Lab 6: Clustering and Community Detection

Ruofan Zhou

May 6, 2014

1 Clustering

1.1 codes

1. mStep of Kmeans:

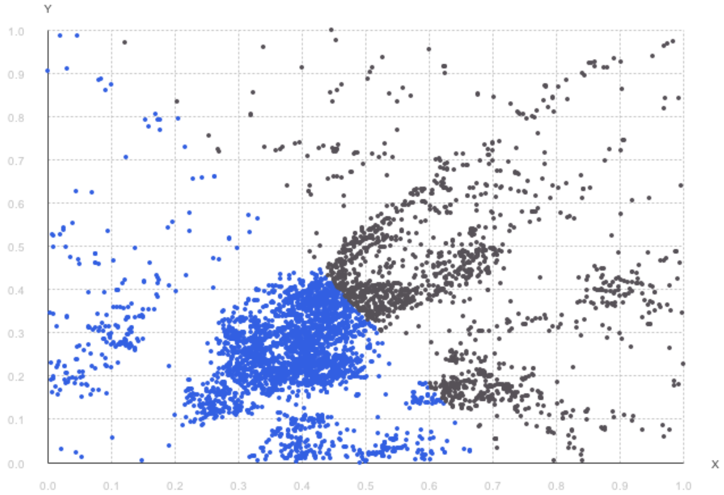
```
public void mStep() {
        // According to the assignment, we update the center of each cluster
        for (int c = 0; c < this.k; ++c) {
                Point2d rnk = new Point2d(0, 0);
                double sum = 0.0;
                for (int p = 0; p < this.data.length; ++p) {
                        if (this.assignments[p] == c) {
                                sum = sum + 1;
                                rnk.set(rnk.getX() + this.data[p].getX(), rnk.getY() + this.data[p].getY());
                        }
                }
                this.centers[c].set(rnk.getX() / sum, rnk.getY() / sum);
        }
    }
```

2. mStep & eStep of Kmeans:

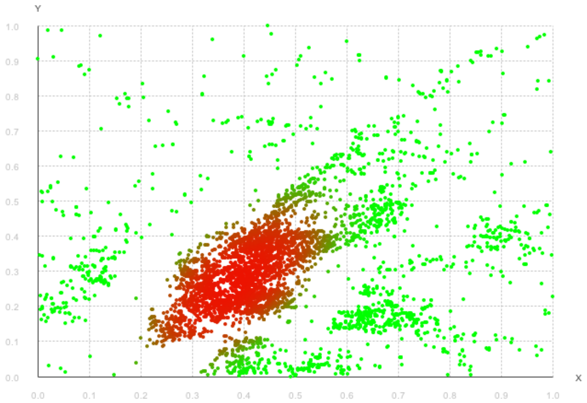
```
public void eStep() {
        // Hint: look at the MultivariateNormalDistribution class used in logLikelihood
        MultivariateNormalDistribution[] pdfs = new MultivariateNormalDistribution[this.k];
        for (int c = 0; c < this.k; c++) {
            pdfs[c] = new MultivariateNormalDistribution(this.mus[c].toArray(),
                    this.sigmas[c].toArray());
        }
        for (int p = 0; p < this.data.length; ++p) {
                double sum = 0.0;
                for (int c = 0; c < this.k; ++c) {
                        sum += pdfs[c].density(this.data[p].toArray()) * this.pi[c];
                }
                for (int c = 0; c < this.k; ++c) {
                        this.gamma[p][c] = this.pi[c] * pdfs[c].density(this.data[p].toArray()) / sum;
                }
        }
    }

public void mStep() {
        for (int c = 0; c < this.k; ++c) {
                double Nsum = 0.0;
                //compute mus
                this.mus[c].set(0.0, 0.0);
                for (int i = 0; i < this.data.length; ++i) {
                        Nsum = Nsum + gamma[i][c];
                        this.mus[c].set(this.mus[c].getX() + gamma[i][c] * this.data[i].getX(),
                                        this.mus[c].getY() + gamma[i][c] * this.data[i].getY());
                }
                this.mus[c].set(this.mus[c].getX() / Nsum, this.mus[c].getY() / Nsum);

                double x = 0.0, y = 0.0, xy = 0.0;
                for (int i = 0; i < this.data.length; ++i) {
                        double a = this.data[i].getX() - this.mus[c].getX();
                        double b = this.data[i].getY() - this.mus[c].getY();
                        x += a * a * this.gamma[i][c];
                        y += b * b * this.gamma[i][c];
                        xy += a * b * this.gamma[i][c];
                }
                this.sigmas[c].set(x / Nsum, y / Nsum, xy / Nsum);
                this.pi[c] = Nsum / this.data.length;
        }
    }
```

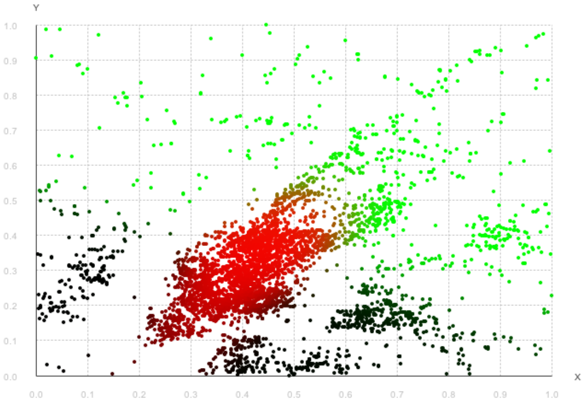
1.2 answers to the questions

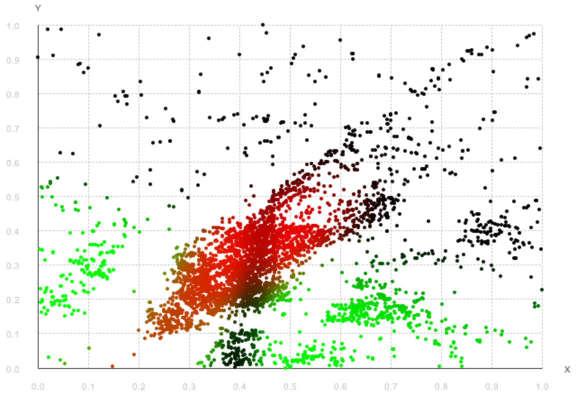
1. The plot of tweet by K-means is below. The geographical distribution of the clusters shows nothing with Manhattan.



2. The measure used in GMM that is the equivalent of the distortion measure in k-means is *LogLikelihood*.
3. GMM plots for tweet are as below:







2 Community Detection

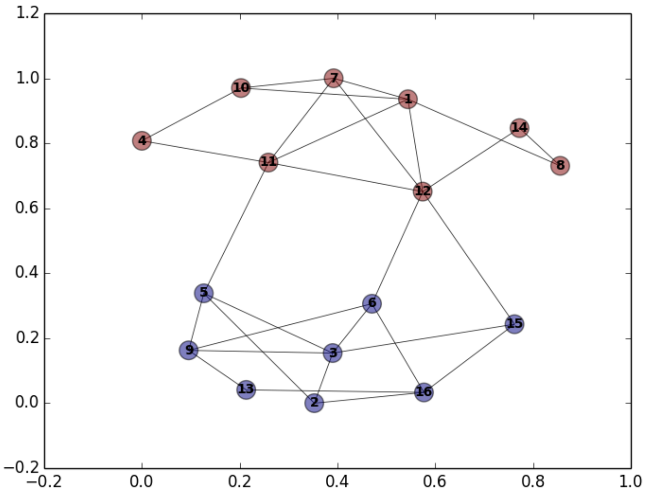
2.1 codes

communityDetection() of Louvain:

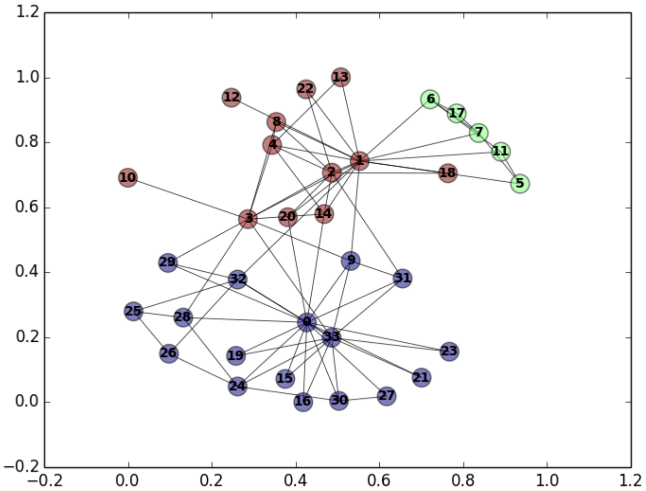
```
public void communityDetection() {
        double delta = 0.0;
        int level = 0;
        do {
                Status s = statusList.get(level ++);
                s.assignCommunities();
                Status s1 = s.getNextLevel();
                statusList.add(s1);
                delta = s1.modularity() − s.modularity();
        } while (delta > CHANGE_MIN);
        System.out.println(statusList.get(level).modularity());
    }
```

2.2 plots

1. Plot of the simple graph and the communities found at the first level of the Louvain method:

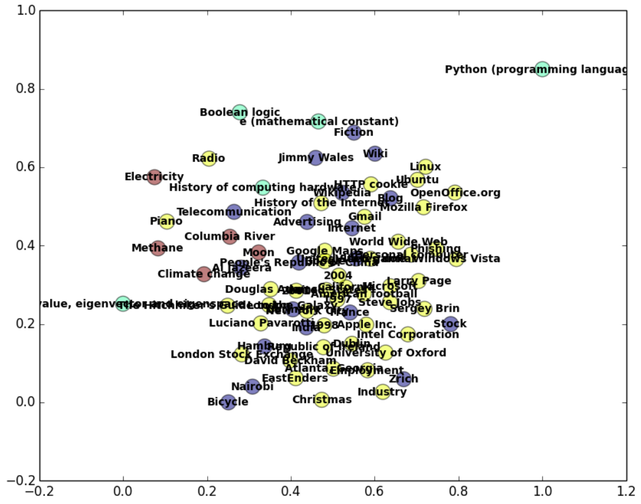


2. Plot of the karate graph and the best communities found by the Louvain method:



3. Highest modularity of the wikipedia graph found by the Louvain method is *0.5236077874099563*.

4. Plot of communities of the node Google and its neighbors from the wikipedia graph:


Python (programming languag
Boolean logic
e (mathematical constant)
Fiction
Radio
Jimmy Wales
Wiki
Linux
Electricity
Ubuntu
History of computing hardware
HTTP cookie
Wikipedia
OpenOffice.org
History of the Internet
Blog
Mozilla Firefox
Telecommunication
Piano
Advertising
Gmail
Internet
Columbia River
World Wide Web
Methane
Moon
Google Maps
Phishing
Personal computer
Windows Vista
People's Republic of China
Al Jazeera
Climate change
Larry Page
Douglas Adams
California
Microsoft
American football
Steve Jobs
value, eigenvector and eigenspace
The Hitchhiker's Guide to the Galaxy
New York City
France
Sergey Brin
Luciano Pavarotti
India
Apple Inc.
Stock
Intel Corporation
Hamburg
Republic of Ireland
Dublin
London Stock Exchange
University of Oxford
David Beckham
Atlanta, Georgia
Employment
EastEnders
Zrich
Nairobi
Industry
Bicycle
Christmas