Snir Shem-tov 204425490  
Oz Blumenfeld 308458314

K-anonymity and ℓ-diversity project

Table of contents

Introduction…………………………………………………………………………………………2  
The algorithm………………………………………………………………………………………3  
The mining algorithm……………………………………………………………………..……5  
Description of the Software…………………………………………………………………6  
Data set ………………………………………………………………………………………………8

Test results …………………………………………………………………………………………9

Evaluation and analysis…………………………………………………………….…………10

Introduction: the challenge of preserving our privacy  
Our society experiences in recent years unprecedented growth in the amount of data that is collected on individuals, organizations, companies and other entities.  
Of particular interest are data containing structured information on individuals.   
Data holders are then faced with the intricate task of releasing data in order to detect interesting trends or correlations, while still protecting the privacy of individuals.  
Privacy-preserving data mining has been proposed as a paradigm of exercising data mining while protecting the privacy of individuals.  
Many approaches were suggested, implemented and theoretically studied for playing this delicate game that requires finding the right path between data hiding and data disclosure. One of these approaches, proposed is k-anonymization.  
The method of k-anonymization suggests modifying the values of the public attributes of the data by means of generalization so that if the database is projected on the subset of the public attributes, each record of the table becomes indistinguishable from at least k−1 other records. Consequently, the private data may be linked to sets of individuals of size no less than k, whence the privacy of the individuals is protected to some extent.  
The model of k-anonymity has been shown to be insufficient to protect against all types of linking attack, whence it must be enhanced by additional security measures such as ℓ-diversity as we described in class.

The algorithm  
 One of the most well studied models of privacy preservation is k-anonymity.  
 The algorithm that Jacob and Tamir came up with is a new algorithm that is designed to achieve k-anonymity with high utility, independently of the underlying utility measure.   
That algorithm is based on a modified version of sequential clustering which is the method of choice in clustering.  
Experimental comparison with four well known algorithms of k-anonymity show that the sequential clustering algorithm is an efficient algorithm that achieves the best utility results.  
Based on the success of the sequential clustering algorithm, Jacob and Tamir came up with the modified algorithm that outputs k-anonymizations which respect the additional security measure of ℓ-diversity.  
  
The algorithm high level explanation:  
**input** Table D||D′ = {R1||S1, . . . , Rn||Sn}, an integer k, a real parameter ℓ ≥ 0.  
**output** A clustering of D||D′ into clusters of size at least k that respect ℓ-diversity.  
  
1.Checks if ℓ-diversity can be achieved for the table of records D||D' if not ℓ parameter is illegal.  
   
2.Call auxiliary algorithm 3,ℓ-diversity respecting splitting procedure, that makes random partitions of the table D||D' into t different clusters.  
  
3.For any record Ri (let Cj be the cluster which record Ri||Si currently belongs and find the cluster Cr which (1) ∆i:j→r is minimal.

4.For each cluster Cj where |Cj|>ωk (2) call algorithm 3 and split Cj into 2 clusters if the two respects ℓ-diversity keep them, otherwise retain Cj.

5.If at least 1 record was moved during the last loop go to step 3 again.

6.While the number of clusters of size smaller than k is greater than 1,  
we unify the two closest small clusters.

7.Return the resulting clustering.

(1) Where ∆i:j→r is a function of calculating the information-loss given record index i, origin cluster j and destination cluster r.

(2) ω is a parameter to limit the size of a cluster, where 1 < ω ≤ 2.

The mining algorithm

The Database we chose is the adult database mainly used for machine learning, data mining purposes and contains 32,561 instances and 14 attributes.

This database is rich with vary information of different personals, but sometimes it wasn't enough in our checks of our implementation,  
So we chose to modify it in order to get a clear picture.

The Database we chose is the adult database mainly used for machine learning, data mining purposes and contains 32,561 instances and 14 attributes.

The algorithm parses each line in the database and correspond each attribute to a field for the representation in our implementation of a person.

After creating each person, we add him to our records table as appears in the article as D||D'.

There were also difficulties involving such a large database such as long running time, since the algorithm repeat the main loop until it converges.   
Of course, convergence is much harder task to complete as the database becoming larger.

Description of the software

Our software contains 5 python files:

Cluster.py

Clusters.py

DBParser.py

Functions.py

Main.py

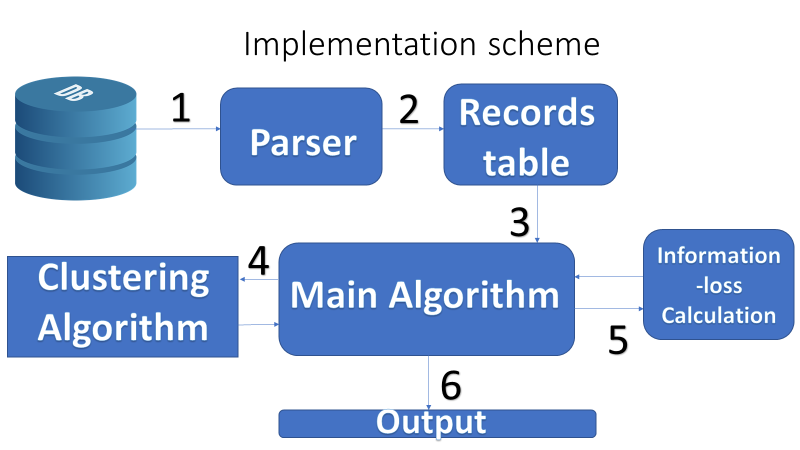
Main class:  
The main class is where we set our parameters for the current run of the program and include the next parameters:  
 1. The number of people records as part of the input database.  
 2. A parameter K sets the K-anonymity we will achieve.  
 3. A parameter ℓ sets the ℓ-diversity we will achieve.

Functions class:

In this class we implemented our algorithms i.e. the main algorithm for k-anonymity and ℓ-diversity, algorithm for random partitioning the records table D||D' to t clusters and some other auxiliary functions.

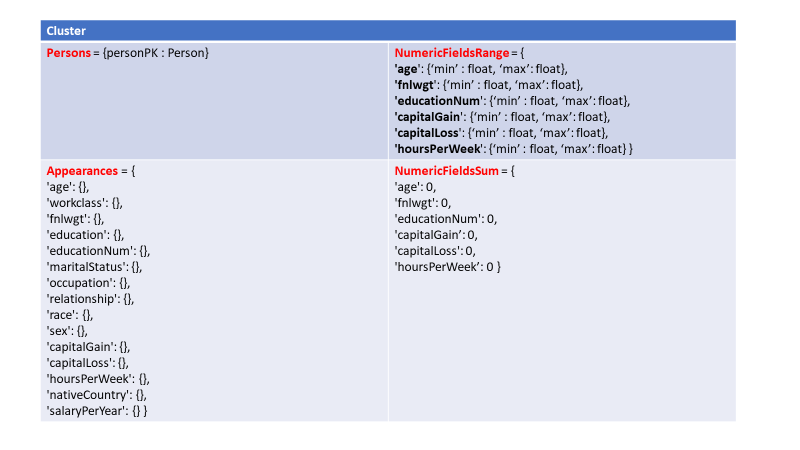
Database parser class:  
We parsed a big database (adults database) and each line in the input database is transformed into a person representation as follows:

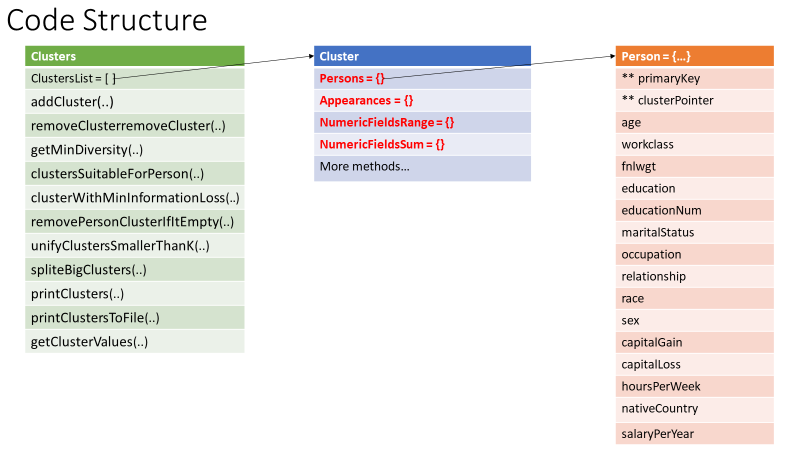
Each person which we save in memory as dictionary in order to reduce the complexity of functions callings repeatedly time after time, so we can access each field (information) with O(1) time complexity.  
Each person we parsed added to our records tables, the same one we would use as an input for our algorithm in order to achieve k-anonymity and ℓ-diversity.



Cluster class:

Contains all persons in the cluster and other statistical data such as appearances for each public and sensitive information for all the persons   
in the clusters.  
As well as the range of each statistical information by maintaining the minimum and maximum for each field.





Clusters:   
This class has list of all clusters that contains all the records from the original table.  
In this class we implemented some important functions such as calculation of information loss function or in other words which cluster fits some person the most.

Data set

The data we receive contains 14 statistics information for each person such as: age, work, education, wage and so on.

The data is stored in a records table and then divided into some number of different clusters.

Furthermore, the data we store in each cluster is records of people that contains quasi-identifiers and personal, sensitive, information.  
While we are finding the right balance between disclosure of this information and the privacy of each person.  
In addition we took the country of origin as the sensitive data for each person record because it has more diversity over other potential data fields.  
Later we also added more countries of origin to the database in order to make the data set more vary and to raise the limit of the diversity for each cluster.

Test results

When the main loop is finished, each person is belonging to the best cluster for him in terms of loss information.  
Likewise, each cluster size is at least ‘k’, and there are at least ‘l’ people with different sensitive value (‘native country’ in our case), so it maintains k-anonymity and l-diversity.

For each cluster, all the persons which belonging to it getting the same fields values: Numeric fields getting the range of all the other people and non-numeric fields getting the union of all the other fields. Of course, the sensitive data will stay as is without any change.

For example, if in some cluster there are 3 people with these values:

Person1 age: 20 education: Bachelors sensitive: A

Person2 age: 25 education: Bachelors sensitive: B

Person3 age :30 education: Masters sensitive: C

So, they will get these values:

Person1 age: 20 - 30 education: Bachelors | Masters sensitive: A

Person2 age: 20 - 30 education: Bachelors | Masters sensitive: B

Person3 age: 20 - 30 education: Bachelors | Masters sensitive: C

The new data file is created in the project path in ‘results’.  
For your convenience, the results appear in the clusters order and not in the original order.

Also, a text file with more details is created in the same path.

Evaluation

The algorithm we implemented is based on the article of K-anonymity and ℓ-diversity and achieve both of these properties for the given database.  
As well the time complexity and space are the same to the original algorithm as we followed the guidelines of Goldberger and Tassa almost entirely.

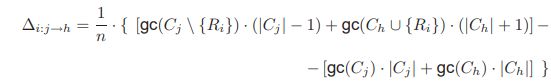
We were delighted to see the results and that the algorithm works nicely and converges quickly enough.  
  
Analysis  
  
Given the database we parse it to its fields and creating records of people and add it to our record table and process continue as we explained before in this report and the presentation which attached to our submission ZIP.

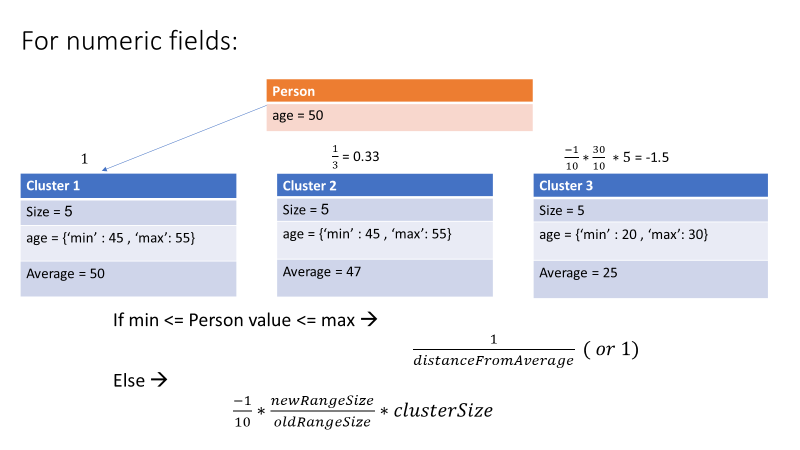
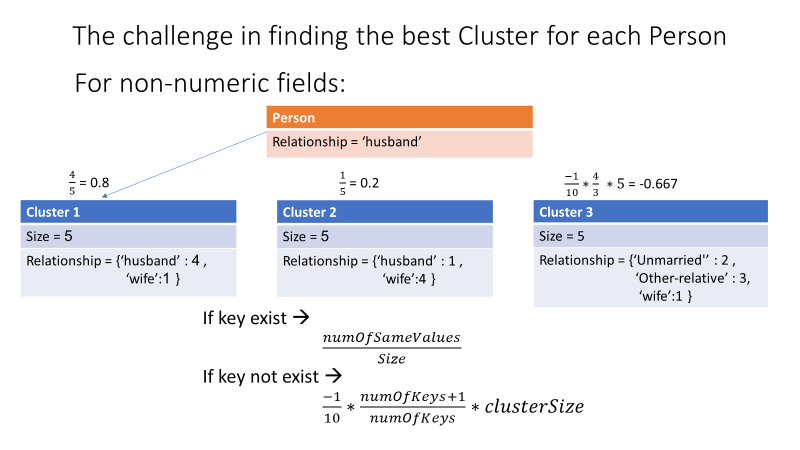
In the result files we can see partition of an input file to clusters of size at least k for the parameters k, l and database size.  
We can see that in each cluster, every record is indistinguishable to at least k-1 records in the cluster.  
Furthermore each cluster is ℓ-diversity in respect of the country of origin which functions as the sensitive identifier in our database.

All of the result files located in the submission file in folder "Results".   
the results data files have the output of the algorithm, database records as part of clusters that have the properties of k-anonymity and l-diversity.

Whereas the .txt files contains the statistical information of each cluster such as range of values in each cluster and so on.

How does it work?

Well, each person records located in the most "suitable" cluster for him,  
When we say suitable, we mean that we checked for all the clusters if the information loss function is negative and we pick for the target cluster, the cluster with the minimal value.  
  


An explanation to the formula:  
Given record i which resides in cluster j, we calculate the gc (Generalization Cost) function for the case we remove Ri from cluster Cj and put it in cluster Ch.  
And subtract it with the result of gc if we are staying in the status quo (meaning without any changes).  
We will move Ri to the Ch with the minimal value if and only if the value is negative.   
In our implementation we followed the same guidelines and implemented it a bit differently followed with approval of the course staff, using the follow heuristic:  


We do this calculation for each field and take all of the results we got and sum them.  
 the maximum result is the most suitable cluster for the given person.