DortmundAI at LeQua 2022: Regularized SLD

Martin Senz, Mirko Bunse

TU Dortmund University, Artificial Intelligence Group, D-44227 Dortmund, Germany

Abstract

The LeQua 2022 competition was conducted with the purpose of evaluating different quantification methods on text data. In the following, we present the solution of our team "DortmundAI", which ranked first in the multi-class quantification task T1B. This solution is based on a modification of the well-known Saerens-Latinne-Decaestecker (SLD) method. Here, the SLD method, which is based on expectation maximization, is extended by a regularization technique. Additional experiments with the test data, which we took out after the competition closed, reveal that our excellent ranking stems primarily from an extensive hyperparameter tuning of the classifier.

Keywords

Quantification, Supervised prevalence estimation

1. Introduction

Quantification is a supervised learning task which consists of training a predictor for class prevalences in a sample of unlabelled data items [1]. This task has received increased attention in recent years because, in many applications, the class distribution of a batch of data is relevant, rather than the prediction of individual instances of the data.

The LeQua 2022 competition [2] was initiated with the intention of evaluating the performance of methods that address quantification. Here, the focus is the quantification of text data, where the data consisted of collected customer reviews from Amazon. Two key learning tasks were formulated:

- (A) Binary quantification of reviews by positive (more than 3 stars) or negative rating (less than 3 stars)
- (B) Multiclass quantification of reviews according to 28 product categories

These tasks were further divided according to the data representation. Namely, the organizers provided vectorized data (1), as well as the raw text data (2), for a total of four tasks: T1A, T1B, T2A, and T2B. For more information about the competition and the evaluation protocol, see [2].

Our contributed solution focuses on the performance of the quantifier, rather than the representation of the data. Therefore, we relied on the vectorized representation and addressed only the tasks T1A and T1B. Our solution ranked first in T1B and fifth in T1A.

In Sec. 2, we describe the quantification method that we used. We complement our presentation in Sec. 3 with additional experiments that we conducted with the test set after the competition was closed.

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

martin.senz@tu-dortmund.de (M. Senz); mirko.bunse@cs.tu-dortmund.de (M. Bunse)

CEUR Workshop Proceedings (CEUR-WS.org)

2. Method

Our solution is based on the well-known quantification method SLD [3], which is extended by a regularization technique. Here, the idea of regularization is taken from another quantification method [4] that is domain-specific for experimental physics. Originally, our extension was proposed for ordinal quantification in particular [5].

We use the following notation. By $\mathbf{x} \in \sigma$ we denote a data item from an unlabelled data set $\sigma = \{\mathbf{x}_i \in \mathcal{X} : 1 \leq i \leq m\}$. By $y \in \mathcal{Y}$ we denote a class from a set of classes $\mathcal{Y} = \{y_1, ..., y_n\}$. Furthermore, $h : \mathcal{X} \to \mathcal{Y}$ represents a *soft* classifier that returns posterior probabilities $[h(\mathbf{x})]_i \equiv \mathbb{P}(y_i \mid \mathbf{x})$. By $\hat{p}_\sigma(y)$ we denote the prevalence of class y, as estimated by a quantification method that receives σ as an input. The goal of quantification is to return a $\hat{p}_\sigma(y)$ that is close to the true prevalence $\mathbb{P}(y \mid \sigma)$.

2.1. Saerens-Latinne-Decaestecker (SLD)

The SLD method [3] follows an expectation maximization approach, which (i) leverages Bayes' theorem in the E-step, and (ii) updates the prevalence estimates in the M-step. Both steps can be combined in a single update rule

$$\hat{p}_\sigma^{(k)}(y_i) = \frac{1}{|\sigma|} \sum_{\mathbf{x} \in \sigma} \frac{\frac{\hat{p}_\sigma^{(k-1)}(y_i)}{\hat{p}_\sigma^{(0)}(y_i)} \cdot [h(\mathbf{x})]_i}{\sum_{j=1}^{n} = \frac{\hat{p}_\sigma^{(k-1)}(y_i)}{\hat{p}_\sigma^{(0)}(y_i)} \cdot [h(\mathbf{x})]_j}. \qquad (1)$$

This update rule is applied until the estimates converge. The initial estimates $\hat{p}_\sigma^{(0)}(y_i)$ are given by the class prevalence values of the training set.

2.2. Regularization in SLD

We employ the regularization technique of the Iterative Bayesian Unfolding [4]. This physics-specific quantification method revolves around an expectation maximization with Bayes' theorem, and thus has a common foundation with SLD.

In IBU, each intermediate estimate $\hat{p}^{(k)}$ is regularized in the following way. First, a low-order polynomial is fitted to $\hat{p}^{(k)}$. Second, a linear interpolation between $\hat{p}^{(k)}$ and this polynomial is used as the prior of the next iteration. Due to the smoothness of low-order polynomials, this replacement of $\hat{p}^{(k)}$ reduces the differences between neighbouring prevalence estimates. Hence, the estimates are regularized towards smooth solutions. The interpolation factor between $\hat{p}^{(k)}$ and the polynomial and the order of the polynomial are hyperparameters of IBU through which the strength of the regularization is controlled.

The IBU regularization is particularly suitable for ordinal quantification tasks [5], where the classes follow a total order $y_i < y_{i+1}$. Without an order, the idea of "neighbouring classes" is not well-defined. However, we hypothesized that the IBU regularization might also benefit non-ordinal multi-class quantification tasks, like T1B in LeQua2022. This hypothesis is based on the idea that smoothing can suppress over- and under-estimations of class prevalences, even if the classes are not totally ordered. One of our motivations to participate in LeQua2022 was to test this hypothesis.

We call our quantification method o-SLD, as it was originally proposed for ordinal quantification [5]. Our method has two hyperparameters, the order of the polynomial and the interpolation factor. The pseudo code is displayed in Alg. 1.

Algorithm 1 o-SLD [5], our regularized version of SLD.

```
input: a soft multi-class classifier $h : \mathcal{X} \to \mathbb{R}^n$, the prevalences $\hat{p}_\sigma^{(0)}(y_i)$ of the training set, a data sample $\sigma = \{\mathbf{x}_i \in \mathcal{X} : 1 \leq i \leq m\}$, a polynomial order $o \in \mathbb{N}$, and an interpolation factor $0 \leq \lambda \leq 1$
output: a class prevalence estimate $\hat{p}_\sigma$
$k \leftarrow 0$
repeat
    $k \leftarrow k + 1$
    if $k > 1$ then
        fit a polynomial $f_o : \mathbb{R} \to \mathbb{R}$ to $\hat{p}_\sigma^{(k-1)}$
        $\hat{p}_\sigma^{(k-1)}(y_i) \leftarrow (1 - \lambda) \cdot \hat{p}_\sigma^{(k-1)}(y_i) + \lambda \cdot f_o(y_i)$ (regularization)
    end if
    update $\hat{p}_\sigma^{(k)}$, as according to Eq. 1 (standard SLD step)
until some distance between $\hat{p}_\sigma^{(k)}$ and $\hat{p}_\sigma^{(k-1)}$ is small
return $\hat{p}_\sigma^{(k)}$
```

3. Evaluation

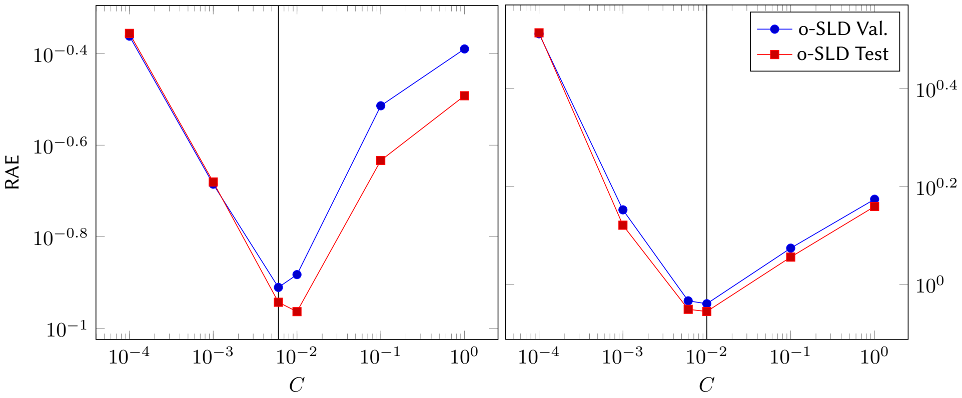
The primary objective of this evaluation is to measure the performance of the o-SLD method. For this purpose, a detailed process of model selection based on the relevant hyperparameters was performed. This process involved the optimization of a variety of model configurations on the training data and the estimation of their performance from the given validation data. To this end, we ran an exhaustive full grid search, where the respective hyperparameter search spaces were iteratively adjusted.

Overall, the following hyperparameters were identified as being relevant:

- Degree $o \in \mathbb{N}$ of the polynomial which replaces $\hat{p}^{(k)}$
- Impact $\lambda \in [0, 1]$ of the linear interpolation between the polynomial and $\hat{p}^{(k)}$
- Inverse of regularization strength C (Logistic Regression)

Initially, Logistic Regression and Support Vector Machines (SVM) were found to be suitable classifier candidates. Since no improvement of the results was observable with SVM, the focus was then put on Logistic Regression classifiers.

Based on the validation results, it became obvious that the choice of C has a high impact on the obtained results. Accordingly, careful tuning of C was essential to get satisfactory results. Considering the hyperparameters o and λ, a configuration with $C = 0.006$ was found for subtask T1A, as well as $C = 0.01$ for task T1B. As can be seen from Fig. 1, there is yet a different $C = 0.01$ for task T1A, which has a smaller test error than the one selected by the validation data. The C parameter has a major impact on the performance of o-SLD.



(a) Task T1A

(b) Task T1B

Figure 1: Influence of the hyperparameter C on the results by o-SLD on the validation and test data. The vertical line symbolizes the minimum validation error found.

Table 1
Impact of the o-SLD regularization hyperparameters, in terms of RAE, for Task T1B and $C = 0.01$.

	$o = 0$	$o = 1$	$o = 2$	$o = 3$	$o = 4$	$o = 5$	$o = 6$
$\lambda = 0.1$	1.27155	1.25751	1.23959	1.22277	1.21605	1.21044	1.19767
$\lambda = 0.01$	0.94607	0.944458	0.944458	0.94142	0.938579	0.938455	0.937126
$\lambda = 0.001$	0.913082	0.913042	0.912714	0.912534	0.912491	0.912623	**0.912485**

Table 2
The final configurations and results of o-SLD and SLD, based on the model selection performed. The testing scores and the SLD results were generated after the completion of the challenge.

Task	Model	Validation RAE	Test RAE	Test AE
T1A	o-SLD ($C = 0.006$, $o = 1$, $\lambda = 0.1$)	0.122869	0.1140	0.0271
	SLD($C = 0.006$)	0.122869	0.1140	0.0271
T1B	o-SLD ($C = 0.01$, $o = 6$, $\lambda = 0.001$)	0.912485	0.8799	**0.0117**
	SLD ($C = 0.01$)	**0.910511**	**0.8780**	0.0118

In the progress of the model selection for task T1B, it also appeared that model configurations with a small influence value λ are preferred, see Tab. 1 for example. Since o-SLD approaches the standard SLD method when λ decreases, this indicates that the additional smooth regularization is not an improvement in T1B.

This indication also shows when comparing the final o-SLD results with the standard SLD in Tab. 2. During the competition, we omitted an evaluation of the standard SLD because the hyperparameter grid of o-SLD included also small regularization impacts, with which the two methods are nearly equivalent.

Synopsis In the LeQua 2022 competition, the presented o-SLD achieved the first place for task T1B. As our experiments from Tab. 2 show, this excellent ranking was not achieved due to the regularization provided by o-SLD, but due to an extensive model selection that focused on optimizing the regularization parameter C. Although o-SLD could not achieve a lower error in this specific competition, the method has the capability to be useful in other quantification tasks, like ordinal quantification.

References

[1] G. Forman, Counting positives accurately despite inaccurate classification, in: European Conference on Machine Learning, 2005, pp. 564–575.

[2] A. Esuli, A. Moreo, F. Sebastiani, G. Sperduti, A detailed overview of LeQua 2022: Learning to quantify, in: Working Notes of the Conference and Labs of the Evaluation Forum, 2022. To appear.

[3] M. Saerens, P. Latinne, C. Decaestecker, Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure, Neural computation 14 (2002) 21–41.

[4] G. D'Agostini, A multidimensional unfolding method based on Bayes' theorem, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 362 (1995) 487–498.

[5] M. Bunse, A. Moreo, F. Sebastiani, M. Senz, Ordinal quantification through regularization, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2022. To appear.