Distribution Tailoring Over Highly Overlapping Sources

DRACO XU, JIWON CHANG, YURONG LIU, and ZHIGONG GAO, University of Rochester, USA

Two of the most common challenges in a data analytics pipeline are a lack of suitable dataset and algorithmic bias. The former is often solved through data integration, where data scientists integrate data from multiple data sources into one. Data distribution tailoring (DT) as defined by Nargesian et al. [5] aims to integrate data in a demographically fair manner to address the latter. DT ensures that the unified set satisfies group count requirements. We present an overlapping generalization of the DT problem which allows multiple data sources to contain the same tuple. There are many real-life scenarios where the proportion of overlapping tuples in data sources are nonnegligible. We study how to optimize the cost function with overlaps in mind and present an approximate algorithm that outperforms original algorithm that does not consider overlap probability for highly overlapping data sources.

Additional Key Words and Phrases: datasets, data integration, algorithmic fairness

ACM Reference Format:
Draco Xu, Jiwon Chang, Yurong Liu, and Zhigong Gao. 2022. Distribution Tailoring Over Highly Overlapping Sources. In . ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Two of the most common challenges in a data analytics pipeline are a lack of suitable dataset and algorithmic bias. A common way to solve the former is data integration, where data scientists integrate data from multiple data sources into one [2]. Although there are algorithmic approaches to solving the issue of bias, the majority of surveyed data scientists collect more data as the first preventative measure [3, 4]. As such, Nargesian et al. has proposed methods of collecting data from multiple sources to meet desired distribution requirements [5].

A major limitation of DT is its assumption that there are no overlapping data points [5]. In the real world, some overlap between data sources, especially web sources, is common [1]. Consider the following motivating example: a data scientist wishes to collect a set of scientific papers from several databases [6]. Since many papers are included in multiple databases, there is non-trivial overlap between data sources. The existing solution does not perform optimally if the data scientists are dealing with data sources with high overlap. In this work, we assume that some statistics on overlap proprbility is given, either exactly or through approximation as in [6]. We propose an optimization that avoids sampling from data sources with a high chance of overlap. Our strategy captures the sampling cost, overlap statistics, and group distributions of each data source.

2 MODEL

2.1 Problem Definition

Suppose there is a bag of n data sources $\mathcal{L} = \{D_1, \ldots, D_n\}$. Each data source D_i is a finite set of tuples. Each data source D_i is associated with a cost of sampling C_i. Furthermore, each tuple belongs to one of m groups $\mathcal{G}_1, \ldots, \mathcal{G}_m$. The user supplies a query $\mathrm{Q} = \{Q_1, \ldots, Q_m\}$ where each Q_i is a non-negative integer. The goal of the data tailoring problem is to collect a unified set of tuples O by querying one random tuple from a data source in each iteration to satisfy $\mathbf{Q}$ while minimizing the total query cost $\sum C_i$ [5]. $O = \cup_{i=1}^{n} O_i$, where O_i is the tuples sampled from D_i. In our overlapping generalization, a tuple in D_i may also be in D_j where $i \neq j$. We focus on sampling with replacement.

2.2 Dynamic Programming

First we analyze how an optimal algorithm would behave through dynamic programming. Our goal is to minimize the expected cost $F(\mathbf{Q})$ given count descriptions $Q = \{Q_1, \ldots, Q_m\}$. According to [5], the cost formula is recursively defined as:

$$F(\mathbf{Q}) = \min_{\forall D_i \in \mathcal{L}} \Big(C_i + \sum_{j=1, Q_j>0}^{m} \mathbb{P}_i^j F_j(\mathbf{Q}) + \big(1 - \sum_{j=1, Q_j>0}^{m} \mathbb{P}_i^j \big) F(\mathbf{Q}) \Big)$$

Let $\mathbb{P}_i^j$ be the ratio of tuples from $\mathcal{G}_j$ in source D_i. In each iteration, the program first pays a flat query cost C_i to obtain a random tuple from D_i. Then, if the tuple was not discarded with probability $\sum_{j=1,Q_j>0}^{m} \mathbb{P}_i^j$, we may proceed to compute $F_j(\mathbf{Q})$, which is the expected cost given the query outcome. If the tuple is discarded with probability $1 - \sum_{j=1,Q_j>0}^{m} \mathbb{P}_i^j$ due to its group requirement already being met, then the query cost $F(\mathbf{Q})$ remains the same.

If there are overlapping tuples in data sources, then a random tuple from D_i may be discarded when it already belongs to the collected set O. This probability can be expressed as $\mathbb{P}(O \mid D_i)$, which refers to the ratio of tuples in D_i that have been already collected in O (may not be collected from D_i due to overlapping). Only when a tuple is not discarded due to overlap will its group matter, and the probability is $1 - \mathbb{P}(O \mid D_i)$. Then in this case, we follow the logic of the original formula. Assuming that $\mathbb{P}(O \mid D_i)$ is given, the cost formula for overlapping DT problem is redefined as:

$$F(\mathbf{Q}) = \min_{\forall D_i \in \mathcal{L}} \Big(C_i + \mathbb{P}(O \mid D_i) F(\mathbf{Q}) + \Big(1 - \mathbb{P}(O \mid D_i)\Big) \Big(\sum_{j=1, Q_j>0}^{m} \mathbb{P}_i^j F_j(\mathbf{Q}) + \big(1 - \sum_{j=1, Q_j>0}^{m} \mathbb{P}_i^j \big) F(\mathbf{Q}) \Big) \Big)$$

Solving for $F(\mathbf{Q})$ and providing a base case:

$$F(\mathbf{Q}) = \begin{cases} 0 & \mathbf{Q} = \{0, \ldots, 0\} \\ \min_{\forall D_i \in \mathcal{L}} \left(\frac{C_i + \big(\sum_{j=1,Q_j>0}^{m} \mathbb{P}_i^j F_j(\mathbf{Q})\big)\big(1 + \mathbb{P}(O|D_i)\big)}{\big(\sum_{j=1,Q_j>0}^{m} \mathbb{P}_i^j\big)\big(1 - \mathbb{P}(O|D_i)\big)} \right) & \text{otherwise} \end{cases} \tag{1}$$

Decreasing $\mathbb{P}(O \mid D_i)$ decreases the numerator and increases the denominator. Consequently, expected cost $F(\mathbf{Q})$ is minimized when the probability of overlap is small given all else equal.

Equation (1) assumes that $\mathbb{P}(O \mid D_i)$ is given. It may be computed using pre-supplied overlap statistics and the inclusion-exclusion principle. Notice that $\mathbb{P}(O \mid D_i) = \mathbb{E}(|O \cap D_i|)/|D_i|$. $O \cap D_i$ contains O_i, but also contains tuples that were collected from other data sources that also exists in D_i. That is, we may have collected some tuples from D_i by sampling from other data sources. The following is the formula for deriving $\mathbb{P}(O \mid D_i)$.

$$\mathbb{P}(O \mid D_i) = \frac{1}{|D_i|} \sum_{k=1}^{n} \Big(\sum_{x \in \mathcal{N}^k, x_1 = i} |\cup_{t=x_j, j>1} O_t| \times \frac{A_x^k}{|\cup_{t=x_j, j>1} D_t|} \Big) \tag{2}$$

Proof. The critical task is to evaluate the size of the source. Let A_x^k denote the size of the k-th overlapping, where $x \in \mathcal{N}^k$ is a list of sources. For example, $A_{(1,2)}^2$ is the red area in the Figure 1. Then the $1-th$ size is $|O_i|$. The $2-th$ size is $\sum_{i \neq j} |O_j| \times \frac{A_{(i,j)}^2}{D_j}$. The $3-th$ size is $\sum_{i \neq j,t} |O_j \cup O_t| \times \frac{A_{(i,j,t)}^3}{|D_j \cup D_t|}$. The $k-th$ size is $\sum_{x \in \mathcal{N}^k, x_1 = i} |\cup_{t=x_j, j>1} O_t| \times \frac{A_x^k}{|\cup_{t=x_j,j>1} D_t|}$.

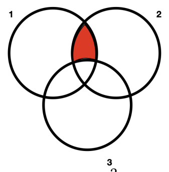


Fig. 1. $A^2_{(1,2)}$

Therefore, the overall size is $\sum_{k=1}^{n}(\sum_{x\in\mathcal{N}^k,x_1=i} \mid \cup_{t=x_j,j>1} O_t| \times \frac{A^k_x}{|\cup_{t=x_j,j>1} D_t|})$. The equation 2 is valid trivially when $k = 1, 2$. When $k = 3$, it could been shown below:

$$\mathbb{P}(O \mid D_1) = \frac{|O_1| + |O_2| \times \frac{|D_1\cap D_2|-|D_1\cap D_2\cap D_3|}{|D_2|} + |O_3| \times \frac{|D_1\cap D_3|-|D_1\cap D_2\cap D_3|}{|D_3|} + (|O_2 \cup O_3|) \times \frac{|D_1\cap D_2\cap D_3|}{|D_2\cup D_3|}}{|D_1|}$$

Let's assume the equation 2 is valid when k, let's prove it for $k + 1$. The only modification we would have to make is the addition of $(k + 1) - th$ size. Let's take the claim that it is $\sum_{x\in\mathcal{N}^{k+1},x_1=i} \mid \cup_{t=x_j,j>1} O_t| \times \frac{A^{k+1}_x}{|\cup_{t=x_j,j>1} D_t|}$, then the equation has been proved for $k + 1$. Therefore the only remaining step is to prove the claim. Let's take a look at it. We would like to know how much of the $\mid \cup_{t=x_j,j>1} O_t|$ falls in to the ones we collected in the D_i, based on the definition of A^k_x, A^{k+1}_x is the size of the $(k + 1) - th$ overlapping (pure), then $\frac{A^{k+1}_x}{|\cup_{t=x_j,j>1} D_t|}$ stands for the uncertain picking from the union of the x. How about the certain part? They have been covered by the lower k, thus it is complete. □

2.3 Selecting Optimal Data Source

Suppose we now have n data sources $D_1, \ldots, D_n$ where each D_i has N_i number of tuples and a cost of sampling C_i. Let the number of tuples of group $\mathcal{G}_j$ in D_i be N^j_i. To simplify notation, we introduce $\mathbb{P}_i$ short for $\mathbb{P}(O \mid D_i)$. Since we would like to minimize the expected cost of choosing a data source, for every group $\mathcal{G}_j$, we redefine $D_{*j,l}$, the optimal data source that requires minimum cost for $\mathcal{G}_j$, as follows

$$D_{*j,l} = \underset{\forall D_i\in\mathcal{L}}{\arg\max}\left(\frac{N^j_i \cdot (1 - \mathbb{P}_i)}{N_i \cdot C_i \cdot (1 + \mathbb{P}_i)}\right) \tag{3}$$

where O^j represents the number of tuples collected for $\mathcal{G}_j$ so far, O^j_i represents the number of tuples of $\mathcal{G}$ collected from D_i, and

$$\mathbb{P}(O \mid D_i) = \frac{1}{|D_i|}\sum_{k=1}^{n}(\sum_{x\in\mathcal{N}^k,x_1=i} \mid \cup_{t=x_j,j>1} O_t| \times \frac{A^k_x}{\mid \cup_{t=x_j,j>1} D_t|}) \tag{4}$$

In order to minimize the cost, we first need to ensure C_i which is on the denominator as small as possible. Then since an optimal selected data source for group $\mathcal{G}$ requires higher ratio of undiscovered tuples in it for $\mathcal{G}$, we maximum the term $\frac{N^J_i}{N_i}$. Finally, following the cost formula we came up in the previous section, we use the overlapping information by adding the term $\frac{1-\mathbb{P}(O|D_i)}{1+\mathbb{P}(O|D_i)}$ to optimize our choice for each group. It then follows from Theorem 1 from the original paper[5] that in each iteration, after identifying the group of which the most cost effective data source requires the maximum expected cost, i.e. min $D_{*j,l}$, defined as the minority group, we choose its corresponding optimal data source, and this method gives us the optimal solution.

Algorithm 1 GeneralDT

```
Require: Group counts $Q_1, \cdots, Q_m$; data sources $\mathcal{L} = \{D_1, \ldots, D_n\}$; $\{\mathbb{P}_1, \ldots, \mathbb{P}_n\}$
Ensure: $O$, target data set
 1: $O \leftarrow \{\}$
 2: $O_i^j \leftarrow 0, \forall 1 \leq i \leq n, 1 \leq j \leq m$
 3: while $\exists Q_j > 0, \forall 1 \leq j \leq m$ do
 4:     $min \leftarrow \infty$
 5:     for $j = 1$ to $m$ do
 6:         if $Q_j == 0$ then continue
 7:         $x \leftarrow \arg\max_{\forall D_i \in \mathcal{L}} \left( \frac{N_i^j \cdot (1-\mathbb{P}_i)}{N_i \cdot C_i \cdot (1+\mathbb{P}_i)} \right)$;
 8:         if $\frac{N_x^j \cdot (1-\mathbb{P}_i)}{N_x \cdot C_x \cdot (1+\mathbb{P}_i)} < min$ then
 9:             $i \leftarrow x$;
10:             $min \leftarrow \frac{N_x^j \cdot (1-\mathbb{P}_i)}{N_x \cdot C_x \cdot (1+\mathbb{P}_i)}$;
11:     $s \leftarrow$ Query$(D_i)$
12:     $j \leftarrow \mathcal{G}(s)$
13:     if $(s \notin O$ AND $Q_j > 0)$ then
14:         add $s$ to $O$; $Q_j \leftarrow Q_j - 1$; $O_i^j \leftarrow O_i^j + 1$;
15:         for each $D_i \in \mathcal{L}$ do
16:             update $\mathbb{P}(O \mid D_i)$
17: return $O$
```

3 APPROXIMATION ALGORITHM

Once we figured out the cost formulas and the probability $\mathbb{P}(O \mid D_i)$, we come up with our optimal algorithm for known distribution case with information on overlapping. Algorithm 1 shows the pseudocode. At each iteration, we first iterate through all the groups and use the formula derived above to calculate out the optimal data source for each group. During this process, we keep updating the group that has maximum expected cost to selecting from its optimal data source as well as the data source itself. Afterwards we just randomly sample a tuple from the chosen data source and accept it if it's not selected before. If the tuple is accepted, we then update Q_j, O_i^j, as well as $\mathbb{P}(O \mid D_i)$ for each data source.

4 EXPERIMENTS

4.1 Implementation

Our source code [7] implements the known binary DT and generalized coupon collector's algorithms from [5] in Python. For both algorithms, we compare three methods of selecting the optimal data source: classic, our proposal, and random baseline. The classic algorithm chooses the optimal data source according to [5]:

$$D_{*j,l} = \arg\max_{\forall D_i \in \mathcal{L}} \frac{N_i^j - O_{i,l}^j}{N_i \cdot C_i}$$

whereas our proposed solution is:

$$D_{*j,l} = \arg\max_{\forall D_i \in \mathcal{L}} \frac{N_i^j \cdot (1 - \mathbb{P}_i)}{N_i \cdot C_i \cdot (1 + \mathbb{P}_i)}$$

Experiment	Number of data sources	Number of groups	Sampling cost	Algorithm
1	2	2	equi-cost	binary DT
2	2	2	skewed	general DT
3	5	2	equi-cost	binary DT
4	5	2	skewed	general DT
5	2	4	equi-cost	general DT
6	2	4	skewed	general DT
7	5	4	equi-cost	general DT
8	5	4	skewed	general DT

Table 1. Configuration of all experiments conducted.

and the random baseline chooses a data source uniformly randomly.

We generate synthetic data with full specification of the overlap distribution, and utilize Python's built-in $O(1)$ set containment search to detect duplicates. The full specifications are as follows:

- **Overlap.** When there are two data sources, $|D_1| = |D_2| = 70$ and $|D_1 \cap D_2| = 30$. When there are five data sources, $|D_1| = \ldots = |D_5| = 40$, pairwise overlap $|D_1 \cap D_2| = |D_1 \cap D_3| = \ldots = |D_4 \cap D_5| = 15$ and any higher degrees of overlap does not exist.
- **Sampling Cost.** In equi-cost scenario, $C_1 = \ldots = C_n = 1$. In skewed-cost scenario with 2 data sources, $C_1 = 2, C_1 = 1$. In skewed-cost scenario with 5 data sources, $(C_1, C_2, C_3, C_4, C_5) = (1, 1.25, 1.5, 1.75, 2)$.
- **Group Distribution.** Group distributions are never uniform and semi-random.
- **Query Count.** The number of tuples required from each group were 5, 10, 20, 30, 40, and 50 such that $Q_1 = (5, 5), Q_2 = (10, 10), \ldots, Q_6 = (50, 50)$ if there are two groups.

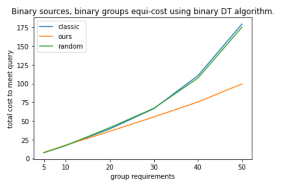
A total of 8 configurations were tested on all three algorithms as shown in Table 1. For each experiment, we tested all three data source selection methods and all six group query count requirements 30 times. The average total cost was computed and plotted in Figure 2. An exception was made to experiment 5 and 6, where a query count of 50 takes too much time and thus was excluded.

4.2 Results

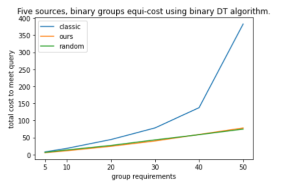
We compare our algorithm with the original binary and general DT algorithm[5] as well as random sampling algorithm for all eight experiments in Figure 2.

First, throughout all the results shown in the Figure 2, we could see it is robust that our algorithm is on par with, or better, than both the classic algorithm and random sampling. The performance of our algorithm remains consistent for all cases, while both random and classic algorithm perform worst in at least some scenarios as seen in the plots. This is to be expected from theoretical analysis since our algorithm strictly considered how overlapping information can be applied for optimization, and hence help make better choice at each iteration. In particular, in Figure 2a, we could see that our algorithm outperforms both the random baseline and the classic algorithm for binary sources.

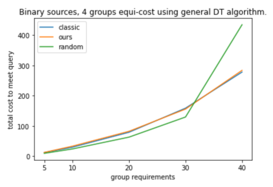
Moreover, we notice that our algorithm have similar performance for equi-cost and skewed-cost case as long as the number of data sources and groups stay the same. Since the cost of selecting from each data source is used as a parameter in determining optimal data source for each group, theoretically whether the the costs are equal have no effect to this process and thus to the result.



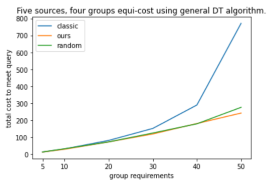
(a)



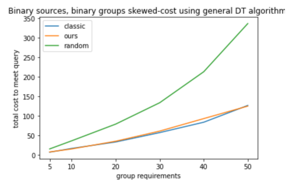
(b)



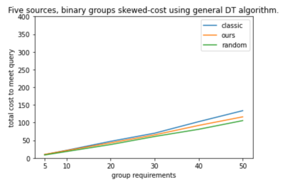
(c)



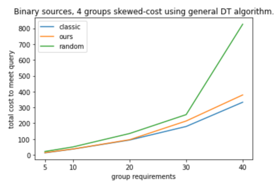
(d)



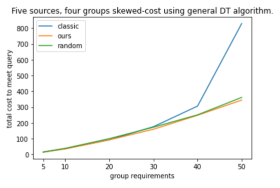
(e)



(f)



(g)



(h)

Fig. 2. Total sampling cost for varying group count requirements. Number of data sources, number of groups, and whether the sampling cost is equal or skewed are varied.

Furthermore, the performance of the random algorithm is interesting. It seems that there is relationship between the performance of the random algorithm and the number of sources and groups. Figure 2f exemplifies on such scenario in which random selection outperformed both alternatives. Since we have not proved optimality of our proposed algorithm for multi-group and skewed-cost generalization, it is possible that our proposed algorithm and the classic algorithm are over-sampling from data sources with high chance of overlap. In these cases, the optimal sampling strategy would be much closer to uniform sampling.

5 EXTENSIONS

5.1 Methods of Obtaining Overlapping Information

This paper assumes that reliable overlap information is readily available. When datasets are sourced in the wild, this assumption rarely holds. As such, methods of efficiently mining overlap statistics, such as [6], may be combined into a single pipeline in a similar manner to [1].

5.2 Known Pairwise Overlapping

In real world, obtaining overall overlapping information among data sources can sometimes be costly, which makes $\mathbb{P}(O \mid D_i)$ hard to compute. However, one can come up with an approximation of this ratio using only pairwise overlapping between data sources. For instance, in 3-data-source case, if we only consider pair-wise overlapping and modify the terms by abandoning $|D_1 \cap D_2 \cap D_3|$ in equation for $\mathbb{P}(O \mid D_1)$, for example, change $\frac{|D_1 \cap D_2| - |D_1 \cap D_2 \cap D_3|}{|D_2|}$ to $\frac{|D_1 \cap D_2|}{|D_2|}$, $\frac{|D_1 \cap D_2 \cap D_3|}{|D_2 \cap D_3|}$ to $\frac{|D_2 \cap D_3|}{|D_2 \cap D_3|} = 1$, we'll obtain an upper bound of $\mathbb{P}(O \mid D_1)$. We may achieve an even lower bound. Through this way, we could still obtain a modified cost formula but using only pairwise information.

5.3 Data Source Selection Optimization

As selecting the data source for the minority group is only proved to be optimal for binary group with equal cost [5], we may come up with a more optimal data source selecting method for the general case that performs better for multiple groups and/or skewed costs. The limited performance of our proposed algorithm displayed in our experiments is the major motivation for further generalization.

6 CONCLUSION

To solve the problem of ensuring data sets used for experiments have proper representation of demographic groups, effective ways of tailoring data sources have been proposed. In this paper, we study how we can improve the general DT algorithm[5] for collecting data from known distribution data sources when there is overlap between them. Specifically, given a tuple from a data source, we define the probability of this tuple have been already collected in the target data set. As there are overlaps between each data source, we use the ratio of overlaps combined with the collected set to estimate this probability, which is applied in the expected cost formula for selecting data source in each iteration. We have conducted experiments to show the effectiveness of our improvement on the original algorithm. We can see that our proposed algorithm has consistent good performances in different scenarios, and outperforms two baseline algorithms significantly in some cases.

REFERENCES

[1] Mirko Bronzi, Valter Crescenzi, Paolo Merialdo, and Paolo Papotti. 2013. Extraction and integration of partially overlapping web sources. *Proceedings of the VLDB Endowment* 6, 10 (2013), 805–816.

[2] AnHai Doan, Alon Halevy, and Zachary Ives. 2012. *Principles of data integration.* Elsevier.

[3] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI conference on human factors in computing systems.* 1–16.

[4] Sara Hooker. 2021. Moving beyond "algorithmic bias is a data problem". *Patterns* 2, 4 (2021), 100241.

[5] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2021. Tailoring data source distributions for fairness-aware data integration. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2519–2532.

[6] Zaiqing Nie, S. Kambhampati, and U. Nambiar. 2005. Effectively mining and using coverage and overlap statistics for data integration. *IEEE Transactions on Knowledge and Data Engineering* 17, 5 (2005), 638–651. https://doi.org/10.1109/TKDE.2005.76

[7] Draco Xu, Jiwon Chang, Yurong Liu, and Zhigong Gao. 2022. overlapping-dt. https://github.com/jiwonac/overlapping-dt.