Graphical Models

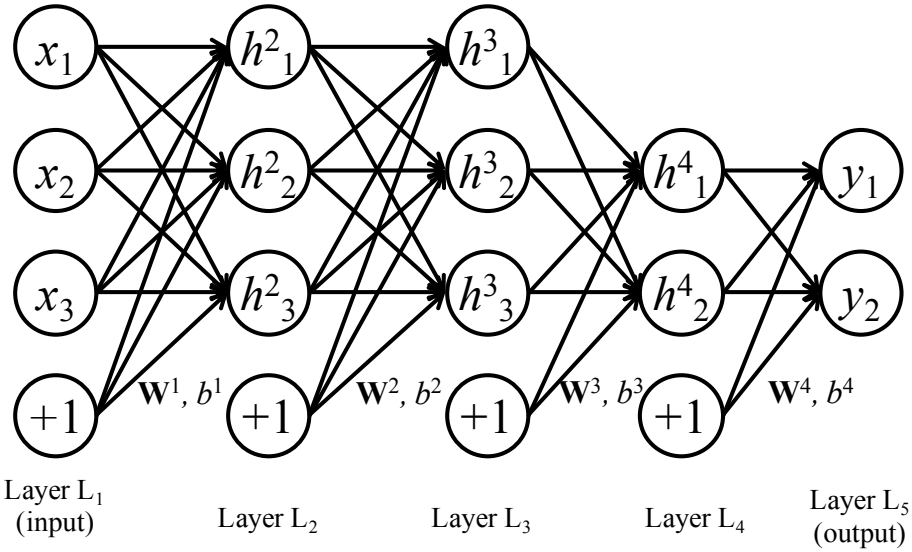
Jie Tang

Department of Computer Science and Technology
Tsinghua University

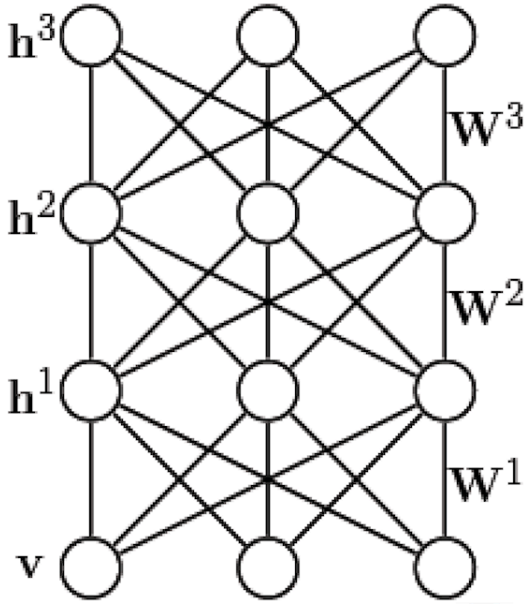
Outline

- Introduction
- Directed Graphical Models
- Undirected Graphical Models
- Factor Graph

Directed vs. Undirected

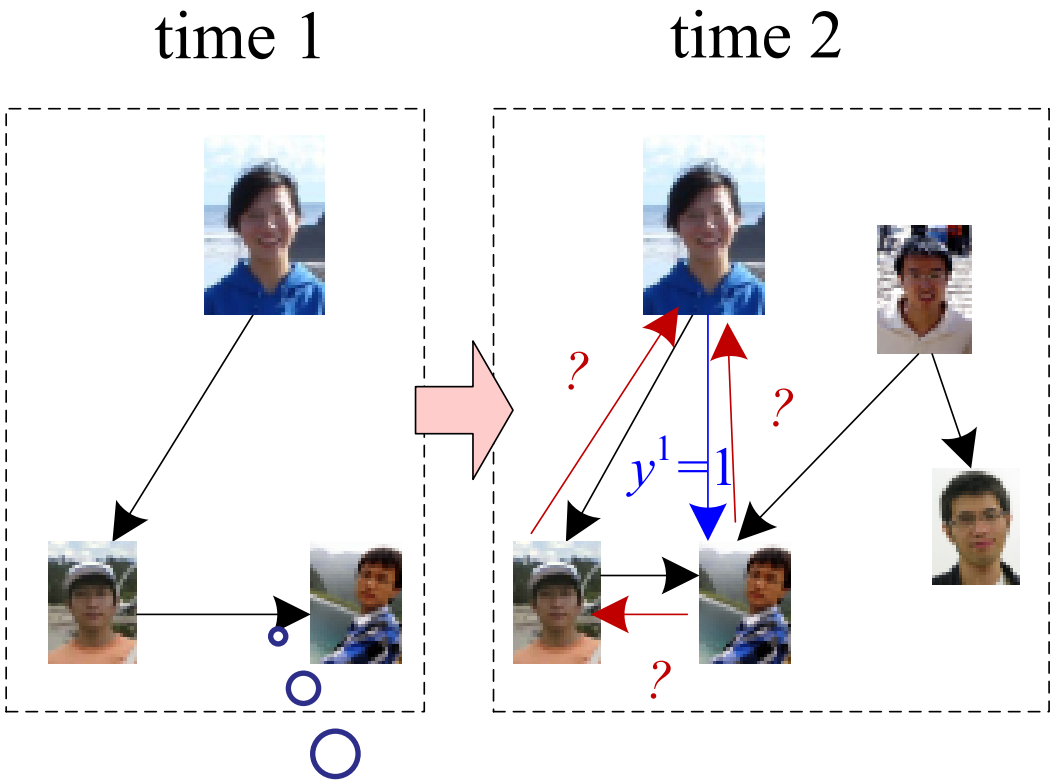


Directed



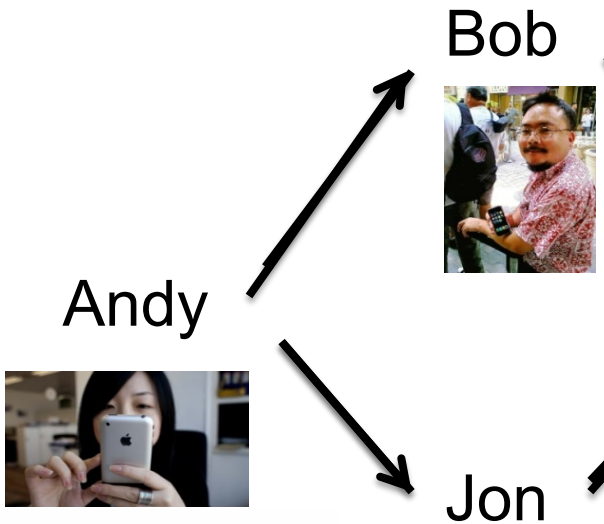
Undirected

Follow back Prediction



When you follow a friend in Twitter, how likely he will follow back?

Retweet Predicting



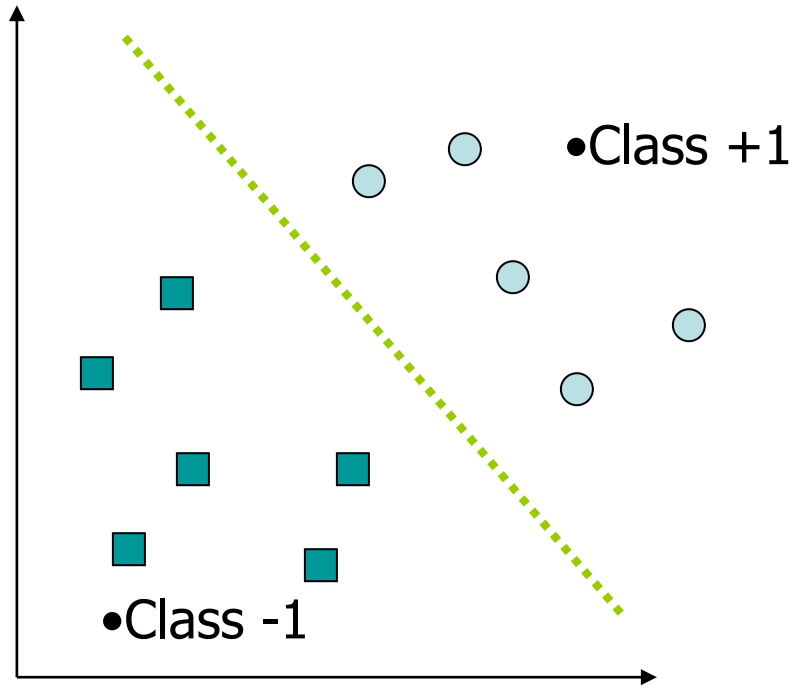


When you post a tweet…





Binary Classifier



Sequence Labeling

- Pos Tagging
  - E.g. [He/PRP] [reckons/VBZ] [the/DT] [current/JJ] [account/NN] [deficit/NN] [will/MD] [narrow/VB] [to/TO] [only/RB] [#/#] [1.8/CD] [billion/CD] [in/IN] [September/NNP] [./.]

- Term Extraction
  - Rockwell International Corp.'s Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co. to provide structural parts for Boeing's 747 jetliners.

IE from Web Page

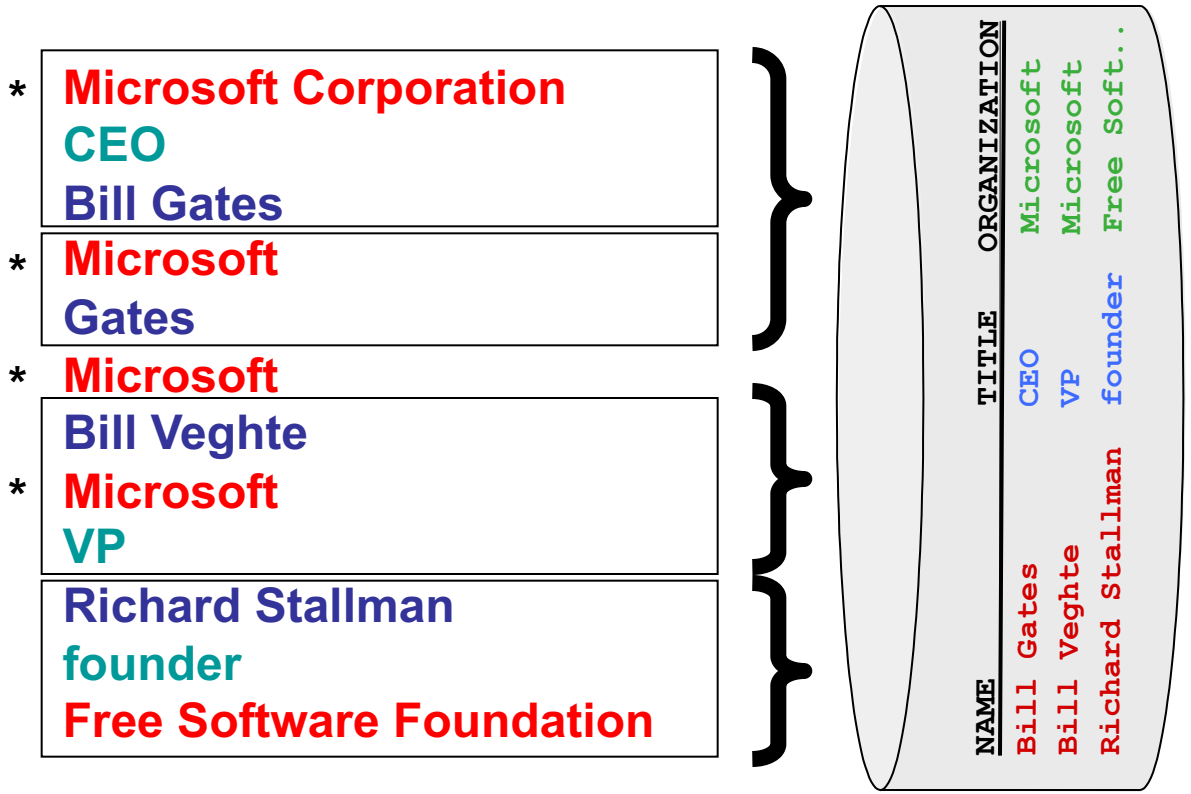
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

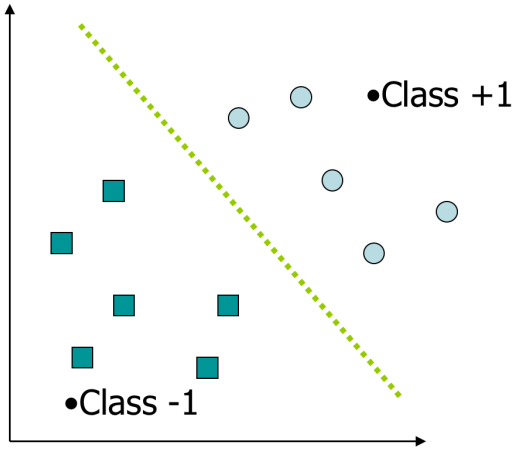
"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

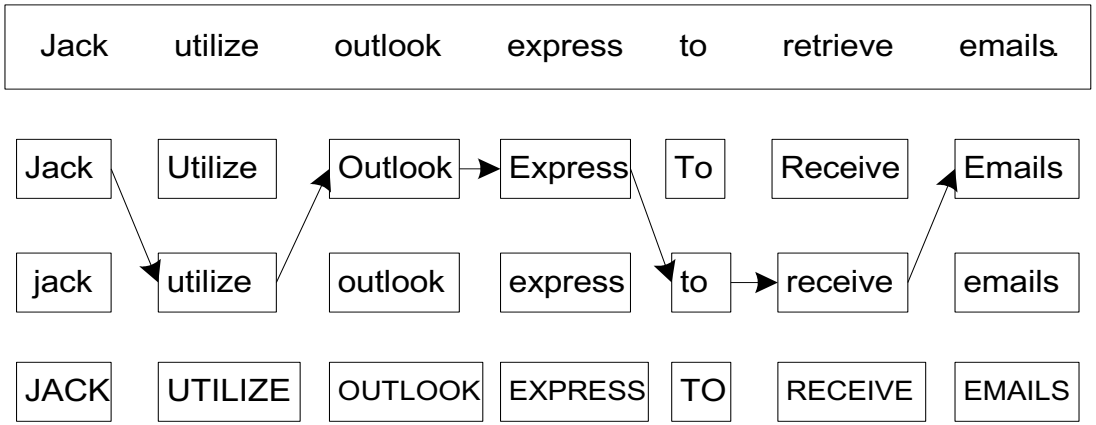
Richard Stallman, founder of the Free Software Foundation, countered saying…



Binary Classifier vs. Sequence Labeling

- Case restoration
  - "jack utilize outlook express to retrieve emails"
  - E.g. SVMs vs. CRFs





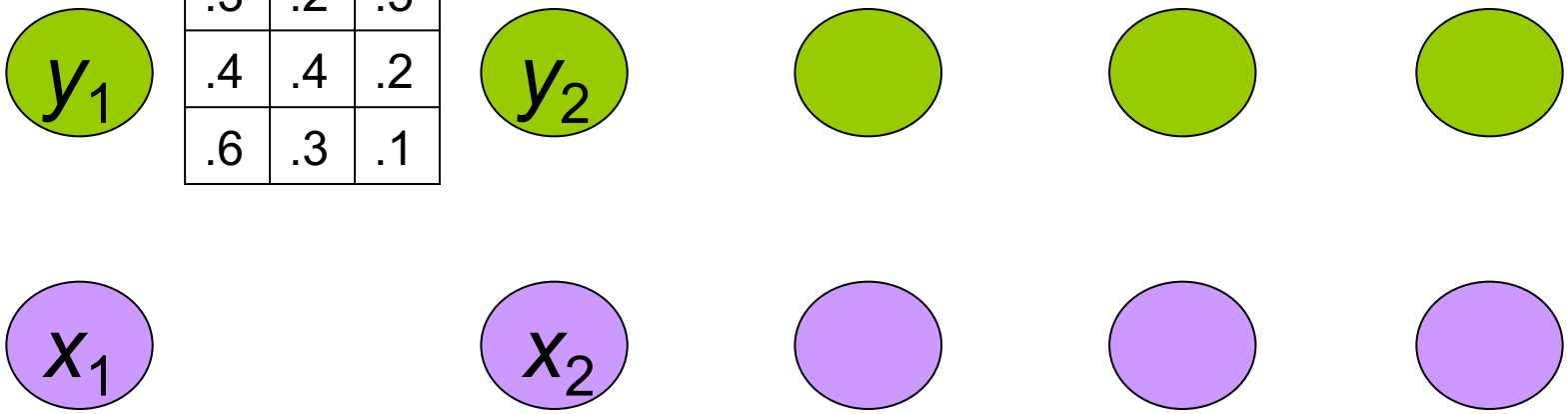
Sequence Labeling Problem

Conditional probability table

.5	.5
.4	.6
.3	.7

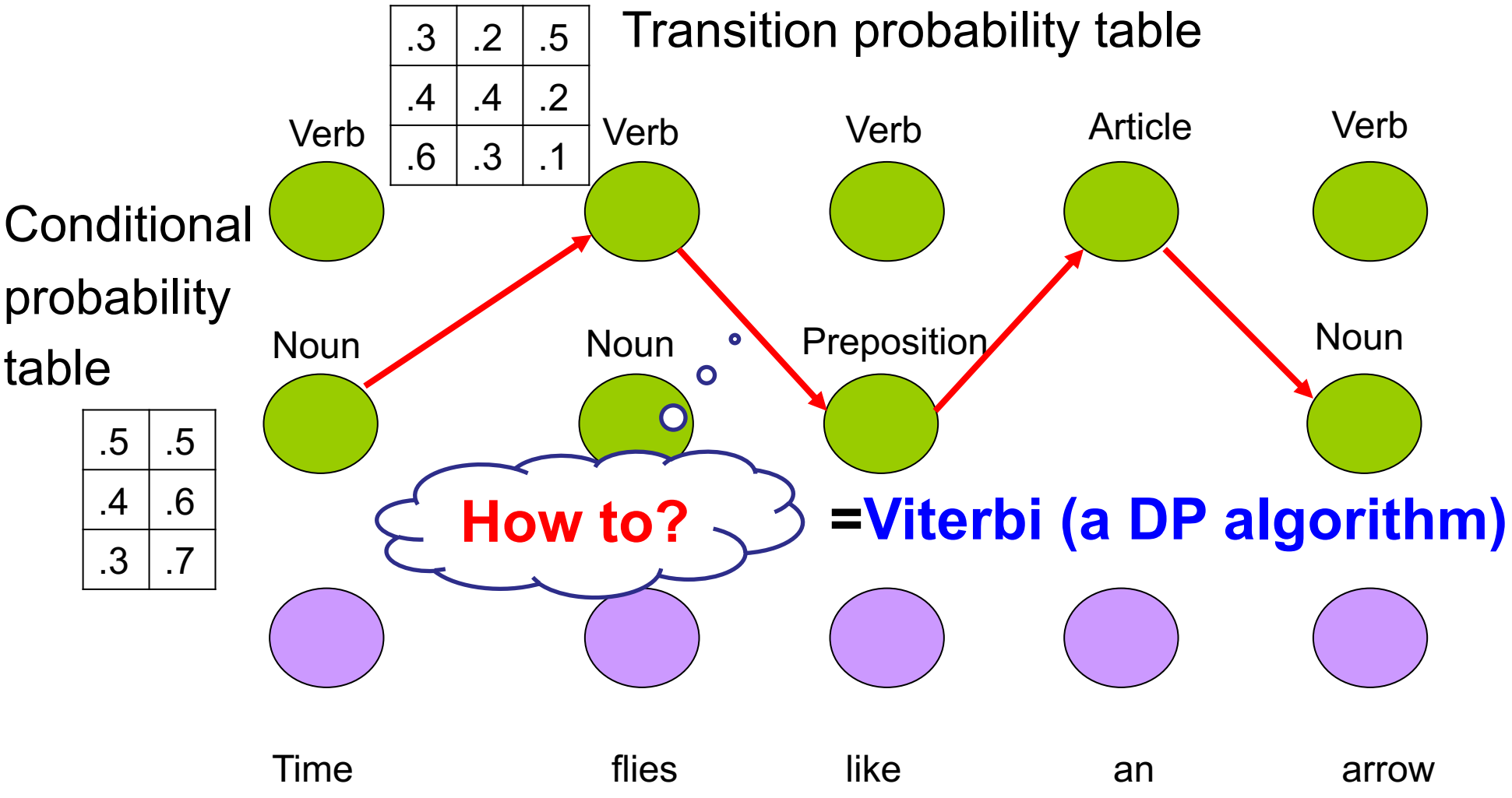
Transition probability table

.3	.2	.5
.4	.4	.2
.6	.3	.1



- Green nodes are ***states***
- Purple nodes are ***observations***

Example: POS Tagging Problem



DP Example 1—汽车加油问题

- 环形公路，有*n*个加油站，每个加油站有油量*gas*[*i*]，到达下一个加油站耗油*cost*[*i*]， 求一辆车从哪个加油站出发能够回到该出发的加油站。

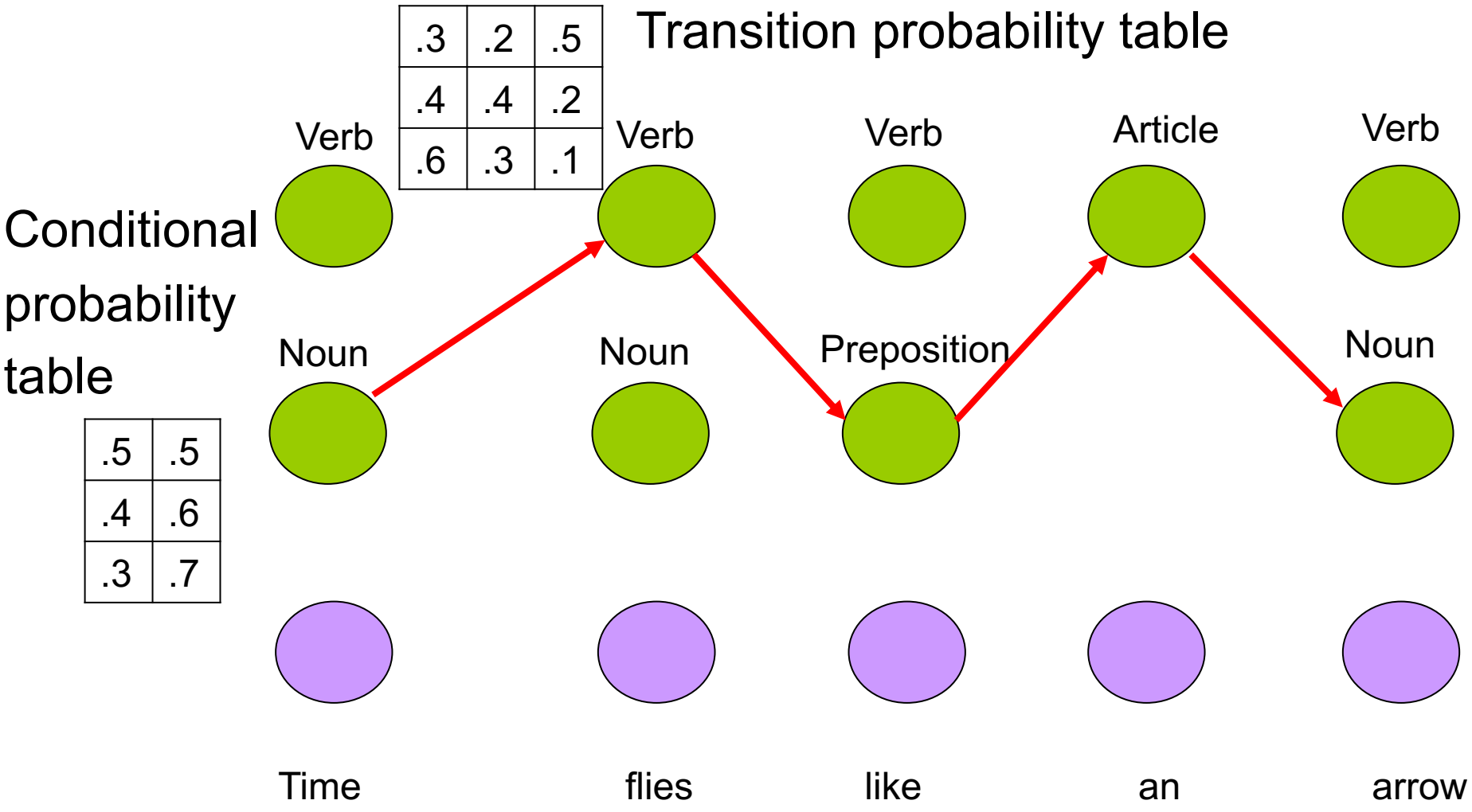
（设解唯一，即所有油量刚好走完整个环形公路）

DP Example 2—两指拨电话问题

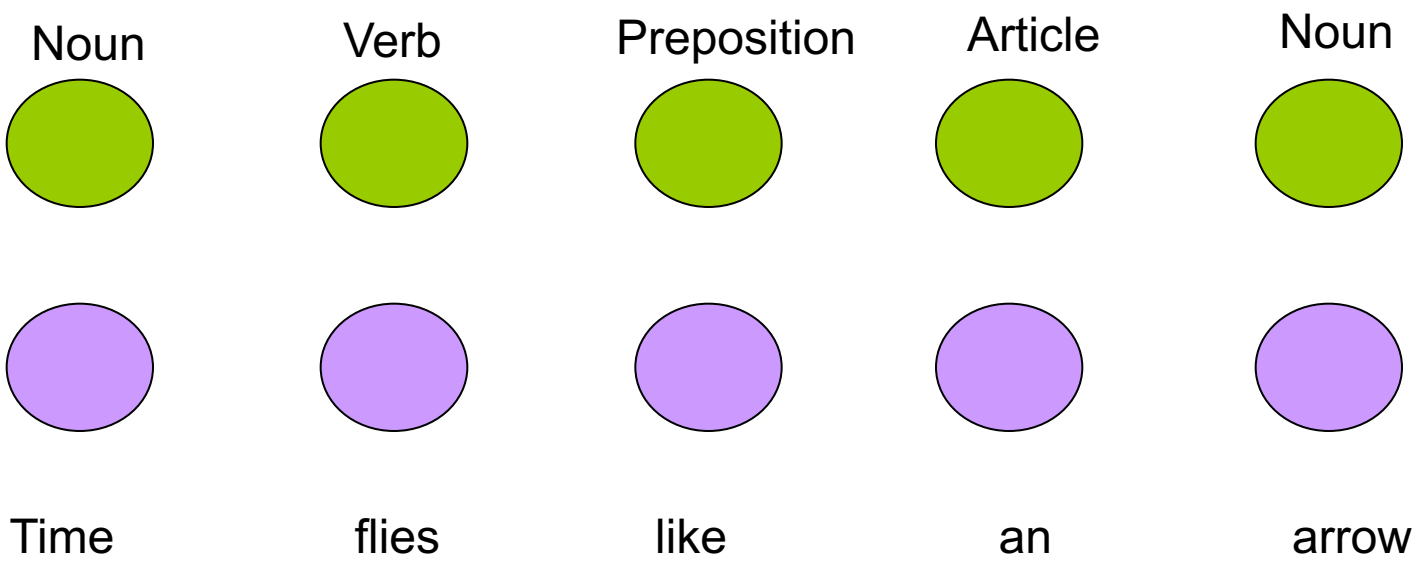
- 描述
  - 小明习惯用两根手指拨电话号码。电话按键的排列如图中所示。最初，他将一根手指放在按键“1”上，并将另一根手指放在按键“2”上，每次他可以将手指上下左右平移，但两根手指永远都挨在一起。若我们用(i, j)表示两个手指放置的位置，那么，如放在按键(5, 6)时，移动一次可以将手指可以移动到(2, 3)，(8, 9)，或者(4, 5)之上。当某个数字键位于任意一根手指之下时，这个按键就可以被按下。
- 输入
  - 第一行包含一个正整数m，表示共有m组测试数据
  - 以下包含m行，每行给定N个0-9的正整数，1 < N < 10000
- 输出
  - 包含m行，每行输出按下输入中的N个数字所需移动手指的最少次数
- 样例输入
  - 26986
  - 74562398701
  - 19880702199108012015050 3
- 样例输出
  - 4
  - 15
  - 45



Example: POS Tagging Problem



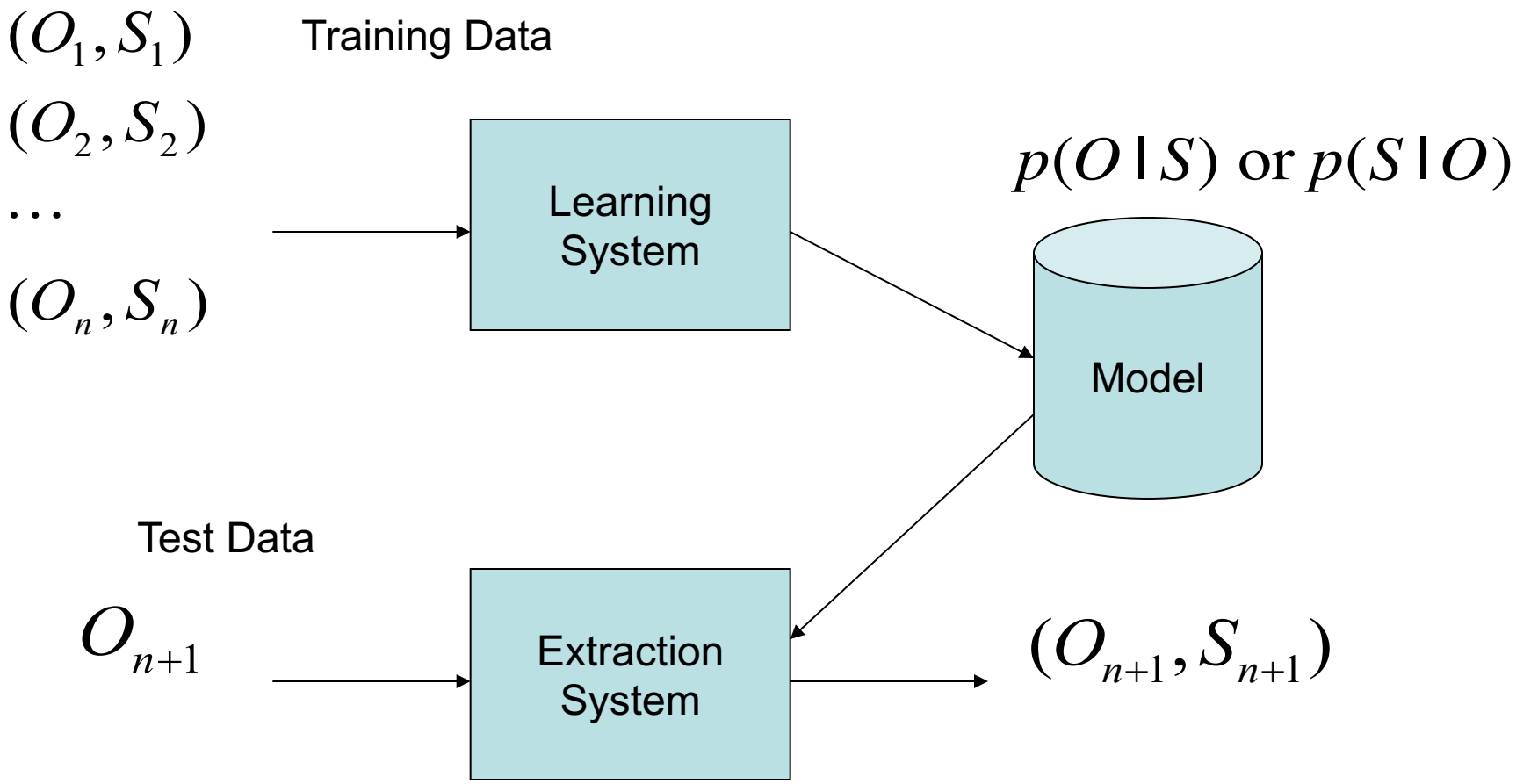
Example: POS Tagging Problem



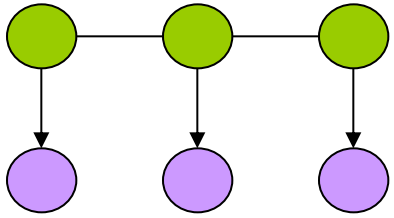
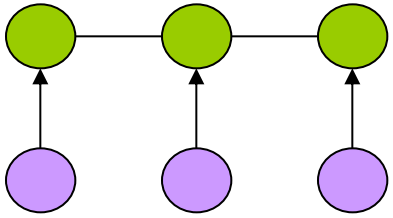
Sequence Labeling Models

- HMM
  - Generative model
  - E.g. Ghahramani (1997), Manning and Schutze (1999)
- MEMM
  - Conditional model
  - E.g. Berger and Pietra (1996), McCallum and Freitag (2000)
- CRFs
  - Conditional model without label bias problem
  - Linear-Chain CRFs
    - E.g. Lafferty and McCallum (2001), Wallach (2004)
  - Non-Linear Chain CRFs
    - Modeling more complex interaction between labels: DCRFs, 2D-CRFs, TCRFs
    - E.g. Sutton and McCallum (2004), Zhu and Nie (2005), Tang et al. (2006)

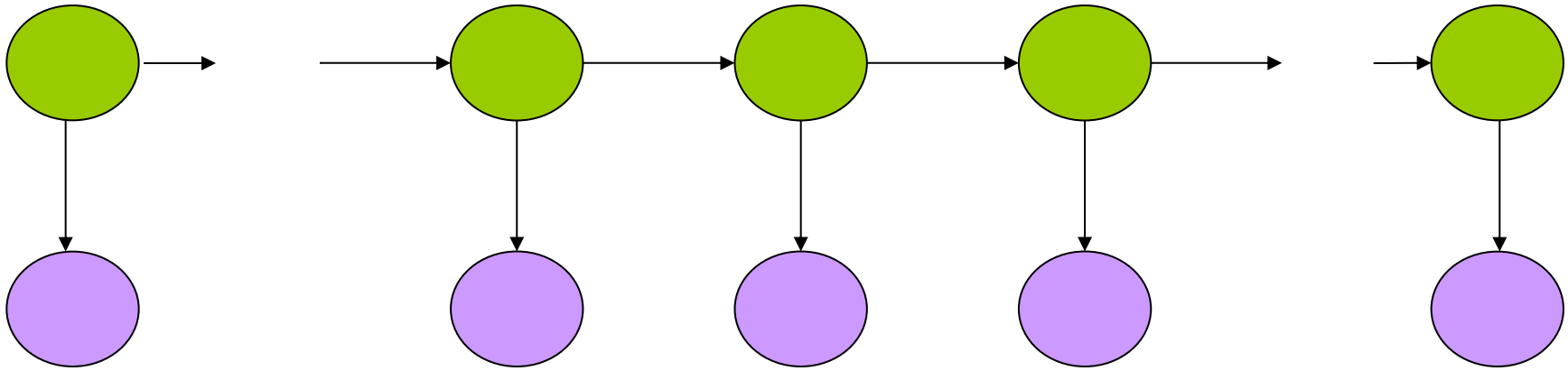
General Framework



Generative vs. Discriminative

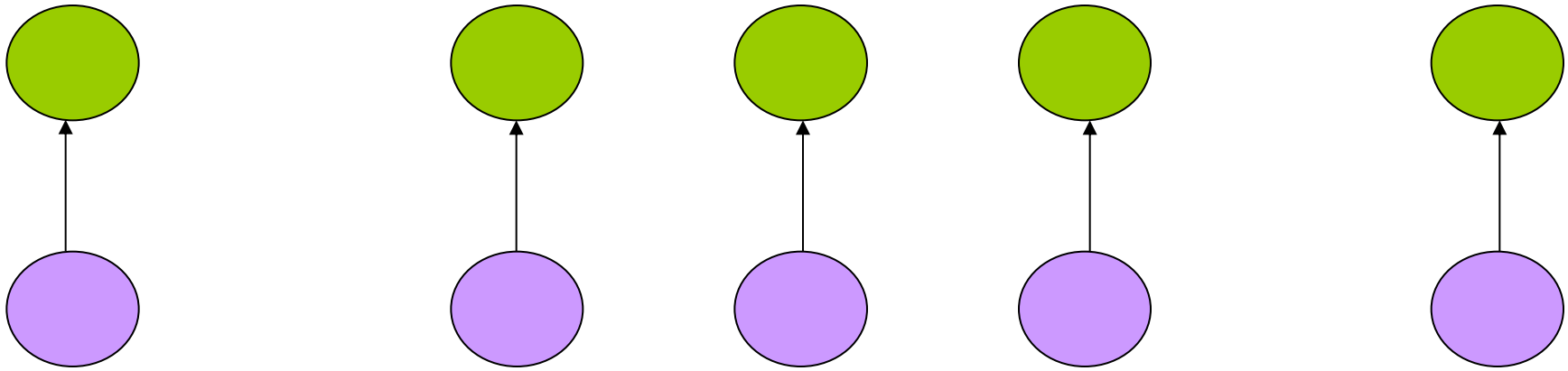
Generative	Discriminative
$P(S\|O)$	$P(S\|O)$
Example: HMM	Example: MaxEnt, MEMM, CRF
 States generates observations	 Observations (features) determine states
Learning = finding model generating observation sequence from state sequence	Learning = finding model mapping observation sequence to state sequence
Tagging = finding most likely state sequence having generated given observation sequence	Tagging = finding most likely state sequence mapped from given observation sequence

Assumption 1: Generative Locally Dependent Model



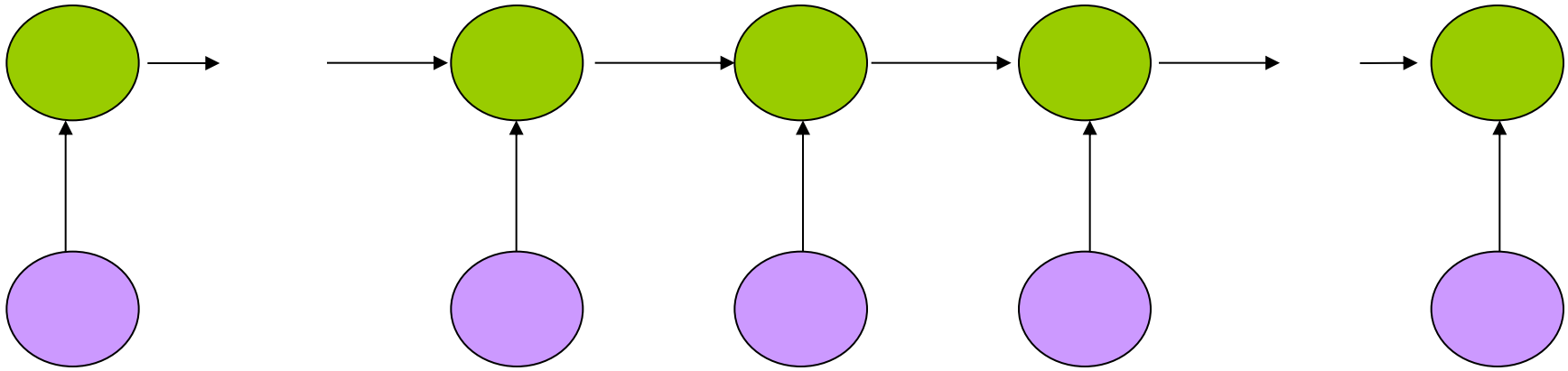
Hidden Markov Model (HMM)

Assumption 2: Discriminative Independent Model



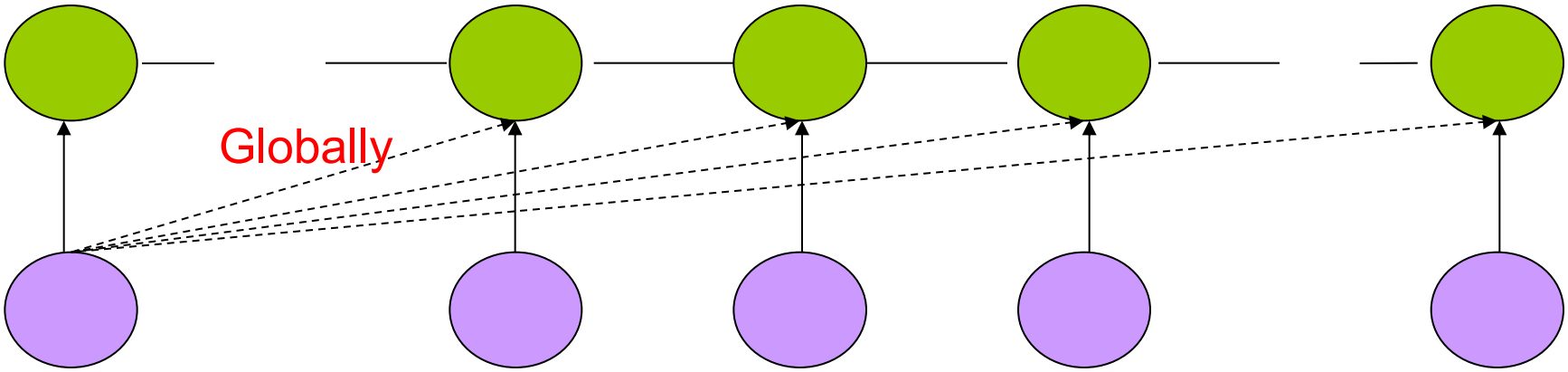
Classifier:
Maximum Entropy Model (ME)
Support Vector Machines (SVM)

Assumption 3: Discriminative Locally Dependent Model



Maximum Entropy Markov Model (MEMM)

Assumption 4: Discriminative Globally Dependent Model



Conditional Random Field (CRF)

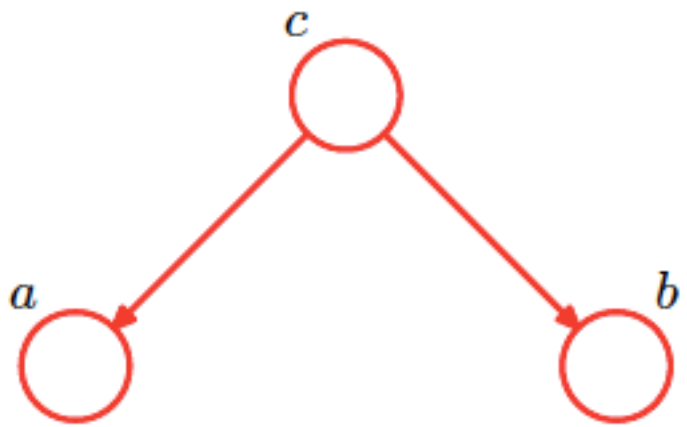
Outline

- Introduction
- Directed Graphical Models
- Undirected Graphical Models
- Factor Graph

Conditional Independence

- Random variables *a*, *b* are independent if p(*a*,*b*)=p(*a*)p(*b*) or p(*a*|*b*)=p(*a*).
- Random variables *a*, *b* are conditionally independent given c if p(*a*,*b*|*c*)=p(*a*|*c*)p(*b*|*c*) or p(*a*|*b*,*c*)=p(*a*|*c*).
- Conditional independence in a Bayesian Network is precisely specified by *d*-separation.

Example 1: Tail-to-Tail



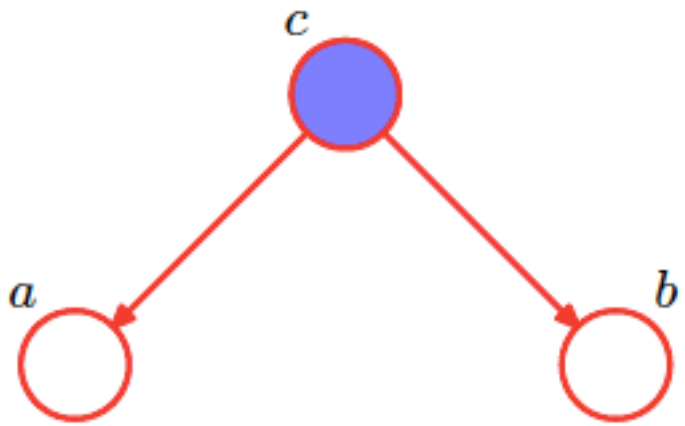
$p(a,b,c)=p(a|c)p(b|c)p(c)$

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

$a \not\perp b|\emptyset$

- Random variable a and b are in general dependent.

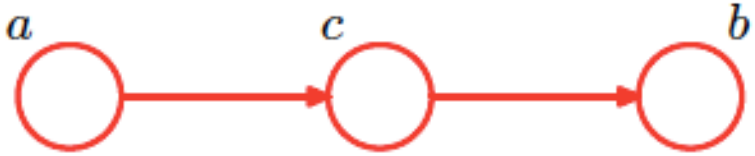
Example 1: Tail-to-Tail



$$p(a,b|c)=p(a,b,c)/p(c)$$
$$=p(a|c)p(b|c)$$
$$a \perp\!\!\!\perp b|c$$

- Random variable a and b are conditionally independent given c.
- An observed c is a tail-to-tail node, blocks the undirected path a-b

Example 2: Head-to-Tail



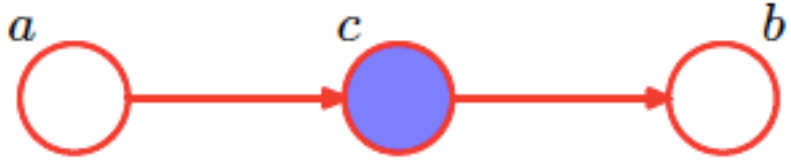
$p(a,b,c)=p(a)p(c|a)p(b|c)$

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

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

- Random variables a and b are in general dependent.

Example 2: Head-to-Tail

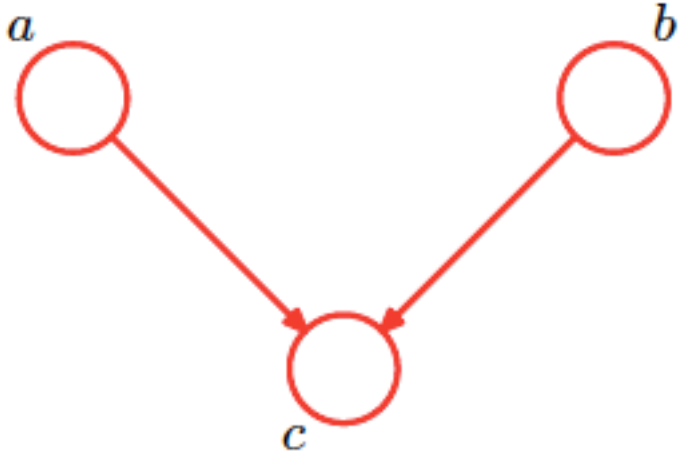


$$p(a,b|c)=p(a,b,c)/p(c)$$
$$=p(a)p(c|a)p(b|c)/p(c)$$
$$=p(a|c)p(b|c)$$

$$a \perp\!\!\!\perp b|c$$

- Random variables a and b are conditionally independent given c.
- An observed c is a head-to-tail node, blocks the path a-b

Example3: Head-to-Head



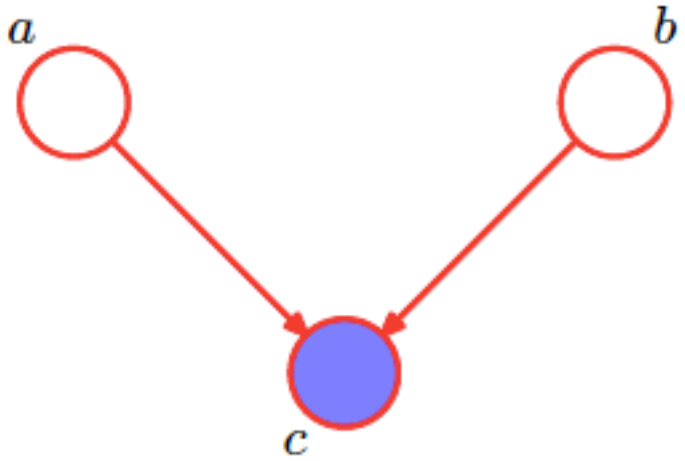
p(a,b,c)=p(a)p(b)p(c|a,b)

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

a⫫b|∅

- Random variables a and b are in general independent.

Example3: Head-to-Head

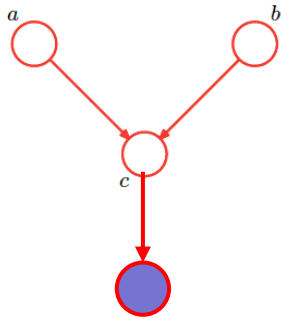


$$p(a,b|c) = p(a,b,c)/p(c)$$
$$= p(a)p(b)p(c|a,b)/p(c)$$
$$a \not\perp b|c$$

- Random variables a and b are dependent given c (observed nodes are shaded)
- An observed c is a head-to-head node, unblocks the undirected path a-b

Summary

- A tail-to-tail node or a head-to-tail node leaves a path unblocked unless it is observed in which case it blocks the path.
- A head-to-head node blocks a path if it is unobserved, but once the node, and/or at least one of its descendants, is observed the path becomes unblocked.



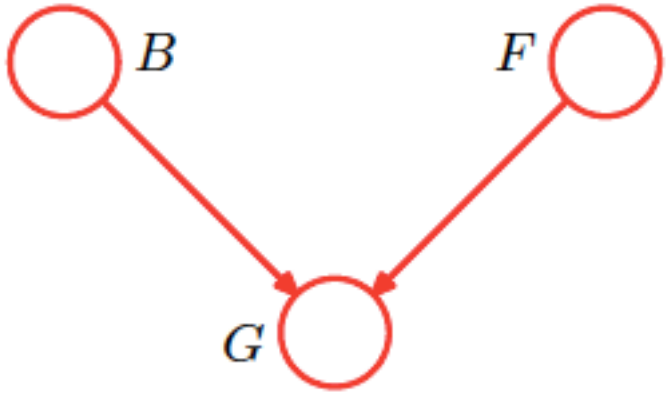
Explaining Away

p(G=1|B=1,F=1) =0.8

p(G=1|B=1,F=0) =0.2

p(G=1|B=0,F=1) =0.2

p(G=1|B=0,F=0) =0.1



p(B=1) =0.9

p(F=1) =0.9

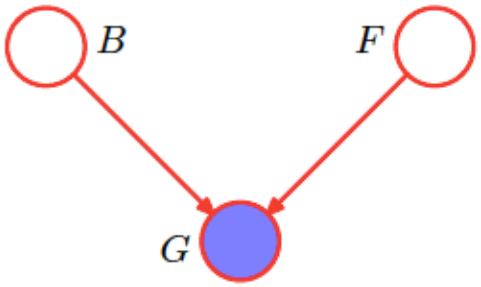
B = Battery (0=flat, 1=fully charged)

F = Fuel Tank (0=empty, 1=full)

G=Fuel Gauge Reading (0=empty, 1=full)

A number of the following pages are from Bishop's slides

Explaining Away



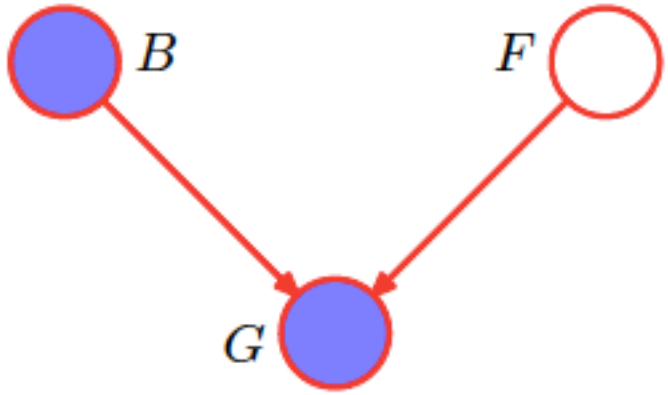
$$p(G=0) = \sum_{B\in\{0,1\}} \sum_{F\in\{0,1\}} p(G=0|B,F)p(B)p(F) = 0.315$$

$$p(G=0|F=0) = \sum_{B\in\{0,1\}} p(G=0|B,F=0)p(B) = 0.81$$

$$p(F=0|G=0) = \frac{p(G=0|F=0)p(F=0)}{p(G=0)} \simeq 0.257$$

- Probability of an empty tank increased by observing G = 0 (p(F=0) = 0.1).

Explaining Away



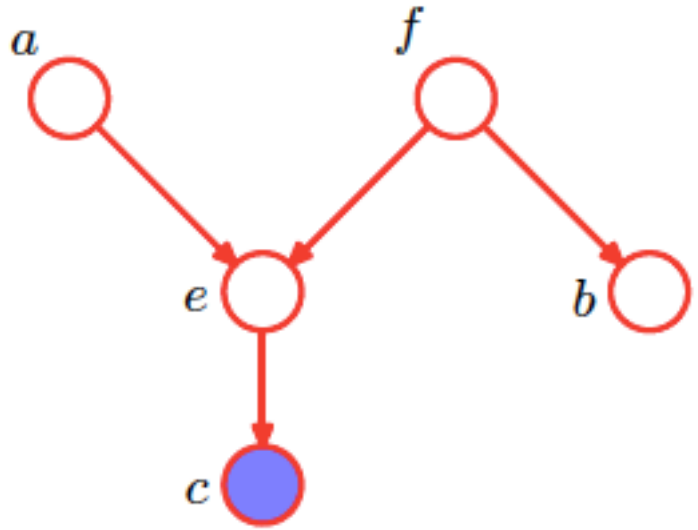
$$p(F=0|G=0,B=0) = \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(F)} \simeq 0.111$$

- Probability of an empty tank reduced by observing B = 0.

D-Separation

- A path from A to B is blocked by C if it includes a node x such that either
  - The path is head-to-tail or tail-to-tail at x and $x \in C$, or
  - The path is head-to-head at x, and neither x nor any of its descendants is in C.
- If all paths from A to B are blocked, A is said to be d-separated from B by C, and the joint distribution over all the variables in the graph will satisfy $A \perp\!\!\!\perp B|C$

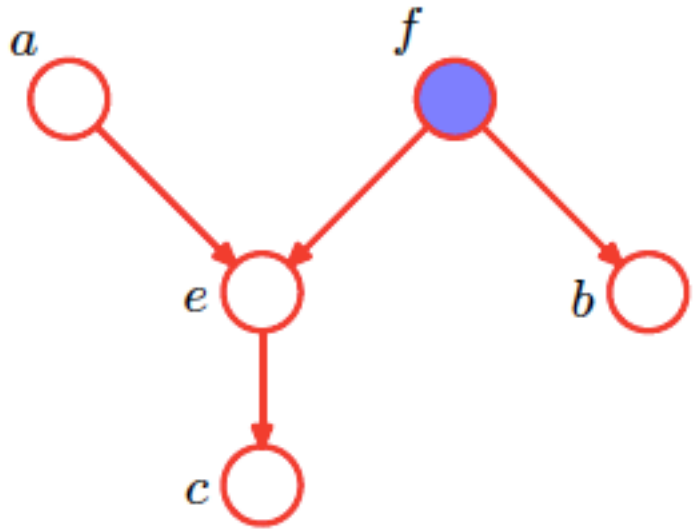
D-Separation Example 1



$a \not\perp b \mid c$

- The path a-b is not blocked by node f because it is a tail-to-tail node for this path and is not observed.
- The path a-b is not blocked by node e because although it is a head-to-head node, it has a descendant c because is in the conditioning set.

D-Separation Example 2

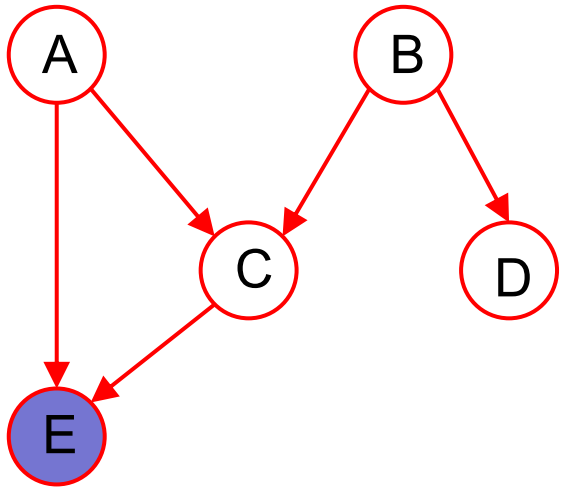


$a \perp\!\!\!\perp b \mid c$

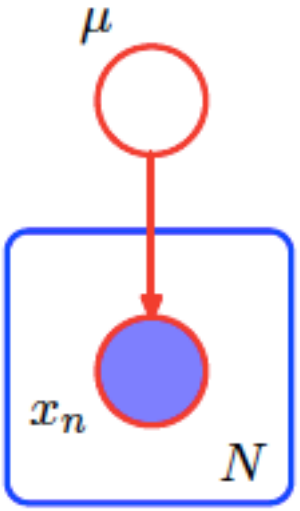
- The path a-b is blocked by node f because this is a tail-to-tail node that is observed.
- The path a-b is also blocked by node e because e is a head-to-head node and neither it nor its descendent are in the conditioning set.

Question

- Assume that the value of E is observed, which pairs of variables are independent in the following Bayesian network? You may select 1 or more options.
- 1) None
- 2) A, B
- 3) A, C
- 4) A, D
- 5) B, C
- 6) B, D
- 7) C, D



D-separation: Infer μ from D={x_1,…,x_N}



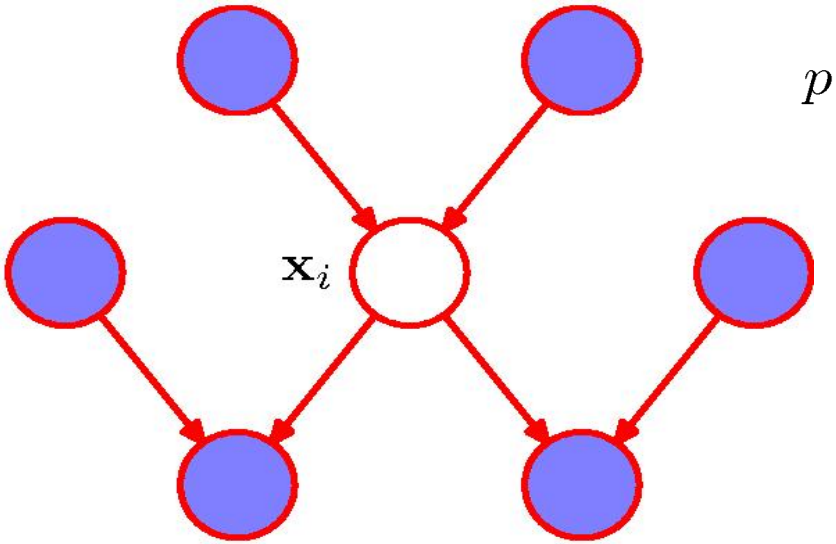
- Suppose we consider the joint distribution of the observations condition on μ.
- Then there is a unique path from any x_i to any other x_j and the path is tail-to-tail with observed node μ.
- Thus every such path is blocked and the observations D are independent given μ, so that

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

However, if we integrate over μ, the observations are in general no longer independent

$$p(\mathcal{D}) = \int_{-\infty}^{\infty} p(\mathcal{D}|\mu)p(\mu)\,\mathrm{d}\mu \neq \prod_{n=1}^{N} p(x_n)$$

The Markov Blanket



$$p(\mathbf{x}_i|\mathbf{x}_{\{j\neq i\}}) = \frac{p(\mathbf{x}_1,\ldots,\mathbf{x}_M)}{\int p(\mathbf{x}_1,\ldots,\mathbf{x}_M)\,\mathrm{d}\mathbf{x}_i}$$

$$= \frac{\prod_k p(\mathbf{x}_k|\mathrm{pa}_k)}{\int \prod_k p(\mathbf{x}_k|\mathrm{pa}_k)\,\mathrm{d}\mathbf{x}_i}$$

- Factors independent of x_i cancel between numerator and denominator.

Directed Graphic Models

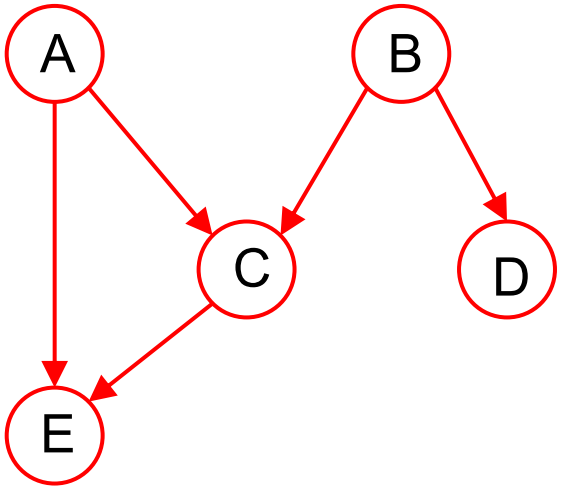
- Also called Bayesian networks.
- A Bayesian network on the DAG is a family of distributions satisfying

$$\{p \mid p(X) = \prod_i p(x_i \mid Pa(x_i))\}$$

  - where $Pa(x_i)$ is the set of parents of x_i
  - $p(x_i|Pa(x_i))$ is the conditional probability distribution (CPD) at x_i.

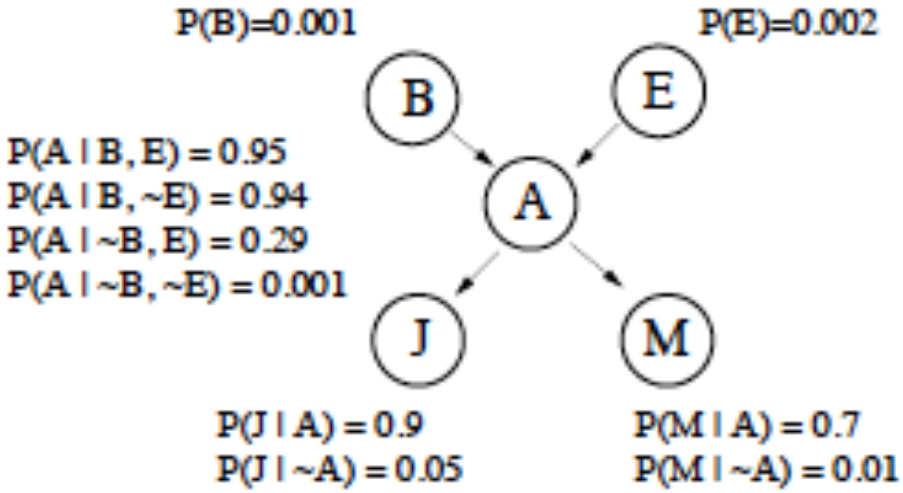
Question

- Given the following Bayesian network, which of these is an appropriate decomposition of the joint distribution P(A,B,C,D,E)?



- 1) P(A)P(B)P(C|A)P(C|B)P(D|B)P(E|A)P(E|C)
- 2) P(A)P(B)P(C|A,B)P(D|B)P(E|A,C)

Example 1: Alarm



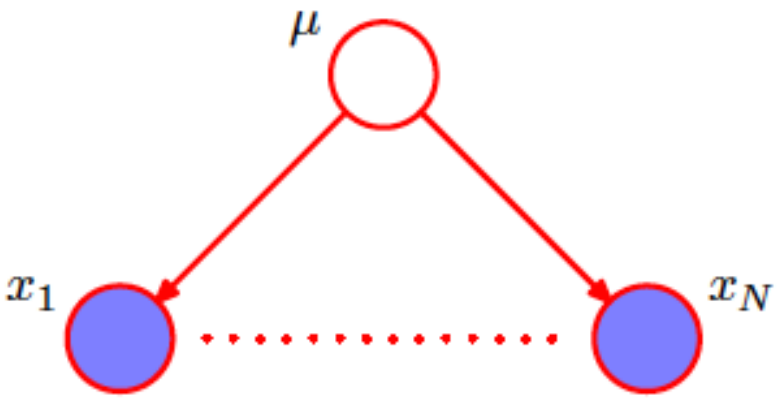
p(B, ~E, A, J, ~M)

= p(B) p(~E) p(A|B, ~E) p(J|A) p(~M|A)

= 0.001×(1-0.002) ×0.94×0.9×(1-0.7)

≃ 0.000253

Example 2: Naïve Bayes



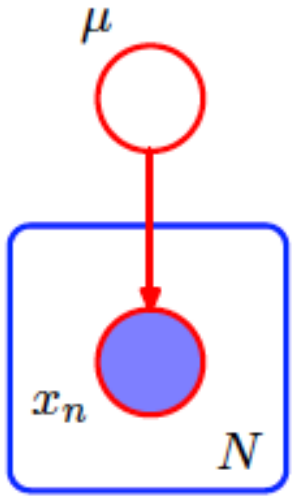
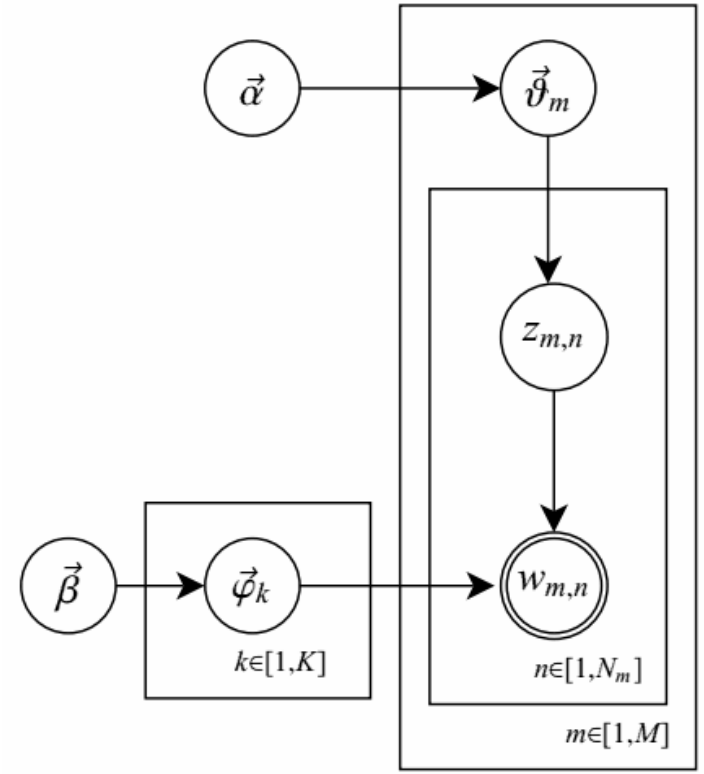


Plate notation

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

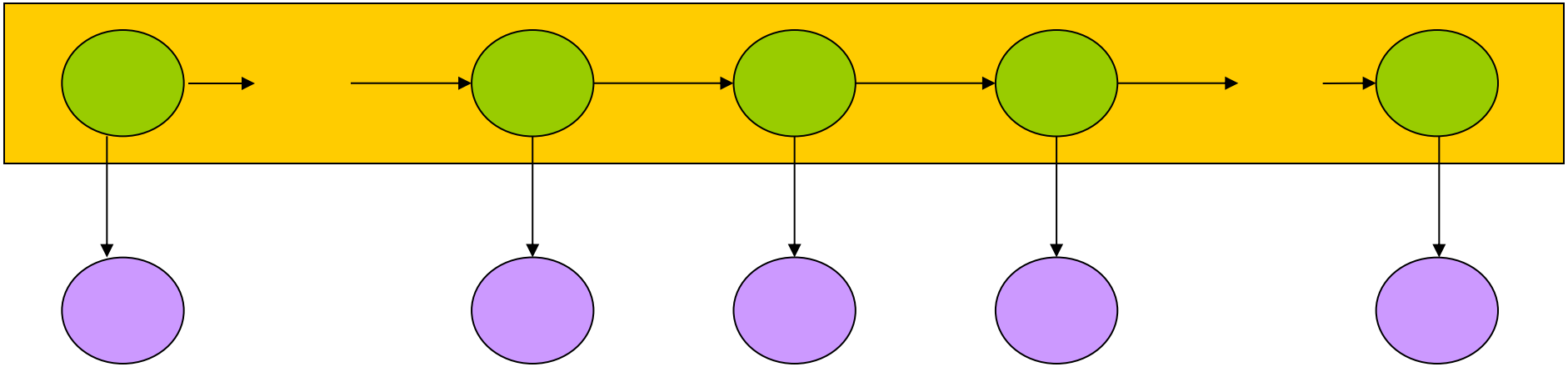
Example 3: Latent Dirichlet Allocation



$$P(\mathcal{W},\mathcal{Z},\Theta,\Phi|\alpha,\beta) = \prod_{k=1}^{K} P(\vec{\phi}_k|\vec{\beta}) \prod_{m=1}^{M} \left(P(\vec{\theta}_m|\vec{\alpha}) \prod_{n=1}^{N_m} P(w_{m,n}|\vec{\phi}_{z_{m,n}}) P(z_{m,n}|\vec{\theta}_m) \right)$$

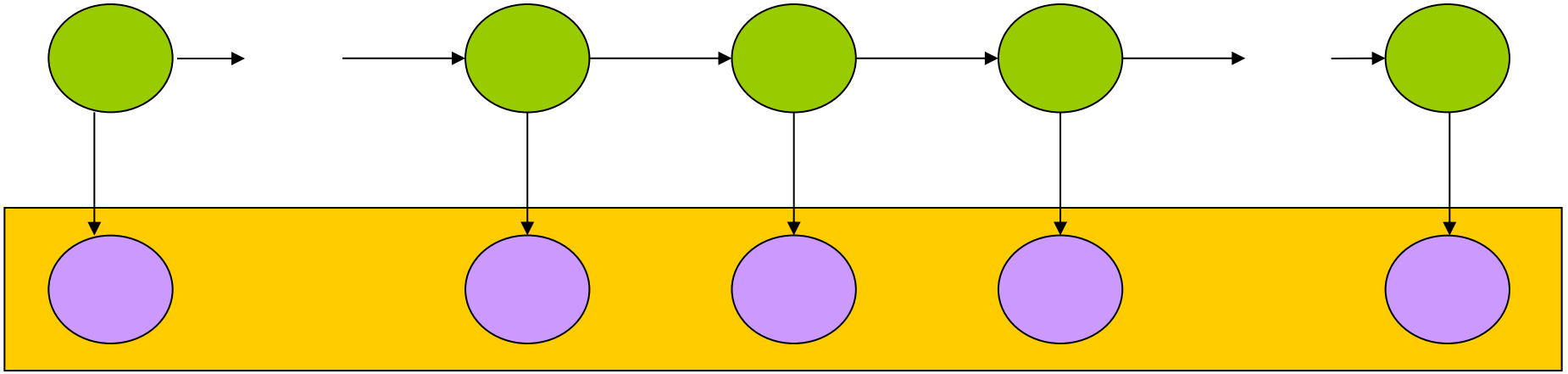
Example 4: Hidden Markov Model (HMM)

What is HMM?



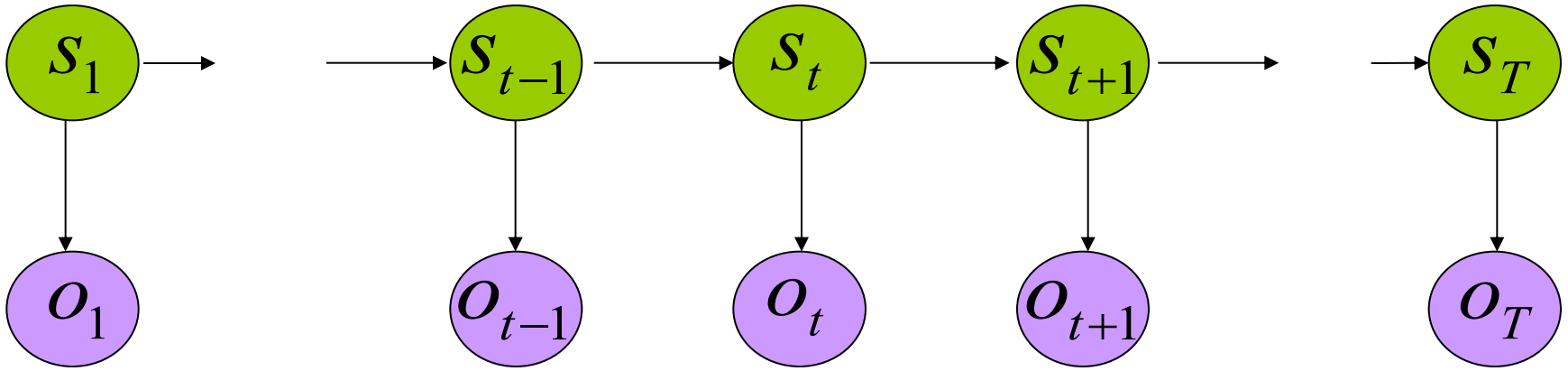
- Green nodes are '***hidden' states***
- State depends only on previous state

What is HMM?



- Purple nodes are ***observations***
- Each state generates an observation

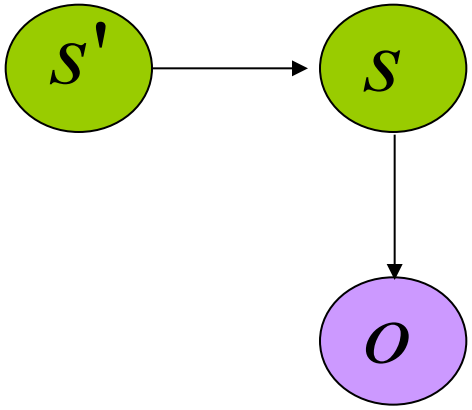
HMM Formalism



- s : {1,2,…,N} are values of hidden states
- o : {1,2,…,M} are values of observations

$$P(O,S) = P(s_1)P(o_1 \mid s_1)\prod_{t=2}^{T} P(s_t \mid s_{t-1})P(o_t \mid s_t)$$

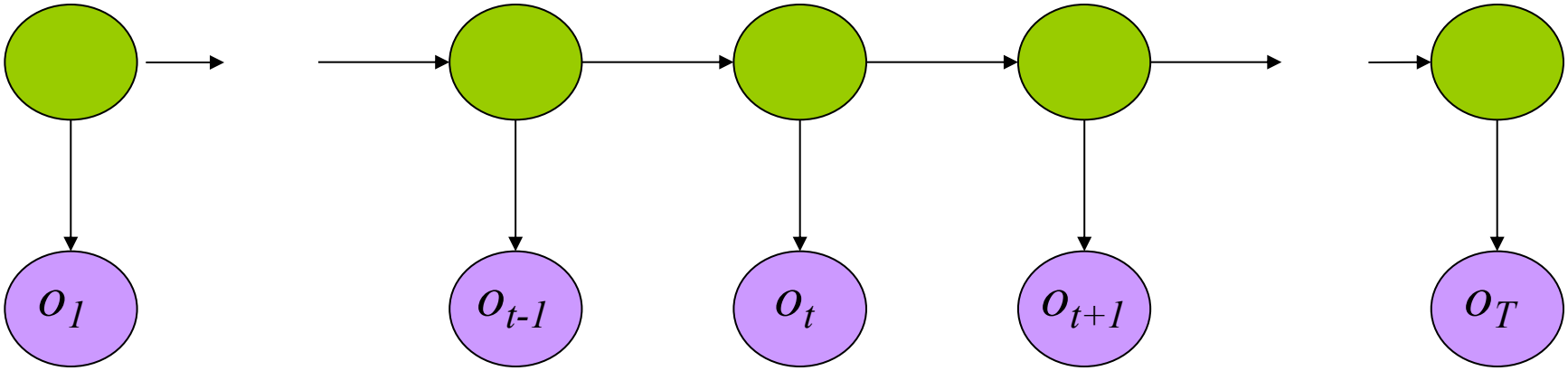
HMM Formalism



$$P(s \mid s')$$

$$P(o \mid s)$$

Tagging



- Viterbi algorithm
  - given observation sequence, compute most likely having generated state sequence

$$\arg\max_S P(S \mid O) = \arg\max_S P(S, O)$$

Summary of HMM

Model

•Baum,1966; Manning, 1999

Applications

•POS tagging (Kupiec, 1992)
•Shallow parsing (Molina, 2002; Ferran Pla, 2000; Zhou, 2000)
•Speech recognition (Rabiner, 1989; Rabiner 1993)
•Gene sequence analysis (Durbin, 1998)
•…

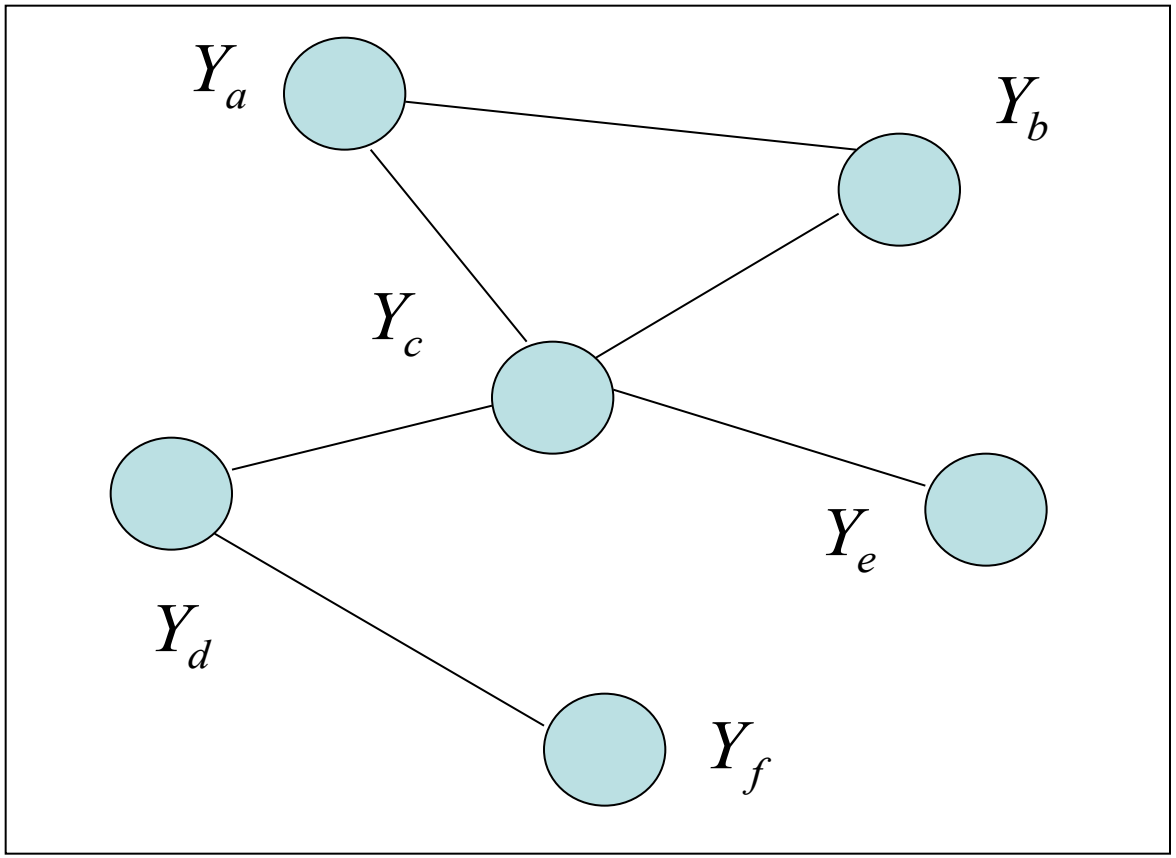
Limitation

•Joint probability distribution $p(x, s)$.
•Cannot represent overlapping features and long range dependences long range dependences between observed elements.

Outline

- Introduction
- Directed Graphical Models
- Undirected Graphical Models
- Factor Graph

Markov Random Field



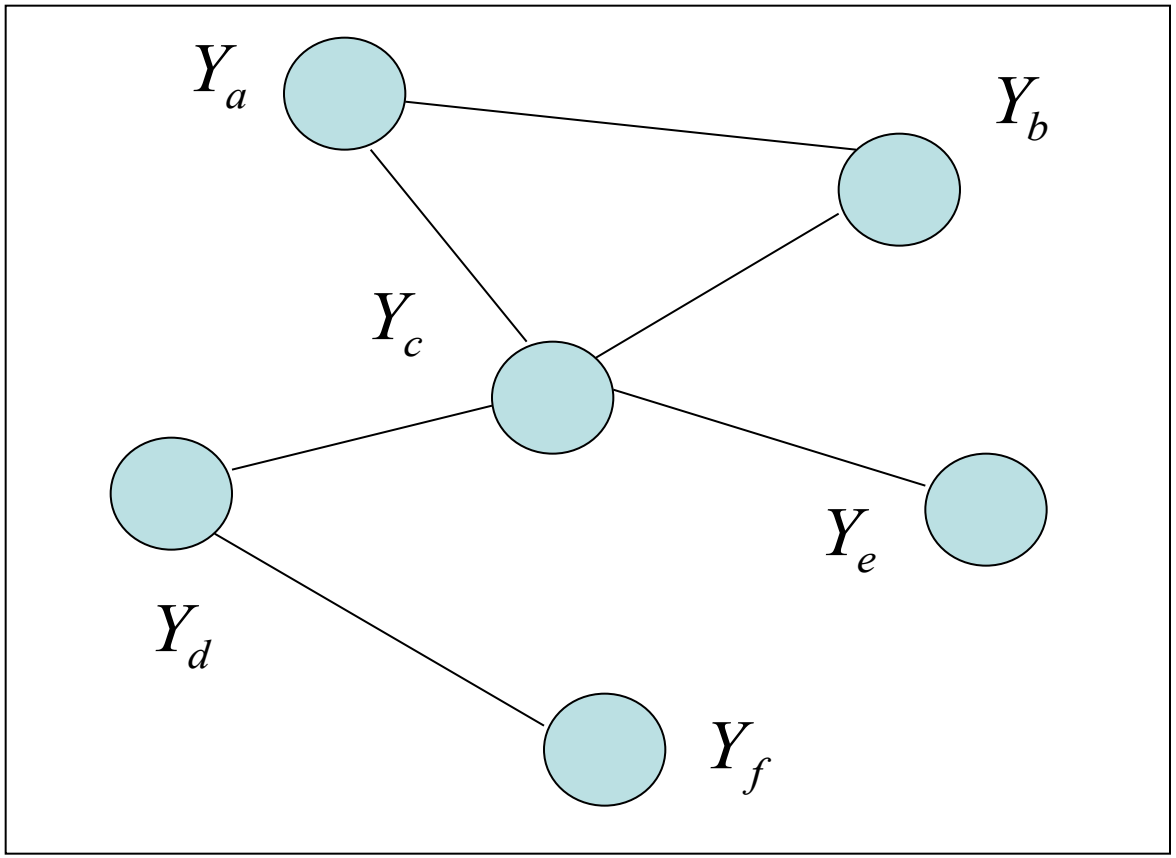
Given an undirected graph $G=(V, E)$ such that $Y=\{Y_v|v\in V\}$, if

The probability of Y_v given X and those random variables corresponding to nodes neighboring v in G. Then (X, Y) is a conditional random field.

Undirected graphical model globally conditioned on X

$$p(Y_v \mid X, Y_u, u \neq v, \{u,v\} \in V) \Leftrightarrow p(Y_v \mid X, Y_u, (u,v) \in E)$$

Markov Random Field



By the Hammersley-Clifford theorem, the probability of a label can be expressed as a Gibbs distribution, so that

$$p(Y) = \frac{1}{Z}\prod_{C}\varphi(Y_C)$$

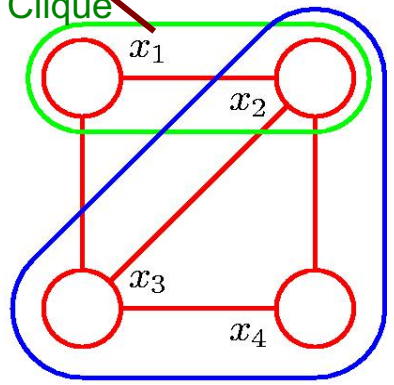
$$Z = \sum_{Y}\prod_{C}\varphi(Y_C)$$

Energy function

$$\varphi(Y_C) = \exp\{-E(Y_C)\}$$

Clique

Maximal Clique



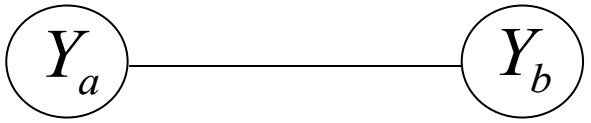
Undirected Graphic Model

- Also called Markov Random Fields
- An undirected graphical model is a family of distributions satisfying

$$\left\{ p \mid p(Y) = \frac{1}{Z} \prod_{C} \varphi(Y_C) \right\}$$

  - where z is the partition function by integrating all the possible values of *X*.

A Toy Example—MRF



$Y_a, Y_b \in \{-1, 1\}$

one clique $\varphi_C(Y_a, Y_b) = e^{aY_aY_b}$

$$p(Y_a, Y_b) = \frac{1}{Z} e^{aY_aY_b}, Z = (e^a + e^a + e^{-a} + e^{-a})$$

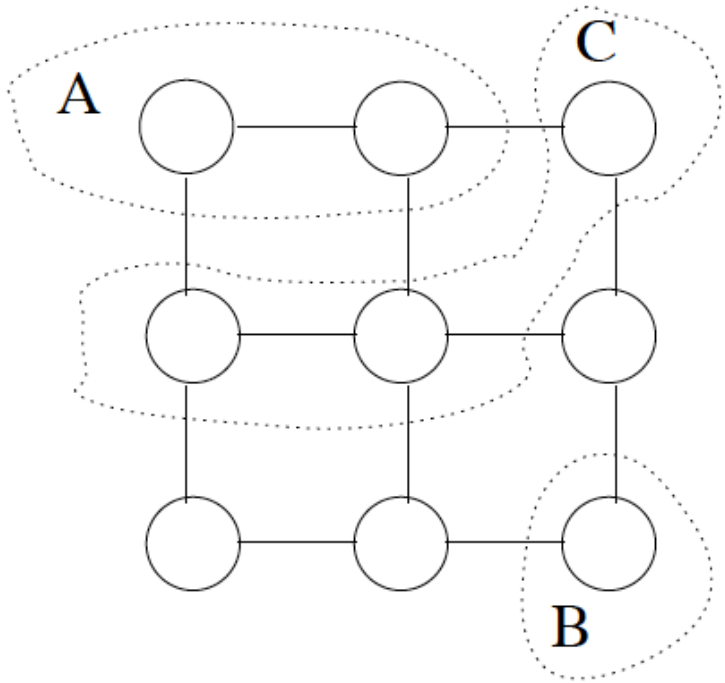
$$p(1,1) = p(-1,-1) = \frac{e^a}{2e^a + 2e^{-a}}$$

Note: when $a>0$, the model favors homogeneous chain

when $a<0$, the model favors inhomogeneous chain

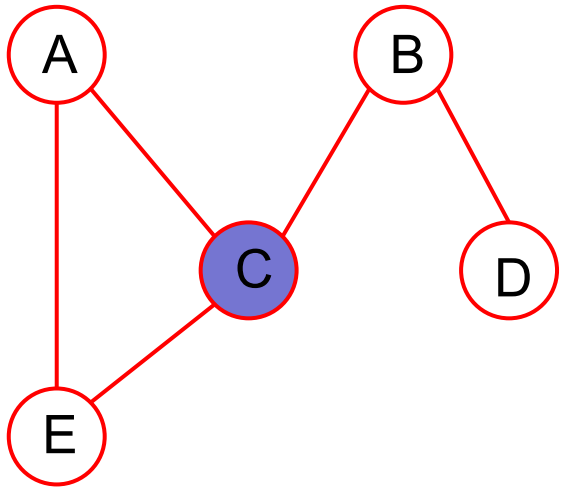
Conditional Independence in Markov Random Fields

- Two group of variables A, B are conditional independent given another group C, if:
- A, B become disconnected by removing C

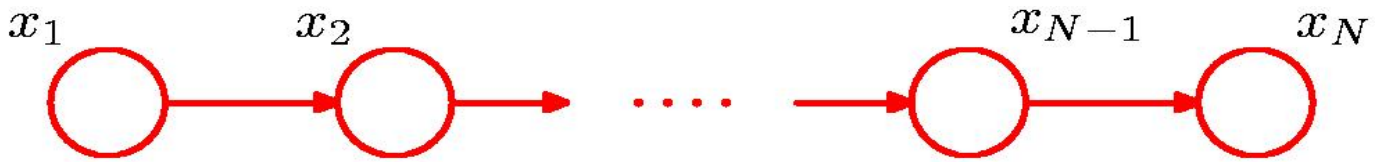


Question

- Assume that the value of C is observed, which pairs of variables are independent in the following Markov network? You may select 1 or more options.
- 1) None
- 2) A, B
- 3) A, D
- 4) A, E
- 5) B, D
- 6) B, E
- 7) D, E

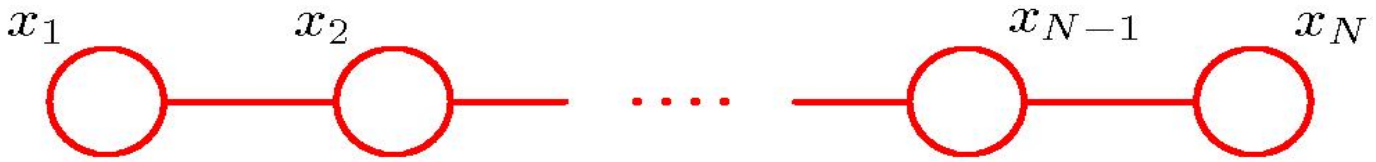


Relationship between the Undirected Graphs and the Directed Graphs



$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\ p(x_3|x_2) \cdots p(x_N|x_{N-1})$$

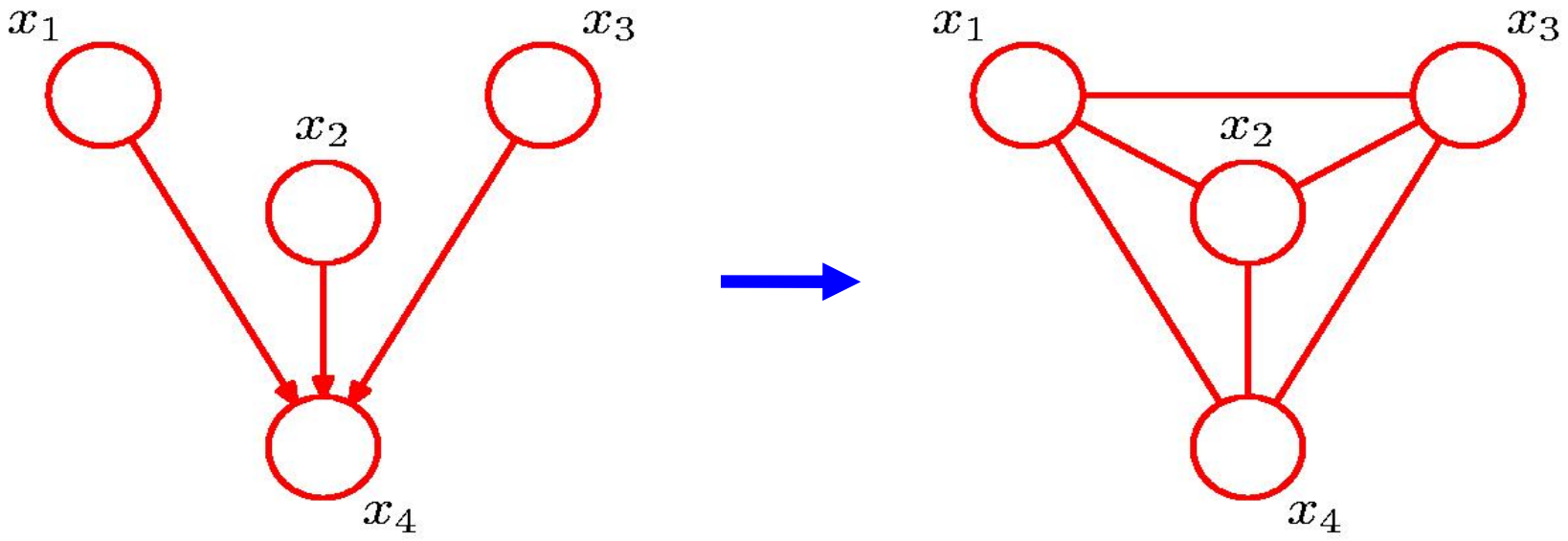
$$p(\mathbf{x}) = \frac{1}{Z}\ \psi_{1,2}(x_1, x_2)\ \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)$$



A number of the following pages are from Bishop's slides

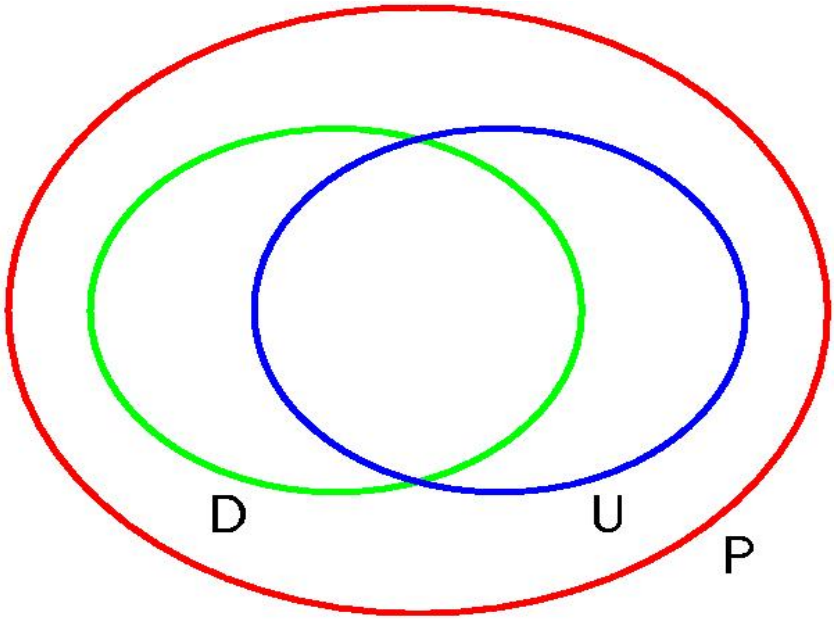
Converting Directed to Undirected Graphs

Additional links

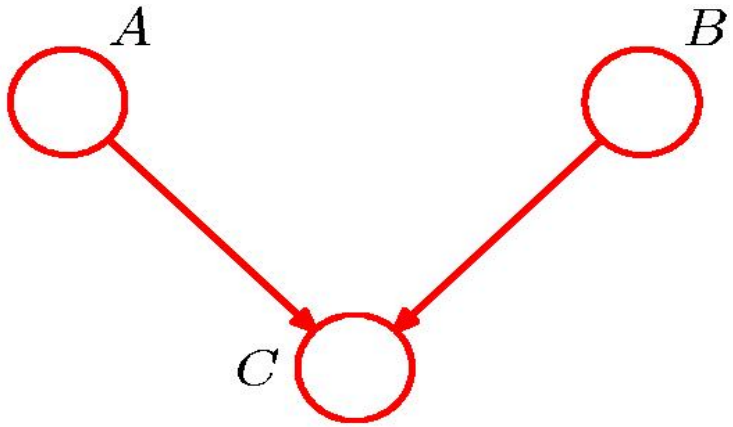


$$
\begin{aligned}
p(\mathbf{x}) &= p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3) \\
&= \frac{1}{Z}\psi_A(x_1,x_2,x_3)\psi_B(x_2,x_3,x_4)\psi_C(x_1,x_2,x_4)
\end{aligned}
$$

Directed vs. Undirected Graphs (1)

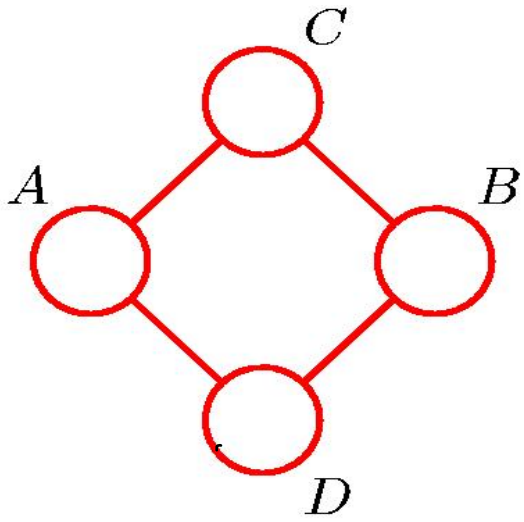


Directed vs. Undirected Graphs (2)



$$A \perp\!\!\!\perp B \mid \emptyset$$

$$A \not\perp\!\!\!\perp B \mid C$$



$$A \not\perp\!\!\!\perp B \mid \emptyset$$

$$A \perp\!\!\!\perp B \mid C \cup D$$

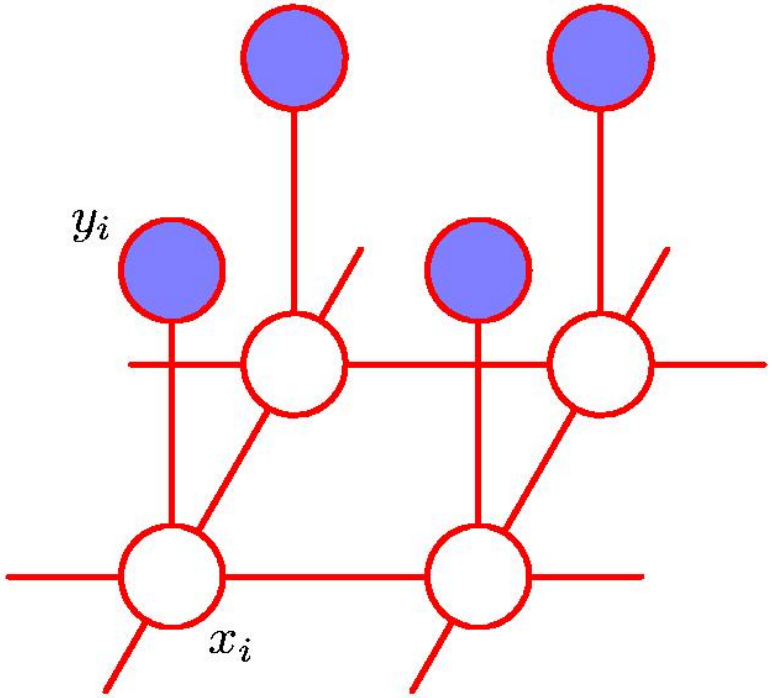
$$C \perp\!\!\!\perp D \mid A \cup B$$

Example 2: Log Linear Models

- Real-valued feature function $f_1(X),\ldots, f_k(X)$
- Real-valued weights $w_1,\ldots, w_k$

$$p(X) = \frac{1}{Z} \exp\left(-\sum_{i=1}^{k} w_i f_i(X)\right)$$

Example 3: The Ising Model



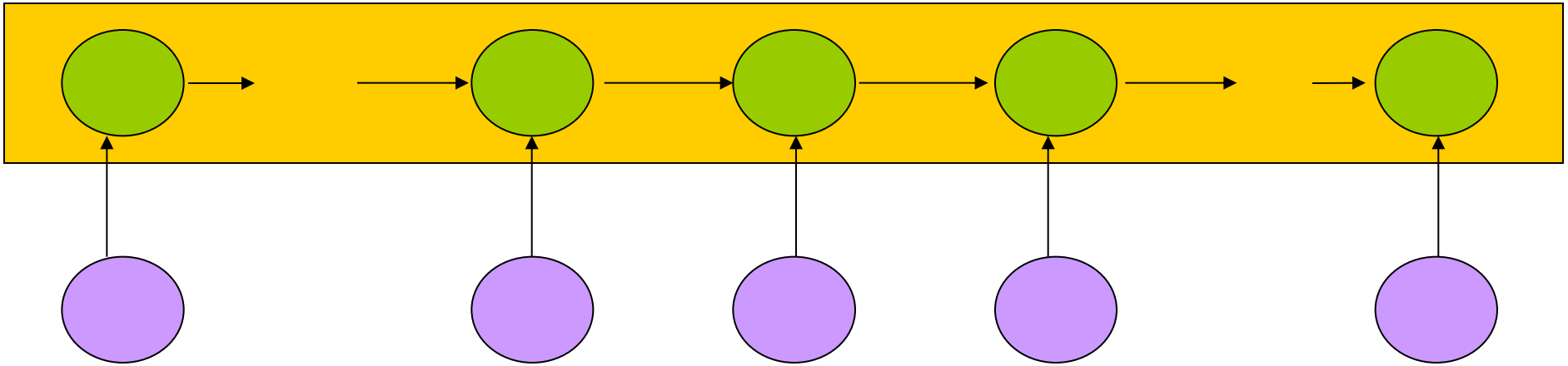
$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\}$$

$x \in \{0,1\}$

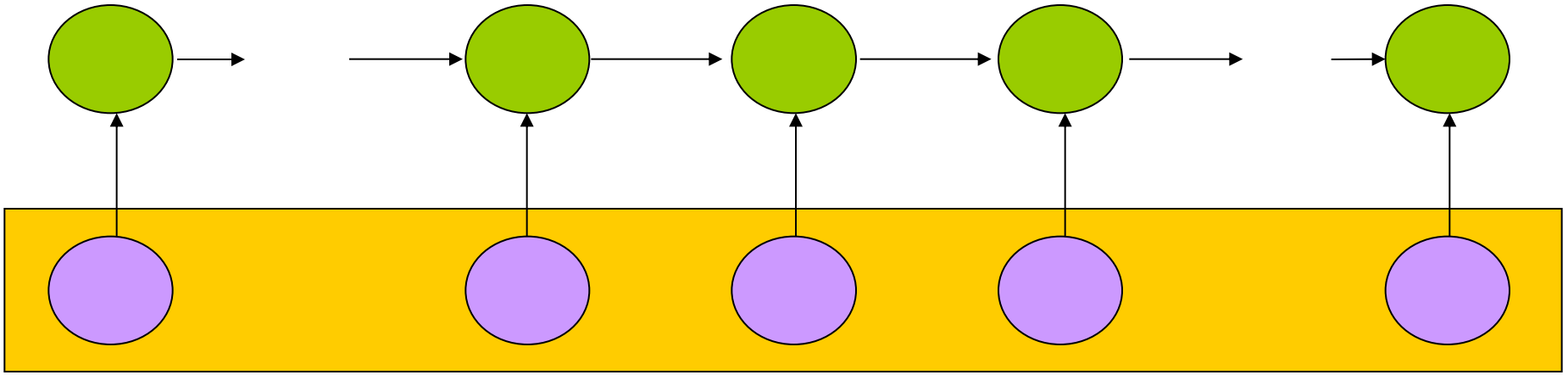
Example 4: Maximum Entropy Markov Model (MEMM)

What is MEMM?



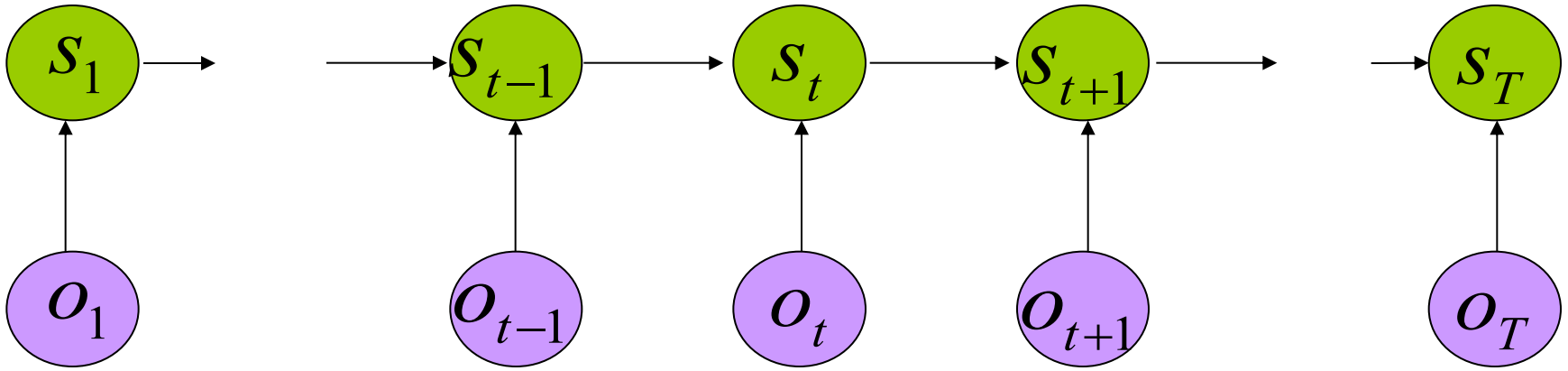
- Green nodes are ***states***
- State depends only on previous state

What is MEMM?



- Purple nodes are ***observations***
- Observations (features of observations) determine states

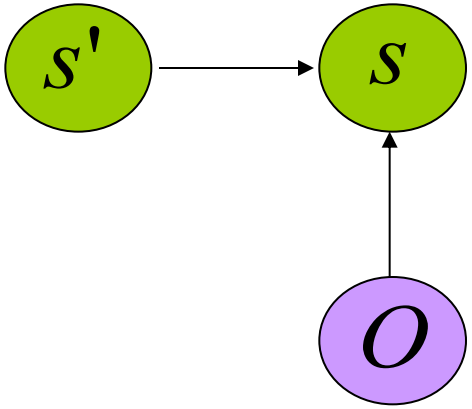
MEMM Formalism



- s : {1,2,…,N } are values for states
- o: {1,2,…,M } are values for observations

$$P(S \mid O) = P(s_1 \mid O) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, O)$$

MEMM Formalism



$$s \leftarrow y, s' \leftarrow y', O \leftarrow x$$

$$P(s \mid s', O) = P(y \mid y', x)$$

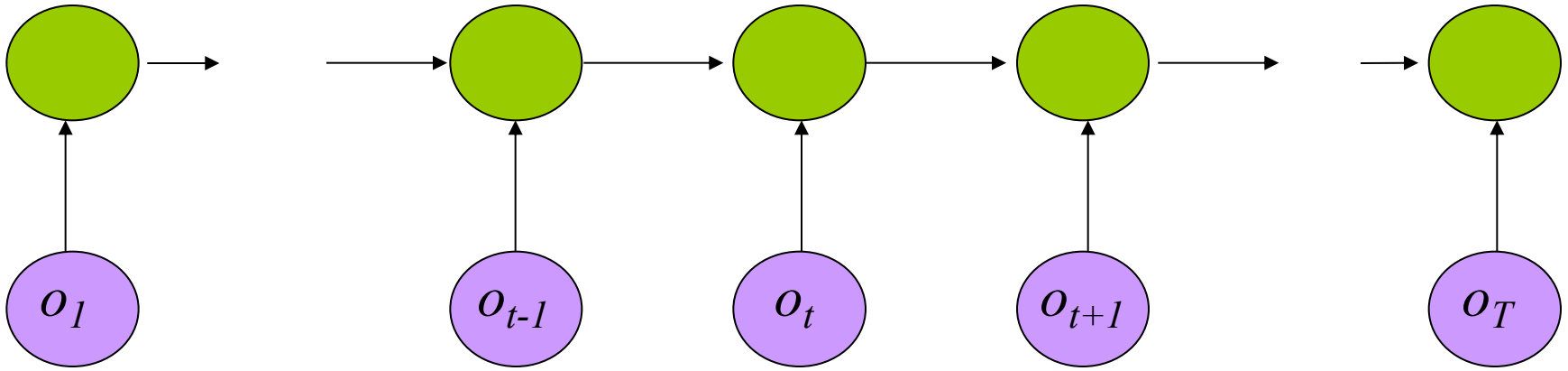
$$P(y \mid y', x) = \frac{\exp(\sum_k \lambda_k f_k(x, y', y))}{Z(y', x)}$$

$$Z(y', x) = \sum_y \exp(\sum_k \lambda_k f_k(x, y', y))$$

Inference in MEMM

- Tagging: given observation sequence, find most likely corresponding state sequence
- Learning: given observation sequence and corresponding state sequence, find model that best explains the matching

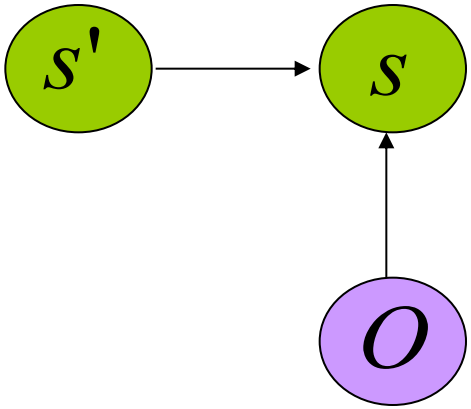
Tagging



- Viterbi algorithm

$$\arg\max_S P(S \mid O) = \arg\max_S P(s_1 \mid O) \prod_{t=1}^{T} P(s_t \mid s_{t-1}, O)$$

Learning



$$(x_1, y_1', y_1), (x_2, y'_2, y_2), \cdots, (x_n, y'_n, y_n)$$

$$P(y \mid y', x) = \frac{\exp(\sum_k \lambda_k f_k(x, y', y))}{Z(x, y')}$$

$$Z(x, y') = \sum_y \exp(\sum_k \lambda_k f_k(x, y', y))$$

$$\arg\max \sum_{i=1}^{n} \log P(y_i \mid y'_i, x_i)$$

Learning Algorithm: IIS

Algorithm 1 : Improved Iterative Scaling

Input: Feature functions $f_1, f_2, \ldots f_n$; empirical distribution $\tilde{p}(x, y)$
Output: Optimal parameter values $\lambda^\star{}_i$; optimal model $p_{\lambda^\star}$

1. Start with $\lambda_i = 0$ for all $i \in \{1, 2, \ldots, n\}$
2. Do for each $i \in \{1, 2, \ldots, n\}$:
   a. Let $\Delta\lambda_i$ be the solution to

$$\sum_{x,y} \tilde{p}(x)p(y|x)f_i(x,y)\exp\Big(\Delta\lambda_i f^{\#}(x,y)\Big) = \tilde{p}(f_i) \tag{16}$$

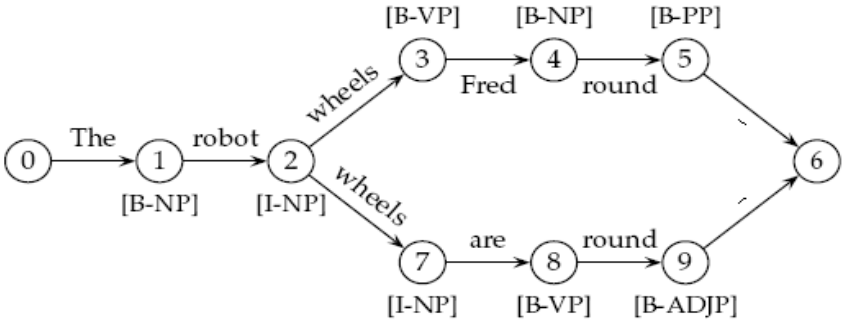
$$\text{where} \quad f^{\#}(x,y) \equiv \sum_{i=1}^{n} f_i(x,y) \tag{17}$$

   b. Update the value of λ_i according to: $\lambda_i \leftarrow \lambda_i + \Delta\lambda_i$
3. Go to step 2 if not all the λ_i have converged

Summary of MEMM

- Discriminative model
- Conditional assumption
- Accuracy is higher than MaxEnt, lower than CRF
- Problem: local model → label bias problem
- MEMM contains MaxEnt as special case

Label Bias Problem



The finite-state acceptor is designed to shallow parse the sentences (chunk/phrase parsing)
1) the robot wheels Fred round
2) the robot wheels are round

Decoding it by:
0123456
0127896

$$p(s \mid o) = p(s_1 \mid o_1) \prod_{t=2}^{T} p(s_t \mid s_{t-1}, o_t)$$

Assuming the probabilities of each of the transitions out of state 2 are approximately equal, the label bias problem means that the probability of each of these chunk sequences given an observation sequence *x* will also be roughly equal irrespective of the observation sequence *x*.

On the other hand, had one of the transitions out of state 2 occurred more frequently in the training data, the probability of that transition would always be greater. This situation would result in the sequence of chunk tags associated with that path being preferred irrespective of the observation sentence.

Summary of MEMM

Model

•Berger, 1996; Ratnaparkhi 1997, 1998

Applications

•Segmentation (McCallum, 2000)

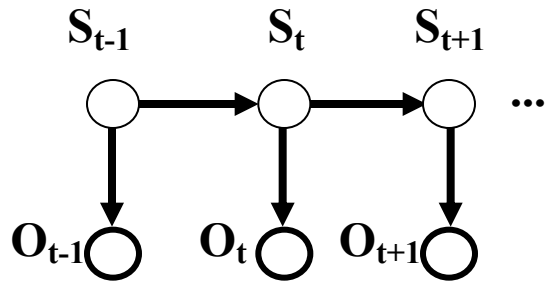
•…

Limitation

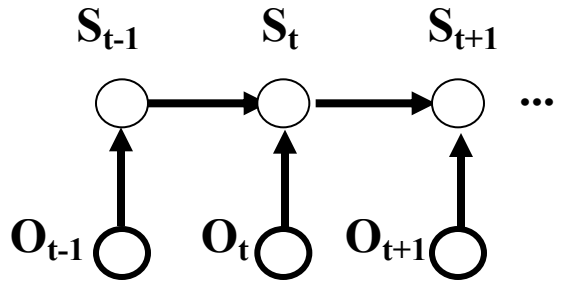
•Label bias problem (HMM do not suffer from the label bias problem)

Conditional Markov Models (CMMs) aka MEMMs aka Maxent Taggers *vs* HMMs

$$p(s,o) = \prod_{t=1}^{T} p(s_t \mid s_{t-1}) p(o_t \mid s_{t-1})$$



$$p(s \mid o) = \prod_{t=1}^{T} p(s_t \mid s_{t-1}, o_{t-1})$$



Example 5: CRFs

MEMM to CRFs

$$p(y_1 \ldots y_N \mid x_1 \ldots x_N) = \prod_{n=1}^{N} p(y_n \mid y_{n-1}, x_n) = \prod_{n=1}^{N} \frac{\exp(\sum_{k=1}^{K} \lambda_k f_k(x_n, y_n, y_{n-1}))}{Z(x_n)}$$

$$= \frac{\exp(\sum_k \lambda_k F_k(\vec{x}, \vec{y}))}{\prod_n Z(x_n)}, \text{ where } F_k(\vec{x}, \vec{y}) = \sum_k f_k(x_n, y_n, y_{n-1})$$

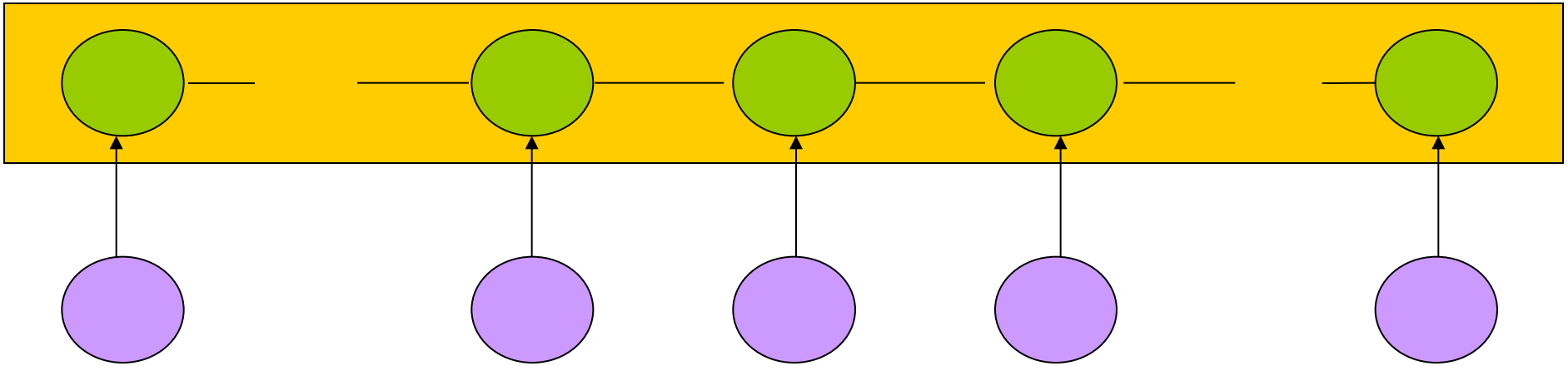
New model

$$\frac{\exp(\sum_k \lambda_k F_k(\vec{x}, \vec{y}))}{Z(\vec{x})}$$

n = 1…N: random variable index

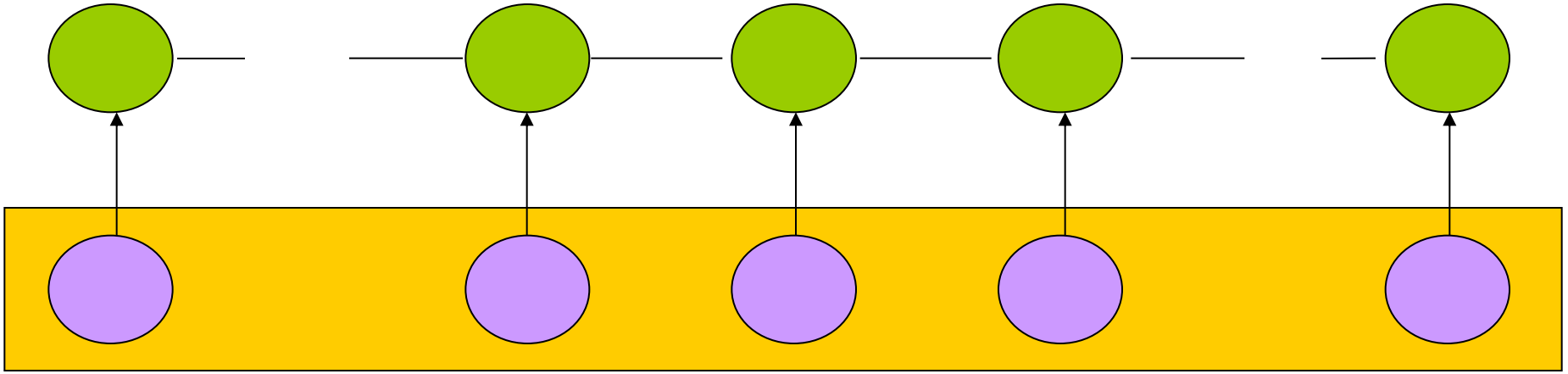
k = 1…K: feature index

What is CRF?



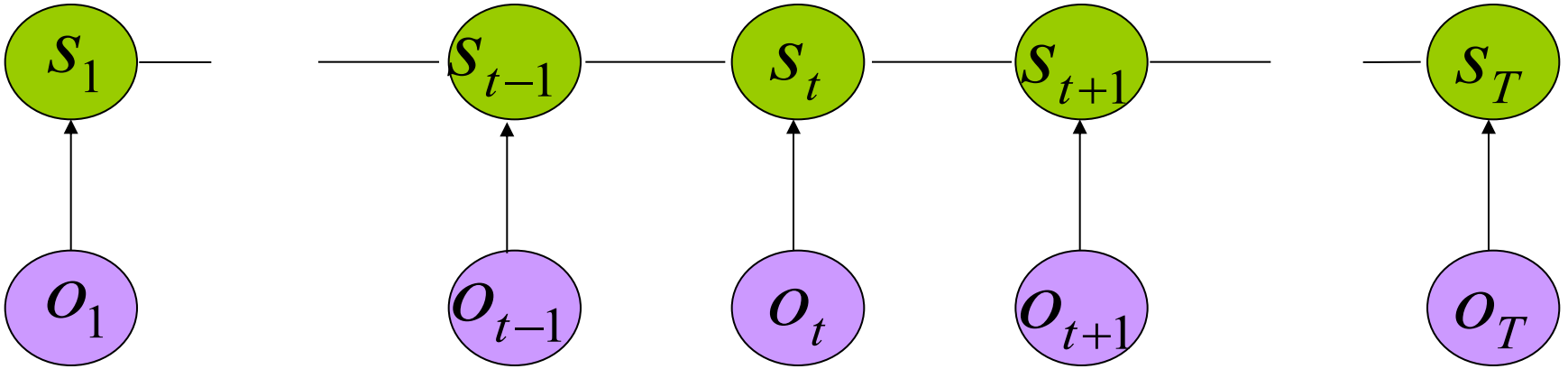
- Green nodes are ***states***
- State depends on neighboring states

What is CRF?



- Purple nodes are ***observations***
- Observations (features of observations) determine states

CRF Formalism



- *s*: {1,2,…,N} are values of states
- *o*: {1,2,…,M } are values of observations

$$P(S \mid O)$$

Back to the CRF

CRF is a Markov Random Fields.

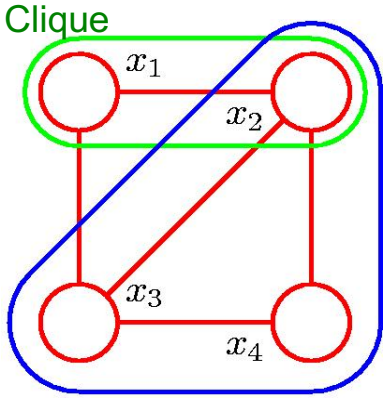
By the Hammersley-Clifford theorem, the probability of a label can be expressed as a Gibbs distribution, so that

$$p(y \mid x, \lambda) = \frac{1}{Z} \exp\left(\sum_k \lambda_k F_k(y, x)\right)$$

$$F_k(y, x) = \sum_{n=1}^{N} f_k(y \mid c_n, x)$$

clique

What is clique?



By only taking consideration of the one node and two nodes cliques, we have

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_l \lambda_l \sum_n t_l(y \mid e_n, x) + \sum_k \mu_k \sum_n s_k(y \mid s_n, x)\right)$$

l : index of features associated to one-node clique

k: index of features associated to two-nodes clique

x, y: random vector

Definition (cont.)

Moreover, let us consider the problem in a first-order chain model, we have

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp\left(\sum_{l} \lambda_l \sum_{n} t_l(y \mid e_n, x) + \sum_{k} \mu_k \sum_{n} s_k(y \mid s_n, x)\right)$$

For simplifying description, let $f_k(y_n, x)$ denote $t_l(y_{n-1}, y_n, x)$ and $s_k(y_n, x)$

$$p(y \mid x, \lambda) = \frac{1}{Z} \exp \sum_{k} \lambda_k F_k(y, x))$$

$$F_k(y, x) = \sum_{n=1}^{N} f_k(y \mid c_n, x)$$

In Labeling

- In labeling, the task is to find the label sequence that has the largest probability

$$\hat{y} = \arg\max_{y} p_{\lambda}(y \mid x) = \arg\max_{y}(\lambda \cdot F(y,x))$$

$$p(y \mid x,\lambda) = \frac{1}{Z}\exp(\sum_{k}\lambda_k F_k(y,x))$$

- Then the key is to estimate the parameter lambda
- Let us first review the optimization formalization

Optimization

- Defining a loss function, that should be convex for avoiding local optimization
- Defining constraints
- Finding a optimization method to solve the loss function
- A formal expression for optimization problem

$$\min_{\theta} f(x)$$

$$s.t. \qquad g_i(x) \geq 0, 0 \leq i \leq k$$

$$h_j(x) = 0, 0 \leq j \leq l$$

Loss Function

Empirical loss vs. structural loss

$$L = \sum_{m=1}^{M} \left| y^{(m)} - f(x^{(m)}, \lambda) \right|$$

$$\min L$$

$$L = \left|\left|\lambda\right|\right| + \sum_{m=1}^{M} \left| y^{(m)} - f(x^{(m)}, \lambda) \right|$$

$$\min L$$

Loss function: Log-likelihood

$$p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_{k} \lambda_k F_k(y, x))$$

$$L(\lambda) = \sum_{m} \left[-\log Z + \sum_{k} \lambda_k F_k(y^{(m)}, x^{(m)}) \right]$$

m: index of data sample

$$L = \sum_{m} \left[\lambda \cdot F(y^{(m)}, x^{(m)}) - \log Z(x^{(m)}) \right] - \frac{\left|\left|\lambda\right|\right|^2}{2\sigma^2} + const$$

IIS Algorithm

First-order numerical optimization

Using Iterative Scaling (GIS, IIS)

- Initialize each λ_k(=0 for example)
- Until convergence
  - Solve $\frac{\delta L}{\delta \lambda_i} = 0$ for each parameter λ_k
  - Update each parameter using λ_k<- $\lambda_k + \Delta\lambda_k$

Parameter estimation

Log-likelihood

$$L(\lambda)=\sum_m\left[-\log Z+\sum_k \lambda_k F_k(y^{(m)},x^{(m)})\right]$$

$$\frac{\delta L_\lambda}{\delta\lambda_k}=\sum_m\left[F_k(y^{(m)},x^{(m)})-\frac{(Z(x^{(m)}))'}{Z(x^{(m)})}\right]$$

$$Z(x^{(m)})=\sum_y \exp\lambda\cdot F(y,x^{(m)})$$

$$\frac{(Z(x^{(m)}))'}{Z(x^{(m)})}=\frac{\sum_y\left(\exp(\lambda\cdot F(y,x^{(m)}))*F_k(y,x^{(m)})\right)}{\sum_y \exp\lambda\cdot F(y,x^{(m)})}$$

Differentiating the log-likelihood with respect to parameter λ_k ⇒

$$=\sum_y\left(\frac{\exp(\lambda\cdot F(y,x^{(m)})}{\sum_y \exp\lambda\cdot F(y,x^{(m)})})*F_k(y,x^{(m)})\right)$$

$$=\sum_y\left(p(y\mid x^{(m)})*F_k(y,x^{(m)})\right)$$

$$=E_{p(Y|x^{(m)})}F_k(Y,x^{(m)})$$

$$\frac{\delta L}{\delta\lambda_k}=E_{p(Y,X)}[F_k(Y,X)]-\sum_m E_{p(Y|x^{(m)},\lambda)}[F_k(Y,x^{(m)})] \Leftarrow$$

By adding the model penalty, it can be rewritten as

$$\frac{\delta L}{\delta\lambda_k}=E_{p(Y,X)}[F_k(Y,X)]-\sum_m E_{p(Y|x^{(k)},\lambda)}[F_k(Y,x^{(m)})]-\frac{\lambda}{\sigma^2}$$

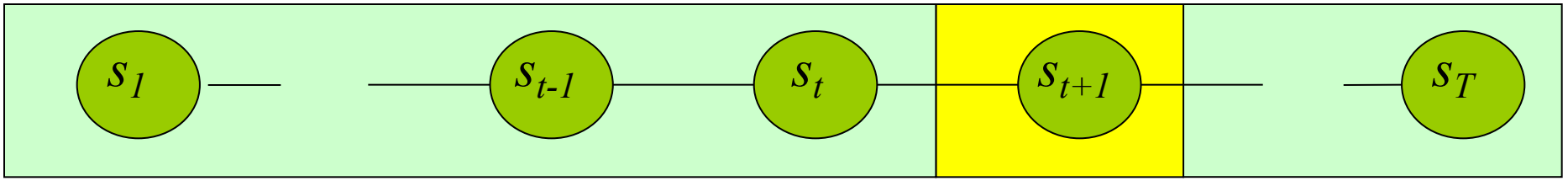
Solve the Optimization

$$L(\lambda) = \sum_m \left[-\log Z + \sum_k \lambda_k F_k(y^{(m)}, x^{(m)}) \right]$$

$$\frac{\delta L}{\delta \lambda_k} = E_{p(Y,X)}[F_k(Y,X)] - \sum_m E_{p(Y|x^{(m)},\lambda)}[F_k(Y,x^{(m)})]$$

- $E_{p(y,x)}F_k(y,x)$ can be calculated easily
- $E_{p(y|x)}F_k(y,x)$ can be calculated by making use of a forward-backward algorithm
- Z can be estimated in the forward-backward algorithm

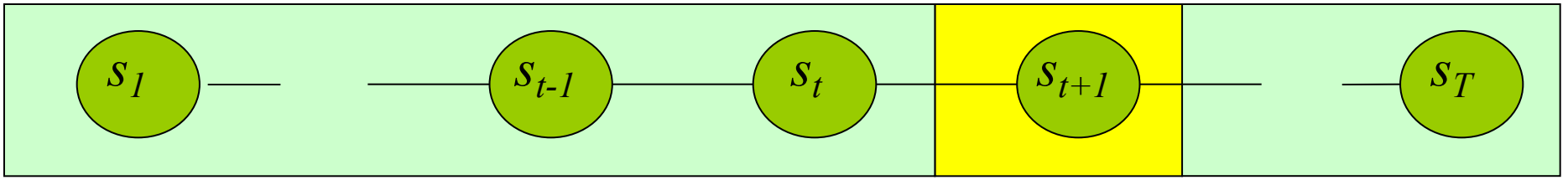
Inference



$$p(\mathbf{s}) = \frac{1}{Z} \prod_{i=1,\ldots,T-1} \varphi(s_i, s_{i+1})$$

$$p(s_{t+1}) = \sum_{s_1} \cdots \sum_{s_t} \sum_{s_{t+2}} \cdots \sum_{s_T} p(\mathbf{s})$$

Inference



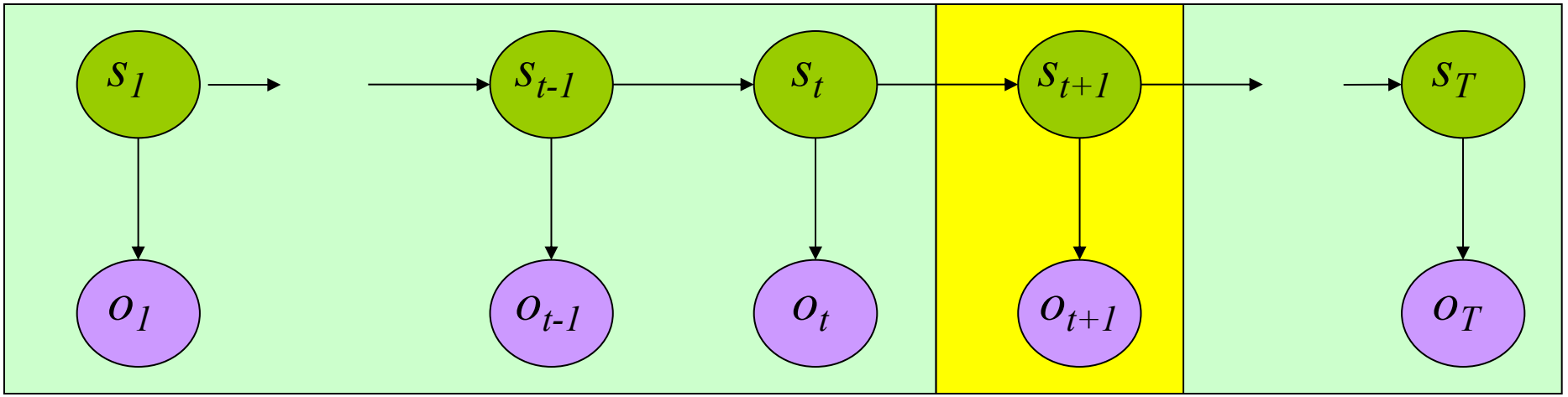
α_t β_t

$$p(s_{t+1}) = \frac{1}{Z}\left[\sum_{s_1}\varphi(s_1,s_2)\cdots\sum_{s_t}\varphi(s_t,s_{t+1})\right]\left[\sum_{s_{t+2}}\varphi(s_{t+1},s_{t+2})\cdots\sum_{s_T}\varphi(s_{T-1},s_T)\right]$$

α_t

β_t

Forward Backward Algorithm

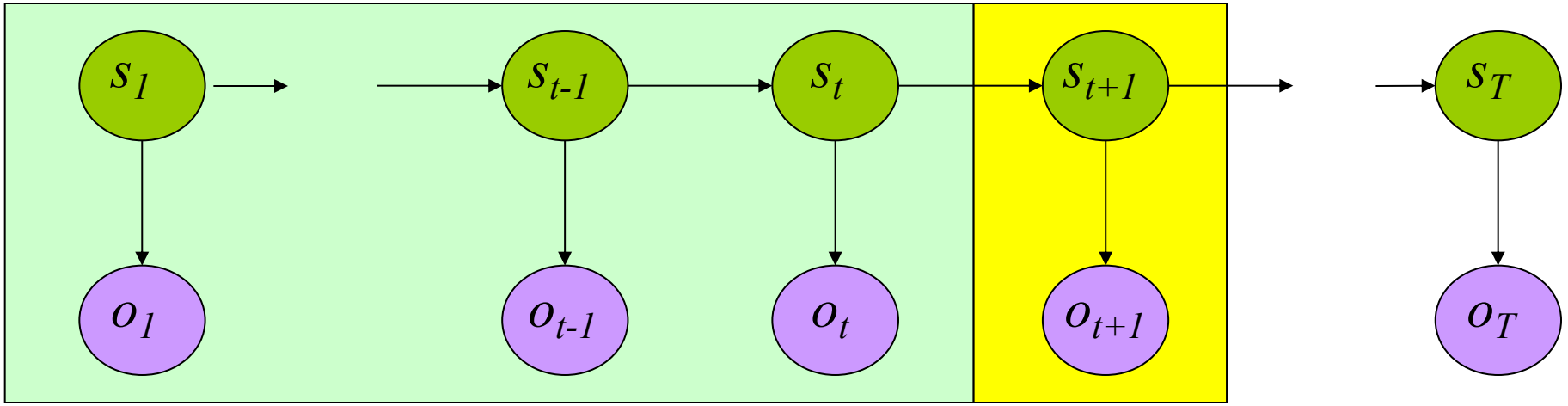


- An efficient algorithm using *dynamic programming*.

$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i \mid \lambda)$$

$$\beta_t(i) = P(o_t \ldots o_T \mid s_t = i, \lambda)$$

Forward Probability



$$\alpha_t(i) = P(o_1 \ldots o_t, s_t = i \mid \mu)$$

$$\alpha_{t+1}(j) = \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

$$\alpha_1(i) = \pi_i b_{io_1}$$

π are initial state probabilities

Forward Probability

$$\alpha_{t+1}(j)$$

$$= P(o_1 \ldots o_{t+1}, s_{t+1} = j)$$

$$= P(o_1 \ldots o_{t+1} \mid s_{t+1} = j) P(s_{t+1} = j)$$

$$= P(o_1 \ldots o_t \mid s_{t+1} = j) P(o_{t+1} \mid s_{t+1} = j) P(s_{t+1} = j)$$

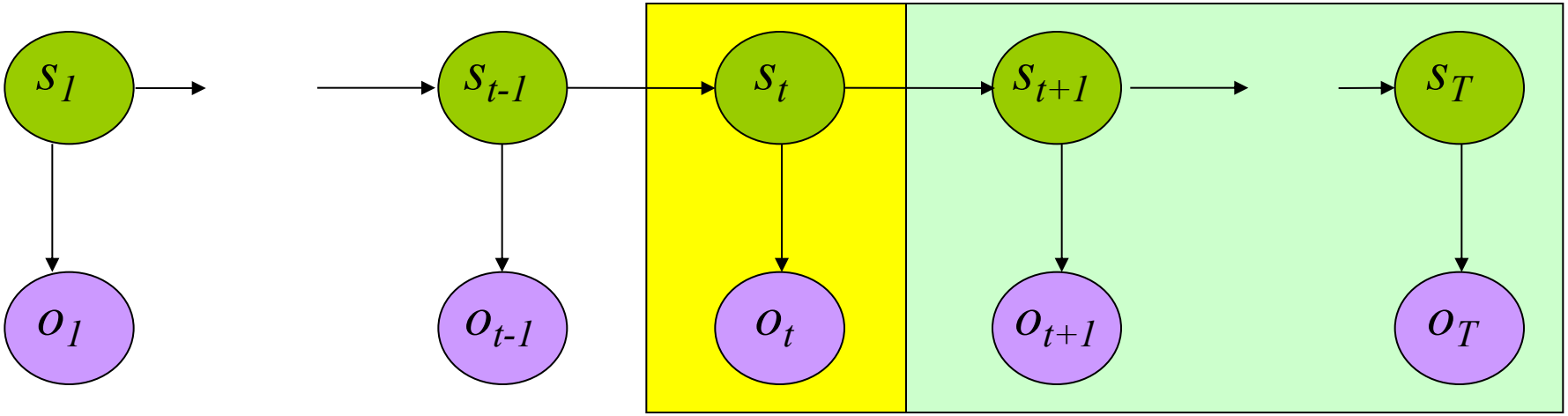
$$= P(o_1 \ldots o_t, s_{t+1} = j) P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(o_1 \ldots o_t, s_t = i, s_{t+1} = j) P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(o_1 \ldots o_t, s_t = i) P(s_{t+1} = j \mid s_t = i) P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

Backward Probability



$$\beta_t(i) = P(o_t \ldots o_T \mid s_t = i)$$

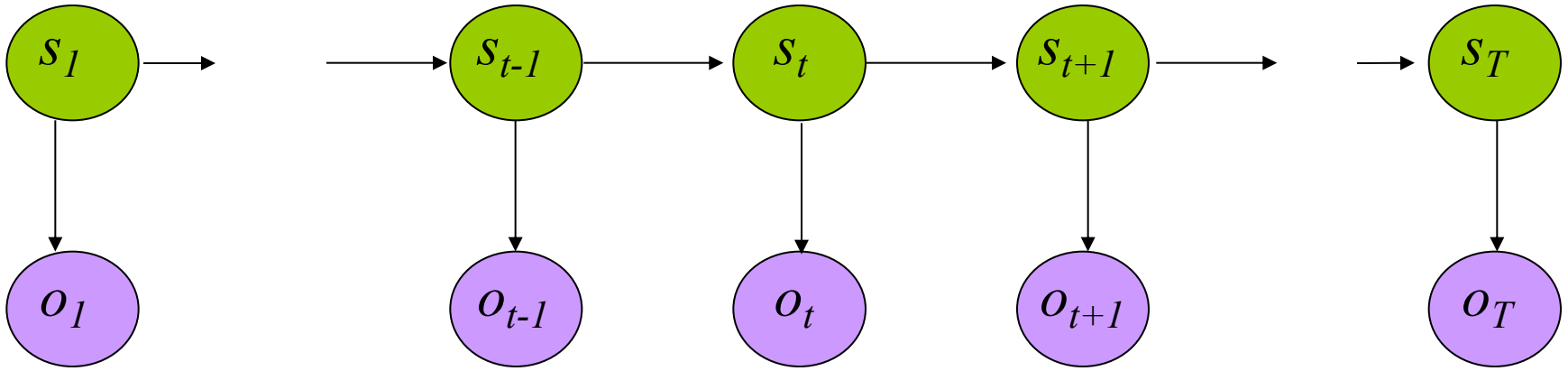
$$\beta_{T+1}(i) = 1$$

$$\beta_t(i) = \sum_{j=1\ldots N} a_{ij} b_{io_t} \beta_{t+1}(j)$$

Inference Algorithm on a Chain

- To compute local marginals:
  - Compute and store all forward messages, $\alpha_t(i)$.
  - Compute and store all backward messages, $\beta_t(i)$.
  - Compute Z at any node s_i
  - Compute marginal probability for all variables required $p(y)$.

Marginal Probability



$$p_t(i,j) = \frac{\alpha_{t-1}(i) a_{ij} b_{jo_t} \beta_t(j)}{\sum_{k=1...N} \alpha_t(k)\beta_t(k)}$$

$$p_t(i) = \frac{\alpha_t(i)\beta_t(j)}{\sum_{k=1...N} \alpha_t(k)\beta_t(k)}$$

Calculating the Expectation

- First we define the transition matrix of y for position x as

$$M_n[y_{n-1}, y_n] = \exp \lambda \cdot f(y_{n-1}, y_n, x)$$

$$E_{p(Y|x^{(m)})}\left[F_k(Y, x^{(m)})\right] = \sum_y p(y \mid x^{(m)}) F_k(y, x)$$

$$= \sum_{n=1}^{N} \sum_{y_{n-1}, y_n} p(y_{n-1}, y_n \mid x^{(m)}) f_k(y_{n-1}, y_n, x^{(k)}) + \sum_{n=1}^{N} \sum_{y_n} p(y_n \mid x^{(m)}) f_k(y_n, x^{(m)})$$

$$p(y_{n-1}, y_n \mid x^{(k)}) = \frac{\alpha_{n-1}(M_n * V_n)\beta_n^T}{Z(x)}$$

$$p(y_n \mid x^{(k)}) = \frac{\alpha_n \beta_n^T}{Z(x)}$$

$$Z(x) = \left[\prod_{n=1}^{N+1} M_n(x)\right] = \alpha_N \cdot 1^T$$

$$\alpha_n = \begin{cases} \alpha_n M_n & 0 < n \le N \\ 1 & n = 0 \end{cases}$$

$$\beta_n^T = \begin{cases} M_{n+1}\beta_{n+1}^T & 1 \le n < N \\ 1 & n = N \end{cases}$$

All state features at position *n*

IIS Algorithm

First-order numerical optimization

Using Iterative Scaling (GIS, IIS)

- Initialize each λ_k(=0 for example)
- Until convergence
  - Solve $\frac{\delta L}{\delta \lambda_k} = 0$ for each parameter λ_k
  - Update each parameter using λ_k<- $\lambda_k + \Delta\lambda_k$

Low efficient!!

Second-order numerical optimization

Using newton optimization technique for the parameter estimation

$$\lambda^{(k+1)} = \lambda^{(k)} + (\frac{\partial^2 L}{\partial \lambda^2})^{-1} \frac{\partial L}{\partial \lambda}$$

Drawbacks: parameter value initialization
And compute the second order (i.e. hesse matrix), that is difficult

Solutions:

- Conjugate-gradient (CG) (Shewchuk, 1994)
- Limited-memory quasi-Newton (L-BFGS) (Nocedal and Wright, 1999)
- Voted Perceptron (Colloins 2002)

Summary of CRFs

Model

•Lafferty, 2001

Applications

•Efficient training (Wallach, 2003)
•Training via. Gradient Tree Boosting (Dietterich, 2004)
•Bayesian Conditional Random Fields (Qi, 2005)
•Name entity (McCallum, 2003)
•Shallow parsing (Sha, 2003)
•Table extraction (Pinto, 2003)
•Signature extraction (Kristjansson, 2004)
•Accurate Information Extraction from Research Papers (Peng, 2004)
•Object Recognition (Quattoni, 2004)
•Identify Biomedical Named Entities (Tsai, 2005)
•…

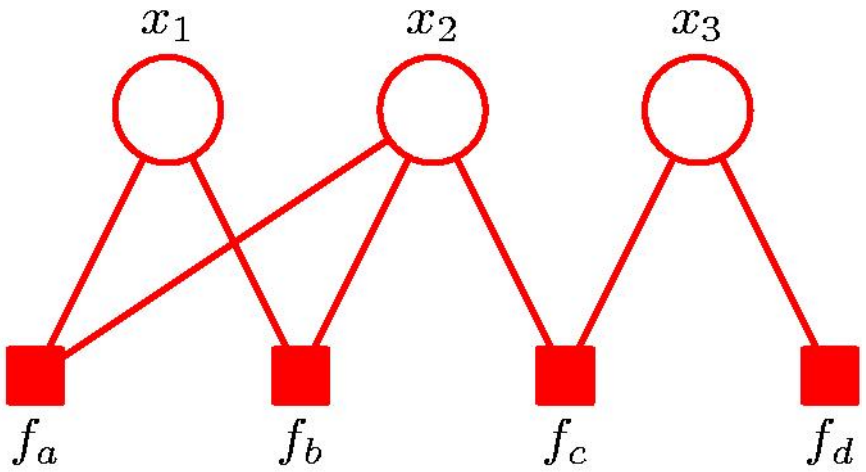
Limitation

•Huge computational cost in parameter estimation

Outline

- Introduction
- Directed Graphical Models
- Undirected Graphical Models
- Factor Graph

Factor Graphs

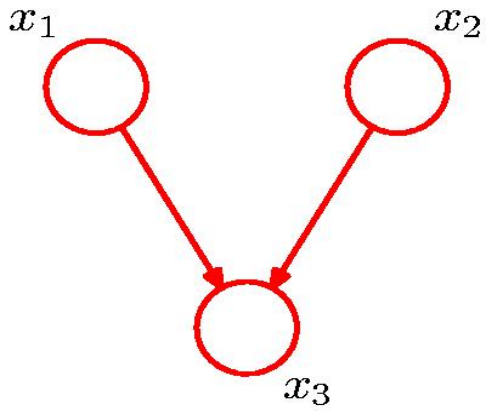


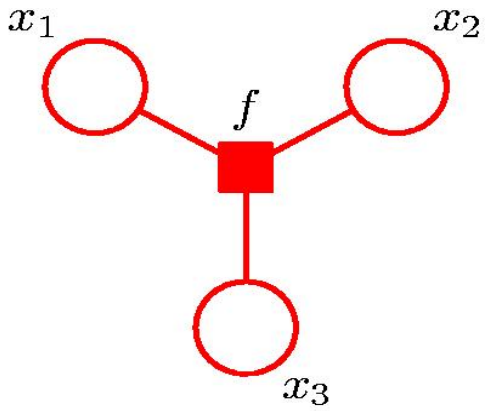
$$p(\mathbf{x}) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3)$$

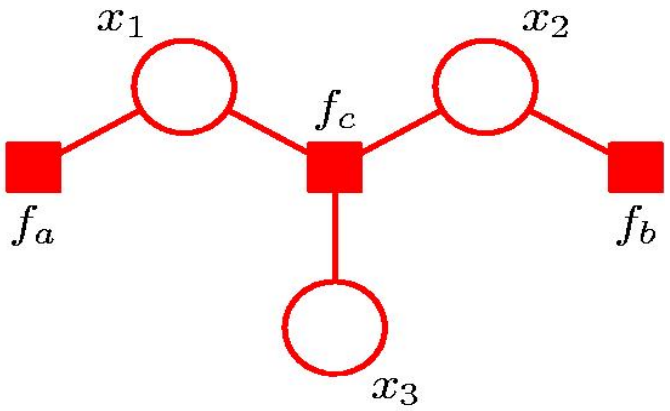
$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s)$$

A number of the following pages are from Bishop's slides

Factor Graphs from Directed Graphs







$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3|x_1,x_2)$$

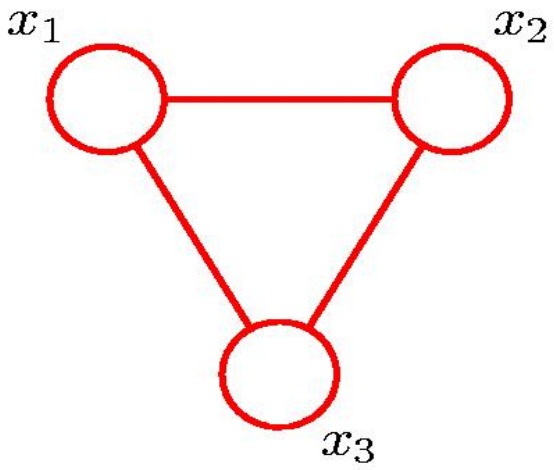
$$f(x_1,x_2,x_3) = p(x_1)p(x_2)p(3|x_1,x_2)$$

$$f_a(x_1) = p(x_1)$$

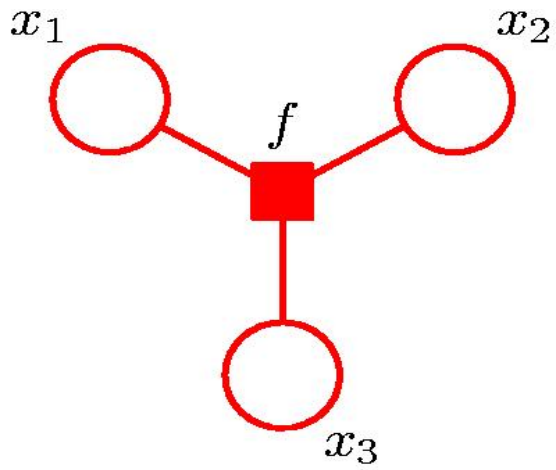
$$f_b(x_2) = p(x_2)$$

$$f_c(x_1,x_2,x_3) = p(x_3|x_1,x_2)$$

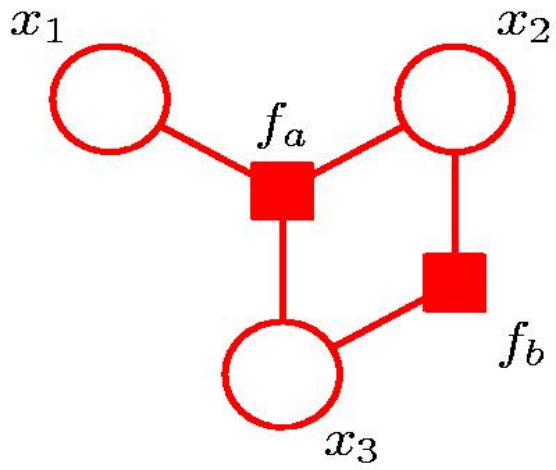
Factor Graphs from Undirected Graphs



$$\psi(x_1, x_2, x_3)$$

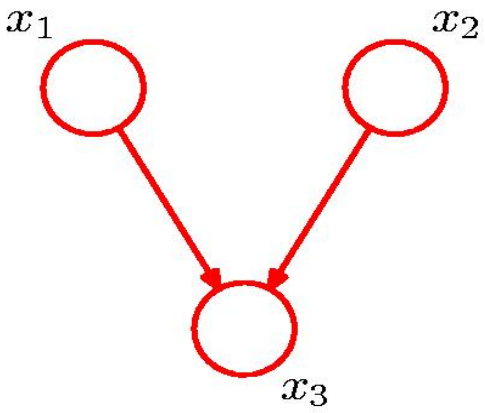


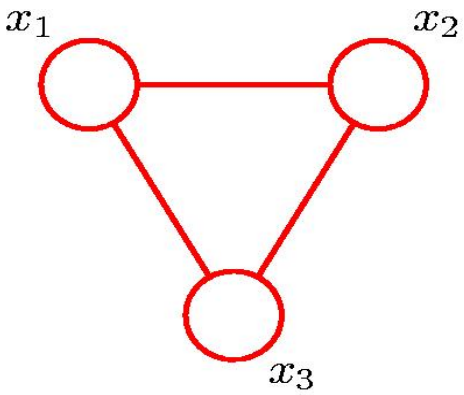
$$f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$$



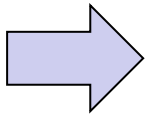
$$f_a(x_1, x_2, x_3) f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$$

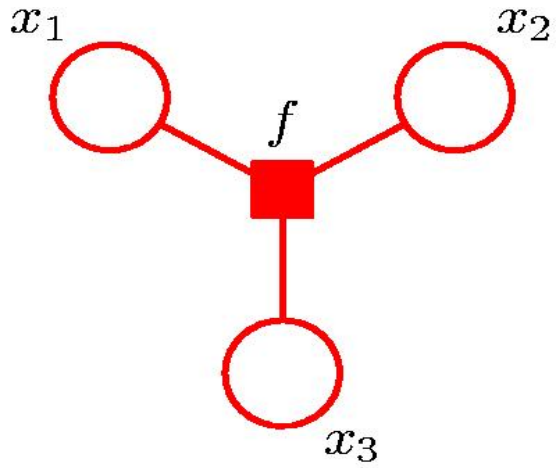
Factorization





$\psi(x_1, x_2, x_3)$





$$f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$$

Example

Inferring Social Ties across Heterogeneous Networks

(WSDM'12, ACM TOIS'16)

Jie Tang*, Tiancheng Lou*, and Jon Kleinberg+

*Tsinghua University

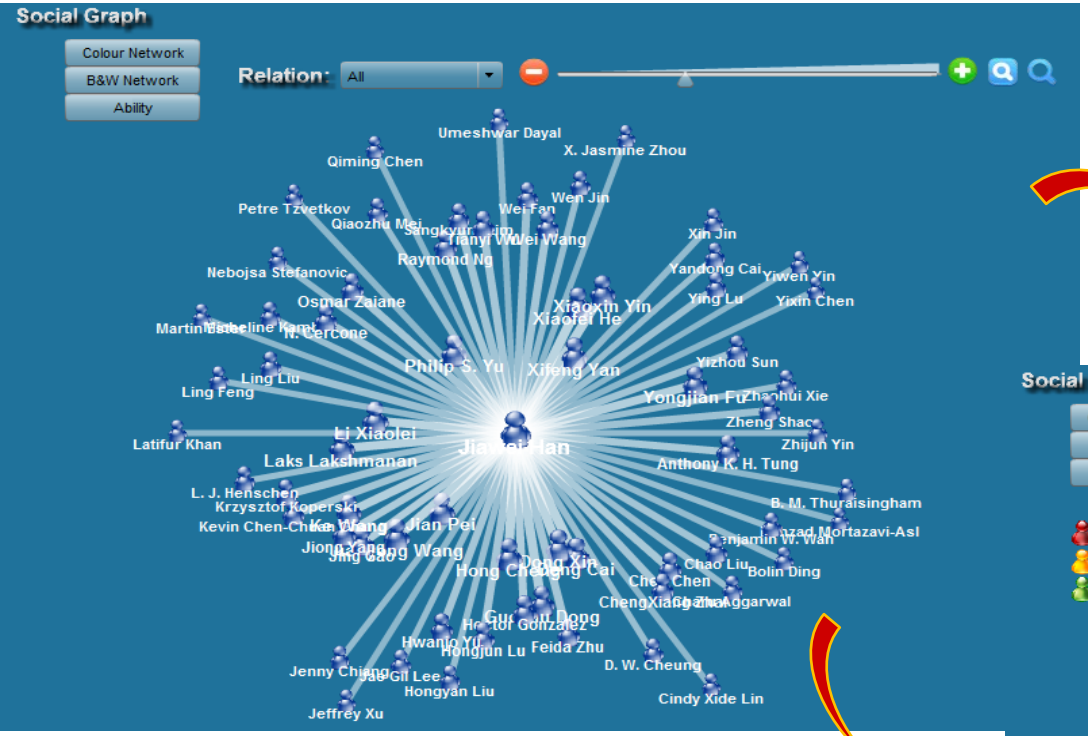
+Cornell University

Real social networks are complex...

- Different social ties have different influence on people
  - Close friends vs. Acquaintances
  - Colleagues vs. Family members
- However, existing networks (e.g., Facebook and Twitter) are trying to lump everyone into one big network
  - FB tries to solve this problem via lists/groups
  - However…
- Google+



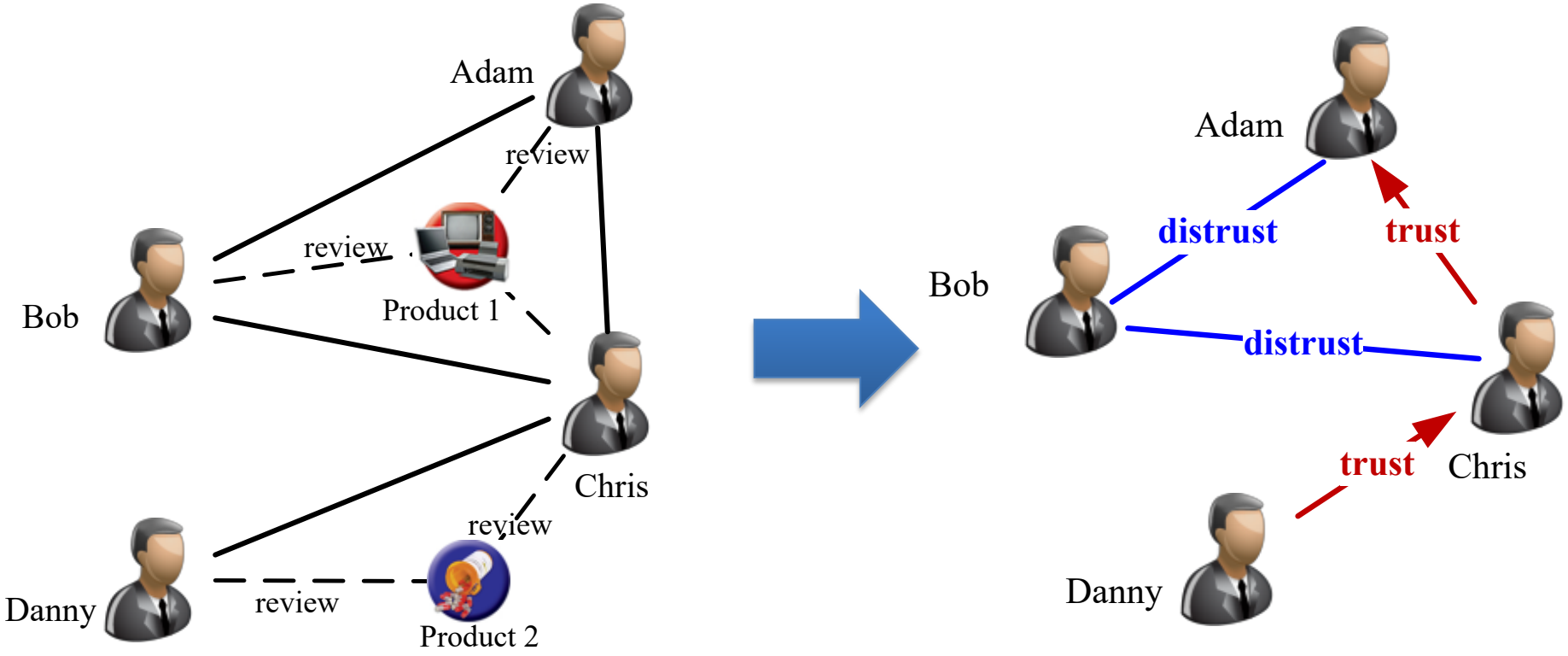
Example 1. Advisor-advisee relationship



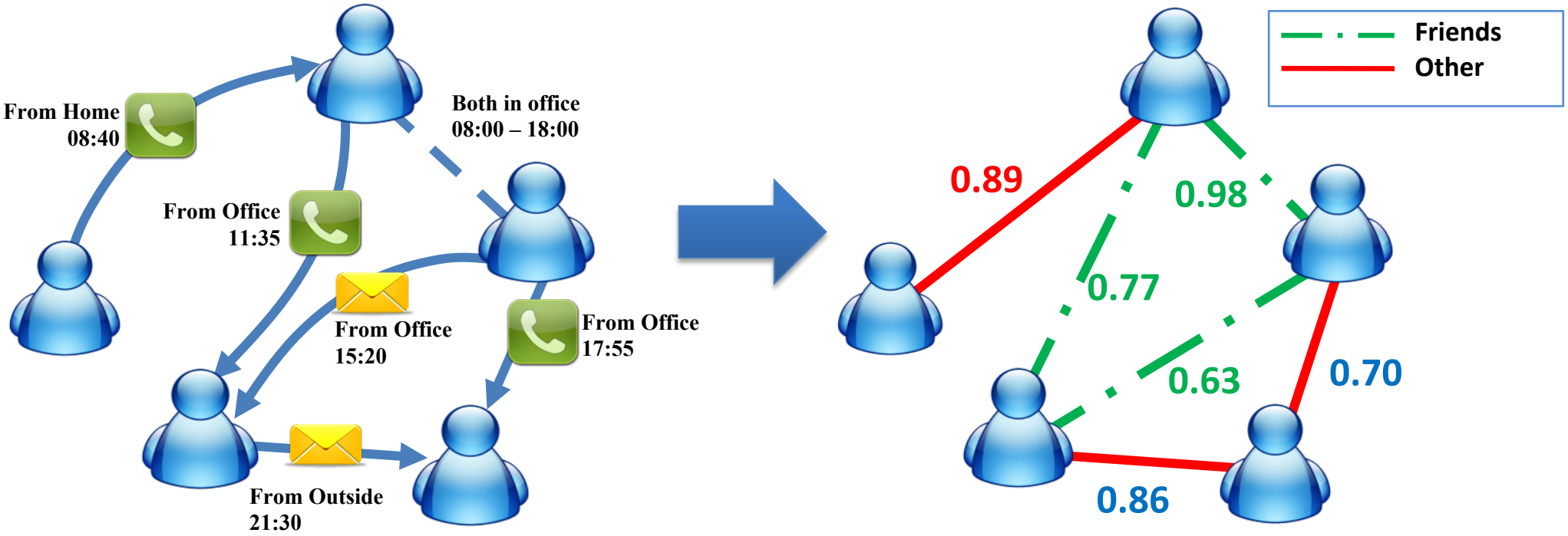
Arnetminer



Example 2. Trustful relationship

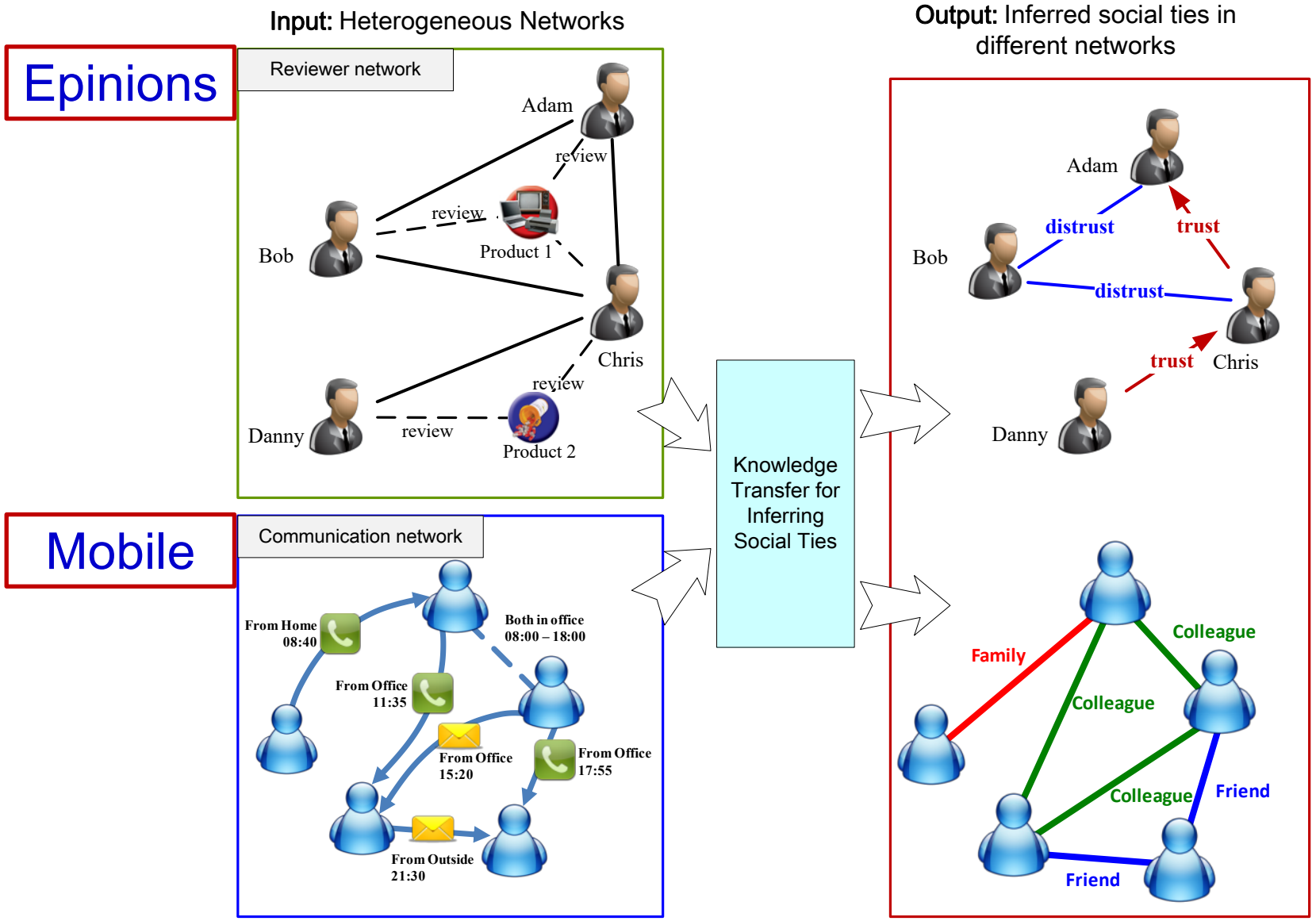


Example 3: Friendship in mobile network



Mobile

Inferring Social Ties Across Networks

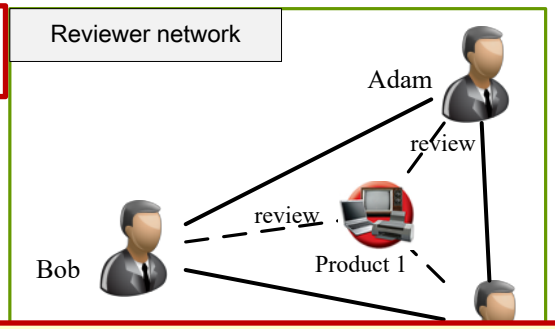


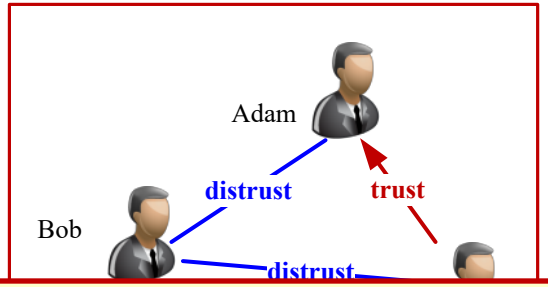
Inferring Social Ties Across Networks

Input: Heterogeneous Networks

Output: Inferred social ties in different networks

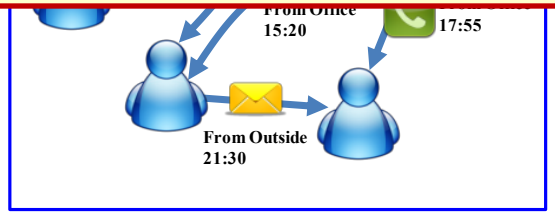
Epinions

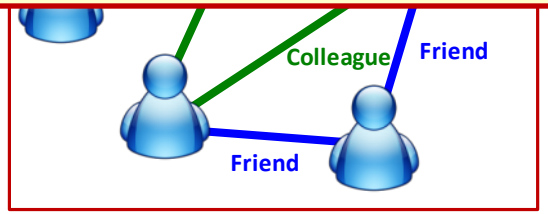




Questions:

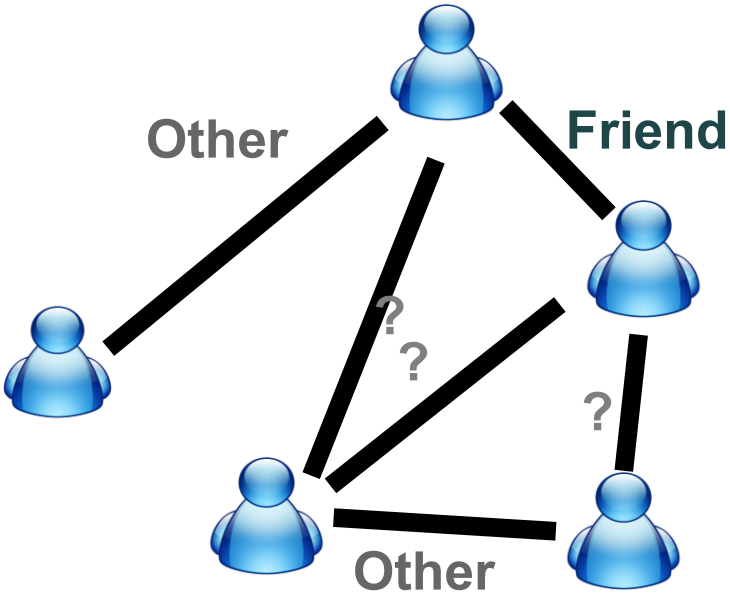
- What are the fundamental forces behind?
- A generalized framework for inferring social ties?
- How to connect the different networks?





Problem Formulation in a Single Network

Input: $G=(V,E^L,E^U,R^L,W)$



V: Set of Users

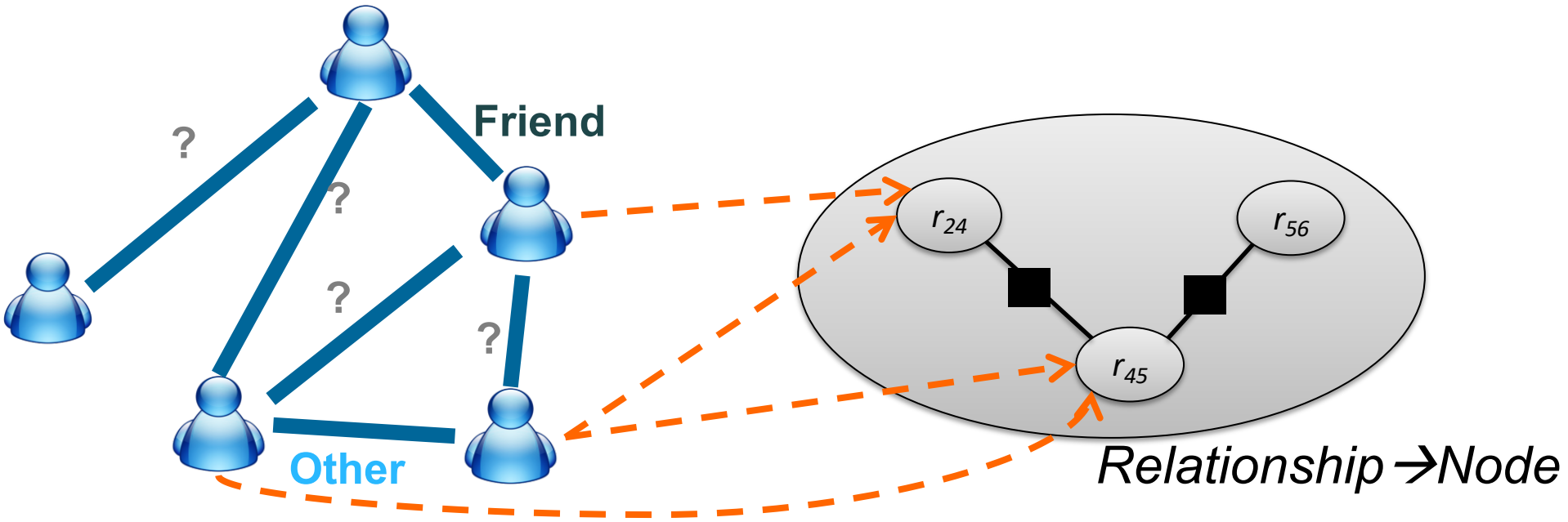
E^L, R^L: Labeled relationships

E^U: Unlabeled relationships

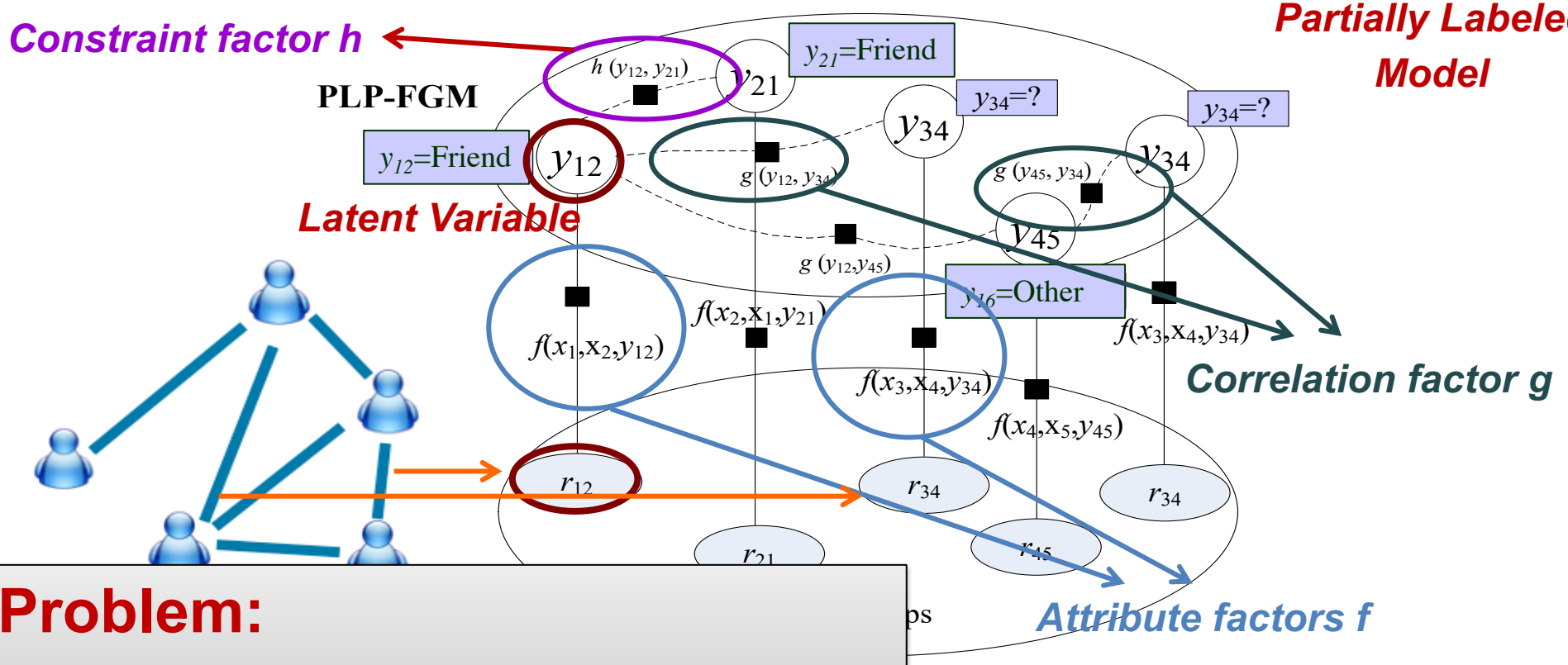
Input: $G=(V,E^L,E^U,R^L,W)$

Output: $f: G \rightarrow R$

Basic Idea



Partially Labeled Pairwise Factor Graph Model (PLP-FGM)



Problem:

For each relationship, identify which type has the highest probability?

A makes call to B immediately after the call to C.

Solutions(con't)

- Different ways to instantiate factors
  - We use exponential-linear functions
    - Attribute Factor:

$$f(y_i, \mathbf{x}_i) = \frac{1}{Z_\lambda} \exp\{\lambda^T \mathbf{\Phi}(y_i, \mathbf{x}_i)\}$$

    - Correlation / Constraint Factor:

$$g(y_i, G(y_i)) = \frac{1}{Z_\alpha} \exp\{\sum_{y_j \in G(y_i)} \alpha^T \mathbf{g}(y_i, y_j)\}$$

$$h(y_i, H(y_i)) = \frac{1}{Z_\beta} \exp\{\sum_{y_j \in H(y_i)} \beta^T \mathbf{h}(y_i, y_j)\}$$

  - Log-Likelihood of labeled Data:

$$\mathcal{O}(\theta) = \log \sum_{Y|Y^L} \exp\{\theta^T \mathbf{S}\} - \log \sum_{Y} \exp\{\theta^T \mathbf{S}\}$$

Parameters to estimate $\theta = [\lambda, \alpha, \beta], s = [\Phi^T, g^T, h^T]^T$

Learning Algorithm

- Maximize the log-likelihood of labeled relationships

Input: learning rate η
Output: learned parameters θ

Initialize θ;
repeat
  Calculate $\mathbb{E}_{p_\theta(Y|Y^L,G)}\mathbf{S}$ using LBP ;
  Calculate $\mathbb{E}_{p_\theta(Y|G)}\mathbf{S}$ using LBP ;
  Calculate the gradient of θ according to Eq. 7:

$$\nabla_\theta = \mathbb{E}_{p_\theta(Y|Y^L,G)}\mathbf{S} - \mathbb{E}_{p_\theta(Y|G)}\mathbf{S}$$

  Update parameter θ with the learning rate η:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \cdot \nabla_\theta$$

until *Convergence*;

Algorithm 1: Learning PLP-FGM.

Expectation Computing

Loopy Belief Propagation

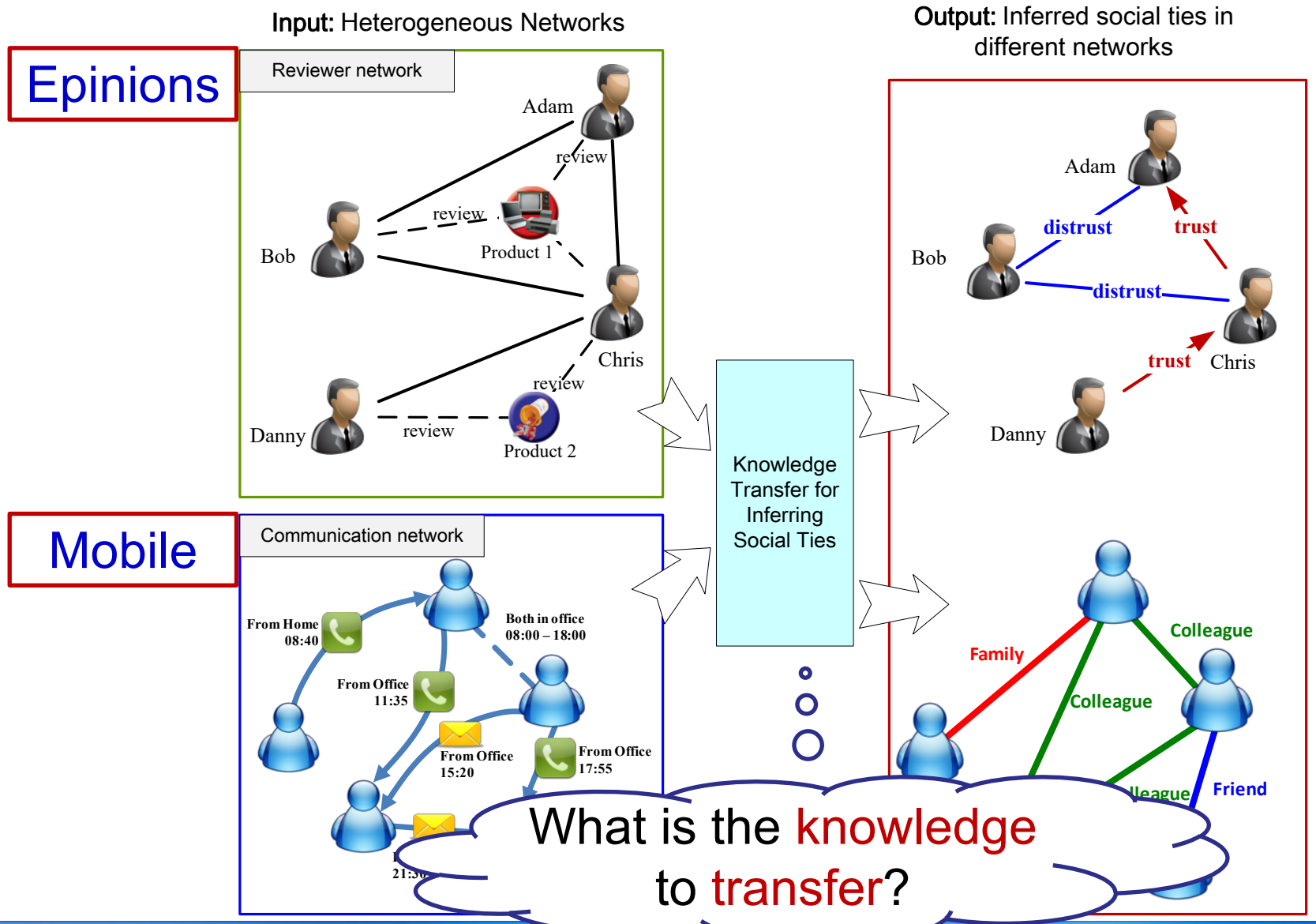
Gradient Ascent Method

Still Challenges?

Questions:

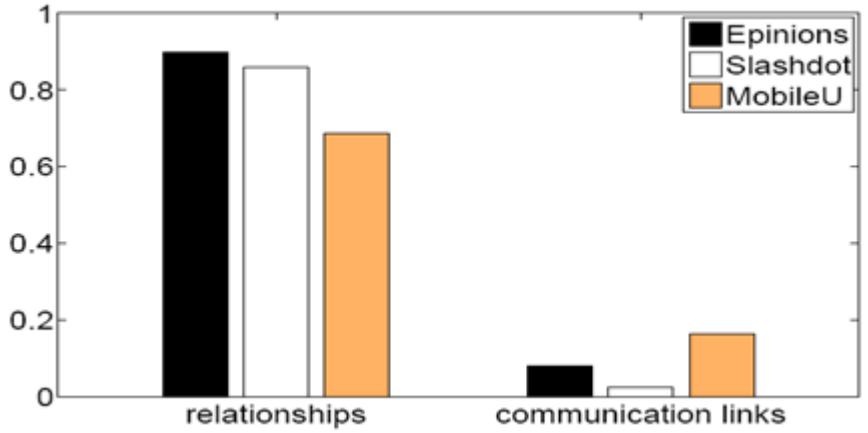
- How to obtain sufficiently training data?
- Can we leverage knowledge from other network?

Inferring Social Ties Across Networks



Social Theories

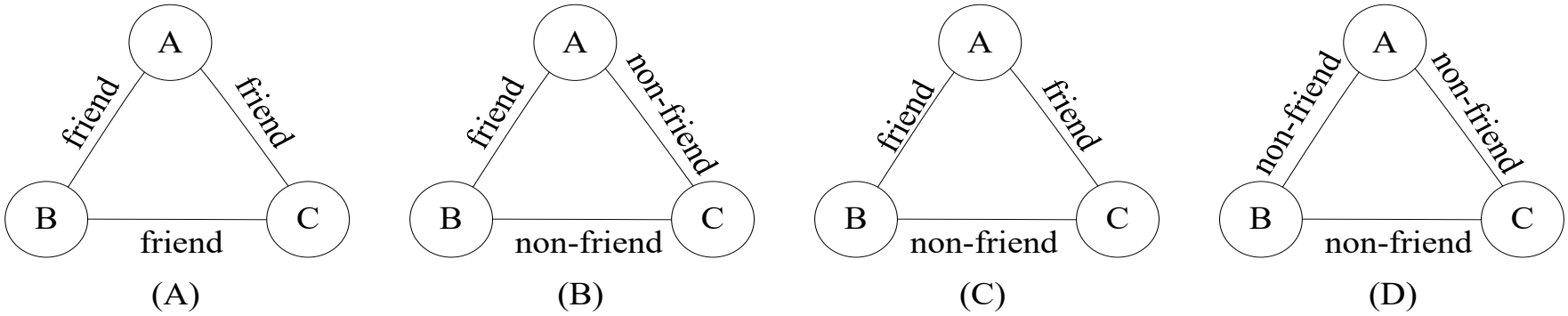
- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory



Observations:

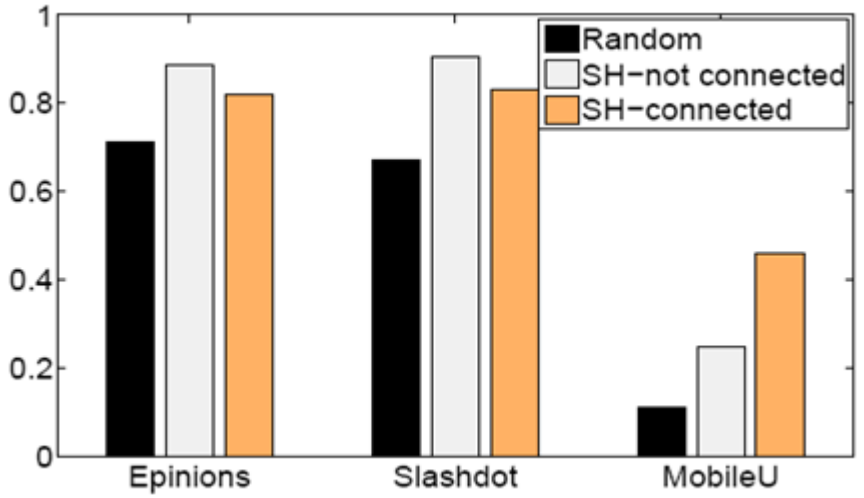
(1) The underlying networks are unbalanced;

(2) While the friendship networks are balanced.

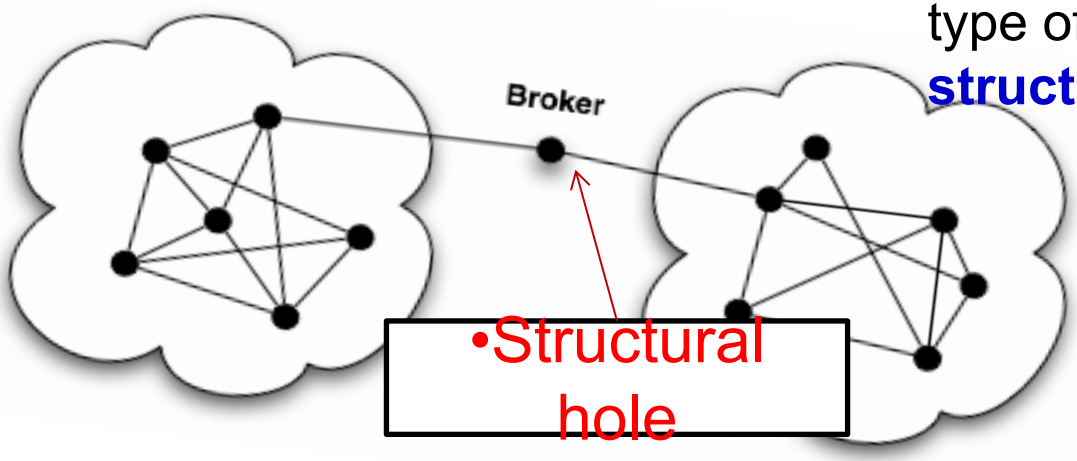


Social Theories—Structural hole

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory

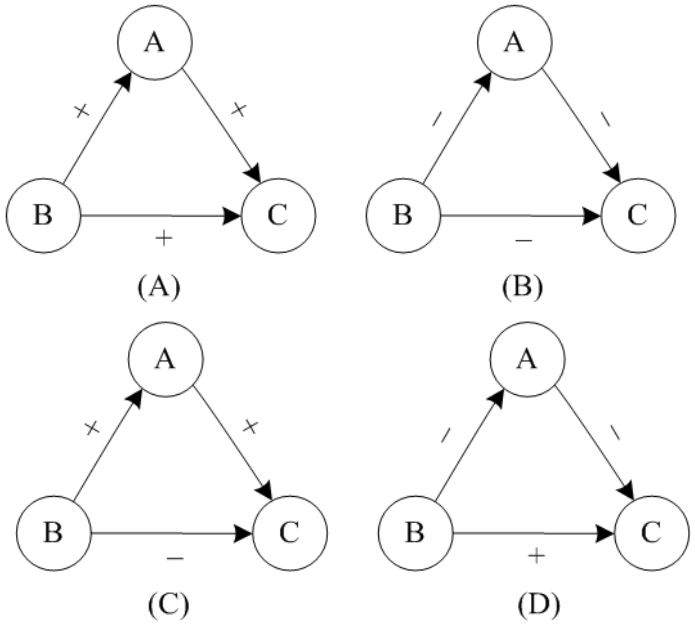


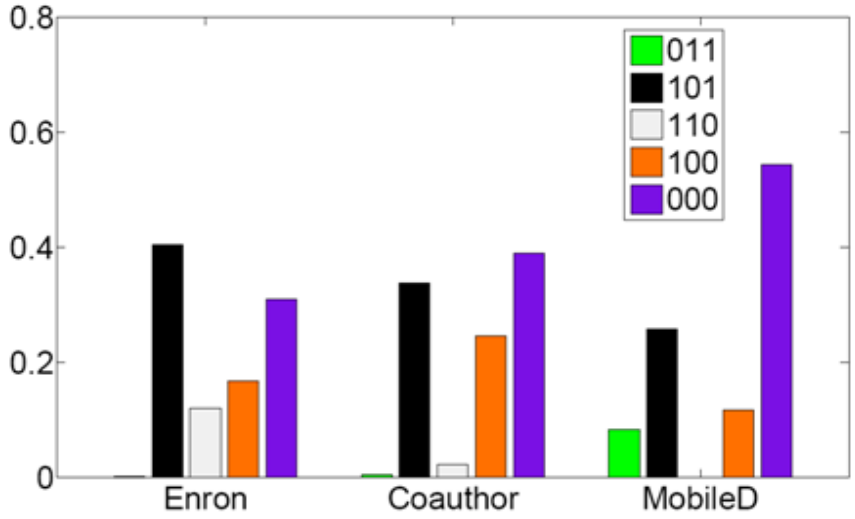
Observations: Users are more likely (+25-150% higher than change) to have the same type of relationship with C if C **spans structural holes**



Social Theories—Social status

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory



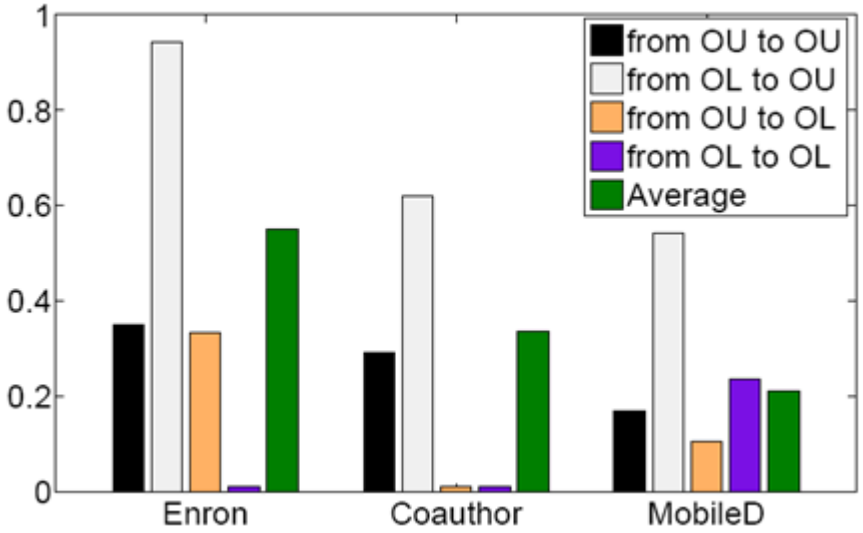


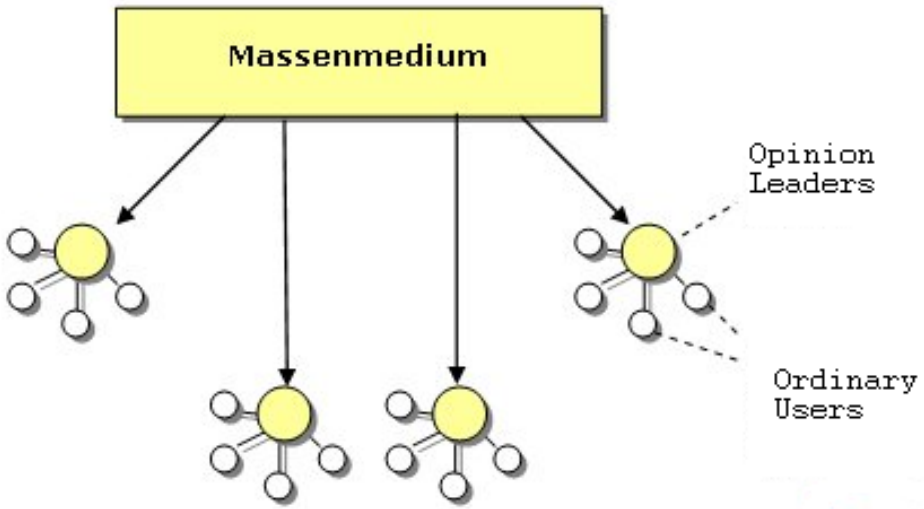
Observations: 99% of triads in the networks satisfy the social status theory

•**Note:** Given a triad (A,B,C), let us use 1 to denote the advisor-advisee relationship and 0 colleague relationship. Thus the number 011 to denote A and B are colleagues, B is C's advisor and A is C's advisor.

Social Theories—Two-step-flow

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory





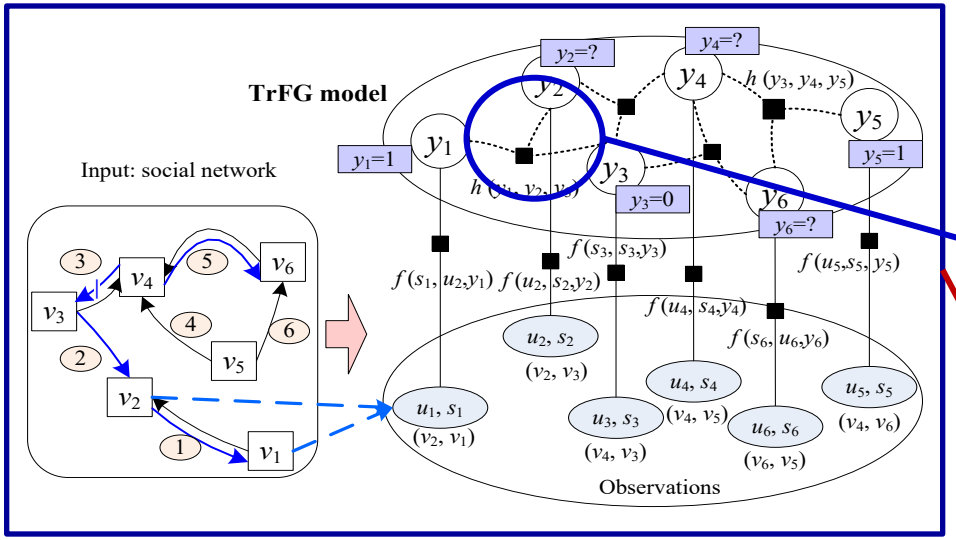
OL : Opinion leader;

OU : Ordinary user.

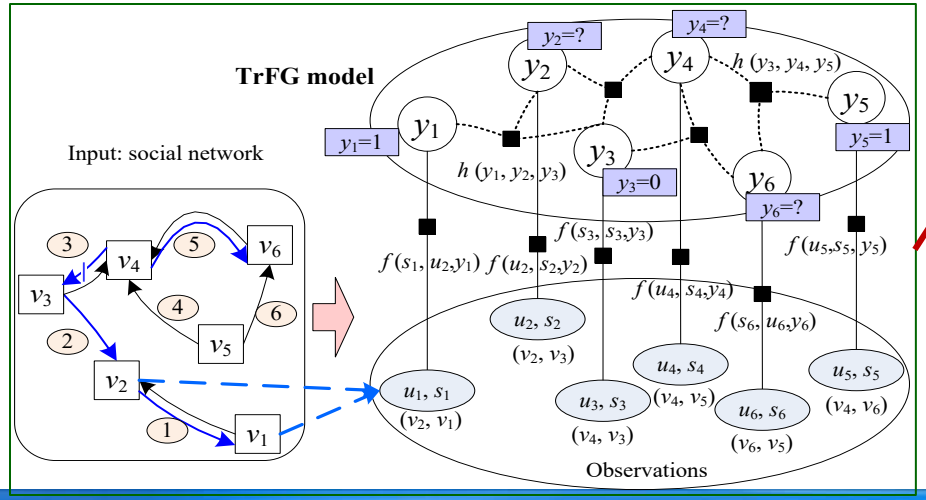
Observations: Opinion leaders are more likely (+71%-84% higher than chance) to have a higher social-status than ordinary users.

Transfer Factor Graph Model

Coauthor network



mobile



Triad-based factor

Bridge via social theories

Mathematical Formulation

Features defined in source network

Features defined in target network

$$\mathcal{O}(\alpha, \beta, \mu) = \mathcal{O}_S(\alpha, \mu) + \mathcal{O}_T(\beta, \mu)$$

$$= \sum_{i=1}^{|V_S|} \sum_{j=1}^{d} \alpha_j g_j(x_{ij}^S, y_i^S) + \sum_{i=1}^{|V_T|} \sum_{j=1}^{d'} \beta_j g'_j(x_{ij}^T, y_i^T)$$

$$+ \sum_k \mu_k \Big(\sum_{c \in G_S} h_k(Y_c^S) + \sum_{c \in G_T} h_k(Y_c^T) \Big)$$

$$- \log Z$$

Triad-based features shared across networks

Data Sets

- **Epinions** a network of product reviewers: 131,828 nodes (users) and 841,372 edges
  - trust relationships between users
- **Slashdot**: 82,144 users and 59,202 edges
  - "friend" relationships between users
- **Mobile**: 107 mobile users and 5,436 edges
  - to infer friendships between users

Undirected network

- **Coauthor**: 815,946 authors and 2,792,833 coauthor relationships
  - to infer advisor-advisee relationships between coauthors
- **Enron**: 151 Enron employees and 3572 edges
  - to infer manager-subordinate relationships between users

Directed network

Results – undirected networks

- **SVM** and **CRF** are two baseline methods
- **PFG** is the proposed partially-labeled factor graph model
- **TranFG** is the proposed transfer–based factor graph model.

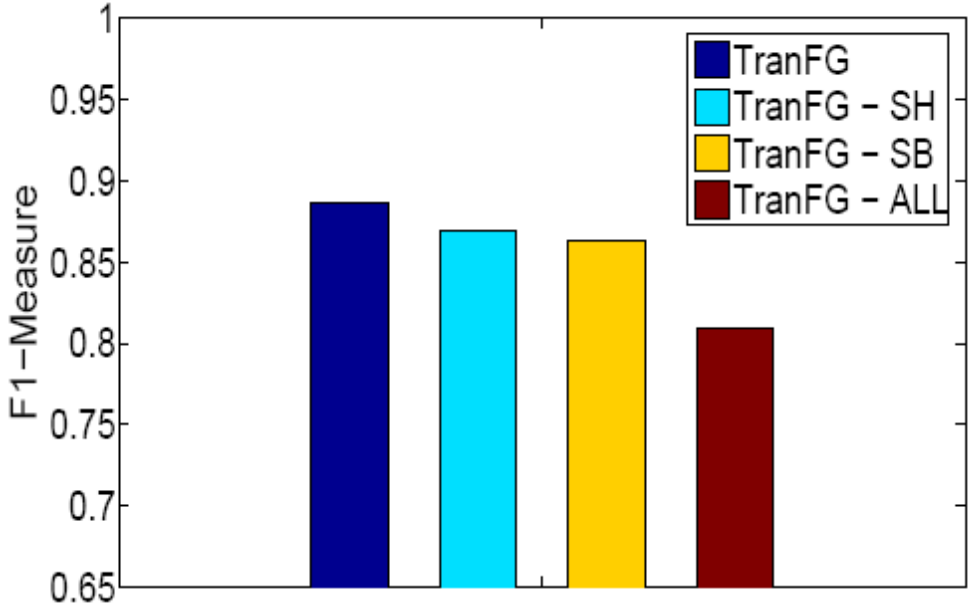
Data Set	Method	Prec.	Rec.	F1-score
Epinions (S) to Slashdot (T) (40%)	SVM	0.7157	**0.9733**	0.8249
	CRF	0.8919	0.6710	0.7658
	PFG	0.9300	0.6436	0.7607
	TranFG	**0.9414**	0.9446	**0.9430**
Slashdot (S) to Epinions (T) (40%)	SVM	0.9132	**0.9925**	0.9512
	CRF	0.8923	0.9911	0.9393
	PFG	**0.9954**	0.9787	**0.9870**
	TranFG	**0.9954**	0.9787	**0.9870**
Epinions (S) to Mobile (T) (40%)	SVM	0.8983	0.5955	0.7162
	CRF	0.9455	0.5417	0.6887
	PFG	**1.0000**	0.5924	0.7440
	TranFG	0.8239	**0.8344**	**0.8291**
Slashdot (S) to Mobile (T) (40%)	SVM	0.8983	0.5955	0.7162
	CRF	0.9455	0.5417	0.6887
	PFG	**1.0000**	0.5924	0.7440
	TranFG	0.7258	**0.8599**	**0.7872**

Results – directed networks

- **SVM** and **CRF** are two baseline methods
- **PFG** is the proposed partially-labeled factor graph model
- **TranFG** is the proposed transfer–based factor graph model.

Data Set	Method	Prec.	Rec.	F1-score
Coauthor (S) to Enron (T) (40%)	SVM	0.9524	0.5556	0.7018
	CRF	0.9565	0.5366	0.6875
	PFG	**0.9730**	0.6545	0.7826
	TranFG	0.9556	**0.7818**	**0.8600**
Enron (S) to Coauthor (T) (40%)	SVM	0.6910	0.3727	0.4842
	CRF	**1.0000**	0.3043	0.4666
	PFG	0.9916	0.4591	0.6277
	TPFG	0.5936	**0.7611**	0.6669
	TranFG	0.9793	0.5525	**0.7065**

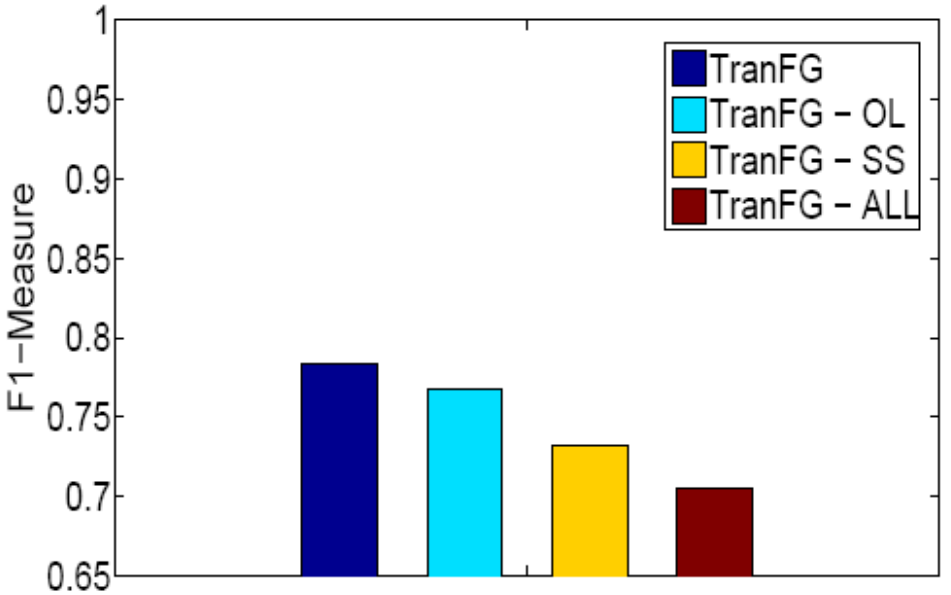
Factor Contribution Analysis



SH-Structural hole;

SB-Social balance.

Undirected Network



OL-Opinion leader;

SS-Social status.

Directed Network

Conclusions and Future Work

- **Conclusions**
  - different types of social ties have essentially different structural patterns in social networks;
  - By incorporating social theories, our proposed model can significantly improve (+4-14%) the inferring accuracy.

- **Future work**
  - Inferring complex relationships between users, e.g., family, colleague, manager-subordinate;
  - Active learning for inferring social ties.

Social Influence Analysis in Large-scale Networks

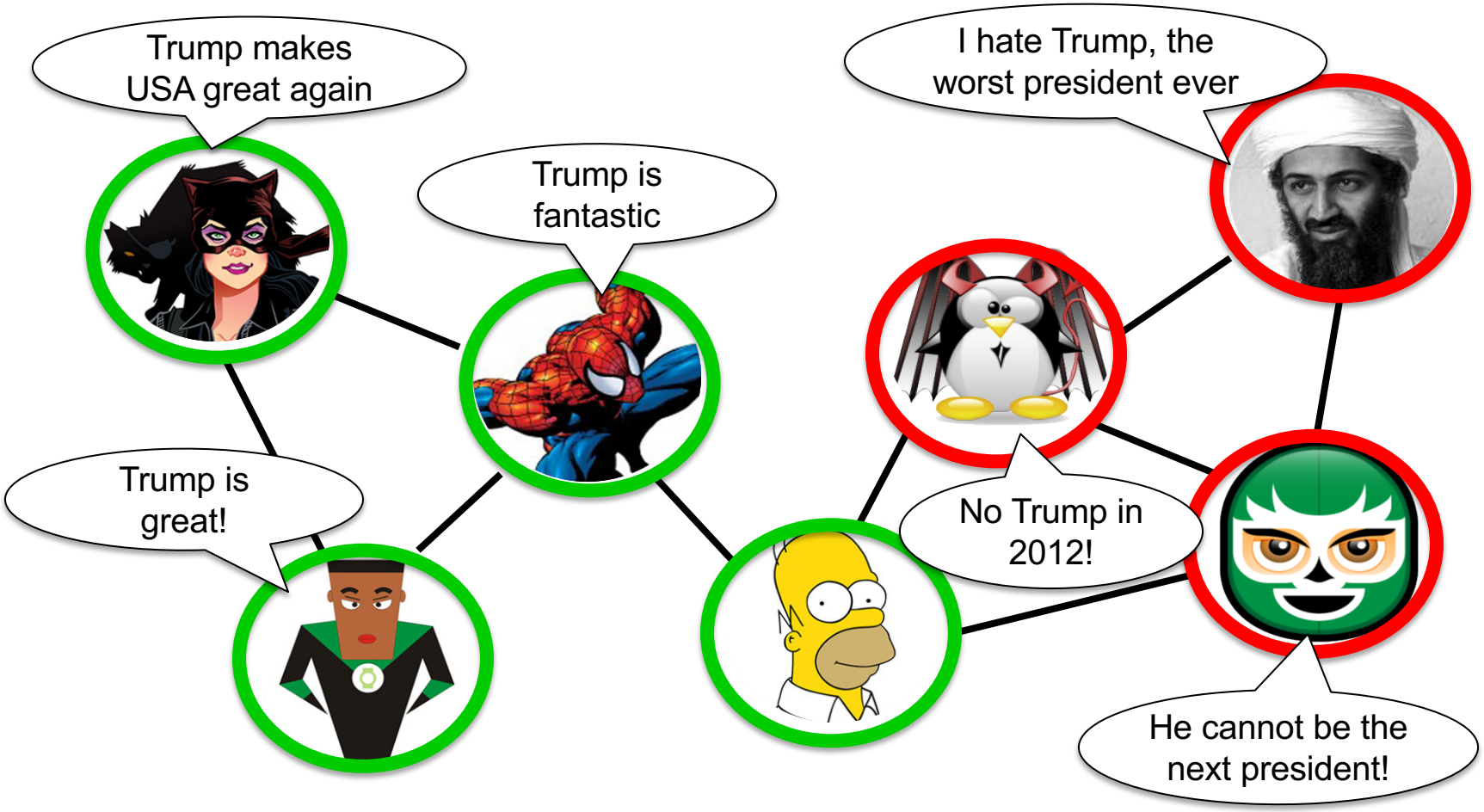
(KDD'09)

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang

Tsinghua University

(Top 8 cited papers among all papers published at KDD in the past 10 years)

User Opinion and Influence: “Love Trump”



Social Interaction: Influence

- Case 1: Social influence and political mobilization[1]
  - Will online political mobilization really work?

A controlled trial (with 61M users on FB)

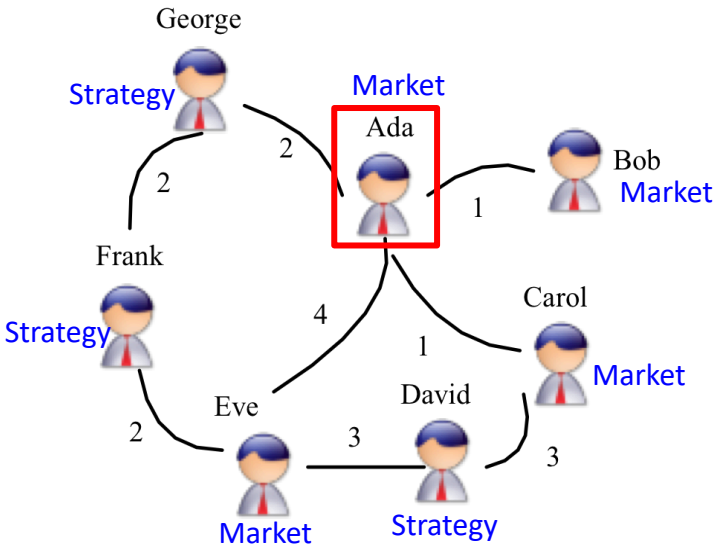
- Social msg group: was shown with msg that indicates one's friends who have made the votes.
- Informational msg group: was shown with msg that indicates how many other.
- Control group: did not receive any msg.



[1] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. I. Kramer, C. Marlow, J. E. Settle and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. Nature, 489:295-298, 2012.

技术成果

- 基于用户影响力的流失好友推荐
  - 目标：期望通过活跃玩家拉动已经流失的好友玩家，以提升触达回流用户的质量以及增加好友回流的可能性（回流率）。
  - 利用游戏中形成的玩家网络分析玩家间的影响力，向用户推荐流失好友。
  - 影响力分析模型：Topical Affinity Propagation (TAP)



Basic Idea:
If a user is located in the center of a "Market" community, and is "similar" to the other users, then she/he would have a strong influence on the other users.
—Homophily theory

系统上线结果

- 基于用户影响力的流失好友推荐
  - **王者荣耀**好友召回线上测试：将发起召回的玩家分成三组，分别使用TAP，亲密度，随机三种算法计算该玩家与好友之间的影响。对每个发起召回的玩家，选取影响最大的流失好友推荐给该玩家，最多可推荐5名好友。
    - 亲密度算法根据游戏行为获得，如给好友送金币（物品），和好友开黑。
  - 统计结果表明TAP组的玩家召回成功率明显高于亲密度组和随机组。

	TAP			亲密度			随机排序		
推荐列表长度	收到召回邀请	成功召回人数	召回成功率	收到召回邀请	成功召回人数	召回成功率	收到召回邀请	成功召回人数	召回成功率
1	3996761	1953709	**48.88%**	6617662	2732167	41.29%			
2	2567279	1272037	**49.55%**	9756330	4116895	42.20%			
3	1449256	727728	**50.21%**	10537994	4474236	42.46%			
4	767239	389588	**50.78%**	9891868	4255347	43.02%			
5	3997251	2024859	**50.66%**	15695743	6589022	41.98%	2016487	949786	47.10%

项目产出及成果

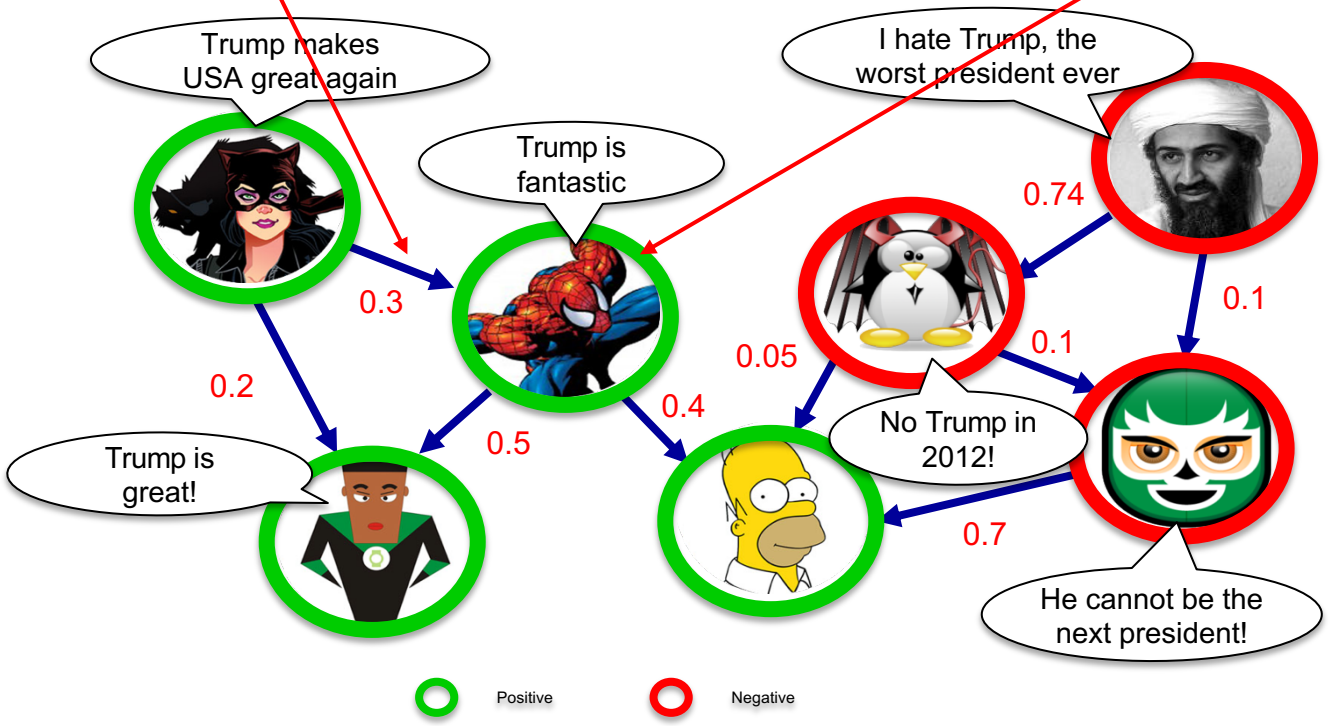
- 系统成果与算法应用
  - 利用TAP算法自动学习不同社交网络的用户影响力，输出多个影响力网络图
  - 应用于用户流失预测，可大幅提高准确度，F1值较亲密度算法提高+20%
  - 2017-2018新落地8个游戏，一年内召回3480万玩家
  - 论文
    - Chenhui Zhang, Sida Gao, Jie Tang, Tracy Xiao Liu, Zhanpeng Fang, and Xu Cheng. Learning Triadic Influence in Large Social Networks. In ASONAM'16.
    - Chenhui Zhang and Jie Tang. Learning Group Representations in Social Networks. KDD'18 (submitted)
  - 专利
    - 唐杰、张晨晖、程序. 一种基于影响力的好友召回应用。
- 下一步工作
  - 基于社群演化的表示学习及其社区推荐

How to do that?

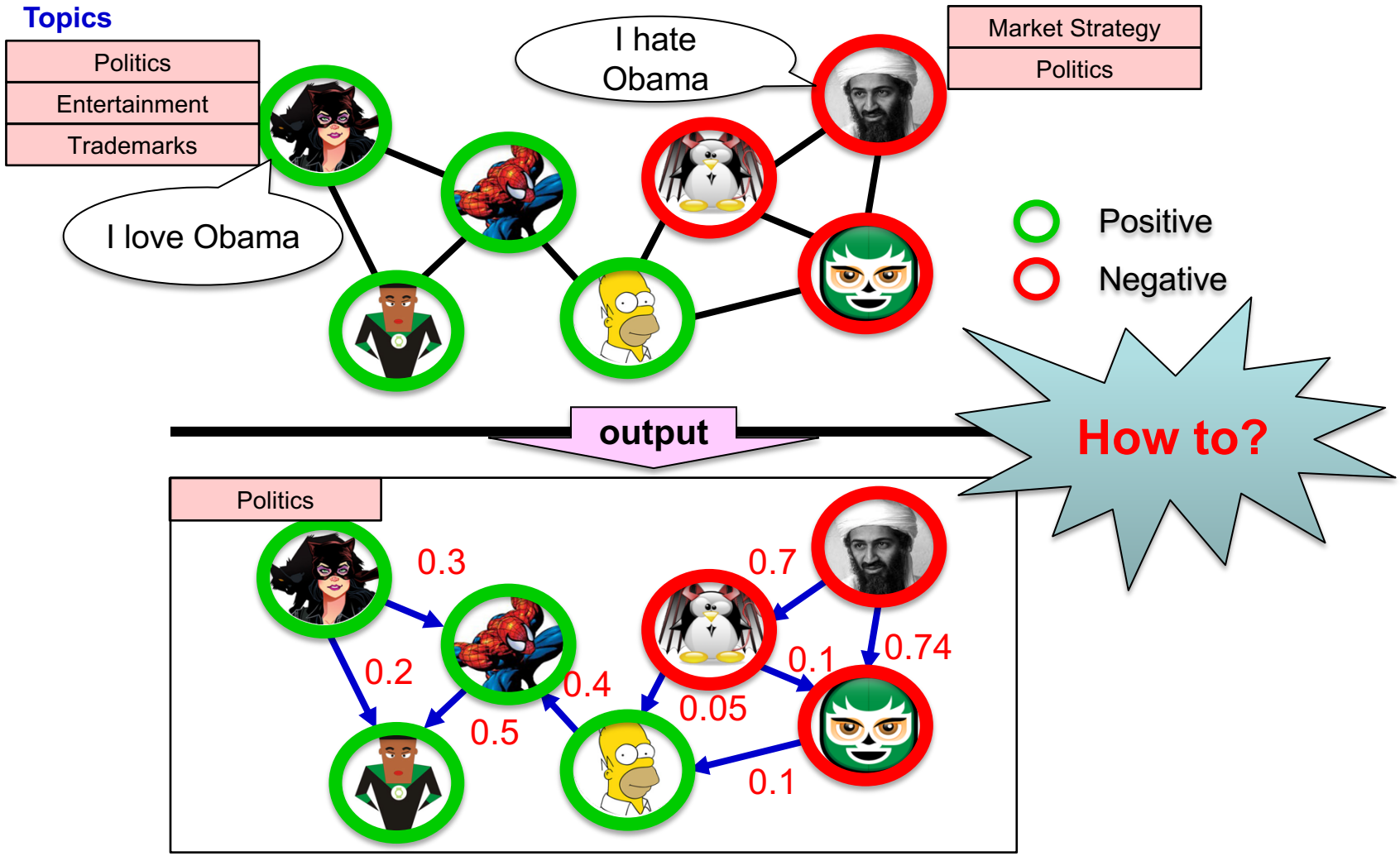
Social Influence Analysis: not just testing

① Who influenced who? What is the **influence probability**?

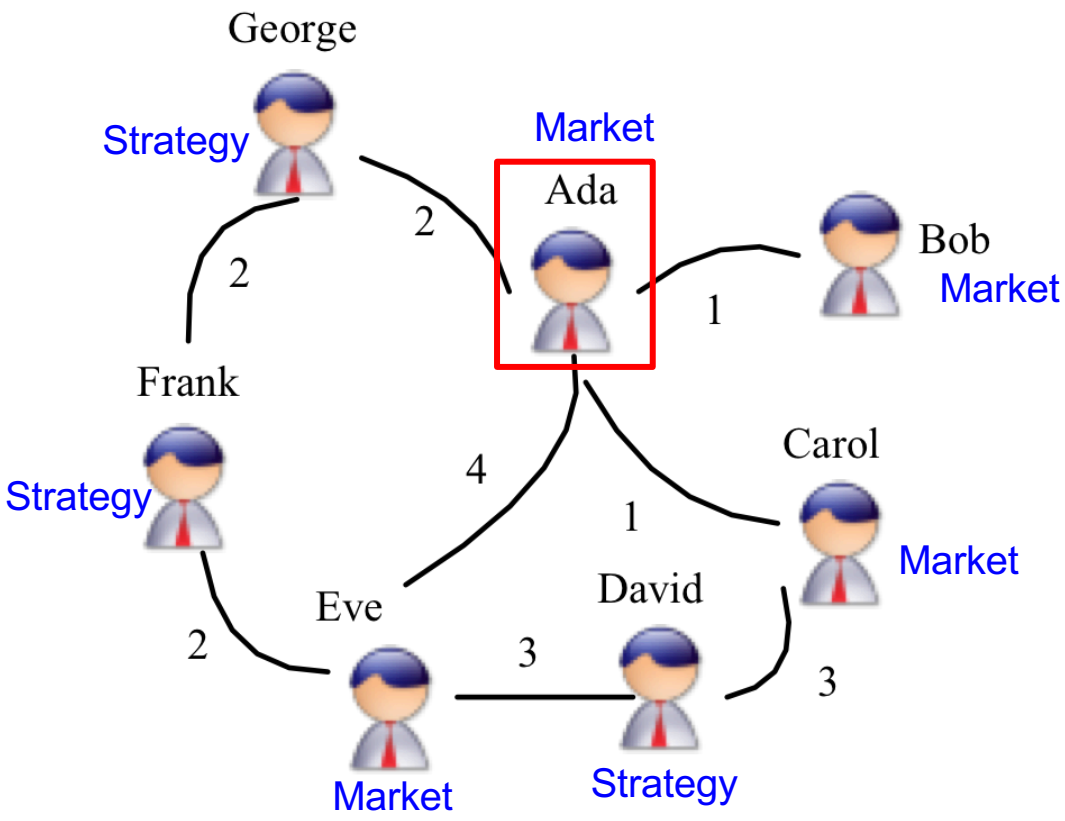
② Can we leverage the social influence to help **practical applications**?



Formulation: Topic-based Social Influence Analysis



The Solution: Topical Affinity Propagation



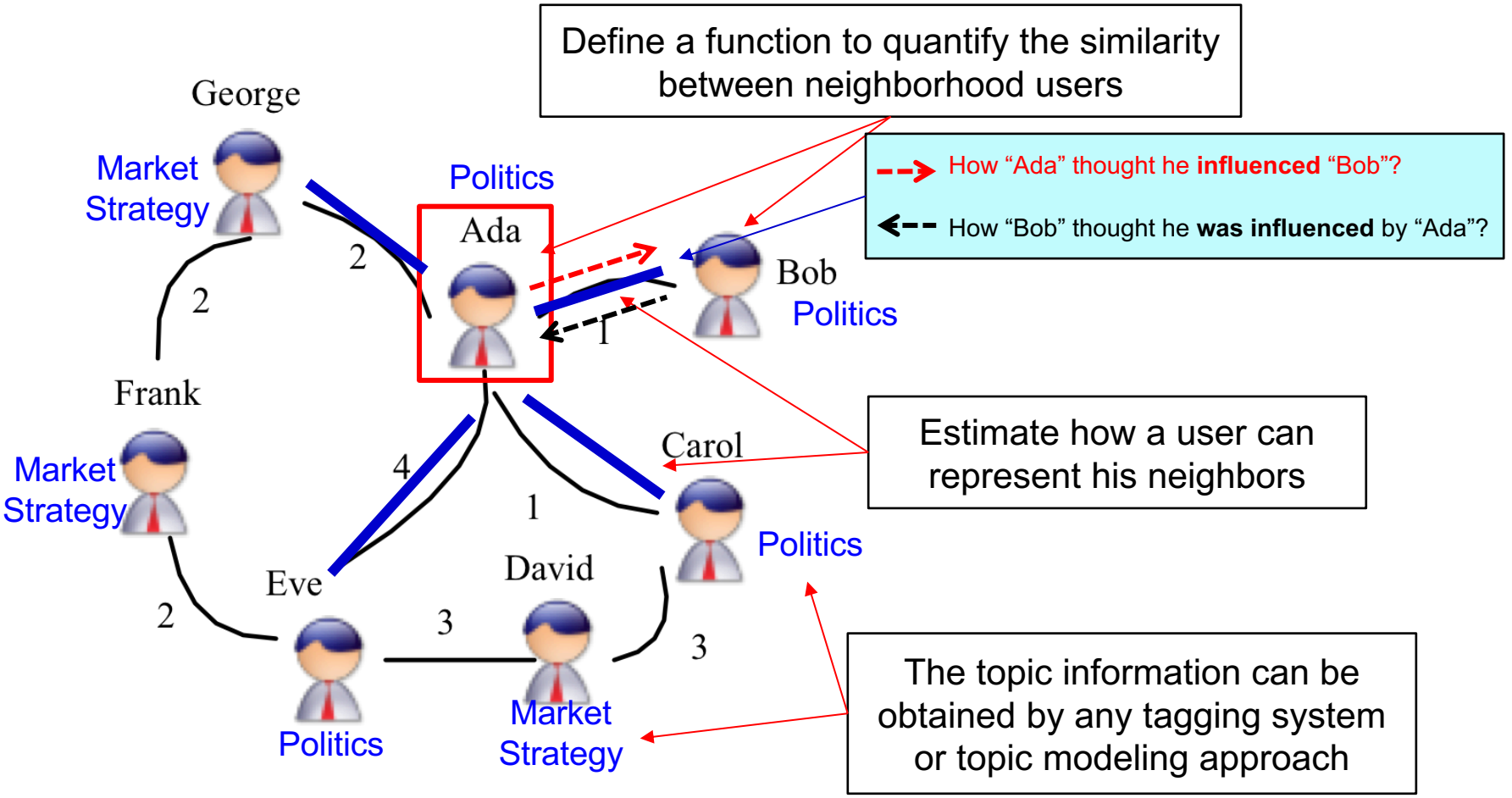
Basic Idea:

If a user is located in the center of a "Market" community, and is "similar" to the other users, then she/he would have a strong influence on the other users.

—Homophily theory

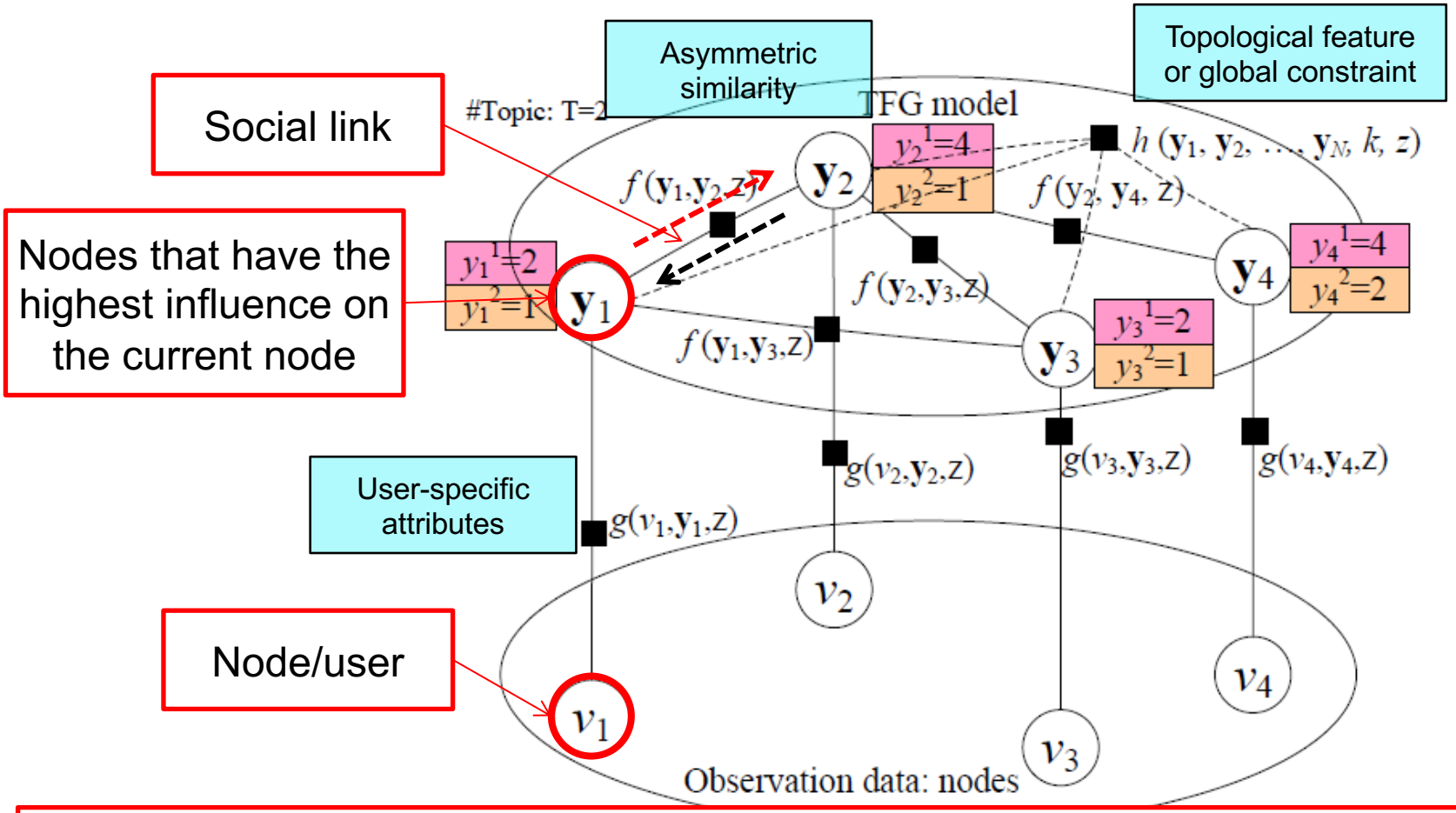
[1] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social Influence Analysis in Large-scale Networks. **KDD'09**, pages 807-816.

The Solution: Topical Affinity Propagation



[1] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social Influence Analysis in Large-scale Networks. **KDD'09**, pages 807-816. (**Top 10 cited paper among all papers published at KDD in the past 10 years)**

Topical Factor Graph (TFG) Model



The problem is cast as identifying which node has the **highest probability** to **influence** another node on a **specific topic** along with the edge.

Topical Factor Graph (TFG)

- Objective function:

$$P(\mathbf{v}, \mathbf{Y}) = \frac{1}{Z} \prod_{k=1}^{N} \prod_{z=1}^{T} h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z) \prod_{i=1}^{N} \prod_{z=1}^{T} g(v_i, \mathbf{y}_i, z) \prod_{e_{kl} \in E} \prod_{z=1}^{T} f(\mathbf{y}_k, \mathbf{y}_l, z)$$

1. How to define?

2. How to optimize?

- The learning task is to find a configuration for all $\{y_i\}$ to maximize the joint probability.

How to define (topical) feature functions?

– Node feature function

similarity

$$g(v_i, \mathbf{y}_i, z) = \begin{cases} \frac{w^z_{i y^z_i}}{\sum_{j \in NB(i)} (w^z_{ij} + w^z_{ji})} & y^z_i \neq i \\ \frac{\sum_{j \in NB(i)} w^z_{ji}}{\sum_{j \in NB(i)} (w^z_{ij} + w^z_{ji})} & y^z_i = i \end{cases}$$

– Edge feature function

$$f(y_i, y_j) = \begin{cases} w[v_i \sim v_j] & y_i = y_j \\ 1 - w[v_i \sim v_j] & y_i \neq y_j \end{cases}$$

or simply binary

– Global feature function

$$h(\mathbf{y}_1, \ldots, \mathbf{y}_N, k, z) = \begin{cases} 0 & \text{if } y^z_k = k \text{ and } y^z_i \neq k \text{ for all } i \neq k \\ 1 & \text{otherwise.} \end{cases}$$

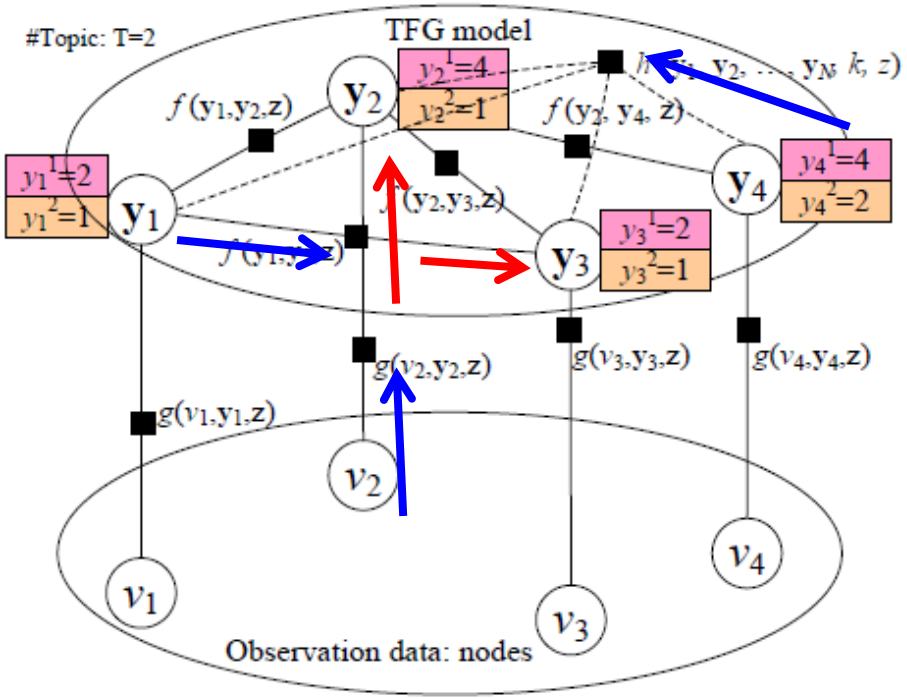
Model Learning Algorithm

Sum-product:

$$m_{y\to f}(y,z) = \prod_{f'\sim y\setminus f} m_{f'\to y}(y,z) \prod_{z'\neq z} \prod_{f'\sim y\setminus f} m_{f'\to y}(y,z')^{(\tau_{z'z})}$$

$$m_{f\to y}(y,z) = \sum_{\sim\{y\}} \left(f(Y,z) \prod_{y'\sim f\setminus y} m_{y'\to f}(y',z) \right)$$

$$+ \sum_{z'\neq z} \tau_{z'z} \sum_{\sim\{y\}} \left(f(Y,z') \prod_{y'\sim f\setminus y} m_{y'\to f}(y',z') \right) \quad (4)$$



- Low efficiency!
- Not easy for distributed learning!

New TAP Learning Algorithm

1. Introduce two new variables *r* and *a, to* replace the original message *m*.

2. Design new update rules:

How user *i* thought he **influenced** user *j*?

m_{ij}

$$r_{ij}^z = b_{ij}^z - \max_{k \in NB(j)} \{b_{ik}^z + a_{ik}^z\}$$

$$a_{jj}^z = \max_{k \in NB(j)} \min\{r_{kj}^z, 0\}$$

$$a_{ij}^z = \min(\max\{r_{jj}^z, 0\}, -\min\{r_{jj}^z, 0\} - \max_{k \in NB(j) \setminus \{i\}} \min\{r_{kj}^z, 0\}), i \in NB(j)$$

How user *j* thought he **was influenced** by user *i*?

[1] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social Influence Analysis in Large-scale Networks. In KDD, pages 807-816, 2009.

The TAP Learning Algorithm

Input: $G = (V, E)$ and topic distributions $\{\theta_v\}_{v \in V}$
Output: topic-level social influence graphs $\{G_z = (V_z, E_z)\}_{z=1}^{T}$

1.1 Calculate the node feature function $g(v_i, \mathbf{y}_i, z)$;
1.2 Calculate b_{ij}^z according to Eq. 8;
1.3 Initialize all $\{r_{ij}^z\} \leftarrow 0$;
1.4 **repeat**
1.5 **foreach** *edge-topic pair* (e_{ij}, z) **do**
1.6 Update r_{ij}^z according to Eq. 5;
1.7 **end**
1.8 **foreach** *node-topic pair* (v_j, z) **do**
1.9 Update a_{jj}^z according to Eq. 6;
1.10 **end**
1.11 **foreach** *edge-topic pair* (e_{ij}, z) **do**
1.12 Update a_{ij}^z according to Eq. 7;
1.13 **end**
1.14 **until** *convergence*;
1.15 **foreach** *node* v_t **do**
1.16 **foreach** *neighboring node* $s \in NB(t) \cup \{t\}$ **do**
1.17 Compute μ_{st}^z according to Eq. 9;
1.18 **end**
1.19 **end**
1.20 Generate $G_z = (V_z, E_z)$ for every topic z according to $\{\mu_{st}^z\}$;

$$b_{ij}^z = \log \frac{g(v_i, \mathbf{y}_i, z)|_{y_i^z = j}}{\sum_{k \in NB(i) \cup \{i\}} g(v_i, \mathbf{y}_i, z)|_{y_i^z = k}}$$

$$r_{ij}^z = b_{ij}^z - \max_{k \in NB(j)} \{b_{ik}^z + a_{ik}^z\}$$

$$a_{jj}^z = \max_{k \in NB(j)} \min\{r_{kj}^z, 0\}$$

$$a_{ij}^z = \min(\max\{r_{jj}^z, 0\}, -\min\{r_{jj}^z, 0\} - \max_{k \in NB(j) \setminus \{i\}} \min\{r_{kj}^z, 0\}), i \in NB(j)$$

$$\mu_{st}^z = \frac{1}{1 + e^{-(r_{ts}^z + a_{ts}^z)}}$$

Experiments

- Data set: (http://arnetminer.org/lab-datasets/soinf/)

Data set	#Nodes	#Edges
Coauthor	640,134	1,554,643
Citation	2,329,760	12,710,347
Film (Wikipedia)	18,518 films 7,211 directors 10,128 actors 9,784 writers	142,426

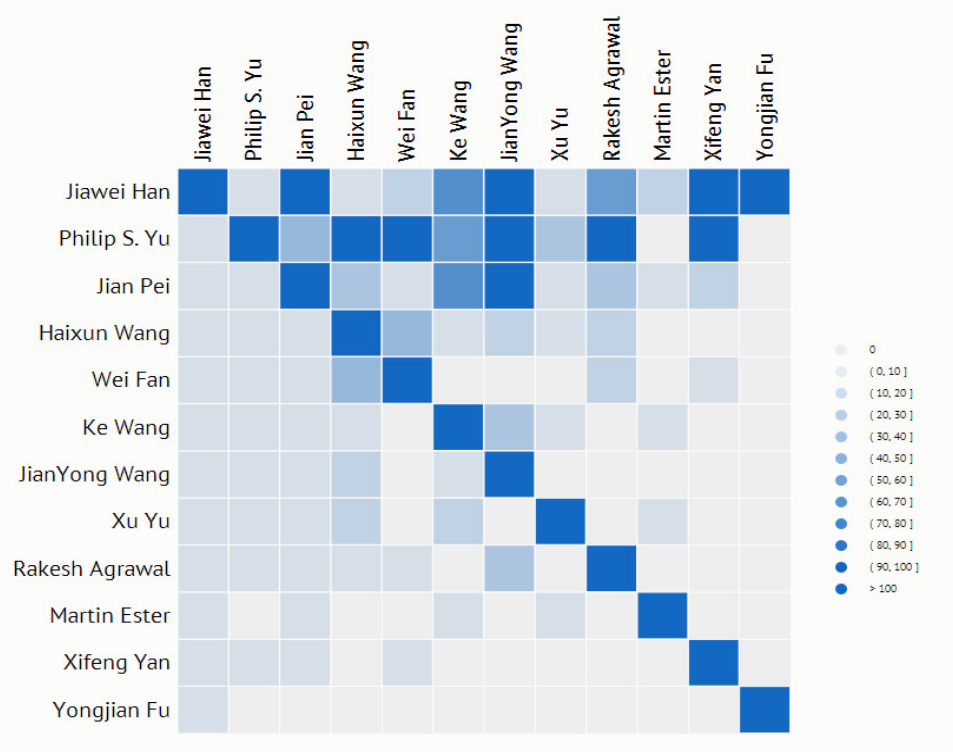
- Evaluation measures
  - CPU time
  - Case study
  - Application

Social Influence Sub-graph on "Data mining"

Table 4: Dynamic influence analysis for Dr. Jian Pei during 2000-2009. Due to space limitation, we only list coauthors who most influence on/by Dr. Pei in each time window.

Year	Pairwise	Influence
2000 - 2001	Influence on Dr. Pei	Jiawei Han (0.4961)
	Influenced by Dr. Pei	Jiawei Han (0.0082)
2002 - 2003	Influence on Dr. Pei	Jiawei Han (0.4045), Ke Wang (0.0418), Jianyong Wang (0.019), Xifeng Yan (0.007), Shiwei Tang (0.0052)
	Influenced by Dr. Pei	Shiwei Tang (0.436), Hasan M.Jamil (0.4289), Xifeng Yan (0.2192), Jianyong Wang (0.1667), Ke Wang (0.0687)
2004 - 2005	Influence on Dr. Pei	Jiawei Han (0.2364), Ke Wang (0.0328), Wei Wang (0.0294), Jianyong Wang (0.0248), Philip S. Yu (0.0156)
	Influenced by Dr. Pei	Chun Tang (0.5929), Shiwei Tang (0.5426), Hasan M.Jamil (0.3318), Jianyong Wang (0.1609), Xifeng Yan (0.1458), Yan Huang (0.1054)
2006 - 2007	Influence on Dr. Pei	Jiawei Han (0.1201), Ke Wang (0.0351), Wei Wang (0.0226), Jianyong Wang (0.018), Ada Wai-Chee Fu (0.0125)
	Influenced by Jian Pei	Chun Tang (0.6095), Shiwei Tang (0.6067), Byung-Won On (0.4599), Hasan M.Jamil (0.3433), Jaewoo Kang (0.3386)
2008 - 2009	Influence on Dr. Pei	Jiawei Han (0.2202), Ke Wang (0.0234), Ada Wai-Chee Fu (0.0208), Wei Wang (0.011), Jianyong Wang (0.0095)
	Influenced by Dr. Pei	ZhaoHui Tang (0.654), Chun Tang (0.6494), Shiwei Tang (0.5923), Zhengzheng Xing (0.5549), Hasan M.Jamil (0.3333), Jaewoo Kang (0.3057)

On "Data Mining" in 2009



Results on Coauthor and Citation

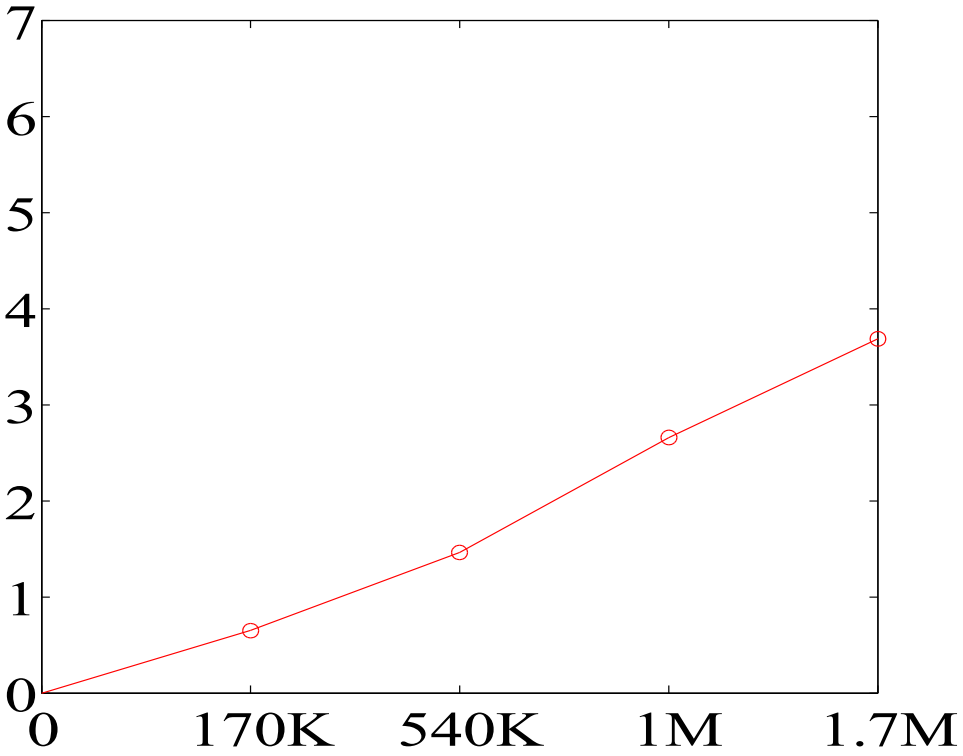
Dataset	Topic	Representative Nodes
Author	Data Mining	Heikki Mannila, Philip S. Yu, Dimitrios Gunopulos, Jiawei Han, Christos Faloutsos, Bing Liu, Vipin Kumar, Tom M. Mitchell, Wei Wang, Qiang Yang, Xindong Wu, Jeffrey Xu Yu, Osmar R. Zaiane
	Machine Learning	Pat Langley, Alex Waibel, Trevor Darrell, C. Lee Giles, Terrence J. Sejnowski, Samy Bengio, Daphne Koller, Luc De Raedt, Vasant Honavar, Floriana Esposito, Bernhard Scholkopf
	Database System	Gerhard Weikum, John Mylopoulos, Michael Stonebraker, Barbara Pernici, Philip S. Yu, Sharad Mehrotra, Wei Sun, V. S. Subrahmanian, Alejandro P. Buchmann, Kian-Lee Tan, Jiawei Han
	Information Retrieval	Gerard Salton, W. Bruce Croft, Ricardo A. Baeza-Yates, James Allan, Yi Zhang, Mounia Lalmas, Zheng Chen, Ophir Frieder, Alan F. Smeaton, Rong Jin
	Web Services	Yan Wang, Liang-jie Zhang, Schahram Dustdar, Jian Yang, Fabio Casati, Wei Xu, Zakaria Maamar, Ying Li, Xin Zhang, Boualem Benatallah, Boualem Benatallah
	Semantic Web	Wolfgang Nejdl, Daniel Schwabe, Steffen Staab, Mark A. Musen, Andrew Tomkins, Juliana Freire, Carole A. Goble, James A. Hendler, Rudi Studer, Enrico Motta
	Bayesian Network	Daphne Koller, Paul R. Cohen, Floriana Esposito, Henri Prade, Michael I. Jordan, Didier Dubois, David Heckerman, Philippe Smets
Citation	Data Mining	Fast Algorithms for Mining Association Rules in Large Databases, Using Segmented Right-Deep Trees for the Execution of Pipelined Hash Joins, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Discovery of Multiple-Level Association Rules from Large Databases, Interleaving a Join Sequence with Semijoins in Distributed Query Processing
	Machine Learning	Object Recognition with Gradient-Based Learning, Correctness of Local Probability Propagation in Graphical Models with Loops, A Learning Theorem for Networks at Detailed Stochastic Equilibrium, The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length, A Unifying Review of Linear Gaussian Models
	Database System	Mediators in the Architecture of Future Information Systems, Database Techniques for the World-Wide Web: A Survey, The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles, Fast Algorithms for Mining Association Rules in Large Databases
	Web Services	The Web Service Modeling Framework WSMF, Interval Timed Coloured Petri Nets and their Analysis, The design and implementation of real-time schedulers in RED-linux, The Self-Serv Environment for Web Services Composition
	Web Mining	Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, Fast Algorithms for Mining Association Rules in Large Databases, The OO-Binary Relationship Model: A Truly Object Oriented Conceptual Model, Distributions of Surfers' Paths Through the World Wide Web: Empirical Characterizations, Improving Fault Tolerance and Supporting Partial Writes in Structured Coterie Protocols for Replicated Objects
	Semantic Web	FaCT and iFaCT, The GRAIL concept modelling language for medical terminology, Semantic Integration of Semistructured and Structured Data Sources, Description of the RACER System and its Applications, DL-Lite: Practical Reasoning for Rich Dls

Scalability Performance

Table 2: Scalability performance of different methods on real data sets. >10hr means that the algorithm did not terminate when the algorithm runs more than 10 hours.

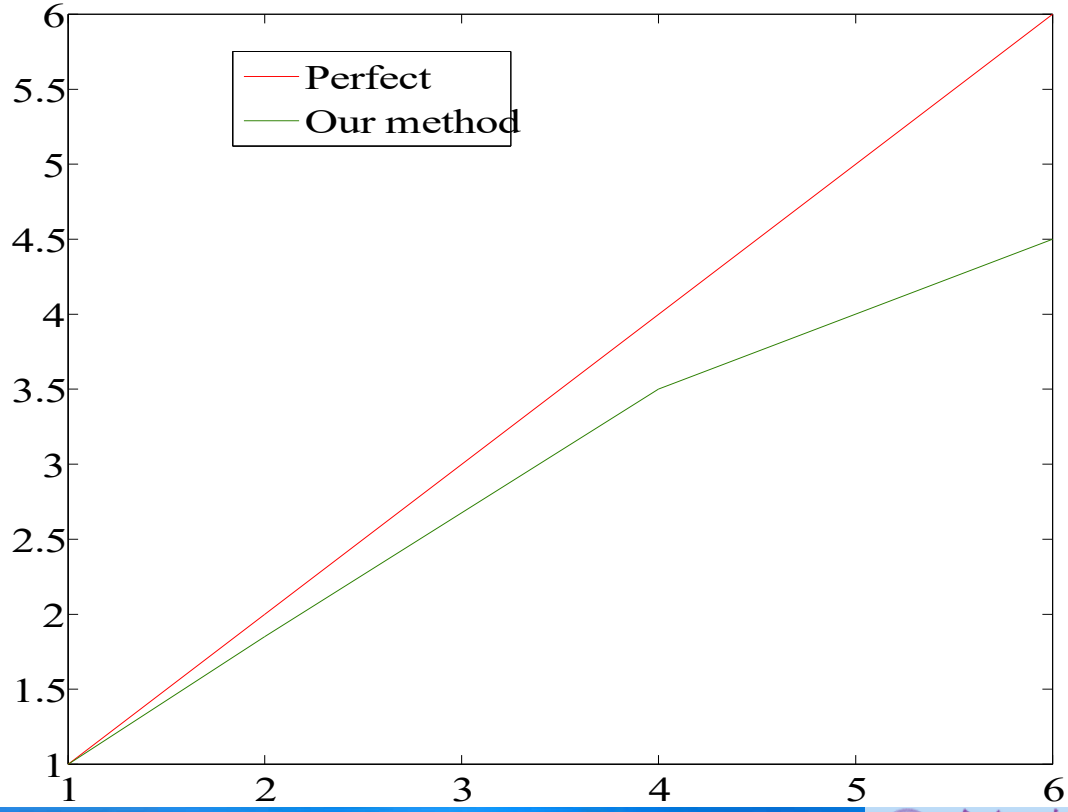
Methods	Citation	Coauthor	Film
Sum-Product	N/A	>10hr	1.8 hr
Basic TAP Learning	>10hr	369s	**57s**
Distributed TAP Learning	**39.33m**	**104s**	148s

Speedup results



Speedup vs. Dataset size

Speedup vs. #Computer nodes



GMNN: Graph Markov Neural Networks

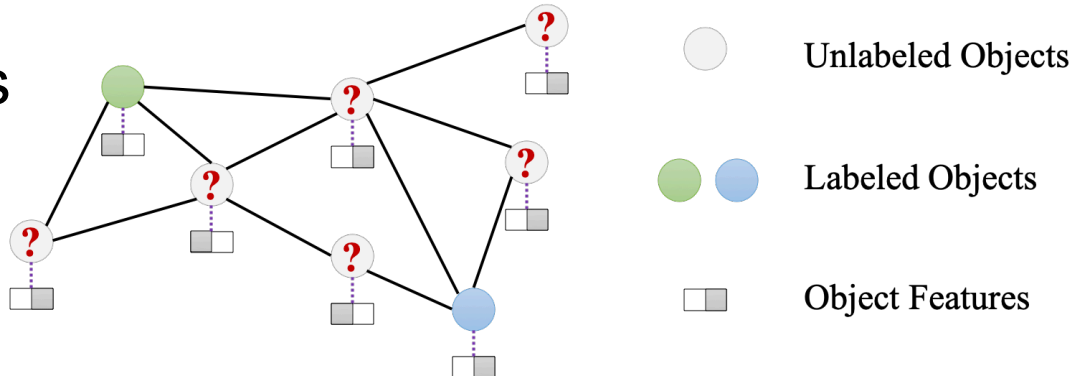
(ICML'19)

Meng Qu, Yoshua Bengio, Jian Tang

MILA

Semi-supervised Object Classification

- Given G= (V, E, $\mathbf{x}_V$)
  - $V = V_L \cup V_U$: objects/nodes
  - E : edges
  - $\mathbf{x}_V$: object features

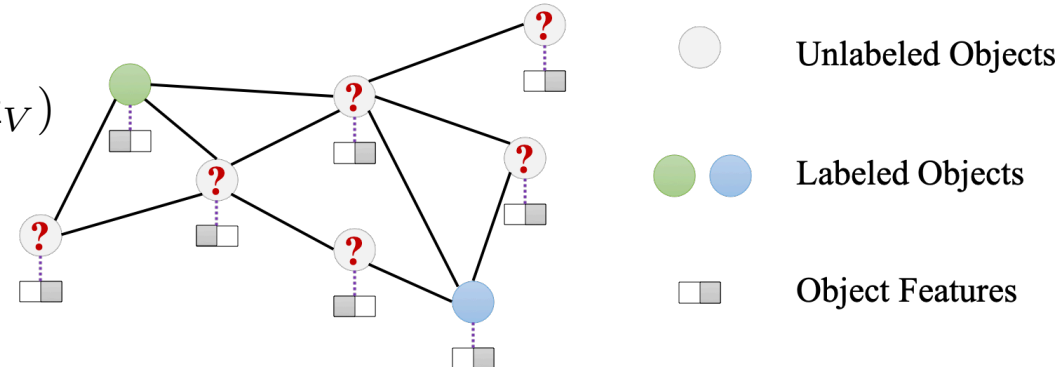


- Give some labeled objects V_L, we want to infer the labels of the rest of objects V_U
- Many other tasks on graphs can be formulated as object classification
  - E.g., link classification

Related Work: Statistical Relational Learning

- Models the joint distribution of the object labels given the object features, i.e., $p(\mathbf{y}_V|\mathbf{x}_V)$ with conditional random fields

$$p(\mathbf{y}_V|\mathbf{x}_V) = \frac{1}{Z(\mathbf{x}_V)} \prod_{(i,j)\in E} \psi_{i,j}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{x}_V)$$



$$\psi_{i,j}(y_i, y_j, \mathbf{x}_V) = \exp(\sum_{k=1}^{K} \lambda_k f_k(y_i, y_j, \mathbf{x}_i, \mathbf{x}_j) + \mu_k g_k(y_i, \mathbf{x}_i))$$

Edge potential functions node potential functions

Optimization of Statistical Relational Learning

- Learning by maximizing the likelihood of the observed labels $\log p(\mathbf{y}_{\mathrm{L}}|\mathbf{x}_{\mathrm{V}})$

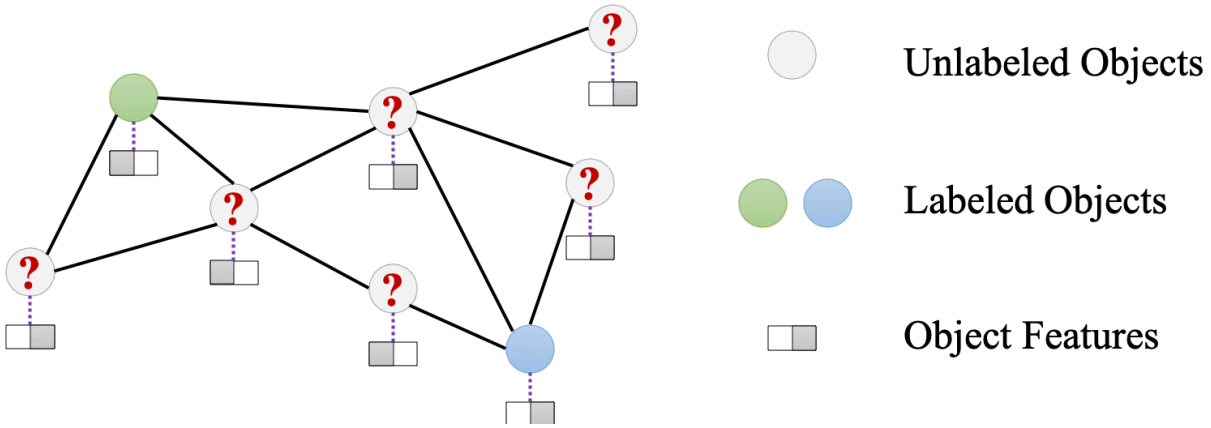
- Inferring the posterior distributions $p(\mathbf{y}_{\mathrm{U}}|\mathbf{y}_{\mathrm{L}},\mathbf{x}_{\mathrm{V}})$ with approximate methods such as loop belief propagation is used

Pros and Cons of Statistical Relation Learning

- Pros
  - Capable of modeling the dependency between the object labels
- Cons
  - Some manually defined potential functions
  - Limited model capacity
  - Inference is difficult due to the complicated graph structures

Related Work: Graph Neural Networks

- Learning effective node representations and then predicting the node labels independently
  - Graph convolutional Networks (Kipf et al. 2016)
  - Graph attention networks (Veličković et al. 2017)
  - Neural message passing (Gilmer et al. 2017)
- Predicting the node labels independently with the node representations



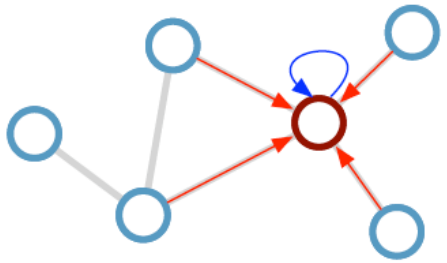
Graph Convolutional Networks (Kipf et al. 2016)

- Iteratively update the node representations by aggregating the node representations of neighbors and its own node representations
  - Starting from the initial node features $H_0 = \mathbf{x}_V$

$$H_{(l+1)} = f(H_l, A)$$

$$\hat{A} = A + I$$

$$f(H_l, A) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H_l W_l)$$



Pros and Cos of Graph Neural Networks

- Pros
  - Learns effective node representations by feature propagations
  - High model compacity by using multiple non-linear graph convolutional layers
- Cons
  - Ignore the dependency between node labels

Can we combine the advantages of both worlds?

GMNN: Graph Markov Neural Networks

- Towards combining statistical relational learning and graph neural networks
- Learning effective node representations for predicting the node labels
- Modeling the label dependencies of nodes
- State-of-the-art performance
  - semi-supervised node classification
  - unsupervised node representation
  - link classification

GMNN: Graph Markov Neural Networks

- Model the joint distribution of object labels $\mathbf{y}_V$ conditioned on object attributes $\mathbf{x}_V$, i.e., $\mathrm{p}_\phi(\mathbf{y}_V|\mathbf{x}_V)$
- Learning the model parameters ϕ by maximizing the lower-bound of log-likelihood of the observed data, $\log \mathrm{p}_\phi(\mathbf{y}_L|\mathbf{x}_V)$

$$\log p_\phi(\mathbf{y}_L|\mathbf{x}_V) \geq$$
$$\mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V) - \log q_\theta(\mathbf{y}_U|\mathbf{x}_V)]$$

Optimization with Pseudolikelihood Variational-EM

- E-step: fix p_{ϕ} and update the variational distribution $\mathrm{q}_{\theta}(\mathbf{y}_U|\mathbf{x}_V)$ to approximate the true posterior distribution $\mathrm{p}_{\phi}(\mathbf{y}_U|\mathbf{y}_L, \mathbf{x}_V)$.
- M-step: fix q_{θ} and update p_{ϕ} to maximize the lower bound

$$\ell(\phi) = \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_L, \mathbf{y}_U|\mathbf{x}_V)]$$

- Directly optimize the joint likelihood is difficult due to the partition function in p_{ϕ}, instead we optimize the pseudolikelihood function

$$\ell_{PL}(\phi) \triangleq \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{V\setminus n}, \mathbf{x}_V)]$$

$$= \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_V)]$$

Inference/E-step: approximate $\mathrm{p}_\phi(\mathbf{y}_U|\mathbf{y}_L,\mathbf{x}_V)$

- Approximate it with variational distribution $\mathrm{q}_\theta(\mathbf{y}_U|\mathbf{x}_V)$. Specifically we use mean-field method:

$$q_\theta(\mathbf{y}_U|\mathbf{x}_V) = \prod_{n\in U} q_\theta(\mathbf{y}_n|\mathbf{x}_V)$$

- We parametrize each variational distribution with a Graph Neural Network

$$q_\theta(\mathbf{y}_n|\mathbf{x}_V) = \mathrm{Cat}(\mathbf{y}_n|\mathrm{softmax}(W_\theta \mathbf{h}_{\theta,n}))$$

Object representations learned by GNN

Inference/E-step: approximate $\mathrm{p}_{\phi}(\mathbf{y}_U|\mathbf{y}_L, \mathbf{x}_V)$

- The optimal variational distribution satisfies:

$$\log q_\theta(\mathbf{y}_n|\mathbf{x}_V) = \mathbb{E}_{q_\theta(\mathbf{y}_{\mathrm{NB}(n)\cap U}|\mathbf{x}_V)}[\log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB}(n)}, \mathbf{x}_V)] + \text{const.}$$

- Estimate the right term by sampling from $\mathrm{q}_\theta(\mathbf{y}_{NB(n)\cap U}|\mathbf{x}_V)$, and then we have

$$q_\theta(\mathbf{y}_n|\mathbf{x}_V) \approx p_\phi(\mathbf{y}_n|\hat{\mathbf{y}}_{\mathrm{NB}(n)}, \mathbf{x}_V)$$

Label distribution of object n by the learning module

Inference/E-step: approximate $\mathrm{p}_\phi(\mathbf{y}_U|\mathbf{y}_L, \mathbf{x}_V)$

- Minimize the KL-divergence between the two distributions
  - The supervision from the learning module is used as pseudo label to train the variational distribution

$$O_{\theta,U} = \sum_{n \in U} \mathbb{E}_{p_\phi(\mathbf{y}_n|\hat{\mathbf{y}}_{\mathrm{NB}(n)}, \mathbf{x}_V)}[\log q_\theta(\mathbf{y}_n|\mathbf{x}_V)]$$

- The variational distribution can also by trained on the labeled data

$$O_{\theta,L} = \sum_{n \in L} \log q_\theta(\mathbf{y}_n|\mathbf{x}_V).$$

- Final objective:

$$O_\theta = O_{\theta,U} + O_{\theta,L}$$

Learning/M-step:

- The log-pseudo likelihood:

$$\ell_{PL}(\phi) \triangleq \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{V\setminus n}, \mathbf{x}_V)]$$

$$= \mathbb{E}_{q_\theta(\mathbf{y}_U|\mathbf{x}_V)}[\sum_{n\in V} \log p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x}_V)]$$

- According to the inference, only the $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x_V})$ is required
- Parametrize $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}}, \mathbf{x_V})$ with another GCN

$$p_\phi(\mathbf{y}_n|\mathbf{y}_{\mathrm{NB}(n)}, \mathbf{x}_V) = \mathrm{Cat}(\mathbf{y}_n|\mathrm{softmax}(W_\phi \mathbf{h}_{\phi,n}))$$

Learning/M-step:

- Estimate the expectation by drawing a sample from $\mathrm{q}_\theta(\mathbf{y}_U|\mathbf{x}_V)$
- Final objective:

$$O_\phi = \sum_{n \in V} \log p_\phi(\hat{\mathbf{y}}_n | \hat{\mathbf{y}}_{\mathrm{NB(n)}}, \mathbf{x}_V)$$

$\hat{\mathbf{y}}_n \sim \mathrm{q}_\theta(\mathbf{y}_U|\mathbf{x}_V)$, if n is unlabeled

$\hat{\mathbf{y}}_n$ is the ground truth label, if n is labeled

Overall Optimization Procedure

- Two Graph Neural Networks Collaborate with each other
  - p_ϕ: learning network, modeling the label dependency
  - q_θ: inference network, learning the object representations
- q_θ infer the labels of unlabeled objects trained with supervision from p_ϕ and labeled objects
- p_ϕ is trained with a fully labeled graph, where the unlabeled objects are labeled by q_θ

Applications: Object/Node Classification

- Train, validation, and test are standard split
- State-of-the-art performance

Category	Algorithm	Cora	Citeseer	Pubmed
SSL	LP	74.2	56.3	71.6
SRL	PRM	77.0	63.4	68.3
	RMN	71.3	68.0	70.7
	MLN	74.6	68.0	75.3
GNN	Planetoid *	75.7	64.7	77.2
	GCN *	81.5	70.3	79.0
	GAT *	83.0	72.5	79.0
GMNN	W/o Attr. in p_ϕ	83.4	**73.1**	81.4
	With Attr. in p_ϕ	**83.7**	72.9	**81.8**

Few-shot Learning Settings

- 5 labeled objects for each class
- The performance gain are even larger

Algorithm	Cora	Citeseer	Pubmed
GCN	74.9	69.0	76.9
GAT	77.0	68.9	75.4
GMNN	**78.6**	**72.7**	**79.1**

Ablation Study on $p_\phi(\mathbf{y_n}|\mathbf{y}_{\mathrm{NB(n)}})$

- 1 mean pooling layer: just take the average distribution of labels of neighbors,
  - label propagation!
  - Model the label dependency in a linear way

Architecture	Cora	Citeseer	Pubmed
1 Mean Pooling Layer	82.4	71.9	80.7
1 GC Layer	83.1	73.1	80.9
2 GC Layers	83.4	73.1	81.4
3 GC Layers	83.6	73.0	81.5

Ablation Study on $q_\theta(\mathbf{y}_n|\mathbf{x}_V)$

- Non-amortized: treat $q_\theta(\mathbf{y}_n|\mathbf{x}_V)$ as parameter, independent of $\mathbf{x}_V$
- 1 Linear Layer: only use the features $\mathbf{x}_n$

Architecture	Cora	Citeseer	Pubmed
Non-amortized	45.3	28.1	42.2
1 Linear Layer	55.8	57.5	69.8
1 GC Layer	72.9	67.6	71.8
2 GC Layers	83.4	73.1	81.4
3 GC Layers	82.0	70.6	80.7

Applications: Unsupervised Node Representation Learning

- There are no labeled nodes!!
- Instead, we introduce a pseudo task. For each node n, we aims to predict the neighbors , i.e., $p(y\,|\mathrm{n}, \mathbf{x}_V)$
- E-step: infer the neighbor distribution for each node with q_θ
- M-step: update the p_ϕ to model the local dependency of the inferred neighbor distributions

Node/Object classification

- DGI: Deep Graph Infomax, Veličković et al. 2019

Category	Algorithm	Cora	Citeseer	Pubmed
GNN	DeepWalk *	67.2	43.2	65.3
	DGI *	82.3	**71.8**	76.8
GMNN	With only q_θ .	78.1	68.0	79.3
	With q_θ and p_ϕ	**82.8**	71.5	**81.6**

Applications: Link Classification

- Construct a dual graph $\tilde{G}$ from the original graph G
  - Each edge in G -> a node in $\tilde{G}$
  - Two nodes in $\tilde{G}$ are connected if the corresponding edges in G share a node

Category	Algorithm	Bitcoin Alpha	Bitcoin OTC
SSL	LP	59.68	65.58
SRL	PRM	58.59	64.37
	RMN	59.56	65.59
	MLN	60.87	65.62
GNN	DeepWalk	62.71	63.20
	GCN	64.00	65.69
GMNN	W/o Attr. in p_ϕ	65.59	66.62
	With Attr. in p_ϕ	**65.86**	**66.83**

Summary of GMNN

- A fundamental problem on graphs: semi-supervised node classification
- GMNN: towards combining statistical relational learning and graph neural networks
  - Model the label dependency with one graph neural network
  - Learn effective node representations with another graph neural network
- State-of-the-art results on semi-supervised node classification, unsupervised node representation, and link classification
- Code available at: https://github.com/DeepGraphLearning/GMNN

References

- Baum, Leonard E., and Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains." The annals of mathematical statistics 37.6 (1966): 1554-1563.
- Manning, Christopher D., Christopher D. Manning, and Hinrich Schütze. Foundations of statistical natural language processing. MIT press, 1999.
- Berger, Adam L., Vincent J. Della Pietra, and Stephen A. Della Pietra. "A maximum entropy approach to natural language processing." Computational linguistics 22.1 (1996): 39-71.
- Ratnaparkhi, Adwait. "Maximum entropy models for natural language ambiguity resolution." (1998).
- Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).
- Tang, Jie, Tiancheng Lou, and Jon Kleinberg. "Inferring social ties across heterogenous networks." Proceedings of the fifth ACM international conference on Web search and data mining. 2012.
- Tang, Jie, et al. "Social influence analysis in large-scale networks." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009.
- Qu, Meng, Yoshua Bengio, and Jian Tang. "Gmnn: Graph markov neural networks." arXiv preprint arXiv:1905.06214 (2019).

Thanks!

HP: http://keg.cs.tsinghua.edu.cn/jietang/

Email: jietang@tsinghua.edu.cn