

---

# Learning to Rank with Partially Labeled Training Data

*Tuong Vinh Truong, Massih-Rezah Amini, Patrick Gallinari*

*University of Pierre and Marie Curie, 8 rue Capitaine Scott 75015 Paris, FRANCE*

Many real life applications involve the ranking of objects instead of their classification. For example, in Document Retrieval the goal is to rank documents from a collection based on their relevancy to a user's query. Recently the supervised learning of ranking functions has attracted considerable attention from the Machine Learning community and most computational models proposed for ranking rely on this paradigm. Labeling large amounts of data may require expensive human resources, especially for ranking problems, and they are unrealistic in most applications. In the other hand, the semi-supervised learning paradigm which considers the possibility of learning from both the labeled and unlabeled examples has attracted the interest of the ML community in the field of classification since 1998.

In this paper, we propose a semi-supervised learning algorithm for ranking. Existing semi supervised ranking algorithms are graph-based transductive techniques which from an observed training dataset, order a specific unlabeled data pool. Our motivation here is to develop a novel inductive approach which from a specific observed training data (labeled and unlabeled) produces a general ranking rule, which ranks unseen examples with high accuracy. Our algorithm is an iterative approach which combines a supervised and a graph-based method. Empirical results on a real-life dataset from the CACM<sup>1</sup> collection have shown the potential of this approach in the context of Document Retrieval.

Keywords: Semi-supervised Learning, Ranking, Area Under the ROC Curve.

## 1 INTRODUCTION

The growing availability of on-line resources requires the conception of generic approaches that are able to automatically find relevant entities with respect to a user's demand. Most of these applications involve the ranking of entities instead of their classification. A usual example is the task of Documents Retrieval (DR), where the goal is to rank documents from a collection based on their relevancy to a given query.

Recently there has been an increasing interest of the Machine Learning (ML) community for the task of ranking [7] in which the goal is to learn an ordering over objects. Progress has been made in formulating different forms of the ranking problem. For the simplicity of presentation we will focus in this paper on the bipartite ranking framework in which instances are either positive or negative and the aim is to learn a scoring function which ranks positive instances above negative ones. This setting encompasses several Information Retrieval applications such as the automatic text summarization and has attracted considerable attention in machine learning community in both theory [1] and practical studies [11][7].

Learning to rank in a supervised setting often requires that an expert examines a large amount of data and assigns labels to all instances in the training set. This labeling process is simply unfeasible in many cases. In the other hand, it has been shown in the classification framework that learning with both labeled and unlabeled data may lead to a more efficient decision rule than learning with labeled examples alone.

In this paper we propose a semi-supervised learning algorithm for ranking. Up to our knowledge, existing semi supervised ranking algorithms are graph-based transductive approaches [6][14][9] which order examples from a given unlabeled dataset. Our motivation in the following is to develop a novel inductive approach which produces from specific observed training data (labeled and unlabeled), a general ranking rule that ranks unseen examples with high accuracy. We have led experiments on the CACM<sup>1</sup> collection gathering titles and abstracts from the journal Communications of the Association for Computer Machinery. Experimental results show encouraging evidence that unlabeled data may be useful for the task of ranking.

In the remaining of the paper, we first present the supervised bipartite framework and a ranking algorithm in section 2. In section 3 we outline our semi-supervised ranking mode and we present the results of our evaluation in section 4. Finally, in section 5 we discuss the outcome of this study.

## 2 THE BIPARTITE RANKING PROBLEM

### 2.1 Supervised framework

In the bipartite ranking framework we consider here, the training set  $S$  is constituted of a set of

---

<sup>1</sup> [http://www.dcs.gla.ac.uk/idom/ir\\_resources/test\\_collections/cacm/](http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/cacm/)

positive instances and a set of negative instances respectively denoted by  $S_1$  and  $S_{-1}$ . The goal is to learn from these instances a real-valued function  $h$  which gives higher scores to positive instances than to negative ones<sup>2</sup>.

The learning of  $h$  is formalized in [7] and it consists in minimizing the ranking loss of  $h$  which corresponds to the probability that a negative instance is ranked higher than a positive one.

$$L_r(S, h) = \frac{1}{|S_1| \times |S_{-1}|} \sum_{x \in S_1} \sum_{x' \in S_{-1}} \mathbb{I}[h(x) < h(x')] \quad (1)$$

Where  $\mathbb{I}[pr]$  is defined to be 1 if the predicate  $pr$  holds and 0 otherwise. The optimization of the ranking loss (1) is tightly related to the optimization of the Area Under the ROC Curve, may consider [1] for further information. More generally, we can optimize the following cost:

$$L_g(S, h) = \frac{1}{|S_1| \times |S_{-1}|} \sum_{x \in S_1} \sum_{x' \in S_{-1}} C_g(x, x', h) \quad (2)$$

Where  $C_g(x, x', h)$  is a function, which determines the penalty that  $h$  ranks  $x'$  higher than  $x$ . It can be equal to  $\mathbb{I}[h(x) < h(x')]$  or any other function approximating this term. Since our method is based on a supervised method, we present in the next section the logistic model for AUC which optimizes a criterion like (2).

## 2.2 Logistic model for AUC

The AUC logistic model [2] is an adaptation of the logistic model proposed for classification. The former produces a linear score function  $h(x) = \sum_{i=1}^d \beta_i x_i$ . In fact the bipartite ranking task can be seen as a classification of pairs constituted by a positive example and a negative example [2][4]. In this case a pair is correctly classified if and only if the score of the positive instance is higher than the score of the negative one. Since the instance pairs represented by  $(x, x')$  can be represented by the difference of their representative vectors  $(x - x')$ , the distributional assumption in the logistic model can be reformulated by:

$$P(\text{positive} | (x, x')) = \frac{1}{1 + e^{-2 \sum_{i=1}^d \beta_i (x_i - x'_i)}}$$

Then the parameters  $\beta = (\beta_1, \dots, \beta_d)$  are learned by maximizing the corresponding binomial likelihood (3):

$$L_{bin}(S, h) = \frac{1}{|S_1| \times |S_{-1}|} \sum_{x \in S_1} \sum_{x' \in S_{-1}} \log \left( 1 + e^{-\sum_{i=1}^d \beta_i (x_i - x'_i)} \right) \quad (3)$$

Instead of optimizing (3), we can optimize the exponential loss defined by (4):

$$L_{exp}(S, h) = \frac{1}{|S_1| \times |S_{-1}|} \sum_{x \in S_1} \sum_{x' \in S_{-1}} e^{-\sum_{i=1}^d \beta_i (x_i - x'_i)} \quad (4)$$

Indeed the binomial log-likelihood  $-E[\log(1 + e^{-2yh(x)})]$  is usually used as an optimization criterion in classification. [8] has shown that optimizing  $L_{exp}$  leads to the same parameters than optimizing  $L_{bin}$ . Since the exponential function  $e^{-(h(x) - h(x'))}$  is an upper bound of the indicator function  $\mathbb{I}[h(x) < h(x')]$ , the general ranking cost (2) encompasses the exponential cost (4). Moreover this latter function is convex and can be minimized by standard optimization functions. An interesting property of exponential loss is the possibility to compute it in time linear of examples by rewriting (4) as follow:

$$L_{exp}(S, h) = \frac{1}{|S_1| \times |S_{-1}|} \sum_{x \in S_1} e^{-\sum_{i=1}^d \beta_i x_i} \sum_{x' \in S_{-1}} e^{\sum_{i=1}^d \beta_i x'_i}$$

<sup>2</sup> We assume that an instance  $x$  is preferred over an instance  $x'$  iff  $h$  gives to  $x$  a higher score than to  $x'$  i.e.  $h(x) > h(x')$ .

### 3 SEMI-SUPERVISED RANKING TASK

#### 3.1 Notation

In the semi supervised ranking setting, we assume, that the training set contains a set of instances with label which reflects a preference or relevance judgment and a set of unlabeled instances. In the bipartite ranking case, an example belongs only to one class (positive or negative). The learner is also given a set  $S_L = \{x_i, y_i\}_{i=1}^L$  of labeled instances and a set  $S_U = \{x_i\}_{i=L+1}^{L+U}$  of unlabeled instances, where  $x_i \in X \subset \mathfrak{R}$  are  $d$ -dimensional vectors and  $y_i \in \{-1, 1\}$  are labels. We denote the set of positive instances included in  $S_L$  by  $S_+$  and the set of negative instances included in  $S_L$  by  $S_-$ .

#### 3.2 Background

They are classically two frameworks in the semi-supervised ranking learning: inductive and transductive. In the latter, the aim is to find the ordering of a fixed unlabeled set whereas in the first category the aim is to find a general ranking rule using both the labeled and unlabeled examples. Up to our knowledge existing semi-supervised methods for the ranking task are graph-based methods which are inherently transductive.

For example, [14] develop a transductive method based on manifold learning. Such methods are known to be efficient in classification. They assume that data are close to a manifold structure, which can be revealed by a large amount of unlabeled data. In [14], the special case of learning with only positive and unlabeled data has been studied. The proposed algorithm first builds a graph approximating the manifold structure of the training data. The vertexes of the graph are labeled and unlabeled data in the training set and its edges are weights reflecting a pairwise distance or similarity between examples at each end of the edge. The algorithm begins by assigning a score of 1 to positive examples and a score of 0 to the remaining. Scores are then propagated to their neighbours through the graph until that a global stable state is reached. At the end the scores induce a total order on the unlabeled data.

Formally, let  $d : X \times X \rightarrow \mathfrak{R}$  denote a metric on  $X$  and  $f$  be a ranking function which assigns to each point  $x_i$  a ranking score  $f_i$ . We can view  $f$  as a vector  $f = [f_1, \dots, f_n]^T$  where  $n$  is the total number of labeled and unlabeled examples in the training set. We also define a vector  $y = [y_1, \dots, y_n]^T$ , in which  $y_i = 1$  if  $x_i$  is a positive example and  $y_i = 0$  otherwise. The algorithm in [14] is depicted in the following.

1. Sort the pairwise distances among points in ascending order. Repeat connecting two points with an edge according to their respective similarity until a connection graph is obtained.
2. Form the affinity matrix  $W$  defined by  $W_{ij} = \exp(-d(x_i, x_j)/2\sigma^2)$  if there is not an edge between  $x_i$  and  $x_j$  and let  $W_{ii} = 0$  otherwise.
3. Symmetrically normalize  $W$  by  $S = D^{-1/2}WD^{1/2}$  in which  $D$  is the diagonal matrix with  $(i, i)$ -element equal to the sum of the  $i$ th row of  $W$ .
4. Iterate  $f(t+1) = f(t) + (1-\alpha)y$  until convergence, where  $\alpha$  is a parameter in  $[0, 1)$ .
5. Let  $f_i^*$  denote the limit of the sequence. Rank each point  $x_i$  according to its ranking scores  $f_i^*$  (largest rank first).

The parameter  $\alpha$  specifies the relative contributions to the ranking scores from neighbours and the initial ranking scores. As in [9], the graph is constructed by the  $K$  nearest neighbours to ensure enough connections for each point while preserving the sparse property of the weighted graph. Since we have experimented on CACM collection where they are very few pertinent documents for each query, we do not assign any score to negative labeled data. We can use alternative methods proposed in [9] to take into account negative labeled examples for balanced collections.

#### 3.3 A semi supervised model for AUC

The supervised method such as the logistic model for AUC has the advantages of optimizing directly a ranking measure such as AUC and of being able to rank unseen examples, whereas the graph-based method exploits efficiently the data structure to rank examples according to their similarity to positive examples. In this paper we intend to combine both methods in order to exploit the advantage of each of the both techniques. Our approach consists in fact to search a compromise between optimizing the exponential cost defined by (4) and while respecting the graph based order on a top ranked unlabeled subset.

Our algorithm is presented below. It first learns a scoring function by minimizing the exponential cost

on a labeled training set. Three steps are then iterated until that the algorithm converges or a maximum number of iteration is reached. At step 1, unlabeled instances are ranked according to the current scoring function and an unlabeled subset is built with the highest scored instances. We denote by  $n$  the size of the unlabeled subset. At step 2, we compute a cost, which reflects the dissimilarity on this subset between the ordering found at the step 1 and the one learnt by the graph-based method. For example, we can use an exponential loss function like  $\sum_{j=1}^n e^{(h(x_{\varphi(j+1)})-h(x_{\varphi(j)}))}$  (with  $\varphi(j)$  a function which returns the index of the example ranked at the position  $j$  by the graph-based method) to preserve the convexity of the criterion. Nevertheless any dissimilarity measure between two orders could be used. At the next step, we optimize the exponential criterion to which we add the cost on the unlabeled subset.

**Algorithm: semi supervised logistic model for AUC**

**Input:**  $S, n$

**Initialisation:** Learn a score function by optimizing a supervised cost  $L_{\text{exp}}$

**Repeat until convergence or until the maximum iteration is reached:**

1. Generate a subset  $S_{UNL}^{(i)}$  from unlabeled data with the  $n$  highest scored instances.

2. Produce an index function  $\varphi^{(i)}$  with the manifold based method.

3. Optimize  $L_{\text{new}}(S, h) = \sum_{x \in S_1} \sum_{x' \in S_{-1}} e^{-\sum_{i=1}^d \beta_i (x_i - x'_i)} + \lambda \sum_{j=1}^n e^{(h(x_{\varphi(j+1)})-h(x_{\varphi(j)}))}$ .

The fixed parameter  $\lambda \in [0,1]$  measures the contribution of the unlabeled examples to the criterion.

**End**

**Output:**  $h(x) = \sum_{i=1}^d \beta_i x_i$

## 4 EXPERIMENTS AND DISCUSSION

### 4.1 Data set

In this section, we have only compared our algorithm with the supervised algorithm [2] to validate our approach and to show that unlabeled data may be useful for the ranking task. We used AUC and average precision as evaluation measures which are usually used in the Information Retrieval community. We have led experiments on the CACM collection gathering titles and abstracts from the journal Communications of the Association for Computer Machinery.

For each query we have omitted pertinent documents, which do not content any word of query and a test set with 50% of the data was created randomly. We kept also only one positive example since the base seems easy to learn and we choose only queries where there are enough positive examples. We have fixed parameters  $\lambda$  at 1, the size of unlabeled subset  $n$  at 10 and the graph is built by 5-nearest-neighbor technique. The supervised method was applied on the labeled examples for each experience. The results on the CACM collection are summarized in Table 1. Numbers shown here are the AUC and the average precision on the test set averaged over five random train/test split. For each query, we present the results of both models.

### 4.2 Results and discussion

Empirical results on average precision show that the supervised method outperforms the semi supervised method. Indeed our algorithm does not optimize directly this criterion. Moreover even if AUC and average precision can be correlated [3], the AUC is more related to negative examples than to positive examples. They are indeed very few positive documents for each query. In fact we could use directly the average precision as supervised criterion [8] or use an  $L_p$  norm. In [12] the author shows that using such technique leads the algorithm to concentrate harder near the top of the ranked list. The criterion used in [12] is moreover close to the AUC.

For the AUC measure, we obtain a loss on the query 17. In this case the positive document contains very few words. We suppose also that this document is far from the remaining examples and could mislead the graph-based similarity. Nevertheless the results on the AUC show an improvement for the majority of queries and a substantial gain for the half of them. On average, the semi supervised algorithm outperforms

the supervised algorithm.

Table1. For each query identified by an integer, we present the results obtained by the model logistic (column Sup) and by our algorithm (column Semi) for AUC and average precision.

	7		10		11		14		17		25	
	Sup	Semi	Sup	Semi	Sup	Semi	Sup	Semi	Sup	Semi	Sup	Semi
AUC	81.4	<b>90.8</b>	80.2	<b>87.9</b>	<b>98</b>	97.9	90.8	<b>93</b>	<b>95.6</b>	90.3	83.2	<b>91.2</b>
Prec	<b>21.6</b>	18.9	20.1	<b>23.8</b>	<b>31.5</b>	25	<b>32.6</b>	28.8	<b>35</b>	27.4	<b>29.4</b>	12.3

  

	27		29		42		43		58		60	
	Sup	Semi	Sup	Semi	Sup	Semi	Sup	Semi	Sup	Semi	Sup	Semi
AUC	89.7	<b>95.8</b>	91.4	<b>91.7</b>	72.1	<b>89.6</b>	<b>98.9</b>	98.4	86.2	<b>87.3</b>	<b>98.8</b>	97.7
Prec	42.9	<b>44.4</b>	11.3	<b>11.6</b>	31.6	<b>36.9</b>	<b>49</b>	38.9	20.2	<b>25.6</b>	<b>57.1</b>	43.5

## 5 CONCLUSION AND PERSPECTIVE

The aim contribution of this work is a method for ranking applications which uses both labeled and unlabeled data to infer a ranking function. Our method combines a supervised method and a graph-based technique. The empirical results show that unlabeled data can be used in order to improve the AUC. For future works it will be interesting to perform our method on others collections and to investigate other unlabeled data dependent cost functions. Moreover since we have only investigated the bipartite ranking we intend to investigate more general ranking cases in a future work.

### ACKNOWLEDGEMENT

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

### REFERENCES

- [1] Agarwal S., Roth D., 2005, Learnability of bipartite ranking functions, Proceedings of the 18th Annual Conference on Learning Theory.
- [2] Amini M.-R., Usunier N., Gallinari P. Automatic text summarization based on word-clusters and ranking algorithms. Proceedings of the 27th European Conference on Information Retrieval.
- [3] Caruana R., Niculescu-Mizil A., 2004, Data mining in metric space: an empirical analysis of supervised learning performance criteria, Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining.
- [4] Clemençon S., Lugosi G., Vayatis N., 2005, Ranking and empirical minimization of U-statistics, Conference on Learning Theory.
- [5] Cortes C., Mohri M., 2004, AUC optimization vs error rate minimization, Advances in Neural Information Processing System.
- [6] Chu W., Ghahramani Z., 2005, Extensions of gaussian processes for ranking: semi-supervised and active learning. Proceedings of the NIPS'05 workshop on Learning to Rank.
- [7] Freund Y., Iyer R., Shapire R.E., Singer Y., 2003, An efficient boosting algorithm for combining preferences, Journal of Machine Learning Research, 4:933-969.
- [8] Friedman J., Hastie T., Tibshirani R., 1998, Additive logistic regression: a statical view of boosting TR.
- [9] He J., Li M., Zhang H.-J., Tong H., Zhang C., 2004, Manifold-ranking based image retrieval, Proceedings of the 12<sup>th</sup> annual ACM international conference on multimedia.
- [10] Metzler D. A., Croft W. B., McCallum A., 2005, Direct maximisation of rank-based metrics for information retrieval, CIIR technical report.
- [11] Rudin C., Cortes C., Mohri M., Shapire R.E., 2005, Margin-based ranking meets boosting in the middle. Conference on Learning Theory.
- [12] Rudin C, 2006, Ranking with a P-Norm Push, Conference on Learning Theory .
- [13] Usunier N., Truong T.V., Amini M.-R., Gallinari P. Proceedings of the NIPS'05 workshop on Learning to Rank.
- [14] Zhou D., Weston J., Gretton A., Bousquet O., and Scholkopf B., Ranking on data manifolds. Cambridge, Mass., 2004. MIT Press