DATALAB @Northeastern

Factorized Graph Representations for semi-supervised learning from sparse data

Krishna Kumar, Paul Langton, Wolfgang Gatterbauer

SIGMOD 2020, Thursday, June 18, 2020, R16: 3:00 – 4:30 pm PT

Slides: https://github.com/northeastern-datalab/factorized-graphs/

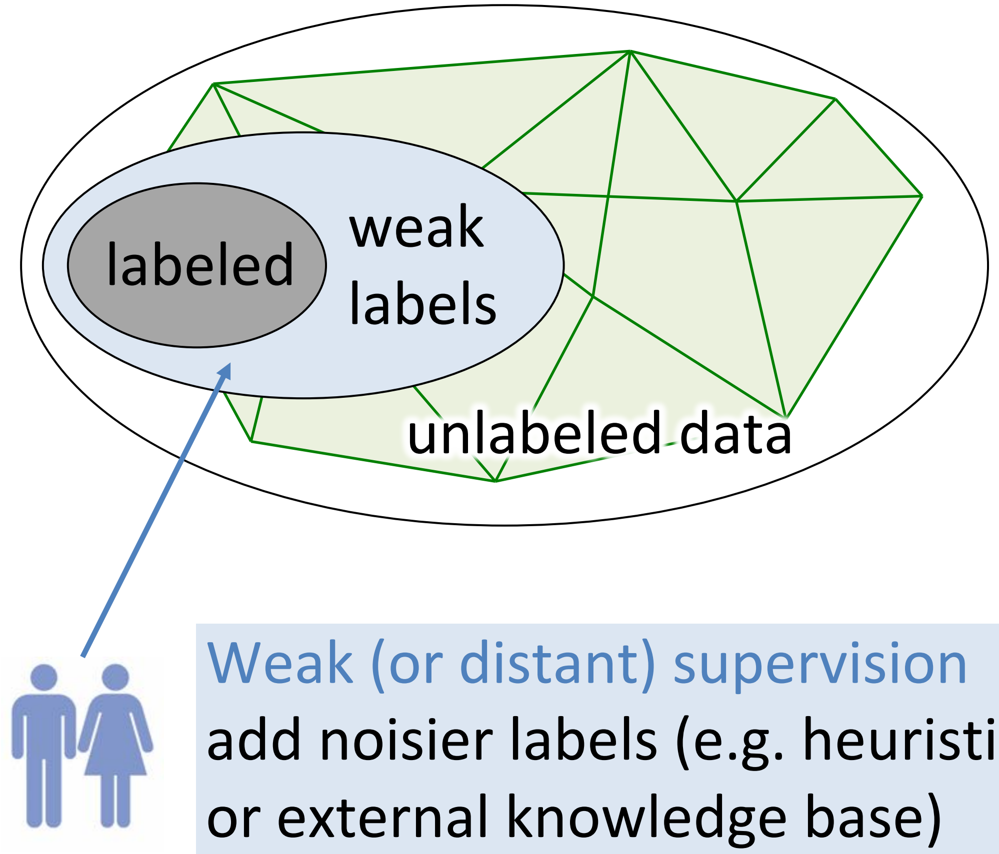
DOI: https://doi.org/10.1145/3318464.3380577

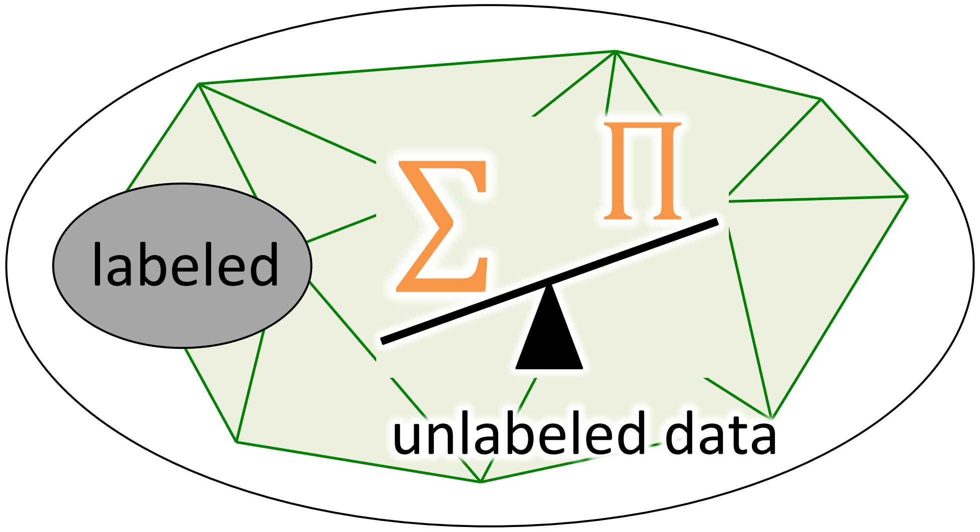
Data Lab: https://db.khoury.northeastern.edu

Learning from few labels with algebraic amplification

Semi-supervised learning
exploit relationships on label distribution (e.g. smoothness in networks)

Algebraic amplification
leverage algebraic properties of the algorithm to amplify signal in sparse data



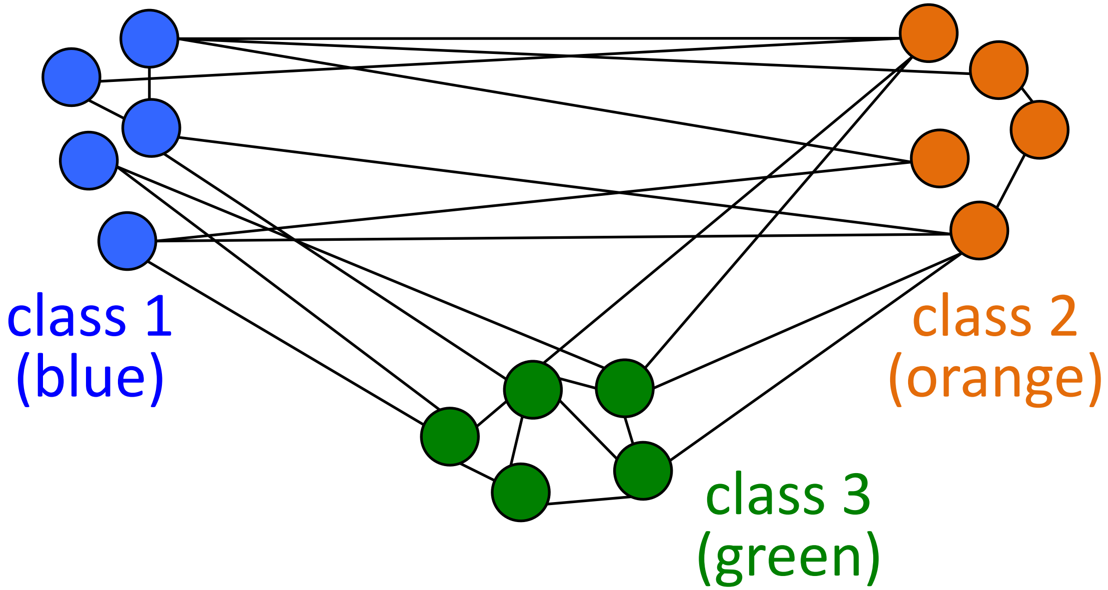


Weak (or distant) supervision
add noisier labels (e.g. heuristics, or external knowledge base)

Algebraic cheating
this requires "nice" algebraic properties; we may have to modify the algorithms ☺

Our focus today: Node classification in undirected graphs

Preference among node classes $\Rightarrow$ **Compatibilities between classes**



$$\mathbf{H} = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

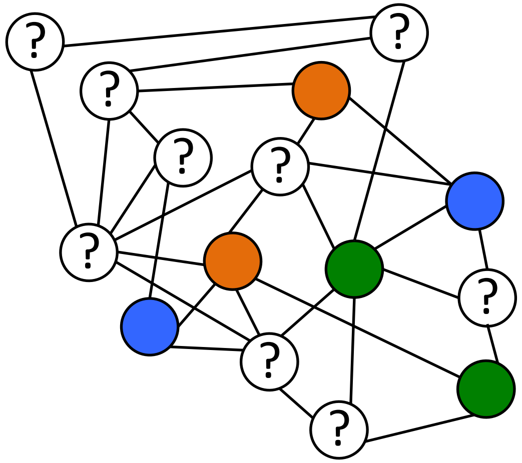
$\Sigma = 1$

orange prefers blue (and v.v.)

green prefers green

Our focus today: Node classification in graphs

Preference among node classes most of which are unlabeled $\Rightarrow$ **Compatibilities between classes not known to us ☹**



$$\mathbf{H}=\cancel{\begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}}$$

$\Sigma=1$

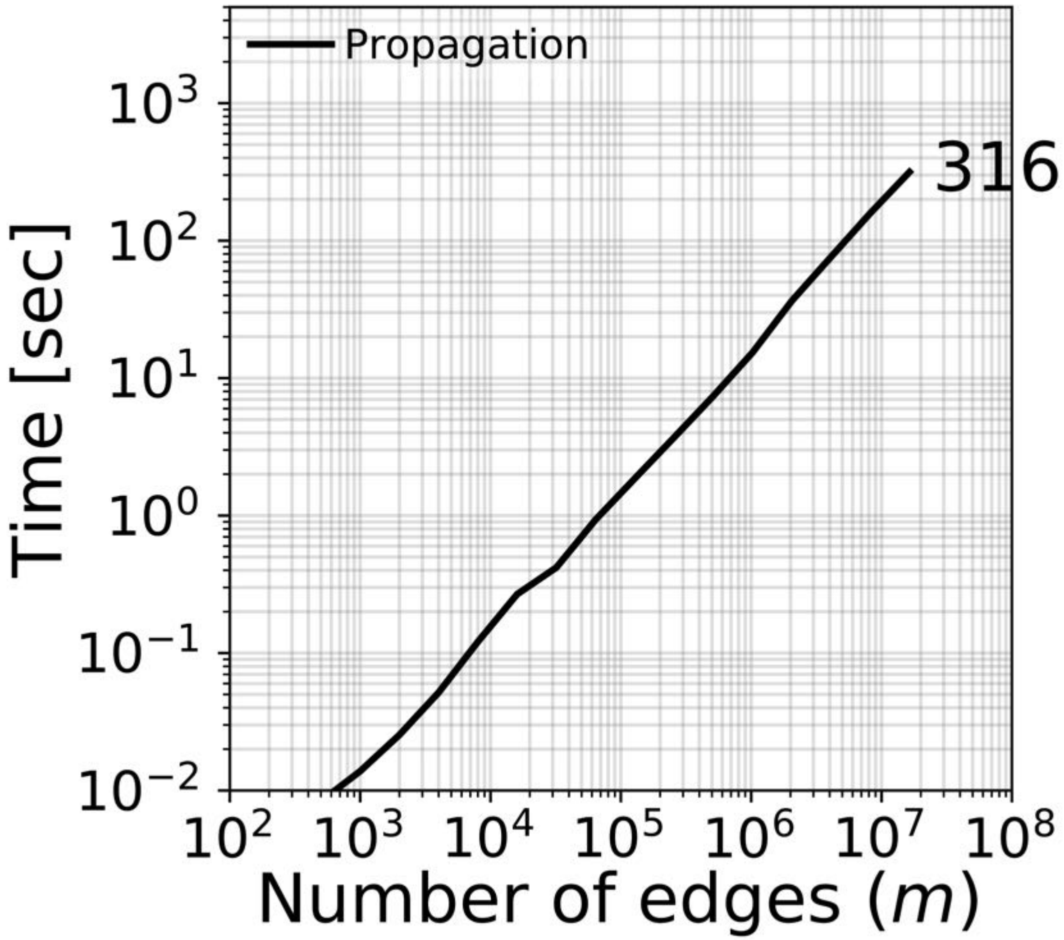
linearized belief propagation,
semi-supervised learning

Goal: Classify the remaining nodes: Estimate & propagate those compatibilities

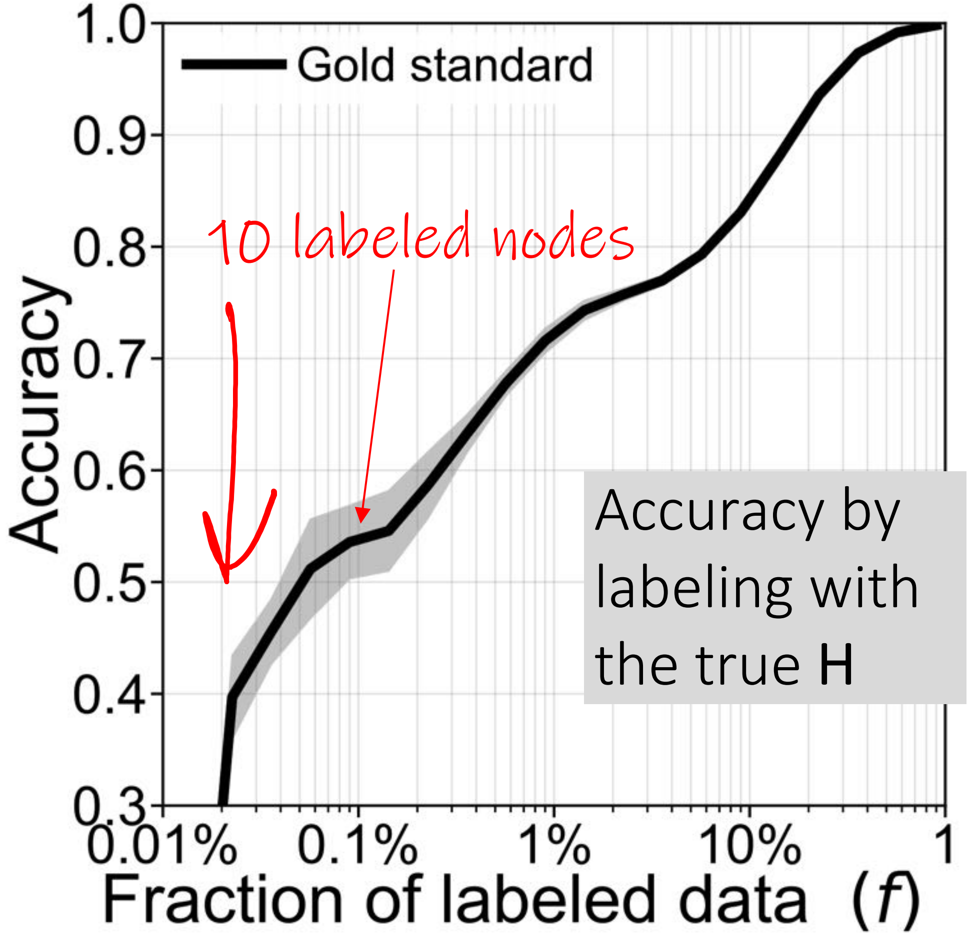
State-of-the-art: Heuristics / domain experts
We will estimate (learn) from sparse data

How well does it work?

Time and Accuracy for label propagation ***if we know* H**



Label propagation linear in # edges

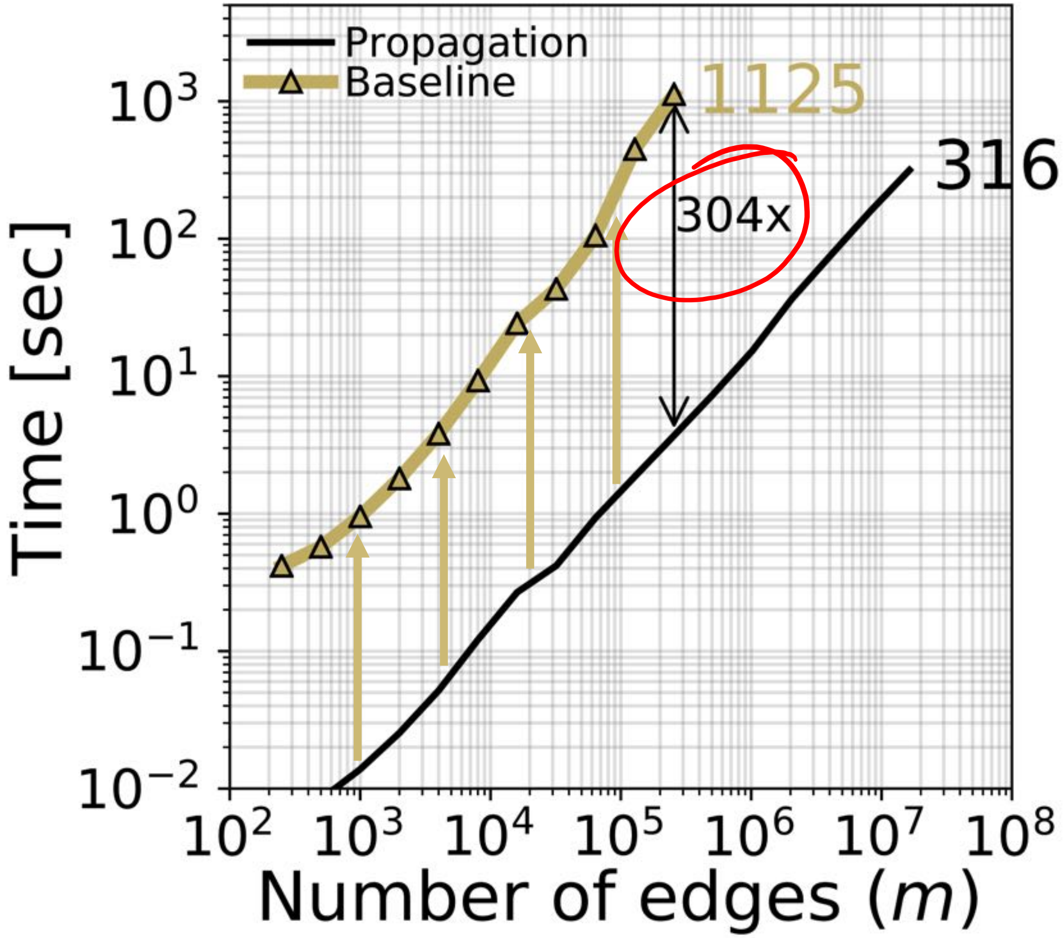


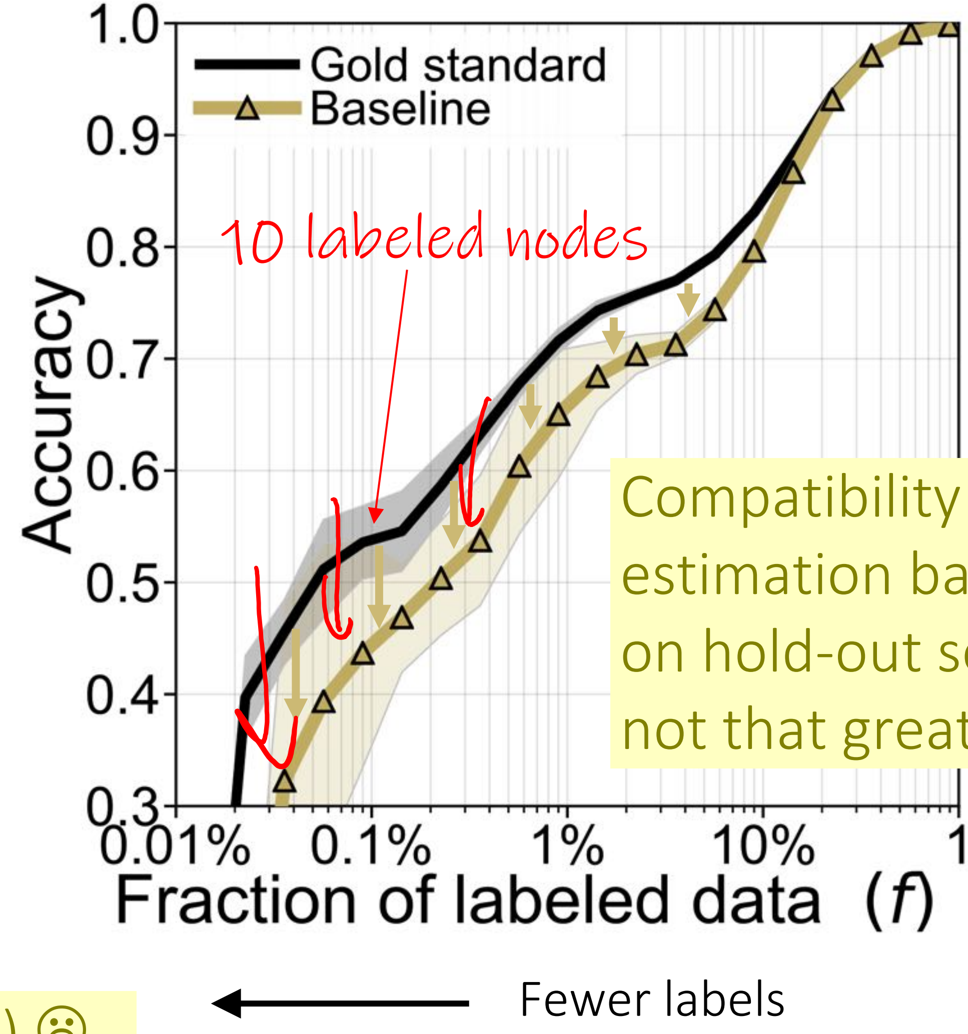
Fewer labels

Details: 10k nodes, degree d=25, **H** =

0.2	0.6	0.2
0.6	0.2	0.2
0.2	0.2	0.6

Time and Accuracy if we need to first estimate H ☹

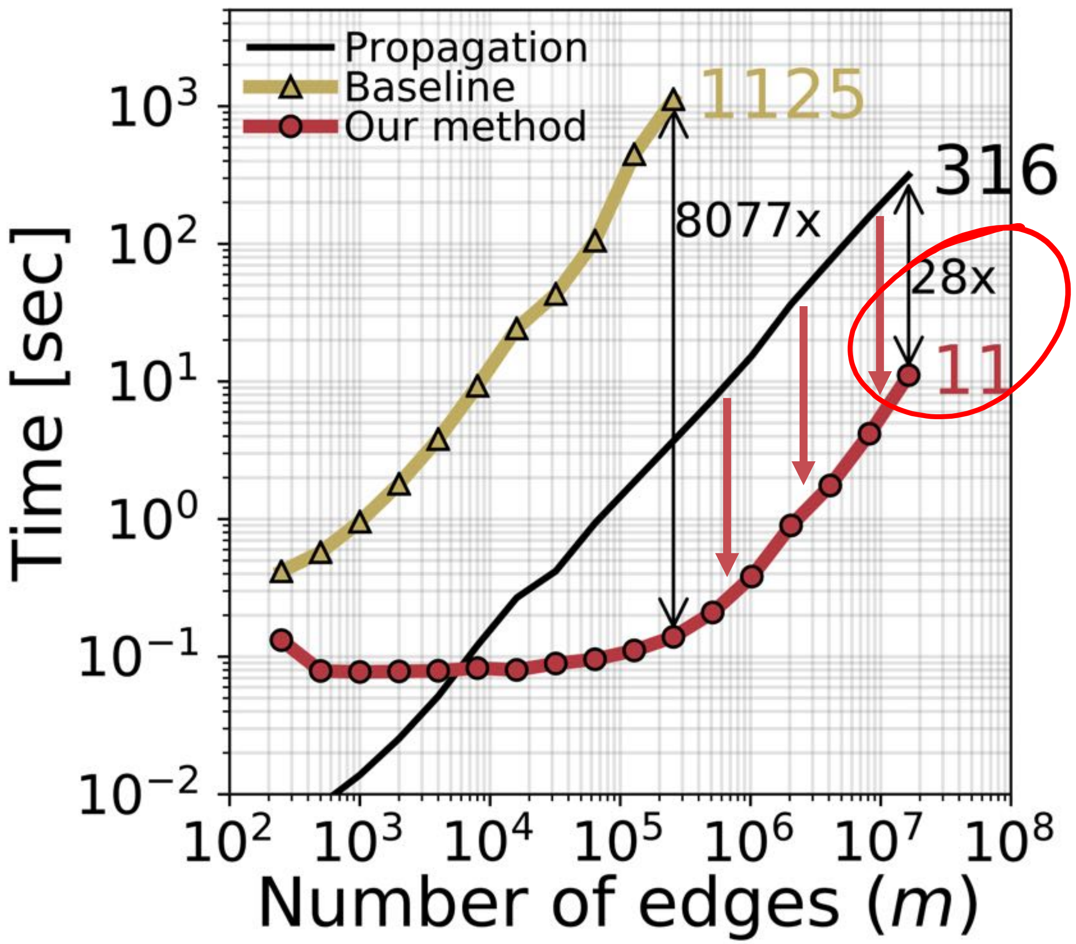


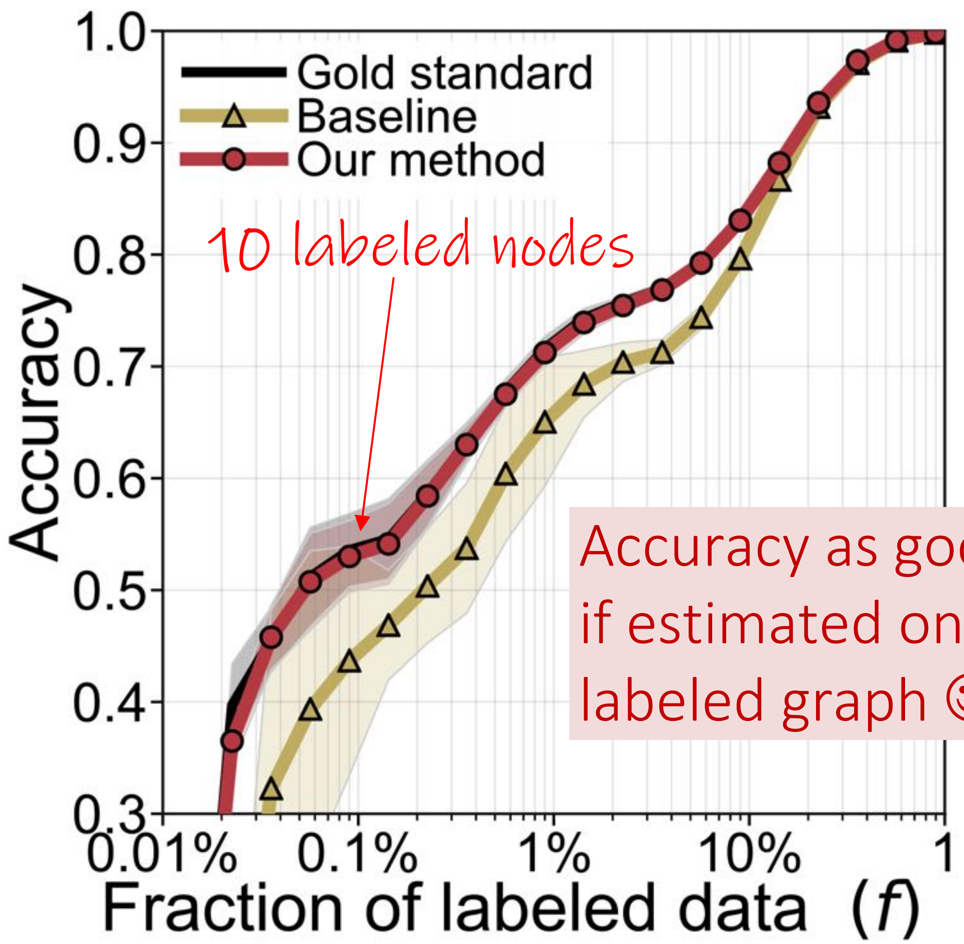


Compatibility estimation based on hold-out sets not that great ☹

Estimation uses inference as subroutine (thus slower) ☹

Time and Accuracy with our method ☺





Fewer labels

Accuracy as good as if estimated on fully labeled graph ☺

Our method for estimating H needs <5% of the time later needed for labeling ☺

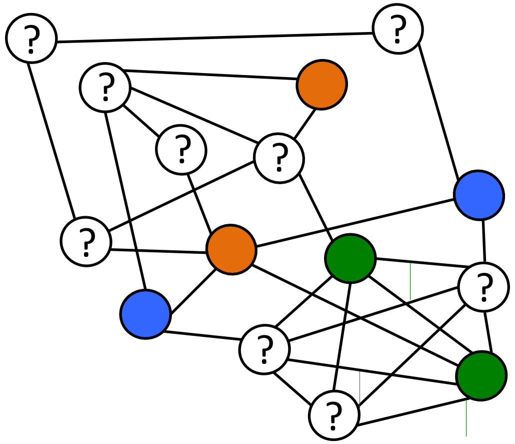
No more need for heuristics or domain experts ☺

What is the trick?

Splitting parameter estimation into two steps

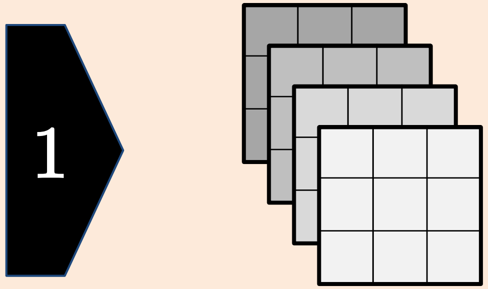
Parameter Estimation

Label Propagation



Sparsely labeled network

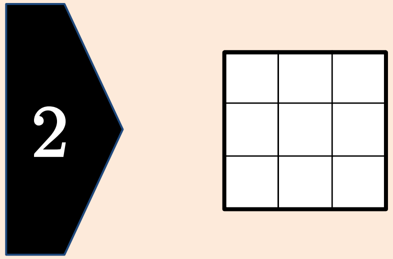
Derived statistics for path lengths 1,2,…,ℓ



$O(mk\ell)$ $k{\times}k$ matrices

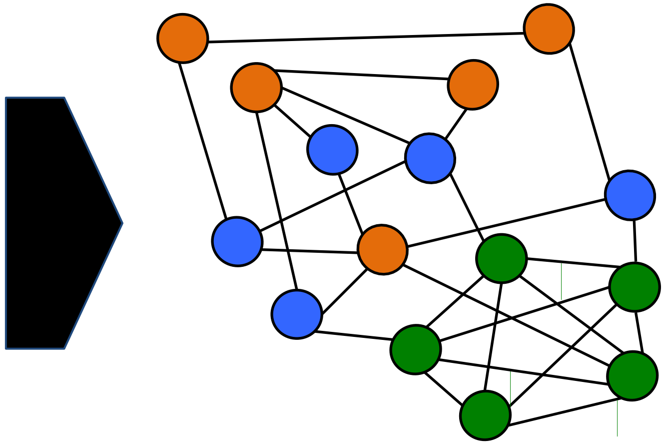
Factorized graph representations

Compatibility matrix



$O(k^4)$ $k{\times}k$ matrix

Optimization



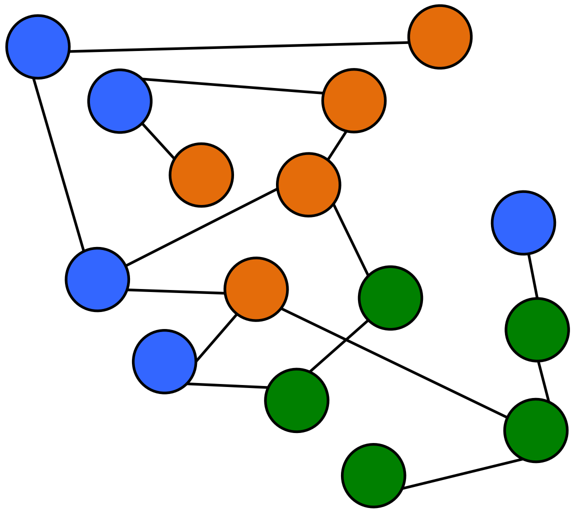
Fully labeled network

independent of graph size

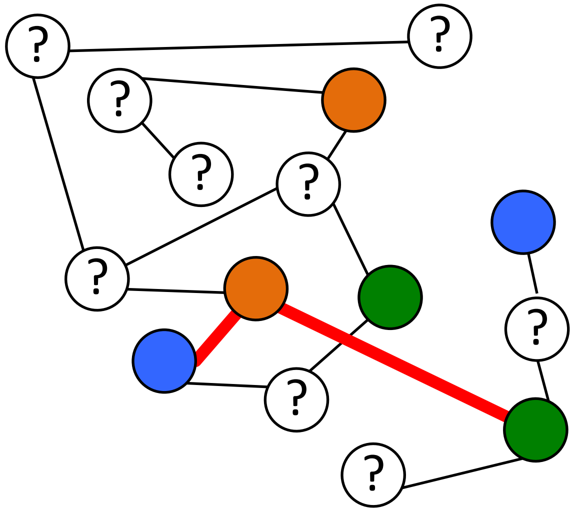
linear in # edges (m)
and # of classes (k)

A myopic view: counting relative neighbor frequencies

Fully labeled graph



Sparsely labeled graph



Neighbor count

$$\mathbf{M}=\begin{bmatrix} 2 & 6 & 2 \\ 6 & 2 & 2 \\ 2 & 2 & 6 \end{bmatrix} \Rightarrow$$

normalize

Gold standard compatibilities

$$\mathbf{H}=\begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$\Sigma=1$

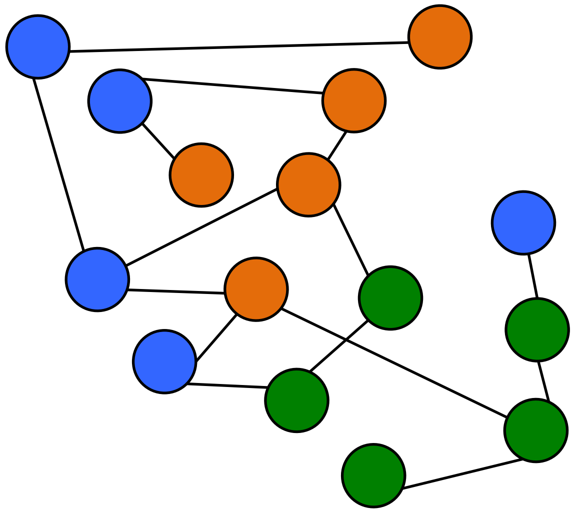
Labeled neighbor count

$$\hat{\mathbf{M}}=\begin{array}{c} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}\begin{array}{l} \Sigma=1 \\ \Sigma=2 \\ \\ \end{array} \Rightarrow \quad \hat{\mathbf{H}}$$

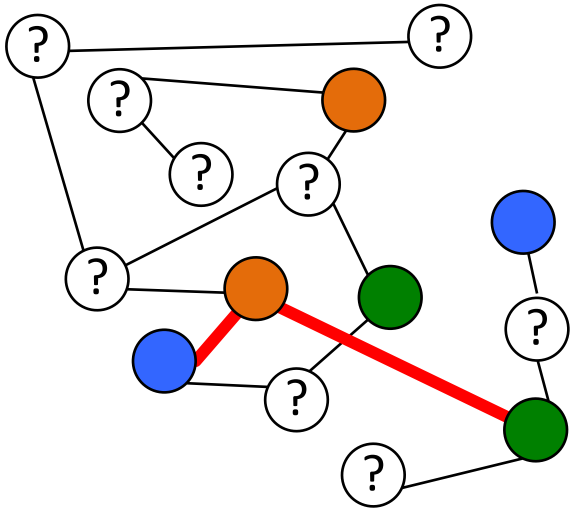
Idea: normalize, then find closest symmetric, doubly-stochastic matrix

A myopic view: counting relative neighbor frequencies

Fully labeled graph



Sparsely labeled graph



Neighbor count **Gold standard compatibilities**

$$\mathbf{M}=\begin{bmatrix} 2 & 6 & 2 \\ 6 & 2 & 2 \\ 2 & 2 & 6 \end{bmatrix} \Rightarrow \mathbf{H}=\begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

normalize

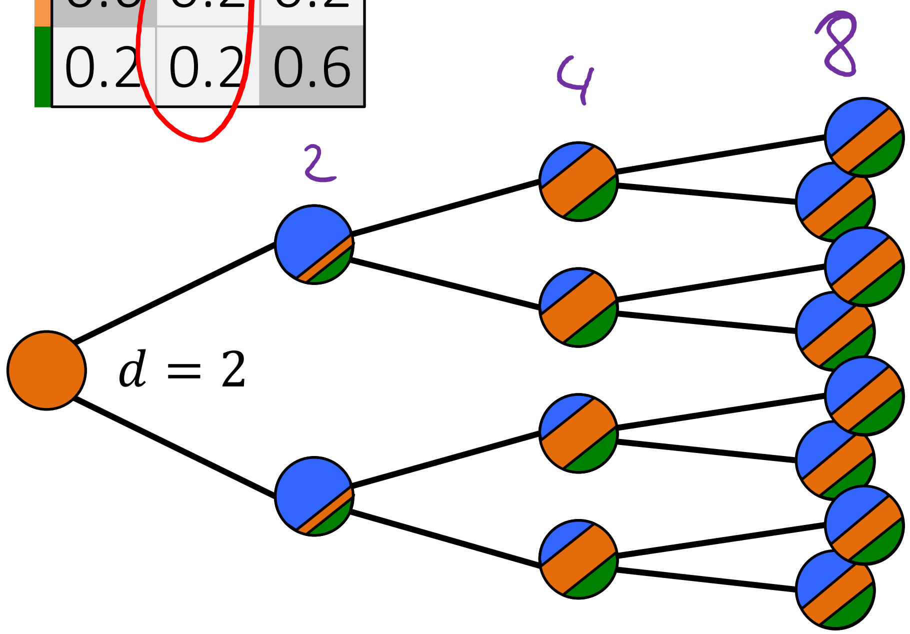
$\Sigma=1$

Assume f=10% labeled nodes.
What is the percentage of edges with labeled end points ?

1% ☹ Few nodes ⇒ even fewer edges mf^2

Distant compatibility estimation (DCE)

$$\mathbf{H} = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$



$$\mathbf{H}^2 = \begin{bmatrix} 0.44 & 0.28 & 0.28 \\ 0.28 & 0.44 & 0.28 \\ 0.28 & 0.28 & 0.44 \end{bmatrix}$$

$$\mathbf{H}^3 = \begin{bmatrix} 0.31 & 0.38 & 0.31 \\ 0.38 & 0.31 & 0.31 \\ 0.31 & 0.31 & 0.38 \end{bmatrix}$$

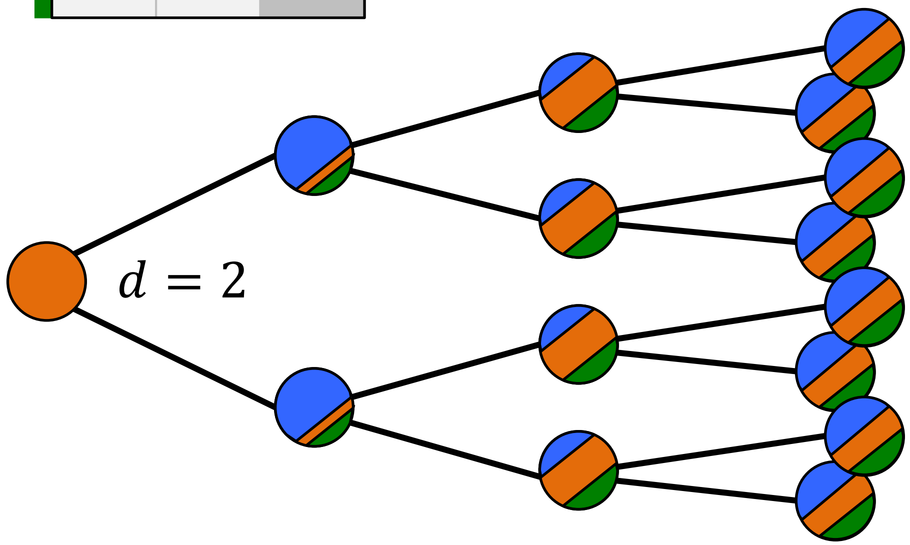
$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \underset{\ell = 1}{\begin{bmatrix} 0.6 \\ 0.2 \\ 0.2 \end{bmatrix}} \quad \underset{\ell = 2}{\begin{bmatrix} 0.28 \\ 0.44 \\ 0.28 \end{bmatrix}} \quad \underset{\ell = 3}{\begin{bmatrix} 0.38 \\ 0.31 \\ 0.31 \end{bmatrix}}$$

Expected signals for neighbors

0.6, 0.44, 0.38, 0.35, ...

Distant compatibility estimation (DCE)

$$\mathbf{H}=\begin{bmatrix}0.2 & 0.6 & 0.2\\ 0.6 & 0.2 & 0.2\\ 0.2 & 0.2 & 0.6\end{bmatrix}$$



	$\ell = 1$	$\ell = 2$	$\ell = 3$
0	0.6	0.28	0.38
1	0.2	0.44	0.31
0	0.2	0.28	0.31

Expected signals for neighbors

graph with:

- m edges
- f fraction labeled nodes
- d node degree

Expected # of labeled neighbors of distance ℓ ?

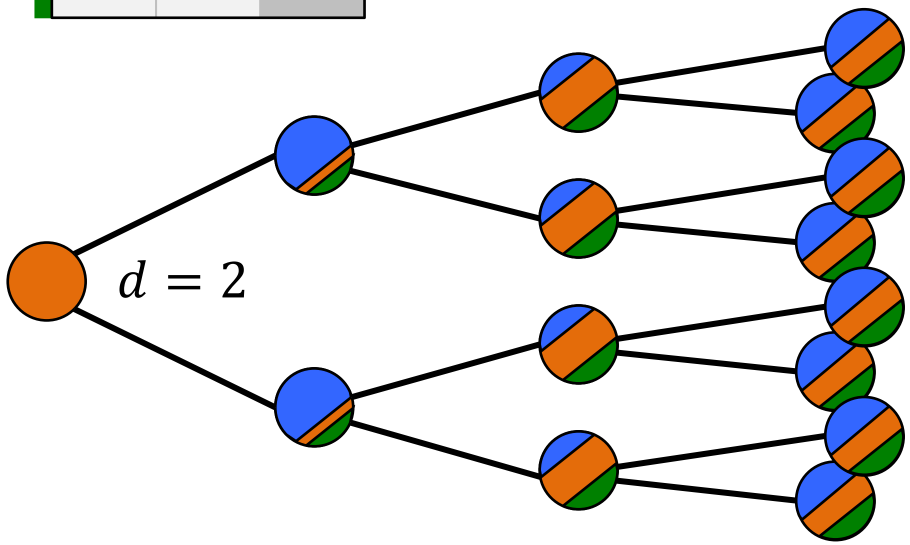
$d^{\ell-1} m f^2$ expected neighbors of distance ℓ

Idea: amplify the signal from observed length-ℓ paths ☺

Distant compatibility estimation (DCE)

DETAILS

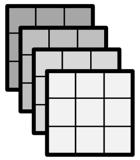
$$\mathbf{H} = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$



	$\ell = 1$	$\ell = 2$	$\ell = 3$
0	0.6	0.28	0.38
1	0.2	0.44	0.31
0	0.2	0.28	0.31

Expected signals for neighbors

distance-smoothed energy function



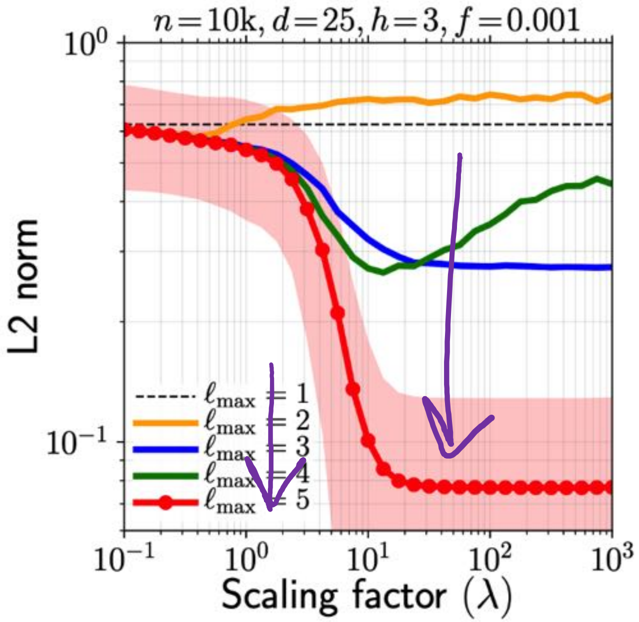
Statistics for path lengths 1, 2, ...

$$E(\mathbf{H}) = \sum_{\ell=1}^{\ell_{\max}} w_\ell \left\| \mathbf{H}^\ell - \widehat{\mathbf{P}}^{(\ell)} \right\|^2$$

$$w_{\ell+1} = \lambda w_\ell \qquad \mathbf{w} = [1, \lambda, \lambda^2, \dots]^\mathsf{T}$$

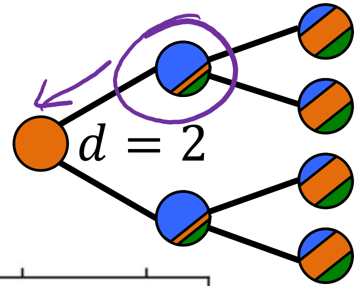
one single hyperparameter ☺

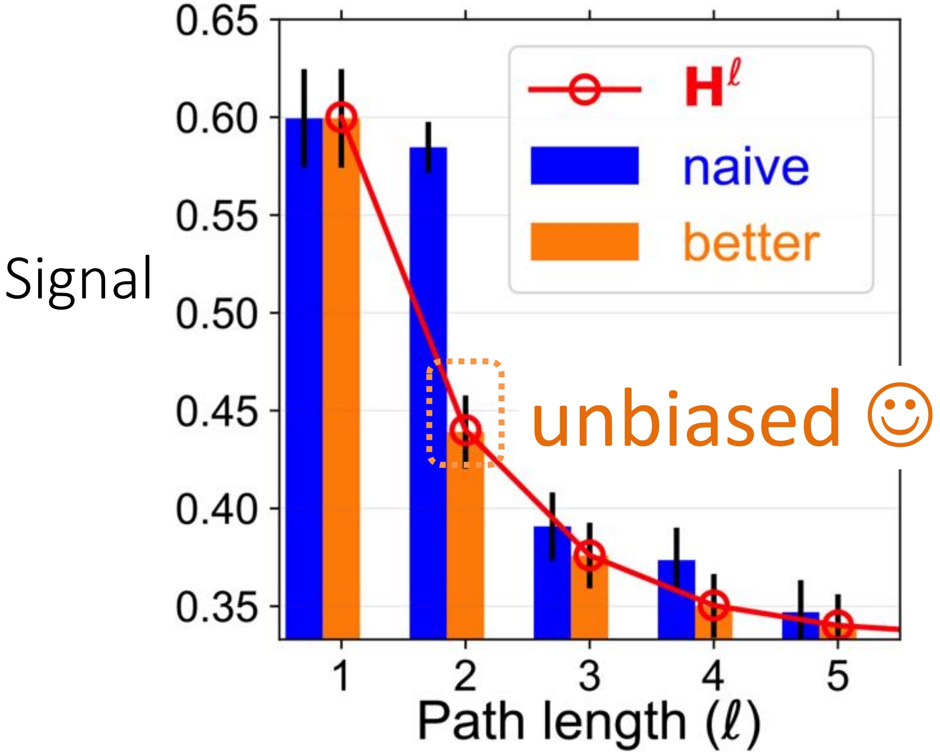
$\|\widehat{\mathbf{H}} - \mathbf{H}\|$
(smaller is better)



Two technical difficulties

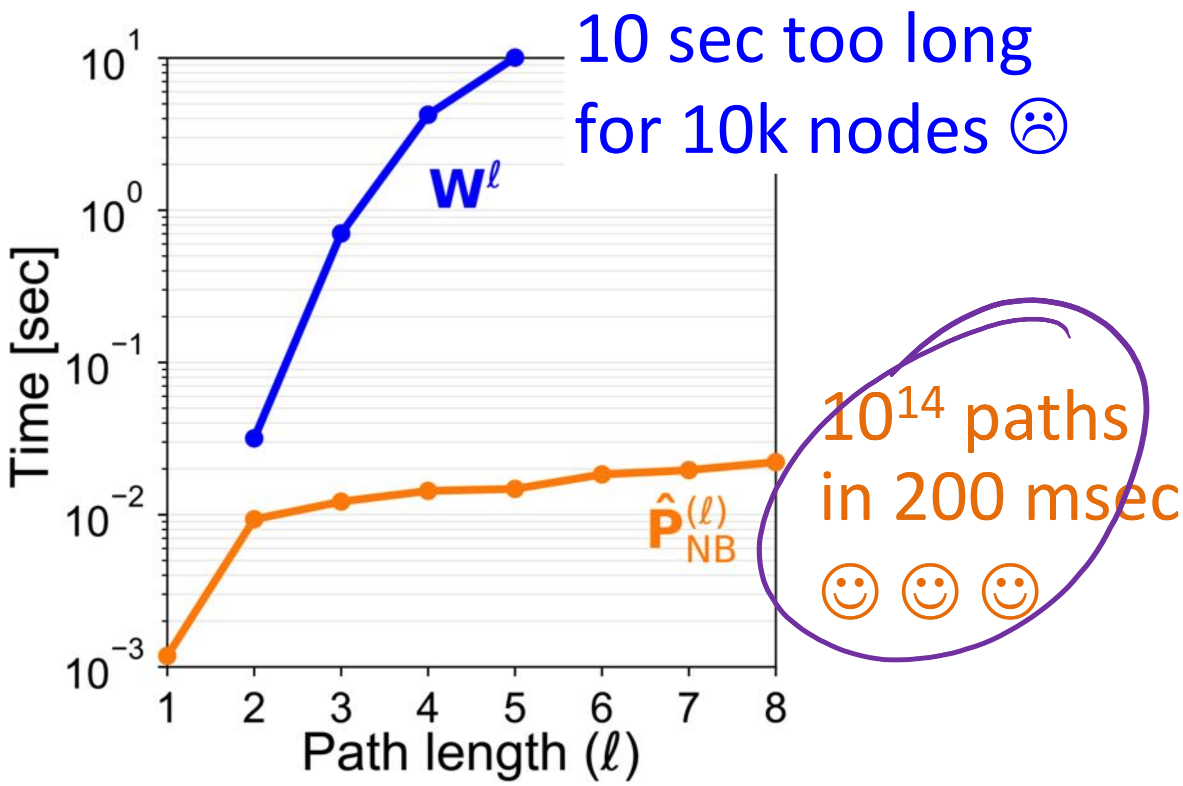
1. Idea from previous page gives biased estimates ☹





1. We must ignore backtracking paths

2. Calculating longer paths leads to dense matrix operations ☹ (**W** = sparse adjacency matrix)



2. Requires more careful re-factorization of the calculation

"factorized graph representations"

Scalable, Factorized Path summation

Details

Proposition 4.2 (Non-backtracking paths). *Let* $\mathbf{W}_{\mathrm{NB}}^{(\ell)}$ *be the matrix with* $W_{\mathrm{NB}\,ij}^{(\ell)}$ *being the number of non-backtracking paths of length* ℓ *from node* i *to* j. *Then* $\mathbf{W}_{\mathrm{NB}}^{(\ell)}$ *for* $\ell \geq 3$ *can be calculated via following recurrence relation:*

$$\mathbf{W}_{\mathrm{NB}}^{(\ell)} = \mathbf{W}\mathbf{W}_{\mathrm{NB}}^{(\ell-1)} - (\mathbf{D} - \mathbf{I})\mathbf{W}_{\mathrm{NB}}^{(\ell-2)} \tag{15}$$

with starting values $\mathbf{W}_{\mathrm{NB}}^{(1)} = \mathbf{W}$ *and* $\mathbf{W}_{\mathrm{NB}}^{(2)} = \mathbf{W}^2 - \mathbf{D}$. □

Algorithm 4.3 (Factorized path summation). *Iteratively calculate the graph summaries* $\hat{\mathbf{P}}_{\mathrm{NB}}^{(\ell)}$, *for* $\ell \in [\ell_{\max}]$ *as follows:*

(1) *Starting from* $\mathbf{N}_{\mathrm{NB}}^{(1)} = \mathbf{W}\mathbf{X}$ *and* $\mathbf{N}_{\mathrm{NB}}^{(2)} = \mathbf{W}\mathbf{N}_{\mathrm{NB}}^{(1)} - \mathbf{D}\mathbf{X}$, *iteratively calculate* $\mathbf{N}_{\mathrm{NB}}^{(\ell)} = \mathbf{W}\mathbf{N}_{\mathrm{NB}}^{(\ell-1)} - (\mathbf{D} - \mathbf{I})\mathbf{N}_{\mathrm{NB}}^{(\ell-2)}$.

(2) *Calculate* $\mathbf{M}_{\mathrm{NB}}^{(\ell)} = \mathbf{X}^{\mathsf{T}}\mathbf{N}_{\mathrm{NB}}^{(\ell)}$.

(3) *Calculate* $\hat{\mathbf{P}}_{\mathrm{NB}}^{(\ell)}$ *from normalizing* $\mathbf{M}^{(\ell)}$ *with Eq. 9.*

Proposition 4.4 (Factorized path summation). *Algorithm 4.3 calculates all graph statistics* $\hat{\mathbf{P}}_{\mathrm{NB}}^{(\ell)}$ *for* $\ell \in [\ell_{\max}]$ *in* $\boxed{O(mk\ell_{\max})}$.

Intuition

Relational algebra

$$\pi_x\big(R(x) \bowtie S(x,y)\big)$$

$$\Rightarrow \quad R(x) \bowtie \pi_x S(x,y)$$

(**X** = thin label matrix)

Linear algebra

$$(W \cdot W) \cdot X$$

$$\Rightarrow \quad W \cdot (W \cdot X)$$

Scalable factorized path summation

Similar ideas of factorized calculation:

- Generalized distributive law
  [Aji-McEliece IEEE TIT '00]
- Algebraic path problems
  [Mohri JALC'02]
- Valuation algebras
  [Kohlas-Wilson AI'08]
- Factorized databases
  [Olteanu-Schleich Sigmod-Rec'16]
- FAQ (Functional Aggregate Queries)
  [AboKhamis-Ngo-Rudra PODS'16]
- Associative arrays
  [Kepner, Janathan MIT-press'18]
- Optimal ranked enumeration
  [Tziavelis+ VLDB'20]

Intuition

Relational algebra

$$\pi_x(R(x) \bowtie S(x,y))$$

$$\Rightarrow \quad R(x) \bowtie \pi_x S(x,y)$$

Linear algebra

(**X** = thin label matrix)

$$(W \cdot W) \cdot X$$

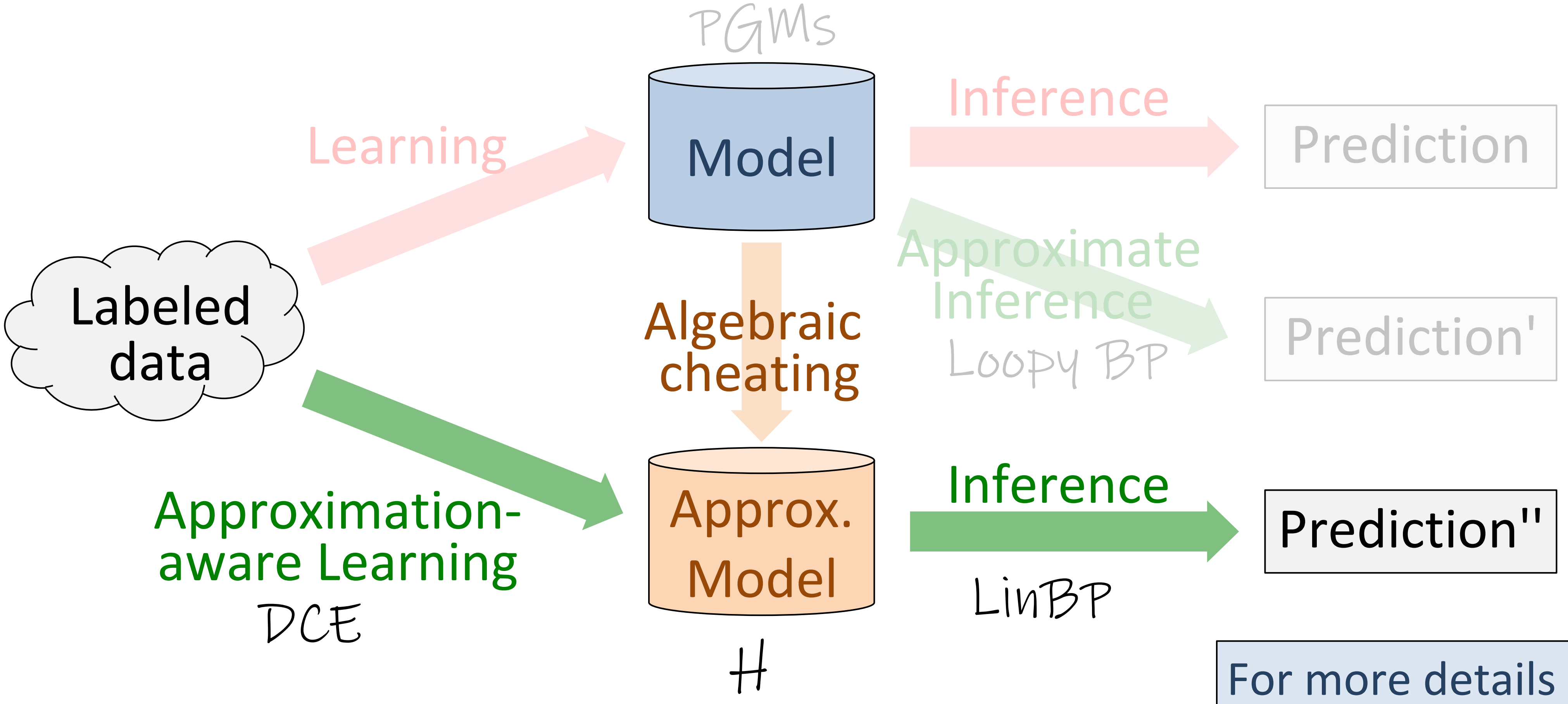
$$\Rightarrow \quad W \cdot (W \cdot X)$$

More details (super happy to discuss further in 1-on-1's)

1. Constrained optimization → unconstrained opt. in free parameters
2. Closed form for gradient: gradient-based optimization even faster
3. Random restarts for optimization: but for an optimization on graph sketches, thus independent of n, yet $O(k^4)$
4. Energy-minimization based explanation of LinBP
5. Originally proposed "centering" for LinBP not necessary
6. Proof of unbiased estimator for equal label distribution
7. Non-backtracking paths in factorized calculation that does not require larger $(2m \times 2m)$ "Hashimoto matrix"
8. Lots of experiments on real graphs
9. Even works on graphs without any labeled neighbors ☺

Back to the big picture

"Algebraic cheating" for approximation-aware learning



[Arxiv 2014] Semi-supervised learning with heterophily
[VLDB 2015] Linearized and Single-pass belief propagation
[AAAI 2017] The linearization of pairwise Markov random fields
[VLDBJ 2017] Dissociation and propagation for approximate lifted inference
[UAI 2018] Dissociation-based oblivious bounds for weighted model counting
[SIGMOD 2019] Anytime approximation in probabilistic databases via scaled dissociations
[SIGMOD 2020] Factorized graph representations for semi-supervised learning from sparse data
Supported by NSF IIS-1762268-CAREER: Scaling approximate inference and approximation-aware learning

For more details please visit

DATALAB @Northeastern

https://db.khoury.northeastern.edu/

Thank you ☺