DEBUGGING TRAINED MACHINE LEARNING MODELS USING FLIP POINTS

Roozbeh Yousefzadeh
Dept. of Computer Science
University of Maryland
College Park, MD 20742, USA
roozbeh@cs.umd.edu

Dianne P. O'Leary
Dept. of Computer Science and
Inst. for Adv. Computer Studies
University of Maryland
College Park, MD 20742, USA
oleary@cs.umd.edu

ABSTRACT

We study and debug a trained machine learning model by interpreting it as a mathematical function and investigating its *flip points*. A flip point is any point that lies on the boundary between two output classes: e.g. for a model with a binary yes/no output, a flip point is any input that generates equal scores for "yes" and "no". The flip point closest to a given input is of particular importance, and this point is the solution to a well-posed optimization problem. The flip point tells us the least change in the data that changes the prediction of the model and thus can identify model biases. We use flip points to identify flaws in the model and to debug the model by generating synthetic training data to correct the flaws. We use the Adult Income dataset as an example. Although we demonstrate our method using neural networks, the use of flip points is model agnostic, and we expect them to be useful for other machine learning models, too.

1 INTRODUCTION

We investigate and debug trained machine learning models by interpreting them as mathematical functions. Models such as neural networks are often deployed in consequential applications, yet they are criticized for their lack of easy interpretation, which undermines confidence in their use. In (Yousefzadeh & O'Leary, 2019), we proposed a novel technique, focused on interpreting trained machine learning models using *flip points*. Here, we focus on using flip points to diagnose and correct flaws in these models. We consider a general machine learning model for which the output is a continuous function of its inputs. For notation, we use $\underline{\mathbf{x}}$ for the vector of inputs to the model and $\underline{\mathbf{z}}$ for the vector of outputs.

1.1 DETERMINING FLIP POINTS

Our results are applicable to machine learning models with an arbitrary number of outputs, but as an example, consider a model with a binary "cancerous"/"noncancerous" output. We assume that the output $\underline{\mathbf{z}}(\underline{\mathbf{x}})$ is normalized (perhaps using $\mathrm{softmax}$) so that the "cancerous" and 'noncancerous" elements of the output sum to one, and we'll assume that $z_1(\underline{\mathbf{x}}) > \frac{1}{2}$ is a prediction of "cancerous", and $z_1(\underline{\mathbf{x}}) < \frac{1}{2}$ is a prediction of "noncancerous". If $z_1(\underline{\mathbf{x}}) = \frac{1}{2}$, then the prediction is undefined.

Now, given a prediction $z_1(\underline{\mathbf{x}}) \neq \frac{1}{2}$ for a particular input $\underline{\mathbf{x}}$, we want to investigate how changes in $\underline{\mathbf{x}}$ can change the prediction, for example, from "cancerous" to "noncancerous". In particular, it would be very useful to find the *least change* in $\underline{\mathbf{x}}$ that makes the prediction change.

Since the output of the model is continuous, $\underline{\mathbf{x}}$ lies in a region of points whose output z_1 is greater than $\frac{1}{2}$, and the boundary of this region is continuous. So what we really seek is a nearby point on that boundary, and we call points on the boundary *flip points*.

The closest flip point $\hat{\mathbf{x}}^c$ is the solution to an optimization problem

$$\min_{\underline{\hat{\mathbf{x}}}} \|\underline{\hat{\mathbf{x}}} - \underline{\mathbf{x}}\|, \tag{1}$$

where $\|.\|$ is a norm appropriate to the data. Our only constraint is $z_1(\underline{\hat{\mathbf{x}}}) = \frac{1}{2}$.

Specific problems might require additional constraints; e.g., if $\underline{\mathbf{x}}$ is an image, upper and lower bounds might be imposed on $\hat{\underline{\mathbf{x}}}$, and discrete inputs will require binary or integer constraints.

The optimization problem is easy to state but requires specialized optimization techniques to solve. We have been quite successful using homotopy optimization when using neural network models (Yousefzadeh & O'Leary, 2019)[Appendix A].

1.2 Comparison with similar methods in the literature

A more complete literature review is found in (Yousefzadeh & O'Leary, 2019), but here we highlight a few papers. Some recent studies have tried to find the least changes in the input that can change the decision of the model. Spangher et al. (2018) have (independently) defined *a flip set* as the set of changes in the input that can flip the prediction of a classifier. Their algorithm applies to linear classifiers only. They only use flip sets to explain the least changes in individual inputs and not to debug the model. Wachter et al. (2018) defined counterfactuals as the possible changes in the input that can produce a different output label and use them to explain the decision of a model. For a continuous model, the closest counterfactual is ill-defined, since there are points arbitrarily close to the decision boundaries, and the numerical algorithm uses enumeration, applicable only to a small number of features. Russell (2019) later suggested integer programming to solve such optimization problems, but the models used as examples are linear with small dimensionality.

2 How flip points help us debug a machine learning model

Using flip points, we can interpret the output of a model, investigate the behavior of the model, and generate synthetic data to change decision boundaries. Here we summarize these ideas, with details in (Yousefzadeh & O'Leary, 2019).

2.1 Interpret the output of a trained model

- Determine the least change in $\underline{\mathbf{X}}$ that alters the prediction of the model. The vector $\hat{\mathbf{x}}^c - \underline{\mathbf{x}}$ is an accurate and clear explanation of the minimum change in the input that makes the outcome different. This is insightful information that can be provided along with the output. For example, in a bond court, a judge could be told what changes in the features of a particular arrestee could produce a "detain" recommendation instead of a "release" recommendation.
- Assess the trustworthiness of the classification for $\underline{\mathbf{X}}$. The distances of incorrectly classified points to their flip points tend to be very small compared to the distances for correct predictions, implying that closeness to a flip point is indicative of how sure we can be of the correctness of a prediction. It is important, of course, that distance be measured in a meaningful way, with input features normalized and weighted in a way that emphasizes their importance.
- Identify uncertainty in the classification of $\underline{\mathbf{X}}$. Often, some of the inputs to a machine learning model are measured quantities that have associated uncertainties. When the difference between $\underline{\mathbf{x}}$ and its closest flip point is less than the uncertainty in the measurements, then the prediction made by the model is quite possibly incorrect, and this information should be communicated to the user.

2.2 Diagnose the behavior of trained models

- Use PCA analysis of the flip points to diagnose possible flaws and biases in the behavior of the trained model. The direction from a single data point to the closest flip point provides sensitivity information for the output of that data point. Using PCA analysis, we extend this insight to an entire dataset or to subsets within a dataset. We form a matrix with one row $\hat{\mathbf{x}}^c - \underline{\mathbf{x}}$ for each data point. PCA analysis identifies the most influential directions for flipping the output and thus the most influential features. This procedure provides clear and accurate interpretations of the model that can be compared to the modeler's expectations. Alternatively, for a given data point, PCA analysis of the directions from the data point to a collection of nearby boundary points can give insight about the shape of the decision boundaries.

- IDENTIFY THE MOST AND LEAST INFLUENTIAL POINTS IN THE TRAINING DATA. Points that are correctly classified and far from their flip points have little influence on setting the decision boundaries for a machine learning model. Points that are close to their flip points are much more influential in defining the boundaries between the output classes.
- IDENTIFY OUT-OF-DISTRIBUTION POINTS IN THE DATA AND INVESTIGATE OVERFITTING. Out-of-distribution points in the training set appear as incorrectly classified points with large distance to the closest flip point. Finding such points can identify errors in the input or subgroups in the data that do not have adequate representation in the training set (e.g., faces of people from a certain race in a facial recognition dataset (Buolamwini & Gebru, 2018)).

2.3 USE SYNTHETIC DATA TO ALTER DECISION BOUNDARIES

We can use flip points as *synthetic data*, adding them to the training set to move the output boundaries of a model insightfully and effectively.

- ADD SYNTHETIC DATA TO THE MODEL AS LABELED POINTS. Suppose that our trained model correctly classifies a training point $\underline{\mathbf{x}}$ but that there is a nearby flip point $\hat{\mathbf{x}}^c$. Adding $\hat{\mathbf{x}}^c$ to the training set, using the same classification as that for $\underline{\mathbf{x}}$, tends to push the classification boundary further away from $\underline{\mathbf{x}}$. Similarly, if our trained model makes a mistake on a given training point $\underline{\mathbf{x}}$, then we can debug the model by adding the flip point $\hat{\mathbf{x}}^c$ to the training set, giving it the same classification as $\underline{\mathbf{x}}$. Moreover, if the flip points indicate that the trained model has biases regarding a certain subset of inputs, for example, making undesirable distinctions between males and females, we can reduce that bias by adding synthetic data generated by finding the closest flip point with different gender.
- TEACH SYNTHETIC DATA TO THE MODEL AS FLIP POINTS. We can alter the decision boundaries of a trained model by adding flip points with, for example, different gender or race, not labeled as a specific class, but labeled as a flip point (output 1/2) between two classes. This can reduce biases in the model.

3 NUMERICAL RESULTS

Numerical results for the Adult Income dataset can be found in (Yousefzadeh & O'Leary, 2019). Here we focus on two financial datasets.

3.1 THE FICO EXPLAINABILITY CHALLENGE.

This dataset has 10,459 observations with 23 features, and each data point is labeled as "Good" or "Bad" risk. We randomly pick 20% of the data as the testing set and keep the rest as the training set. We regard all features as continuous, since even "months" can be measured that way.

Eliminating redundant features. The condition number of the matrix formed from the training set is 653. Rank-revealing QR factorization (RR-QR) (Chan, 1987), finds that features "MSinceMostRecentTradeOpen", "NumTrades90Ever2DerogPubRec", and "NumInqLast6Mexcl7days" are the most dependent columns; discarding them leads to a training set with condition number 59. Using the data with 20 features, we train a neural network with 5 layers, achieving 72.90% accuracy on the testing set. A similar network trained with all 23 features achieved 70.79% accuracy, confirming the effectiveness of our decision to discard three features.

Interpreting individual outputs. As an example, consider the first datapoint, corresponding to a person with "Bad" risk performance. Table 1 shows the change between the data point and its closest flip point, for 5 features. The change in other features is close to zero.

Identifying influential features. Using RR-QR on the matrix of directions between datapoints labeled "Bad" and their flip points, the three most influential features are "AverageMInFile", "NumInqLast6M", and "NumBank2NatlTradesWHighUtilization". Similarly for the directions that flip a "Good" to a "Bad", the three most influential features are "AverageMInFile", "NumInqLast6M", and "NetFractionRevolvingBurden". In both cases, "ExternalRiskEstimate" has no influence.

We perform PCA analysis on the subset of directions that flip a "Bad" to "Good" risk performance. The first principal component reveals that, for this neural network, the most prominent features with

positive impact are "PercentTradesNeverDelq" and "PercentTradesWBalance", while the features with most negative impact are "MaxDelqEver" and "MSinceMostRecentDelq". These conclusions are similar to the influential features reported by Chen et al. (2018), however, our method provides more detailed insights.

Table 1: Difference in features for FICO dataset point #1 and its closest flip point.

Data	Input #1	Closest flip point (relaxed)	Closest flip point (integer)
AverageMInFile	84	105.6	111.2
NumSatisfactoryTrades	20	24.1	24
MSinceMostRecentDelq	2	0.6	0
NumTradesOpeninLast12M	1	1.7	2
NetFractionRevolvingBurden	33	19.4	8.5

Studying redundant variables and their effect on behavior of model and generalization error. Interestingly, for the model trained on all 23 features, the most significant features in flipping its decisions are "MSinceMostRecentTradeOpen", "NumTrades90Ever2DerogPubRec" and "NumInqLast6Mexcl7days", exactly the three dependent features that we discarded. This reveals an important vulnerability of machine learning models regarding their training sets. When dependent features are included in the training set, it might not affect the accuracy on the training set, but it adversely affects the generalization error. Additionally, the decision of the trained model is more susceptible to changes in the dependent features, compared to changes in the independent features. One can argue that the dependent features are confusing the trained model.

Revealing patterns in directions to the closest flip points. Figure 1 shows the directions to the closest flip points for features "NumInqLast6M" and "NetFractionRevolvingBurden". Directions are distinctly clustered for flipping a "Bad" label to "Good" and vice versa.

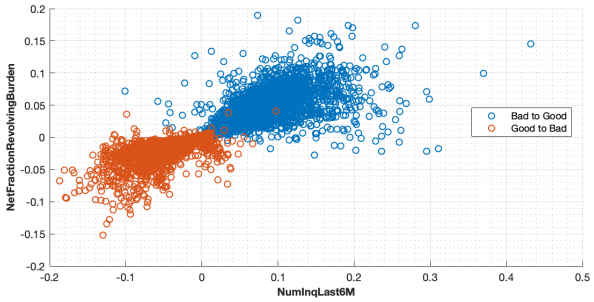

Figure 1: Directions between the inputs and their closest flip point for two influential features.

Furthermore, Figure 2 shows the directions in coordinates of the first two principal components. We can see that the directions are clearly clustered into two convex cones, exactly in opposite directions. Also, we see misclassified inputs are relatively close to their inputs while correct predictions can be close or far. Overall, misclassified inputs have similar patterns compared to correct classifications, which explains why the model cannot distinguish them from each other.

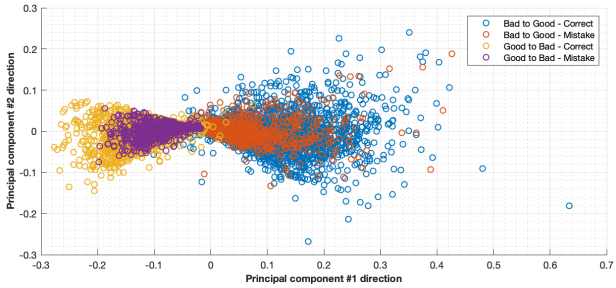

Figure 2: Change between the inputs and their flip points in the first two principal components

3.2 DEFAULT OF CREDIT CARD CLIENTS

This dataset from the UCI Machine Learning Repository (Dheeru & Karra Taniskidou, 2017) has 30,000 observations, 24 features, and a binary label indicating whether the person will default on the next payment or not.

We binarize the categorical variables "Gender", "Education", and "Marital status". The condition number of the training set is 129 which implies linear independence of features. Using a 10-fold cross validation on the data, we train a neural network with 5 layers, to achieve accuracy of 81.8% on the testing set. When calculating the closest flip points, we require the categorical variables to remain discrete.

Identifying influence of features. We perform RR-QR decomposition on the directions to the flip points. The results show that "BILL-AMT3" and "BILL-AMT5" are the most influential features, and "Age" has the least influence in the predictions. In fact, there is no significant change between the age of all the inputs and their closest flip points.

Revealing patterns in how the trained model treats the data. We briefly make some observations about the overall behavior of the trained model. The influence of gender is not significant in the decisions of the model, as only about 0.5% of inputs have a different gender than their flip points. However, we observe that those changes are not gender neutral. We see that for flipping a "no default" to "default", changing the gender from "Female" to "Male" has occurred 5 times more often than the opposite. Similarly, for flipping a "default" to "no default", gender has changed from "Female" to "Male", 5 times more often than the opposite. We also observe that changing the marital status from "Married" to "Single" is helpful in flipping "no default" predictions to "default". This kind of analysis can be performed for all the features, in more detail.

Flip points can deal with flaws and can reshape the model. Similar to the study by Spangher et al. (2018), we train a model with a subset of the training set, where young individuals are under-sampled. In both our training and testing sets, about 52% of individuals have age less than 35. We keep the testing set as before, but remove 70% of the young individuals from the training set. After training a new model, we obtain 80.83% accuracy on the original testing set. We also observe that the "Age" is the 3^{rd} most influential feature in flipping its decisions. Moreover, PCA analysis shows that having less Age has a negative impact on the "no default" prediction and vice versa.

We consider all the data points in the training set labelled as "default" that have closest flip point with older age, and all the points labelled "no default" that have closest flip point with younger age. We add all those flip points to the training set, with the same label as their corresponding data point, and train a new model using the appended training set. Investigating the behavior of the new model reveals that Age has become the 11^{th} influential feature and it is no longer significant in the first principal component of directions to flip points; hence, the bias against Age has been reduced.

Adding synthetic data to the training set has great potential to change the behavior of model, but we cannot rule out unintended consequences. Therefore, it is important to interpret the overall behavior of the reshaped model with respect to all features, and ensure that it behaves as we expect. By investigating the influential features and PCA analysis, we see that the model has been altered only with respect to the Age feature, and the overall behavior of model has not changed.

4 CONCLUSIONS

Flip points for a particular data point, either a training point or a point with unknown label, give insights into how and why a model makes a particular prediction or classification. They determine the smallest change in each input that alters the prediction of the model. Additionally, distance to the flip points assesses the trustworthiness of the output when the input features have uncertainties.

Flip points also enable us to debug a trained model, diagnosing flaws and biases. PCA analysis of the directions identifies the most important features and combinations of features, and this information can be checked by the data expert for plausibility. Flip points themselves are candidates for synthetic data points that can be added to the training set to change the behavior of the model.

Thus, flip points are a valuable tool for interpreting and debugging machine learning models.

REFERENCES

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Conference on Fairness, Accountability and Transparency, pp. 77–91, 2018.

Tony F Chan. Rank revealing QR factorizations. Linear Algebra and its Applications, 88:67–82, 1987.

Chaofan Chen, Kangcheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. An interpretable model with globally consistent explanations for credit risk. arXiv preprint arXiv:1811.12615, 2018.

Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Chris Russell. Efficient search for diverse coherent explanations. In Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19, pp. 20–28, 2019.

Alexander Spangher, Berk Ustun, and Yang Liu. Actionable recourse in linear classification. In Proceedings of the 5th Workshop on Fairness, Accountability and Transparency in Machine Learning, 2018.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harvard Journal of Law & Technology, 31(2), 2018.

Roozbeh Yousefzadeh and Dianne P O'Leary. Interpreting neural networks using flip points. arXiv preprint arXiv:1903.08789, 2019.