\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The BLOCK\_DBSCAN program was compiled under Windows using c++ with CodeBlocks 10.05.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

===================================================================================

**Files**

===================================================================================

Unzip the "BLOCK\_DBSCAN.zip" file, which will produce a folder mainly containing:

- a project file named "BLOCK\_DBSCAN.cbp".

- a dataset folder named "data" .

- a c++ file named “BLOCK\_dbscan.cpp” is the main function file.

- other c++ files

===================================================================================

**Environment configuration**

===================================================================================

Step1:

- Download CodeBlocks in <http://www.codeblocks.org/>

- Download TDM-GCC-32 in <http://tdm-gcc.tdragon.net/download>/

Step2:

- Open CodeBlocks choose “setting” ->”compiler and debugger”->”ToolChain executables”,and set the parameters like the “Figure 1” shows:

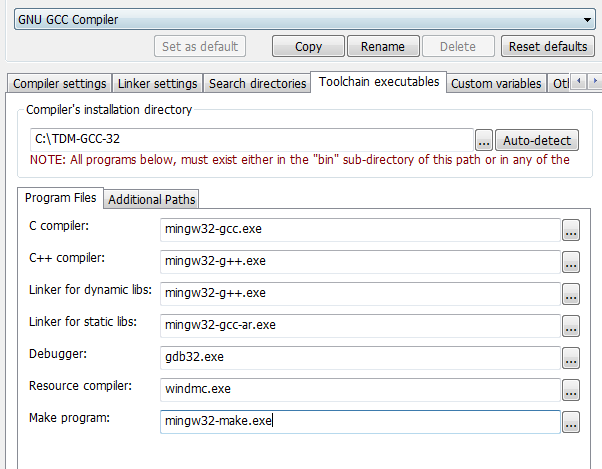


Figure 1: Configuring the compiler options

===================================================================================

**Dataset Format**

===================================================================================

The dataset should be given in a text file of the following format:

- In every line, there are $d+1$ numbers: the id of the point and the $d$ coordinate values, where the id is an integer from 0 to $n-1$ and each coordinate value is an integer.

For instance, the first 10 lines of the sample dataset "points\_new.txt"(who’s data number is 2000 and dimension is 2) are shown as below:

-3.6258594e-01 1.7603038e-02 0

-3.5135438e-01 8.1635324e-02 1

-3.2205003e-01 9.3703875e-02 2

-3.3384577e-01 7.6720818e-02 3

-3.2645484e-01 1.0577463e-01 4

-3.2423144e-01 6.2867479e-02 5

-3.8838441e-01 1.8080861e-02 6

-3.3523700e-01 8.7956681e-02 7

-3.4289436e-01 4.8453561e-02 8

-3.9319662e-01 -1.4828753e-02 9

There are 3 numbers in each line: 0 is the id of this point and (-3.6258594e-01, 1.7603038e-02) are its coordinates. Namely, this line specifies that the point with id = 0 is (-3.6258594e-01,1.7603038e-02). Analogously, the rest nine lines above specify the coordinates of the point with id = 1, 2, 3, 4, 5, 6, 7, 8, 9 respectively.

===================================================================================

**An example of quick start**

===================================================================================

Step1:

Open project “BLOCK\_DBSCAN.cbp” in Codeblocks.

Step2:

Open “BLOCK\_dbscan.cpp”.

In line 22 : myClusterAnalysis.Init("data\\points\_new.txt",0.05,55);

-The first parameter "data\\points\_new.txt" represents the origin dataset that you want to do clustering.

-The second parameter “0.05” represents Epsilon(eps).

-The third parameter “55” represents MinPts.

In line 29:myClusterAnalysis.WriteToFile("data\\result.txt");

-The parameter "data\\result.txt" represents the results file that you want to write in.

Step3:

Open “DataPoint.h”.

In line 4 :const int DIME\_NUM = 2+1;

-The value of the DIME\_NUM should be set as $d+1$.

Step4:

Open “point.cpp”.

If you want to use different distance type:

1. Euclidean distance: annotate line 42-49. Release note in line 32-39.(shown as Figure 2)

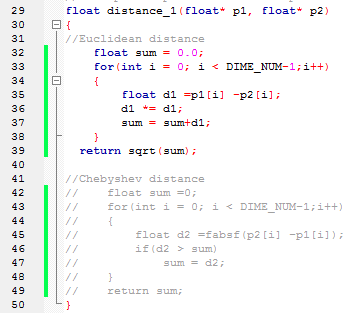


Figure 2: Choose Euclidean distance

1. Chebyshev distance: annotate line 32-39. Release note in line 42-49.(shown as Figure 3)

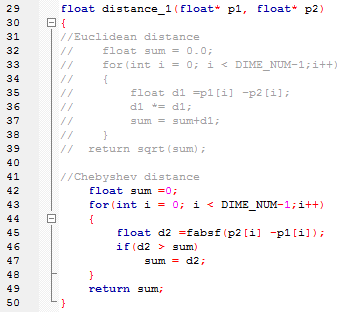


Figure 3: Choose Chebyshev distance

Step5:

-Press the “Build and run” button in CodeBlocks and the code will be run.

The result will perform like Figure 4:

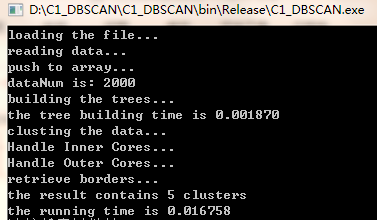


Figure 4: The result of BLOCK\_DBSCAN

===================================================================================

**Output Format**

===================================================================================

The output of our program consists of one file(“data\\result”):

The first 10 lines in the file are as shown below:

0 1

1 1

2 1

3 1

4 1

5 1

6 1

7 1

8 1

9 1

The first line means that the point with id = 0 whose cluster ID is 1. Analogously, the rest lines specify which cluster the point should belong to. And in particular, if cluster ID is -1, that represents this point is a noise point.