DEALING WITH THE PROBLEM OF OVERLAPPING CLUSTERS PRIOR TO DATAMATCHING USING CLUSTERING TECHNIQUES

G.SOMASEKHAR
SCSE,VITUniversity,Vellore,India
gidd.somasekhar2014@vit.ac.in

K.KARTHIKEYAN
Associate Professor, SAS,VITUniversity,Vellore,India
k.karthikeyan@vit.ac.in

ABSTRACT: Datamatching and Bigdatamatching are current interesting research topics which are elaborately discussed in several research studies.When the initial dataset is huge,the best suitable technique for datamatching is canopy clustering.But this clustering technique assigns the records or datapoints to multiple overlapping clusters which introduces redundant pair comparisons when similar records share more than one cluster. An approach has been proposed that avoids such redundant pair comparisons prior to datamatching phase.Sofar a very few research studies are carried out on the redundant-free similarity computation.The new algorithms proposed in the approach serve two major purposes.i)Explaining an incremental procedure for creation of clusters which is more useful in the process of datamatching.ii)Redundant-free pair selection for datamatching.The approach does not require post-processing of final result.Compared to Kolb's approach[3],our approach reduces the complexity in the datamatching phase.

KEYWORDS: Bigdata,Bigdatamatching,datamatching,datapoint,record,canopyclustering,redundancy,similarity.

INTRODUCTION

Modern techniques for data generation,data collection and data storage made huge amount of data available to the users.These huge chunks of data are usually stored as continuous text such as personal demographic data,bibliographic information,phone and mailing lists.This data must be integrated in order to enrich the dataquality and draw valuable conclusions.Integration of such data involves two major problems.i)Structural heterogeneity ii)Syntactical heterogeneity.The former takes place when the sourcedata has no common schema.In this case,schema reconciliation techniques[21,22,23] are used to solve the problem.The latter takes place when syntactically different records refer to same realworld entity.Datamatching is a solution to this problem for which a specific tool called Dedoop can be used.Prior to deduplication with dedoop,a pair of record collections those are going to be matched should be selected from the pool of records and given as input.For grouping records those related to similar entity,efficient clustering techniques like canopy clustering can be used.The Kolb's approach[3] implements the dedoop tool and the mapreduce phase of deduplication is more complex involving unnecessary condition checkings in every reducer process.Our approach uses clustering and redundant-free pair selection techniques prior to deduplication(i.e.,datamatching) to reduce the complexity in Kolb's approach.

RELATED WORK

Fraud detection and Anamoly detection in various fields like finance,science and so on is the main advantage of datamatching.Many research works were carried out in the area of datamatching,which was referred to with many names,such as,e.g.,Record Linkage(Fellegi and Sunter[13]1969;Winkler[14] 1990),Deduplication(Sarawagi and Bhamidipaty[17] 2002),Entity-Name Matching(Cohen and Richman[16] 2002).Efficient clustering techniques for bigdatamatching were proposed by McCullum(2000)[2],Chaudhuri(2005)[26]and Anderberg[20] .A solution was given by McCullum for grouping records in canopies and limiting the expensive similarity computation with in each canopy.A two-phase approach was proposed by Chaudhuri,based on nearest neighbours computation.

Several indexing techniques were summarized for datamatching by Peter christen[1].Fast blocking methods for datamatching were proposed by Baxter,Christen and Churches[7].Iterative datamatching was mentioned by Bhattacharya and Getoor[25]. A new datamatching method was explained by Hu,Li and Feng[12] which can be applied when the source data is in the form of graphs.Some mapreduce based techniques for datamatching were proposed by Kolb,Thor and Rahm in [5,9].A few research studies were performed in the area of canopy based bigdatamatching[3].Canopy clusteringwas explained by McCullum,Nigam and Ungar[2] in detail.An incremental clustering algorithm for deduplication was proposed by Costa,Manco and Ortale[6]. The problem of overlapping clusters was identified by Kolb[3] and a mapreduce based solution was given to it using the dedooptool.The approach proposed in this paper solves the same problem in the pre-dedoop phase with minimal clustering overhead.

PROBLEM DOMAIN AND DEFINITION

Data from multiple sources should be integrated to enrich data quality and to reduce data acquisition cost which facilitates and improves data analysis task.Schema mapping,data fusion and data matching are the different methods evolved in this process where the last method is focused in this paper. The records from initial pool of data may refer to different entities such as customers,patients,employees,students,travelers or tax payers.Grouping of these records pertaining to similar entities is called “data matching”.When the initial data grows to huge amounts,the same process is called Bigdata matching.Canopy clustering is a datamining technique used for Bigdata matching.Creation of canopy clusters with sample dataset is shown in Fig.1.

In this clustering method, ‘n’ number of datapoints or records are selected at random from huge dataset and they are named as canopy centroid points.Two threshold values namely loose threshold(t_l) and tight threshold(t_t) are also determined.The canopy cluster formation is having two stages.In the first stage a cheap similarity metric is taken and in the second stage expensive similarity metric is taken.First stage is advantageous because with minimum computation clusters are formed.Second stage involves maximum computation overhead.

For all canopy centroid points the similarity measure has been calculated with all the remaining points in the whole dataset using cheap similarity metric.Canopies are formed using loose threshold t_l.For each canopy,the records having it’s similarity value with that centroid point $<=t_t$ are removed from the dataset and a canopy cluster is formed.Similarly ‘n’ canopy clusters are formed for ‘n’ canopies.In the latter stage,expensive similarity metric is applied to each cluster and

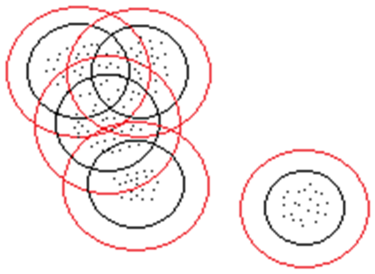

Figure 1.creation of canopy clusters(red circles denote canopies,black circles denote their clusters and dots denote data points or records)

the final set of candidate record pairs is achieved.This is our required data from the initial huge dataset.

Advantages of standard Canopy Clustering:
a)It is suitable for Bigdata applications when initial datasets are very huge.
b)Data sets can be huge in three ways.

- There can be huge number of records in the dataset.
- Each record can have many features.
- There can be many clusters.

This technique is efficient when the problem is big in all these three ways at once.(i.e.,millions of datapoints,many thousands of features and many thousands of clusters.)

c)Canopy clustering is most efficient method of clustering.

Though Canopy clustering is more appropriate for Bigdatamatching,it gives rise to the problem of overlapping clusters[3].This problem causes redundant pair comparisons when similar datapoints or records share more than one cluster.

PROBLEM STATEMENT

In the datamatching process, each cluster is taken at a time and subjected to expensive pairwise similarity computation.As each cluster has overlapping regions(sub-clusters),these regions may be repeated while computing similarities.
Given 'n' number of clusters those are initially created from a bigdata pool by using canopy clustering.Then 'k' number of subclusters can be generated from 'n' clusters,Where a subset of 'm'subclusters from 'k' subclusters may be repeated 'v' times or less than 'v'times.'v' is the maximum degree of redundancy. This redundancy should be avoided in order to reduce the total datamatching time.

PROBLEM SOLVING AND INNOVATIVE CONTENT

In the ultimate cluster set, each cluster is divided into sub-clusters where the component subclusters of a cluster when collected form the total cluster. Now the target data pool is the sets of subclusters of all the clusters.For each cluster, a pair of subclusters is selected for data matching with dedoop. In each such pair,a same subcluster may be taken twice to form the pair which may be treated as deduplication .(i.e.,grouping records from the same subcluster or database).Finally a set of pairs is obtained.From this set of pairs,the repeated(redundant) pairs are deleted. Finally a resultant set of unique linkage pairs is achieved which is the input to the datamatching with a specific datamatching tool “dedoop”.Parallel dedoop executions can be carried out to get the datamatching result as early as possible. The following are the algorithms for solving the problem.

```
CREATE-CLUSTERS(C,D,d,μ)
Input: A pool of data D;
        Initial cluster set C;
        Similarity threshold μ;
        New set of datapoints or records d;
Output: A cluster set $C^1$ of  D U d.
1:    $D^1 \longleftarrow D$; $C^1 \longleftarrow C$;
2:    Let $C^1=\{C_1,C_2,\ldots,C_n\}$ and $d=\{d_1,d_2,\ldots,d_m\}$;
3:    for  i=1....m do
4:        CSF($d_i$,$C^1$,μ);
5:    end for
```

```
CSF($d_i$,$C^1$, μ)
CSF1:    $CC^1$= Ø ; CMI($d_i$)= Ø;
CSF2:    for  k=1....n do
CSF3:        if SIM($d_i$,Centroid($C_k$))>= μ then
CSF4:            CMI($d_i$) $\longleftarrow$ CMI($d_i$) U “$C_k$$”;
CSF5:        $CC^1 \longleftarrow CC^1$ U {$C_k$};
CSF6:        end if
CSF7:    end for
CSF8:    $k^1$= sizeof($CC^1$);
CSF9:    for J=1…$k^1$ do
CSF10:        $C_J \longleftarrow C_J$ U {$d_i$};
CSF11:   end for
CSF12:   if  CMI($d_i$)= Ø and $CC^1$= Ø then
CSF13:       create a new cluster $C_{n+1}$= {$d_i$};
CSF14:       $C^1 \longleftarrow C^1$U {$C_{n+1}$};
CSF15:       n=n+1;
CSF16:   end if
CSF17:   FORM-SUBCLUSTERS($C^1$);
```
(Note: $C_k$and C_J are member clusters of the cluster sets $C^1$and CC^1 respectively.
CMI(d_i) extracts the cluster membership information of datapoint d_i .
SIM(d_i,Centroid(C_k)) computes the Jaccard similarity distance between data points d_i and Centroid(C_k).
Centroid(C_k) extracts the centroid point of the cluster C_k.
CSF denotes Cheap Similarity Function andULP denotes Unique Linkage Pair).

```
FORM-SUBCLUSTERS($C^1$)
Input: Initial sub-clusters set SC= Ø;
```

```
Output: Final set of SC Which can be taken as input for FORM-ULPS( );
F1:   for i=1…n do
F2:      for each data point $P_J$ in $C_i$ do
F3:         if SC= Ø then
F4:            SC ← SC U $SC_1$;
F5:            CMI(Centroid($SC_1$))=CMI($P_J$);
F6:            $SC_1$ ← $SC_1$U {$P_J$};
F7:         end if
F8:         for k=1…Sizeof(SC) do
F9:            t=1;
F10:           if CMI(Centroid($SC_k$))=CMI($P_J$) then
F11:              $SC_k$ ← $SC_k$ U {$P_J$};
F12:              t=2;
F13:           end if
F14:           if t=2 then
F15:              go to F23;
F16:           end if
F17:        end for
F18:        if t=1 then
F19:           last=Sizeof(SC);
F20:           CMI(Centroid($SC_{last}$))=CMI($P_J$);
F21:           $SC_{last}$ ← $SC_{last}$ U {$P_J$};
F22:        end if
F23:     end for
F24:  end for
F25:  FORM-ULPS(SC);
```

```
FORM-ULPS(SC)
Input: empty set ULP;
Output: Non-empty sets ULP and SC which can be taken as input for parallel Dedoop
applications.
FU1: N=Sizeof(SC);
FU2: for l=1 … N do
FU3:    UL1=CMI(Centroid($SC_l$))^CMI(Centroid($SC_l$));
FU4:    b=REDUNDANCY-CHECK(UL1);
FU5:    if b=0 then
FU6:       ULP ← ULP U {UL1};
FU7:    end if
FU8:    for p=l+1…N do
FU9:       if CMI(Centroid($SC_l$))=CMI(Centroid($SC_P$)) then
FU10:         go to FU8;
FU11:      end if
FU12:      if CMI(Centroid($SC_l$)) ∩ CMI(Centroid($SC_P$))≠ Ø then
FU13:         UL2=CMI(Centroid($SC_l$))^CMI(Centroid($SC_P$));
FU14:         b=REDUNDANCY-CHECK(UL2);
FU15:      end if
```

```
FU16:     if b=0 then
FU17:         ULP ← ULP U {UL2};
FU18:     end if
FU19:   end for
FU20: end for
```

```
REDUNDANCY-CHECK(UL)
R1:  SZ=sizeof(ULP);
R2:  for q=1…SZ do
R3:     if UL=ULPqthen
R4:        return 1;
R5:     end if
R6:  end for
R7:  return 0;
```

The algorithm CREATE-CLUSTERS is responsible for generation of new clusters. This algorithm satisfies the incrementality requirement. We intend to cope with the clustering problem in an incremental setting. 'D' is the clustered initial data pool. Data points set 'd' must be integrated within a previously clustered data pool 'D'. Each data point in d must either be associated with atleast one cluster in C^1 or lead to creation of a new cluster. If a data point in d is associated with multiple clusters, then it's cluster membership information CMI is updated to reveal all the clusters related to that data point.For example if a datapoint $d_i$is associated with multiple clusters namely, C_1,C_2 and C_3then it's CMI is updated to C_1\$$C_2$\$$C_3$\$.The Jaccard similarity distance is taken to find whether a datapoint d_i relates to a cluster C_k.

Jaccard similarity distance=SIM(d_i,Centroid(C_k))=(d_i∩ Centroid(C_k))/(d_i U Centroid(C_k))   (1)

If this distance is greater than or equal to the selected threshold 'μ' then the data point $d_i$is associated with the cluster C_k.

The algorithm FORM-SUBCLUSTERS generates component subclusters for each cluster. Somesubclusters may be repeated in the final subcluster set SC due to overlapping clusters.
The algorithm FORM-ULPS generates possible Unique Linkage Pairs of subclusters.For deduplication within each subcluster, samesubcluster is repeated in the pair.The remaining unique linkage pairs are formed by checking the condition of common cluster. When both subclusters in the pair belong to common cluster then only that pair is a valid pair.

The algorithm REDUNDANCY-CHECK checks whether any pair is repeated. If a pair is repeated then it skips that pair. If a pair is a new one then it is added to the set ULP. Let us once examine a sample overlapping clusters scenario for understanding.

The scenario of sample overlapping among three canopy clusters is shown in Fig.2.For simplicity subclusters and ULPs are named in Table 1. Let C_1^1,C_2^1,C_3^1 denote non-overlapping subcluster regions of C_1,C_2 and C_3respectively.The above scenario is a best example to understand the problem. There may be several overlapping clusters in realtime. Finally, the sets ULP and SC are the results of the proposed algorithms. These can be taken as input for the parallel Dedoopapplications.Each pair in the set ULP is considered as input to the data matching tool Dedoop where each subcluster from that pair is extracted from the set SC. The implementation of

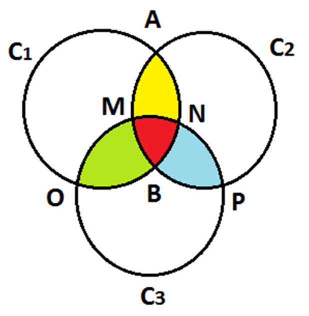

Figure 2. Sample overlapping among three canopy clusters C_1,C_2,C_3

Table 1.sample canopy clusters, their subclusters and unique linkage pairs for fig.2

Canopy cluster	subclusters	Corresponding Unique Linkage subcluster pairs
C_1	C_1^1,AMN, MBN,MOB	C_1^1-C_1^1, AMN-AMN, MBN-MBN, MOB-MOB, C_1^1-AMN, C_1^1-MBN, C_1^1-MOB, AMN-MBN, AMN-MOB, MBN-MOB
C_2	C_2^1,AMN, MBN,NBP	C_2^1- C_2^1, NBP-NBP, C_2^1-AMN, C_2^1-MBN, C_2^1-NBP, AMN-NBP, MBN-NBP
C_3	C_3^1,MOB, MBN,NBP	C_3^1- C_3^1, C_3^1-MOB, C_3^1-MBN, C_3^1-NBP, MOB-NBP

Dedoop tool is explained by Kolb,Thor and Rahm[4].The total process does not need post-processing of the final data matching results. The Kolb's approach[3] is a map reduce based approach and involves unnecessary condition checkings for finding redundant matching pairs in each reducer process. Our approach avoids redundancy with minimal clustering overhead excluding the unnecessary condition checking in each reducer process of the Kolb's approach.

RESULTS AND COMPARISON

Table 2.execution times for different data pools

Different sizes of data pools	Time to form clusters and ULPs(in minutes)
1k	12
2k	25
3k	38
4k	51
5k	64

The algorithms are checked for different datasets such as bank credentials data set and medical diagnosis datasets to group records pertaining to same bank and same disease respectively. The algorithms execution times corresponding to different data pools are tabulated in Table 2.Assume all clusters are uniformly distributed.

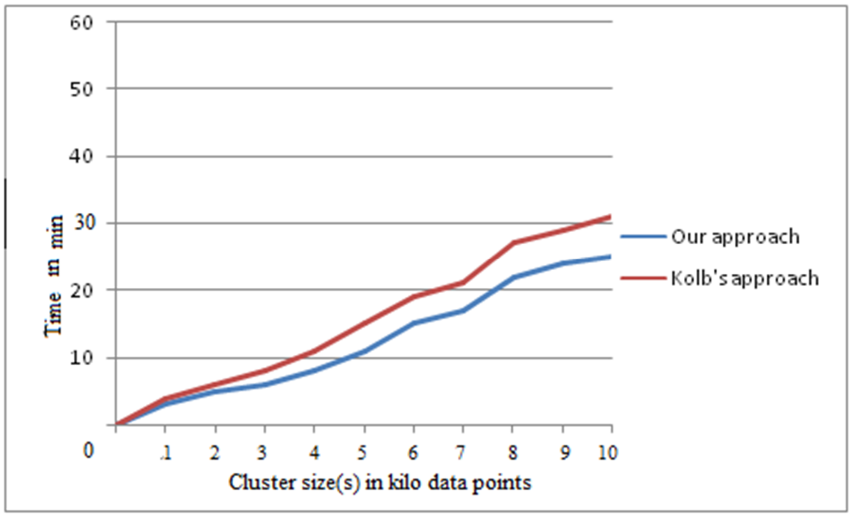

Figure 3. Execution times comparison for different degrees of redundancy

The approach is compared with Kolb's approach as shown in Fig.3 for different degrees of redundancy.100k of data points are taken in the data pool and it is partitioned into 100 clusters. We successively increase the initial cluster size by 1000 data points up to s=10,000.This results in a pair overlap of 25% for s=2000 and 81% for s=10,000.In the Kolb's approach checking is made for each pair in a cluster whether it has to be evaluated or redundant. Our approach eliminates this overhead.

CONCLUSION

This paper has presented a new approach to avoid redundant similarity computations caused by overlapping clusters. The experimental results showed that the proposed approach is better when compared to Kolb's approach. It does not require post-processing of final results.

REFERENCES

Peter Christen,”A survey of indexing techniques for scalable record linkage and deduplication”,*IEEE Transactions on Knowledge and Data Engineering,IEEE,*vol.24 Issue 9, pp. 1537-1555,September 2012.

Andrew McCallum,Kamal Nigam and Lyle H.Ungar,”Efficient Clustering of High Dimensional Datasets with application to Reference Matching”, *In Proc. of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 169-178,2000

Lars Kolb, Andreas Thor and Erhard Rahm”Don’t Match Twice:Redundancy-free Similarity Computation with MapReduce”,*DanaC’13 Proceedings of the Second Workshop on Data Analytics in the Cloud,ACM,*pp. 1-5,2013.

Lars Kolb, Andreas Thor and Erhard Rahm,”Dedoop:Efficient deduplication with Hadoop”,*In Proceedings of the VLDB endowment*,vol.5,No.12,pp. 1878-1881,2012.

Lars Kolb,Andreas Thor and Erhard Rahm.”Multipass Sorted Neighborhood Blocking With MapReduce”, *Journal:Computer Science - Research and Development*,vol. 27 Issue 1,pp. 45-63,February 2012.

Gianni Costa , Giuseppe Manco and Riccardo Ortale,”An incremental clustering scheme for data de-duplication”,*Journal:Datamining and Knowledge Discovery,*vol.20,Issue 1,pp. 152-187,springer,January 2010.

Baxter, R., Christen, P. and Churches, T.,”A comparison of fast blocking methods for record linkage”, *In Proc. Of ACM SIGKDD’03 workshop on Data Cleaning,Record Linkage and Object Consolidation,*pp. 27-29,2003.

P. Christen and K. Goiser,”Quality and Complexity Measures for Data Linkage and Deduplication”, *Quality Measures in Data Mining,* ser. *Studies in Computational Intelligence, F. Guillet and H. Hamilton, eds.,* vol.43, Springer, pp. 127-151, 2007.

Kolb, L., Thor, A. and Rahm, E.,”Parallel sorted neighborhood blocking with mapreduce”, *In Proc. BTW’11,*pp. 45-64,2011.

Jingtao Zhou, Mingwei Wang, Han Zhao, Shusheng Zhang, and Chao Zhang,”Concept Capture Based On Column Matching and Clustering”,*SKG, 2005, Semantics, Knowledge and Grid, International Conference on, Semantics, Knowledge and Grid, International Conference on 2005,* pp. 71, IEEE,2006.

Harris T. Lin, Sanghack Lee, Ngot Buiand VasantHonavar,”Learning Classifiers from Distributional Data”,*2013 IEEE International Congress on Big Data,*pp. 302-309,IEEE,2013.

Huiqi Hu, Guoliang Li and JianhuaFeng,”Fast Similar Subgraph Search with Maximum Common Connected Subgraph Constraints”, *2013 IEEE International Congress on Big Data.*pp. 181-188,IEEE,2013.

Fellegi IP, Sunter AB,A theory for record linkage. J Am Stat Assoc 64:pp.1183–1210,1969

Winkler WE,”String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage”, *In: Proceedings of the section on survey research methods, American Statistical Association,* pp. 354–359,1990.

Cohen W and Richman J ,”Learning to match and cluster entity names”, *In: Proceedings of the ACM SIGIR workshop on mathematical/formal methods in information retrieval,* pp. 13–18,2001.

Cohen WW and Richman J ,“Learning to match and cluster large high-dimensional data sets for data integration*”, In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining,* pp. 475–480,2002.

Sarawagi S and BhamidipatyA,"Interactive deduplication using active learning", *In: Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining,* pp. 269–278,2002.

W. E.Winkler, "Methods for evaluating and creating data quality", *Journal: Information Systems - Special issue: Data quality in cooperative informationsystems,*vol. 29 Issue 7, pp. 531-550, Elsevier Information Systems, October 2004.

W. E. Winkler, "Overview of record linkage and current research directions,*" US Bureau of the Census, Tech. Rep.,* 2006.

M. R. Anderberg," Cluster Analysis for Application",*Academic Press, 1973.*

Agichtein E and GantiV , "Mining reference tables for automatic text segmentation", *In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining,* pp. 20–29,2004.

Monge AE and ElkanCP,"Automatic segmentation of text into structured records", *In: Proceedings of the ACM SIGMOD conference on management of data,*pp. 175-186,2001.

Cesario E, Folino F, Locane A, Manco G and OrtaleR,"Boosting text segmentation via progressive classification", *J KnowlInfSyst 15(3):*pp. 285–320,2008.

Ananthakrishna R, Chaudhuri S and GantiV,"Eliminating fuzzy duplicates in data warehouses",*In:Proceedings of the international conference on very large databases,* pp. 586–597,2002.

Bhattacharya I and GetoorL,"Iterative record linkage for cleaning and integration", *In: Proceedings of the SIGMOD workshop on research issues on data mining and knowledge discovery,* pp. 11–18,2004.

Chaudhuri S, Ganti V and MotwaniR ,"Robust identification of fuzzy duplicates", *In: Proceedings of the international conference on data engineering,* pp. 865–876,2005.