Data Science Lab

Data Representation Learning is critical step in data analysis since it can entangle and hide the different explanatory factors of variation behind the data.

Specifically, for **networked data analysis**, how to learn the intrinsic structure and discover valuable information from data becomes more and more urgent, important and challenging.

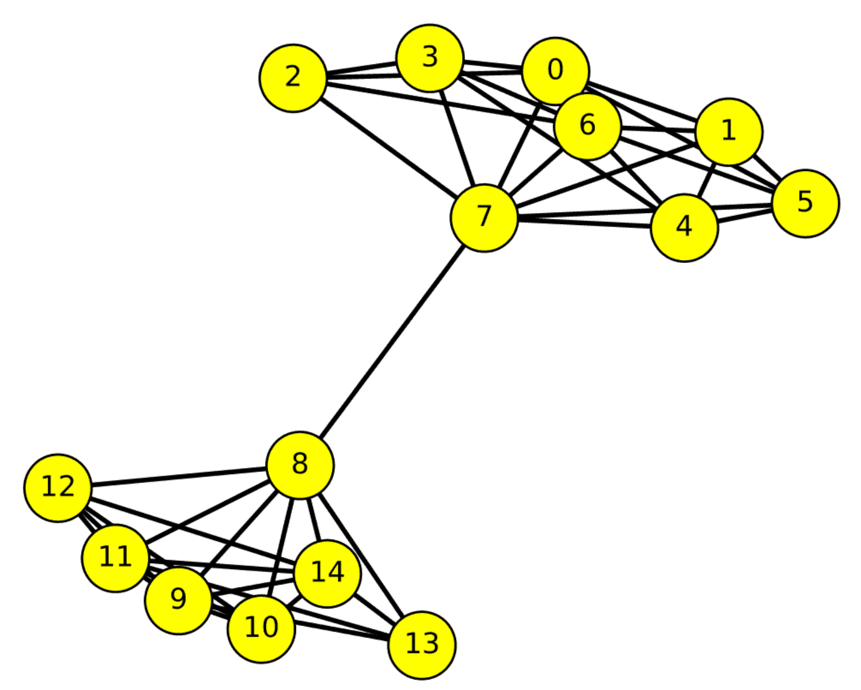
Goal of this course:

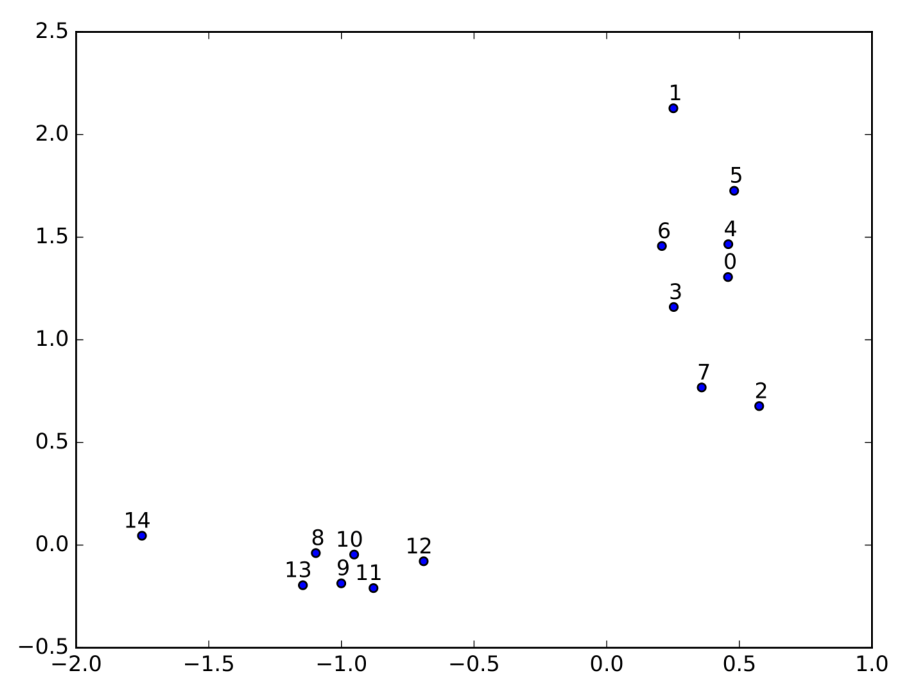
Implement and evaluate a Network Representation Learning framework

Chair of Data Science

Fatemeh Salehi Rizi

Network Representation Learning





Organization

- Project work in Teams of 5 people

- 3 Phases:
  1. Problem definition
  2. Implementation (code on gitlab)
  3. Evaluation

- Written report per phase (single document, section per phase)
  - Template at: https://github.com/fatemehsrz/Data_Science_Lab
  - Phase 1: (3-4) pages
  - Phase 2: (3-4) pages
  - Phase 3: (3-4) pages

- Final Presentation (20min presentation each team)

- Gitlab at: https://gitlab.fim.uni-passau.de/users/sign_in

Grading

- Submission Deadlines:
  - Phase 1 : Report 05.12.2019
  - Phase 2 : Report 05.01.2020
  - Phase 3 : Report 05.02.2020
  - Presentations (04.02 to 13.02.2020)
  - Final Report 30.02.2019

Missing a deadline means failing the report

- Final talk must be presented by all team members

- Final Grade is a weighted average of the team grades in phase 1-3

  - 20% from the phase 1
  - 40% from the phase 2
  - 40% from the phase 3

Table of Content

- Introduction
  - - Graph Data
  - - Networkx and Syntactic Graphs
  - - Autoencoders with Keras
  - - Feedforward Networks with PyTorch
  - - Matrix decomposition with SVD
- Graph Representation Learning
  - - Word2Vec (Gensim)
  - - DeepWalk [1] (Gensim)
  - - LINE [2] (PyTorch)
  - - SDNE [3] and HOPE [4] (Optional)
- Graph Evaluation Tasks
  - - Link Prediction
  - - Node Classification
  - - Graph Clustering (optional)
  - - Network Visualization
  - - Evaluation Metrics (F1, AUC, NMI)

[1] https://arxiv.org/pdf/1403.6652.pdf
[2] https://arxiv.org/pdf/1503.03578.pdf
[3] https://www.kdd.org/kdd2016/papers/files/rfp0191-wangAemb.pdf
[4] https://www.kdd.org/kdd2016/papers/files/rfp0184-ouA.pdf

Projects

1) A Unified Framework for Graph Representation Learning

- Take graph datasets available at [1]
- Implement a framework which generates embeddings for node2vec [2], HARP [3], WalkLets [4] and struc2vec [5]
- Evaluate representations via graph mining tasks (Link Prediction, Node Classification, Network Visualization)

2) Graph Embedding with Self-clustering

- Take graph datasets available at [1]
- Implement GEMSEC [6] in PyTorch
- Evaluate communities comparing to ComE [7], DANMF [8]
- Evaluate representations via graph mining tasks (Node Classification)

[1] https://github.com/fatemehsrz/Datasets_Graph_Embedding

[2] https://cs.stanford.edu/~jure/pubs/node2vec-kdd16.pdf

[3] https://arxiv.org/pdf/1706.07845.pdf

[4] https://arxiv.org/pdf/1605.02115.pdf

[5] https://arxiv.org/pdf/1704.03165.pdf

[6] https://arxiv.org/pdf/1802.03997.pdf

[7] https://sentic.net/community-embedding.pdf

[8] https://smartyfh.com/Documents/18DANMF.pdf

Papers Source Codes

- node2vec: https://github.com/aditya-grover/node2vec
- HARP: https://github.com/GTmac/HARP
- WalkLets: https://github.com/benedekrozemberczki/walklets
- Struc2vec: https://github.com/leoribeiro/struc2vec
- GEMSEC: https://github.com/benedekrozemberczki/GEMSEC
- ComE: https://github.com/vwz/ComE
- DANMF: https://github.com/benedekrozemberczki/DANMF

Surveys

1) https://arxiv.org/pdf/1709.07604.pdf
2) https://arxiv.org/pdf/1705.02801.pdf
3) https://arxiv.org/pdf/1801.05852.pdf
4) https://arxiv.org/pdf/1711.08752.pdf

Schedule

Session	Lecture Topic
1: 22/10/19	Introduction
2: 24/10/19	Introduction
3: 29/10/19	Graph Representation Learning
4: 31/10/19	Graph Representation Learning
5: 05/11/19	Graph Mining Tasks
6: 07/11/19	Graph Mining Tasks
12/11/2019 to 13/02/2020	Team work and Presentations

Python Scientific/Data Science Stack

- Data Science Handbook (https://jakevdp.github.io/PythonDataScienceHandbook/)
- Python
  - Cheatsheets - https://ehmatthes.github.io/pcc/cheatsheets/README.html
  - Setup with Anaconda - https://github.com/molybdaen/data-science-tutorials/blob/master/ws17.md
  - Details - https://github.com/zieglerk/python-tutorials/blob/master/DSiP-2-Python-Programing-Basics.ipynb
- Jupyter Notebooks
- Numpy
  - Cheatsheet - https://github.com/kailashahirwar/cheatsheets-ai/blob/master/Numpy.png
  - Details - https://github.com/zieglerk/python-tutorials/blob/master/DSiP-3-NumPy.ipynb
- Scipy
  - Cheatsheet - https://github.com/kailashahirwar/cheatsheets-ai/blob/master/Scipy.png
  - Details - https://github.com/zieglerk/python-tutorials/blob/master/DSiP-4-Scipy.ipynb

Python Scientific/Data Science Stack

- Matplotlib
  - Cheatsheet - https://github.com/kailashahirwar/cheatsheets-ai/blob/master/Matplotlib.png
  - Details - https://github.com/zieglerk/python-tutorials/blob/master/DSiP-5-Matplotlib.ipynb
- Pandas
  - Cheatsheet - https://github.com/kailashahirwar/cheatsheets-ai/blob/master/Pandas-3.png
  - Details - https://github.com/zieglerk/python-tutorials/blob/master/DSiP-6-Pandas.ipynb
- Scikit-Learn
  - Cheatsheet - https://github.com/kailashahirwar/cheatsheets-ai/blob/master/Scikit%20Learn.png
  - Details - https://scikit-learn.org/stable/
- [Keras/PyTorch]

Python Scientific/Data Science Stack

Sources (explore them for further information)

- Cheatsheets from https://github.com/kailashahirwar/cheatsheets-ai (further useful cheatsheets available)
- Details (mostly) from former course material
    - https://github.com/molybdaen/data-science-tutorials
    - https://github.com/zieglerk/python-tutorials
    - https://github.com/mgrani/LODA-lecture-notes-on-data-analysis