Data Mining and Knowledge Discovery manuscript No.
(will be inserted by the editor)

Regularization-Based Methods for Ordinal Quantification

Mirko Bunse[1] · **Alejandro Moreo**[2] ·
Fabrizio Sebastiani[2] · **Martin Senz**[1]

Received: June 2023 / Accepted: date

Abstract Quantification, i.e., the task of predicting the class prevalence values in sets of unlabeled data items, has received increased attention in recent years. However, most quantification research has concentrated on developing algorithms for binary and multi-class problems in which the classes are not ordered. Here, we study the ordinal case, i.e., the case in which a total order is defined on the set of $n > 2$ classes. We give three main contributions to this field. First, we create and make available two datasets for ordinal quantification (OQ) research that overcome the inadequacies of the previously available ones. Second, we experimentally compare the most important OQ algorithms proposed in the literature so far. To this end, we bring together algorithms proposed by authors from very different research fields, such as data mining and astrophysics, who were unaware of each others' developments. Third, we propose a novel class of regularized OQ algorithms, which outperforms existing algorithms in our experiments. The key to this gain in performance is that our regularization prevents ordinally implausible estimates, assuming that ordinal distributions tend to be smooth in practice. We informally verify this assumption for several real-world applications.

Keywords Quantification · Class prior estimation · Learning to quantify · Ordinal classification · Unfolding

1 Introduction

Quantification is a supervised learning task that consists of training a predictor, on a set of labeled data items, that estimates the relative frequencies $p_\sigma(y_i)$ (a.k.a. *prevalence values*, or *prior probabilities*, or *class priors*) of the classes

(1) Lamarr Institute for Machine Learning and Artificial Intelligence, TU Dortmund University, 44227 Dortmund, Germany, E-mail: {mirko.bunse,martin.senz}@cs.tu-dortmund.de
(2) Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy, E-mail: {alejandro.moreo,fabrizio.sebastiani}@isti.cnr.it

of interest $\mathcal{Y} = \{y_1, \ldots, y_n\}$ in a sample σ of unlabeled data items (Forman, 2005) – see also (Esuli et al., 2023; González et al., 2017) for recent surveys. In other words, a trained *quantifier* (i.e., an estimator of class prevalence values) must return a *predicted distribution* $\hat{\mathbf{p}}_\sigma = (\hat{p}_\sigma(y_1), \ldots, \hat{p}_\sigma(y_n))$ of the unlabeled data items in σ across the classes in $\mathcal{Y}$, where $\hat{\mathbf{p}}_\sigma$ must coincide as much as possible with the true, unknown distribution $\mathbf{p}_\sigma$. Quantification is also known as "learning to quantify", "supervised class prevalence estimation", and "class prior estimation".

Quantification is important in many disciplines, e.g., market research, political science, ecological modeling, the social sciences, and epidemiology. By their own nature, these disciplines are only interested in aggregate (as opposed to individual) data. Hence, classifying individual unlabeled instances is usually not a primary goal in these fields, while estimating the prevalence values $p_\sigma(y_i)$ of the classes of interest is. For instance, when classifying the tweets about a certain entity (e.g., about a political candidate) as displaying either a Positive or a Negative stance towards the entity, political scientists or market researchers are usually not interested in the class of a specific tweet, but in the fraction of these tweets that belong to each class (Gao and Sebastiani, 2016).

A predicted distribution $\hat{\mathbf{p}}_\sigma$ could, in principle, be obtained by means of the "classify and count" method (CC), i.e., by training a standard classifier, classifying all the unlabeled data items in σ, and computing the fractions of data items that have been assigned to each class in $\mathcal{Y}$. However, it has been shown that CC delivers poor prevalence estimates, and especially so when the application scenario suffers from *prior probability shift* (Moreno-Torres et al., 2012), the (ubiquitous) phenomenon according to which the distribution $\mathbf{p}_U$ of the *unlabeled* test data items U across the classes is different from the distribution $\mathbf{p}_L$ of the *labeled* training data items L. As a result, a plethora of quantification methods have been proposed in the literature – see e.g., (Bella et al., 2010; Esuli et al., 2018; González and del Coz, 2021; González-Castro et al., 2013; Pérez-Gállego et al., 2019; Saerens et al., 2002) – whose goal is to generate accurate class prevalence estimations even in the presence of prior probability shift.

The vast majority of the methods proposed so far deals with quantification tasks in which $\mathcal{Y}$ is a plain, unordered set. Very few methods, instead, deal with *ordinal quantification* (OQ), the task of performing quantification on a set of $n > 2$ classes on which a total order "$\prec$" is defined. Ordinal quantification is important, though, because totally ordered sets of classes ("ordinal scales") arise in many applications, especially ones involving human judgments. For instance, in a customer satisfaction endeavor, one may want to estimate how a set of reviews of a certain product is distributed across the set of classes $\mathcal{Y} =$ {1Star, 2Stars, 3Stars, 4Stars, 5Stars}, while a social scientist might want to find how inhabitants of a certain region are distributed in terms of their happiness with health services in the area, i.e., how they are distributed across the classes in $\mathcal{Y} =$ {VeryUnhappy, Unhappy, Happy, VeryHappy}.

As a field, quantification is inherently related to the field of classification. This is especially true of the so-called "aggregative" family of quantification

algorithms, which, in order to return prevalence estimates for the classes of interest, rely on the output of an underlying classifier. As such, a natural and straightforward approach to ordinal quantification might simply consist of replacing, within a multi-class aggregative quantification method, the standard multi-class classifier with an *ordinal* classifier, i.e., with a classifier specifically devised for classifying data items according to an ordered scale. However, the experiments we have run (see Section 6.3) show that this simple solution does not suffice; instead, actual OQ methods are required.

This paper is an extension to an initial study on OQ that we conducted recently (Bunse et al., 2022). It contributes to the field of OQ in four ways.

First, we develop and make publicly available two datasets for evaluating OQ algorithms, one consisting of textual product reviews and one consisting of telescope observations. Both datasets stem from scenarios in which OQ arises naturally, and they are generated according to a strong, well-tested protocol for the generation of datasets oriented to the evaluation of quantifiers. This contribution fills a gap in the state-of-the-art because the datasets that have previously been used for the evaluation of OQ algorithms were inadequate, for reasons we discuss in Section 2.

Second, we perform the most extensive experimental comparison of OQ algorithms that have been proposed in the literature to date, using the two previously mentioned datasets. This contribution is important because some algorithms (e.g., the ones of Section 4.3.1 and 4.3.2) have so far been evaluated only on an arguably inadequate test-bed (see Section 2) and because some other algorithms (e.g., the ones of Section 4.3 and 4.4) have been developed by authors from very different research fields, such as data mining and astrophysics, which were utterly unaware of each others' developments.

Third, we formulate an *ordinal plausibility assumption*, i.e., the assumption that ordinal distributions that appear in practice tend to be "smooth". Here, a smooth distribution is one that can be represented by a histogram with at most a limited amount of (upward or downward) "humps". We informally show that this assumption is verified in many real-world applications.

Fourth, we propose a class of new OQ algorithms, which introduces ordinal regularization into existing quantification methods. The effect of this regularization is to discourage the prediction of distributions that are not smooth and, hence, would tend to lack plausibility in OQ tasks. Using the datasets mentioned above, we run extensive experiments which show that our algorithms, which are based on ordinal regularization, outperform their state-of-the-art competitors. In the interest of reproducibility, we make publicly available all the datasets and all the code that we use.

This paper is organized as follows. In Section 2 we review past work on ordinal quantification. Section 3 is devoted to presenting preliminaries, including an illustration of the evaluation measures that we are going to use in the paper (Section 3.2) and our formulation of the ordinal plausibility assumption (Section 3.3). In Section 4 we present previously proposed ordinal quantification algorithms, while in Section 5 we detail the ones that we propose in this work. Section 6 is devoted to our experimental comparison of new and

existing OQ algorithms. In Section 7 we look back at the work we have done and discuss alternative notions of ordinal plausibility. We finish in Section 8 by giving concluding remarks and by discussing future work. The Appendix includes a discussion on how reasonable it is to postulate the smoothness of real-life ordinal distributions (Section A), and additional experimental results obtained (a) by using alternative measures of the prediction error of ordinal quantifiers (Section B.1), or (b) by using alternative datasets (Section B.2).

2 Related work

Quantification, as a task of its own right, was first proposed by Forman (2005), who observed that some applications of classification only require the estimation of class prevalence values and that better methods than "classify and count" can be devised for this purpose. Since then, many methods for quantification have been proposed (Esuli et al., 2023; González et al., 2017). However, most of these methods tackle the binary and/or multi-class problem with unordered classes. *Ordinal* quantification was first discussed in (Esuli and Sebastiani, 2010), where an evaluation measure (the *Earth Mover's Distance* – see Section 3.2) was proposed for it. However, it was not until 2016 that the first true OQ algorithms were developed, the *Ordinal Quantification Tree* (OQT – see Section 4.3.1) by Da San Martino et al. (2016) and *Adjusted Regress and Count* (ARC – see Section 4.3.2) by Esuli (2016). In the same years, the first data challenges that involved OQ were staged (Higashinaka et al., 2017; Nakov et al., 2016; Rosenthal et al., 2017). However, except for OQT and ARC, the participants in these challenges used "classify and count" with highly optimized classifiers, instead of true OQ methods; this attitude persisted also in later challenges (Zeng et al., 2019, 2020), likely due to a general lack of awareness in the scientific community that more accurate methods than "classify and count" existed.

Unfortunately, the data challenges, in which OQT and ARC were evaluated (Nakov et al., 2016; Rosenthal et al., 2017), tested each quantification method only on a single sample of unlabeled data items, which consisted of the entire test set. This evaluation protocol is not adequate for quantification because quantifiers issue predictions for sets of data items, not for individual data items as in classification. Measuring a quantifier's performance on a single sample is thus akin to, and as insufficient as, measuring a classifier's performance on a single data item. As a result, our current knowledge of the relative merits of OQT and ARC lacks solidity.

However, even before the previously mentioned developments had taken place, methods that we would now call OQ algorithms had been proposed within experimental physics. In this field we often need to estimate the distribution of a continuous physical quantity. However, physicists consider a histogram approximation of a continuous distribution sufficient for many physics-related analyses (Blobel, 2002). This conventional simplification essentially maps the values of a continuous target quantity into a set of classes endowed

with a total order, and the problem of estimating the continuous distribution becomes one of OQ (Bunse, 2022b). Early on, physicists had termed this problem "unfolding" (Blobel, 1985; D'Agostini, 1995), a term that was unfamiliar to data mining / machine learning researchers and that, hence, prevented them from realizing that the "ordinal quantification" algorithms they used and the "unfolding" algorithms that physicists used, were actually addressing the very same task. This connection was discovered only recently by Bunse (2022b), who argued that OQ and unfolding are in fact the same problem. In the following we deepen these connections, to find that ordinal regularization techniques proposed in the physics literature are able to improve the ability of well-known quantification methods at performing OQ.

Castaño et al. (2024) have recently proposed a different approach to OQ. This approach does not rely on regularization, but on loss functions tailored to the OQ setting. The two approaches are orthogonal, in the sense that they target different characteristics of quantification algorithms which can be combined. In this paper, we therefore extend our initial study (Bunse et al., 2022) with combinations of the two approaches, i.e., with algorithms that use ordinal loss functions in conjunction with ordinal regularization.

3 Preliminaries

In this section, we introduce our notation, we discuss measures for evaluating the prediction error of OQ methods, and we provide a measure for evaluating the smoothness of ordinal distributions. Understanding these types of measures will help us better understand the OQ methods that are to be presented in Sections 4 and 5.

3.1 Notation

By $\mathbf{x} \in \mathcal{X}$ we indicate a data item drawn from a domain $\mathcal{X}$, and by $y \in \mathcal{Y}$ we indicate a class drawn from a set of classes $\mathcal{Y} = \{y_1, \ldots, y_n\}$, also known as a *code frame*; in this paper we will only consider code frames with $n > 2$, on which a total order "$\prec$" is defined. The symbol σ denotes a *sample*, i.e., a non-empty set of unlabeled data items in $\mathcal{X}$, while $L \subset \mathcal{X} \times \mathcal{Y}$ denotes a set of labeled data items $(\mathbf{x}, y)$, which we use to train our quantifiers.

By $p_\sigma(y)$ we indicate the true prevalence of class y in sample σ, by $\hat{p}_\sigma(y)$ we indicate an estimate of this prevalence, while by $\hat{p}^Q_\sigma(y)$ we indicate an estimate of $p_\sigma(y)$ as obtained by a quantification method Q that receives σ as input. By $\mathbf{p}_\sigma = (p_\sigma(y_1), \ldots, p_\sigma(y_n))$ we indicate a distribution of the elements of σ across the classes in $\mathcal{Y}$; $\hat{\mathbf{p}}_\sigma$ and $\hat{\mathbf{p}}^Q_\sigma$ can be interpreted analogously. All of $\mathbf{p}_\sigma$, $\hat{\mathbf{p}}_\sigma$, $\hat{\mathbf{p}}^Q_\sigma$, are probability distributions, i.e., are elements of the unit (n-1)-simplex Δ^{n-1} (aka *probability simplex*, or *standard simplex*), defined as

$$\Delta^{n-1} = \left\{ (p_1, \ldots, p_n) \in \mathbb{R}^n : p_i \geq 0, \sum_{i=1}^{n} p_i = 1 \right\} \tag{1}$$

In other words, Δ^{n-1} is the domain of all vectors that represent probability distributions over $\mathcal{Y}$.

As customary, we use lowercase boldface letters ($\mathbf{p}$, $\mathbf{q}$, ...) to denote vectors, and uppercase boldface letters ($\mathbf{M}$, $\mathbf{C}$, ...) to denote matrices or tensors; we use subscripts to denote their elements and projections, e.g., we use $\mathbf{p}_i$ to denote the i-th element of $\mathbf{p}$, $\mathbf{M}_{ij}$ to denote the element of $\mathbf{M}$ at the i-th row and j-th column, and bullets to indicate projections (with, e.g., $\mathbf{M}_{i\bullet}$ indicating the i-th row of $\mathbf{M}$). We indicate distributions in boldface in order to stress the fact that they are *vectors* of class prevalence values and because we will formulate most of our quantification methods by using matrix notation. We will often write $\mathbf{p}$, $\hat{\mathbf{p}}$, $\hat{\mathbf{p}}^Q$, instead of $\mathbf{p}_\sigma$, $\hat{\mathbf{p}}_\sigma$, $\hat{\mathbf{p}}^Q_\sigma$, thus omitting the indication of sample σ when clear from context.

3.2 Measuring quantification error in ordinal contexts

The main function for measuring quantification error in ordinal contexts that we use in this paper is the *Normalized Match Distance* (NMD), defined by Sakai (2018) as

$$\mathrm{NMD}(\mathbf{p}, \hat{\mathbf{p}}) = \frac{1}{n-1}\,\mathrm{MD}(\mathbf{p}, \hat{\mathbf{p}}) \tag{2}$$

where $\frac{1}{n-1}$ is just a normalization factor that allows NMD to range between 0 (best prediction) and 1 (worst prediction).[1] Here, MD is the well-known *Match Distance* (Werman et al., 1985), defined as

$$\mathrm{MD}(\mathbf{p}, \hat{\mathbf{p}}) = \sum_{i=1}^{n-1} d(y_i, y_{i+1}) \cdot |\hat{P}(y_i) - P(y_i)| \tag{3}$$

where $P(y_i) = \sum_{j=1}^{i} p(y_j)$ is the prevalence of y_i in the cumulative distribution of $\mathbf{p}$, $\hat{P}(y_i) = \sum_{j=1}^{i} \hat{p}(y_j)$ is an estimate of it, and $d(y_i, y_{i+1})$ is the "semantic distance" between consecutive classes y_i and y_{i+1}, i.e., the cost we incur in mistaking y_i for y_{i+1} or vice versa. Throughout this paper, we assume $d(y_i, y_{i+1}) = 1$ for all $i \in \{1, 2, \ldots, n-1\}$.

MD is a widely used measure in OQ evaluation (Bunse et al., 2018; Castaño et al., 2024; Da San Martino et al., 2016; Esuli and Sebastiani, 2010; Nakov et al., 2016; Rosenthal et al., 2017), where it is often called *Earth Mover's Distance* (EMD); in fact, MD is a special case of EMD as defined by Rubner et al. (1998).[2] Since NMD and MD differ only by a fixed normalization factor, our experiments closely follow the tradition in OQ evaluation. The use of NMD

[1] Alternative measures for quantification error are discussed in Section B.1.

[2] To see the intuition upon which MD and EMD are based, if the two distributions are interpreted as two different ways of scattering a certain amount of "earth" across different "heaps", their MD and EMD are defined to be the minimum amount of work needed for transforming one set of heaps into the other, where the work is assumed to correspond to the sum of the amounts of earth moved times the distance traveled for moving them. In other words, MD and EMD may be seen as computing the minimal "cost" incurred in

is advantageous because the presence of the normalization factor $\frac{1}{n-1}$ allows us to compare results obtained on different datasets characterized by different numbers n of classes; this would not be possible with MD or EMD, whose scores tend to increase with n.

To obtain an overall score for a quantification method Q on a dataset, we apply Q to each test sample σ. The resulting estimated distribution $\hat{\mathbf{p}}_\sigma^Q$ is then compared to the true distribution $\mathbf{p}_\sigma$ via NMD, which yields one NMD value for each test sample. The final score for method Q is the average NMD value across all samples σ in the test set, which characterizes the average prediction error of Q. We test for statistically significant differences between quantification methods in terms of a paired Wilcoxon signed-rank test.

3.3 Measuring the plausibility of distributions in ordinal contexts

Any probability distribution over $\mathcal{Y}$ is a legitimate ordinal distribution. However, some ordinal distributions, though legitimate, are hardly *plausible*, i.e., they hardly occur in practice. For instance, assume that we are dealing with how a set of book reviews is distributed across the set of classes $\mathcal{Y} = \{$1Star, 2Stars, 3Stars, 4Stars, 5Stars$\}$; a distribution such as

$$\mathbf{p}_{\sigma_1} = (0.20, 0.10, 0.05, 0.20, 0.45)$$

is both legitimate and plausible, while a distribution such as

$$\mathbf{p}_{\sigma_2} = (0.02, 0.47, 0.02, 0.47, 0.02)$$

is legitimate but hardly plausible.

What makes $\mathbf{p}_{\sigma_2}$ lack plausibility is the fact that it describes a highly dissimilar behavior of neighboring classes, despite the semantic similarity that ordinality imposes on the class neighborhood. As shown in Figure 1, the dissimilarity of neighboring classes in $\mathbf{p}_{\sigma_2}$ manifests in sharp "humps" of prevalence values. For instance, a sequence (0.02, 0.47, 0.02) of prevalence values, such as the one that occurs in $\mathbf{p}_{\sigma_2}$ for the last three classes (an "upward" hump), hardly occurs in practice. Sequences such as (0.47, 0.02, 0.47), such as the one that occurs in $\mathbf{p}_{\sigma_2}$ for the middle three classes (a "downward" hump), also hardly occur in practice.

In the rest of this paper, a *smooth* ordinal distribution is one that tends not to exhibit (upward or downward) humps of prevalence values across consecutive classes; conversely, a *jagged* ordinal distribution is one that tends to exhibit such humps. We will thus take smoothness to be a *measure of ordinal plausibility*, i.e., a measure of how likely it is, for a distribution with a certain form, to occur in real-life applications of OQ.

transforming one distribution into the other, where the cost is computed as the probability mass that needs to be shuffled around from one class to another, weighted by the "semantic distance" between the classes involved. The use of MD is restricted to the case in which a total order on the classes is assumed; EMD is more general, since it also applies to cases in which no order on the classes is assumed.

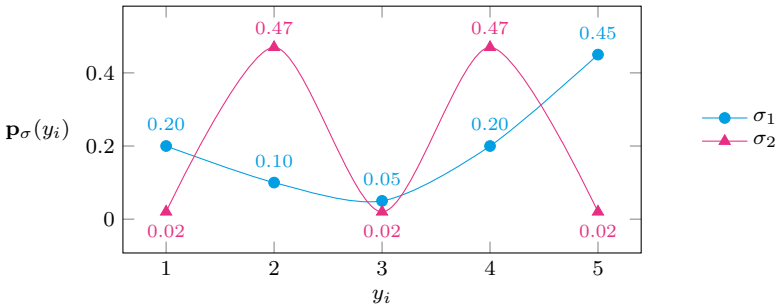

Fig. 1 Two ordinal distributions $\mathbf{p}_{\sigma_1}$ (blue circles) and $\mathbf{p}_{\sigma_2}$ (red triangles). The interpolating lines are displayed only for establishing a visual coherence among the dots.

As a measure of the jaggedness (the opposite of smoothness) of an ordinal distribution we propose using

$$\xi_1(\mathbf{p}_\sigma) = \frac{1}{\min(6, n+1)} \sum_{i=2}^{n-1} (-p_\sigma(y_{i-1}) + 2 \cdot p_\sigma(y_i) - p_\sigma(y_{i+1}))^2 \qquad (4)$$

where $\frac{1}{\min(6,n+1)}$ is just a normalization factor to ensure that $\xi_1(\mathbf{p}_\sigma)$ ranges between 0 (least jagged) and 1 (most jagged); therefore, $\xi_1(\mathbf{p}_\sigma)$ is a measure of jaggedness and (1-$\xi_1(\mathbf{p}_\sigma)$) a measure of smoothness.[3]

The intuition behind Equation 4 is that, for an ordinal distribution to be smooth, the prevalence of a class y_i should be as similar as possible to the average prevalence of its two neighboring classes y_{i-1} and y_{i+1}; $\xi_1(\mathbf{p}_\sigma)$ is nothing else than a (normalized) sum of these (squared) differences across the classes in the code frame. In our example above, $\xi_1(\mathbf{p}_{\sigma_1}) = 0.009$ indicates a very smooth distribution and $\xi_1(\mathbf{p}_{\sigma_2}) = 0.405$ indicates a fairly jagged distribution.

By way of example, Figure 2 displays the class distributions for each of the 28 product categories in the ordinal dataset of 233.1M Amazon product reviews made available by McAuley et al. (2015) (see also Section 6.1.2), while Figure 3 displays the class distribution of the ordinal dataset of the FACT telescope (see also Section 6.1.3). It is evident from these figures that all these ordinal distributions are fairly smooth, in the sense indicated above. For instance, the 28 class distributions from the Amazon dataset tend to exhibit a moderate downward hump in the first three classes (or in the last three classes), but tend to be smooth elsewhere, with their value of $\xi_1(\mathbf{p}_\sigma)$ ranging in [0.007,0.037]; likewise, the class distribution for the FACT telescope also tends to exhibit an upward hump in classes 4 to 6 but to be smooth elsewhere, with a value of $\xi_1(\mathbf{p}_\sigma) = 0.0115$. Appendix A presents other real-life examples, which show that smoothness is a pervasive phenomenon in ordinal distributions.

[3] The subscript "1" indicates that $\xi_1(\mathbf{p}_\sigma)$ measures the deviation of $\mathbf{p}_\sigma$ from a polynomial of degree one. More details on this deviation, as well as alternative measures of jaggedness, are discussed in Section 7.

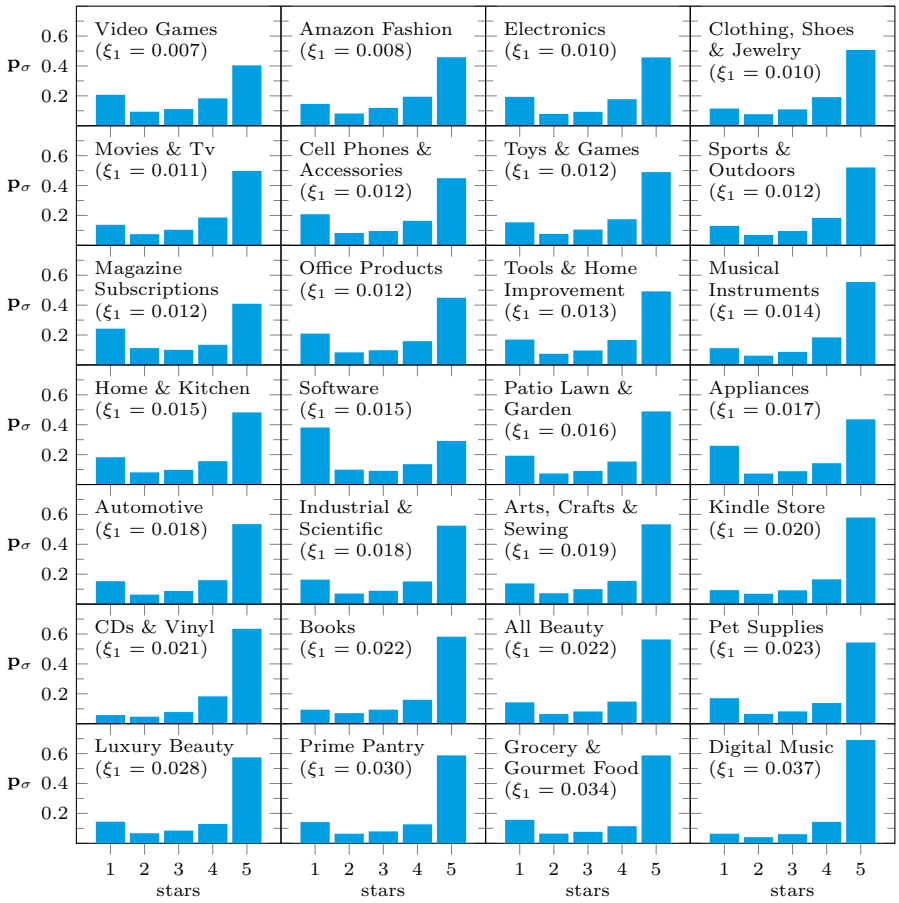

Fig. 2 The class distribution $\mathbf{p}_\sigma$ of each of the 28 product categories in the AMAZON dataset (see Section 6.1.2). The categories are ordered (from left to right, then from top to bottom) in terms of their $\xi_1(\mathbf{p}_\sigma)$ score.

It is easy to see that the most jagged distribution ($\xi_1(\mathbf{p}_\sigma)$=1) is not unique; for instance, assuming a 7-point scale by way of example, distributions

(0.000, 0.000, 1.000, 0.000, 0.000, 0.000, 0.000)
(0.000, 0.000, 0.000, 1.000, 0.000, 0.000, 0.000)
(0.000, 0.000, 0.000, 0.000, 1.000, 0.000, 0.000)

are the most jagged distributions ($\xi_1(\mathbf{p}_\sigma)$=1). The least jagged distribution is also not unique; examples of least jagged distributions ($\xi_1(\mathbf{p}_\sigma)$=0) on a 5-point scale are

(0.200, 0.200, 0.200, 0.200, 0.200)
(0.198, 0.199, 0.200, 0.201, 0.202)
(0.000, 0.100, 0.200. 0.300, 0.400)
(0.202, 0.201, 0.200, 0.199, 0.198)

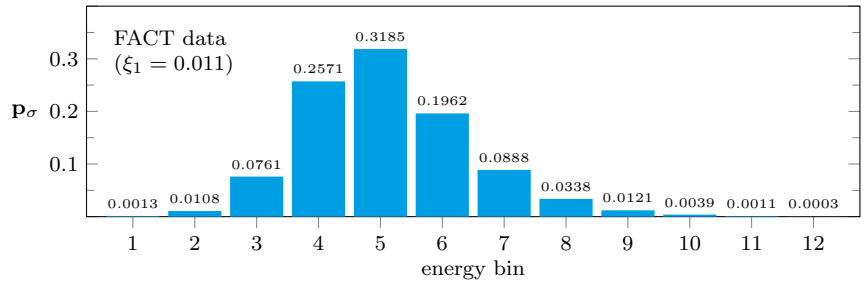

Fig. 3 The class distribution $\mathbf{p}_\sigma$ of the ordinal dataset of the FACT telescope (see Section 6.1.3), along with its $\xi_1(\mathbf{p}_\sigma)$ score.

...

Luckily enough, uniqueness of the most jagged distribution and uniqueness of the least jagged distribution turn out not to be required properties as far as our work is concerned. Indeed, jaggedness plays a central role both in the (regularization-based) methods that we propose (see Section 5) and in the data sampling protocol that we use for testing purposes (see Section 6.1.1), but neither of these contexts requires these uniqueness properties.

4 Existing multi-class quantification methods

In this section we introduce a number of known (non-ordinal and ordinal) multi-class quantification methods that we use as baselines in our experiments. Our novel OQ methods from Section 5 build upon a selection of these baselines.

4.1 Problem setting

In the multi-class quantification setting we want to estimate a distribution $\mathbf{p} \in \Delta^{n-1}$, where $n > 2$ and where Δ^{n-1} is the probability simplex from Equation 1 and where $\mathbf{p}$ represents the class prevalences within a testing sample σ. At our disposal is a validation dataset V, where we denote by V_i those data items that belong to class $y_i \in \mathcal{Y}$, i.e.,

$$V_i \;=\; \big\{\mathbf{x} \in \mathcal{X} : (\mathbf{x}, y) \in V,\, y = y_i\big\} \tag{5}$$

Let $f : \mathcal{X} \to \mathbb{R}^d$ be a *transformation* function that *embeds* any data point into a d-dimensional vector. For example, f might be a soft classifier, so that each data point is represented as an d-dimensional vector of posterior probabilities, with d equal to the number of classes n; or f may instead be a binning function, in which case f returns one-hot d-dimensional vectors with d the number of bins. Many alternative choices for f exist, each of which gives rise to a different quantification method; see, e.g., those of Section 4.2.

Moreover, let $S \in \mathbb{N}^{\mathcal{X}}$ be any bag (or multi-set) of an arbitrary number of data items, where each data item is drawn from the feature space $\mathcal{X}$. For any choice of f and S, we denote by

$$\phi_f(S) \;=\; \frac{1}{|S|} \sum_{\mathbf{x} \in S} f(\mathbf{x}) \tag{6}$$

the mean embedding of S, as represented by f.

With embeddings of this kind, the multi-class quantification problem can be framed as solving for $\mathbf{p} \in \Delta^{n-1}$ the system of linear equations

$$\mathbf{q} = \mathbf{M}\mathbf{p} \tag{7}$$

where the vector $\mathbf{q} = \phi_f(\sigma) \in \mathbb{R}^d$ is a mean embedding of the test sample and the columns of the matrix $\mathbf{M} = [\phi_f(V_1), \cdots, \phi_f(V_n)] \in \mathbb{R}^{d \times n}$ contain the class-wise mean embeddings of the validation sample. Note that V coincides with our training set L if k-fold cross-validation is employed.

Multiple quantification algorithms have been proposed by quantification researchers, and many of them can be seen, as conceptualized by Firat (2016) and formally proven by Bunse (2022b), as different ways of solving Equation 7. In the next sections, when introducing previously proposed quantification algorithms, we indeed present them as different means of solving Equation 7, even if their original proposers did not present them as such. Since we will formulate in this way also our novel algorithms, Equation 7 will act as a unifying framework for quantification methods of different provenance.

A naive solution of Equation 7 would be $\mathbf{M}^\dagger\mathbf{q}$, where $\mathbf{M}^\dagger$ is the Moore-Penrose pseudo-inverse, which exists for any matrix $\mathbf{M}$, even if $\mathbf{M}$ is not invertible. This solution is shown to be a minimum-norm least squares solution (Mueller and Siltanen, 2012), which unfortunately is not guaranteed to be a distribution, i.e., it is not guaranteed to be an element of the probability simplex Δ^{n-1}.

A recent and fairly general proposal is to minimize a loss function $\mathcal{L}$ and use a soft-max operator in order to guarantee that the result is indeed a distribution (Bunse, 2022a), i.e.,

$$\hat{\mathbf{p}} = \mathrm{softmax}\big(\mathbf{l}^*\big) \in \Delta^{n-1} \tag{8}$$

where

$$\mathbf{l}^* = \operatorname*{arg\,min}_{\mathbf{l} \in \mathbb{R}^n} \mathcal{L}\big(\,\mathrm{softmax}(\mathbf{l}); \mathbf{M}, \mathbf{q}\big) \tag{9}$$

is a vector of latent quantities and where the i-th output of the soft-max operator in Equation 9 is $\mathrm{softmax}_i(\mathbf{l}) = \exp(\mathbf{l}_i)/(\sum_{j=1}^{n} \exp(\mathbf{l}_j))$. Due to the soft-max operator, these latent quantities lend themselves to be interpreted as (translated) log-probabilities. In our implementation, we establish the uniqueness of $\mathbf{l}^*$ by fixing the first dimension to $\mathbf{l}_1 = 0$, which reduces the minimization of $\mathcal{L}$ to $(n-1)$ dimensions without sacrificing the optimality of $\mathbf{l}^*$.

What remains to be detailed in the following subsections are the different choices of loss functions $\mathcal{L}$ and feature transformations f that the different multi-class quantification methods employ.

4.2 Non-ordinal quantification methods

In the following, we introduce some important multi-class quantification methods which do not take ordinality into account. These methods provide the foundation for their ordinal extensions, which we develop in Section 5.

4.2.1 Classify and Count and its adjusted and/or probabilistic variants

The basic **Classify and Count (CC)** method (Forman, 2005) employs a "hard" classifier $h : \mathcal{X} \to \mathcal{Y}$ to generate class predictions for all data items $\mathbf{x} \in \sigma$. The fraction of predictions for a given class is directly used as its prevalence estimate, i.e.,

$$\hat{p}_\sigma^{\mathrm{CC}}(y_i) = \frac{1}{|\sigma|} \cdot \big|\{\mathbf{x} \in \sigma : h(\mathbf{x}) = y_i\}\big| \tag{10}$$

In the probabilistic variant of CC, called **Probabilistic Classify and Count (PCC)** by Bella et al. (2010), the hard classifier is replaced by a "soft" classifier $s : \mathcal{X} \to \Delta^{n-1}$ (with Δ^{n-1} the probability simplex from Equation 1) that returns a vector of (ideally well-calibrated) posterior probabilities $s_i(\mathbf{x}) \equiv \Pr(y_i|\mathbf{x})$, i.e.,

$$\hat{p}_\sigma^{\mathrm{PCC}}(y_i) = \frac{1}{|\sigma|} \cdot \sum_{\mathbf{x} \in \sigma} s_i(\mathbf{x}) \tag{11}$$

CC and PCC are two simplistic quantification methods, which do not attempt to solve Equation 7 for $\mathbf{p}$ and, hence, are biased towards the class distribution of the training set. Despite this inadequacy, these two methods are often used by practitioners, usually due to unawareness of the existence of more suitable quantification methods.

Adjusted Classify and Count (ACC) by Forman (2005) and **Probabilistic Adjusted Classify and Count (PACC)** by Bella et al. (2010) are based on the idea of applying a correction to the estimates $\hat{\mathbf{p}}_\sigma^{\mathrm{CC}}$ and $\hat{\mathbf{p}}_\sigma^{\mathrm{PCC}}$, respectively. These two methods estimate the (hard or soft, respectively) misclassification rates of the classifier on a validation set V; the correction of the estimates $\hat{\mathbf{p}}_\sigma^{\mathrm{CC}}$ and $\hat{\mathbf{p}}_\sigma^{\mathrm{PCC}}$ is then obtained by solving Equation 7 for $\mathbf{p}$, where $\mathbf{q} = (\hat{p}_\sigma(y_1), \ldots, \hat{p}_\sigma(y_n))$ is the distribution as estimated by CC or by PCC, respectively (see Equations 10 and 11), and where

$$\mathbf{M}_{ij} = \frac{1}{|V_i|} \cdot \big|\{\mathbf{x} \in V_i : h(\mathbf{x}) = y_i\}\big| \tag{12}$$

in the case of ACC, or where

$$\mathbf{M}_{ij} = \frac{1}{|V_i|} \cdot \sum_{\mathbf{x} \in V_i} s_i(\mathbf{x}) \tag{13}$$

in the case of PACC, and where V_i is the set of validation data items that belong to class y_i; see Equation 5. In other words, the feature transformation $f(\mathbf{x})$ of ACC is a one-hot encoding of hard classifier predictions $h(\mathbf{x})$, and the feature transformation $f(\mathbf{x})$ of PACC is the output $s(\mathbf{x})$ of a soft classifier (Bunse, 2022b; Firat, 2016).

Both ACC and PACC use a least-squares loss

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}) = \|\mathbf{q} - \mathbf{M}\mathbf{p}\|_2^2 \tag{14}$$

to solve Equation 7 for $\mathbf{p}$ (Bunse, 2022a). We implement this solution as a minimization in terms of Equation 8.

4.2.2 The HDx and HDy distribution-matching methods

For other choices of feature transformations and loss functions, we obtain other quantification algorithms. Two other popular and non-ordinal quantification algorithms are HDx and HDy (González-Castro et al., 2013), which compute feature-wise (HDx) or class-wise (HDy) histograms and minimize the average Hellinger distance across all histograms.

Let d be the number of histograms and let b be the number of bins in each histogram. To ease our notation, we now describe $\mathbf{q} \in \mathbb{R}^{d \times b}$ and $\mathbf{M} \in \mathbb{R}^{d \times b \times n}$ as tensors. Note, however, that a simple concatenation

$$(\mathbf{q}_{11}, \mathbf{q}_{12}, \ldots, \mathbf{q}_{1b}, \mathbf{q}_{21}, \ldots, \mathbf{q}_{db}) \in \mathbb{R}^{db}$$
$$(\mathbf{M}_{11\bullet}, \mathbf{M}_{12\bullet}, \ldots, \mathbf{M}_{1b\bullet}, \mathbf{M}_{21\bullet}, \ldots, \mathbf{M}_{db\bullet}) \in \mathbb{R}^{db \times n}$$

yields again Equation 7, the system of linear equations that uses vectors and matrices instead of tensor notation.

The **HDx** algorithm computes one histogram for each feature in σ, i.e.,

$$\mathbf{q}_{ij} = \frac{1}{|\sigma|} \cdot \big|\{\mathbf{x} \in \sigma \;:\; b_i(\mathbf{x}) = j\}\big| \tag{15}$$

where $b_i(\mathbf{x}) : \mathcal{X} \to \{1, \ldots, b\}$ returns the bin of the i-th feature of $\mathbf{x}$. Accordingly, the tensor $\mathbf{M}$ counts how often each bin of each histogram co-occurs with each class, i.e.,

$$\mathbf{M}_{ijk} = \frac{1}{|V_k|} \cdot \big|\{\mathbf{x} \in V_k : b_i(\mathbf{x}) = j\}\big| \tag{16}$$

As a loss function, HDx employs the average of all feature-wise Hellinger distances, i.e.,

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}) \;=\; \frac{1}{d} \sum_{i=1}^{d} \mathrm{HD}(\mathbf{q}_{i\bullet},\, \mathbf{M}_{i\bullet\bullet}\mathbf{p}) \tag{17}$$

where

$$\mathrm{HD}(\mathbf{a},\,\mathbf{b}) \;=\; \sqrt{\sum_{i=1}^{b}\left(\sqrt{\mathbf{a}_i}-\sqrt{\mathbf{b}_i}\right)^2} \tag{18}$$

is the Hellinger distance between two histograms of a feature.

The **HDy** algorithm uses the same loss function, but operates on the output of a "soft" classifier $s : \mathcal{X} \to \Delta^{n-1}$, as if this output was the original feature representation of the data. Hence, we have

$$\begin{aligned}\mathbf{q}_{ij} &= \frac{1}{|\sigma|}\cdot\left|\{\mathbf{x}\in\sigma \;:\; b_i(s(\mathbf{x}))=j\}\right| \\ \mathbf{M}_{ijk} &= \frac{1}{|V_k|}\cdot\left|\{\mathbf{x}\in V_k : b_i(s(\mathbf{x}))=j\}\right|\end{aligned} \tag{19}$$

where s is a soft classifier that returns posterior probabilities $s_i(\mathbf{x}) \equiv \Pr(y_i|\mathbf{x})$ (or some monotonous transformation thereof). Like ACC and PACC, we implement HDx and HDy as a minimization in terms of Equation 8.

4.2.3 The Saerens-Latinne-Decaestecker EM-based method (SLD)

The **Saerens-Latinne-Decaestecker (SLD)** method (Saerens et al., 2002), also known as "EM-based quantification", follows an iterative expectation maximization approach, which (i) leverages Bayes' theorem in the E-step, and (ii) updates the prevalence estimates in the M-step. Both steps can be combined in the single update rule

$$\hat{p}_\sigma^{(k)}(y_i) = \frac{1}{|\sigma|}\sum_{\mathbf{x}\in\sigma}\frac{\frac{\hat{p}_\sigma^{(k-1)}(y_i)}{\hat{p}_\sigma^{(0)}(y_i)}\cdot s_i(\mathbf{x})}{\sum_{j=1}^{n}\frac{\hat{p}_\sigma^{(k-1)}(y_j)}{\hat{p}_\sigma^{(0)}(y_j)}\cdot s_j(\mathbf{x})} \tag{20}$$

which is applied until the estimates converge. Here, the "(k)" superscript indicates the k-th iteration of the process and $p_\sigma^{(0)}(y)$ is initialized with the class prevalence values of the training set.

4.3 Ordinal quantification methods from the data mining literature

In this section and in Section 4.4 we describe existing ordinal *quantification* methods. While this section describes methods that had been proposed in the data mining / machine learning / NLP literature, and that their proposers indeed call "quantification" methods, Section 4.4 describes methods that were introduced in the physics literature, and that their proposers call "unfolding" methods.

4.3.1 Ordinal Quantification Tree (OQT)

The OQT algorithm (Da San Martino et al., 2016) trains a quantifier by arranging probabilistic binary classifiers (one for each possible bipartition of the ordered set of classes) into an *ordinal quantification tree* (OQT), which is conceptually similar to a hierarchical classifier. Two characteristic aspects of training an OQT are that (a) the loss function used for splitting a node is a quantification loss (and not a classification loss), e.g., the Kullback-Leibler Divergence, and (b) the splitting criterion is informed by the class order. Given a test data item, one generates a posterior probability for each of the classes by having the data item descend all branches of the trained tree. After the posteriors of all data items in the test sample have been estimated this way, PCC is invoked in order to compute the final prevalence estimates.

The OQT method was only tested in the SemEval 2016 "Sentiment analysis in Twitter" shared task (Nakov et al., 2016). While OQT was the best performer in that sub-task, its true value still has to be assessed, since the above-mentioned sub-task evaluated participating algorithms on one test sample only. In our experiments, we test OQT in a much more robust way. Since PCC (the final step of OQT) is known to be biased, we do not expect OQT to exhibit competitive performances.

4.3.2 Adjusted Regress and Count (ARC)

The ARC algorithm (Esuli, 2016) is similar to OQT in that it trains a hierarchical classifier where (a) the leaves of the tree are the classes, (b) these leaves are ordered left-to-right, and (c) each internal node partitions an ordered sequence of classes in two such sub-sequences. One difference between OQT and ARC is the criterion used in order to decide where to split a given sequence of classes, which for OQT is based on a quantification loss (KLD), and for ARC is based on the principle of minimizing the imbalance (in terms of the number of training examples) of the two sub-sequences. A second difference is that, once the tree is trained and used to classify the test data items, OQT uses PCC, while ARC uses ACC.

Concerning the quality of ARC, the same considerations made for OQT apply, since ARC, like OQT, has only been tested in the Ordinal Quantification sub-task of the SemEval 2016 "Sentiment analysis in Twitter" shared task (Nakov et al., 2016); despite the fact that it worked well in that context, the experiments that we present here are more conclusive.

4.3.3 The Match Distance in the EDy method

Castaño et al. (2024) have recently proposed EDy, a variant of the EDx method (Kawakubo et al., 2016) which employs the MD from Equation 3 to measure the distance between soft predictions $s(\mathbf{x})$. Since MD addresses the order of classes, we regard EDy as a true OQ method.

The underlying idea of EDy, following the idea of EDx, is to choose the estimate $\mathbf{p}$ such that the energy distance between $\mathbf{q}$ and $\mathbf{Mp}$ is minimal. This distance can be written as

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}) \;=\; 2\mathbf{p}^\top \mathbf{q} - \mathbf{p}^\top \mathbf{M}\mathbf{p} \tag{21}$$

where

$$\begin{aligned} \mathbf{q}_i \;&=\; \frac{1}{|\sigma| \cdot |V_i|} \sum_{\mathbf{x}\in\sigma} \sum_{\mathbf{x}'\in V_i} \mathrm{MD}\big(s(\mathbf{x}), s(\mathbf{x}')\big) \\ \mathbf{M}_{ij} \;&=\; \frac{1}{|V_j| \cdot |V_i|} \sum_{\mathbf{x}\in V_j} \sum_{\mathbf{x}'\in V_i} \mathrm{MD}\big(s(\mathbf{x}), s(\mathbf{x}')\big) \end{aligned} \tag{22}$$

describe the average MD between data items of different classes (in case of $\mathbf{M}$) and between data items of σ and individual classes (in case of $\mathbf{q}$). In other words, the feature representation of the MD-based variant of EDy is

$$f_i(\mathbf{x}) \;=\; \frac{1}{|V_i|} \sum_{\mathbf{x}'\in V_i} \mathrm{MD}\big(s(\mathbf{x}), s(\mathbf{x}')\big) \tag{23}$$

Alternatively, the distance between samples could be measured in other ways than $\mathrm{MD}(s(\mathbf{x}), s(\mathbf{x}'))$, e.g., in terms of the Euclidean distance $\|\mathbf{x}-\mathbf{x}'\|_2$. However, with the MD being a suitable measure for ordinal problems, we regard Equation 21 as the best fitting and most promising variant of EDx and EDy. In experiments with ordinal data, this variant is recently shown to exhibit state-of-the-art performances (Castaño et al., 2024).

4.3.4 The Match Distance in the PDF method

Another proposal by Castaño et al. (2024) is PDF, an OQ method that minimizes the MD between two ranking histograms. In this method, a ranking function $r : \mathcal{X} \to \mathbb{R}$ is required. Such a function can be obtained from any multi-class soft-classifier $s : \mathcal{X} \to \Delta^{n-1}$ by taking

$$r(\mathbf{x}) \;=\; \sum_{i=1}^{n} i \cdot s_i(\mathbf{x}) \tag{24}$$

such that $r(\mathbf{x})$ is a real value between 1 and n and such that any value $r(\mathbf{x}) \in [i - \frac{1}{2}, i + \frac{1}{2})$ can be interpreted as a prediction for class i.

Having a ranking function, we can compute a one-dimensional histogram of the ranking values of σ and another one-dimensional histogram of the ranking values of the training set, weighted by an estimate $\mathbf{p}$. Castaño et al. (2024) choose $\mathbf{p}$ such that it minimizes the MD between these two histograms, i.e.,

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}) \;=\; \mathrm{MD}(\mathbf{q}, \mathbf{Mp}) \tag{25}$$

where

$$\begin{aligned} \mathbf{q}_i &= \frac{1}{|\sigma|} \cdot \big|\{\mathbf{x} \in \sigma \ : \ b(r(\mathbf{x})) = i\}\big| \\ \mathbf{M}_{ij} &= \frac{1}{|V_j|} \cdot \big|\{\mathbf{x} \in V_j : b(r(\mathbf{x})) = i\}\big| \end{aligned} \tag{26}$$

and where $b(\mathbf{x}) : \mathcal{X} \to \{1, 2, \ldots, B\}$ returns the bin index of $r(\mathbf{x})$. In other words, the feature transformation of PDF is a one-hot encoding of $b(r(\mathbf{x}))$.

4.4 Ordinal quantification methods from the physics literature

Similar to some of the methods discussed in Sections 4.2 and 4.3, experimental physicists have proposed additional adjustments that solve, for $\mathbf{p}$, the system of linear equations from Equation 7. These "unfolding" methods have two particular aspects in common.

The first aspect is that the feature transformation f is assumed to be a partition $c : \mathcal{X} \to \{1, \ldots, t\}$ of the feature space, and

$$\mathbf{q}_i = \frac{1}{|\sigma|} \cdot \big|\{\mathbf{x} \in \sigma : c(\mathbf{x}) = i\}\big| \tag{27}$$

$$\mathbf{M}_{ij} = \frac{1}{|V_j|} \cdot \big|\{\mathbf{x} \in V_j : c(\mathbf{x}) = i\}\big| \tag{28}$$

with $\mathbf{M} \in \mathbb{R}^{t \times n}$; here, i indexes the representation for the ith partition in $\mathbf{q}$ and $\mathbf{M}$, while j indexes the class being modeled in $\mathbf{M}$. In other words, these methods were defined without supervised learning in mind, which differentiates them from all the methods introduced in the previous sections. However, note that, once we replace partition c with a trained classifier h, Equations 27 and 28 become exactly Equations 10 and 12, which define the ACC method.

Another possible choice for c is to partition the feature space by means of a decision tree; in this case, (i) it typically holds that $t > n$, and (ii) $c(\mathbf{x})$ represents the index of a leaf node (Börner et al., 2017). Here, we choose $c = h$ (i.e., we plug in supervised learning) for performance reasons and for establishing a high degree of comparability between quantification methods.

The second aspect of "unfolding" quantifiers, which is central to our work, is the use of a regularization component that promotes what we have called (see Section 3.3) "ordinally plausible" solutions. Specifically, these methods employ the assumption that ordinal distributions are smooth (in the sense of Section 3.3); depending on the algorithm, this assumption is encoded in different ways, as we will see in the following paragraphs.

4.4.1 Regularized Unfolding (RUN)

Regularized Unfolding (RUN) (Blobel, 1985, 2002) has been used by physicists for decades (Aartsen et al., 2017; Nöthe et al., 2017). Here, the

loss function $\mathcal{L}$ consists of two terms, a negative log-likelihood term to model the error of $\mathbf{p}$ and a regularization term to model the plausibility of $\mathbf{p}$.

The negative log-likelihood term in $\mathcal{L}$ builds on a Poisson assumption about the distribution of the data. Namely, this term models the counts $\bar{\mathbf{q}}_i = |\sigma| \cdot \mathbf{q}_i$, which are observed in the sample σ, as being Poisson-distributed with the rates $\lambda_i = \mathbf{M}_{i\bullet}^\top \bar{\mathbf{p}}$. Here, $\bar{\mathbf{p}}_i = |\sigma| \cdot \mathbf{p}_i$ are the class counts that would be observed under a prevalence estimate $\mathbf{p}$.

The second term of $\mathcal{L}$ is a Tikhonov regularization term $\frac{1}{2}\left(\mathbf{C}_1\mathbf{p}\right)^2$, where

$$\mathbf{C}_1 = \begin{pmatrix} -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 & -1 \end{pmatrix} \in \mathbb{R}^{(n-2)\times n} \tag{29}$$

This term introduces an inductive bias towards smooth solutions, i.e., solutions which are (following the assumption we have made in Section 3.3) ordinally plausible. The choice of the Tikhonov matrix $\mathbf{C}_1$ ensures that $\frac{1}{2}\left(\mathbf{C}_1\mathbf{p}\right)^2$ measures the jaggedness of $\mathbf{p}$, i.e.,

$$\frac{1}{2}\left(\mathbf{C}_1\mathbf{p}\right)^2 = \frac{1}{2}\sum_{i=2}^{n-1}\left(-\mathbf{p}_{i-1} + 2\mathbf{p}_i - \mathbf{p}_{i+1}\right)^2 \tag{30}$$

which only differs from $\xi_1(\mathbf{p}_\sigma)$, our measure of ordinal plausibility from Equation 4, in terms of a constant normalization factor.[4] (Indeed, subscript "1" in $\mathbf{C}_1$ is there to indicate that the goal of $\mathbf{C}_1$ is to minimize $\xi_1(\mathbf{p}_\sigma)$.) Combining the likelihood term and the regularization term, the loss function of RUN is

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}, \tau) \;=\; \sum_{i=1}^{t}\left(\mathbf{M}_{i\bullet}^\top \bar{\mathbf{p}} - \bar{\mathbf{q}}_i \cdot \ln(\mathbf{M}_{i\bullet}^\top \bar{\mathbf{p}})\right) \;+\; \frac{\tau}{2}\left(\mathbf{C}_1\mathbf{p}\right)^2 \tag{31}$$

and an estimate $\hat{\mathbf{p}}$ can be chosen in terms of Equation 8. Here, $\tau \geq 0$ is a hyper-parameter which controls the impact of the regularization.

4.4.2 Iterative Bayesian Unfolding (IBU)

Iterative Bayesian Unfolding (IBU) by D'Agostini (1995, 2010) is still popular today (Aad et al., 2021; Nachman et al., 2020). This method revolves around an expectation maximization approach with Bayes' theorem, and thus has a common foundation with the SLD method. The E-step and the M-step of IBU can be written as the single, combined update rule

$$\hat{p}_\sigma^{(k)}(y_i) = \sum_{j=1}^{t} \frac{\mathbf{M}_{ij} \cdot \hat{p}_\sigma^{(k-1)}(y_i)}{\sum_{l=1}^{n} \mathbf{M}_{lj} \cdot \hat{p}_\sigma^{(k-1)}(y_l)}\, \mathbf{q}_i \tag{32}$$

[4] The factor $\frac{1}{2}$ is a convention in the regularization literature, motivated by the fact that this factor yields $\mathbf{Cp}$ as the first derivative of the regularization term, an outcome that facilitates theoretical analyses of regularization. For our purposes the normalization factor has no impact.

One difference between IBU and SLD is that $\mathbf{q}$ and $\mathbf{M}$ are defined via counts of hard assignments to partitions $c(\mathbf{x})$ (see Equation 27), while SLD is defined over individual soft predictions $s(\mathbf{x})$ (see Equation 20).

Another difference between IBU and SLD is regularization. In order to promote solutions which are ordinally plausible, IBU smooths each intermediate estimate $\hat{\mathbf{p}}^{(k)}$ by fitting a low-order polynomial to $\hat{\mathbf{p}}^{(k)}$. A linear interpolation between $\hat{\mathbf{p}}^{(k)}$ and this polynomial is then used as the prior of the next iteration in order to reduce the differences between neighboring prevalence estimates. The order of the polynomial and the interpolation factor are hyper-parameters of IBU through which the regularization is controlled.

4.4.3 Other quantification methods from the physics literature

Other methods from the physics literature that perform what we here call "quantification" go under the name of "unfolding" methods, and are based on similar concepts as RUN and IBU. We focus on RUN and IBU due to their long-standing popularity within physics research. In fact, they are among the first methods that have been proposed in this field, and are still widely adopted today, in astro-particle physics (Aartsen et al., 2017; Nöthe et al., 2017), high-energy physics (Aad et al., 2021), and more recently in quantum computing (Nachman et al., 2020). Moreover, RUN and IBU already cover the most important aspects of unfolding methods with respect to OQ.

Several other unfolding methods are similar to RUN. For instance, the method proposed by Hoecker and Kartvelishvili (1996) employs the same regularization as RUN, but assumes different Poisson rates, which are simplifications of the rates that RUN uses; in preliminary experiments, here omitted for the sake of conciseness, we have found this simplification to typically deliver less accurate results than RUN. Two other methods (Schmelling, 1994; Schmitt, 2012) employ the same simplification as (Hoecker and Kartvelishvili, 1996) but regularize differently. To this end, Schmelling (1994) regularizes with respect to the deviation from a prior, instead of regularizing with respect to ordinal plausibility; we thus do not perceive this method as a true OQ method. Schmitt (2012) adds to the RUN regularization a second term which enforces prevalence estimates that sum up to one; however, implementing RUN in terms of Equation 8 already solves this issue. Another line of work evolves around the algorithm by Ruhe et al. (2013) and its extensions (Bunse et al., 2018). We perceive this algorithm to lie outside the scope of OQ because it does not address the order of classes, like the other "unfolding" methods do. Moreover, the algorithm was shown to exhibit a performance comparable to, but not better than RUN and IBU (Bunse et al., 2018).

5 New ordinal versions of multi-class quantification algorithms

In the following, we develop algorithms which modify ACC, PACC, HDx, HDy, SLD, EDy, and PDF with the regularizers from RUN and IBU. Through these

modifications, we obtain o-ACC, o-PACC, o-HDx, o-HDy, and o-SLD, the OQ counterparts of these well-known non-ordinal quantification algorithms, as well as o-EDy and o-PDF, which combine ordinal loss functions and feature representations with an ordinal regularizer. In doing so, since we employ the regularizers but not any other aspect of RUN and IBU, we preserve the general characteristics of the original algorithms. In particular, we do not change the feature representations and we maintain the original loss functions of these methods. Therefore, our extensions are "minimal", in the sense of being directly addressed to ordinality, without introducing any undesired side effects in the original methods.

5.1 Tikhonov regularization in multi-class algorithms

The OQ counterparts of most algorithms—ACC, PACC, HDx, HDy, EDy, and PDF—are constructed by defining a novel, OQ-oriented loss function that adds the Tikhonov regularizer from Equation 30 to the original loss function of each algorithm. This ordinal extension is defined through the regularized loss

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}, \tau) \;=\; \mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}) \;+\; \frac{\tau}{2}\,(\mathbf{C}_1\mathbf{p}\,)^2 \tag{33}$$

where $\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q})$ is the original loss function of any existing (not necessarily ordinal) quantification algorithm. The hyper-parameter $\tau \geq 0$ and the Tikhonov matrix $\mathbf{C}_1$ are the ones introduced by physicists to address ordinality in the RUN method of Section 4.4.1. Like before, we minimize Equation 33 with the soft-max operator from Equation 8.

If we apply the above definition of a regularized loss to ACC and PACC (see Section 4.2.1), we obtain o-ACC and o-PACC, the ordinal counterparts of these methods. The respective feature transformation and loss function of ACC and PACC are maintained, such that the only novelty is the regularization term that promotes ordinally plausible solutions.

Similarly, if we apply the above definition to HDx and HDy (see Section 4.2.2), we obtain o-HDx and o-HDy; if we apply the definition to EDy and PDF (see Sections 4.3.3 and 4.3.4), we obtain o-EDy and o-PDF. In all of these cases, the only novelty is the regularization term.

Among the extended methods, o-EDy and o-PDF stand out in the sense that they combine multiple approaches to addressing ordinality. In the case of o-EDy, an ordinal feature transformation (the one of EDy) is combined with an ordinal regularizer (the one of RUN). In the case of o-PDF, an ordinal loss function (the one of PDF) is regularized to further promote solutions that are ordinally plausible. In all other extensions—o-ACC, o-PACC, o-HDx, and o-HDy—the one and only aspect concerning ordinality is the regularizer.

5.2 o-SLD

Unlike the other methods, SLD does not explicitly minimize a loss function. Hence, our ordinal extension o-SLD uses, instead of a Tikhonov regularization term, the ordinal regularization approach of IBU in SLD. Namely, our method does not use the latest estimate directly as the prior of the next iteration, but a smoothed version of this estimate. To this end, we fit a low-order polynomial to each intermediate estimate $\hat{\mathbf{p}}^{(k)}$ and use a linear interpolation between this polynomial and $\hat{\mathbf{p}}^{(k)}$ as the prior of the next iteration. Like in IBU, we consider the order of the polynomial and the interpolation factor as hyper-parameters of o-SLD.

6 Experiments

The goal of our experiments is to uncover the relative merits of OQ methods originating from different fields. We pursue this goal by carrying out a thorough comparison of these methods on representative OQ datasets. In the interest of reproducibility we make all the code publicly available.[5]

6.1 Datasets and pre-processing

We conduct our experiments on two large datasets that we have generated for the purpose of this work, and that we make available to the scientific community. The first dataset, named Amazon-OQ-BK, consists of product reviews labeled according to customers' judgments of quality, ranging from 1Star to 5Stars. The second dataset, Fact-OQ, consists of telescope observations each labeled by one of 12 totally ordered classes. These datasets originate in practically relevant and very diverse applications of OQ.

6.1.1 The data sampling protocol

We start by dividing each data set into a set L of training data items, a pool of validation (i.e., development) data items, and a pool of test data items. These three sets are disjoint from each other, and we obtain each of them through stratified sampling from the original data source. We set the size of the training set to 20,000 data items, use half of the remaining items for the validation pool, and use the other half for the testing pool.

From both the validation pool and the test pool, we separately extract samples (i.e., sets of data items) σ to be predicted during quantifier evaluation. Following Esuli et al. (2022), each sample is generated in two steps. First, we randomly draw a ground-truth vector $\mathbf{p}_\sigma$ of class prevalence values. We realize this step in three different ways that are still to be detailed in the following paragraphs. Second, we draw from the pool of data (be it our validation pool

[5] `https://github.com/mirkobunse/regularized-oq`

or our test pool) a fixed-size sample σ of data items that realizes the class prevalence values of $\mathbf{p}_\sigma$. We set the size of σ to 1,000 data items. For validation, we draw 1,000 such samples and for testing, we draw 5,000 samples. All data items in a pool are replaced after the generation of each sample, where our initial split into a training set, a validation pool, and a test pool already ensures that each validation sample is disjoint from each test sample and that the training set is disjoint from all other samples.

Through the above approach, we can predict the prevalence values of each σ through quantification methods and compare the outcomes with the ground-truth vector $\mathbf{p}_\sigma$. By drawing many $\mathbf{p}_\sigma$ at random, we can test the quantification methods in many different instances of prior probability shift.

Real prevalence vectors The most realistic way of drawing $\mathbf{p}_\sigma$ is to draw it uniformly at random from the set of those prevalence vectors that are exhibited by samples that naturally occur in the data. We call these vectors *real* prevalence vectors due to their natural occurrence.

For Amazon-OQ-BK (to be detailed in Section 6.1.2), each natural sample consists of all reviews that address one individual *product*. Hence, each $\mathbf{p}_\sigma$ corresponds to the prevalence of customer ratings for a single product. For Fact-OQ (to be detailed in Section 6.1.3), each natural sample consists of telescope observations that are distributed according to a parametrization of the Crab Nebula (Aleksić et al., 2015) and are thus representative of data that physicists expect to handle in practice.

While real prevalence vectors provide the most realistic (and therefore the most sensible) setting for quantifier evaluation, they also bear two shortcomings. First, they are not available for standard classification data sets, preventing these sets from being used for quantifier evaluation with real prevalence vectors. Due to this reason, we make available Amazon-OQ-BK and Fact-OQ as actual quantification data sets with real prevalence vectors. Second, since the distribution of real prevalence vectors differs between data sets, quantifiers cannot easily be compared across multiple data sets. Due to these shortcomings, we evaluate not only in terms of real prevalence vectors, but also in terms of two other evaluation protocols.

Artificial Prevalence Protocol (APP) Perhaps the most common way of drawing $\mathbf{p}_\sigma$ is to draw it uniformly at random from Δ^{n-1}, the set of all possible prevalence vectors (Forman, 2005).

By picking all prevalence vectors with the same probability and without any dependence on the data, APP allows us to compare performance across multiple datasets. Moreover, it is capable of re-purposing any standard classification data set for the evaluation of quantifiers and it demands a high performance of quantification methods throughout Δ^{n-1}, which is another desirable property. However, this demand is without any consideration of whether some $\mathbf{p}_\sigma$ is realistic or "ordinally plausible", in the sense of Section 3.3. Therefore, APP

Table 1 Characteristics of ground-truth class prevalence distributions $\mathbf{p}_\sigma$, which are sampled through different protocols and for both datasets. We consider the average jaggedness $\xi_1(\mathbf{p}_\sigma)$, as according to Equation 4, and the average amount of prior probability shift $\text{NMD}(\mathbf{p}_L, \mathbf{p}_\sigma)$, as according to Equation 2, of $\mathbf{p}_\sigma$. The row "real prevalence vectors" reports these values for naturally occurring samples in our datasets. Values in **boldface** indicate the protocols that we employ in our experiments.

protocol	AMAZON-OQ-BK		FACT-OQ	
	$\xi_1(\mathbf{p}_\sigma)$	$\text{NMD}(\mathbf{p}_L, \mathbf{p}_\sigma,)$	$\xi_1(\mathbf{p}_\sigma)$	$\text{NMD}(\mathbf{p}_L, \mathbf{p}_\sigma,)$
real prevalence vectors	**.0372**	**.1385**	**.0125**	**.1297**
APP	**.0995**	**.2817**	**.0641**	**.2411**
APP-OQ (66%)	.0452	.2786	.0403	.2420
APP-OQ (50%)	**.0330**	**.2775**	.0335	.2425
APP-OQ (33%)	.0221	.2773	.0266	.2430
APP-OQ (20%)	.0145	.2774	.0211	.2433
APP-OQ (5%)	.0054	.2780	**.0124**	**.2469**

has a tendency of over-emphasizing performance in regions of Δ^{n-1} which are unlikely to ever appear in practice.

APP-OQ for ordinal plausibility Since we take smoothness (in the sense of Section 3.3) as a criterion for ordinal plausibility, we counteract this shortcoming of APP by further devising APP-OQ(x%), a protocol similar to APP but for the fact that only the x% smoothest samples are retained. Hence, when evaluating a quantifier, we perform hyper-parameter optimization on the x% smoothest validation samples and test on the x% smoothest test samples generated by APP.

To use the above approach, we need to decide on a percentage x to use. To make this choice, we characterize the $\mathbf{p}_\sigma$ that result from different choices of x in terms of their average jaggedness $\xi_1(\mathbf{p}_\sigma)$ and in terms of the average amount of prior probability shift $\text{NMD}(\mathbf{p}_L, \mathbf{p}_\sigma)$ that they generate. We compare these characteristics with those of the real prevalence vectors and choose the value of x that yields the most realistic values of $\xi_1(\mathbf{p}_\sigma)$.

The results from Table 1 convey that APP-OQ, while becoming smoother with smaller values of x, produces *constant* amounts of prior probability shift. In this sense, the quantification tasks of APP-OQ become more ordinally plausible but not simpler. Hence, APP-OQ retains the beneficial coverage of Δ^{n-1} that APP exhibits. The most suitable percentage for AMAZON-OQ-BK turns out to be 50% while the percentage for FACT-OQ turns out to be 5%. This difference stems from the smoother distributions that FACT-OQ exhibits in the real prevalence vectors.

In a nutshell, each of the above protocols provides a different perspective, which we combine by always reporting the results of all three protocols side by side. Real prevalence vectors provide the most realistic evaluation but do not allow to compare performance across multiple data sets, APP is the most common approach that provides a bridge to previous works, and APP-OQ

seeks to balance these two perspectives for ordinal quantification in particular.

6.1.2 The Amazon-OQ-BK *dataset*

We make available the Amazon-OQ-BK dataset[6], which we extract from an existing dataset by McAuley et al. (2015), consisting of 233.1M English-language Amazon product reviews;[7] here, a data item corresponds to a single product review. As the labels of the reviews, we use their "stars" ratings, and our code frame is thus $\mathcal{Y} = \{\mathsf{1Star}, \mathsf{2Stars}, \mathsf{3Stars}, \mathsf{4Stars}, \mathsf{5Stars}\}$, which represents a sentiment quantification task (Esuli and Sebastiani, 2010).

The reviews are subdivided into 28 product categories, including "Automotive", "Baby", "Beauty", etc. We restrict our attention to reviews from the "Books" product category, since it is the one with the highest number of reviews. We then remove (a) all reviews shorter than 200 characters because recognizing sentiment from shorter reviews may be nearly impossible in some cases, and (b) all reviews that have not been recognized as "useful" by any users because many reviews never recognized as "useful" may contain comments, say, on Amazon's speed of delivery, and not on the product itself.

We convert the reviews into vectors by using the RoBERTa transformer (Liu et al., 2019) from the Hugging Face hub.[8] To this aim, we truncate the reviews to the first 256 tokens and fine-tune RoBERTa via prompt learning for a maximum of 5 epochs on our training data, using the model parameters from the epoch with the smallest validation loss monitored on 1000 held-out reviews randomly sampled from the training set in a stratified way. For training, we set the learning rate to $2e^{-5}$, the weight decay to 0.01, and the batch size to 16, leaving the other hyper-parameters at their default values. For each review, we generate features by first applying a forward pass over the fine-tuned network, and then averaging the embeddings produced for the special token [CLS] across all the 12 layers of RoBERTa. In our initial experiments, this latter approach yielded slightly better results than using the [CLS] embedding of the last layer alone. The embedding size of RoBERTa, and hence the number of dimensions of our vectors, amounts to 768.

6.1.3 The Fact-OQ *dataset*

We extract our second dataset, called Fact-OQ,[9] from the open dataset of the FACT telescope (Anderhub et al., 2013);[10] here, a data item corresponds to a single telescope recording. We represent each data item in terms of the 20 dense features that are extracted by the standard processing pipeline[11] of the

[6] https://zenodo.org/record/8405476 (v0.2.1)
[7] http://jmcauley.ucsd.edu/data/amazon/links.html
[8] https://huggingface.co/docs/transformers/model_doc/roberta
[9] https://zenodo.org/record/8172813 (v0.2.0)
[10] https://factdata.app.tu-dortmund.de/
[11] https://github.com/fact-project/open_crab_sample_analysis/

telescope. Each of the 1,851,297 recordings is labeled with the energy of the corresponding astro-particle, and our goal is to estimate the distribution of these energy labels via OQ. While the energy labels are originally continuous, astro-particle physicists have established a common practice of dividing the range of energy values into ordinal classes, as argued in Section 4.4. Based on discussions with astro-particle physicists, we divide the range of continuous energy values into an ordered set of 12 classes. As a result, our quantifiers predict histograms of the energy distribution that have 12 equal-width bins.

Note that, since we are using NMD as our evaluation measure, we can meaningfully compare the results we obtain on Amazon-OQ-BK (which uses a 5-class code frame) with the results we obtain on Fact-OQ (which uses a 12-class code frame); this would not have been possible if we had used MD, which is not normalized by the number of classes in the code frame.

6.1.4 The UCI and OpenML datasets

Additionally to our experiments on Amazon-OQ-BK and Fact-OQ, we also carry out experiments on a collection of public datasets from the UCI repository[12] and OpenML.[13] To identify these datasets, we first select all regression datasets (i.e., datasets consisting of data items labeled by real numbers) in UCI or OpenML that contain at least 30,000 data items. We then try to apply equal-width binning (i.e., bin the data according to their label by constraining the resulting bins to span equal-width intervals of the x axis) to each such dataset, in such a way that the binning process produces 10 bins (which we view as ordered classes) of at least 1000 data items each. We only retain the datasets for which such a binning is possible. In these cases, in order to retain as many samples as possible, we maximize the distance between the leftmost and rightmost boundaries of each bin (which implies, among other things, using *exactly* 10 bins). We also remove all the data items that lie outside the 10 equidistant bins. From this protocol, we obtain the 4 datasets UCI-blog-feedback-OQ, UCI-online-news-popularity-OQ, OpenML-Yolanda-OQ, and OpenML-fried-OQ, which we make publicly available.[14]

We present the results obtained on these datasets in Appendix B.2. The reason why we confine these results to an appendix is that, unlike Amazon-OQ-BK and Fact-OQ, the data of which these datasets consist are not "naturally ordinal". In other words, in order to create these datasets we use data that were originally labeled by real numbers (i.e., data suitable for metric regression experiments), bin them by their label, and view the resulting bins as ordinal classes. The ordinal nature of these datasets is thus somehow questionable, and we thus prefer not to consider them as being on a par with Amazon-OQ-BK and Fact-OQ, which instead originate from data that its users actually treat as being ordinal.

[12] `https://archive.ics.uci.edu/ml/index.php`

[13] `https://www.openml.org/`

[14] `https://zenodo.org/record/8177302` (v0.2.0)

Table 2 Performance of classifiers in terms of average NMD (lower is better) in the AMAZON-OQ-BK dataset for the APP-OQ protocol. **Boldface** indicates the best classifier for each quantification method, or a classifier not significantly different from the best one in terms of a paired Wilcoxon signed-rank test at a confidence level of $p = 0.01$. For LR we present standard deviations, while for all other classifiers we show the average deterioration in NMD with respect to LR. PCC, PACC, and SLD require a soft classifier, which means that ORidge and LAD cannot be embedded in these methods.

	CC	PCC	ACC	PACC	SLD
LR	**0.0404** ±0.0134	**0.0502** ±0.0167	0.0224 ±0.0084	**0.0187** ±0.0072	**0.0163** ±0.0062
OLR-AT	0.0424 (+5.0%)	0.0526 (+4.9%)	**0.0218** (-2.7%)	0.0203 (+8.2%)	0.0216 (+32.8%)
OLR-IT	0.0412 (+2.0%)	0.0548 (+9.3%)	0.0230 (+5.4%)	0.0199 (+6.3%)	0.0679 (+316.2%)
ORidge	0.0472 (+16.9%)	—	**0.0221** (+1.2%)	—	—
LAD	**0.0408** (+1.0%)	—	0.0229 (+4.6%)	—	—

6.2 Results: Non-ordinal quantification methods with ordinal classifiers

In our first experiment, we investigate whether OQ can be solved by non-ordinal quantification methods built on top of ordinal classifiers. To this end, we compare the use of a standard multi-class logistic regression (LR) with the use of several ordinal variants of LR. In general, we have found that LR models, trained on the deep RoBERTa embedding of the AMAZON-OQ-BK dataset, are extremely powerful models in terms of quantification performance. Therefore, approaching OQ with ordinal LR variants embedded in non-ordinal quantifiers could be a straightforward solution worth investigating.

The ordinal LR variants we test are the "All Threshold" variant (OLR-AT) and the "Immediate-Threshold variant" (OLR-IT) of (Rennie and Srebro, 2005). In addition, we try two ordinal classification methods based on discretizing the outputs generated by regression models (Pedregosa et al., 2017); the first is based on *Ridge Regression* (ORidge) while the second, called *Least Absolute Deviation* (LAD), is based on linear SVMs.

Table 2 reports the results of this experiment, using the non-ordinal quantifiers of Section 4.2 and following the APP-OQ protocol (the results for other protocols were by and large similar and are omitted for conciseness). The fact that the best results are almost always obtained by using, as the embedded classifier, non-ordinal LR shows that, in order to deliver accurate estimates of class prevalence values in the ordinal case, it is not sufficient to equip a multi-class quantifier with an ordinal classifier. Moreover, the fact that PCC obtains worse results when equipped with the ordinal classifiers (OLR-AT and OLR-IT) than when equipped with the non-ordinal one (LR) suggests that the posterior probabilities computed under the ordinal assumption are of lower quality.

Overall, these results suggest that, in order to tackle OQ, we cannot simply rely on ordinal classifiers embedded in non-ordinal quantification methods. Instead, we need proper OQ methods.

6.3 Results: Ordinal quantification methods

In our main experiment, we compare our proposed methods o-ACC, o-PACC, o-HDx, o-HDy, o-SLD, o-EDy, and o-PDF with several baselines, i.e.,

1. the non-ordinal quantification methods CC, PCC, ACC, PACC, HDx, HDy, and SLD (see Section 4.2);
2. the ordinal quantification methods OQT, ARC, EDy, and PDF (see Section 4.3); and
3. the ordinal quantification methods IBU and RUN from the "unfolding" tradition (see Section 4.4).

We compare these methods on the Amazon-OQ-BK and Fact-OQ datasets, using real prevalence vectors and the APP and APP-OQ protocols.

Each method is allowed to tune the hyper-parameters of its embedded classifier, using the samples of the validation set. We use logistic regression on Amazon-OQ-BK and random forests on Fact-OQ; this choice of classifiers is motivated by common practice in the fields where these datasets originate, and from our own experience that these classifiers work well on the respective type of data. To estimate the quantification matrix $\mathbf{M}$ of a logistic regression consistently, we use k-fold cross-validation with $k = 10$, by now a standard procedure in quantification learning (Forman, 2005). Since random forests are capable of producing out-of-bag predictions at virtually no extra cost, they do not require additional hold-out predictions from cross-validation to estimate the generalization errors of the forest (Breiman, 1996). Therefore, we use the out-of-bag predictions of the random forest to estimate $\mathbf{M}$ in a consistent manner, without further cross-validating these classifiers.

After the hyper-parameters of the quantifier, including the hyper-parameters of the classifier, are optimized, we apply each method to the samples of the test set. The results of this experiment are summarized in Tables 3 and 4. These results convey that our proposed methods outperform the competition on both datasets and under all protocols; at least, they perform on par with the competition. In each protocol, o-SLD is the best method on Amazon-OQ-BK while o-PACC and o-SLD are best methods on Fact-OQ.

For all methods, we observe that the ordinally regularized variant is always better than or equal to the original, non-regularized variant of the same method. This observation can also be made with respect to EDy and PDF, the two recent OQ methods that address ordinality through ordinal feature transformations (EDy) and loss functions (PDF). We further recognize that the non-regularized EDy and PDF often loose even against non-ordinal baselines, such as SLD and HDy. From this outcome, we conclude that, in addressing ordinality, regularization is indeed a more important aspect than those feature transformations and loss functions that have been proposed so far.

Regularization even improves performance in the standard APP protocol, where the sampling does not enforce any smoothness. First of all, this finding demonstrates that regularization leads to a performance improvement that cannot be dismissed as a mere byproduct of simply having smooth ground-

Table 3 Average performance in terms of NMD (lower is better) for the Amazon-OQ-BK data. We present the results of the protocols APP, APP-OQ, and real prevalence vectors. The best performance in each column is highlighted in **boldface**. According to a Wilcoxon signed rank test with $p = 0.01$, all other methods are statistically significantly different from the best method.

		APP	APP-OQ	real
non-ordinal baselines	CC	.0534 ± .0183	.0434 ± .0149	.0295 ± .0188
	PCC	.0611 ± .0208	.0496 ± .0168	.0319 ± .0194
	ACC	.0306 ± .0150	.0255 ± .0151	.0166 ± .0087
	PACC	.0289 ± .0108	.0243 ± .0101	.0164 ± .0085
	HDx	.0281 ± .0098	.0248 ± .0091	.0177 ± .0100
	HDy	.0277 ± .0102	.0236 ± .0090	.0168 ± .0100
	SLD	.0217 ± .0099	.0200 ± .0071	**.0145 ± .0064**
ordinal baselines	OQT	.0688 ± .0244	.0563 ± .0194	.0302 ± .0160
	ARC	.0617 ± .0213	.0509 ± .0167	.0251 ± .0141
	IBU	.0311 ± .0114	.0254 ± .0088	.0167 ± .0087
	RUN	.0301 ± .0112	.0248 ± .0090	.0166 ± .0087
	EDy	.0297 ± .0107	.0251 ± .0092	.0174 ± .0080
	PDF	.0303 ± .0110	.0258 ± .0095	.0180 ± .0091
new ordinal methods	o-ACC	.0306 ± .0150	.0255 ± .0151	.0166 ± .0087
	o-PACC	.0289 ± .0108	.0243 ± .0101	.0164 ± .0087
	o-HDx	.0281 ± .0098	.0248 ± .0091	.0176 ± .0095
	o-HDy	.0277 ± .0102	.0236 ± .0090	.0168 ± .0100
	o-SLD	**.0194 ± .0083**	**.0190 ± .0085**	**.0145 ± .0063**
	o-EDy	.0290 ± .0104	.0245 ± .0090	.0174 ± .0080
	o-PDF	.0296 ± .0107	.0252 ± .0097	.0182 ± .0091

Table 4 Same as Table 3 but using Fact-OQ in place of Amazon-OQ-BK.

		APP	APP-OQ	real
non-ordinal baselines	CC	.0401 ± .0108	.0321 ± .0102	.0565 ± .0037
	PCC	.0442 ± .0109	.0388 ± .0115	.0626 ± .0026
	ACC	.0261 ± .0078	.0275 ± .0069	.0201 ± .0101
	PACC	.0216 ± .0070	.0226 ± .0066	.0157 ± .0049
	HDx	.0490 ± .0233	.0439 ± .0110	.0445 ± .0194
	HDy	.0219 ± .0125	.0223 ± .0060	.0172 ± .0101
	SLD	.0192 ± .0052	.0164 ± .0042	.0129 ± .0038
ordinal baselines	OQT	.0484 ± .0132	.0403 ± .0129	.0402 ± .0031
	ARC	.0464 ± .0103	.0422 ± .0107	.0468 ± .0034
	IBU	.0208 ± .0054	.0165 ± .0043	.0134 ± .0037
	RUN	.0226 ± .0057	.0179 ± .0048	.0128 ± .0036
	EDy	.0209 ± .0052	.0180 ± .0043	.0131 ± .0035
	PDF	.0262 ± .0072	.0251 ± .0065	.0171 ± .0053
new ordinal methods	o-ACC	.0225 ± .0057	.0171 ± .0033	.0137 ± .0028
	o-PACC	.0193 ± .0061	**.0136 ± .0031**	**.0096 ± .0029**
	o-HDx	.0348 ± .0238	.0254 ± .0185	.0310 ± .0883
	o-HDy	.0218 ± .0171	.0142 ± .0033	.0105 ± .0029
	o-SLD	**.0187 ± .0049**	.0153 ± .0039	.0120 ± .0033
	o-EDy	.0207 ± .0054	.0155 ± .0040	.0115 ± .0031
	o-PDF	.0230 ± .0060	.0159 ± .0035	.0100 ± .0029

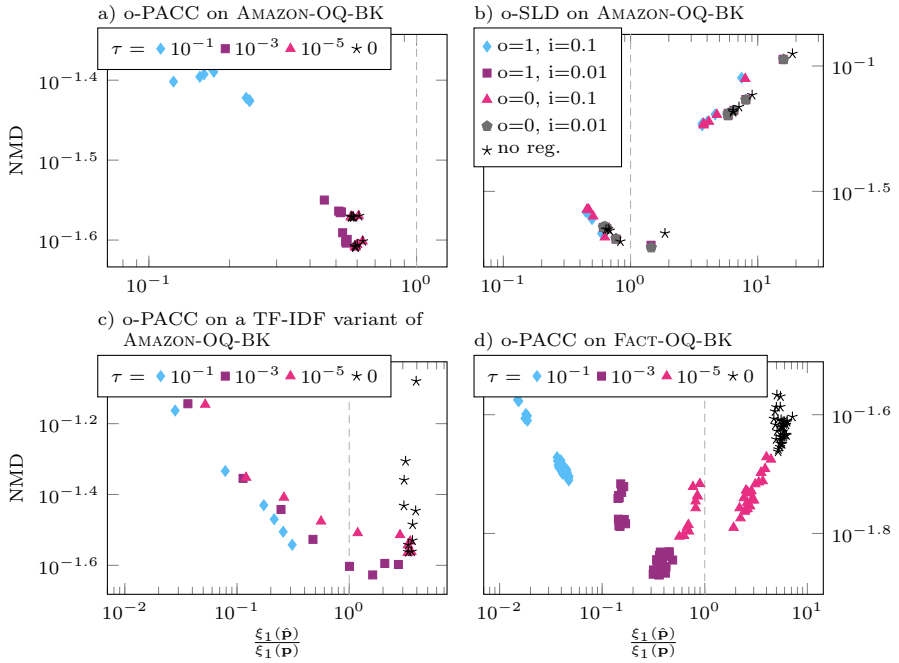

Fig. 4 Each point represents one hyper-parameter combination in the space of the average validation error (y axis) and the average ratio between the jaggedness of the predictions $\hat{\mathbf{p}}$ and the jaggedness of the ground-truth vectors $\mathbf{p}$ (x axis) during APP-OQ. Colors and shapes represent the regularization parameters of the hyper-parameter combinations. Our proposed ordinal regularization is beneficial for configurations that are otherwise too jagged, i.e., for configurations that are located to the right of the vertical line at $\frac{\xi_1(\hat{\mathbf{p}})}{\xi_1(\mathbf{p})} = 1$.

truth prevalence vectors (such as in APP-OQ and with real prevalence vectors). Instead, regularization appears to result in a systematic improvement of OQ predictions. We attribute this outcome to the fact that, even if no smoothness is enforced, neighboring classes are still hard to distinguish in ordinal settings. Therefore, an unregularized quantifier can easily tend to over- or under-estimate one class at the expense of its neighboring class. Regularization, however, effectively controls the difference between neighboring prevalence estimates, thereby protecting quantifiers from a tendency towards the over- or under-estimation of particular classes. This effect persists even if the evaluation protocol, like APP, does not enforce smooth ground-truth prevalence vectors. Hence, the performance improvement due to regularization can be attributed (at least in part) to the similarity between neighboring classes, a ubiquitous phenomenon in ordinal settings.

Experiments carried out on the UCI and OpenML datasets reinforce the above conclusions. We provide these results in the appendix.

6.4 Results: Limitations of ordinal regularization

Table 3 lists several cases in which, if evaluated on the Amazon-OQ-BK data, some of our ordinal variants (e.g., o-ACC, o-PACC, o-HDx, and o-HDy) perform only on par with (and do not outperform) the non-ordinal methods they extend; hence, regularization is not able to improve quantification performance in these particular cases.

The reason for this observation is that our embedding representation of the Amazon-OQ-BK data often leads to predictions that are *already smooth* without any regularization. Due to this smoothness property of the data, any additional smoothing through regularization bears the danger of over-smoothing (i.e., of predictions that tend to be smoother than the ground-truth) which, in turn, can increase the prediction error.

Figure 4 illustrates this issue by plotting the average validation NMD over the average ratio $\frac{\xi_1(\hat{\mathbf{p}})}{\xi_1(\mathbf{p})}$ between the jaggedness of the predictions, $\xi_1(\hat{\mathbf{p}})$, and the jaggedness of the ground-truth vectors, $\xi_1(\mathbf{p})$. Here, ratios smaller than one indicate that the predictions tend to be less jagged than the ground-truth; in other words, they tend to be too smooth and, hence, often exhibit high NMD values. Since regularization adds smoothness to predictions, we expected a benefit in NMD only for those predictions that are otherwise too jagged, with ratios above one. Examples of improvements are o-SLD with the Amazon-OQ-BK data (sub-plot b in Figure 4) or o-PACC with the Fact-OQ-BK data (sub-plot d). However, PACC with Amazon-OQ-BK (sub-plot a) turns out to be already too smooth, even without any regularization. Therefore, adding regularization cannot further decrease the NMD on this data set.

The high smoothness within sub-plot (a) is a consequence of the powerful embedding representation that we employ for the Amazon-OQ-BK data (see Section 6.1.2). To demonstrate this claim, we repeat the same experiment with the same data and the same classifier, but employ a weaker TF-IDF representation instead of the embeddings. As we can see in sub-plot (c), the weaker representation leads again to predictions that are too jagged and, hence, can benefit from regularization. The complete results of the TF-IDF representation can be found in Appendix B.

We conclude that smoothness can not only be achieved through regularization but also through data representations, although methods in the latter direction remain open to future research. Regularization benefits quantification performance only if the predictions are otherwise too jagged, a setting that can be verified by evaluating $\frac{\xi_1(\hat{\mathbf{p}})}{\xi_1(\mathbf{p})}$. Regularization parameters provide a fine-grained control over the smoothness that predictions exhibit.

7 Other notions of smoothness for ordinal distributions

In Section 3.3 we have introduced the notion of "jaggedness" (and that of smoothness, its opposite), and we have proposed the $\xi_1(\mathbf{p}_\sigma)$ function as a measure of how jagged an ordinal distribution $\mathbf{p}_\sigma$ is. We have then proposed

ordinal quantification methods that use a Tikhonov matrix $\mathbf{C}_1$ whose goal is to minimize this measure, as in the regularization term of Equation 30. The key assumption behind $\xi_1(\mathbf{p}_\sigma)$ and $\mathbf{C}_1$ is a key assumption of ordinality: that neighboring classes are similar.

However, note that $\xi_1(\mathbf{p}_\sigma)$ is by no means the only conceivable function for measuring jaggedness, and that other alternatives are possible in principle. For instance, one such alternative might be

$$\xi_0(\mathbf{p}_\sigma) = \frac{1}{2}\sum_{i=1}^{n-1}(p_\sigma(y_i) - p_\sigma(y_{i+1}))^2 \tag{34}$$

where $\frac{1}{2}$ is a normalization factor to ensure that $\xi_0(\mathbf{p}_\sigma)$ ranges between 0 (least jagged distribution) and 1 (most jagged distribution). For instance, the two distributions in the example of Section 3.3 yield the values $\xi_0(\mathbf{p}_{\sigma_1}) = 0.0375$ and $\xi_0(\mathbf{p}_{\sigma_2}) = .4050$.

A matrix analogue to the $\mathbf{C}_1$ matrix of Section 4.4.1, whose goal is to minimize $\xi_0(\mathbf{p}_\sigma)$ instead of $\xi_1(\mathbf{p}_\sigma)$, would be

$$\mathbf{C}_0 = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n} \tag{35}$$

By using $\mathbf{C}_0$, one could build regularization-based ordinal quantification methods based on $\xi_0(\mathbf{p}_\sigma)$ rather than on $\xi_1(\mathbf{p}_\sigma)$.

The main difference between $\xi_0(\mathbf{p}_\sigma)$ and $\xi_1(\mathbf{p}_\sigma)$ is that, for each class y_i, in $\xi_1(\mathbf{p}_\sigma)$ we look at the prevalence values of *both* its right neighbor and its left neighbor, while in $\xi_0(\mathbf{p}_\sigma)$ we look at the prevalence value of its right neighbor *only*. Unsurprisingly, $\xi_0(\mathbf{p}_\sigma)$ has a different behavior than $\xi_1(\mathbf{p}_\sigma)$. For example, unlike for $\xi_1(\mathbf{p}_\sigma)$, for $\xi_0(\mathbf{p}_\sigma)$ there is a unique least jagged distribution, namely, the uniform distribution $p_\sigma(y) = \frac{1}{n}\ \forall\, y \in \mathcal{Y}$.

More importantly, $\xi_0(\mathbf{p}_\sigma)$ and $\xi_1(\mathbf{p}_\sigma)$ are not monotonic functions of each other; for instance, given the distributions $\mathbf{p}_{\sigma_2}$ (from Section 3.3) and $\mathbf{p}_{\sigma_3} = (0.00, 0.00, 0.00, 0.00, 1.00)$, it is easy to check that $\xi_1(\mathbf{p}_{\sigma_2}) > \xi_1(\mathbf{p}_{\sigma_3})$ but $\xi_0(\mathbf{p}_{\sigma_2}) < \xi_0(\mathbf{p}_{\sigma_3})$. Hence, the choice of the jaggedness measure indeed makes a difference in methods that regularize with respect to jaggedness. Ultimately, it seems reasonable to have the designer choose which function ideally reflects the notion of "ordinal plausibility" in the specific application being tackled.

While the particular mathematical form of $\xi_0(\mathbf{p}_\sigma)$, as from Equation 34, may seem empirical, a mathematical justification comes from the following observation: in fact, $\xi_0(\mathbf{p}_\sigma)$ measures the amount of deviation from a polynomial of degree 0 (i.e., from a constant line) of our predicted distribution $\hat{\mathbf{p}}_\sigma$. This observation reveals the meaning of the subscript "0" in $\xi_0(\mathbf{p}_\sigma)$. In contrast, $\xi_1(\mathbf{p}_\sigma)$ measures the amount of deviation from a polynomial of degree 1 (i.e.,

from any straight line) of $\hat{\mathbf{p}}_\sigma$. Indeed, all of the least jagged distributions (according to ξ_1) listed at the end of Section 3.3 are perfect fits to a straight line (assuming equidistant classes). For instance,

$$\mathbf{p}_{\sigma_4} = (0.0, 0.1, 0.2, 0.3, 0.4) \tag{36}$$

represents the sequence of points $((1, 0.0), (2, 0.1), (3, 0.2), (4, 0.3), (5, 0.4))$ that lies on the straight line $y = \frac{1}{10}x - \frac{1}{10}$.

Yet another notion of jaggedness might be implemented by the function

$$\xi_2(\mathbf{p}_\sigma) = \frac{1}{8}\sum_{i=1}^{n-3}(3p_\sigma(y_{i+1}) - 3p_\sigma(y_{i+2}) + p_\sigma(y_{i+3}) - p_\sigma(y_i))^2 \tag{37}$$

which measures the amount of deviation from a polynomial of degree 2 (i.e., a parabola); while $\xi_1(\mathbf{p}_\sigma)$ penalizes the presence of *any* hump in the distribution, $\xi_2(\mathbf{p}_\sigma)$ would penalize the presence of *more than one* hump. For instance, the distribution

$$\mathbf{p}_{\sigma_5} = (0.129, 0.093, 0.127, 0.231, 0.405) \tag{38}$$

would be a perfectly smooth distribution according to $\xi_2(\mathbf{p}_\sigma)$ but not according to $\xi_0(\mathbf{p}_\sigma)$ and $\xi_1(\mathbf{p}_\sigma)$ because it produces points that lie on the parabola $y = 0.035x^2 - 0.141x + 0.235$, which is displayed in Figure 5. A matrix analogue of $\xi_2(\mathbf{p}_\sigma)$ would be

$$\mathbf{C}_2 = \begin{pmatrix} -1 & 3 & -3 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -3 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 3 & -3 & 1 \end{pmatrix} \in \mathbb{R}^{(n-3)\times n} \tag{39}$$

In fact, we can produce matrices that penalize the deviation from polynomials of *any* chosen degree. To achieve this goal, we first need to multiply – with the transpose of itself, an arbitrary amount of times – a square variant of $\mathbf{C}_0$,

$$\mathbf{C}' = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n\times n} \tag{40}$$

which is the original $\mathbf{C}_0$ matrix with one additional row appended at the end. Second, we need to omit the outermost rows of this multiplication. That is, omitting the last row of $\mathbf{C}'$ yields $\mathbf{C}_0$, omitting the first and last rows of $(\mathbf{C}')^\top\mathbf{C}'$ yields $\mathbf{C}_1$, omitting the first one and the last two rows of $((\mathbf{C}')^\top\mathbf{C}')^\top\mathbf{C}'$ yields $\mathbf{C}_2$, up to only a constant factor. This procedure provides us with matrices $\mathbf{C}_3$, $\mathbf{C}_4$, ... that correspond to jaggedness measures $\xi_3(\mathbf{p}_\sigma)$, $\xi_4(\mathbf{p}_\sigma)$, ... and penalize deviations from polynomials of degree 3, 4, and so on.

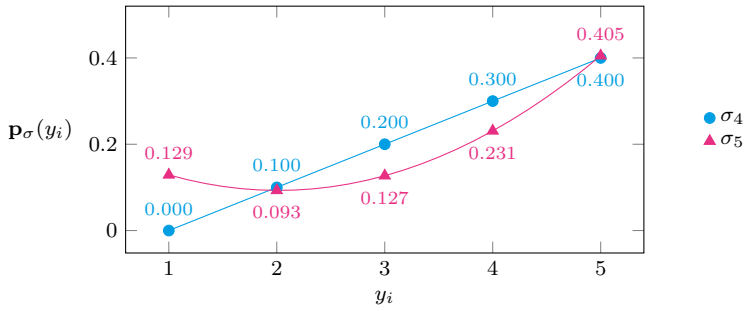

Fig. 5 The ordinal distributions $\mathbf{p}_{\sigma_4}$ (blue circles) and $\mathbf{p}_{\sigma_5}$ (red triangles). The lines display perfect polynomial fits of degree 1 (blue) and degree 2 (red).

In this article, we have chosen ξ_1 as our primary measure of jaggedness because ξ_1 reflects the assumption of ordered classes in a *minimal* sense. In contrast to ξ_0, it permits many different distributions that are all least jagged. Using ξ_0 would instead promote the uniform distribution exclusively, which would remain the least jagged distribution even if the order of the classes was randomly shuffled and, hence, meaningless in terms of OQ. In contrast to ξ_2 (or ξ_3, ξ_4, ...), our chosen ξ_1 is more general in the sense that it does not impose any certain shape (like parabolas, third-order polynomials, etc.) other than the most simple shape that exhibits small differences between consecutive classes. Hence, we consider ξ_1 to be the most suitable notion of jaggedness for studying the general value of regularization in OQ. It reflects the minimal OQ assumption that neighboring classes are similar, in the sense that they have similar prevalence values. We conceive other notions of jaggedness, used to reflect particular OQ applications, to be covered in future work.

8 Conclusions

We have carried out a thorough investigation of ordinal quantification, which includes (i) making available two datasets for OQ, generated according to the strong extraction protocols APP and APP-OQ and according to real prevalence vectors, which overcome the limitations of existing OQ datasets, (ii) showing that OQ cannot be profitably tackled by simply embedding ordinal classifiers into non-ordinal quantification methods, (iii) proposing seven OQ methods (o-ACC, o-PACC, o-HDx, o-HDy, o-SLD, o-EDy, and o-PDF) that combine intuitions from existing, ordinal and non-ordinal quantification methods and from existing, physics-inspired "unfolding" methods, and (iv) experimentally comparing our newly proposed OQ methods with existing non-ordinal quantification methods, ordinal quantification methods, and "unfolding" methods, which we have shown to be OQ methods under a different name. Our newly proposed OQ methods outperform the competition, a finding that our appendix confirms with additional error measures and datasets.

At the heart of the success of our newly proposed methods lies regularization, which is motivated by the ordinal plausibility assumption, i.e., the assumption that typical OQ class prevalence vectors are smooth. In future work, we plan to investigate other ways of achieving ordinal plausibility, to address different notions of smoothness, and to develop regularization terms that address characteristics of other quantification problems outside of OQ.

Acknowledgments

We thank Pablo González for clarifying the details of the experiments reported by Castaño et al. (2024). The work by M.B., A.M., and F.S. has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871042 (SoBigData++). A.M. and F.S. have further been supported by the AI4Media project, funded by the European Commission (Grant 951911) under the H2020 Programme ICT-48-2020, and by the SoBigData.it, FAIR, and QuaDaSh (P2022TB5JF) projects funded by the Italian Ministry of University and Research under the NextGenerationEU program. The authors' opinions do not necessarily reflect those of the funding agencies.

References

Aad G, Abbott B, Abbott DC, et al. (2021) Measurements of the inclusive and differential production cross sections of a top-quark–antiquark pair in association with a Z boson at $\sqrt{s} = 13$ TeV with the ATLAS detector. European Physics Journal C 81(8)

Aartsen MG, Ackermann M, Adams J, et al. (2017) Measurement of the ν_μ energy spectrum with IceCube-79. European Physics Journal C 77(692), DOI 10.1140/epjc/s10052-017-5261-3

Aleksić J, et al. (2015) Measurement of the Crab Nebula spectrum over three decades in energy with the MAGIC telescopes. Journal of High Energy Astrophysics 5-6:30–38, DOI https://doi.org/10.1016/j.jheap.2015.01.002

Anderhub H, Backes M, Biland A, et al. (2013) Design and operation of FACT, the first G-APD Cherenkov telescope. Journal of Instrumentation 8(6), DOI 10.1088/1748-0221/8/06/P06008

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana MJ (2010) Quantification via probability estimators. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2010), Sydney, AU, pp 737–742, DOI 10.1109/icdm.2010.75

Blobel V (1985) Unfolding methods in high-energy physics experiments. Tech. Rep. DESY-84-118, CERN, Geneva, CH, DOI 10.5170/CERN-1985-009.88

Blobel V (2002) An unfolding method for high-energy physics experiments. In: Proceedings of the Conference on Advanced Statistical Techniques in Particle Physics, Durham, UK, pp 258–267, extended version available at `https://www.desy.de/~sschmitt/blobel/punfold.ps`

Börner M, Hoinka T, Meier M, Menne T, Rhode W, Morik K (2017) Measurement/simulation mismatches and multivariate data discretization in the machine learning era. In: Proceedings of the 27th Conference on Astronomical Data Analysis Software and Systems (ADASS 2017), Santiago, CL, pp 431–434

Breiman L (1996) Out-of-bag estimation. Tech. rep., Department of Statistics, University of California, Berkeley, US

Bunse M (2022a) On multi-class extensions of adjusted classify and count. In: Proceedings of the 2nd International Workshop on Learning to Quantify (LQ 2022), Grenoble, IT, pp 43–50

Bunse M (2022b) Unification of algorithms for quantification and unfolding. In: Proceedings of the Workshop on Machine Learning for Astroparticle Physics and Astronomy, pp 459–468, DOI 10.18420/INF2022_37

Bunse M, Piatkowski N, Morik K, Ruhe T, Rhode W (2018) Unification of deconvolution algorithms for Cherenkov astronomy. In: Proceedings of the 5th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2018), Torino, IT, pp 21–30, DOI 10.1109/DSAA.2018.00012

Bunse M, Moreo A, Sebastiani F, Senz M (2022) Ordinal quantification through regularization. In: Proceedings of the 33rd European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML / PKDD 2022), Grenoble, FR, pp 36–52

Castaño A, González P, González JA, del Coz JJ (2024) Matching distributions algorithms based on the Earth mover's distance for ordinal quantification. IEEE Transactions on Neural Networks and Learning Systems 35(1):1050–1061, DOI 10.1109/TNNLS.2022.3179355

Da San Martino G, Gao W, Sebastiani F (2016) Ordinal text quantification. In: Proceedings of the 39th ACM Conference on Research and Development in Information Retrieval (SIGIR 2016), Pisa, IT, pp 937–940, DOI 10.1145/2911451.2914749

D'Agostini G (1995) A multidimensional unfolding method based on Bayes' theorem. Nuclear Instruments and Methods in Physics Research: Section A 362(2-3):487–498

D'Agostini G (2010) Improved iterative Bayesian unfolding. ArXiv:1010.0632 [physics.data-an]

Esuli A (2016) ISTI-CNR at SemEval-2016 Task 4: Quantification on an ordinal scale. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, US, pp 92–95, DOI 10.18653/v1/s16-1011

Esuli A, Sebastiani F (2010) Sentiment quantification. IEEE Intelligent Systems 25(4):72–75

Esuli A, Moreo A, Sebastiani F (2018) A recurrent neural network for sentiment quantification. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), Torino, IT, pp 1775–1778, DOI 10.1145/3269206.3269287

Esuli A, Moreo A, Sebastiani F, Sperduti G (2022) A detailed overview of LeQua 2022: Learning to quantify. In: Working Notes of the 13th Conference

and Labs of the Evaluation Forum (CLEF 2022), Bologna, IT

Esuli A, Fabris A, Moreo A, Sebastiani F (2023) Learning to quantify. Springer Nature, Cham, CH

Firat A (2016) Unified framework for quantification, arXiv:1606.00868v1 [cs.LG] 2 Jun 2016

Forman G (2005) Counting positives accurately despite inaccurate classification. In: Proceedings of the 16th European Conference on Machine Learning (ECML 2005), Porto, PT, pp 564–575, DOI 10.1007/11564096_55

Gao W, Sebastiani F (2016) From classification to quantification in tweet sentiment analysis. Social Network Analysis and Mining 6(19):1–22, DOI 10.1007/s13278-016-0327-z

González P, Castaño A, Chawla NV, del Coz JJ (2017) A review on quantification learning. ACM Computing Surveys 50(5):74:1–74:40, DOI 10.1145/3117807

González P, del Coz JJ (2021) Histogram-based deep neural network for quantification (abstract). In: Proceedings of the 1st International Workshop on Learning to Quantify (LQ 2021), Virtual Event

González-Castro V, Alaiz-Rodríguez R, Alegre E (2013) Class distribution estimation based on the Hellinger distance. Information Sciences 218:146–164, DOI 10.1016/j.ins.2012.05.028

Higashinaka R, Funakoshi K, Inaba M, Tsunomori Y, Takahashi T, Kaji N (2017) Overview of the 3rd Dialogue Breakdown Detection challenge. In: Proceedings of the 6th Dialog System Technology Challenge, Long Beach, US

Hoecker A, Kartvelishvili V (1996) SVD approach to data unfolding. Nuclear Instruments and Methods in Physics Research: Section A 372(3):469–481

Kawakubo H, du Plessis MC, Sugiyama M (2016) Computationally efficient class-prior estimation under class balance change using energy distance. IEICE Transactions on Information Systems 99-D(1):176–186, DOI 10.1587/transinf.2015EDP7212

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) RoBERTa: A robustly optimized BERT pretraining approach. ArXiv:1907.11692

McAuley JJ, Targett C, Shi Q, van den Hengel A (2015) Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2015), Santiago, CL, pp 43–52, DOI 10.1145/2766462.2767755

Moreno-Torres JG, Raeder T, Alaíz-Rodríguez R, Chawla NV, Herrera F (2012) A unifying view on dataset shift in classification. Pattern Recognition 45(1):521–530, DOI 10.1016/j.patcog.2011.06.019

Mueller JL, Siltanen S (2012) Linear and nonlinear inverse problems with practical applications. Society for Industrial and Applied Mathematics, Philadelphia, US, DOI 10.1137/1.9781611972344

Nachman B, Urbanek M, de Jong WA, Bauer CW (2020) Unfolding quantum computer readout noise. npj Quantum Information 6(84), DOI 0.1038/

s41534-020-00309-7
Nakov P, Ritter A, Rosenthal S, Sebastiani F, Stoyanov V (2016) SemEval-2016 Task 4: Sentiment analysis in Twitter. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, US, pp 1–18, DOI 10.18653/v1/s16-1001
Nöthe M, Adam J, Ahnen ML, et al. (2017) FACT – Performance of the first Cherenkov telescope observing with SiPMs. In: Proceedings of the 35th International Cosmic Ray Conference (ICRC 2017), pp Busan, KR
Pedregosa F, Bach F, Gramfort A (2017) On the consistency of ordinal regression methods. Journal of Machine Learning Research 18:55:1–55:35
Pérez-Gállego P, Castaño A, Quevedo JR, del Coz JJ (2019) Dynamic ensemble selection for quantification tasks. Information Fusion 45:1–15, DOI 10.1016/j.inffus.2018.01.001
Rennie JD, Srebro N (2005) Loss functions for preference levels: Regression with discrete ordered labels. In: Proceedings of the IJCAI 2005 Workshop on Advances in Preference Handling
Rosenthal S, Farra N, Nakov P (2017) SemEval-2017 Task 4: Sentiment analysis in Twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017), Vancouver, CA, pp 502–518, DOI 10.18653/v1/s17-2088
Rubner Y, Tomasi C, Guibas LJ (1998) A metric for distributions with applications to image databases. In: Proceedings of the 6th International Conference on Computer Vision (ICCV 1998), Mumbai, IN, pp 59–66
Ruhe T, Schmitz M, Voigt T, Wornowizki M (2013) DSEA: A data mining approach to unfolding. In: Proceedings of the 33rd International Cosmic Ray Conference (ICRC 2013), Rio de Janeiro, BR, pp 3354–3357
Saerens M, Latinne P, Decaestecker C (2002) Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. Neural Computation 14(1):21–41, DOI 10.1162/089976602753284446
Sakai T (2018) Comparing two binned probability distributions for information access evaluation. In: Proceedings of the 41st International ACM Conference on Research and Development in Information Retrieval (SIGIR 2018), Ann Arbor, US, pp 1073–1076, DOI 10.1145/3209978.3210073
Sakai T (2021) A closer look at evaluation measures for ordinal quantification. In: Proceedings of the CIKM 2021 Workshop on Learning to Quantify, Virtual Event
Schmelling M (1994) The method of reduced cross-entropy: A general approach to unfold probability distributions. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 340(2):400–412
Schmitt S (2012) TUnfold, an algorithm for correcting migration effects in high-energy physics. Journal of Instrumentation 7(10)
Werman M, Peleg S, Rosenfeld A (1985) A distance metric for multidimensional histograms. Computer Vision, Graphics, and Image Processing 32:328–336

Zeng Z, Kato S, Sakai T (2019) Overview of the NTCIR-14 Short Text Conversation task: Dialogue Quality and Nugget Detection subtasks. In: Proceedings of the 14th Workshop on NII Testbeds and Community for Information access Research (NTCIR 2019), Tokyo, JP, pp 289–315

Zeng Z, Kato S, Sakai T, Kang I (2020) Overview of the NTCIR-15 Dialogue Evaluation task (DialEval-1). In: Proceedings of the 15th Workshop on NII Testbeds and Community for Information access Research (NTCIR 2020), Tokyo, JP, pp 13–34

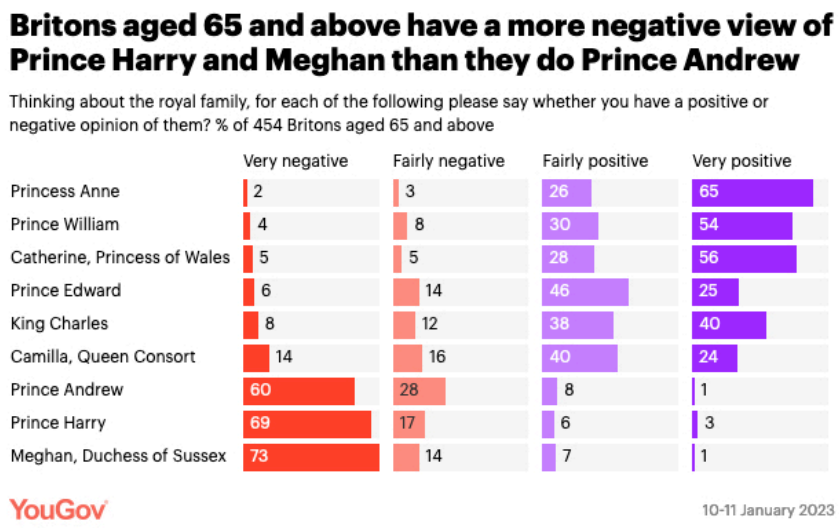

Fig. 6 Nine ordinal distributions, one for each major member of the British royal family, all of them using an ordinal code frame of 4 classes (from VeryNegative to VeryPositive).

A Are all ordinal distributions smooth?

In Section 3.3 we have made the claim that smoothness is a characteristic property of ordinal distributions in general. We have supported this claim by showing that the 29 distributions resulting from the dataset of 233.1M Amazon product reviews made available by (McAuley et al., 2015) and the dataset of telescope recordings of the FACT telescope made available by (Anderhub et al., 2013) (see Figure 2), are indeed fairly smooth. This observation, that smoothness is pervasive in ordinal distributions, suggests that our experiments do not "overfit" the two datasets we use for testing, i.e., AMAZON-OQ-BK and FACT-OQ. Concerning this suggestion, also note that we have selected the "Books" category from the former dataset only because it is the largest among its 28 categories, and not for any other reason.

In this section we report other examples which show that smoothness is indeed a characteristic property of ordinal distributions in general. Each such example is an ordinal distribution resulting from a survey run by the YouGov market research company and freely available on its website.[15]

For instance, Figure 6,[16] taken from the report of a survey of the public opinion on members of the British royal family, shows 9 ordinal distributions, one for each major member of the family, all of them using an ordinal code frame of 4 classes (from VeryNegative to VeryPositive); it is easy to note that all of them are fairly smooth (with $\xi_1(\mathbf{p}_\sigma)$ ranging in [.015,.106] and averaging .047).

A second example is the one illustrated in Figure 7,[17] which concerns a survey of the public opinion on British politician Keir Starmer; here the four ordinal distributions are less smooth than those of the previous example because they exhibit an upward hump in the three middle classes, but they are all fairly smooth elsewhere. Here, $\xi_1(\mathbf{p}_\sigma)$ ranges in [.040,.069] and averages .051.

[15] https://yougov.co.uk/

[16] Downloaded from https://yougov.co.uk/topics/politics/articles-reports/2023/01/12/prince-harrys-popularity-falls-further-spare-hits-

[17] Downloaded from https://yougov.co.uk/topics/politics/articles-reports/2023/04/04/three-years-what-do-britons-make-keir-starmers-tim

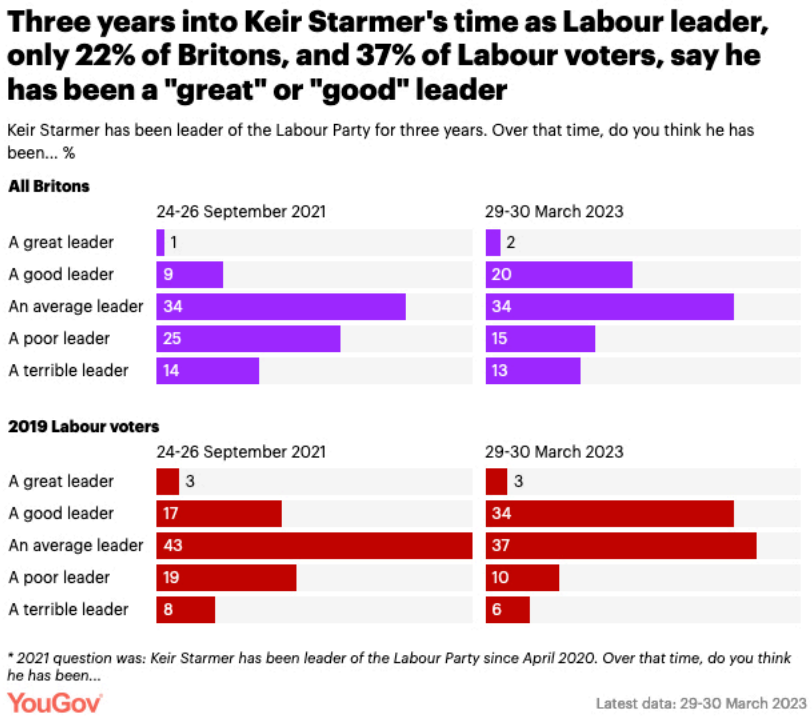

Fig. 7 Four ordinal distributions about public opinion on Keir Starmer, all of them using an ordinal code frame of 5 classes (from GreatLeader to TerribleLeader).

Our third example (see Figure 8)[18] displays a situation similar to the one illustrated by our first one. This is about the public opinion on how good the government of British Prime Minister Rishi Sunak was in delivering his pledges, and displays five ordinal distributions on a 5-point scale, ranging from VeryWell to VeryBadly; here, all the distributions are very smooth, with $\xi_1(\mathbf{p}_\sigma)$ ranging in [.001,.024] and averaging .009.

All in all, this is further evidence, although of an anecdotal nature, that the basic intuition on which our methods rest, i.e., that ordinal distributions tend to be fairly smooth, is indeed verified in practice.

B Other results

The following results complete the experiments we have shown in the main paper.

B.1 Other measures for evaluating ordinal quantifiers

Another function for measuring the quality of OQ estimates is the *Root Normalized Order-aware Divergence* (RNOD), proposed by Sakai (2018) and defined as

[18] Downloaded from `https://yougov.co.uk/topics/politics/articles-reports/2023/03/27/few-britons-think-government-doing-good-job-delive`

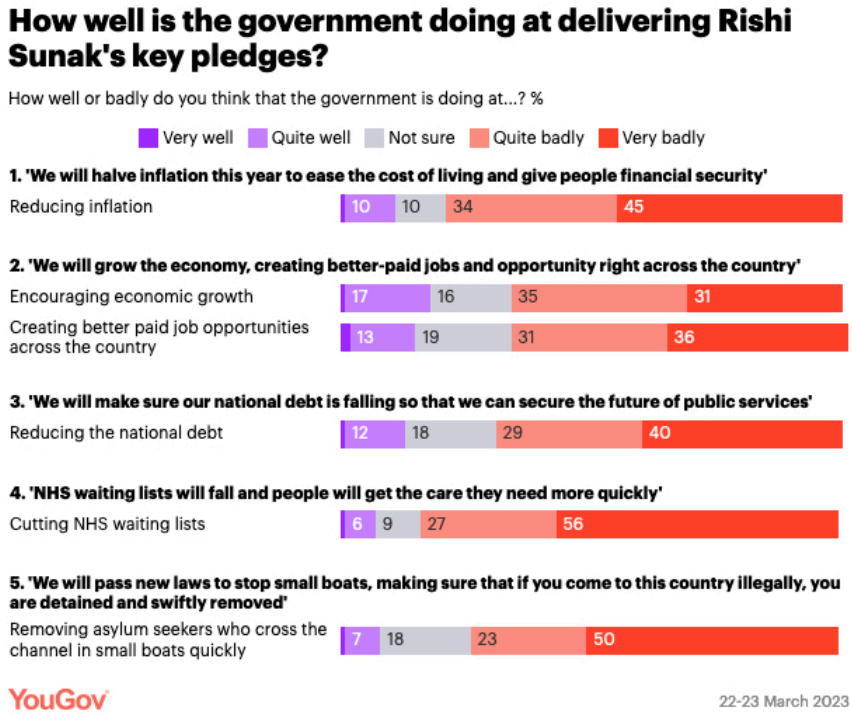

Fig. 8 Five ordinal distributions about public opinion on how good the government of British Prime Minister Rishi Sunak was in delivering his pledges, all of them using an ordinal code frame of 5 classes (from VeryWell to VeryBadly).

$$\mathrm{RNOD}(\mathbf{p}, \hat{\mathbf{p}}) = \sqrt{\frac{\sum_{y_i \in \mathcal{Y}^*} \sum_{y_j \in \mathcal{Y}} d(y_j, y_i)(p(y_j) - \hat{p}(y_j))^2}{|\mathcal{Y}^*|(n-1)}} \tag{41}$$

where $\mathcal{Y}^* = \{y_i \in \mathcal{Y} \mid p(y_i) > 0\}$.

So far, we have focused on NMD because RNOD hiddenly (i.e., without making it explicit) penalizes more heavily those mistakes (i.e., "transfers" of probability mass from a class to another) that are closer to the extremes of the code frame than those mistakes that are closer to its center. Other measures by Sakai (2021) also exhibit this problem, such as *Root Symmetric Normalized Order-aware Divergence* and *Root Normalized Average Distance-Weighted sum of squares*. A more detailed argument on why these measures are not satisfactory for OQ is given by Esuli et al. (2023, Section 3.2.2).

Despite this inadequacy, we here include an evaluation in terms of RNOD. That is, we repeat all of our experiments by replacing the NMD evaluation function with the RNOD evaluation function discussed in Section 3.2. Note that by adopting RNOD we are not simply replacing the evaluation measure, but also the criterion for model selection. That is to say, we re-run all the experiments anew, this time optimizing hyper-parameters by minimizing RNOD in place of NMD.

From examining the RNOD results from Tables 5 and 6, we note that, while some methods change positions in the ranking, as compared to their ranks in terms of NMD, our general conclusions from the NMD evaluation also hold in terms of RNOD.

Table 5 Same as Table 3 (Amazon-OQ-BK data) but using RNOD (see Equation 41) instead of NMD as the evaluation measure.

		APP	APP-OQ	real
non-ordinal baselines	CC	.1158 ± .0463	.0835 ± .0280	.0511 ± .0303
	PCC	.1324 ± .0515	.0974 ± .0333	.0575 ± .0313
	ACC	.0675 ± .0317	.0505 ± .0241	.0338 ± .0184
	PACC	.0623 ± .0275	.0474 ± .0198	.0334 ± .0177
	HDx	.0592 ± .0248	.0469 ± .0199	.0373 ± .0200
	HDy	.0611 ± .0271	.0473 ± .0205	.0363 ± .0209
	SLD	.0469 ± .0254	**.0414 ± .0184**	**.0318 ± .0162**
ordinal baselines	OQT	.1448 ± .0622	.1052 ± .0377	.0592 ± .0320
	ARC	.1297 ± .0570	.0947 ± .0349	.0477 ± .0272
	IBU	.0692 ± .0301	.0510 ± .0193	.0338 ± .0182
	RUN	.0669 ± .0298	.0498 ± .0198	.0338 ± .0184
	EDy	.0663 ± .0292	.0506 ± .0213	.0379 ± .0199
	PDF	.0681 ± .0301	.0525 ± .0226	.0378 ± .0204
new ordinal methods	o-ACC	.0675 ± .0317	.0505 ± .0241	.0338 ± .0184
	o-PACC	.0624 ± .0275	.0475 ± .0198	.0334 ± .0177
	o-HDx	.0592 ± .0248	.0469 ± .0199	.0372 ± .0199
	o-HDy	.0611 ± .0271	.0473 ± .0205	.0363 ± .0207
	o-SLD	**.0418 ± .0208**	.0421 ± .0182	.0321 ± .0159
	o-EDy	.0647 ± .0284	.0496 ± .0213	.0393 ± .0201
	o-PDF	.0660 ± .0290	.0513 ± .0228	.0387 ± .0206

Table 6 Same as Table 4 (Fact-OQ data) but using RNOD (see Equation 41) instead of NMD as the evaluation measure.

		APP	APP-OQ	real
non-ordinal baselines	CC	.1164 ± .0321	.0734 ± .0150	.1010 ± .0067
	PCC	.1268 ± .0323	.0836 ± .0188	.1135 ± .0051
	ACC	.1068 ± .0406	.1132 ± .0342	.0849 ± .0313
	PACC	.0892 ± .0358	.0965 ± .0360	.0659 ± .0266
	HDx	.1833 ± .0636	.1630 ± .0427	.2061 ± .0702
	HDy	.0896 ± .0397	.0936 ± .0441	.0703 ± .0354
	SLD	.0728 ± .0248	.0627 ± .0204	.0505 ± .0193
ordinal baselines	OQT	.1307 ± .0321	.0880 ± .0198	.0755 ± .0054
	ARC	.1190 ± .0309	.0787 ± .0206	.0851 ± .0066
	IBU	.0800 ± .0264	.0563 ± .0111	.0449 ± .0086
	RUN	.0877 ± .0281	.0592 ± .0115	.0432 ± .0135
	EDy	.0829 ± .0277	.0598 ± .0187	.0403 ± .0110
	PDF	.1060 ± .0358	.1056 ± .0342	.0632 ± .0200
new ordinal methods	o-ACC	.0881 ± .0287	.0559 ± .0106	.0413 ± .0082
	o-PACC	.0764 ± .0287	**.0460 ± .0114**	**.0281 ± .0090**
	o-HDx	.1197 ± .0433	.0699 ± .0352	.0669 ± .1124
	o-HDy	.0876 ± .0408	.0479 ± .0117	.0300 ± .0239
	o-SLD	**.0693 ± .0242**	.0506 ± .0119	.0423 ± .0090
	o-EDy	.0825 ± .0278	.0525 ± .0134	.0381 ± .0136
	o-PDF	.0916 ± .0307	.0527 ± .0115	.0295 ± .0090

Table 7 Results, evaluated in terms of NMD, of the experiments performed on additional datasets obtained from OpenML.

method	OpenML-Yolanda-OQ		OpenML-fried-OQ	
	APP	APP-OQ (20%)	APP	APP-OQ (20%)
CC	.0847 ± .0240	.0836 ± .0230	.0282 ± .0067	.0230 ± .0055
PCC	.1111 ± .0461	.1081 ± .0484	.0459 ± .0105	.0408 ± .0108
ACC	.0751 ± .0214	.0740 ± .0186	.0166 ± .0059	.0180 ± .0058
PACC	.0593 ± .0210	.0536 ± .0169	.0150 ± .0063	.0164 ± .0054
HDx	.0596 ± .0223	.0578 ± .0308	.0963 ± .0803	.0993 ± .0860
HDy	.0615 ± .0267	.0593 ± .0359	.0132 ± .0093	.0139 ± .0042
SLD	.0789 ± .0170	.0792 ± .0152	.0278 ± .0050	.0291 ± .0049
OQT	.2639 ± .0705	.2602 ± .0710	.0413 ± .0100	.0345 ± .0079
ARC	.2180 ± .0554	.2158 ± .0557	.0441 ± .0174	.0427 ± .0191
IBU	.0544 ± .0163	.0480 ± .0163	.0130 ± .0040	.0129 ± .0033
RUN	.0538 ± .0147	.0466 ± .0151	.0163 ± .0058	.0207 ± .0058
EDy	.0532 ± .0151	.0515 ± .0147	.0133 ± .0042	.0119 ± .0033
PDF	.0964 ± .0293	.0954 ± .0265	.0132 ± .0037	.0126 ± .0034
o-ACC	.0526 ± .0152	.0443 ± .0138	.0165 ± .0059	.0130 ± .0039
o-PACC	**.0429 ± .0123**	**.0328 ± .0096**	.0149 ± .0063	.0123 ± .0029
o-HDx	.0529 ± .0156	.0457 ± .0161	.0688 ± .0884	.0704 ± .1042
o-HDy	.0471 ± .0387	.0371 ± .0294	.0130 ± .0120	.0116 ± .0028
o-SLD	.0658 ± .0181	.0657 ± .0181	.0249 ± .0049	.0235 ± .0047
o-EDy	.0480 ± .0142	.0434 ± .0136	.0093 ± .0030	**.0078 ± .0023**
o-PDF	.0524 ± .0159	.0454 ± .0161	**.0087 ± .0049**	.0083 ± .0027

B.2 Results on other datasets

We have repeated our experiments from Tables 3 and 4 with four additional datasets that we have obtained from the UCI machine learning repository[19] and from OpenML[20]. We discuss these additional datasets here, in the appendix, because they have two disadvantages, as compared to our main datasets Amazon-OQ-BK and Fact-OQ.

First, the additional datasets do not have separate "real" samples that we could predict or use to determine an appropriate percentage for APP-OQ. Therefore, we have to omit the real evaluation protocol and we have to make an ad-hoc choice about the percentage of APP samples that we maintain in APP-OQ. We set this percentage to 20%, which lies between the 50% used for Amazon-OQ-BK and the 5% used for Fact-OQ.

The second disadvantage of the additional datasets is that their original purpose is not OQ, not even ordinal classification, but regression. Therefore, we have equidistantly binned the range of their target variables to 10 ordinal classes that we can predict in OQ. We have chosen to use binned regression datasets because we were not able to find datasets that are originally ordinal and have a sufficient number of data items; in fact, APP and APP-OQ require huge datasets for drawing a training set and two large pools, one for validation and one for testing.

The results of our experiments with the additional data are reported in Tables 7 and 8. Despite the shortcomings of the employed data, these results confirm our main conclusions on OQ: the regularized methods consistently improve over their original non-regularized versions, at least being on par with these versions.

Table 9 further presents the results obtained with a weaker representation of the Amazon-OQ-BK data. Instead of the powerful RoBERTa embeddings used before, we now use a TF-IDF representation of the product reviews.

[19] `https://archive.ics.uci.edu/`

[20] `https://www.openml.org/`

Table 8 Results, evaluated in terms of NMD, of the experiments performed on additional datasets obtained from UCI.

method	UCI-BLOG-FEEDBACK-OQ		UCI-ONLINE-NEWS-POPULARITY-OQ	
	APP	APP-OQ (20%)	APP	APP-OQ (20%)
CC	.0850 ± .0325	.0780 ± .0310	.1357 ± .0420	.1226 ± .0386
PCC	.0894 ± .0370	.0835 ± .0365	.1018 ± .0446	.0970 ± .0456
ACC	.0773 ± .0299	.1402 ± .0410	.1042 ± .0464	.0996 ± .0478
PACC	.1099 ± .0455	.1089 ± .0419	.0850 ± .0325	.0865 ± .0362
HDx	.1045 ± .0509	.0973 ± .0518	.2075 ± .1133	.1966 ± .1251
HDy	.0704 ± .0455	.0891 ± .0530	.1175 ± .0523	.1120 ± .0579
SLD	**.0454 ± .0150**	**.0389 ± .0116**	.0892 ± .0337	.0810 ± .0299
OQT	.0953 ± .0379	.0842 ± .0350	.2007 ± .0594	.1915 ± .0591
ARC	.1248 ± .0368	.1073 ± .0324	.3192 ± .0843	.3209 ± .0837
IBU	.0631 ± .0217	.0557 ± .0189	.0814 ± .0285	**.0714 ± .0279**
RUN	.0821 ± .0249	.0765 ± .0232	.0881 ± .0349	.0865 ± .0368
EDy	.0563 ± .0189	.0497 ± .0150	.1356 ± .0397	.1388 ± .0384
PDF	.0652 ± .0209	.0584 ± .0181	.1220 ± .0419	.1147 ± .0382
o-ACC	.0681 ± .0198	.0623 ± .0200	.0940 ± .0385	.0941 ± .0413
o-PACC	.0744 ± .0287	.0658 ± .0249	.0808 ± .0296	**.0723 ± .0334**
o-HDx	.0982 ± .0506	.0888 ± .0534	.1075 ± .1158	**.1009 ± .1305**
o-HDy	.0645 ± .0338	.0617 ± .0342	.0905 ± .0353	.0795 ± .0304
o-SLD	**.0454 ± .0151**	**.0387 ± .0116**	**.0769 ± .0292**	**.0698 ± .0275**
o-EDy	.0944 ± .0348	.0833 ± .0224	**.0764 ± .0264**	**.0675 ± .0256**
o-PDF	.0967 ± .0331	.0931 ± .0298	.0821 ± .0295	.0711 ± .0260

Table 9 Same as Table 3 but using a TF-IDF representation instead of RoBERTa embeddings for the AMAZON-OQ-BK data.

		APP	APP-OQ	real
non-ordinal baselines	CC	.0877 ± .0337	.0751 ± .0308	.0444 ± .0253
	PCC	.1101 ± .0452	.1010 ± .0466	.0673 ± .0392
	ACC	.0328 ± .0208	.0339 ± .0212	.0276 ± .0184
	PACC	.0276 ± .0154	.0279 ± .0161	.0234 ± .0112
	HDy	.0256 ± .0125	.0264 ± .0127	.0235 ± .0125
	SLD	.0495 ± .0204	.0433 ± .0153	.0414 ± .0124
ordinal baselines	IBU	.0294 ± .0139	.0247 ± .0107	**.0177 ± .0078**
	RUN	.0302 ± .0152	.0307 ± .0151	.0252 ± .0122
	EDy	.0480 ± .0220	.0429 ± .0193	.0293 ± .0150
	PDF	.0661 ± .0280	.0694 ± .0286	.0544 ± .0319
new ordinal methods	o-ACC	.0324 ± .0208	.0258 ± .0165	.0224 ± .0155
	o-PACC	.0268 ± .0150	.0240 ± .0144	.0202 ± .0086
	o-HDy	.0246 ± .0115	.0234 ± .0102	.0219 ± .0100
	o-SLD	.0488 ± .0199	.0413 ± .0127	.0271 ± .0108
	o-EDy	**.0223 ± .0104**	**.0221 ± .0102**	.0214 ± .0082
	o-PDF	.0267 ± .0132	.0229 ± .0092	.0213 ± .0104

Table 10 Hyper-parameter grid used for the optimization of the quantification methods employed in the experiments reported in Tables 3 and 4.

method	parameter	values
CC	no parameters	
PCC	no parameters	
ACC	no parameters	
PACC	no parameters	
HDx	number of bins per feature	$\{2, 3, 4\}$
HDy	number of bins per class	$\{2, 4\}$
SLD	no parameters	
OQT	fraction of held-out data	$\{\frac{1}{3}\}$
ARC	fraction of held-out data	$\{\frac{1}{3}\}$
RUN	τ	{1e-3, 1e-1, 1e1}
IBU	order of polynomial	$\{0, 1\}$
	interpolation factor	$\{1e-2, 1e-1\}$
EDy	ground distance	{MD}
PDF	number of bins per class	$\{5, 10\}$
o-ACC	τ	{1e-5, 1e-3, 1e-1}
o-PACC	τ	{1e-5, 1e-3, 1e-1}
o-HDx	number of bins per feature	$\{2, 3, 4\}$
	τ	{1e-5, 1e-3, 1e-1}
o-HDy	number of bins per class	$\{2, 4\}$
	τ	{1e-5, 1e-3, 1e-1}
o-SLD	order of polynomial	$\{0, 1\}$
	interpolation factor	{1e-2, 1e-1}
o-EDy	ground distance	{MD}
	τ	{1e-5, 1e-3, 1e-1}
o-PDF	number of bins per class	$\{5, 10\}$
	τ	{1e-5, 1e-3, 1e-1}

B.3 Hyper-parameter grids

In our experiments each method has the opportunity to optimize its hyper-parameters on the validation samples of the respective evaluation protocol. These hyperparameters consist (i) of the parameters of the quantifier and (ii) of the parameters of the classifier with which the quantifier is equipped. After taking out preliminary experiments, which we omit here for conciseness, we have chosen slightly different hyper-parameter grids for the different datasets.

To this end, Tables 10 and 11 present the parameters for the Amazon-OQ-BK dataset. For instance, CC can choose between 10 hyper-parameter configurations of the classifier (2 class weights $\times$ 5 regularization strengths) but does not introduce additional parameters on the quantification level. We note that preliminary results revealed that the fraction of held-out data does not considerably affect the results of OQT and ARC. Therefore, and since those methods are computationally expensive, we decided to fix the proportion of the held-out split to $\frac{1}{3}$ and do not include this hyper-parameter in the exploration.

Tables 10 and 12 present the parameters for the Fact-OQ data. For conciseness, they also contain the parameters for the UCI and OpenML datasets.

Table 11 Hyper-parameter grid used for the optimization of the classifiers employed in the Amazon-OQ-BK experiments reported in Table 3.

classifier	parameter	values
Logistic Regression	class weight	{balanced, unbalanced}
	regularization parameter C	$\{0.001, 0.01, 0.1, 1.0, 10.0\}$

Table 12 Hyper-parameter grids used for the optimization of the classifiers employed in the Fact-OQ, OpenML, and UCI experiments reported in Tables 4, 7 and 8.

classifier	parameter	values
Random Forests	class weight	{balanced, unbalanced}
	splitting criterion	{Gini index, Entropy}
	maximum depth	$\{4, 8, 12\}$
	minimum examples per leaf node*	$\{1, 4, 16\}$

* fixed to 1 for the OpenML and UCI experiments.

C A differentiable surrogate loss for o-PDF

For o-PDF, we introduce another modification in addition to regularization, to facilitate the minimization of the resulting loss function. To understand this additional modification, recognize that the MD between one-dimensional histograms is merely the L_1 norm between the corresponding cumulative histograms, see Equation 3 and Castaño et al. (2024). As an L_1 norm, $\text{MD}(\mathbf{q}, \mathbf{Mp})$ is not differentiable at all points where $\mathbf{q}_i = [\mathbf{Mp}]_i$ for some i. Hence, its gradient is not always defined and numerical optimization methods, which require the gradient, can easily run into errors whenever the gradient is undefined.

To counteract this issue, our method o-PDF employs instead the squared L_2 norm between the cumulative histograms as a surrogate loss function. Hence, the regularized loss function of o-PDF is

$$\mathcal{L}(\mathbf{p};\, \mathbf{M}, \mathbf{q}, \tau) \;=\; \|\text{CDF}(\mathbf{q}) - \text{CDF}(\mathbf{Mp})\|_2^2 \;+\; \frac{\tau}{2}(\mathbf{C}_1\mathbf{p})^2 \tag{42}$$

where $\mathbf{q}$ and $\mathbf{M}$ are defined through Equation 25. Equation 42 behaves similar, in the vicinity of the optimum, to a direct minimization of MD. At the same time, it is continuously differentiable and therefore not prone to errors during a numerical minimization that leverages the function's derivatives.

An alternative solution would be to re-arrange the L_1 norm as follows:

$$\arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 = \arg\min_{\mathbf{x}} \sum_i |\mathbf{x}_i| = \arg\min_{\mathbf{x},\mathbf{t}} \sum_i \mathbf{t}_i \text{ s.t. } \mathbf{x}_i \leq \mathbf{t}_i, -\mathbf{x}_i \leq \mathbf{t}_i$$

The downside of this alternative is the introduction of two inequality constraints, which requires constrained minimization techniques. Our soft-max approach is otherwise unconstrained, facilitating optimization in terms of a reduced complexity and a greater availability of methods. Hence, we replace the L_1 norm with the L_2 norm to maintain these advantages in o-PDF.

In contrast, the original PDF implementation uses constrained minimization anyway. Hence, it does not encounter the introduction of additional constraints as a problem and can minimize the L_1 norm directly, at the cost of a more complex optimization problem and a limited choice of optimization techniques. To properly attribute performance values to the PDF method, we use the original implementation of Castaño et al. (2024) without any changes (in particular, with a direct minimization of the L_1 norm).