Big Data and Anonymity

# Introduction

This report describes the anonymization models specifications in big data. Any developed anonymization model should distinguish between big data and conventional data. Most anonymization models were developed for conventional data, where a limited number of records with limited computation costs are the main features. When data grows up, then a need for up scaling the anonymization models to reduce the computation costs and increase security. Larger size of data may increase the number and levels of user’s access. Hence, security and performance are the two major concerns in big data.

# Anonymization Models Specifications

Any big data anonymization model should consider some specifications and facts in big data:

## Equivalency Increase

Theorem 1: since k-anonymity value is constant, and doesn’t change with the increase number of records. Hence, the percentage of equivalent records proportioned extrusive with the increase number of records.

Proof:

We need to prove that the increase number of records can help the least frequent attributes to gain the equivalency.

Let a domain of data D with a total number of records N, and the k-anonymity = ǩ. Suppose that D contains m qid’s and one sensitive attribute. Based on k-anonymity model, the probability of records appearances in the domain D is described as:

(1)

Where P[qid] is the probability for each qid, and P[S] is the sensitivity probability.

There are correlations of n = 1 / β appearances for qid values, where n denotes the number of appearances. This means that any qid record must be one of the n appearances. Therefore, any record equal or higher than n would be equivalent and repeated. This is true if we suppose that each probability appears only once, hence, the minimum N must be equal to ǩ / β:

(2)

Referring to k-anonymity, the equivalency is defined as the number of equivalent records q ≥ ǩ for each appearance, for instance, if ǩ =5, then each distinguished record must appear five times in N before gaining the k-anonymity. However, not all records are equivalent in N. Some qid records may appear less frequently, and they probably need a larger N to reach the k-anonymity equivalency. Based on equation 2, we can generalize the minimum value of N to include the number of appearances ñ:

(3)

*Where* ñ d*enotes the number of appearances,* ñ ⊂ n

Equations 2 and 3 supposed that each record has equal number of appearance to the other records. However, this is not a common case. Hence, some records are less frequently appearing than others, which makes some records reach the equivalency, while other records fail. However, equation 3 describes only one scenario. Nevertheless, any scenario should consider the appearance value ñ, and the three constants of (ǩ, β, n). The variable ñ may carry one of two scenarios; stable, or increase, but the variable ñ never decreases. However, in real world, the value of ñ usually increases, while the other scenario is possible in limited cases.

Moreover, the equivalency percentage Q can be calculated by dividing the number of equivalent values over N:

Where Q is the equivalency percentage, and q is the frequency of equivalent values in N records

Based on equation 4, there is a direct proportion between the frequency and N. Also, equation 3 shows that Nmin will be higher, if there is any increase in ñ.

In the conclusion, equations 3 and 4 prove that a higher number of records, will lead to a higher number of q, and this is true in both of ñ stability or increase cases. Equation 4 is more general, and doesn’t depend on equivalency frequency ñ, which proves that N α q.

Example: in adult data we assign 3 qid values with {age, education, sex}. Let us investigate the above theorem for a small, medium and large size of records.

### Experiment 1:

Number of records N =10,000

Ǩ = 10

P[age] = P[1-100]=0.01

P[education] = P[Y5-6, Y7-8, Y9, Y10, Y11, Y12, HS-grade, Some-college] = 0.125

P[sex] = P[Male, Female] = 0.5

P[S] = P[<=50K, >50K] = 0.5

β = 0.01 X 0.125 X 0.5 = 0.000313 ≈ 0.0003

The n = 1/β value describes the number of appearances. n =1/β = 3195 probable appearances

While the appearances in 10,000 records was ñ = 1741, which presents around 50% of the probable appearances.

The number of equivalent records in k-anonymity, where Ǩ = 10 is q=6272, which presents around Q=60% of the total number of records

### Experiment 2:

The n=1/β value describes the number of appearances. n = 3195 probable appearances

While the appearances in 20,000 records was ñ = 2196, which presents around 69% of the probable appearances.

The number of equivalent records in k-anonymity, where Ǩ = 10 is q=14828, which presents around Q=75% of the total number of records.

### Experiment 3:

The n=1/β value describes the number of appearances. n = 3195 probable appearances

While the appearances in 32,561 records was ñ = 2498, which presents around 78% of the probable appearances.

The number of equivalent records in k-anonymity, where Ǩ = 10 is q=26846, which presents around Q=82% of the total number of records.

### Experiments results

The three experiments show an increase in equivalency percentage Q. Figure 1 shows the increase of Q between [10,000 -32561] records.

Figure 1, Equivalency percentage increase in Adult data

Also, the three experiments show an increase value of ñ, as shown in Figure 2.

Figure 2, Number of appearances increase

## The Gained Information Decrease.

Data is usually disturbed after anonymization, and information gained is affected accordingly. The hidden data on anonymization may affect the user’s queries result. The impact of anonymization can be measured by Laplace equations using InfoGain and Scores. The scores= InfoGain / AnonyLoss.

Current anonymization models were designed for conventional data. Therefore, the grouping process is the major task on anonymizing data. Data is usually grouped into equivalent records, known by compression. This technique supports masking operations. As explained before, data equivalency increases extrusive with the size of data, which will result a larger group of equivalent records. This will end up with a large data loss; if the equivalent records were not handled and anonymized properly.

The current anonymization processes start with a complete generalization, then grouping, and finally specialization. This technique provokes a large size of equivalency, and involves a useless anonymization process for unneeded equivalent data.

Eventually, grouping data based on equivalency is an acceptable technique if it was handled properly on data masking. This evolves a better technique on masking data by skipping the equivalent data masking, and applying masks on semi-equivalent or non-equivalent data only.

## The Parallel Distributed Environment

Big data is handled by a parallel distributed environment. Anonymization models should consider this on designing masking processes, by splitting tasks into sub-tasks, and distributing them among multi-computers. This technique is able to carry out multi-tasks at the same time to cope with the massive data volume.

Splitting tasks should affect the information gained. In another word, splitting data into small junk of data may negatively affect the information gained. This is inasmuch the previous equivalency increase fact. For instance, a data of 10,000 record will be extensively anonymized more than a data of 30,000 records. This is because of the lower equivalency in the lower number of records.

However, parallel distributed environments have limited size of handling data on each time retrieval, such as in hadoop map/reduce. This size of data retrieved can be pre-configured in hadoop file system HDFS. A tradeoff between the maximum size and information gained should be studied carefully to determine the best fit size. Hence, we need to further investigate this concern in our future search.

## Gradual Access

Big data nature is public. Big data is a prone to external attacks more likely than the conventional data. Many users from many organizations may enquire big data analytics. The large increase number of users may require a robust access control model to manage a proper discriminated access for variants of user’s privileges. The access control model should be granularly integrated with the distributed environment to manage gradual levels of access, and without affecting the analytics performance.

# Multi-Dimensional Sensitivity-Based Anonymization model (MDSBA)

MDSBA model is developed to resolve the above concerns, by considering the previously mentioned facts. The MDSBA model is implemented with considering the followings: gradual access for multi-level users, implementing Role-Based Access Control (RBAC) in MapReduce environment, and proposing an anonymization model with a subtle performance for the parallel distributed environment. MDSBA adopts multi-dimension technique for performing a high level of computation for MapReduce.

MDSBA splits data horizontally instead of vertically. Data is split based on attributes values rather than using a small junk of data records. This method serves the increase percentage of data equivalency fact. Data is split twice into four different groups with two levels, which enables a better multi-task approach in the distributed environment. Moreover, data is categorized into three different categories; equivalent, semi-equivalent, and non-equivalent. Equivalent data is defined as the number of equivalent records that is higher than or equal the k value in k-anonymity. Equivalent data can’t be anonymized, while anonymization is applied on semi-equivalent and non-equivalent only.

The semi-equivalent is defined by at least two qid values equivalency. The semi-equivalent is a middle case between fully and none equivalent data. The semi-equivalent records are grouped separately, and the grouping process is completed by grouping the equivalent qid’s only. The non-equivalent records are grouped with one qid only, which is the highest qid probability value.

MDSBA can reduce the data loss by using two techniques; skipping the masking process on equivalent records, and distinguishing between semi and non-equivalent records on applying masking process. The masking of non-equivalent records induces extra penalty on anonymization. This penalty is necessary to generalize the diverted values in an interval or a taxonomy tree.

MDSBA is reliable and can be implemented by using Pig Latin, Hive, Spark, Java or any other scripting languages, or even a combination of them. MDSBA was proposed to mimic the MapReduce environment, where a master server controls the slave nodes or (workers). Each node is configured to run map, and/or reduce. In the recommended cluster structure, as shown in Figure 1, the map tasks are conducted on multi data nodes (slaves). The data nodes read directly from the repository file, and process the query. The data node output is emitted to the reduce task nodes over the network.

Therefore, the master server may run the map processing on one node, and the key/pair value is emitted to another node. The master server creates a job, and each job contains three main tasks, map, shuffle, and reduce. Users trigger the job by using a script, which contains queries, and each query may contains one or more tasks. The job tracker creates a job, and divides tasks between nodes. Since each node is directly connected to the data or file repository; then each data node reads part of the file/data. As mentioned before, the data node reads a limited size of the data, and this can be determined by the HDFS accommodation size.

MDSBA takes the MapReduce structure and the processing steps into consideration. Therefore, two jobs can be assigned to the MapReduce. The first job includes; read from data, filter data, group data, and filter again, to create both of SG and NG groups. As shown in Figure 3, an example of Pig Latin script for the first two jobs.

Job Tracker

(Primary Name Node)

Client

Master

Slaves

Results

Reduce Tasks

(Data Node 4)

Map Tasks

(Data Node3)

Map Tasks

(Data Node2)

Map Tasks

(Data Node1)

Figure 3, MapReduce structure