Technical Note on the Cluster Identification Algorithm

Taras Hrendash *CERGE-EI*

This document provides supporting information and technical details of functioning of the algortim programmed in R language.

Input data

This algorithm is implemented based on the Disambiguation and Co-authorship Networks of the U.S. Patent Inventor Database (1975 - 2010). This dataset identifies individual inventors from the U.S. utility patent database and features information necessary for cluster identification in a present setup which includes:

1. A unique identifier of individual inventor.
2. A unique identifier of collaboration team (i.e. a patent number).
3. Geo-coded location (longitude and latitude) of each inventor at a time of patent application.[1]

The source code was developed to use the `invpat.csv` file from the `invpat_final.zip` archive downloadable via the link above as a raw intput. Importantly, in order to match the required properties outlined above, one should use the subset of patents from the original database with a single year of application. For illustration purposes, the dataset is also restricted to a single US Patent Class. It is, however, possible to implement the cluster identification algorithm by running the R code based on alternative input file featuring similar data.[2]

Content of the archive

The R code running through the algorithm and visualization of the results is split into the following parts contained in separate `.R` files:

1. `global.R` sets the working directory, installs and loads the required packages[3]. To list the most important of them:

   1.1. `data.table` is used for data manipulation;

   1.2. `igraph` is used for network construction and manipulation;

   1.3. `ggplot2`, `ggmap` and `GGally` are used for visualization of the outputs (maps of clusters and networks of linkages within clusters).

[1] Geographical location is supposed to be uniquely defined for each individual inventor, thus, an appropriate example of the dataset that can be used to implement the cluster identification algorithm would be a subset of the above-mentioned database at a given application year.

[2] The R code refers to the names of variables from the original input file. It might be necessary to adjust the naming of variables, if alternative dataset is used.

[3] On its first run, this file should be manually edited and run separately to set the appropriate working directory (where the content of archive is extracted to) and install required packages. Later on, this file is called internally from the main part of the code and should not be run separately.

2. `prepare.R` pre-processes the raw input `invpat.csv` file[4] from the `/data` directory and saves the working subset in `.RData` format[5].

3. `functions.R` contains user-defined functions used in the main part of code[6].

4. `run.R` contains the main part of the R code used to implement the cluster identification algorithm. It consists of 4 sections:

   4.1. **Section 0** reduces the entire original patent database to the subset of patents that belong to a given US Patent Class and were filed in a given application year. The output subset is stored as `invpat_sub.RData` file in the `/data` directory.[7]

   4.2. **Section 1** performs a construction of the collaboration layer (network) of the search space for cluster identification algorithm. Particularly, for each team identifier (i.e. a patent number) all possible pairs of inventors are created and used to construct an undirected and unweighted network. The output `igraph` object comprising a network is stored as `collaboration_network.RData` file in a dynamically created `/aux` directory.

   4.3. **Section 2** performs a construction of the geographical layer (network). For simplification, the unique locations are used as nodes in this network. The matrix of pairwise distances between all locations is created. The distance matrix is converted to multiple adjacency matrices for each value in the range of distance thresholds. The latter finnaly are used to create the `igraph` objects comprising geographical networks. They are not, however, stored as they do not directly convey the information about partitioning of inventors, but rather locations, into spatially dense groups. The lists of corresponding inventor codes partitioned into groups are stored as `geo_components_*.RData` files in dynamically created `/geocomponents` directory.

   4.4. **Section 3** implements the cluster identification algorithm itself, which is described in details below.

5. `plot.R` is used for visualization of identified clusters.

Definition of terms

Component is a maximal subnetwork (i.e. it is not possible to add any other node to the subnetwork without violating the following condition) such that every pair of nodes in the subnetwork is connected by a path (sequence of edges). On the one extreme, a separated node is a component of size one. On the other extreme, the entire network can be a component if the latter condition is satisfied for every pair of its nodes.

There are two types of components used in the algorithm:

1. *connected component*, i.e. component at the collaboration layer;

2. *geo-component*, i.e. component at the geographical layer.

[4] This file is not included to the distributed archive because of the large size. In order to run the `prepare.R` code, it is necessary to download the original dataset from the link above and place it to the `/data` directory.

[5] The output subset `invpat.RData` is already included to the distributed archive. Running of the `prepare.R` code is optional.

[6] This file is called internally from the main part of the code whenever it's content is essential for running the program.

[7] Running this section is optional and is intended for changing the subsetting parameters, i.e. US Patent Class and/or application year.

There are two types of R objects that are used to represent components (effectivelly, to partition the network) throughout the cluster identification algorithm[8]:

1. list of vectors, where each vector consists of inventor codes belonging to a separate component
2. `igraph` object with an attribute 'membership', which is a named vector of length equal to the total number of nodes in the network, where each element's name is an inventor's code and each element's value is an ID of the component to which a given inventor belongs.

For example, consider the following network:

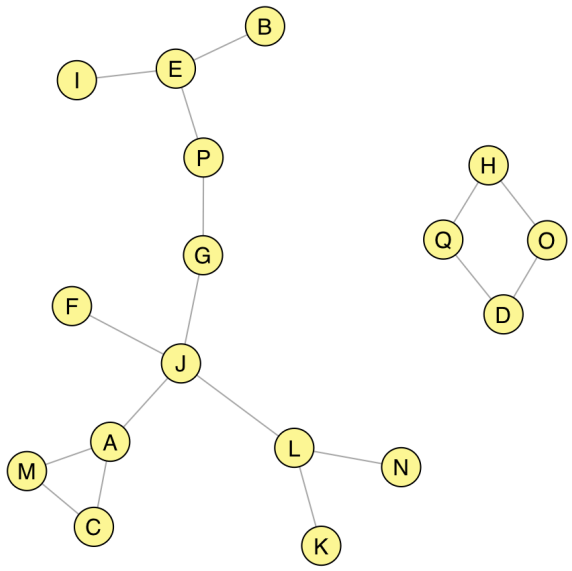

Figure 1: Network components

According to the graph, there are two components (see the definition above) that can be defined in the network. The two different ways of representation would, thus, looks as follows:

```
> components_igraph
$membership
A B C D E F G H I J K L M N O P Q
1 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 2

> components_list
[[1]]
 [1] "A" "B" "C" "E" "F" "G" "I" "J" "K" "L" "M" "N" "P"

[[2]]
[1] "D" "H" "O" "Q"
```

[8]Distinction between the two types of R objects is purely technical and has no conceptual link to the steps of the cluster identification algorithm described below.

Cluster identification algorithm

As it was mentioned earlier, **Section 3** of the `run.R` code implements the cluster identification algorithm which itself involves the following conceptual stages:

1. In the geographical network, find its components (*geo-components*), i.e. groups of inventors where the distance to the nearest neighbor within a group does not exceed the distance threshold.

2. For each *geo-component*, create an induced subnetwork of the collaboration network constituting all 'members' of a *geo-component* and collaboration ties among them and split it into *connected components*.

3. If there is more than one *connected component* per *geo-component*, for each *connected component* find its corresponding *geo-components*, otherwise consider a candidate from the Step 2 as a technology cluster identified at the first iteration.

4. Iterate Steps 2 and 3 until all *geo-components* entirely coincide with *connected components*.

To trace these steps in the code, note that the Step 1 is implicitly implemented in the **Section 2** of the `run.R` code. Iterations of Steps 2 and 3 are implemented inside the `for` loop over the input *geo-components* between line `142` and `176` in the **Section 3** of the `run.R` code and can be depicted with the following diagram:

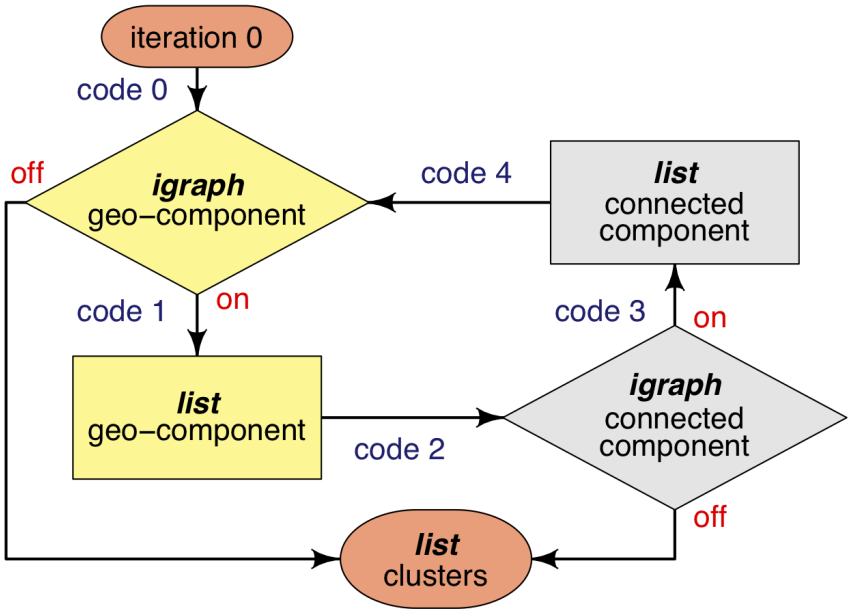

Figure 2: Algorithm iterations

Effectively, there is no preset number of iterations to be run, so the Steps 2 and 3 are repeated inside the `while` loop with a terminating condition specified according to the Step 4 above.

For technical convenience, initial iteration is separated out of the main `while` loop and is implemented with the `code 0` chunk.

Consequently, Step 2 is implemented with `code 1` and `code 2`; Step 3 with `code 3` and `code 4` chunks. Note that `code 1` and `code 3` chunks are auxiliary and are used to convert the components

from one *form of representation* to another (see the example above), whereas `code 3` and `code 4` chunks are conceptually related to the Steps 2 and 3 of the algorithm and are used to split one *type* of components into a set of components of another *type*.

As an ultimate output, code `run.R` and its **Section 3** in particular produce the lists of clusters (groups of inventor codes) for each pair of the size and distance thresholds stored as `clusters_*_*.RData` files in dynamically created `/clusters` directory.

Visualization of clusters

For visualization of identified clusters refer to the `plot.R` file which should be executed independently of the main `run.R` code. The code involves the following steps:

1. For a given combination of size and distance thresholds, which should be manually set in the code, load the list of clusters (character vectors of inventor codes) from a corresponding `/clusters/clusters_*_*.RData` file and convert it to `data.frame` with a column of inventor codes and a column of cluster IDs corresponding to the former.

2. Add geographical coordinates of inventors belonging to the clusters by merging the `data.frame` created on the previous step with the input dataset `/data/invpat_sub.RData`.

3. For each cluster, create an induced subnetwork of the collaboration network and construct the `data.frame` of coordinates for the endpoints of all adges in this subnetwork.

4. For each cluster, plot the locations of inventors belonging to the cluster represented by the circles of the radius equal to the distance threshold, overlay the former graph with the net of pairwise collaboration ties among inventors (locations) within the cluster.

5. For each cluster, plot the structure of all collaboration ties within the cluster using the Fruchterman-Reingold layout algorithm.

Output plots for the Step 4 and 5 are stored in `.png` format in dynamically created directories `/maps` and `/networks` respectively.