

# A Classifier from Bottom-Up

By Tiefei Zhang

## Goal:

This is a simple classifier based on a bottom-up flavor. There are two types of nodes in a 2D plane, two nodes belong to a same type are selected each time, leading to a line being determined by these two nodes. Image this line generates two parallel brothers, and one of these moves in the up direction and the other in the opposite direction to include more nodes of the same type. These two lines stop when a node of different node is touched.

By this way, nodes between these two lines are separated from the other type. By iterating the above steps, two types of nodes are separated by a series of pairs of parallel lines.

## Platform & Data Set

The source file-hw3.cpp is developed under Dev-C++, Window XP. The execution file hw3.exe can run successfully. One hundred nodes(x, y) in data.txt comprise the input data set, which are generated randomly. The first fifty members make up the type 1 data set and the left fifty, the type 2.

The output of the simulation is some line-pairs each following a number denotes the number of data between the line-pair. The line is formulated as follows:

$$y = bx + c$$

where b is the slope and c is the offset, so each line is denoted by two values: b and c.

I adopted matlab7.5.0 to draw the input-nodes and output-lines. The parameter of the lines are from the output of line\_classification.exe.

## 1. Some Tricks

1) Given a node  $(x_i, y_i)$  and a line  $y = bx + c$ , for simplicity, I use  $|y_i - b_i - xc|$

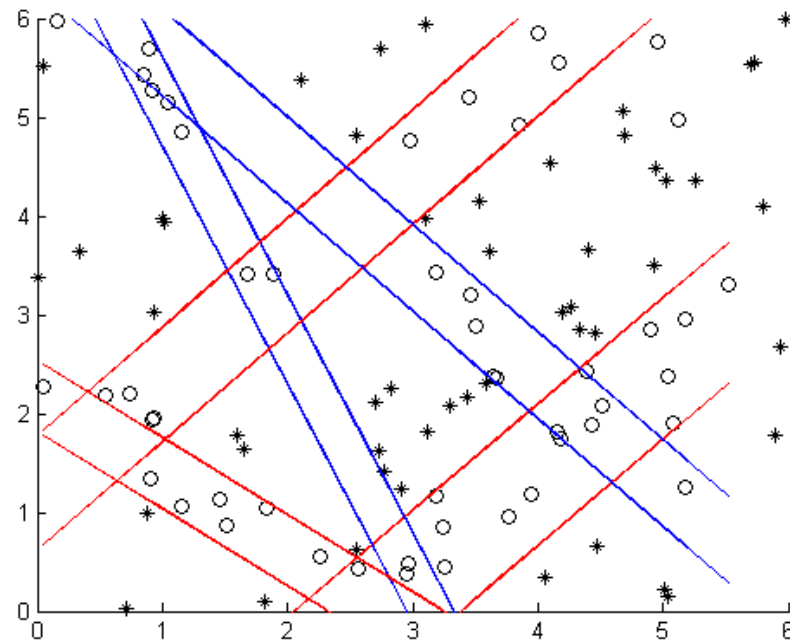
as the measure of distance instead of  $|\frac{ax_i + y_i + c}{\sqrt{a^2 + 1}}|$ .

2) As our goal is to classify the data, the same can be done if we manage to separate one class from the other. Based on this idea, I try to separate type1 data by always randomly choosing two points from type1 class.

3) To ensure a considerable number of nodes being classified each time, a threshold is defined to bound the least number between two lines.

## 2. Result

- 1) The simulation works correctly but leaves three points that can not be classified, which is, I think, due to the initialization value of the rand function. If more time spent, and carefully initialization of rand function, no nodes will be left.
- 2) The output graph:



## 3. Result

This implementation is based on a paper from Professor Cheng-Yuan Liou, but I do not know which paper it is☹.

Professor Cheng-Yuan Liou: <http://www.csie.ntu.edu.tw/~cyliou/>