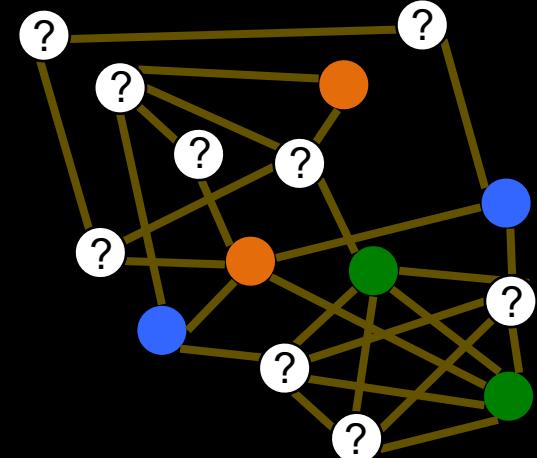


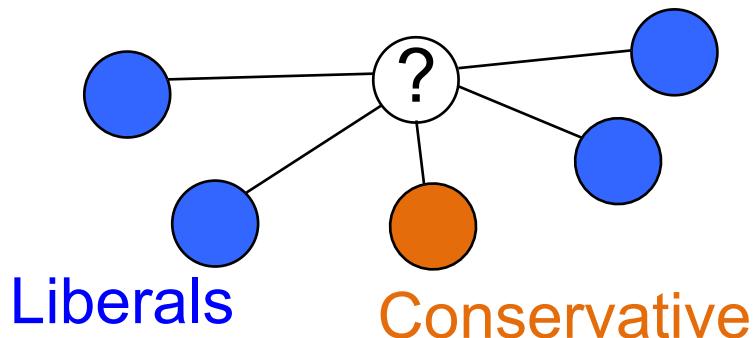
Linearized and Single-Pass Belief Propagation



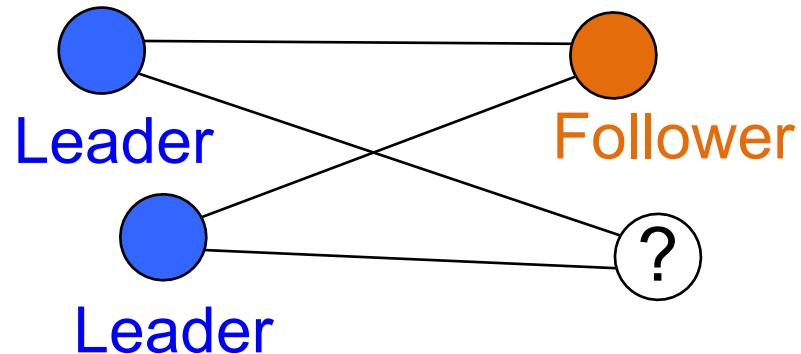
Wolfgang Gatterbauer, Stephan Günnemann¹,
Danai Koutra², Christos Faloutsos

VLDB 2015 (Sept 2, 2015)

"Birds of a feather..." (Homophily)



"Opposites attract" (Heterophily)



"Guilt-by-association"

- recommender systems
- semi-supervised learning
- random walks (PageRank)

Disassortative networks

- biological food web
- protein interaction
- online fraud settings

Affinity matrix (also potential or heterophily matrix)

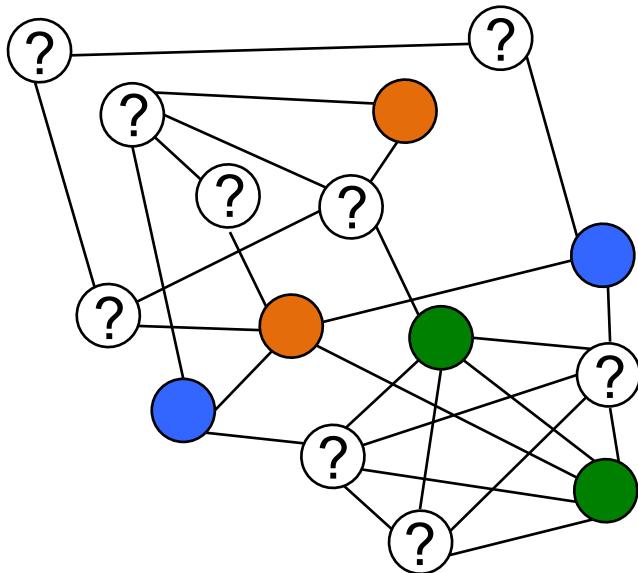
$$H = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix} \end{matrix}$$

The matrix H is a 2x2 matrix representing the affinity between two groups (1 and 2). The diagonal elements are 0.8, and the off-diagonal elements are 0.2. A red circle highlights the element at position (1,1).

$$H = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \end{matrix}$$

The matrix H is a 2x2 matrix representing the affinity between two groups (1 and 2). The diagonal elements are 0.2, and the off-diagonal elements are 0.8. A red circle highlights the element at position (2,1).

The overall problem we are trying to solve



$$\mathbf{H} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 0.1 & 0.8 & 0.1 \\ 0.8 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{matrix} \end{matrix}$$

k=3 classes

Problem formulation

Given:

- undirected graph G
- seed labels
- affinity matrix H
(symmetric)

Find:

- remaining labels

Good: Graphical models
can model this problem



Bad: Belief Propagation
on graphs with loops
has convergence issues



Roadmap

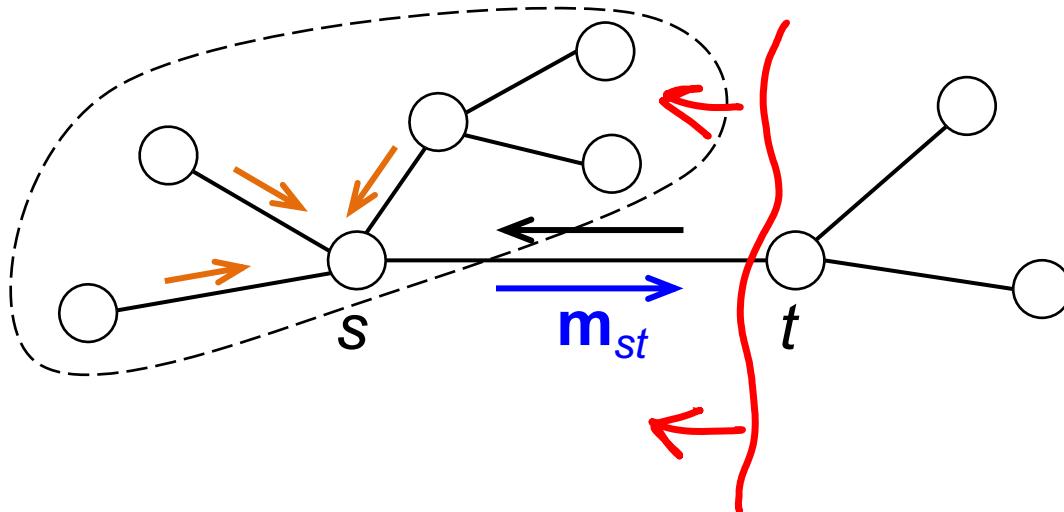
1) **Background: "BP"**
Belief Propagation

2) Our solution: "LinBP"
Linearized Belief Propagation

3) Experiments & conclusions

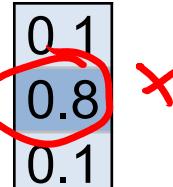
Belief Propagation

Pearl [1988]



Message m_{st} from s to t summarizes everything s knows except information obtained from t ("echo cancellation" EC)

Belief Propagation

- Iterative approach that sends messages between nodes
- Repeat until messages converge
- Result is label distribution ("beliefs") at each node $\mathbf{b}_t = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.1 \end{bmatrix}$ 

Problems on graphs with loops:

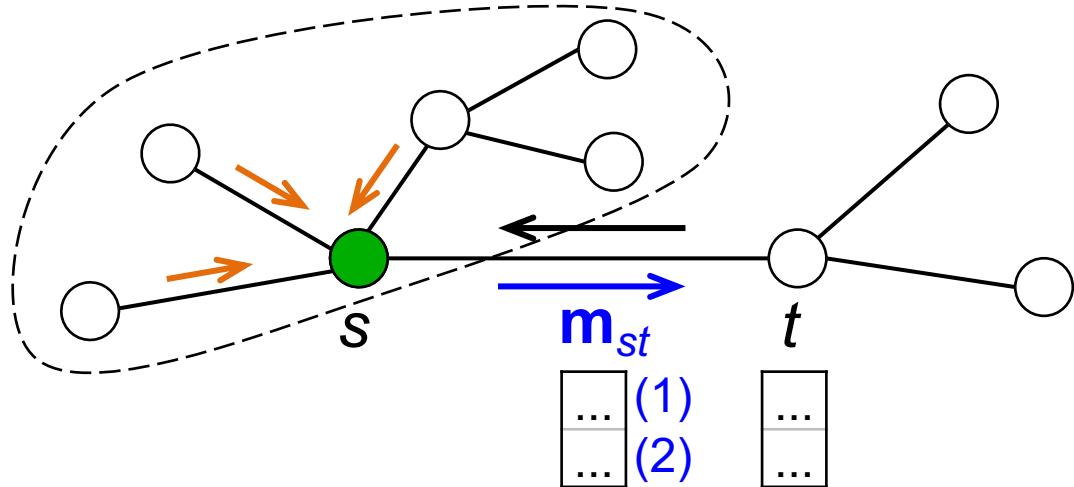
- Approximation with no guarantees of correctness
- Worse: algorithm may not even converge if graph has loops 😞
- Still widely used in practice

Belief Propagation

Pearl [1988]

Weiss [Neural C.'00]

DETAILS



- 1) Initialize all message entries to 1
- 2) Iteratively: calculate messages for each edge and class

$$m_{st}(i) \propto \sum_j H(j, i) e_s(j) \prod_{u \in N(s) \setminus t} m_{us}(j)$$

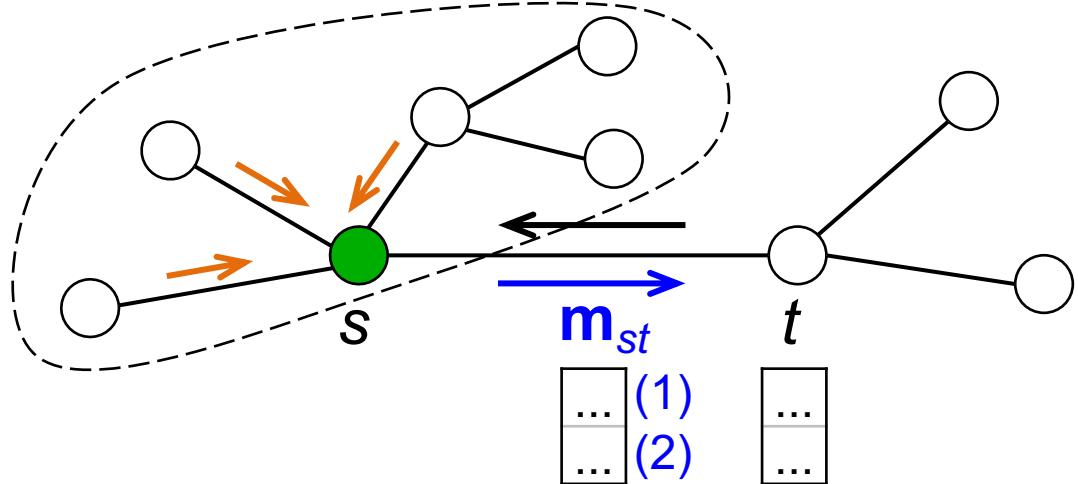
affinity \sum_j explicit beliefs "echo cancellation" EC

- 3) After messages converge: calculate final beliefs

$$b_s(i) \propto e_s(i) \prod_{u \in N(s)} m_{us}(i)$$

Belief Propagation

DETAILS



1) Initialize all message entries to 1

2) Iteratively: calculate messages for each edge and class

$$m_{st} \propto H(e_s \odot \underset{u \in N(s) \setminus t}{\bullet} m_{us})$$

affinity matrix explicit beliefs component-wise multiplication "echo cancellation" EC

3) After messages converge: calculate final beliefs

$$b_s \propto e_s \odot \underset{u \in N(s)}{\bullet} m_{us}$$

Often does not converge
on graphs with loops



Roadmap

1) Background: "BP"
Belief Propagation

2) Our **solution**: "LinBP"
Linearized Belief Propagation

3) Experiments & Conclusions

Key Ideas: 1) Centering + 2) Linearizing BP

Original Value = Center point + Residual

$$H = \left[\frac{1}{k} \right]_{k \times k}$$

$$\begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{matrix}$$

+

Residual ①

$$\hat{H} = \begin{matrix} -0.23 & 0.46 & -0.23 \\ 0.46 & -0.23 & -0.23 \\ -0.23 & -0.23 & 0.46 \end{matrix}$$

$$e, b = \frac{1}{k} \mathbf{1} \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

+

$$\hat{e}, \hat{b} = \begin{matrix} -0.13 \\ 0.26 \\ -0.13 \end{matrix}$$

$$m = \mathbf{1} \begin{matrix} 1 \\ 1 \\ 1 \end{matrix}$$

+

$$\hat{m} = \begin{matrix} 0.1 \\ -0.2 \\ 0.1 \end{matrix}$$

②

Formula Approximation

Logarithm	$\ln(1 + x)$	$\approx x \left(-\frac{x^2}{2} + \frac{x^3}{3} - \dots \right)$
-----------	--------------	---

Division	$\frac{\frac{1}{k} + x}{1+y}$	$\approx \frac{1}{k} + x - \frac{y}{k}$
----------	-------------------------------	---

Linearized Belief Propagation

DETAILS

$$\mathbf{m}_{st} \propto H\left(\mathbf{e}_s \odot \bigodot_{u \in N(s) \setminus t} \mathbf{m}_{us}\right)$$

BP

not linear



$$\mathbf{b}_s \propto \mathbf{e}_s \odot \bigodot_{u \in N(s)} \mathbf{m}_{us}$$

$$\hat{\mathbf{m}}_{st} \leftarrow \hat{H}\left(\hat{\mathbf{e}}_s + \frac{1}{k} \cdot \sum_{u \in N(s) \setminus t} \hat{\mathbf{m}}_{us}\right)$$

LinBP

linear



$$\hat{\mathbf{b}}_s \leftarrow \hat{\mathbf{e}}_s + \frac{1}{k} \cdot \sum_{u \in N(s)} \hat{\mathbf{m}}_{us}$$

no more normalization necessary

Matrix formulation of LinBP

DETAILS

Update equation

$$\hat{\mathbf{B}} \leftarrow \hat{\mathbf{E}} + \mathbf{W} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{H}} - \underbrace{\mathbf{D} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{H}}^2}_{\text{EC}}$$

final beliefs (n x k) explicit beliefs graph affinity matrix degree matrix EC

The diagram illustrates the update equation for LinBP. It shows the components of the equation as matrices: t (n x k), $\hat{\mathbf{E}}$ (n x k), \mathbf{W} (n x m), $\hat{\mathbf{B}}$ (n x m), $\hat{\mathbf{H}}$ (m x m), \mathbf{D} (m x m), and $\hat{\mathbf{H}}^2$ (m x m). The diagram shows the addition of $\hat{\mathbf{E}}$ and the product of \mathbf{W} and the product of $\hat{\mathbf{B}}$ and $\hat{\mathbf{H}}$. It also shows the subtraction of the product of \mathbf{D} and the product of $\hat{\mathbf{B}}$ and $\hat{\mathbf{H}}^2$. Labels include S , t , and EC .

Closed form

$$\text{vec}(\hat{\mathbf{B}}) = (\mathbf{I} - \hat{\mathbf{H}} \otimes \mathbf{W} + \hat{\mathbf{H}}^2 \otimes \mathbf{D})^{-1} \text{vec}(\hat{\mathbf{E}})$$

EC

The diagram illustrates the closed form equation for LinBP. It shows the matrix multiplication and summation in the equation. A red bracket underlines the term $(\mathbf{I} - \hat{\mathbf{H}} \otimes \mathbf{W} + \hat{\mathbf{H}}^2 \otimes \mathbf{D})$ and a blue bracket underlines the term $\hat{\mathbf{H}}^2 \otimes \mathbf{D}$. The label EC is placed above the blue bracket.

Convergence Spectral radius of (...) < 1

Scale with appropriate ϵ : $\hat{\mathbf{H}} \leftarrow \epsilon \cdot \hat{\mathbf{H}}_0$

Roadmap

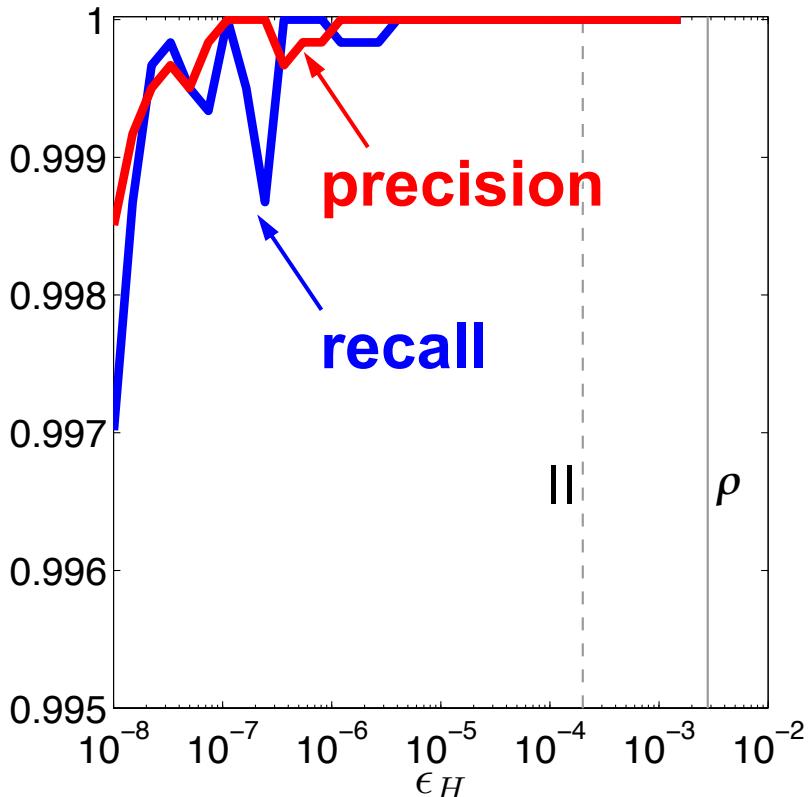
1) Background: "BP"
Belief Propagation

2) Our solution: "LinBP"
Linearized Belief Propagation

3) Experiments & Conclusions

Experiments

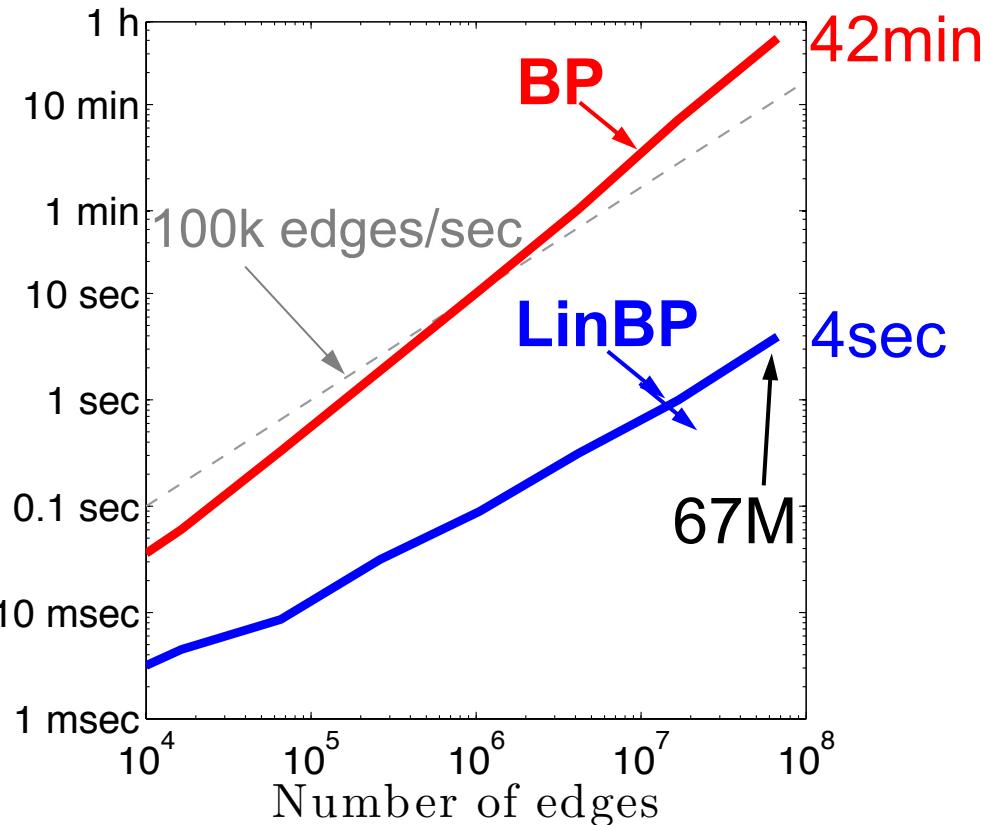
1) Great labeling accuracy (LinBP against BP as ground truth)



LinBP & BP give similar labels

Reason: when BP converges, then our approximations are reasonable

2) Linear scalability (10 iterations)



LinBP is seriously faster!

Reason: LinBP can leverage existing optimized linear algebra packages

What else you will find in the paper and TR

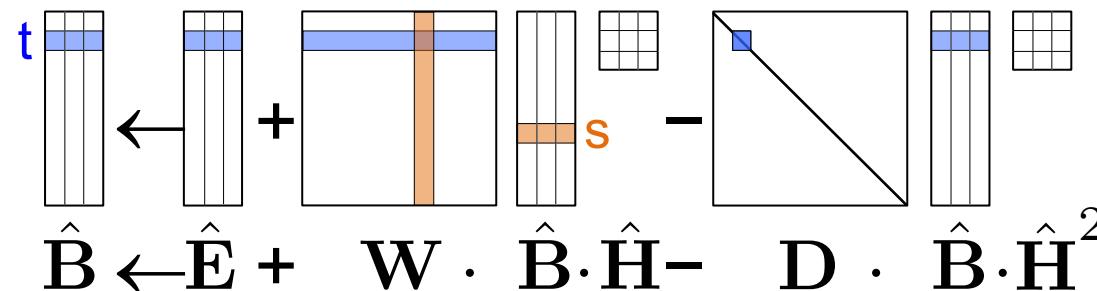
- Theory
 - Full derivation
- SQL (w/ recursion)
 - Implementation with standard aggregates
- Single-pass Belief Propagation (SBP)
 - Myopic version that allows incremental updates
- Experiments with synthetic and DBLP data set

Take-aways

Goal: propagate multi-class heterophily from labels

Problem: How to solve BP convergence issues 😞

Solution: Center & linearize BP \Rightarrow convergence & speed 😊



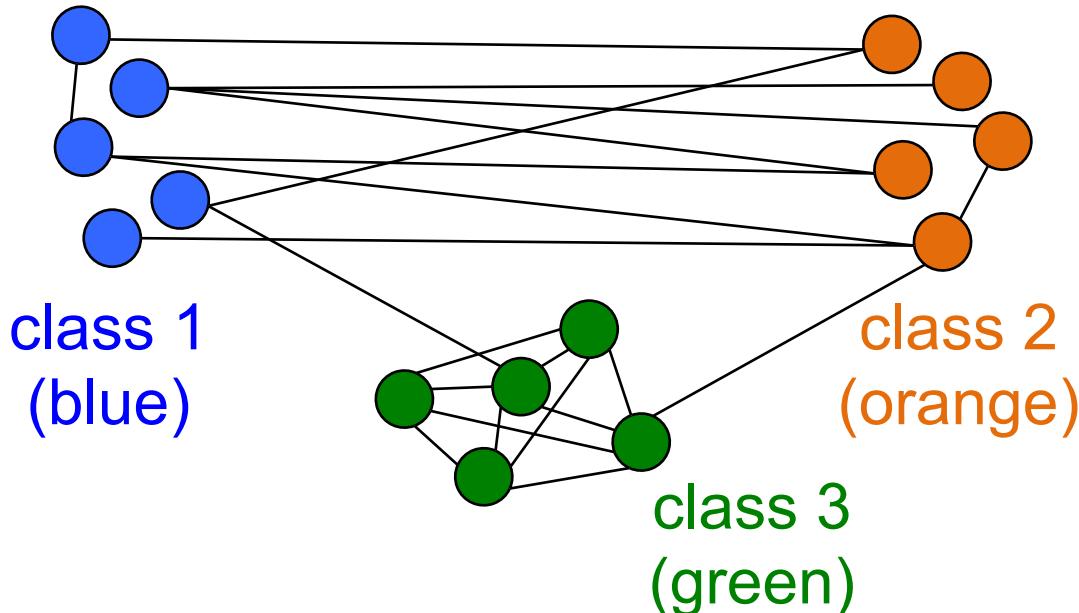
- Linearized Belief Propagation (LinBP)
 - Matrix Algebra, convergence, closed-form
 - SQL (w/ recursion) with standard aggregates
- Single-pass Belief Propagation (SBP)
 - Myopic version, incremental updates

<http://github.com/sslh/linBP/>

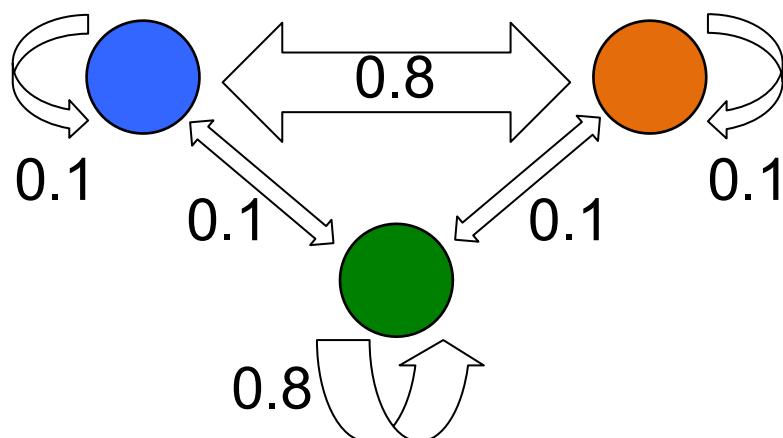
Thanks 😊

BACKUP

More general forms of "Heterophily"



Class-to-class interaction



Potential (P)

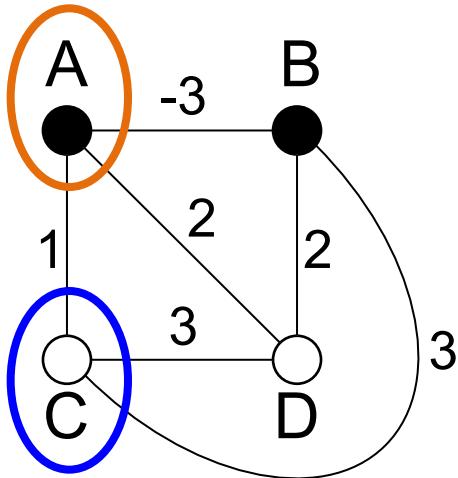
	1	2	3
1	1	8	1
2	8	1	1
3	1	1	8

Affinity matrix (H)

	1	2	3
1	0.1	0.8	0.1
2	0.8	0.1	0.1
3	0.1	0.1	0.8

BP has convergence issues

DETAILED EXAMPLE



Edge potentials

$$\mathbf{H} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Edge weights

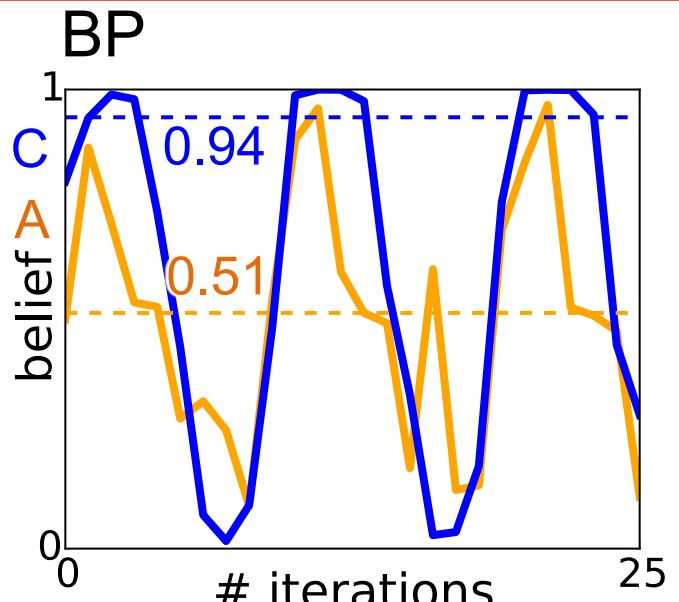
$$H_{ij}^e \propto H_{ij}^w$$

Explicit beliefs

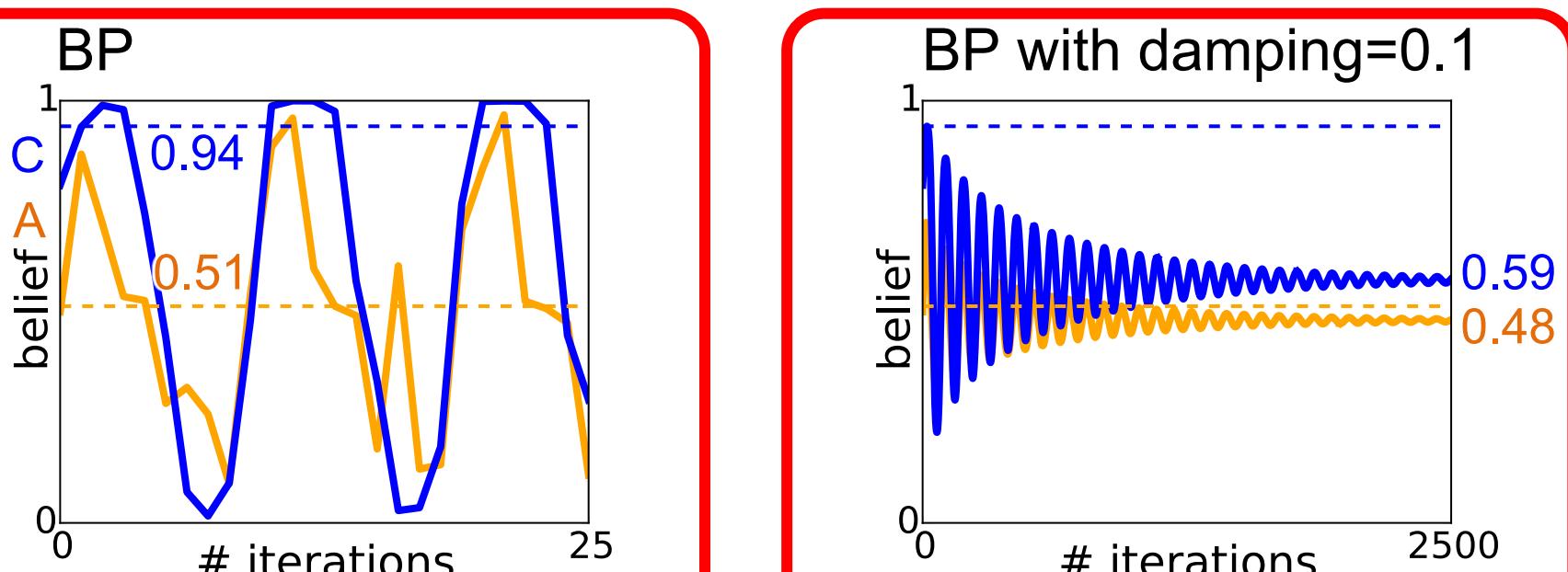
$$\mathbf{e}_A = \mathbf{e}_B = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$$

Nodes w/o priors:

$$\mathbf{e}_C = \mathbf{e}_D = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$



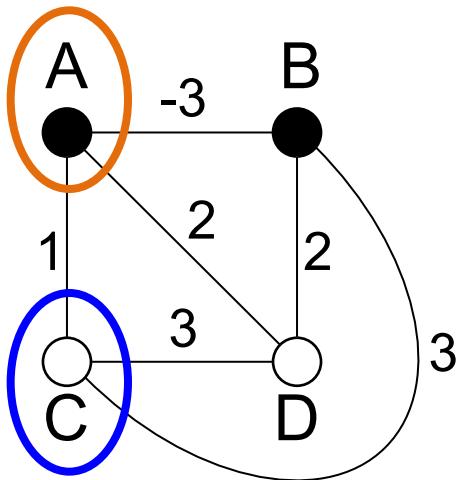
1) no guaranteed converge



2) damping: many iterations

BP has convergence issues

DETAILED EXAMPLE



Edge potentials

$$\mathbf{H} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Edge weights

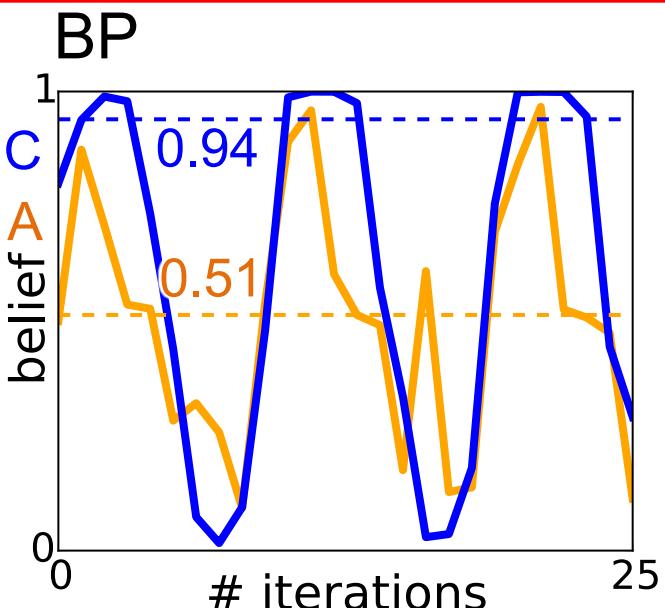
$$H_{ij}^e \propto H_{ij}^w$$

Explicit beliefs

$$\mathbf{e}_A = \mathbf{e}_B = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$$

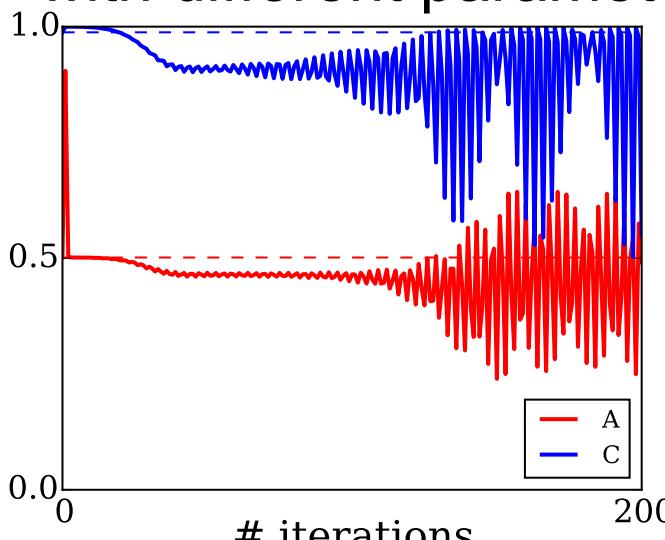
Nodes w/o priors:

$$\mathbf{e}_C = \mathbf{e}_D = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$



1) no guaranteed converge

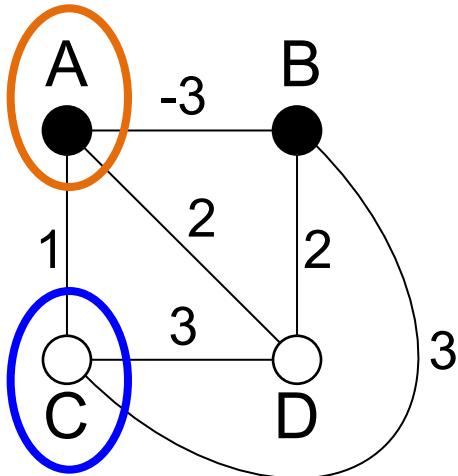
BP with different parameters



3) unstable fix points possible

LinBP can always converge

DETAILED EXAMPLE



Edge potentials

$$\mathbf{H} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Edge weights

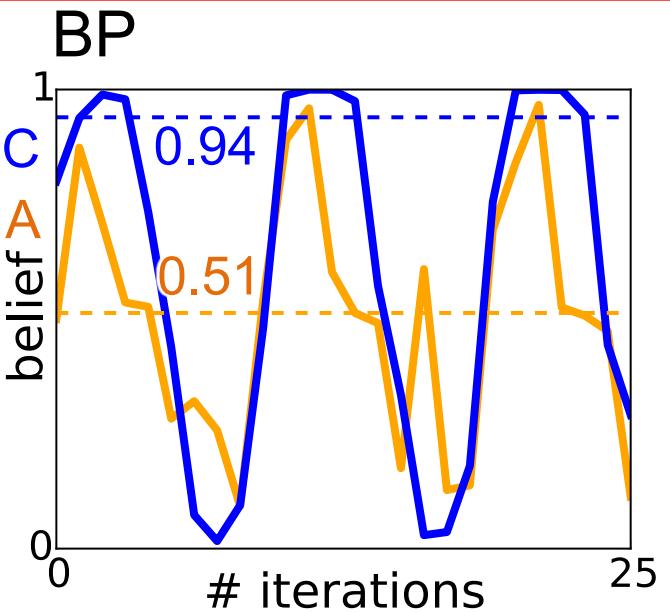
$$H_{ij}^e \propto H_{ij}^w$$

Explicit beliefs

$$\mathbf{e}_A = \mathbf{e}_B = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$$

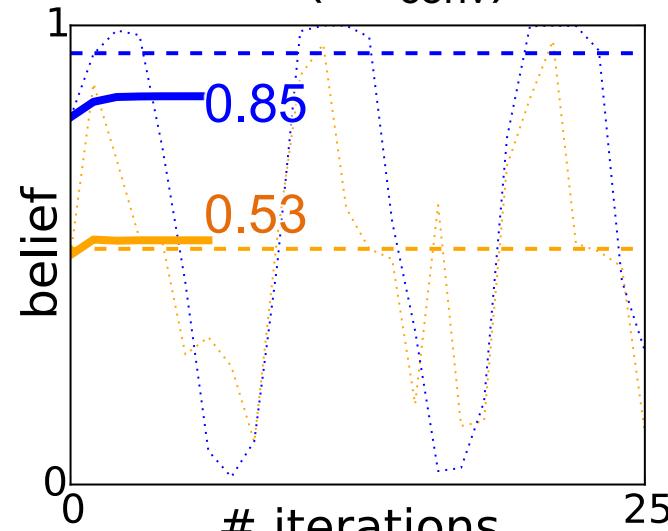
Nodes w/o priors:

$$\mathbf{e}_C = \mathbf{e}_D = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$



1) no guaranteed converge

LinBP with $(\varepsilon / \varepsilon_{\text{conv}}) = 0.1$



LinBP: guaranteed converge

LinBP in SQL

Algorithm 1: (LinBP) Returns the final beliefs F with LinBP for a weighted network W with explicit beliefs X , coupling strengths H , and calculated tables D and H_2 .

Input: $W(s, t, w), E(v, c, b), H(c_1, c_2, h), D(v, d), H_2(c_1, c_2, h)$

Output: $B(v, c, b)$

Initialize final beliefs for explicit nodes:

$B(s, c, b) := \text{Explanation}(s, c, b)$

for $i \leftarrow 1$ **to** l **do**

Create two temporary views:

$V_1(t, c_2, \text{sum}(w \cdot b \cdot h)) := W(s, t, w), B(s, c_1, b), H(c_1, c_2, h)$

$V_2(s, c_2, \text{sum}(d \cdot b \cdot h)) := D(s, d), B(s, c_1, b), H_2(c_1, c_2, h)$

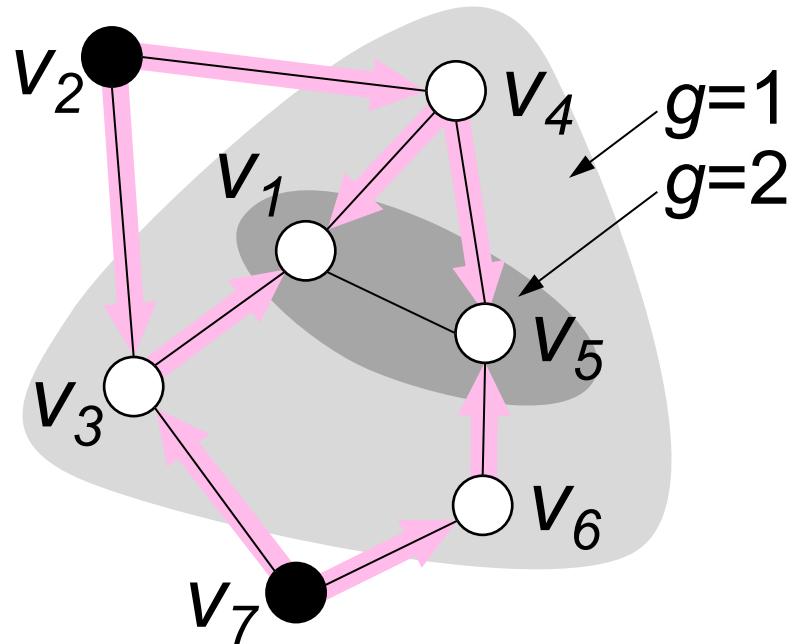
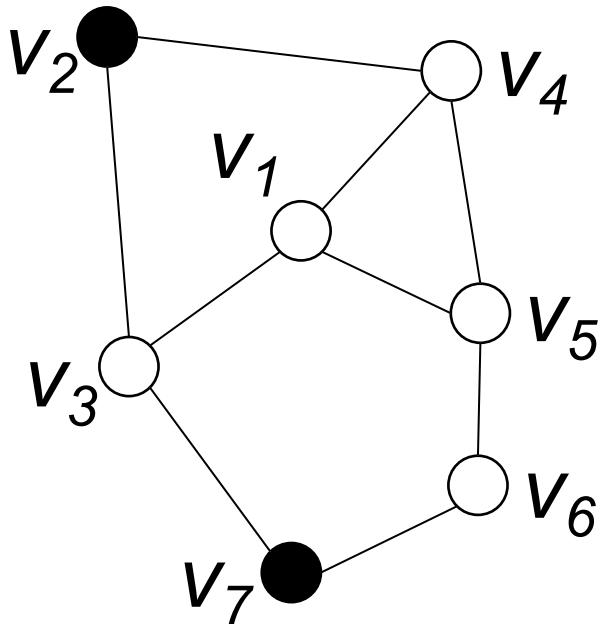
Update final beliefs:

$B(v, c, b_1 + b_2 - b_3) := E(v, c, b_1), V_1(v, c, b_2), V_2(v, c, b_3)$

return $B(v, c, b)$

```
create view V1 as
select W.t as v,
       h.c2 as c,
       sum(w*b*h) as b
  from W, F, H
 where W.s = F.v
   and F.c = H.c1
 group by W.t, H.c2;
```

Single-Pass BP: a "myopic" version of LinBP

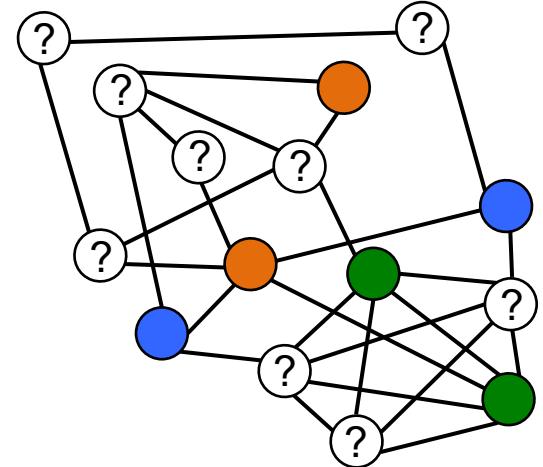


$$\hat{\mathbf{H}} \leftarrow \epsilon \cdot \hat{\mathbf{H}}_0$$

$$\hat{\mathbf{b}}_{v_1} = \epsilon^2 \cdot \hat{\mathbf{H}}_o^2 (2\hat{\mathbf{e}}_{v_2} + \hat{\mathbf{e}}_{v_7})$$

POSTER

Linearized and Single-Pass Belief Propagation



Wolfgang Gatterbauer, Stephan Günnemann¹,
Danai Koutra², Christos Faloutsos

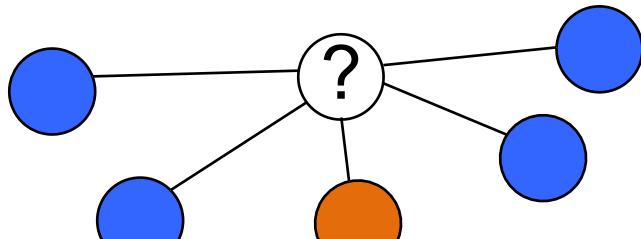
VLDB 2015



Carnegie Mellon
Database Group

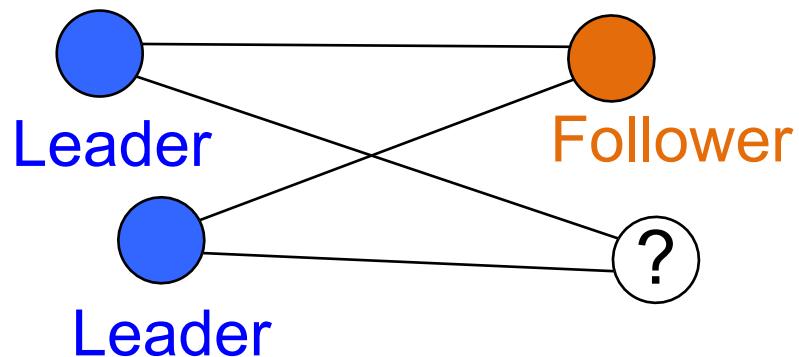
¹ Technical University of Munich
² University of Michigan

"Birds of a feather..." (Homophily)



Political orientation

"Opposites attract" (Heterophily)



"Guilt-by-association"

- recommender systems
- semi-supervised learning
- random walks (PageRank)

Disassortative networks

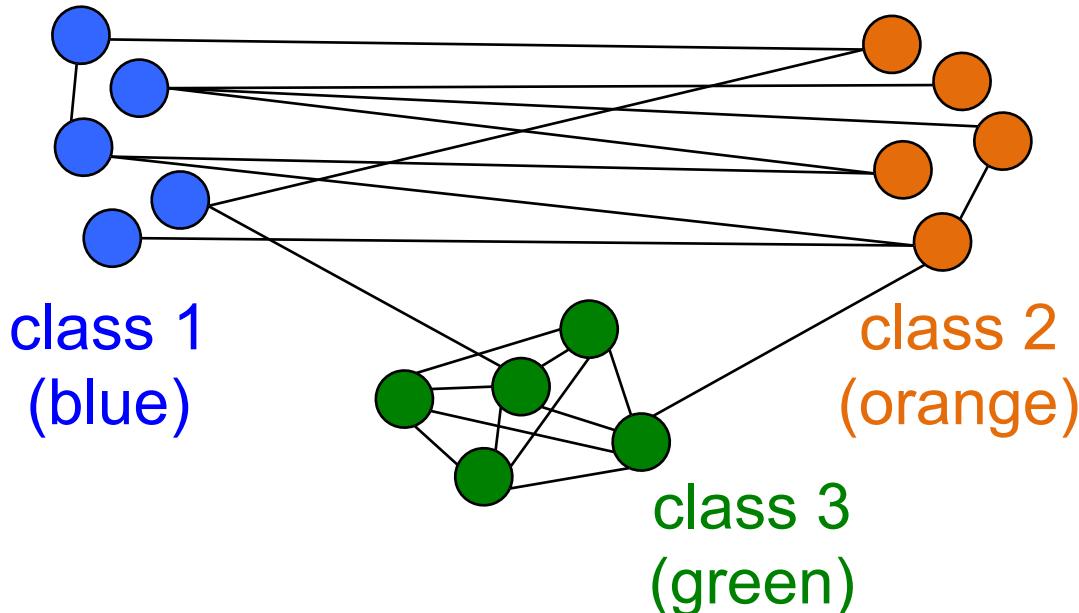
- biological food web
- protein interaction
- online fraud settings

Affinity matrix (also potential or heterophily matrix)

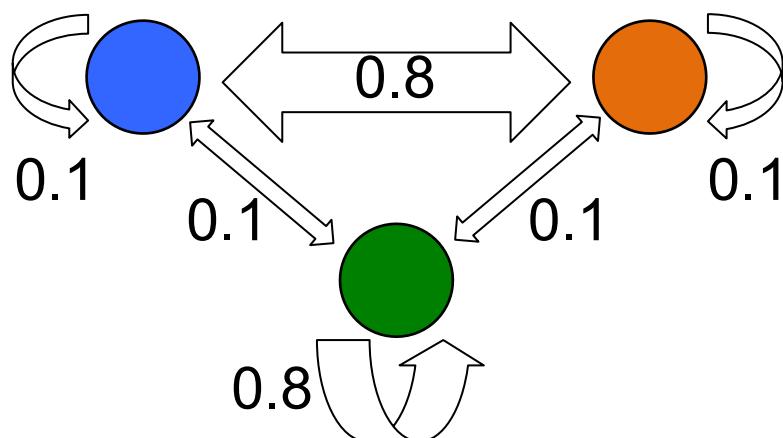
$$H = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix} \end{matrix}$$

$$H = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \end{matrix}$$

More general forms of "Heterophily"



Class-to-class interaction



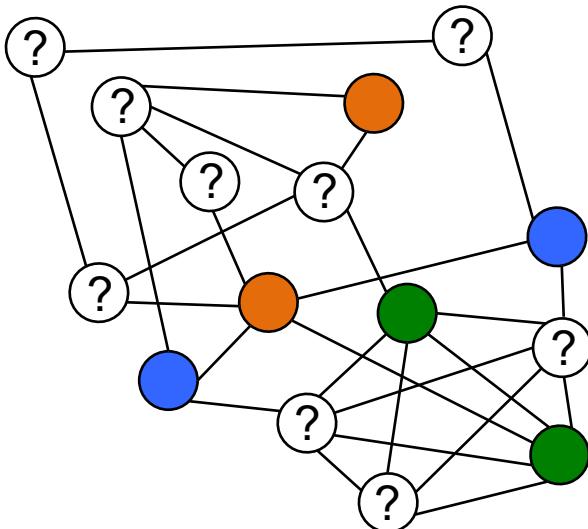
Potential (P)

	1	2	3
1	1	8	1
2	8	1	1
3	1	1	8

Affinity matrix (H)

	1	2	3
1	0.1	0.8	0.1
2	0.8	0.1	0.1
3	0.1	0.1	0.8

The problem we are trying to solve



$$\mathbf{H} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 0.1 & 0.8 & 0.1 \\ 0.8 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{matrix} \end{matrix}$$

k=3 classes

Problem formulation

- Given:
- undirected graph G
 - seed labels
 - affinity matrix H
(symmetric, doubly stoch.)
- Find:
- remaining labels

Good: Graphical models can model this problem

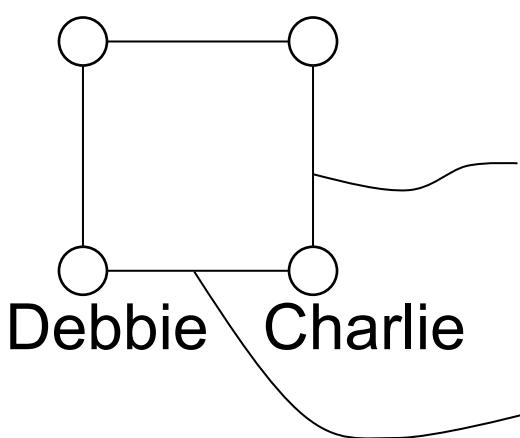


Bad: Belief Propagation on graphs with loops has convergence issues



Markov networks (also Markov Random Fields)

Alice Bob



"Misconception example" [Koller,Friedman\[2009\]](#)
Does a student think + or -?

$$H_{BC} = \begin{array}{c|cc} & \text{B}\backslash\text{C} & + & - \\ \hline + & 0.9 & 0.1 \\ - & 0.1 & 0.9 \end{array}$$

Bob and Charlie tend to agree (homophily)

$$H_{CD} = \begin{array}{c|cc} & \text{C}\backslash\text{D} & + & - \\ \hline + & 0.1 & 0.9 \\ - & 0.9 & 0.1 \end{array}$$

Charlie and Debbie tend to disagree (heterophily)

Inference tasks

- Marginals (belief distribution)
- Maximum Marginals (classification)

$$b_D = \begin{array}{c|c} & + & 0.6 \\ \hline - & 0.4 \end{array}$$

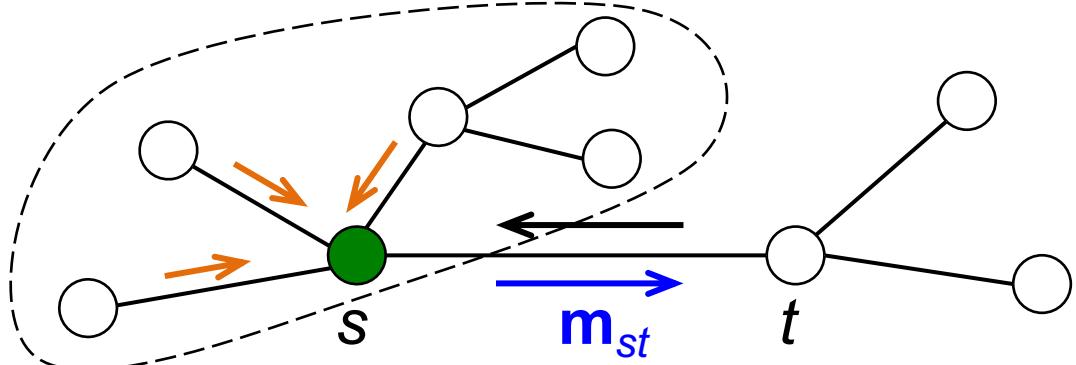
Maximum Marginal

But Inference is typically hard



Belief Propagation

DETAILS



- 1) Initialize all message entries to 1
- 2) Iteratively: calculate messages for each edge and class

$$m_{st} \propto H(e_s \odot \bigodot_{u \in N(s) \setminus t} m_{us})$$

edge potential
explicit (prior) beliefs componentwise-multiplication operator
EC

multiply messages from
all neighbors except t
("echo cancellation" EC)

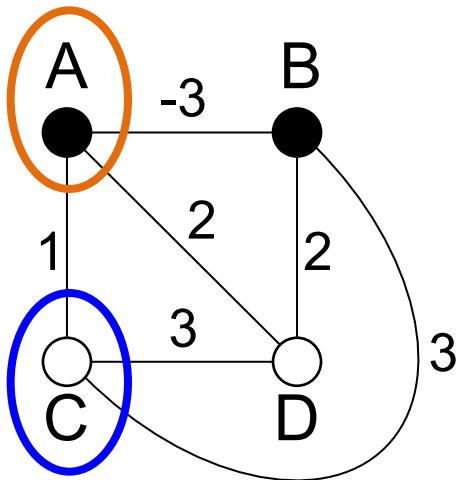
- 3) After messages converge: calculate final beliefs

$$b_s \propto e_s \odot \bigodot_{u \in N(s)} m_{us}$$

Does often not converge
on graphs with loops



Illustration of convergence issue with BP



Edge potentials

$$\mathbf{H} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Edge weights

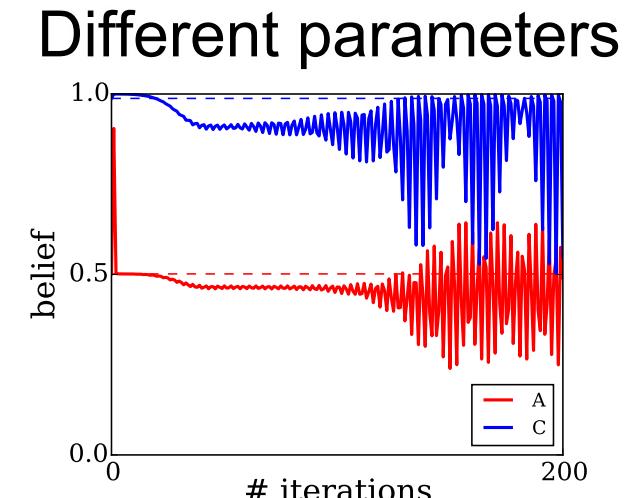
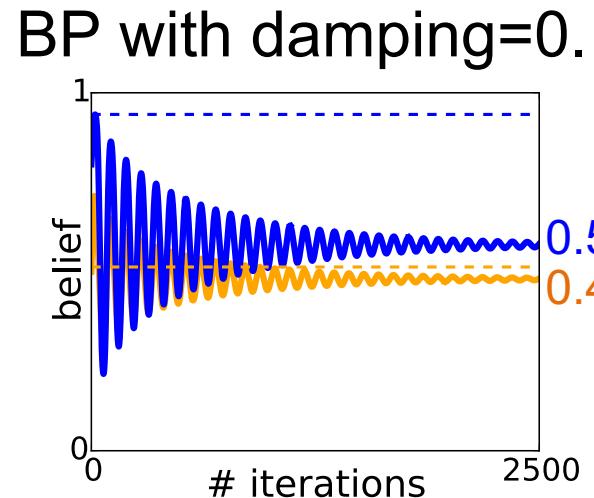
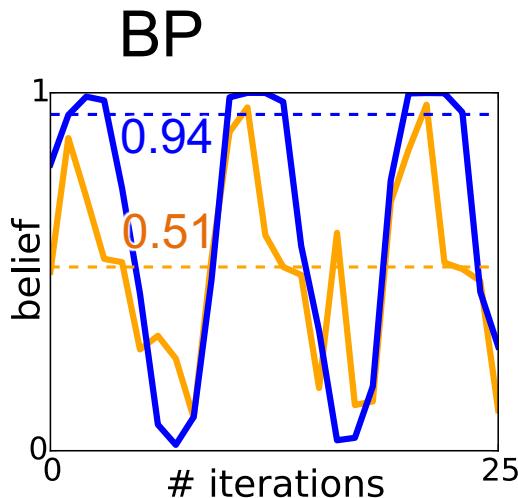
$$H_{ij} \propto H_{ij}^w$$

Explicit beliefs

$$\mathbf{e}_A = \mathbf{e}_B = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}$$

Nodes w/o priors:

$$\mathbf{e}_C = \mathbf{e}_D = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$



1) no guaranteed convergence

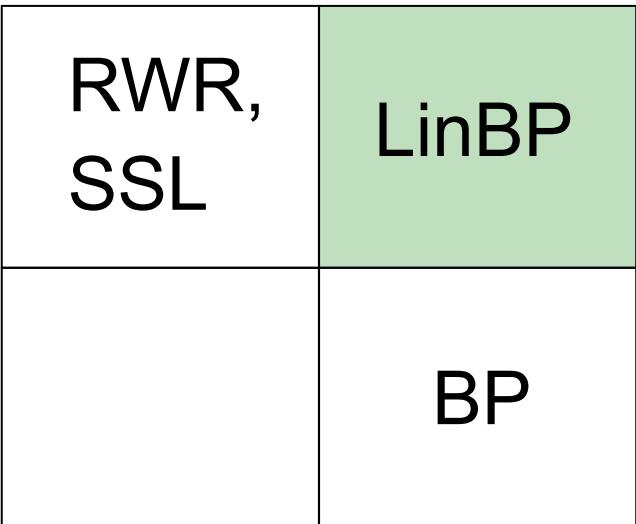
2) damping: many iterations

3) BP can have unstable fix points

Properties of "Linearized Belief Propagation"

Linear
Algebra

Iterative



homoph. heteroph.

Expressiveness

Improved
computational
properties

LinBP properties:

- Heterophily
- Matrix Algebra
- Closed form
- convergence criteria
- SQL with recursion

Key Idea: Centering + Linearizing BP

Original Value = Center point + Residual

$$\begin{array}{l}
 \mathbf{H} = \left[\frac{1}{k} \right]_{k \times k} + \hat{\mathbf{H}} \\
 \mathbf{e}, \mathbf{b} = \frac{1}{k} \mathbf{1} + \hat{\mathbf{e}}, \hat{\mathbf{b}} \\
 \mathbf{m} = \mathbf{1} + \hat{\mathbf{m}}
 \end{array}$$

\mathbf{H}	$\left[\frac{1}{k} \right]_{k \times k}$	$\hat{\mathbf{H}}$
\mathbf{e}, \mathbf{b}	$\frac{1}{k} \mathbf{1}$	$\hat{\mathbf{e}}, \hat{\mathbf{b}}$
\mathbf{m}	$\mathbf{1}$	$\hat{\mathbf{m}}$

	Formula	Maclaurin series	Approximation
Logarithm	$\ln(1 + \epsilon) = \epsilon - \frac{\epsilon^2}{2} + \frac{\epsilon^3}{3} - \dots$		$\approx \epsilon$
Division	$\frac{\frac{1}{k} + \epsilon_1}{1 + \epsilon_2} = \left(\frac{1}{k} + \epsilon_1 \right) \left(1 - \epsilon_2 + \epsilon_2^2 - \dots \right)$		$\approx \frac{1}{k} + \epsilon_1 - \frac{\epsilon_2}{k}$

Linearized Belief Propagation

DETAILS

$$\mathbf{m}_{st} \propto H\left(\mathbf{e}_s \odot \bigodot_{u \in N(s) \setminus t} \mathbf{m}_{us}\right)$$

BP

not linear



$$\mathbf{b}_s \propto \mathbf{e}_s \odot \bigodot_{u \in N(s)} \mathbf{m}_{us}$$

$$\hat{\mathbf{m}}_{st} \leftarrow \hat{H}\left(\hat{\mathbf{e}}_s + \frac{1}{k} \cdot \sum_{u \in N(s) \setminus t} \hat{\mathbf{m}}_{us}\right)$$

LinBP

$$\hat{\mathbf{b}}_s \leftarrow \hat{\mathbf{e}}_s + \frac{1}{k} \cdot \sum_{u \in N(s)} \hat{\mathbf{m}}_{us}$$

linear



Messages remain centered !

Matrix formulation of LinBP

DETAILS

Update equation

$$\hat{\mathbf{B}} \leftarrow \hat{\mathbf{E}} + \mathbf{W} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{H}} - \underbrace{\mathbf{D} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{H}}^2}_{\text{EC}}$$

final beliefs (n x k) explicit beliefs graph affinity matrix degree matrix EC

t t t
+
graph S
-
affinity matrix degree matrix EC

Closed form

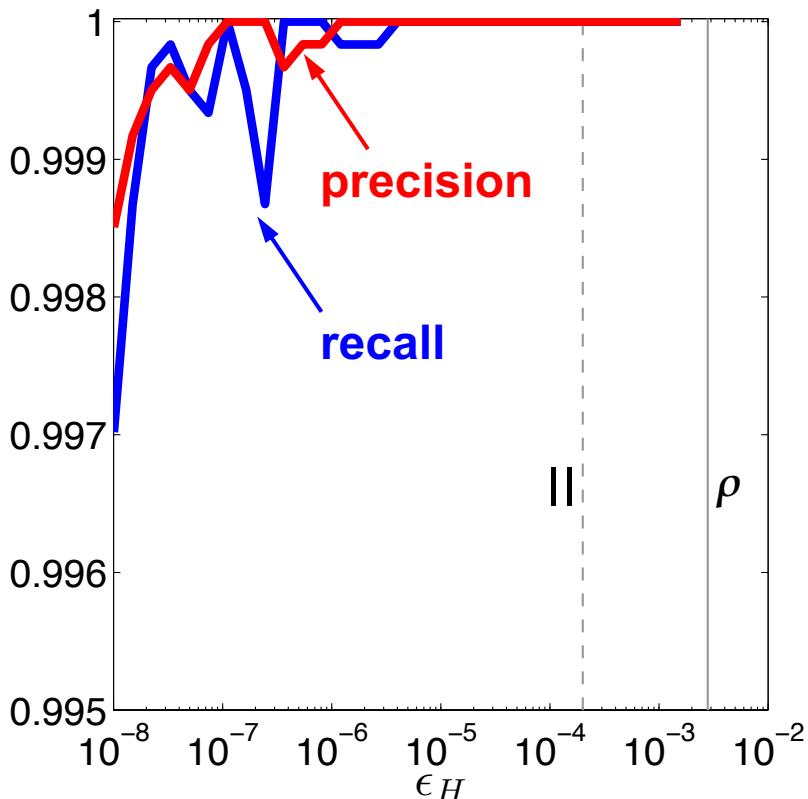
$$\text{vec}(\hat{\mathbf{B}}) = \underbrace{(\mathbf{I} - \hat{\mathbf{H}} \otimes \mathbf{W} + \hat{\mathbf{H}}^2 \otimes \mathbf{D})^{-1}}_{\text{EC}} \text{vec}(\hat{\mathbf{E}})$$

Convergence Spectral radius of (...) < 1

Scale with appropriate ϵ : $\hat{\mathbf{H}} \leftarrow \epsilon \cdot \hat{\mathbf{H}}_0$

Experiments

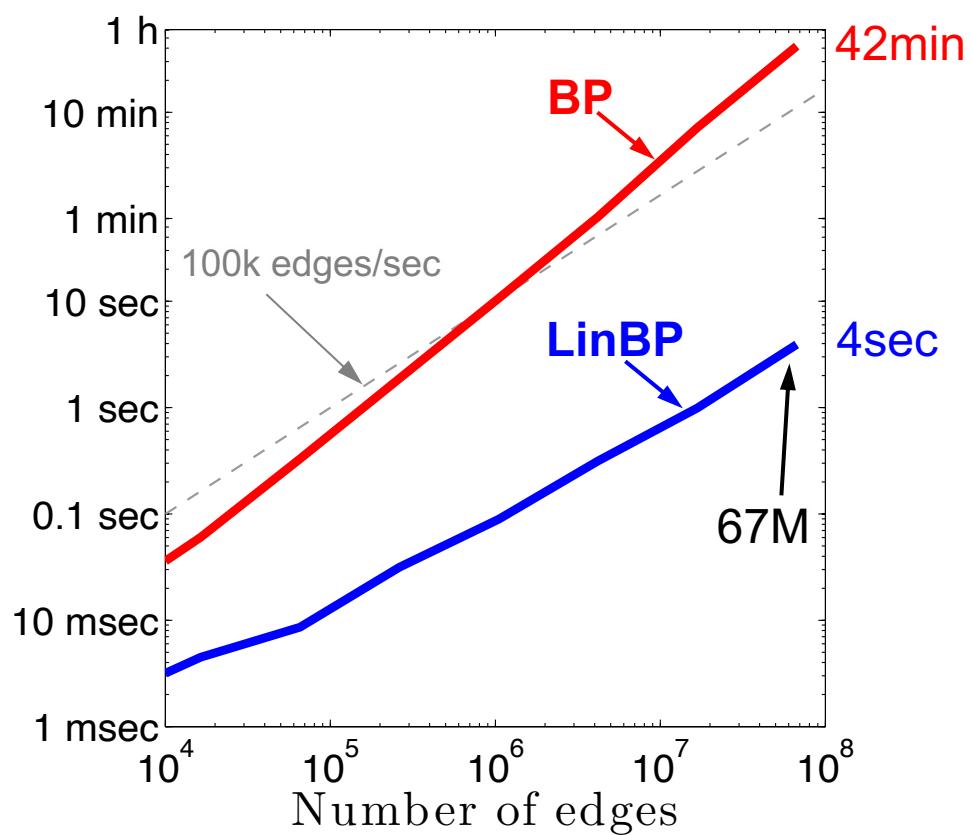
1) Great labeling accuracy (LinBP against BP as ground truth)



LinBP & BP give similar labels

Reason: when BP converges, then our approximations are reasonable

2) Linear scalability (10 iterations)



LinBP is seriously faster!

Reason: LinBP can leverage existing optimized matrix multiplication

LinBP in SQL

Algorithm 1: (LinBP) Returns the final beliefs F with LinBP for a weighted network W with explicit beliefs X , coupling strengths H , and calculated tables D and H_2 .

Input: $W(s, t, w), E(v, c, b), H(c_1, c_2, h), D(v, d), H_2(c_1, c_2, h)$

Output: $B(v, c, b)$

Initialize final beliefs for explicit nodes:

$B(s, c, b) := \text{Explanation}(s, c, b)$

for $i \leftarrow 1$ **to** l **do**

Create two temporary views:

$V_1(t, c_2, \text{sum}(w \cdot b \cdot h)) := W(s, t, w), B(s, c_1, b), H(c_1, c_2, h)$

$V_2(s, c_2, \text{sum}(d \cdot b \cdot h)) := D(s, d), B(s, c_1, b), H_2(c_1, c_2, h)$

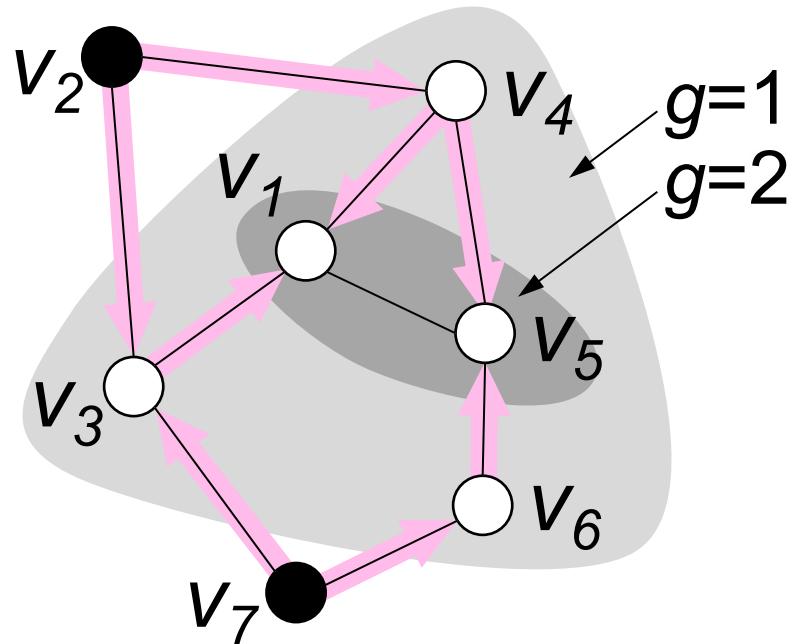
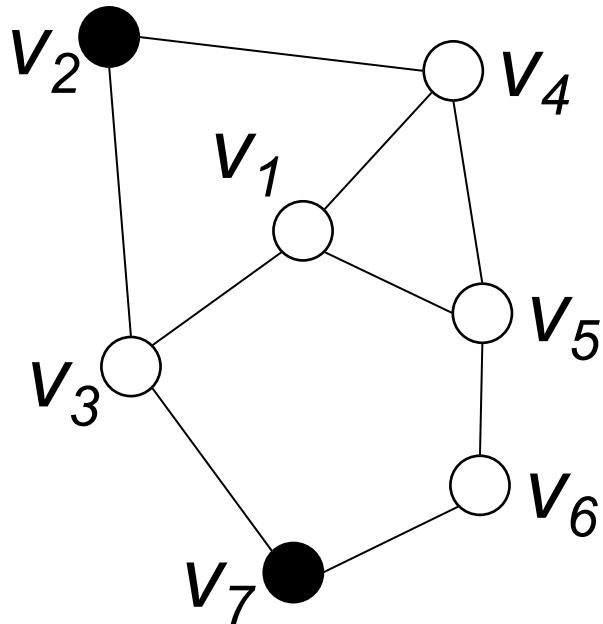
Update final beliefs:

$B(v, c, b_1 + b_2 - b_3) := E(v, c, b_1), V_1(v, c, b_2), V_2(v, c, b_3)$

return $B(v, c, b)$

```
create view V1 as
select W.t as v,
       h.c2 as c,
       sum(w*b*h) as b
  from W, F, H
 where W.s = F.v
   and F.c = H.c1
 group by W.t, H.c2;
```

Single-Pass BP: a "myopic" version of LinBP



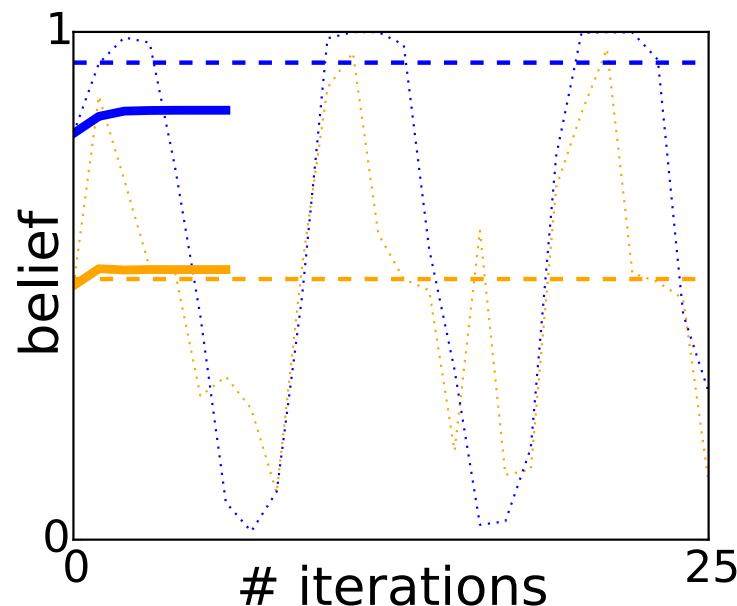
$$\hat{\mathbf{H}} \leftarrow \epsilon \cdot \hat{\mathbf{H}}_0$$

$$\hat{\mathbf{b}}_{v_1} = \epsilon^2 \cdot \hat{\mathbf{H}}_o^2 (2\hat{\mathbf{e}}_{v_2} + \hat{\mathbf{e}}_{v_7})$$

Take-aways

Goal: propagate multi-class heterophily from labeled data

Problem: How to solve the convergence issues of BP 😞



Solution: Let's linearize BP to make it converge & speed it up ☺

- Linearized Belief Propagation (LinBP)
 - Matrix Algebra, convergence, closed-form
 - SQL (w/ recursion) with standard aggregates
- Single-pass Belief Propagation (SBP)
 - Myopic version, incremental updates

Come to our talk on Wed, 10:30 (Queens 4)

