Using *paraminerLC*

Dominique Bouthinon

L.I.P.N, UMR-CNRS 7030, Université Paris-Nord,
93430 Villetaneuse, France
dominique.bouthinon@lipn.univ-paris13.fr

1 Introduction

paraminer [1] is a generic and parallel algorithm for closed patterns mining that solves three tasks : mining frequent itemsets, mining frequent connected relational graphs and mining gradual itemsets. *paraminerLC* adapts *paraminer* code and structures to mine frequent local closed patterns [2] associated to k-communities [3].

paraminerLC is integrated in a work made with Henry Soldano and Guillaume Santini (henry.soldano, guillaume.santini@lipn.univ-paris13.fr) [2].

2 An example

Let us illustrate the use of *paraminerLC* to find local patterns and related local knowledge in an attributed graph. Let us consider a musical tastes dataset D [4] containing 18 subsets of {*rock, folk, pop, blues, jazz*}. Each subset corresponds to the musical tastes of one person. Moreover, we have friendship relations between these 18 persons.

The dataset and the friendships relations can be represented by an attributed graph G, where each vertex corresponding to the name of one person and the label associated with the vertex represents the musical tastes of this person (see figure 1). The vertices are connected when the corresponding persons are friends.

In this case study we consider the graph G_T, derived from G, where the vertices are the triangles found in G. So a vertex in G_T represents a friendship relation shared by 3 persons. The label associated with a triangle is the intersection of the labels of the persons involved in the triangle, so this label represents their common musical tastes. Two triangles of G_T are connected when they share 2 persons (see figure 2).

Let us call D_T the data set whose each element corresponds to the label associated with one triangle of G_T. Let p be a pattern then $ext_{_T}(p)$ is the *support set* of p in D_T, that is the set of triangles whose the description contains p. For instance $ext_{_T}(\{folk, jazz, rock\}) = \{ABD, ABC, BCD, ACD\}$.

The goal of *paraminerLC* is to find *local* closed patterns from (frequent) closed patterns found in the data set D_T. A local closed pattern takes into

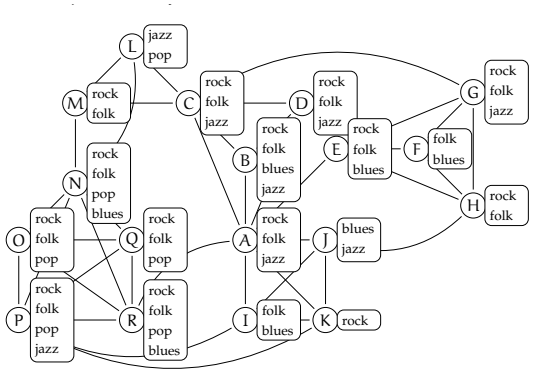

Fig. 1 . Graph G representing the friendship relations and the musical tastes of a group of persons.

account the relationships between the persons. For instance, suppose that we have found the closed pattern $\{rock, folk\}$ in D_T. This pattern induces the subgraph $G_T(ext_T(\{folk, rock\}))$ made of all coloured vertices and edges figure 2. We see that $G_T(ext_T(\{folk, rock\}))$ has two connected components (in red and green figure 2, called *triangles components*, involving at least 4 persons. We define a 3-community as the set of persons involved in a triangles component, for instance the green triangles component leads to the 3-community $ABCD$.

Each 3-community has a local closed pattern, that is the greatest description shared by the persons inside the community. So, $\{rock, folk, jazz\}$ is the local closed pattern of the 3-community $ABCD$. This local closed pattern derives from the closed pattern $\{rock, folk\}$, this is why *paraminerLC* outputs the triple ($\{rock, folk\}$, $ABCD$, $\{rock, folk, jazz\}$). It corresponds to the local implication rule $ABCD\{rock, folk\} \rightarrow ABCD\{jazz\}$. This local rule add some knowledge specific of this group of friends : when we consider persons that like $rock$ and $folk$, in this 3-community they also like $jazz$.

3 ParaminerLC

paraminerLC is written in $C{+}{+}11$ and needs the *boost/graph* library installed.

3.1 Install *boost/graph*

The instructions to install *boost/graph* are given at `http://www.boost.org/doc/libs/1_59_0/libs/graph/doc/index.html` or `http://sourceforge.net/projects/boost/?source=typ_redirect`.

Once *boost/graph* is installed you must define an environment variable BOOST set to the path to the directory containing *boost*. For instance for Linux systems

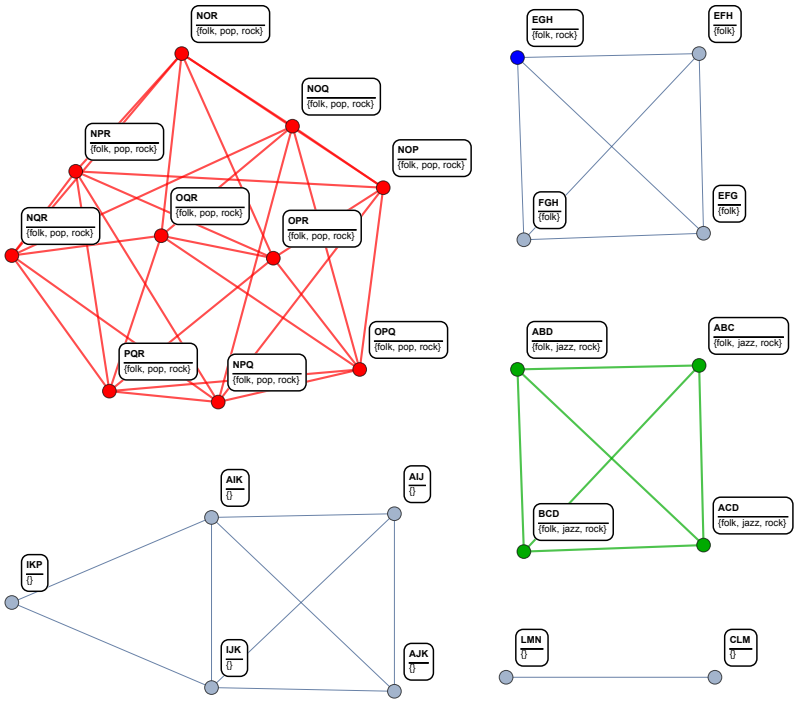

Fig. 2 . Graph G_T representing the triangles derived from G and their connections.

put the following instruction in your *.bashrc* (adapting the actual path to boost, here we assume that *boost* has been installed at */usr/local/boost_1_57_0'*) :

```
export BOOST='/usr/local/boost_1_57_0'
```

3.2 Install *paraminerLC*

Once you have extracted the *paraminerLC.zip* file you obtain the following directories :

- *paraminerLC_downlad* : it contains the following sub-directories
  - *doc* : it contains *paraminerLC_doc.pdf* (the file you are reading).
  - *data* : it contains example files that can be used as input data files of *paraminerLC*
  - *src* : it contains all source programs needed to produce the executable program.

3.3 Compile *paraminerLC*

You can easily compile the program from sources located in the directory *src*: simply type *make* to obtain an executable file *paraminerLC* (your system must have the gnu *g++* compiler, you can edit the *makefile* in the directory *src* to adapt it with anoher compiler).

3.4 Run *paraminerLC*

To run *paraminerLC* type

```
paraminerLC <data_file> <frequency> <community_threshold>
```

<*frequency*> is the minimal frequency of the closed patterns you will consider on your dataset.

<*community_threshold*> is the minimal number of objects inside a community.

<*data_file*> is the name of the file that contains the data in a specific format described next section.

For instance when you type

```
paraminerLC ../data/mougel.dia 3 5
```

paraminerLC will output quintuplets (c, n, t, e, l) where :

- c is a closed pattern (present in at least 3 triangles),
- n is the size of the component of triangles we derived from c,
- e is the communauty (of at least 5 persons) associated with the triangles component,
- t is the first triangle (as a triplet of persons) of the triangles component,
- l is the local closed pattern of the communauty e.

Consider for instance the quintuplet

```
{"rock", "folk"}, {"N", "O", "P", "Q", "R"}, 10, {"N", "P", "R"}, {"rock", "pop", "folk"}
```

means that from the closed pattern {"$rock$","$folk$"} we derived a connected component of 10 triangles whose first triangle is made of appexes {"N","P","R"}, the communauty associated with this component is {"N","O","P","Q","R"}, and this communauty has a local closed pattern {"$rock$","pop","$folk$"} .

4 Input data format

A data file contains several sections separated by $\#------$. Each section contains data as illustrated in the following example (comments are not in the file) :

```
# ----- fichier
data/mougel.dia             /* name of the file */
# ----- objets
A B C D...                  /* names of the persons involved in the triangles */
# ----- langage
blues folk jazz pop rock    /* items describing the persons */
# ----- description         /*  descriptions of the N triangles in terms of persons */
                              /* (numbers refer to persons) */
0 2 3                       /* first triangle is made of the persons A, C and D */
1 2 3                       /* second triangle is made of the persons B, C and D */
...
# ----- itemset           /* descriptions of the N triangles in terms of items */
                          /* (numbers refer to items) */
1                         /*  first triangle : {folk} */
0 1 2 3                   /*  second triangle :  {blues, folk, jazz, pop} */
...
# ----- adjacences        /* descriptions of the adjacencies between the N triangles */
                          /* (numbers refer to triangles) */
1 3                       /* first triangle is adjacent to the second and fourth */
0 2                       /* second triangle is adjacent to the first and third  */
...
#------end
```

References

1. Negrevergne, B., Termier, A., Rousset, M.C., Méhaut, J.F.: Paraminer: a generic pattern mining algorithm for multi-core architectures. Data Mining and Knowledge Discovery (2013) 1–41

2. Soldano, H., Santini, G., Bouthinon, D.: Local knowledge discovery in attributed graphs. In Esposito, A., ed.: 27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI, Vietri sul Mare, Italy, IEEE Computer Society (November 9-11 2015) to appear
3. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043) (Jun 2005) 814–818
4. Mougel, P.N.: Finding homogeneous collections of dense subgraphs using constraint-based data mining approaches. PhD thesis, Lyon, INSA (2012)