MACHINE LEARNING ON SETS OF DOCUMENTS CONNECTED IN GRAPHS

Janez Brank and *Jure Leskovec*
Department of Knowledge Technologies
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773778; fax: +386 1 4251038
e-mail: janez.brank@ijs.si, jure.leskovec@ijs.si

ABSTRACT

This paper deals with the problem of machine learning on sets of documents connected into graphs. Our strategy is to represent each document by a diverse set of heterogeneous attributes, including traditional binary and categorical attributes, textual attributes, and attributes derived from the graphs. We present experiments on two datasets, showing the usefulness of graph-based attributes and the importance of weighting the different attributes suitably before learning. On the download estimation task, the approach presented here achieved the best results on the KDD Cup 2003 challenge.

1 INTRODUCTION

Traditionally, machine learning dealt with datasets where each instance is represented by a moderate set of numerical and/or categorical attributes. Recently, however, using machine learning techniques on other types of data has been attracting increasing amounts of attention. Among other things, machine learning has been applied to textual data, chiefly by the text categorization community. The growth of the World Wide Web and other social networks has also increased the interest in the analysis of data organized in graphs (in the graph-theoretic sense).

In this paper we present two case studies of datasets that lie at the intersection of these various interests. The datasets consist of documents that contain text, are additionally described with traditional attribute data, and are also connected in one or more graphs. The challenge here is to improve the success of machine learning techniques on these datasets by combining these different types of attributes.

2 THE KDD CUP DOWNLOAD ESTIMATION TASK

KDD Cup is an annual data mining challenge, organized in connection with the ACM SIGKDD conference. In 2003, one of the challenges involved estimating the interest generated by scientific papers, as measured by the number of times that a paper was downloaded from a web server.

The dataset consisted of 29014 papers from the "high energy physics — theory" area of arXiv.org, a well-known repository of preprints. Apart from the full text of all papers (in TeX format), some more structured metadata was also available for each paper: title, author names, journal in which the paper was subsequently published (if any), etc. Additionally, the citation graph for this set of papers was provided. The graph has one vertex for each paper, and an edge whenever one paper cites another. (Citations pointing outside the dataset, and those pointing inside from papers outside the dataset, are not recorded in this graph.)

For papers that were published in the arXiv during a "training period" of 6 months (1566 papers), we were also provided with the number of times that each paper was downloaded from arXiv's web server. Only downloads in the first 60 days since publication were included in these counts. In addition to the number of downloads, timestamps of individual downloads were also provided.

The task is to predict the number of downloads (in the first 60 days since the publication of each paper) for the papers from a "test period", covering 3 months. The measure used to evaluate the predictions on the KDD Cup was the sum of absolute values of prediction errors; however, only the 50 most frequently downloaded papers from each month were taken into account (other papers and the predictions for them were ignored by this evaluation measure). Note that this is only approx. 20% of the papers (there are around 250 per month).

It could be argued that this task definition is problematic from certain respects, for example in its focus on only the most frequently downloaded papers, and in the fact that the data available both to the learner and the predictor include a lot of material that is useful for our download estimation task but is not available when the paper is initially published in the archive (e.g. citations of the paper by other papers, or the name of journal in which it is eventually published). Thus, the task as defined here is somewhat unrealistic. However, the advantage of accepting this task definition is that we can compare our results with those of the other KDD Cup 2003 participants.

3 OUR APPROACH TO THE DOWNLOAD ESTIMATION PROBLEM

According to the problem specification, only the (approximately) 20% most frequently downloaded papers will be used to evaluate our predictions. Of course, since we do

now know in advance which papers these will be, we must submit predictions for all papers from the test period, but the predictions on papers other than the top 20% will be ignored. This means that it does not matter how wrong our predictions on these other papers are; it makes sense to focus solely on the top 20% of papers.

One way to achieve this is to have our model treat each paper as if it belonged to the top 20%; if it doesn't really belong to the top 20%, our prediction error may be greater, but this will not affect the evaluation of our predictions. To ensure that our model treats each paper as if it belonged to the top 20%, we will use only the most frequent papers to train the model. It would be natural to use the top 20% of the papers from the training period, but experiments have shown that using slightly more training papers gives better results; thus we use 30% of the training set.

We will represent each paper by a vector containing features from various sources (see the next section); then we will use support vector regression [1] to train a linear regression model for predicting the number of downloads. (We did not experiment with nonlinear SVM, mostly because experience in text categorization shows that nonlinear SVM performs only marginally better than linear SVM. Since our task also involves textual and high-dimensional data, we conjecture that the benefits of using nonlinear SVM may similarly be very small.)

Most of our experiments focus on testing different combinations of features and different weights that can be applied to the various features before training. We use 10-fold cross-validation (CV) on papers from the training period to compare these different representations. As mentioned in the previous paragraph, only the top 30% of the training papers are used to train the SVM regression model; during evaluation on the validation set, only the top 20% of the papers from the validation set are used. This ensures that the evaluation conditions are similar to what would later actually be used to evaluate our test period predictions.

4 FEATURES USED IN THE DOWNLOAD ESTIMATION PROBLEM

4.1 Author, Abstract, Address

For each paper, we know its abstract, which is typically one paragraph of text. We can use this to represent a paper using the well-known "bag of words" paradigm; that is, we introduce an attribute for each word that occurs in any of the abstracts; a paper is then represented as a vector of TF-IDF weights, in which each component gives the number of occurrences (or "term frequency", TF) of the corresponding word in the abstract, multiplied by a value that is intended to reduce the influence of very common words ("inverse document frequency", IDF); finally this vector is normalized to unit Euclidean norm, to remove the influence of the document length.

Similarly, we can represent a document using the authorship information. Although this has been originally provided in a human-friendly rather than machine-readable form, the data can be cleaned relatively easily and we can introduce one attribute for each author; then each document is represented by a vector of binary values in which each component tells if a particular person is one of the authors of this paper or not. It can also be beneficial to normalize these vectors before training.

Another way in which we tried to use the textual information that has been provided about the papers was to try extracting the addresses of the institutions with which the authors are affiliated. We hoped that reputable institutions employ better-known authors whose papers attract more downloads, and that therefore institution information would be useful in estimating the number of downloads. Since these institution names and addresses were not available to us in the files containing the abstracts and other metadata, we had to extract them directly from the TeX source of the papers (the extraction process relies on heuristics and is somewhat inaccurate, often extracting more text than would be necessary).

Of course, one hopes that different representations will be useful on different papers and that therefore a combined representation might be more successful than any of the individual representations. Thus, if a document is represented by the vector $(x_1, \ldots, x_r)$ under one representation and by $(y_1, \ldots, y_s)$ under another, a combined representation may have the form $(\alpha x_1, \ldots, \alpha x_r, \beta y_1, \ldots, \beta y_s)$, for some suitable weights α and β that can be used to balance the influence of different groups of attributes.

Average prediction error (cross-validation)		
Representation	*training set*	*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	**135.70**
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 authors, abstracts, and institution addresses.

Noting that the address information did not turn out to be useful (Table 1), we adopt the "author + abstract" representation as the baseline for further experiments.

4.2 Using the Citation Graph

Although we are interested in the downloads that occur within the first 60 days since the publications of a paper in the arXiv, and in this time a paper typically does not yet have any citations, the citation graph may nevertheless be useful since, in the long term, important and influential papers obtain more citation, while in the short term, such

papers might have been recognized by readers immediately upon publication, based on the title, abstract and authors, and might have therefore been downloaded more often. Thus, although the relationship between citations and the download count as defined in our task is not causal, there may nevertheless be a correlation that could be used to improve the download estimation.

We use the citation graph as a source of attributes in two ways. One is to compute numeric attributes describing the position of each paper as a node within the graph; in-degree and out-degree are obvious candidates, but we also experimented with hub and authority weights (based on Kleinberg's HITS algorithm [2], which was originally designed to assess the importance of web pages based on the structure of the graph of hyperlinks between the pages), and with PageRank [3], based on ideas similar to HITS). These experiments show that in-degree, authority weight and PageRank are helpful for download estimation, but since they are closely correlated, using two or three of them at the same time does not improve the results further. Similarly, out-degree and hub weight are closely correlated, but they are not useful for download estimation; this is not surprising, as a paper is not likely to be interesting or important merely because it cites interesting and important papers (everybody cites those, after all).

The other way of using the graph is to introduce one binary attribute in_p for each paper p. In the vector describing a paper q, let the component $q.in_p$ equal 1 if p cites q, and 0 otherwise. Analogously, we can define attributes that tell if q points to a certain other paper or not. These attributes are referred to as "in-links" and "out-links" respectively.

Representation	*Average test error during cross-validation*
AA (= Abstract + Author, from previous section)	135.70
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 + 0.9 out-links	**123.72**

Table 2. The performance of representations with graph-based features.

It is worth noting that it is important to multiply these attributes with a suitable weight before training. For example, the average in-degree in our citation graph is approximately 10, while the normalized TF-IDF vectors from the previous section have all components between 0 and 1 (mostly closer to 0). When these two representations are combined, the in-degree will far outweigh the other attributes unless it is first multiplied by some small weight.

4.3 Miscellaneous Statistics

Journal information. For approximately 72% of papers, we know the jurnal where the paper has eventually been published. We can thus introduce a binary attribute for each journal, indicating whether a particular paper has been published there or not (additionally, there is one attribute for papers with no journal information).

We also tried computing numeric attributes based on the journal information. For example, we can compute the average number of downloads over all papers from a journal, and then introduce an attribute that gives, for each paper p, the average number of downloads over all training papers published in the same journal as p. This might be promising as different journal do have different average download counts; however, variance within each journal is typically larger than these differences. It turned out that attributes of this type lead to overfitting and were not useful for our task.

Title length. We observed that many of the most frequently downloaded papers have relatively short titles. Thus we used the number of characters and the number of words in the title as attributes. Similarly, we experimented with attributes giving the number of characters and number of words in the abstract, the number of authors, the year of publication, and the average length of title words. Most of these attributes were not useful, except for the title length in characters.

Clustering. We clustered the papers into 26 clusters using recursive 2-means. We can introduce a binary attribute for each cluster, indicating whether a particular paper is a member of that cluster. We also tried introducing attributes such as the average number of downloads over all training papers from the same cluster as the paper under consideration. Another interesting attribute is the distance of the paper from the centroid of its cluster; this attribute was moderately helpful, suggesting that the number of downloads is slightly higher if the paper is nearer the centroid (perhaps papers far from their cluster's centroid lack a clear focus and a distinct audience?).

	Average prediction error	
Representation	*cross-validation on training set*	*true test set*
Triv. model (predict training set median)	152.26	**181.11**
Author + Abstract (AA)	135.86	**155.38**
AA + 0.004 in-degree	127.69	146.77
AA + 0.005 in-degree + 0.5 in-links + 0.8 out-links	123.72	143.06
previous + 0.25 journal	121.12	143.38
previous + 0.004 title-characters (*)	119.58	140.30
(*) + 1.3 title-word-length	118.94	139.75
(*) + 0.9 title-word-lenth + 0.1 (year – 2000) (**)	118.81	138.69
(**)+ 0.7 cluster-median + 0.35 clus. centroid dist.	117.23	**137.81**
Our submission on KDD Cup 2003	118.89	141.60
Second best entry on KDD Cup 2003		146.34
Third best entry on KDD Cup 2003		158.39

Table 3. Performance of different representations on the download estimation problem. "True test set" refers to the 150 papers that were actually used for evaluation by the KDD Cup 2003 organizers.

Our KDD Cup submission is slightly worse than the best model reported above, because we introduced some features only after the KDD Cup deadline. The high test set errors are due to a single outlier in the true test set.

5 PREDICTING THE INCOME OF FILMS

5.1 Task Description

To test the methodology described above on some other dataset besides the KDD Cup 2003 download estimation task, we defined another similar dataset based on the Internet Movie Database (IMDB). The database currently contains information about more than 300000 movies, including data such as titles, actors, directors, genres, plot summaries, taglines (short slogans used to advertise the movie), etc. However, it has to be emphasized that not all of these types of data are available for all the movies. Additionally, several relations are defined on this set of movies, which implicitly define graphs similar to the citation graph we've seen in the download estimation tasks. Examples of these relations include "is a sequel of", "is a spoof of", "is a remake of", and the most populous and interesting "references" (described by IMDB as "dialogue or situations in the former [movie] reference or pay homage to the latter"). Some of the data in IMDB also refers to the business and commercial aspects of films; in particular, the gross income from the screenings of the film in the U.S. is known for 3163 movies. We decided to try predicting this gross income, as it would form a problem similar to the download estimation task (in both cases the predicted value is essentially a kind of short-term popularity that can often be closely related to marketability and hype).

The structure of our experiments is similar to that of the download estimation experiments. We focused on films from the period 1980–2003 (2588 films), using the period 1980–1995 for training and 1996–2003 for testing.

5.2 Attributes used

Due to space considerations, we cannot present these attributes in as much detail as in the download estimation task. We used binary attributes to represent individual actors and directors; each attribute specifies if that person participated in the making of a movie or not. (We also tried introducing similar attributes for producers, scriptwriters and special-fx people, but they were not useful.) Year of production is a problematic attribute in our task, because the test period covers different years than the training period; thus, whatever we could learn about individual years from the training set will be useless on the test set.

Most movies also mention one or two genres to which they belong; thus, binary attributes for individual genres can be introduced. Similar to the case of journals for download estimation, there are considerable differences in average income between genres but also even greater variance within each genre; thus, these attributes are not useful.

We also worked with several groups of textual attributes, all based on the "bag of words" approach: title words, tagline, plot summary, user comments. Note that treating user comments as bags of words are quite problematic, as many comments praise some aspects of a film but criticize others, and in the resulting bag both "positive" and "negative" words will be mixed indiscriminately.

The relations between films, mentioned in the previous subsection, can be seen as forming 16 directed graphs. Attributes that can be derived from these graphs again include in- and out-degrees, hub and authority values, and binary attributes indicating who are the neighbors of a vertex.

The results of these experiments are shown in table 4.

	Average prediction error [M$]	
Representation	*training set*	*test set*
Triv. model (predict training set median)	17.67	**17.68**
Actors	13.41	16.37
Actors + 9 directors	10.47	15.46
Actors + 9 directors + 0.5 country (*)	10.12	15.31
(*) + 1.5 tagline (**)	10.06	15.12
(**) + 0.14 out-degrees	8.20	13.30
(**) + 0.05 hub-weights	8.25	13.48
(**) + 0.14 out-degrees + 3 out-links	7.30	13.03
(**) + 0.05 hub-weights + 3 out-degrees	7.38	**12.94**
Automated tuning of all parameters	8.12	**12.51**

Table 4. Performance of different representations on the IMDB problem.

6 CONCLUSIONS AND FUTURE WORK

We have presented two case studies exploring the issues of combining binary, categorical, textual and graph-based attributes for regression problems on datasets consisting of sets of documents connected in graphs. In both cases we saw that although the problems are hard and even the best models found so far still have relatively high prediction errors, we have nevertheless achieved considerable improvements relative to the naïve baseline models with constant predictions. Much of this is due to the combined effects of several small improvements in performance that have been contributed by various individual attributes and groups of related attributes. We saw that appropriate weighting of attributes can be very important to allow the representation to be used to its best potential. Graph-based attributes have been found to be very useful on both datasets.

This work could be extended in many interesting ways. The download estimation task could be made more realistic by not allowing the predictor to use any information that is not available immediately upon publication of a paper. Various other attributes and external sources of information could be considered (for example: are downloads related to the dates of conferences, days of week, seasonal patterns, etc.?). Both the download estimation task and the movie income prediction task open up questions related to modeling popularity and modeling user decisions (what is the reader thinking when deciding whether to download a

paper or not?). Normalization and standardization of attributes could be investigated as potentially useful alternatives to the weighting as used in our experiments. Different schemes for choosing the weights of attributes or attribute groups could be considered.

References

[1] A. Smola, B. Schölkopf. A tutorial on support vector regression. NeuroCOLT Rept. NC2-TR-1998-030, October 1998.

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

[3] L. Page, S. Brin, R. Motwani, T. Winograd. The PageRank citation ranking: bringing order to the web. Tech. Rept. SIDL-WP-1999-020, Stanford University, January 1998.