2019sp-cs-8501-Deep2Read Scribe Notes: Junction Tree Variational Autoencoder for Molecular Graph Generation

Scribe: Arshdeep Sekhon

June 1, 2019

1 Motivation

Junction Tree VAE(JT VAE) generates molecular graphs in a two step process: First it generates a tree scaffold structure over predefined substructures, and then decodes the tree into a molecular graph. The key advantage is that this ensures chemical validity at all steps of incremental generation of the graph as only valid substructures are added.

2 Method

Instead of generating molecules node by node in a sequential manner, JT-VAE generates molecules substructure by substructure, which ensures chmeical validity at every step. The VAE first generates a tree structured object (a junction tree) that represents the scaffold of subgraph components and their coarse relative arrangements. . In the second phase, the subgraphs are assembled together into a molecular graph.

2.1 Junction Tree

This maps a graph G into a *junction tree* by contracting certain vertices into a single node so that G becomes cycle-free. A junction tree $_G = (V, E, \mathcal{X})$ is a connected labeled tree whose node set is $= \{C_1, \cdots, C_n\}$ and edge set is E. Junction trees are labeled trees with label vocabulary X corresponding to dictionary associated with induced subgraphs.

2.2 Graph and Tree Encoder

The Graph G is encoded by a Neural Message Passing framework to give node embeddings h_v. The final graph representation is $h_G = \sum h_v$.

Similarly, the tree is also encoded into z_T.

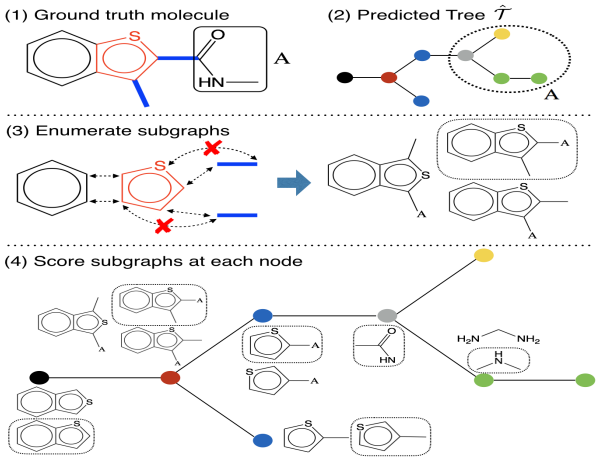

Figure 1: Overview of the method: A molecular graph G is first decomposed into its junction tree T_G, then encoded both the tree and graph into their latent embeddings z_T and z_G. To decode the molecule, we first reconstruct junction tree from z_T , and then assemble nodes in the tree back to the original molecule.

2.3 Tree Decoder

The tree is constructed from z_T in a top-down fashion by generating one node at a time. For every visited node, the decoder first makes a topological prediction, whether this node has children to be generated. When a new child node is created, the corresponding label is predicted. The decoder backtracks when a node has no more children to generate. The output is T_G.

2.4 Graph Decoder

The last step involves decoding the graph whose junction tree is T_G in previous step. This is a structured prediction task over possible graphs whose junction tree is T_G.

3 Evaluation

The method is evaluated on the following criteria on the ZINC dataset:

- Molecule Reconstruction and Validity: To compute validity, 1000 latent vectors from the prior distribution $N(0, I)$ are sampled, and decode each of these vectors 100 times and the percentage of decoded molecules that are chemically valid are reported.
- Bayesian Optimization: This tests how the model can produce novel molecules with desired properties. After learning the VAE, a sparse

Gaussian process (SGP) to predict y(m) given its latent representation is trained. Log Likelihood, top 3 molecules and RMSE is reported.

- Constrained Molecule Optimization: This is specific to drug discovery: to modify given molecules to improve specified properties.A property predictor F (parameterized by a feed-forward network) is trained jointly with JT-VAE to predict y(m) from the latent embedding of m. To optimize a molecule m, start from its latent representation, and apply gradient ascent in the latent space to improve the predicted score.

Table 1: Reconstruction accuracy and prior validity results.

Method	**Recon**	**Validity**
CVAE	44.6%	0.7%
GVAE	53.7%	7.2%
SD-VAE	76.2%	43.5%
GraphVAE	-	13.5%
Atom-by-Atom LSTM	-	89.2%
JT-VAE	**76.7%**	**100.0%**

Table 2: Predictive performance of sparse Gaussian Processes trained on different VAEs.

Method	**LL**	**RMSE**
CVAE	-1.812 ± 0.004	1.504 ± 0.006
GVAE	-1.739 ± 0.004	1.404 ± 0.006
SD-VAE	-1.697 ± 0.015	1.366 ± 0.023
JT-VAE	-1.658 ± 0.023	1.290 ± 0.026

Table 3: Constrained optimization result of JT-VAE: mean and standard deviation of property improvement, molecular similarity and success rate under constraints $sim(m, m') \geq \delta$ with varied δ.

δ	**Improvement**	**Similarity**	**Success**
0.0	1.91 ± 2.04	0.28 ± 0.15	97.5%
0.2	1.68 ± 1.85	0.33 ± 0.13	97.1%
0.4	0.84 ± 1.45	0.51 ± 0.10	83.6%
0.6	0.21 ± 0.71	0.69 ± 0.06	46.4%

4 Conclusion

JT-VAE generates molecules in a sequential manner and ensures chemical validity at every step by ensuring only valid substructures are added. To generate good molecules, a second step of optimization needs to be performed.