SYSTEMS INC

B921014-U-2R03

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - THE LEARNING SYSTEM (TASKS C AND D)

NAS5-30127 /N-43-CR

26 September 1992

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

Prepared by:

JJM Systems, Inc.
One Ivybrook Boulevard, Suite 190
Ivyland, PA 18974

TABLE OF CONTENTS

JJM SYSTEMS INC

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

JJM SYSTEMS INC

LIST OF ACRONYMS

KEE	Knowledge Engineering Environment
VEG	VEGetation Workbench

JJM SYSTEMS INC

SECTION 1.0

INTRODUCTION

The VEGetation Workbench (VEG) is an expert system that infers vegetation characteristics from spectral reflectance data. The first generation expert system has been implemented using the Knowledge Engineering Environment (KEE) by Intellicorp. VEG is described in references 1 and 2. VEG contains more than one thousand KEE units. Some units hold samples of reflectance data that are used for testing and demonstrating the system. Other units contain the methods and rules used for processing the data and the graphics required for the interface. When VEG is used to infer vegetation characteristics, additional units are created to hold both the intermediate and final results of processing the data.

VEG includes a data base of historical cover type data. This consists of data for 18 different cover types and, for each cover type, this includes data collected at several different solar zenith angles and wavelengths. In the first generation version of VEG, this data base was used to estimate the error term in calculations made by VEG. For example, when the spectral hemispherical reflectance of an unknown cover type was being calculated, the historical cover type data that best matched the unknown cover type was identified and interpolated and extrapolated to match the unknown sample. Techniques were applied to the unknown cover type data to estimate the spectral hemispherical reflectance. The same techniques were also applied to the historical cover type data. The spectral hemispherical reflectance of the historical cover type data was available in the data base so the error in the calculation could be calculated. The error obtained by applying the techniques to the historical cover type data gave an estimate of the error term involved in applying the techniques to the unknown cover type data.

This report describes the implementation of a learning system that uses the data base of historical cover type data to learn class descriptions of classes of cover types. These classes can include broad classes such as soil or vegetation or more specific classes such as forest, grass and wheat. The classes can also include subclasses based on continuous parameters such as 0-30% ground cover, 31-70% ground cover and 71-100% ground cover. The learning system is designed to handle any combination of directional view angles such as (0 0), (30 50), (45 60), (10 135), (40 225) where the first value in each pair is the zenith angle and the second value is the relative azimuth angle. The learning system uses sets of positive and negative examples from the data base of historical cover types to find the most important features that uniquely distinguish each class. For example, the learning system found that for solar zenith 45°, wavelength 0.68 µm and the view angles (15 182), (75 90), (0 0) and (35 45), the class 0-30% ground cover was best distinguished by the following hypotheses:-

```
((FIRST-MAX (75 90)) NIL)
((FIRST-MAX (15 182)) T)
((GREATER-THAN (15 182) (75 90)) T)
```

The first hypothesis says that the maximum reflectance value is not at the view angle (75 90). The second hypothesis says that the maximum reflectance value is at the view angle (15 182). The third hypothesis says that the reflectance at the view angle (15 182) is greater than the reflectance at the view angle (75 90). These hypotheses describe the class 0-30% ground cover for the solar zenith angle, view angles and wavelength specified. In a typical run of the learning system, class descriptions for several alternative classes such as the classes 0-30%, 31-70% and 71-100% ground cover are learned from the data base of historical cover types. These class descriptions are then used to classify an unknown cover type by finding the class that best matches the unknown cover type data.

VEG contains more than one thousand units (objects). In order to conserve memory, the learning system was developed as a separate layer sitting on top of VEG. It is only loaded when needed. VEG does not need the learning system to achieve its various goals. The learning system is fully integrated with VEG and uses some VEG objects in its processing.

Tasks C and D have been completed. The learning system has been implemented and fully integrated into the first generation version of VEG. The learning system is described in detail in this report. A Sun cartridge tape containing KEE and Common Lisp code for the learning system has been delivered to the NASA GSFC technical representative.

SECTION 2.0

OVERVIEW OF THE LEARNING SYSTEM

The learning system has been implemented as a separate knowledge base from VEG. The learning system is loaded only when it is used. It is invoked by selecting the option LEARN.CLASS.DESCRIPTION from either the VEG "Research Mode" or the VEG "Automatic Mode" top level menu. Only when one of these options has been selected, is the learning system loaded. If either of these options is subsequently selected again, the learning system knowledge base is not re-loaded. The learning system is not stand-alone. It requires that VEG be loaded first.

The organization of KEE units in the learning system knowledge base is shown in Figure 2-1. This figure shows the general organization of the learning system. The learning system is composed of three basic components: data bases, learning methods and learning rules. The units whose names end in .AV are ActiveValue units. They are attached to slots in other units and they contain methods that are activated when values are added to or removed from the slots to which they are attached. The use of specific ActiveValue units will be discussed below. The rule class LEARNING.RULES contains rules which determine which hypotheses to test for each training class.

Several units such as ENTER.LEARNING.DATA and OPTION.3 are positioned in the hierarchy of KEE units in the learning system as members of the class LEARNING.METHODS. This organization of units is for convenience only. The member units of the class LEARNING.METHODS are grouped together because they have similar uses. However, they are not strictly members of the class LEARNING.METHODS, because they do not inherit any slots or slot values from the class unit LEARNING.METHODS. The member units of the class LEARNING.METHODS contain slots required by the different methods involved in processing the learning data. For example, the slots CLASS.PARAMETER, DIRECTIONAL. VIEW.ANGLES, SOLAR.ZENITH, VALUE and WAVELENGTH in the unit ENTER. LEARNING.DATA are used to hold data entered via the interface. The slots in the unit ENTER.LEARNING.DATA are shown in Figure 2-2.

The unit LEARNING.METHODS contains no member slots that are inherited by its member units. However, this unit contains a number of own slots as listed in Figure 2-3. Some of these slots are used to ensure that the methods in the learning system are executed in the correct sequence. For example, at the beginning of a run, the slot DONE.ENTER.LEARNING.DATA.P has the value NIL. After data has been entered into the learning system, the value of this slot is changed to T. The next method in the processing of the learning data checks the value of this slot and only proceeds if the slot has the value T, indicating that the necessary data has been entered.

When a training class has been defined, a subclass of the unit TRAINING.DATABASES is set up. Data defining the class is stored in the new training problem unit. The positive and negative training sets for the class are subsequently set up as a hierarchy of units which are subclasses of the training problem unit. This is discussed in detail in Section 3. The positive and negative training set units inherit slots from the unit TRAINING.DATABASES. This unit also contains own slots which contain data that is common to all the training problems. For example, the slot MINIMUM.SET.SIZE holds the minimum acceptable training set size. Figure 2-4 shows the slots in the unit TRAINING.DATABASES.

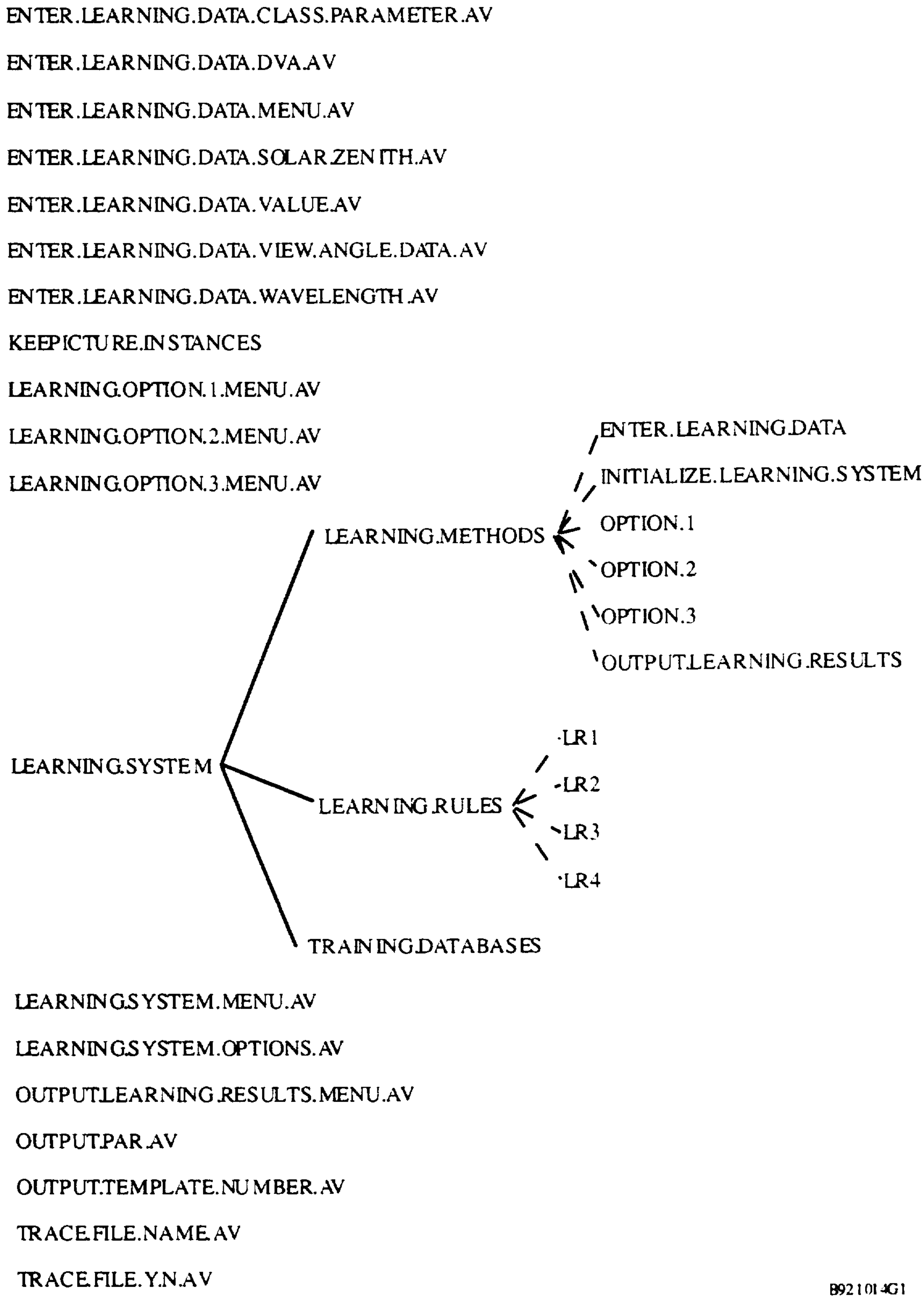


Figure 2-1

The Organization of KEE Units in the Learning System

CLASS
CLASS.PARAMETER
DIRECTIONAL.VIEW.ANGLES
MENU
MESSAGE
POSSIBLE.COVER.TYPES
POSSIBLE.DESCRIPTIONS
SOLAR.ZENITH
VALUE
VIEW.ANGLE.DATA
VIEW.ANGLE.DATA.MESSAGE
WAVELENGTH

B921014G1

Figure 2-2

The Slots in the Unit ENTER.LEARNING.DATA

DONE.CLASSIFY.COVER.TYPES.P
DONE.CLASSIFY.SAMPLE.P
DONE.ENTER.LEARNING.DATA.P
DONE.LEARN.CLASS.DESCRIPTIONS.P
MESSAGE
OPTION.NUMBER
SEARCH.DEPTH
SEARCH.OVER
TRACE.FILE
YES.NO

B921014G3

Figure 2-3

Own Slots in the Unit LEARNING.METHODS

Member Slots : BEST.CLASS BEST.SCORE CLASS CORRECT.MATCHING.COVER.TYPES COVER.TYPE HYPOTHESES INCORRECT.MATCHING.COVER.TYPES NEG.TRAINING.SET POS.TRAINING.SET PREVIOUS.BEST.SCORE REFLECTANCE.DATA SAMPLE.SCORE SCORES SOLAR.ZENITH VIEW.ANGLE.DATA VIEW.ANGLE.DATA.MESSAGE WAVELENGTH Own Slots : CURRENT.CLASSES MAXIMUM.SET.SIZE MINIMUM.SET.SIZE NUM.SCORES PERFORMANCE.SCORE

B921014G4

Figure 2-4

Slots in the Unit TRAINING.DATABASES

The KEE system contains a graphics package called "ActiveImages". This package was used to build the interface for VEG. Using this package, a comprehensive interface was built for the learning system, and this was fully integrated into the existing VEG system. The interface allows the scientist to run VEG and select options at all stages of the run by clicking the mouse over the appropriate menu option. The interface allows a scientist with no knowledge of KEE, Common Lisp or the detailed structure of VEG to use the system with ease. The only time that the scientist needs to use the keyboard during a run is when he/she enters new data manually. All other operations are controlled by the mouse. The interface allows the scientist to focus on the data and the functions performed by VEG. It abstracts away most of the underlying detailed complexity of the VEG system.

In the VEG "Research Mode," the learning system is invoked by selecting the option LEARN.CLASS.DESCRIPTION from the top level menu, as shown in Figure 2-5. Mousing on the option SELECT.OPTION causes the learning system to be loaded (if it has not already been loaded). The learning system main menu then appears on the screen. This menu is shown in Figure 2-6. If the user mouses on one of the option numbers, a brief description of the option appears in the box below the option menu. If the user then mouses on SELECT.OPTION, the option is selected and the menu for the selected option replaces the learning system main menu on the screen. The operation in the VEG "Research Mode" of each of the options in the learning system is discussed in detail in the next section. The operation of the learning system in the VEG "Automatic Mode" is discussed in Section 4.

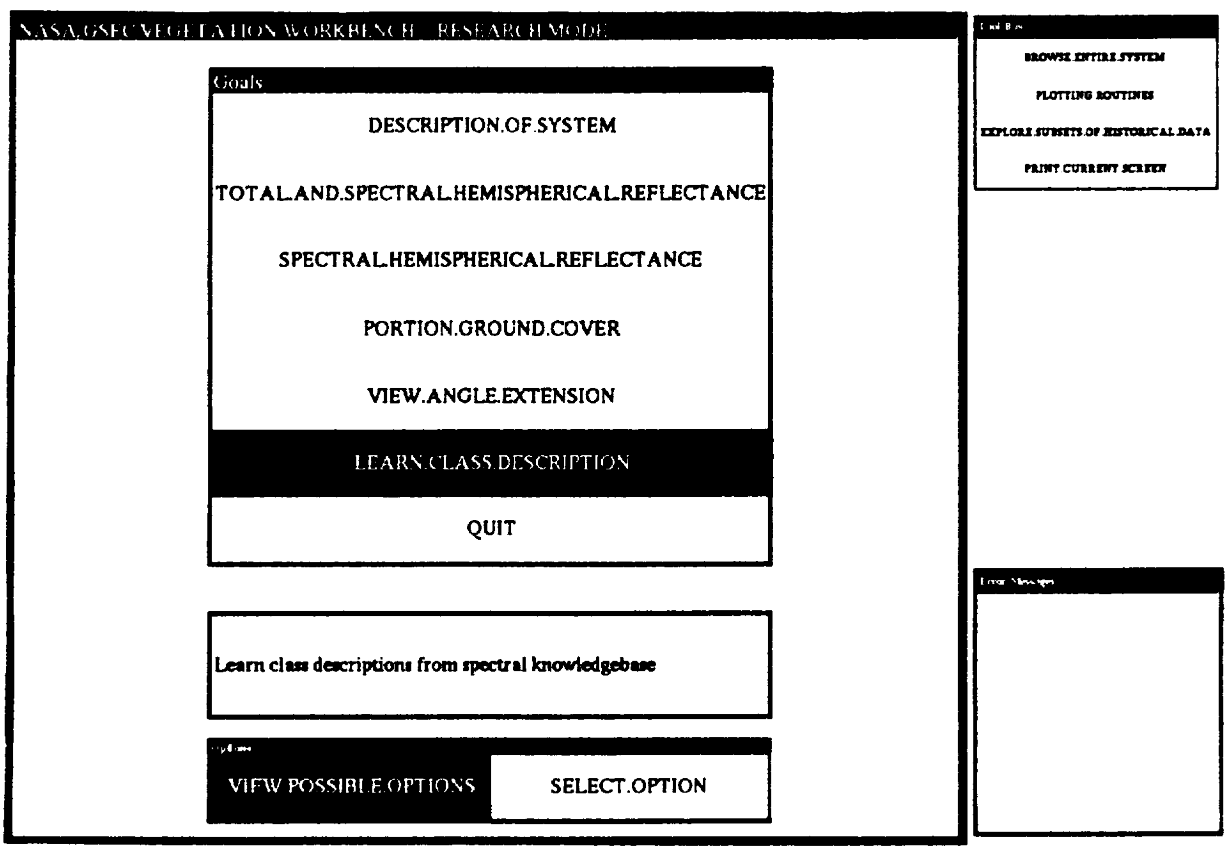


Figure 2-5

VEG "Research Mode" Top Level Menu

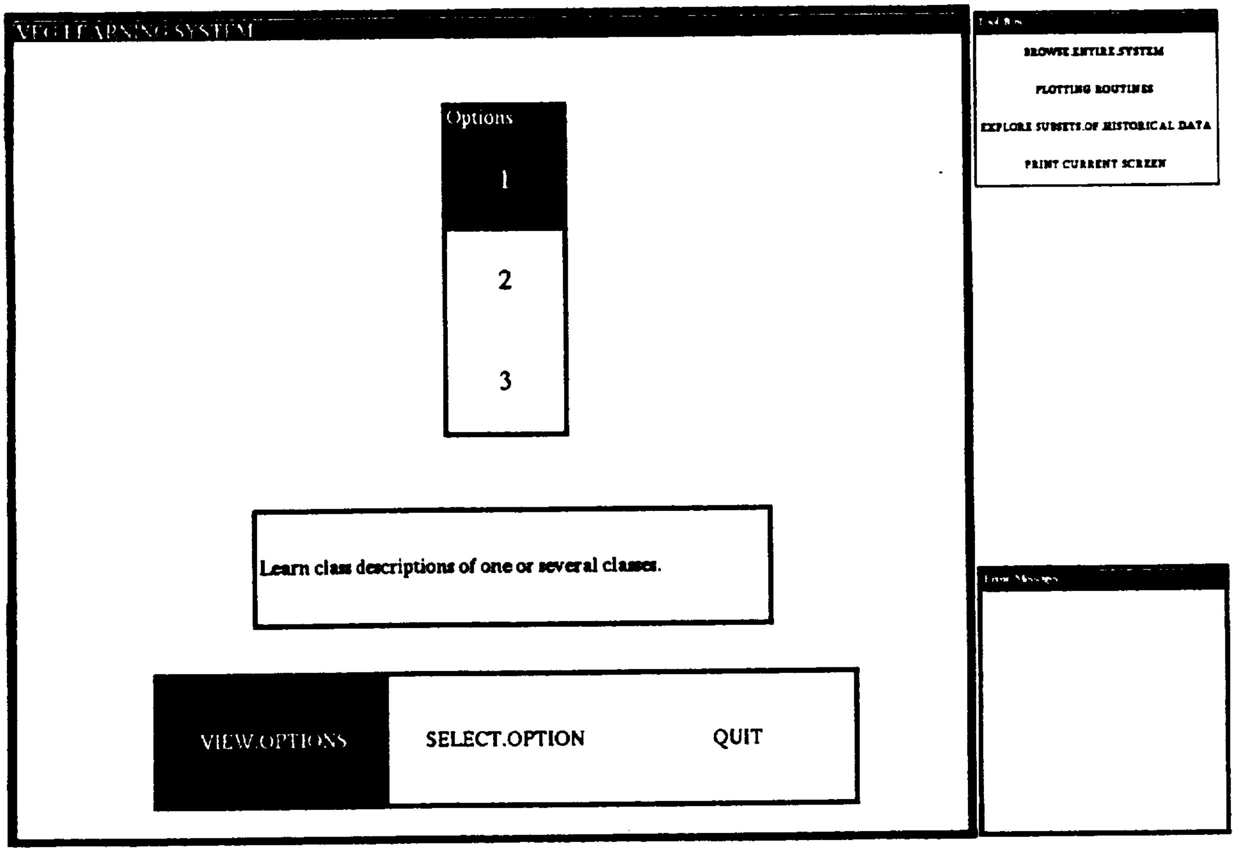


Figure 2-6

The Learning System Main Menu

SECTION 3.0

DETAILED DESCRIPTION OF THE LEARNING SYSTEM IN THE VEG "RESEARCH MODE"

In the VEG "Research Mode," the learning system presents the user with three different options. In option 1, the system learns class descriptions for one or more classes. In option 2, the system learns class descriptions for one or more classes and then uses the learned classes to classify an unknown sample. Option 3 allows the user to test the system's classification performance. In this option, the system learns class descriptions for one or more classes and then classifies the appropriate samples in the data base. The percentage of correctly classified samples is then used to summarize the degree of classification accuracy achieved by the learning system. All three options in the learning system are described in detail in this section.

3.1 OPTION 1

In option 1, the user enters data to define one or more training problems. The system then learns the class descriptions for the training problems. Finally, the results are output on the screen and the user has the option of writing the results to a file. Figure 3-1 shows the menu for Option 1. The user selects each step by mousing on the appropriate option in the learning system option 1 menu.

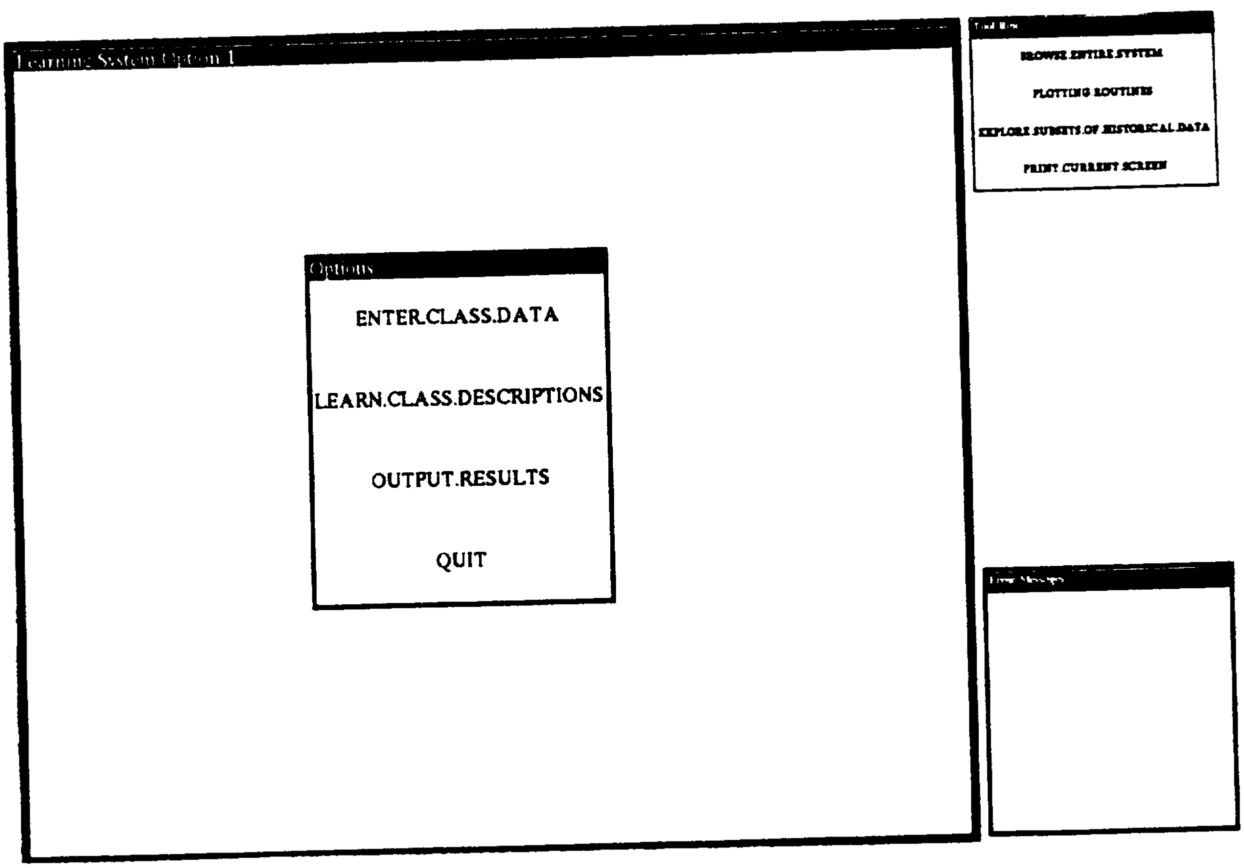


Figure 3-1

The Menu for Option 1

If the user attempts to select the menu options in the wrong order, an error message (lower right) is displayed in the Error Message box. Figure 3-2 illustrates the case when the user attempted to output the results before the system had learned the class descriptions. The steps involved in Option 1 are described in detail in this subsection.

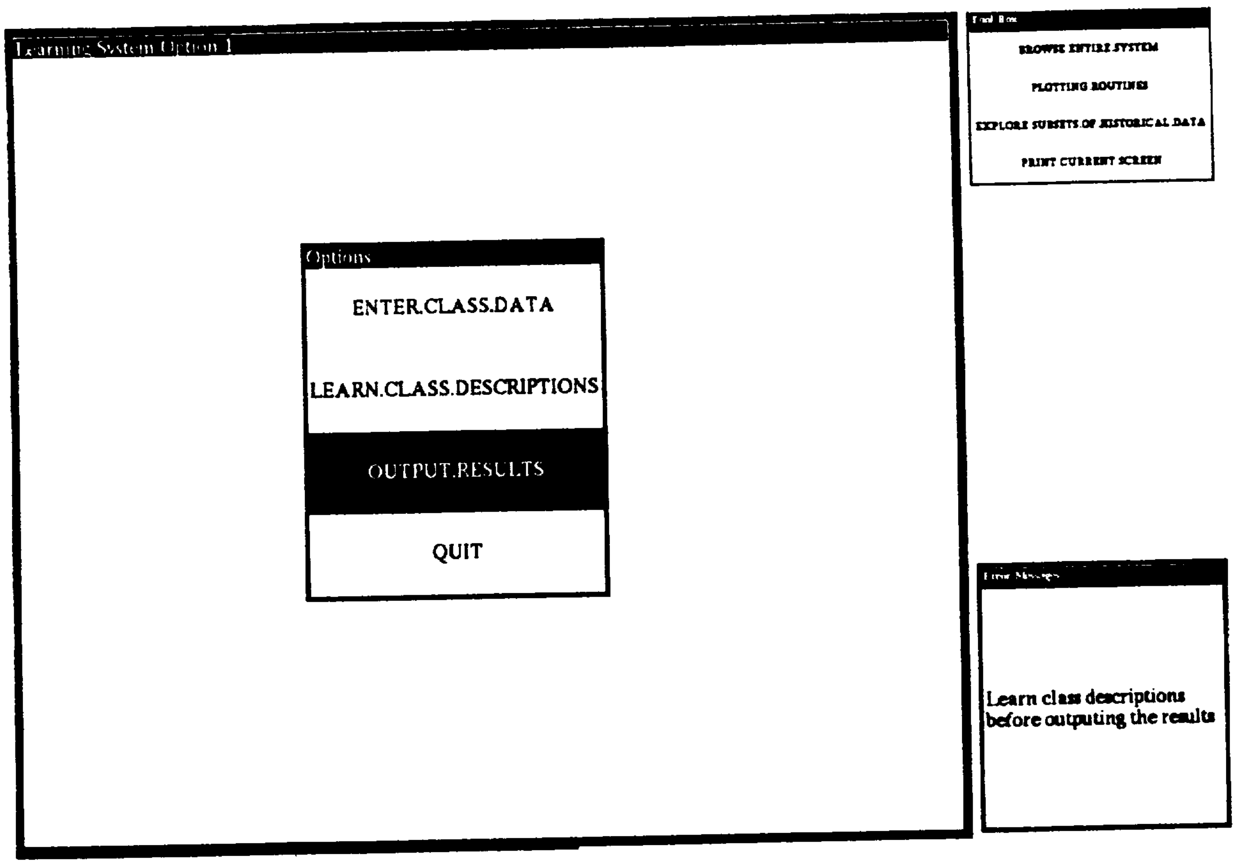


Figure 3-2

An Error Message is Displayed if Steps are Selected Out of Sequence

SYSTEMS INC

3.1.1 Enter Learning Data

When the user selects the option ENTER.CLASS.DATA from the option 1 menu, any previously entered training problems are deleted, and the "Enter Learning Data" interface is opened. This interface allows the user to enter data to define the training problems that are to be investigated. It is shown in Figure 3-3. The data is stored temporarily in slots in the unit ENTER.LEARNING.DATA that was shown in Figure 2-2. When the user enters a value for the solar zenith, the value is stored in the SOLAR.ZENITH slot of this unit. The ActiveValue unit ENTER.LEARNING.DATA.SOLAR.ZENITH.AV is attached to the slot. When the value of the SOLAR.ZENITH slot is changed, a Lisp function in the ActiveValue unit is executed. This function checks that the solar zenith is between 0 and 90. If the value is out of range, an error message is displayed and the value is not retained.

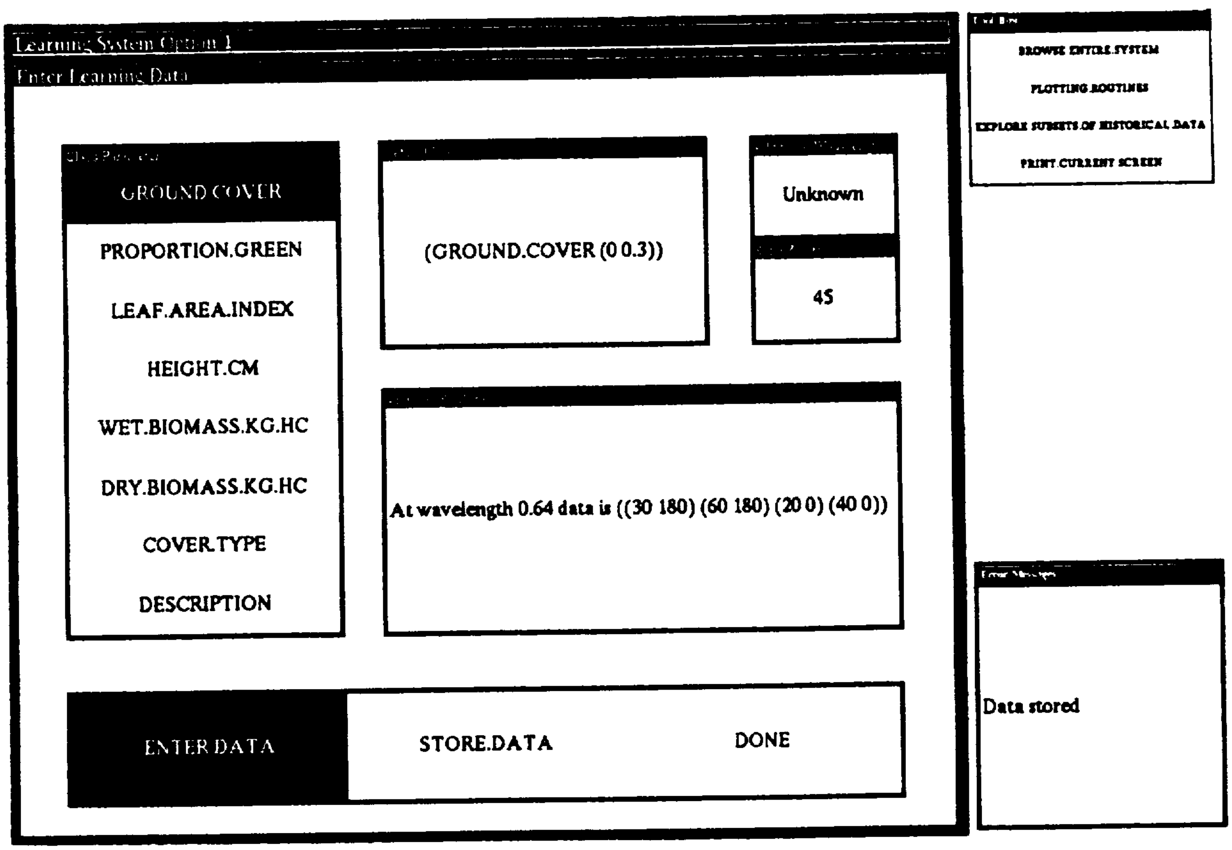


Figure 3-3

The Enter Learning Data Interface

The learning system allows the user to define a training problem consisting of more than one wavelength, with a different set of view angles at each wavelength. If the user enters a value into the box labelled "Additional Wavelength" in the interface, this value is stored and the Lisp function in the unit ENTER.LEARNING.DATA.WAVELENGTH.AV is executed. This function checks that the value for the wavelength is in range. If it is in range, an additional window is opened. This window prompts the user to enter the directional view angles for the new wavelength. The window is shown in Figure 3-4. The directional view angles are stored in the slot DIRECTIONAL.VIEW.ANGLES of the unit ENTER.LEARNING.DATA. An ActiveValue is attached to the slot in order to validate the view angle data. Once valid view angle data has been entered, the window is closed and a list such as (0.64 ((0 0)(30 180)(60 180))) is constructed from the data. This list means that at wavelength 0.64 μm, the directional view angles were (0 0), (30 180) and (60 180). This list is stored in the slot VIEW.ANGLE.DATA. A message is also constructed from this data, stored in the slot VIEW.ANGLE.DATA.MESSAGE and displayed in the box labelled "View Angle Data" in the interface, shown in Figure 3-3.

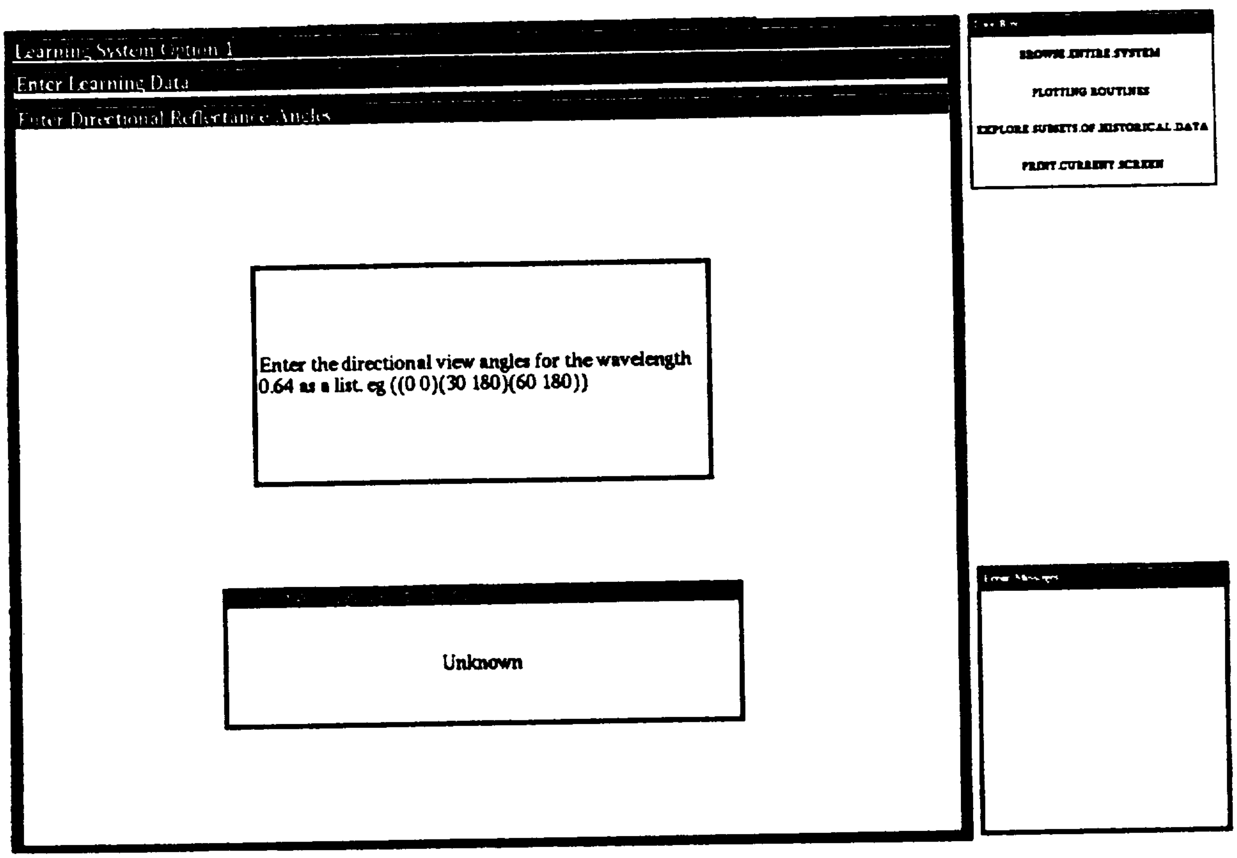


Figure 3-4

Window Through which Directional View Angles are Entered

In order to define the class whose description is to be learned, the user first selects a parameter by mousing on the Class Parameter menu. The window shown in Figure 3-5 then opens. This window prompts the user to define the class. In the case of a continuous parameter such as GROUND.COVER, the window prompts the user to enter the maximum and minimum values for the class as a list and informs the user of the range of possible input values. In the case of a discrete parameter such as DESCRIPTION, the screen displays the possible values of the parameter and prompts the user to enter the value for the parameter in the class. Figure 3-6 shows an example of this case. For example, if the parameter is DESCRIPTION, the class might be FOREST. An ActiveValue once again checks the validity of the entered data and prompts the user to enter the data again if it is invalid. Once valid data has been entered, the window is closed. A list such as (GROUND.COVER (0 0.3)) is constructed, stored in the slot CLASS and displayed on the screen as shown in Figure 3-3. This example represents the class of 0-30% ground cover. Additional class parameters can then be defined if necessary. For example, a class might be defined as forest with 70-100% ground cover.

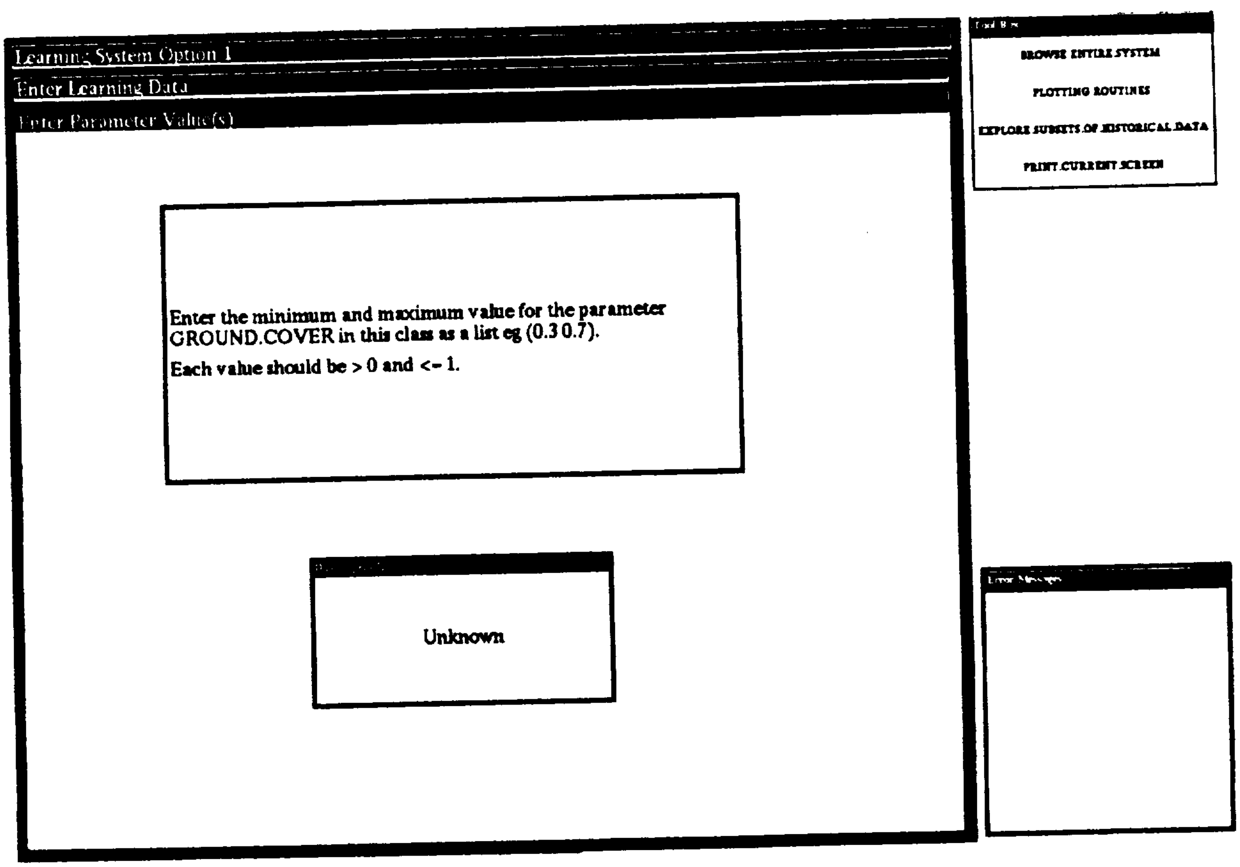


Figure 3-5

Defining a Ground Cover Class

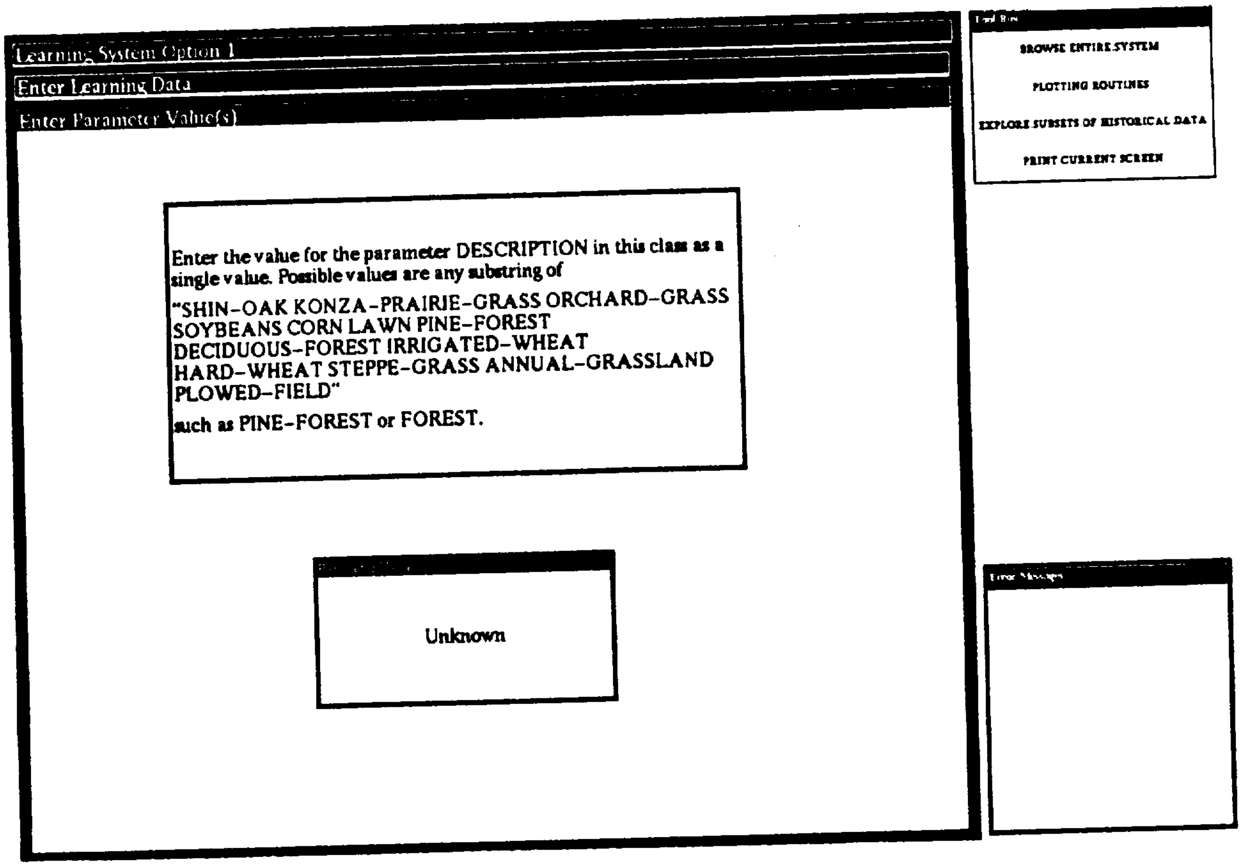


Figure 3-6

Defining a Class for a Discrete Description

When all the data for a training problem has been entered, the user can mouse on STORE.DATA. The system checks to make sure that a complete set of data has been entered. If the data is incomplete, the user is prompted to enter the missing data. If the data is complete, a new unit is created as a subclass of the unit TRAINING.DATABASES. The solar zenith, wavelengths and associated view angles and class definition for the problem are copied into the new training problem unit. The user can then select DONE if all training problems have been entered, or he/she can enter data for additional classes such as 31-70% ground cover and select the option STORE.DATA to store the data for each additional training problem. When the user selects the option "DONE," the "Enter Learning Data Interface" is closed and the menu for Option 1 appears on the screen again.

3.1.2 Learn Class Descriptions

The second step in option 1 is for the system to learn the class descriptions for the classes that were defined in the previous step. Learning the class descriptions involves several steps. First, the system uses the data base of historical cover types to set up the positive and negative training sets for each problem. Rules are run to determine the set of possible hypotheses for each problem. Next, the hypotheses are tested on the training sets to determine the discrimination score for each hypothesis. The scores are sorted in order to determine the best discriminating hypothesis for each training problem. Finally, compound hypotheses containing two or more hypotheses are constructed. These are tested in order to determine the best discriminating compound hypotheses for each training problem. The steps involved in learning class descriptions are described in detail in this subsection.

The process of learning class descriptions can take several minutes, especially when the training problem has a large number of view angles. For this reason, the message "Learning class descriptions..." is displayed at the beginning of the process. When the process has ended, this message is replaced by the message "Finished learning class description".

The first step in learning the class descriptions is to generate the training sets. The system searches the data base of historical cover types in the VEG knowledgebase and finds the cover types that best match the training problem. A cover type matches the training problem if it has data at all the wavelengths specified in the training problem, if its solar zenith is close to the training problem solar zenith and if it has a value for every parameter specified in the class definition. For example, a cover type that has no value in its ground cover slot cannot be included in either of the training sets for the class 0-30% ground cover. Once a matching cover type has been identified, the values in the slots for each parameter in the class definition are examined. If the cover type data fits the class definition, the name of the cover type is stored in the slot POS.TRAINING.SET of the training problem unit. Otherwise, it is stored in the slot NEG.TRAINING.SET. In the first search through the data base, each matching cover type whose solar zenith is within 10% of the training problem solar zenith is identified and added to the appropriate training set slot. If insufficient cover types have been found for the training sets, the search is then repeated. In the second search, matching cover types whose solar zenith is within 20% of the training problem solar zenith are identified. The process of increasing the bounds on the solar zenith and searching through the data base is continued until either the positive or negative training set exceeds the maximum permissible size, both training sets exceed the minimum permissible size or the bounds have increased to $\pm$ 100%. If, when the search ends, either training set is found to be empty, a message is displayed on the screen and the process of learning class descriptions is stopped. Figure 3-7 show this case.

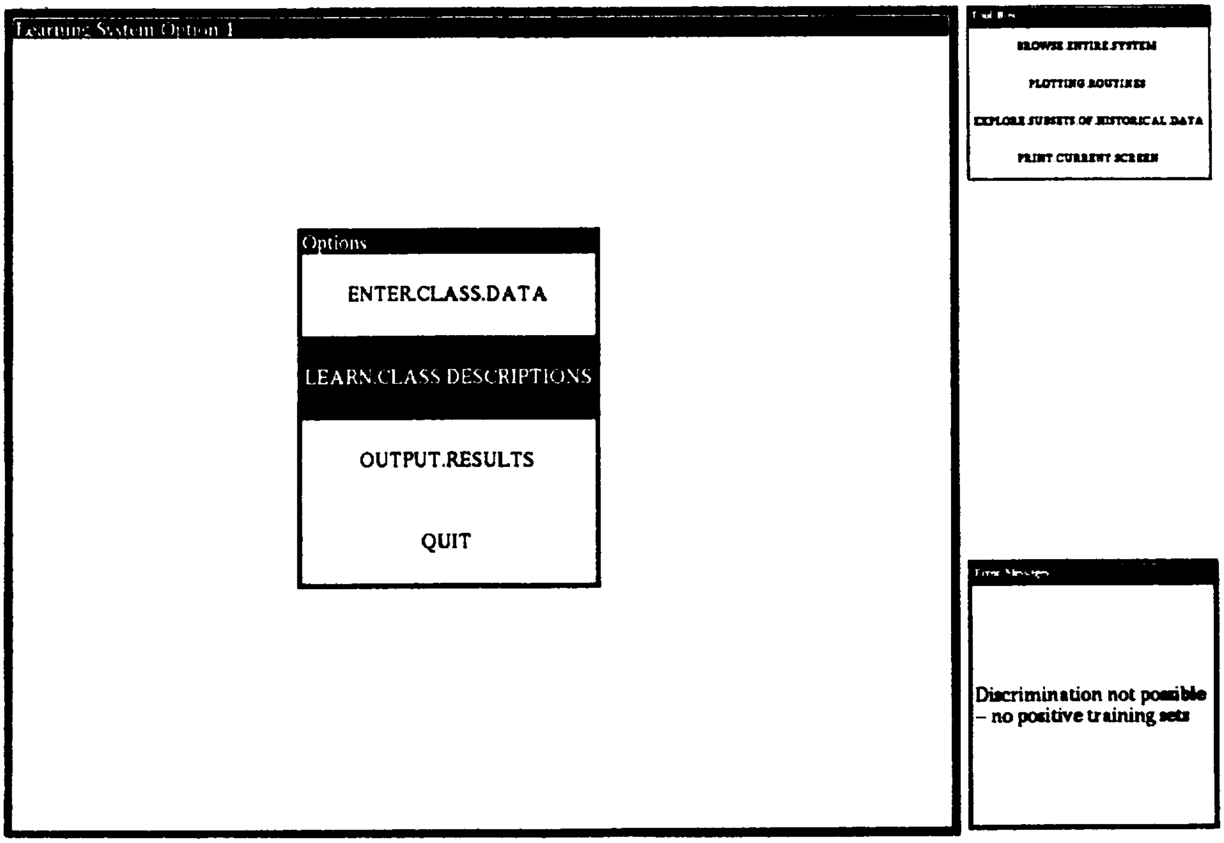
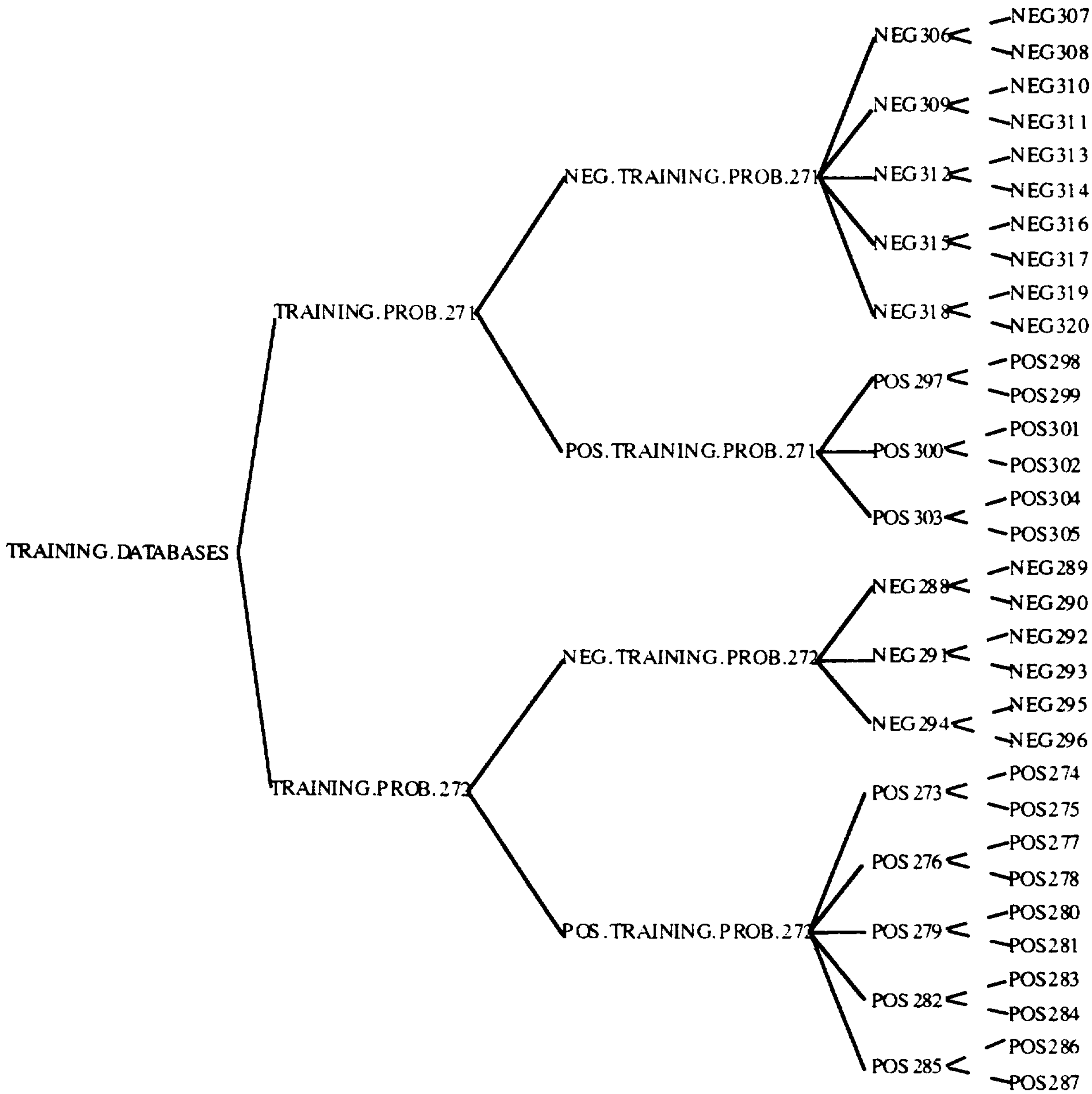


Figure 3-7

Learning Class Descriptions is Terminated if Either Training Set is Empty

Once the cover types that will make up the positive and negative training sets for each training problem have been identified, a hierarchy of units is created to hold the training set data. Figure 3-8 illustrates the hierarchy of units in the training sets for two training problems named TRAINING.PROB.271 and TRAINING.PROB.272. In this figure, the size of the training sets has been reduced so that the entire hierarchy will fit on the page. The learning system is usually run with a minimum training set size of 8 units. The class units NEG.TRAINING.PROB.271 and POS.TRAINING.PROB.271 are the parent units for the negative and positive training sets respectively for TRAINING.PROB.271. A subclass (such as the unit POS297) of the unit POS.TRAINING.PROB.271 is created corresponding to each cover type in the positive training class of the training problem and the name of the cover type is stored in this unit. In the problem illustrated in Figure 3-8, each training problem contains a set of view angles at two different wavelengths. Two member units, such as POS298 and POS299, are created for each subclass unit -- one corresponding to each wavelength in the training problem. The raw reflectance data from the cover type data at the appropriate wavelength is interpolated and extrapolated to match the view angles in the training problem at each wavelength. This is then stored in the appropriate member unit. The wavelength is also stored in this unit so that, later in the processing, the system can determine which data corresponds to each wavelength.



B921014G5

Figure 3-8

The Hierarchy of KEE Units in the Training Sets for Two Training Problems

Once the training sets have been set up, rules in the rule class LEARNING.RULES are run in order to determine the set of possible hypotheses that can be constructed for the data in each training set. The rules are shown in Figure 3-9. The left-hand side of each rule tests the view angle data. If the rule fires, the appropriate lisp function is called. Each lisp function generates hypotheses which are added to the HYPOTHESES slot of the training problem unit.

The rule LR.1 fires if the view angle data at a particular wavelength contains at least two view angles. The right-hand side of this rule calls the lisp function TRY-DIRECTION-RELATIONSHIPS which generates direction relationships for every possible pair of view angles in the data and adds these to the HYPOTHESES slot of the training problem unit. An example of a direction relationship that might be generated by this function is

(GREATER-THAN 0.64 (60 180) (30 180))

This relationship represents the hypothesis that at wavelength 0.64 µm, the reflectance at the view angle(60 180) is greater than the reflectance at view angle (30 180).

If the view angle data at a particular wavelength contains at least three view angles, the rule LR.2 fires and the lisp function TRY-MAX-MIN-RELATIONSHIPS is called. This function generates relationships such as (FIRST-MAX 0.64 (60 180)) and (FIRST-MIN 0.64 (60 180)) for each angle in the view angle data and adds them to the HYPOTHESES slot of the training problem unit. The relationship (FIRST-MIN 0.64 (60 180)) represents the hypothesis that at wavelength 0.64 µm the minimum value of the reflectance is at the view angle (60 180).

The rule LR.3 fires if the view angle data at a wavelength contains at least four view angles. The right-hand side of this rule calls the function TRY-SECOND-MAX-MIN-RELATIONSHIPS which generates relationships such as (SECOND-MAX 0.64 (60 180)) and (SECOND-MIN 0.64 (60 180)) for every angle in the view angle data. These relationships are also added to the HYPOTHESES slot of the training problem unit. The relationship (SECOND-MAX 0.64 (60 180)) represents the hypothesis that the second highest reflectance value in the data is at the view angle (60 180).

The left-hand side of rule LR.4 calls the lisp function FULL-STRING-DATA-IN-PLANE. This function returns T if the view angle data consists entirely of one full string in one azimuthal plane and NIL otherwise. Note that if the data contains more than one full string, the function returns NIL. If this rule fires, the lisp function TRY-BACKSCATTER>FORWARDSCATTER-RELATIONSHIP is called. This function adds a relationship such as (BACKSCATTER>FORWARDSCATTER 0.64) to the HYPOTHESES slot of the training problem unit. This relationship represents the hypothesis that at wavelength 0.64 µm the average reflectance value in the backscatter data is greater than the average reflectance value in the forwardscatter data.

When the forward chaining of the rules has been completed, the set of all possible separate hypotheses for each training problem has been stored in the HYPOTHESES slot of the training problem unit. If the training problem contains data at more than one wavelength, the set of hypotheses may contain hypotheses at each wavelength. The current system does not permit a single hypothesis to refer to more than one wavelength, but the system has been designed to facilitate the addition of this capability at a later date.

```
(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (> (LENGTH (SECOND ?D))
            1))
   THEN
   (LISP(TRY-DIRECTION-RELATIONSHIPS ?TRAINING.PROB ?D)))

RULE : LR1


(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (>= (LENGTH (SECOND ?D))
             3))
   THEN
   (LISP (TRY-MAX-MIN-RELATIONSHIPS ?TRAINING.PROB ?D)))

RULE : LR2


(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (>= (LENGTH (SECOND ?D))
             4))
   THEN
   (LISP (TRY-MAX-MIN-RELATIONSHIPS ?TRAINING.PROB ?D)))

RULE : LR3


(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (FULL-STRING-DATA-IN-PLANE (SECOND ?D)))
   THEN
   (LISP
    (TRY-BACKSCATTER>FORWARDSCATTER-RELATIONSHIP ?TRAINING.PROB ?D)))


RULE : LR4
```

B921014 G6

Figure 3-9

The Rules in the Learning System

At the end of the run, the set of the best single hypotheses and the set of the best compound hypotheses for each training problem are displayed on the screen. The scientist can choose to run the system for compound hypotheses combining up to five single hypotheses (level 5). The scientist might also choose to trace the testing process. The trace provides a file consisting of the set of best compound hypotheses at each level. The purpose of this is to allow the scientist to look at predictive improvements with various combinations from the set of single hypotheses.

When the set of all possible hypotheses for each training problem has been generated, the user is asked whether the hypothesis testing should be traced. The screen shown in Figure 3-10 is displayed. If the user left clicks on "YES", the screen shown in Figure 3-11 is displayed, prompting the user to enter the name of the trace file. If the named file already exists, the user is asked to confirm that the file can be overwritten, or enter a new file name. The screen shown in Figures 3-10 and 3-11 is then closed.

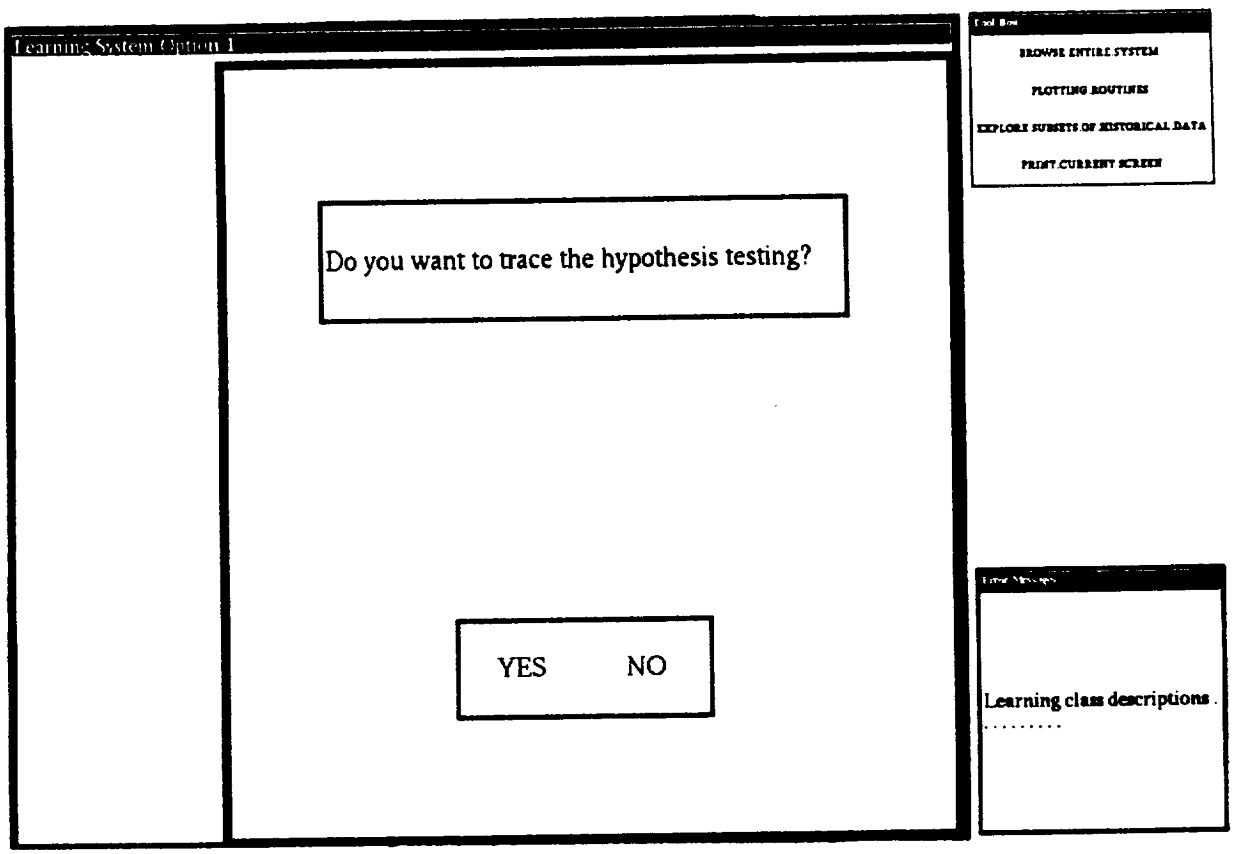


Figure 3-10

The Screen that Asks the User whether the Hypothesis Testing Should be Traced

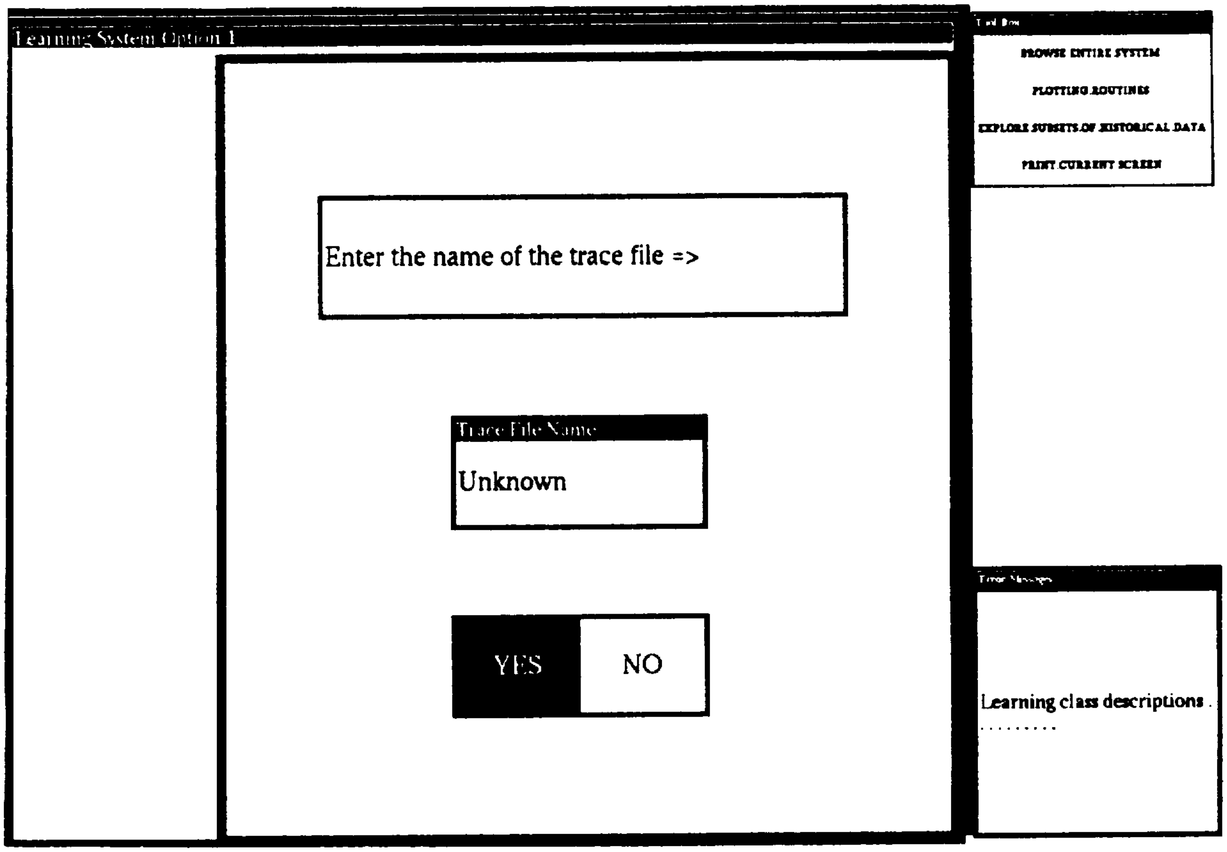


Figure 3-11

Entering the Name of the Trace File

The next step in learning the class descriptions is to determine the discrimination score for each separate hypothesis. Each hypothesis such as (GREATER-THAN 0.64 (60 180)(30 180)) is tested on each sample in the positive and negative training sets. The sample score is 1 if the hypothesis is true and 0 otherwise. The discrimination score is calculated as

$$\left[\frac{1}{p}\sum_{i=1}^{p} S_i\right] - \left[\frac{1}{n}\sum_{j=1}^{n} S_j\right] \qquad (1)$$

where each sample score is S, S_i is the ith positive sample score, S_j is the jth negative sample score, p is the number of samples in the positive training set and n is the number of samples in the negative training set. Thus a discrimination score of 1 for a hypothesis represents the case where the hypothesis is true for all samples in the positive training set and false for all samples in the negative training set, i.e., 1-0=0. This represents perfect discrimination. A score of 0 is the break even point where there is no effective discrimination between the positive and negative training sets, e.g. 0-0=0 or 0.5-0.5=0 or 1-1=0. A score of less than zero for a hypothesis represents the case where the hypothesis is true for more samples in the negative training set than in the positive training set. In this case, the converse of the hypothesis would yield a positive discrimination score. For each hypothesis such as (GREATER-THAN 0.64 (60 180)(30 180)) two separate scores are calculated. The order of the elements is re-ordered and two scores such as

(((((GREATER-THAN (60 180)(30 180)) T) 0.64)) 0.4) (2)

and

(((((GREATER-THAN (60 180)(30 180)) NIL) 0.64)) -0.4) (3)

are reported. In this example, the score (((((GREATER-THAN (60 180)(30 180)) T) 0.64)) 0.4) means that the hypothesis that the reflectance at angle (60 180) is greater than the reflectance at angle (30 180) for the wavelength 0.64 µm produced a discrimination score of 0.4. The discrimination score in (2) is calculated directly by testing the hypothesis (GREATER-THAN 0.64 (60 180)(30 180)) on all the data in the positive and negative training sets. The discrimination score in (3), -0.4, is calculated as minus one multiplied by the discrimination score in (2). Scores such as (2) and (3) are calculated for each hypothesis and then pushed onto a list. This list is then put into the HYPOTHESES slot of the training problem unit, replacing the previous values in the slot. The list is also sorted by the discrimination score and the set of best scores is stored in the SCORES slot of the training problem unit.

The number of scores to be stored per wavelength is stored in the NUM.SCORES slot of the unit TRAINING.DATABASES as shown in Figure 2-4. If this slot has the value 5 and the number of wavelengths in the training problem is 2, then the best ten scores are stored in the SCORES slot of the training problem. This set of scores contains the best scores for the problem and may contain any number from zero to ten scores for each wavelength. Figure 3-12 shows an example of the value in the SCORES slot of a training problem. In this case, nine scores are for wavelength 0.64 µm and one score is for wavelength 0.82 µm. If a trace file has been named, the contents of the SCORES slot of each training problem are written to the trace file.

```
(((((GREATER-THAN  (20 0)  (40 0) T)  0. 64)) 0.5476)
(((((FIRST-MIN  (40 0)  T)  0.64))  0.5476)
(((((GREATER-THAN  (45 135) (75 135))  T) 0.82))  0.5119)
(((((FIRST-MIN  (20 0))  NIL)  0.64)) 0.4762)
(((((SECOND-MIN  (20 0)) T)  0.64))  0.4762)
(((((GREATER-THAN  (30 180)  (60 180)) T)  0.64))  0.3571)
(((((FIRST-MAX  (30 180)) T)  0.64))  0.3571)
(((((FIRST-MAX  (60 180))  NIL) 0.64)) 0.3571)
(((((SECOND-MAX  (60 180)) T)  0.64))  0.3571)
(((((SECOND-MIN  (40 0))  NIL)  0.64))  0.3333)
```

B921014G8

Figure 3-12

The Values in the SCORES Slot of a Training Problem

At this stage in the processing of the learning data, the HYPOTHESES slot of each training problem contains all the possible separate hypotheses for the problem. For each hypothesis the slot holds the discrimination score and also the positive and negative training set scores as given by the first and second terms respectively in equation (1). The SCORES slot of each training problem contains the set of the best single hypotheses for the training problem. The next step in the learning of class descriptions is to construct compound hypotheses. A compound hypothesis is composed of the combination of two or more individual hypotheses. The idea is that the interactions between various individual hypotheses may account for more variance (be more predictive) than any individual hypothesis. All the hypotheses contained in the HYPOTHESES slot of a training problem are considered as potential parts of compound hypotheses, and not just the set of the best single hypotheses stored in the SCORES slot.

Before compound hypotheses are constructed, the set of hypotheses for each training problem is reduced by removing from the HYPOTHESES slot any hypothesis that could not be combined with another hypothesis to form a compound hypothesis with a discrimination score better than the current best score. A compound hypothesis is (T) if all the separate hypotheses in the compound hypothesis are (T). Thus a compound hypothesis cannot have a discrimination score that is greater than the minimum of its components' positive training set scores. No hypothesis whose positive training set score is less than or equal to the current best score can form part of a compound hypothesis whose discrimination score is better than the current best score. For this reason, every hypothesis whose positive training set score is less than or equal to the current best score for the problem is removed from the HYPOTHESES slot.

For example, consider the hypotheses listed in Table 3-1. The best single hypothesis is hypothesis D which has a discrimination score of 0.85. If hypothesis B is combined with another hypothesis, the positive training set score for the resulting compound hypothesis cannot be greater than 0.84. Thus, whatever the negative score, the overall discrimination score for a compound hypothesis containing hypothesis B cannot be greater than 0.84. Consequently, hypothesis B cannot form part of a compound hypothesis that has a discrimination score of greater than 0.85. Hypothesis B is removed from the HYPOTHESES slot. Even though the discrimination scores of hypothesis A and hypothesis C are relatively low, the combination of these hypotheses could have a discrimination score greater than 0.85. For example, if all positive training set units that scored (T) for hypothesis A also scored (T) for hypothesis C, and no negative training set unit that scored (T) for hypothesis A also scored (T) for hypothesis C, the discrimination score of the compound hypotheses formed from hypothesis A and hypothesis C would be 0.9 - 0.0 = 0.9. The biggest benefit from combining these hypotheses together would be to reduce the negative score. Hypotheses A and C are retained in the HYPOTHESES slot even though their discrimination scores are relatively low because they could form part of a high scoring compound hypothesis.

Table 3-1

Examples of Hypothesis Scores

Hypothesis	Positive Training Set Score	Negative Training Set Score	Discrimination Score
A	0.9	0.3	0.6
B	0.84	0.01	0.83
C	0.95	0.6	0.35
D	0.89	0.04	0.85

The content of the HYPOTHESES slot is reduced further. If a hypothesis scores one for both the positive and negative training sets, it does not discriminate at all and it cannot contribute towards improved discrimination in a compound hypotheses so it is removed from the HYPOTHESES slot. If a hypothesis scores zero for the negative training set, combining it with other hypotheses cannot reduce the negative training set score and thus increase the overall discrimination score. Hypotheses that score zero for the negative training set are also removed from the HYPOTHESES slot. At the end of this step, the HYPOTHESES slot of each training problem contains only those hypotheses that could potentially be combined with other hypotheses to form a compound hypothesis with a discrimination score greater than the current best score for the problem. It should be noted that compound hypotheses can be constructed from any of the remaining hypotheses in the HYPOTHESES slot of the training problem.

The HYPOTHESES slot of a training problem may contain in excess of fifty hypotheses, even after it has been reduced. The number of possible compound hypotheses for some training problems is immense. The problem of dealing with such a large number of potential compound hypotheses was the subject of much effort. Several alternative strategies were experimented with before a successful solution to the problem was found. The first attempt was to implement a breadth-first search. Compound hypotheses that had been investigated were stored on an explored list. Each time a compound hypothesis was investigated, all possible combinations of the hypothesis and other hypotheses from the HYPOTHESES slot were constructed and stored on an unexplored list. Checks were made to prevent duplication of compound hypotheses on the unexplored list and to prevent the same hypothesis being investigated more than once. This involved sorting all the separate hypotheses within a compound hypothesis into a standard order so that comparisons could be made. This strategy was rejected because it was slow and the system frequently crashed because it ran out of memory. The second attempt was to implement a heuristic search with a depth bound. This strategy was also very limiting since, in many cases, the system ran out of memory even before the search of compound hypotheses consisting of only two separate hypotheses was completed.

It was decided to try a completely different approach. The new approach was a "Generate and Test" approach. This strategy involved testing some compound hypotheses that the previous version would have recognized as not being possible solutions. However, the new approach had a much reduced memory requirement compared with the previous version because explored and unexplored lists were not kept. All possible compound hypotheses were generated systematically in a way that made duplication impossible. This also greatly reduced the processing time because the separate hypotheses did not need to be sorted within a compound hypothesis and no checks for duplication needed to be made. This approach was successful. Tests have shown that learning class descriptions can take as long as twenty three hours, but no test has failed because of memory problems. The user has the option of interrupting the learning at any time and using the intermediate results. The implementation of this strategy will now be described in detail.

After the number of hypotheses in the HYPOTHESES slot of each training problem has been reduced, a button labelled "INTERRUPT" is displayed on the screen as shown in Figure 3-13. The user can left click on this box at any time during the processing of the compound hypotheses in order to interrupt the processing. If the processing is interrupted, the best results obtained up to the time of interruption are retained for later use.

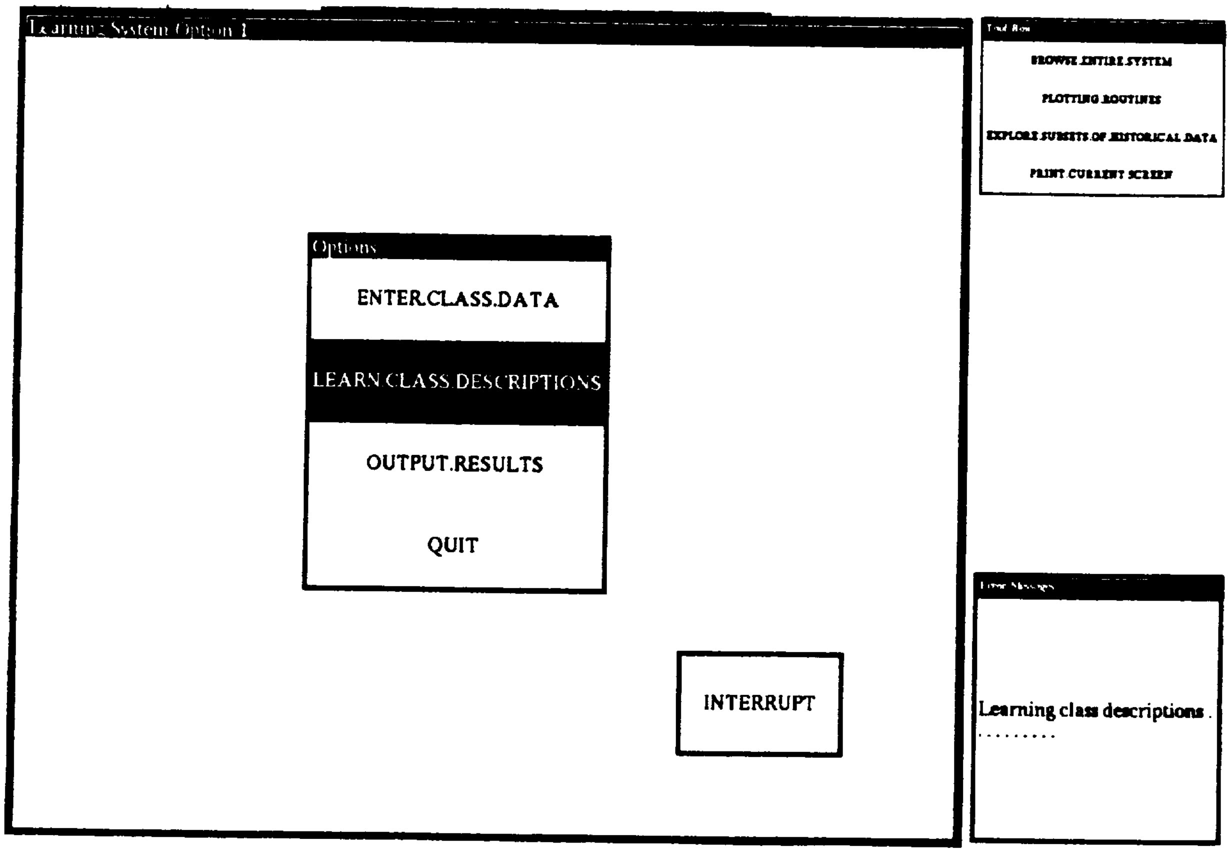


Figure 3-13

The INTERRUPT Button

For each training problem, the best single hypothesis score is stored in the slot PREVIOUS.BEST.SCORE of the training problem unit. Unless the best score for the problem is 1.0, the learning system next constructs and calculates the discrimination scores for all the hypotheses at level 2 (i.e. compound hypotheses each containing two separate hypotheses). Each compound hypothesis is generated and its discrimination score is immediately calculated. If the score is better than the previous best discrimination score for the problem, the hypothesis and score are placed in the BEST.SCORE slot of the problem replacing the previous values in the slot. If the score equals the best discrimination score for the problem and is better than the best discrimination score for single hypotheses for the problem, the compound hypothesis and score are added to the BEST.SCORE slot of the problem. If the best discrimination score for a single hypothesis is equaled by a compound hypothesis, the compound hypothesis and score are not added to the BEST.SCORE slot. Once the level 2 search has been completed, the contents of the BEST.SCORE slot of each training problem are written to the trace file if tracing has been selected. Then each training problem is considered again in turn. A check is made to determine whether a 10% improvement in the best discrimination score for the problem was achieved by searching at level 2 compared with considering only the single hypotheses. If this level of improvement has occurred, the level 3 hypotheses for the problem are constructed and processed. If a 10% improvement has not occurred, no further compound hypotheses are constructed or tested for that training problem. Constructing and testing hypotheses a level at a time continues until insufficient improvement is achieved or the user interrupts the processing by left clicking on the "INTERRUPT" button. The user can also control the level of compound hypothesis testing by setting a depth bound. Depth in this context refers to the number of hypotheses in the combination. This bound is stored in the slot SEARCH.DEPTH of the unit LEARNING.METHODS. If a trace file is in use, the time that the learning ended is written to the trace file and the file is closed. The message "Finished learning class descriptions" is displayed on the screen.

This step completes the learning of class descriptions. At the end of this step the SCORES slot of each training problem unit contains the best single hypotheses that discriminate the training class and the BEST.SCORE slot of each training problem unit contains the set of best compound hypotheses that discriminate the training class.

3.1.3 Output Results

The final step in learning system Option 1 is to output the results. When the user selects this step, the screen shown in Figure 3-14 is displayed. The solar zenith angle, wavelength(s) with associated view angles and the class definition for the first class to be reported are displayed in appropriately labelled boxes. In a text image labelled "Results" the names of the cover types in the positive and negative training sets are displayed, along with the best discrimination hypothesis scores for the class. If more than one training problem has been investigated, the user can view the next or the previous class by mousing on the appropriate options on the menu at the bottom of the screen.

As Task B of the current contract, an interface was implemented. This interface enabled the results of VEG to be written to a file. The interface worked for all the VEG subgoals except LEARN.CLASS.DESCRIPTIONS. This interface has been extended to include the results from the learning system.

When the user selects "QUIT" from the menu shown in Figure 3-14, another screen is opened. The user is asked whether or not the results should be written to a file. Left clicking on "NO" returns the user to the option 1 menu. Otherwise, the user is prompted to enter the name of the output file. A check is made to see if the file exists. If the file already exists, the user has the option of entering a new file name or overwriting the existing file. When an acceptable file name has been entered, the screen shown in Figure 3-15 is opened. This screen enables the user to select the parameters to be written to the file or select a template for a standard format. Each time the user left clicks on a parameter in the "Output Parameters" menu, the parameter is added to the slot FORMAT.LIST of the unit OUTPUT.LEARNING.RESULTS. When the user left clicks on "DONE", the data is written to the file and the option 1 menu is once again displayed. Instead of selecting the output parameters separately, the user can select a standard template. When the user left clicks on "STANDARD.TEMPLATE", the screen shown in Figure 3-16 is opened. Left clicking on the required template selects it. As soon as a template has been selected, the data is written to the file and the user is returned to the option 1 menu. Selecting "QUIT" from this menu returns the user to the main menu for the learning system.

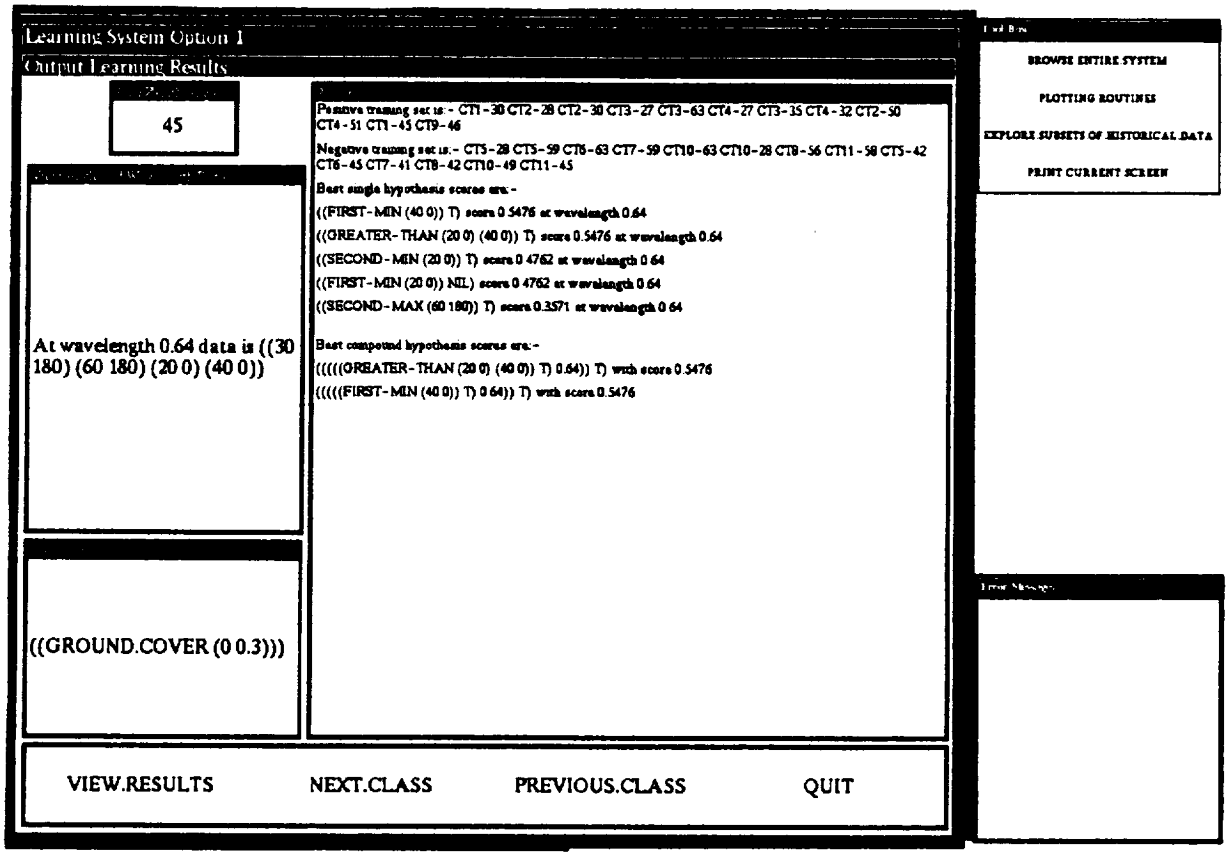


Figure 3-14

The Output Screen for Option 1 of the Learning System

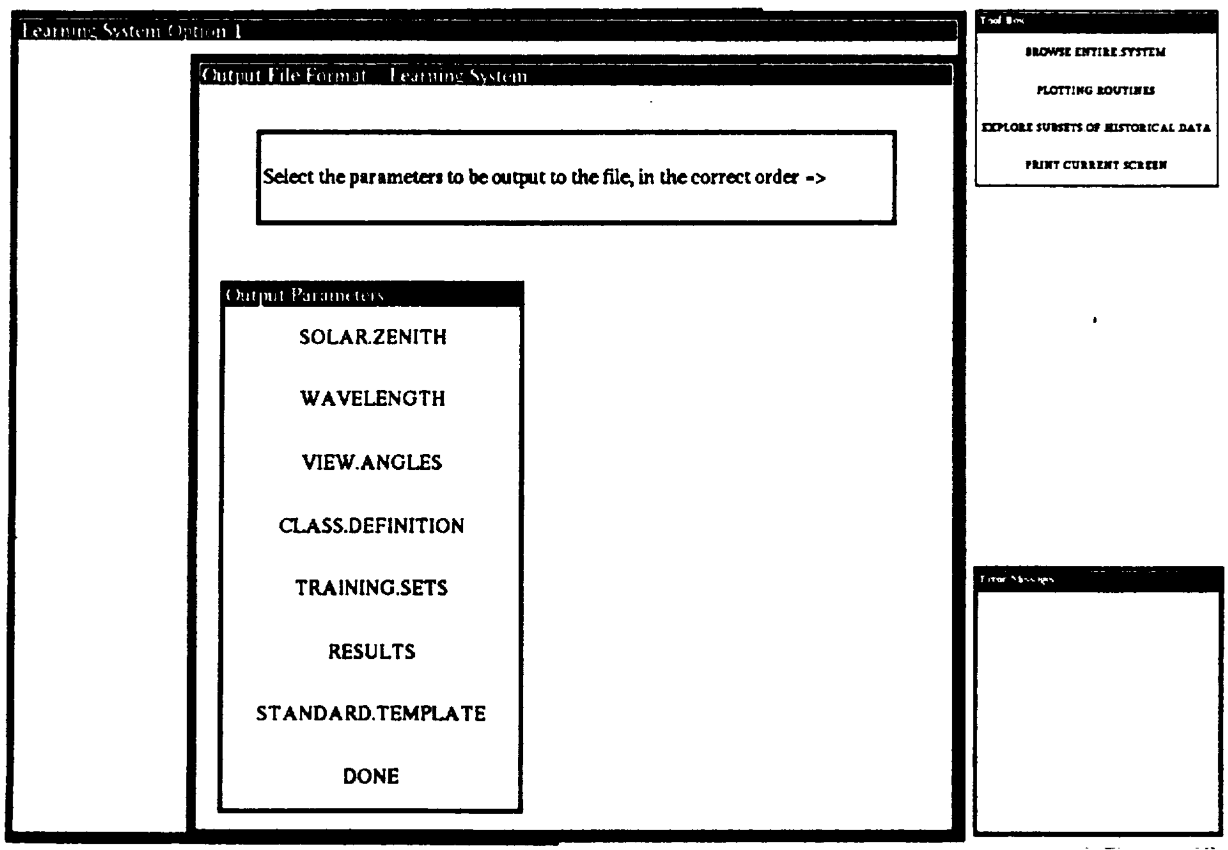


Figure 3-15

The Output File Format Menu for the Learning System

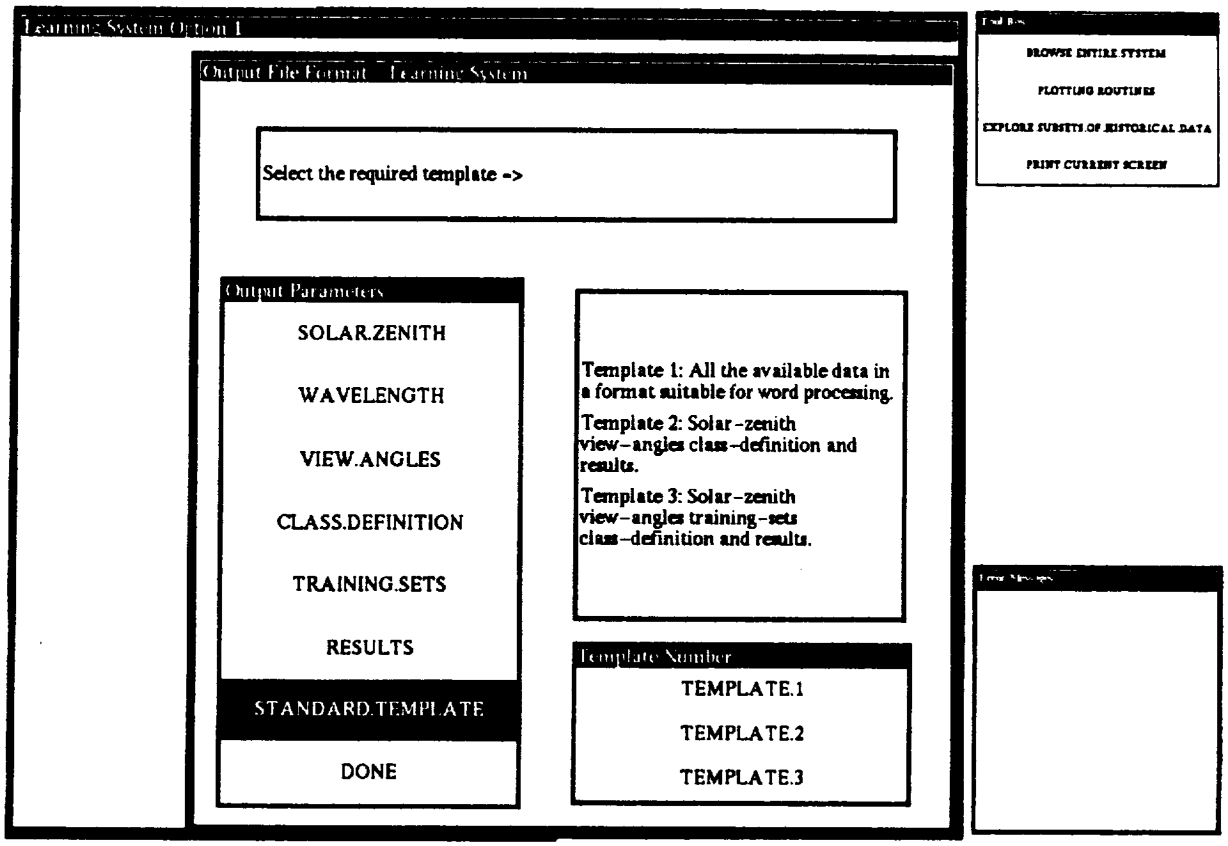


Figure 3-16

The Screen that Enables the User to Select a Standard Template

3.2 OPTION 2

Referring back to Figure 2-6, option 2 is used to have the system learn class descriptions for one or more classes and then use the classes to classify an unknown sample. In this option, the user either enters a new sample or selects a sample from the set of examples already stored in VEG. The user then enters one or more training classes. This class data is used together with the solar zenith, wavelength(s) and view angles from the sample to define the training problem(s). The system learns the class descriptions for the training problem(s) as in option 1. The sample is then classified to determine the degree to which it matches each of the defined classes, and to identify the class that it best fits. Figure 3-17 shows the menu for option 2. The steps involved in option 2 are described in detail in this subsection.

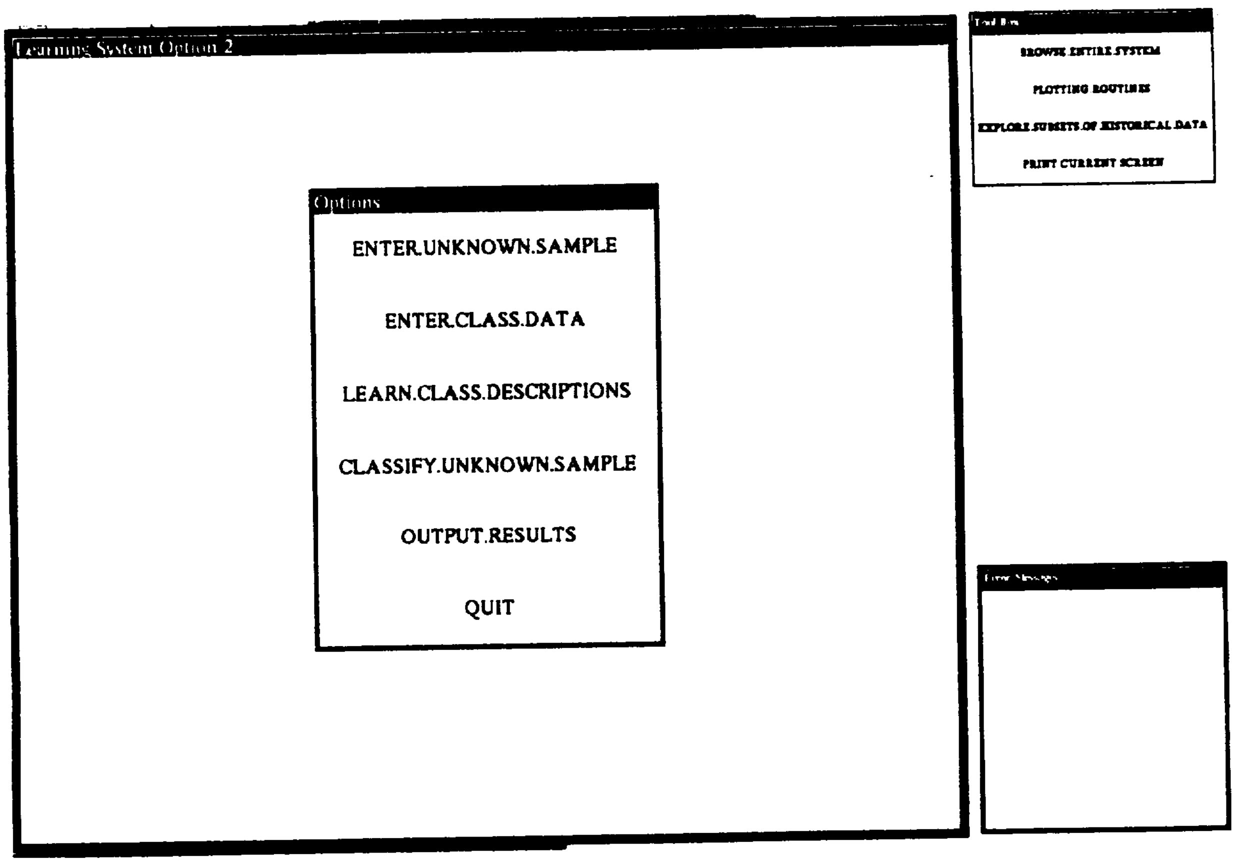


Figure 3-17

The Menu For Option 2

3.2.1 Enter Unknown Sample

When the user selects the option ENTER.UNKNOWN.SAMPLE from the option 2 menu, the Enter Data Interface that was originally developed for the Estimate Spectral Hemispherical Reflectance subgoal of VEG is opened. The first screen allows the user to choose between entering original data and selecting an example data set already stored in VEG. If the user selects the first of these options, the screen shown in Figure 3-18 is displayed. This screen enables the user to enter and store the data for the original sample that is to be investigated. If the user elects to select an example data set, a different screen is displayed. Descriptions of the available sets of sample data are displayed on this screen and the user can mouse on the sample name to select the required sample. The names of the units containing the newly entered or selected sample data are stored in slots of the unit ESTIMATE.HEMISPHERICAL.REFLECTANCE. It should be noted that this option uses objects from the underlying VEG layer.

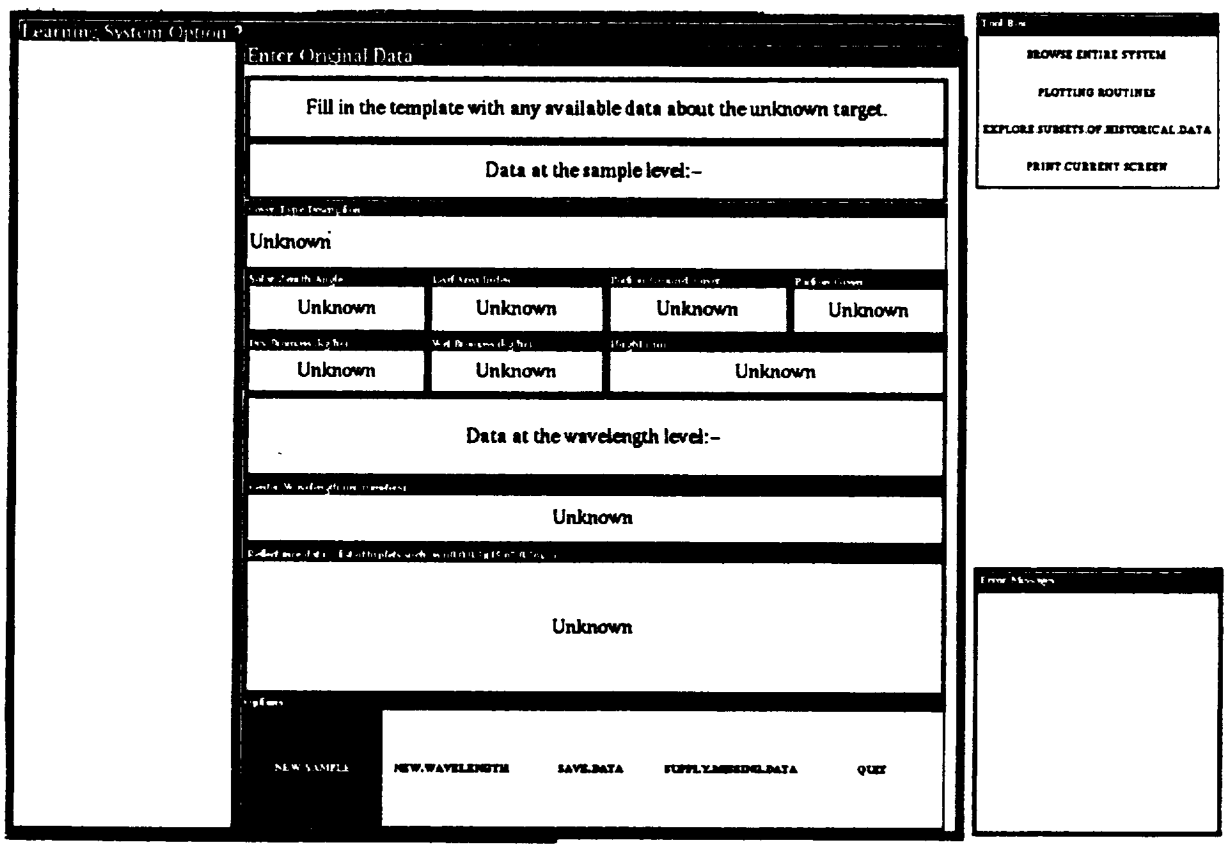


Figure 3-18

The Screen Through Which Original Sample Data is Entered

3.2.2 Enter Learning Data

When the user selects ENTER.LEARNING.DATA from the option 2 menu, the Lisp function MATCH-CLASSES-TO-SAMPLE is called. This function checks that the step ENTER.UNKNOWN.SAMPLE has been completed. It then deletes any previously entered training problems and initializes the values in the slots of the unit ENTER.LEARNING.DATA (Figure 2-2). The values of the directional view angles, solar zenith and wavelength from the sample are copied into the appropriate slots of the unit ENTER.LEARNING.DATA so that the training problem matches the unknown sample that is being classified. The mouse left functions of the boxes labelled "Additional Wavelength", "Solar Zenith" and "View Angle Data" in the Enter Learning Data interface are disabled to prevent the user changing any of the values that have been set to match the unknown sample. The Enter Learning Data interface is then opened. As shown in Figure 3-19, the solar zenith and view angle data are displayed. In order to complete the definition of the training problems, the user must then enter the class parameters and values and store the data as in option 1 (described in section 3.1.1).

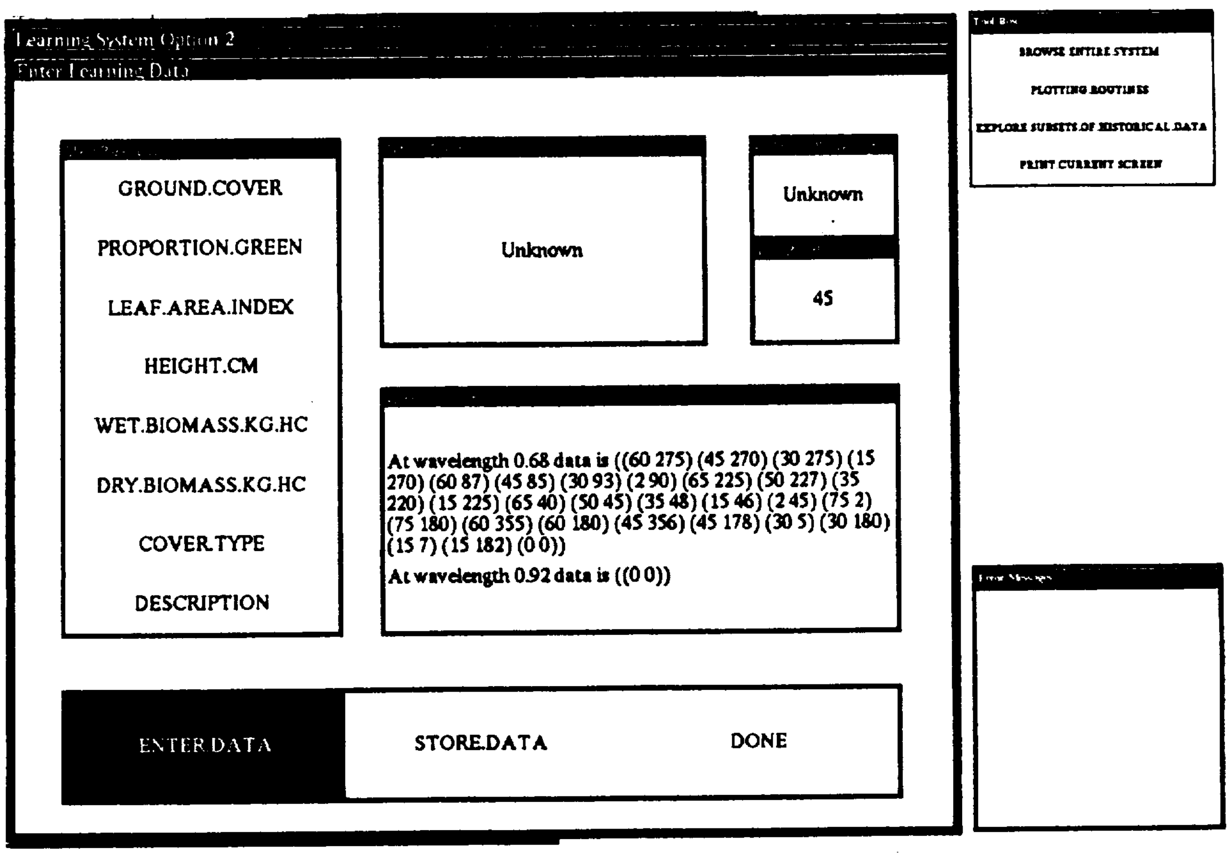


Figure 3-19

The Enter Learning Data Interface For Option 2

3.2.3 Learn Class Descriptions

The step LEARN.CLASS.DESCRIPTIONS in option 2 is identical to the same step in option 1. This step was described in detail in section 3.1.2.

3.2.4 Classify Unknown Sample

The class descriptions are used to classify the unknown sample by testing to determine which class has the most evidence that the sample belongs to that class. The relative score of evidence that is computed to determine whether a sample belongs to a particular class is calculated as :

$$\text{Score} = 1 - \frac{E_{Opp}}{E_{Sup}} \tag{4}$$

or

$$\text{Score} = -1 + \frac{E_{Sup}}{E_{Opp}} \tag{5}$$

where E_{Sup} is the sum of the discriminating scores of instances in the class solution that are true for the sample data, and E_{Opp} is the sum of discriminating scores that are false for the sample data. Equation (4) is used if the supporting evidence is greater than the opposing evidence. Otherwise, equation (5) is used.

When the user selects CLASSIFY.UNKNOWN.SAMPLE from the option 2 menu, the Lisp function GET-SCORES-FOR-SAMPLE is called. This function first makes sure that the prerequisite steps have been completed. A list of wavelengths and reflectance data in the unknown sample is constructed. The function CLASSIFY-SAMPLE is called with this list as the argument. Within the function CLASSIFY-SAMPLE, local variables BEST-SCORE and BEST-CLASS are initialized to the values -2 and NIL, respectively. CLASSIFY-SAMPLE iterates through the training problems. For each problem, the score given by equation (4) or (5) is calculated and stored in the slot SAMPLE.SCORE of the training problem. If the score is greater than the value stored in BEST-SCORE, the value of BEST-SCORE is reset to the score and the value of BEST-CLASS is set to the name of the training problem unit that yielded the score. The function CLASSIFY-SAMPLE returns the name of the training problem unit that gave the highest score according to equation (4) or (5). The slot BEST.CLASS of this unit is set to T to indicate that it is the class that has the most evidence that the unknown sample belongs to that class.

3.2.5 Output Results

The final step in option 2 is to output the results. This step is the same as was described for option 1 in section 3.1.3 except that additional results are output in option 2. For each training problem, the score for the unknown sample for that class is also displayed in the box labelled "Results." In addition, when the results for the best class for the sample are displayed, a message is added to the box labelled "Results" stating that the class being displayed is the best class for the sample.

Figures 3-20, 3-21 and 3-22 show the results of classifying the unknown sample stored in the units SAMPLE3, W5 and W6 in VEG. Three possible classes were investigated. These were (DESCRIPTION WHEAT), (DESCRIPTION GRASS) and (DESCRIPTION FOREST). The support scores for the unknown sample in these classes were -1.0000, -1.0000 and 0.0000, respectively. The system correctly reported that the class (DESCRIPTION FOREST) was the best class for the unknown sample. This test is discussed in detail in Section 5.2.

JJM SYSTEMS INC

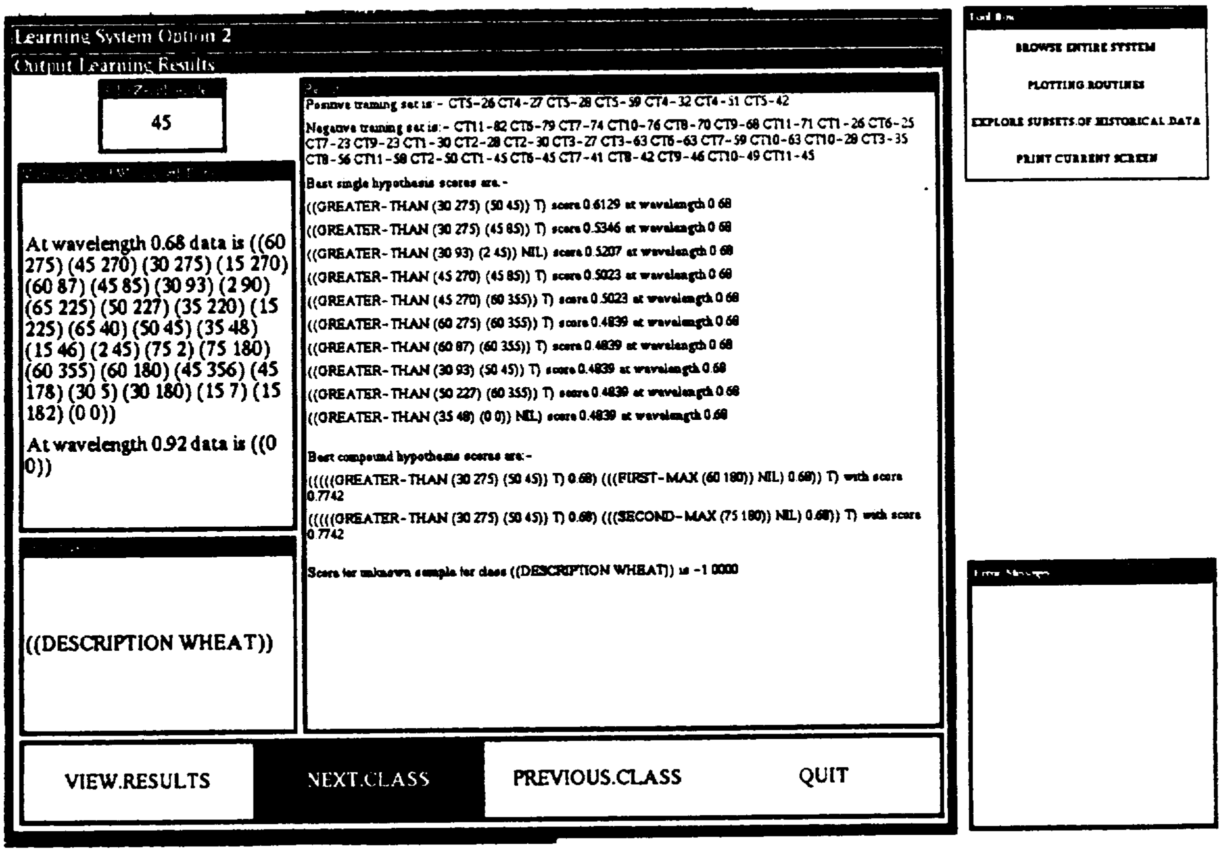


Figure 3-20

Results for SAMPLE3 and the Class (DESCRIPTION WHEAT)

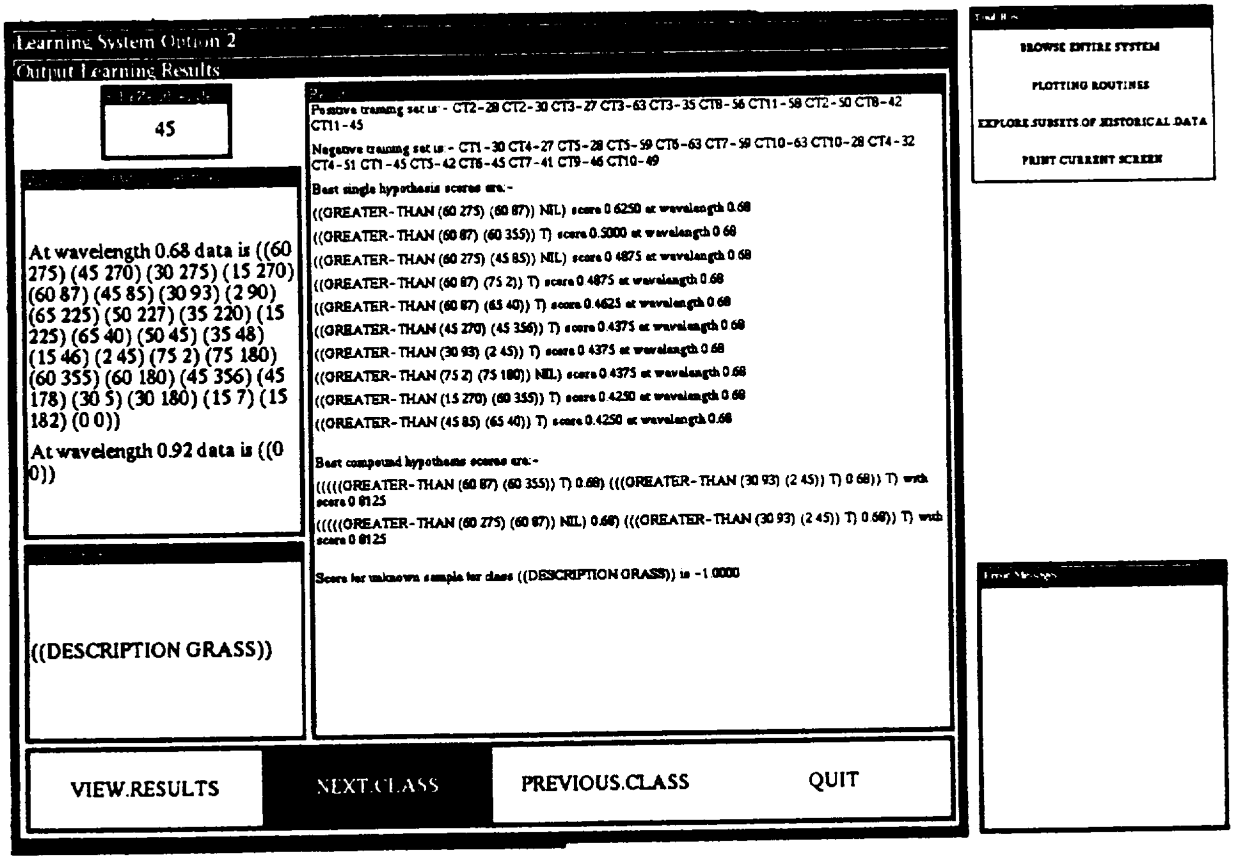


Figure 3-21

Results for SAMPLE3 and the Class (DESCRIPTION GRASS)

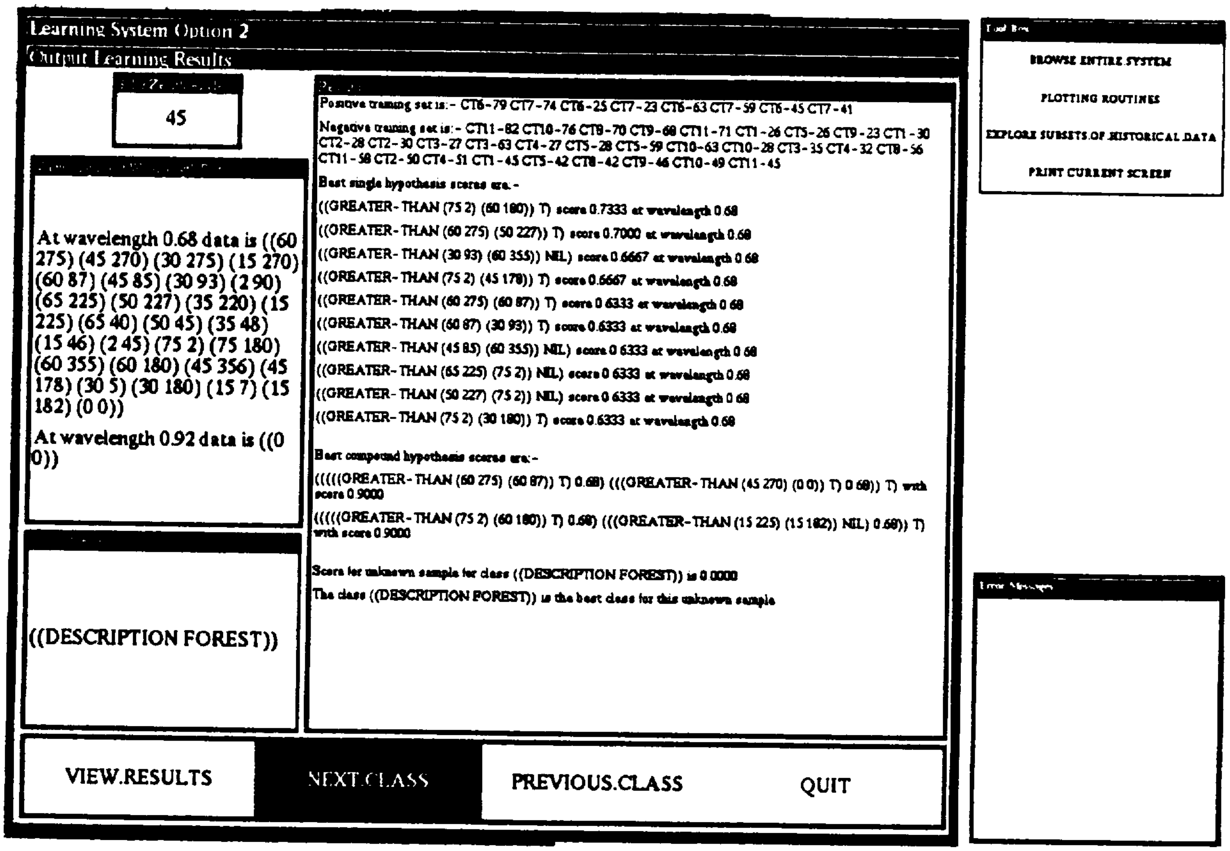


Figure 3-22

Results for SAMPLE3 and the Class (DESCRIPTION FOREST)

3.3 OPTION 3

Referring to Figure 2-6, the purpose of option 3 is to allow the user to test or validate the new classes defined using option 1 or 2. User defined classification systems can also be tested using this option. Class data is entered for one or more classes. The system then learns the class descriptions as in options 1 and 2. The cover types in the positive training set for each class are then classified according to the learned classes. The percentage of correctly classified cover types forms a measure of the classification accuracy achieved by the learning system. Figure 3-23 shows the menu for option 3.

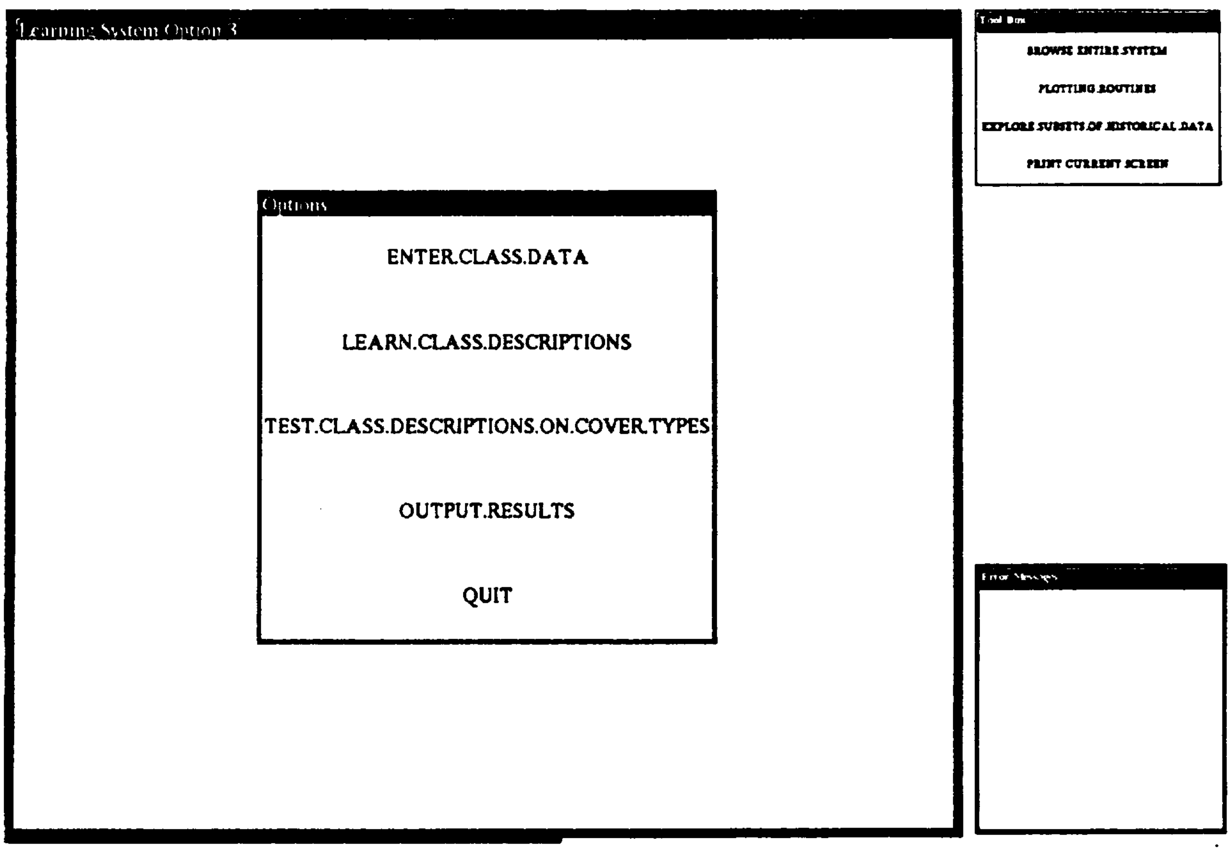


Figure 3-23

The Menu for Option 3

3.3.1 Enter Class Data and Learn Class Descriptions

These options can be selected and run exactly as in option 1. However, option 3 may be selected by the user immediately after running option 1 or option 2 as a means of estimating the accuracy of the results obtained by the previous option. In this case, the user can omit the steps ENTER.CLASS.DATA and LEARN.CLASS.DESCRIPTIONS in option 3, and the learning system will use the class data entered and the class descriptions learned in the previous option.

3.3.2 Test Class Descriptions on Cover Types

When the user selects this option, the Lisp function TEST-CLASS-DESCRIPTIONS-ON-COVER-TYPES is called. This function first makes sure that class data has been entered and class descriptions have been learned. It then processes each training problem in turn. Each cover type in the positive training set is classified using the function CLASSIFY-SAMPLE that was discussed in section 3.2.4. If the best class for the cover type is the class for which the cover type is a member of the positive training set, the cover type is considered to have been correctly classified. This process is repeated for all the cover types in the positive training sets of all the training problems.

The classification accuracy achieved by the learning system is calculated as the number of correctly classified cover types divided by the total number of cover types in the positive training sets of all the training problems. This result is stored in the slot PERFORMANCE.SCORE of the unit TRAINING.DATABASES.

3.3.3 Output Results

As in options 1 and 2, the final step in option 3 is to output the results. The results for option 3 are the same as those of option 1 except for the inclusion of additional data in the Results box. The cover types that were both correctly and incorrectly classified as belonging to the class are listed together with the performance score for all the classes. Figures 3-24, 3-25 and 3-26 show the results obtained when option 3 was applied to the data and classes defined in option 2 and briefly described in Section 3.2.5. The system's classification performance score was 0.8800. These results are discussed in detail in Section 5.2.

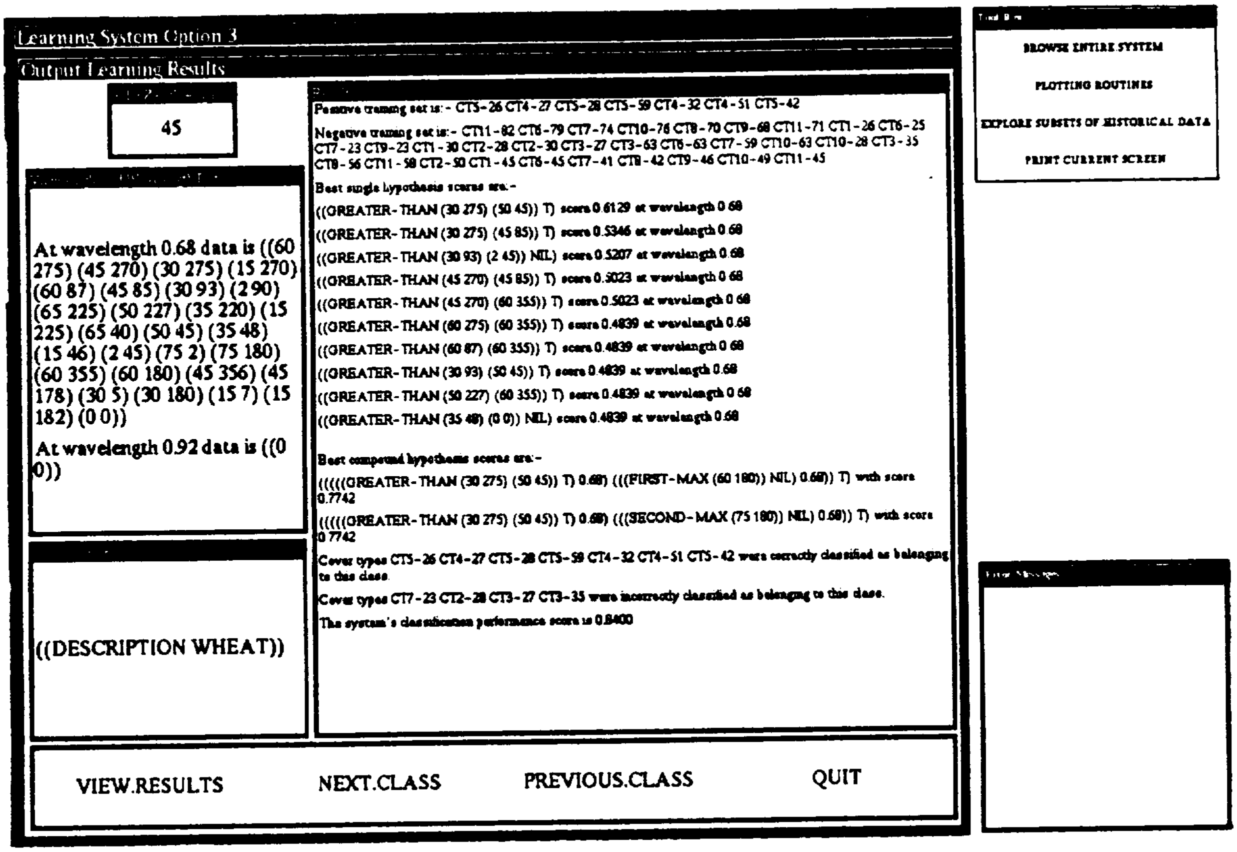


Figure 3-24

Results for SAMPLE3 and the Class (DESCRIPTION WHEAT) in Option 3

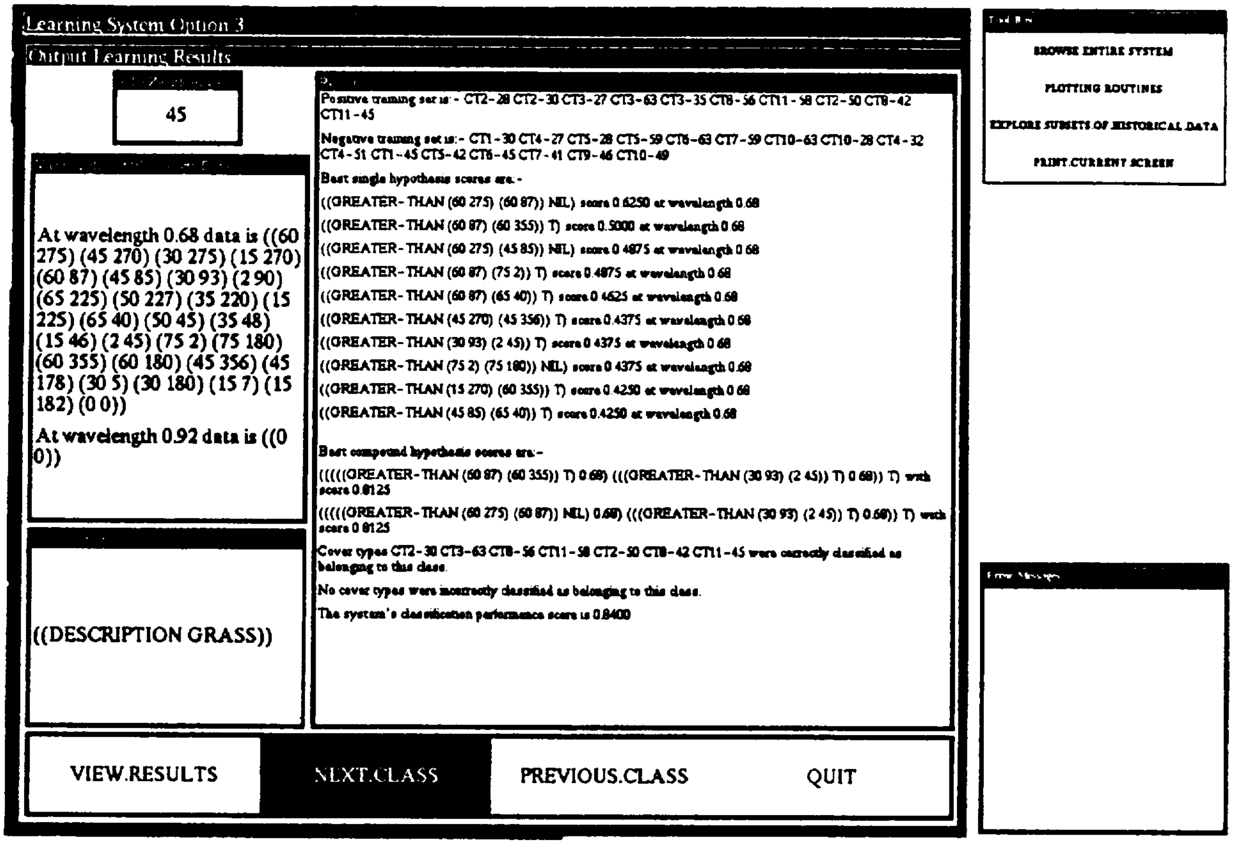


Figure 3-25

Results for SAMPLE3 and the Class (DESCRIPTION GRASS) in Option 3

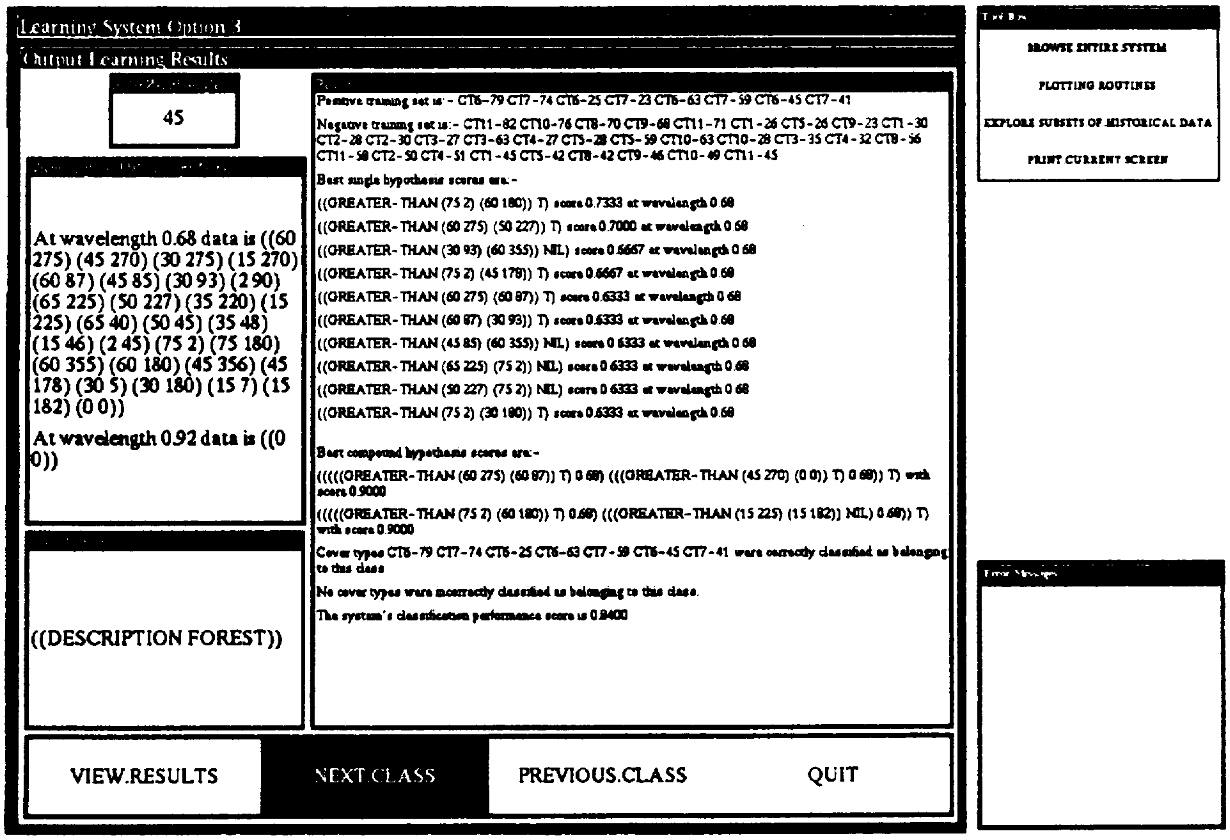


Figure 3-26

Results for SAMPLE3 and the Class (DESCRIPTION FOREST) in Option 3

SECTION 4.0

THE LEARNING SYSTEM IN THE VEG "AUTOMATIC MODE"

The preceding section described the operation of the learning system in the VEG "Research Mode." The learning system can also be run in the VEG "Automatic Mode." Figure 4-1 shows the current version of the "Automatic Mode" top level menu as it first appears on the screen. This screen differs from the screen shown in the reports for Tasks A and B of this contract. In the current version of VEG, the boxes that allow the user to make the selections needed for each subgoal are displayed on the screen only after the subgoal has been selected. If the user selects the subgoal LEARN.CLASS.DESCRIPTIONS, additional boxes are displayed, as shown in Figure 4-2. These boxes allow the user to enter the input and output file names, select the formats of the input and output files and also enter the class descriptions that are to be learned.

The interface to a file of unknown cover type data was described in detail in the report on Task A. This same interface is used by the learning system. When the user enters the name of the input file into the box labelled "Input File Name," the input file interface is opened allowing the user to specify the file format, as described in the previous report.

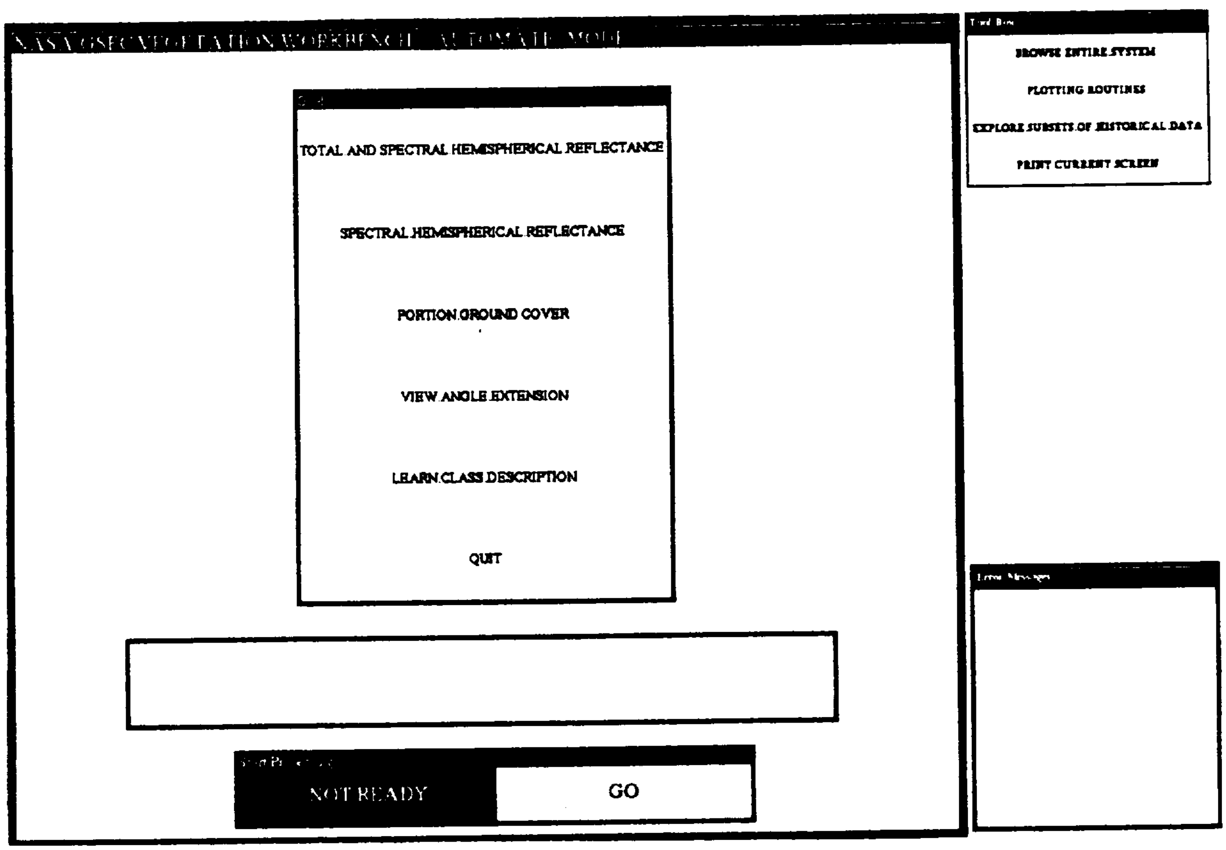


Figure 4-1

The VEG "Automatic Mode" Top Level Menu

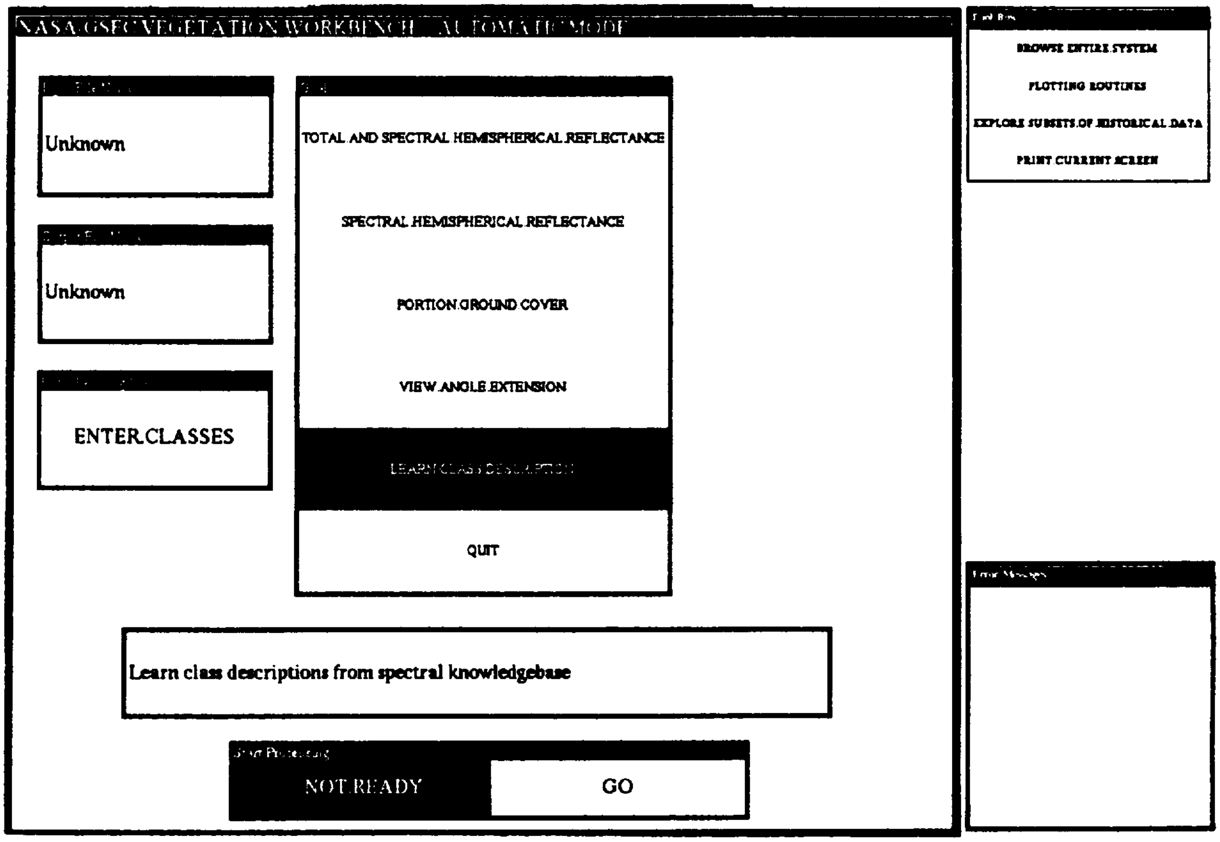


Figure 4-2

Selecting the Learning System from the VEG "Automatic Mode" Top Level Menu

When the user enters the name of the output file, a check is made to determine whether the file already exists. If the file exists, the user is asked whether the file should be overwritten. If the existing file is not to be overwritten, the user is prompted to enter a different file name. The interface shown in Figures 3-15 and 3-16 is then opened, allowing the user to select which data should be written to the output file.

When the user left clicks on the box labelled "Enter Learning Classes," the screen shown in Figure 4-3 is opened. This screen is the same as the screen shown in Figure 3-3 except that the boxes labelled "Additional Wavelength," "Solar Zenith" and "View Angle Data" have been closed. The screen shown in Figure 4-3 enables the user to enter the parameters and ranges of parameter values that define the classes that are to be learned. The selection of parameters and parameter values was described in Section 3.1.1.

Left clicking on "GO" runs the learning system. A check is made to make sure that input and output file names and a class description for at least one class have been entered. If any of these is missing, the user is prompted to enter the missing data.

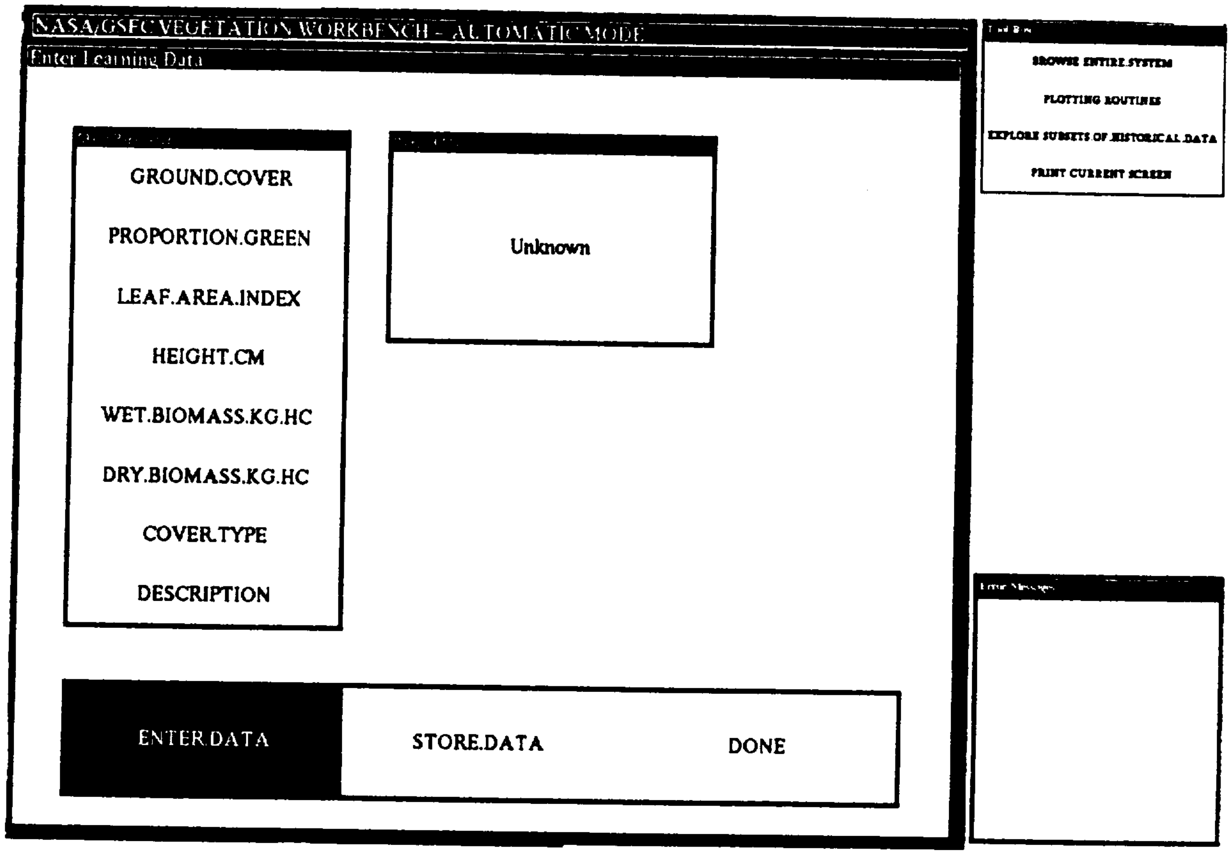


Figure 4-3

Selecting the Class Parameters in the VEG "Automatic Mode"

All the data from the input file is read and stored in units that are subclasses and members of the unit TARGET.DATA. The learning system then processes the data a sample at a time.

The wavelength(s), view angles and solar zenith of the sample under consideration are copied into the appropriate slots of each training problem unit. This data together with the class data entered at the beginning of the run defines one or more training problems that match the current sample. A similar method was applied in Option 2 of the learning system in the VEG "Research Mode." The system then learns the class descriptions of the training problems as was described in Section 3.1.2. The only difference is that the user is not offered the option of writing the intermediate results to a trace file. The sample is then classified according to the learned class descriptions to determine which class it best fits. The cover types in the positive training sets of all the training problems are then classified to measure the classification accuracy of the learning system using the same method as in Option 3 of the learning system in the VEG "Research Mode." The results for the current sample are written to the output file.

Since there may be more than one sample to process, the learning system is re-initialized after each sample is processed (except for the training problem class parameters). This process continues until all samples have been processed.

SECTION 5.0

TESTING AND RESULTS

All the options in the learning system have been tested. The tests included runs in both the VEG "Research Mode" and the "Automatic Mode." In some cases, errors in the coding were found. The code was debugged and the tests repeated until the system performed correctly. The trace files and results files from the test runs are presented in Appendix B.

5.1 TEST 1

This was a simple test in which Option 1 in the "Research Mode" was selected. One class having a ground cover of 0-30% was defined. Four random view angles at wavelength 0.68μm were selected. These were (15 182), (75 90), (0 0) and (35 45). A solar zenith of 45° was chosen. The best single hypothesis scores were:

((GREATER-THAN (15 182) (75 90)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (15 182)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (75 90)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 90) (0 0)) NIL) score 0.7500 at wavelength 0.68
((GREATER-THAN (75 90) (35 45)) NIL) score 0.5833 at wavelength 0.68

Compound hypotheses, each consisting of two separate hypotheses were constructed and tested. However, none produced a discrimination score that was better than the best discrimination score of the separate hypotheses. Thus no further compound hypotheses were investigated and the best compound hypothesis scores were reported as:

(((((FIRST-MAX (75 90)) NIL) 0.68)) T) with score 0.7738
(((((FIRST-MAX (15 182)) T) 0.68)) T) with score 0.7738
(((((GREATER-THAN (15 182) (75 90)) T) 0.68)) T) with score 0.7738

5.2 TEST 2

This test was designed to test Option 2 of the learning system in the VEG "Research Mode." It was briefly described in Sections 3.2.5 and 3.3.3. The sample of cover type data stored in the VEG units SAMPLE3, W5 and W6 was selected as the unknown cover type to be classified. This sample had twenty eight view angles at wavelength 0.68μm stored in W5 and one nadir view angle at wavelength 0.92μm, stored in W6. The reflectance data for SAMPLE3 at wavelength 0.68μm was ((0 0 0.043)(15 182 0.043) (15 7 0.043) (30 180 0.054) (30 5 0.043) (45 178 0.066) (45 356 0.044) (60 180 0.076) (60 355 0.054) (75 180 0.089) (75 2 0.067) (2 45 0.01) (15 46 0.03) (35 48 0.04) (50 45 0.05) (65 40 0.06) (15 225 0.02) (35 220 0.03) (50 227 0.04) (65 225 0.05) (2 90 0.01) (30 93 0.02) (45 85 0.03) (60 87 0.04) (15 270 0.02) (30 275 0.03) (45 270 0.05) (60 275 0.06)). The solar zenith was 45°. Three classes were selected. These were (DESCRIPTION FOREST), (DESCRIPTION GRASS) and (DESCRIPTION WHEAT). The value of the slot SEARCH.DEPTH in the unit LEARNING.METHODS was set to 2. This was to restrict the compound hypotheses investigated to those containing only two separate hypotheses. This test was originally carried out using a search depth of 5. However, after 23 hours of processing, the system was still only considering the hypotheses consisting of 3 separate hypotheses. Each class had approximately 250 separate hypotheses. The run was interrupted because it was concluded that it would not end in a reasonable length of time. The test was repeated with a search depth of 2. This test took 1 hour to complete.

The support scores for the unknown sample in the classes (DESCRIPTION WHEAT), (DESCRIPTION GRASS) and (DESCRIPTION FOREST) were -1.0000, -1.0000 and 0.0000, respectively. The first two of these scores reflected the fact that the unknown sample matched none of the best compound hypotheses for the classes (DESCRIPTION WHEAT) and (DESCRIPTION GRASS). The best compound hypotheses for the class (DESCRIPTION FOREST) were :

(((((GREATER-THAN (60 275)(60 87)) T) 0.68)
    (((GREATER-THAN (445 270)(0 0)) T) 0.68)) T) with score 0.9000 (6)

(((((GREATER-THAN (75 2)(60 180)) T) 0.68)
    (((GREATER-THAN (15 225)(15 182)) NIL) 0.68)) T) with score 0.9000 (7)

Hypothesis (6) was true for the SAMPLE3 data but hypothesis (7) was not true because the separate hypothesis (((GREATER-THAN (75 2)(60 180)) T) 0.68) was not true. By equation (5), the support score for the class (DESCRIPTION FOREST) was 0.0000. The learning system correctly reported this score. The system correctly reported that the class (DESCRIPTION FOREST) was the best class for the unknown sample. The results of this test are shown in Figures 3-20, 3-21 and 3-22. The trace files and output files for the test are included in Appendix B. This test showed that Option 2 of the learning system in the VEG "Research Mode" was working correctly.

5.3 TEST 3

This test was a continuation of test 2. The tester exited Option 2 and immediately selected Option 3 of the learning system in the VEG "Research Mode." The purpose of this test was to determine the classification accuracy of test 2. The option TEST.CLASS.DESCRIPTIONS.ON. COVER.TYPES was selected from the Option 3 menu. Then the results were output as shown in Figures 3-24, 3-25 and 3-26. Several cover types were correctly classified as belonging to each of the three classes. No cover types were incorrectly classified as belonging to either the class (DESCRIPTION FOREST) or (DESCRIPTION GRASS). However, cover types CT7-23, CT2-28, CT3-27 and CT3-35 were incorrectly classified as belonging to the class (DESCRIPTION FOREST). Cover types CT2-28, CT3-27 and CT3-35 should have been classified as belonging to the class (DESCRIPTION GRASS) and cover type CT7-23 should have been classified as belonging to the class (DESCRIPTION FOREST). The system's performance score of 0.8400 reflected the fact that 21 of the 25 cover types in the positive training sets were correctly classified. This was an acceptable score. This test showed that Option 3 of the learning system in the VEG "Research Mode" was operating correctly.

5.4 TEST 4

The purpose of test 4 was to test the classification accuracy of the learning system. Six separate runs were carried out using test data suggested by the NASA GSFC technical representative. Option 3 of the learning system in the VEG "Research Mode" was selected. The results are summarized in Table 5-1. Listings of the trace files and output files for these runs can be found in Appendix B. In run 6, poorly dispersed view angles were used. As expected, the performance score was lower in run 6 than in the other runs. The performance scores in the various runs were consistent with the expectations of the NASA GSFC technical representative based on his scientific knowledge of the characteristics of the cover types involved in the tests.

Table 5-1

Test 4 Data and Results

Run #	User Defined Classes	View Angles Available	Wavelength (μm)	Solar Zenith Angle	Performance Score
#1	Ground Cover (0-30%) and (31-100%)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	35°	1.00
#2	Ground Cover (0-30%) and (31-100%)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	70°	0.87
#3	Ground Cover (0-30%), (31-60%) and (61-100%)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	35°	0.9524
#4	Plant Height (>10 m) and (<10 m)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	45°	0.7368
#5	Ground Cover (0-30%) and (31-100%)	Well-Dispersed Data (0 0)(30 45)(60 45) (30 135)(60 135) (30 225)(60 225) (30 315)(60 315)	0.68	40°	0.9565
#6	Ground Cover (0-30%) and (31-100%)	Poorly Dispersed Data (0 0)(10 0)(15 0)	0.68	40°	0.6087

5.5 TEST 5

Test 5 was designed to test the LEARN.CLASS.DESCRIPTIONS part of the learning system in detail and allow the scientist to study the improvement in the discrimination score as increasingly more complex compound hypotheses were constructed and tested. Two learning problems were studied in this test suite.

In run 1 of test 5, Option 2 of the learning system in the VEG "Research Mode" was selected. SAMPLE5 was selected as the unknown sample to be classified. This sample had 16 view angles at wavelength 0.68μm and one view angle at wavelength 0.92μm. The solar zenith was 71°. The classes 0-30%, 31-70% and 71-100% ground cover were defined.

The system took 11 hours to learn the class descriptions for these classes. Studying the file test4-run1-trace, which is included in Appendix B, allows the scientist to study the improvement in the discrimination scores as increasingly more complex compound hypotheses were constructed and tested. This improvement is summarized in Table 5-2. For the class 0-30% ground cover (TRAINING.PROB.276), the best single hypothesis score was 0.8444. This score was increased to 0.9000 when compound hypotheses at level 2 (consisting of 2 separate hypotheses) were considered. Since the improvement between single hypotheses and level 2 hypotheses was less than 10%, no further compound hypotheses were investigated for this training problem. For the class 31-70% ground cover, the best single hypothesis score was 0.4917. This training problem showed the greatest improvement in discrimination score when progressively more complex hypotheses were investigated. The best compound hypothesis discrimination scores for this training problem were 0.8000, 0.9000 and 0.9333 at levels 2, 3 and 4 respectively. Level 5 hypotheses were not investigated since less than a 10% improvement was obtained between the level 3 and level 4 scores. The best single hypothesis discrimination score for the class 71-100% ground cover was 0.8182. This score was increased to 0.9091 when level 2 hypotheses were considered. No improvement was achieved when level 3 hypotheses were investigated for this training problem.

The unknown sample was classified. The classification scores for the classes 0-30%, 31-70% and 71-100% were -0.933, -1.000 and 0.5000 respectively. The sample was classified as belonging to the class 71-100% ground cover because this class had the highest classification score. Option 3 was also selected for this run. The system's classification performance score was calculated as 0.8929.

In run 2 of test 5, Option 2 of the learning system in the VEG "Research Mode" was again selected. SAMPLE3 was selected as the unknown sample to be classified. This sample had 28 view angles at wavelength 0.68μm and one view angle at wavelength 0.92μm. The solar zenith was 45°. The classes 0-30%, 31-70% and 71-100% ground cover were defined. The learning system took 6 hours to learn the class descriptions for these classes. The discrimination scores for the various different levels of compound hypotheses are given in Table 5-3. The best single hypothesis discrimination score for the class 0-30% ground cover was 1. This score represents perfect discrimination so no compound hypotheses were investigated for this class. The best single hypothesis score of 0.5000 for the class 31-70% ground cover was increased to 0.9000 and 0.9667 for the compound hypotheses at levels 2 and 3 respectively. The class 71-100% ground cover had the best discrimination score of 0.9412 at level 1 and 1 at level 2. Run 2 took a shorter time than run 1 of test 5 because less compound hypotheses were investigated. SAMPLE3 had classification scores of -1.0000, 1.0000 and 1.0000 for the classes 0-30%, 31-70% and 71-100% ground cover respectively. The learning system classified the sample as belonging to the class 71-100% ground cover. This class was selected rather than 31-70% ground cover because of the order in which the classes were considered. In fact, the learning system classified the sample as belonging to two different classes in this case. This result has been referred to the NASA GSFC technical representative for interpretation.

Table 5-2

Results of Test 5 Run 1

Class	Training Problem Unit	Best Discrimination Scores				
		Level 1	Level 2	Level 3	Level 4	Level 5
0-30% Ground Cover	TRAINING.PROB.276	0.8444	0.9000	-	-	-
31-70% Ground Cover	TRAINING.PROB.277	0.4917	0.8000	0.9000	0.9333	-
71-100% Ground Cover	TRAINING.PROB.278	0.8182	0.9091	0.9091	-	-

Table 5-3

Results of Test 5 Run 2

Class	Training Problem Unit	Best Discrimination Scores				
		Level 1	Level 2	Level 3	Level 4	Level 5
0-30% Ground Cover	TRAINING.PROB.3	1.0000	-	-	-	-
31-70% Ground Cover	TRAINING.PROB.4	0.5000	0.9000	0.9667	-	-
71-100% Ground Cover	TRAINING.PROB.5	0.9412	1	-	-	-

Test 5 showed that improved discrimination scores can be achieved by investigating progressively more complex compound hypotheses. However, this investigation can be extremely time consuming because of the large number of possibilities involved.

5.6 TEST 6

Test 6 was designed to test the operation of the learning system in the VEG "Automatic Mode." Two learning problems were studied in this test. In each case, a file was constructed to contain the same data as in units of the samples of target data already stored in VEG. The learning problems were solved by the learning system running in the "Automatic Mode," using the data from the specially constructed files. Then the same problems were solved by the learning system running in the "Research Mode," using both options 2 and 3. For each learning problem, the results of the two runs were then compared to make sure the learning system was giving the same results in the "Automatic Mode" as in the "Research Mode." The search depth was limited to 2 in these tests so that the runs would be completed in a reasonable length of time.

In the first learning problem, the unknown sample was SAMPLE7 and the classes were (DESCRIPTION GRASS) and (DESCRIPTION FOREST). In the second problem ,the unknown sample was SAMPLE1 and the classes were 0-30%, 31-70% and 71-100% ground cover. For both learning problems, the learning system returned the same training sets, hypothesis scores, classification scores and performance scores when it was running in the "Automatic Mode" as when it was running in the "Research Mode." The output files for test 6 are included in Appendix B. From the results of test 6, it was concluded that the learning system was operating correctly in the "Automatic Mode."

SECTION 6.0

EXTENDING THE BROWSER TO INCLUDE THE LEARNING SYSTEM

A toolbox is provided in VEG. The toolbox enables the user to browse the system, plot the reflectance data, explore the historical data and print out the screen. The browser enables the user to examine the hierarchy of class, subclass and member units in VEG and display the values of slots.

The learning system was implemented as a separate knowledge base to VEG so it was not accessible to the original version of the browser. The scope of the browser has been extended so that any knowledge base (including the learning system) can be browsed. An additional option "KB" has been added to the browser menu. When the user selects this option, a menu of all the currently loaded knowledge bases is displayed. The user can left click on a knowledge base to select it. The name of the knowledge base is displayed in a box labelled "Current KB" (Figure 6-1). A menu containing all the top level units in the knowledge base, except ActiveImage or ActiveValue units, is then displayed. The user can left click on the name of a unit to select which hierarchy of units to browse. The user can browse up or down the hierarchy of units and display slot values as in the previous version of the browser. When the user attempts to browse "up" from the top level or "down" from the bottom level in a tree of units, a menu containing all the top level units in the knowledge base, except ActiveImage or ActiveValue units, is displayed. The user can select which hierarchy of units to browse next. Figure 6-1 shows a screen dump of VEG with the browser in use.

All the options in the browser were tested using VEG, the learning system and several different system knowledge bases such as KEEINTERFACE. All the tests were successful, showing that the new version of the browser was working correctly. The code for the browser is presented in Appendix C.

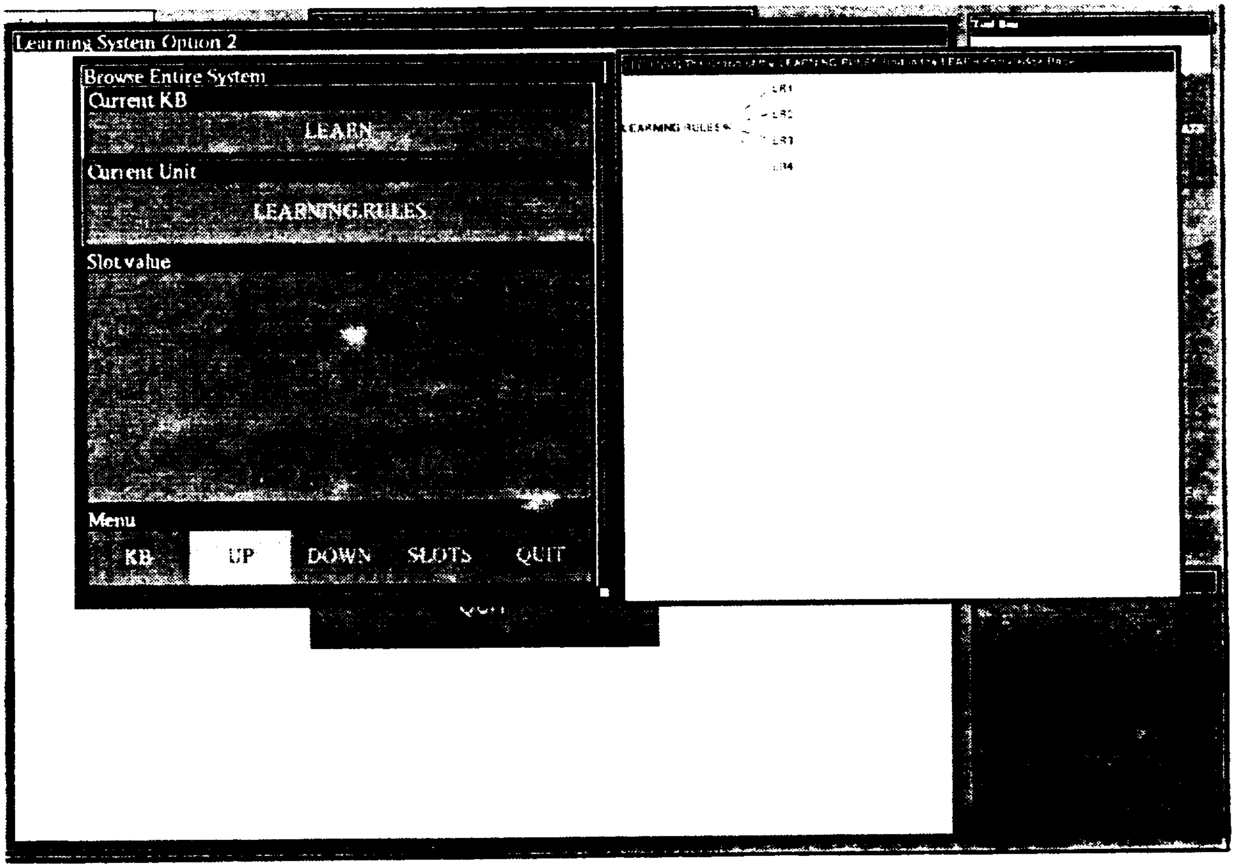


Figure 6-1

Screen Dump of the VEG with the Browser in Use

SECTION 7.0

CONCLUSIONS AND RECOMMENDATIONS

The learning system has been successfully implemented and integrated with the underlying VEG system. It provides additional tools for the scientist to use in exploring classification problems and for archiving results for a variety of research purposes. The browser within VEG has also been expanded to allow the scientist to browse the computational state of the VEG and learning systems.

The combined VEG and learning systems also manage complexity for the scientist, allowing one to focus on important scientific hypotheses rather than on trivial levels of data, computation and step management. It focuses the scientist on the "big" picture rather than burying one in a sea of detail.

The flexibility of the system allows the scientist a platform to conduct any number of explorations of a large body of reflectance data in a very short period of time. What took days in the past can now be accomplished in minutes. This means that the scientist can be much more productive and expansive in his/her thinking than would have been allowable without the time contraction that this system provides.

Finally, the system allows the results of the learning system to be written to a file. This provides a means of creating a ready history that is automatically maintained and is at the service of the scientist.

The current system suggests some interesting challenges for future development. These include:

a) the improved management of computational complexity in forming and testing hypotheses.
b) the development of a fully relational data base interface for the archiving of historical data.
c) additional tools and computational mechanisms to support the exploratory work of the scientist.
d) the generalization of this system to data from other sources such as radar.
e) migration of the system from Lisp to an ANSI standard C/C++ production quality version with explicit memory management rather than implicit memory management via garbage collection.
f) a distributed capability so that large historical data bases could be developed, manages and made available over a network for research work from distributed locations.

The key goal in all of these concepts is to empower the scientist so that he/she can be more productive and creative in his/her work.

REFERENCES

1. Kimes, D. S., Harrison, P. R. and Ratcliffe, P. A. 1991. A Knowledge-Based Expert System for Inferring Vegetation Characteristics. International Journal of Remote Sensing Vol 12, 10, pp. 1987-2020.

2. Kimes, D. S., Harrison, P. A. and Harrison, P. R. 1992. New Developments of a Knowledge Based System (VEG) for Inferring Vegetation Characteristics. International Geoscience and Remote Sensing Symposiom, Houston, Texas, May 1992.

APPENDIX A

LISTING OF METHODS FILES FOR THE LEARNING SYSTEM

```lisp
;;; learn-methods.lisp
;;;
;;; Main methods for the learning system
;;;
;;; Written by Ann and Patrick Harrison
;;; Created 10th July 1992
;;; Last modified 21st September 1992

(in-package 'kee)

;;;-----------------------------------------------------------------------
;;; Top level methods to generate the training sets and scores
;;;-----------------------------------------------------------------------

(defun start-learning-system ()
"Starts the learning system.  Initializes and displays the learning system top
level menu."
  (clear-prompt)
  (remove.all.values 'learning.methods 'option.number)
  (remove.all.values 'learning.system 'learning.options)
  (put.value 'learning.system 'message "")
  (put.value 'learning.system 'learning.menu 'view.options)
  (unitmsg 'viewport-learning.system.1 'open-panel!))

(defun learn-option-1 ()
"Displays the main menu for learning system options 1."
  (put.value 'learning.methods 'option.number 1)
  (remove.all.values 'option.1 'menu)
  (unitmsg 'viewport-option.1.1 'open-panel!))

(defun learn-option-2 ()
"Displays the main menu for learning system options 2."
  (put.value 'learning.methods 'option.number 2)
  (put.value 'estimate.hemispherical.reflectance 'done.enter.data.p nil)
  (remove.all.values 'option.2 'menu)
  (unitmsg 'viewport-option.2.2 'open-panel!))
```

```
(defun match-classes-to-sample()
"Checks to make sure an unknown sample has been selected or entered. If not
the function displays an error message and returns nil. Otherwise the function
calls the function set-up-training parameters to set up the necessary class
parameters to match the sample and then initializes and opens the interface for
entering the rest of the training class description(s)."
   (when (null (get.value 'estimate.hemispherical.reflectance
                     'done.enter.data.p))
     (my-documentation-print
      "Enter unknown sample before entering class data")
     (return-from match-classes-to-sample nil))
   (clear-prompt)
   (wipe-out-results 'dummy)
   (initialize-enter-learning-data)
   (set-up-training-parameters)
   (unitmsg 'windowpane-wavelength-of-enter.learning.data.1 'open!)
   (unitmsg 'windowpane-solar.zenith-of-enter.learning.data.2 'open!)
   (unitmsg 'windowpane-view.angle.data.message-of-enter.learning.data.4
          'open!)
   (deactivate-eld-windows)
   (unitmsg 'viewport-enter.learning.data.2 'open-panel!))

(defun set-up-training-parameters ()
"Sets up the directional view angles, solar zenith and wavelength in the
training set to match the unknown sample that is being classified."
  (let ((current-samples (get.values 'estimate.hemispherical.reflectance
                             'current.sample.wavelengths)))
    (get-view-angle-data current-samples)
    (put.value 'enter.learning.data 'solar.zenith
         (get.value (first current-samples) 'solar.zenith))))

(defun get-view-angle-data (current-samples)
"Stores a list of wavelengths and corresponding view angles in the
view.angle.data slot of enter.learning.data."
  (dolist (sample current-samples)
    (add.value 'enter.learning.data 'view.angle.data
         (list (get.value sample 'wavelength)
              (let ((result ()))
                  (dolist (point (get.value sample 'reflectance.data) result)
                    (push (butlast point) result)))))))

(defun learn-option-3 ()
"Displays the main menu for learning system options 3."
  (put.value 'learning.methods 'option.number 3)
  (remove.all.values 'training.databases 'performance.score)
  (remove.all.values 'option.3 'menu)
  (unitmsg 'viewport-option.3.1 'open-panel!))
```

```
(defun train ()
"If no training classes have been defined the function simply prints an error
message and returns nil.  Otherwise it generates the training sets, runs rules
to generate possible hypotheses and searches for the relationships that best
define the class."
  (let ((current-classes (get.values 'training.databases 'current.classes)))
    (when (null (get.value 'learning.methods 'done.enter.learning.data.p))
      (my-documentation-print
       "Enter classes before learning class descriptions")
      (return-from train nil))
    (my-documentation-print "Learning class descriptions . . . . . . . . . . ")
    (initialize-training-sets `dummy)
    (catch 'no-training-sets
      (dolist (problem current-classes)
            (generate-training-sets problem)
            (interp-extrap-training-data problem))
      (forward.chain 'learning.rules)
      (put.value 'learning.methods 'done.learn.class.descriptions.p t)
     (ask-about-trace-file))))

;;;----------------------------------------------------------------------
;;; Methods to enter data to define the classes to be classified
;;;----------------------------------------------------------------------

(defun open-enter-learning-data-interface ()
"Method to open the interface for entering data to define a class to be
characterized."
  (clear-prompt)
  (wipe-out-results 'dummy)
  (initialize-enter-learning-data)
  (unitmsg 'windowpane-wavelength-of-enter.learning.data.1 'open!)
  (unitmsg 'windowpane-solar.zenith-of-enter.learning.data.2 'open!)
  (unitmsg 'windowpane-view.angle.data.message-of-enter.learning.data.4
          'open!)
  (reactivate-eld-windows)
  (unitmsg 'viewport-enter.learning.data.2 'open-panel!))

(defun not-in-use (&rest lis)
"Function installed in the mouseleftfn! slot of the wavelength image in the
enter.learning.data interface.  Displays a message and prevents further input
of additional wavelengths when option 2 is in use."
  (declare (ignore lis))
  (my-documentation-print
"In option 2 use only the wavelengths, solar zenith and directional view angles
that match the unknown sample"))

(defun deactivate-eld-windows ()
"Puts the function not-in-use in the mouseleftfn! slot of the appropriate
images in the enter.learning.data interface."
  (put.value 'windowpane-wavelength-of-enter.learning.data.1
     'mouseleftfn! #'not-in-use)
  (put.value 'windowpane-solar.zenith-of-enter.learning.data.2
     'mouseleftfn! #'not-in-use))
```

```
(defun reactivate-eld-windows ()
"Puts the function simple-value-display-actuator back into the mouseleftfn!
 slot of the appropriate images in the enter.learning.data interface."
  (put.value 'windowpane-wavelength-of-enter.learning.data.1
     'mouseleftfn! #'simple-value-display-actuator)
  (put.value 'windowpane-solar.zenith-of-enter.learning.data.2
     'mouseleftfn! #'simple-value-display-actuator))

(defun initialize-enter-learning-data ()
"Initializes the slots in the unit enter.learning.data ready for the entry of
new class(es) to be learned."
   (remove.all.values 'training.databases 'performance.score)
   (remove.all.values 'enter.learning.data 'class.parameter)
   (remove.all.values 'enter.learning.data 'solar.zenith)
   (remove.all.values 'enter.learning.data 'value)
   (remove.all.values 'enter.learning.data 'wavelength)
   (remove.all.values 'enter.learning.data 'class)
   (remove.all.values 'enter.learning.data 'view.angle.data)
   (remove.all.values 'training.databases 'current.classes)
   (put.value 'enter.learning.data 'message "")
   (remove.all.values 'enter.learning.data 'view.angle.data.message)
   (put.value 'enter.learning.data 'menu 'enter.data))

(defun valid-directional-view-angles (data)
"Returns t if the directional view angles are valid and nil otherwise."
   (and (listp data)
        (dolist (point data t)
          (unless (and (listp point)
                       (= (length point) 2))
            (return-from valid-directional-view-angles nil))
          (let ((z (zenith point))
                (a (azimuth-360 point)))
            (unless (and (numberp z) (>= z 0) (<= z 90)
                         (numberp a) (>= a 0) (<= a 360))
              (return-from valid-directional-view-angles nil))))))

(defun store-directional-view-angles (dva)
"Stores the new wavelength and corresponding view angles in the view angle data
slot of the unit enter.learning.data."
   (add.value 'enter.learning.data 'view.angle.data
      (list (get.value 'enter.learning.data 'wavelength)
            dva))
   (remove.all.values 'enter.learning.data 'wavelength))
```

```
(defun store-learning-data()
"Creates the required unit and stores the data defining a class to be learned
in the knowledgebase."
  (let ((class (get.values 'enter.learning.data 'class))
        (view-angle-data
         (get.values 'enter.learning.data 'view.angle.data))
        (solar-zenith (get.value 'enter.learning.data 'solar.zenith)))
    (if (and class view-angle-data solar-zenith)
        (let ((new-unit (create.unit    (gentemp "TRAINING.PROB.")
                                   'learn 'training.databases)))
          (put.values new-unit 'class class)
          (put.values new-unit 'view.angle.data
             view-angle-data)
          (put.values new-unit 'view.angle.data.message
             (get.values 'enter.learning.data 'view.angle.data.message))
          (put.value new-unit 'solar.zenith solar-zenith)
          (dolist (va view-angle-data)
            (add.value new-unit 'wavelength (first va)))
          (add.value 'training.databases 'current.classes new-unit)
          (put.value 'learning.methods 'done.enter.learning.data.p t)
          (my-documentation-print "Data stored"))
        (my-documentation-print "Data incomplete  - not stored"))))

(defun valid-parameter (value min max)
"Returns t if the value is a list of numbers, each is within range and the
first number is less than the second.  Otherwise returns nil."
  (and (listp value)
       (= (length value) 2)
       (let ((f (first value))
             (s (second value)))
         (and (numberp f)(numberp s)
              (< f s)
              (>= f min)(<= f max)
              (>= s min)(<= s max)))))

(defun valid-parameter-values(new-val new-parameter)
"Returns t if the parameter value(s) entered are valid for a possible
training class in the learning system and nil otherwise."
  (case new-parameter
    (cover.type (member new-val (get.values 'enter.learning.data
                                      'possible.cover.types)))
    (description (search (string new-val )
                         (get.value 'enter.learning.data
                           'possible.descriptions)))
    (ground.cover (valid-parameter new-val 0 1))
    (proportion.green (valid-parameter new-val 0 1))
    (leaf.area.index (valid-parameter new-val 0 1))
    (height.cm (valid-parameter new-val 0 3000))
    (wet.biomass.kg.hc (valid-parameter new-val 0 1))
    (dry.biomass.kg.hc (valid-parameter new-val 0 1))))
```

```
(defun store-parameter-values (new-val new-parameter)
"Stores the new parameter and value in the class slot of the unit
enter.learning.data."
  (add.value 'enter.learning.data 'class (list new-parameter new-val)))

;;;------------------------------------------------------------------------
;;; Methods to generate the training sets
;;;------------------------------------------------------------------------

(defun data-available (object class)
"Returns t if the object has data for each parameter included in the class
definition."
  (dolist (par class t)
    (unless (get.value object (first par))
      (return-from data-available nil))))

(defun wavelengths-available (covertype class-wavelengths)
"Returns t if covertype data is available for each of the class wavelengths and
 nil otherwise."
  (let ((cover-wavelengths (get.values covertype 'wavelengths)))
    (dolist (wave class-wavelengths t)
      (dolist (band cover-wavelengths
                (return-from wavelengths-available nil))
          (when (and (>= wave (first band))
                     (<= wave (second band)))
            (return t))))))
```

```
(defun generate-training-sets (problem)
"Method to generate the training sets for a class definition. This function
extracts the parameters to be matched and then performs a search through the
historical data to find the cover types that best match the required
parameters.  The search ends when sufficient cover types have been found or
it has been established that insufficient cover types are available."
  (let* ((class (get.values problem 'class))
          (class-solar-zenith (get.value problem 'solar.zenith))
        (solar-zenith-interval (* 0.10 class-solar-zenith))
          (class-wavelengths (get.values problem 'wavelength))
          (min-set-size (get.value 'training.databases 'minimum.set.size))
          (max-set-size (get.value 'training.databases 'maximum.set.size)))
    (do* ((n 1 (1+ n))
           (solar-zenith-int solar-zenith-interval
                              (* n solar-zenith-interval))
           (num-pos-sets 0 (length (get.values problem 'pos.training.set)))
           (num-neg-sets 0 (length (get.values problem 'neg.training.set))))
          ((or (> num-pos-sets max-set-size)
               (> num-neg-sets max-set-size)
               (and (> num-pos-sets min-set-size)
                  (> num-neg-sets min-set-size))
               (= n 10)) ;Sets full or insufficient data available
           (cond ((zerop num-pos-sets)
                  (my-documentation-print
            "Discrimination not possible - no positive training sets")
                  (throw 'no-training-sets))
                 ((zerop num-neg-sets)
                  (my-documentation-print
                  "Discrimination not possible - no negative training sets")
                  (throw 'no-training-sets))))
  ;Consider every possible wavelength and sun angle for every cover-type
    (dolist (covertype (unit.children 'historical.cover.types 'subclass))
       (when (and (data-available covertype class)
                           ;Data available for classification parameter
                  (wavelengths-available covertype class-wavelengths))
          (dolist (sun-angle (unit.children covertype 'subclass))
           (when
                (solar-zenith-p sun-angle class-solar-zenith ;Sun angle matches
                                solar-zenith-int)
             (add-to-training-set problem class sun-angle))))))))
```

```
(defun pos-training-example (object class)
"Return t if the example is in the positive training class and nil otherwise."
 (dolist (par class t)
  (let* ((parameter (first par))
         (class-value (get.value object parameter))
         (val (second par)))
   (cond ((listp val)                    ;Continuous parameter
          (unless (and (<= class-value (second val))
                       (>= class-value (first val)))
            (return-from pos-training-example nil)))
         ((eq parameter 'cover.type)
          (unless (eq class-value val)
            (return-from pos-training-example nil)))
         (t                              ;Parameter is description
          (unless (search (string val) class-value)
            (return-from pos-training-example nil)))))))

(defun add-to-training-set (problem class cover-type)
"Adds the cover type name to the positive.training.set or negative.training.set
 slot of the training problem unit, as approproate."
  (if (pos-training-example cover-type class)
     (add.value problem 'pos.training.set cover-type)
     (add.value problem 'neg.training.set cover-type)))

(defun delete-sub-units (parent)
"If parent is a training problem unit, this function deletes all the subunits
of the training set unit."
  (when (member parent (unit.children 'training.databases 'subclass))
    (dolist (ts (unit.children parent 'subclass))
      (dolist (ts1 (unit.children ts 'subclass))
         (dolist (un (unit.children ts1 'member))
           (delete.unit un))
         (delete.unit ts1))
      (delete.unit ts))))

(defun initialize-training-set(parent)
"Initializes a training set by removing all the units in the positive and
negative training sets and by removing the values in the hypotheses, scores and
training set slots of the training problem unit."
  (delete-sub-units parent)
  (remove.all.values parent 'hypotheses)
  (remove.all.values parent 'scores)
  (remove.all.values parent 'pos.training.set)
  (remove.all.values parent 'neg.training.set)
  (remove.all.values parent 'best.score)
  (remove.all.values parent 'best.class)
  (remove.all.values parent 'sample.score))

(defun initialize-training-sets (self)
"Initializes all the training sets."
  (declare (ignore self))
  (dolist (prob (unit.children 'training.databases 'subclass))
    (initialize-training-set prob)))
```

```
(defun wipe-out-results (self)
"Removes all the current training problems."
  (declare (ignore self))
  (put.value 'learning.methods 'done.enter.learning.data.p nil)
  (put.value 'learning.methods 'done.classify.sample.p nil)
  (put.value 'learning.methods 'done.classify.cover.types.p nil)
  (put.value 'learning.methods 'done.learn.class.descriptions.p nil)
  (remove.all.values 'training.databases 'current.classes)
  (remove.all.values 'training.databases 'performance.score)
  (dolist (prob (unit.children 'training.databases 'subclass))
    (delete-sub-units prob)
    (delete.unit prob)))

;;;-----------------------------------------------------------------------
;;; Functions used to extrapolate the cover type data to set up the training
;;; set unit
;;;-----------------------------------------------------------------------

(defun create-training-set-parent-units(problem)
"Creates the positive and negative training set parent units as subclasses of
the training problem unit."
  (let ((unit-name (string (unit.name problem))))
        (values (create.unit
                       (intern (string-append "POS." unit-name))
                       'learn problem)
                  (create.unit
                       (intern (string-append "NEG." unit-name))
                       'learn problem))))

(defun interp-extrap-training-data(problem)
"Sets up the units be the parents of the positive and negative training data
sets and calls the function to create the child units to hold the training
data."
  (let ((view-angle-data (get.values problem 'view.angle.data)))
    (multiple-value-bind (pos-parent neg-parent)
         (create-training-set-parent-units problem)
      (interp-extrap-training-data-aux view-angle-data pos-parent
                                       (get.values problem 'pos.training.set)
                                       "POS")
      (interp-extrap-training-data-aux view-angle-data neg-parent
                                       (get.values problem 'neg.training.set)
                                       "NEG"))))
```

```
(defun interp-extrap-training-data-aux (view-angle-data parent
                                         training-set str)
"Interpolates and extrapolates the covertype data and stores it in the
appropriate units in the positive and negative training data sets."
  (dolist (sun-angle training-set)
    (let ((new-unit (create.unit (gentemp str) 'learn parent)))
      (put.value new-unit 'cover.type sun-angle)
      (dolist (vad view-angle-data)
          (let ((wave-unit (create.unit (gentemp str) 'learn nil new-unit))
               (wave (first vad)))
            (put.value wave-unit 'wavelength wave)
            (put.value wave-unit 'reflectance.data
               (match-unaltered-target-data (second vad)
                                         (get-wave-unit wave sun-angle))))))))

(defun get-wave-unit (wave sun-angle)
"Finds the appropriate member unit of the unit sun-angle that has the
required wavelength to match the training class."
  (dolist (wav (unit.children sun-angle 'member))
    (when (and (>= wave (get.value wav 'wavelength.min))
               (<= wave (get.value wav 'wavelength.max)))
      (return-from get-wave-unit wav))))

;;;-----------------------------------------------------------------------
;;;Functions used to test possible discriminating relationships
;;;-----------------------------------------------------------------------

(defun first-max (data flag direction &rest lis)
"Tests whether the maximum of all the reflectance values in the data is at
direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
    (if (eq (= (third point)
               (apply #'max (mapcar #'third data)))
            flag)
        1
        0)))

(defun first-min (data flag direction &rest lis)
"Tests whether the minimum of all the reflectance values in the data is at
direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
    (if (eq (= (third point)
               (apply #'min (mapcar #'third data)))
            flag)
        1
        0)))
```

```
(defun second-max (data flag direction &rest lis)
"Tests whether the second maximum of all the reflectance values in the data is
at direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
 (declare (ignore lis))
 (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
  (if (eq (= (third point)
             (third (second (sort data #'> :key #'third))))
          flag)
      1
      0)))

(defun second-min (data flag direction &rest lis)
"Tests whether the second minimum of all the reflectance values in the data is
at direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
 (declare (ignore lis))
 (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
  (if (eq (= (third point)
             (third (second (sort data #'< :key #'third))))
          flag)
      1
      0)))

(defun greater-than (data flag direction1 direction2)
"Tests whether the reflectance value at direction 1 is greater than the
reflectance value at direction 2. If this test gives a result equal to the
flag, the function returns 1.  Otherwise it returns 0."
 (let ((point1 (find direction1 data :test #'zeniths-and-azimuths-match))
       (point2 (find direction2 data :test #'zeniths-and-azimuths-match)))
 (if (eq (> (third point1)(third point2))
         flag)
    1
    0)))
```

```
(defun backscatter>forwardscatter (data flag &rest lis)
"Tests whether the average backscatter reflectance value in the data is greater
than the average forwardscatter reflectance value.  If this test gives a
result equal to the flag, the function returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((sum-back 0)
        (num-back 0)
        (sum-forward 0)
        (num-forward 0))
    (dolist (ang data)
      (let ((az (azimuth-360 ang)))
        (cond ((>= az 180)               ;backscatter angle
               (incf num-back)
               (incf sum-back (third ang)))
              ((< az 180)                ;forwardscatter angle
               (incf num-forward)
               (incf sum-forward (third ang)))
              (t nil))))                 ;nadir - ignore
    (if (eq (> (/ sum-back num-back)          ;average back-scatter
               (/ sum-forward num-forward))   ;average back-scatter
            flag)
        1
        0)))

(defun score (problem func wave &optional arg1 arg2)
"Calculates the best discrimination score for a relationship.  For example it
might test ((GREATER-THAN (60 180)(20 0)) T) and ((GREATER-THAN (60 180)(20 0))
 NIL).  The function returns the set of data containing the higher of these
scores (eg (((GREATER-THAN (60 180)(20 0)) T) 0.6543)) or
(((GREATER-THAN (60 180)(20 0)) ?) 0.0001) if both tests gave a discrimination
score of 0."
  (let ((training-data-sets (unit.children problem 'subclass)))
    (multiple-value-bind (num-pos pos-total)
        (score-aux (second training-data-sets) func wave arg1 arg2)
      (multiple-value-bind (num-neg neg-total)
          (score-aux (first training-data-sets) func wave arg1 arg2)
        (let ((pos-score (/ pos-total num-pos))
              (neg-score (/ neg-total num-neg)))
          (values (make-result func arg1 arg2 t
                               (- pos-score neg-score)
                       pos-score
                       neg-score
                       wave)
                  (make-result func arg1 arg2 nil
                               (- neg-score pos-score)
                       (- 1 pos-score)
                       (- 1 neg-score)
                       wave)))))))
```

```
(defun score-aux (training-set func wave arg1 arg2)
"Returns the number of elements and the total score for a training set."
 (let ((total 0)
       (num 0))
  (dolist (parent (unit.children training-set `subclass))
   (dolist (unit (unit.children parent 'member))
       (when (= wave (get.value unit 'wavelength))
        (incf num)
        (incf total
             (funcall func (get.value unit 'reflectance.data)
                      t arg1 arg2)))))
  (values num total)))

(defun make-result (func arg1 arg2 flag score pos-score neg-score wave)
"Sets up the format of a set of result data to allow for different numbers of
arguments without including unnecessary nils."
 (cond ((null arg1)
        `(((((,func)            ;No arguments
            ,flag)
            ,wave))
            ,score ,pos-score ,neg-score))
       ((null arg2)
        `(((((,func ,arg1)       ;1 argument
            ,flag)
            ,wave))
            ,score ,pos-score ,neg-score))
       (t
        `(((((,func ,arg1 ,arg2)
            ,flag)
            ,wave))
            ,score ,pos-score ,neg-score))))

;;;--------------------------------------------------------------------------
;;;Functions required by learning system rules
;;;--------------------------------------------------------------------------

(defun try-direction-relationships (unit view-angle-data)
"Sets up all possible distinct greater-than relationships for a set of data
and adds them to the hypotheses slot of the problem unit."
 (let ((hypotheses nil)
       (wave (first view-angle-data))
       (view-angles (second view-angle-data)))
  (do ((first-ang (first view-angles)(first remaining-angles))
        (remaining-angles (rest view-angles)(rest remaining-angles)))
       ((null remaining-angles))
    (dolist (second-ang remaining-angles)
       (push (list 'greater-than wave first-ang second-ang) hypotheses)))
  (add.values unit 'hypotheses hypotheses)))
```

```
(defun try-max-min-relationships (unit view-angle-data)
"Sets up all possible distinct first-max and first-min relationships for a set
of data and adds them to the hypotheses slot of the problem unit."
 (let ((hypotheses nil)
        (wave (first view-angle-data))
        (view-angles (second view-angle-data)))
   (dolist (ang view-angles)
    (push (list 'first-max wave ang) hypotheses)
    (push (list 'first-min wave ang) hypotheses))
   (add.values unit 'hypotheses hypotheses)))

(defun try-second-max-min-relationships (unit view-angle-data)
"Sets up all possible distinct second-max and second-min relationships for a
set of data and adds them to the hypotheses slot of the problem unit."
 (let ((hypotheses nil)
        (wave (first view-angle-data))
        (view-angles (second view-angle-data)))
   (dolist (ang view-angles)
    (push (list 'second-max wave ang) hypotheses)
    (push (list 'second-min wave ang) hypotheses))
   (add.values unit 'hypotheses hypotheses)))

(defun full-string-data-in-plane(view-angles)
"Returns t if the string data is a full string in a plane and nil otherwise."
  (let ((copy-view-angles (copy-list view-angles)))
  (multiple-value-bind (backscatter-az forwardscatter-az)
     (get-azimuth-angles (azimuth-360 (first copy-view-angles)))
   (let ((nadir (find-if #'(lambda (x) (zerop (zenith x))) copy-view-angles))
          (backscatter (sort (find-half-string backscatter-az copy-view-angles)
                             #'< :key #'zenith))
          (forwardscatter (sort (find-half-string forwardscatter-az
                                                  copy-view-angles)
                                #'< :key #'zenith)))
     (and (= (length copy-view-angles)   ;All data makes up only 1 string
             (+ (if (null nadir) 0 1)
                (length backscatter)
                (length forwardscatter)))
          (half-string backscatter)
          (half-string forwardscatter))))))

(defun try-backscatter>forwardscatter-relationship (unit view-angle-data)
"Adds backscatter>forwardscatter to the hypotheses slot of the problem unit."
    (add.value unit 'hypotheses
          (list 'backscatter>forwardscatter (first view-angle-data))))
```

```
;;;-----------------------------------------------------------------------
;;; Functions to Output Results of Learning System
;;;-----------------------------------------------------------------------

(defun open-output-learning-results-interface ()
"Test to make sure the pre-requisite steps have been carried out - if not
prints error message and exits.  Opens the interface for displaying the results
of the learning system."
  (let ((option-number (get.value 'learning.methods 'option.number)))
    (case option-number
        (1 (when (not (get.value 'learning.methods
                                'done.learn.class.descriptions.p))
            (my-documentation-print
             "Learn class descriptions before outputing the results")
            (return-from open-output-learning-results-interface nil)))
        (2 (when (not (get.value 'learning.methods 'done.classify.sample.p))
            (my-documentation-print
             "Classify the unknown sample before outputting the results")
            (return-from open-output-learning-results-interface nil)))
        (3 (when (not
                   (get.value 'learning.methods 'done.classify.cover.types.p))
            (my-documentation-print
"Test class descriptions on cover types before outputting the results")
            (return-from open-output-learning-results-interface nil)))))
  (clear-prompt)
  (put.value 'output.learning.results 'message "")
  (put.value 'output.learning.results 'current.class.number 0)
  (remove.all.values 'output.learning.results 'menu)
  (unitmsg 'viewport-output.learning.results.1 'open-panel!)
  (display-learning-results 0))

(defun display-learning-results (num)
"Displays the results for one class in the learning system."
  (let* ((problem (nth num (get.values 'training.databases 'current.classes)))
         (class (get.values problem 'class)))
    (put.value 'output.learning.results 'class class)
    (put.values 'output.learning.results 'view.angle.data.message
         (get.values problem 'view.angle.data.message))
    (put.value 'output.learning.results 'solar.zenith
         (get.value problem 'solar.zenith))
    (put.value 'output.learning.results 'message
         (get-scores (get.values problem 'scores) class problem))))
```

```
(defun get-scores (scores class problem)
"Displays the classification scores for a class.  If option 2 has been
selected it also displays the class score for the unknown sample and
indicates if this class is the best class for the sample.  If option 3 has
been selected it also displays the classification performance score for
the class operating on the data base of cover type data."
  (let ((result "")
        (option-number (get.value 'learning.methods 'option.number)))
    (setf result (format () "Positive training set is:-~{ ~S~}  Negative training set is:-~{ ~S~}  Best
single hypothesis scores are:-  "
                          (get-unit-names
                           (get.values problem 'pos.training.set))
                          (get-unit-names
                           (get.values problem 'neg.training.set))))
    (dolist (score scores result)
     (let ((sc (first (first score))))
          (setf result (string-append result (format ()
             "~S score ~,4F at wavelength ~S  "
             (first sc)
             (second score)
             (second sc))))))
    (setf result (string-append result (format ()
          "  Best compound hypothesis scores are:-  ")))
    (dolist (score (get.values problem 'best.score))
     (setf result (string-append result (format ()
        "~S with score ~,4F  " (first score)(second score)))))
    (case option-number
     (2
        (setf result (string-append result (option-2-result problem class))))
     (3
        (setf result (string-append result (option-3-result problem)))))
    result))

 (defun option-2-result (problem class)
  (let ((result (format ()
                          "  Score for unknown sample for class ~S is ~,4F"
                          class
                          (get.value problem 'sample.score))))
    (when (get.value problem 'best.class)
     (setf result (string-append result (format ()
          "  The class ~S is the best class for this unknown sample" class))))
    result))
```

```
(defun option-3-result (problem)
 (let ((result "")
        (correct-matching-c-t
         (get-unit-names
          (get.values problem 'correct.matching.cover.types)))
        (incorrect-matching-c-t
         (get-unit-names
          (get.values problem 'incorrect.matching.cover.types))))
   (if correct-matching-c-t
        (setf result (format ()
"Cover types~{ ~S~} were correctly classified as belonging to this class.  "
                  correct-matching-c-t))
        (setf result (format ()
"No cover types were correctly classified as belonging to this class.  ")))
   (if incorrect-matching-c-t
        (setf result (string-append result (format ()
"Cover types~{ ~S~} were incorrectly classified as belonging to this class.  "
                  incorrect-matching-c-t)))
        (setf result (string-append result (format ()
"No cover types were incorrectly classified as belonging to this class.  "))))
   (string-append result (format ()
"The system's classification performance score is ~,4F"
          (get.value 'training.databases 'performance.score)))))

(defun display-next-learning-results ()
"Displays the next set of learning results, if any."
  (let* ((old-num (get.value 'output.learning.results 'current.class.number))
          (new-num (if (= (- (length (get.values 'training.databases
                                                 'current.classes))
                             old-num)
                          1)  ;Was displaying last class
                       0     ;Display first class again
                       (1+ old-num))))  ;Change to next class
    (put.value 'output.learning.results 'current.class.number new-num)
    (display-learning-results new-num)))

(defun display-previous-learning-results ()
"Displays the previous set of learning results, if any."
  (let* ((old-num (get.value 'output.learning.results 'current.class.number))
          (new-num (if (> old-num 0)    ;Was not displaying first class
                       (1- old-num)     ;Change to previous class
                       (1- (length (get.values 'training.databases
                                               'current.classes))))))
                         ;Change to last class
    (put.value 'output.learning.results 'current.class.number new-num)
    (display-learning-results new-num)))
```

```
;;;-------------------------------------------------------------------------
;;; Methods to classify an unknown sample
;;;-------------------------------------------------------------------------

(defun get-scores-for-sample ()
"If the scores slot of the first training class unit is empty, print an error
message and return nil.  Otherwise call the function classify-sample to
classify the unknown sample and return the best class for the sample."
  (let ((current-samples (get.values 'estimate.hemispherical.reflectance
                                     'current.sample.wavelengths)))
    (when (or (null current-samples)
              (not (get.value 'learning.methods
                              'done.learn.class.descriptions.p)))
      (my-documentation-print
"Enter unknown sample and learn class descriptions before classifying the sample")
      (return-from get-scores-for-sample nil))
    (my-documentation-print "Classifying unknown sample")
    (put.value
        (classify-sample (get-sample-data current-samples) 2)
        'best.class t)
    (put.value 'learning.methods 'done.classify.sample.p t)
    (my-documentation-print "Finished classifying unknown sample")))

(defun get-sample-data (samples)
"Returns a list of the wavelengths and reflectance data for all the samples."
  (let ((result ()))
    (dolist (sam samples result)
      (push (list (get.value sam 'wavelength)
                  (get.value sam 'reflectance.data))
            result))))
```

```
(defun classify-sample (data option-num)
"Data is a list of wavelengths and corresponding directional reflectance data.
Classify the data by testing the best class hypotheses for each class on the
data and determining the matching score for the class. Return the best class
for the sample."
 (let ((best-score -2)
       (best-class nil))
   (dolist (problem (get.values 'training.databases 'current.classes))
    (let ((class-ev (get.values problem 'best.score))
          (e-sup 0)
          (e-opp 0))
        (dolist (ev class-ev)
         (let ((hyp-score (second ev))
               (total 0))
           (dolist (relation (first (first ev)))
            (let* ((rel (first (first relation)))
                   (func (first rel))
                   (arg1 (second rel))
                   (arg2 (third rel))
                   (flag (second (first relation)))
                   (wave (second relation)))
               (dolist (dat data)
                (when (= wave (first dat))
                  (incf total (funcall func (second dat)
                                flag arg1 arg2))))))
           (if (= total (length (first (first ev))))
               (incf e-sup hyp-score)
               (incf e-opp hyp-score))))
        (let ((score (if (zerop (length class-ev))
                         0
                         (if (> e-sup e-opp)
                           (- 1 (/ e-opp e-sup))
                           (- (/ e-sup e-opp) 1)))))
          (when (= option-num 2)(put.value problem 'sample.score score))
          (when (> score best-score)
           (setf best-score score)
           (setf best-class problem)))))
   best-class))

;;;-----------------------------------------------------------------------
;;; Methods to test class descriptions on cover types
;;;-----------------------------------------------------------------------

(defun test-class-descriptions-on-cover-types ()
"Checks that the pre-requisite steps have been carried out - if not prints an
error message and exits the function."
  (when (not (get.value 'learning.methods 'done.learn.class.descriptions.p))
    (my-documentation-print
"Learn class descriptions before testing the class descriptions on the data base of cover types")
    (return-from test-class-descriptions-on-cover-types nil))
  (test-class-descriptions-on-cover-types-aux))
```

```
(defun test-class-descriptions-on-cover-types-aux()
"Tests the class definitions on the cover types contained in the positive
training sets for each posible class definition.  If the best class for the
sample is the class for which it is a positive training example, add 1 to
pos-total.  Return the score which is the proportion of correctly classified
samples.  In this version it is assumed that each class contains only one
wavelength."
  (let ((pos-total 0)
        (num-pos 0))
    (dolist (problem (get.values 'training.databases 'current.classes))
      (let ((training-data-sets (unit.children problem 'subclass)))
        (dolist (parent (unit.children (second training-data-sets)
                                       `subclass))
          (let ((data ()))
            (dolist (unit (unit.children parent 'member))
              (push (list (get.value unit 'wavelength)
                          (get.value unit 'reflectance.data))
                    data))
            (incf num-pos)
            (let ((best-class (classify-sample data 3)))
              (cond ((eq best-class problem)
                     (incf pos-total)
                     (add.value best-class 'correct.matching.cover.types
                          (get.value parent 'cover.type)))
                    (t (add.value best-class 'incorrect.matching.cover.types
                            (get.value parent 'cover.type))))))))))
     (put.value 'training.databases 'performance.score
          (/ pos-total num-pos)))
   (put.value 'learning.methods 'done.classify.cover.types.p t))

(load "learn-methods1")

(load "learn-methods2")

(load "learn-methods3")
```

```
;;; learn-methods1.lisp
;;;
;;; Additional methods for constructing and testing compound hypotheses in the
;;; learning system
;;;
;;; Written by Ann and Patrick Harrison
;;; Created 30th July 1992
;;; Last modified18th September 1992

(in-package 'kee)

;;;-----------------------------------------------------------------------
;;;Additional methods for improved search technique
;;;-----------------------------------------------------------------------

(defun ask-about-trace-file ()
"Display the screen to ask the user whether or not the hypothesis testing
should be traced."
  (remove.all.values 'learning.methods 'yes.no)
  (remove.all.values 'learning.methods 'trace.file)
  (put.value 'learning.methods 'message
    "Do you want to trace the hypothesis testing?")
  (unitmsg 'viewport-learning.methods.1 'open-panel!))

(defun generalization-search ()
"Opens the trace file, if necessary and then calls the function to perform an
exhaustive search and testing of all hypotheses up to the depth specified."
   (let ((trace-file (get.value 'learning.methods 'trace.file))
         (current-classes (get.values 'training.databases 'current.classes)))
     (if trace-file
         (with-open-file (trace-str trace-file :direction :output
                                    :if-does-not-exist :create)
           (generalization-search-aux-1 current-classes trace-str))
         (generalization-search-aux-1 current-classes nil))))
```

```
(defun generalization-search-aux-1 (current-classes trace-str)
"Calls the function to test the single hypotheses. Then displays the interrupt
button and calls another function to test the compound hypotheses. At the end
of the search write the time to the trace file and removes the interrupt button
from the screen. "
  (let ((option-number (get.value 'learning.methods 'option.number)))
    (test-level-1 current-classes trace-str)
    (display-interrupt-button option-number)
    (catch 'search-over
     (generalization-search-aux-2
         current-classes
         (get.value 'learning.methods 'search.depth)
         trace-str))
    (when trace-str
     (multiple-value-bind (a b c d e f) (get-decoded-time)
         (princ (format ()
                        "~%Learning completed ~S/~S/~S at time ~S.~S.~S"
                        f e d c b a)
                trace-str)))
    (my-documentation-print "Finished learning class descriptions")
    (remove-interrupt-button option-number)))

(defun generalization-search-aux-2 (current-classes level str)
"Searches at increasingly deeper levels for the best hypothesis for each
problem. Completes the search at one level for all the problems before moving
to the next level. This is so that if the search is interrupted an
intermediate result for all problems is available."
  (let ((current-best-score nil))
    (when (> level 1)
     (dolist (problem current-classes)
         (setf current-best-score (second (get.value problem 'best.score)))
         (put.value problem 'previous.best.score current-best-score)
         (when (/= current-best-score 1)
          (test-level-2 problem current-best-score)
          (when str
           (princ-report 2 problem str))))
     (when (> level 2)
         (dolist (problem current-classes)
          (setf current-best-score (get-improved-score problem))
          (when current-best-score
           (test-level-3 problem current-best-score)
           (when str
            (princ-report 3 problem str))))
         (when (> level 3)
          (dolist (problem current-classes)
           (setf current-best-score (get-improved-score problem))
           (when current-best-score
            (test-level-4 problem current-best-score)
            (when str
                (princ-report 4 problem str))))
```

```
        (when (> level 4)
         (dolist (problem current-classes)
          (setf current-best-score (get-improved-score problem))
          (when current-best-score
               (test-level-5 problem current-best-score)
               (when str
                 (princ-report 5 problem str))))))))))

(defun display-interrupt-button (option-number)
  (remove.all.values 'learning.methods 'search.over)
  (case option-number
    (1 (unitmsg 'windowpane-search.over-of-learning.methods.2 'open!))
    (2 (unitmsg 'windowpane-search.over-of-learning.methods.1 'open!))
    (3 (unitmsg 'windowpane-search.over-of-learning.methods.3 'open!))))

(defun remove-interrupt-button (option-number)
  (case option-number
    (1 (unitmsg 'windowpane-search.over-of-learning.methods.2 'close!))
    (2 (unitmsg 'windowpane-search.over-of-learning.methods.1 'close!))
    (3 (unitmsg 'windowpane-search.over-of-learning.methods.3 'close!))))

(defun princ-level-1-scores (problem str)
"Writes the level 1 scores for a problem to the trace file."
  (princ (format ()
                 "~%Problem ~S Level 1~%Best scores"
                 (unit.name problem)) str)
  (dolist (score (get.values problem 'scores))
    (princ (format () "~% ~S Overall score ~,4F Positive ~,4F Negative ~,4F"
                   (first score)(second score)(third score)(fourth score))
           str)))

(defun princ-report (level problem str)
"Writes the scores for one problem and level."
  (princ (format ()
                 "~%Problem ~S Level ~S~%Best scores"
                 (unit.name problem) level) str)
  (dolist (score (get.values problem 'best.score))
    (princ (format () "~% ~S Overall score ~,4F Positive ~,4F Negative ~,4F"
                   (first score)(second score)(third score)(fourth score))
           str)))

(defun get-improved-score (problem)
"Returns the current best score for the problem if it is at least 10% greater
than the previous best score and nil otherwise."
  (let ((previous-best-score (get.value problem 'previous.best.score))
        (current-best-score (second (get.value problem 'best.score))))
    (when (> (- current-best-score previous-best-score)
             (* previous-best-score 0.1))
      (put.value problem 'previous.best.score current-best-score)
      current-best-score)))
```

```
(defun test-level-1 (current-classes trace-str)
"Tests all the single (level 1) hypotheses for all classes.  Copies all
possible components of compound hypotheses in the HYPOTHESES slot of each
problem."
  (dolist (problem current-classes)
    (let ((scores nil))
      (dolist (hyp (get.values problem 'hypotheses))
          (multiple-value-bind (score-t score-nil)
            (apply #'score problem hyp)
           (push score-t scores)
           (push score-nil scores)))
      (setf scores (sort scores #'> :key #'second))
      (put.values problem 'hypotheses scores)
      (put.values problem 'scores
           (initial-get-best-scores (copy-tree scores) problem))
      (when trace-str
          (princ-level-1-scores problem trace-str))
      (initial-reduce-hypothesis-set problem))))

(defun test-level-2 (problem best-score-prev-level)
"Tests all the possible compund hypotheses cocnsiting of two single hypotheses anded together
(level 2 hypotheses) for a problem."
  (let ((hyps (get.values problem 'hypotheses)))
    (dolist (a hyps)
      (let ((a1 (first a)))
          (dolist (b (rest (member a hyps :test #'equal)))
            (when (get.value 'learning.methods 'search.over)
              (throw 'search-over))
            (test-hyp problem
                    (append a1 (first b)) best-score-prev-level))))))

(defun test-level-3 (problem best-score-prev-level)
"Tests all the possible level 3 hypotheses for a problem."
  (reduce-hypothesis-set problem)
  (let ((hyps (get.values problem 'hypotheses)))
    (dolist (a hyps)
      (let ((a1 (first a)))
          (dolist (b (rest (member a hyps :test #'equal)))
            (let ((b1 (first b)))
              (when (get.value 'learning.methods 'search.over)
                (throw 'search-over))
              (dolist (c (rest (member b hyps :test #'equal)))
                (test-hyp problem
                        (append a1 b1 (first c)) best-score-prev-level)))))))))
```

```
(defun test-level-4 (problem best-score-prev-level)
"Tests all the possible level 4 hypotheses for a problem."
  (reduce-hypothesis-set problem)
  (let ((hyps (get.values problem 'hypotheses)))
    (dolist (a hyps)
      (let ((a1 (first a)))
           (dolist (b (rest (member a hyps :test #'equal)))
             (let ((b1 (first b)))
               (dolist (c (rest (member b hyps :test #'equal)))
                 (let ((c1 (first c)))
                      (when (get.value 'learning.methods 'search.over)
                        (throw 'search-over))
                      (dolist (d (rest (member c hyps :test #'equal)))
                        (test-hyp problem
                                  (append a1 b1 c1 (first d))
                                  best-score-prev-level))))))))))

(defun test-level-5 (problem best-score-prev-level)
"Tests all the possible level 5 hypotheses for a problem."
  (reduce-hypothesis-set problem)
  (let ((hyps (get.values problem 'hypotheses)))
    (dolist (a hyps)
      (let ((a1 (first a)))
           (dolist (b (rest (member a hyps :test #'equal)))
             (let ((b1 (first b)))
               (dolist (c (rest (member b hyps :test #'equal)))
                 (let ((c1 (first c)))
                      (dolist (d (rest (member c hyps :test #'equal)))
                        (let ((d1 (first d)))
                          (when (get.value 'learning.methods 'search.over)
                            (throw 'search-over))
                          (dolist (e (rest (member d hyps :test #'equal)))
                            (test-hyp problem
                                      (append a1 b1 c1 d1 (first e))
                                      best-score-prev-level))))))))))))

(defun test-hyp (problem this-hyp best-score-prev-level)
"Calls a function to test a hypothesis. Updates the best.score slot when
necessary. Only adds to the best score slot a score that is better than the
best score at the previous level."
  (let* ((this-score (funcall #'complex-score problem this-hyp))
         (this-actual-score (second this-score))
         (best-actual-score (second (get.value problem 'best.score))))
    (when (> this-actual-score best-score-prev-level)
      (cond ((> this-actual-score best-actual-score)
             (put.value problem 'best.score this-score)
             (setf best-actual-score this-actual-score))
            ((= this-actual-score best-actual-score)
             (add.value problem 'best.score this-score))))))
```

```
(defun initial-get-best-scores (scores problem)
"Puts all the relationships having the best actual score into the best.score
slot of the problem. Then returns the best number of scores."
   (let* ((best-hyp-and-score (first scores))
          (best-actual-score (second best-hyp-and-score)))
     (put.value problem 'best.score
         (convert-to-best-format best-hyp-and-score))
     (dolist (this-hyp (rest scores))
        (if (= best-actual-score (second this-hyp))
           (add.value problem 'best.score (convert-to-best-format this-hyp))
           (return-from nil))))
   (let ((required-num-scores (* (get.value 'training.databases 'num.scores)
                                  (length (get.values problem 'wavelength))))
         (actual-num-scores (length scores)))
     (if (<= actual-num-scores required-num-scores)
         scores
         (butlast scores (- actual-num-scores required-num-scores)))))

(defun convert-to-best-format (single-hyp)
"Converts the format of a single hypothesis to the format consistent with the
way compound hypotheses will be stored in the best.score slot."
  `((,(first single-hyp) t) ,@(rest single-hyp)))

(defun initial-reduce-hypothesis-set (problem)
"This heuristic function reduces the list of possible hypotheses. A compound
hypothesis cannot have a discrimination score that is greater than the minimum
of its components' positive training set scores so all hypotheses with positive
score less than the current best score are removed. Also, if a hypotheses
scores 1 for both the positive and negative training sets it does not
discriminate at all and it is removed from the hypothesis set. If a hypothesis
scores 0 for the negative training set, combining it with other hypotheses
cannot reduce this value so it is removed from the set of hypotheses."
  (let ((best-actual-score (second (get.value problem 'best.score))))
    (dolist (this-hyp (get.values problem 'hypotheses))
      (let ((this-pos-score (third this-hyp))
            (this-neg-score (fourth this-hyp)))
        (when (or (<= this-pos-score best-actual-score) ;Cannot be better
                  (and (= this-pos-score 1)            ;Hyp true for all +ve
                       (= this-neg-score 1))           ;and -ve training set
                  (zerop this-neg-score))              ;Cannot be better
          (remove.value problem 'hypotheses this-hyp))))))

(defun reduce-hypothesis-set (problem)
"This heuristic function reduces the list of possible hypotheses. A compound
hypothesis cannot have a discrimination score that is greater than the minimum
of its components' positive training set score so all hypotheses with positive
score less than the current best score are removed. This function is called
after the level 2 and subsequent level searches."
  (let ((best-actual-score (second (get.value problem 'best.score))))
    (dolist (this-hyp (get.values problem 'hypotheses))
      (when (< (third this-hyp) best-actual-score)    ;Cannot be better
        (remove.value problem 'hypotheses this-hyp)))))
```

```
(defun complex-score (problem data)
"Calculates the discrimination score for a relationship.  For example it
might test ((GREATER-THAN (60 180)(20 0)) T) and return the scores for this
hypothesis."
    (let ((training-data-sets (unit.children problem 'subclass)))
        (multiple-value-bind (num-pos pos-total)
          (complex-score-aux (second training-data-sets) data)
          (multiple-value-bind (num-neg neg-total)
            (complex-score-aux (first training-data-sets) data)
           (let ((pos-score (/ pos-total num-pos))
                 (neg-score (/ neg-total num-neg)))
            (make-complex-result data t
                                 (- pos-score neg-score)
                                 pos-score
                                 neg-score))))))


(defun complex-score-aux (training-set data)
"Returns the number of elements in the training set and the number that
matched the hypothesis."
  (let ((total 0)
        (num 0))
    (dolist (parent (unit.children training-set `subclass))
      (incf num)
      (incf total (find-score data parent)))
    (values num total)))

(defun find-score (data parent)
"Tests whether a training set member  matches a compound hypothesis - ie all
the single hypotheses in the compound hypothesis are true for the reflectance
data in the training set member.  Returns 1 if the data matches the hypothesis
and nil otherwise."
  (let ((total 0))
    (dolist (dat data)
      (let* ((wave (second dat))
             (rel (first dat))
             (func (first (first rel)))
             (arg1 (second (first rel)))
             (arg2 (third (first rel)))
             (old-flag (second rel)))
        (dolist (unit (unit.children parent 'member))
          (when (= wave (get.value unit 'wavelength))
            (incf total
                  (funcall func (get.value unit 'reflectance.data)
                           old-flag arg1 arg2))))))
    (if (= total (length data))
        1
        0)))

(defun make-complex-result (data flag s1 s2 s3)
"Puts the elements of a hypothesis into the required form."
  `((,data ,flag) ,s1 ,s2 ,s3))
```

```
;;; learn-methods2.lisp
;;;
;;; Output of data from the Learning System to a file
;;;
;;; Written by Ann Harrison
;;; Created 11th September 1992
;;; Last Modified 21st September 1992

(in-package 'kee)

(defun ls-open-output-to-file-interface ()
"Opens the interface for outputting the results to a file."
  (remove.all.values 'output.learning.results 'output.parameters)
  (remove.all.values 'output.learning.results 'format.list)
  (put.value 'output.learning.results 'message
"Select the parameters to be output to the file, in the correct order =>")
  (unitmsg 'viewport-output.learning.results.2 'open-panel!))

(defun ls-open-output-to-file-template-interface ()
"Opens the window that allows the user to select the format for the
reflectance data."
  (put.value 'windowpane-output.parameters-of-output.learning.results.5
     'mouseleftfn!
     'deactivate-left-mouse)
  (remove.all.values 'output.learning.results 'template.number)
  (put.value 'output.learning.results 'message
     "Select the required template =>")
  (unitmsg 'windowpane-template.message-of-output.learning.results.8 'open!)
  (unitmsg 'windowpane-template.number-of-output.learning.results.7 'open!))

(defun ls-store-template (template)
"Stores the format corresponding to the specified format."
  (put.values 'output.learning.results 'format.list
    (case template
        (template.1 '(template.1))
        (template.2 '(results class.definition view.angles solar.zenith))
        (template.3 '(results class.definition training.sets view.angles
                      solar.zenith)))))

(defun ls-write-results-to-file ()
"Writes the results to a file in the specified format."
  (let ((format-list (reverse (get.values 'output.learning.results
                                'format.list))))
    (with-open-file (out-str (get.value '9.output 'output.file.name)
                             :direction :output
                             :if-does-not-exist :create)
      (if (eq (first format-list) 'template.1)
          (ls-output-data-to-file
           (get.values 'training.databases
                'current.classes)
           out-str)
          (ls-write-results-to-file-aux out-str format-list)))))
```

```
(defun ls-write-results-to-file-aux (out-str format-list)
   (dolist (class (get.values 'training.databases
                       'current.classes))
    (dolist (parameter format-list)
        (if (eq parameter 'done)
          nil
          (ls-write-simple-results-to-file out-str class parameter)))))

(defun ls-write-simple-results-to-file (out-str class parameter)
 (case parameter
  (solar.zenith (princ
                  (get.value class 'solar.zenith) out-str))
  (wavelength (princ
                 (get.values class 'wavelength) out-str))
  (view.angles (princ (get.values class 'view.angle.data.message) out-str))
  (class.definition (princ (get.value class 'class) out-str))
  (training.sets (ls-princ-training-sets class out-str))
  (results (ls-write-results class out-str)))
 (princ " " out-str))

(defun ls-write-results (class out-str)
 (dolist (score (get.values class 'scores))
  (let ((sc (first (first score))))
   (princ (format () "~S score ~,4F at wavelength ~S "
                   (first sc)
                   (second score)
                   (second sc)) out-str)))
 (dolist (score (get.values class 'best.score))
  (princ (format () "~S with score ~,4F  "
                 (first score)(second score)) out-str))
 (let ((sample-score (get.value class 'sample.score)))
  (when sample-score
   (let ((this-class (get.value class 'class)))
       (princ (format ()
                    "Score for unknown sample for class ~S is ~,4F "
                    this-class
                    sample-score) out-str)
       (when (get.value class 'best.class)
        (princ (format ()
                      "The class ~S is the best class for the sample "
                      this-class) out-str)))))
 (let ((performance-score
        (get.value 'training.databases 'performance.score)))
  (when performance-score
   (princ (format ()
                  "The system's classification performance score was ~,4F "
                  performance-score)
                  out-str))))
```

```
(defun ls-princ-training-sets (class out-str)
 (princ (format ()
        "Positive training set is:--{ ~S~} Negative training set is:--{ ~S~} "
        (get-unit-names (get.values class 'pos.training.set))
        (get-unit-names (get.values class 'neg.training.set)))
        out-str)
 (when (= (get.value 'learning.methods 'option.number) 3)
  (let ((correct-matching-c-t
          (get-unit-names
           (get.values class 'correct.matching.cover.types)))
         (incorrect-matching-c-t
          (get-unit-names
           (get.values class 'incorrect.matching.cover.types))))
    (if correct-matching-c-t
         (princ (format ()
"Cover types~{ ~S~} were correctly classified as belonging to this class "
             correct-matching-c-t) out-str)
         (princ
"No cover types were correctly classified as belonging to this class "
        out-str))
    (if incorrect-matching-c-t
         (princ (format ()
"Cover types~{ ~S~} were incorrectly classified as belonging to this class"
             correct-matching-c-t) out-str)
         (princ
"No cover types were incorrectly classified as belonging to this class"
        out-str)))))

(defun ls-output-data-to-file (classes out-str)
 (let ((class (first classes)))
  (princ (format () "~%Solar Zenith Angle:- ~S"
                  (get.value class 'solar.zenith))
          out-str)
  (terpri out-str)
  (princ "View Angle Data:- " out-str)
  (dolist (mess (get.values class 'view.angle.data.message))
   (terpri out-str)
   (princ mess out-str)))
 (dolist (class classes)
  (let ((class-def (get.value class 'class)))
   (princ (format () "~2%Class Definition:- ~S"
                  class-def)
          out-str)
   (ls-get-scores (get.values class 'scores) class-def class out-str))))
```

```
(defun ls-get-scores (scores class problem out-str)
"Writes to the file the classification scores for a class.  If option 2 has
been selected it also writes the class score for the unknown sample and
indicates if this class is the best class for the sample.  If option 3 has
been selected it also writes the classification performance score for
the class operating on the data base of cover type data."
  (princ (format () "~%Positive training set is:-~{ ~S~}"
                 (get-unit-names
                  (get.values problem 'pos.training.set)))
         out-str)
  (princ (format () "~%Negative training set is:-~{ ~S~}"
                 (get-unit-names
                  (get.values problem 'neg.training.set)))
         out-str)
  (princ (format () "~%Best single hypothesis scores are:-") out-str)
  (dolist (score scores)
   (let ((sc (first (first score))))
     (princ (format () "~%~S score ~,4F at wavelength ~S"
                    (first sc)
                    (second score)
                    (second sc)) out-str)))
  (princ (format () "~%Best compound hypothesis scores are:-")  out-str)
  (dolist (score (get.values problem 'best.score))
   (princ (format () "~%~S with score ~,4F  "
                  (first score)(second score)) out-str))
  (let ((sample-score (get.value problem 'sample.score)))
   (when sample-score
     (ls-option-2-result problem class sample-score out-str)))
  (let ((performance-score
         (get.value 'training.databases 'performance.score)))
   (when performance-score
     (ls-option-3-result problem performance-score out-str))))

(defun ls-option-2-result (problem class sample-score out-str)
"Write to the file the results that apply to option 2 only."
  (princ (format ()
                 "~%Score for unknown sample for class ~S is ~,4F"
                 class
                 sample-score) out-str)
  (when (get.value problem 'best.class)
   (princ (format ()
                  "~%The class ~S is the best class for this unknown sample"
                  class) out-str)))
```

```
(defun ls-option-3-result (problem performance-score out-str)
"Writes to the file the results that apply to option 3 only."
  (let ((correct-matching-c-t
          (get-unit-names
           (get.values problem 'correct.matching.cover.types)))
         (incorrect-matching-c-t
          (get-unit-names
           (get.values problem 'incorrect.matching.cover.types))))
   (if correct-matching-c-t
        (princ (format ()
"~%Cover types~{ ~S~} were correctly classified as belonging to this class."
               correct-matching-c-t) out-str)
        (princ (format ()
"~%No cover types were correctly classified as belonging to this class.")
               out-str))
   (if incorrect-matching-c-t
        (princ (format ()
"~%Cover types~{ ~S~} were incorrectly classified as belonging to this class."
               incorrect-matching-c-t) out-str)
        (princ (format ()
"~%No cover types were incorrectly classified as belonging to this class.")
               out-str))
   (princ (format ()
"~%The system's classification performance score is ~,4F"
        performance-score)
          out-str)))
```

```
;;; learn-methods3.lisp
;;;
;;; Code for the Learning System - Automatic Mode
;;;
;;; Written by Ann and Patrick Harrison
;;; Created 18th September 1992
;;; Last Modified 25th September 1992

(in-package 'kee)

(defun auto-enter-classes ()
  (clear-prompt)
  (wipe-out-results 'dummy)
  (unitmsg 'windowpane-wavelength-of-enter.learning.data.1 'close!)
  (unitmsg 'windowpane-solar.zenith-of-enter.learning.data.2 'close!)
  (unitmsg 'windowpane-view.angle.data.message-of-enter.learning.data.4
           'close!)
  (remove.all.values 'enter.learning.data 'class)
  (remove.all.values 'enter.learning.data 'class.parameter)
  (put.value 'enter.learning.data 'menu 'enter.data)
  (unitmsg 'viewport-enter.learning.data.2 'open-panel!))

(defun auto-learn-class-descriptions (input-file output-file classes)
  (remove.all.values 'estimate.hemispherical.reflectance 'new.samples)
  (catch 'invalid-data
    (input-data-from-file input-file)
    (with-open-file (output-str output-file :direction :output
                                :if-does-not-exist :create)
      (dolist (sample (get.values 'estimate.hemispherical.reflectance
                                  'new.samples))
          (auto-learn-class-descriptions-aux output-str classes sample)
          (dolist (uni (unit.children sample 'member))
            (delete.unit uni))
          (delete.unit sample)))
    (initialize-auto-system)))

(defun auto-learn-class-descriptions-aux (output-str classes sample)
  (let ((wavelength-units (unit.children sample 'member)))
    (complete-training-classes classes sample wavelength-units)
    (catch 'no-training-sets
      (dolist (problem classes)
          (generate-training-sets problem)
          (interp-extrap-training-data problem))
      (forward.chain 'learning.rules)
      (generalization-search-aux-1 classes nil)
      (put.value (classify-sample (get-sample-data wavelength-units) 2)
            'best.class t)
      (test-class-descriptions-on-cover-types-aux)
      (ls-output-data-to-file classes output-str))))
```

C-2

```
(defun complete-training-classes (classes sample wavelength-units)
  (remove.all.values 'enter.learning.data 'view.angle.data)
  (remove.all.values 'enter.learning.data 'view.angle.data.message)
  (get-view-angle-data wavelength-units)
  (let* ((view-angle-data
            (get.values 'enter.learning.data 'view.angle.data))
           (view-angle-data-message
            (get.values 'enter.learning.data 'view.angle.data.message))
           (solar-zenith (get.value sample 'solar.zenith))
           (wavelengths (mapcar #'first view-angle-data)))
    (dolist (problem classes)
      (put.values problem 'view.angle.data view-angle-data)
      (put.values problem 'view.angle.data.message
            view-angle-data-message)
      (put.value problem 'solar.zenith solar-zenith)
      (put.values problem 'wavelength wavelengths)
      (initialize-training-set problem)
      (delete-sub-units problem)))
  (remove.all.values 'training.databases 'performance.score))
```

APPENDIX B

TEST RESULTS

test1-trace

Problem TRAINING.PROB.387 Level 1
Best scores
((((GREATER-THAN (15 182) (75 90)) T) 0.68)) Overall score 0.7738 Positive 0.9167 Negative 0.1429
((((FIRST-MAX (15 182)) T) 0.68)) Overall score 0.7738 Positive 0.9167 Negative 0.1429
((((FIRST-MAX (75 90)) NIL) 0.68)) Overall score 0.7738 Positive 0.9167 Negative 0.1429
((((GREATER-THAN (75 90) (0 0)) NIL) 0.68)) Overall score 0.7500 Positive 0.7500 Negative 0.0000
((((GREATER-THAN (75 90) (35 45)) NIL) 0.68)) Overall score 0.5833 Positive 0.5833 Negative 0.0000

Problem TRAINING.PROB.387 Level 2
Best scores
(((((FIRST-MAX (75 90)) NIL) 0.68)) T) Overall score 0.7738 Positive 0.9167 Negative 0.1429
(((((FIRST-MAX (15 182)) T) 0.68)) T) Overall score 0.7738 Positive 0.9167 Negative 0.1429
(((((GREATER-THAN (15 182) (75 90)) T) 0.68)) T) Overall score 0.7738 Positive 0.9167 Negative 0.1429
Learning completed 1992/9/22 at time 12.58.57

test1-option1

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((15 182) (75 90) (0 0) (35 45))

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 CT9-46
Negative training set is:- CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT11-58 CT5-42 CT6-45 CT7-41 CT8-42 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (15 182) (75 90)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (15 182)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (75 90)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 90) (0 0)) NIL) score 0.7500 at wavelength 0.68
((GREATER-THAN (75 90) (35 45)) NIL) score 0.5833 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MAX (75 90)) NIL) 0.68)) T) with score 0.7738
(((((FIRST-MAX (15 182)) T) 0.68)) T) with score 0.7738
(((((GREATER-THAN (15 182) (75 90)) T) 0.68)) T) with score 0.7738

test2-trace

Problem TRAINING.PROB.3 Level 1
Best scores
((((GREATER-THAN (30 275) (50 45)) T) 0.68)) Overall score 0.6129 Positive 1.0000 Negative 0.3871
((((GREATER-THAN (30 275) (45 85)) T) 0.68)) Overall score 0.5346 Positive 0.8571 Negative 0.3226
((((GREATER-THAN (30 93) (2 45)) NIL) 0.68)) Overall score 0.5207 Positive 0.7143 Negative 0.1935
((((GREATER-THAN (45 270) (45 85)) T) 0.68)) Overall score 0.5023 Positive 0.8571 Negative 0.3548
((((GREATER-THAN (45 270) (60 355)) T) 0.68)) Overall score 0.5023 Positive 0.8571 Negative 0.3548
((((GREATER-THAN (60 275) (60 355)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (60 87) (60 355)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (30 93) (50 45)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (50 227) (60 355)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (35 48) (0 0)) NIL) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161

Problem TRAINING.PROB.2 Level 1
Best scores
((((GREATER-THAN (60 275) (60 87)) NIL) 0.68)) Overall score 0.6250 Positive 1.0000 Negative 0.3750
((((GREATER-THAN (60 87) (60 355)) T) 0.68)) Overall score 0.5000 Positive 1.0000 Negative 0.5000
((((GREATER-THAN (60 275) (45 85)) NIL) 0.68)) Overall score 0.4875 Positive 0.8000 Negative 0.3125
((((GREATER-THAN (60 87) (75 2)) T) 0.68)) Overall score 0.4875 Positive 0.8000 Negative 0.3125
((((GREATER-THAN (60 87) (65 40)) T) 0.68)) Overall score 0.4625 Positive 0.9000 Negative 0.4375
((((GREATER-THAN (45 270) (45 356)) T) 0.68)) Overall score 0.4375 Positive 1.0000 Negative 0.5625
((((GREATER-THAN (30 93) (2 45)) T) 0.68)) Overall score 0.4375 Positive 1.0000 Negative 0.5625
((((GREATER-THAN (75 2) (75 180)) NIL) 0.68)) Overall score 0.4375 Positive 1.0000 Negative 0.5625
((((GREATER-THAN (15 270) (60 355)) T) 0.68)) Overall score 0.4250 Positive 0.8000 Negative 0.3750
((((GREATER-THAN (45 85) (65 40)) T) 0.68)) Overall score 0.4250 Positive 0.8000 Negative 0.3750

Problem TRAINING.PROB.1 Level 1
Best scores
((((GREATER-THAN (75 2) (60 180)) T) 0.68)) Overall score 0.7333 Positive 1.0000 Negative 0.2667
((((GREATER-THAN (60 275) (50 227)) T) 0.68)) Overall score 0.7000 Positive 1.0000 Negative 0.3000

((((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) Overall score 0.6667 Positive 1.0000 Negative 0.3333
((((GREATER-THAN (75 2) (45 178)) T) 0.68)) Overall score 0.6667 Positive 1.0000 Negative 0.3333
((((GREATER-THAN (60 275) (60 87)) T) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (60 87) (30 93)) T) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (45 85) (60 355)) NIL) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (65 225) (75 2)) NIL) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (50 227) (75 2)) NIL) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (75 2) (30 180)) T) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667

Problem TRAINING.PROB.3 Level 2
Best scores
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((FIRST-MAX (60 180)) NIL) 0.68)) T) Overall score 0.7742 Positive 1.0000 Negative 0.2258
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((SECOND-MAX (75 180)) NIL) 0.68)) T) Overall score 0.7742 Positive 1.0000 Negative 0.2258
Problem TRAINING.PROB.2 Level 2
Best scores
(((((GREATER-THAN (60 87) (60 355)) T) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) Overall score 0.8125 Positive 1.0000 Negative 0.1875
(((((GREATER-THAN (60 275) (60 87)) NIL) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) Overall score 0.8125 Positive 1.0000 Negative 0.1875

Problem TRAINING.PROB.1 Level 2
Best scores
(((((GREATER-THAN (60 275) (60 87)) T) 0.68) (((GREATER-THAN (45 270) (0 0)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
(((((GREATER-THAN (75 2) (60 180)) T) 0.68) (((GREATER-THAN (15 225) (15 182)) NIL) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
Learning completed 1992/9/28 at time 9.29.39

test2-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((60 275) (45 270) (30 275) (15 270) (60 87) (45 85) (30 93) (2 90) (65 225) (50 227) (35 220) (15 225) (65 40) (50 45) (35 48) (15 46) (2 45) (75 2) (75 180) (60 355) (60 180) (45 356) (45 178) (30 5) (30 180) (15 7) (15 182) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (DESCRIPTION WHEAT)
Positive training set is:- CT5-26 CT4-27 CT5-28 CT5-59 CT4-32 CT4-51 CT5-42
Negative training set is:- CT11-82 CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT8-56 CT11-58 CT2-50 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49 CT11-45

Best single hypothesis scores are:-
((GREATER-THAN (30 275) (50 45)) T) score 0.6129 at wavelength 0.68
((GREATER-THAN (30 275) (45 85)) T) score 0.5346 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) NIL) score 0.5207 at wavelength 0.68
((GREATER-THAN (45 270) (45 85)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (45 270) (60 355)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (60 275) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (30 93) (50 45)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (50 227) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (35 48) (0 0)) NIL) score 0.4839 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((FIRST-MAX (60 180)) NIL) 0.68)) T) with score 0.7742
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((SECOND-MAX (75 180)) NIL) 0.68)) T) with score 0.7742
Score for unknown sample for class (DESCRIPTION WHEAT) is -1.0000

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT2-28 CT2-30 CT3-27 CT3-63 CT3-35 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45
Negative training set is:- CT1-30 CT4-27 CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT4-32 CT4-51 CT1-45 CT5-42 CT6-45 CT7-41 CT9-46 CT10-49
Best single hypothesis scores are:-
((GREATER-THAN (60 275) (60 87)) NIL) score 0.6250 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.5000 at wavelength 0.68
((GREATER-THAN (60 275) (45 85)) NIL) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (75 2)) T) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (65 40)) T) score 0.4625 at wavelength 0.68
((GREATER-THAN (45 270) (45 356)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (75 2) (75 180)) NIL) score 0.4375 at wavelength 0.68
((GREATER-THAN (15 270) (60 355)) T) score 0.4250 at wavelength 0.68
((GREATER-THAN (45 85) (65 40)) T) score 0.4250 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 87) (60 355)) T) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
(((((GREATER-THAN (60 275) (60 87)) NIL) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 2) (60 180)) T) score 0.7333 at wavelength 0.68
((GREATER-THAN (60 275) (50 227)) T) score 0.7000 at wavelength 0.68
((GREATER-THAN (30 93) (60 355)) NIL) score 0.6667 at wavelength 0.68
((GREATER-THAN (75 2) (45 178)) T) score 0.6667 at wavelength 0.68
((GREATER-THAN (60 275) (60 87)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (60 87) (30 93)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (45 85) (60 355)) NIL) score 0.6333 at wavelength 0.68

((GREATER-THAN (65 225) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (50 227) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (75 2) (30 180)) T) score 0.6333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 275) (60 87)) T) 0.68) (((GREATER-THAN (45 270) (0 0)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (75 2) (60 180)) T) 0.68) (((GREATER-THAN (15 225) (15 182)) NIL) 0.68)) T) with score 0.9000
Score for unknown sample for class (DESCRIPTION FOREST) is 0.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample

test3-option3

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((60 275) (45 270) (30 275) (15 270) (60 87) (45 85) (30 93) (2 90) (65 225) (50 227) (35 220) (15 225) (65 40) (50 45) (35 48) (15 46) (2 45) (75 2) (75 180) (60 355) (60 180) (45 356) (45 178) (30 5) (30 180) (15 7) (15 182) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (DESCRIPTION WHEAT)
Positive training set is:- CT5-26 CT4-27 CT5-28 CT5-59 CT4-32 CT4-51 CT5-42
Negative training set is:- CT11-82 CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT8-56 CT11-58 CT2-50 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (30 275) (50 45)) T) score 0.6129 at wavelength 0.68
((GREATER-THAN (30 275) (45 85)) T) score 0.5346 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) NIL) score 0.5207 at wavelength 0.68
((GREATER-THAN (45 270) (45 85)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (45 270) (60 355)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (60 275) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (30 93) (50 45)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (50 227) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (35 48) (0 0)) NIL) score 0.4839 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((FIRST-MAX (60 180)) NIL) 0.68)) T) with score 0.7742
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((SECOND-MAX (75 180)) NIL) 0.68)) T) with score 0.7742
Score for unknown sample for class (DESCRIPTION WHEAT) is -1.0000
Cover types CT5-26 CT4-27 CT5-28 CT5-59 CT4-32 CT4-51 CT5-42 were correctly classified as belonging to this class.
Cover types CT7-23 CT2-28 CT3-27 CT3-35 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8400

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT2-28 CT2-30 CT3-27 CT3-63 CT3-35 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45
Negative training set is:- CT1-30 CT4-27 CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT4-32 CT4-51 CT1-45 CT5-42 CT6-45 CT7-41 CT9-46 CT10-49

Best single hypothesis scores are:-
((GREATER-THAN (60 275) (60 87)) NIL) score 0.6250 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.5000 at wavelength 0.68
((GREATER-THAN (60 275) (45 85)) NIL) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (75 2)) T) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (65 40)) T) score 0.4625 at wavelength 0.68
((GREATER-THAN (45 270) (45 356)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (75 2) (75 180)) NIL) score 0.4375 at wavelength 0.68
((GREATER-THAN (15 270) (60 355)) T) score 0.4250 at wavelength 0.68
((GREATER-THAN (45 85) (65 40)) T) score 0.4250 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 87) (60 355)) T) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
(((((GREATER-THAN (60 275) (60 87)) NIL) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000
Cover types CT2-30 CT3-63 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8400

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 2) (60 180)) T) score 0.7333 at wavelength 0.68
((GREATER-THAN (60 275) (50 227)) T) score 0.7000 at wavelength 0.68
((GREATER-THAN (30 93) (60 355)) NIL) score 0.6667 at wavelength 0.68
((GREATER-THAN (75 2) (45 178)) T) score 0.6667 at wavelength 0.68
((GREATER-THAN (60 275) (60 87)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (60 87) (30 93)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (45 85) (60 355)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (65 225) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (50 227) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (75 2) (30 180)) T) score 0.6333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 275) (60 87)) T) 0.68) (((GREATER-THAN (45 270) (0 0)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (75 2) (60 180)) T) 0.68) (((GREATER-THAN (15 225) (15 182)) NIL) 0.68)) T) with score 0.9000
Score for unknown sample for class (DESCRIPTION FOREST) is 0.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample
Cover types CT6-79 CT7-74 CT6-25 CT6-63 CT7-59 CT6-45 CT7-41 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8400

test4-run1-trace

Problem TRAINING.PROB.441 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) T) 0.91)) Overall score 0.7778 Positive 0.8889 Negative 0.1111
((((FIRST-MIN (60 315)) NIL) 0.91)) Overall score 0.7778 Positive 1.0000 Negative 0.2222
((((GREATER-THAN (60 315) (15 315)) T) 0.91)) Overall score 0.5556 Positive 0.6667 Negative 0.1111
((((GREATER-THAN (60 315) (0 0)) T) 0.91)) Overall score 0.5556 Positive 0.5556 Negative 0.0000

Problem TRAINING.PROB.440 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) NIL) 0.91)) Overall score 0.7778 Positive 0.8889 Negative 0.1111
((((FIRST-MIN (60 315)) T) 0.91)) Overall score 0.7778 Positive 0.7778 Negative 0.0000
((((GREATER-THAN (60 315) (15 315)) NIL) 0.91)) Overall score 0.5556 Positive 0.8889 Negative 0.3333
((((GREATER-THAN (60 315) (0 0)) NIL) 0.91)) Overall score 0.5556 Positive 1.0000 Negative 0.4444
Learning completed 1992/9/22 at time 13.11.13

test4-run1-option3

Solar Zenith Angle:- 35
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28
Negative training set is:- CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) T) score 1.0000 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) T) score 0.7778 at wavelength 0.91
((FIRST-MIN (60 315)) NIL) score 0.7778 at wavelength 0.91
((GREATER-THAN (60 315) (15 315)) T) score 0.5556 at wavelength 0.91
((GREATER-THAN (60 315) (0 0)) T) score 0.5556 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) with score 1.0000
Cover types CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32
Negative training set is:- CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) NIL) score 1.0000 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) NIL) score 0.7778 at wavelength 0.91
((FIRST-MIN (60 315)) T) score 0.7778 at wavelength 0.91
((GREATER-THAN (60 315) (15 315)) NIL) score 0.5556 at wavelength 0.91
((GREATER-THAN (60 315) (0 0)) NIL) score 0.5556 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) with score 1.0000
Cover types CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 1.0000

test4-run2-trace

Problem TRAINING.PROB.515 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.6364 Positive 1.0000 Negative 0.3636
((((GREATER-THAN (60 315) (30 315)) T) 0.91)) Overall score 0.5864 Positive 0.9500 Negative 0.3636
((((SECOND-MAX (15 135)) NIL) 0.91)) Overall score 0.4955 Positive 0.9500 Negative 0.4545
((((FIRST-MAX (30 135)) NIL) 0.91)) Overall score 0.4682 Positive 0.6500 Negative 0.1818
((((FIRST-MIN (60 315)) NIL) 0.91)) Overall score 0.4545 Positive 1.0000 Negative 0.5455

Problem TRAINING.PROB.514 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 0.6364 Positive 0.6364 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) NIL) 0.91)) Overall score 0.5864 Positive 0.6364 Negative 0.0500
((((SECOND-MAX (15 135)) T) 0.91)) Overall score 0.4955 Positive 0.5455 Negative 0.0500
((((FIRST-MAX (30 135)) T) 0.91)) Overall score 0.4682 Positive 0.8182 Negative 0.3500
((((FIRST-MIN (60 315)) T) 0.91)) Overall score 0.4545 Positive 0.4545 Negative 0.0000

Problem TRAINING.PROB.515 Level 2
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7182 Positive 0.9000 Negative 0.1818

Problem TRAINING.PROB.514 Level 2
Best scores
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) Overall score 0.6364 Positive 0.6364 Negative 0.0000

SYSTEMS INC

Problem TRAINING.PROB.515 Level 3
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((GREATER-THAN (15 135) (30 135)) NIL) 0.91)) T) Overall score 0.7591 Positive 0.8500 Negative 0.0909
Learning completed 1992/9/22 at time 13.15.12

test4-run2-option3

Solar Zenith Angle:- 70
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-28 CT10-28 CT7-41 CT5-42 CT6-45 CT8-42 CT11-45 CT10-49 CT5-59 CT6-79 CT7-59 CT8-56 CT11-82 CT11-58 CT6-63 CT7-74 CT8-70 CT10-76 CT10-63 CT11-71
Negative training set is:- CT1-30 CT2-28 CT2-30 CT4-32 CT3-35 CT1-45 CT9-46 CT2-50 CT4-51 CT3-63 CT9-68
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) T) score 0.6364 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) T) score 0.5864 at wavelength 0.91
((SECOND-MAX (15 135)) NIL) score 0.4955 at wavelength 0.91
((FIRST-MAX (30 135)) NIL) score 0.4682 at wavelength 0.91
((FIRST-MIN (60 315)) NIL) score 0.4545 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((GREATER-THAN (15 135) (30 135)) NIL) 0.91)) T) with score 0.7591
Cover types CT5-28 CT10-28 CT7-41 CT5-42 CT6-45 CT8-42 CT11-45 CT10-49 CT5-59 CT6-79 CT7-59 CT8-56 CT11-82 CT11-58 CT6-63 CT7-74 CT8-70 CT10-76 CT10-63 CT11-71 were correctly classified as belonging to this class.
Cover types CT9-46 CT4-51 CT3-63 CT9-68 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8710

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT4-32 CT3-35 CT1-45 CT9-46 CT2-50 CT4-51 CT3-63 CT9-68
Negative training set is:- CT5-28 CT10-28 CT7-41 CT5-42 CT6-45 CT8-42 CT11-45 CT10-49 CT5-59 CT6-79 CT7-59 CT8-56 CT11-82 CT11-58 CT6-63 CT7-74 CT8-70 CT10-76 CT10-63 CT11-71
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) NIL) score 0.6364 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) NIL) score 0.5864 at wavelength 0.91
((SECOND-MAX (15 135)) T) score 0.4955 at wavelength 0.91
((FIRST-MAX (30 135)) T) score 0.4682 at wavelength 0.91
((FIRST-MIN (60 315)) T) score 0.4545 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) with score 0.6364
Cover types CT1-30 CT2-28 CT2-30 CT4-32 CT3-35 CT1-45 CT2-50 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8710

test4-run3-trace

Problem TRAINING.PROB.642 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.5789 Positive 1.0000 Negative 0.4211
((((FIRST-MIN (60 315)) NIL) 0.91)) Overall score 0.3684 Positive 1.0000 Negative 0.6316
((((FIRST-MIN (45 315)) T) 0.91)) Overall score 0.3333 Positive 0.3333 Negative 0.0000
((((GREATER-THAN (15 315) (0 0)) T) 0.91)) Overall score 0.2865 Positive 0.4444 Negative 0.1579
((((SECOND-MIN (30 315)) T) 0.91)) Overall score 0.2865 Positive 0.4444 Negative 0.1579

Problem TRAINING.PROB.641 Level 1
Best scores
((((GREATER-THAN (45 315) (0 0)) T) 0.91)) Overall score 0.6333 Positive 0.8333 Negative 0.2000
((((GREATER-THAN (45 315) (30 315)) T) 0.91)) Overall score 0.5600 Positive 1.0000 Negative 0.4400
((((GREATER-THAN (45 315) (15 135)) T) 0.91)) Overall score 0.5533 Positive 0.8333 Negative 0.2800
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.4400 Positive 1.0000 Negative 0.5600
((((GREATER-THAN (60 315) (30 315)) T) 0.91)) Overall score 0.4400 Positive 1.0000 Negative 0.5600

Problem TRAINING.PROB.640 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) NIL) 0.91)) Overall score 0.7778 Positive 0.8889 Negative 0.1111
((((FIRST-MIN (60 315)) T) 0.91)) Overall score 0.7778 Positive 0.7778 Negative 0.0000
((((GREATER-THAN (60 315) (15 315)) NIL) 0.91)) Overall score 0.5556 Positive 0.8889 Negative 0.3333
((((GREATER-THAN (60 315) (0 0)) NIL) 0.91)) Overall score 0.5556 Positive 1.0000 Negative 0.4444

Problem TRAINING.PROB.642 Level 2
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) Overall score 0.7368 Positive 1.0000 Negative 0.2632
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7368 Positive 1.0000 Negative 0.2632

Problem TRAINING.PROB.641 Level 2
Best scores
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) Overall score 0.7133 Positive 0.8333 Negative 0.1200

Problem TRAINING.PROB.642 Level 3
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91) (((SECOND-MIN (15 135)) NIL) 0.91)) T) Overall score 0.8421 Positive 1.0000 Negative 0.1579

(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) Overall score 0.8421 Positive 1.0000 Negative 0.1579

Problem TRAINING.PROB.641 Level 3
Best scores
(((((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (0 0)) T) 0.91) (((GREATER-THAN (15 315) (30 135)) NIL) 0.91)) T) Overall score 0.7533 Positive 0.8333 Negative 0.0800
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (0 0)) NIL) 0.91)) T) Overall score 0.7533 Positive 0.8333 Negative 0.0800
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) Overall score 0.7533 Positive 0.8333 Negative 0.0800

Problem TRAINING.PROB.642 Level 4
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91) (((SECOND-MIN (15 135)) NIL) 0.91)) T) Overall score 0.8947 Positive 1.0000 Negative 0.1053
Learning completed 1992/9/22 at time 13.21.43

test4-run3-option3

Solar Zenith Angle:- 35
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT7-59 CT8-56 CT7-23 CT10-49 CT6-25 CT6-45 CT7-41 CT8-42 CT10-28
Negative training set is:- CT5-59 CT11-58 CT2-50 CT4-51 CT9-46 CT9-23 CT1-26 CT1-45 CT3-27 CT4-27 CT5-26 CT11-45 CT1-30 CT2-28 CT2-30 CT5-28 CT5-42 CT3-35 CT4-32
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) T) score 0.5789 at wavelength 0.91
((FIRST-MIN (60 315)) NIL) score 0.3684 at wavelength 0.91
((FIRST-MIN (45 315)) T) score 0.3333 at wavelength 0.91
((GREATER-THAN (15 315) (0 0)) T) score 0.2865 at wavelength 0.91
((SECOND-MIN (30 315)) T) score 0.2865 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91) (((SECOND-MIN (15 135)) NIL) 0.91)) T) with score 0.8947
Cover types CT7-59 CT8-56 CT7-23 CT10-49 CT6-25 CT6-45 CT7-41 CT8-42 CT10-28 were correctly classified as belonging to this class.
Cover types CT5-59 CT11-58 CT11-45 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8750

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT5-59 CT11-58 CT5-26 CT11-45 CT5-28 CT5-42
Negative training set is:- CT3-63 CT6-63 CT10-63 CT7-59 CT8-56 CT2-50 CT4-51 CT7-23 CT9-46 CT9-23 CT10-49 CT1-26 CT1-45 CT3-27 CT4-27 CT6-25 CT6-45 CT1-30 CT2-28 CT2-30 CT7-41 CT8-42 CT10-28 CT3-35 CT4-32

Best single hypothesis scores are:-
((GREATER-THAN (45 315) (0 0)) T) score 0.6333 at wavelength 0.91
((GREATER-THAN (45 315) (30 315)) T) score 0.5600 at wavelength 0.91
((GREATER-THAN (45 315) (15 135)) T) score 0.5533 at wavelength 0.91
((GREATER-THAN (60 315) (45 315)) T) score 0.4400 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) T) score 0.4400 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (0 0)) T) 0.91) (((GREATER-THAN (15 315) (30 135)) NIL) 0.91)) T) with score 0.7533
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (0 0)) NIL) 0.91)) T) with score 0.7533 (((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) with score 0.7533
Cover types CT5-26 CT5-28 CT5-42 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8750

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32
Negative training set is:- CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) NIL) score 1.0000 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) NIL) score 0.7778 at wavelength 0.91
((FIRST-MIN (60 315)) T) score 0.7778 at wavelength 0.91
((GREATER-THAN (60 315) (15 315)) NIL) score 0.5556 at wavelength 0.91
((GREATER-THAN (60 315) (0 0)) NIL) score 0.5556 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) with score 1.0000
Cover types CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8750

test4-run4-trace

Problem TRAINING.PROB.952 Level 1
Best scores
((((FIRST-MIN (45 315)) T) 0.91)) Overall score 0.4667 Positive 0.5000 Negative 0.0333
((((GREATER-THAN (45 315) (15 135)) NIL) 0.91)) Overall score 0.3750 Positive 0.8750 Negative 0.5000
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (45 315) (0 0)) NIL) 0.91)) Overall score 0.3417 Positive 0.8750 Negative 0.5333
((((FIRST-MIN (15 315)) T) 0.91)) Overall score 0.3250 Positive 0.6250 Negative 0.3000

Problem TRAINING.PROB.951 Level 1
Best scores
((((FIRST-MIN (45 315)) NIL) 0.91)) Overall score 0.4667 Positive 0.9667 Negative 0.5000
((((GREATER-THAN (45 315) (15 135)) T) 0.91)) Overall score 0.3750 Positive 0.5000 Negative 0.1250

((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 0.3667 Positive 0.3667 Negative 0.0000
((((GREATER-THAN (45 315) (0 0)) T) 0.91)) Overall score 0.3417 Positive 0.4667 Negative 0.1250
((((FIRST-MIN (15 315)) NIL) 0.91)) Overall score 0.3250 Positive 0.7000 Negative 0.3750

Problem TRAINING.PROB.952 Level 2
Best scores
(((((GREATER-THAN (45 315) (15 135)) NIL) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) Overall score 0.6750 Positive 0.8750 Negative 0.2000

Problem TRAINING.PROB.951 Level 2
Best scores
(((((FIRST-MIN (45 315)) NIL) 0.91) (((FIRST-MIN (15 315)) NIL) 0.91)) T) Overall score 0.6667 Positive 0.6667 Negative 0.0000

Problem TRAINING.PROB.952 Level 3
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) Overall score 0.7417 Positive 0.8750 Negative 0.1333
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7417 Positive 0.8750 Negative 0.1333
(((((GREATER-THAN (45 315) (15 135)) NIL) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7417 Positive 0.8750 Negative 0.1333

Problem TRAINING.PROB.951 Level 3
Best scores
(((((FIRST-MIN (45 315)) NIL) 0.91) (((FIRST-MIN (15 315)) NIL) 0.91)) T) Overall score 0.6667 Positive 0.6667 Negative 0.0000
Learning completed 1992/9/22 at time 14.18.15

test4-run4-option3

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (HEIGHT.CM (1000 3000))
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((FIRST-MIN (45 315)) T) score 0.4667 at wavelength 0.91
((GREATER-THAN (45 315) (15 135)) NIL) score 0.3750 at wavelength 0.91
((GREATER-THAN (60 315) (45 315)) T) score 0.3667 at wavelength 0.91
((GREATER-THAN (45 315) (0 0)) NIL) score 0.3417 at wavelength 0.91
((FIRST-MIN (15 315)) T) score 0.3250 at wavelength 0.91

Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) with score 0.7417
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) with score 0.7417
(((((GREATER-THAN (45 315) (15 135)) NIL) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) with score 0.7417
Cover types CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41 were correctly classified as belonging to this class.
Cover types CT8-70 CT9-68 CT11-71 CT5-26 CT10-63 CT10-28 CT3-35 CT11-58 CT4-51 CT5-42 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7368

Class Definition:- (HEIGHT.CM (0 1000))
Positive training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Negative training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Best single hypothesis scores are:-
((FIRST-MIN (45 315)) NIL) score 0.4667 at wavelength 0.91
((GREATER-THAN (45 315) (15 135)) T) score 0.3750 at wavelength 0.91
((GREATER-THAN (60 315) (45 315)) NIL) score 0.3667 at wavelength 0.91
((GREATER-THAN (45 315) (0 0)) T) score 0.3417 at wavelength 0.91
((FIRST-MIN (15 315)) NIL) score 0.3250 at wavelength 0.91
Best compound hypothesis scores are:-
(((((FIRST-MIN (45 315)) NIL) 0.91) (((FIRST-MIN (15 315)) NIL) 0.91)) T) with score 0.6667
Cover types CT11-82 CT10-76 CT1-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT4-32 CT8-56 CT2-50 CT1-45 CT8-42 CT9-46 CT10-49 CT11-45 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7368

test4-run5-trace

Problem TRAINING.PROB.1106 Level 1
Best scores
((((GREATER-THAN (30 45) (60 45)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((FIRST-MIN (60 45)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((SECOND-MIN (60 45)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((SECOND-MIN (60 315)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((GREATER-THAN (30 315) (60 315)) NIL) 0.68)) Overall score 0.8182 Positive 0.8182 Negative 0.0000

Problem TRAINING.PROB.1105 Level 1
Best scores
((((GREATER-THAN (30 45) (60 45)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909
((((FIRST-MIN (60 45)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909
((((SECOND-MIN (60 45)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909
((((SECOND-MIN (60 315)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909

((((GREATER-THAN (30 315) (60 315)) T) 0.68)) Overall score 0.8182 Positive 1.0000 Negative 0.1818

Problem TRAINING.PROB.1106 Level 2
Best scores
(((((SECOND-MIN (60 315)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000 (((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000

Problem TRAINING.PROB.1105 Level 2
Best scores
(((((SECOND-MIN (30 45)) NIL) 0.68) (((GREATER-THAN (30 225) (60 315)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((SECOND-MIN (30 45)) NIL) 0.68) (((SECOND-MIN (30 315)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((GREATER-THAN (60 135) (60 225)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((SECOND-MIN (30 45)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

Problem TRAINING.PROB.1106 Level 3
Best scores
(((((SECOND-MIN (60 315)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000

(((((SECOND-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000

Problem TRAINING.PROB.1105 Level 3
Best scores
(((((SECOND-MIN (30 45)) NIL) 0.68) (((GREATER-THAN (30 225) (60 315)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((SECOND-MIN (30 45)) NIL) 0.68) (((SECOND-MIN (30 315)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((GREATER-THAN (60 135) (60 225)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((SECOND-MIN (30 45)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
Learning completed 1992/9/22 at time 14.36.6

test4-run5-option3

Solar Zenith Angle:- 40
View Angle Data:-
At wavelength 0.68 data is ((0 0) (30 45) (60 45) (30 135) (60 135) (30 225) (60 225) (30 315) (60 315))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Negative training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Best single hypothesis scores are:-
((GREATER-THAN (30 45) (60 45)) NIL) score 0.8258 at wavelength 0.68
((FIRST-MIN (60 45)) NIL) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 45)) NIL) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 315)) NIL) score 0.8258 at wavelength 0.68
((GREATER-THAN (30 315) (60 315)) NIL) score 0.8182 at wavelength 0.68
Best compound hypothesis scores are:-
(((((SECOND-MIN (60 315)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 315)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 315)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091

(((((SECOND-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091
(((((FIRST-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((FIRST-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((FIRST-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091
Cover types CT5-26 CT6-25 CT8-56 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.9565

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Negative training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Best single hypothesis scores are:-
((GREATER-THAN (30 45) (60 45)) T) score 0.8258 at wavelength 0.68
((FIRST-MIN (60 45)) T) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 45)) T) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 315)) T) score 0.8258 at wavelength 0.68
((GREATER-THAN (30 315) (60 315)) T) score 0.8182 at wavelength 0.68
Best compound hypothesis scores are:-
(((((SECOND-MIN (30 45)) NIL) 0.68) (((GREATER-THAN (30 225) (60 315)) T) 0.68)) T) with score 0.9091
(((((SECOND-MIN (30 45)) NIL) 0.68) (((SECOND-MIN (30 315)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((GREATER-THAN (60 135) (60 225)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((SECOND-MIN (30 45)) NIL) 0.68)) T) with score 0.9091
Cover types CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46 were correctly classified as belonging to this class.
Cover types CT5-28 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.9565

test4-run6-trace

Problem TRAINING.PROB.1200 Level 1
Best scores
((((GREATER-THAN (0 0) (10 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((GREATER-THAN (0 0) (15 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((GREATER-THAN (10 0) (15 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((FIRST-MAX (0 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((FIRST-MIN (0 0)) T) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000

Problem TRAINING.PROB.1199 Level 1
Best scores
((((GREATER-THAN (0 0) (10 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((GREATER-THAN (0 0) (15 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((GREATER-THAN (10 0) (15 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((FIRST-MAX (0 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((FIRST-MIN (0 0)) NIL) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182

Problem TRAINING.PROB.1200 Level 2
Best scores
(((((FIRST-MIN (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((FIRST-MAX (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((FIRST-MIN (0 0)) T) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((FIRST-MAX (0 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((GREATER-THAN (10 0) (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((GREATER-THAN (0 0) (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((GREATER-THAN (0 0) (10 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000

Problem TRAINING.PROB.1199 Level 2
Best scores
(((((FIRST-MIN (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((FIRST-MAX (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((FIRST-MIN (0 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((FIRST-MAX (0 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((GREATER-THAN (10 0) (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((GREATER-THAN (0 0) (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((GREATER-THAN (0 0) (10 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
Learning completed 1992/9/22 at time 14.38.10

test4-run6-opt3

Solar Zenith Angle:- 40
View Angle Data:-
At wavelength 0.68 data is ((0 0) (10 0) (15 0))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Negative training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Best single hypothesis scores are:-
((GREATER-THAN (0 0) (10 0)) NIL) score 0.1818 at wavelength 0.68
((GREATER-THAN (0 0) (15 0)) NIL) score 0.1818 at wavelength 0.68
((GREATER-THAN (10 0) (15 0)) NIL) score 0.1818 at wavelength 0.68
((FIRST-MAX (0 0)) NIL) score 0.1818 at wavelength 0.68
((FIRST-MIN (0 0)) T) score 0.1818 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (15 0)) NIL) 0.68)) T) with score 0.1818
(((((FIRST-MAX (15 0)) T) 0.68)) T) with score 0.1818
(((((FIRST-MIN (0 0)) T) 0.68)) T) with score 0.1818
(((((FIRST-MAX (0 0)) NIL) 0.68)) T) with score 0.1818
(((((GREATER-THAN (10 0) (15 0)) NIL) 0.68)) T) with score 0.1818
(((((GREATER-THAN (0 0) (15 0)) NIL) 0.68)) T) with score 0.1818
(((((GREATER-THAN (0 0) (10 0)) NIL) 0.68)) T) with score 0.1818
Cover types CT10-49 CT8-42 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.6087

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Negative training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Best single hypothesis scores are:-
((GREATER-THAN (0 0) (10 0)) T) score 0.1818 at wavelength 0.68
((GREATER-THAN (0 0) (15 0)) T) score 0.1818 at wavelength 0.68
((GREATER-THAN (10 0) (15 0)) T) score 0.1818 at wavelength 0.68
((FIRST-MAX (0 0)) T) score 0.1818 at wavelength 0.68
((FIRST-MIN (0 0)) NIL) score 0.1818 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (15 0)) T) 0.68)) T) with score 0.1818
(((((FIRST-MAX (15 0)) NIL) 0.68)) T) with score 0.1818
(((((FIRST-MIN (0 0)) NIL) 0.68)) T) with score 0.1818
(((((FIRST-MAX (0 0)) T) 0.68)) T) with score 0.1818 (((((GREATER-THAN (0 0) (15 0)) T) 0.68)) T) with score 0.1818
(((((GREATER-THAN (0 0) (10 0)) T) 0.68)) T) with score 0.1818
Cover types CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46 were correctly classified as belonging to this class.
Cover types CT5-26 CT6-25 CT8-56 CT5-28 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.6087

test5-run1-trace

Problem TRAINING.PROB.278 Level 1
Best scores
((((GREATER-THAN (60 355) (45 225)) T) 0.68)) Overall score 0.8182 Positive 1.0000 Negative 0.1818
((((GREATER-THAN (40 5) (0 0)) T) 0.68)) Overall score 0.8182 Positive 1.0000 Negative 0.1818
((((GREATER-THAN (40 85) (60 355)) NIL) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (20 92) (45 43)) NIL) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (60 355) (60 48)) T) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (60 355) (15 222)) T) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (60 48) (45 225)) T) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (65 90) (60 355)) NIL) 0.68)) Overall score 0.7182 Positive 0.9000 Negative 0.1818
((((GREATER-THAN (60 355) (60 230)) T) 0.68)) Overall score 0.7091 Positive 0.8000 Negative 0.0909
((((GREATER-THAN (20 92) (60 355)) NIL) 0.68)) Overall score 0.6364 Positive 1.0000 Negative 0.3636

Problem TRAINING.PROB.277 Level 1
Best scores
((((SECOND-MAX (60 230)) T) 0.68)) Overall score 0.4917 Positive 0.6250 Negative 0.1333
((((FIRST-MAX (75 225)) T) 0.68)) Overall score 0.4750 Positive 0.8750 Negative 0.4000
((((GREATER-THAN (65 90) (60 355)) T) 0.68)) Overall score 0.4667 Positive 1.0000 Negative 0.5333
((((FIRST-MIN (40 5)) T) 0.68)) Overall score 0.4333 Positive 0.5000 Negative 0.0667
((((GREATER-THAN (40 5) (60 48)) NIL) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (75 225) (45 225)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (60 48) (45 43)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (60 355) (45 225)) NIL) 0.68)) Overall score 0.3417 Positive 0.8750 Negative 0.5333
((((GREATER-THAN (60 48) (45 225)) NIL) 0.68)) Overall score 0.3417 Positive 0.8750 Negative 0.5333
((((GREATER-THAN (65 90) (40 85)) T) 0.68)) Overall score 0.3333 Positive 1.0000 Negative 0.6667

Problem TRAINING.PROB.276 Level 1
Best scores
((((GREATER-THAN (65 90) (30 220)) NIL) 0.68)) Overall score 0.8444 Positive 0.9000 Negative 0.0556
((((GREATER-THAN (65 90) (45 225)) NIL) 0.68)) Overall score 0.8333 Positive 1.0000 Negative 0.1667
((((GREATER-THAN (60 355) (15 42)) NIL) 0.68))Overall score 0.7889 Positive 0.9000 Negative 0.1111

((((GREATER-THAN (60 48) (30 220)) NIL) 0.68)) Overall score 0.7889 Positive 0.9000 Negative 0.1111
((((GREATER-THAN (20 92) (60 48)) T) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 355) (40 5)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 355) (20 0)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 355) (45 43)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 48) (0 0)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 230) (45 225)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556

Problem TRAINING.PROB.278 Level 2
Best scores
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

Problem TRAINING.PROB.277 Level 2
Best scores
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68)) T) Overall score 0.8000 Positive 1.0000 Negative 0.2000

Problem TRAINING.PROB.276 Level 2
Best scores
(((((GREATER-THAN (60 48) (30 220)) NIL) 0.68) (((FIRST-MIN (40 5)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((SECOND-MIN (40 85)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MIN (20 92)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MAX (60 355)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (40 5)) T) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 230)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (20 92) (40 5)) T) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (0 0)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (45 225)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (15 222)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 48)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 222)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (30 220)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (60 355)) T) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 48) (30 220)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 42)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000

Problem TRAINING.PROB.278 Level 3
Best scores
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

Problem TRAINING.PROB.277 Level 3
Best scores
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000

Problem TRAINING.PROB.277 Level 4
Best scores
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68) (((FIRST-MAX (45 225)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((FIRST-MIN (30 45)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 92) (20 0)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (75 225)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 0) (15 222)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667

Learning completed 1992/9/18 at time 4.2.45

test5-run1-option2

Solar Zenith Angle:- 71
View Angle Data:-
At wavelength 0.92 data is ((0 0))
At wavelength 0.68 data is ((65 90) (40 85) (20 92) (60 355) (40 5) (20 0) (75 225) (60 48) (60 230) (45 43) (45 225) (30 45) (30 220) (15 42) (15 222) (0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT10-49 CT6-45 CT8-56 CT10-63 CT7-59 CT6-79 CT6-63 CT10-76 CT8-70 CT7-74
Negative training set is:- CT11-45 CT9-46 CT1-45 CT4-51 CT2-50 CT11-58 CT11-82 CT5-59 CT3-63 CT11-71 CT9-68

Best single hypothesis scores are:-
((GREATER-THAN (60 355) (45 225)) T) score 0.8182 at wavelength 0.68
((GREATER-THAN (40 5) (0 0)) T) score 0.8182 at wavelength 0.68
((GREATER-THAN (40 85) (60 355)) NIL) score 0.7273 at wavelength 0.68
((GREATER-THAN (20 92) (45 43)) NIL) score 0.7273 at wavelength 0.68
((GREATER-THAN (60 355) (60 48)) T) score 0.7273 at wavelength 0.68
((GREATER-THAN (60 355) (15 222)) T) score 0.7273 at wavelength 0.68
((GREATER-THAN (60 48) (45 225)) T) score 0.7273 at wavelength 0.68
((GREATER-THAN (65 90) (60 355)) NIL) score 0.7182 at wavelength 0.68
((GREATER-THAN (60 355) (60 230)) T) score 0.7091 at wavelength 0.68
((GREATER-THAN (20 92) (60 355)) NIL) score 0.6364 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.6 8)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091

(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) with score 0.9091
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 0.5000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT5-28 CT5-26 CT5-42 CT11-45 CT11-58 CT11-82 CT5-59 CT11-71
Negative training set is:- CT10-28 CT9-23 CT7-23 CT6-25 CT4-27 CT3-27 CT2-28 CT 1-26 CT4-32 CT3-35 CT2-30 CT1-30 CT8-42 CT7-41 CT10-49 CT9-46 CT6-45 CT1-45 CT8 -56 CT4-51 CT2-50 CT10-63 CT7-59 CT6-79 CT6-63 CT3-63 CT10-76 CT9-68 CT8-70 CT7 -74
Best single hypothesis scores are:-
((SECOND-MAX (60 230)) T) score 0.4917 at wavelength 0.68
((FIRST-MAX (75 225)) T) score 0.4750 at wavelength 0.68
((GREATER-THAN (65 90) (60 355)) T) score 0.4667 at wavelength 0.68
((FIRST-MIN (40 5)) T) score 0.4333 at wavelength 0.68
((GREATER-THAN (40 5) (60 48)) NIL) score 0.3667 at wavelength 0.68
((GREATER-THAN (75 225) (45 225)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (60 48) (45 43)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (60 355) (45 225)) NIL) score 0.3417 at wavelength 0.68
((GREATER-THAN (60 48) (45 225)) NIL) score 0.3417 at wavelength 0.68
((GREATER-THAN (65 90) (40 85)) T) score 0.3333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68) (((FIRST-MAX (45 225)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((FIRST-MIN (30 45)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T)
with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 92) (20 0)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (75 225)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 0) (15 222)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333

(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is -1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT4-32 CT3-35 CT2-30 CT1-30 CT9-46 CT1-45 CT4-51 CT2-50 CT3-63 CT9-68
Negative training set is:- CT8-42 CT7-41 CT5-42 CT11-45 CT10-49 CT6-45 CT8-56 CT11-58 CT11-82 CT10-63 CT7-59 CT6-79 CT6-63 CT5-59 CT11-71 CT10-76 CT8-70 CT7-74
Best single hypothesis scores are:-
((GREATER-THAN (65 90) (30 220)) NIL) score 0.8444 at wavelength 0.68
((GREATER-THAN (65 90) (45 225)) NIL) score 0.8333 at wavelength 0.68
((GREATER-THAN (60 355) (15 42)) NIL) score 0.7889 at wavelength 0.68
((GREATER-THAN (60 48) (30 220)) NIL) score 0.7889 at wavelength 0.68
((GREATER-THAN (20 92) (60 48)) T) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 355) (40 5)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 355) (20 0)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 355) (45 43)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 48) (0 0)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 230) (45 225)) NIL) score 0.7444 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 48) (30 220)) NIL) 0.68) (((FIRST-MIN (40 5)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((SECOND-MIN (40 85)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MIN (20 92)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MAX (60 355)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (40 5)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 230)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (20 92) (40 5)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (0 0)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (45 225)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (15 222)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 48)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 222)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (30 220)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (60 355)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 48) (30 220)) NIL) 0.68)) T) with score 0.9000

(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 42)
) NIL) 0.68)) T) with score 0.9000
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -0.9333

test5-run1-option3

The system's classification performance score is 0.8929

test5-run2-trace

Problem TRAINING.PROB.5 Level 1
Best scores
((((GREATER-THAN (60 275) (50 227)) T) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (45 85) (60 355)) NIL) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (75 2) (45 178)) T) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (60 355) (15 182)) T) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (50 227) (60 355)) NIL) 0.68)) Overall score 0.8889 Positive 0.8889 Negative 0.0000
((((GREATER-THAN (60 275) (30 93)) T) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176
((((GREATER-THAN (60 275) (35 220)) T) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176
((((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176
((((GREATER-THAN (60 87) (30 93)) T) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176

Problem TRAINING.PROB.4 Level 1
Best scores
((((GREATER-THAN (60 275) (60 355)) T) 0.68)) Overall score 0.5000 Positive 1.0000 Negative 0.5000
((((GREATER-THAN (65 40) (45 356)) T) 0.68)) Overall score 0.4333 Positive 1.0000 Negative 0.5667
((((GREATER-THAN (65 40) (30 5)) T) 0.68)) Overall score 0.4333 Positive 1.0000 Negative 0.5667
((((GREATER-THAN (60 355) (45 356)) T) 0.68)) Overall score 0.4333 Positive 1.0000 Negative 0.5667
((((FIRST-MAX (75 180)) T) 0.68)) Overall score 0.4167 Positive 0.7500 Negative 0.3333
((((FIRST-MIN (45 356)) T) 0.68)) Overall score 0.3750 Positive 0.3750 Negative 0.0000
((((GREATER-THAN (75 2) (60 180)) NIL) 0.68)) Overall score 0.3750 Positive 0.8750 Negative 0.5000
((((SECOND-MAX (60 180)) T) 0.68)) Overall score 0.3667 Positive 0.5000 Negative 0.1333
((((GREATER-THAN (65 225) (50 227)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (65 225) (65 40)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333

Problem TRAINING.PROB.3 Level 1
Best scores
((((GREATER-THAN (30 275) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (65 40) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (50 45) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (15 46) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((FIRST-MIN (75 2)) T) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (60 275) (75 2)) T) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (65 40) (45 356)) NIL) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (65 40) (30 5)) NIL) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (35 48) (75 2)) T) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (75 2) (60 355)) NIL) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000

Problem TRAINING.PROB.5 Level 2
Best scores
(((((GREATER-THAN (60 355) (0 0)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) Overall score 1.0000 Positive 1.0000 Negative 0.0000
"
Total of 91 similar compound hypotheses each with discrimination score 1.0000
"
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) T) Overall score 1.0000 Positive 1.0000 Negative 0.0000

Problem TRAINING.PROB.4 Level 2
Best scores
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (60 355) (45 356)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (30 5)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (45 356)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000

Problem TRAINING.PROB.4 Level 3
Best scores
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (60 355) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) Overall score 0.9667 Positive 1.0000 Negative 0.0333
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (30 5)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) Overall score 0.9667 Positive 1.0000 Negative 0.0333
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) Overall score 0.9667 Positive 1.0000 Negative 0.0333
Learning completed 1992/9/21 at time 21.40.8

test5-run2-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.92 data is ((0 0))
At wavelength 0.68 data is ((60 275) (45 270) (30 275) (15 270) (60 87) (45 85) (30 93) (2 90) (65 225) (50 227) (35 220) (15 225) (65 40) (50 45) (35 48) (15 46) (2 45) (75 2) (75 180) (60 355) (60 180) (45 356) (45 178) (30 5) (30 180) (15 7) (15 182) (0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT10-28 CT10-63 CT7-59 CT6-63 CT8-56 CT10-49 CT8-42 CT7-41 CT6-45
Negative training set is:- CT5-59 CT5-28 CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT11-58 CT4-32 CT3-35 CT4-51 CT2-50 CT11-45 CT9-46 CT5-42 CT1-45
Best single hypothesis scores are:-
((GREATER-THAN (60 275) (50 227)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (45 85) (60 355)) NIL) score 0.9412 at wavelength 0.68
((GREATER-THAN (30 93) (60 355)) NIL) score 0.9412 at wavelength 0.68
((GREATER-THAN (75 2) (45 178)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (60 355) (15 182)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (50 227) (60 355)) NIL) score 0.8889 at wavelength 0.68
((GREATER-THAN (60 275) (30 93)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 275) (35 220)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (30 275) (50 45)) NIL) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 87) (30 93)) T) score 0.8824 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 355) (0 0)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (0 0)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (15 225) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (15 225) (60 355)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (65 40)) NIL) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (65 40)) NIL) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (50 227) (65 40)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (50 227) (65 40)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (65 225) (75 2)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (65 225) (75 2)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (2 90) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (2 90) (60 355)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000

(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((SECOND-MIN (45 356)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (45 356) (15 7)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (60 355) (0 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (15 225) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (50 227) (65 40)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (65 225) (75 2)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (2 90) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (0 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (2 45)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (2 90)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (15 225)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (45 85)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (45 356) (30 5)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (15 182)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (45 356) (15 7)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (75 2) (15 182)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (75 2) (30 180)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (15 46) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (50 45) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (50 227) (75 2)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (35 220)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (35 220)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (30 93)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000

(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((SECOND-MIN (45 356)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (45 356) (15 7)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (60 355) (0 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (15 225) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (35 220) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (35 220) (65 40)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (50 227) (65 40)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (65 225) (75 2)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (2 90) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (60 275) (35 220)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (60 355) (15 182)) T) 0.68)) T) with score 1.0000

(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (75 2) (45 178)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 1.0000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT11-82 CT11-71 CT5-26 CT5-59 CT5-28 CT11-58 CT11-45 CT5-42
Negative training set is:- CT6-79 CT10-76 CT7-74 CT9-68 CT8-70 CT9-23 CT7-23 CT6-25 CT1-26 CT10-28 CT10-63 CT7-59 CT6-63 CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT8-56 CT4-32 CT3-35 CT4-51 CT2-50 CT10-49 CT9-46 CT8-42 CT7-41 CT6-45 CT1-45
Best single hypothesis scores are:-
((GREATER-THAN (60 275) (60 355)) T) score 0.5000 at wavelength 0.68
((GREATER-THAN (65 40) (45 356)) T) score 0.4333 at wavelength 0.68
((GREATER-THAN (65 40) (30 5)) T) score 0.4333 at wavelength 0.68

((GREATER-THAN (60 355) (45 356)) T) score 0.4333 at wavelength 0.68
((FIRST-MAX (75 180)) T) score 0.4167 at wavelength 0.68
((FIRST-MIN (45 356)) T) score 0.3750 at wavelength 0.68
((GREATER-THAN (75 2) (60 180)) NIL) score 0.3750 at wavelength 0.68
((SECOND-MAX (60 180)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (65 225) (50 227)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (65 225) (65 40)) T) score 0.3667 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (60 355) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) with score 0.9667
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (30 5)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) with score 0.9667
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) with score 0.9667
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is 1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT4-32 CT3-35 CT4-51 CT2-50 CT9-46 CT1-45
Negative training set is:- CT10-28 CT10-63 CT7-59 CT6-63 CT5-59 CT5-28 CT11-58 CT8-56 CT11-45 CT10-49 CT8-42 CT7-41 CT6-45 CT5-42
Best single hypothesis scores are:-
((GREATER-THAN (30 275) (75 2)) T) score 1.0000 at wavelength 0.68
((GREATER-THAN (65 40) (75 2)) T) score 1.0000 at wavelength 0.68
((GREATER-THAN (50 45) (75 2)) T) score 1.0000 at wavelength 0.68
((GREATER-THAN (15 46) (75 2)) T) score 1.0000 at wavelength 0.68
((FIRST-MIN (75 2)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (60 275) (75 2)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (65 40) (45 356)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (65 40) (30 5)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (35 48) (75 2)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 2) (60 355)) NIL) score 0.9167 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (15 46) (75 2)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (50 45) (75 2)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (65 40) (75 2)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (75 2)) T) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -1.0000

test5-run2-option3

The system's classification performance score is 0.9655

test6-run1-auto

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((60 180) (60 0) (45 180) (45 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT2-28 CT2-30 CT3-27 CT3-63 CT3-35 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45
Negative training set is:- CT1-30 CT4-27 CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT4-32 CT4-51 CT1-45 CT5-42 CT6-45 CT7-41 CT9-46 CT10-49
Best single hypothesis scores are:-
((SECOND-MAX (45 180)) T) score 0.2625 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.2000 at wavelength 0.68
((SECOND-MAX (60 180)) NIL) score 0.2000 at wavelength 0.68
((GREATER-THAN (60 0) (45 180)) NIL) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) NIL) score 0.1875 at wavelength 0.68
((FIRST-MIN (60 0)) T) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 180) (60 0)) T) score 0.1250 at wavelength 0.68
((FIRST-MAX (60 0)) NIL) score 0.1250 at wavelength 0.68
((SECOND-MIN (45 180)) NIL) score 0.1250 at wavelength 0.68
((SECOND-MIN (45 0)) T) score 0.1250 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (60 0)) T) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MIN (60 0)) T) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000
Cover types CT2-28 CT2-30 CT3-63 CT2-50 CT11-45 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7222

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MIN (45 0)) NIL) score 0.4333 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) T) score 0.4000 at wavelength 0.68
((FIRST-MIN (60 0)) NIL) score 0.3667 at wavelength 0.68
((SECOND-MIN (60 0)) T) score 0.2500 at wavelength 0.68

((GREATER-THAN (60 180) (45 180)) T) score 0.2083 at wavelength 0.68
((FIRST-MAX (45 180)) NIL) score 0.2083 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.1833 at wavelength 0.68
((SECOND-MIN (45 180)) T) score 0.1833 at wavelength 0.68
((GREATER-THAN (45 180) (45 0)) T) score 0.1333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (60 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((FIRST-MIN (45 180)) NIL) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((SECOND-MAX (45 0)) NIL) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((FIRST-MIN (60 0)) NIL) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((SECOND-MIN (45 0)) NIL) 0.68)) T) with score 0.5000
Score for unknown sample for class (DESCRIPTION FOREST) is 1.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample
Cover types CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41 were correctly classified as belonging to this class.
Cover types CT3-27 CT3-35 CT8-56 CT11-58 CT8-42 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7222

test6-run1-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.92 data is ((0 0))
At wavelength 0.68 data is ((60 180) (60 0) (45 180) (45 0))

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT3-63 CT3-27 CT2-30 CT2-28 CT11-58 CT8-56 CT3-35 CT2-50 CT11-45 CT8-42
Negative training set is:- CT10-28 CT10-63 CT7-59 CT6-63 CT5-59 CT5-28 CT4-27 CT1-30 CT4-32 CT4-51 CT10-49 CT9-46 CT7-41 CT6-45 CT5-42 CT1-45
Best single hypothesis scores are:-
((SECOND-MAX (45 180)) T) score 0.2625 at wavelength 0.68
((SECOND-MAX (60 180)) NIL) score 0.2000 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.2000 at wavelength 0.68
((FIRST-MIN (60 0)) T) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 0) (45 180)) NIL) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) NIL) score 0.1875 at wavelength 0.68
((SECOND-MIN (45 180)) NIL) score 0.1250 at wavelength 0.68
((SECOND-MIN (45 0)) T) score 0.1250 at wavelength 0.68
((FIRST-MAX (60 0)) NIL) score 0.1250 at wavelength 0.68
((GREATER-THAN (60 180) (60 0)) T) score 0.1250 at wavelength 0.68

Best compound hypothesis scores are:-
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MIN (60 0)) T) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((FIRST-MIN (60 0)) T) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT7-23 CT6-25 CT7-59 CT6-63 CT7-41 CT6-45
Negative training set is:- CT11-82 CT10-76 CT11-71 CT9-68 CT8-70 CT9-23 CT5-26 CT1-26 CT10-28 CT10-63 CT5-59 CT5-28 CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT11-58 CT8-56 CT4-32 CT3-35 CT4-51 CT2-50 CT11-45 CT10-49 CT9-46 CT8-42 CT5-42 CT1-45
Best single hypothesis scores are:-
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MIN (45 0)) NIL) score 0.4333 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) T) score 0.4000 at wavelength 0.68
((FIRST-MIN (60 0)) NIL) score 0.3667 at wavelength 0.68
((SECOND-MIN (60 0)) T) score 0.2500 at wavelength 0.68
((FIRST-MAX (45 180)) NIL) score 0.2083 at wavelength 0.68
((GREATER-THAN (60 180) (45 180)) T) score 0.2083 at wavelength 0.68
((SECOND-MIN (45 180)) T) score 0.1833 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.1833 at wavelength 0.68
((GREATER-THAN (45 180) (45 0)) T) score 0.1333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (60 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((FIRST-MIN (45 180)) NIL) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((SECOND-MAX (45 0)) NIL) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((FIRST-MIN (60 0)) NIL) 0.68)) T) with score 0.5000

(((((FIRST-MIN (45 0)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((SECOND-MIN (45 0)) NIL) 0.68)) T) with score 0.5000
Score for unknown sample for class (DESCRIPTION FOREST) is 1.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample

test6-run1-option3

The system's classification performance score is 0.7222

test6-run2-auto

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((75 0) (75 180) (60 0) (60 180) (45 0) (45 180) (30 0) (30 180) (15 0) (15 180) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT6-45 CT7-41 CT8-42 CT10-49
Negative training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT3-35 CT4-32 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT9-46 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (45 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (75 0) (60 180)) T) score 0.8889 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (0 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (45 0) (30 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (30 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (60 0) (15 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (45 0) (15 0)) T) score 0.8235 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (45 0) (30 0)) T) 0.68) (((GREATER-THAN (60 0) (15 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (0 0)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (15 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 0) (45 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 1.0000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample
Cover types CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT6-45 CT7-41 CT8-42 CT10-49 were correctly classified as belonging to this class.
Cover types CT11-82 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8966

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT11-82 CT11-71 CT5-26 CT5-28 CT5-59 CT11-58 CT5-42 CT11-45

Negative training set is:- CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT2-50 CT4-51 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (60 180)) NIL) score 0.5000 at wavelength 0.68
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MAX (60 180)) T) score 0.4583 at wavelength 0.68
((FIRST-MAX (75 180)) T) score 0.4167 at wavelength 0.68
((GREATER-THAN (60 180) (45 180)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (75 0) (60 0)) T) score 0.3417 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 0) (30 0)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 180) (60 180)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (45 0) (0 0)) NIL) score 0.3083 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((FIRST-MIN (75 0)) NIL) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (30 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (45 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (60 0)) T) 0.68)) T) with score 0.8083
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is 1.0000
Cover types CT5-26 CT5-28 CT5-59 CT11-58 CT5-42 CT11-45 were correctly classified as belonging to this class.
Cover types CT9-46 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8966

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 CT9-46
Negative training set is:- CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT11-58 CT5-42 CT6-45 CT7-41 CT8-42 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (60 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (15 0)) NIL) score 0.9167 at wavelength 0.68
((FIRST-MIN (75 0)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (0 0)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 0) (45 180)) NIL) score 0.7143 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) NIL) score 0.7143 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (75 0)) T) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (15 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (30 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (45 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (60 0)) NIL) 0.68)) T) with score 0.9167
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -1.0000
Cover types CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 were correctly classified as belonging to this class.

Cover types CT11-71 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8966

test6-run2-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((75 0) (75 180) (60 0) (60 180) (45 0) (45 180) (30 0) (30 180) (15 0) (15 180) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT6-45 CT7-41 CT8-42 CT10-49
Negative training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT3-35 CT4-32 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT9-46 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (45 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (75 0) (60 180)) T) score 0.8889 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (0 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (45 0) (30 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (30 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (60 0) (15 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (45 0) (15 0)) T) score 0.8235 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (45 0) (30 0)) T) 0.68) (((GREATER-THAN (60 0) (15 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (0 0)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (15 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 0) (45 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 1.0000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT11-82 CT11-71 CT5-26 CT5-28 CT5-59 CT11-58 CT5-42 CT11-45
Negative training set is:- CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT2-50 CT4-51 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (60 180)) NIL) score 0.5000 at wavelength 0.68
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MAX (60 180)) T) score 0.4583 at wavelength 0.68
((FIRST-MAX (75 180)) T) score 0.4167 at wavelength 0.68
((GREATER-THAN (60 180) (45 180)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (75 0) (60 0)) T) score 0.3417 at wavelength 0.68
((FIRST-MIN (75 0)) NIL) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) T) score 0.3083 at wavelength 0.68

((GREATER-THAN (75 0) (30 0)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 180) (60 180)) T) score 0.3083 at wavelength 0.68

Best compound hypothesis scores are:-
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (30 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (45 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((FIRST-MIN (75 0)) NIL) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (60 0)) T) 0.68)) T) with score 0.8083
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is 1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 CT9-46
Negative training set is:- CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT11-58 CT5-42 CT6-45 CT7-41 CT8-42 CT10-49 CT11-45
Best single hypothesis scores are:-
((FIRST-MIN (75 0)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (60 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (15 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (0 0)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 0) (45 180)) NIL) score 0.7143 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) NIL) score 0.7143 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (75 0) (15 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (30 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (45 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (60 0)) NIL) 0.68)) T) with score 0.9167
(((((FIRST-MIN (75 0)) T) 0.68)) T) with score 0.9167
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -1.0000

test6-run2-option3

The system's classification performance score is 0.8966

APPENDIX C

LISTING OF FUNCTIONS REQUIRED BY THE NEW VERSION OF THE BROWSER

```
;;;-----------------------------------------------------------------------
;;; Functions Required by the Browser
;;;-----------------------------------------------------------------------

(defun show (unit kb-name)
"Draws a graph consisting of the unit and its immediate children."
  (let ((unit (unit unit (get.value 'browser 'current.kb)))
        (kb (kb kb-name)))
    (put.value 'browser 'message "")
    (put.value 'browser 'current.unit unit)
    (graph-unit unit
                :unit-predicate
                #'(lambda (subunits)
                    (cond ((eq subunits unit) unit)
                          ((and (or (member unit
                                          (unit.parents subunits 'subclass))
                                    (member unit
                                          (unit.parents subunits 'member)))
                                (eq (unit.kb subunits) kb))
                           unit)
                          (t nil))))))

(defun down (unit)
"Calls the function show to draw the graph consisting of the unit's child and
its children, if any.  Allows the user to select which child's tree is to be
displayed from a menu.  If the unit has no children, the function get-tree is
called so that the user can select one of the top level units in the kb."
  (let* ((kb (get.value 'browser 'current.kb))
         (children (get-children unit kb))
         (len (length children)))
    (cond ((= len 0)
           (get-tree kb))
          ((= len 1)
           (show (first children) kb))
          (t (let* ((menu
                     (make-cascading-menu :pop-up children))
                    (choice (choose-from-menu menu :position
                                  (make-position :x 800 :y 400))))
               (show choice kb))))))
```

```
(defun up (unit)
"Calls the function show to graph the subtree including the parent of the unit
and all the other units that have the same parent.  If the unit has more than
one parent, it allows the user to select the required parent from a menu."
 (let* ((kb (get.value 'browser 'current.kb))
        (parents (get-parents unit kb)))
   (cond ((= (length parents) 1) (show (first parents) kb))
         (t
          (let ((menu
                 (make-cascading-menu :pop-up
                                      (mapcar #'(lambda (unit)
                                                  (unit.name unit))
                                              (get-parents unit kb)))))
            (let ((choice (choose-from-menu menu :position
                                            (make-position :x 800 :y 400))))
              (if choice
                  (show choice kb))))))))

(defun get-parents (unit kb)
"Returns a list of the parents, both member and subclass, of the unit.  Adds to
the list the unit that is at the top of the tree containing the unit in the kb.
If the unit has no parents, a list of the top level units in the kb is
returned."
 (let ((lis ())
       (tree-top (unit (get.value 'browser 'tree.top))))
   (dolist (n (append (unit.parents unit 'subclass)
                      (unit.parents unit 'member))
            (cond ((member tree-top lis) lis)
                  ((null lis)(get-top-level-units kb))
                  (t (cons tree-top lis))))
     (if (and n (eq (unit.kb n) (kb kb)))
         (push n lis)))))

(defun get-children (unit kb)
"Returns the children, both member and subclass, of a unit."
 (let ((children (mapcar #'(lambda (unit) (unit.name unit))
                         (append (unit.children unit 'subclass)
                                 (unit.children unit 'member))))
       (result ()))
   (dolist (child children result)
     (when (eq (unit.kb child) (kb kb))
       (push child result)))))
```

```
(defun show-slots (thisunit)
"Allows the user to select a slot from a menu and then displays the value of
the slot."
  (let ((lis (set-difference
             (unit.slot.names thisunit)
             (unit.slot.names 'classes))))
    (cond ((null lis)
           (put.value 'browser 'message (format ()
                 "The unit ~S has no slots" (unit.name thisunit)))
           nil)
          (t (let* ((men (make-cascading-menu :pop-up lis))
                    (choice (choose-from-menu men :position
                                  (make-position :x 800 :y 400))))
               (when (member choice lis)
                 (put.value 'browser 'message
                   (format () "~%The value of ~s = ~s"
                           choice
                           (get.values thisunit choice)))
                 (show-slots thisunit)))))))

(defun browse-kb ()
"Allows the user to select a different kb for the browser."
  (let* ((kbs (unit.children 'knowledgebases 'member))
         (menu
          (make-cascading-menu :pop-up
                       (mapcar #'(lambda (unit)
                                   (unit.name unit))
                               kbs)))
         (choice (choose-from-menu menu :position
                               (make-position :x 800 :y 400))))
    (when choice
      (put.value 'browser 'current.kb choice)
      (put.value 'browser 'top.level.units
            (get-top-level-units choice))
      (get-tree choice))))
```

```
(defun get-tree (kb)
"Allows the user to select which tree to be displayed if more than one tree is
available from the top level in the kb."
  (let ((possible-units (get-unit-names
                          (get.value 'browser 'top.level.units))))
    (cond ((null possible-units)
           (put.value 'browser 'message (format ()
          "It is not possible to browse the kb ~S. Select a different kb."
             kb))
           (browse-kb))
          ((= (length possible-units) 1)
           (let ((tree-top (first possible-units)))
             (put.value 'browser 'tree.top tree-top)
             (show tree-top kb)))
          (t (let* ((menu
                      (make-cascading-menu :pop-up possible-units))
                     (choice (choose-from-menu menu :position
                                   (make-position :x 800 :y 400))))
               (cond (choice
                       (show choice kb)
                       (put.value 'browser 'tree.top choice))
                      (t (let ((tree-top (first possible-units)))
                           (put.value 'browser 'tree.top tree-top)
                           (show tree-top kb)))))))))

(defun get-top-level-units (kb)
"Returns a list of the top level units in the kb. Note that this function
excludes activevalue and activeimage units and any other units that are members
of a class other than classes or entities from another kb."
  (let ((classes (unit.children 'classes 'member))
        (result ()))
    (dolist (class classes result)
      (when (and (eq (kb.name (unit.kb class)) kb)        ;In correct kb
                 (= (length (unit.parents class 'member)) 1);Classes only
                 (null                               ;member parent
                  (remove 'classes                   ;Entities & classes only
                          (remove 'entities          ;class parent
                                  (mapcar #'unit.name
                                          (unit.parents class 'subclass))))))
        (push class result)))))

;;; BROWSER CONTROL

(defparameter outw
  (create-kee-output-window :region (make-region :left 550 :bottom 220
                                                 :width 500 :height 500)
                            :title "BROWSER"
                            :border 5
                            :activate-p nil)
  "Defines the output display window for the browser.")
```

```
(defun browse-entire-system ()
"Initializes the slots in the unit browser.  Opens the browser window.  Then
activates and exposes the output window for the browser and graphs the tree
including the unit workbench and its immediate children from the kb veg."
   (remove.all.values 'browser 'browser.menu)
   (put.value 'browser 'current.kb 'veg)
   (put.value 'browser 'top.level.units (get-top-level-units 'veg))
   (put.value 'browser 'tree.top 'workbench)
   (unitmsg 'viewport-browser.1 'open-panel!)
   (activate outw)
   (expose outw)
   (show 'workbench 'veg))
```