Citation Prediction Using Time Series Approach KDD Cup 2003 (Task 1)

Manjunatha J N[1] Sivaramakrishnan K R[2]
Raghavendra Kumar Pandey[3] M Narasimha Murthy[4]
[123]Department of Electrical Engineering, [4]Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India.
{[1]manju,[2]sivaram,[3]rkp}@ee.iisc.ernet.in, [4]mnm@csa.iisc.ernet.in

ABSTRACT

In this article we describe our experiences in building the winning system for KDD Cup 2003, Task 1. This year's competition was based on a very large archive of research papers that provides an unusually comprehensive snapshot of a particular social network in action; in addition to the full text of research papers, it includes both explicit citation structure and partial data on the downloading of papers by users. It provides a framework for testing general network and usage mining techniques, which can be explored via four varied and interesting tasks. Each task is a separate competition with its own specific goal. In task 1 the goal is to predict the change in number of citations to each paper in the archive over time.

The contest was very challenging because the given data was not in a format suitable for conventional data mining techniques. So we had to do a considerable amount of data processing. Also there were different sources of data like tex files, citation graph, slac-date database. So we had to make a decision about which sources to use and how much to use.

Keywords

Citation prediction, KNNC, Network Usage Mining.

[1] INTRODUCTION

KDD cup has been a testing ground for application of mining tools to solve contemporary learning problems using data from past experiences. This year's contest was to apply mining to network management problem given a snapshot of a particular network.

The data consisted of

- The LaTeX source of all papers in the hep-th portion of the arXiv through May 1, 2003 (SOURCE).

- The abstracts for all of the hep-th papers in the arXiv. Each abstract contained arXiv submission date, revised date(s), title, authors, abstract (ABSTRACT).

- The SLAC/SPIRES dates for all hep-th papers (SLAC).

- The complete citation graph for the hep-th papers, obtained from SLAC/SPIRES. Each node was labeled by its unique ID (CITATION).

In our approach was to convert the given data into time series and to use it conventional regression techniques to predict the change in number of citations over time.

Outline In section 2 we describe our approach briefly, in section 3 we give the complete algorithm and in section 4 we discuss implementation and results.

[2] TIME SERIES BASED APPROACH

Our approach has these three major steps: Conversion of given data into time series data, Converting time series data into a form suitable for regression and Prediction using regression.

In converting the given data into time series, we used the SLAC dates database and the citation graph. For each paper a time series over the quarters was constructed. This time series was used in the second step to get the patterns. These patterns were used to train a system that predicts the change in number of citations.

[3] DESCRIPTION OF THE APPROACH

1. Preprocessing of given data

Here we convert the given data into time series format.

We use SLAC file that contains, for each paper the estimated date of its actual publication in the sorted order of its id. Then we use the CITATION file that contains, the list of all references in the form of <citing paper id , cited paper id> format to find the number of citations made to each paper in each month which is generated as a time series.

2. Data Preparation for Regression

Then we created a table CiteTab containing number of citations for each quarter, for each paper. The quarters overlap each other. For example Jan-Mar is a quarter, also Feb-Apr and Mar-May. This gives an increased amount of training data. This was a very important factor which affected our results as overlapping quarters means more data and more data normally results in better prediction. This has the added advantage of smoothing out any noise in the time series.

After this we created another table CiteDif containing difference in number of citations over last quarter, for each quarter and paper. Note that here while calculating difference we used non-overlapping quarters. For example, to get the change in number of citations in Jun-Aug we use {citations in Jun-Aug – citations in Mar-May}.

Using CiteDif we created input for regression. That is in the form of a tuple and it's functional value. For each paper, for each quarter, if the number of citations in the previous quarter is more

than some threshold T using CiteTab, we created a tuple containing previous D entries in difference table (CiteDif) and current entry as it's functional value. Add this tuple to set KnncTrainTup which is later used as input for learning the parameters of the regression function. Here we used threshold T to screen out entries which would not have a considerable change in number of citations. We also create test tuples using last D columns of CiteDif for each paper in KnncTestTup.

For i in Paper id

For j in Quarter

If($CiteTab_{i,j}$>T)

Add($CiteDif_{i,j-D}$,....$CiteDif_{i,j-1}$,$CiteDif_{i,j}$) to KnncTrainTup

end If

end Loop

Add($CiteDif_{i,n-D}$,...$CiteDif_{i,n-1}$,$CiteDif_{i,n}$) to KnncTestTup

end Loop

Note that T and D are decision parameters.

3. Prediction using KNNC

For each test tuple we found K nearest neighbors and we predicted their mean functional value as answer.

For each x in KnncTestTup

Find y[1..K] from KnncTrainTup, with functional value of z[1..K], which are K nearest neighbors of x

Assign functional value of x as (z[1]+...+z[K])/K

end loop

Note that here K is a decision parameter.

[4] IMPLEMENTATION AND RESULTS

We implemented this algorithm in C in Linux platform.

For doing regression we tried other options like Support Vector Regression, but we found out that, it did not result in considerable decrease in error.

[5] ACKNOWLEDGMENTS

Our sincere thanks to Supercomputer Education and Research Center(SERC) and Laboratory for Internet Technologies and E-Commerce(LITEC) for providing us the computational resources during the work.

[6] REFERENCES

Duda, Hart and Stork, Pattern Classification (2nd Edition). John Wiley & Sons.

Ronan Collobert and Samy Bengio. SVMTorch: Support Vector Machines for Large-Scale Regression Problems, Journal of Machine Learning Research, Volume 1, 143-160, 2001.

[7] About the authors:

M.Narasimha Murty is a professor at the Indian Institute of Science. His research interests are in pattern clustering and data mining.

J N Manjunatha is a masters student in System Science and automation at Electrical Engineering Department, Indian Institute of Science. His research interests are in data mining, temporal mining and machine learning.

K R Sivaramakrishnan is a masters student in System Science and automation at Electrical Engineering Department, Indian Institute of Science. His research interests are in natural language processing and machine learning.

Raghavendra Kumar Pandey is a masters student in System Science and automation at Electrical Engineering Department, Indian Institute of Science. His research interests are in stochastic modeling and data mining.