Hy T. Son

Chapter I: Introduction

Chapter II: Graph Kernels

Chapter III: Graph Neural Networks

Chapter IV: Visualization

Chapter V: Results

Chapter VI: Conclusion

Molecular clustering

Author: Hy Truong Son
Advisor: Prof. Risi Kondor
TTIC 31220 - Unsupervised Learning and Data Analysis

The University of Chicago

Chicago, May 2017

Contents

Chapter I: Introduction

Chapter II: Graph Kernels

Chapter III: Graph Neural Networks

Chapter IV: Visualization

Chapter V: Results

Chapter VI: Conclusion

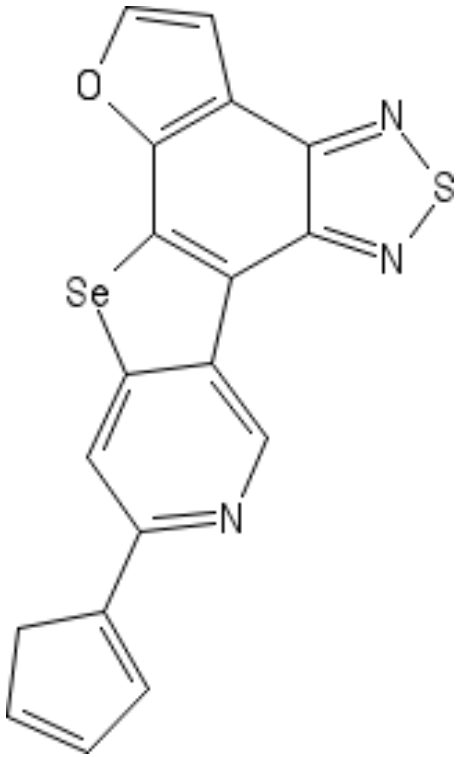
1. Chapter I: Introduction
2. Chapter II: Graph Kernels
3. Chapter III: Graph Neural Networks
4. Chapter IV: Visualization
5. Chapter V: Results
6. Chapter VI: Conclusion

Molecular chemical representation

Hy T. Son

Chapter I: Introduction
Chapter II: Graph Kernels
Chapter III: Graph Neural Networks
Chapter IV: Visualization
Chapter V: Results
Chapter VI: Conclusion

Harvard Clean Energy Project (HCEP) Dataset 1

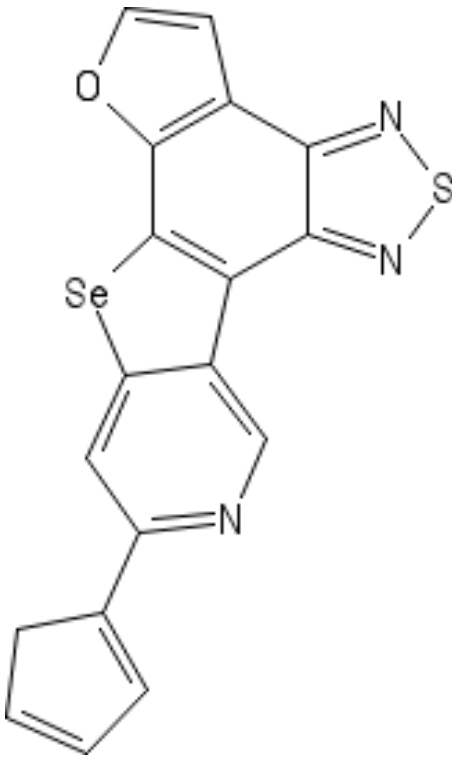


Compound: C18H9N3OSSe

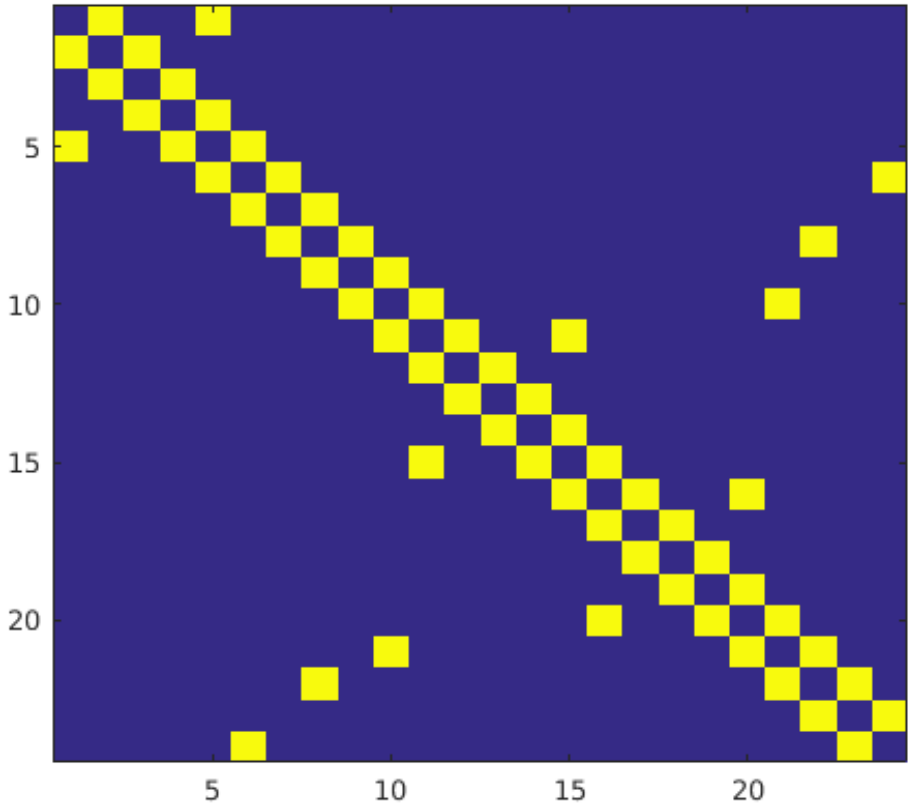
SMILES: C1C=CC=C1c1cc2[Se]c3c4occc4c4nsnc4c3c2cn1

Power Conversion Efficiency (PCE, range 0 - 11): 5.16195

Molecular graph representation



C18H9N3OSSe

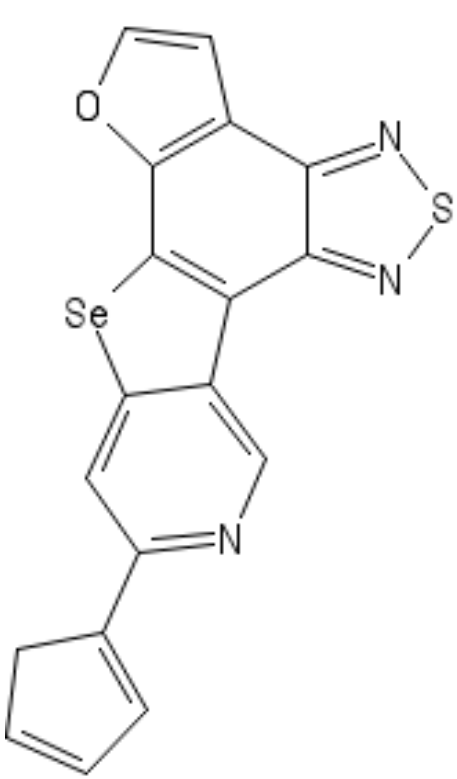


Adjacency matrix

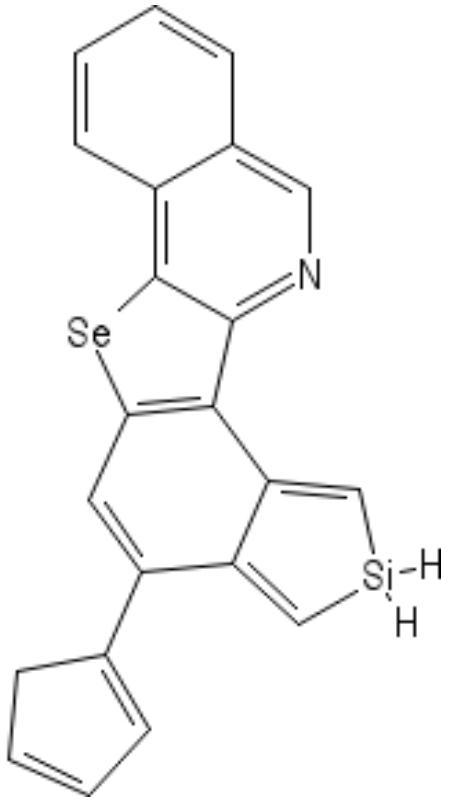
Measure the similarity among molecules?

Hy T. Son

Chapter I: Introduction
Chapter II: Graph Kernels
Chapter III: Graph Neural Networks
Chapter IV: Visualization
Chapter V: Results
Chapter VI: Conclusion

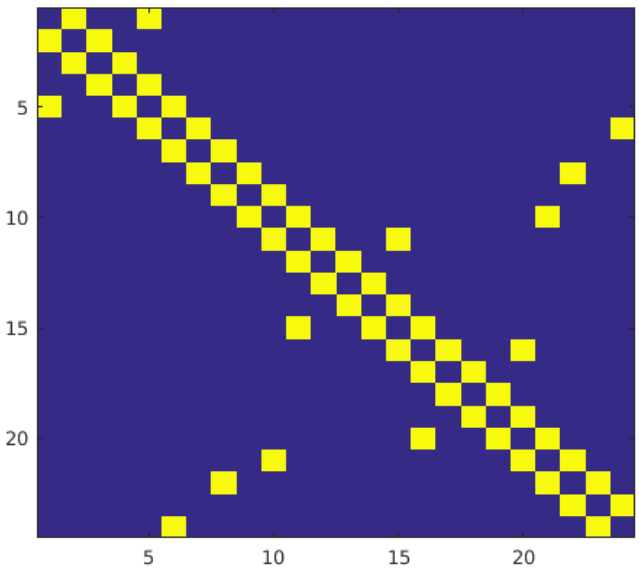


C18H9N3OSSe

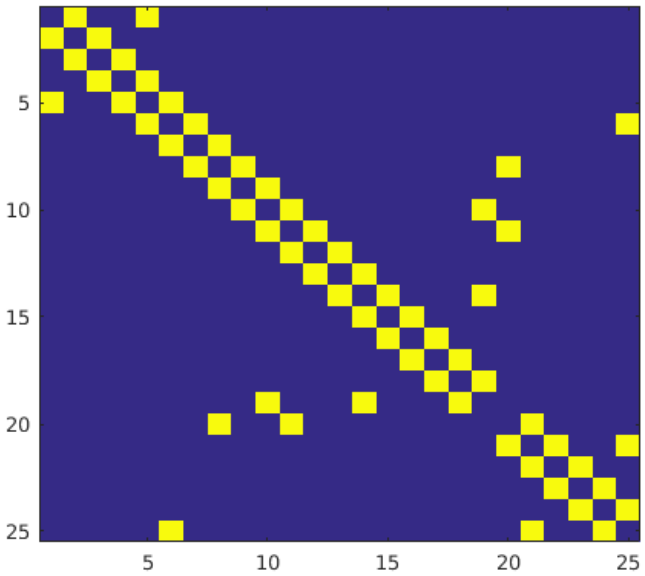


C22H15NSeSi

Measure the similarity among graphs?



C18H9N3OSSe



C22H15NSeSi

Positive Semi-definite Graph Kernel

Given 2 graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Each vertex is associated with a feature vector $f : V \to \Omega$. A graph kernel between G_1 and G_2 can be defined as:

$$\mathcal{K}_{graph}(G_1, G_2) = \frac{1}{|V_1|} \cdot \frac{1}{|V_2|} \cdot \sum_{v_1 \in V_1} \sum_{v_2 \in V_2} k_{base}(f(v_1), f(v_2))$$

where k_{base} is the based kernel and that can be:

- Linear: $k_{base}(x, y) = \langle x, y \rangle_{norm} = x^T y / (\|x\| \cdot \|y\|)$
- Quadratic: $k_{base}(x, y) = (\langle x, y \rangle_{norm} + q)^2$
- RBF: $k_{base}(x, y) = exp(-\gamma \|x - y\|^2)$

Weisfeiler-Lehman Graph Kernel - Part 1

Given a set of N graphs $\mathcal{G} = \{G^{(1)}, .., G^{(N)}\}$ where $G^{(i)} = (V^{(i)}, E^{(i)})$ $(1 \leq i \leq N)$. Let $f^G : V \to \Omega$ be the initial feature vector for each vertex of graph $G = (V, E)$. Let $S_k^G(v)$ be the set of vectors for vertex $v \in V$ of graph G at Weisfeiler-Lehman level k.

Weisfeiler-Lehman Graph Kernel - Part 2

Set of graphs: $\mathcal{G} = \{G^{(1)}, .., G^{(N)}\}$
Set of vertex features: $\mathcal{F} = \{f^{G^{(1)}}, .., f^{G^{(N)}}\}$
Number of WL levels: $K \in \mathbb{N}$

function Dictionary-WL()
01. Universal dictionary: $\mathcal{D} \leftarrow \emptyset$
02. for $i = 1 \rightarrow N$:
03. Initialize the WL level 0
04. for each $v \in V^{(i)}$:
05. $S_0^{G^{(i)}}(v) \leftarrow \{f^{G^{(i)}}(v)\}$
06. $\mathcal{D} \leftarrow \mathcal{D} \cup \{S_0^{G^{(i)}}(v)\}$
07. end for

Weisfeiler-Lehman Graph Kernel - Part 3

08. Build the WL level 1, 2, .., K
09. for $k = 1 \rightarrow K$:
10. for each $v \in V^{(i)}$:
11. $S_k^{G^{(i)}}(v) \leftarrow S_{k-1}^{G^{(i)}}(v)$
12. for each $(u, v) \in E^{(i)}$:
13. $S_k^{G^{(i)}}(v) \leftarrow S_k^{G^{(i)}}(v) \cup S_{k-1}^{G^{(i)}}(u)$
14. end for
15. $\mathcal{D} \leftarrow \mathcal{D} \cup \{S_k^{G^{(i)}}(v)\}$
16. end for
17. end for
18. end for
19. **return** $\mathcal{D}$, $\mathcal{S} = \{S_0^{G^{(1)}}, .., S_0^{G^{(N)}}, ..., S_K^{G^{(1)}}, .., S_K^{G^{(N)}}\}$

end function

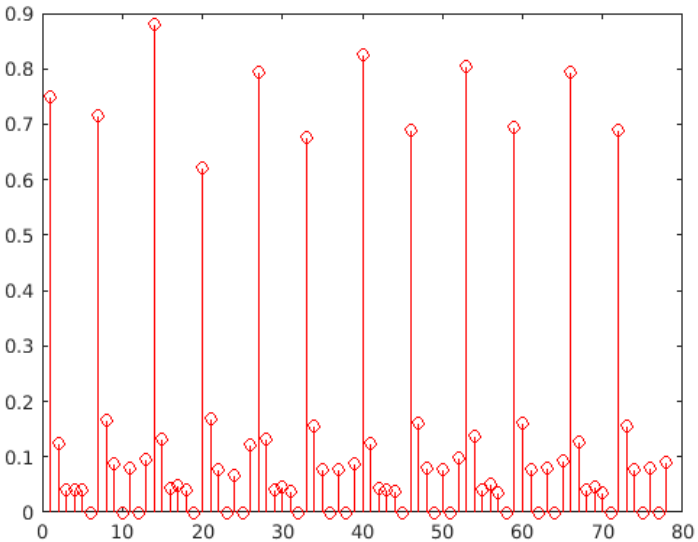
Weisfeiler-Lehman Graph Kernel - Part 4

We define a **receptive field** of radius R at a vertex v is the set of all vertices that are reachable from v by a path has length no greater than R. For example, a receptive field can be represented as [C C C Se], or [C C N N N N S S S].
The number of different receptive field of radius 3 in 50K molecules of HCEP dataset is: **11,172**. For each molecule, we can find all of its receptive fields, and map the frequencies into a vector of length **11,172** as a fingerprint.
I call this method as **Dictionary Weisfeiler-Lehman Graph Kernel**. The original method is **Morgan molecular fingerprint** that only supports binary feature.

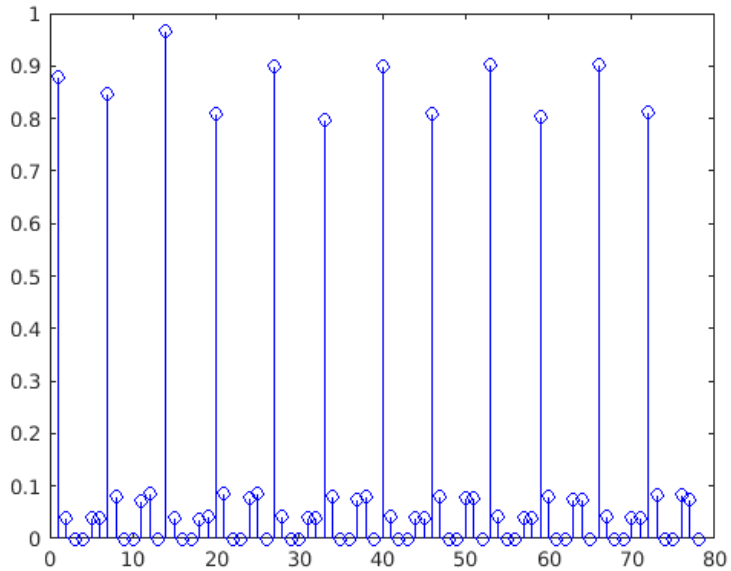
Hy T. Son

Chapter I: Introduction
Chapter II: Graph Kernels
Chapter III: Graph Neural Networks
Chapter IV: Visualization
Chapter V: Results
Chapter VI: Conclusion

Histogram-Alignment Weisfeiler-Lehman Graph Kernel 2



C18H9N3OSSe



C22H15NSeSi

Graph (Covariant) Neural Networks - Part 1

Chapter I: Introduction
Chapter II: Graph Kernels
Chapter III: Graph Neural Networks
Chapter IV: Visualization
Chapter V: Results
Chapter VI: Conclusion

Input graph: $G = (V, E)$
Vertex feature vectors: $f : V \to \Re^d$
Learning target: $T \in \Re$
Number of levels: $L \in \mathbb{N}$
The receptive field: $\mathcal{R}(v)$ where $v \in V$
Non-linearity: σ
Outer-product: $\otimes$

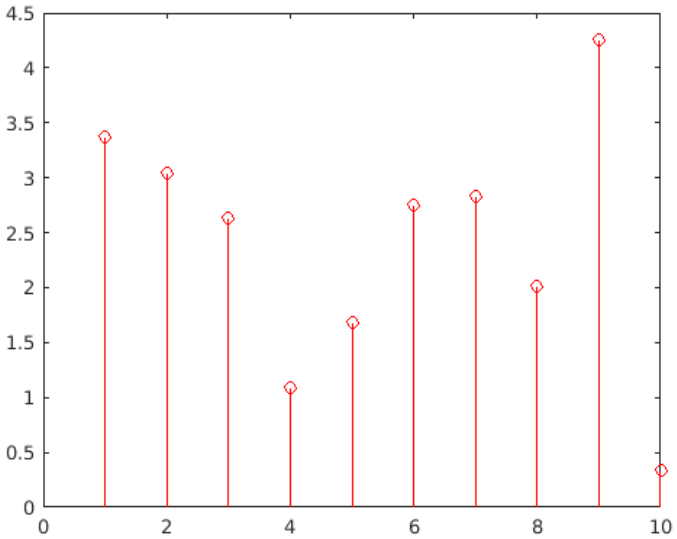
Graph (Covariant) Neural Networks - Part 2

function GCN($G = (V, E)$, $T \in \Re$, $f : V \to \Re^d$, $L \in \mathbb{N}$)

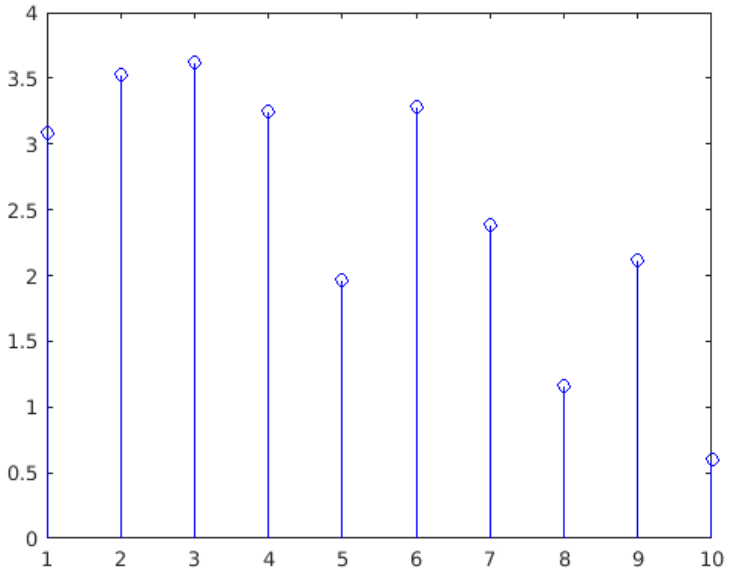
01. Level 0: $\phi_0^v \leftarrow \sigma(W_0^{(1)} f(v))$ $(\forall v \in V)$
02. for each level $l = 1 \to L$:
03. for each $v \in V$:
04. $\Phi \leftarrow \bigcup_{u \in \mathcal{R}(v)} \phi_{l-1}^u$
05. $\Theta \leftarrow \sum_{i<j} \Phi_i \otimes \Phi_j$
06. $\Psi_i \leftarrow \sum_k \Theta_{i,k} + \Theta_{k,i}$
07. $\phi_l^v \leftarrow \sigma(W_l^{(1)} f(v) + W_l^{(2)} \Psi)$
08. end for
09. end for
10. Graph feature: $\phi_G \leftarrow \cup_{v \in V} \phi_L^v$
11. Linear regression: $U \leftarrow argmin_{u \in \Re^h} (u^T \phi_G - T)^2$

end function

Graph (Covariant) Neural Networks - Part 3

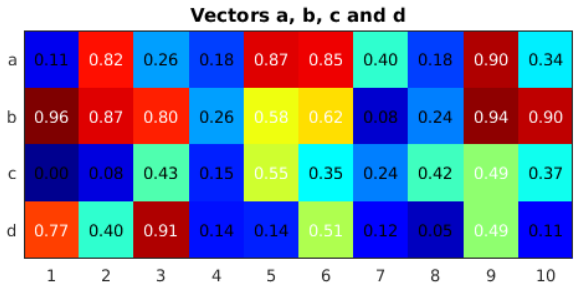


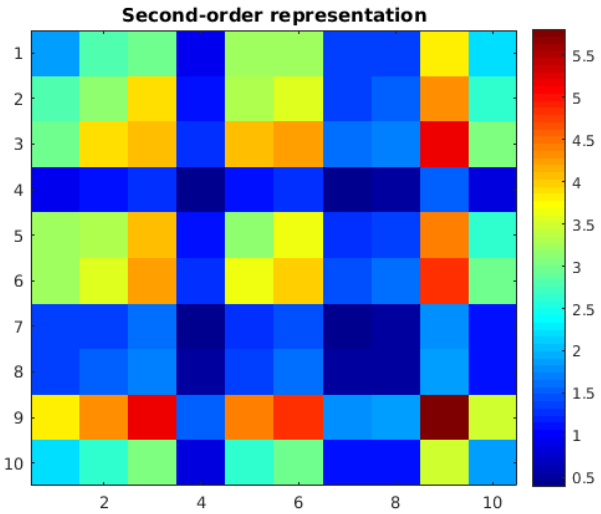
C18H9N3OSSe

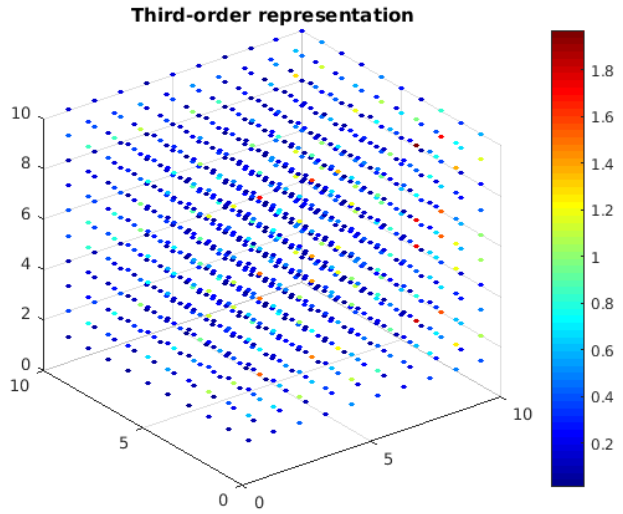


C22H15NSeSi

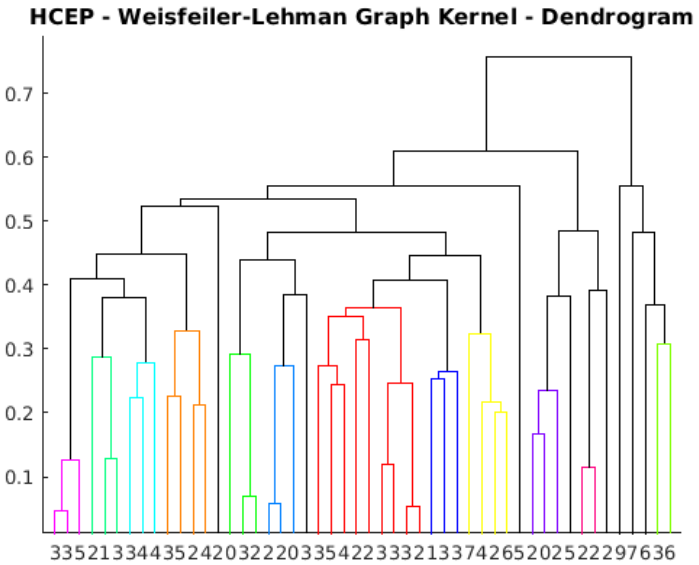
Higher-order representation of a set of vectors

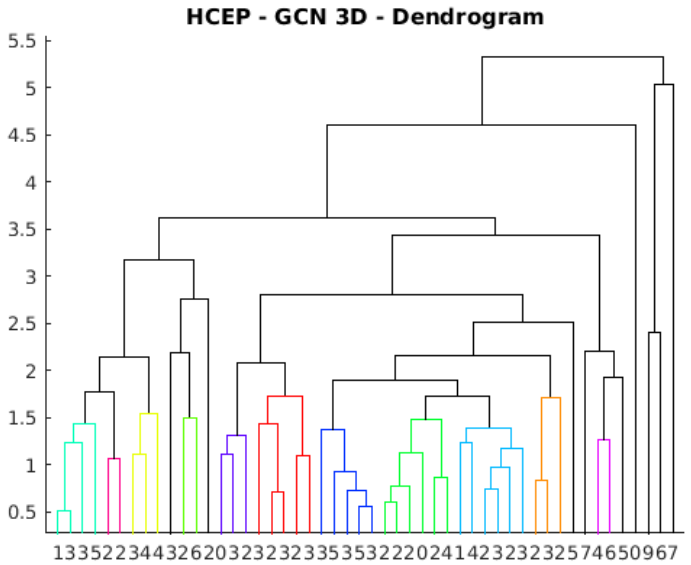




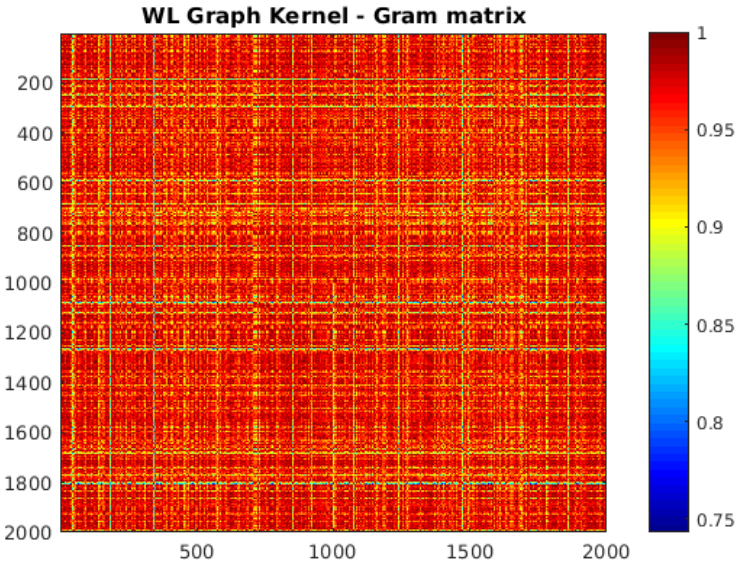


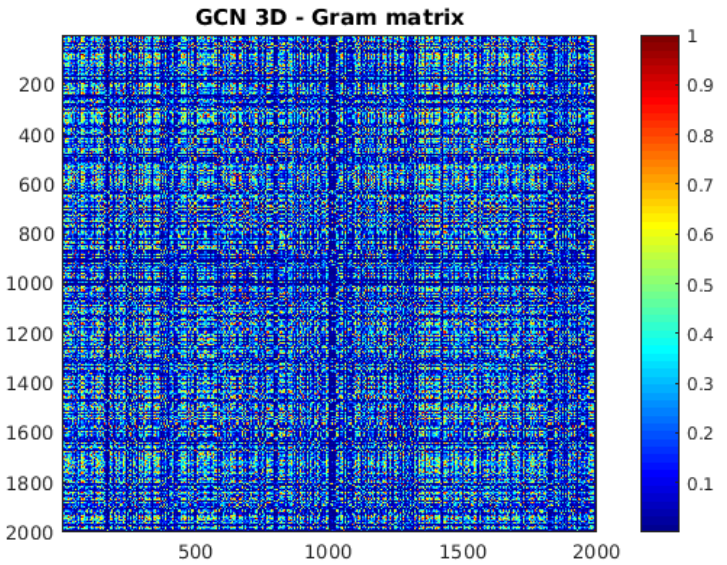
Hierarchical Clusterings



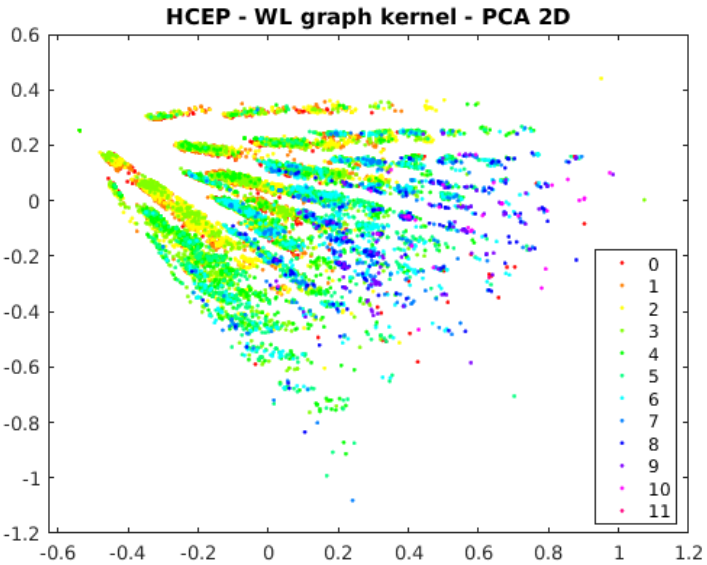


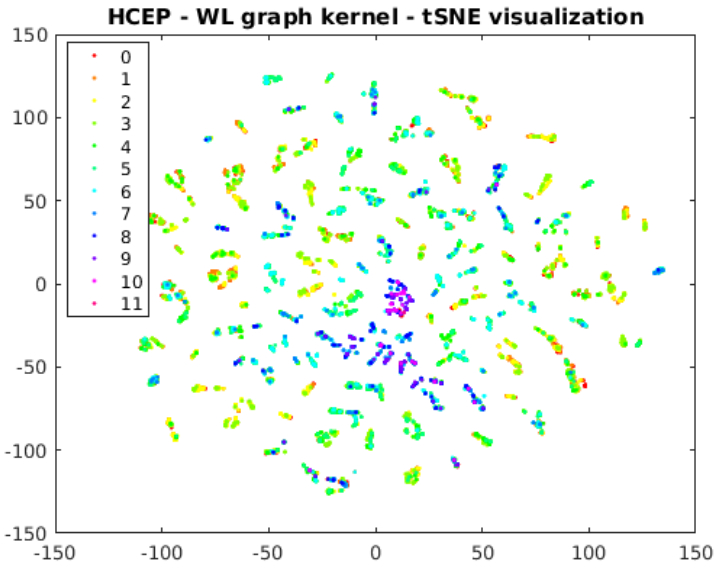
Kernel PCA


WL Graph Kernel - Gram matrix
200
400
600
800
1000
1200
1400
1600
1800
2000
500
1000
1500
2000
1
0.95
0.9
0.85
0.8
0.75

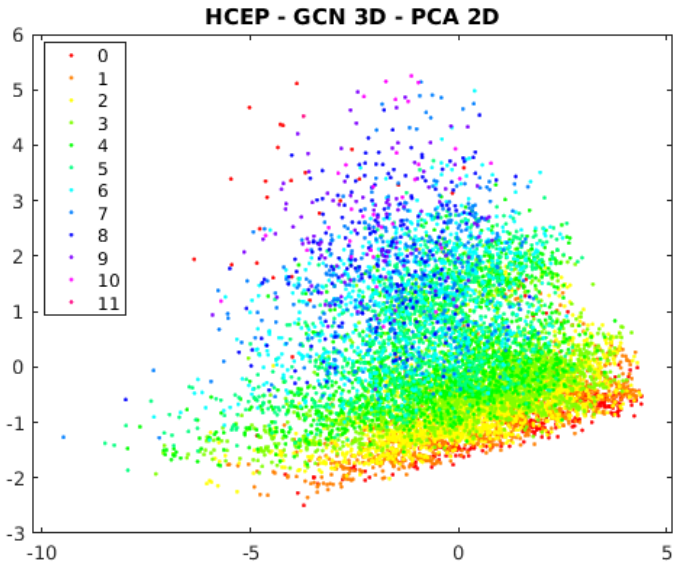

GCN 3D - Gram matrix
200
400
600
800
1000
1200
1400
1600
1800
2000
500
1000
1500
2000
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

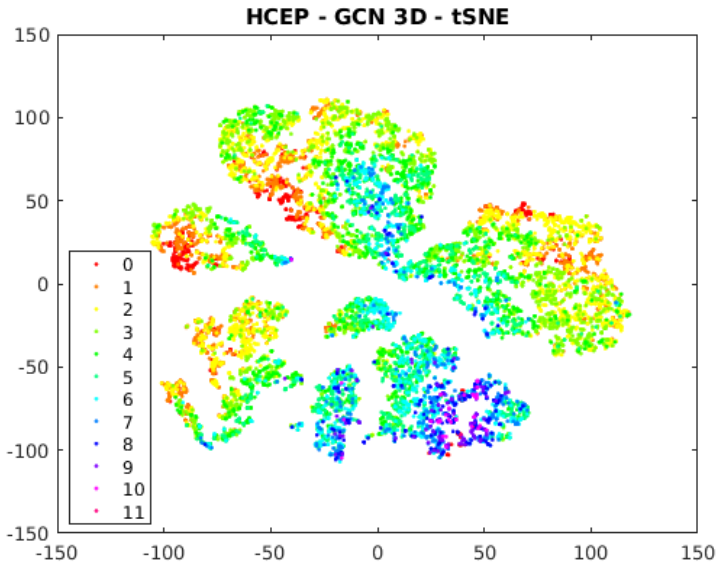
Weisfeiler-Lehman Graph Kernel





Graph (Covariant) Neural Networks





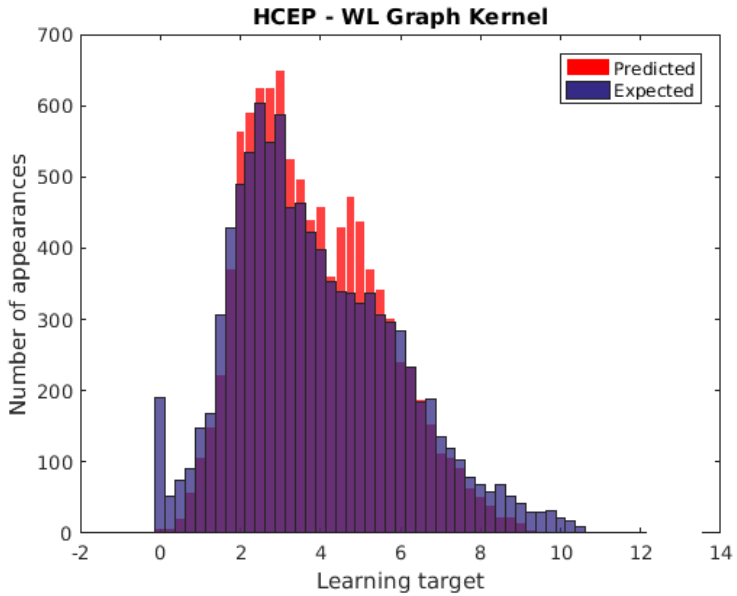
Experimental results - MAE

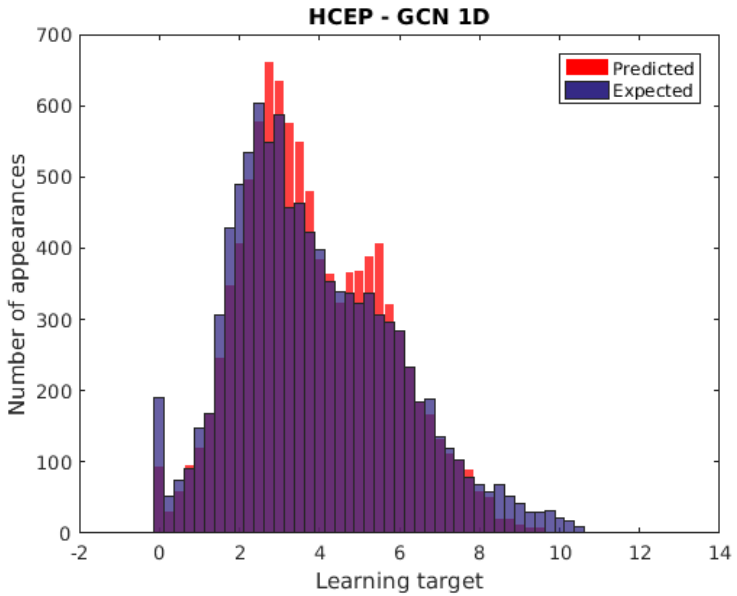
	Linear regression	Gaussian Process	SVM
WL	0.805135	0.760736	0.878423
GCN 1D	**0.539711**	**0.528121**	**0.493815**
GCN 2D	0.624043	0.653314	0.560620
GCN 3D	0.594986	0.602821	0.551861

Experimental results - RMSE

	Linear regression	Gaussian Process	SVM
WL	1.096222	1.093145	1.202667
GCN 1D	**0.800283**	**0.833758**	**0.775039**
GCN 2D	0.895965	0.940610	0.843593
GCN 3D	0.876125	0.883956	0.834215

Experimental results - Distribution





Conclusion

- Graph Neural Networks outperform Graph Kernels in HCEP dataset
- The top features of GCNs give us clustering information of molecules (with both PCA and t-SNE)

Reference

1 https://cepdb.molecularspace.org/

2 Nils M. Kriege, Pierre-Louis Giscard, *On Valid Optimal Assignment Kernels and Applications to Graph Classification*, NIPS 2016

3 David Duvenaud, Dougal Maclaurin, Jorge Aguilera Iparraguirre, Rafael Gomez Bombarelli, Timothy Hirzel, Alan Aspuru Guzik, Ryan P. Adams, *Convolutional Networks on Graphs for Learning Molecular Fingerprints*.

Chapter I: Introduction

Chapter II: Graph Kernels

Chapter III: Graph Neural Networks

Chapter IV: Visualization

Chapter V: Results

Chapter VI: Conclusion

Thank you very much for your attention!