Generalized Median Graph Estimation based on Block Coordinate Descent

Nicolas Boria, Sébastien Bougleux, Benoit Gaüzère and Luc Brun

Normandie Univ, ENSICAEN, UNICAEN, CNRS, GREYC, 14000 Caen, France
Normandie Univ, INSA ROUEN, LITIS 76000 Rouen, France

May 3rd 2019

1. Preliminaries
   - Motivation
   - The concept of edit-distance
   - Set-Median and Generalized Median Graph
2. The algorithm
   - General description
   - Initialization
   - Block coordinate descent
3. Experimental Evaluation
4. Conclusion and Perspectives

1. Preliminaries
   - Motivation
   - The concept of edit-distance
   - Set-Median and Generalized Median Graph
2. The algorithm
3. Experimental Evaluation
4. Conclusion and Perspectives

In classification and clustering, the concept of **prototype** or **representative** for a class of elements can be a core concept.

While a **mean** or **median** element can be rather easy to compute in vectorial spaces, its computation is a challenge in more complex spaces, such as the spaces of graphs.

We propose a method to compute approximate prototypes for sets of graphs.

Definition of GED

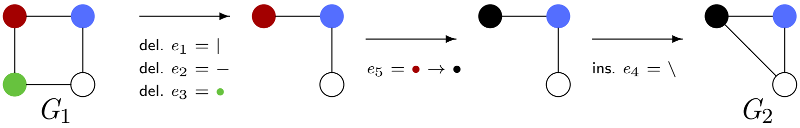
Assigning edit costs to various **edit operations** (inserting/deleting an element, changing a label) on graphs, the **Graph edit-distance** (GED) measures the minimal cost of an **edit path** between two graphs.

Example of edit path between two graphs G_1 and G_2

Computing the Graph Edit Distance is NP-Hard.

Definition of GED

Assigning edit costs to various **edit operations** (inserting/deleting an element, changing a label) on graphs, the **Graph edit-distance** (GED) measures the minimal cost of an **edit path** between two graphs.

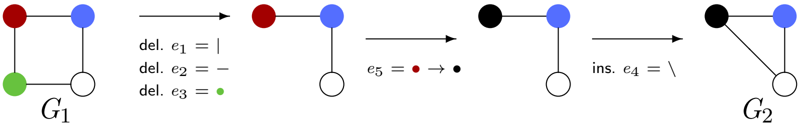


Example of edit path between two graphs G_1 and G_2

Computing the Graph Edit Distance is NP-Hard.

Definition of GED

Assigning edit costs to various **edit operations** (inserting/deleting an element, changing a label) on graphs, the **Graph edit-distance** (GED) measures the minimal cost of an **edit path** between two graphs.

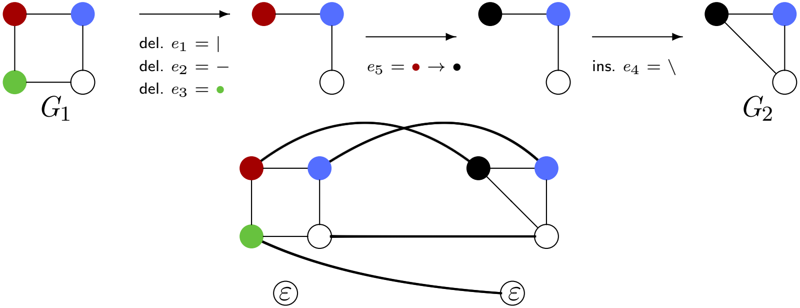


Example of edit path between two graphs G_1 and G_2

Computing the Graph Edit Distance is NP-Hard.

GED as Quadratic Assignment Problem

Any elementary Edit Path between $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ can be represented as an assignment between vertices of $V_1 \cup \{\varepsilon\}$ and $V_2 \cup \{\varepsilon\}$.



GED as Quadratic Assignment Problem

The problem of GED is thus interpreted as Quadratic Assignment Problem with Editions (QAPE) :

$$GED(G_1, G_2) = \min_x \{\frac{1}{2}x^\top \Delta x + \mathbf{c}^\top x\}$$

with cost matrices : Δ for edge assignment and $\mathbf{c}$ for node assignment.

Set-Median and Generalized Median Graph

Consider a set $S = \{G_1, G_2, ..., G_{|S|}\}$ of graphs, and $\mathbb{G}$ the space of all graphs.

The **set-median** graph is defined as :

$$G' = \arg\min_{G \in S} \sum_{G_i \in S} GED(G, G_i)$$

The **generalized median** graph is defined as :

$$\widetilde{G} = \arg\min_{G \in \mathbb{G}} \sum_{G_i \in S} GED(G, G_i)$$

1 Preliminaries

2 The algorithm
- General description
- Initialization
- Block coordinate descent

3 Experimental Evaluation

4 Conclusion and Perspectives

General description

Remind that the computation of $GED(G, G_i)$ is NP-Hard. Let $c(x_i, G, G_i)$ denote its approximation through the assignment x_i.

The algorithm initializes $\widetilde{G}$ by computing the set median :

$$\widetilde{G} = \arg\min_{G \in S} \sum_{G_i \in S} c(x_i, G, G_i) \tag{1}$$

Then, it iterates the two following minimizations until convergence :

$$\widetilde{G} \leftarrow \arg\min_{G \in \mathbb{G}_{\widetilde{n}}} \sum_{i=1}^{|S|} c(x_i, G, G_i) \tag{2}$$

$$\forall i \in \{1, \ldots, |S|\}, \quad x_i \leftarrow \arg\min_{x} c(x, \widetilde{G}, G_i) \tag{3}$$

General description

Remind that the computation of $GED(G, G_i)$ is NP-Hard. Let $c(x_i, G, G_i)$ denote its approximation through the assignment x_i.

The algorithm initializes $\widetilde{G}$ by computing the set median :

$$\widetilde{G} = \arg\min_{G \in S} \sum_{G_i \in S} c(x_i, G, G_i) \tag{1}$$

Then, it iterates the two following minimizations until convergence :

$$\widetilde{G} \leftarrow \arg \min_{G \in \mathbb{G}_{\widetilde{n}}} \sum_{i=1}^{|S|} c(x_i, G, G_i) \tag{2}$$

$$\forall i \in \{1, \ldots, |S|\}, \quad x_i \leftarrow \arg\min_{x} c(x, \widetilde{G}, G_i) \tag{3}$$

Initialization

In order to compute the set-median of S, the edit distance between all pairs of graphs in S must be approximated, so that an approximate sum of distances (SOD) can be computed for each graph of S.

Denoting by ALG the GED heuristic used to compute distances, and $C(\texttt{ALG})$ its complexity, the initialization phase requires $O(|S|^2 C(\texttt{ALG}))$ operations.

Block coordinate descent

We denote by $\widetilde{n}$ the order of graph $\widetilde{G}$ and by $\mathbb{G}_{\widetilde{n}}$ the space of graphs of order $\widetilde{n}$.

The first minimization in the block coordinate descent is :

$$\widetilde{G} \leftarrow \arg \min_{G \in \mathbb{G}_{\widetilde{n}}} \sum_{i=1}^{|S|} c(x_i, G, G_i)$$

This minimization can be solved analytically for most cost functions. Let us analyze the example of constant cost functions on symbolic graphs with labeled vertices and unlabeled edges.

Block coordinate descent

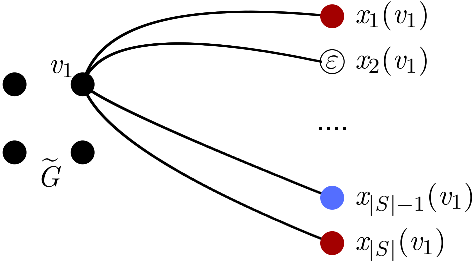
The number of vertices in $\widetilde{G}$ is fixed, and so are the assignments x_i between vertices of $\widetilde{G}$ and vertices of each G_i.

Given these two fixed parameters, we must decide :

- a label $l(v_j)$ for each vertex v_j of $\widetilde{G}$.
- whether the edge $(v_j, v_{j'})$ is part of $\widetilde{G}$ for each pair of vertices $(v_j, v_{j'})$ in $\widetilde{G}$.

Computing optimal vertex labels w.r.t. assignments

Each vertex v_j of $\widetilde{G}$ is assigned to one vertex $x_i(v_j)$ in each $G_i \cup \{\varepsilon\}$.



In order to minimize the edit-cost regarding vertices, each vertex v_i is given the most frequent label among its assigned vertices.

Computing optimal vertex labels w.r.t. assignments

Similarly, each pair of vertices $(v_j, v_{j'})$ of $\widetilde{G}$ is assigned to a pair of vertices $(x_i(v_j), x_i(v_{j'}))$ in each $G_i \cup \{\varepsilon\}$.

Let $S_{jj'}$ denote the set of graphs G_i of S where the edge $(x_i(v_j), x_i(v_{j'}))$ exists, and let c_{ed} and c_{ei} denote the constant edge-deletion and edge-insertion costs. The following rule applies regarding edges :

- the edge $(v_j, v_{j'})$ exists in $\widetilde{G}$ iff $(|S| - |S_{jj'}|)c_{ed} \leq |S_{jj'}|c_{ei}$

More involved analysis must led in the cases of labeled edges, but the optimal label can always be derived in polynomial time.

General block coordinate descent structure

$$\widetilde{G} \leftarrow \arg \min_{G \in \mathbb{G}_{\widetilde{n}}} \sum_{i=1}^{|S|} c(x_i, G, G_i)$$

update the structure and labels of $\widetilde{G}$ w.r.t. assignments x_i's. Done analytically.

$$\forall i \in \{1, \ldots, |S|\}, \quad x_i \leftarrow \arg\min_{x} c(x, \widetilde{G}, G^p)$$

update the assignments x_i's w.r.t. to the updated graph $\widetilde{G}$. Done heuristically.

repeat until convergence.

1 Preliminaries

2 The algorithm

3 Experimental Evaluation

4 Conclusion and Perspectives

datasets

Evaluated on two datasets :

- Monoterpenoides : 286 graphs divided in 8 classes. Symbolic labels on both vertices and edges.
- Letter(HIGH) : 2250 graphs divided in 15 classes. labels on vertices corresponding to positions in the plane

Two series of experiments were led :

- the first evaluates the SOD of solutions computed by the algorithm.
- The second evaluates the relevance of the set-median (SM) and generalized median (GM) as classifying tools.

SOD evaluation

First experiment : extracting 50 random vertices from each class in Letter, and 10 in Monotepernoides, the average SOD was extracted using different GED heuristics for phase 1 (initialization) and phase 2 (block coordinate descent).

Algorithms		Letter (HIGH)				Monoterpenoides			
1st phase	2nd phase	SOD SM	t(SM)	SOD GM	t(GM)	SOD SM	t(SM)	SOD GM	t(GM)
Bipartite	Bipartite	142.69	0.01	87.80	$6 * 10^{-4}$	402.50	0.002	253.11	$8 * 10^{-4}$
Bipartite	IPFP	142.87	0.013	87.61	0.003	398.01	0.002	128.45	0.179
IPFP	IPFP	135.99	0.057	87.22	0.003	202.75	0.162	104.11	0.136
mBipartite	mBipartite	142.04	0.014	89.47	$9 * 10^{-4}$	283.94	0.027	186.15	0.01
mBipartite	mIPFP	142.19	0.018	87.66	0.013	281.14	0.031	83.11	0.545
mIPFP	mIPFP	135.99	0.274	87.23	0.015	106.10	1.159	75.08	0.288

Table – SOD computed using different GED approximations.

Classification

Second experiment : extracting random trainset of size 10% and 30% from each class. Tested $1nn$ classifier on the rest of the dataset using :

- only the set-median (SM)
- only the generalized median (GM)
- the whole trainset (TS)

Classification

Letter (HIGH) Dataset

TS	1st phase	2nd phase	pt	% SM	t(SM)	% GM	t(GM)	% TS	t(TS)
10%	mBipartite	mBipartite	0.023	76.42	0.325	82.82	0.325	83.01	5.275
	mBipartite	mIPFP	0.195	77.40	5.857	84.16	5.771	83.30	110.48
	mIPFP	mIPFP	0.447	78.24	5.951	84.60	5.801	82.95	111.84
30%	mBipartite	mBipartite	0.181	79.94	0.251	84.24	0.250	87.24	11.44
	mBipartite	mIPFP	0.878	81.83	4.323	86.06	4.234	86.86	239.14
	mIPFP	mIPFP	3.437	81.59	4.316	86.08	4.245	86.86	240.96

Monoterpenoides Dataset

TS	1st phase	2nd phase	pt	% SM	t(SM)	% GM	t(GM)	% TS	t(TS)
10%	mBipartite	mBipartite	0.054	32	0.984	29.44	0.957	51.86	3.830
	mBipartite	mIPFP	1.586	53.38	47.96	57.49	51.03	60.69	186.85
	mIPFP	mIPFP	2.044	54.06	47.31	62.38	48.01	60.69	187.83
30%	mBipartite	mBipartite	0.373	36.39	0.747	34.28	0.732	67.92	8.571
	mBipartite	mIPFP	5.148	54.06	36.54	67.79	37.07	75.82	419.81
	mIPFP	mIPFP	15.38	58.37	36.15	74.12	36.57	75.94	419.31

1 Preliminaries

2 The algorithm

3 Experimental Evaluation

4 Conclusion and Perspectives

Conclusion

We proposed an algorithm that computes a generalized median graph with SOD much lower than that of set-median.

GM with reasonably low SOD can be constructed, even with less accurate initialization.

Used in a 1nn classifier, the GM performance is similar to that of the entire trainset, while the classification process is much faster.

Future/current works

Currently developing and testing extended versions of the algorithm that allows the order of the median to be modified.

Extending tests, including k-means in order to produce k different representatives for a class, and implementation of other algorithms from the literature, mostly based on graph embeddings in vector space.

Thanks for your attention