# Ranking Structured Documents: A Large Margin Based Approach for Patent Prior Art Search

**Yunsong Guo** and **Carla Gomes**
Department of Computer Science
Cornell University
{guoys, gomes}@cs.cornell.edu

## Abstract

We propose an approach for automatically ranking structured documents applied to patent prior art search. Our model, SVM Patent Ranking ($\text{SVM}_{PR}$) incorporates margin constraints that directly capture the specificities of patent citation ranking. Our approach combines patent domain knowledge features with meta-score features from several different general Information Retrieval methods. The training algorithm is an extension of the Pegasos algorithm with performance guarantees, effectively handling hundreds of thousands of patent-pair judgements in a high dimensional feature space. Experiments on a homogeneous essential wireless patent dataset show that $\text{SVM}_{PR}$ performs on average 30%-40% better than many other state-of-the-art general-purpose Information Retrieval methods in terms of the NDCG measure at different cut-off positions.

## 1 Introduction

Patents protect intellectual property rights by granting the invention's owner (or his/her assignee) the right to exclude others from making, using, selling, or importing the patented invention for a certain number of years, usually twenty years, in exchange for the public disclosure of the details of the invention. Like any property right, patents may be transacted (e.g., sold, licensed, mortgaged, assigned or transferred) and therefore they have economic value. For a patent to be granted, the patent application must meet the legal requirements related to patentability. In particular, in order for an invention to be patentable, it must be *novel*. Patent applications are subject to official examination performed by a patent examiner to judge its patentability, and in particular its novelty. In most patent systems *prior art* constitutes information that has been made available to the public in any form before the filing date that might be relevant to a patent's claims of originality. Previously patented inventions are to a great extent the most important form of prior art, given that they have already been through the patent application scrutiny.

The decision by an applicant to cite a prior patent is arguably "tricky" [Lampe, 2007]: on one hand, citing a patent can increase the chance of a patent being accepted by the United States Patent and Trademark Office (USPTO), if it shows a "spillover" from the previously granted patent; on the other hand, citing a prior patent may invalidate the new patent if it shows evidence of intellectual property infringement. As a result, some applicants choose to cite only remotely related patents, even if they are aware of other more similar patents. It is thus part of the responsibility of the USPTO examiner to add patent citations, beyond those provided by the applicant. This is important since more than 30% of all patent applications granted from 2001 to 2007 actually have zero applicant citations. However, this is also a tedious procedure often resulting in human error, considering the huge amount of patents to be possibly cited and the fact that different examiners with different expertise backgrounds are likely to cite different sets of patents for the same application. Furthermore, while applicants tend to cite patents that facilitate their invention, denoting to some extent *spillovers* from a patent invention, examiners tend to cite patents that block or limit the claims of later inventions, better reflecting the patent *scope* and therefore representing a better measure of the private value of a patent.

We focus on the problem of automatically ranking patent citations to previously granted patents: given a new patent application, we are interested in identifying an ordered list of previously granted patents that examiners and applicants would consider useful to include as prior art patent citations.

In recent years several margin based methods for document ranking have been proposed, e.g., [Herbrich *et al.*, 2000; Nallapati, 2004; Cao *et al.*, 2006; Joachims, 2002; Chapelle *et al.*, 2007; Yue *et al.*, 2007]. For example, in the approach by [Herbrich *et al.*, 2000; Nallapati, 2004] the ranking problem is transformed into a binary classification problem and solved by SVM. [Cao *et al.*, 2006] considers different loss functions for misclassification of the input instances, applied to the binary classification problem; [Joachims, 2002] uses click through data from search engines to construct pair-wise training examples. In addition, many other methods learn to optimize for a particular performance measure such as accuracy [Morik *et al.*, 1999], ROC area [Herschtal and Raskutti, 2004; Carterette and Petkova, 2006], NDCG [Burges *et al.*, 2005; 2006; Chapelle *et al.*, 2007] and MAP [Yue *et al.*, 2007].

Note that the patent prior art search problem differs from the traditional learning to rank task in at least two important ways. Firstly, in learning to rank we measure the relevance

of query-document pairs, where a query is usually a short sentence-like structure. In patent ranking, we are interested in patent-patent relevance, where a patent is a full document much longer in length than a normal query. We will also see in the experiment section that treating a document as a long query often results in inferior performance. Secondly, in the case of patent prior art search, we have two entities, namely the examiner and applicant, who decide what patents to cite for a new application. As we mentioned above, the differences in their citation behaviors are not only a matter of level of relevance, but also of strategy.

To address the various issues concerning patent prior art search, we propose a large-margin based method, SVM Patent Ranking (SVM$_{PR}$), with constraint set directly capturing the different importance between citations made by examiners and citations made by applicants. Our approach combines patent-based knowledge features with meta-score features, based on several different ad-hoc Information Retrieval methods. Our experiments on a real USPTO dataset show that SVM$_{PR}$ performs on average 30%-40% better than other state-of-the-art ad-hoc Information Retrieval methods on a wireless technology patent dataset, in terms of the NDCG measure. To the best of our knowledge, this is the first time a machine learning approach is used for automatically generating patent prior art citations, which is traditionally handled by human experts.

## 2 Numerical Properties of Patent Citations

From January 1, 2001 to the end of 2007, the USPTO has granted 1,263,232 patents. The number of granted patents (per year) is displayed in the left plot of Figure 1. We observe a steady and slow increase in the number of patents granted per year from 2001 ($\sim$180,300 patents) to 2003 ($\sim$187,132 patents), and then a steep decrease until the all-time low of $\sim$157,866 in 2005. Interestingly there is a sharp increase in the number of patents in 2006, achieving the highest number to date with $\sim$196,593 patents.

There are obvious differences between examiner and applicant citations. (See e.g., [Alcacer and Gittelman, 2006; Sampat, 2004].) The average number of patent citations added by the examiner and applicant is presented in the middle plot of Figure 1. The average number of patent citations added by the examiner is relatively stable, ranging from 5.49 to 6.79 , while the average number of applicant citations increases significantly from 8.69 in 2001 to 15.11 in 2007. In addition, the distribution of patent citations by the applicant is extremely uneven: $\sim$372,372 (29.5%) patents have no applicant citations, and $\sim$19,388 (1.5%) patents have more than 100 applicant citations, while the number of patents with more than 100 examiner citations is only 40. The rightmost plot of Figure 1 compares the number of patents with no more than 20 citations made by the examiner, versus that made by the applicant. As clearly displayed, a large portion of the patents have 0 applicant citation, and the mode number of examiner citation is 3, with $\sim$134,323 patents.

## 3 SVM$_{PR}$ for Patent Ranking

### 3.1 Some Notations

We denote the set of patents whose citations are of interest by $\mu$; and the set of patents that patents in $\mu$ cite, by $\nu$.

In addition, $\forall p_i \in \mu$ and $\forall p_j \in \nu$, we define the citation matrix $\mathbf{C}$ with entries $C_{ij}$ to be

$$C_{ij} = \begin{cases} 2 & \text{if patent } p_i \text{ cites patent } p_j \text{ by examiner} \\ 1 & \text{if patent } p_i \text{ cites patent } p_j \text{ by applicant} \\ 0 & \text{otherwise} \end{cases}$$

The numerical values seem to be arbitrary here but the reason for such choices will be clear in the next section. Moreover, we denote $\Phi(p_i, p_j)$ as the feature map vector constructed from patents $p_i$ and $p_j$. Details of $\Phi$ are presented in section 3.3.

### 3.2 SVM$_{PR}$ Formulation

Our model is called SVM Patent Ranking (SVM$_{PR}$). It is formulated as the following large margin quadratic optimization problem with the constraint set directly capturing the specificities of patent ranking:

OPTIMIZATION PROBLEM I

$$\min_{\mathbf{w},\xi\geq 0} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{|\mu||\nu|^2}\sum_{p_i\in\mu}\sum_{p_j\in\nu}\sum_{p_k\in\nu}\xi_{ijk} \qquad (1)$$

subject to:

$\forall p_i \in \mu, \forall p_j \in \nu, \forall p_k \in \nu,$

$$\mathbf{w}^T\Phi(p_i, p_j) - \mathbf{w}^T\Phi(p_i, p_k) \geq \Delta(p_i, p_j, p_k) - \xi_{ijk}$$

where

$$\Delta(p_i, p_j, p_k) = \begin{cases} C_{ij} - C_{ik}, & \text{if } C_{ij} - C_{ik} > 0 \\ -\infty, & \text{otherwise} \end{cases}$$

Here $\Delta(p_i, p_j, p_k)$ is the loss function when $C_{ij} - C_{ik} > 0$ and the learned $\mathbf{w}$ parameter scores less with $\Phi(p_i, p_j)$ than $\Phi(p_i, p_k)$. The magnitude of $C_{ij} - C_{ik}$ represents the severity of the loss: a mistake of ranking an uncited document higher than a document cited by an examiner is more serious than ranking the same uncited document higher than some applicant cited document. If $C_{ij} - C_{ik} \leq 0$, no loss will be incurred, and we set $\Delta(p_i, p_j, p_k)$ to $-\infty$ to invalidate the constraint.

The objective function (1) is the usual minimization that trades off between $\|\mathbf{w}\|^2$, the complexity of the model, and $\sum \xi_{ijk}$ , the upper bound of the total training loss defined by $\Delta$. $\lambda$ is the tradeoff constant.

The above formulation considers all tuples (i,j,k) to ensure that examiner cited patents are ranked higher than applicant cited patents, which are again ranked higher than uncited patents, by an absolute margin "1" using the linear discriminant score $\mathbf{w}^T\Phi$. The drawback is that it contains $\mathcal{O}(|\mu||\nu|^2)$ constraints, which poses a challenge for any reasonable training process. In fact, we do not need to include all tuples as constraints, but rather only consider the extreme values of the ranking scores of the three different citation groups (by examiner, by applicant and not cited). Therefore, we can construct the following alternative formulation:

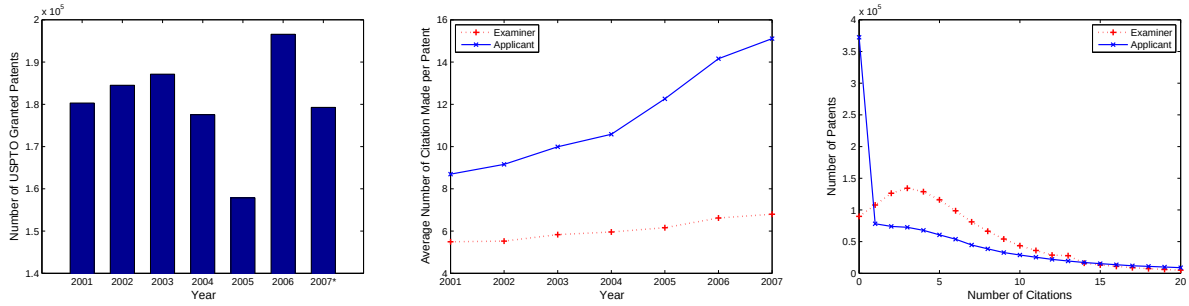


Figure 1: USPTO Patent Data. Left panel: patents per year; Middle panel: yearly citations by examiner and applicant; Right panel: frequency of examiner citations and applicant citations.

OPTIMIZATION PROBLEM II (SVM$_{PR}$)

$$\min_{\mathbf{w},\xi\geq 0} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{2|\mu|}\sum_{p_i\in\mu}(\xi_i^1+\xi_i^2) \qquad (2)$$

subject to: $\forall p_i \in \mu$

$$\begin{aligned}
\min_{p_j\in\{p:p\in\nu\wedge C_{ip}=2\}} \mathbf{w}^T\Phi(p_i,p_j) \quad &- \\
\max_{p_k\in\{p:p\in\nu\wedge C_{ip}=1\}} \mathbf{w}^T\Phi(p_i,p_k) \quad &\geq \quad 1-\xi_i^1 \\
\\
\min_{p_j\in\{p:p\in\nu\wedge C_{ip}=1\}} \mathbf{w}^T\Phi(p_i,p_j) \quad &- \\
\max_{p_k\in\{p:p\in\nu\wedge C_{ip}=0\}} \mathbf{w}^T\Phi(p_i,p_k) \quad &\geq \quad 1-\xi_i^2
\end{aligned}$$

It is not difficult to see that the constraint set of Optimization Problem 1 implies the constraint set of Optimization Problem 2, and vice versa, since in the latter case we made the changes that the constraints only need to be satisfied at the extreme values. The advantage is obvious because this leads to a convex quadratic optimization problem with only $\mathcal{O}(|\mu|)$ constraints. Our training algorithm will be based on this formulation.

We note that if the distinction between examiner citations and applicant citations is ignored, i.e, $C_{ij}$=1 iff patent $p_i$ cites $p_j$, OPTIMIZATION PROBLEM I can be transformed into the ordinal regression formulation in [Herbrich *et al.*, 2000] in a straightforward way. The ordinal regression formulation would require $\mathcal{O}(|\mu|^2|\nu|^2)$ constraints, treating each $\Phi(p_i,p_j)$ as an individual example; or $\mathcal{O}(|\mu||\nu|^2)$ constraints if the examples $\Phi(p_i,p_j)$ are first grouped by $p_i$. In order to differentiate examiner and applicant citations we applied the SVM multi-rank ordinal regression in [Chu and Keerthi, 2005] to our transformed problem with $\mathcal{O}(|\mu||\nu|)$ examples. Unfortunately the ordinal regression method could not handle the scale of the problem, as it fails to complete the training process within 48 hours of runtime. In contrast, our approach, SVM$_{PR}$, exploits the important distinction between examiner and applicant citations, leading to the efficient formulation and training process.

After the optimization phase, we use the linear discriminant function $\mathbf{w}^T\Phi(p,q)$ to rank a new patent $p$ against any patent $q$ to be considered for the prior art citation. Our target ranking for $p$ is that all examiner-cited patents are ranked higher than all applicant-cited patents, and all cited patents are ranked higher than all patents not cited. We evaluate our learned patent ranking with the target ranking by NDCG.

It is worthwhile to note that our approach is also related to the margin based methods for label ranking [Shalev-Shwartz and Singer, 2006], in which the goal is to learn an ordered preference list of labels for each instance by learning local weight vectors for each class label. In contrast, our approach is to learn a global weight vector and differentiate class labels (cited by examiner, or applicant, or not cited) by the linear discriminant score, using the patent specific feature map vectors.

### 3.3 Feature Map Construction

One major difference between assessing patent similarity and query-document similarity, a traditional task in IR, is that patents are full documents, which are usually significantly longer than an average query. Of course we can treat one patent as a long query, but this will not result in good performance for our task as we will see in the experiment section.

A key step in our approach for the training and testing procedure is the construction of a feature map for pairs of patents $p_i$ and $p_j$. Intuitively, $\Phi(p_i,p_j)$ represents information we know about how similar $p_i$ and $p_j$ are. In our approach, $\Phi(p_i,p_j)$ is a column vector composed of two parts: the domain knowledge patent pair features and the meta score features.

#### Domain Knowledge Patent Pair Features

The domain knowledge features between patents $p_i$ and $p_j$ come from the study of patent structure without any citation information of either $p_i$ or $p_j$. Here applicants and inventors are used interchangeably. We have 12 binary features in $\Phi(p_i,p_j)$ as follows:

*1*. Both patents are in the same US patent class; *2*. Both patents have at least one common inventor (defined by same surname and first name); *3*. Both patents are from the same assignee company (e.g., Microsoft, HP etc.); *4*. Both patents are proposed by independent applicants (i.e, no assignee company); *5-7*. First inventors of the two patents come from the same city, state or country; *8*. $p_i$'s patent date is later than $p_j$'s patent date (so $p_i$ can possibly cite $p_j$); *9-12*. Both patents made 0 claims, 1-5 claims, 6-10 claims or more than 10 claims.

**Meta Score Features**

We also make use of ad-hoc IR methods to score the patent pairs. The scores are then changed into feature vector representation in $\Phi$.

We used the Lemur information retrieval package[1] to obtain the scores on patent pairs by six ad-hoc IR methods: TF-IDF, Okapai, Cosine Similarity and three variations of the KL-Divergence ranking method. Additional descriptions of these methods can be found in Section 4.3. Each method is essentially a scoring function $\psi(q,d) \in Q \times D \mapsto \mathbb{R}$ where $Q$ is a set of queries and $D$ is a set of documents whose relevance to the queries is ranked by the real-valued scores. We use the patents' titles and abstracts available from the USPTO patent database to obtain the meta score features.

We can view each sentence of a patent $p_i \in \mu$ as a query whose relevance with respect to a patent $p_j \in \nu$ is to be evaluated. Let $S$ be the set of all sentences from all patents in $\mu$. We associate a binary vector $t^s$ of length 50 with each $s \in S$ according to $\psi(s, p_j)$, following the representation in [Yue *et al.*, 2007]:

$$\forall i \in \{1,\ldots,50\},\ t_i^s = \begin{cases} 1, & \psi(s,p_j) \geq Pt(2(i-1)) \\ 0, & otherwise \end{cases}$$

where $Pt(x)$ is the $x^{th}$ percentile of $\{\psi(q,p_j) : q \in S\}$. The $t^s$ vector is a binary representation of how relevant sentence $s$ is with respect to $p_j$, compared to all other sentences in $S$. We repeat this procedure for all six ad-hoc IR methods and concatenate the results to obtain the vector $t^s$ of length 300.

Let $(s_1, s_2, \ldots, s_{m_i})$ be the $m_i$ sentences of $p_i$ sorted in non-increasing order according to $\psi_{TF-IDF}(s, p_j)$. The meta score feature vector between $p_i$ and $p_j$ is defined as

$$\Psi(p_i,p_j) = \sum_{l=1}^{m_i} \frac{t^{s_l}}{2^{(l-1)}} \quad (3)$$

In other words, Equation (3) is a weighted sum of $t^s$ from each sentence $s$, discounting sentences that are ranked as less relevant exponentially by $\psi_{TF-IDF}$. We also tried other alternative weighting schemes, such as harmonic discounting, but none of them performed empirically as well as the exponential weight discount.

Hence, the feature map $\Phi(p_i,p_j)$ for any $p_i \in \mu$ and $p_j \in \nu$ is the concatenation of the 12 knowledge domain features and $\Psi(p_i,p_j)$, with a total of 312 features.

### 3.4 The Training Algorithm

The training algorithm of SVM$_{PR}$ that optimizes (2) with respect to **w** is an extension of Pegasos [Shalev-Shwartz *et al.*, 2007]. Pegasos is an SVM training algorithm that alternates between a gradient descent step and a projection step. It operates solely in the primal space, and has proven error bounds. Our training algorithm is presented in Algorithm 1.

Among the four input parameters, $\mu$ and $\nu$ have the same meaning as in the previous sections; $T$ is the total number of iterations; $\lambda$ is the constant in the SVM$_{PR}$ formulation. In our experiments $T$ and $\lambda$ are fixed at 200 and 0.1. In general the performance is not sensitive to any reasonable $\lambda$ setting. Lines 6-9 calculate the extreme values of the discriminant function $\mathbf{w}^T\Phi(p_i,p_j)$ for patents $p_j$ grouped by their citation relation with $p_i$ in order $\mathcal{O}(|\nu|)$. The set $A$ is the set of violated constraints with respect to OPTIMIZATION PROBLEM II, or equivalently the most violated constraints of OPTIMIZATION PROBLEM I. Line 17 updates **w** using the violated constraints in $A$. This step is in order $\mathcal{O}(|\mu|)$ as $|A| \leq 2|\mu|$. Lines 18-20 ensure the 2-norm of **w** is not too big, which is a condition used in [Shalev-Shwartz *et al.*, 2007] to prove the existence of an optimal solution. Line 22 outputs the final **w** parameter with the best validation set performance. The performance measure is described in Section 4.2. In summary, the runtime for each iteration is $\mathcal{O}(|\mu||\nu|)$, so the runtime for SVM$_{PR}$ training is $\mathcal{O}(T|\mu||\nu|)$, given precalculated mapping $\Phi$.

**Algorithm 1** SVM$_{PR}$ Training Algorithm

1: Input: $\mu$, $\nu$, T, $\lambda$
2: $w_0$=**0**
3: **for** t = 1, 2, . . ., T **do**
4: $A$=$\emptyset$
5: **for** i = 1, 2, . . ., $|\mu|$ **do**
6: $p^e_{min}$=arg$min_{p_j \in \nu \wedge C_{ij}=2} w_{t-1}^T \Phi(p_i,p_j)$
7: $p^a_{max}$=arg$max_{p_j \in \nu \wedge C_{ij}=1} w_{t-1}^T \Phi(p_i,p_j)$
8: $p^a_{min}$=arg$min_{p_j \in \nu \wedge C_{ij}=1} w_{t-1}^T \Phi(p_i,p_j)$
9: $p^n_{max}$=arg$max_{p_j \in \nu \wedge C_{ij}=0} w_{t-1}^T \Phi(p_i,p_j)$
10: **if** $w_{t-1}^T(\Phi(p_i,p^e_{min}) - \Phi(p_i,p^a_{max})) < 1$ **then**
11: $A$=$A \cup (p^e_{min}, p^a_{max})$
12: **end if**
13: **if** $w_{t-1}^T(\Phi(p_i,p^a_{min}) - \Phi(p_i,p^n_{max})) < 1$ **then**
14: $A$=$A \cup (p^a_{min}, p^n_{max})$
15: **end if**
16: **end for**
17: $w_t$=(1-$\frac{1}{t}$)$w_{t-1}$+$\frac{1}{\lambda t|\mu|} \sum_{(p_j,p_k)\in A}(\Phi(p_i,p_j) - \Phi(p_i,p_k))$
18: **if** $\|w_t\| > \frac{1}{\sqrt{\lambda}}$ **then**
19: $w_t$=$\frac{1}{\sqrt{\lambda}} \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|}$
20: **end if**
21: **end for**
22: Output: $\mathbf{w}_t$ with the minimum validation error.

Theoretically, we can show that the number of iterations needed for SVM$_{PR}$ to converge to a solution of accuracy $\epsilon$ from an optimal solution is $\tilde{\mathcal{O}}(\frac{R^2}{\lambda\epsilon})$ where $R$=$\max_{p_i\in\mu, p_j\in\nu} 2\|\Phi(p_i,p_j)\|$. This result follows from Corollary 1 of [Shalev-Shwartz *et al.*, 2007]. In practice, our training algorithm always completes within five hours of runtime in the experiments.

## 4 Empirical Results

### 4.1 Dataset

For our experiments we focused on Wireless patents granted by the USPTO. We started with data from 2001 since this is the first year USPTO released data differentiating examiner and applicant citations. We used a list of Essential Wireless Patents (EWP), a set of patents that are considered essential for the wireless telecommunication standard specifications being developed in 3GPP – Third Generation Partner-

[1]Lemur Toolkit v4.5, copyright (c) 2008 University of Massachusetts and Carnegie Mellon University

ship Project – and declared in the associated ETSI (European Telecommunications Standards Institute) database. We considered three versions of the dataset: the original patent files, the patent files after applying a Porter stemmer [Porter, 1980], and the patent files after applying a Porter stemmer and common stopwords removal. The Porter stemmer reduces different forms of the same word to its original "root form", and the stopword removal eliminates the influence of common but non-informative words, such as "a" and "maybe", in the ranking algorithms.

In our experiment, $\mu$ is the set of essential wireless patents (2001-2007) that made at least one citation to any other patent in 2001-2007, and $\nu$ is the set of patents from 2001-2007 that has been cited by any patent in $\mu$. This dataset currently contains $\sim$197,000 patent-pair citation judgements. This is significantly larger in scale than the OHSUMED dataset widely used as a benchmark dataset for information retrieval tasks, which contains 16,140 query-document pairs with relevance judgement [Hersh *et al.*, 1994]. Our goal is to learn a good discriminant function $\mathbf{w}^T\Phi(p_i, p_j)$ for $p_i \in \mu$ and $p_j \in \nu$. We randomly split the patents in $\mu$ into 70% training, 10% validation and 20% test set in 10 independent trials to assess the performance of SVM$_{PR}$ and other benchmark methods.

### 4.2 Performance Measure

Given a patent in $\mu$, we rank all patents in $\nu$ by the score of $\mathbf{w}^T\Phi$, and evaluate the performance using the Normalized Discounted Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2000].

NDCG is a widely used performance measure for multi-level relevance ranking problems. NDCG incorporates not only the position but also the relevance level of a document in a ranked list. In our problem, there are three levels of relevance between any two patents $p_i$ and $p_j$, as defined by $C_{ij}$, with 2 the most relevant and 0 the least. In other words, if $p_j$ is cited by an examiner in $p_i$, it has a relevance value of 2, and so on. Given an essential wireless patent $p_i$, and a list of patents $\pi$ from $\nu$, sorted by the relevance scoring function, the NDCG score for $p_i$ at position $k$ ($k \leq |\nu|$) is:

$$NDCG_{p_i}@k = N_{p_i} \sum_{j=1}^{k} \frac{2^{C_{i\pi_j}} - 1}{log(j+1)} \quad (4)$$

$N_{p_i}$ is the normalization factor so that the perfect ranking function, where patents with higher relevance values are always ranked higher, will have a NDCG of 1. The final NDCG performance score is averaged over all essential patents in the test set.

### 4.3 Benchmark Methods

In this section, we briefly describe the six ad-hoc IR methods implemented by the Lemur IR package (with default parameters) that we used as benchmarks, and how they are used to rank the patent citations. Given a query $q$ and a document $d$, each of the six methods can be described as a function $\psi(q, d)$ whose value, a real number, is often regarded as the measure of relevance. The six methods are presented in Table 1. Details of the last three KL-Divergence methods with different smoothing priors can be found in [Zhai and Lafferty, 2001].

Table 1: Ad-hoc IR Methods as Benchmark

| Method | $\psi(q, d)$ |
|---|---|
| *TFIDF* | Term freq(q,d)*Inv. doc freq(q) |
| *Okapi* | The BM25 method in [Robertson *et al.*, 1996] |
| *Cossim* | Cosine similarity in vector space |
| $KL_1$ | KL-Divergence with a Dirichlet prior |
| $KL_2$ | KL-Divergence with a Jelinek-Mercer prior |
| $KL_3$ | KL-Divergence with an absolute discount prior |

For each of the six ranking methods, given a wireless essential patent $p_i \in \mu$, we score it with all patents in $\nu$, by treating $p_i$ as the query and $\nu$ as the pool of documents. The methods are evaluated using NDCG with the ranked patent lists. Since we used the patent date feature in SVM$_{PR}$ which effectively indicates that certain citations are impossible, to be fair for the benchmark methods, we set all returned scores $\psi(p_i, p_j)$ from the benchmark methods to -$\infty$, if patent $p_i$ is an earlier patent than $p_j$.

### 4.4 NDCG Results

We evaluate the NDCG at positions 3, 5, 10, 20, and 50. The NDCG score is averaged using 10 independent trials. For SVM$_{PR}$, the maximum number of iterations is 200, and the test performance is evaluated when the best validation performance is achieved within the iteration limit. For the benchmark methods, the performance is reported on the same test sets as SVM$_{PR}$. The results are presented in Figure 2. First of all, SVM$_{PR}$ outperforms the benchmark methods by a significant margin for all five positions. Referring to Table 2 for the numerical comparison with the best performance of any benchmark method, SVM$_{PR}$ outperforms the best result of the benchmark methods by 16% to 42%. Among all benchmark methods, the KL-Divergence with Dirichlet prior scored the highest, with more than 60% of all tests. Comparing the different document pre-processing procedures, we found that applying the Porter stemmer alone actually hurts the performance of SVM$_{PR}$ by a significant 10% to 17% in comparison to using the original patent documents, while the influence on the benchmark methods is marginal. The overall best performance is achieved with SVM$_{PR}$ when applying the stemming and stopword removal, as highlighted in Table 2. All the performance differences between SVM$_{PR}$ and the best alternative benchmark are significant by a signed rank test at the 0.01 level.

### 4.5 Results on A Random Dataset

We repeated the experiments on a non-homogeneous random dataset to understand better whether SVM$_{PR}$ learns and benefits from the internal structure of the patent set $\mu$, and justify our decision to group homogeneous patents together during training.

We randomly sampled the set $\mu$, and among the patents cited by patents in $\mu$, we randomly selected the set $\nu$, while keeping the same number of patent-pair citation judgments as before. We then repeated the experiments described above with the Porter stemmer and stopword removal applied. Now

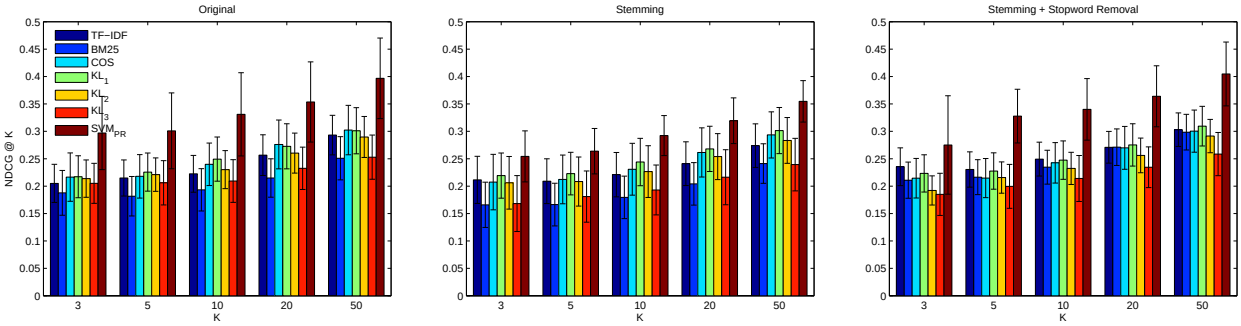


Figure 2: NDCG Scores of SVM$_{PR}$ and Benchmark Methods

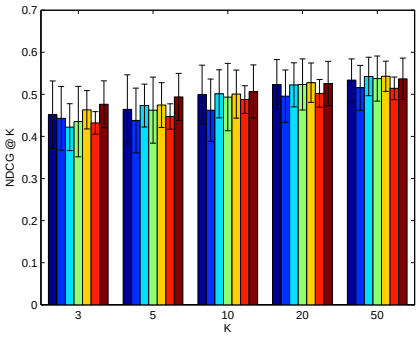


Figure 3: NDCG Scores on A Random Dataset

instead of a structured essential patent set, $\mu$ is an arbitrary set with little internal similarities among its members. Because the patents in $\mu$ are quite unrelated, the patents they cite in $\nu$ are non-homogeneous too. In other words, this is an easier task than the previous one since we are learning to rank patents in $\nu$ that are more distinguishable than before.

The results are presented in Figure 3. We observed that the new performance differences among SVM$_{PR}$ and other benchmark methods are largely indistinguishable (best alternative method performance is within 5% of SVM$_{PR}$). This follows from our intuition that the random dataset lacks a homogeneous citation structure to be learned, and the reasonable methods would perform comparably well, although the learned ranking is less informative as it only differentiates irrelevant patents.

## 5 Conclusion

In this paper we focused on the problem of patent prior art search which is traditionally a tedious task requiring significant expert involvement. Our proposed approach based on large margin optimization incorporates constraints that directly capture patent ranking specificities and ranks patent citations to previously granted patents by a linear discriminant function $\mathbf{w}^T\Phi$, where $\mathbf{w}$ is the learned parameter and $\Phi$ is the feature map vector consisting of patent domain knowledge features and meta score features. Experiments on a wireless technology patent set show that SVM$_{PR}$ consistently outperforms other state-of-the-art general IR methods, based on the NDCG performance measure.

## Acknowledgments

We thank Bill Lesser and Aija Leiponen for useful discussions about patents and Aija Leiponen for her suggestions concerning wireless patents as well as for providing us with the essential wireless patent data. We thank Thorsten Joachims for the discussions on ranking with margin based methods. We also thank the reviewers for their comments and suggestions. This research was supported by AFOSR grant FA9950-08-1-0196, NSF grant 0713499 and NSF grant 0832782.

## References

[Alcacer and Gittelman, 2006] Juan Alcacer and Michelle Gittelman. How do i know what you know? patent examiners and the generation of patent citations. *Review of Economics and Statistics*, 88(4):774–779, 2006.

[Burges *et al.*, 2005] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine Learning*, 2005.

[Burges *et al.*, 2006] C. J. C. Burges, R. Ragno, and Q.V. Le. Learning to rank with non-smooth cost functions. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, 2006.

[Cao *et al.*, 2006] Yunbo Cao, Jun Xu, Tie Yan Liu, Hang Li, Yalou Huang, and Hsiao Wuen Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference*, 2006.

[Carterette and Petkova, 2006] Ben Carterette and Desislava Petkova. Learning a ranking from pairwise preferences. In *Proceedings of the annual international ACM SIGIR conference*, 2006.

[Chapelle *et al.*, 2007] Olivier Chapelle, Quoc Le, and Alex Smola. Large margin optimization of ranking measures. In *NIPS Workshop on Machine Learning for Web Search*, 2007.

[Chu and Keerthi, 2005] Wei Chu and S. Sathiya Keerthi. New approaches to support vector ordinal regression. In

Table 2: SVM$_{PR}$ and Benchmark Performance Comparison

| K | Original | | | Stemming | | | Stemming and Stopword | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\psi_{best}$ | $SVM_{PR}$ | Impr.(%) | $\psi_{best}$ | $SVM_{PR}$ | Impr.(%) | $\psi_{best}$ | $SVM_{PR}$ | Impr.(%) |
| 3 | 0.217 | **0.297** | 36.7 | 0.219 | 0.254 | 16.0 | 0.235 | 0.275 | 16.9 |
| 5 | 0.226 | 0.301 | 33.4 | 0.223 | 0.264 | 18.3 | 0.230 | **0.328** | 42.3 |
| 10 | 0.249 | 0.331 | 32.8 | 0.244 | 0.292 | 19.8 | 0.249 | **0.340** | 36.4 |
| 20 | 0.276 | 0.354 | 28.1 | 0.268 | 0.319 | 19.1 | 0.275 | **0.364** | 32.3 |
| 50 | 0.302 | 0.397 | 31.2 | 0.301 | 0.355 | 17.7 | 0.310 | **0.405** | 30.8 |


*Proceedings of the International Conference on Machine Learning*, 2005.

[Herbrich *et al.*, 2000] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, 2000.

[Herschtal and Raskutti, 2004] A. Herschtal and B. Raskutti. Optimising area under the ROC curve using gradient descent. In *Proceedings of the International Conference on Machine Learning*, 2004.

[Hersh *et al.*, 1994] William R. Hersh, Chris Buckley, T. J. Leone, and David H. Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR Conference*, 1994.

[Järvelin and Kekäläinen, 2000] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference*, 2000.

[Joachims, 2002] T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2002.

[Lampe, 2007] Ryan Lampe. *Strategic Citation*. unpublished working paper, 2007.

[Morik *et al.*, 1999] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach. In *Proceedings of the International Conference on Machine Learning*, 1999.

[Nallapati, 2004] Ramesh Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR Conference*, 2004.

[Porter, 1980] M.F. Porter. An algorithm for suffix stripping. In *Program 14(3)*, pages 130–137, 1980.

[Robertson *et al.*, 1996] S. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3 (1996). In *Text REtrieval Conference*, 1996.

[Sampat, 2004] B. Sampat. Examining patent examination: an analysis of examiner and applicant generated prior art. In *National Bureau of Economics, 2004 Summer Institute, Cambridge, MA*, 2004.

[Shalev-Shwartz and Singer, 2006] Shai Shalev-Shwartz and Yoram Singer. Efficient learning of label ranking by soft projections onto polyhedra. *J. Mach. Learn. Res.*, 7:1567–1599, 2006.

[Shalev-Shwartz *et al.*, 2007] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, 2007.

[Yue *et al.*, 2007] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference*, 2007.

[Zhai and Lafferty, 2001] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Research and Development in Information Retrieval*, 2001.