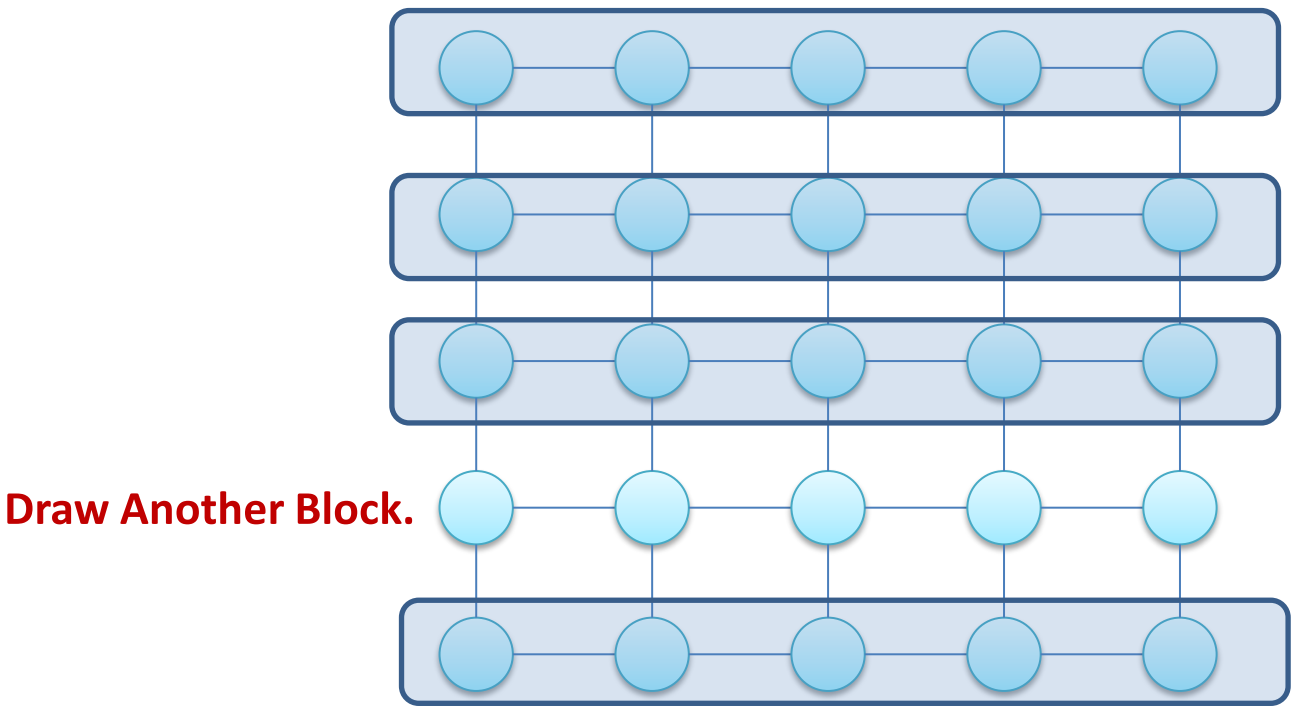
Drawing from a block X_b **jointly** may need 1 pass of VE.

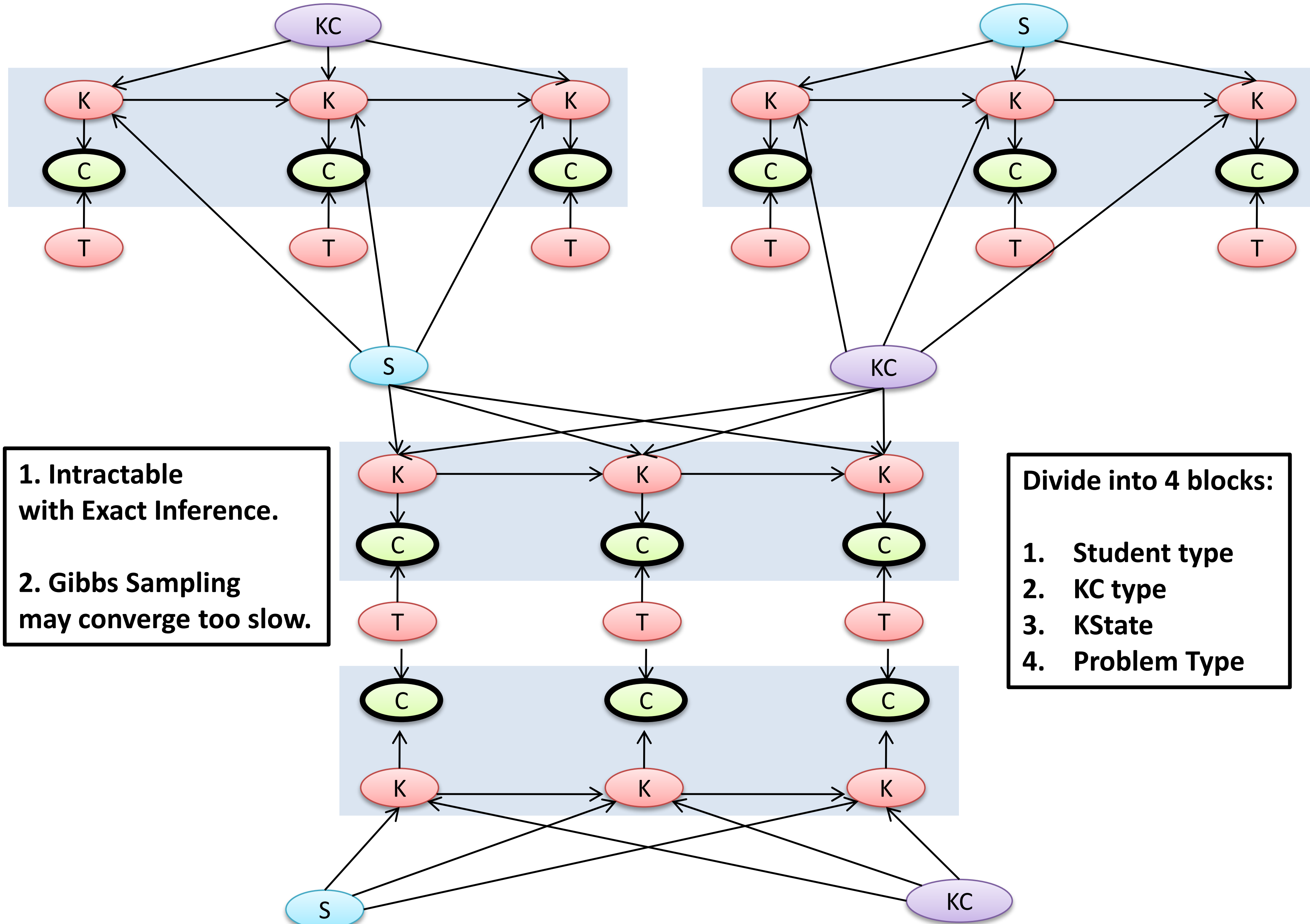


Draw E from:
f(D=d,E)

Block Gibbs Sampling by VE

Drawing from a block $\mathbf{X_b}$ **jointly** may need 1 pass of VE.





1. Intractable
with Exact Inference.

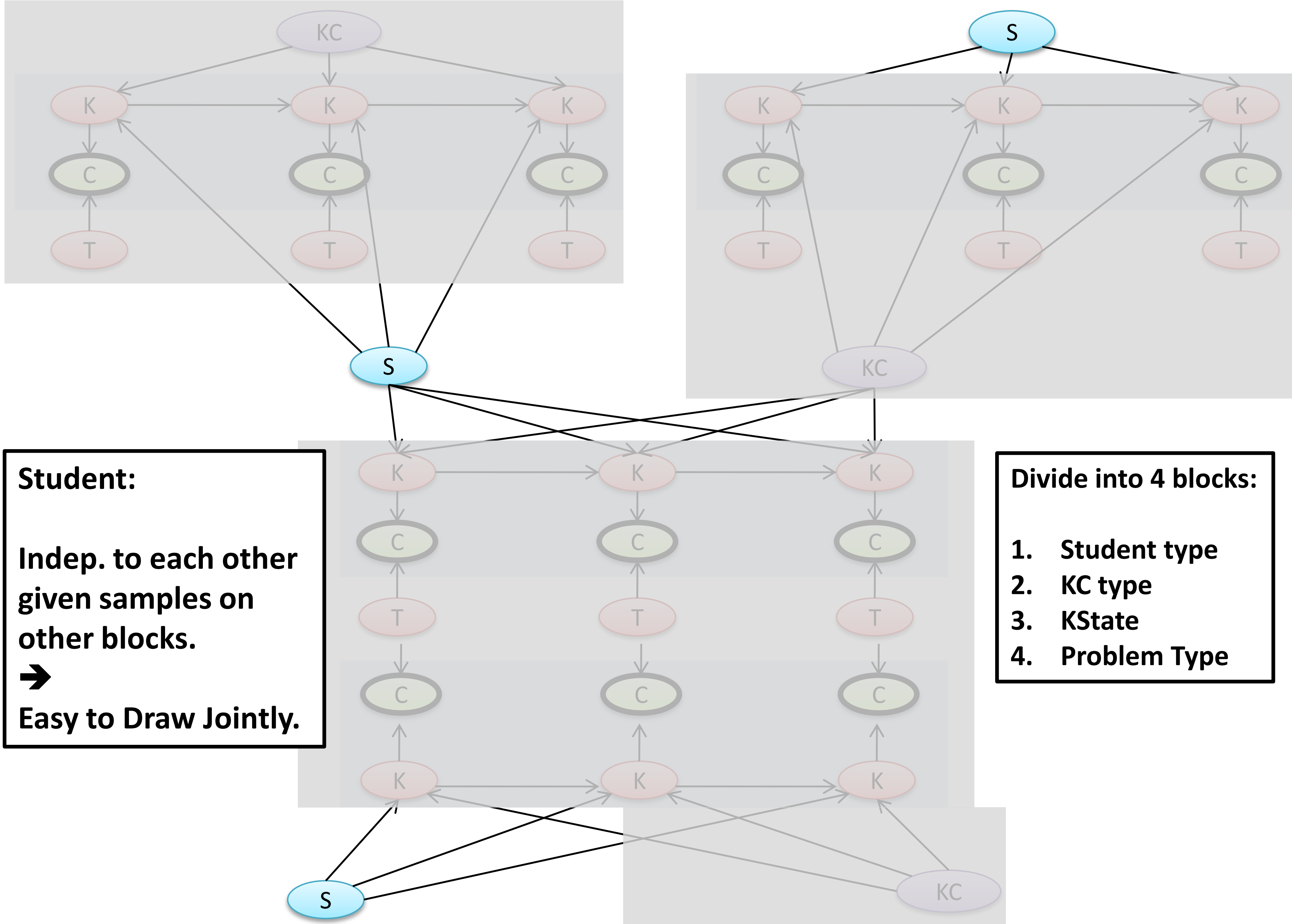
2. Gibbs Sampling
may converge too slow.

Divide into 4 blocks:

1. Student type
2. KC type
3. KState
4. Problem Type

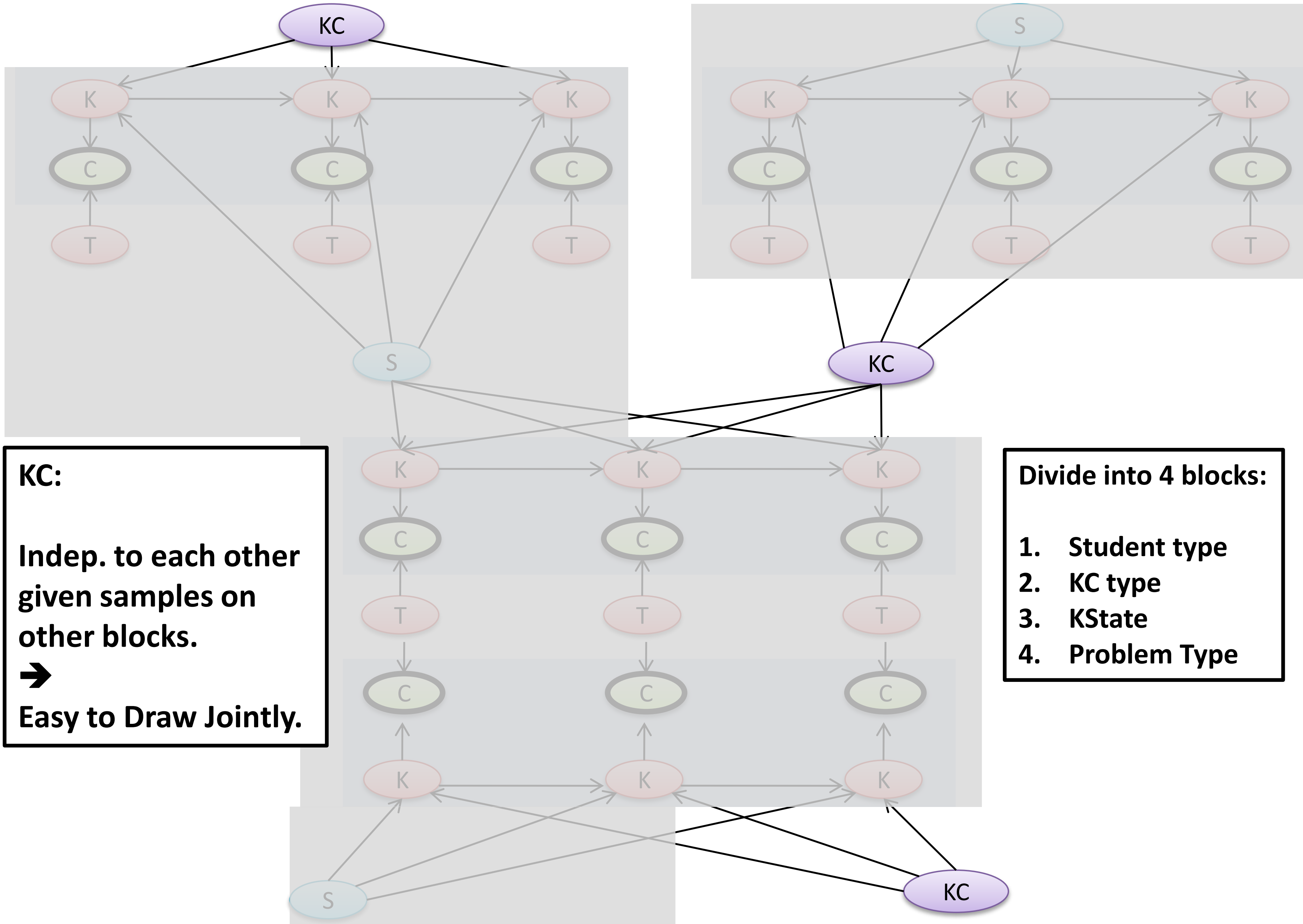
Student:

Indep. to each other given samples on other blocks.
➔
Easy to Draw Jointly.



Divide into 4 blocks:

1. **Student type**
2. **KC type**
3. **KState**
4. **Problem Type**

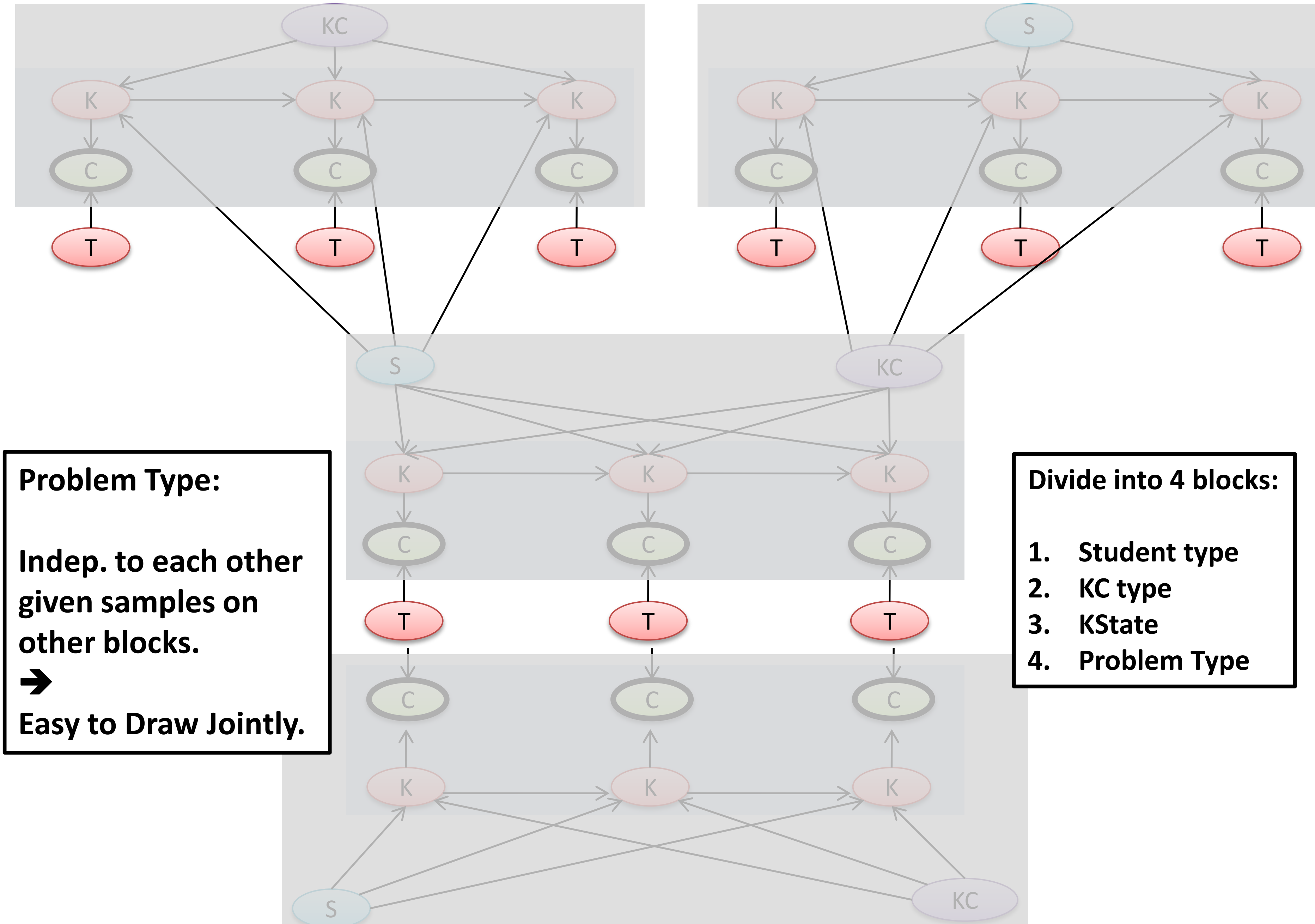


KC:

Indep. to each other given samples on other blocks.
➔
Easy to Draw Jointly.

Divide into 4 blocks:

1. **Student type**
2. **KC type**
3. **KState**
4. **Problem Type**

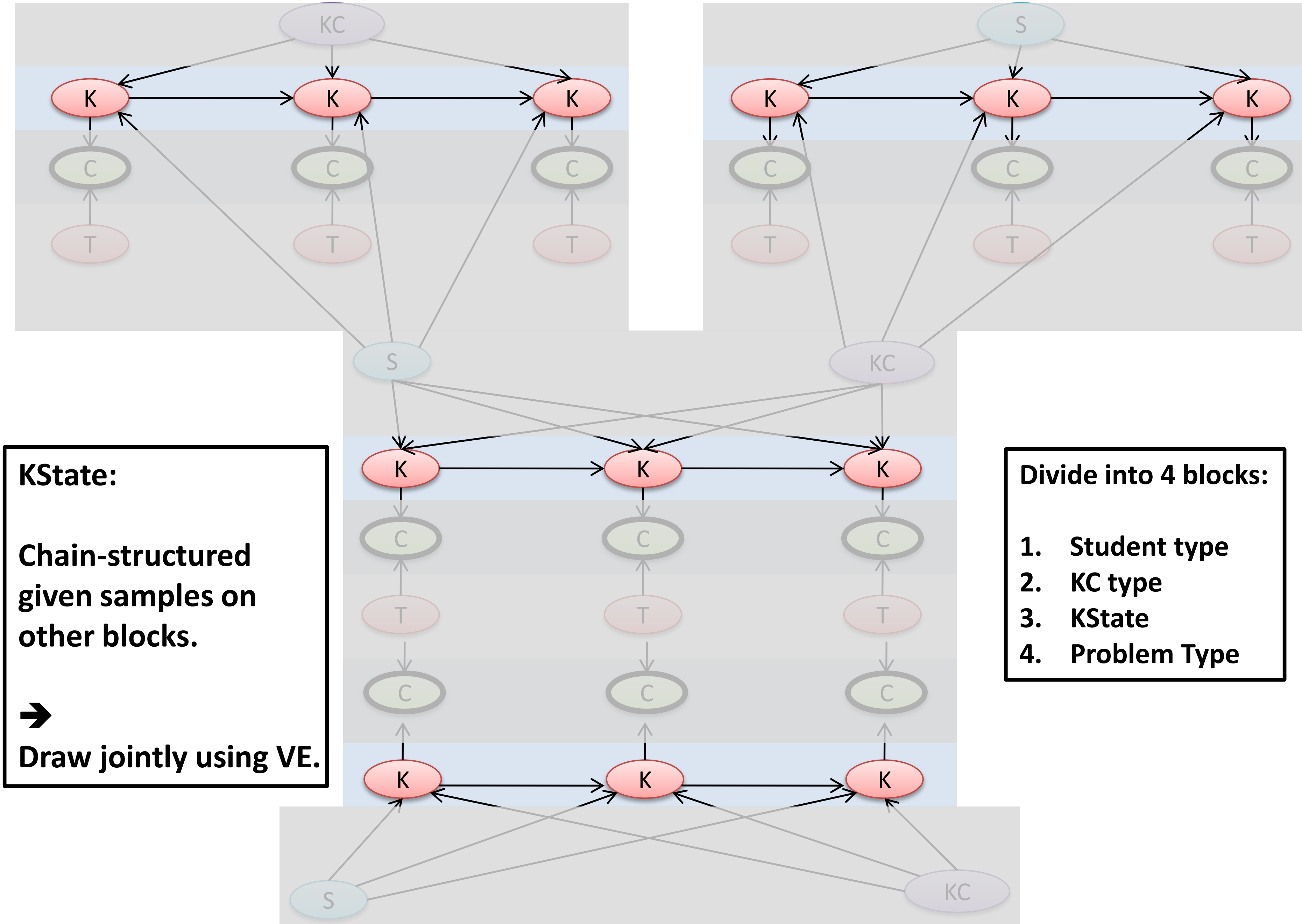


Problem Type:

Indep. to each other given samples on other blocks.
➔
Easy to Draw Jointly.

Divide into 4 blocks:

1. **Student type**
2. **KC type**
3. **KState**
4. **Problem Type**



KState:

Chain-structured given samples on other blocks.

➔
Draw jointly using VE.

Divide into 4 blocks:

1. **Student type**
2. **KC type**
3. **KState**
4. **Problem Type**

Agenda

- When to use Approximate Inference ?
- Forward Sampling & Importance Sampling
- Markov Chain Monte Carlo (MCMC)
- Collapsed Particles

Collapsed Particle

Exact: $E_{P(X)}[f(X)] = \sum_X P(X) * f(X)$

Particle-Based: $\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f(X^{(n)})$

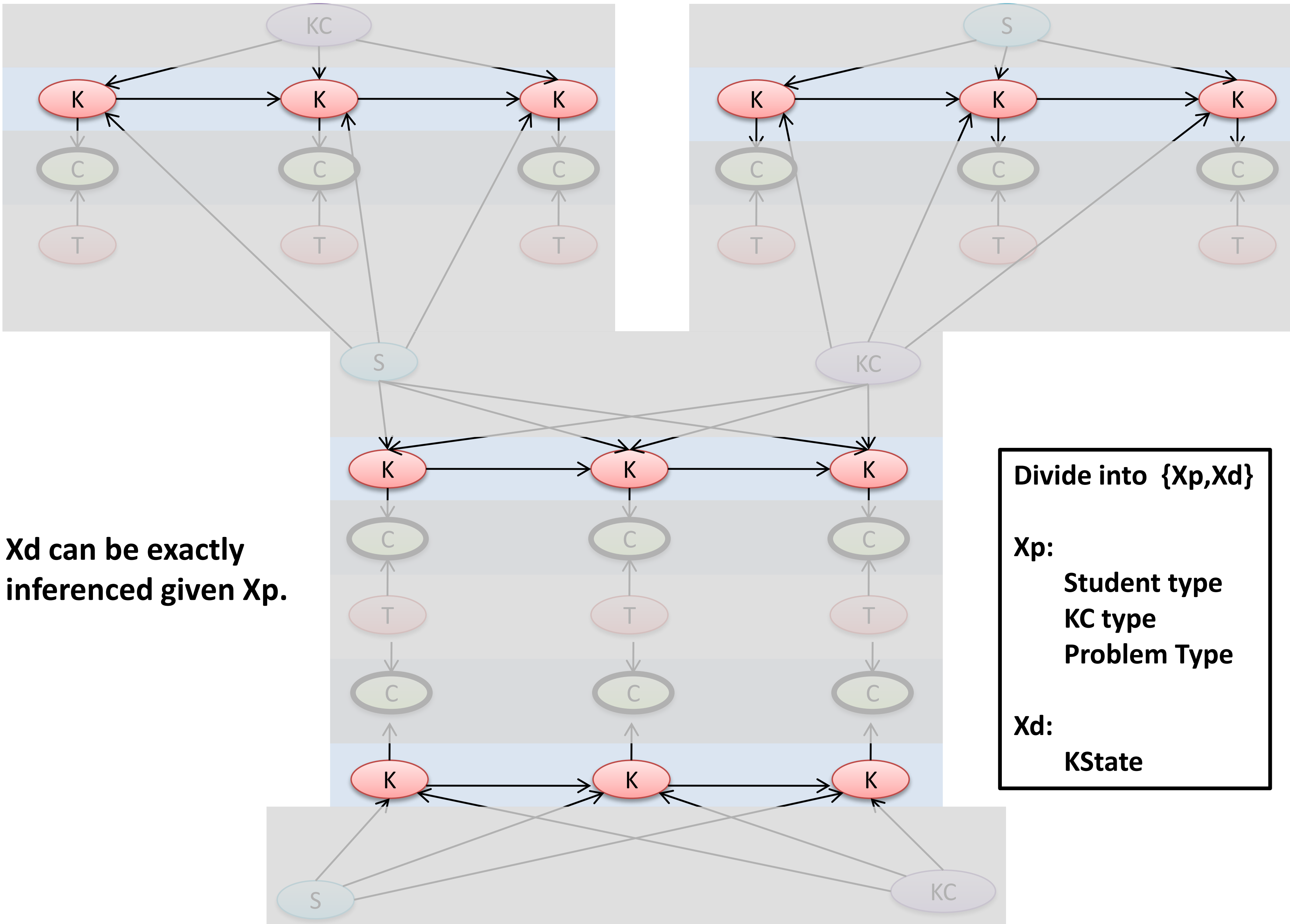
Collapsed-Particle:

Divide X into 2 parts {X_p, X_d} , where X_d can do inference given X_p

$$E_{P(X)}[f(X)] = \sum_X P(X) * f(X) = \sum_{Xp} P(X_p) \sum_{Xd} P(X_d \mid X_p) * f(X)$$

$$\hat{E}_{P(X)}[f(X)] = \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{Xd} P(X_d \mid X_p^{(n)}) f(X_d, X_p^{(n)}) \right)$$

(If $\mathbf{X_p}$ contains few variables, Var. can be much reduced !!)



Xd can be exactly inferenced given Xp.

Divide into {Xp,Xd}

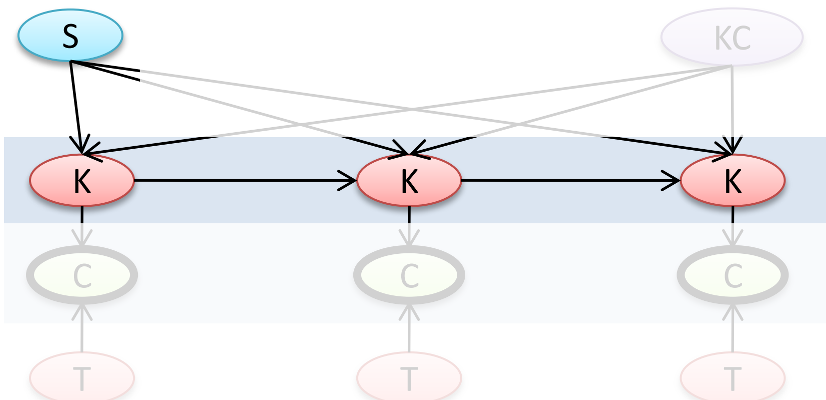
Xp:

- **Student type**
- **KC type**
- **Problem Type**

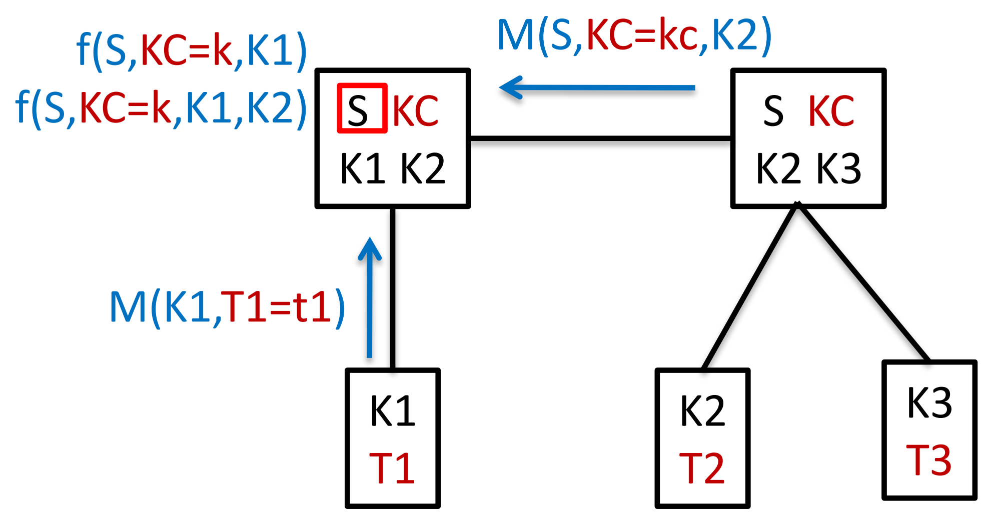
Xd:

- **KState**

Collapsed Particle with VE



To draw $\mathbf{X_k}$,
Given all other variables in **Xp**
sum out all other variables in **Xd**



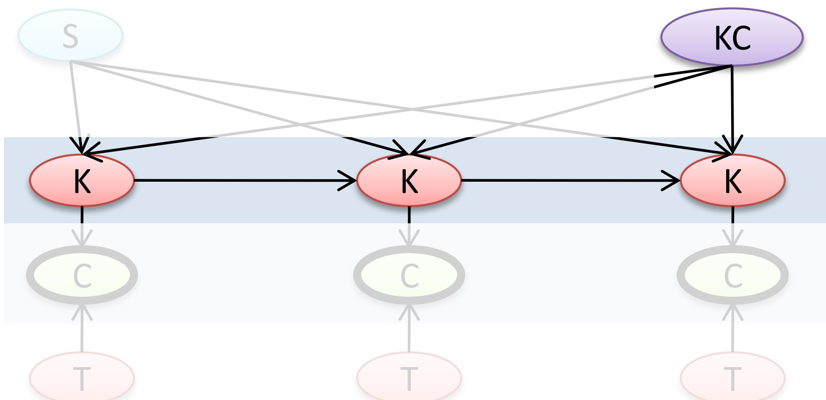
Draw S (given KC=k & T=t) from:

M(S)=

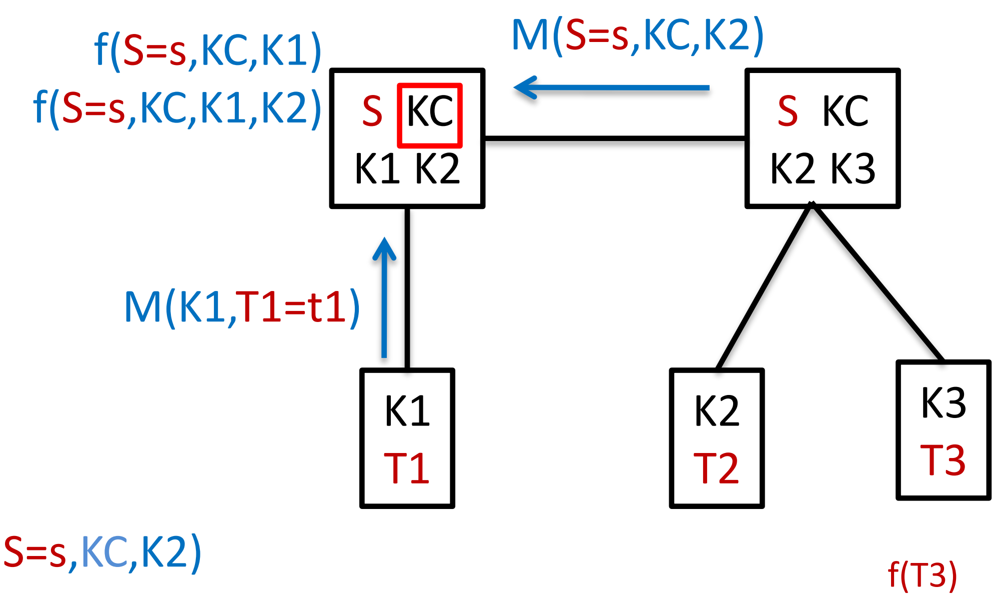
$\sum_{K1,K2}$ F(S,KC=k,K1,K2) M(K1,T1=t1) M(S, KC=k,K2)

f(T3)

Collapsed Particle with VE



To draw $\mathbf{X_k}$,
Given all other variables in **Xp**
sum out all other variables in **Xd**

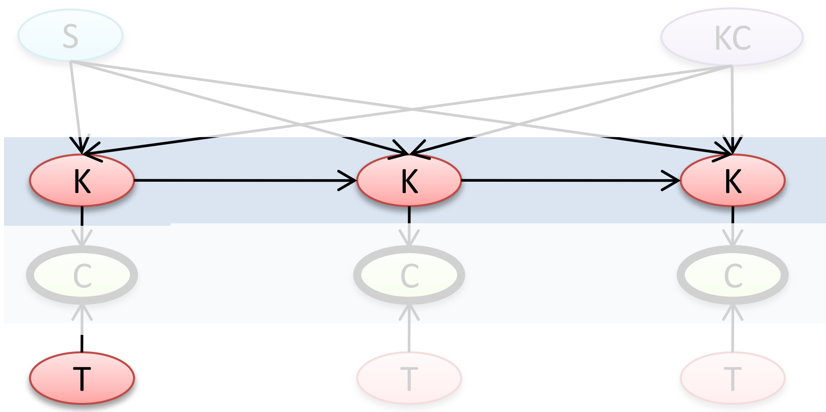


Draw KC (given S=s & T=t) from:

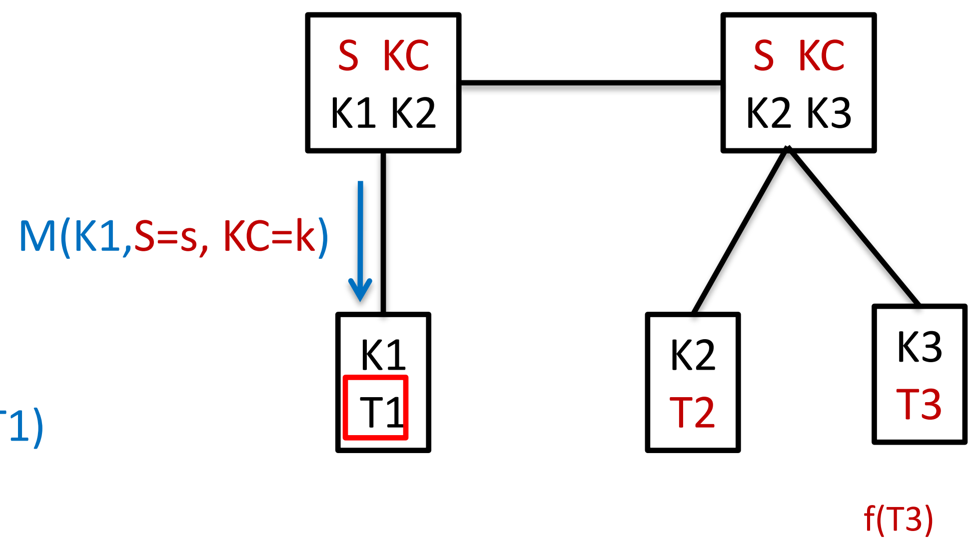
$$M(KC)=$$

$$\sum_{K1,K2} F(S=s,KC,K1,K2)\ M(K1,T1=t1)\ M(S=s,KC,K2)$$

Collapsed Particle with VE



To draw X_k ,
Given all other variables in **Xp**
sum out all other variables in **Xd**



Draw T1 (given S=s & KC=k) from:

$$M(T1) = \sum_{K1} M(K1,S=s,KC=k)\ F(K1,T1)$$

Collect Samples

Xp	**Xd**
(S, KC, T1, T2, T3)	(K1, K2, K3)
(Intel, Quick, Hard, Easy, Hard)	({1/3,1/3,1/3} , {1/4,1/4,1/2}, {1/2,1/2,0})
(Intel, Slow, Easy, Easy, Hard)	({1/2,1/2,1/4} , {1/5,4/5,0}, {1/4,1/4,1/2})
…..	…..
(Dull, Slow, Easy, Easy, Hard)	({1/3,1/3,1/3} , {1/4,1/4,1/2}, {1/2,1/2,0})
Average	Average

$$\hat{E}_{P(X)}[f(X)] = \frac{1}{N}\sum_{n=1}^{N} \left(\sum_{Xd} P(X_d \mid X_p^{(n)}) f(X_d, X_p^{(n)}) \right)$$