Parallel implementation of Binary Classification

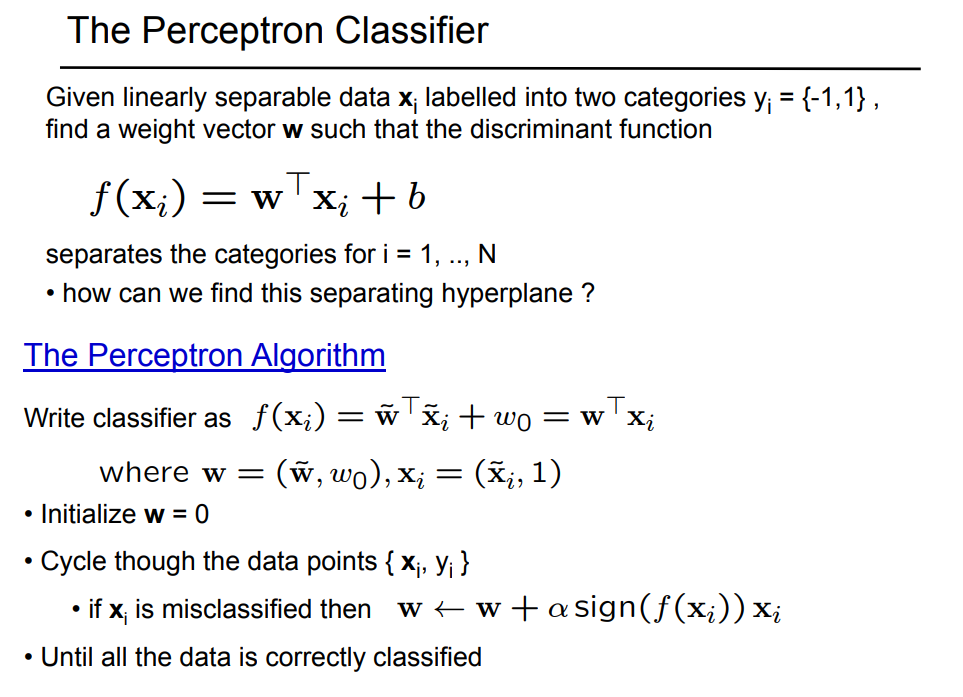
Final project

Course 10324, Parallel and Distributed Computation

2019 Spring Semester

# Problem Definition

Given a set of **N** points in **K**-dimensional space. Each point **X** is marked as belonging to set **A** or **B**. Implement a Simplified Binary Classification algorithm to find a Linear Classifier. The result depends on the maximum iteration allowed and a value of the chosen parameter ****. The purpose of the project is to define a minimal value of **** that leads to the Classifier with acceptable value of Quality of Classifier.

****

# Sequential Implementation of Simplified Binary Classification algorithm

1. Set  = 0
2. Choose initial value of all components of the vector of weights W to be equal to zero.
3. Cycle through all given points Xi in the order as it is defined in the input file
4. For each point Xi define a sign of discriminant function f(Xi) = WT Xi. If the values of vector W is chosen properly then all points belonging to set A will have positive value of f(Xi) and all points belonging to set B will have negative value of f(Xi). The first point P that does not satisfies this criterion will cause to stop the check and immediate redefinition of the vector W:

W = W + [\*sign(f(P))] P

1. Loop through stages 3-4 till one of following satisfies:
   1. All given points are classified properly
   2. The number maximum iterations **LIMIT** is reached
2. Find Nmis - the number of points that are wrongly classified, meaning that the value of f(Xi) is not correct for those points. Calculate a Quality of Classifier q according the formula

q = Nmis / N

1. Check if the Quality of Classifier is reached (q is less than a given value QC).
2. Stop if q < QC.
3. Increment the value of :  =  + 0. Stop if > MAX
4. Loop through stages 2-9

Input data and Output Result of the project

You will be supplied with the following data

* **N** - number of points
* **K** – number of coordinates of points
* Coordinates of all points with attached value: 1 for those that belong to set A and-1 for the points that belong to set B.
* **0** – increment value of , **max** – maximum value of 
* **LIMIT** – the maximum number of iterations.
* **QC** – Quality of Classifier to be reached

# Input File format

The first line of the file contains **N K 0**  **max LIMIT QC**.

Next lines are coordinates of all points, one per line, with attached value 1 or -1.

For example:

200000 5 0.1 2.5 200 0.025

2.3 4. 5 6. 55 -2.3 13.3 1

76.2 -3.56 50.0 12 -0.7 -1

…

45.23 20 -167.1 98.0 -113.2 -1

# Output File format

The output file contains following information

* The minimal value of**** with **q < QC** and a value of **q**. If for every checked value of **** the value of **q** is bigger than **QC** – “Alpha is not found” is printed.
* Values of corresponding weights W

# Output File format

Alpha minimum = VALUE q = 0.021

W1

W2

W3

…

WK

For example:

Alpha minimum = 1.75

12.2

-3.4

25.5

16.34

0.12

# Requirements

* Implement the Simplified Binary Classification algorithm explained in the class (see above).
* The input file **input.txt** initially is known for one machine only. The results must be written to the file **output.txt** on the same machine
* The computation time of the parallel program must be faster than sequential solution.
* Be ready to demonstrate your solution running on at least two computers (if MPI is used)
* **No code sharing between students is allowed.** Each part of code, if any, which was incorporated to your project must be referenced according to the academic rules.
* Be able to explain each line of the project code, including those that was reused from any source.
* The set contains at least **100000** but not more than **500000** points, the number of dimensions is less than **20,** the number of values of ****to check is less than **100,** no more than **1000** iterations**.**

# Grade Policy

* **60 points** for the effective **proper** parallel implementation of the problem with two components: ***MPI+OpenMP*** or ***OpenMP+ CUDA*** or ***MPI+CUDA***. The project that produce wrong results will not be accepted.
* **10 points** for implementation in ***MPI+OpenMP+CUDA*** configuration.
* **10 points** for the documentation of your solution – clear explanation what and how the problem was parallelized, what is a rational of choosing the specific architecture, complexity evaluation.
* **10 points** for the code quality – modularity, generality, self-explanatory, organization.
* **10 points** for the Load Balancing.

***Additional Bonus for the project grade***

**5 points** for implementation under LINUX OS

**5 points** for implementation with OpenCL

**5 points** for implementation of sophisticated variation the algorithm(must be approved by lecturer).

**5 points** for your own proposal (must be approved by lecturer).

# הפרויקט יוגדר כמטלת הקורס. הגשת התוכנה והתיעוד רק דרך מערכת Moodle לאחר ההגנה.

# יישום והגשת הפרויקט ביחידים בלבד.

בהצלחה