Time Series Forecasting through Clustering - A Case Study

Vipul Kedia*
International Institute of Information Technology
Hyderabad, India
vipul@students.iiit.net

Vamsidhar Thummala†
International Institute of Information Technology
Hyderabad, India
vamsidhar@students.iiit.net

Kamalakar Karlapalem
International Institute of Information Technology
Hyderabad, India
kamal@iiit.net

ABSTRACT

Time series forecasting plays an important role in many day to day applications, and is often used as a tool for planning in many areas. In this paper, we propose a generic methodology for time series forecasting. We use a subset of the dataset to build up the system model by compressing the information through clustering and coming up with inherent patterns in the data. These patterns are represented as curves that any time series from the given set is expected to follow. It then facilitates the forecasting through linear regression by matching to the closest pattern to each time-series that has to be predicted. We applied this approach on Kddcup 2003 dataset for predicting the citations of the research papers and found the results to be on par with best results.

1. INTRODUCTION

Forecasting is important for taking decisions about future events. Typically, past information and patterns are exploited to predict the future. In case of time series data, the past information is a series of stamped data sets and forecasting involves future generation of the datasets [18, 3, 10]. In this paper, we present a methodology for forecasting time series through clustering.

1.1 Time Series Forecasting

Time series forecasting, or time series prediction, takes an existing series of data x_{i-n}, ..., x_{i-2}, x_{i-1}, x_i and forecasts x_{i+1}, x_{i+2}, ... the data values. The goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately. Examples of data series include financial data series like stocks, physically observed data series like sunspots, weather and mathematical data series like fibonacci sequence, integrals of differential equations etc.

Here, we focused on predicting the citations of scientific research papers with a technique called time series forecasting through clustering. When research paper X refers research paper Y, then Y is said to be cited by X and the prediction of citations is the number of such references the paper Y gets in the given amount of time. The popularity and the importance of a research paper is based on its number of citations.

In the next subsections, the major difficulties inherent in predicting the citations of research papers and the importance of forecasting the citations are presented. Section 2 presents the related work. Section 3 gives the general methodology and the model of the predicting system. Section 4 presents the case study of our approach on Kddcup 2003 task I dataset. Section 5 discuss about the evaluation of the system.

1.2 Difficulties in forecasting

There are certain difficulties encountered while forecasting the citations for the given data. The first difficulty is the availability of *limited quantity of the data* for the recently published research papers. A research paper published recently might have very less number of citations and prediction of future citations becomes difficult if the forecasting time is anywhere more than 40% of the lifetime of the research paper.

A second difficulty is presence of *noise*. Noisy data can be of two types *i*) erroneous data and *ii*) and components that obsure the underlying form of the data series. The examples of erroneous data are measurement errors and a change in measurement methods or metrics. In this paper, we do not concern about erroneous data points. An example of a component that obscures the underlying form of the data series is an additive high-frequency component due to which there is a sudden spike observed in the curve which can mislead prediction.

A third difficulty is *non-stationarity*, data that do not have the same statistical properties (e.g., mean and variance) at each point in time. A simple example of a nonstationary series is the fibonacci sequence: at every step the sequence takes on a new, higher mean value.

Inorder to eliminate the presence of noise and nonstationarity and to capture the overall behaviour of the curve, the technique we used is to take the cumulative data values and plotted them against time. Cumulative data captures the

*Currently studying at IIM, Ahmedabad, India.

†Currently working at Convergys, Hyderabad, India.

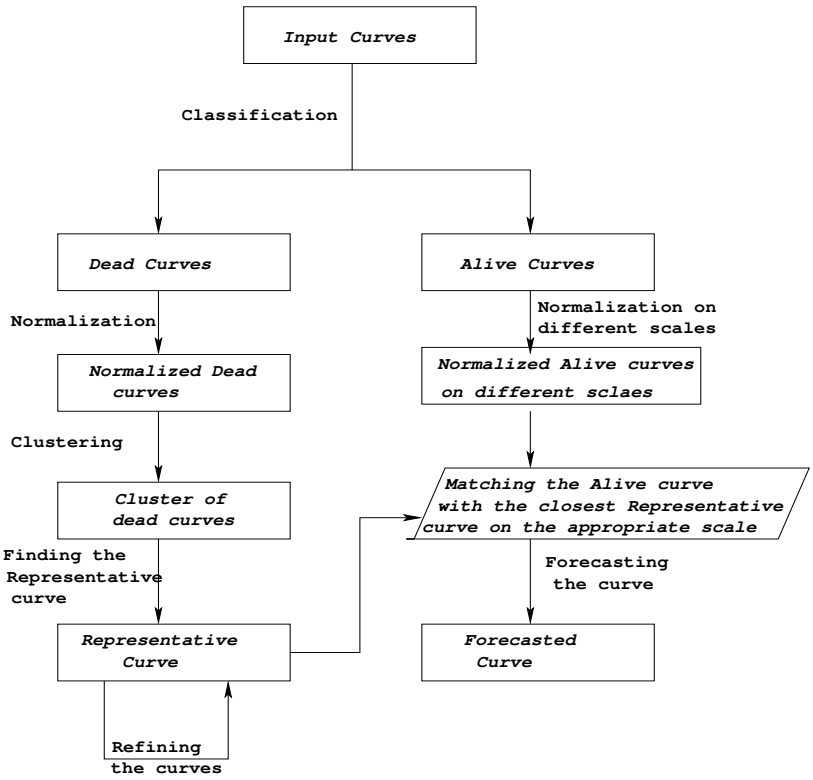

Figure 1: Flow chart of methodology

overall pattern of the curve since it is a monotonically increasing one. The slope of the curve gives an idea about how the data is varying with time. The data series ...., $x_{i-4}, x_{i-3}, x_{i-2}, x_i$ becomes
..., $\sum_{i=1}^{k-4} x_{i-4}, \sum_{i=1}^{k-3} x_{i-3}, \sum_{i=1}^{k-2} x_{i-2}, \sum_{i=1}^{k} x.$

A fourth difficulty is *selecting the forecasting technique.* From statistics to artificial intelligence, there are myriad choices of techniques. One of the simplest techniques is to search a data series for similar past events and use the matches to make a forecast. One of the most complex techniques is to train a model on the series and use the model to make a forecast. We used the first techinque with modifications and the results show that our technique is scalable and can perform better.

1.3 Importance

With time series forcasting, there are numerous advantages like stock prediction in financial markets, prediction of weather reports and prediction of the growth of mobile market, prediction of growth of traffic etc. Forecasting the number of citations, the research paper receives also plays a vital role in deciding the importance of the topic. If a group of research papers on the same topic are going to receive high citations on the average, then the topic becomes hot topic for the future. A new researcher can choose that topic to work on. Also, one can make use of this feature in designing a ranking algorithm to search the repostitory of research papers.

2. RELATED WORK

Time series forecasting can be done by either linear prediction or non-linear prediction. In linear prediction, there exist a large number of linear models, such as Auto Regression, Moving Average, Auto Regression Integrated Moving Average etc. Auto Regression Moving Average is a popular approach and is widely used in all time series analyis and discrete signal processing [3, 4, 16, 15]. But linear regression models fail to forecast accurately if there is non-linearity in the data. In non-linear prediction, quite widely popular approach is the use of Artificial Neural Networks[19, 17, 11, 21, 20, 9, 13]. The main disadvantage with the Neural Networks is it is complex to built and takes huge training time. There has been some work done by Das et.al. [6] and [5, 1] on finding rules/pattern from the time series data by forming subsequences of data and clustering them and finding rules based on the sequence of clusters. Forecasting time-series by Kohonen classification is done by Lendasse et.al.[12]. Other work related to time series on matching and similarity serach is done by Faloutsos et.al.[8, 2, 7, 14].

3. METHODOLOGY

Our basic hypothesis is that in any given set of related time series data, there are distinct patterns that are followed by individual time series. A task of predicting the future direction of such data would involve:
a) Identifying the distinct patterns followed by the members of the set
b) For each times series, identify the particular pattern it is

following so far and predict based on that pattern.

Our methodology works best for a set of time series on which we can apply the concept of a lifecycle. By lifecycle we mean that each time series has a region of birth, growth, maturity and decline followed by death. Birth refers to the initial periods of the time series and death refers to that point in time after which the data values for each period are close to zero. These kinds of time series are fairly common in the world and examples would include monthly sales of a product, number of citations a paper receives, the sales of tickets for a particular performance and so on. Thus any given set of such a time series will consist of series which have reached their death as well as others which are in intermediate stages of their lifecycle. We use the so called "*dead*" series for identifying the patterns in the data.

In cases of time series where the concept of lifecycle does not apply, i.e. series which go on indefinitely, we can use the members with that have been active for the longest time to represent the dead series. Examples of such series are monthly telephone bills, credit card spending etc. However, our technique has not been tested on such a series.

We divide the dataset into two parts: the dead time series and the alive time series. They are classified based on the values in the recent time periods. Based on the nature of the dataset, we can come up with appropriate rules to distinguish the dead series. A typical rule may say that for a series to be classified as dead, it should have less than a threshold value in the last 'n' time periods. For example, a series is dead if it has a value less than 2 in the last 4 time periods. Another possible rule is that a series is dead if the cumulative value in the last 'n' time periods is less than x% of its cumulative value so far. An example of such a rule is that a series is dead if in the last 4 time periods it has received less than 5% of its total cumulative value.

Different time series data will have different range of values in each time period. Also in many cases, the time series data may not follow a regular pattern and can have a zigzag curve nature. For example, one time series may have values from 20-40 in each period whereas another may have values from 0-4 in each period. To get rid of these differences and intermidate spikes or impluses, we use a cumulative value for each period. This helps us in smoothing of curves, since the cumulative curve is monotonically increasing one. So a time series 2 2 4 4 2 6 will be represented as 2 4 8 12 14 20. The next step is to normalize the dead series on a scale of 0 to 1. So the above time series becomes 0.1 0.2 0.4 0.6 0.7 1.0. In this way all the dead time series are now normalized n a scale of 0 to 1.

We now use clustering to identify the patterns in these series. We come up with a set of representative patterns based on manual observation. We then use distance based clustering to find how many series follow each pattern identified by us. A permissible deviation is defined from each point on the pattern and all the series which fall consistently in the permissible neighborhood are considered to follow that pattern. At this stage we eliminate the patterns which do not have a minimum number of series following them. We also analyze those time series which were not assigned to any pattern, and if needed, come up with more patterns. The next step is to refine the manual patterns by taking the means of each point of all the time series in a cluster and then coming up with the new pattern. The whole process is repeated till the incremental improvement becomes insignificant. Hence, we get a set of patterns which is followed by the members of the given dataset. Each time series which is alive (i.e. not dead) is expected to be following one of these patterns in its life so far. Our task is now to identify which pattern it is following. This is a bit complicated because we do not know at which stage of its lifecycle a given series is. For example, if we take the time series of monthly product sales, we do not know that so far, what percentage of total sales the product has achieved. To determine this, we can use apriori knowledge (i.e. we know that on average, a product receives x% sales in its first year). In case it is not available, we can assume different values (say 20%, 25%, 30% and so on) and see which value gives us the best match with a pattern. Suppose we assume that the sales of the product is 40%. This means that we will get the monthly cumulative sales for the product as described above, but instead of normalizing it on a scale of 0 to 1, we normalize it on a scale of 0 to 0.4. Now we compare the series to each of the patterns (we take the relevant portion from 0 to 0.4 for each pattern) and see which pattern gives the closest match. Based on that, we predict how the time series will move in future. Refer section 4.10 for more details on prediction process. The flow chart of methodology is shown in Figure 1.

4. APPLICATION TO KDDCUP 2003 DATASET

4.1 Task Description

4.1.1 Input

1. The LaTeX source of about 30000 papers in the through published upto March 1, 2003.
2. The abstracts for all of the papers. For each paper the abstract file contains:
   - submission date
   - title
   - authors
   - abstract
3. The SLAC/SPIRES dates for all papers.
4. The complete citation graph for all papers, obtained from SLAC/SPIRES. Each node will be labeled by its unique ID.

4.1.2 Output

For each paper P in the collection, the predicted difference between

- The number of citations P will receive from papers submitted during the period May 1, 2003 - July 31, 2003, and
- The number of citations P will receive from papers submitted during the period February 1, 2003 - April 30, 2003. (So if there were more citations during the period May 1, 2003 - July 31, 2003, then the prediction should be a positive number)

4.2 Getting the citation graph and the citation lifelines

The first step is to invert the citation graph given in the input to give us the reverse citation graph. The input citation graph is a two column vector with the first column being the paperid and the second column is the citationid (the papers it cites). Paperid is the unique id given to each paper. Each paper P cites a list of other papers P_1, P_2, ... ,P_t. So for paper P the citation graph is

$P \quad P_1$
$P \quad P_2$

$P \quad P_t$

For our purpose we need a graph of each paper and the citations it has received. So the columns of given graph in the input are inverted. The inverse citation graph is a two-column vector in which the first column is the paperid, and the second column has the paperids of the papers which cite it. Each paper P has a list of citations P_1, P_2, .., P_t which means P is cited by paper P_1, paper P_2, .. , paper P_t. If a paper does not receive any citations during its lifeline[1] then there will be no edge for that paper in the graph.

The total numbers of papers are 29,014 of which the 6,318 papers have no citations. The remaining 22,696 papers have at least one citation or more.

4.3 Getting citation lifelines for the papers

The next step is to compute for each paper, the number of citations it received every month from January 1992 to January 2003. This is done to observe how the pattern of citations may vary for a research paper during its lifetime. This is useful for predicting the future citations. For each paper, we are provided with the SLAC dates, or the date on which it entered the SLAC database. We took these dates to be the approximation of the publication date of the paper. So the number of citations a paper receives in a given month is the same as the number of papers which cite the paper and have the SLAC date given month. Let P be the paper and it has total C citations. Let M be the total number of months, which we are considering and m_1, m_2, .., m_n be the first, second and the nth month in that time period. If P receives c_1, c_2, . , c_n citations in month m_1, month m_2, ..., m_n, then $c_1 + c_2 + ... + c_n = C$. For the given data set, we in the obtained a the month wise citation matrix is of size 22,696 x 133 months.

The citation lifeline of a given paper is considered to be a sequence of continuous months during which it received more than 85% of its total citations. We kept the threshold at 85% because it is observed that in many cases, the papers have very sparse citations in the beginning and end of their lifelines. If the sparse months are not neglected, then the lifelines got unnecessarily elongated. Once we find the shortest period with 85% citations, for the given paper, we look at the 6 closest months on either sides of it. We extend the lifelines for the paper only if it received more than 4% citations in the neighboring 6 months. The lifeline for different papers varied from 1 to 120.

[1] we use lifeline and lifetime interchangebly in this paper

Since we have to predict the citations for three month periods, we grouped the citations taking three consecutive months starting from the most recent month i.e. January 2003. Thus we now have 44 citation periods instead of 133 with which we have started. For all future computations the time period used is three months.

4.4 Classification of Dead and Alive papers

Using the trial and error method, the best partition of the set into dead and alive papers is obtained by using the constraint that for a paper to be classified as alive, it should have received more than one citation during the last time period and more than three citations during the last two time periods. Otherwise it is considered to be dead. For the given data set with the above heuristics, we find 2,056 alive papers and 20,640 dead papers. Other heuristics tried gave either too many alive papers or too many dead papers.

4.5 Predicting the future citations

After dividing the dataset into two parts of the alive and the dead papers, we normalized the curves of the dead papers on a scale from 0 to 1. This is done because the number of citations varied from one period to another, and the curve we obtained with the actual value has a zigzag structure. Hence we used the cumulative citations for normalization to get a monotonically increasing curve. Let a paper P with a lifeline of K periods have a total of C citations. The number of citations it receives in each time period is given as $c_1, c_2, c_3, .., c_K$. The sum of all c_i adds up to C. The normalized citations can be represented as $n_1, n_2, n_3, .., n_K$ where

$$n_i = \frac{\sum_{j=1}^{i} c_j}{C}$$
$$\text{and } C = \sum_{i=1}^{k} c_i$$

Each paper has different lengths of lifelines. The number of time periods varies from one to forty. Out of the 20640 dead papers, about 7000 had lifelines of one period, about 1400 papers each for the lifelines of 2-5 and the number decrease from a few hundreds with lifelines of 15 periods down to a few (about 10-20 each) papers with lifelines above 30 periods. Therefore to compare the curves of the papers, there is the need to sample the normalized curves a fixed number of times in order to get the same number of points in each curve. We tried with sampling each curve at 20 and 50 equally spaced points.

The sampling is done as follows. Let there be a paper P with a lifeline of K periods with normalized citation values as described above. Let us want to sample the lifeline at S equispaced points. For this, divide the interval between every two successive time periods into S equal parts to get a total of K x S points. Now start from the first point and take every Kth point to get a total of S sampled points. Thus the lifeline of every paper is now made of K points.

4.6 Clustering of curves of dead papers

We plotted the cumulative data of the dead papers and we observed a leaf patterned graph. Having sampled the

curves of the dead papers, the next step is to cluster the curves based on the similarity in shapes. Each cluster was initiated with a representative curve. The initial set of the representative curves is obtained by manual observation of the distinct curves appearing in the dataset. We started with a set of 12 representative curves i.e. we expected to get 12 clusters of the curves. We choose the 12 representative curves such that these curves represent the entire structure of the data set. We used distance based K-means, single-link based clustering, and Euclidean distance function to calculate distances. Refer Figure 2 for Manual Represenative curves. A permissible neighborhood is defined for each point in the representative curves. We call that neighborhood as 'ϵ'. The corresponding point of any curve is considered to be in the permissible neighborhood of the point on the representative curve if the two points varied from each other in the range of ϵ. The number of such similar points are counted for each curve with the given representative curve. If this number exceeded the threshold, then the given curve was considered to be a member of the cluster. After repeated experiments with different values of ϵ ranging from 0.001 to 0.1, we got the minimum error for the ϵ value 0.05. We took this as ideal value and the threshold for the number of similar points is kept at 15 for curves sampled at 20 points and at 40 for curves sampled at 50 points.

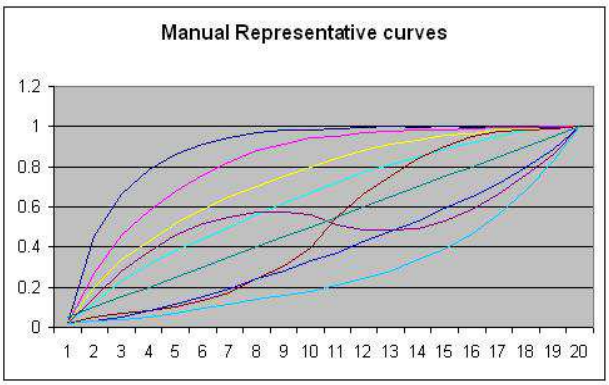

Figure 2: Manual Representative curves

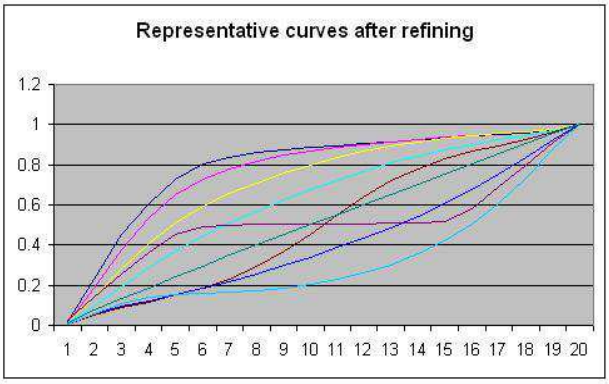

Figure 3: Representative curves after refinement

4.7 Refinement of the representative curves

Each cluster obtained as described above is represented by a curve with respect to which all members satisfied the membership condition. In the first iteration of clustering, this curve is taken to be the manually determined curve. After getting the cluster member, the means of the corresponding points of each of the curves was computed. The curve so obtained is taken to be the new representative curve for the cluster. The clustering process is repeated with the new representative curve. This iteration goes on till the curve obtained by taking the means of the members of a cluster did not vary considerably from the curve used to initiate the cluster. This curve was taken to be the final representative curve. Refer Figure 3 for curves after refinement.

We started with 12 curves. After clustering iterations we find that the clusters of three of these curves have fewer than 20 members. We neglected these curves for the final predictions. The number of the representative curves was kept low because if there are many representative curves, there was a high probability that any given curve of an alive paper would match closely with more than one representative curves. This is because the representative curves were considerably close to each other during the first and the last quarters. Hence unless they is reasonably spaced during the middle region, there ϵ intervals would overlap at almost all points. Thus for our final predictions only the 9 representative curves, obtained as described above are used.

4.8 Matching the curves of the alive papers with the representative curves

Having obtained the representative curves, the next step is to find the closest matching representative curve for each alive paper. The first issue involved is that we did not know at what stage of its lifeline a given alive paper is. For this we considered the possibilities that the paper may be anywhere from 50% to 90% of its total lifeline at the given stage. The motivation for this assumption was that the average lifeline for the dead papers was around 4 time periods, and due to our partitioning constraint described above, each alive paper will have more than two time periods in its lifeline at least. On studying the result we find that we obtained the best results by assuming that the papers were in between 60 to 82 percent of their lifeline. This is so because in the case of 50 to 60 percent, the sample points were not enough in number to accurately match a representative curve. In case of 90 to 100 percent, the matching was accurate but the predictions were unsatisfactory due to the fact that most of the representative curves were either flat so late in their lifeline or has sudden spikes in the number of citations.

The next issue is to find which representative curve it matched most closely. For this purpose, we normalized each active curve on scales from 0.6 to 0.82. We then sampled the normalized curve at 12 to 17 points (corresponding to dead curves with 20 sampled points) or 30 to 41 points (corresponding to dead curves sampled at 50 points) depending on the scale of normalization. We use the same method for sampling and normalization as described in the case of dead papers. We compare the so obtained normalized and sampled alive curves with the representative curves. For the comparison we use only the points corresponding to the sampled points of the alive curves i.e. suppose an alive curve

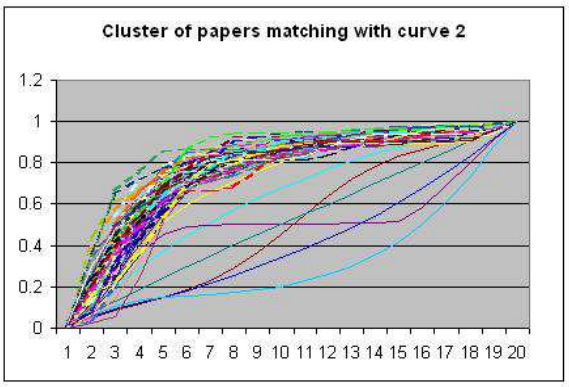

Figure 4: Cluster of dead papers matching with representative curve 2

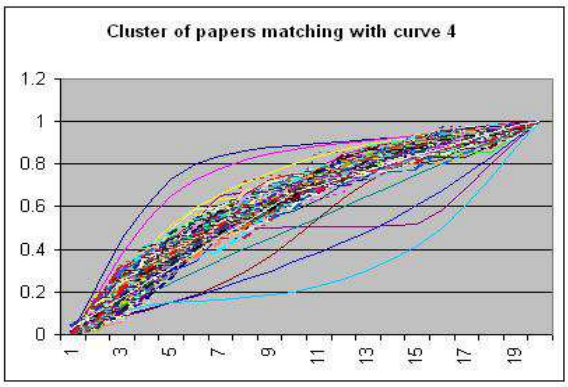

Figure 5: Cluster of dead papers matching with representative curve 4

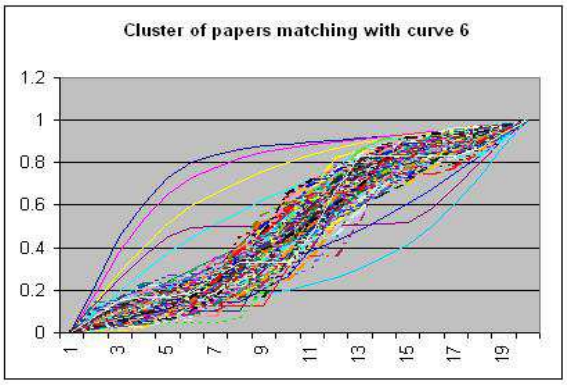

Figure 6: Cluster of dead papers matching with representative curve 6

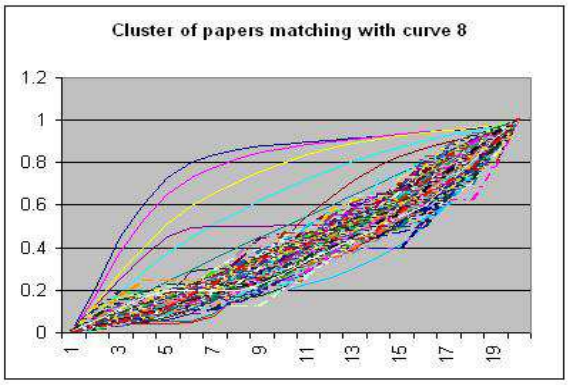

Figure 7: Cluster of dead papers matching with representative curve 8

was normalized on a scale of 0.6 and then sampled at 12 points. We would then compare the curve so obtained with only the first 12 points of each of the representative curves to see which curve it matches the most closely.

Initially, for finding the most similar curve, we used the same method as described in clustering the dead papers i.e. we defined a margin for each point and counted the number of points for each curve which fall within the margin. But, we find that since in the initial period, the representative curves are very close to each other, the alive curves matched multiple representative curves closely. To avoid this, we decided to apply variable margins. So for the first point of each representative curve, we kept the margin at 0.04 and we kept on incrementing this margin by 0.002 till we come to the middle of the curve. Then we again start decreasing the margins by 0.002 for each successive point till we come to the end of the curve. This is because the curves were well spaced in the middle regions and we could afford wider margins. A further improvement was to impose a constraint that for any curve to be considered matching a representative curve, it must match the latest three time periods i.e. if the number of sample points are 12, then the 10th, 11th and 12th points must satisfy the similarity constraint for the corresponding points of any representative curve.

Using the above method, we find the percentage similarity for each alive curve on 12 normalization scales from 0.6 to 0.82. For each such curve we compared it to the corresponding points for each of the representative curves. We calculated the percentage of points which matched in each case. For each alive paper, we noted the representative curve and the normalization scale which gave us the highest percentage similarity. Refer Figures 4, 5, 6, 7, for the dead papers matching with closest representative curves and Figures 8, 9 for the alive papers matching with closest representative curves on different scales. Refer Figures 10, 11, 12, 13 for the sample alive curve matching with matching with closest representative curves on scales 0.58, 0.66, 0.74 and 0.82. Similarly it is done for all the curves.

4.9 Prediction for the next time period

As described above, for each alive paper we get the best match normalization scale and the representative curve it has matched. Let the normalization scale be N ($0.6 <= N <= 0.82$). Let the representative curve C be made of points c_1, c_2, c_3 ..c_M ..c_K where M is the number of sample points of the alive curve and K is the total number of

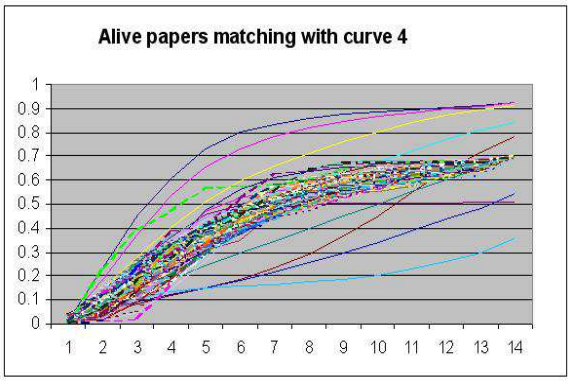

Figure 8: Cluster of alive papers matching with representative curve 4

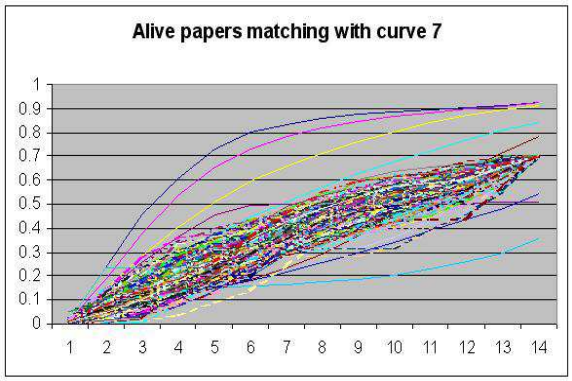

Figure 9: Cluster of alive papers matching with representative curve 7

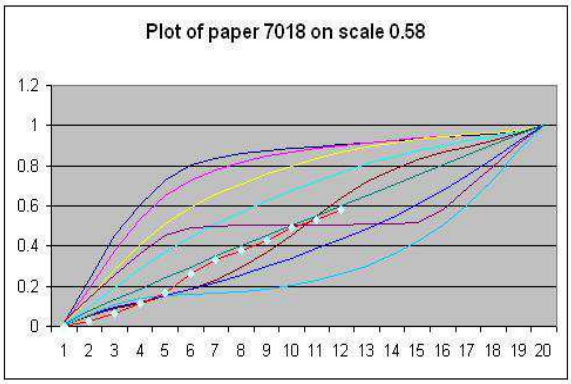

Figure 10: A sample curve matching on the 0.58 scale

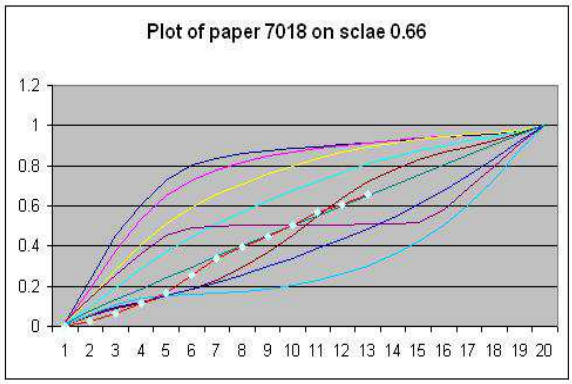

Figure 11: A sample curve matching on the 0.66 scale

sampled points for the dead curves. As mentioned before, in our case, K is 20 or 50 and corresponding values of M can be between 12 and 17 or 30 and 41. Let the number of time periods in the lifeline of the alive paper be L. Let the total number of citations that the alive paper has received so far be R_L. So we have to predict the number of citations in the time period $L+1$ or R_L+1.

The first step of the prediction is to find that to how many sample periods on the normalized scale do $L+1$ time periods of the lifeline correspond to. L time periods of the lifeline correspond to K sampled points. Let $L+1$ time periods will correspond to p sample points. We can find p as $p = \frac{L+1}{L}K$

Let $p_i = \lfloor p \rfloor$

and $p_i = p - p_i$

Let now the normalized extension value (on the scale of 0 to N) be V_N. Then
$V_N = c_{p_i} + (c_{p_{i+1}} - c_{p_i})p_f$

So the net extension (on the normalized scale of N) is

$E = V_N - c_M$

So now the predicted number of citations for the next time period is
$R_{L+1} = \frac{E.R_L}{N}$

Hence in this way obtain the prediction for next time period. To get the prediction for the two time periods, repeat the above procedure again after predicting for one time period.

5. EVALUATION OF THE SYSTEM

The actual values of the citations for the months of February to April 2003 were released on May 12, 2003. We evaluated our prediction system against this data. The percentage of error in the intervals of percentage of papers predicted is shown in Table 1. We found that the system was relatively more accurate for papers with relatively low values for the average citations per time period. The prediction was inaccurate if the number of citations in the recent months was high. This was because in case of higher values, the values attain a more random nature and it is difficult to fit a curve matching them. Another problem was of spiking. We found

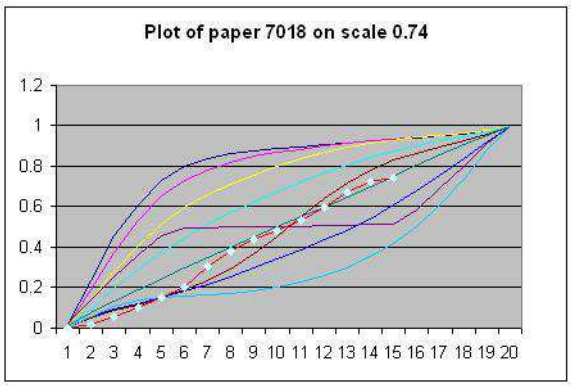

Figure 12: A sample curve matching on the 0.74 scale

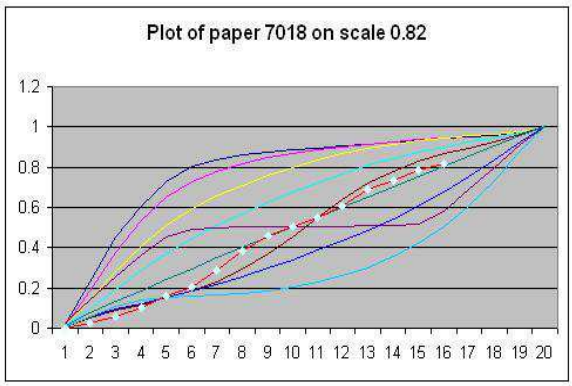

Figure 13: A sample curve matching on the 0.82 scale

Table 1: % Error in the prediction

Percentage Error	Papers in interval
>= 0 and <= 5%	3.62%
> 5% and <= 10%	5.8%
> 10% and <= 15%	3%
> 15% and <= 20%	3.6%
> 20% and <= 30%	8.6%
> 30% and <= 40%	8.3%
> 40%	9.2%
>= −5% and < 0%	5.2%
>= −10% and < −5%	3.54%
>= −15% and < 10%	5.78%
>= −20% and < −15%	4.1%
>= −30% and < −20%	10%
>= −40% and < −30%	9.3%
< −40%	20.1%

that the representative curves which fell below the diagonal or the linear curve, went flat for upto the first 80% time periods or so. After that, for every successive time period, they spiked suddenly. So if a paper matched the flat region, but the prediction fell in the spike region, then we get disproportionately high values for the predicted periods. To avoid this, we tried to use linear extrapolation for the curves which matched the lower diagonal representative curves, but could not do so in most cases because the results still remained substantially different from the obtained value. We will see this by an example.

106048 3 26 39 24 24 38

106112 1 3 5 3 4 9

Given above are two papers and their lifelines. The actual values for the next 3 months were 20 and 6 respectively. Our predicted values were 23 and 3 respectively. Now if we used linear extrapolation then the values we get are about 50 and 13 respectively, which are even more varying than our predictions. The reason for this is that a look at the dataset will tell us that the curves are all zigzag in nature and actually quite random. It is therefore very difficult to predict the next value based on the previous ones, because it may rise or fall in an indeterministic manner. What we did was to model the problem so as to transform it into a curve fitting and extrapolation problem with monotonically increasing curves. Here again the limitation was that firstly the curves which match are with a permissible error margin. So it.s not an exact match. Within that error margin, the predictions may actually vary by upto 20% based on various factors. Further, rather than trying to guess which stage of the lifeline the paper is actually at (which is again in some sense indeterministic) we assume that stage of the lifeline which shows the best match with any of the representative curves. So what it amounts to is that we try and force a paper to match one of the 9 curves, whereas in real life each citation curve might me unique in itself. The broad idea is to see how the citations vary and then try and predict based on other curves how that curve may proceed. In actuality, every paper is independent and we really cannot say that because another paper went this way, the current paper will also do so. At best we are trying to generalize based on the given dataset to get a broad classification of the patterns that various citation curves follow and predict the progress of a partial curve using that. Another limitation is that in the initial stages, many curves are almost similar, so even when we say that the given curve fits a particular representative curve, it may be closely similar to another representative curve as well. In such cases it is likely that the curve may follow any of them. We can at most make a random guess about which one it will follow. The above are the limitations of the system.

6. CONCLUSION

We have proposed a novel approach for forecasting time series data by identifying patterns using clustering. This is followed by prediction method which uses these patterns as a base to predict future trends. This approach yielded good results when applied to the Kddcup 2003 dataset.

The advantages of this technique is that it combines manual intervention and computing power to yield more efficient results. It can be used even when we have very small dataset, even with as few as as hundred time series. Further it allows us to compare time series of different lengths by normalising them on a scale of 0 to 1. It also avoids an extensive learning phase as is the case for many other techniques.

The disadvantages include the fact that it works more effectively when we are dealing with time series where the concept of a lifecycle is applicable. Another problem is that the predictions are less accurate if there have been heavy variations from the normal trend in the recent time periods.

Further refinements to this technique can be by taking moving averages when creating the initial curves for clustering. Future work can be to automate the selection of the patters without any manual intervention and also coming up with a better distance based function for clustering and non linear forecasting over the representative curve. This methodology can be used to predict a range of values rather than specific values. This will render more accuracy to the system.

7. REFERENCES

[1] Adya, M. Collopy, and F. Kennedy. Heuristic identification of time series features: an extension of rule-based forecasting. 1997.

[2] R. Agarwal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. pages 69–84. In Proceedings of the FODO Conference, 1993.

[3] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. Time series analysis: Forecasing and control (3rd ed.). Prentice-Hall, 1994.

[4] C. Chatfield. The analysis of time series. Chapman and Hall, 1989.

[5] M. W. Craven and J. W. Shavlik. Understanding time-series networks: A case study in rule extraction. 1998.

[6] G. Das, K. I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1998.

[7] C. Faloutsos, H. V. Jagadish, and B. Yi. Efficient retrieval of similar time sequences under time wrapping. pages 201–208. In Proceedings of the ICDE Conference, 1998.

[8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. pages 419–129. In Proceedings of the ACM SIGMOD Conference, 1994.

[9] J. Faraday and C. Chatfield. Time series forecasting with neural networks: A case study. Reserach Report, University of Bath, 1995.

[10] A. Harver. Time series models (2nd ed.). Hemel Hempstead, 1993.

[11] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks. Techincal Report LA-UR-97-2662, Los Alamos National Laborator, Los Alamos, 1987.

[12] A. Lendasse, M. Verleysen, and E. de Bodt. Forecasting time-series by kohonen classification. pages 221–226. In Proceedings of European Symposium on Aritifical Network, 1998.

[13] F. Lin, X. H. Yu, S. Grego, and R. Irons. Time series forecasting with neural networks. 1995.

[14] J. Lin, E. Keogh, and W. Truppel. Clustering of streaming time series is meaningless. pages 56–65. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003.

[15] A. Pankratz. Forecasting with dynamic regression models. pages 17–25. Wiley-Interscience Publication, 1991.

[16] M. Priestley. Spectral analysis and time series. Academic Press, 1981.

[17] E. Wan. Time series prediction by using a connectionist network with internal delay line. pages 195–217, 1993.

[18] A. S. Weigend and N. A. Gershenfeld. Time series prediction: Forecasting the future and understanding the past. Addison Wesley, 1994.

[19] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future: A connectinist approach. Number 1, pages 193–209. International Journal of Neural Systems, 1990.

[20] P. Werbos. *Beyond Regression: News Tools for Prediction and Analyis in the Behavioural Sciences.* PhD thesis, 1974.

[21] P. Werbos. Generalization of backpropagation with application to a recurrant gas market model. Neur. Net., 1988.