Feature selection in text classification: Identifying spurious words with causal inference methods

Zixuan Zhao[1,6], Hengzhuang Li[2,7], Jinxuan Chen[3,8], Yang Li[4,9], Jiayun Song[5,10]

[1]Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
[2]School of Computer Science and Technology, Huazhong University of Science & Technology, Wuhan, China
[3]Department of Information Management and Information System, Central South University, Changsha, China
[4]School of Art and Sciences, The Ohio State University, Columbus, U.S.
[5]WLSA Shanghai Academy, Shanghai, China

[6]uwzhaozixuan@gmail.com
[7]buggistlee@gmail.com
[8]chenjinxuan0811@gmail.com
[9]youngliloveses@gmail.com
[10]silviasong2004@gmail.com

Abstract. As has been scrutinized by many, non-causal model may contain spurious correlations that act like shortcuts during the prediction phase, undermining cross-domain accuracy. This can be caused by biased training data that contains spurious words with neutral meanings yet can induce the model to predict wrongly. Based on this assumption, we propose a series of methods to detect these spurious words before feeding the model with the training data. We used advanced causal inference methods which are arising novas in recent studies, such as propensity score matching and inverse propensity score weighting to facilitate the feature selection before training. We experimented with multiple approaches to estimate propensity scores and got profound improvements. We further experimented with BERT model to evaluate the effectiveness of feature selection and find that the model performance with in-domain and out-of-domain testing samples is boosted after we remove the spurious words detected by our methods in the training data.

Keywords: Sentiment Analysis, Cross-domain Accuracy, Kernel Method, PSM, IPW, BERT.

1. Introduction

Sentiment Analysis or opinion mining is a process that determine emotional tendency of subjective texts, whether it is positive, negative, or neutral [1]. It is commonly used in business or social domain [2]. For example, determining which words have positive or negative meanings can help companies to analyze audiences' responses to movies or books, and thus understand the market trends [3]. There are many sentiment analysis models proposed and have obtained significant sentiment classifier. However, when training sentiment classifier, the model is usually affected by some spurious words, which have neutral meaning but positively or negatively associated with certain document domain [4]. This phenomenon, is part of the confounding effects, will worsen the performance of the prediction model [4]. Additionally, these words are referred to as confounding variables [4]. To reduce the confounding effects in sentiment analysis, our experiment employs feature selection technique for causal inference [5].

Analyzing how words in documents affect sentiment lacks a well-defined control group in nature [6]. This lack of randomly assigned control group for assessing treatment (i.e., words) effects is an important reason to apply causal inference [7]. Causal inference uses a causality argument to calculate where there is a direct causal relationship between two events [6]. It is different from correlation and can increase the accuracy of our analysis on the sentiment of words [6].

Propensity score is a statistical tool that is designed to adjust for the absence of randomization in observational study [8]. It is defined as the probability to be assigned to a treatment group given all the confounding variables [8]. In the context of sentiment classification of texts, it is described as probability that the words appear in a document [6]. One big advantage of using propensity score is its reduction of features space. Instead of directly conditioning on confounding space, propensity score reduces the dimension of confounding space to one single treatment assignment dimension [9].

We experimented with three approaches to calculate the propensity score for each important words we selected from training set. The three approaches are logistic regression, kernel regression and neural network. We then used each set of calculated propensity score resulting from each approach as a metric to measure the causal effects.

There is previous work done by other researchers who use different approaches to assess causal effects utilizing propensity score. In Feature Selection as Causal Inference: Experiments with Text Classification, Michael J. Paul uses propensity scores to match documents one to one in order to probe the difference between two similar sentences differing by their inclusion of the studied word [6]. In Identifying Spurious Correlations for Robust Text Classification, Zhao Wang and Aron Culotta use the BERT model to match words by the similarity of their context to estimate whether a word is genuine or spurious [4]. Their common goal is to study the sentiment of words without being influenced by their contexts.

In our experiment, we used two methods—Propensity score matching (PSM) [10]and Inverse propensity weighting (IPW) to measure each important word’s causal effect. We identified spurious words based on quantile and removed them from original documents for feature selection. After preprocessing of original documents, we fed the documents to pretrained Bidirectional Encoder Representations from Transformers (BERT). The reason why we choose BERT is that it is a sentiment classification model that is feature-based, and fine-tuning based [11], making it compatible with our work with minimal effort for fine-tuning [12]. Propensity score matching (PSM) [10]uses the propensity score, which is the probability of a word appearing in a context. We match words according to this score to eliminate the bias brought by the sentence’s context to the word. Inverse propensity weighting (IPW) [10]gives weight to sentences by how frequently the word appears in a certain context. Contexts in which the word appears more will be given lower weighing to reduce bias [10]. Thus, the higher the average treatment effect (ATE) is, the more differences there are when the word appears and does not appear, indicating that it has more likelihood to be genuine [10]. Lastly, we fine-tuned BERT [11] to match the word with similar semantic meanings [13]. Using it respectively before and after feature selection to see how feature selection will affect the result.

2. Challenges & Motivation

As has been scrutinized by many researchers, a non-causal model may contain spurious correlations that can undermine the model’s generalization leading to poor performance in out-of-domain settings. In our problem setting, the spurious correlations are caused by spurious words that have neutral meanings but play a crucial role in determining the labels of the training set. The spurious words create shortcuts for the model to heuristically predict the labels of the testing samples.

Because of the high probability of the existence of spurious words, we propose to adopt multiple methods to facilitate the process of causal inference, rid the models of spurious correlations, and thus improve the accuracy within the domain and across the domain. In detail, an intuitive solution to the spurious correlations carried by training samples is to rule out the words that have biased meanings in the dataset. In the setting of text classification, words are cast into vectors whose every dimension of the feature space represents one specific word. Therefore, the process of ruling out the spurious words is reducted to feature selection through which we achieve obvious improvement in the model's performance.

3. Methods

3.1. Propensity Scores

The largest benefit we get from using propensity scores is a reduction in feature space. By using only propensity scores, we performed a reduction in dimensionality on the confounder space to a single treatment assignment dimension. As mentioned in other work, propensity scores can represent the probabilities of each word appearing in a document [14]. While almost all previous work focuses on traditional logistic regression, we are interested in exploring the kernel method to calculate propensity scores in this section.

The kernel method means to map the data from the original sample space to a higher-dimensional feature space so that the linearly inseparable data in the sample space becomes linearly separable in the feature space. We have the axiom that if the original space is finite-dimensional, that is, the attributes are limited, then there must be a high-dimensional feature space that makes the data linearly separable.

Assuming that the dimension of our data is m, a kernel function is a symmetric function defined on the m×m dimensional space. We can calculate the kernel matrix by the kernel function with the original data. The kernel matrix is always positive semi-definite which denotes that as long as the kernel matrix with respect to a symmetric function is positive semi-definite, it can be used as a kernel matrix. Meanwhile, for a positive semi-definite matrix, we can always find a corresponding mapping function [15]. Here we will introduce the representer theorem [16].

We experimented with multiple methods to estimate the propensity scores, iteratively named kernel methods, random forest, and neural networks. Kernel methods help to handle non-linear regression problems, which perfectly suits our problem setting, as we assume that the regression problem of predicting the probability a context contains a particular word is most likely to be a non-linear problem. Random forest is also an advanced method to estimate probabilities, and many works have gone through this path. Given the nature of the neural network, it also fits well in our problem setting. The key for neural networks to succeed in estimating the propensity scores is that it acts by outputting the probability of every class.

The Representer Theorem *Let* $\mathbb{H}$ *be the reproducing kernel Hilbert space corresponding to the kernel function* $\mathbb{K}$, $\|h\|$ *be the norm of* h *in the* $\mathbb{H}$ *space, for any monotonically increasing function* $\Omega\colon[0,\infty]\mapsto\mathbb{R}$, *and any non-negative Loss function* $l\colon\mathbb{R}^m\mapsto[0,\infty]$, *we have the optimization problem like (1):*

$$\min_{h\in H} F(h)=\Omega(\|h\|_H)+l(h(x_1),h(x_2),\ldots,h(x_m)) \tag{1}$$

The solution can always be translated as (2):

$$h^*(m)=\sum_{i=1}^{m}\alpha_i\mathcal{K}(x,x_i) \tag{2}$$

The representer theorem has no restrictions on the loss function l and Ω only requires monotonically increasing regularization which doesn't have to be a convex function. For a general loss function and a regularization term, the optimal solution to the optimization problem $h^*(m)$ can be translated as a linear combination of kernel functions $\mathcal{K}(x,x_i)$ which shows the huge superiority of the kernel function.

The representer theorem tells us that the solutions to some regularization functionals in high or infinite dimensional spaces lie in finite-dimensional subspaces spanned by the representers of the data. It effectively reduces the computationally cumbersome or infeasible problems in high or infinite dimensional spaces to optimization problems on the scalar coefficients.

3.2. Causal Inference

Randomized control trails (RCT) are considered crucial to conduct effective measurement of a treatment or intervention [17]. It reduces most bias in examining the causal relationship between the treatment and response by randomly assigning participants into a treated or control group [18]. In text classification, however, this randomization process can't be directly applied since documents or sentences are pre-treated with words [17]. As a result, it is not reasonable to randomly assign words to documents [17]. To resolve this problem and conduct effective causal inference analysis, we experimented with two methods to simulate RCT, both of which made use of propensity score.

The two different approaches to complete this simulated RCT in our experiment were propensity score matching and inverse propensity weighting. We used these two approaches to get a value for each word measuring its causal effect on a document or sentences' sentiment. In IPW, this value represents the average treatment effect (ATE) of the word based on the training dataset. In PSM, this value represents the average treatment effect of the treated group (ATT) of the word based on the training set. Then this measurement of treatment effect was used to identify spurious words by thresholding. The resulting ATEs of important words from airline comments turned out to be very dense in distribution. We plotted multiple percentile lines (95th, 75th, 50th, 25th, 5th) and cutting words beyond 5th percentile gave the most rational result of "genuine" words. Thus, we removed words beyond 5th percentile from the original airline comments as our move towards reducing "spurious" words from data set that causes bias in identify causal relationship.

3.2.1. Propensity Score Matching (PSM)

Our fundamental technique is the propensity score matching (PSM) [10] which allows us to simulate the random assignment of treatment groups (in our experiment, they mean the sentences with the presence of the word) and control groups (the sentences without the presence of the word) by matching them into pairs with similar propensity score. The propensity score is the probability of such a word appearing in certain contexts, and it is calculated using different kernel logistic regression models. PSM helps mitigate negative selection bias by directly comparing the results of treated and control groups [10].

In our experiments, we employ different kernel logistic regression models, such as linear regression model, sigmoid regression model, etc., to calculate the propensity score and thus, compare the matching results.

3.2.2. Inverse Probability Weighting

IPW, compared to PSM, assign weights to each document to balance existence of confounders in both treated and untreated group by creating a pseudo population [17]. As a result, the ATE estimated is the ATE for the pseudo-population [17]. The weight is calculated as the following:

$$w_i = \frac{Z}{e_i} + \frac{(1-Z_i)}{1-e_i} \tag{3}$$

where Z_i indicates whether the document contains the word and e_i represents the propensity score for the document.

Note that in our experiment, the PS is clipped since extreme PSs (very close to 0 or 1) results in large weights using equation (3) above. Also, study showed that truncating large weights following logistic regression can improve accuracy and precision of final ATE estimator [19].However, the study also showed that the amount of truncation is essential. So far, we only used 0.1 as the truncation threshold, which gave a ATE distribution as shown in Figure 1. Some ATEs appeared less compared to the rest, therefore, the density map in Figure 2 was shown as a complement to Figure 1 to give a clearer view. We argue that this ATE estimator is unbiased as below:

$$ATE = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\frac{\mathrm{w_i Y_i}}{\hat{p}_i} - \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\frac{(1-\mathrm{w_i})\mathrm{Y_i}}{(1-\hat{p}_i)} \tag{4}$$

The equation (4) for calculating the average treatment effect (ATE) in IPW is as above. The value w represents whether a word appears in a document or not. If it exists, then w is 1; and if it does not exist, w is 0. The value Y is the label of the document, 1 if positive, and 0 if negative. The value p is the propensity we have calculated, the probability of a word appearing in a certain context. This equation is an unbiased estimator because the expected value of w for every word is exactly p. Thus through this method, we can give weighing to documents according to their context, reducing the effect of this confounding variable, and calculate ATE without introducing additional bias. The larger the value of ATE is, the more difference there is in the meaning of a document depending on the existence of that certain word, meaning the word is more likely to be a genuine word instead of a spurious one.

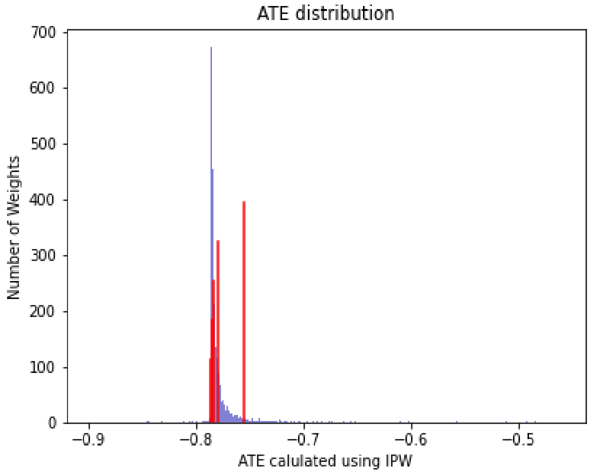

Figure 1. ATE distribution of words trained using movie dataset with quantiles (95th, 75th, 50th, 25th, 5th) shown in red vertical line.

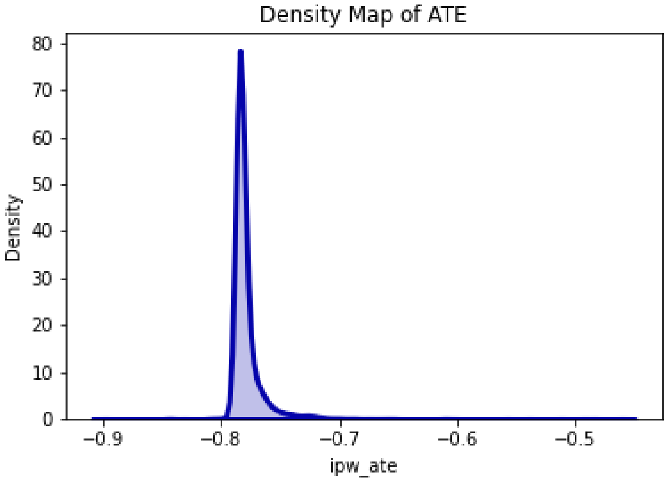

Figure 2. ATE Density Map.

3.3. Sentiment Classifier: BERT

In the experiments, we use BERT [11] as our sentiment classification model to improve the performance on text classification. BERT has achieved amazing results in many natural language processing and natural language understanding tasks [20]. It can consider the information from both sides of a character at the same time and thus capture contextual information [11].

BERT can be pre-trained on Masked Language Task, and pre-trained word embeddings are important for better performance on the tasks. In the process to test the causal relationship, we fine-tuned a 12-layer BERT model to get quicker development and better results. We use 60% data for fine-tuning, 40% for test. In fine-tuning, we use 90% data for training and 10% data for validation. We chose Adam as the optimizer.

Each layer contains a Transformer block with 12 self-attention heads, and the total hidden size of BERT is 768. The maximum size of input is 512. And we added a special token [CLS] as the first token of a sequence and another special token [SEP] to separate sentences [20].

After selecting feature, we fine-tuned BERT again to get the performance after feature selection and compared the result with that before the feature selection.

4. Experiments

The workflow of our experiments is shown as Figure 3. We first calculated the propensity scores for one dataset and used the three proposed methods to find the spurious words, and experiment with the dataset after removing the spurious words; in addition, we remove the found spurious words from the other dataset and evaluate the performance of document classifier.

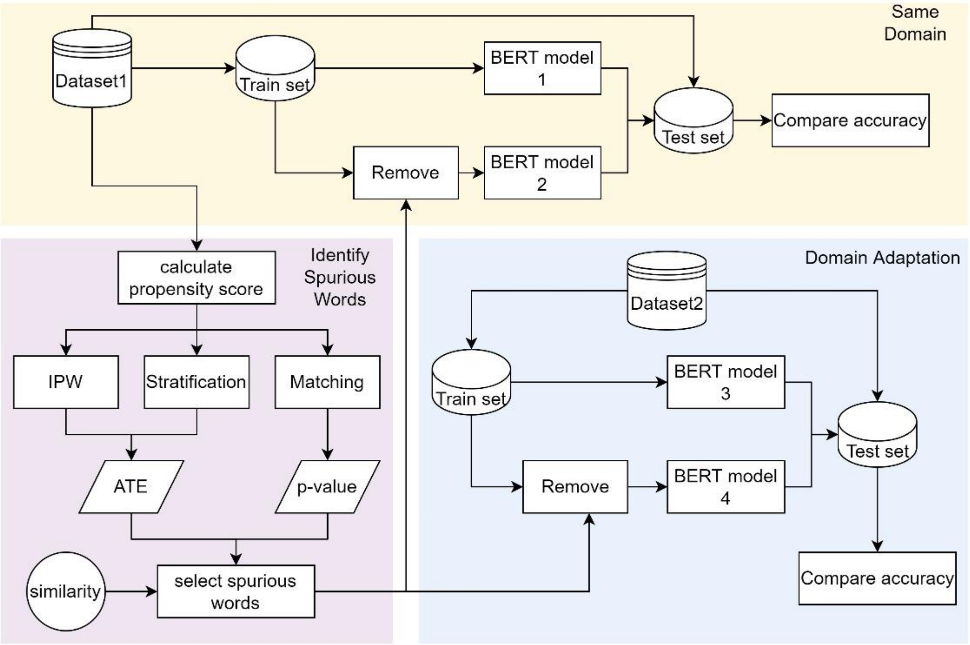

Figure 3. Architecture of Experiments.

4.1. Data

We used datasets of reviews and comments from two domains:

Movies We scraped 16,098 movie reviews from IMDB [20] of different kinds of movies. The score range of the review is from 0 to 10, and reviews rated ≥ 7 are labeled positive and reviews ≤ 4 are labeled negative [6].

Airline comments US Airline comments from Twitter, which is available on Kaggle. There are three kinds of labels (positive, negative and neutral), we only keep the positive and negative comments for our classification task. In the end we got 11,541 comments.

Amazon review Product reviews from Amazon, spanning May 1996 - July 2014 [2]. The score range of the review is from 0 to 5, and reviews rated > 3 are labeled positive and reviews < 3 are labeled negative.

4.2. *Calculate Propensity Scores*

As we use the kernel function to calculate propensity scores, The selection of the kernel function is crucial. We tried a variety of kernel functions: linear kernel, polynomial kernel, gaussian kernel, laplacian kernel, exponential kernel, sigmoid kernel, and many of them output similar distribution of probabilities. However, the distributions of positive class and negative class align better using the gaussian kernel and sigmoid kernel, so we focus on these two kernel methods in most cases. As for random forest, we simply call APIs to help the implementation, with the purpose that we are only interested in the probabilities of its output rather than itself. Different from the random forest, we implemented a toy neural network from scratch, hoping not to slow down the process of propensity score calculation, given that many modern neural networks cost much.

Besides that, we found that most of our data are unbalanced. For example, some words appear only in very few documents, or some words appear in most documents. We first preprocess the documents to remove stop words from the documents. At the same time, for those words that only appear in a few documents, we tried the weighted model with the 'balanced' field. We found that whether weighted or not has little effect on the final result, so we no longer focus on this.

4.3. *Kullback-Leibler Divergence*

Our goal is to make the output result centered around 0.5 because we can match better with compact propensity scores. We use KL divergence to create more overlaps. KL divergence is used to represent information gain or relative entropy, which measures the difference between two distributions [21]. We modify the sigmoid function in logistic regression and add a penalty term β*KL. We use the 0/1 distribution as the target distribution. First, we calculate the KL divergence of the probability distribution of the model output and the target distribution and then add this value to the sigmoid function described above.

$$\begin{aligned} D_{KL}(p|q) &= H(p,q) - H(p) \\ &= -\sum_x p\,(x)\log q(x) - \sum_x -p(x)\log p(x) \\ &= -\sum_x p\,(x)\,(\log q(x) - \log p(x)) \\ &= -\sum_x p\,(x)\log\frac{q(x)}{p(x)} \end{aligned} \tag{5}$$

Then, we can modify the sigmoid function $g(x)$ in logistic regression as below:

$$g(x) = g(x) + \beta \times D_{LK}(p||q) \times (q - g(x)) \tag{6}$$

These are the histograms of the experimental results. It can be seen in Figure 4 that the use of traditional logistic regression makes the propensity score polarized, which increases the difficulty of matching, and also makes it impossible to find two texts with the same score but different labels; Figure 5 is the result after using Gaussian kernel optimization. It can be seen that there are significantly more overlapping parts after optimization, but the distribution of propensity scores is uneven; Figure 6 is the logistic regression optimized using KL divergence. It can be seen that the distribution of propensity scores is symmetrical along 0.5, which meets our requirements for propensity scores.

4.4. *Clip*

When implementing inverse probability weighting, we find that propensity scores near 0 will cause the ATE to be too large which overshadows the efficiency of the method. One of the most simplistic approaches to solve this problem is to clip the propensity scores to an interval we give manually. For example, we can give an interval [0.1, 0.9] so that the output propensity scores will lie in the interval we give. We implement this approach by modifying the sigmoid function in logistic regression.

Table 1. Accuracy achieved namely by using nothing, IPW with logistic regression, IPW with gaussian kernel, IPW with neural networks, PSM with logistic regression, PSM with gaussian kernel, PSM with neural networks. The leftmost column represents the testing data. The uppermost row represents the training data where spurious words are ruled out with the proposed methods.

	Movie SW	Movie LR-IPW	Movie RBF-IPW	Movie NN-IPW	Movie LR-PSM	Movie RBF-PSM	Movie NN-PSM
Movie	91.0	**91.5**	91.0	91.0	91.1	90.9	91.1
Tweet	68.3	73.7	78.0	82.0	84.1	82.8	**84.6**
Amazon	80.9	**81.9**	79.8	80.5	80.2	79.5	80.6
	Tweet SW	Tweet LR-IPW	Tweet RBF-IPW	Tweet NN-IPW	Tweet LR-PSM	Tweet RBF-PSM	Tweet NN-PSM
Movie	62.8	**71.0**	54.7	58.0	53.5	57.5	53.5
Tweet	93.8	**94.1**	93.3	93.6	93.1	93.2	93.0
Amazon	43.7	47.6	37.3	37.1	35.2	36.9	35.2
	Amazon SW	Amazon LR-IPW	Amazon RBF-IPW	Amazon NN-IPW	Amazon LR-PSM	Amazon RBF-PSM	Amazon NN-PSM
Movie	84.3	84.5	**85.7**	85.5	84.8	85.0	85.0
Tweet	67.8	**76.7**	70.7	66.8	65.2	69.7	67.0
Amazon	90.8	**91.1**	89.8	90.1	89.9	90.0	89.8

4.5. IPW

We used this method to assign weights to each documents using the equation (4) (equation for weights). Then the average treatment effect was calculated as (4). Afterwards, ATEs calculated were used to partition important words into spurious and genuine group, where we experimented with different thresholds (95th, 75th, 50th, 25th, 5th percentile lines).

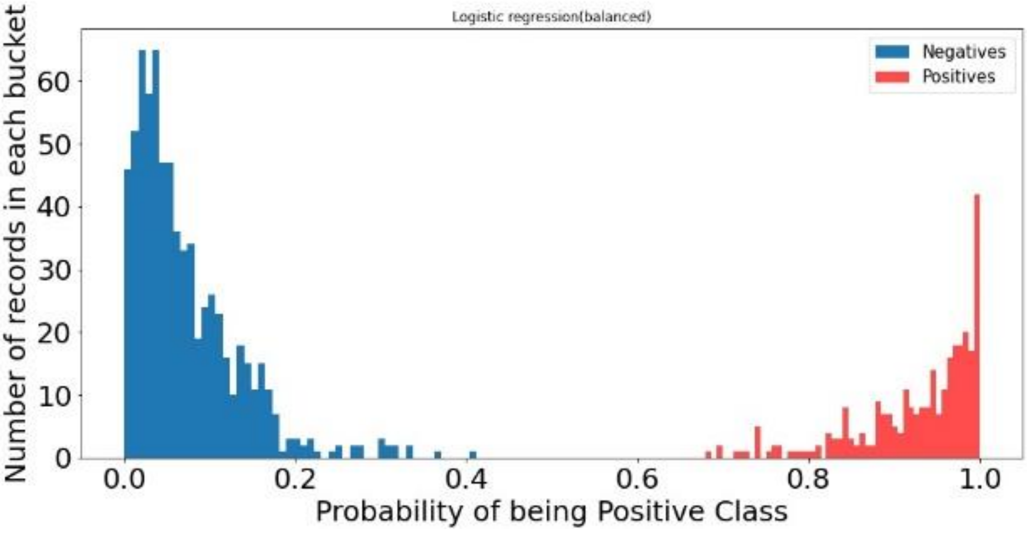

Figure 4. Logistic Regression.

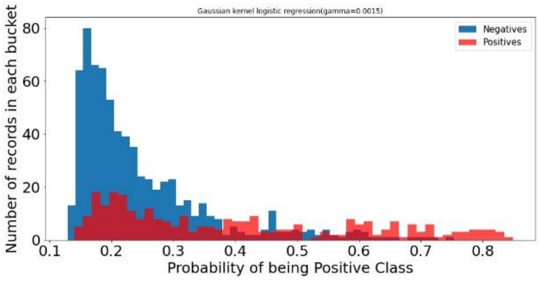

Figure 5. Gaussian Kernel Logistic Regression.

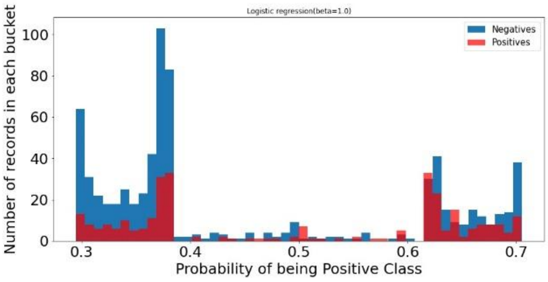

Figure 6. KL Divergence Logistic Regression.

5. Results & Conclusions

5.1. Results of Sentiment Analysis

The results we get from are shown in Table 1. The rows represent the domain of the testing data. The columns represent the training data where spurious words are ruled out. The column label Movie SW acts as a baseline, meaning that we do nothing with the original training set, and so do Tweet SW and Amazon SW. LR-IPW means that we use propensity scores estimated by logistic regression to implement the inverse propensity score weighting method. The rest column labels can be done within the same manner. According to the result, we can see that causal inference methods can indeed somehow improve the performance of text classifiers, with different methods of estimating propensity scores have different impacts on the final outputs. The key result is that no method have a negative impact on the model, demonstrating the correctness of out proposed approaches, with some methods performing worse in one setting but perform better in another setting. The key result is that IPW method works better when using the logistic regression as the propensity score calculation method, and PSM method performs better result with gaussian kernel. Table2 and Table 3 are mainly about the MCC and AUC when proposed methods are used to detect spurious words for the three datasets. The tables show below the MCC results and the AUC results.

Table 2. MCC results.

	Movie SW	Movie LR-IPW	Movie RBF-IPW	Movie NN-IPW	Movie LR-PSM	Movie RBF-PSM	Movie NN-PSM
Movie	78.4	**79.2**	78.4	78.4	78.5	78.2	78.6
Tweet	47.4	48.1	57.0	61.6	62.5	61.8	**64.1**
Amazon	48.7	43.6	49.0	**49.6**	48.2	48.6	48.6
	Tweet SW	Tweet LR-IPW	Tweet RBF-IPW	Tweet NN-IPW	Tweet LR-PSM	Tweet RBF-PSM	Tweet NN-PSM

Table 2. (continued).

Movie	42.9	**50.4**	36.0	38.2	35.2	38.4	35.2
Tweet	81.1	**81.7**	79.3	80.3	78.2	78.6	78.2
Amazon	26.1	**29.0**	21.3	21.1	20.0	21.6	20.0
	Amazon SW	Amazon LR-IPW	Amazon RBF-IPW	Amazon NN-IPW	Amazon LR-PSM	Amazon RBF-PSM	Amazon NN-PSM
Movie	60.6	61.2	**64.4**	64.0	62.0	62.6	62.6
Tweet	46.4	**55.3**	49.5	45.7	44.4	48.5	46.0
Amazon	69.5	**70.7**	67.0	67.5	67.3	67.5	67.3

Table 3. AUC results.

	Movie SW	Movie LR-IPW	Movie RBF-IPW	Movie NN-IPW	Movie LR-PSM	Movie RBF-PSM	Movie NN-PSM
Movie	88.9	87.4	88.8	88.8	88.9	**89.0**	**89.0**
Tweet	79.3	79.4	84.6	86.6	86.1	86.4	**87.1**
Amazon	77.3	72.0	**78.3**	78.2	77.3	78.1	77.4
	Tweet SW	Tweet LR-IPW	Tweet RBF-IPW	Tweet NN-IPW	Tweet LR-PSM	Tweet RBF-PSM	Tweet NN-PSM
Movie	72.4	**77.4**	67.2	60.1	66.5	69.0	66.5
Tweet	90.7	90.5	89.1	89.8	87.6	87.8	87.6
Amazon	64.1	66.5	60.3	60.1	59.2	60.3	59.2
	Amazon SW	Amazon LR-IPW	Amazon RBF-IPW	Amazon NN-IPW	Amazon LR-PSM	Amazon RBF-PSM	Amazon NN-PSM
Movie	76.2	77.3	**79.7**	79.3	77.8	78.5	78.5
Tweet	78.7	**83.8**	79.7	78.3	77.4	80.0	78.4
Amazon	82.2	**83.5**	82.2	81.9	82.5	82.4	82.8

5.2. Conclusions

Kernel methods were adopted to facilitate the calculation of propensity scores, which outstripped the traditional logistic regression in experiments because they can transfer the linearly inseparable problem to a linearly separable problem by mapping the original data set into a high-dimensional feature space. As for implementation, Kullback-Leibler divergence was introduced to optimize the distribution balance of propensity scores. Two causal inference methods were proposed in this paper: PSM and IPW, to anatomize the causal effect on document classes in sentiment analysis. While the two methods can theoretically lead to features with spurious correlations, the IPW can significantly help improve the performance of the document classifier, which was shown apparently in experiments. On the other hand, the experiments evidenced that feature selection resulting from the proposed methods can also perform well on out-of-domain data, such that spurious words found within the Movie data set may also be considered spurious in the Tweet data set. In future work, more approaches to generating propensity scores and causal inference will be experimented to improve the efficiency and performance of document classifiers.

References

[1] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques." arXiv preprint cs/0205070 (2002).

[2] Bing, L. "Sentiment Analysis and Opinion Mining (Synthesis Lectures on Human Language Technologies)." University of Illinois: Chicago, IL, USA (2012).

[3] Carosia, A. E. O., Guilherme Palermo Coelho, and A. E. A. Silva. "Analyzing the Brazilian financial market through Portuguese sentiment analysis in social media." Applied Artificial Intelligence 34.1 (2020): 1-19.

[4] Wang, Zhao, and Aron Culotta. "Identifying spurious correlations for robust text classification." arXiv preprint arXiv:2010.02458 (2020).

[5] Feder, Amir, et al. "Causal inference in natural language processing: Estimation, prediction, interpretation and beyond." arXiv preprint arXiv:2109.00725 (2021).

[6] Michael J. Paul. 2017. "Feature Selection as Causal Inference: Experiments with Text Classification." In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 163–172, Vancouver, Canada. Association for Computational Linguistics.

[7] Dehejia, Rajeev H., and Sadek Wahba. "Propensity score-matching methods for nonexperimental causal studies." Review of Economics and statistics 84.1 (2002): 151-161.

[8] Gordon J. G. Asmundson. Comprehensive clinical psychology 2nd Edition: Vol. 3: Research Methods in Clinical Psychology. Elsevier Science Ltd, 2022.

[9] Austin, Peter C. "An introduction to propensity score methods for reducing the effects of confounding in observational studies." Multivariate behavioral research 46.3 (2011): 399-424.

[10] P.R. Rosenbaum and D.B. Rubin. "The central role of the propensity score in observational studies for causal effects." Biometrika 70(1983):41-55.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: pre-training of deep bidirectional transformers for language understanding." CoRR, abs/1810.04805.

[12] Katherine A. Keith, David Jensen, and Brendan O’Connor. 2020. "Text and causal inference: a review of using text to remove confounding from causal estimates." Computation and Language, arXiv:2005.00649.

[13] Sun, Chi, et al. "How to fine-tune bert for text classification?." China national conference on Chinese computational linguistics. Springer, Cham, 2019.

[14] Caliendo, Marco, and Sabine Kopeinig. "Some practical guidance for the implementation of propensity score matching." Journal of economic surveys 22.1 (2008): 31-72.

[15] Elisseeff, André, and Jason Weston. "A kernel method for multi-labelled classification." Advances in neural information processing systems 14 (2001).

[16] Schölkopf, Bernhard, Ralf Herbrich, and Alex J. Smola. "A generalized representer theorem." International conference on computational learning theory. Springer, Berlin, Heidelberg, 2001.

[17] Eduardo Hariton and Joseph J. Locascio. "Randomised controlled trials - the gold standard for effectiveness research." BJOG 125(13):1716, 2018.

[18] Deborah Lai, Daniel Wang, Matthew McGillivray, Shadi Baajour, Ali S Raja, and Shuhan He. "Assessing the quality of randomization methods in randomized control trials." Healthcare Volume 9 Issue 4, 2021.

[19] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 188–197, Hong Kong, China. Association for Computational Linguistics.

[20] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts. 2011. "Learning word vectors for sentiment analysis." In Annual Meeting of the Association for Computational Linguistics (ACL).

[21] Goldberger, Jacob, Shiri Gordon, and Hayit Greenspan. "An Efficient Image Similarity Measure Based on Approximations of KL- Divergence Between Two Gaussian Mixtures." ICCV. Vol. 3. 2003.