A Short Journey through Whole Graph Embedding Techniques

Mario Guarracino
Department of Economics and Law
University of Cassino and Southern Lazio

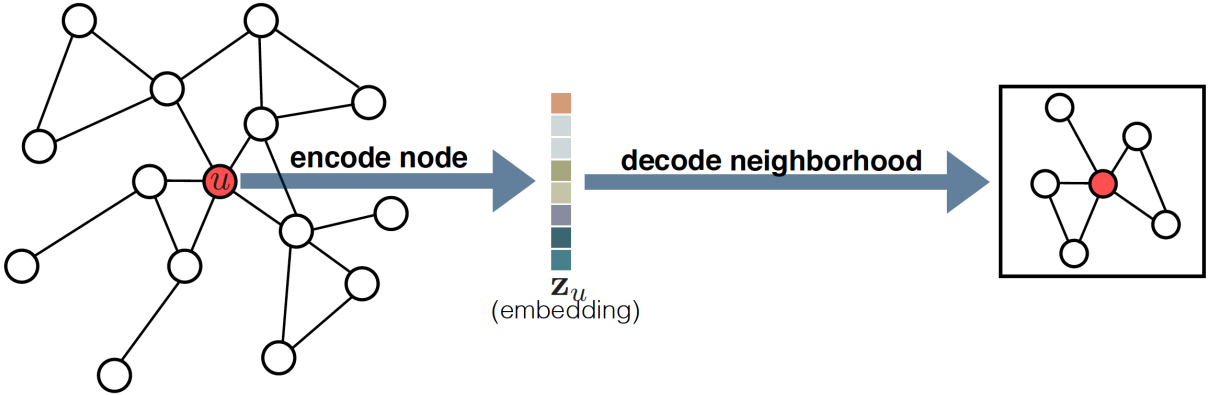
High Performance Computing and Networking Institute
National Research Council

Graph embedding

- Any method that computes or *learns* a mapping from a graph to a vector space, while preserving relevant graph properties.
- Given a graph $G = (V, E)$, a **graph embedding** of dimension d is an encoding

$$\text{ENC}: v \in V \rightarrow \mathbf{z}_v \mathbb{R}^d,$$

- preserving some proximity measure defined on the nodes of graph *G*



Graph embedding

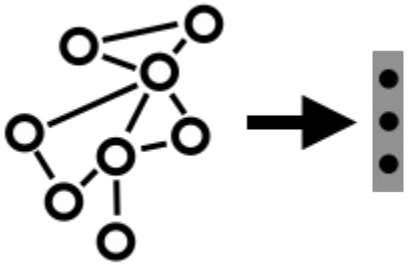
- Any method that computes or *learns* a mapping from a graph to a vector space, while preserving relevant graph properties.
- **Node embeddings** used to study interactions among entities, identifying groups of nodes behaving similarly or finding global and local connectivity patterns in a given network

Whole-graph embedding

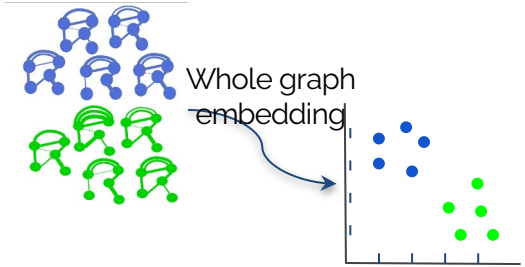
Given a set of graphs $\mathcal{G} = \{G_1, \dots, G_m\}$ with the same set of vertices V, a **whole-graph embedding** of dimension d is an encoding ENC

$$\mathrm{ENC}: G_i \in \mathcal{G} \longrightarrow y_i \in \mathbb{R}^d, i \in 1, \dots, |\mathcal{G}|$$

such that ENC preserves some proximity measure defined on G



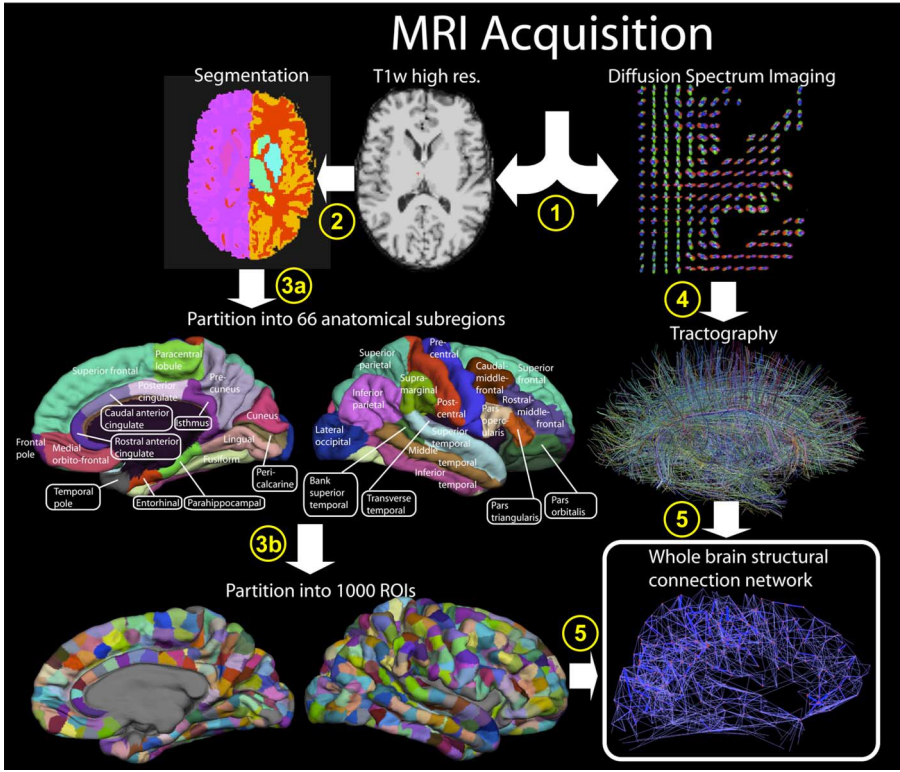
graph embedding



- *Applications*
  - Graphs classification
  - Graphs clustering
  - Visualization

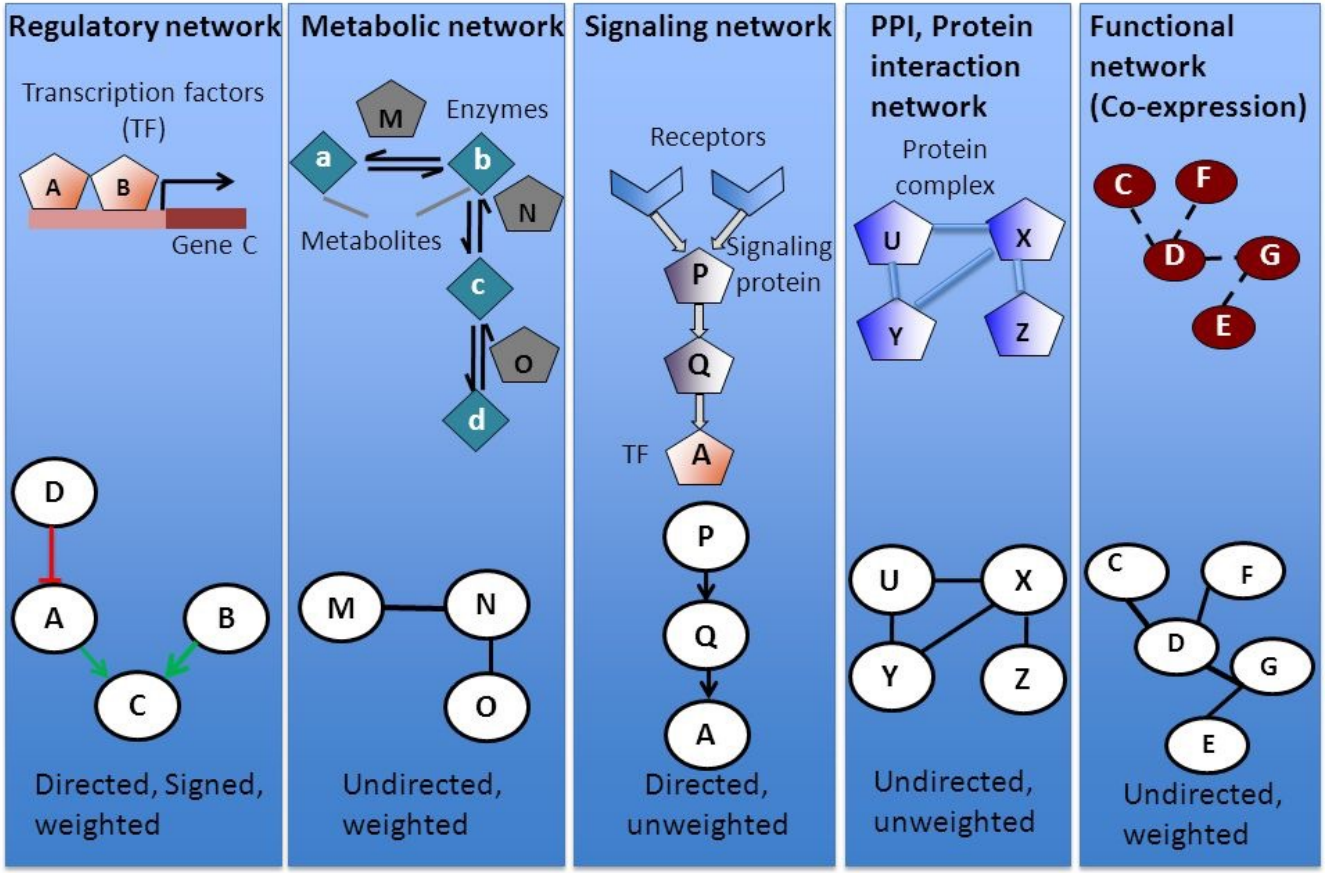
Functional brain networks

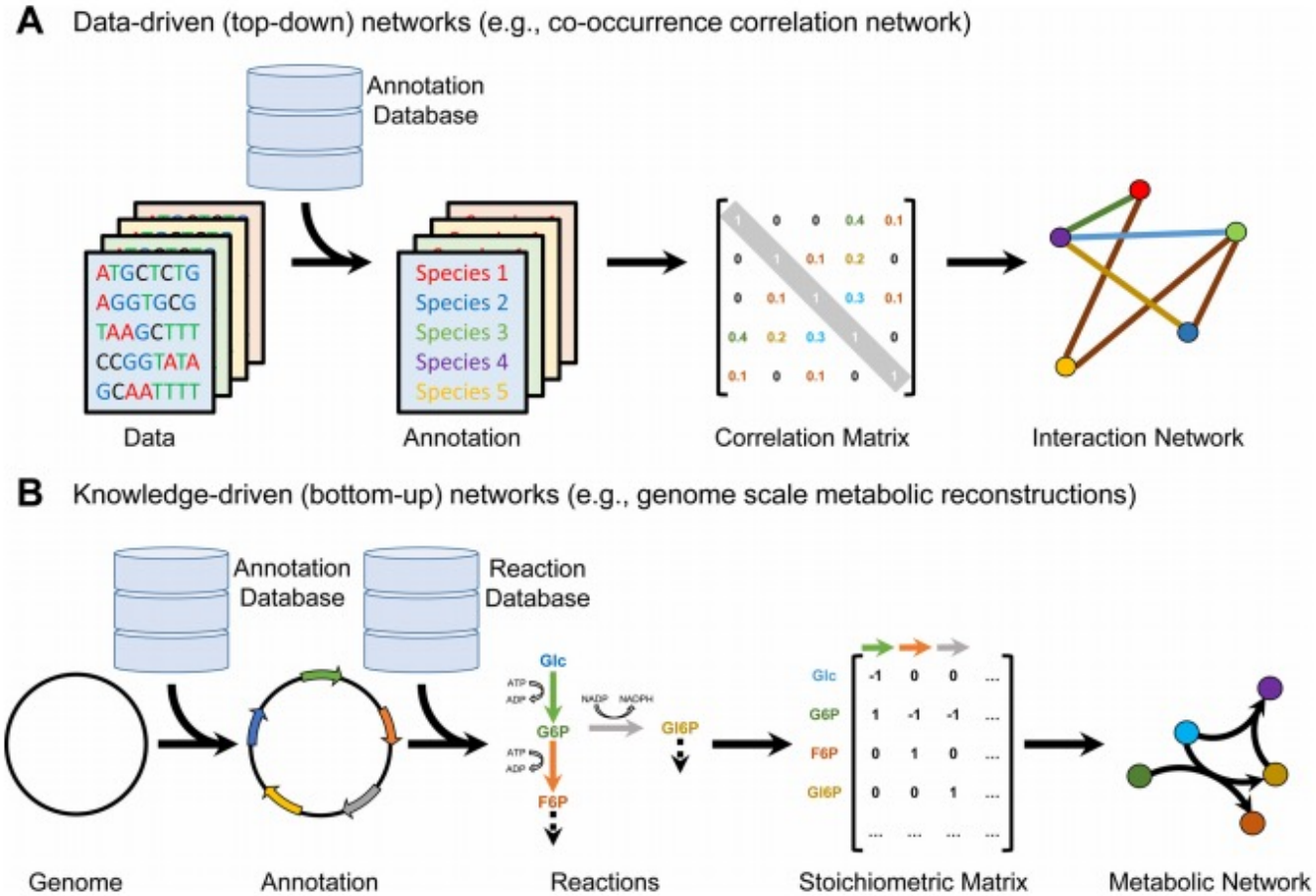
- One network for each patient
- All networks the same nodes
- Nodes represent neural regions of interest



Hagmann et al., *Mapping the Structural Core of Human Cerebral Cortex*, PLoS Biology, 2008

Biological Networks

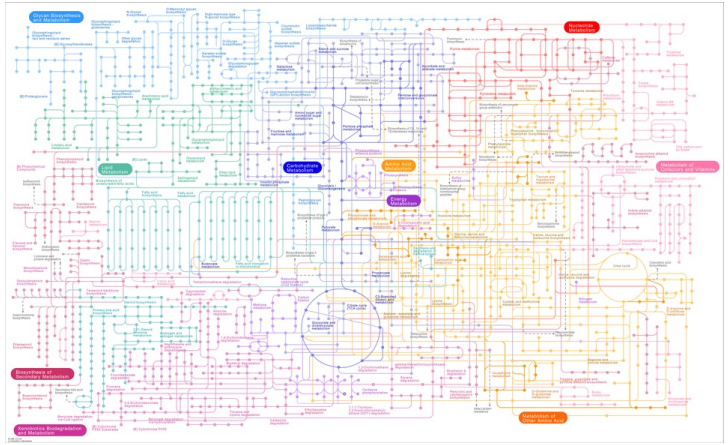


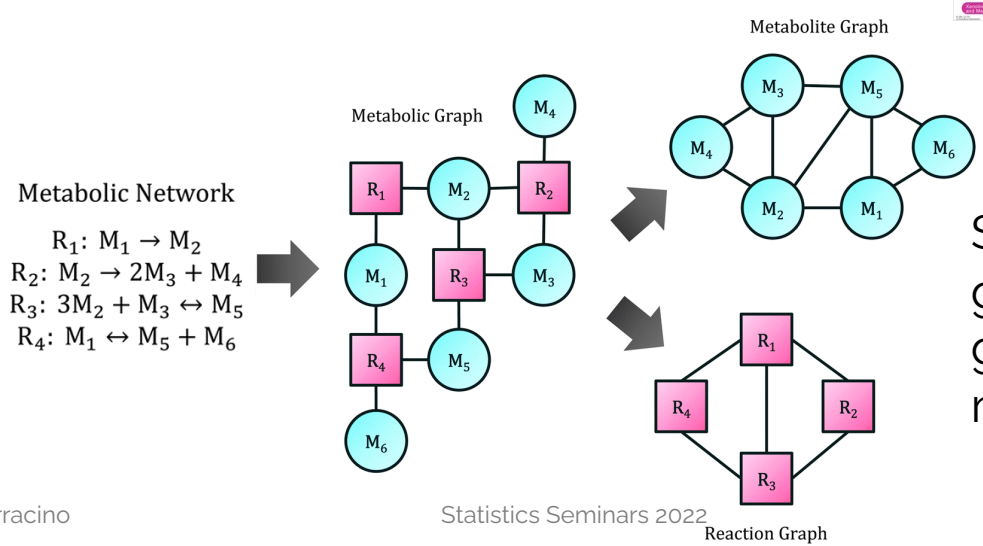


Bauer, Eugen, and Ines Thiele. "From Network Analysis to Functional Metabolic Modeling of the Human Gut Microbiota." MSystems 3.3 (2018): e00209-17.

Metabolic Networks

Representation of the metabolic structure of a cell (chemical reactions, involved metabolites, and associated genes)





Several types of graphs can be generated from metabolic networks

Approaches to WGE

Matrix Factorization

- ❑ Laplacian Eigenmaps
- ❑ Adjacency Spectral Embedding
- ❑ Joint Embedding
- ❑ ...

Graph Kernels

- ❑ Weisfeiler-Lehman (WL)
- ❑ Shortest Path (SP)
- ❑ Random Walk (RW)
- ❑ DeepWL
- ❑ ...

Neural Network-based

- ❑ Patchy-San
- ❑ Graph2vec
- ❑ sub2vec
- ❑ Anonymous walk embeddings
- ❑ Autoencoders
- ❑ GraphSAGE
- ❑ DGCNN
- ❑ U2GNN
- ❑ ...

Approaches to WGE

Matrix Factorization

- ❑ Laplacian Eigenmaps
- ❑ Adjacency Spectral Embedding
- ❑ **Joint Embedding**
- ❑ ...

Neural Network-based

- ❑ Patchy-San
- ❑ **Graph2vec**
- ❑ sub2vec
- ❑ Anonymous walk embeddings
- ❑ Autoencoders
- ❑ GraphSAGE
- ❑ DGCNN
- ❑ U2GNN
- ❑ ...

Graph Kernels

- ❑ Weisfeiler-Lehman (WL)
- ❑ **Shortest Path (SP)**
- ❑ Random Walk (RW)
- ❑ DeepWL
- ❑ ...

Matrix Factorization

- Represent graph properties in the form of a matrix[1] and factorize this matrix to obtain the embedding
- In most cases, the input is a graph and the output is a set of node embeddings
- For WGE, the input is a set of graphs and the output is a set of graph embeddings

[1] e.g., node adjacency matrix, Laplacian matrix, or node transition probability matrix

MF: Joint Embedding (JE)

- Given a set of graphs $\mathcal{G}=\{G_1, \ldots, G_m\}$ on a common set of vertices V, JE simultaneously embeds all G_i into $\mathbb{R}^d$ representing each G_i with a vector $\lambda_i \in \mathbb{R}^d$ (*loading*)
- It identifies a linear subspace spanned by rank one symmetric matrices and projects graph adjacency matrices into this subspace

$$(\bar{\lambda}_1, \ldots, \bar{\lambda}_m, \bar{h}_1, \ldots, \bar{h}_d) = \operatorname{argmin}_{\lambda_i, \|h_k\|=1} \sum_{i=1}^{m} \|A_i - \sum_{k=1}^{d} \lambda_i[k] h_k h_k^{\mathsf{T}}\|_F^2$$

where A_i = adjacency matrix of G_i, $\lambda_i = (\lambda_i[1], \ldots, \lambda_i[d])$

- Loadings λ_i are the extracted features for inference problems

[Wang et al, *Joint embedding of graphs*, IEEE Trans on PAMI, 2019]

Graph Kernels

- Based on the comparison of graph sub-structures[1] via kernels (similarity measures)
- The inner product of vector representations of graph sub-structures is used for pairwise graph comparisons

For WGE, embedding dimension d
is the number of graphs in the dataset m
(dimension of the Gram matrix)

[1] e.g., shortest paths, random walks, graphlets, and subtree patterns

Shortest Path (SP) kernel

Count pairs of shortest paths in two graphs that have the same starting and ending nodes as well as the same length

1. Graphs G_1 and G_2 are transformed into shortest-path graphs S_1 and S_2
   - same set of nodes as G_1 and G_2, there exists an edge between a pair of nodes in S_i which are connected by a walk in G_i, edges labeled by shortest distance of their nodes. [F.M. $O(|V|^3)$])
2. SP kernel graph on $S_1=(V_1,E_1)$ and $S_2=(V_2,E_2)$ defined as

$$k_{SP}\left(S_1,S_2\right)=\sum_{e_1\in E_1}\sum_{e_2\in E_2}k_{walk}^{(1)}\left(e_1,e_2\right)$$

$k_{walk}^{(1)}$ = positive definite kernel on edge walks of length 1

$O(|V|^4)$

Graph2vec[1]

Inspired by the neural document embedding model [2]

- Given a labeled graph *G*, sample *c* *rooted subgraphs* of degree d* from *G* to learn a representation of *G*
- Graph2vec = feed-forward NN to learn distributed representations of graphs

* subgraph including all nodes reachable in *d* hops from the node

[1] Narayanan et al., graph2vec: Learning Distributed Representations of Graphs, ArXiv, 2017
[2] Le and Mikolov, *Distributed Representations of Sentences and Documents*, Proc. ICML, 2014

Graph2vec[1]

Inspired by the neural document embedding model [2]

- A graph is viewed as a document
- The *subgraphs of degree d rooted in* each node are the words composing the document
  - Subgraph extraction and relabeling follow the WL refinement

[1] Narayanan et al., graph2vec: Learning Distributed Representations of Graphs, ArXiv, 2017
[2] Le and Mikolov, *Distributed Representations of Sentences and Documents*, Proc. ICML, 2014

Graph2vec

Algorithm 1: GRAPH2VEC $(\mathbb{G}, D, \delta, \mathfrak{e}, \alpha)$

input : $\mathbb{G} = \{G_1, G_2, ..., G_n\}$: Set of graphs such that each graph $G_i = (N_i, E_i, \lambda_i)$ for which embeddings have to be learnt
D: Maximum degree of rooted subgraphs to be considered for learning embeddings. This will produce a vocabulary of subgraphs, $SG_{vocab} = \{sg_1, sg_2, ...\}$ from all the graphs in $\mathbb{G}$
δ: number of dimensions (embedding size)
$\mathfrak{e}$: number of epochs
α: Learning rate

output: Matrix of vector representations of graphs $\Phi \in \mathbb{R}^{|\mathbb{G}| \times \delta}$

begin
  Initialization: Sample Φ from $R^{|\mathbb{G}| \times \delta}$
  for $e = 1$ *to* $\mathfrak{e}$ **do**
    $\mathfrak{G} =$ SHUFFLE $(\mathbb{G})$
    for *each* $G_i \in \mathfrak{G}$ **do**
      for *each* $n \in N_i$ **do**
        for $d = 0$ *to* D **do**
          $sg_n^{(d)} :=$ GETWLSUBGRAPH(n, G_i, d)
          $J(\Phi) = -\log \Pr\,(sg_n^{(d)} | \Phi(G))$
          $\Phi = \Phi - \alpha \frac{\partial J}{\partial \Phi}$
  return Φ

Algorithm 2: GETWLSUBGRAPH (n, G, d)

input : n: Node which acts as the root of the subgraph
$G = (N, E, \lambda)$: Graph from which subgraph has to be extracted
d: Degree of neighbours to be considered for extracting subgraph

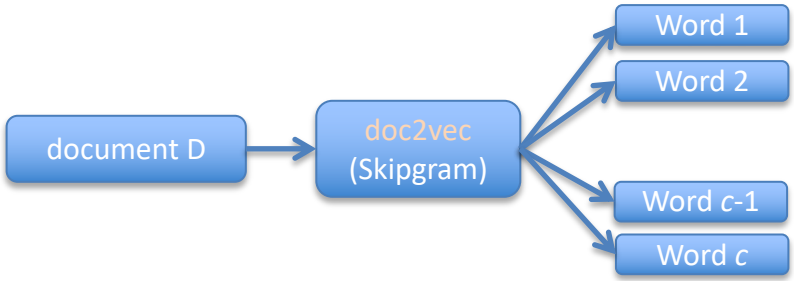
output: $sg_n^{(d)}$: Rooted subgraph of degree d around node n

begin
  $sg_n^{(d)} = \{\}$
  if $d = 0$ **then**
    $sg_n^{(d)} := \lambda(n)$
  else
    $\mathcal{N}_n := \{n' \mid (n, n') \in E\}$
    $M_n^{(d)} := \{$GETWLSUBGRAPH$(n', G, d-1) \mid n' \in \mathcal{N}_n\}$
    $sg_n^{(d)} := sg_n^{(d)} \cup$ GETWLSUBGRAPH
    $(n, G, d-1) \oplus sort(M_n^{(d)})$
  return $sg_n^{(d)}$

WL = Weisefeiler-Lehman

Netpro2vec[1]

Inspired by a neural document embedding model [2]

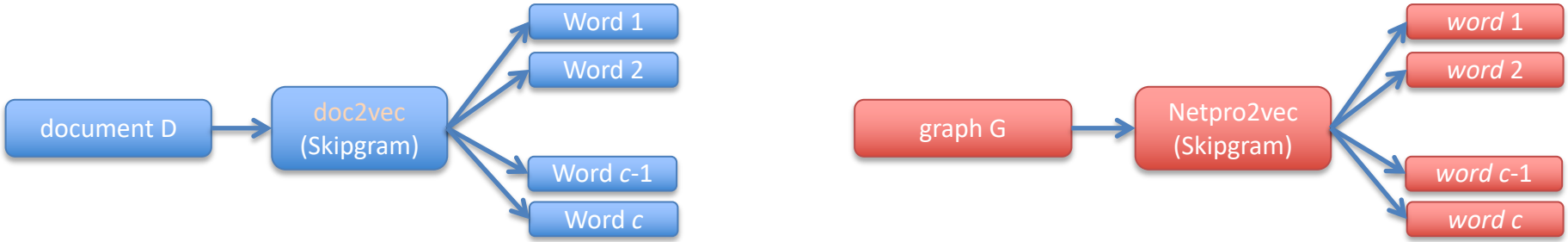


- Given a document *D*, sample *c* words from *D* and use them to learn a representation of *D*
- Doc2vec = feed-forward NN to learn distributed representations of document sequences

[1] I. Manipur et al., *Netpro2vec: a Graph Embedding Framework for Biomedical Applications*, IEEE/TCBB 2021
[2] Le and Mikolov, *Distributed Representations of Sentences and Documents*, Proc. ICML, 2014

Netpro2vec[1]

Inspired by a neural document embedding model [2]

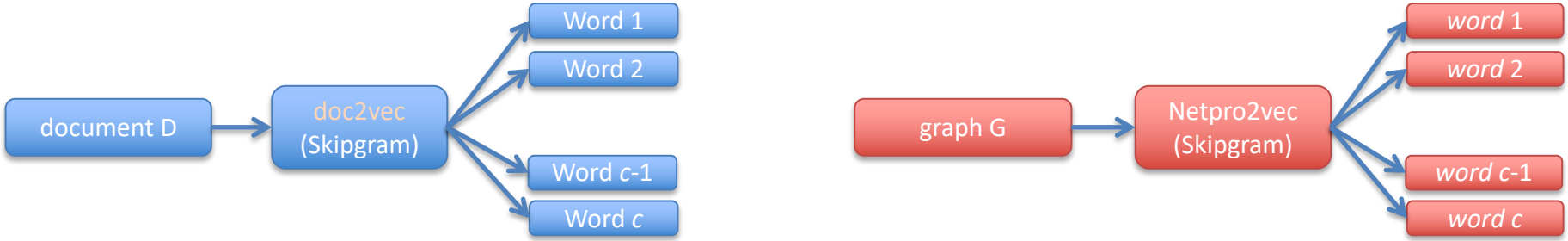


- Given a labeled graph *G*, *extract c words* from *G* and use them to learn a representation of *G*
- Doc2vec = feed-forward NN to learn distributed representations of graphs

[1] I. Manipur et al., *Netpro2vec: a Graph Embedding Framework for Biomedical Applications*, IEEE/TCBB 2021
[2] Le and Mikolov, *Distributed Representations of Sentences and Documents*, Proc. ICML, 2014

Netpro2vec[1]

Inspired by a neural document embedding model [2]



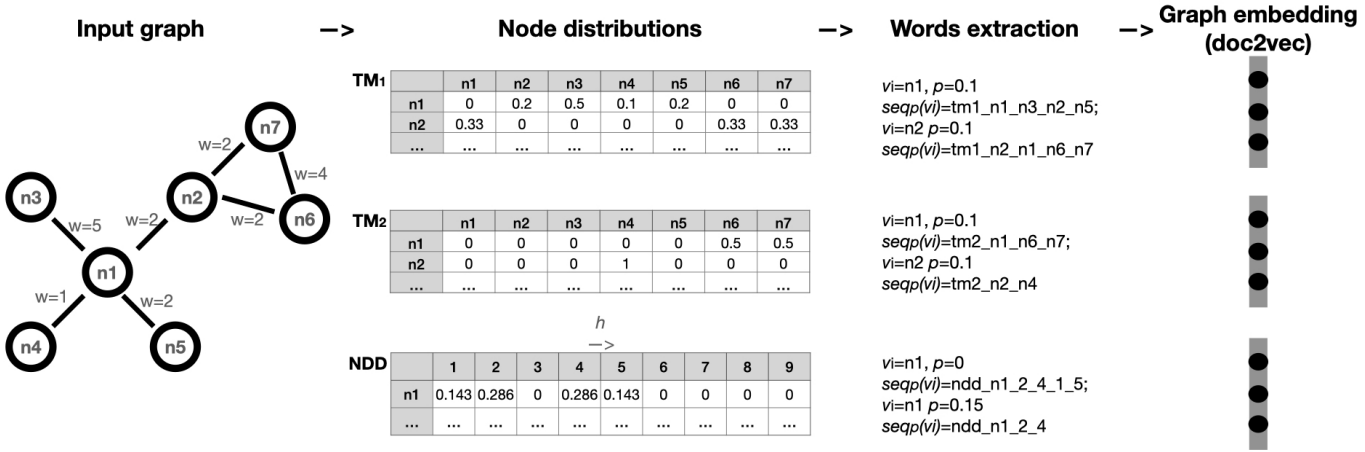
- A graph is viewed as a document
- But how can *words* be extracted by graphs?

[1] I. Manipur et al., *Netpro2vec: a Graph Embedding Framework for Biomedical Applications*, IEEE/TCBB 2021
[2] Le and Mikolov, *Distributed Representations of Sentences and Documents*, Proc. ICML, 2014

Netpro2vec[1]: *words* extraction

Use probability distributions (PDs) to represent each graph (e.g., NDD, TM)

Words are obtained by concatenating the labels of the nodes whose PDs exceed a threshold *p*



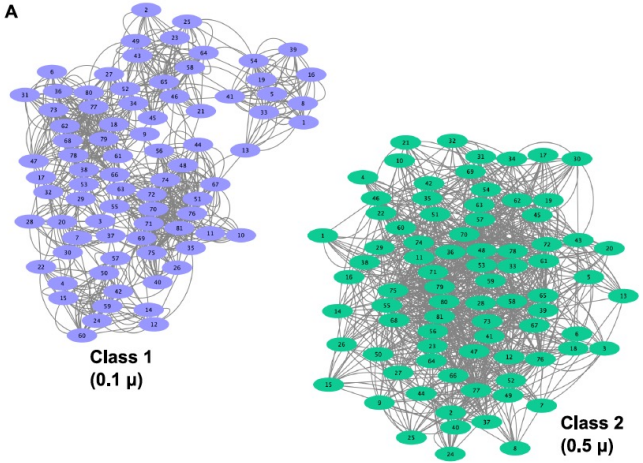
[1] I. Manipur et al., *Netpro2vec: a Graph Embedding Framework for Biomedical Applications*, IEEE/TCBB 2022
https://github.com/cds-group/Netpro2vec

LFR Dataset

Generated[1] using

Lancichinetti–Fortunato–Radicchi (LFR) software[2]

- 1600 undirected and unweighted graphs
  - A. 600 with μ=0.1
  - B. 1000 with μ=0.5
- 81 nodes for each graph



[1] Gutiérrez-Gómez and Delvenne, DAE, *Applied Network Science*, 2019
[2] Lancichinetti et al, *Physical review E*, 2008

Dataset available at https://github.com/leoguti85/GraphEmbs

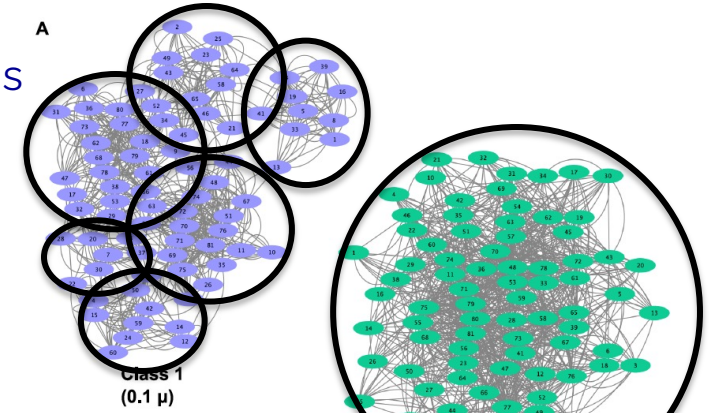
LFR Dataset

Generated[1] using

Lancichinetti–Fortunato–Radicchi (LFR) software[2]

- 1600 undirected and unweighted graphs
  - A. 600 with μ=0.1
  - B. 1000 with μ=0.5
- 81 nodes for each graph

μ controls the strength of the community arrangements



6 communities

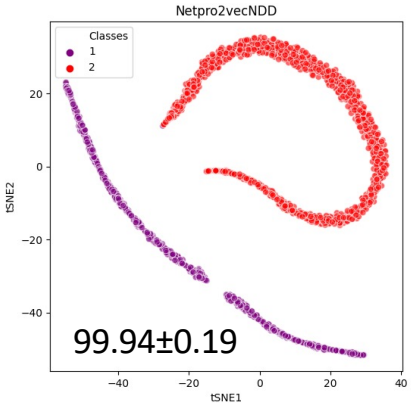
5 communities

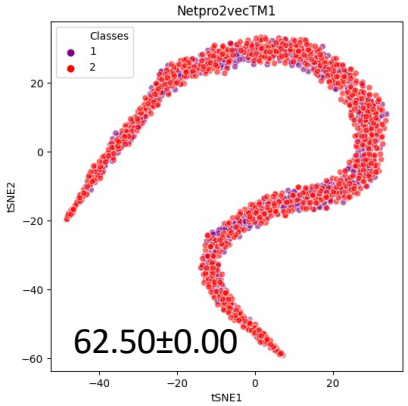
[1] Gutiérrez-Gómez and Delvenne, DAE, *Applied Network Science*, 2019

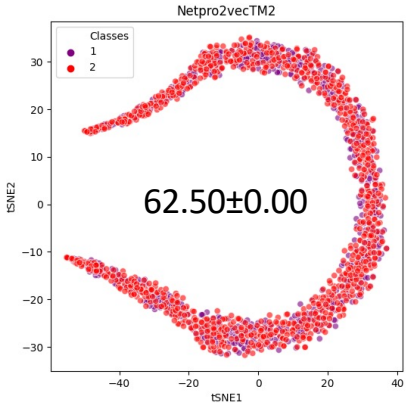
[2] Lancichinetti et al, *Physical review E*, 2008

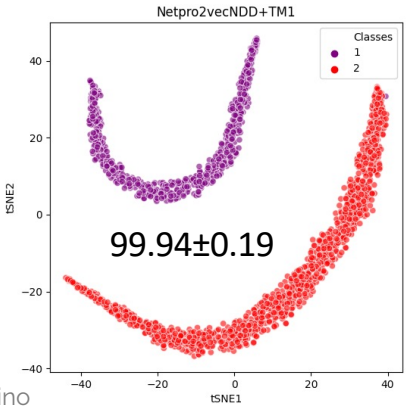
Dataset available at https://github.com/leoguti85/GraphEmbs

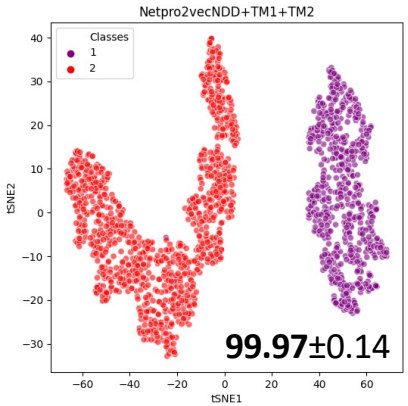
Netpro2vec: t-SNE visualization of LFR

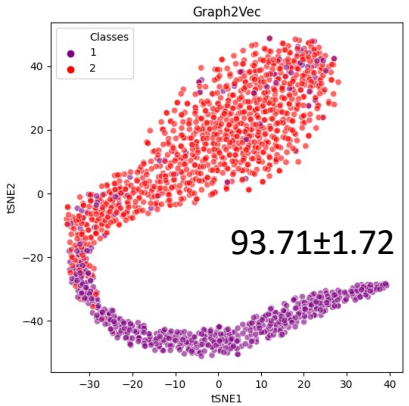












Considerations (Netpro2vec)

- Large-scale experiments on several biomedical datasets
- Performance results are comparable/outperform those of other WGL methods
- The chosen representations (e.g., NDD, TM) for extracting graph *words* analyze not only single node neighborhoods but a wider space
- Context-independent → can be applied to any type of data

I. Manipur et al., *Netpro2vec: a Graph Embedding Framework for Biomedical Applications*, IEEE/TCBB 2022

Research directions

- Include additional node and edge properties to better describe the graphs.
- Adopt the same framework for different tasks such as clustering, regression, link prediction, and variable selection.
- Embed high order models, such as hypergraphs.

Summary

Whole-graph embedding for classification and clustering problems involving sets of networks representing instances of the system under study

- Different approaches to whole-graph embedding
  - Matrix factorization
  - Graph Kernels
  - Neural Networks
  - Probability distribution measures
- Datasets and links to publicly available software https://github.com/cds-group/GraphDatasets
- Setting method parameters
- Netpro2vec
- Enhancement of Graph2vec