Semi-Supervised Learning

William Cohen

Outline

- The general idea and an example (NELL)
- Some types of SSL
  - Margin-based: transductive SVM
  - Generative: seeded k-means, seeded EM
  - Nearest-neighbor like: graph-based SSL

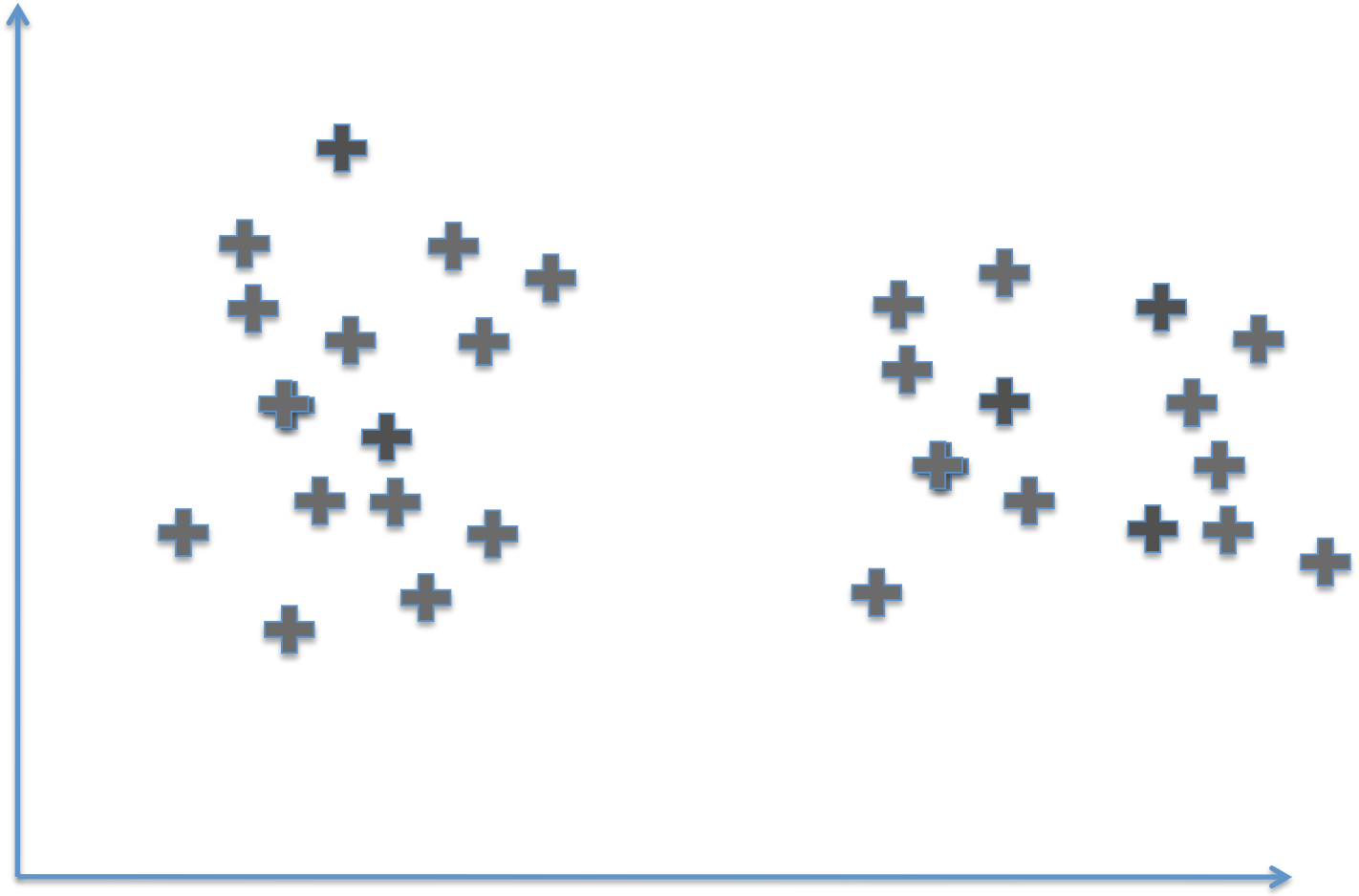
- Some research results with graph-based SSL

INTRO TO SEMI-SUPERVISED LEARNING (SSL)

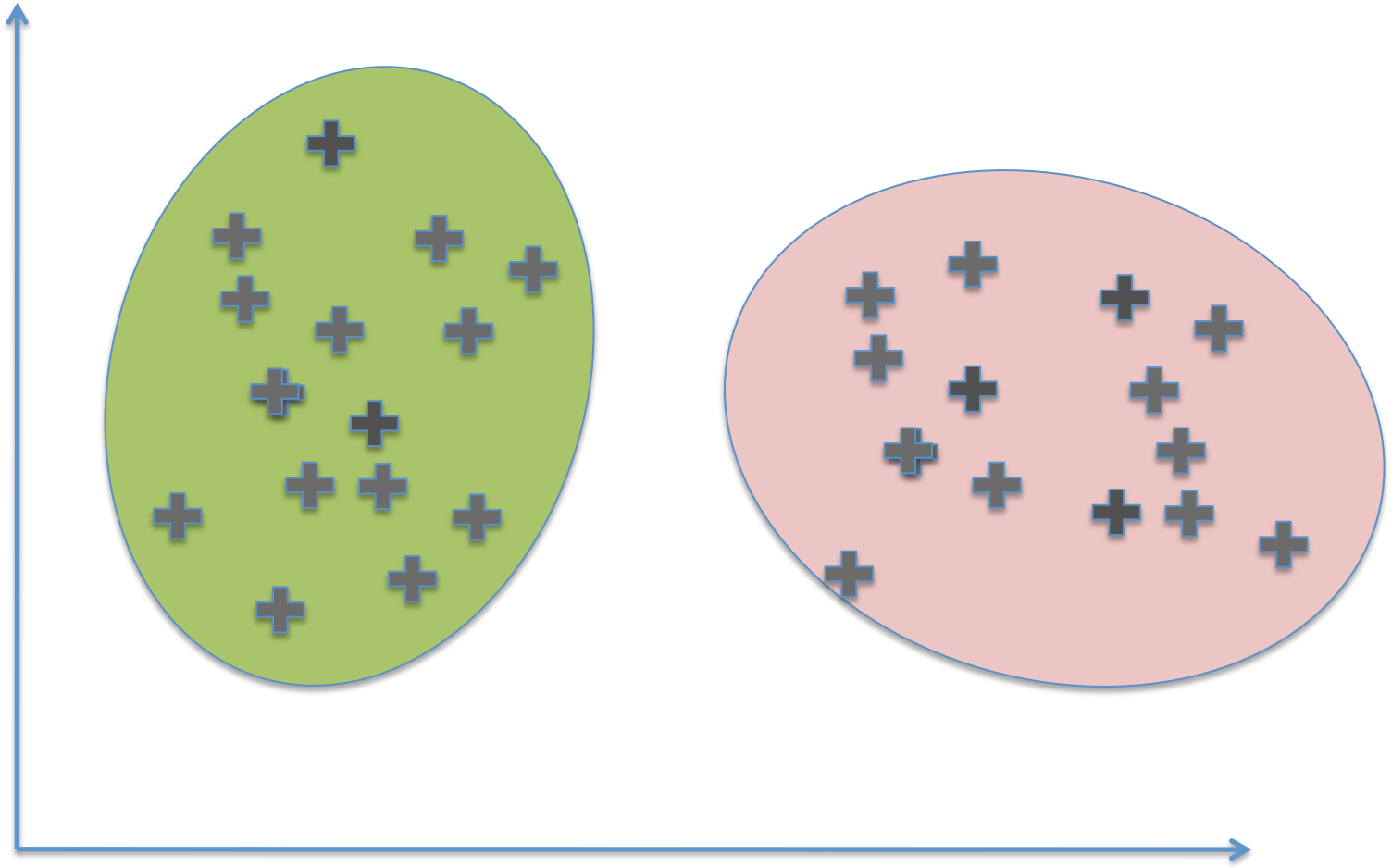
Semi-supervised learning

- Given:
  - A pool of labeled examples L
  - A (usually larger) pool of unlabeled examples U
- Option 1 for using L and U :
  - Ignore U and use supervised learning on L
- Option 2:
  - Ignore labels in L+U and cluster, or perform PCA, etc to reduce dimension, then use supervised learning
- Question:
  - Can you use both L and U to do better?

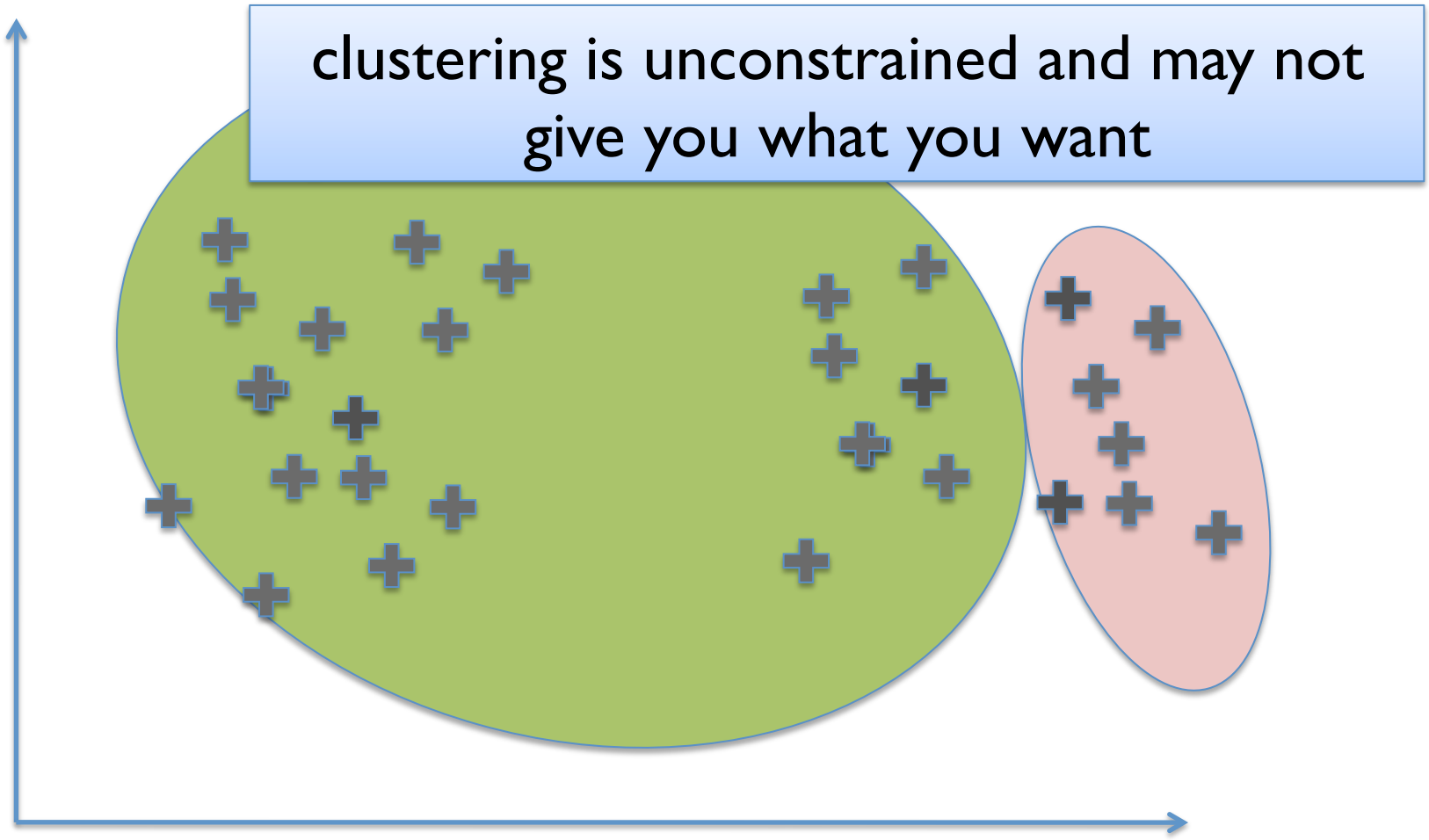
SSL is Somewhere Between Clustering and Supervised Learning



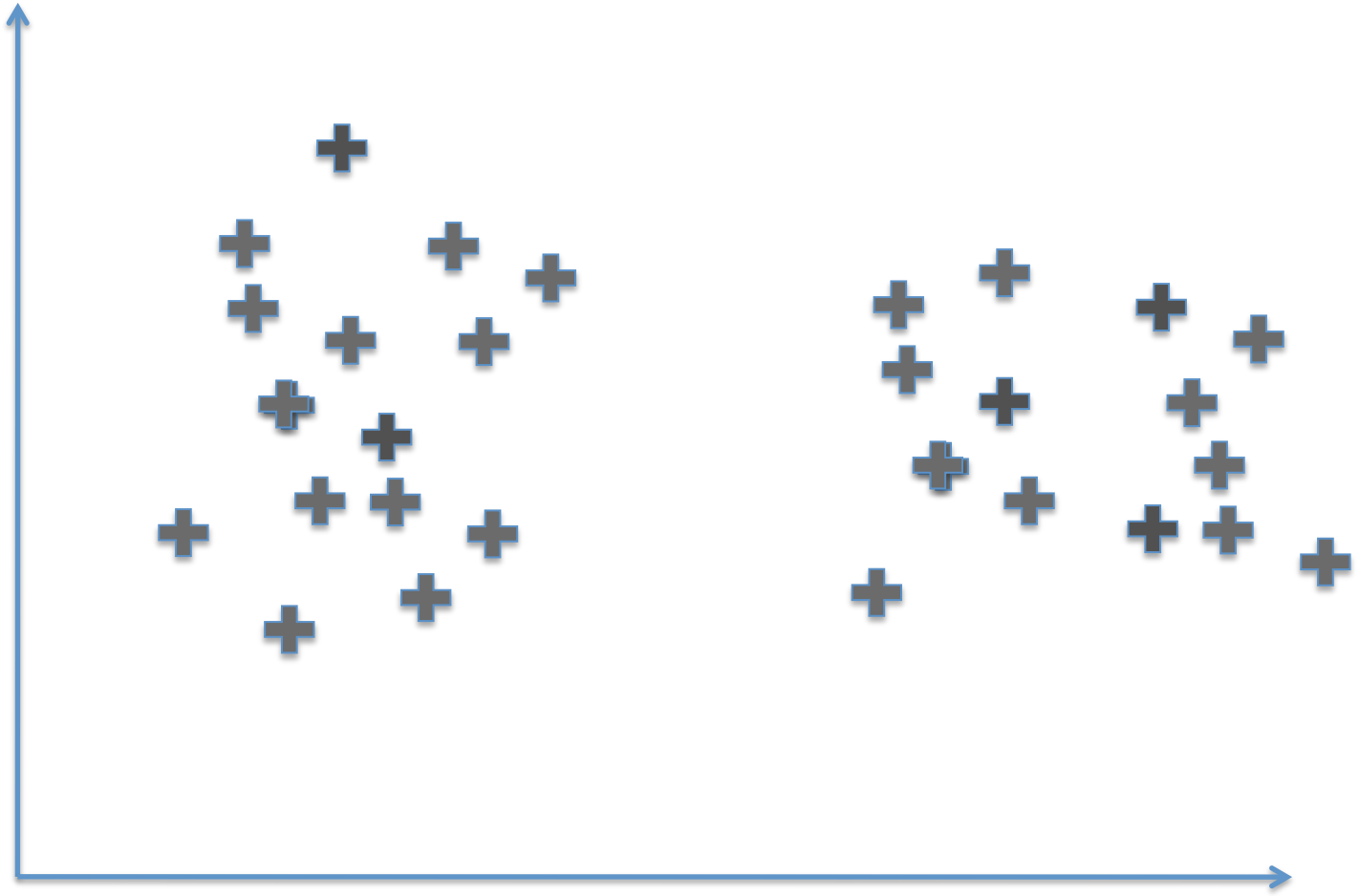
SSL is Between Clustering and SL



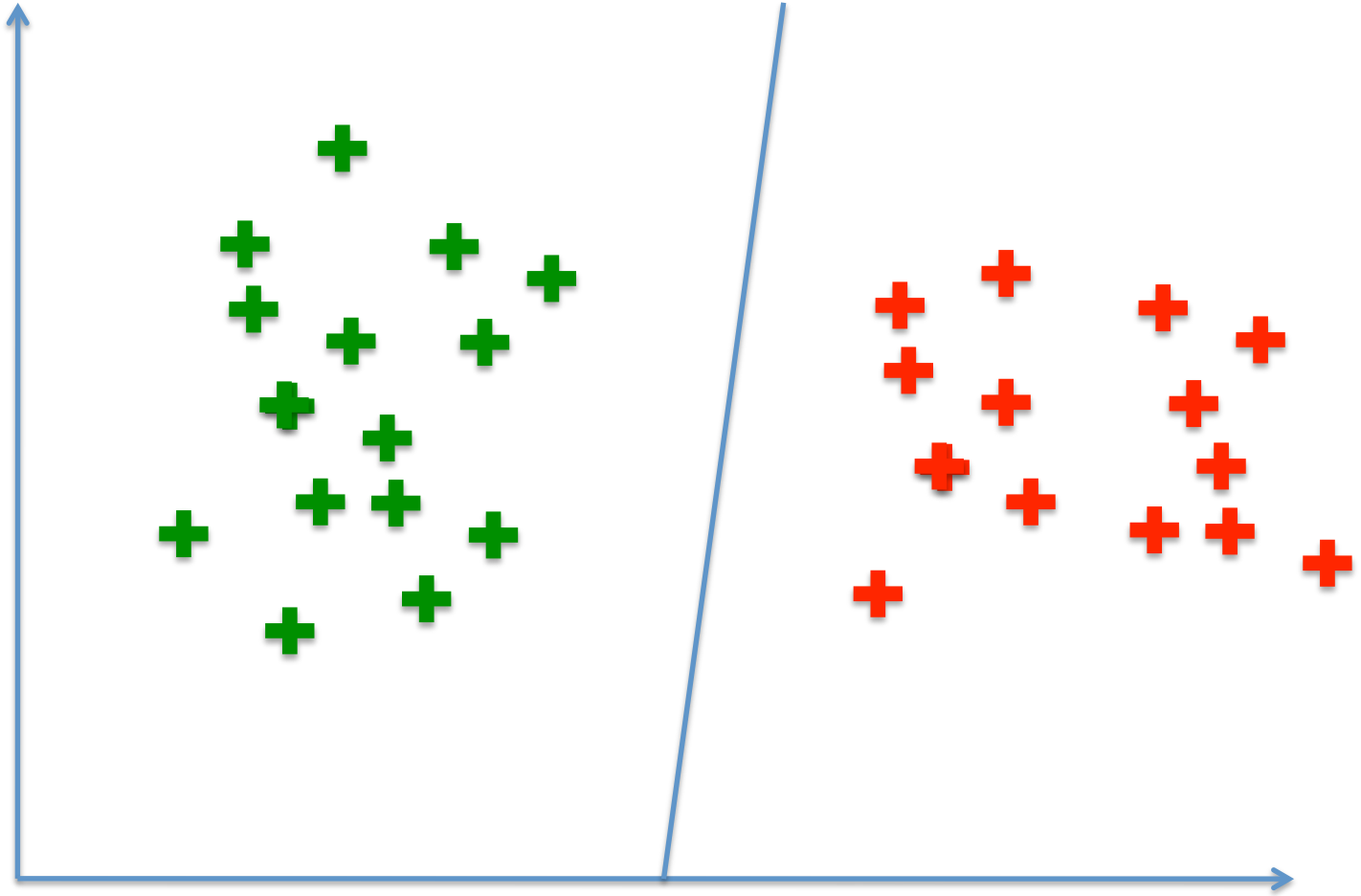
SSL is Between Clustering and SL



SSL is Between Clustering and SL

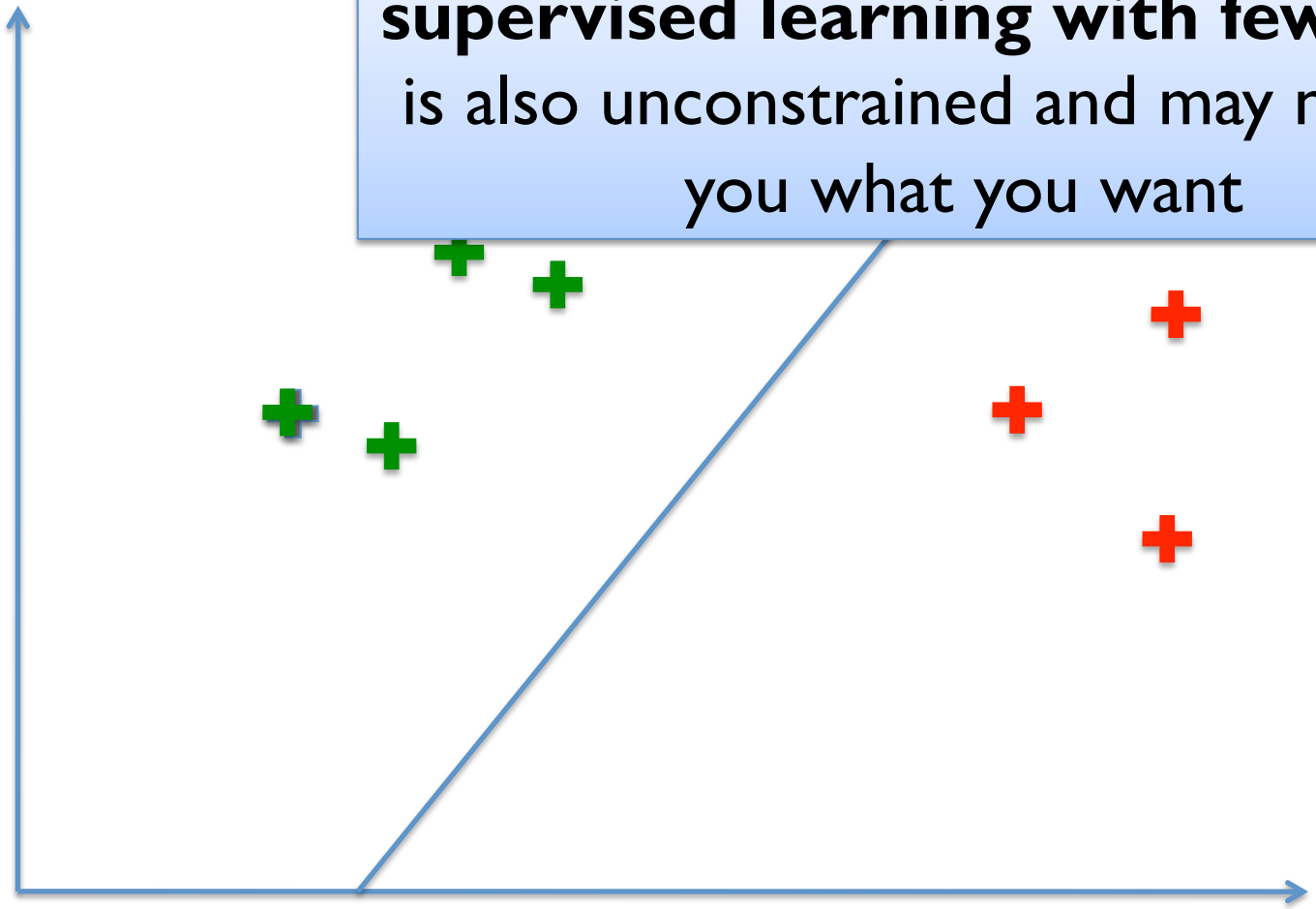


SSL is Between Clustering and SL

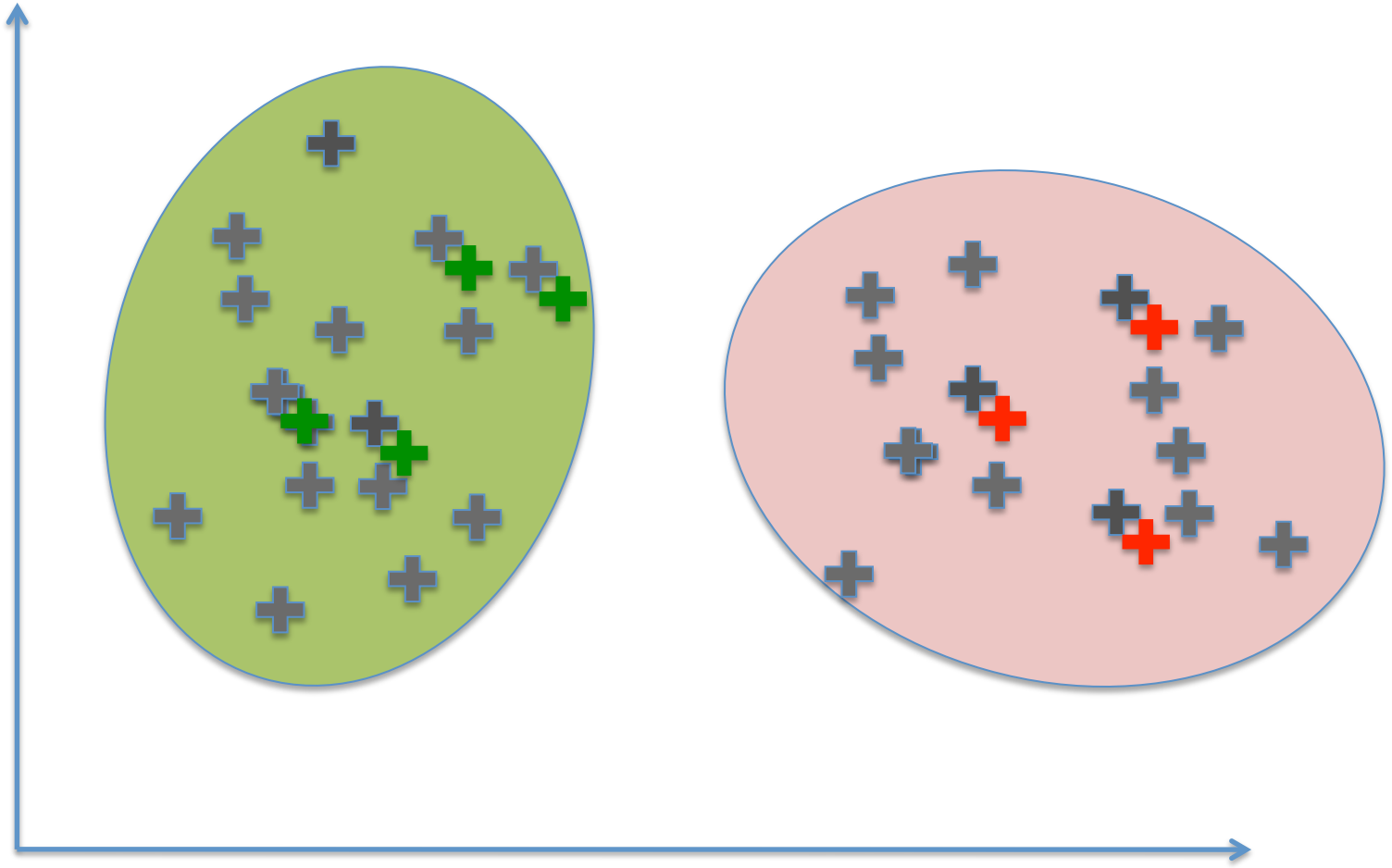


SSL is Between Clustering and SL

supervised learning with few labels is also unconstrained and may not give you what you want



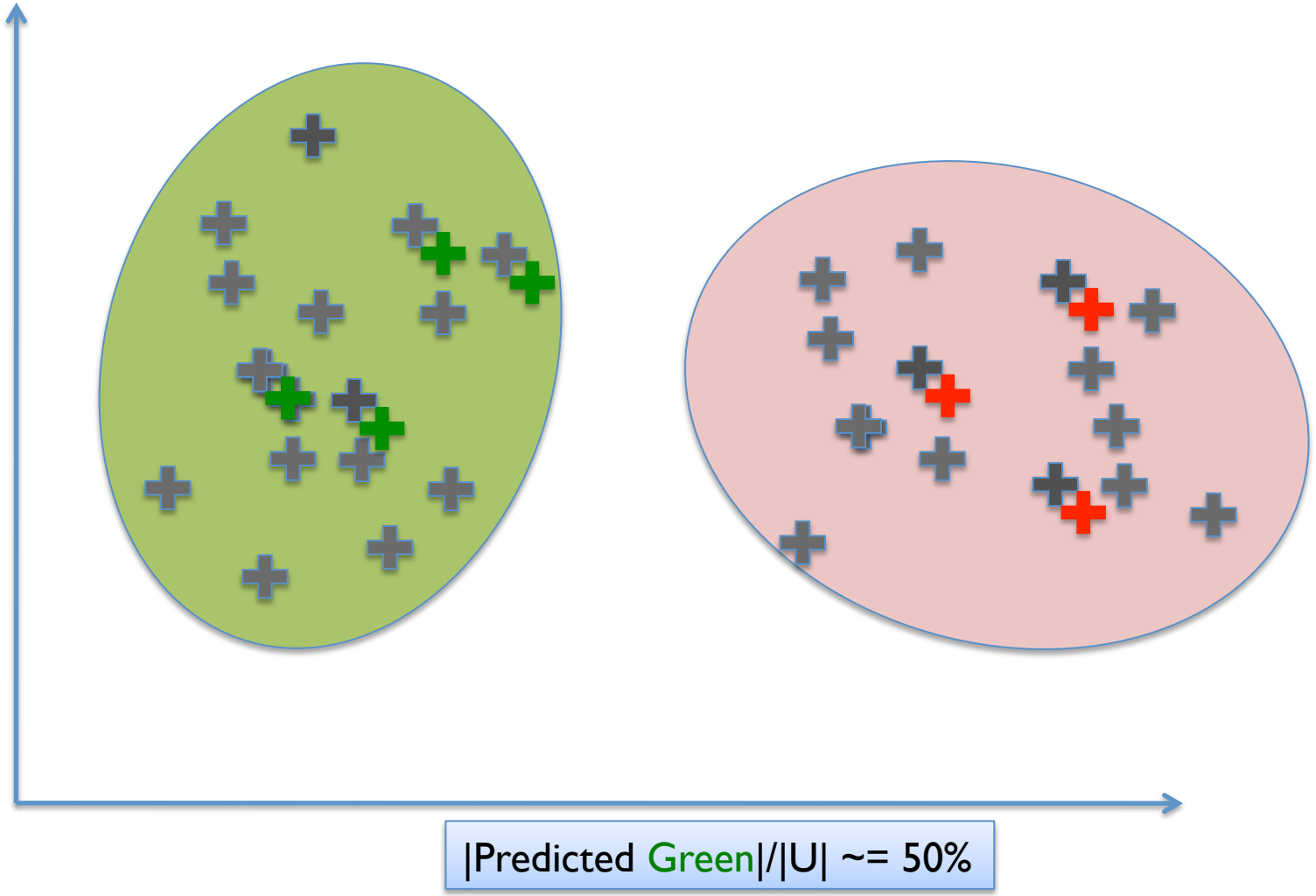
SSL is Between Clustering and SL



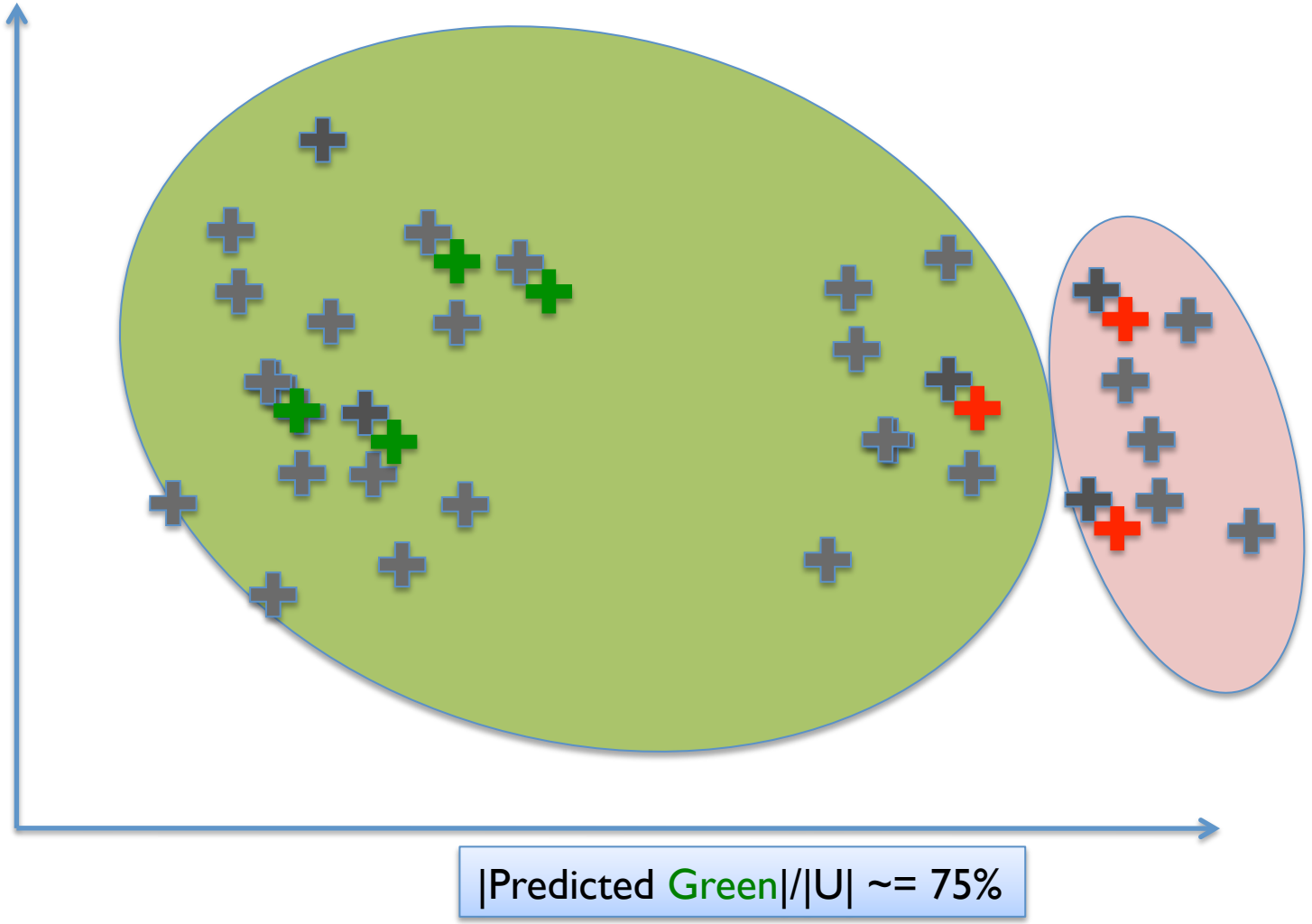
Some general issues with SSL

- How much unlabeled data do you want?
  - Suppose you're optimizing $J = J_L(L) + J_U(U)$
  - If |U| >> |L| does J_U dominate J?
    - If so you're basically just clustering
  - Sometimes we need to balance J_L and J_U
- Besides L, what other information about the task is useful?
  - Common choice: true frequency of classes

SSL is Between Clustering and SL



SSL is Between Clustering and SL



SSL in Action: The NELL System

Outline

- The general idea and an example (NELL)
- Some types of SSL
  - Margin-based: transductive SVM
  - Generative: seeded k-means, seeded EM
  - Nearest-neighbor like: graph-based SSL

TRANSDUCTIVE SVM

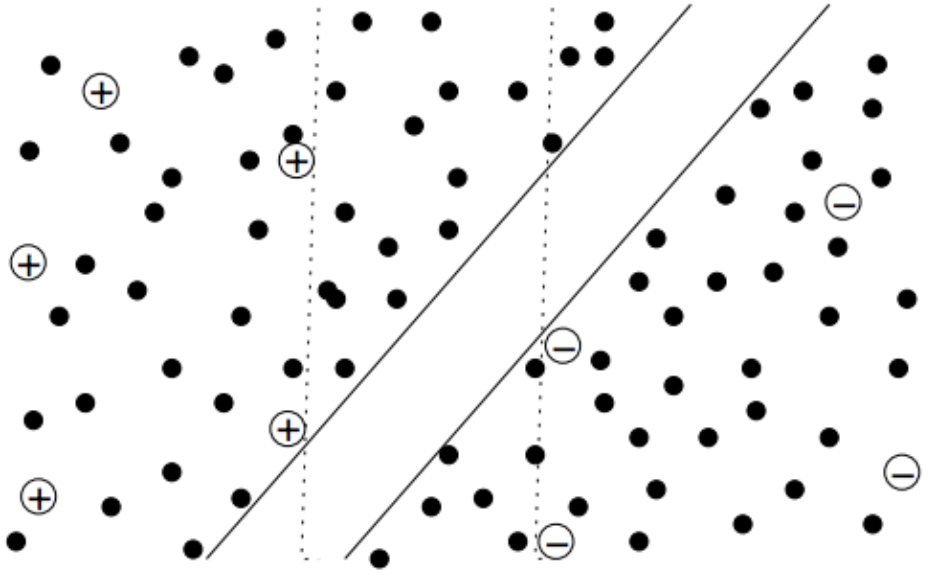
Two Kinds of Learning

- **Inductive** SSL:
  - Input: training set
    - $(x_1,y_1),\ldots,(x_n,y_n)$
    - $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$
  - Output: classifier
    - $f(x) = y$
  - Classifier can be run on any test example *x*

- **Transductive** SSL:
  - Input: training set
    - $(x_1,y_1),\ldots,(x_n,y_n)$
    - $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$
  - Output: classifier
    - $f(x_i) = y$
  - Classifier is only defined for x_i's *seen at training time*

Transductive Support Vector Machines

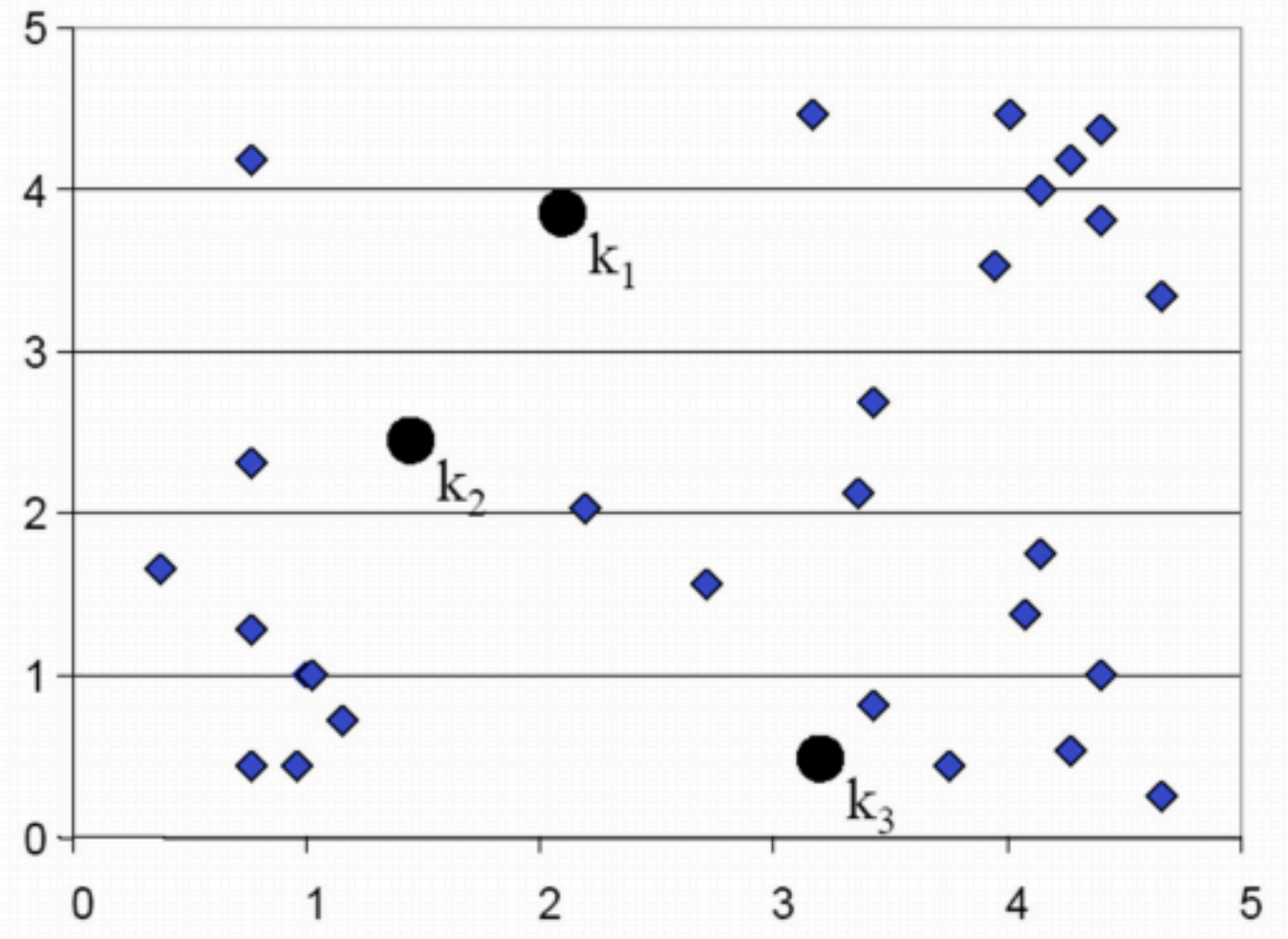
Instead of finding maximum margin between labelled points, optimize over both margin and labels of unlabelled points.



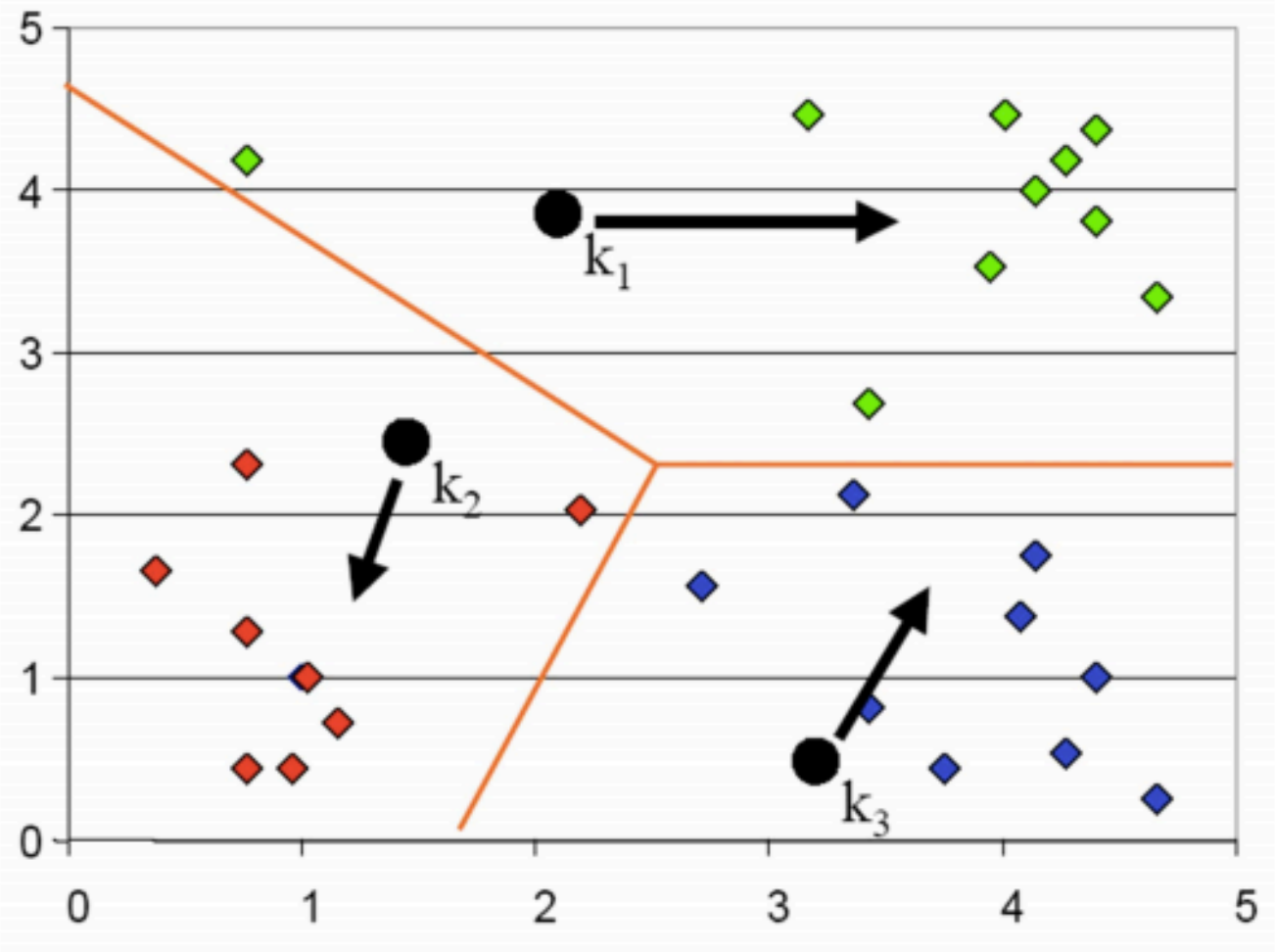
Effective but not convex and more expensive than supervised SVM

SEMI-SUPERVISED K-MEANS AND MIXTURE MODELS

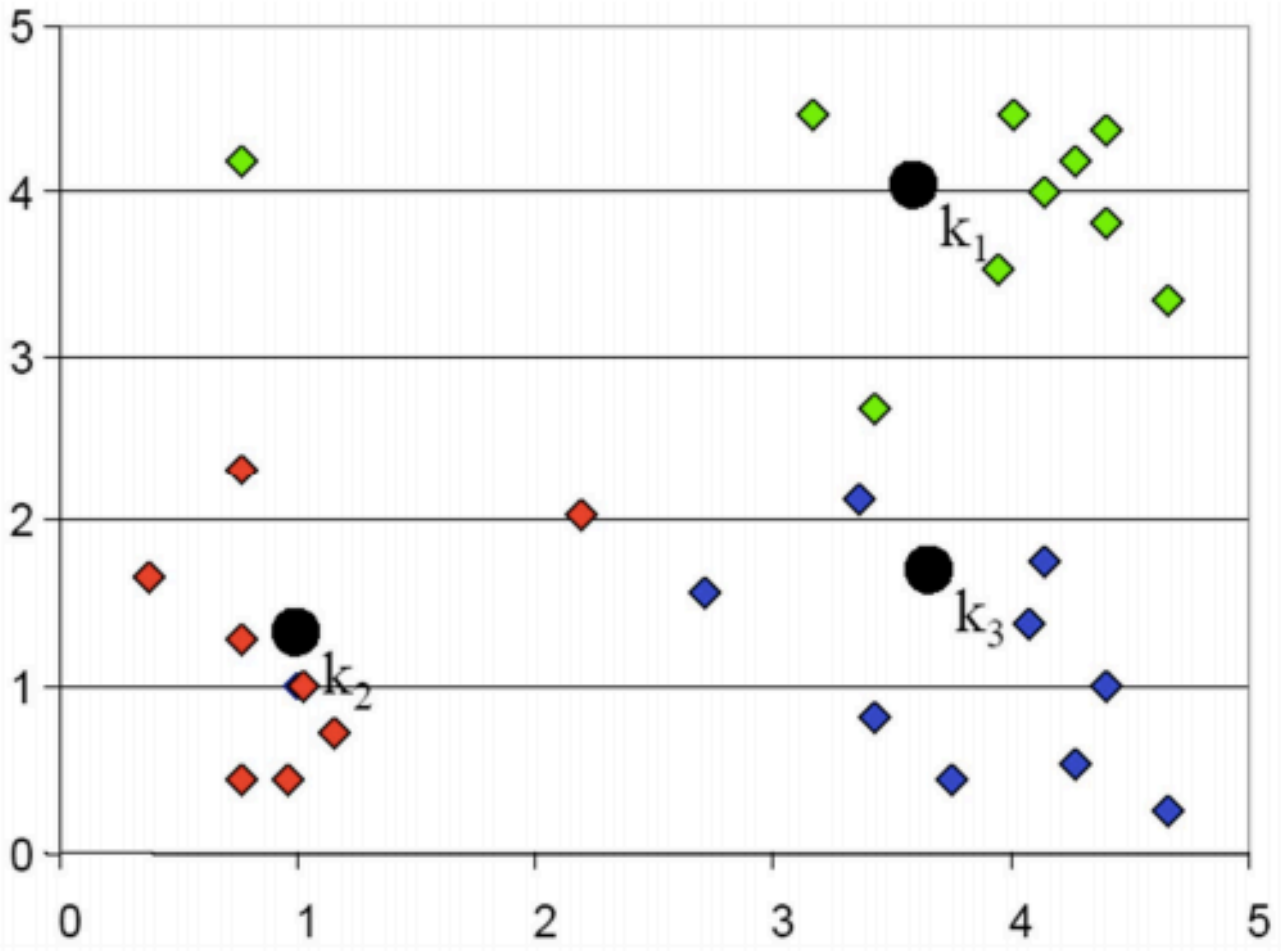
K-means Clustering: Step 1



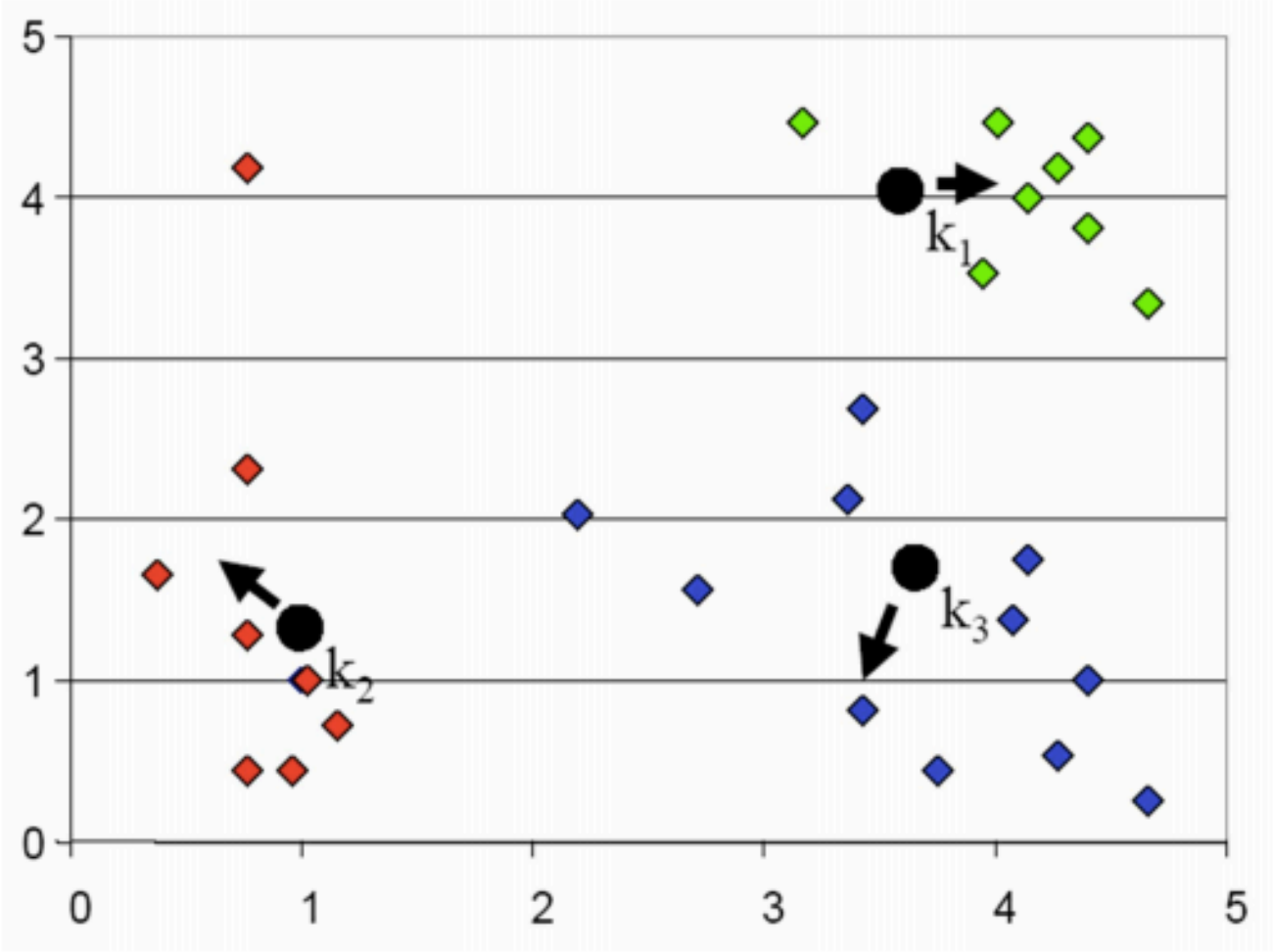
K-means Clustering: Step 2



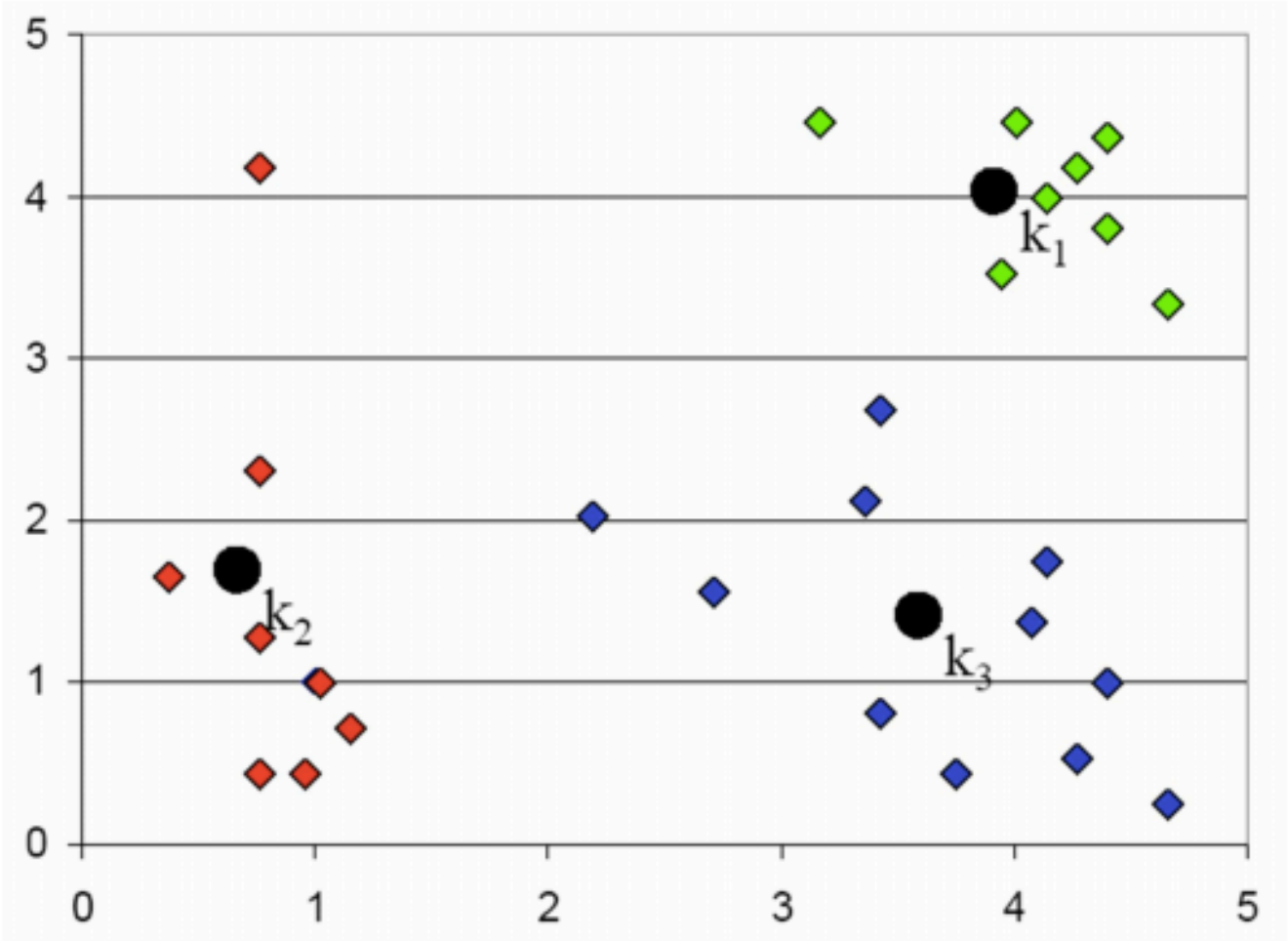
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



K-Means

Algorithm

1. Decide on a value for *k*.
2. Initialize the *k* cluster centers randomly if necessary.
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.
5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise go to 3.

Seeded k-means

Algorithm

1. Decide on a value for *k*.
2. Initialize the *k* cluster centers
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.
5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise go to 3.

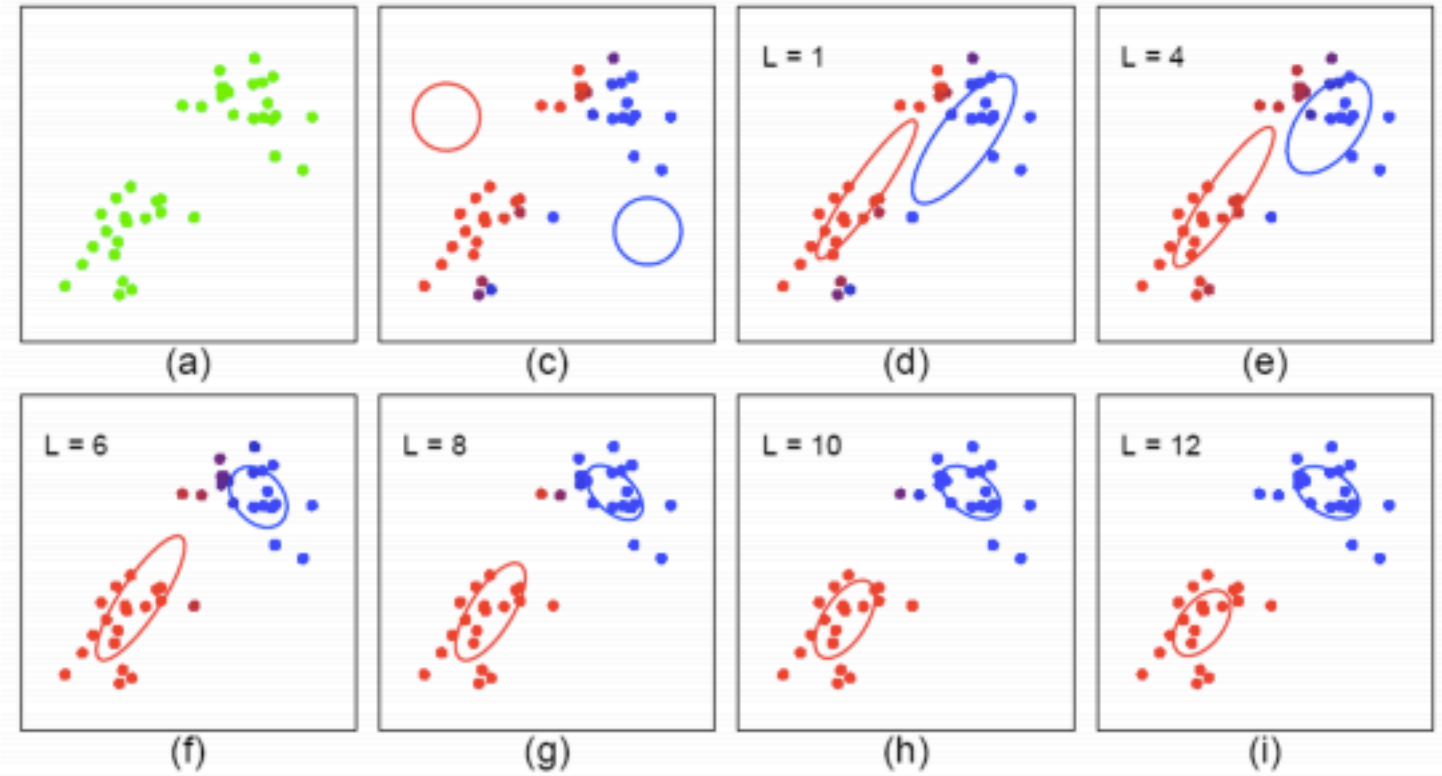
k is the number of classes

using the labeled "seed" data

except keep the seeds in the class they are *known* to belong to

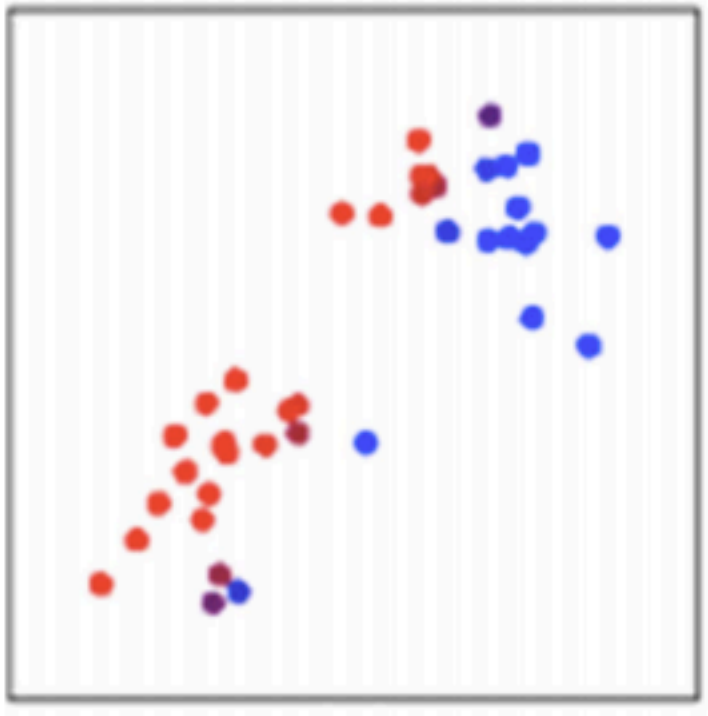
What about E/M ("soft k-means")?

- Start:
  - "Guess" the centroid μ_k and coveriance Σ_k of each of the K clusters
- Loop



What about E/M ("soft k-means")?

- A "soft" k-means



E:

$$\tau_n^{k(t)} = \left\langle z_n^k \right\rangle_{q^{(t)}} = p(z_n^k = 1 \mid x, \mu^{(t)}, \Sigma^{(t)})$$

z^k=1 for seed (x,y=k)

M:

$$\pi_k^* = \sum_n \tau_n^{k(t)} \Big/ N = \langle n_k \rangle \Big/ N$$

$$\mu_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} x_n}{\sum_n \tau_n^{k(t)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_n \tau_n^{k(t)}}$$

Sample results with SSL

Machine Learning, 39, 103–134, 2000.

Text Classification from Labeled and Unlabeled Documents using EM

KAMAL NIGAM knigam@cs.cmu.edu
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ANDREW KACHITES MCCALLUM mccallum@justresearch.com
Just Research, 4616 Henry Street, Pittsburgh, PA 15213, USA; School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

SEBASTIAN THRUN thrun@cs.cmu.edu
TOM MITCHELL tom.mitchell@cmu.edu
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Editor: William Cohen

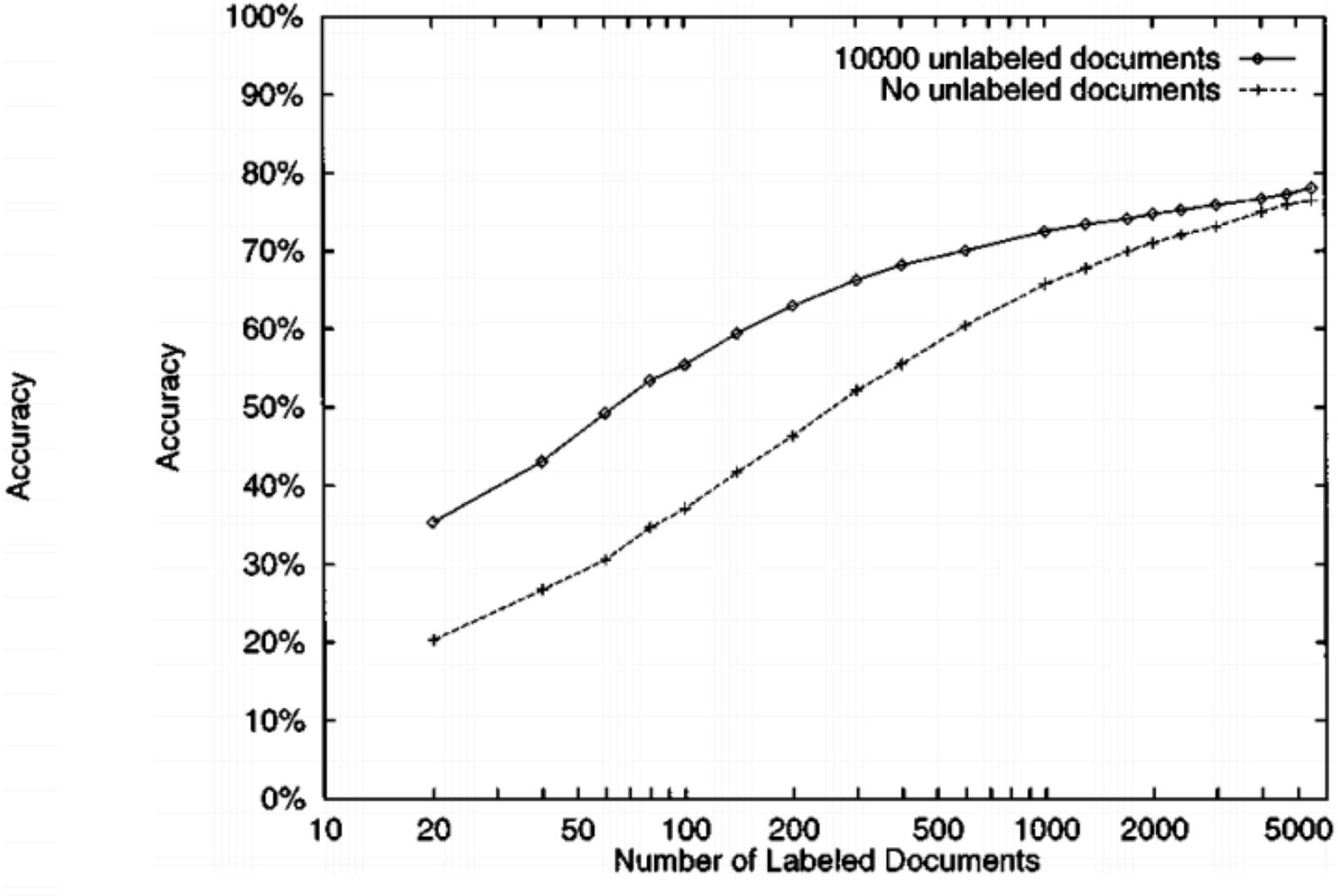
Sample results with SSL

Table 1. The basic EM algorithm described in Section 5.1.

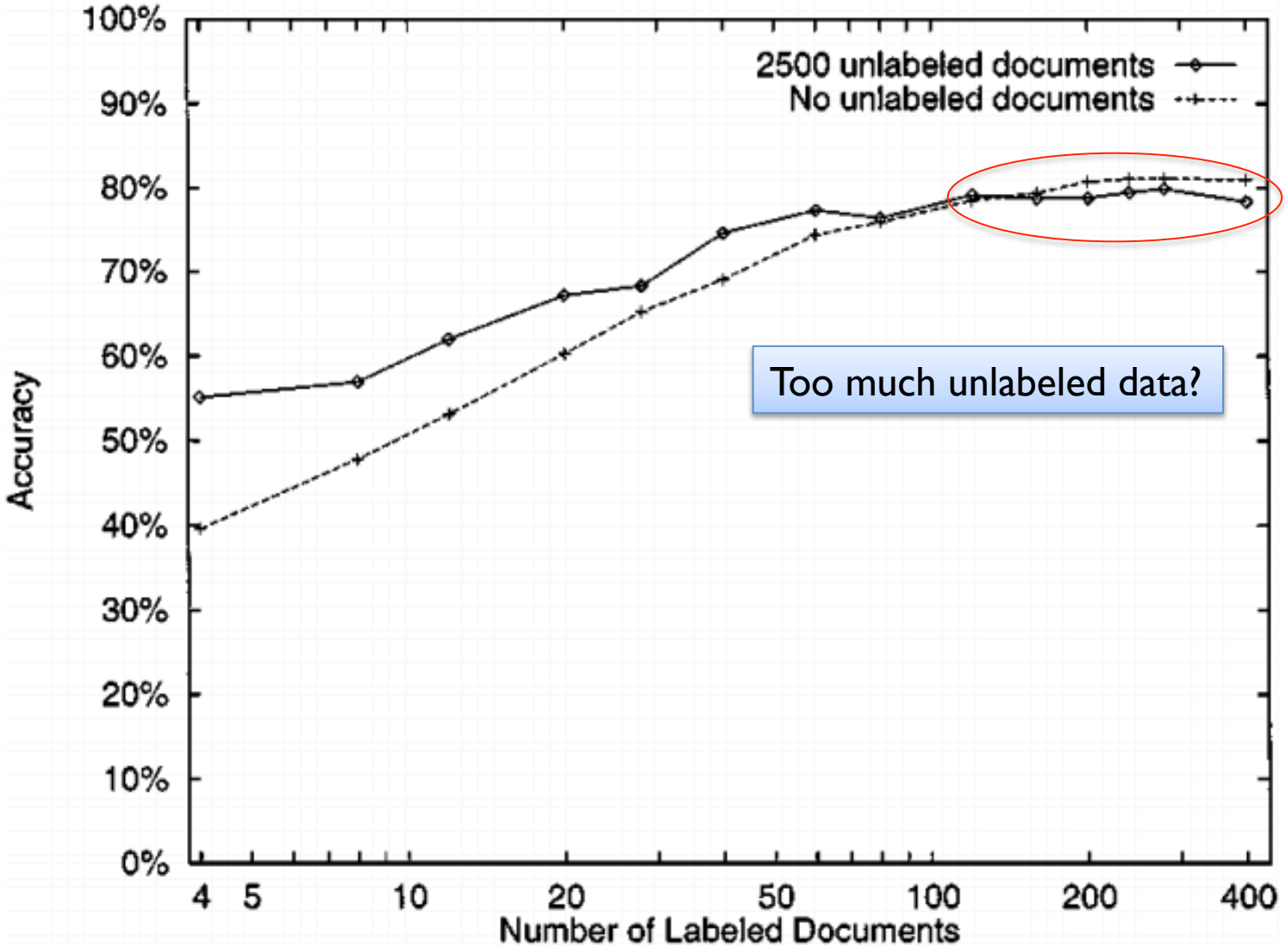
- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D} \mid \theta)\mathrm{P}(\theta)$ (see Eqs. (5) and (6)).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta \mid \mathcal{D}; \mathbf{z})$ (the complete log probability of the labeled and unlabeled data, and the prior) (see Eq. (10)).
  - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, i.e., the probability that each mixture component (and class) generated each document, $\mathrm{P}(c_j \mid d_i; \hat{\theta})$ (see Eq. (7)).
  - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D} \mid \theta)\mathrm{P}(\theta)$ (see Eqs. (5) and (6)).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and [illegible]licts a class label.

Might *downweight* the unlabeled data

Sample results with SSL



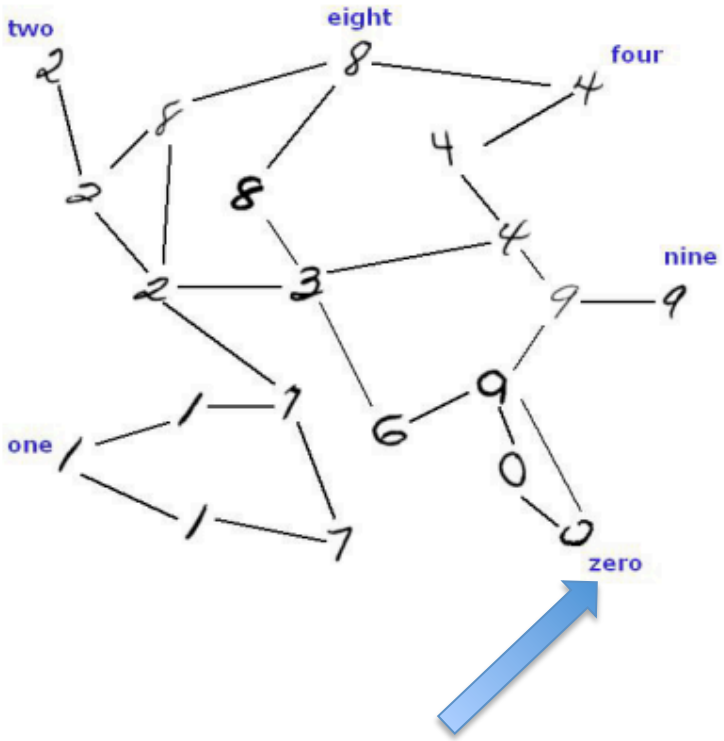
Sample results with SSL



SEMI-SUPERVISED LEARNING WITH GRAPHS: THE MAIN IDEA

Next few slides pilfered from Zoubin Ghahramani

"K-NN graph"

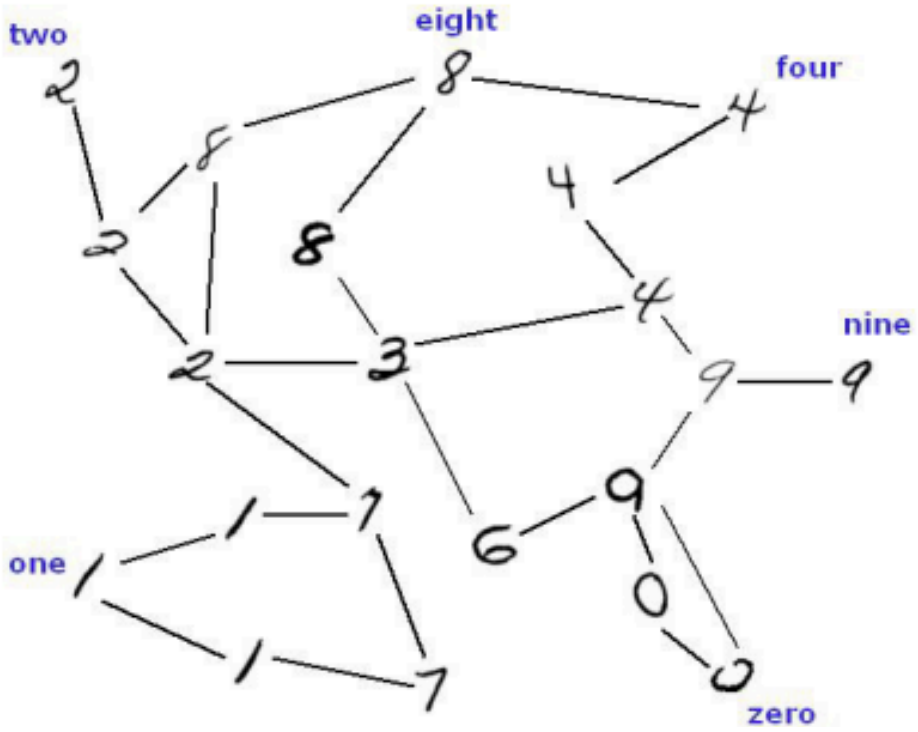


Observed label

Harmonic fields – with Lafferty and Zhu

- Idea: Construct a graph connecting similar data points
- Let the hidden/observed labels be random variables on the nodes of this graph (i.e. the graph is an MRF)
- Intuition: Similar data points have similar labels
- Information "propagates" from labeled data points
- Graph encodes intuition

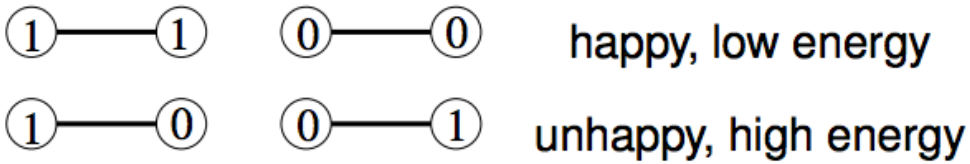
Harmonic fields – with Lafferty and Zhu



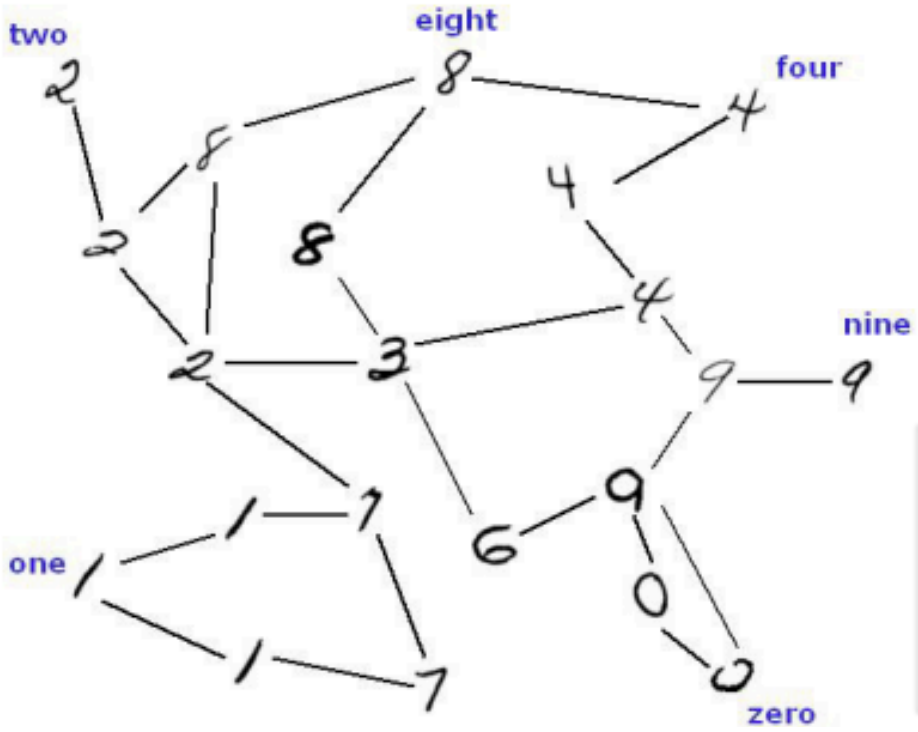
Problem: minimizing energy is **expensive**

GLZ solution: **relax** y's to be real numbers

energy: $E(\mathbf{y}) = \frac{1}{2}\sum_{i,j} w_{ij}\,(y_i - y_j)^2$



Harmonic fields – with Lafferty and Zhu

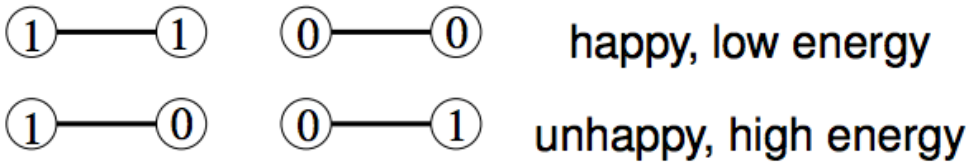


Solution: relax y's to be real numbers

At solution*:

$$\Delta \mathbf{f} = 0 \text{ or } f_i = \frac{\sum_{j \sim i} w_{ij} f_j}{\sum_{j \sim i} w_{ij}}, \; i \in U$$

energy: $E(\mathbf{y}) = \frac{1}{2} \sum_{i,j} w_{ij} (y_i - y_j)^2$



*ZGL also enforce a class prior

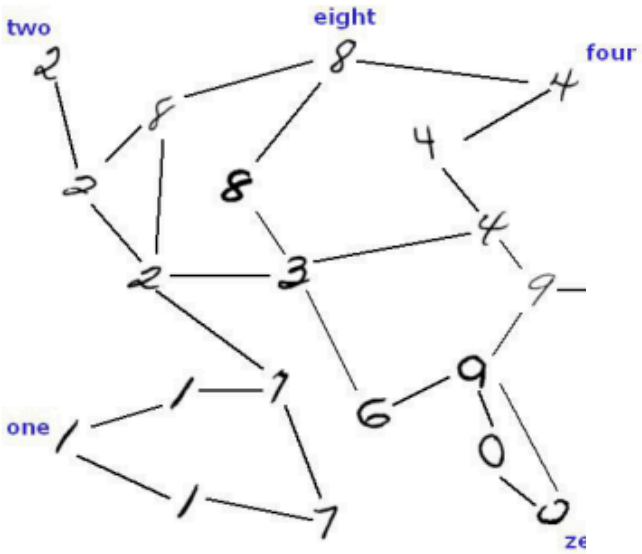
HF/CoEM/wvRN

- Breaking it down:
  - Step 1: For each seed example (x_i, y_i):
    - Let $V^0(i,c) = [\![y_i = c]\!]$
  - Step 2: for t=1,…,T --- *T is about 5*
    - Let $V^{t+1}(i,c)$ =weighted average of $V^{t+1}(j,c)$ for all j that are linked to i, and renormalize

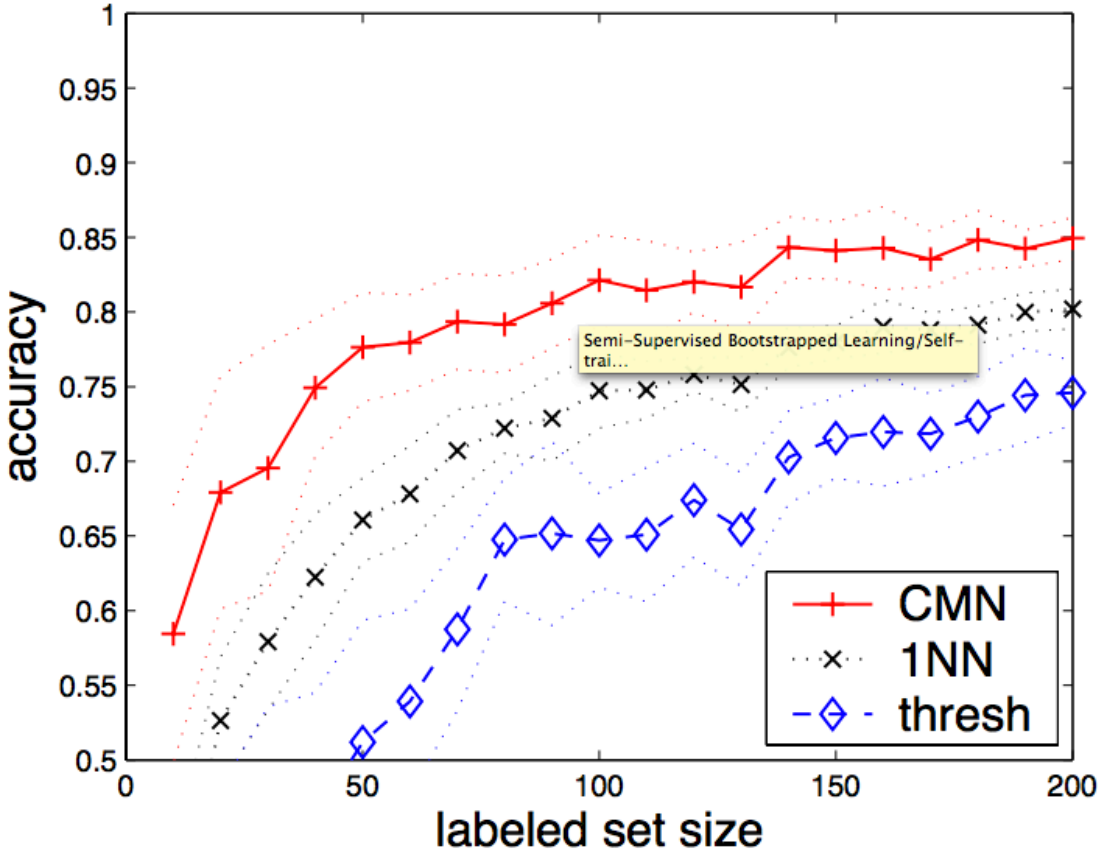
$$V^{t+1}(i,c) = \frac{1}{Z}\sum_j w_{i,j} V^t(j,c)$$

    - For seeds, reset $V^{t+1}(i,c) = [\![y_i = c]\!]$

Harmonic fields – with Lafferty and Zhu

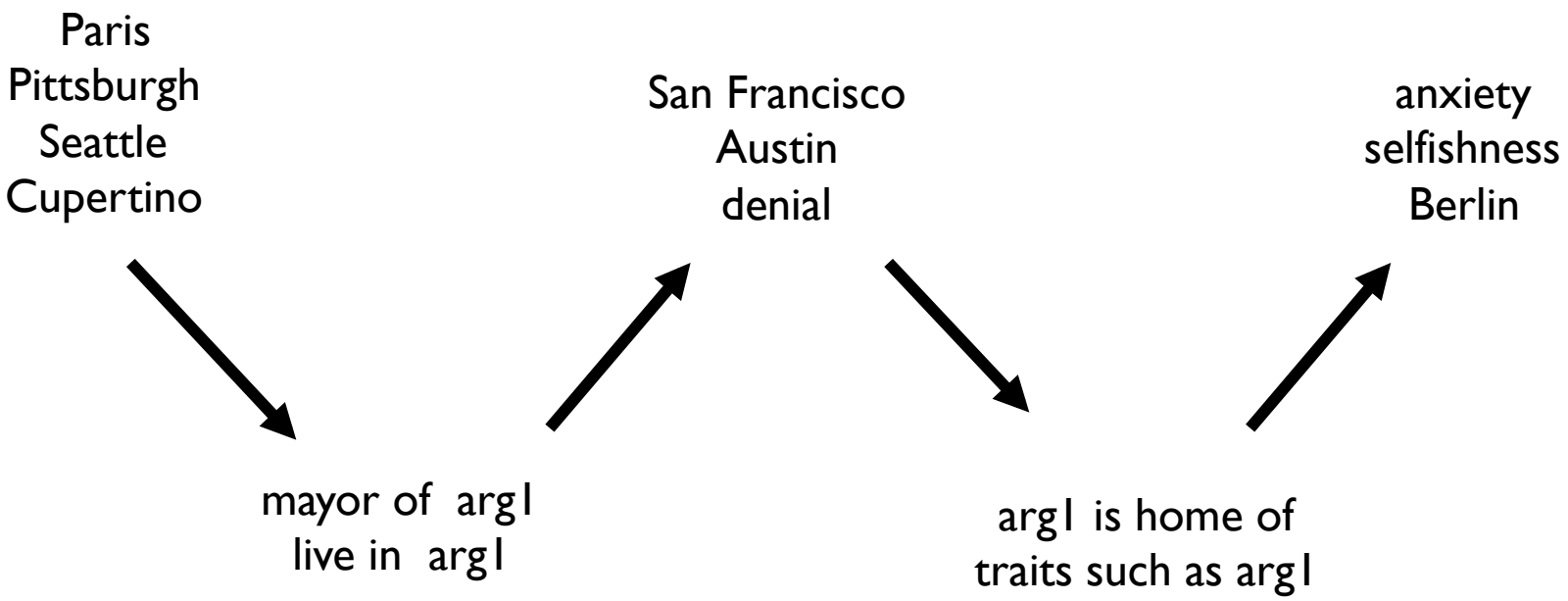


This family of techniques is called "Label propagation"

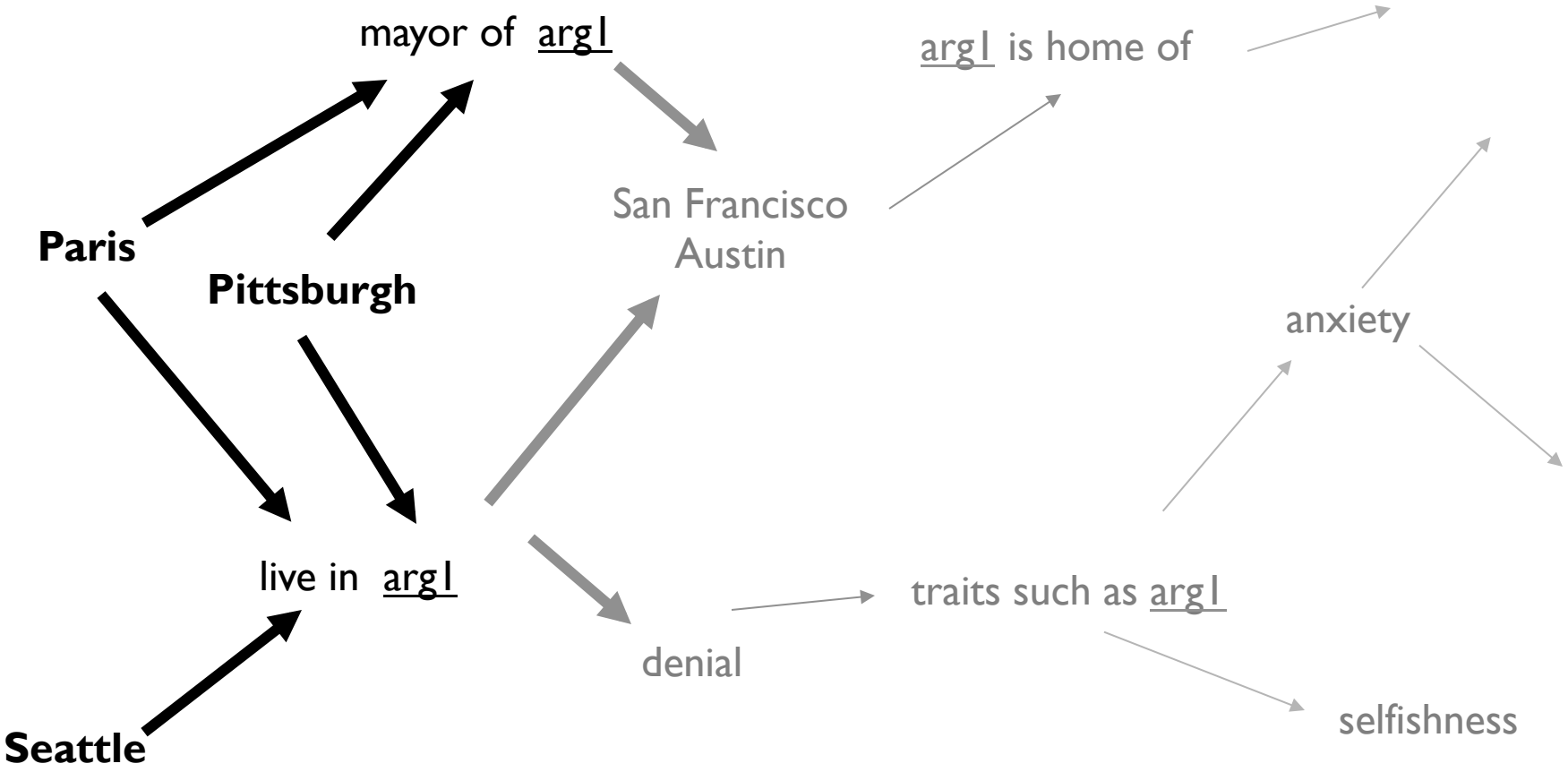


NELL: Uses Co-EM == HF

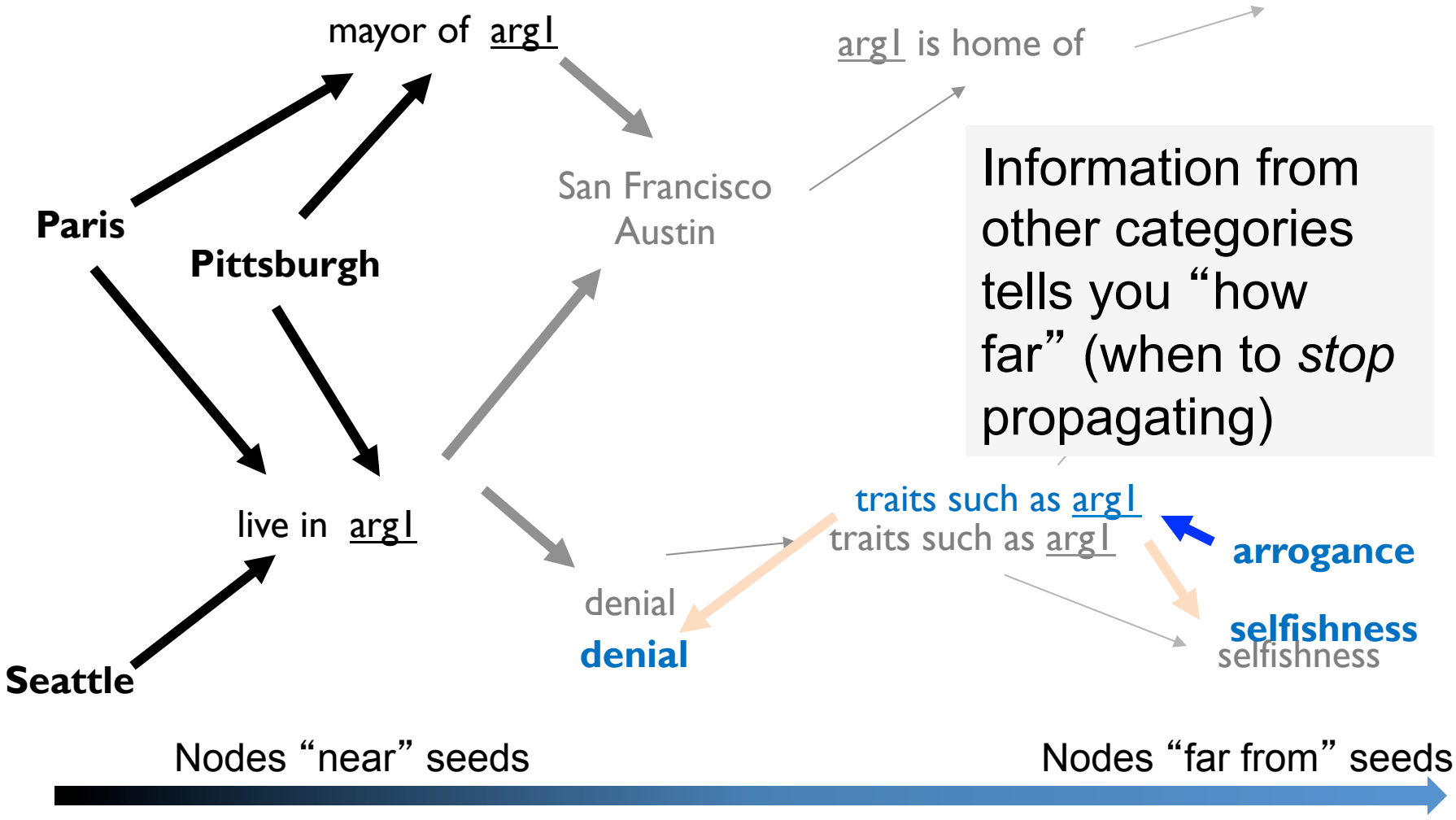
Extract cities:



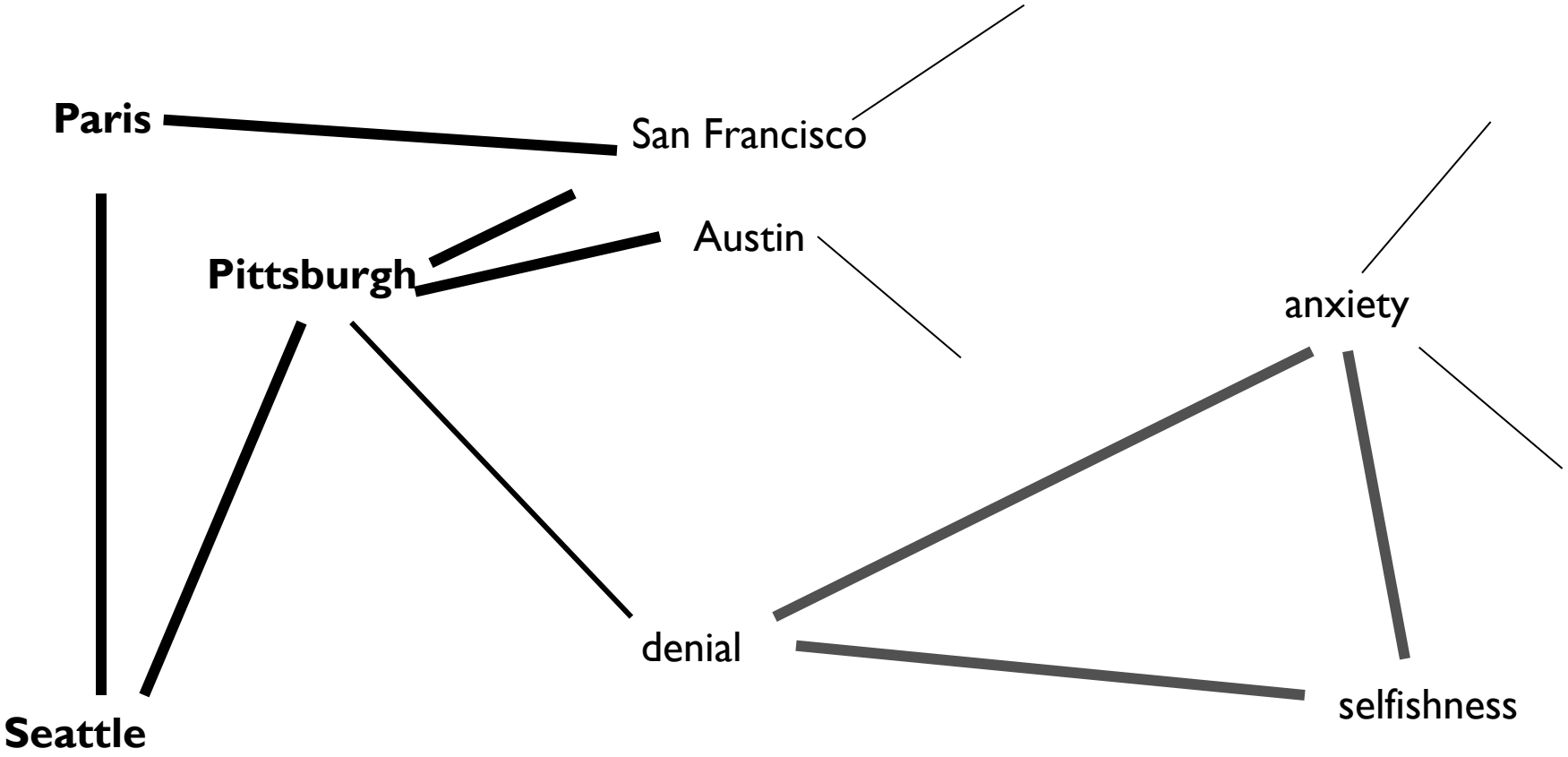
Semi-Supervised Bootstrapped Learning via Label Propagation



Semi-Supervised Bootstrapped Learning via Label Propagation



Difference: graph construction is not instance-to-instance but instance-to-feature



Two Kinds of Learning

- Inductive SSL:
  - Input: training set
    - $(x_1,y_1),\ldots,(x_n,y_n)$
    - $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$
  - Output: classifier
    - $f(x) = y$
  - Classifier can be run on any test example *x*

- Transductive SSL:
  - Input: training set
    - $(x_1,y_1),\ldots,(x_n,y_n)$
    - $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$
  - Output: classifier
    - $f(x_i) = y$
  - Classifier is only defined for x_i's *seen at training time*

Some Other Label Propagation Methods

MultiRankWalk

Semi-Supervised Classification of Network Data Using Very Few Labels

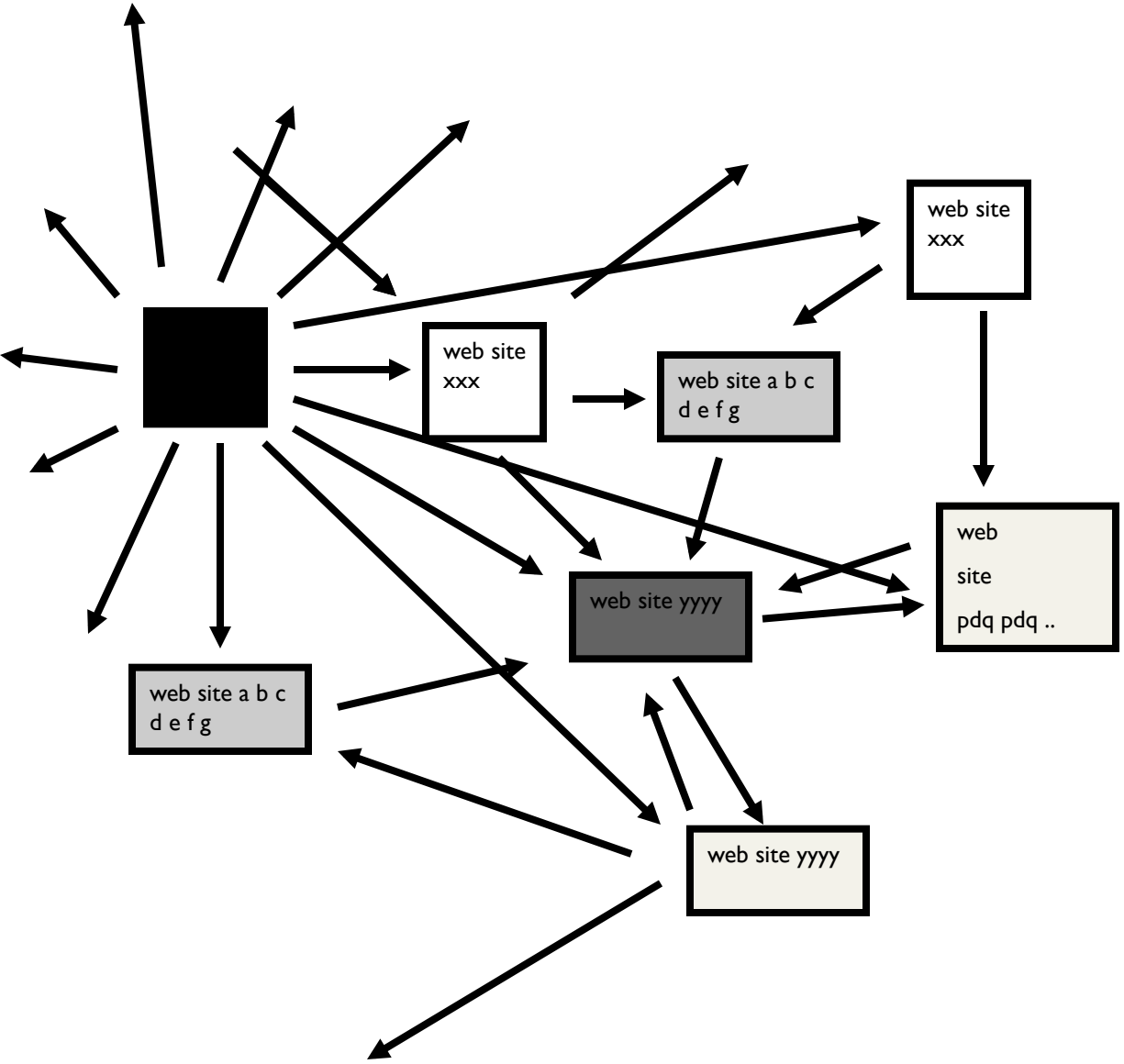
Frank Lin
Carnegie Mellon University, Pittsburgh, Pennsylvania
Email: frank@cs.cmu.edu

William W. Cohen
Carnegie Mellon University, Pittsburgh, Pennsylvania
Email: wcohen@cs.cmu.edu

Two related questions

- How do you propagate labels?
- How do you choose the seeds?
  - At random?
  - …?

Google's PageRank



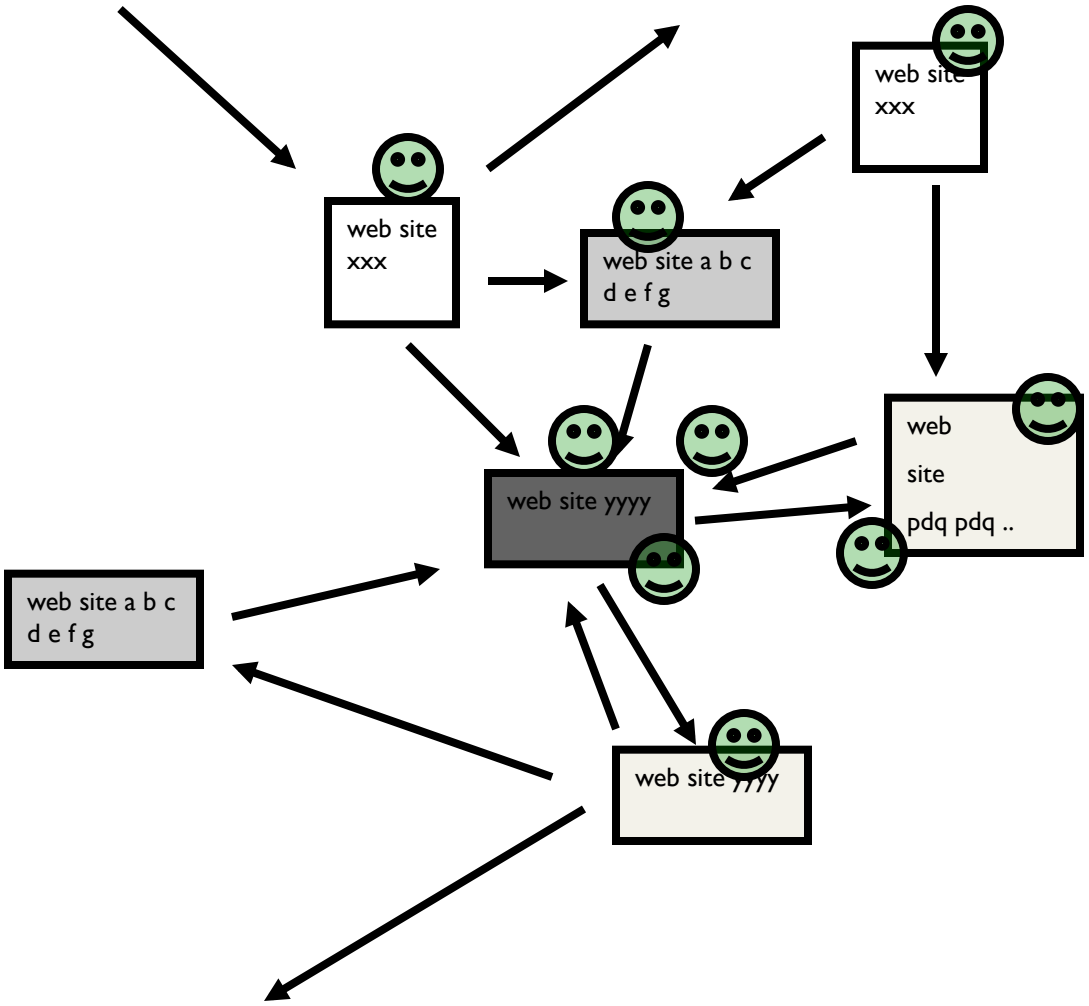
Inlinks are "good" (recommendations)

Inlinks from a "good" site are better than inlinks from a "bad" site

but inlinks from sites with many outlinks are not as "good"...

"Good" and "bad" are relative.

Google's PageRank

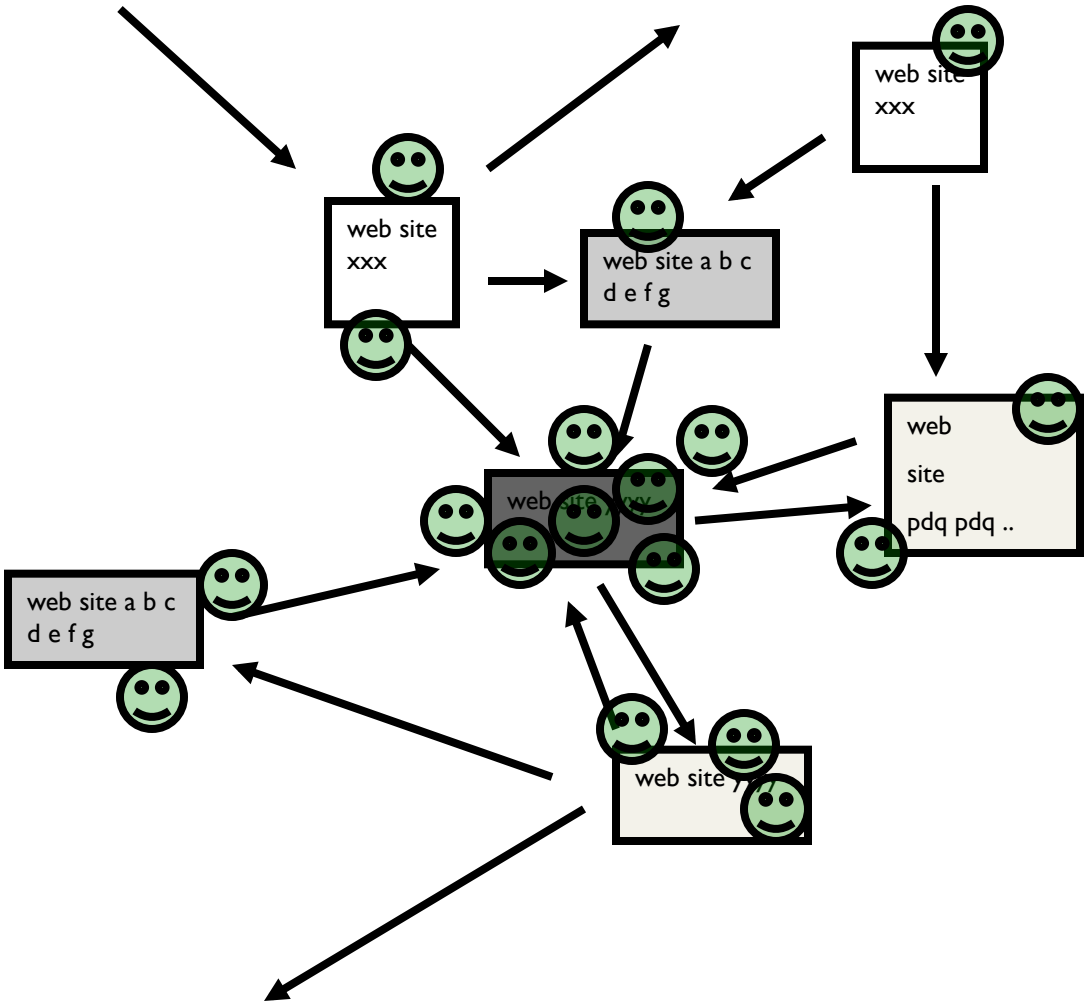


Imagine a "pagehopper" that always either

- follows a random link, or
- jumps to random page

Google's PageRank

(Brin & Page, http://www-db.stanford.edu/~backrub/google.html)



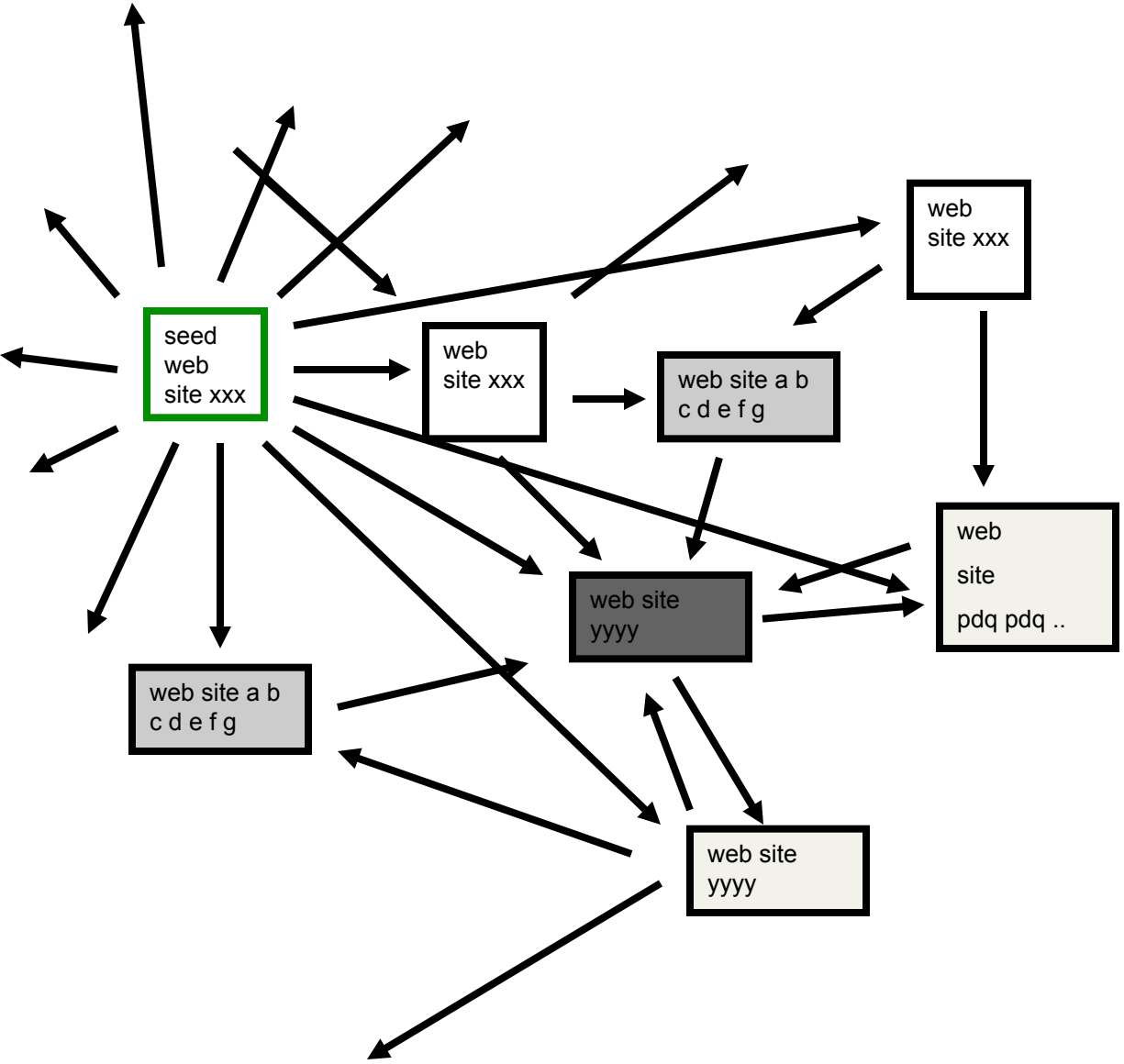
Imagine a "pagehopper" that always either

- follows a random link, or
- jumps to random page

PageRank ranks pages by the amount of time the pagehopper spends on a page:

- or, if there were many pagehoppers, PageRank is the expected "crowd size"

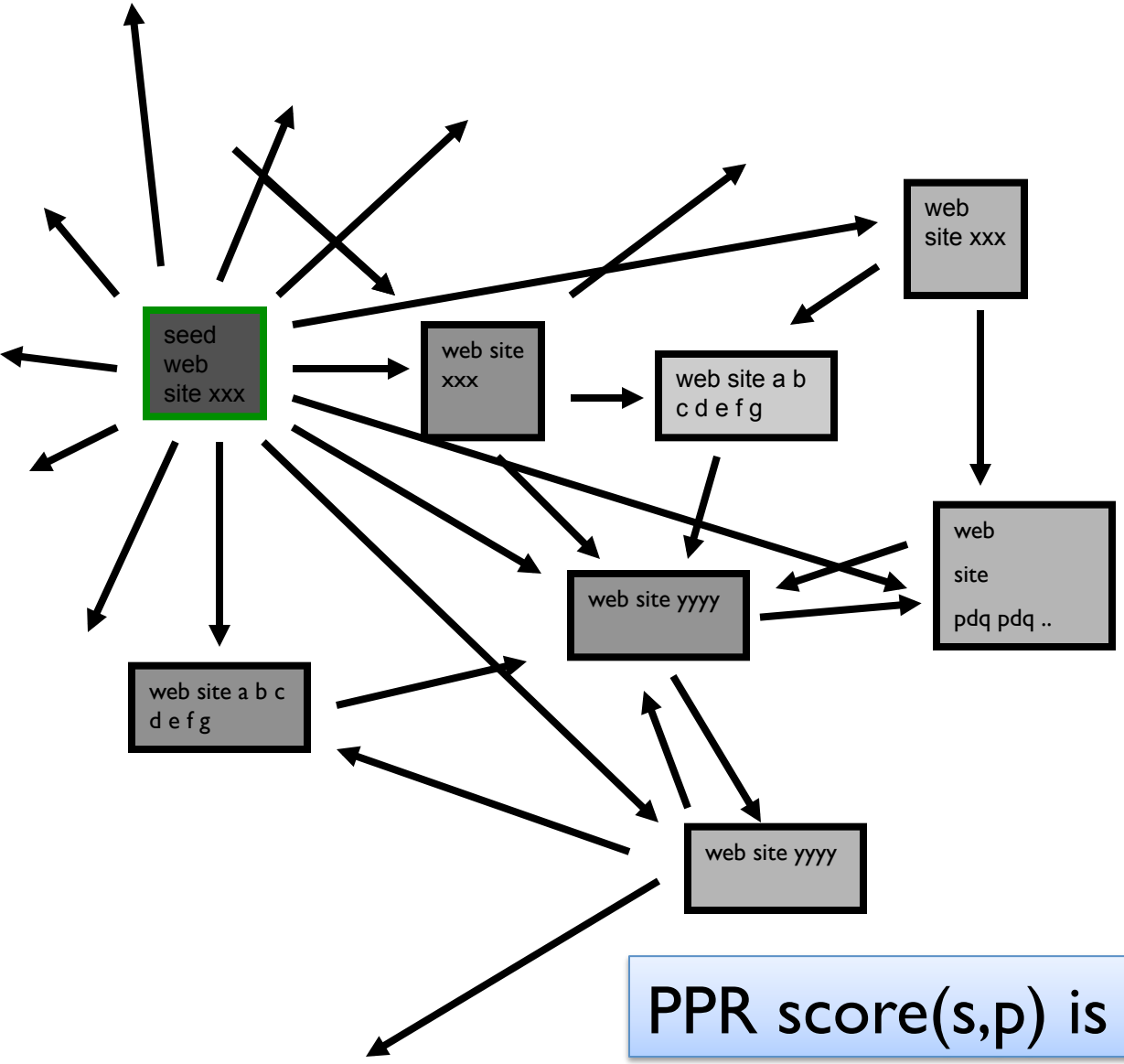
Personalized PageRank



Personalized PageRank (or Random walk with Reset): *same process*, except:

- when the PageHopper "jumps", it always jumps to a *particular page* (aka *resets* to a *seed page)*
- page score of p *relative to seed s* is the ssp of being at a page p *with resets to s*
- ➔ score(s,p)

Personalized PageRank



Personalized PageRank (or Random walk with Reset): *same process*, except:

- when the PageHopper "jumps", it always jumps to a *particular page* (aka *resets* to a *seed page)*
- page score of p relative to seed s is the ssp of being at a page p
- ➔ score(s,p) is a *closeness of* s and p

PPR score(s,p) is *closeness of* s and p

PageRank

- Let $\mathbf{r} = (1/N, \ldots, 1/N)$
  - dimension = #nodes N
- Let A = adjacency matrix: $[a_{ij}=1 \Leftrightarrow$ i links to j$]$
- Let $W = [w_{ij} = a_{ij}/\text{outdegree}(i)]$
  - w_{ij} is probability of jump from i to j
- Let $\mathbf{v}^0 = (1,1,\ldots,1)$ or whatever else you like
- Repeat until converged:
  - Let $\mathbf{v}^{t+1} = c\mathbf{r} + (1\text{-}c)\mathbf{W}\mathbf{v}^t$
    - c is probability of jumping "anywhere randomly"

Personalized PageRank/RWR

- Let $\mathbf{r}$ = *any row vector of probabilities*
  - dimension = #nodes I

e.g. r(s)=1 only for seed s

- Let A = adjacency matrix: [a_{ij}=1 ⇔ i links to j]
- Let W = [$w_{ij} = a_{ij}$/outdegree(i)]
  - w_{ij} is probability of jump from i to j
- Let $\mathbf{v}^0$ = (1,1,….,1) or whatever else you like
- Repeat until converged:
  - Let $\mathbf{v}^{t+1} = c\mathbf{r} + (1\text{-}c)\mathbf{W}\mathbf{v}^t$
    - c is probability of reset
- After convergence $\mathbf{v}(i)$ is similarity of x_i to s

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

RandomWalk: fixpoint of:

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

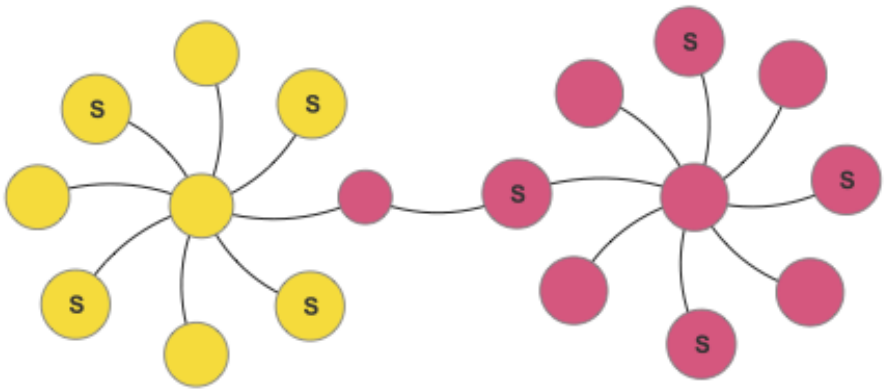
Fig. 1. The MultiRankWalk algorithm.

Seed selection

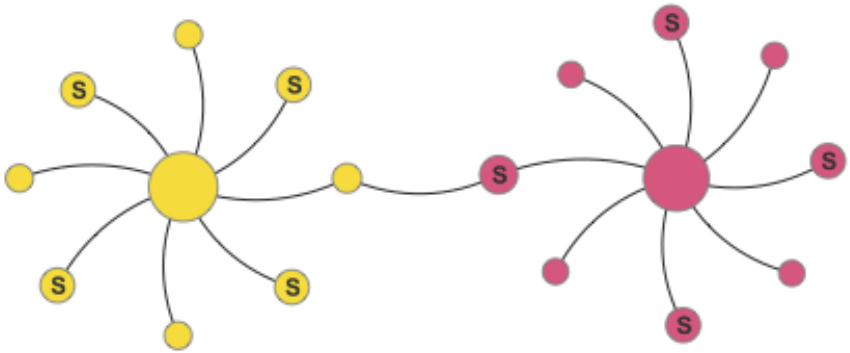
1. order by PageRank, degree, or randomly
2. go down list until you have at least *k* examples/class

MultiRank Walk vs HF/wvRN/CoEM

Seeds are marked S

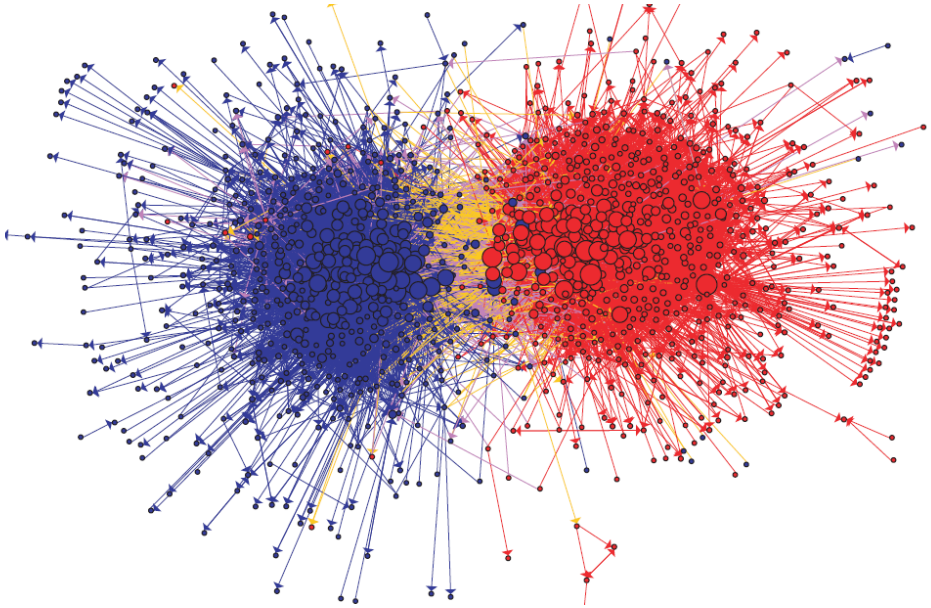


HF

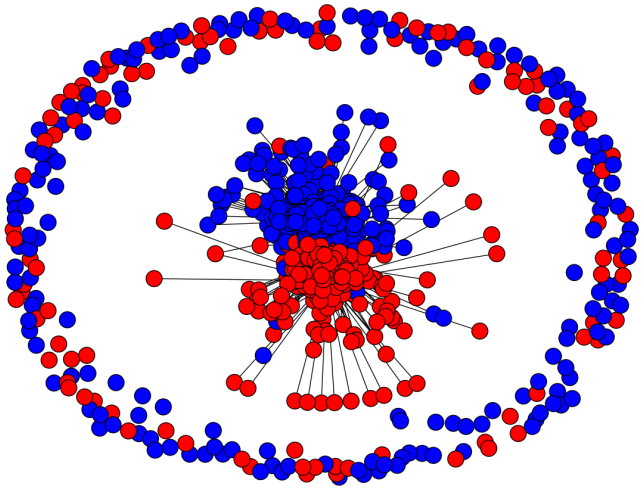


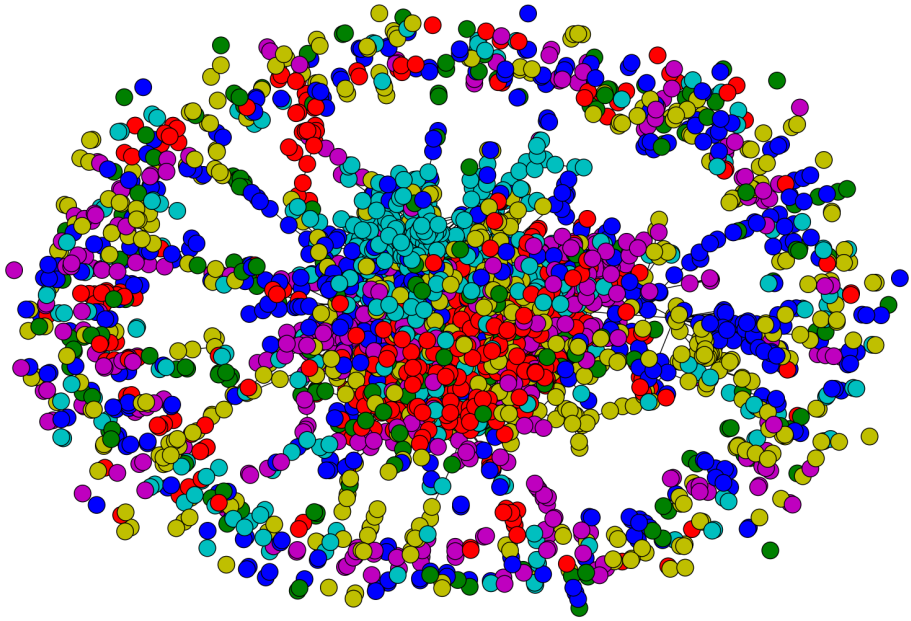
MRW

SSL on Network Datasets



- UBMCBlog
- AGBlog
- MSPBlog
- Cora
- Citeseer



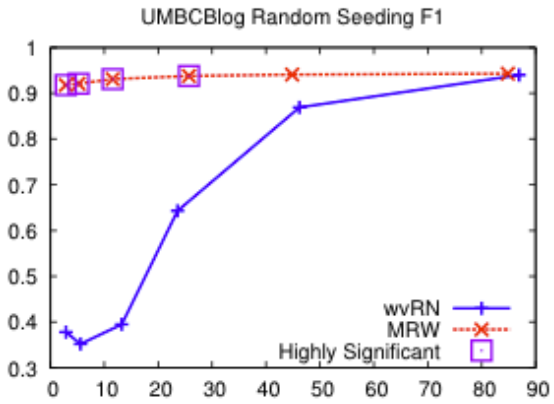


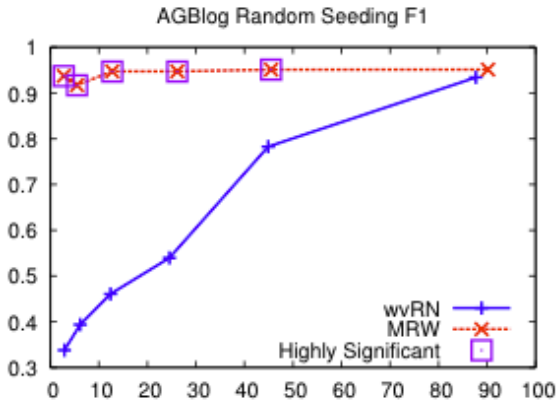
Results – Blog data

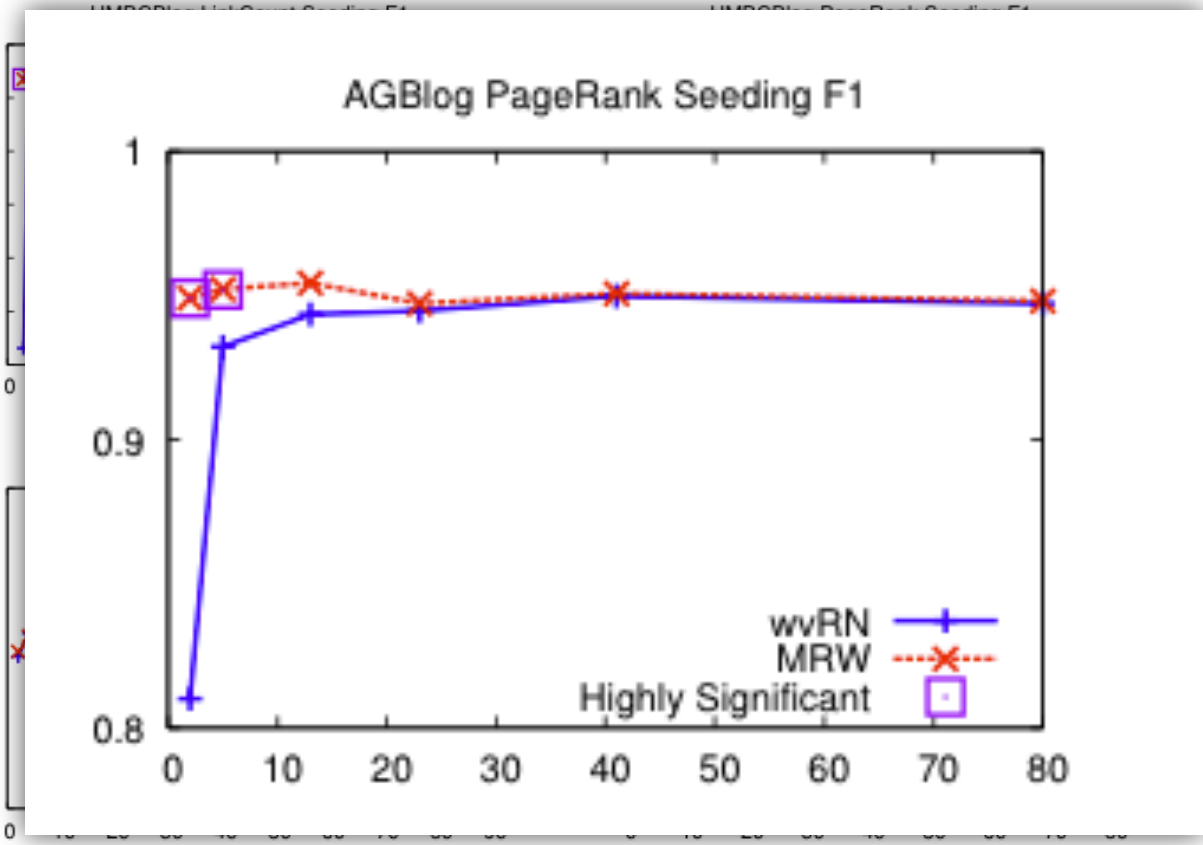
Random

Degree

PageRank

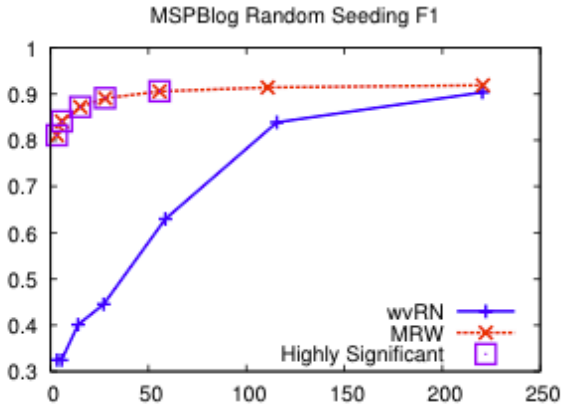




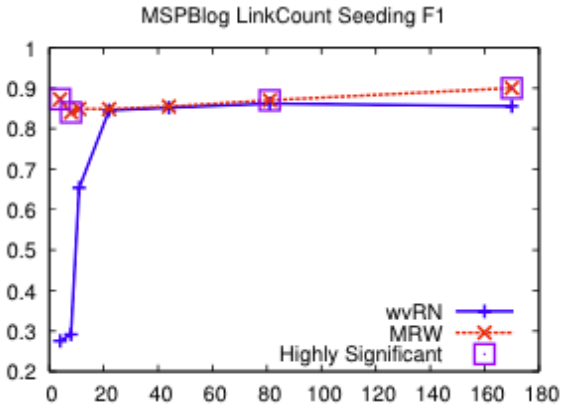


Results – More blog data

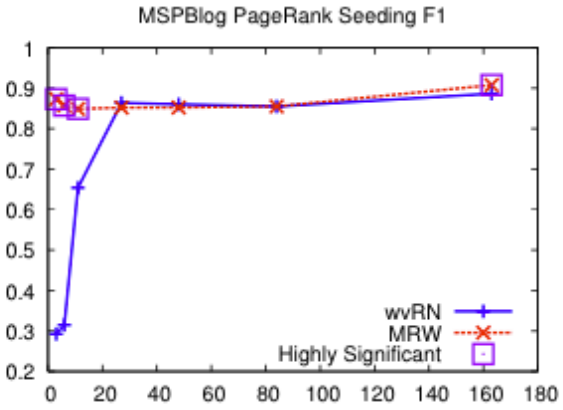
Random



Degree

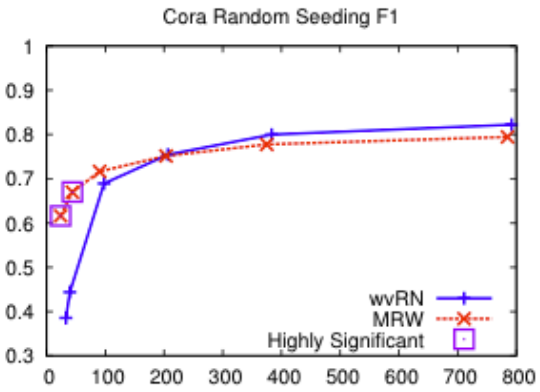


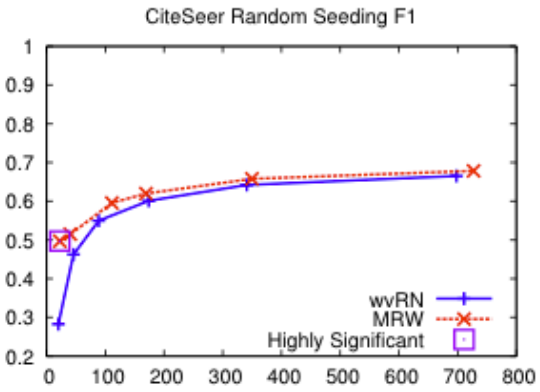
PageRank



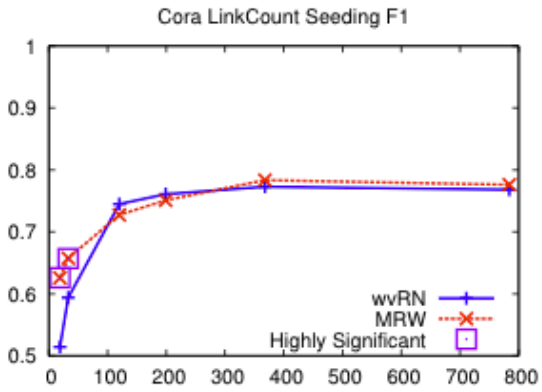
Results – Citation data

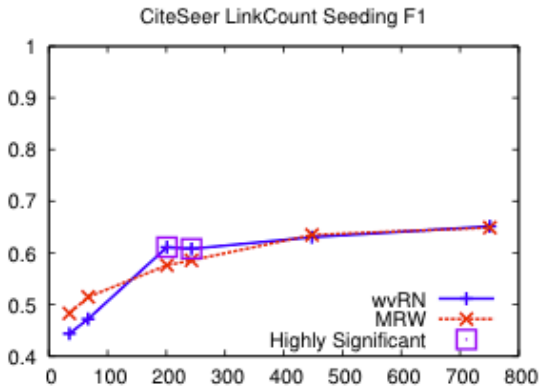
Random



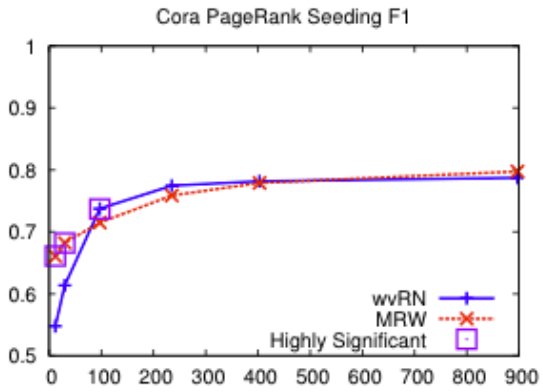


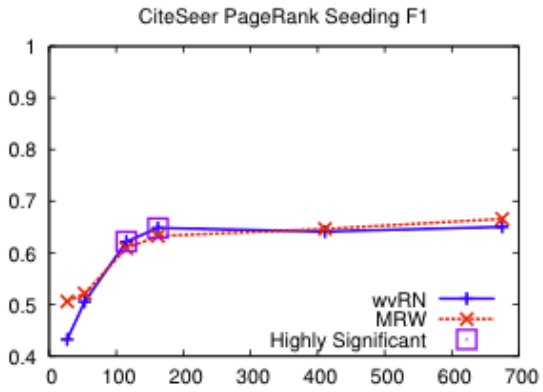
Degree



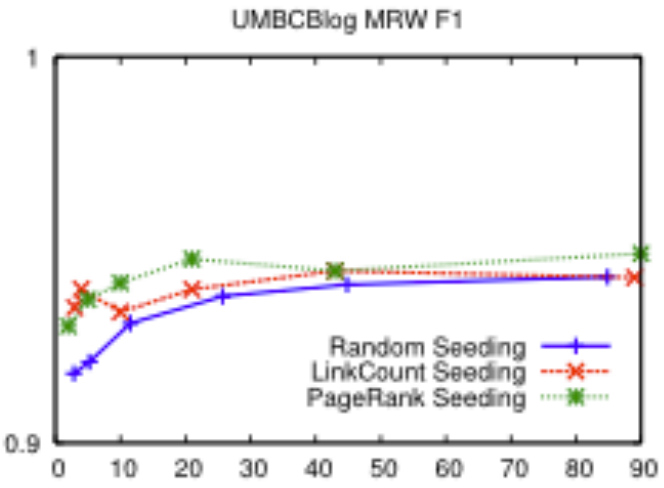


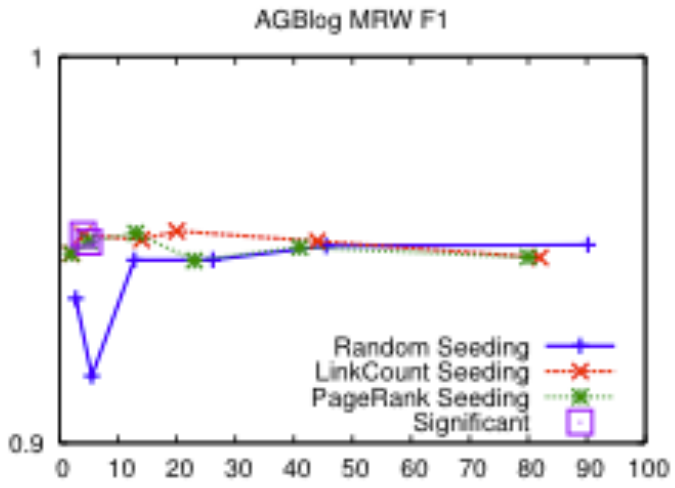
PageRank

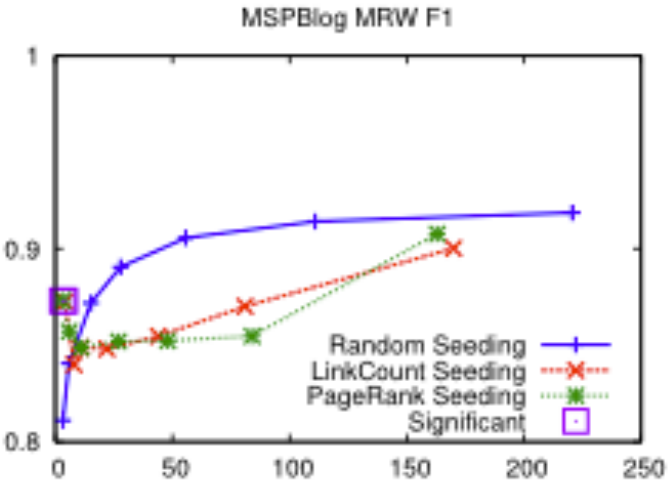


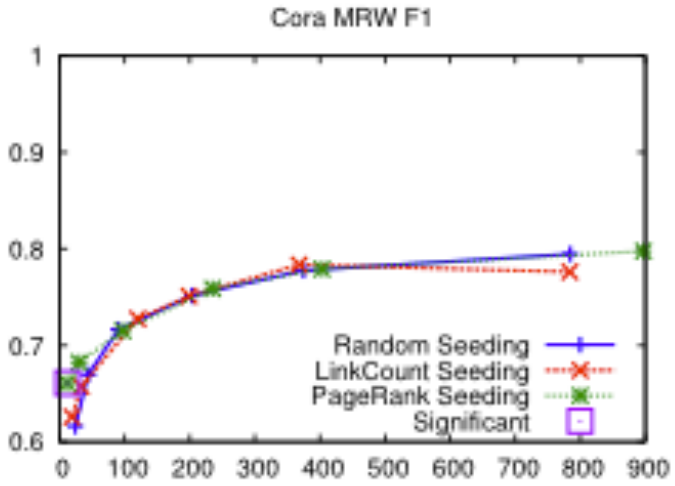


Seeding – MultiRankWalk

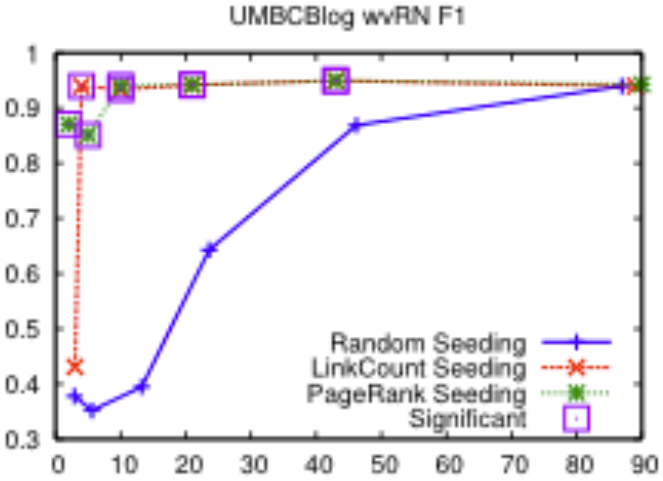


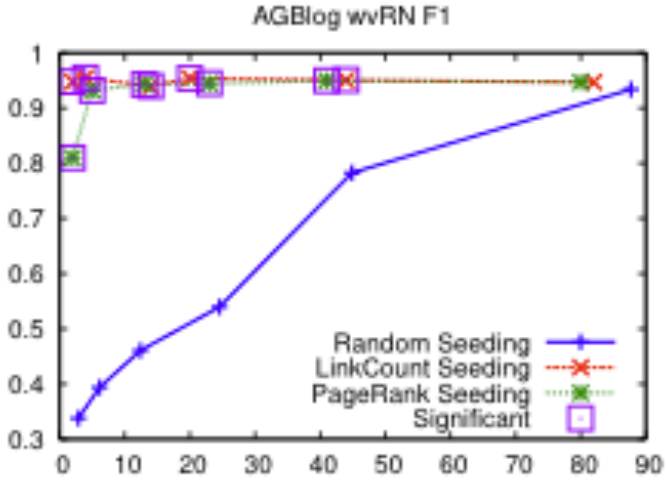


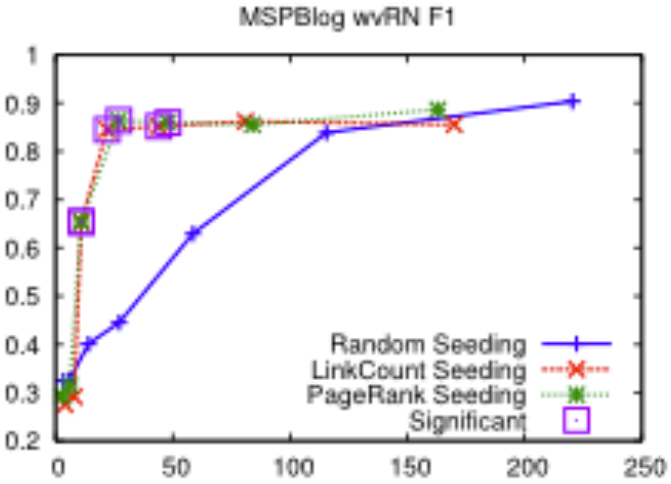


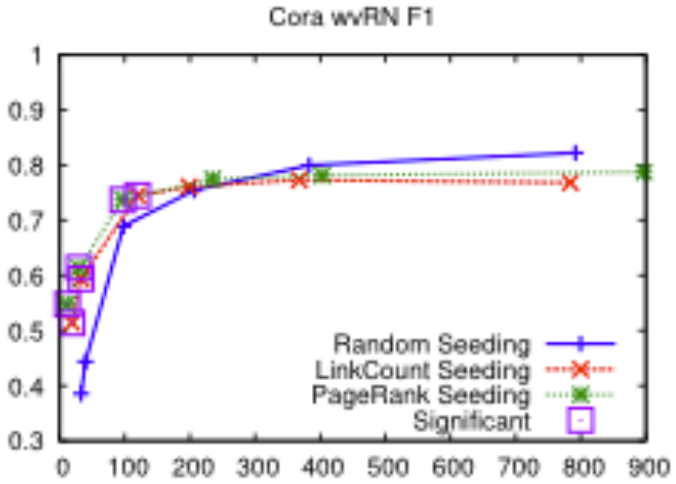


Seeding – HF/wvRN









MultiRankWalk vs wvRN/HF/CoEM

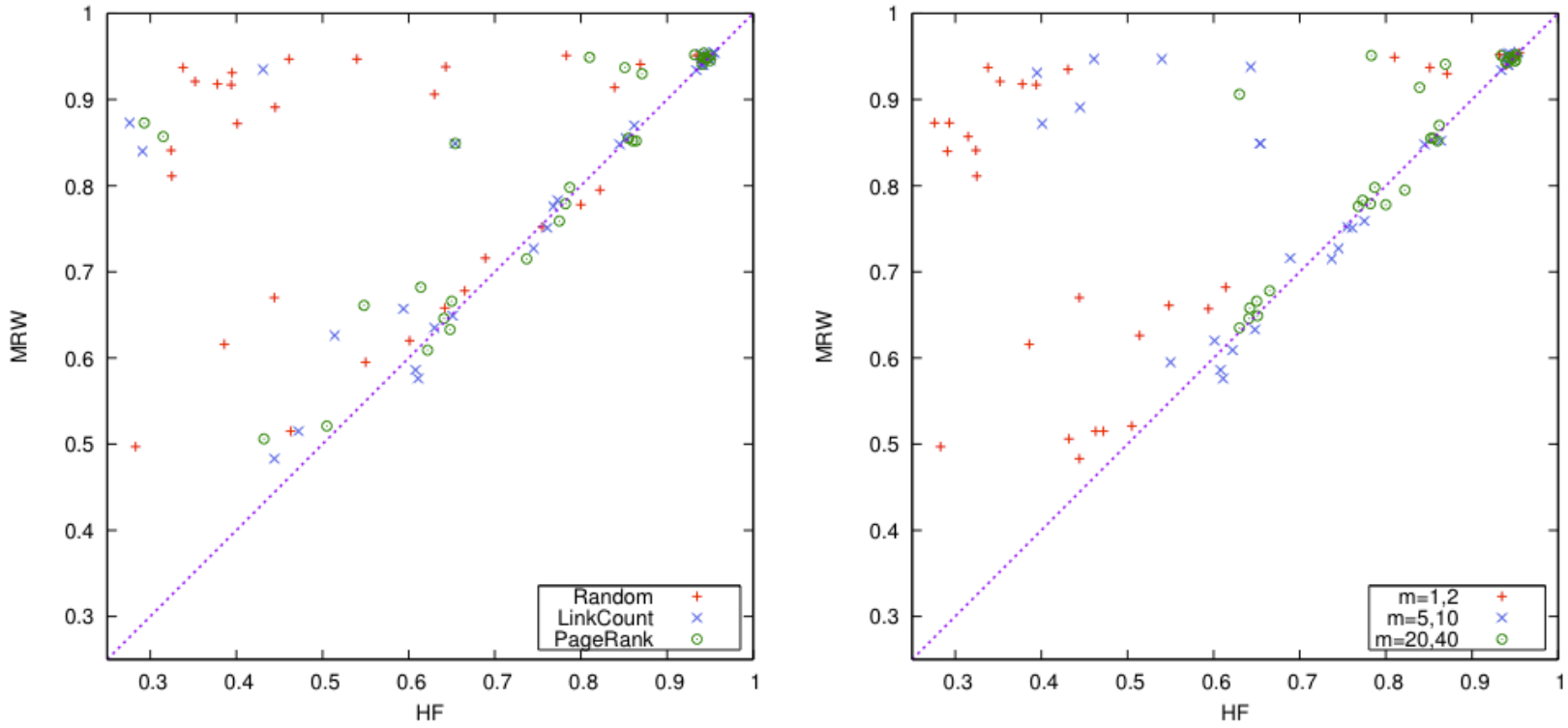
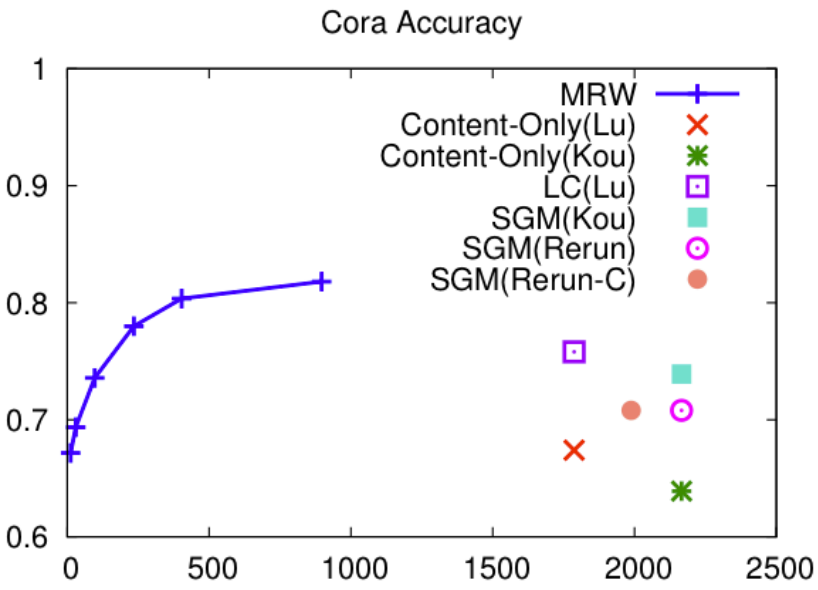


Figure 2.6: Scatter plots of HF F1 score versus MRW F1 score. The left plot marks different seeding preferences and the right plot marks varying amount of training labels determined by m.

How well does graph-based SSL work?



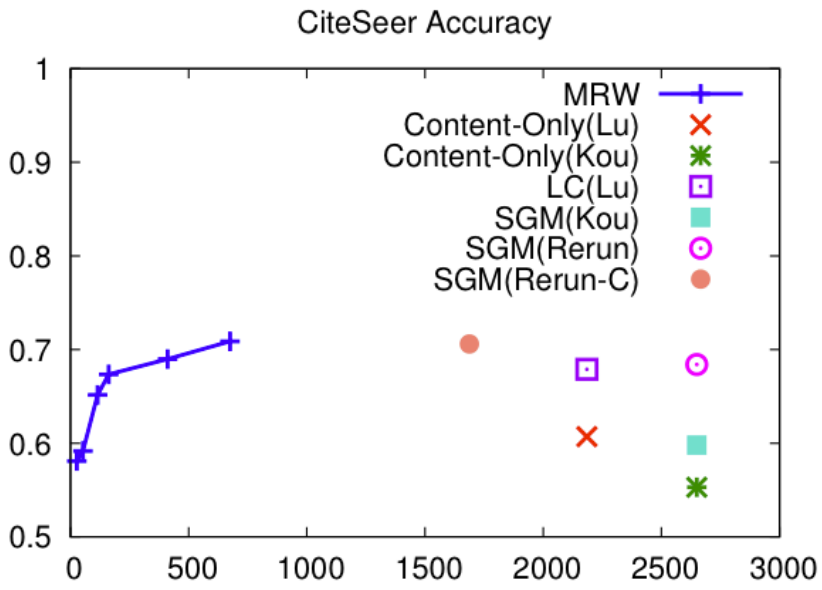


Fig. 5. Citation datasets results compared to supervised relational learning methods. The x-axis indicates number of labeled instances and y-axis indicates labeling accuracy.

Some Other Label Propagation Methods

Modified Adsorption

Partha Talukdar

Notations

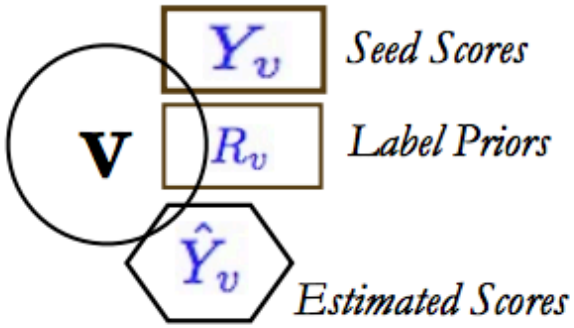
$\hat{Y}_{v,l}$: score of estimated label l on node v

$Y_{v,l}$: 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)

Smooth

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

Graph Laplacian
L = D - W (PSD)

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

Match Seeds (hard)

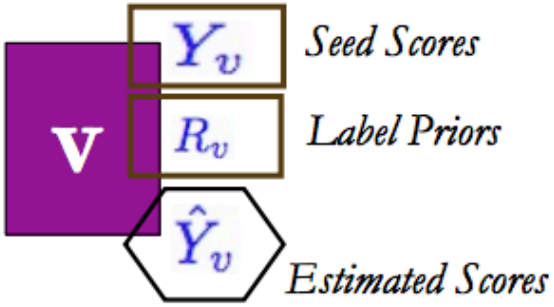
- Smoothness
  - two nodes connected by an edge with high weight should be assigned similar labels
- Solution satisfies harmonic property

Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

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

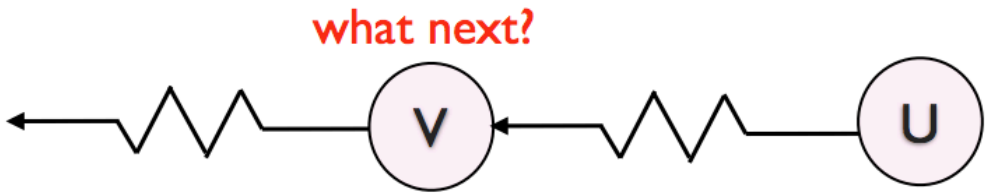
- m labels, +1 dummy label
- $\boldsymbol{M} = \boldsymbol{W}^{\prime\top} + \boldsymbol{W}^{\prime}$ is the symmetrized weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for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



Let's talk about W and R…..

- $\boldsymbol{M} = \boldsymbol{W}^{\dagger} + \boldsymbol{W}'$ is the symmetrized weight matrix

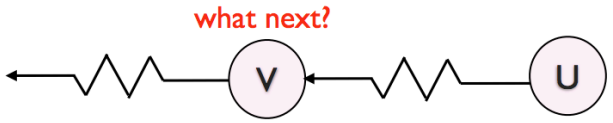
Random Walk View



- Continue walk with prob. $\mathbf{p_v^{cont}}$
- Assign V's seed label to U with prob. $\mathbf{p_v^{inj}}$
- Abandon random walk with prob. $\mathbf{p_v^{abnd}}$
  - assign U a dummy label

- $M = W^{\top} + W'$ is the symmetrized weight matrix

Random Walk View



- Continue walk with prob. p_v^{cont}
- Assign V's seed label to U with prob. p_v^{inj}
- Abandon random walk with prob. p_v^{abnd}
  - assign U a dummy label

New Edge Weight

$$W'_{uv} = p_u^{cont} \times W_{uv}$$

$$S_{uu} = \sqrt{p_u^{inj}}$$

$$R_{u\top} = p_u^{abnd}$$, and 0 for non-dummy labels

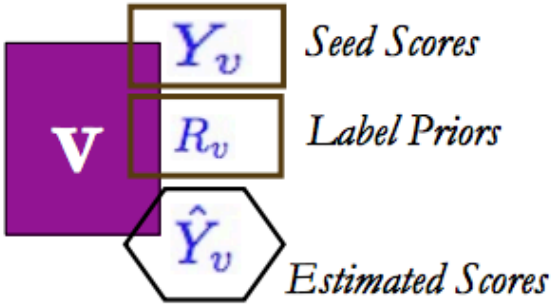
Dummy Label

Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

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

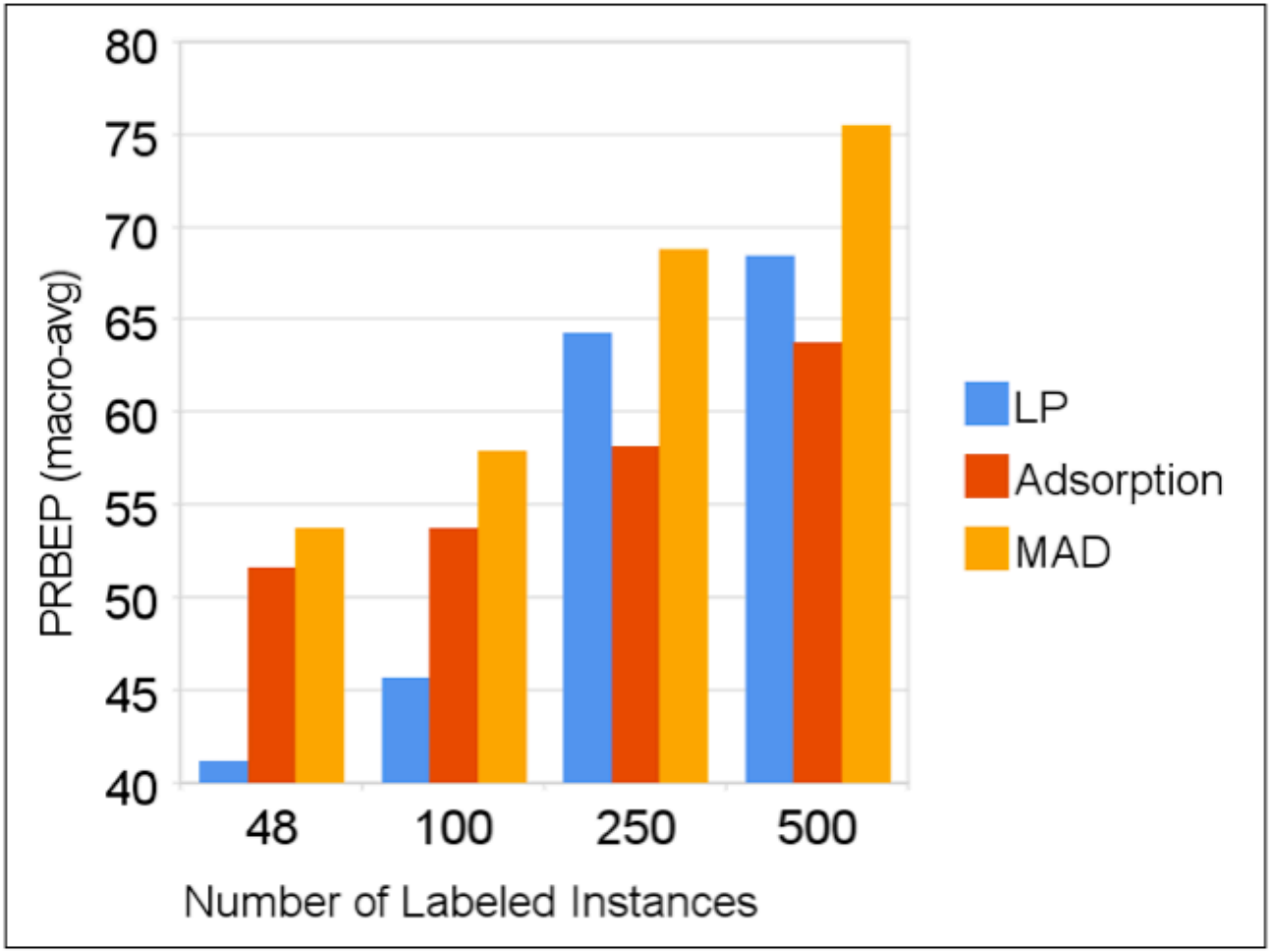
- m labels, $+1$ dummy label
- $\boldsymbol{M} = \boldsymbol{W}^{\prime\top} + \boldsymbol{W}^{\prime}$ is the symmetrized weight matrix
- $\hat{\boldsymbol{Y}}_{vl}$: weight of label l on node v
- $\boldsymbol{Y}_{vl}$: seed weight for label l on node v
- $\boldsymbol{S}$: diagonal matrix, nonzero for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



Inputs $\boldsymbol{Y}, \boldsymbol{R} : |V| \times (|L|+1)$, $\boldsymbol{W} : |V| \times |V|$, $\boldsymbol{S} : |V| \times |V|$ diagonal
$\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{Y}$
$\boldsymbol{M} = \boldsymbol{W}' + \boldsymbol{W}'^{\top}$
$Z_v \leftarrow \boldsymbol{S}_{vv} + \mu_1 \sum_{u \neq v} \boldsymbol{M}_{vu} + \mu_2 \quad \forall v \in V$
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

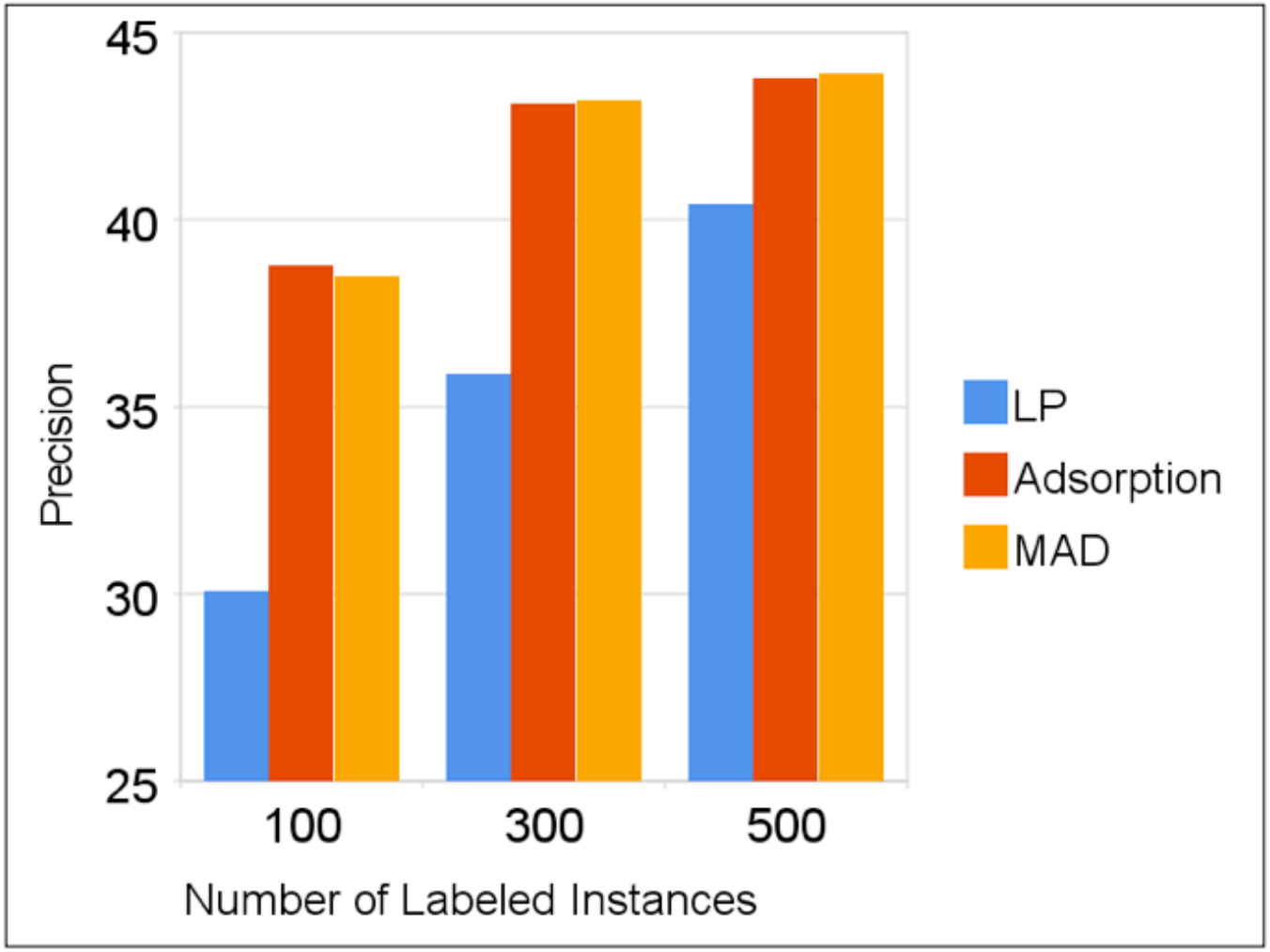
- Extends Adsorption with well-defined optimization
- Importance of a node can be discounted
- Easily Parallelizable: Scalable

Text Classification



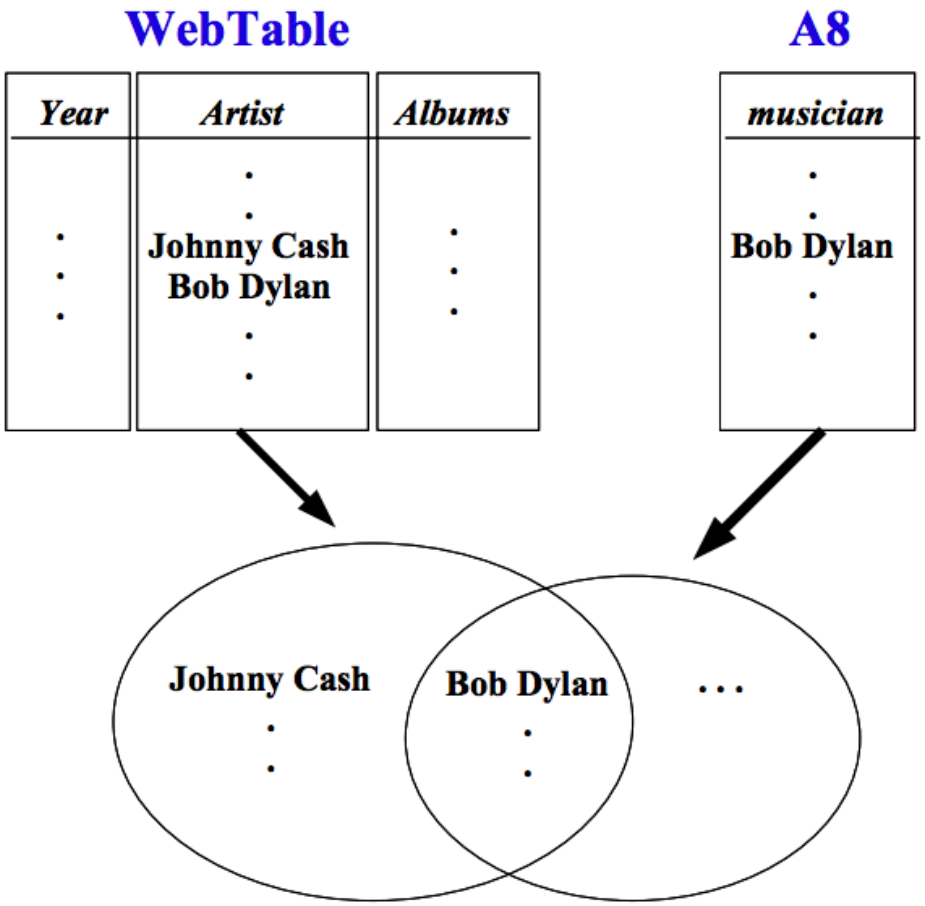
PRBEP (macro-averaged) on WebKB Dataset, 3148 test instances

Sentiment Classification

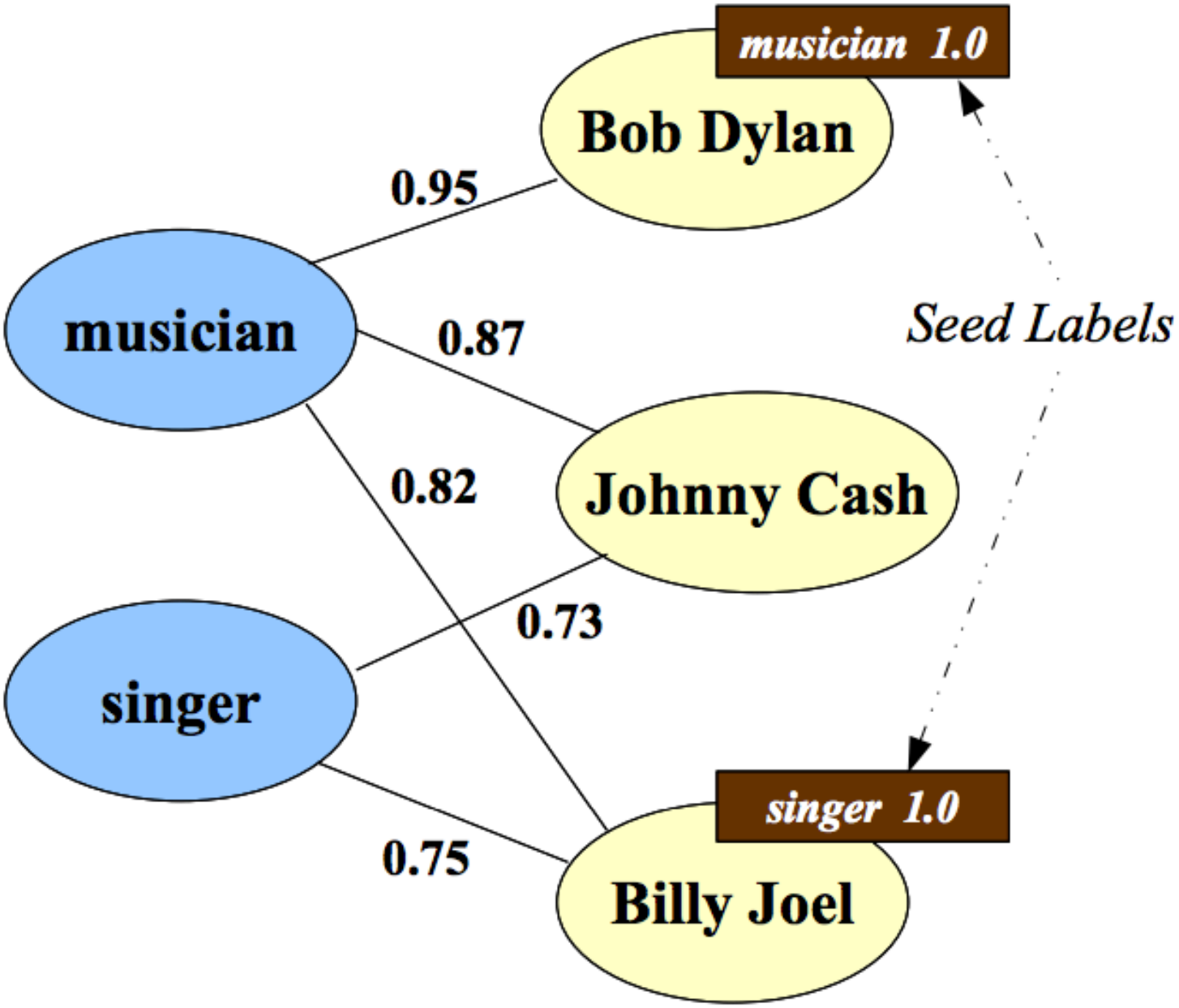


Precision on 3568 Sentiment test instances

Assigning class labels to WebTable instances



Score (musician, Johnny Cash) = 0.87

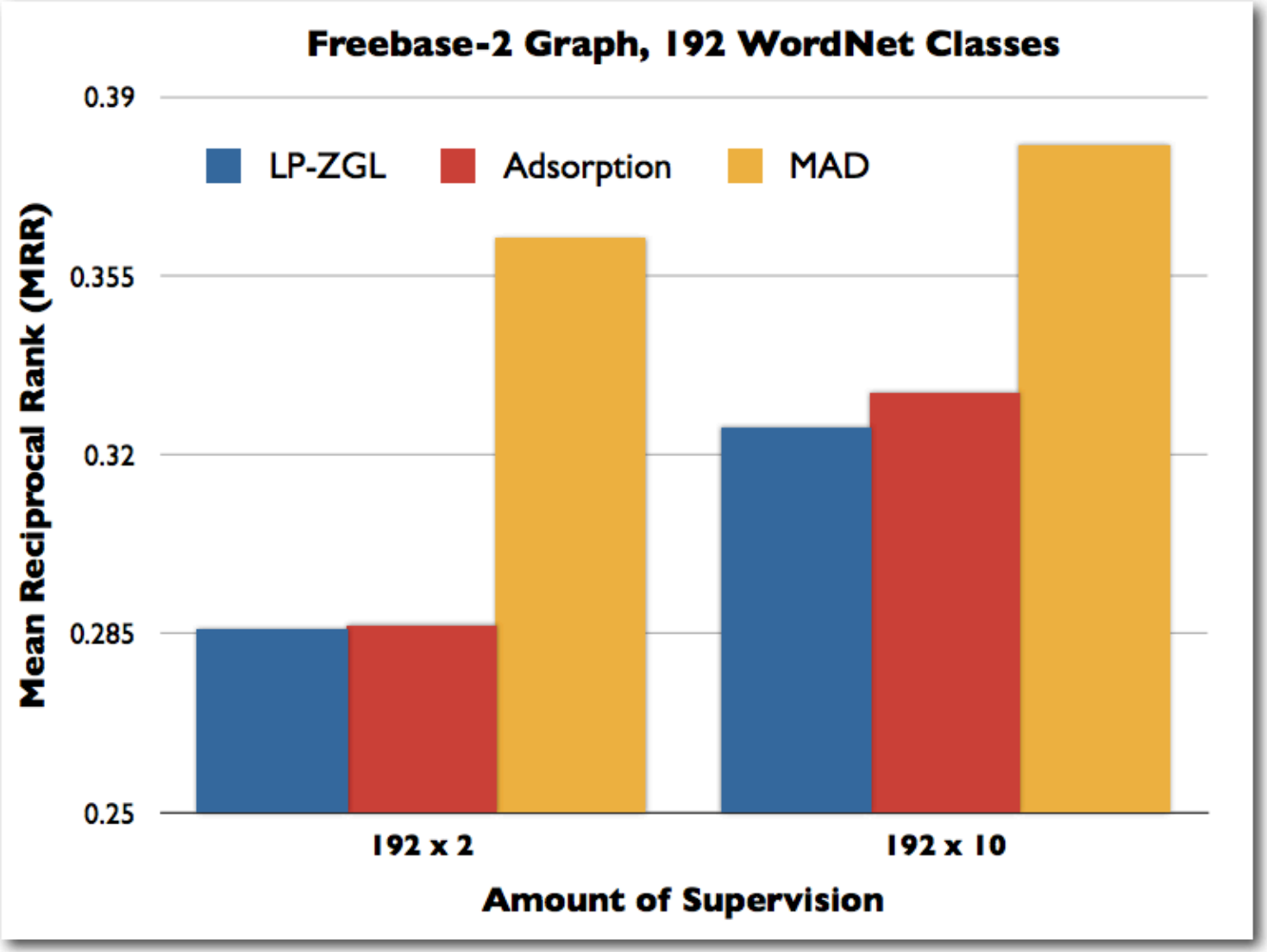

musician 1.0
Bob Dylan
0.95
musician
0.87
Seed Labels
0.82
Johnny Cash
0.73
singer
singer 1.0
0.75
Billy Joel

New (Class, Instance) Pairs Found

Class	A few non-seed Instances found by Adsorption
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, …

Total classes: **9081**

Class-Instance Acquisition



Graph with 303k nodes, 2.3m edges.