ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Detecting Similar Contents Using FNNS-LSH Model in E-Learning Environment

Lokeshbabu.R

PG Scholar, Department of Computer Science and Engineering, Arunai Engineering College, Thiruvannamalai, India

ABSTRACT: Similarity search is one of the fundamental problem in many fields of computer science like machine learning, data mining and recommended systems. Approximate FNNS by converting text document into sets by using shingling for reducing similarity between sets. Jaccard index or similarity used to compare similarity and diversity of a sets, by using the characteristic matrix and signature matrix. In characteristic shingle they contain large sets to reduce the size, minhash is implemented. Bloom filter provide a constant time. signature matrix to find pair of column it provide a difficulty to overcome LSH technique is used .we can find duplicate copy of set and semantically similar text or sets.

e-learning concept is including to refer the document which are easy to access the file in online reference detecting similar items from two or more file.

KEYWORDS*:* Fast Nearest Neighbourhood Search (FNNS), jaccard similarity,Locality Sensitive hashing(LSH), e-learning System.

I. INTRODUCTION

A advancement in the area of computation have forced to use complex data object, but extract such are more time consuming. In addition to it, Similarity search as become a primary computation task in range of application such as PatterRecognition, statistical analysis etc. The extract match is done by the distance concept In Similarity concept, object are represented as points in high dimensional space user queries always try to retrieve the nearest object. The main goal of similarity text document is to find pair that are near duplicate and semantically similar text and provide an efficient outcomes to the e-learning environment. In e-learning major problem for finding similarity in web based approach or media ,online studies etc.,

In e-learning system learners can learn any where at any time in online and also there may be a unauthorized detection of data or documents presented .Learners can interact them and produce some problem of high dimension they need to reduced and provide efficient data to learn for the readers .The learners measures similarity by the significant increase in performance and accuracy under different conditions. e-learning system improves the field of online resources they provide, an interface between the users with number of system provided

II. RELATED WORK

The Fast Nearest Neighbor search algorithm performs well for less-dimensional with a cluster of data but the complexity in both criteria (time and space) increases exponentially as the dimensions goes high, known as “curse of dimensionality” In previous datasets provide some improvement over the wide range.

Kyung Mi Lee and Keon Myung Lee [1] in this method proposed an algorithm to reduce false negatives in LSH. The false negative rate items are allowed to be hashed in multiple buckets at the extreme boundary of the bucket. The algorithm results in better accuracy but it has the cost of computation and memory space.

David Gorisse et al. [2] included Chi2 distance for Approximate Nearest Neighbor Search for high Dimensional data. In this they have defined different hash function. The Algorithm provides better Performance and exact LSH On image or video provided.

Sariel Har-Peled,Piotr Indyk,Rajeev Motwani [3] They develop a hashing based high-dimensional approximate similarity search scheme called LSH. They linearly depends on the size of the dataset Instead of partitioning method, it uses several hashing methods to hash the points to increase the probability of collision for objects which are similar than for those which are not. The near neighbors can be determined by hashing the queried point and retrieving those elements stored in buckets containing those points. This LSH scheme is applied when the points are given in binary Hamming space. The fundamental drawback of LSH is fast and simple only when the points are available in the Hamming space. It is further discussed that the algorithm can be used so that data can be extended to be in hamming distance, but it is achieved at time complexity cost and in increased error (by a large factor)data.

In M. Datar el al [4] improvement in the running time of the earlier algorithm has been purposed for the L2 norm which provide some outcomes results in an efficient Fast Approximate Nearest Neighbor scheme (FANN) for the case Lp norm (p<1). It takes O(log(n)) to search exact near neighbor for data having growth boundary conditions. The proposed LSH algorithm works directly on data points distributed in the Euclidean space and do not perform any per-processing. The proposed scheme gives 40 times faster results than kd-tree.

Kang Ling, Gangshan Wu [5] have propounded a new frequency based LSH scheme known as FBLSH using a p-stable distribution function as hash function and frequency threshold number is set. Items which collide more than the threshold are considered as nearest neighbor. The scheme is space efficient but it fails to address the time complexity issue.

G. Junhao et al.[6] proposed an LSH algorithm, To provide a Collision Counting LSH (C2LSH) that uses a m-base LSH functions to provide a dynamic compound hash functions instead of traditionally used static compound hash function. In C2LSH a collision threshold is used for data object to increase the query quality.

Hua et al. in [7] They proposed an algorithm LSBF which improves approximate membership query. This algorithm uses Locality sensitive hashing functions instead of uniform and independent hash functions in bloom filter to provide effective. Using LSH functions in bloom filter elements are stored locally sensitive thus can be search quickly. To reduce false positive they can be use bloom filter with a extra small size which verifies the AMQ query result.

A. Rajaraman, J. Ullman[8]In this concept they provide a movie rating based on recommended system by the user they can be easily find out and provide the outcomes

Kang Ling, Gangshan Wu in [10] have consists a new frequency based LSH scheme known as FBLSH where hash function and frequency threshold number is set which are used by the p-stable distribution function. Items which collide more than the threshold are considered as nearest neighbor. The scheme is space efficient but it fails to address the time complexity issue

III. COMPONENTS

1)Similarity Search

The Metric Space Approach focuses on finding the efficient ways for locating user relevant information in collections of objects, the similarity quantified uses a pair wise distance measure. Similarity search has become a primary computational task in a range of application area. It includes pattern recognition, data mining, biomedical databases, Multimedia information retrieval, machine learning, data compression, computer vision and statistical data analysis. The exact match has rare meaning in these environments whereas proximity/distance concepts (similarity/dissimilarity) are typically much more beneficial for searching

They consists of Two Search in Similarity they are:

- Vector Spaces
- Metric Spaces

2)Lexical Similarity

In linguistics, lexical similarity will be defined as the degree of similarity among the two given word sets of any two given languages. The lexical similarity of 1 (or 100%) means the overall similarity among vocabularies and lexical similarity of 0 means there are no common words. Shingling refers to the detection of near-duplicate items. positive integer k and a sequence of terms in a document D, the k-shingles of D are defined as a set of all the successive sequences of k terms in D. The general approach followed to convert a document into a set is to shingle the document. k-shingles define a set of all k size non repeatable substrings of the document, and group them as single object. The set of k-shingles of a document with n words takes space O(kn). The space goes on decrease as items are repeated in the document.

For example, if k=3 and a document may contains text “This LSH Project is efficient ” then the shingle‟s set will be {“This LSH Project”, “LSH Project is”, “Project is efficient”}.

3)Semantic Similarity

Define over a set of text files where the distance among files is achieved from the equality of their semantic content as compared to similarity based on their syntactical representation (i.e. String type).The semantic relationship strength is calculated based on mathematical tools.

4)Duplicate

There are abundant of duplicate web pages in the mirror sites on the internet. Two such documents may differ from each other only by small portion of text. So it is very important to find and remove such duplicates search results to enhance the reliability of internet sources. Exact Duplicate is very easy to test for the two Documents if they are exactly similar or not. The two Documents are compared character by character to test exact duplicates and they are not same if they ever differ .However it is not the right method to test the similarity of the documents. It is very convenient for removing the redundancy in the system. This technique is used to find and remove duplicate documents, audio, photos, etc, The exact duplicates of documents in the selected folder. The duplicates of the documents are checked and will be deleted from the folder. By reducing duplicate data or documents dimensional we reduce and provide an efficient outcomes.

5)Near Duplicate

Near-duplicates are based on the identical primitive text which is altered and post processed and results in two discrete files. Near-duplicates can also be related to show the identical scene or event. Multimedia content analysis faces objection for the detection of near-duplicates. Both informational and redundant signals are carried by near-duplicate. For example, near duplicates may supply rich visual evidences for summarizing videos from various origins. On the contrary, browsing of streaming web videos over internet is an intense time consuming task due to the exaggerated amount of near-duplicates. Subsequently, Academia,& industry and Government organizations have shown a keen interest in near-duplicates using numerous multimedia applications and for the web scale search, detection and elimination of near-duplicates.

6)Bloom filter

Bloom Filter, a probabilistic data structure that uses multiple hash functions to store data in a large bit array.It reduces O(n) search time to constant time.

The search accuracy depends on:

1)The size of Bloom Filter

2) Number of hash functions

The major advantage of multiple hash functions is that search accuracy is improved. This approach is very effective when speed matters more than space. Suppose we have a set u having n elements, which are to be hashed using Bloom Filter. The size of the Bloom Filter is m. Size of Bloom Filter is usually taken larger than the size of data to be stored to avoid collisions. K, the number of hash functions, depends on the size of data to be stored and the size of the bit array (bloom filter).

The insertion and membership testing of an item are of complexity *O(k)* in the m bits bloom filter and with k number of hash functions. Every time an element is added to the set or checked for set membership, the element needs to run through the *k* hash functions.

DOI: 10.15680/IJIRCCE.2016. 0402033

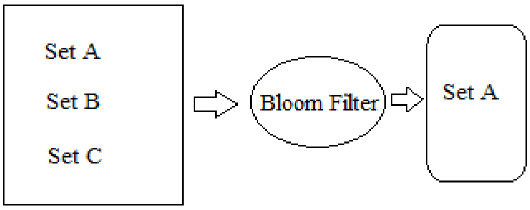

Fig1:Bloom filter

Bloom Filter is used to improve space complexity and it also provides constant search time for membership test
7)Fast Nearest neighbor search

Fast Nearest neighbor search (NNS) is an optimization problem to find nearest neighbors in metric spaces. FNNS is an optimization problem to locate the most similar items in a given dataset. Similarity or closeness is usually expressed in terms of a dissimilarity function which says “The less similar are the objects the larger are the function values”. Due to an application of assigning to a residence the nearest post office, nearest neighbor search problem was also known as post-office problem and direct generalization of this problem is a k-NN search, where the k closest points.

The curse of dimensionality in the NNS context basically means that Euclidean distance is unhelpful in high dimensions because all vectors are almost in a equal distant to the search query vector (imagine multiple points lying more or less on a circle with the query point at the center; Distance from finded point to search point will be almost same)

IV. SYSTEM ARCHITECTURE

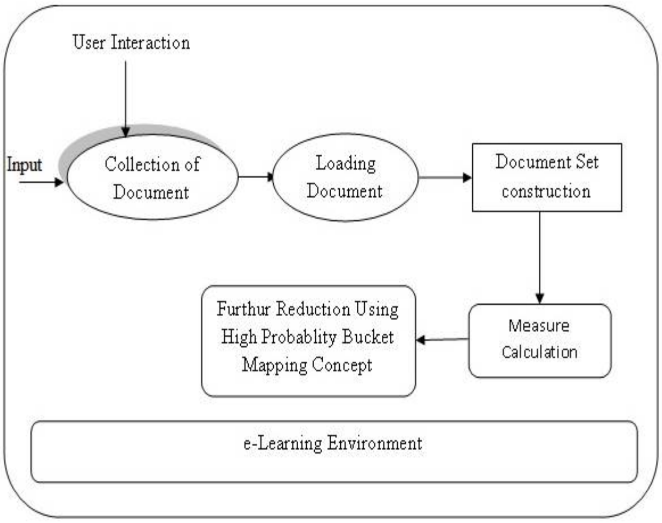

Fig2:System Architecture

a. Collection of Documents

Input that are provided by Documents are collected and stored .There may be a stop words Each Stop words are removed because they do not provide meaningful result. By the result of Collected documents in many fields , one of the major filed is e-learning provide efficient learning for the user to interact with it.

b. k-Shingle

Documents are converted by the steps using K-Shingle method. K-Shingle consecutive words group them as a single object. So the set of all 1-shingles is exactly the bag of words model. An alternative name to k-shingle is an k-gram.

D1: This is my book

D2:Book my is This

D3:Library consists of many book

Each Documents shingle size is calculated as 1 by

D1UD2U D3={This},{is},{my},{book},{library},{consists},{of},{many}.

c. Measure calculation

Jaccard Distance :The Jaccard Distance J-DIS(S1,S2), between two sets is defined as one minus Jaccard similarity between those two sets i.e. J-DIS (S1, S2) = 1 – J-SIM(S1, S2). Jaccard distance is defined for sets. Jaccard Similarity J-SIM(S1, S2) between two sets is calculated as the ratio between the intersection size and the union size of the sets. Jaccard Distance has all the constraints of a distance measure

1)Since the size of intersection is always less than or equal to the size of the union, Jaccard Distance satisfies the non-negative constraints.

2)The size of union and intersection of two sets can never be same at the same time except the case when both sets are same. Jaccard Similarity is one only when same sets are used. Further, Jaccard Distance is strictly positive

3)Union and intersection of two sets are always symmetric; Jaccard Distance satisfies symmetric axiom.

4)Jaccard Similarity always satisfies triangular inequality, and so does Jaccard Distance.

$$\text{J-SIM}(S1,S2)=|S1\cap S2| / |S1\cup S2|$$

d. Charterstic matrix/minhash

To find similar documents in a given collection We can use Shingling and Minhashing Techniques. Minhashing is a type of LSH techniques in which independent permutations of a given column are used for finding the similarity of items. These Minhashing provides large sets to short signature and use several independent hash function to create signature matrix hash function h. pi (C)= the number of the first(in the permuted order pi)row in which column C has value1:h

$$pi\ (C) = \min pi(C)$$

Input matrix are created and they are listed and computed as a signature matrix .We represent the sets in characteristic matrix say M. The size of this matrix is (m x n) where m is the number of rows and n is the number of sets. The columns of the matrix are the sets. The entry Mij= “1”, if i^{th} shingle is present in j^{th} set, otherwise Mij=“0”, where 0<=i<=m, and 0<=j<=n. Some hash functions are defined which are much less than number of row. All these hash functions generate some permutations of present rows. These hash functions will we used to create Minhash signature matrix. Let there are three documents D1,D2,AND D3 and their shingles sets are DS1 {A, F, G}, DS2 {A, B, C}, DS3 {A, E, F, G}, respectively. Shingles set {A, B, C, E, F, G} after hashing every shingle gets a bucket number let bucket numbers are {0, 1, 2, 3, 4}.

e. Locality Sensitive Hashing

Focus on pairs of signatures likely to be from similar documents .In general hashing, closed (near) items may be hashed in different locations after hashing, but in case of Locality Sensitive Hashing items maintain their closeness even after hashing (mapping) In LSH, candidate pairs are those pairs which hash to the same bucket when banding technique is applied and comparisons are performed only on candidate pairs rather than comparing every pair like in linear search. If the requirement is to find exact match then one of the techniques to process data like Map Reduce, Twitter storm etc. can be used. These techniques are based on parallelism so result in time reduction. But these methods require extra hardware.

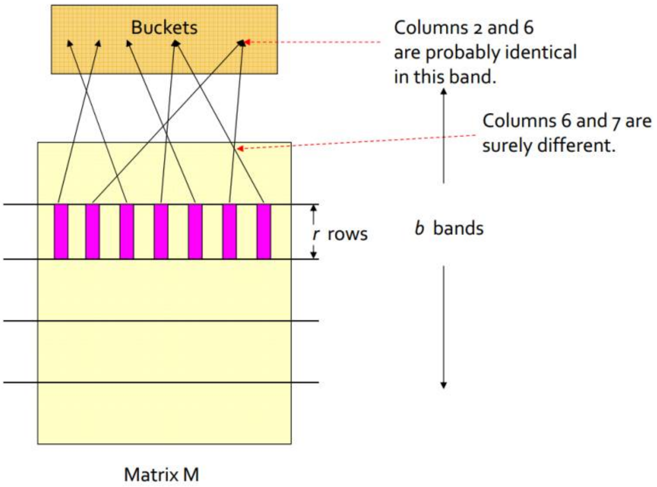
Buckets
Columns 2 and 6 are probably identical in this band.
Columns 6 and 7 are surely different.
r rows
b bands
Matrix M

Fig 3:Locality Sensitive Hashing

Locality Sensitive Hashing is used for finding candidate pairs so that only subsets of items are compared. Sometime exact result is desired in many cases we only want pairs that are most similar. The similarity level is defined by some threshold. In these cases the desired result is called FastNearest Neighbor Search. When finding similar documents, a family of functions called Minhash functions is combined with banding technique to separate pairs at low distance from the pairs at large distance Another important consideration is that false negatives and false positives should be avoided as far as possible. Further, the combined function must take very less time than the number of pairs. Let us assume d1 and d2 are two distance parameter (d1<d2) according to some distance measurement d and F is family of functions. F is called (d1, d2, p1, p2)-sensitive if for all f belongs to F; Probability of f(x) =f(y) is

Atleast p1 if d(x,y)<d1

Atmost p2 if d(x,y)>d2

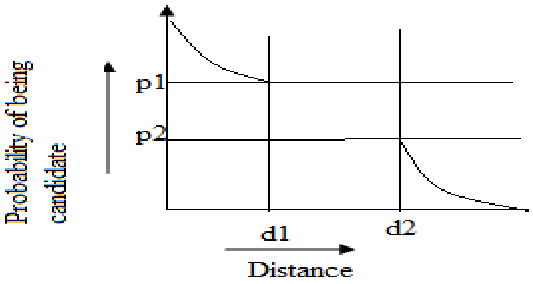

Fig4:Behavior of a (d1 , d2, p1, p2)-sensitive function

V. Conclusion

In this project we have proposed Similar detection of datasets by reducing high Dimensional in e-learning environment by Locality sensitive hashing (LSH) which can be used in parallel environment and also in Euclidean Distance or Hamming Distance in multiple data sets .In Further work we can implement parallel environment with Fast

DOI: 10.15680/IJIRCCE.2016. 0402033

FNNS. Simple hash functions are used in Bloom Filter and in Locality Sensitive Hashing. Both accuracy and speed depends mainly upon those hash functions. Effective in terms of speed and accuracy by using some fast data (problem) specific hash functions. The problem of false positive due to bloom filter and the problem of false negative and false positive due to LSH will be minimized by using better hash functions..

REFERENCES

1. K. Mi Lee, K. Myung Lee, "Similar pair identification using locality-sensitive hashing technique", Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), pp.2117-2119, 20-24 Nov.
2. D. Gorisse, M. Cord, and F. Precioso, "Locality-Sensitive Hashing for Chi2 Distance", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.2, pp.402-409, Feb. 2012
3. P. Indyk and R. Motwani, “Approximate nearest neighbor: towards removing the curse of dimensionality”, Proc. Symposium on Theory of Computing, 1998.
4. M. Datar el al., “Locality-sensitive hashing scheme based on p-stable distributions”, Proc. ACM Symposium on Computational Geometry, 2004.
5. K. Ling; G. Wu, "Frequency Based Locality Sensitive Hashing", International Conference on Multimedia Technology (ICMT), pp.4929-4932, 26-28 July 2011 .
6. G. Junhao et al., “Locality-Sensitive Hashing Scheme Based on Dynamic Collision Counting”, ACM, 2012
7. Y. Hua et al., "Locality-Sensitive Bloom Filter for Approximate Membership Query", IEEE Transactions on Computers, vol.61, no.6, pp. 817-830, June 2012.
8. A. Rajaraman, J. Ullman, “Mining of Massive Datasets”, Cambridge University Press, December 30, 2011 .
9. Dasgupta, R. Kumar, and T. Sarlós, “Fast Locality- Sensitive Hashing”, ACM conference, New York, USA, pp. 1073-1081, 2011.
10. K. Ling; G. Wu, "Frequency Based Locality Sensitive Hashing", International Conference on Multimedia Technology (ICMT), pp.4929-4932, 26-28 July 2011
11. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions”, Comm. ACM, pp. 117-122, 2008.
12. L. Qin et al., “Multi-probe LSH: Efficient indexing for high-dimensional similarity search”, Proc. VLDB, 2007.
13. R. Panigrahy, “Entropy-based nearest neighbor algorithm in high dimensions”, Proc. ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, USA, pp. 1186- 1195, 2006.
14. U. Manber, “Finding similar files in a large file system”, Proc. USENIX Conference, 1994.