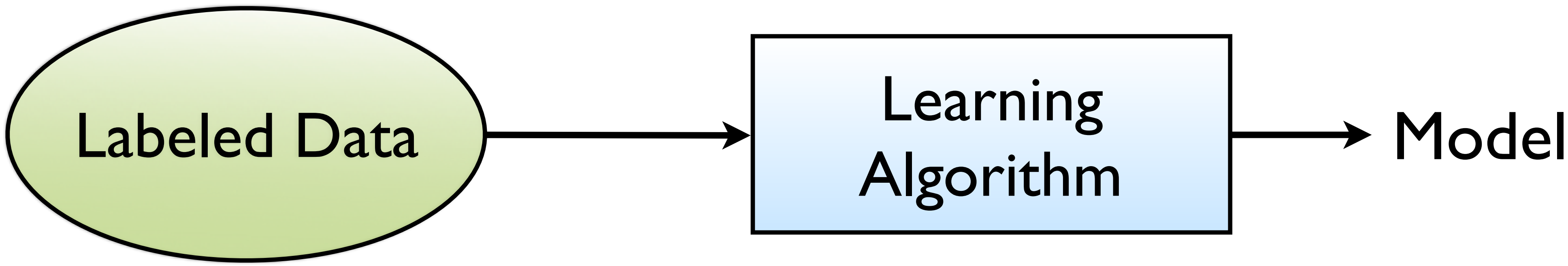
Scalable Learning & Inference Over Graphs

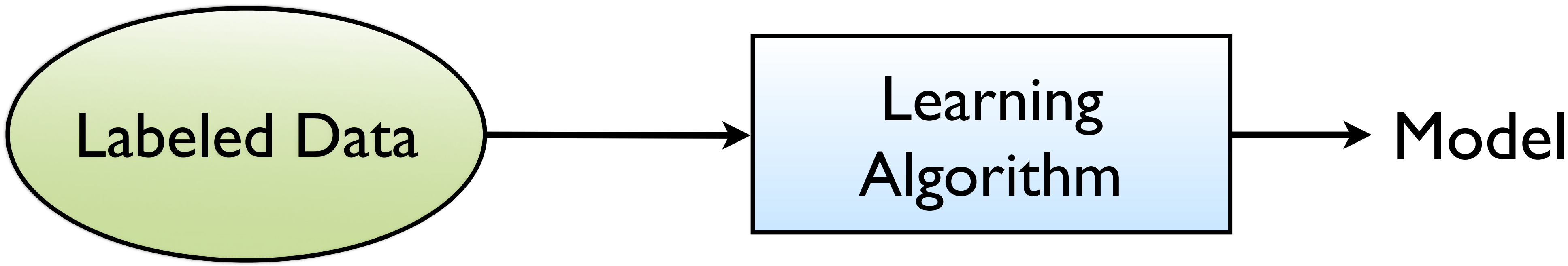
Partha Pratim Talukdar
Indian Institute of Science
ppt@cds.iisc.ac.in

February 17, 2015

Supervised Learning



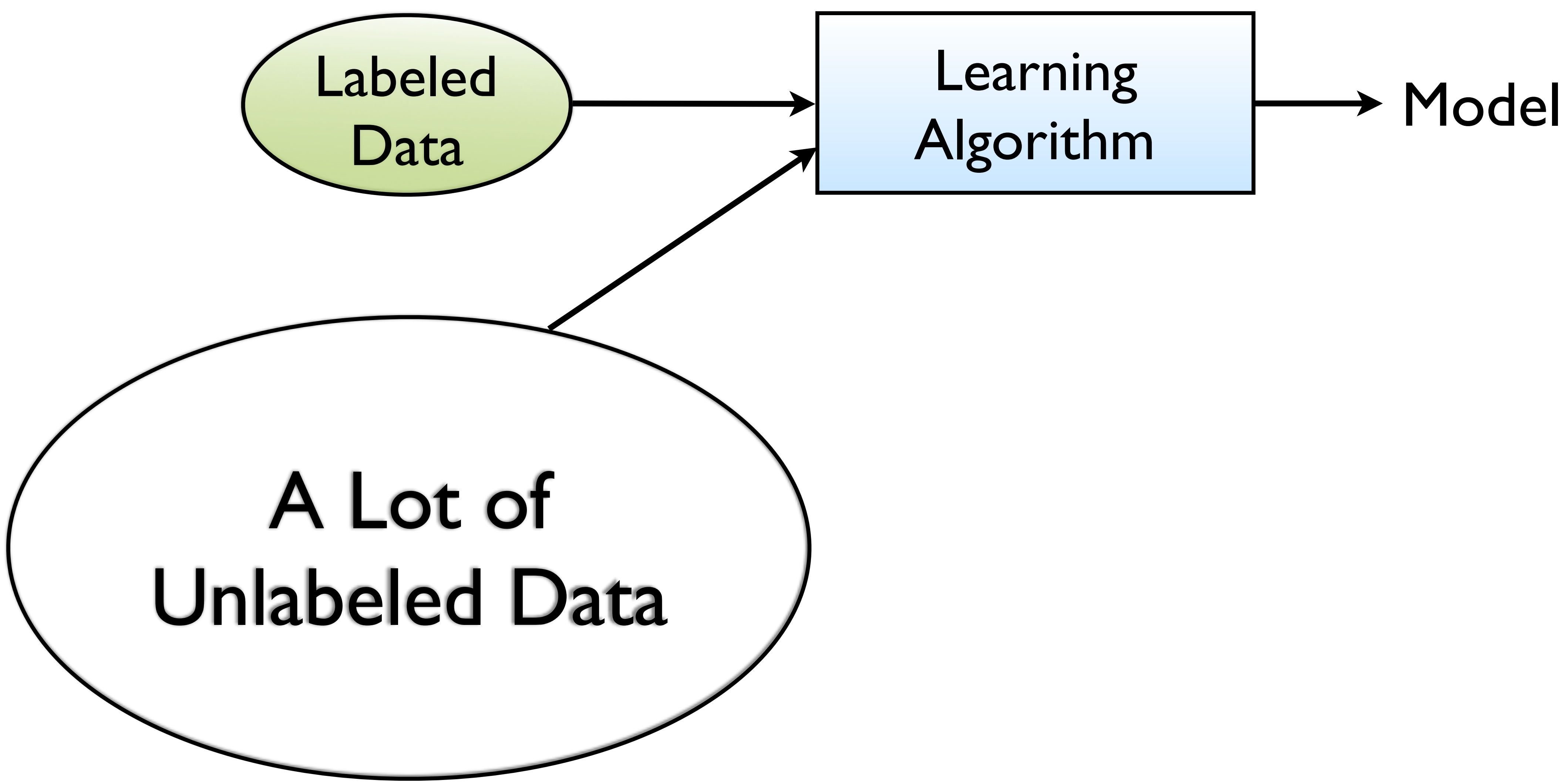
Supervised Learning



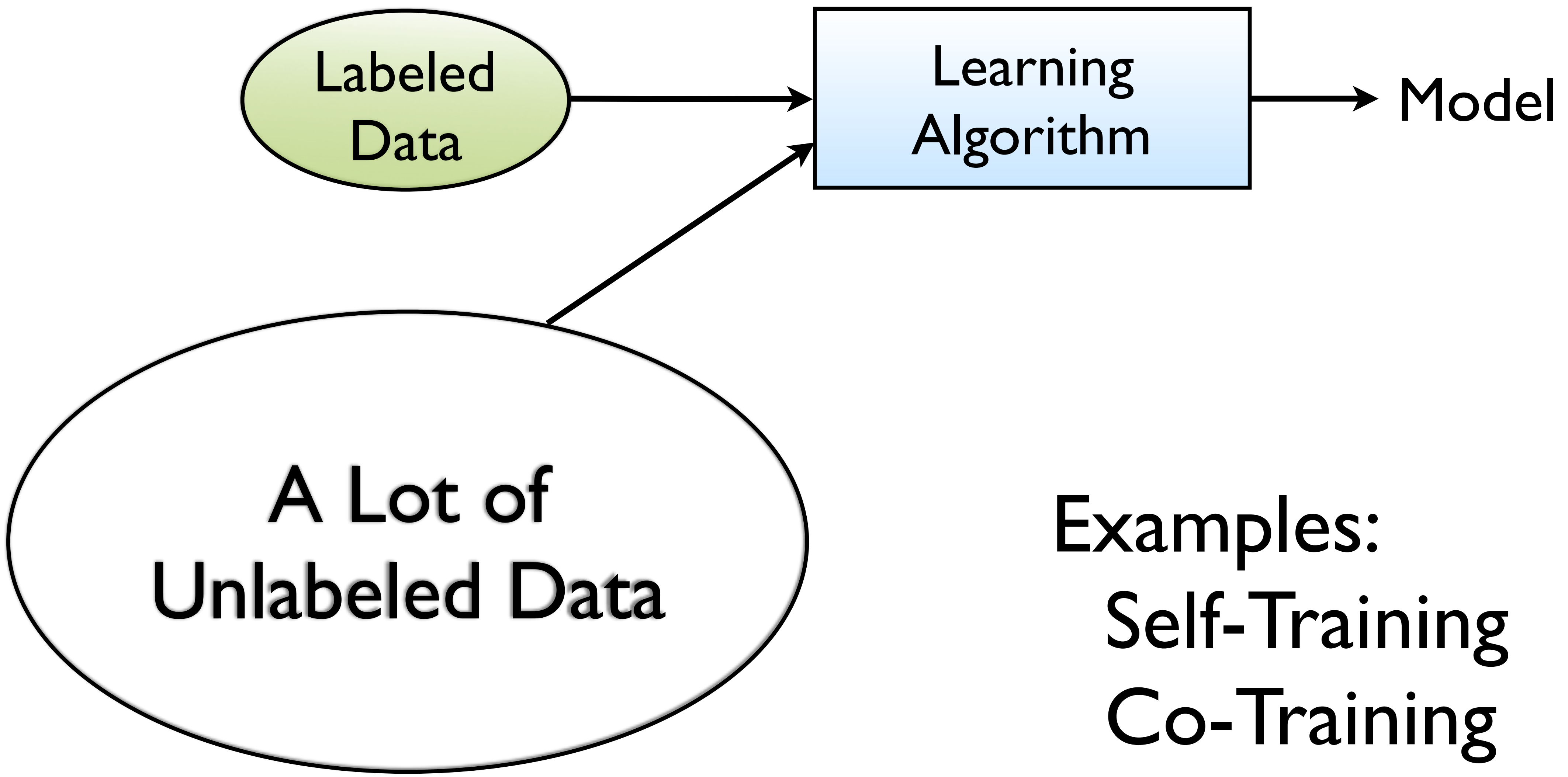
Examples:

- Decision Trees
- Support Vector Machine (SVM)
- Maximum Entropy (MaxEnt)

Semi-Supervised Learning (SSL)



Semi-Supervised Learning (SSL)



Examples:

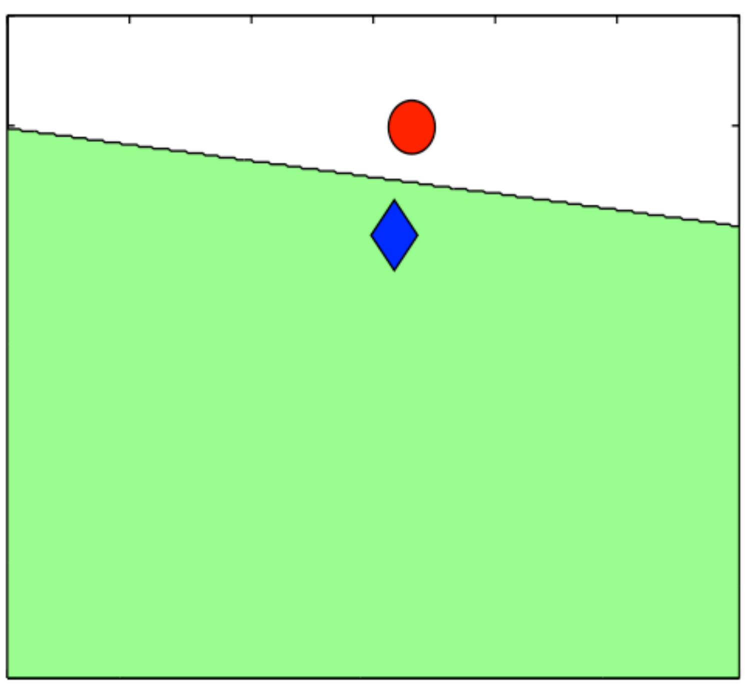
- Self-Training
- Co-Training

Why SSL?

How can unlabeled data be helpful?

Why SSL?

How can unlabeled data be helpful?

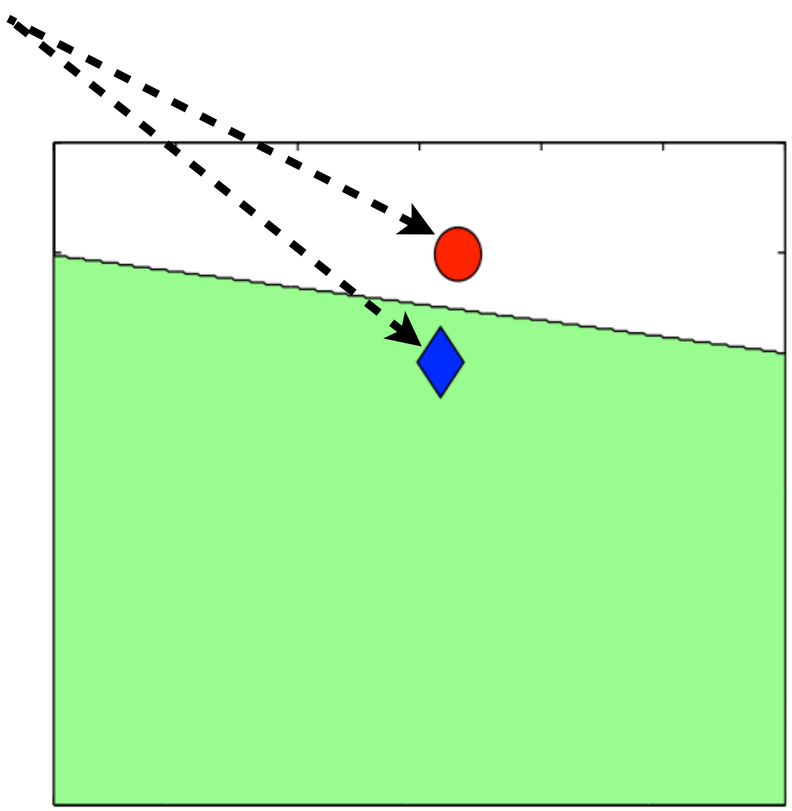


Without Unlabeled Data

Why SSL?

How can unlabeled data be helpful?

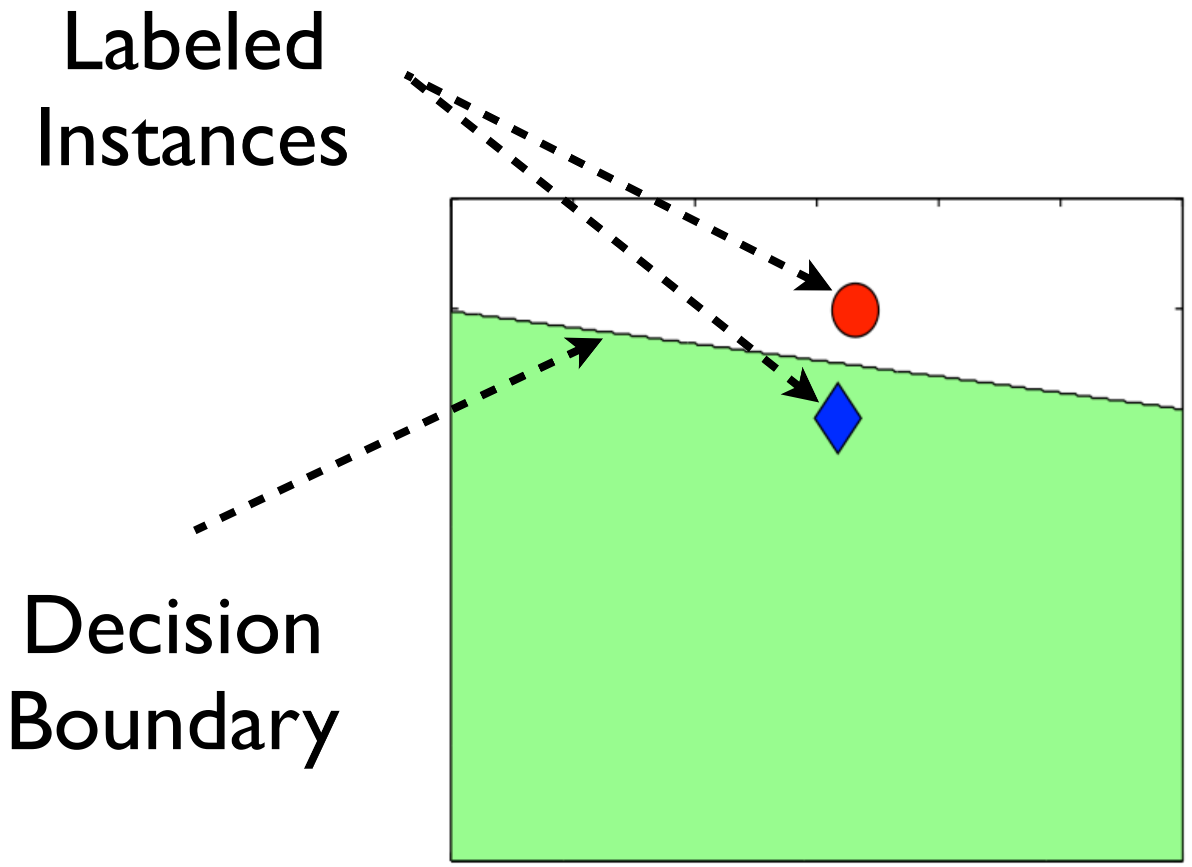
Labeled
Instances



Without Unlabeled Data

Why SSL?

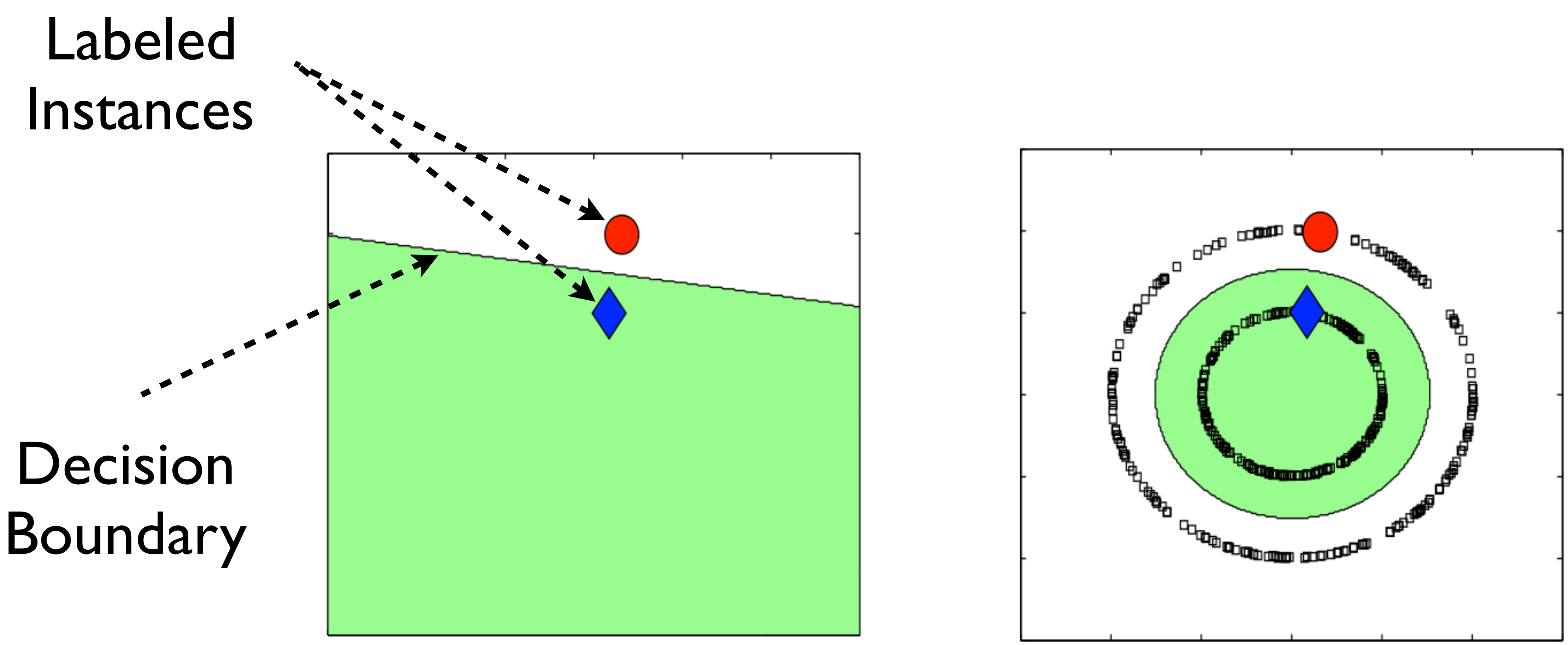
How can unlabeled data be helpful?



Without Unlabeled Data

Why SSL?

How can unlabeled data be helpful?

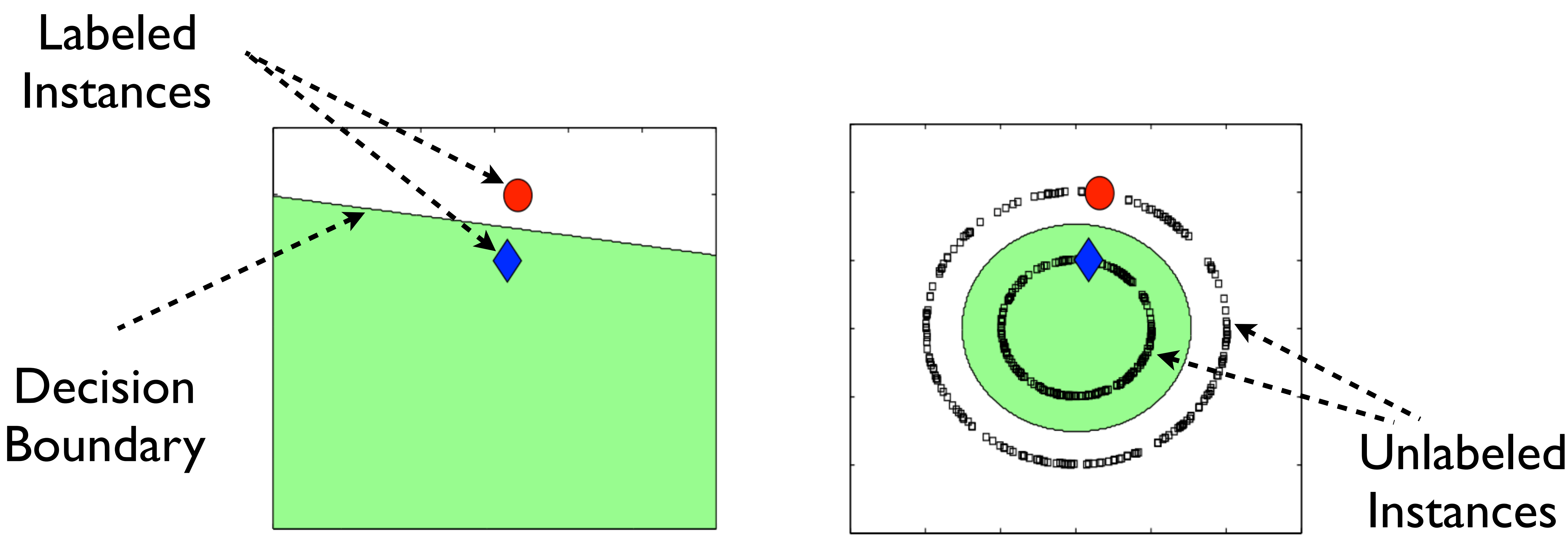


Without Unlabeled Data

With Unlabeled Data

Why SSL?

How can unlabeled data be helpful?

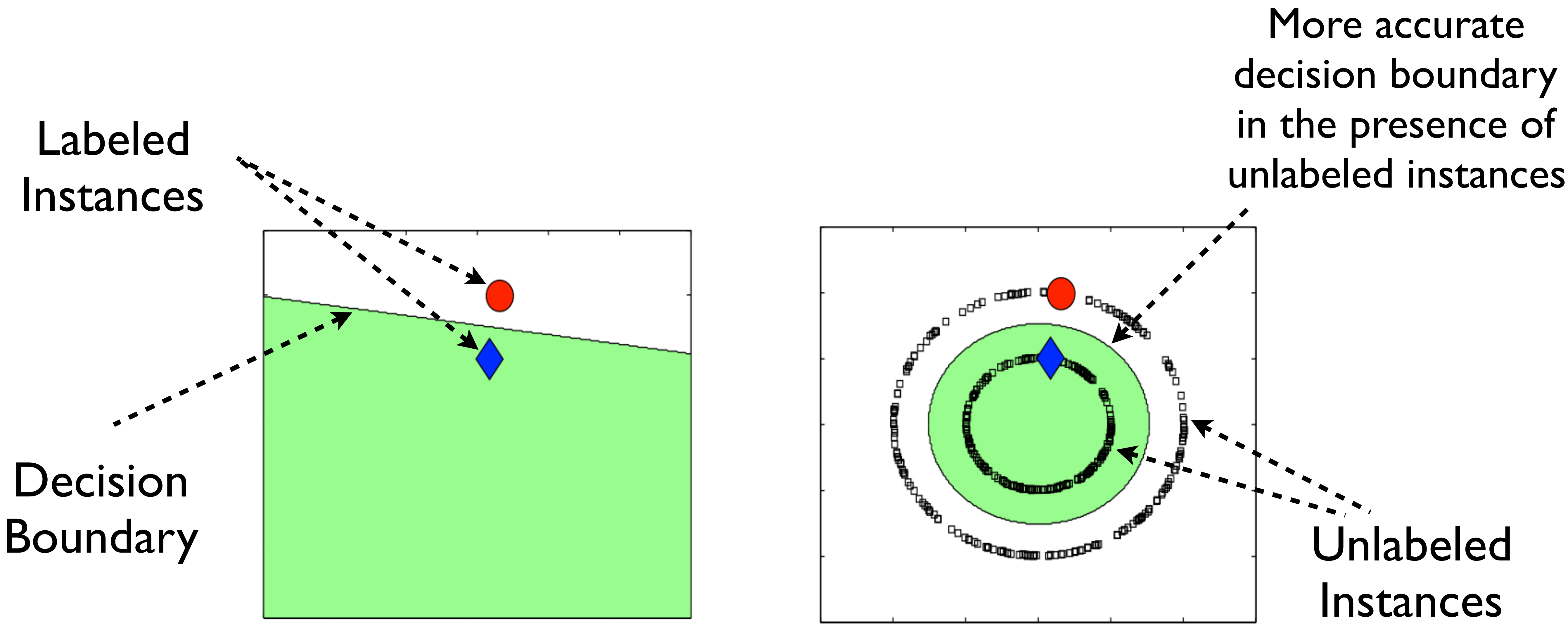


Without Unlabeled Data

With Unlabeled Data

Why SSL?

How can unlabeled data be helpful?



Without Unlabeled Data

With Unlabeled Data

Example from [Belkin et al., JMLR 2006]

Inductive vs Transductive

Inductive vs Transductive

Supervised
(Labeled)

Semi-supervised
(Labeled + Unlabeled)

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)		
Semi-supervised (Labeled + Unlabeled)		

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	
Semi-supervised (Labeled + Unlabeled)		

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	X
Semi-supervised (Labeled + Unlabeled)		

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	X
Semi-supervised (Labeled + Unlabeled)	Manifold Regularization	

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	X
Semi-supervised (Labeled + Unlabeled)	Manifold Regularization	Label Propagation (LP), MAD, MP, ...

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	X
Semi-supervised (Labeled + Unlabeled)	Manifold Regularization	Label Propagation (LP), MAD, MP, ...

Most Graph SSL algorithms are non-parametric

(i.e., # parameters grows with data size)

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	X
Semi-supervised (Labeled + Unlabeled)	Manifold Regularization	Label Propagation (LP), MAD, MP, ...

Focus of this talk

Most Graph SSL algorithms are non-parametric

(i.e., # parameters grows with data size)

Inductive vs Transductive

	Inductive (Generalize to Unseen Data)	Transductive (Doesn't Generalize to Unseen Data)
Supervised (Labeled)	SVM, Maximum Entropy	X
Semi-supervised (Labeled + Unlabeled)	Manifold Regularization	Label Propagation (LP), MAD, MP, ...

Focus of this talk

Most Graph SSL algorithms are non-parametric

(i.e., # parameters grows with data size)

See Chapter 25 of SSL Book: http://olivier.chapelle.cc/ssl-book/discussion.pdf

Two Popular SSL Algorithms

- Self Training

Two Popular SSL Algorithms

- Self Training
- Co-Training

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

Randomly choose $2p + 2n$ examples from U to replenish U'

Why Graph-based SSL?

Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...

Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data

Why Graph-based SSL?

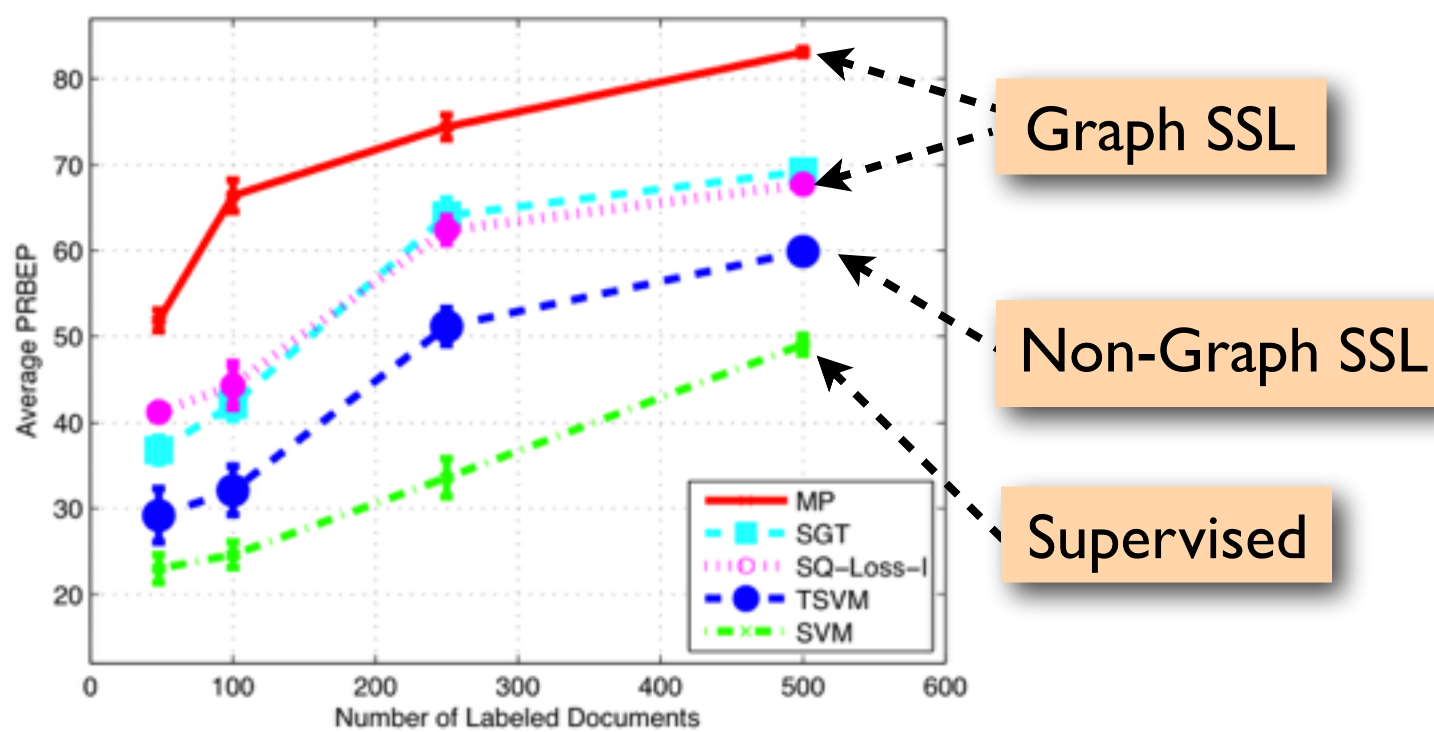
- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Easily parallelizable, scalable to large data

Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Easily parallelizable, scalable to large data
- Effective in practice

Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Easily parallelizable, scalable to large data
- Effective in practice

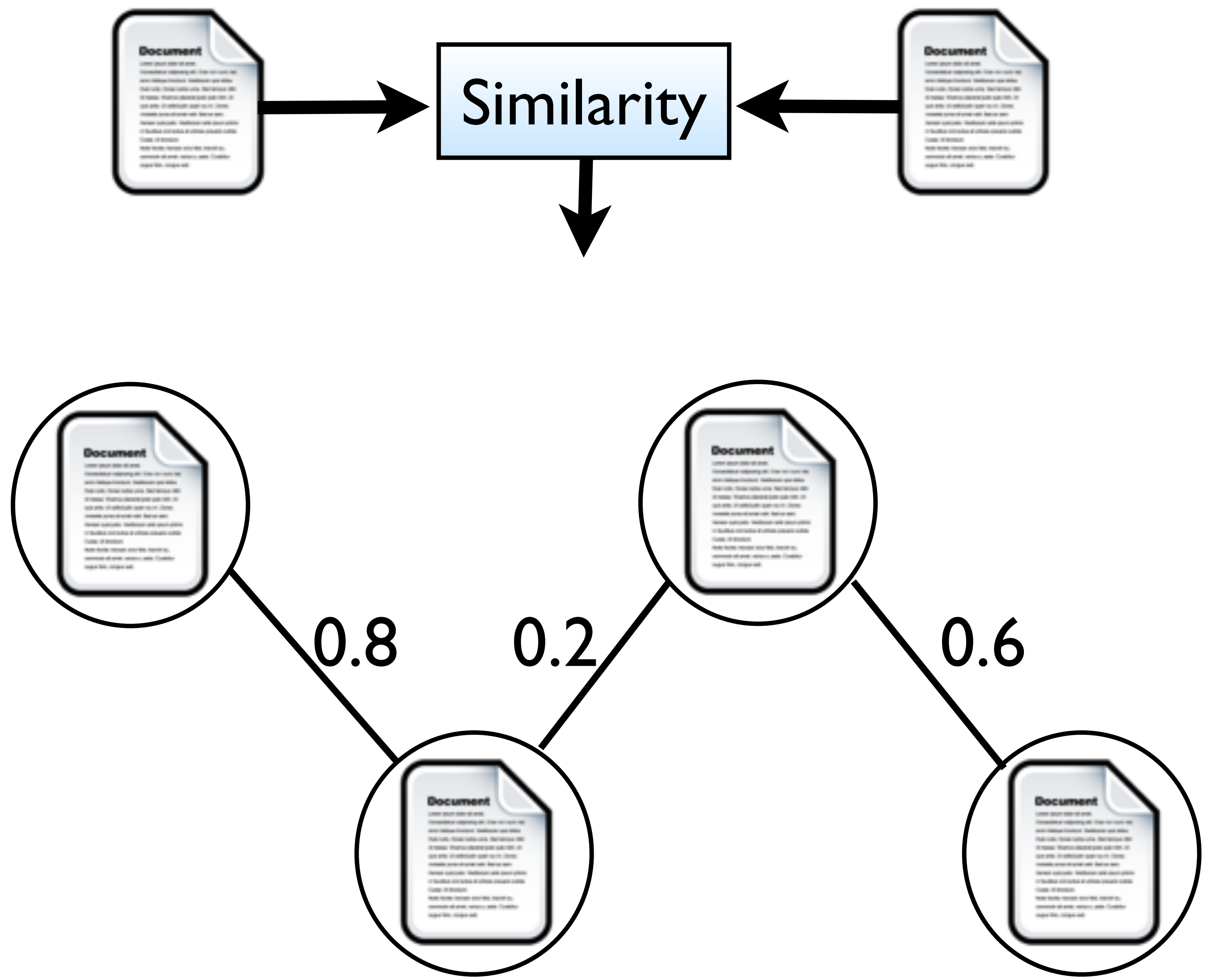


Text Classification

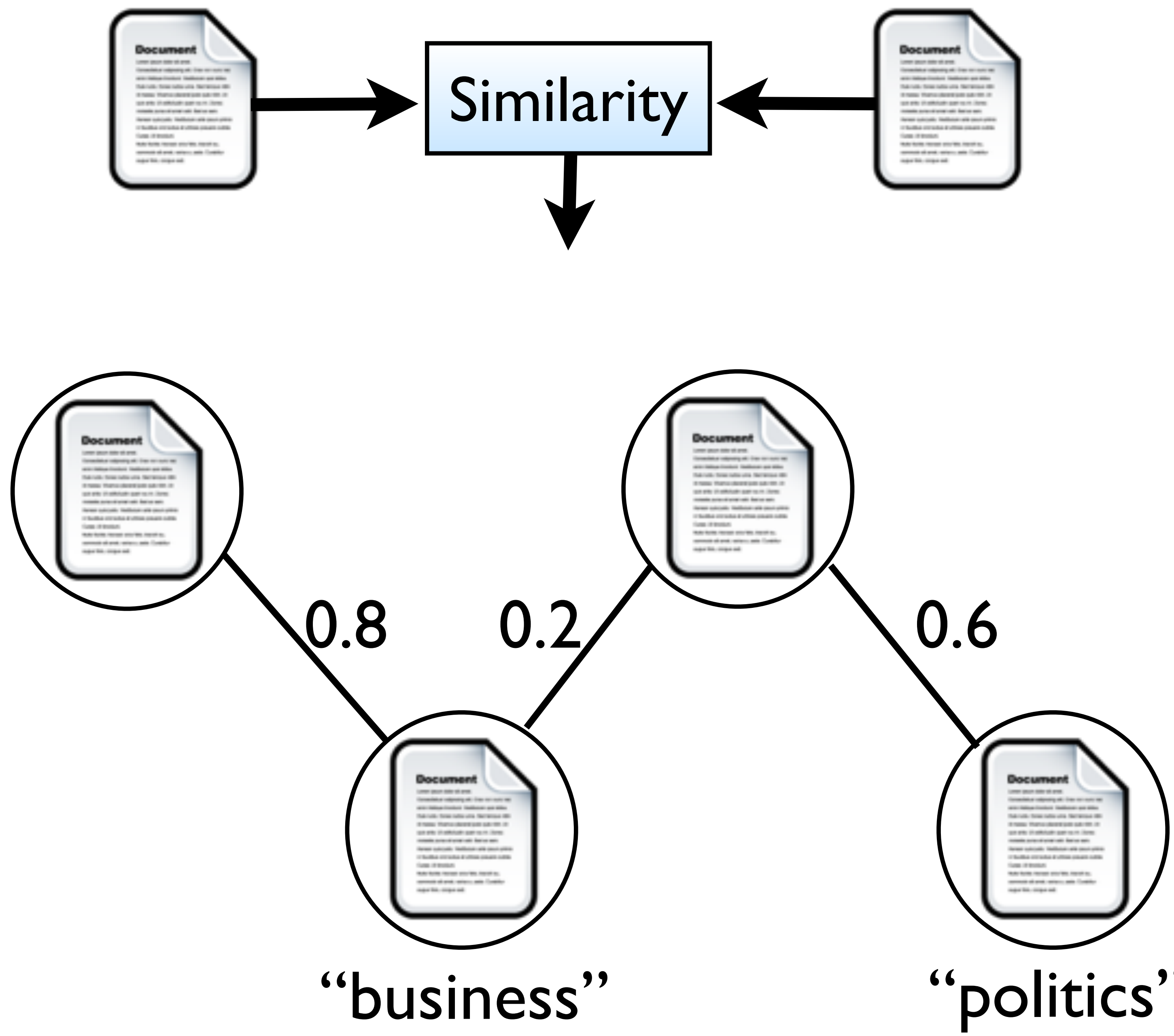
Graph-based SSL

Graph-based SSL

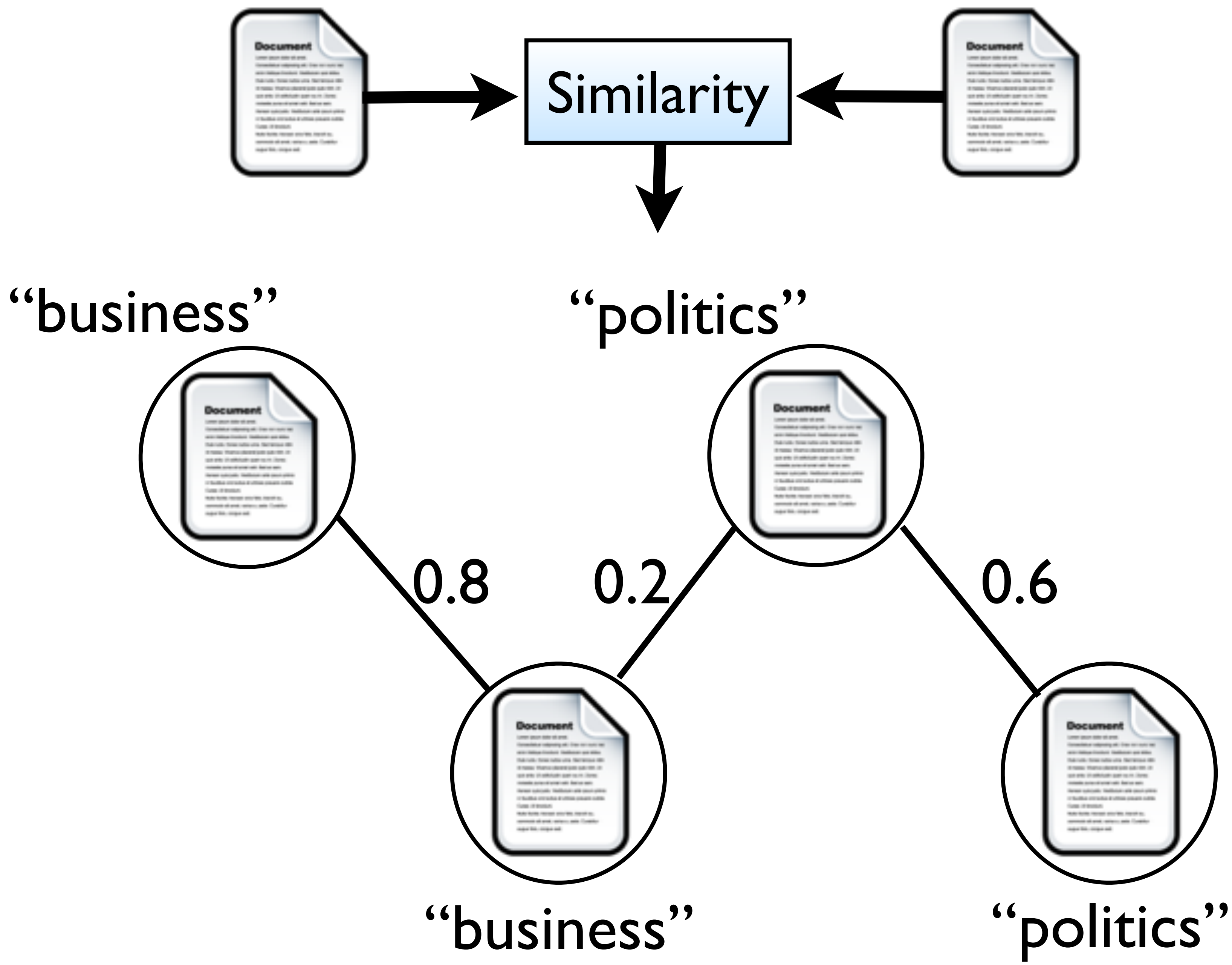
Graph-based SSL



Graph-based SSL



Graph-based SSL



Graph-based SSL

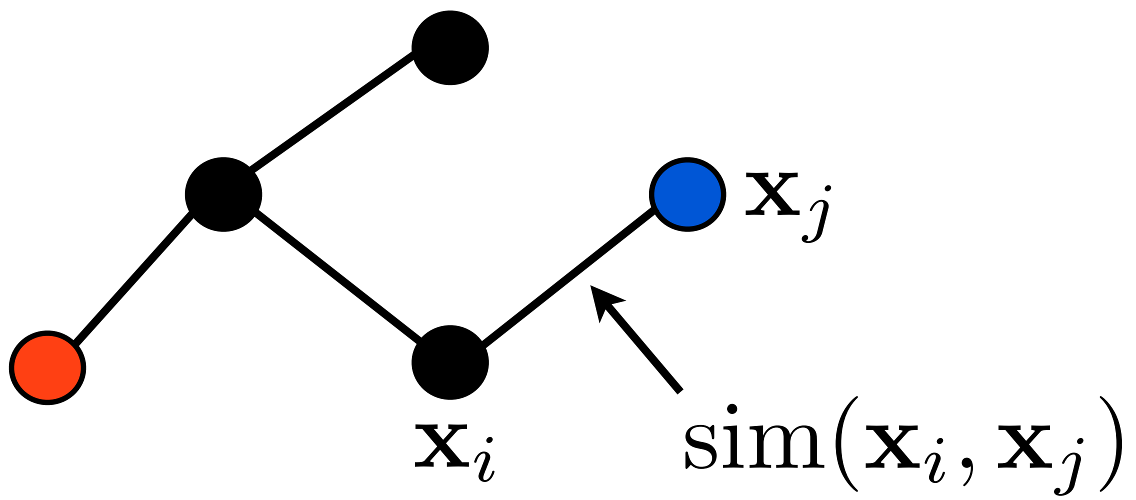
Graph-based SSL

Smoothness Assumption

If two instances are similar according to the graph, then output labels should be similar

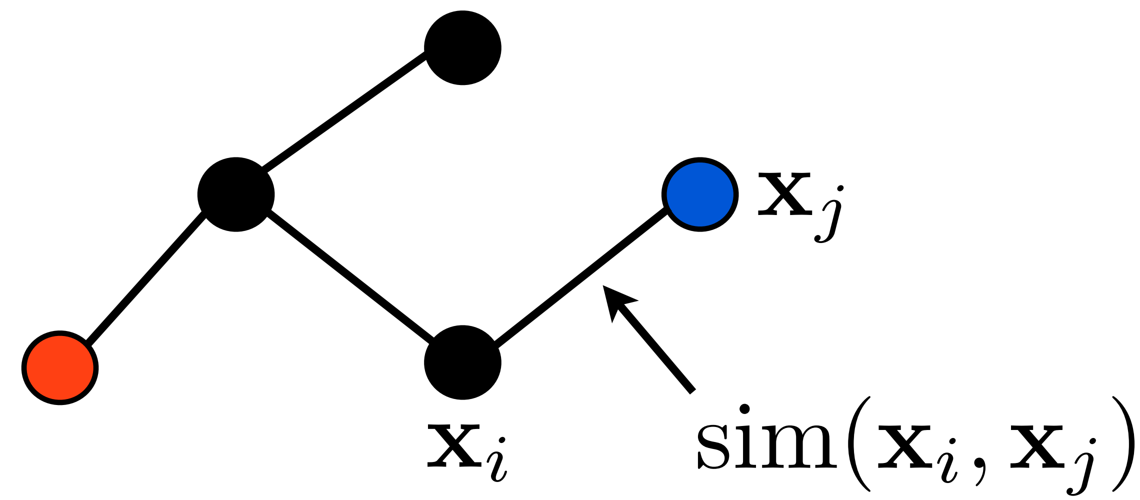
Graph-based SSL

Smoothness Assumption
If two instances are similar according to the graph, then output labels should be similar



Graph-based SSL

Smoothness Assumption
If two instances are similar according to the graph, then output labels should be similar



- Two stages
  - Graph construction (if not already present)
  - Label Inference

Outline

- Motivation
- Graph Construction
- Inference Methods
- Scalability
- Applications
- Conclusion & Future Work

Outline

- Motivation
- Graph Construction
- Inference Methods
- Scalability
- Applications
- Conclusion & Future Work

Graph Construction

- Neighborhood Methods
  - k-NN Graph Construction (k-NNG)
  - e-Neighborhood Method
- Metric Learning
- Other approaches

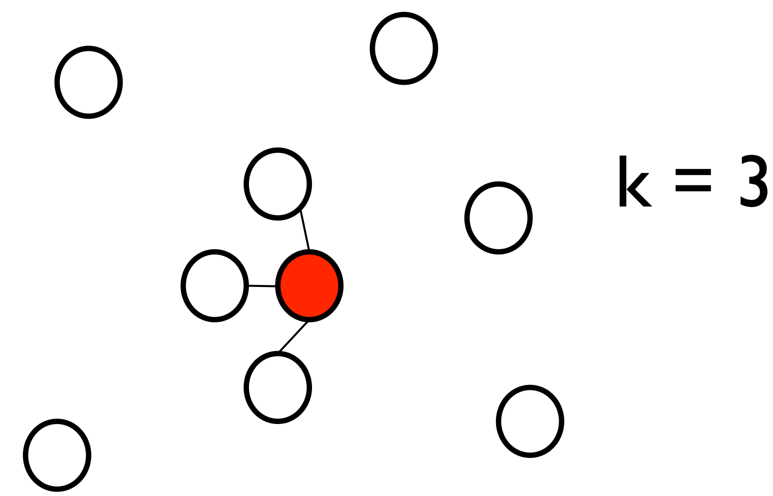
Neighborhood Methods

Neighborhood Methods

- k-Nearest Neighbor Graph (k-NNG)
  - add edges between an instance and its k-nearest neighbors

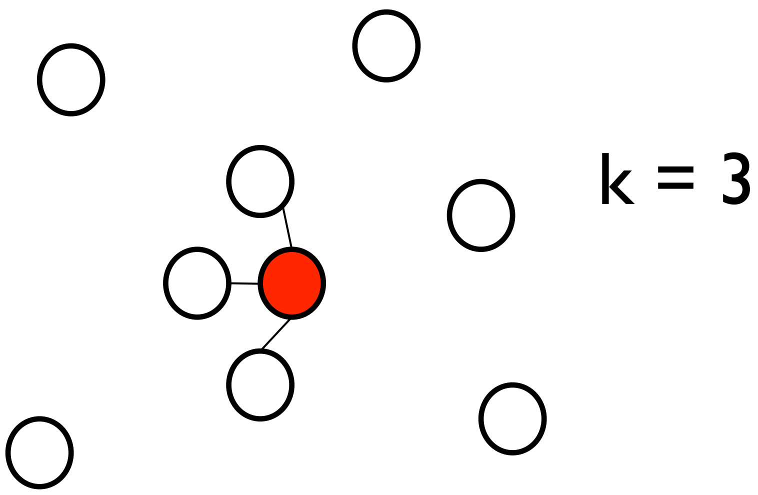
Neighborhood Methods

- k-Nearest Neighbor Graph (k-NNG)
  - add edges between an instance and its k-nearest neighbors



Neighborhood Methods

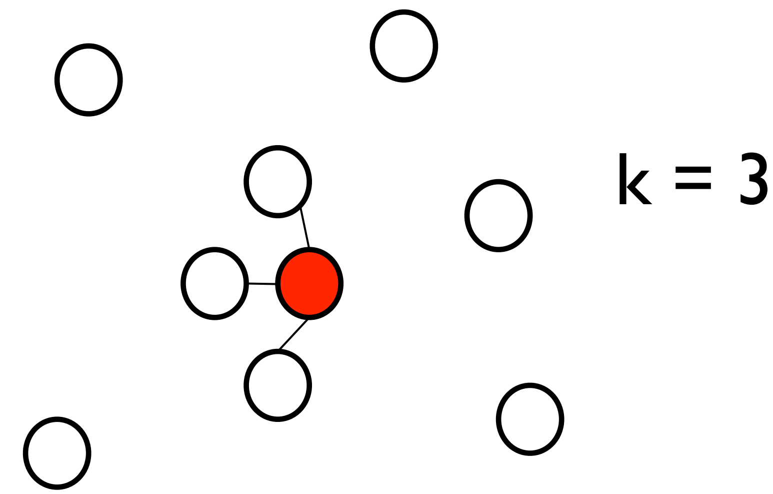
- k-Nearest Neighbor Graph (k-NNG)
  - add edges between an instance and its k-nearest neighbors



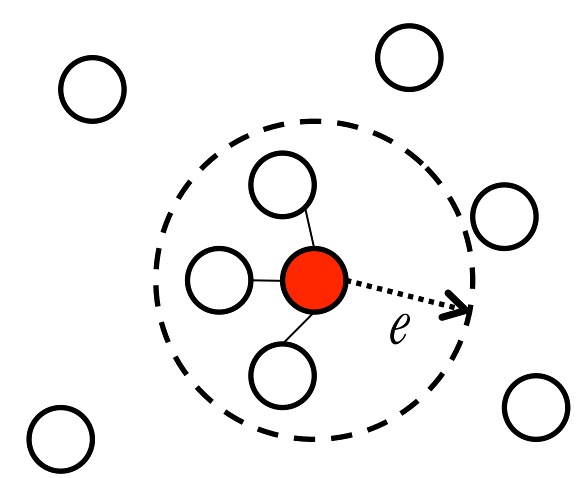
- e-Neighborhood
  - add edges to all instances inside a ball of radius e

Neighborhood Methods

- k-Nearest Neighbor Graph (k-NNG)
  - add edges between an instance and its k-nearest neighbors



- e-Neighborhood
  - add edges to all instances inside a ball of radius e



Issues with k-NNG

Issues with k-NNG

- Not scalable (quadratic)

Issues with k-NNG

- Not scalable (quadratic)
- Results in an asymmetric graph

Issues with k-NNG

- Not scalable (quadratic)
- Results in an asymmetric graph
  - b is the closest neighbor of a, but not the other way

Issues with k-NNG

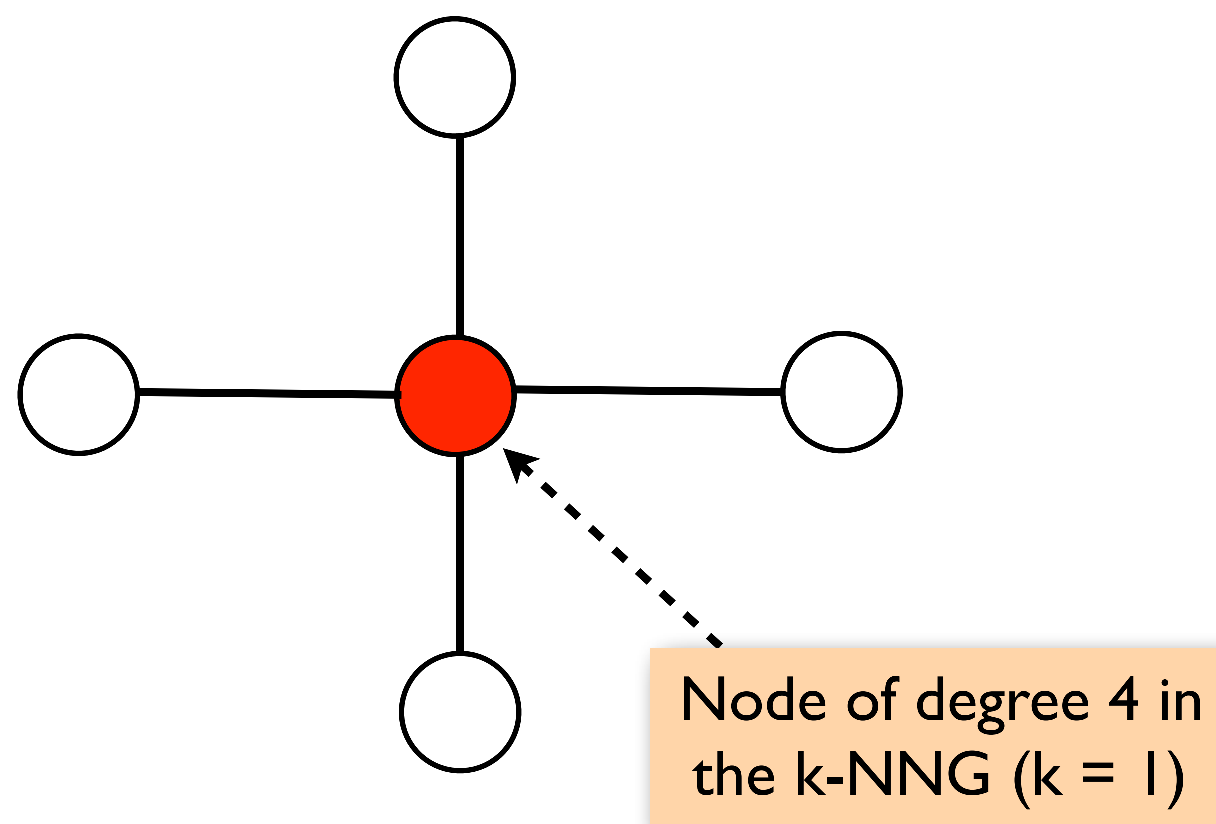
- Not scalable (quadratic)
- Results in an asymmetric graph
  - b is the closest neighbor of a, but not the other way

- Results in irregular graphs
  - some nodes may end up with higher degree than other nodes

Issues with k-NNG

- Not scalable (quadratic)
- Results in an asymmetric graph
  - b is the closest neighbor of a, but not the other way

- Results in irregular graphs
  - some nodes may end up with higher degree than other nodes



Issues with ϵ-Neighborhood

Issues with ϵ-Neighborhood

- Not scalable

Issues with e-Neighborhood

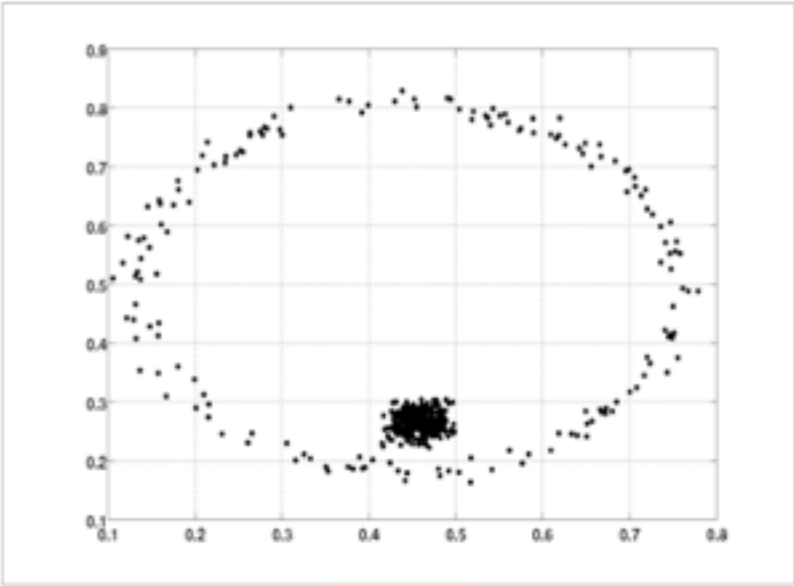
- Not scalable
- Sensitive to value of e : not invariant to scaling

Issues with ϵ-Neighborhood

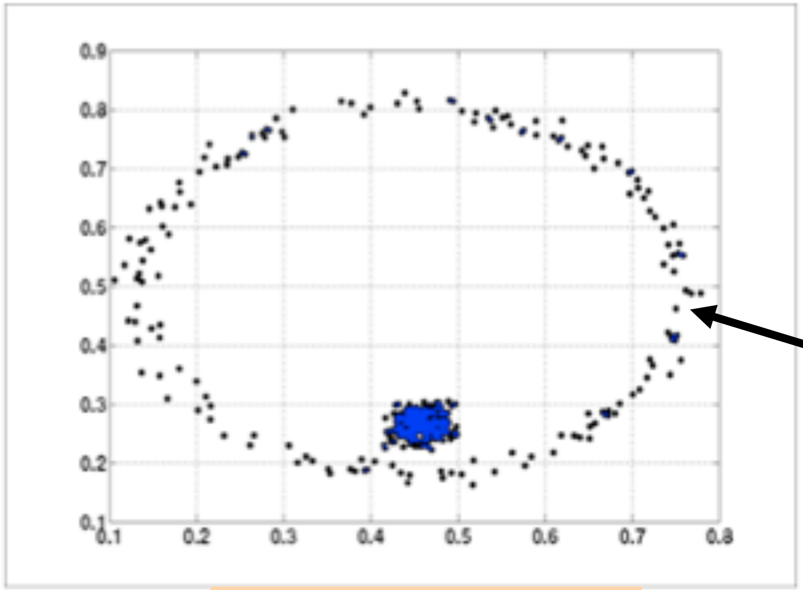
- Not scalable
- Sensitive to value of e : not invariant to scaling
- Fragmented Graph: disconnected components

Issues with *e*-Neighborhood

- Not scalable
- Sensitive to value of e : not invariant to scaling
- Fragmented Graph: disconnected components



Data



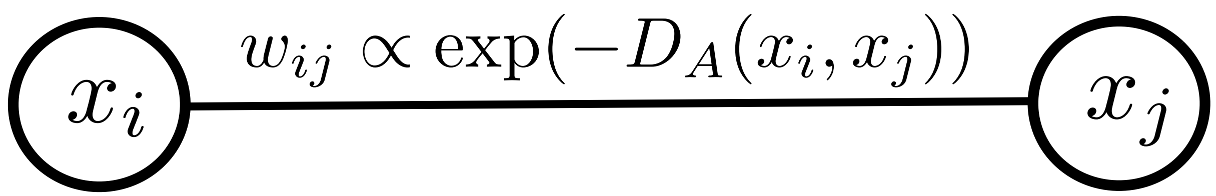
Disconnected

e-Neighborhood

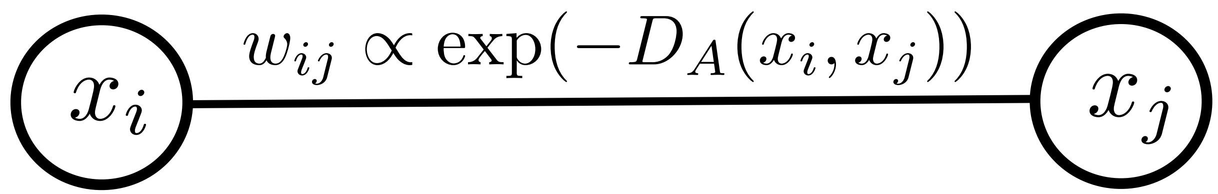
Figure from [Jebara et al., ICML 2009]

Graph Construction using Metric Learning

Graph Construction using Metric Learning



Graph Construction using Metric Learning



$$D_A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j)$$

Estimated using Mahalanobis metric learning algorithms

Graph Construction using Metric Learning

$$x_i \xrightarrow{\;w_{ij} \propto \exp(-D_A(x_i, x_j))\;} x_j$$

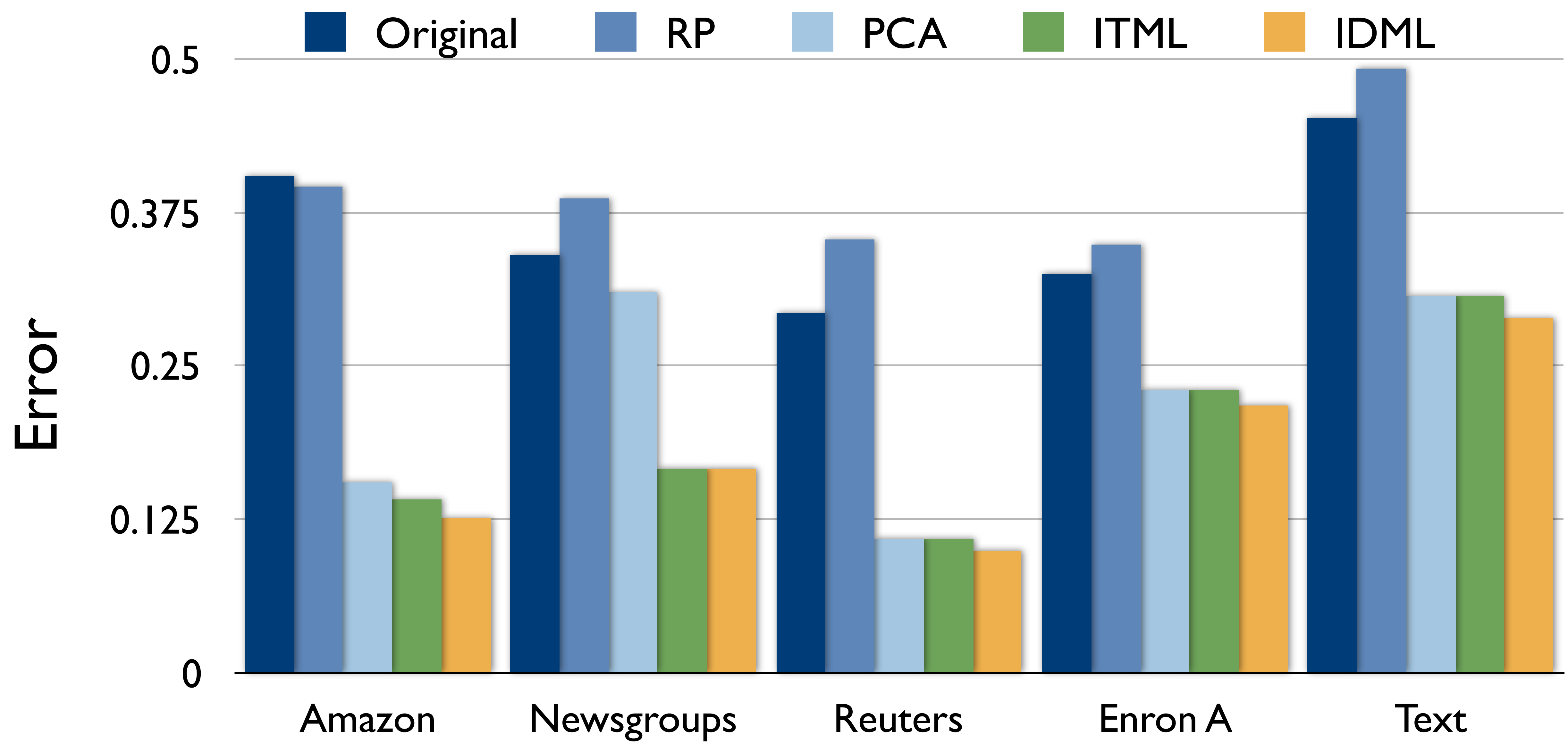
$$D_A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j)$$

Estimated using Mahalanobis metric learning algorithms

- Supervised Metric Learning
  - ITML [Kulis et al., ICML 2007]
  - LMNN [Weinberger and Saul, JMLR 2009]
- Semi-supervised Metric Learning
  - IDML [Dhillon et al., UPenn TR 2010]

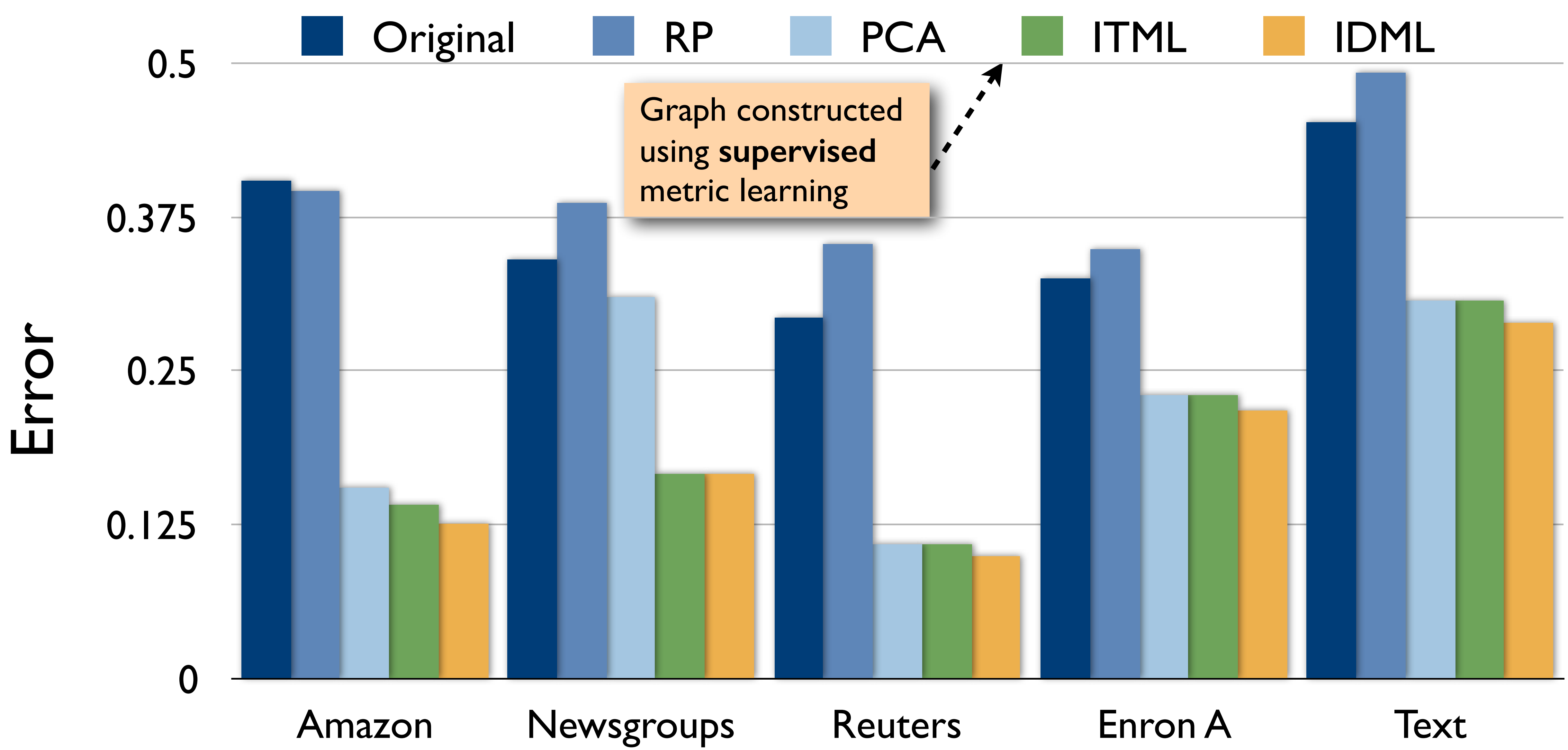
Benefits of Metric Learning for Graph Construction

Benefits of Metric Learning for Graph Construction



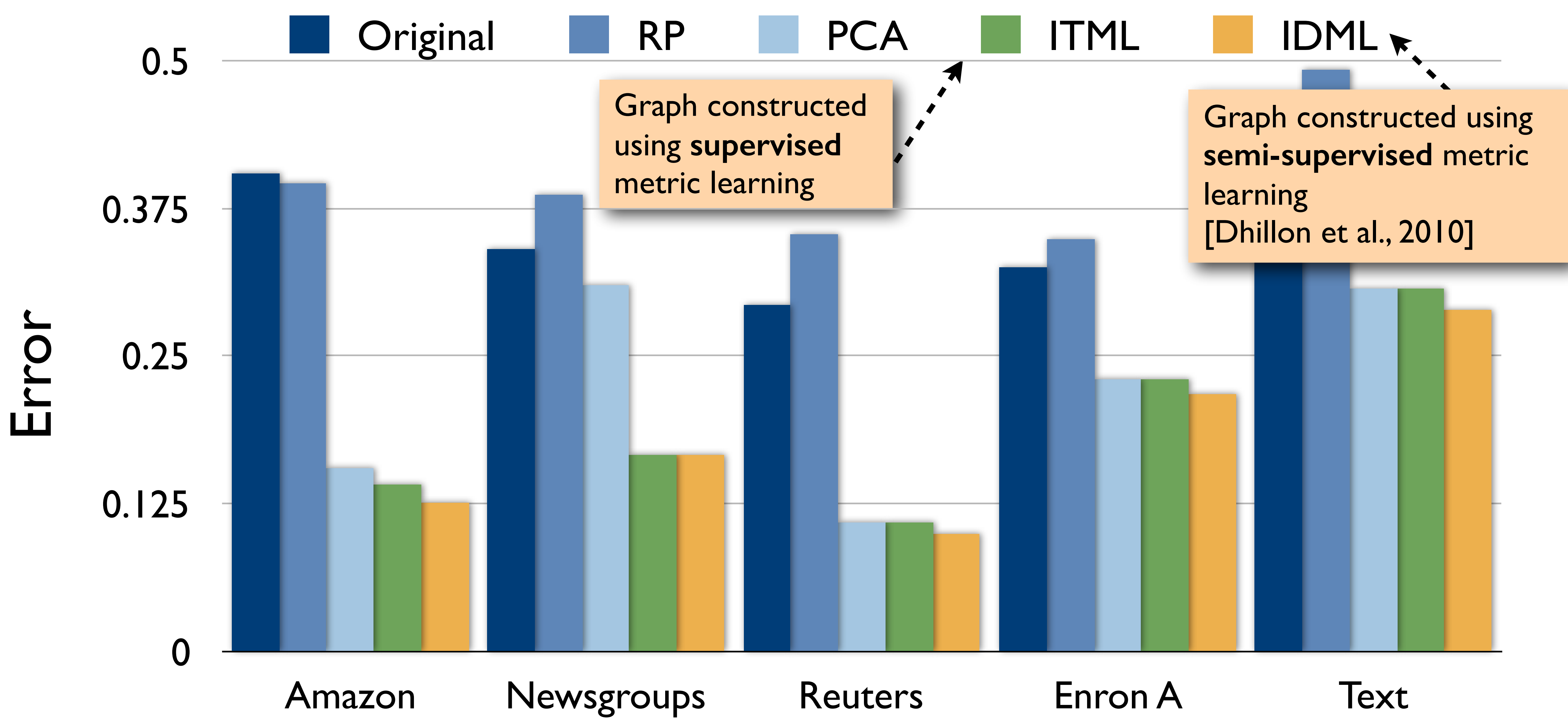
100 seed and1400 test instances, all inferences using LP

Benefits of Metric Learning for Graph Construction



100 seed and1400 test instances, all inferences using LP

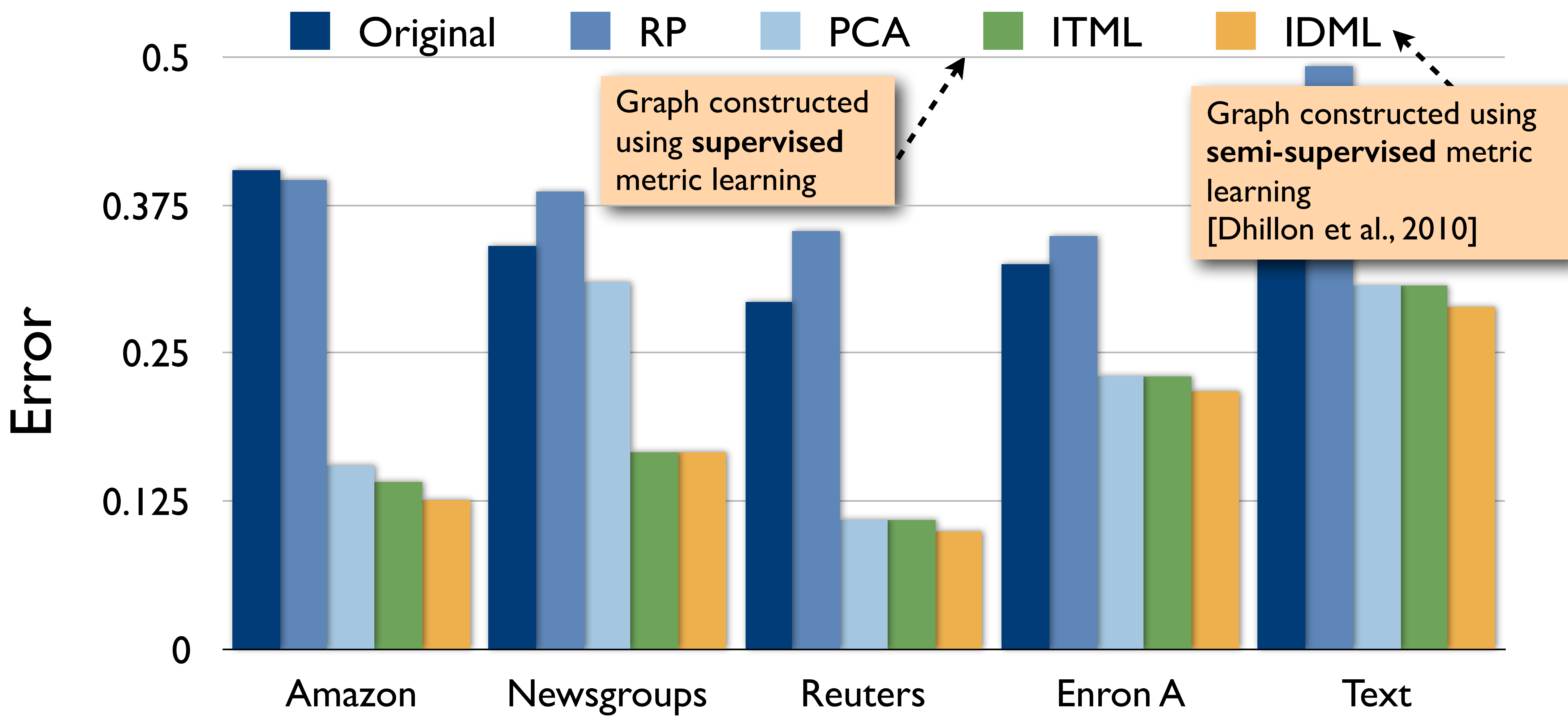
Benefits of Metric Learning for Graph Construction



100 seed and1400 test instances, all inferences using LP

[Dhillon et al., UPenn TR 2010]

Benefits of Metric Learning for Graph Construction



100 seed and1400 test instances, all inferences using LP

Careful graph construction is critical!

[Dhillon et al., UPenn TR 2010]

Other Graph Construction Approaches

- Local Reconstruction
  - Linear Neighborhood [Wang and Zhang, ICML 2005]
  - Regular Graph: b-matching [Jebara et al., ICML 2008]
  - Fitting Graph to Vector Data [Daitch et al., ICML 2009]
- Graph Kernels
  - [Zhu et al., NIPS 2005]

Outline

- Motivation
- Graph Construction
- Inference Methods
    - Label Propagation
    - Modified Adsorption
    - Measure Propagation
    - Sparse Label Propagation
    - Manifold Regularization
- Scalability
- Applications
- Conclusion & Future Work

Graph Laplacian

Graph Laplacian

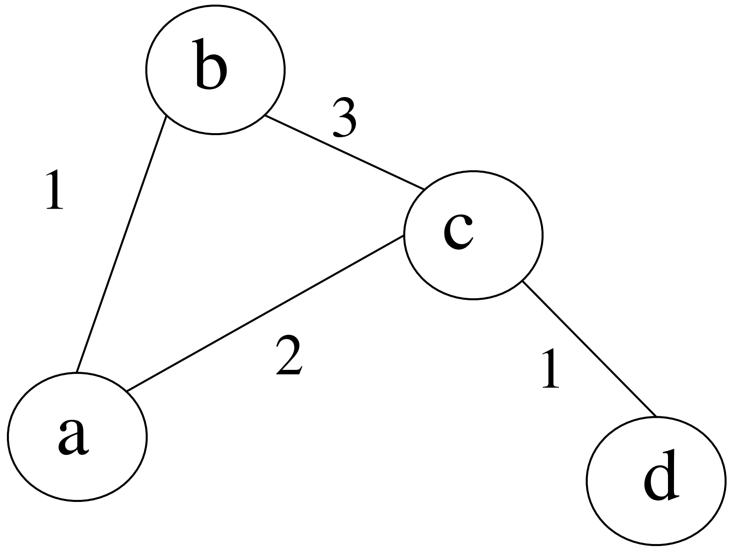
- Laplacian (un-normalized) of a graph:

$$L = D - W, \text{where } D_{ii} = \sum_j W_{ij}, \;\; D_{ij(\neq i)} = 0$$

Graph Laplacian

- Laplacian (un-normalized) of a graph:

$$L = D - W, \text{where } D_{ii} = \sum_j W_{ij},\ \ D_{ij(\neq i)} = 0$$

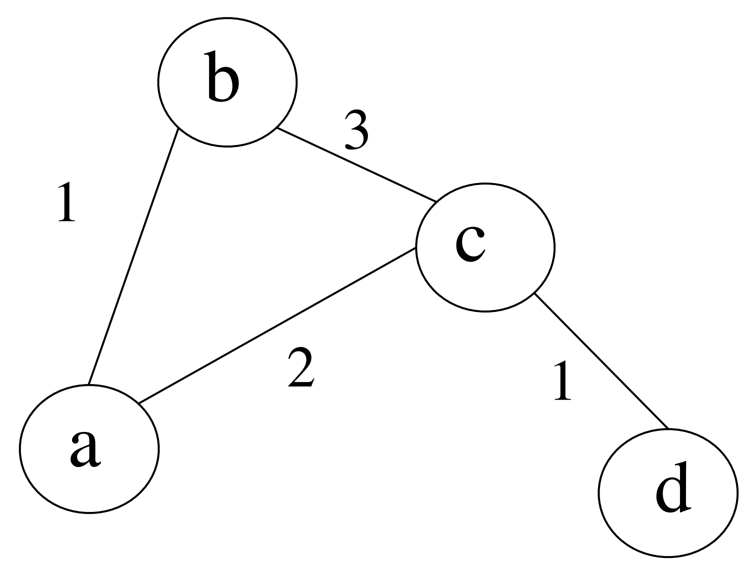


Graph Laplacian

- Laplacian (un-normalized) of a graph:

$$L = D - W, \text{where } D_{ii} = \sum_j W_{ij},\ \ D_{ij(\neq i)} = 0$$

$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \left(\begin{array}{cccc} \mathbf{3} & \mathbf{-1} & \mathbf{-2} & \mathbf{0} \\ \mathbf{-1} & \mathbf{4} & \mathbf{-3} & \mathbf{0} \\ \mathbf{-2} & \mathbf{-3} & \mathbf{6} & \mathbf{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{1} \end{array} \right) \end{array}$$



Graph Laplacian (contd.)

- *L* is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Graph Laplacian (contd.)

- *L* is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction *f* over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$

Graph Laplacian (contd.)

- L is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction f over the graph in terms of the Laplacian:

Measure of Non-Smoothness

$$f^T L f = \sum_{i,j} W_{ij}(f_i - f_j)^2$$

Graph Laplacian (contd.)

- *L* is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Vector of scores for single label on nodes

Measure of Non-Smoothness

$$f^T L f = \sum_{i,j} W_{ij}(f_i - f_j)^2$$

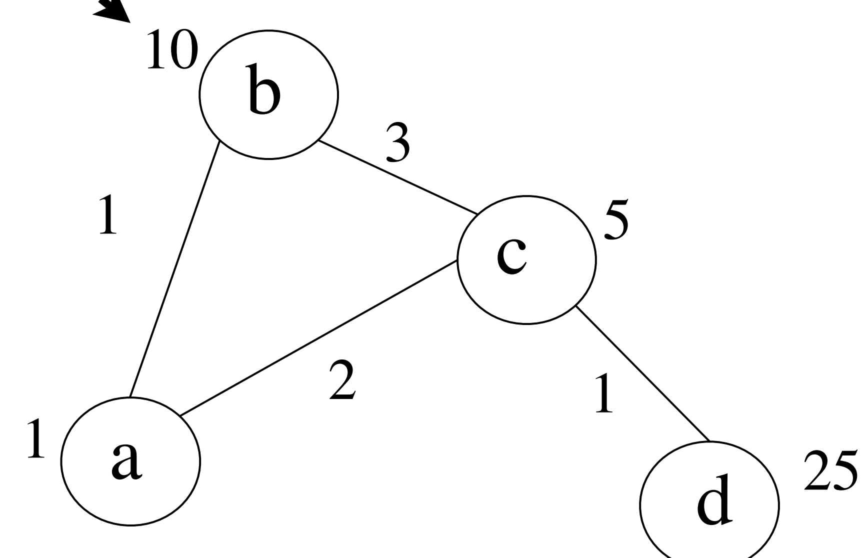
Graph Laplacian (contd.)

- L is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction f over the graph in terms of the Laplacian:

Vector of scores for single label on nodes

Measure of Non-Smoothness

$$f^T L f = \sum_{i,j} W_{ij}(f_i - f_j)^2$$



$$f^T = [1 \;\; 10 \;\; 5 \;\; 25]$$

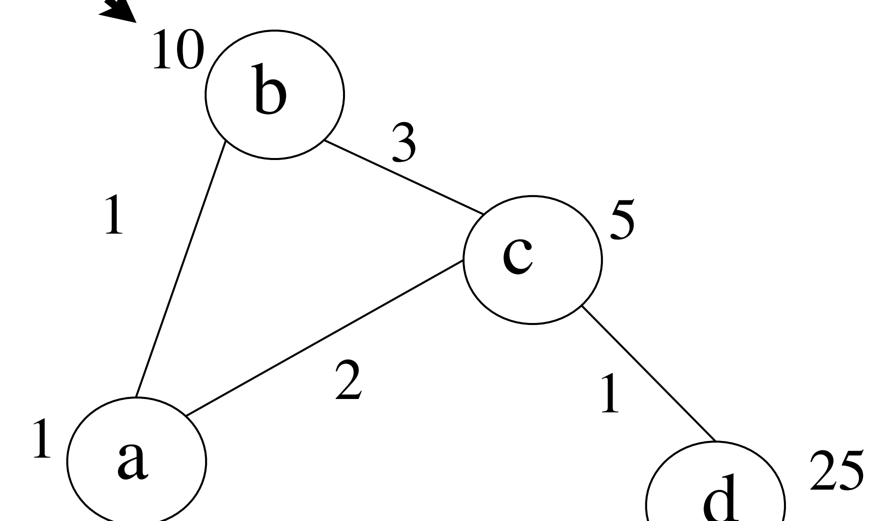
Graph Laplacian (contd.)

- *L* is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction *f* over the graph in terms of the Laplacian:

Vector of scores for single label on nodes

Measure of Non-Smoothness

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$



$$f^T = [1 \;\; 10 \;\; 5 \;\; 25]$$

$$f^T L f = 588$$

Not Smooth

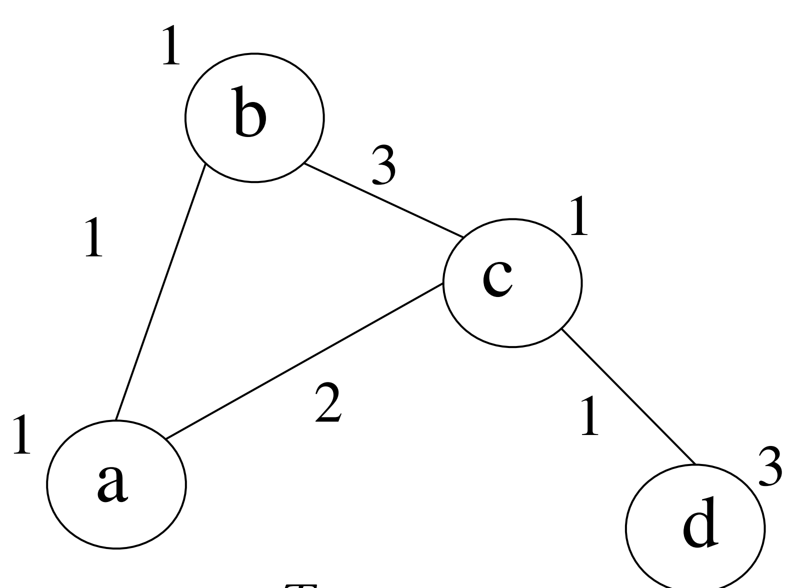
Graph Laplacian (contd.)

- L is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction f over the graph in terms of the Laplacian:

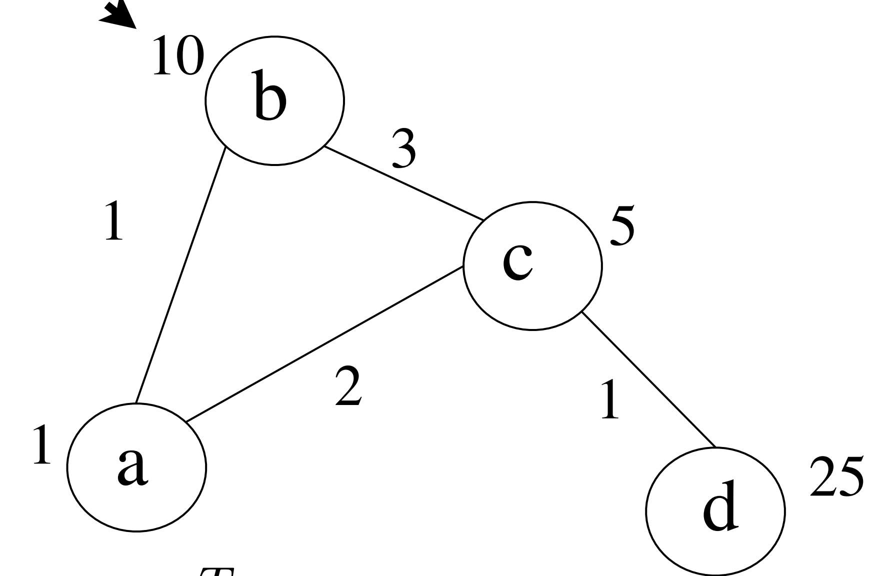
Vector of scores for single label on nodes

Measure of Non-Smoothness

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$



$f^T = [1113]$
$f^T L f = 4$



$f^T = [1\ 10\ 5\ 25]$
$f^T L f = 588$

Not Smooth

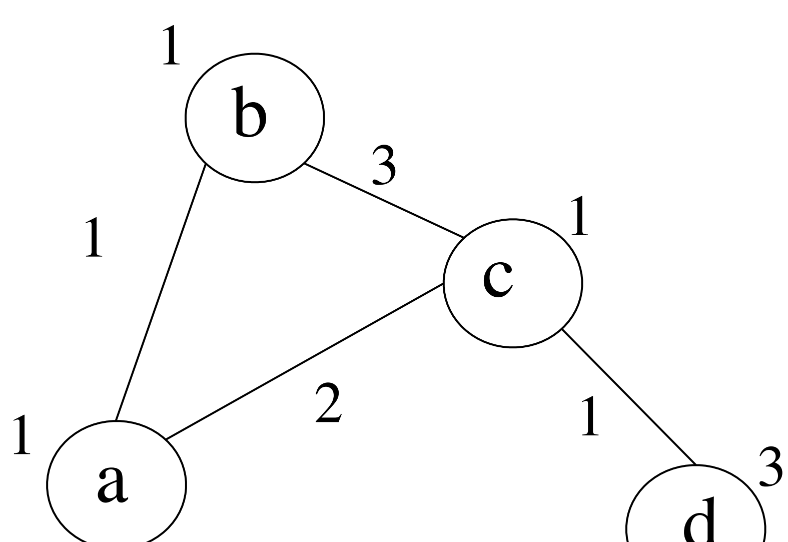
Graph Laplacian (contd.)

- L is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction f over the graph in terms of the Laplacian:

Vector of scores for single label on nodes

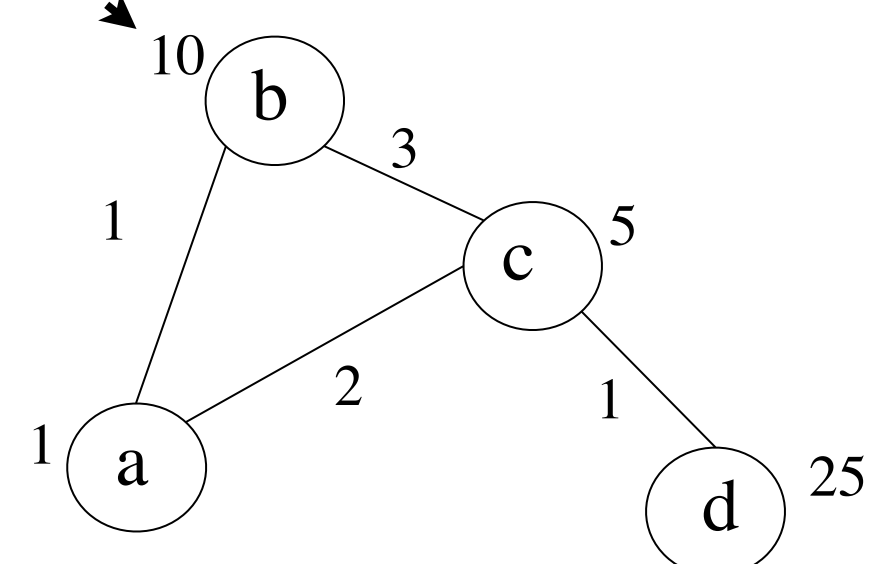
Measure of Non-Smoothness

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$



$f^T = [1113]$
$f^T L f = 4$

Smooth



$f^T = [1 \ 10 \ 5 \ 25]$
$f^T L f = 588$

Not Smooth

Outline

- Motivation
- Graph Construction
- Inference Methods
  - Label Propagation
  - Modified Adsorption
  - Measure Propagation
  - Sparse Label Propagation
  - Manifold Regularization
- Scalability
- Applications
- Conclusion & Future Work

Notations

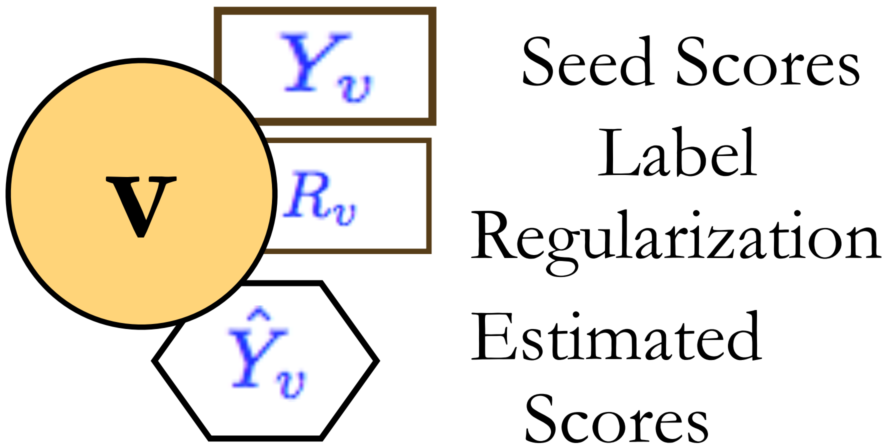
$Y_{v,l}$: score of seed label l on node v

$\hat{Y}_{v,l}$: score of estimated label l on node v

$R_{v,l}$: regularization target for label l on node v

S : seed node indicator (diagonal matrix)

W_{uv} : weight of edge (u, v) in the graph



LP-ZGL [Zhu et al., ICML 2003]

$$\arg\min_{\hat{Y}} \sum_{l=1}^{m} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2 = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that $Y_{ul} = \hat{Y}_{ul}, \; \forall S_{uu} = 1$

Graph Laplacian

LP-ZGL [Zhu et al., ICML 2003]

Smooth

$$\arg\min_{\hat{Y}} \boxed{\sum_{l=1}^{m} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2} = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that $Y_{ul} = \hat{Y}_{ul}, \ \forall S_{uu} = 1$

Graph Laplacian

LP-ZGL [Zhu et al., ICML 2003]

Smooth

$$\arg\min_{\hat{Y}} \boxed{\sum_{l=1}^{m} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2} = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that $\boxed{Y_{ul} = \hat{Y}_{ul},\ \forall S_{uu} = 1}$

Match Seeds (hard)

Graph Laplacian

LP-ZGL [Zhu et al., ICML 2003]

Smooth

$$\arg\min_{\hat{Y}} \boxed{\sum_{l=1}^{m} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2} = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that $\boxed{Y_{ul} = \hat{Y}_{ul},\ \forall S_{uu} = 1}$

Match Seeds (hard)

Graph Laplacian

- Smoothness
  - two nodes connected by an edge with high weight should be assigned similar labels

LP-ZGL [Zhu et al., ICML 2003]

Smooth

$$\arg\min_{\hat{Y}} \boxed{\sum_{l=1}^{m} W_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2} = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that $\boxed{Y_{ul} = \hat{Y}_{ul},\ \forall S_{uu} = 1}$

Match Seeds (hard)

Graph Laplacian

- Smoothness
  - two nodes connected by an edge with high weight should be assigned similar labels
- Solution satisfies harmonic property

Outline

- Motivation
- Graph Construction
- Inference Methods
  - Label Propagation
  - Modified Adsorption
  - Manifold Regularization
  - Spectral Graph Transduction
  - Measure Propagation
- Scalability
- Applications
- Conclusion & Future Work

Modified Adsorption (MAD)

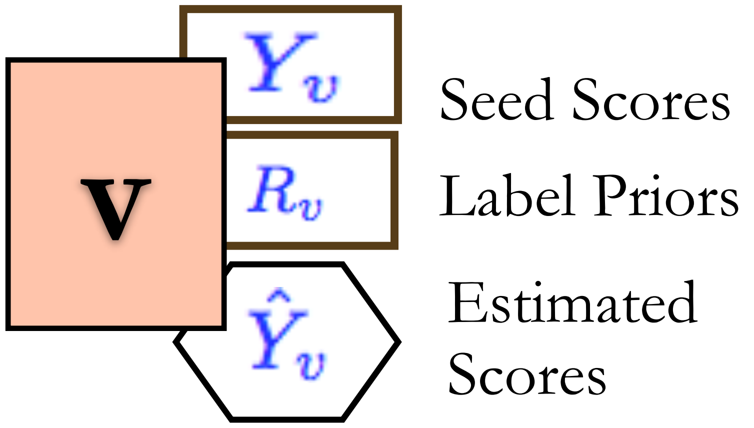
[Talukdar and Crammer, ECML 2009]

Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

$$\arg\min_{\hat{\boldsymbol{Y}}} \sum_{l=1}^{m+1} \left[\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2 + \mu_1 \sum_{u,v} \boldsymbol{M}_{uv} (\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2 + \mu_2 \|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2 \right]$$

- m labels, $+1$ dummy label
- $\boldsymbol{M} = \boldsymbol{W}'^{\top} + \boldsymbol{W}'$ is the symmetrized weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v



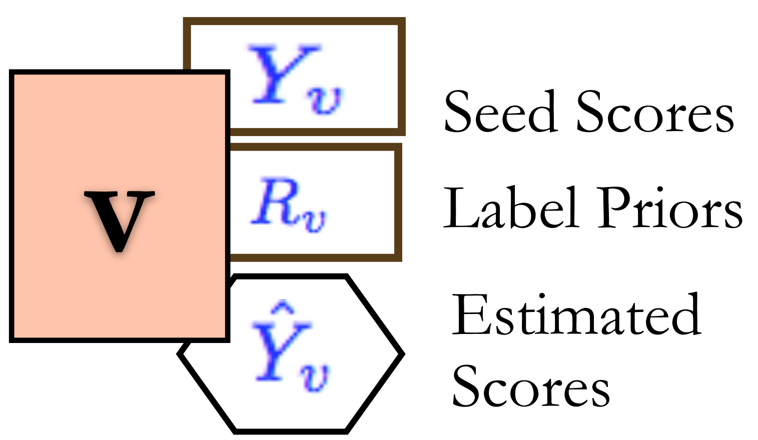
Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

Match Seeds (soft)

$$\arg\min_{\hat{\boldsymbol{Y}}} \sum_{l=1}^{m+1} \left[\boxed{\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2} + \mu_1 \sum_{u,v} \boldsymbol{M}_{uv}(\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2 + \mu_2 \|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2 \right]$$

- m labels, $+1$ dummy label
- $\boldsymbol{M} = \boldsymbol{W}'^{\top} + \boldsymbol{W}'$ is the symmetrized weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v

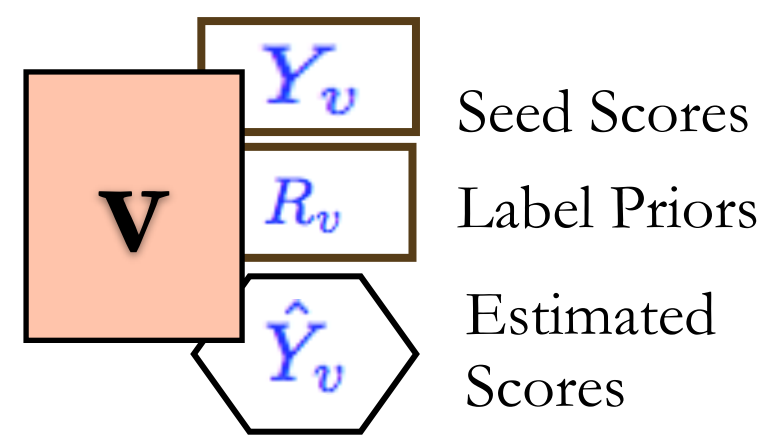


Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

$$\arg\min_{\hat{\boldsymbol{Y}}} \sum_{l=1}^{m+1} \left[\overset{\text{Match Seeds (soft)}}{\boxed{\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2}} + \mu_1 \overset{\text{Smooth}}{\boxed{\sum_{u,v} \boldsymbol{M}_{uv}(\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2}} + \mu_2 \|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2 \right]$$

- m labels, $+1$ dummy label
- $\boldsymbol{M} = \boldsymbol{W}'^{\top} + \boldsymbol{W}'$ is the symmetrized weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v

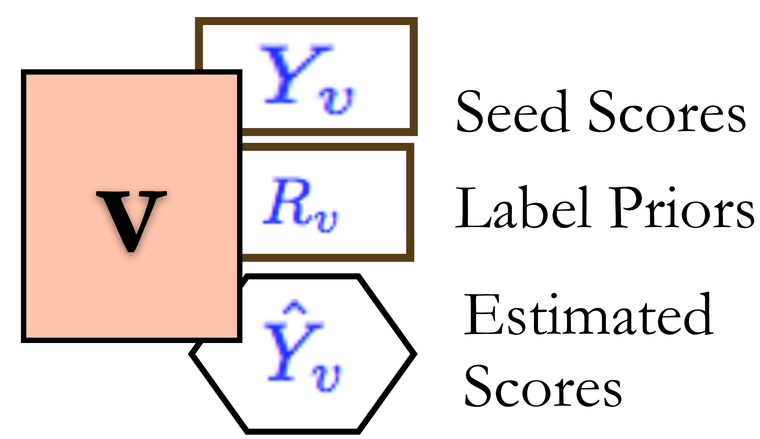


Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

$$\underset{\hat{\boldsymbol{Y}}}{\arg\min} \sum_{l=1}^{m+1} \left[\overset{\text{Match Seeds (soft)}}{\boxed{\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2}} + \mu_1 \overset{\text{Smooth}}{\boxed{\sum_{u,v} \boldsymbol{M}_{uv}(\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2}} + \overset{\text{Match Priors (Regularizer)}}{\boxed{\mu_2\|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2}} \right]$$

- m labels, +1 dummy label
- $\boldsymbol{M} = \boldsymbol{W}'^{\top} + \boldsymbol{W}'$ is the symmetrized weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v



Modified Adsorption (MAD)

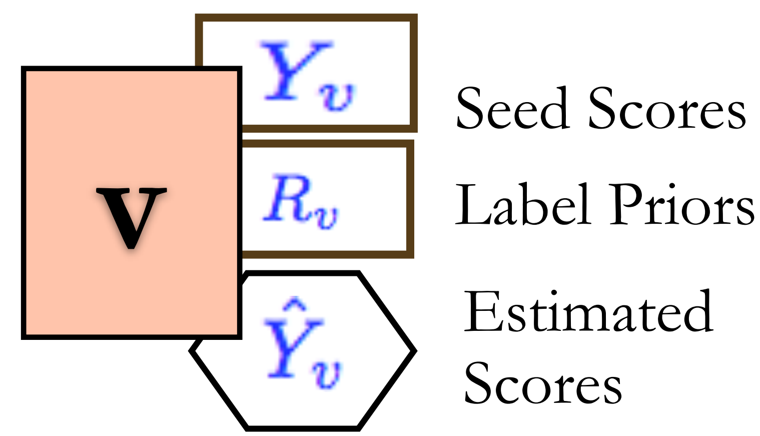
[Talukdar and Crammer, ECML 2009]

$$\arg\min_{\hat{\boldsymbol{Y}}} \sum_{l=1}^{m+1} \Big[\overbrace{\boxed{\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2}}^{\text{Match Seeds (soft)}} + \mu_1 \overbrace{\boxed{\sum_{u,v} \boldsymbol{M}_{uv}(\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2}}^{\text{Smooth}} + \overbrace{\boxed{\mu_2 \|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2}}^{\text{Match Priors (Regularizer)}} \Big]$$

- m labels, $+1$ dummy label

for *none-of-the-above* label

- $\boldsymbol{M} =$ ed weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v



Modified Adsorption (MAD)

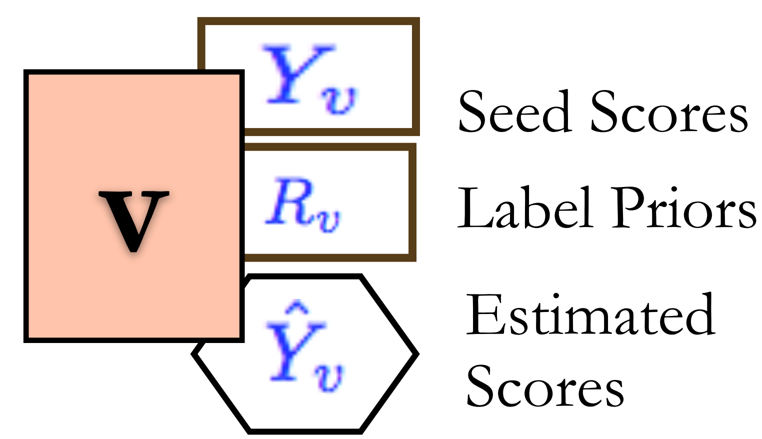
[Talukdar and Crammer, ECML 2009]

$$\arg\min_{\hat{\mathbf{Y}}} \sum_{l=1}^{m+1} \left[\overset{\text{Match Seeds (soft)}}{\boxed{\|\boldsymbol{S}\hat{\mathbf{Y}}_l - \boldsymbol{S}\mathbf{Y}_l\|^2}} + \mu_1 \overset{\text{Smooth}}{\boxed{\sum_{u,v} \boldsymbol{M}_{uv}(\hat{\mathbf{Y}}_{ul} - \hat{\mathbf{Y}}_{vl})^2}} + \overset{\text{Match Priors (Regularizer)}}{\boxed{\mu_2 \|\hat{\mathbf{Y}}_l - \boldsymbol{R}_l\|^2}} \right]$$

- m labels, $+1$ dummy label

for *none-of-the-above* label

- $\boldsymbol{M} =$ ...d weight matrix
- $\hat{\mathbf{Y}}_{vl}$: weight of label l on node v
- $\mathbf{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v



MAD has extra regularization compared to LP-ZGL [Zhu et al, ICML 03]; similar to QC [Bengio et al, 2006]

Modified Adsorption (MAD)

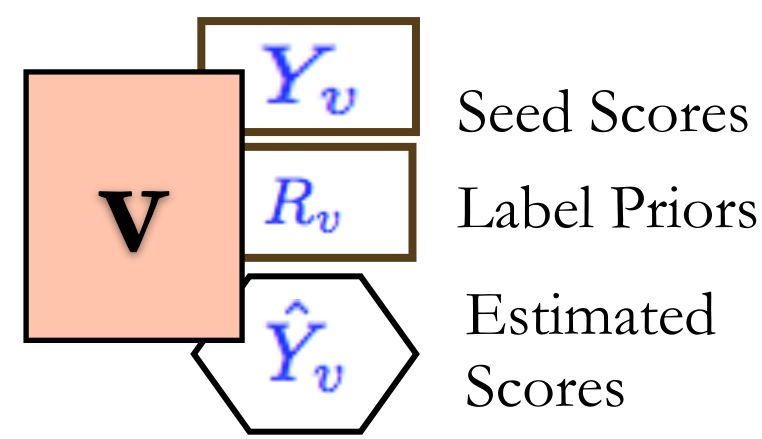
[Talukdar and Crammer, ECML 2009]

$$\arg\min_{\hat{\boldsymbol{Y}}} \sum_{l=1}^{m+1} \Big[\overbrace{\boxed{\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2}}^{\text{Match Seeds (soft)}} + \mu_1 \overbrace{\boxed{\sum_{u,v} \boldsymbol{M}_{uv}(\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2}}^{\text{Smooth}} + \overbrace{\boxed{\mu_2\|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2}}^{\text{Match Priors (Regularizer)}} \Big]$$

- m labels, $+1$ dummy label

for *none-of-the-above* label

- $\boldsymbol{M} =$ ...ed weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for seed nodes
- $\boldsymbol{R}_{vl}$: regularization target for label l on node v



MAD's Objective is Convex

MAD has extra regularization compared to LP-ZGL [Zhu et al, ICML 03]; similar to QC [Bengio et al, 2006]

Solving MAD Objective

Solving MAD Objective

- Can be solved using matrix inversion (like in LP)
  - but matrix inversion is expensive

Solving MAD Objective

- Can be solved using matrix inversion (like in LP)
  - but matrix inversion is expensive
- Instead solved exactly using a system of linear equations (Ax = b)
  - solved using Jacobi iterations
  - results in iterative updates
  - guaranteed convergence
  - see [Bengio et al., 2006] and
    [Talukdar and Crammer, ECML 2009] for details

Solving MAD using Iterative Updates

Inputs $\boldsymbol{Y}, \boldsymbol{R} : |V| \times (|L|+1)$, $\boldsymbol{W} : |V| \times |V|$, $\boldsymbol{S} : |V| \times |V|$ diagonal

$\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{Y}$

$\boldsymbol{M} = \boldsymbol{W}' + \boldsymbol{W}^{\dagger}$

$Z_v \leftarrow \boldsymbol{S}_{vv} + \mu_1 \sum_{u \neq v} \boldsymbol{M}_{vu} + \mu_2 \quad \forall v \in V$

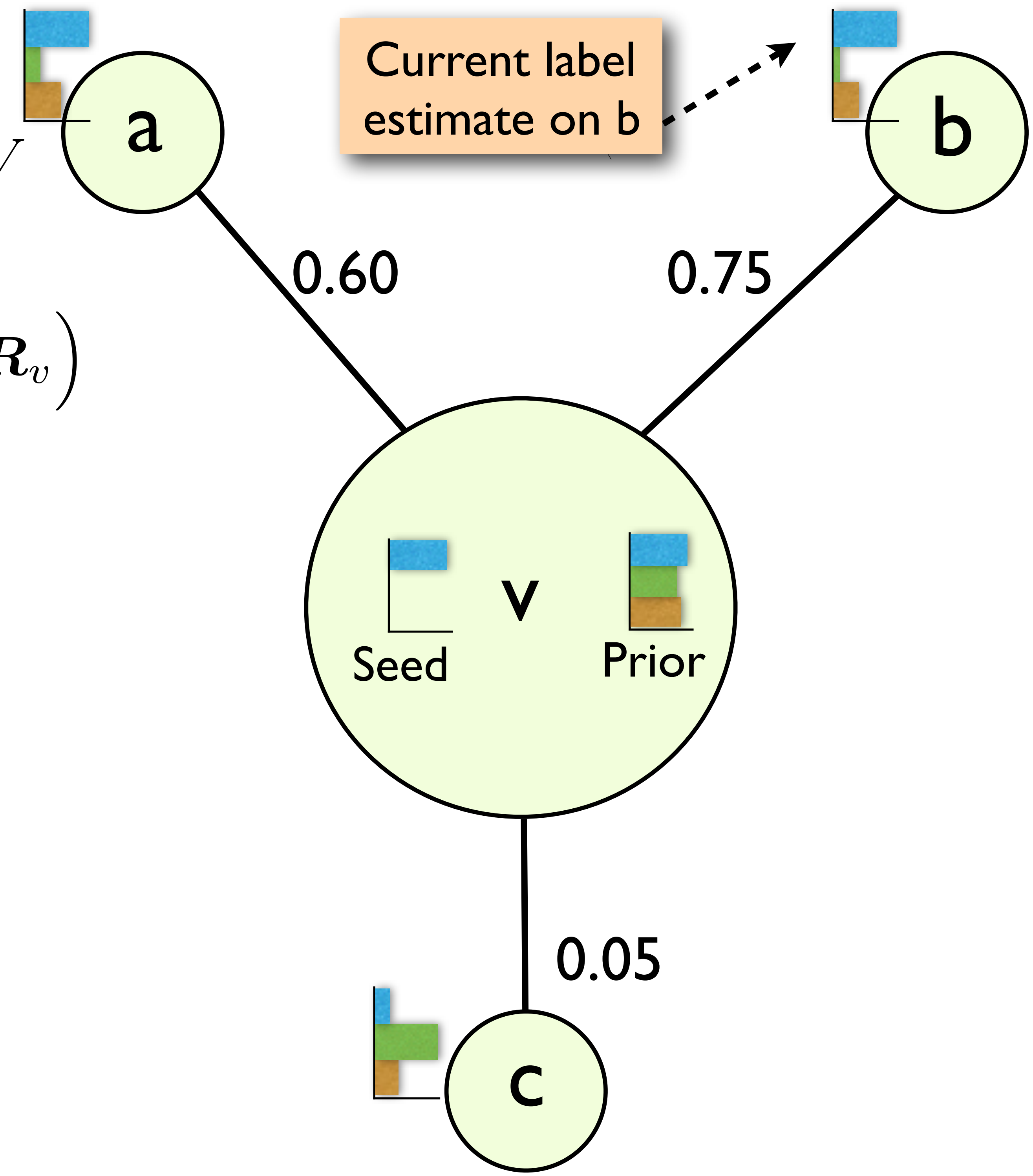
repeat

  for all $v \in V$ **do**

    $\hat{\boldsymbol{Y}}_v \leftarrow \frac{1}{Z_v}\left((\boldsymbol{S}\boldsymbol{Y})_v + \mu_1 \boldsymbol{M}_{v\cdot}\hat{\boldsymbol{Y}} + \mu_2 \boldsymbol{R}_v\right)$

  end for

until convergence



Solving MAD using Iterative Updates

Inputs $\boldsymbol{Y}, \boldsymbol{R} : |V| \times (|L|+1)$, $\boldsymbol{W} : |V| \times |V|$, $\boldsymbol{S} : |V| \times |V|$ diagonal

$\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{Y}$

$\boldsymbol{M} = \boldsymbol{W}' + \boldsymbol{W}^{\dagger}$

$Z_v \leftarrow \boldsymbol{S}_{vv} + \mu_1 \sum_{u \neq v} \boldsymbol{M}_{vu} + \mu_2 \quad \forall v \in V$

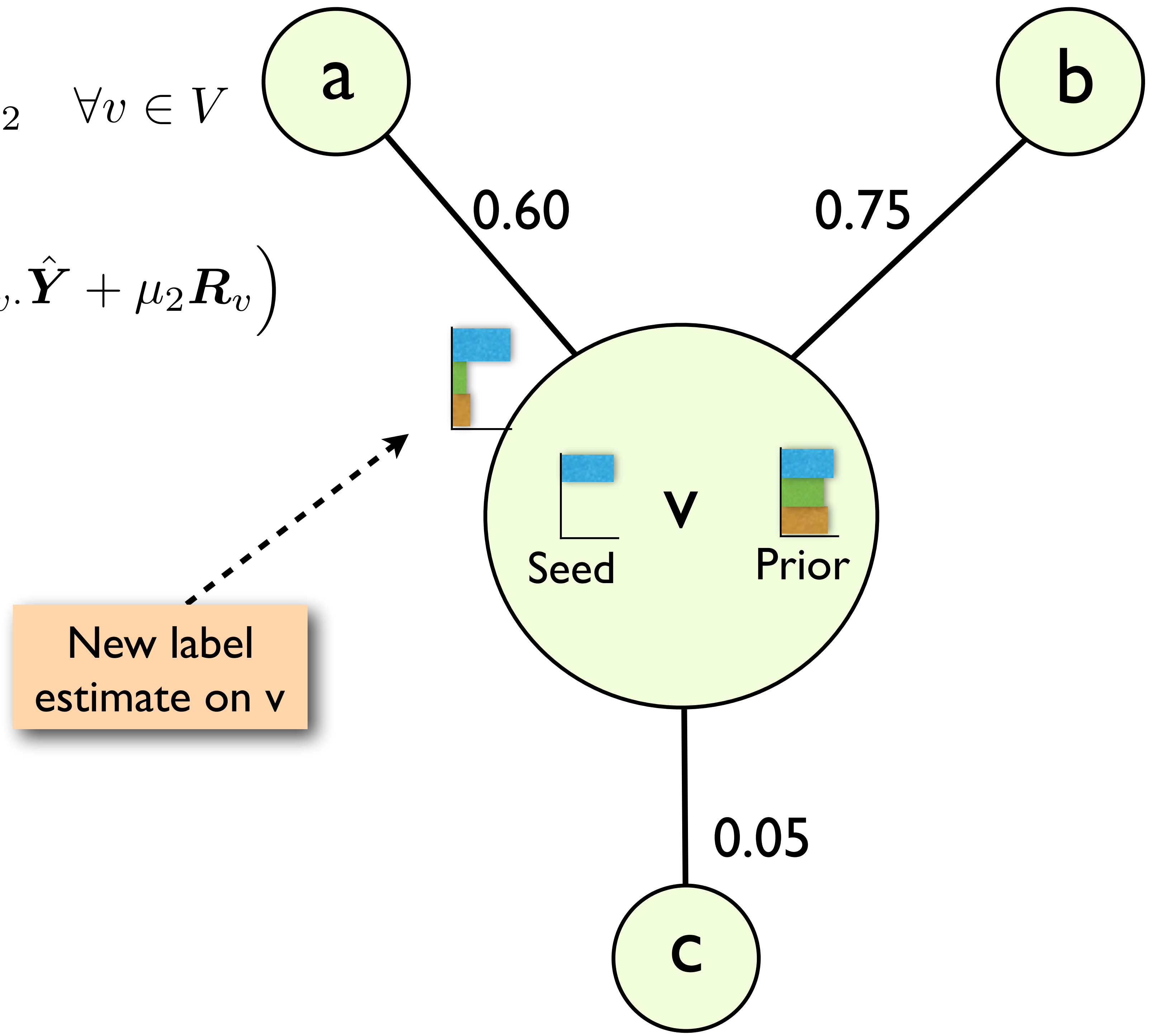
repeat

  for all $v \in V$ **do**

    $\hat{\boldsymbol{Y}}_v \leftarrow \frac{1}{Z_v} \left((\boldsymbol{SY})_v + \mu_1 \boldsymbol{M}_{v\cdot} \hat{\boldsymbol{Y}} + \mu_2 \boldsymbol{R}_v \right)$

  end for

until convergence



Solving MAD using Iterative Updates

Inputs $\boldsymbol{Y}, \boldsymbol{R} : |V| \times (|L|+1)$, $\boldsymbol{W} : |V| \times |V|$, $\boldsymbol{S} : |V| \times |V|$ diagonal

$\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{Y}$

$\boldsymbol{M} = \boldsymbol{W}' + \boldsymbol{W}^{\dagger}$

$Z_v \leftarrow \boldsymbol{S}_{vv} + \mu_1 \sum_{u \neq v} \boldsymbol{M}_{vu} + \mu_2 \quad \forall v \in V$

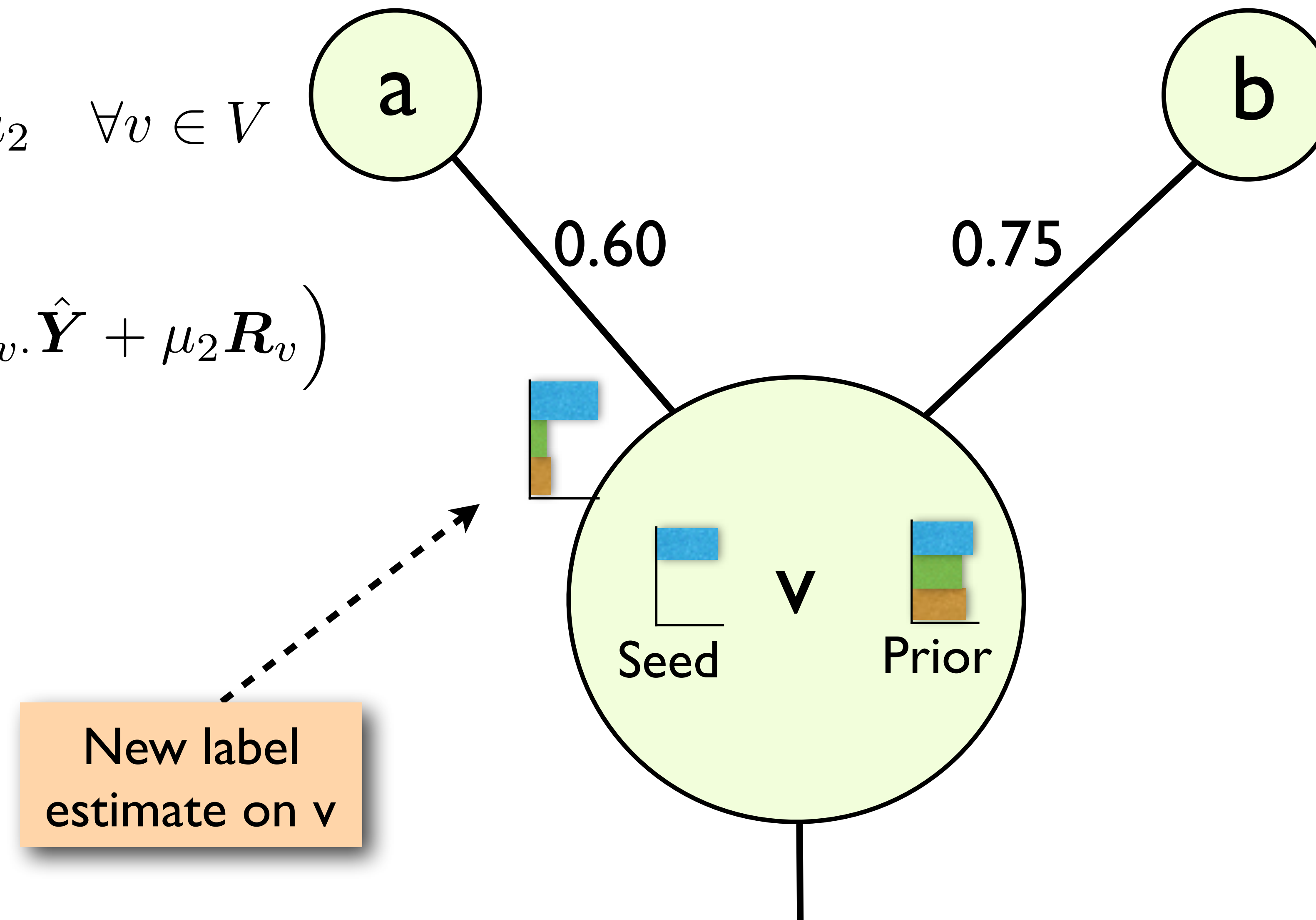
repeat

  for all $v \in V$ **do**

    $\hat{\boldsymbol{Y}}_v \leftarrow \frac{1}{Z_v}\left((\boldsymbol{SY})_v + \mu_1 \boldsymbol{M}_{v\cdot}\hat{\boldsymbol{Y}} + \mu_2 \boldsymbol{R}_v\right)$

  end for

until convergence



- Importance of a node can be discounted
- Easily Parallelizable: Scalable (more later)

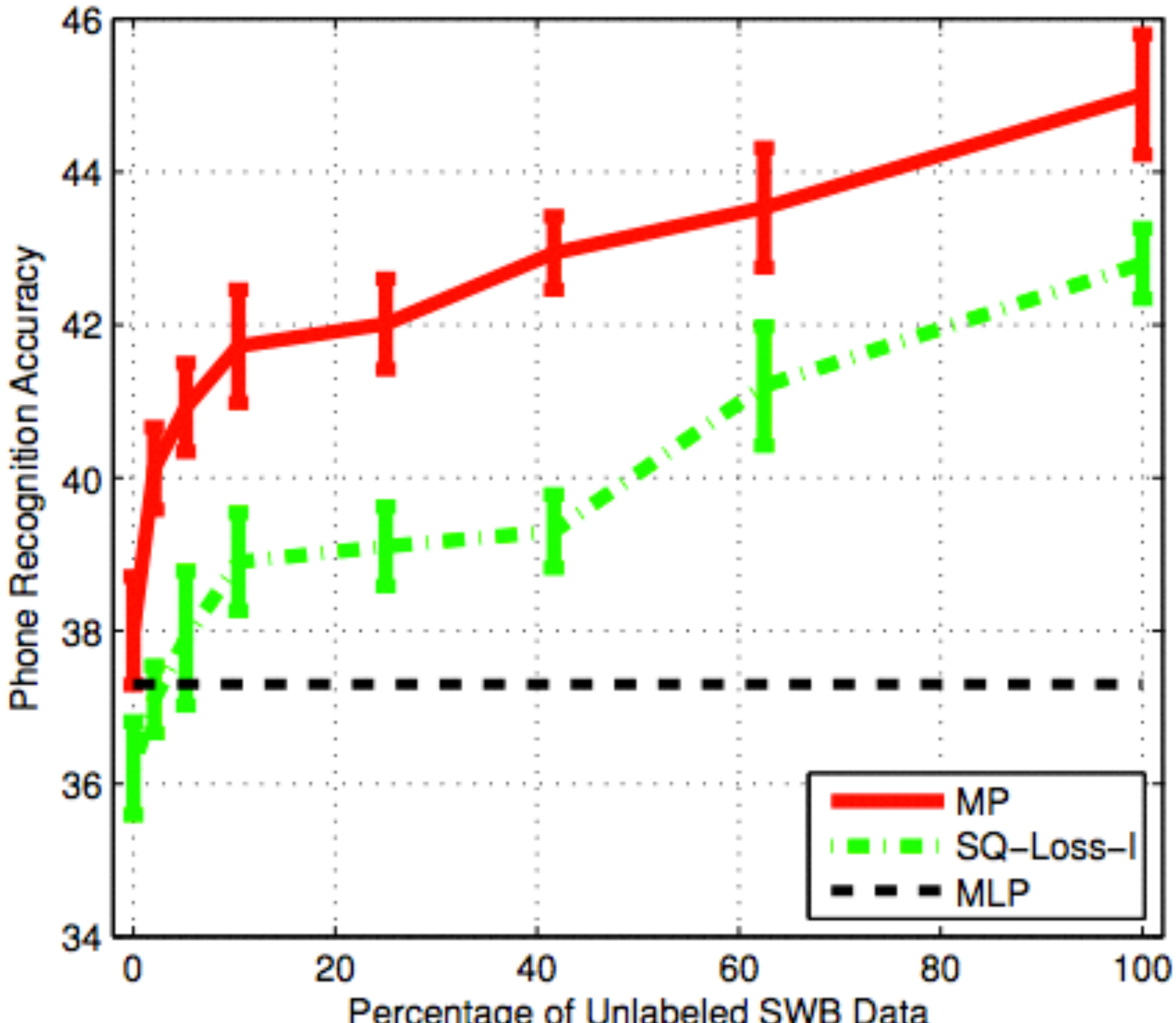
Other Graph-based SSL Methods

- TACO [Orbach and Crammer, ECML 2012]
- SSL on Directed Graphs
  - [Zhou et al, NIPS 2005], [Zhou et al., ICML 2005]
- Spectral Graph Transduction [Joachims, ICML 2003]
- Graph-SSL for Ordering
  - [Talukdar et al., CIKM 2012]
- Learning with dissimilarity edges
  - [Goldberg et al., AISTATS 2007]

Outline

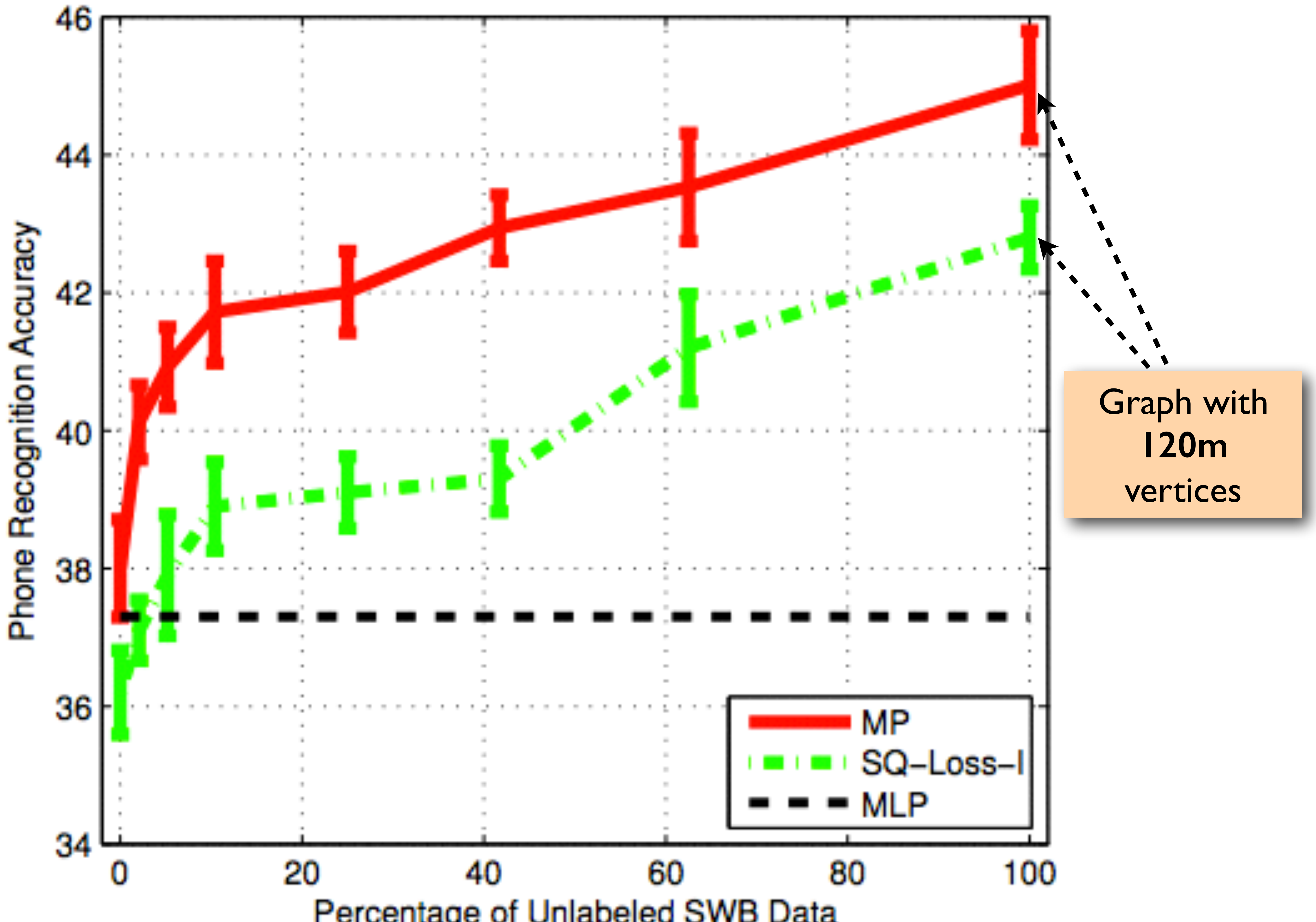
- Motivation
- Graph Construction
- Inference Methods
- Scalability
  - Scalability Issues
  - Node reordering
  - MapReduce Parallelization
- Applications
- Conclusion & Future Work

More (Unlabeled) Data is Better Data



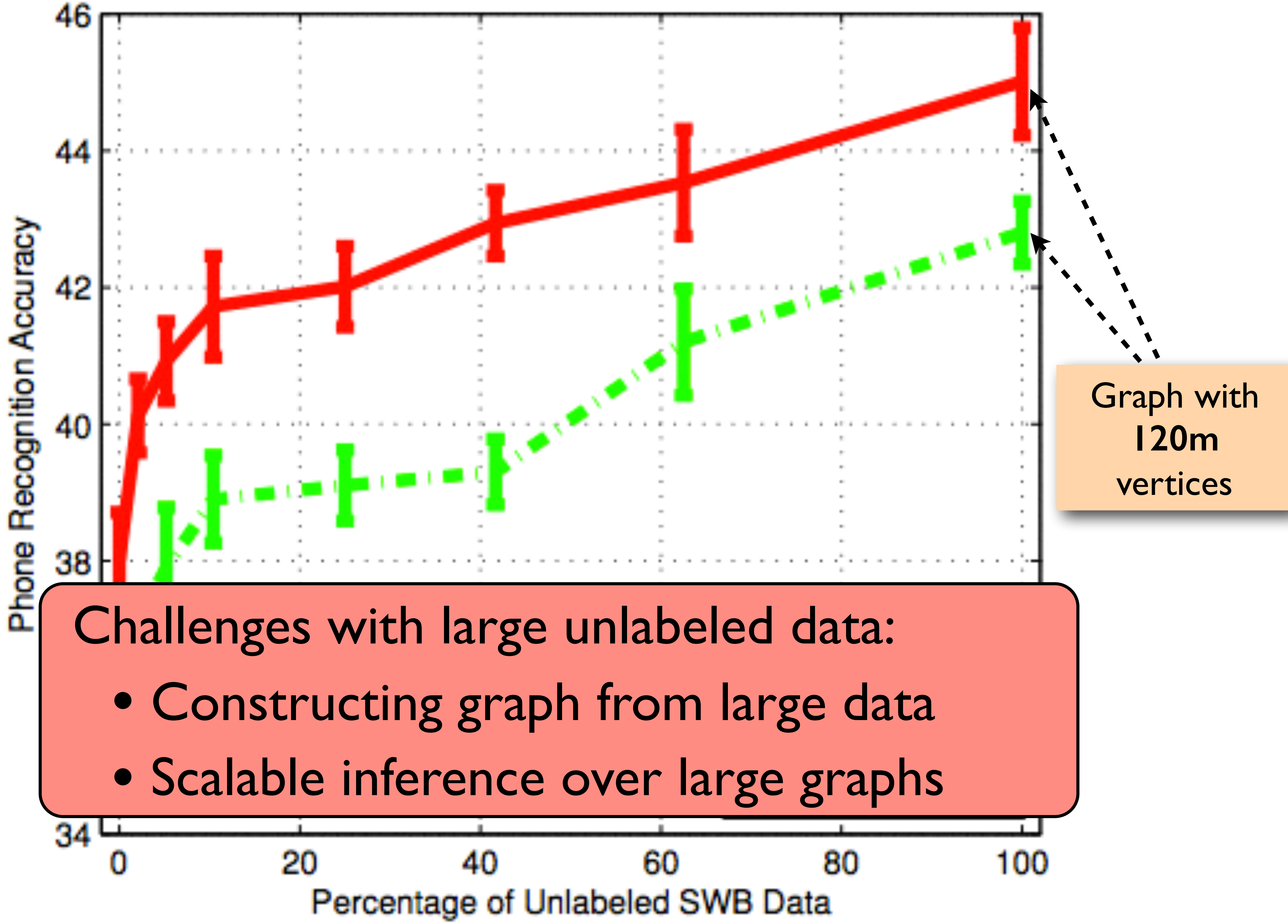
[Subramanya & Bilmes, JMLR 2011]

More (Unlabeled) Data is Better Data



[Subramanya & Bilmes, JMLR 2011]

More (Unlabeled) Data is Better Data



Challenges with large unlabeled data:

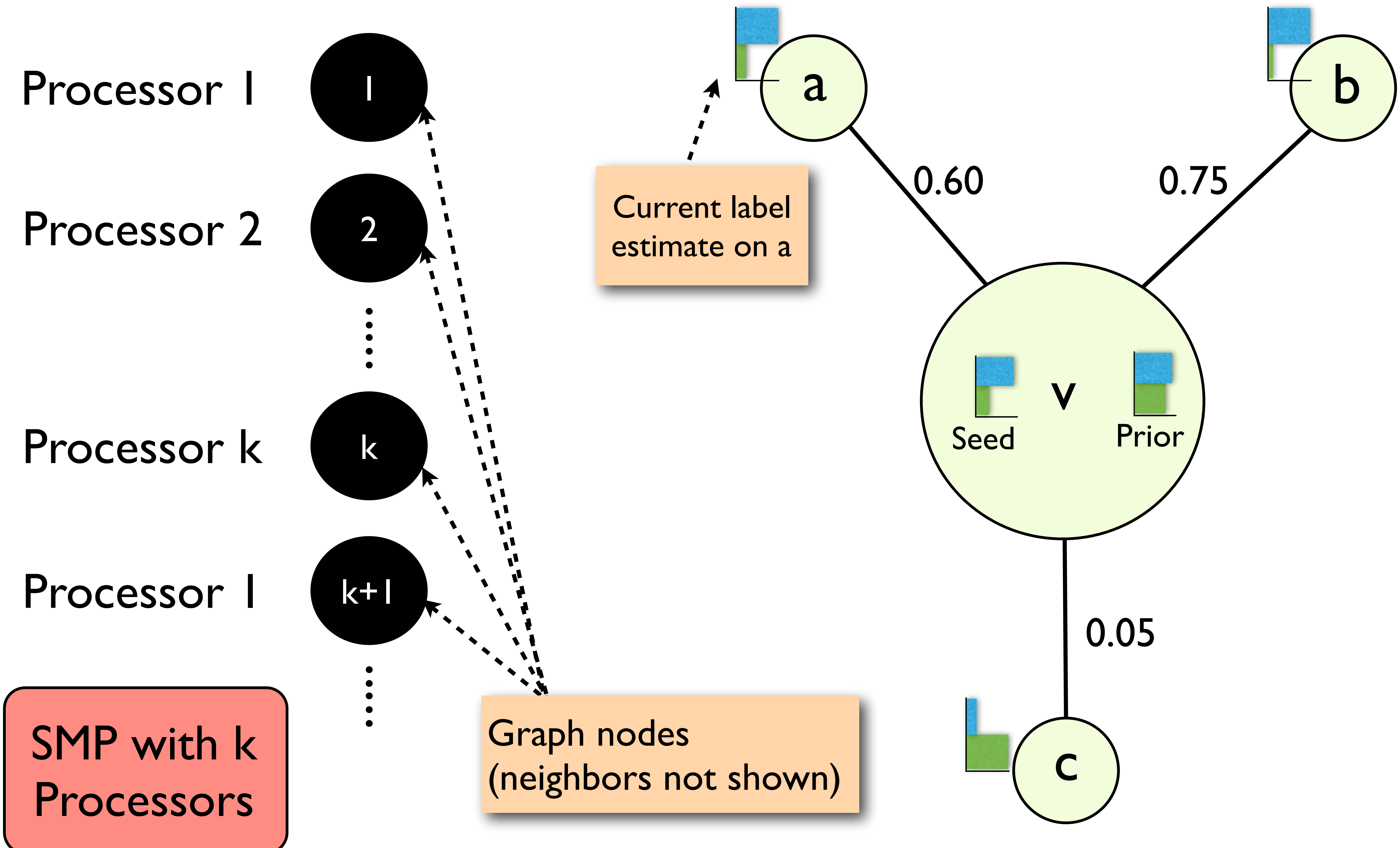
- Constructing graph from large data
- Scalable inference over large graphs

[Subramanya & Bilmes, JMLR 2011]

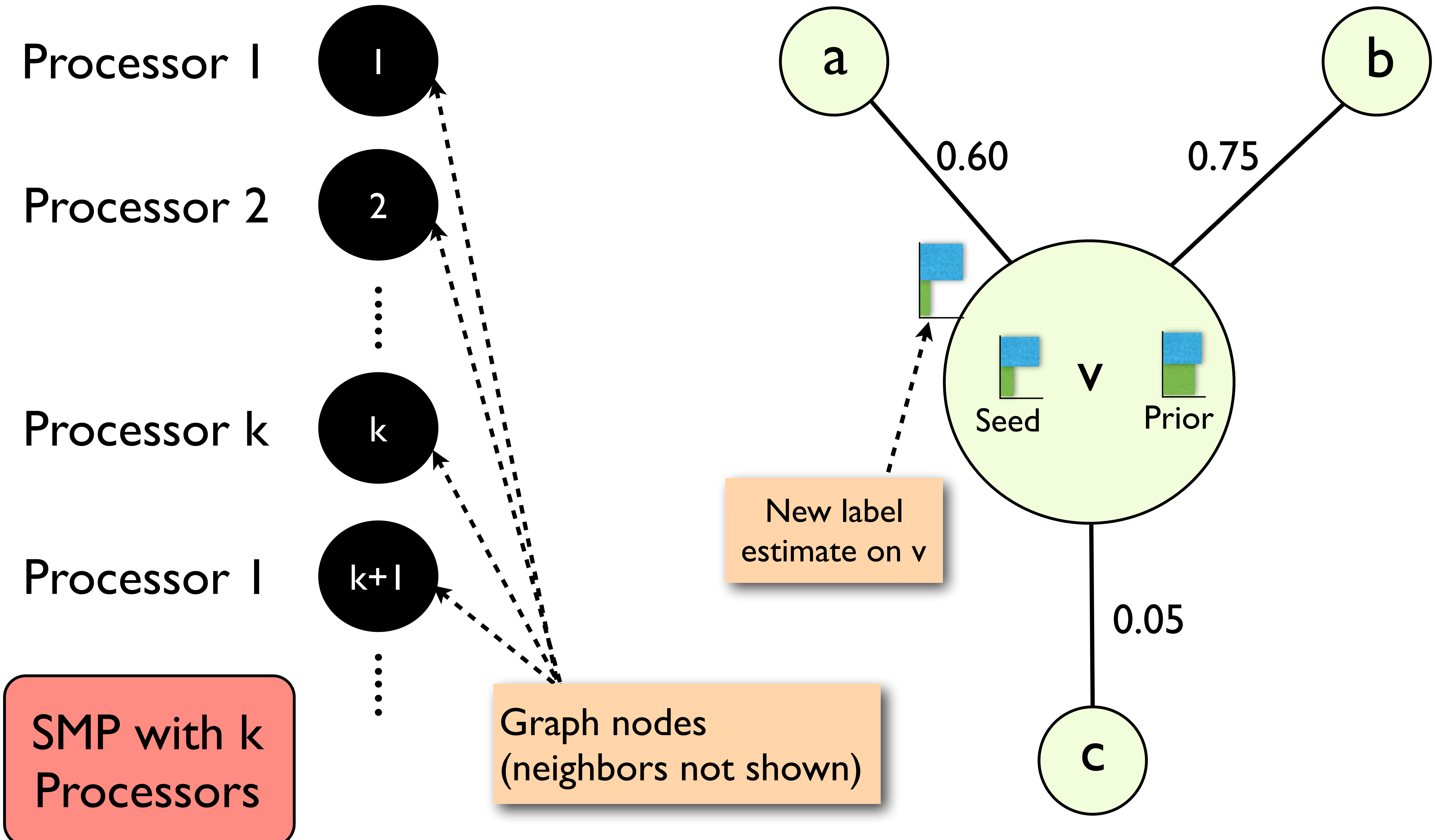
Outline

- Motivation
- Graph Construction
- Inference Methods
- Scalability
  - Scalability Issues
  - Node reordering [Subramanya & Bilmes, JMLR 2011; Bilmes & Subramanya, 2011]
  - MapReduce Parallelization
- Applications
- Conclusion & Future Work

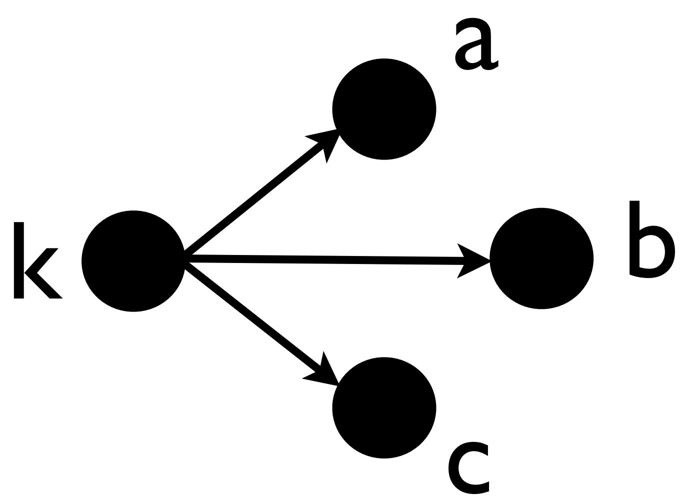
Label Update using Message Passing



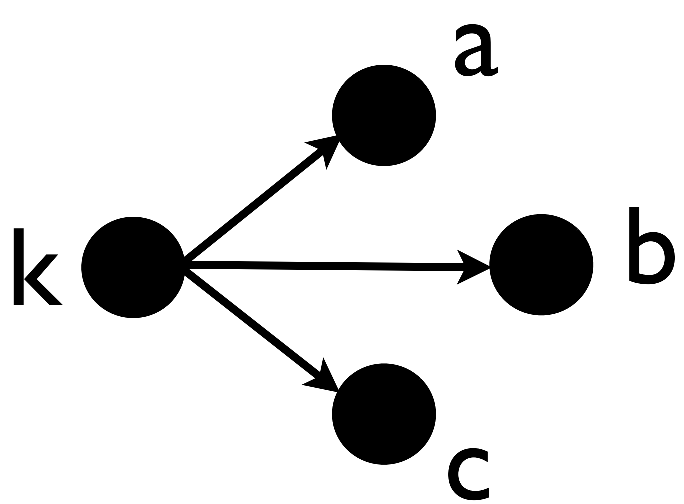
Label Update using Message Passing



Node Reordering Algorithm : Intuition

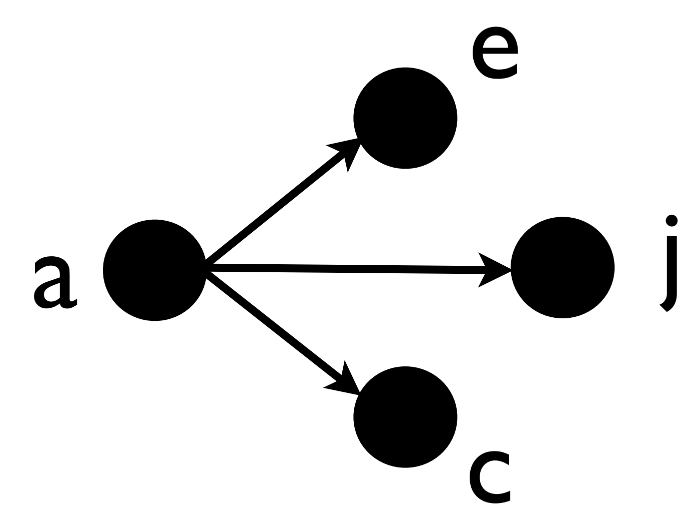


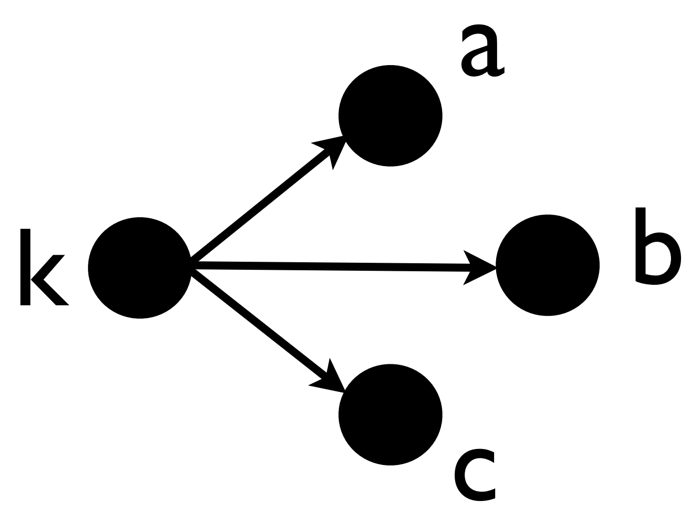
Node Reordering Algorithm : Intuition



Which node should be processed along with k: the one with highest intersection of neighborhood with k

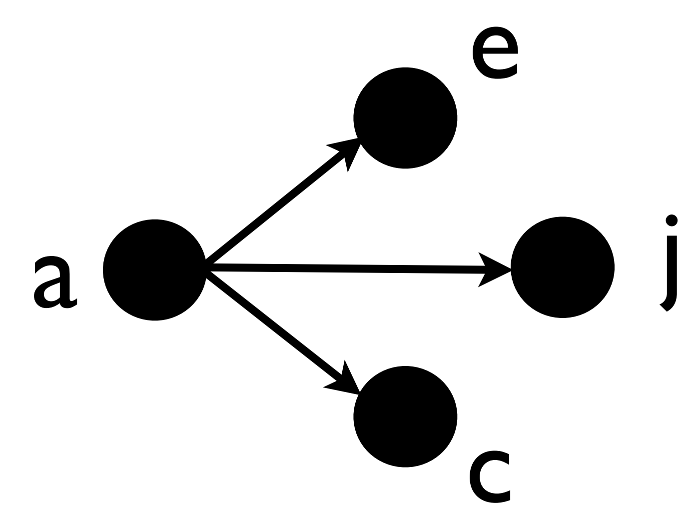
Node Reordering Algorithm : Intuition

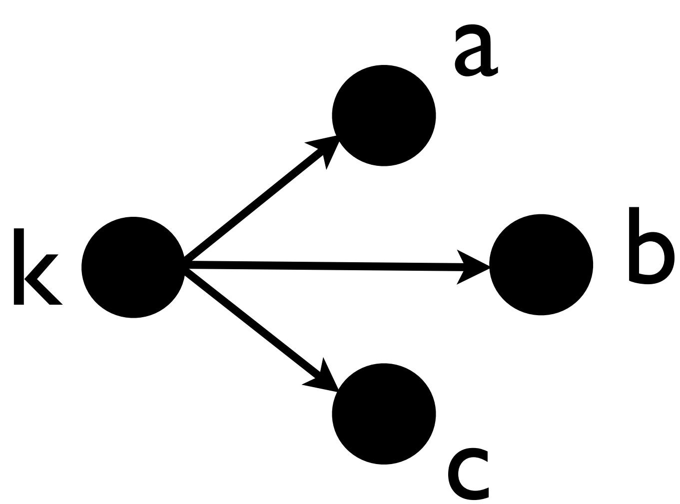


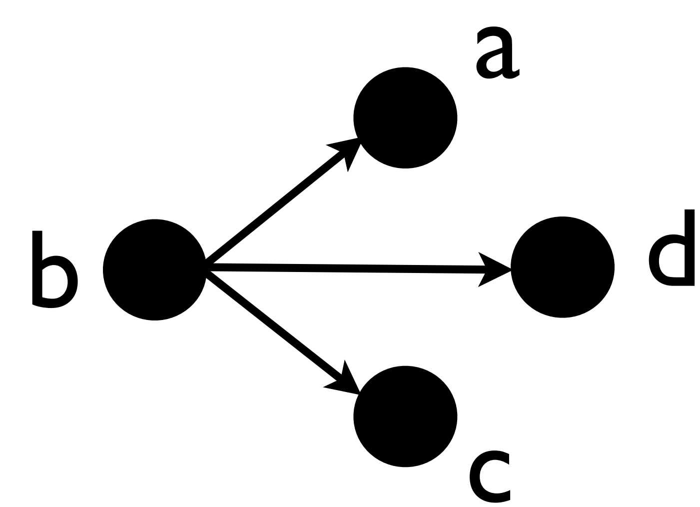


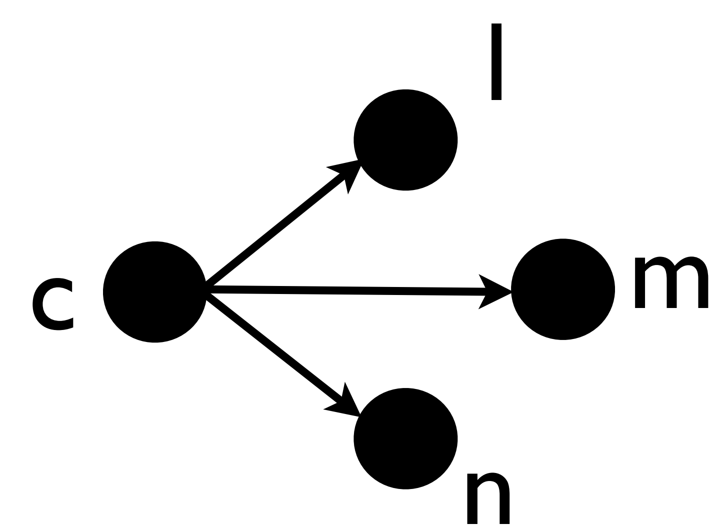
Which node should be processed along with k: the one with highest intersection of neighborhood with k

Node Reordering Algorithm : Intuition



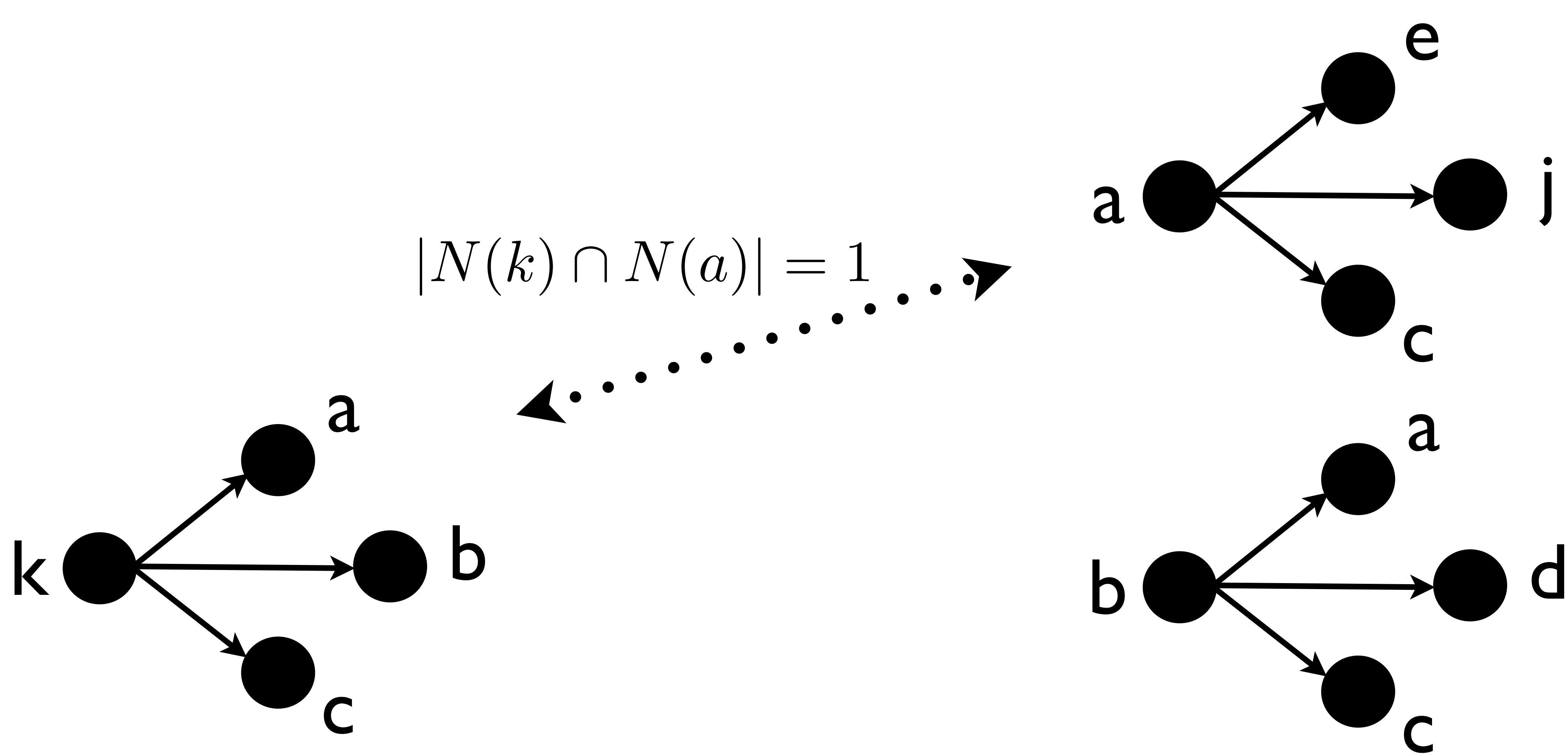


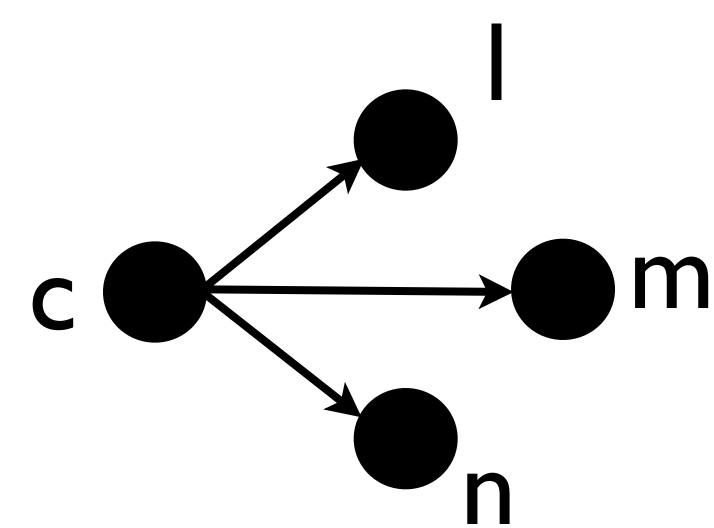




Which node should be processed along with k: the one with highest intersection of neighborhood with k

Node Reordering Algorithm : Intuition



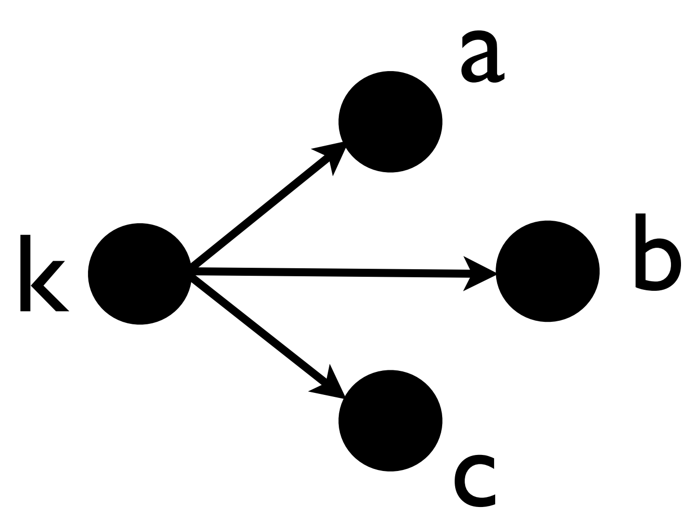


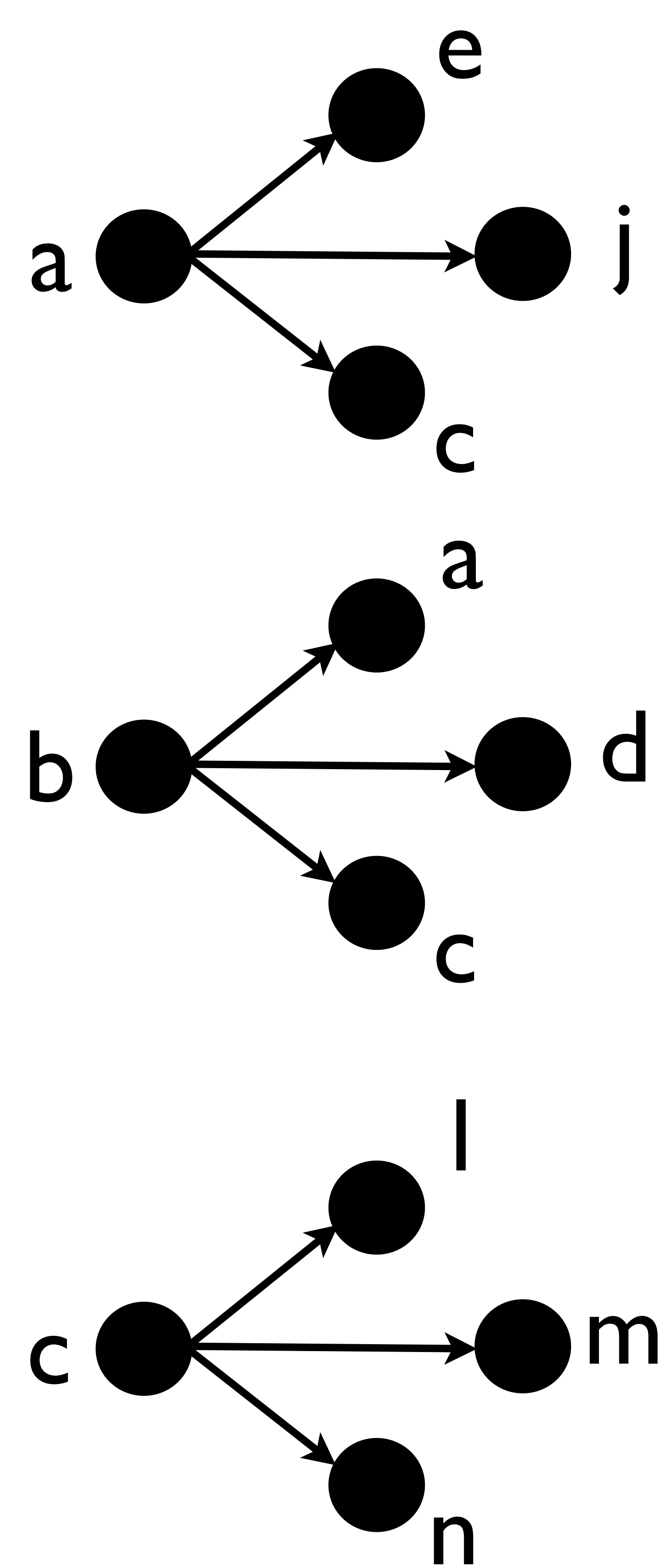
Which node should be processed along with k: the one with highest intersection of neighborhood with k

Node Reordering Algorithm : Intuition

Cardinality of Intersection

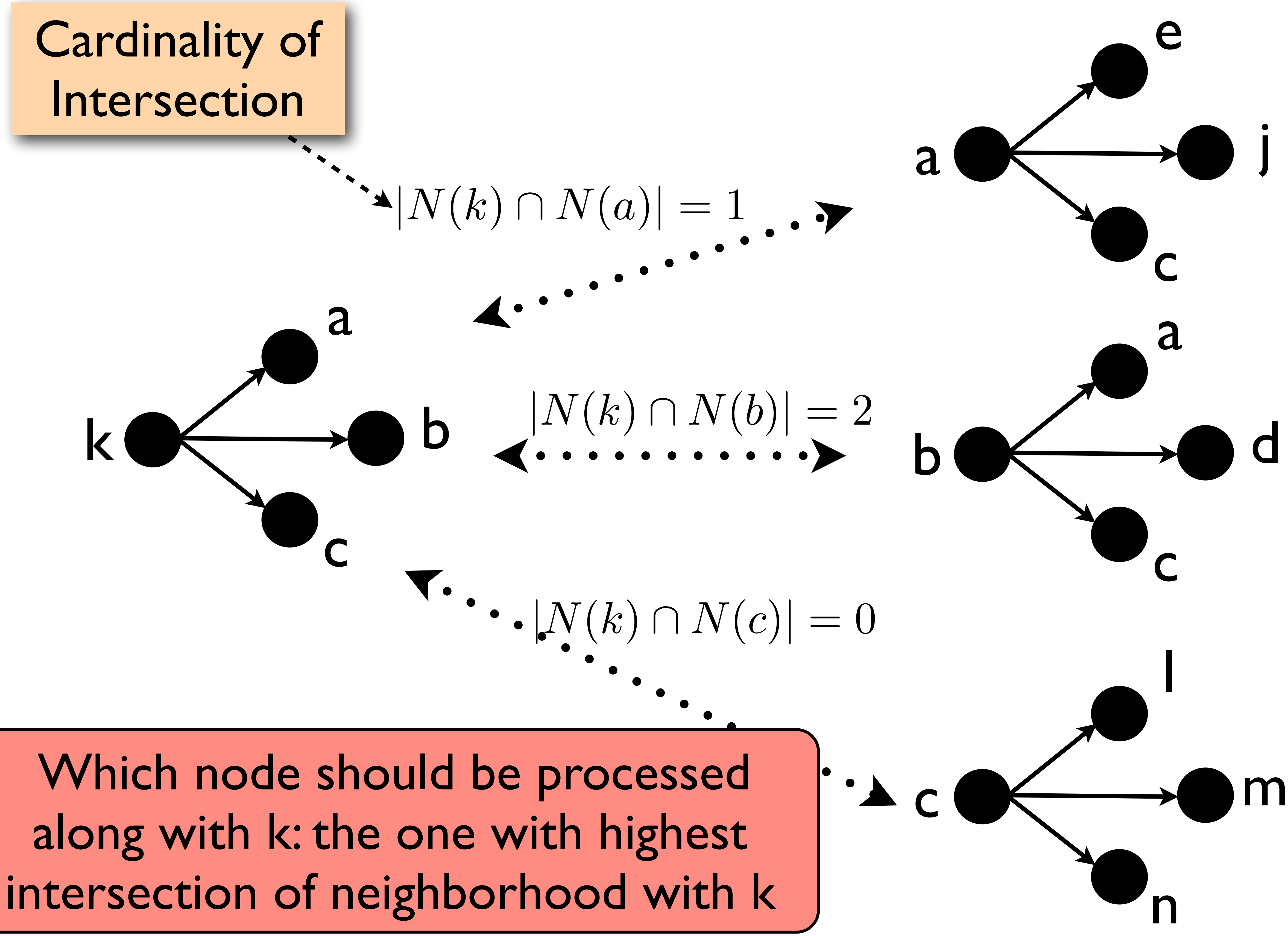
$$|N(k) \cap N(a)| = 1$$





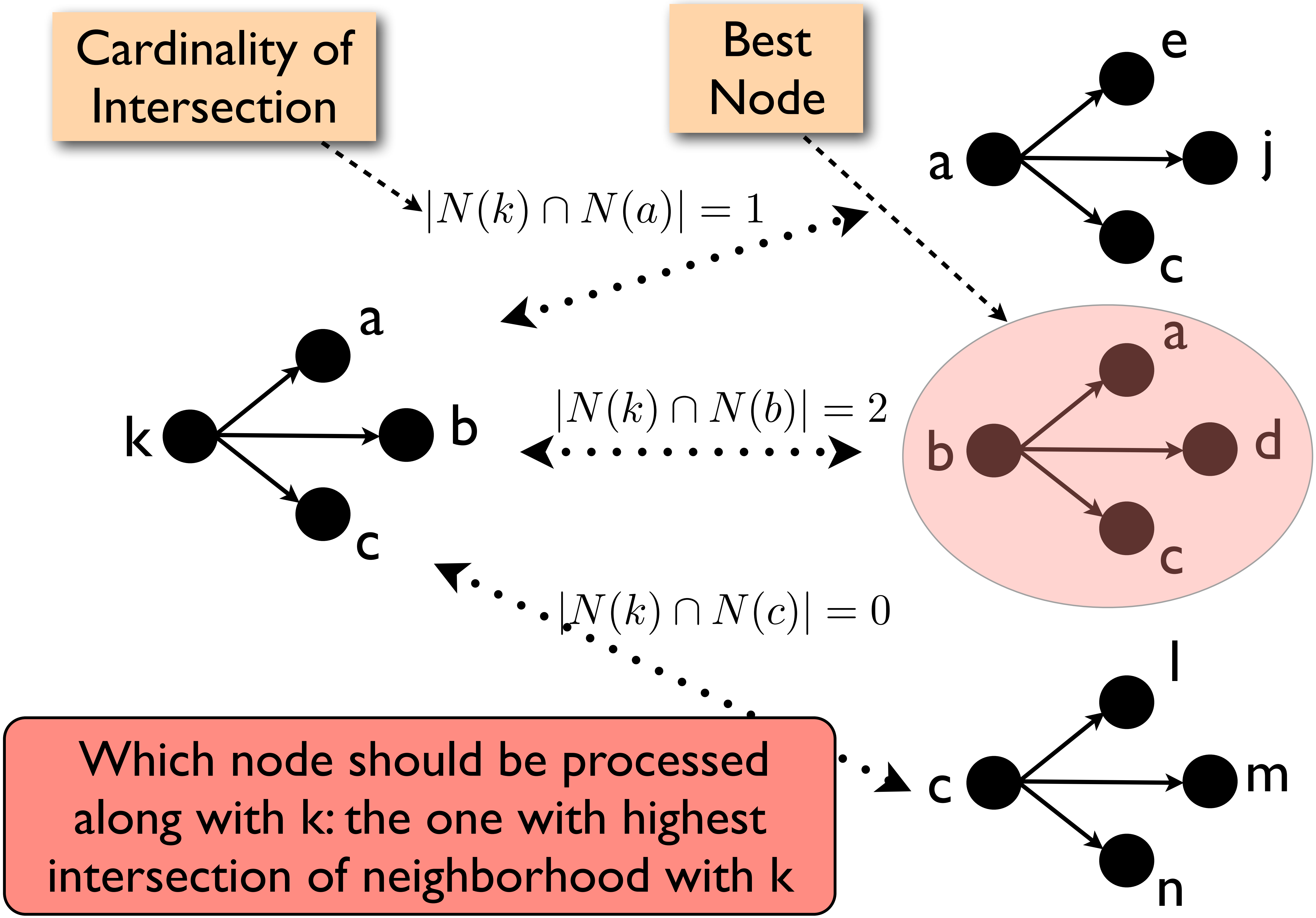
Which node should be processed along with k: the one with highest intersection of neighborhood with k

Node Reordering Algorithm : Intuition

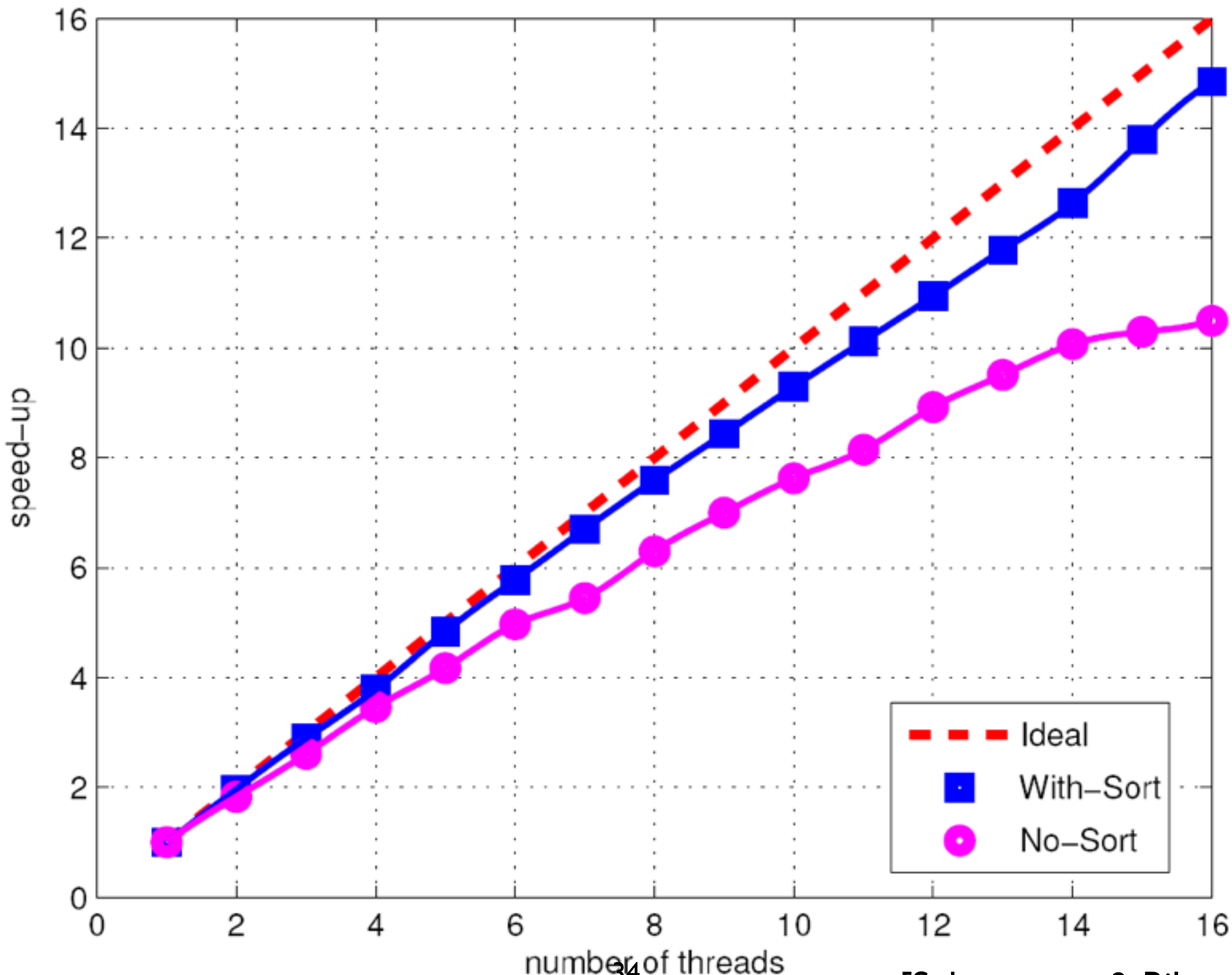


Which node should be processed along with k: the one with highest intersection of neighborhood with k

Node Reordering Algorithm : Intuition



Speed-up on SMP after Node Ordering

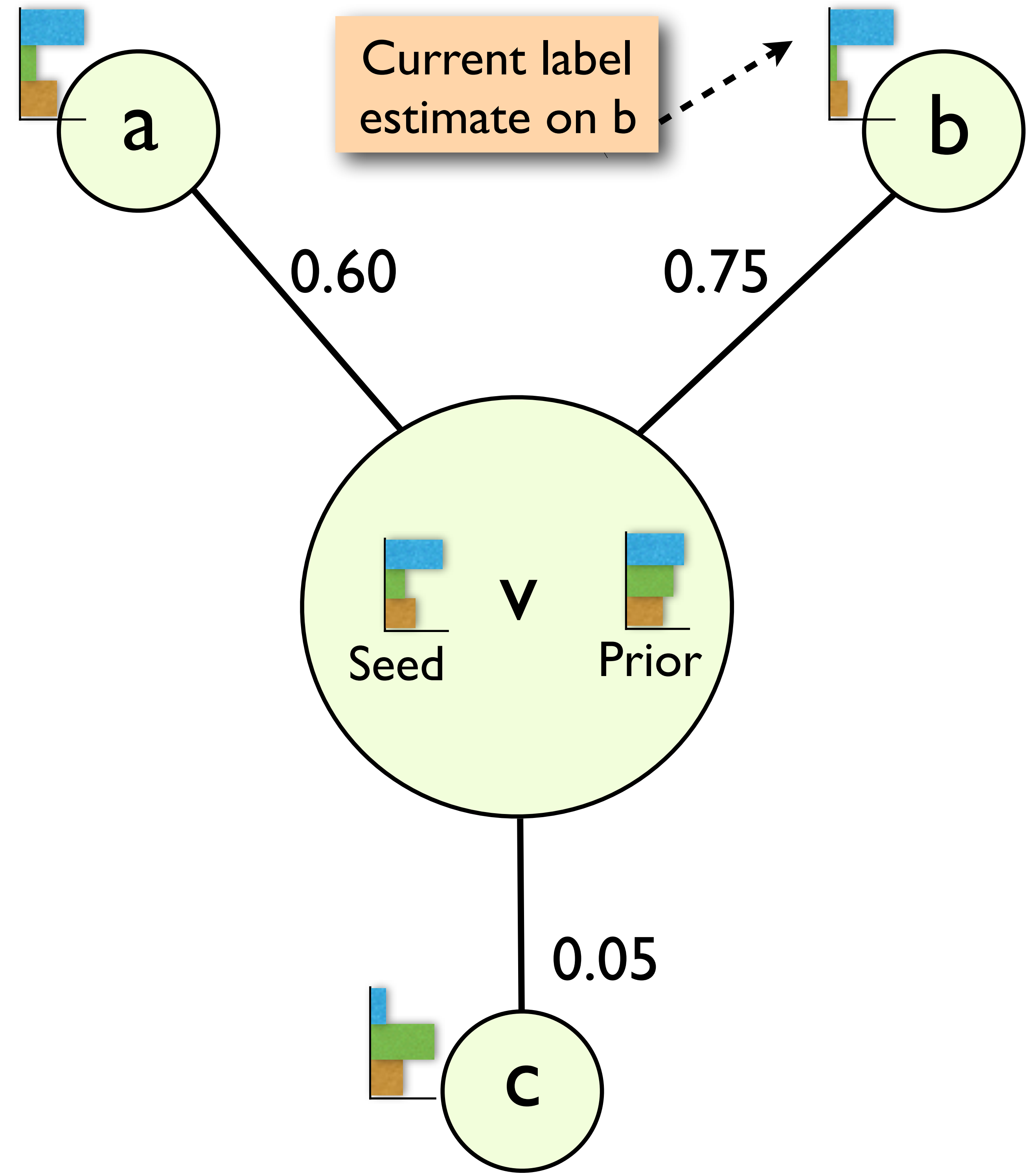


[Subramanya & Bilmes, JMLR, 2011]

Outline

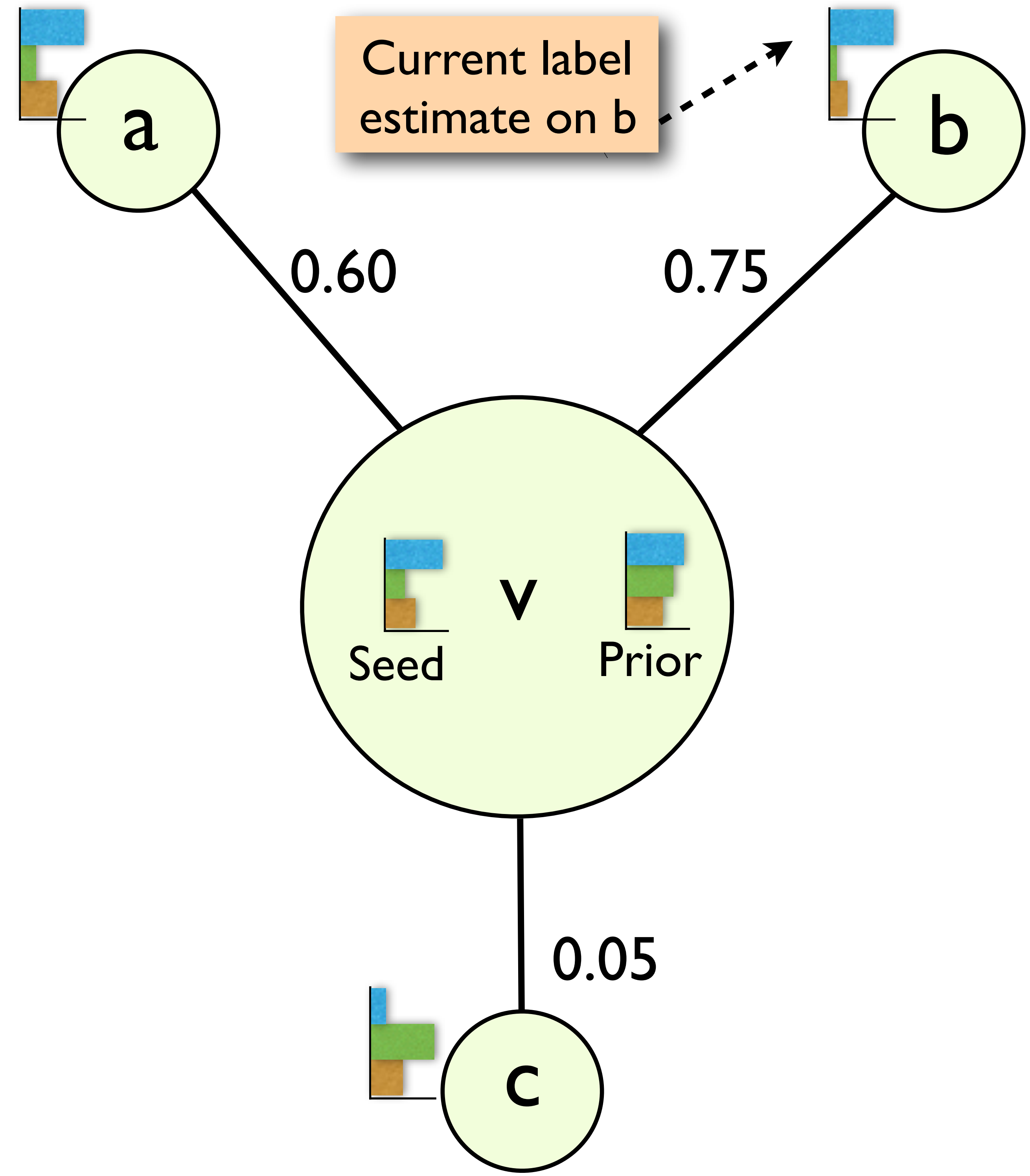
- Motivation
- Graph Construction
- Inference Methods
- Scalability
  - Scalability Issues
  - Node reordering
  - MapReduce Parallelization
- Applications
- Conclusion & Future Work

MapReduce Implementation of MAD



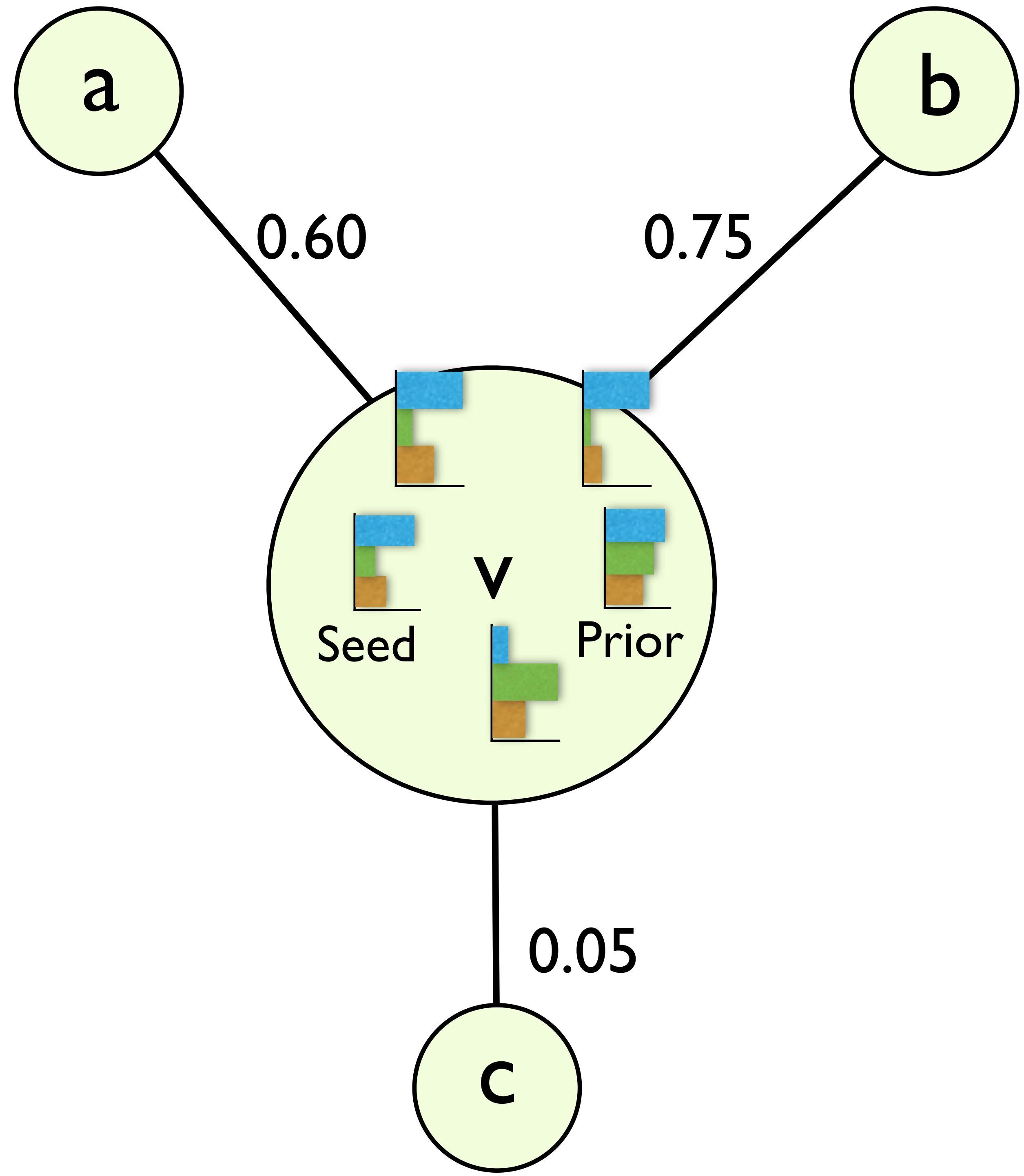
MapReduce Implementation of MAD

- Map
  - Each node send its current label assignments to its neighbors



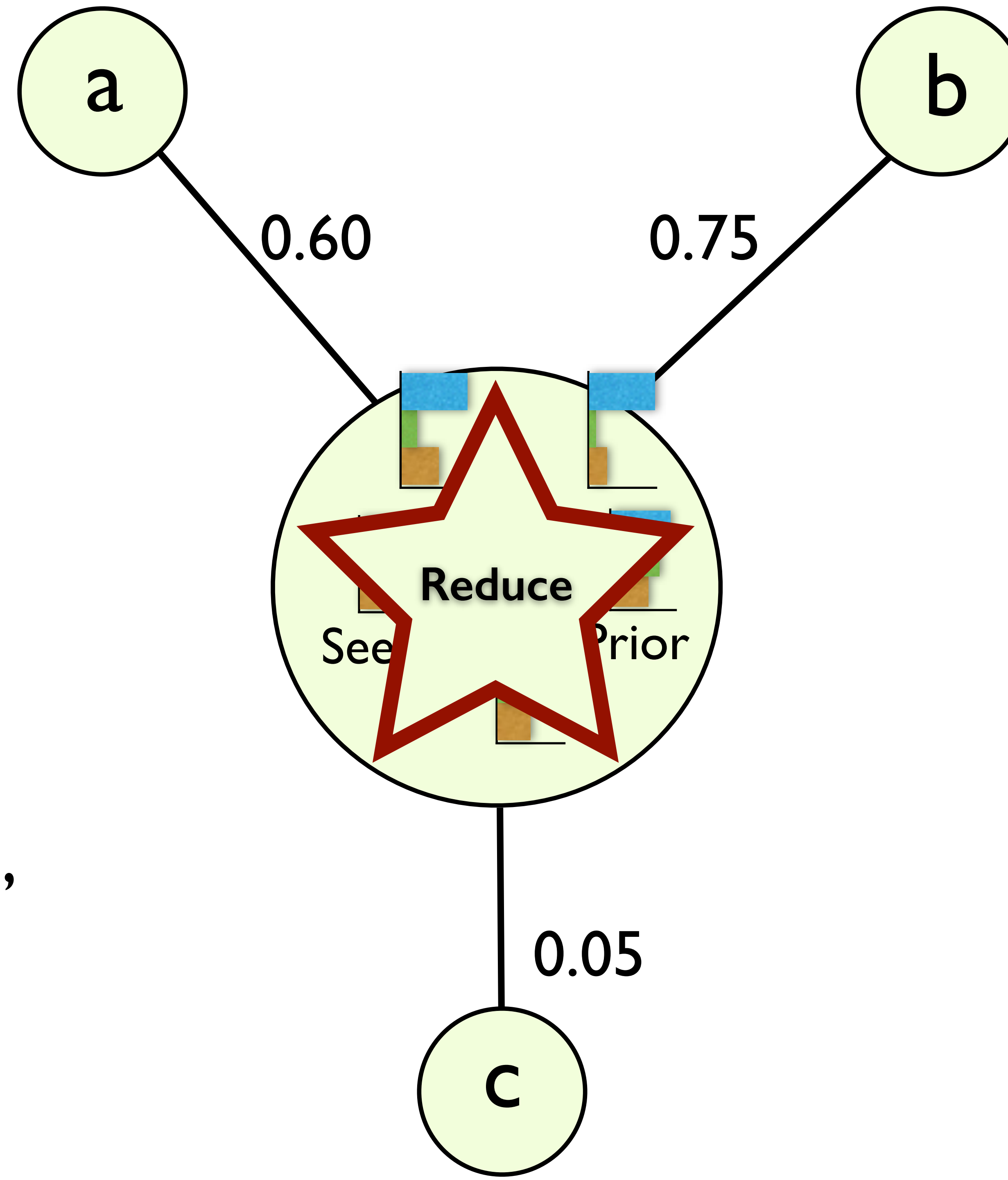
MapReduce Implementation of MAD

- Map
  - Each node send its current label assignments to its neighbors



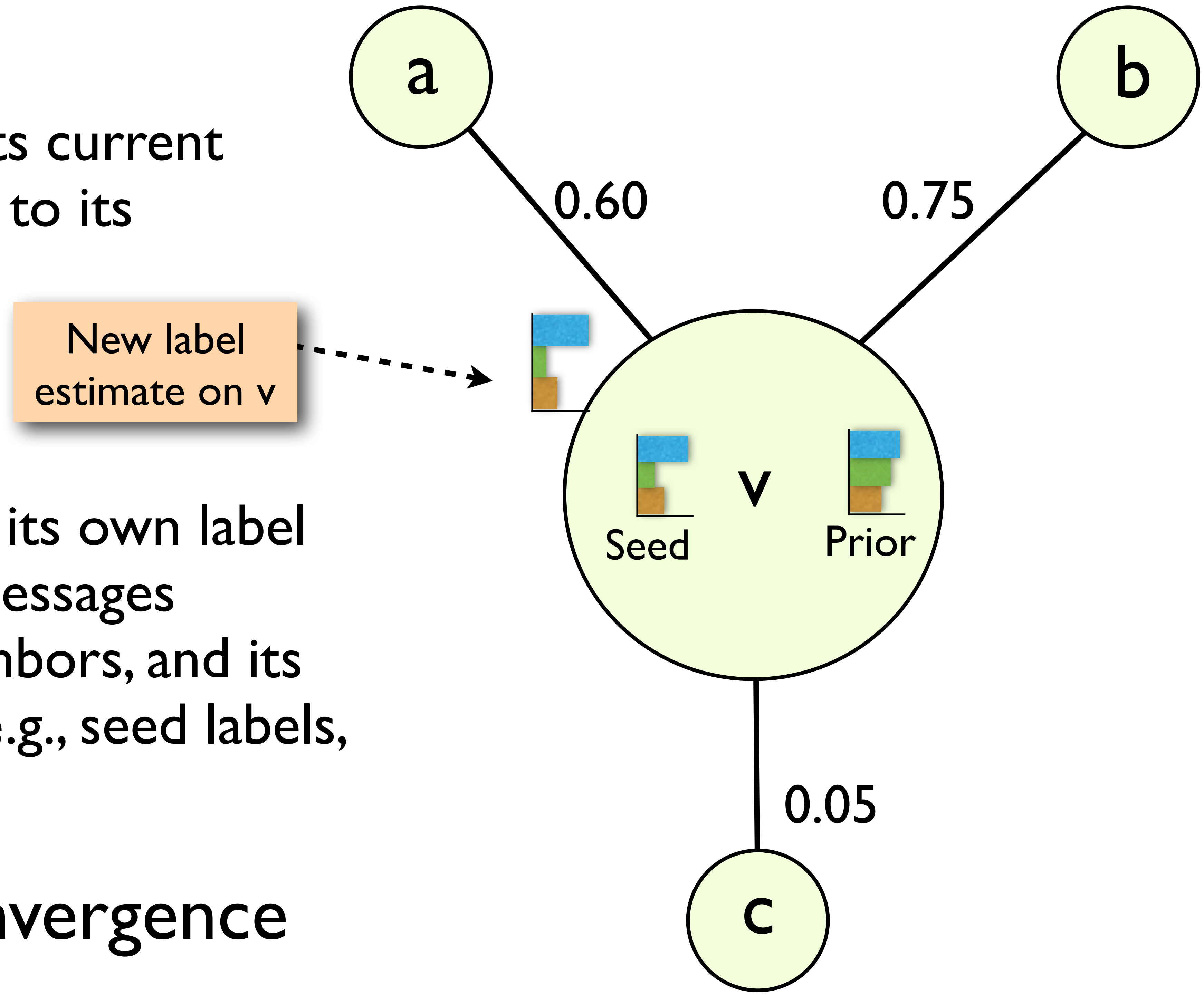
MapReduce Implementation of MAD

- Map
  - Each node send its current label assignments to its neighbors
- Reduce
  - Each node updates its own label assignment using messages received from neighbors, and its own information (e.g., seed labels, reg. penalties etc.)
- Repeat until convergence



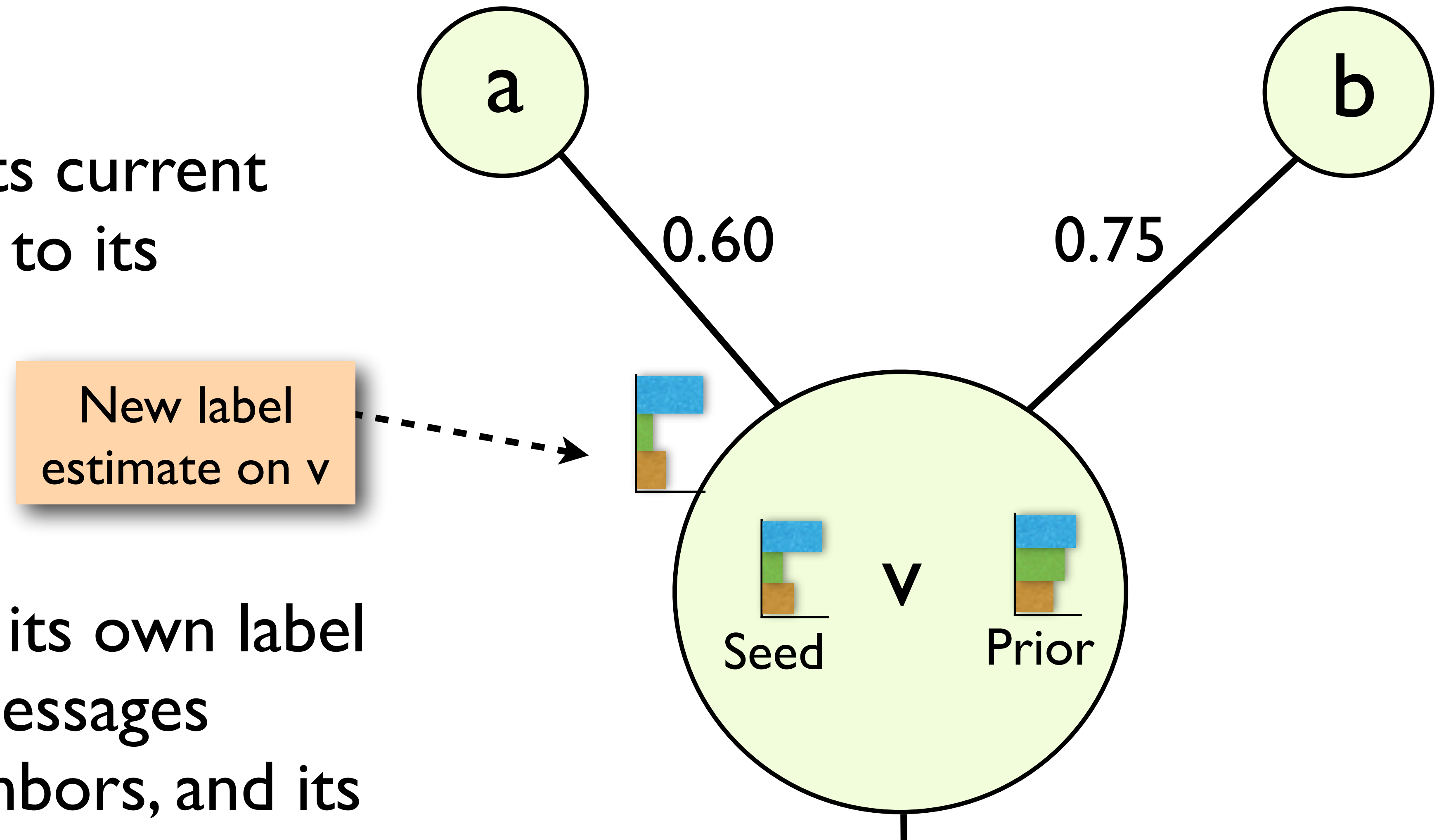
MapReduce Implementation of MAD

- Map
  - Each node send its current label assignments to its neighbors
- Reduce
  - Each node updates its own label assignment using messages received from neighbors, and its own information (e.g., seed labels, reg. penalties etc.)
- Repeat until convergence



MapReduce Implementation of MAD

- Map
  - Each node send its current label assignments to its neighbors
- Reduce
  - Each node updates its own label assignment using messages received from neighbors, and its own
  - reg.
- Repe



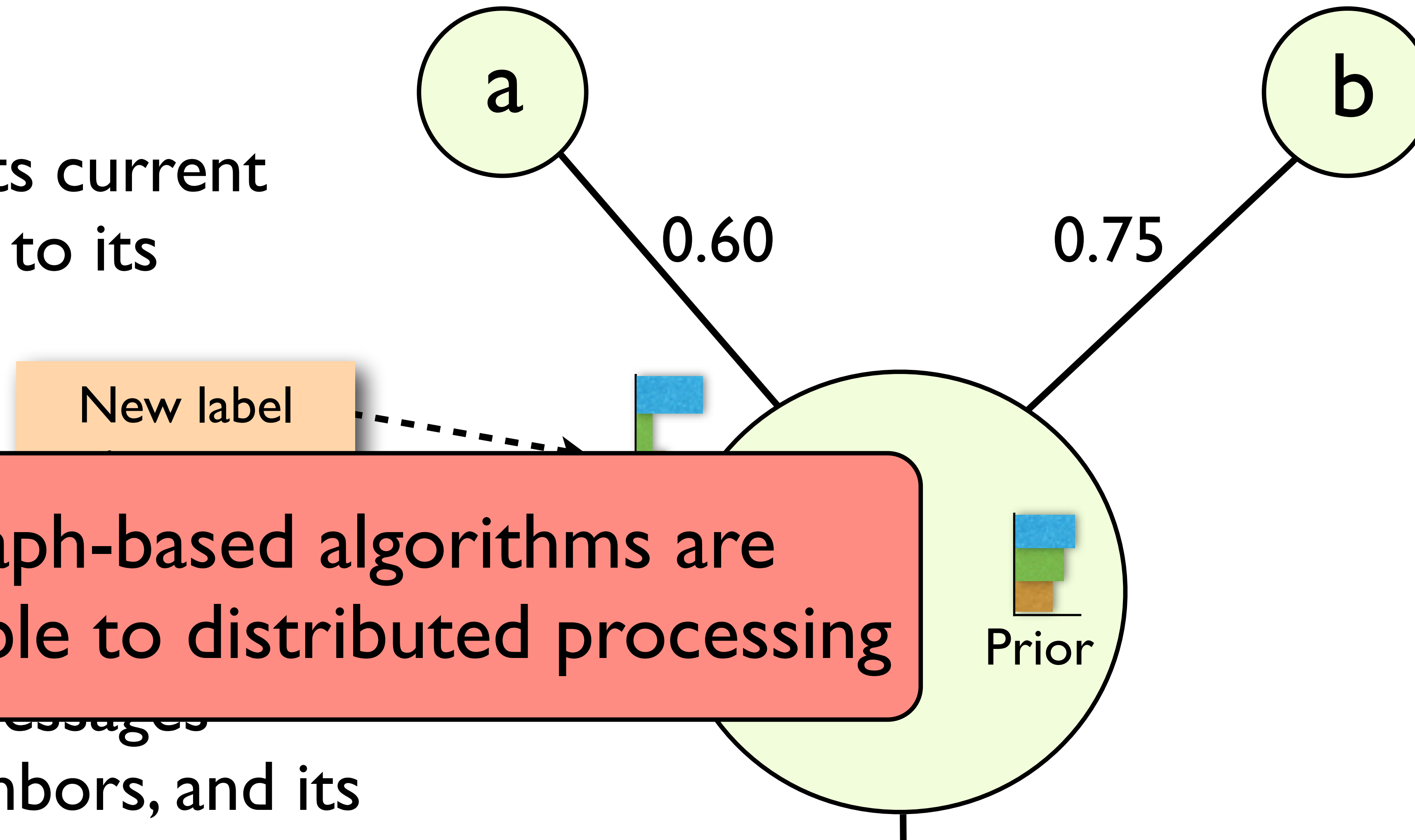
Code in Junto Label Propagation Toolkit

(includes Hadoop-based implementation)

https://github.com/parthatalukdar/junto

MapReduce Implementation of MAD

- Map
  - Each node send its current label assignments to its neighbors



- Reduce
  - Each node [illegible] assignment [illegible] received from neighbors, and its own [illegible] reg.
- Repe[illegible]

Graph-based algorithms are amenable to distributed processing

Code in Junto Label Propagation Toolkit
(includes Hadoop-based implementation)
https://github.com/parthatalukdar/junto

When to use Graph-based SSL and which method?

When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
  - or, when the data is expected to lie on a manifold

When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
  - or, when the data is expected to lie on a manifold
- MAD, Quadratic Criteria (QC)
  - when labels are not mutually exclusive
  - MADDL: when label similarities are known

When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
  - or, when the data is expected to lie on a manifold
- MAD, Quadratic Criteria (QC)
  - when labels are not mutually exclusive
  - MADDL: when label similarities are known
- Measure Propagation (MP)
  - for probabilistic interpretation

When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
  - or, when the data is expected to lie on a manifold
- MAD, Quadratic Criteria (QC)
  - when labels are not mutually exclusive
  - MADDL: when label similarities are known
- Measure Propagation (MP)
  - for probabilistic interpretation
- Manifold Regularization
  - for generalization to unseen data (induction)

Graph-based SSL: Summary

Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data

Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key

Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key
- Scalable: Node Reordering and MapReduce

Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key
- Scalable: Node Reordering and MapReduce
- Can handle labeled as well as unlabeled data

Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key
- Scalable: Node Reordering and MapReduce
- Can handle labeled as well as unlabeled data
- Can handle multi class, multi label settings

Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key
- Scalable: Node Reordering and MapReduce
- Can handle labeled as well as unlabeled data
- Can handle multi class, multi label settings
- Effective in practice

Open Challenges

Open Challenges

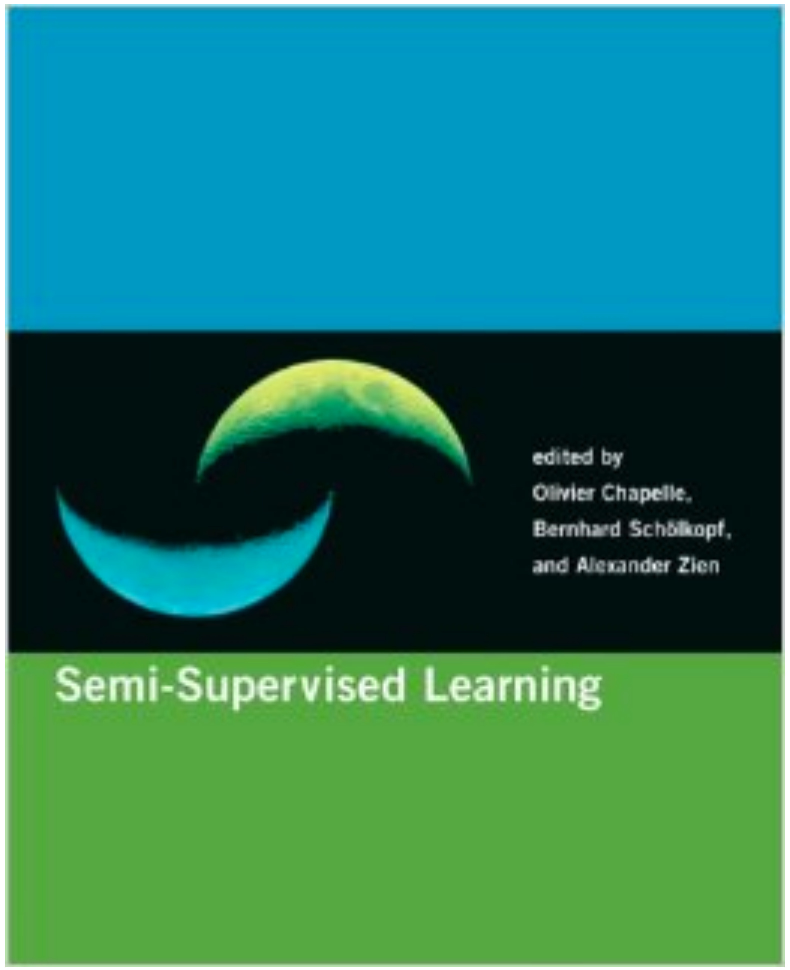
- Graph-based SSL for Structured Prediction
  - Algorithms: Combining Inductive and graph-based methods
  - Applications: Constituency and dependency parsing, Coreference

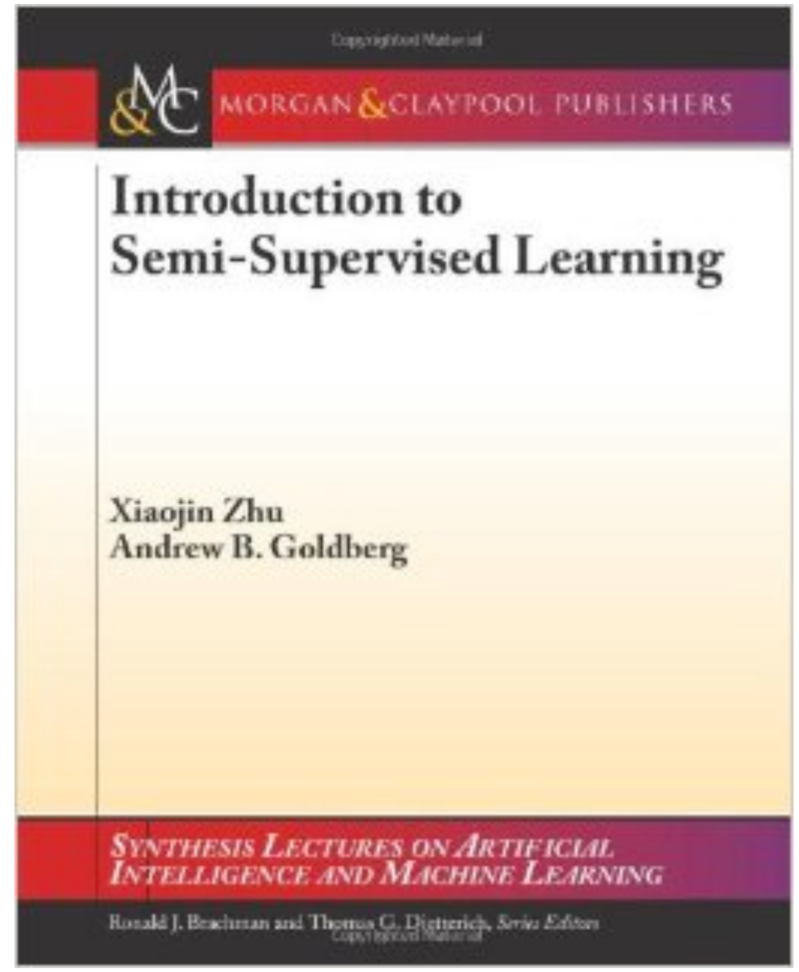
Open Challenges

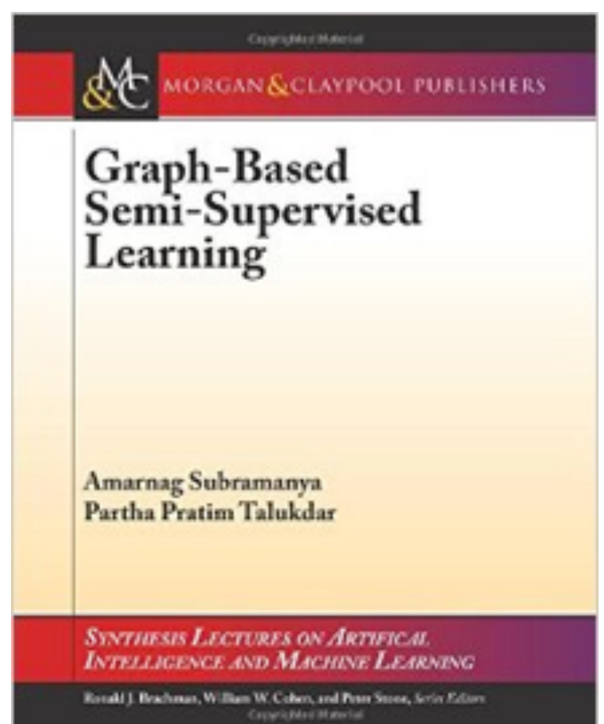
- Graph-based SSL for Structured Prediction
  - Algorithms: Combining Inductive and graph-based methods
  - Applications: Constituency and dependency parsing, Coreference
- Scalable graph construction, especially with multi-modal data

Open Challenges

- Graph-based SSL for Structured Prediction
  - Algorithms: Combining Inductive and graph-based methods
  - Applications: Constituency and dependency parsing, Coreference
- Scalable graph construction, especially with multi-modal data
- Extensions with other loss functions, sparsity, etc.


edited by
Olivier Chapelle,
Bernhard Schölkopf,
and Alexander Zien
Semi-Supervised Learning


MORGAN & CLAYPOOL PUBLISHERS
Introduction to
Semi-Supervised Learning
Xiaojin Zhu
Andrew B. Goldberg
SYNTHESIS LECTURES ON ARTIFICIAL
INTELLIGENCE AND MACHINE LEARNING
Ronald J. Brachman and Thomas G. Dietterich, Series Editors


MORGAN & CLAYPOOL PUBLISHERS
Graph-Based
Semi-Supervised
Learning
Amarnag Subramanya
Partha Pratim Talukdar
SYNTHESIS LECTURES ON ARTIFICAL
INTELLIGENCE AND MACHINE LEARNING

References (I)

[1] A. Alexandrescu and K. Kirchhoff. Data-driven graph construction for semi-supervised graph-based learning in nlp. In NAACL HLT, 2007.
[2] Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi-supervised learn- ing for structured variables. NIPS, 2006.
[3] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In WWW, 2008.
[4] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. J. Mach. Learn. Res., 3:1183–1208, 2003.
[5] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research, 7:2399–2434, 2006.
[6] Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. Semi-supervised learning, 2006.
[7] T. Berg-Kirkpatrick, A. Bouchard-Coˆt´e, J. DeNero, and D. Klein. Painless unsupervised learning with features. In HLT-NAACL, 2010.
[8] J. Bilmes and A. Subramanya. Scaling up Machine Learning: Parallel and Distributed Approaches, chapter Parallel Graph-Based Semi-Supervised Learning. 2011.
[9] S. Blair-goldensohn, T. Neylon, K. Hannan, G. A. Reis, R. Mcdonald, and J. Reynar. Building a sentiment summarizer for local service reviews. In In NLP in the Information Explosion Era, 2008.
[10] M. Cafarella, A. Halevy, D. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. VLDB, 2008.
[11] O. Chapelle, B. Scho¨lkopf, A. Zien, et al. Semi-supervised learning. MIT press Cambridge, MA:, 2006.
[12] Y. Choi and C. Cardie. Adapting a polarity lexicon using integer linear program- ming for domain specific sentiment classification. In EMNLP, 2009.
[13] S. Daitch, J. Kelner, and D. Spielman. Fitting a graph to vector data. In ICML, 2009.
[14] D. Das and S. Petrov. Unsupervised part-of-speech tagging with bilingual graph- based projections. In ACL, 2011.
[15] D. Das, N. Schneider, D. Chen, and N. A. Smith. Probabilistic frame-semantic parsing. In NAACL-HLT, 2010.
[16] D. Das and N. Smith. Graph-based lexicon expansion with sparsity-inducing penalties. NAACL-HLT, 2012.
[17] D. Das and N. A. Smith. Semi-supervised frame-semantic parsing for unknown predicates. In ACL, 2011.
[18] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In ICML, 2007.
[19] O. Delalleau, Y. Bengio, and N. L. Roux. Efficient non-parametric function induction in semi-supervised learning. In AISTATS, 2005.
[20] P. Dhillon, P. Talukdar, and K. Crammer. Inference-driven metric learning for graph construction. Technical report, MS-CIS-10-18, University of Pennsylvania, 2010.

References (II)

[21] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In CIKM, 1998.
[22] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. ACM Transaction on Mathematical Software, 3, 1977.
[23] J. Garcke and M. Griebel. Data mining with sparse grids using simplicial basis functions. In KDD, 2001.
[24] A. Goldberg and X. Zhu. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, 2006.
[25] A. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. AISTATS, 2007.
[26] M. Hu and B. Liu. Mining and summarizing customer reviews. In KDD, 2004.
[27] T. Jebara, J. Wang, and S. Chang. Graph construction and b-matching for semi-supervised learning. In ICML, 2009.
[28] T. Joachims. Transductive inference for text classification using support vector machines. In ICML, 1999.
[29] T. Joachims. Transductive learning via spectral graph partitioning. In ICML, 2003.
[30] M. Karlen, J. Weston, A. Erkan, and R. Collobert. Large scale manifold transduction. In ICML, 2008.
[31] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In Proceedings of the 20th International conference on Computational Linguistics, 2004.
[32] F. Kschischang, B. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. Information Theory, IEEE Transactions on, 47(2):498–519, 2001.
[33] K. Lerman, S. Blair-Goldensohn, and R. McDonald. Sentiment summarization: evaluating and learning user preferences. In EACL, 2009.
[34] D.Lewisetal.Reuters-21578.http://www.daviddlewis.com/resources/testcollections/reuters21578, 1987.
[35] J. Malkin, A. Subramanya, and J. Bilmes. On the semi-supervised learning of multi-layered perceptrons. In InterSpeech, 2009.
[36] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In EMNLP, 2002. [37] D. Rao and D. Ravichandran. Semi-supervised polarity lexicon induction. In EACL, 2009.
[38] A. Subramanya and J. Bilmes. Soft-supervised learning for text classification. In EMNLP, 2008.
[39] A. Subramanya and J. Bilmes. Entropic graph regularization in non-parametric semi-supervised classification. NIPS, 2009.
[40] A. Subramanya and J. Bilmes. Semi-supervised learning with measure propagation. JMLR, 2011.

References (III)

[41] A. Subramanya, S. Petrov, and F. Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In EMNLP, 2010.
[42] P. Talukdar. Topics in graph construction for semi-supervised learning. Technical report, MS-CIS-09-13, University of Pennsylvania, 2009.
[43] P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. ECML, 2009.
[44] P. Talukdar and F. Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In ACL, 2010.
[45] P. Talukdar, J. Reisinger, M. Pa¸sca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly-supervised acquisition of labeled class instances using graph random walks. In EMNLP, 2008.
[46] B. Van Durme and M. Pasca. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In AAAI, 2008.
[47] L. Velikovich, S. Blair-Goldensohn, K. Hannan, and R. McDonald. The viability of web-derived polarity lexicons. In HLT-NAACL, 2010.
[48] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In ICML, 2006.
[49] J. Wang, T. Jebara, and S. Chang. Graph transduction via alternating minimization. In ICML, 2008.
[50] R. Wang and W. Cohen. Language-independent set expansion of named entities using the web. In ICDM, 2007.
[51] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research, 10:207–244, 2009.
[52] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase- level sentiment analysis. In HLT-EMNLP, 2005.
[53] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scho¨lkopf. Learning with local and global consistency. NIPS, 2004.
[54] D. Zhou, J. Huang, and B. Scho¨lkopf. Learning from labeled and un- labeled data on a directed graph. In ICML, 2005.
[55] D. Zhou, B. Scho¨lkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In NIPS, 2005.
[56] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University, 2002.
[57] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.
[58] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In ICML, 2003.
[59] X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph- based methods for inductive and scalable semi-supervised learning. In ICML, 2005.

Thanks!

Web: http://graph-ssl.wikidot.com/