Concept-Based Explanations on Tabular Learning

Jihye Choi
jihye@cs.wisc.edu

Varsha Pendyala
varsha.pendyala@wisc.edu

Abstract

The interpretability of machine learning models has been an essential area of research for the safe deployment of machine learning systems. One particular approach is to attribute model decisions to high-level concepts that humans can understand. However, such concept-based explainability for Deep Neural Networks (DNNs) has been studied mostly on image domain. In this paper, we extend TCAV, the concept attribution approach, to tabular learning, by providing an idea on how to define concepts over tabular data. On a synthetic dataset with ground-truth concept explanations and a real-world dataset, we show the validity of our method in generating interpretability results that match the human-level intuitions. On top of this, we propose a notion of fairness based on TCAV that quantifies what layer of DNN has learned representations that lead to biased predictions of the model. Also, we empirically demonstrate the relation of TCAV-based fairness to a group fairness notion, Demographic Parity.

1 Introduction

Despite the successful generalizability of deep learning models in various application domains, limited understanding of their behavior prevents their deployment in full scale. The study of interpretability in machine learning gained greater attention recently, to explore what factors influence a model's prediction. In the real world, most commonly available and highly valuable data is in the form of tables. While the deep learning models are highly successful in computer vision, speech recognition and natural language processing, there is relatively less usage of such models for learning involving tabular datasets. As a result, majority of the work on interpretability of neural networks has not been studied in the context of tabular datasets.

Several approaches were proposed to explain the model's prediction to an input in terms of its input features. In case of deep neural networks for image understanding, feature attributions for model prediction are typically obtained as a saliency map over pixels of the input image [14, 13, 15]. However, feature importance scores alone may not aid us in explaining the influence of high-level human friendly concepts on model predictions. For instance, how do we measure the influence of a pattern, object or the background in an input image on model's decision?

Recent work shows the existence of detectors for higher level concepts at higher layers and lower level concepts at lower layers [4]. This enables the network to converge to representations in a bottom-up manner while inducing linear separability of higher level features [12, 1]. Supported by these findings, authors in [11] proposed a concept attribution approach: *Testing with Concept Activation Vectors* (TCAVs).

Through this project we extended the TCAV based interpretability to deep learning over tabular datasets. In addition, we show how the TCAV can be viewed as a tool for estimating model fairness. We considered a state-of-the-art neural network that is specially designed for the tabular datasets: *TabNet* [3]. Our experimental results show that TCAV based interpretability has been effective in revealing the biases present in the data used to train the TabNet.

2 Related work

2.1 Interpretability method based on concept attribution

TCAV approach proposed in [11] generates insights by quantifying model's sensitivity with respect to user-defined concepts defined in the form of examples.

Concept Activation Vector Given a set of examples for every concept, the first step of TCAV is to find a vector for each concept in the space of activations at a layer l. These vectors are referred as Concept Activation Vectors (CAVs). Each CAV points in the direction of activations of examples of corresponding concept. CAV for a concept is normal to the hyperplane that separates positive and negative examples corresponding to that concept. For instance, to learn a CAV for checkerboard pattern, positive examples can be all those images with checkerboard pattern and negative examples could be any random images which do not have that pattern. The hyperplane that separates them is its CAV.

Conceptual sensitivity Given a test image, it is possible to quantify the influence of a user-defined concept on the neural network's decision. For a concept C with unit CAV vector v_C^l computed from the activations of layer l with m nodes, sensitivity of class k with respect to C is given by:

$$
\begin{aligned}
S_{C,k,l}(x) &= \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(x) + \epsilon v_C^l) - h_{l,k}(f_l(x))}{\epsilon} \\
&= \nabla h_{l,k}(f_l(x)) \cdot v_C^l
\end{aligned}
\tag{1}
$$

where $f_l(x)$ denotes the activation of input x computed at layer l and $h_{l,k} : R^m \to R$.

TCAV score for any target class k with respect to C is computed using all the inputs belonging to the class k as follows: when $|X_k|$ is the number of images from class k,

$$
TCAV_{C,k,l} = \frac{\sum_{x \in X_k} S_{C,k,l}(x)}{|X_k|}
$$

2.2 Deep Neural Networks for Tabular data

Unlike in the image domain, machine learning with canonical Deep Neural Networks (DNNs) remains under-explored, with variants of ensemble decision trees still dominating most applications. Tree-based methods have certain benefits indeed: they are representationally efficient, interpretable in their basic form, and fast to train. However, recent studies suggest that DNN-based architectures would lead to performance improvements, particularly for large datasets [9], and it is worth to explore deep learning for tabular data. Since canonical DNNs based on stacked convolutional layers or multi-layer perceptrons are likely to be overparametrized over tabular data [7], the need for a DNN architecture suitable for tabular learning emerges.

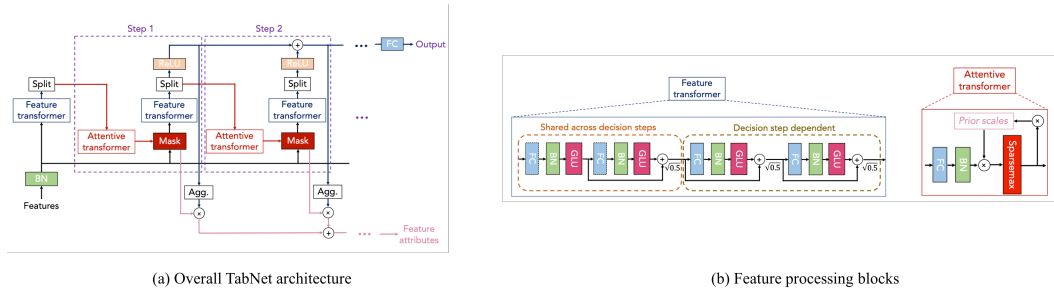(a) Overall TabNet architecture　　　　　　　　　(b) Feature processing blocks

Figure 1: TabNet architecture from [3]. (a) Using a sequential attention mechanism, TabNet chooses a subset of semantically meaningful features to process in each decision step. A complete feature vector X is passed to each decision step. (b) i^{th} step uses the processed information from $(i-1)^{th}$ step to process subset of features of X using feature transformers. Each step outputs the processed feature representation to be aggregated into the overall decision.

TabNet   Arik *et al.* [3] proposed a novel DNN architecture that uses sequential attention to choose which features to reason from at each decision step (Figure 1). TabNet outperforms other neural network and decision tree variants across tabular datasets from diverse domains. Throughout this paper, we adopt the TabNet architecture for classification tasks.

3   TCAV based interpretability for tabular data

3.1   User-defined concepts for tabular data

We apply the TCAV approach to explain the influence of user-defined concepts over classifications tasks defined on tabular datasets. Given that the tabular data is structured, we define concepts over columns of table data in the form of boolean predicates. This automates the process of providing positive and negative examples to the binary classifier learning a CAV.

Example concepts for Adult Income Dataset [6] are shown in Figure 2.

```
def doctorate(x):
        if x['education']=='Doctorate':
                return True
        return False

def male(x):
        if x['gender']=='Male':
                return True
        return False

def female(x):
        if x['gender']=='Female':
                return True
        return False
```

Figure 2: Predicates over Adult Income Dataset

3.2   Training CAVs with tabular data

We train *CTGAN*, a generative model described in [16] to generate satisfying and unsatisfying assignments to our boolean predicates for all user-defined concepts. Through our experiments in section 5.1 we show that CTGAN has been very effective in computing reliable CAVs.

Now we describe how using a CTGAN to generate data for concept learning in case of non-pictorial data is equivalent to manually providing positive, negative example images for concept learning in case of image domain.

Concepts in image domain   Let $stripes$ denote the user-defined concept. All the images with stripes can be thought of samples drawn from a distribution $Stripe_X$. Note that the positive images for $stripes$ concept are not sampled from the original data distribution D_{XY} where X is the random vector corresponding to all the pixels of an input image, Y is the class label.

$$\text{striped images} \sim Stripe_X$$

If we know the distribution $Stripe_X$, all the positive images for $stripes$ concept can be drawn from it and given to the linear classifier to obtain the CAV vector. However, since we do not know $Stripe_X$, we would create synthetic images with stripes to feed to the binary linear classifier. Since it is pictorial data, it is possible for us to assign intensity values to different pixels to compose realistic striped images that captures joint distribution $Stripe_X$ among its pixels.

Concepts in tabular domain   Let *age < 25* denote the user-defined concept. Denote the original data distribution as D_{XY}, where X denotes the random vector corresponding to all the columns of an input row, Y is the class label. Assume A_X to be the distribution from which positive examples for *age < 25* can be drawn and B_X from which negative examples can be drawn.

$$\text{input data with age} < 25 \sim A_X$$

$$\text{input data with age} \geq 25 \sim B_X$$

3

Since we do not know A_X, B_X, we can provide examples of row data with age < 25, age ≥ 25 to the binary classifier to learn CAV. Since it is non-pictorial data, it may not be possible for us to assign values to different column features to compose realistic age < 25 or age ≥ 25 data that captures respective join distributions A_X, B_X among its columns.

So, we can follow below procedures to generate examples that captures joint distributions A_X, B_X.

Procedure 1

- Use the available natural data (i.e train/test data) where age < 25 to train a $CTGAN_{age<25}$.
- Use the natural data where age ≥ 25 to train $CTGAN_{age\geq25}$.
  Note that sampling the data from $CTGAN_{age<25}$ or $CTGAN_{age\geq25}$ is equivalent to sampling from A_X or B_X respectively.
- Sample examples from $CTGAN_{age<25}$, $CTGAN_{age\geq25}$ to learn the binary classifier corresponding to CAV.

For a random vector r representing a table row, we have:
$$Pr_{A_X}(r) = Pr_{D_X}(r|r['age'] < 25)$$
$$Pr_{B_X}(r) = Pr_{D_X}(r|r['age'] \geq 25)$$
Using the above relations, we can follow the below simplified procedure to generate examples which capture A_X, B_X joint distributions:

Procedure 2

- Use the entire available natural data (i.e train/test data) to train a $CTGAN_X$.
  Note that sampling the data from $CTGAN_X$ is equivalent to sampling from D_X.
- Then sample the datapoints from $CTGAN_X$ to accumulate positive and negative examples for every concept we are interested in learning.

4 TCAV based fairness

This section describes our idea and method: how to use TCAV based interpretability for tabular data (Section 3) as a tool for diagnosing fairness level of a trained models.

Without loss of generality, we consider a classifier $f : \mathbb{R}^n \to \mathbb{R}^m$ with inputs $X \in \mathbb{R}^n$ and a output layer with m neurons. Let Y be a target label such that $Y = \{y_1, y_2, ..., y_m\}$ and $A = A_1, A_2, A_3, ...$ be a protected features among n features of X (e.g. A = 'female', 'male', for a protected feature 'gender'). We assume X, A, Y are generated from an underlying distribution D i.e. $(X, A, Y) \sim D$.

The first step in our method is to define a concept of interest C in terms of A. More specifically, a concept is described as a predicate that assigns true for samples having A_1 (that is, satisfying examples) and false for samples without A_1 (that is, unsatisfying examples). Then, by setting target class as $y_k \in Y$, we train two linear classifiers to obtain CAV vector $v_{C_1}^l$ and $v_{C_2}^l$ in Equ. 3, where C_1 is $X[A] == A1$ and C_2 is $X[A]! = A1$, respectively.

Given the above, we propose a definition of fairness that involves TCAV scores for C_1, C_2, y_k 1. The intuition behind the definition is that if a classifier's prediction is independent to the protected attribute, its conceptual sensitivity gauged over samples that possess that attribute or not, should be equal.

Definition 1 *A classifier f satisfies fairness in regards to concept attributions under a distribution D if its prediction $f(X)$ is equally sensitive to C_1 and C_2 - that is, if $|TCAV_{C_1,k,l} - TCAV_{C_2,k,l}| = 0$ for all $y_k \in Y$.*

Considering that TCAV score is basically a dot product between two vectors, $h_{l,k}(f_l(x))$ and v_C^l, we introduce to quantify the sensitivity of a prediction to a concept as an angle between the two vectors.

$$\angle TCAV_{C,k,l} = \frac{\sum_{x\in X_k} \angle (h_{l,k}(f_l(x)), v_C^l)}{|X_k|} \qquad (2)$$

where $\angle (h_{l,k}(f_l(x)), v_C^l) = \cos^{-1}(\frac{h_{l,k}(f_l(x)) \cdot v_C^l}{||h_{l,k}(f_l(x))||||v_C^l)||})$. Using the notion of TCAV angle, as the angle between $h_{l,k}(f_l(x))$ and v_C^l gets close to $90°$, the classifier becomes fairer. Our definition of fairness in 1 can be rewritten as follows.

Definition 2 *A classifier f satisfies fairness in regards to concept attributions under a distribution D if $\angle TCAV_{C,k,l} = 90°$ for all $y_k \in Y$ and $C = \{C_1, C_2\}$.*

The advantage of our definition over other fairness definitions such as Demographic Parity or Equalized Odds [8] is that it allows to tell what layers of neural networks have learned representations that contribute to biased predictions.

5 Experiments

Through our experiments we provide empirical evidence to the following: **(1)** TCAV scores of a target class with respective to various user-defined concepts can throw light on the inherent biases within the training data. **(2)** Generative models can be very effective in generating example data for training reliable CAVs. **(3)** TCAV can be viewed as a fairness estimation tool.

We perform experiments on classification tasks in synthetic settings and in published benchmark settings. We show evidence that TCAV results align with intended concepts of interest, with tabular datasets that are also studied in [3]. We use the same input data processing, and hyperparameters for training TabNets as in [3].

Based on the open-source implementations [10, 2] we develop codes for obtaining TCAV interpretability on tabular datasets in Tensorflow. For every user-defined concept, a CAV is trained given the activation layer outputs that correspond to the tensors resulting from the cumulative sum over ReLU outputs at each decision step in Figure 1(a).

5.1 TCAV based interpretability

Synthetic Dataset   We consider a synthetic tabular dataset for binary classification task from [5] consisting 1M training samples and a separate test set of 100k samples. The synthetic dataset is constructed in such a way that the ground truth output of the dataset Y only depends on the two dimensional XOR of first two features $X1$ and $X2$. Hence, only $X1$ and $X2$ serve as true features, where each dimension itself is independent of Y but the combination of them has a joint effect on Y. More specifically, for a sample to belong to the class of $Y = 0$, it is implied to have $X1X2 > 0$, whereas the output label Y is 1, if and only if the sample has the property of $X1X2 < 0$.

The concepts of our interest for each of which CAVs are trained are as follows:

- C1: $X1 > 0$ AND $X2 < 0$
- C2: $X1 < 0$ AND $X2 > 0$
- C3: $X5 > 0$
- C4: $X10 < 0$
- C5: $X1 > 0$ AND $X2 > 0$
- C6: $X1 < 0$ AND $X2 < 0$

When $Y = 1$, C1 and C2 are the positively relevant concepts, where C5 and C6 are negatively relevant concepts to the predictions. On the contrary, when $Y = 0$, having C1 and C2 affects the output negatively, but C5 and C6 become positively relevant concepts. For both of the cases, C3 and C4 are irrelevant concepts to determining the output label. We observe that the empirical results align with this intended design of concepts. In Figure 3(a), TCAV scores for C1 and C2 are negative values, whereas they are positively large in Figure 3(b). It means that having properties defined as C1 and C2 satisfied leads to a high chance for the sample to be predicted as class 1. This applies to C5 and C6 in the opposite manner and agrees with our intuition as well. Please note that the accuracies of linear classifier for C3 and C4 are no better than random guessing, and resulting CAVs are not meaningful enough, so ommitted after statistical testing as proposed in [11].

(a) TCAV for all layers in TabNet when target label is class 0 (i.e., Y = 0)

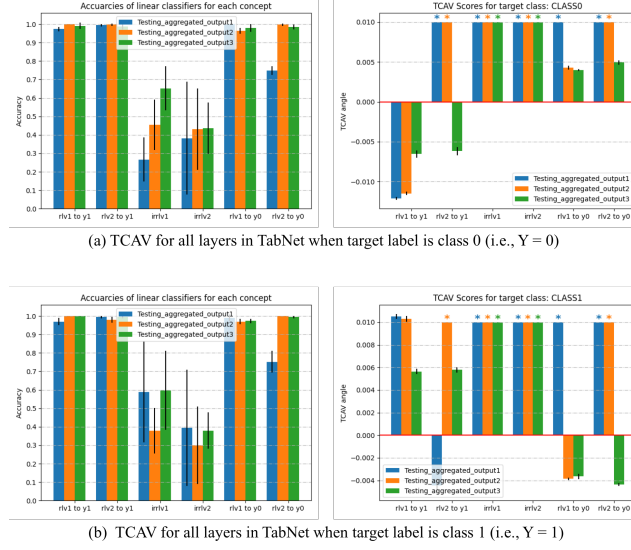(b) TCAV for all layers in TabNet when target label is class 1 (i.e., Y = 1)

Figure 3: The accuracies of CAVs and TCAV scores over synthetic dataset. On the X axis, 'rlv1 to y1', 'rlvl to y1', 'rlv1 to y0' and 'rlv1 to y0' correspond to C1, C2, C5 and C6, respectively. 'irrlv1' and 'irrlv2' represent C3 and C4. '*'s mark CAVs ommitted after statistical testing.

Table 1: Datasets used to obtain concept attribution. Dataset to train CAVs is a set of data from which the satisfying and unsatisfying assignments of given predicates are taken to train the CAVs. Dataset to compute TCAV score is a set of data from which the inputs corresponding to the given target class are used to compute TCAV score.

Dataset to train CAVs	Dataset to compute TCAV score
Training data of TabNet	Test data
Test data	Test data
Synthetic data	Test data
GAN data	Test data

Adult Income Dataset [6]   We consider Adult Income Dataset, which is the most commonly used benchmark dataset for binary classification. The task is to distinguish whether a persons income is above $50k from demographic and employment related features. CAVs are trained for following five concepts:

- C1: age < 25
- C2: native-country == Mexico
- C3: gender == Female
- C4: native-country == India
- C5: education == Doctorate.

In the training data, we observed following trend with respect to the above mentioned sub-groups:

$$Pr(Y = 0|C1) \geq Pr(Y = 0|C2) \geq Pr(Y = 0|C3) \geq Pr(Y = 0|C4) \geq Pr(Y = 0|C5) \quad (3)$$

where $Y = 0$ for an individual implies, their income is less than $50K$.

Table 1 lists four different ways in which we trained CAVs and computed TCAV scores.

We generated synthetic data mentioned in Table 1 by uniformly sampling each column of the table in it's respective domain: numerical features are sampled from the uniform distribution with min and max corresponding to the respective numerical column in training data, categorical data is uniformly sampled from all the unique values corresponding to the respective categorical column in training data.
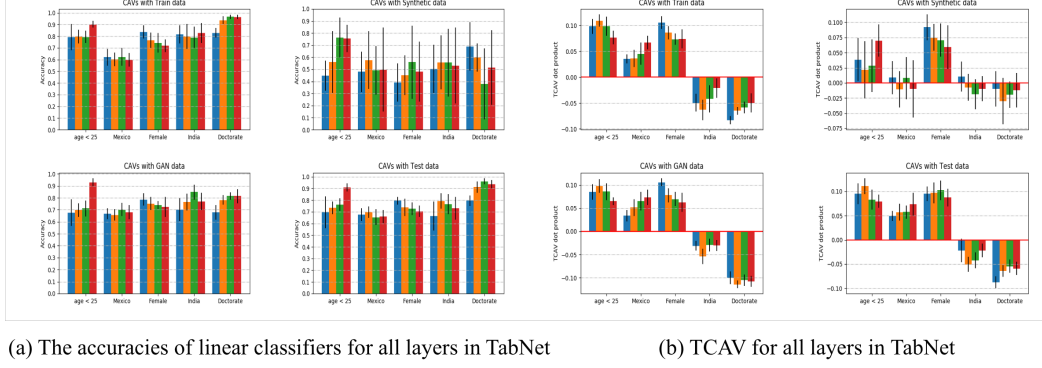
6

(a) The accuracies of linear classifiers for all layers in TabNet   (b) TCAV for all layers in TabNet

Figure 4: The accuracies of CAVs and TCAV scores over Adult Income Dataset

(a) TCAV for all layers in TabNet when target label is class 0 (i.e., Y = 0)

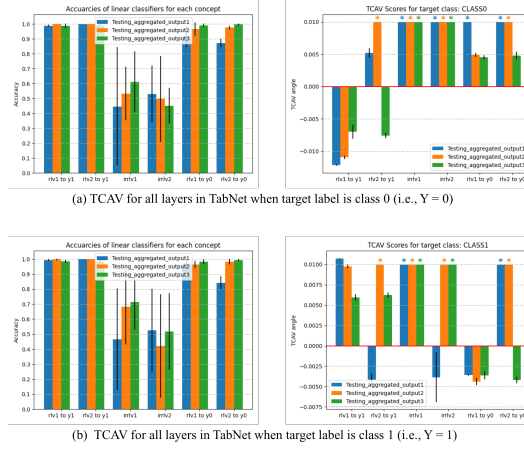(b) TCAV for all layers in TabNet when target label is class 1 (i.e., Y = 1)

Figure 5: The accuracies of CAVs and TCAV scores over synthetic dataset. The only difference with the settings of Figure 3 is that the values of features corresponding the concepts are replaced with values uniformly drawn within the range of the original values

From the results shown in Figure 4(b) we can see that the TCAV scores across user-defined concepts indeed reveal the biases within the training data by following the same trend across concepts as in 3. Different colors in figures 4(a), 4(b) correspond to different decision step activations: $blue$ to red for first layer to the last one.

We can see that the GAN based TCAV scores in Figure 4(b) with various user-defined concepts follow the same trend as those for which CAVs were trained with training or test data. This implies, when we do not have enough test data available for training CAVs, we can use GAN to generate satisfying and unsatisfying assignments for the boolean predicates.

Also we observed that the assignments coming from the synthetic data could not result in reliable CAVs. This can be noted from the very high standard deviation of the accuracy of linear classifiers at every decision step, when synthetic data is used to train the CAVs (shown in Figure 4(a)).

To further verify our argument on why the interpretability fails with the modified data, we perform the same experiment in the synthetic setting that is described early in this section. Unlike in the real-world data, the features in domain X are independent of each other, so that we expect to obtain reliable CAVs even with uniformly sampled data. Figure 5 shows almost identical interpretability results with 3.
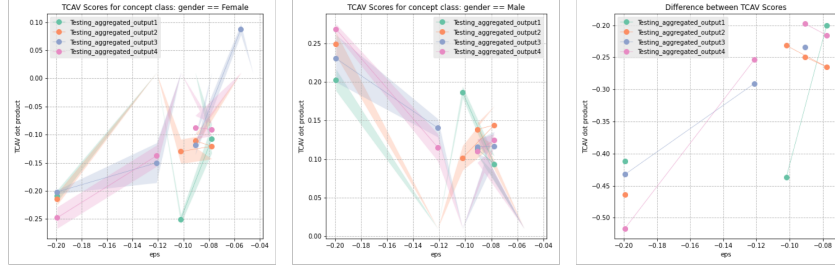
5.2 TCAV based fairness

To show the validity of our fairness definition 1, 2, we empirically demonstrate that the fairness level quantified by our definition aligns with that of other commonly used fairness metrics. Since TCAV
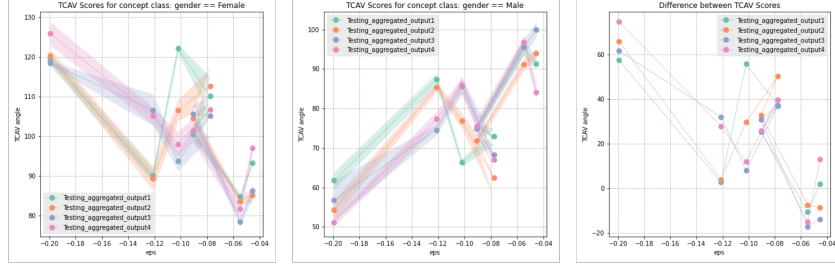
(a) TCAV angles for all layers in TabNet and the difference between TCAV angles between satisfying and unsatisfying groups

(b) TCAV scores for all layers in TabNet and the difference between TCAV scores between satisfying and unsatisfying groups

Figure 6: TCAV resuls given a binarized protected feature 'gender'

provides global interpretability that is measured not on a single samples, but all samples in a class, it is natural to consider the TCAV-based fairness notions as a global fairness. In this section, we investigate the correspondence between TCAV-based fairness and a famous group fairness metric, Demographic Parity.

Demographic Parity can be thought of as a version of the US Equal Employment Opportunity Commission's "four-fifths rule" that requires "selection rate for any race, sex, or ethnic group [must be at least] four-fifths (4/5) (or eighty percent) of the rate for the group with the highest rate." In our experiments, we use Adult Income Dataset where $Y = \{'>50K', '<=50K\}$ and a binarized protected feature 'gender', i.e. $A = \{'female', 'male'\}$. CAVs are trained for two concepts, C1: X['gender'] == 'female' and C2 : X['gender'] == 'male'. Then, Demographic Parity can be quantified as follows,

$$\epsilon = Pr[\hat{y} ='> 50K'|A =' Female'] - Pr[\hat{y} ='> 50K'|A =' Female']$$

where $\hat{y}$ is the predicted label by a trained model. As ϵ is close to zero, the classifier is fair.

With multiple versions of modified training data that are unbalanced in terms of ϵ, TabNet models are trained to have different fairness levels of Demographic Parity. Figure 6 shows TCAV scores and TCAV angles of multiple TabNets. By the definition, when ϵ is far from zero in negative direction, it implies that the classifier is more biased toward $A =$ 'male', and we observe that it is agreed with TCAV results: in Figure 6(a), TCAV angle is much larger for C1 but is smaller for C2 compared to $90°$, and in Figure 6(b), TCAV score is negatively large for C1, but positively large for C2. For both of the plots, the difference between TCAV results gets close to zero as ϵ goes to zero.

6 Conclusion and future work

In this paper, we presented a method to extend the use of concept attribution interpretability, TCAV, to the domain of tabular learning using TabNet. Our experiments suggested TCAV results in interpretability that agrees with natural high-level concepts indeed. We then showed how they can be used to quantitatively measure the fairness notion of a trained neural networks, and provided empirical evidence on the correspondence between TCAV-based fairness and group fairness such as Demographic Parity.

TCAV can be used to any deep neural networks where we can extract deep representations from layers of the model. Hence, the fairness tool presented here is applicable to other deep neural networks for tabular learning, other than TabNets. Moreover, there are several promising avenues for future work based on our TCAV-based fairness. While we have provided empirical evidence on its relation to Demographic Parity, studying its correspondence to other group fairness notions or providing theoretical evidence would bring new insights. Finally, one could ask for ways to extend this idea for further sophisticated diagnosis of fairness, such as finding a proxy for a protected features.

References

[1] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[2] S. O. Arik and T. Pfister. *Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors*. https://github.com/google-research/google-research/tree/master/tabnet.

[3] S. O. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442*, 2019.

[4] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[5] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018.

[6] D. Dua and C. Graff. UCI machine learning repository, 2017.

[7] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org.

[8] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[9] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. Patwary, M. Ali, Y. Yang, and Y. Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[10] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. *Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors*. https://github.com/tensorflow/tcav.

[11] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

[12] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pages 6076–6085, 2017.

[13] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[14] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: Removing noise by adding noise. arxiv 2017. *arXiv preprint arXiv:1706.03825*.

[15] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[16] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, 2019.