A Graph Pre-image Method Based on Graph Edit Distances

Linlin Jia
linlin.jia@insa-rouen.fr

Paul Honeine, paul.honeine@univ-rouen.fr
Benoit Gaüzère, benoit.gauzere@insa-rouen.fr

Normandie Université, INSA Rouen et Université de Rouen, LITIS Lab

Acknowledgements

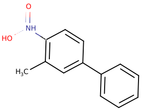
21/01/2021

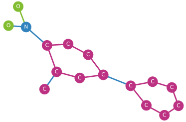
Overview

Graph data

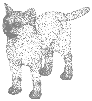
Original data	**Graph representation**	**Original data**	**Graph representation**

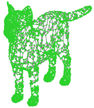
chemoinformatics: molecule



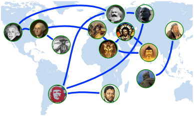


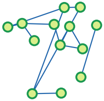
computer vision: 3D point cloud



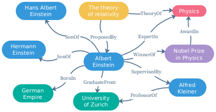


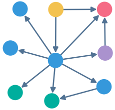
social media: social network



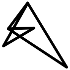


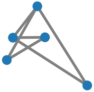
knowledge graph





computer vision: handwriting

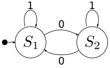




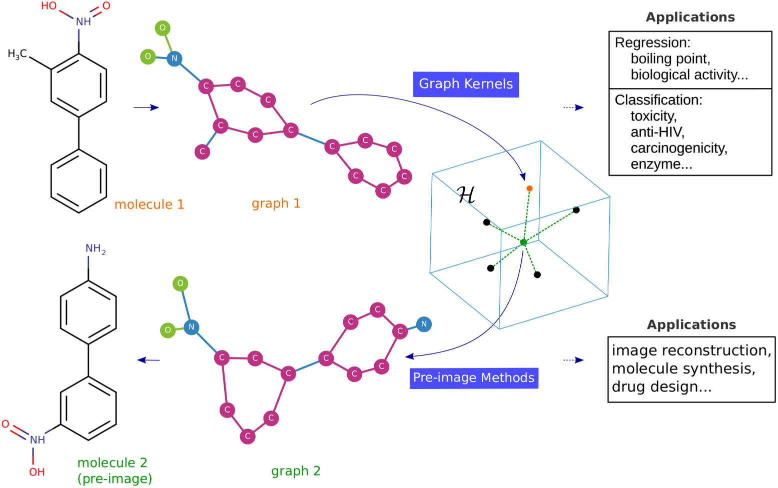
state transition

State-transition table

Input / Current state	0	1
S₁	S₂	S₁
S₂	S₁	S₂



Graph kernels and pre-image problem



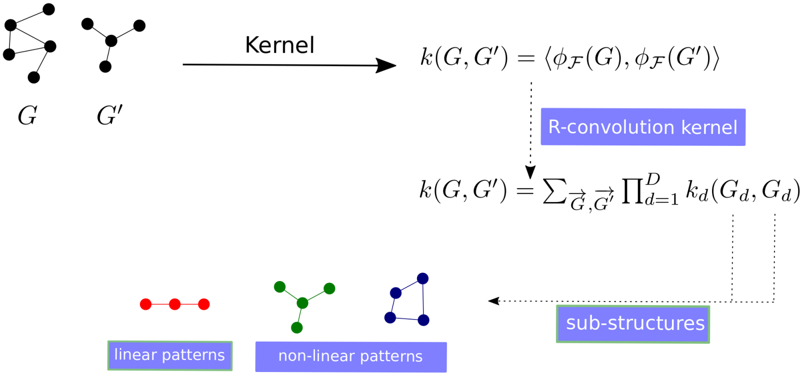
Current methods to construct graphs

- **Based on iterations:**
  - on kernel space
  - restricted to inserting or removing edges
  - small number of inserted/removed edges
  - random construction $\rightarrow$ quality decrease, time consuming
  - no labelling information
- **Based on graph edit distance (GED):**
  - a nature way to construct graphs
  - only explore the graph space

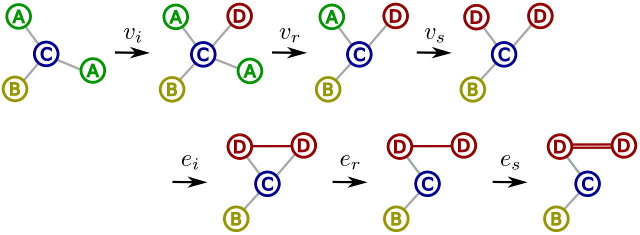
Our method: GED as a direction + an iterative graph pre-image method.

Overview

Linear and non-linear graph kernels



Graph edit distances

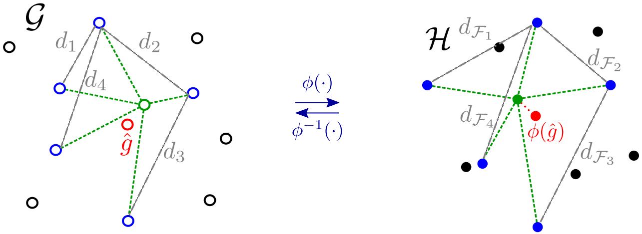


$$(v_{i_0}, v_{r_0}, v_{s_0}, v_{s_1}, v_{r_1} \dots) \to \pi \quad \longrightarrow \quad \texttt{ged}(G_1, G_2) = \min_{\pi_1, \dots, \pi_k \in \Pi(G_1, G_2)} \sum_{i=1}^{k} c(\pi_i)$$

$$\begin{cases} \boldsymbol{\omega} = [n_{vr}, n_{vi}, n_{vs}, n_{er}, n_{ei}, n_{es}]^\top \\ \boldsymbol{c} = [c_{vr}, c_{vi}, c_{vs}, c_{er}, c_{ei}, c_{es}]^\top \end{cases} \quad \longrightarrow \quad \texttt{ged}(G_i, G_j) = \boldsymbol{\omega}^\top \boldsymbol{c}$$

Overview

Align graph space and kernel space



$$d_1 = d_{\mathcal{F}_1}, \quad d_2 = d_{\mathcal{F}_2}, \quad d_3 = d_{\mathcal{F}_3}, \quad d_4 = d_{\mathcal{F}_4}, \quad \ldots$$

$$\begin{cases} \texttt{ged}(G_i, G_j) = \boldsymbol{\omega}^\top \boldsymbol{c} \\ d_{\mathcal{F}}(\phi(G_i), \phi(G_j)) = \sqrt{k(G_i, G_i) + k(G_j, G_j) - 2k(G_i, G_j)} \end{cases}$$

$$\longrightarrow \quad \underset{\boldsymbol{c},\boldsymbol{\omega}}{\arg\min} \sum_{i,j=1}^{N} \left(\texttt{ged}^{i,j} - d_{\mathcal{F}}^{i,j} \right)^2$$

Align graph space and kernel space

$$
\begin{tikzcd}
\boldsymbol{W} \arrow[d, bend right] \\
\boldsymbol{c} \arrow[u, bend right]
\end{tikzcd}
$$

$$
\begin{aligned}
&\arg\min_{\boldsymbol{c}} \quad \left\|\boldsymbol{W}^\top \boldsymbol{c} - \boldsymbol{d}_{\mathcal{F}}\right\|^2 \\
&\text{subject to} \quad \boldsymbol{c} > \boldsymbol{0} \\
&\qquad\qquad c_{vr} + c_{vi} \geqslant c_{vs} \\
&\qquad\qquad c_{er} + c_{ei} \geqslant c_{es}
\end{aligned}
$$

`CVXPY, scipy`

$\forall\ G_i, G_j,$

$\mathtt{ged}(G_i, G_j) = \boldsymbol{\omega}(i,j)^\top \boldsymbol{c}$

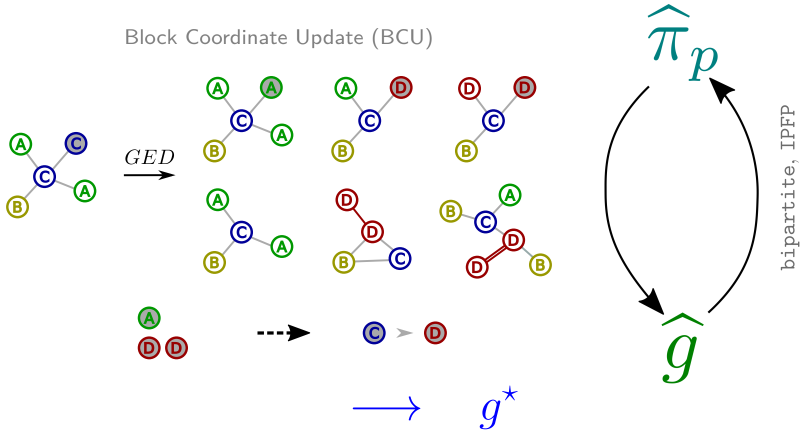
`bipartite, IPFP`

$\longrightarrow \quad \boldsymbol{c}_{optimized}$

- $\boldsymbol{W}^\top$: the N^2-by-6 matrix with rows $\boldsymbol{\omega}(i,j)^\top$
- $\boldsymbol{d}_{\mathcal{F}}$: the vector of N^2 entries $d_{\mathcal{F}}(\phi(G_i), \phi(G_j))$, for $i, j = 1, \ldots, N$

Generate graph pre-image

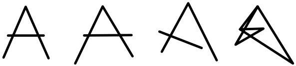
Using $c_{optimized}$, alternately iterate the following procedures:



Overview

Experiment settings

- Datasets:
  - the `Letter` datasets
  - Handwritting letters with different distortions
  - nodes with continuous labels



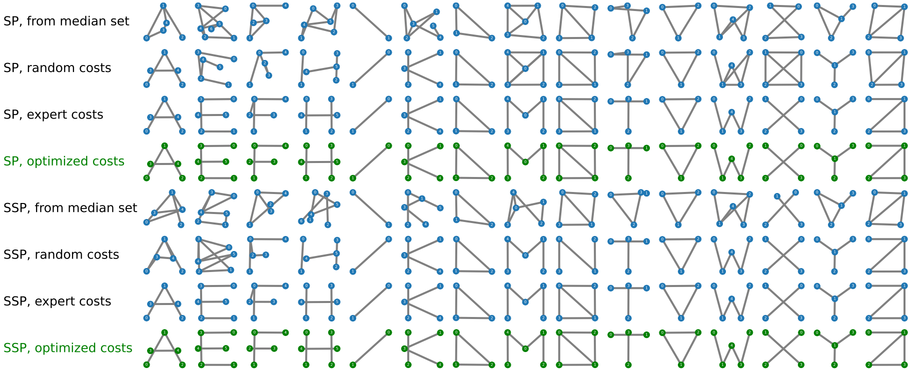
- Configurations:
  - m`bipartite` and m`IPFP` with $m = 40$
  - size of median set: 150
  - maximum number of iterations: 6
  - number of parallelized jobs: 28

Distances in kernel space

Datasets	Graph Kernels	Algorithms	$d_{\mathcal{H}}$ GM	Running Times		
				Optimization	Generation	Total
Letter-high	Shortest paths (SP)	From median set	0.469	-	-	-
		IAM (random costs)	0.467	-	142.59	142.59
		IAM (expert costs)	**0.451**	-	30.31	30.31
		IAM (optimized costs)	0.460	5968.92	26.55	5995.47
	structural sp (SSP)	From median set	0.459	-	-	-
		IAM (random costs)	0.435	-	30.22	30.22
		IAM (expert costs)	**0.391**	-	29.71	29.71
		IAM (optimized costs)	**0.394**	24.79	25.60	50.39
Letter-med	Shortest paths (SP)	From median set	0.469	-	-	-
		IAM (random costs)	0.303	-	25.61	25.61
		IAM (expert costs)	**0.288**	-	26.93	26.93
		IAM (optimized costs)	**0.288**	23.72	24.79	48.52
	structural sp (SSP)	From median set	0.478	-	-	-
		IAM (random costs)	0.286	-	24.77	24.77
		IAM (expert costs)	**0.248**	-	27.51	27.51
		IAM (optimized costs)	**0.248**	27.06	29.24	56.30
Letter-low	Shortest paths (SP)	From median set	0.166	-	-	-
		IAM (random costs)	**0.116**	-	26.47	26.47
		IAM (expert costs)	**0.116**	-	24.87	24.87
		IAM (optimized costs)	**0.116**	26.35	29.97	56.31
	structural sp (SSP)	From median set	0.148	-	-	-
		IAM (random costs)	0.103	-	30.22	30.22
		IAM (expert costs)	**0.086**	-	29.43	29.43
		IAM (optimized costs)	0.104	21.95	24.59	46.53

Pre-images constructed by different algorithms

Pre-images generated as median graphs for each letter of *Letter-high* dataset using random costs, expert costs and optimized costs:



Overview

- Conclusion:
  - a novel method to estimate graph pre-images
  - align kernel space and graph space
  - generate better pre-images than other methods
- Future work:
  - construct pre-images as arbitrary graphs
  - establish convergence proof
  - extend to graphs with symbolic labels
  - extend non-constant costs

Questions

Thank you.

Any questions?