Lamarr at LeQua2024: Regularized Soft-Max Likelihood Maximization

Tobias Lotz and Mirko Bunse

Lamarr Institute for Machine Learning and Artificial Intelligence,
44227 Dortmund, Germany
{tobias.lotz,mirko.bunse}@cs.tu-dortmund.de

Abstract. As members of the Lamarr Institute, we participated in the open LeQua2024 competition. The goal in this competition was to predict the prevalences of classes in unlabeled sets of data, given a labeled training set. Our submission builds on the regularized maximization of a likelihood function with constraints that are implemented through a soft-max operator. Ultimately, this method ranked in the top three across all four disciplines of LeQua2024; most notably, we achieved the first place in discipline T4, a binary quantification task with covariate shift. In this paper, we detail our approach to the competition.

Keywords: Quantification · prior probability shift · covariate shift.

1 Introduction

LeQua2024[1] was a competition hosted for the evaluation of quantification methods. These methods estimate class prevalences in unlabeled sets of data, i.e., they estimate how often each class appears in each data set [4]. To learn the correspondence between class labels and feature vectors, a labeled training set is provided. Unlike classification, quantification is not concerned with predicting the label of each individual data item; what matters are aggregate predictions for sets of data items [5]. These predictions are complicated by shifts between the training distribution and the target distributions.

LeQua2024 consists of four disciplines that are separately evaluated. Each discipline represents a different quantification setting. Our team from the Lamarr Institute ranked in the top three across all disciplines.

T1 Binary quantification with prior probability shift (3rd place).
T2 Multi-class quantification with prior probability shift (2nd place).
T3 Ordinal quantification with prior probability shift (2nd place).
T4 Binary quantification with covariate shift (1st place).

In Sec. 2, we introduce the quantification method that we used across all disciplines. Sec. 3 details our optimization of the method's hyper-parameters. We conclude with Sec. 4.

[1] See https://lequa2024.github.io

2 Method

Let C be the number of classes and let $\mathcal{P} = \{\mathbf{p} \geq 0 : 1 = \sum_{i=1}^{C} \mathbf{p}_i\}$ be the set of valid class prevalence vectors. Making a prediction, such that $\hat{\mathbf{p}}_i$ is an estimate of $\mathbb{P}(Y = i)$ in set unlabeled data items, can be realized through the constrained minimization of a loss function $\ell : \mathcal{P} \to \mathbb{R}$,

$$\hat{\mathbf{p}} \;=\; \underset{\mathbf{p} \in \mathcal{P}}{\arg\min} \;\; \ell(\mathbf{p}) \tag{1}$$

where ℓ is typically defined in terms of an unlabeled data set $\mathcal{D} = \{x \in \mathcal{X}\}$.

Our approach to all disciplines of LeQua2024 evolves around three aspects: the choice of the loss function, the implementation of the $\mathbf{p} \in \mathcal{P}$ constraint, and regularization. Regularization is the only aspect through which we adapt our method to the particularities of each discipline.

Loss Function We employ the negative log-likelihood loss proposed by Alexandari et al. [1]. This choice is motivated by an exceptional performance recently reported for this loss function in combination with kernel density estimates [6] and by the winning performance of a closely related method in the first edition of LeQua in 2022 [7]. We opted against kernel densities because our initial experiments did not verify a performance improvement due to them.

Constraints We ensure that $\hat{\mathbf{p}} \in \mathcal{P}$ by optimizing over a latent vector $\mathbf{l} \in \mathbb{R}^{C-1}$ that we feed through a soft-max operator $\sigma : \mathbb{R}^{C-1} \to \mathcal{P}$ [2].

Regularization In T3, we promote solutions that we deem ordinally plausible [3]. In all other disciplines, we promote uniform solutions, assuming that the testing protocol of the competition exhibits a slight preference towards these outcomes. The impact of regularization is controlled through a hyper-parameter $\tau \geq 0$.

Combining all of the above aspects, our prediction $\hat{\mathbf{p}} = \sigma(\hat{\mathbf{l}}) \in \mathcal{P}$ is obtained after optimizing

$$\hat{\mathbf{l}} \;=\; \underset{\mathbf{l} \in \mathbb{R}^{C-1}}{\arg\min} \; -\sum_{i=1}^{C} \log \sum_{x \in \mathcal{D}} \frac{\hat{\mathbb{P}}(Y = i \mid X = x)}{\hat{\mathbb{P}}(Y = i)} \cdot [\sigma(\mathbf{l})]_i + r(\mathbf{l}) \tag{2}$$

$$\text{where} \quad [\sigma(\mathbf{l})]_i \;=\; \begin{cases} \frac{1}{1 + \sum_{j=1}^{C-1} \exp(\mathbf{l}_j)} & \text{if } i = 1 \\ \frac{\exp(\mathbf{l}_i)}{1 + \sum_{j=1}^{C-1} \exp(\mathbf{l}_j)} & \text{else} \end{cases}$$

$$\text{and} \quad r(\mathbf{l}) \;=\; \begin{cases} \tau \cdot \sum_{i=1}^{C-1} \Big([\sigma(\mathbf{l})]_i - [\sigma(\mathbf{l})]_{i+1} \Big)^2 & \text{for T1, T2, T4} \\ \tau \cdot \sum_{i=1}^{C-2} \Big([\sigma(\mathbf{l})]_i - 2[\sigma(\mathbf{l})]_{i+1} + [\sigma(\mathbf{l})]_{i+2} \Big)^2 & \text{for T3} \end{cases}$$

We estimate the posteriors $\hat{\mathbb{P}}(Y = i \mid X = x)$ through a multi layer perceptron (MLP) classifier, except for discipline T4, where a logistic regression outperformed the MLP on the validation set. The prior $\hat{\mathbb{P}}(Y = i)$ is estimated on the training set, which is also used to train the classifier within each discipline.

The numerical optimization of the loss function in Eq. 2 is realized through an unconstrained Newton conjugate gradient trust-region method [9], as it is implemented in the SciPy package [8]. The full implementation of our approach is publicly available on GitHub[2].

3 Hyper-Parameter Optimization

Each discipline of LeQua2024 provides, in addition to a labeled training set, a validation set for hyper-parameter optimization. This validation set consists of multiple sets $\mathcal{D}$ of data items, each sampled with a discipline-specific type of distribution shift. Performance evaluations on the validation data act as an estimate of the final performance on the test data; for the latter, the ground-truth remained hidden throughout the competition. Participants were able to choose those hyper-parameters that perform best during validation.

In order to optimize our hyper-parameter selection, we employed a coarse-to-fine grid-search adaptation strategy, starting from heuristically chosen starting grids. For numeric parameters, if the optimal value is located at the smallest or largest value in the grid, we shift it in such a way, that the currently optimal value now lies at the center. Otherwise, we decrease the difference of the candidate values, in order to allow finer updates to further improve the performance.

The final hyper-parameters that we selected for the different classifiers can be seen in tables 1 and 2. We note that regularization only had a minor impact on the validation performance of our method.

Table 1. Final hyper-parameters used for MLP classifiers.

task	hidden_layer_size	activation	alpha	learning_rate	solver	τ
T1	512	tanh	1e−1	1e−3	sgd	0
T2	512	tanh	1e−1	1e−5	adam	0
T3	320	tanh	1e−6	1e−3	adam	1e−3

Table 2. Final hyper-parameters used for logistic regressions.

task	C	class_weight	τ
T4	0.43571	None	1e−5

[2] See https://github.com/tobiaslotz/lequa2024

4 Conclusion

Our participation in LeQua2024 evolves around a maximum likelihood estimate with constraints that are implemented through a soft-max operator. We employed this estimate across all four disciplines of LeQua2024, with a discipline-specific regularization that only played a minor role on quantification performance. Our method achieved top-ranking results throughout the competition.

The reasons for this outcome—and their implications on future research—remain yet to be discussed. Two essential prerequisites for this discussion are *i)* specific information about the submissions of the other teams and *ii)* specific information about the pre-processing of LeQua's data. Lacking both prerequisites at the moment, we are looking forward to the conclusions that are to be drawn by LeQua's organizers.

References

1. Alexandari, A., Kundaje, A., Shrikumar, A.: Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In: Int. Conf. on Mach. Learn. pp. 222–232 (2020), http://proceedings.mlr.press/v119/alexandari20a.html
2. Bunse, M.: On multi-class extensions of adjusted classify and count. In: Int. Worksh. on Learn. to Quantify: Meth. and Appl. pp. 43–50 (2022), https://lq-2022.github.io/proceedings/CompleteVolume.pdf
3. Bunse, M., Moreo, A., Sebastiani, F., Senz, M.: Ordinal quantification through regularization. In: Europ. Conf. on Mach. Learn. and Knowl. Discov. in Databases. pp. 36–52. Springer (2022). https://doi.org/10.1007/978-3-031-26419-1_3
4. Esuli, A., Fabris, A., Moreo, A., Sebastiani, F.: Learning to Quantify. Inform. Retriev. Series, Springer (2023). https://doi.org/10.1007/978-3-031-20467-8
5. Forman, G.: Quantifying counts and costs via classification. Data Mining and Knowl. Discov. pp. 164–206 (2008). https://doi.org/10.1007/s10618-008-0097-y
6. Moreo, A., González, P., del Coz, J.J.: Kernel density estimation for multiclass quantification (2024), https://arxiv.org/abs/2401.00490
7. Senz, M., Bunse, M.: DortmundAI at LeQua 2022: Regularized SLD. In: Conf. and Labs of the Eval. Forum. pp. 1911–1915 (2022), http://ceur-ws.org/Vol-3180/paper-152.pdf
8. Virtanen, P., et al.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Meth. pp. 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
9. Wright, S.J., Nocedal, J.: Numerical optimization. Operations Res. and Financial Engin., Springer, 2nd edn. (2006). https://doi.org/10.1007/978-0-387-40065-5