Resolving Power Online Resource 1

Colin Beam PhD

Abstract

This supplement provides additional examples and details of the resolving power approach using the diabetes readmissions data from the main text. The findings are preliminary though they can serve as a basis for future research using additional datasets.

1 Linear approximation of signal curves

This section uses Appendix A's linear approximation method to estimate resolving power. We apply the linear approach to the diabetes readmissions data and then compare the results to those from the main text. Previously, we formed a fine grid of many $(\text{AUROC}, \text{AUPRC})$ pairs both above and below the baseline model. Now we only find three points: The baseline, one point by nudging the class distributions slightly closer, and one point by nudging them slightly further apart. The approach from Section 6 can be used to determine the nudge size (the $\delta + \epsilon$) needed to achieve a specific AUROC increment. We use a .001 AUROC increment and then fit a line to the three points to approximate the signal curve.

Results are shown in Table 1 below. The AUROC and AUPRC resolving power estimates are nearly identical to those obtained using the more comprehensive grid approach (shown in Table 2 from the main text). In this example, then, a linear signal curve provides a good approximation while requiring much less computation. Moreover, recall that resolving power's central aim is to inform metric choice for model search. This task only requires approximate relative magnitudes; precise resolving power estimates are not necessary.

Table 1 Summary of linear approximation results.

metric	κ	resolving power
AUROC	0.0148	67.6
AUPRC	0.0250	40.1
H-measure	0.0190	52.7

Heretofore, the focus has exclusively been on comparisons between the AUROC and the AUPRC, but resolving power can easily be applied to other metrics. One criticism of the AUROC is that it implies that different classifiers apply different costs to the two types of misclassification.[1] But costs should depend on the relative severity of the outcomes, not on which classifier is used. The H-measure addresses this issue by using the same cost distribution to compare different classifiers (Hand, 2009; Hand and Anagnostopoulos, 2023).

Table 1 shows the resolving power estimate for the H-measure using the default beta prior cost distribution as described in Hand and Anagnostopoulos (2023). The H-measure has resolving power that is intermediate of the AUROC and the AUPRC. This is another example in which different evaluation metrics may be more suitable for different stages of model development. Despite its theoretical shortcomings, the AUROC may still be preferable for model search. Once search is complete, one could then switch to the more informative H-measure to communicate the results.

2 Resolving power as a guide for model search

This section illustrates how resolving power can be incorporated into the model development process. It also serves as an initial test of the framework's utility in guiding model search. The approach taken resembles Kohavi et al (1995)'s study that compares how various resampling methods affect classifier accuracy for several real-world datasets. Since the population characteristics of real data cannot be known, Kohavi uses the holdout method to estimate the "true" accuracy with a relatively large test set. This section employs a similar method by repeatedly applying the three-way-holdout method to the diabetes readmissions data. Twenty percent of the data is used to fit the model, five percent to validate, and the remaining seventy-five percent is the holdout set.

The first step is to use the training data to estimate resolving power for the candidate metrics. To keep things simple we only compare the AUROC with the AUPRC. We use XGBoost for the baseline model, which is a popular implementation of gradient boosted trees (Chen and Guestrin (2016)). The baseline model trains for 25 rounds with the following hyperparameters:

```
# xgb baseline parameters
baseline_params <- list(
  num_rounds = 25,
  objective = "binary:logitraw",
  booster = "gbtree",
  # xgboost parameter list:
  eta = 0.1,
  gamma = 0,
  max_depth = 6,
  min_child_weight = 1,
  max_delta_step = 0,
  subsample = 1,
```

[1] As one reviewer observed in their comments, this property equally applies to the AUPRC.

```
  colsample_bytree = 1,
  lambda = 1,
  alpha = 0
)
```

Once we obtain the baseline effects we can estimate resolving power in the usual way. Since the linear approximation method appears to give good estimates, we also use it for this analysis. Table 1 shows summary statistics for 30 simulation runs in which the diabetes data is randomly divided into training and test sets, and resolving power is estimated on the training set.

Table 2 Summary of resolving power estimates using 25 percent of the diabetes readmissions data over 30 random splits. The AUROC had greater resolving power for every random split.

ROC mean	ROC median	PRC mean	PRC median
33.42	33.47	12.65	12.7

The AUROC's resolving power is on average between 2 to 3 times larger than that of the AUPRC. Also note that the power estimates are substantially smaller than estimates from the previous section. This is expected since we are now only using one quarter of the data to train and validate the model.

If the resolving power estimates are accurate then the AUROC should be a better guide for finding the best models. Whether it is in fact better will depend on several factors. First, the estimates may not be accurate, which probably depends most strongly on the additive improvements assumption. That topic is explored in the next section. Another key factor is the topography of the model search space. If the evaluation points lie across a steep region of the search space then the model quality signal will overwhelm the noise for all metrics. The different evaluation metrics will always agree so the choice of metric for search will not matter. Conversely, search may occur over a relatively flat portion of the space. For instance, some of the points in the tuning grid may span a local optimum. Then once again, the choice of evaluation metric will clearly not matter. But now we should expect the metrics to agree on about $1/n \times 1/n$ proportion of the trials where n is the number of evaluation points lying on the optimum. Thus, for resolving power to make a difference, we need to search in a region of intermediate steepness. Finally, a more fundamental concern is that the small gains achieved in training may prove ephemeral when the model is applied in the real world, as Hand (2006) argues will often be the case.

The above concerns noted, let us proceed with our test of the resolving power prediction. We repeatedly train and tune XGBoost on twenty-five percent of the sample and then assess performance on the remaining holdout set. The experiment uses the same hyperparameters as above except now applies random search over a two-parameter grid with eta $\in [.01, .3]$ and number rounds $\in [25, 150]$. Since the goal is to compare each metric's ability to guide search, we use both the AUROC and the AUPRC to select a model on each iteration and then compare performance on the test set.

The experiment was repeated a total of 200 times. Of those 200, the AUROC and the AUROC agreed exactly half of the time on the best model. Of the remaining 100 disagreements, using the AUROC to select the best model lead to better AUROC test set performance on 70 of the trials and better AUPRC performance on 68 of the trials.[2]

Table 3 summarizes the average AUROC and AUPRC scores when using the ROC versus PRC to tune the model. On average using the AUROC for model choice gives a slight average boost to both the test AUROC and AUPRC. This seems to be driven by a larger degradation in performance for the AUPRC relative to trials where the two metrics agree, as can be gleaned by comparing the first and second rows.

Table 3 Top row is average performance for trials where ROC and PRC model choice disagrees. Disagreement occurred on 100 out of 200 total trials.

agree	tune ROC, test ROC	tune PRC, test ROC	tune ROC, test PRC	tune PRC, test PRC
FALSE	0.628	0.623	0.156	0.153
TRUE	0.630	0.630	0.156	0.156

In summary, we find provisional evidence of a small but detectable effect. Using the AUROC for model tuning leads to final model performance that is on average better by about .005 AUROC units and by about .003 AUPRC. This result is only for a single dataset over a constrained search space, so additional investigation over a variety of datasets and algorithms is needed before any conclusions can be made with confidence.

3 Model improvement assumptions

Recall that the two general model improvement assumptions:

1. That model improvement is a mostly homogeneous process, and
2. That we can approximate this process.

This section addresses the second assumption. The paper specifically assumes additive fixed increments where all scores in each distribution shift by a constant amount. For many classifiers we can test this assumption by observing how risk scores evolve as the algorithm learns. For example, we can monitor risk scores as the number of iterations increases for gradient boosted trees or neural networks.

This section again uses the diabetes readmissions data and the XGBoost model. The model was trained over 25, 50, and 75 iterations with the same random seed, allowing us to observe risk scores evolve over their common trajectories. The cross-validation AUROC estimates respectively are 0.641, 0.651, and 0.653, indicating continued improvement as the iterations increased.

[2] If we want the experiment to have the most verisimilitude to an actual application then we would estimate resolving power for each iteration to determine which we would have selected for model search. But given the Table 1 findings, we are probably safe to assume that the AUROC would be chosen each time.

Figure 1 shows that the positive and negative class risk scores become more dispersed as the model learns. We also see that the median risk score decreases for both classes. Our interest is in the mechanism that drives this change.

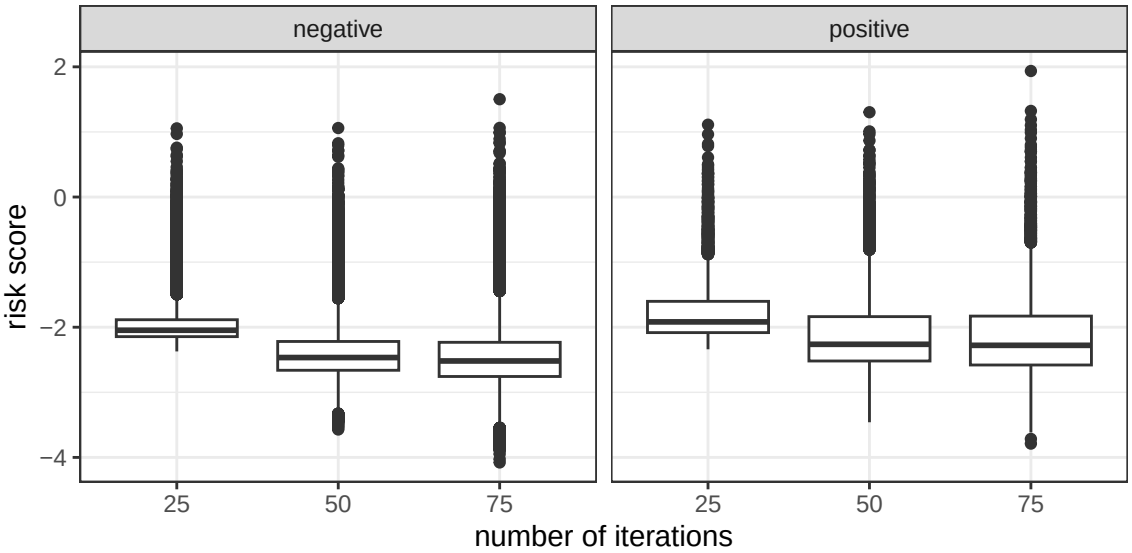

Fig. 1 Boxplots of negative and positive class risk scores over increasing number of boosting iterations

We can assess the improvement mechanism by exploring how risk scores change across different locations in the class distributions. If the additive improvement assumption holds, then all scores should change by a similar amount regardless of their location in the distributions. Figure 2 plots how each risk score changes from the 25 to 50 iteration model (first_change) and from the 50 to 75 iteration model (second_change).

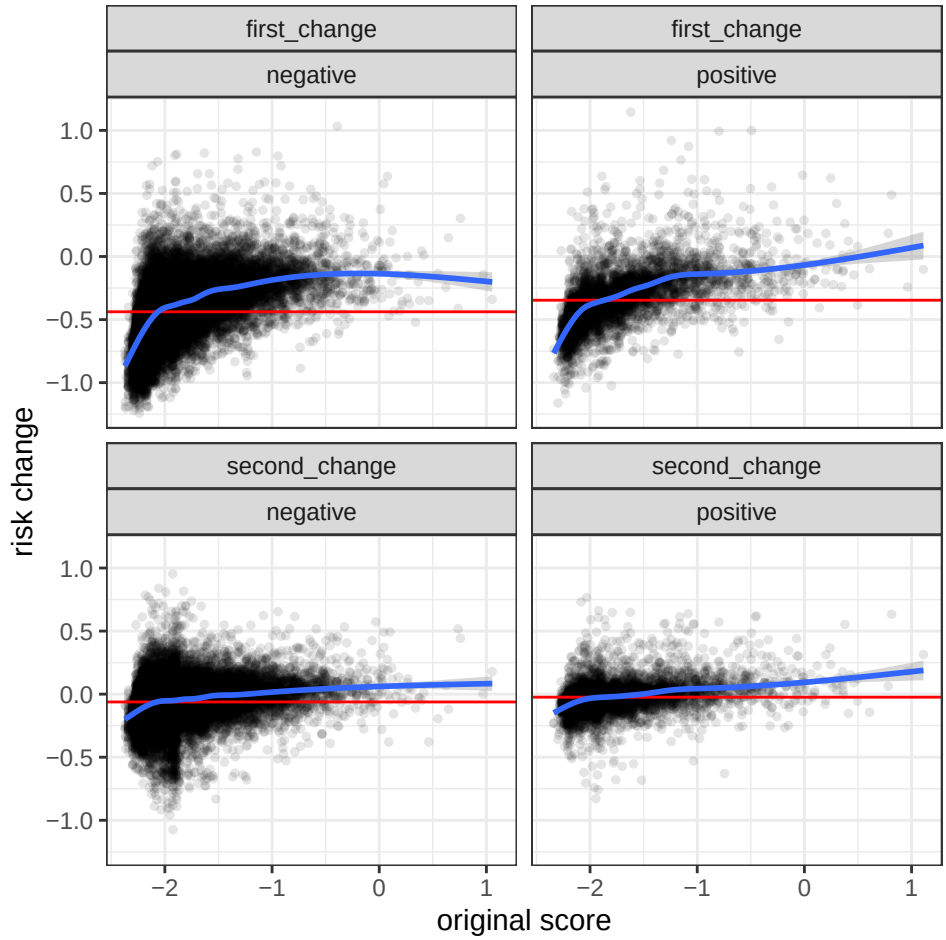

Fig. 2 Risk change versus original score for the positive and negative classes. Average change is shown by the horizontal red lines while blue lines are fit by a cubic spline.

The red line shows the average change while the blue lines show a cubic spline fit. If the additive increment assumption was approximately true, risk score changes would be tightly clustered around the average change, with no trend across distribution location. What we see in actuality is larger negative changes in the lower tails of the distributions. It appears as if improvement occurs by predicting the negative class with more confidence. Thus, there is some evidence that the additive improvement assumption is violated. Using the above data, we could also formulate various statistical tests of the additive improvement assumption.

If we are especially concerned about the veracity of the improvement mechanism, we could just use the observed changes in risk scores to model classifier improvement. This empirical path can then be used to estimate the signal curve.

Finally, we might extend this type of analysis to also test the "homogeneous improvement" assumption. We can observe how improvement proceeds under various hyperparamter settings to see if different model types change the risk scores in reliably different ways.

References

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794

Hand DJ (2006) Classifier Technology and the Illusion of Progress. Statistical Science 21(1):1 – 14. https://doi.org/10.1214/088342306000000060, URL https://doi.org/10.1214/088342306000000060

Hand DJ (2009) Measuring classifier performance: a coherent alternative to the area under the roc curve. Machine learning 77(1):103–123

Hand DJ, Anagnostopoulos C (2023) Notes on the h-measure of classifier performance. Advances in Data Analysis and Classification 17(1):109–124

Kohavi R, et al (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, Montreal, Canada, pp 1137–1145