Half Transductive Ranking

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

[1]NEC Labs America, Princeton, NJ
{bbai,dgrangier,collober}@nec-labs.com

[2] Google Research, New York, NY
{jweston,corinna,mohri}@google.com

[3] NYU Courant Institute, New York, NY
mohri@cs.nyu.edu

Abstract

We study the standard retrieval task of ranking a fixed set of items given a previously unseen query and pose it as the *half transductive* ranking problem. The task is *transductive* as the set of items is fixed. Transductive representations (where the vector representation of each example is learned) allow the generation of highly nonlinear embeddings that capture object relationships without relying on a specific choice of features, and require only relatively simple optimization. Unfortunately, they have no direct out-of-sample extension. *Inductive* approaches on the other hand allow for the representation of unknown queries. We describe algorithms for this setting which have the advantages of both *transductive* and *inductive* approaches, and can be applied in unsupervised (either reconstruction-based or graph-based) and supervised ranking setups. We show empirically that our methods give strong performance on all three tasks.

1 Introduction

The task of ranking a set of objects (e.g. documents) given a query is the core task of Information Retrieval. In most setups, the set of objects to rank is fixed (or slowly growing) while the set of submitted queries is unknown. This environment gives rise to an interesting learning problem, *Half Transductive Ranking* (HTR), in which all the objects to rank are available at training time, while the test queries are only revealed once the model is trained. At training time one may or may not have examples of desired (query, object ranking) pairs, but in either case unlike the objects to be ranked, the queries at test time may not have been seen before during the training phase. Hence, this setup is partly *transductive* since the item set is given in advance, and they are the only items the model will ever need to rank. This setup should not be confused with *semi-supervised* learning where one considers using auxiliary unlabeled data for a ranking task. For instance, supervised Latent Dirichlet Allocation sLDA (Blei and McAuliffe, 2007) relies on an additional categorical variable associated with each document, while (Duh and Kirchhoff, 2008) considers that some of the test queries (without relevance assessment) are available during training. Our setup does not require the availability of such information and deals with both supervised and unsupervised tasks.

Although frequent in Information Retrieval, previous literature mainly ignores the specific aspect of a fixed set of objects and considers the inductive setup in which the ranking model is learned to generalize to both new objects and new queries. Most ranking models learn a scoring function which assigns a real valued score to a feature vector describing a query/object pair. Given a query, the function is applied to each object and objects are ranked by decreasing scores. Models relying on this paradigm include ranking perceptrons or SVMs (Joachims, 2002), rankNet (Burges *et al.*, 2005) or ListNet (Cao *et al.*, 2007), amongst others. Hence, these approaches can be referred to as learning a *functional* embedding. Methods such as Latent Semantic Indexing (LSI) (Deerwester *et al.*, 1990) or Locality Preserving Projections (LPP) (He and Niyogi, 2003) can also be included in this class of algorithms.

In contrast, transductive approaches have been proposed in the literature, where the learnt similarity measure can only be assessed between objects given at training time. These approaches, such as as Isomap

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 6 of JMLR: W&CP 6.

(Balasubramanian *et al.*, 2002), Locally Linear Embedding (Roweis and Saul, 2000) or Laplacian Eigenmaps (Belkin and Niyogi, 2003), perform nonlinear dimensionality reduction by learning a vector for each object available during training. Compared to inductive approaches, transductive strategies allow the generation of highly nonlinear embeddings, while relying on simple optimization. Moreover, they focus on the characteristics of the relationship between the objects rather than relying on characteristics that might depend on a specific choice of features.

Although appealing, these methods can hardly be applied in our case, as our setup is only *half transductive*: all objects are available at training time but the test queries are not. Of course, one could advocate for using out-of-sample extensions that allow embedding a new example into a learned space, see e.g. (Bengio *et al.*, 2003; Trosset and Priebe, 2008). Such a strategy is however not desirable in a retrieval context since: (i) it requires solving a (potentially expensive) optimization problem after the submission of a new query, which is the most time critical period for a retrieval system; and (ii) a high quality representation of the query is integral to the task and should not be seen as an "extension" to the algorithm, hence one would like to learn the transductive and inductive representations jointly at training time.

This paper proposes a direct solution to the Half Transductive Ranking problem by directly optimizing a ranking function which consists of a functional embedding for queries and a transductive embedding for objects. This approach hence benefits from the advantages of the transductive approaches mentioned above, while retaining the generalization ability of inductive approaches for coping with new queries.

In the following, we present how the proposed approach can be applied in various contexts: in (i) an unsupervised setup, only objects are available for training, (ii) a graph setup, both objects and proximity information between objects are available for training, (iii) a supervised setup, objects and training queries along with corresponding relevance information are available for training. For each setup, our experiments demonstrate the effectiveness of our strategy compared to alternative approaches: our method outperforms inductive methods and provides the ability to use transductive methods where they would otherwise not apply. As a further advantage, our experiments also stress the scalability of the proposed learning strategy which allows learning a transductive representation, even when dealing with millions of transductive objects.

The remainder of the paper is organized as follows: Section 2 describes the proposed approach, Section 3 presents experimental results and Section 4 concludes.

2 Inductive, Transductive and Half Transductive Ranking

We are given a fixed set of m items $y_1, \ldots, y_m$. We consider the task of ranking these items given a new query x, unknown at training time.

The following presents inductive and transductive ranking and then introduces the notion of half transductive ranking, which is a mixture of inductive and transductive learning, that has the advantages of both methods.

Inductive ranking A typical *inductive* approach represents the query x and items y_i using a joint feature representation $\Phi(x, y_i) \in \mathbb{R}^d$ and then one ranks an item y_i given x using, for instance, a linear function:

$$f(x, y_i) = w \cdot \Phi(x, y_i).$$

Examples of this approach include the margin ranking perceptron (Collins and Duffy, 2001), e.g. applied to re-ranking parse trees, or SVM-MAP (Yue *et al.*, 2007), and metric learning algorithms like (Weinberger and Saul, 2008) can also be put in this class.

Ranking methods such as employing LSI as an embedding step also fall under this category. In that case the representations for query and target are separate bags of words $\phi(x)$ and $\phi(y_i)$ where the target y_i is ranked with:

$$f(x, y_i) = \phi(x)^\top W^\top W \phi(y_i)$$

where W is a linear embedding[1]. In contrast to methods like SVM-MAP, the parameters of LSI are learnt with an unsupervised reconstruction objective. Whether supervised or unsupervised learning is involved, typically inductive methods are linear models due to the computational cost of kernel methods when ranking a large set of objects. For example, ranking SVM or SVM-MAP are typically used in a linear setup (Joachims, 2002; Yue *et al.*, 2007). Nonlinear neural networks have also been used over a relatively small number of features (Burges *et al.*, 2005; Salakhutdinov and Hinton, 2007).

Transductive ranking A *transductive* approach assigns a vector $v_i \in \mathbb{R}^n$ to each object y_i that will be learnt using a supervised or unsupervised signal and crucially *does not involve any feature representation* $\Phi(\cdot)$ *at all.* In this sense, it can be said to be a nonlinear method. Ranking is then typically achieved by

[1] Typically W and $\phi(\cdot)$ are normalized to make this a cosine similarity.

measuring the distances in the embedding space $\mathbb{R}^n$

$$f(y_i, y_j) = ||v_i - v_j||_2$$

although other distance metrics than the 2-norm given here can be used. Note that the query must be one of the objects y_i for this approach to make sense as only these objects are embedded, hence the *transductive* name (Vapnik, 1998). This technique provides a point-to-point correspondence between the input space and the intrinsic space in which the data lie. Such methods have multiplied in recent years and are most commonly rooted either in factor analysis (e.g. principal component analysis) or multidimensional scaling, including the following methods and their variants: kernel PCA (Scholkopf *et al.*, 1999), Isomap (Balasubramanian *et al.*, 2002), Locally Linear Embedding (Roweis and Saul, 2000) and Laplacian Eigenmaps (Belkin and Niyogi, 2003). For example the latter embeds points by minimizing the function

$$\sum_{ij} L(v_i, v_j, A_{ij}) = \sum_{ij} A_{ij} ||v_i - v_j||^2$$

with respect to A, where A is a similarity ("affinity") matrix, under constraints that ensure a trivial solution is not reached. An overall review of this family of methods can be found in (Lee and Verleysen, 2007).

Half transductive ranking In this paper we propose a half transductive ranking algorithm that has the advantage of both the inductive and transductive approaches.

We start by, as in transductive approaches, defining a vector v_i for each item y_i in the fixed set to be ranked. We also introduce a function $x \to W\phi(x)$ which can project any query, including queries available solely at test time, from its feature vector $\phi(x)$. We then rely on the dot-product as a scoring function,

$$f(x, y_i) = v_i^\top W \phi(x)$$

where $W \in \mathbb{R}^{n \times d}$, as well as $v \in \mathbb{R}^{m \times n}$ are the parameters to be learned and d is the dimension of the chosen *functional* feature space. The choice of the dot-product as a scoring function simplifies gradient computation for training. However, Euclidean distances or other metrics could be used as well.

We next describe how to train these kind of models under differing types of supervision.

2.1 Reconstruction-based $\frac{1}{2}$TR

Linear unsupervised ranking methods trained based on a reconstruction objective are very popular approaches, including LSI, pLSA and LDA. The task is to learn a low dimensional "latent semantic" space where the semantics (e.g. topics) captured are used to measure similarity for ranking. Such methods have been shown to improve over using the original feature space for ranking, despite the task being unsupervised. HTR gives us the ability to define a nonlinear version of LSI that is highly scalable to train.

To do this we minimize the following (nonlinear) reconstruction objective:

$$\gamma \sum_i ||V v_i - \phi(y_i)||^2 + \sum_{k \neq l} \max(0, 1 - v_k^\top W \phi(y_k) + v_k^\top W \phi(y_l)) \quad (1)$$

with respect to V, W and v, where $V \in \mathbb{R}^{d \times n}$ and $W \in \mathbb{R}^{n \times d}$. In this unsupervised task, one is given objects y_i at training time, but only has access to queries x at test time. Hence training is done from the y_i objects only, which are used to model both queries and documents (implicitly, this is what LSI is doing as well). The first term of our objective takes the *transductive* point-wise representation v_i and tries to reconstruct its corresponding object y_i from it using a linear mapping V. The second term trains the inductive part of the model: for any k, it ensures that v_k is closer to $W\phi(y_k)$, the corresponding functional projection, than to any other functional projection $W\phi(y_l)$, $l \neq k$. At test time, ranking with respect to a new query x is performed according to its functional projection

$$f(x, y_i) = v_i^\top W \phi(x). \quad (2)$$

The second term of this objective measures the margin ranking loss that we are actually interested in. In this perspective, the first term can be seen as a regularizer which avoids trivial solutions: one can remark that without this first term, setting $v_i = W\phi(y_i)$ will cancel the margin ranking loss for an appropriate choice of $\|W\|$. Thus, the first term of (1) which forces reconstruction of the object from the transductive representation can be seen as a regularizer that prevents this overfitting from occurring. Note also that the reconstruction error from the first term should be less than a linear reconstruction, e.g. LSI, in the absence of features shared between the objects to be reconstructed. However, an algorithm utilizing the first term alone would be a *transductive* method with no direct out-of-sample extension.

We propose to train this model using stochastic gradient descent, (see, e.g. (Burges *et al.*, 2005)): iteratively, one picks a random triplet of objects with

indices i, k and l and makes a gradient step:

$$\begin{aligned} V &\leftarrow V + 2\lambda\gamma V\big(v_i - \phi(y_i)\big)v_i^\top \\ v_i &\leftarrow v_i + 2\lambda\gamma V^\top(Vv_i - \phi(y_i)) \\ W &\leftarrow W + \lambda v_k\big(\phi(y_k) - \phi(y_l)\big)^\top, \\ &\quad \text{if} \quad 1 - v_k^\top W\phi(y_k) + v_k^\top W\phi(y_l) > 0 \\ v_k &\leftarrow v_k + \lambda W\big(\phi(y_k) - \phi(y_l)\big), \\ &\quad \text{if} \quad 1 - v_k^\top W\phi(y_k) + v_k^\top W\phi(y_l) > 0. \end{aligned}$$

We choose the (fixed) learning rate λ which minimizes the training error. Convergence (or early stopping) is assessed with a validation set. Stochastic training is highly scalable and is easy to implement for our model. Moreover, it exhibits good properties toward avoiding poor local optima for non-convex optimization problems (Bottou, 2004). In our experiments, we initialized the matrices V, W and v randomly using a normal distribution with mean zero and standard deviation one.

2.2 Graph-based $\frac{1}{2}$TR

Graph-based unsupervised learning such as kernel PCA, Isomap, Locally Linear Embedding and Laplacian Eigenmaps are popular *transductive* unsupervised learning approaches, that are inherently nonlinear. For each object y_i an embedding vector v_i is learnt such that the given graph of object relationships is maintained in the new space. These methods hence suffer from the *out-of-sample problem* – they can only be applied to the set of objects $y_1, \ldots, y_m$ that they are trained with.

The HTR algorithm can be trained in a similar way to these algorithms, but yields a natural out-of-sample extension. Here, we consider the half transductive analog of Laplacian Eigenmaps[2]. Consider that for each pair of objects y_i and y_j we are given the value $A_{ij} \in \{0, 1\}$ which indicates whether these two objects are adjacent (i.e. A is their adjacency matrix). We then learn an embedding that tries to preserve this matrix by minimizing:

$$\gamma \sum_{i,j} L(v_i, v_j, A_{ij}) + \sum_{i,j} L(W\phi(y_i), v_j, A_{ij}) \qquad (3)$$

with respect to W and v, where

$$L(z, z', A_{ij}) = \begin{cases} ||z - z'||_1 & \text{if } A_{ij} = 1, \\ \max(0, 1 - ||z - z'||_1) & \text{if } A_{ij} = 0 \end{cases} \qquad (4)$$

Again, the first term of equation (3) is a classic *transductive* manifold-learning type loss function parameterized by v, and the second term trains the model to work for new *out-of-sample* data, parameterized by W. At test time one employs equation (2).

We again use stochastic training. Given a random pair with indices i, j we make an update:

$$\begin{aligned} v_i &\leftarrow v_i + \lambda\gamma g(v_i - v_j) \\ v_j &\leftarrow v_j - \lambda\gamma g(v_i - v_j) - \lambda g(W\phi(y_i) - v_j) \\ W &\leftarrow W + \lambda g(\phi(y_i) - v_j)^\top \end{aligned} \qquad (5)$$

where

$$g(x) = \begin{cases} -\text{sgn}(x) & \text{if } A_{ij} = 1, \\ \text{sgn}(x) & \text{if } A_{ij} = 0 \text{ and } ||x||_1 < 1, \\ 0 & \text{otherwise.} \end{cases}$$

and $\text{sgn}(x) = [\text{sgn}(x_1), \text{sgn}(x_2), ..., \text{sgn}(x_n)]^\top$ for a n-vector x. We iterate between picking a random positive and negative pair (i.e. with A_{ij} equal to 1 and 0).

2.3 Supervised $\frac{1}{2}$TR

We next consider the "Learning to Rank" (Joachims, 2002; Herbrich *et al.*, 2000) setting where one is given a set of objects $y_1, \ldots, y_m$, as before, as well as a set of known preference relations:

$$(x, y_p, y_n) \in \mathcal{R}$$

expressed as a set of tuples $\mathcal{R}$ (labeled data), where each tuple contains a query x, a relevant target y_p and an non-relevant (or lower ranked) target y_n. We would like to learn a function $f(\cdot)$ such that $f(x, y_p) > f(x, y_n)$, expressing that y_p should be ranked higher than y_n.

The standard solution to this kind of task is to use a functional approach typically using a linear model based on a few hand chosen input features, or nonlinear over a few hand-chosen features, and then training using an SVM or similar model, see e.g. (Joachims, 2002; Burges *et al.*, 2005).

We propose the following nonlinear half transductive method. We minimize the following:

$$\begin{aligned} \gamma \sum_{(x,y_p,y_n)\in\mathcal{R}} & R\Big(k(x, y_p), k(x, y_n)\Big) \\ + \sum_{(x,y_p,y_n)\in\mathcal{R}} & R\Big(v_p^\top W\phi(x), v_n^\top W\phi(x)\Big) \end{aligned} \qquad (6)$$

where

$$R(z, z') = \max(0, 1 - z + z'),$$

and

$$k(x, y) = \phi(y)^\top W^\top W\phi(x)$$

yielding a final ranking model (2) as before.

[2]Our approach is not identical to Laplacian Eigenmaps, e.g. here we chose the 1-norm for the matching function due to the simplicity of the gradient updates.

Hence, once again we have an objective with two terms. The second term ensures that relevant objects, using the nonlinear *transductive* embedding, are highly ranked given their *functionally* embedded queries. The first term is a regularizer that embeds the objects *functionally* as well. This controls the capacity of the model as when γ is increased the model becomes more linear.

The gradient updates for this setting, given a random triple with (x, y_p, y_n), are as follows:

$$
\begin{aligned}
W &\leftarrow W + \lambda\gamma W\phi(x)\big(\phi(y_p) - \phi(y_n)\big)^\top, \\
&\quad \text{if} \quad 1 - k(x, y_p) + k(x, y_n) > 0 \\
W &\leftarrow W + \lambda\big(v_p - v_n\big)\phi(x)^\top, \\
&\quad \text{if} \quad 1 - v_p^\top W\phi(x) + v_n^\top W\phi(x) > 0 \\
v_n &\leftarrow v_n - \lambda W\phi(x), \\
&\quad \text{if} \quad 1 - v_p^\top W\phi(x) + v_n^\top W\phi(x) > 0. \\
v_p &\leftarrow v_p + \lambda W\phi(x), \\
&\quad \text{if} \quad 1 - v_p^\top W\phi(x) + v_n^\top W\phi(x) > 0.
\end{aligned}
$$

Like for the other settings, these updates ensure a good scalability. Indeed, the training algorithm only accesses the W matrix and two rows v_p and v_n of the v matrix. This means that the v matrix could be stored over distributed storage and scale to very large transductive sets. In the next section, we actually present an example where the transductive set contains more than a million items.

3 Experiments

We now present experiments in the three setups described above: (i) an unsupervised reconstruction setup, only objects are available for training (Section 3.1), (ii) an unsupervised graph setup, objects and proximity information between objects are available for training (Section 3.2); and (iii) a supervised setup, objects and training queries along with corresponding relevance information are available for training (Section 3.3).

3.1 Reconstruction-based $\frac{1}{2}$TR

In order to evaluate unsupervised learning, we choose a labeled dataset. The Reuters Corpus Volume II is an archive of 804,414 newswire stories that have been manually categorized into 103 topics. The corpus covers four major groups: corporate/industrial, economics government/social, and markets. The topic classes form a tree which is typically of depth 3. Following (Salakhutdinov and Hinton, 2007), we define the relevance of one document to another to be the fraction of the topic labels that agree on the two paths from the root to the two documents. Hence, at test time, the model takes a novel query document and ranks the training items such that the items sharing the larger number of topics with the query should appear on top.

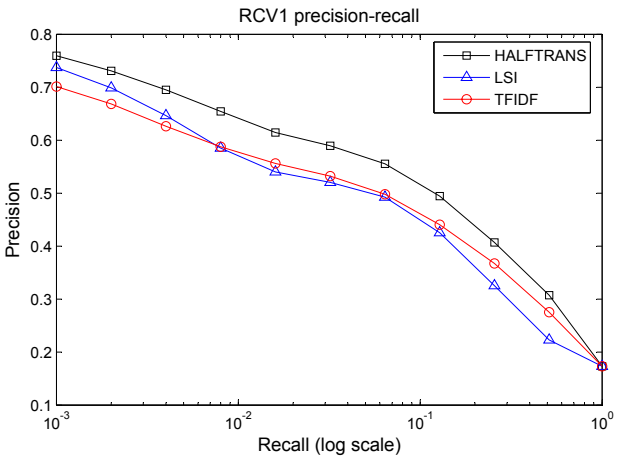


Figure 1: Half Transductive LSI (HALFTRANS) models documents with a nonlinear semantic representation and outperforms standard (linear) LSI and TFIDF schemes on the RCV2 benchmark.

The data was randomly split into 402,207 training and 402,207 test articles. The training set was further randomly split into 302,207 training and 100,000 validation documents. The data was preprocessed so that common stopwords were removed, and we considered the top 30,000 most frequent words[3].

We compare tf-idf with cosine similarity (TFIDF), LSI[4] and half transductive (nonlinear) LSI as defined in equation (1). Hyperparameters such as the embedding dimension are chosen using the validation set. The results are given in Figure 1. The results show that half transductive LSI (embedding dimension $n = 100$) outperforms the baselines, including standard LSI (also with $n = 100$). The modeling of documents with a nonlinear semantic representation in half transductive LSI might explain this outcome.

3.2 Graph-based $\frac{1}{2}$TR

To evaluate our graph-based model, we rely on the USPS dataset of hand-written digits[5] which consists

[3]Note that (Salakhutdinov and Hinton, 2007) used only 2,000 words, probably motivated by computational issues, which yields rather unrealistic, poor results.

[4]We use the SVDLIBC software `http://tedlab.mit.edu/~dr/svdlibc/` and the cosine similarity in the latent concept space.

[5]`http://www.cs.toronto.edu/~roweis/data.html`

of 16x16 pixel digits (10 classes) with 7,329 examples for training and 1,969 examples for testing which has been used in for embedding algorithms before (Saul and Roweis, 2003). Our setup constructs an embedding from the training set, and the test set is used for evaluating the quality of the learnt embedding. Given an embedded test point, we measure whether the nearest embedded train point shares the same label. Our evaluation hence measures the performance of a one nearest neighbor classifier over the learned space.

We compare Laplacian Eigenmaps (Belkin and Niyogi, 2003) and Laplacian Eigenmaps with out-of-sample extension (Bengio *et al.*, 2003) to our half transductive approach. As standard Laplacian Eigenmaps has no out-of-sample extension, we learn a *transductive* embedding on all 9298 examples (train and test) at once, and report error rates on the test set in this setting. This gives us an idea of what kind of error rate we would like our out-of-sample extended methods to achieve. For the other methods, including ours, we only train on the training set. We report results for an embedding dimension of 100, although other choices $(10, 50)$ yield similar conclusions. For our half transductive approach (3), we report results for two feature choices, i.e. linear features $\phi(x) = x$ and RBF features $\phi(x) = (K(x, y_1), \ldots, K(x, y_m))$ where $K(x', y') = \exp(-\frac{||x'-y'||^2}{2\sigma^2})$ and $y_1, \ldots, y_m$ are the training examples.

Table 1 reports the test error. It shows that our half transductive method provides good out-of-sample performance – as good as Laplacian Eigenmaps trained in the unfair setup granting access to the test points at training time. Our algorithm needs no out-of-sample extension, yielding an elegant solution to this problem. Figure 2 depicts the embedding given by half transduction with embedding dimension $n = 2$ for training points (left) and test points (right), which can clearly be seen to agree well.

Table 1: Empirical results for graph-based unsupervised learning on USPS.

Algorithm	1-NN Loss
Laplacian Eigenmaps (train+test)	0.0513
Laplacian Eigenmaps + O.S.E	0.0510
1/2-Transductive LE (Linear)	0.0673
1/2-Transductive LE (RBF)	0.0508

3.3 Supervised $\frac{1}{2}$TR

For our supervised learning to rank experiments, we do not rely on benchmark databases like LETOR (Liu *et al.*, 2007) since embedding algorithms cannot be evaluated: LETOR only provides features describing the match of document/query pairs and does not provide separate features describing each document and each query. TREC on the other hand offers a much smaller dataset (500 queries). We propose an approach which learns rich document/query relationships from bag-of-word features. This is a challenging learning problem which requires many query / document pairs. Unfortunately, large datasets are available only within search companies. To the best of our knowledge, Wikipedia is the closest publicly available resource that allows reporting reproducible results. Therefore, we employ Wikipedia and use its link structure to build a large scale ranking task.

We follow the setup of (Bai *et al.*, 2009): the dataset consists of 1,828,645 English Wikipedia documents and 24,667,286 links[6] which is randomly split into two portions, 70% for training and validation, 30% for testing. The link structure is used to provide relevance labels considering the following task: given a query document x, rank the other documents such that if x links to y then y should be highly ranked. Of course, links between train and test documents are considered unavailable at training time.

In these experiments, we compare HTR to several alternatives: (i) a margin ranking perceptron – similar to a ranking SVM – with the following feature representation:

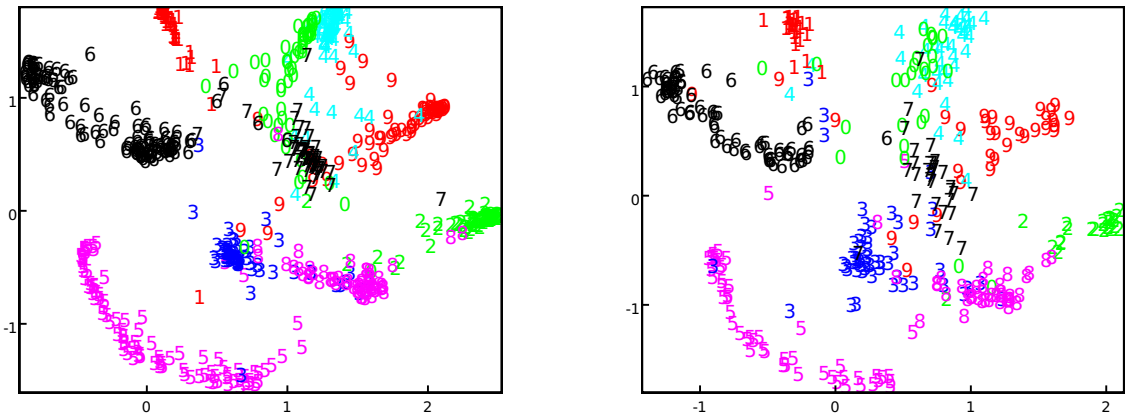
$$\Phi_{((i-1)\mathcal{D}+j)}(x, y) = (xy^\top)_{ij} \tag{7}$$

where $\Phi_s(\cdot)$ is the s^{th} dimension in our feature space, $\mathcal{D}$ is the dictionary size, and x, y are bag-of-words vectors, (ii) tf-idf with cosine similarity (TFIDF), (iii) Query Expansion (Zighelnic and Kurland, 2008), (iv) LSI, (v) a margin ranking perceptron relying on the Hash Kernels with hash size h (Shi *et al.*, 2009), and (vi) Supervised Semantic Indexing (SSI) (Bai *et al.*, 2009).

For all methods, hyperparameters – such as the embedding dimension $n \in \{50, 100, 200, 500, 1000\}$, or $h \in \{1M, 3M, 6M\}$ – are chosen using a validation set. For the HTR algorithm we used linear functional features, $\phi(x) = x$. The Margin Perceptron model and SSI can be seen as the inductive alternatives to the proposed HTR algorithm. For each method, we measure the ranking loss (the percentage of non-relevant documents appearing above a relevant one), precision at 10 (P@10, the percentage of top 10 items which are relevant) and mean average precision[7] (MAP, the aver-

[6] Links to calendar years are not considered as they provide little information while being very frequent.

[7] For computational reasons, MAP and P@10 were measured by averaging over a fixed set of 100 test queries for which all relevant documents and 10,000 non-relevant documents were ranked.



(a) Half transductive training set embedding

(b) Half transductive test set embedding

Figure 2: Half transductive embedding of digits from USPS. The left plot shows the transductive embedding learnt on the training data, and the right plot shows the functional embedding of the test data (which has also been learnt solely from the training data). A random subsample of 1/10 of the training and 1/5 of the testing data is shown.

Table 2: Empirical results for document-document ranking on Wikipedia (limited dictionary size of $\mathcal{D}$ =30,000 words).

Algorithm	Rank-Loss	MAP	P@10
TFIDF	1.62%	0.329±0.010	0.163±0.006
Query Expansion	1.62%	0.330±0.010	0.163±0.006
LSI	1.28%	0.346±0.011	0.170±0.007
Margin Ranking Perceptron (Bai *et al.*, 2009)	0.41%	0.477±0.011	0.212±0.007
SSI (Bai *et al.*, 2009)	0.301%	0.517±0.011	0.229±0.007
1/2-Transductive Ranking	**0.202%**	**0.557±0.012**	**0.241±0.007**

aged of the precision measured at each position where a relevant document appears).

We report results in two settings, (i) where we we used only the top 30,000 most frequent words in order to compare the margin ranking perceptron which would otherwise not fit in memory; and (ii) where we used all 2.5 million words in Wikipedia. The results are given in Tables 2 and 3. HTR outperforms all other ranking methods. TFIDF, Query Expansion and LSI are not trained from the supervised signal, and perform worst. The margin ranking perceptron, SSI and Hash Kernels are trained from the supervised signal (preference relations) and perform better, but still can only model linear relationships via their linear embedding. The nonlinear embedding provided by HTR captures nonlinear features of documents, resulting in superior retrieval performance. Regarding hyperparameter validation, we noticed that our approach is not very sensitive to the embedding dimension, i.e. 200 yields the best performance but the other experimented values gave similar results. The regularization parameter γ in (6) is however important, e.g. if it is set to $\gamma = 0$ a test ranking loss of 0.38% is obtained in the limited dictionary case.

4 Conclusions

In this work we studied the task of ranking a known (fixed) set of items with respect to a previously unseen query. Although this is a common setting in Information Retrieval, to our knowledge, it has not been defined and studied as the *Half Transductive Ranking* problem before. This work proposes several natural algorithms within this framework. In contrast to semi-supervised learning, our framework does not use auxiliary unlabeled data to complement a supervised task, but proposes to rely on non-linear embedding through transduction as a key point. For the ranking items available during training, *transductive* representations learning a parameter vector for each item allow the generation of highly nonlinear embeddings that focus on the relationships between the items rather than focusing on a specific choice of features. For unknown

Table 3: Empirical results for document-document ranking on Wikipedia (unlimited dictionary size, all $\mathcal{D} = 2.5M$ words).

Algorithm	Rank Loss	MAP	P@10
TFIDF	0.842%	0.432±0.012	0.193±0.007
Query Expansion	0.842%	0.432±0.012	0.193±0.007
LSI	0.721%	0.433±0.012	0.193±0.007
SSI	0.158%	0.547±0.012	0.239±0.008
Hash Kernels	1.37%	0.335±0.01	0.164±0.007
1/2-Transductive Ranking	**0.106%**	**0.613±0.012**	**0.256±0.008**

(new) queries, *functional* representations allow their embedding with good generalization properties.

The algorithms we propose combine the advantages of *transductive* and *inductive* techniques, resulting in strong performance in various setups, spanning both unsupervised and supervised tasks.

References

Bai, B., Weston, J., Collobert, R., and Grangier, D. (2009). Supervised Semantic Indexing. In *ECIR*, pages 761–765.

Balasubramanian, M., Schwartz, E., Tenenbaum, J., de Silva, V., and Langford, J. (2002). The Isomap algorithm and topological stability. *Science*, **295**(5552), 7–7.

Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, **15**(6), 1373–1396.

Bengio, Y., Paiement, J., Vincent, P., Delalleau, O., Le Roux, N., and Ouimet, M. (2003). Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In *NIPS*.

Blei, D. M. and McAuliffe, J. D. (2007). Supervised topic models. In *NIPS*, pages 121–128.

Bottou, L. (2004). Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to Rank Using Gradient Descent. In *ICML*, pages 89–96.

Cao, Z., Qin, T., Liu, T., Tsai, M., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136.

Collins, M. and Duffy, N. (2001). New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 263–270.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *JASIS*, **41**(6), 391–407.

Duh, K. and Kirchhoff, K. (2008). Learning to rank with partially-labeled data. In *SIGIR*, pages 251–258.

He, X. and Niyogi, P. (2003). Locality preserving projections. In *NIPS*.

Herbrich, R., Graepel, T., and Obermayer, K. (2000). *Advances in Large Margin Classifiers*, chapter Large margin rank boundaries for ordinal regression.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *SIGKDD*, pages 133–142.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer, New York.

Liu, T., Xu, J., Qin, T., Xiong, W., and Li, H. (2007). Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank*.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**(5500), 2323–2326.

Salakhutdinov, R. and Hinton, G. (2007). Semantic Hashing. In *SIGIR Workshop on Information Retrieval and Applications of Graphical Models*.

Saul, L. and Roweis, S. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, **4**, 119–155.

Scholkopf, B., Smola, A. J., and Müller, K. R. (1999). Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352.

Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A., Strehl, A., and Vishwanathan, V. (2009). Hash Kernels. In *AISTATS*.

Trosset, M. W. and Priebe, C. E. (2008). The out-of-sample problem for classical multidimensional scaling. *Comput. Stat. Data Anal.*, **52**(10), 4635–4642.

Vapnik, V. N. (1998). *Statistical Learning Theory*.

Weinberger, K. and Saul, L. (2008). Fast solvers and efficient implementations for distance metric learning. In *ICML*, pages 1160–1167.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *SIGIR*, pages 271–278.

Zighelnic, L. and Kurland, O. (2008). Query-drift prevention for robust query expansion. In *SIGIR*, pages 825–826.