Half-Transductive Ranking

Bing Bai[1]
Ronan Collobert[1]
Jason Weston[2]
Corinna Cortes[2]
David Grangier[1]
Mehryar Mohri[2][3]

[1] NEC Labs America, Princeton, NJ
[2] Google Research, New York, NY
[3] NYU Courant Institute, New York, NY

Outline

- Learning to Rank: Functional & Transductive Rankings
- Half-Transductive Ranking
- Experiments & Results
- Summary

Outline

- **Learning to Rank: Functional & Transductive Rankings**
- Half-Transductive Ranking
- Experiments & Results
- Summary

Learning to Rank

Framework

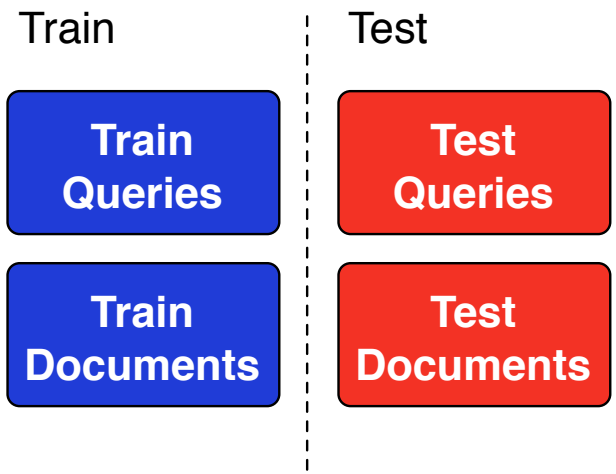
Input	document set D, query q
Output	document ranking s.t. most relevant document on top
Model	scoring function: query $\times$ doc $\rightarrow \mathbb{R}$
	sort decreasingly $\{\text{score}(q, d), \quad \forall d \in D\}$

Our Goal

- Benefit from the overlap between train & test documents.

Functional Ranking

- Generalizing to new documents and new queries.

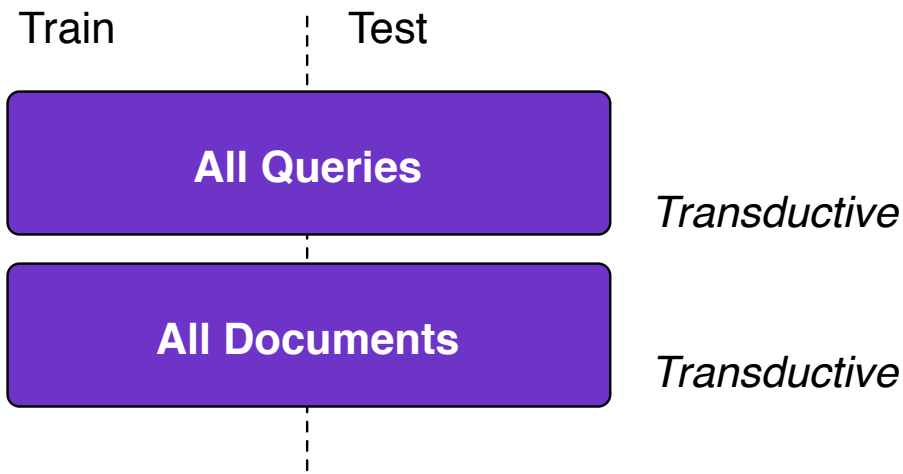


- $\text{score}(q, d)$ is a function of features $\phi(q, d)$.

e.g. Ranking Perceptron, Ranking SVM, RankNet, SoftRank; LSI, . . .

Transductive Ranking

- Generalizing to new rankings over a fixed object set.



- Each object (document/query) is assigned a parameter vector.
- $\mathrm{score}(q, d) =$ distance between learned vectors.
- Learning focuses on proximity information, not on features.
- Typically no out-of-sample extension.

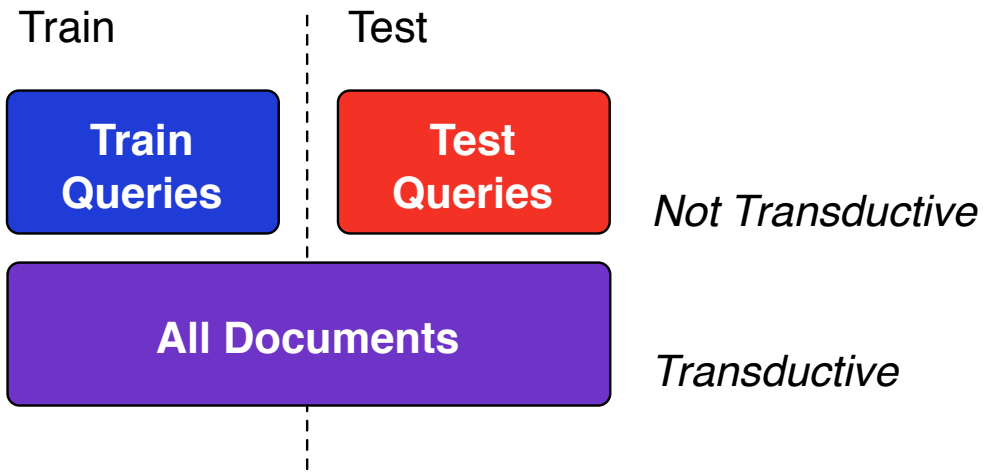
e.g. Multi-dimensional Scaling, IsoMap, Locally Linear Embedding...

Outline

- Learning to Rank: Functional & Transductive Rankings
- **Half-Transductive Ranking**
- Experiments & Results
- Summary

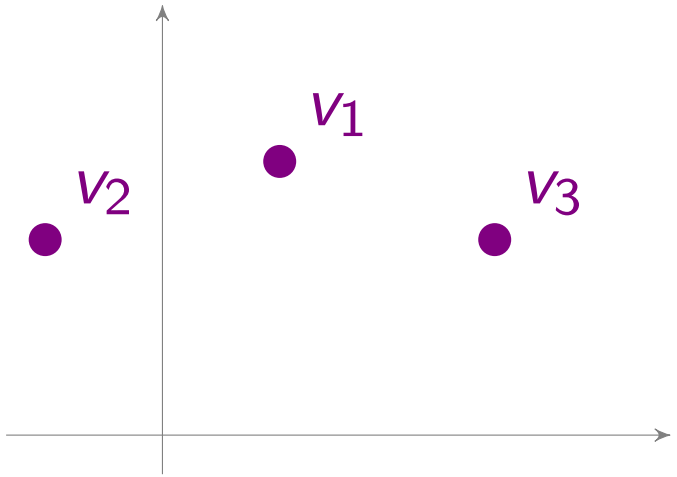
Half-Transductive Ranking

- Generalizing to new queries for a fixed document set.
- Training queries and test queries are distinct.
- All documents are available at training time.



New learning problem !

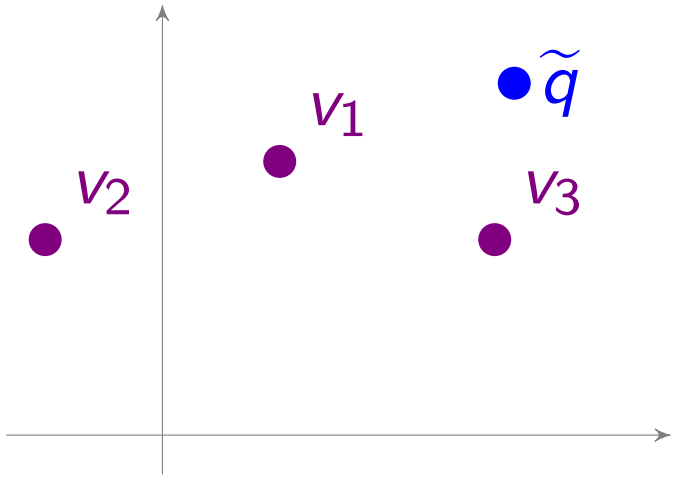
Half-Transductive Ranking



Embedding documents and queries in an n-dimensional space

- documents are embedded transductively
  For each $d_i \in D$, we learn a parameter vector $v_i \in \mathbb{R}^n$

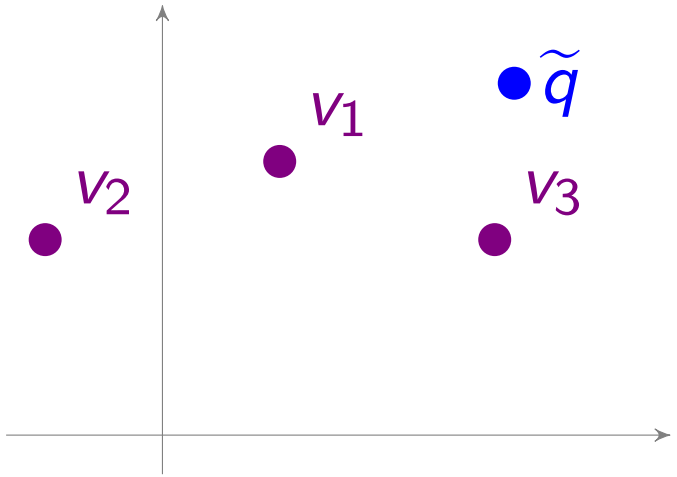
Half-Transductive Ranking



Embedding documents and queries in an n-dimensional space

- documents are embedded transductively
  For each $d_i \in D$, we learn a parameter vector $v_i \in \mathbb{R}^n$
- queries are embedded functionally
  We learn a parameter matrix W and project queries as $\widetilde{q} = Wq$

Half-Transductive Ranking



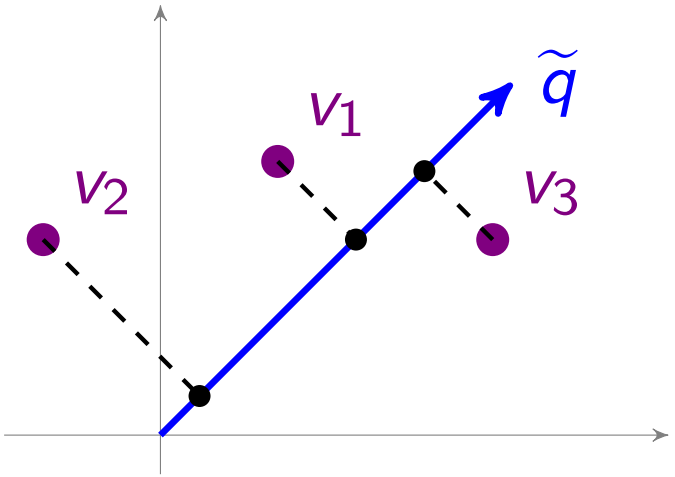
Embedding documents and queries in an n-dimensional space

- documents are embedded transductively
  For each $d_i \in D$, we learn a parameter vector $v_i \in \mathbb{R}^n$
- queries are embedded functionally
  We learn a parameter matrix W and project queries as $\widetilde{q} = Wq$

- ranking scores $\leftrightarrow$ dot-product in the embedding space

$$\text{score}(q, d_i) = \langle Wq, v_i \rangle = q^\top W^\top v_i$$

Half-Transductive Ranking



ranking	
1^{st}	d_3
2^{nd}	d_1
3^{rd}	d_2

Embedding documents and queries in an n-dimensional space

- documents are embedded transductively
  For each $d_i \in D$, we learn a parameter vector $v_i \in \mathbb{R}^n$
- queries are embedded functionally
  We learn a parameter matrix W and project queries as $\widetilde{q} = Wq$

- ranking scores $\leftrightarrow$ dot-product in the embedding space
  Document order is provided by the projection of v along $\widetilde{q}$

Half-Transductive Ranking: Supervision Types

We consider three training algorithms for three kinds of setup:

1. Unsupervised: only objects available for training.

2. Objects + proximity information available for training.

3. Objects + supervision via Known Preference Relations.

Half-Transductive Ranking: Supervision Types

We consider three training algorithms for three kinds of setup:

1. Unsupervised: only objects available for training.
   Nonlinear version of classical functional approach: LSI...

2. Objects + proximity information available for training.
   Natural out-of-sample extension for classical transductive approaches.

3. Objects + supervision via Known Preference Relations.
   Nonlinear version of classical functional approaches: Ranking Perceptron/SVM, etc.

Half-Transductive Ranking: Known Preference Relations

Learning of $\{v_i\}_{i=1}^m$ and W from preference triplets

- (q, d^+, d^-) means d^+ *is prefered to* d^- *for query* q.
- pairwise hinge loss

$$L^{\mathrm{HTR}} = \sum_{(q,d^+,d^-)} \max(0, 1 - \mathrm{score}(q, d^+) + \mathrm{score}(q, d^-))$$

Half-Transductive Ranking: Known Preference Relations

Learning of $\{v_i\}_{i=1}^m$ and W from preference triplets

- (q, d^+, d^-) means d^+ *is prefered to* d^- *for query* q.
- pairwise hinge loss

$$L^{\mathrm{HTR}} = \sum_{(q,d^+,d^-)} \max(0, 1 - q^\top W^\top v^+ + q^\top W^\top v^-)$$

Half-Transductive Ranking: Known Preference Relations

Learning of $\{v_i\}_{i=1}^m$ and W from preference triplets

- (q, d^+, d^-) means d^+ *is prefered to* d^- *for query* q.
- pairwise hinge loss

$$L^{\mathrm{HTR}} = \sum_{(q,d^+,d^-)} \max(0, 1 - q^\top W^\top v^+ + q^\top W^\top v^-)$$

Regularization

- push the half-transductive solution toward the functional solution

$$L = L^{\mathrm{HTR}} + \gamma \sum_{(q,d^+,d^-)} \max(0, 1 - q^\top W^\top W d^+ + q^\top W^\top W d^-)$$

Half-Transductive Ranking: Known Preference Relations

Learning of $\{v_i\}_{i=1}^m$ and W from preference triplets

- (q, d^+, d^-) means d^+ *is prefered to* d^- *for query* q.
- pairwise hinge loss

$$L^{\mathrm{HTR}} = \sum_{(q,d^+,d^-)} \max(0, 1 - q^\top W^\top v^+ + q^\top W^\top v^-)$$

Regularization

- push the half-transductive solution toward the functional solution

$$L = L^{\mathrm{HTR}} + \gamma \sum_{(q,d^+,d^-)} \max(0, 1 - q^\top W^\top W d^+ + q^\top W^\top W d^-)$$

Learning algorithm

- stochastic gradient descent

Half-Transductive Ranking: Graph-based learning

Learn nonlinear relationships via a graph (adjacency matrix A).

- **Standard Transductive objective:**

$$L^{\mathrm{TR}} = \sum_{i,j} L(v_i, v_j, A_{ij})$$

Half-Transductive Ranking: Graph-based learning

Learn nonlinear relationships via a graph (adjacency matrix A).

- **Standard Transductive objective:**

$$L^{\mathrm{TR}} = \sum_{i,j} L(v_i, v_j, A_{ij})$$

$$\text{We use:} \quad L(z, z', A_{ij}) = \begin{cases} ||z - z'||_1 & \text{if } A_{ij} = 1, \\ \max(0, 1 - ||z - z'||_1) & \text{if } A_{ij} = 0 \end{cases}$$

Half-Transductive Ranking: Graph-based learning

Learn nonlinear relationships via a graph (adjacency matrix A).

- **Standard Transductive objective:**

$$L^{\mathrm{TR}} = \sum_{i,j} L(v_i, v_j, A_{ij})$$

$$\text{We use:} \quad L(z, z', A_{ij}) = \begin{cases} ||z - z'||_1 & \text{if } A_{ij} = 1, \\ \max(0, 1 - ||z - z'||_1) & \text{if } A_{ij} = 0 \end{cases}$$

+ Inductive (out-of-sample extension) part of objective:

- $$L = \gamma L^{\mathrm{TR}} + \sum_{i,j} L(W\phi(y_i), v_j, A_{ij})$$

Half-Transductive Ranking: Graph-based learning

Learn nonlinear relationships via a graph (adjacency matrix A).

- **Standard Transductive objective:**

$$L^{\mathrm{TR}} = \sum_{i,j} L(v_i, v_j, A_{ij})$$

$$\text{We use:} \quad L(z, z', A_{ij}) = \begin{cases} ||z - z'||_1 & \text{if } A_{ij} = 1, \\ \max(0, 1 - ||z - z'||_1) & \text{if } A_{ij} = 0 \end{cases}$$

+ Inductive (out-of-sample extension) part of objective:

- $$L = \gamma L^{\mathrm{TR}} + \sum_{i,j} L(W\phi(y_i), v_j, A_{ij})$$

Learning algorithm

- Learn Parameters $\{v_i\}_{i=1}^m$ and W by stochastic gradient descent

Half-Transductive Ranking: Unsupervised reconstruction

Learn low dimensional nonlinear "latent semantic" space for ranking.

- **Transductive reconstruction objective:** *transductive* point-wise representation v^+ reconstructs d^+ using linear mapping V.

$$L^{\mathrm{TR}} = \sum_{d^+} ||Vv^+ - d^+||^2$$

Half-Transductive Ranking: Unsupervised reconstruction

Learn low dimensional nonlinear "latent semantic" space for ranking.

- **Transductive reconstruction objective:** *transductive* point-wise representation v^+ reconstructs d^+ using linear mapping V.

$$L^{\mathrm{TR}} = \sum_{d^+} ||Vv^+ - d^+||^2$$

Half-Transductive Ranking: Unsupervised reconstruction

Learn low dimensional nonlinear "latent semantic" space for ranking.

- **Transductive reconstruction objective:** *transductive* point-wise representation v^+ reconstructs d^+ using linear mapping V.

$$L^{\mathrm{TR}} = \sum_{d^+} ||Vv^+ - d^+||^2$$

+ Inductive part of objective:

- v^+ should be closer to Wd^+ than any other Wd^-.

$$L = \gamma L^{\mathrm{TR}} + \sum_{d^- \neq d^+} \max(0, 1 - {v^+}^\top W d^+ + {v^+}^\top W d^-)$$

Half-Transductive Ranking: Unsupervised reconstruction

Learn low dimensional nonlinear "latent semantic" space for ranking.

- **Transductive reconstruction objective:** *transductive* point-wise representation v^+ reconstructs d^+ using linear mapping V.

$$L^{\mathrm{TR}} = \sum_{d^+} ||Vv^+ - d^+||^2$$

+ Inductive part of objective:

- v^+ should be closer to Wd^+ than any other Wd^-.

$$L = \gamma L^{\mathrm{TR}} + \sum_{d^- \neq d^+} \max(0, 1 - {v^+}^\top W d^+ + {v^+}^\top W d^-)$$

Learning algorithm

- Learn Parameters $\{v_i\}_{i=1}^m$ and W, V by stochastic gradient descent

Outline

- Learning to Rank: Functional & Transductive Rankings
- Half-Transductive Ranking
- **Experiments & Results**
- Summary

Experiments & Results : Supervised Half-Transduction

Context

Half-transductive ranking with Supervised preference relations.

Learn to rank from bag-of-word features

Baselines

Unsupervised	$tf \cdot idf$
	Latent Semantic Indexing (LSI)
Supervised	ranking perceptron with hash kernel [1]
	Supervised Semantic Indexing (SSI) [2] functional counterpart

[1] Shi, Petterson Langford, Smola, Strehl, Vishwanathan – AISTATS'09
[2] Bai, Weston, Grangier, Collobert, Qi, Sadamasa, Chapelle, Weinberger – CIKM'09

Experiments & Results : Supervised Half-Transduction

Wikipedia Dataset

- 2M documents, 24M links

Ranking Problem

- Related Document Search
- Links provide relevance labels and are not available as features
- Given a 'query' document q, find the documents d^+ to which q links.

Setup

- Vocabulary: 2.5M words
- Split: 70% train, 30% test

Experiments & Results : Supervised Half-Transduction

Algorithm	Rank-Err (%)	MAP	P@10
TFIDF	0.84	0.43	0.19
LSI	0.72	0.43	0.19
Perceptron	0.35	0.49	0.22
SSI	0.16	0.55	0.24
Half-Transductive	**0.11**	**0.61**	**0.26**

Experiments & Results : Supervised Half-Transduction

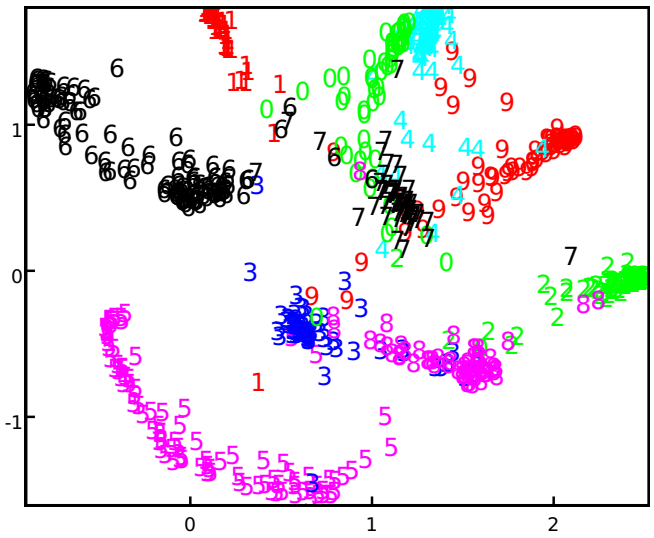
Algorithm	Rank-Err (%)	MAP	P@10	
TFIDF	0.84	0.43	0.19	
LSI	0.72	0.43	0.19	
Perceptron	0.35	0.49	0.22	supervision
SSI	0.16	0.55	0.24	
Half-Transductive	**0.11**	**0.61**	**0.26**	

Experiments & Results : Supervised Half-Transduction

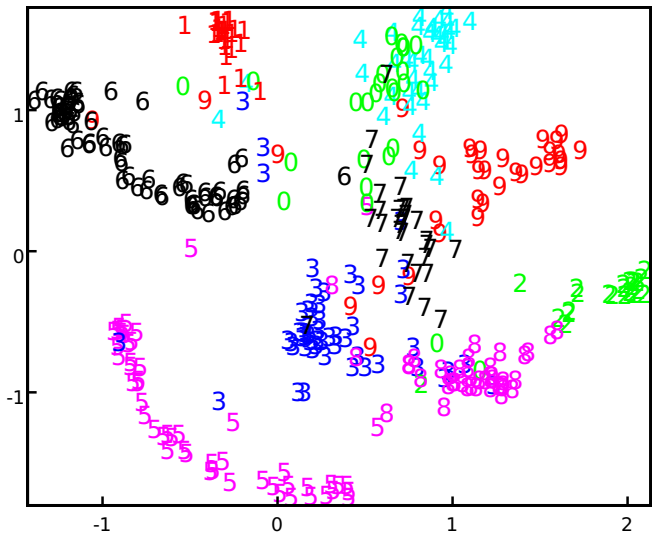
Algorithm	Rank-Err (%)	MAP	P@10	
TFIDF	0.84	0.43	0.19	
LSI	0.72	0.43	0.19	
Perceptron	0.35	0.49	0.22	
SSI	0.16	0.55	0.24	functional
Half-Transductive	**0.11**	**0.61**	**0.26**	half-transductive

Experiments & Results : Graph-based Learning

HTR Training set embedding:

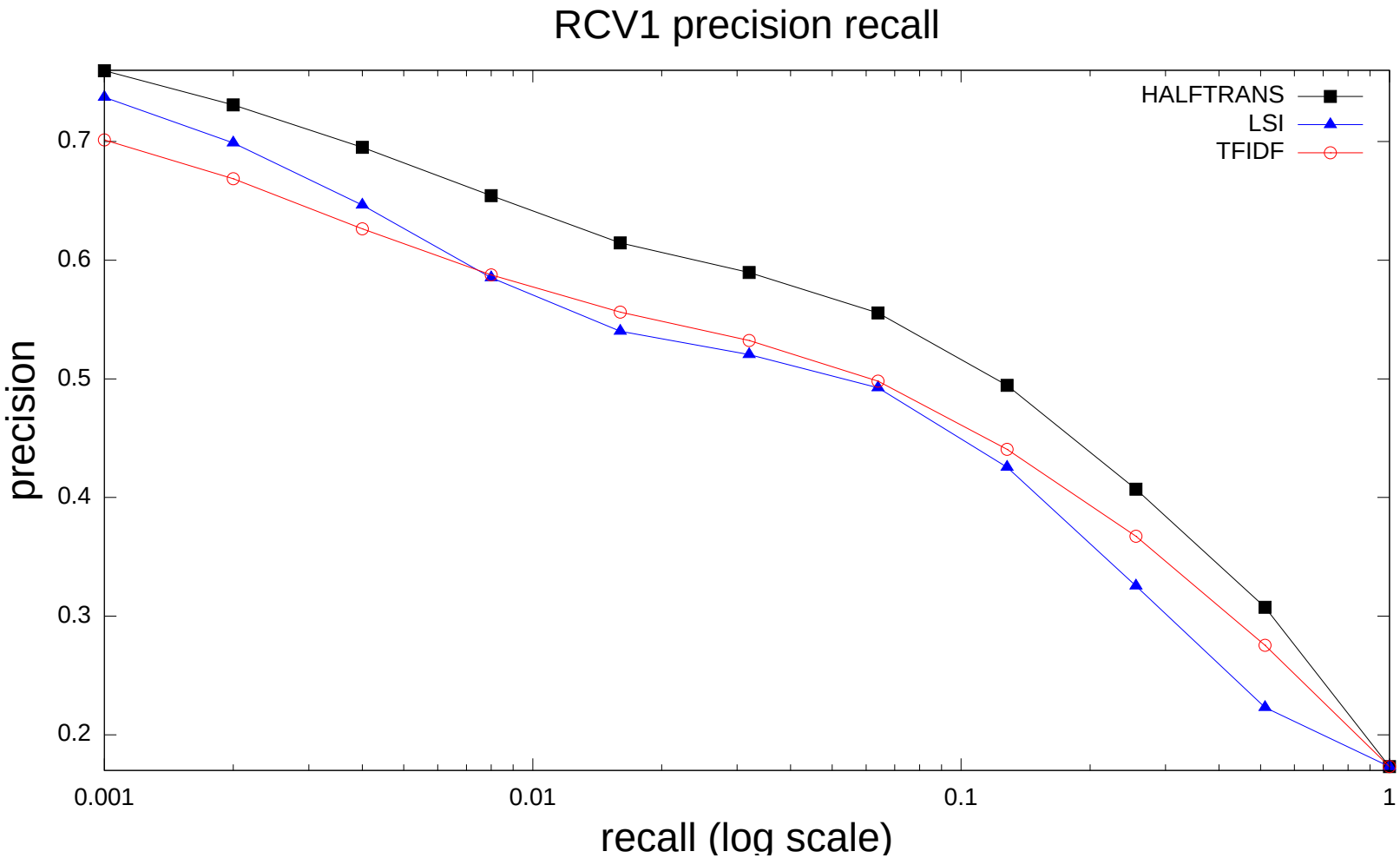


HTR Test set embedding:



Algorithm	1-NN Loss
Laplacian Eigenmaps (train+test) [Belkin & Niyogi, 2003]	0.0513
Laplacian Eigenmaps + O.S.E [Bengio et. al., 2003]	0.0510
1/2-Transductive LE (Linear)	0.0673
1/2-Transductive LE (RBF)	0.0508

Experiments: Reconstruct-based Unsupervised Learning



RCV II: 804,414 documents, 103 topics, 30k words, 50% train & test.
Unsupervised, but evaluate ranking using labels.

Outline

- Learning to Rank: Functional & Transductive Rankings
- Half-Transductive Ranking
- Experiments & Results
- **Summary**

Summary

Half-Transductive Learning

- $\neq$ functional: benefit from overlap between train & test documents.
- $\neq$ transductive: generalize to new queries.

Advantages

- Doc. representation not tied to features
- Empirical gain

Future Research

- Other loss functions, other applications (e.g. recommender systems)
- Late breaking AISTATS poster on image annotation extends this work.