COUNTATA: Dataset Labeling Using Pattern Counts*

Yuval Moskovitch
University of Michigan
yuvalm@umich.edu

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

ABSTRACT

Information regarding the counts of attributes combination is central to the profiling of a data set. It may reveal bias; it can help determine fitness for use. While counts of individual attribute values may be stored in some data set profiles, there are too many combinations of attributes for it to be practical to store counts for each combination. To this end, we present the notion of storing a "label" of limited size that can be used to obtain good estimates for these counts. A label contains information regarding the count of selected patterns–attributes values combinations–in the data. We define an estimation function, that uses this label to estimate the count of every pattern. Intuitively, there is a trade-off between the label size and its estimation error. We propose a demonstration of COUNTATA, a system that allows the user to examine this trade-off as well as the label's count information. We will demonstrate the usefulness of COUNTATA using real-life data, and illustrate the effectiveness of our estimation paradigm.

PVLDB Reference Format:
Yuval Moskovitch and H. V. Jagadish. COUNTATA: Dataset Labeling Using Pattern Counts. *PVLDB*, 13(12): 2829 - 2832, 2020.
DOI: https://doi.org/10.14778/3415478.3415486

1. INTRODUCTION

"Found data" is data that was not collected as part of the development of data-driven algorithms and tools, but was rather acquired independently, possibly assembled by others for different purposes. Usage of existing data is very common this days as a result of the emerging variety of publicly available datasets, and their online accessibility. While convenient, this may lead to incompatibility of the data to the user's desired task, which in turn can result in discriminating or unfair decisions, algorithmic racism and biased models [8].

Data-driven methods are increasingly being used in domains such as fraud and risk detection, where data-driven algorithmic decision making may affect human life. For instance, risk assessments tools, which predict the likelihood of a defendant to re-offend, are widely used in courtrooms across the US [4]. ProPublica, an independent, non-profit newsroom that produces investigative journalism in the public interest, conducted a study on the risk assessment scores output by a software developed by Northpointe, Inc. They found that the software discriminated based on race: blacks were scored at greater risk of re-offending than the actual, while whites were scores at lower risk than actual.

Information regarding the attributes' values such as their type, distribution statistics, common patterns, and attributes correlations and dependencies may assist in mitigating misuse of data and reduce algorithmic bias and racism. This flavor of information can be extracted in the process of data profiling, a standard step preformed by analysts when using "found data". While informative and useful, data profiling is hard to do well, is usually not automated, and requires significant effort. To help both the data analyst and the data user, the notion of a "nutrition label" has been suggested [11, 9, 7, 12, 10, 13]. The basic idea of a nutrition label is to capture, in a succinct label, data set properties of interest. Perhaps the single most important such property is a profile of the counts of various attribute value combinations. For instance, an analyst may wish to ensure a (close) to real world distribution in the attribute's values of the data, such as equal number of male and female. Another concern may be the lack of adequate representation in the data for a particular group [5], such as divorced African-American female, or contrarily, a high percentage of data that represents the same group (data skew) [6].

To this end we propose to label datasets with information regarding the count of different patterns (attributes values combinations) in the data, which can be useful to determine fitness for use. Needless to say, there is a combinatorial number of such combinations possible. So, storing individual counts for each is likely to be impossible. Thus, we focus on techniques to estimate these counts while storing only a limited amount of information.

EXAMPLE 1.1. *COMPAS is the risk assessment commercial tool made by Northpointe, Inc. The COMPAS dataset was collected and published by ProPublica as part of their investigation [1]. The full dataset contains 60,843 tuples with 29 attributes. Figure 2 depicts a label with partial counting information of a simplified version, including only six*

*This research has been supported in part by NSF grants 1741022 and 1934565.

Proceedings of the VLDB Endowment, Vol. 13, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3415478.3415486

attributes: gender, age group, race, marital status, legal status and supervision level. This dataset description depicts the possible values of each attributes, and their count in the data (Figure 2b), with the addition of some attributes values combination count, legal status and supervision level in this example (Figure 2a). Some immediate observations that can be made based on this information is that female and male are not equally represented in the data, and due to the low number of widows in the data, there is a high possibility that the number of Hispanic female widows is inadequate for the development of non-biased algorithm using this data.

If we know the marginal distributions (or individual attribute value counts), we can make an independence assumption and estimate the joint distribution (multi-attribute intersection counts); but if we are additionally given selected intersection counts, how should we use these to estimate other intersection counts not provided? We present a model for this estimation in Section 2. Given the estimation procedure, each label entails an error with respect to the real count of patterns in the data. Intuitively, a label is the count of value combinations of a selected subset of attributes. The problem is then to choose a label that minimizes the error, where the number of value combinations is limited by a given space budget. We show in [3] that this problem is NP-hard and present an efficient heuristic. We propose to demonstrate our solution, which we have implemented in a system called COUNTATA (for "COUNTing labels of dATAset"). The system allows the user to examine datasets' labels, counting information of selected patterns, and detection of skews and underrepresented patterns. The framework is designed to assist the data-owners to determine the desired bound over the generated data label. We will demonstrate COUNTATA using real world datasets, let the audience interactively explore the datasets, and show the trade-off between the label size and its accuracy.

Related Work. With the increasing interest in data equity in recent years, multiple lines of work have focused on labeling data and models in order to improve transparency, accountability and fairness in data science. Different data labeling models were studied in [9, 7, 12]. Other works focused on model labeling [10, 13]. Our proposed label model may be assimilated as a widget or a module in the above models. While the idea of a nutritional label has been very nicely argued for in these works, the actual content of the label is either manually generated, or at most has an aspiration towards automated generation beyond the simplest properties. Our work establishes the first critical widget that provides substantive information about a data set and is constructed in a completely automated manner.

2. TECHNICAL BACKGROUND

We (informally) introduce the model underlying COUNTATA, using examples. See [3] for a full description of the theory and notation. We assume the data is represented using a single relational database, and that the relation's attributes values are categorical. Attribute with continuous values domain may be converted to categorical domain by bucketizing them into ranges, as commonly done in practice to present aggregate results.

	Gender	Age group	Race	Marital status
1	Female	under 20	African-American	single
2	Male	20-39	African-American	divorced
3	Male	under 20	Hispanic	single
4	Male	20-39	Caucasian	married
5	Female	20-39	African-American	divorced
6	Male	20-39	Caucasian	divorced
7	Female	20-39	African-American	married
8	Male	under 20	African-American	single
9	Female	20-39	Caucasian	divorced
10	Male	under 20	Caucasian	single
11	Male	20-39	Hispanic	divorced
12	Female	under 20	Hispanic	single
13	Female	20-39	Hispanic	married
14	Female	under 20	Caucasian	single
15	Female	20-39	Caucasian	married
16	Male	20-39	Hispanic	married
17	Male	20-39	African-American	married
18	Female	20-39	Hispanic	divorced

Figure 1: Sample data from a simplified version of the COMPAS dataset

2.1 Patterns count information

Given a database D with attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$, we use $Dom(A_i)$ to denote the active domain of A_i for $i \in [1..n]$. A *pattern* p is a set $\{A_{i_1} = a_1, \ldots, A_{i_k} = a_k\}$ where $\{A_{i_1}, \ldots, A_{i_k}\} \subseteq \mathcal{A}$ and $a_j \in Dom(A_{i_j})$ for each A_{i_j} in p. We use $Attr(p)$ to denote the set of attributes in p.

EXAMPLE 2.1. *Figure 1 depicts a fragment of a simplified version of the COMPAS database containing only the attributes gender, age group, race and marital status.* $p =\{$*age group= under 20, marital status = singe*$\}$ *is a possible pattern and* $Attr(p) =\{$*age group, marital status*$\}$.

We say that a tuple $t \in D$ *satisfies* the pattern p if $t.A_i = a_i$ for each $A_i \in Attr(p)$. The count $c_D(p)$ of a pattern p is then the number of tuples in D that satisfy p.

EXAMPLE 2.2. *Consider again the database given in Figure 1. The tuples 1, 3, 8, 10, 12, and 14 satisfy the pattern* $p =\{$*age group= under 20, marital status = single*$\}$ *and thus the count of* p *is* $c_D(p) = 6$.

While full count of each pattern provides detailed and accurate description of the data, it can be extremely large. In fact it can have the same size as the data.

EXAMPLE 2.3. *As a simple example, consider a database* D *with* n *binary attributes* $A_1, \ldots, A_n$, *where each value combination* $(b_1, \ldots, b_n)$, *for* $b_i \in \{0, 1\}$, *appears exactly once. In this case the database, as well as the patterns count, includes* 2^n *tuples.*

One way we could control the size of stored information is to keep counts only for individual attribute values, and estimate counts for attribute value combinations, assuming independence.

EXAMPLE 2.4. *Continuing with Example 2.3, given the counts* $c_D(\{A_i = b_i\}) = \frac{2^n}{2}$, *the count of the pattern* $\{A_1 = 0, A_2 = 0, A_3 = 0\}$ *may be estimated as*

$$2^n \cdot \prod_{i=1}^{3} \frac{c_D(\{A_i = 0\})}{c_D(\{A_i = 0\}) + c_D(\{A_i = 1\})} = 2^n \cdot \left(\frac{1}{2}\right)^3 = 2^{n-3}$$

Intuitively, under the assumption that there are no correlations, the count of the pattern $\{A_1 = 0, A_2 = 0, A_3 = 0\}$ *is* *the relative portion of the data (total number of 2^n tuples), that have the value 0 in the attribute A_1, A_2 and A_3, which is reflected in the sub-expressions $\frac{c_D(\{A_i=0\})}{c_D(\{A_i=0\})+c_D(\{A_i=1\})}$ in the computation. In general, the count of the pattern $p = \{A_{i_1} = b_{i_1}, \ldots, A_{i_k} = b_{i_k}\}$ can be computed as*

$$|D| \cdot \prod_{j=1}^{k} \frac{c_D(\{A_{i_j} = b_{i_j}\})}{c_D(\{A_{i_j} = 0\}) + c_D(\{A_{i_j} = 1\})}$$

However, when we introduce correlations, the counts of individual attributes are no longer sufficient to provide a good estimation, as we next demonstrate.

Example 2.5. *As a simple example, consider a database D with n binary attributes as described in Example 2.3, except that the values in the attributes A_1 are replaced such that the value of A_1 is equal to the value of A_2 for every tuple. The real count of the pattern $\{A_1 = 0, A_2 = 0, A_3 = 0\}$ is now 2^{n-2}, where using only the individual count the pattern count estimation is 2^{n-3} with the same computation shown in Example 2.4.*

We may remedy this problem by using additional count information. In the above example, the counts of the patterns $p = \{A_1 = b_1, A_2 = b_2\}$ for $b_i \in \{0, 1\}$ is sufficient to provide an exact estimate for each pattern in the database.

Example 2.6. *Given the patterns count $c_D(\{A_1 = 0, A_2 = 0\}) = 2^{n-1}$ we can compute the count of $\{A_1 = 0, A_2 = 0, A_3 = 0\}$ as $2^{n-1} \cdot \frac{c_D(\{A_3=0\})}{c_D(\{A_3=0\})+c_D(\{A_3=1\})} = 2^{n-1} \cdot \frac{1}{2} = 2^{n-2}$.*

In general, the count of any pattern $p = \{A_{i_1} = b_{i_1}, \ldots, A_{i_k} = b_{i_k}\}$ (that contains $\{A_1 = b_1, A_2 = b_2\}$ for $b_i \in \{0, 1\}$) can be computed as

$$c_D(\{A_1 = b_1, A_2 = b_2\}) \cdot \prod_{j=3}^{k} \frac{c_D(\{A_{i_j} = b_{i_j}\})}{c_D(\{A_{i_j} = 0\}) + c_D(\{A_{i_j} = 1\})}$$

Real world datasets are typically complex, and have correlations among attributes. One possible way to tackle this problem is to store more information about these (large) deviations from our initial independence assumption. The challenge is to spend wisely a limited space budget to capture exactly the deviations that induce greatest error in the estimates, as we next explain.

2.2 Patterns count based labels

We next present our notion of data label. A label $L_S(D)$ of D is defined with respect to a subset S of the database attributes and contains: (1) the pattern count (PC) for each possible pattern over S (i.e., p with $Attr(p) = S$), and (2) value count (VC) of each value appearing in D.

Example 2.7. *Consider the database fragment given in Figure 1, the label resulting from use of the attributes set S = {age group, marital status} consists of the following:*

$$\begin{aligned}
PC = \{&(\{\textit{age group} = \textit{under 20, marital status} = \textit{single}\}, 6)\\
&(\{\textit{age group} = \textit{20-39, marital status} = \textit{married}\}, 6),\\
&(\{\textit{age group} = \textit{20-39, marital status} = \textit{divorced}\}, 6)\}\\
VC = \{&(\{\textit{gender} = \textit{female}\}, 9), (\{\textit{gender} = \textit{male}\}, 9),\\
&(\{\textit{age group} = \textit{under 20}\}, 6),\\
&(\{\textit{age group} = \textit{20-39}\}, 12),\\
&(\{\textit{race} = \textit{African-American}\}, 6),\\
&(\{\textit{race} = \textit{Hispanic}\}, 6), (\{\textit{race} = \textit{Caucasian}\}, 6),\\
&(\{\textit{marital status} = \textit{single}\}, 6),\\
&(\{\textit{marital status} = \textit{divorced}\}, 6),\\
&(\{\textit{marital status} = \textit{married}\}, 6)\}
\end{aligned}$$

The label resulting from use of the attributes set S' = {gender, age group} consists of the same VC set and the following PC set:

$$\begin{aligned}
PC = \{&(\{\textit{gender} = \textit{female, age group} = \textit{under 20}\}, 3)\\
&(\{\textit{gender} = \textit{male, age group} = \textit{under 20}\}, 3),\\
&(\{\textit{gender} = \textit{female, age group} = \textit{20-39}\}, 6),\\
&(\{\textit{gender} = \textit{male, age group} = \textit{20-39}\}, 6)\}
\end{aligned}$$

Note that for a given database D, the VC set is similar in every label of D. We next explain how the data labels can be used to estimate the count of every pattern in the database.

Given a databse D with attributes $\mathcal{A}$ and a subset of attributes $S \subseteq \mathcal{A}$ we use P_S to denote the set of all possible patterns over S such that $c_D(p) > 0$. Let S_1 and S_2 be two subsets of attributes such that $S_1 \subseteq S_2 \subseteq \mathcal{A}$. Given a pattern $p \in P_{S_2}$, we use $p|_{S_1}$ to denote the pattern that results when p is restricted to include only the attributes of S_1. Given a label $l = L_{S_1}(D)$ of D using S_1, we may estimate the count of each pattern in P_{S_2} as follows.

$$Est(p, l) = c_D(p|_{S_1}) \cdot \prod_{A_i \in S_2 \setminus S_1} \frac{c_D(\{A_i = p.A_i\})}{\sum_{a_j \in Dom(A_i)} c_D(\{A_i = a_j\})}$$

Example 2.8. *Consider again the database given in Figure 1, and the label $l = L_S(D)$ generated using S ={age group, marital status} shown in Example 2.7. The estimate of the pattern p ={gender = female, age group = 20-39, marital status = married} using l is*

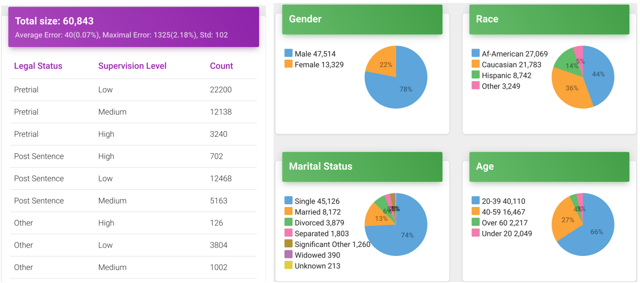
$$\begin{aligned}
&Est(p, l) =\\
&\quad c_D(\textit{age group} = \textit{20-39, marital status} = \textit{married}) \cdot\\
&\quad \frac{c_D(\{\textit{gender} = \textit{female}\})}{\sum_{a_j \in Dom(\textit{gender})} c_D(\{\textit{gender} = a_j\})} = 6 \cdot \frac{9}{18} = 3
\end{aligned}$$

Using the label $l' = L_{S'}(D)$ generated from S' = {gender, age group}, with a similar computation we obtain

$$\begin{aligned}
&Est(p, l') = c_D(\textit{gender} = \textit{female, age group} = \textit{20-39}) \cdot\\
&\quad \frac{c_D(\{\textit{marital status} = \textit{married}\})}{\sum_{a_j \in Dom(\textit{marital status})} c_D(\{\textit{marital status} = a_j\})} =\\
&\qquad 6 \cdot \frac{6}{18} = 2
\end{aligned}$$

We can then define the error of a label with respect to a pattern and a set of patterns.

$$Err(l, p) = |c_D(p) - Est(p, l)|$$



Total size: 60,843

Average Error: 40(0.07%), Maximal Error: 1325(2.18%), Std: 102

Legal Status	Supervision Level	Count
Pretrial	Low	22200
Pretrial	Medium	12138
Pretrial	High	3240
Post Sentence	High	702
Post Sentence	Low	12468
Post Sentence	Medium	5163
Other	High	126
Other	Low	3804
Other	Medium	1002

(a) Pattern Counts **(b)** Value Counts (partial)

Figure 2: Dataset Label

EXAMPLE 2.9. *Reconsider the estimates $Est(p,l)$ and $Est(p,l')$ of the pattern p ={gender = female, age group = 20-39, marital status = married} shown in Example 2.8. The count of the pattern p in the database is* 3*, thus the error of l with respect to p is* 0 *and the error of l' is* 1.

Abusing notation, we use $Err(l,\mathcal{P})$, for a set of patterns $\mathcal{P}$, to denote the maximum error in the estimate for any individual pattern in $\mathcal{P}$. We choose to focus on the maximum error (rather than mean for instance), as this definition of error is stiffer and gives us a sense of the error "bound" over a large number of patterns in the database. In [3] we show that the problem of finding the optimal label (i.e., the label with minimal error) with a bounded size in NP-hard, and present a heuristic for it. Due to space limitation we omit the discussions of size and estimation accuracy trade-off and the algorithm's performance, see [3] for full details.

3. SYSTEM OVERVIEW

COUNTATA's back-end side is implemented in Python 3 and runs on macOS Catalina. The user interacts with the system using a dedicated user interface (shown in Figures 2), implemented in React with Material-UI framework. The dataset's label view includes the set of data attributes, and their values distribution (i.e., the set VC described in Section 2.2), as shown in Figure 2b. For each attribute COUNTATA presents the count of every possible value, and a visual display using a pie chart. The presentation may be manually refined and attributes can be filtered-out in order to adjust the information to the user's interest. Figure 2a depicts the PC set of the label (the counts of the patterns in P_S for S ={legal status, supervision level} in the example presented in the figure). The system also provides information regarding the label's maximal error, mean error and the standard deviation (on the table's header).

The user can define a pattern by specifying (some of the) attribute's values. Once the pattern is fed to the system, COUNTATA presents the user with estimated count of the requested pattern and the range of values estimation with the average error. The user may also specify a threshold for skew or underrepresented patterns detection. Given a threshold T, COUNTATA present the set of patterns with count above/below T, where each pattern is associated with it's count estimation and the average error range.

4. DEMONSTRATION SCENARIO

We will demonstrate the usefulness of COUNTATA in assessing the count of pattern using real world datasets. In particular we will use:

- The Blue Nile dataset collected and used in [5] of diamonds catalog of the online jewelry retailer Blue Nile.
- The COMPAS dataset that was collected and published by ProPublica [1]
- The Default of Credit Card Clients Dataset [2], which contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

The audience will be asked to play the role of data scientist, examining the benefits of COUNTATA in estimating the patterns count, detecting underrepresented groups and skews.

The users will first select a dataset and load the label (generated using the algorithm in [3]) to COUNTATA. We will then browse through the label and ask the participants to insert a pattern whose count they wish to estimate. For demonstration purposes, we will present the real count of the selected pattern along with the system's estimation count. We will then ask the users to set thresholds for the skew and underrepresented pattern detection mood. We will observe the results and compare the estimated counts to the real patterns counts.

Finally, we will let the audience "look under the hood". In particular, we will show the trade-off between label size and accuracy by considering labels with varying sizes for the selected dataset, highlighting the difference in the number of tuples presented to the user in the label information and the error in the patterns count.

5. REFERENCES

[1] Compas recidivism risk score data and analysis. `https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis`.

[2] Default of credit card clients data set. `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`.

[3] Patterns count-based labels for datasets [technical report]. `https://web.eecs.umich.edu/~yuvalm/docs/labelsFull.pdf`.

[4] J. Angwin, J. Larson, L. Kirchner, and S. Mattu. Machine bias, May 2016.

[5] A. Asudeh, Z. Jin, and H. V. Jagadish. Assessing and remedying coverage for a given dataset. In *ICDE*, 2019.

[6] I. Y. Chen, F. D. Johansson, and D. A. Sontag. Why is my classifier discriminatory? In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, 2018.

[7] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. M. Wallach, H. D. III, and K. Crawford. Datasheets for datasets. *CoRR*, abs/1803.09010, 2018.

[8] J. Gu and D. Oelke. Understanding bias in machine learning. *CoRR*, abs/1909.01866, 2019.

[9] S. Holland, A. Hosny, S. Newman, J. Joseph, and K. Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards. *CoRR*, abs/1805.03677, 2018.

[10] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. In *FAT**. ACM, 2019.

[11] J. Stoyanovich and B. Howe. Nutritional labels for data and models. *IEEE Data Eng. Bull.*, 42(3):13–23, 2019.

[12] C. Sun, A. Asudeh, H. V. Jagadish, B. Howe, and J. Stoyanovich. Mithralabel: Flexible dataset nutritional labels for responsible data science. In *CIKM*. ACM, 2019.

[13] K. Yang, J. Stoyanovich, A. Asudeh, B. Howe, H. V. Jagadish, and G. Miklau. A nutritional label for rankings. In *SIGMOD*. ACM, 2018.