An Examination of Metrics that Describe User Models

Eric G. Van Inwegen Yan Wang Seth Adjei Neil Heffernan
100 Institute Rd
Worcester, MA, 01609-2280
+1-508-831-5569
{egvaninwegen, ywang14, saadjei, nth} @wpi.edu

ABSTRACT

We hypothesize that there are two basic ways that a user model can perform better than another: 1.) having test data averages that match the prediction values[1] (we call this the *coherence* of the model) and 2.) having fewer instances near the mean prediction (we call this the *differentiation* of the model). There are several common metrics used to determine the goodness of user models, which include: AUC, RMSE, and R-squared. These metrics conflate coherence and differentiation, which can sometimes lead to confusion, especially if metrics don't agree. By using synthetic and real data, we demonstrate how six different metrics respond to changes in coherence and differentiation. We believe that user model analyses will be improved if authors report the coherence and differentiation, as well as to include AUC/A', RMSE and R^2. Lastly, we share a simplified spreadsheet that enables readers to examine these effects on their own datasets and models.

1. INTRODUCTION

One of the goals of many in the online educational community is to more accurately predict whether a student will get the next question correct. If an algorithm can accurately predict that a student will get the next problem correct, an Intelligent Tutoring System could prevent a student from doing more work than is necessary, or continue giving students work until they have mastered a topic. In order to predict student responses, algorithms such as Knowledge Tracing [2], Performance Factors Analysis [11], and tabling methods [17] etc. have been developed. (See [3] for a thorough review of various user models.) Looking at only papers presented at EDM 2014, we find xx new models or modifications proposed [14]. When new models are presented, they are often compared to the results of old models. Common metrics used to determine when a model is better than another include AUC/A', RMSE, MAE, and R-squared. There has been some work done (e.g. [1]) looking into what sort of range of values we should expect from various metrics (as well as how well can the models do to rediscover known parameters).

An ideal user model would perfectly predict student responses. Current models predict the probability that a student-problem-instance (hereafter "instance") will be correct. Models such as Knowledge-Tracing ("KT"), Performance Factors Analysis ("PFA"), and their derivatives create a theoretically continuous range of predictions from 0.00 to 1.00. Tabling models (eg. [17]) may predict a finite range of numbers, but many have been modified with a regression and thus create a continuous (or near-continuous) range of values. However, even the "continuous" models can only create a maximum of number of predictions (one per instance). This means that even continuous models have only a finite number of predictions, often with multiple instances sharing the same prediction value.

There are two basic properties of a model that will make it more accurate: 1.) How well a prediction matches the aggregate test-data for that value, and 2.) How well the model (by incorporating more / better features) can make predictions away from the mean and closer to 0 or 1. We refer to these two concepts as coherence and differentiation, respectively.

If we look at the most naive method to predict correctness on the next problem, we can imagine that it would be to either predict majority class (1, if more than 1/2 of students are right throughout the training data, 0 otherwise) or to predict the average correctness value from the training data[2]. The ideal goal would be to perfectly predict ones and zeros; this would give two prediction values. For example, if we examined a data set and determined that 75% of the students get the next question correct, then the naive mean model would predict 0.75 for all instances, while the absolute ideal model would (correctly) predict 0.00 for 1/4 of the instances and 1.00 for the other 3/4.

The state of knowledge modeling is clearly somewhere in the middle. E.g. if a dataset has an average of 0.75, and a model predicts 0.80 for a particular instance, that model "thinks" that that student at that time on that problem is more likely to be right than the average. In order to better understand the significance of the predictions that models make, we find that we need to start examining two (currently unreported) properties of predictions.

1.1 "Coherence"

Given a large enough data-set, we argue that an accurate model's predictions should match the test data average for a given group of instances. For example, if a model were to identify a group of instances and give that group a predicted value of 0.25, we argue that the model is accurate if exactly one out of every four students (on average) gets the correct answer. If the model predicts 0.25, but only one out of every ten gets it right, the model's "scores" by most metrics will be improved, however, it is not as accurate as a similar model that groups that same instances together, but predicts 0.10.

1.2 "Differentiation"

A naive model of student knowledge might just use the average score from a training dataset and make a prediction of that probability for all instances in the test data. Arguably, more complicated user models seek to find reasons *not* to do this. The more features that a model can incorporate to move predictions away from the mean value, the better a model is at claiming that *this* instance of student interaction is likely to be right while *that* instance is likely to be wrong. By manipulating features, models (hopefully) learn how to make reliable predictions that are different from the mean. We use the term "differentiation" in much the same way as "distribution". We do so to avoid a possible confusion with the idea of the distribution of the training

[1] E.g. If a model identifies a group of 100 student-problem-instances (based on the model's parameters) and gives that group a prediction value of 0.75, the test data average matches the prediction if 75 instances are correct.

[2] E.g. A model that predicts 1.00 when the training data average is 0.75 will score an RMSE of 0.5, if the test data has the same average. Using 0.75 as the prediction results in an RMSE of 0.43.

data. It may be helpful to the reader to think of "differentiation" as the distribution of the prediction values.

1.3 Common Metrics Used as Evaluators

In the educational data mining community, models are ranked against each other by using a range of metrics. Five common ones are AUC, A', R^2, RMSE, and MAE; we make the case to replace R^2 with Efron's R^2. We are not the first to suggest Efron's R^2; e.g. [9]. Nor are we the first to examine how metric scores can be misleading [18]. Some work has been done to identify the "best" metric [4]; we think (and hope to show) that any one metric can give insufficient information to "score" models.

We hope to show how these metrics may be inconsistent. All six metrics above involve a summation in their calculations. This summation (across the predictions and test data) conflates coherence and differentiation; this conflation can make it difficult to compare models if one is better at coherence, and the other is better at differentiation. In this paper, we seek to illustrate how these metrics respond to changes in coherence and differentiation by analyzing the results of synthetic and real data and models.

1.4 Essential Questions

One of the goals of this analysis is to demonstrate the need to have a more in-depth analysis of user models and look "under the hood" at why and where they do, or do not, produce accurate predictions. To improve the user models, we should be looking at coherence and differentiation as separate properties of models. Doing so may allow us to find areas where one model has weaknesses that can be improved upon; it may also be possible to selectively ensemble.

1.) How do commonly used metrics compare when model-test data interactions vary along only single elements at a time? E.g. how does modifying models' differentiation ability, while keeping coherences identical (and vice versa) affect metric scores?

2.) When do metrics fail to agree?

3.) Is there a more useful metric or combination of metrics that lets the user modelling community identify the particular strengths and weaknesses of models?

2. BACKGROUND

The sub-sections that follow are designed to take novice readers through the metric calculations. The more mathematically-fluent may wish to skip to the methods and results.

2.1 Area Under the Curve and A-Prime

The metric commonly referred to as AUC refers to the area under the receiver operator characteristic (ROC) curve; this metric has been frequently used and described in a variety of articles [6, 7, 10, 12, to name a few]. The ROC curve is often used in medical studies to differentiate tests that are useful for ruling in vs. tests that are useful for ruling *out*. Tests that are useful for ruling in will have a low false positive rate (FPR) (i.e. few of the times that the test states that a person has a condition will actually be wrong). Tests that are useful for ruling out will have a low false negative rate (FNR) (i.e. few of the times that the test "clears" a patient will be wrong). The ideal test would have both a low FPR and FNR; this test does not always exist. The ROC curve is made up of a plot of true positive rates (TPR) to false positive rates. An AUC of 0.500 means that the tests (or model) have the predictive power of a coin toss; an AUC of 1.000 means that the tests (or model) are accurate 100% of the time.

When used to measure user models, the ROC curve values of TPR are found by finding the fraction of positive values (correct scores) are found above any given model prediction; the highest prediction value always score a TPR = 0. The FPR is found as 1-True Negative Rate (TNR). TNR is the fraction of incorrect scores *below* a given value; the highest prediction value has a TNR = 1; therefore the highest prediction value has an FPR of 0. AUC is the area under this curve (found by finding the trapezoidal area under two consecutive points). A' is determined by a different method; it can be thought of as the probability of a randomly selected negative result being ranked lower than a randomly selected positive result. [7] With ideal methods of calculation, AUC and A' should have the same value.

What AUC and A' fail to measure is the coherence of the model to test data. AUC and A' measure how well a model does at sorting test data, with all of the negatives (i.e. 0's in EDM) ranked below the positives (1's in EDM). The following two models would score a 1.00 (i.e. "perfect") in A' / AUC, as well as in R^2.

Table 1: Simple demonstration model

Model 1		Model 2	
Prediction	Correct	Prediction	Correct
0.01	0	0.98	0
0.01	0	0.98	0
0.99	1	0.99	1
0.99	1	0.99	1

In terms of our language, both models have been able to perfectly differentiate the instances; whatever features these models use to separate student responses, those features have correctly separated the students who got it right from the students who got it wrong. On the other hand, the algorithms that the models have used to predict the chances of success are drastically different. The coherence of the model 1 to the test data is very good, while model 2's coherence is almost entirely wrong.

2.2 R^2 vs. Efron's R^2

R^2 can be thought of as a way to capture how much of the variation of a sample is contained within the explanation. Calculated as the square of Pearson's r, it can only have values from 0 (no fit between explanation and data) and 1 (perfect match). (E.g. see page p187 of [15].) Efron [5] modifies R^2 to compare the error of a model to that of a naive model (predicting the mean of the data). The power of Efron's R^2 is that models that are well ordered[3], but have low coherence between the predictions and the actual values (e.g. Model 2 in Table 1) will have lower scores with Efron's R^2; traditional R^2 only relies on the ordering ability. The use of Efron's R^2 as a metric has been done in user modeling. (E.g. see [9].) Although its use is not widespread, we think that the demonstrations that follow will show that it differentiates models better than traditional R^2. For example, in the models of Table 1, the R^2 for both is 1.00, while Efron's R^2 is 0.9996 and -0.9210 for Models 1 and 2, respectively. A negative score indicates that the model is has more error than simply predicting the mean value.

3. METHODS

In order to visualize the impact of differentiation and coherence on the various metrics, we generate not synthetic data, but rather synthetic model outputs. That is to say that we are not concerned

[3] By "well ordered", we mean that negative values (incorrect student responses) are ranked below positive values (correct student responses). AUC and A' predominantly score models' abilities to rank positives and negatives. [12]

with *how* a given model creates its predictions, we are only exploring a variety of models' predictions and test-data averages to examine the impact on the scoring metrics. In order to make the calculations (somewhat) easily replicable, we examine simplistic models that make only 11 different predictions.

To compare model outputs, a spreadsheet was created that allows the user to input prediction value, test group average, and number of instances within that group, for up to eleven groups[4]. The spreadsheet then calculates values for AUC, A', R^2, Efron's R^2, RMSE, and MAE. The benefit of doing all of this in a spreadsheet is that the user can see how the calculations are done. The benefit of using tabled data, as opposed to individual data lines, is that it saves space and is easier for the researcher to change to compare the effects of coherence and differentiation. A publicly shared copy of the spreadsheet can be found at: http://tinyurl.com/kznthk7.

To test the sensitivities of the metrics to differentiation and coherence, a baseline model-test data interaction must be defined. Our simplest incarnation is a model with 11 prediction values (0.00, 0.10... 0.90, 1.00), equal numbers of instances in each prediction bin (100), and a perfect coherence. This model-test interaction is called the "Ideal-Flat". In most of our synthetic comparisons, the Ideal-Flat is present to give us a way to compare one trend to another. The "Flat-Ideal" is characterized in Table 2.

Table 2: Characteristics of the "Ideal-Flat" model-test data interaction. The colors are used consistently to refer to prediction values, test-data averages, and numbers of instances.

Prediction Values	Test Data Averages	Numbers of Instances
0.00	0.00	100
0.10	0.10	100
0.20	0.20	100
0.30	0.30	100
0.40	0.40	100
0.50	0.50	100
0.60	0.60	100
0.70	0.70	100
0.80	0.80	100
0.90	0.90	100
1.00	1.00	100
AUC: 0.8636	A': 0.8636	RMSE: 0.3873
R^2: 0.4000	Ef-R^2: 0.4000	MAE: 0.3000

3.1 Impact of Differentiation

To test how the metrics change based on differentiation, a few basic patterns are explored.

1.) The basic impact of concavity (using linearly changing differentiations that are either maximum or minimum at 0.50)

2.) The effect of changing concavity on a skewed differentiation that more resembles real data.

3.) The effect of the average prediction value on the metrics.

4.) The effect of a differentiation with a large number of predictions at only one extreme.

3.2 Impact of Coherence

Although there are a potentially infinite number of ways that a model and the aggregate test data can lack coherence, we only test two. We test the effects of:

1.) Maintaining the prediction values, but symmetrically changing the test group averages

2.) Maintaining the differentiation, but symmetrically changing the prediction values.

4. RESULTS AND DISCUSSION

4.1 Impact of Differentiation

We should point out that, in order to make 1 a "perfect" score for all metrics, we are using 1-RMSE and 1-MAE. (Ordinarily, a score of 0 is "perfect" for error analysis; however, to show how the metric scores change, it is useful to have all scores "point" in the same direction.) We use this convention in all of our metric plots. In the metric plots, the horizontal axes represent the different models' outputs in each comparison.

4.1.1 Basic Concavity Effects of Differentiation

The first test is to determine how the concavity of the Differentiation impacts the scores of the metrics. Figure 1 is a plot of the six metrics as a differentiation changes from an exceptionally steep "V" (10,100 at 0.00 and 1.00 and 100 at 0.50, changing linearly between) to flat to increasingly steep "A" (100 at 0.00 and 1.00 and 10,100 at 0.50). The maximum values of the "V's" and "A's" are (10,100, 5,100, 2,600, 1,100, 600, 350).

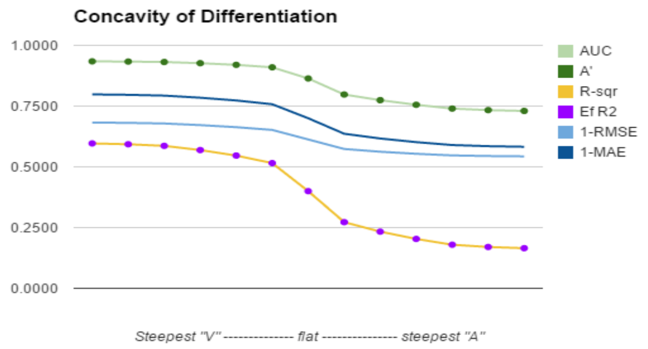

Figure 1: AUC, A', R^2, Efron's R^2, (1-RMSE), and (1-MAE) for 13 synthetic models that differ only in the patterns of differentiations. The model on the far left has 10,100 instances for prediction values 0.00 and 1.00, 100 instances at 0.50, and changes linearly between. The model on the far right has the reverse trend (10,100 at 0.50 and 100 at 0.00 and 1.00). All models in this comparison have test data averages that match the prediction values.

From Figure 1, we can see that all six metrics are most responsive nearest the Flat Ideal; the largest change in metric scores occurs nearest to the change in concavity of the differentiation. One way to interpret this is that small gains in these metrics will be made until a given model can start to truly differentiate; i.e. have fewer predictions near the mean and more near 0.00 and 1.00.

4.1.2 Changing Concavity on only part of the Differentiation

Examining a plot of the differentiation of Knowledge Tracing, Performance Factors Analysis, and a new model being presented elsewhere by three of this paper's authors, we have found that all three have a differentiation with few instances below 0.50, a peak

[4] This should be considered a first demonstration of principle. We are not suggesting that models be limited to 11 predictions

Table 3: Differentiation patterns used in Figure 2

P	test	"_A"			"_="			"_M"
0.00	0.00	80	80	80	80	80	80	80
0.10	0.10	80	80	80	80	80	80	80
0.20	0.20	80	80	80	80	80	80	80
0.30	0.30	80	80	80	80	80	80	80
0.40	0.40	80	80	80	80	80	80	80
0.50	0.50	400	800	1200	1600	2000	2400	2800
0.60	0.60	1600	1600	1600	1600	1600	1600	1600
0.70	0.70	2800	2400	2000	1600	1200	800	400
0.80	0.80	2800	2400	2000	1600	1200	800	400
0.90	0.90	1600	1600	1600	1600	1600	1600	1600
1.00	1.00	400	800	1200	1600	2000	2400	2800

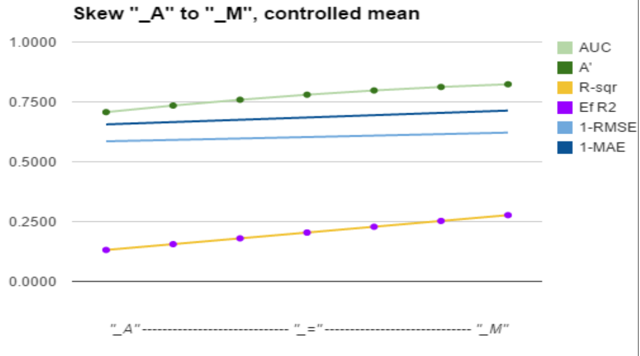

Figure 2: The effect of changing concavity on only part of a differentiation pattern.

between 0.70 and 0.80, and then a decrease down to 1.00. Table 3 shows the eight differentiation patterns used to generate the values plotted in Figure 2. For all patterns in this comparison, the test averages match the prediction values. Also controlled are the total number of predictions (10,000) and the average prediction (and test average) value (0.7280).

4.1.3 Effect of Peak Location on Metrics

If models are seen as an attempt to differentiate away from the training data mean, it would be a useful comparison to see how the metrics change for the same differentiation ability but centered on a different value. If we take the "_A" model from Table 2 and shift the peak, we can get an idea of how the average value of the training data (and thus model's predictions) affect the metrics.

The implication of this trend is that the average prediction value (which tends to follow the average training data value) impacts the score across the six metrics. Any data set that has an average close to 0.500 will naturally score poorly across the metrics.

4.1.4 Effect of a Large Number of Extreme Predictions

So far, there has been agreement in all six metrics. The following demonstration shows that error-based metrics and ordering metrics do not always agree. Figure 4 demonstrates the effect of a single-sided skew on the six metrics. The leftmost differentiation pattern is the "Flat-Ideal"; the rightmost is flat from 0.00 to 0.50, but then increases linearly to 10,100 at a prediction of 1.00. (The interim points follow the same pattern as the "V's" in Figure 1.) Another way to visualize "_/" is as a (goalie's) hockey-stick.

For the first time in this series of demonstrations, we have a disagreement between some of the metrics. The error-based metrics are improving, while the ordering & variability metrics are worsening (after a slight improvement). This can be explained by observing that the error metrics will have smaller average errors as there are more predictions near 1.00, while the ordering metrics will start to have more negatives above the mean than does the Flat Ideal.

Table 4: Differentiation patterns used in Figure 3

P	test	"_/|"	"_A"					"A_"	"|_"
0.0	0.0	80	80	80	80	80	80	400	1920
0.1	0.1	80	80	80	80	80	400	1600	2800
0.2	0.2	80	80	80	80	400	1600	2800	2800
0.3	0.3	80	80	80	400	1600	2800	2800	1600
0.4	0.4	80	80	400	1600	2800	2800	1600	400
0.5	0.5	80	400	1600	2800	2800	1600	400	80
0.6	0.6	400	1600	2800	2800	1600	400	80	80
0.7	0.7	1600	2800	2800	1600	400	80	80	80
0.8	0.8	2800	2800	1600	400	80	80	80	80
0.9	0.9	2800	1600	400	80	80	80	80	80
1.0	1.0	1920	400	80	80	80	80	80	80
	avg	0.816	0.728	0.637	0.546	0.454	0.363	0.272	0.184

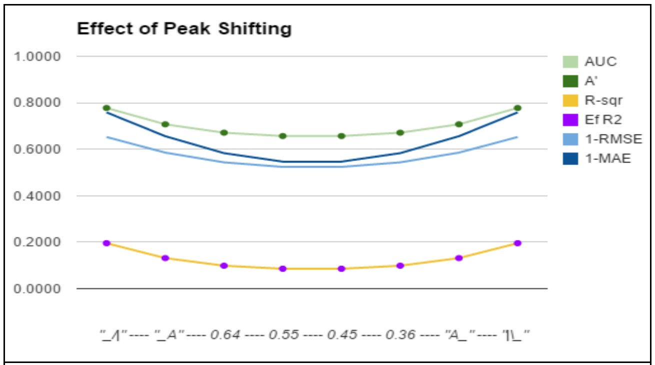

Figure 3: The effects of shifting a peak of predictions on the six metrics.

4.2 Impact of Coherence

The next comparisons will explore the effect of coherence (or, more properly, a loss of coherence) on metric scores. When there is a loss of coherence is when Efron's R^2 gives a more meaningful measure than standard R^2.

4.2.1 Five Models that achieve different test group averages

If the loss of coherence follows a symmetrical pattern, the metrics have improved score if the coherence serves to increase the ordering power. Likewise, when the mis-coherence serves to diminish the differentiation, the models and metrics all fare worse. In the next example, five models are compared; two where the

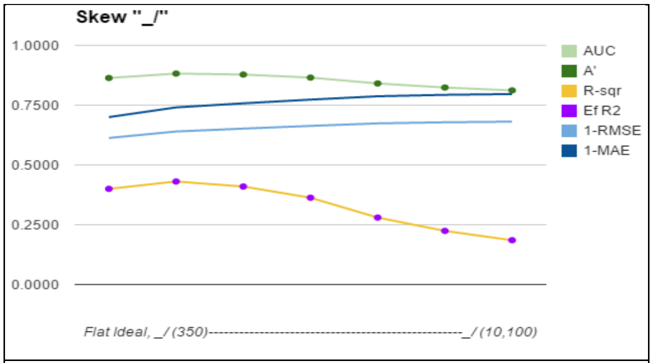

Figure 4: Demonstrating the effects of skewing predictions.

Table 5: Models A, B, C, and D. The models all predict the same (left red column), but the test data averages do not match the predictions (except for the Ideal Flat).

p - all	A - test	B - t	IF - t	C - t	D - t	n
0.00	0.00	0.00	0.00	0.15	0.25	100
0.10	0.01	0.01	0.10	0.22	0.30	100
0.20	0.02	0.13	0.20	0.29	0.35	100
0.30	0.17	0.25	0.30	0.36	0.40	100
0.40	0.33	0.38	0.40	0.43	0.45	100
0.50	0.50	0.50	0.50	0.50	0.50	100
0.60	0.67	0.68	0.60	0.57	0.55	100
0.70	0.83	0.75	0.70	0.64	0.60	100
0.80	0.98	0.88	0.80	0.71	0.65	100
0.90	0.99	0.99	0.90	0.78	0.70	100
1.00	1.00	1.00	1.00	0.85	0.75	100

model's predictions are less confident than the test data averages (models "A" and "B"), two where the prediction is overconfident (models "C" and "D"), and the ideal flat model. In this comparison, all five models make the same 11 predictions (and all have 100 instances per prediction). However, the test data averages vary from the predictions. The models' predictions and test data averages are in Table 4.

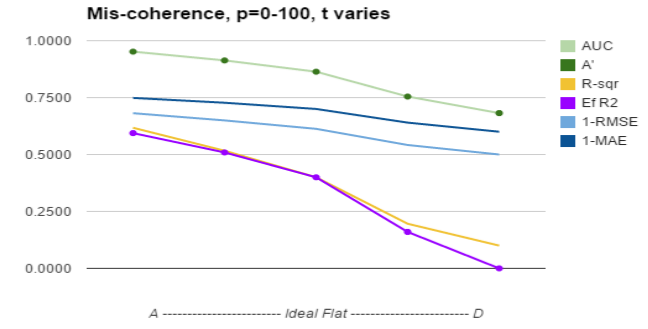

Figure 5: Five models that have the same prediction values, but different averages for the test data within the prediction groups.

4.1.3 Five Models with different prediction values

In this next exploration, the test-group average will remain at 0, 0.10... 0.90, 1.00, but the prediction values will vary. Table 3 summarizes the changes to the prediction values; Figure 4 display the metric scores for the models described in table 3.

E - p	F - p	I.F. - p	G - p	H - p	Test	n
0.25	0.15	0.00	0.00	0.00	0.00	100
0.30	0.22	0.10	0.01	0.01	0.10	100
0.35	0.29	0.20	0.13	0.02	0.20	100
0.40	0.36	0.30	0.25	0.17	0.30	100
0.45	0.43	0.40	0.38	0.33	0.40	100
0.50	0.50	0.50	0.50	0.50	0.50	100
0.55	0.57	0.60	0.62	0.66	0.60	100
0.60	0.64	0.70	0.75	0.83	0.70	100
0.65	0.71	0.80	0.88	0.98	0.80	100
0.70	0.78	0.90	0.99	0.99	0.90	100
0.75	0.85	1.00	1.00	1.00	1.00	100

Table 6: Models E-H; metric scores are in Figure 6

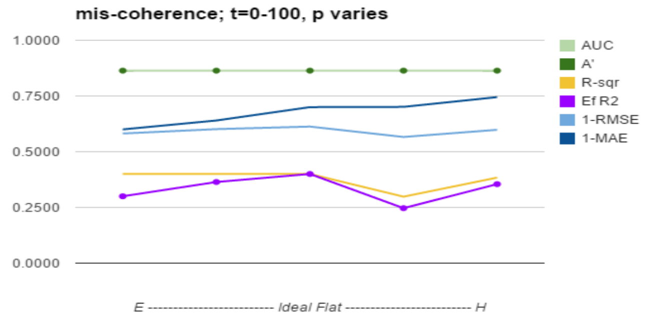

Figure 6: Five models that all achieve the same differentiation and test-data averages, but having differing prediction values. An interesting point not easily apparent in the graph is that 1-RMSE does not perfectly track R^2 for the first three values; there are times when RMSE and R^2 respond differently as error metrics.

Since the groupings, differentiation patterns and ordering of correct and incorrect percentages don't change, AUC and A' are the same across the five models. The other four metrics have a rather complex relationship with these models. What should be learned from these examples is that these metrics do not always agree, because they measure different things. As a general rule, RMSE improves when the test group average is more extreme (closer to 0 or 1) than the prediction.

4.3 Real Data Analyzed for Coherence and Differentiation

4.3.1 Example 1 - Three Models trained on a large (~400K) data set

If we take real data and models and try to analyze it in the same manner, we, of course have the problem that there are vastly more than 11 prediction values. However, as an example, we can take real data and force it to look like the synthetic data presented above.

In another paper [16], we have submitted a new user model. The mechanics of this new model are outside of the scope of this paper. However, the results of that model can be used as an example here. In that paper, the new model, called "SuperBins" (SB), is compared to Knowledge Tracing (KT) and Performance Factors Analysis (PFA), and found to be "better", according to RMSE, R^2, and AUC. The analysis of why / where one model is better than another is not included in that paper, but perhaps it should have been.

If we "shoehorn" the model-test interaction of SB, KT, and PFA into only 11 groups, we will certainly lose precision (and the metric scores suffer somewhat), but the analysis is useful. To do so, we average the prediction values (according to their frequency) across eleven equal lengths *of prediction values* of the data set; we do the same for the test data averages. I.e., The average prediction value from 0 to 0.0909, as weighted by the frequency of each prediction was found to be 0.08 for the SuperBins model. Within that range, there were 5 instances, all were wrong. There were no predictions in that range for KT. There were nine for PFA (eight were right), with an average prediction value of 0.01. The first row in Table 7 displays this information.

Table 7: Example 1: A Coherence-Frequency Table of results from three knowledge models trained and tested on the same real dataset (80/20). Model results have been averaged across 11 intervals to be comparable to the synthetic model results used elsewhere in this paper. The prediction and test values are the weighted averages of each model within the ranges on the left. * KT had no predicted values smaller than 0.15 for this dataset.

	SB					*KT*					PFA
Range	**pred**	**test**	**n**		*pred*	*test*	*n*		pred	test	n
0.0000 - 0.0909	**0.08**	**0.00**	**5**		*				0.01	0.78	9
0.0910 - 0.1818	**0.14**	**0.13**	**516**		*0.16*	*0.75*	*4*		0.13	0.53	17
0.1819 - 0.2727	**0.22**	**0.23**	**892**		*0.24*	*0.30*	*64*		0.23	0.46	56
0.2728 - 0.3636	**0.31**	**0.32**	**1829**		*0.33*	*0.28*	*704*		0.31	0.49	168
0.3637 - 0.4545	**0.41**	**0.41**	**3235**		*0.40*	*0.36*	*2565*		0.41	0.42	643
0.4546 - 0.5454	**0.50**	**0.51**	**4878**		*0.51*	*0.48*	*6978*		0.50	0.49	3539
0.5455 - 0.6363	**0.60**	**0.60**	**6355**		*0.60*	*0.61*	*8776*		0.61	0.59	7376
0.6364 - 0.7272	**0.69**	**0.69**	**9772**		*0.69*	*0.71*	*12149*		0.70	0.70	25819
0.7273 - 0.8181	**0.79**	**0.79**	**25296**		*0.78*	*0.78*	*18518*		0.77	0.78	25580
0.8182 - 0.9090	**0.86**	**0.87**	**23347**		*0.87*	*0.85*	*23600*		0.87	0.87	13811
0.9091 - 1.0000	**0.97**	**0.97**	**3074**		*0.95*	*0.95*	*5841*		0.97	0.96	2181
Metrics	**AUC 0.728**	**R^2 0.145**	**RMSE 0.406**		*AUC 0.710*	*R^2 0.115*	*RMSE 0.413*		AUC 0.653	R^2 0.058	RMSE 0.426
	stdev (pred): 0.166				*stdev(pred): 0.147*				stdev(pred): 0.107		

By looking at just the classic metrics (AUC, R^2, RMSE), we cannot say *why* one model scores better than another. However, by looking at a table of the predictions and test data averages (which we call a "Coherence-Frequency Table"), we can learn more about the model's coherence and differentiation.

From 0.60 and up, all three models have very similar coherence; that is the predictions closely match the test data averages. However, KT has over-predicted in three of the 6 groups below 0.60. Since the other two groupings have small numbers of instances (n<100), we are reluctant to draw conclusions of accuracy in those cases. PFA appears to be reasonably consistent, as long as there are large numbers (n>200) in each group; however, one could argue that PFA consistently *under*-predicts in this range. Others [13] have previously reported on KT over-reporting; however, this analysis allows a researcher to examine under or over reporting in finer detail.

Although there is not as large of a change in differentiation (e.g. moving from a uni-modal distribution to a bimodal one), what we can say is that PFA has done the worst of the three at moving instances away from the training mean. The major reason why SB scores so well against the other two has to do with its ability to differentiate and bring more predictions below 0.50, while maintaining coherence.

The easiest way to measure the differentiation of the prediction values might be to report the standard deviation of prediction values. In this case, the values are: SB: 0.166; KT: 0.147; PFA: 0.107. As a way to compare to the "ideal", we could report either the standard deviation of the test data (0.439), or the standard deviation of the training data (0.440).

4.3.2 Example 2 - Two model variations trained on a small (~32K, single skill) dataset.

Curious to see if the idea of differentiation and coherence can also apply to model outputs from smaller analyses, we applied two very similar algorithms to a single skill dataset. Using the PFA-Decay algorithm introduced in [8], we applied two different decay values (0.5 and 0.9) to only one skill. We selected a dataset that we knew (from previous use) would generate near identical RMSE values, but different AUC values. Since our training set was relatively small, and the two models are very similar, the range of prediction values is much less than 0-1. Using the same

Table 8: Example 2 Two similar models trained on a small (one skill, ~40K instances) dataset. The average NPC of the training data is 0.7907; the three bins nearest the training average have been underlined.

	PFA with 0.9 Decay										*PFA with 0.5 Decay*
Range	**Pred**	**Test**	**n-TOT**	**n-HKS**	**n-LKS**		*Pred*	*Test*	*n-TOT*	*n-HKS*	*n-LKS*
0.088 - 0.167	**0.1299**	**0.1579**	**19**	**0**	**19**						
0.168 - 0.245	**0.2122**	**0.4167**	**24**	**4**	**20**						
0.246 - 0.324	**0.2806**	**0.3000**	**40**	**19**	**21**						
0.325 - 0.403	**0.3612**	**0.4459**	**74**	**41**	**33**						
0.404 - 0.481	**0.4426**	**0.5192**	**104**	**39**	**65**		*0.4437*	*0.3962*	*212*	*64*	*148*
0.482 - 0.560	**0.5223**	**0.5568**	**176**	**52**	**124**		*0.5279*	*0.5759*	*415*	*133*	*282*
0.561 - 0.638	**0.5970**	**0.5811**	**296**	**86**	**210**		*0.5983*	*0.6258*	*310*	*110*	*200*
0.639 - 0.717	**0.6749**	**0.6492**	**573**	**189**	**384**		*0.6864*	*0.6663*	*950*	*410*	*540*
0.718 - 0.795	**0.7593**	**0.7131**	**1471**	**604**	**867**		*0.7584*	*0.7224*	*508*	*178*	*330*
0.796 - 0.874	**0.8475**	**0.8737**	**2826**	**1489**	**1337**		*0.8333*	*0.8180*	*1599*	*732*	*867*
0.875 - 0.953	**0.9017**	**0.8702**	**1333**	**864**	**469**		*0.8827*	*0.8861*	*2942*	*1760*	*1182*
	AUC: 0.695; RMSE: 0.393; R^2: 0.089						*AUC: 0.694; RMSE: 0.394; R^2: 0.088*				
	stdev(predictions): 0.128						*stdev(predictions): 0.124*				

Table 9: Example 2, cont'd: Coherence-Frequency Table of PFA with two decay values applied to a single skill dataset that has been separated by student knowledge. The prediction ranges are the same as in Table 8.

PFA 0.9 Decay-HKS			*PFA 0.5 Decay-HKS*			**PFA 0.9 Decay-LKS**			*PFA 0.5 Decay-LKS*		
pred	**test**	**n**	*pred*	*test*	*n*	**pred**	**test**	**n**	*pred*	*test*	*n*
						0.141	**0.167**	**12**			
0.229	**0.714**	**7**				**0.210**	**0.250**	**20**			
0.298	**0.478**	**23**				**0.286**	**0.200**	**25**			
0.365	**0.400**	**35**				**0.365**	**0.469**	**32**	*0.399*	*0.109*	*46*
0.441	**0.424**	**33**				**0.445**	**0.538**	**65**	*0.444*	*0.561*	*198*
0.527	**0.617**	**47**	*0.542*	*0.444*	*45*	**0.530**	**0.557**	**183**	*0.519*	*0.557*	*273*
0.608	**0.575**	**73**	*0.597*	*0.567*	*90*	**0.611**	**0.602**	**342**	*0.618*	*0.566*	*433*
0.681	**0.636**	**110**	*0.676*	*0.635*	*189*	**0.687**	**0.615**	**602**	*0.674*	*0.681*	*335*
0.758	**0.747**	**281**	*0.763*	*0.743*	*435*	**0.759**	**0.762**	**724**	*0.762*	*0.761*	*566*
0.836	**0.804**	**850**	*0.843*	*0.818*	*533*	**0.826**	**0.850**	**1274**	*0.844*	*0.844*	*1698*
0.905	**0.907**	**1928**	*0.902*	*0.896*	*2095*	**0.892**	**0.852**	**270**			
AUC: 0.694		**RMSE: 0.356**	*AUC: 0.685*		*RMSE: 0.358*	**AUC: 0.673**		**RMSE: 0.424**	*AUC: 0.681*		*RMSE: 0.423*
stdev(pred): 0.115			*stedev(pred): 0.086*			**stedev(pred): 0.128**			*stedev(pred): 0.134*		
avg (pred): 0.841			*avg (pred): 0.849*			**avg (pred): 0.737**			*avg (pred): 0.724*		

methods as in example one, we created a range-averaged frequency table to compare coherence and differentiation. The results are Table 8.

At first glance, one might be inclined to score PFA with 0.9 Decay higher on differentiation; it has a wider range of scores. However, these models are essentially tied across all metrics. Even trying to subdivide the number of instances by High Knowledge Students and Low Knowledge Students (defined as simply above or below the median score of students' prior question correctness averages) doesn't tell us too much more. In this case, the models result in nearly identical outcomes.

We wanted to test the idea of whether refitting the models to high and low knowledge students gives meaningful differences. Table 9 below has the same range values as in Table 8. In this analysis, the students were separated before the models were run; this allows us to determine if the models work better for a certain subset of the student population.

Once the dataset has been divided by student knowledge, and the two models rerun, what we find is that (in this model-dataset interaction), PFA with 0.9 Decay does a better job of prediction for high knowledge students than PFA with 0.5 Decay. The reverse, however, is true for low knowledge students. In analyzing the Coherence-Frequency Table for the low knowledge students, we might (at first glance) be inclined to state that PFA 0.9 has done a better job at differentiation because there is a wider range of scores.

4.3.3 Example 3 - Coherence-Frequency Table based on prior knowledge scores

A Coherence-Frequency Table could also be used to test how well a given model applies to other features. For instance, with the two models from the previous example (PFA with 0.9 Decay and PFA with 0.5 Decay), we can test the model results coherence not across prediction ranges, but across prior knowledge ranges. To do so, we used all five folds, and averaged by student before we generated the results found in Table 10.

Table 10: Example 3: Coherence-Frequency Table arranged by student prior knowledge score.

Range (0-1.00)	**avg prior**	**avg pred PFA 0.9**	*avg pred PFA 0.5*	**avg npc**	**n students**
0.0000 - 0.0909	0.0379	**0.5299**	*0.6180*	0.4098	5
0.0910 - 0.1818	0.1296	**0.6817**	*0.7060*	0.5602	16
0.1819 - 0.2727	0.2270	**0.7308**	*0.7370*	0.6304	31
0.2728 - 0.3636	0.3175	**0.7649**	*0.7648*	0.7293	90
0.3637 - 0.4545	0.4183	**0.7626**	*0.7626*	0.7563	149
0.4546 - 0.5454	0.5068	**0.7814**	*0.7803*	0.7661	387
0.5455 - 0.6363	0.5953	**0.8011**	*0.8017*	0.8379	850
0.6364 - 0.7272	0.6839	**0.8199**	*0.8214*	0.8800	1587
0.7273 - 0.8181	0.7721	**0.8321**	*0.8344*	0.9056	1761
0.8182 - 0.9090	0.8553	**0.8440**	*0.8457*	0.9336	938
0.9091 - 1.0000	0.9412	**0.8582**	*0.8631*	0.9656	205

What we find here is that both models over-predict for the lower six groups of student prior knowledge, and under-predict for the top five groups of students. This could give a model designer input into how to adjust their model for a future iteration.

5. CONCLUSION

Most of the time, the traditional metrics agree with each other and tend to give straight-forward meanings when used to compare the predictive power of models. However, there are times when the metrics disagree. Two examples we have shown are: when differentiation / ordering is the same, but predictions vary, and when a model overwhelmingly predicts at just one extreme. There also may be times (e.g. when the training average is near 0.50) when the differences in scores between two models is very minimal and thus model comparison is difficult.

We conclude that, if we are to accurately compare knowledge predicting models to each other, we need to look at new metrics, in addition to a mix of old metrics. We do not believe that we are proposing the "ultimate" single metric that will definitively state which model is "better". We are stating that we believe model comparison is improved when it contains (AUC or A'), (R^2 or Efron's R^2), RMSE, and the standard deviation of the predictions. A more thorough comparison might also include ROC curve analysis and / or a Coherence-Frequency Table analysis in an attempt to identify regions of habitual over or under prediction. However, we are aware of the limitations of space in papers. One space-saving solution might be to give summary statistics and embed a tinyurl with a link to a static webpage with more detailed graphical / chart-based analyses.

5.1 Future Work

Stepping back, what is our contribution? The trailing author on this paper, Heffernan, has contributed to the community a large number of user models, but is honestly confused about "How should we in the EDM community evaluate models?" Some authors have already recommended we should report multiple metrics; we should be skeptical of a result that reports only one metric. One potential solution would be for the EDM community to adopt a "best practices guide" (and updates it on a, perhaps, annual or biennial basis).

We are tempted to recommend to the EDM community that the methods presented here might be a logical next metric to look at. As we show in example 3, grouping instances can be based on some other student feature; this sort of analysis could give a researcher further insight into trade-offs that have been happening behind the scenes. (E.g., a researcher might find that some groups of students are over predicted, while others are under-predicted; care, of course, will need to be made to ensure that false groupings are not created.)

We think we have raised more questions than we have answers, and encourage our colleagues to help the EDM community come up with better ways to evaluate models.

6. ACKNOWLEDGEMENTS

We would like to thank Ryan Baker and Joseph Beck for taking the time to discuss these ideas with us and make suggestions. We also thank multiple NSF grant (ACI-1440753, DRL-1252297,DRL-1109483,DRL-1316736 & DRL-1031398) , the US Dept of Ed's (IES R305A120125 & R305C100024 and GAANN), the ONR, and the Gates Foundation.

7. REFERENCES

[1] Beck, J. E., & Xiong, X. (2013). Limits to accuracy: How well can we do at student modeling. *Educational Data Mining*.

[2] Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4(4), 253-278.

[3] Desmarais, M. C., & d Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, *22*(1-2), 9-38.

[4] Dhanani, A., Lee, S. Y., Phothilimthana, P., & Pardos, Z. (2014). A comparison of error metrics for learning model parameters in bayesian knowledge tracing. Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley.

[5] Efron, B. (1978). Regression and ANOVA with zero-one data: Measures of residual variation. *Journal of the American Statistical Association*, *73*(361), 113-121.

[6] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

[7] Fogarty, J., Baker, R. S., & Hudson, S. E. (2005). Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. *Proceedings of Graphics Interface 2005*. Canadian Human-Computer Communications Society.

[8] Gong, Y., Beck, J. E., & Heffernan, N. T. (2011). How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education*, *21*(1), 27-46.

[9] Gong, Y., Beck, J. E., & Ruiz, C. (2012). Modeling multiple distributions of student performances to improve predictive accuracy. *User Modeling, Adaptation, and Personalization*, 102-113.

[10] Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29-36.

[11] Pavlik Jr, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis--A New Alternative to Knowledge Tracing. *Online Submission*.

[12] Powers, D. M. (2012). The problem of area under the curve. *International Conference on Information Science and Technology (ICIST), 2012*. IEEE.

[13] Qiu, Y., Pardos, Z. & Heffernan, N. (2012). Towards data driven user model improvement. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Florida Artificial Intelligence Research Society (FLAIRS 2012). pp. 462-465.

[14] Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (eds.) Proceedings of the 7th International Conference on Educational Data Mining.

[15] Steel, R. G., & Torrie, J. H. (1960). *Principles and procedures of statistics with special reference to the biological sciences.* New York: McGraw Hill.

[16] Van Inwegen, E. G., Adjei, S. A., Wang, Y., & Heffernan, N. T. "Using Partial Credit and Response History to Model User Knowledge" *submitted to Educational Data Mining 2015,* in review.

[17] Wang, Y., & Heffernan, N. T. (2011). The" Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *FLAIRS Conference*.

[18] Yudelson, M., Pavlik Jr, P. I., & Koedinger, K. R. (2011). User Modeling–A Notoriously Black Art. *User Modeling, Adaption and Personalization*, 317-328.