EECS 837
Spring 2014

Programming Project

Due: midnight from April 26 to April 27, 2014

Implement the discretization algorithm, based on conditional entropy, Multiple Scanning approach, using any programming language you wish. **Your project should be implemented on one of the departmental computers, e.g., cycle1, under Linux**. Potential cutpoints for any numerical attribute should be the averages of consecutive numerical values. Intervals should be listed in the following format: a number, two dots, a number, e.g., 3.27..5.64.

Input data files will be in the LERS format. Input data files will start from the list of variable declarations. This list starts form "<", then a space, then a sequence of the following symbols: "a", "x", or "d", separated by spaces; then another space, the last symbol of the list is ">". You may ignore this line. The second list of the input data file starts from "["; then a space, then comes a list of attribute and decision names, separated by spaces; then another space, and then "]". The decision will be always the last variable, all remaining variables are attributes. The following part contains values of the attributes and decision. The input data file may contain comments. A comment starts from "!"; everything that follows "!", until the end of the line, is the comment. Obviously, comments should be ignored during reading of data. Variable values are separated by spaces. Spaces should be understood not only as ordinary spaces but also as white space characters, such as the end of line, tab, etc. Any line of the input data file may start from one or more spaces and one or more spaces may end it. Note that a "line" does not need to be a physical line (it is rather a paragraph). In this project, attribute values are numerical and decision values are symbolic. Examples of symbolic values are: *medium*, 12..14, 1.25..2.37, etc. All input data sets will be consistent. The discretized data file should be consistent as well.

Your program should ask the user for the name of the input data file, say `test.d`. The expected response of the user is a name followed by pressing the <RETURN> key. The next question is about the number of scans. The expected response of the user is a non-negative integer followed by pressing the <RETURN> key. The program should create two files, the first one should be named `test.int`, with information about discretization (for all numerical attributes intervals should be listed) and the other one, named `test.data`, with the discretized data file in the format of LERS.

General Remarks. Your program should be able to deal with unexpected answers of the user and not crush but rather repeat the question. Use of recursion is not encouraged. You should expect input data files of any size, more than a thousand cases and more than a hundred attributes**.**

Include all comments, including instructions about compiling and linking, in a single file called **read.me**. Do not forget to include your name and KUID#. When you are ready to submit the project, send ALL necessary source files, makefile (if any), and the read.me file by e-mail to the grader and to the instructor. Do not send object files, executable files, and test data files. Late projects will be accepted with 10% penalty per day up to five days.