The Download Estimation Task on KDD Cup 2003

Janez Brank and Jure Leskovec
Jožef Stefan Institute
Jamova 39, Ljubljana, Slovenia
janez.brank@ijs.si, jure.leskovec@ijs.si
http://ai.ijs.si/kddcup03/

ABSTRACT

This paper describes our work on the Download Estimation task for KDD Cup 2003. The task requires us to estimate how many times a paper has been downloaded in the first 60 days after it has been published on *arXiv.org*, a preprint server for papers on physics and related areas. The training data consists of approximately 29000 papers, the citation graph, and information about the downloads of a subset of these papers. Our approach is based on an extension of the bag-of-words model, with linear SVM regression as the learning algorithm. We describe our experiments with various kinds of features. We focus particularly on issues of feature construction and weighting, which turns out to be quite important for this task.

1. INTRODUCTION

We are given a set of 29014 scientific papers from the "hep-th" (high-energy physics — theory) area of *arXiv.org*, a well-known preprint server. They cover the period from January 1992 to February 2003.

For each paper, we get its full text (in the TEX format) and a separate structured file containing the abstract of the paper and some metadata (such as title and author names). We also get a citation graph, showing all citations where one paper from the dataset cites another paper from the dataset.

For each paper submitted in a 6 month "training period" (1566 papers), we also know how many times (and when) this paper was downloaded from *arXiv*'s web servers in the first 60 days after it was added to the archive.

Our task is to predict the number of downloads (in the first 60 days since their inclusion in the archive) of the papers submitted during a 3 month "test period" (678 papers).

To evaluate a model, the sum of absolute errors (differences between predicted and true download counts) is to be used. However, the KDD Cup rules state that only the 50 most frequently downloaded papers from each of three test months would be used for evaluation. Note that this is only approx. 20 % (150/678 $\approx$ 22 %) of all test papers.

The distribution of download counts is approximately log-normal in shape (average: 193.6; std. dev.: 185.9; median: 142). Most downloads occur within the first few days since a paper is added to the archive, probably because people notice it on the "New" and "Recent" pages of *arXiv.org.*

2. OUR APPROACH

Data representation. In information retrieval and text categorization, the traditional way of working with textual documents is by treating them as "bags of words": the order of words in the document is completely ignored, but the number of occurrences of each word is preserved.

We extend the bag-of-words idea by introducing additional features. One can view a paper as being a bag containing not just words but other kinds of items as well, e. g. authors, citations, a journal, and so on. Equivalently, this can be seen as adding new components to the vectors that represent the documents. Different schemes of normalization and weighting can then be applied to different parts of the vectors. Most of our work focused on experimenting with various sets of features to see which of them lead to more successful models.

To obtain predictors, we use the support vector regression algorithm [5; 2]. Based on experience from the field of text categorization, we will limit ourselves to linear models and perform no feature selection except discarding terms that occur in less than two papers on the whole dataset.

The structure of our experiments. We were given the correct number of downloads for 1566 papers. We used ten-fold cross-validation on this set to compare different models and to choose best features and weights. At the end we trained the final model on the entire training set of 1566 papers. We used this final model to obtain predictions for the 678 test papers which were then submitted to KDD Cup.

The KDD Cup rules state that only predictions on the 50 most frequently downloaded papers of each month will be used for evaluation. This is about 20 % of all papers. As there is no need to predict accurately on the other papers, we will use just the top 30 % of training papers as input to the SVM learner (top 20 % would seem more natural but performance turns out to be better at 30 %), and similarly just the top 20 % of the test papers to evaluate the model.

A trivial model. As a baseline against which to compare other models, we considered a very simple model that always outputs some constant value. To minimize the average absolute error (which will be used for evaluation), it makes sense to use the median download count as our prediction. The average error of this trivial model was quite large (152.5 downloads; the avg. download count over 20 % of the most frequently downloaded papers is around 458.7).

3. ABSTRACT, AUTHOR, ADDRESS

An obvious thing to try is to represent each paper by its

Representation	Average error on the training	test set
Author	63.66	146.38
Abstract	62.46	149.28
Address	80.60	154.06
Abstract + Address	42.20	142.89
Abstract + Author	37.62	**135.80**
Address + Author	49.19	143.38
Abstract + Address + Author	32.29	136.64
1.2 Abstract + 0.6 Address + Author	31.56	**134.70**

Table 1: The performance of various representations based on the author, abstract, and address features.

abstract and title. Both are seen by the readers on the "What's new" page. Each paper is treated as a bag containing the words from the title and abstract. We use the TF-IDF weighting (well-known from information retrieval) and normalize the resulting vector to unit length.

We also extract **author** names and then define one feature for each author. This feature will be nonzero if the corresponding person is an author of the paper. We normalize the vector to unit length.

Another source of information is the **address** of the institution(s) where the authors work. One imagines that a paper is more likely to be downloaded if its authors are from a reputable and well-known institution. We represent a paper by the bag of words from the address (or addresses) found in that paper. The resulting vector is again based on TF-IDF weighting and normalized to unit length.

The results of there experiments are shown in Table 1. The "Author" representation is the best of the three, with an average error of 146.4. Unfortunately, this is a rather small improvement over 152.3, which is the error of the best trivial model from Section 2.

We can also **combine several representations**. In this case vectors resulting from two or more representations are simply concatenated together. Additionally, we also apply different weights to different parts of the combined representation, to adjust their influence on the learning process and the resulting predictions. We choose these weights through cross-validation. We will use this same principle to extend our representations throughout the rest of this paper.

Table 1 shows the results. We decided to adopt the "Abstract + Author" ("AA") representation as the baseline for further experiments. Its performance is only slightly worse than that of the best (but considerably more complex) representation found.

4. USING THE CITATION GRAPH

We were also given the citation graph of 342437 edges which covers all the citations within our dataset of 29014 papers. The citation graph does not contain any information about citations outside the dataset.

We are interested in downloads within the first two months of a paper's presence in the archive. In this time a paper probably does not accumulate many citations, so it seems reasonable to suppose that the number of downloads is not directly influenced by the citations. However, in the long term, citations are an indication of the quality and relevance of the paper; and better and more relevant papers are probably more likely to be downloaded. A heavily downloaded paper is likely to be more widely read, and thus more likely to influence other researchers and be cited by them.

Representation	Average test set error
AA (= Author + Abstract, from Sec. 3)	135.705
AA + 0.004 in-degree	**127.62**
AA + 0.055 authority	128.04
AA + 0.2 PageRank	130.50
AA + 0.1 out-degree	134.69
AA + 0.09 hub	134.97
AA + 0.9 in-links	131.87
AA + 1.0 out-links	132.47
AA + 0.004 in-degree + 0.8 in-links	125.28
AA + 0.004 in-degree + 0.9 out-links	124.23
AA + 0.005 in-degree + 0.5 in-links + 0.8 out-links	**123.72**

Table 2: Summary of experiments with attributes based on the citation graph. The smallest error that that can be achieved by each representation (by selecting suitable weights of the graph-based attributes) is shown.

Weights implied by the graph. Using the citation graph, we can obtain various numerical features for each paper. **In-degree** (the number of times a paper was cited) is widely regarded as an indicator of the quality and relevance of papers. One could also look at the **out-degree**, the number of references to other papers that occur at the end of a paper; there is no obvious reason why this should be relevant (except for survey papers which typically have lots of references and are also read by many people). We also tried HITS (hubs and authorities) [3] and PageRank [4].

In-degree, authority weight, and PageRank are strongly correlated. Similarly, out-degree and hub weight are strongly correlated. These two groups of features are practically independent of each other. None of these features is really strongly correlated to the number of downloads.

The in-degree is a helpful attribute (table 2), but we have to consider the issues of scaling. The attributes in our baseline "AA" representation have values in the range $[0, 1]$. If we simply add the in-degree as a new attribute, with an average value of 11.8, it will appear much more important to the learner than the other attributes. So we multiply the in-degree attribute by some constant weight in order to bring it into the range $[0, 1]$ and prevent it from predominating over other attributes.

Authority has effects similar to those of in-degree. As expected, out-degree and hub weight are not really useful. We also tried using several of these attributes at the same time, but could not improve the accuracy in this way.

Using the connectivity information directly. Another way of using the citation graph is to introduce as many new features as there are documents. If our paper is cited by another paper then the corresponding feature is nonzero. We call this representation **in-links**. Another set of attributes, called **out-links**, can be defined analogously for outgoing citations.

The results are sumarized in table 2. Adding these attributes to the representation significantly decreases the test set error. Weighting these attributes is not really helpful.

5. MISCELLANEOUS STATISTICS

Journal information. Most of the papers have a "Journal" field in their abstract file, usually stating the journal where the paper was or will be published. Although the readers do not see the journal name when downloading the paper, a paper that has appeared in a reputable journal might be an interesting and relevant one and has a larger

		Average CV error		Actual test set
	Representation	Train.	Test	(150 docs.)
Triv.	**model** (median of train. set)	150.18	**152.26**	**181.11**
(AA)	**author + abstract**	37.62	135.86	155.38
	AA + 0.004 in-degree	35.99	127.69	146.77
(R_1)	AA + 0.005 in-degree + 0.5 in-links + 0.8 out-links	29.90	123.74	143.06
(R_2)	R_1 + 0.25 journal	29.42	121.12	143.38
(R_3)	R_2 + 0.004 title-characters	29.14	119.58	140.30
	R_3 + 1.3 title-word-length	29.21	118.94	139.75
(R_4)	R_3 + 0.9 title-word-length + 0.1 (year − 2000)	29.17	118.81	138.69
	R_4 + 0.7 cluster-median + 0.35 clus.-centroid-distance	28.81	**117.23**	**137.81**
Our submission on KDD Cup 2003:				
	SVM C parameter = 0.7, AA + 0.006 in-degree + 0.7 in-links + 0.85 out-links + 0.35 journal + 0.006 title-characters + 0.3 cluster-average	31.80	118.89	141.60
Second best entry on KDD Cup 2003				146.34
Third best entry on KDD Cup 2003				158.39

Table 3: A comparison of the performance of increasingly complex representations. Apart from errors measured during cross-validation, the table also shows the error achieved by each representation if the full training set is used for training and testing is done on the same 150 papers from the test period that have been used for evaluation on the KDD Cup. Our submission to the KDD Cup is worse than some of the models shown in the upper part of the table because we only thought of some of the attributes after the KDD Cup deadline.

number of downloads.

We will introduce one feature for each journal. If a paper appeared in a joural, a corresponding feature will be nonzero. The journal attribute, when added to the best representation from the previous section, makes a non-negligible contribution: the error decreases from 123.72 to 121.08.

For each journal we can compute the average number of downloads over all the training papers which belong to this journal. We can then use this as an additional feature in each document. We can also use the standard deviation or the median instead of the average. Experiments show that these new features aren't useful — they lead to ovefitting.

Title length. We observed that many of the highly downloaded papers tend to have relatively short titles. Perhaps there is something about these plain, self-confident titles that attracts the readers. We tried using the title length as a new attribute. With suitable weighting, this reduced the error from 121.08 to 119.59. We also experimented with the length of the abstract, the number of authors and the year of publication as additional attributes. None of these were really useful.

Clustering. We tried clustering the papers and using information about cluster membership and cluster statistics as additional features. Binary cluster membership features caused overfitting. More useful were the median number of downloads in the cluster and the distance of the paper from the centroid of its cluster. These reduced the error to 117.23.

6. CONCLUSIONS AND FUTURE WORK

A look back. Table 3 shows the performance of models obtained from richer and richer representations, from the basic "author + abstract" representation to the best representation found.

We used cross-validation all the time, but with our relentless tuning of the features and their weights, we could still overfit the training data. Thus it is nice to see that we do not overfit. The model that has been found best during cross-validation is also the best on the previously unseen test data. Interestingly, if we had used the predictions of the trivial model as a KDD Cup submission, we would still achieve the ninth or even eighth place (of 18 teams).

Conclusions. The download estimation task was not a very simple one. Although our predictions were the most accurate among the 18 participants, the average error is still quite large. The average number of downloads is approx. 454.5, and the average error of our predictions is 141.5. We are facing a sort of law of diminishing returns: more and more new features have to be tried before a good one is found, and these decreases in error are getting smaller and smaller; besides, the features need to be carefully weighted. (Note that we tried several other attributes besides those presented here. Additionally, we tried a few other learning algorithms such as nearest neighbors and bagging.)

Future work. We could extend our approach by using n-grams (groups of up to n adjacent words) from the abstract and address; cleaning up the institution address data (currently extracted automatically), which has so far proven unexpectedly useless; and using the length of the paper as a new attribute.

However, all of these extensions are still based on the same underlying thinking. They are only small steps from the representations considered in this paper. To achieve larger improvements, it would be necessary to introduce features that say something genuinely new about the documents.

We could try to investigate what influences a reader's decision to download a paper. There are probably questions such as "Is this an interesting paper?" and "Will this paper help me with my own research? Is it something I need to know to keep up with my field?" The most highly downloaded papers are not necessarily the most widely read, important or influential but are also in some sense exceptional, and there is something about them that excites the curiosity of readers. They are in a sense outliers.

We would like to encourage everyone interested in our work to visit our web site at `http://ai.ijs.si/kddcup03`. Here you can find more about the experiments, a detailed technical report [1] and already extracted features and data we used for our experiments.

Acknowledgments. Thanks to Marko Grobelnik and Dunja Mladenić for their encouragement and support.

7. REFERENCES

[1] J. Brank, J. Leskovec: *The Download Estimation Task on KDD Cup 2003.* Tech. Rept., Jožef Stefan Institute, 2003. `http://ai.ijs.si/kddcup03/`

[2] C.-C. Chang, C.-J. Lin: *libSVM: A library for support vector machines* (version 2.3). Dept. of Comp. Science and Information Engineering, Nat'l Taiwan University, April 2001.

[3] J. M. Kleinberg: *Authoritative sources in a hyperlinked environment.* J. of the ACM, 46(5):604–632, September 1999.

[4] L. Page, S. Brin, R. Motwani, T. Winograd: *The PageRank citation ranking: Bringing order to the web.* Report SIDL-WP-1999-0120, Stanford University, Jan. 29, 1998.

[5] A. J. Smola, B. Schölkopf: *A tutorial on support vector regression.* NeuroCOLT2 Rept. NC2-TR-1998-030, Oct. 1998.