Title
A neutral zone classifier for three classes with an application to text mining

Authors
Friel, Dylan C
Li, Yunzhe
Ellis, Benjamin
et al.

Publication Date
2023

DOI
10.1002/sam.11639

A neutral zone classifier for three classes with an application to text mining

Dylan C. Friel, Yunzhe Li, Benjamin Ellis, Daniel R. Jeske, Herbert K.H. Lee, Philip H. Kass

Abstract

A classifier may be limited by its conditional misclassification rates more than the overall misclassification rate. In the case that one or more of the misclassification rates are high, a neutral zone may be introduced to lower, and possibly balance, conditional misclassification rates. A neutral zone is incorporated into a three-class classifier with its region determined by controlling conditional misclassification rates. The neutral zone classifier is illustrated with a text mining application whose aim is to classify written comments associated with student evaluations of teaching.

Keywords: classification, neutral zone, text mining, sentiment analysis, Word2Vec

1. Introduction

Classification of observations into groups is an objective for many applications. For example, patients might be classified as diseased or not, or loan applications might be classified as high risk or not. While classification problems often involve only two categories, any number of classes may be of interest. A common procedure for classification is to obtain the probabilities that an observation belongs to each of the possible classes, and then assign it to the class with the largest probability. A drawback of the hard boundaries used in this approach is the forced classification of ambiguous observations into a specific class. Introduction of a “neutral zone” between the hard classification boundaries delays a formal decision. If an observation falls in the neutral zone, it does not immediately receive classification into one of the classes. Instead, it will be labeled “neutral.” A practitioner may subsequently engage in follow-up investigations of the observations that lie in the neutral zone before making a final classification decision.

The development of a neutral zone in the context of a classification problem has been explored in several works. A neutral zone was proposed in the context of two classes factoring in the costs of misclassification [1-2]. Alternatively, a neutral zone was proposed for the same context but with the aim of controlling both the false positive and false negative error rates [3-4]. Beyond two classes, there is framework for a three-class neutral zone that can be created based on the cost of misclassification [5].

A consequence of the implementation of a neutral zone is the recognition that follow-up is necessary to reach a final classification decision for observations that fall into the neutral zone. While this reduces misclassifications overall, the follow-up investigation adds cost to the overall decision. The accuracy-cost tradeoff depends on consequences of making misclassifications, which in healthcare applications, for example, are typically severe.

When evaluating the quality of a classifier, the overall misclassification rate may mask underlying weakness. For example, when the classes are not balanced, the overall misclassification rate may be low if observations are predominantly classified in the largest class. It is important to analyze the conditional misclassification rates, especially when classes are not balanced, to assess the overall performance of a potential classifier. When using a neutral zone classifier, it is also important to quantify the rates associated with making neutral zone assignments.

In this paper, we develop a three-class neutral zone classifier that meets specified criteria for conditional misclassification rates. We will aim for balance in the conditional misclassification rates, but other choices may be implemented. We make no assumptions about the class-conditional distributions. The rest of the paper is outlined as follows. Section 2 presents the motivating application for the development of a three-class neutral zone classifier. In Section 3, we present the formulation of the classifier. Section 4 returns to our motivating application and shows successful implementation of the classifier. Finally, Section 5 offers a summary of the work presented in this paper.

2. Motivating Application

The motivating application for this work is to classify written comments associated with student evaluations of teaching as reflecting positive, negative, or mixed feelings about a student's overall experience in the class. The data we use are comments written by undergraduate students for teaching evaluations at the University of California, Santa Cruz (UCSC) and the University of California, Riverside (UCR). Student evaluations are an important factor when evaluating the performance of instructors. These evaluations consist of both Likert scale questions as well as open-ended questions where the student may leave comments in their own words. The effectiveness of the Likert-scale questions has been greatly researched [6-9]. The scope of topics covered in written comments left by students is examined in Ross et al. [10]. The focus of our work is on assessing the overall sentiment of the comments left by students.

Whereas numerical evaluations from instructor reviews are presented in summary form, written comments are typically presented verbatim in the order in which they were recorded. There may be hundreds of evaluations from a single course. The reviewer of the comments is left on their own to extract the overall message of the comments. In the worst cases, the written comments can be glossed over or cherry picked to support an established narrative. If the comments could be classified before the review, they could be sorted to assist the reviewer in getting a more representative understanding of the comments. While we strongly recommend that reviewers read all comments in an evaluation, a sorted ordering would make the review process more systematic and efficient.

There are a wide variety of comments that may appear in instructor evaluations. We define three major categories in which these comments may be classified: positive, negative, other. For "positive" comments, the overall interpretation of the comment is that the instructor is doing a great job and the evaluator would recommend this instructor to other students. A "negative" comment conveys that the instructor is not doing a good job and the student would not recommend this instructor to other students. “Other” comments are those that are decidedly mixed with both positive and negative remarks or comments that seem to provide no evaluation of the instructor.

Our data set comprises 104,143 comments from evaluations conducted at UCSC and 34,749 comments from evaluations at UCR. The courses where these comments originated were medium to large enrollment undergraduate classes in both STEM and non-STEM fields and were taught between Fall 2018 and Summer 2021. To obtain the true label for each comment, a team of three undergraduate students was employed, and the label was determined via majority rules voting. Initially, two students read and labeled each comment. If the labels from the two students agreed, it was assigned as the true label. If the two students disagree, a third student was summoned to give a label. If the third student’s label agrees with one of the first two, it was assigned as the true label. If there is disagreement between all three students, a graduate student researcher made the final determination of the true label. The comments from UCSC resulted in approximately 63% positive comments, 13% negative comments, and 24% other comments. The comments from UCR were approximately 66% positive, 15% negative, and 19% other.

After obtaining the true labels, C, we used a multinomial logistic regression model to build a classifier [11]. Details will be explained further in Section 4, including how to extract numerical features from the text comments. Let p_0, p_1, and p_2 denote the predicted probabilities of the classes negative, positive, and other, respectively. The standard logistic regression classifier would be defined as

$$\hat{C} = \begin{cases} \text{Negative} & p_0 > p_1 \text{ and } p_0 > p_2 \\ \text{Positive} & p_1 > p_0 \text{ and } p_1 > p_2 \\ \text{Other} & p_2 > p_0 \text{ and } p_2 > p_1 \end{cases}$$

The results from such a classifier applied to the UCSC and UCR data (fit on training data and applied to an independent test set of data) are presented in Table 1 and Table 2, respectively. While the overall misclassification rates are about 20%, it can be seen in each that there is a large difference in the (bolded) class-conditional misclassification rates. This is undesirable in the context of instructor evaluations. Namely, we would not want misclassified positive comments to inflate the number of negative comments just as we would not want a misclassified negative comment to go unnoticed by being classified as a positive comment. Our goal for this application is to incorporate a neutral zone to improve the balance in the conditional misclassification rates as well as reduce the overall misclassification rate.

Table 1: Class-conditional classification rates for a standard logistic regression classifier on comments from student evaluations of teaching at UCSC.

	Predicted Label			Conditional Misclassification
True Label	Positive	Negative	Other	Rate
Positive	0.921	**0.021**	**0.058**	**0.079**
Negative	**0.232**	0.515	**0.253**	**0.485**
Other	**0.277**	**0.127**	0.596	**0.404**
Overall Misclassification Rate				0.212

Table 2: Class-conditional classification rates for a standard logistic regression classifier on comments from student evaluations of teaching at UCR.

	Predicted Label			Conditional Misclassification
True Label	Positive	Negative	Other	Rate
Positive	0.927	**0.026**	**0.047**	**0.073**
Negative	**0.201**	0.602	**0.197**	**0.398**
Other	**0.401**	**0.218**	0.381	**0.619**
Overall Misclassification Rate				0.224

3. Neutral Zone Classifier

The proposed classifier assumes that the probability that an observation belongs to each class has been obtained, and these probabilities sum to one. While our application utilizes a multinomial logistic regression model for this purpose, the probabilities can be obtained in a variety of other ways including a neural network classifier or a Bayes classifier. Traditionally, an observation would be assigned to the class with the largest probability. A drawback of this method is the hard boundary when the probabilities for each class are close. Consider a simple example where the probabilities that an observation belongs to each class are 0.35, 0.32, and 0.33. The traditional classifier would assign the observation to the class with probability 0.35. However, with such ambiguity the observation might easily belong to any category, and we can expect a high probability of misclassification. The introduction of a neutral zone creates regions between classes, identifying and labeling these borderline observations as "neutral." The observations that fall into the neutral zone are left for further investigation through follow-up. We next explore the alternatives for constructing the neutral zone boundaries.

3.1 Symmetric Boundaries

Yu et al. [5] developed a minimum cost neutral zone classifier for three classes where a neutral zone region between class boundaries is uniformly created by a single constant, L. The experimenter determines L based on the cost of misclassification. While we do not know the cost of misclassification in our current setting, we can take this same approach but choose L to achieve desired misclassification rates. Letting N denote the label for the neutral zone, the symmetric neutral zone classifier is defined as

$$\hat{C} = \begin{cases} 0 & p_0 > p_1 + L \text{ and } p_0 > p_2 + L \\ 1 & p_1 > p_0 + L \text{ and } p_1 > p_2 + L \\ 2 & p_2 > p_0 + L \text{ and } p_1 > p_1 + L \\ N & \text{otherwise} \end{cases} \tag{1}$$

If $L \in [0,1]$ starts at zero and is increased toward one, the conditional misclassification rates go to zero. Therefore, if we find the first L such that $P\left(\hat{C} = i \middle| C = j\right) \leq \alpha_{ij}$ for $i, j = 0,1,2$ and $i \neq j$, then there always will be a solution. The optimal L is the smallest L such that each conditional misclassification rate is less than or equal to its target size. Figure 1a sketches the general shape of the symmetric neutral zone classifier. While this symmetric approach allows a uniform upper bound on the conditional misclassification rates, it generally will not produce a classifier with balanced conditional misclassification rates.

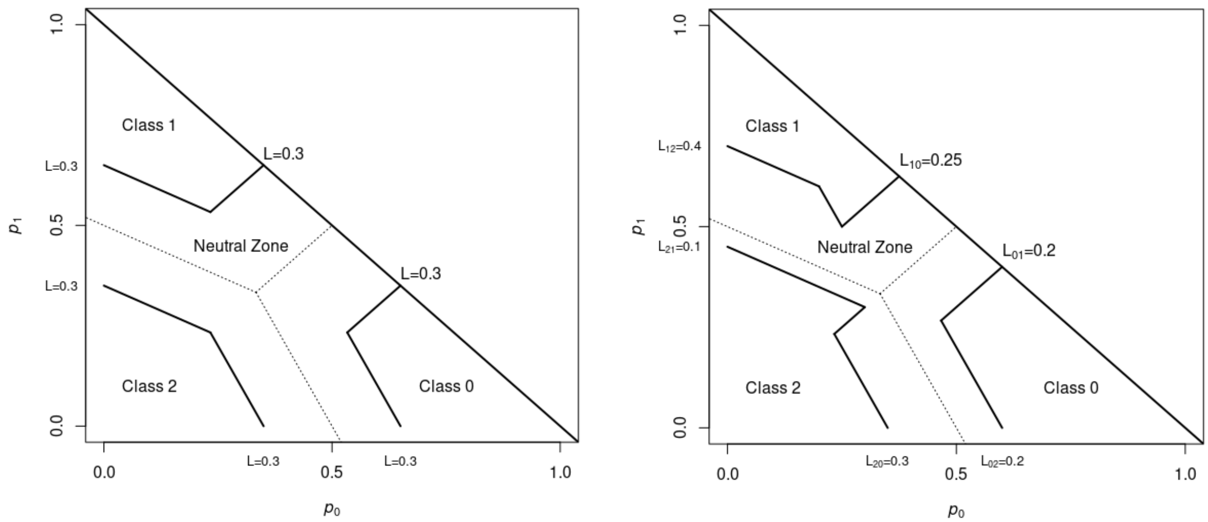
3.2 Asymmetric Boundaries

Rather than using a single L to define neutral zone regions, an alternative is to separately choose an L for each pairwise decision boundary individually. Namely, we define the asymmetric neutral zone classifier as

$$\hat{C} = \begin{cases} 0 & p_0 > p_1 + L_{01} \text{ and } p_1 > p_2 \text{ or } p_0 > p_2 + L_{02} \text{ and } p_2 > p_1 \\ 1 & p_1 > p_0 + L_{10} \text{ and } p_0 > p_2 \text{ or } p_1 > p_2 + L_{12} \text{ and } p_2 > p_0 \\ 2 & p_2 > p_0 + L_{20} \text{ and } p_0 > p_1 \text{ or } p_2 > p_1 + L_{21} \text{ and } p_1 > p_0 \\ N & \text{otherwise} \end{cases} \tag{2}$$

where $L_{ij} \in [0,1]$ is the size of the neutral zone when deciding on class i over class j. If all $L_{ij} = 0$, then $\hat{C}$ is the traditional classifier that has no neutral zone.

Otherwise, for an observation to be assigned a class, the probability of that class must be larger than the probabilities of the other two classes by margins defined by their respective L_{ij}'s. Each L_{ij} only affects the classification boundary when class j is the second-most likely category after class i. Figure 1b sketches the general shape of the asymmetric neutral zone classifier.



(a) (b)

Figure 1: Symmetric (a) and Asymmetric (b) neutral zone classifiers. Dotted lines represent the boundaries of the no neutral zone classifier.

It can be verified that the geometrical area of the neutral zone as a proportion of the entire classification region is $\sum_{\mathrm{i}} \sum_{\mathrm{j}} \frac{\mathrm{L_{ij}}}{12}\left(2 - \mathrm{L_{ij}}\right)/(1/2)$ for $i, j = 0,1,2$ and $i \neq j$. This area may be used to roughly compare the size of alternative neutral zone classifiers. However, this area differs from the proportion of observations that fall within the neutral zone due to the fact that the latter depends on the underlying class-conditional distributions of the features. Notwithstanding that, Table 3 displays the area in terms of the proportion of the entire classification region for different choices of $L_{ij} = L$, i.e. the single L scenario which generates a symmetric neutral zone.

Table 3: Proportion of classification region that is taken up by the neutral zone for cases when $L_{ij} = L$.

L	Area of Neutral Zone
0.05	0.0975
0.10	0.1900
0.25	0.4375
$1 - 1/\sqrt{2}$	0.5000
0.50	0.7500

3.3 Controlling Conditional Misclassification Rates

Conditional misclassification rates of the proposed neutral zone classifier can be controlled by selecting L_{ij} such that $P(\hat{C} = i | C = j) \leq \alpha_{ij}$. If $\alpha_{ij} = \alpha$, for all i, j, and some constant α, better balance of the conditional misclassification rates will be achieved. As the L_{ij}'s approach one, the conditional misclassification rates go to zero. Both the symmetric and the asymmetric neutral zones will either give the same predicted class as the traditional classifier or change the predicted class to "neutral." Thus, no new misclassifications are introduced by using the neutral zone classifier. For each i, the pair of L_{ij}'s are found jointly since a single L_{ij} affects only two of the six conditional misclassification rates. For example, (L_{01}, L_{02}) are found from the equations $P(\hat{C} = 0 | C = 1) \leq \alpha_{01}$ and $P(\hat{C} = 0 | C = 2) \leq \alpha_{02}$ and similarly for (L_{10}, L_{12}) and (L_{20}, L_{21}). When there is more than one solution, the practitioner can select the one that minimizes the overall probability of a neutral zone classification.

3.4 Grid Search

A straightforward approach to finding the L_{ij} is to use a grid search as follows. First, the p_0, p_1, and p_2 probabilities are obtained for all the observations in the training data set. As explained in the previous section, we find the L_{ij} two at a time. Consider the case of finding L_{01} and L_{02}. For each (L_{01}, L_{02}) on a unit grid, use the predicted classes for the training data to estimate $P(\hat{C} = 0 | C = 1)$ and $P(\hat{C} = 0 | C = 2)$. Then choose the (L_{01}, L_{02}) that gives the conditional misclassification rates closest to α_{01} and α_{02} without exceeding them. Perform this same search similarly to find (L_{10}, L_{12}) and (L_{20}, L_{21}).

3.5 Feature Space Representation

In some situations, the classifier in Equation (2) can be inverted to display the decision boundaries in the feature space. We illustrate this in a Bayes classification setting with two dimensions.

Let π_i represent the prior class probabilities. Suppose the features in each class follow the probability density function $f_i(x)$. Then $p_i = \pi_i f_i(x) / \sum_{j=0}^{2} \pi_j f_j(x)$ are the posterior class probabilities. These probabilities are used in $\hat{C}$ from Equation (2) to obtain the predicted classes. Let A_0, A_1, A_2, and A_N denote the regions in the feature space that correspond to the predicted labels 0, 1, 2, and N, respectively, from Equation (2).

$$A_0 = \{x: p_0 > p_1 + L_{01}, p_1 > p_2\} \cup \{x: p_0 > p_2 + L_{02}, p_2 > p_1\}$$

$$A_1 = \{x: p_1 > p_0 + L_{10}, p_0 > p_2\} \cup \{x: p_1 > p_2 + L_{12}, p_2 > p_0\}$$

$$A_2 = \{x: p_2 > p_0 + L_{20}, p_0 > p_1\} \cup \{x: p_2 > p_1 + L_{21}, p_1 > p_0\}$$

$$A_N = \overline{A_0 \cup A_1 \cup A_2}$$

The six conditional misclassification probabilities are calculated as

$$P(\hat{C} = i | C = j) = \int_{A_i} f_j(x)\ dx, \quad i, j \in \{0,1,2\}, i \neq j$$

In addition, the conditional neutral zone rates are given by

$$P(\hat{C} = N | C = j) = \int_{A_N} f_j(x)\ dx, \quad j \in \{0,1,2\}$$

Although the regions of integration are complex, the integrals can be evaluated easily using simulation techniques. We can use the simulated distributions in the grid search explained in Section 3.4 to estimate the conditional misclassification rates and determine the L_{ij}.

When the $f_i(x)$ are bivariate normal, the regions A_0, A_1, A_2, and A_N can be graphed in the feature space. This is demonstrated in Figure 2 and 3. The linear boundaries in Figure 2 are a consequence of assumed equal covariance matrices, while the spherical boundaries in Figure 3 are a consequence of unequal, but diagonal, covariance matrices.

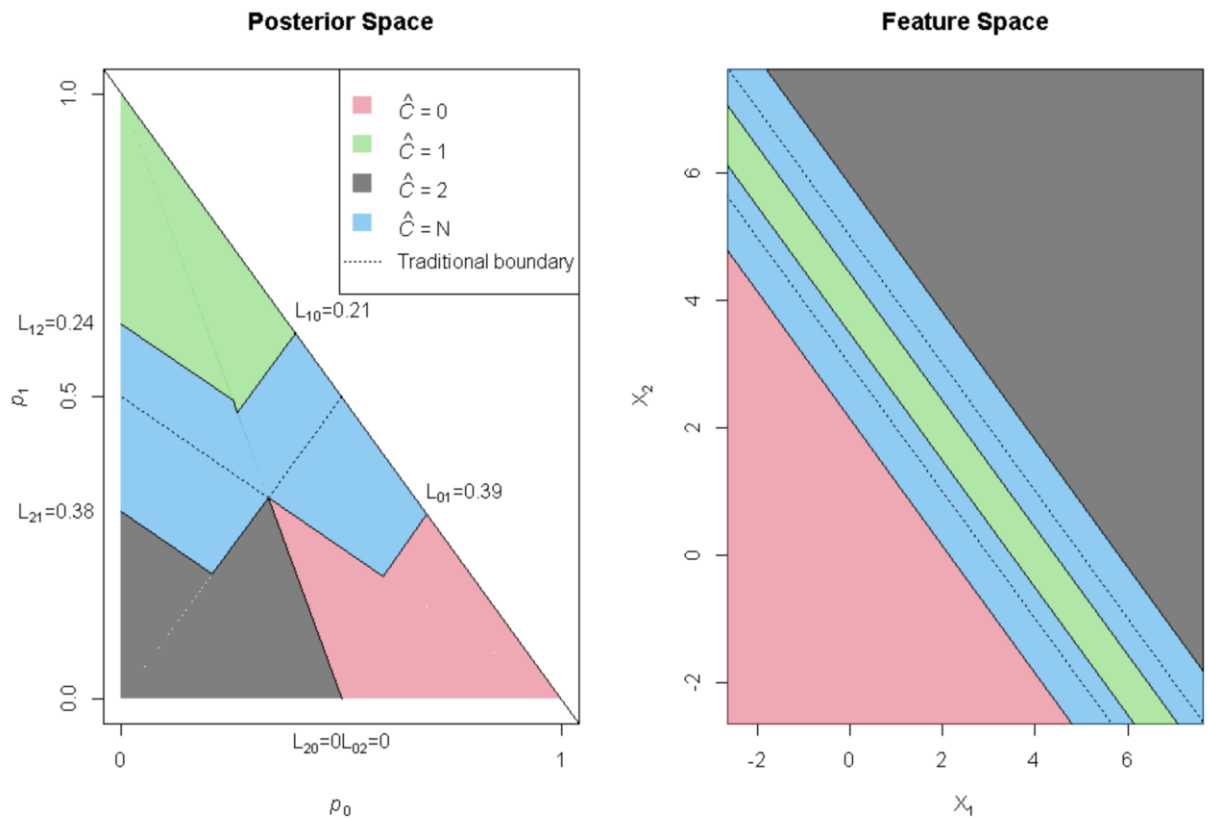


Figure 2: Neutral zone in the feature and posterior space for $X \sim N(\vec{\mu}_i, \Sigma)$ where $\vec{\mu}_0 = (1,1)$, $\vec{\mu}_1 = (3,3)$, $\vec{\mu}_2 = (5,5)$, $\Sigma_0 = \Sigma_1 = \Sigma_2 = I_2$, and $\pi_i = 1/3$ for $i = 0,1,2$. The L_{ij} are found to give conditional misclassification probabilities less than or equal to 0.1. The dotted lines show the no neutral zone classifier boundaries.

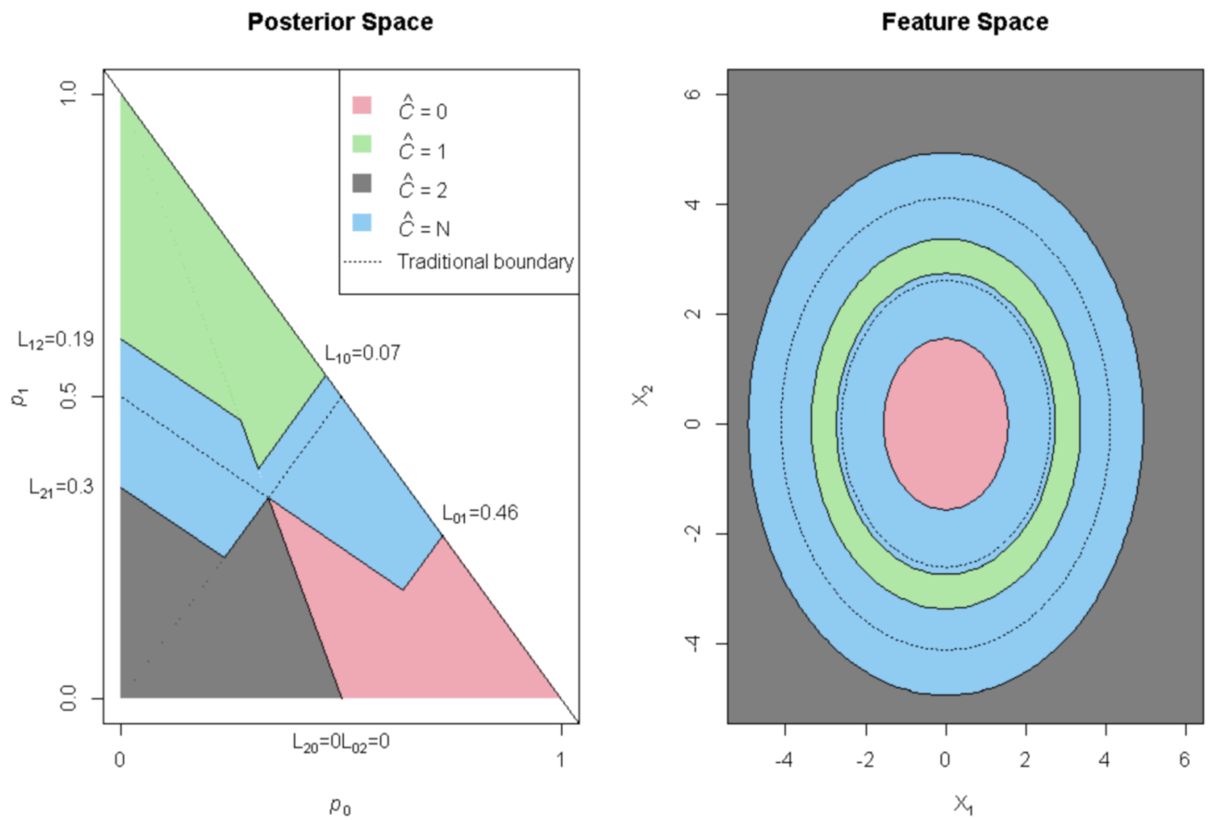


Figure 3: Neutral zone in the feature and posterior space for $X \sim N(\vec{\mu}_i, \Sigma)$ where $\vec{\mu}_0 = (1,1)$, $\vec{\mu}_1 = (3,3)$, $\vec{\mu}_2 = (5,5)$, $\Sigma_0 = I_2$, $\Sigma_1 = 2 * I_2$, $\Sigma_2 = 3 * I_2$, and $\pi_i = 1/3$ for $i = 0,1,2$. The L_{ij} are found to give conditional misclassification probabilities less than or equal to 0.1. The dotted lines show the no neutral zone classifier boundaries.

4. Example Application

4.1 Word2Vec

We return to our motivating application from Section 2. When working with text data, we first need to transform the text into numeric values. We investigated several directions to achieve this and found Word2Vec to be the most effective for our purposes [12-14]. The main purpose of Word2Vec is to try to predict words that are written together [15-17]. A key step in that process is mapping of words to numerical features. Word2Vec is a mapping based on a neural network and was originally proposed with a choice of two algorithms: continuous bag-of-words (CBOW) and skip-gram. In this paper, we will focus on the former. The CBOW algorithm attempts to predict a “center” word based on the given “context” words.

A visual map of the CBOW algorithm is displayed in Figure 4. When training the CBOW algorithm, we move through each word in each comment, treating it as the center word, w. The context words are determined in a window around w. The window size is inputted by the user. The input layer of the neural network consists of one-hot vectors $b_1, b_2, \dots, b_c$ representing the context words. The one-hot vectors have length d, where d is the number of words in the entire corpus of comments and are zero everywhere except for a 1 at the position of the word in a dictionary formed from the corpus. These input vectors are used to extract rows from a to-be-determined $d \times m$ matrix, W, where m is inputted by the user. An element-wise sum on the extracted rows creates the latent vector u_w. Then matrix multiplication is performed with u_w and another to-be-determined matrix U. The result is a vector, v, which is inputted to a softmax function that uses a normalization transformation to create a vector, p, of length d that consists of the probability that each word is the center word. This vector of probabilities is used with the one-hot vector of the true center word to compute the loss function. The cumulative loss is an aggregation of the loss from performing this process with each word as the center word. The fitting process solves for the W and U that minimize the aggregated loss.

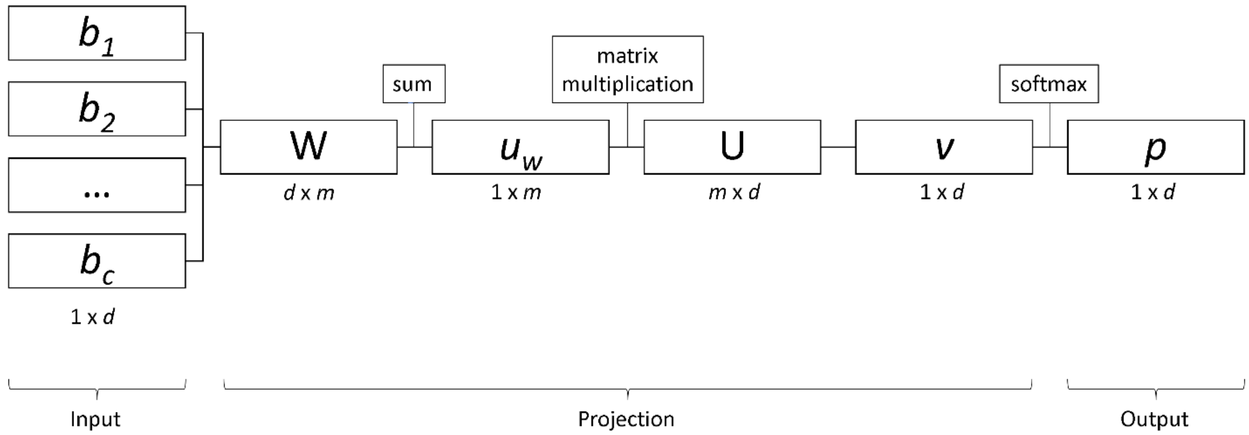


Figure 4: Map of the CBOW algorithm used in Word2Vec

There are several hyperparameters that may be adjusted in Word2Vec. These include the window size – number of words to consider around the center word, number of features – length of the latent vector for each word, and occurrence threshold – number of times a word must occur to be considered one of the d words. We have used the default values recommended for each of these in the R package word2vec [18].

In our application, we do not need to predict words given context words. Instead, we can extract the matrix of word embeddings, W, from within the projection layer of Word2Vec. With the fitted W, we have a matrix where each row represents a word in our corpus and the columns represent numerical features. For each word in a comment, we extract the corresponding rows of W to get a matrix of word embeddings for the comment. After normalizing the vector of column sums to account for the length of the comment, we obtain a numeric vector of length m that can be used as the features in a multinomial logistic classifier. In the following sections, we use 5-fold cross-validation to evaluate the accuracy of the classifier. Each training set is used to fit the Word2Vec model, fit the multinomial logistic regression model, and find the L_{ij}'s.

4.2 UCSC Data

First, we analyze 104,143 comments from instructor evaluations at the University of California, Santa Cruz. Recall that the comments labeled as “negative,” “positive,” and “other” have been defined as Class 0, Class 1, and Class 2, respectively. The results of a largest-probability classifier using multinomial logistic regression were presented earlier in Table 1. In this section, we incorporate an asymmetric neutral zone to the same data to lower the conditional misclassification rates. Setting each $\alpha_{ij} = \alpha = 0.1$ with an 80-20 train-test split of the data, we get the L_{ij}’s seen in Figure 5. The points plotted in this figure are the 20,828 observations from test set. Approximately 20% of the test set falls into the neutral zone.

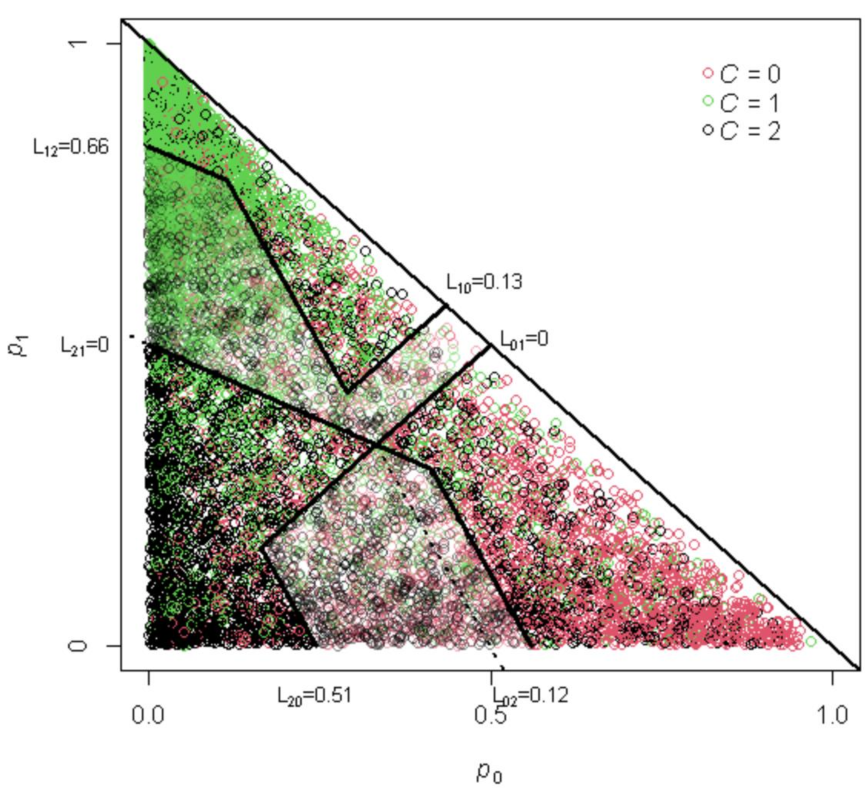


Figure 5: Asymmetric neutral zone applied to a test set of comments from UCSC. Neutral zone is indicated by the transparent points. Dotted lines show the no neutral zone boundaries.

Table 3 displays the 5-fold cross-validation estimates of the conditional misclassification rates for the UCSC data. It clearly shows that all six conditional misclassification rates have been lowered to be less than or equal 0.1. In two cases, the conditional misclassification rates are much lower than the target. As seen in Table 1, these two values were lower than the target before incorporating the asymmetric neutral zone, which explains why the corresponding L_{ij}'s are zero. Notice that the overall misclassification rate of the classifier has been reduced from about 20% to about 10% by employing the neutral zone. The improved accuracy is the result of approximately 20% of the comments getting classified as neutral because they are too ambiguous to be confidently assigned to any of the classes.

Table 3: Asymmetric neutral zone classifier applied to classification of comments from student evaluations of teaching at UCSC.

	Predicted Label				Conditional
True Label	Positive	Negative	Other	Neutral	Misclassification Rate
Positive	0.763	0.017	0.045	0.174	0.063
Negative	0.100	0.459	0.097	0.344	0.196
Other	0.100	0.100	0.470	0.330	0.200
Overall Rate				0.234	0.114

4.3 UCR

Next, we incorporate the asymmetric neutral zone into the multinomial logistic classifier from the 34,739 comments from University of California, Riverside. Again, we set each $\alpha_{ij} = \alpha = 0.1$ with an 80-20 train-test split of the data and obtain Figure 6. Approximately 30% of the 6,948 test set observations fall into the neutral zone.

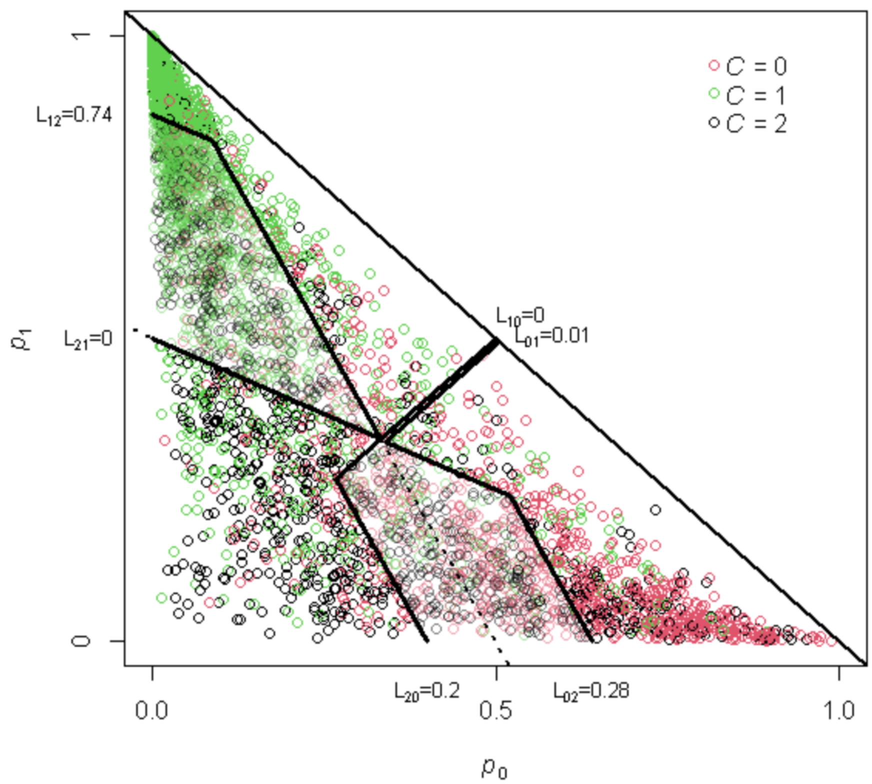


Figure 6: Asymmetric neutral zone applied to a test set of comments from UCR. Neutral zone is indicated by the transparent points. Dotted lines show the no neutral zone boundaries.

In Table 4, we present the results of 5-fold cross-validation for the asymmetric neutral zone classifier applied to UCR data. All conditional misclassification rates have been lowered appropriately to be less than or equal to the target value. As with the UCSC data, there are two instances where the conditional misclassification rate is much lower than the target. This is a result of these rates being lower than the target α before the neutral zone was implemented and gives one of the L_{ij}'s a value of zero. Notice that while the conditional misclassification rates are roughly comparable for the two campuses, the set of L_{ij}'s needed to achieve that are different. The UCR data leads to slightly more comments being labeled as neutral.

Table 4: Asymmetric neutral zone classifier applied to classification of comments from student evaluations of teaching at UCR.

True Label	Predicted Label: Positive	Negative	Other	Neutral	Conditional Misclassification Rate
Positive	0.734	0.015	0.036	0.215	0.051
Negative	0.078	0.396	0.100	0.427	0.178
Other	0.098	0.103	0.281	0.518	0.201
Overall Rate				0.303	0.098

The main difference between the two universities is how the comments were prompted. The comments at UCR were all responses to a single, broad question asking the student to "comment on how the instructor's teaching helped your learning of the material in this course." On the other hand, UCSC used different, more targeted questions to prompt comments from students. While the fitted classifiers from the two campuses are similar, we recommend that each university develop its own classifier. R code is provided in the supplementary material to create the asymmetric neutral zone classifier from any set of training data.

5. Summary

In this paper, we have developed a neutral zone classifier for the three-class setting that can improve the balance of conditional misclassification rates and lower the overall misclassification rate. No assumptions are necessary about the class-conditional distributions; therefore, this classifier may be employed in any three-class scenario where the probabilities for each class are obtained from any of a variety of methods that create them. There are two major benefits of the employment of a neutral zone: avoidance of misclassifying borderline observations and control of the conditional misclassification rates. The conditional misclassification rates of the neutral zone classifier are never worse than if the neutral zone was not used, and the overall misclassification rate will always be better.

This work was motivated by student comments written for instructor evaluations. We have shown how Word2Vec and multinomial logistic regression may be combined to analyze text data with three classes. The neutral zone classifier in this setting assists a reviewer in the reading of many comments by providing at a glance the frequency of comments that are classified as positive, negative, or other. The predicted labels also allow the comments to be grouped so that they can be presented to the reviewer in sorted order, which aids the selection of a representative sample of the comments for full reading.

REFERENCES

1. Jeske, D. R., Liu, Z., Bent, E., & Borneman, J. (2007). Classification Rules that Include Neutral Zones and Their Application to Microbial Community Profiling. *Communication in Statistics—Theory and Methods, 36*(10), 1965-1980. doi:10.1080/03610920601126514
2. Benecke, S., Jeske, D. R., Ruegger, P., & Borneman, J. (2013). Bayes neutral zone classifiers with applications to nonparametric unsupervised settings. *Journal of Agricultural, Biological, and Environmental Statistics, 18*, 39-52.
3. Jeske, D. R., Zhang, Z., & Smith, S. (2020). Construction, visualization and application of neutral zone classifiers. *Statistical Methods in Medical Research, 29*(5), 1420-1433.
4. Jeske, D. R., & Smith, S. (2018). Maximizing the usefulness of statistical classifiers for two populations with illustrative applications. *Statistical Methods in Medical Research, 27*(8), 2344-2358.
5. Yu, H., Jeske, D. R., Ruegger, P., & Borneman, J. (2010). Neutral Zone Classifiers Using a Decision-Theoretic Approach With Application to DNA Array Analyses. *Journal of Agricultural, Biological and Environmental Statistics, 15*, 474-490.
6. Boring, A., Ottoboni, K., & Stark, P. (2016). Student evaluations of teaching (mostly) do not measure teaching effectiveness. *ScienceOpen Research, 0*(0), 1-11. doi:10.14293/S2199-1006.1.SOR-EDU.AETBZC.v1
7. Ho, A., Thomsen, L., & Sidanius, J. (2009). Perceived Academic Competence and Overall Job Evaluations: Students' Evaluations of African American and European American Professors. *Journal of Applied Social Psychology, 39*, 389-406. doi:10.1111/j.1559-1816.2008.00443.x
8. Mengel, F., Sauermann, J., & Zolitz, U. (2019). Gender Bias in Teaching Evaluations. *Journal of the European Economic Association, 17*(2), 535-566. Retrieved from https://doi.org/10.1093/jeea/jvx057

9. Miller, J., & Chamberlin, M. (2000). Women Are Teachers, Men Are Professors: A Study of Student Perceptions. *Teaching Sociology, 28*(4), 283-298. Retrieved from https://doi.org/10.2307/1318580

10. Ross, D., Boatright, D., Nunez-Smith, M., Jordan, A., Chekroud, A., & Moore, E. Z. (2017). Differences in words used to describe racial and gender groups in Medical Student Performance Evaluations. *PLoS One, 12*(8). doi:10.1371/journal.pone.0181659

11. Kutner, M., Nachtsheim, C., Neter, J., & Li, W. (2005). *Applied Linear Statistical Models*. Chicago: McGraw-Hill Irwin.
12. Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. New York: Springer New York Inc.
13. Nielsen, F. A. (2011). AFINN. Informatics and Mathematical Modelling, Technical University of Denmark. http://www2.compute.dtu.dk/pubdb/pubs/6010-full.html
14. Rinker, T. W. (2021). Sentiment: Calculate Text Polarity Sentiment version 2.9.0. https://github.com/trinker/sentimentr
15. Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations.*

16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems,* 26.

17. Goldberg, Y. & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722. arXiv:1402.3722. https://doi.org/10.48550/arXiv.1402.3722
18. Wijffels, J. (2021). word2vec: Distributed Representations of Words. R package version 0.3.4. https://CRAN.R-project.org/package=word2vec