Mining Student Learning Data to Develop High Level Pedagogic Strategy in a Medical ITS

Michael V. Yudelson[1,6], Olga Medvedeva[1], Elizabeth Legowski[1], Melissa Castine[1], Drazen Jukic[1,3], and Rebecca S. Crowley[1,2,4,5]

[1]Centers for Pathology and Oncology Informatics, University of Pittsburgh School of Medicine
[2]Center for Biomedical Informatics, University of Pittsburgh School of Medicine
Departments of Dermatology [3] and Pathology [4], University of Pittsburgh School of Medicine
[5]Intelligent Systems Program, University of Pittsburgh
[6]School of information Sciences, University of Pittsburgh

Abstract

We report the results of mining student learning data from SlideTutor – a cognitive tutor in a medical diagnostic domain. The analysis was aimed at finding both individual learning patterns as well as common misconceptions that students possessed. We have discovered that indeed there are distinct learner stereotypes: hint-driven learners, failure-driven learners, and a mixed group of learners that cannot be attributed to either one of the above two types. We have also found that students often make similar mistakes confusing certain visual features and diagnostic hypotheses. Our goal is to reuse the discovered patterns to engineer cross-case pedagogic interventions, enhancing our current immediate feedback methods with higher-level pedagogic reasoning. This paper describes the data-mining activities and potential implications of the data for pedagogic design.

1. Introduction

SlideTutor is an intelligent tutoring system with model tracing for teaching visual classification problem solving in microscopic pathology. SlideTutor is a tutoring system that implements both case-based and knowledge-based approaches (Clancey and Letsinger 1981, Clancey 1987, Clancey 1993). While working with SlideTutor, students examine "virtual slides" at different zoom levels. Students point at different locations of the slides, identify features, and specify feature attributes. Based on the identified sets of features they form hypotheses and diagnoses.

The effect of the tutoring has been previously studied (Crowley et al. 2005). The results of the study have shown that the tutor has a strong effect on diagnostic performance in both multiple-choice and case diagnostic tests. Learning gains were retained one week after a four-hour tutoring session. Although all of the students did demonstrate significant learning gains, their performance did not depend on case-based vs. knowledge-based interfaces. Nor did learning gains correlate with the level of postgraduate training or previous computer knowledge or experience. Thus, based only on outcomes of the study, we can conclude that students do learn, but we cannot tell how and what are the decisive factors.

SlideTutor's design is based on a set of integrated ontologies including ontologies for domain content and pedagogy (Crowley and Medvedeva 2006). All current pedagogic interventions are based on typical model tracing feedback – including hints and bugs targeting specific goals or skills. These skills are entirely case-focused, in that they will be instantiated specifically for the case that the student is working on. It is also very effective for training in this domain is to tutor "across cases". For example, when a student mistakenly suggests a specific feature – which is not present – instruction could focus on helping the student by showing other cases that have the suggested feature, and helping students to learn the differences between features. These "cross case" interventions are difficult because they require understanding about student misconceptions, which is simply not currently available. Thus, data-mining provided a unique opportunity to re-purpose existing experimental data to obtain an understanding of how we could apply higher level, cross-case interventions, with the goal of including these interventions in our pedagogic ontology.

2. Research Questions

The prior study analysis did show that students do learn but didn't tie the learning gains to any of the factors (Crowley et al. 2005). Thus the purpose of this paper is to find out how individual users learn in SlideTutor. What common mistakes do they make while solving cases? Are they using certain strategies when interacting with the tutor and how can they be classified based on those strategies? Our high level goal is to determine how mined common misconceptions and individual strategies can be used to provide for cross-case tutoring. We want to use the information about user behaviors as guidance for a high-level pedagogic intervention.

3. Methods and Results

In this section we present a sequence of results that we have obtained while analyzing student learning data stored in the SlideTutor database. SlideTutor collected three basic types of events. (Crowley & Medvedeva 2003, Medvedeva et al. 2005) *Interface events* record low-level human-computer interaction such as pressing a button or selecting a menu item. *Client events* capture combinations of interface events that represent the most atomic discrete subgoal, such as identifying a feature, suggesting a hypothesis, or asking for a hint. Client events are answered by *tutor responses*. Tutor responses indicate the response of the system to the last student action including the type of error for incorrect actions and the best-next-step at this point in the problem space. The database recorded the activity of 21 users. Although this number is not high, each user contributed enough data points to yield statistically significant results of analysis: users had to solve 20 problems identifying an average of 12 goal items in each.

A suite of MATLAB scripts were used for data mining and for data analysis and transformation. The following subsections present our findings in a logical order from most basic to most abstract.

3.1. Preliminary Analysis of Usage Behavior

SlideTutor registers three types of tutor response for any student action: 'confirm' for user's success, 'failure' for mistake and 'hint' if user asks for help. Our first step was to look at user activity "in the raw" and to try to single out possible patterns.

We have looked at user activity with respect to two factors. First, which types of subgoals students are trying to define in a problem: features, feature-attributes, or hypotheses. And second, which tutor event is generated for user action: confirm, failure, or hint. For each of the 20 problems the users have solved for each type of subgoal we computed the number of confirms, failure, and hint events normalized by the total number of all events for specific type of subgoal. In addition to the normalized number of hints, an average "hint depth" was calculated for each type of subgoal within problems. In contrast to counting hints here, each hint event was counted not as 1, but as a ratio. The numerator of the ratio being the maximum detail level of the hint detail the user saw and the denominator being the maximum hint detail level available. For example, if within a certain problem the user asked for hints twice and in the first case explored 4 out of 5 levels and in the second – 3 out of 5 levels of the hint, then for that problem his/her average hint depth would be (4/5+3/5)/2=0.7.

Figure 1 shows normalized activity curves for example student 1 who participated in the study. As can be seen from the plots, for the first 10 problems the relative failure rate is zero, while confirms and hints tie at .5 and hint depth is 1 (maximal). This means that a student constantly asks for hints, explores them to the maximum level of detail, and only then takes action. However, after about 10 problems s/he seems to have gained enough confidence to work on his/her own: failures go up, hints and hint depths go down, and confirms go up.

These four curves – confirms, failures, hints, and hint depths – were computed for each student and for each of the target subgoals types (features, feature-attributes, and hypotheses), plus an additional fourth one depicting activity regarding all subgoals.

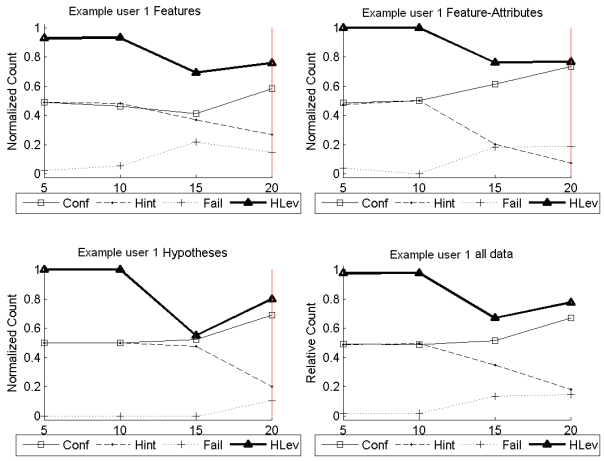

Figure 1. Activity of example user 1 grouped by 5 problems

If we look at Figure 2 that shows activity curves for example student 2, we will see a different picture. Hint rate is low through all 20 problems the student has solved; failures are high (especially for feature identification). Only confirms are still going up in general.

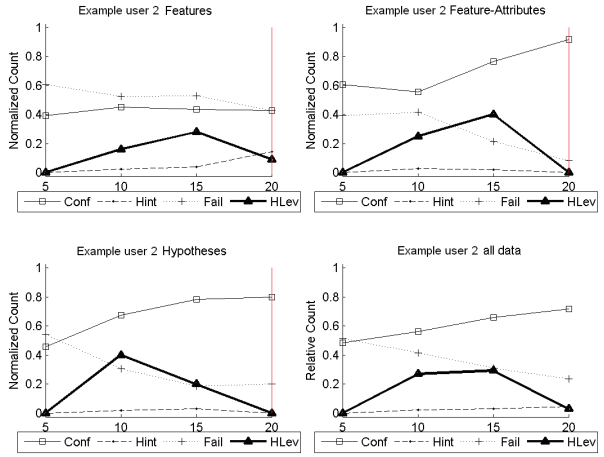

Figure 2. Activity of example user 2 grouped by 5 problems

The students whose activity data is shown in Figure 1 and Figure 2 are antipodes in terms of their hinting and failing behaviors. One has a lot of hints and a few failures, the other has many failures and a few hints. These students are the characteristic representatives of the two stereotypes we have singled out and defined as:

- "hint-driven" learners, and
- "failure-driven" learners.

In the following section we will show results on how participating students can be assigned to these stereotypes by using statistical methods.

3.2. Hint-Driven and Failure-Driven Learners

The major finding of the previous section is that there are user stereotypes based on the patterns of using the tutoring system. In this section we will talk about how users can be classified given their activity data.

User hinting and failing behavior was used as a basis for the classification decision. Namely, whether the number of qualified hints the user requests per problem is significantly different from the number of failures. By qualified hints we mean that the hint should be explanatory enough to be counted. After a hint request, a sequence of hint messages became available to the user. Hint messages ranged from very general (displayed automatically) to very detailed. It was up to the user whether to explore all levels of the hints or to stop at some point. Compare for example a hint with the depth 1 "there's something important in the slide" and hint with depth 5 "look at location X for feature Y".

User classification was done automatically by applying paired t-tests to hinting and failing activity data. This classification was first done separately for all types of subgoals the users had to define in a problem. Finally, the overall class was assigned to each student. The three types of activities were considered separately because identifying features, feature-attributes or hypotheses required different skills. Feature and feature-attribute identification required a search for visual clues. In contrast, hypotheses identification requires recognition of patterns of the features and attributes that support or refute a diagnosis. Below are the results of user classification we have obtained (Table 1).

Significant user preference of hints or failures in each of the categories (features, feature attribute-values, and hypotheses) is marked by capital "H" or "F" respectively. If there is no significant preference yet a user asks for more hints than s/he has failures (or vice versa) then "h" or "f" is assigned.

The general classification was done manually using the following rule. If both feature and hypothesis classifications are "h" or "H", then the user is classified as a hint-driven learner and "H" is assigned to the class column. If both feature and hypothesis classifications are "f" or "F" then user is considered to be a failure-driven learner and "F" is a value in the class column. If neither of the two previous conditions was met, then the user could not be classified and a dash mark "-" was put in the class column. Out of 21 users, 4 were classified as hint-driven learners, 9 as failure-driven learners and 8 could not be classified as either one or another and were referred to as the "mixed group".

User	Features	Hypotheses	Class
1	H*	H	H
2	f	h	-
3	F	F	F
4	h	h	-
5	f	f	-
6	h	H	H
7	F	f	F
8	F	F	F
9	f	H	-
10	f	H	-
11	F	F	F
12	f	H	-
13	F	f	F
14	F	F	F
15	F	f	F
16	F	h	-
17	F	f	F
18	f	f	F
19	f	H	-
20	H	H	H
21	H	H	H

* H - hints are significantly preferred, F - failures are significantly preferred, f|h - no significant preference with a higher mean frequency of hints (h) or failures (f)

Table 1 Classification of users as hint-driven or failure-driven

We were mostly interested in "static" classification of users by learning strategy. Namely, what was the user's global learning style over the course of the 20 problems that s/he solved (pertaining to defining features or defining hypotheses or defining anything at all). However, we also noticed that learning behaviors tend to change as users progress through the case sequence.

Figure 3 shows an example of behavior stereotype for hypothesis identification averaged over 5 problems. Classification points are marked with an x. In addition the relative success rate (correct answers) is also shown as rectangles. We can see that the user starts as a marginally failure-driven learner (marked on y-axis as "f") and his/her success rate is about .3. Then the user "converts" to a strong hint-driven learner (marked as "H" on y-axis) and his/her success rates increases up to about .5. Finally the

Normalized Errors per goal/confusion features		Confusion Features																								
	Goal Features	amyloid	blister	elongate keratinocytes	eosinophil rich inflammatory infiltrate	epithelial necrosis	fibrin	fibrosis	homogenous material	hypergranulosis	isolated eosinophils	isolated lymphocytes	isolated neutrophils	mast cell rich inflammatory infiltrate	mucin	neutrophil rich inflammatory infiltrate	nuclear dust	papillae preserved	point-of-entry vesicle	predominantly lymphocytic inflammatory infiltrate	sclerosis	solar elastosis	thick collagen bundles	thrombi
1	amyloid																							
2	blister	0.00	0.14	0.02	0.00	0.02	0.00	0.01	0.00	0.02		0.01			0.01		0.01	0.01	0.01	0.03	0.01	0.02	0.02	0.00
3	elongate keratinocytes																							
4	eosinophil rich inflammatory infiltrate				0.15	0.02	0.02	0.02			0.09	0.02	0.02	0.02		0.03	0.02	0.01		0.19	0.01	0.02	0.03	0.01
5	epithelial necrosis	0.02		0.10		0.33	0.07	0.17	0.02	0.10	0.07	0.07	0.02		0.07			0.10	0.07		0.14	0.02	0.10	0.05
6	fibrin																							
7	fibrosis	0.02		0.10	0.01	0.10	0.06	0.30	0.10	0.13	0.02	0.05			0.10	0.05		0.08	0.05			0.07	0.06	0.02
8	homogenous material	0.05		0.05		0.07		0.10		0.02	0.02	0.05			0.14						0.07		0.21	
9	hypergranulosis																							
10	isolated eosinophils	0.05		0.02	0.07	0.12	0.02	0.31	0.02		0.17	0.10	0.05	0.02	0.07	0.07	0.07	0.12			0.14	0.05	0.07	0.02
11	isolated lymphocytes																							
12	isolated neutrophils																							
13	mast cell rich inflammatory infiltrate																							
14	mucin				0.05	0.11	0.04	0.06	0.01	0.01	0.15	0.08	0.07	0.02	0.16			0.02	0.01	0.42	0.02	0.02	0.04	0.05
15	neutrophil rich inflammatory infiltrate	0.01		0.01	0.05	0.07		0.03	0.01	0.01	0.08	0.05	0.05	0.02	0.01	0.17	0.05	0.02	0.02	0.36	0.02	0.02	0.02	
16	nuclear dust	0.01		0.01	0.05	0.12	0.01	0.07	0.01	0.01	0.12	0.07	0.08	0.03			0.10	0.03	0.02	0.40	0.03	0.03	0.05	0.03
17	papillae preserved			0.05		0.05		0.02		0.02		0.07			0.12			0.21					0.12	
18	point-of-entry vesicle																							
19	predominantly lymphocytic inflammatory infiltrate	0.02		0.04	0.03	0.02	0.01	0.06	0.00	0.05	0.02	0.05	0.00		0.05	0.05		0.05	0.02	0.11	0.03	0.06	0.05	0.01
20	sclerosis	0.07		0.10		0.10	0.05		0.21	0.21		0.05			0.21	0.05		0.10	0.10		0.26	0.17		
21	solar elastosis	0.02		0.02				0.05			0.02	0.05			0.10						0.05		0.10	
22	thick collagen bundles	0.05		0.14		0.14	0.07		0.19	0.24	0.02	0.10			0.26	0.05		0.17	0.12			0.12	0.24	0.02
23	thrombi	0.02			0.10	0.17	0.07	0.10	0.05		0.05	0.02	0.05	0.02	0.07		0.45	0.02		0.40	0.05	0.05	0.07	0.43

Table 2. Feature identification: subgoals vs. misconception confusions

user ends up as a marginally hint-driven learner (marked as "h" on y-axis) and the success rate goes up to .7.

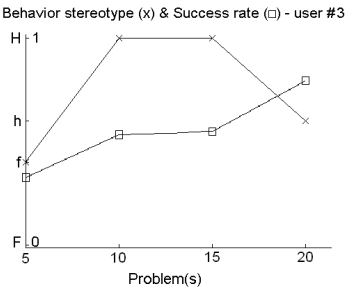

Figure 3. Behavior stereotype and success rate for groups of 5 problems of hypothesis identification for example user 3

3.3. Most Common Confusions

When a user is trying to define a subgoal in the problem (e.g. a feature on the slide) and makes a mistake, this mistake may be due to one of two reasons. First, a feature can be present on the slide but in a different location. Second, the feature is not on the slide and is not a subgoal of the problem. The former situation implies that the user does not know the subgoal well. The latter situation is of more interest to us since it means that the user might have mixed one subgoal with another. In this section we tried to mine for feature and hypothesis "confusion" pairs. The methods will be explained with features as an example. The results will be presented for both.

Feature and Hypothesis Confusion. A 4D data hypercube was constructed. The four dimensions of the cube were: user, problem, goal feature, and misconception feature. The data in the cells contained the number of errors a user made in a certain problem by specifying a misconception feature (that was not a goal of the problem), while some other feature was. The population of the hypercube was done in the following manner.

> If user u while solving problem p that has a set of goal features F_p, successfully defined a subset of goal features $F_p^* \subset F_p$. And after that s/he attempts to define feature f_e that is not among goals of problem p (f_e is a misconception): $f_e \notin F_p$. Then all goal features that were not successfully defined – $F_p \setminus F_p^*$ – are said to be confused with feature f_e, and $\forall f \in F_p \setminus F_p^*$ hypercube cell values $[u, p, f, f_e]$ will be increased.

For example: there are three goal features to be identified in a problem: A, B, and C. The user has already correctly identified feature A. Then user attempts to identify feature D, which is not in the problem (misconception feature). We say that the user confused feature D with features B, and C. Note that if feature A had not been defined, then we would conclude that the user confused feature D with all three features A, B, and C.

Although this blame assignment is not completely accurate, it is the best guess that we can make. Errors with identification of goal features were excluded from further analysis. In addition, values corresponding to feature "blister" were suppressed (set to zero) because blister was a goal feature of every problem.

After constructing the data cube, we calculated a 2D error plane that had goal features and misconception features as dimensions and a number of errors made by users as cell values. The errors in cells were normalized by the number of problems, in which features occurred as goals, and by the number of users that saw those features in problems. 2D plane is shown in Table 2. Cell values show how frequently a feature is present in the case (goal feature marked in the row header) is being confused with features that are absent from the case (misconception features, marked in the column header). For example the frequency of the user saying that the feature "amyloid" (column 1) is present while feature "fibrosis" (row 7) is present in a problem is equal to .02 (bear in mind that these frequencies are normalized).

To better understand the relations between features (when they are goals or misconceptions), we looked at mistakes with feature identification from two points: when the feature is a goal of the problem and when it is not. Although these two situations seem similar we found that confusions do not always happen both ways. Namely if a feature A, when being a goal of the problem, is often confused with some absent feature B, it does not necessarily mean that feature B, while not being a goal of the problem, is confused with goal feature A. What we are interested in are the "mutual" confusions.

To investigate feature confusions we have "cross-sliced" the 2D plane: we plotted rows that represent error rates of misconception features for each goal feature and combined them with plots of columns that represent goal feature error rates for each misconception feature. Thus, each feature is described from the point of its being a goal and misconception.

Figure 4 presents examples of a cross-slice plot and mutual confusion for features "epithelial necrosis", "fibrosis", and "thrombi". The x-axes of the plots denote 23 features. The positive parts of the cross-slices are shown as bars above the x-axis that denote error rates of confusion with other features when a certain feature is a goal (this feature is in the title of each cross-slice plot). The negative part of each cross-slice is shown as bars below the x-axis that denotes error rates of confusion with other features when a certain feature is a misconception.

In Figure 4 when feature "epithelial necrosis" (topmost plot) is a goal (bars above zero), users often confuse it with feature #7 "fibrosis" (positive bar against x=7 is higher on average). Conversely, when feature "fibrosis" (middle plot) is a misconception, it is often confused with feature #5 "epithelial necrosis" (negative bar against it is low on average). Features "epithelial necrosis" and "fibrosis" are mutually confused. However features "epithelial necrosis" and "thrombi" (bottom plot) are not mutually confused. When "thrombi" is a misconception, it is often confused with "epithelial necrosis", for feature "thrombi" negative bar against feature #5 "epithelial necrosis" is rather low. But when "epithelial necrosis" is a goal it is not often confused with "thrombi".

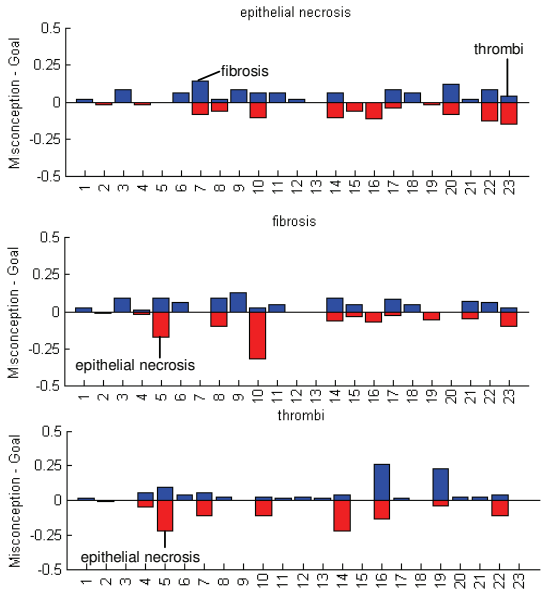

Figure 4. Cross-slice plots of features "epithelial necrosis", "fibrosis", and "thrombi".

Notice that in Figure 4 out of all positive and negative bars only 3 or 4 are significantly higher than the others. To bring out such significant confusions we reduced the 2D error plane (Table 2) to a smaller set of simple rules. The rules would be of form "feature X when a goal is confused with a misconception feature Y".

To do so, the 2D error plane was filtered and only significantly high confusion errors were taken. Rows and columns were considered separately. Rows represented goal features that were confused with misconception features, columns – misconception features when confused with goal features. Although it might seem that considering rows and columns separately is unnecessary, this separation gives us two asymmetric views on each feature.

First, when a feature is a goal in the problem, what are other features that are not in the problem that are more likely to be mentioned by users (because they are confused with this goal feature)? Second, when a feature is not a goal in the problem, what are the goal features that it would be confused with?

Mean plus N standard deviations filtering condition was used to select significant feature confusions. We used 1 and 2 as values for N. Namely, if an error in a row or a column is higher than a row/column mean + 1(2) standard deviations, then the error is significant. The results of the filtering are shown in Figure 5 as a bipartite graph. Features are shown twice as goals (left) and misconceptions (right).

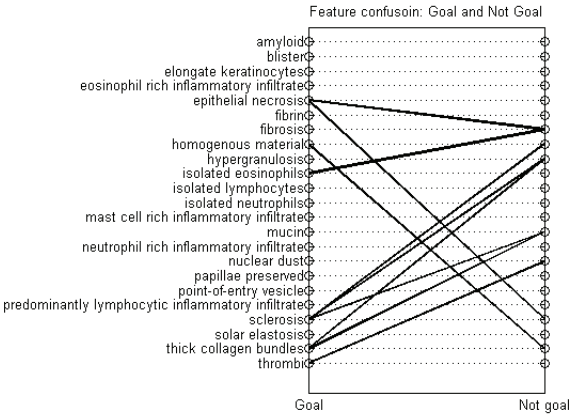

Figure 5. Feature confusion graph (mean + 1 standard deviations filtering)

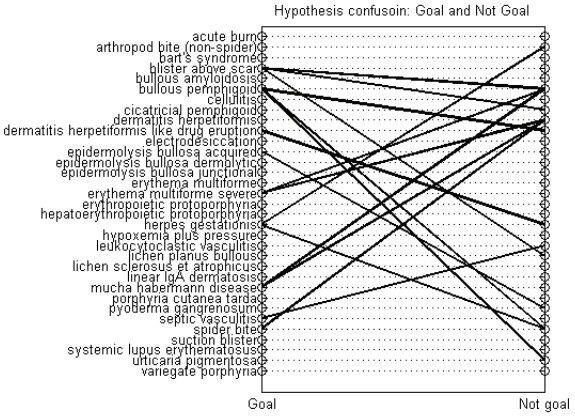

Figure 6. Hypothesis confusion graph (mean + 1 standard deviations filtering)

Figure 5 should be read as follows. If a goal feature A is significantly often confused with some other misconception feature B, plus if misconception feature B is significantly often confused with goal feature A, then there is a line connecting feature A on the left and feature B on the right. Figure 6 displays the same information for hypothesis confusion.

Feature and Hypothesis Confusion across User Stereotypes. Results of feature and hypothesis confusion shown above describe the user population as a whole. We have broken the overall confusion matrix into confusion graphs for each of the behavior stereotypes (hint-driven learners, failure-driven learners, and the mixed group). These per-stereotype confusion graphs for features and hypotheses are shown in Figure 7 and Figure 8 respectively. The leftmost graph on both figures denotes confusions of users learning from hints; the rightmost – the confusions of users learning from failures; and the middle one – the confusions of users who do not adopt any particular behavior pattern (the mixed group). Although the number of users in each of the behavior groups is different, the confusions are computed based on the data normalized by the number of users.

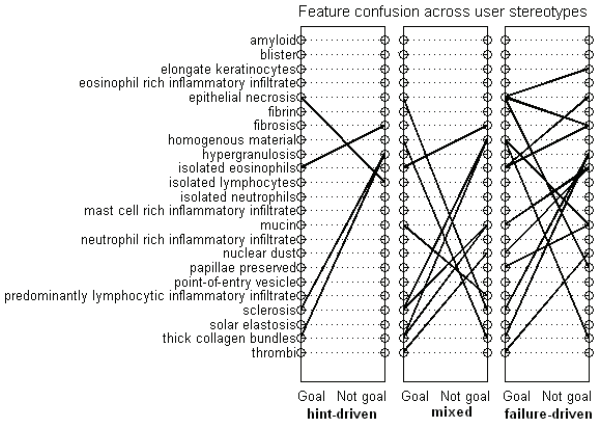

Figure 7. Feature confusion across user stereotypes

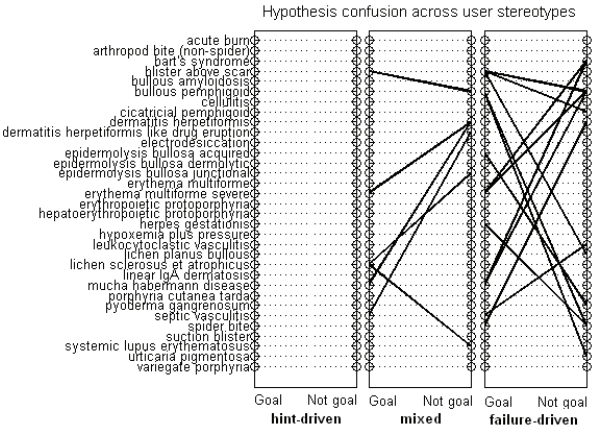

Figure 8. Hypothesis confusion across user stereotypes

3.4. Interesting dependencies in the data.

We also found another interesting pattern in the user data that is not related to the learning behavior stereotypes. The number of hints the user asks for while defining features was negatively correlated with the number of years s/he

has been in the postdoctoral medical (residency) training ($r = -.646$). This means that the further into residency a student is, the less s/he will rely upon hints while defining features. This can mean that as students received more training in dermatopathology they become more confident in his/her knowledge and do not rely on hints.

4. Discussion

SlideTutor is currently working within a space of a single problem and does not maintain a long-term model of a student. In this section we discuss how the obtained results related to common misconceptions and learning behaviors can be used to broaden the scope of SlideTutor, enable it to operate across multiple cases, and to make it capable of high-level pedagogic interventions that would enhance its pedagogic breadth.

4.1 Remedying Feature and Hypothesis Confusion

The most frequently confused among certain features or hypotheses are the places in the domain where a tutor should be especially expressive in order to articulate the difference of the mistaken knowledge of subgoals. When the tutor is confined to a current problem, it is impossible to explain these crucial differences.

In case of features, where the problem often is due to the similarity of the visual representation, we envision SlideTutor to take the student out of the current problem context. The tutor will have to pull different cases where the troublesome features are present and display the features visually side-by-side, helping the student to grasp the difference. This of course heavily relies on the availability of such cases. Specifically, mining of large data sets could be of enormous value in constructing representations of commonly confused visual features and "look-alike" diagnoses. These relationships could be mined across all users to inform interventions for individual users. As students spend time using the tutor, the same mechanisms could construct user-specific confusion models, which could be used to tailor high-level interventions to the student.

For hypotheses the situation is a little different. They require reasoning on the sets of features (and often corresponding feature attributes). When two hypotheses are confused, the tutor will have to indicate where the reasoning of the student failed and present a similar case, where a similar set of features leads to the same conclusion about hypotheses.

While accommodating for reduction of confusion errors, it is also important to maintain the balance of the feature representation in the case pool. After consulting an expert dermapathologist, we found that not all of the frequent confusions are due to failure to distinguish features or hypotheses.

Although some of the detected confusions did "make sense" from an expert point of view, some did not. For example, the feature "sclerosis" can be confused with the feature "homogenous material" (due to similar visual expression), but frequent confusion of the feature "thrombi" and the feature "nuclear dust" does not seem logical. In case of hypotheses, "blister above scar" can be confused with "bullous pemphigoid" and "cicatricial pemphigoid" (because these hypotheses have overlapping support feature sets), but "erythema multiforme" should not be confused with "dermatitis herpetiformis" (because there are no common features in their support feature sets). Among the possible reasons of the "illogical" confusions, we have selected the following:

- representation and case-authoring problem – the distribution areas of features on the slide sometimes overlap partially or in whole, which increases a chance of making an error;
- overgeneralization problem - students try to pick up a pattern of feature distribution among problems – e.g. if a feature A is seen in k consecutive problems, the student is more likely to say feature A is there when it's not; in general the distribution of features in cases is not uniform, which creates a problem of over-learning certain features at the cost of the others;
- knowledge base problem – hint messages prompting users were not always optimal.

4.2 Accommodating for Learning Behaviors

Since students learn differently, they should be treated differently as they learn in the tutoring system. If a student is learning by reviewing the tutor's hint messages (hint-driven user), the tutor should pay more attention to this student's hinting activity. For example, instead of gradual concretization of the hint information regarding a feature when hint message changes from "something is out there" to "at this location X is located", at a certain point a comparison to a similar situation from a different case should be drawn. In this case the learner will not only get what s/he is seeking, but also will reinforce the learned information with an additional example.

If a student is failure-driven and learns from corrective "bug" messages that follow his/her mistakes, the tutor should use examples from analogous cases to make a clear distinction about the error situation. In conjunction with checking for the most common misconceptions, this would help failure-driven students learn faster by exploiting the behavior they rely upon while learning.

Since it is not always possible to tell whether a user is a hint-driven or a failure-driven learner on the global scale, an "immediate" learning style can be used in order to choose the appropriate pedagogic intervention. The technique of maintaining both long-term and short-term information about the user's preferred learning behavior might be extremely beneficial.

4.3 "Gaming" Behaviors

Recent work on "gaming the system" is of significant importance in interpreting these findings. Gaming is a behavior pattern where a user is merely taking advantage of the system but does not actually learn anything (Baker et al, 2004).

We discovered gaming patterns in both hint-driven and failure-driven learners. The failure-driven students, for example, were trying to ease his/her job of placing a feature on the slide by first placing a pointer in some random location. Then, having received a bug message that s/he is trying to identify a correct feature but in the wrong location, s/he chaotically moved the pointer, trying to "hit" the correct area on the slide. This gaming pattern can be determined by looking at a student's slide exploration activity. If a student does not move the focus point across the slide and/or does not zoom in/out, but immediately attempts to pinpoint a feature – that might confirm gaming. Also, a frequency of consecutive bug messages about wrong location can reveal the same pattern.

In case of hint-abusive users, an indication of gaming will typically be a shorter time period between the requests for a more detailed hint. Hint abusers tend to quickly skip to the last most detailed message that contains direct instructions.

5. Conclusions and Future Work

The goal of the data mining we performed and described in this paper was to come from the fact that students do learn visual classification problems in the medical domain using SlideTutor to understanding how they actually learn. Although this paper is only the first step in that direction, it does reveal some important facts.

We observed different patterns users follow while using the system, as well as things common to all users. Given the results, we determined that our goals for future work are the following:

- alter the case-set of SlideTutor so that the features and hypotheses are represented more uniformly and closer to the frequency of occurrence in real-life medical practice (in order to prevent confusions due to the skewed frequency of occurrences in cases);
- continue investigation of the student learning patterns towards being able to differentiate patterns that are beneficial and those that are not (gaming patterns);
- build the behavior-mining component into the SlideTutor to make it capable of detecting and acting upon learning patterns and their changes on-the-fly;
- have information about the most common user misconceptions available to the tutor (and possibly allow the tutor to request information about such misconceptions to be re-mined either by a separate tool or a component in the tutor's architecture);
- make SlideTutor's pedagogic component adaptable to the situations when a learner's error is a common misconception or learner's behavior follows a certain pattern to provide a more effective tutoring;
- make SlideTutor capable of reaching across cases to draw comparative examples to reduce confusion errors, and to accommodate different learning strategies.

References

Clancey, W.J, and Letsinger, R. (1981) NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. *Proceedings of the Seventh Intl Joint Conf on AI*, Vancouver, BC. 1981; 829-835

Clancey, W.J. *Knowledge-Based Tutoring - The GUIDON Program*. Cambridge, MA: MIT Press, 1987

Clancey, W.J. *Heuristic Classification*. Artificial Intelligence 27:289-350, 1993

Crowley, R.S., and Medvedeva, O.P. (2003) A General Architecture for Intelligent Tutoring of Diagnostic Classification Problem Solving. *Proc AMIA Symp*, *2003*: 185-189.

Crowley, R.S., and Medvedeva, O.P. (2006) An intelligent tutoring system for visual classification problem solving. *Artif Intell Med*. 2006 Jan;36(1):85-117. Epub 2005 Aug 10. PMID: 16098717.

Crowley, R., Legowski, E., Medvedeva, O., Tseytlin, E., Roh, E., Jukic, D. (2005) An ITS for medical classification problem solving. Effects of tutoring and representations. *In: C.-K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.) Proceedings of 12th International Conference on Artificial Intelligence in Education, AIED 2005*, (Amsterdam, July 18-22, 2005). Amsterdam: IOS Press, pp. 192-199.

Baker, R.S., Corbett, A.T., Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 531-540.

Medvedeva, O., Chavan, G., Crowley, R.S. (2005) A data collection framework for capturing ITS data based on an agent communication standard. *Proceedings of the 20th Annual Meeting of the American Association for Artificial Intelligence, 2005*, Pittsburgh, PA.