

# The Popularity of Articles in PubMed

L. Smith\* and W.J. Wilbur

National Center for Biotechnology Information, Bethesda, Maryland, USA

**Abstract:** The PubMed search engine displays query results in reverse chronological order, which is appropriate for users interested in the latest publications. The purpose of this paper is to use machine learning to order documents by popularity, or the predicted frequency that an article is viewed by the average PubMed user. Other research on general search engine usability has applied machine learning to order documents by their relevance to a given query. The approach here takes a global view of popularity across all users in a given time period, independent of their information need. An effective method for learning popularity from clickthrough data is identified, and a novel measure of success in this task is proposed. The resulting model shows that the topic of an article has the largest single influence on its popularity, and its publication date has a strong secondary influence. Possible applications and extensions are discussed.

**Keywords:** Clickthrough data, document ranking, machine learning.

## INTRODUCTION

The PubMed<sup>1</sup> search engine displays biomedical citations for MEDLINE documents that match a user query [1]. By default, results are displayed in reverse order that they are entered into the database. This method of ranking is appropriate for users who are interested in finding the most recent publications in a particular field. But PubMed serves a diverse population of user interests, and its usability depends on how well it serves those interests, which in turn depends on how well those interests are understood. This paper demonstrates that it is feasible to discover the features of documents that predict their popularity over a given time period.

PubMed usability has been the subject of previous research. For example, a method known as “PubMed related articles” [2] was developed to suggest related articles to a user, given that they have chosen to view a particular article. Our approach takes a global view, to determine which articles are interesting to PubMed users in general, without any information about a particular user’s information needs or interests.

There is also an active area of research known as “learning to rank,” which is aimed at improving the usability of search engines in general [3]. The standard setup is to devise a learning scheme that is either based on user relevance feedback, which is costly to obtain, or based on web server logs, which are commonly available and voluminous [4]. These schemes seek to discover an intelligent “retrieval function,” which scores documents by their relevance to a particular query.

Our approach may be viewed as a simplified version of learning-to-rank, since the result is a global ranking of

documents in order of their predicted popularity. When applied to a particular query, the reordered results seek to place those matching documents first that were globally more popular.

Server Logs record accesses to individual documents, from which it is easy to determine the number of times each document was accessed during a particular time interval. This *click count* data is believed to be an important ingredient in satisfactory ordering of search results. The goal of this paper is to investigate whether or not it is possible to automatically predict the ordering by click count for PubMed articles based solely on their contents. This would allow the broad popularity of new articles to be estimated even before they are available to users. However, this approach ignores the variability of user interests, and accounting for specific user interests is a possible area of future research [5].

## METHODS

To test our hypothesis that popularity has a document model, click count data was collected from PubMed web server logs for all MEDLINE documents viewed over a six month time period. Several baseline and machine learning methods were employed to predict the ranking of documents by click count, and an intuitive and practical measure of prediction called *click recall* was introduced.

### Click Count Data

Click counts were collected from PubMed web logs over six months in 2008. A filtering process was applied to insure that the clicks represented actual human viewers (that is, eliminating article accesses by automated web crawlers, robots, or “spiders”). Some of the documents were observed to have a high degree of variability in click count from month to month. So, to reduce the influence of this unexplained effect, the click count for each document was defined as the *truncated sum* of clicks over four of the six months. This is analogous to the *truncated mean*, a well-known statistical method for mitigating possible outliers by first removing the extreme values. In this case the highest and lowest click

\*Address correspondence to this author at the Computational Biology Branch, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, Maryland, NCI, USA; Tel: 240-535-2217; Fax: 301-480-2290; E-mail: lsmith@ncbi.nlm.nih.gov

<sup>1</sup><http://www.ncbi.nlm.nih.gov/pubmed/>

counts were removed before summing. This resulted in 11.4 million documents with a nonzero click count sum, the largest being 27,347 for a 2002 article on large scale genome analysis [6].

The full set of documents was taken from a snapshot of MEDLINE records available on March 1, 2008 (the starting date of the click count data). There were a total of 17.4 million records. Documents in the top 25% of click count were labeled *positive*, and the remaining 75% *negative*.

The corpus was randomly divided into several subsets for machine learning. Two training subsets were constructed. A *default training set* was formed by randomly sampling two thirds of the positive documents and two thirds of the negative documents. An *enhanced training set* was constructed with the same positive documents as the default training set, together with all of the documents in the bottom one sixth of click count as negatives. Finally, a *test set* was formed from all documents not appearing in either of the training sets, and all comparisons were made on this held-out set. The sizes of the sets, and the numbers of positive and negative documents in each, are shown in Table 1.

**Table 1. Composition of Corpus. The Total Size, Number of Positive Documents, and Number of Negative Documents in the Corpus, Default Training Set, Enhanced Training Set, and Test Set**

Set	Size	Positives	Negatives
Corpus	17,357,380	4,339,346	13,018,034
Default Training	11,571,586	2,892,897	8,678,689
Enhanced Training	5,785,793	2,892,897	2,892,896
Test	4,821,495	2,892,896	3,375,046

### Baseline Methods

Five baseline (untrained) methods were considered for ranking documents to predict popularity.

1. PubMed ID: the default sort order in PubMed, which is the reverse order in which documents were received.
2. Publication date: very close to PubMed ID.
3. Related click counts: the average click count of 100 “related articles” for each document (excluding the document itself). Related articles were determined by the algorithm described in [2]. The click counts in the average were weighted by the related-article score. This is motivated by the hypothesis that documents related to other popular documents should themselves be popular.
4. Document length: the number of features in a document. This is motivated by the observation that popular documents often had very lengthy abstracts or author lists.
5. Random. In order to assess statistical significance, the average performance was obtained for 100 random orderings.

### Machine Learning

Machine learning determines numeric weights for a set of features associated with each document. Documents are then scored by summing the weights of their features and ordered by this score. Our goal is to exhibit an effective machine learning algorithm by varying the “cost model” while holding other variables fixed. To this end, we chose a simple and standard method for identifying features.

The features for each document were taken from words in the title, author, sponsoring institution, journal name, and abstract fields of the MEDLINE record. Words were delimited by non-alphanumeric characters, and required to contain at least one letter. Each word was mapped to lower case, marked to indicate the field it came from, and recorded as a feature for the document (the number of times a feature appeared in a document was not recorded). The publication date was also coded into one of eight features depending on the time elapsed from the publication date up until March 1, 2008. These correspond to (1) pre-publication, (2) zero to two months, (3) two to six months, (4) six to twelve months, (5) one to two years, (6) two to five years, (7) five to ten years, and (8) more than ten years.

Three machine learning methods were compared: category learning, ordinal regression, and log-linear probability. Each method determines feature weights by empirical risk minimization [7], minimizing an associated cost function. Let  $\bar{X}_i$  denote the binary feature vector for training document  $i$  and let  $y_i = 1$  for positive documents (i.e. top 25% of click count) and  $y_i = -1$  for all other documents.

For category learning, we will use  $z_i$  to denote the score of document  $i$ :

$$z_i(\theta, \bar{w}) = \theta + \bar{w} \cdot \bar{X}_i \quad (1)$$

where  $\theta$  is a threshold parameter,  $\bar{w}$  is a weight vector,  $\bar{X}_i$  is the (binary) feature vector, and  $\bar{w} \cdot \bar{X}_i$  is the dot product. Note the dependence of  $z_i$  on the parameters  $\theta$  and  $\bar{w}$ . Category learning determines a threshold  $\theta$  and weight vector  $\bar{w}$  that minimizes an empirical cost function of the form:

$$C = \frac{1}{2} \lambda |\bar{w}|^2 + \frac{1}{T} \sum_{i=1}^T f_m(y_i, z_i(\theta, \bar{w})) \quad (2)$$

where  $f_m$  is a loss function, giving a positive loss for misclassification. Here,  $T$  is the size of the training set. We considered several loss functions given in Table 2 and described below. The regularization parameter  $\lambda$  is equal to

$$\lambda = \lambda' \left\langle |\bar{X}| \right\rangle^2$$

where  $\left\langle |\bar{X}| \right\rangle$  is the average Euclidean norm of the feature vectors in the training set, and  $\lambda'$  must be chosen. In a preliminary study, the optimal  $\lambda'$  was found to be  $10^{-5}$  (to a power of 10), and this value was used in all experiments.

**Table 2. Loss Functions in Machine Learning. The Arguments  $z_i$  and  $z_j$  are Document Scores for Documents  $i$  and  $j$  Respectively (Equation (1) for Category Learning and Equation (3) for Ordinal Regression), and  $y_i$  is the Class Indicator. The Variables  $n_i$  and  $n_j$  Denote the Length (Number of Features) of the Corresponding Document**

Num	Description	Category Learning (single document)	Ordinal Regression (pair of documents)
1	Support vector machine (hinge loss)	$f_1(y_i, z_i) = \begin{cases} 1 - y_i z_i & \text{if } y_i z_i \leq 1 \\ 0 & \text{if } 1 \leq y_i z_i \end{cases}$	$g_1(z_i, z_j) = f_1(1, z_i - z_j)$
2	Modified Huber	$f_2(y_i, z_i) = \begin{cases} -4y_i z_i & \text{if } y_i z_i \leq -1 \\ (1 - y_i z_i)^2 & \text{if } -1 < y_i z_i \leq 1 \\ 0 & \text{if } 1 \leq y_i z_i \end{cases}$	$g_2(z_i, z_j) = f_2(1, z_i - z_j)$
3	Modified Huber (length norm)	$f_3(y_i, z_i) = f_2(y_i, z_i / n_i)$	$g_3(z_i, z_j) = f_2(1, z_i / n_i - z_j / n_j)$
4	Modified Huber (root norm)	$f_4(y_i, z_i) = f_2(y_i, z_i / n_i^{1/2})$	$g_4(z_i, z_j) = f_2(1, z_i / n_i^{1/2} - z_j / n_j^{1/2})$

For ordinal regression we will likewise use  $z_i$  to denote the score of document  $i$ , which is the same as for category learning but without the threshold parameter:

$$z_i(\bar{w}) = \bar{w} \cdot \bar{X}_i \quad (3)$$

Ordinal regression [8, 9] determines a weight vector  $\bar{w}$  that minimizes an empirical cost function of the form:

$$C = \frac{1}{2} \lambda |\bar{w}|^2 + \frac{1}{T} \sum_{(i,j)} g_m(z_i(\bar{w}), z_j(\bar{w})) \quad (4)$$

where  $g_m$  is a loss function, giving a positive loss for misordering. It is important to note the summation here is over a set of pairs  $(i, j)$  sampled from the training set (default or enhanced) such that  $y_i = 1$  and  $y_j = -1$ , in other words, they are given in the correct ordering. The number of pairs sampled for each  $i$  is variable and will be denoted by  $M$ . The values  $M = 1$  and  $M = 10$  were considered.

Two functions were used in combinations to define loss functions shown in Table 2: the *hinge loss* of Support Vector Machines [9] and the *modified Huber loss* [10]. In addition, two different ways of normalizing for the total number of features in a document were considered, motivated by the observation that some documents have an unusually large number of features, and might randomly receive a high score, crowding out better candidates from the highest ranks. In length normalization, the  $z_i$  are divided by the number of features appearing in document  $i$ ,  $n_i$ . In root normalization, the scores are divided by  $n_i^{1/2}$ . All formulas are shown explicitly in Table 2.

Finally, a log-linear, maximum likelihood model was applied to the click count data. A score for documents in this model,  $z_i$ , is calculated by equation (3). The log linear model determines a weight vector  $\bar{w}$  that minimizes the

negative log of the likelihood, assuming an exponential form for the probability:

$$H = -\sum_{i=1}^T C_i z_i(\bar{w}) + C \log(Z) \quad (5)$$

$$Z = \sum_{i=1}^T \exp(z_i(\bar{w})), \quad C = \sum_{i=1}^T C_i$$

Here  $C_i$  is the click count of the  $i$ -th document in the training set. This method did not require regularization.

All of the methods were minimized using a gradient descent algorithm, implemented in a parallel computing environment.

#### Average Click Recall

Two measures were calculated for comparing predicted rankings by popularity. First, the standard statistic for comparing rankings is the Spearman rho, which is simply the correlation of rank values. However, this statistic is relatively insensitive to differences in the top ranks, and provides no information regarding performance on subsets which might be returned from a query. An average normalized cumulative gain statistic [11] was therefore developed that avoids both of these defects.

Given a query  $Q$ , the documents in the test set that satisfy it can be reordered according to the ranking under consideration. The cumulative gain (at position  $p$ ), as defined in [11] (and see also [12]) is:

$$CG_p(Q) = \sum_{i=1}^p rel_i, \quad (6)$$

where  $rel_i$  is the relevance (to the query) of the document at position  $i$ , and which we take to be the click count for that document. The maximum attainable value of  $CG_p(Q)$  is called the ideal cumulative gain, and denoted  $ICG_p(Q)$  (this can be obtained simply by using the order determined by

$rel_i$  itself). The normalized cumulative gain is the ratio, a number between 0 and 1, which we call the *click recall*:

$$CR_p(Q) = CG_p(Q) / ICG_p(Q). \quad (7)$$

The final statistic for a ranking, the *average click recall at 20*, or  $ACR_{20}$ , is defined by averaging the  $CR_{20}(Q)$  over a large collection of typical PubMed queries. The number 20 was chosen because it is the default number of articles presented by PubMed in response to a user query.

The set of queries used to average  $ACR_{20}$  were obtained from the words in the NLM Medical Subject Headings [13], corresponding to a large variety of biomedical topics assigned by human indexers to records in MEDLINE. In all, there were just over nine thousand unique queries.

In order to establish a significance level for the  $ACR_{20}$ , it was computed for each of 100 random rankings. The maximum value of those 100 values, 0.0810, is an estimate for the 1% significance level. In other words, we can say  $p < 0.01$  whenever  $ACR_{20} > 0.0810$  (computed on the same test set).

## RESULTS

The  $ACR_{20}$  and Spearman's rho for all experiments is shown in Table 3. All of the  $ACR_{20}$  results are statistically significant compared to random ( $p < 0.01$ ), based on random sampling. All of the Spearman rho results are also statistically significant (any rho  $> 0.0013$  has  $p < 0.01$ ).

The best baseline method was ordering by date ( $ACR_{20} = 0.4371$ ), which is very close to the default PubMed ID ordering. The best machine learning method was log-linear ( $ACR_{20} = 0.6886$ ), and it performed substantially better than all other machine learning methods ( $ACR_{20} = 0.6007$  for category training). Ordinal regression did best on the enhanced training set with  $M = 10$  pair over-sampling ( $ACR_{20} = 0.5256$ ), but was still well below the category learning performance. Finally, there was no improvement from length normalization of any of the category or ordinal models.

## DISCUSSION

### Baseline Results

Among the baseline methods, date ordering performed the best on the test set with  $ACR_{20} = 0.4371$ . This was slightly better than the default ordering by PubMed ID with  $ACR_{20} = 0.4268$ . Ordering by related clicks was a close third with  $ACR_{20} = 0.4080$ , providing the first evidence that articles are popular if their topic is popular. When documents are ordered by their length, there is a small but significant result,  $ACR_{20} = 0.2792$ , which is explored further below.

### Log-Linear Model

The  $ACR_{20}$  performance of the log-linear probability model is much higher than any other method. This is consis-

tent with the hypothesis that click count is best viewed as a probabilistic phenomenon. Another explanation is that the log-linear model uses more information in the click count magnitudes than category learning (which uses only quartiles of click count) and ordinal regression (which uses only the relative order of click counts).

### Ordinal Regression

The results also imply that category learning (based on quartiles) is more effective than our ordinal regression approach. We believe the explanation is simple. The total number of available ordered pairs is  $2.5 \times 10^{13}$  in the default training set and  $8.4 \times 10^{12}$  in the enhanced training set, while the best performance for ordinal regression was seen in the enhanced training set with  $M = 10$  using only  $2.1 \times 10^7$  pairs. A higher sampling rate would likely perform much better (although such experiments are prohibitively large).

### Document Length

The machine learning methods tended to rank documents with more features higher. For example, in the log-linear model, the correlation between rank and document length was -0.6398. To investigate whether this correlation was meaningful or accidental, the loss functions were normalized in two different ways. But in all normalization experiments, performance was degraded, suggesting that document length does indeed provide some information about whether a document is popular. A possible explanation for this correlation is that popular subjects are also more heavily funded, and the resulting articles may contain more results than average, making them longer.

### Model Features

Out of 5.5 million possible features, one half million had weights of absolute magnitude 0.001 or greater. Some features in each category with highest magnitude are shown in Table 4. As an indicator of the significance of these magnitudes, document scores from the log-linear model on the test set ranged from -4.57 to 8.26, and the median document score was -0.79.

Each of these features (with negative weights included) can be intuitively connected with popular trends or preferences. The list in Table 4 has some notable implications. In the publication date features column, for example, the preference for recent documents can be clearly seen in the decreasing weights. And in the abstract features column, the words "the" and "and" are among the largest positive weights, which acts to score documents with abstracts higher than those without (since most abstracts contain the words "the" and "and"). However, it is problematic to generalize about topic popularity based on individual weights of abstract features. Since the majority of features for a document come from its abstract (when it has one), their individual weights may be small while their combined contribution may still dominate the document score.

### Feature Subsets

The log-linear model was also trained on varying subsets of features. The single field that gave the highest click recall

**Table 3. Machine Learning Performance. The Cross-Validated  $ACR_{20}$  and Spearman's rho for Baseline and Machine Learning of Overall Popularity**

<b>Baselines</b>			
<b>Sort Method</b>		$ACR_{20}$	<b>rho</b>
PubMed ID		0.4268	0.4312
date		0.4371	0.6376
date (all docs <sup>1</sup> )		0.3448	0.6376
related clicks		0.4080	0.5909
record length		0.2792	0.5709
random <sup>2</sup>		0.0810	-
random (all docs <sup>1</sup> )		0.0469	-
<b>Machine Learning Models<sup>3</sup></b>			
<b>Type</b>	<b>Loss</b>	<b>Train</b>	<b>M</b>
		$ACR_{20}$	<b>rho</b>
C	1	D	
C	1	E	
C	2	D	
C	2	E	
O	1	D	1
O	1	D	10
O	1	E	1
O	1	E	10
O	2	D	1
O	2	D	10
O	2	E	1
O	2	E	10
C	3	D	
O	3	E	10
C	4	D	
O	4	E	10
<b>Log-Linear Model</b>			
<b>Sort Method</b>		$ACR_{20}$	<b>rho</b>
log-linear		0.6886	0.7655
log-linear (all docs <sup>2</sup> )		0.6365	0.7661
<b>Log-Linear Model with Feature Subsets</b>			
<b>Feature Subset</b>		$ACR_{20}$	<b>rho</b>
abstract		0.5052	0.6681
date		0.4280	0.5969
abstract + date		0.6292	0.7376
abstract + date + journal		0.6623	0.7456

<sup>1</sup>Cross-validation result on full corpus. All other results on evaluated on the (same) test set.

<sup>2</sup>The values shown for "random" are the maximum  $ACR_{20}$  observed in 100 random orderings, which is an estimate for the 99% confidence level.

<sup>3</sup>First column, Type: category learning (C), ordinal regression (O); Loss: hinge loss (1), Huber loss (2), Huber loss with length normalization (3), Huber loss with root normalization (4); Train: default training set (D), enhanced training set (E); M: ordinal regression multiplier, 1 or 10. See text for details.

**Table 4. Features and Weights. As Determined by the Log-Linear Model to Predict Popularity, these Features had the Highest Magnitude Weights in each Class of Features**

Publication Date		Abstract		Title		Journal		Institution	
pre-pub	0.15	autism	0.39	autism	0.40	nature	0.85	cambridge	0.27
0-2 months	0.74	autophagy	0.33	guidelines	0.39	lancet	0.76	switzerland	0.27
2-6 months	0.44	micornas	0.30	pathophysiology	0.32	jama	0.65	netherlands	0.27
6-12 months	0.19	the	0.29	author	-0.32	bmj	0.62	rockefeller	0.26
1-2 years	-0.14	penis	0.28	available	-0.31	neuron	0.62	physics	-0.25
2to5 years	-0.56	sox2	0.27	transl	-0.31	optics	-0.60	sweden	0.25
5-10 years	-0.97	mir	0.26	anatomy	0.28	new	0.58	germany	0.25
>10 years	-1.57	and	0.26	statement	0.28	annual	0.54	stanford	0.25
		foxp3	0.25	meta	0.27	zealand	-0.54	oxford	0.22
		th17	0.25	rabbit	-0.26	circulation	0.53	france	0.22

was the abstract field, with  $ACR_{20} = 0.5052$ , which is consistent with the interpretation that popularity depends primarily on a document having an abstract that discusses a popular topic. The next most significant single field was the publication date  $ACR_{20} = 0.4280$ , which is near the baseline result for date (but lower because the features are date ranges instead of actual dates as in the baseline). The best pair of fields was the combination of abstract and date, with  $ACR_{20} = 0.6292$ , and the best triple of fields is abstract, date, and journal, with  $ACR_{20} = 0.6623$ . But the gain from adding more fields is too small to draw definite conclusions. The implication of this is that popularity is defined first by topic, then by publication date, and finally, and to a lesser extent, by considering the journal of publication. The influence that individual authors and institutions have on popularity is almost completely subsumed by those three considerations.

### Scoring all of MEDLINE

In order to simulate the situation where one does not have click counts, we applied our model to obtain a score for every document in MEDLINE. To make this realistic, the log-linear model was trained three times, each time holding out a different third from the corpus for scoring. The three sets of scored documents were then merged to give cross-validated scores for all documents. The document with largest cross-validated score had a click count of 3,185 (rank 60 out of 17.4 million). This article, published in 2008, is a review of autophagy in higher eukaryotes [14].

With these combined  $ACR_{20} = 0.6365$ , which is still high compared with  $ACR_{20} = 0.3448$  obtained from date ranking of all documents. Note that the  $ACR_{20}$  is slightly lower on the entire corpus than 0.6886 computed on the test set. This is because the ideal cumulative gain used in equation (7) is smaller for the test set than for the entire corpus, as it is a maximization taken over a smaller set.

To illustrate the application of these scores, we applied them to the PubMed query "Parkinson's Disease" (PD). This

query returned 45,151 documents in the full corpus, and these were reordered using the cross-validated document scores. The top twenty of these documents were published over a period from 2004 to 2008. Fourteen were reviews, and only one journal, *Recent Pat CNS Drug Discover*, was represented twice. The highest click count appearing among these top twenty documents was 2,006, a very general review article about oxidative cell damage [15]. The lowest click count in the top twenty lists was 416 (the average click count over all 45,151 documents was 31.7 with a median of 10). In summary, of these twenty documents, ten are about pathophysiology of Parkinson's disease, or classes of neurodegenerative diseases which include Parkinson's disease. Eight of the documents deal with therapeutic options that are applicable to Parkinson's disease. One discusses an antipsychotic agent which is also a first line agent for the treatment of Parkinson's disease. Finally, the most irrelevant document mentions PD in the name of its sponsoring institution, but not elsewhere.

### CONCLUSIONS AND FUTURE WORK

The results show that it is feasible to learn popularity from click through data, and to reorder an average PubMed query to obtain 63.65% of the maximum possible click count in the top 20 documents. Ranking in this way is more successful at presenting the user with popular documents than date ranking, which gives only 34.48%. It also suggests that click-recall is a meaningful measure of success in ranking popularity, and that the log-linear probability model performs better than category or ordinal models.

Given an effective method for extracting popularity from click through data, there are several different ways that it can be applied. For example, emerging popular topics can be quickly detected, and subtle shifts in popularity can be tracked over time. In fact, our implementation of machine learning (a gradient descent algorithm carried out in a parallel environment on 32 CPUs) produced a result in about three hours, so a popularity model could be updated on a daily basis. Success in this task could further improve popularity predictions and sensitivity to trends.

This approach to global popularity could also be combined with mainstream learning-to-rank research. For example, the extent to which individual user preferences differ from global popularity can be detected. Or, in combination with user clustering, such as [5], popularity could be modeled within broad categories of users, such as lay public versus biomedical researchers.

#### ACKNOWLEDGMENTS

This research was supported by the Intramural Research Program of the NIH, NLM, and NCBI. We would like to thank Zhiyong Lu, Mike DiCuccio and John Moe of the NCBI for providing resource support.

#### REFERENCES

- [1] E.W. Sayers, T. Barrett, D.A. Benson, S.H. Bryant, K. Canese, V. Chetvernin, "Database resources of the National Center for Biotechnology Information", *Nucleic Acids Res.*, vol. 37(Database Issue), pp. 5-15, 2009.
- [2] J. Lin, W. Wilbur, "PubMed related articles: a probabilistic topic-based model for content similarity", *BMC Bioinformatics*, vol. 8, p. 423, 2007.
- [3] H. Li, T-Y. Liu, C.X. Zhai, "*Learning to Rank for Information Retrieval (LR4IR)*", Proceedings of the ACM SIGIR Conference; Boston, MA: ACM 2009, pp. 41-5.
- [4] T. Joachims, "*Optimizing Search Engines Using Clickthrough Data*", Proceedings of the ACM SIGKDD Conference; Edmonton, Alberta: ACM 2002, pp. 133-42.
- [5] J-T. Sun, X. Wang, D. Shen, J-J. Zeng, Z. Chen, "*Mining Clickthrough Data for Collaborative Web Search*", Proceedings of the 15th international conference on World Wide Web; Edinburgh, Scotland: ACM 2006, pp. 947-8.
- [6] R.L. Strausberg, E.A. Feingold, L.H. Grouse, J.G. Derge, R.D. Klausner, F.S. Collins, "Generation and initial analysis of more than 15,000 full-length human and mouse cDNA sequences", *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 26, pp. 16899-903, 2002.
- [7] V. Vapnik, "*The Nature of Statistical Learning Theory*", New York: Springer-Verlag, 1995.
- [8] Y. Cao, J. Xu, T-Y. Liu, H. Li, Y. Huang, H-W. Hon, "Adapting Ranking SVM to Document Retrieval", SIGIR; ACM Press, New York, NY 2006, pp. 186-93.
- [9] T. Joachims, "*Training Linear SVMs in Linear Time*", Proceedings of the ACM SIGKDD Conference; ACM 2006, pp. 217-26.
- [10] T. Zhang, "*Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms*", Proceedings of the Twenty-first International Conference on Machine learning; Omnipress 2004, pp. 918-22.
- [11] K. Järvelin, J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques", *ACM Trans. Inform. Syst.*, vol. 20, no. 4, pp. 422-46, 2002.
- [12] Wikipedia-contributors. Discounted cumulative gain. Wikipedia, The Free Encyclopedia; [8 February 2011]; [http://en.wikipedia.org/w/index.php?title=Discounted\\_cumulative\\_gain&oldid=407363550](http://en.wikipedia.org/w/index.php?title=Discounted_cumulative_gain&oldid=407363550)
- [13] Medical Subject Headings (MeSH). U.S. National Library of Medicine 2009; <http://www.nlm.nih.gov/mesh/>
- [14] D. Klionsky, H. Abeliovich, P. Agostinis, D. Agrawal, G. Aliev, D. Askew, "Guidelines for the use and interpretation of assays for monitoring autophagy in higher eukaryotes", *Autophagy*, vol. 4, no. 2, pp. 151-75, 2008.
- [15] M. Valko, D. Leibfritz, J. Moncol, M. Cronin, M. Mazur, J. Telser, "Free radicals and antioxidants in normal physiological functions and human disease", *Int. J. Biochem. Cell Biol.*, vol. 39, no. 1, pp. 44-84, 2007.

Received: January 24, 2011

Revised: February 11, 2011

Accepted: April 14, 2011

© Smith and Wilbur; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.