Discreteness in Generative Models

-- *Generating Graphs*

Hao Dong

Peking University

Table of contents

北京大學
PEKING UNIVERSITY

- Introduction
- Generating graphs as sequences
- Autoregressive graph generating
- Global graph generating

Why it matters

- The typical application of graph neural network is representation learning, e.g. mining hidden information from social networks, knowledge graphs with unsupervised node embedding.
  - Graph generating is infeasible and not very useful, for graphs such as social networks, which may contain billions of nodes

- A Comprehensive Survey on Graph Neural Networks, 2019
  - “The majority of graph autoencoders for graph generation are designed to solve the molecular graph generation problem, which has a high practical value in drug discovery.”

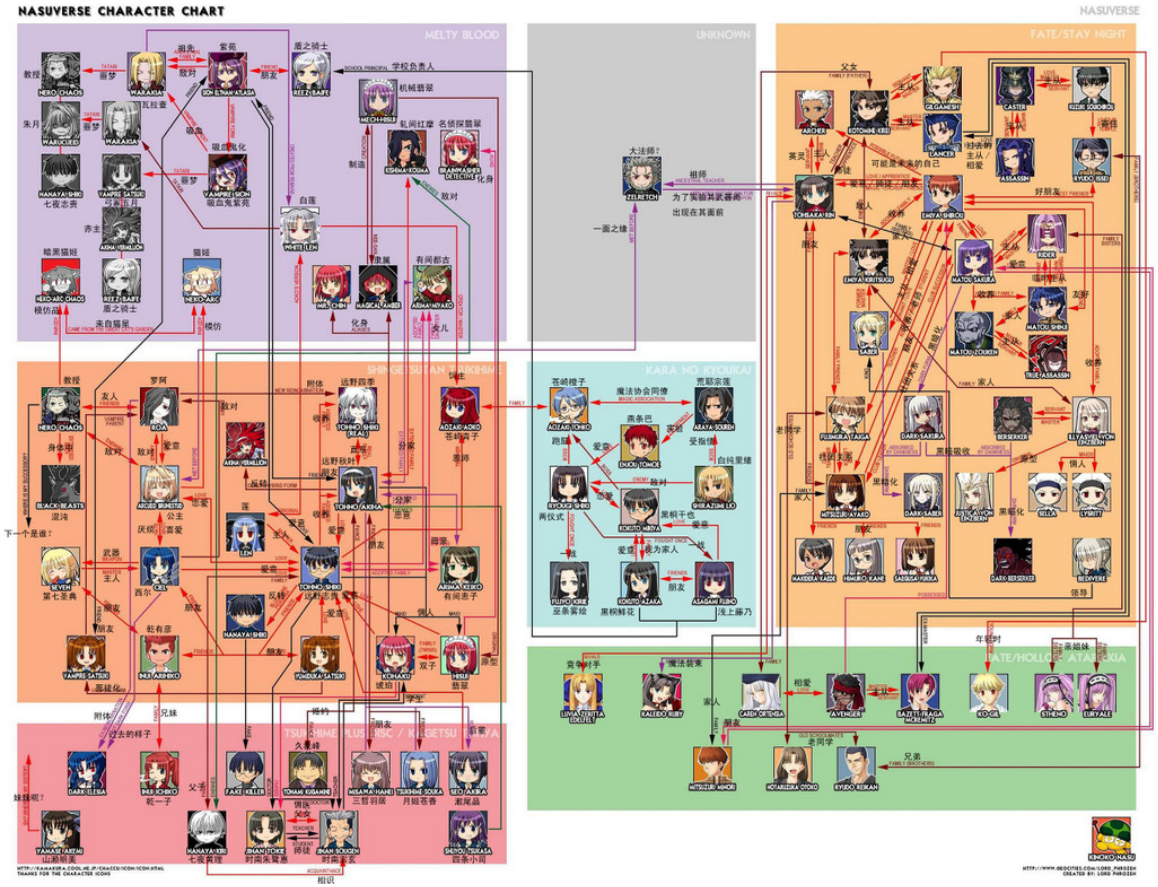
- Furthermore, graph generating model can be used for augmenting graph datasets.

Various graphs

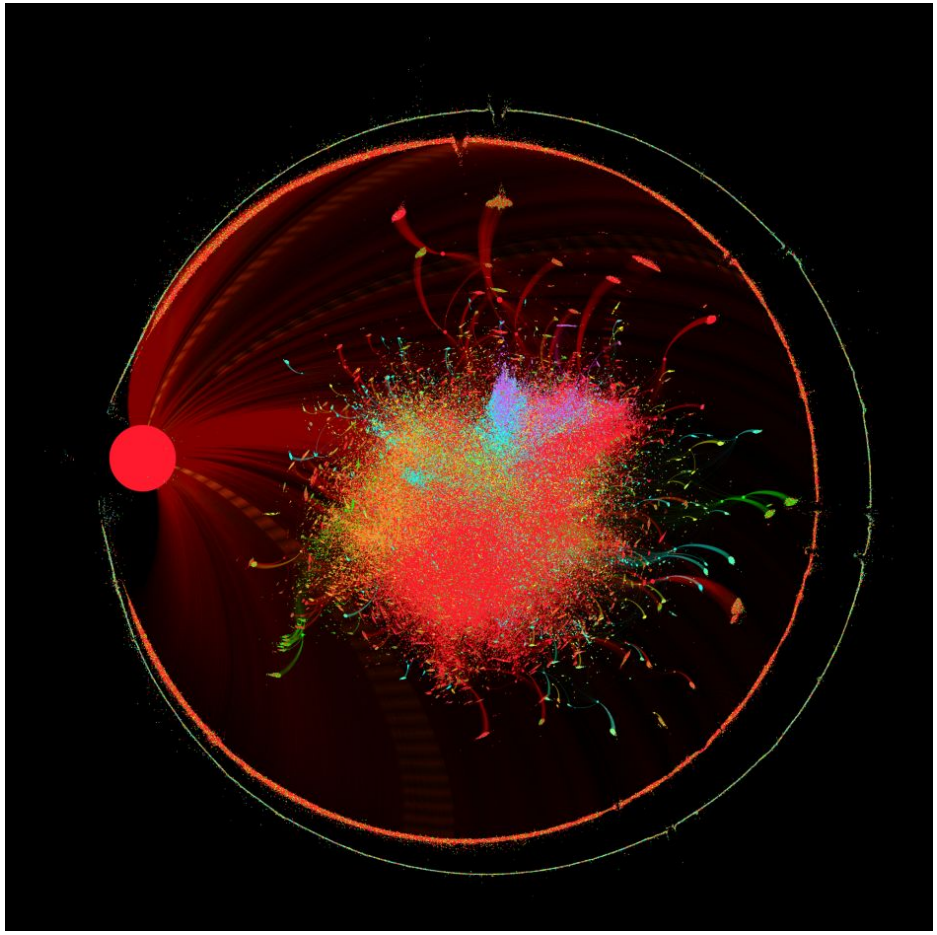
- Graphs can be interpreted in many ways. In another words, many data modality can be represented as graphs.

- Notice how the abstract concepts of node, edge, node label, edge label, etc. are grounded in each case.

Social Networks



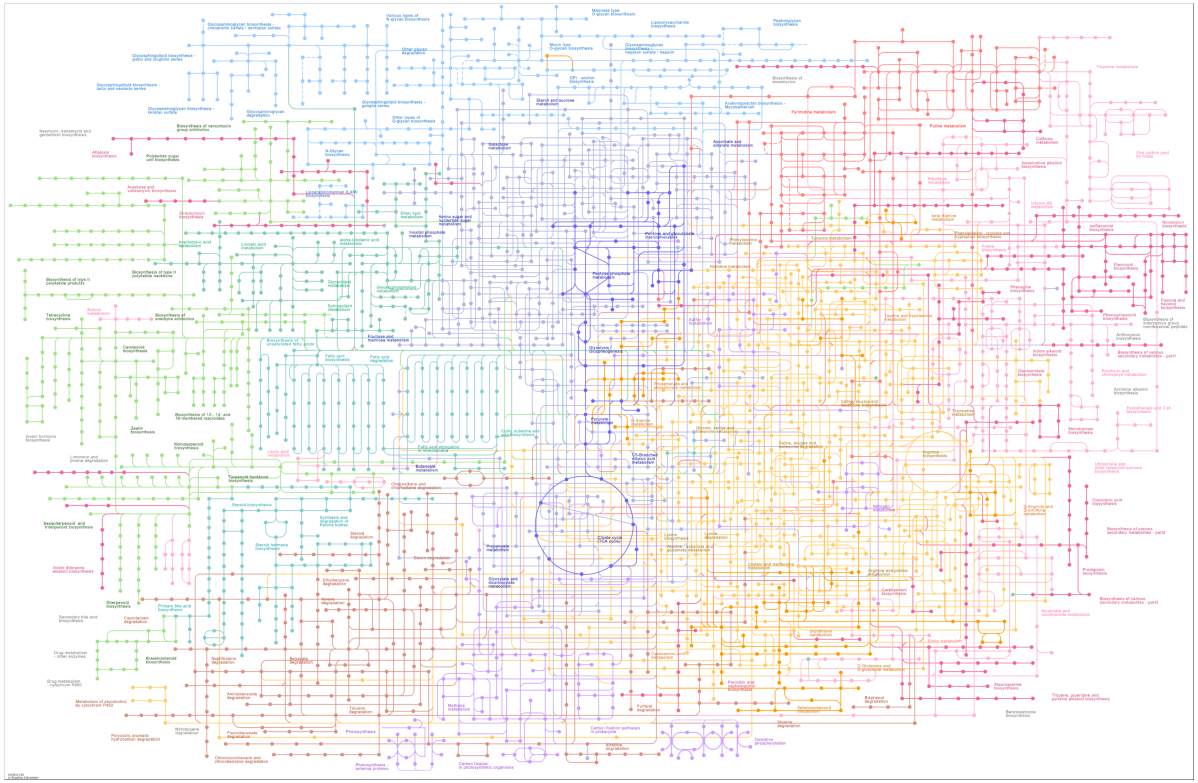
Citations



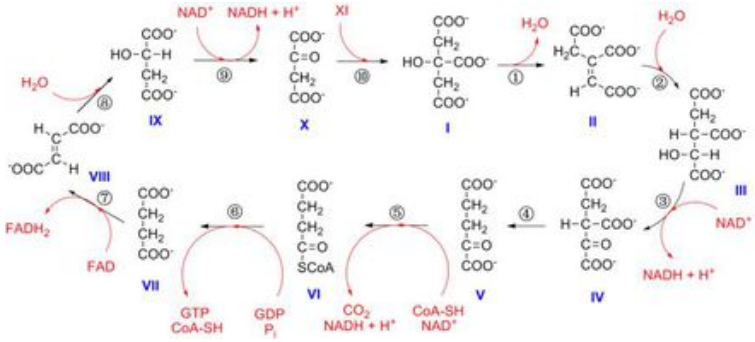
- Graph visualization of citations of papers published on Nature

Biology	(57.28%)
Physics	(11.17%)
Medicine	(9.11%)
Chemistry	(7.23%)
Psychology	(7.05%)
Geology	(4.33%)
Computer Science	(1.49%)
Mathematics	(1.05%)
Economics	(0.31%)
Engineering	(0.31%)
Sociology	(0.27%)
Materials Science	(0.23%)

Metabolic pathway



- A local view of the tricarboxylic acid cycle that is fundamental to respiration



Various graphs

- Molecules
  - will be discussed in details later
  - especially small organic molecules, large molecules (proteins and DNAs) are better modeled as sequences
- Knowledge Graphs
- Many computation problems
  - Reducibility Among Combinatorial Problem, 1972
  - many combinatorial problems (in particular, all NP complete problems) can be converted to problems about graphs

北京大學
PEKING UNIVERSITY

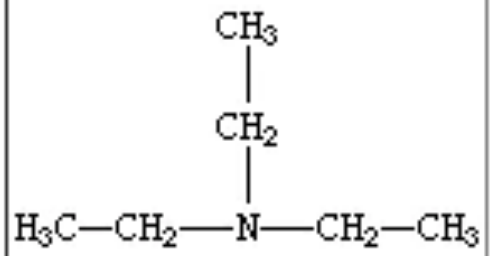
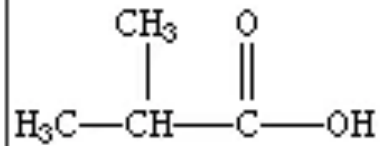
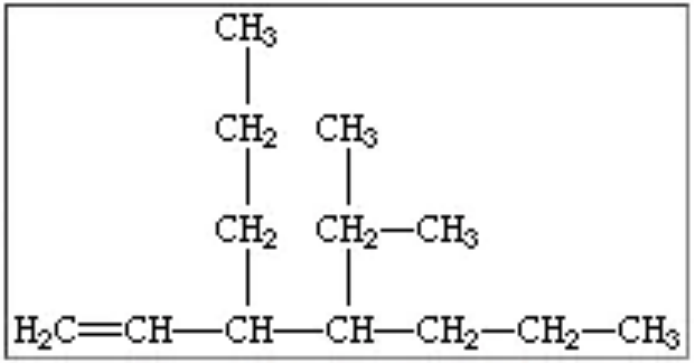
- Introduction
- Generating graphs as sequences
- Autoregressive graph generating
- Global graph generating

Generating graphs as sequences

- There are many general ways to serialize molecules, e.g. SMILES
  - notice that this method can be applied to general graphs with discrete labels
- Use brackets to indicate side chains, and numbers to identify rings
- Hydrogens are often omitted.

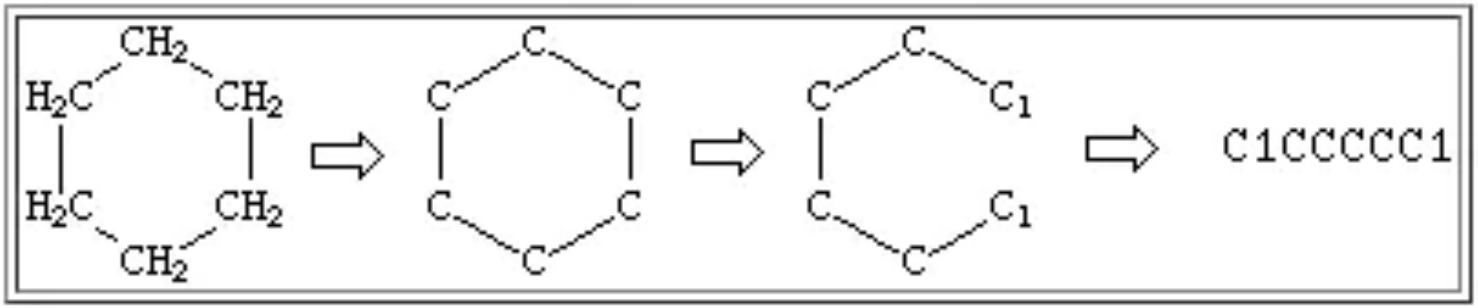
Generating graphs as sequences

- A linear molecule can be represented as a linear string.
- Brackets to indicate side chains, such that trees can be represented.

		
CCN(CC)CC	CC(C)C(=O)O	C=CC(CCC)C(C(C)C)CCC
Triethylamine	**Isobutyric acid**	**3-propyl-4-isopropyl-1-heptene**

Generating graphs as sequences

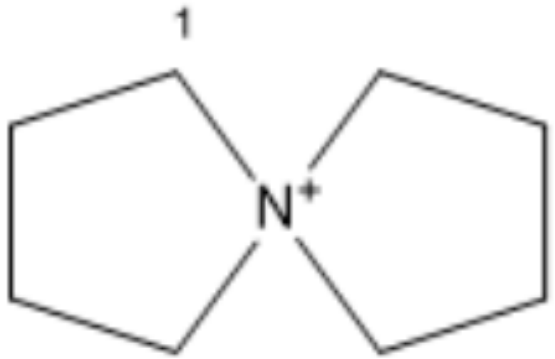
- Use numbers to indicate rings (loops), two consecutive atoms with the same number are connected.



- Molecule on the right:

C1CCC[N+]12CCCC2 or

C1CCC[N+]11CCCC1



Exercise

- Draw graph representation of SMILES, and notice their difference
  - CCCCCC
  - C1CCCCC1
  - C1CC1C2CC2
  - C1CC2C1CC2
  - C12C3C4C1C2C34

Generating graphs as sequences

- A sequence generating model can then be directly applied on serialized graphs

- A graph can be serialized in different ways, although there are standards for choosing the beginning point and choosing the main chain to ensure a unique serialization for ease of comparison, the specification is however, artificial, and sometimes impedes learning

北京大學
PEKING UNIVERSITY

- Introduction
- Generating graphs as sequences
- Autoregressive graph generating
- Global graph generating

Autoregressive graph generating

- Learning deep generative models of graphs, 2018
- Iteratively connects nodes and edges to the graph
- Particularly useful when a part of the target is known

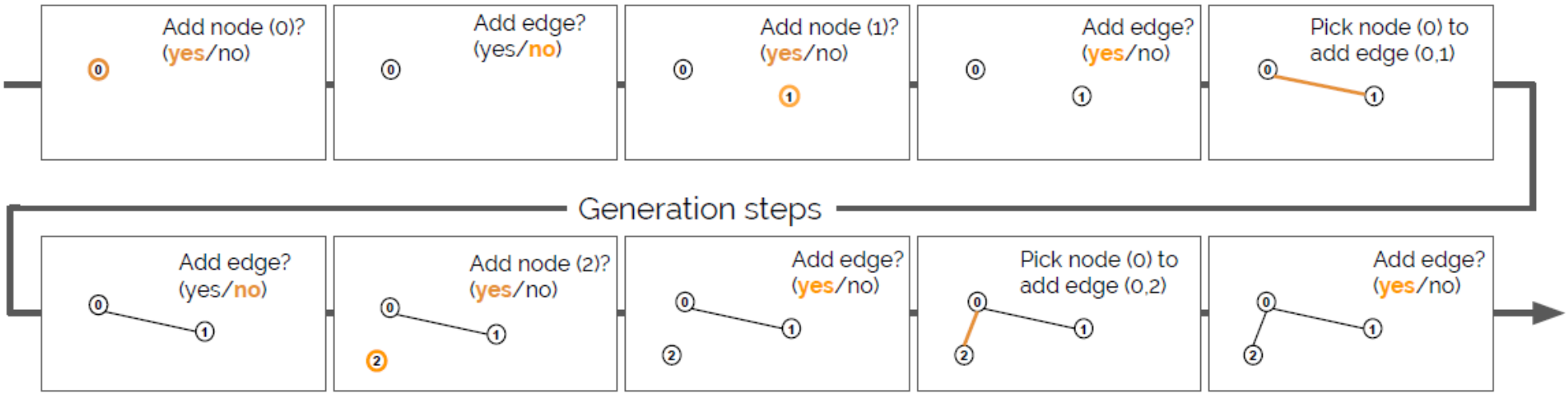


Figure 1. Depiction of the steps taken during the generation process.

Autoregressive graph generating

Table 2. Molecule generation results. N is the number of permutations for each molecule the model is trained on. Typically the number of different SMILES strings for each molecule < 100.

Arch	Grammar	Ordering	N	NLL	%valid	%novel
LSTM	SMILES	Fixed	1	21.48	93.59	81.27
LSTM	SMILES	Random	< 100	**19.99**	93.48	83.95
LSTM	Graph	Fixed	1	22.06	85.16	80.14
LSTM	Graph	Random	$O(n!)$	63.25	91.44	91.26
Graph	Graph	Fixed	1	20.55	**97.52**	90.01
Graph	Graph	Random	$O(n!)$	58.36	95.98	**95.54**

Table 3. Negative log-likelihood evaluation on small molecules with no more than 6 nodes.

Arch	Grammar	Ordering	N	Fixed	Best	Marginal
LSTM	SMILES	Fixed	1	17.28	15.98	15.90
LSTM	SMILES	Random	< 100	**15.95**	15.76	15.67
LSTM	Graph	Fixed	1	16.79	16.35	16.26
LSTM	Graph	Random	$O(n!)$	20.57	18.90	15.96
Graph	Graph	Fixed	1	16.19	**15.75**	15.64
Graph	Graph	Random	$O(n!)$	20.18	18.56	**15.32**

- Limitations
  - Still dependent on an artificial ordering, the ordering has more entropy than the graph itself.

北京大學
PEKING UNIVERSITY

- Introduction
- Generating graphs as sequences
- Autoregressive graph generating
- Global graph generating

Global graph generating

- Motivation: to avoid arbitrary ordering imposed in serialized and autoregressive graph generating
- GraphVAE: Towards generation of small graphs using variational autoencoders. 2018
- For graphs of no more than k nodes, predict the edges as a kxk dense matrix

Global graph generating

- Similar to an image VAE, the adjacency matrix is encoded into a posterior distribution of latent variable of fixed dimension d.

- The order of nodes doesn't matter, when comparing output of the decoder to the input, the graphs are compared up to isomorphism, e.g. $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ are considered equivalent.

Limitation

- As the cost of avoiding arbitrary ordering, the problem of graph matching is introduced, i.e., the output graph must be aligned with the ground truth in order to compute the loss

- The graph isomorphism problem is not known to be solvable in polynomial time nor to be NP-complete

- A practical second order matching algorithm has memory cost of $O(k^4)$, which severely limits the scale of problems it can apply on

To avoid the limitation:

- Without graph matching, global graph generating can be more performant (up to $O(k^2)$)
- Graph GAN, discriminator loss doesn't need aligning a pair of graphs to compute
- For graph VAEs, randomly shuffle the node order of input, and requires the node order of the output being the same.

Summary

- Introduction
- Generating graphs as sequences
- Autoregressive graph generating
- Global graph generating