

Using Association Rules to Guide a Search for Best Fitting Transfer Models of Student Learning

Jonathan Freyberger, Neil T. Heffernan, and Carolina Ruiz

Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA 01609
frey@wpi.edu, nth@wpi.edu, and ruiz@wpi.edu

Abstract: We say a *transfer model* is a mapping between the questions in an intelligent tutoring system and the knowledge components (i.e., skills, strategies, declarative knowledge, etc) needed to answer a question correctly. [JKT00] showed how you could take advantage of 1) the Power Law Of Learning, 2) an existing transfer model, and 3) set of tutorial log files, to learn a function (using logistic regression) that will predict when a student will get a question correct. In the main conference proceeding [CHK2004] give an example of using this technique for transfer model selection. Koedinger and Junker [KJ99] also conceptualized a search space where each state is a new transfer model. The operators in this search space split, add or merge knowledge components based upon factors that are tagged to questions. Koedinger and Junker called this method *learning factors analysis* emphasizing that this method can be used to study *learning*. Unfortunately, the search space is huge and searching for good fitting transfer models is exponential. The main goal of this paper is show a technique that will make searching for transfer models more efficient. Our procedure implements a search method using association rules as a means of guiding the search. The association rules are mined from a dataset derived from student-tutor interaction logs. The association rules found in the mining process determine what operations to perform on the current transfer model. We report on the speed up achieved. Being able to find good transfer models quicker will help intelligent tutor system builders as well as cognitive science researchers better assess what makes certain problems hard and other problems easy for students.

Keywords: transfer models, logistic regression, association rules, prediction, search, learning factors analysis

Introduction

Intelligent tutoring systems are able to provide instruction to students and to model their current knowledge by knowing what skill(s) (we use the term *skill* and *knowledge component* interchangeable) a student needs to solve a given problem. However, creating an accurate model of a student's knowledge can be quite difficult [KLEG92]. Problems usually have more than one skill associated with them. *Transfer* of knowledge is the process by which practice on one type of problems makes a student more likely to do well on a different type of problem. We assume that transfer only happens between problems types if the two problem types share knowledge components that are required to solve both types of problems. *Transfer models* reify assumptions of when knowledge component are shared between problem types. Creating and comparing *transfer models* can make it easier to assess what skills are required for a given problem. A transfer model is a mapping between different types of problems and the skills needed to solve them. If two different problem types have one or more skills in common then this suggests that transfer between these two different problems exists. Thus, students practicing with one type of problem might cause them to perform better on the other type of problem. We further make a large simplifying assumption that if the model predicts transfer then it should happen, as we do not model students not realizing that transfer is possible.

For instance, a hypothetical transfer model is shown below for two question types. Suppose you gave students both 1) problems that involved calculating the area of a circle, given the radius, as well as 2) problems where

they were given the area and had to find the radius. Would students transfer learning between these two question types?

Question Type (Example)\ Skills	Skill1 (Formula)	Skill2(Apply Forward)	Skill3(Apply Backward)
Circle Area (If the radius is 5 what is the area?)	1	1	0
Circle Radius (If the area of a circle is 10 what is the radius?)	1	0	1

Alternatively put, if you gave them practice on a bunch of one type would they improve at the other type? This is an empirical question, but we can represent several different answers to this question by making transfer models that embody different predictions of “transfer”. The above model predicts that practice on *circle area* problems will transfer to increased performance on circle radius to the extent that students get better at skill1, retrieving the formula for the area of a circle. If we removed the column for skill1, it would predict that students would not transfer any learning between the two question types because the two questions would no longer have any overlapping knowledge components (in this case we might want to rename skill2 to be “Calculate circle area” and skill3 to “Calculate radius given area”). Alternatively, if we removed the column labeled skill2, the transfer model would predict that *circle-radius* problems would be strictly harder than *circle area* problem because the *circle-radius* problems involve a superset of the knowledge components used in the circle-area problems. This transfer model (i.e., without skill2) would be reasonable if you thought that it was reasonably easy to take a formula and evaluate it, but very hard to do the square-root required to calculate the radius.

In this work we will allow each skill to have two parameters that define 1) the learning rate for each skill (some skills are easier to learn than others) and 2) the initial difficulty for each skill (some student have had more experience than others at each skill, and this is one reason why some skills will be more difficult than others). One of the authors has done some previous work on creating an automatic search process for generating transfer models [OH03] using a blind brute force approach in finding the best fitting transfer model for a given dataset. This paper reports on work that has developed a guided search process. A data mining technique, association rules, is used to guide our search. We have found that by mining for association rules, in the instances that transfer model predicts incorrectly, information can be obtained on how to guide the search process more efficiently toward finding a better model. A guided search process allows for quicker discovery of good fitting transfer models for a given dataset. In addition, the transfer models generated from the search process have the potential of discovering new information as to what exactly makes a particular problem difficult and others easy.

Background

The work described in this paper draws its resources from five main areas: 1) the student-tutor interaction logs from the online algebra tutor Ms. Lindquist, 2) transfer model definition as applied to a set of log files, 3) transfer model evaluation and comparison, 4) searching for transfer models, and 5) association rules. These will now be discussed in order.

Ms. Lindquist

Ms. Lindquist is an algebra tutor designed to help students learn how to write algebraic expressions [H01], [HK98]. It can be found at <http://www.algebratutor.org>. The data for this work comes from the log files Ms. Lindquist generates when a student uses the tutor. The main conference presents analyses of student learning found in these log files [HC04]. Ms. Lindquist is targeted towards 7th and 8th grade students and takes a dialog-based approach. Every problem Ms. Lindquist gives is a word problem. An example problem is “Mary is 800 ft from the dock. If she rows 40 feet per minute how many feet is she from the dock after x minutes.” The student is expected to answer $800-40*x$. An illustration of the student-tutor dialog is given in Table 1. For now, only look at the column headed “Example Question” and the column headed “Response” which indicates if the student's input was marked as correct or not.

Scenario Identifier		Question Types		Factor: Task directions		Factor: Steps		Attempt at problem type		Hint Level		Components: difficulty parameters												Components: learning parameters																								
1	1	0	0	2	1	0	0	0	0	0	0	Response Time (Seconds)	Response (Correct or not)	Arithmetic	Comprehending One-Step	Compose Concrete	Articulating One-Step	Articulating Variable	Articulating Composed Expression	Arithmetic	Comprehending One-Step	Compose Concrete	Articulating One-Step	Articulating Variable	Articulating Composed Expression	Arithmetic	Comprehending One-Step	Compose Concrete	Articulating One-Step	Articulating Variable	Articulating Composed Expression																	
1	1	0	0	2	1	0	0	0	0	0	0			0	2	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
Example Question																																																
Scenario 1: Anna is rowing a boat in a lake and is 800 yards from the dock which she started. She rows back towards the dock at 40 yards per minute for "m" minutes and stops to rest. How far is she from the dock now? Tutor: Please write an expression for the distance Anne has left to row.																																																
2	1											130	N																																			
3	1	C		2	1	0								2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
Tutor: Hmm. No. We will come back to the distance Ann has left to row. Let me try to break this down for you. Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes?																																																
4	1											38	N																																			
Student: 580																																																
5	1	C		1	1	0								1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
Tutor: Hmm. No. Let me ask you a different question. Please compute the distance rowed so far if the number of minutes rowing was 3 minutes?																																																
6	1											19	N																																			
Student: 12																																																
7	1	C		1	2	1								1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
Tutor: Hmm. No. Try again please. Please compute the distance rowed so far if the number of minutes rowing was 3 minutes? Hint: Use the fact that her speed rowing is 40 yards per minute and the number of minutes rowing is 3 minutes.																																																
8	1											9	Y																																			
Student: 120																																																
9	1	C		2	2	0								2	2	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
Tutor: Correct. Now, lets try this question again: Please compute the distance Anne has left to row if the number of minutes rowing was 3 minutes?																																																
10	1											24	Y																																			
Student: 680																																																
11	1	A		2	1	0								0	2	0	2	1	1	3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
Tutor: Correct. Using only 800, 40, and 3 write an expression for the distance Anne has left to row																																																
12	1											18	N																																			
Student: 120-800																																																
13	1	A		1	1	1								0	1	0	1	0	0	3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
Tutor: No. Your answer of "120-800" equals "-680", but you wanted to write an expression that equaled "680". But that is only because you didn't say where the 120 came from. What math did you do to get the 120?																																																
14	1											22	Y																																			
Student: 3*40																																																
15	1	A		2	2	0								0	2	0	2	0	1	3	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0												
Tutor: Correct. Now, lets try this question again: Using only 800, 40, and 3 write an expression for the distance Ann has left to row																																																
16	1											18	Y																																			
Student: 800-40*3																																																
17	1	S		2	2	0								0	2	0	2	1	1	3	6	1	3	0	1																							
Tutor: Correct. Now use "m" instead of 3 to again express the distance Ann has left to row																																																
18	1											9	Y																																			
Student: 800-40*m																																																
19	1																																															
Tutor: Correct. Lets do a new problem																																																
20	2	S		2	1	0								0	2	0	2	1	1	3	8	1	5	1	2																							
Scenario 2: Michael works as a waiter. Saturday he worked "h" hours. The restaurant pays him an hourly wage of \$3 an hour. He also made \$40 dollars in tips. Write an expression for the total amount he earned Saturday Night. Tutor: Write an expression for the total amount he earned Saturday Night.																																																

Table 1: Example transfer model mapped to a dataset

Transfer model definition as applied to a set of log files

A transfer model is a mapping between different types of problems and the knowledge components needed to solve them. A transfer model is represented by a table, where the rows represent the different problem types and the columns indicate the knowledge components needed to solve each problem type. Ms. Lindquist records in log factors about each question a student answers such as the number of steps the question involves or what type

of question it is. If there are two factors; say *number of steps* and *task-directions*, where the number of steps for a question can be either one or two, and the possible values for task-directions can be of one of 3 values (compute [i.e., “Can you compute a value?”, Answer=680], articulate [i.e., “Can you articulate what math you would need to do to get a value?”, Answer=800-3*40], and symbolize [i.e., “Can you write an expression that has a variable?”, Answer=800-40m],), a transfer model that crossed these two factors would have six rows.

Table 1 shows the result of mapping an example transfer model, with these two factors, to a sample Ms. Lindquist session. In particular, you will see that columns labeled with “Difficulty parameters” are the knowledge components, and their mapping to particular question types depends completely upon the two factors (in column 2, “Task-Directions”, and 3 “Steps”) that when crossed result in 6 different question-types.

The first 9 columns in Table 1 represent the information Ms. Lindquist logs while a student is using the tutor. Column 7 in Table 1 shows an example of the questions a Ms. Lindquist asks a student based on the student’s previous answers. The “difficulty parameters” shown in columns 10 through 15 represent the skills the student needs to solve each of the questions Ms Lindquist asks. Columns 16 through 21 represent the learning parameter for each skill. A learning parameter value represent the number of times a student has answered correctly a question (and thus got positive feedback enabling him to “learn”) that involved that particular skill. We see that the learning parameters don’t start to increase until the 9th row, because that is the first time that the student gets positive feedback that he did something correct. In particular, in row 7 we see that the student was asked to do a problem that involved doing what the transfer model predicted involved using both the “Arithmetic” and the “Comprehend One-Step” knowledge components. We see that by the last row in the table, the student has been given positive feedback on arithmetic three times, and thus our model will expect the student to be more likely to get a problem correct that involves arithmetic than a student that had fewer opportunities to get positive feedback on arithmetic. (Note that [CHK04] deals with the this counting in a more sophisticated manner than we do here.) It is because we are modeling different *learning* rates for each knowledge component that this technique is called “*learning* factor analysis” rather than just “*difficulty* factor analysis.”

Transfer model evaluation and comparison

Following [JKT00], and explained in greater detail in [CHK04], we applied logistic regression to predict the student’s response (binary), based upon 1) student characteristics (like pretest scores) as well as 2) item characteristics (which we call difficulty parameters that are shown in columns 10-15), and 3) characteristics of items that differ with student practice (called learning parameter and shown in the last 6 columns of Table 1). (We have also looked at modeling that fact that some students might be better learners than others, but this work only reports on different rates of learning due to different knowledge components, not do to different student.) A logistic regression is used when the dependent variable is binary. In the case of transfer models, we want to use a logistic regression to predict whether a student will get the answer to a given problem correct or not. Thus, whether the student answered the problem correct or not is the binary dependent variable. A value of one (response=1) means the student answered the problem correctly and a value of zero (response=0) represents the student answered the problem incorrectly. For this work a statistical package, S-Plus, was used to perform logistic regressions.

In model selection, a perennial problem is that as you add more parameter you can get a better fit, but is the increased fit worth the extra parameters? We can compare two models by using the Bayesian Information Criterion (BIC) [R95] that applies a penalty for using more parameters to allow you so say that you have the “best fitting most parsimonious” model. For one model *m* to be more statistically significant than another model *n*, the BIC value of *m* must be at least 10 BIC points lower than *n*’s BIC. Overall accuracy will be used to determine the strength of a given transfer model on its own. ([CHK04] also used a k-foldout strategy in addition to BIC with somewhat similar results)

	Arithmetic	compute an expression	articulate an expression	symbolize an expression	generalize an expression
	Steps	Qtype=compute	qtype=articulate	qtype=symbolize	Qtype=generalize
problem_type0	1	1	0	0	0
problem_type1	2	1	0	0	0
problem_type2	1	0	1	0	0
problem_type3	2	0	1	0	0
problem_type4	1	0	0	1	0
problem_type5	2	0	0	1	0
problem_type6	1	0	0	0	1
problem_type8	2	0	0	0	1

Table 2: Example transfer model, the first row shows potential names for the skills, the second row shows which factor value pair the skill is associated with, qtype= question type

Searching for transfer models: A search space of transfer models

Koedinger and Junker [KJ99] conceptualized a search space where each state is a new transfer model. The operators in this search space included *split*, *add* and *merge*. In this work we will ignore the *merge* operator. (Extending [KJ99] we have also added the *map* operator that is similar to *add* but works for nominal instead of categorical data.) This approach was called by [KJ99] *learning factors analysis* because it starts with factors. In *task-directions* (Table 1, column 3) and *steps* (column 4) are factors that are simply tags to individual questions. An example of another factor would be a binary factor for whether the word problem involves money (suspecting that problems that involve money might be easier for students). Table 2 shows the results of doing a *map* of *number-of-steps* (creating the second column in the transfer model) as well as an *add* of *task-directions* (which creates a new column for each of the 4 values of *task-directions*). Note that in Table 1 we did not have an example of a question that was tagged with the task-direction="generalize". Given that we cross the two factors (*number-of-steps* and *task-directions*), (where the number-of-steps for a question is quantitative and is either one or two) and the task-directions can be of one of four types (compute, articulate, symbolize and generalize), a transfer model for these would have eight rows. The first row in Table 2 shows potential names for the skills in the transfer model. Creating names for skills in a transfer model is a highly creative process. When a factor is included into a transfer model it is referred to as a knowledge component or skill. Thus, the 5 rightmost columns in the above transfer model represent skills.

The more complicated way of incorporating a factor is by the *split* operator. The split operator takes as input 1) a knowledge component to split that is already in the transfer model, and 2) a factor to split the knowledge component by. A split will create a new transfer model by remove the existing knowledge component and replacing it by knowledge components for each of the values of the factor. Splitting makes use of an existing skill *s* in the transfer model and a factor *f*. The existing skill is replaced with *n* new skills where *n* is the number of values the factor *f* can have. Figure 1 shows how a *split* operation is performed. A refined *split* can also be performed where a skill is only split into two new skills based on one value for a factor *f*.

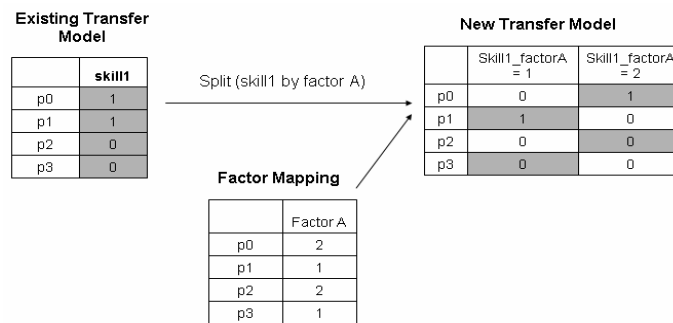


Figure 1: Example of a split operation, FactorA has two possible values and when it is used to split the skill *skill1*, two new skills are created.

Given a collection of data instances, each one described in terms of a set I of attribute-value pairs or features (sometimes called “items”), association rules are rules of the form $X \Rightarrow Y$, where X and Y are disjoint subsets of I [AIS93]. Apriori is the standard algorithm to generate association rules [AS94]. It is probable to find many associations in a dataset so interest is usually focused on rules that have high accuracy and/or large support. Support of the association rule $X \Rightarrow Y$ is the percentage of instances in the dataset that contain both X and Y . Confidence of $X \Rightarrow Y$ is the percentage of instances that contain Y among those that contain X . Another measure that is used in addition to support and confidence is lift. The lift of $X \Rightarrow Y$ is the ratio between the conditional probability of Y given X , and the prior probability of Y in the dataset. Yet another measure for evaluating an association rules is Chi-square. Chi-square is a means of determining how well an association rule would apply to a larger dataset. For the purpose of mining association rules from tutorial logs, we used the AprioriSetsAndSequences (ASAS) algorithm. ASAS is an extension to the Apriori algorithm [P04]. ASAS allows one to mine temporal data. In addition, ASAS provides enhanced pruning methods for association rule mining.

Our Search Algorithm

Now that we have reviewed the prior work, we will discuss our new search algorithm. For this work, we used association rules to guide a search process in finding transfer models that predict student's success. The transfer models themselves provide information as to what skills are required to solve a particular type of problem. The search process starts with a list of factors, a dataset and an empty transfer model as the root node. The dataset is a table where the rows represent a question Ms. Lindquist gave to a student and the columns are the factors for each question. There is one additional column in the dataset that indicates whether the student answered the question correctly or not. Looking back at Table 1, columns 3, 4 and 9 are an example of what would be found in the dataset.

Each node in the search tree represents a new transfer model. To generate new transfer model nodes from a current transfer model node, the search algorithm performs one of three operations on the current transfer model. The three operations are *map*, *add* and *split*. Only one operation is performed on a transfer model at a time and each operation includes only one new skill in the current transfer model, creating a new transfer model node. The new transfer model node becomes a child of the transfer model node it was created from.

Since the root node contains an empty transfer model, only add or map operations can be performed on the root node to create the first level of transfer models. A list of possible skills to include in the transfer model is generated by mining the dataset for factors associated with a student's response being correct or incorrect. The initial set of mined rules have the form “factor=value \Rightarrow response=(0|1) [Conf: #, Sup: #, Lift: #, Chi-square: #]” where *factor* is the name of a factor in the dataset (e.g., steps, question type), *value* is one of the possible values the factor can have (e.g. steps can have a value of either 1 or 2), *response=0* means students' response is incorrect and *response=1* means student's response is correct. A hypothetical set of association rules obtained from mining the dataset is shown in Figure 2.

```

steps=1  $\Rightarrow$  response=1
           [conf: 0.9, sup: 0.6, lift: 1.02, Chi-Square: 30]
qtype=2  $\Rightarrow$  response=0
           [conf: 0.9, sup: 0.4, lift: 1.25, Chi-Square: 40]
qtype=3  $\Rightarrow$  response=1
           [conf: 0.9, sup: 0.5, lift: 1.04, Chi-Square: 20]

```

Figure 2: Hypothetical set of association rules obtained from mining the Log+Factors dataset

The first rule in Figure 2 is interpreted by the search process as potentially adding a skill for one step problems to the initial model. Since *steps* happens to be a quantitative factor it can be mapped into the initial transfer model as well. The second rule suggests adding a skill for questions of type two, and the third rule suggests adding a skill for questions of type three.

Thus, in this hypothetical situation the search process would create four child transfer model nodes from the root node. The first child node would represent the initial transfer model with a skill for one steps problems added.

The second would represent the initial transfer model with the factor steps mapped into the initial transfer model. The third child node would represent the initial transfer model with a skill for questions of type two added. The fourth child node would represent the initial transfer model with a skill for questions of type three added. These first level nodes can be referred to as the base models. It is from these transfer models that all other transfer models are built.

Transfer models built from the base models are constructed a little differently than how the base models are constructed. First the base models are mapped to the dataset (which we call a Log+Factors dataset), creating what we call a Log+Factors+TransferModel dataset. A Log+Factors dataset is just a dataset where each instance represents a question a student was asked. Columns 3, 4 and 9 in Table 1 compose a Log+Factors dataset. Each question in a Log+Factors dataset is marked with the factors related to that question. In a Log+Factors+TransferModel dataset questions are marked with their associated factors along with specific skills need to solve each question. Columns 9 through 21 in Table 1 compose a Log+Factors+TransferModel dataset. A logistic regression is run over the Log+Factors+TransferModel dataset. The instances that the logistic regression predicted incorrectly are then mined for association rules using the AprioriSetsAndSequences algorithm [P04].

Thus, we are looking for commonalities in the instances in the Log+Factors+TransferModel dataset predicted incorrectly by the logistic regression. Given a data instance, that is a problem posed to a student, the logistic regression outputs a numeric value between 0 and 1 representing the predicted probability of the student answering the problem correctly. We use a threshold of 0.5 to convert this value into a Boolean prediction: if the output of the logistic regression is less than 0.5, we predict that the student will answer the problem incorrectly, if greater than or equal to 0.5 we predict correct. These predictions are compared against the actual response from the student. The instances in which the prediction from the logistic regression and the correctness of the actual student response disagree are considered incorrectly predicted by the logistic regression. It is these incorrectly predicted instances that are mined for association rules that will determine the next operation to perform on a transfer model. These association rules are used to determine the new transfer models to be created from the base models.

For the initial transfer model only the operations add and map can be used. For all transfer models created from the base models the split operation can be used as well. Association rules of the form “factor=value ==> response=(1|0)” are interpreted as potential add or map operations to be performed on the base model. Association rules of the form “skill=value && factor=value ==> response=(0|1)” are interpreted as a potential split operation to be performed on the base model where *skill* in the transfer model is split by *factor = value*.

Figure 3 shows a hypothetical set of association rules that could be obtained by evaluating a base transfer model with one skill for one step problems and mining the instances that the model predicts incorrectly.

```

steps=2 ==> response=1
    [Conf: 0.9, Sup: 0.5, Lift: 1.10, Chi: 10]
steps=1 && qtype=2 ==> response=0
    [Conf: 0.8, Sup: 0.7, Lift: 1.20, Chi: 14]

```

Figure 3: Hypothetical set of association rules mined from base model

The first rule in Figure 3 suggests creating a new transfer model with an additional skill for two step problems. For the second rule in Figure 3, since there is already a skill for one step problems in the parent transfer model, this rule suggests splitting that skill by the factor question type when it has a value of two.

The same process for generating transfer models from the base models is repeated for every child node of the base models. The transfer model is expanded by mapping it to the Log+Factors dataset, creating a Log+Factors+TransferModel dataset which is then inputted into a logistic regression. Next, association rules are mined from the instances in the Log+Factors+TransferModel dataset that the logistic regression predicted incorrectly, and new transfer models are created from the current transfer model based on the association rules obtained.

For every transfer model that is evaluated, a BIC value for the model is obtained. In addition, the overall accuracy for a model is calculated. Overall accuracy for a model is the number of correctly predicted instances (i.e. the logistic regression created from the model predicted a correct response and the actual response was correct, or the logistic regression predicted an incorrect response and the actual response was incorrect) divided by the total number of instances. These two numbers help determine how good a transfer model is and are used in guiding and terminating the search process.

A path in the search space is explored until no new association rules can be found in the instances the transfer model, through a logistic regression, predicts incorrectly; if there are no instances predicted incorrectly; if no new skill can be added; or if no significant improvement is achieved from the transfer model's parent model (i.e. overall accuracy). In addition, the BIC for a transfer model is used to determine when to stop exploring one path and start exploring another. Low BIC values are desired, so when a transfer model's branch reaches a specified threshold (BIC values for transfer models differ from one dataset to another) further searching on that path will stop.

Several different ways were explored as to how to guide the search process. In addition, several different search algorithms were considered for the search process. The two types of search algorithms that were tested were depth first search and greedy search. As far as heuristics for guiding the search three different heuristic were considered. The heuristics rank each transfer model node available for expansion and the best ranked node is expanded first. The first heuristic used was called the *equal-weight* heuristic and ranked each node by summing up the four metrics for the association rule used to create the model, namely support, confidence, lift, and chi-square. The other two heuristics that were used both attempted to predict a nodes BIC using the four metrics for the association rule used to create the model along with the BIC and overall accuracy for the node's parent node. One heuristic used a linear regression to do this prediction and the other used an artificial neural network. The two heuristics are referred to as the *regression* heuristic and *ANN* heuristic.

Results

In evaluating the search process, the transfer models generated from the search process are scrutinized by domain experts to assess how logical they are. The transfer models are rated on how applicable they are to a larger set of students. Whether the transfer model illuminates any new information about why students struggle with certain problems and have an easy time with others is an important factor as well. The objective evaluation criterion for the transfer model generated is how parsimonious the model is. Models with few skills are preferred. How cognitively plausible the model is is also a criterion. If there are valid reasons for the skills being in the transfer model, then this model is preferred over other models that have high accuracy but no clear connection between the skills and problems. A transfer model's overall accuracy and BIC are also criteria used to determine how good a transfer model is.

The experiments and results for this work are based on a dataset obtained from Ms. Lindquist. The dataset contains questions from Ms. Lindquist's section 2 and contains three factors: number of steps, question type and whether the question deals with money or not. A question can have one or two steps and there are four different types of questions in the dataset. The four types of questions are *symbolize*, *articulate*, *generalize* and *compute*. The primary dataset used to run the majority of the experiments contains 751 instances representing data for about 42 students.

Comparing the performance of the original blind search method to the search guided by association rules was done to assess how worthwhile it is to use association rules. Determining whether the search process using association rules is better required comparing how long it takes to find a good model using association rules with how long it takes to find a good model using the blind search method. If the association method can find comparably good models in less time than the blind search method, then using association rules is beneficial. Figure 4 shows how the three heuristic search methods using association rules compared to the blind search method.

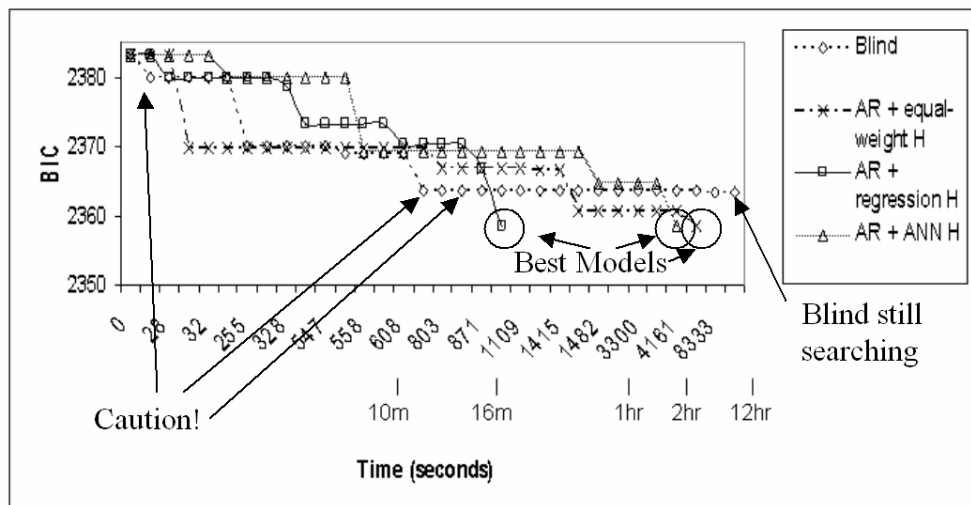


Figure 4: Comparison of heuristic searches to blind search

Figure 4 is a graph of BIC value versus time. The graph shows how quickly each search found a transfer model with a lower BIC in the search space. We plotted the BIC of the best transfer model found over time. As can be seen from looking at Figure 4 the three heuristic methods using association rules performed significantly better than the blind search method. Of course we do not know before hand when you have found the “best” and there could exist better fitting models that none of the searches found. The *regression* heuristic search performed the best and was able to find a transfer model with the lowest BIC value in only 16 minutes. The blind search method was not able to find a transfer model with as low a BIC value even after 12 hours. Figure 4 marks a few places (at 28 seconds and between 803 and 871 seconds) that if we had stopped the search, the blind search method would have appeared best. This could be just because the blind method got lucky in picking a branch of the search space to explore that happened to have a good model, or it could be symptomatic of some overly greediness of our guided search method. We will have to apply this method to other data sets to better study when.

Conclusion

In conclusion the search algorithm that this work has created allows for good transfer models to be found quickly. While it is not guaranteed to find the same transfer models as the brute-force search, it seems that it works well in practice, where we were able to find better models in a few hours as opposed to a few days for the brute force search. By using association rules to prune the search space in combination with a heuristic (such as linear regression) good transfer models were found in a reasonable amount of time.

References

- [AIS93] R. Agrawal, T. Imielinski and A. Swami: Mining Association Rules between Sets of Items in Large Databases, *Proc. of the 20th VLDB Conference*, pp. 207-216, 1993.
- [AS94] R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules, *Proc. of the ACM SIGMOD Conference on Management of Data*, pp. 487-499, 1994.
- [CHK2004] E. Croteau, N Heffernan, K Koedinger: Why Are Algebra Word Problems Difficult? Using Tutorial Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model, *7th Intelligent Tutoring Conference*: Maceio, Brazil, 2004.
- [KLEG92] S. Katz, A. Lesgold, G. Eggan, and M. Gordin: Modeling the Student in Sherlock II, *Artificial Intelligence in Education*, 1992 3(4), 495-518.

[H01] N. Heffernan: *Intelligent Tutoring Systems are Missing the Tutor: Building a More Strategic Dialog-Based Tutor*, PhD thesis, CMU, 2001.

[HC04] N. T. Heffernan, E. Croteau: Web-Based Evaluations Showing Differential Learning for Tutorial Strategies Employed by the Ms. Lindquist Tutor, *7th Intelligent Tutoring Conference*: Maceio, Brazil, 2004.

[HK98] N. Heffernan, K. Koedinger: A Developmental Model for Algebra Symbolization: The Results of a Difficulty Factors Assessment, *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, pp. 484-489, Hillsdal, NJ, Erlbaum, 1998.

[JKT00] B. Junker, K. Koedinger, M. Trottni: Finding improvements in student models for intelligent tutoring systems via variable selection for a linear logistic test model. Presented at *Annual North American Meeting of Psychometric Society*, Vancouver, BC, Canada, 2000.

[KJ99] K. Koedinger, B. Junker: Learning Factors Analysis: Mining student-tutor interactions to optimize instruction. Presented at Social Science Data Infrastructure Conference. New York University. November, 12-13, 1999.

[OH03] L. Orlova and N. Heffernan: Learning Factors Talk, presentation, Artificial Intelligence Research Group, Dept. of Computer Science, WPI, 2003.
<http://nth.wpi.edu/presentations/LearningFactorsTalkOct2003.ppt>

[P04] K. Pray: *Mining Association Rules from Time Sequence Attributes*, Master's Thesis, Dept of Computer Science, WPI, 2004.

[R95] A. E. Raftery: Bayesian model selection in social research. *Sociological Methodology* (Peter V. Marsden, ed.), Cambridge, Mass.: Blackwells, pp. 111-196, 1995.