Paul-Arthur Thiéry

Alexandre Van Wesel

Biclustering Report

Introduction

As we have learned during this project, Biclustering is a data mining strategy which allows to group data with resembling subsets of attributes in clusters. This allows researchers to find patterns in large sets of data. However, the notion of close resemblance on specific subsets was something that we had a hard time understanding, which is reflected in our strategy.

We chose a dataset of country flags around which we built our algorithm.

Strategy

As we were confused by the principles of Biclustering, our original strategy was something that later appeared to us as clustering strategy.

We decided to compute a numerical difference score between every flag. These scores, named *distance*, were in a matrix. This would later allow us to cluster similar flags by grouping flags with low distances. The distance is computed by adding the difference between each attribute of two flags. Because some non-binary values exist, these difference are scaled from 0 to 100. A distance can therefore go from 0 to 3000.

Our next step is to find valid first flags around which we can create our clusters. Preliminarily, we cut a low percentage of flags with the highest average distance from the group in order to avoid starting a cluster with a greatly off-centered flag.

Originally, we would select the remaining pairs with the greatest distance, then use the most centered of its flags, but this wasn't reliable and caused a lot of problems. We later decided to search instead of a pattern of equidistant flags, which proved much more reliable.

Once our first flags are set, we can create a Stack of remaining Flags. We get the distance between the poped Flag and every initial element of our clusters, then add it to the cluster where the distance was shortest.

This gives us what we originally thought to be a biclustered set, but appears instead to be a simple clustered array. Our next step is to take these clusters and, for each flag's atribute, get its difference with every other flag's attributes in the cluster. This will allow us to find subsets where at least 3 attributes are identic. Once a subset is created, the algorithm looks in other clusters if other flags respect this subset's rules and adds them to the subset.

All these subset represent our biclustered data.

Issues Encountered

The main issue we encountered, as mentionned in the previous part, was to understand the concept of Biclustering. For most of the project, we believed that biclustering was the idea of grouping flags based on multiple criteria. This later appeared to be false, and is just a normal

clustering strategy. Other than that, we encountered technical issues regarding the selection of the initial Flags to add to each cluster, which we later overcame thanks to the idea of finding patterns of equidistant flags.

Results

Currently, as we haven't fully implemented the algorithm resulting from our corrected understanding of biclustering, our results are invalid. By choosing a 6 clusters and a removal rate of 20% for off-centered flags, the results are as follow :

```
Distance Done Time : 254
Sorting and Median Done Time : 40
First Clusters Done Time : 49
Final Biclustering done time : 29
[Mongolia, USSR, Romania, Malaysia, Laos, Kampuchea, French-Polynesia]
[Montserrat, US-Virgin-Isles, Tuvalu, Turks-Cocos-Islands, St-Vincent, St-
Helena, Niue, Hong-Kong, Fiji, Falklands-Malvinas, Dominica, Cayman-Islands,
British-Virgin-Isles, Bermuda, Belize]
[Morocco, Yugoslavia, Vietnam, USA, UAE, Turkey, Tunisia, Trinidad-Tobago,
Togo, Thailand, Taiwan, Syria, Surinam, Sudan, Spain, South-Yemen, South-
Africa, Singapore, Sierra-Leone, Seychelles, Senegal, Saudi-Arabia, Rwanda,
Puerto-Rico, Portugal, Peru, Panama, Pakistan, Oman, North-Yemen, Nigeria,
Niger, Mexico, Mauritius, Mauritania, Mali, Maldive-Islands, Madagascar, Libya,
Liberia, Lebanon, Kuwait, Japan, Ivory-Coast, Ireland, Iraq, Iran, Indonesia,
Hungary, Haiti, Guinea, Grenada, Greenland, Ghana, Gambia, Ethiopia, Egypt,
Ecuador, Djibouti, Congo, Comorro-Islands, Colombia, China, Chile, Chad, Cape-
Verde-Islands, Canada, Cameroon, Burundi, Burkina, Brazil, Bolivia, Benin,
Bangladesh, Bahrain, Austria, Algeria, Albania]
[Mozambique, Zimbabwe, Zambia, Zaire, Venezuela, Vanuatu, Uganda, Tanzania,
Swaziland, St-Kitts-Nevis, Sri-Lanka, Sao-Tome, Paraguay, Papua-New-Guinea,
Malawi, Lesotho, Kenya, Jordan, Jamaica, Guyana, Guinea-Bissau, Germany-FRG,
Germany-DDR, Gabon, Equatorial-Guinea, Central-African-Republic, Bulgaria,
Brunei, Angola, Afghanistan]
[Malta, Vatican-City, UK, Tonga, Switzerland, South-Korea, Poland, Norway,
Monaco, Liechtenstein, Italy, Gibraltar, French-Guiana, France, Faeroes,
Denmark, Czechoslovakia, Bhutan, Belgium, Andorra]
[Marianas, Western-Samoa, Uruguay, Sweden, St-Lucia, Somalia, Solomon-Islands,
San-Marino, Qatar, Philippines, North-Korea, Nicaragua, New-Zealand,
Netherlands-Antilles, Netherlands, Nepal, Nauru, Micronesia, Luxembourg,
Kiribati, Israel, India, Iceland, Honduras, Guatemala, Guam, Greece, Finland,
El-Salvador, Dominican-Republic, Cyprus, Cuba, Costa-Rica, Cook-Islands, Burma,
Botswana, Barbados, Bahamas, Australia, Argentina, Antigua-Barbuda, Anguilla,
American-Samoa]
```

However, we know that we are missing a second part of the algorithm, which is finding similar subsets and creating them. We hope to have this done before the presentation in order to display our progression