Job Title Classification Strategies for the German Labor Market

Masterthesis

submitted by

Rahkakavee Baskaran

at the

Universität Konstanz

Department of Politics and Public Administration

Center for Data and Methods

1.Gutachter: Prof. Dr. Susumu Shikano

2.Gutachter: JunProf Juhi Kulshresthra

Konstanz, November 29, 2021

Contents

Abbreviations

SVM Support Vector Machine 5

NB NB 5
(Naive Bayes)

LR LR 5
Logistic Regression

OA overall accuracy 3

ROC receiver operating characteristics 5

TP True positives 3

TN True negatives 4

FN False negatives 3

FP False positives 4

1 Introduction

2 Related work

2.1 Short text classifcation

3 Data and taxonomy

4 Pipeline

5 Preprocessing

6 Evaluation metrics

There exists several metrics for the evaluation of classification approaches in the literature (Fatourechi et al., 2008). The choice of appropriate measurements is a crucial step for obtaining a qualitative comparison in the performance between the baseline algorithms and the new approaches. Often researchers rely on popular metrics like overall accuracy (OA). However, especially for multiclass and inbalanced dataset tasks it is difficult to rely only on one measure like OA In order to select appropriate metrics for comparison in the following the most important metrics will be discussed focussing on multiclass classification and imbalanced data sets.

Most metrics rely on a confusion matrix. For the multiclass case this confusion matrix is defined as follows (Kautz et al., 2017):

	positive examples			
positive prediction	$c_{1,1}$	$c_{1,2}$	$\dots$	$c_{1,n}$
	$c_{2,1}$	$c_{i,j}$		
	$\vdots$		$\ddots$	$\vdots$
	$c_{n,1}$		$\dots$	$c_{n,n}$

Table 1: Confusion Matrix (edited after (Kautz et al., 2017, 113)

From the confusion matrix follows that $c_{i,j}$ defines examples which belong to class j and are predicted as class i. Given that k is the current class, True positives (TP) is defined as $tp_k = c_{k,k}$, thus examples which are correctly predicted as the current class k. False negatives (FN) are defined as those examples which not belonging to the current

class k, but are predicted as k. Formally $fn_k = \sum_{i=1,i\neq k}^{n} c_{i,k}$. Next, True negatives (TN), are examples belonging to the current class m, but are not predicted as m. Formally $tn_k = \sum_{i=1,i\neq k}^{n} \sum_{j=1,j\neq k}^{n} c_{i,j}$. Last, False positives (FP) are defined as examples not belonging to class k, but are predicted as such. Formally this can be expressed as: $fp_k = \sum_{i=1,i\neq k}^{n} c_{k,i}$ (Kautz et al., 2017)

As mentioned the OA is one of most common metric for performance evaluation. It represents how well the classifier classifies across all classes correctly. Formally, given that N is number of examples and K the number of all classes, this can be expressed as (Branco et al., 2017):

$$OA = \frac{1}{K} \sum_{i=1}^{K} \frac{tp_k + tn_k}{N}$$

Following the formula an accuracy of 1 means that all examples are correctly classified, while a 0 mean that each example is classified with the wrong class. (Berthold et al., 2020) Although OA is a widley used metric it is critized for favouring the majority classes, thus not reflecting minority classes appropriatly in unbalanced datasets (Berthold et al., 2020; Fatourechi et al., 2008)

Two more popular metrics are precision and recall. Precision respresents how well the classifier detects actual positive examples among the positive predicted examples. Recall, also called sensitivity, in contrast, represents how many examples are labelled as positive among the actual positive examples (Berthold et al., 2020). For the multiclass scenario, two different calculation approaches for each of the metrics are proposed: micro and macro average (Branco et al., 2017). In the macro approach first the metric is calculated for each class k against all other classes. The average of all of them is built. Formally:

$$precision_{macro} = \frac{1}{K} \sum_{i=1}^{k} \frac{tp_i}{tp_i + fp_i}$$

$$recall_{macro} = \frac{1}{K} \sum_{i=1}^{k} \frac{tp_i}{tp_i + fn_i}$$

In contrast the micro approach aggregates the values, which can be formally expressed as follows:

$$precision_{micro} = \frac{\sum_{i=1}^{K} tp_i}{\sum_{i=1}^{K} tp_i + fp_i}$$

$$recall_{micro} = \frac{\sum_{i=1}^{K} tp_i}{\sum_{i=1}^{K} tp_i + fn_i}$$

There is a trade-off between precision and recall (Buckland and Gey, 1994). The F-measure capture both precision and recall by taking the harmonic mean between both. It is calculated as follows (Branco et al., 2017; Pan et al., 2016):

$$F_{micro} = 2 \cdot \frac{precision_{micro} \cdot recall_{micro}}{precision_{micro} + recall_{micro}}$$

$$F_{macro} = 2 \cdot \frac{precision_{macro} \cdot recall_{macro}}{precision_{macro} + recall_{macro}}$$

Apart from the trade-off between recall and precision, there is also a tradeoff between sensitivity and specificty (1- sensitivity). Using a receiver operating characteristics (ROC), which plots the specifity against the sensitivity the trade-off can be visualized for different thresholds. The area under the curve then can be used to obtain the performance of the classifier. A large area indicates a better classifier (Berthold et al., 2020; Espíndola and Ebecken, 2005).

As shown above, there are several metrics for evaluating the performance of a classifier, with the metrics having different focuses. Since the job title classification involves multiclass classification and the descriptive analysis show that the data is clearly unbalanced, at least for some classes in level 5, it is not reasonable to base the evaluation solely on the OA. Taking precision, recall and the harmonic mean into account would capture the performance of the minority classes as well. The ROC curve does gives, due to its visualization a good impression for the performance, but it is not feasible for high number of classes. Following this argumentation the performance of the classifiers will be evaluated with accuracy, precision, recall, F-measure and Cohen's Kappa.

7 Baseline Algorithms

In order to compare different feature selection methods solid baselines are necessary. As pointed out in the literature review NB (NB), LR (LR) and Support Vector Machine (SVM) have several advantages for text classifcation tasks. In the following based on a theoretical discussion of each classifier, the exact modeling of the baseline classifiers is justified.

7.1 Naive Bayes Classifier

The NB, a family of probabilistic classifiers, uses Bayes' rule in order to determine the most likely class for each document (Schneider, 2005). All NB classifiers rely on the conditional independence assumptions which means, that "features are independent of each other, given the category variable" (Xu, 2018, 48). Depending on whether the features are discrete or continous, different distributions, so-called event models are proposed. While for continous features Gaussian distribution is well-suited, for categorical features usually Bernoulli or multinomial distibutions are applied (Xu, 2018).

7.2 Support Vector Machines

However, SVM also performed well for text classification. Especially for multiclass tasks, as mentioned in the literature review, often different versions of the algorithm are used and showed good performance (Aiolli and Sperduti, 2005; Angulo et al., 2003; Benabdeslem and Bennani, 2006; Guo and Wang, 2015; Mayoraz and Alpaydm, 1999; Tang et al., 2019; Tomar and Agarwal, 2015). In general SVM has several advantages for text classifcation. First, text classifcation usually has a high dimensional input space. SVM can handle these large features since they are able to learn independently of the dimensionality of the feature space. In addition SVMs are known to perform well for dense and sparse vectors, which is usually the case for text classification (Joachims, 1998). Empirical results, for example Joachims (1998) or Liu et al. (2010) confirm the theoretical expectations. It is, therefore, a reasonable option to use a basic version of the SVM algorithm as a baseline.

The general idea of a SVM is to map "the input vectors x into a high-dimensional feature space Z through some nonlinear mapping chosen a priori [...], where an optimal separating hyperplane is constructed" (Vapnik, 2000, 138). In SVM this optimal hyperplane maximizes the margin, which is simply put the distance from the hyperplane to the closest points, so called Support Vectors, across both classes (Han et al., 2012). Formally, given a training data set with n training vectors $x_i \in R^n, i = 1, ..., n$ and the target classes $y_1, ...y_i$ with $y_i \in \{-1, 1\}$, the following quadratic programming problem (primal) has to be solved in order to find the optimal hyperplane:

$$\min_{w,b} \frac{1}{2} w^T w$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1$$

where $\phi(x_i)$ transforms x_i into a higher dimensional space, w corresponds to the weight and b is the bias (Chang and Lin, 2001; Jordan et al., 2006) The given optimzation function assumes that the data can be separated without errors. This is not always possible, which is why Cortes et al. (1995) introduce a soft margin SVM, which allows for missclassfication (Vapnik, 2000). By adding a regularization parameter C with $C > 0$ and the corresponding slack-variable ξ the optimization problem changes to (Chang and Lin, 2001; Han et al., 2012):

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq, i = 1, ..., n$$

Introducing Lagrange multipliers α_i and converting the above optimization problem into a dual problem the optimal w meets (Chang and Lin, 2001; Jordan et al., 2006):

$$w = \sum_{I=1}^{n} y_i \alpha_i \phi(x_i)$$

with the decision function (Chang and Lin, 2001):

$$\text{sgn } (w^T \phi(x) + b) = sgn(\sum_{i=1}^{n} y_i \alpha K(x_i, x) + b)$$

$K(x_i, x)$ corresponds to a Kernel function, which allows to calculate the dot product in the original input space without knowing the exact mapping into the higher space (Han et al., 2012; Jordan et al., 2006).

In order to apply SVM to multiclass problems several approaches have been proposed. One stratetgy is to divide the multi-classifcation problem into several binary problems. A common approach here is the one-against-all method. In this method as many SVM classifiers are constructed as there are classes k. The k-th classifier assumes that the examples with the k label are positive labels, while all the other examples treated as negative. Another popular approach is the one-against-one method. In this approach $k(k-1)/2$ classifiers are constructed allowing to train in each classifier the data of two classes (Hsu and Lin, 2002). Besides dividing the multiclass problem into several binary problems, some researches propose approaches to solve the task in one single optimization

problem, like Crammer and Singer (2001). [1].

In order to find a strong baseline I checked SVM's with different parameters for the SVM, as well as different multiclass approaches. It appears that a SVM using a soft margin with a $C = 1$ and a one-vs-rest approach has the best results. I also test different kernels, like RBF Kernel or linear kernel. The linear kernel, formally $k(x, x') = x^T x'$, achieved the best results, which is why I choose it for the baseline.

8 Implementation of ...

9 Experimental results

10 Discussion and Limitations

[1]For a detailed overview of all different methods and the method of Crammer and Singer (2001) see Hsu and Lin (2002); Crammer and Singer (2001)

References

Aiolli, F. and Sperduti, A. (2005). Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, 6:817–850.

Angulo, C., Parra, X., and Català, A. (2003). K-svcr. a support vector machine for multi-class classification. *Neurocomputing*, 55:57–77.

Benabdeslem, K. and Bennani, Y. (2006). Dendogram based svm for multi-class classification. *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pages 173–178.

Berthold, M. R., Borgelt, C., Höppner, F., Klawonn, F., and Silipo, R. (2020). *Guide to Intelligent Data Science*. Springer, 2 edition.

Branco, P., Torgo, L., and Ribeiro, R. P. (2017). Relevance-based evaluation metrics for multi-class imbalanced domains. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10234:698–710.

Buckland, M. and Gey, F. (1994). The relationship between recall and precision. *Journal of the American society for information science*, 45.

Chang, C.-C. and Lin, C.-J. (2001). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2.3:1–27.

Cortes, C., Vapnik, V., and Saitta, L. (1995). Support-vector networks editor. *Machine Leaming*, 20:273–297.

Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

Espíndola, R. P. and Ebecken, N. F. F. (2005). On extending f-measure and g-mean metrics to multi-class problems. *WIT Transactions on Information and Communication Technologies*, 35:25–34.

Fatourechi, M., Ward, R. K., Mason, S. G., Huggins, J., Schlögl, A., and Birch, G. E. (2008). Comparison of evaluation metrics in classification applications with imbalanced datasets. *Proceedings - 7th International Conference on Machine Learning and Applications, ICMLA 2008*, pages 777–782.

Guo, H. and Wang, W. (2015). An active learning-based svm multi-class classification model. *Pattern Recognition*, 48:1577–1597.

Han, J., Kamber, M., and Pei, J. (2012). *Data Mining: Concepts and Techniques*. Elsevier Inc., 3 edition.

Hsu, C. W. and Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *European Conference on Machine Learning*, pages 137–142.

Jordan, M., Kleinberg, J., and Schölkopf, B. (2006). *Pattern Recognition and Machine Learning*. Springer.

Kautz, T., Eskofier, B. M., and Pasluosta, C. F. (2017). Generic performance measure for multiclass-classifiers. *Pattern Recognition*, 68:111–125.

Liu, Z., Lv, X., Liu, K., and Shi, S. (2010). Study on svm compared with the other text classification methods. *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, 1:219–222.

Mayoraz, E. and Alpaydm, E. (1999). Support vector machines for multi-class classification. *Lecture Notes in Computer Science*, 1607:833–842.

Pan, W., Narasimhan, H., Protopapas, P., Kar, P., and Ramaswamy, H. G. (2016). Optimizing the multiclass f-measure via biconcave programming. *IEEE 16th International Conference on Data Mining (ICDM)*, pages 1101–1106.

Schneider, K. (2005). Techniques for improving the performance of naive bayes for text classification.

Tang, L., Tian, Y., and Pardalos, P. M. (2019). A novel perspective on multiclass classification: Regular simplex support vector machine. *Information Sciences*, 480:324–338.

Tomar, D. and Agarwal, S. (2015). A comparison on multi-class classification methods based on least squares twin support vector machine. *Knowledge-Based Systems*, 81:131–147.

Vapnik, V. N. (2000). *The nature of statistical learning theory*. Springer.

Xu, S. (2018). Bayesian naïve bayes classifiers to text classification:. *Journal of Information Science*, 44:48–59.