

# Report on K-Mean Algorithm Implementation for Discrete Cover problem

Raj Kamal  
Roll no :-09012321

May 5, 2013

## ABSTRACT

The purpose of this report is to give manual of the continuous and discrete version of k-MEAN algorithm. We have used the Continuous and Discrete K Mean Algorithm to approximate the solution of Discrete Square Unit Cover Problem. Given  $n$  red point and  $m$  blue point. We have the red points using blue points. A red point is said to be covered if we draw a circle of radius  $rad$  center around a blue point out of blue point set, then that red point must lie inside the circle. We will use Discrete version of K Mean algorithm to arrive to the answer.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Code</b>	<b>5</b>
2.1	Continuos K Mean Algorithm . . . . .	6
2.1.1	Usage . . . . .	6
2.2	Descrete K Mean Algorithm . . . . .	8
2.2.1	Usage . . . . .	8

# Chapter 1

## Introduction

- **Header Files**

1. **iostream.h**

This is standard C++ header file for input and output operations.

2. **cstdio**

This is standard C++ header file for using functions defined in C

3. **fstream.h**

This is standard C++ header file for file operations

4. **cstdlib**

This is standard C++ header file for using exit() functions

5. **cmath**

This is standard C++ header file for mathematical functions like sqrt() square root.

6. **vector**

This is standard C++ header file for implementing vectors and vector related functions. Vectors are dynamic array.

7. **glu.h glut.h glui.h**

This is standard OpenGL Header files for implementing GUI .

- **User Defined Header File**

*user\_def.h* is user defined library file .All Global variables, Function Prototypes and User defined Functions are listed here.All parameters like height,width of frame are listed here.Any ways definitions are explicitly defined in other files.

- **Generating Functions**

*algo\_fun.cpp* contains functions relating to generating points.All different generating functions are defined here.

- **Algorithm Functions**

All algorithm related functions defined in *algo\_fun.cpp*.Like creating cluster, changing median , creating actual cluster.

- **Display Functions**

All algorithms related to display are defined in *display\_fun.cpp*.Like displaying points, displaying actual cluster and all.

- **KeyBoard Functions**

Several keys are used for implementing functions. All these are defined in *keyboard\_fun.cpp*. All the important keys are displayed to console.

- **Mouse Functions**

Mouse left and middle buttons are used for implementing functions that are defined in *mouse\_fun.cpp*. It includes functions like drawing red point and blue points. Also functionality associated with right click are defined in *main.cpp* due to some technical issues.

- **Main Function**

Like every *C++* project contains one main function, that main function is defined in *main.cpp*. It also contains definitions of Global variables which can be set here. Also it contains the functionality definitions of right click of mouse button.

- **menu Function**

It displays the menu at console and defined in *menu.cpp*

## Chapter 2

# Code

- Install some of the libraries  
`sudo apt-get install freeglut freeglut3-dev g++`
- The code is made using C and C++ libraries.
- It implements the Continuous K Mean Algorithm and Discrete K mean algorithm in background and display the results.
- Several Global Variables are defined in *main.cpp* file where any user can put their value
- There are several functionality with keyboard and mouse .Any time program hang then its the case see the command prompt ,it must be asking for input and if exited we can see *error.txt* to see the error.

## 2.1 Continuos K Mean Algorithm

### 2.1.1 Usage

Initially A blank screen will appear and menu on terminal. Any option written in menu in terminal will work when it is pressed while output screen window is open. We can generate points and then press **t** to get result.

- Press **esc** to quit the program.
- Press **0** to display the menu again.
- Press **Left mouse** to draw red point
- Press **Middle mouse** to draw blue point
- Press **Right mouse** button to explore further option.
  1. **Generate** .It opens another menu and pressing red blue or cluster will generate random red blue cluster. It will require input which will be taken at terminal.
  2. **Display** .It opens another menu and pressing the field will display that in terminal or output window.
  3. **Set**. It helps to set the variables used in program.
  4. **Exit** ,It will cause program to quit.
- We can generate specific number of red point at arbit position by pressing **r** while output window is open.
- Similary we can generate blue points by pressing **b**.
- We can generate both red and blue points at once by pressing **a**.  
**NOTE** we can set domain by changing upper and lower limits in *main.cpp*.
- We can check the feasibility by pressing **F**. It will display on terminal whether the points are feasible or not feasibility.
- We can output complete results with the red points and blue points by pressing **t**
- we can output results with the present cluster size keeping same red and blue points by pressing **I**
- we can output results with the incremented cluster size keeping same red and blue points by pressing **i**

**NOTE** We can get result in one step by pressing **t**. But if we want to see how it is actually showing results with incrementing cluster size for that first press **I** and then keep pressing **i** and see the results step by step. Also at any point of time if you want to see the output with present cluster size press **I**.

- We can see the point and cluster by pressing **A**.
- we can see the Blue points only by pressing **B**.
- we can see the red points only by pressing **R**.

- we can see the cluster only by pressing **C**.
- we can see the actual cluster by eliminating redundant cluster by pressing **D**.
- we can see the blue cluster,(contributing to result) corresponding to cluster algorithm found by pressing **E**.
- we can see the final cover or result by pressing **G**

**NOTE** Here algorithm found cluster at position where blue point are not situated. Initially it found cluster with the set value which we can see by pressing **C**. In that there may be redundant cluster we can see the result after removing them by pressing **D**. Now the cluster head may be not at blue point so we search for corresponding blue points which we can see by pressing **E** and finally we can see results by pressing **G**. Here only results that have been produced by pressing **i,I,t** are shown. It does not run any algorithm. It just display.

- we can see how many red points are generated by pressing **1**.
- we can see how many blue points are generated by pressing **2**.
- we can see how many cluster are there by pressing **3**.
- we can see how many cluster out of cluster resulted by algorithm participated in covering by pressing **4**.
- we can see number of blue point required to cover by pressing **5**.
- We can get complete result by pressing **6**.

**NOTE** we can see results in terminal only.

- we can regenerate same number of points and then display the result by pressing **h**. It is useful when we are analyzing results for fixed number of points. we can get an idea about how many points in the average required to cover.
- We can change cluster value and display the result by pressing **K,L,M**.
- We can do statistical analysis by pressing **S**. Here algorithm is run for 100,200,500,700,1000 red point. blue points are generated and cover is found and result is saved in file *result.txt*, other analysis are stored in *data.txt* and *convergence.txt*. We can change the number of red point by opening *keyboard\_fun.cpp* and changing line 374 and 376. Here input is asked for upper and lower bound for red and blue points..

**NOTE** For the first time pressing left mouse may not display red point but as middle mouse button is pressed, all points drawn can be seen. It happens for first time only.



## 2.2 Descrete K Mean Algorithm

### 2.2.1 Usage

Initially A blank screen will appear and menu on terminal. Any option written in menu in terminal will work when it is pressed while output screen window is open. We can generate result and press **r** to get result.

- Press **esc** to quit the program.
- Press **m** to display the menu again.
- Press **Left mouse** to draw red point
- Press **Middle mouse** to draw blue point
- We can clear the screen by pressing **A**.
- We can generate specified number of red and blue points at random places by pressing **w**.
- We can generate specifeid number of red point at random places by pressing **B**
- We can generate specified number of blue point at random places by pressing **R**.

**NOTE** we can set domain by changing upper and lower limits in *main.cpp*.

- we can check feasiblity by pressing **r**. Result will be displayed on terminal.
- We can output complete results with the red points and blue points by pressing **t**
- we can output results with the present cluster size keeping same red and blue points by pressing **I**
- we can output results with the incremented cluster size keeping same red and blue points by pressing **i**

**NOTE** We can get result in one step by pressing **t**. But if we want to see how it is actually showing resutls with incrementing cluster size for that first press **I** and then keep pressing **i** and see the results step by step. Also at any point of time if you want to see the ouptut with present clsuter size press **I**.

- we can show red points and blue points by pressing **a**.
- we can show red points only by pressing **j**.
- we can show blue points only by pressing **b**.
- we can display cluster points by pressing **c**.
- we can display actual clsuter point that is cover by pressing **d**.
- we can show results on terminal by pressing **4**.
- we can show number of red points, blue points and cover size by pressing **1,2,3** respectively.
- we can ulter K and incrementing factor by pressing **k,l,K**

- we can enter new radius by pressing **N**.
- we can show results by maintaining same cluster by pressing **u**
- we can shoe results by maintaining same cluster size but different cluster by pressing **v**.

**NOTE**The option **u,v** are given for analysis purpose.we can get idea about chossing dif-fernt cluster how result is going to get affected.

- For statistical analysis purpose we press **e**.we can details of it in line 121 of *keyboard\_fun.cpp*.All results are displayed in *result.txt*, *data.txt* and analyis part in *convergence.txt*.

**NOTE** For the first time pressing left mouse may not display red point but as middle mouse button is pressed , all points drawn can be seen.It happens for first time only.