Introduction to Data Mining
Project 1: Data Pre-Processing

In this project, students are to program data pre-processing techniques on gene expression datasets.

The dataset (P1InputData.csv) provided in the project folder contains 62 samples collected from colon-cancer patients of two classes; there are 22 positive tuples and 40 negative ones. Each tuple (row) consists of the readings for the genes and the class (which is the last column) on one biopsy. Each gene is an attribute. The columns are separated by ",". We number the genes 1 to N in the left-to-right order; we will refer to the genes using gi where i is a column number; for example the first gene (column) is called g1.

Your program should work on other datasets with similar formats but they may have different number of rows and different number of columns (perhaps also different class names). You can assume that there are exactly two classes. So your program may need a scan of the data to determine the number of genes and the number of instances/rows.

Your (compiled) program will be run using the following command-line command:

```
java DPP datafilename k m
```

Your program will perform the following two tasks.

Task 1. Discretize, rank, and select the top-k genes of the data using the entropy-based method (for 2 intervals). This task produces three files; only the k highest ranked genes (in the information gain order) will be included in these files:

(a) A file (called entropyRank.csv) containing the entropy-based ranking of the k genes, in decreasing information gain order. Each row of the file should contain a gene number, the split value determined by the entropy-based binning method for the gene, and the information gain of the split. The three values should be separated by commas.

(b) A file (called entropyItemMap.csv); each row in this file contains four comma-separated values gi, lb,rb,j, where gi is a gene ID, lb is the left bound and rb is right bound of an interval/bin, and j is the integer to be used to represent the interval in the itemized data.

(c) A file (called entropyItemizedData.txt) containing the itemized data for the top-k genes. Each row in this file corresponds to a row in the input data. Use commas to separate items in each row. The first item of the rows should be about the highest ranked gene, the second item should be about the second highest ranked gene, and so on.

Task 2. Discretize the top-k genes selected in Task 1, using the equi-density binning method into m intervals. This task will produce two files:

(a) A file (called equiDensityItemMap.csv); each row of this file contains four comma-separated values gi, lb,rb,j, where gi is a gene ID, lb is the left bound and rb is right bound of an interval, and j is the integer to be used to represent the interval in the itemized data.

(d) (b) A file (called equiDensityItemizedData.txt) containing the itemized data for the top-k genes. Again the first item of the rows should be about the highest entropy-ranked gene, the second item should be about the second highest entropy-ranked gene, and so on.

Slide 46 of 03Preprocessing.ppt contains an example dataset, a discretization map, and an itemized dataset. The dataset you will work with does not contain the row for gene names.

You need to write your program in Java. When complied your program will produce an executable program called DPP. The program should be able to do the two tasks described above using a command of the form “java DPP datafilename 3 4” when it is run with the “datafilename” file in the same folder where the program is. The “3” is the k (number of genes for entropy-based ranking) and the “4” is the m (number of intervals for equidensity).

Submission Guidelines: Submit on pilot a zipped file containing a folder containing your program files and a file called ReportYourLN (pdf, doc, or txt) containing all of the resulting files produced by your program when k=3 and m=4. YourLN refers to your last name. ReportYourLN should contain three sections (How to compile my program, entitled Entropy Results and EquiDensity Results respectively); the "How to compile my program" section should contain one or two lines indicating how to compile your program, the "Entropy Results" section should contain the three files produced by Task 1 in the order described above, and the "Equidensity Results" section should contain the two files produced by Task 2 in the order described above.

We will test your program using command line commands.

Correctness, elegance, and readability will be factors to be considered in the marking process. If your program is found to not to compile then you will get zero marks for this project. If it does not produce the correct output files then it is incorrect. Efficiency and documentation may also be considered in the marking process. Test your program on a different computer (not used by you in the developing phase) to avoid hard-wiring errors.

The Work Must Be Your Own: You cannot use code found from the Internet or from any other sources in this project.

Additional information, including necessary corrections, will be posted on pilot as news items.

The dataset used in this project was first published by U. Alon, et al. in PNAS, 96:6745-6750, 1999.