International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 7 No. 3 Aug. 2014, pp. 1034-1037

http://www.ijias.issr-journals.org/

A nascent approach to mine outliers using compression

Swati Vashisht[1], Shubhi Gupta[1], and Atul Mani[2]

[1]Computer Science, Amity Group of Institutions, U.P., India

[2]Mechanical Engineering, RKGEC, U.P., India

ABSTRACT: Outlier mining is concerned with the data objects that do not comply with the general behavior or model of the data, such data Objects, which are either different from or inconsistent with the remaining set of data. Studying the extra ordinary behavior of outliers helps uncovering the knowledge hidden behind them and providing an approach to the decision makers to make profit or improve the service quality. Hence, mining for outliers is an important data mining research with numerous applications, including credit card fraud detection, criminal activities in E-commerce, unusual usages of telecommunication services, Weather Forecasting etc. Moreover, it is useful in digital and customized marketing for identifying the spending behavior of customers with extremely low or extremely high incomes, or in medical diagnose for finding unusual results to various medical treatments.

Some data mining techniques discard outliers as noise or exceptions. While in some applications, these exceptions are considered more interesting than regularly occurring ones like in terrorism attack. Challenges in outlier detection include finding appropriate data models, the dependence of outlier detection systems on the application involved, finding techniques to distinguish outliers from error or exception, and providing justification for identification outliers. Outliers can be detected through N-gram technique but this technique is using a large storage space to store metadata and data dictionary. There are a number of compression models e.g. Content tree weighting method, LZ77, LZ78, LZW that are used in compressing text & image. Burrows–Wheeler transform (block sorting preprocessing that makes compression more efficient). Hence in this paper we are giving a compression technique (LZW compression model) that can compress data and will make this algorithm more efficient in terms of storage space.

KEYWORDS: Outliers, Compression, N-gram technique, weighting methods, storage space.

1 INTRODUCTION

Outliers can be caused by measurement or execution error, for example the display of an employee salary in negative could be caused by a program default setting of an unrecorded salary. Alternatively, outliers may be result of inherent data variability. The salary of the executive officers of a company could naturally stand out as an outlier among the salary of the other employees in the firm.

Outlier detection and analysis is an interesting data mining task, referred to as Outlier Mining that has a lot of real life applications in many different domains. Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. In contrast to traditional data mining task that aims to find general pattern applicable to a large amount of data, outlier detection aims finding of the rare data whose behavior is very exceptional when compared with rest large amount of data.

Outlier mining can be described as: Given a set of n data points or objects, and k, the expected number of outliers, find the top k objects that are considerably dissimilar, exceptional, or inconsistent with respect to the remaining data.

Corresponding Author: Swati Vashisht

1.1 N-GRAM TECHNIQUE

N-gram techniques have been studied and used in many information retrieval tasks. They have been applied in different domains such as language identification [5], document categorization and comparison, robust handling of noisy text and many other domains of natural language processing applications [3]. The success of n-gram-based systems is because the strings are discomposed into smaller parts causing errors (misspelled words, typographical errors and errors arising from object character recognition (OCR)) to affect only a limited number of such parts rather than the whole word. The number of n-grams (higher order n-grams) common to two strings is a measure of the similarity between the words. This measure is resistant to a large variety of textual errors.

N-gram techniques have been used for a number of text processing activities. N-grams of a string of length k is an n contiguous slice of the string into substrings each of size n. N gram systems suffer from large memory requirements because of the huge number of n-gram vectors resulting from the slicing. For example, a string of length k has k-n+1 possible n-grams ignoring all n-grams with trailing or preceding blanks. This problem notwithstanding, n-gram systems have some advantages over full word matching in web page categorization for the following reasons:

(1) A large number of document are posted on the web without going through any form of thorough error checking because they might be too costly or the documents may be time dependent. Such documents may contain significant amounts of errors making full word matching less efficient. In such cases, n-gram techniques offer more efficient and effective means of comparing strings because n-grams systems support partial matching of strings.

(2) Having same length n-grams, gives a great advantage to this technique as compared to words that may have different lengths.

N-gram models are widely used in statistical natural language processing. In speech recognition, phonemes and sequences of phonemes are modeled using a *n*-gram distribution. For parsing, words are modeled such that each *n*-gram is composed of *n* words. For language identification, sequences of characters (*e.g.*, letters of alphabets) are modeled for different languages.[7] .However, the term n-gram can be used for any set of consecutive characters occurring in a string (e.g., n-gram consisting of second, third and fifth characters in a string).The n-grams from a string of length k is obtained by sliding a window of size n over the string, starting at the first position and moving the window one position at a time until it reaches the end of the string. The set of characters that appear in the window at any position forms the n-grams of that string. For example, the string 'intelligence' has 'intel','ntell','telli','ellig','llige','ligen','igenc','gence'5-grams. The number of possible n-grams resulting from a string of length k is (k-n+1), where n is the size of the n-gram.

A study of different n-gram sizes reveals n-grams that are too short tend to capture similarities between words that are due to factors other than semantic relatedness. It may happen that having a significant number of common n-grams, a word seems related but they are not. Similarly, n-grams that are too long fail to capture similarities between similar but different words. N-grams that are too long behave like full word match. Notwithstanding, n-grams of reasonable lengths are able to determine similarity between different but related words better than n-grams of shorter lengths.

Data compression is the process of encoding data so that it takes less storage space or less transmission time than it would if it were not compressed. A code is a mapping from input symbols to sequences of bits. The process of converting an input sequence into a binary sequence by replacing each symbol in message with its corresponding code word is called encoding. When the purpose of encoding is to produce a compact representation of the original message, the process is called compression.[1]

Lossless data compression algorithms exploit regularities in the data to produce compressed representations from which the data can be reconstructed exactly. Lossless compression methods have been used traditionally on text compression and on compression on text encoded on binary data files such as semi structured documents, database files, computer programs etc.

Statistical modeling algorithms for text include:

- Context tree weighting method (CTW)
- Burrows–Wheeler transform (block sorting preprocessing that makes compression more efficient)
- LZ77 (used by DEFLATE) & LZ78
- LZW

Lempel Ziv Algorithm: These compression techniques use a symbol dictionary to represent recurring patterns. The dictionary is dynamically updated during a compression as new patterns occur. For data transmissions, the dictionary is

passed to a receiving system so it knows how to decode the characters. For file storage, the dictionary is stored with the compressed file.[2]

2 PROBLEM FORMULATION

A large number of techniques exist for detecting outliers from the web but almost none of the existing algorithms compress the given documents before detection.

Hence, in this paper we proposed a compression algorithm before N-Gram technique which will compress the data dictionary and preprocessed data and then N-gram technique analyze the contents of the Meta data of the web pages of related category and identify the web pages which are having significantly different content as compared to other pages.

3 PROPOSED WORK

Outlier detection can be done using N-gram technique but it takes a large space to store reside metadata here we are applying LZW algorithm for data compression before using this data for outlier detection.

3.1 COMPRESSION WITH LZW TECHNIQUE

LZW initiates with a dictionary as the "standard" character set. It then reads data bits at a time and encodes the data as index value given in the dictionary. When it comes across a new substring it adds it to existing dictionary; and in case of a substring it has already seen, it just reads in a new character and concatenates it with the current string to get a new substring. The next time LZW revisits a substring, it will be encoded using a single number, while repeating these steps compression is achieved.[2]

Pseudo code:

```
string s;
char ch;
s = empty string;
while (there is still data to be read)
{
ch = read a character;
if (dictionary contains s+ch)
   {
         s = s+ch;
   }
else
   {
         encode s to output file;
         adds+ch to dictionary;
         s = ch;
   }
}
```

Encode s to output file;

Now, suppose we wish to compress “xyxyyxyyy” and that we are only using the initial dictionary:

Index	Entry
0	x
1	y

Table 1. Encoding Steps

ENCODER OUTPUT		STRING TABLE	
OUTPUT CODE	**SYMBOL**	**NEW ENTRY**	**STRING**
0	x	2	xy
1	y	3	yx
2	xy	4	xyy
3	yx	5	yxy
1	y	6	yy
6	yy		

Encoded output is "012316".

4 CONCLUSION

The main advantage of compression algorithm is: It can improve detection function because compressed data will take less time for matching a pattern. N-gram technique would be more efficient in determining similarity between different but related words in text. The storage space problem can be solved but it requires additional computational cost. N-grams support partial matching of strings with errors and compressed data can become a little complex as compared to original data.

Compression algorithms can cause delay but in real time environment many applications may not tolerate any delays, in which case we may need to tune the compression levels and/or implement other techniques to remove those delays. Moreover, compression affects the portability of files in which case recipient system needs same software algorithm to decompress same data.

5 FUTURE WORK

Area of future research includes experimental evaluation of N-gram technique in terms of response time to improve its efficiency, different weighting methods for data dictionary words that will help to decrease response time. In future, a new approach could be given to match a full word that can minimize errors due to fixed length matching in N-gram technique. Future scope also includes analysis of performance of different data outlier mining algorithms.

REFERENCES

[1] Bratko Andrej, *Text mining using data compression model, 2012,* [Online] Available: http://eprints.fri.uni-lj.si/1925/1/Bratko1.pdf

[2] Sayood Khalid, *Introduction to data compression*, 4th Edition, Morgan Kaufmann Publishers, 2012.

[3] BCavnar., Trenkle M.J., "*N-Gram-Based Text Categorization"*, Proceedings of 3rd Annual Symposium on Document and Information Retrieval, 1994

[4] Charu. Aggarwal, Philip S. Yu, "*Outlier Detection for High Dimensional Data",* IBM T. J. Watson Research Center, Yorktown Heights, ACM SIGMOD 2001 May 21-24

[5] Damashek, M. Gauging, *Similarity with N-Grams: Language Independent Categorization of Text, Science*, 267(1995) pp 843-848

[6] FAngiulli, C.Pizzuti, "*Fast Outlier Detection in High Dimensional Space"*, In proceedings of PKDD'02,pp 25-36,2002

[7] Ted Dunning, *Statistical Identification of Language*, New Mexico State University. Technical Report MCCS 94-273

[8] V. Barnett and T. Lewis. *Outliers in Statistical,* 3rd Edition, John Wiley & Sons, 1994.

[9] Zengyou He , XiaofeiXu, Shengchun Deng, *A Fast Greedy Algorithm for Outlier Mining,* [online] Available: http://www.researchgate.net/publication/1958347_A_Fast_Greedy_Algorithm_for_Outlier_Mining