Semi-Automated Estimation of Weighted Rates for E-commerce Catalog Quality Monitoring

Mauricio Sadinle, Karim Bouyarmane, Grant Galloway, Shioulin Sam,
Changhe Yuan, and Ismail Tutar

Amazon.com, Inc.
{sadinlem,bouykari,gggallow,shioulin,ychanghe,ismailt}@amazon.com

Abstract. Product catalogs represent the backbone of e-commerce websites. Given these catalogs' constant evolution, we need to closely monitor the quality of their product information. Identifying defective product information, however, often requires human auditing, which makes catalog monitoring expensive. In this article, we investigate approaches for tracking weighted rates over time, here defined as the fraction of customer attention that goes to items with a particular defect. We focus on these metrics, given that to improve customer trust we need to minimize their exposure to listings with defective information. We assume that the gold standard for detecting defects comes from human auditors, but to avoid collecting audits at each point in time, we leverage existing machine learning classifiers. However, simply replacing human auditor decisions with automated predictions generally leads to large biases in the estimated weighted rates. We instead leverage classifiers while obtaining approximately unbiased and low variance estimators of the weighted rate of interest. We rely on being able to evaluate the quality of the classifier using audits at a baseline time, and then extrapolate its performance to the target times. We perform extensive simulation studies to stress-test our proposed estimation approaches under a variety of scenarios representative of our use cases. Our proposed estimation approach is related to the task of *quantification* in machine learning, and so we draw connections throughout the document.

Keywords: Quantification · Prior probability shift · Label shift.

1 Introduction

Product catalogs are the backbone of e-commerce websites, as they provide the information that is presented to customers. Maintaining customer trust requires identifying defects in product information, which usually needs human inspection for detection. For instance, product information on Amazon.com is consolidated from contributions by individual sellers [1]. These consolidated product attributes frequently contain defects, such as inconsistencies or erroneous values due to honest mistakes by sellers, system errors, and bad actors who intentionally introduce corrupted information. This causes detrimental performance of a variety of customer-facing applications; for instance, displaying such imperfect information to customers erodes their trust.

Given the scale of e-commerce product catalogs, it is nearly impossible to manually inspect all of their information. An important task in ensuring high catalog quality involves monitoring quality metrics. This monitoring is often done through careful human inspection of random samples of product entries collected periodically. Even with carefully designed samples, when business goals require tight control of these metrics at high frequency, monitoring through human auditing becomes extremely expensive. This creates the need for automated procedures that allow to monitor quality metrics while maintaining strong guarantees on their accuracy.

In this article, we investigate a methodology for estimating weighted rates in a semi-automated way. The reason for using weighted rates in our use cases is that not all products are equally important for customers. Given a signal of customer engagement with each product, we are interested in monitoring the fraction of such signal among products with a defect. For instance, we might want to track the fraction of customer visits to product pages that contain erroneous information; from a customer-centric point of view, this is a more important metric to monitor and aim to reduce than the simple fraction of catalog products with erroneous information.

For our use cases there typically exist machine learning classifiers in production for detecting defects. These classifiers tend to be complex and are trained on audited data collected over time. Given that retraining such classifiers for the sake of metric measurement is burdensome and not cost-efficient, we propose to use them in their existing form to predict defects. However, it is well known that simply replacing human auditor decisions with classifier predictions generally leads to large biases in the estimated metrics [9, 12, 6, 7]. To leverage existing classifiers while obtaining approximately unbiased and low variance estimators, we rely on being able to evaluate the quality of the classifier using audits at a baseline time. We then assume that the performance of the classifier in terms of its true and false positive rates can be extrapolated to the target time. Our methodology constitutes an extension of techniques proposed for the machine learning task of *quantification*, reviewed next.

1.1 Quantification

Forman [7] introduced the *quantification* task to address the following problem: how can we use labeled training data from a baseline population to estimate the proportion of a class in a target population where we only have unlabeled data. This task is related to the fundamental problem of estimating a proportion using an imperfect diagnostic tool, studied earlier in epidemiology [9, 12], in the context of mechanical sorting devices [11], among others [10]. A seemingly obvious solution to the quantification task is to train a classifier on the labeled data, use it to predict the class for the unlabeled data, and then simply summarize the proportion of class predictions. This approach, known as *classify and count* [6, 7], is known to perform poorly, as it is generally guaranteed to be biased, except for a few restrictive conditions [8, 10]. Forman [6, 7] recognized this, and proposed alternative approaches for estimation, including an *adjusted classify and count* (ACC) approach that is guaranteed to work well under certain conditions; we will refer back to this method later in this article. Interestingly, the ACC approach had also been derived earlier by other authors [9, 12, 11], which shows the ubiquity of the quantification problem.

Forman [7, 8] also introduced *cost-quantification*, as the task of estimating total costs for each class using class predictions by imperfect classifiers. This task seems to have received less attention in the literature; for instance, a 2017 review [10] only included the proposed solutions by Forman [7] in 2006, and to the best of our knowledge no further advances have been proposed for cost-quantification since then, despite many advances for the simpler quantification task [13–15]. The automated estimation of weighted rates, as in this paper, is closely related to cost-quantification, given that if we can estimate the total of a class, we can also estimate the fraction it represents with respect to the total cost across classes. The methods that we propose in this article therefore also contribute to cost-quantification solutions. Our contribution consists in showing that an analog of the ACC approach is valid for estimating weighted rates under two assumptions that allow to extrapolate and simplify the true and false positive rates of the classifier. We also investigate approaches for dealing with the classifiers' thresholds that lead to weighted rate estimators with low variance, and compare approaches for constructing confidence intervals.

2 Methodology

We shall think of a product catalog at a time t as a collection of features on N_t products. A product i has a *known* non-negative measure of importance for customers, or weight, at time t denoted W_{it}. Let Y_{it} denote the defect indicator for product i at time t, 1 if defective, 0 otherwise. This indicator Y_{it} is unknown and determining its true value requires human auditing.

Estimation target: Formally, our goal is to estimate the *weighted rate* R_t at a time t:

$$R_t = \sum_{i=1}^{N_t} W_{it} Y_{it} \Big/ \sum_{i=1}^{N_t} W_{it}. \tag{1}$$

We assume that we do not have audited data from the catalog at time t, and so we rely on the existence of a classifier to predict the status Y_{it} of a product i. Let $h(\cdot)$ denote a generic classifier that takes in a feature vector X_{it} of product i at time t, and outputs a predicted status $\hat{Y}_{it}$ for product i, that is, $\hat{Y}_{it} = h(X_{it}) \in \{0, 1\}$. The classifier $h(\cdot)$, for instance, can be obtained from thresholding a score at a cutpoint c, say $h(X_{it}) = I[g(X_{it}) > c]$, where $I(\cdot)$ is the indicator function and $g(\cdot)$ may represent a score obtained from a model or from some complicated procedure. For now we assume that $h(\cdot)$ is fixed, but later we compare approaches to handle classification thresholds.

We also rely on having an audited sample at a baseline time, which we use to estimate the true and false positive rates of the classifier at that baseline time. In practice, we implement *measurement cycles* that start with collection of audits to evaluate the performance of the classifier, and then use that information to produce automated estimates for the remainder of the cycle; see Appendix A at `https://bit.ly/3wJK5Mj` for a more detailed description.

2.1 The Proposed Weighted Rate Estimator

To derive the proposed estimator of the weighted rate, we first do a slight rewriting of the estimation target. To this end, let $(W, Y, \hat{Y})$ be a random vector that takes with probability $1/N_t$ each of the catalog values at time t, $\{(W_{it}, Y_{it}, \hat{Y}_{it})\}_{i=1}^{N_t}$. With this formulation, our estimation target can be equivalently written as

$$R_t = \frac{\sum_{i=1}^{N_t} W_{it} Y_{it}}{\sum_{i=1}^{N_t} W_{it}} = \frac{(1/N_t)\sum_{i=1}^{N_t} W_{it} Y_{it}}{(1/N_t)\sum_{i=1}^{N_t} W_{it}} = \frac{E_t(WY)}{E_t(W)},$$

where $E_t(\cdot)$ denotes the expected value using the values of the catalog at time t.

The quantity that we would obtain from simply using the predictions $\hat{Y}_{it}$ instead of the true values Y_{it} is here denoted as R_t^{raw}, and it is given by

$$R_t^{raw} = \frac{\sum_{i=1}^{N_t} W_{it} \hat{Y}_{it}}{\sum_{i=1}^{N_t} W_{it}} = \frac{E_t(W\hat{Y})}{E_t(W)},$$

which generally will differ from the target R_t. Our strategy to derive the proposed weighted rate estimator requires connecting R_t and R_t^{raw} through the classification performance of $h(\cdot)$. First, note that we assume the weights W_{it} to be known, and therefore $E_t(W)$ to be known, allowing us to focus on connecting $E_t(WY)$ with $E_t(W\hat{Y})$. Note that, by the law of total expectation, we can write

$$\begin{aligned} E_t(WY) &= E_t[W\ P_t(Y = 1 \mid W)], \qquad (2) \\ E_t(W\hat{Y}) &= E_t[W\ P_t(\hat{Y} = 1 \mid W)]. \end{aligned}$$

Also, by the law of total probability,

$$P_t(\hat{Y} = 1 \mid W) = p_{1|1,t}(W)\ P_t(Y = 1 \mid W) + p_{1|0,t}(W)\ [1 - P_t(Y = 1 \mid W)], \qquad (3)$$

where $p_{1|a,t}(W) = P_t(\hat{Y} = 1 \mid Y = a, W)$ denotes the true positive rate (TPR) for $a = 1$, and the false positive rate (FPR) for $a = 0$, as a function of the weights at time t. From equation (3), we can establish the relationship

$$P_t(Y = 1 \mid W) = \frac{P_t(\hat{Y} = 1 \mid W) - p_{1|0,t}(W)}{p_{1|1,t}(W) - p_{1|0,t}(W)}, \qquad (4)$$

which resembles the basis for the ACC estimator of simple proportions [6], although here it appears conditional on a value W of the weights. Replacing equation (4) into (2) above, we obtain the identity

$$E_t(WY) = E_t\left[W\ \frac{P_t(\hat{Y} = 1 \mid W) - p_{1|0,t}(W)}{p_{1|1,t}(W) - p_{1|0,t}(W)}\right]. \qquad (5)$$

Creating an estimator based on this expression is not straightforward. Firstly, estimating the TPR and FPR functions, $p_{1|1,t}(W)$ and $p_{1|0,t}(W)$, for the catalog at time t would require collecting audited data at time t, which defeats the purpose of automating the estimation approach. The validity of our proposed estimator therefore relies on being able to extrapolate the performance of the classifier from a baseline time to the target time t.

Extrapolation assumption (EA): The TPR and FPR at the time of interest t are the same as at the baseline time.

Additionally, although not strictly required, we also work under an extra assumption to favor a simple estimator.

Simplifying assumption (SA): The TPR and FPR are constant as a function of the weights.

We discuss the plausibility of these assumptions in detail in Section 2.2. The EA can be written as $P_0(\hat{Y} = 1 \mid Y = a, W) = P_t(\hat{Y} = 1 \mid Y = a, W)$, for $a = 0, 1$. Under the EA, we can ignore the time subindex and simply write $p_{1|a}(W) = P(\hat{Y} = 1 \mid Y = a, W)$, for $a = 0, 1$. Then, the SA can be written as $p_{1|a}(W) = p_{1|a}(W')$ for any two values of the weights W and W', where $a = 0, 1$. Under the SA we can simplify the notation and write $p_{1|1} = p_{1|1}(W)$ and $p_{1|0} = p_{1|0}(W)$.

Given the EA and SA, expression (5) simplifies as

$$E_t(WY) = E_t\left[W \, \frac{P_t(\hat{Y} = 1 \mid W) - p_{1|0}}{p_{1|1} - p_{1|0}}\right] = \frac{E_t(W\hat{Y}) - p_{1|0}E_t(W)}{p_{1|1} - p_{1|0}},$$

and we obtain

$$R_t = \frac{E_t(WY)}{E_t(W)} = \frac{E_t(W\hat{Y})/E_t(W) - p_{1|0}}{p_{1|1} - p_{1|0}} = \frac{R_t^{raw} - p_{1|0}}{p_{1|1} - p_{1|0}}. \tag{6}$$

Interestingly, this has the same form as the ACC estimator for simple proportions [9, 12, 6, 7], except that here R_t and R_t^{raw} are weighted rates.

Given expression (6), we propose to estimate the weighted rate as

$$\hat{R}_t = \frac{\hat{R}_t^{raw} - \hat{p}_{1|0}}{\hat{p}_{1|1} - \hat{p}_{1|0}}, \tag{7}$$

where $\hat{R}_t^{raw}$ is estimated from a very large random sample from the catalog at time t, or preferably $\hat{R}_t^{raw}$ is taken exactly as R_t^{raw}, if computational resources allow. The estimated TPR and FPR, $\hat{p}_{1|1}$ and $\hat{p}_{1|0}$, are obtained from the audited data from baseline. The appropriate estimators for each of these quantities depend on the sampling scheme [16], but as long as they are consistent, the consistency of $\hat{R}_t$ is guaranteed by the continuous mapping theorem [19] because the true value $R_t = (R_t^{raw} - p_{1|0})/(p_{1|1} - p_{1|0})$ is a continuous function of R_t^{raw}, $p_{1|1}$, and $p_{1|0}$. This argument serves as the proof of the following result.

Theorem 1 (statistical consistency). *Under EA and SA, assume that $\hat{R}_t^{raw}$, $\hat{p}_{1|1}$, and $\hat{p}_{1|0}$ are statistically consistent estimators for R_t^{raw}, the TPR, and the*

FPR, respectively. Then, the proposed estimator $\hat{R}_t$ is statistically consistent for the target rate R_t.

Statistical consistency of our estimator is an important property, as it guarantees that as the sample sizes increase, the estimator converges in probability to the true value that we want to estimate [19]. In particular, it implies that our estimator is approximately unbiased for large sample sizes. Working with statistically consistent estimators $\hat{R}_t^{raw}$, $\hat{p}_{1|1}$, and $\hat{p}_{1|0}$ is relatively standard; for instance, with simple random samples $\mathcal{S}_0$ of size n_0 at baseline, and $\mathcal{S}_t$ of size $n_t \gg n_0$ at time t, the following estimators are consistent:

$$\hat{R}_t^{raw} = \sum_{i \in \mathcal{S}_t} W_{it}\hat{Y}_{it} \Big/ \sum_{i \in \mathcal{S}_t} W_{it}; \;\; \hat{p}_{1|a} = \sum_{i \in \mathcal{S}_0} \hat{Y}_{i0} I(Y_{i0} = a) \Big/ \sum_{i \in \mathcal{S}_0} I(Y_{i0} = a), \; a = 0, 1.$$

More intricate estimators will be needed under more complex sampling schemes, but those details go beyond the scope of this paper. The proposed estimator $\hat{R}_t$ heavily relies on the assumptions EA and SA, which we discuss next.

2.2 Discussion of Assumptions

To examine the plausibility of the assumptions EA and SA, let us expand the TPR and FPR in terms of the classifier $h(\cdot)$ and the product's features X,

$$P_t(\hat{Y} = 1 \mid Y = a, W) = \int P_t(\hat{Y} = 1 \mid x, Y = a, W) f_t(x \mid Y = a, W) dx,$$

where $P_t(\hat{Y} = 1 \mid x, Y = a, W) = I[h(x) = 1]$ since the automated procedure $h(\cdot)$ only uses the features X as input, and $f_t(x \mid Y = a, W)$ represents the distribution of the features X at time t among products with $Y = a$ and weight W. We can see that $P_t(\hat{Y} = 1 \mid Y = a, W)$ might depend on the time t and the product weight W only if the distribution of the features X changes from time 0 to t and for different values of the product's weight W among the two groups of products with and without the characteristic of interest. This leads to sufficient conditions for the assumptions above.

Sufficient condition for extrapolation assumption: The distributions of the features X among products with and without the characteristic of interest, and for the different values of importance, are the same at time 0 and at time t, that is, $f_t(x \mid Y = a, W) = f_0(x \mid Y = a, W)$.

This is a conditional version of what is sometimes referred to as the *prior probability shift* assumption [5, 18]. To examine this sufficient condition, let us say that $Y = 1$ indicates that a product contains a defect in a specific attribute. In such case, this condition says that the distribution of the features used to predict defects, among products that are defective $Y = 1$ and that have a specific importance W, is the same at baseline and at time t. In other words, we expect to see the same indications of defects at baseline and at time t among defective products that have the same importance. A similar interpretation would apply among non-defective products.

Sufficient condition for simplifying assumption: The distributions of the features X among products with and without the characteristic of interest Y are the same regardless of the importance of the products, that is, $f(x \mid Y = a, W) = f(x \mid Y = a)$.

Continuing with the example of defects, this condition says that the distribution of the features used to predict defects among defective products is the same regardless of how popular the product is. Namely, we expect to see the same indicators of defects among defective products, regardless of how important they are. A similar interpretation would apply among non-defective products.

The EA is a fundamental assumption that allows us to borrow information from the audited sample at baseline to obtain an estimate for follow-up times. We need this assumption to extrapolate the performance of the classifier $h(\cdot)$. On the other hand, the SA is not strictly necessary, as in principle we can use the audited data at baseline to build models of the probabilities $P_t(\hat{Y} = 1 \mid Y = a, W)$ and obtain a more flexible estimator; we discuss this further in Section 4. Nevertheless, the SA allows us to obtain an initial simple estimator on which we can build and improve upon.

2.3 Dealing with Classifier Thresholds

The proposed estimator (7) of the weighted rate was derived assuming that the classifier $h(\cdot)$ is fixed, however, the classifier might be obtained as $h(x) = I[g(x) > c]$, that is, it depends on thresholding a score $g(x)$. We study two approaches for handling the cutpoint c, although we assume that the score function $g(x)$ is fixed, as in our use cases where it is already trained at the baseline time.

Variance Minimization Given a threshold c, we can use the classifier $h(x) = I[g(x) > c]$ to obtain an estimate $\hat{R}_t = (\hat{R}_t^{raw} - \hat{p}_{1|0})/(\hat{p}_{1|1} - \hat{p}_{1|0})$, where each of $\hat{R}_t^{raw}$, $\hat{p}_{1|1}$ and $\hat{p}_{1|0}$ are implicitly functions of the threshold c. Given the classifier $h(x)$, we can obtain an analytical approximation of the variance of $\hat{R}_t$, as shown in Appendix B at `https://bit.ly/3wJK5Mj`. We denote the estimated variance given threshold c as V_c. The variance minimization approach simply takes a grid of u threshold values, $c_1, \ldots, c_u$, computes the estimated variance given each threshold, $V_1, \ldots, V_u$, and selects the threshold c^* that minimizes the estimated variance. The final weighted rate estimator is computed from the classifier $h(x) = I[g(x) > c^*]$.

Variance minimization has been implemented for quantification before, for instance [17] used it within a mixture model approach to quantification.

Median Sweep Forman [7] studied different strategies for choosing classification thresholds to obtain reliable estimation of the prevalence of a class, and found that the approach known as *median sweep* was the best in terms of leading to the lowest bias. These results were replicated recently [15], and therefore we implement median sweep along with our proposed weighted rate estimator.

Median sweep consists in computing the estimates $\hat{R}_{t,1}, \ldots, \hat{R}_{t,u}$ according to the threshold values in a grid $c_1, \ldots, c_u$, and returning the median estimate. This approach is theoretically justified, given that each of the estimators $\hat{R}_{t,1}, \ldots, \hat{R}_{t,u}$ corresponding to a fixed grid $c_1, \ldots, c_u$ is guaranteed to be statistically consistent, as shown in Theorem 1, and thereby asymptotically unbiased, as long as $\hat{R}_t^{raw}$, $\hat{p}_{1|0}$ and $\hat{p}_{1|1}$ are estimated in a statistically consistent way. Under such reasonable conditions, the median of these individual estimators inherits the statistical consistency and asymptotic unbiasedness.

We also explore the performance of a *trimmed median sweep* approach by Forman [7], who proposed to use median sweep after discarding estimates from thresholds that lead to $|\hat{p}_{1|1} - \hat{p}_{1|0}| < 0.25$, in order to provide more stability to the ACC estimator.

2.4 Confidence Intervals

We also propose approaches to build confidence intervals, using analytical methods and the bootstrap [4].

Analytic Confidence Interval Given an estimator $\hat{R}_t = (\hat{R}_t^{raw} - \hat{p}_{1|0})/(\hat{p}_{1|1} - \hat{p}_{1|0})$ obtained from a specific classifier $h(\cdot)$, for instance obtained from a specific threshold, we can use the analytical variance formula derived in Appendix B (at `https://bit.ly/3wJK5Mj`) to obtain an estimate of the variance $\widehat{\text{var}}(\hat{R}_t)$, and form a confidence interval based on the asymptotic normality of $\hat{R}_t$. A $100(1-\alpha)\%$ confidence interval, with $\alpha \in (0,1)$, is given by $\hat{R}_t \pm z_{1-\alpha/2}\sqrt{\widehat{\text{var}}(\hat{R}_t)}$, where $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of a standard normal distribution, say $z_{0.975} = 1.96$ for a 95% confidence interval. Despite the simplicity of this interval, its actual coverage might be lower than 95%, given that the analytic variance formula $\widehat{\text{var}}(\hat{R}_t)$ is obtained from an asymptotic analysis that might be less accurate for small samples. Furthermore, if the threshold to obtain $\hat{R}_t$ comes from a threshold selection procedure subject to randomness from the sampling, such as the variance minimization approach presented above, then the estimated variance $\widehat{\text{var}}(\hat{R}_t)$ might underestimate the true variance of $\hat{R}_t$, and the analytical confidence interval might not actually have the promised coverage.

Using an analytical confidence interval along with the estimators obtained from the median sweep approach is more challenging, given that deriving the analytical variance of the median of correlated estimators is complex. Instead, we turn our attention to the bootstrap [4] as a flexible way of obtaining estimates of variances and confidence intervals.

Bootstrap Confidence Intervals The basis of the bootstrap [4] is to take samples with replacement from the original sample, of the same size as the original sample, and for each of these new samples repeat the estimation procedure. For instance, if we denote $\hat{R}_t^{\dagger(b)}$ the estimate obtained via variance minimization or median sweep from a bootstrap sample b, then we can use the bootstrap estimates obtained from B independent bootstrap samples, $\hat{R}_t^{\dagger(1)}, \ldots, \hat{R}_t^{\dagger(B)}$, to compute confidence intervals in two ways. First, we can simply find the $\alpha/2$ and $1-\alpha/2$ quantiles of the bootstrap estimates, and take those as the bounds of the $100(1-\alpha)\%$ confidence interval; we refer to this as the *bootstrap quantile* approach. A second approach is to compute the variance of the bootstrap estimates, $\widehat{\text{var}}_{boot}(\hat{R}_t^\dagger)$, and use it to construct confidence intervals as $\hat{R}_t^\dagger \pm z_{1-\alpha/2}\sqrt{\widehat{\text{var}}_{boot}(\hat{R}_t^\dagger)}$, where $\hat{R}_t^\dagger$ is the estimate obtained via variance minimization or median sweep from the original sample; we refer to this as the *bootstrap standard error* approach. Given that in the estimator $\hat{R}_t = (\hat{R}_t^{raw} - \hat{p}_{1|0})/(\hat{p}_{1|1} - \hat{p}_{1|0})$, we assume that the variability from $\hat{R}_t^{raw}$ is negligible in comparison to the variability from $\hat{p}_{1|1}$ and $\hat{p}_{1|0}$, we only apply the bootstrap to the audited sample collected at baseline.

In the next section we compare the actual coverage of the five confidence intervals detailed here: for variance minimization we compute the analytical approach in addition to the two bootstrap approaches, whereas for median sweep we compare the two bootstrap confidence intervals.

3 Performance Comparison

3.1 Existing Estimators

Weighted rates of the form $R_t = \sum_{i=1}^{N_t} W_{it}Y_{it}/\sum_{i=1}^{N_t} W_{it}$ can be estimated using techniques for cost-quantification, as mentioned in Section 1.1: since in our use cases the weights W_{it} are known, we only need to estimate the total $\sum_{i=1}^{N_t} W_{it}Y_{it}$. To the best of our knowledge, the existing approaches for cost-quantification are due to Forman [7, 8, 10]. Here we consider two of those.

First, the *classify and total* (CT) approach simply replaces Y_{it} with $\hat{Y}_{it}$, and so this estimator leads to our $\hat{R}_t^{raw}$; we consider this estimator to show the reader how biased this approach can be. Second, the *grossed-up total* approach takes the CT estimator and multiplies it by the ratio $\hat{r}_t^{acc}/\hat{r}_t^{cc}$, where $\hat{r}_t^{cc} = \sum_{i=1}^{N_t} \hat{Y}_{it}/N_t$ is the classify and count estimator for the simple rate r_t, and $\hat{r}_t^{acc} = (\hat{r}_t^{cc} - \hat{p}_{1|0})/(\hat{p}_{1|1} - \hat{p}_{1|0})$ is its adjusted version. The resulting estimator for the weighted rate is $\hat{R}_t^{gut} = \hat{R}_t^{raw}\hat{r}_t^{acc}/\hat{r}_t^{cc}$. This approach is derived from a rule of three, that is, assuming that these ratios are equal: $\sum_{i=1}^{N_t} W_{it}Y_{it}/\sum_{i=1}^{N_t} W_{it}\hat{Y}_{it} = \sum_{i=1}^{N_t} Y_{it}/\sum_{i=1}^{N_t} \hat{Y}_{it}$.

The remaining approaches proposed by Forman [7, 8] for cost-quantification rely on the following idea. The total weight in the positive class can be written as $\sum_{i=1}^{N_t} W_{it}Y_{it} = \mu_t^+ N_t r_t$, where $r_t = \sum_{i=1}^{N_t} Y_{it}/N_t$ is the simple rate and $\mu_t^+ = \sum_{i=1}^{N_t} W_{it}Y_{it}/\sum_{i=1}^{N_t} Y_{it}$ is the mean weight among the positive class. If we know or have a good estimate of μ_t^+, then we can simply use quantification techniques to estimate r_t, and then estimate the total cost as $\mu_t^+ N_t \hat{r}_t$. In the applications studied by Forman [7, 8], it was reasonable to assume that μ_t^+ did not change over time, and so it could be estimated from the audited data at baseline. However, programs to improve data quality of e-commerce catalogs often target products with the largest weights, which directly impacts the value of

μ_t^+ over time. Because of this, we do not consider these approaches, as assuming that μ_t^+ is constant is unreasonable in our use cases.

3.2 Simulation Design

To compare the performance of the proposed and existing estimation approaches, we opt for conducting extensive simulation studies where we generate synthetic catalogs under a variety of scenarios that reflect characteristics of our use cases. We opt for this approach, given that we want to obtain an estimation strategy that can be reliably deployed across different circumstances, and a simulation study allows us to control the characteristics of the scenarios that we want to explore. Furthermore, given that we are restricted from publishing results obtained on datasets from our organization, creating synthetic scenarios that reflect characteristics of our use cases seems like a good compromise. We generate synthetic catalogs of size $N_t = 10^6$, and each simulation run involves one catalog for a baseline time $t = 0$ and one for a follow-up time $t > 0$. The exact details of their construction are given in Appendix C at `https://bit.ly/3wJK5Mj`, but here we present a brief description.

For baseline, a catalog is generated with a proportion of defective items, $r_0 = 0.1, 0.2, 0.3$. We then generate product weights using distributions obtained from actual numbers of visits to product pages in the Amazon.com website during a fixed time period and for a specific category of products. This is done such that the weighted rate R_0 is a specific fraction d of the proportion of defective products r_0. Given that for many of our use cases we expect defects to be more prevalent among products with lower weights, we expect $R_0 < r_0$. In particular, we take $R_0 = d\, r_0$ for $d = 1/4, 1/2, 3/4$. We generate synthetic product features to predict defects so that we obtain different levels of classification difficulty, here characterized by the true and false positive rates of the classifier; we consider three scenarios by fixing TPR$=0.5, 0.7, 0.9$ and FPR$=0.05$, which reflect a range of use cases, from cases where classifiers are in their infancy and do no yet reach high accuracy, to cases where mature classifiers have been developed and reach relatively high accuracy.

To generate the catalog at time $t > 0$, we fix different values of the percent change $\Delta = 100(R_t - R_0)/R_0$ of the weighted rate from time 0 to $t > 0$; we take $\Delta = -50\%, -25\%, +25\%, +50\%$ to cover a range of relatively large changes. The different combinations of Δ and R_0 considered here lead to a wide range of scenarios for the weighted rate R_t going from 1.25% to 33.75%, which is representative of the rates that we observe in our use cases.

Given a pair of synthetic catalogs for baseline and for time $t > 0$, we repeat 1000 times the estimation process of the weighted rate R_t with each of the competing estimation approaches. For all approaches, we start with sampling with replacement n_0 products from the baseline catalog, and record their ground truth values Y_{i0} (analog to auditing), along with their weights W_{i0} and model scores $g(X_{i0})$. We explore three sampling scenarios with $n_0 = 500, 1000, 2000$. In this simulation study we do not consider sampling from the catalog at time t, as we use the exact R_t^{raw} in computing the estimator (7), given that R_t^{raw} only depends on the classifier predictions $\hat{Y}_{it} = I[g(X_{it}) > c]$, which do not involve auditing resources. If this is not tenable in practice, we need to estimate R_t^{raw} using a large sample such that its induced variability is negligible in comparison with the baseline sample.

3.3 Results

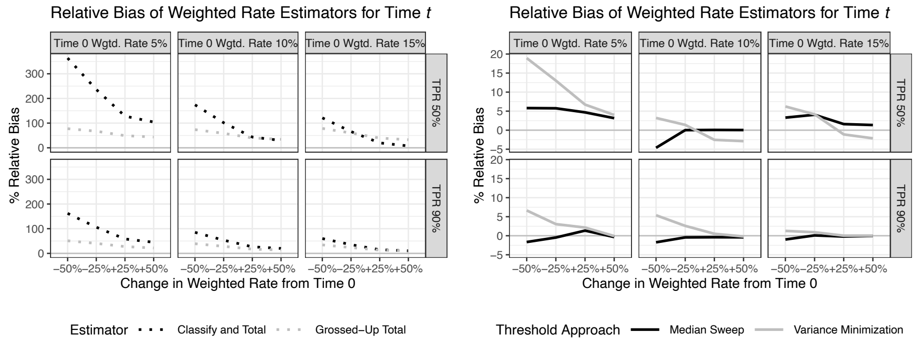
Estimators' Bias and Variance For each of the catalog scenarios described above, we summarize the performance of the different estimation approaches in terms of their bias and standard deviation. In Figures 1a and 1b we present the bias results for sample size $n_0 = 1000$ and for baseline weighted rates such that $R_0 = r_0/2$; the results for other n_0 and relationships between R_0 and r_0 are similar to the results presented here, in terms of leading to the same conclusions on which estimation approach is best. We also omit results for TPR=70%, as the performance is in between that of TPR=50% and TPR=90%. The vertical axis in the panels of Figures 1a and 1b show the estimation bias as a percentage of the true value R_t.

In Figure 1a we present the results for the classify and total, and the grossed-up total approaches [7, 8], which in some scenarios lead to relative bias of up to 350% and 90% respectively. The bias obtained from these approaches is too large to consider them reliable, and so we do not further study them.

In Figure 1b, we present the bias results for our proposed approaches, that is, estimator (7) along with median sweep (MS) or variance minimization (VM) to handle the classification threshold. The performance of the trimmed MS approach is virtually the same as the basic MS, so we omit it. To illustrate the results, consider the top left panel in Figure 1b, which shows a relative bias for the VM approach of almost 20% when the initial (baseline) weighted rate is 5% and the change is -50%, that is, when the weighted rate that we want to estimate at time t is 2.5%. In such case, a 20% relative bias means that the VM approach is on average returning 3% instead of 2.5%. While this is a small bias overall, the MS approach has relative biases of less than around 6% across all scenarios considered here. Undoubtedly, MS leads to a more reliable estimation approach in terms of bias, although the performance of the VM approach comes in close.

We can also see from comparing the rows of panels in Figure 1b that working with a high quality classifier (TPR=90%) generally leads to lower biases, especially when the weighted rates are small. Figure 1b also indicates that it is easier to unbiasedly estimate larger weighted rates. Another striking conclusion from looking at the first row of Figure 1b is that even with a very low quality classifier (TPR=50%) we can still obtain estimation approaches with relatively small bias, an encouraging sign of the reliability of the proposed estimation approaches for different use cases.

A reliable estimator should also have a small variance. In Figure 2 we present the standard deviation of the proposed estimation approaches under the same conditions presented for Figure 1b. We find that in most scenarios both VM and MS lead to nearly the same standard deviation, but VM can sometimes lead to higher estimation variance. This result seems counter-intuitive, given that



(a) Existing methods.

(b) Proposed methods.

Fig. 1: Relative bias of classifier-based weighted rate estimation approaches. Note the different scales of the vertical axes.

by design VM should lead to the lowest variance. However, VM here uses an analytic approximation to the actual variance of the estimator based on large samples, which leads to an approach that does not actually reduce the estimation variance for small samples. Additionally, a factor that might contribute to the good performance of MS is that, in a sense it corresponds to an ensemble of classifiers, one per threshold in our grid, which are working together to estimate R_t; ensemble methods are known to both reduce bias and variance of learning algorithms [3].

Confidence Intervals' Coverage and Length We now present the performance of the five methods to build confidence intervals described in Section 2.4. If a procedure to construct confidence intervals truly leads to a confidence level of $100(1-\alpha)\%$, that means that if we were to repeat the measurement process (starting from random sampling) many times, then $100(1-\alpha)\%$ of those times the observed confidence interval would contain the true value of the parameter. Unfortunately, some confidence interval procedures might be misleading if their actual coverage is different from their nominal one. To ensure that a confidence interval procedure is reliable, it is customary to conduct a simulation study where we repeat the measurement process many times under a fixed set of conditions, and compute the actual coverage or confidence of the confidence intervals by computing the proportion of times that the intervals contain the true value of the parameter of interest. A confidence interval procedure is reliable if the actual coverage is around the nominal one.

In Figure 3 we present the actual coverage of the five confidence interval procedures described in Section 2.4. Undoubtedly, the bootstrap quantile confidence interval obtained from the median sweep procedure is the most reliable of these five approaches, given that its actual coverage is nearly the nominal 95%. In fact, the performance of the four bootstrap-based confidence intervals is gener-

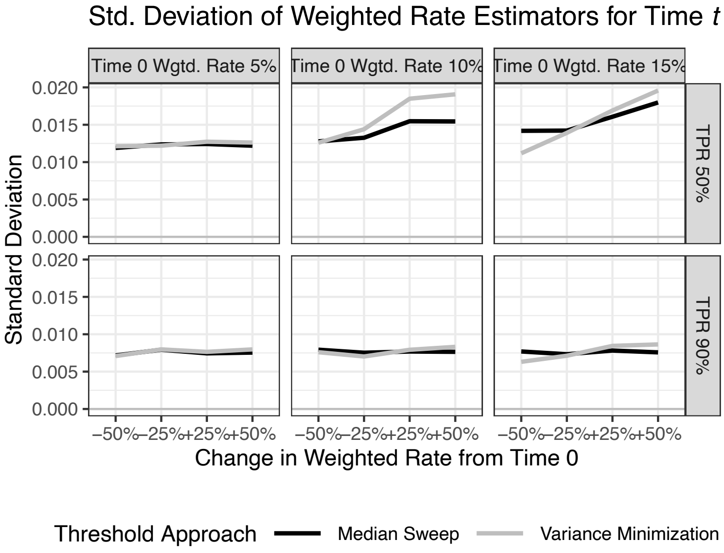


Fig. 2: Standard deviation of proposed weighted rate estimation approaches.

ally reasonable. The worst performance overall is obtained from the confidence interval based on the analytic approximate variance of the estimator obtained from the variance minimization approach. This might occur due to the analytic variance formula not accounting for the variability that comes from the threshold selection, which in turn leads to lower actual coverage of the analytic confidence interval.

Finally, an important property of a good confidence interval procedure is that it does not lead to unnecessarily wide confidence intervals. In this simulation study we also computed the average length of the confidence intervals obtained under each approach, and found that the average lengths are very similar for all approaches across all scenarios. In the interest of space, we do not present plots with these results.

Given these results, our final recommendation is to use median sweep to deal with the thresholds in the classifiers, and to use bootstrap quantile confidence intervals to quantify the uncertainty in the estimation.

4 Discussion and Extensions

Our proposed estimation approach, using median sweep to deal with the thresholds of the classifiers and bootstrap confidence intervals to quantify estimation uncertainty, is currently being implemented in our organization to produce estimates of weighted rates for several types of catalog defects. Our implementation

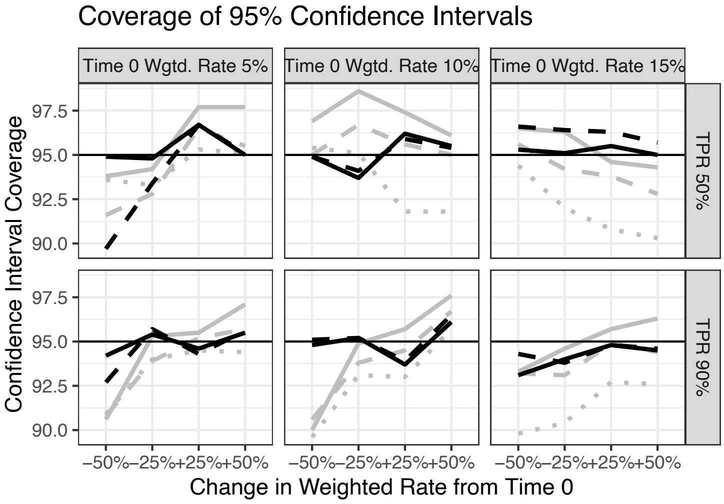


Fig. 3: Actual coverage of nominal 95% confidence intervals (CIs). CIs based on variance minimization: grey dotted lines: analytic confidence interval; grey dashed lines: bootstrap standard error; grey solid lines: boostrap quantiles. CIs based on median sweep: black dashed lines: bootstrap standard error; black solid lines: boostrap quantiles.

consists of measurement cycles, which are marked by baseline times, when we collect audited data, and followed up by automated estimation.

Intuitively, for follow-up times close to the baseline time of the cycles, the proposed estimation approach should be reliable, given that the extrapolation assumption (EA) should approximately hold. As we move farther away from the baseline, the EA might become more questionable. In our use cases, we plan to start from short measurement cycles, say monthly periods, and based on the audited data test the hypothesis of whether the TPR and FPR are the same at the beginning of the cycles. If we repeatedly fail to reject the hypothesis, we expand the measurement cycles, as this indicates that the EA holds for longer in that particular use case.

Regarding the simplifying assumption (SA) used to derive our proposed estimator, it says that the TPR and FPR do not depend on the product weights. This seems initially reasonable, given that the classifiers that we work with use product features exclusively, and not measures of engagement of customers with the products. Nevertheless, the SA can be examined using audited data, for instance by regressing the predicted indicators of defects on the weights, separately for audited products with and without the defect. For use cases when there is evidence of an association, a simple solution is to stratify the estimation domain based on weight intervals, proceed with the estimation as described here separately within each stratum, and aggregate the per-stratum estimates to obtain an overall estimate of the weighted rate, where the aggregation is done weighting the strata by their relative share of the products' weights. This stratified approach requires the SA to hold within stratum, which is more tenable. Intuitively, in the extreme case where there is one stratum per weight value the assumption holds exactly. However, while estimation based on a very fine stratification will alleviate the bias induced by violating SA, it will lead to a large estimation variance. Selecting the right stratification then involves a bias-variance tradeoff which will change depending on the use case.

Acknowledgements The authors would like to thank George Forman, Bunyamin Sisman, Kee Kiat Koo, Florian Verhein, Wenyi Wu, and Vito Mandorino for their insightful feedback.

References

1. Amazon Seller Central: Suggest changes to your product detail page. `https://sellercentral.amazon.com/gp/help/external/200335450?ref=efph_200335450_cont_201950630&language=en_US`, accessed: 2022-06-13
2. Bishop, Y.M.M., Fienberg, S.E., Holland, P.W.: Discrete Multivariate Analysis: Theory and Practice. Springer (1974)
3. Dietterich, T.G.: Ensemble learning. In: The Handbook of Brain Theory and Neural Networks, p. 405. 2nd edn. (2003)
4. Efron, B., Tibshirani, R.: Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science pp. 54–75 (1986)
5. Fawcett, T., Flach, P.A.: A response to webb and ting's "on the application of roc analysis to predict classification performance under varying class distributions". Machine Learning **58**(1), 33–38 (2005)
6. Forman, G.: Counting positives accurately despite inaccurate classification. In: Proceedings of the European Conference on Machine Learning (ECML'05). pp. 564–575 (2005)
7. Forman, G.: Quantifying trends accurately despite classifier error and class imbalance. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'06). pp. 157–166. ACM (2006)
8. Forman, G.: Quantifying counts and costs via classification. Data Min. Knowl. Discov. **17**(2), 164–206 (2008)
9. Gart, J.J., Buck, A.A.: Comparison of a screening test and a reference test in epidemiologic studies ii. a probabilistic model for the comparison of diagnostic tests. Am. J. Epidemiol. **83**(3), 593–602 (1966)
10. González, P., Castaño, A., Chawla, N.V., Coz, J.J.D.: A review on quantification learning. ACM Computing Surveys **50**(5), 74 (2017). https://doi.org/https://doi.org/10.1145/3117807
11. Grassia, A., Sundberg, R.: Statistical precision in the calibration and use of sorting machines and other classifiers. Technometrics **24**(2), 117–121 (1982)
12. Levy, P.S., Kass, E.H.: A three-population model for sequential screening for bacteriuria. Am. J. Epidemiol. **91**(2), 148–154 (1970)

13. Maletzke, A., dos Reis, D., Cherman, E., Batista, G.: Dys: A framework for mixture models in quantification. In: The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19). pp. 4552–4560. AAAI (2019)
14. Meertens, Q.A., Diks, C.G.H., van den Herik, H.J., Takes, F.W.: Improving the output quality of official statistics based on machine learning algorithms (2021)
15. Schumacher, T., Strohmaier, M., Lemmerich, F.: A comparative evaluation of quantification methods (2021)
16. Särndal, C.E., Swensson, B., Wretman, J.: Model Assisted Survey Sampling. Springer-Verlag Publishing (1992)
17. Tasche, D.: Minimising quantifier variance under prior probability shift (2021)
18. Vaz, A.F., Izbicki, R., Stern, R.B.: Quantification under prior probability shift: the ratio estimator and its extensions. Journal of Machine Learning Research **20**(79), 1–33 (2019)
19. Wasserman, L.: All of statistics: a concise course in statistical inference. Springer Science & Business Media (2013)