**Relief Algorithm with Multiple Processors**

**CMPE300**

**Meltem Arslan**

**27.01.2021**

**Introduction**

In this Project, we are expected to write the Relief Algorithm with multiple processors working in parallel. We have p processors with one master processor and p-1 slave processors. All I/O operations are being handled by the master processor P0 then data is separated into p-1 group in order to be scattered to slave processors. After the scatter operation, each processor will have equal number of instances. To order to apply the Relief algorithm we need 2 helper functions: diff and Manhattan Distance. The main operations of the Relief Algorithm are being executed in the slave processors. Each slave will print its outputs to the screen. After all processes finish in each processors the results will be sent to the master processor. The master processor will print the result recieved from the slave processors to the screen.

**Program Execution**

I compile my code with:

>mpic++ -o <outputfile> ./cmpe300\_mpi\_2016400117.cpp

I run my code with (P: number of processors):

>mpirun –oversubscribe -np <P> <outputfile> <inputfile>

**Program Structure**

First the master reads the data values from the input file and the data will be scattered to the p-1 slave processors. Each processor has their instance values with A features and labels.

The Manhattan distance is calculated for each 2-instance given as a parameter

Then we get the maximum and the minimum values of specified feature.

Then I calculated the diff using hit, miss, max and min.

After getting the sorted weights, each processor sends the indexes of T max weights to the master.

Each processor and the master prints the indexes without duplication

**Difficulties Encountered**

The most difficult part of the Project was installing the openmpi. I was doing most of my projects on Windows, so first I tried to install MPICH2 under Windows but I could not execute the sample program because of the problem about “user credentials”. Although I spend my hours to solve the problem by trying all possible solutions suggested on some forums on the web, I could not solve it. Then I decided to continue with virtual machine.

The second major problem for me was to understand the mpi structure since it was the first time for me with mpi. It was confusing for me to use multiple processes, multiple instances, and multiple features. And having label data at the end of the feature values made difficult to calculate index values of array(because first I used 1 dimensional array to hold the data since I did not know whether it is possible to scatter 2-dimensional array or not). Then I converted the 1-d array into 2-d array, it was easier to calculate its indexes.

**Conclusion**

Although it was very difficult for me to build the general structure of the project and to understand the Relief and diff calculations, it was very satisfying since I learned lots of useful information with it.