Bias Analysis and Mitigation in Data-Driven Tools Using Provenance

Yuval Moskovitch
University of Michigan
yuvalm@umich.edu

Jinyang Li
University of Michigan
jinyli@umich.edu

H. V. Jagadish
University of Michigan
jag@umich.edu

ABSTRACT

Fairness and bias mitigation in data-driven systems has been extensively studied in recent years. In this paper, we suggest a novel approach towards fairness analysis and bias mitigation utilizing the notion of provenance, which was shown to be useful for similar tasks in the context of data and process analyses. We illustrate the idea using a simple use-case demonstrating a scenario of mitigating bias caused by inadequate minority group representation. We conclude with an outline of opportunities and challenges in developing provenance-based solutions for bias analysis and mitigation in data-driven systems.

ACM Reference Format:
Yuval Moskovitch, Jinyang Li, and H. V. Jagadish. 2022. Bias Analysis and Mitigation in Data-Driven Tools Using Provenance. In *14th International Theory and Practice of Provenance (TaPP'22), June 17, 2022, Philadelphia, PA, USA*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3530800.3534528

1 INTRODUCTION

Data-driven tools are widely used these days. These tools are gradually supplanting humans in a wide range of application domains, from deciding who should get a loan [1], to automated hiring [2], students grading [3], and even in assessing the risk of paroling convicted criminals [4]. With the increasing use of data-driven tools, we also witness many cases where these tools are biased. The increasing impact of data-driven methods on society and their effect on human life, has given rise to increasing interest in the study of algorithmic fairness and bias mitigation.

The data used in the development of data-driven decision systems typically undergoes numerous phases, and bias may be introduced at different points [23, 27]. Biased results may stem from historical bias. For instance, a tool developed by Amazon[1] for hiring was discriminating against women, since the data used for training was resumes submitted to the company over a 10-year period. Most came from men, a reflection of male dominance across the tech industry. Insufficient data representation of minority groups, lack of diversity in data sets and data skews are other sources of bias [5, 9].

[1]https://www.businessinsider.com/amazon-built-ai-to-hire-people-discriminated-against-women-2018-10

TaPP'22, June 17, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9349-2/22/06…$15.00
https://doi.org/10.1145/3530800.3534528

id	Gender	Major	GPA	Internship Hours (IH)
1	M	CS	3.75	125
2	F	AM	3.8	150
3	F	IS	3.5	125
4	M	AM	3.9	90
5	F	CS	3.8	120
6	F	CS	3.8	80
7	M	IS	3.5	125
8	M	CS	3.5	95
9	M	CS	3.8	130
10	F	IS	3.75	140
11	M	CS	3.85	115
12	F	CS	3.65	125

Figure 1: Job Applicants Dataset

Representation issues may present in the collected data, arise from (over/under) sampling, or emerge as a result of pre-processing and data cleaning [6, 15]. Bias may also be caused by generalizing from detailed data to statements in a broader context [21, 26]. For instance, doctors have over-diagnosed ADHD for years after making generalizations to age, sex, the maturity of the children[2].

We envision the use of provenance to aid data scientists in the process of fairness analysis and bias mitigation. Provenance based solutions would provide explanations, detect the origin of unfairness, and provide the user actionable information to mitigate bias efficiently. Intuitively, fairness is a property of the dataset that is induced by the tuples composing it. Namely, values of individual tuples in the data are combined to compute a fairness measure. For instance, data representation, dictated by the individuals in the data, affects the fairness measures of a data-driven system. Provenance encapsulates information about items in the data set and the effect of data manipulation on them. Particularly, it may be used to determine or explain the effect of such manipulation on the existence of tuples in the data or their value, which in turn can be used to determine the effect on the representation of different groups in the data. Thus, provenance can be useful in detecting or explaining steps in the computation that alter the fairness of the data.

In what follows, we demonstrate the idea with a simple use-case, showing the use of provenance in maintaining an inadequate representation of protected groups in the data where the data transformation includes only relational algebra operations. We then outline opportunities and challenges in the pathway towards provenance-based solutions for more intricate cases.

2 RELATIONAL ALGEBRA AND REPRESENTATION CONSTRAINTS

We (informally) present a simple use-case, demonstrating the use of provenance in maintaining adequate groups representation through a running example.

[2]https://www.medicalnewstoday.com/articles/325595##age-related-factors

Example 2.1. Consider a tech company that is looking to hire new employees. The company is looking for applicants with a degree in Computer Science and a GPA of at least 3.7 who have at least 100 hours internship experience at the company. The company's data analyst uses the candidates data shown in Figure 1 to extract potential candidates for a job interview. The applicants dataset consists of candidates who had internships at the company and majored in Computer Science (CS) or other related fields, such as Applied Math (AM), and Information Systems (IS). It includes information regarding the candidates: gender, major, GPA and internship hours. The following query can be used to select candidates that meet the company's requirement form the dataset.

```
SELECT *
FROM Applicants AS a
WHERE a.Major='CS' AND a.GPA >= 3.7 AND a.IH >= 100
```

Candidates selected by the query are highlighted.

The increasing interest and awareness for diversity and underrepresented minority groups in a variety of positions, affect organizations, particularly in their recruitment process. Many law firms, government agencies, courts, and nonprofit organizations facilitate hiring programs based on diversity. E.g., diversity statements have become an integral part of the materials submitted as part of applications for academic faculty positions. Continuing with our running example, note that among the four selected applicants, there is a single female candidate. To increase diversity, the company may wish to increase the number of females invited for a job interview. For example, they may decide to have at least three females invited. This can be achieved by relaxing the criteria set by the company for an interview. E.g., selecting applicants who have majored in other fields, and/or loosening the conditions over the GPA and/or the candidates' internship hours.

Intuitively, we wish to allow users to pose constraints on the cardinality of groups in the data in the result of a sequence of data manipulation queries. These constraints may express lower bounds on minority group representation, as in the above example, but would also allow users to set upper bounds on group representation, e.g., to avoid data skews. Constraints could also be used to state proportions between groups in the data. For instance, the number of black female should be no less than the number of white male in the output. This problem, extends the notion of query refinement, which was studied in the context of SQL queries in, e.g., [19, 24]. While our problem is more general since it considers a *set of constraints on groups* in the data, we build on the notion of query refinement defined in [24] to formulate the problem.

2.1 Problem Formulation

We consider SPJU queries with selection predicates that include range ($<, \leq, \geq, >$) and equality ($=$). Predicates can be defined on numeric or categorical attributes. We use the definition of [24] for refinement of numeric and categorical predicates. Due to space constraints and for ease of presentation, in the rest of the paper we assume the numeric predicates are in the form of $A_i \geq C_i$ where A_i is an attribute and C_i is a constant, and consider only constraints on the minimal representation of groups in the result of a query. Therefor we focus on query *relaxation*, i.e., increasing the number of tuples in groups satisfying a predicate. Our solution extends to a set of (general) constraints on groups representation in the result of a of query supporting all the aforementioned forms of predicates.

The notation of relaxation as defined in [24] distinguishes between numeric and categorical predicates. Given a numeric predicate $P_i : A_i \geq C_i$ (over an attribute A_i), a relaxation of P_i is any predicate $P_i' : A_i \geq C_i'$ where $C_i \geq C_i'$. Categorical attribute permit only equality predicates and their relaxation is done through the notion of *expansion*[3], the process of disjunctively adding additional predicates to a categorical predicate. Namely, if P_i is a categorical predicate over an attribute A_i, a relaxation P_i' is an expansion of P_i. Finally, a query Q' with predicates $P_1', \ldots, P_k'$ is a relaxation of the query Q with predicates $P_1, \ldots, P_k$ if P_i' is a relaxation of P_i.

Example 2.2. Consider again the query Q given in Example 2.1. The following query Q' relaxes Q with respect to the categorical attribute Major and the numerical attribute GPA.

```
SELECT *
FROM Applicants AS a
WHERE (a.Major='CS' or a.Major='IS')
       AND a.GPA >= 3.65 AND a.IH >= 100
```

Data groups and representation constraints. Let D be a dataset, Q a query and $Q(D)$ the result of executing Q over D. We define groups in $Q(D)$ using a conjunction of conditions $\mathcal{G} = \wedge_i(A_i \; op \; v_i)$ where A_i are distinct data attributes in $Q(D)$, and op can be one of $\{=, \leq, <, \geq, >\}$. For instance, in our running example $\mathcal{G}$ is the condition $Gender = F$. We use $Q(D)_{\mathcal{G}}$ to denote the data tuples in $Q(D)$ that satisfy the condition $\mathcal{G}$. A representation constraint over $Q(D)_{\mathcal{G}}$ is an expression of the form $Q(D)_{\mathcal{G}} \; op \; x$, where $op \in \{\leq, <, >, \geq\}$ and x can be a constant or $\alpha \cdot Q(D)_{\mathcal{G}'}$ for some other data groups defined using $\mathcal{G}'$ and $\alpha \in \mathbb{R}$. As mentioned above, for simplicity we assume a single constraint of the form $Q(D)_{\mathcal{G}} \geq x$ where x is a constant, but the solution extends to set of any type of the aforementioned constraints. Given a set of constraints, there may be multiple ways to relax a query in order to fulfill them.

Example 2.3. Continuing with our example, the constraint over the number of female student is $Q(D)_{Gender=F} \geq 3$. The result of the query Q' given in Example 2.2, a relaxation of Q from Example 2.1, satisfies the constraint. Other plausible relaxations of Q that satisfy the constrain are the query relaxation that relaxes the Major predicate to be `(a.Major='CS' or a.Major='IS' or a.Major='AM')`, or the relaxation that relaxes the predicate GPA to be `a.GPA >= 3.65` and the IH predicate to be `a.IH >= 80`.

The relaxations depicted in the above example suggest the company various ways to achieve its diversity goal. Each applies different modifications to the query. Intuitively, minimal modifications to the original query are preferred, e.g., a relaxation that relaxes the predicate GPA to be `a.GPA >= 3.65` is preferred over one that modifies the predicate GPA to be `a.GPA >= 3.5`, however, relaxations that modify different attributes may be incomparable when no additional preferences information is provided by the end-user. To this, we are interested in the set of minimal relaxations.

Given a query with a set of predicate $\mathcal{P}$ and their relaxation $\mathcal{P}'$, we say that $\mathcal{P}'$ *satisfies* a constraint if the output of the corresponding relaxed query satisfies the constraint. Given a constraint c, a

[3]The categorical relaxation of [24] also include the roll-up. For simplicity we do not consider hierarchy over categorical attribute, however our model can support roll-up relaxations as well.

minimal relaxation of $\mathcal{P}$ is a relaxation $\mathcal{P}'$ such that $\mathcal{P}'$ satisfies c, and there is no relaxation of $\mathcal{P}'$ such that $\mathcal{P}'' \neq \mathcal{P}'$ and $\mathcal{P}''$ satisfies c. Our goal is then to find all minimal relaxations satisfying the given constraint. Intuitively, these relaxations form a skyline [7]. We note that there may be many minimal relaxations, this could be addressed by exploiting methods for refining, reducing and ranking the set of skyline points (see, e.g., [20]).

Our proposed solution utilizes provenance to find the set of minimal relaxations. In particular we leverage the notion of hypothetical reasoning [11] to examine the effect of possible relaxations on the outcome of the query.

2.2 Provenance Model

We next depict our provenance model. We leverage the idea of conditional tables (c-tables) [17], where tuples are associated with conditions. To capture the possible relaxations, we annotate tuples in the data with the query selection conditions. Finally, we follow the semiring model [16] to propagate the annotations through query evaluation. Intuitively, when applying a selection predicate over the data, instead of actually deleting tuples that do not meet the selection criteria, we annotate each tuple t in the data with a variable (condition) v such that v is evaluated to 1 if t satisfies the selection predicate and 0 otherwise. Namely, if Q is a selection query over D with predicates (numerical and categorical) over the attributes $A_1, \ldots, A_k$, we annotate each tuple $t \in Q(D)$ as $prov(t) = \prod_{i=1}^{i=k} A_{i[t.A_i]}$ Where $[t.A_i]$ denotes the value of the attribute A_i in t. Using the resulting provenance expression we can construct inequality expressions that express the constraint. The provenance inequality of the constraint $Q(D)_{\mathcal{G}} \geq x$ is $\sum_{t \in Q(D)_{\mathcal{G}}} prov(t) \geq x$

Example 2.4. Consider again our running example and the constraint $Q(D)_{Gender=F} \geq 3$. The corresponding provenance inequality is $M_{AM} \cdot G_{3.8} \cdot IH_{150} + M_{IS} \cdot G_{3.5} \cdot IH_{125} + M_{CS} \cdot G_{3.8} \cdot IH_{120} + M_{CS} \cdot G_{3.8} \cdot IH_{80} + M_{IS} \cdot G_{3.75} \cdot IH_{140} + M_{CS} \cdot G_{3.65} \cdot IH_{125} \geq 3$. Where M and G are short for Major and GPA respectively.

Relaxations through valuation. We now establish the connection between relaxations and valuation of the provenance inequality p of a constraint. A relaxation $\mathcal{P}$ that satisfies a given constraint should correspond to a valuation that satisfies the inequality. Recall that the set of variables A_{iv} in p correspond to the predicates in $\mathcal{P}$. The valuation assign a value of 0 or 1 to each such variable as follows. If A_i is a numerical attribute and P_i is $A_i \geq C_i$ then $val_{\mathcal{P}}(A_{iv}) = 1$ if $v \geq C_i$ and 0 otherwise . If A_i is a categorical attribute and P_i is a predicate over A_i then $val_{\mathcal{P}}(A_{iv}) = 1$ if v satisfies P_i and 0 otherwise. We denote by $T_{\mathcal{P}}(p)$ the truth value of the inequality resulting by applying $val_{\mathcal{P}}(A_{iv})$ on each variable A_{iv} in p.

Example 2.5. The truth value $T_{\mathcal{P}}(p)$ of the provenance inequality p from Example 2.4 and set of predicates $\mathcal{P}$ appearing in the query presented in Example 2.1 is false since the only term in p evaluated to 1 is $M_{CS} \cdot G_{3.8} \cdot IH_{120}$ and $1 \not\geq 3$. The result of the valuation using the set of predicates $\mathcal{P}'$ from the query Q' depicted in Example 2.2 is true since the terms $M_{IS} \cdot G_{3.75} \cdot IH_{140}$ and $M_{CS} \cdot G_{3.65} \cdot IH_{125}$ are also evaluated to 1.

We can show the following property of the model.

Proposition 2.6. *Let D be a dataset and Q be a query. Q satisfies a constraint $Q(D)_{\mathcal{G}} \geq x$ if and only if $T_{\mathcal{P}}(p) = True$ where p is the provenance inequality of the constraint over $Q(D)$ and $\mathcal{P}$ is the set of predicates appearing in Q.*

	ΔG	ΔIH	ΔM_{IS}	ΔM_{AM}
1	m_1	m_1	m_4	m_4
2	m_4	m_5	m_6	m_6
3	m_5	m_6	m_5	m_1
4	m_6	m_2	m_2	m_5
5	m_2	m_4	m_1	m_2

Figure 2: Minimal changes table. m_i corresponds to the i'th element in the provenance inequality given in Example 2.4.

Given the provenance inequality of the constraint over $Q(D)$ we can examine the effect of query relaxations on the constraint satisfaction without the need to access the data and execute the potential query relaxations. Furthermore, the provenance inequality may guide the relaxations search as we next explain.

2.3 Generating Minimal Relaxations

We next present a method for generation of minimal relaxations. Intuitively, given a set of predicates $\mathcal{P}$ and a provenance inequality p, if $T_{\mathcal{P}}(p) = False$, a minimal relaxation $\mathcal{P}'$ increases the number of terms in p that are evaluated to 1. For each term m_l in p such that m_l is evaluated to 0, we can derive the minimal relaxation $\mathcal{P}'_{m_l}$ required to flip the evaluation of m_l, and the minimal changes in this relaxation with respect to the **attributes** of numeric predicates and **values** of categorical predicates. For numeric predicates, the minimal change between $P_i : A_i \geq C_i$ in $\mathcal{P}$ and $P'_i : A_i \geq C'_i$ in $\mathcal{P}'$ is $C_i - C'_i$. For categorical predicates, we consider a minimal change with respect to each possible attribute value, so if P_i is a categorical predicate over A_i, the minimal change with respect to the (possible attribute) value v is 0 if v satisfies both P_i and P'_i, 1 if v satisfies P'_i but not P_i, and ∞ otherwise.

Example 2.7. Consider the provenance inequality shown in Example 2.4. The minimal required relaxation to flip the valuation of the first term $m_1 = M_{AM} \cdot G_{3.8} \cdot IH_{150}$ is to relax the predicate Major to be `(a.Major='CS' or a.Major='AM')`. The minimal changes with respect to G and IH are 0 and the minimal change with respect to M_{AM} is 1 (and 0 with respect to M_{CS} and ∞ for M_{IS}).

The solution is based on Fagin's Threshold Algorithm [13]. We sort the terms in the provenance inequality based on the minimal changes in the minimal relaxation required to flip their evaluation with respect to each numeric attribute and categorical value. To this end we use the minimal changes table (MCT) with a column ΔA_i for each numeric attribute and ΔA_{iv} for each value v of a categorical attribute. The values in MCT are terms in the provenance inequality sorted in an ascending order by their minimal change with respect to each column. Ties are broken based on their order in the leftmost column of MCT, and for the leftmost column, arbitrarily. Figure 2 depicts the table constructed for our running example. Since we consider only a simple case of relaxation, columns corresponding to categorical predicates that appear in the predicate set (ΔM_{CS}) and terms that are evaluated to 1 by predicates set in the given query (m_3) are omitted from the table.

Searching with the MCT. To search for minimal relaxations, we traverse the MCT in a left-right top-down fashion. The algorithm maintains a result set $\mathcal{R}$, adding (and removing) relaxations to (and

from) $\mathcal{R}$ as it traverses the table. For each term m_l in the table, the algorithm first computes the minimal relaxation required $\mathcal{P}'_{m_l}$ to flip the evaluation of m_l. If $\mathcal{P}'_{m_l}$ is a satisfying relaxation then it is compared with the previous relaxations found in $\mathcal{R}$, to determine the new set $\mathcal{R}$. Otherwise, let r and c be the current cell's row and column indices in the search respectively. The algorithm computes the minimal relaxation required $\mathcal{P}'_{\mathcal{M}}$ to flip the evaluation of all the terms in any subset of terms $\mathcal{M}$ appearing in column c and rows $1, \ldots, r$ which include the term m_l. If for any such subgroup $\mathcal{M}$, $\mathcal{P}'_{\mathcal{M}}$ is a satisfying relaxation, $\mathcal{P}'_{\mathcal{M}}$ is used to update $\mathcal{R}$. Once a first minimal relaxation $\mathcal{P}'_{\mathcal{M}}$ was found, where $\mathcal{M}$ is either a single term or a set of terms, we can compute a *stop line*, which intuitively indicates areas of the table that are not relevant for the search since they are not minimal with respect to the found relaxation. For each column c in the table, the stop line is the maximal row r such that $m_l \in \mathcal{M}$ appears in row r and column c of the table.

Example 2.8. Consider again our running example and the *MCT* depicted in Figure 2. The algorithm first considers the relaxation $\mathcal{P}'_{m_1}$ shown in Example 2.7. This is not a satisfying relaxation, thus the algorithm continues to the next cells in the first row. None of the corresponding relaxations are satisfying. Next, the algorithm start traversing the second row of the table. Since $\mathcal{P}_{m_4}$ is not a satisfying relaxation, it considers the relaxation $\mathcal{P}_{\{m_4,m_1\}}$ which relaxes the predicate Major and IH. This is a satisfying relaxation, and therefore it is added to the result set $\mathcal{R}$. At this point, the stop line (marked in blue in the table) is computed. The grey part of the table can be avoided by the algorithm.

3 CHALLENGES AND OPPORTUNITIES

We have demonstrated the use of provenance for bias mitigation in the context of group representation in the result of SPJU queries. Our solution, inspired by the use of provenance for hypothetical reasoning, utilizes the semiring model to generate provenance inequalities that corresponds to constraints over the representation of groups in the data. We then use these inequalities to find minimal relaxations that satisfy the constraints with an algorithm based on Fagin's Threshold Algorithm. We next highlight intriguing direction as well as challenges in the development of provenance-based solution for bias analysis and mitigation in data-driven tools.

Data-driven tools and pipelines comprise multiple phases, e.g., data collection or integration, cleaning and analysis, model training, and result evaluation and analysis. To enable bias mitigation across these phases models for complex transformation languages are necessary. Provenance has been studied for different query languages and data transformations (see, e.g., [8, 10, 14, 16]). These solutions could serve as the inspiration for such provenance based methods. A major challenge in this regard is handling the model training phase. We note that [29] has propose the use of provenance for incremental model updates for linear and logistic regression models. Another intriguing direction towards this goal is to leverage the line of works on machine learning algorithm representation using relational algebra [18, 22], for which provenance models exists.

The solution presented in this paper focuses on adequate group representation, which is an important facet of fairness. There is a wealth of study on algoritmic bias, and a broad scope of fairness definitions and use-cases. E.g., the notion of fairness was studied in the context of classification, ranking and recommendation (see, e.g., [25, 28]). Considering group fairness and individual fairness [12], and supporting a wide range of definitions may require new models and probably novel algorithms for provenance usage.

Acknowledgments. This research was supported in part by NSF under grants 1741022, 1934565 and 2106176.

REFERENCES

[1] The algorithm that beats your bank manager. https://www.forbes.com/sites/parmyolson/2011/03/15/the-algorithm-that-beats-your-bank-manager/?sh=4fbafadc1ae9, 2011.
[2] Can an algorithm hire better than a human? https://www.nytimes.com/2015/06/26/upshot/can-an-algorithm-hire-better-than-a-human.html, 2015.
[3] When algorithms give real students imaginary grades. https://www.nytimes.com/2020/09/08/opinion/international-baccalaureate-algorithm-grades.html, 2020.
[4] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica, May*, 23(2016), 2016.
[5] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. Assessing and remedying coverage for a given dataset. In *ICDE*. IEEE, 2019.
[6] Sumon Biswas and Hridesh Rajan. Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline. In *ESEC/FSE*, 2021.
[7] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The skyline operator. In *ICDE*. IEEE Computer Society, 2001.
[8] Pierre Bourhis, Daniel Deutch, and Yuval Moskovitch. Equivalence-invariant algebraic provenance for hyperplane update queries. In *SIGMOD*. ACM, 2020.
[9] Irene Y. Chen, Fredrik D. Johansson, and David A. Sontag. Why is my classifier discriminatory? In *NeurIPS*, 2018.
[10] James Cheney, Laura Chiticariu, and Wang Chiew Tan. Provenance in databases: Why, how, and where. *Found. Trends Databases*, 1(4), 2009.
[11] Daniel Deutch, Zachary G. Ives, Tova Milo, and Val Tannen. Caravan: Provisioning for what-if analysis. In *CIDR*. www.cidrdb.org, 2013.
[12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *ITCS*.
[13] Ronald Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1), 1999.
[14] Floris Geerts, Grigoris Karvounarakis, Vassilis Christophides, and Irini Fundulaki. Algebraic structures for capturing the provenance of SPARQL queries. In *ICDT*, 2013.
[15] Stefan Grafberger, Julia Stoyanovich, and Sebastian Schelter. Lightweight inspection of data preprocessing in native machine learning pipelines. In *CIDR*, 2021.
[16] Todd J. Green, Gregory Karvounarakis, and Val Tannen. Provenance semirings. In *PODS*. ACM, 2007.
[17] Tomasz Imielinski and Witold Lipski Jr. Incomplete information in relational databases. *J. ACM*, 31(4), 1984.
[18] Dimitrije Jankov, Shangyu Luo, Binhang Yuan, Zhuhua Cai, Jia Zou, Chris Jermaine, and Zekai J. Gao. Declarative recursive computation on an RDBMS. *Proc. VLDB Endow.*, 12(7), 2019.
[19] Nick Koudas, Chen Li, Anthony K. H. Tung, and Rares Vernica. Relaxing join and selection queries. In *VLDB*, 2006.
[20] Xuemin Lin, Yidong Yuan, Qing Zhang, and Ying Zhang. Selecting stars: The k most representative skyline operator. In *ICDE*. IEEE Computer Society, 2007.
[21] Yin Lin, Brit Youngmann, Yuval Moskovitch, H. V. Jagadish, and Tova Milo. On detecting cherry-picked generalizations. *Proc. VLDB Endow.*, 15(1), 2021.
[22] Shangyu Luo, Zekai J. Gao, Michael N. Gubanov, Luis Leopoldo Perez, and Christopher M. Jermaine. Scalable linear algebra on a relational database system. In *ICDE*. IEEE Computer Society, 2017.
[23] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), 2021.
[24] Chaitanya Mishra and Nick Koudas. Interactive query refinement. In *EDBT*, 2009.
[25] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: An overview. *CoRR*, abs/2104.05994, 2021.
[26] Babak Salimi, Johannes Gehrke, and Dan Suciu. Bias in OLAP queries: Detection, explanation, and removal. In *SIGMOD*, 2018.
[27] Harini Suresh and John V. Guttag. A framework for understanding unintended consequences of machine learning. *CoRR*, abs/1901.10002, 2019.
[28] Sahil Verma and Julia Rubin. Fairness definitions explained. In *FairWare@ICSE*. ACM, 2018.
[29] Yinjun Wu, Val Tannen, and Susan B. Davidson. Priu: A provenance-based approach for incrementally updating regression models. In *SIGMOD*. ACM, 2020.