Transaction Encoding Algorithm (TEA) for Distributed Data

A.Anbarasi
Research Scholar
Dept of Computer Science
Bharathiar University

D.Sathyasrinivas
Asst Prof/HOD
Dept. Of Computer Applications
Karpagam University

Dr.K.Vivekanandan
Professor
Dept Of Management
Bharathiar University

ABSTRACT

Analysis of huge datasets has been a major concern in almost all areas of technology in the past decade and the role of data mining has become so crucial as a result of this crisis. As the data sizes in these datasets increase, from gigabytes to terabytes or even larger the complexity in collecting and warehousing these massive dataset as such in a single site is practically impossible as it may not have enough main memory to hold all the data. Therefore they are accumulated usually in geographically distributed sites. The challenge in distributed data mining is how to learn as much knowledge from distributed databases as we do from the centralized database without costing too much communication bandwidth. A solution to distributed data mining is that the massive dataset can be collected and warehoused in a single site if its dimensionality is reduced. The dimension reduction algorithms are generally classified into feature selection, feature extraction and random projection. In this paper we propose a dimension reduction algorithm, which is different from all of these methods, to encode the transactions which reduce the size of transaction that in turn reduces the communication cost. Experimental results on a datasets demonstrate the performance of our proposed algorithm.

Keywords Centralized Database, Data Mining, Distributed Data Mining, Dataset Dimension reduction.

1. INTRODUCTION

Recent development of high throughput data acquisition technologies in a number of domains (e.g., biological sciences, commerce etc) together with advances in digital storage, computing, and communications technologies have resulted in the explosion of distributed data repositories created and maintained by autonomous entities. Data mining technology extended to these data rich domains offer extraordinary opportunities in computer assisted data-driven knowledge acquisition in a number of applications including, data-driven scientific discovery, data-driven decision making in business, monitoring and control of complex systems and security informatics[Syed Zahid, Hassan Zaidi, Syed Sibte Raza Abidi and Selvakumar Manickam][Lamersdorf, M. Merz][Philip Machanick]. Data mining in distributed systems can be carried out in two different fashions: data from distributed locations are transferred to a central processing center where distributed databases will be combined into a data warehouse before any further processing is to be done. During this process, large amounts of data are moved through the network. A second framework is to carry out local data mining first and then deriving global knowledge by integrating partial knowledge obtained from local databases [Wu-Shan Jiang, Ji-Hui Yu.] [Talia]. It is expected that by integrating the knowledge instead of data, network bandwidth can be saved and computational load can be more evenly distributed. Since the partial knowledge only reflects properties of the local database, how to integrate this partial knowledge into the global knowledge in order to represent characteristics of the overall data collection remains a problem.

Against this background, we propose a simple but efficient algorithm called Transaction Encoding Algorithm (TEA), which does not disturb the structure of original data specified by users. The paper is organized as follows. Section 2 presents related work on minimizing communication cost. It is followed by section 3 which outlines our algorithm. Section 4 presents the experimental results of our implementation. Conclusion and future work and open research problems in the area in is explained in section 5.

2. PREVIOUS WORK

One of the problems with high-dimensional datasets of astronomy, biology, remote sensing, economics, and consumer transactions, is that, in many cases, not all the measured variables are important for understanding the underlying phenomena of interest. While certain computationally expensive novel methods [L.Breiman] can construct predictive models with high accuracy from high-dimensional data, it is still of interest in many applications to reduce the dimension of the original data prior to any modeling of the data. The goal of dimensionality reduction is to embed high-dimensional data samples in a low-dimensional space so that most of 'intrinsic information' contained in the data is preserved. Once dimensionality reduction is carried out appropriately, the compact representation of the data can be used for various succeeding tasks such as visualization, classification, etc. Dimensionality reduction methods can be grouped in various ways: (1) feature selection or feature extraction, (2) linear or nonlinear, (3) supervised or unsupervised, and (4) local or global. In feature selection, a subset of original features is selected in the end. In feature extraction, new features are extracted using some mapping (linear or nonlinear) from the original set of features. Linear methods [Deon Garrett, David A. Peterson, Charles W. Anderson, and Michael H. Thaut] use a linear mapping to extract new features from original features. Similarly, nonlinear methods Sammon's mapping [Sammon Jr., J.W.], locally linear embedding [S.T. Roweis and L.K. Saul], and ISOMAP [John Aldo Lee, Amaury Lendasse, Michel Verleysen] use a non-linear mapping to extract new features. A simple approach to dimensionality reduction is feature selection, which consists of determining an optimal subset of K features by exhaustively exploring all the possible combinations of D features. Most feature selection procedures use the classification error as the evaluation function. This makes exhaustive search computationally infeasible in practice, even for moderate values of D. The simplest method consists of evaluating the D features individually and selecting the K most discriminant ones, but it does not take into account dependencies among features. Classical dimensionality reduction techniques include unsupervised algorithms such as principal component analysis (PCA) technique [J.E. Jackson] [I.T. Jolliffe] and supervised algorithms such as linear discriminant analysis (LDA) [], canonical correlation analysis (CCA) [D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor], and partial least squares (PLS)[HIoskuldsson, A] [Garthwaite, P.H].

PCA is probably the most popular linear dimension reduction technique [J.E. Jackson] [I.T. Jolliffe] for computing lower-dimensional representations of multivariate data. It constructs a representation of the data with a set of orthogonal basis vectors that are the eigenvectors of the covariance matrix generated from the data, which can also be derived from singular value decomposition. By projecting the data onto the dominant eigenvectors, the dimension of the original dataset can be reduced with little loss of information.

Linear Discriminant Analysis (LDA) was the first statistical criterion for low rank linear separation, and it is still the most popular supervised linear feature extractor [D. Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. Duin]. LDA tries to maximize the dispersion among classes while minimizing the inner dispersion of each class, which is known as Fisher criterion. LDA provides a closed, eigen decomposition based solution to the maximum likelihood criterion in the homoscedastic case.

3. PROBLEM DESCRIPTION

This paper introduces an effective approach to the problem of dimension reduction which makes the reduction process very effective and also provides a much more compact representation than traditional dimensionality reduction techniques. The technique is based on sum of subset approach. Here we formulate dimensionality reduction as follows: Given a transaction set of data items S={x, y, z}, the power set of S, written

P(S)={{}, {x}, {y}, {z}, {x,y}, {x,z},{y,z},{x,y,z}), is the set of all subsets of S, including the empty set and S itself. The set of non-empty subsets of S may be denoted by P1(S). If set S is assumed as powers of 2, i.e. for example S = {2, 4, 8, 16}, then the power set

P1(S) = {{2}, {4}, {8}, {16}, {2, 4}, {2, 8}, {2, 16}, {4, 8}, {4, 16}, {8, 16}, {2, 4, 8, 16}}. An interesting and surprising property found in this power set is that the sum of the subsets are unique i.e. 2, 4, 8, 16, 6, 10, 18, 12, 20, 24, 30. This property is the motivation for choosing power set approach over the classical dimension reduction methods.

3.1 TEA Data Structure

Transaction encoding algorithm(TEA) transform a transaction into a small dimension transaction with all properties of its original form. In an encoded transaction items are represented by numbers. By this way, the new transaction is much smaller than the previous one and can be transmitted easily and so the cost of communication is reduced.

The data structure used in TEA consists of three parts as follows:

Let D = {T1, . . . , TK} be a database of customer transactions at a store. Each transaction, say Ti, is a collection of items purchased by a customer. A non-empty set of items is called an itemset. An itemset is denoted as I={i1,i2,…,in}, where each ij is an item from some ordered finite or interested items in store. Each transaction in the database is an itemset and is a subset of I (T □ I). Items in itemset I is stored in an m × n matrix C. Each column of C corresponds to an item. Item in each column in each row in C is given the

numeric value from the set {21, 22, 23,.,.,.,.,2n). For example,

the itemset I of 18 items is shown in Figure 1.

Item M	Item N	Item O	Item P	Item Q	Item R
Item G	Item H	Item I	Item J	Item K	Item L
Item A	Item B	Item C	Item D	Item E	Item F

Figure 1: An Example itemset I of 18 items

The numeric value assigned to items in row 1, 2 and 3 is 21, 22, 23, 24, 25, 26.

The vector M consist the last or the largest item in each row. The largest item in each row of C is stored in M as shown Figure 2.

Item F	Item L	Item R

Figure 2: Vector M for itemset I

The matrix E records the reduced version of transactions in D. The number of columns in the row in E is equal to the number of rows in C.

3.2 Algorithm

Algorithm DIM, in figure 3, computes the dimension (i.e. number of rows and columns) of matrix C where the dataitems of I is stored. The maximum number of column in a row is restricted to 14 so that the sum of the row in C does not exceed the value 32,767. Whenever finding an exact dimension is not possible the algorithm computes a dimension for C with the minimum number of unused columns in the last row.

```
DIM (DD)
Input: DD size a transaction I
Output: number of rows and columns
        required to store dataitems in I

 CD = True
 PrevDiff = DD; Sign = 1
IF SQRT(DD) > 14 THEN
   Numerator = 14; Inc=-1
  ELSE
    IF ISODD(DD) THEN
      Numerator = 3; Inc=2
    ELSE
      Numerator = 2; Inc=1
    END
  END

 WHILE CD
   IF MOD(DD, Numerator) > 0 THEN
     IF (Numerator* (DD/Numerator + 1) – DD)
          < PrevDiff THEN
      P1 = Numerator; P2 = Numerator *
            (DD/Numerator + 1)
        PrevDiff = Numerator *
                       (DD/Numerator + 1) – DD
        Numerator = Numerator + inc
      ELSE
        CD = False
      END
    ELSE
      CD = True
    END
  END
  IF (P1<14) AND (P2<14) AND (P2< P1) THEN
    Temp = P1;  P1 = P2;  P2 = Temp
  END
  RETURN (P1 , P2)
```

Figure 3 Algorithm DIM

Algorithm LOAD, figure 4, stores dataitems of I in C and the last element in each row in the respective column in M.

```
LOAD (C, M, T, m, n)
 k = 1;
  FOR i = 1 to m
    FOR j = 1 to n
      C(i,j) = T(k)
      k=k+1
    END

    M (i) = T(k-1)
 END
```

Figure 4 Algorithm LOAD

Algorithm TEA, figure 5, takes database D and matrix C as inputs. Picks transaction after transaction in D. From each transaction items are taken one after another and the cell (row i and column j) in C whose dataitem matches is identified. Once the cell in C identified value 2j is added to the previous value in ith column in E. E is inserted in RD.

```
TEA (D, I, RD)

 Input: Database D, Transaction I (Ordered).
        Every transaction in D is subset of I

 Output: Code matrix C & Reduced database RD

   S = SIZE (I)

    IF ISPRIME(S) THEN S=S+1;
   (m, n) = DIM (S)
    Create arrays E & M with m columns
   Create matrix C with m rows and n columns
   Call LOAD (C, M, T)

    FOR every transaction ti  in D
          Initialize E with Zeros
     FOR every dataitem dk in ti
       p = 1
       WHILE  dk > M(p)
         p = p + 1
       END
       q = 1
       WHILE  C(p, q) > dk
         q = q + 1
       END
       E(p) = E(p) + 2q
```

```
            Inserted E in RD

        Initialize E with zeros.
    END
    END
```

Figure 5 Transaction Encoding Algorithm (TEA)

Once the reduced database RD is transferred to the destination site the following algorithm is implemented to decode and store the transaction in the database at that site.

```
FOR every transaction ti  in R
    FOR every dataitem dk in ti
        n = 1
        WHILE  dk > 2n
            n = n + 1
        END
        n = n-1
        WHILE  (dk - 2n) > 0
            tk = C(k, n)
            n = n - 1
        END
    END
END
```

Figure 5 Decoding Algorithm

Table 1: The transactional database D

For the explanation of the algorithm, we will use the following example. To reduce the dimension of itemsets from transactional database D (see Table 1) first, finite items or interested items in store, say

I = {bread, bun, burger, butter, cheese, egg, fruit bread, honey, jam, milk, sauce, sugar,

pizza	sauce	sugar	sweet bun	
egg	fruit bread	honey	jam	milk
bread	bun	burger	butter	cheese

Table 2: Representation of item set I in C

The reduced form of transactions in D is given in table 3.

34	50	10
58	16	02
18	16	18
56	00	00
38	28	10

Table 3: Reduced database RD

Items
jam, bread, cheese, egg, milk, sugar, pizza
bread, butter, cheese, jam, burger, pizza
butter, jam, sauce, sweet bun
butter, cheese, burger
bread, jam, cheese, honey, sugar, fruit bun, pizza

4. EXPERIMENTAL RESULTS

As shown in Table 4, we use these randomly generated sample data to simulate the process of TEA algorithm. The data in the table indicate that different customers purchase different types of goods, the first column data is the serial number of customers, and the rest the commodities in the market basket. We simulated the algorithm for 100 actions for transactions with 150 commodities, 100 commodities, 50 commodities 15 commodities and 5 commodities respectively. The results of our simulated study are shown in table 5.

Cust .No:	Items in Basket							
	1	2	3	o o o o	o o o o	o o o o	149	150
1	Biscuit	Bread	Cheese	o o o o	o o o o	o o o o	Tooth Paste	Face Powder
2	Bread	Bun	Cake	o o o o	o o o o	o o o o	Candle	
o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o
o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o
o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o	o o o o
100	Bread	Jam	Cheese					

Table 4: Sample data to illustrate the market basket transaction

S.No.	Number of Commodities in a transaction	Time taken for Communication	
		Transaction as such	After reducing Transaction dimension by TEA algorithm
1. T	150	.30 Seconds	0.75 Milli Seconds
2.	100	.18 Seconds	0.60 Milli Seconds
3.	050	.08 Seconds	0.37 Milli Seconds
4.	025	.06 Seconds	0.18 Milli Seconds
5.	010	.1Mmilli Seconds	0.05 illi Seconds

Table 5: Experiment results

5. CONCLUSIONS

There are two important variables that influence the communication time: the number of transactions and the number of items in a transaction. The dimension reduction algorithm, TEA, is dependent on the number of transactions. The number of items in a transaction is irrespective for the TEA algorithm because every transaction is represented by 15 column whether a transaction contains 15 items or above it or below it. Our experiments showed that transactions with items and above show encouraging result. Next we planned to work to improve algorithm to handle Reuters Corpus Volume 1 (RCV1) data set which contains over 800,000 documents (300,000-dimension). However, when the database contains numeric values (i.e. iris dataset, stock dataset etc.) our TEA algorithm does not support. Based on the encouraging observed results, as future work, we intend to improve the algorithm to reduce size of transactions holding numeric values.

6. REFERENCES

[1]. Y. Akbas, C. Takma Canonical correlation analysis for studying the relationship between egg production traits and body weight, egg weight and age at sexual maturity in layers Czech Journal of Animal Science, 50, pp.163–168, 2005 (4).

[2]. L. Breiman., Random forests, Technical report, Department of Statistics, University of California, 2001.

[3]. Deon Garrett, David A. Peterson, Charles W. Anderson, and Michael H. Thaut, Comparison of Linear, Nonlinear, and Feature Selection Methods for EEG Signal Classification IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 11 Issue. 2, pp.141 – 144, 2003

[4].Garthwaite, P.H., 1994. An interpretation of partial least squares. Journal American Statistical. Association. 89, pp.122–127, 1988.

[5]. A. J. Guarino, A Comparison of First and Second Generation Multivariate Analyses: Canonical Correlation Analysis and Structural Equation Modeling 1, Florida Journal of Educational Research, 2004, Vol. 42, pp. 22 – 40 22

[6]. D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, Canonical correlation analysis: An overview with applications to learning methods, Neural Comput., vol. 16, pp. 2639–2664, 2004.

[7]. HIoskuldsson, A., PLS regression methods, Journal of Chemometrics. 2, 211–228. 1988.

[8]. J.E. Jackson. “A User's Guide to Principal Components”. New York: John Wiley and Sons, 1991.

[9]. John Aldo Lee, Amaury Lendasse, Michel Verleysen Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis Neuro computing 57 (2004) 49 – 76

[10]. I.T. Jolliffe. “Principal Component Analysis”. Springer-Verlag, 1986.

[11]. Lamersdorf, M. Merz (Eds.), Trends in Distributed Systems for Electronic Commerce, Lecture Notes in Computer Science, vol. 1402, Springer-Verlag, Berlin Heidelberg New York, June 1998

[12]. Philip Machanick, A distributed systems approach to secure Internet mail, Security,Volume 24, Issue 6, September 2005, Pages 492-499

[13]. D. Ridder, O. Kouropteva, O. Okun, M. Pietikainen, and R. Duin, “Supervised locally linear embedding,” in Proc. Artif. Neural Netw. Neural Inf. Process., 2003, pp. 333–341.

[14]. S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science 290: 2323-2326.

[15]. Salah Aidarous Stephen B. Weinstein, Distributed Systems for Telecommunications IEEE Network January/February 1994

[16]. Sammon Jr., J.W., “A nonlinear mapping for data structure analysis” IEEE Transactions on Computers, C-18, 401-409. 1969

[17]. Syed Zahid, Hassan Zaidi, Syed Sibte Raza Abidi and Selvakumar Manickam. Distributed Data Mining From Heterogeneous Healthcare Data Repositories: Towards an Intelligent Agent-Based Framework, Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002)

[18]. Talia, D.“Grid-Based Distributed Data Mining Systems, Algorithms and Services, 9th International Workshop on High Performance and Distributed Mining, Bethesda April 22, 2006

[19]. J. J. Verbeek, S. T. Roweis, and N. Vlassis. Nonlinear CCA and PCA by alignment of local models. In Advances in Neural Information Processing Systems 16, 2000

[20]. Wu-Shan Jiang, Ji-Hui Yu., Distributed Data Mining on the Grid, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.