An Overview of LeQua 2024, the 2nd International Data Challenge on Learning to Quantify

Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani, and Gianluca Sperduti

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
{firstname.lastname}@isti.cnr.it

Abstract. LeQua 2024 is a data challenge about methods and systems for "learning to quantify" (a.k.a. "quantification", or "class prior estimation"), i.e., for training predictors of the relative frequencies of classes $\mathcal{Y} = \{y_1, ..., y_n\}$ in sets of unlabelled datapoints. While these predictions could be easily achieved by first classifying all datapoints via a classifier and then counting how many datapoints have been assigned to each class, a growing body of literature has shown this approach to be suboptimal, especially when the training data and the test data are affected by some form of dataset shift, and has proposed better methods. The goal of this data challenge is to provide a setting for the comparative evaluation of methods for learning to quantify. LeQua 2024 is the 2nd edition of the LeQua challenge, following the successful 1st edition of 2022.

In LeQua 2024, four tasks were offered. The first three tasks (T1, T2, T3) tackle learning to quantify under prior probability shift, while the fourth task (T4) tackles learning to quantify under covariate shift; T1 and T4 are about binary quantification, T2 is about single-label multiclass quantification, while T3 is about ordinal quantification. For all such tasks, data are provided to participants in ready-made vector form. In this overview article we describe in detail the structure of the data challenge and the results obtained by the participating teams.

1 Learning to Quantify

In a number of applications involving classification, the final goal is not determining which class (or classes) individual unlabelled datapoints (e.g., textual documents, images, or other) belong to, but estimating the *prevalence* (or "relative frequency", or "prior probability", or "prior") of each class $y \in \mathcal{Y} = \{y_1, ..., y_n\}$ in the unlabelled data. Training predictors of the class prevalence values in unlabelled data is known as *learning to quantify* (LQ – a.k.a. *quantification*, or *class prior estimation*) [14, 19, 22].

LQ has several applications in fields (such as the social sciences, political science, market research, epidemiology, and ecological modelling) which are inherently interested in characterising *aggregations* of individuals, rather than the individuals themselves; disciplines like the ones above are usually *not* interested in finding the needle in the haystack, but in characterising the haystack. For instance, in most applications of tweet sentiment classification we are not concerned with estimating the true class (e.g., Positive, or Negative, or Neutral) of individual tweets. Rather, we are concerned with estimating the relative frequencies of these classes in the set of unlabelled tweets under study; or, put in another way, we are interested in estimating as accurately as possible the true distribution of tweets across the classes.

It has by now unequivocally been shown that performing quantification by classifying each unlabelled instance and then counting, for each class, the instances that have been attributed to the class (the "classify and count" method), usually leads to poor quantification accuracy (see e.g., [3, 7, 9, 12, 13, 23, 33, 34]), due to (a) classifier bias and the mismatch between classification loss and quantification loss, and (b) the presence of dataset shift (see below). This suboptimality of "classify and count" also evokes "Vapnik's principle" [49], which states

> If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem.

In our case, the problem to be solved directly is quantification, while the more general intermediate problem is classification.

One reason why "classify and count" is suboptimal is that many application scenarios suffer from *dataset shift* [31, 41], defined as the situation in which the distribution $P(X, Y)$ from which the labelled training data L have been drawn is different from the distribution $Q(X, Y)$ from which the unlabelled data U have been drawn. The presence of dataset shift means that the well-known IID assumption, on which most learning algorithms for training classifiers hinge, does not hold. In turn, this means that "classify and count" will perform suboptimally on sets of unlabelled datapoints that exhibit dataset shift with respect to the training set, and that the higher the amount of this shift, the worse we can expect "classify and count" to perform.

As a result of the suboptimality of the "classify and count" method, LQ has slowly evolved as a task in its own right, different (in goals, methods, techniques, and evaluation measures) from classification [14, 22]. For the near future it is easy to foresee that the interest in LQ will increase, due (a) to the increased awareness that "classify and count" is a suboptimal solution when it comes to prevalence estimation, and (b) to the fact that, with larger and larger quantities of data becoming available and requiring interpretation, in more and more scenarios we will only be able to afford to analyse these data at the aggregate level rather than individually.

LeQua 2024 (`https://lequa2024.github.io/`) follows in the footsteps of LeQua 2022 [16, 17], the first edition of this data challenge. LeQua 2022 was the first data challenge ever to be entirely devoted to quantification; while this topic had surfaced in previous shared tasks, it had never been their real focus.

2 Setting up LeQua 2024

In quantification, a *datapoint* (usually represented as $\mathbf{x}$) is the individual unit of information; for instance, a textual document, an image, a video, are examples of datapoints. As in LeQua 2022, as datapoints we use textual objects (and, more specifically, product reviews); however, this choice causes no loss of generality, since these textual objects are provided to the participants already in vector form.

A datapoint $\mathbf{x}$ has a *class label*, i.e., it belongs to a certain class $y \in \mathcal{Y} = \{y_1, ..., y_n\}$; in this case we indicated by y the label of $\mathbf{x}$. In LeQua 2024, the classes are either the merchandise classes to which the products belong, or the sentiment scores that the authors have attached to the reviews they have written (see Section 2.2 for more). Some datapoints are such that their label is known to the quantification algorithm, and are thus called *labelled datapoints*; we typically use them as training examples for the quantifier-training algorithm. Some other datapoints are such that their label is unknown to the quantifier-training algorithm and to the trained quantifier, and are thus called *unlabelled datapoints*; for testing purposes we use datapoints whose label we hide to the quantifier-training algorithm and to the trained quantifier, and thus play the role of unlabelled datapoints.

Unlike a classifier, a quantifier must not predict labels for individual datapoints, but must predict *prevalence values* for *samples* (i.e., sets) of unlabelled datapoints. A prevalence value $p_\sigma(y_i)$ for a class $y_i \in \mathcal{Y}$ and a sample σ is a number in [0,1] such that the prevalence values $p_\sigma(y_1), ..., p_\sigma(y_n)$ for the classes in $\mathcal{Y}$ sum up to 1; in other words, $p_\sigma(y_1), ..., p_\sigma(y_n)$ are a *distribution* of the datapoints of σ over $\mathcal{Y}$. Note that when, in the following, we use the term "label", we always refer to the label of an individual datapoint (and not of a sample of datapoints; samples do not have labels, but prevalence values for classes).

2.1 The tasks

LeQua 2022 offered four tasks. In two of them, all (training, development, and test) datapoints were provided to participants in ready-made vector form, while in the other two the datapoints were provided in their original textual form. Each of these modalities included two variants: binary quantification and single-label multiclass quantification. In all four tasks, the data were characterised by prior probability shift (see below).

LeQua 2024 also offered four tasks (called T1, T2, T3, T4), but these tasks were (at least partially different) from those of LeQua 2024; in particular,

– In all LeQua 2024 tasks, the datapoints were provided to the participants in ready-made vector form; the goal was to allow the participants to concentrate on optimising their quantification methods, rather than spending time on optimising the process for producing vectorial representations of the datapoints.
– LeQua 2024 includes tasks characterised by different types of dataset shift.

Task	Codeframe structure	Type of dataset shift
T1	Binary	Prior probability shift
T2	Single-label multiclass	Prior probability shift
T3	Ordinal	Prior probability shift
T4	Binary	Covariate shift

Table 1: Main characteristics of the four tasks offered within LeQua 2024.

The main characteristics of the four tasks offered within LeQua 2024 are succinctly described in Table 1.

For each task, participant teams were required not to use any kind of (training / development / test) datapoints other than those provided for that task.

2.2 The data pool

The data we use are Amazon product reviews from a large crawl of such reviews (the same crawl we had used for LeQua 2022). From the result of this crawl we remove (a) all reviews shorter than 200 characters, and (b) all reviews that have not been recognised as "useful" by any user;[1] this yields the "pool" Ω of reviews that we use for our experimentation.

As for the set $\mathcal{Y}$ of class labels,

- for the two binary tasks (T1 and T4) we use two *sentiment* labels, i.e., Positive, which encompasses 4-stars and 5-stars reviews, and Negative, which encompasses 1-star and 2-stars reviews (we discard 3-stars reviews);
- for the multiclass task (T2) we use 28 *topic* labels, representing the merchandise class the product belongs to (e.g., Automotive, Baby, Beauty);[2]
- for the ordinal task (T3) we use the original sentiment label, which ranges on {1-Star, 2-Stars, 3-Stars, 4-Stars, 5-Stars}.

In order to generate the vectorial representations of the reviews we use the ELECTRA-Small model [10], giving each review in input to the generator and using the last hidden state of the model as the representation of the review. Each review is thus represented by a real-valued vector with 256 dimensions.

The data we use for LeQua 2024 are different from the ones we used for LeQua 2022. The main difference is that (a) different datapoints are chosen for the training samples, for the validation samples, and for the test samples, and (b) a different vectorisation is used (while we here use the ELECTRA-Small model, LeQua 2022 used a vectorisation method based on GloVe vectors [39]).

[1] This is meant to filter our "bogus" reviews (e.g., "I'm giving this 1 star because the package was damaged!") that would be difficult for any classifier to label correctly.

[2] The set of 28 topic classes is flat, i.e., there is no hierarchy defined upon it.

2.3 Types of dataset shift and types of data extraction protocols

Dataset shift is defined as the situation in which (i) the training (and development) data that are used for training a model are sampled from a joint distribution $P(X, Y)$, (ii) the unlabelled data on which the trained model is deployed are sampled from a joint distribution $Q(X, Y)$, and (iii) $P(X, Y) \neq Q(X, Y)$.

In LeQua 2024 we consider two types of dataset shift, i.e.,

1. *prior probability shift* (also known as *label shift*), defined as the case in which $P(Y) \neq Q(Y)$ and $P(X|Y) = Q(X|Y)$;
2. *covariate shift*, defined as the case in which $P(X) \neq Q(X)$ and $P(Y|X) = Q(Y|X)$.

The literature on quantification has mostly tackled prior probability shift, and only a few papers [5, 23, 48] have touched on the relationships between quantification and covariate shift.

2.4 The baseline systems

We made the participants aware of the availability of QuaPy [32], a Python-based open-source library[3] for quantification research and development that provides implementations of methods, evaluation measures, parameter optimisation routines, and evaluation protocols.

The implementations of the quantification methods we used as baselines can be accessed via GitHub.[4] These methods include:

– **Classify and Count** (CC): This is the trivial baseline, consisting in training a standard classifier h on the training set L, using this classifier to classify all the data items $\mathbf{x}$ in the sample σ, counting how many such items have been attributed to class y_i, doing this for all classes in $\mathcal{Y}$, and dividing the resulting counts by the cardinality $|\sigma|$ of the sample.

– **Probabilistic Classify and Count** (PCC) [2]: This is a probabilistic variant of CC where the "hard" classifier h is replaced by a "soft" (probabilistic) classifier s, and where counts are replaced by expected counts.

– **Adjusted Classify and Count** (ACC) [19]: This is an "adjusted" variant of CC in which the prevalence values predicted by CC are subsequently corrected by considering the misclassification rates of classifier h, as estimated on a held-out validation set. For our experiments, this held-out set consists of 40% of the training set.

– **Probabilistic Adjusted Classify and Count** (PACC) [2]: This is a probabilistic variant of ACC where the "hard" classifier h is replaced by a "soft" (probabilistic) classifier s, and where counts are replaced by expected counts. Equivalently, it is an "adjusted" variant of PCC in which the prevalence values predicted by PCC are corrected by considering the (probabilistic versions

[3] `https://github.com/HLT-ISTI/QuaPy`

[4] Check the branch `https://github.com/HLT-ISTI/QuaPy/tree/lequa2024/LeQua2024`

of the) misclassification rates of soft classifier s, as estimated on a held-out validation set. For our experiments, this held-out set consists of 40% of the training set.
- The **Saerens-Latinne-Decaestecker** algorithm (SLD) [43] (see also [15]): This is a method based on Expectation Maximization, whereby the posterior probabilities returned by a soft classifier s for data items in an unlabelled set U, and the class prevalence values for U, are iteratively updated in a mutually recursive fashion.
- **DM**, a multiclass implementation of the distribution-matching approach that adheres to the framework proposed by [4, 18], in which the divergence measure to minimise is the Hellinger Distance.
- The recently proposed **KDEy** algorithm [35], a distribution-matching method that models the distribution of the posterior probabilities using kernel density estimation. In particular, we adopt the variant that uses the Kullback-Leibler divergence as the target loss to minimise, which is akin to maximising the likelihood of the test data.

For all methods, we have trained the underlying classifiers via logistic regression, as implemented in the `scikit-learn` framework.[5] Note that all these methods are natively multiclass.

We optimize two hyperparameters of the logistic regression learner by exploring C (the inverse of the regularization strength) in the range $\{10^{-4}$, 10^{-3}, ..., $10^{+4}\}$ and CLASS_WEIGHT (indicating the relative importance of each class) in {"balanced", "not-balanced"}. For DM, we also explore the number of bins in the range $\{2, 3, \ldots, 10, 12, \ldots, 32, 64\}$. For KDEy, we additionally explore the bandwidth in the range $\{0.01, 0.02, \ldots, 0.20\}$. For each quantification method, model selection is carried out by choosing the combination of hyperparameters that yields the lowest official evaluation error used for each task across all validation samples made available to it.

2.5 Task T1: Binary Quantification under Prior Probability Shift

Task T1 is essentially the same as task T1A in LeQua 2022, i.e., it is a binary quantification task on data affected by prior probability shift, and was re-proposed in LeQua 2024 in order to monitor the progress of the field on what can be considered the "mother" of all quantification tasks.

As mentioned in Section 2.2, the datapoints here have binary sentiment labels, obtained by considering as Positive all 4-stars and 5-stars reviews and by considering as Negative all 1-star and 2-stars reviews.

To obtain our data, first of all we removed from our pool Ω all reviews scored "3-stars". We then obtained the L_1 training set by randomly extracting 5000 reviews from Ω (different from those we used in LeQua 2022), after which we removed them from Ω; the prevalence values of the positives and of the negatives in the extracted set are 0.78 and 0.22, respectively. Subsequently, we simulated

[5] `https://scikit-learn.org/stable/index.html`

the presence of prior probability shift by extracting from Ω development samples and test samples according to the *artificial prevalence protocol* (APP), by now a standard protocol for artificially injecting prior probability shift in the data to be used in the evaluation of quantifiers.

In the general multiclass case, the APP consists of taking the set Ω of datapoints remaining after the extraction of the training set, and extracting from it a number of subsets (the *development samples* and the *test samples*), each characterised by a *predetermined* vector $(p_\sigma(y_1), ..., p_\sigma(y_n))$ of prevalence values, where $y_1, ..., y_n$ are the classes of interest. In other words, for extracting a sample σ, we generate a vector of prevalence values, and randomly select datapoints from Ω accordingly (i.e., by class-conditional random selection of datapoints, until the desired class prevalence values are obtained). Note that in the data released to the participants, for each development sample only the prevalence values that characterise the sample, and not the label of each individual datapoint, was disclosed. The goal of the APP is to generate samples characterised by widely different vectors of prevalence values; this is meant to test the robustness of a *quantifier* (i.e., of an estimator of class prevalence values) in confronting class prevalence values possibly different (or very different) from the ones of the set it has been trained on. For doing this we draw the vectors of class prevalence values uniformly at random from the set of all legitimate such vectors, i.e., from the *unit* $(n-1)$-*simplex* of all vectors $(p_\sigma(y_1), ..., p_\sigma(y_n))$ such that $p_\sigma(y_i) \in [0, 1]$ for all $y_i \in \mathcal{Y}$ and $\sum_{y_i \in \mathcal{Y}} p_\sigma(y_i) = 1$. For this we use the Kraemer algorithm [47], whose goal is that of sampling in such a way that all legitimate class distributions are picked with equal probability. For each vector thus picked we randomly generate a test sample.

Note that the APP indeed simulates prior probability shift, since

- the fact that the samples are randomly selected according to a pre-specified vector of probability values different (in general) from the one that characterises the training set, simulates condition $P(Y) \neq Q(Y)$;
- the fact that the samples are drawn from the same data source from which the training data are drawn simulates condition $P(X|Y) = Q(X|Y)$;
- these two conditions are (see Bullet 1 in Section 2.3) what altogether characterise prior probability shift.

In the binary case ($n = 2$), for generating the D_1 development set we used the APP to extract 1000 development samples consisting of 250 reviews each; for generating the U_1 test set we extracted, in the same way, 5000 test samples also consisting of 250 reviews each. The prevalence values for all samples in U_1 were disclosed to the participants after the end of the challenge.

The evaluation measure. In a theoretical study on the adequacy of evaluation measures for quantification [46], *relative absolute error* (RAE) and *absolute error* (AE) have been found to be, for binary and multiclass quantification, the most satisfactory, and are thus the only measures used in LeQua 2022. RAE and AE are defined as

$$\mathrm{RAE}(p_\sigma, \hat{p}_\sigma) = \frac{1}{n} \sum_{y \in \mathcal{Y}} \frac{|\hat{p}_\sigma(y) - p_\sigma(y)|}{p_\sigma(y)} \tag{1}$$

$$\mathrm{AE}(p_\sigma, \hat{p}_\sigma) = \frac{1}{n} \sum_{y \in \mathcal{Y}} |\hat{p}_\sigma(y) - p_\sigma(y)| \tag{2}$$

where p_σ is the true distribution on sample σ, $\hat{p}_\sigma$ is the predicted distribution, $\mathcal{Y}$ is the set of classes of interest, and $n = |\mathcal{Y}|$. Note that RAE is undefined when at least one of the classes $y \in \mathcal{Y}$ is such that its prevalence in the sample σ of unlabelled datapoints is 0. To solve this problem, in computing RAE we smooth all $p_\sigma(y)$'s and $\hat{p}_\sigma(y)$'s via additive smoothing, i.e., we take $\underline{p}_\sigma(y) = (\epsilon + p_\sigma(y))/(\epsilon \cdot n + 1)$, where $\underline{p}_\sigma(y)$ denotes the smoothed version of $p_\sigma(y)$ and the denominator is just a normalising factor (same for the $\underline{\hat{p}}_\sigma(y)$'s); following [21], we use the quantity $\epsilon = 1/(2|\sigma|)$ as the smoothing factor. In Equation 1 we then use the smoothed versions of $p_\sigma(y)$ and $\hat{p}_\sigma(y)$ in place of their original non-smoothed versions; as a result, RAE is now always defined.

As the official measure according to which systems are ranked, we use RAE; we also compute AE results, but we do not use them for ranking the systems. The official score obtained by a given quantifier is the average value of the official evaluation measure (RAE) across all test samples; for each system we also compute and report the value of AE. For T1 (but we will do the same for T2, T3, T4 too) we use the Wilcoxon signed-rank test at different confidence levels ($\alpha = 0.05$ and $\alpha = 0.001$) to identify all participant runs that are *not* statistically significantly different from the best run, in terms of RAE and in terms of AE.

2.6 Task T2: Single-Label Multiclass Quantification under Prior Probability Shift

Similarly to Task T1, Task T2 is essentially the same as task T1B in LeQua 2022, i.e., it is a single-label multi-class quantification task on data affected by prior probability shift, and was reproposed in LeQua 2024. Aside from the fact that T1 is binary and T2 is multiclass, the two subtasks are very similar.

Task T2 uses 28 topic labels (the same as in T1B of LeQua 2022), representing the merchandise classes to which Amazon products belong to. We have randomly sampled 20,000 reviews from the pool Ω for use as the training set L_2. Following the same protocol adopted for Task T1, we remove the reviews of L_2 from Ω, and from this reduced pool we extract a development set D_2 composed of 1,000 development samples, each composed of 1,000 reviews. Any review appearing in a development sample is then removed from Ω, after which we proceed with the extraction of the U_2 test set, composed of 5,000 test samples consisting of of 1,000 reviews each. As the evaluation measures, here too we compute RAE (which is also used for ranking the systems) and AE. All other choices made in the design of this experimental setting are the same as for T1.

2.7 Task T3: Ordinal Quantification under Prior Probability Shift

Task T3 is about quantification using an ordinal scale, under prior probability shift; this task is new in LeQua 2024, since no ordinal scales were used in LeQua 2022. The quantification task is defined on a 1-star to 5-stars scale based on the scores assigned to the reviews by their authors. The total order relation among the five possible ratings makes this problem different from a single-label multi-class problem, since misassigning a probability mass to a class faraway from the correct class is a more serious mistake than misassigning it to a class closer in the total order.

For this first edition of the task we chose to follow the natural distribution of data in the pool. The training set is thus composed of samples, i.e., sets of reviews; each sample is composed of reviews of the same product. This is different from the training sets of the other tasks, which are just composed of reviews and leave the use of possible sampling strategies for training up to the participants. Obviously, nothing prevents participants to T3 from building different samples from the entire set of reviews contained in the training set of the task.

The training set L_3 is composed of 100 samples, each associated to a specific product; the 100 products were randomly selected from those with at least 200 reviews in Ω. Each sample in L_3 is composed of exactly 200 reviews. For products with more than 200 reviews we randomly sampled 200 reviews using a stratified random selection. This sampling protocol is known as the *natural prevalence protocol* (NPP). In the case of T3 the PPS among the samples is thus originated by the natural difference in the reviews that products of different quality receive. Our choice of the NPP in place of the APP is motivated by a shortage of data, that would prevent a good APP sampling: the selection of 6,100 products with at least 200 reviews ended up with a large portion of products having just a little more than 200 reviews[6], making the APP eventually produce many unrealistic samples mostly composed of duplicate documents.

Samples in L_3 were provided to participants with the star rating of each review. We then removed the 100 products selected for the training set from the pool and all of their reviews, and we identified a set of 1,000 products for the development set D_3. Each development sample is composed of 200 reviews. Being a development set, the participants were provided with the prevalence of star ratings for each sample, and not the star rating of the single reviews. After removing also the products in the development set from the pool, we sample the test set U_3. The test is composed of 5,000 test samples, related to 5,000 different products, each sample consisting of 200 reviews. Participants had no access to labels or prevalence values for samples in U_3, which have been released publicly after the end of the challenge.

The evaluation measure. The evaluation of quantification predictions for T3 requires taking into account the ordinal scale of the labels when comparing the

[6] This is due to a long-tailed distribution in the original pool, in which few products have many reviews, and many products have very few reviews.

true and predicted distributions. A measure derived from the Earth Mover's Distance, the *Normalised Match Distance* (NMD) [44, 50], takes into account the ordinal relations between classes. NMD is defined as

$$\mathrm{NMD}(p_\sigma, \hat{p}_\sigma) = \frac{1}{n-1} \sum_{j=1}^{n-1} d(y_j, y_{j+1}) \cdot |\sum_{i=1}^{j} \hat{p}_\sigma(y_i) - \sum_{i=1}^{j} p_\sigma(y_i)| \qquad (3)$$

where $\frac{1}{n-1}$ is a normalisation factor that allows NMD to range between 0 (best) and 1 (worst), and $d(y_i, y_{i+1})$ is the distance in the ordinal scale among two consecutive labels, which we assume to be always 1. Given all the samples in the test set U_3, each with a true distribution and a predicted distribution, we evaluate NMD for each sample and compute the mean NMD value across all the samples.

In this first edition of T3 we sampled the distributions randomly from the pool, thus replicating in the validation and the test sets the natural unbalance of star-rating distributions towards a high number of stars. By doing a simple mean across all the samples, the more frequent distributions skewed towards a high number of stars give a bigger contribution to NMD. For this reason we evaluate also a macro version of NMD (Macro-NMD). We define $n-1$ bins, one of each interval from 1 to n, assigning each sample to the bin corresponding to the average of the ordinal labels in the sample. The NMD value is computed for each bin separately, and the Macro-NMD is the mean the resulting $n-1$ NMD values. In this way the Macro-NMD gives to the whole spectrum of mean ratings an equal relevance, regardless of how the samples are distributed.

2.8 Task T4: Binary Quantification under Covariate Shift

Another novel task for LeQua is the binary quantification under covariate shift. Any task presented so far in LeQua 2022, and the other tasks of LeQua 2024 are concerned only with PPS, i.e., $P(Y) \neq Q(Y)$, while this novel challenge adds also covariate shift, i.e., $P(X) \neq Q(X)$.

The quantification task of T4 is the same of T1, with the fundamental difference in the sampling process that generates the various sets. The training set L_4 is composed of 5,000 reviews. Reviews are sampled from the a restricted pool that includes only reviewer for products in the Books or in the Electronics categories. We use these two domains to simulate covariate shift because they are the biggest ones. We consider as negative the reviews with one or two stars, and as positive those with four or five stars, discarding the reviews with three stars. The sampling for L_4 is composed of 90% of Books reviews and 10% Electronics reviews. We chose this ratio to be able to simulate a broad range of covariate shift in the development and test samples by varying the ratio of sampling among the two categories. For the training set the sampling is stratified with respect to the sentiment labels, i.e., we replicate the natural distribution of labels from the sampling pool. The development set D_4 is composed of 1,000 development samples, each one composed of 250 reviews. The sampling pool of D_4 is the one of L_4 minus the reviews already included in L_4. The sampling

Table 2: The teams who participated in LeQua 2024 and the tasks for which they submitted runs.

	T1	T2	T3	T4
Lamarr	x	x	x	x
UniOeste	x			
TeamCUFE	x	x	x	x
UniOviedo(Team1)		x	x	x
UniOviedo(Team2)			x	
UniLeiden	x			
UNSW	x	x		

strategy of development samples uses APP on sentiment labels and also on the two categories. APP on sentiment labels generates PPS, while APP on the two categories generates covariate shift. For example, a sample can be created setting a 80%-20% distribution on sentiment labels and a 40%-60% distribution among categories. In this case, 32% of the reviews will be sampled from positive reviews in Books, 48% from positive reviews in Electronics, 8% of negative reviews in Books, and the remaining 12% from negative reviews in Electronics. The test set U_4 is generated in the same way of D_4. The test set samples are 5,000, each one composed of 250 reviews.

Since this is a binary quantification task, the evaluation measures we use are the same as in T1, i.e., RAE (our official evaluation measure for ranking the systems), and AE.

2.9 Preventing data reuse across tasks

All the tasks use the same pool of documents to sample from. The training data and the development data of each task differs from the one of the other tasks, but they are closely related. This holds specially for reviews with sentiment labels or star ratings in tasks T1, T2, and T4. In this scenario, a possible optimisation strategy could be merging all the training data in order to improve a sentiment classifier accuracy, which can give a sensible boost also to quantification accuracy. This and similar reuse of data across tasks does not add any useful contribution to the knowledge on the quantification problems and methods. We prevented this by using for each task a different random shuffle of the dimensions of the vectors produced by embedding model, so that the vectors are informative only within the task they belong.

3 The participating systems

Seven teams submitted runs to LeQua 2024. As shown in in Table 2, there is a quite balanced participation across all the tasks, with a minimum of three participants in T4, and a maxium of five partipants in T1 and T3. This is a significant difference from LeQua 2022, when most of the participants focused on T1A (now T1). Two teams, Lamarr and TeamCUFE, participated to all four Tasks. We here list the teams in alphabetical order:

– **Lamarr** [28] submitted a run each for all four tasks. All their runs are based on solving an optimisation problem based on the negative log-likelihood loss proposed by [1], optimising a latent representation that is then passed through a softmax to convert it to a probability distribution. The loss has a regularisation component added that promotes distributions tending towards uniformity, with a specific version customised for T3 that promotes a smooth transition of distribution values across the ordinal scale. The contribution of the regularisation component is controlled by a parameter. The posteriors given as input to the optimisation process are obtained training a multi layer perceptron for T1, T2, and T3, and a Logistic Regressor for T4. All the hyperparameters of the classifiers and the one of the regularisation loss have been optimised running a grid search, evaluated on the validation data.
– **UniOeste** [29] submitted a run for T1. The idea is pretty straightforward. The system consists of an ensemble of several binary quantifiers. Different well-known quantification systems from the literature are used, including DyS, HDy, and SLD (among many others), and different classifiers are trained at the basis, including XGBoost, CatBoost, Random Forest, and SVMs. Each quantifier issues a prediction for the test bag and the predictions are then ranked based on the quantifiers' performance. Only the output of the top-performing quantifiers are used to produce the final estimation, which is obtained by averaging the class prevalence predictions of each member of the committee.
– **TeamCUFE** submitted a run each for all four tasks. This team did not give a description of their methods, and thus they cannot be included in the description of results.
– **UniOviedo(Team1)** [40] submitted a run each for T2, T3, and T4. They employed a deep learning method that relies on a novel (permutation-invariant) pooling layer, which models the distribution of bag instances in a latent space as a mixture of Gaussian distributions with learnable mean and covariance matrices. The network uses parallel pooling layers of this type and enhances their combined utility by regularizing them towards minimal Centered Kernel Alignment (CKA). This method follows the "symmetric" approach, where training instances are bags labelled by prevalence (rather than individual data items labelled by class), thus functioning as a bag-based regressor. As such, the network can be trained with specific error metrics in mind. The authors optimised the network for the official evaluation metrics used in each of the tasks they participated in (RAE for T2, T4, and NMD for T4). One of the key differences compared to most other participant teams is that UniOviedo(Team1) utilised part of the validation samples not only for model selection but also for training the model. Additionally, they

also applied some data augmentation heuristics for increasing the number of training bags.

- **UniOviedo(Team2)** submitted a run for T3. Although the participating team did not submit a notebook description of their method, the members (consisting of David Pérez Román and Juan José del Coz from the University of Oviedo) have informed us that the method they applied correspond to their implementation, as made available in the `QuantificationLib` package [6], of the method EDy [8]. EDy belongs to the distribution-matching family of methods and is a variant of the original Energy Distance method proposed by [25]. EDy relies on the Earth Mover Distance (EMD – also known as the Wasserstein loss) as the divergence measure, which is particularly well-suited for ordinal problems, as is the case of T3.
- **UniLeiden** [26] submitted a run for T1. This team used the Continuous Sweep method [27], using an optimised SVM with RBF kernel as the base classifier. They made a comparison on validation data against three other quantifiers, i.e., Median Sweep [20], SLD [42], and DyS [30]. They found that Sweep-based methods performed better than the other methods when the underlying classifier performed poorly, and vice-versa, indicating a link with the results of [45] in which Median Sweep method performed better than SLD and DyS on datasets with smaller training sets (and thus with a likely lower performance of the classifier) than those used in LeQua.
- **UNSW** [11] participated in T1 and T2. This team proposed two ensemble methods for these tasks. For T1, they proposed a Multiple Classifiers - Single Quantifier method (MC-SQ). This method uses an ensemble in which all the members are instances of the same aggregative quantifier (they used DyS [30]) equipped with different classifiers. The classification algorithms used to form the ensemble include Logistic Regression, Linear Discriminant Analysis, Support Vector Machines, Light Gradient Boosting Machines, Gradient Boosting, and CatBoost. The simple rationale of the method is to have many different classifier which are all known to be good overall performers and to exploit the strength of the ensemble to filter out the cases in which some of them may perform worse. Each classifier-quantifier pair in the ensemble was subjected to a joint hyperparameter optimisation. This means that each DyS instance in the ensemble has its own optimised number of bins that is dependent on the specific classifier the quantifier is paired with. Given a test sample, all the quantifiers in the ensemble make their predictions, and the median value is taken as the final prediction of the MC-SQ method. For T2, UNSW followed an approach that is the opposite of T1: a single classification algorithm paired with many different quantifiers, i.e., a Single-Classifier - Multiple Quantifiers (SC-MQ) method. In this case the authors identified Logistic Regression as the most stable and best performing classifier, and decide to evaluate the ensemble approach varying the quantification methods. They use four quantification methods: Energy Distance (EDy) [25], Kernel Density Estimation (KDEy) [36], Generalized Probabilistic Adjusted Classify & Count (GPACC) [18], and a newly proposed EMQ-ini method. The EMQ-ini method is a variant of EMQ [42] that uses the priors

from GPACC as the initial priors for the unlabelled set, instead of directly using the priors from the classifier. Also for T2 the optimisation of each classifier-quantifier pair in the ensemble is performed jointly.

4 Results

In this section we discuss the results obtained by the participant teams in the four subtasks we have proposed. The evaluation campaign started on February 15, 2024, with the release of the training sets ($L_1 \dots L_4$) and of the development sets ($D_1 \dots D_4$); alongside them, the participant teams were provided with a dummy submission, a format checker, and the official evaluation script.[7] The unlabelled test sets ($U_1 \dots U_4$) were released on May 1, 2024; and runs had to be submitted by June 15, 2024.

We used Codalab (`https://codalab.org/`) as the platform for the submission of runs by the teams; each team could submit up to three runs per subtask. The official competition can be accessed at.[8] In this edition we set up a second Codalab instance using the same validation set provided to them. The validation version is available at.[9] This second instance allowed teams to have an immediate evaluation of their methods on validation data, allowing them to check the correctness of their submissions and the consistence of their evaluations with the one performed by the official evaluation platform.

The true labels of the unlabelled test sets were released on May 3, 2024, after the submission period was over and the official results had been announced to the participants. In the rest of this section we discuss the results that the participants' systems and the baseline systems have obtained in the Binary Quantification task (T1, Section 4.1), the Single-Label Multi-Class Quantification task (T2, Section 4.2), the Ordinal Quantification task (T3, Section 4.3), and the Covariate Shift task (T4, Section 4.4),

In the sections to come, we use the following notational conventions for the tables displaying the results of the participant teams. The first column corresponds to the official measure used for ranking the participant systems (RAE in T1, T2, and T4; NMD in T3) while the second column displays the secondary evaluation measure (AE in T1, T2, and T4; Macro-NMD in T3). Results are averaged across the 5,000 test samples. **Boldface** indicates the best method for a given evaluation measure. Superscripts † and ‡ denote the methods (if any) whose scores are *not* statistically significantly different from the best one according to the Wilcoxon signed-rank test at different confidence levels: symbol † indicates $0.001 < p$-value < 0.05 while symbol ‡ indicates $0.05 \leq p$-value. The absence of any such symbol indicates p-value ≤ 0.001 (i.e., that the difference in performance between the method and the best one is statistically significant at a high confidence level). Baseline methods are typeset in *italic*.

[7] `https://github.com/HLT-ISTI/LeQua2024_scripts/tree/main`
[8] `https://codalab.lisn.upsaclay.fr/competitions/18965`
[9] `https://codalab.lisn.upsaclay.fr/competitions/19100`

4.1 Task T1: Binary Quantification under Prior Probability Shift

Rank	Run	RAE	AE
1	UNSW	**0.09811** ± **0.27043**	0.02063‡± 0.01608
2	*KDEy*	0.10179‡± 0.30431	**0.02043** ± **0.01589**
3	Lamarr	0.10653†± 0.31885	0.02128 ± 0.01667
4	*DM*	0.10699 ± 0.28473	0.02175 ± 0.01669
5	UniOeste	0.10850‡± 0.35492	0.02096†± 0.01648
6	*SLD*	0.11103‡± 0.36698	0.02113 ± 0.01660
7	*PACC*	0.13390 ± 0.46333	0.02399 ± 0.01809
8	UniLeiden	0.13917 ± 0.53773	0.02379 ± 0.01818
9	*ACC*	0.16439 ± 0.60318	0.02644 ± 0.02037
10	*CC*	0.97742 ± 3.91905	0.07955 ± 0.04816
11	*PCC*	1.26562 ± 5.11243	0.10175 ± 0.05985
12	TeamCUFE	2.53730 ± 10.82087	0.22472 ± 0.15197

Table 3: Results of Task T1, binary quantification under prior probability shift.

Table 3 shows the results of the participating teams in T1. The team obtaining the best averaged result is UNSW. In this task, UNSW used a variant of an ensemble method called MC-SQ that combines the output of different classifiers with one aggregative quantifier (see Section 3). This method is not only the one scoring the lowest RAE, but also the one showcasing the smallest variance of the lot. Notwithstanding this, the differences in performance with respect to four other methods (KDEy, Lamarr, UniOeste, and SLD) are not statistically significant according to the statistical test.

In terms of AE, the best performing method is KDEy. This may come as a surprise, as KDEy was originally proposed with multiclass problems in mind, and is not expected to bring to bear any significant advantage in the binary case. Be it as it may, the differences in performance with respect to UNSW (MC-SL) and UniOeste are not significant.

The four CC-variants are relegated to the bottom half of the results table. The UniLeiden's system performance is positioned between that of the baseline methods PACC and ACC in the results table. This is significant, since UniLeiden proposes a variant of the Medium Sweep algorithm, which in turn is an improved variant of ACC. Despite the improvement brought to bear by UniLeiden, the original PACC still seems to perform slightly better in terms of RAE, while at the same time displaying a smaller variance. As a final remark, the (bad) performances of the "unadjusted" variants (CC and PCC) are not comparable with the rest of the methods in the table, yielding errors close to one order of magnitude higher. TeamCUFE's system produces even higher errors, both for RAE and AE, and with much higher variation.

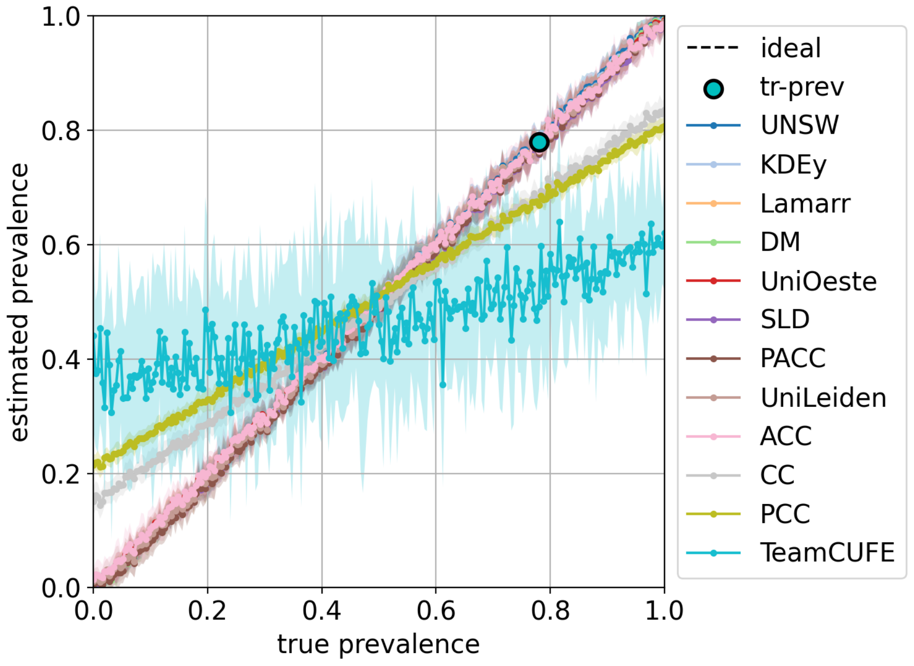

Fig. 1: Diagonal plot for T1.

Figure 1 shows a so-called "diagonal plot" which displays the estimated prevalence value for the positive class (y-axis) as a function of the true positive prevalence (x-axis). The plot is termed "diagonal" because the behaviour of an ideal quantifier is represented by the diagonal line from (0,0) to (1,1). This plot reveals that most of the methods perform fairly well at predicting the positive class prevalence. Exceptions are CC and PCC. The reason for this is that neither of these methods applies any correction to the raw counts obtained from a hard or soft classifier, respectively. Interestingly, CC and PCC do not intersect the diagonal at the point where the true prevalence equals the training prevalence (marked on the plot as a cyan dot), which would be expected since CC and PCC are known to be biased towards the training prevalence. The reason for this deviation is that the classifier's hyperparameters were optimized through model selection on the validation samples, which are designed to exhibit significant variations in prior probability shifts. As a result, the best hyperparameters are those that manage to shift the classifier's bias from the training prevalence to 0.5, which yields a lower averaged RAE across all validation samples. The most noisy output is attained by TeamCUFE. Unfortunately, we do not have details on how this method performs.

Figure 2 shows a different plot that we may dub the "error-by-shift" plot, in which the error (here displayed as log(RAE) to better highlight performance

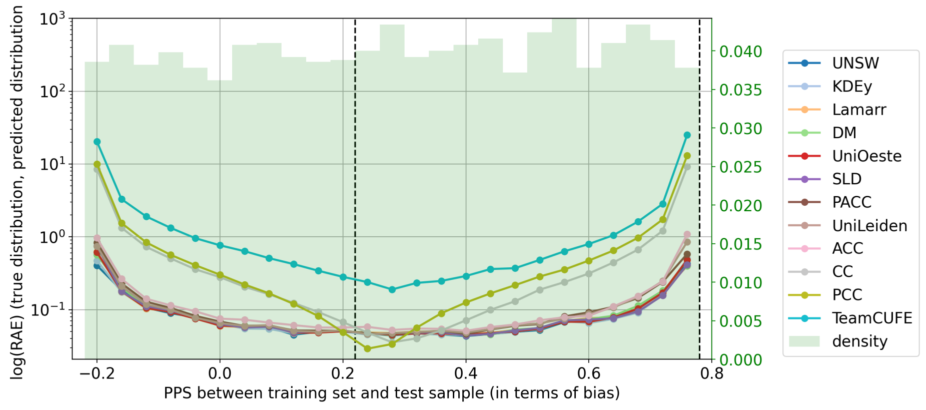

Fig. 2: Error-by-shift plot for T1.

differences) is shown as a function of the amount of PPS of the test samples with respect to the training sample. Here, we measure PPS in terms of the signed difference between the training positive prevalence and the test positive prevalence. The greened region represents the density of experiments carried out, which is close to uniform. Something which seems evident from the plot is that most of the methods yield very small errors across the majority of the shift spectrum, with higher errors concentrated at the extremes of the spectrum, i.e., where all instances are either positive or negative. CC, PCC, and TeamCUFE achieve the smallest error at around 0.25 of shift, rather than at 0 (no shift), as one might expect. This aligns with the previous observation that, through model selection, these methods have shifted their inherent biases from the training prevalence (0.78) to 0.5 in order to minimize the validation error. Consequently, these methods happen to be biased towards the point 0.78-0.5 $\approx$ 0.25, where they obtain the best results. For the rest of the methods, the differences in performance are very thin.

4.2 Task T2: Single-Label Multiclass Quantification under Prior Probability Shift

Table 4 shows the results obtained by the participating teams in T2. UniOviedo(Team1) obtained the best RAE score in the multiclass quantification problem under prior probability shift. This score is not only the smallest, but also the one displaying the smallest variation. The Wilcoxon test reveals this score is statistically significantly better than the rest of the methods with high confidence. In terms of AE, instead, the UNSW team obtained the best averaged score, which is statistically significantly better than the rest of the methods. In this case, the variant employed by the UNSW team corresponds to the a configuration SC-MQ, i.e., to one in which there is only one classifier generating predictions for an ensemble of aggregation methods (see Section 3).

Rank	Run	RAE	AE
1	UniOviedo(Team1)	**0.92173** ± **0.70476**	0.02097 ± 0.00566
2	Lamarr	1.03016 ± 0.84658	0.01412 ± 0.00322
3	UNSW	1.07856 ± 0.96585	**0.01274** ± **0.00353**
4	*SLD*	1.16158 ± 0.99066	0.01343 ± 0.00346
5	*PACC*	1.19418 ± 1.13479	0.01552 ± 0.00424
6	*KDEy*	1.20166 ± 1.05091	0.01367 ± 0.00367
7	*DM*	1.27189 ± 1.09683	0.01578 ± 0.00405
8	*ACC*	1.34787 ± 1.16063	0.01640 ± 0.00427
9	*CC*	2.30963 ± 1.38323	0.01660 ± 0.00310
10	*PCC*	2.67505 ± 1.60472	0.01931 ± 0.00337
11	TeamCUFE	4.02872 ± 2.12809	0.02587 ± 0.00334

Table 4: Results of Task T2, single-label multiclass quantification under prior probability shift.

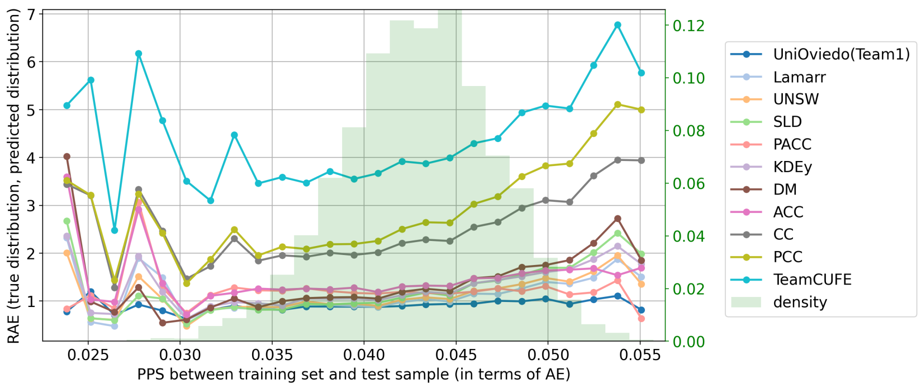

Fig. 3: Error-by-shift plot for T2.

Figure 3 displays the "error-by-shift" plot of T2. In this case, we measure the amount of shift in terms of absolute error (since the signed difference we used in Figure 2 is only defined in the binary case). This plot reveals some interesting facts. First, that most methods degrade their performance at increasing levels of PPS. UniOviedo(Team1) seems to be the most robust in this respect, though. Second, the performance of all methods seems erratic at very high and (specially) very smaller amounts of PPS. However, the density of experiments (greened background) tells us that the number of experiments involved in such cases is very small, which explains the higher variability. This plot also reveals that the distribution of the "amount of shift" generated via APP is close to normal; something that was already echoed in previous research [38].

Rank	Run	NMD	Macro-NMD
1	UniOviedo(Team1)	**0.06438 ± 0.04773**	$0.08007^{\ddagger}$ ± 0.02397
2	Lamarr	0.06585 ± 0.04706	**0.07878 ± 0.02038**
3	*PCC*	0.06680 ± 0.05069	$0.09004^{\ddagger}$ ± 0.03532
4	*KDEy*	0.06904 ± 0.04982	$0.08297^{\ddagger}$ ± 0.01993
5	UniOviedo(Team2)	0.07212 ± 0.05310	$0.08771^{\ddagger}$ ± 0.02564
6	*CC*	0.07974 ± 0.04760	$0.08820^{\ddagger}$ ± 0.01524
7	TeamCUFE	0.10726 ± 0.07755	$0.18647^{\ddagger}$ ± 0.11300
8	*DM*	0.10939 ± 0.06160	$0.11445^{\ddagger}$ ± 0.01200
9	*SLD*	0.11107 ± 0.06903	$0.10947^{\ddagger}$ ± 0.01220
10	*ACC*	0.11944 ± 0.06496	$0.12473^{\ddagger}$ ± 0.01498
11	*PACC*	0.12363 ± 0.06522	$0.12830^{\ddagger}$ ± 0.01189

Table 5: Results of Task T3, ordinal quantification under prior probability shift.

4.3 Task T3: Ordinal Quantification under Prior Probability Shift

Table 5 shows the results of the participating teams in T3. Also in this case, UniOviedo(Team1) achieved the best result for the official evaluation metric, which in this case is NMD since this is an ordinal problem. The score obtained by UniOviedo(Team1) is statistically significantly better than the rest of the participating systems. However, in terms of Macro-NMD, the Lamarr team achieved the best result. Nevertheless, according to the Wilcoxon test, no method appears to be statistically significantly different in terms of average rank performance.

It was expected that UniOviedo(Team1) and Lamarr would perform well in this task, as both teams implemented solutions that take into account the ordinal nature of the data. For example, UniOviedo(Team1) directly optimised for the evaluation loss, while Lamarr explicitly regularised their solutions towards a uniform distribution, which ultimately favours smooth solutions. What was unexpected, however, was the seemingly good performance of PCC, a method that not only disregards the ordinal nature of the data but also does not attempt to counter any shift in the priors. This is a clear indication that the samples provided may be characterized by a lower degree of prior probability shift compared to tasks T1 and T2. The reason why, is that the samples are "natural" (generated via NPP), i.e., are not generated via artificially (via APP) imposing widely varying degrees of prior probability shift, as was instead the case for T1 and T2.

Figure 4 displays the averaged performance of the top-5 methods as a function of different levels of "jaggedness" of the tested distributions. More precisely, we compute the jaggedness of a distribution p_σ as:

$$J(p_\sigma) = \frac{1}{2} \sum_{i=2}^{n-1} \left(-p_\sigma(y_{i-1}) + 2p_\sigma(y_i) - p_\sigma(y_{i+1})\right)^2 \tag{4}$$

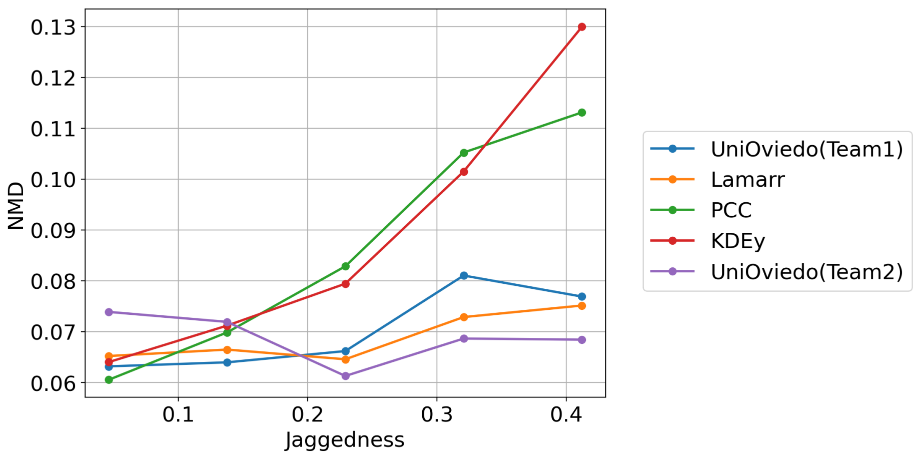

Fig. 4: Quantification performance as a function of the distribution "jaggedness".

The plot reveals that all methods degrade their performance as the distributions become "jaggy". However, UniOviedo(Team1), Lamarr, and UniOviedo(Team2) seem to behave more robustly across the entire spectrum. That Lamarr and UniOviedo(Team2) performed well in T3 was to be expected, as both methods have been designed with ordinal considerations in mind. Also, UniOviedo(Team1) performed well in T3, despite the fact that this method is not specifically suited for ordinal data. Notwithstanding this, note that UniOviedo(Team1) is trained using the validation data as well, which inherently showcase many plausible patterns of ordinal distributions, so one might expect that this method learns to handle the ordinality which resides in the distributions. In contrast, other methods such as PCC and KDEy, which are completely agnostic to the ordinal nature of the data, tend to perform worse in difficult scenarios. The fact that, notwithstanding this, PCC and KDEy rank second and third in the table is a consequence of the fact that most of the samples display low levels of jaggedness.

4.4 Task T4: Binary Quantification under Covariate Shift

Table 6 reports the results obtained for T4. In this case, the Lamarr team scored the best result in terms of the official evaluation measure (RAE), while SLD obtained the best AE. Somehow surprisingly, though, the RAE score obtained by UniOviedo(Team1) is not statistically significantly different from the Lamarr's score, even though UniOviedo(Team1) ranked 5th. This may be explained by the variance of their system, which is markedly higher than that of methods SLD, DM, and KDEy, which occupy the 2nd, 3rd, and 4th positions in the rank, respectively.

As recalled from Section 2.8, task T4 is not only characterized by covariate shift but also by a shift in the priors. It is thus interesting to disentangle the

Rank	Run	RAE	AE
1	Lamarr	**0.10930 ± 0.33394**	0.02191 ± 0.01765
2	*SLD*	0.11497 ± 0.39872	**0.02013 ± 0.01594**
3	*DM*	0.11559 ± 0.32287	0.02341 ± 0.01853
4	*KDEy*	0.11804 ± 0.32916	0.02380 ± 0.01865
5	UniOviedo(Team1)	0.12975‡ ± 0.42556	0.02122 ± 0.01617
6	*ACC*	0.26187 ± 1.09686	0.03054 ± 0.02454
7	*PACC*	0.28918 ± 1.33046	0.03078 ± 0.02480
8	*CC*	1.11975 ± 4.33704	0.08292 ± 0.04915
9	*PCC*	1.45553 ± 5.63934	0.10735 ± 0.06316
10	TeamCUFE	2.49902 ± 9.93390	0.23017 ± 0.16531

Table 6: Results of Task T4, binary quantification under covariate shift.

systems' performance in terms of both types of shifts, separately. Figure 5 displays the performance of the methods (on logarithmic scale) as a function of the amount of prior shift. The trends we observe are, by and large, in line with those of T1. Figure 6 instead displays the distribution of the errors for the top-5 methods as a function of the prevalence of the Books domain in the test samples, therefore effectively reflecting the amount of covariate shift with respect to the training distribution. This plot shows a weak tendency to improve as the prevalence of Books in the test samples increases, thereby approximating the training mixture (made of 90% Books and 10% Electronics) and reducing the amount of covariate shift. This tendency is more evident for Lamarr, the best performer system for this task. One possible reason why the covariate shift has a weak effect on the results may be that the embeddings were generated using the ELECTRA-Small model [10], which is specifically trained to capture sentiment polarity, rather than topical information, which may therefore be obscured.

Interestingly, PCC has obtained a poor score. This is relevant since this method is considered to behave robustly in the presence of covariate shift. However, it is also known that its performance degrades when some shift in the priors is also at play [24], as is the case for T4. Figure 7 compares the performance of PCC against Lamarr, the top performer in Table 6, proving that PCC's performance in terms of RAE is out of scale.

Figure 8 reports the diagonal plot for T4. Interestingly enough, most methods seem to perform very well notwithstanding the fact that the underlying distributions are affected by covariate shift as well. As witnessed for T1, CC, PCC, and TeamCUFE struggle to obtain good predictions for the entire spectrum, since these methods seem to be strongly biased toward the center of the distribution.

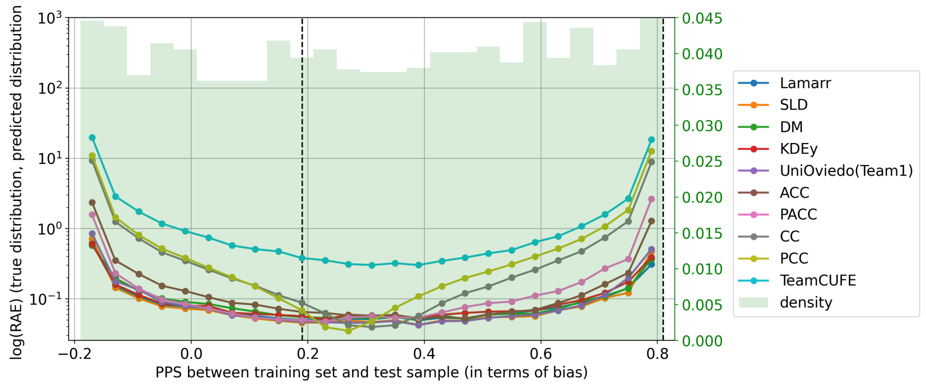

Fig. 5: Error-by-shift plot for T4.

5 Final remarks

In this edition of LeQua, we have observed several tendencies worth remarking. First, that the "symmetric approach" adopted by UniOviedo(Team1) shows promising performance across many different scenarios and involving differently characterised types of dataset shift. This symmetric approach regards the problem as a regression one, in which both the training instances and the test instances are composed by bags of individual datapoints. Such an approach thus requires many bags, each labelled with its class prevalence values. The team used (among other things) the validation samples otherwise provided for model selection. Given that such samples were representative of the type of shift at play in each task, the model was able to learn the particularities of each problem. UniOviedo(Team1) thus leveraged more information during the training phase than the competitors; however, this strategy is not unfair, as the validation samples were made available to all participation teams. Yet another differentiating aspect of this method has to do with its ability to optimise specific loss functions. This capability allowed UniOviedo(Team1) to tune their models for the very same evaluation measures used to rank the participating systems. Perhaps the most important differentiating aspect of this method is the fact that no classifier is involved. This appears promising from the point of view of the Vapnik's principle (outlined in Section 1), since the method is trained to solve the quantification problem *directly*.

One notable observation is that the binary quantification problem (exemplified by T1) is almost a "solved problem", with most methods performing remarkably well in this case. In contrast, the multiclass quantification problem remains significantly more challenging, leaving substantial room for improvement.

Compared to the previous edition, we observed many methods clearly outperforming SLD. This is noteworthy, since SLD is widely considered a very hard-to-beat system [1,37,38] and was one of the top-performing methods in the past LeQua 2022 edition.

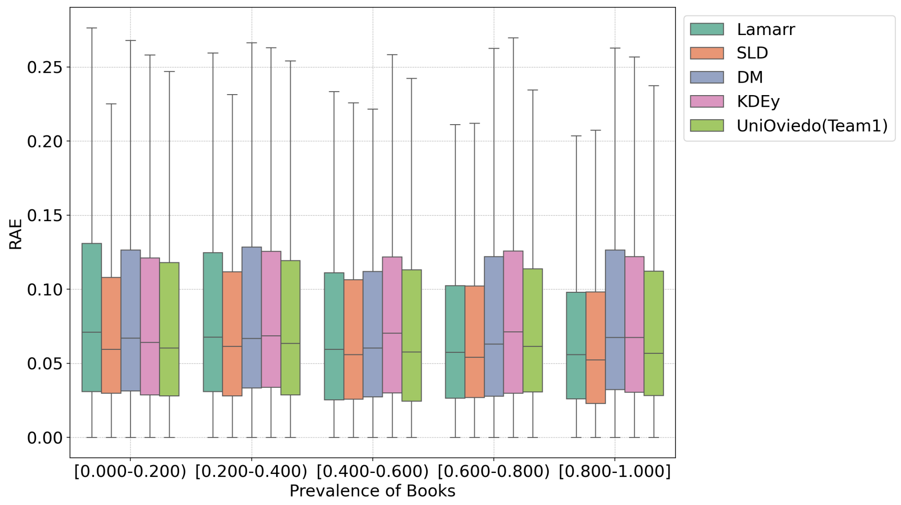

Fig. 6: Errors of the top-5 methods at different proportions of book and electronics domains.

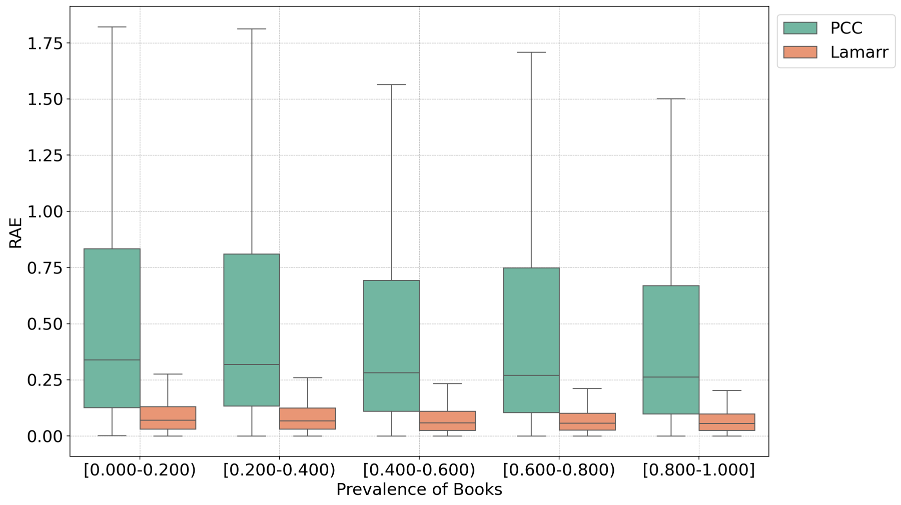

Fig. 7: A comparison of the errors produced by PCC and the top performer method Lamarr, at different proportions of book and electronics domains.

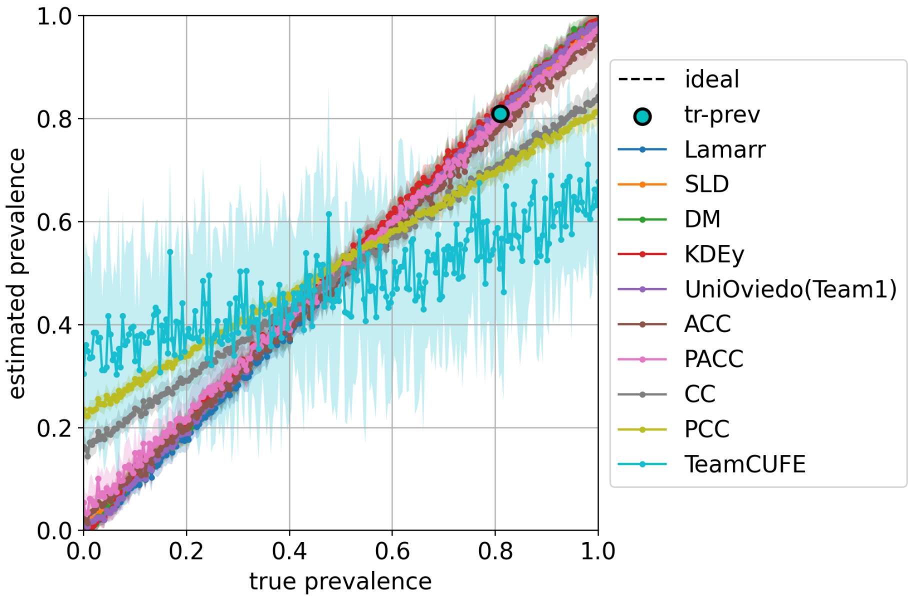

Fig. 8: Diagonal plot for T4.

Another important point concerns the difficulty in simulating meaningful levels of prior shift in ordinal problems. Plausible ordinal distributions impose certain constraints (e.g., smoothing requirements) that conflict with protocols designed to artificially alter them. In future editions, we plan to explore more sophisticated mechanisms to challenge participant systems with more abrupt shifting conditions in ordinal problems.

Looking back, we believe that using a sentiment-specific feature extractor reduced the impact of the covariate-shift effect introduced in task T4. For future editions, we plan to incorporate alternative representation mechanisms that better account for the controlled mixture of domains.

To conclude, we believe that LeQua 2024 has provided participating teams with the opportunity to stress-test their cutting-edge systems in a controlled setting, offering valuable insights to the community.

Acknowledgments

This work has been supported by the SoBigData++ project, funded by the European Commission (Grant 871042) under the H2020 Programme INFRAIA-2019-1, and by the SoBigData.it and FAIR projects funded by the European Commission under the NextGenerationEU program. This work has also been funded by the QuaDaSh project (P2022TB5JF) “Finanziato dall’Unione europea —Next Generation EU, Missione 4 Componente 2 CUP B53D23026250001”.

The authors' opinions do not necessarily reflect those of the European Commission.

References

1. Alexandari, A., Kundaje, A., Shrikumar, A.: Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In: Proceedings of the 37th International Conference on Machine Learning (ICML 2020). pp. 222–232. Virtual Event (2020)
2. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Quantification via probability estimators. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2010). pp. 737–742. Sydney, AU (2010). https://doi.org/10.1109/icdm.2010.75
3. Bunse, M., Moreo, A., Sebastiani, F., Senz, M.: Regularization-based methods for ordinal quantification. arXiv:2310.09210 [cs.LG] (2023)
4. Bunse, M., Morik, K.: Unification of algorithms for quantification and unfolding. In: Proceedings of the 2nd International Workshop on Learning to Quantify (LQ 2022). pp. 1–10. Grenoble, IT (2022)
5. Card, D., Smith, N.A.: The importance of calibration for estimating proportions from annotations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2018). pp. 1636–1646. New Orleans, US (2018). https://doi.org/10.18653/v1/n18-1148
6. Castaño, A., Alonso, J., González, P., Pérez, P., del Coz, J.J.: QuantificationLib: A python library for quantification and prevalence estimation. SoftwareX **26**, 101728 (2024)
7. Castaño, A., Alonso, J., González, P., del Coz, J.J.: An equivalence analysis of binary quantification methods. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI-23). pp. 6944–6952. Washington, US (2023)
8. Castaño, A., González, P., González, J.A., del Coz, J.J.: Matching distributions algorithms based on the Earth Mover's Distance for ordinal quantification. IEEE Transactions On Neural Networks and Learning Systems (2022). https://doi.org/10.1109/TNNLS.2022.3179355, forthcoming
9. Castaño, A., González, P., González, J.A., del Coz, J.J.: 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 (2024). https://doi.org/10.1109/TNNLS.2022.3179355
10. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training text encoders as discriminators rather than generators. In: Proceedings of the 8th International Conference on Learning Representations (ICLR 2020). Addis Ababa, ET (2020), `https://openreview.net/pdf?id=r1xMH1BtvB`
11. Donyavi, Z., Li, F., Batista, G.: Ensemble Learning to Quantify: The CSE UNSW Team at LeQua 2024. In: Working Notes of the Learning to Quantify: Methods and Applications (LQ 2024) workshop, co-located at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2024), Vilnius, Lituania. Vilnius, LI (2024)
12. Donyavi, Z., Serapio, A., Batista, G.: MC-SQ: A highly accurate ensemble for multi-class quantification. In: Proceedings of the 23rd SIAM International Conference on Data Mining (SDM 2023). pp. 622–630. Minneapolis, US (2023). https://doi.org/10.1137/1.9781611977653.ch70

13. Dussap, B., Blanchard, G., Chérief-Abdellatif, B.: Label shift quantification with robustness guarantees via distribution feature matching. In: Proceedings of the 34th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML / PKDD 2023). pp. 69–85. Torino, IT (2023). https://doi.org/10.1007/978-3-031-43424-2_5
14. Esuli, A., Fabris, A., Moreo, A., Sebastiani, F.: Learning to quantify. Springer Nature, Cham, CH (2023). https://doi.org/10.1007/978-3-031-20467-8
15. Esuli, A., Molinari, A., Sebastiani, F.: A critical reassessment of the Saerens-Latinne-Decaestecker algorithm for posterior probability adjustment. ACM Transactions on Information Systems **39**(2), Article 19 (2021). https://doi.org/10.1145/3433164
16. Esuli, A., Moreo, A., Sebastiani, F., Sperduti, G.: A concise overview of LeQua 2022: Learning to quantify. In: Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022). pp. 362–381. Bologna, IT (2022). https://doi.org/10.1007/978-3-031-13643-6_23
17. Esuli, A., Moreo, A., Sebastiani, F., Sperduti, G.: 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 (2022)
18. Firat, A.: Unified framework for quantification (2016), arXiv:1606.00868v1 [cs.LG] 2 Jun 2016
19. Forman, G.: Counting positives accurately despite inaccurate classification. In: Proceedings of the 16th European Conference on Machine Learning (ECML 2005). pp. 564–575. Porto, PT (2005). https://doi.org/10.1007/11564096_55
20. Forman, G.: BNS feature scaling: An improved representation over TF.IDF for SVM text classification. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008). pp. 263–270. Napa Valley, US (2008)
21. Forman, G.: Quantifying counts and costs via classification. Data Mining and Knowledge Discovery **17**(2), 164–206 (2008). https://doi.org/10.1007/s10618-008-0097-y
22. González, P., Castaño, A., Chawla, N.V., del Coz, J.J.: A review on quantification learning. ACM Computing Surveys **50**(5), 74:1–74:40 (2017). https://doi.org/10.1145/3117807
23. González, P., Moreo, A., Sebastiani, F.: Binary quantification and dataset shift: An experimental investigation. Data Mining and Knowledge Discovery **38**(4), 1670–1712 (2024). https://doi.org/10.1007/s10618-024-01014-1
24. González, P., Moreo, A., Sebastiani, F.: Binary quantification and dataset shift: an experimental investigation. Data Mining and Knowledge Discovery pp. 1–43 (2024)
25. Kawakubo, H., Du Plessis, M.C., Sugiyama, M.: Computationally efficient class-prior estimation under class balance change using energy distance. IEICE Transactions on Information and Systems **99**, 176–186 (2016)
26. Kloos, K.: UniLeiden at LeQua2024: Evaluating Continuous Sweep and Comparison Using Underlying Classifiers. In: Working Notes of the Learning to Quantify: Methods and Applications (LQ 2024) workshop, co-located at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2024), Vilnius, Lituania. Vilnius, LI (2024)
27. Kloos, K., Karch, J.D., Meertens, Q.A., de Rooij, M.: Continuous Sweep: An improved, binary quantifier. arXiv:2308.08387 [stat.ML] (2023)

28. Lotz, T., Bunse, M.: Lamarr at LeQua2024: Regularized Soft-Max Likelihood Maximization. In: Working Notes of the Learning to Quantify: Methods and Applications (LQ 2024) workshop, co-located at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2024), Vilnius, Lituania. Vilnius, LI (2024)
29. Luth, L., Daniel, O., Gomes, G., Maletzke, A.: UniOeste at LeQua 2024: Combining the top-ranked quantifiers. In: Working Notes of the Learning to Quantify: Methods and Applications (LQ 2024) workshop, co-located at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2024), Vilnius, Lituania. Vilnius, LI (2024)
30. Maletzke, A., Moreira dos Reis, D., Cherman, E., Batista, G.: DyS: A framework for mixture models in quantification. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019). pp. 4552–4560. Honolulu, US (2019). https://doi.org/10.1609/aaai.v33i01.33014552
31. Moreno-Torres, J.G., Raeder, T., Alaíz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012). https://doi.org/10.1016/j.patcog.2011.06.019
32. Moreo, A., Esuli, A., Sebastiani, F.: QuaPy: A Python-based framework for quantification. In: Proceedings of the 30th ACM International Conference on Knowledge Management (CIKM 2021). pp. 4534–4543. Gold Coast, AU (2021). https://doi.org/10.1145/3459637.3482015
33. Moreo, A., Francisco, M., Sebastiani, F.: Multi-label quantification. ACM Transactions on Knowledge Discovery and Data **18**(1), Article 4 (2023). https://doi.org/10.1145/3606264
34. Moreo, A., González, P., del Coz, J.J.: Kernel density estimation for multiclass quantification. arXiv:2401.00490 [cs.LG] (2023). https://doi.org/10.48550/arXiv.2401.00490
35. Moreo, A., González, P., del Coz, J.J.: Kernel density estimation for multiclass quantification (2024), `https://arxiv.org/abs/2401.00490`
36. Moreo, A., González, P., del Coz, J.J.: Kernel density estimation for multiclass quantification. Machine Learning (2024), forthcoming
37. Moreo, A., Sebastiani, F.: Re-assessing the "classify and count" quantification method. In: Proceedings of the 43rd European Conference on Information Retrieval (ECIR 2021). vol. II, pp. 75–91. Lucca, IT (2021). https://doi.org/10.1007/978-3-030-72240-1_6
38. Moreo, A., Sebastiani, F.: Tweet sentiment quantification: An experimental re-evaluation. PLOS ONE **17**(9), 1–23 (September 2022). https://doi.org/10.1371/journal.pone.0263449
39. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 12th Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). pp. 1532–1543. Doha, QA (2014)
40. Pérez-Mon, O., González, P.: UniOvi Team at LeQua 2024: Quantification via Gaussian Latent Space Representations. In: Working Notes of the Learning to Quantify: Methods and Applications (LQ 2024) workshop, co-located at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2024), Vilnius, Lituania. Vilnius, LI (2024)
41. Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D. (eds.): Dataset shift in machine learning. The MIT Press, Cambridge, US (2009). https://doi.org/10.7551/mitpress/9780262170055.001.0001
42. Saerens, M., Decaestecker, C.: Decision-making with unknown priors in supervised classification (2010), unpublished manuscript

43. Saerens, M., Latinne, P., Decaestecker, C.: Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. Neural Computation **14**(1), 21–41 (2002). https://doi.org/10.1162/089976602753284446
44. Sakai, T.: 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). pp. 1073–1076. Ann Arbor, US (2018). https://doi.org/10.1145/3209978.3210073
45. Schumacher, T., Strohmaier, M., Lemmerich, F.: A comparative evaluation of quantification methods. arXiv preprint arXiv:2103.03223 (2021)
46. Sebastiani, F.: Evaluation measures for quantification: An axiomatic approach. Information Retrieval Journal **23**(3), 255–288 (2020). https://doi.org/10.1007/s10791-019-09363-y
47. Smith, N.A., Tromble, R.W.: Sampling uniformly from the unit simplex. Tech. rep., Johns Hopkins University (2004), `https://www.cs.cmu.edu/~nasmith/papers/smith+tromble.tr04.pdf`
48. Tasche, D.: Class prior estimation under covariate shift: No problem? In: Proceedings of the 2nd International Workshop on Learning to Quantify (LQ 2022). pp. 11–26. Grenoble, IT (2022)
49. Vapnik, V.: Statistical learning theory. Wiley, New York, US (1998)
50. Werman, M., Peleg, S., Rosenfeld, A.: A distance metric for multidimensional histograms. Computer Vision, Graphics, and Image Processing **32**, 328–336 (1985)