Jin Gu
Department of Automation, Tsinghua University
Email: jgu@tsinghua.edu.cn
Phone: (010) 62794294-866

Chapter 3
Local Probabilistic Models

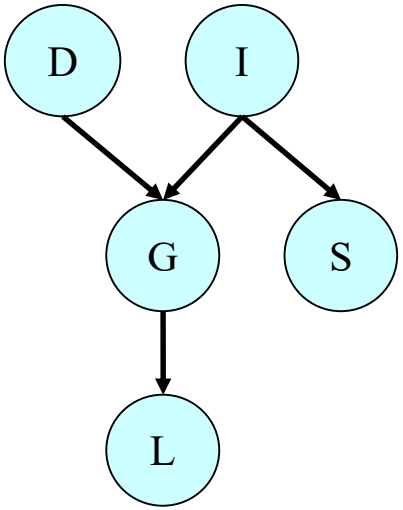
2021 Fall

Jin Gu (古槿)

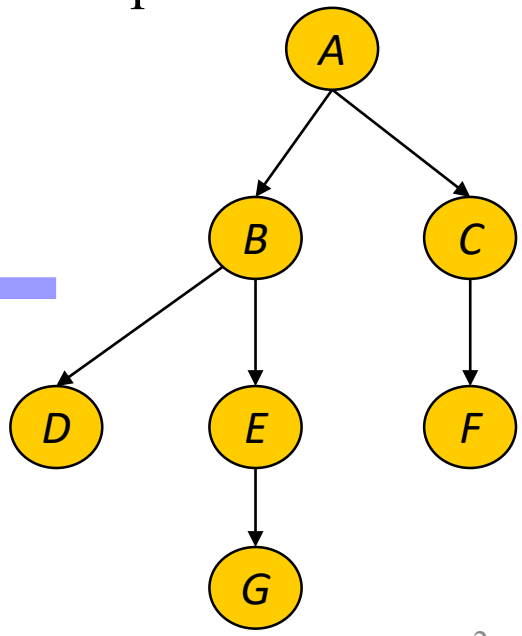
Bayesian Networks (Intuitive)

Can we find a simple graph model to equally or partially represent the probability with the same independences?

- Directed acyclic graph (DAG) G
  - Nodes $X_1, \ldots, X_n$ represent random variables
- G encodes local independence assumptions
  - X_i is independent of its non-descendants given its parents
  - Formally: $(X_i \perp NonDesc(X_i) \mid Pa(X_i))$



$$E \perp \{A, C, D, F\} \mid B$$



Factorization Theorem ***

If we define the independences in G as $X_i \perp NonDesc(X_i) | Pa(X_i)$

- G is an I-Map of $P \rightarrow P(X_1,...,X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i))$

G is a given graph. If G is an I-Map of P, P can be factorized according to G.

- $P(X_1,...,X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i)) \rightarrow G$ is an I-Map of P

G is a given graph. If P can be factorized according to G, G is an I-Map of P.

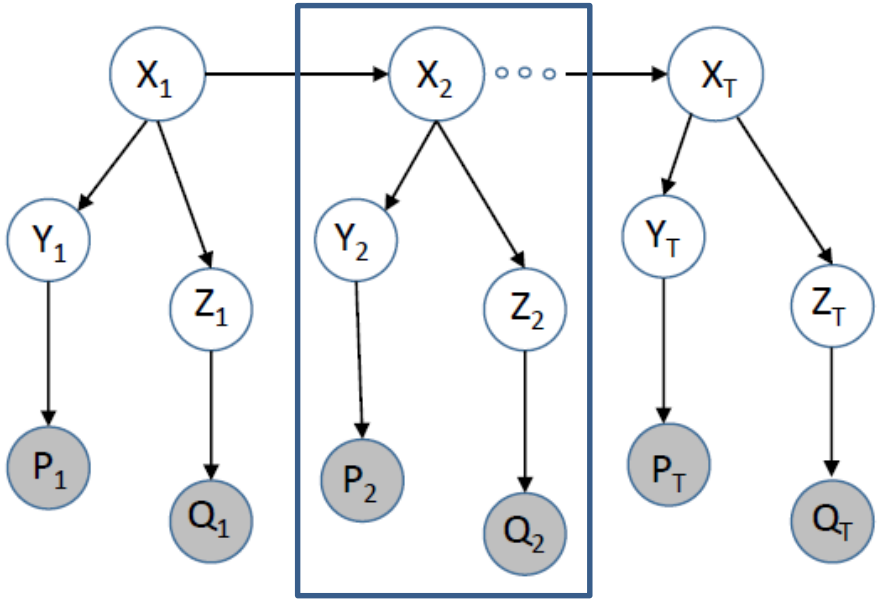
Recall Assignment #2 (5)

Here is a HCI system which can recognize human's instructions based on his gesture (by a camera) and voice (by a microphone). The raw signals detected by the two sensors should be treated as the observations of the real gestures and voices. The computer samples the sensors 10 times per second. According to these descriptions:

(1) Please draw a Bayesian network to model the human instruction recognition process at a given time.

(2) Please extend your model to consider the continuous sampling process across a period of time.

(You can use the symbols you like to represent the essential variables you think, but please explain the meaning each variable symbol stands for.)

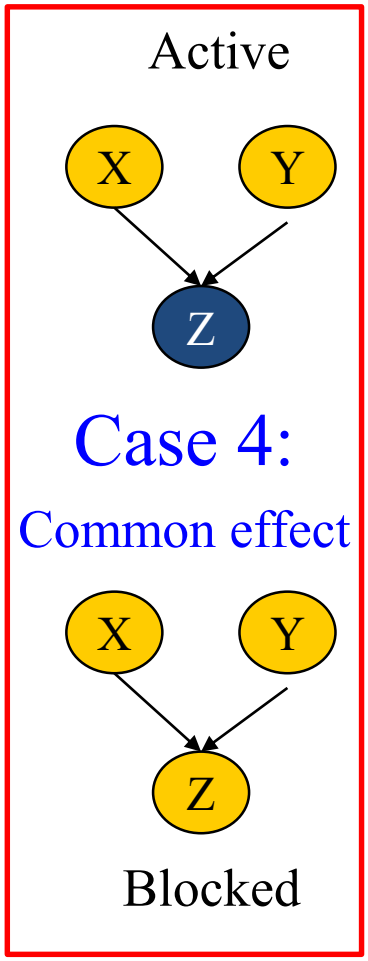


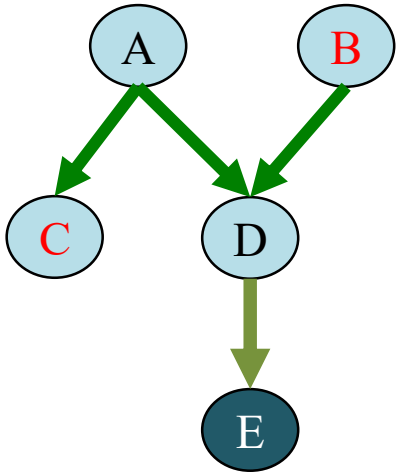
Multi-stream HMMs

Comments

- The Factorization Theorem Tells Us How to Deal With the Probability Based on Graph
  - Always put parent variables as the conditions
  - Represent you model according to logic order (generative model), ***but not always***
- Any operation on graph is equal to the reasoning in original probabilistic problems
  - For example, d-Sep can find independences
  - Removal of edges means additional independence assumptions
- Why is the model called as "Bayesian" Network?
  - In real applications, we usually need to infer "reasons" based on the observed "results"
  - Another name is ***belief*** network

V-Structure Activation in BN





d-Sep(B,C|E) = no

Child or *its descendants* will activate v-structures in BNs (cause dependences)

Comments: trees have no v-structure!

"Independences" are **NOT** Enough

- As in the student example $I \rightarrow S$
- But what is $P(I)$, and what is $P(S|I)$?

P(I,S)

I	*S*	*P(I,S)*
i^0	s^0	0.665
i^0	s^1	0.035
i^1	s^0	0.06
i^1	s^1	0.24

Joint parameterization

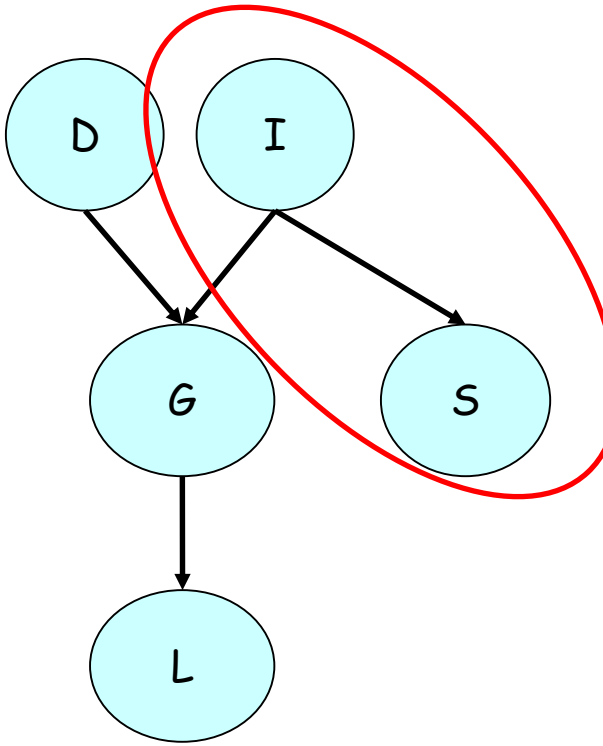
P(I)

I	
i^0	i^1
0.7	0.3

P(S|I)

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

Conditional parameterization



Local Probabilistic Models

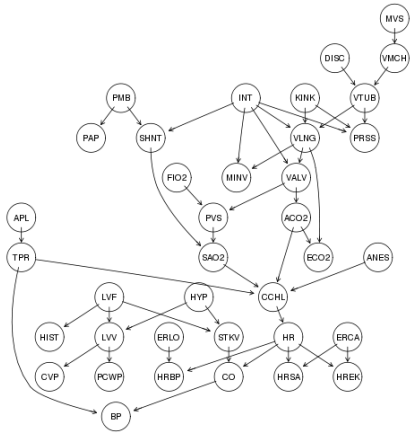
Conditional Probability Distributions (CPDs)

The Basic Structures in BNs

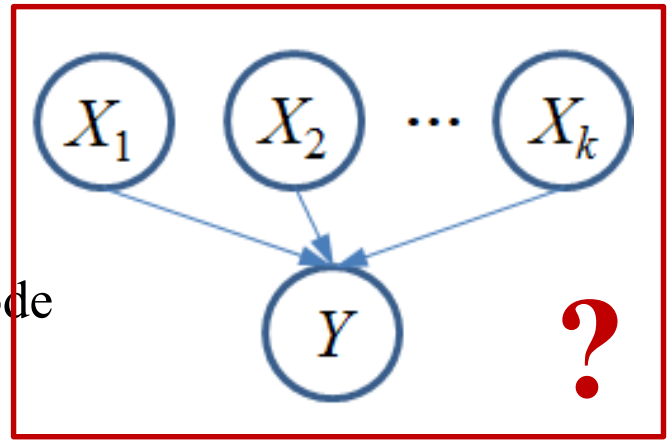
- Recall the probability factorization over graph

$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P(X_i \mid Pa(X_i))$$

- All the required representations are the local conditional probability distributions (CPDs) encoded by the target variable given its parents

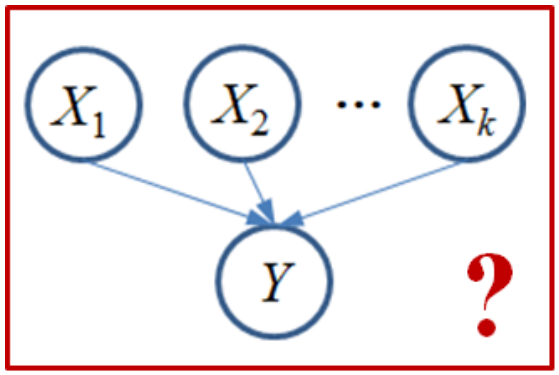


Enumerate each node with its parents



Outlines

- Local dependences in Bayesian networks
- Discrete CPDs
  - Tabular CPDs
  - Rule CPDs
  - Independence of causal influence
- Continuous CPDs
- A few CPDs in neural networks
- Bayesian networks representation examples

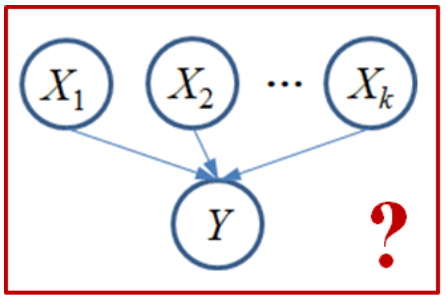


Textbook References

- **Textbook 1**
  - Chapter 5.1~5.4
  - Chapter 5.6

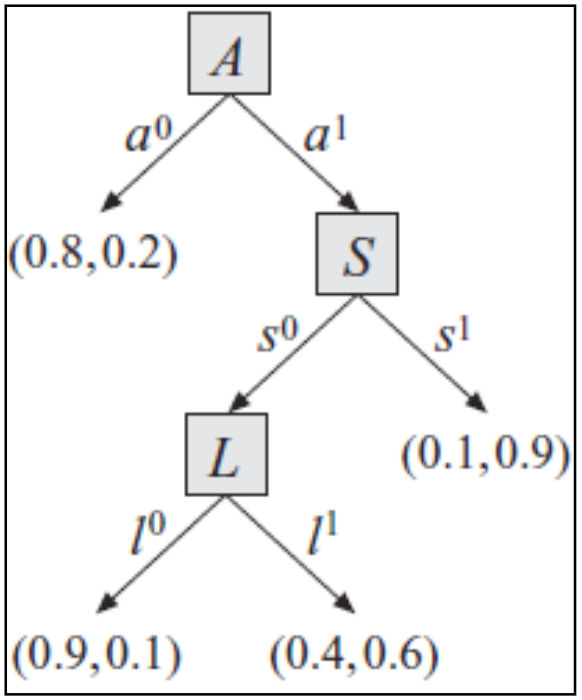
- **Textbook 2**
  - Chapter 11 (Mixture Models)
  - Chapter 12 (Latent Factor Models)

Tabular CPDs



- For a local structure: $(Y|X_1, X_2, \cdots, X_k)$, if all variables are binary, there should be 2^k independent parameters.
  - 2^k different configurations for $X_1, X_2, \cdots, X_k$
  - For each configuration for $X_1, X_2, \cdots, X_k = \vec{x}$, we need $P(Y = y_0|\vec{x})$
- The required parameters are exponentially increasing as the number of parents in BN local structures

Rule CPDs



- We have a set of rules from the distribution
- For $X = x$ and a configuration of a subset of its parents e, with a probability p, we can define a **RULE**:

$$\rho : \langle e, x; p \rangle$$

- Tree CPDs can be directly transformed into a rule CPD

$\rho_1 : \langle a^0, j^0; 0.8 \rangle$
$\rho_2 : \langle a^0, j^1; 0.2 \rangle$
$\rho_3 : \langle a^1, s^0, l^0, j^0; 0.9 \rangle$
$\rho_4 : \langle a^1, s^0, l^0, j^1; 0.1 \rangle$
$\rho_5 : \langle a^1, s^0, l^1, j^0; 0.4 \rangle$
$\rho_6 : \langle a^1, s^0, l^1, j^1; 0.6 \rangle$
$\rho_7 : \langle a^1, s^1, j^0; 0.1 \rangle$
$\rho_8 : \langle a^1, s^1, j^1; 0.9 \rangle$

A rule-based CPD $P(X \mid \mathrm{Pa}_X)$ *is a set of rules* $\mathcal{R}$ *such that:*

- *For each rule* $\rho \in \mathcal{R}$*, we have that* $Scope[\rho] \subseteq \{X\} \cup \mathrm{Pa}_X$.
- *For each assignment* $(x, \boldsymbol{u})$ *to* $\{X\} \cup \mathrm{Pa}_X$*, we have precisely one rule* $\langle \boldsymbol{c}; p \rangle \in \mathcal{R}$ *such that* $\boldsymbol{c}$ *is compatible with* $(x, \boldsymbol{u})$*. In this case, we say that* $P(X = x \mid \mathrm{Pa}_X = \boldsymbol{u}) = p$.
- *The resulting CPD* $P(X \mid \boldsymbol{U})$ *is a legal CPD, in that*

$$\sum_x P(x \mid \boldsymbol{u}) = 1.$$

■

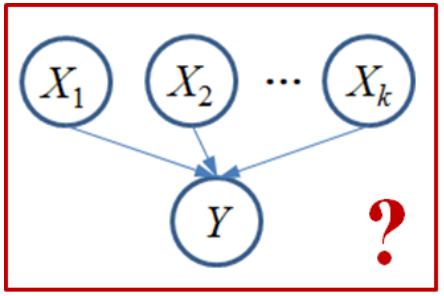
- Rule CPDs can model any complex CPDs, but not any rule CPD can be compactly transformed as a tree CPD. For example, no rule includes all the parent nodes.

COMMENTS: If you have big memory, use **hash table** to store rule-CPDs. You will get computational complexity O(1) for finding a target probability.

Independence of Causal Influence

- "Causal" / "Associative"

- The computer system will run down **if any one** of CPU, memory, hard disk, power or OS encounters some problems.
- The bank data center will run down **if all** the big computers run down at the same time.
- You will success **if the addictive** efforts of your body health, diligence, intelligence and persistence are enough.



Noisy-Or Model

- The failure rate of CPU is f_1
- The failure rate of MEM is f_2
- The failure rate of DISK is f_3
- The failure rate of POWER is f_4
- The failure rate of OS is f_5
- The failure rate of other events is f_0

- Question: the failure rate of your computer?

Noisy-Or Model

- Define $f_1 \sim f_k$ as noisy parameters (in normal condition, there should be no hardware / software failure, so we treat the failure as a noise)
- Define f_0 as leak probability

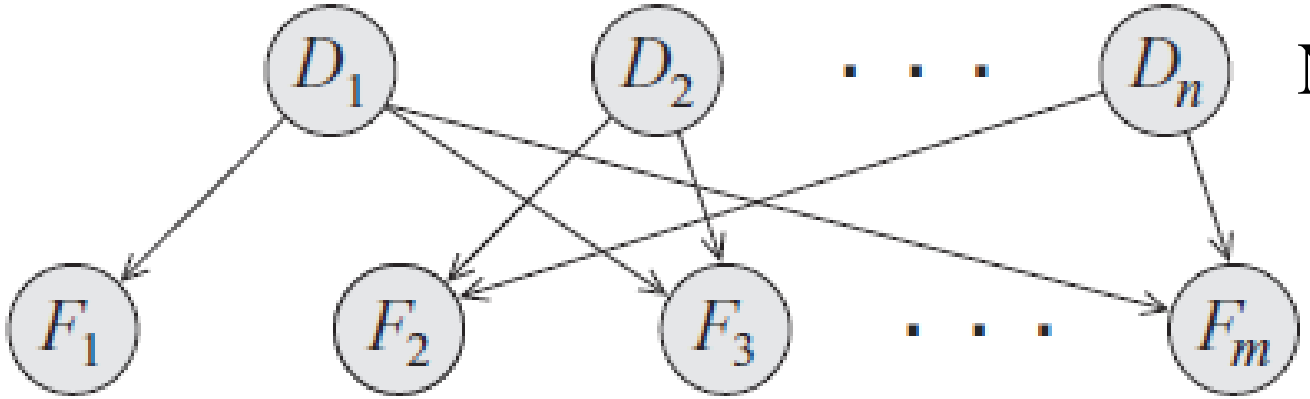
$$P(F = False) = (1 - f_0)\prod_i (1 - f_i)$$

$$P(F = True) = 1 - (1 - f_0)\prod_i (1 - f_i)$$

- **Noisy-or**: any parent can cause the same effect of its child with some probability

BN2O Networks

- Extension from Noisy-or model & Naive Bayes
- A disease can cause multiple phenotypes or symptoms, and multiple diseases may have share symptom(s).
- We can only observe symptoms, but we want to know diseases (most drugs are used for treating diseases).



Need to Infer

Observed

You need to specify the symptoms for each disease from big clinical data or medical instructions.

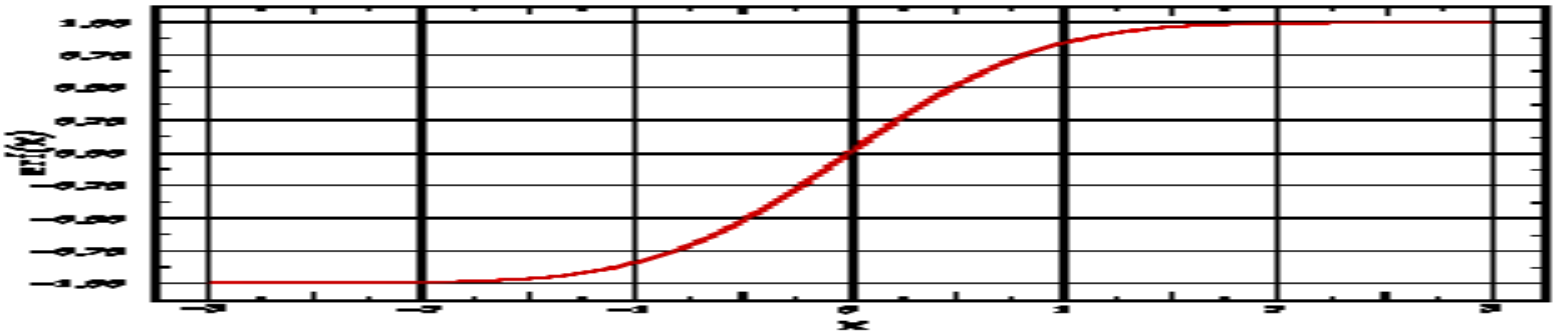
The Generalized Linear Models

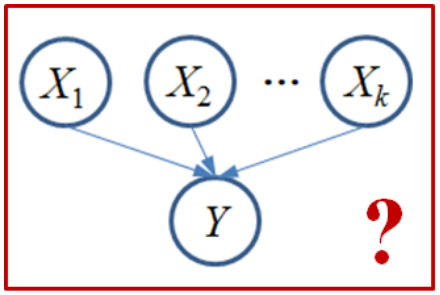
- You will be more likely to be successful **if the summarized** score of the body health, diligence, intelligence, and persistence is higher.

$$s = w_h s_h + w_d s_d + w_i s_i + w_p s_p$$

$$P(success) = P(s) = ?$$

Sigmoid Function!!

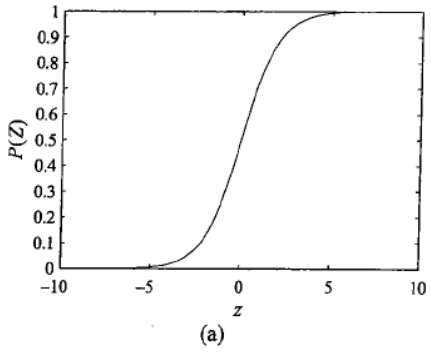


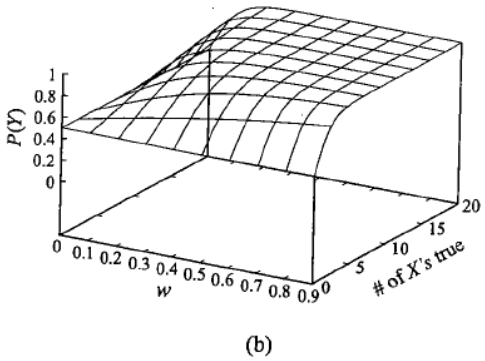


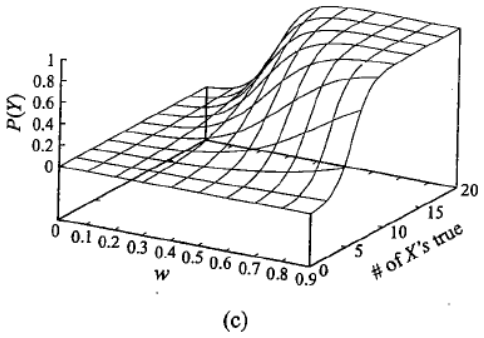
Logistic CPDs

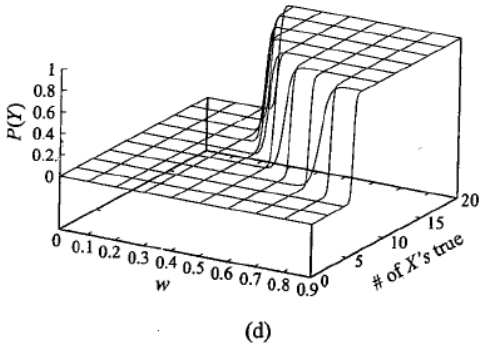
- Sigmoid function

$$\text{sigmoid}(s) = \frac{e^s}{1+e^s}$$



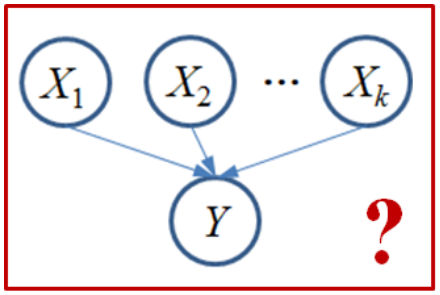






- **Logistic CPD**

$$P\left(Y = y^1 \mid X_1, \cdots\right) \qquad \text{sigmoid}\left(w_0 + \sum_{i=1}^{k} w_i X_i\right)$$



Logistic CPDs

- Odds

$$O = \frac{P\left(Y = y^1 \mid X_1, \cdots\right)}{P\left(Y = y^0 \mid X_1, \cdots\right)} \qquad e^Z$$

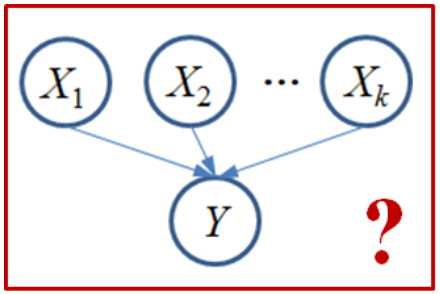
- Larger odds (>1) mean the configuration is more likely to cause the positive result

- The change of log odds of binary variables
  - X are binary

$$\Delta O = \frac{O\left(X_{-j}, X_j = x_j^1\right)}{O\left(X_{-j}, X_j = x_j^0\right)} = e^{w_j}$$

- $w_j > 0$ means positive contribution

Generalized Linear Models

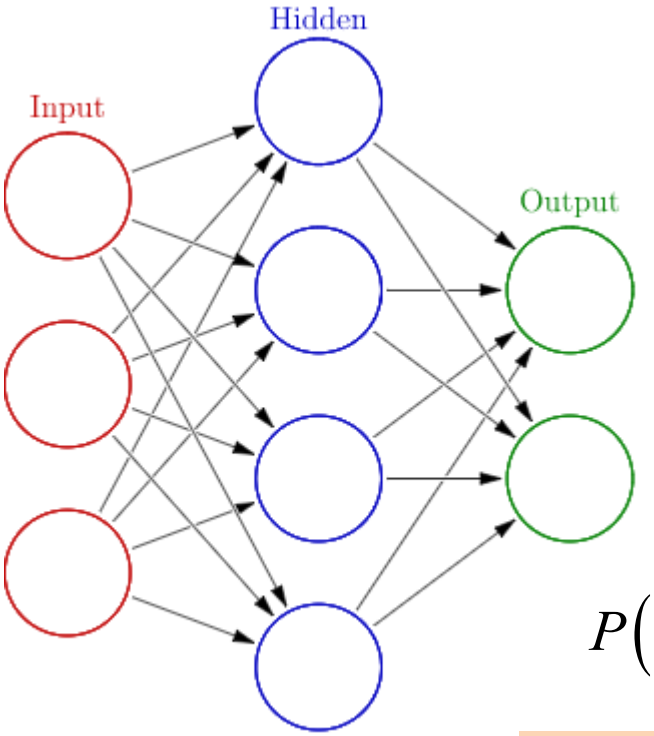


- Linear Gaussian
  - $y = w_0 + \sum_{i=1}^{k} w_i x_i + \varepsilon, \varepsilon \in \mathcal{N}(0, \sigma^2)$
    - $\mathrm{p}(y) \sim \mathcal{N}\left(w_0 + \sum_{i=1}^{k} w_i x_i, \sigma^2\right)$

- Binomial distribution (Logistic CPD)
  - $\mathrm{P}(y = 1|x) = x, x = \mathrm{sigmoid}\left(w_0 + \sum_{i=1}^{k} w_i x_i\right)$
- Poisson distribution
  - $\mathrm{P}(y = k|X) = \lambda^k e^{-\lambda}/k!, \lambda = w_0 + \sum_{i=1}^{k} w_i x_i$

Activation Function in NNs

- Three-layer NNs



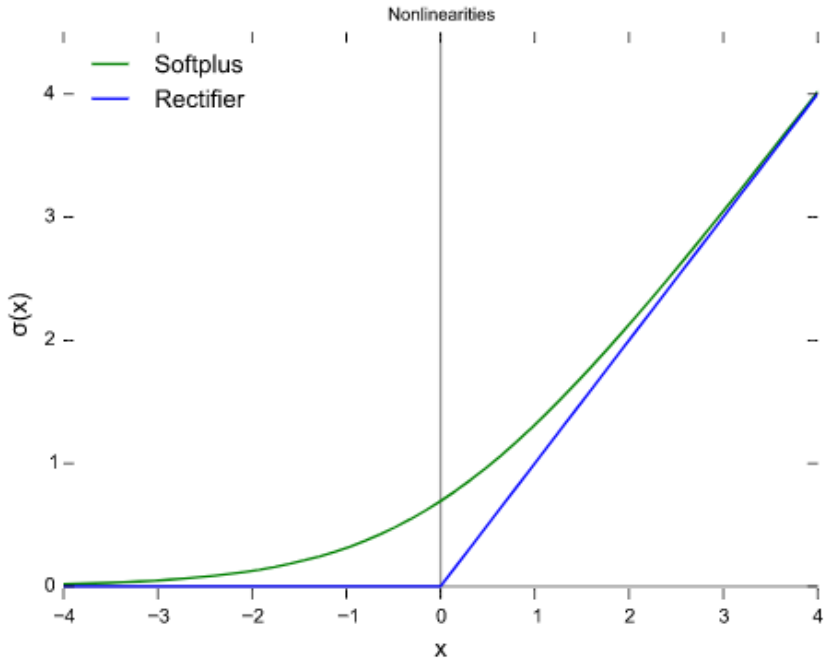
- Output of previous neurons
  - Activated (x_i=1)
  - Non-activated (x_i=0)
- Input of a neuron
  - $s_j = w_0 + \sum_i w_{i,j} x_i$
- Activation function
  - Sigmoid function

$$P\left(Y = y^1 \mid X_1, \cdots \right. \qquad \text{sigmoid}\left(w_0 + \sum_{i=1}^{k} w_i X_i \right)$$

Drawback: the gradient is too small during the two extremes of the sigmoid function!

ReLU: Rectified Linear Units

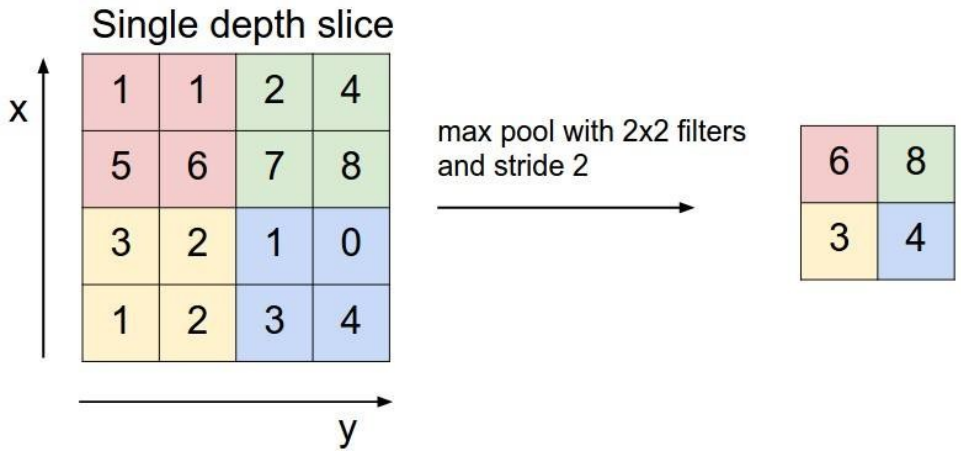
- Rectifier & softplus
  - Rectifier: $y = f(s) = \max(0, s)$
  - Softplus: $y = f(s) = \ln(1 + \exp(s))$



Comments: ReLU is the most used function for avoiding gradient dropping during backward propagation

Pooling Function

- Max pooling / median pooling (filtering)
- Advantages
  - Reduce noises for *median* pooling
  - Shift invariant for *max* pooling
  - *Q: How about rotation & inflation invariant?*

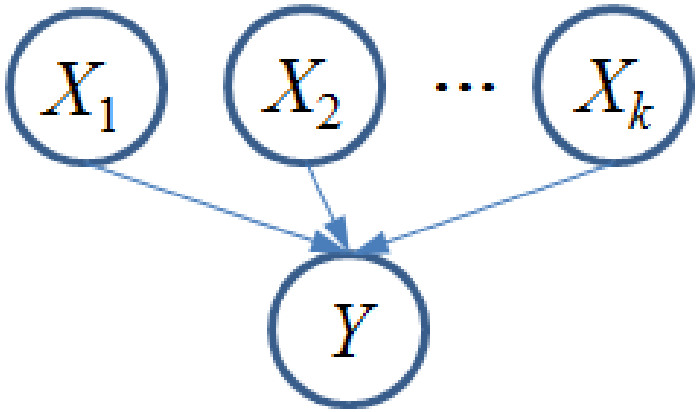


The General Formulation

- Noisy-or model
- Generalized linear model

- Noisy-max model
- Noisy-and model
- …
- …

- We can significantly reduce the number of parameters for proper representation of CPDs



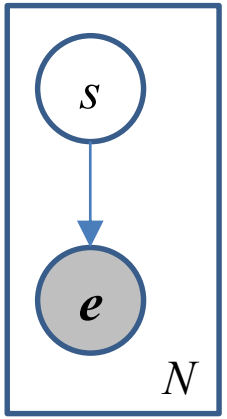
Summary: From I-Map to CPDs

- "Independences" are not enough to represent a probability.
- Bayesian Networks separate a large joint distribution into a set of local probabilistic models or conditional probability distributions. We further need above models to represent the local distributions efficiently.
- Local models are designed for $P(X|Pa_X)$

BN Representation Example #1

- Cancer is a general disease category consisting of many different types of diseases. For example, breast cancer has four major subtypes: normal-like, basal, luminal A and luminal B, with distinct clinical outcomes. Each subtype has different gene expression patterns. We need to infer the subtypes based on the observed gene expressions.

- ***Variables***: subtypes (*s*), gene expressions (***e***)
- ***Relation***: subtypes define the patterns of expressions

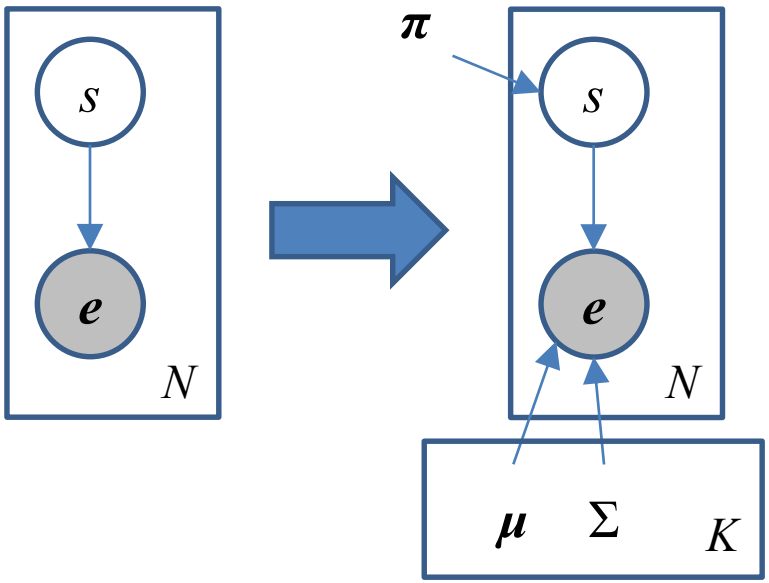


1) ***Circles*** represent variables. ***Shadow/white*** circles represent observed/unobserved variables.

2) ***Rectangle*** indicates the variables in the same logic layer. The ***number*** (*N*) at the right-bottom corner represents the number of samples.

BN Representation Example #1

- ***Variables***: subtypes (s), gene expressions ($\boldsymbol{e}$)
- ***Relation***: subtypes define the patterns of expressions
- Important: carefully check the independences in your model



- Consider the local CPDs
- Different subtypes have different occurrence probability → (s)

$$P(s=k)=\pi_i, \quad \sum_{k=1}^{K}\pi_i=1$$

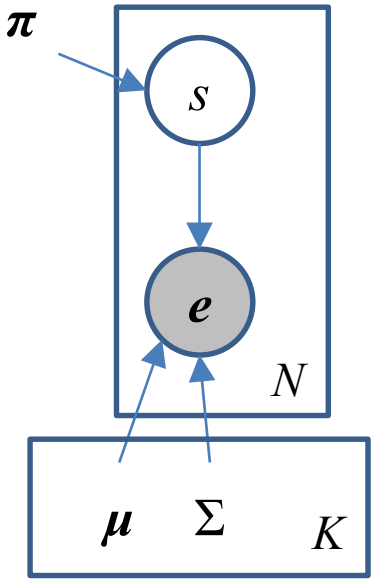
- Each expression pattern (indicated by i) independently follows different Gaussian distributions given the cancer subtype → ($\boldsymbol{e}|s$)

$$p(\boldsymbol{e}\,|\,s=k)\sim \mathrm{N}(\boldsymbol{\mu}_k,\Sigma_k)$$

Classical *Gaussian Mixture Model!*

BN Representation Example #1

$$P(s=k)=\pi_i, \quad \sum_{k=1}^{K}\pi_i=1$$



$$p(\boldsymbol{e}\mid s=k)=\mathrm{N}(\boldsymbol{\mu}_k,\Sigma_k)$$

- How about the complete probabilistic model?

$$p(s=k,\boldsymbol{e})=P(s=k)\,p(\boldsymbol{e}\mid s=k)$$

$$=\pi_k\times\frac{\exp\left(-\frac{1}{2}(\boldsymbol{e}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\boldsymbol{e}-\boldsymbol{\mu}_k)\right)}{\sqrt{(2\pi)^k|\Sigma_k|}}$$

Multiply all the CPDs in your model!!

$$p(S,E)=\prod_{n=1}^{N}p(s[n]=k,\boldsymbol{e}[n])$$

For learning, you have N i.i.d. samples

BN Representation Example #2

- Cancer is usually caused by *independent* driving processes (factors), such as sustainable proliferation, resistance to cell death, immune escape and promoted vascular growth, etc.
- These driving processes have combined effects on the gene expressions patterns.
- We want to infer the driving processes based on large-scale gene expression datasets.

- ***Variables***: driving factors (z_i), gene expressions (e_j)
- ***Relations***: ***z*** determines the distribution of ***e***
- ***Local CPDs***
  - Driving process z_i follows a standard Gaussian distribution

$$p(z_i) \sim \mathrm{N}(0,1) \quad i = 1, \cdots$$

  - Expression e_j also follows a Gaussian distribution but its mean is determined by a linear model of ***z***

$$p(e_j \mid \boldsymbol{z}) \sim \mathrm{N}\left(\alpha_0 + \sum_{i=1}^{K} \alpha_i z_i, \tau_j^2\right)$$

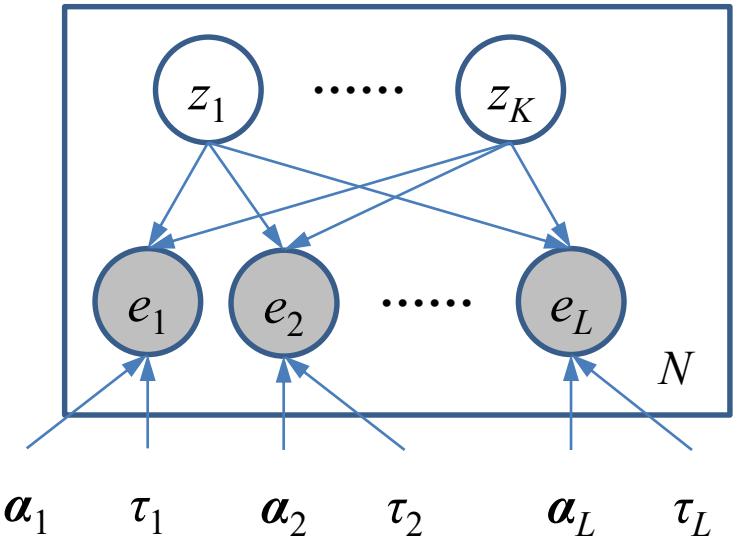
Orthogonal *Latent Facotr Model!*

- ***Variables***: independent driving processes (z_i), gene expressions (e_j)
- ***Relations***: ***z*** determines the distribution of ***e***
- ***Local CPDs***
  - Driving process z_i follows a standard Gaussian distribution

$$p(z_i) \sim \mathrm{N}(0,1) \quad i = 1, \cdots$$

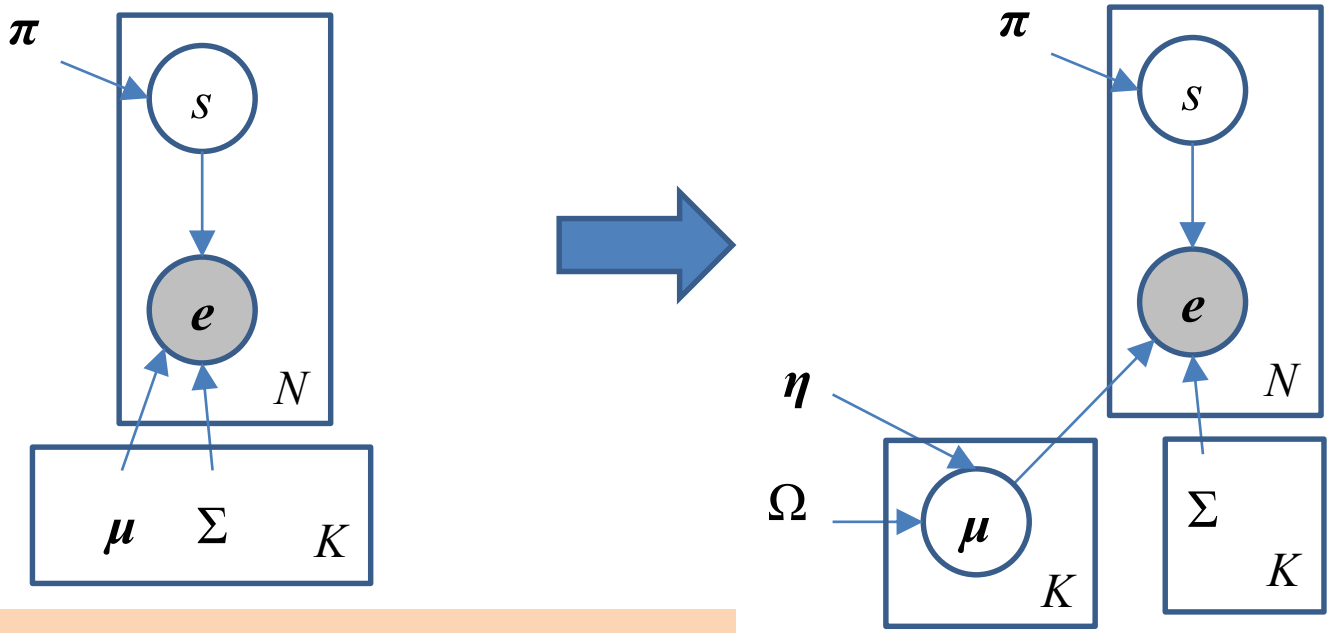
  - Expression e_j also follows a Gaussian distribution but its mean is determined by a linear model of ***z***

$$p(e_j \mid \boldsymbol{z}) \sim \mathrm{N}\left(\alpha_0 + \sum_{i=1}^{K} \alpha_i z_i, \tau_j^2\right)$$



BN Representation Example #3

- Hierarchical Bayesian models
  - Key points: some parameters are random variables and further conditional on a few hyper-parameters



Comments: hierarchical Bayesian models have strong representation capability

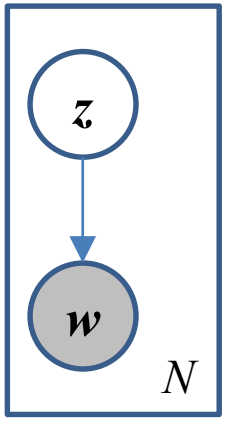
BN Representation Example #4

- Topic Models in NLP
  - We have *M* documents
  - Each document has one or more topics (such as sport, finance, politics, entertainment, etc.)
  - Each topic has different word usage (such as Qian Yang, Ming Yao, Olympics, football for sport)
- Can we draw a model for learning the topics from the documents?

Well-Known *Latent Dirichlet Allocation (LDA)*!

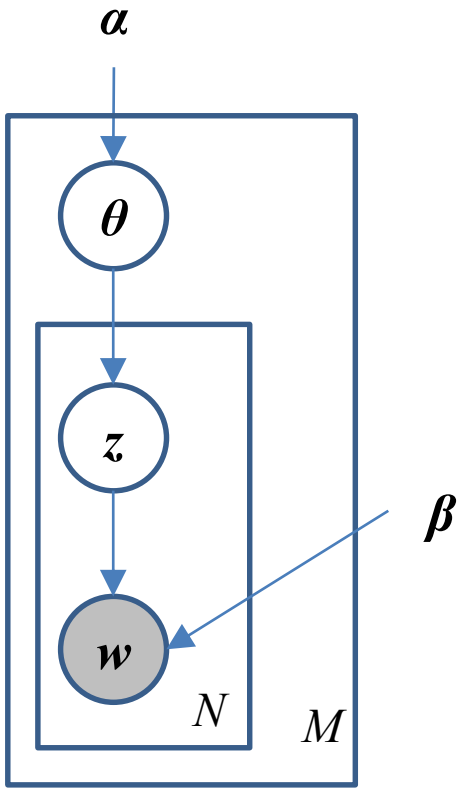
BN Representation Example #4

- Define variables for each document
  - Word (as binary vector): $\boldsymbol{w}_n$
  - Topic (as binary vector): $\boldsymbol{z}_n$
- Determine edges & local CPDs

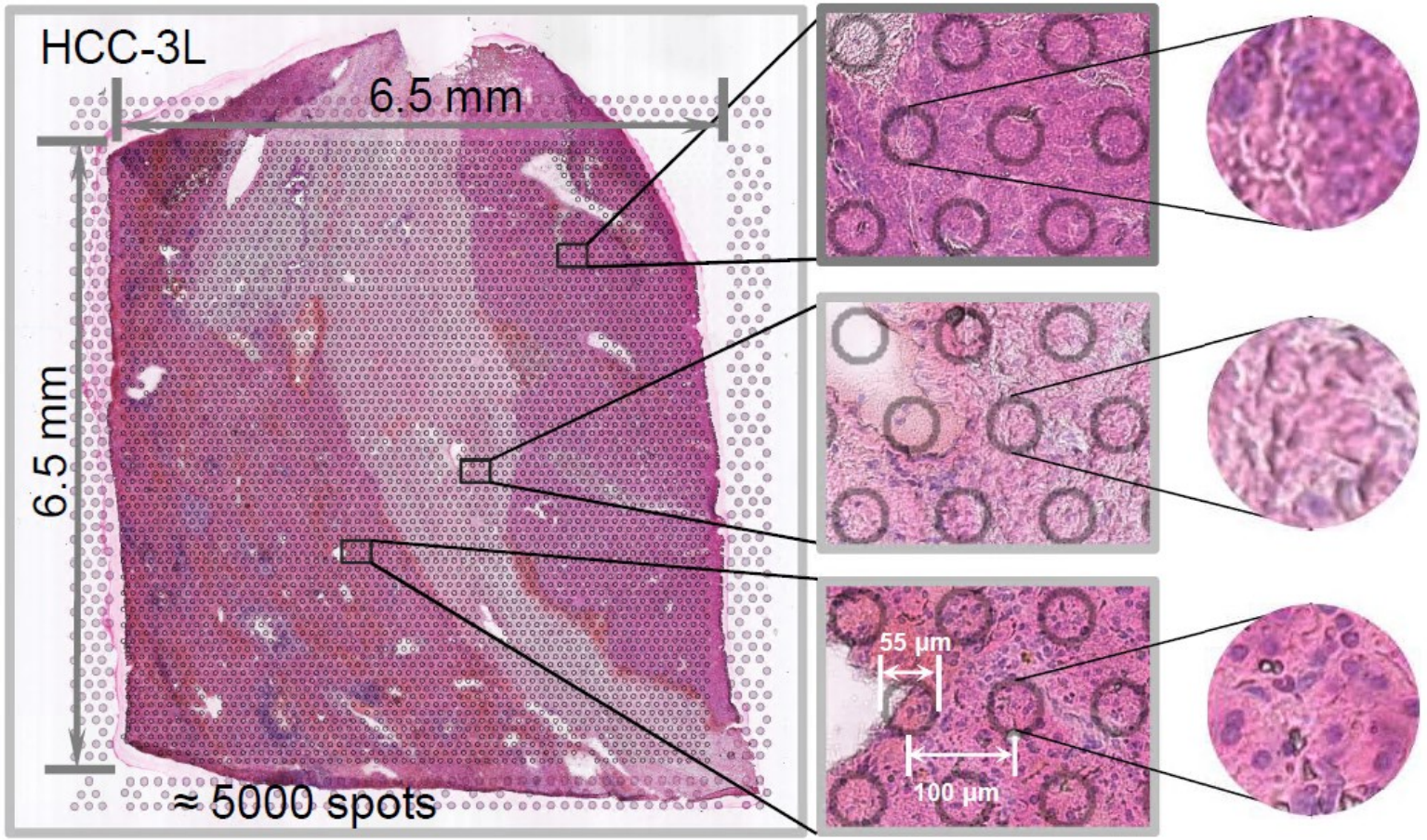


z follows a multinomial distribution
(the document generates one topic at a time)

w follows a multinomial distribution conditional on *z*
(the topic generates one word, repeating *N* times)



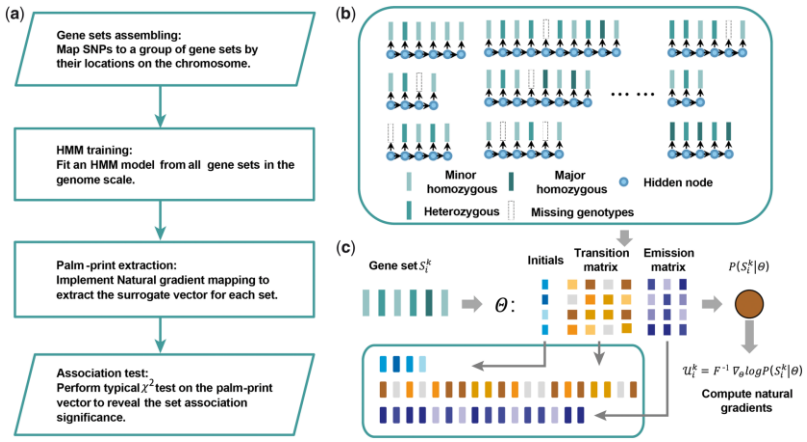
Course Project: Multi-Model Spatial Information Modeling



Course Project: 2-Page Proposal

- Project selection
  - Assigned projects: segmentation of multi-model spatial information of tumor tissues
  - Task #1, use LDA or other models to cluster the spatial spots according to the expression data
  - Task #2, use MRF or other models to segment and partition the histological image
- Proposal deadline: ***November 8*** (网络学堂)
- ***Everyone*** should submit his own proposal

Own Projects: Some Examples



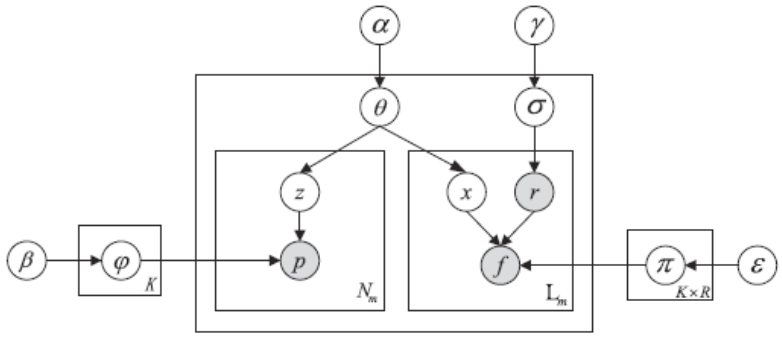


Fig. 4. Graphical representation of *socoLDA*. The Observed variables p, f, and r denote travel packages, co-travelers, and relationship types between users and co-travelers, separately. The hidden variables x and z denote topics assigned for travel packages and topics assigned for co-travelers, separately.

Hidden Markov models
Bao et al., *Briefings in Bioinformatics*, 2017

Extended Latent *Dirichlet* Allocation (LDA)
He et al., *Information & Managemen*t, 2016

Methods & Theories
Project: Training deep generative models by MCEM

Image analysis
Project: 基于MRI图像的心脏功能辅助检测