User Manual

Justin Phan

As of November 2021

1 Visualizations

1.1 Mirrored In-Line Coordinates Plot

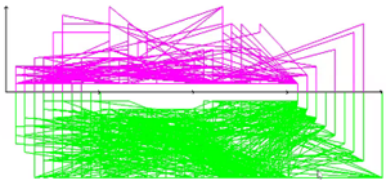

Figure 1: Mirrored In-Line Coordinates Plot

```
AboveGraphOrderly()
{
    glBegin(GL\_LINE\_STRIP);
            glVertex2f(a1[i][1], a1[i][2]);
            glVertex2f(a1[i][4] + 10, a1[i][3]);
            glVertex2f(a1[i][4] + 10, a1[i][5]);
            glVertex2f(a1[i][7] + 20, a1[i][6]);
            glVertex2f(a1[i][7] + 20, a1[i][8]);
            glVertex2f(a1[i][9] + 30, a1[i][9]);
            glEnd();
}
```

This function must be changed depending on the dimensions of the data set and the coordinate order.

```
BelowGraphOrderly()
{
    glBegin(GL\_LINE\_STRIP);
            glVertex2f(a1[i][1], 0);
            glVertex2f(a1[i][1], 0 - a1[i][2]);
            glVertex2f(a1[i][4] + 10, 0 - a1[i][3]);
            glVertex2f(a1[i][4] + 10, 0 - a1[i][5]);
            glVertex2f(a1[i][7] + 20, 0 - a1[i][6]);
            glVertex2f(a1[i][7] + 20, 0 - a1[i][8]);
            glVertex2f(a1[i][9] + 30, 0 - a1[i][9]);
            glVertex2f(a1[i][9] + 30, 0);
            glEnd();
}
```

This function must be changed depending on the dimensions of the data set and the coordinate order.

1.2 Partial dynamic In-Line Coordinate Plot

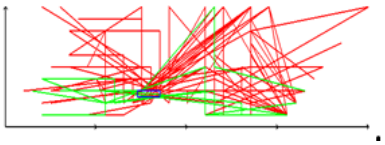

Figure 2: Mirrored In-Line Coordinates Plot

AboveGraphOrderly function (see above)

1.3 Fully dynamic In-Line Coordinate Plot

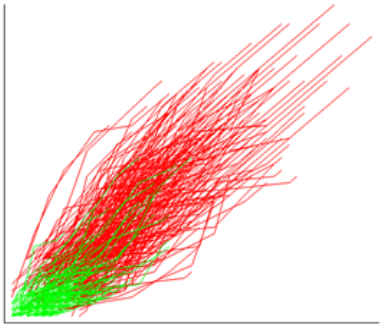

Figure 3: Mirrored In-Line Coordinates Plot

```
dynamiclinecoordinate()
{
    glBegin(GL\_LINE\_STRIP);
            glVertex2f(A1[i][1], A1[i][2]);
            glVertex2f(A1[i][1] + A1[i][3], A1[i][4]);
            glVertex2f(A1[i][1] + A1[i][3] + A1[i][5], A1[i][6]);
            glVertex2f(A1[i][1] + A1[i][3] + A1[i][5] + A1[i][7], A1[i][8]);
            glVertex2f(A1[i][1] + A1[i][3] + A1[i][5] + A1[i][7] + A1[i][9], A1[i][9]);
            glEnd();
}
```

This function must be changed depending on the dimensions of the data set and the coordinate order.

1.4 Partial dynamic In-Line Coordinate Plot with Boxes

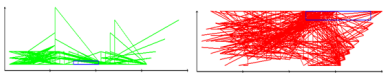

Figure 4: Mirrored In-Line Coordinates Plot

AboveGraphOrderly function (see above)

```
drawingBox
{
glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);
        glRectf(x1, y1, x2, y2);
}
```

To use this function modify x1, y1, x2, and y2 coordinates to plot a box at a certain position.

2 Classification

After drawing the visualizations this program can run an exhaustive grid search to classify a data set. An exhaustive grid search is not practical in very high dimension data sets.

Decision tree or reduce grid search area depending on data trends can be used to optimize the grid search run time.

2.1 Functions used in Classification

1. cohenSutherlandClip function:

   Check if a sample crosses or touches a box

2. Howmanylines function

   How many samples are found in the box using the cohenSutherlandClip function.

3. RemainingLines function

   Stores remaining samples that weren't in the box.

4. LinesThroughBox function

   Store samples that that were in the box.