COUNTERFACTUAL ACCURACIES FOR ALTERNATIVE MODELS

Umang Bhatt
University of Cambridge
usb20@cam.ac.uk

Muhammad Bilal Zafar
Bosch Center for AI (BCAI)
muhammadbilal.zafar@de.bosch.com

Krishna Gummadi
MPI Software Systems
gummadi@mpi-sws.org

Adrian Weller
University of Cambridge &
The Alan Turing Institute
aw665@cam.ac.uk

ABSTRACT

Typically we fit a model by optimizing performance on training data. Here we focus on the case of a binary classifier that predicts 'yes' or 'no' for any given test point. We explore a notion of confidence in a particular prediction by asking: *If we were to fit an alternative classifier from our model class to the same training data, how much training accuracy would we have to give up so that the prediction for the test point would change?*

1 INTRODUCTION

Suppose you are a loan officer using an algorithmic decision-making system to support your abilities. You have done your due diligence by scrutinizing the training data then optimizing and evaluating the model's performance, and now feel comfortable to use the model on unseen test points in the real world. Yet upon seeing a prediction on a particular test individual which would lead to their loan being denied, you wonder before making your final decision: what if there were an alternative model – with only slightly worse performance on the training data – which instead granted this individual a loan? If so, this could give you pause before potentially making a mistake by denying the loan.

The idea that multiple classifiers can fit a training dataset well, leading to different stories about the relationship between the input features and output response, is not new (Breiman, 2001), but has received recent attention under the theme of *predictive multiplicity* (Fisher et al., 2019; Marx et al., 2019). Fisher et al. (2019) consider the ϵ-set of models, i.e. those whose empirical training loss is within ϵ of a baseline classifier.

In this paper, we solve a similar problem: we want to find the minimum ϵ such that the empirical ϵ-set contains at least one model with different predictions for a selected test point. Specifically, if we have a classifier that denies a test individual of a loan, we are interested in finding an alternate classifier that would have granted the individual a loan. Using our approach, called *counterfactual accuracy*, we first quantify how these classifiers differ, provide intuition behind our approach, and finally experiment with our approach on real-world datasets.

2 METHODOLOGY

We restrict ourselves to standard binary classification tasks, where our goal is to find a classifier f from a family of functions $\mathcal{F}$ such that f learns a mapping between inputs $\boldsymbol{x} \in \mathbb{R}^d$, vectors of d real-valued features, and labels, $y \in \{-1, 1\}$. Given a training dataset $\mathcal{D}^n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ from some underlying, unknown distribution $\mathcal{D}$ and a nonnegative loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, our goal is to learn a classifier f that minimizes training error yet performs well on *unseen* test data. The expected loss of f is given by: $R(f) = \mathbb{E}_{\mathcal{D}}\left[\ell\left(f(\boldsymbol{x}), y\right)\right]$. Since we do not know $\mathcal{D}$, we calculate the average loss $\widehat{R}$ over the training dataset, $\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(f(\boldsymbol{x}_i), y_i\right)$. If we want our classifier to learn a linear decision boundary parameterized by $\boldsymbol{\theta}$, we assume it has the following form: $f(\boldsymbol{x};\boldsymbol{\theta}) = \text{sgn}(\boldsymbol{\theta}^T\boldsymbol{x})$, where $\boldsymbol{\theta}^T\boldsymbol{x}$ represents the distance between the $\boldsymbol{x}$ and the linear separator and where sgn$(\cdot)$ is the standard sign function. For notational simplicity, we drop the parameters $\boldsymbol{\theta}$ from $f(\boldsymbol{x};\boldsymbol{\theta})$ in the rest of the document. For classifiers, irrespective of loss function, we define the number of training errors, M, as: $M(f) = n\widehat{R}_{0/1}(f) = \sum_{i=1}^{n} \mathbb{1}\left[f(\boldsymbol{x}_i) \neq y_i\right]$. Let $M_\text{o} = M(f_\text{o})$ be the number of errors made by the empirical risk minimization (ERM) solution on $\mathcal{D}^n$. We now outline our approach and then highlight its properties.

2.1 Our Approach

In an ERM setup, one finds the optimal classifier f_o parameterized by $\boldsymbol{\theta}$ as follows:

$$f_\text{o} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{N} \ell\left(f(\boldsymbol{x}_i), y_i\right)$$

. Given an unseen test point $\boldsymbol{z}$, its predicted label is given by $f_\text{o}(\boldsymbol{z})$. Now, we want to find an alternate classifier $f_{\boldsymbol{z}}$ parameterized by $\boldsymbol{\theta'}$ such that we minimize average loss over $\mathcal{D}^n$ *with the condition* that the predicted label of $\boldsymbol{z}$ if flipped, that is, $f_\text{o}(\boldsymbol{z}) \neq f_{\boldsymbol{z}}(\boldsymbol{z})$. With this setup, we find an alternate classifier via:

$$\begin{gathered} f_{\boldsymbol{z}} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{N} \ell\left(f(\boldsymbol{x}_i), y_i\right) \\ \text{s.t. } f_\text{o}(\boldsymbol{z}) \neq f_{\boldsymbol{z}}(\boldsymbol{z}) \end{gathered} \tag{1}$$

Let $\mathcal{F}'_{\boldsymbol{z}} = \{f \in \mathcal{F} : f(\boldsymbol{z}) \neq f_\text{o}(\boldsymbol{z})\}$. Note that $\mathcal{F}'_{\boldsymbol{z}} \subseteq \mathcal{F}$. Let the number of training errors made by $f_{\boldsymbol{z}}$ be denoted by $M_{\boldsymbol{z}} = M(f_{\boldsymbol{z}})$. The expected loss of $f_{\boldsymbol{z}}$ is given by: $R(f_{\boldsymbol{z}}) = \mathbb{E}_{\mathcal{D}}\left[\ell\left(f_{\boldsymbol{z}}(\boldsymbol{x}), y\right)\right]$.

Definition 1 (Extra Loss)**.** Let the *extra loss* (EL) of $f_{\boldsymbol{z}}$ and f_o be:

$$C(f_{\boldsymbol{z}}, f_\text{o}) = R(f_{\boldsymbol{z}}) - R(f_\text{o}) = \mathbb{E}_{\mathcal{D}}\left[\ell\left(f_{\boldsymbol{z}}(\boldsymbol{x}), y\right)\right] - \mathbb{E}_{\mathcal{D}}\left[\ell\left(f_\text{o}(\boldsymbol{x}), y\right)\right] \tag{2}$$

$C(f_{\boldsymbol{z}}, f_\text{o})$ tells us how much our loss suffers when we introduce a constraint to conflict the predictions of $f_{\boldsymbol{z}}$ and f_o on a test point $\boldsymbol{z}$. For ease of reading, we let $C_{\boldsymbol{z}} = C(f_{\boldsymbol{z}}, f_\text{o})$. Since we do not know $\mathcal{D}$, we calculate an empirical variant over our training dataset. The average loss of $f_{\boldsymbol{z}}$ over our training dataset is given by $\widehat{R}(f_{\boldsymbol{z}}) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(f_{\boldsymbol{z}}(\boldsymbol{x}_i), y_i\right)$.

Definition 2 (Empirical Extra Loss)**.** Let the *empirical extra loss* (EEL) of $f_{\boldsymbol{z}}$ and f_o over a training dataset $\mathcal{D}^n$ be given by:

$$\widehat{C}(f_{\boldsymbol{z}}, f_\text{o}) = \widehat{R}(f_{\boldsymbol{z}}) - \widehat{R}(f_\text{o}) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(f_{\boldsymbol{z}}(\boldsymbol{x}_i), y_i\right) - \ell\left(f_\text{o}(\boldsymbol{x}_i), y_i\right) \tag{3}$$

Specifically when we let $\ell = \ell_{0/1}$, the training accuracy of classifier f is given by $1 - \widehat{R}(f)$. We can rewrite the *empirical extra loss* $\widehat{C}_{\boldsymbol{z}}$ as the difference in training accuracy between f_o and $f_{\boldsymbol{z}}$: we call this the *counterfactual accuracy*, denoted by $\tilde{C}_{\boldsymbol{z}}$.[1]

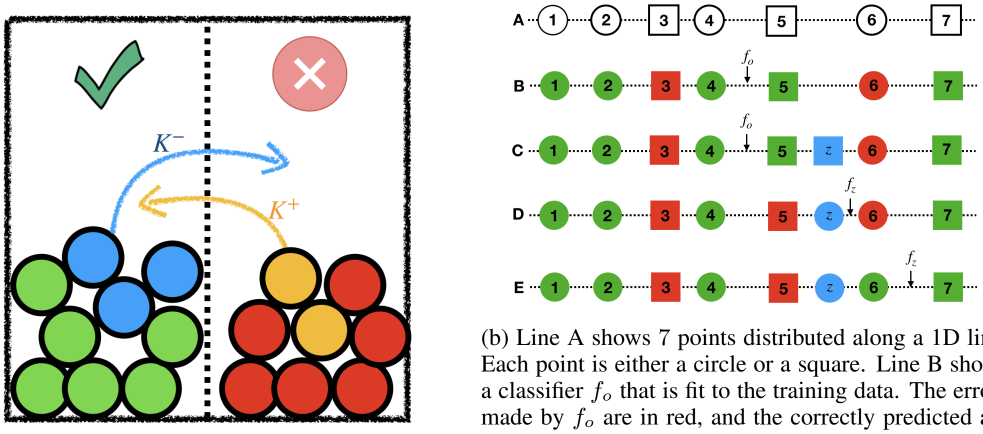
Definition 3 (Counterfactual Accuracy)**.** Let the *counterfactual accuracy* of $f_{\boldsymbol{z}}$ and f_o over a training dataset $\mathcal{D}^n$ be given by:

$$\tilde{C}_{\boldsymbol{z}} = \widehat{R}(f_{\boldsymbol{z}}) - \widehat{R}(f_\text{o}) = \left(1 - \widehat{R}(f_\text{o})\right) - \left(1 - \widehat{R}(f_{\boldsymbol{z}})\right) \tag{4}$$

2.2 Predicted Label Flips in Training Data

We can highlight the set of training points whose predicted labels flip as a result of the new constraint. The number of training points whose classification went from correct under f_o to incorrect under $f_{\boldsymbol{z}}$ is given by: $K^- = \sum_{i=1}^{n} \mathbb{1}\left[f_{\boldsymbol{z}}(\boldsymbol{x}_i) \neq y_i\right] \mathbb{1}\left[f_\text{o}(\boldsymbol{x}_i) = y_i\right]$. Similarly, the number of training points whose classification went from incorrect under f_o to correct under $f_{\boldsymbol{z}}$ is:

[1] Here we do not use "counterfactual" in the causal sense as in Pearl (2009); we use the term since it captures what **would** happen to training accuracy if we were to constrain our optimization objective to flip the predicted label of a specific datapoint from its ERM prediction.



(a) Green points are those f_{o} and $f_{\boldsymbol{z}}$ get correct (6). Red points are those f_{o} and $f_{\boldsymbol{z}}$ get incorrect (6). Blue points are those f_{o} gets correct but $f_{\boldsymbol{z}}$ gets incorrect ($K^- = 3$). Gold points are those f_{o} gets incorrect but $f_{\boldsymbol{z}}$ gets correct ($K^+ = 2$). $f_{\boldsymbol{z}}$ gets the blue and red points incorrect (9), but gets the green and gold points correct (8). The *counterfactual accuracy*, $\tilde{C}_{\boldsymbol{z}}$ between f_{o} and $f_{\boldsymbol{z}}$, is $\frac{1}{17}$

(b) Line A shows 7 points distributed along a 1D line. Each point is either a circle or a square. Line B shows a classifier f_o that is fit to the training data. The errors made by f_o are in red, and the correctly predicted are in green. Note that any f_o between point 4 and point 5 would achieve optimal accuracy. Line C shows a test point z that appears on the line and will be predicted to be a square under f_o. Line D shows a candidate classifier f_z that flips the label of z but reaches sub-optimal performance. Line E shows an alternate classifier f_z that flips the label of z and reaches optimal performance. Note that any f_o between point 6 and point 7 would achieve a *counterfactual accuracy*, $\tilde{C}_z$, of 0

Figure 1: Building intuition around *counterfactual accuracy*

$K^+ = \sum_{i=1}^{n} \mathbb{1}\left[f_{\boldsymbol{z}}(\boldsymbol{x}_i) = y_i\right] \mathbb{1}\left[f_{\text{o}}(\boldsymbol{x}_i) \neq y_i\right]$. Thus, the total number of predicted labels that changed from f_{o} to $f_{\boldsymbol{z}}$ is $K = K^- + K^+$. Let K_i be the number of points whose predicted labels changed when we added a constraint to flip the label of test point, $\boldsymbol{z}_i$. We can relate the number of flipped points K to the number of training errors M, and can bound K, K^+, and K^- when constraining for test point $\boldsymbol{z}_i$ as:

$$n \geq K_i \geq 0; \;\; M_{\text{o}} \geq K_i^+ \geq 0; \;\; n - M_{\text{o}} \geq K_i^- \geq 0$$

In Figure 1a, we visualize label flips for a contrived example and walk through an example of how one would calculate *counterfactual accuracy*. Under 0/1 loss, we can rewrite *counterfactual accuracy*, $\tilde{C}_{\boldsymbol{z}}$, in terms of label flips.

$$\tilde{C}_{\boldsymbol{z}} = \frac{M_{\boldsymbol{z}} - M_{\text{o}}}{n} = \frac{K^- - K^+}{n} \tag{5}$$

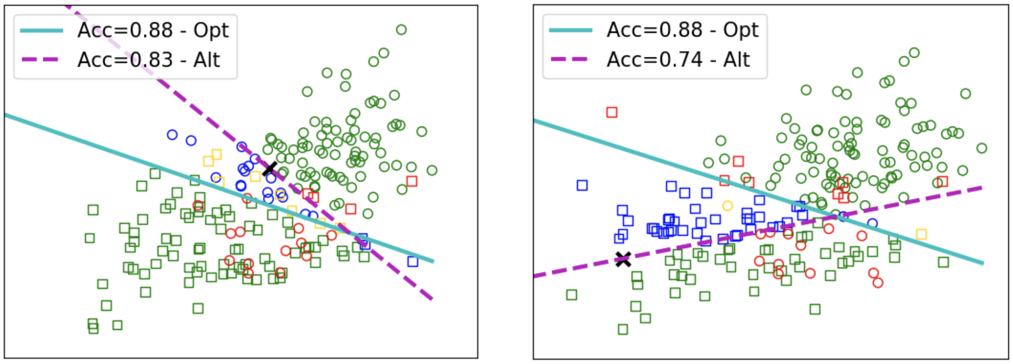
If we rewrite the training dataset errors of the alternate classifier $f_{\boldsymbol{z}}$ as $M_{\boldsymbol{z}} = M_{\text{o}} + K^- - K^+$, the second equality follows. *Counterfactual accuracy*, a specific case of EEL, captures how much model suffers (in terms of training error) from the additional constraint in the ERM optimization.

We could also run our procedure on training datapoints. Once we add a constraint to flip the predicted class of a training point $\boldsymbol{x}_i$, we could obtain $f_{\boldsymbol{x}_i}$ via an ERM, except we would exclude $\boldsymbol{x}_i$ from the training set; that is, we would only train on $n-1$ samples. When we calculate the EEL of the alternate classifier as well as the number of label flips K_i, we would only use $\mathcal{D}_i^n = \mathcal{D}^n \setminus \{\boldsymbol{x}_i, y_i\}$.

3 Experiments

To build further intuition around *counterfactual accuracy*, consider a one dimensional setting. In Figure 1b, we distribute points across a line: our goal is to build a classifier, which will be a point on the line, that distinguishes between circles and squares. With this example, we illustrate that our alternate classifier need not lie directly near the test point $\boldsymbol{z}$ and that the accuracy of the optimal and alternate classifiers can be identical, implying that the *counterfactual accuracy* $\tilde{C}_z = 0$.

In practice, we would ideally optimize the $0/1$ loss for all datasets; unfortunately, this can be computationally expensive. Instead, we optimize over surrogate convex losses (hinge loss and log loss).



(a) Using log loss, the *counterfactual accuracy*, $\tilde{C}_{\boldsymbol{z}_1}$, at $\boldsymbol{z}_1$ is 0.05

(b) Using hinge loss, the *counterfactual accuracy*, $\tilde{C}_{\boldsymbol{z}_2}$, at $\boldsymbol{z}_2$ is 0.14

Figure 2: Experimentation with *counterfactual accuracy*: let the solid cyan line represent f_o and the dotted magenta line represent $f_{\boldsymbol{z}}$ where $\boldsymbol{z}$ is the point denoted by the black X. Let circle and square denote the true label of each data point. Green points are those that both classifiers get correct. Red points are those that both classifiers get wrong. Blue points are those that f_o got right but $f_{\boldsymbol{z}}$ got wrong. Gold points are those that f_o got wrong but $f_{\boldsymbol{z}}$ got right.

When using hinge loss, we additionally regularize the norm of $\boldsymbol{\theta}$ to train a support vector machine in its primal formulation (Chapelle, 2007). In Figure 2, we find an optimal classifier for a synthetic dataset of two overlapping Gaussians, as used in Zafar et al. (2017)), and then we constrain for a random test point to obtain *counterfactual accuracy*. The loss function used is denoted inline. Though the alternate model's decision boundary appears nearly on top of the constrained test point in the synthetic dataset example, this is an artifact of the solver we use (Sequential Least Squares Programming) and not of our formulation.

We also find *counterfactual accuracies* for the following datasets: Adult (predicting income level given 1994 US census information) (Dua & Graff, 2017) and COMPAS (predicting criminal re-offense given demographics and history) (Angwin et al., 2016). On the Adult dataset's test set, we get an average *counterfactual accuracy* of 0.667%, which we means we gain less than a percentage of training error when we introduce our constraint for a test point. On the Adult test set, we get an average number of label flips of 225.08. Similarly, for the COMPAS dataset, we get an average *counterfactual accuracy* of 1.437% over a test set, and get an average number of label flips of 260.43. On both datasets, the large number of label flips and the small *counterfactual accuracy* can be attributed to the large dimensionality of the data. When we recalculate *counterfactual accuracy* using Equation 5, we get the exact same values as listed above, suggesting that the use of a surrogate loss may not hinder our objective.

On both real world datasets, we find that the Kendall rank correlation coefficient between the *counterfactual accuracy* of $\boldsymbol{z}$ and the distance from $\boldsymbol{z}$ to the decision boundary is strongly positive: we think this due to limited flexibility of our model class, $\mathcal{F}$. Were we to introduce non-linearity into the model, we may see a performance bump and a decrease in this correlation.

4 Related Work

To the best of our knowledge, no existing work uses difference in average loss between optimal and constrained alternate classifiers as a proxy for other metrics; however, there are a few works that relate to our formulation. Firstly, Marx et al. (2019) does analysis of a "pathological" classifier which forces a datapoint to have a different classification than that of a baseline classifier; however, whereas they using a mixed integer program (MIP) to find an optimal solution only for 0/1 loss, we add our constraint to any standard, convex loss function. The empirical ϵ-Rashomon set in Fisher et al. (2019) (ϵ-level set in Marx et al. (2019)) is defined as: $\widehat{S}_\epsilon = \{f \in \mathcal{F} : \widehat{R}(f) \leq \widehat{R}(f_o) + \epsilon\}$. $\widehat{S}_\epsilon$ can be seen as the set of all classifiers in $\mathcal{F}$ that have an average loss no more than ϵ greater than the average loss of f_o. While these works study how to deal with varying predictions in $\widehat{S}_\epsilon$, we essentially solve a dual problem where we want to find the minimum ϵ such that the empirical ϵ-Rashomon set contains at least one model with different predictions for a test point of interest, $\boldsymbol{z}_i$. More concretely, we want to find $\epsilon_i > 0$, where $\epsilon_i = \min\ \epsilon \quad \text{s.t.}\ \exists f_i \in \widehat{S}_\epsilon : f_i(\boldsymbol{z}_i) \neq f_{\text{o}}(\boldsymbol{z}_i)$. Moreover, Letham et al. (2016) looks across $\widehat{S}_\epsilon$ to identify the two models which have maximally different predictions. We do something similar but different: we ask how far does your average loss of a model need to suffer in order to change the prediction of an unseen test point.

5 Conclusion

In this paper, we propose the concept of *counterfactual accuracy*, the empirical extra loss suffered by a classifier when we force a point's prediction to flip. We can use this quantity to estimate the uncertainty associated with a point. Future work can look at how the parameters between f_{o} and $f_{\boldsymbol{z}}$ change, $||\boldsymbol{\theta}' - \boldsymbol{\theta}||$, as this may convey information about the point being constrained. Moreover, a Bayesian formulation, which would model the posterior over the parameters given the data $P(\boldsymbol{\theta}|\mathcal{D}^n)$, is suitable for this problem; in this setup, we want to know the difference in the probability mass of parameters that classifies a point as 1 and the probability mass that classifies the same point as -1. In future work, we hope to address a Bayesian variation of *counterfactual accuracy* and manage the intractability of estimating the difference in probability mass.

Acknowledgements

AW acknowledges support from the David MacKay Newton research fellowship at Darwin College, The Alan Turing Institute under EPSRC grant EP/N510129/1 & TU/B/000074, and the Leverhulme Trust via the Leverhulme Centre for the Future of Intelligence (CFI). UB acknowledges support from DeepMind and the Leverhulme Trust via CFI and from the Partnership on AI.

References

David Alvarez-Melis, Hal Daumé III, Jennifer Wortman Vaughan, and Hanna Wallach. Weight of evidence as a basis for human-oriented explanations. *arXiv:1910.13503*, 2019.

Julia Angwin, Jeff Larson, Surya Mattu, Lauren Kirchner, and ProPublica. Machine bias, 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.

Olivier Chapelle. Training a support vector machine in the primal. *Neural computation*, 19(5): 1155–1178, 2007.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019. URL http://jmlr.org/papers/v20/18-760.html.

Benjamin Letham, Portia A Letham, Cynthia Rudin, and Edward P Browne. Prediction uncertainty and optimal experimental design for learning dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6):063110, 2016.

Charles T. Marx, Flavio du Pin Calmon, and Berk Ustun. Predictive multiplicity in classification. *arXiv:1909.06677*, 2019.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pp. 962–970, 2017.