Semi-supervised learning
SSL (on graphs)

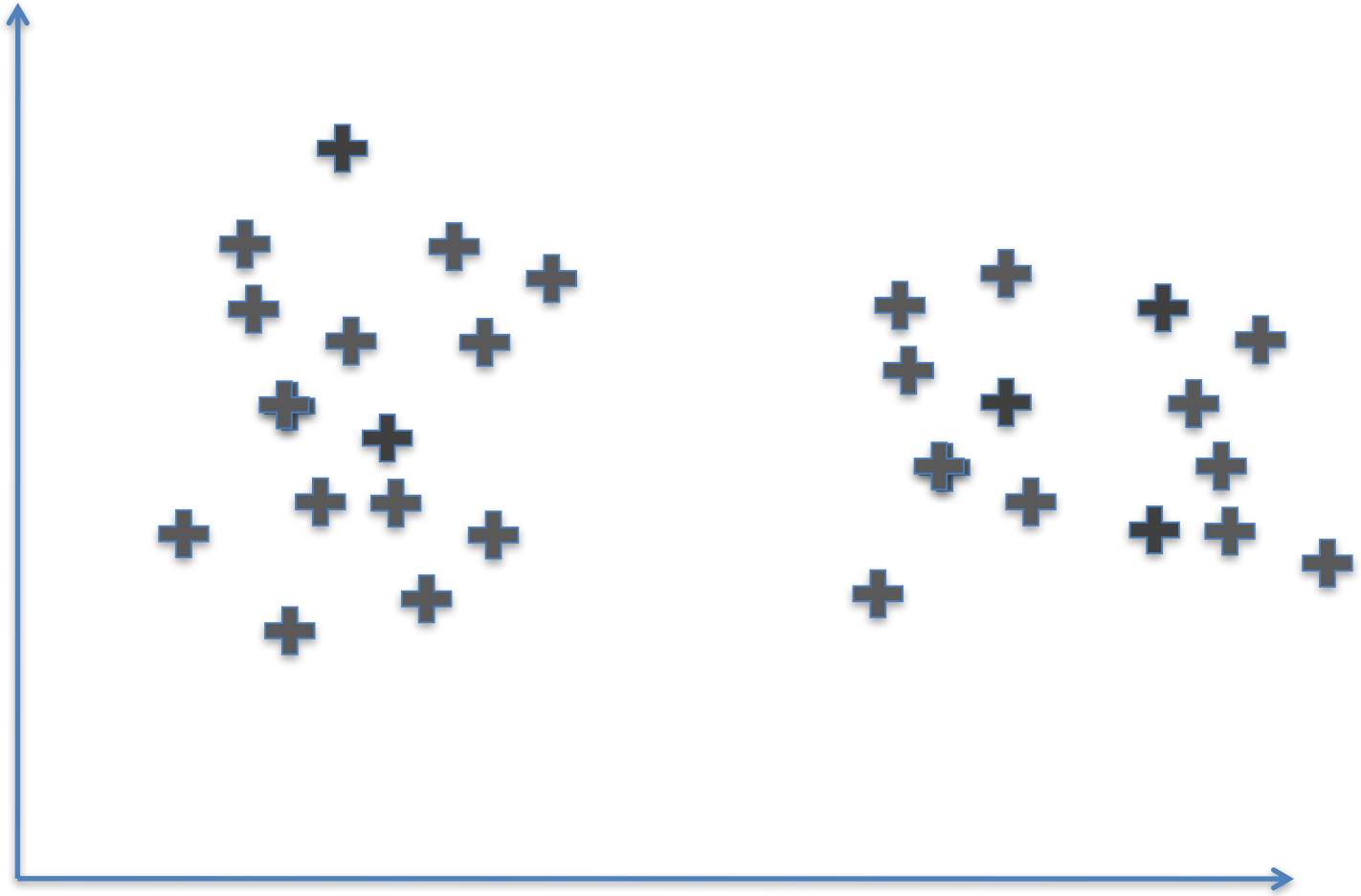
Announcement

- No office hour for William after class today!

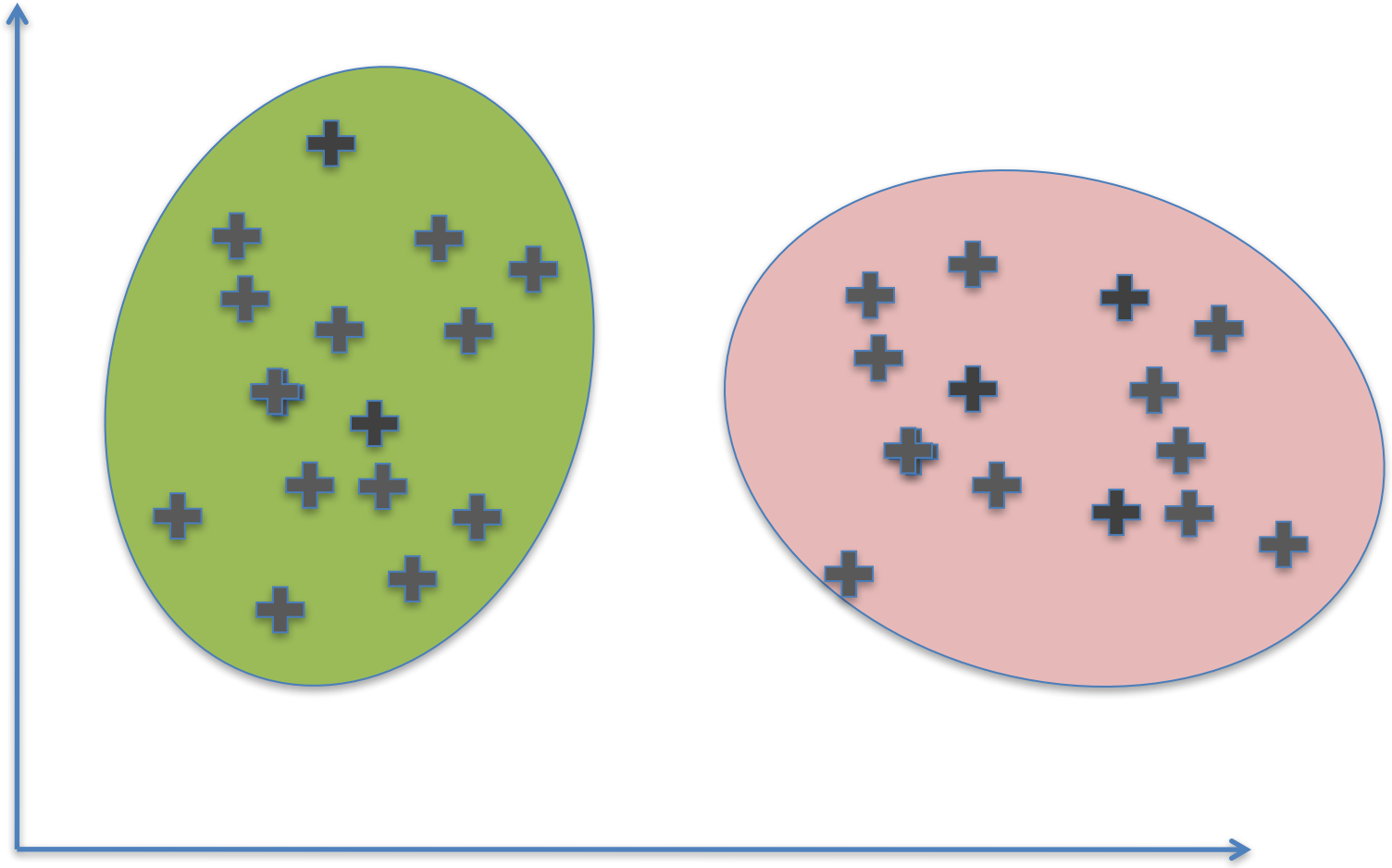
Semi-supervised learning

- 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 use k-means, etc find clusters; then label each cluster using L
- Question:
  - Can you use both L and U to do better?

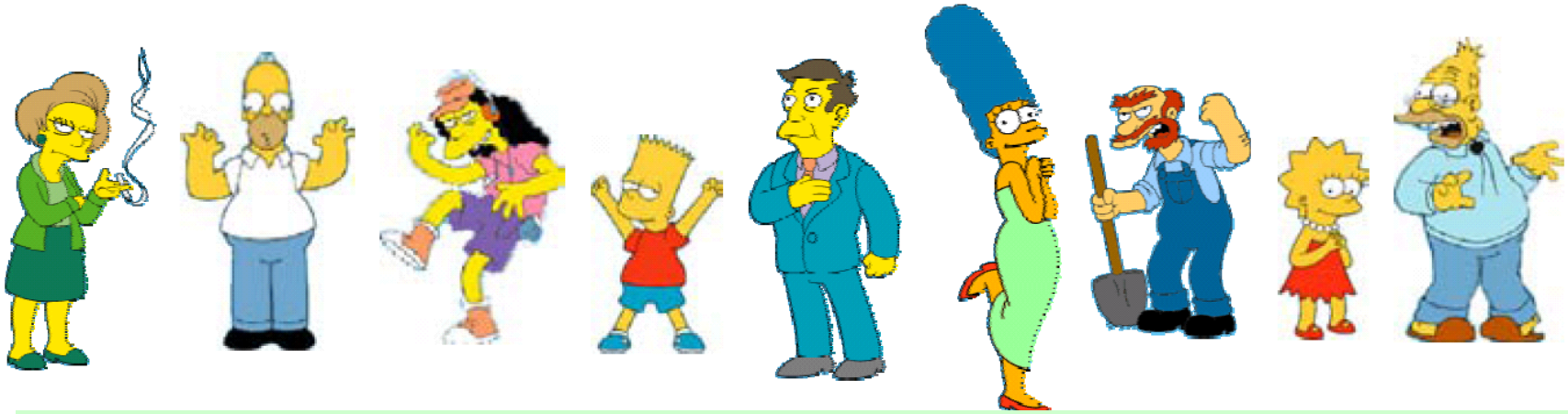
SSL is Somewhere Between Clustering and Supervised Learning



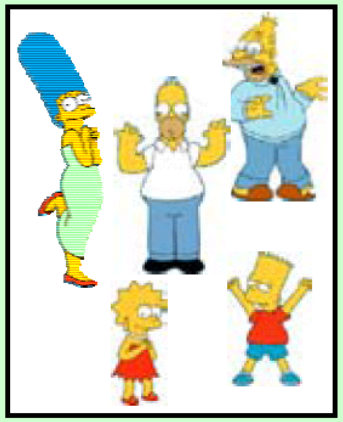
SSL is Between Clustering and SL



What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family

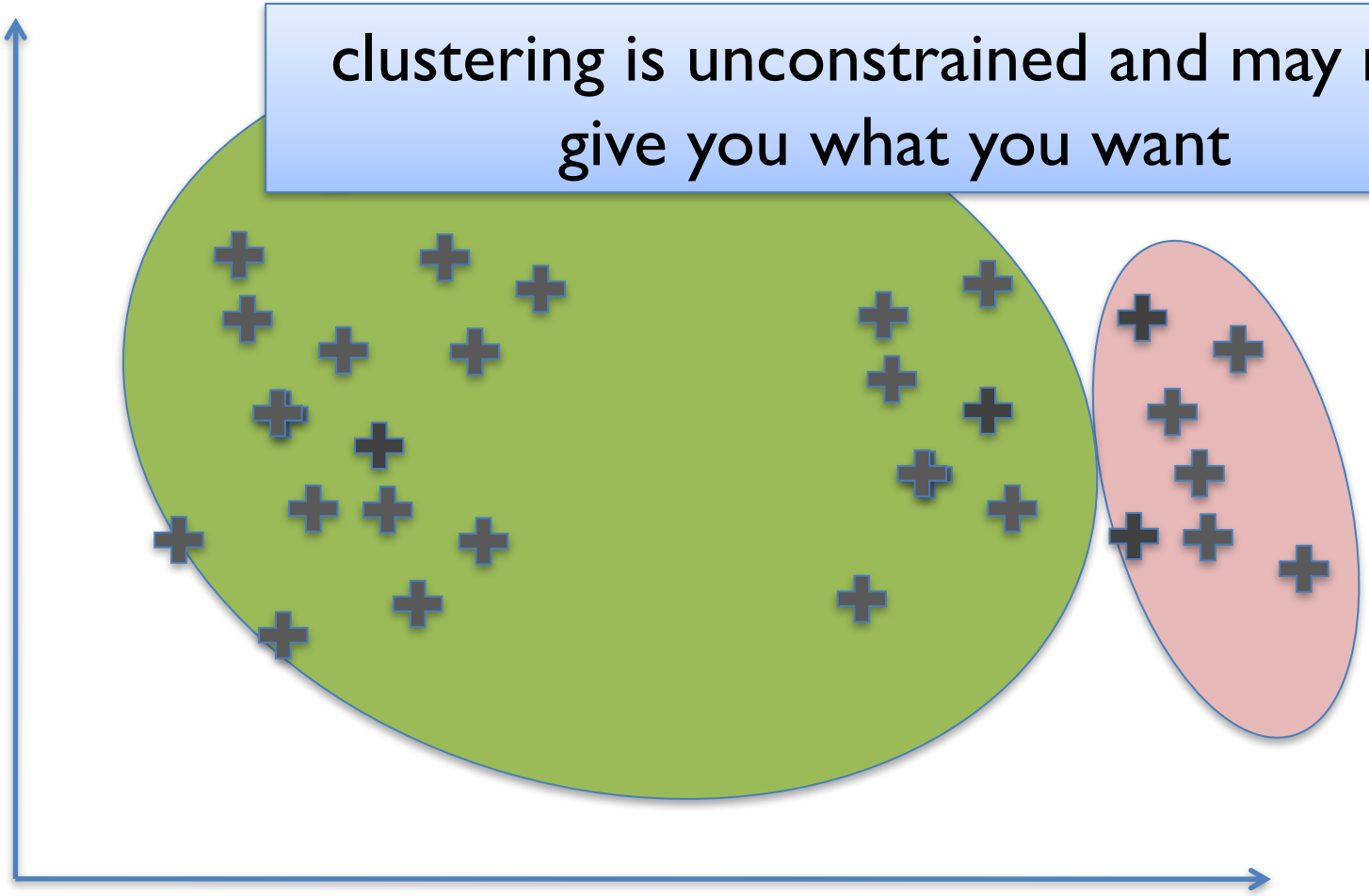
School Employees

Females

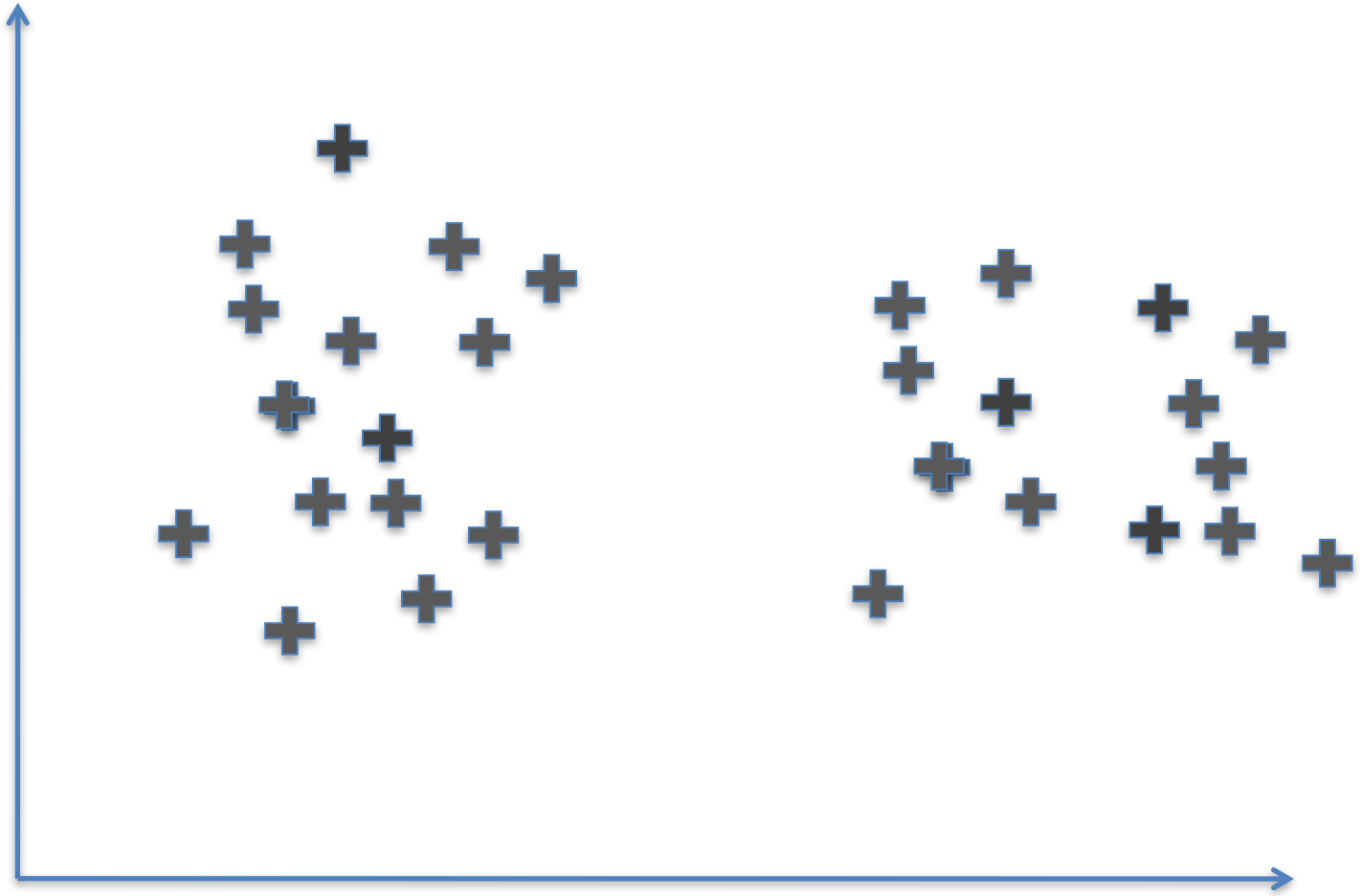
Males

slides: Bhavana Dalvi

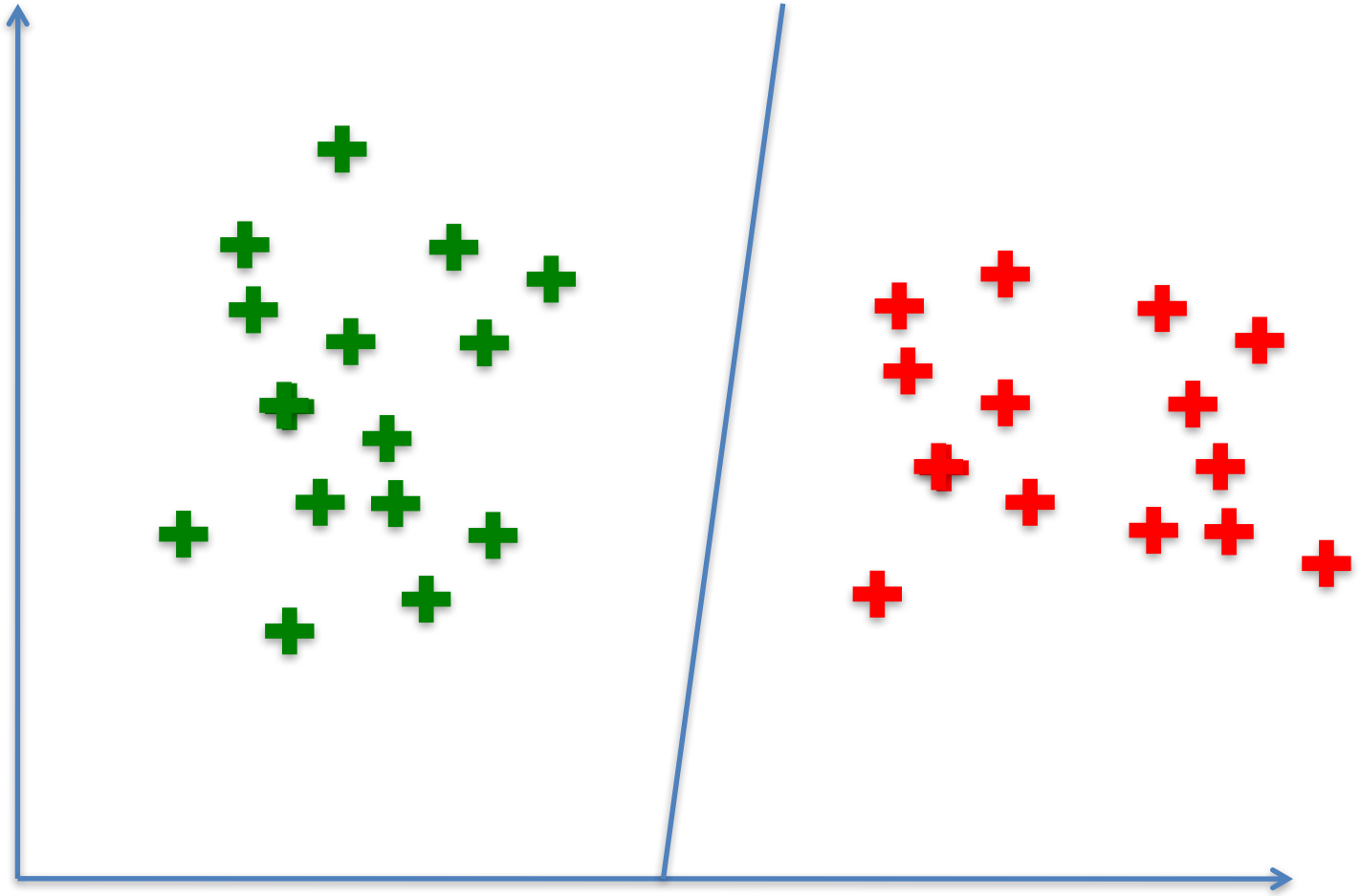
SSL is Between Clustering and SL



SSL is Between Clustering and SL

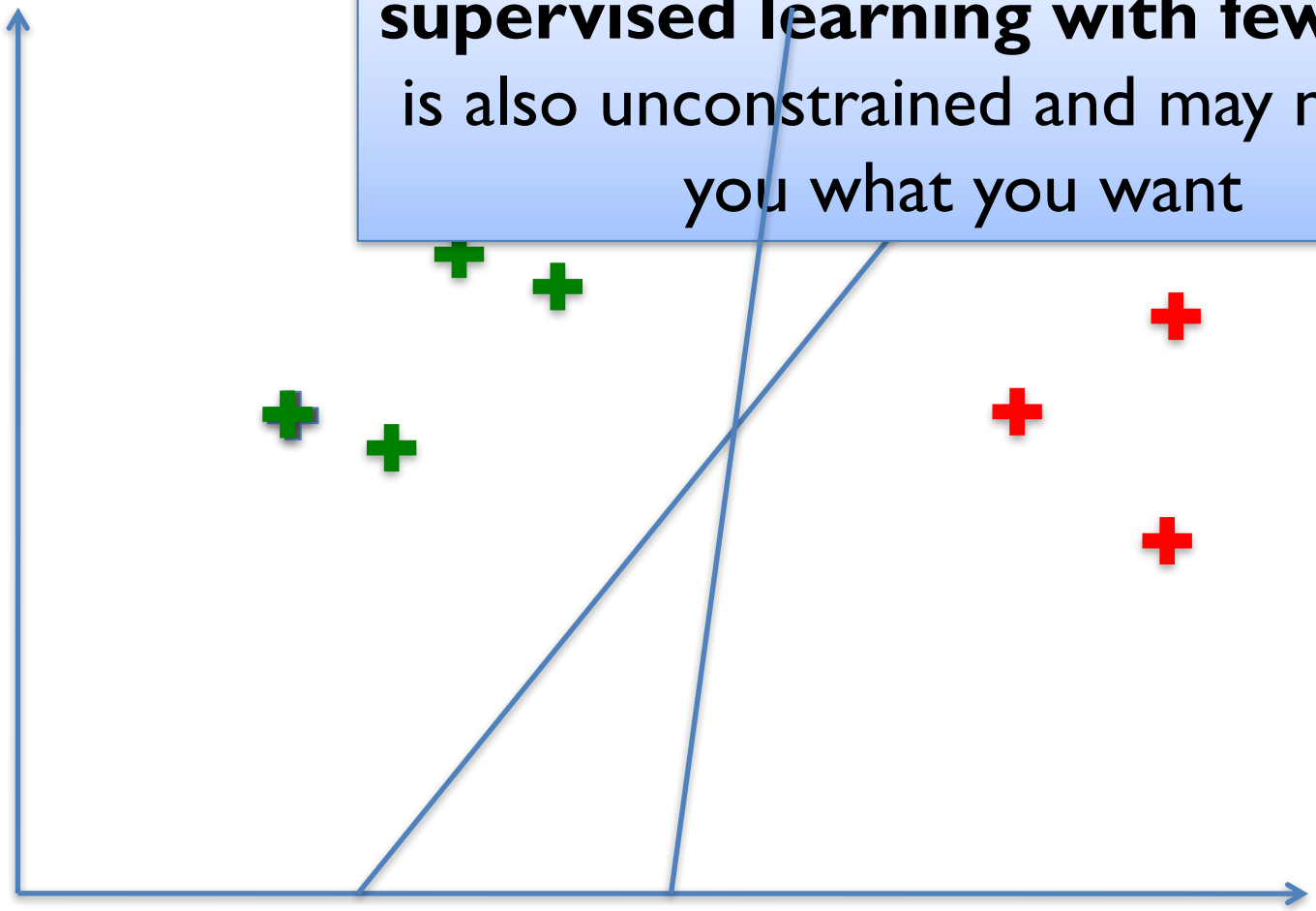


SSL is Between Clustering and SL

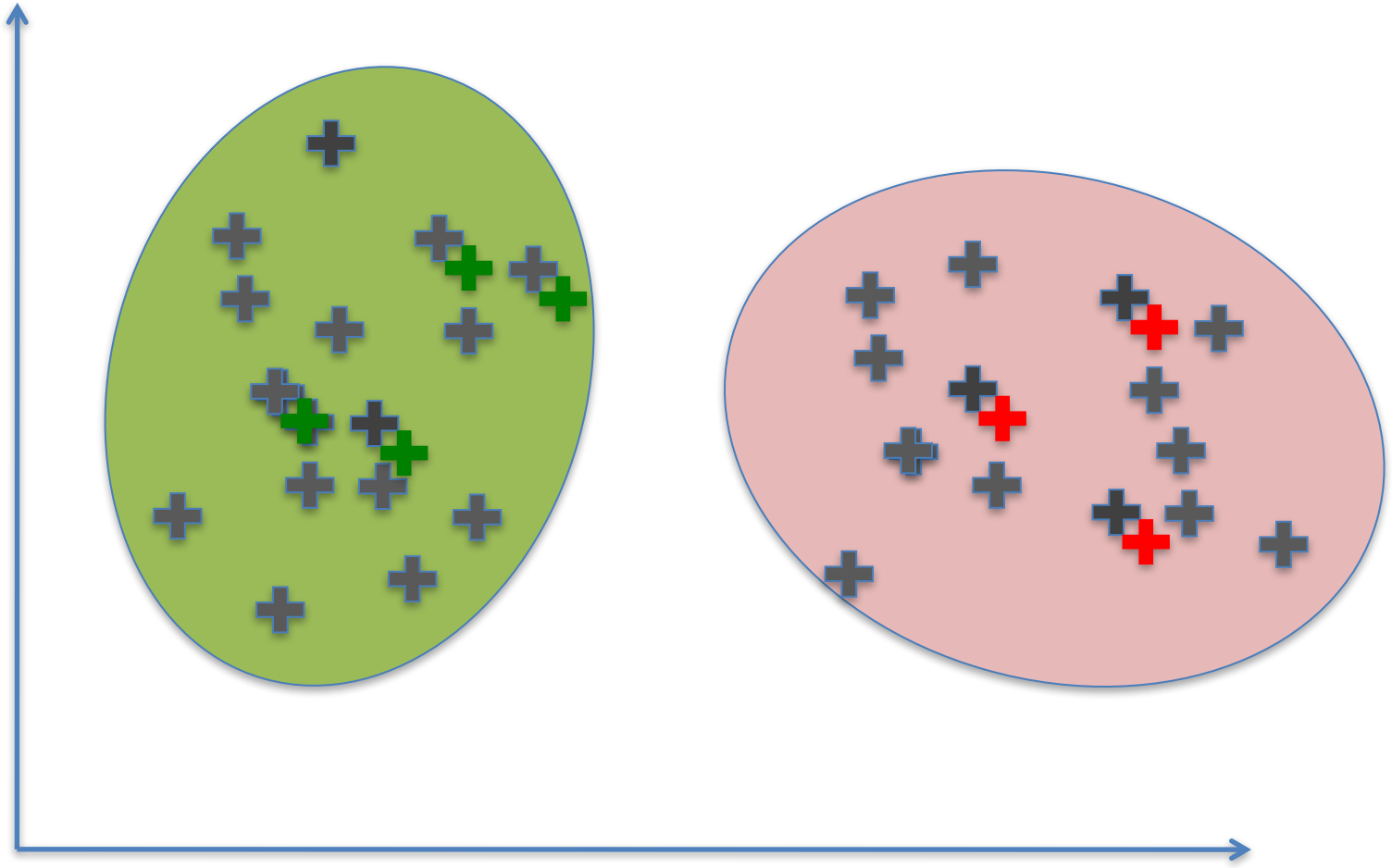


SSL is Between Clustering and SL

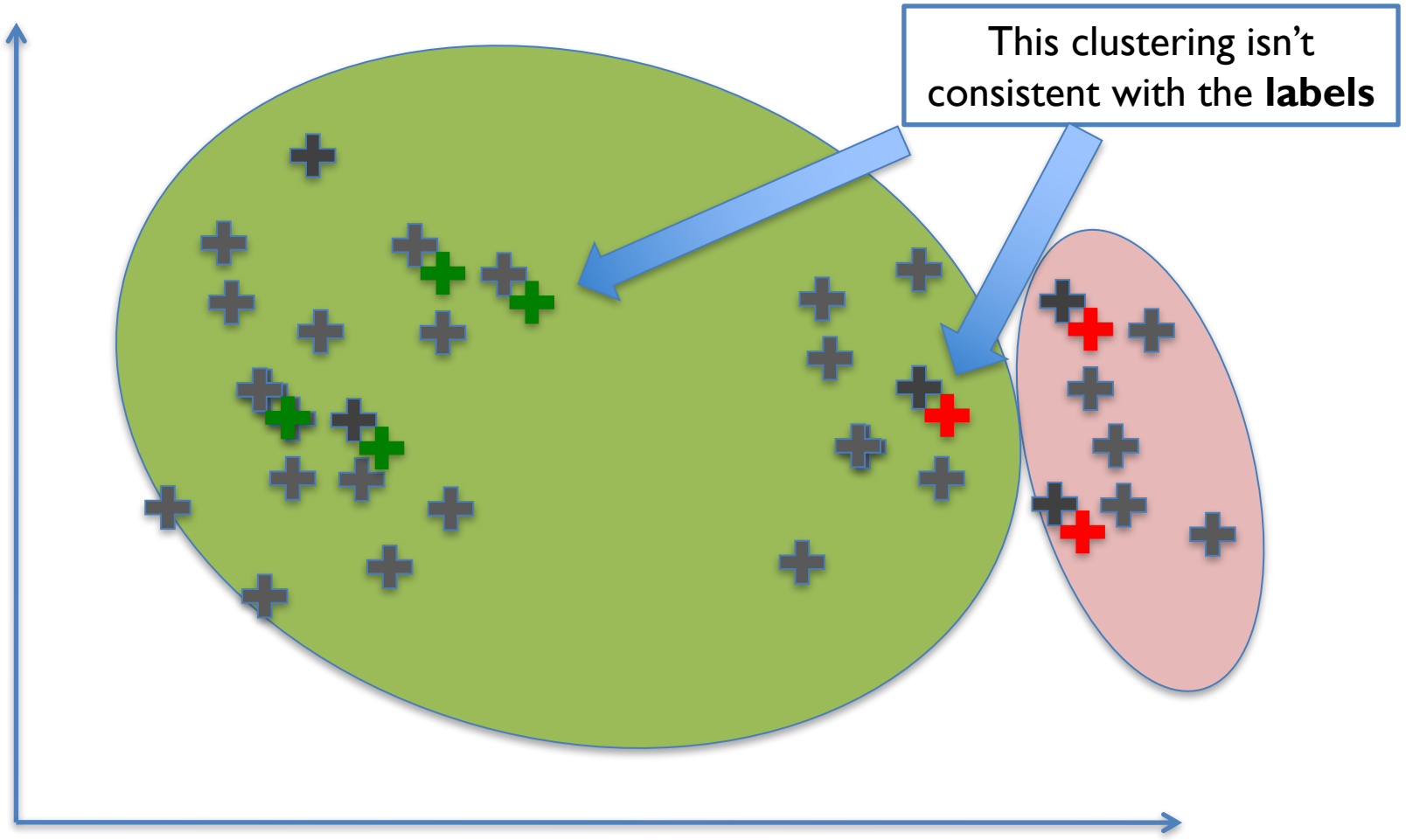
supervised learning with few labels is also unconstrained and may not give you what you want



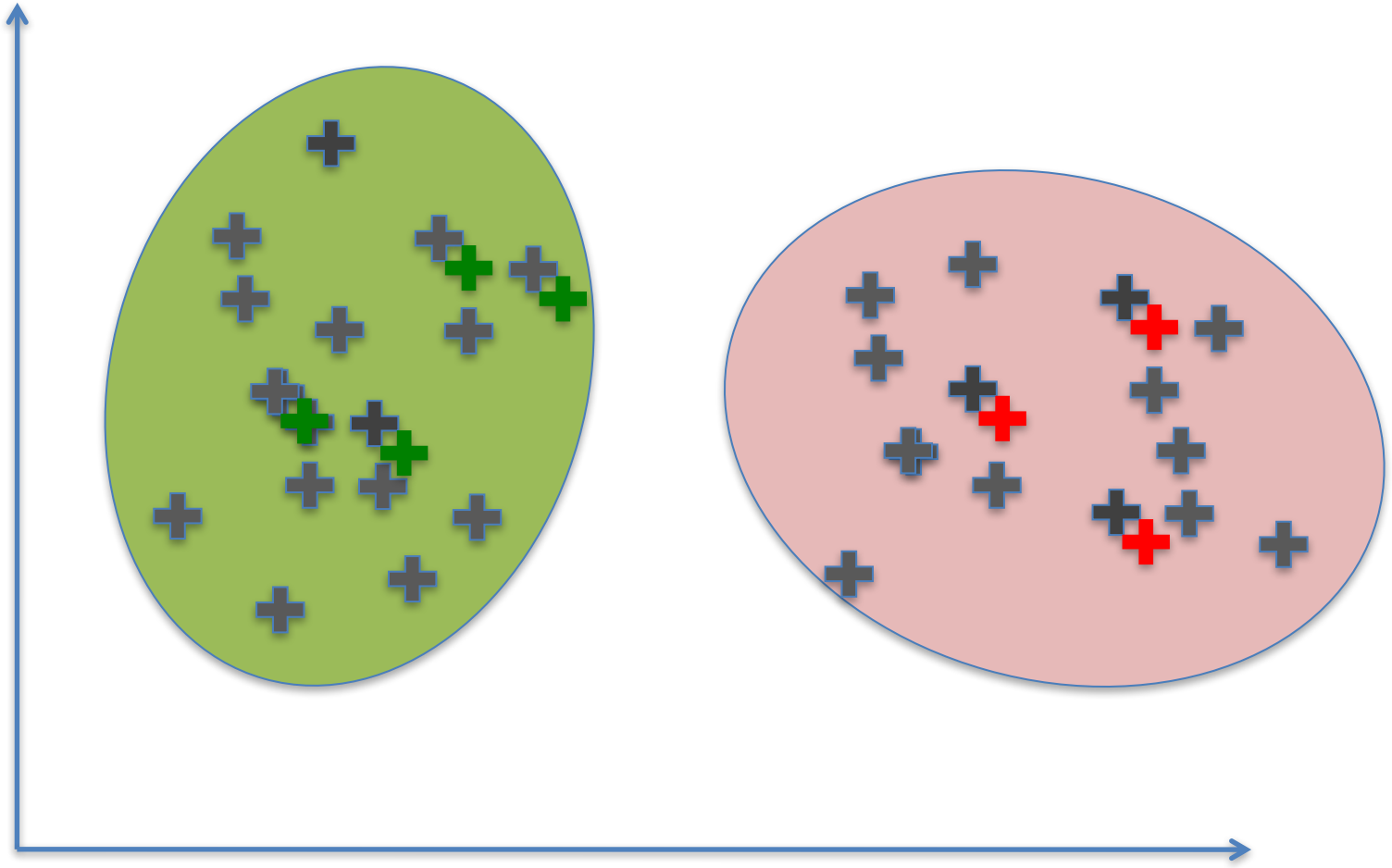
SSL is Between Clustering and SL



SSL is Between Clustering and SL



SSL is Between Clustering and SL



SSL in Action: The NELL System

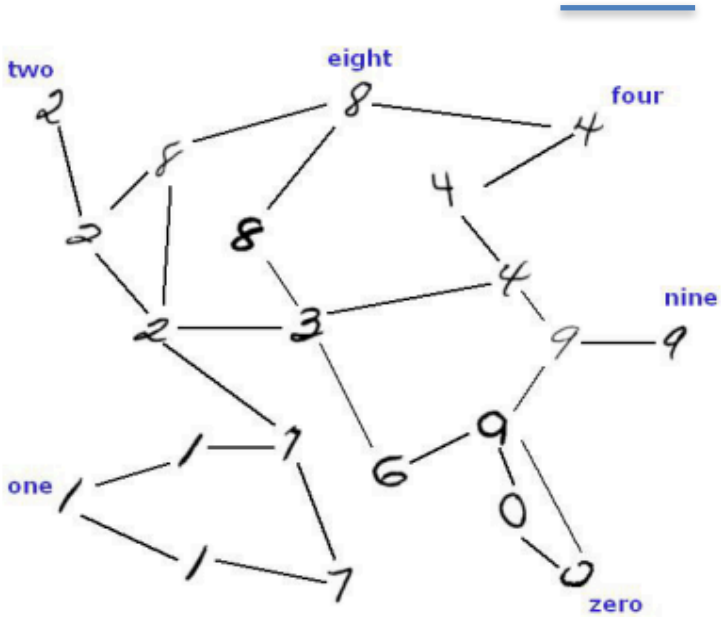
Type of SSL

- Margin-based: transductive SVM
  - Logistic regression with entropic regularization
- Generative: seeded k-means
- Nearest-neighbor like: graph-based SSL

Harmonic Fields
aka coEM aka wvRN…

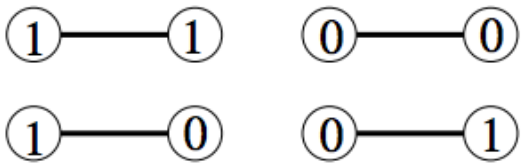
- Idea: construct a graph connecting the most similar examples (k-NN graph)
- Intuition: **nearby points should have similar labels** – labels should "propagate" through the graph
- Formalization: try and minimize "energy" defined as:

Harmonic fields – Gharamani, Lafferty and Zhu



In this example y is a length-10 vector

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



happy, low energy

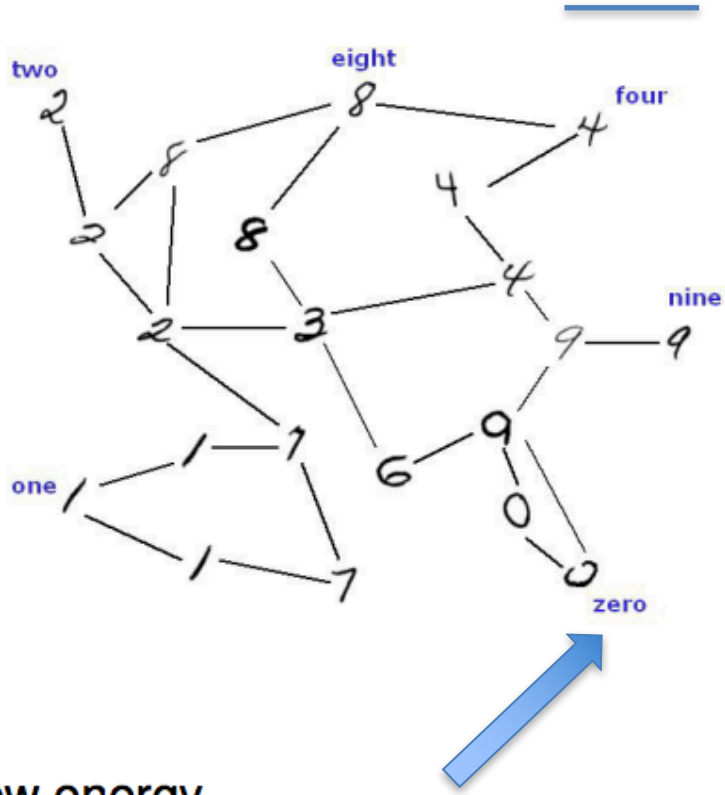
unhappy, high energy

Observed label

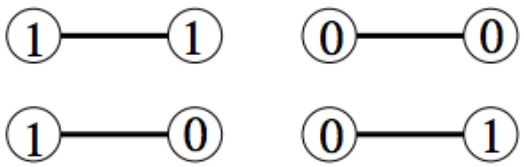
- Result 1: at the minimal energy state, each node's value is a **weighted average of its neighbor's weights**:

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

Harmonic fields – Gharamani, Lafferty and Zhu



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



happy, low energy

unhappy, high energy

Observed label

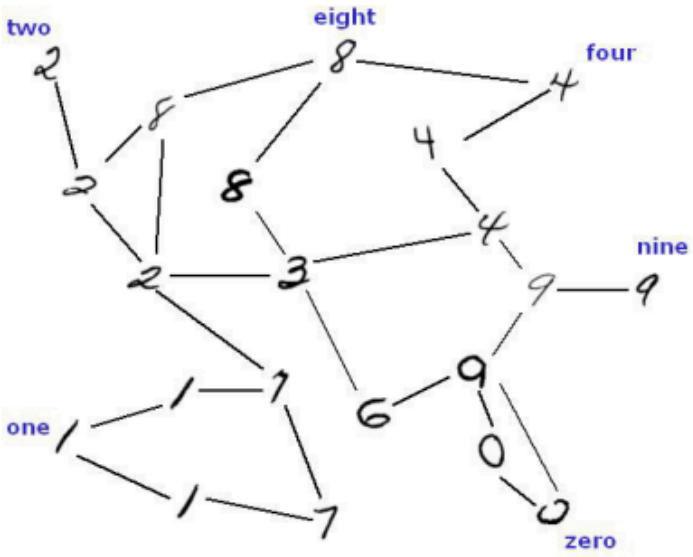
"Harmonic field" LP algorithm

- Result 2: you can reach the minimal energy state with a simple iterative algorithm:
  - 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

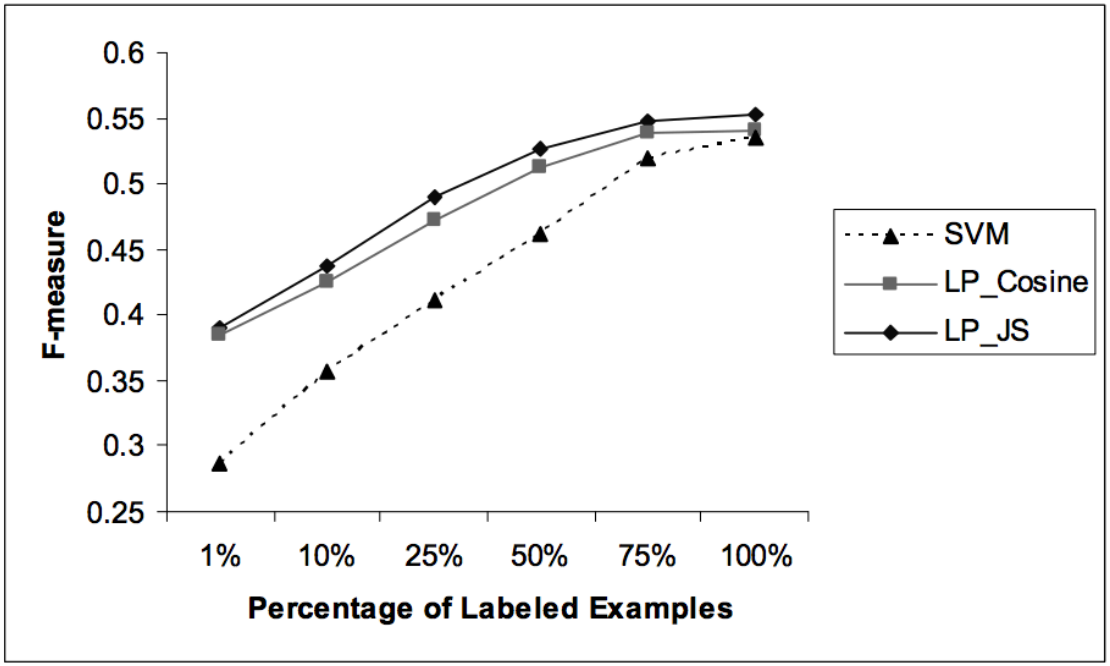
$$V^{t+1}(i,c) = \frac{1}{Z}\sum_j w_{i,j} V^t(j,c)$$

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

Harmonic fields – Gharamani, Lafferty and Zhu



This family of techniques is called "Label propagation"

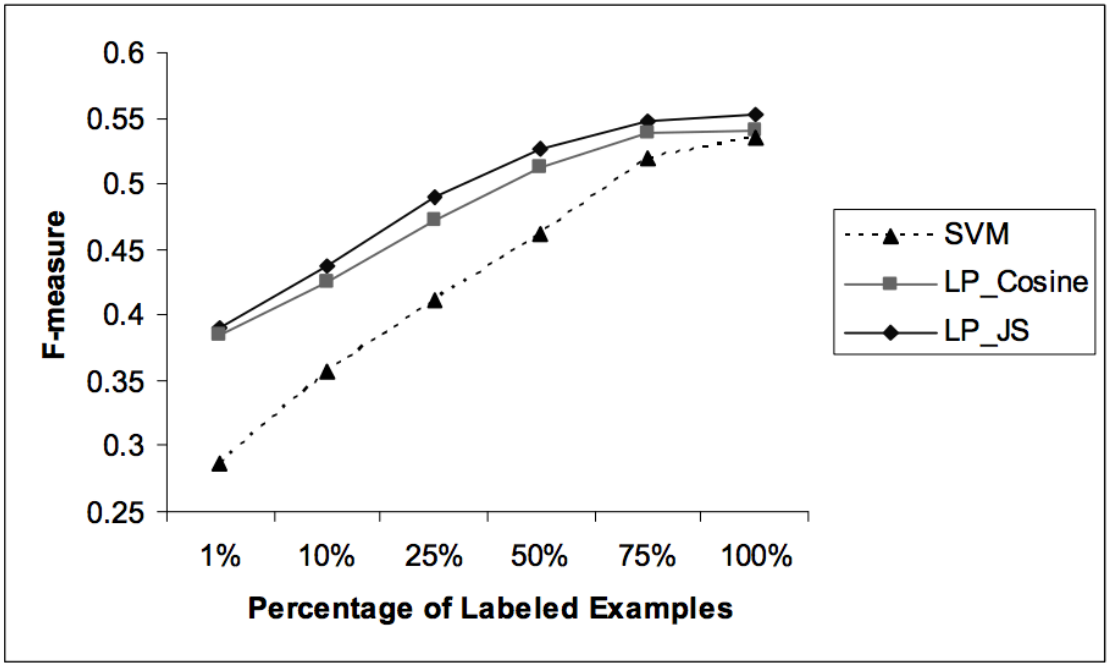


Harmonic fields – Gharamani, Lafferty and Zhu

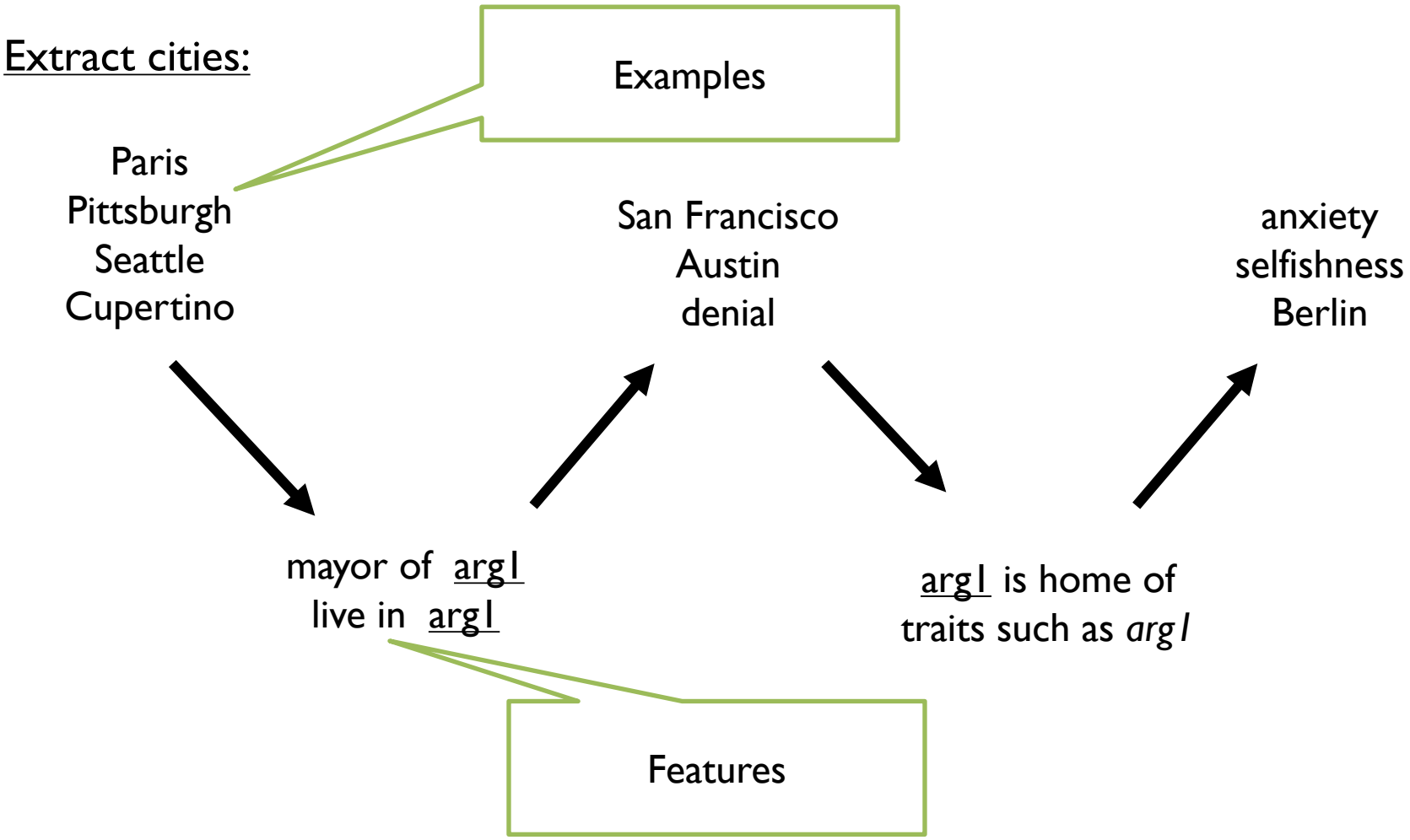
This family of techniques is called "Label propagation"

This experiment points out some of the issues with LP:

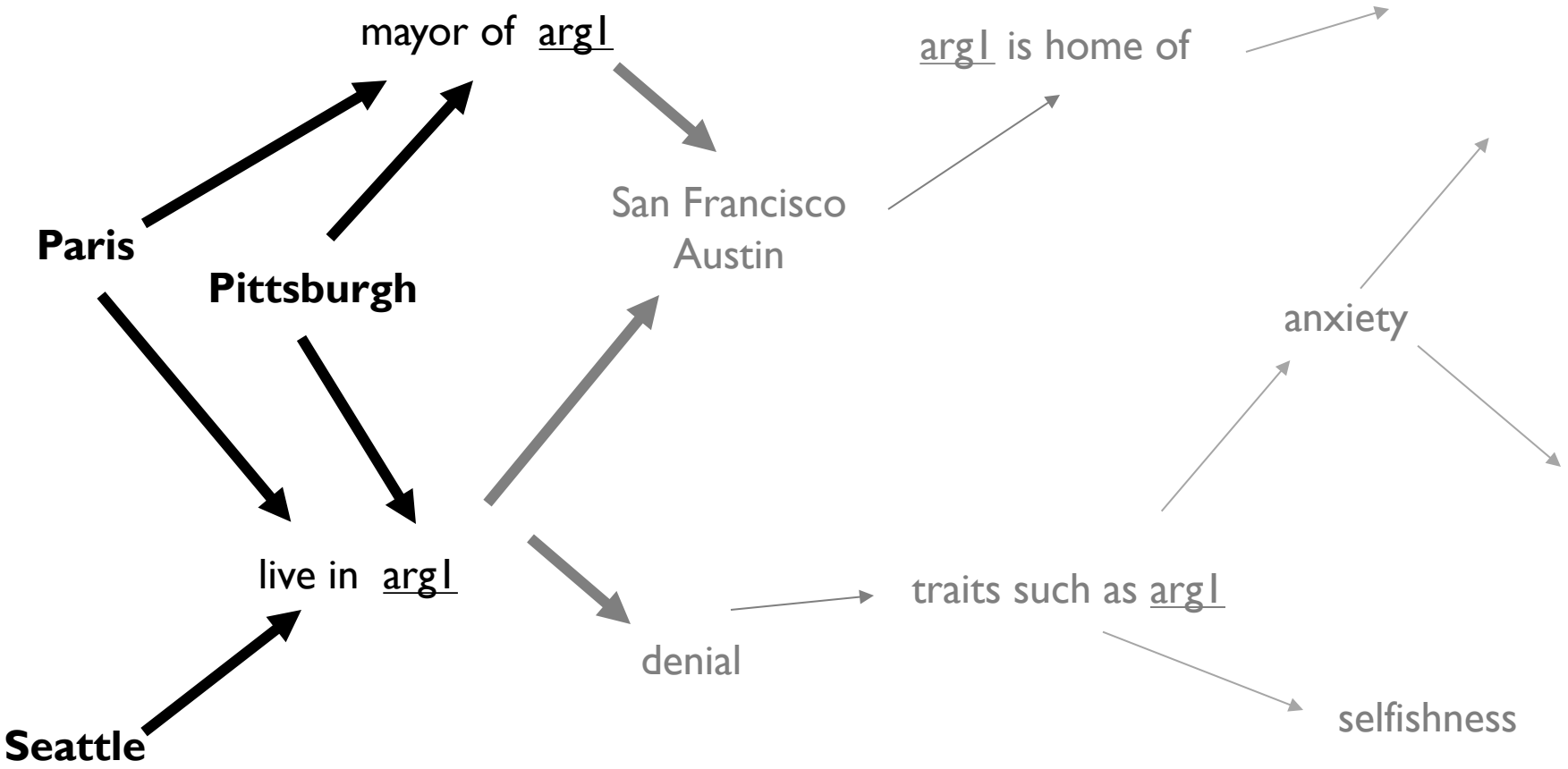
1. What distance metric do you use?
2. What energy function do you minimize?
3. What is the right value for K in your K-NN graph? Is a K-NN graph right?
4. If you have lots of data, how expensive is it to build the graph?



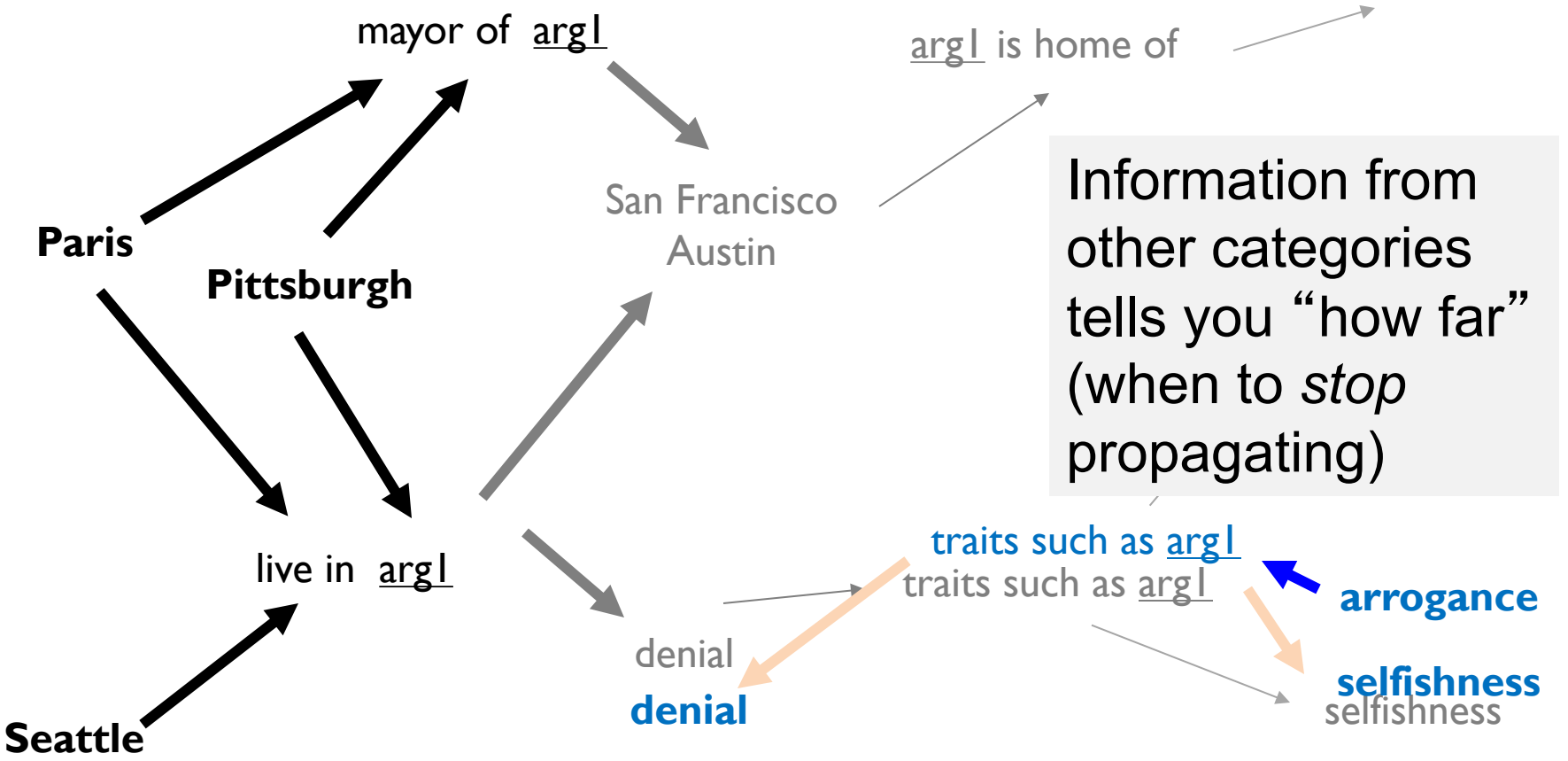
NELL: Uses Co-EM ~= HF



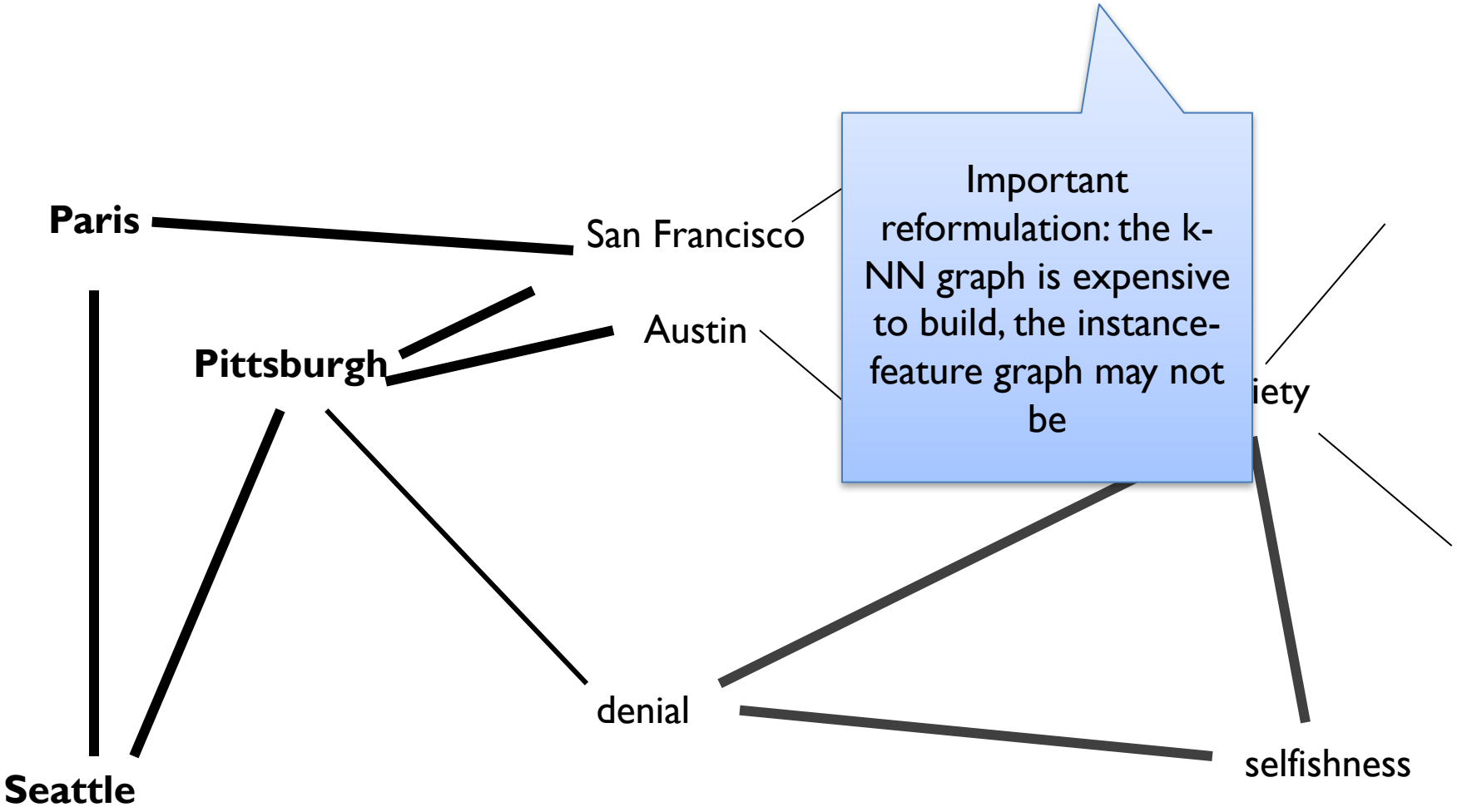
Semi-Supervised Bootstrapped Learning via Label Propagation



Semi-Supervised Bootstrapped Learning via Label Propagation



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



Some other 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
  - Often we need to balance J_L and J_U
- Besides L, what other information about the task is useful (or necessary)?
  - Common choice: relative frequency of classes
  - Various ways of incorporating this into the optimization problem

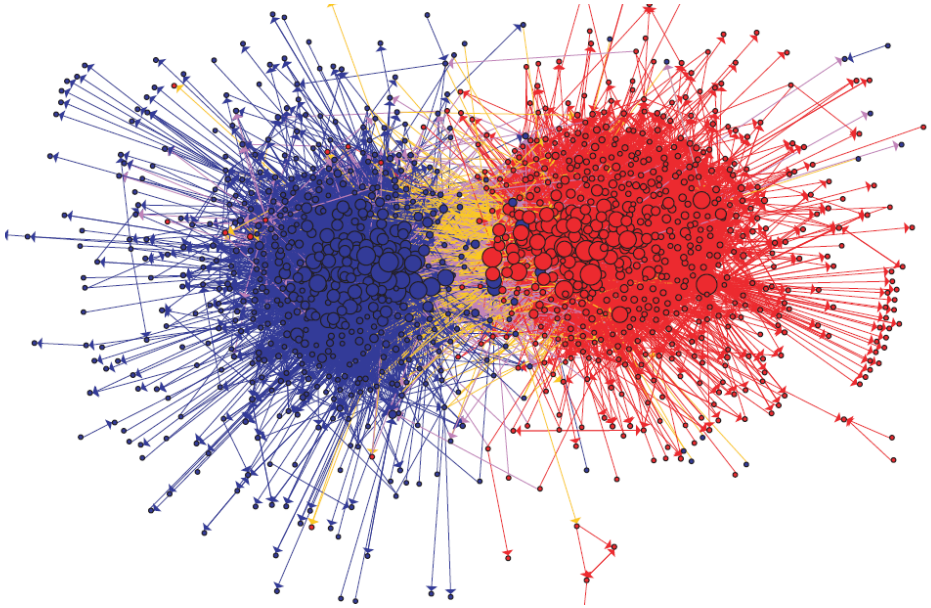
Semi-Supervised Classification of Network Data Using Very Few Labels

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

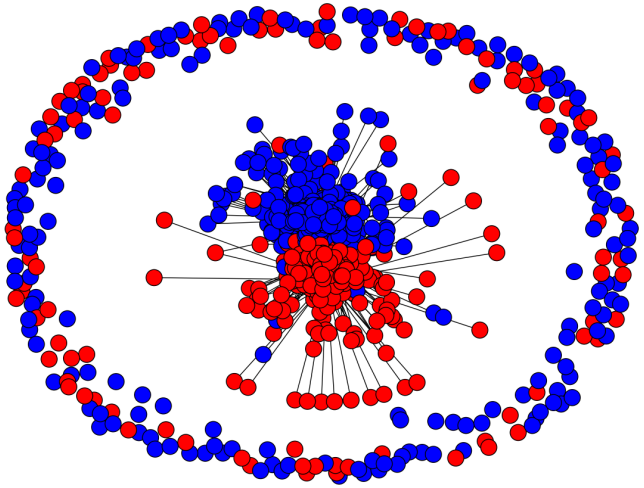
William W. Cohen
Carnegie Mellon University, Pittsburgh, Pennsylvania
Email: wcohen@cs.cmu.edu

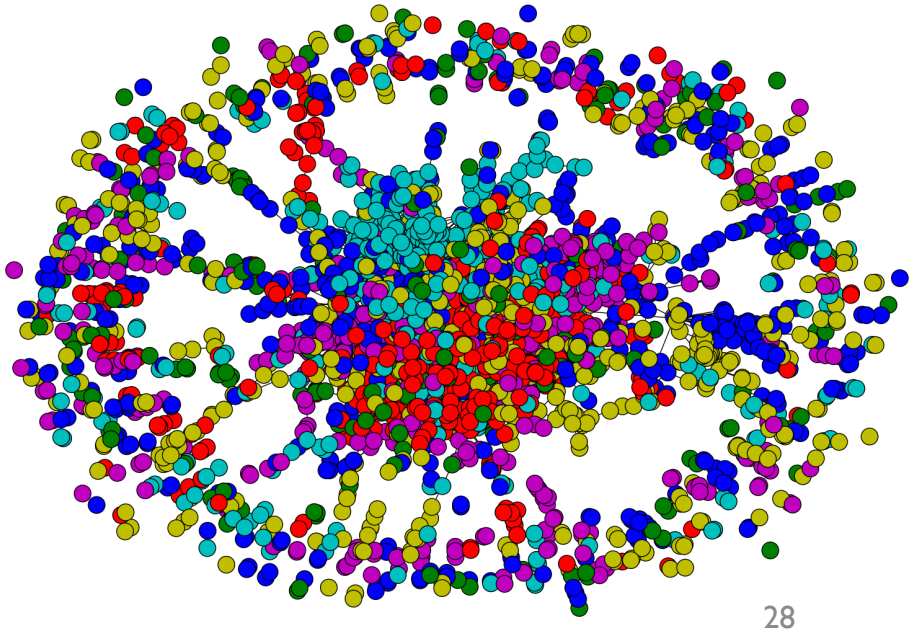
ASONAM-2010 (Advances in Social Networks Analysis and Mining)

Network Datasets with Known Classes



- UBMCBlog
- AGBlog
- MSPBlog
- Cora
- Citeseer





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.

RWR - fixpoint of:

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

aka Personalized PageRank

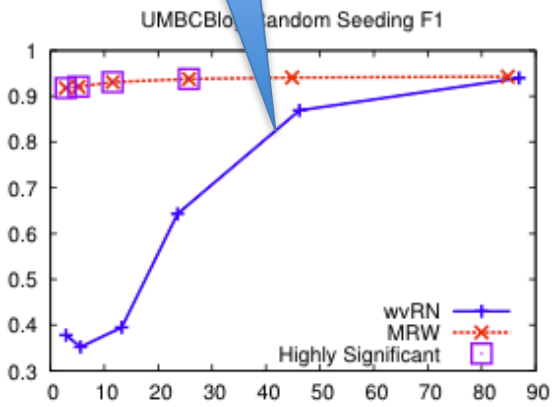
Seed selection

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

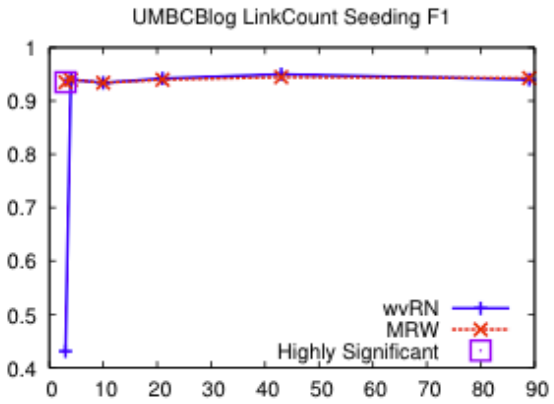
HF method

Results – Blog data

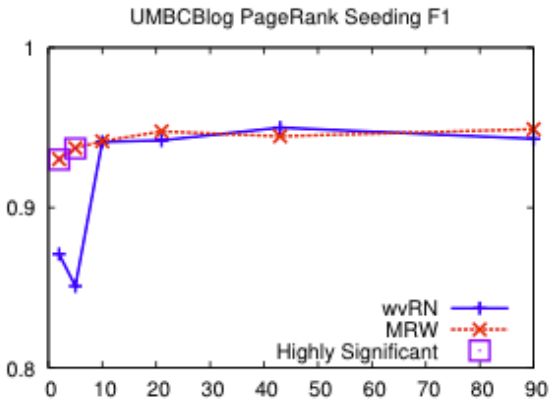
Random

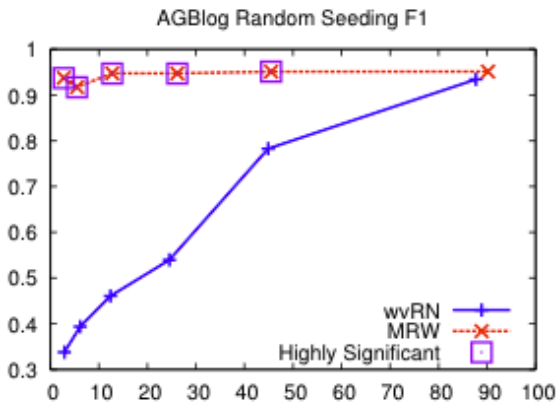


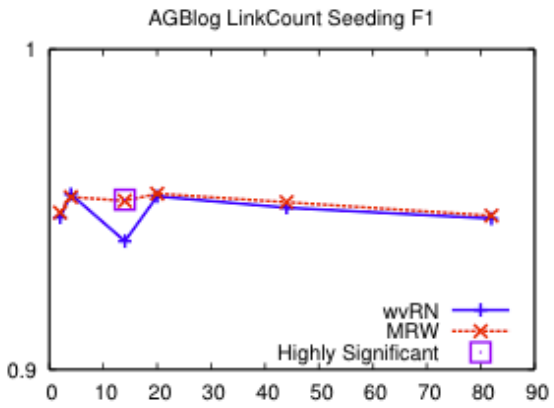
Degree

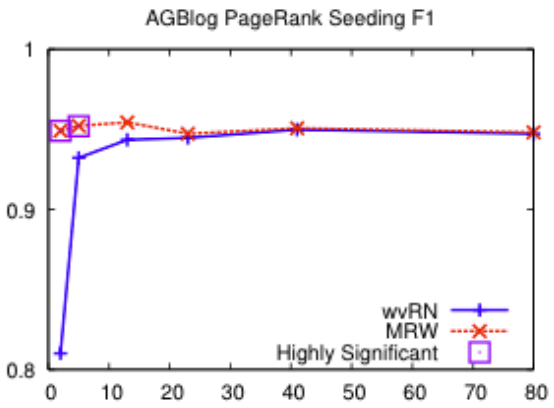


PageRank



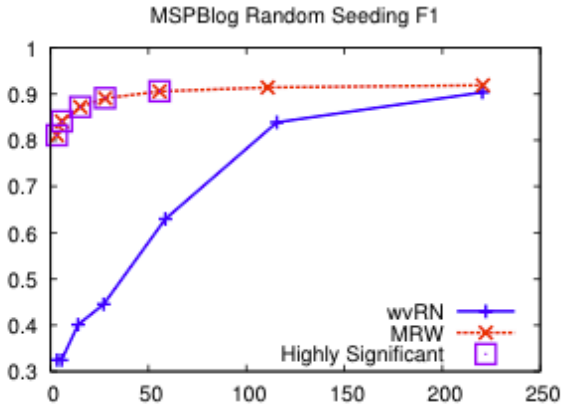




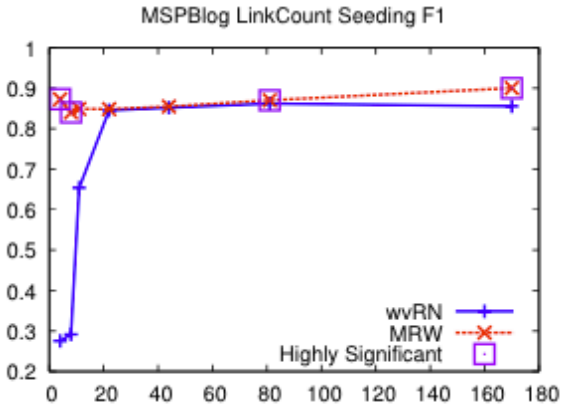


Results – More blog data

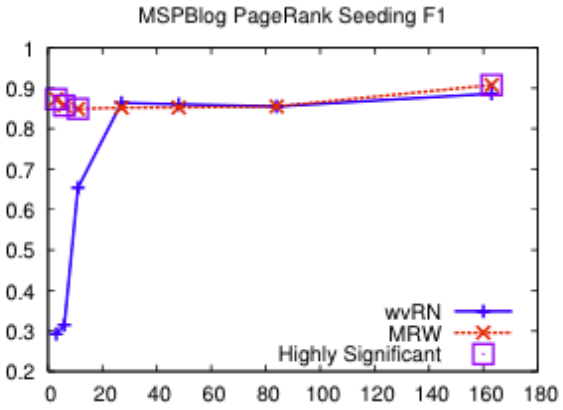
Random



Degree

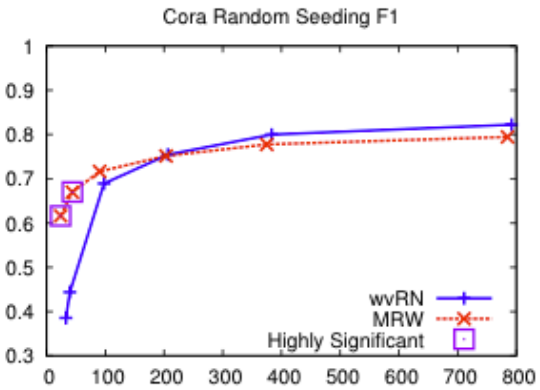


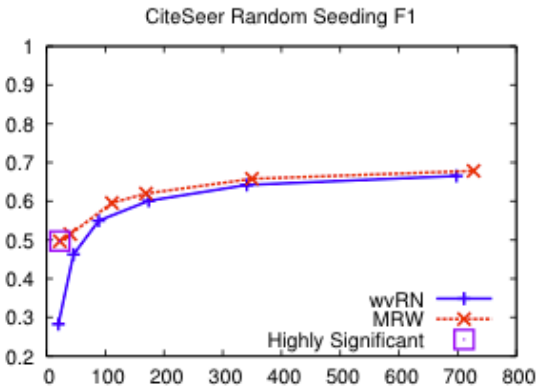
PageRank



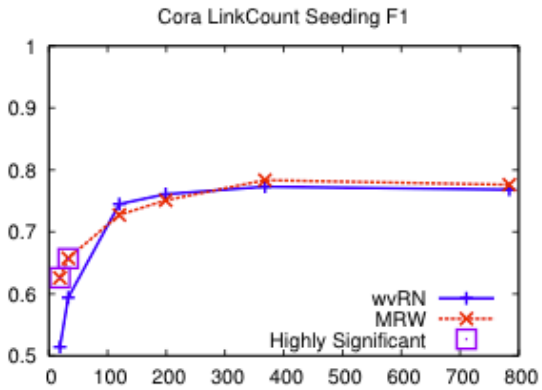
Results – Citation data

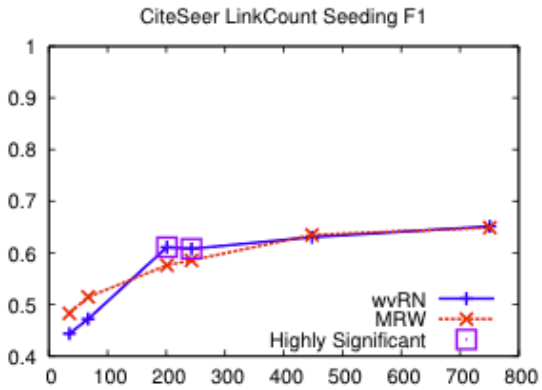
Random



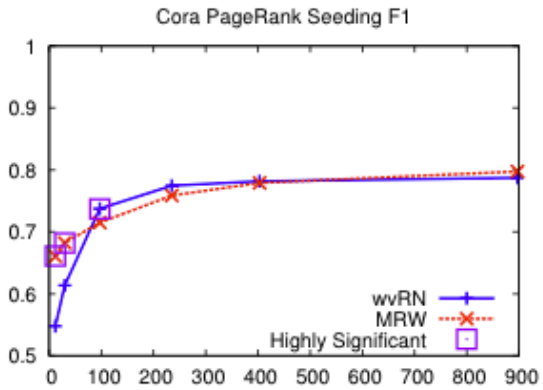


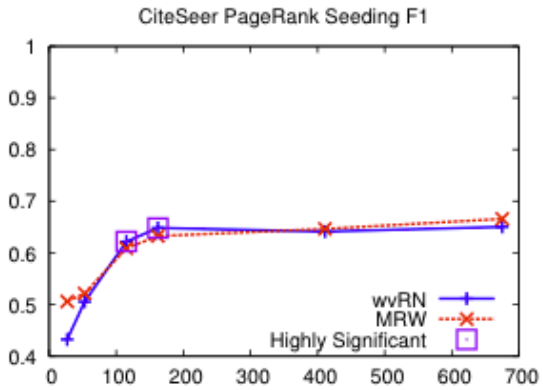
Degree



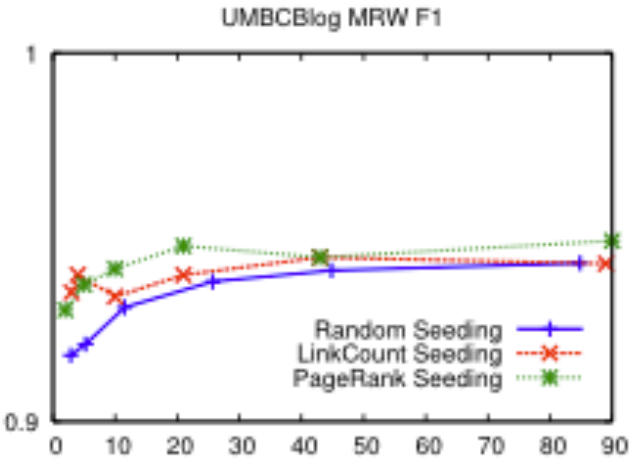


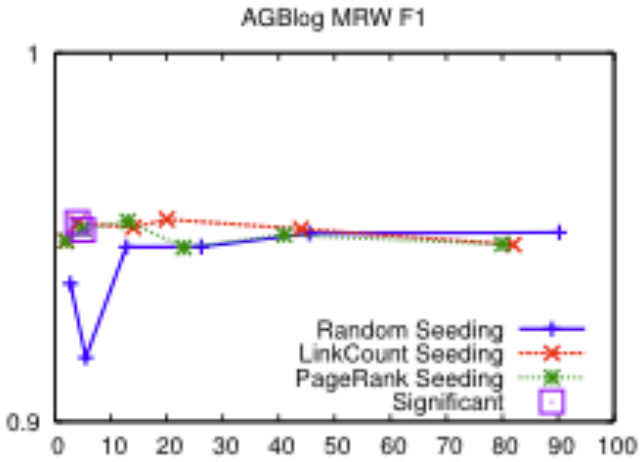
PageRank

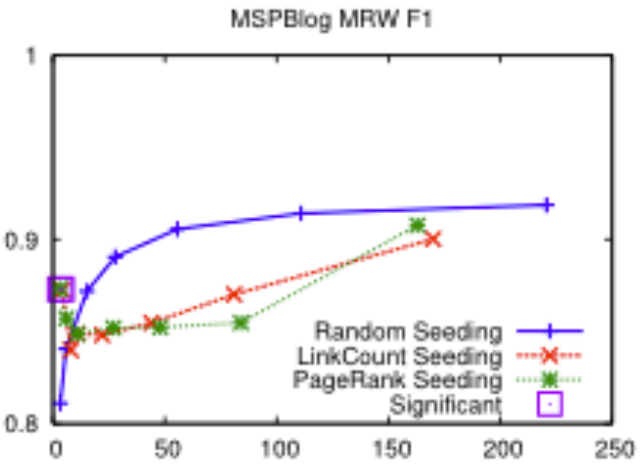


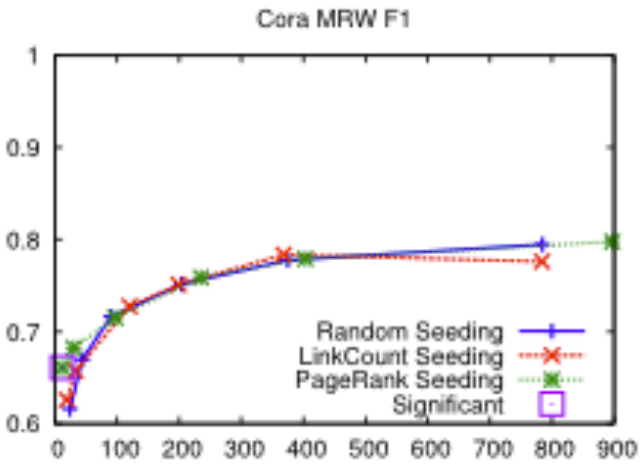


Seeding – MultiRankWalk

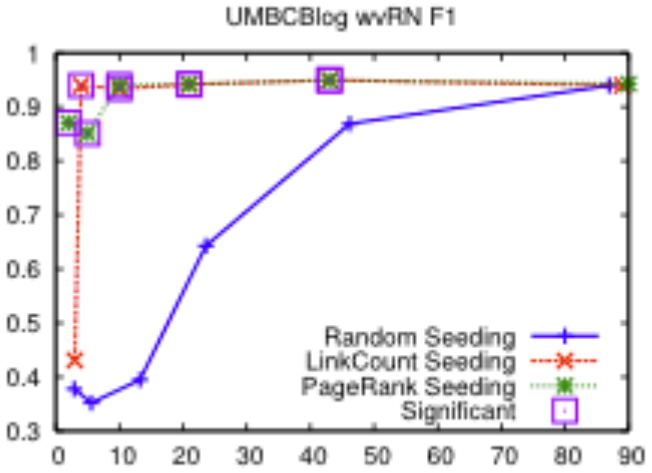


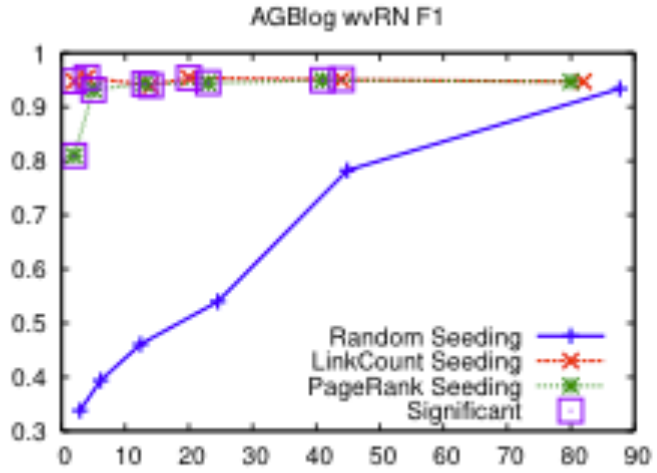


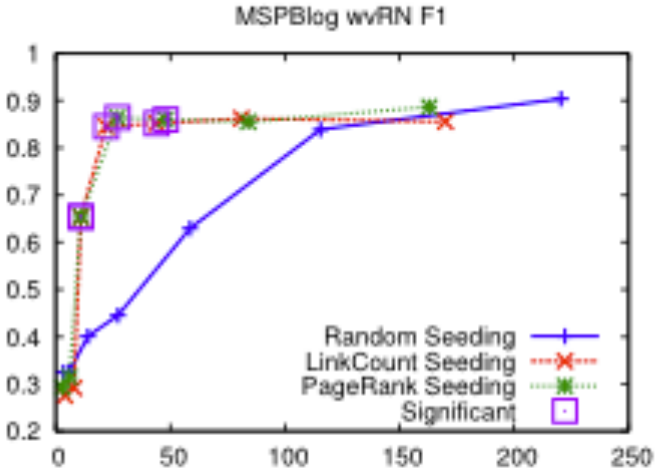


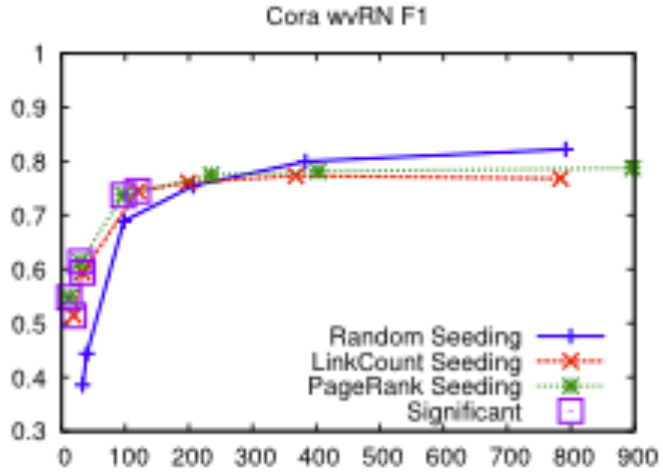


Seeding – HF/wvRN



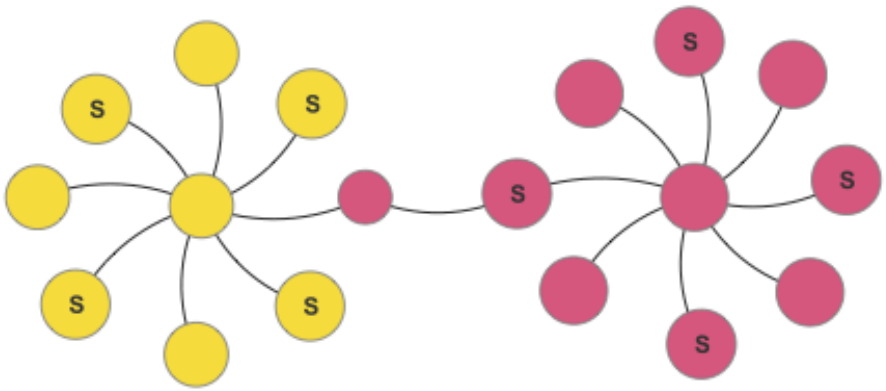




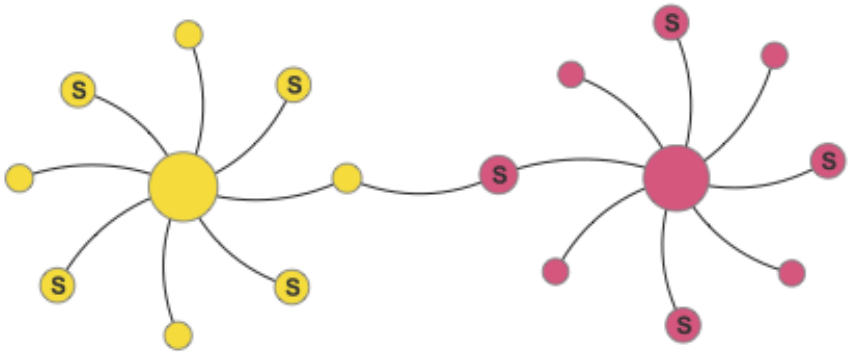


MultiRank Walk vs HF/wvRN/CoEM

Seeds are marked S

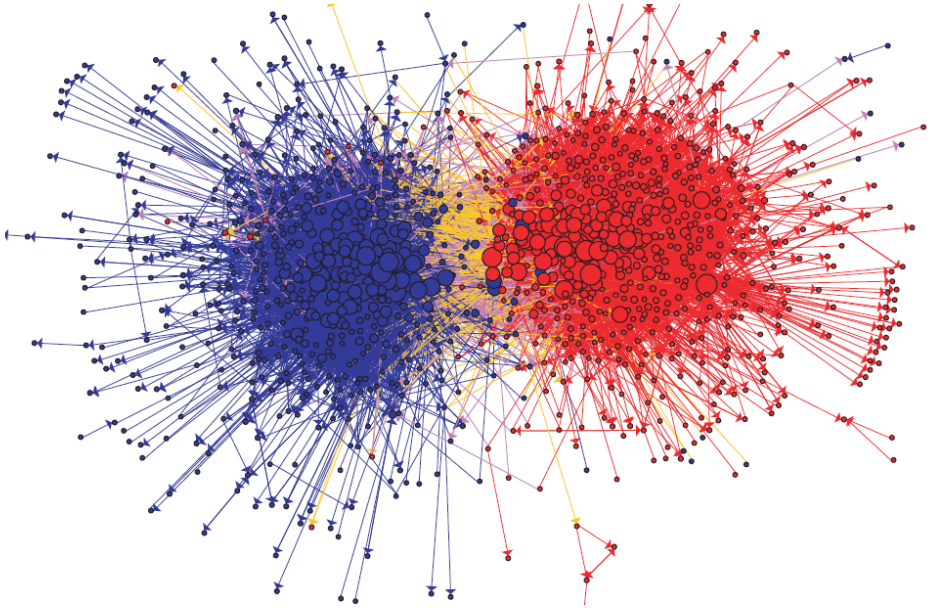


HF

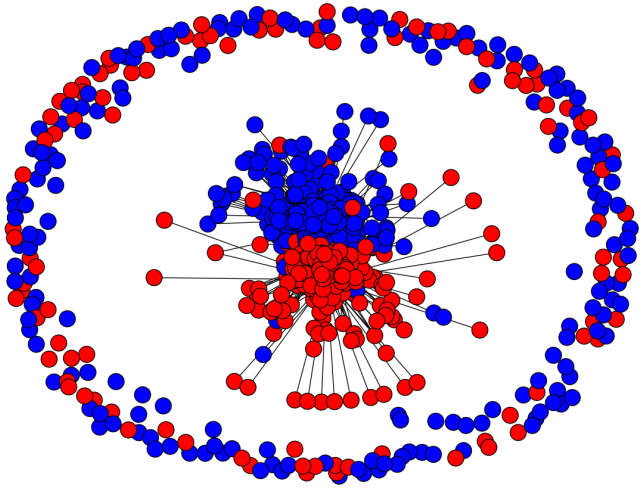


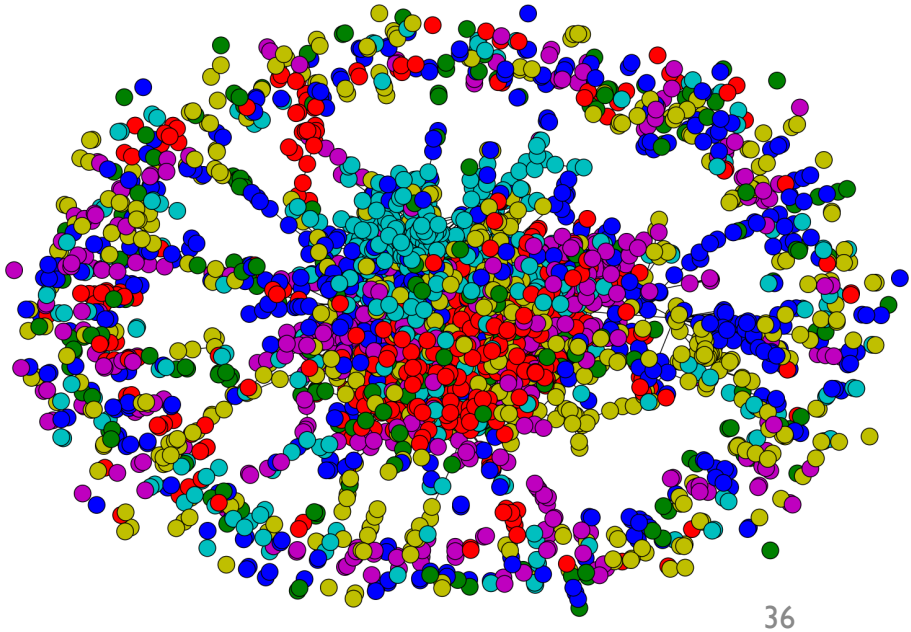
MRW

Back to Experiments: Network Datasets with Known Classes



- UBMCBlog
- AGBlog
- MSPBlog
- Cora
- Citeseer





MultiRankWalk vs wvRN/HF/CoEM

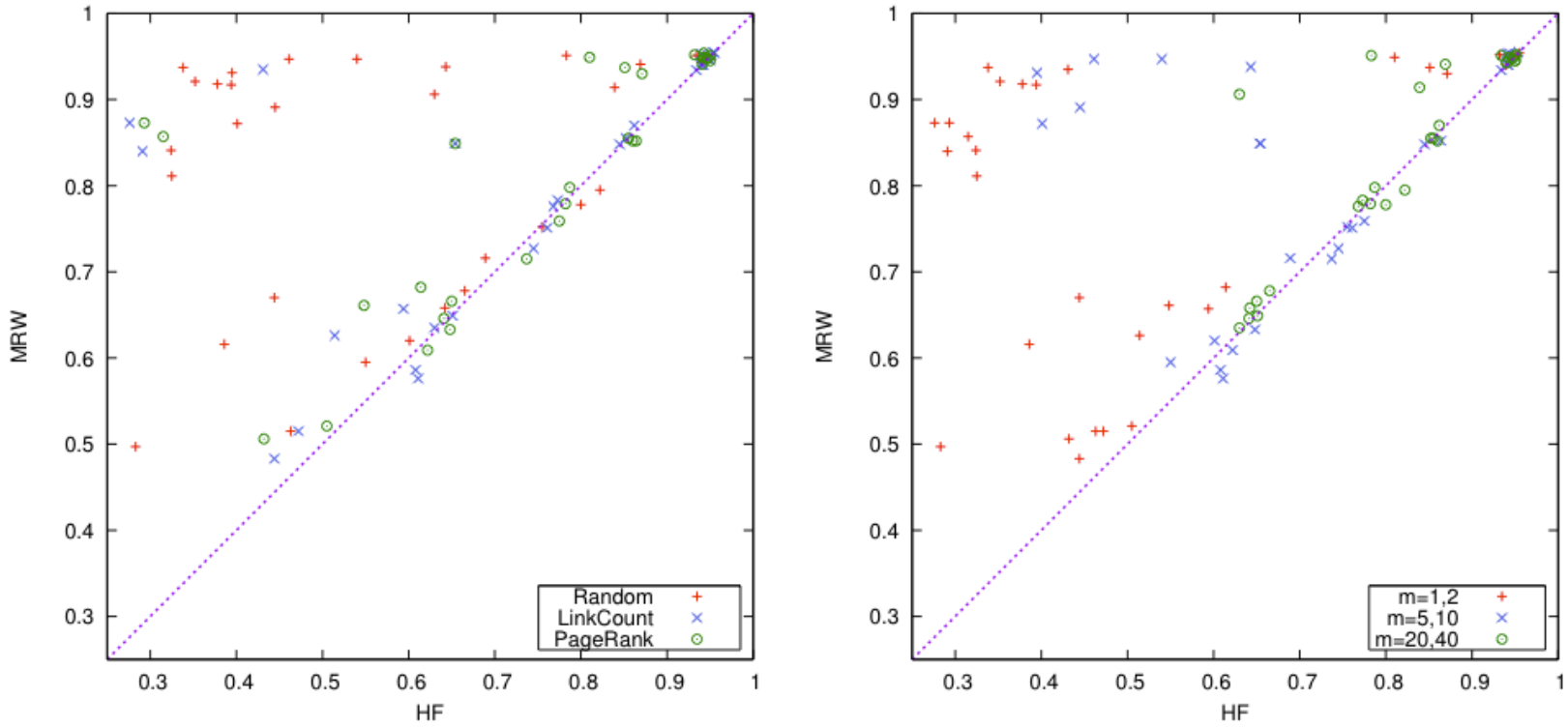


Figure 2.6: Scatter plots of HF F1 score versus MRW F1 score. The left plot marks different seeding preferences and the right plot marks varying amount of training labels determined by m.

Harmonic Fields
aka coEM aka wvRN

CoEM/HF/wvRN

S. A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the Multi-Relational Data Mining Workshop (MRDM) at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

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 $\mathcal{N}_i$:

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

"Simple relational classifier" is same as the *harmonic field* – the score of each node in the graph is the harmonic (linearly weighted) average of its neighbors' scores.

CoEM/wvRN/HF

- Another justification of the same algorithm goes back to 2003
- … start with co-training with a naïve Bayes learner

- **Inputs:** An initial collection of labeled documents and one of unlabeled documents.
- Loop while there exist documents without class labels:
  - Build classifier A using the A portion of each document.
  - Build classifier B using the B portion of each document.
  - For each class C, pick the unlabeled document about which classifier A is most confident that its class label is C and add it to the collection of labeled documents.
  - For each class C, pick the unlabeled document about which classifier B is most confident that its class label is C and add it to the collection of labeled documents.
- **Output:** Two classifiers, A and B, that predict class labels for new documents. These predictions can be combined by multiplying together and then renormalizing their class probability scores.

Table 1: The co-training algorithm described in Section 3.3.

CoEM/wvRN/HF

- One algorithm with several justifications….
- One is to start with co-training with a naïve Bayes learner
- And compare to an EM version of naïve Bayes
  - E: soft-classify unlabeled examples with NB classifier
  - M: re-train classifier with soft-labeled examples

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	788	–0–	3.3%
Co-training	12	776	5.4%
EM	12	776	4.3%
Naive Bayes	12	–0–	13.0%

CoEM/wvRN/HF

- A second experiment
  - each + example: concatenate features from two documents, one of class A+, one of class B+
  - each - example: concatenate features from two documents, one of class A-, one of class B-
  - features are prefixed with "A", "B" ➔ disjoint

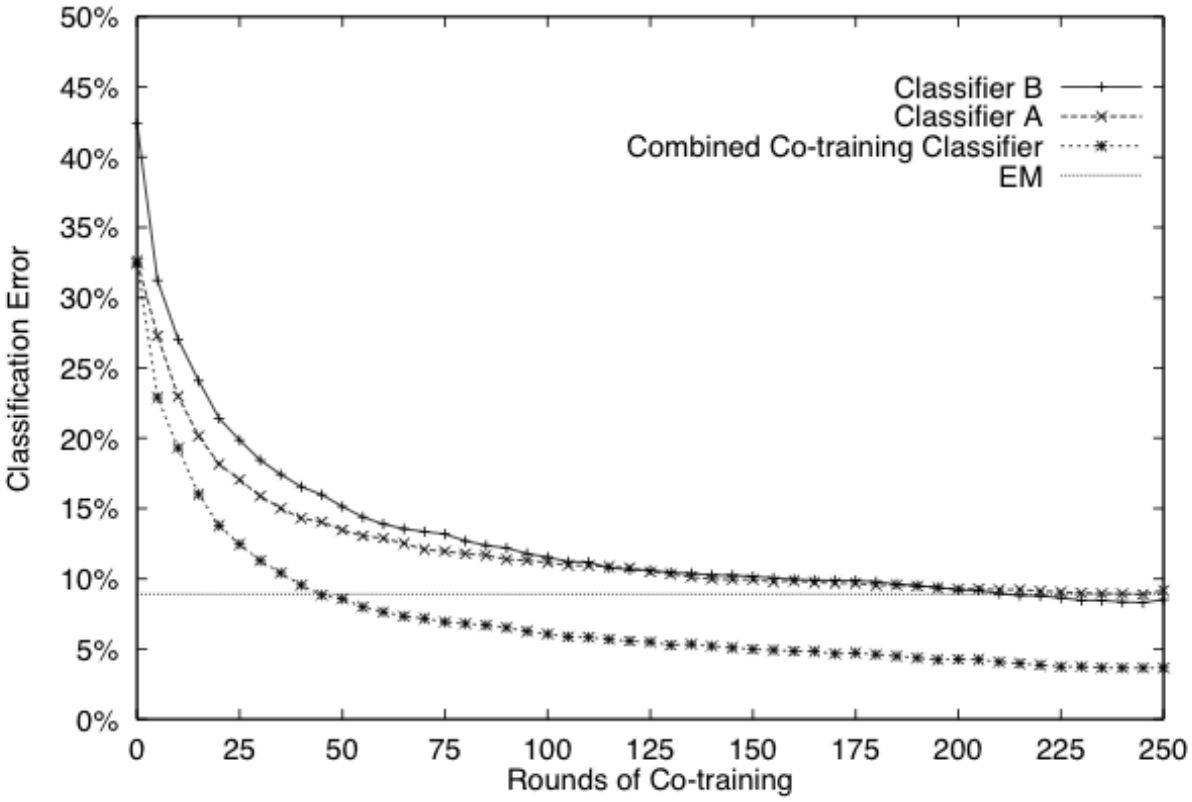
Table 3: The setup of the News 2x2 dataset. This data has class-conditional independence and redundancy between its two feature sets.

Class	Feature Set A	Feature Set B
Pos	comp.os.ms-windows.misc	talk.politics.misc
Neg	comp.sys.ibm.pc.hardware	talk.politics.guns

CoEM/wvRN/HF

- A second experiment
  - each + example: concatenate features from two documents, one of class A+, one of class B+
  - each - example: concatenate features from two documents, one of class A-, one of class B-
  - features are prefixed with "A", "B" ➔ disjoint
- NOW co-training outperforms EM

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	1006	–0–	3.9%
Co-training	6	1000	3.7%
EM	6	1000	8.9%
Naive Bayes	6	–0–	34.0%



CoEM/wvRN/HF

- Co-training with a naïve Bayes learner

- *vs* an EM version of naïve Bayes
  - E: soft-classify unlabeled examples with NB classifier
  - M: re-train classifier with soft-labeled examples

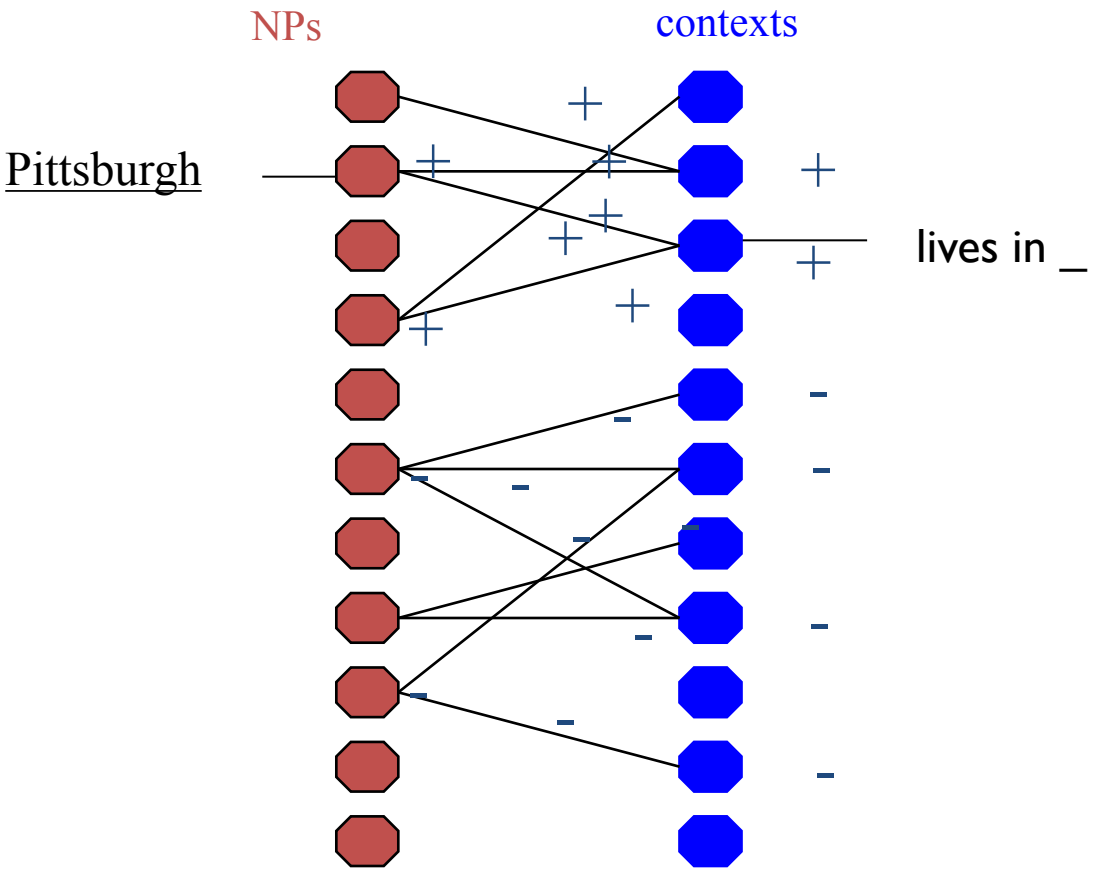
Method	Uses Feature Split? Yes	No
Incremental	co-training	self-training
Iterative	co-EM	EM

incremental hard assignments

iterative soft assignments

Method	Uses Random Feature Split? Yes	No
Incremental	5.5%	5.8%
Iterative	5.1%	8.9%

Co-EM for a Rote Learner: equivalent to HF on a bipartite graph



SSL AS OPTIMIZATION

SSL as optimization and Modified Adsorption

slides from Partha Talukdar

Notations

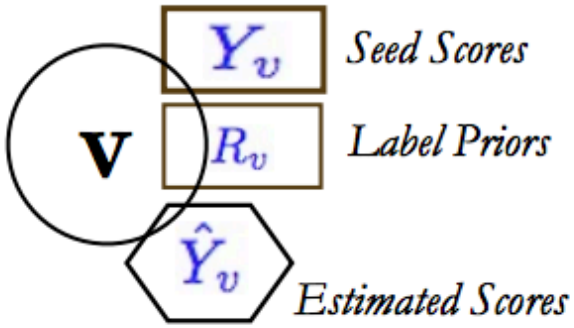
$\hat{Y}_{v,l}$: score of estimated label l on node v

$Y_{v,l}$: 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)

yet another name for HF/wvRN/coEM

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

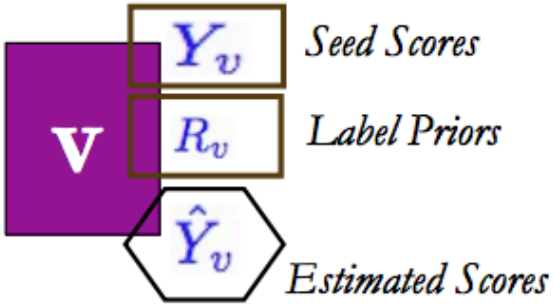
Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

match seeds | smoothness | prior

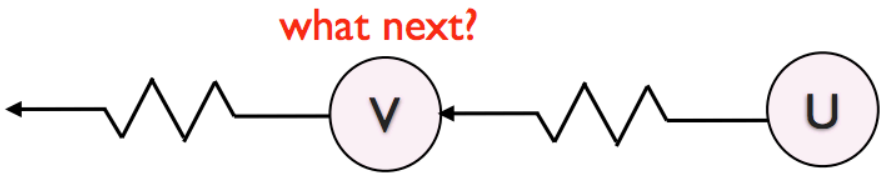
$$\arg\min_{\hat{\boldsymbol{Y}}} \sum_{l=1}^{m+1} \left[\|\boldsymbol{S}\hat{\boldsymbol{Y}}_l - \boldsymbol{S}\boldsymbol{Y}_l\|^2 + \mu_1 \sum_{u,v} \boldsymbol{M}_{uv}(\hat{\boldsymbol{Y}}_{ul} - \hat{\boldsymbol{Y}}_{vl})^2 + \mu_2 \|\hat{\boldsymbol{Y}}_l - \boldsymbol{R}_l\|^2 \right]$$

- 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



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

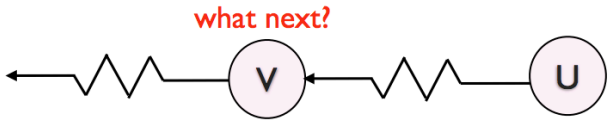
Adsorption SSL algorithm



- 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

- $\boldsymbol{M} = \boldsymbol{W}^{\top} + \boldsymbol{W}'$ 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

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

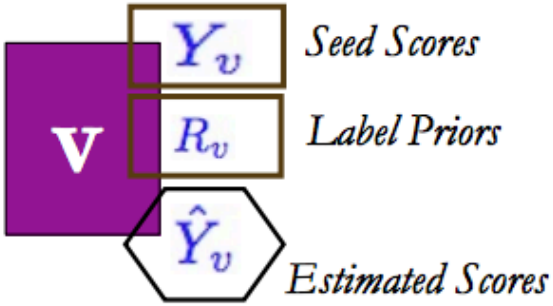
Dummy Label

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



Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

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

How to do this minimization?

First, differentiate to find min is at

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

Jacobi method:

- To solve Ax=b for x
- Iterate:

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

- … or:

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

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

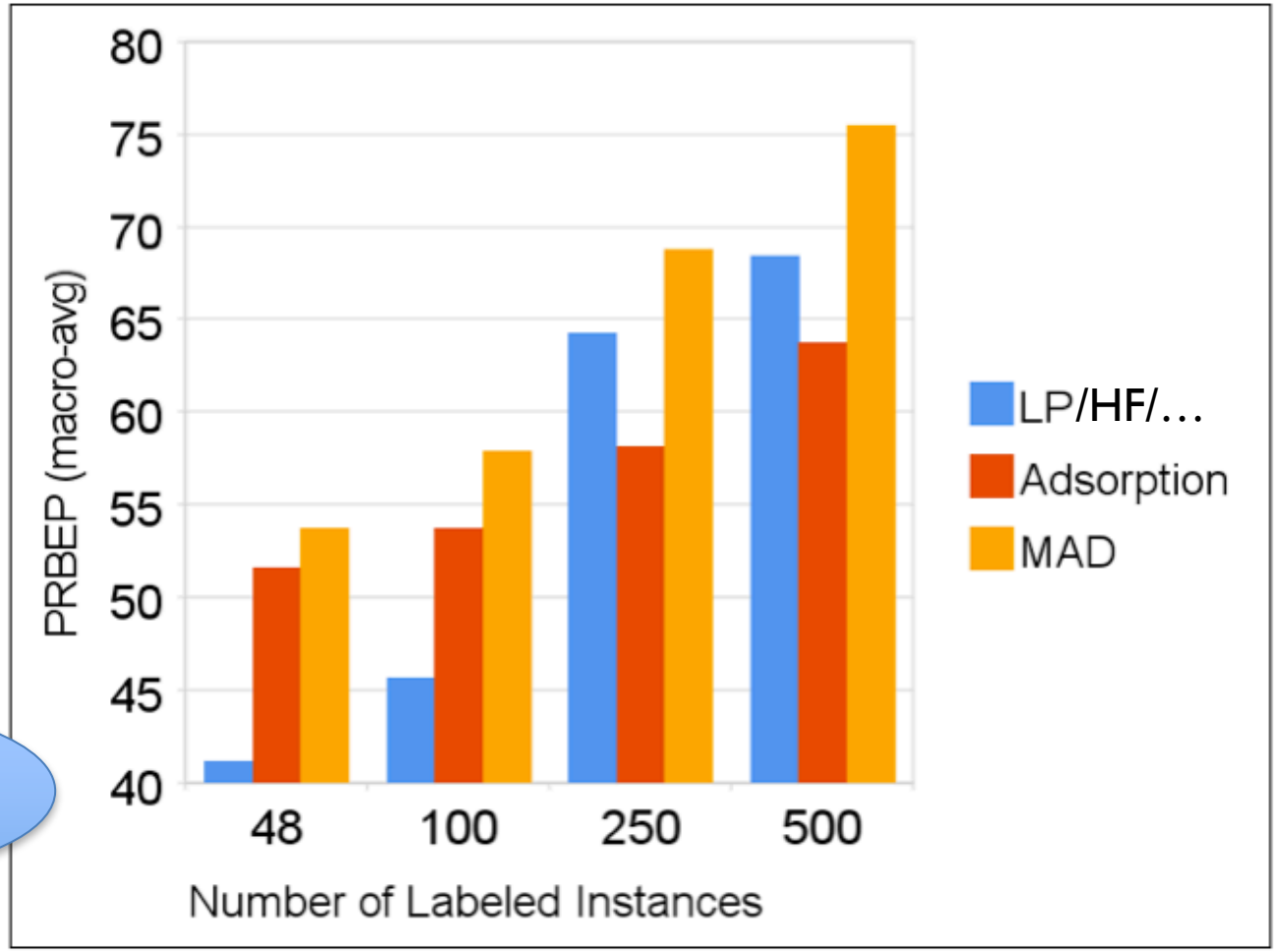
- Extends Adsorption with well-defined optimization
- Importance of a node can be discounted
- Easily Parallelizable: Scalable

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

Code in Junto Label Propagation Toolkit
(includes Hadoop-based implementation)
http://code.google.com/p/junto/

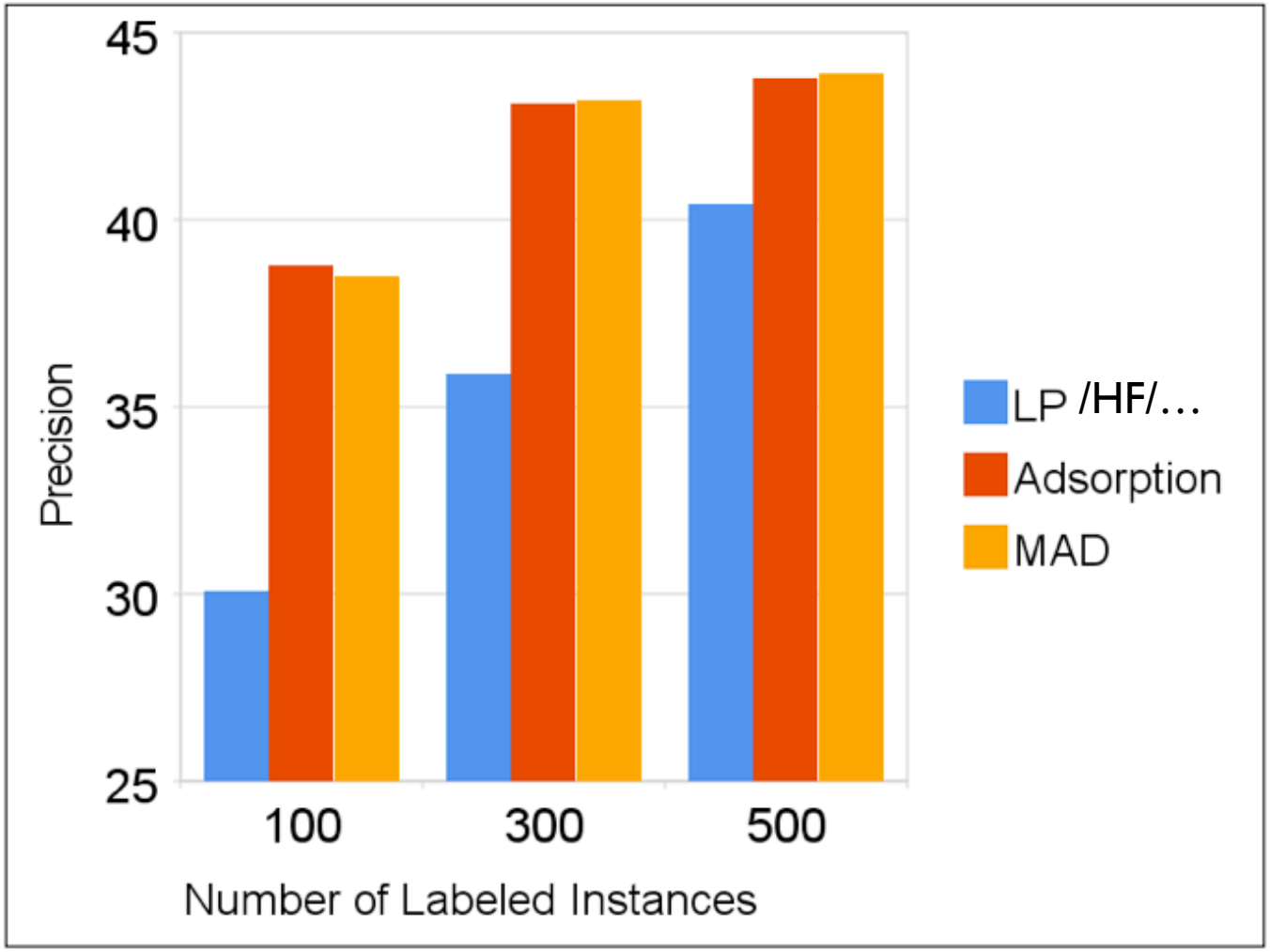
Text Classification



precision-recall break even point

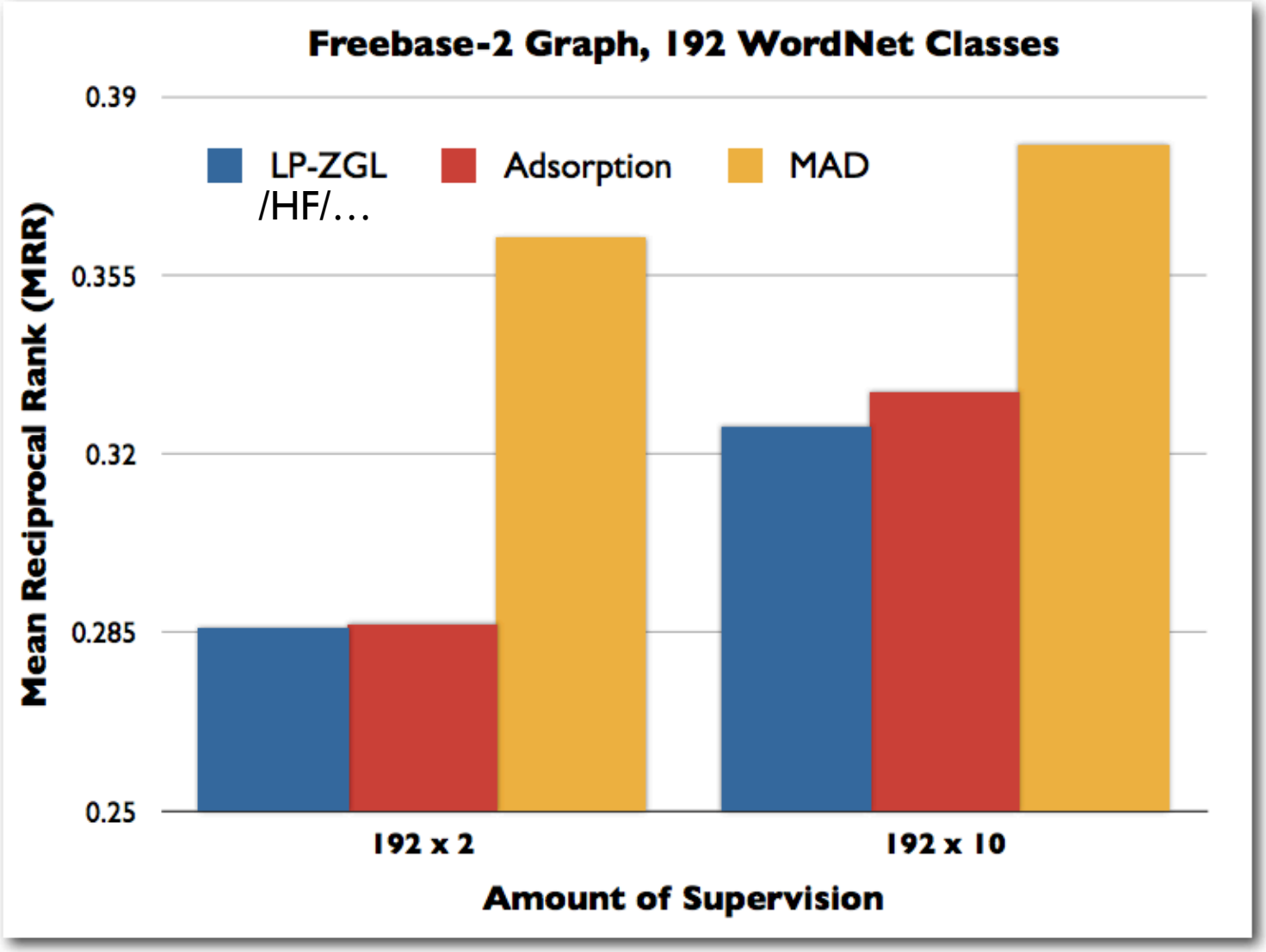
PRBEP (macro-averaged) on WebKB Dataset, 3148 test instances

Sentiment Classification



Precision on 3568 Sentiment test instances

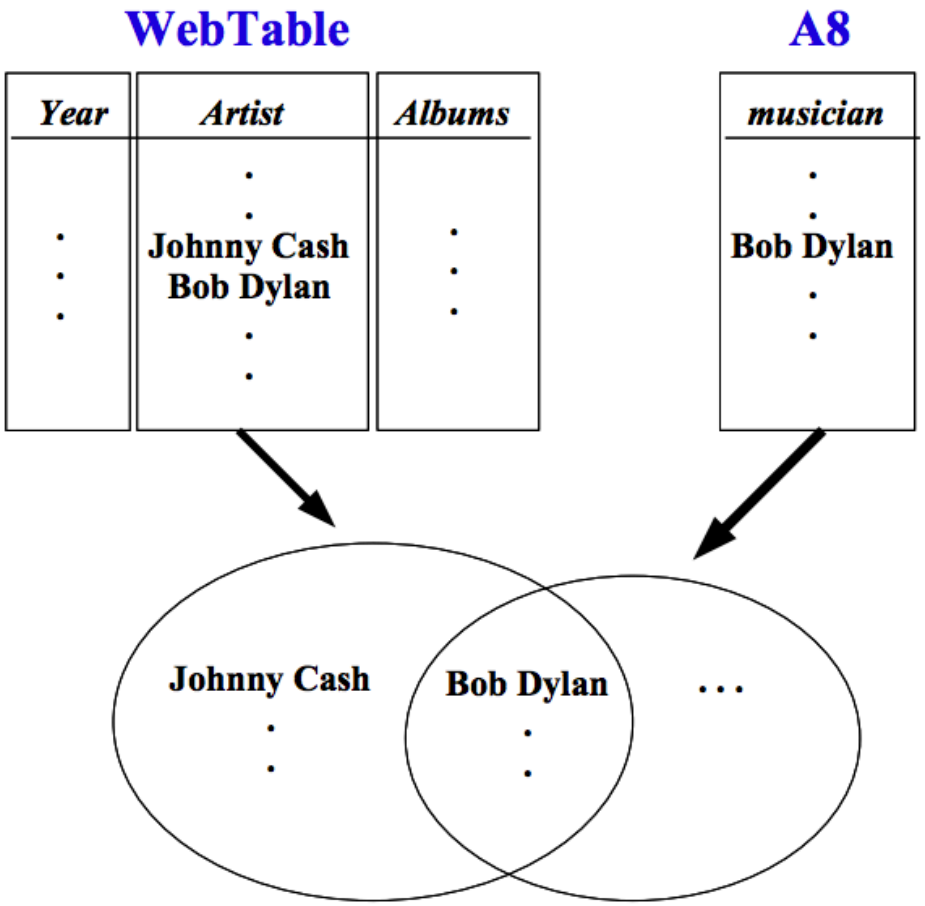
Class-Instance Acquisition



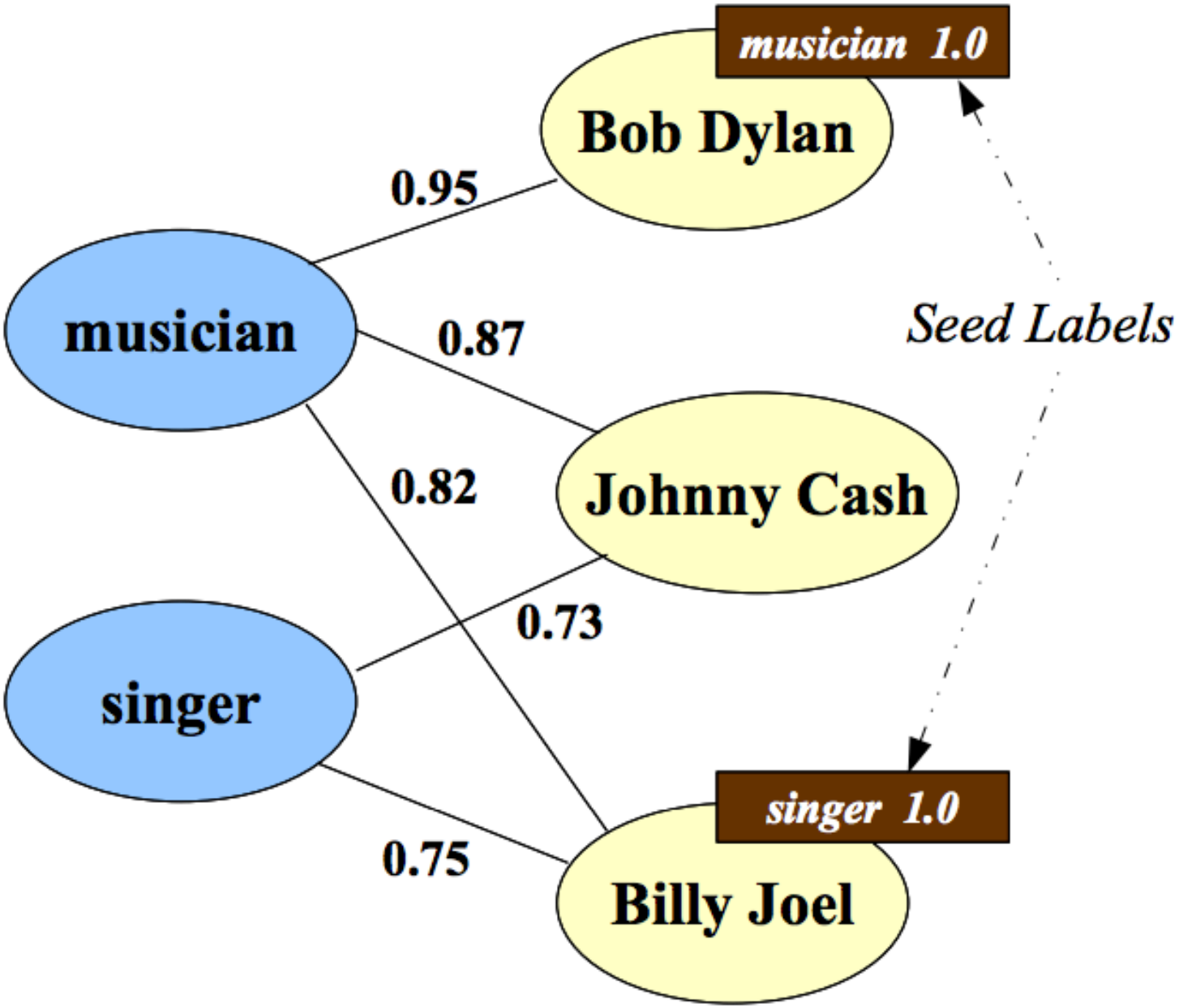
Graph with 303k nodes, 2.3m edges.

Assigning class labels to WebTable instances

from HTML tables on the web that are used for data, not formatting



from mining patterns like "musicians such as Bob Dylan"


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**

MAD SKETCHES

Followup work (AIStats 2014)

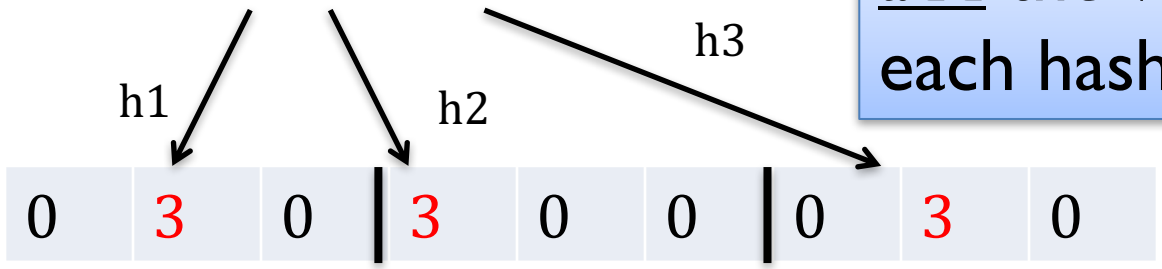
- Propagating labels requires usually small number of optimization passes
  - Basically like label propagation passes
- Each is linear in
  - the number of edges
  - and the number of labels being propagated
- Can you do better?
  - basic idea: store labels in a countmin sketch
  - which is basically an compact approximation of an object→double mapping

Count-min sketches

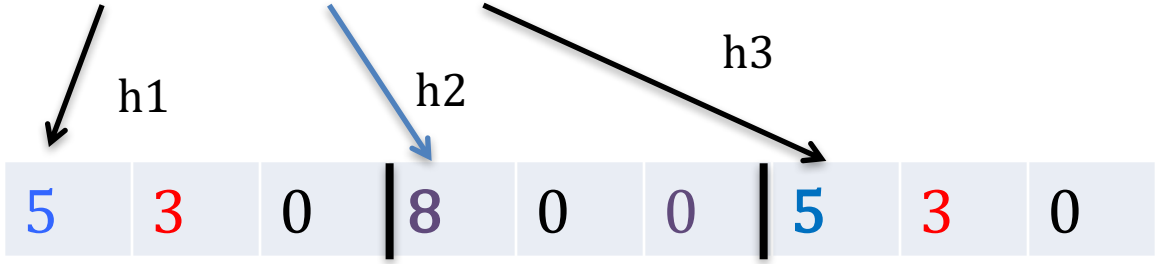
split a real vector into k ranges, one for each hash function

cm.inc("fred flintstone", 3):

add the value to each hash location



cm.inc("barney rubble",5):

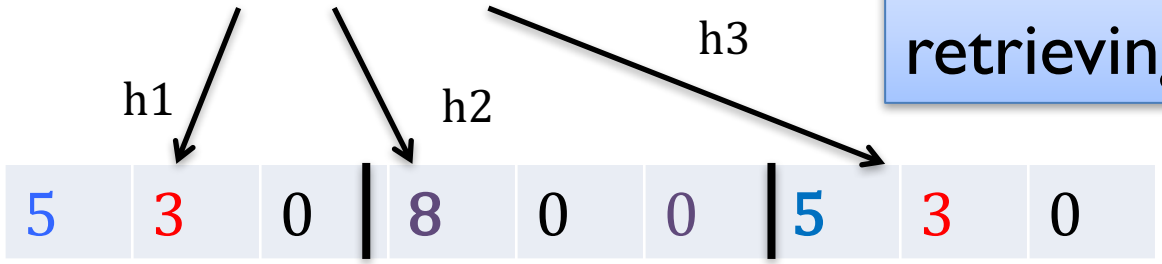


Count-min sketches

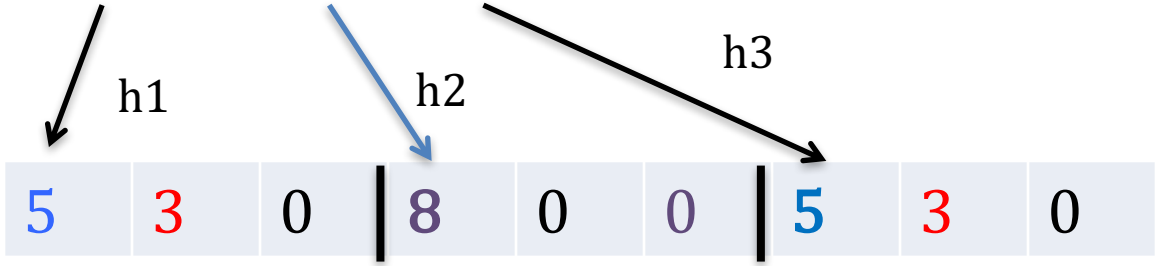
split a real vector into k ranges, one for each hash function

cm.get("fred flintstone"): 3

take min when retrieving a value



cm.get("barney rubble): 5

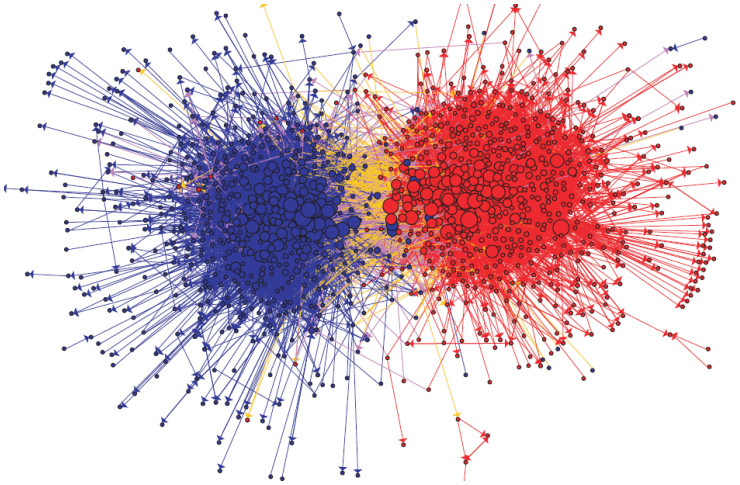


Followup work (AIStats 2014)

- Propagating labels requires usually small number of optimization passes
  - Basically like label propagation passes
- Each is linear in
  - the number of edges
  - ~~and the number of labels being propagated~~
  - the sketch size
  - sketches can be combined linearly without "unpacking" them: sketch($a\mathbf{v} + b\mathbf{w}$) = a*sketch($\mathbf{v}$)+b*sketch($\mathbf{w}$)
  - sketchs are good at storing *skewed distributions*

Followup work (AIStats 2014)

- Label distributions are often very skewed
  - sparse initial labels
  - community structure: labels from other subcommunities have small weight

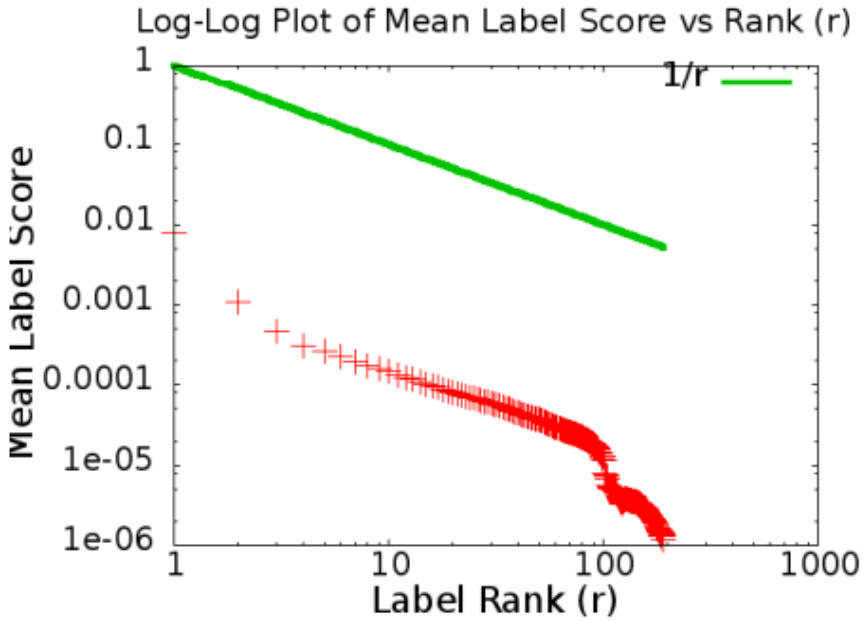


Followup work (AIStats 2014)

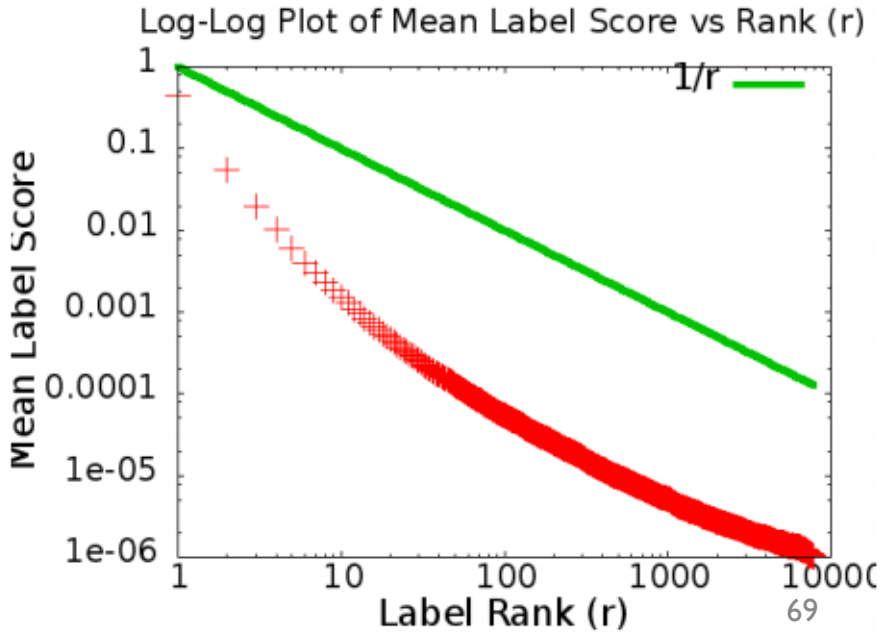
"self-injection": similarity computation

Name	Nodes (n)	Edges	Labels (m)	Seed Nodes	$k-$Sparsity	$\lceil \frac{ek}{\epsilon} \rceil$	$\lceil \ln \frac{m}{\delta} \rceil$
Freebase	301,638	1,155,001	192	1917	2	109	8
Flickr-10k	41,036	73,191	10,000	10,000	1	55	12
Flickr-1m	1,281,887	7,545,451	1,000,000	1,000,000	1	55	17

Freebase



Flick-10k



Followup work (AIStats 2014)

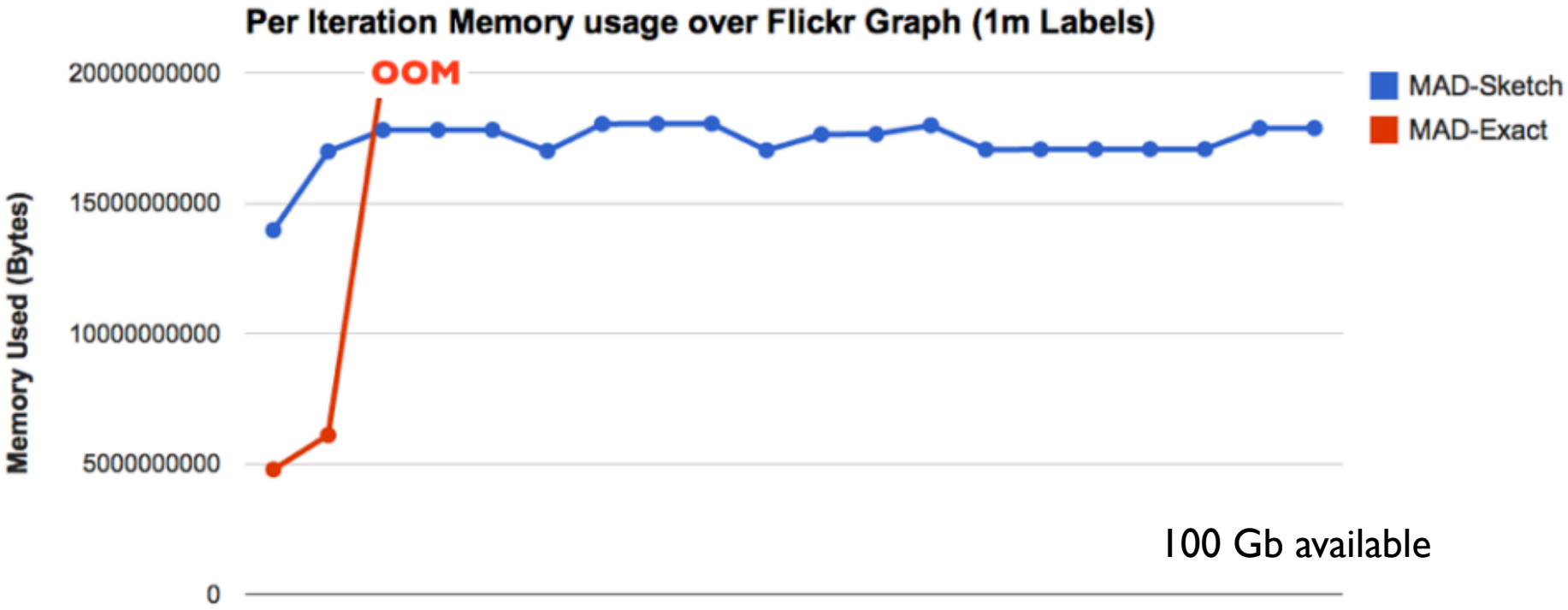
Name	Nodes (n)	Edges	Labels (m)	Seed Nodes	$k-$Sparsity	$\lceil \frac{ek}{\epsilon} \rceil$	$\lceil \ln \frac{m}{\delta} \rceil$
Freebase	301,638	1,155,001	192	1917	2	109	8
Flickr-10k	41,036	73,191	10,000	10,000	1	55	12
Flickr-1m	1,281,887	7,545,451	1,000,000	1,000,000	1	55	17

	Average Memory Usage (GB)	Total Runtime (s) [Speedup w.r.t. MAD-Exact]	MRR
MAD-Exact	3.54	516.63 [1.0]	0.28
MAD-Sketch ($w = 109, d = 8$)	2.68	110.42 [4.7]	0.28
MAD-Sketch ($w = 109, d = 3$)	1.37	54.45 [9.5]	0.29
MAD-Sketch ($w = 20, d = 8$)	1.06	47.72 [10.8]	0.28
MAD-Sketch ($w = 20, d = 3$)	1.12	48.03 [10.8]	0.23

Freebase

Followup work (AIStats 2014)

Name	Nodes (n)	Edges	Labels (m)	Seed Nodes	$k-$Sparsity	$\lceil \frac{ek}{\epsilon} \rceil$	$\lceil \ln \frac{m}{\delta} \rceil$
Freebase	301,638	1,155,001	192	1917	2	109	8
Flickr-10k	41,036	73,191	10,000	10,000	1	55	12
Flickr-1m	1,281,887	7,545,451	1,000,000	1,000,000	1	55	17



Even more recent work

Large Scale Distributed Semi-Supervised Learning Using Streaming Approximation

Sujith Ravi
Google Inc., Mountain View, CA, USA
sravi@google.com

Qiming Diao[1]
Carnegie Mellon University, Pittsburgh, PA, USA
Singapore Mgt. University, Singapore
qiming.ustc@gmail.com

AIStats 2016

Differences: objective function

$$\mathcal{C}(\hat{\mathbf{Y}}) = \mu_1 \sum_{v \in V_l} s_{vv} ||\hat{\mathbf{Y}}_\mathbf{v} - \mathbf{Y}_\mathbf{v}||_2^2$$

$$+ \mu_2 \sum_{v \in V, u \in \mathcal{N}(v)} w_{vu} ||\hat{\mathbf{Y}}_\mathbf{v} - \hat{\mathbf{Y}}_\mathbf{u}||^2$$

$$+ \mu_3 \sum_{v \in V} ||\hat{\mathbf{Y}}_\mathbf{v} - \mathbf{U}||_2^2$$

$$s.t. \sum_{l=1}^{L} \hat{Y}_{vl} = 1, \forall v$$

seeds

smoothness

close to uniform label distribution

normalized predictions

Differences: scaling up

- Updates done in parallel with Pregel
- Replace count-min sketch with "streaming approach"
  - updates from neighbors are a "stream"
    - break stream into sections
      - maintain a list of $(y, Prob(y), \Delta)$
      - filter out labels and end of section if $Prob(y)+\Delta$ is small

Results with EXPANDER