Empirical evaluation of the BCOC method on multi-domain sentiment analysis data sets

Brian Keith Norambuena[1] and Claudio Meneses Villegas[1]

Departamento de Ingeniería de Sistemas y Computación, Universidad Católica del Norte
Av. Angamos 0610, Antofagasta, Chile.
{brian.keith@ucn.cl,cmeneses@ucn.cl}

Abstract. Sentiment analysis and opinion mining is an area that has experienced considerable growth over the last decade. This area of research attempts to determine the feelings, opinions, emotions, among other things, of people on something or someone. In particular, this article discusses the task of determining the polarity of reviews using an ordinal classification technique called Barycentric Coordinates for Ordinal Classification (BCOC). The aim of this analysis is to explore the viability of the application of BCOC on the field of sentiment analysis. This method is based on the hypothesis that the ordinal classes can be represented geometrically inside a convex polygon on the real plane by using barycentric coordinates. A set of experiments were conducted to evaluate the capability and performance of the proposed approach relative to a baseline, using accuracy as the general measure of performance. In general, the method is competitive with the baseline. For sentiment analysis with four classes, the results show improvements in the overall accuracy.

Keywords: Opinion Mining, Ordinal Classification, Sentiment Analysis, Barycentric Coordinates

1 Introduction

Opinions are central to almost all human activities due to their crucial influence on people's behavior. Every time that there is a need to make a decision, human beings seek to know other people's opinions. In the real world, companies and organizations want to know the public's opinion about their products and services. Moreover, shoppers want to know what other customers think about a product before purchasing it. In the past, people turned to their friends and family for opinions, whereas companies relied on surveys or focus groups. However, nowadays the explosive growth of social media and the increase in the available sources of data has made individuals and organizations use the information provided by these to support their decision-making process. The field of sentiment analysis, also known as opinion mining [6] has been developing in this context.

One of the main tasks in sentiment analysis is determining the polarity, though it should be noted that sentiment analysis encompasses several other tasks apart from polarity determination. This can be seen as a classification problem in general, as most of the literature does. The most common approaches to deal with this task are support vector machines and the naive Bayes classifier [8]. A literature review shows that most works focus on binary classification; the multi-class case has not been exhaustively studied [8].

Most works on determining polarity use a binary classification approach [8]. Other studies that use multi-class classification with five levels apply regression methods and then a transformation into the corresponding class [6]. However, recent works such as [10] illustrate that multi-class classification of polarity at the full document level remains elusive, even when using the deep learning approach proposed in that article.

This work seeks to extend the evaluation of the BCOC method [5], since it this method showed promising results in its first evaluation. However, more extensive experimentation is required in order to assess the value of this proposal. Thus, this paper seeks to fill this gap by providing an evaluation of the BCOC method on several sentiment analysis data sets.

In most cases, determining the polarity can be seen as a task of ordinal classification (in the multi-class case, because in the binary case order has no relevance), where the ordering of the different classes corresponds to the natural order provided by their different labels (i.e., the order of the classes is: very negative, negative, neutral, positive and very positive). In spite of the ordinal nature of the problem and that ordinal classification methods have been widely studied, upon reviewing the literature it can be noted that these characteristics of the data have not yet been exploited exhaustively to obtain better classifiers.

The term ordinal classification makes reference to the supervised learning problem of classification where classes have a natural order imposed on them due to the characteristics of the concept studied. When the problem has, in fact, an ordinal nature, it would be expected that this order would also be present in the input space [10]. In contrast to nominal classification, there is an ordinal relationship among the categories and it is different from regression in that the possible number of ranks is finite and the exact differences between each rank are not defined. In this way, ordinal classification lies somewhere between nominal classification and regression [9]. Determining if the problem has an ordinal nature requires knowledge on the domain and problem themselves.

Most of the methods in the field of sentiment analysis correspond to techniques for nominal data, namely, data in which the class labels belong to a set with no natural order. In contrast to this approach, the problem can be tackled with ordinal classification methods (sometimes called ordinal regression), which lies in a middle ground between classic classification and regression [11].

In a problem of authentic ordinal nature, this order is also expected to be present in the input space [4]. Defining a space where the ordinal nature of the data is evident could prove useful for the functioning of a classifier. The hypothesis behind the proposed method is based on the intuition that exploiting the ordinal nature of the classes should bring about a positive effect in a classifier's performance, in particular in problems of multi-class classification.

2 Description of the BCOC method

In this section the BCOC method is described, taking as reference the original proposal [5]. The central idea of this method arises from the triangular representation used by SentiWordNet 3.0 to model the different terms [1] and could be seen as one possible generalization of it.

The BCOC method uses a barycentric coordinates system [3] as its basis to represent the input and to carry out the classification, this representation gives the method its name. It should be noted that the coordinate transformation is a simple mathematical function. Intuitively, this barycentric coordinates representation is much closer to the structure of opinions. In a certain way, this is implicit in the representation used by SentiWordNet 3.0.

While the triangle could be considered for a problem of 3 classes, the method generalizes to a problem with n classes, taking as a base the case of $n = 3$. While the idea originates from the representation of terms in a sentiment analysis lexicon, the notion is general enough to be used in other classification contexts, given that the labels have an ordinal structure.

The BCOC method is based on the use of the vertices of a convex polygon inscribed in a semi-circumference to represent the different classes. An example can be seen in Figure 1. Note that the order of the classes is preserved by the relation of distance between the points that represent them. It is clear that the class "--" is more distant from "++" than from "-" or "+". This is easily confirmed by the geometric intuition of the representation and it is easy to prove by applying the triangle inequality from linear algebra and elementary geometry.

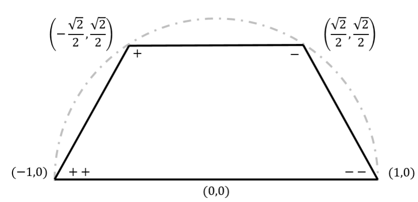

Fig. 1. Representation of four ordinal classes.

The method is based on the construction of n classifiers using a one-vs-all approach. Using these classifiers, an estimate of the probability of belonging to each class must be obtained. The results of these classifiers are transformed into a geometric representation and are then classified using a new classifier that is trained in function of the new representation.

Each class $C_i (i = 1, ..., n)$ is associated with a point x_i which corresponds to a vertex in a convex polygon of n sides (as seen in the example of Figure 1), then the system of barycentric coordinates is utilized to obtain the position of the example (taking into account the result of all classifiers) within the polygon. Each point x_i can be determined using the formula in Equation 1. Which generates the vertices of a n-sided polygon inscribed in a semi-circumference. The point $(1,0)$ is associated with the lowest class and the point $(-1,0)$ with the highest class with respect to the order of the labels.

$$x_i = \left(\cos\left[\frac{\pi(i-1)}{n-1}\right], \sin\left[\frac{\pi(i-1)}{n-1}\right]\right), \quad i = 1, ..., n \tag{1}$$

Given an example from the data set, a probability p_i is generated for each class i from the lower level classifiers. This probability represents the chance that the current instance belongs to class i (obtained in a one-vs-all fashion). Due to the fact that each probability is obtained using independent classifiers, the sum of the obtained probabilities will not necessarily add up to 1. However, the definition of barycentric coordinates requires that the sum of the weights of each point add up to 1 [3]. Therefore, it is necessary to normalize the resulting probabilities using Equation 2.

$$\lambda_i = \frac{p_i}{\sum_{j=1}^{n} p_j}, \quad i = 1, ..., n \tag{2}$$

This normalized coefficient is denoted by λ_i and represents the weight of the class i for the current instance. These coefficients in turn correspond with the barycentric coordinates themselves and can be turned directly into a point in the plane inside the convex polygon. According to the generalized version of barycentric coordinates, the formula to obtain the final representation of each point inside a convex polygon is shown in Equation 3.

$$x = \sum_{i=1}^{n} \lambda_i \cdot x_i = \frac{\sum_{i=1}^{n} p_i \cdot x_i}{\sum_{i=1}^{n} p_i} \tag{3}$$

The point x is finally fed into an additional classifier that is in charge of determining the class. In the exceptional case when $\sum_{i=0}^{n} p_i = 0$, in order to avoid division by zero, x is assigned to the centroid of the polygon. It should be noted that this is equivalent to the case where all $p_i = 1$, but with the assigned classes reversed, in both cases, the result will be assigned to the centroid.

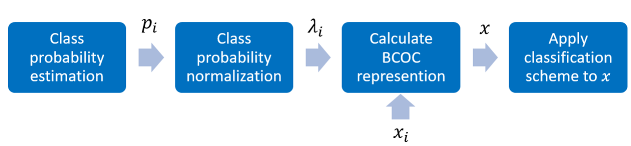

Fig. 2. Process flow of the BCOC method.

The general process flow of the BCOC method is represented in Figure 2. The training required before applying BCOC can be summarized as follows:

1. For each class train a one-vs-all classifier that can provide probability estimates.
2. For each instance in the data set estimate the class probabilities using the previous classifiers.
3. For each instance in the data set normalize the class probabilities according to Equation 2.
4. For each instance obtain the new representation in the BCOC two-dimensional space using Equation 3.
5. Train a multi-class classifier using the new feature space as input.

The application of the BCOC method on a test data set is similar:

1. For each instance in the data set estimate the class probabilities using the previously trained one-vs-all classifiers.
2. For each instance in the data set normalize the class probabilities according to Equation 2.
3. For each instance obtain the new representation in the BCOC two-dimensional space using Equation 3.
4. Apply the final multi-class classifier on the new feature space.

3 Materials and Methods

The present section describes the data sets used, the evaluation metrics utilized and the preprocessing of the data sets. The method is evaluated in the task of determining sentiment polarity on various domains, using the Multi-Domain Sentiment Data Set (version 2.0) from the works of Blitzer et al. [2]. This multi-domain data set contains different kinds of reviews. All the data sets have four classes (very negative, negative, positive and very positive) and there is no neutral class. Also, the 2.0 version of the data set has several more domains than the four used in the original work [2].

Standard preprocessing techniques (tokenization, stop-word filtering, and stemming) are applied. Afterwards, a representation is obtained using TF-IDF. The final representation is obtained by applying LSA [6] (using the $n = 100$ most important components, as recommended by scikit-learn documentation [7]).

Regarding the evaluation of the methods, a 10-fold cross-validation approach is utilized, with folds fixed within a domain for the different algorithms. In particular, the overall accuracy obtained by each one of the classifiers generated by the 10-fold cross validation is used as the main metric. Even though accuracy is a simple metric and does not take into account all the aspects of the classification, it allows obtaining a useful estimate to evaluate the performance attained.

The implementation of the methods was carried out in Python using the scikit-learn library [7]. Naive Bayes (NB), Linear Support Vector Classification (SVC) and Ordinal Logistic Regression (LR) methods (adapted for the multi-class case) are used as a comparison baseline. While both NB and SVC are not specialized in ordinal classification, their results on ordinal data sets are competitive with those obtained using more specialized methods, thus they provide a starting point for evaluation. These two methods have been selected because of their wide use in sentiment analysis [8]. On the other hand, LR has been selected due to its innate ability to handle ordinal data sets.

When this method was first introduced, it was only tested against nominal classifiers (NB and SVC) and the selected architectures could have been improved. Thus, this work seeks to improve previous works [5]. Regarding the BCOC method, two architectures are used: NB-LR and LR-LR. The first uses NB classifiers in the lower level and an LR classifier in the superior level (in contrast with the original paper that used a linear support vector machine for the superior level). The second architecture will use both LR in the lower level and the superior level. It is important to note that logistic regression was not evaluated in the previous work, and thus, its effects remain unknown. As in the original proposal [5], the inferior level classifiers are parameterized in the same way as individual classifiers used in the baseline. The superior level classifier is parameterized with the default values of the scikit-learn library.

4 Results and discussion

Table 1 shows the results obtained on the data sets provided by Blitzer et al. [2]. For an easier presentation of results, the data sets have been grouped into four distinct categories:

- **Data sets with less than 1k instances**: for these data sets, the analysis of results shows no improvements in the case of *automotive* and *tools_&_hardware*. However, there are marginal improvements when the BCOC method is applied in the case of *musical_instruments* and *office_products*. However, due to the small size of these data sets, the variability of these accuracy results makes them less reliable.
- **Data sets with more than 1k instances and less than 5k instances**: for these data sets there are mostly improvements in the results when using the BCOC approach, however this increase of the average accuracy is, in general, not statistically significant. In particular, improvements were found in the *baby*, *beauty*, *cellphones & service*, *gourmet food*, *grocery* and *outdoor living* data sets. On the other hand, for the data sets of *jewelry & watches*, *magazines*, and *software* no improvements were found using the proposed BCOC architecture.
- **Data sets with more than 5k instances and less than 10k instances**: for these data sets, the analysis of results found mostly marginal improvements. In particular, the *apparel*, *camera & photo* and *computer & video games* data sets showed slight improvements in accuracy, while the *health & personal care* and *sports & outdoors* did not present favorable results.
- **Data sets with more than 10k instances**: for these data sets, all the accuracy results (*dvd*, *electronics*, *kitchen & housewares*, *music*, *toys & games* and *video*) showed at least marginal improvements when using the proposed BCOC architecture.

Table 1. Accuracy (%) results obtained for the baseline and the proposed BCOC architectures on the different data sets.

Data sets	Size	Baseline			BCOC architectures	
		Naive Bayes	Linear SVC	LR	NB-LR	LR-LR
apparel	9246	64.54±1.99	68.47±2.09	69.53±1.43	64.65±1.54	**70.12±1.49**
automotive	736	55.03±7.22	62.75±5.53	**63.18±5.28**	57.09±6.32	63.05±3.76
baby	4256	58.18±2.12	63.98±1.68	64.59±2.09	58.76±1.90	**65.11±3.56**
beauty	2884	66.92±2.87	71.15±1.78	71.21±2.53	66.82±2.41	**71.84±1.81**
camera & photo	7408	60.92±1.45	65.24±1.63	65.97±2.41	60.65±0.99	**66.71±1.65**
cell phones & service	1023	42.32±4.35	58.74±6.41	57.67±3.67	40.86±7.62	**59.34±4.56**
computer & video games	2771	62.61±2.75	64.42±3.28	64.81±2.84	62.61±2.74	**65.10±3.40**
dvd	124438	60.60±0.48	63.03±0.45	65.31±0.40	60.60±0.29	**65.41±0.36**
electronics	23009	54.04±1.2	61.00±1.16	62.05±0.92	54.11±1.07	**62.67±0.68**
gourmet food	1575	72.64±3.93	74.73±3.65	74.16±5.55	72.89±3.51	**74.28±3.33**
grocery	2632	69.53±3.26	70.82±2.36	71.54±2.74	70.02±2.52	**72.23±1.69**
health & personal care	7225	57.11±2.00	62.96±2.61	**64.66±1.55**	57.20±1.90	64.55±1.42
jewelry & watches	1981	59.06±3.22	63.30±1.83	**64.01±4.20**	59.77±3.28	63.81±1.58
kitchen & housewares	19856	58.71±1.12	65.52±1.19	66.68±0.57	58.69±0.61	**66.72±1.09**
magazines	4191	59.51±2.23	64.61±2.01	**65.81±2.39**	58.79±2.45	65.55±2.89
music	174180	70.19±0.38	70.21±0.26	71.39±0.20	70.19±0.18	**71.55±0.38**
musical instruments	332	57.17±8.91	55.12±10.66	57.21±4.81	54.85±8.53	**61.45±8.29**
office products	431	65.18±6.73	62.43±7.99	**65.66±9.11**	65.66±5.12	65.65±4.57
outdoor living	1599	57.53±2.84	64.35±3.89	64.91±4.30	57.35±3.91	**66.04±3.45**
software	2390	41.21±4.43	58.20±2.72	**60.33±3.14**	40.71±4.47	59.71±1.98
sports & outdoors	5728	56.32±1.61	62.29±2.42	**63.55±2.04**	56.30±2.67	63.50±1.58
tools & hardware	112	66.89±15.57	**88.56±15.8**	64.77±13.57	68.71±11.30	64.09±13.81
toys & games	13147	56.41±1.28	63.35±1.46	63.56±1.49	56.60±1.28	**64.10±0.51**
video	36180	62.39±0.91	66.73±0.56	67.74±0.88	62.36±1.00	**67.84±0.87**

While differences were mostly marginal and not statistically significant (e.g., LR vs. LR-LR using a t-test for *apparel*), the majority of the best results were given to the BCOC architectures (LR-LR), with basic logistic regression as the runner-up. This result is expected since logistic regression is a decent method for ordinal classification, but on the other hand, BCOC acts as a small ensemble based on logistic regression. As an ensemble of logistic regression units, the BCOC architecture could be interpreted as a pseudo neural network, with some previous knowledge imparted on its weights (i.e., the ordinal structure of the classes through their associated vertices). Under the assumption that this representation helps when representing ordinal data, this should perform relatively well when compared to another scheme that does not consider this information.

5 Conclusions

In this work, we have provided an empirical evaluation of the BCOC method using two architectures in its basic one-vs-all approach, one using Naive Bayes in the lower level classifier and another one using Logistic Regression for its probability estimations. Both of these architectures used Logistic Regression for the high-level classifier. Experiments have been carried out on multi-domain data sets from the field of sentiment analysis. It has been shown that this proposal yields competitive results in multiple domains and in some cases superior results.

The BCOC method is based on a combination of the probabilities from multiple one-vs-all classifiers. Thus, different class distributions may affect the classification probabilities and in turn, might produce different results. This issue has yet to be addressed since in its current state it would seem that the BCOC method that the distribution of the classes is biased toward neutral classes.

It should be noted that several challenges remain, such as exploring the underlying assumptions of the method and defining formal framework to justify the usage of this method. In this context, the relationship between BCOC and other classifiers and machine learning methods must also be studied, considering its similarity to a neural network when using an LR-LR architecture.

Considering the results from previous work [5], it is believed that further experimentation is required in order to determine the applicability of this proposal, but as mentioned before, the insight behind the LR-LR architecture and its similarity to neural networks could provide a starting point for further research.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. vol. 10, pp. 2200–2204 (2010)
2. Blitzer, J., Dredze, M., Pereira, F., et al.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: ACL. vol. 7, pp. 440–447 (2007)
3. Coxeter, H.S.M., Coxeter, H.S.M., Coxeter, H.S.M., Coxeter, H.S.M.: Introduction to geometry, vol. 136. Wiley New York (1969)
4. Huhn, J.C., Hullermeier, E.: Is an ordinal class structure useful in classifier learning? International Journal of Data Mining, Modelling and Management 1(1), 45–67 (2008)
5. Keith, B., Meneses, C.: Barycentric coordinates for ordinal sentiment classification (2017)
6. Liu, B.: Opinion mining and sentiment analysis. In: Web Data Mining, pp. 459–526. Springer (2011)
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12(Oct), 2825–2830 (2011)
8. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowledge-Based Systems 89, 14–46 (2015)
9. Sánchez Monedero, J.: Challenges in ordinal classification: artificial neural networks and projection-based methods (2014)
10. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). vol. 1631, p. 1642. Citeseer (2013)
11. Wang, D., Zhai, J., Zhu, H., Wang, X.: An improved approach to ordinal classification. In: International Conference on Machine Learning and Cybernetics. pp. 33–42. Springer (2014)