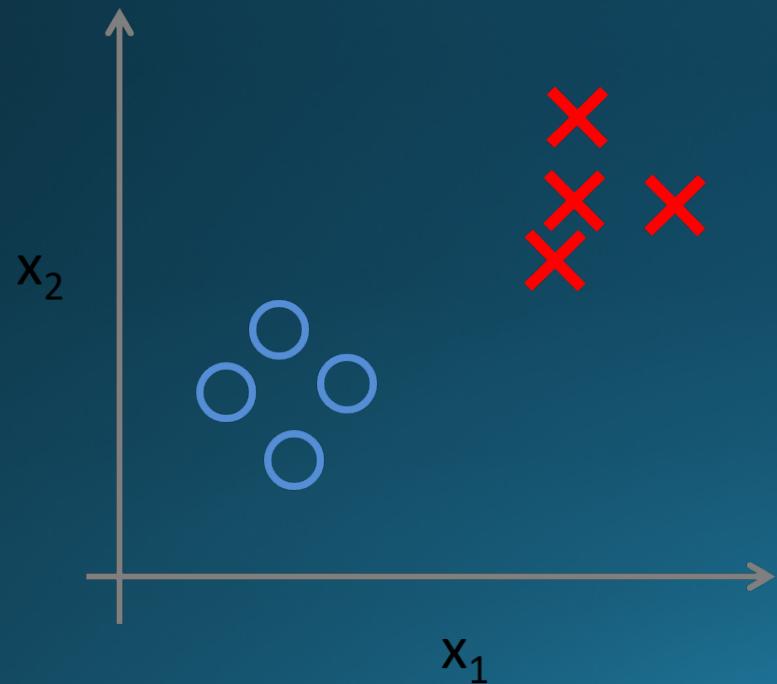


CSCI 4360/6360 Data Science II

# Semi-Supervised Learning

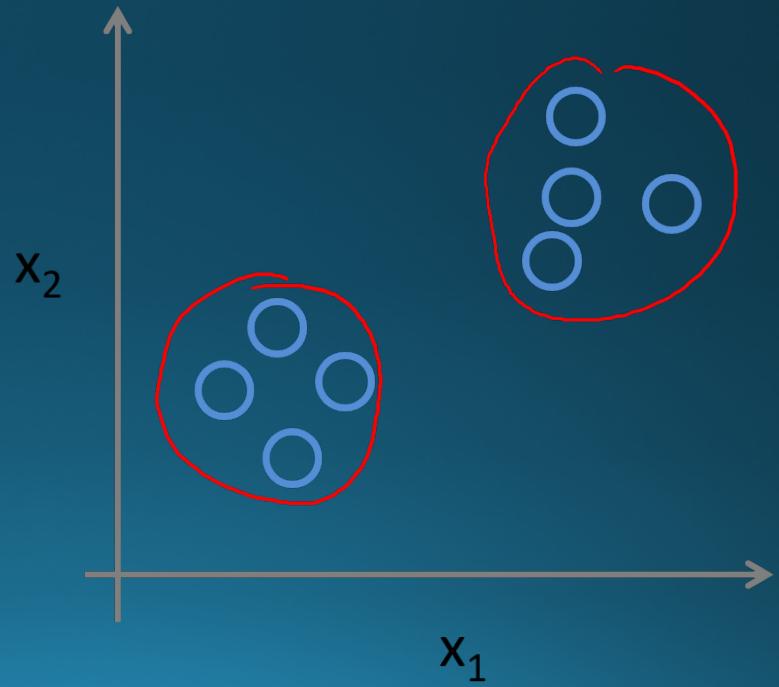
# Supervised learning

## Supervised Learning



# Unsupervised learning

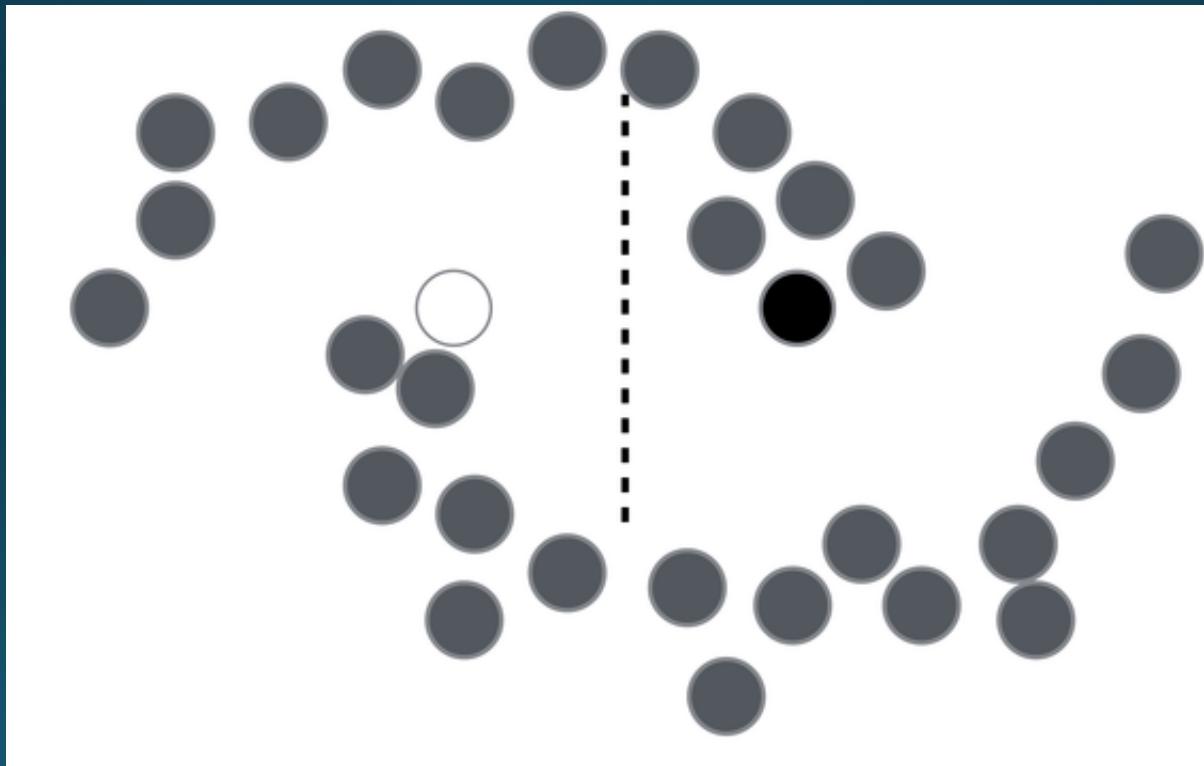
Unsupervised Learning



# Semi-supervised learning

- Basically a hybrid!
- Given:
  - A pool of labeled examples L
  - A (usually larger) pool of unlabeled examples U
- **Can you improve accuracy somehow using U?**

# Semi-supervised Learning



# Semi-Supervised Bootstrapped Learning/Self-training

Extract cities:

Paris  
Pittsburgh  
Seattle  
Cupertino

San Francisco  
Austin  
denial

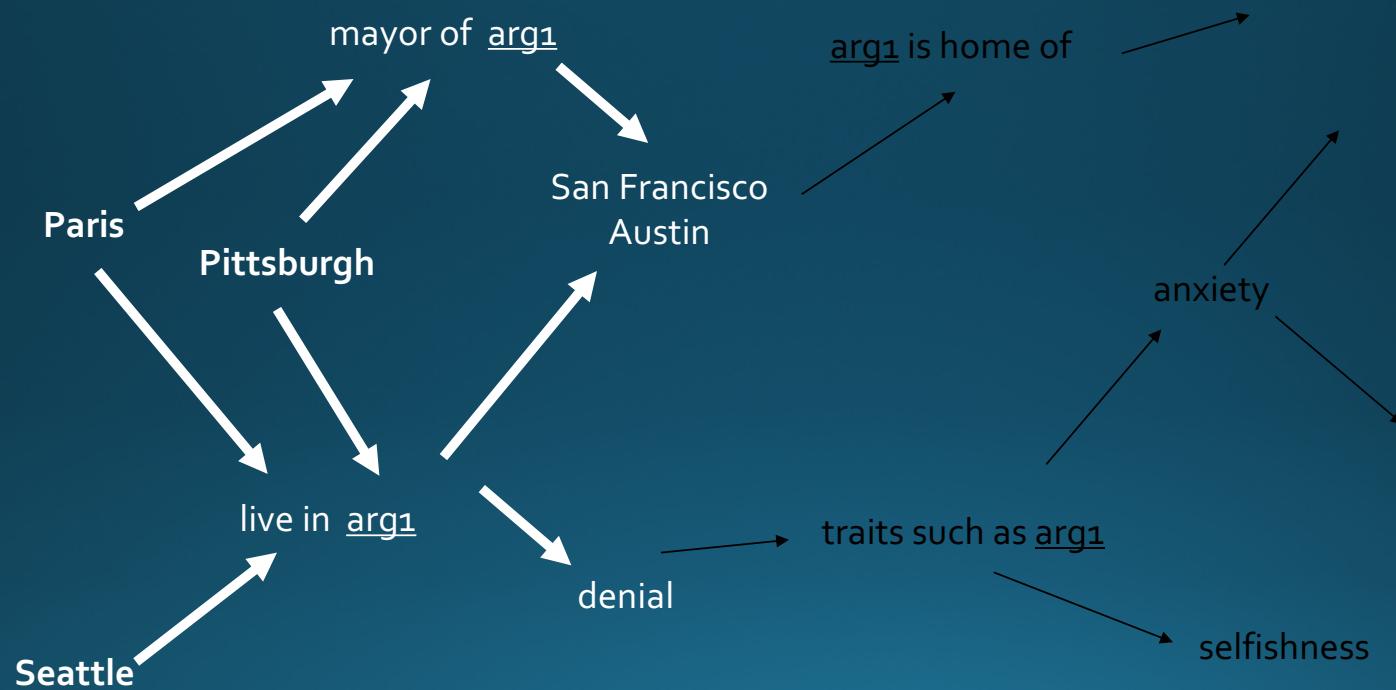
anxiety  
selfishness  
Berlin

mayor of arg1  
live in arg1

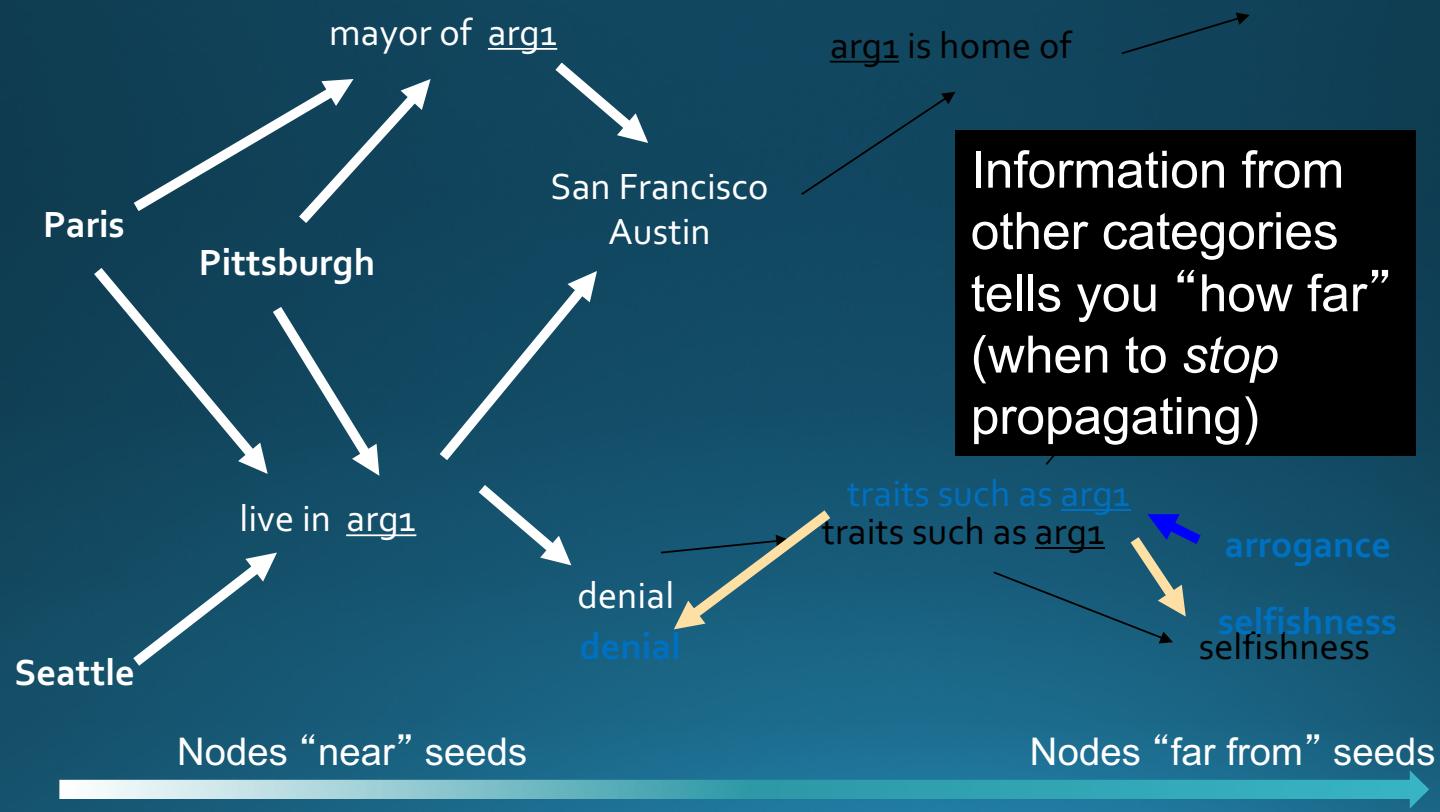
arg1 is home of  
traits such as arg1



# Semi-Supervised Bootstrapped Learning via Label Propagation

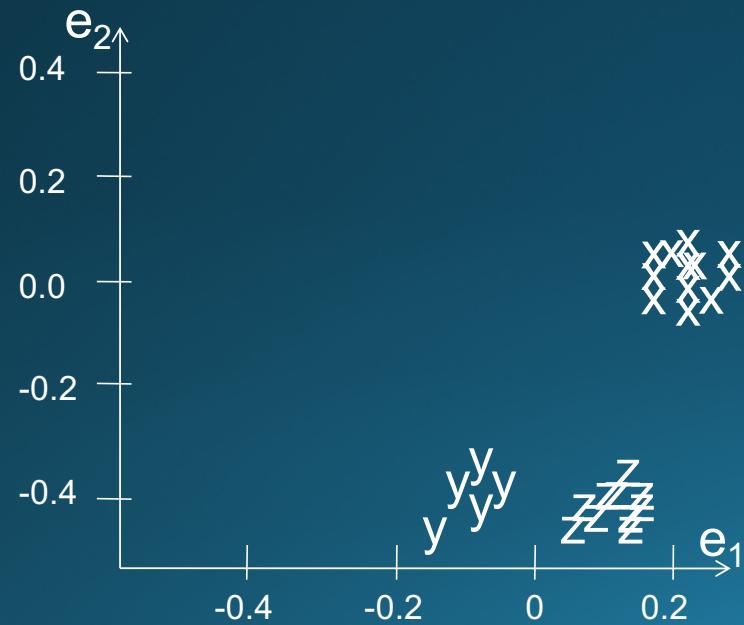


# Semi-Supervised Bootstrapped Learning via Label Propagation

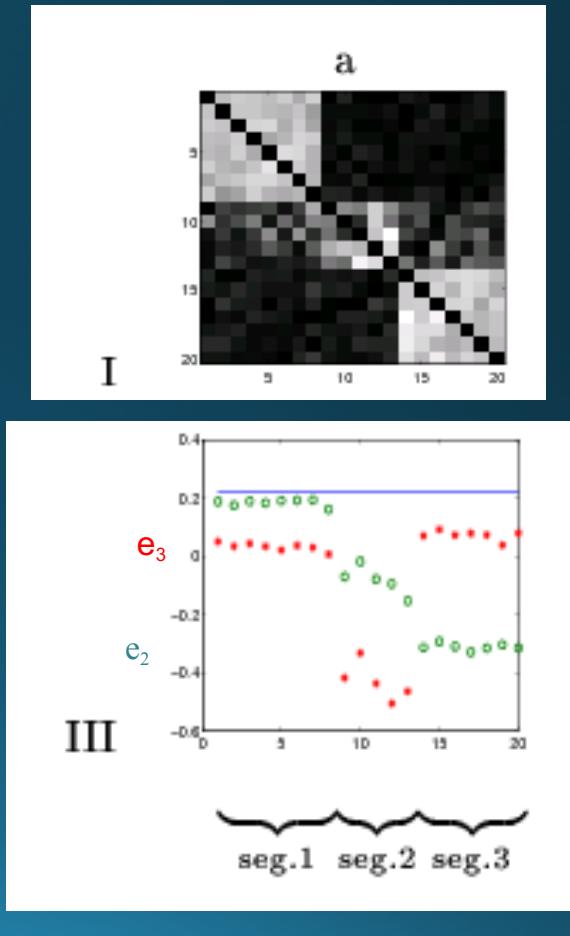


# Spectral Clustering

- Graph = Matrix
  - $W^*v_1 = v_2$  “propogates weights from neighbors”

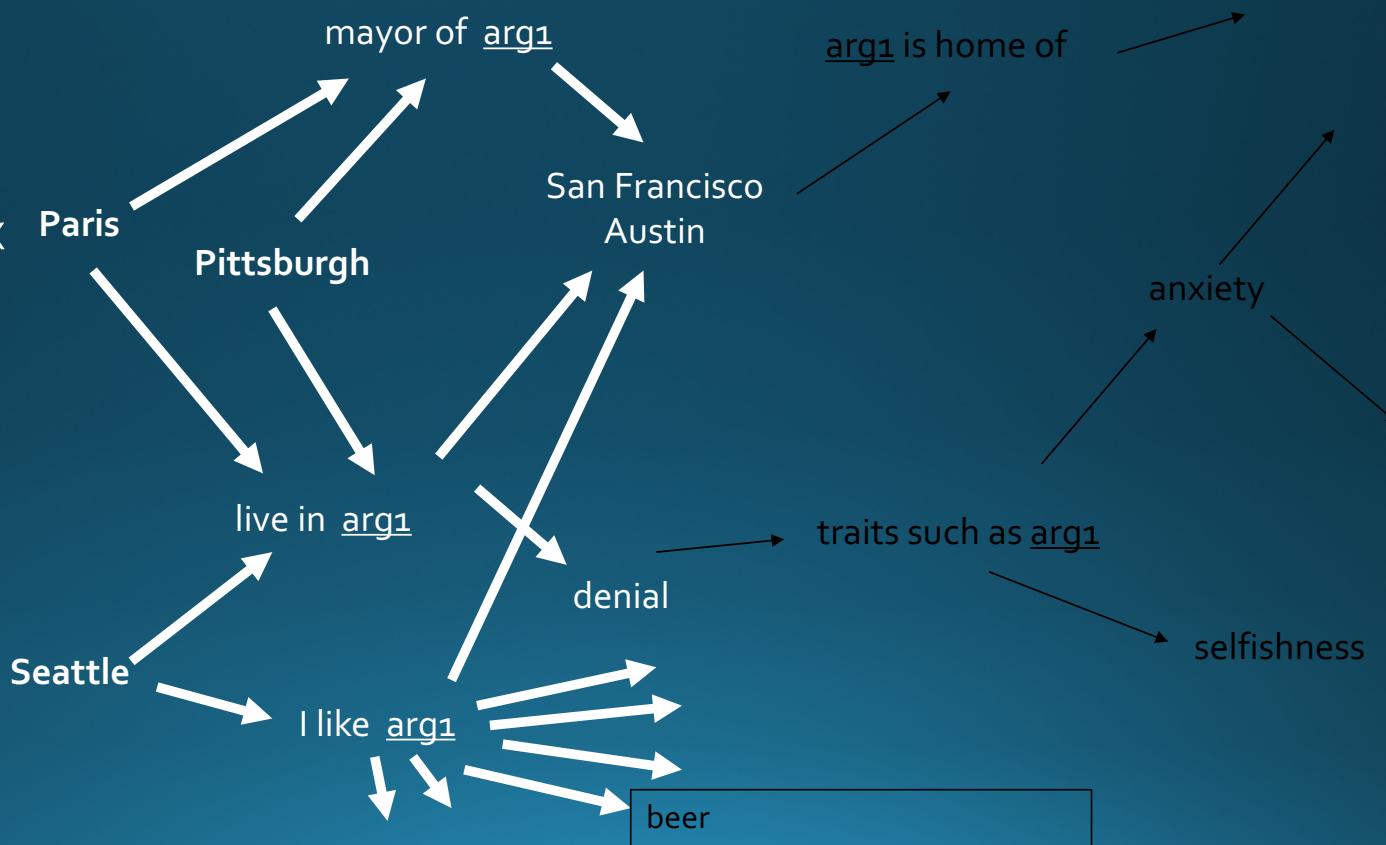


[Shi & Meila, 2002]



# Semi-Supervised Learning as Label Propagation on a Graph

- Propagate label to “nearby” nodes
  - X is “near” Y if there is a high probability of reaching Y from X
- Propagation methods
  - Personalized PageRank
  - Random walk
- Rewards multiple paths
- Penalizes longer and “high fanout” paths



## Semi-Supervised Classification of Network Data Using Very Few Labels

Frank Lin

Carnegie Mellon University, Pittsburgh, Pennsylvania  
Email: [frank@cs.cmu.edu](mailto:frank@cs.cmu.edu)

William W. Cohen

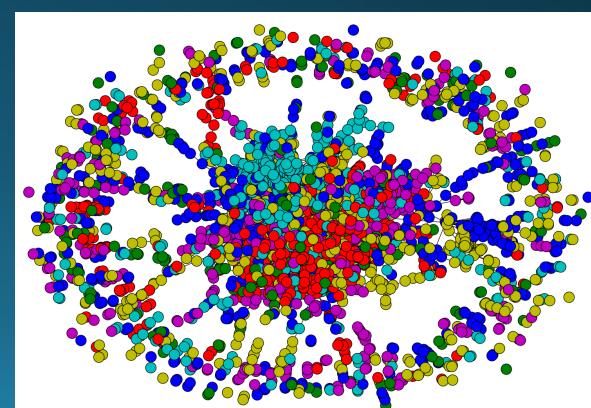
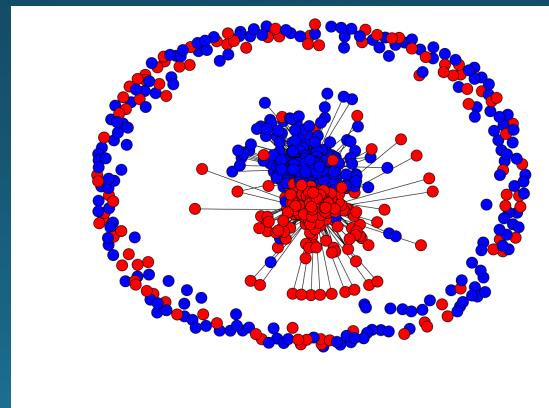
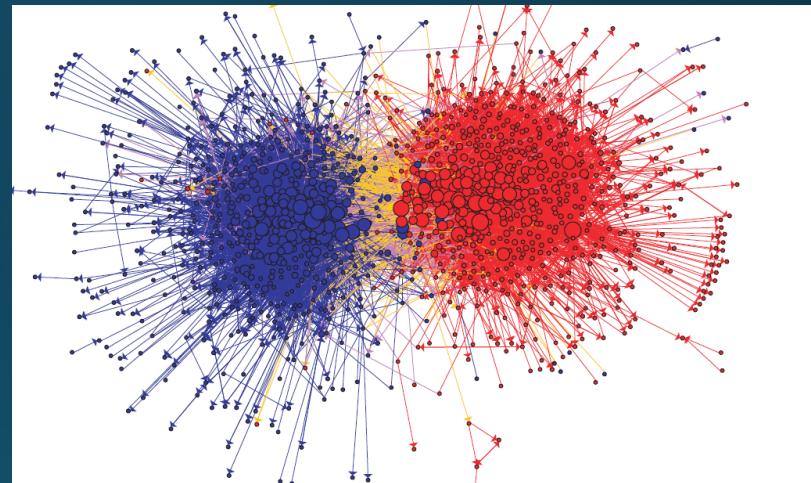
Carnegie Mellon University, Pittsburgh, Pennsylvania  
Email: [wcohen@cs.cmu.edu](mailto:wcohen@cs.cmu.edu)

ASONAM-2010 (Advances in Social Networks Analysis and Mining)

# Network Datasets with Known Classes

- UBMCBlog
- AGBlog
- MSPBlog
- Cora
- Citeseer

	Nodes	Edges	Density
UMBCBlog	404	2725	0.01670
AGBlog	1222	19021	0.01274
MSPBlog	1031	9316	0.00876
Cora	2485	5209	0.00084
CiteSeer	2110	3757	0.00084



# MultiRankWalk

- Seed Selection
  - Order by PageRank or degree, or even randomly
  - Traverse list until you have  $k$  examples/class

$$\vec{r} = (1 - d)\vec{u} + dW\vec{r}$$

**Given:** A graph  $G = (V, E)$ , corresponding to nodes in  $G$  are instances  $X$ , composed of unlabeled instances  $X^U$  and labeled instances  $X^L$  with corresponding labels  $Y^L$ , and a damping factor  $d$ .  
**Returns:** Labels  $Y^U$  for unlabeled nodes  $X^U$ .

**For each class  $c$**

- 1) Set  $\mathbf{u}_i \leftarrow 1, \forall Y_i^L = c$
- 2) Normalize  $\mathbf{u}$  such that  $\|\mathbf{u}\|_1 = 1$
- 3) Set  $R_c \leftarrow RandomWalk(G, \mathbf{u}, d)$

**For each instance  $i$**

- Set  $X_i^U \leftarrow argmax_c(R_{ci})$

Fig. 1. The MultiRankWalk algorithm.

# Comparison: wvRN

- One definition [MacSkassy & Provost, JMLR 2007]:...

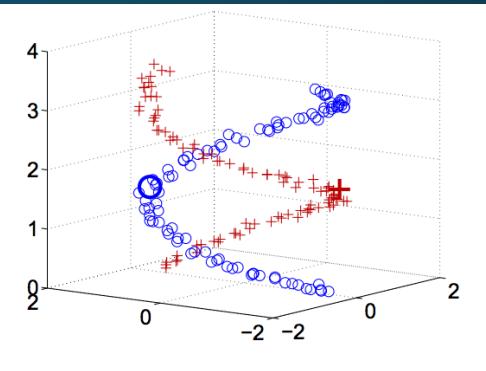
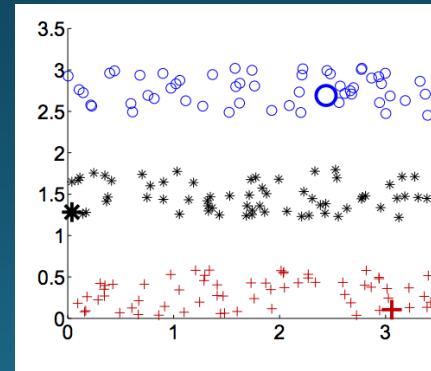
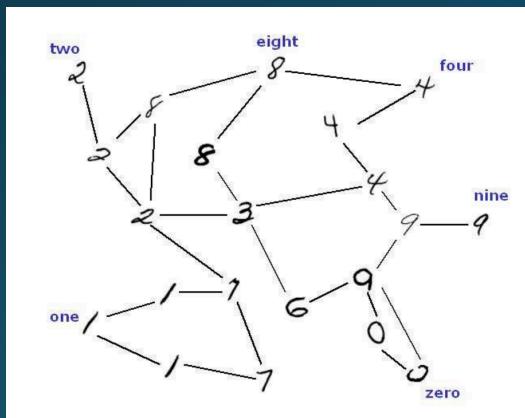
**Definition.** Given  $v_i \in \mathbf{V}^U$ , the weighted-vote relational-neighbor classifier (wvRN) estimates  $P(x_i|\mathcal{N}_i)$  as the (weighted) mean of the class-membership probabilities of the entities in  $\mathcal{N}_i$ :

$$P(x_i = c|\mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} w_{i,j} \cdot P(x_j = c|\mathcal{N}_j),$$

- Does this look familiar?
- **Homophily!**

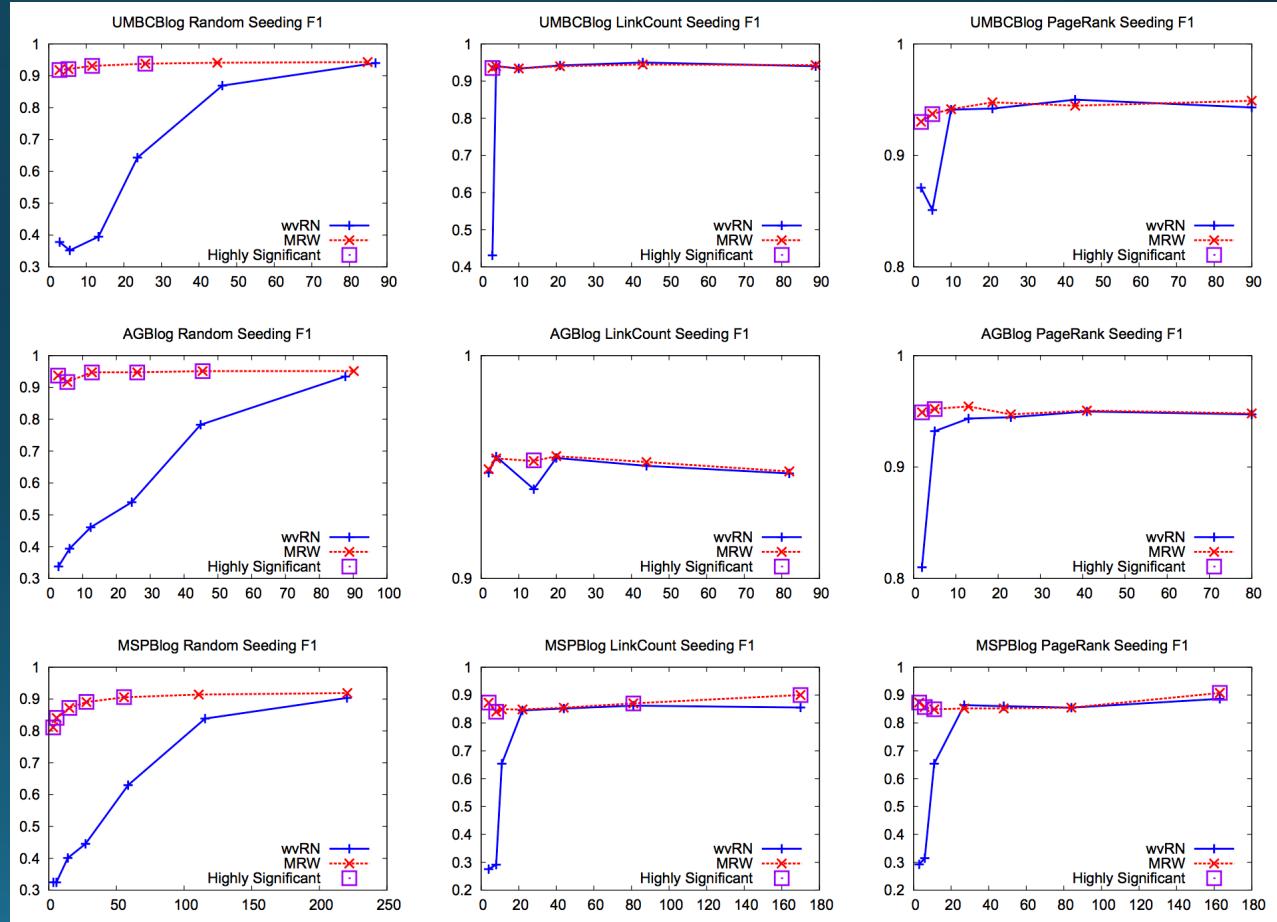
# Comparison: HF

- Another definition in [X. Zhu, Z. Ghahramani, and J. Lafferty, ICML 2003]
- A **harmonic field** – the score of each node in the graph is the harmonic, or linearly weighted, average of its neighbors' scores (harmonic field, HF)



# MRW versus wvRN

- MRW is easily the method to beat
- wvRN matches MRW **only** when seeding is **not** random
- Still takes a larger number of labeled instances compared to MRW



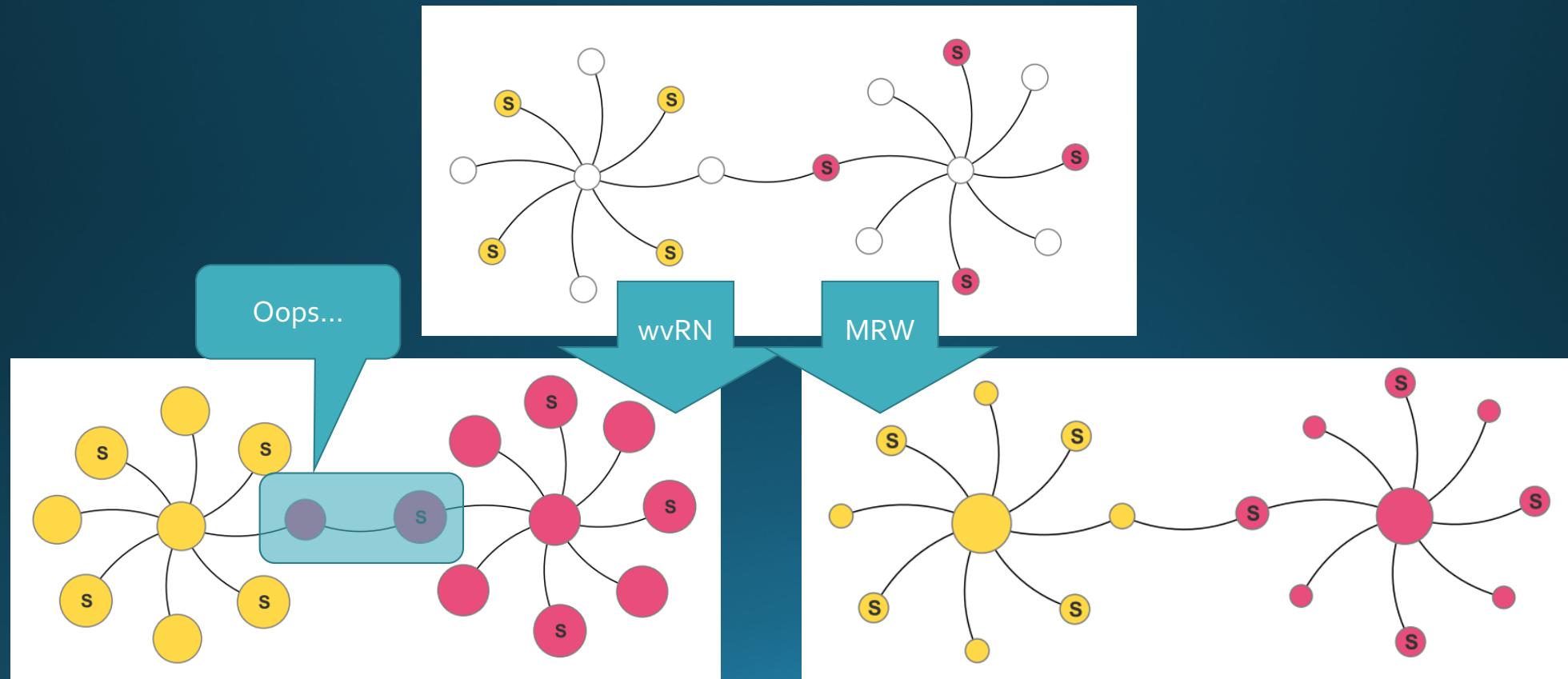
# Why is MRW > wvRN?

- Start with wvRN & HF objectives
- Do not account for graph *structure*
  - Or location of seeds
- **Graph-walk methods do not have these constraints**
  - And directly account for graph structure

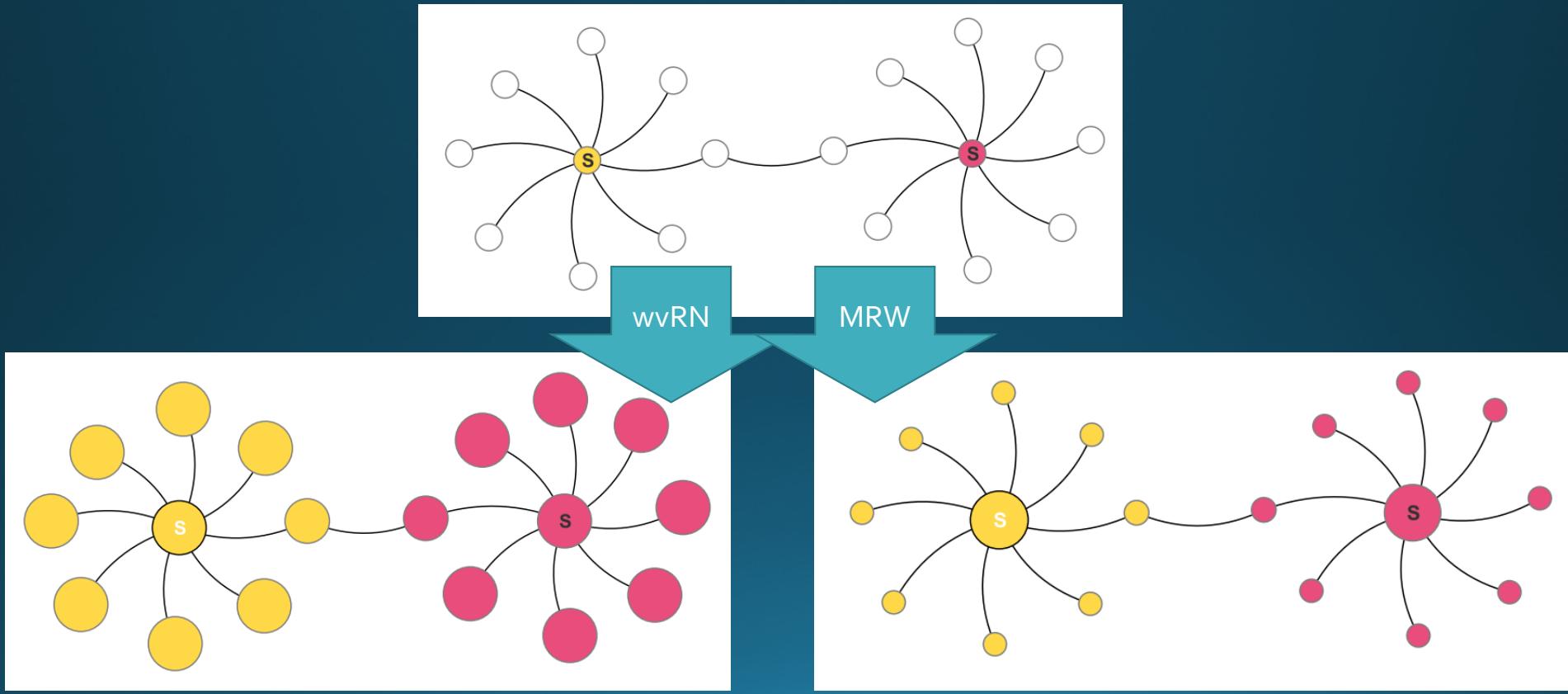
$$(6.2) \quad P(x_i = c|N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot P(x_j = c|N_j)$$

$$(6.3) \quad f(j) = \frac{1}{d_j} \sum_{i \sim j} w_{i,j} \cdot f(i)$$

# Why is MRW > wvRN?



# Why is MRW > wvRN?



# Notations

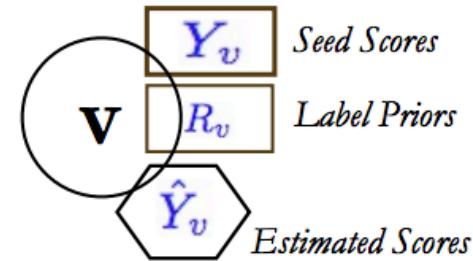
$\hat{Y}_{v,l}$  : score of estimated label  $l$  on node  $v$

$Y_{v,l}$  : score of seed 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)

$$\arg \min_{\hat{Y}} \sum_{l=1}^m W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 \quad \text{Smooth}$$

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

Graph Laplacian  
 $L = D - W$  (PSD)

Match Seeds (hard)

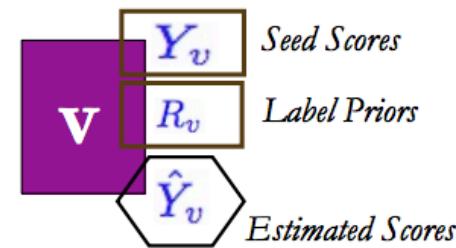
- Smoothness
  - two nodes connected by an edge with high weight should be assigned similar labels
- Solution satisfies harmonic property

# Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

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

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



# Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

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

How to do this minimization?

First, differentiate to find min is at

$$(\mu_1 \mathbf{S} + \mu_2 \mathbf{L} + \mu_3 \mathbf{I}) \hat{\mathbf{Y}}_l = (\mu_1 \mathbf{S}\mathbf{Y}_l + \mu_3 \mathbf{R}_l) .$$

**Jacobi method:**

- To solve  $\mathbf{Ax}=\mathbf{b}$  for  $\mathbf{x}$

- Iterate:

$$\mathbf{x}^{(k+1)} = D^{-1}(\mathbf{b} - R\mathbf{x}^{(k)}).$$

- ... or:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n.$$

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

$$\hat{\mathbf{Y}} \leftarrow \mathbf{Y}$$

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

$$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$$

repeat

    for all  $v \in V$  do

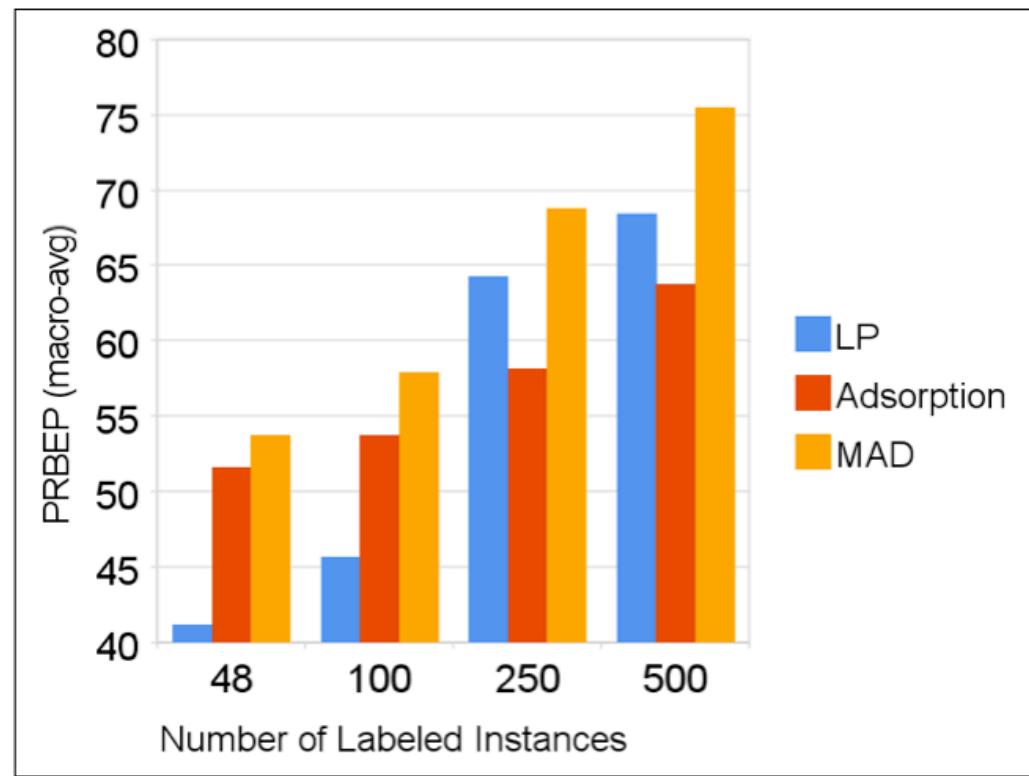
$$\hat{\mathbf{Y}}_v \leftarrow \frac{1}{Z_v} \left( (\mathbf{S}\mathbf{Y})_v + \mu_1 \mathbf{M}_v \cdot \hat{\mathbf{Y}} + \mu_2 \mathbf{R}_v \right)$$

    end for

until convergence

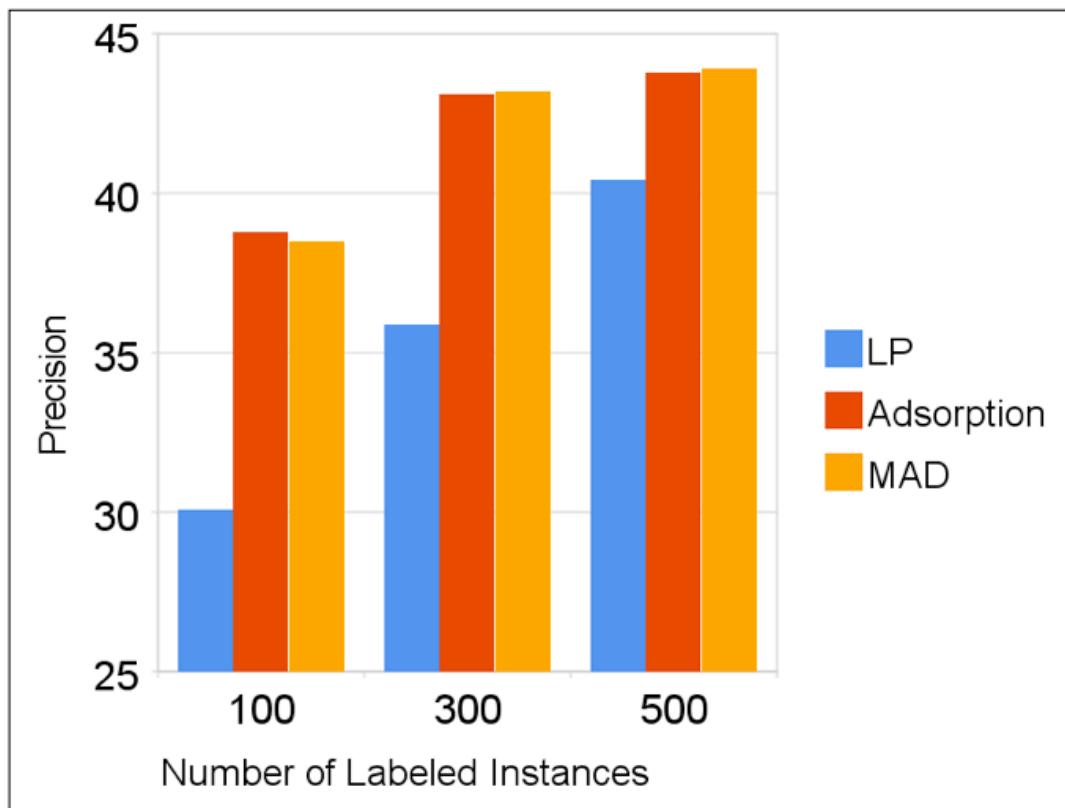
- Extends Adsorption with well-defined optimization
- Importance of a node can be discounted
- Easily Parallelizable: Scalable

# Text Classification



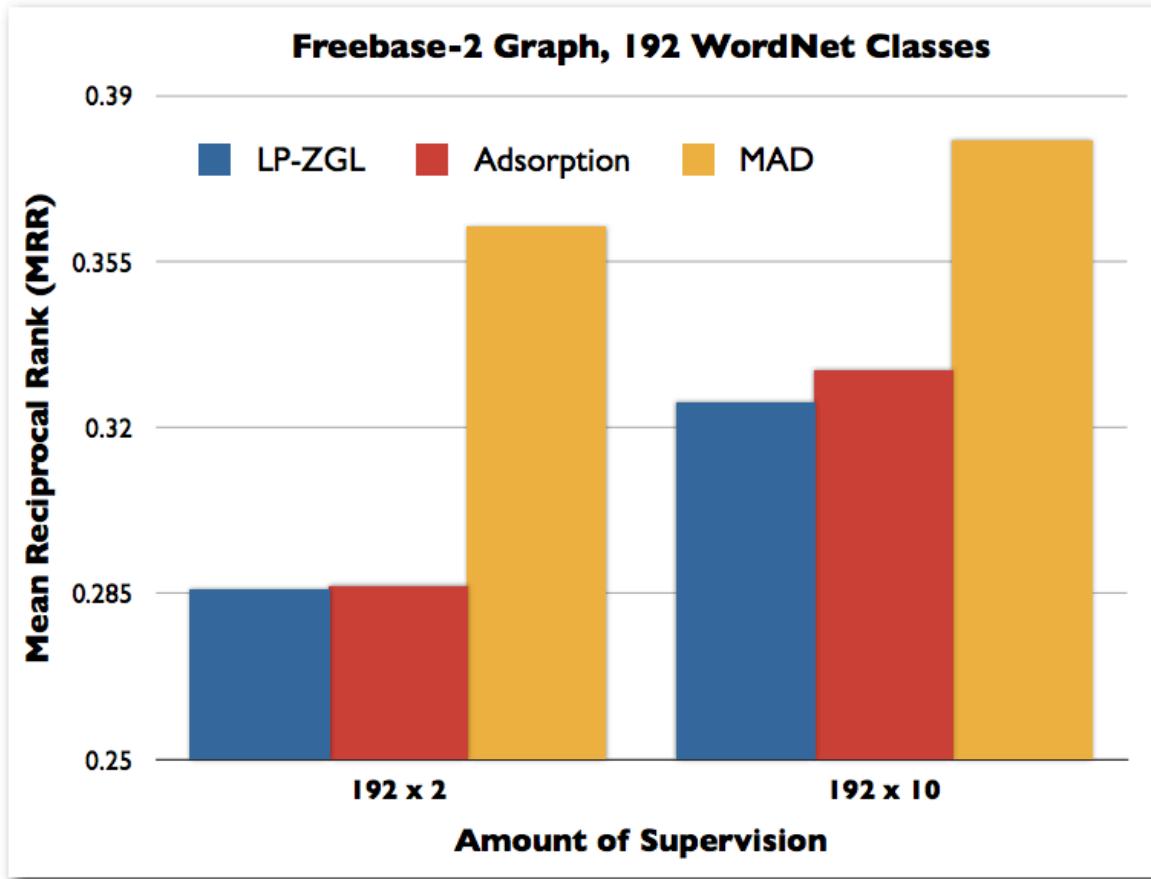
PRBEP (macro-averaged) on WebKB  
Dataset, 3148 test instances

# Sentiment Classification



Precision on 3568 Sentiment test instances

# Class-Instance Acquisition



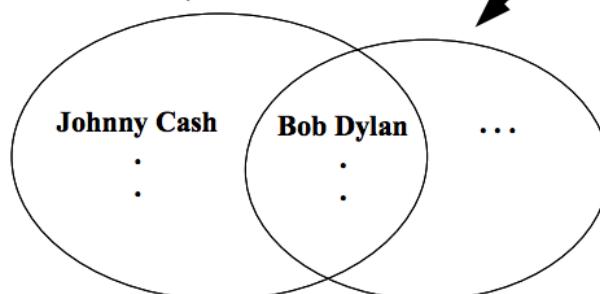
## ASSIGNING CLASS LABELS TO WEBTABLE INSTANCES

WebTable

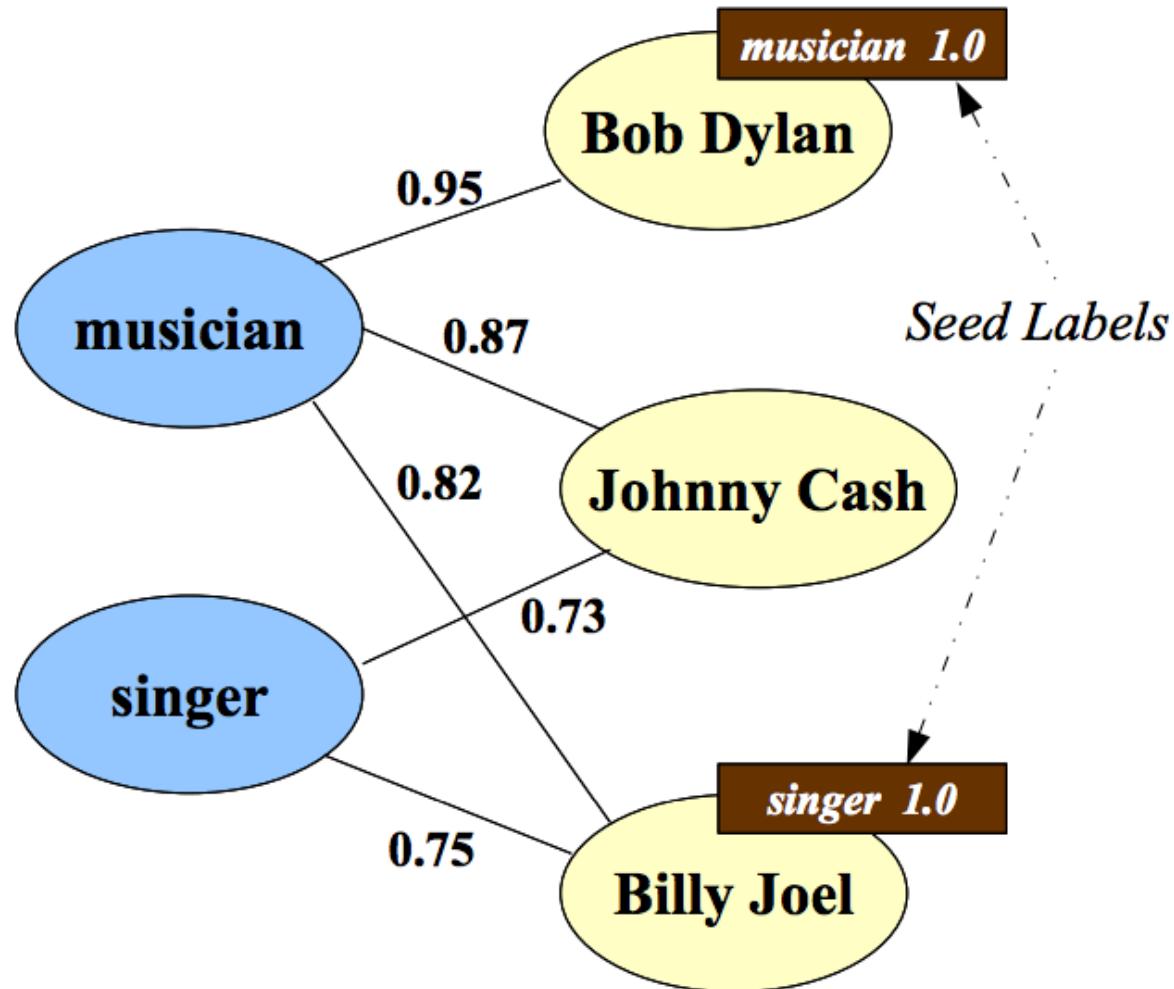
<i>Year</i>	<i>Artist</i>	<i>Albums</i>
:	:	:
:	Johnny Cash	:
:	Bob Dylan	:
:	:	:

A8

<i>musician</i>
:
Bob Dylan
:
:



*Score (musician, Johnny Cash) = 0.87*



# New (Class, Instance) Pairs Found

<b>Class</b>	<b>A few non-seed Instances found by Adsorption</b>
Scientific Journals	Journal of Physics, Nature, Structural and Molecular Biology, Sciences Sociales et sante, Kidney and Blood Pressure Research, American Journal of Physiology-Cell Physiology, ...
NFL Players	Tony Gonzales, Thabiti Davis, Taylor Stubblefield, Ron Dixon, Rodney Hannan, ...
Book Publishers	Small Night Shade Books, House of Ansari Press, Highwater Books, Distributed Art Publishers, Cooper Canyon Press, ...

# Modern SSL

- Graph Laplacians
  - Enforces graph structure
  - Imposes smoothness on labels
- Graph embeddings
  - “Embedding” ~ “context”
- Transductive -> Inductive
  - Transductive: learns the unlabeled data from the labeled data + structure
  - Inductive: generalizes to completely unobserved data

# Graph Laplacians

- Reformulate SSL objective as two distinct terms:

$$f^T L f = \frac{1}{2} \sum_{i,j} W_{ij} (f(i) - f(j))^2$$

Weighted sum of supervised loss over *labeled* instances

$$J(f) = f^T L f + \sum_{i=1}^l \lambda (f(i) - y_i)^2 = f^T L f + (f - y)^T \Lambda (f - y)$$

Graph Laplacian regularization term

# Graph Embeddings

- Remember word embeddings with word2vec?
- **Context!**
- Estimate “context” of each node with a random walk over neighborhood of a fixed window size
- Skipgram-based model, DeepWalk

$$-\sum_{(i,c)} \log p(c|i) = -\sum_{(i,c)} \left( \mathbf{w}_c^T \mathbf{e}_i - \log \sum_{c' \in \mathcal{C}} \exp(\mathbf{w}_{c'}^T \mathbf{e}_i) \right)$$

- $\mathcal{C}$  is set of all possible context
- $w$ 's are parameters of Skipgram
- $e_i$  is embedding of node  $i$

# Inductive SSL

- You start with  $X^l$  (labeled) and  $X^U$  (unlabeled), hoping their combination will result in a superior model
- Semi-supervised learning yields predictions on  $X^U$ 
  - Transductive learning
- What if a *completely unobserved* data point shows up?
  - Inductive learning—a concept often left out in SSL literature
- **Convert your SSL framework to classification!**

Transductive learning (note embeddings)

$$p(y|\mathbf{x}, \mathbf{e}) = \frac{\exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{e})^T] \mathbf{w}_y}{\sum_{y'} \exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{e})^T] \mathbf{w}_{y'}},$$

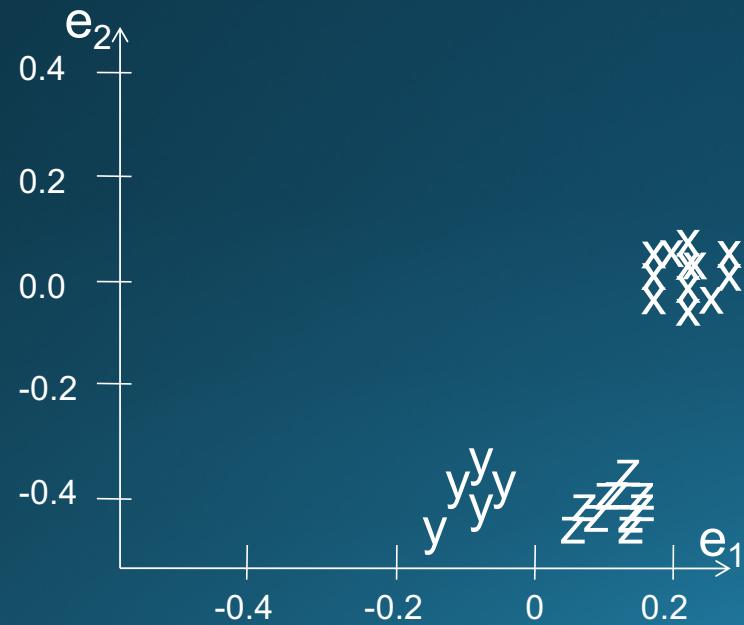
Inductive learning (dependent only on  $x$ )

$$p(y|\mathbf{x}) = \frac{\exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{x})^T] \mathbf{w}_y}{\sum_{y'} \exp[\mathbf{h}^k(\mathbf{x})^T, \mathbf{h}^l(\mathbf{x})^T] \mathbf{w}_{y'}}$$

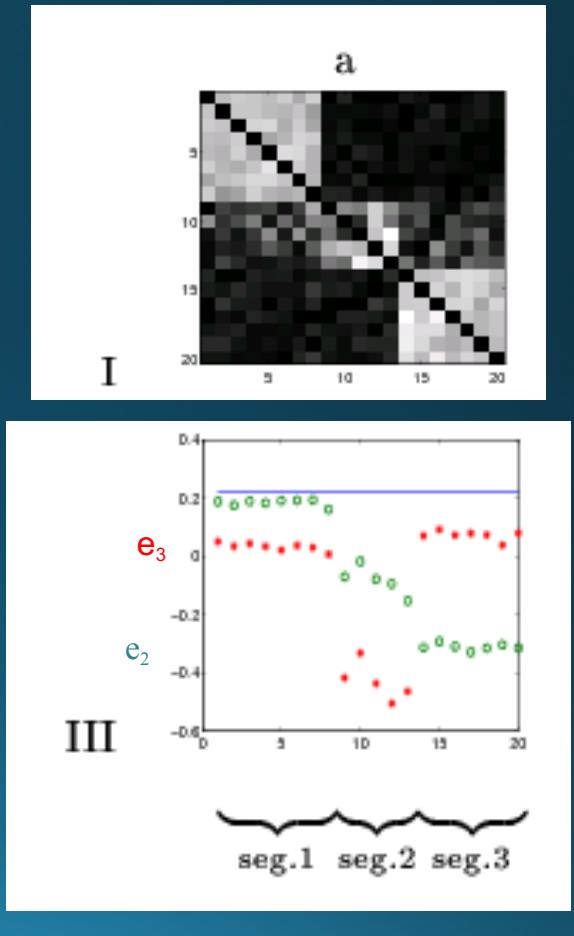
Quick digression to unsupervised learning...

# Spectral Clustering

- Graph = Matrix
  - $W^*v_1 = v_2$  “propogates weights from neighbors”



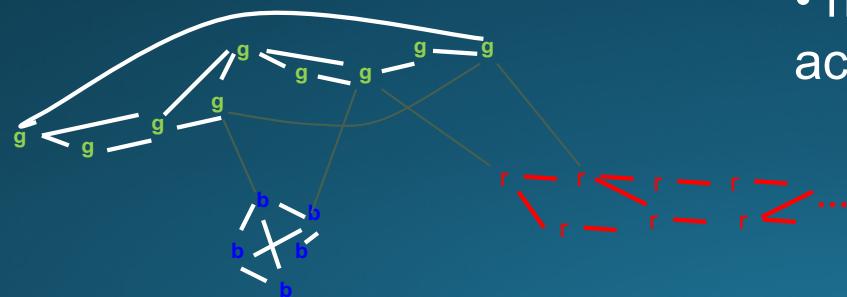
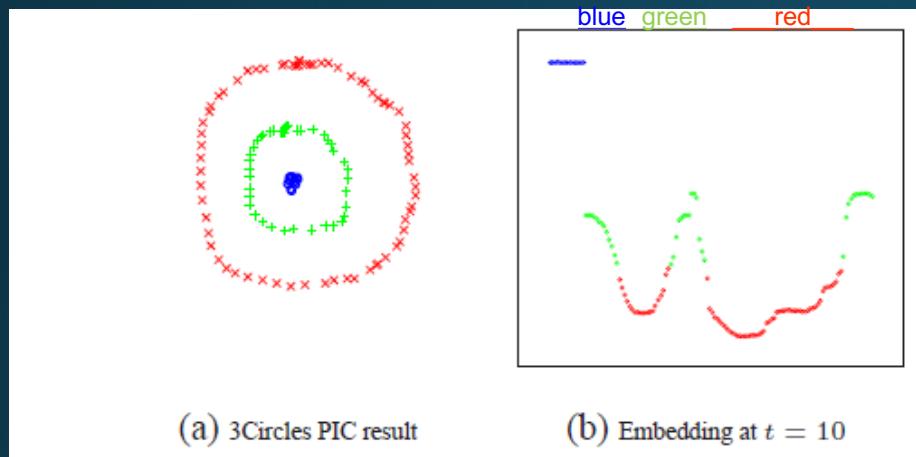
[Shi & Meila, 2002]



## Repeated averaging with neighbors as a clustering method

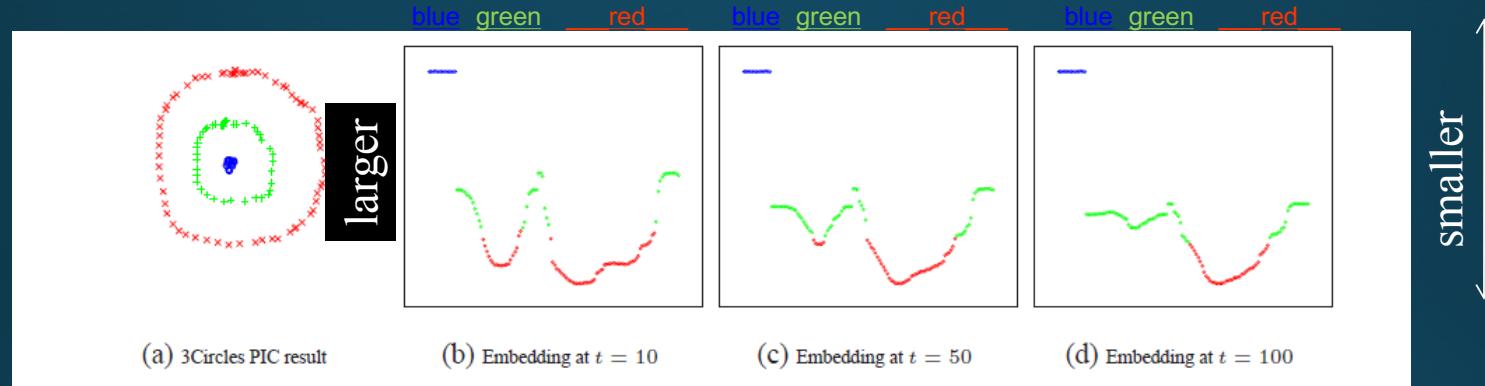
- Pick a vector  $v^0$  (maybe at random)
- Compute  $v^1 = Wv^0$ 
  - i.e., replace  $v^0[x]$  with *weighted average* of  $v^0[y]$  for the neighbors  $y$  of  $x$
- Plot  $v^1[x]$  for each  $x$
- Repeat for  $v^2, v^3, \dots$
- Variants widely used for *semi-supervised* learning
  - clamping of labels for nodes with known labels
- Without clamping, will converge to constant  $v^t$
- What are the *dynamics* of this process?

Repeated averaging with neighbors on a sample problem...

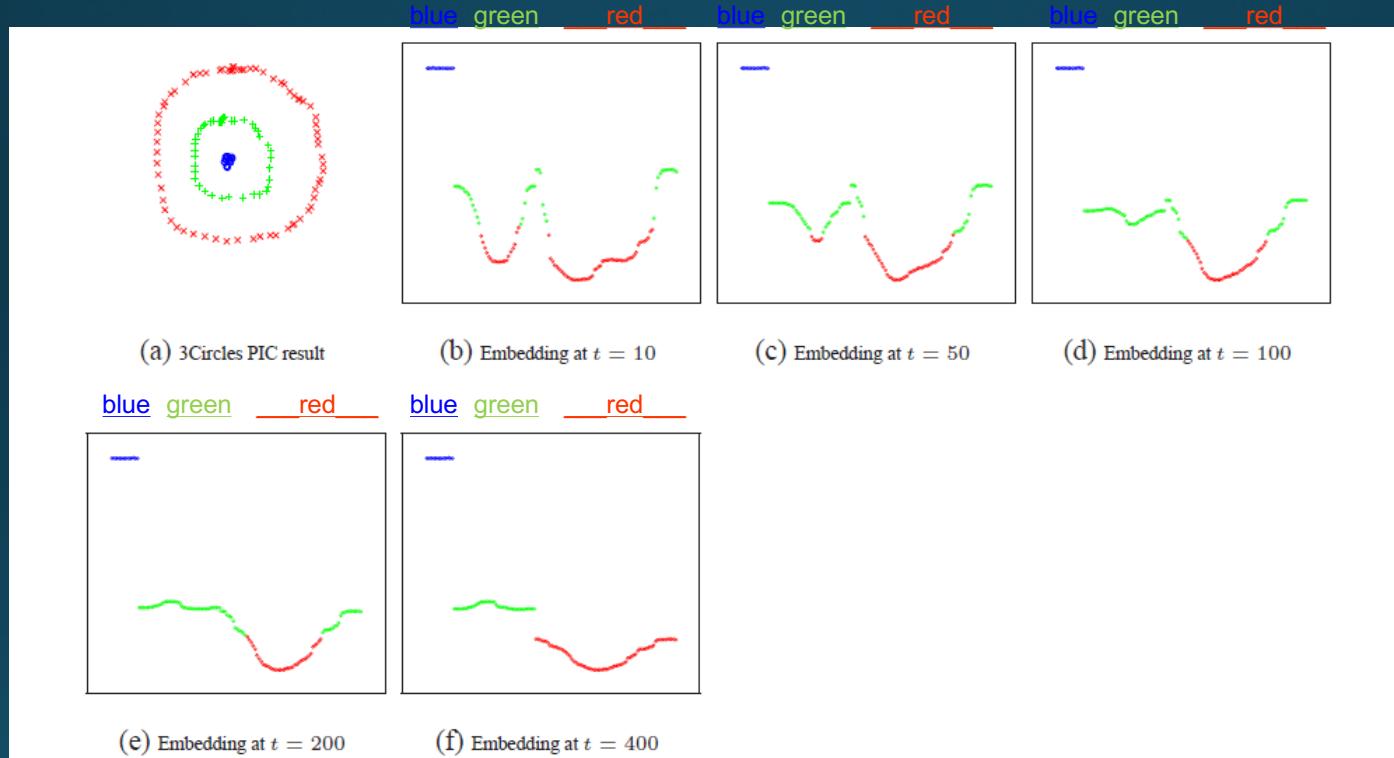


- Create a graph, connecting all points in the 2-D initial space to all other points
  - Weighted by distance
- Run power iteration for 10 steps
- Plot node id  $x$  vs  $v^{10}(x)$ 
  - nodes are ordered by actual cluster number

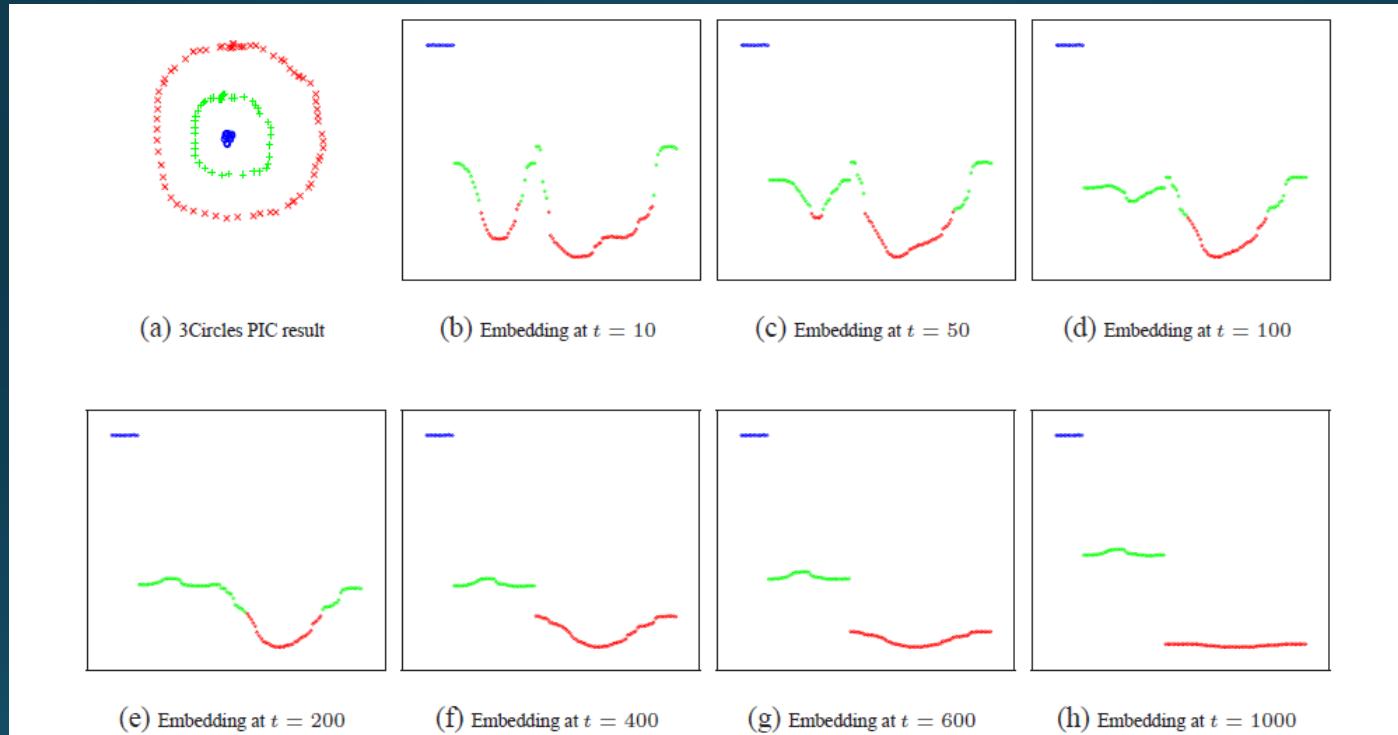
Repeated averaging with neighbors on a sample problem...



Repeated averaging with neighbors on a sample problem...



Repeated averaging with neighbors on a sample problem...



# PIC: Power Iteration Clustering

- Run power iteration (repeated averaging w/ neighbors) with early stopping

1. Pick an initial vector  $\mathbf{v}^0$ .
2. Set  $\mathbf{v}^{t+1} \leftarrow \frac{W\mathbf{v}^t}{\|W\mathbf{v}^t\|_1}$  and  $\delta^{t+1} \leftarrow |\mathbf{v}^{t+1} - \mathbf{v}^t|$ .
3. Increment  $t$  and repeat above step until  $|\delta^t - \delta^{t-1}| \simeq 0$ .
4. Use  $k$ -means to cluster points on  $\mathbf{v}^t$  and return clusters  $C_1, C_2, \dots, C_k$ .

- $\mathbf{v}^0$ : random start, or “degree matrix”  $D$ , or others
- Easy to implement, and relatively efficient (& easily parallelized!)
- Empirically, often **better** than traditional spectral methods
  - Surprising given embedded space is 1-dimensional!

# References

- “Semi-Supervised Classification of Network Data Using Very Few Labels”,  
<https://lti.cs.cmu.edu/sites/default/files/research/reports/2009/cmultio9017.pdf>
- “New Regularized Algorithms for Transductive Learning”,  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.220.42&rep=rep1&type=pdf>
- “Semi-supervised Learning in Gigantic Image Collections”,  
<http://papers.nips.cc/paper/3633-semi-supervised-learning-in-gigantic-image-collections.pdf>
- “Revisiting Semi-Supervised Learning with Graph Embeddings”,  
<http://proceedings.mlr.press/v48/yang16.pdf>