Embedding as a Tool for Algorithm Design

Le Song

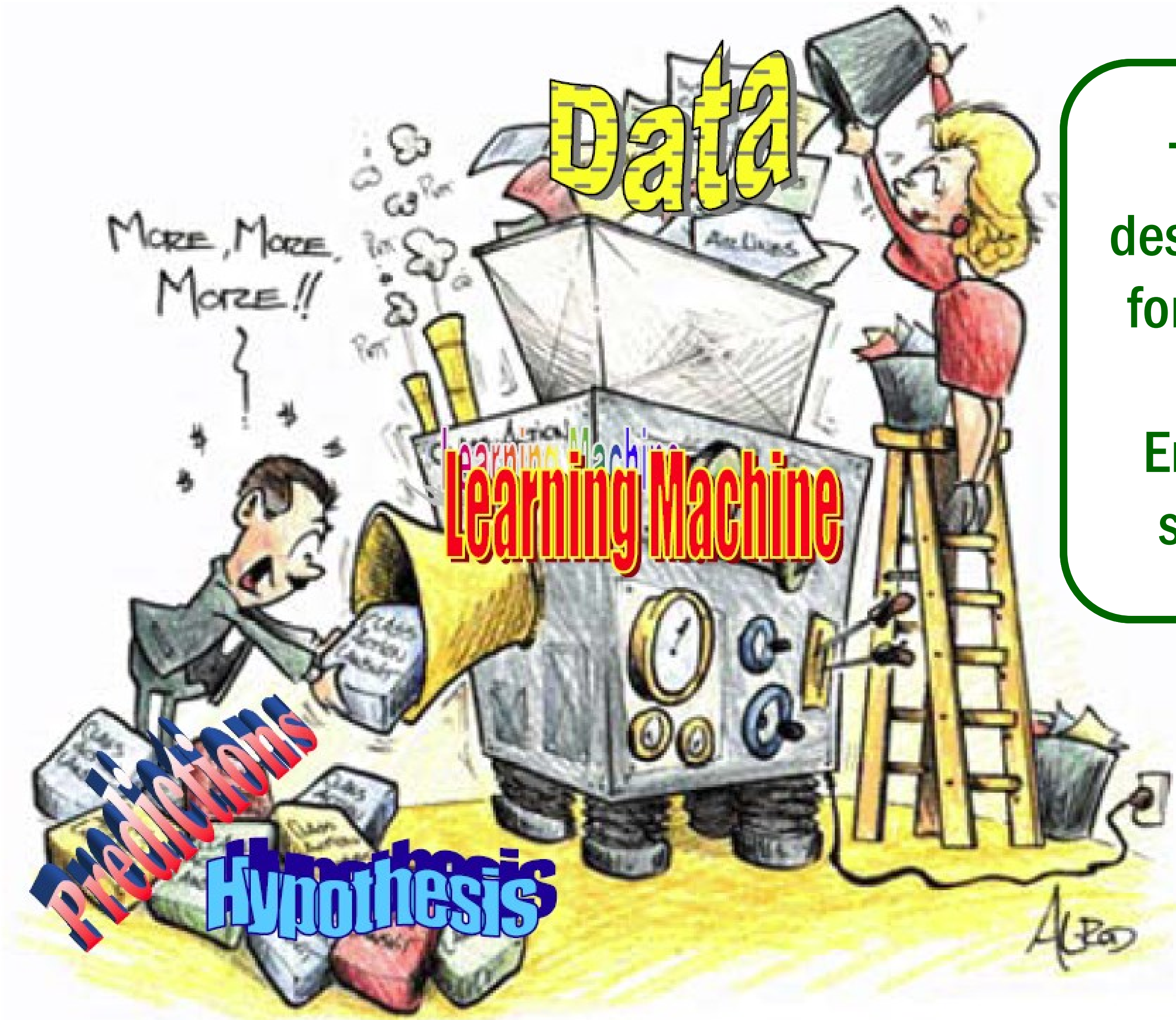
College of Computing

Center for Machine Learning

Georgia Institute of Technology

What is machine learning (ML)

Design algorithms and systems that can improve their <u>performance</u> with <u>data</u>

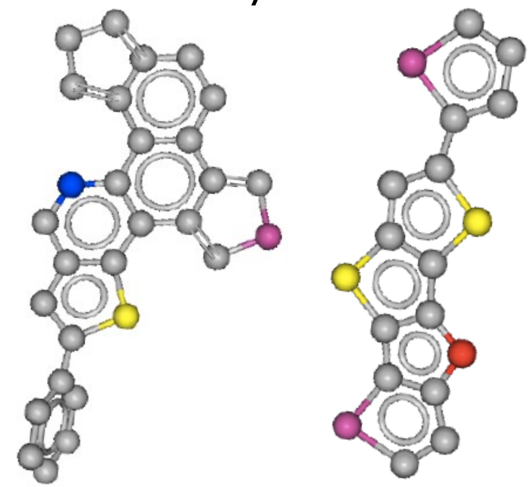


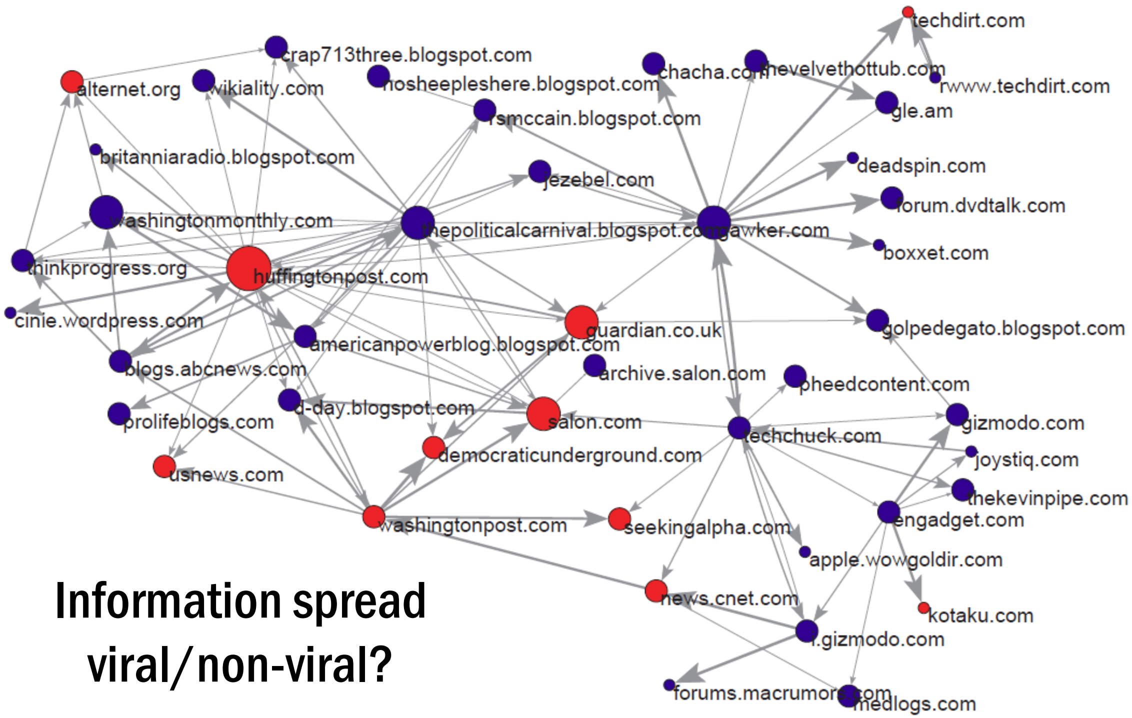
The best design pattern for big data?

Embedding structures

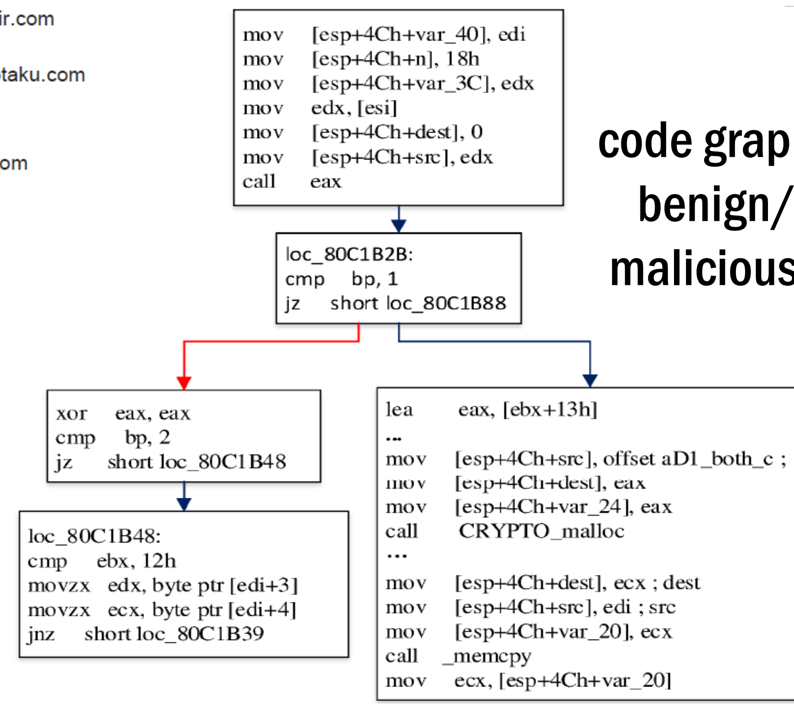
Ex 1: Prediction for structured data

Drug/materials effective/ineffective?



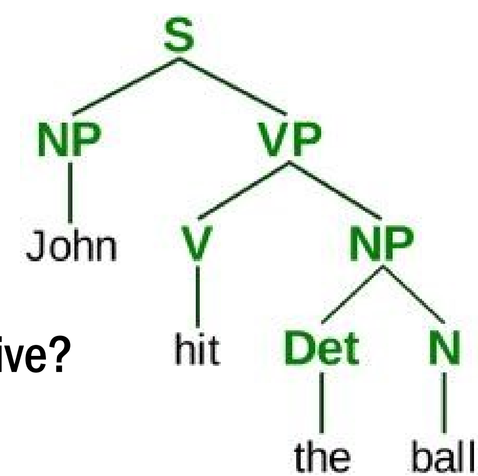


Information spread viral/non-viral?

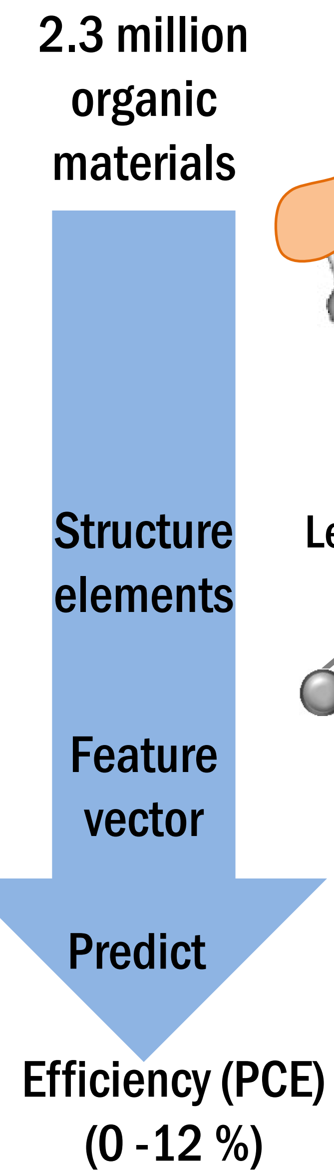


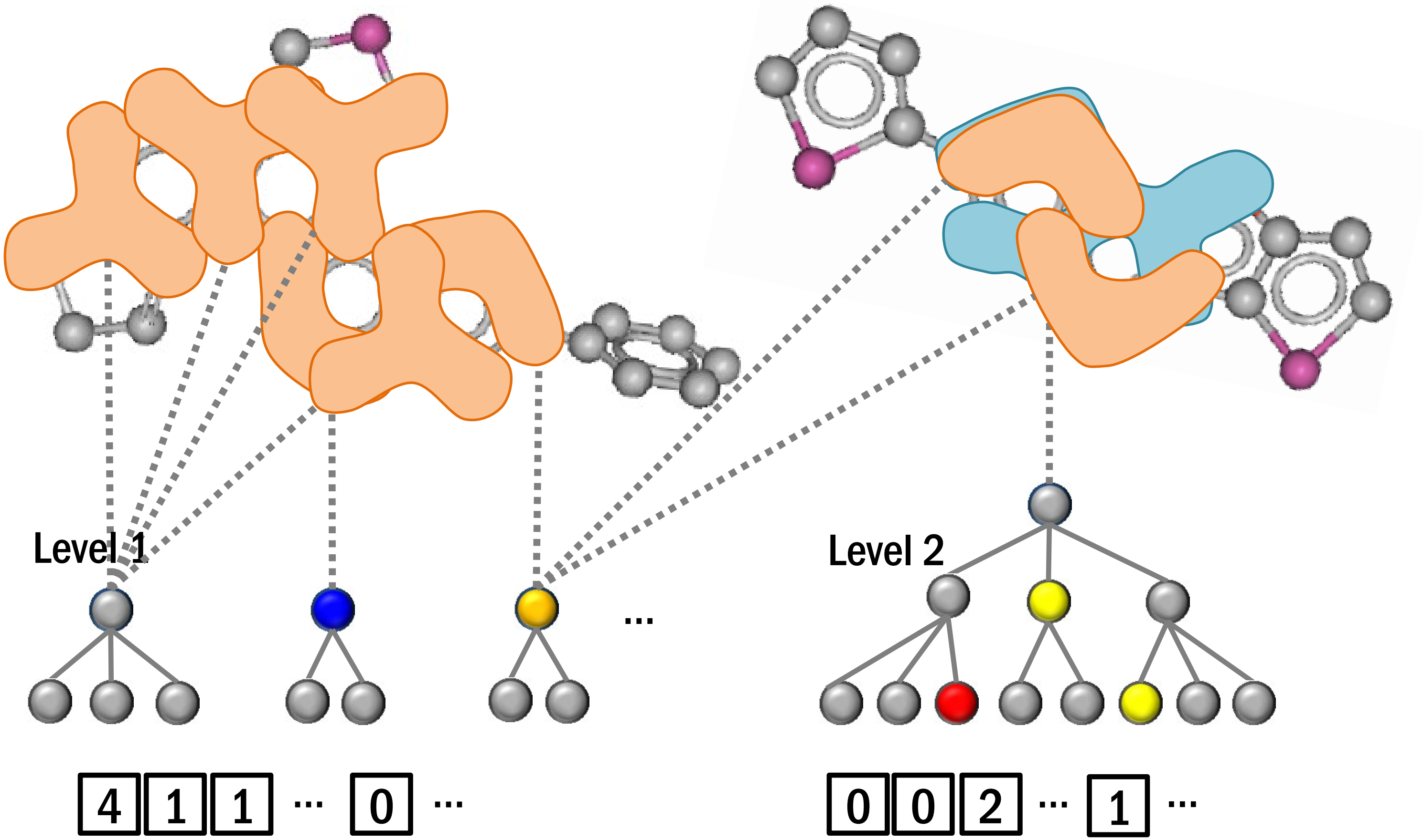
code graphs benign/malicious?

Natural language positive/negative?



Big dataset, explosive feature space

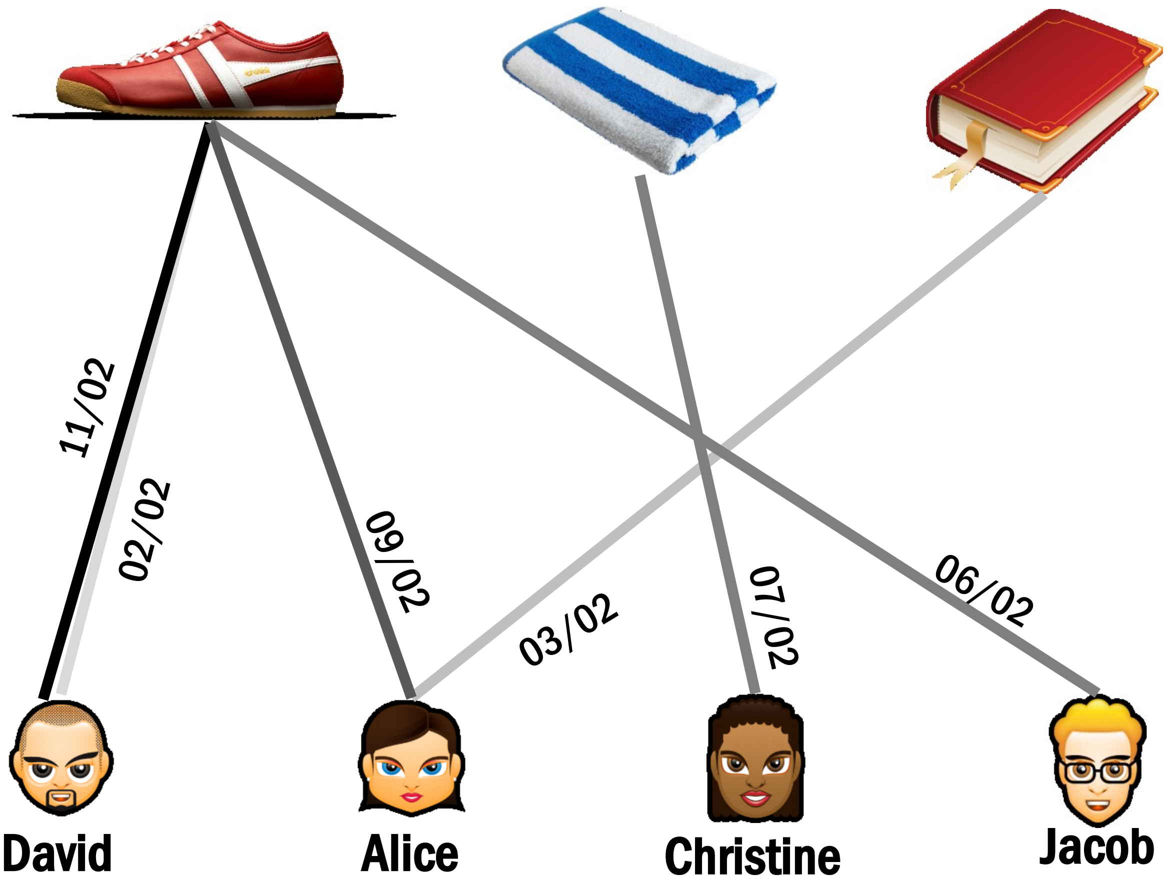




method	dimension	MAE
Level 6	1.3 billion	0.096
Embedding	0.1 million	0.085

Reduce model size by 10,000 times!

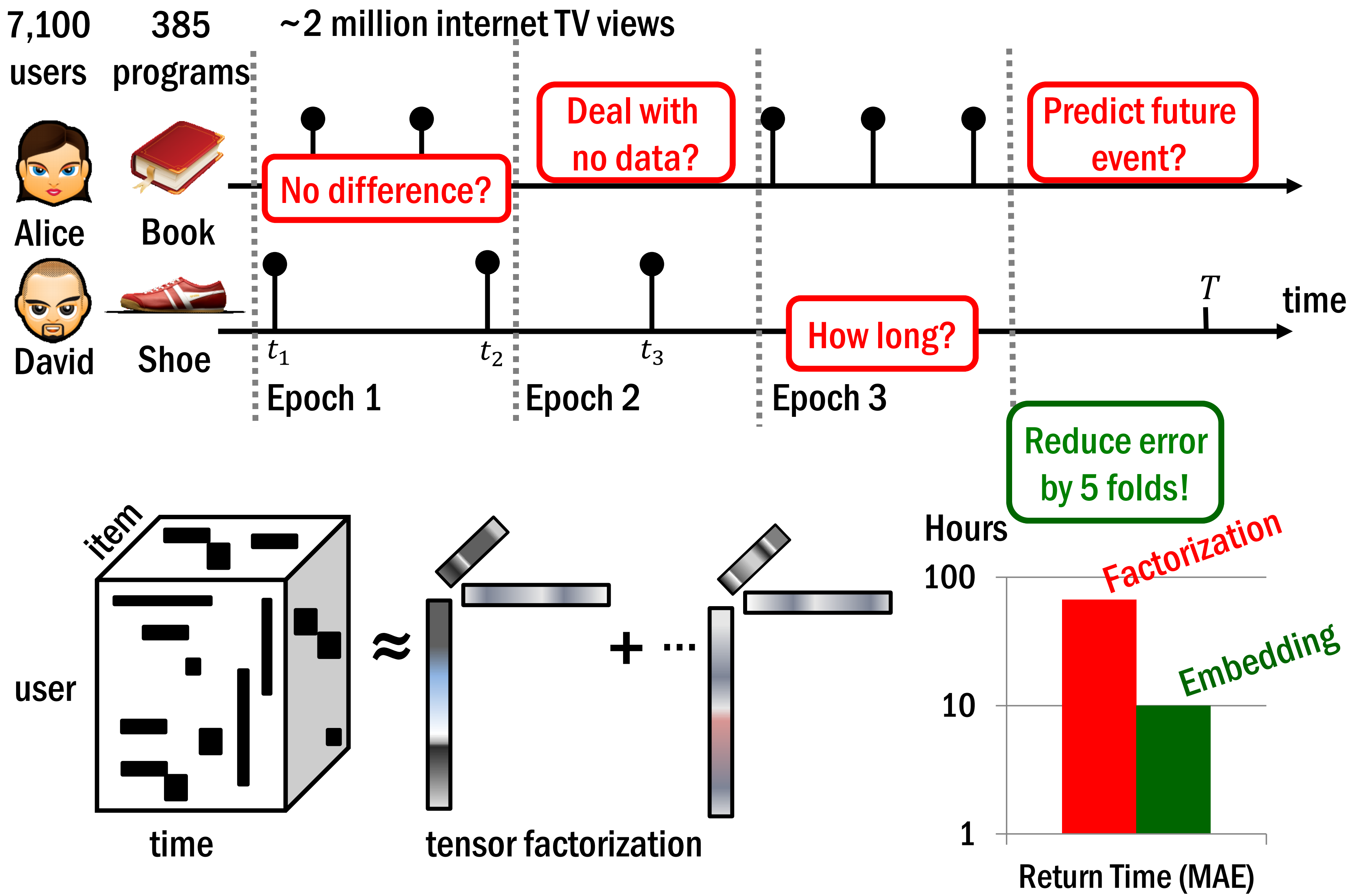
Ex 2: Social information network modeling



who and when
will do what?

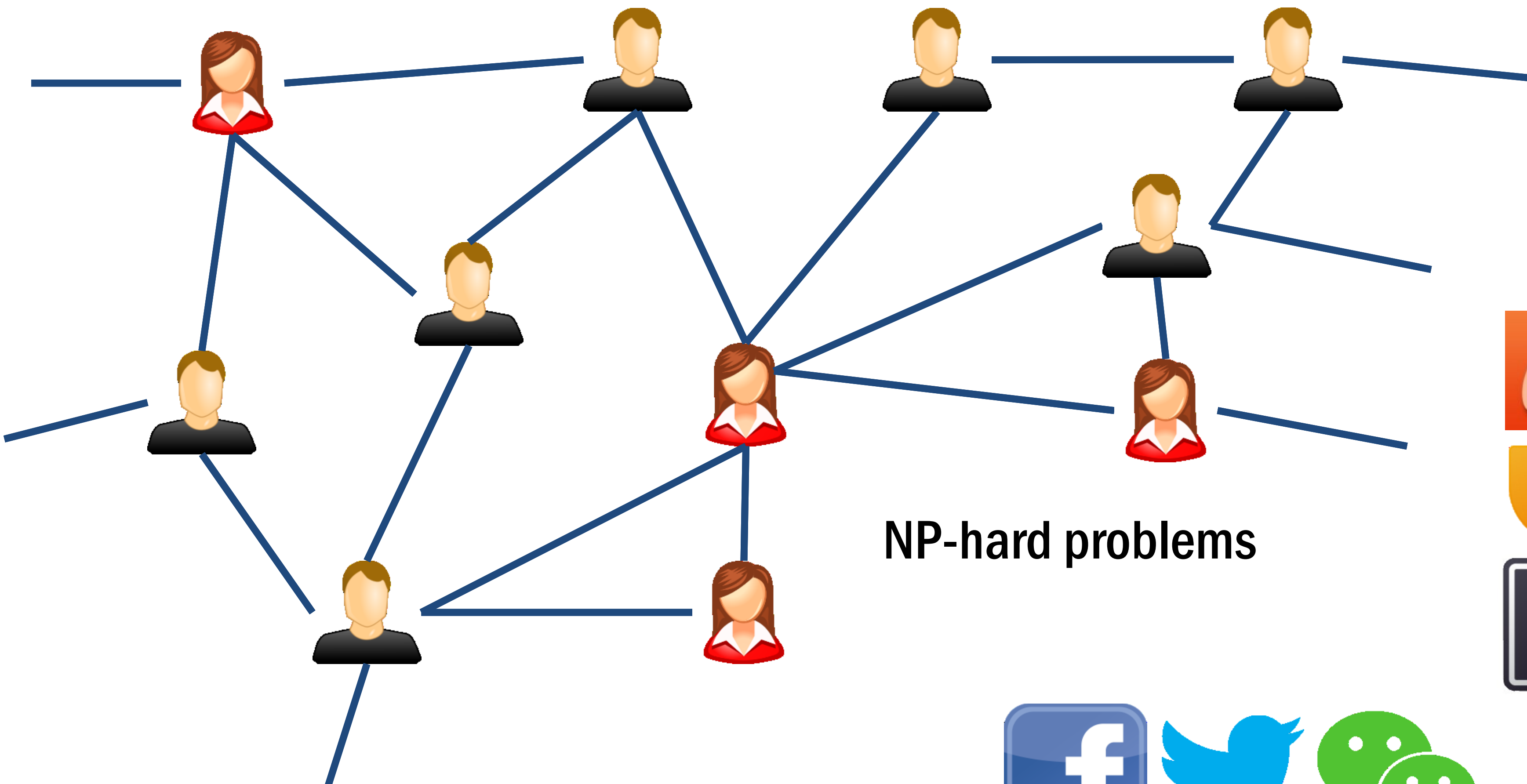
amazon spotify PANDORA last.fm reddit yelp WeChat

Complex behavior not well modeled



Ex 3: Combinatorial optimizations over graphs

Application	Optimization Problem
Influence maximization Community discovery Resource scheduling	Minimum vertex/set cover Maximum cut Traveling salesman



NP-hard problems

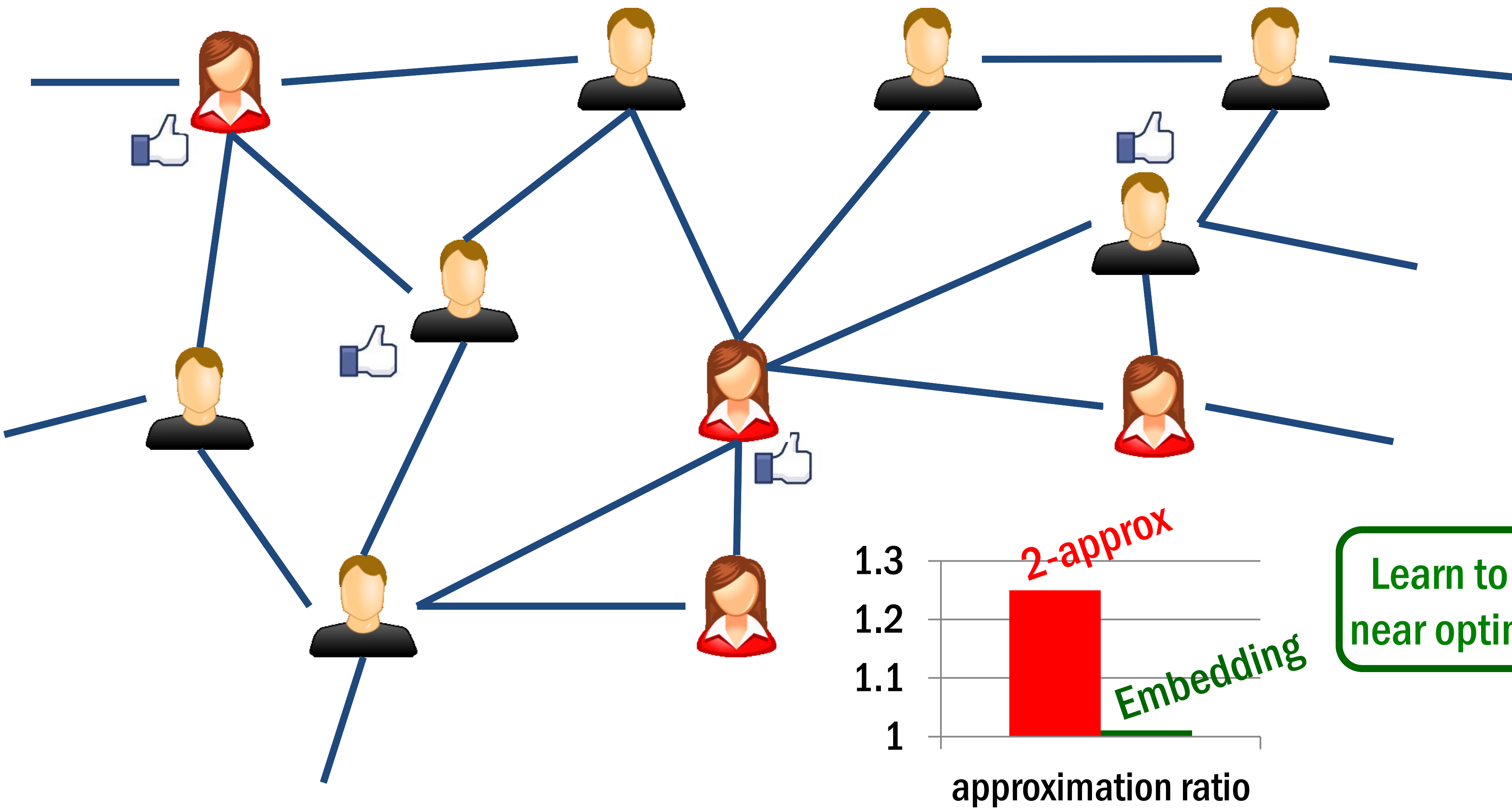
Simple heuristics do not exploit data

2 - approximation for minimum vertex cover

Repeat till all edges covered:

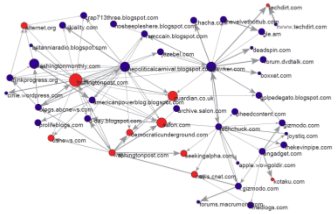
1. Select uncovered edge with largest total degree

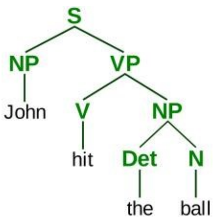
Decision not data-driven. Can we learn from data?

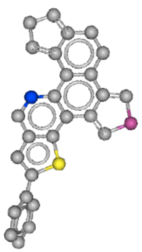


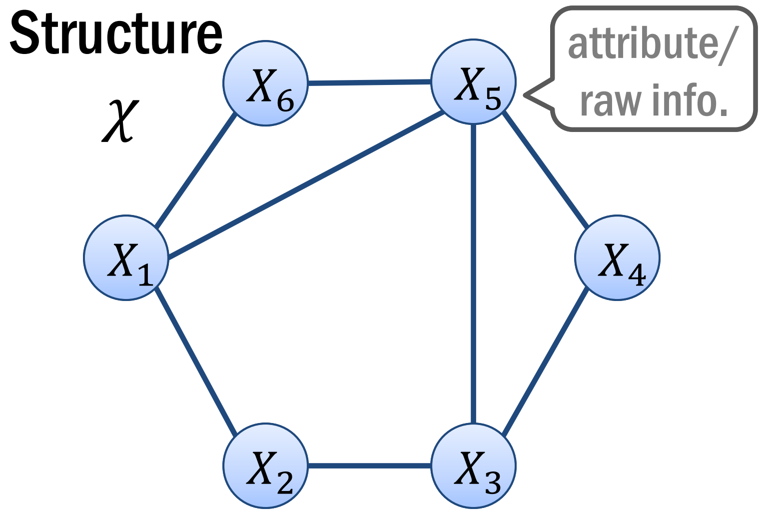
Learn to be near optimal!

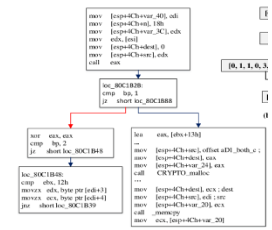
Fundamental problems

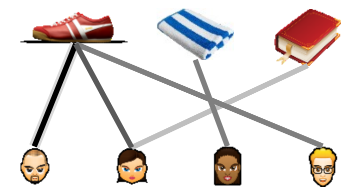


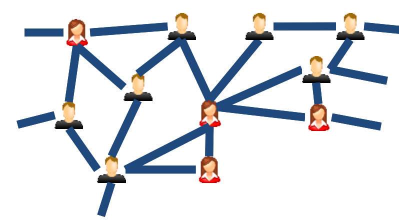












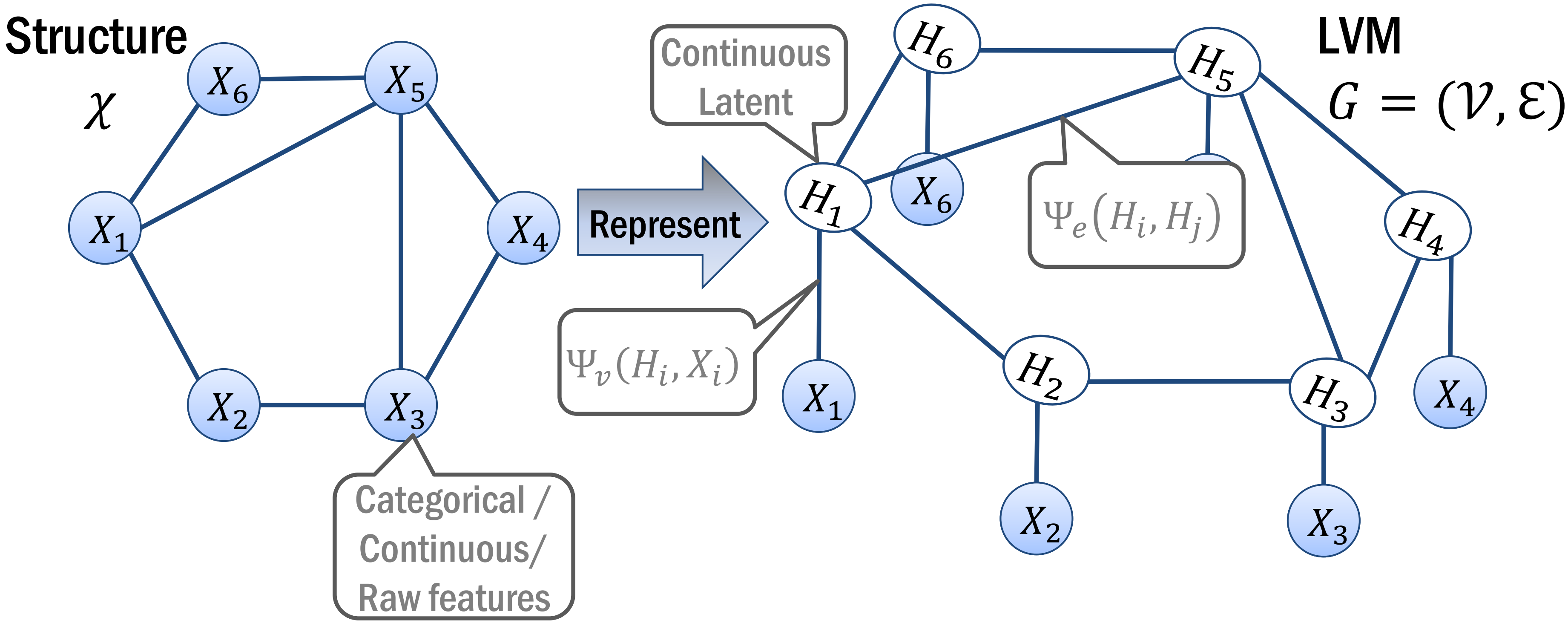
How to describe node?

How to describe entire structure?

How to incorporate various info.?

How to do it efficiently?

Represent structure as latent variable model (LVM)

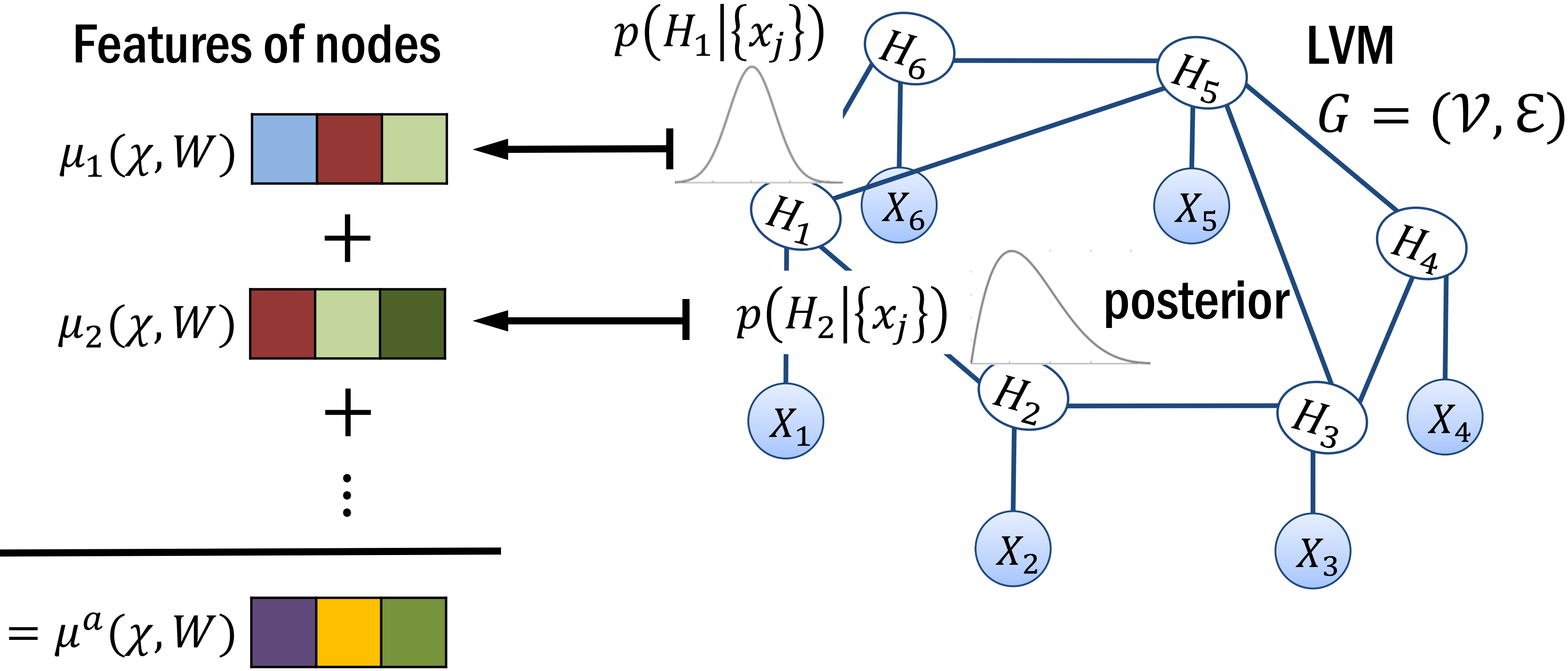


Joint likelihood

$$p(\{H_i\}, \{X_i\}) \propto \prod_{i \in \mathcal{V}} \underbrace{\Psi_v(H_i, X_i | \theta_v)}_{\text{Nonnegative node potential}} \prod_{(i,j) \in \mathcal{E}} \underbrace{\Psi_e(H_i, H_j | \theta_e)}_{\text{Nonnegative edge potential}}$$

[Dai, Dai & Song 2016]

Posterior distribution as features



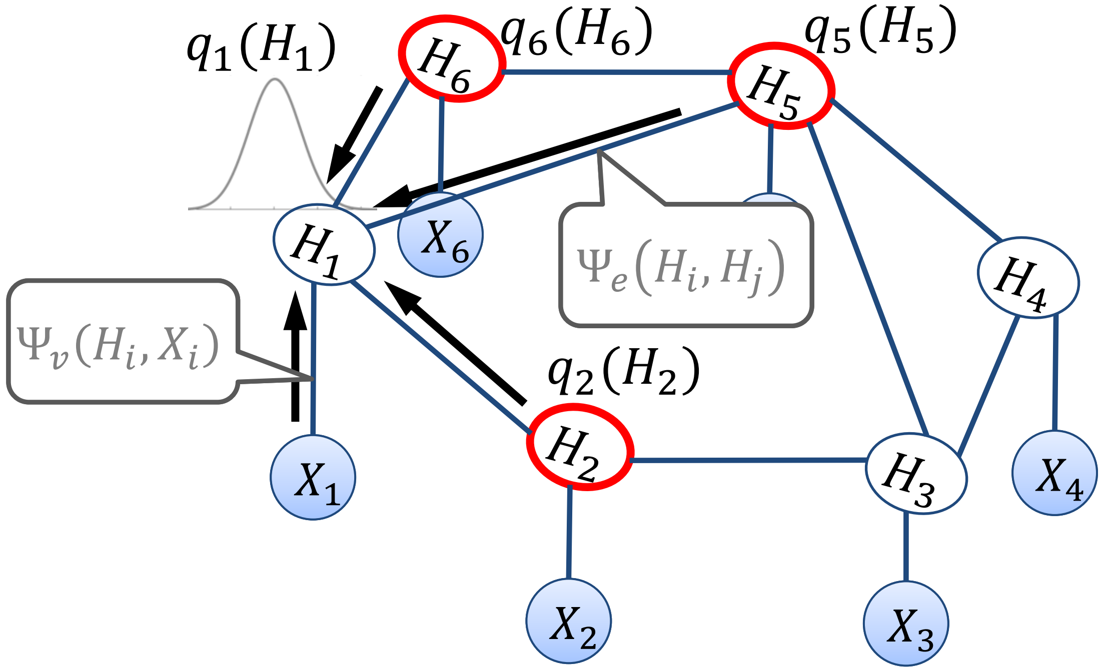
Features of the entire structure

$$p(H_i|\{x_j\}) = \frac{\sum_{all\ H_j\ except\ H_i} p(\{H_j\}, \{x_j\})}{p(\{x_j\})}$$

Capture both nodal and topological info.
Aggregate information from distant nodes

[Dai, Dai & Song 2016]

Mean field algorithm aggregates information



Approximate posterior

$$p(H_i | \{x_j\}) \approx q_i(H_i)$$

via fixed point update

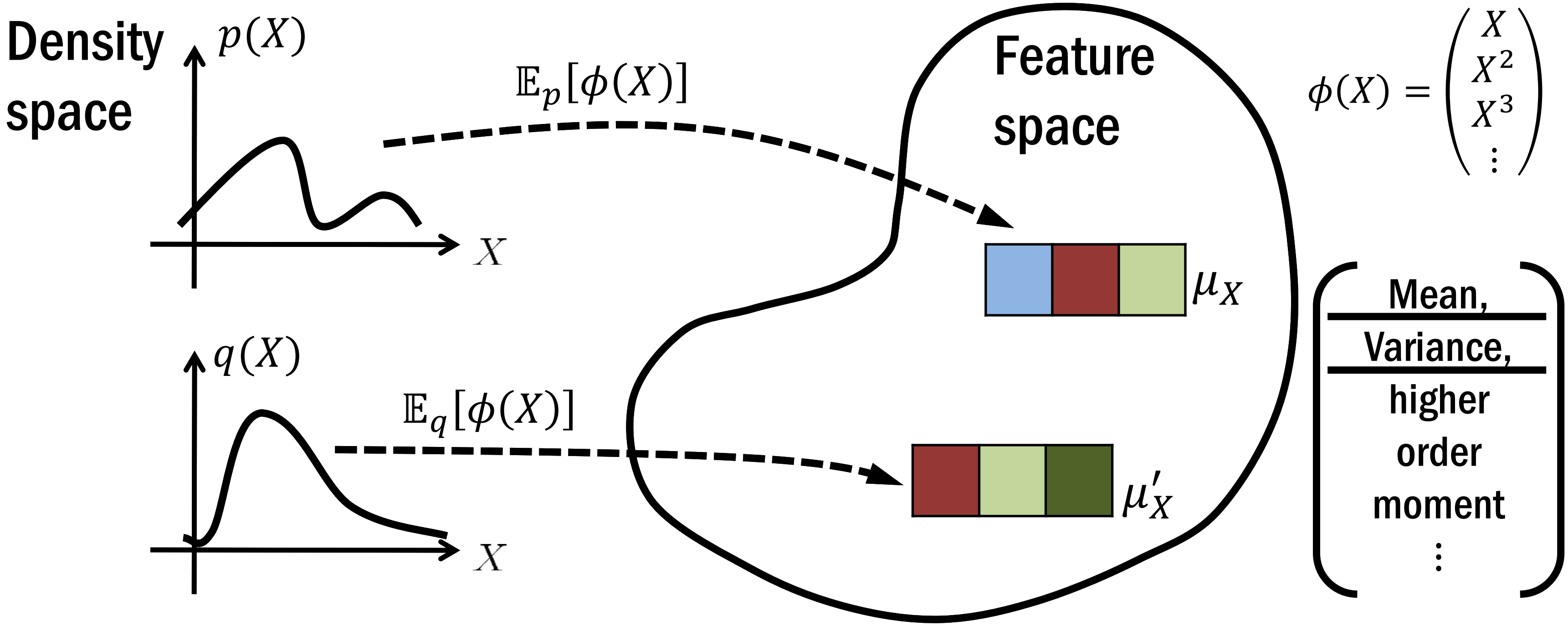
1. Initialize $q_i(H_i), \forall\, i$

2. Iterate many times

$$\underbrace{q_i(H_i) \leftarrow \Psi_v(H_i, X_i) \cdot \prod_{j \in \mathcal{N}(i)} \exp\left(\int_{\mathcal{H}} q_j(H_j) \log(\Psi_e(H_i, H_j))\, dH_j\right), \forall\, i}_{\boxed{\mathcal{T} \circ \left(X_i, \{q_j(H_j)\}_{j \in \mathcal{N}(i)}\right)}}$$

[Song et al. 11a,b]
[Song et al. 10a,b]

Embedding of distribution



Injective for rich nonlinear feature $\phi(x)$

μ_X is a sufficient statistic of $p(X)$

Operator View

$$\mathcal{T} \circ p(x) = \widetilde{\mathcal{T}} \circ \mu_X$$

[Smola, Gretton, Song & Scholkopf. 2007]

Structure2vec (S2V): embedding mean field

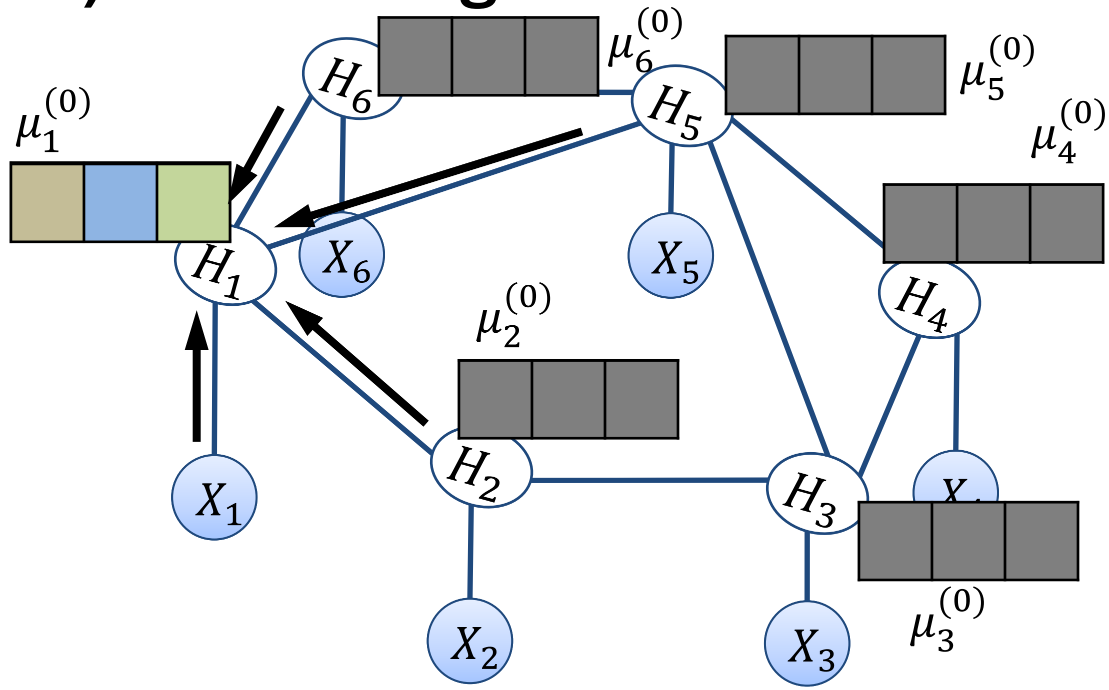
Approximate embedding of

$$p\left(H_i \middle| \{x_j\}\right) \mapsto \mu_i$$

via fixed point update

1. Initialize $\mu_i, \forall\, i$

2. Iterate many times



$$\boxed{\mu_i \leftarrow \widetilde{\mathcal{T}} \circ \left(X_i, \{\mu_j\}_{j \in \mathcal{N}(i)} \right), \forall\, i}$$

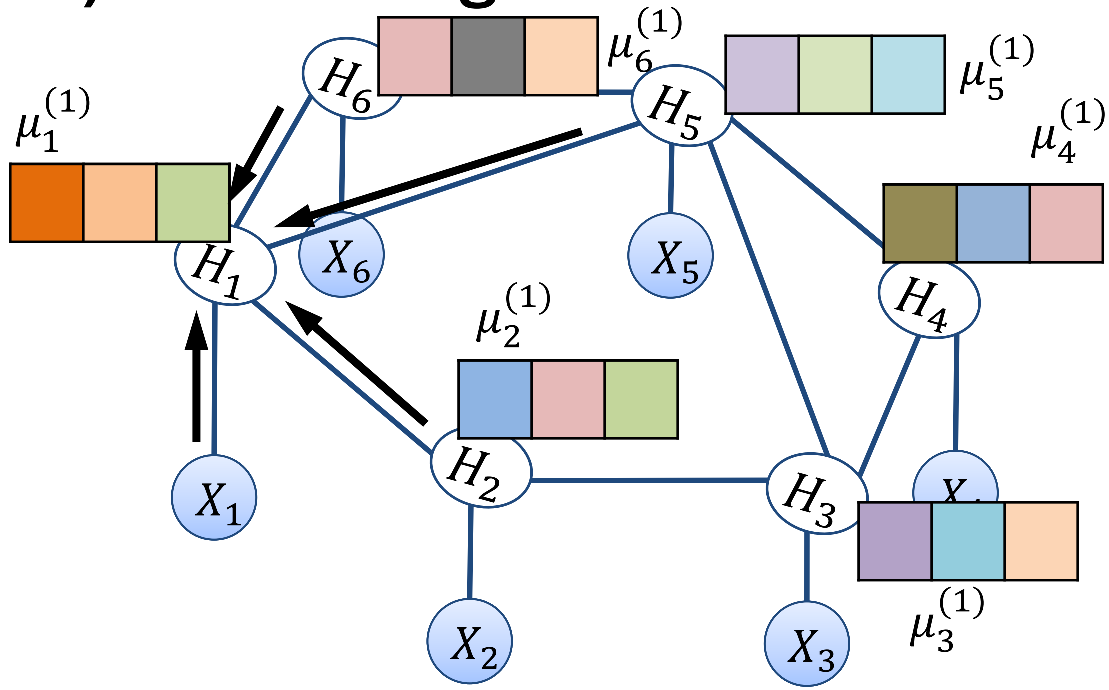
Structure2vec (S2V): embedding mean field

Approximate embedding of

$$p\left(H_i \middle| \{x_j\}\right) \mapsto \mu_i$$

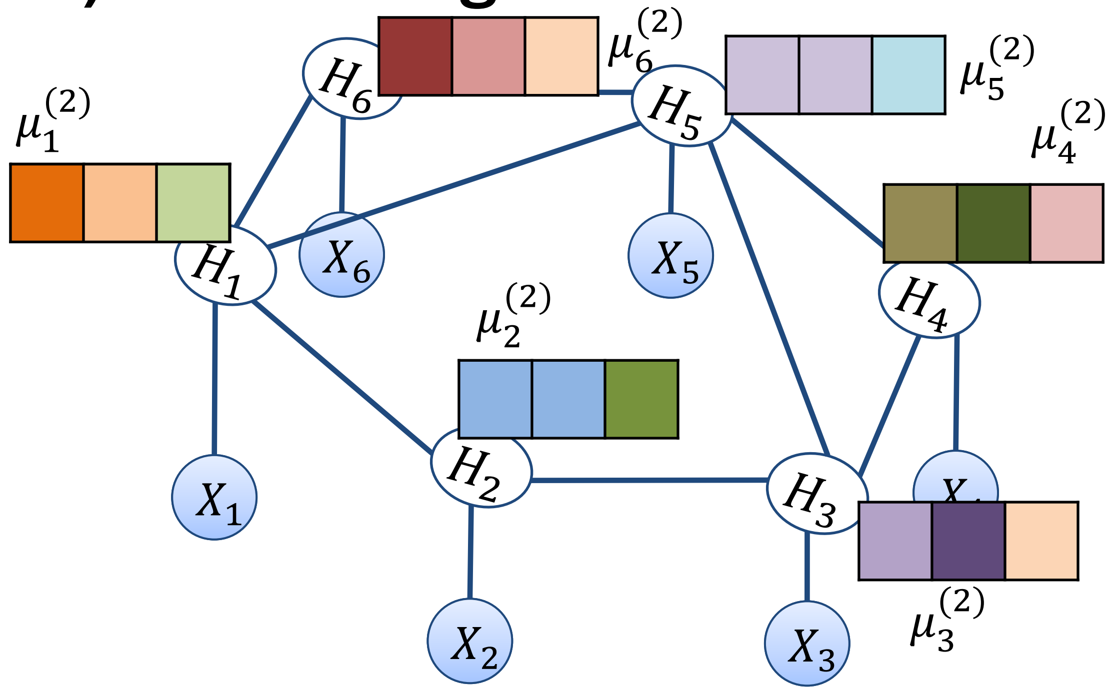
via fixed point update

1. Initialize $\mu_i, \forall\, i$
2. Iterate many times



$$\boxed{\mu_i \leftarrow \widetilde{\boldsymbol{\mathcal{T}}} \circ \left(X_i, \{\mu_j\}_{j \in \mathcal{N}(i)} \right), \forall\, i}$$

Structure2vec (S2V): embedding mean field



Approximate embedding of

$$p\left(H_i \middle| \{x_j\}\right) \mapsto \mu_i$$

via fixed point update

1. Initialize $\mu_i, \forall\, i$
2. Iterate many times

$$\mu_i \leftarrow \widetilde{\mathcal{T}} \circ \left(X_i, \{\mu_j\}_{j \in \mathcal{N}(i)} \right), \forall\, i$$

How to parametrize $\widetilde{\mathcal{T}}$?
Depends on unknown $\Psi_v(H_i, X_i)$ and $\Psi_e(H_i, H_j)$

Directly parameterize nonlinear mapping

$$\mu_i \leftarrow \widetilde{\mathcal{T}} \circ \left(X_i, \{\mu_j\}_{j \in \mathcal{N}(i)} \right)$$

Any universal nonlinear function will do

Eg. assume $\mu_i \in \mathcal{R}^d, X_i \in \mathcal{R}^n$, neural network parameterization

$$\mu_i \leftarrow \sigma \left(W_1 X_i + W_2 \sum_{j \in \mathcal{N}(i)} \mu_j \right)$$

$\max\{0,\cdot\}$
$\tanh(\cdot)$
sigmoid$(\cdot)$

$d \times n$ matrix

$d \times d$ matrix

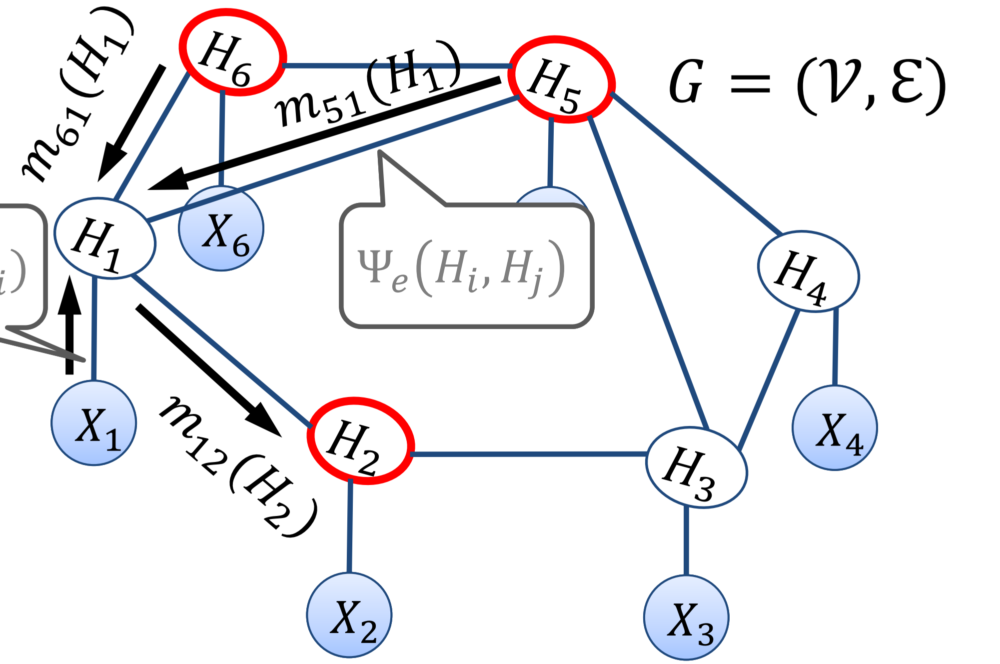
Learn with supervision, unsupervised learning, or reinforcement learning

Embedding belief propagation

Approximate $p(H_i | \{x_j\}, \theta)$ as

$$q_i(H_i) = \Psi_v(H_i, x_i | \theta) \cdot \underbrace{\prod_{j \in \mathcal{N}(i)} m_{ji}(H_i)}$$

$$\boxed{\mathcal{T}' \circ \left(X_i, \{m_{\ell i}(H_i)\}_{\ell \in \mathcal{N}(i)} \right)}$$



1. Initialize $m_{ij}(H_j), \forall i, j$

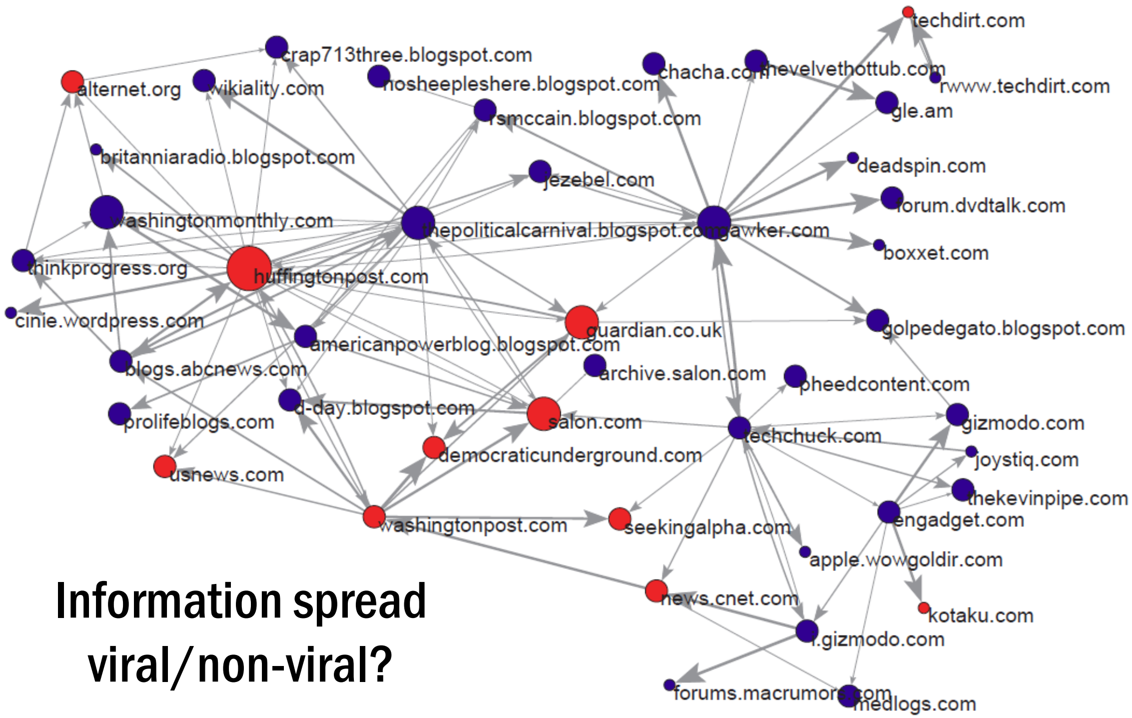
2. Iterate many times

$$m_{ij}(H_j) \leftarrow \underbrace{\int_{\mathcal{H}} \Psi_v(H_i, X_i | \theta) \Psi_e(H_i, H_j | \theta) \cdot \prod_{\ell \in \mathcal{N}(i) \setminus j} m_{\ell i}(H_i) \, dH_i}, \forall i, j$$

$$\boxed{\mathcal{T} \circ \left(X_i, \{m_{\ell i}(H_i)\}_{\ell \in \mathcal{N}(i) \setminus j} \right)}$$

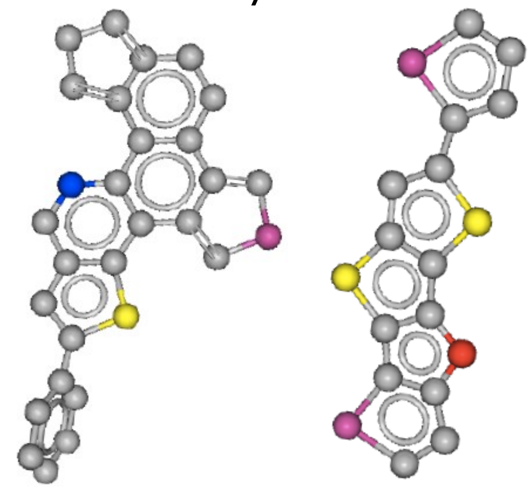
[Song et al. 11a,b]
[Song et al. 10a,b]

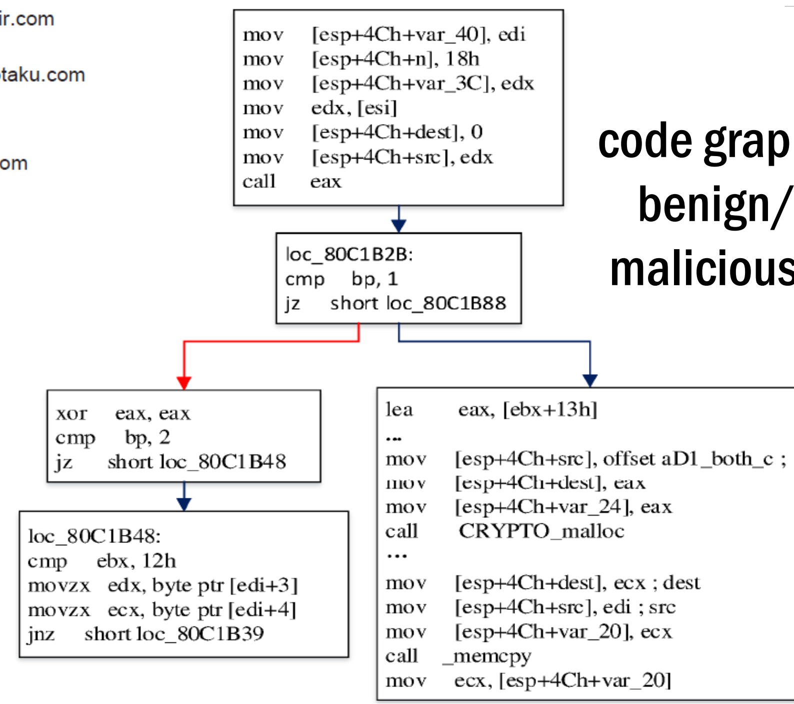
Ex 1: Prediction for structured data



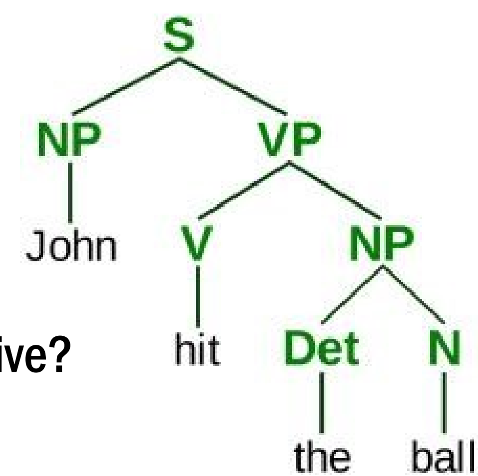
Information spread viral/non-viral?

Drug/materials effective/ineffective?





code graphs benign/malicious?



Natural language positive/negative?

Algorithm learning

Given m data points $\{\chi_1, \chi_2, \dots, \chi_m\}$

And their labels $\{y_1, y_2, \dots, y_m\}$

Estimate parameters W and V via

$$\min_{V,W} \; L(V,W) := \sum_{i=1}^{m} (y_i - V^\top \mu^a(W, \chi_i))^2$$

Computation	Operation	Similar to
Objective $L(V,W)$	A sequence of nonlinear mappings over graph	Graphical model inference
Gradient $\frac{\partial L}{\partial W}$	Chain rule of derivatives in reverse order	Back propagation in deep learning

10,000x smaller model but accurate prediction

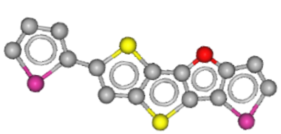
Harvard clean energy project: predict material efficiency (0-12)

2.3 million organic molecules

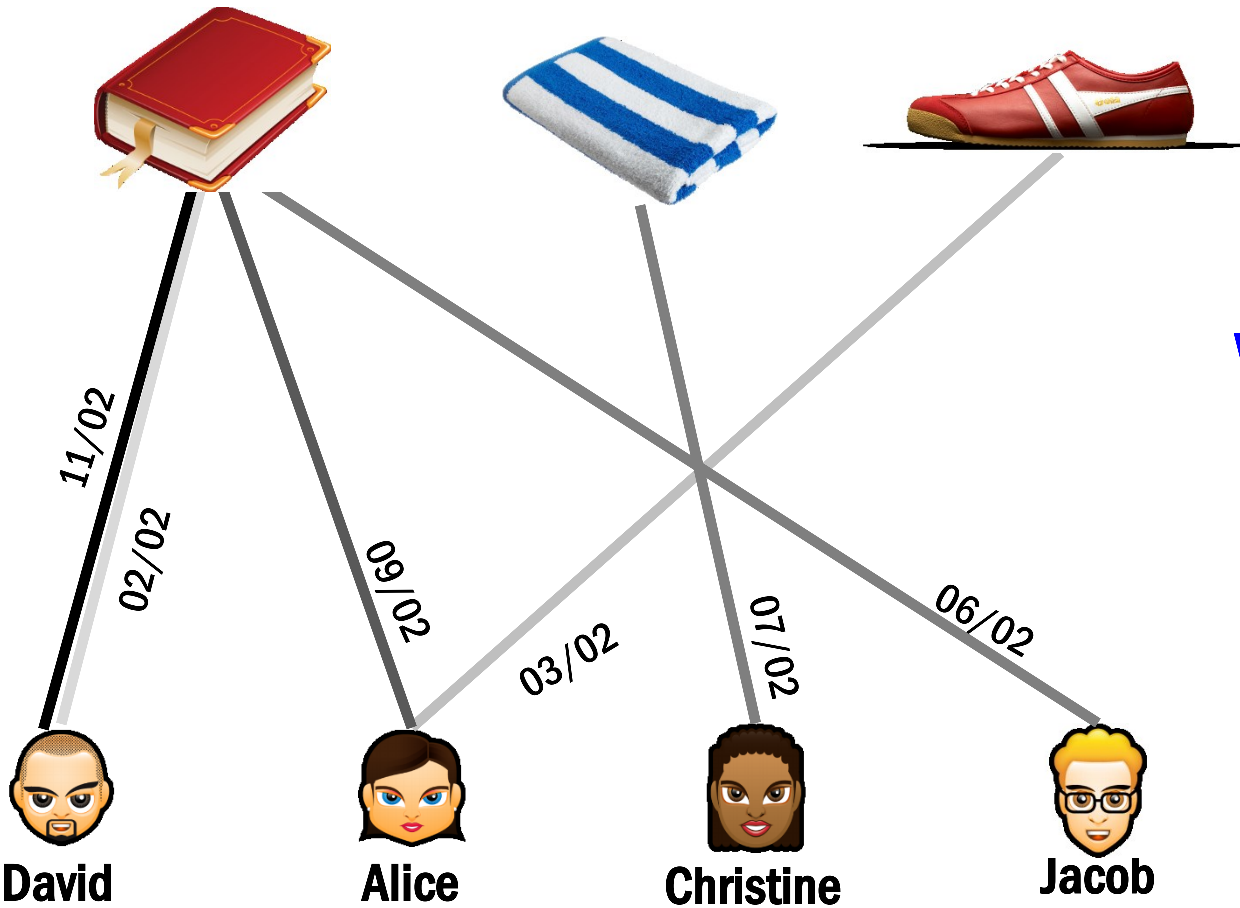
90% for training, 10% data for testing

	Test MAE	Test RMSE	# parameters
Mean predictor	1.986	2.406	1
WL level-3	0.143	0.204	1.6 m
WL level-6	0.096	0.137	1.3 b
S2V-MF	0.091	0.125	0.1 m
S2V-BP	0.085	0.117	0.1 m

~4% relative error

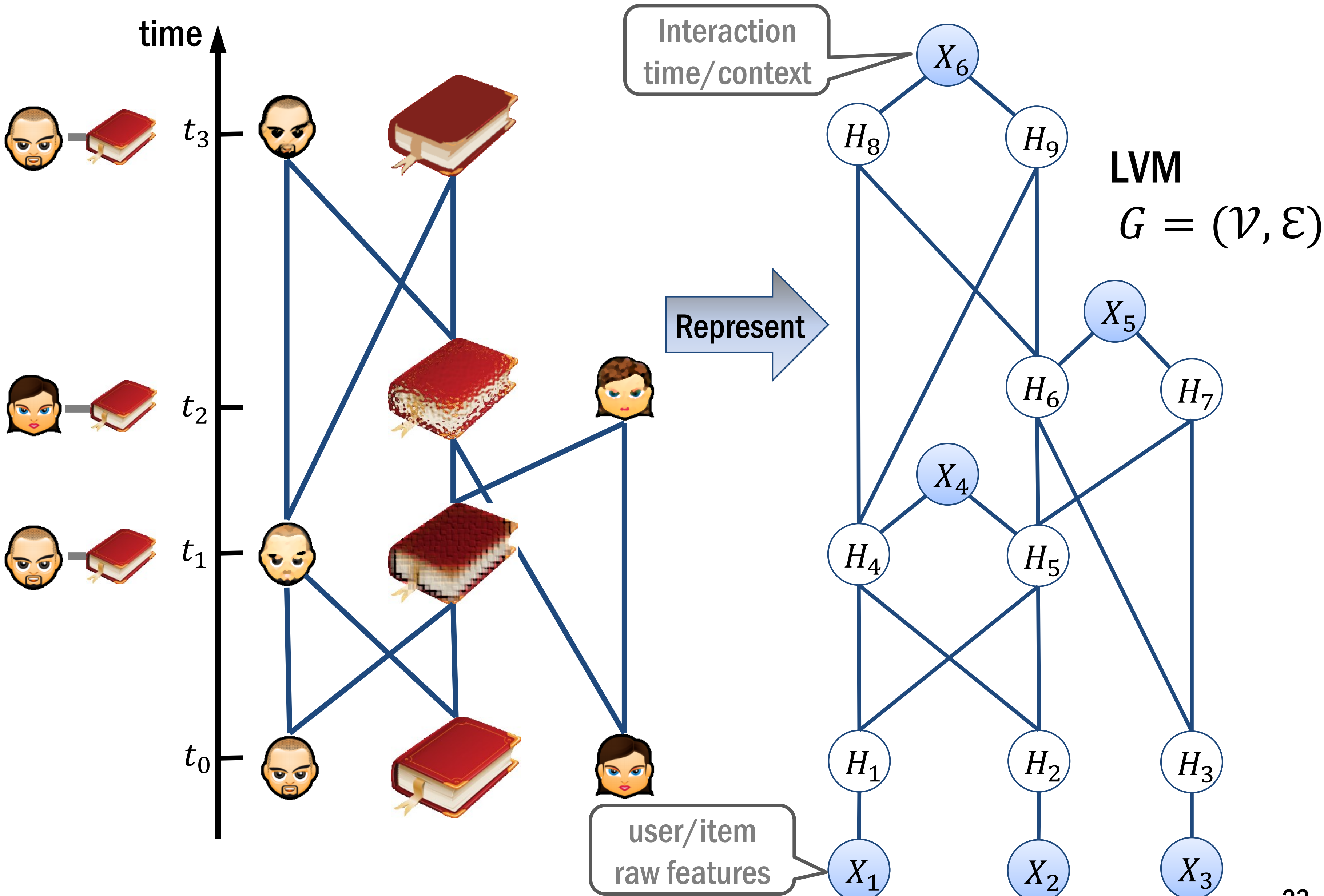


Ex 2: Social information network modeling

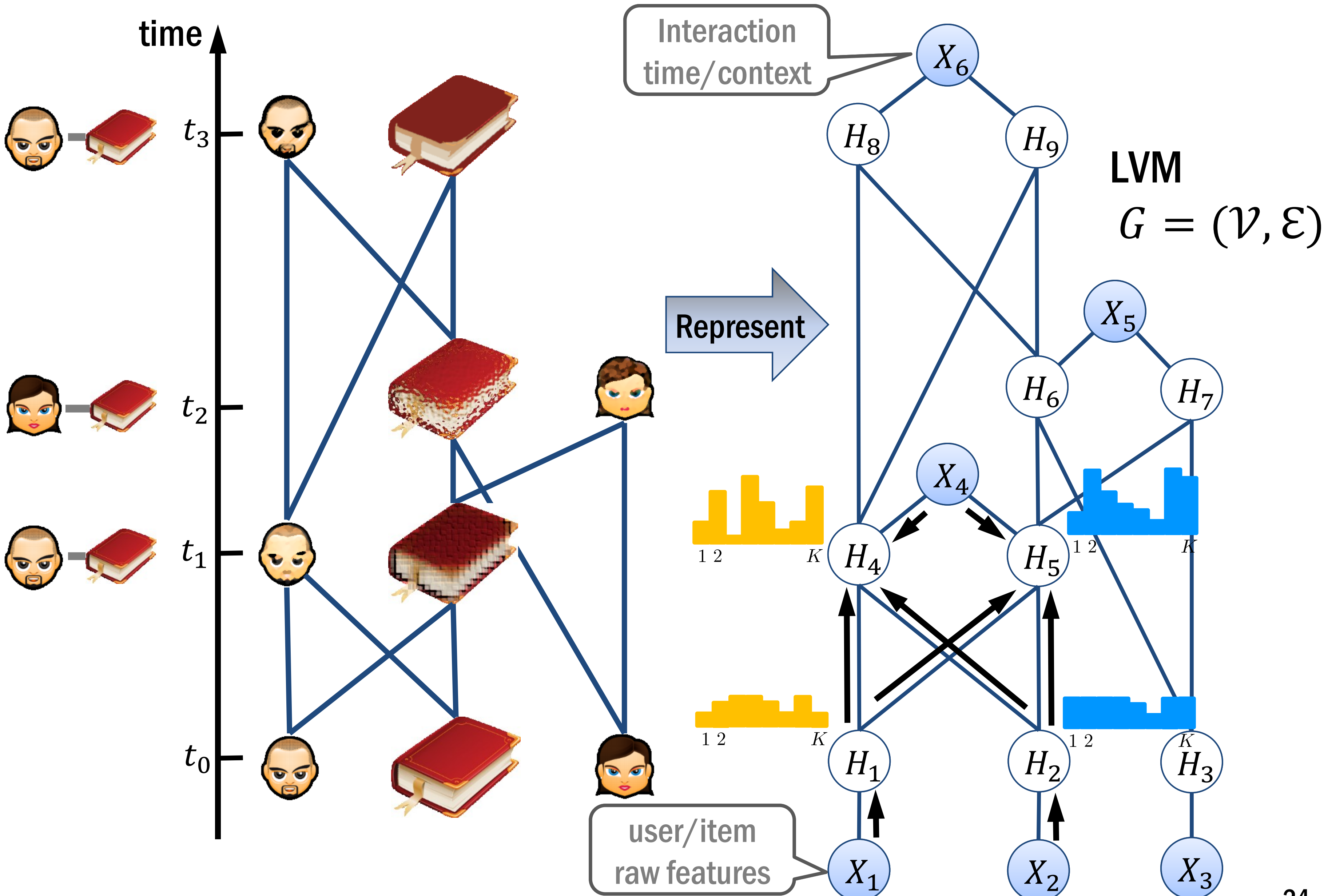


who and when will do what?

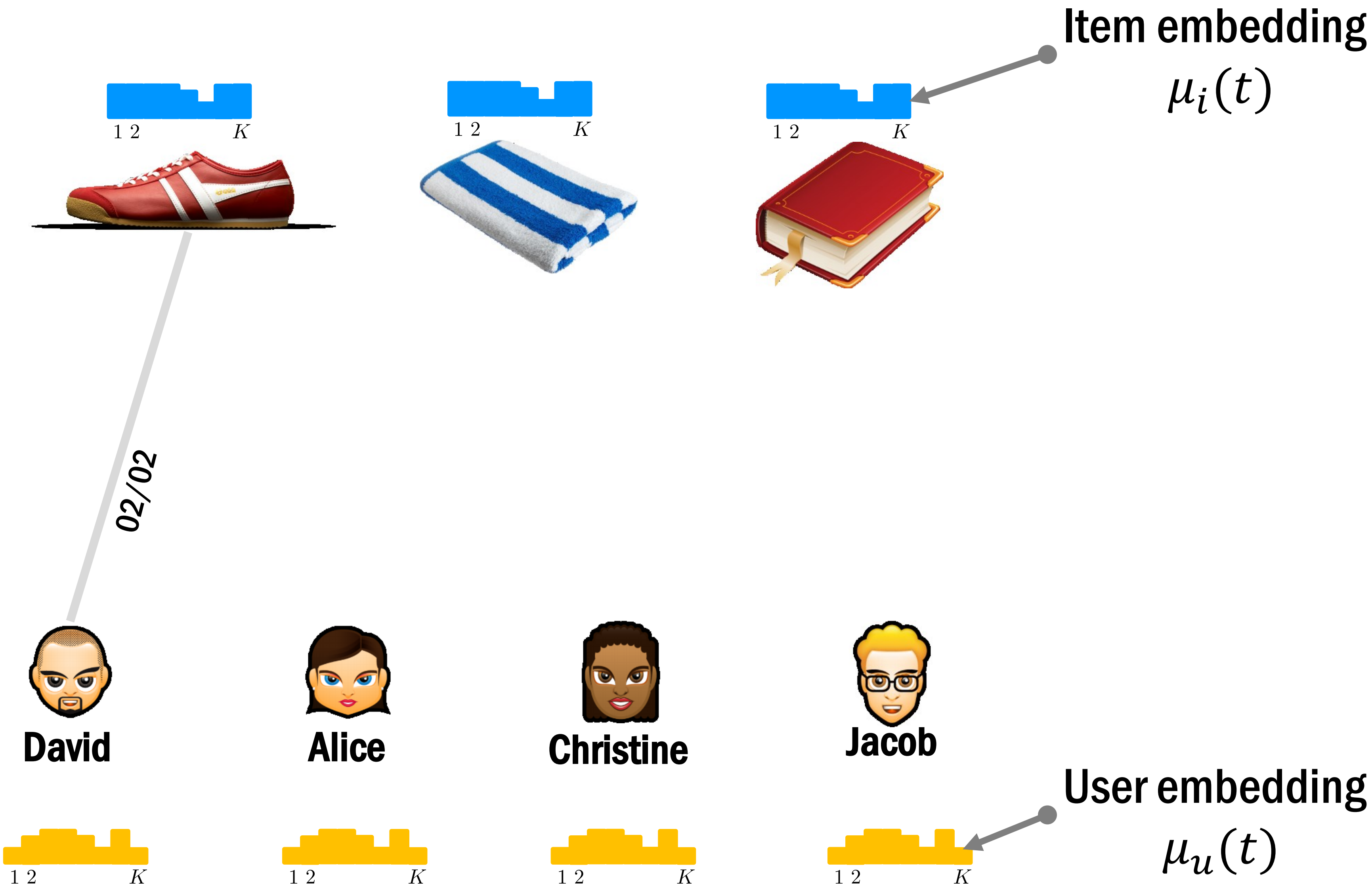
Unroll: time-varying dependency structure



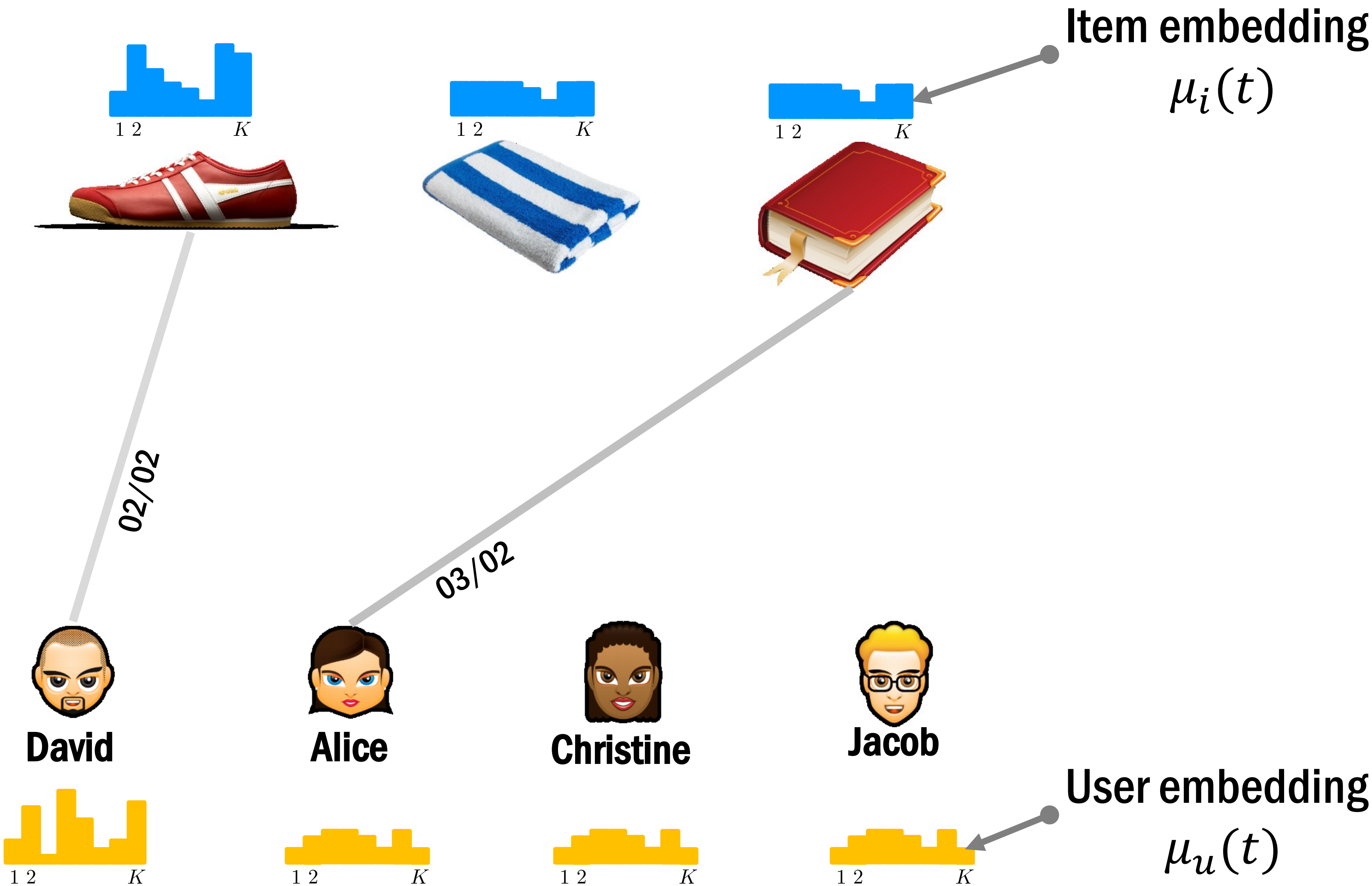
Embed filtering/forward belief propagation



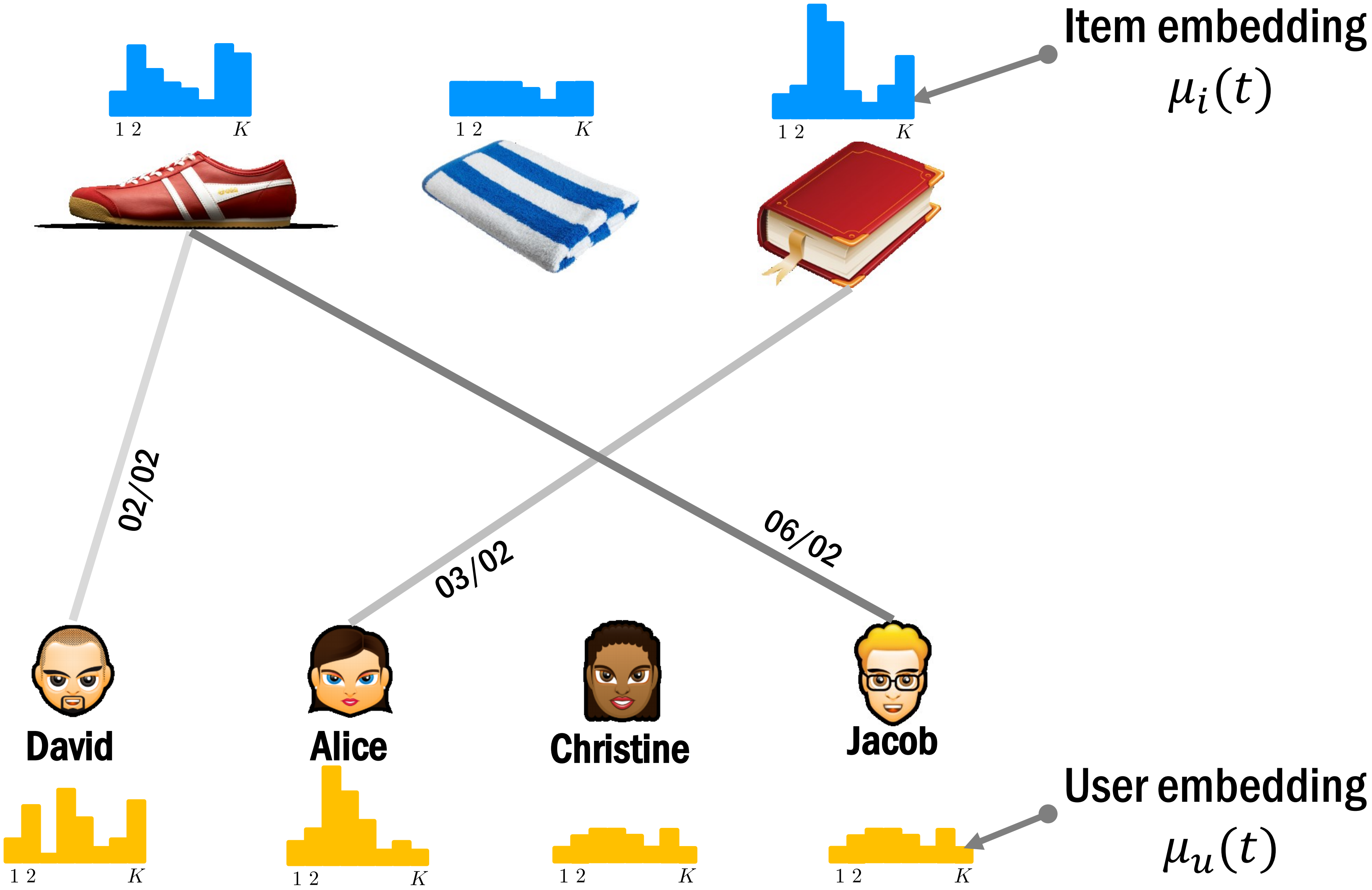
Co-evolutionary embedding



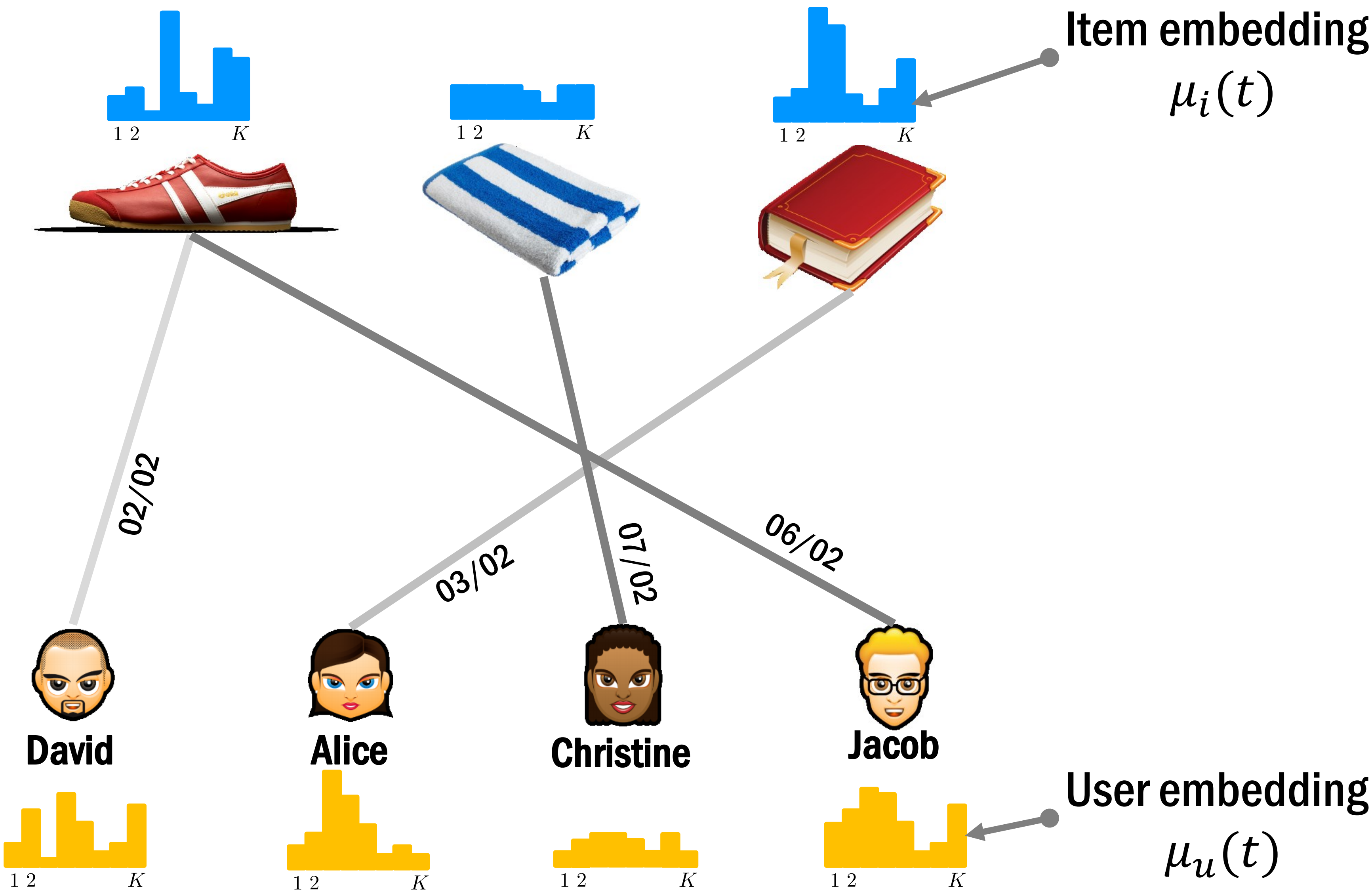
Co-evolutionary embedding



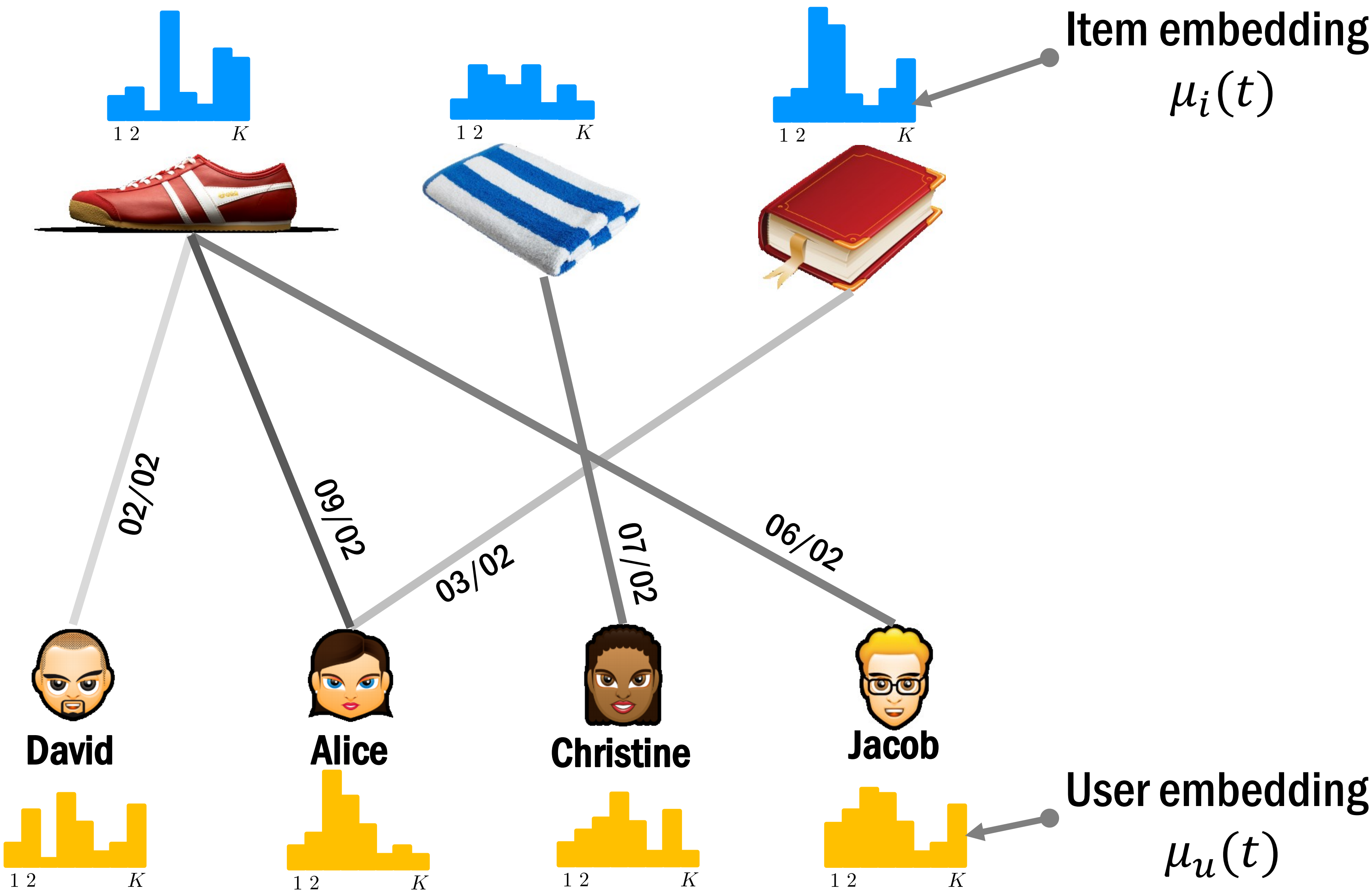
Co-evolutionary embedding



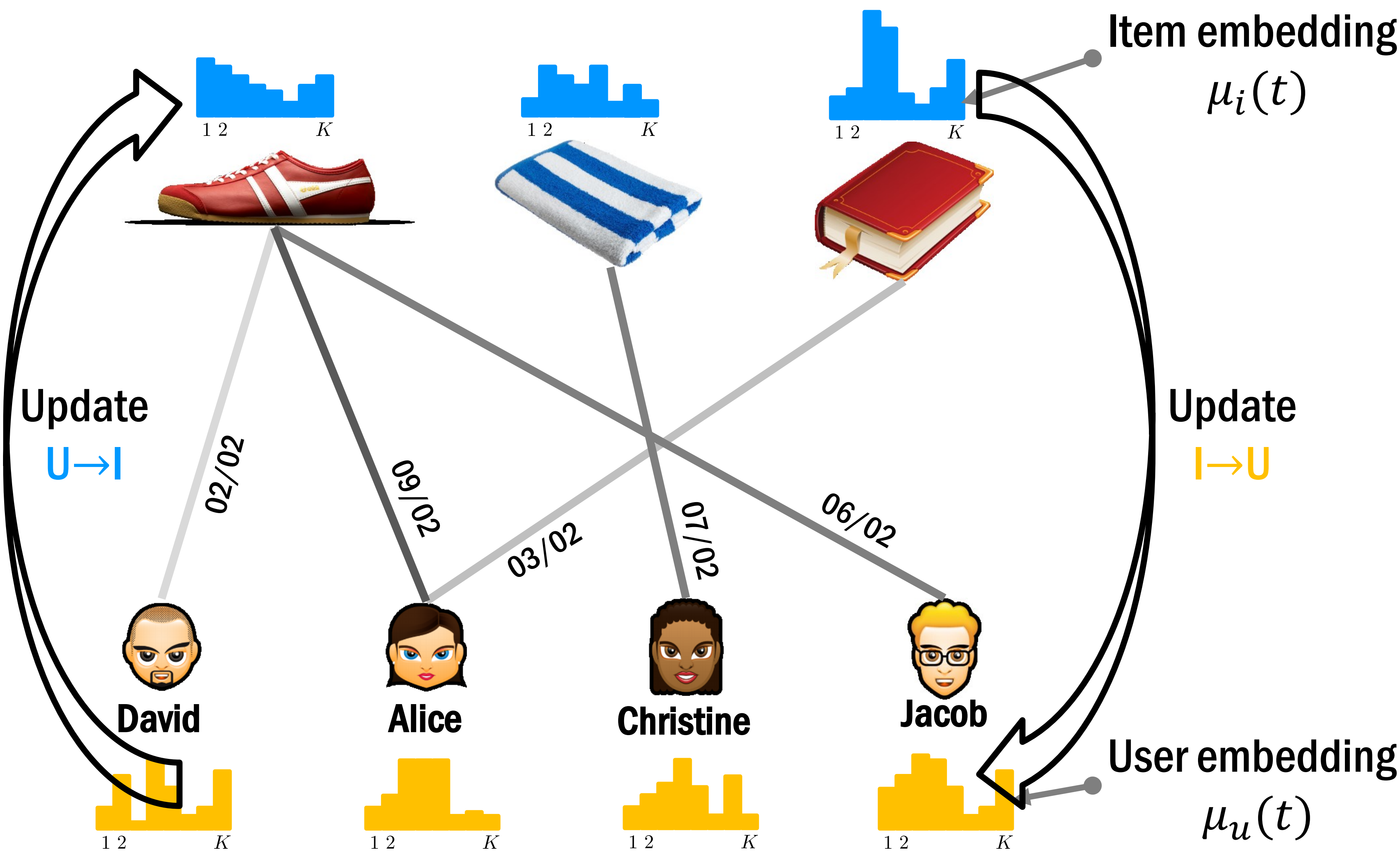
Co-evolutionary embedding



Co-evolutionary embedding

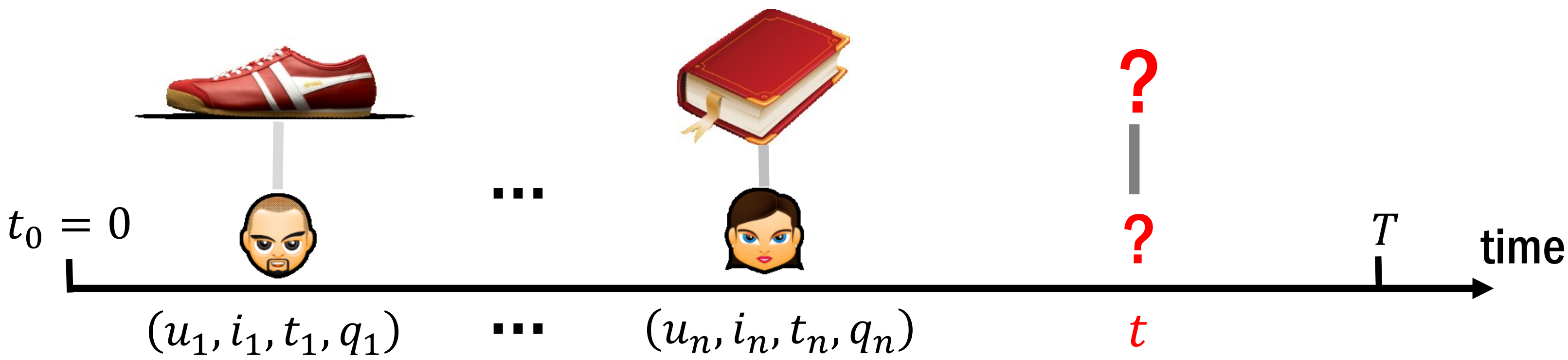


Co-evolutionary embedding



From embedding to next interaction time

Link embedding with interaction data using generative model



Intensity of interaction determined by compatibility and time-lapse

$$\lambda_{ui}(t|t_n) = \boxed{\exp\left(\mu_u^\top(t_n)\mu_i(t_n)\right)} \cdot \boxed{(t - t_n)}$$

Density function

$$p_{ui}(t|t_n) = \lambda_{ui}(t|t_n)\, S_{ui}(t|t_n)$$

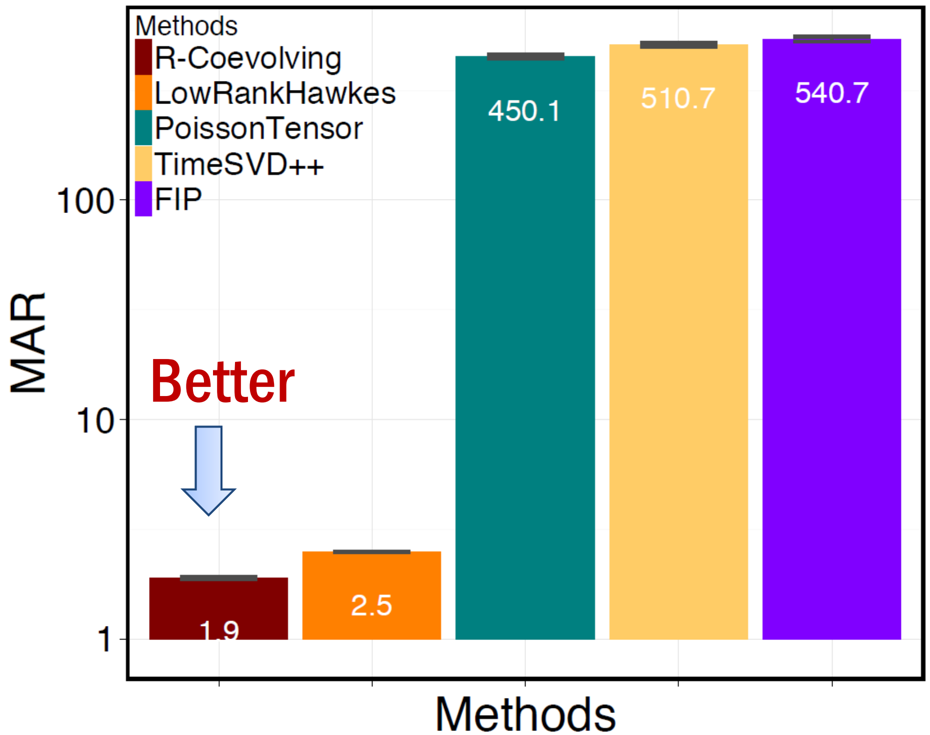
Survival function

$$S_{ui}(t|t_n) = \exp\left(-\int_{t_n}^{t} \lambda_{ui}(\tau)d\tau\right)$$

Embedding leads to better prediction

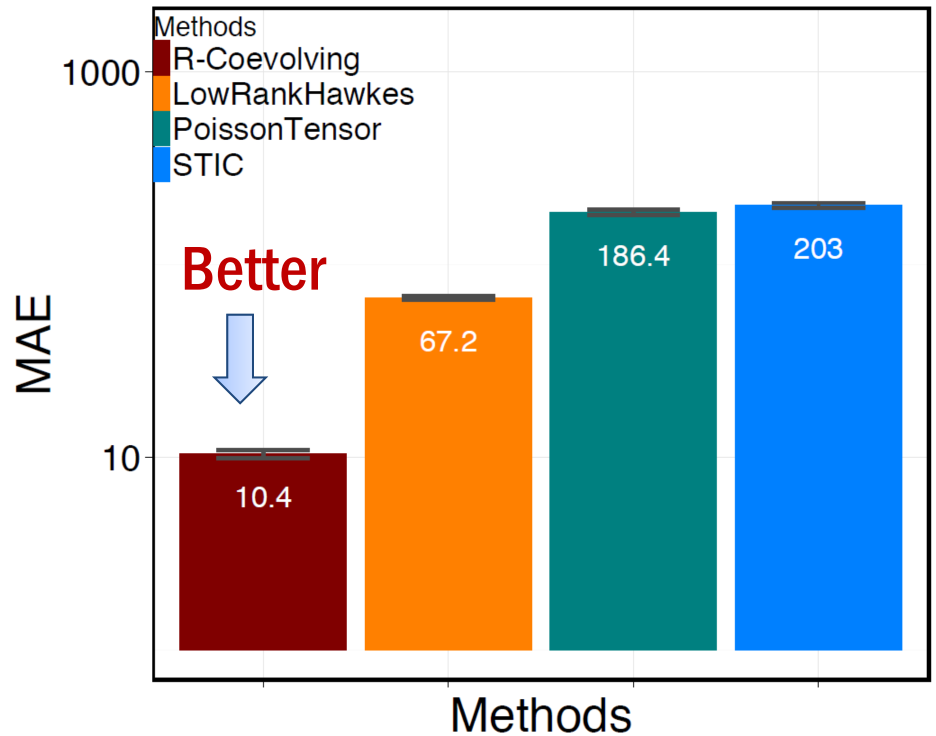
Reddit dataset: prediction of discussion forum participation

1,000 users, 1403 groups, ~10K interactions



Next item prediction

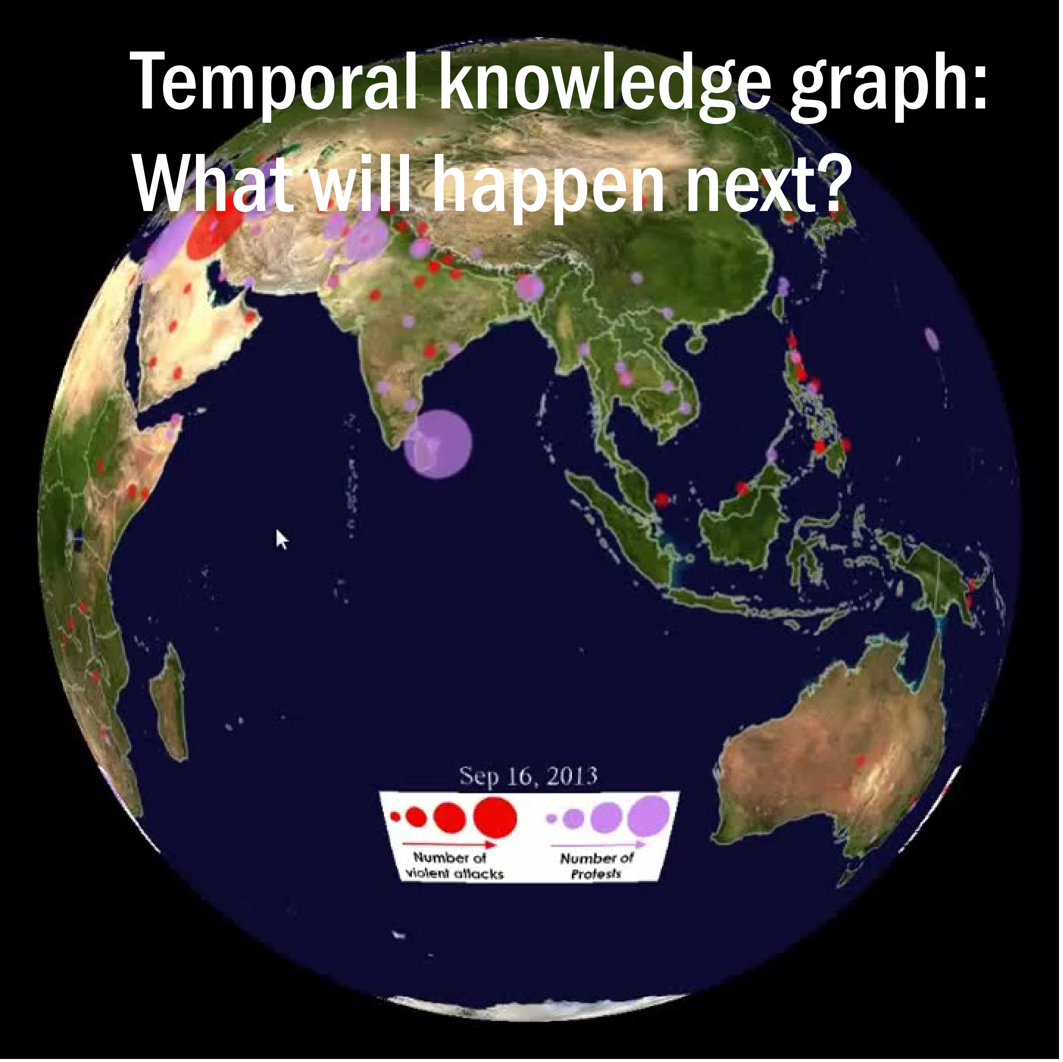
MAR: mean absolute rank difference



Return time prediction

MAE: mean absolute error (hours)

Temporal knowledge graph: What will happen next?



GDELT database:

Events in news media

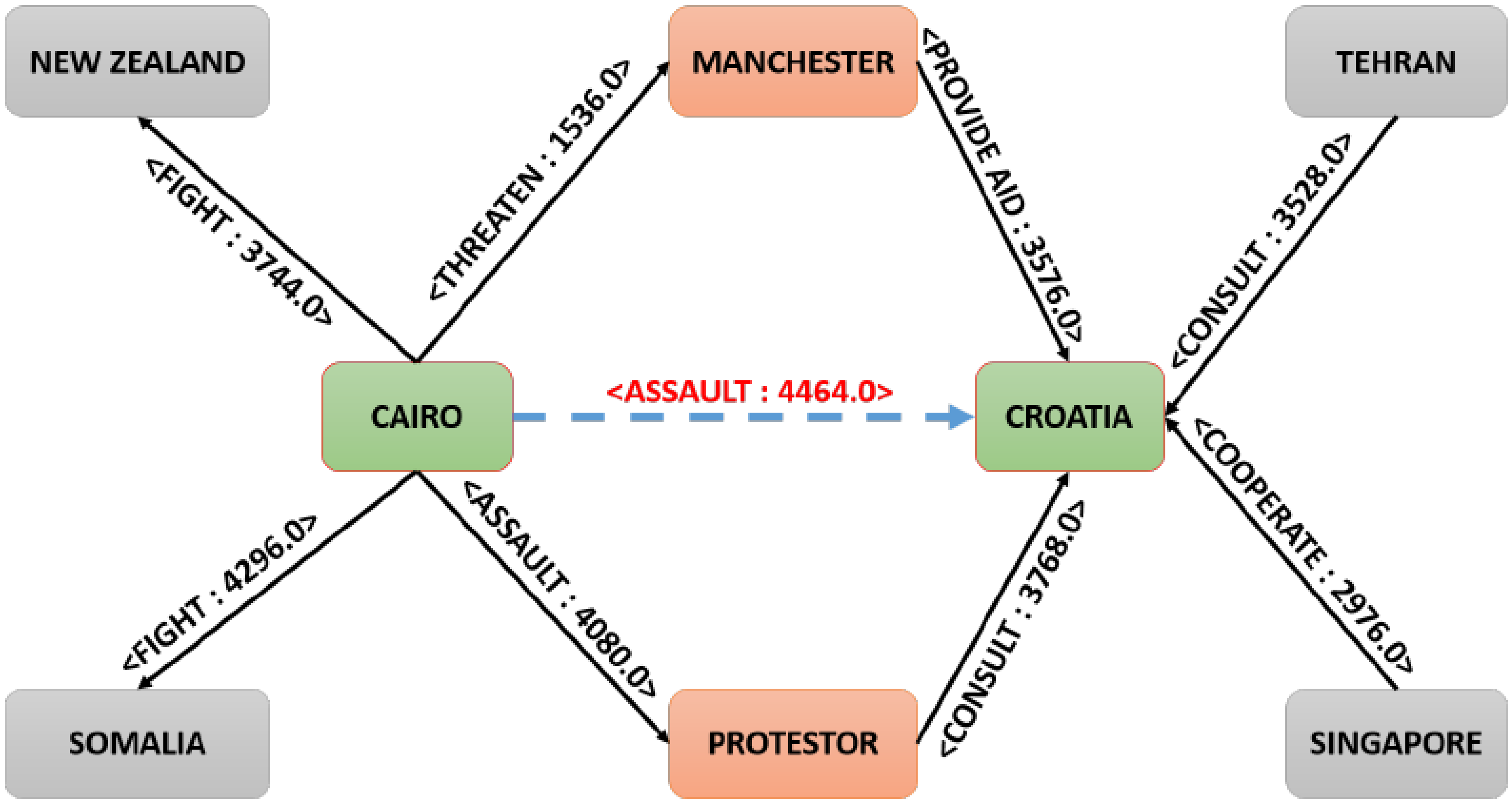
Total archives span >215 years, trillion of events

Event (knowledge item):

- Subject --- relation --- object
- Time

Reasoning over time I

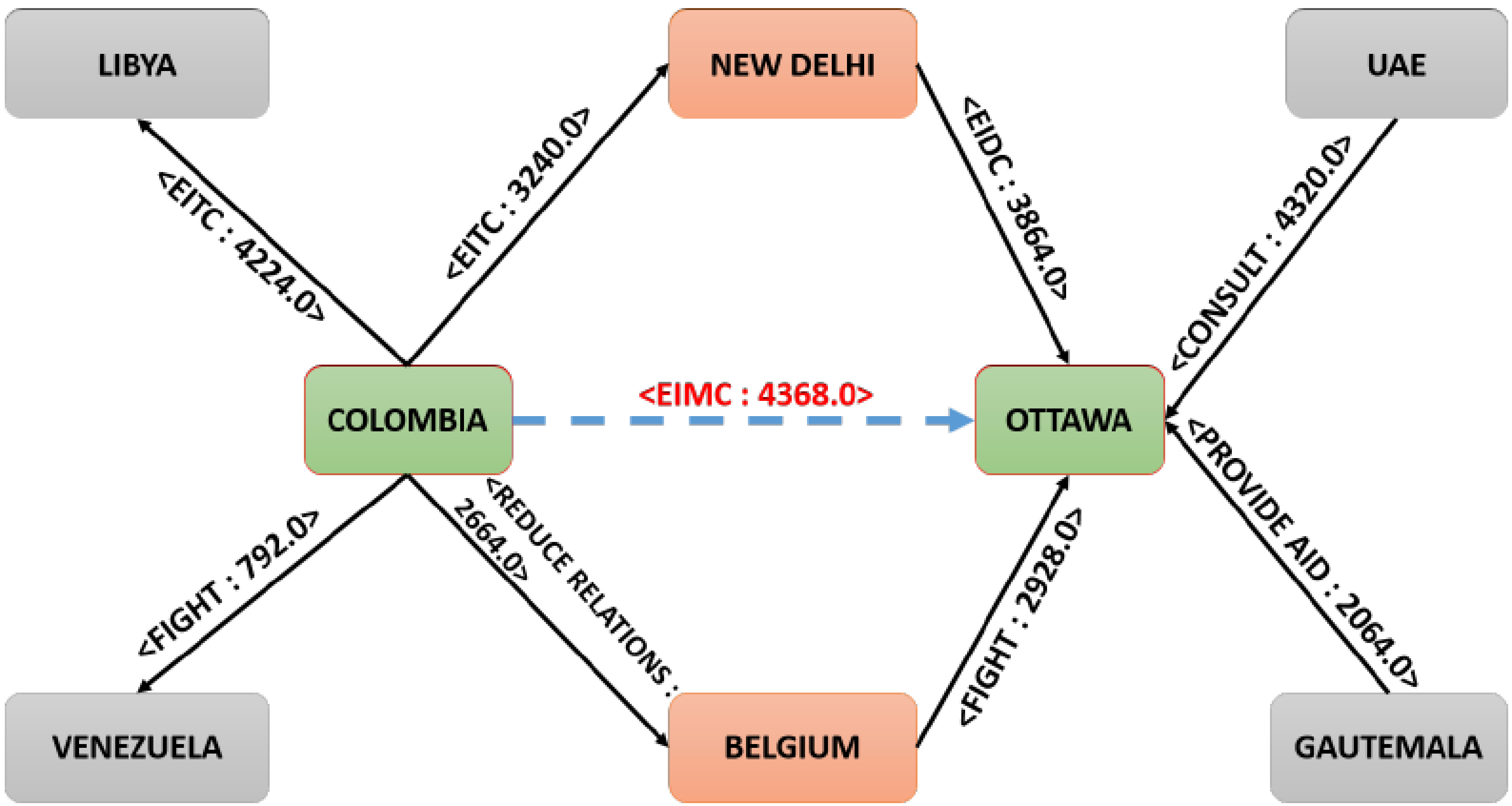
Enemy's friend is enemy



Reasoning over time II

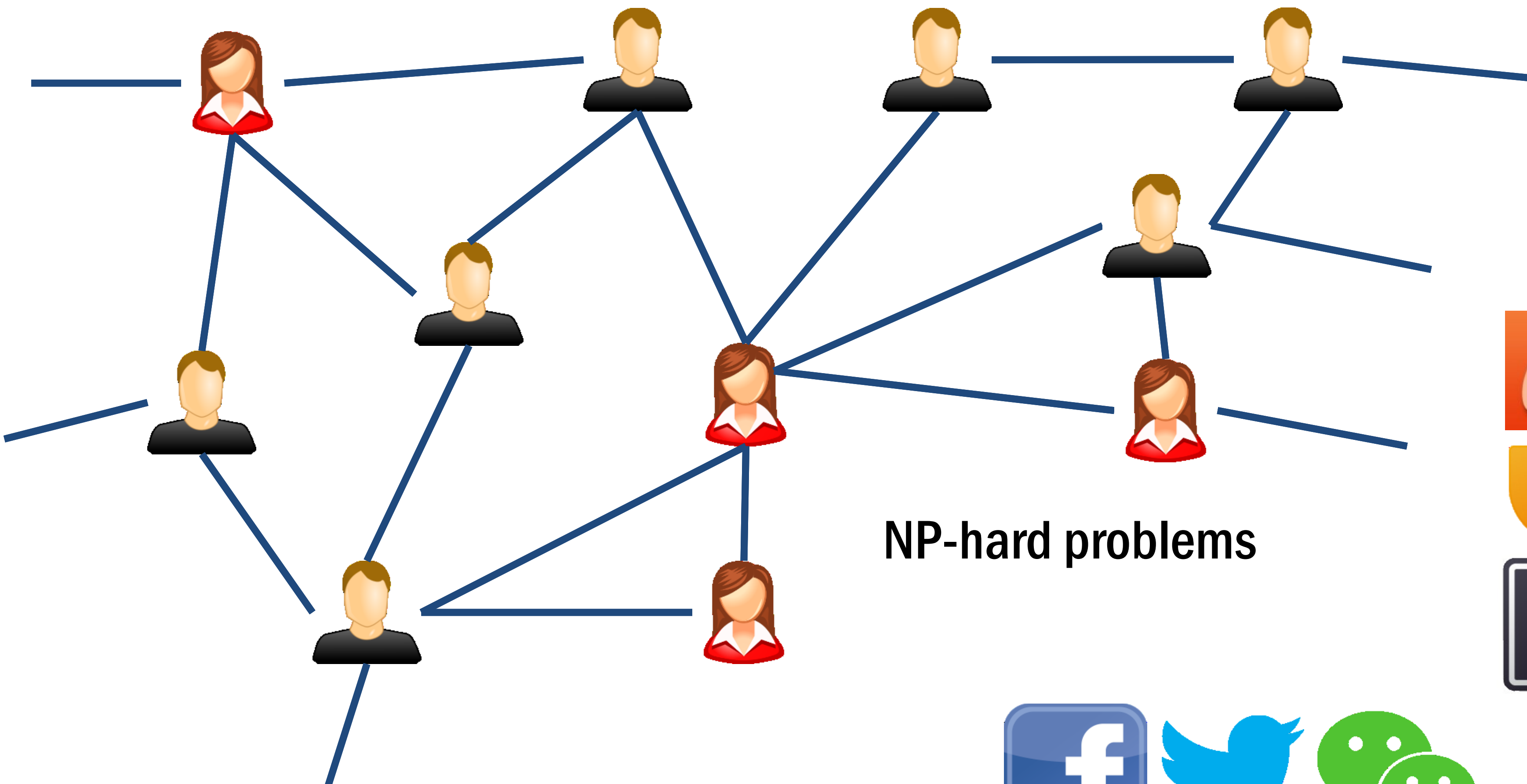
Friends' friend is a friend, common enemy improves bond

EITC / EIDC / EIMC: some form of cooperation



App 3: Combinatorial optimizations over graphs

Application	Optimization Problem
Influence maximization Community discovery Resource scheduling	Minimum vertex/set cover Maximum cut Traveling salesman



NP-hard problems

Combinatorial optimization as MDP

Minimum vertex cover: smallest number of nodes to cover all edges

$$\min_{x_i \in \{0,1\}} \sum_{i \in \mathcal{V}} x_i$$

$$s.t.\, x_i + x_j > 0, \forall (i,j) \in \mathcal{E}$$

Repeat:

1. Compute total degree of each uncovered edge
2. Select both ends of uncovered edge with largest total degree

Until all edges are covered

multistage decision making problem

$r^t = \sum_{i \in \mathcal{V}} x_i^t - \sum_{i \in \mathcal{V}} x_i^{t+1} = -1$

State S: current set of nodes selected

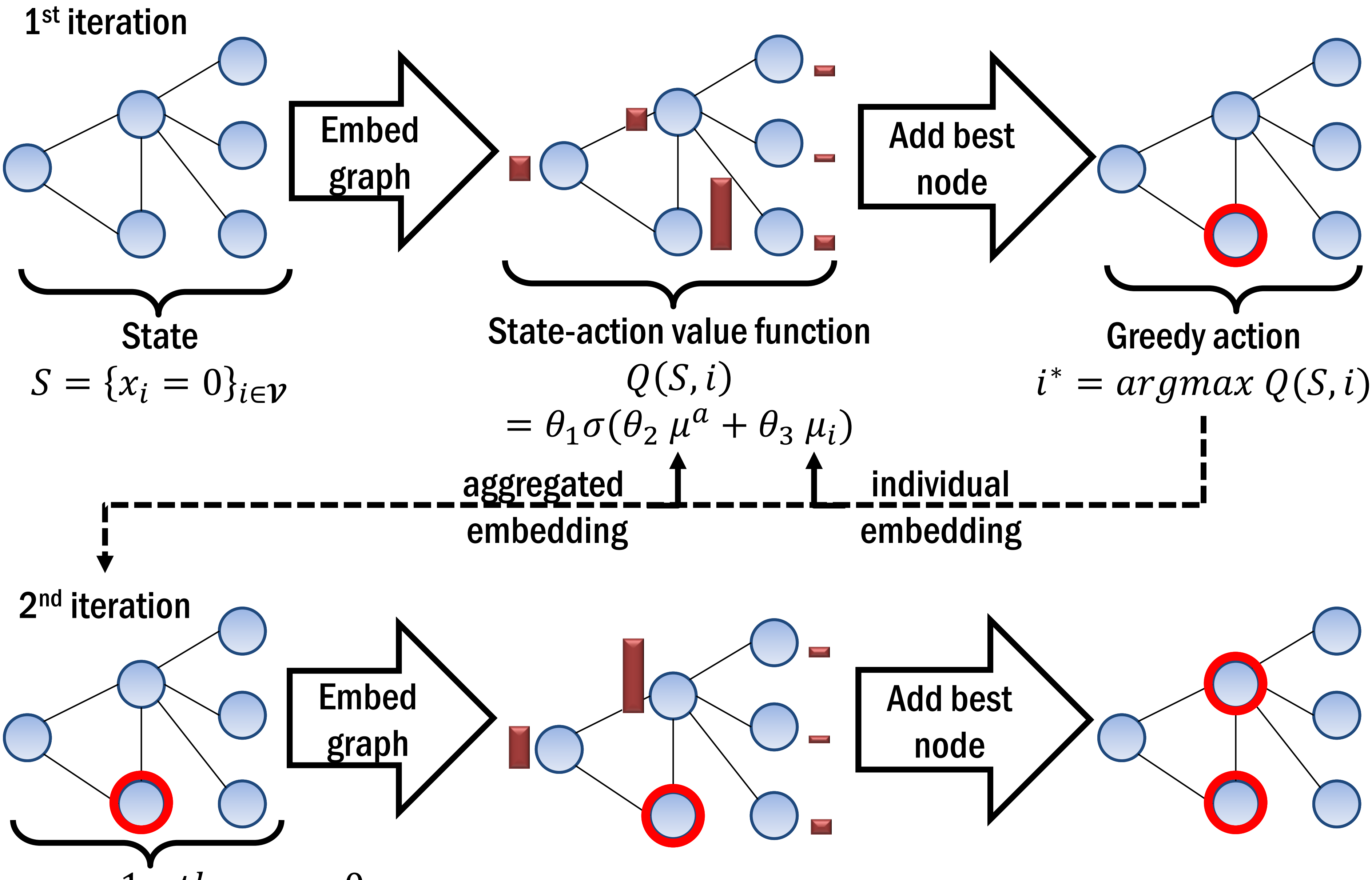
Action value function: $Q(S,i)$

Greedy policy:

$i^* = argmax_i\, Q(S,i)$

Update state S

Graph embedding for state-action value function

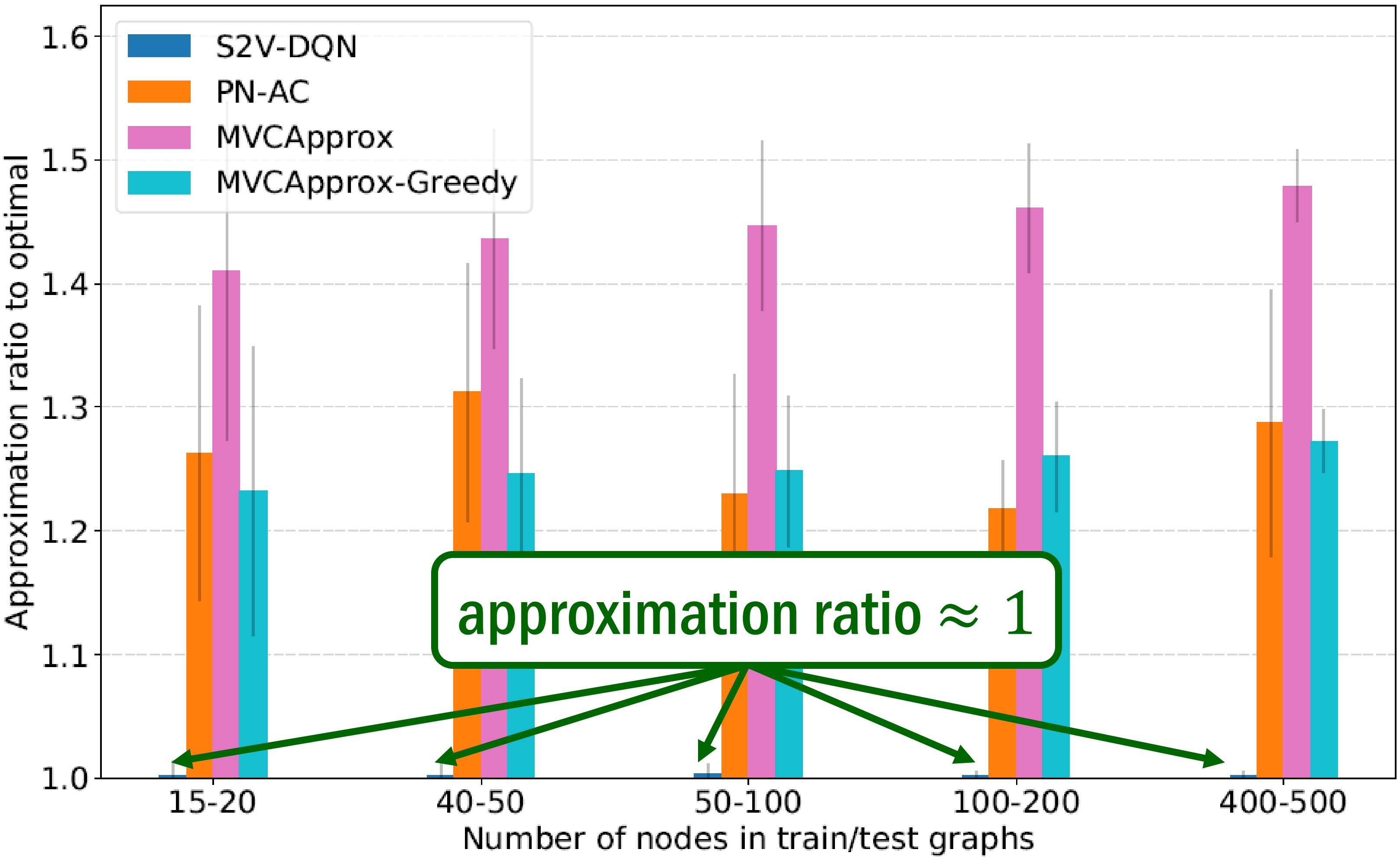


Embedding leads to better heuristic algorithm

Minimum vertex cover: smallest number of nodes to cover all edges

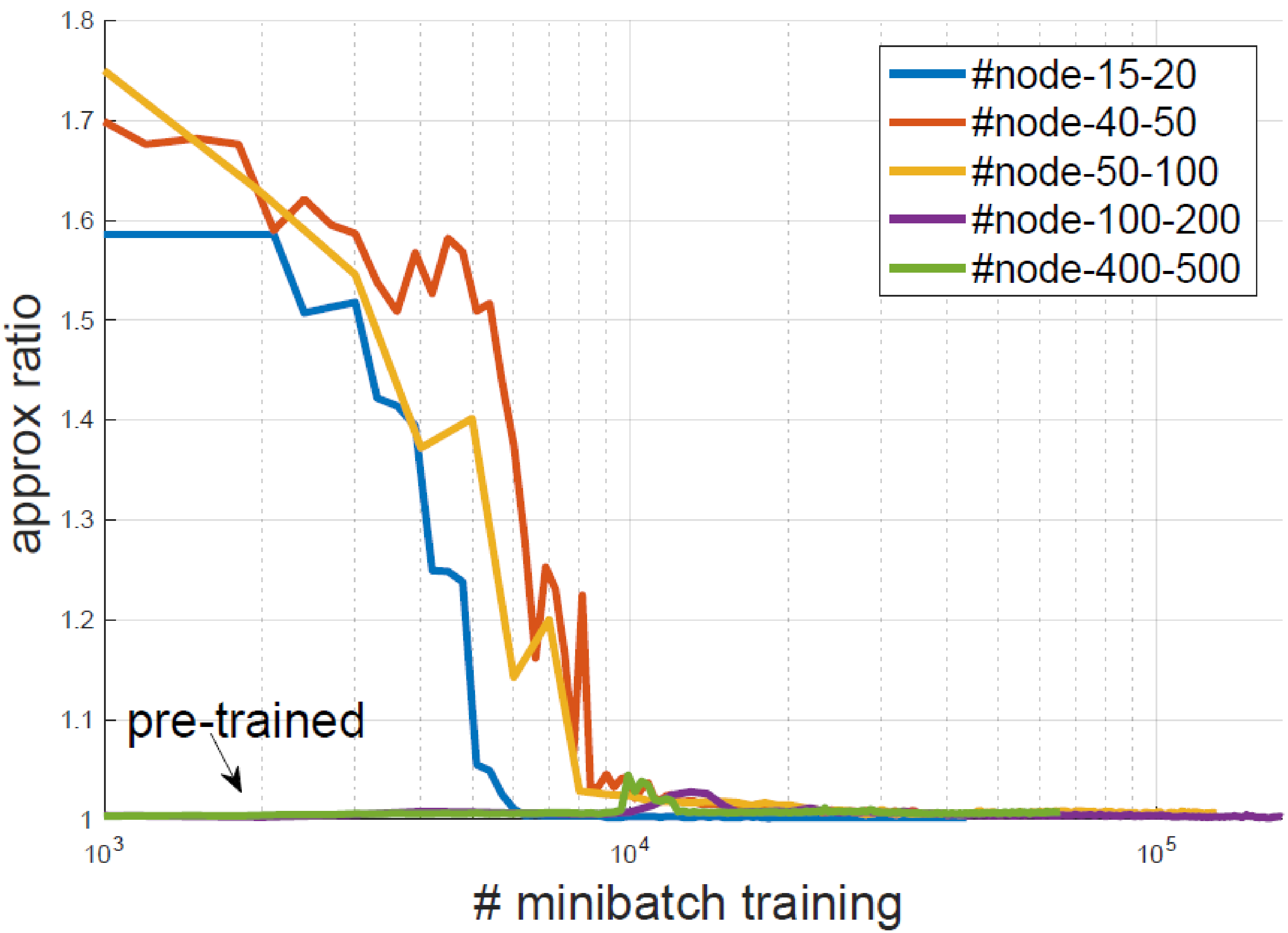
A distribution of scale free networks

Optimal approximated by running CPLEX for 1 hour



Training converge quite fast

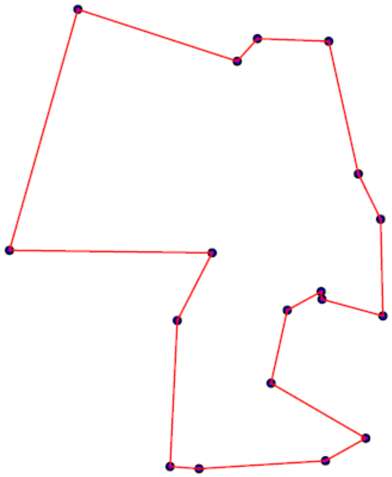
Pre-training: initialize embedding parameters with ones trained with smaller networks

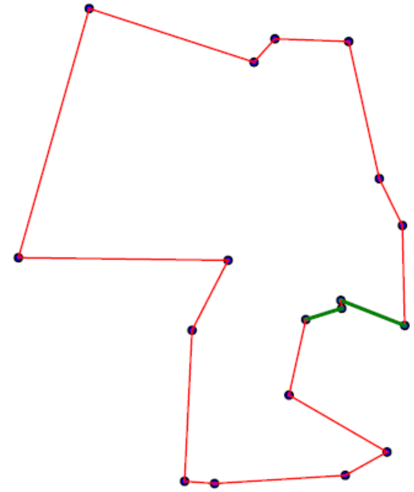


Also good for traveling salesman problem

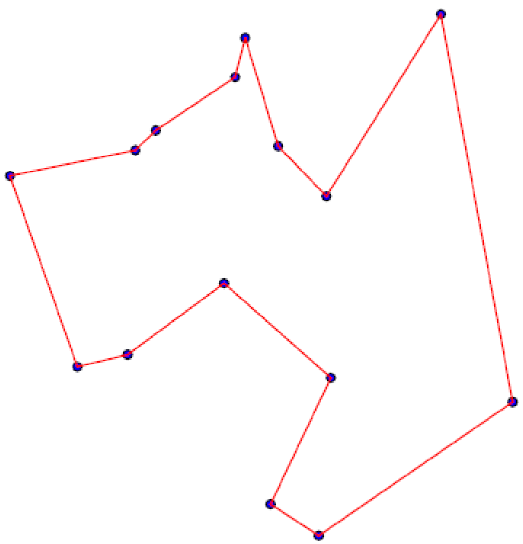
Optimal

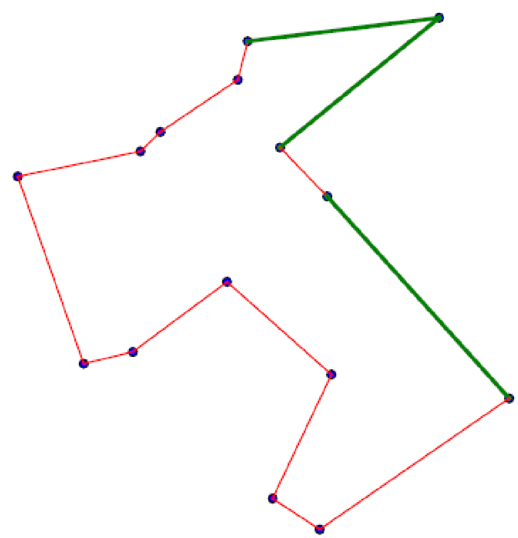
Embedding





0.07% longer

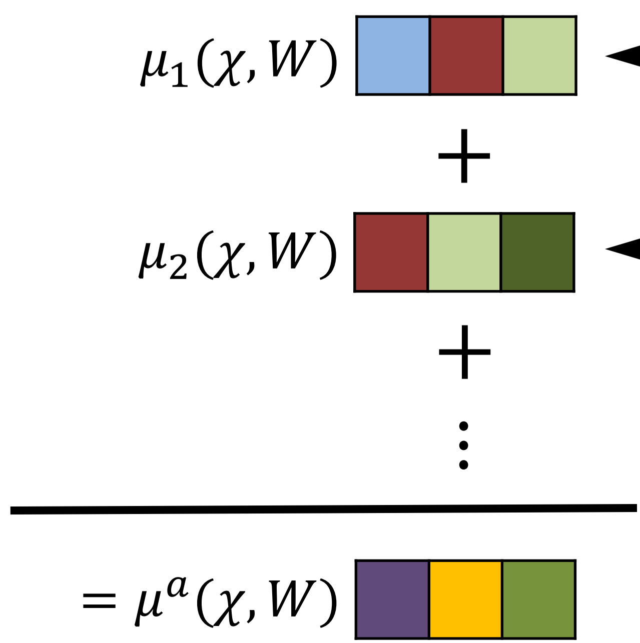


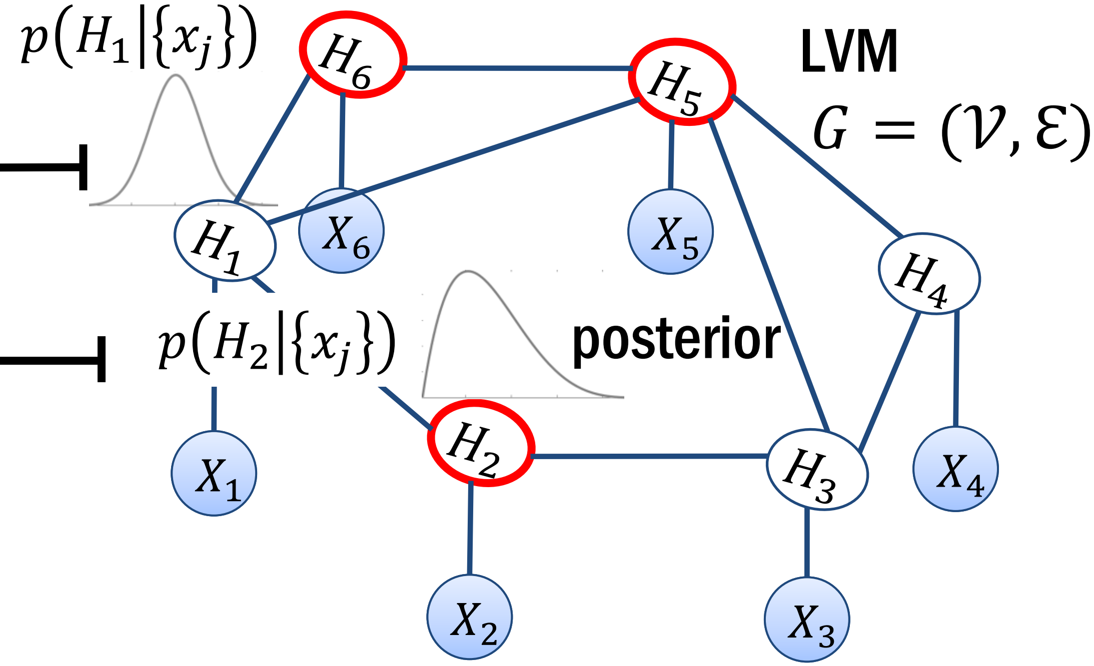


0.5% longer

Embedding as a tool for algorithm design

Embedding of node





Embedding of entire structure

- Embedding structures
- Learn better? Nonconvex & RL?
- New system & programming language?

[Dai, Dai & Song 2016]