HHS Public Access

Author manuscript

Priv Stat Databases. Author manuscript; available in PMC 2021 September 01.

Published in final edited form as:

Priv Stat Databases. 2020 September ; 12276: 136–148. doi:10.1007/978-3-030-57521-2_10.

Multivariate Top-Coding for Statistical Disclosure Limitation

Anna Oganian[1], **Ionut Iacob**[2], **Goran Lesaja**[2,3]

[1]National Center for Health Statistics, 3311 Toledo Rd, Hyattsville, MD, 20782, U.S.A.

[2]Georgia Southern University, Department of Mathematical Sciences, P.O. Box 8093, Statesboro, GA 30460, U.S.A.

[3]United States Naval Academy, Mathematics Department, 121 Blake Road, Annapolis, MD 21402, U.S.A.

Abstract

One of the most challenging problems for national statistical agencies is how to release to the public microdata sets with a large number of attributes while keeping the disclosure risk of sensitive information of data subjects under control. When statistical agencies alter microdata in order to limit the disclosure risk, they need to take into account relationships between the variables to produce a good quality public data set. Hence, Statistical Disclosure Limitation (SDL) methods should not be univariate (treating each variable independently of others), but preferably multivariate, that is, handling several variables at the same time. Statistical agencies are often concerned about disclosure risk associated with the extreme values of numerical variables. Thus, such observations are often top or bottom-coded in the public use files. Top-coding consists of the substitution of extreme observations of the numerical variable by a threshold, for example, by the 99th percentile of the corresponding variable. Bottom coding is defined similarly but applies to the values in the lower tail of the distribution. We argue that a univariate form of top/bottom-coding may not offer adequate protection for some subpopulations which are different in terms of a top-coded variable from other subpopulations or the whole population. In this paper, we propose a multivariate form of top-coding based on clustering the variables into groups according to some metric of closeness between the variables and then forming the rules for the multivariate top-codes using techniques of Association Rule Mining within the clusters of variables obtained on the previous step. Bottom-coding procedures can be defined in a similar way. We illustrate our method on a genuine multivariate data set of realistic size.

Keywords and phrases:

Statistical disclosure limitation (SDL); top-coding; hierarchical clustering; association rule mining; dimensionality reduction; genetic algorithm

aoganyan@cdc.gov.

1 Introduction

Many national surveys conducted by government agencies have a large number of attributes of different types. Some examples of such surveys in the USA are the National Health Interview Survey [15], the Behavioral Risk Factor Surveillance System [4], the Current Population Survey [7], and the American Community Survey [1]. Government statistical agencies have an obligation by law to protect the privacy and confidentiality of their respondents who can be individuals or enterprises. This is usually done by altering—we use the term *masking*—the original data before release, for example, by aggregating categorical values, swapping data values for selected records, adding noise to numerical values, or synthesizing some or all of the responses. See [12, 13] for more details.

Records that have extreme or very large values of numerical attributes are often a subject of concern about disclosure risk associated with these values. One way of addressing such a risk is to top code numerical attributes which are considered as ”visible” or possibly known from other publicly available data sources and which are not a subject to very frequent variation. For example, a person’s height can be top-coded to 75 inches, so all the individuals who are taller than 75 inches are recorded in the category “75 inches and above”. Such top-coding thresholds are chosen by the data protectors. Typically these thresholds are the estimates of the upper percentiles of the corresponding variable, for example, 95th, 97th, or 99th percentiles.

However, when top-coding thresholds are determined independently of other variables, protection may be inadequate for some groups of individuals. For example, assume the attribute weight is top-coded to 300 pounds for all the respondents. However, a female respondent with such a top-coded weight whose race/ethnicity is Asian could be more extreme as opposed to a respondent with the same weight who is a white male [14]. Being more extreme and rare, these individuals are more likely to be subject to re-identification. Thus, from the disclosure risk perspective it would be desirable to determine appropriate top-codes for the individuals in this group, different from those for the rest of the population. First, such subgroups should be identified. In some cases, as in the example above, it may be intuitive and easy. However, in general, in data sets with a large number of attributes, such a task is not always trivial. In this paper we propose a procedure that we call *multivariate top-coding*. It consists of identifying sub-populations/groups of records that require adjusted top-codes and then computing such top-codes for these groups.

1.1 Contribution and plan of the paper

The main contribution of the paper is a new multivariate top-coding procedure which is based on clustering variables and using techniques of Association Rule Mining (ARM) [2] to determine the sub-populations that should be top-coded differently from others. In Section 2 our multivariate top coding procedure is described. In Section 3 we illustrate the application of this procedure to a genuine data set of realistic size. Concluding remarks are given in Section 4.

2 The description of the multivariate top-coding method

Assume there is a microdata set D with p variables and the data protector decides to top-code numerical variables $T = \{T_1, \cdots, T_k\} \in D$, $k < p$. If there are many variables in D, then the number of possible combinations of categories of variables can be extremely high, and each such combination defines a potential sub-population or group of individuals. Thus, identification of the groups of individuals which require adjusted top-codes can be computationally very demanding. To make it feasible, we propose first to cluster the variables in D into groups, where each group is formed around each numerical variable T_i that is selected for top-coding. Next, we perform the search of the sub-populations that should have special top-codes for T_i within the vertical partition corresponding to the cluster of variables around T_i.

2.1 Clustering approach

In [16], several methods of clustering of variables were described and compared within the framework of disclosure limitation. These are hierarchical clustering methods that operate on the dissimilarity matrix which represent pairwise dissimilarities or "distances" between the variables. The metric of distance is based on the squared canonical correlation which can be computed for variables of different types (see [6, 16]). The dissimilarity matrix is created as a lower triangular $p \times p$ matrix DM with elements $DM[i, j] = 1 - r[i, j]$ for $i > j$, and 0 otherwise, where $r[i, j]$ is a squared canonical correlation between variables V_i and V_j.

In [16], *K-Link* and *Single-Link* methods were ranked high among the best performing clustering methods within the framework of disclosure limitation. These methods, however, may produce big clusters where some of the variables within the cluster may have low correlation, which is not optimal for our case. For example, if the variable income is being top-coded, then variables that are not correlated with income most likely will not be included by the subsequent ARM in the description of those sub-populations which need special top-codes for income.

A better way to group the variables for multivariate top-coding is to include in each cluster only the closest variables to T_i, which are no further than $1 - h$ from T_i. The cut-off value h depends of the preferences of the data protector, intuitively representing a trade-off between accuracy/utility and computational complexity of the procedure. In this approach multiple cluster membership is allowed so the same variables may be used to describe different sub-populations. For example, sex and race could define different subpopulations such as "Asian females" and "white males" that should have different top-codes for a person's weight. This simple variable grouping is much faster than other clustering algorithms as it does not even require computation of the whole dissimilarity matrix DM, but only those rows of DM which correspond to the variables in T. Once variables are clustered in k groups, each one centered at some $T_i \in \{T_1, \cdots, T_k\}$, the search of sub-populations that require special top-codes for each of T_i will be done within the corresponding cluster. To accomplish this search we propose to use Association Rule Mining (ARM), a popular machine learning rule-based methodology for discovering interesting relationships between the variables. There are several reasons why we decided to use ARM. First, the problem of multivariate top-coding, as we outlined it above, can be expressed as a search of association rules for variables T_i. An

association rule [2] is an expression of the form $X \rightarrow Y$, where X and Y express conditions on the attributes of the following form:

$$V_i = cat_{i_l} \wedge \cdots V_f \in [l_f, u_f] \cdots \wedge V_j = cat_{j_m} \tag{2.1}$$

where $V_i, V_j, \cdots V_f$ are the variables from the data set D, $cat_{i_l}, \cdots cat_{j_m}$ are the categories of the categorical variables, and $[l_f, u_f]$ are specific intervals within the domains of the corresponding continuous variables. In the paper we call the antecedent of the rule, X, a "LHS of the rule", and the consequent of the rule, Y, a "RHS of the rule".

The association rules that we are proposing for multivariate top-coding are of the form:

$$\left(V_i = cat_{i_l}\right) \wedge \left(\cdots V_j = cat_{j_m}\right) \rightarrow T_i < threshold \tag{2.2}$$

For example, ($Sex = Female$) ∧ ($Height$ < 65 inches) → ($Weight$ < 200).

Another reason for using ARM is that these techniques are designed to work well for large data bases. ARM algorithms are implemented in many software packages, including R.

2.2 Background on ARM

Association rule $X \rightarrow Y$ is characterized by its support and confidence. According to the original definition and notation used in [2], a support of X, the antecedent of the rule, is defined as the proportion of records in the database D that satisfy the expression X:

$$Supp(X) = |\{r \in D \mid X \subseteq r\}|/|D|$$

where r denotes a record in D and $|\cdot|$ means cardinality.

A confidence of the rule is defined as the proportion of the records in D that contains X which also contains Y:

$$Conf(X \rightarrow Y) = Supp(X \cup Y)/Supp(X)$$

The standard Apriori [3] algorithm (or other well known algorithms, for example, ECLAT [20], FP GROWTH [11] or ASSOC [10]) can be used to mine association rules where all the attributes are categorical. The procedure usually consists of two steps. The first step is to mine the so called set of frequent itemsets, that is, to find expressions X with support higher than a predefined minimal support of the rule, $MinSupp$. The second step is to discover all the rules with the confidence higher than a predefined minimal confidence of the rule, $MinConf$.

Mining association rules on both categorical and numerical attributes, often called mining quantitative association rules, have been covered significantly less in the literature. There is no method that is considered a "gold standard" for quantitative association rules. The difficulty of mining these rules stems from the fact that numerical attributes are usually

defined on a wide range of different values. It's not practical to work on all possible numeric values, as is done for categorical values, because in most cases, there are many such values and each numeric value does not appear frequently.

In [18], a genetic-based algorithm called QuantMiner for mining quantitative association rules was proposed. QuantMiner works directly on a set of rule templates - preset formats - specifying which attributes occur in the LHS and the RHS of the rule. Templates can be chosen by the user or computed by the system.

For categorical variables QuantMiner computes frequent itemsets similar to Apriori; that is, finds frequently occurring instantiations $V_i = cat_{i_l} \wedge \cdots V_j = cat_{j_m}$. Then it generates a rule template for each such instantiation. For each rule template, the algorithm looks for the best intervals of the numerical attributes occurring in that template, which is achieved using the Genetic Algorithm.

The algorithm starts with an initial population of rules for each rule template. Different rules in the initial population have different intervals for continuous variables, randomly chosen within their domains. In the following generations, the intervals are subject to change by genetic operators of mutation and crossover [18]. The mutation operator changes the lower or the upper bound of the interval. The crossover operator consists of taking two intervals, called parents, at random and generating new intervals in such a way so that the new interval is either inherited from one of the parents or formed by mixing the bounds of the two parents. These operators are applied to the rules of each generation. After each application, the fitness of each rule is evaluated and the best rules, according to the chosen fitness function, are selected for the next generation. This process repeats over $GenN$ generations ($GenN$ is a parameter of the algorithm). After the last generation is created the best rule for each rule template is selected from the corresponding population of rules and included in the output.

The fitness function used in QuantMiner is proportional to the Gain of the rule ([9]) and the length of the intervals in the rule:

$$Fitness(Rule) = Gain(Rule) * \prod_j \left(1 - Prop_j\right)^2 \tag{2.3}$$

where $Gain$ is defined as follows:

$$\begin{aligned} Gain(Rule) &= Gain(LHS \rightarrow RHS) \\ &= (Conf(LHS \rightarrow RHS) - MinConf) * Supp(LHS) \end{aligned} \tag{2.4}$$

and where $Prop_j$ is the ratio of the interval length to the length of the domain of V_f.

2.3 Multivariate top-coding using ARM

Let P be a percentile rank chosen by the data protector to compute top-code thresholds for variable T_i. For example, if $P = 99$ then 99th percentile of T_i serves as a top-code threshold for this variable. For each variable T_i, let $Clust_i$ be the cluster of variables that contains T_i.

We propose the following procedure to determine which sub-populations may need special top-codes (that is, lower than the rest of the population) for each variable T_i in T.

1. Compute the P-th percentile for the variable T_i using all the records in the data set. Denote this marginal percentile as Z_i.
2. Mine the following type of association rules on the vertical partition of the data that corresponds to the cluster of variables $Clust_i$:

$$X \rightarrow T_i < (Z_i - \Delta) \tag{2.5}$$

   The LHS of the rule X represents any combination of the variables/categories from $Clust_i$, in the form given by expression (2.1). The RHS of the rule is the expression that makes the implication (that is, the rule) true. In the RHS of the rule we introduce parameter Δ which is the minimal difference between Z_i and the percentile for a particular sub-population that should "get" its own top-coding threshold, different from Z_i. Δ can be chosen by the data protector for practical reasons in order not to have too many top-codes which are not very different from Z_i.
3. Choose the rules with the confidence equal to P/100 or higher. Denote this set of rules as S. Note that, the confidence of a rule is the probability

$$P(T_i < (Z_i - \Delta) \mid X) \tag{2.6}$$

   Hence, the LHS of such rules defines sub-populations for which the P-th percentile of the variable T_i is at most $Z_i - \Delta$. Thus, extreme observations in these subpopulations might need to be protected by adjusting, that is lowering, their top-code thresholds.
4. For each subpopulation defined by the LHS of the rules mined on the previous step, compute the P-th percentile for T_i using the records that belong to these subpopulations. The computed percentiles may serve as the top-codes for these subpopulations.

To find quantitative association rules (step 2 of the procedure above) we used a modified QuantMiner procedure: we changed the way how interval boundaries of numerical variables that appear on the LHS of the rules are calculated. We also changed the form of the fitness function. Regarding the calculation of interval boundaries, in the original version of QuantMiner both ends of the intervals are subject to change by the operators of crossover and mutation and the shortest intervals are being sought. However, we fixed the lower end of the intervals at the minimal value of the domain for those numerical variables that appear on the LHS of the rule and are positively correlated with the top-coded variable T_i. If the numerical variable on the LHS of the rule is negatively correlated with T_i, then the lower end of the interval is subject to change and the upper end is fixed. This is done in order not to exclude the individuals with values of numerical variables close to the boundaries of the domain from protection by top-coding who should otherwise be protected. For example, assume the variable *Income* is being top-coded and the numerical variable *Hours*, hours worked per week, is positively correlated with *Income*. So, the association rule has the form:

$Hours \in [l, u] \rightarrow Income \leq threshold$. If $l \neq min(Hours)$, then those individuals with $Hours \in [min(hours), l]$ will not be top-coded, but those with $hours \in [l, u]$ will. This, however, leaves the former groups of individuals unprotected. It also does not make sense given the nature of relationship between *Hours* and *Income*.

As mentioned above, we also modified the form of the fitness function. Contrary to [18] our fitness function favors larger intervals of numerical variables V_f on the LHS of the rule, subject to the resulting rule satisfying minimal confidence and minimal support.

$$Fitness(Rule) = \begin{cases} \prod_j (Prop_j)^2, & \text{if } Supp(Rule) \geq MinSupp \text{ and } Conf(Rule) \geq MinConf \\ 0, & \text{otherwise} \end{cases} \quad (2.7)$$

The reason of this modification is again not to exclude any individuals from protection that otherwise should be protected. Indeed, larger intervals typically correspond to larger groups of individuals having values of numerical variables within these intervals. Thus, the largest intervals on the LHS of the rule in our algorithm define the largest sub-population for which expression (2.6) is true. Hence, these individuals need lower top-codes for variable T_i than the top-codes for rest of the population.

Finally, it is important to note that the procedure of bottom-coding is a straightforward conversion of the top-coding procedure described above.

3 Numerical experiments

We applied our approach of multivariate top-coding to a genuine multivariate data set that was downloaded from the UCI Machine Learning Repository [8]. This is a sample drawn from the Public Use Microdata Samples (PUMS) person-level 1990 US Census file. We will refer to this file as Census in the paper. In our experiments we used 66 numerical and categorical variables from this data set. Full description of the variables can be found in [5]. Some variables were excluded from the experiments, such as allocation flags, serial number and some others because they would not be used in practice. There are 1.8 million records in our data set.

To illustrate our approach we choose the variables *Income*1 - wages or salary earned by the individuals in 1989 and *Age* for top-coding. These types of variables are usually top-coded. As outlined in section 2, we first found clusters of variables around these two variables. For *Age*, the cluster consisted of the following variables: *Relat*1 - relationship of the respondent to the householder (householder is defined later in the text) with 13 categories, *Marital* - marital status with 5 categories, *Disable*2 - work prevented status with two categories, *Income*5 - social security income in 1989 (a numerical variable), *Rlabor* - employment status with 7 categories, *Work*89 - worked or not in 1989 with two categories, and *Yearsch* - educational attainment with 18 categories. In our experiments *Yearsch* was treated as a numerical variable, hence, the output rules were given in the form if $Yearsch < i \rightarrow$

*Income*1 < *Y*, which is more meaningful than potentially a large number of rules, each one differing by a particular category in *Y earsch*.

The cluster of variables around *Income*1 includes the following variables: *Class* - class of worker with 10 categories, *IndustryClass* - industry class with 13 categories, *Ocupclass* - occupation class with 8 categories, *Relat*1 - relationship within the household with 13 categories, *Disable*1 - work limitation with three categories, *Rlabor* - employment status with 7 categories, *Hour*89 - numerical variable denoting usual hours worked per week the year before the interview, *Week*89 - weeks worked the year before the interview, and *Y earsch* - educational attainment with 18 categories.

The default minimal support of the rules in QuantMiner is set up to be 10%, but in our experiments, we lowered the minimal support to 1% in order to be able to identify small sub-populations (of the size of 1% of the data set or larger) which may require their own top-codes. For the data set of this size, it means that the size of these sub-populations should be at least 18, 000. The main constraint on lowering support of the rules is the computational burden, because many more subpopulations need to be checked, and, as a consequence, many more potential rules should be tested by the algorithm.

It should be noted that the main purpose of the proposed procedure is to assist the data protector in the otherwise daunting task of going through the large number of possible combinations of the relevant attributes in a big data set in order to find rarely observed extreme observations of top-coded variables for certain groups of records or sub-populations. These sub-populations are usually associated with lower values of the numerical variables subject to top-coding. Our rules are meant to bring such special cases to the data protector's attention. However, the decision about whether to use these rules to apply top-codes or not depends on many factors, such as a particular scenario of data release, SDL practice at a particular institution, and preferences of data protectors. In any case, such decisions are usually made together with the subject area specialists. Furthermore, some of the rules may be obvious, or they may be always observed in the data; for example, the rules that have confidence equal to 100%. Thus, not every automatically mined rule should imply top-coding. In some instances, the rules that have confidence equal to 100% may be used with the goal to check and find incorrectly recorded observations or the values that are not plausible.

Due to space limitation, below we present a selection of rules for *Age* and *Income*1 that are representative for this data set. They have attribute categories that appear most frequently. It should be noted that, the rules presented below are not our recommendations for top-coding for this particular data set nor any similar data set. The rules and results presented in this section are for the illustration of our method of multivariate top-coding only. In our experiments we used a 99-th percentile as a parameter for top-coding thresholds. So, for the groups of individuals that fit the description that appears in the LHS of the rules, at least 99% of individuals in the data set have *Income*1 (or *Age*) below the threshold that appears on the RHS of the rules. It is worth noting, that univariate top-code thresholds for this data set using the same parameter *P*, that is, the 99-th marginal percentile rank, would be $88, 000 for *Income*1 and 87 years old for *Age*. Hence, univariate top-coding would imply that

these thresholds would apply to all the individuals, regardless of their other characteristics. Below we list several age- and income- related rules as an illustration.

Age-related rules:

*Relat*1 = Son/daughter of the householder → *Age* < 60

*Relat*1 = Other persons in group quaters ∧ *Work*89 = Yes → *Age* < 60

*Relat*1 = Housemate ∧ *Work*89 = Yes → *Age* < 65

Marital = Never married ∧ *Relat*1 = Housemate ∧ *Work*89 = Yes → *Age* < 60

Marital = Never married ∧ *Income*5 = 0 ∧ *Work*89 = Yes → *Age* < 60

Marital = Never married ∧ *Rlabor* = Civilian employee, at work → *Age* < 65

Marital = Never married ∧ *Rlabor* = Civilian employee, at work ∧ *Income*5 ∈ [0.0; 2500.0] → *Age* < 60

Marital = Never married ∧ *Disable*2 = No, not prevented from working ∧ *Work*89 = Yes → *Age* < 65

*Income*1-related rules:

*Hour*89 < 35 → *Income*1 < 28, 000

*Week*89 < 40.0 → *Income*1 < 40, 000

*Relat*1 = Son/daughter of the householder → *Income*1 < 40, 000

*Relat*1 = Grandson/granddaughter of the householder → *Income*1 < 33, 000

*Relat*1 = Persons in group quarters → *Income*1 < 35, 000

*Relat*1 = Other nonrelative of the householder → *Income*1 < 47, 000

*Relat*1 = Other relative of the householder → *Income*1 < 45, 000

*Relat*1 = Householder ∧ *Hour*89 < 35 → *Income*1 < 35, 000

*Disabl*2 = Yes, limited in kind or amount of work → *Income*1 < 50, 000

Rlabor = Institutionalized persons → *Income*1 < 30, 000

Occupclass = Service → *Income*1 < 40, 000

Occupclass = Farming → *Income*1 < 55, 000

Class = Employee of private for profit company ∧ *Y earsch* = High school diploma or less → *Income*1 < 55, 000

*Relat*1 = Husband/wife ∧ *Y earsch* = High school diploma or less → *Income*1 < 40, 000

Some of the rules presented above seem intuitive or common sense. One example of such rules are those that have income on the RHS and *Hour*89 (usual hours worked per week in 1989) and *Week*89 (weeks worked in 1989) on the LHS. These two variables are positively correlated with income. These rules, in essence, describe part-time workers in the previous

year. Therefore, the rules suggests lower top-codes for income for these individuals compared to the rest of the population.

Another example of rules that are intuitive are the rules that involve *Relat*1 (relationship of the respondent to the householder) on the LHS of the rules. For instance, when *Relat*1 = *son*/*daughter*, then the threshold for the *Income*1 and *Age* may be lower comparative to other groups of individuals. According to the documentation on 1990 Census data files [19], in most cases, a householder is the person, or one of the persons, in whose name the home is owned, being bought, or rented. Higher income respondents may be expected to be householders themselves, rather than living with a parent-householder, which may be one of the reasons for lower income and possibly younger age for these types of respondents. Similar reasoning may be applied to the rules that involve other relatives of the householder and their respective top-codes. Note, that in some (possibly rare) instances when several members of the family can be linked together, the advanced age of the son may allow the intruder to get a good estimate of the age of a parent, despite the fact that the age of the parent was top-coded. For example, if the age of a son of the householder is 75 years old (which is above the threshold limit in the corresponding rule above), and the age of a parent-householder is 95 years old, then univariate top-coding, at 87 years old will only apply to the householder, but not to the son. However, based on the age of the son, the intruder would know that the parent must be older than the threshold value of 87 years old, and most likely around 95 years old. Such an extreme age and such a rare combination (if present in the data) could lead to the re-identification of these individuals if univariate top-coding is used. Presence of other variables could improve the assessment of the intruder even further.

Rules that include a combination of the following three characteristics: marital status =“Never married” combined with zero or small values of social security income in the previous year (variable *Income*5), no disability, and worked during the previous year (*Work*89 = *yes*) for the most part describe a younger group of respondents in this data set; thus, the 99-th percentile of age for this group of individuals found by the rules is generally smaller than for the rest of the population.

Another characteristic that is related to income is the occupation of the respondent (*Occupclass* variable). The rules identified some occupation classes with lower values of *Income*1 in this data set. For example, *Occupclass* = *Service* which includes cooks, waiters and waitresses, housekeepers, cleaners, maids and housemen, hairdressers, welfare service aides and some others, has a lower 99-th percentile of income than the others, which agrees with the literature on the subject [17]. Also, according to the rules *Occupclass* = *Farmers* has a lower 99-th percentile of *Income*1 as well.

As expected, rules that included the variable *Y earsch*, educational attainment, indicated that if educational attainment is less than high school, then *Income*1 is limited, especially for certain categories of individuals in the data set.

We conclude this section by emphasizing that the focus of the paper is not the discussion and analysis of particular rules, but the development and description of the methodology to obtain such rules. Deeper analysis of the rules obtained by our procedure should be done by

the data protector and subject area specialist for each particular data set and the scenario of data release.

4 Concluding remarks and future work

In this paper we propose a new approach for multivariate top-coding for disclosure limitation in large databases with many attributes of different types. We outlined an automated procedure that can help the data protector to find subpopulations that may need their own top-codes, lower than the rest of the population. Such a procedure may be used as an aid for the data collecting organizations in the disclosure review process as an alternative, or in addition, to their regular procedures. Such procedures often involve identification of risky combinations of the variables, which is often based on intuition as well as knowledge of a particular data set. In big data sets these procedures may be complicated and computationally involved as they require computation of many tabulations to identify potentially rare/risky combinations of the categories of these attributes. Thus, an automated procedure to identify such cases can be helpful especially when the data protector intends to release data sets with many attributes of different types, such as big government surveys.

To reduce the complexity of the problem we outlined a two-step approach which consists first of clustering the variables around the top-coded variables, using squared canonical correlations, then running our association rule mining algorithm on a vertical partition of the data that consist of the variables that are in the same cluster with the top-coded variables. This two-step approach makes association rule mining and the subsequent work with the rules by subject area specialists computationally feasible.

We would like to note that the association rules found by the proposed approach are meant to bring to the data protector’s attention particular combinations of the attributes that are rarely associated with the extreme values of the numerical variable that is subject to protection. Data protectors can choose topcoding or some other technique for protection of these groups of individuals. For example, synthesis can be used to impute safer values of numerical attributes.

Our future work consists of finding efficient ways for further reduction of the number of association rules. Another direction of future research is to investigate possible ways of incorporation of the data protector’s preferences and knowledge in the algorithm. For example, certain individual characteristics are more visible or noticeable than the others; for instance, amputations/missing limbs, walking aids and some others. So, we will investigate the best way of weighting the variables/characteristics on the clustering step and the association rule mining algorithm as well.

Acknowledgments

The findings and conclusions in this paper are those of the authors and do not necessarily represent the official position of the Centers for Disease Control and Prevention. The first author would like to thank Ellen Galantucci from the Bureau of Labor Statistics for the helpful discussion on the content of Section 3. Also we would like to thank John Pleis from the National Center for Health Statistics for the careful review of the paper.

A Appendix. Variables in the Census data set mentioned in the paper

Class - Class of worker. Categories: 0 N/a, Unemployed who never worked. 1 Employee of a private for profit company. 2 Employee of a private not for profit company. 3 Local government employee. city, county, etc. 4 State government employee. 5 Federal government employee. 6 Self employed in own not incorporated business. 7 Self employed in own incorporated business. 8 Working without pay in family business or farm. 9 Unemployed, last worked in 1984 or earlier.

IndustryClass - Industry class. Categories: 1 Agriculture. 2 Mining. 3 Manufacturing. 4 Transportation. 5 Wholesale trade. 6 Retail trade. 7 Finance. 8 Business. 9 Personal services. 10 Entertainment. 11 Professional. 12 Public administration.

Occupclass - occupation class. Categories: 1 Managerial. 2 Professional. 3 Technical. 4 Sevice. 5 Farming. 6 Precision. 7 Operators. 8 Military.

*Relat*1 - Relationship to the householder. Categories: 0 Householder. 1 Husband/wife 2 Son/daughter 3 Stepson/stepdaughter 4 Brother/sister 5 Father/mother 6 Grandchild 7 Other relative 8 Roomer/boarder/foster child 9 Housemate/roommate 10 Unmarried partner 11 Other non related. 12 Institutionalized person. 13 Other person in group quarters.

*Disable*1 - Work limitation. Categories: 0 N/a. 1 Yes, Limited in kind or amount of work. 2 No, not Limited.

Rlabor - Employment status. Categories: 0 N/a 1 Civilian employee, at work. 2 Civilian employee, with a job but not at work. 3 Unemployed. 4 Armed forces, at work. 5 Armed forces, with a job but not at work. 6 Not in labor force.

*Hour*89 - Usual hours worked per week the year before the interview. This is a numerical variable with range from 0 to 99.

*Week*89 - Weeks worked the year before the interview. This is a numerical variable with range from 0 to 52.

Y earsch - educational attainment. Categories: 0 N/a. 1 No school completed. 2 Nursery school. 3 Kindergarten. 4 1st, 2nd, 3rd, or 4th grade. 5 5th, 6th, 7th, or 8th grade. 6 9th grade. 7 10th grade. 8 11th grade. 9 12th grade, No diploma. 10 High school graduate, diploma or GED. 11 Some College, But no degree. 12 Associate degree in College, Occupational. 13 Associate degree in College, Academic Program. 14 Bachelors degree. 15 Masters degree. 16 Professional degree. 17 Doctorate degree.

References

1. ACS: American community survey. United States Census Bureau, https://www.census.gov/programs-surveys/acs
2. Agrawal R, Imielinski T, Swami A: Mining association rules between sets of items iń large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD ‘93. pp. 207 – 216 (6 1993)

3. Agrawal R, Srikant R: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB. pp. 487 – 499. Santiago, Chile (9 1994)
4. BRFSS: Behavioral risk factor surveillance system. Centers for Disease Control and Prevention (CDC), https://www.cdc.gov/brfss/index.html
5. Census: US census (1990) data set. UCI Machine Learning Repository (2017), https://archive.ics.uci.edu/ml/datasets/US+Census+Data+%281990%29
6. Chavent M, Kuentz-Simonet V, Liquet B, Saracco J: ClustOfVar: An R Package for the Clustering of Variables. Journal of Statistical Software 50(i13), 1 – 16 (2012) [PubMed: 25317082]
7. CPS: Current population survey. United States Census Bureau, https://www.census.gov/programs-surveys/cps.html
8. Dheeru D, Karra Taniskidou E: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2017), http://archive.ics.uci.edu/ml
9. Fukuda T, Morimoto Y, Morishita S, Tokuyama T: Mining optimized association rules for numeric attributes. In: Proceedings of the 15th ACM SIGACTSIGMOD - SIGART PODS96. pp. 182 – 191. ACM Press (1996)
10. Hájek P, Havránek T: Mechanizing Hypothesis Formation: Mathematical Foundations foŕ a General Theory. Springer-Verlag (1978)
11. Han: Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD ‘00. p. 112 (2000)
12. Hundepool A, DomingoFerrer J, Franconi L, Giessing S, Nordholt ES, Spicer K, de Wolf P: Handbook on Statistical Disclosure Control (version 1.2). ESSNET, SDC project (2010), http://neon.vb.cbs.nl/casc
13. Hundepool A, DomingoFerrer J, Franconi L, Giessing S, Nordholt ES, Spicer K, de Wolf P: Statistical Disclosure Control. Wiley (7 2012)
14. NHANES: National Health and Nutrition Examination Survey. Centers for Disease Control and Prevention (CDC). National Center for Health Statistics (NCHS), https://www.cdc.gov/nchs/data/factsheets/factsheet_nhanes.htm
15. NHIS: National Health Interview Survey. Centers for Disease Control and Prevention (CDC). National Center for Health Statistics (NCHS), https://www.cdc.gov/nchs/nhis/index.htm
16. Oganian A, Iacob I, Lesaja G: Grouping of variables to facilitate SDL methods in multivariate data sets. In: Domingo-Ferrer J, M. F (ed.) Privacy in Statistical Databases, Lecture Notes in Computer Science. vol. 11126, pp. 187–199. Springer-Verlag (2018)
17. Ross M, Bateman N: Meet the low-wage workforce. Tech. rep, Brookings (2019)
18. Salleb-Aouissi A, Vrain C, Nortet C, Xiangrong Kong X, Vivek Rathod V, Cassard D: Quantminer for mining quantitative association rules. Journal of Machine Learning Research 14(61), 3153–3157 (2013), http://jmlr.org/papers/v14/salleb-aouissi13a.html
19. U.S. Department of Commerce Economics and Statistics Administration. BUREAU OF THE CENSUS: 1990 Census of Population and Housing. Public Use Microdata Samples. United States (1990)
20. Zaki MJ: Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12(3), 372390 (2000)