Predicting Bias in Machine Learned Classifiers Using Clustering

Robert Thomson[1], Elie Alhajjar[1], Joshua Irwin[2], and Travis Russell[1]

[1] United States Military Academy, West Point NY 10996, USA
{Robert.Thomson,Elie.Alhajjar,Travis.Russell}@usma.edu
[2] Pennsylvania State University, State College PA, 16801, USA
jpi5033@psu.edu

Abstract. We investigate the problem of diagnosing bias in machine learned classifiers by examining performance on the clusters of a testing set. We propose an algorithm for predicting and mitigating the relative bias in the classifier. We examine the performance of this algorithm in a few well-studied datasets and discuss potential applications.

Keywords: Machine Learning · Bias · Clustering.

1 Introduction

As machine learning becomes a common component of technology used by experts and non-experts alike, examples of its shortcomings have become more and more visible to the general public. In particular, a number of high profile cases of machine learned classifiers making embarrassing predictions along culturally sensitive lines, like race and gender, have found their way into national headlines [2]. These examples have contributed to an ongoing debate about diversity in the technology sector and threaten to undermine public opinion on the use of data in industry.

In academic circles, investigations into bias in machine learning have largely focused on providing statistical evidence that a machine learned classifier is biased against data points with some previously identified attribute (for example, gender or race). Mitigation is achieved by adding new data points with the biased attribute and repeating the training until the previously identified bias is mitigated [5]. This method operates on the assumption that one has already identified a particular attribute as the cause of the bias, but it does not explain how one would identify such an attribute in the first place. This method also seems to ignore the possibility that other unknown attributes could be contributing to data-level bias or that adding new data points with some chosen attribute could inadvertently introduce new sources of bias.

In this report, we propose a simple algorithm for diagnosing and mitigating bias in a dataset. In short, the algorithm consists of clustering a dataset, training a classifier on a subset of the dataset, and then testing its performance on the clusters of the remaining data points. If the classifier performs poorly on one cluster relative to the other clusters, we identify that cluster as the location of the bias. We then mitigate by adding additional data points to the training set which are members of the biased cluster. This process is 'blind' in the sense that the user need not know which data points are subject to bias a priori or even be able to identify any explainable attribute as the cause of the bias.

Our method is based on the assumption that points which are similar (hence, members of some common cluster) are more likely to be grouped into the same class by a machine learned classifier. We rely on random clustering methods so that data points are not simply clustered by some common attribute but by their overall location in a feature space. This allows us to potentially identify and mitigate non-explainable sources of bias.

To illustrate this idea, consider the images in Fig. 1 below. These images were randomly selected from a common cluster of the MNIST dataset of hand written digits (see section 4 for more details). It is not clear by inspection whether the second image should be classified as a 3 or a 5 since it is similar to the other two images. We found that a machine learned classifier performed poorly on this cluster.

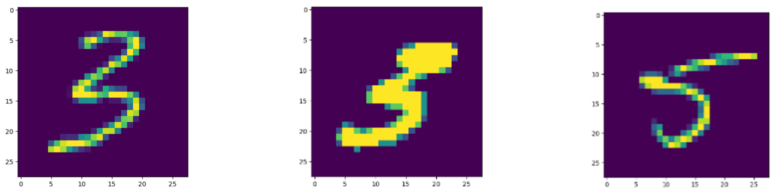

Fig. 1. Random images selected from a biased cluster in the MNIST dataset.

We test our algorithm by applying it to several well-known datasets. We use only built-in machine learning and clustering packages in Keras and Matlab. Interestingly, our methods allow us to identify explainable attributes associated to biased clusters in some of these well-known datasets. We hope this work serves as a proof-of-concept from which more rigorous research and discussion will follow.

Our work is inspired by methods in the literature on the class imbalance problem. Class imbalance occurs when one is asked to classify data points into one of several classes, at least one of which has relatively few instances (e.g. detection of fraudulent credit card transactions). Training a classifier on a highly imbalanced dataset frequently leads to bias against data points with minority class values. Researchers have addressed this issue by using clustering to pinpoint the location of minority class data points in a feature space and then mitigate bias by oversampling within the identified clusters. See [1] for more details on these methods.

Our paper is organized as follows. In section 2, we review basic concepts from machine learning and clustering. In section 3, we present the details of our algorithm for diagnosing and mitigating bias. In section 4, we discuss empirical results obtained by applying our algorithm to several well-known datasets. In section 5, we identify some questions raised by our investigation and discuss future work.

2 Background

In this section we review the basic notation and terminology from machine learning and clustering. For more details, we refer the reader to [7]. Readers familiar with the fundamental concepts of machine learning and clustering may feel comfortable continuing to section 3.

2.1 Supervised Machine Learning

Suppose we are given a dataset X consisting of pairs $\{x, r\}$ where x is a vector of attributes chosen from a set of attribute vectors A and r is a label chosen from a list of possible class labels $C = \{r_1, r_2, \ldots\}$. The goal of supervised machine learning is to train a classification function $f: A \rightarrow C$which takes as input an attribute vector x and returns as output an appropriate label $f(x)$. Because the entire universe of correctly labeled data points is not accessible, the function f must be trained on some small subset $T \subset X$called the training set.

There are many different algorithms for training classification functions. In some cases, the mechanics of the classification function are easy to describe and thus the relation between the output label $f(x)$ and the input vector x can be explained in human language. In other cases, it is not clear why an output label $f(x)$ is assigned to an input vector x, so the classification function can be regarded as a black box which assigns class labels to attribute vectors without any known explanation. In our work, we will use only built-in packages available in Matlab and Keras since our analysis is focused on data-level procedures rather than algorithm-level procedures. Hence, we will regard machine learned classifiers as black boxes.

2.2 Clustering

Many datasets which occur in data analysis do not come equipped with well-defined labels. In this case, one can still produce a classification function using a set of techniques collectively known as clustering. In short, clustering takes an unlabeled dataset consisting of attribute vectors and assigns each attribute vector to a cluster.

The basic principle behind clustering is that similar attribute vectors should be grouped in the same cluster. Consequently, many clustering algorithms require the user to specify a metric $d(x, y)$ which defines the distance between two attribute vectors x and y.

For the datasets under consideration in this paper, we rely on k-means clustering. The k-means algorithm takes a dataset and a positive integer k as input. From the dataset, we select k randomly sampled data points $x_1, x_2, \ldots, x_k$, called centroids. A classification function is then produced by assigning each attribute vector x to a cluster j from the list $[k] = \{1, 2, \ldots, k\}$ by determining which of the data points $x_1, x_2, \ldots, x_k$ minimizes the value $d(x, x_j)$.

3 Methodology

In this section we describe the proposed algorithm which is the main topic of our paper. The algorithm consists of roughly two procedures: diagnosis and mitigation.

3.1 Diagnosis of Bias

We begin by describing a procedure for diagnosing bias in a dataset. We broadly divide our algorithm into three steps: clustering, training, and testing.

Clustering. The process begins by partitioning a dataset into clusters. As described in subsection 2.2, we rely primarily on the k-means clustering algorithm. The parameter k is dependent upon the dataset and is chosen heuristically. To apply k-means, we define the distance $d(x,y)$ between attribute vectors x and y based on the type of attributes present. When the attributes consist of only numeric values, we define the distance $d(x,y)$ to be Euclidean distance. When the attribute vectors consist of categorical attributes, we encode the categories as binary strings and define the distance between two classes to be the Hamming distance [4]. The cluster values for the data points are set aside and are not used again until we reach the testing phase of the algorithm. Finally, the dataset is arbitrarily partitioned into training, testing, and pooling sets. The pooling set will be used in the mitigation algorithm, but not in the diagnosis algorithm.

Training. The next step in the algorithm is to train a classification function on the training set following standard machine learning procedures. Note that the training procedure is independent of the clustering procedure - i.e., the clustering does not affect the outcome of the training procedure in any way.

Testing. The final step in the diagnosis algorithm is testing. This consists of comparing the output of the classifier on the testing set to the actual labeled values of data points in the testing set. Results are sorted by cluster, so that the classifier's overall performance can be compared with its performance on the individual clusters. If the classifier's performance is poor on cluster j, then we say the classifier is biased against cluster j.

3.2 Mitigation of Bias

We consider two methods of mitigating bias: the first method we call 'pooling and replacement' and the second we call 'pooling and supplement'.

In both cases, the user first identifies a cluster j against which the given classifier is biased. The training set is then augmented by adding additional data points from the pooling set with the condition that the additional data points are members of the cluster j. A new classifier is then trained on the augmented training set. Finally, the new classifier is applied to the testing set using one of the following two options.

Pooling and Replacement. With this option, the new classifier obtained by training on the augmented training set is applied to the entire testing set. Hence, the old classifier is replaced by the new classifier. One potential disadvantage of this method is that the performance of the classifier on certain clusters may diminish. This possibility is addressed by the following option.

Pooling and Supplement. With this option, the new classifier is applied only to data points in the cluster j, while the previous classifier is applied to the other points in the testing set. In other words, if the original classifier is given by the function $f(x)$ and the new classifier is given by the function $g(x)$, the final classifier is given by

$$h(x) = \begin{cases} g(x), x\ is\ in\ cluster\ j \\ f(x), otherwise \end{cases} .$$

This method has the advantage that the performance of the classifier is guaranteed to be consistent on all clusters besides the cluster j.

It is important to note that the pooling and supplement option can only be applied to arbitrary data points if there is some method for determining which cluster a new data point belongs to since this method treats data points in cluster j and data points in other clusters differently. If clustering is performed using k-means, this is not a problem, since a new data point can be assigned to a cluster by minimizing the value $d(x, x_j)$ over the set of centroids $\{x_1, x_2, \ldots, x_k\}$.

4 Empirical Results

In this section we summarize the results of our experiments. We begin with a description of the datasets we studied and then discuss the results of our analysis. Three of the datasets were collected from the University of California at Irvine Machine Learning Repository and are well-known [3]. For classification on these datasets, we used a decision tree ensemble package in Matlab. The fourth dataset consists of images of hand-written digits and is also well-known [6]. For classification on this dataset, we used a convolutional neural network algorithm in Keras. We provide brief descriptions of these datasets for the convenience of the reader.

1984 Congressional Voting Records Dataset. This dataset is a record of voting outcomes for all members of the U.S. House of Representatives from the year 1984. The attributes are voting outcomes ('y', 'n', or '?') on bills. An attribute '?' indicates that the lawmaker was not present or abstained from voting. The class labels are the political parties of the lawmakers (Democrat or Republican).

Pima Indians Diabetes Dataset. This dataset is a medical survey of Pima Indians. The attributes are numerical values representing medical observations like age, weight and blood pressure. The class labels are 0 or 1, with 0 indicating that the patient has not been diagnosed with diabetes and 1 indicating that the patient has been diagnosed.

Adult Census Dataset. This dataset consists of demographic information about individuals listed in a 1994 U.S. census database. Some attributes are numerical (e.g. age, number of hours worked per week) while most are categorical, such as demographic information. For clustering, we ignored numerical attributes and based clusters only on categorical attributes.

MNIST. This dataset consists of 28 by 28 pixel images of hand-written Arabic numerals. The goal of classification is to determine which of the numerals 0, 1, 2, … 9 corresponds to the image. For clustering, we applied principal component analysis to reduce the dimension of the images, then clustered the reduced images into 20 clusters with k-means using Euclidean distance.

Table 1. A summary of our experimental results on four datasets: 1984 Congressional Voting Records, Pima Indians Diabetes, Adult Census, and MNIST Datasets (respectively).

Dataset	Congress	Pima	Census	MNIST
Accuracy	91.5%	77.0%	83.9%	96.5%
Accuracy on biased cluster	68.2%	57.1%	53.2%	92.8%
Accuracy on cluster after mitigation	86.4%	71.4%	60.5%	95.0%
Accuracy after replacement	95.0%	77.3%	84.5%	96.9%
Accuracy after supplement	93.5%	78.1%	84.1%	96.8%

Results. The results of our experiments are summarized in Table 1 above. All values are percent accuracy, which we define to be the number of correctly classified data points in the testing set (resp. cluster) divided by the size of the dataset (resp. cluster).

Discussion. We conclude this section by discussing a few observations from our experiments. Even though the clustering procedures we applied were random, a review of the clustered data points in the testing set revealed that data points in the biased datasets had some explainable characteristics of interest. In particular, we found that data points in the biased cluster for the Pima Indians dataset had an average recorded glucose level of 165 in contrast with the overall average glucose level of 121 in the testing set. For the 1990 Census Dataset, we discovered that 83.7% of the data points in the biased cluster identified as black males. These observations provide some empirical evidence that our blind method is in fact capable of discovering the kinds of bias which have found their way into news headlines.

5 Conclusion and Future Work

In summary, we have demonstrated a simple algorithm for diagnosing and mitigating bias. One advantage of our method is that the user need not identify the source of bias since the source of the bias is identified by a random clustering algorithm. Nonetheless, we have demonstrated that explainable sources of bias can be detected by our methods.

We conclude by summarizing future work we hope to pursue. We employed the k-means algorithm with either Euclidean or Hamming distance as the tool for clustering. We did not explore the effectiveness of other clustering algorithms or distance metrics. Furthermore, our experimentation relied on a single random clustering and a single 'black-box' machine learning algorithm. It would be interesting to investigate the effectiveness of our methods with several different machine learning algorithms. It would also be interesting to investigate whether a more robust algorithm could be developed by combining the results of several random clusterings rather than relying on a single clustering. Finally, in the datasets which we explored, parameters such as the size of the testing, training, and pooling sets as well as the number of clusters were chosen heuristically based on experimental results. A more rigorous method of parameter searching could potentially increase the effectiveness of our algorithm.

References

1. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: Synthetic Minority Oversampling Technique. Journal of Artificial Intelligence Research, vol. 16, pp. 321-357 (2002).
2. Devlin, H.: Discrimination by Algorithm: Scientists Devise Test to Detect AI Bias. The Guardian (online), Dec. 18, 2016.
3. Dua, D., Karra-Taniskidou, E.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, last accessed 2018/04/19.
4. Hamming, R.W.: Error Detecting and Error Correcting Codes. The Bell System Technical Journal, 29(2), 147–160 (1950).
5. Hardt, M., Prince, E., Srebro, N.: Equality of Opportunity in Supervised Learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems 2016, pp. 3323-3331. Curran Associates Inc., Barcelona, Spain (2016).
6. LeCun, Y., Cortes, C., Burges, C.: The MNIST Dataset of Handwritten Digits, http://yann.lecun.com/exdb/mnist/, last accessed 2018/04/26.
7. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning From Theory to Algorithms. 1st edn. Cambridge University Press, New York (2014).

Disclaimer

This research was supported in part by the DARPA Explainable Artificial Intelligence (XAI) program. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of the Army, Department of Defense, or the U.S. Government. The authors would also like to thank Dr. Frank Ritter for his comments.