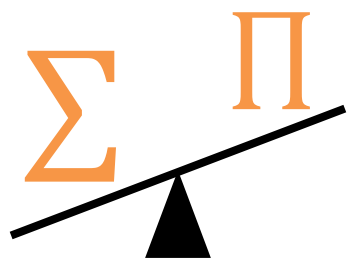
DATA Lab @ Northeastern

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



Algebraic Amplification for semi-supervised learning from sparse data

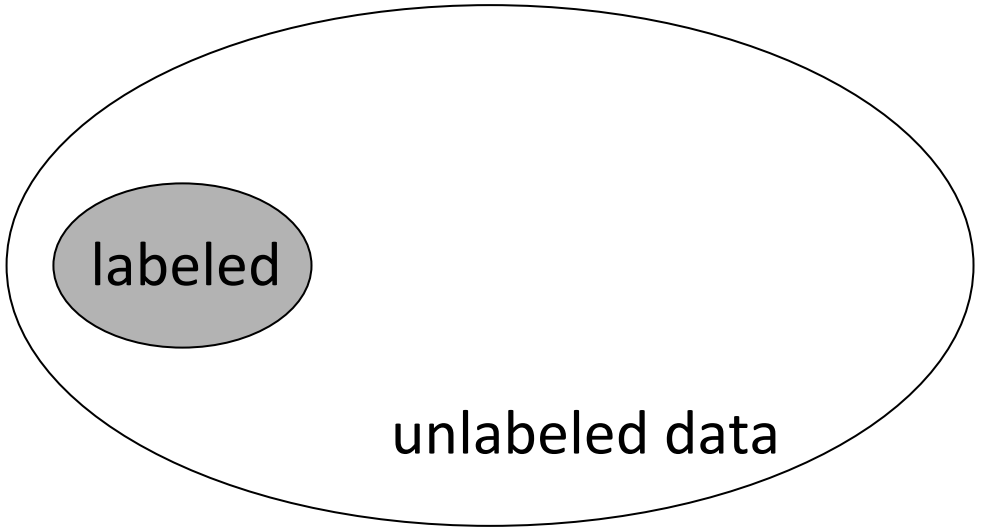
Wolfgang Gatterbauer

currently applying for PhD programs

Based on joint work with **Krishna Kumar** and **Paul Langton**

North East Database Day 2020 (Jan 27, 2020)

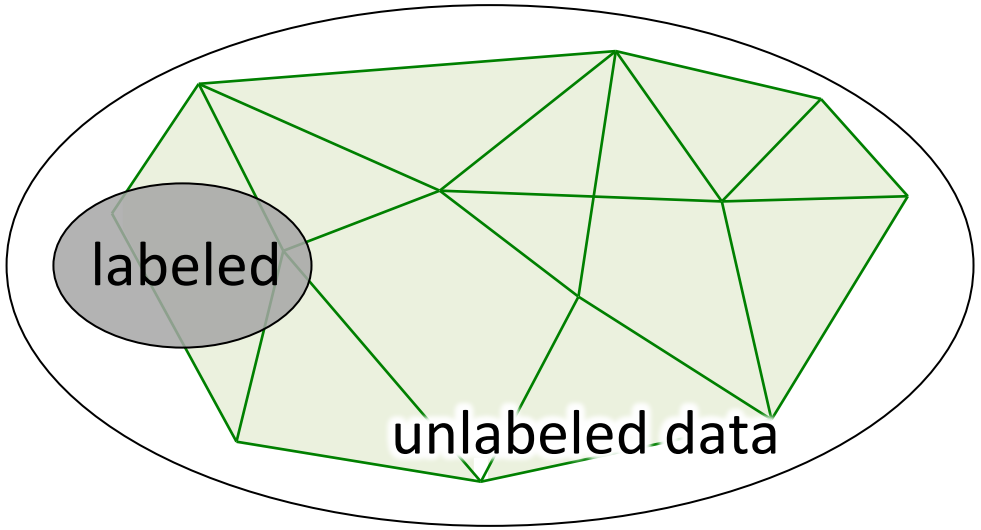
Learning from few labels



Learning from few labels

Semi-supervised learning

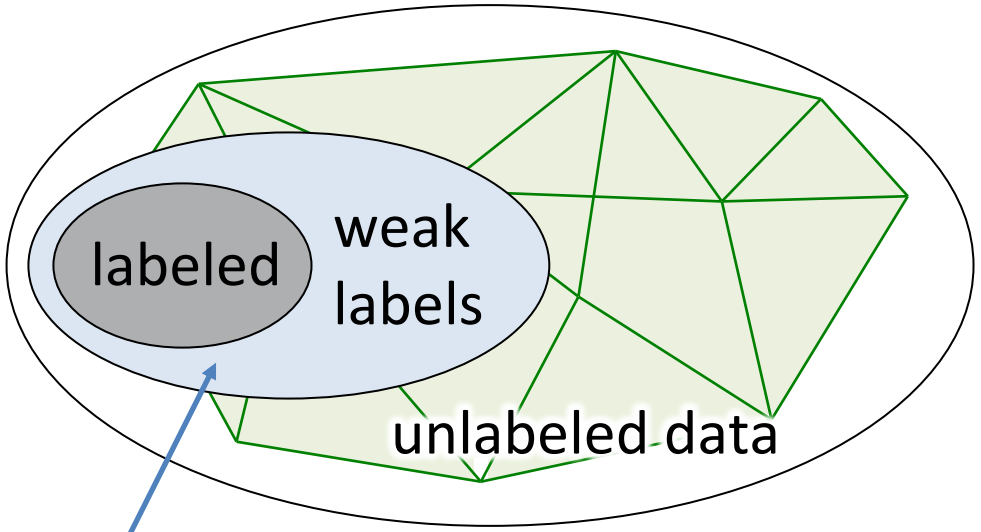
exploit relationships on label distribution (e.g. smoothness in networks)



Learning from few labels

Semi-supervised learning

exploit relationships on label distribution (e.g. smoothness in networks)

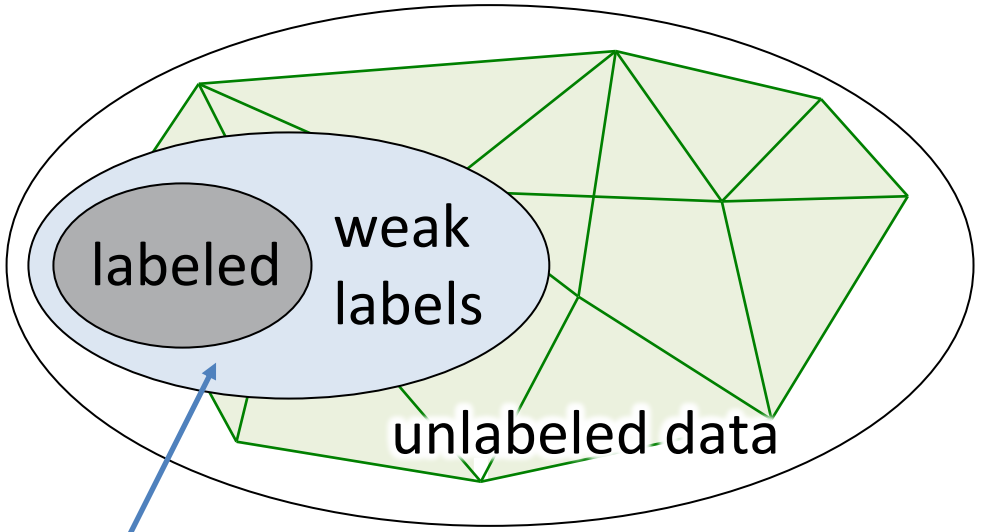


Weak (or distant) supervision

add noiser labels (e.g. heuristics, or external knowledge base)

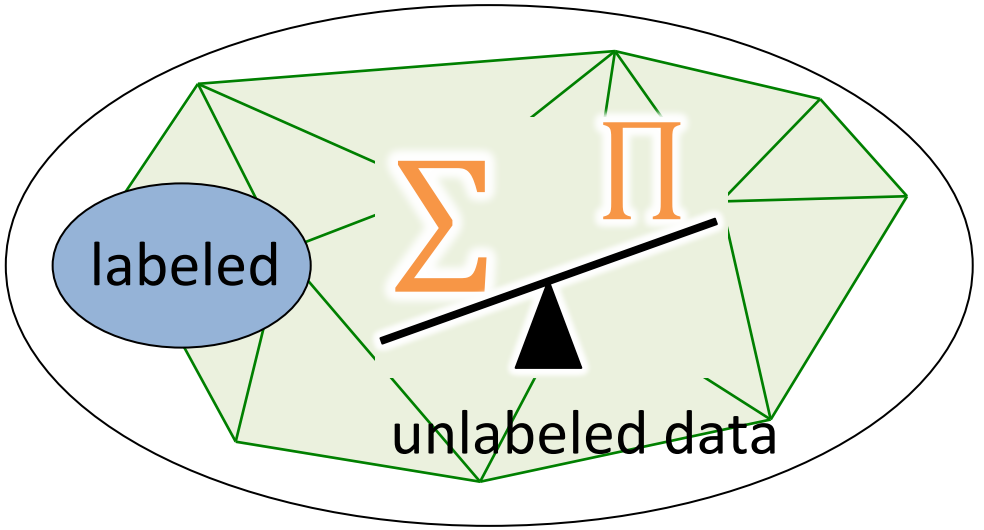
Learning from few labels with algebraic amplification

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



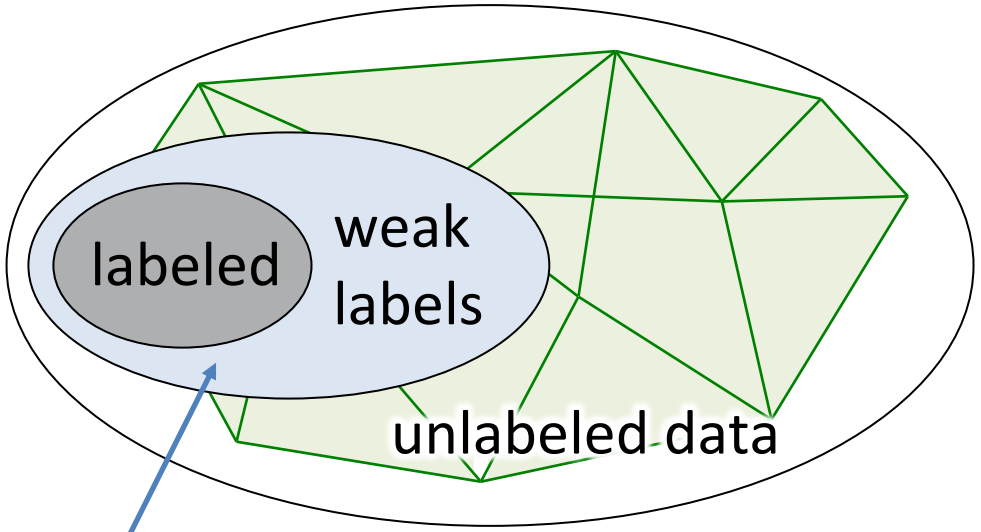
Weak (or distant) supervision
add noiser labels (e.g. heuristics, or external knowledge base)

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



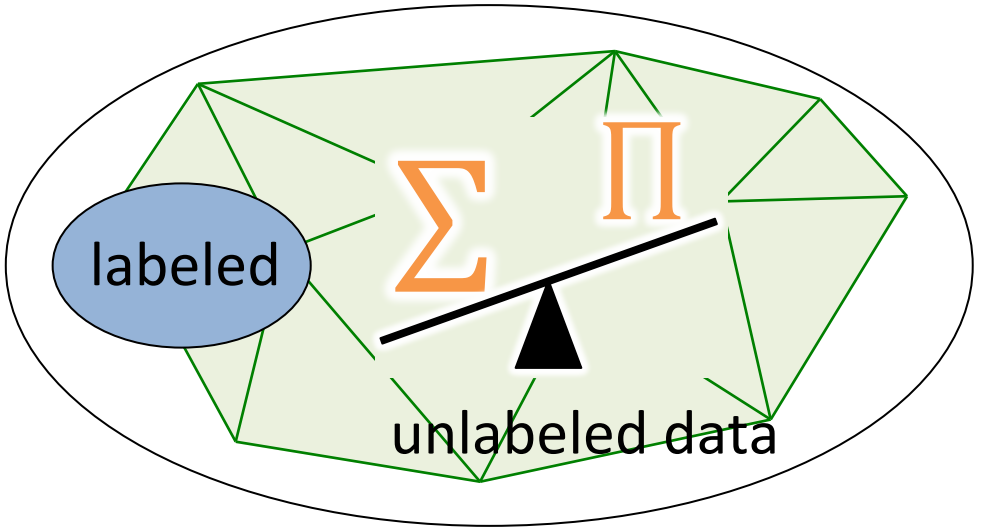
Learning from few labels with algebraic amplification

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



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

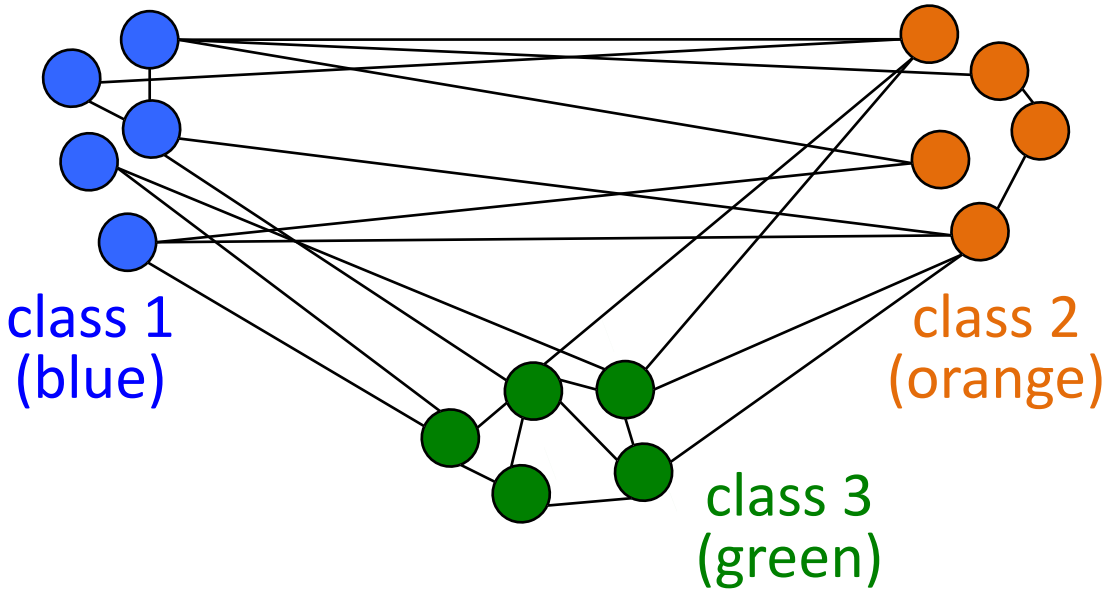
Algebraic amplification
leverage algebraic properties of the algorithm to amplify signal in sparse data



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

Classes with different relative attractions

If we could see the true labels:



blue prefers orange

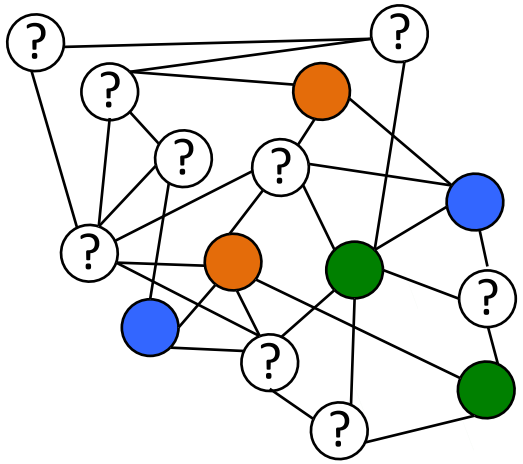
green prefers green

Compatibilities between classes

$$\mathbf{H}=\begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \quad \Sigma=1$$

The problem we are trying to solve

Given: Graph & few labels



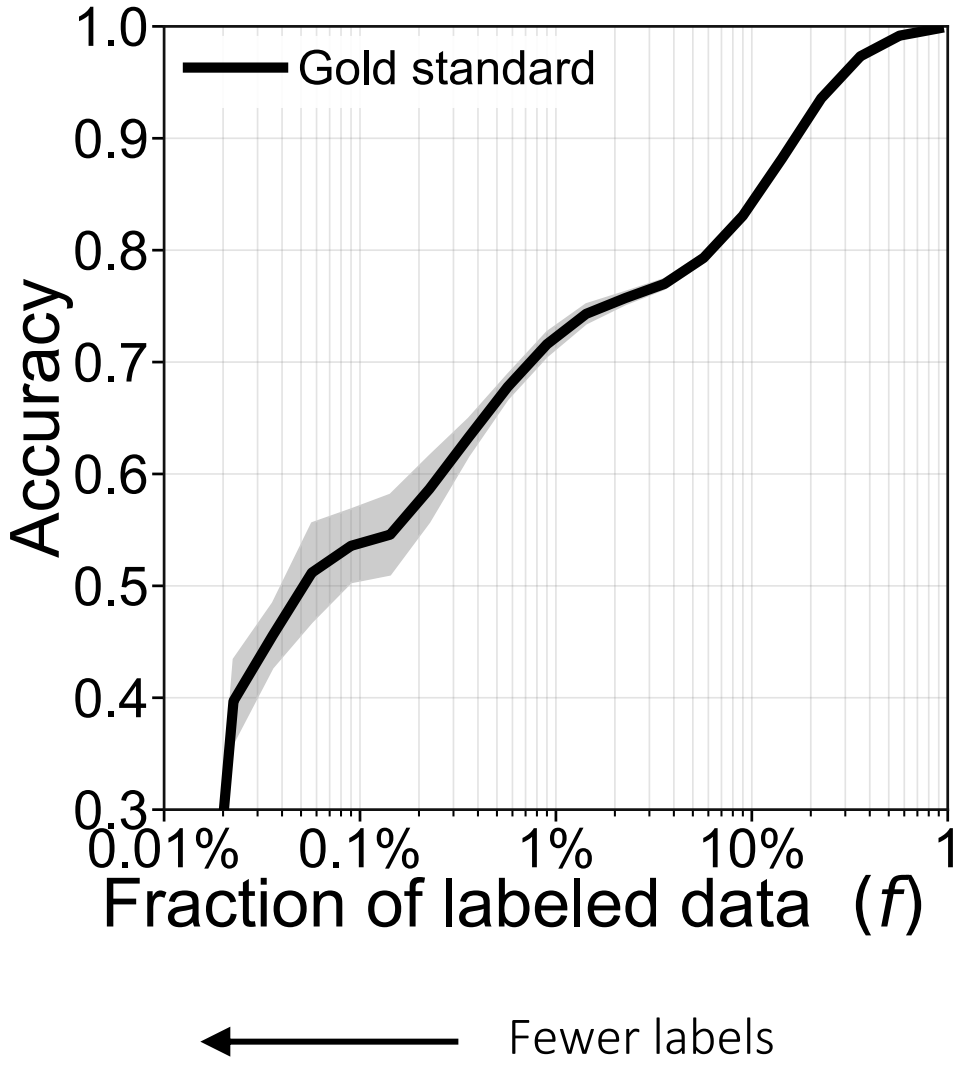
Compatibilities between classes not known ☹

$$\mathbf{H} = \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \quad \Sigma=1$$

⇒ Classify the remaining nodes

How well does this work?

How well does it work?

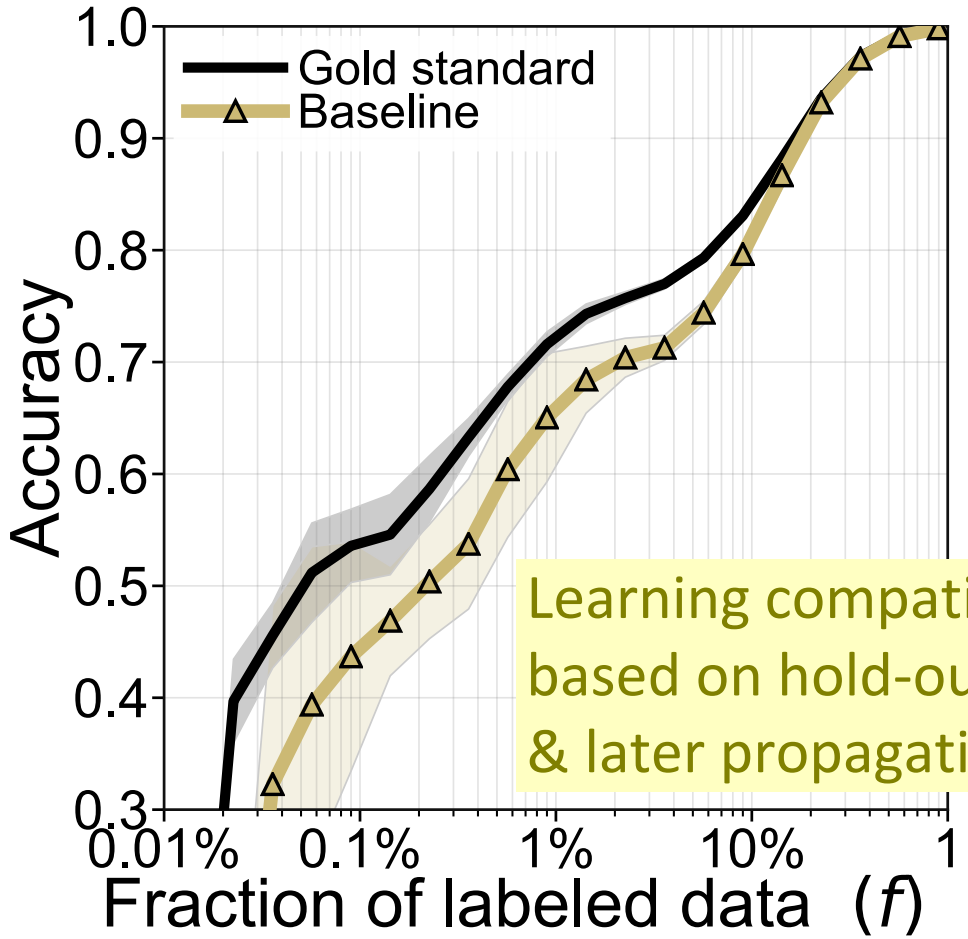


Gold standard: Assume we could estimate the compatibilities on the fully labeled graph, then use those to label the rest

← Fewer labels

Details: 10k nodes

How well does it work?

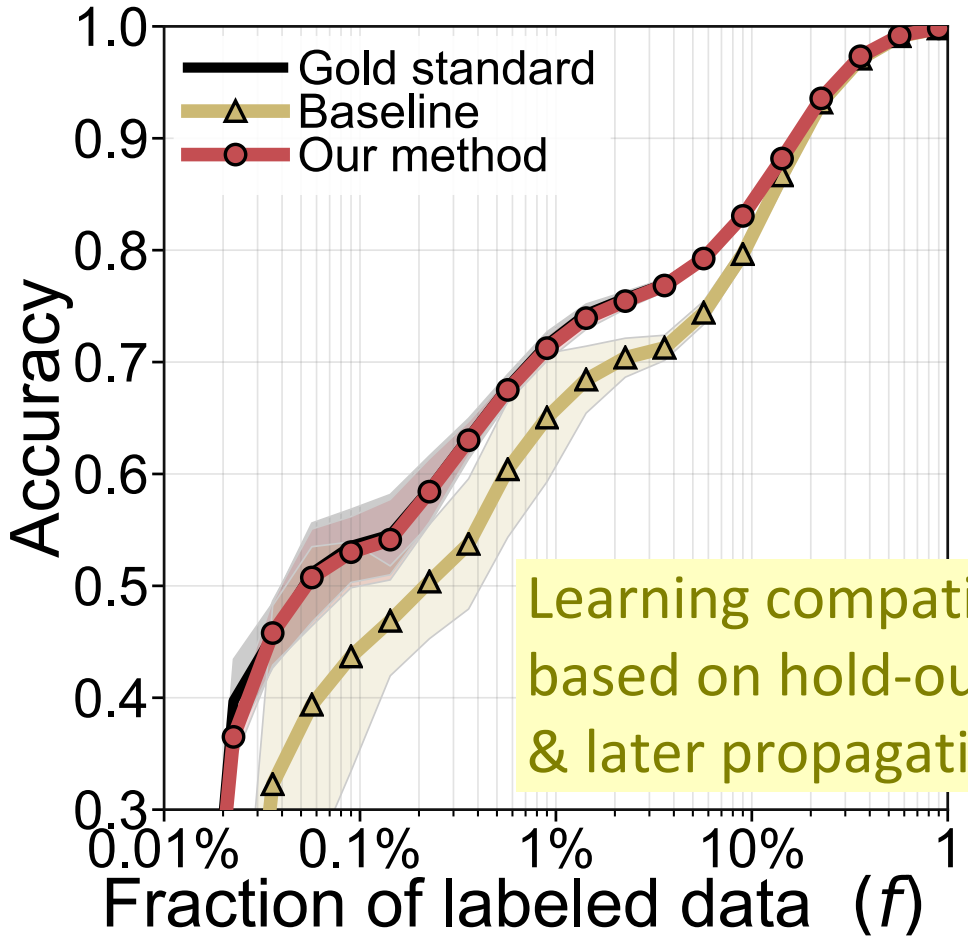


Learning compatibilities based on hold-out sets & later propagating them

Gold standard: Assume we could estimate the compatibilities on the fully labeled graph, then use those to label the rest

← Fewer labels

How well does it work?

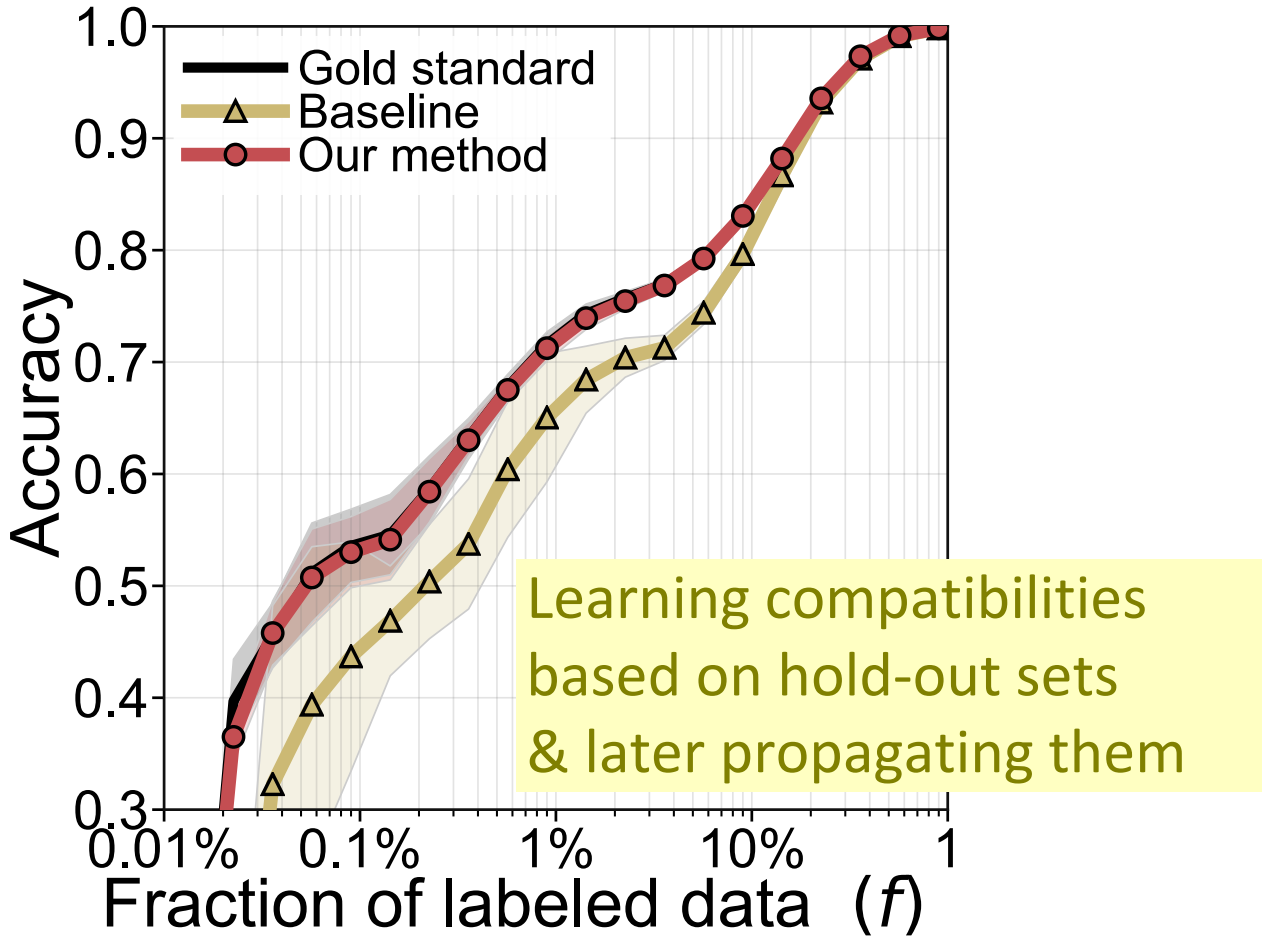


Learning compatibilities based on hold-out sets & later propagating them

Gold standard: Assume we could estimate the compatibilities on the fully labeled graph, then use those to label the rest

Labeling accuracy as if estimated on fully labeled graph

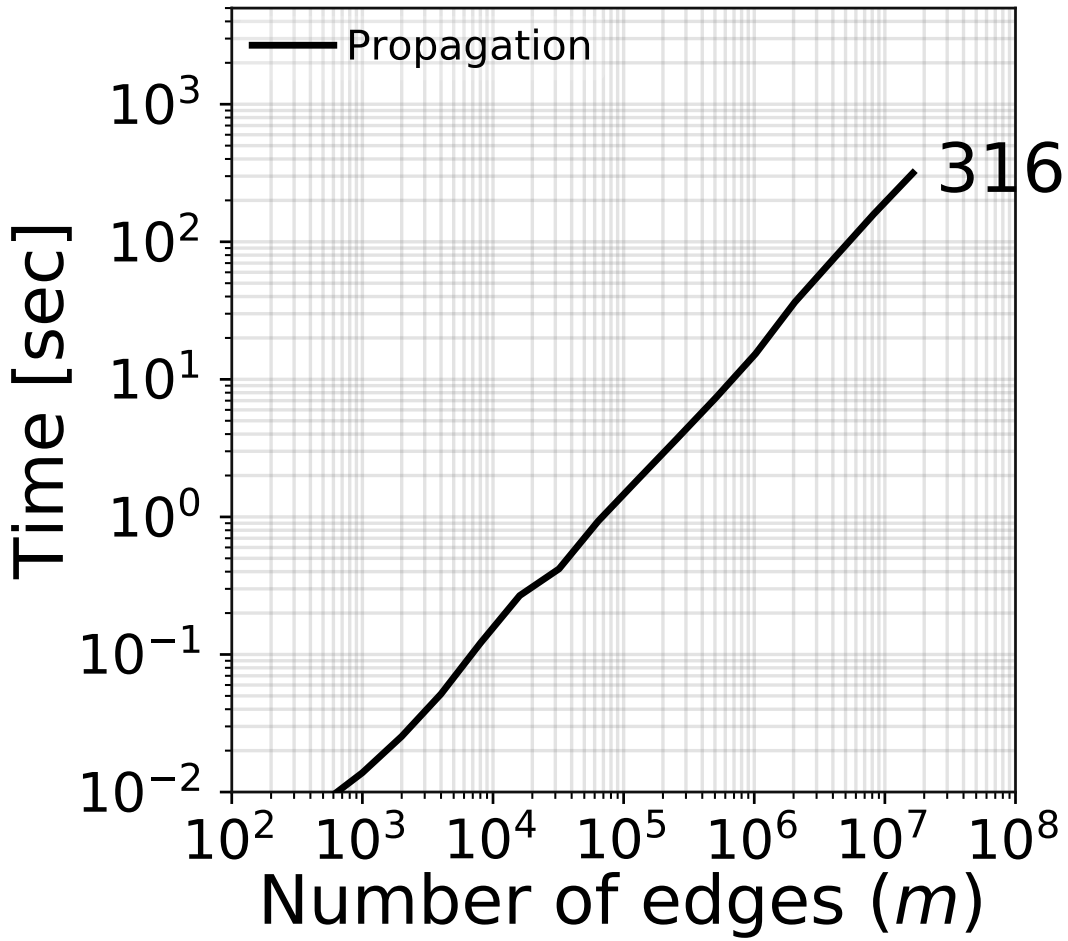
How well does it work?



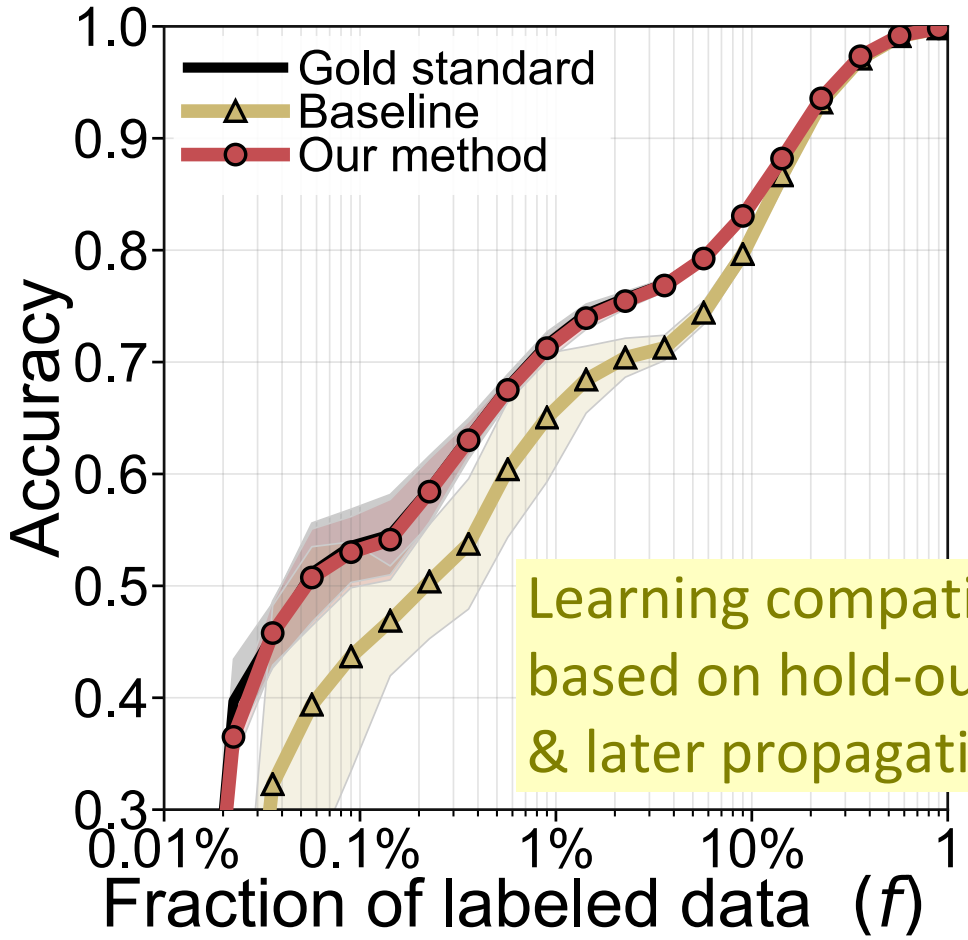
Learning compatibilities
based on hold-out sets
& later propagating them

Labeling accuracy as if estimated
on fully labeled graph

And how fast?



How well does it work?

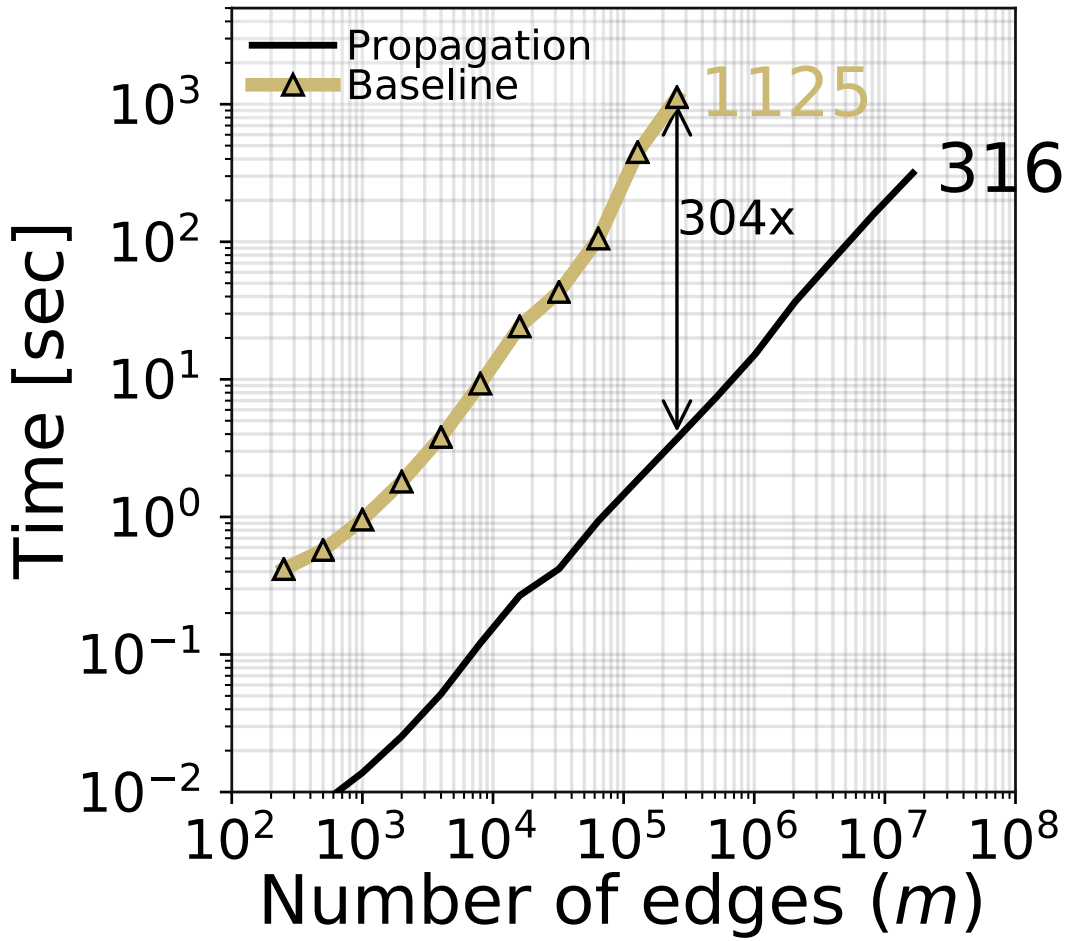


Learning compatibilities
based on hold-out sets
& later propagating them

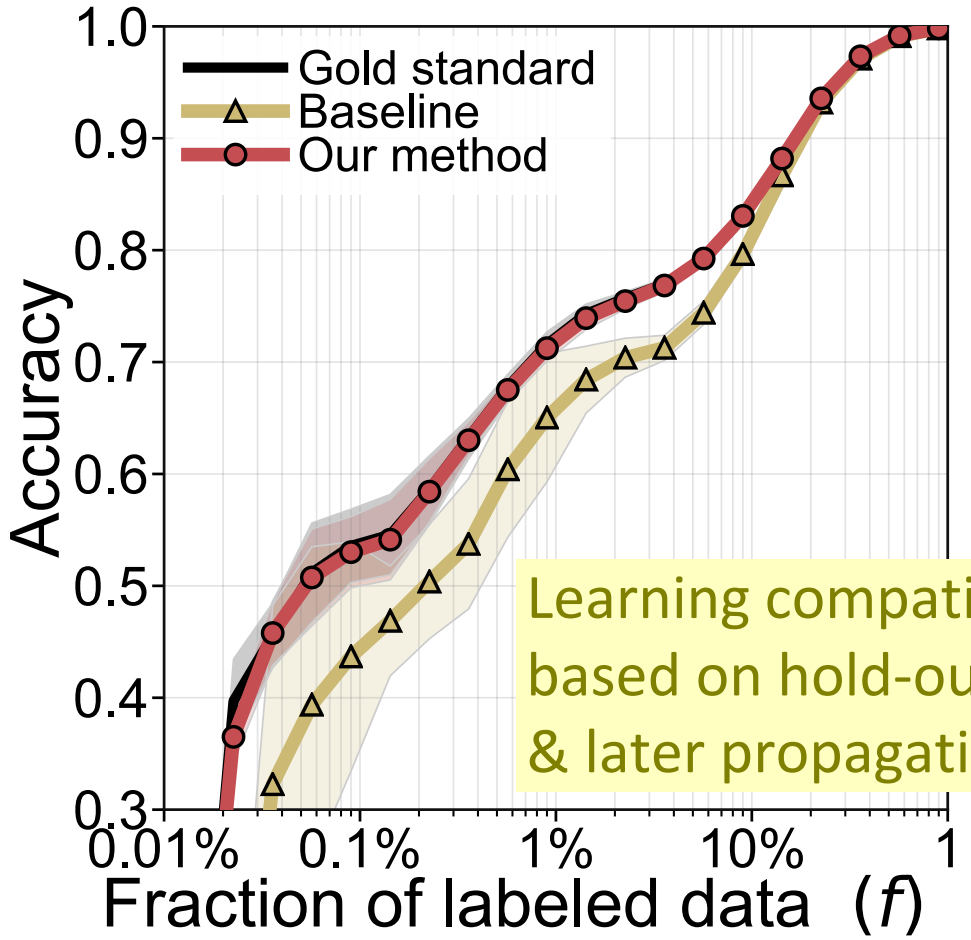
Labeling accuracy as if estimated
on fully labeled graph

And how fast?

Learning uses inference as subroutine (thus slower) ☹



How well does it work?

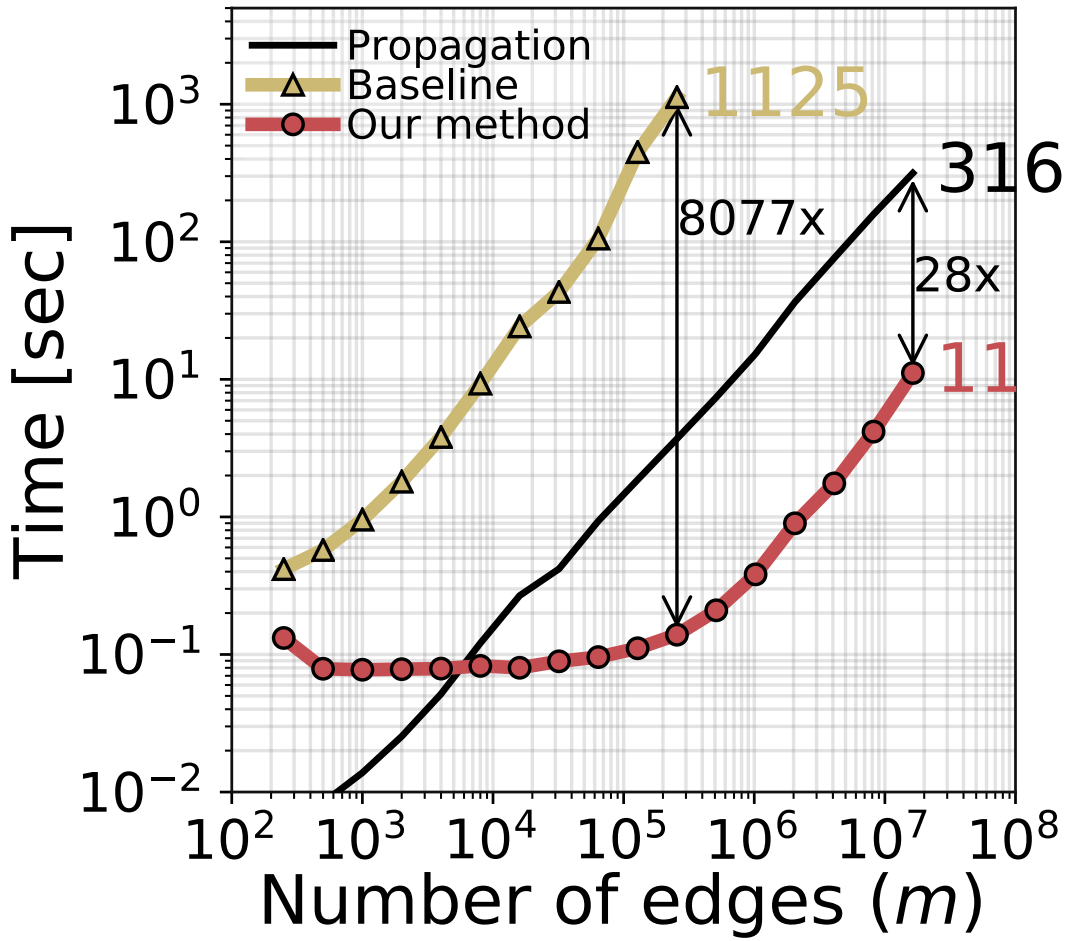


Learning compatibilities based on hold-out sets & later propagating them

Labeling accuracy as if estimated on fully labeled graph

And how fast?

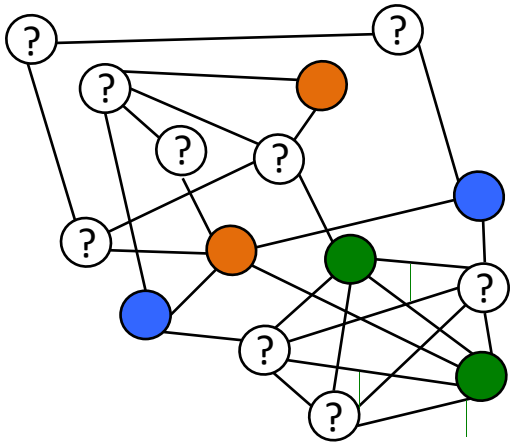
Learning uses inference as subroutine (thus slower) ☹



Our method needs <5% of time for labeling. No more need for heuristics or domain experts ☺

What is the trick?

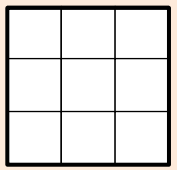
Overall approach



Sparsely labeled network

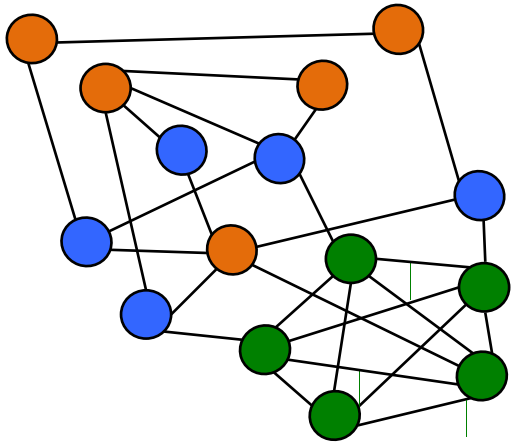
Compatibility Estimation

Compatibility matrix



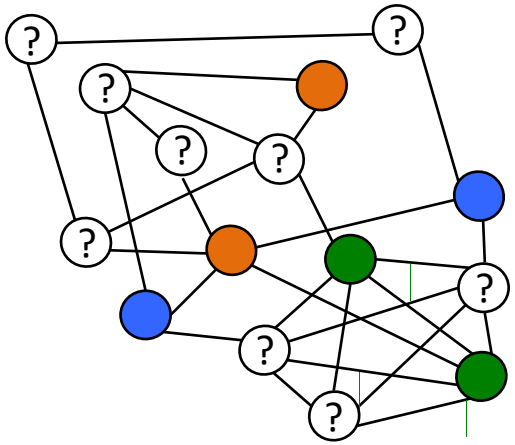
$k \times k$ matrix

Label Propagation



Fully labeled network

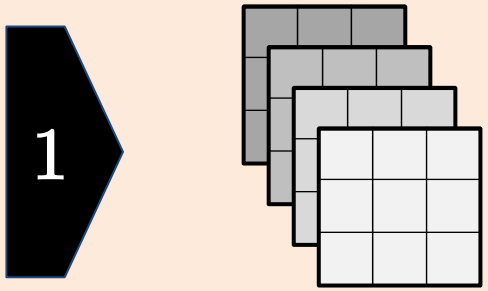
Overall approach



Sparsely labeled network

Compatibility Estimation

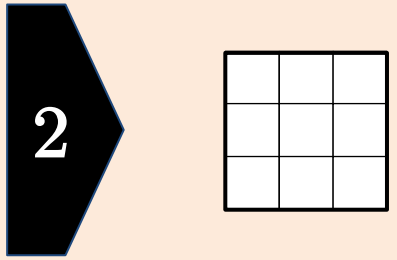
Derived statistics for path lengths 1,2,…,ℓ



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

Factorized graph representations

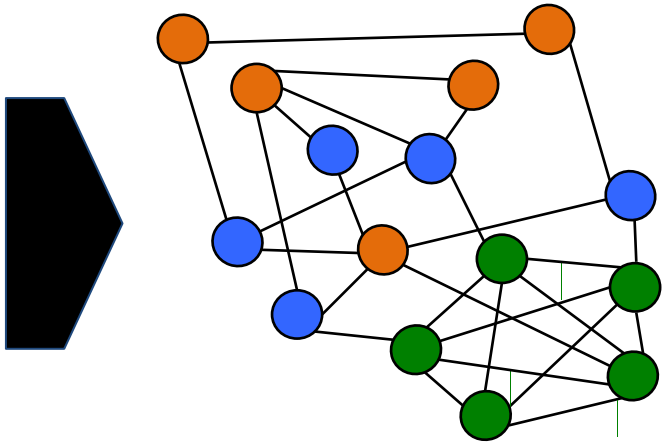
Compatibility matrix



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

Optimization

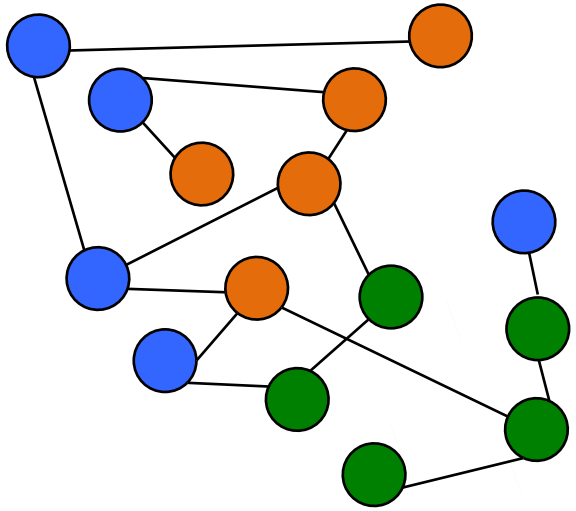
Label Propagation



Fully labeled network

A myopic view: counting relative neighbor frequencies

Fully labeled graph



Neighbor count

$\mathbf{M}=$

2	6	2
6	2	2
2	2	6

$\Rightarrow$ normalize

Compatibilities

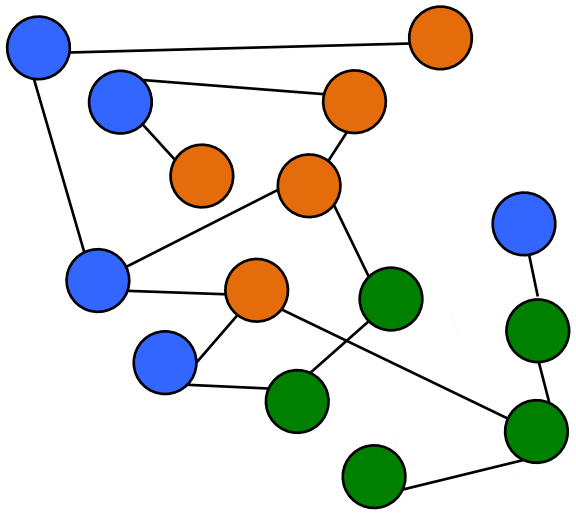
$\mathbf{H}=$

0.2	0.6	0.2
0.6	0.2	0.2
0.2	0.2	0.6

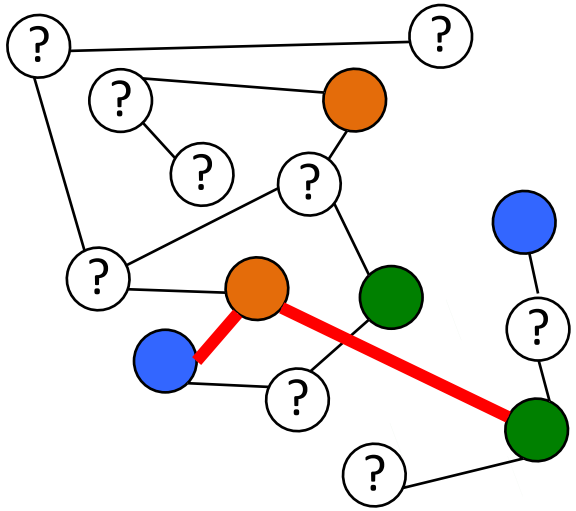
$\Sigma=1$

A myopic view: counting relative neighbor frequencies

Fully labeled graph



Sparsely labeled graph



Few nodes ⇒ even fewer edges

Prop. to [f=fraction of labels]^2 ☹

Neighbor count

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

normalize ⇒

Compatibilities

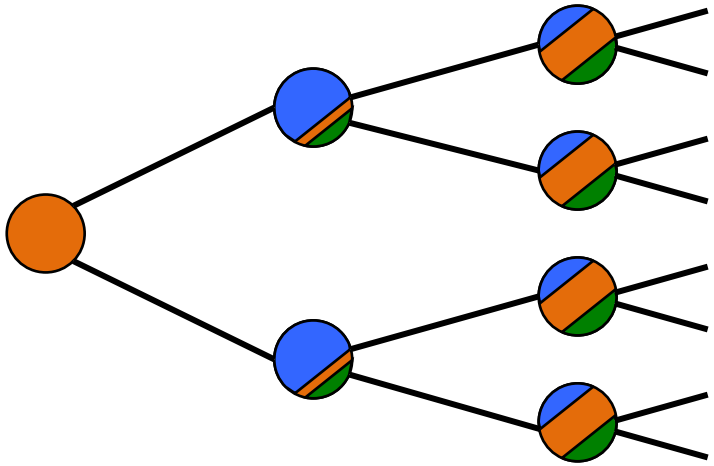
$$\mathbf{H}=\begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Σ=1

$$\widehat{\mathbf{M}}=\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Distant compatibility estimation (DCE)

0.2	0.6	0.2
0.6	0.2	0.2
0.2	0.2	0.6

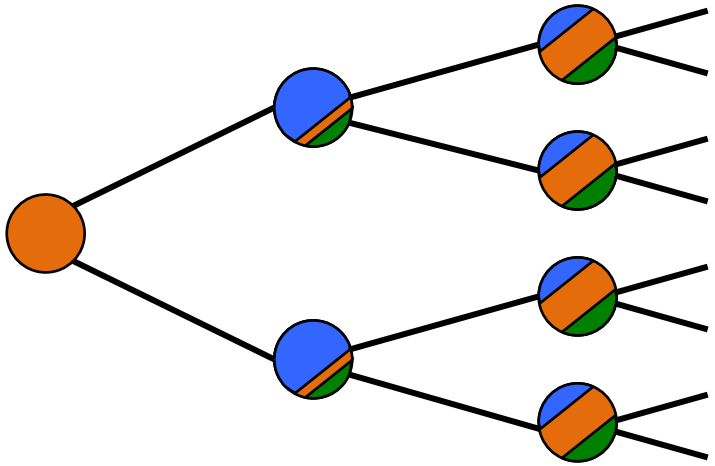


0	0.6	0.28	0.38
1	0.2	0.44	0.31
0	0.2	0.28	0.31

Expected signals for neighbors

Distant compatibility estimation (DCE)

0.2	0.6	0.2
0.6	0.2	0.2
0.2	0.2	0.6



graph with:

- m edges
- f labeled nodes
- d node degree

$d^{L-1}mf^2$ expected neighbors of distance L

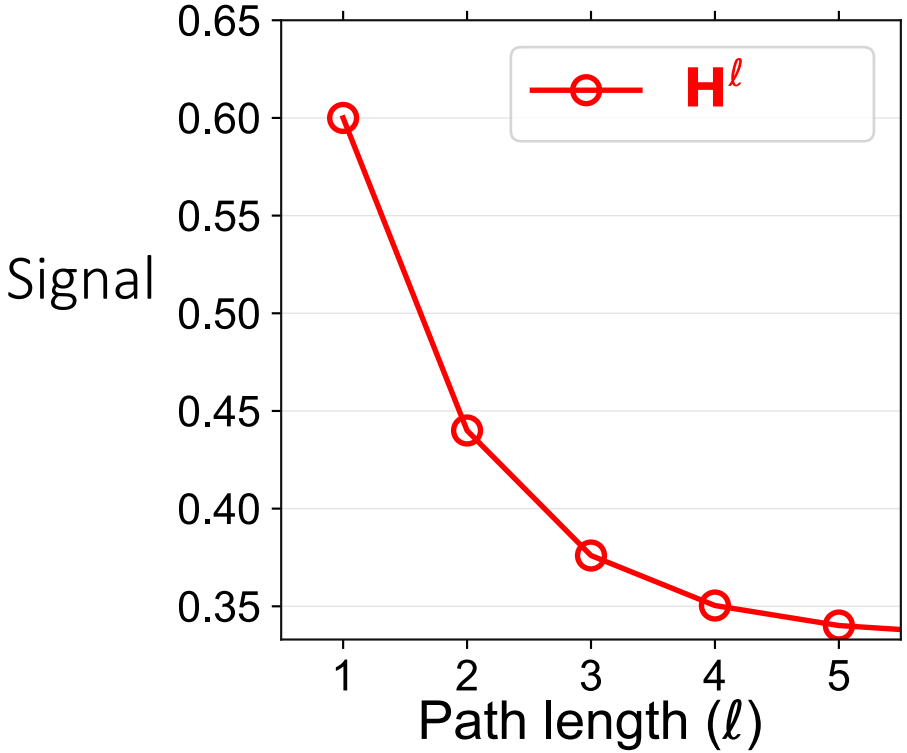
0	0.6	0.28	0.38
1	0.2	0.44	0.31
0	0.2	0.28	0.31

Expected signals for neighbors

Idea: amplify the signal from observed length-L compatibility matrices $\mathbf{H}^2$, $\mathbf{H}^3$, … $\mathbf{H}^L$ ☺

Two technical difficulties

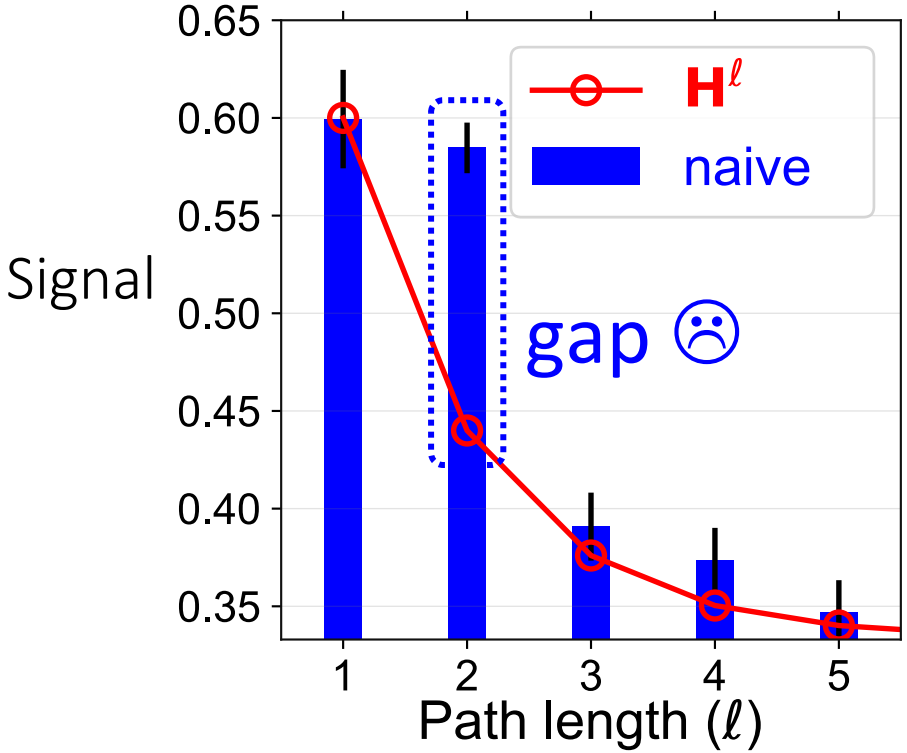
Longer paths dampen signal



Two technical difficulties

Longer paths dampen signal

1. Idea from previous page gives biased estimates ☹

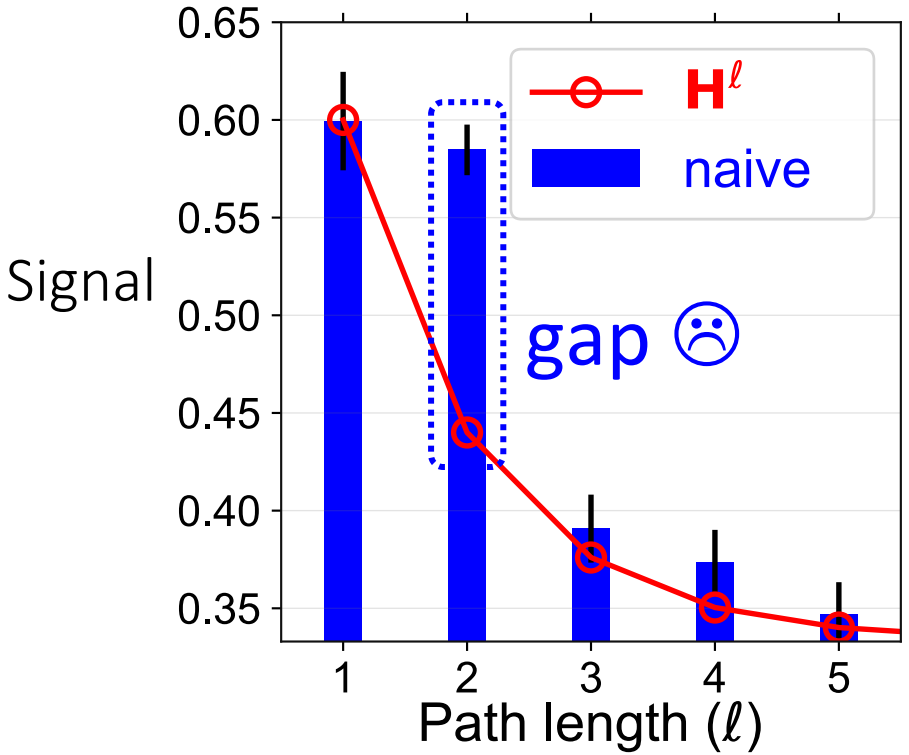


?

Two technical difficulties

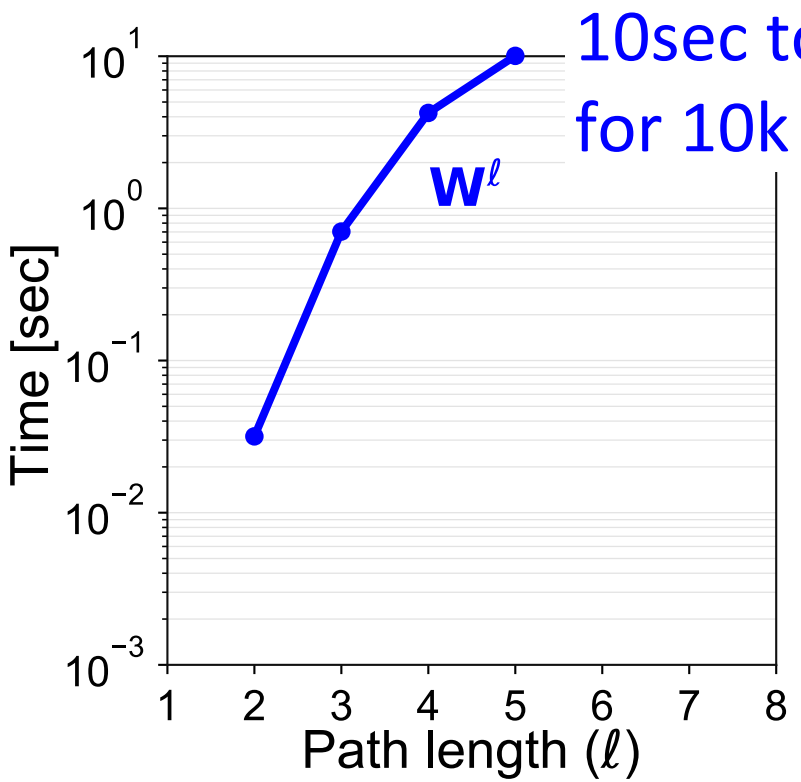
Longer paths dampen signal

1. Idea from previous page gives biased estimates ☹



?

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

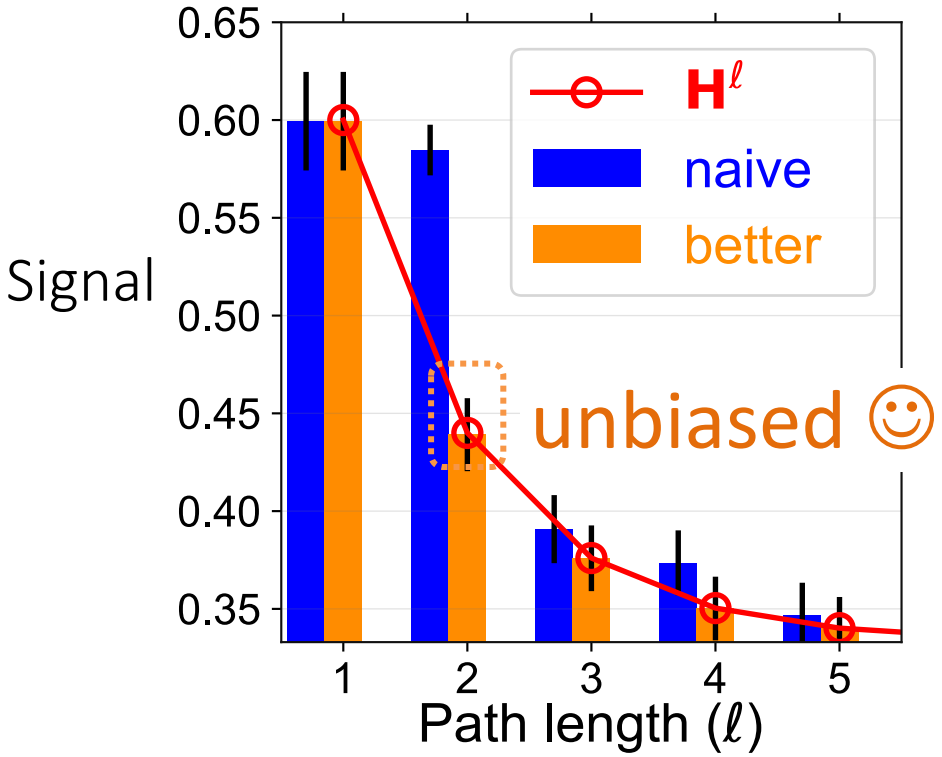


?

More careful algebra!

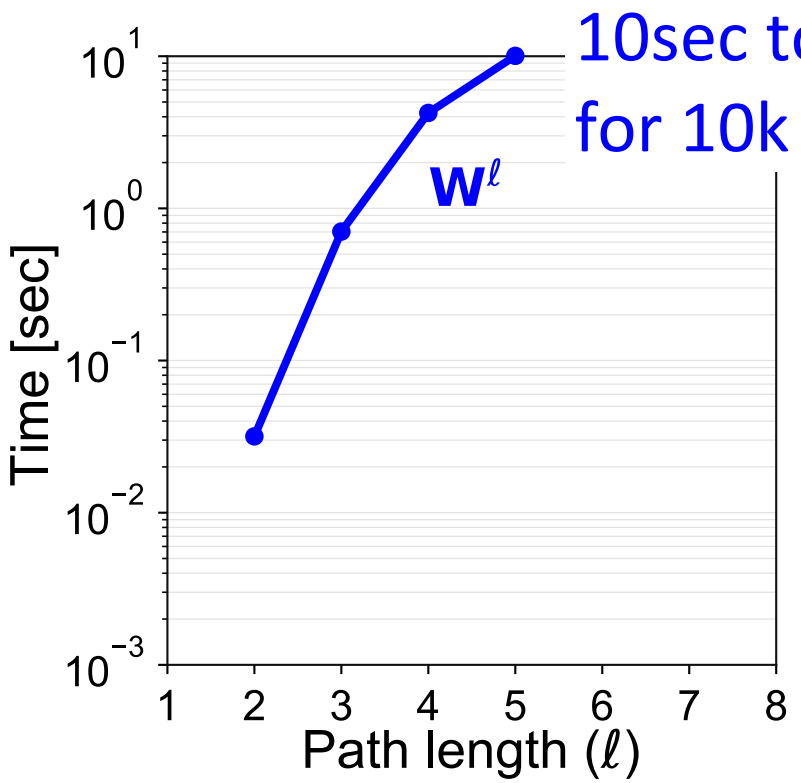
Longer paths dampen signal

1. Idea from previous page gives biased estimates ☹



1. We must only consider non-backtracking paths (requires more careful path aggregation)

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

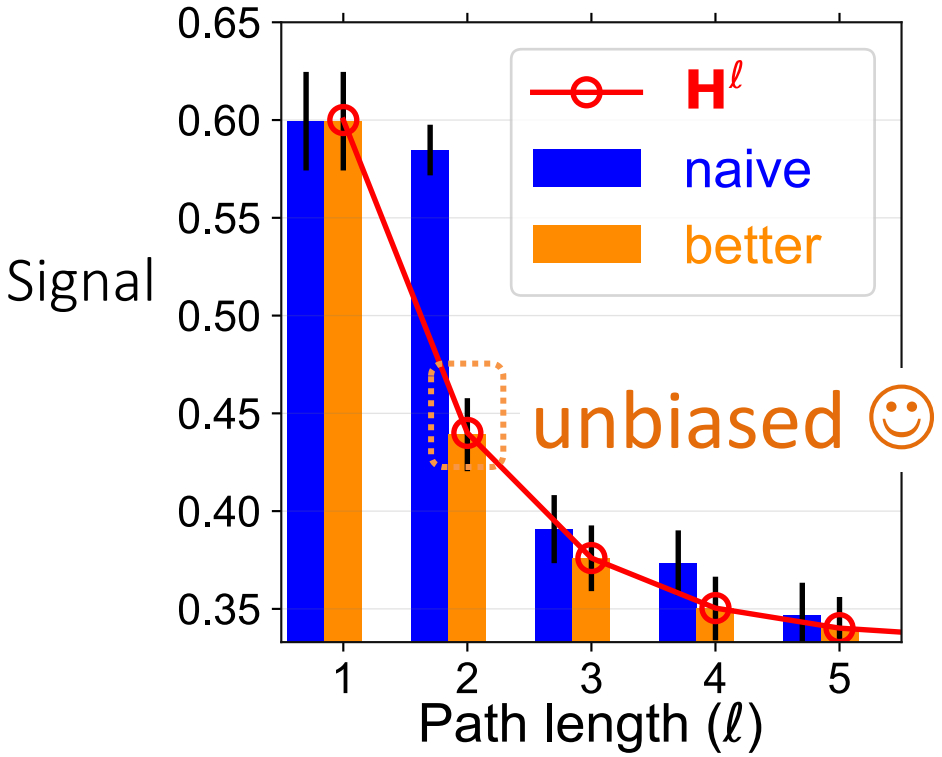


?

More careful algebra!

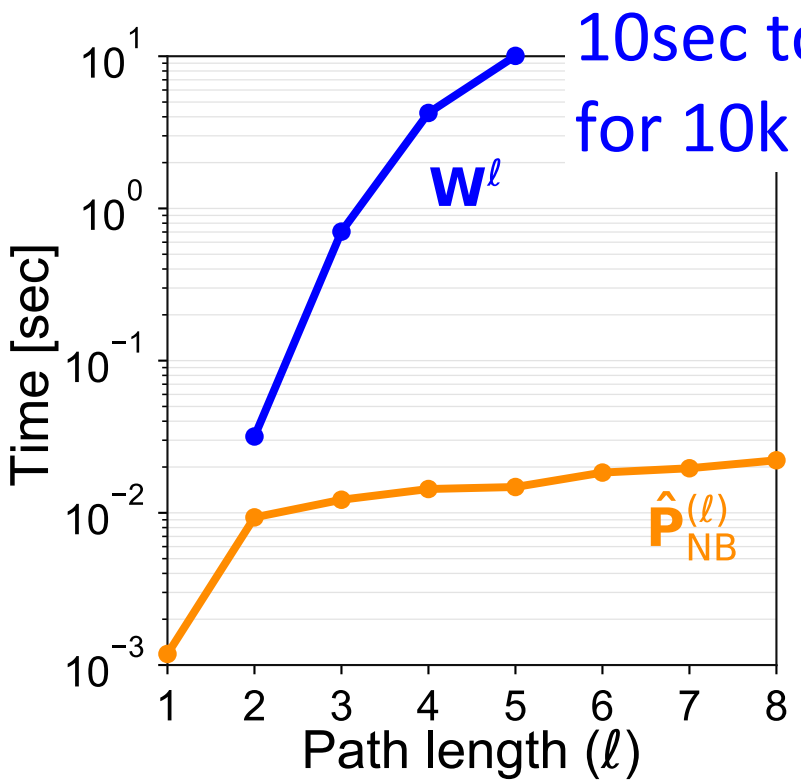
Longer paths dampen signal

1. Idea from previous page gives biased estimates ☹



1. We must only consider non-backtracking paths (requires more careful path aggregation)

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



10sec too long for 10k nodes ☹

10^{14} paths in 200 msec ☺ ☺ ☺

2. Requires more careful re-factorization of the calculation

Scalable, Factorized Path summation

Details

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

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

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

ALGORITHM 4.3 (FACTORIZED PATH SUMMATION). *Iteratively calculate the graph summaries* $\hat{\mathbf{P}}_{\text{NB}}^{(\ell)}$, *for* $\ell \in [\ell_{\max}]$ *as follows:*

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

(2) *Calculate* $\mathbf{M}_{\text{NB}}^{(\ell)} = \mathbf{X}^{\top}\mathbf{N}_{\text{NB}}^{(\ell)}$.

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

PROPOSITION 4.4 (FACTORIZED PATH SUMMATION). *Algorithm 4.3 calculates all graph statistics* $\hat{\mathbf{P}}_{\text{NB}}^{(\ell)}$ *for* $\ell \in [\ell_{\max}]$ *in* $\boxed{O(mk\ell_{\max})}$.

Intuition

$$\pi_y(\mathrm{R}(\mathrm{x}) \bowtie \mathrm{S}(\mathrm{x}, \mathrm{y}))$$

$$\Rightarrow \quad \mathrm{R}(\mathrm{x}) \bowtie \pi_y \mathrm{S}(\mathrm{x}, \mathrm{y})$$

Scalable, Factorized Path summation

Details

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

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

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

ALGORITHM 4.3 (FACTORIZED PATH SUMMATION). *Iteratively calculate the graph summaries* $\hat{\mathbf{P}}_{\text{NB}}^{(\ell)}$, *for* $\ell \in [\ell_{\max}]$ *as follows:*

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

(2) *Calculate* $\mathbf{M}_{\text{NB}}^{(\ell)} = \mathbf{X}^{\top}\mathbf{N}_{\text{NB}}^{(\ell)}$.

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

PROPOSITION 4.4 (FACTORIZED PATH SUMMATION). *Algorithm 4.3 calculates all graph statistics* $\hat{\mathbf{P}}_{\text{NB}}^{(\ell)}$ *for* $\ell \in [\ell_{\max}]$ *in* $O(mk\ell_{\max})$.

Intuition

Relational algebra

$$\pi_y(\mathrm{R}(\mathrm{x}) \bowtie \mathrm{S}(\mathrm{x}, \mathrm{y}))$$

$$\Rightarrow \quad \mathrm{R}(\mathrm{x}) \bowtie \pi_y \mathrm{S}(\mathrm{x}, \mathrm{y})$$

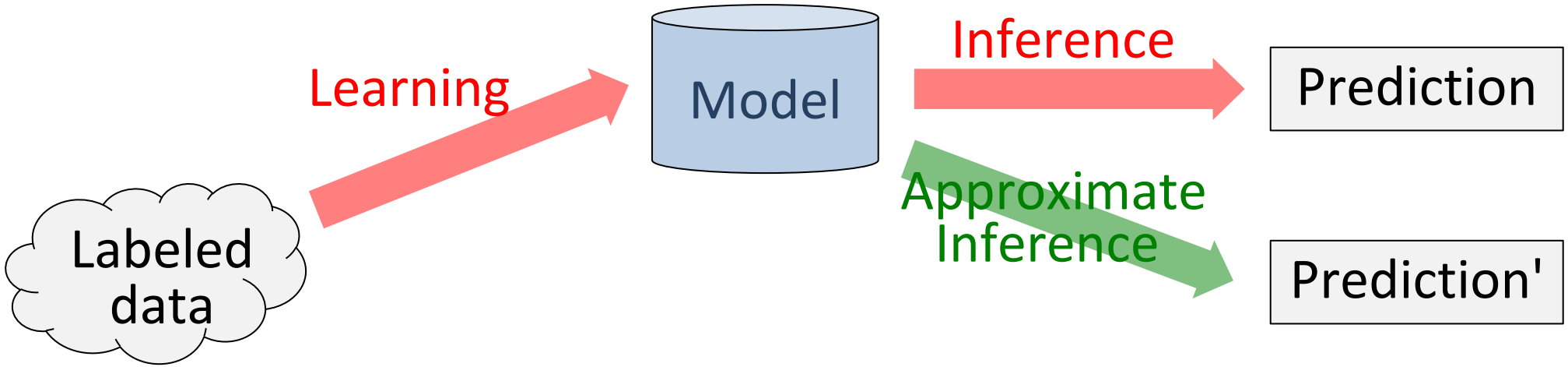
(**X** = thin label matrix)

Linear algebra

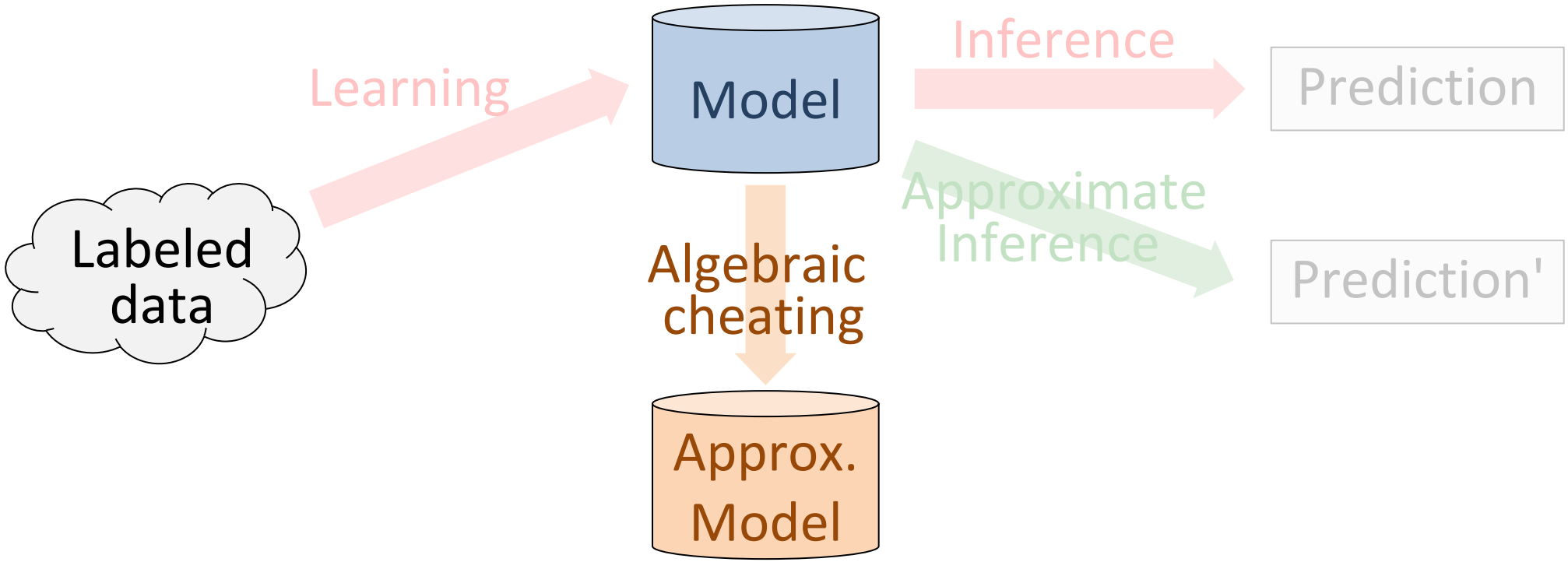
$$(\mathrm{W} \cdot \mathrm{W}) \cdot \mathrm{X}$$

$$\Rightarrow \quad \mathrm{W} \cdot (\mathrm{W} \cdot \mathrm{X})$$

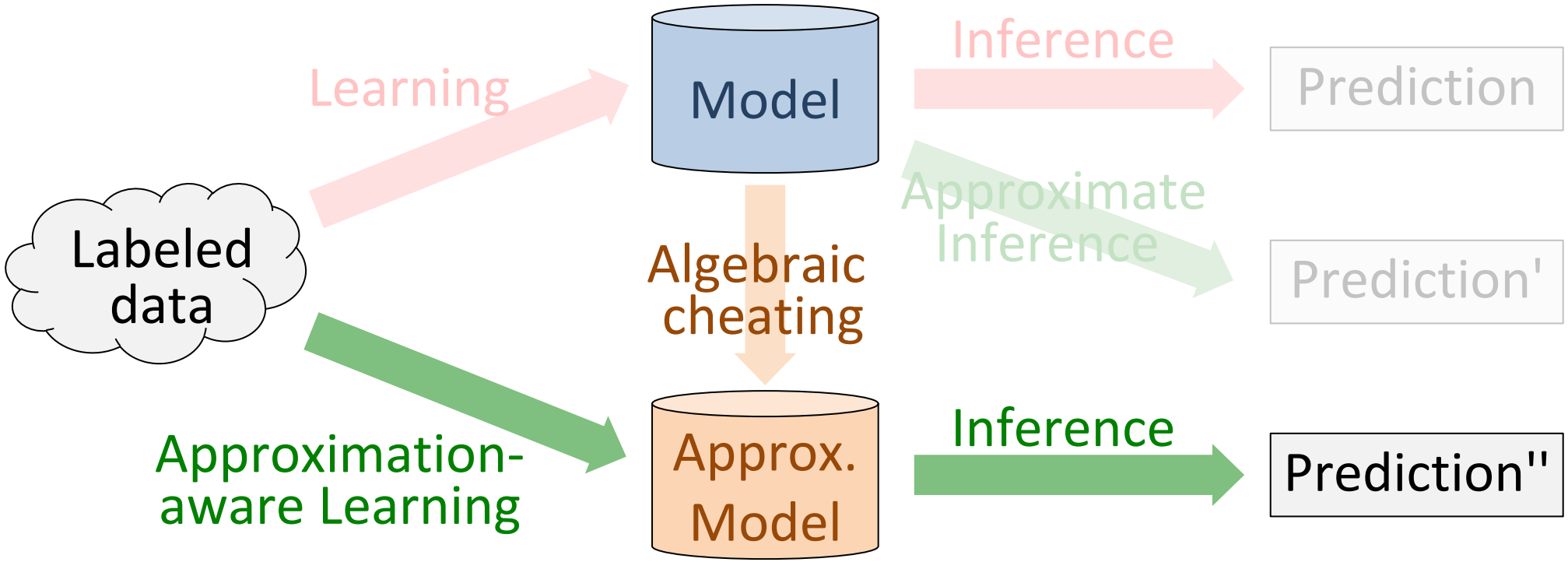
"Algebraic cheating" for Approximate-aware learning



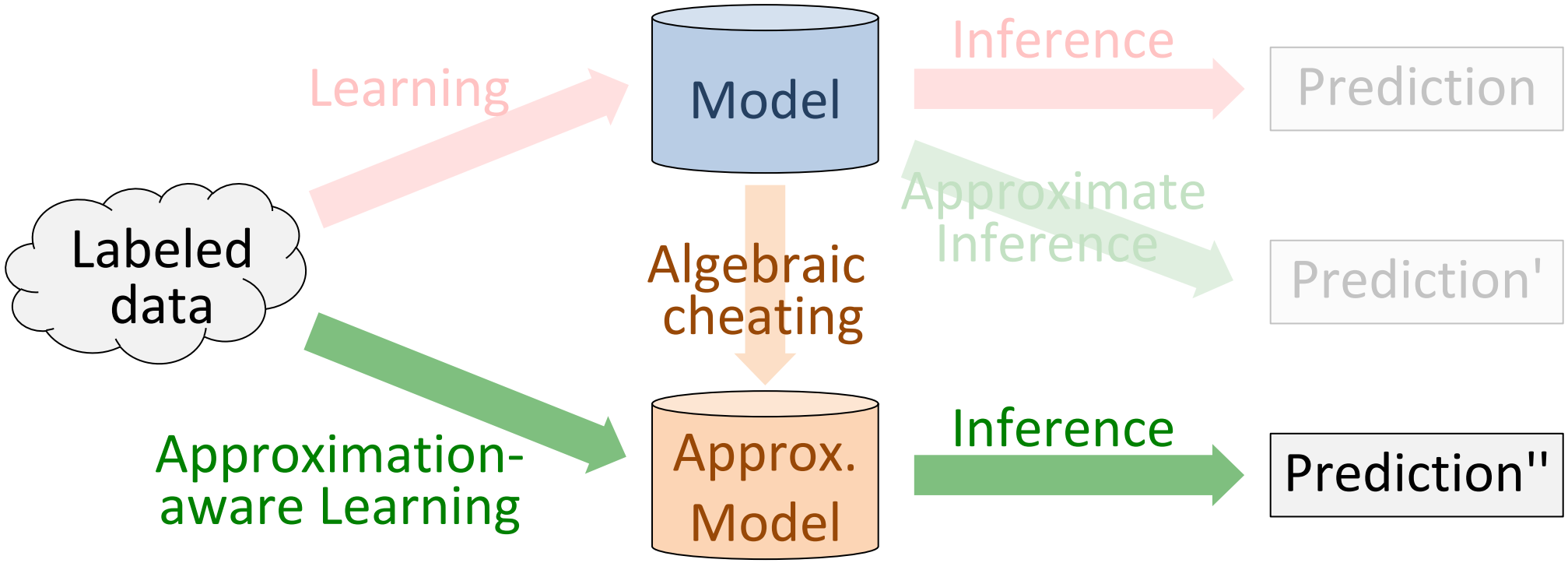
"Algebraic cheating" for Approximate-aware learning



"Algebraic cheating" for Approximate-aware learning



"Algebraic cheating" for Approximate-aware learning



Please come and talk to us today, or visit us for a talk

DATA Lab @ Northeastern

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

Thank you ☺

[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 in sparsely labeled graphs
Thanks to NSF for IIS-1762268-CAREER: Scaling approximate inference and approximation-aware learning