Coverage-based Queries: Nondiscrimination Awareness in Data Transformation

Chiara Accinelli*, Barbara Catania and Giovanna Guerrini

University of Genoa, Italy

Abstract

When people-related data are used in automated decision making processes, social inequities can be amplified. Thus, the development of technological solutions satisfying nondiscrimination requirements, in terms of, e.g., fairness, diversity, and coverage, is currently one of the main challenges for the data management and data analytics communities. In particular, coverage constraints guarantee that a dataset includes enough items for each (protected) category of interest, thus increasing diversity with the aim of limiting the introduction of bias during the next analytical steps. While coverage constraints have been mainly used for designing data repair solutions, in our study we investigate their effects on data processing pipelines, with a special reference to data transformation. To this aim, we first introduce coverage-based queries as a means for ensuring coverage constraint satisfaction on a selection-based query result, through rewriting. We then present two approximate algorithms for coverage-based query processing: the first, covRew, initially introduced in [3], relies on data discretization and sampling; the second, covKnn is a novel contribution and relies on a nearest neighbour approach for coverage-based query processing. The algorithms are experimentally compared with respect to efficiency and effectiveness, on a real dataset.

Keywords

nondiscrimination, data transformation, coverage, rewriting

1. Introduction

Today, we are surrounded by information that is increasingly being used to make decisions that could have an impact on people's lives. It is therefore very important to understand the nature of this social impact and take responsibility for it. To this aim, the development of responsible and nondiscrimination-aware technological solutions is one of the main challenges in data management and analytics and the satisfaction of ethical requirements is becoming a crucial element for asserting the quality of a dataset [9]. More precisely, nondiscrimination-aware data decision systems should take humans in the loop in all the data processing steps, from acquisition to wrangling and analysis: from one hand, they must ensure *transparency* and *interpretability*, for tracing the whole process; from the other, they should guarantee *nondiscrimination* with respect to protected groups of individuals. This is achieved by characterizing groups of interests in terms of sensitive attributes, like, e.g., gender or economical status, and relying on the

ITADATA2022: The 1st Italian Conference on Big Data and Data Science, September 20–21, 2022, Milan, Italy

*Corresponding author.

chiara.accinelli@dibris.unige.it (C. Accinelli); barbara.catania@dibris.unige.it (B. Catania); giovanna.guerrini@dibris.unige.it (G. Guerrini)

0000-0002-6626-9470 (C. Accinelli); 0000-0002-6443-169X (B. Catania); 0000-0001-9125-9867 (G. Guerrini)

specification of properties, like *fairness*, i.e., lack of bias [14], *diversity*, i.e., the degree to which different kinds of objects are represented in a dataset [8], and *coverage* [4], guaranteeing a sufficient representation of any category of interest in a dataset.

A significant ongoing research effort [8, 18] aims at a holistic treatment of nondiscrimination along the stages of the data processing life-cycle, rather than as a constraint on the final result: the sooner you spot the problem fewer problems you will get in the last analytical steps of the chain. Approaches have been proposed both with reference to front-end (e.g., OLAP queries [13], set selection [19], ranking [21]) and back-end stages (e.g, dataset repair during data acquisition, with a special focus on coverage in [4, 5, 12] and causal fairness [15]). We refer the reader to [7] for a short survey.

In our work, we are interested in nondiscrimination constraints defined in terms of coverage. Indeed, it is well known that the quality of a classifier depends not only on the used algorithm but also on the number of instances, i.e., the coverage, of each group in the dataset [16]. While coverage constraints have been mainly used for repairing raw datasets before any transformation [4, 5, 12], we investigate their effects on the data processing pipeline, with a special reference to data transformed through query execution.

Example 1. *Consider the StudentPerformance dataset*[1] *and suppose we want to predict who, among the best students, has parents with a high level of education (e.g. from graduation upwards), through a classification task. The information about the lunch at school allows us to deduce information about the economical status of the student family, thus* `lunch` *can be taken as sensitive attribute. Suppose we would like to train a model which accuracy does not deeply depend on the economical status of the student family, assuming that the lower the accuracy difference between the considered student groups the lower the model discriminates against them. Now suppose to qualify the best students with two selection conditions over math and reading subjects:* `math_score` $\geq$ `80 AND reading_score` $\geq$ `80`*. This query returns just 13 students with* `free/reduced` *lunch out of 143 individuals. When training a Random Forest classifier on the query result, we get a model with an accuracy equal to 0.71 and an accuracy difference equal to 0.28 (accuracy for standard lunch = 0.78, accuracy for reduced lunch = 0.5). Thus, the insufficient number of students with reduced lunch in the query result affects the accuracy difference of the model. Now suppose to modify the initial query with the following one:* `math_score` $\geq$ `71 AND reading_score` $\geq$ `68`*. This query returns 74* `free/reduced` *instances out of 350 individuals. The accuracy of the classification model trained on this new result slightly changes (0.69) but the accuracy difference is improved and corresponds to 0.006. The improvement is due to the higher number, i.e., the higher coverage, of protected instances in the query result. The price to pay is a relaxation of the original query.* ◊

In this paper, we present the main results we achieved for ensuring coverage constraint satisfaction during data transformation represented in terms of selection-based queries. In order to avoid disparate treatment discrimination [6], and similarly to what has been done in [13] for OLAP queries and causal fairness and, more recently, in [17] for range queries and different types of associational fairness, the input selection-based query, when executed over a given dataset *I*, is modified through *rewriting*, obtaining a new query that, when executed over *I*, satisfies the given constraints and is still close to the initial request. As far as we know and according

[1] It is available at https://www.kaggle.com/spscientist/students-performance-in-exams.

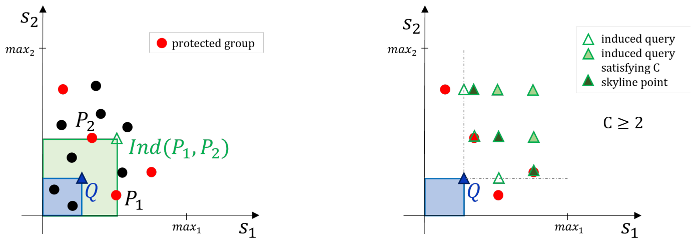
to [16], no other solutions addressing coverage-based rewriting have been proposed so far. More precisely, we first introduce *coverage-based queries* as a means for ensuring coverage constraint satisfaction in selection-based query results. We then present two approximate algorithms for coverage-based query processing: the first, `covRew`, initially introduced in [3], relies on data discretization and sampling; the second, `covKnn`, presented here for the first time, relies on a nearest neighbour approach for coverage-based query processing. The approaches are then experimentally compared with respect to efficiency and effectiveness, on a real dataset.

The remainder of this paper is organized as follows. Section 2 presents preliminary definitions and the reference problem. `covRew` is briefly described in Section 3 while Section 4 presents `covKnn`, an alternative nearest neighbour approximate approach for coverage-based query processing. Experimental results are presented in Section 5. Finally, Section 6 discusses related work and Section 7 concludes and outlines future work directions.

2. Problem Definition

Dataset. We assume data are represented as a collection of tabular datasets (e.g., relations in a relational database, data frames in the Pandas analytical environment, called *tables* in the following) $I \equiv \{I_1, ..., I_r\}$, with disjoint schemas, for the sake of simplicity. Let $A_1, ..., A_m$ be the attributes of I, D_{A_j} denote the domain of A_j. A set of discrete-valued attributes $\mathcal{S} = \{S_1, ..., S_n\}$ are of particular concern since they allow the identification of protected groups and are called *sensitive attributes*. Examples of sensitive attributes are `gender` and `race`.
Data transformation operations. We focus on *selection-based data transformations* (or *queries*) over stored or computed datasets, in data processing pipelines that might alter the representation (i.e., the *coverage*) of specific groups of interests, defined in terms of sensitive attribute values. Examples are data slicing operations in Pandas[2] or ColumnTransformers in Scikit-Learn[3]. We consider boolean combinations of atomic selection conditions $sel_i \equiv A_i \theta v_i$, $v_i \in D_{A_i}$, $\theta \in \{<, \leq, >, \geq\}$, A_i numeric attribute,[4] $i = 1, \ldots, d$, $A_i \notin \mathcal{S}$. A selection-based query Q is thus denoted by $Q\langle v_1, ..., v_d\rangle$ or $Q\langle \bar{v}\rangle$, $\bar{v} \equiv (v_1, ..., v_d)$. Attributes $A_1, ..., A_d$ are called *dimensions* of the selection-based query.
Coverage constraints. Conditions over the number of entries belonging to a given protected group of interest returned by the execution of a selection-based query can be specified in terms of *coverage constraints* [4, 12]. A *coverage constraint* C has the form $\downarrow^{S_{l_1},...,S_{l_h}}_{s_{l_1},...,s_{l_h}} \geq k$ and specifies that the minimum number of instances with sensitive attribute S_{l_i} equal to s_{l_i}, $i = 1, ..., h$, in a query result has to be k. As an example, $\downarrow^{\texttt{gender}}_{\texttt{female}} \geq 10$ specifies that a query result should include at least 10 female individuals. The group a coverage constraint refers to is called *protected group* and is characterized by the selection condition $PG \equiv S_{l_1} = s_{l_1} \wedge ... \wedge S_{l_h} = s_{l_h}$. Even if coverage thresholds are usually application specific and should be determined through statistical analysis, the central limit theorem suggests that the number of representative should be around 30 and, according to [20], at least 20 to 50 instances for each group are required.

[2] https://pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/03_subset_data.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html
[4] For the sake of simplicity, selection attributes are assumed numeric. The approach can be easily extended to deal with any other ordered domain.



(a) Input and induced queries

(b) Skyline points

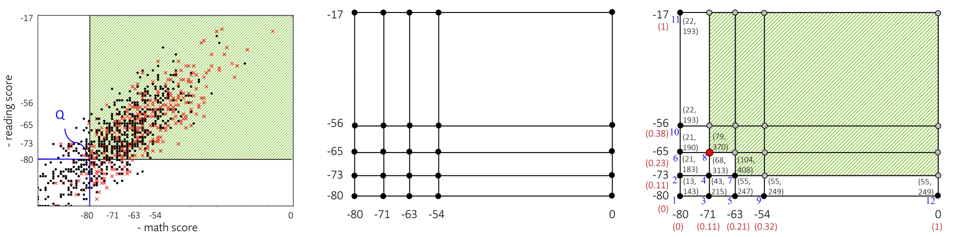
Figure 1: Coverage-based query properties

Definition of coverage-based queries. Let C be a set of coverage constraints over a set of sensitive attributes S and let $Q\langle\overline{v}\rangle$ be a selection-based query. A *coverage-based query* ξ^C_Q for C and Q is a selection-based query that, given a dataset I, stretches the result $Q(I)$ as little as possible so that constraints in C are satisfied. More precisely, for each dataset I: *(i)* a tuple $\overline{u}$ exists such that $\xi^C_Q(I) = Q\langle\overline{u}\rangle(I)$; *(ii)* $Q \subseteq \xi^C_Q$; *(iii)* all coverage constraints are satisfied by $\xi^C_Q(I)$; *(iv)* ξ^C_Q is the closest query to Q in terms of *minimality*, i.e., result cardinality, and *proximity*, i.e., syntactic closeness. Minimality means that no other query Q' satisfying conditions *(i)–(iii)* exists such that $card(Q'(I)) < card(\xi^C_Q(I))$; *proximity* means that $Q\langle\overline{u}\rangle$ is the closest query to $Q\langle\overline{v}\rangle$, according to the Euclidean distance (defined in a unit space) between $\overline{v}$ and $\overline{u}$, satisfying properties *(i)–(iv)*.

Properties. It has been proved that a coverage-based query ξ^C_Q satisfies the following properties:

- *(P1)* It can be represented in a *canonical form* in which each selection condition has the form $A_i \leq v_i$ or $A_i < v_i$; $Q\langle\overline{u}\rangle$ can thus be represented as point $\overline{u}$ in the d-dimensional space defined by selection attributes (see Q in Figure 1(a)).
- *(P2)* Let $\xi^C_Q(I) \equiv Q\langle\overline{u}\rangle(I)$. It can be proved that $\overline{u}$ coincides with the upper right vertex of the minimum bounding box of at most d distinct points in I, Q, and the origin of the space (see the green triangle in Figure 1(a)). Such vertices are called *induced points* and the set of all induced points corresponds to the search space for coverage-based queries.
- *(P3)* There is a relationship between $\overline{u}$ and the skyline of induced points corresponding to queries that, when executed over I, satisfy C. The dominance relation, needed for the skyline computation, is defined over selection attributes, assuming the lower the better (see Figure 1(b)). It can be proved that $\overline{u}$ coincides with the skyline point corresponding to the query with the minimal cardinality at the lowest distance from Q.

The problem. The problem we address concerns the design of efficient algorithms for processing ξ^C_Q. The designed algorithms improve the naïf approach derived from properties P1, P2, and P3, which is inherently inefficient due to the size of the search space and the skyline computation. We do not rely on any index data structure, so that both stored and computed datasets can be considered. The proposed approximate algorithms are briefly described in the next sections while an optimized precise one is currently under evaluation.



(a) Data distribution

(b) Multi-dimensional grid

(c) Cardinality estimates and solution

Figure 2: Data representation and processing

3. Approximate Coverage-based Query Processing through Discretization and Sampling

In order to improve the efficiency of the naïf approach, the approach proposed in [1, 2, 3] relies on a discretized search space instead of generating and visiting the whole set of induced points. Additionally, cardinality estimation, needed for constraint and minimality checking, is performed through a sample-based approach. The proposed technique, called `covRew`, can be applied over any dataset for which a sample is available or can be easily computed on the fly, and relies on two main steps:

- **Pre-processing**. The approximate search space is generated by considering the intersection points of a grid obtained by discretizing each axis (one for each selection attribute A_i in $Q\langle\overline{v}\rangle$, from the query value v_i to the maximum value of A_i in I), using standard binning approaches (e.g., equi-width and equi-depth). Each point on the grid corresponds to a selection-based query of type $Q\langle\overline{a}\rangle$, thus satisfying conditions *(i)* and *(ii)* of the reference problem.
- **Processing**. Given I, $Q\langle\overline{u}\rangle$ such that $\xi_Q^C(I) = Q\langle\overline{u}\rangle(I)$ is then computed by visiting the discretized search space starting from $\overline{v}$, one point after the other, at increasing distance from $\overline{v}$, checking minimality and proximity in the approximated space. The properties of the discretized search space and the canonical form are considered for pruning the space (algorithm `covRewP` in [2]), possibly increasing the number of points to be visited at different iterations (algorithms `covRewI` and `covRewIP` in [2], depending on whether pruning is considered together with iterations). The trade-off between accuracy and efficiency of the proposed algorithms has been evaluated on both synthetic and real-world datasets (see [2] for details).

Example 2. *Consider the scenario introduced in Example 1. Q can be rewritten in canonical form as* `-math_score` ≤ `-80 AND -reading_score` ≤ `-80`. *Figure 2(a) shows the dataset projected over attributes* `-math_score` *and* `-reading_score`. *Points are colored and shaped according to the sensitive attribute values: red crosses for* `free/reduced` *lunches and black dots for* `standard` *lunches. The result of Q* *in canonical form corresponds to the region at the bottom left side of the query point (-80,-80). The space to be discretized corresponds to the green rectangle. By applying the equi-depth approach to each axis with 4 bins, we obtain the grid in Figure 2(b). The discretized search space corresponds to the grid intersection points: each point represents a selection-based query containing Q. During the processing step, the points of the grid are visited, under different strategies, starting from (-80, -80), at increasing distance from it (blue numbers represent the visiting order). For each visited point $Q\langle\overline{a}\rangle$, we estimate the cardinality of $\sigma_{lunch=free/reduced}(Q\langle\overline{a}\rangle)(I)$, needed for checking constraint satisfaction, and of $Q\langle\overline{a}\rangle(I)$, by relying on a sample-based approach (see the pairs of numbers associated with each point in Figure 2(c)). The result corresponds to point $\overline{u}$ =(-71,-65) (see Figure 2(c)): it satisfies the coverage constraint ($\sigma_{lunch=free/reduced}(Q\langle\overline{a}\rangle)(I) = 79$, property (iii)), has the minimum cardinality (370), at the lowest distance from Q (property (iv)). Distances are computed in the reference unit space (normalized values are shown in red in Figure 2(c)). Query Q is thus rewritten into* `-math_score ≤ -71 AND -reading_score ≤ -65`. ◊

The number of cardinality estimations in covRew is in $O(n^d)$, where n is the number of bins used for the discretization of the axis and d is the number of selection conditions. As shown in [2], pruning strategies help in reducing such exponential complexity in real cases. On the other hand, thanks to the sample-based approach, covRew performance does not depend on the dataset cardinality.

4. A Nearest Neighbour Approach to Approximate Coverage-based Query Processing

Similarly to the naïf approach derived from properties P1, P2, and P3 in Section 2, covRew works in a space of query points to identify an approximation of $Q\langle\overline{u}\rangle$, needed to process the coverage-based query $\xi_Q^C(I)$. An alternative approach could be that of computing an approximation of $Q\langle\overline{u}\rangle$ directly from the data instances. To this aim, we first look for the missing number of protected group instances closest to the query point, with respect to an input distance function f, and then use them to compute the induced query representing an approximation of $Q\langle\overline{u}\rangle$. More precisely, given one coverage constraint $C \equiv \downarrow_{s_{l_1},\ldots,s_{l_h}}^{S_{l_1},\ldots,S_{l_h}} \geq k$, a selection-based query $Q\langle\overline{v}\rangle$, and a dataset I, the proposed approach, called covKnn, relies on four main steps:[5]

1. compute the set of protected group instances that are not included in $Q(I)$, corresponding to $S_{PG} \equiv \sigma_{PG}(I - Q(I))$;
2. determine the number of missing protected instances $h = k - card(\sigma_{PG}(Q(I)))$;
3. identify the h nearest neighbours to $\overline{v}$, with respect to f, in S_{PG};
4. return the query induced by the h nearest neighbours as an approximation of $Q\langle\overline{u}\rangle$.

It is easy to show that covKnn guarantees the satisfaction of properties *(i)*-*(iii)* of coverage-based queries; however, minimality and proximity (property *(iv)*) are not necessarily satisfied by the induced query even if they are implicitly taken into account during the nearest neighbour computation.

[5] covKnn can be easily extended to deal with a set of constraints by considering the query induced by the union of the h_i missing instances for each coverage constraint C_i in the input set.

One additional consideration is needed for the distance function to be used. As observed in [11], those based on the general theory of vector p-norms, like the Euclidean or Manhattan distances, are distances between points. They are therefore not suitable when considering distances from boxes corresponding to a query space, as in our case, since it is not obvious which point inside the box should be treated as the reference point. In such a context, the user would usually prefer answers that are close to the periphery of the box. The *box query distance* proposed in [11] achieves this goal. Given an input query $Q\langle\overline{v}\rangle$ and a database instance point $\overline{a}$ that does not belong to $Q(I)$, when considering selection-based queries in canonical form, it is computed as follows:

$$\sqrt{\sum_{i}^{d}(d_i(v_i, a_i))^2} \qquad \text{where } d_i(v_i, a_i) = \begin{cases} a_i - v_i & \text{if } a_i > v_i \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The box query distance can be customized to user needs by using weights. In our case, the idea is to use weights that make the distance of a point $\overline{a}$ to each query axis closer to the area corresponding to the contribution of $\overline{a}$ to the induced query, for that axis. To this aim, we use the *Inverse Aspect-Ratio* weights presented in [11].

Example 3. *Consider the scenario introduced in Example 1. Under* covKnn, *the processing proceeds in this way: (i) the search space S_{PG} is first computed (see the white space in Figure 3(a)): it contains 342 points out of the 1000 points of the whole dataset; (ii) assuming that $Q(I)$ contains 13 protected group instances, the number of missing instances is determined as h = 70-13 = 57; (iii) the 57 nearest neighbour points to the query point (-80, -80) in S_{PG} are computed, using the box query distance (green points in Figure 3(a)); (iv) the induced query of the detected points is then computed (see Figure 3(b)): it corresponds to the query* `-math_score ≤ -70 AND -reading_score ≤ -69`. ♢

In covKnn, no cardinality estimate is required (minimality is not checked and the coverage constraint is satisfied by the induced query by construction). The complexity is $O(md)$, where m is the cardinality of S_{PG} since, for each instance in S_{PG}, the distance from Q is computed according to Eq. (1). As a final remark, we notice that the obtained result is different from the one obtained with covRew (see Example 2): they represent two possible and potentially distinct approximations of the coverage-based query result.

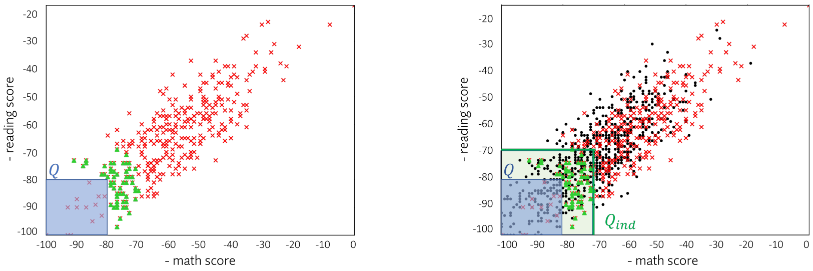
5. Experimental Evaluation

5.1. Experimental setup

All experiments were conducted on a machine with an Intel(R) Core(TM) i99900K 3.60GHz CPU and 16GB main memory, running Ubuntu 20.04. All programs were coded in Python 3.8 and the dataset is stored on PostgreSQL 9.6.19.1.

Dataset. The experiments refer to the *Adult* dataset,[6] exploited for assessing many non-discrimination data management techniques. It contains 48,842 individuals, described by six

[6] https://archive.ics.uci.edu/ml/datasets/Adult



(a) Query nearest neighbour points

(b) Induced query

Figure 3: Query nearest neighbours and the corresponding induced query

numerical attributes and eight categorical attributes. For our experiments, we use attributes `sex`, `race`, and `marital_status` as sensitive attributes.

Queries and coverage constraints. We identified 9 selection-based queries $Q_{i,j}$ with a variable number d of selection conditions and selectivity %sel (see Table 1): i corresponds to the number of selection conditions and j points out the selectivity (h corresponds to %sel $\leq$ 15%, m to 15% $<$ %sel $<$ 40%, and l to %sel $\geq$ 40%). We then considered many problem instances by combining the selection-based queries with various coverage constraints, requiring a different number of protected group instances. Table 2 presents the considered instances pointing out the cardinality of the initial query result $card(Q(Adult))$, the cardinality of each protected group in the initial query result $card(Q \downarrow_{C_i} (Adult))$, the missing number of protected group instances, according to the selected coverage constraint, and the cardinality of S_{PG} (#space).

Algorithms. The experiments compare `covKnn` and `covRewIP`; the precise algorithm derived from the properties presented in Section 2 is used as a baseline. `covRewIP` is the grid-based algorithm guaranteeing the best tradeoff between accuracy and efficiency according to [2], executed using the equi-depth discretization, with 32 bins, applied over a random sample of size 9,604 of the *Adult* dataset. In the `covKnn` implementation, we rely on a naïf top-sort approach, in which we keep in a buffer the best k points seen so far, for nearest neighbour computation.[7]

Distance function. We performed many experiments, considering many different weights for the *box query distance* proposed in [11]. The reported experimental results refer to the box query distance with squared *Inverse Aspect-Ratio* weights (see [11] for details).

Efficiency and effectiveness indicators. Efficiency is analyzed in terms of execution time for determining the new query to be executed (averaged over 10 executions). For better pointing out the impact of the dataset size, we include the time to load the dataset for `covKnn` (0.19 s) and to generate and load the sample for `covRewIP` (0.071 s).[8] Effectiveness is evaluated by comparing: (i) the achieved *semantic relaxation*, called *relaxation degree* in [2], as the rate between the cardinality of $\xi_Q^C(Adult) - Q(Adult)$ (i.e, the number of new tuples returned) and of $Q(Adult)$; (ii) the *syntactical relaxation*, i.e., the proximity of $Q\langle\overline{u}\rangle$ with respect to the initial query $Q\langle\overline{v}\rangle$. The baseline approach is used only for the analysis of the accuracy since experiments, that are

[7] Notice that, since the box query distance is not a metric function and since, in general, the dataset might not be stored, we cannot rely on usual data structures, like k-d trees.

[8] The time for computing the query result is not included since it is equal for both the approaches.

Table 1
Selected queries

IdQ	Query	d	%sel
$Q_{2,h}$	age ≤ 46 AND education_num ≥ 14	2	5%
$Q_{2,m}$	hours_per_week ≥ 41 AND age ≥ 38	2	16%
$Q_{2,l}$	education_num ≤ 11 AND hours_per_week ≤ 40	2	54%
$Q_{3,h}$	education_num ≥ 13 AND age ≤ 34 AND hours_per_week ≤ 40	3	5%
$Q_{3,m}$	age ≤ 54 AND education_num ≥ 13 AND capital_gain ≤ 3000	3	20%
$Q_{3,l}$	age ≤ 39 AND hours_per_week ≤ 40 AND capital_loss ≤ 2500	3	40%
$Q_{4,h}$	age ≤ 34 AND education_num ≥ 13 AND hours_per_week ≤ 40 AND capital_loss ≤ 1400	4	5%
$Q_{4,m}$	capital_gain ≤ 1500 AND age ≤ 34 AND capital_loss ≤ 500 AND hours_per_week ≥ 38	4	28%
$Q_{4,l}$	education_num ≤ 13 AND hours_per_week ≥ 32 AND capital_gain ≤ 450 AND age ≤ 49	4	57%

Table 2
Problem instances

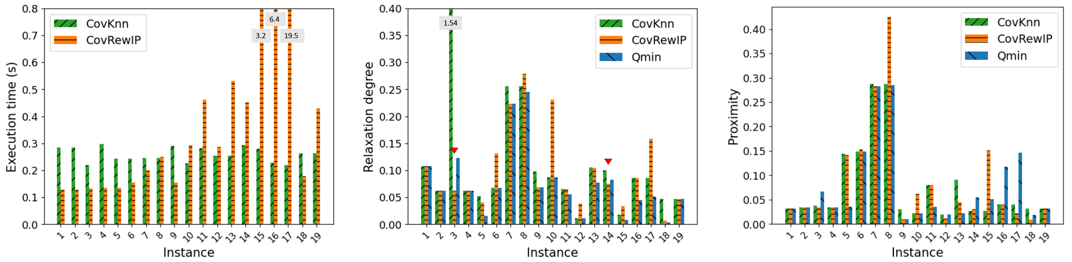
IdI	IdQ	C	$card(Q(I))$	$card(Q \downarrow_{C_i} (I))$ [#added]	#space
I1	$Q_{2,h}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 780$	2,447	730 [50]	15,462
I2	$Q_{2,m}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 1450$	7,962	1387 [63]	14,805
I3	$Q_{2,m}$	$\downarrow^{\texttt{sex,marital\_status}}_{\texttt{female,married-civ-spouse}} \geq 280$	7,962	249 [31]	2,231
I4	$Q_{2,m}$	$\downarrow^{\texttt{marital\_status}}_{\texttt{married-civ-spouse}} \geq 5700$	7,962	5,545 [155]	16,834
I5	$Q_{2,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 10,600$	26,504	10,525 [75]	5,667
I6	$Q_{2,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 11,200$	26,504	10,525 [675]	5,667
I7	$Q_{2,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 12,000$	26,504	10,525 [1475]	5,667
I8	$Q_{2,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 13,000$	26,504	10,525 [2475]	5,667
I9	$Q_{3,h}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 1250$	2,603	1,197 [53]	14,995
I10	$Q_{3,h}$	$\downarrow^{\texttt{race}}_{\texttt{black}} \geq 210$	2,603	194 [16]	4,491
I11	$Q_{3,m}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 3100$	9,186	2,960 [140]	13,232
I12	$Q_{3,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 8500$	20,064	8,444 [56]	7,748
I13	$Q_{3,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 9000$	20,064	8,444 [556]	7,748
I14	$Q_{4,h}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 1230$	2,507	1,159 [71]	15,033
I15	$Q_{4,m}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 4400$	13,541	4,330 [70]	11,862
I16	$Q_{4,m}$	$\downarrow^{\texttt{race}}_{\texttt{black}} \geq 1450$	13,541	1,369 [81]	3,316
I17	$Q_{4,m}$	$\downarrow^{\texttt{sex,race}}_{\texttt{female,black}} \geq 680$	13,541	633 [47]	1,675
I18	$Q_{4,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 8650$	27,606	8,594 [56]	7,598
I19	$Q_{4,l}$	$\downarrow^{\texttt{sex}}_{\texttt{female}} \geq 9100$	27,606	8,594 [506]	7,598

not reported here for space constraints, show that its average performance is worst than those of the approximate approaches, especially for large-scale datasets.

5.2. Experimental results

Impact of the search space size on performance. From Figure 4(a), we observe that, as expected, the time increases while increasing the size of the search space *m*, for a fixed number of selection conditions. The search space size can increase by either considering queries with the same or similar selectivity and varying the considered protected group (e.g., I3 < I2 < I4; I10 <I9; I17 < I16 < I15) or keeping the same protected group for queries with different selectivity (e.g., I5 < I1; I12 < I9; I18 < I14). This behaviour does not hold for covRewIP since its performance does not depend on the number of protected instances.

Impact of the search space dimension on performance. As expected, covKnn time increases also when increasing the number of selection conditions (i.e, the dimension of the search space)



(a) Execution time

(b) Relaxation degree

(c) Proximity

Figure 4: Comparison with respect to efficiency and effectiveness

for a given dataset size (e.g., I1 < I9 < I14). The increase however is limited and is due to the distance computation on points with a higher number of coordinates. On the other hand, covRewIP performance, depending on the applied pruning, might exponentially increase while increasing *d* and, as a consequence, the size of the discretized search space (see, e.g., I15, I16, I17). covKnn thus performs better with higher dimensions; for a low number of selection conditions this behaviour is less evident, due to the high cost for reading the whole dataset in covKnn.

Impact of coverage constraints on performance. Figure 4 shows that the coverage constraint threshold, and thus the number of nearest neighbours to be detected, has a limited impact on covKnn performance. This can be observed by changing the threshold while keeping the same query and the same protected group (see, e.g., I5-I8). An opposite behaviour can be observed for covRewIP: performance decreases while increasing the threshold since a larger subset of the discretized search space has to be visited. Experiments that are not reported here for space constraints show that covKnn performance is sensible to the number of coverage constraints since, for each of them, a specific search has to be computed on a distinct search space (see, e.g., I3 and I4). This does not hold for covRewIP.

Accuracy. Figures 4(b) and (c) compare covKnn, covRewIP, and the precise solution Q_{min}, computed using the baseline approach, with respect to relaxation degree and proximity. Notice that, since covRewIP relies on sampling for estimations, the rewritten query might not satisfy the constraint (see I3 and I14). In general, the covKnn and covRewIP accuracy is quite good and often the generated solutions coincide with the optimal ones (see I1, I2, I4, I19), which obviously always achieve the lowest relaxation degree (and the lower proximity to equal relaxation, see Section 2). When the solutions do not coincide, covRewIP often achieves the lowest relaxation degree, with an average 2.2% additional relaxation with respect to the precise solution (8.1% for covKnn). This is because, differently from covKnn, it takes minimality into account. Situations for which the covRewIP relaxation degree is higher than the covKnn one probably correspond to estimation errors due to the sample usage and the search space discretization.

6. Related Work

Coverage constraints have been first introduced in [4] in the context of data repair. When protected categories are defined in terms of many attributes, the identification of attribute patterns associated with coverage problems might lead to performance issues, due to their combinatorial explosion. Efficient techniques, inspired from set enumeration and association rule mining, have therefore been presented. Once the lack of coverage has been identified, the smallest number of data points needed to hit all the "uncovered spaces" has to be identified and additional data acquired. Since data acquisition has a cost, techniques have been presented for determining the patterns that can be covered given a certain maximum cost [4]. An efficient approach for coverage analysis, given a set of attributes across multiple tables, is presented in [12]. The approaches in [4, 12] deal with categorical attributes with low cardinality; the coverage-based data repair problem has been extended to ordinal and continuous-valued attributes in [5]. The detection of groups, defined by the intersection of multiple attributes, with a low coverage is also at the basis of the MithraCoverage web application [10].

Coverage-based queries, presented in this paper, rely on rewriting to avoid disparate treatment discrimination during selection-based query execution. Another non-discrimination aware technique based on rewriting has been proposed in [13] for OLAP queries. Here, the bias is defined in terms of causal fairness (checking for causal relationships from the sensitive attributes to the outcome). A fairness-aware rewriting approach for range queries has been recently proposed [17], where fairness constraints correspond to the desired distribution for the groups of interest. The approach relies on binary sensitive attributes and a specialized index data structure. As far as we know and according to [16], no further approaches have been proposed so far for coverage-based rewriting of data transformations.

7. Concluding remarks

In this paper, we summarize our work on nondiscrimination-aware data transformation. We first introduce coverage-based queries as a means for guaranteeing the satisfaction of a set of coverage constraints on a selection-based query result, through rewriting. We then present two approximate algorithms for coverage-based query processing: the first, covRew, was already presented in [2]; the second, covKnn, relies on a nearest neighbour approach and is proposed in this paper with the aim of overcoming some limitations of covRew. The techniques are experimentally compared with respect to efficiency and effectiveness. The obtained results, some of which are not reported in this paper for space constraints, show that, for small or medium size datasets, covKnn can generate approximate solutions faster than covRew, at the price of a higher relaxation. On the other hand, covRew could be the right choice for huge datasets, thanks to the sample-based approach, and when we want to minimize the achieved relaxation. We are currently working on the design and evaluation of algorithms for the computation of the optimal rewriting, in terms of minimality and proximity, and on the integration of coverage-based queries in a relational DBMS.

References

[1] C. Accinelli, B. Catania, G. Guerrini, and S. Minisi. The impact of rewriting on coverage constraint satisfaction. In *Proc. EDBT/ICDT Workshops*, 2021.

[2] C. Accinelli, B. Catania, G. Guerrini, and S. Minisi. A coverage-based approach to nondiscrimination-aware data transformation. *ACM J. Data Inf. Qual.*, 2022.

[3] C. Accinelli, S. Minisi, and B. Catania. Coverage-based rewriting for data preparation. In *Proc. EDBT/ICDT Workshops*, 2020.

[4] A. Asudeh, Z. Jin, and H. V. Jagadish. Assessing and remedying coverage for a given dataset. In *Proc. ICDE*, pages 554–565, 2019.

[5] A. Asudeh, N. Shahbazi, Z. Jin, and H. V. Jagadish. Identifying insufficient data coverage for ordinal continuous-valued attributes. In *Proc. SIGMOD*, pages 129–141, 2021.

[6] S. Barocas and A. D. Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[7] B. Catania, G. Guerrini, and C. Accinelli. Fairness & friends in the data science era. *AI & Society*, 2022.

[8] M. Drosou, H. V. Jagadish, E. Pitoura, and J. Stoyanovich. Diversity in big data: A review. *Big Data*, 5(2):73–84, 2017.

[9] D. Firmani, L. Tanca, and R. Torlone. Ethical dimensions for data quality. *ACM J. Data Inf. Qual.*, 12(1):2:1–2:5, 2020.

[10] Z. Jin, M. Xu, C. Sun, A. Asudeh, and H. V. Jagadish. MithraCoverage: A system for investigating population bias for intersectional fairness. In *Proc. SIGMOD*, pages 2721–2724, 2020.

[11] A. Kadlag, A. V. Wanjari, J. Freire, and J. R. Haritsa. Supporting exploratory queries in databases. In *Proc. DASFAA*, pages 594–605, 2004.

[12] Y. Lin, Y. Guan, A. Asudeh, and H. V. Jagadish. Identifying insufficient data coverage in databases with multiple relations. *Proc. VLDB Endow.*, 13(11):2229–2242, 2020.

[13] B. Salimi, J. Gehrke, and D. Suciu. Bias in OLAP queries: Detection, explanation, and removal. In *Proc. SIGMOD*, pages 1021–1035, 2018.

[14] B. Salimi, B. Howe, and D. Suciu. Data management for causal algorithmic fairness. *IEEE Data Eng. Bull.*, 42(3):24–35, 2019.

[15] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *Proc. SIGMOD*, pages 793–810, 2019.

[16] N. Shahbazi, Y. Lin, A. Asudeh, and H. V. Jagadish. A survey on techniques for identifying and resolving representation bias in data. *CoRR*, abs/2203.11852, 2022.

[17] S. Shetiya, I. P. Swift, A. Asudeh, and G. Das. Fairness-aware range queries for selecting unbiased data. In *Proc. ICDE*, 2022.

[18] J. Stoyanovich, B. Howe, and H. V. Jagadish. Responsible data management. *Proc. VLDB Endow.*, 13(12):3474–3488, 2020.

[19] J. Stoyanovich, K. Yang, and H. V. Jagadish. Online set selection with fairness and diversity constraints. In *Proc. EDBT*, pages 241–252, 2018.

[20] S. Sudman. *Applied Sampling*. Academic Press, inc., 1976.

[21] M. Zehlike, K. Yang, and J. Stoyanovich. Fairness in ranking: A survey. *CoRR*, abs/2103.14000, 2021.