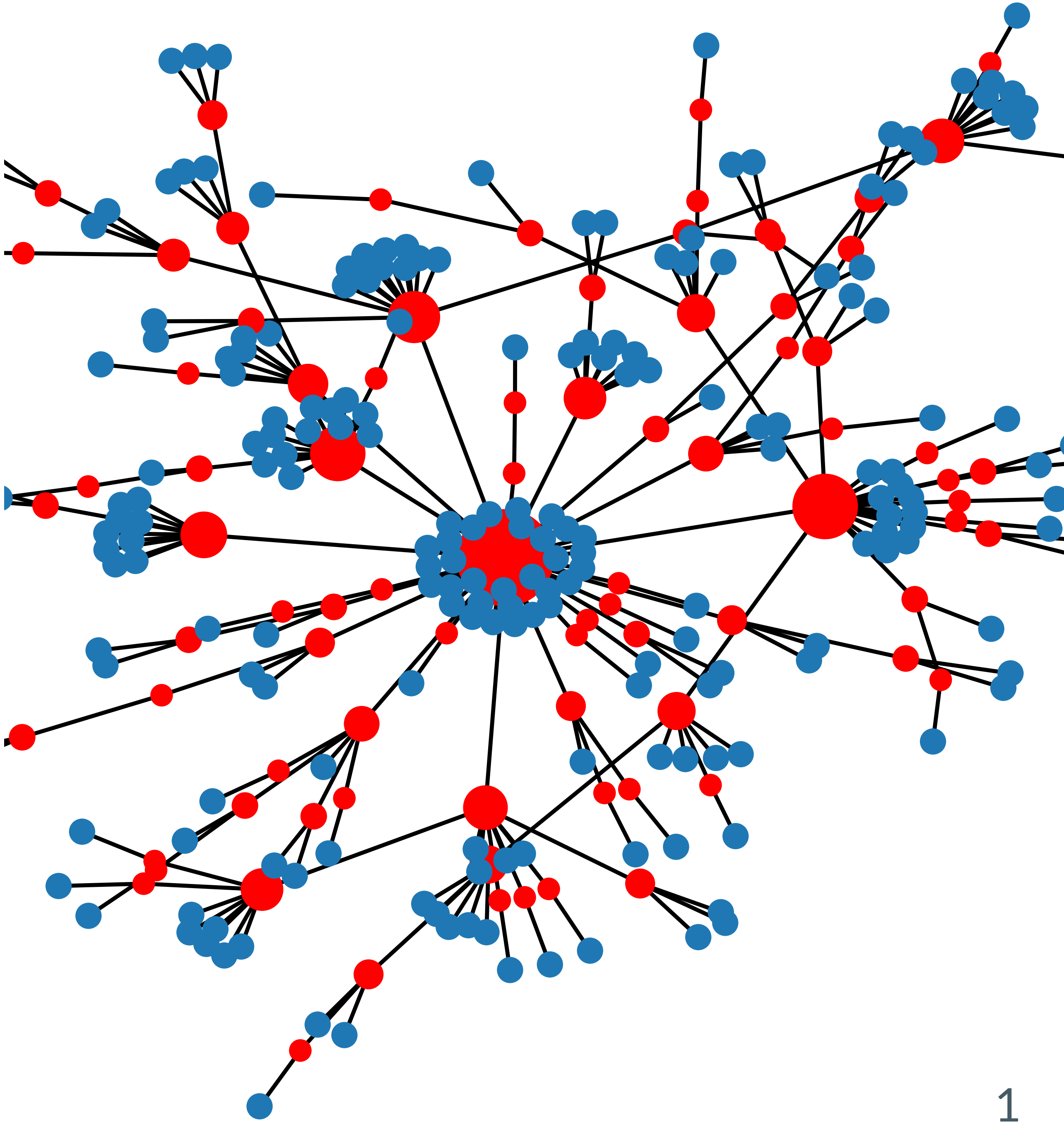
Graph models and generators

Anna Yakovlieva

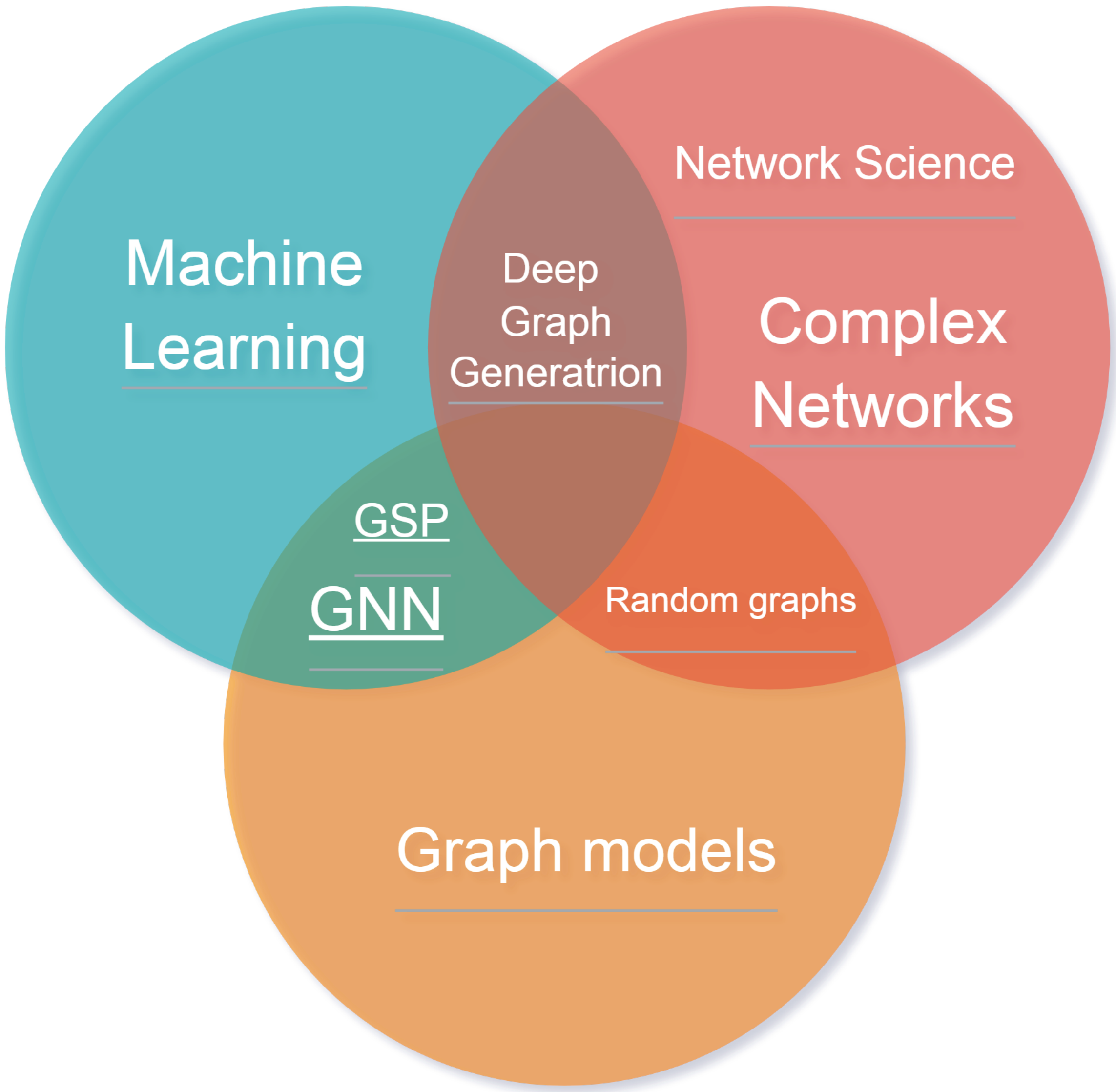
24.11.2021

github.com/yakovlieva00/sm445/slides.pdf



Contents

1. Complex Networks
2. Topological characteristics of graphs
3. Tasks for complex networks
4. Graph models
5. Generators
6. Why DGG?
7. DGG and Applications



Complex Networks

- Facebook, Twitter, Youtube, TikTok
- arxiv.org, World Wide Web
- Transport, etc.
- COVID-19:
    - Policy forecasting
    - Comparison of countries
    - Anti-vaxers management
    - Detection of mutations

Local characteristics of graphs

- nodes and edges
- degree of a node,
- incidence matrix
- digraphs
- subgraphs
- cliques
- paths
- distances

Topological characteristics of graphs

- mean degree,
- degree distribution
- mean path
- eccentricity
- radius and diameter
- clustering coefficient
- efficiency

The structure of complex networks

- centrality
- degree of centrality
- betweenness centrality
- subgraphs and motives
- prestige
- PageRank
- power ratings

Tasks for complex networks

- resistant to attacks, endurance assessment, safety
- assessment, synchronization in networks
- self-organized criticality
- cascading damage, and spread of infections
- maximizing the spread of influence
- community search

Graph models

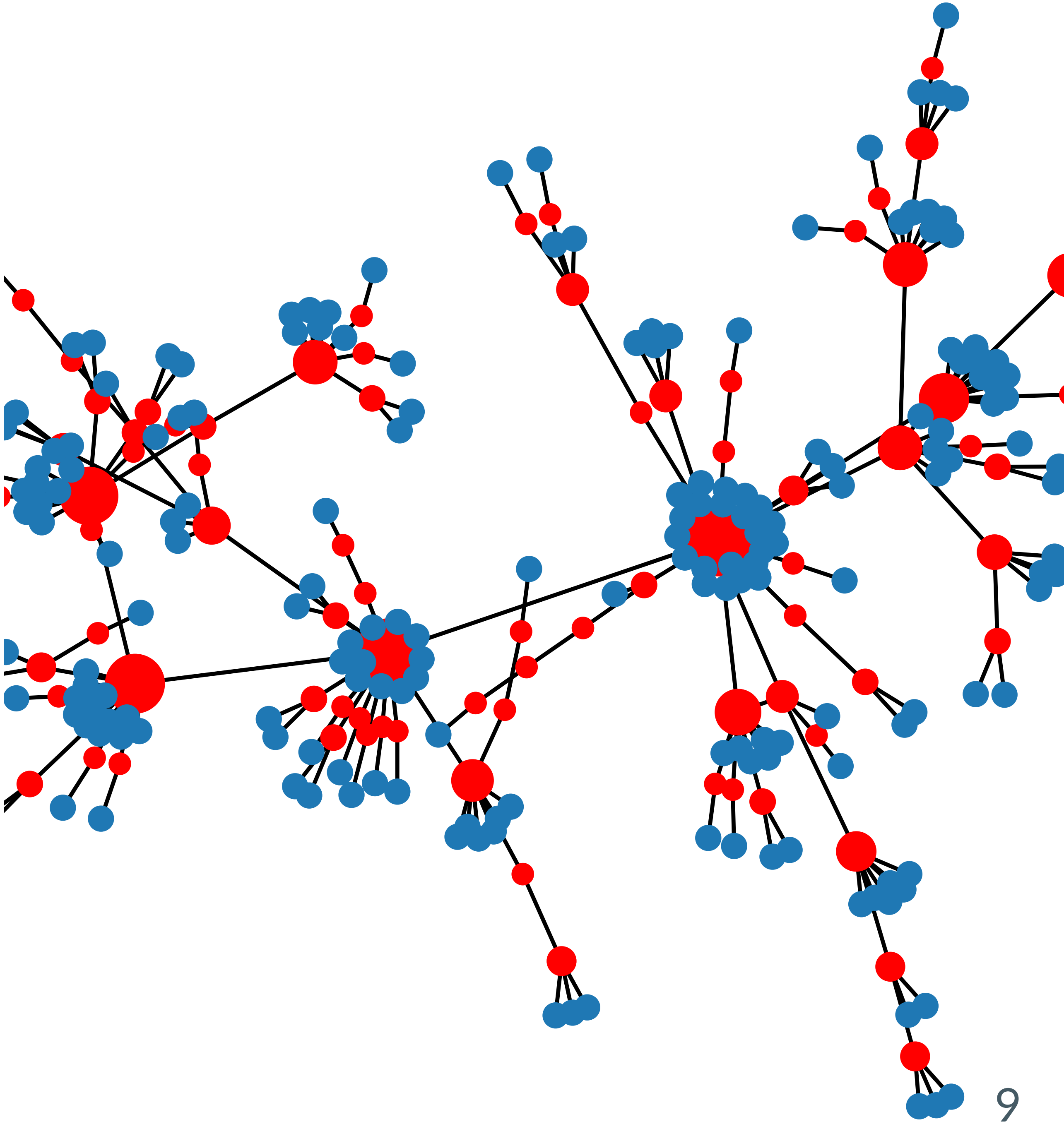
- Small World Model (Watts-Strogatz)
- Erdos-Renyi model
- Barabasi-Albert model
- Bollobashi-Riordan model
- LCD model
- Buckley-Ostguts model
- Mori model, Copy model, Chung-lu model, Yanson-Luchak model, Kronecker model

Barabasi-Albert model

Preferential attachment model

- nodes: n , edges: $k \cdot n$
- diameter $d \sim 5 - 7$
- probability nodes of degree d (power law):

$$p_d = d^{-\gamma}$$



Deep Graph Generation Models

Why?

Disadvantages of old models:

- can simulate only some of the statistical characteristics of real networks
- limited capabilities for modeling complex structures
- can't simulate heavy tails of distributions
- can't learn the graph structure from data

Deep Graph Generation Models

The modern approach is to form a graph model based on a training set of graphs. Instead of formulating a mathematical model, we use a given set of graphs to obtain a graph generation model with similar characteristics.

- VAE model
- GAN model
- Autoregressive models
- RL-based models

VAE model

VAEs are the most popular approach in deep generative models.
In order to train VAE we use encoder and decoder:

$$Z \sim q_\theta(Z|G), \hat{A} \sim p_\theta(A|Z), Z \sim p(Z)$$

The idea in VAE is to use an encoder and a decoder together, training them to reconstruct the input graph using a posterior distribution to generate output graph G based on a random hidden variable Z.

GNN is used for enocoder, MLP is used for decoder.

GAN model

We define a generator and a discriminator and train together in a competitive game. Generator is a trainable function $g_\theta : R^d \to X$.
The generator network is trained to generate realistic graphs $x \in X$.
At the same time, we define the discriminator $d_\theta : X \to [0, 1]$. The purpose of the discriminator is to separate real graphs and graphs obtained by the generator.
The discriminator determines the probability that $x \in X$ is a fake.
MLP is used for as a generator, GNN is used as a discriminator for classification.

Evaluating DGG

How to evaluate the quality of DGG models?

Comparison of different statistical indicators
of the generated graphs:

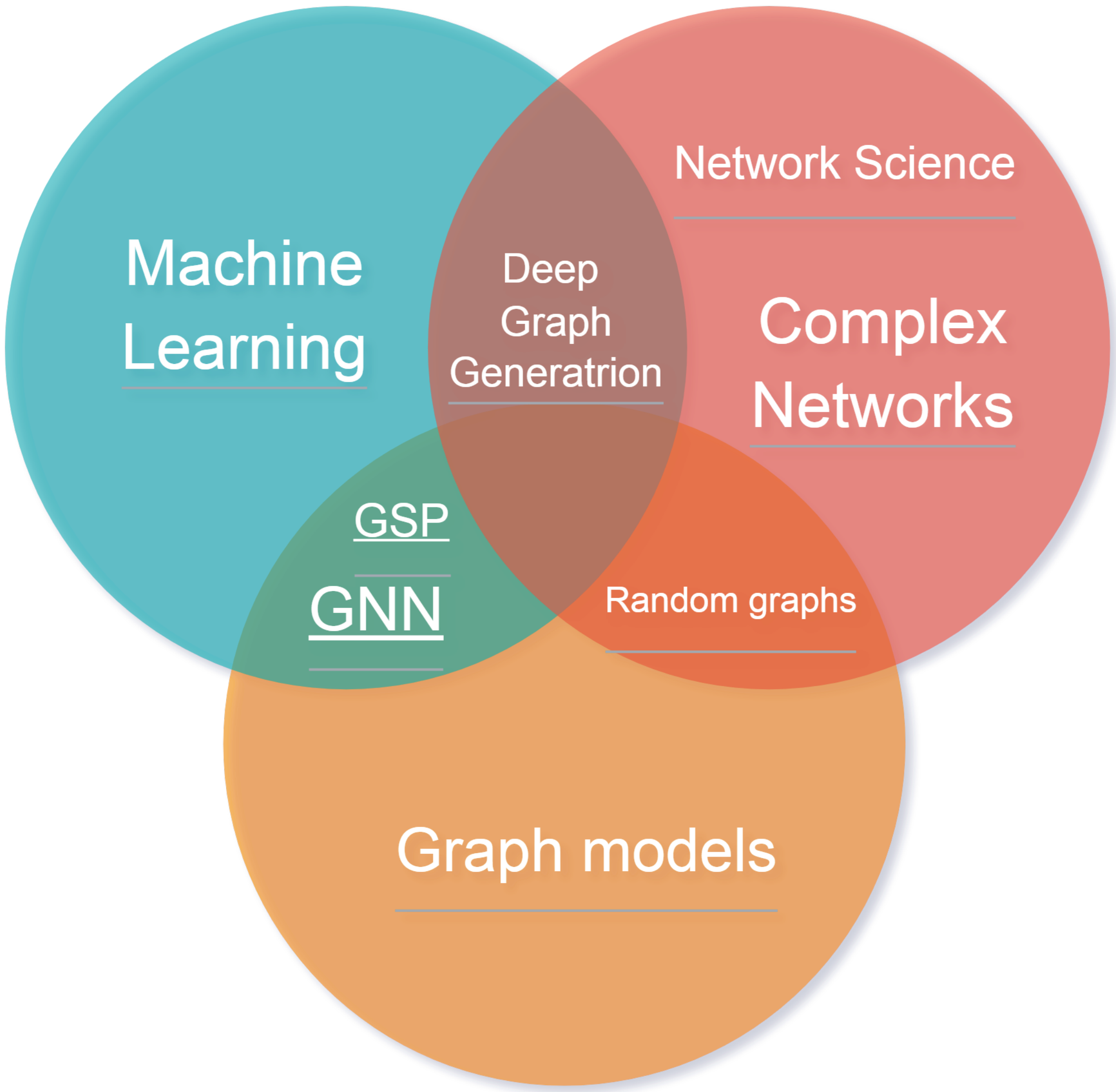
- distribution of degrees
- cluster coefficients
- distance between a group of synthetic graphs and graphs of the test sample.

DGG Applications

- generation in molecular chemistry
  - drug design
  - material discovery
- non-molecular generation:
  - social network
  - AMR (semantic graphs in NLP)
  - scene generation

Summary

- Graph models play a key role in graph learning
- Modern DGG models build graph models from data
- The development of graph models is essential for disease control



Thank you for your attention!