International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072

AN EFFICIENT SHORTEST PATH APPROACH USING BFS ALGORITHM WITH SMC PROTOCOL FOR PRIVACY PRESERVING RECORD LINKAGE

S.Abirami [1], Dr.J.Suguna [2]

[1] *Research Scholar, Department of Computer Science, Vellalar College for Women Tamilnadu, India*
[2] *Associate Professor, Department of Computer Science, Vellalar College for Women Tamilnadu, India*

Abstract - ***In data mining, clustering is a method of grouping data into different groups, so that the data in each group share similar trends and patterns. The fuzzy clustering is a method that allows the objects to belong to several clusters simultaneously, with different degrees of membership. Record linkage is the process of matching records from several databases that refer to the same entities. Linking large collections of records that protecting their privacy have arisen recently as an intriguing problem in the core of the domain of privacy preserving record linkage (PPRL). To overcome this problem, proposed the well-known Breadth First Search (BFS) technique for identifying candidate record pairs, which have undergone an anonymization transformation and also find the shortest path. The matching of pairs can be implemented by a Secure Multiparty Computational (SMC) based protocol that performing homomorphic distance computations. There are three well known measures (jaccard, Euclidean, Hamming) used for the matching process. The Pairs Completeness (PC), Pairs Quality (PQ), and the Reduction Ratio (RR) metrics are employed to evaluate the efficiency with respect to accuracy in finding the truly matched record pairs. Finally concluded that the proposed scenario yields superior performance than the existing scenario through Locality Sensitive Hashing (LSH).***

Key Words: *Privacy Preserving Record Linkage, Breadth First Search, Fuzzy Clustering, Secure Multi party computation, Bloom Filter.*

1. INTRODUCTION

1.1 DATA MINING

Data mining [6] refers to extracting or mining knowledge from large amounts of data. Data mining is a non trivial process that identifying valid, novel, potentially useful and also ultimately understandable patterns in data. The process term implies that data mining consists of many steps, the non trivial means process is not straight forward and some search or inference is involved. The pattern term is an expression in some language describing a subset of data, finding structures from data, in general making any high level description of a set of data.

Data mining also popularly known as Knowledge Discovery in Databases (KDD), refers to the non-trivial extraction of implicit, previously unknown and potentially useful information from data in databases. This process consists basically of steps that are performed before carrying out data mining, such as data selection, data cleaning, pre-processing, and data transformation.

Clustering [7] is a process of grouping objects with similar properties. In fuzzy clustering [10], the data points can belong to more than one cluster, and associated with each of the points are membership grade which indicate the degree to which the data points belong to the different clusters.

There are many other terms carrying a similar or slightly different meaning to data mining such as knowledge mining from databases, knowledge extraction, Data/pattern analysis, Data archaeology and Data dredging. A standard definition for data mining is the non-trivial extraction of implicit, previously unknown, and potentially useful knowledge from data.

1.2 PRIVACY PRESERVING RECORD LINKAGE (PPRL)

Record linkage plays a central role in many data integration and data mining tasks that involve data from multiple sources. In many record linkage applications, identifiers have to be encrypted to preserve privacy. Therefore, a method for approximate string comparison in private record linkage is needed. The problem of finding records that represent the same individual in separate databases without revealing the identity of the individuals is called the “private record linkage”, “blind data linkage”, or “privacy-preserving record linkage” problem. Privacy-preserving record linkage (PPRL) [4] becomes increasingly important to match and integrate records with sensitive data. PPRL not only has to preserve the anonymity of the persons or entities involved but should also be highly efficient and scalable to large datasets.

The thesis is organized as follows:

Section 2 discusses the overview of the existing system with Locality Sensitive Hashing (LSH) and Efficient Transitive Region (ETR).
Section 3 discusses about the proposed Breadth First Search (BFS) algorithm with Secure Multi-Party Computational protocol.
Section 4 presents the experimental results of research and the performance of the algorithm.
Section 5 concludes this dissertation.

2. OVERVIEW OF EXISTING SYSTEM

Locality Sensitive Hashing (LSH) is a basic primitive in large scale data processing algorithms that are designed to operate on objects in high dimensions and it is a hashing based high-dimensional approximate similarity search scheme. They also reduce the dimensionality of high-dimensional data. LSH hashes input items so that similar items map to the same "buckets" with high probability (the number of buckets being much smaller than the universe of possible input items). LSH differs from conventional and cryptographic hash functions because it aims to maximize the probability of a "collision" for similar items. LSH has much in common with data clustering and nearest neighbor search.

An *LSH family F* is defined for a metric space *M* = (M ,d), a threshold R > 0 and an approximation factor c > 1. This family *F* is a family of functions h : M → S which map elements from the metric space to a bucket s ∈ S. The LSH family satisfies the following conditions for any two points p, q ∈ *M*, using a function h ∈ *F* which is chosen uniformly at random:

- if d(p, q) ≤ R , then h (p) = h (q) (i.e., *p* and *q* collide) with probability at least P_1,
- if d(p, q) ≤ cR, then h (p) = h (q) with probability at most P_2.

A family is interesting when $P_1 > P_2$. Such a family *F* is called *(R,cR,* P_1 *,* P_2 *) -sensitive*.

Efficient Transitive Region (ETR) is a performance-based initiative that utilizes the shortest path transitive regions to construct the shortest path. ETR constructs an auxiliary shortest path step by step, instead of considering single step at a time using shortest path trees during traversals.

ETR when used along with graph clustering technique, which usually requires graph analyses thereby grouping similar vertices together. It introduces an optimization strategy based on subgraph traversal to improve the performance of the clustering process. It also helps in achieving scalability without incurring extra overhead and index construction.

3. PROPOSED SCENARIO

Breadth First Search (BFS) [3] is a graph search algorithm for visiting all nodes connected to some node v_0 in a graph **G = (V, E)** exactly once. It can also be viewed as computing single source shortest paths on unweighted graphs. In particular, the algorithm visits nodes by increasing order of their distance to v_0, where the distance of a node w to v is the length of a shortest path from *w* to v.

The implementation of BFS follows from a simple observation. For $i \geq 0$, let F_i be the set of nodes with distance i to v_0. Notice that, given F_0,..., F_i , one can compute F_{i+1} by simply taking the neighbors of F_i which are not in F_0 .... F_i. This is because a node *w* is not in F_0 .... F_i if it has no path of length $0 \leq j \leq i$ to v_0, hence the length of a shortest path from *w* to v_0 is at least *i* +1. the other hand, if *w* is a neighbor of node $v \in F_i$, then it has a path of length at most *i* + 1 to v_0.

The algorithm is as follows:
Step 1: Color v_0 gray and color every other node white, and place v_0 onto a queue *Q*.
Step 2: If *Q* is non-empty, pop off the front node *v* of *Q*, visit *v*, and change the color of *v* to black. Otherwise, terminate.
Step 3: For each white-colored neighbor *w* of *v*, color *w* gray and push *w* onto the end of *Q*.
Step 4: Repeat (1) and (2).

```
BFS (G,s) // G is the graph and s is the starting node
for each vertex u ∈ V[G] – {s}
do color[u] ← WHITE // color of vertex u
d[u] ← ∞   // distance from source s to vertex u
TT[u] ← NIL   // predecessor of u
color[s] ← GRAY
d[s] ← 0
TT[s] ← NIL
Q ← Ø
ENQUEUE(Q,s)
While Q ← Ø      // iterates as long as there are gray vertices. LINES 10-18
do u ← DEQUEUE(Q)
for each v ∈ Adj[u]
```

do if color[v] = WHITE // discover the undiscovered adjacent vertices
then color[v] ← GRAY // enqueued whenever painted gray
d[v] ← d[v] + 1
TT[v] ← u
ENQUEUE(Q,v)
color[u] ← BLACK // painted black whenever dequeued

It is an induction to prove that, for all i ≥ 0 and 1 ≤ k ≤ $|F_i|$, the ($|F_0|$ +....+ $|F_{i-1}|$ +k) node to be pushed onto Q is a node in F_i, and furthermore, each node is pushed onto Q at most once. It is also modify the algorithm to record the distance of each node from v_0 (if there is no path from v_0 to a node, it can be set to ∞, effectively $|V|$).

The space complexity of BFS is O(|V |) for the queue and the colors of the nodes. The time complexity depends on the representation of G and the size of the connected component $G' = (V', E')$ that v_0 lies in. Regardless of representation, the algorithm requires O($|V'|$) time to push and pop nodes from the queue and color them.
If the adjacency list representation (each node has a list of neighbors) is used, then the algorithm requires O($|E'|$) time for processing neighbors (each edge is examined twice; once from each end), for a total of **O($|V'| + |E'|$)** time. The BFS algorithm can be adapted to solve the connected components problem we saw in the last tutorial. The approach will be to run BFS on an arbitrary node v_0, and then to repeatedly look for white nodes; if one exists, then run BFS on it but color the nodes a different color for each BFS; otherwise, terminate. At the end, the color of a node will determine the connected component it lies in.
The BFS could save more computation cycles that allow accurate information provided to the right people at the right time. Two considerations when forming a data warehouse are data cleansing (including entity resolution) and with schema integration (including record linkage). The uncleansed and fragmented data requires time to decipher and may lead to increased costs for an organization, so the data cleansing and schema integration can save a great many (human) computation cycles and can lead to higher organizational efficiency.

3.1 SECURE MULTI-PARTY COMPUTATIONAL PROTOCOL (SMC)

Secure multi-party computation (also referred to as secure computation or multi-party computation/MPC) [2] is a subfield of cryptography with the goal to create methods for parties to jointly compute a function over their inputs, and keeping these inputs private. An SMC protocol is useful in situations where owners of data sets are not willing to sharing them, even in anonymized representation, with a third-party. By employing homomorphic mechanism, matching of the corresponding record pairs can be accomplished by the owners themselves, by simply exchanging encrypted records and calculating the intermediate variables.
A popular cryptographic system is the partially homomorphic Paillier cryptosystem, which performs homomorphic addition computation. Successive encryption of the same number generates different cipher texts with high probability. A trusted authority is required in order to issue a public (*pk*) private (*pv*) key pair, needed for the encryption and decryption respectively. Given two numbers, n1 and n2, encryption is performed using the public key, which is denoted as $\widetilde{n1} = \mathcal{E}_{pk}(n1)$ and $\widetilde{n2} = \mathcal{E}_{pk}(n2)$, respectively.

3.2 QUALITY MEASURES

The total number of matched and non-matched record pairs are denoted with nM and nN, respectively, with nM + nM = nA × nB for the linkage of two databases, and nM + nN = nA(nA − 1)/2 for the deduplication of one database. The number of true matched and true non-matched candidate record pairs generated by an indexing technique is denoted with sM and sN, respectively, with sM + sN <_ nM + nN. The following measures are used to evaluate the efficiency, with respect to accuracy of the framework in finding the truly matched record pairs [5].

The **Reduction Ratio,** RR = 1.0− sM+sN / nM+nN , measures the reduction of the comparison space, i.e. the fraction of record pairs that are removed by an indexing technique. The higher the RR value, the less candidate record pairs are being generated. However, reduction ratio does not take the quality of the generated candidate record pairs into account (how many are true matches or not).

Pairs Completeness, PC = sM / nM , is the number of true matched candidate record pairs generated by an indexing technique divided by the total number of true matched pairs. It measures how effective an indexing technique is in not removing true matched pairs. PC corresponds to *recall* as used in information retrieval.

Finally, **Pairs Quality,** PQ = sM / sM+sN , is the number of true matched candidate record pairs generated by an indexing technique divided by the total number of candidate pairs generated. A high PQ value means an indexing technique is efficient and generates mostly true matched candidate pairs. On the other hand, a low PQ value means a large number of non-matches are also

generated. PQ corresponds to *precision* as used in information retrieval.

4. RESULTS AND DISCUSSION

The UCI Machine Learning Repository is a collection of datasets, domain theories and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The description of the crime dataset obtained from the UCI repository for the research work is shown in the following table.

Table -1: Dataset Description

DATASET	ATTRIUTES	DATA POINTS
Crime dataset	14	167

The **pairs completeness** metric values of Breadth First Search algorithm (BFS) are compared with the existing Locality Sensitive Hashing algorithm (LSH) and Efficient Transitive Region (ETR). It is found that the Breadth First Search algorithm yields better results than LSH and ETR. The experimental are shown in Table -2 and figure -1.

Table -2: Pairs Completeness Rate between ETR,LSH,BFS.

No of nodes	ETR	LSH	BFS
1	0.90	0.93	0.97
2	0.88	0.95	0.98
3	0.70	0.93	0.95
4	0.60	0.92	0.94

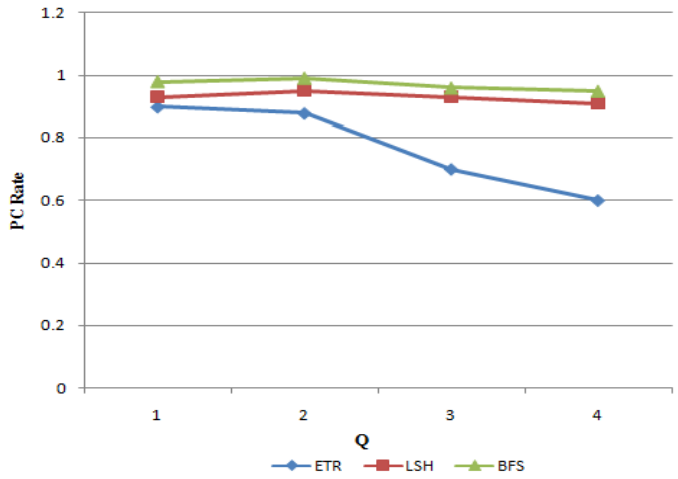

Figure -1: Pairs Completeness Rate

The **pairs quality** metric values of Breadth First Search algorithm (BFS) are compared with the existing Locality Sensitive Hashing algorithm (LSH) and Efficient Transitive Region (ETR). It is found that the Breadth First Search algorithm yields better results than LSH and ETR. The experimental are shown in Table -3 and figure -2.

Table -3: Pairs Quality Rate between ETR,LSH,BFS

No of nodes	ETR	LSH	BFS
1	0.09	0.16	0.18
2	0.08	0.12	0.10
3	0.04	0.10	0.08
4	0.02	0.06	0.07

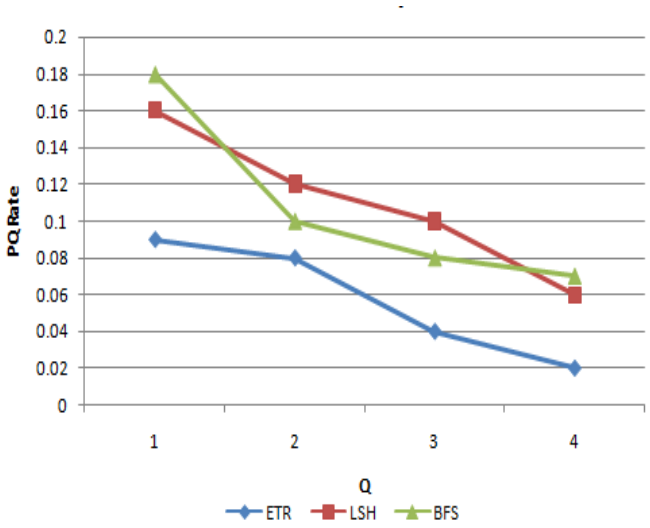

Figure -2: Pairs Quality Rate

PERFORMATION EVALUATION METRIC LEVEL

The resultant clustering was evaluated for execution time analysis to measure its quality. The execution time values of Breadth First Search algorithm are compared with the existing Locality Sensitive Hashing algorithm (LSH) and Efficient Transitive Region (ETR). It is found that the Breadth First Search algorithm (BFS) yields better results than LSH and ETR. The experimental are shown in Table -4 and figure -3.

Table -4 Execution Time between ETR,LSH,BFS.

Measures	ETR	LSH	BFS
PC(Pairs Completeness)	0.543	0.307	0.884
PQ(Pairs Quality)	0.004	0.005	0.007
RR(Reduction Ratio)	0.831	0.985	0.999
4	0.02	0.06	0.07

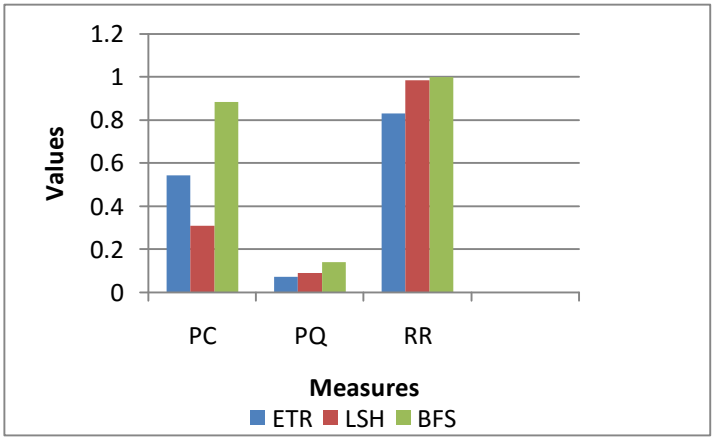

Figure -3 Execution Time

The metrics such as accuracy and time factors are used to compare the existing and proposed scenario using the performance metrics. In existing scenario, the accuracy values are lower and time complexity is high. In proposed scenario, the accuracy value is higher and time complexity is reduced significantly.

5. CONCLUSION

The Breadth First Search based shortest path approach is applied to preserving the privacy of the record linkage and to compare the formulated record pairs homomorphically Secure Multi-Party Computational protocol is used. The experimental results indicate high performance in finding truly matched record pairs and significant reduction in the number of candidate pairs obtained by Breadth First Search algorithm which is better than the Locality Sensitive Hashing algorithm. The system results are also verified with the test dataset and centralized dataset using appropriate validation measures.

REFERENCES

[1] **Al-Lawati. A, Lee. D, and McDaniel. P,** "Blocking-aware private record linkage," in Proc. 2nd Int. Workshop Inf. Quality Inf. Syst., 2005, pp. 59–68. @ 5

[2] **Atallah. J and Mand Du. W,** "Secure Multi-Party Computation Geometry." *Seventh International Workshop on Algorithms and Data Structures (WADS 2001)*, Providence, Phode Island, USA, Aug 8- 10, 2001, pp 136-152.

[3] **Bader. D.A and Madduri. K,**" Designing multithreaded algorithms for breadth first search and st-connectivity on the Cray MTA-2". In Proc. 35th Int'l. Conf. on Parallel Processing (ICPP 2006), pages 523-530, August 2006.

[4] **Bonomi. L, Xiong. L, Chen. R and Fung B. C. M.,** "Frequent grams based embedding for privacy preserving record linkage," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage., 2012, pp. 1597–1601. @ 3

[5] **Christen. P and Goiser. K.,** "Quality and complexity measures for data linkage and deduplication," in *Quality Measures in Data Mining*, ser. Studies in Computational Intelligence, F. Guillet and H. Hamilton, Eds., vol. 43. Springer, 2007, pp. 127–151.

[6] **Han, J., Kamber, M.,** *Data Mining Concepts and Techniques*, Morgan Kaufmann Publisher, 2001

[7] **Kotsiantis. S, Pintelas. P**, "Recent Advances in Clustering: A Brief Survey", WSEAS Transactions on Information Science and Applications, Vol. 1, No. 1 (73-81), 2004.

[8] **Manuel Then and Moritz Kaufmann.,** " The more the merrier: Efficient Multi-Source Graph Traversal," 2014.

[9] **Mukherjee. S, Chen. Z and Gangopadhyay. A.,** "A fuzzy programming approach for data reduction and privacy in distance-based mining", Int. J. Information and Computer Security, Vol. 2, No. 1, 2008. @ 4

[10] **Naga Lakshmi .M and Sandhya Rani .K.,** " A Privacy Preserving Clustering method based on Fuzzy Approach and Random Rotation Perturbation", Vol 04, Special Issue01; 2013.