Inka Simola
Knowledge discovery coursework part II

Documentation for command-line tool that uses hierarchical agglomerative clustering on specific 'Video Games Sales' data set (vgsales.csv, https://www.kaggle.com/gregorut/videogamesales)

1. FILES

hierarchical_agglomerative_clustering.py
libs/agglomerative.py

2. DATA MINING METHOD

Hierarchical agglomerative clustering is an unsupervised method for clustering data based on the proximity of individual data points to each other. The algorithm first calculates a distance matrix between all pairs of data points using a distance function. Then it converts each data point to a singleton and repeatedly merges the pair of singletons/clusters with the smallest separation into a cluster until there are C clusters left.

The distance function chosen for this implementation was Euclidean distance between three floating point attributes of a data point. For calculating the minimum distance between two clusters the average linkage method was used.

While hierarchical agglomerative clustering does not require a preset number of clusters, this algorithm uses cluster count as an ending criterion. Ideally a dendrogram would have been built for each value of C between zero and data set size, but this would have been impractical and computationally expensive.

Hierarchical agglomerative clustering results can be evaluated by using a cophenetic correlation coefficient (CPCC). During clustering, a cophenetic distance matrix is updated during each merge to reflect the distance at which two data points were first brought into the same cluster with each other. After clustering has finished, Pearson's product moment correlation coefficient is calculated between the distance matrix and the cophenetic distance matrix. The higher the value, the better the fit between the data and the clustering method used. By varying the value of C and the linkage method chosen, and optimal fit can theoretically be achieved.

Within the time constraints of the coursework schedule, implementing CPCC evaluation was however omitted for two reasons:

1) Computational time constraints. Cycling through several possible combinations of C values and three different linkage methods would have taken too long because of the size of the data set.

2) Evaluation results are most useful for deciding which of the linkage methods to use for calculating cluster distances. As only one of three linkage method variants was implemented, the value of implementing CPCC evaluation would have been limited.

3. DATA SET

The data set vgsales.csv contains 16600 rows of video game sales data. Each row corresponds to a game released between years 1984 - 2016. The unit used in the sales columns is 'millions of copies sold' and there are no missing values.

Index	Column name	Type	Usage in clustering
0	Rank	integer	
1	Name	string	
2	Platform	string	attribute (contains attribute values)
3	Year	integer or 'N/A'	attribute (contains attribute values)
4	Genre	string	attribute (contains attribute values)
5	Publisher	string	attribute (contains attribute values)
6	NA_Sales	float	x / y / z
7	EU_Sales	float	x / y / z
8	JP_Sales	float	x / y / z
9	Other_Sales	float	x / y / z
10	Global_Sales	float	x / y / z

4. IMPLEMENTATION

This data mining tool is a command line application written in Python that uses Matplotlib for visualizing the results. The program expects command line arguments and a specific data set (vgsales.csv) as input, and outputs at least one file. The first automatically created file is a csv file and contains a modified version of the input file with cluster labels added. The rest of the files are user-commissioned png files containing 3d plots of the data, with user-defined axis limits. See samples in results folder.

The implementation consists of the following stages:

Import stage:
- Checking for invalid command line arguments
- Data import from .csv file

Preprocessing stage:
- Columns converted to integers/ floats as per type description
- Data points with missing attribute values ('N/A') deleted if attribute used for narrowing data set
- Two additional columns added, for copy of Rank and cluster number
- Original rank values converted to singletons (=lists with one entry)

Data mining stage:
- Distances between all pairs of data points calculated and stored in a min heap (distance matrix)
- While number of clusters < C :
  - Pop new distance from min heap
  - Add contents of cluster/singleton b to contents of cluster/singleton a
  - Calculate new cluster centre for cluster a
  - Mark cluster/singleton b marked as used
  - Decrement cluster count by one
  - Report progress every 100 clusters

Postprocessing stage:
- Clusters labeled 1, 2, 3...
- Each data point given a cluster label
- CPCC evaluation is missing (see section 2)

- Results written to csv file with filename given by user
- Results plotted in png files with user-defined zoom values

5. COMMAND LINE ARGUMENTS

The program is run by typing:

python hierarchical_agglomerative_clustering.py

...followed by one of the following lists of command line arguments:

[output file name] [C] [x index] [y index] [z index]
[output file name] [C] [x index] [y index] [z index] [attribute] [attr_value]

- The indices of x, y, and z refer to the indices of the columns of vgsales.csv. Permitted values are 6-10.
- Attribute refers to the index of the column with the attribute used to narrow down the data set, e.g. Genre. Permitted values are 2-5.
- Attr_value refers to values in the attribute column, e.g. Puzzle. All values found in the attribute column are permitted.

When clustering is finished, the user is prompted for a positive decimal number that sets the maximum value for x, y and z coordinates in the visualization plot. The value refers to millions of copies sold. The data set maximum is 82.74 in Global_Sales for Wii Sports, the extreme outlier of the data set. Most data points have x, y and z values below 30.

6. USAGE EXAMPLES

To mine for...

20 clusters based on NA and JP sales of games released in 2007:
python hierarchical_agglomerative_clustering.py results.csv 20 6 6 8 3 2007

50 clusters based on JP sales and global sales on games for the Wii platform:
python hierarchical_agglomerative_clustering.py results.csv 50 8 8 10 2 Wii

7 clusters based on global sales of games from publisher Take-Two Interactive:
python hierarchical_agglomerative_clustering.py results.csv 7 10 10 10 5 Take-Two\ Interactive

100 clusters based on NA, EU and JP sales:
python hierarchical_agglomerative_clustering.py results.csv 100 6 7 8

(Beware: the last operation takes a while)

7. DISCLAIMER

Hierarchical agglomerative clustering is inherently slow and no optimization has been done on this piece of code. In the absence of CPCC evaluation and a farm of powerful computers the optimal cluster number for each mining task has to be approximated by examining the 3d plots and manually trying out different values for C. Consider narrowing down the dataset by platform, year, genre or publisher if clustering seems to be taking too long.