KULeuven at LeQua 2022: Model Calibration in Quantification Learning

Teodora Popordanoska, Matthew B. Blaschko

Processing Speech and Images, Dept. of Electrical Engineering (ESAT), KU Leuven, Belgium

Abstract

In many decision-making applications it is of particular interest to obtain aggregated properties about the data sample. *Quantification learning* (or *prevalence estimation*) is the task of training predictors via supervised learning to estimate the class distribution of an unlabelled test set. In this paper, we describe our perspective on the "LeQua 2022: Learning to Quantify" challenge, and our entry in "the vector task - T1A," which achieved 1st place on the leaderboard in terms of relative absolute error, and 2nd place in terms of absolute error. We explored the relationship between probabilistic calibration and quantification, concluding that calibration can help with quantification, but this typically requires an i.i.d. assumption or known distribution shift.

Keywords

quantification learning, prevalence estimation, uncertainty calibration, bias

1. Introduction

We explore the task of class prevalence estimation from the perspective of uncertainty calibration. Calibrated uncertainty estimates of the predictions from probabalistic classifiers are important in many applications. Numerous techniques have been proposed to address the issue of miscalibration, and they generally fall under two categories. One category is **trainable calibration strategies**, which modify the training objective, usually by integrating a differentiable calibration measure in the form of a regularizer. Some examples include KDE-XE [1], MMCE [2] and Mix-n-Match [3]. A second category are **post-hoc calibration methods**, which rescale the predictions from the classifier after training. The most notable examples of this cateogry are Platt scaling [4] and its single-parameter version called temperature scaling [5].

A few works have reported the effects of calibration in quantification learning. On the one hand, calibration has been shown to be crucial for a well-known and highly competitive method for estimating class probabilities on unlabelled sets — namely, the Saerens-Latinne-Decaestecker (SLD) method [6]. Through large scale experimentation across multiple learners and varying amounts of distribution shift, the authors concluded that SLD is ineffective, and often detrimental, when the classifier has not been calibrated. On the other hand, calibration has been shown to deteriorate the results for other probabalistic quantifiers [7]. Both works used a simple post-hoc method to calibrate the classifier — namely, Platt scaling [4]. However,

https://codalab.lisn.upsaclay.fr/competitions/4134#results

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

teodora.popordanoska@kuleuven.be (T. Popordanoska); matthew.blaschko@esat.kuleuven.be (M. B. Blaschko)

CEUR Workshop Proceedings (CEUR-WS.org)

Karandikar et al. [8] demonstrate that under dataset shift, using calibration regularized training objectives result in better uncertainty estimates compared to post-hoc calibration methods. Since the main challenge in quantification stems from the fact that the data distribution changes between the training and testing phase, we investigate empirically whether KDE-XE, as a trainable calibration strategy, can provide further improvements on this task.

2. Methods

Machine learning is based on the notion of empirical risk minimization (ERM). This principle minimizes risk over a training set of data:

$$f^* := \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \tag{1}$$

for some loss function ℓ such as hinge loss or logistic loss. In the case of binary labels (i.e. $y \in \{0, 1\}$) and in the limit that $n \to \infty$ it is well known [9] that nearly all standard convex loss functions are equivalent and statistically consistent, i.e. f^* converges in probability to the Bayes optimal classifier that is the minimizer of the risk functional:

$$\mathcal{R}_P(f) := \int \ell(f(x), y) dP(x, y). \tag{2}$$

It is key to note that this implies the assumption that the data distribution P exists and that the samples are drawn i.i.d. from it. Although learning under data shift is also a well studied field, it is essentially unavoidable that a specific form of data shift must be assumed [10].

Our analysis of quantification learning is based on the analysis of the *bias* of a probabilistic predictor.

Definition 1 (Bias). *Let f be a function that predicts from an input space $\mathcal{X}$ the probability that the label $y \in \{0, 1\}$ is 1. The* bias *of f with respect to a data distribution P is:*

$$\text{Bias}_P(f) := \mathbb{E}_{(x,y)\sim P}[f(x) - y]. \tag{3}$$

It is straightforward to see that, for a given distribution P, we would like the magnitude of the bias to be minimized w.r.t. the empirical distribution of a data sample in order to ensure that $\sum_i f(x_i)$ is equal to the true number of positive elements within the sample.

It is in general difficult to ensure that the bias of a predictor is minimized, but there is a related notion of *calibration error* that can be used to assess if under a certain distribution P the confidence of $f(x) \in [0, 1]$ is equal to the probability that $y = 1$.

Definition 2 (L_p calibration error [11]). *The L_p calibration error of $f : \mathcal{X} \to [0, 1]$ is:*

$$\text{CE}_p(f) := \mathbb{E}\left[|\mathbb{E}[y|f(x)] - f(x)|^p\right]^{\frac{1}{p}} \tag{4}$$

It has recently been noted that calibration error can be used to control the bias of a predictor due to the following result:

Theorem 1 (Proposition 1 in [12]). *L_p calibration error provides an upper bound on the bias of a predictor of the form:*

$$\mathrm{CE}_1(f) \geq |\,\mathrm{Bias}(f)|. \tag{5}$$

We refer the reader to [12] for the proof.[1] Furthermore, as Figure 1 in [12] shows, in most cases the empirical values for $|\,\mathrm{Bias}\,|$ and CE_1 are equal. Thus, if we have a means to find a calibrated classifier, e.g. by trainable calibration methods [2, 1], we will simultaneously minimize the calibration error (with respect to the data distribution) and the bias of a predictor, thereby improving the accuracy of the quantification.

An additional result governing calibration error is that it is itself upper bounded by the MSE [13, 14] in the following sense:

$$\mathrm{MSE}(f) - \mathrm{CE}_2(f)^2 = \mathbb{E}[(1 - \mathbb{E}[y|f(x)])\mathbb{E}[y|f(x)]] \geq 0. \tag{6}$$

The inequality is due to the fact that the expectation is over a variance of Bernoulli random variables.

A simple consequence of this inequality is that minimizing MSE also minimizes calibration error, which in turn minimizes bias. It is therefore an empirical question whether simple maximization of accuracy is sufficient, or whether an additional calibration operation will yield the lowest possible bias. It is this question that we also sought to answer in our experiments.

A final important point is that all results involve an expectation with respect to a data distribution. Importantly, the competition performance metric appears to average over a number of distribution shifts, thereby making the evaluation more robust in a sense, but also limiting the ability to have the method specialize to a given distribution. This makes the appropriate notion of MSE, CE, and bias not entirely well defined, though we can of course use the training distribution as a proxy.

3. Experiments

3.1. Data

In task T1A, we tackle a binary quantification problem, where vector representations of the training, development, and test documents were provided. The training set consists of 5000 instances sampled from Amazon product reviews. The development and test set contain 1000 and 5000 samples, respectively, and each sample has 250 documents. The samples are generated with the so-called artificial prevalence protocol (APP), which manipulates the class distribution by sub-sampling, and creates a uniform distribution of class prevalence across the samples. More details about the experimental setting can be found in [15].

[1] We note that the proof in [12] is straightforwardly generalized to other L_p norms, for $p \geq 1$, by replacing the absolute value by $|\cdot|^p$, which is also convex and Jensen's inequality can therefore also be applied.

3.2. Metrics

The official measure for the challenge is relative absolute error (RAE), defined as:

$$RAE(p_\sigma, \hat{p}_\sigma) = \frac{1}{n} \sum_{y \in \mathcal{Y}} \frac{|\hat{p}_\sigma(y) - p_\sigma(y)|}{p_\sigma(y)} \tag{7}$$

where p_σ is the true distribution on sample σ and $\hat{p}_\sigma$ is the predicted distribution. The problem that occurs when at least one of the classes y is zero is resolved by additive smoothing.

3.3. Baselines

As part of the QuaPy library [7], several advanced quantification methods were provided as baselines, with Logistic Regression (LR) as the underlying classifier. We found the best performing quantification method on the development set, for several choices of a classifier, to be the expectation-maximization-based SLD method. Therefore, our analysis will focus on this method. Esuli et al. [6] reported a large improvement of the SLD method when used with calibrated classifiers and proposed calibration with Platt scaling (PS) [4]. We will denote this quantification method as **SLD_PS**.

Following the observation in [8] that uncertainty estimates of trainable calibration methods generalize better under dataset shift, we use KDE-XE [1], where a consistent and differentiable estimator of calibration error is used as a regularizer alongside cross-entropy (XE) loss to train a classifier. KDE refers to Kernel Density Estimation, which is used to estimate the conditional expectation $\mathbb{E}[y|f(x)]$ in Equation 4. Since SLD was used as a quantification method because of its superior performance, this baseline will be referred to as **SLD_KDE**.

3.4. Results

The empirical evaluation consists of two parts: in the first experiment (I) we consider a trainable calibration strategy and compare it with an already-known method, whereas the second experiment (II) aims at maximizing the accuracy of the underlying classifier.

In particular, in the first experiment we explore whether the trainable KDE-XE method has an advantage over the post-hoc Platt scaling (PS) strategy. As an underlying classifier, we consider a Multi Layer Perceptron (MLP) and a Logistic Regression (LR) model, implemented in PyTorch. From the results reported in the top four rows of Table 1, we notice that both for MLP and LR the PS calibration works better in combination with SLD. Therefore, for the next experiment and the final predictions we only consider PS calibration.

In the second experiment, we investigated several options for the classifier, such as LR, MLP, Random Forrest, Support Vector Machines (SVM), and classifiers based on gradient boosting (XGBoost and LightGBM). For each of the considered classifiers, we performed a randomized hyperparameter search to optimize for accuracy. Random search defines the search space as a bounded domain of hyperparameters and randomly samples points in that domain. The top-three best performing classifiers in terms of accuracy on a validation set are shown in the bottom three rows of Table 1. For the final predictions, we chose the classifier (SVM) and quantification method (SLD) that minimized the mean RAE across samples (MRAE) on the development set. More specifically, the optimal hyperparameters for the chosen SVM classifier were found to be an RBF kernel, with $C = 5.73437$ and $\gamma = 0.00146$, which were sampled from a loguniform distribution in the range $[1 \times 10^{-5}, 1 \times 10^{5}]$.

We note that in the first experiment, no hyperparameter optimization was performed for the respective classifiers, hence the difference in accuracy, for instance, between SLD_PS_LR baselines between the two experiments.

Table 1
MRAE and Accuracy for the baseline quantification methods and different choices of the underlying classifier. The accuracy is reported on a validation set. Best MRAE and Accuracy is marked in bold.

Experiment	Method	MRAE ↓	Accuracy ↑
I	SLD_PS_LR	0.133	0.861
	SLD_KDE_LR	0.272	0.861
	SLD_PS_MLP	0.279	0.850
	SLD_KDE_MLP	0.900	0.855
II	SLD_PS_SVM	**0.117**	**0.881**
	SLD_PS_MLP	0.137	0.870
	SLD_PS_LR	0.139	0.868

4. Discussion and Conclusions

The SLD quantification method, with an underlying Platt re-scaled SVM classifier yielded the lowest MRAE on the development set and was used to generate the predictions on the test set. We believe that the reasons why this combination worked well are: (i) SLD with calibrated classifiers has been show to be a very strong baseline, especially for binary settings [7, 6] (ii) the specific type of post-hoc calibration method used assumes that the calibration error can be corrected by applying a sigmoid function to the predictions, which has been empirically justified for SVMs with common kernel functions [4, Section 2.1] (iii) The SVM model achieved highest accuracy across different classifiers.

To conclude, we compared the performance of calibration regularized training and post-hoc Platt scaling in the task of quantification and found that KDE-XE does not perform very well under severe dataset shift. As a future work, it would be interesting to investigate further the role of calibration in prevalence estimation for defined classes of distribution shift.

Acknowledgments

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

References

[1] T. Popordanoska, R. Sayer, M. Blaschko, Calibration Regularized Training of Deep Neural Networks using Kernel Density Estimation, Openreview (2022).

[2] A. Kumar, S. Sarawagi, U. Jain, Trainable calibration measures for neural networks from kernel mean embeddings, in: ICML, 2018.

[3] J. Zhang, B. Kailkhura, T. Y.-J. Han, Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning, in: International Conference on Machine Learning, 2020.

[4] J. C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999, pp. 61–74.

[5] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, 2017. arXiv:1706.04599.

[6] A. Esuli, A. Molinari, F. Sebastiani, A critical reassessment of the saerens-latinne-decaestecker algorithm for posterior probability adjustment, ACM Transactions on Information Systems 39 (2020) 1–34. doi:10.1145/3433164.

[7] A. Moreo, A. Esuli, F. Sebastiani, QuaPy: A Python-Based Framework for Quantification, Association for Computing Machinery, New York, NY, USA, 2021, p. 4534–4543.

[8] A. Karandikar, N. Cain, D. Tran, B. Lakshminarayanan, J. Shlens, M. C. Mozer, B. Roelofs, Soft calibration objectives for neural networks, in: NeurIPS, 2021.

[9] P. L. Bartlett, M. I. Jordan, J. D. Mcauliffe, Convexity, classification, and risk bounds, Journal of the American Statistical Association 101 (2006) 138–156.

[10] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N. D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.

[11] A. Kumar, P. S. Liang, T. Ma, Verified uncertainty calibration, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019.

[12] T. Popordanoska, J. Bertels, D. Vandermeulen, F. Maes, M. B. Blaschko, On the relationship between calibrated predictors and unbiased volume estimation, in: Medical Image Computing and Computer-Assisted Intervention, 2021.

[13] A. Murphy, A new vector partition of the probability score, Journal of Applied Meteorology 12 (1973) 595–600.

[14] M. Degroot, S. Fienberg, The comparison and evaluation of forecasters., The Statistician 32 (1983) 12–22.

[15] A. Esuli, A. Moreo, F. Sebastiani, G. Sperduti, Overview of lequa 2022: Learning to quantify, in: Working Notes of the 2022 Conference and Labs of the Evaluation Forum (CLEF 2022), Bologna, IT, 2022.