Ensemble Learning to Quantify: The CSE UNSW Team at LeQua 2024

Zahra Donyavi, Feiyu Li, and Gustavo Batista

University of New South Wales, Sydney, Australia
{z.donyavi, feiyu.li, g.batista}@unsw.edu.au

Abstract. LeQua 2024 is a data challenge to facilitate the comparative evaluation of quantification methods for class-prior estimation, also known as quantification or learning to quantify. The challenge focuses on training predictors, termed "quantifiers," to estimate the relative frequencies of classes within sets of unlabeled data points. Notably, the datasets are affected by class prevalence shifts, exhibiting prevalences in the test set that differ from the training set. We propose two ensemble methods, Multiple Classifiers - Single Quantifier (MC-SQ) and Single Classifier - Multiple Quantifiers (SC-MQ), for binary and multi-class quantification tasks. Additionally, we introduce EMQ-ini, a new variation of the Expectation–Maximization algorithm for Quantification (EMQ) method. This variation uses the predicted target prior from the quantifier Generalized Probabilistic Adjusted Classify & Count (GPACC) as the initial point of log-likelihood maximization. We use EMQ-ini as one of the base quantifiers of SC-MQ. Our MC-SQ method ranked first in Mean Relative Absolute Error (MRAE), the official competition performance measure, and second in Absolute Error (AE) on the binary quantification task. Our SC-MQ method ranked third in MRAE and first in AE for the multi-class quantification task.

Keywords: Prevalence estimation · Target prevalence shift · Quantification.

1 Introduction

Quantification learning is used in many real-world scenarios where the objective is to predict the behavior of groups. It is particularly useful in sentiment analysis, which tracks how overall opinions about products, people, or organizations change over time [1]. Instead of classifying individual behaviors, quantification focuses on estimating the distribution of opinions, providing insights into broader trends in public sentiment.

The simplest quantification approach, known as *Classify & Count* (CC), directly applies classification to quantification problems. However, this method suffers from systematic bias in which the error increases linearly as we approach the more skewed class distributions [2]. To address this, researchers have proposed novel quantification methods to provide more accurate estimates of class distributions in the presence of class prevalence shifts.

LeQua 2024 is a competition that challenges participants to evaluate various techniques for binary, multi-class, and ordinal quantification tasks using real-world Amazon product review datasets. The challenge encompasses four tasks:

Task T1 evaluates binary quantifiers on data affected by prior probability shift (label shift), akin to Task T1A of LeQua 2022.

Task T2 assesses single-label multi-class quantifiers operating on data points belonging to one of $L > 2$ classes, with data affected by prior probability shift, similar to Task T1B of LeQua 2022.

Task T3 new to LeQua 2024, evaluates ordinal quantifiers handling a set of $L > 2$ ordered classes, also involving data affected by prior probability shifts.

Task T4 another new addition, evaluates binary quantifiers on data affected by covariate shifts.

Our contributions focus on Tasks T1 and T2. For the binary quantification task T1, we employ *Multiple Classifiers - Single Quantifier* (MC-SQ) [3], an ensemble method that achieved the top rank in MRAE. MC-SQ leverages multiple classifiers combined with a single quantifier. For the multi-class quantification task T2, we introduce *Single Classifier - Multiple Quantifiers* (SC-MQ), an ensemble approach that secured the third rank in MRAE. SC-MQ combines a single classifier with multiple quantifiers, including our proposed *Expectation-Maximization Quantifier with Initialization Adaptation* (EMQ-ini) method.

The performance of our ensemble methods can be attributed to the combination of multiple classifiers and quantifiers, leveraging the strengths of diverse models while mitigating individual weaknesses. These ensemble approaches enhance overall quantification accuracy. Furthermore, extensive hyperparameter tuning optimized the performance of each component within the ensembles, contributing to the top-ranking results.

This paper is organized as follows: Section 2 describes the classification and quantification methods employed in our ensemble approaches, MC-SQ and SC-MQ. Section 3 details the evaluation process and comprehensive hyperparameter tuning strategy. Finally, Section 4 concludes our work and presents directions for future research.

2 Methods

This section describes our proposed ensemble approaches, MC-SQ and SC-MQ, employed for the binary quantification task (T1) and multi-class quantification task (T2).

2.1 MC-SQ Approach for T1

For T1, we employ MC-SQ, an ensemble approach that previously achieved the top rank in T1B for the LeQua 2022 competition, based on our post-competition experiments [3]. Figure 1 illustrates the MC-SQ architecture, which consists of an ensemble of six pairs of classifiers and quantifiers. We introduce diversity by varying the base classifiers while keeping the base quantifier fixed. The chosen quantifier is *Distribution y-Similarity* (DyS) [4] as it has been recognized as one of the top-performing quantifiers for binary problems in the comparative study conducted by [5].

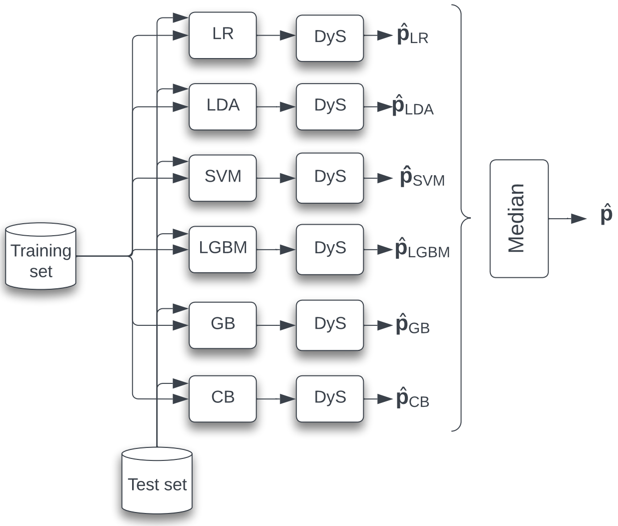
Training set
Test set
LR
LDA
SVM
LGBM
GB
CB
DyS
DyS
DyS
DyS
DyS
DyS
$\hat{\mathbf{p}}_{\mathrm{LR}}$
$\hat{\mathbf{p}}_{\mathrm{LDA}}$
$\hat{\mathbf{p}}_{\mathrm{SVM}}$
$\hat{\mathbf{p}}_{\mathrm{LGBM}}$
$\hat{\mathbf{p}}_{\mathrm{GB}}$
$\hat{\mathbf{p}}_{\mathrm{CB}}$
Median
$\hat{\mathbf{p}}$

Fig. 1. Schematic of the proposed MC-SQ ensemble approach.

Our approach encompasses the following classifiers: Logistic Regression (LR), Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Light Gradient Boosting Machines (LGBM), Gradient Boosting (GB), and CatBoost (CB). The rationale behind selecting these algorithms stems from their diverse learning paradigms and their frequent success in various Machine Learning applications. As initially proposed, MC-SQ [3] also incorporates Random Forest and Naive Bayes as base classifiers. However, an extensive evaluation of the competition dataset showed that pairing these classifiers with DyS exhibited higher quantification error rates on the validation data. Consequently, we removed them from our ensemble and introduced CatBoost to maintain diversity among the base classifiers while leveraging its strength in quantification tasks.

We conduct comprehensive hyperparameter tuning for each classifier and quantifier combination to optimise our ensemble approach. This process systematically explores the parameters space to identify the configurations that yielded the best quantification performance on the validation set. Section 3 provides further details on our methodology and experimental setup, including the comprehensive hyperparameter tuning process employed to optimize the performance of our ensemble approach.

2.2 SC-MQ Approach for T2

For the T2 task, we introduce SC-MQ, an ensemble approach that combines diverse quantification algorithms with a single base classifier. We evaluated various classifiers as potential base classifiers by measuring the quantification error over the validation set. Our analysis showed that logistic regression (LR), as the base classifier, coupled with the assessed quantifiers, yielded the lowest quantification error on the validation set. Consequently, we selected LR as the base classifier for our ensemble, aiming to optimize overall quantification performance. By fixing the *Single Classifier* as Logistic Regression, we aim to minimize the variability introduced by the classification component. This allows the quantification algorithms to be the primary source of diversity within the ensemble. Figure 2 shows the architecture of our proposed SC-MQ method.

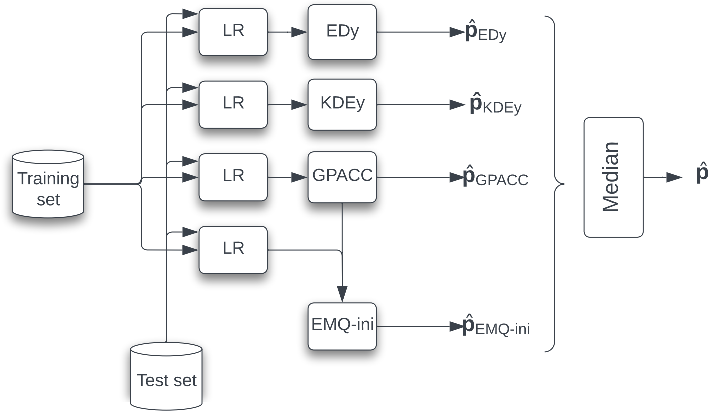
Training set
Test set
LR
LR
LR
LR
EDy
KDEy
GPACC
EMQ-ini
$\hat{\mathbf{p}}_{\text{EDy}}$
$\hat{\mathbf{p}}_{\text{KDEy}}$
$\hat{\mathbf{p}}_{\text{GPACC}}$
$\hat{\mathbf{p}}_{\text{EMQ-ini}}$
Median
$\hat{\mathbf{p}}$

Fig. 2. Schematic of the proposed SC-MQ ensemble approach.

We employ four quantification methods in our experiments: *Energy Distance* (EDy) [6], *Kernel Density Estimation* (KDEy) [7], *Generalized Probabilistic Adjusted Classify & Count* (GPACC) [8], and our newly proposed EMQ-ini. We select these methods based on our prior knowledge of quantification techniques and their ability to handle multi-class problems. Each algorithm represents a distinct approach to quantification. For KDEy, we specifically utilize the Maximum

Likelihood (ML) variation, as it showed the best performance for multi-class datasets among all KDEy variants in the original study [7].

Complementing these established quantifiers, we introduce EMQ-ini, a novel variation of the EMQ method [9] we are currently working on. EMQ-ini uses the predicted target priors from GPACC as the initial point of log-likelihood maximization. This competition provides an opportunity to test our idea and evaluate the performance of EMQ-ini on the quantification tasks in a real-world setting. We discuss the EMQ-ini method in detail in the next section.

2.3 EMQ-ini

EMQ-ini is a new quantification method designed to enhance the performance of traditional EM-based quantifiers. It leverages an initial estimate from GPACC to provide a more informed starting point for the EM process, improving convergence speed and accuracy, especially in scenarios where class distribution shifts are significant between the training and target domains.

Alg. 1 provides a detailed description of EMQ-ini. Suppose we have $\mathbf{y} = \{y_i\}_{i=1}^{L}$ as the set of labels, and $\mathbf{x}$ is the unlabelled test set with N instances sampled from the target domain. EMQ-ini takes three inputs: $\hat{p}_{ini}(\mathbf{y})$, which is an estimate of the target prevalence from GPACC, used as an initiation for EM; $p_t(\mathbf{y})$, as the class prevalence from a training set; and $\hat{p}_t(\mathbf{y}|\mathbf{x})$ is the estimate posterior class probabilities (*scores*) from a classifier trained on training set sampled from the source domain.

EMQ-ini updates the scores before applying the EM iterations. This update is based on the Bayes' rule, similar to the *E step* in EMQ. It begins by computing the prevalence ratio $\mathbf{r}$, which is the element-wise division of prevalence estimate $\hat{p}_{ini}(\mathbf{y})$ by the training prevalence $\hat{p}_t(\mathbf{y})$. This ratio is then used to compute the scores $\hat{p}_{ini}(\mathbf{y}|\mathbf{x})$.

The main iterative process of the EMQ-ini is similar to EMQ and consists of the following steps:

- **E-step**: In this step, the algorithm computes the updated posterior probabilities $\hat{p}^{(s)}(\mathbf{y}|\mathbf{x})$ for the current iteration s. This is done by adjusting the initial posterior probabilities $\hat{p}_{ini}(\mathbf{y}|\mathbf{x})$ based on the ratio of the current prevalence estimate $\hat{p}^{(s)}(\mathbf{y})$ to the initial prevalence estimate $\hat{p}_{ini}(\mathbf{y})$, using Bayes' rule.
- **M-step**: In this step, the algorithm updates the prevalence estimate $\hat{p}^{(s+1)}(\mathbf{y})$ for the next iteration $s+1$. This is done by taking the average of the updated posterior probabilities $\hat{p}^{(s)}(\mathbf{y}|\mathbf{x})$ across all instances in the dataset.

The E-step and M-step are iteratively performed until a stopping condition is met (e.g., convergence or maximum iterations reached).

After the iterative process, the final prevalence estimate $\hat{p}(\mathbf{y})$ is set to the last computed prevalence estimate $\hat{p}^{(s)}(\mathbf{y})$.

Algorithm 1: EM Quantifier with Initialization Adaptation.

```
Input: $\hat{p}_{ini}(\mathbf{y})$, $p_t(\mathbf{y})$, $\hat{p}_t(\mathbf{y}|\mathbf{x})$
Output: $\hat{p}(\mathbf{y})$
Prevalence ratio: $\mathbf{r} \leftarrow \frac{\hat{p}_{ini}(\mathbf{y})}{p_t(\mathbf{y})}$;
Updated posterior: $\hat{p}_{ini}(\mathbf{y}|\mathbf{x}) \leftarrow \left[\frac{r_i \cdot \hat{p}_t(y_i|\mathbf{x})}{\sum_{j=1}^{L} r_j \cdot \hat{p}_t(y_j|\mathbf{x})} \textbf{ for } i \leftarrow 1 \textbf{ to } L\right]$;
State: $s \leftarrow 0$;
Target prevalence estimate in state $s$: $\hat{p}^{(s)}(\mathbf{y}) \leftarrow \hat{p}_{ini}(\mathbf{y})$;
while stopping condition = false do
    E step: $\hat{p}^{(s)}(\mathbf{y}|\mathbf{x}) \leftarrow \left[\frac{\frac{\hat{p}^{(s)}(y_i)}{\hat{p}_{ini}(y_i)} \cdot \hat{p}_{ini}(y_i|\mathbf{x})}{\sum_{j=1}^{L} \frac{\hat{p}^{(s)}(y_j)}{\hat{p}_{ini}(y_j)} \cdot \hat{p}_{ini}(y_j|\mathbf{x})} \textbf{ for } i \leftarrow 1 \textbf{ to } L\right]$;
    M step: $\hat{p}^{(s+1)}(\mathbf{y}) \leftarrow \frac{1}{N} \sum_{x \in \mathbf{x}} \hat{p}^{(s)}(\mathbf{y}|x)$;
    $s \leftarrow s + 1$;
$\hat{p}(\mathbf{y}) \leftarrow \hat{p}^{(s)}(\mathbf{y})$;
return $\hat{p}(\mathbf{y})$;
```

3 Evaluation

The performance of the proposed methods is evaluated on both the validation and test sets for Tasks T1 and T2. Tables 1 and 3 present the results in terms of Mean Absolute Error (MAE) and Mean Relative Absolute Error (MRAE) for the validation and test sets, respectively.

For the binary quantification task T1, Table 1 compares the performance of the individual classifiers coupled with the DyS quantifier and the proposed MC-SQ ensemble method. Among the individual methods, LR-DyS and the MC-SQ ensemble achieved the best MAE of 0.0206 on both the validation and test sets. However, MC-SQ outperformed LR-DyS regarding MRAE, achieving the lowest scores of 0.0869 and 0.0981 on the validation and test sets, respectively.

The parameters for the classifiers and quantifiers used in the T1 task were selected using grid search and optimization to minimize the MRAE metric. This process was facilitated by the QuaPy library [10]. Table 2 presents the selected parameters obtained through this optimization procedure. Notable parameters include the regularization and gamma parameters for Logistic Regression and Support Vector Machines and the number of bins (nbins) used by the DyS quantifier, which varied across the different methods. Any other parameters or hyperparameters not explicitly mentioned in the parameters table are set to their respective default values as defined by the implemented methods.

For the multi-class quantification task T2, Table 3 compares the performance of the proposed SC-MQ ensemble with its individual components: EDy, EMQ-ini,

Table 1. Performance comparison of methods on validation and test sets for T1.

Method	Validation set		Test set	
	MAE	MRAE	MAE	MRAE
LR-DyS	**0.0206**	0.0910	**0.0206**	0.1024
LDA-DyS	0.0218	0.0987	0.0218	0.1121
SVM-DyS	0.0213	0.1023	0.0217	0.1026
LGBM-DyS	0.0253	0.1034	0.0255	0.1220
GB-DyS	0.0258	0.1052	0.0258	0.1176
CB-DyS	0.0233	0.0954	0.0233	0.1145
MC-SQ	**0.0206**	**0.0869**	**0.0206**	**0.0981**

Table 2. Classifier and Quantifier selected parameters for T1.

Method	Classifier	Quantifier
LR-DyS	C = 10, class-weight = balanced	nbins = 40
LDA-DyS	None	nbins = 30
SVM-DyS	C = 24.1967, gamma = 0.0114	nbins = 30
LGBM-DyS	None	nbins = 30
GB-DyS	None	nbins = 16
CB-DyS	Depth = 2, learning_rate = 0.1, l2_leaf_reg = 7, iterations = 900	nbins = 18

KDEy, and GPACC. The SC-MQ ensemble achieved the best performance, with MAE scores of 0.0129 and 0.0127 on the validation and test sets, respectively. It also obtained the lowest MRAE scores of 1.1160 and 1.0786 on the validation and test sets.

Table 3. Performance comparison of methods on validation and test sets for T2.

Method	Validation set		Test set	
	MAE	MRAE	MAE	MRAE
EDy	0.0137	1.3053	0.0135	1.2390
EMQ-ini	0.0139	1.1351	0.0137	1.1038
KDEy	0.0179	1.4542	0.0176	1.4355
GPACC	0.0155	1.2021	0.0155	1.1950
SC-MQ	**0.0129**	**1.1160**	**0.0127**	**1.0786**

The performance of SC-MQ can be attributed to the combination of a robust base classifier (Logistic Regression) with a diverse set of quantification algorithms, including the novel EMQ-ini method proposed in this work. Among the individual quantifiers, EMQ-ini was the best-performing single quantifier, outperforming other methods like EDy, KDEy, and GPACC. By leveraging the strengths of multiple quantifiers, with EMQ-ini being the strongest contributor while maintaining a consistent base classifier, the ensemble could effectively capture the diverse characteristics of the multi-class quantification problem.

Table 4 presents the selected parameters using the grid search and minimizing the MRAE for the classifier and quantifiers used in the T2 task. Notable parameters include the regularization parameter for Logistic Regression, the bandwidth parameter for KDEy, and the solver used by GPACC. Additionally, EMQ-ini utilized the exact training prevalences during the initialization step.

Table 4. Classifier and Quantifier selected parameters for T2.

Method	**Classifier**	**Quantifier**
EDy	C = 1, class-weight = balanced	None
KDEy	C = 100, class-weight = None	bandwidth = 0.14
GPACC	C = 0.1, class-weight = balanced	solver = minimize
EMQ-ini	C = 1, class-weight = None	exact_train_prev = True

4 Conclusion

The results demonstrate the effectiveness of the proposed ensemble approaches, MC-SQ and SC-MQ, in addressing the binary and multi-class quantification tasks, respectively. By combining diverse classifiers and quantifiers, these methods could leverage the strengths of individual components while mitigating their weaknesses, leading to improved quantification performance on both tasks.

For future work, we will investigate EMQ-ini in more depth, focusing on the effect of the initial point of maximum likelihood optimization on this method. Additionally, we will evaluate EMQ-ini on various datasets to ensure its generality and robustness across different scenarios.

References

1. A. Moreo and F. Sebastiani, "Tweet sentiment quantification: An experimental re-evaluation," *PLoS One*, vol. 17(9), 2022.
2. G. Forman, "Quantifying counts and costs via classification," *Data Min Knowl Discov*, vol. 17, no. 2, pp. 164–206, 2008.
3. Z. Donyavi, A. Serapio, and G. Batista, "Mc-sq: A highly accurate ensemble for multi-class quantification," in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 2023, pp. 622–630.
4. A. Maletzke, D. dos Reis, E. Cherman, and G. Batista, "Dys: a framework for mixture models in quantification," in *AAAI Conference*, vol. 33, no. 01, 2019, pp. 4552–4560.
5. T. Schumacher, M. Strohmaier, and F. Lemmerich, "A comparative evaluation of quantification methods," *arXiv preprint arXiv:2103.03223*, 2021.
6. J. J. del Coz, "Unioviedo (team2) at lequa 2022: Comparison of traditional quantifiers and a new method based on energy distance." in *CLEF (Working Notes)*, 2022, pp. 1869–1874.

7. A. Moreo, P. González, and J. J. del Coz, "Kernel density estimation for multiclass quantification," *arXiv preprint arXiv:2401.00490*, 2023.
8. A. Firat, "Unified framework for quantification," *arXiv preprint arXiv:1606.00868*, 2016.
9. M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure," *Neural computation*, vol. 14, no. 1, pp. 21–41, 2002.
10. A. Moreo, A. Esuli, and F. Sebastiani, "Quapy: a python-based framework for quantification," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4534–4543.