Graph Partitioning with AMPL

Antonio Mucherino

Laboratorie d'Informatique, École Polytechnique

Course on Operations Research (ISC610A)

Semester I - 2008/09

TD5 - December 4^{th} 2008

Recalling some definitions: Clustering

We already know what a clustering problem is.

- Let X be a set of samples whose partition is unknown.
- Let us suppose that there is no previous knowledge about the data (no training set is available).

Definition

Clustering is aimed at finding a partition $\{C_1, C_2, \ldots, C_K\}$ of the set of data, such that

$$X = \bigcup_{i=1}^{K} C_i, \quad \forall i, j | 1 \leq i < j \leq K \quad C_i \cap C_j = \emptyset.$$

- Each cluster represents a subset of features of the samples that it contains.

Recalling some definitions: Graph

We already know what a graph is.

Definition

A *graph* is an ordered pair $G = (V, E)$ comprising a set V of vertices or nodes together with a set E of edges or links, which are 2-element subsets of V.

- **Undirected graph**: a graph in which edges have no orientation.
- **Directed graph** or **Digraph**: a graph $G = (V, A)$, where A is a set of *ordered* pairs of vertices, even called arcs or directed edges.
- **Weighted graph**: a graph in which numbers (weights) are assigned to each edge. It can be *directed* and *undirected*. It is denoted by $G = (V, E, w)$ or $G = (V, A, w)$, where w represents the weights.

Recalling some definitions: Graph partitioning

Definition

Graph partitioning is the clustering problem of finding a suitable partition of a set of data represented through a graph G.

- Each cluster is a subgraph of the graph G, i.e. a subset of its vertices.
- Intuitively, the best partition is the one that separates sparsely connected dense subgraphs from each other.
- sparsely connected: the number of edges between vertices belonging to *different* clusters is minimal.
- dense: the number of edges between vertices belonging to *the same* cluster is maximum.

Formulating an optimization problem

How can we solve a graph partitioning problem?

- We need to find a partition in clusters of a weighted undirected graph $G = (V, E, c)$, where
  - V is the set of vertices of G,
  - E is the set of edges of G,
  - c is the set of weights eventually assigned to the edges.
- This problem can be formulated as a global optimization problem.
- We want the number of edges between vertices belonging to different clusters to be minimal.
- Therefore, we need to solve a minimization problem, subject to a certain number of contraints.
- We will solve this problem by CPLEX/AMPL.

Parameters and Variables

Parameters

- V, set of vertices of G
- E, set of edges of G
- c, set of weights of G
- K, number of desired clusters in the partition

Variables

- x_{uk}, binary, indicates if the vertex u is contained into the cluster $k \leq K$:

$$x_{uk} = \begin{cases} 1 & \text{if } u \in k^{th} \text{ cluster} \\ 0 & \text{otherwise} \end{cases}$$

ÉCOLE POLYTECHNIQUE

Objective function

What do we need to minimize?

- We want the total weights of the edges between different clusters to be as minimum as possible:


Think it out: you should be able to give an answer within 1 minute!

Objective function

What do we need to minimize?

- We want the total weights of the edges between different clusters to be as minimum as possible:

$$\min \frac{1}{2} \sum_{k \neq l \leq K} \sum_{(u,v) \in E} c_{uv} x_{uk} x_{vl}$$

Think it out: you should be able to give an answer within 1 minute!

Constraints

Constraint I

- Each vertex must be assigned to only one cluster:

$$\forall u \in V \quad \sum_{k \leq K} x_{uk} = 1$$

Constraint II

- The trivial solution (all the verteces into one cluster) must be excluded:

$$\forall k \in K \quad \sum_{u \in V} x_{uk} \geq 1$$

Constraints

Constraint III (in general, optional)

- Each cluster cannot exceed a certain cardinality:

$$\forall k \leq K \quad \sum_{u \in V} x_{uk} \leq C$$

Constraint IV (in general, optional)

- Vertices having different color cannot be clustered together:

$$\forall u \neq v \in V, k \neq l \leq K, x_{uk} x_{vl} \leq \gamma_{uv}$$

where

$$\gamma_{uv} = \begin{cases} 1 & \text{if } u \text{ and } v \text{ have the same color} \\ 0 & \text{otherwise} \end{cases}$$

Constraints

Constraint V (in general, optional, substitutes Constraint II)

- Empty clusters can be controlled:

$$\forall k \leq K \quad \sum_{u \in V} x_{uk} \geq z_k$$

where

$$z_k = \begin{cases} 1 & \text{if cluster } k \text{ is not empty} \\ 0 & \text{otherwise} \end{cases}$$

The term

$$\sum_{k \leq K} z_k$$

can be added to the objective function, in order to require the minimum possible number of clusters, by forcing some of the K clusters to be empty.

ÉCOLE POLYTECHNIQUE

Writing the model in AMPL

You have 20 minutes for writing the discussed model in AMPL.

Remember that:

- the term that controls the number of clusters must be added to the objective function.
- all the 5 constraints must be included in the model.
- if you don't remember all the details about the model, go on **www.antoniomucherino.it** and download the slides of the lecture held on November 20th.

Some observations (1/2)

The objective function contains a product of binary terms.
How do we handle that?

- We introduce a new variable w_{ukvl} representing the product of the two binary variables.
- We substitute the products with the new variable w_{ukvl} everywhere, as for example in the objective function:

$$\min \frac{1}{2} \sum_{k \neq l \leq K} \sum_{(u,v) \in E} c_{uv} w_{ukvl}$$

- We add linearization constraints:

$$\forall u \in V, v \in V, l \in K, k \in K \;:\; (u,v) \in E \text{ or } (v,u) \in E$$

$$w_{ukvl} \leq x_{uk} \qquad w_{ukvl} \leq x_{vl} \qquad w_{ukvl} \geq x_{uk} + x_{vl} - 1$$

Some observations (2/2)

We need to choose a maximum cardinality C for the constraint III:

$$\forall k \leq K \quad \sum_{u \in V} x_{uk} \leq C$$

What value can we give to C?

One possible choice is:

$$C = \lceil \frac{|V|}{2} \rceil.$$

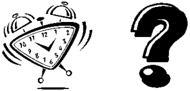
Note that, in AMPL, we can write the constraint as:

```
subject to cardinality {k in K} :
sum{v in V} x[v,k] <= ceil(card{V}/2);
```

ÉCOLE POLYTECHNIQUE

Writing the model in AMPL

You have other 10 minutes for writing the model in AMPL.

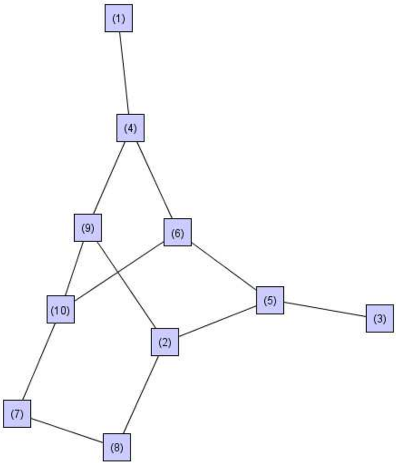


Remember that:

- you need to add a variable w that substitutes the product of binary variables.
- in the third constraint, you need to define a certain maximum cardinality C of the clusters.
- if you don't remember other details about the model, go on **www.antoniomucherino.it** and download the slides of the lecture held on November 20^{th}.

A random graph

This is a random graph.



random.dat

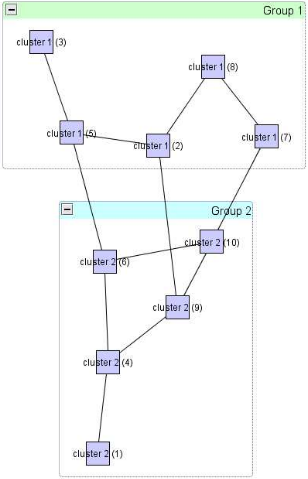
```
# AMPL dat file "random.dat"

param n := 10;  # number of vertices
param m := 12;  # number of edges/arcs

# graph is undirected
# edge : cost indicator
param : E : c I :=
  4 9 1 1
  6 10 1 1
  7 10 1 1
  2 8 1 1
  8 7 1 1
  2 5 1 1
  2 9 1 1
  9 10 1 1
  4 1 1 1
  5 3 1 1
  6 5 1 1
  4 6 1 1
;

param lambda :=
1 1
2 2
......
10 10
;
```

clustering.run

```
# clustering.run

# model:
model clustering.mod;

# data:
data random.dat;
##data Zachary.dat;
##data proogle.dat;

# maximum number of clusters
let kmax := 2;

# solver:
option solver cplex;

# solving the problem
solve;

# printing the result
display x;
```

Finding two subgraphs

By using your model, are you able to find this clustering?



The model: clustering.mod (1/2)

```
# clustering.mod
# model for graph partitioning

param n >= 1, integer;  # number of vertices
param m >= 1, integer;  # number of edges
set V := 1..n;
set E within {V,V};

# edge weights
param c{E};  # edge weights
param I{E};  # edge inclusions

# vertex colours
param lambda{V};
param gamma{u in V, v in V : u != v} :=
  if (lambda[u] = lambda[v]) then 0 else 1;

# max number of clusters
param kmax default n;
set K := 1..kmax;

# original problem variables
var x{V,K} binary;
# linearization variables
var w{V,K,V,K} >= 0, <= 1;
# cluster existence variables
var z{K} binary;
```

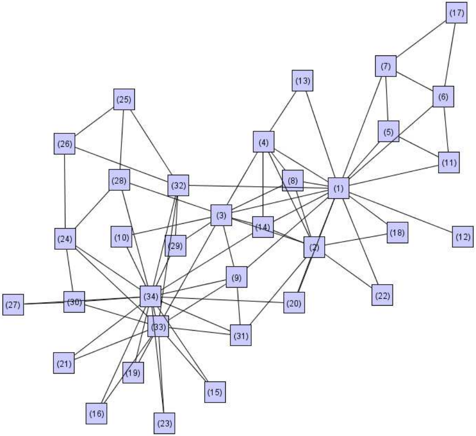
The model: clustering.mod (2/2)

```
# model
minimize intercluster :
  sum{k in K, l in K, (u,v) in E : k != l} I[u,v] * c[u,v] * w[u,k,v,l] +
  sum{k in K} z[k];

# constraints
subject to assignment {v in V} : sum{k in K} x[v,k] = 1;
subject to cardinality {k in K} : sum{v in V} x[v,k] <= ceil(card{V}/2);
subject to existence {k in K} : sum{v in V} x[v,k] >= z[k];
subject to diffcolours {u in V, v in V, k in K, l in K : u != v and k != l} :
  w[u,k,v,l] <= gamma[u,v];

# linearization constraints

subject to lin1 {u in V, v in V, h in K, k in K : (u,v) in E or (v,u) in E} :
  w[u,h,v,k] <= x[u,h];

subject to lin2 {u in V, v in V, h in K, k in K : (u,v) in E or (v,u) in E} :
  w[u,h,v,k] <= x[v,k];

subject to lin3 {u in V, v in V, h in K, k in K : (u,v) in E or (v,u) in E} :
  w[u,h,v,k] >= x[u,h] + x[v,k] - 1;
```

ÉCOLE POLYTECHNIQUE

The Zachary graph

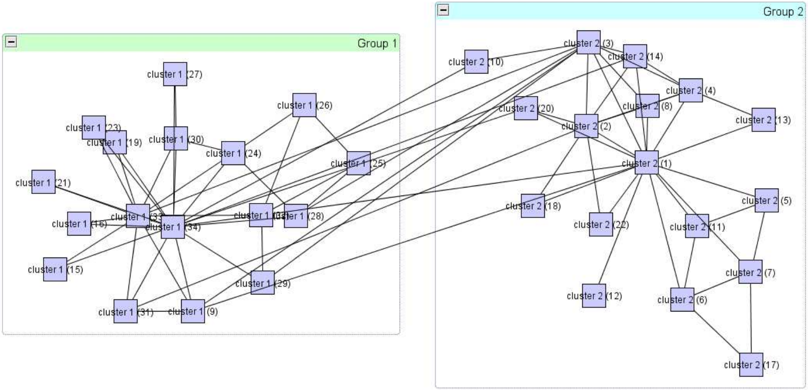
Represents the social communications between members in a karate club.



Download the dat file from www.antoniomucherino.it

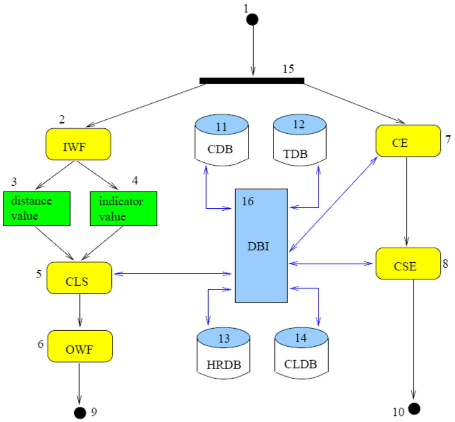
Finding two subgraphs

Are you able to find this clustering?



Proogle project

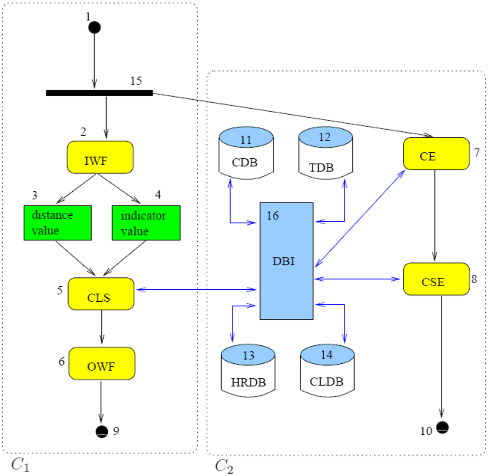
This is the software diagram of the Proogle project.



Download the dat file from www.antoniomucherino.it

Finding two subgraphs

Are you able to find this clustering?



The proposed exercises can be downloaded from:

www.antoniomucherino.it