GROUP 2

GOLDMINERS

INSTALLATION AND SETUP

Installation of Armadillo and mlpack and other required packages is needed for this implementation.
The following are the commands for their installation on Linux

```
$ apt install libmlpack-dev mlpack-bin libarmadillo-dev
$ sudo apt-get install libblas-dev liblapack-dev
$ apt-get install libatlas-base-dev
```

Instructions for compilation and executing the code

```
$ g++ filename.cpp -larmadillo -lmlpack
$ ./a.out
```

The code can be found on this github repository
https://github.com/yemaedahrav/CS568_GoldMiners

CODE DOCUMENTATION

ALGORITHM

The main file containing the implementation of the algorithm is the `partition.cpp` which reads input from a single file `data.in`.

There is a **spectral_coclustering** class which inherits from **base_clustering** class.
The **base_clustering** class implements the **SVD** functions according to whether the input matrix A is sparse or not.

After the **SVD**, left and right singular vectors U and V are obtained from which Z_2 and **Kmeans** is applied on it to obtain the clusters.
The values are stored in the assignments of dense row vector data type

The **spectral_coclustering** class implements the **fit()**, **get_assignments()**, **print_assignments()** which consists of the clusters produced by the algorithm.

Along with these, **isZero()**, **isSparse()** are used for checking the sparsity of the matrix.
The **scale_normalize()** function applies scale normalization on the input

INPUT FORMAT AND PREPROCESSING

The word definition files `MED.terms, CRAN.terms and CISI.terms` contain the original words and their global weights.
The given input files are in the HarwellBoeing sparse matrix format, the `converter.cpp` file is used to convert those into Matrix Market format.

When combining documents and words from two different 'classes/labels' we need to create a common collection of words by taking the union of the words from both sets.
The `combine_words.cpp and combine_words_three.cpp` do this task as the store all the words uniquely by using an map STL container

For testing on real datasets we need to combine these Medline, Cranfield and Cisi datasets and generate MedCisi, MedCran and MedCranCisi datasets
The `combine_matrix.cpp` and `combine_matrix_three.cpp` generate these datasets. Two files are generated for each dataset `input.in` and `labels.in`

EVALUATION

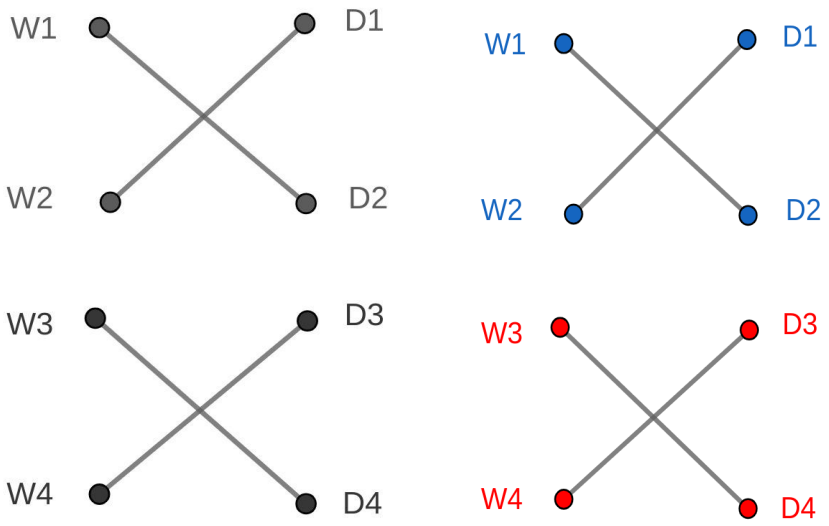
RESULTS ON DUMMY DATA

Dummy datasets contain a small 4x4 matrix and the algorithm clusters it into two clusters/ bipartitioning of the graph.

The results are very good on this toy data

```
milind@Workstation73:~/data_mining/new_files$ ./a.out
The cluster assignments are:
       0       0       1       1       0       0       1       1

The confusion matrix for the obtained clustering is:
       4       0
       0       4
```



The given input graph, the bipartitioning is denoted by the colour, which is computed by the algorithm

COMPARING RESULTS FROM ACTUAL DATASETS USED BY THE ORIGINAL AUTHOR

1. **MedCisi**

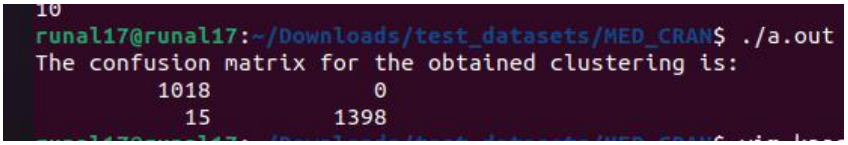
```
runal17@runal17:~/Downloads/test_datasets/MED_CISI$ ./a.out
The confusion matrix for the obtained clustering is:
          72        1460
         961           0
```

	Medline	Cisi
$\mathcal{D}_0$:	970	0
$\mathcal{D}_1$:	63	1460

$\mathcal{W}_0$: cells patients blood hormone renal rats cancer
$\mathcal{W}_1$: libraries retrieval scientific research science system book

Table 3: Bipartitioning results for MedCisi

2. MedCran

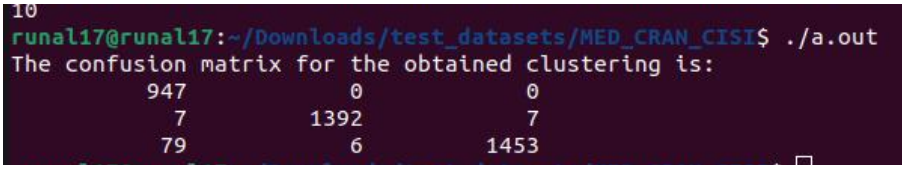


	Medline	Cranfield
$\mathcal{D}_0$:	1026	0
$\mathcal{D}_1$:	7	1400

$\mathcal{W}_0$: patients cells blood children hormone cancer renal
$\mathcal{W}_1$: shock heat supersonic wing transfer buckling laminar

Table 2: Bipartitioning results for MedCran

3. Classic3



	Med	Cisi	Cran
$\mathcal{D}_0$:	965	0	0
$\mathcal{D}_1$:	65	1458	10
$\mathcal{D}_2$:	3	2	1390

$\mathcal{W}_0$: patients cells blood hormone renal cancer rats
$\mathcal{W}_1$: library libraries retrieval scientific science book system
$\mathcal{W}_2$: boundary layer heat shock mach supersonic wing