An Enabling Technology

Evaluators like the independent trio evaluator can enable many applications that can increase the safety and fairness of AI systems. This notebooks explores some of them and gives suggestions for possible experiments to carry out.

Auto-ML

The vast majority of data that an AI system will process is unlabeled. Otherwise, why was the AI used? Can we leverage that unlabeled data somehow? Could we devise protocols that can train good models using unlabeled data? And can we do this robustly? The experiments in this section explore these questions.

Finding least correlated ensembles on large unlabeled test sets

The problem of correlated classifiers is the Achilles heel of algebraic evaluators. This presents the engineer of robust AI systems that uses them with a challenge - how do we build ensembles of classifiers that are, as best we can, independently correlated in their errors on a large unlabeled test set? We use the failures of the independent trio formula to reject possible candidates. Here are some suggestions on what sort of scans you could do to see how well this works on the data and the classifiers of interest to you.

1. We expect classifiers trained differently to perform differently. Thus differences in training decorrelate their errors also. By training on disjoint sets of data, features, or algorithms you can get pair error correlations in the order of 5% or less. Carry out training protocols that scan for the most independent classifier ensembles by discarding the ones that make the evaluator fail.

- **1.1.** Are there bounds on the correlations observed in the selected ones?
- **1.2.** What about the rejected ones? Were good configurations rejected? What was the least error correlation that triggered the rejection? Is there a difference between failures that resulted in real algebraic numbers versus those that gave imaginary algebraic numbers?
- **1.3.** Now measure the error in the evaluation estimates of the accepted configurations. Compare that to the error in the rejected ones.
- **1.4.** Also compare it to various other random guessing that would be sensible - a real number between 0 and 1. These random guessers are good proxies for having no information except the one that you need to make in order to decode the multiple answers given by algebraic evaluators. For example, the comparison we made between the evaluator and ground truth in TheCoreTheorem.nb was based on picking the one based on our side knowledge about the value of the true prevalence. For binary classification this means that a fair comparison with random performers would mean that you guess a value of the prevalence with a width specified by your side knowledge. In our case, random guessing on the 0 to 1/2 interval.

An example experiment can be found under experiments/AutoML/LeastErrorCorrelatedTrios.nb.

Fairness Alarms

Different methods for ameliorating the harm of biased AI systems have been proposed and criticized. Common to some of them, is the use of purely observational statistics like false positives, etc. But how do we know that a system that was designed for X measure of fairness is actually working on unlabeled data? The experiment in this section suggests one possible way to use the independent trio evaluator to help with this problem.

1. Various approaches to creating and measuring the fairness of AI algorithms have been proposed in the scientific literature. There is even some disagreement about whether some approaches are better than others. The experiments proposed here are based on a bias mitigation training strategy by,
2. “Contrastive Counterfactual Fairness in Algorithmic Decision-Making” by Mutlu, Yousefi, and Garibay discusses a way to use counterfactual reasoning ideas from Causal Data Science. They propose an algorithm for augmenting training data to mitigate the bias of AI algorithms. Their approach can be combined with the algebraic evaluators to create a monitor for fairness on unlabeled data.

   2.1. The UCI Adult dataset was used in their paper. Using the same dataset, implement their data augmentation technique to create fairer AI algorithms. Using the same ideas about AutoML mentioned before, suppose that you constructed a monitoring ensemble that was roughly independent on its test errors. Can you use the accuracy estimates given by the algebraic evaluator for the members of that ensemble to, in turn, estimate their fairness on unlabeled data? What is the estimation error of your fairness thermometer on unlabeled data?

Error-Correction Codes

Hamming famously quipped - “Damn it, if the computer knows I am wrong, why can’t it fix it?” He went on to invent error correcting codes. What error-correcting codes can you invent with the independent trio evaluator? Here we propose one way to build a code to help you get going with your own ideas.

The connection between algebraic evaluators and error-correcting for AI algorithms is as follows. Constructing evaluation ideals for noisy judges relies on the implicit assumption that classification labels allow us to disjointly separate the test set. This means that the following equation is always true by definition,

$f_{\text{voting pattern}} = f_{\text{voting pattern when label } a \text{ was correct}} + f_{\text{voting pattern when label } b \text{ was correct}} + f_{\text{voting pattern when label ....}}$

For binary classification that means the number of instances of a particular voting pattern is a sum of two unknown integers. Each unknown integer tells you the correct number of instances for that label. Error-correcting with this observation follows immediately. For each voting pattern, compute your estimate of the number of instances for each label. Error minimization proceeds by labeling all the instances of the label that gave the smallest count as the other label.

A quick example illustrates this,

10 instances of (a,b,a) = (3 when “a” was correct) + (7 when “b” was correct)
We minimize the error in these decisions by then proceeding to label all these instances as actually being “b” instances. Note that we picked an example that would give the opposite answer than ensemble decision algorithm commonly known as “wisdom of the crowd.”

1. How effective is this error correcting method? Try it out using your monitoring ensemble schemes. How often does it reduce or increase errors when you plot it against some statistic of the ensemble’s error correlations?

2. Another approach to error correction is to create as large of a monitoring ensemble as possible and then check for good members to include in the decisioning ensemble. We know from Condorcet’s famous Jury Theorem that better than average independent judges have high odds of returning correct decisions. Using four classifiers allows you to test 6 possible trios for failures of the independent evaluator. Can your deciding ensembles decrease their error by eliminating the estimates from failing trios and just using the decisions from apparently independent ones?