ROUGH SET BASED DISTANCE LEARNING ALGORITHM AND ITS IMPLEMENTATION

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of
Master of Science
In
Computer Science
University of Regina

By
Aileen H. Liang
Regina, Saskatchewan
January 28, 2001

© Copyright 2001: Aileen H. Liang
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-60237-0
Abstract

With the growing popularity of the World Wide Web (WWW), Web-based distance education is increasingly popular in order to overcome various problems that central classroom teaching faces, such as distance, time scheduling, size limits, cost, and individual learning barriers. The WebCT system is an example of a web-based instruction tool that enables instructors to create and customize their courses for distance post-secondary education. A WebCT course allows students to do assignments, quizzes, and a final examination on the World Wide Web. If a student fails the final examination, then the student needs to study the course material again. Therefore, the performance of online students and the lack of contact and feedback between online students and the instructor, inherent to the course delivery mode, become growing concerns.

Inductive Learning is a research area in Artificial Intelligence. It has been used to model the knowledge of human experts by using a carefully chosen sample of expert decisions to infer decision rules. Rough Set based Inductive Learning uses Rough Set theory to compute decision rules.

The primary goal of the thesis is to investigate how to provide contact between students and teacher in distance education. In particular, we focus on WebCT education. We discuss how to use Rough Set theory in WebCT to permit decision rules to be induced that are important to both students and instructors. We propose the Rough Set Based Distance Learning Algorithm and describe its implementation using Java to make it more portable in a distance delivery environment.
Acknowledgments

I would like to take this opportunity to thank many people who made this work possible.

Many thanks go first to my supervisor, Dr. Brien Maguire, for his valuable advice and support throughout this research project. I am grateful to Dr. Maguire for providing valuable guidance during my research and employment experience.

I am very grateful for the academic and financial support given by the Faculty of Graduate Studies and Research and the Department of Computer Science.

My special thanks go to Dr. Yao, Dr. Ziarko, and Dr. Johnson for their valuable suggestions during my research. Thanks also to Dr. Adatia for his valuable comments.

Lastly, I would like to give my thanks to my beloved husband, Mengchi Liu, for his encouragement and support in my study.
## Contents

Abstract i
Acknowledgments ii
Table of Contents iii
List of Tables v
List of Figures vi

**Chapter 1  Introduction** 1

**Chapter 2  Distance Education** 5
  2.1 Distance Teaching ........................................ 6
  2.2 Distance Learning ........................................ 7

**Chapter 3  WebCT System** 9
  3.1 Introduction ............................................. 9
  3.2 WebCT Architecture ....................................... 10
  3.3 WebCT Course Design ................................... 11

**Chapter 4  Rough Sets and Inductive Learning** 23
  4.1 Rough Sets ............................................... 23
  4.1.1 Information System ................................... 23
  4.1.2 Knowledge Reduction ................................ 26
  4.1.3 Approximation Space ................................. 28
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Inductive Learning</td>
<td>30</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Inductive Learning Algorithm</td>
<td>30</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Decision Tree</td>
<td>31</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Rough Set based Distance Learning Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>Using RSDLA</td>
<td>36</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Implementation of RSDLA</td>
<td>44</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Conclusion</td>
<td>55</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>56</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Information Table ........................................... 24
4.2 Reduced Information Table .................................. 28
5.1 Student Information Database ............................... 36
5.2 Vertically Collapsed Student Information Table ............ 37
5.3 Horizontally Collapsed Student Information Table .......... 39
5.4 Further Collapsed Student Information Table ............... 42
6.1 Student Information Database ............................... 45
6.2 Collapsed Student Information Table ....................... 46
6.3 Vertically Collapsed Student Information Table ............ 48
6.4 Horizontally Collapsed Student Information Table .......... 50
6.5 Further Horizontally Collapsed Student Information Table ... 52
### List of Figures

3.1 Hierarchy of WebCT System from Students' Perspective .......................... 11
3.2 Hierarchical Structure of WebCT System ........................................... 12
3.3 WebCT Homepage ........................................................................... 13
3.4 WebCT Course Files ........................................................................ 14
3.5 WebCT Course Path Editor ............................................................... 15
3.6 WebCT Course Page Editor ............................................................... 16
3.7 WebCT Course Glossary ................................................................. 17
3.8 WebCT Course Reference ................................................................. 18
3.9 WebCT Course Question Editor ........................................................ 19
3.10 WebCT Course Student Management ................................................ 20
3.11 WebCT Course Quiz Design ............................................................ 21
4.1 Medical decision tree ........................................................................ 32
6.1 Decision Tree 1 ............................................................................... 50
6.2 Decision Tree 2 ............................................................................... 52
6.3 Web Learning Guidance .................................................................... 53
Chapter 1

Introduction

With the growing popularity of the World Wide Web, Web-based distance education is increasingly popular in order to overcome various problems that central classroom teaching has, such as distance, time scheduling, size limits, cost, and individual learning barriers.

In particular, a pure web-based teaching system called WebCT (Web Course Tools) has been developed at the University of British Columbia to enable instructors to create and customize their courses for distance post-secondary education [1, 17].

WebCT usage has grown rapidly as it is both a useful and inexpensive education tool. Its user base nearly doubled in size during the past year [7]. Today, there are more than 70,000 instructors teaching nearly 12 million online students in over 174,000 WebCT courses at 1,528 colleges and universities in more than 57 countries [7].

However, WebCT has a deficiency common to distance education delivery [13, 23]; that is, there is no way to accurately measure the achievement of students. The main problem is that students do not know what they do not know because there are no lectures given. If a student fails, he/she does not know where the problem is. If students need to repeat the course again, they do not know the sections on which to focus. Similarly, the course designer might not know whether the notes are clear or presented in the best possible order as there is no feedback from students other than the informal chat facility in WebCT.

Since the number of WebCT users is rapidly increasing, the performance of WebCT students is becoming an important issue. The self learning ability is important, but
if there is no guidance from the WebCT system, students may not obtain satisfactory results. This may dull the enthusiasm for WebCT. While online instruction is good, guided online instruction is better.

To solve the problem of lack of contact between students and instructor, some rules on reasons for failure are needed. Such rules can advise new students about the rules which apply to them, based on past course grade history. If students pass, these rules nevertheless advise them on their weak areas, so they can better prepare for future courses because passing does not mean one hundred percent comprehension of the course material. If students fail, these rules inform them of the areas in which they are weak, and suggest to them those sections they should focus on if they repeat the course. Student specific information is not as useful to students because it does not tell them which concepts (sections) are needed as prerequisites for other concepts (sections), as the student could have had a bad day on the day of the quiz.

In other words, the lack of contact problem can be solved by using inductive learning method to discover knowledge from the course grade history.

Rough Set theory, introduced by Zdzislaw Pawlak, is a mathematical tool for dealing with vagueness and uncertainty [30, 36, 37, 38, 51]. Vagueness is caused by the ambiguity of exact meaning of terms used in a knowledge domain, uncertainty in data, or in knowledge itself. To deal with vagueness, normally statistics are used for handling likelihood. The advantage of rough set theory is that it does not need any preliminary or additional information about data (like probability in statistics, grade of membership, or the value of possibility in fuzzy set theory). Another advantage of the rough set approach is its easy of use and its simple algorithms.

When we represent concepts as sets of examples, we can use Rough Set formalism to generate generalized descriptions of the given concepts. This is in fact the goal of an inductive learning system and has been illustrated by Wong and Wong [47], Wong and Ziarko [48], Wong, Ziarko, and Ye [49], and Hadjimichael and Wasilewska [22]. Rough Set theory has the ability to approximate a set $C$ using two other sets: a set of objects definitely in $C$ and a set of objects which may be in $C$. This approach lets us then give a more general description of the set $C$. See Chapter 4 for
details. Rough Set theory has been applied to knowledge discovery in various domains [15, 22, 24, 32, 42, 44, 45]. In [44], a rule induction method is introduced, which extracts not only classification rules but also other medical knowledge needed for diagnosis. It was evaluated on three clinical databases. Experimental results show that the authors’ proposed method correctly induces diagnostic rules and estimates the statistical measures of rules. In [44], the authors again explain the problem in a medical context, that is, when a patient suffers from several diseases and has complicated symptoms, making a differential diagnosis is very difficult. Therefore, three models for reasoning about complications are introduced and modeled by using characterization and a rough set model.

By using Rough Set-based inductive learning in WebCT education, we can generate the rules needed by both students and instructors. Students are informed of the reasons in general for success or failure. A rule is one such as “if a student failed the final examination, then there is a 70% chance that he/she failed Quiz 6” (suppose it is the last quiz). Special cases can be deduced as well. For example, if a student failed Quizzes 1 and 6 and also the final exam, then the student may have failed Quiz 1 because he had a bad day. In fact, Quiz 1 is not the background material needed to pass the final. Quiz 1 may simply be introductory material. Finally, rules help the instructor to reorganize the material to provide a better prerequisite ordering of the material.

The primary goal of the thesis is to investigate how to offset the loss of feedback information not available because of the lack of contact between students and teacher in distance education. In particular, we focus on WebCT. We discuss how to use Rough Set theory in WebCT to permit decision rules to be induced that are important to both students and instructors. The discovered rules can guide students in their learning. For repeating students, they provide the reasons for past failure in the course. For new students, the rules indicate which sections need extra effort in order to pass the course, alerting the students to the core sections of the course. These rules help the instructor to realize the problematic areas and better organize the notes to explain concepts more clearly and offer more examples where needed.

This thesis is organized into seven chapters. The Distance Education issue is
discussed in Chapter 2. Chapter 3 focuses on the WebCT System and how to design a WebCT course. Rough Set theory and Inductive Learning are discussed in Chapter 4. The Distance Learning Algorithm and its usage are presented in Chapter 5. In Chapter 6, the implementation details of RSDL A are discussed. The last chapter offers conclusions and points out possible future development.

Parts of Chapters 1, 3, and 5 are from the paper "Rough Set based WebCT Learning" [26]. Parts of Chapters 2, 4 and 5 are from the paper "Using Rough Sets in Distance Education" [25]. Parts of Chapters 2, 4 and 6 are from the paper "Java Implementation of Distance Learning Algorithm for Web-Based Course Delivery" [27].
Chapter 2

Distance Education

As the World Wide Web becomes more and more advanced, applications of World Wide Web based technology to support distance education and computer aided learning are increasing rapidly. Distance Education contains two parts: Distance Teaching and Distance Learning [28, 11]. The definition of Distance Education has two forms [16]. One definition that is commonly used in Europe, Asia, and Australia is that distance education is the offering of educational programs designed to facilitate a learning strategy which does not depend on day-to-day contact teaching, but makes use of the potential of students to study on their own. It provides interactive study materials and decentralized learning facilities where students can seek academic and other forms of educational assistance when they need it. The definition commonly used in United States is that distance education links the teacher and students in several geographic locations via technology that allows for interaction. This suggests a remote classroom model with a camera in one classroom and the teaching transmitted to the remote classrooms, thus allowing for synchronous communication. Whatever the form or emphasis found in a definition of distance education, the separation of student and instructor is a key factor. From the simple correspondence course, to broadcast TV with reverse audio or to specialized video-conferencing tools, such as Proshare [2], PictureTel [3], Polycom [4], Trinicom [5], or Zydacron [6], and web-based courses [29], distance education has helped many people obtain college credits, complete training, or update knowledge to adapt to the new information society by eliminating the need for instructor and students to meet in a classroom.
2.1 Distance Teaching

Teaching distance education courses is not an easy task. It involves psychological issues because the instructor not only writes the online lectures, but also has to consider how the online learner will react to what is presented [8, 31, 33, 40, 39]. There are several learning factors that the instructor should be aware of when preparing online lectures. These includes interactivity, learner adaptability, situated learning, and retention [12].

Interactivity is a set of measurable instructional qualities such as “time on task,” and “immediacy of feedback.” The instructor needs to remember that every new piece of data – a new concept, an unfamiliar word, or a new graphic symbol – causes the same effect in the learner. It is something which must be added to the learner’s set of knowledge. No matter how open the learner’s mind is to new ideas, the act of integrating new information into the reader’s set of knowledge requires mental effort, and if the effort of integrating too many new things overwhelms the learner, the learning process breaks down. Therefore, the amount of new information that a learner must handle at any one time should be kept both interesting and to a minimum [33].

The learner adaptability of the learning system to the individual student is another instructional concern in distance teaching. In human terms, this adaptability comes in the form of a tutor. Computer systems that emulate the methods of human tutors are intelligent tutoring systems. Due to the nature of distance learning over the web, the learner can move around as he or she feels fit. Therefore, the instructor or course designer should make the system easy for the learner to navigate. Simple, obvious, sensible navigation elements will keep the learner from becoming too frustrated with the interface, and, therefore, with the content itself.

Situated learning is an approach that mimics the circumstances in which the learners are expected to use their knowledge. This mimicking can involve role playing or any other teaching device that will reconstruct an experience similar to “real life.” Moot court, simulation, and role playing are all common teaching methods for situating the learning experience.
Retention of information is the ability to recall the knowledge on demand after instruction. Anything learned decays as time passes. However, the more practice a student has in a realistic setting, the more likely the knowledge is to "stay" in the learner's mind. An optimal instructional system would provide highly interactive, realistic (situated) training, preferably in a time frame very close to its actual use in a job or on a project. Further, the optimal training system would be readily accessible, so that any forgotten information or skill training could be revisited on demand. Using Hypertext over the Web to link definitions or new terms is one such example.

For web-based teaching, powerful tools can be used to enhance the presentation of teaching materials. These include HTML documents, images in GIF and JPEG format, animation in Java applets and embedded JavaScript.

2.2 Distance Learning

Successful learning takes place in all cultures. Many learning theories and models have been proposed: models based on human biology, psychology, sociology, and educational theory [14]. The role of the teacher is significant in these models, but distance learning is very different from these models. Unlike classroom learning, distance education has no face to face contact. Consequently there is no face to face body language and feedback communication between students and teacher. Students miss the range of hints typically offered by instructors as to what is important. Instructors miss the opportunity to gauge where the material is being understood. Students learn from the teacher through the Web. In this model, the Web presents structured information, and also provides the communication medium for the necessary interaction.

For web-based learning, the Web is the online student's source of information, electronic book, teacher, and the communication medium between teacher and student [14]. Firstly, any supporting information is stored on the Web. Students read/print the online notes, find assignment specifications, do quizzes, take exams, and search for important elements of the course. They also use the Web to access sites related to the course.
Secondly, the web is often an electronic replacement for the standard course textbook. Nowadays, many institutions, although they are not dedicated to Distance Education, have used the Web to present information in a more structured way. Therefore, it is well accepted that navigation of structured information on the Web site improves the learning process. In distance education, online students follow screen instructions to read material, activate multimedia demonstrations, take self-correcting quizzes or other activities. Web sites are their most useful electronic book.

Thirdly, the web is the online students’ teacher. Students may never see the course instructor while taking an online course, but they believe that the instructor is always there. He or she has prepared the course material for them and is responsible for this class. The instructor is the person who cares about their studies and is willing to help them whenever they have problems. For example, some web sites allow students to answer questions on Web pages, the instructor checks the pages periodically and marks answers whenever they see signs of problems in the students’ understanding of the material.

Finally, the Web is used as a communication medium between instructor and students. When students have problems and search for help, the course instructor is the first one that students contact. Therefore, they ask questions through the Web. Communication between students and teacher and between students and other students is through email, chat rooms, and shared online work space.

Thus, there are many positive aspects to distance delivery of instructional material over the Web. However, there is still the problem of lack of face to face communication. Before discussing how Rough Set Inductive Learning can assist in offsetting this problem, we’ll examine in more detail WebCT, an example of a popular Web based distance delivery tool.
Chapter 3

WebCT System

3.1 Introduction

WebCT stands for Web Course Tools. Developed at the University of British Columbia, it is a web-based instructional delivery tool that enables instructors to create and customize their courses for distance post-secondary education [1, 17].

WebCT provides a great deal of flexibility for students, instructors, and teaching assistants to access WebCT software without the need to download or install any kind of additional software. WebCT provides a set of educational tools to facilitate learning, communication, and collaboration. For instructors, it provides a set of administrative tools to assist them in the process of management and continuous improvement of the course. For students, it provides a set of organized navigatable pages which guide students in their learning.

The popularity of WebCT is due to the following reasons: WebCT solves students' scheduling conflicts. It allows beginners to work and practice at their own pace. Students can be accepted from different cities or countries and there is no limit on the number of students. Also, people can easily update their knowledge from home by using a web browser. Plus, the cost to the educational institute is low. For example, it only costs $4 to register a WebCT course account at the University of Regina. WebCT system now has a consistent user interface, ease of navigation, and several tools to help new users get started. Another key factor fueling WebCT's growth is the interaction between students and faculty that has been enhanced by
3.2 WebCT Architecture

The WebCT system uses a client/server architecture, which is a computing model for the development of computerized systems. This model is based on the distribution of functions between two types of independent and autonomous processes: client and server. The client normally interacts with the user and requests specific services from the server on behalf of the user. The server provides requested services for clients. Both clients and servers can reside in the same computer or in different computers connected by a network. The WebCT system consists of WebCT software and a number of hierarchically organized files for each course. The user accesses the data on the server through a web browser. All the WebCT software resides on and runs off a WebCT server, which means any changes made to courses are accessible to students immediately after the change is made. It provides an environment to cover all aspects of a course such as tools for creating the course materials, lectures, exercises, quizzes, lab materials, discussion groups, and reference materials and it is password protected.

Using WebCT, the author has designed a Java language course at the University of Regina. Figure 3.1 presents the hierarchy of the WebCT system from the student's point of view based in this WebCT course. When students enter their user ID and password, they come to a homepage that contains six hyperlinks. The first is the Course Content link that allows students to access the notes organized by the course

links from WebCT, the course management platform, to WebCT.com, the e-Learning hub. The company expanded the number of links between WebCT and WebCT.com, advancing the vision of a truly integrated e-Learning Environment that helps address the new set of challenges inherent in making education relevant to a student pursuing lifelong learning. Also helping to fuel the adoption of WebCT's integrated e-learning environment was the availability of quality course material from the company's broad spectrum of content partners. Today, over 20 different content providers offer more than 500 e-Learning Resource Packs that can help faculty jumpstart teaching WebCT courses with robust online course materials.
designer. The second is the Bulletin Board link that is used to post any class information such as assignment or quiz information. Both the instructor and students can access the Bulletin Board. The third hyperlink, Mail, allows private email to be sent between student and student and between student and instructor. The fourth hyperlink, FAQ, contains frequently asked questions from students. The next link, Calendar, marks special events for the course. The last one, Exams, links to the quizzes and final examination.

Figure 3.2 presents the internal hierarchical structure of the WebCT system that is visible only to the course designer. The course designer creates course materials and course settings by manipulating the hyperlinks in the diagram.

3.3 WebCT Course Design

WebCT has a default homepage which the course designer can modify to suit the particular course needs. The default homepage usually contains two frames. The
Figure 3.2: Hierarchical Structure of WebCT System

top frame contains the student’s view of the home page. The bottom frame contains the designer toolbar, which gives the course designer access to the options for each page. The top frame has five parts. The first part is called Banner. It is used as a graphical identifier for a course and can contain an image (GIF or JPEG format) and text. The second part is Header which is placed between the banner and the course contents. It is formatted in HTML and can include images. Course Content, the third part, contains the links to the various sections of the course. The fifth part is Counter. The counter keeps track of the number of accesses to the homepage. The bottom frame has only one part that is called the Designer Toolbar, which is only visible to the course designer when the course designer is connected. The Designer Toolbar gives the course designer access to the options for each page. The options are file management, path editor, page editor, glossary editing, reference editor, question
editor, student management, designing quiz, and bulletin board. Figure 3.3 shows the modified homepage for the Java programming language course.

*File Management:* When creating a course on the WebCT server, a directory is also created into which the course designer places the files that make up the course. The File Manager is the tool which gives the designer access to that directory and with which he or she can manipulate the files. The File manager allows the designer to upload files from a local computer to the WebCT server, to use ZIP compression of
multiple files for easy transfer of large numbers of files, and to browse WebCT built-in files and images. File Management includes the standard file operations such as creating, copying, moving, and deleting files and directories. The designer also can edit text and HTML files directly on the server.

There are two types of files: WebCT Build-in Files, organized beneath the WebCT Built-in Files directory and Self-Build Files, which make up the course and are organized beneath the Course Files directory. To make the course clear and concise, subdirectories are usually used to organize these files.
Path Editor: The WebCT Path Editor allows the designer to create, organize, and order the sequence of content pages which make up the various sections of the course. For example, the author’s Java course, consists of more than 40 individual HTML pages. The Path Editor helps to lay out the table of contents for these pages. To reduce the cluster in the table of contents, the subtopics can be shown or hidden.

Page Editor: The Page editor modifies and enhances the pages contained in the
As personal computers and the World Wide Web become more and more popular, people have a chance to view a vast variety of information through web browsers (the two most popular being Netscape Navigator or Microsoft Internet Explorer).

Figure 3.6: WebCT Course Page Editor

paths. It allows the designer to edit directly the HTML or text of a page, modify the appearance of a page by changing the color scheme background, link words in the text of the page to the glossary, create self-test questions for students to test their knowledge of the current topic, add entries for the current page into the course index, create references to additional information, such as WWW sites, course text books or journal articles, define learning goals for the current topic, add an access counter to record the number of visits to the page, and add links to WebCT tools to the button.
The Glossary tool, students on the homepage, on a toolpage or in the glossary course content. The designer are possible.

**Glossary**

- **applet**: A program that can be downloaded across the World Wide Web and run as part of a page displayed by a browser such as Netscape Navigator or Microsoft Internet Explorer.
- **array**: A mechanism used to hold a collection of elements all of the same type. It starts from 0.
- **boolean**: A variable of this type can hold only one of the two values: true or false.
- **char**: A character is a datatype in Java which can hold a single character such as "a" or "3". Notice that we can put a numeric field in a char datatype as long as it is in single or double quotes.
- **constructor**: A Java built-in method used to set up an instance.
- **float**: A variable of this type can hold a number with a fractional or decimal part like 3.7 or -23.45.
- **identifier**: An identifier is a variable name that can be any combination of letters, digits, -, or $ of any length, but it cannot contain #, $, or start with digits.
- **integer**: A variable that is a whole number that doesn’t contain fractions or decimal places and is in the range of -2,147,483,648 to 2,147,483,647.
- **length**: A Java built-in method which returns the length of a previously defined array. This method has to be used with the array name. For example, names.length where names is the array’s name.
- **linked list**: A data structure that contains a data field and a pointer to the next element in the link. This data structure is often favorable over an array because it is easier to insert elements anywhere within the linked list. As well, a linked list is dynamic rather than static, which means that the number of elements does not have to be explicitly coded.
- **overloading**: A mechanism that allows a method to be used in different circumstances. For example, we could declare two methods with the same name but with different parameters. As long as we declare the methods to have different parameters, we can use both method names.
- **package**: A mechanism that allows a programmer to group related classes together and give this group a unique name. The programmer can then use any method her/his defined in the packages.
- **queue**: A queue is a structure that simulates waiting. There are three types of queues: first in first out, first in last out and priority.
- **string**: The string data type is not a basic data type but rather a class. However,

![View Glossary](image)

**Figure 3.7: WebCT Course Glossary**

bar.

**Glossary Editing**: A course’s glossary gathers all the important definitions of the course content. The designer can create a fully-searchable glossary for the course. The glossary can contain images as well as text, so that illustrative glossary definitions are possible.

A course glossary can be accessed from the “Glossary” tool which can be placed on the homepage, on a toolpage or in the button bar of individual pages. From the Glossary tool, students can list glossary entries by starting letter, list the whole
glossary, or search the glossary for a particular keyword.

The course glossary can also be accessed with individual glossary term definitions that have hyper-links to words in the body of the HTML pages on the various paths which make up the course. When the student clicks on one of these hyper-links, he or she will be shown the definition for the particular word that was selected.

*Reference Editor:* The Reference editor allows the designer to associate the course content with external references so that students can find supplementary information on the current topic. The external references can be any of three types: reference to
Multiple Choice Questions

Select a question or answer:

<table>
<thead>
<tr>
<th>#</th>
<th>Question</th>
<th>Answer 1</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A typical declaration of an array would be.</td>
<td>int(5) years = new years[5];</td>
<td>Wrong, the integer 5 should be placed after declaring the space (new).</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>int(1) years - new years[5];</td>
<td>Correct</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>int(1) years - new years[5];</td>
<td>Wrong, it doesn’t declare the space for the int array.</td>
</tr>
</tbody>
</table>

Figure 3.9: WebCT Course Question Editor

textbooks, reference to articles, and reference to URLs.

Question Editor: The question editor is used to generate self-test questions which are associated with the content of current page. These self-tests are automatically graded by WebCT, but no marks are recorded for the students - the goal of the self-tests is to reinforce the ideas presented on the current page.

Student Management: Student management creates and keeps track of student accounts. The designer can add or remove student accounts, allow or disallow Guest
account creation, create an arbitrary number of columns to store student data, and can easily upload student records from other file formats such as a spreadsheet.

*Designing Quiz:* The WebCT homepage icon “Online Quizzes and Surveys” is the page where designer can design a quiz. There are 3 kinds of quiz formats, filling blanks, matching, and comprehension. The first two kinds are automatically marked by the system and the third one has to be marked by the online teacher.

*Bulletin Board:* The bulletin board is one of WebCT’s main communication tools. It
allows students and instructors to compose messages and electronically post them for others to read and reply to if they wish.

Since the performance of online students is our major research issue, quizzes and final examinations are the main focus here. The following discussion provides detailed information on the quizzes and final examination of this course (note: this information can be varied from instructor to instructor). There are six quizzes, one final examination, and twenty section notes in this Java course. The first section introduces the Java language. The second section teaches students the editor that is used to write Java programs. The third starts to introduce simple Java programs. The fourth is about variables and data types. The fifth talks about the selection statement. The sixth is about the String class. The seventh deals with repetition statements. The eighth and ninth introduce one dimensional and two dimensional
arrays respectively. The tenth talks about file access. The eleventh introduces objects and classes. The twelfth is about Information Hiding. The thirteenth deals with objects within objects. The fourteenth is about Queues. The fifteenth introduces Packages. The sixteenth is about Linked Lists. The seventeenth introduces Inheritance. The eighteenth is about Polymorphism. The nineteenth introduces Applets and the last is about the Graphical User Interface. There is a quiz after every three or four sections, depending on how each section is related to the others. For example, the materials in Quiz 6 are from section seventeen to twenty covering Inheritance, Polymorphism, Applets, and the Graphical User Interface. These are the core sections in this class.

This WebCT course provided the initial motivation for the use of Rough Set Inductive Learning to provide more information to students about the key components of the course as presented in the WebCT framework. In the next chapter, we discuss the details of Rough Sets and Inductive Learning.
4.1 Rough Sets

Rough Set theory was introduced by Pawlak in the early 1980s [30, 36, 37, 38, 51]. The intuition behind Rough Set theory is as follows. When dealing with sets of individuals, we may not have the means to distinguish individual set elements. The elements may possess some measurable characteristics but we can only distinguish classes of elements rather than individuals because of the limited resolution of our perception mechanism. Thus, elements within classes are indistinguishable. Rough Set theory provides a collection of methods which can be applied to the analysis of the individuals. The methods are particularly applicable to knowledge discovery problems. In this section, we introduce the principal notions of Rough Set theory and Inductive Learning.

4.1.1 Information System

An information system is a tuple $I = (\mathcal{O}, \mathcal{A}, \mathcal{V}, f)$, where

- $\mathcal{O} = \{o_1, ..., o_n\}$ is a finite, non-empty set of objects;

- $\mathcal{A}$ is a finite set of attributes that is further classified into two disjoint subsets, condition attributes $\mathcal{C}$ and decision attributes $\mathcal{D}$; that is, $\mathcal{A} = \mathcal{C} \cup \mathcal{D}$;
An information system can be viewed as a table (or relation) in a relational database, in which columns are labeled by attributes, rows are labeled by the objects and the entry in column $A$ and row $o$ has the value $f(A, o)$. Each row in the table represents information or knowledge about some object in $O$. Some attributes are identified as decision attributes, and the other attributes are the condition attributes. The objects do not need to be distinguished by their attributes or by their relationship to other objects.

**Example 4.1.1** Consider the information table in Table 4.1. It represents an information system $I = (O, A, V, f)$, where

- $O = \{e_1, e_2, ..., e_8\}$ is the set of objects.
- $A = \{Headache, Pain, Temp, Flu\}$ is the set of attributes.
- $C = \{Headache, Pain, Temp\}$ is the set of condition attributes.
- $D = \{Flu\}$ is the set of decision attributes.
Let $\text{Ind} \subseteq A$. We define a binary relation $R_{\text{Ind}}$ based on $\text{Ind}$, called an indiscernibility relation, as follows:

$$R_{\text{Ind}} = \{(o_i, o_j) \in O \times O | \forall A \in \text{Ind}, f(A, o_i) = f(A, o_j)\}$$

Informally, $o_i$ and $o_j$ are indiscernible by a set of attributes $\text{Ind}$ if and only if they have the same attribute values.

For example, for the information system in Example 4.1.1, let $\text{Ind} = \{\text{Headache}\}$. Then

$$R_{\text{Headache}} = \{(e_1, e_1), (e_2, e_2), (e_3, e_3), (e_4, e_4), (e_5, e_5), (e_6, e_6), (e_7, e_7), (e_8, e_8),$$

$$(e_1, e_2), (e_1, e_3), (e_2, e_3), (e_4, e_5), (e_4, e_6), (e_4, e_7),$$

$$(e_4, e_8), (e_5, e_6), (e_5, e_7), (e_5, e_8), (e_6, e_7), (e_6, e_8), (e_7, e_8),$$

$$(e_2, e_1), (e_3, e_1), (e_3, e_2), (e_5, e_4), (e_6, e_4), (e_7, e_4),$$

$$(e_8, e_4), (e_6, e_5), (e_7, e_5), (e_8, e_5), (e_7, e_6), (e_8, e_6), (e_8, e_7)\}$$

One can check that $R_{\text{Ind}}$ is an equivalence relation on $O$, which therefore partitions $O$ into equivalence classes. We also use $R_{\text{Ind}}$ to denote the family of the equivalence classes of the relation $R_{\text{Ind}}$. We call equivalent classes of the relation elementary sets in $O$. For any element $o_i$, the equivalence class to which $o_i$ belongs is represented as $[o_i]_{R_{\text{Ind}}}$. Let $Y$ be an equivalence class in $R_{\text{Ind}}$. We use $\text{Des}_{R_{\text{Ind}}}(Y)$ to denote the description; that is, the set of attribute values, of the equivalent class $Y$. Each equivalence class based on decision attributes is called a concept.

Consider the information system in Example 4.1.1 again. We have the following equivalence class families:

$$R_{\text{Headache}} = \{\{e_1, e_2, e_3\}, \{e_4, e_5, e_6, e_7, e_8\}\}$$

$$R_{\text{Pain}} = \{\{e_1, e_2, e_3, e_4, e_6, e_8\}, \{e_5, e_7\}\}$$
\( R_{\text{Temp}} = \{ \{ e_1, e_4 \}, \{ e_2, e_5, e_7 \}, \{ e_3, e_6, e_8 \} \} \)
\( R_{\text{Flu}} = \{ \{ e_1, e_4, e_5 \}, \{ e_2, e_3, e_6, e_7, e_8 \} \} \)
\( R_{\text{Headache,Pain}} = \{ \{ e_1, e_2, e_3 \}, \{ e_4, e_6, e_8 \}, \{ e_5, e_7 \} \} \)
\( R_{\text{Headache,Temp}} = \{ \{ e_1 \}, \{ c_2 \}, \{ e_3 \}, \{ e_4 \}, \{ e_5, e_7 \}, \{ e_6, e_8 \} \} \)
\( R_{\text{Pain,Temp}} = \{ \{ e_1, e_4 \}, \{ e_2 \}, \{ e_3, e_6, e_8 \}, \{ e_5, e_7 \} \} \)
\( R_{\text{Headache,Pain,Temp}} = \{ \{ e_1 \}, \{ e_2 \}, \{ e_3 \}, \{ e_4 \}, \{ e_5, e_7 \}, \{ e_6, e_8 \} \} \)

\[
\begin{align*}
[e_1]_{R_{\text{Flu}}} &= [e_4]_{R_{\text{Flu}}} = [e_5]_{R_{\text{Flu}}} = \{ e_1, e_4, e_5 \} \in R_{\text{Flu}} \\
[e_2]_{R_{\text{Flu}}} &= [e_3]_{R_{\text{Flu}}} = [e_6]_{R_{\text{Flu}}} = [e_7]_{R_{\text{Flu}}} = [e_8]_{R_{\text{Flu}}} = \{ e_2, e_3, e_6, e_7, e_8 \} \in R_{\text{Flu}}
\end{align*}
\]

\( \text{Desc}_{R_{\text{Headache,Pain,Temp}}} (\{ e_1 \}) = \{ \text{Headache} = \text{yes}, \text{Pain} = \text{yes}, \text{Temp} = \text{normal} \} \)
\( \text{Desc}_{R_{\text{Headache,Pain,Temp}}} (\{ e_5, e_7 \}) = \{ \text{Headache} = \text{no}, \text{Pain} = \text{no}, \text{Temp} = \text{high} \} \)
\( \text{Desc}_{R_{\text{Flu}}} (\{ e_1, e_4, e_5 \}) = \{ \text{Flu} = \text{no} \} \)
\( \text{Desc}_{R_{\text{Flu}}} (\{ e_2, e_3, e_6, e_7, e_8 \}) = \{ \text{Flu} = \text{yes} \} \)

The set \( \{ e_1, e_4, e_5 \} \in R_{\text{Flu}} \) defines a concept well people for people who have no flu, whereas the set \( \{ e_2, e_3, e_6, e_7, e_8 \} \in R_{\text{Flu}} \) defines a concept sick people for people who have flu.

An information system provides information or knowledge about real-world objects. Objects are characterized by selected features represented by the condition attributes. However, information about objects may not be sufficient to characterize objects without ambiguity because some objects are characterized by the same condition attribute values. Two objects are indiscernible whenever they have the same values for all the attributes under consideration.

Continue with the above example. Objects \( e_5 \) and \( e_7 \) are indiscernible with respect to all condition attributes Headache, Pain, and Temp. So are \( e_6 \) and \( e_8 \).

### 4.1.2 Knowledge Reduction

In an information system, all of the knowledge is not always necessary to define some categories available in the knowledge considered. In other words, not all conditional attributes are necessary to categorize the objects in the information system. Some attributes may be redundant or dispensable with respect to the decision
Let $\mathcal{R}_C$ and $\mathcal{R}_D$ be families of equivalence classes. The \textit{C positive region} of $D$, denoted by $POS_C(D)$ is the set:

$$POS_C(D) = \bigcup_{X \in \mathcal{R}_D} Y \in \mathcal{R}_C \text{ where } Y \subseteq X$$

Consider the families of equivalence classes in the last subsection, we have

$$POS_{Headache}(Flu) = \emptyset$$
$$POS_{Pain}(Flu) = \emptyset$$
$$POS_{Temp}(Flu) = \{e_1, e_4\} \cup \{e_3, e_6, e_8\} = \{e_1, e_3, e_4, e_6, e_8\}$$
$$POS_{Headache,Pain}(Flu) = \emptyset$$
$$POS_{Headache,Temp}(Flu) = \{e_1\} \cup \{e_2\} \cup \{e_3\} \cup \{e_4\} = \{e_1, e_2, e_3, e_4, e_6, e_8\}$$
$$POS_{Pain,Temp}(Flu) = \{e_1, e_4\} \cup \{e_2\} \cup \{e_3, e_6, e_8\} = \{e_1, e_2, e_3, e_4, e_6, e_8\}$$
$$POS_{Headache,Pain,Temp}(Flu) = \{e_1\} \cup \{e_2\} \cup \{e_3\} \cup \{e_4\} = \{e_1, e_2, e_3, e_4, e_6, e_8\}$$

Let $\mathcal{R}_C$ and $\mathcal{R}_D$ be families of equivalence classes. We say that $A \in C$ is $D$-dispensable in $C$ if the following holds:

$$POS_{C-(A)}(D) = POS_C(D)$$

Informally, if a set of attributes and its superset define the same indiscernibility relation, then any attribute that belongs to the superset, but not the subset, is redundant.

Consider the families of the equivalence class $\mathcal{R}_{Headache,Pain,Temp}$ and $\mathcal{R}_{Headache,Pain}$. Since we have $POS_{Headache,Pain,Temp}(Flu) = POS_{Headache,Temp}(Flu) = \{e_1, e_2, e_3, e_4\}$, the attribute Pain is redundant with respect to Flu, whereas the set \{Headache, Temp\} does not contain any redundant attributes.

For an information system, if a subset of condition attributes has no redundant attributes, then it is said to be \textit{minimal}. A \textit{reduced} information table is obtained by projecting on the minimal subset of condition attributes and decision attributes.

For the information table in Table 4.1, the reduced information table is shown in Table 4.2.
The rough set philosophy is based on the idea of classification. The most important issue addressed in Rough Set theory is the idea of imprecise knowledge. The imprecise knowledge containing imprecise concepts can be defined approximately by employing two precise concepts called lower and upper approximation. The lower approximation of a concept consists of all objects which surely belong to the concept whereas the upper approximation of the concept consists of all objects which possibly belong to the concept in question. The difference between the lower and upper approximation is a boundary region of the concept and consists of all objects which cannot be classified with certainty to the concept or its complement employing available knowledge.

Let $Y \subseteq C$ be a concept and $Ind \subseteq C$. The lower approximation of $Y$ based on $R_{Ind}$ is defined as

$$Y = \bigcup_{[o_i]_{R_{Ind}} \subseteq Y} [o_i]_{R_{Ind}}$$

Informally, $Y$ is the union of all those elementary sets that are contained by $Y$. The upper approximation of $Y$ based on $R_{Ind}$ is defined as:

$$\bar{Y} = \bigcup_{[o_i]_{R_{Ind}} \cap Y \neq \emptyset} [o_i]_{R_{Ind}}$$

In other words, $\bar{Y}$ is the union of those elementary sets that have a non-empty intersection with $Y$.

<table>
<thead>
<tr>
<th>Medical</th>
<th>Headache</th>
<th>Temp</th>
<th>Flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>yes</td>
<td>normal</td>
<td>no</td>
</tr>
<tr>
<td>$e_2$</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>$e_3$</td>
<td>yes</td>
<td>very_high</td>
<td>yes</td>
</tr>
<tr>
<td>$e_4$</td>
<td>no</td>
<td>normal</td>
<td>no</td>
</tr>
<tr>
<td>$e_5$</td>
<td>no</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>$e_6$</td>
<td>no</td>
<td>very_high</td>
<td>yes</td>
</tr>
<tr>
<td>$e_7$</td>
<td>no</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>$e_8$</td>
<td>no</td>
<td>very_high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 4.2: Reduced Information Table

4.1.3 Approximation Space
Example 4.1.2 Let the concept $\textit{Sick} = \{e_2, e_3, e_6, e_7, e_8\}$ denote sick people in the information table in Example 4.1.1. The lower and upper approximations based on $\mathcal{R}_{\text{Headache}}$ are as follows:

$$
\textit{sick} = \emptyset \\
\overline{\textit{sick}} = \{e_1, e_2, e_3\} \cup \{e_4, e_5, e_6, e_7, e_8\} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}
$$

The lower and upper approximations based on $\mathcal{R}_{\text{Temp}}$ are as follows:

$$
\textit{sick} = \{e_3, e_6, e_8\} \\
\overline{\textit{sick}} = \{e_2, e_5, e_7\} \cup \{e_3, e_6, e_8\} = \{e_2, e_3, e_5, e_6, e_7, e_8\}
$$

The lower and upper approximations based on $\mathcal{R}_{\text{Headache,Temp}}$ are as follows:

$$
\textit{sick} = \{e_2\} \cup \{e_3\} = \{e_2, e_3\} \\
\overline{\textit{sick}} = \{e_2\} \cup \{e_3\} \cup \{e_5, e_7\} \cup \{e_6, e_8\} = \{e_2, e_3, e_5, e_6, e_7, e_8\}
$$

The lower approximation of a concept consists of all objects which surely belong to the concept whereas the upper approximation of the concept consists of all objects which possibly belong to the concept in question. The set $\overline{Y} - Y$ is called the boundary region of the concept and consists of all objects which cannot be classified with certainty to the concept or its complement employing available knowledge. If an element is in $\overline{Y} - Y$, we cannot be certain if it is in $Y$. We call $\overline{Y} - Y$ the doubtful region.

In order to measure the degree of certainty in determining whether or not elements in $\mathcal{O}$ are members of $Y$, the notion of a discriminant index of $Y$ has been introduced. It is defined as follows:

$$
\alpha_{\mathcal{R}_{\text{Ind}}}(Y) = 1 - |\overline{Y} - Y|/|\mathcal{O}|
$$

If $\alpha_{\mathcal{R}_{\text{Ind}}}(Y) = 1$, which means $\overline{Y} = Y$, then the concept $Y$ is precise. If $\alpha_{\mathcal{R}_{\text{Ind}}}(Y) = 0$, which means $\overline{Y} = \mathcal{O}$ and $Y = \emptyset$, then the concept $Y$ is completely uncertain.
Continuing with Example 4.1.2, we have

\[
\begin{align*}
\alpha_{R_{\text{Headache}}}(\text{Sick}) &= 1 - 8/8 = 0 \\
\alpha_{R_{\text{Temp}}}(\text{Sick}) &= 1 - 3/8 = 0.625 \\
\alpha_{R_{\text{Headache},\text{Temp}}}(\text{Sick}) &= 1 - 3/8 = 0.625
\end{align*}
\]

4.2 Inductive Learning

Inductive Learning, a research area in Artificial Intelligence, is used to model the knowledge of human experts by using a carefully chosen sample of expert decisions and inferring decision rules automatically, independent of the subject of interest. Rough Set based Inductive Learning uses Rough Set theory to compute general decision rules [49, 41]. It can be used to determine the relationship between the set of attributes and the concept.

Inductive Learning (learning from examples) is the oldest and best understood problem in artificial intelligence [49]. Many existing expert systems were built by manually encoding the knowledge of human experts [18, 34, 35, 46]. Encoding processes as such can be very time consuming as they require close collaboration between computer professionals and experts of the subject domain. To design expert systems in this way is rather inefficient particularly because the same tedious task has to be performed for each specialized application.

4.2.1 Inductive Learning Algorithm

Based on the notions of approximation classification of a set presented above, we now introduce the inductive learning algorithm proposed by Ziarko and Wong [48].

Given an information system \( I = (\mathcal{O}, \mathcal{C} \cup \mathcal{D}, \mathcal{V}, f) \) and a subset of objects \( Y \subseteq \mathcal{O} \) representing a single concept, we can compute the discriminant index \( \alpha_B(Y) \) for any subset of student attributes \( B \subseteq \mathcal{C} \). Using the discriminant indices as attribute selection criteria, we can choose an appropriate subset of attributes in \( \mathcal{C} \) to construct the decision rules for this particular class \( Y \). The same procedure can be applied
repeatedly to derive decision rules for every concept involved in the knowledge representation system $I$.

**INPUT:** Information system $I = (\mathcal{O}, \mathcal{C} \cup \mathcal{D}, \mathcal{N}, f)$,

**PROCEDURE:**

**Step 0.** LET $j = 1$;

**Step 1.** LET $\mathcal{O}' = \mathcal{O}$, $\mathcal{C}' = \mathcal{C}$, $B = \emptyset$, and $Y = Y_j$;

**Step 2.** COMPUTE the set of discriminant indices

$$\{\alpha_{B'}(Y) \mid B' = B \cup \{c\}, \forall c \in \mathcal{C}'\}$$

in $I = (\mathcal{O}, \mathcal{C} \cup \mathcal{D}, \mathcal{N}, f)$;

SELECT the set of attributes $B' = B \cup \{c\}$ with the highest value $\alpha_{B'}(Y)$

LET $B = B'$;

**Step 3.** IF ($Y = \emptyset$) GOTO Step 4.

4) IDENTIFY those equivalence classes: $\{X_1, ..., X_r\}$ contained by $Y$;

5) OUTPUT deterministic decision rules:

$$\{\text{Des}(X_k) \Rightarrow \text{Des}(Y) \mid k = 1, 2, ..., r\};$$

**Step 4.** LET $\mathcal{O}' = \mathcal{O}' - ((\mathcal{O}' - \overline{Y}) \cup \overline{Y})$, $Y = Y - Y$

IF ($\mathcal{O}' \neq \emptyset$) GOTO Step 6.

LET $\mathcal{C}' = \mathcal{C}' - B$

IF $\mathcal{C}' \neq \emptyset$ GOTO Step 3

**Step 5.** OUTPUT deterministic or non-deterministic decision rules:

$$\{\text{Des}(X_k) \Rightarrow \text{Des}(Y) \mid k = 1, 2, ..., r\};$$

**Step 6.** Let $j = j + 1$; If ($j \leq n$) GOTO step 1.

**END PROCEDURE.**

**OUTPUT:** Decision rules for EVERY CLASS $Y_j \subseteq T$ of the expert classification.

### 4.2.2 Decision Tree

Decision Tree is a well-known method that is used to find decision rules. Quinlan [49] suggested that classification (decision) rules in the form of a decision tree can be constructed for a collection of objects characterized by attributes and attribute values. The attribute with the highest discriminant index value should be chosen to be the root of the tree. Each leaf node of a tree corresponds to an equivalence class
of the root attribute. Those leaf nodes containing objects of different expert classes require further classification. In such cases, the same procedure can be performed for each of these leaves by choosing the best attribute among the available set of attributes, except that attributes already used to reach the current leaf node should not be included.

To illustrate Quinlan's method, consider again the collection of objects given in Table 4.2. As shown above, Temp has a higher alpha value than Headache. According to Quinlan's mechanism, we then partition the objects into a number of disjoint blocks (equivalence classes) based on the values of the attribute Temp as shown in Figure 4.1.

The leaf node high in Figure 4.1 contains objects of two expert classes based on the decision attribute Flu: yes and no. This means that further classification is necessary for this subset of objects. We can compute the reduction of the entropy value for each of the remaining attributes. Since Headache is the only attribute left, no further classification can be evaluated; therefore, we obtain the rules that are:

- If \{Temp = normal\} \implies Flu = No
- If \{Temp = high\} \implies Flu = No
- If \{Temp = high\} \implies Flu = yes
- If \{Temp = very\_high\} \implies Flu = Yes

Thus, according to Quinlan, this is an non-deterministic decision tree because the tree is terminated with some leaf nodes still containing objects of different expert classes.
Decision tree has two roles in this research. One is to allow us to visualize the decision rule discovery process. The other is to simplify the implementation of the Rough Set based Distance Learning Algorithm. See Chapter 6 for details.

In the next chapter, we apply Rough Set Inductive Learning to the WebCT context.
Chapter 5

Rough Set based Distance Learning Algorithm

We have discussed Rough Set theory in Chapter 4 and we know that Rough Sets are used as a mathematical tool for approximate modeling of classification problems.

Our concern with the WebCT system and distance delivery provision is that there is no way to accurately measure the achievement of students. Such measurement is essential to any education system [9, 10]. Therefore, we extend this idea and propose a model to use Rough Sets in distance education. We analyze student grade information and form an information table to find the rules behind the information table; that is, by analyzing the student grade history, we determine the main material, which students failed to understand as shown by their results in assignments and quizzes, that results in their final failure. The assumption is that the lack of understanding of key areas of the course material results in failure on the final exam. Thus, we use a table of results from course work to determine the rules associated with failure on the final exam. Then we can inform students in subsequent courses of the core sections of the course and provide guidance for online students.

The Inductive Learning Algorithm proposed by Wong and Ziarko [48] was not designed for a distance learning situation for the following reasons. First, it only outputs deterministic rules at the intermediate level. For distance education, nondeterministic rules at the intermediate step can inform online students about important information useful in guiding their distance learning. Second, for distance education, we are
primarily interested in one concept $Y$, such that $\text{Des}(Y) = \{\text{Fail}\}$, i.e., the failure concept, because we want to find out what causes online students to fail. But the Inductive Learning Algorithm covers multiple concepts. Thus, we have adapted the Inductive Learning Algorithm to a distance education environment and the result is the Rough Set Based Distance Learning Algorithm (RSDLA) [26]. Unlike the Inductive Learning Algorithm, RSDLA calculates the knowledge reduction as introduced in section 4.1.2.

The following is the RSDLA algorithm.

INPUT: Failure Concept $Y$, Domain $O$, Set of Condition Attributes $C$, Decision Attribute $D$

ALGORITHM:
Step 1. COMPUTE the knowledge reduction with respect to $D$
Step 2. LET $O' = O$, $C' = C$, $Q = \emptyset$;
Step 3. WHILE ($C' \neq \emptyset$) {
    1) COMPUTE the set of discriminant indices
       $\{\alpha_{Q'}(Y) \mid Q' = Q \cup \{c\}, \forall c \in C'\}$;
    2) SELECT the set of attributes $Q' = Q \cup \{c\}$ with the highest
       values $\alpha_{Q'}(Y)$
    3) LET $Q = Q'$;
    4) IDENTIFY those equivalence classes: $\{X_1, ..., X_r\}$ contained by $Y$;
    5) OUTPUT deterministic or non-deterministic decision rules:
       $\{\text{Des}(X_k) \Rightarrow \text{Des}(Y) \mid k = 1, 2, ..., r\}$;
    6) LET $O' = O' - ((O' - Y) \cup Y)$
    7) IF ($O' \neq \emptyset$)
       LET $C' = C' - Q$;
    ELSE
       STOP;
}
Step 4. Output deterministic or non-deterministic decision rules:
$\{\text{Des}(X'_i) \Rightarrow \text{Des}(Y) \mid \forall$ equivalence class $X'_i$ of the relation $C'$ on $O'\}$
Table 5.1: Student Information Database

<table>
<thead>
<tr>
<th>Quiz 1</th>
<th>Quiz 2</th>
<th>Quiz 3</th>
<th>Quiz 4</th>
<th>Quiz 5</th>
<th>FINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>96</td>
<td>91</td>
<td>81</td>
<td>100</td>
<td>93</td>
</tr>
<tr>
<td>s_2</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_3</td>
<td>92</td>
<td>82</td>
<td>74</td>
<td>100</td>
<td>86</td>
</tr>
<tr>
<td>s_4</td>
<td>86</td>
<td>87</td>
<td>57</td>
<td>90</td>
<td>92</td>
</tr>
<tr>
<td>s_5</td>
<td>96</td>
<td>93</td>
<td>73</td>
<td>100</td>
<td>92</td>
</tr>
<tr>
<td>s_6</td>
<td>47</td>
<td>28</td>
<td>46</td>
<td>80</td>
<td>65</td>
</tr>
<tr>
<td>s_7</td>
<td>58</td>
<td>35</td>
<td>71</td>
<td>100</td>
<td>79</td>
</tr>
<tr>
<td>s_8</td>
<td>89</td>
<td>69</td>
<td>54</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>s_9</td>
<td>90</td>
<td>72</td>
<td>58</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>s_{10}</td>
<td>76</td>
<td>76</td>
<td>54</td>
<td>30</td>
<td>68</td>
</tr>
<tr>
<td>s_{11}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_{12}</td>
<td>40</td>
<td>81</td>
<td>67</td>
<td>30</td>
<td>81</td>
</tr>
<tr>
<td>s_{13}</td>
<td>52</td>
<td>59</td>
<td>56</td>
<td>100</td>
<td>81</td>
</tr>
<tr>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>s_{329}</td>
<td>100</td>
<td>76</td>
<td>64</td>
<td>80</td>
<td>86</td>
</tr>
</tbody>
</table>

END ALGORITHM.

OUTPUT: Decision rules for the concept.

5.1 Using RSDLAD

To illustrate this idea, we use an example here. Table 5.1 is an information table from an introductory computer science class (CS 100) at the University of Regina in which students use the Web for their course materials. In this table, there are 329 students, 5 quizzes, and one final examination.

In order to determine the core material or rules behind this table, we convert the percentage marks into pass or fail marks (marks greater than or equal to 50 is p and less than 50 is f), combine entries with the same values, and generate a small table. Table 5.2 shows the collapsed student information table that contains 14 sample classes with a total of 83 students. We omit some extreme cases such as
passing (204 students) and failure (26 students) in all areas and several specific cases with only one student are not represented. The first class has a total number of 6 students who passed Quiz 1 and fail Quizzes 2, 3, 4, 5, and the final. The second class has 3 students who failed Quizzes 1, 2, and 3 and passed Quizzes 4, 5, and the final. The rest read the same way.

In order to use the Rough Set based Distance Learning algorithm [26], we first obtain the domain $\mathcal{O}$ and two concepts $Y_{\text{pass}}$ and $Y_{\text{fail}}$ from the decision attribute (FINAL):

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\}$$

$$Y_{\text{pass}} = \{e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_{12}, e_{13}, e_{14}\}$$

$$Y_{\text{fail}} = \{e_1, e_9, e_{10}, e_{11}, e_{14}\}$$

The equivalence classes for conditional attributes and decision attributes as well
as the positive region of \textit{Final} are as follows:

\[
\mathcal{R}_{\text{final}} = \{\{e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_{12}, e_{13}\}, \{e_1, e_9, e_{10}, e_{11}, e_{14}\}\}
\]

\[
\mathcal{R}_{Q_1,Q_2,Q_3,Q_4,Q_5} = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_7\},
\{e_8\}, \{e_9\}, \{e_{10}\}, \{e_{11}\}, \{e_{12}\}, \{e_{13}\}, \{e_{14}\}\}
\]

\[
\text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5}(\text{Final}) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\}
\]

In order to compute the reduct with respect to \textit{Final}, we need to find dispensable attributes: that is, attributes whose removal does not change the positive region of \textit{Final}.

\[
\mathcal{R}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_1} = \{\{e_1\}, \{e_2, e_8\}, \{e_3, e_{13}\}, \{e_4, e_7, e_8\}, \{e_5\}, \{e_9\},
\{e_{10}\}, \{e_{11}\}, \{e_{12}\}, \{e_{14}\}\}
\]

\[
\text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_1}(\text{Final}) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\}
\]

\[
\neq \text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5}(\text{Final})
\]

\[
\mathcal{R}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_2} = \{\{e_1\}, \{e_2, e_8\}, \{e_3, e_{11}\}, \{e_4, e_6, e_7\}, \{e_5, e_{13}\}, \{e_9\},
\{e_{10}\}, \{e_{12}\}, \{e_{14}\}\}
\]

\[
\text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_2}(\text{Final}) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{12}, e_{13}, e_{14}\}
\]

\[
\neq \text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5}(\text{Final})
\]

\[
\mathcal{R}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_3} = \{\{e_1\}, \{e_2, e_{13}\}, \{e_3, e_6\}, \{e_4, e_7, e_{11}\}, \{e_5, e_8\}, \{e_9\},
\{e_{10}\}, \{e_{12}\}, \{e_{14}\}\}
\]

\[
\text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_3}(\text{Final}) = \{e_1, e_2, e_3, e_5, e_6, e_8, e_9, e_{10}, e_{12}, e_{13}, e_{14}\}
\]

\[
\neq \text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5}(\text{Final})
\]

\[
\mathcal{R}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_4} = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4, e_7\}, \{e_5, e_{14}\}, \{e_6\}, \{e_7\}, \{e_8\}, \{e_9\},
\{e_{10}\}, \{e_{12}\}, \{e_{13}\}\}
\]

\[
\text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_4}(\text{Final}) = \{e_1, e_2, e_3, e_4, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}\}
\]

\[
\neq \text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5}(\text{Final})
\]

\[
\mathcal{R}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_5} = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4, e_7\}, \{e_5\}, \{e_6\}, \{e_8\}, \{e_9\},
\{e_{10}, e_{11}\}, \{e_{12}\}, \{e_{13}\}, \{e_{14}\}\}
\]

\[
\text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5-Q_5}(\text{Final}) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\}
\]

\[
= \text{POS}_{Q_1,Q_2,Q_3,Q_4,Q_5}(\text{Final})
\]
Table 5.3: Horizontally Collapsed Student Information Table

<table>
<thead>
<tr>
<th>Quiz 2</th>
<th>Quiz 3</th>
<th>Quiz 4</th>
<th>FINAL</th>
<th>NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>e₁</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>e₂</td>
<td>f</td>
<td>f</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₃</td>
<td>f</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₄</td>
<td>p</td>
<td>f</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₅</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₆</td>
<td>f</td>
<td>f</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₇</td>
<td>p</td>
<td>f</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₈</td>
<td>p</td>
<td>f</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₉</td>
<td>f</td>
<td>p</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>e₁₀</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>f</td>
</tr>
<tr>
<td>e₁₁</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>f</td>
</tr>
<tr>
<td>e₁₂</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>p</td>
</tr>
<tr>
<td>e₁₃</td>
<td>f</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>e₁₄</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>f</td>
</tr>
</tbody>
</table>

Therefore, attributes Quiz1 and Quiz5 are dispensable. The reduct is shown in Table 5.3.

As we are only interested in what sections cause students to fail the course, we use the fail concept in our discussion; that is, $Y = Y_{\text{fail}}$.

To find the rules, we first find the indiscernibility classes based on Quiz 2, which are $\{e₁, e₂, e₃, e₅, e₉, e₁₃\}$ and $\{e₄, e₅, e₈, e₉, e₁₃, e₁₄\}$.

The lower approximation is $Y = \bigcup_{Ω_i ⊆ Y} Ω_i = \emptyset$. The upper approximation is $\overline{Y} = \bigcup_{Ω_i ∩ Y ≠ \emptyset} Ω_i = \{e₁, e₂, e₃, e₄, e₅, e₆, e₇, e₈, e₉, e₁₀, e₁₁, e₁₂, e₁₃, e₁₄\} = E$

The discriminant index of a concept $Y$ is defined using the following formula:

$$\alpha_{Q₁}(Y) = 1 - \frac{|\overline{Y} - Y|}{|Ω|}$$

Therefore, the discriminant index of Quiz 2 is $\alpha_{Q₂}(Y) = 1 - (83 - 0)/83 = 0$ that determines how well the singleton set of attributes consisting of Quiz 2 specifies the membership in $Y$ (the fail concept).
We continue to find indiscernibility classes based on Quiz 3 that are \{e_1, e_2, e_4, e_6, e_7, e_8, e_{12}\} and \{e_3, e_5, e_9, e_{10}, e_{11}, e_{14}\}.

The lower approximation is \( \mathcal{Y} = \bigcup_{i \in \mathcal{Y}} \Omega_i = \emptyset \). The upper approximation is \( \overline{\mathcal{Y}} = \bigcup_{i \in \mathcal{Y} \neq \emptyset} \Omega_i = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\} \).

The discriminant index of Quiz 3 is \( \alpha_{Q3}(\mathcal{Y}) = 1 - |\overline{\mathcal{Y}} - \mathcal{Y}|/|\mathcal{O}| = 0 \).

The indiscernibility classes based on Quiz 4 are \{e_1, e_4, e_9, e_{12}, e_{14}\} and \{e_2, e_3, e_5, e_6, e_7, e_8, e_{10}, e_{11}, e_{13}\}.

The lower approximation is \( \mathcal{Y} = \bigcup_{i \in \mathcal{Y}} \Omega_i = \emptyset \). The upper approximation is \( \overline{\mathcal{Y}} = \bigcup_{i \in \mathcal{Y} \neq \emptyset} \Omega_i = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\} \).

The discriminant index of Quiz 4 is \( \alpha_{Q4}(\mathcal{Y}) = 1 - |\overline{\mathcal{Y}} - \mathcal{Y}|/|\mathcal{O}| = 0 \).

Because all of the discriminant indices are 0, we cannot determine the highest value; therefore, we do not have any rules generated here. We merge one conditional attribute with each other attribute and see the result. The merged conditional attributes are Quizzes 2 and 3, Quizzes 3 and 4, and Quizzes 2 and 4.

For Quizzes 2 and 3, we find the following indiscernibility classes:

\[
\begin{align*}
    ff & : \{e_1, e_2, e_6\}, \\
    fp & : \{e_3, e_9, e_{13}\}, \\
    pp & : \{e_5, e_{10}, e_{11}, e_{14}\}, \\
    pf & : \{e_4, e_7, e_8, e_{12}\}.
\end{align*}
\]

where \(xy\) means Quiz 2 has value \(x\) and Quiz 3 has value \(y\). The lower approximation is \( \mathcal{Y} = \bigcup_{i \in \mathcal{Y}} \Omega_i = \emptyset \). The upper approximation is \( \overline{\mathcal{Y}} = \bigcup_{i \in \mathcal{Y} \neq \emptyset} \Omega_i = \{e_1, e_2, e_3, e_5, e_6, e_9, e_{10}, e_{11}, e_{13}, e_{14}\} \). The discriminant index of Quizzes 2 and 3 is \( \alpha_{Q2,Q3}(\mathcal{Y}) = 1 - |\overline{\mathcal{Y}} - \mathcal{Y}|/|\mathcal{O}| = 1 - (55 - 0)/83 = 0.33 \).

We find the indiscernibility classes for Quizzes 3 and 4 that are:

\[
\begin{align*}
    ff & : \{e_1, e_{12}\}, \\
    fp & : \{e_2, e_4, e_6, e_7, e_8\}, \\
    pp & : \{e_3, e_5, e_{10}, e_{11}, e_{13}\}, \\
    pf & : \{e_9, e_{14}\}.
\end{align*}
\]

The lower approximation is \( \mathcal{Y} = \bigcup_{i \in \mathcal{Y}} \Omega_i = \{e_9, e_{14}\} \). The upper approximation
is \( \bar{Y} = \bigcup_{\Omega_i \cap Y \neq \emptyset} \Omega_i = \{e_1, e_3, e_5, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\} \). The discriminant index of Quizzes 3 and 4 is \( \alpha_{Q2,Q3}(Y) = 1 - |\bar{Y} - Y|/|\mathcal{O}| = 1 - (53 - 5)/83 = 0.42 \).

We find the indiscernibility classes for Quizzes 2 and 4 that are:

- \(ff\): \{e_1, e_9\},
- \(fp\): \{e_2, e_3, e_6, e_{13}\},
- \(pp\): \{e_4, e_5, e_7, e_8, e_{10}, e_{11}\},
- \(pf\): \{e_{12}, e_{14}\}.

The lower approximation is \( Y = \bigcup_{\Omega_i \subseteq Y} \Omega_i = \{e_1, e_9\} \). The upper approximation is \( \bar{Y} = \bigcup_{\Omega_i \cap Y \neq \emptyset} \Omega_i = \{e_1, e_4, e_5, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{14}\} \). The discriminant index of Quizzes 2 and 4 is \( \alpha_{Q2,Q4}(Y) = 1 - |\bar{Y} - Y|/|\mathcal{O}| = 1 - (56 - 9)/83 = 0.43 \).

Because \( \alpha_{Q2,Q4}(Y) \) has the highest discriminant index, it best determines the membership in \( Y \), which is Quiz 2 fail and Quiz 4 fail. Thus, we obtain the first rule:

\[ R_1 : \{\text{Quiz2} = f, \text{Quiz4} = f\} \Rightarrow \{\text{Final} = f\} \]

Let us see if we can find a new domain and compute new rules.

The elements that are not needed are:

\[(E - \bar{Y}) \cup (Y) = \{e_2, e_3, e_6, e_{13}\} \cup \{e_1, e_9\}\]

By checking against Table 5.3, we can see that the first set is not in the fail concept and the second set has been handled by rule 1. Therefore, the new set of elements is:

\[E - ((E - \bar{Y}) \cup (Y)) = \{e_4, e_5, e_7, e_8, e_{10}, e_{11}, e_{12}, e_{14}\}\]

Based on this, we obtain the further collapsed information table Table 5.4.

The domain \( \mathcal{O} \) and fail concept for this selected information table are \( \mathcal{O} = \{e_4, e_5, e_7, e_8, e_{10}, e_{11}, e_{12}, e_{14}\} \) and \( Y_{fail} = \{e_{10}, e_{11}, e_{14}\} \).

Now, we merge Quizzes 2, 3, and 4 that is the only combination. The indiscernibility classes based on Quizzes 2, 3, and 4 are

- \(pfp\): \{e_4, e_7, e_8\},
- \(ppp\): \{e_5, e_{10}, e_{11}\}, and
The lower approximation is $\bar{Y} = \bigcup_{i : \Omega_i \neq \emptyset} \Omega_i = \{e_{14}\}$. The upper approximation is $\overline{\bar{Y}} = \bigcup_{i : \Omega_i \neq \emptyset} \Omega_i = \{e_5, e_{10}, e_{11}, e_{14}\}$. The discriminant index of Quizzes 2, 3, and 4 is

$$\alpha_{Q2,Q3,Q4}(\bar{Y}) = 1 - \frac{|\overline{\bar{Y}} - \bar{Y}|}{|\mathcal{O}|} = 1 - \frac{(19 - 2) / 47}{0.65} = 0.65$$

where Quiz 2 is pass, Quiz 3 is pass, and Quiz 4 is fail.

Since there are no more condition attributes, we stop here. Thus, we obtain the second rule as follows:

$$R_2 : \{\text{Quiz} 4 = f\} \Rightarrow \{\text{Final} = f\}$$

A question that arises is how much we can believe in the two rules. Since the rules themselves do not tell us this, we need to evaluate the strength of the four rules.

The strength of a rule can be measured as follows:

$$\text{# of positive students covered by the rule}$$
$$\text{# of students covered by the rule (including both positive and negative)}$$

By this definition, the first rule above has a strength of $\frac{2}{3}$, that is, 100%. Class $e_1$ and $e_3$ from Table 5.3 are two positive examples covered by the rule. The second
rule has a strength of $\frac{2}{7}$, that is, 29%. Class $e_{14}$ from Table 5.3 is a positive example and class $e_{12}$ is a negative example covered by this rule.

Suppose that we do not have information on Quiz 2 for a student who failed the final exam because the student skipped this module but we do have information that he/she failed Quiz 4. In applying the first rule to this student, there is a 27% chance that the reason for failure was solely the failure of Quiz 4. However, there is a higher probability that the reason for failure was due to additionally failing Quiz 2 because the strength of first rule is 100%. It is a certain rule. Hence, it would be applied in favor of the one of 27% strength.

The instructor also gains information from the rules that would normally be obtained from students in a conventional lecture setting. There is assumed to be an ordering on the modules tested by the quizzes. For example, Quiz 4 requires knowledge of modules tested by at least some of the previous quizzes. Rules induced from the information table might indicate that, in fact, the previous modules are not valuable for preparing the student for a given module.
Chapter 6

Implementation of RSDL

Before discussing the implementation, we first introduce the problem again. There is a text file that contains student records including assignments, quizzes, and one final examination. Failure in the final examination means failure in the course. Therefore, the main issue here is that we want to determine the material students failed to understand as shown by their results in assignments and quizzes. The assumption is that the lack of understanding of key areas of the course material results in failure on the final exam. Thus, we use a table of results from course work to determine the rules associated with failure of the final exam. Then we can inform students in subsequent courses of the core sections of the course and provide guidance for online students. The following example, Table 6.1, is from a Java class at the University of Regina in which students use the Web for their course materials. In this table, there are 115 students, 6 quizzes, and one final examination.

We chose the Java programming language to do the implementation. The reasons for choosing this programming language included the following factors. Firstly, Java is becoming more and more popular. Secondly, it has various utilities that can simplify the programming task. Thirdly, Java aims to become a universal programming language running on a wide variety of machines. Finally, it can easily be used with web applications. To implement this work, we divide the task into five classes. The five classes are: RoughSet class, ReadData class, Reduct class, Inductive class, and Strength class.
Table 6.1: Student Information Database

<table>
<thead>
<tr>
<th></th>
<th>Quiz 1</th>
<th>Quiz 2</th>
<th>Quiz 3</th>
<th>Quiz 4</th>
<th>Quiz 5</th>
<th>Quiz 6</th>
<th>FINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>98</td>
<td>100</td>
<td>90</td>
<td>89</td>
<td>91</td>
<td>85</td>
<td>90</td>
</tr>
<tr>
<td>$s_2$</td>
<td>100</td>
<td>90</td>
<td>30</td>
<td>45</td>
<td>55</td>
<td>32</td>
<td>40</td>
</tr>
<tr>
<td>$s_3$</td>
<td>68</td>
<td>70</td>
<td>80</td>
<td>89</td>
<td>91</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>$s_4$</td>
<td>88</td>
<td>100</td>
<td>80</td>
<td>69</td>
<td>75</td>
<td>85</td>
<td>81</td>
</tr>
<tr>
<td>$s_5$</td>
<td>76</td>
<td>65</td>
<td>50</td>
<td>70</td>
<td>46</td>
<td>49</td>
<td>46</td>
</tr>
<tr>
<td>$s_6$</td>
<td>88</td>
<td>70</td>
<td>90</td>
<td>79</td>
<td>87</td>
<td>86</td>
<td>80</td>
</tr>
<tr>
<td>$s_7$</td>
<td>98</td>
<td>78</td>
<td>47</td>
<td>69</td>
<td>91</td>
<td>85</td>
<td>60</td>
</tr>
<tr>
<td>$s_8$</td>
<td>88</td>
<td>90</td>
<td>80</td>
<td>69</td>
<td>77</td>
<td>85</td>
<td>80</td>
</tr>
<tr>
<td>$s_9$</td>
<td>80</td>
<td>70</td>
<td>80</td>
<td>75</td>
<td>80</td>
<td>68</td>
<td>78</td>
</tr>
<tr>
<td>$s_{10}$</td>
<td>80</td>
<td>90</td>
<td>30</td>
<td>45</td>
<td>55</td>
<td>43</td>
<td>45</td>
</tr>
<tr>
<td>$s_{11}$</td>
<td>98</td>
<td>78</td>
<td>47</td>
<td>70</td>
<td>91</td>
<td>85</td>
<td>76</td>
</tr>
<tr>
<td>$s_{12}$</td>
<td>80</td>
<td>70</td>
<td>82</td>
<td>75</td>
<td>80</td>
<td>87</td>
<td>89</td>
</tr>
<tr>
<td>$s_{13}$</td>
<td>49</td>
<td>36</td>
<td>30</td>
<td>45</td>
<td>41</td>
<td>43</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$s_{115}$</td>
<td>58</td>
<td>70</td>
<td>90</td>
<td>60</td>
<td>78</td>
<td>97</td>
<td>87</td>
</tr>
</tbody>
</table>

RoughSet Class: This class imports the ReadData class, Reduct class, Inductive class, and Strength class. It contains the main method that makes this class the executable class. It defines several two dimensional arrays to hold the converted information table and invokes the methods that are defined in the imported classes.

ReadData Class: This class reads the student information needed from any text file, converts percentage marks into pass or fail marks, combines entries with the same values, and generates a small table. The StringTokenizer class predefined in the Java language is used to process the text file. Using Table 6.1 as our example, the ReadData class reads Table 6.1, combines the student records and derives the collapsed student information table that contains the eight sample classes shown in the table Table 6.2 below. The total number of students in Table 6.2 is 104. The other 11 students received failure scores in all areas and do not need to be considered here.
Table 6.2: Collapsed Student Information Table

<table>
<thead>
<tr>
<th></th>
<th>Quiz 1</th>
<th>Quiz 2</th>
<th>Quiz 3</th>
<th>Quiz 4</th>
<th>Quiz 5</th>
<th>Quiz 6</th>
<th>FINAL</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>74</td>
</tr>
<tr>
<td>e2</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>4</td>
</tr>
<tr>
<td>e3</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>3</td>
</tr>
<tr>
<td>e4</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>6</td>
</tr>
<tr>
<td>e5</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>1</td>
</tr>
<tr>
<td>e6</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>5</td>
</tr>
<tr>
<td>e7</td>
<td>f</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>7</td>
</tr>
<tr>
<td>e8</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>f</td>
<td>p</td>
<td>4</td>
</tr>
</tbody>
</table>

**Reduct Class:** The ReadData class has successfully derived the collapsed table, but it may contain redundant condition attributes, which means that some condition attributes may not determine the decision at all. Therefore, we need to find the reduct and remove the redundant attributes.

To implement this step, linear searching and sorting algorithms are used to accomplish this task and a three dimensional array is used to hold the collection of the equivalence class families:

\[
R_{Q1} = \{\{e1, e2, e3, e4, e5, e6, e8\}, \{e7\}\}
\]

\[
R_{Q2} = \{\{e1, e2, e4, e5, e6, e7, e8\}, \{e3\}\}
\]

\[
R_{Q3} = \{\{e1, e6, e7, e8\}, \{e2, e3, e4, e5\}\}
\]

\[
R_{Q4} = \{\{e1, e3, e4, e5, e7, e8\}, \{e2, e6\}\}
\]

\[
R_{Q5} = \{\{e1, e2, e3, e4, e5, e7, e8\}, \{e6\}\}
\]

\[
R_{Q6} = \{\{e1, e4, e7\}, \{e2, e3, e5, e6, e8\}\}
\]

\[
R_{Final} = \{\{e1, e4, e7, e8\}, \{e2, e3, e5, e6\}\}
\]

A two dimensional array is used to hold the intersect of the family \(R = \{Q_1, ..., Q_6\}\)
induced classification:

\[ \mathcal{R}_{Q_1,\ldots,Q_6} = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_7\}, \{e_8\}\} \]

By unioning the equivalence classes of \(\mathcal{R}_{Q_1,\ldots,Q_6}\) which are included in some equivalence classes of \(\mathcal{R}_{\text{Final}}\), the positive region of \(\text{Final}\) is obtained and stored in an one-dimensional array.

The positive region of \(R\) is held in an one dimensional array:

\[ \text{POS}_{Q_1,\ldots,Q_6}(\text{Final}) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} \]

In order to compute the reduct of \(R\) with respect to \(\text{Final}\), we then have to find out whether the family \(R\) is \(\text{Final}\)-dependent or not. We use a while loop to remove \(Q_1, Q_2, Q_3, Q_4, Q_5,\) and \(Q_6\) respectively and check the positive region in the mean time. If the positive region of each quiz is different from the positive region of \(\text{Final}\), we keep this condition attribute in order to find the reduct. The following is the result:

\[
\begin{align*}
\mathcal{R}_{Q_1,\ldots,Q_6-Q_1} &= \{\{e_1, e_7\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \{e_6\}, \{e_8\}\} \\
\text{POS}_{Q_1,\ldots,Q_6-Q_1}(\text{Final}) &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} = \text{POS}_{Q_1,\ldots,Q_6}(\text{Final}) \\
\mathcal{R}_{Q_1,\ldots,Q_6-Q_2} &= \{\{e_1\}, \{e_2\}, \{e_3, e_5\}, \{e_4\}, \{e_6\}, \{e_7\}, \{e_8\}\} \\
\text{POS}_{Q_1,\ldots,Q_6-Q_2}(\text{Final}) &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} = \text{POS}_{Q_1,\ldots,Q_6}(\text{Final}) \\
\mathcal{R}_{Q_1,\ldots,Q_6-Q_3} &= \{\{e_1, e_4\}, \{e_2\}, \{e_3\}, \{e_5, e_8\}, \{e_6\}, \{e_7\}\} \\
\text{POS}_{Q_1,\ldots,Q_6-Q_3}(\text{Final}) &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\} \neq \text{POS}_{Q_1,\ldots,Q_6}(\text{Final}) \\
\mathcal{R}_{Q_1,\ldots,Q_6-Q_4} &= \{\{e_1\}, \{e_2, e_5\}, \{e_3\}, \{e_4\}, \{e_6\}, \{e_7\}, \{e_8\}\} \\
\text{POS}_{Q_1,\ldots,Q_6-Q_4}(\text{Final}) &= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} = \text{POS}_{Q_1,\ldots,Q_6}(\text{Final}) \\
\mathcal{R}_{Q_1,\ldots,Q_6-Q_5} &= \{\{e_1, e_7\}, \{e_2, e_3, e_5\}, \{e_4\}, \{e_6\}, \{e_8\}\} \\
\text{POS}_{Q_1,\ldots,Q_6-Q_5}(\text{Final}) &= \{e_1, e_2, e_3, e_4, e_5, e_7\} \neq \text{POS}_{Q_1,\ldots,Q_6}(\text{Final}) \\
\mathcal{R}_{Q_1,\ldots,Q_6-Q_6}(\text{Final}) &= \{\{e_1, e_7, e_8\}, \{e_2, e_3, e_4, e_5\}, \{e_6\}\} \\
\text{POS}_{Q_1,\ldots,Q_6-Q_6}(\text{Final}) &= \{e_1, e_6, e_7, e_8\} \neq \text{POS}_{Q_1,\ldots,Q_6}(\text{Final})
\end{align*}
\]

Because each positive region of Quiz 3, Quiz 5, and Quiz 6 does not match the
positive region of Final, Quiz3, Quiz5 and Quiz6 are indispensable. Therefore, we obtain the reduct and put it into a two-dimensional array as shown in Table 6.3.

**Inductive Class:** The remaining work is to find the rules. This class uses several arrays repeatedly to hold the domain, concept, indiscernibility classes, lower and upper approximations, and discriminant indices in a while loop. It then shows the alpha values, the discriminant indices. The following is the outcome:

**Domain:** \( O = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} \)

**Failure Concept:** \( Y = \{e_2, e_3, e_5, e_6\} \)

**Indiscernibility based on Quiz 3:**

\[
Y_1 = \{e_1, e_6, e_7, e_8\} \quad Des(Y_1) = \{Quiz3 = P\}
\]

\[
Y_2 = \{e_2, e_3, e_4, e_5\} \quad Des(Y_2) = \{Quiz3 = F\}
\]

**upper approximation:** \( \bar{Y} = \{e_1, e_6, e_7, e_8, e_2, e_3, e_4, e_5\} \)

**lower approximation:** \( \underline{Y} = \{\} \)

**discriminant index:** \( \alpha_{Q_3} = 1 - \frac{\mid \bar{Y} - \underline{Y} \mid}{\mid O \mid} = 0 \)

**DISCRIMINANT INDEX:** \( \alpha_{Q_3}(Y) = 1 - \frac{\mid \bar{Y} - \underline{Y} \mid}{\mid O \mid} \)

**discriminant index of Quiz 3** = 0.0
Indiscernibility based on Quiz 5:

\[ Y_1 = \{e_1, e_2, e_3, e_4, e_5, e_7, e_8\} \quad Des(Y_1) = \{Quiz5 = P\} \]
\[ Y_2 = \{e_6\} \quad Des(Y_2) = \{Quiz5 = F\} \]
lower approximation: \( \underline{Y} = \{e_6\} \)
upper approximation: \( \overline{Y} = \{e_1, e_2, e_3, e_4, e_5, e_7, e_8\} \)

DISCRIMINANT INDEX: \( \alpha_{Q_5}(Y) = 1 - \frac{|\overline{Y} - \underline{Y}|}{|\overline{Y}|} \)
discriminant index of Quiz 5 = 0.125

Indiscernibility based on Quiz 6:

\[ Y_1 = \{e_1, e_4, e_7\} \quad Des(Y_1) = \{Quiz6 = P\} \]
\[ Y_2 = \{e_2, e_3, e_5, e_6, e_8\} \quad Des(Y_2) = \{Quiz6 = F\} \]
lower approximation: \( \underline{Y} = \{\} \)
upper approximation: \( \overline{Y} = \{e_2, e_3, e_5, e_6, e_8\} \)

DISCRIMINANT INDEX is: \( \alpha_{Q_6}(Y) = 1 - \frac{|\overline{Y} - \underline{Y}|}{|\overline{Y}|} \)
discriminant index of Quiz 6 = 0.375

The following table shows the \( \alpha \) values of the three quizzes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>( \alpha ) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiz 3</td>
<td>0.00</td>
</tr>
<tr>
<td>Quiz 5</td>
<td>0.125</td>
</tr>
<tr>
<td>Quiz 6</td>
<td>0.375</td>
</tr>
</tbody>
</table>

Because Quiz 6 has the highest discriminant index that best determines its membership in the concept \( Y \), we obtain the first decision rule.

\[ Rule1 : \{Quiz6 = f\} \Rightarrow \{Final = f\} \]

The decision tree in Figure 6.1 is used here to find the new domain which is the same as the formula mentioned in the Distance Learning Algorithm:
Based on the decision tree, we obtain the horizontally reduced table as follows:

We then merge Quiz 6 with the rest condition attribute to find the highest discriminant indices. That produces two combinations: Quiz 3 and Quiz 6, Quiz 5 and Quiz 6. The following table shows the result of the second round of the program. The outcome is as follows:

Domain: $O = \{e_2, e_3, e_5, e_6, e_8\}$
Failure Concept: $Y = \{e_2, e_3, e_5, e_6\}$
Indiscernibility based on Quizzes 3 and 6:

- **Domain:** \( O = \{e_2, e_3, e_5\} \)
- **Failure Concept:** \( Y = \{e_8, e_9\} \)
- **Lower approximation:** \( Y = \{e_2, e_3, e_5\} \)
- **Upper approximation:** \( Y = \{e_2, e_3, e_5, e_6, e_8\} \)
- **DISCRIMINANT INDEX:** \( \alpha_Q(Y) = 1 - \frac{|Y - Y|}{|O|} \)
- **Discriminant index of Quiz 3 and 6:** 0.6

Indiscernibility based on Quizzes 5 and 6:

- **Lower approximation:** \( L(Y) = \{e_2\} \)
- **Upper approximation:** \( U(Y) = \{e_4, e_5, e_1, e_3\} \)
- **DISCRIMINANT INDEX:** \( \alpha_Q(Y) = 1 - \frac{|Y - Y|}{|O|} \)
- **Discriminant index of Quiz 5:** 0.20

The discriminant indices for both Quizzes 3, 6 and Quizzes 5,6 are shown in the following table:

<table>
<thead>
<tr>
<th>Show Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>***********************</td>
</tr>
<tr>
<td>Quizzes 3,6</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>***********************</td>
</tr>
</tbody>
</table>

Because the discriminant index of Quizzes 3 and 6 has the high value, we thus obtain the second rule.

**Rule2**: \( \{Quiz3 = f, Quiz6 = f\} \Rightarrow \{Final = f\} \)

We continue the decision tree in Figure 6.2.

Based on the decision tree, we find the further horizontally reduced information table, Table 6.5.
Because we have only one condition attribute and by merging Quizzes 5 and 6 with this attribute, the third rule is obtained.

\[ Rule 3 : \{Quiz 5 = f, Quiz 6 = f \} \Rightarrow \{Final = f\} \]

The condition attribute now is empty. We stop the while loop and have successfully obtained the rules.

**Strength Class**: We have the rules available now, but are we confident that online students should believe in these rules? The Strength class provides support for the rules. It finds the strength of each rule by using the following formula:

\[
\frac{\text{\# of positive cases covered by the rule}}{\text{\# of cases covered by the rule (including both positive and negative)}}
\]

The strength of the rules is held in an one-dimensional array and at the end of
the program it writes the result into a HTML file and posts this information about the rules on the Web. The output is shown in Figure 6.3.

Dear Web Learners:

The following rules are obtained based on the previous online students' records by using Rough Set theory. They present valuable information on the materials that caused students to fail the course. These materials are the core sections of this course and need extra effort in order to pass this class. Please be careful on those materials related to the Quizzes.

Rule1: \{Quiz6 = f\} \implies \{Final = f\}

This rule has a probability of 76.46%

Rule2: \{Quiz3 = f, Quiz6 = f\} \implies \{Final = f\}

This rule has a probability of 100.0%

Rule3: \{Quiz5 = f, Quiz6 = f\} \implies \{Final = f\}

This rule has a probability of 100.0%

Figure 6.3: Web Learning Guidance

The posting presents the information about the rules. A question arises: how should those rules be used and what do they mean? The rules are mainly used to measure previous online student performance, to guide repeating and new online students to focus their studies, and to provide information to the course instructor to modify or reorder the prerequisite online notes. The first rule has a probability of 56.52% that means if a student fails the Quiz 6, he or she has 76.46% possibility of
failing the final. The second and third rule say that if a student fails to understand the materials related to Quizzes 3 and 6 or Quizzes 5 and 6, then the student has a 100% possibility of failing the course. Therefore, the materials related to Quiz 3, 5, and 6 are the core materials in this online class.

The implementation works for any set of course results that are comparable to the CS100 and Java course examples used in the thesis. The essential requirement is that the course assignment and quizzes prepare the students for material covered on the final exam.
Chapter 7

Conclusion

This thesis discusses Distance Education, especially web-based computer science education using WebCT. One of the problems with this form of delivery of course materials is that of understanding the students’ ability to learn the material as presented. This thesis proposes to use the Rough Set based Distance Learning method to address this problem. The Rough Set based Distance Learning Algorithm (RSDLA) was implemented using the Java programming language. RSDLA helps people overcome the lack of student/teacher feedback with WebCT delivery, making Web based learning more effective. Rough Set Based WebCT Learning permits decision rules to be induced that are important to both students and instructors. It thus guides students in their learning. For repeating students, it specifies the areas they should focus on according to how they fit the rules. For new students, it tells them which sections need extra effort in order to pass the course. For example, if Quiz 6 is failed, there is a 56 percent chance that the student will fail the final. Therefore, students are made aware of the importance of Quiz 6 related materials. Rough Set Based WebCT can also guide the instructor about the best order in which to present the notes. Based on this analysis, the instructor may reorganize or rewrite the course notes by providing more examples and explaining some concepts in more detail.

Rough Set Based Distance Learning improves the state-of-the-art of Web learning by offsetting the lack of student/teacher feedback and providing both students and teacher with the insights needed to study better or improve course content.

In the present work, we just distinguish two values: pass and fail. It would be
useful to have more values such as grade levels of A, B, C, D and F. In order to support such grade levels, the Dominance-based Rough Set approach [19, 20, 21, 43] has to be used. The author discussed this potential enhancement with Drs. Roman Slowinski and Salvatore Greco at the RSCTC 2000 international conference in Banff and intends to pursue it. Identifying near passes on quizzes would add another useful attribute. Some relevant works in this respect include [50].

Thus, our next step will be to extend the RSDLA to achieve finer results. Instead of converting marks to “P” and “F”, we can convert them to “GP,” “P,” “GF,” and “F” to represent the performance of students where GP is in a range of 100 to 80, P is 50 to 79, GF is 40 to 49, and F is below 40. This will allow the RSDLA to incorporate information about near passes and near failures.
Bibliography


