A Appendix: Polytomous IRT models for algorithm evaluation using AIRT

We discuss the results of the polytomous AIRT approach here. First, we discretize the data using quantiles.

A.0.1 Continuous to polytomous

We consider contiguous intervals to transform continuous performance metrics to polytomous responses. Let us denote the original performance of test instance i for algorithm j by y_{ij}. First we transform the performance results such that higher values indicate better performance levels. If large values of the original performance metric indicate worse performances, then we transform it using $z_{ij} = \max_{k,l} y_{kl} - y_{ij}$, otherwise we let $z_{ij} = y_{ij}$. Then we compute the 20, 40, 60 and 80 percentiles of z_{ij} along with the minimum and maximum and denote these values increasing order by a_0, a_1, a_2, a_3, a_4 to a_5.

Thus we have performance values z_{ij} increasing with good performance and contiguous intervals $[a_0, a_1)$, $[a_1, a_2)$, $[a_2, a_3)$, $[a_3, a_4)$ and $[a_4, a_5]$ covering all performance values. Then the polytomous response x_{ij} is computed as

$$x_{ij} = \begin{cases} \text{P1 , if} & z_{ij} \in [a_0, a_1) \\ \text{P2 , if} & z_{ij} \in [a_1, a_2) \\ \text{P3 , if} & z_{ij} \in [a_2, a_3) \\ \text{P4 , if} & z_{ij} \in [a_3, a_4) \\ \text{P5 , if} & z_{ij} \in [a_4, a_5] \end{cases}, \tag{9}$$

giving us P1 < P2 < P3 < P4 < P5. Similarly, a different number of polytomous responses can be computed.

After transforming the performance data, we fit a polytomous IRT model and reinterpret it using the AIRT framework.

A.1 Classification algorithms

We use the classification algorithm dataset used in Section 4.1, which has the performance of 10 classification algorithms on 235 test instances investigated by Muñoz et al. (2018). To convert performance values to polytomous data such that higher values represent higher accuracies, we consider 1 − error rate as the performance metric and map values based on quantiles as discussed in equation (9).

Figure 29 shows the IRT trace lines for the 10 classification algorithms. The parameter θ increases with dataset easiness and the performance levels increase from P1 to P5. We see that there are no anomalous algorithms in the portfolio. Table 6 gives the AIRT evaluation metrics of the algorithms. To calculate the MSE, a change between two successive performance levels, such as P1 and P2 is taken as 1. Of the algorithms in the portfolio, QDA is the most stable algorithm followed by LDA, Naïve Bayes and Random Forest. Indeed, we see that the trace lines for QDA, RF, LDA and Naïve Bayes have smoother transitions compared to the others. The least stable algorithm is RBF_SVM followed by L_SVM, KNN, J48, and CART. QDA is the least effective

algorithm as seen by Figure 30. The Random Forests has the lowest easiness threshold signifying that it finds many datasets easy.

Table 6: MSE, AUCDF, Area Under Actual Effectiveness Curve (AUAEC) and Predicted Effectiveness Curves (AUPEC), Stability and Easiness thresholds for classification algorithms.

Algorithm	MSE	AUCDF	AUAEC	AUPEC	Stability	Easiness
Naïve Bayes	0.9064	0.9069	0.5543	0.5569	0.7433	1.1306
LDA	1.0638	0.9043	0.5596	0.5612	0.8012	1.1130
QDA	2.4213	0.8122	0.3644	0.1782	2.1019	2.1964
CART	0.4979	0.9532	0.6404	0.6532	0.2651	0.7953
J48	0.3745	0.9559	0.6702	0.6739	0.2580	0.6198
KNN	0.4170	0.9617	0.6644	0.6707	0.2308	0.7595
L_SVM	0.3830	0.9633	0.6537	0.6676	0.2185	0.6510
poly_SVM	0.7106	0.9282	0.5128	0.5149	0.4558	1.1447
RBF_SVM	0.2766	0.9803	0.6606	0.6670	0.1204	0.7949
Random Forest	0.9702	0.9223	0.6676	0.6936	0.6478	0.0751

A.2 Outlier detection algorithms

We use the outlier detection algorithm dataset used in Section 4.2, which has the performance of 8 outlier detection algorithms on 3142 datasets used in Kandanaarachchi et al. (2019). The performance metric used in the study is the area under the Receiver Operator Characteristic (ROC) curve, which increases with better performance. We use the quantile transformation detailed in Section A.0.1 to transform the data to polytomous.

Figure 31 shows the IRT trace lines for these algorithms. We see that KDEOS_Median_IQR and KDEOS_Min_Max are both anomalous algorithms. This can be seen from the placement of the curves for each algorithm. For the other algorithms P5 is the most probable score for high values of θ, but for KDEOS it is the most probable score for low values of θ. Similarly, all other curves are reversed for both KDEOS algorithms. Table 7 gives AIRT evaluation metrics for the outlier detection algorithms and Figure 32 gives the associated curves. From Table 7 we see that Ensemble_Median_IQR is also anomalous. However, it is also the most stable algorithm in the portfolio. From the tracelines for Ensemble_Median_IQR we see that it does not discriminate well between different values of θ, and as such it is weakly anomalous. In terms of stability, Ensemble_Median_IQR and FAST_ABOD_Min_Max are the most stable algorithms in the portfolio, followed by LOF_Min_Max and iForest_Median_IQR. In terms of effectiveness, FAST_ABOD_Min_Max is the best performing algorithm, followed by a cluster of algorithms including KNN_Median_IQR, iForest_Median_IQR and Ensemble_Median_IQR. The two KDEOS algorithms are the least effective algorithms.

A.3 Graph coloring algorithms

We use the graph coloring algorithms data used in Section 4.3 from Smith-Miles et al. (2014) for this analysis and make the data polytomous as before. For this data the 60 and 80 percentiles are the same giving us 4 unique

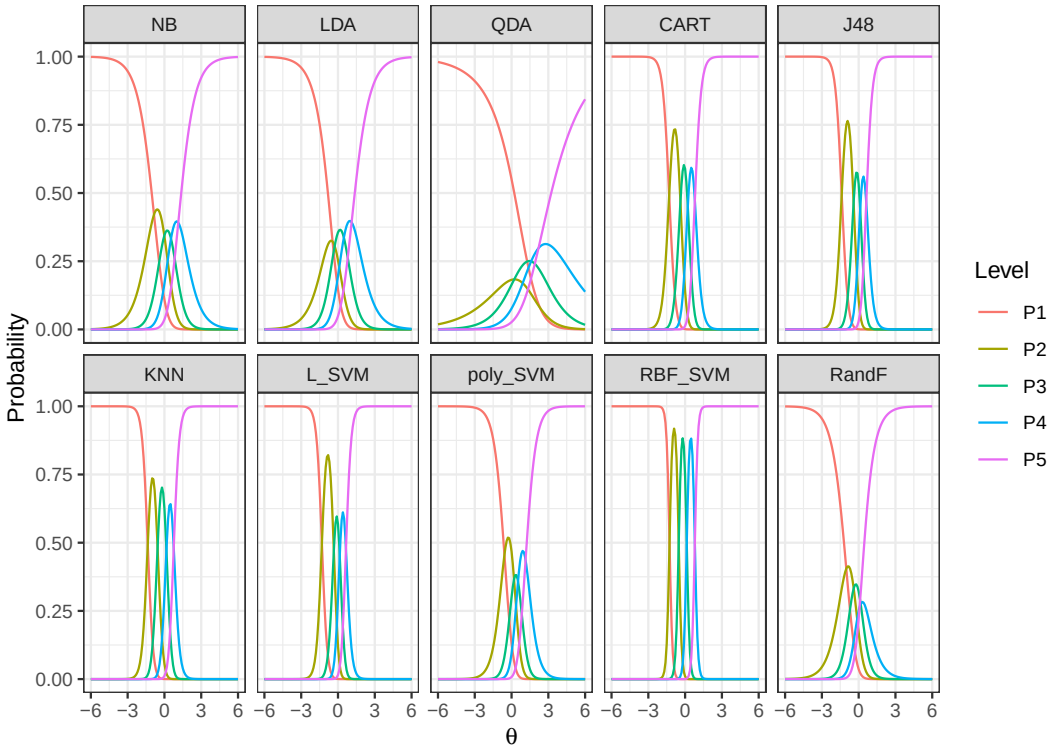

Figure 29: *IRT trace lines for 10 classification algorithms. The performance levels satisfy $P1 < P2 < P3 < P4 < P5$ with increasing θ denoting easier test instances/datasets.*

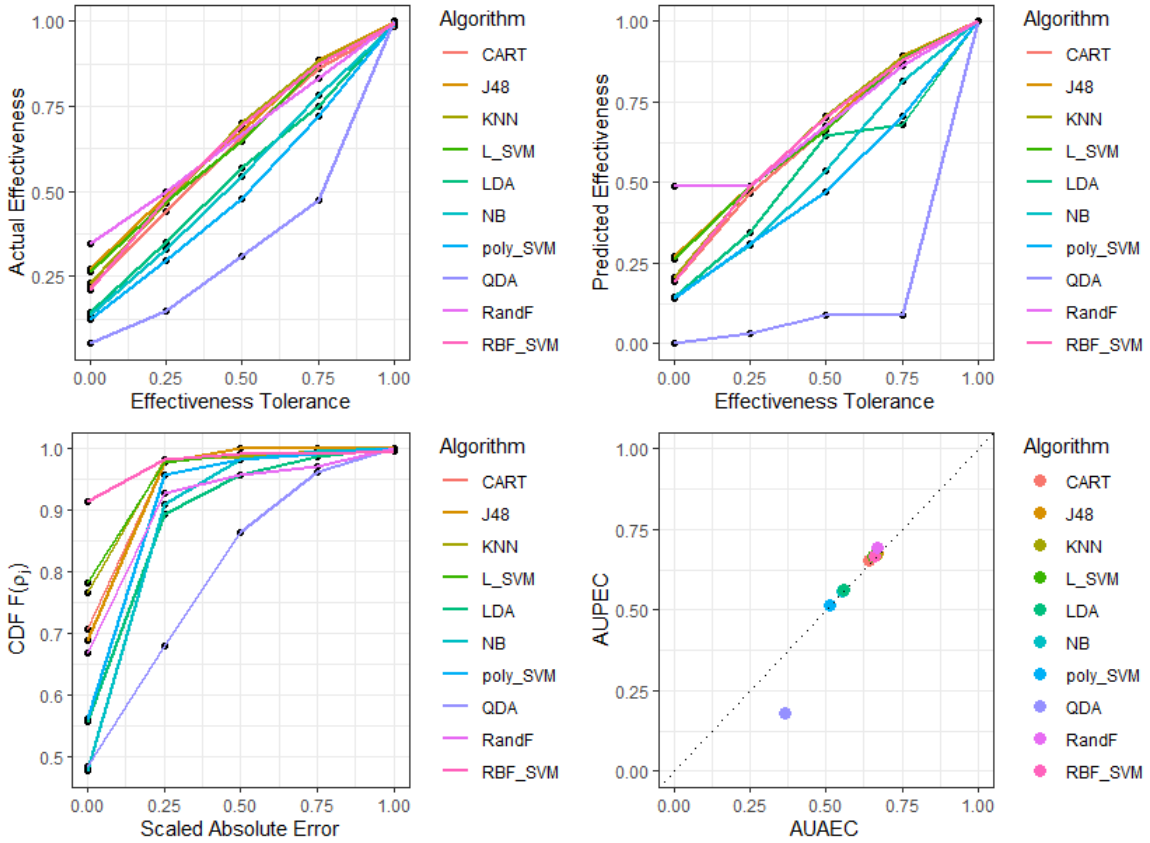

Figure 30: *Classification algorithm evaluation metrics for polytomous IRT: the actual and predicted effectiveness curves on the top row and the CDFs $F(\rho_j)$ and (AUAEC, AUPEC) on the bottom row.*

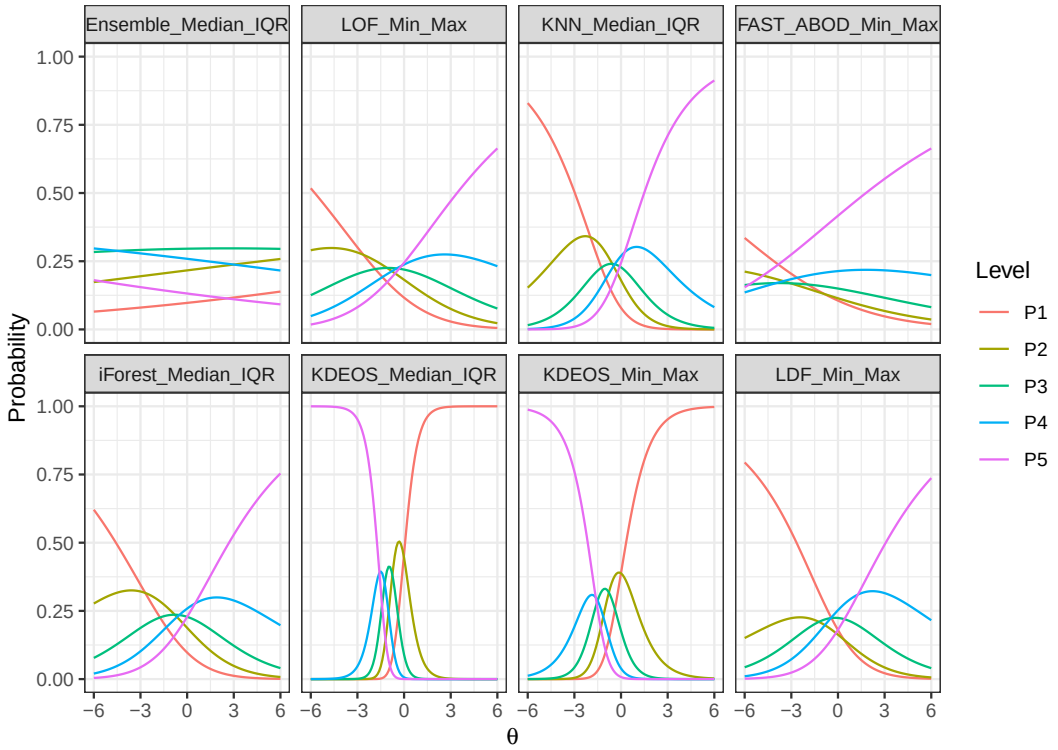

Figure 31: *IRT trace lines for 8 outlier detection-normalization algorithms. The performance levels satisfy $P1 < P2 < P3 < P4 < P5$ with increasing θ denoting easier test instances/datasets.*

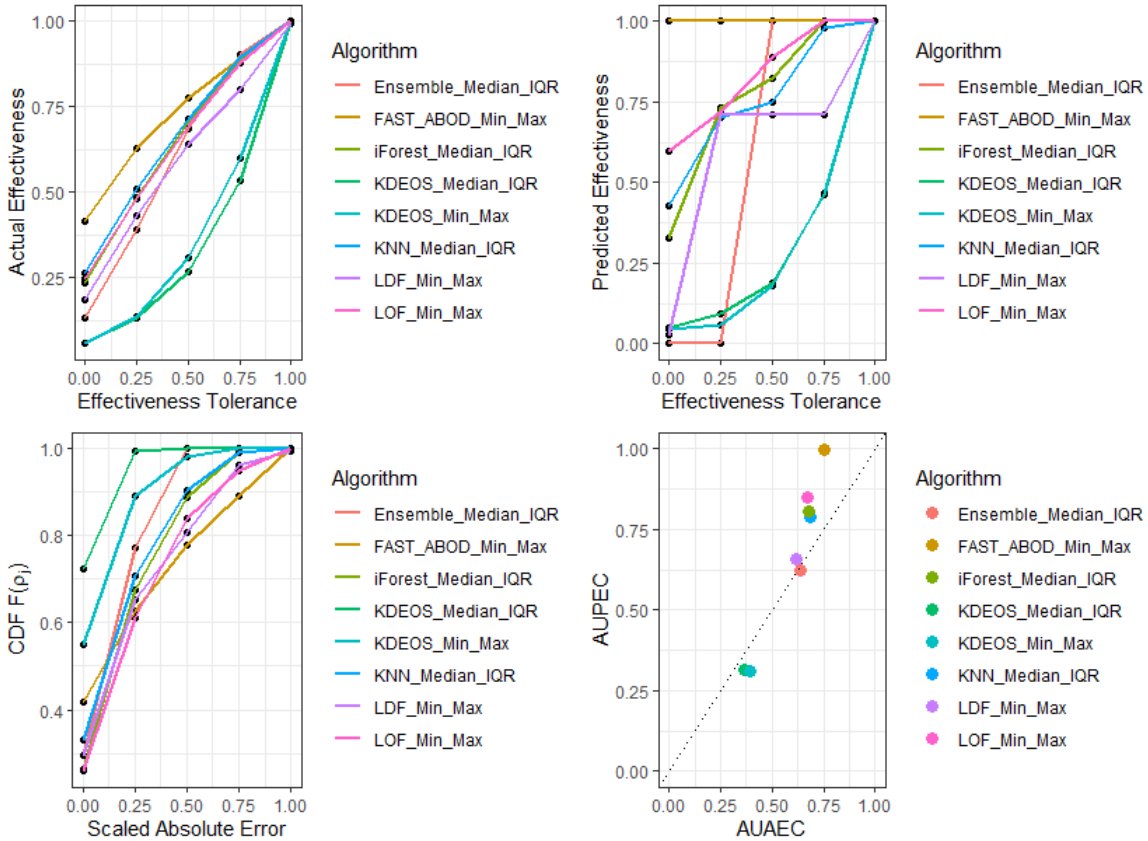

Figure 32: *Outlier detection algorithm evaluation metrics for polytomous IRT: the actual and predicted effectiveness curves on the top row and the CDFs $F(\rho_j)$ and (AUAEC, AUPEC) on the bottom row.*

Table 7: MSE, AUCDF, Area Under Actual Effectiveness Curve (AUAEC) and Predicted Effectiveness Curves (AUPEC), Stability, Anomalous indicator and Easiness Threshold for outlier detection algorithms.

Algorithm	MSE	AUCDF	AUAEC	AUPEC	Stability	Anomalous	Easiness
Ensemble_Median_IQR	1.3918	0.8547	0.6363	0.6250	33.5721	1	-22.6970
LOF_Min_Max	3.0681	0.7574	0.6695	0.8523	5.8531	0	-0.1602
KNN_Median_IQR	2.0875	0.8172	0.6869	0.7850	2.3867	0	0.2356
FAST_ABOD_Min_Max	3.5656	0.7516	0.7516	1.0000	11.1799	0	-7.4587
iForest_Median_IQR	2.3278	0.7970	0.6750	0.8045	4.0946	0	0.5117
KDEOS_Median_IQR	0.2979	0.9639	0.3648	0.3167	0.4694	1	-1.5858
KDEOS_Min_Max	0.8638	0.9122	0.3933	0.3068	0.9960	1	-1.6261
LDF_Min_Max	3.0204	0.7668	0.6152	0.6601	3.8393	0	1.2835

response values denoted by P1 to P4.

Figure 33 gives the IRT trace lines for the 8 graph coloring algorithms. We see that none of the algorithms are anomalous and for all algorithms the probability of achieving a performance of P3 never peaks. Table 8 gives the AIRT metrics and Figure34 shows associated curves. In terms of effectiveness, algorithms in this portfolio display a large variance and are placed along the line AUAEC = AUPEC starting as low as $(0.25, 0.21)$ and ending up at $(0.97, 1.00)$. The most effective algorithm is HEA, followed by TabuCol, PartialCol and AntCol. The least effective algorithm is RandomGreedy. In terms of stability, AntCol and HEA are the top two algorithms having stability values of 1.9658 and 1.9568 respectively. HEA also has the lowest easiness threshold, signifying it is a good algorithm. The trace lines for HEA and AntCol have much smoother transitions compared to the rest, while a P4 result becomes mostly likely for lower values of θ for HEA compared to AntCol resulting in a higher AUPEC for HEA. The least stable algorithm is DSATUR followed by Bktr, as can be seen from Table 8 and the sharp transitions in trace lines.

Table 8: MSE, AUCDF, Area Under Actual Effectiveness Curve (AUAEC) and Predicted Effectiveness Curves (AUPEC), Stability and Easiness threshold for graph coloring algorithms.

Algorithm	MSE	AUCDF	AUAEC	AUPEC	Stability	Easiness
RandomGreedy	0.1675	0.9746	0.2583	0.2122	0.6565	0.6486
DSATUR	0.1447	0.9762	0.5107	0.4929	0.2485	0.0827
Bktr	0.1193	0.9818	0.6408	0.6423	0.2775	-0.3947
HillClimber	0.1643	0.9788	0.7651	0.7521	0.3445	-0.9014
HEA	0.1985	0.9732	0.9732	1.0000	1.9568	-5.6786
PartialCol	0.3911	0.9446	0.9161	0.9385	0.6024	-1.1252
TabuCol	0.4079	0.9464	0.9337	0.9869	1.1136	-2.3020
AntCol	1.2673	0.8438	0.8155	0.8831	1.9658	-2.6928

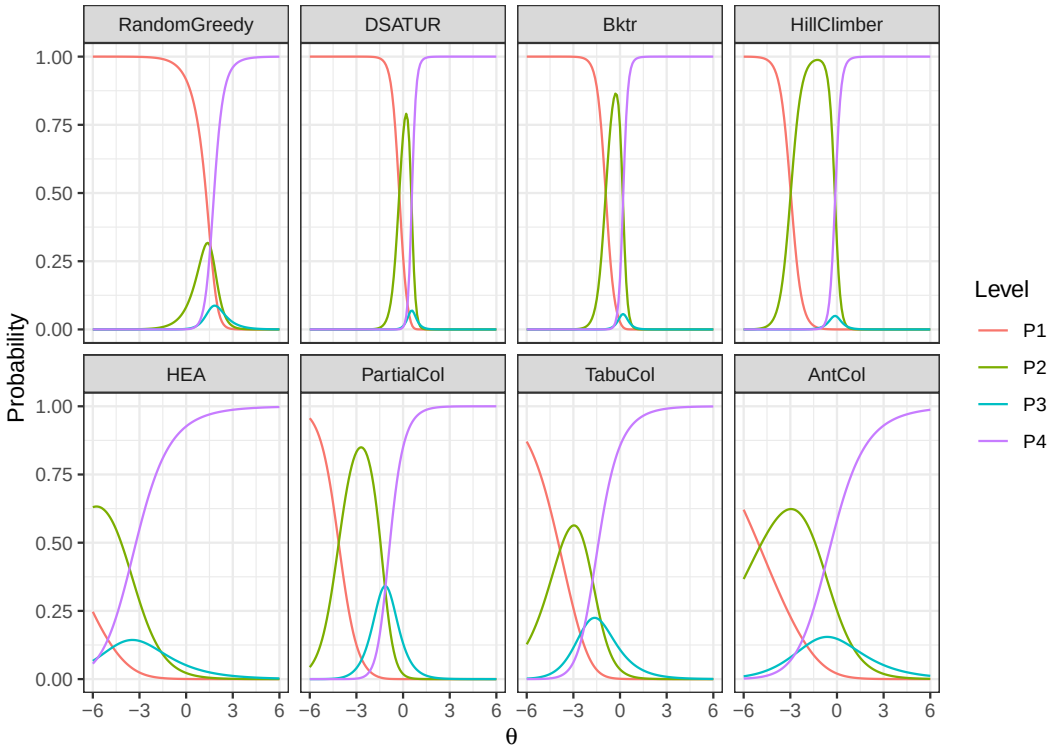

Figure 33: *Trace lines for 8 graph coloring algorithms. The performance levels satisfy $P1 < P2 < P3 < P4$ with increasing θ denoting easier test instances/datasets.*

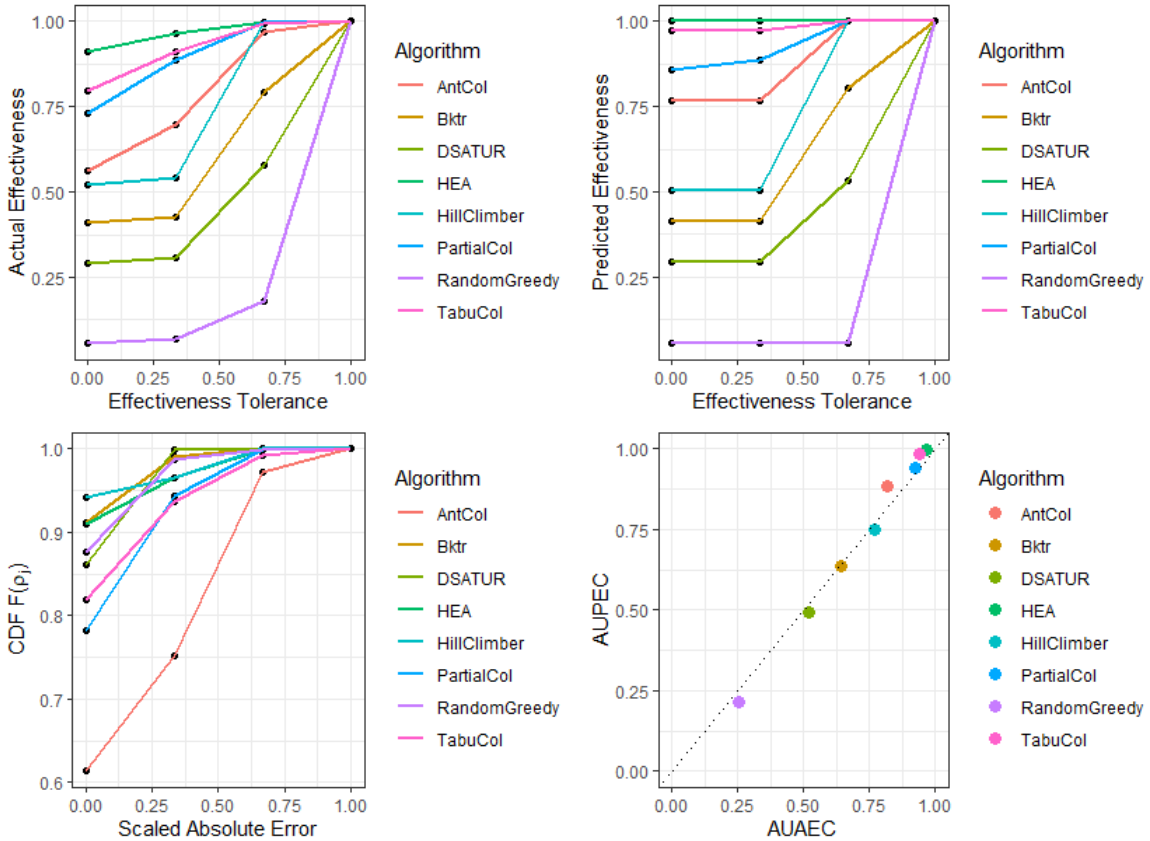

Figure 34: *Graph coloring algorithm evaluation metrics for polytomous IRT: the actual and predicted effectiveness curves on the top row and the CDFs $F(\rho_j)$ and (AUAEC, AUPEC) on the bottom row.*