Quality prediction of proteins models with spherical convolutions on three-dimensional graphs

Nikita Pavlichenko

Moscow Institute of Physics and Technology

Course: My first scientific paper
Group 793, 2020
Consultants: I. Igashov, S. Grudinin

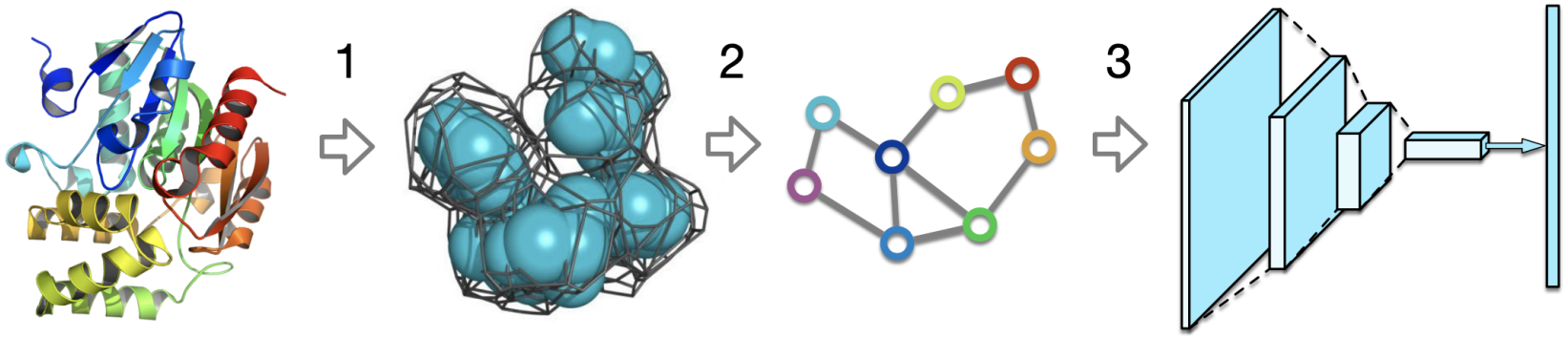
Goal of research

- Develop the new type of convolution operations on three-dimensional graphs that would be able to capture the 3D graph structure;
- Apply these operations to a node regression problem: prediction the quality of proteins model (Protein Quality Assessment).

Problem statement

- Protein can be represented as a graph - a chain of amino acids
- Its properties are determined by its folding
- There are lots of folding models and it is expensive to find the right one for each protein experimentally
- Another way is to predict the quality of these models with machine learning algorithms - the regression problem on graphs. It also called Protein Quality Assessment problem

Problem statement



- Each protein model is described by a list of its amino residues
- Each residue has several features $x(v_i)$: its coordinates (x, y, z) and one categorical feature - the one-hot encoded type of this residue. All rotated or biased models must be isomorphic.
- To construct a graph we build a Voronoi diagram on these amino residues. The cells of this diagram are nodes and if two cells have the same edge, we put an edge between corresponding nodes in the graph.

Problem statement

- For each residue, we have a measure of the quality of its location - CAD score. It is already evaluated experimentally, so we will use it as a target for supervised learning.
- For each model i, we will predict the quality of each residue j. The common choice for loss function is MSE:

$$\sum_{i}^{n} \sum_{j}^{m_i} (\hat{y}_{ij} - \text{CAD-score}_{ij})^2$$

Graph Convolutional Networks

For the last several years the most common approaches to Protein Quality Assessment include deep learning methods and Graph Colvolutional Networks in particular.

Idea

- A very common method is Graph Convolutional Network
- l-th layer can be represented as $H^{(l)} = AH^{(l-1)}W^{(l)}$, where A is the adjacency matrix, $W^{(l)}$ is weights matrix

Props and Cons

- It was successfully applied for PQA before
- It does not capture a local protein structure

Spherical convolutions

- Amino acids are connected one by another so we can define a local coordinate system for every amino acid
- So, let's project it's neighbors onto a unit sphere.
- Consider a function of spherical coordinates. It can be expanded as a series of spherical harmonics.

$$f(\phi, \psi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} f_l^m Y_l^m(\phi, \psi)$$

- Leave only a few coefficients and write it in a matrix view considering $\boldsymbol{W}_l^m$ a weight matrix

$$f(\Omega) \approx f_W(\Omega) = \sum_{l=0}^{L} \sum_{m} \boldsymbol{W}_l^m Y_l^m(\Omega)$$

Spherical Convolutional Network

- Introduce Spherical Convolution operation

$$f_W \circ v_i = \sum_{v_j \in \mathcal{N}(v_i)} f_W(\Omega_i^j) x(v_i),$$

where Ω_i^j denotes spherical coordiantes of the vertex j in local coordinate system of the vertex i.

- Spherical convolution layer:

$$\boldsymbol{X} \longrightarrow \boldsymbol{X}' = \sigma(f_W \circ \boldsymbol{X}) = \sigma\left(\sum_{l,m} Y_l^m(\boldsymbol{A}_\Omega)\boldsymbol{X}\boldsymbol{W}_l^m\right),$$

$\boldsymbol{X} \in \mathbb{R}^{n \times d}$ - input features, $\boldsymbol{X}' \in \mathbb{R}^{n \times d'}$ - output features, σ - non-linearity e.g. ELU.

- Learn $\boldsymbol{W}_l^m$ matrices using Adam optimizer

Computational experiment

To compare the proposed approach we have trained both GCN and SCN networks on CASP data and compared the quality metrics.

Dataset

- We use CASP competition data - more than 100k protein models made by participants and scored experimentally to get the CAD-score. CASP8-11 (80k models) as a training sample and CASP12 (5k models) as a testing sample.
- Spherical harmonics are precalculated: we use an order of 5 and it takes 60GB of HDD

Computational experiment

GCN baseline

- 8 GCN layers with 4 linear layers at the beginning as an encoder and 5 linear layers at the end is the best setup for GCN
- Learn on CPU since the loading from HDD is a bottleneck

Algorithm

- Optimal architecture is 5 spherical convolution layers
- All code is written in PyTorch
- Learn on GPU in 4 parallel subprocesses

Computational experiment

Results

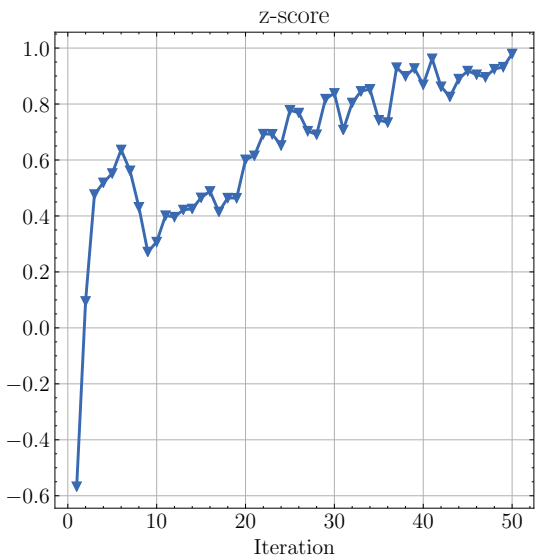
- The common choice for a quality metric is the Pearson correlation between ground truth and predictions
- The best achieved result for GCN is 0.57 comparing with 0.787 with spherical convolutions

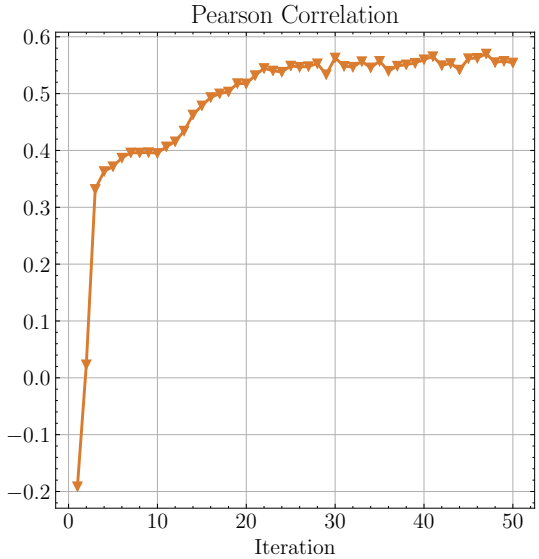
	ρ	r	rank	z-score
ProQ3D	0.801	0.750	11.961	1.670
VoroMQA	0.803	0.766	17.171	1.410
SBROD	0.685	0.762	23.579	1.282
Ornate	0.828	0.781	10.776	1.780
GCN	0.570	0.510	31.667	0.979
SCN	0.787	0.720	18.462	1.411

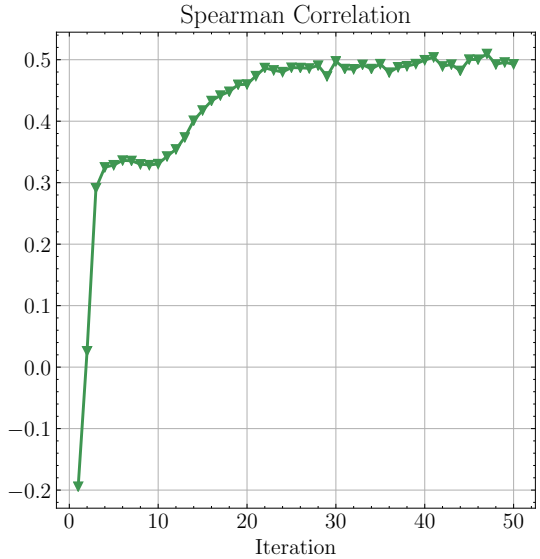
Table: Comparsion of Pearson, Spearman correlation coefficients and z-score and rank of the state-of-the-art algorithms, our GCN baseline and Spherical Convolutional Network on CASP12 dataset.

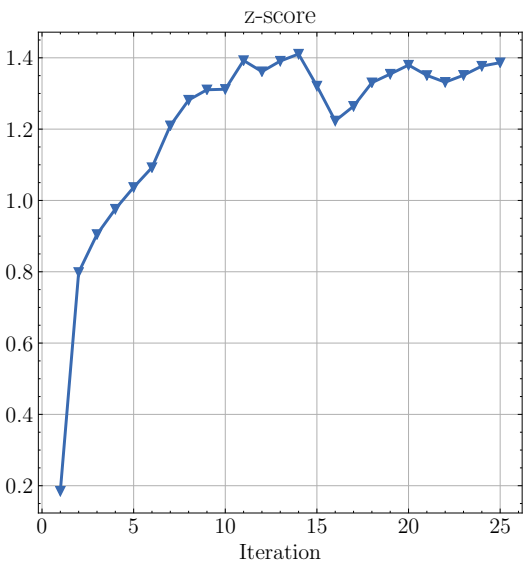
Computational experiment

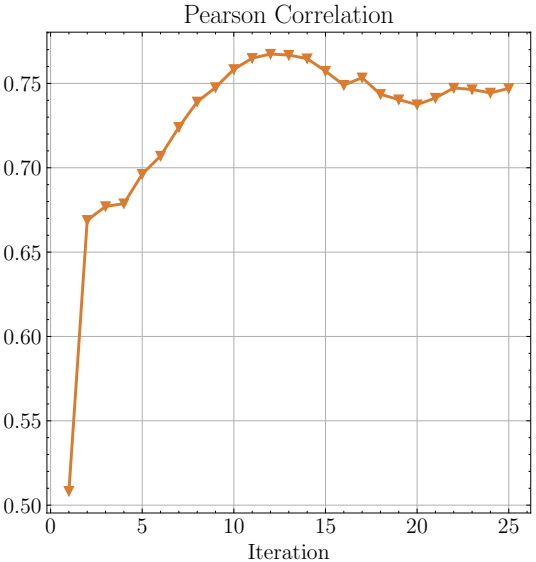
Learning curves of two algorithms show that SCN achieves better results with fewer iterations

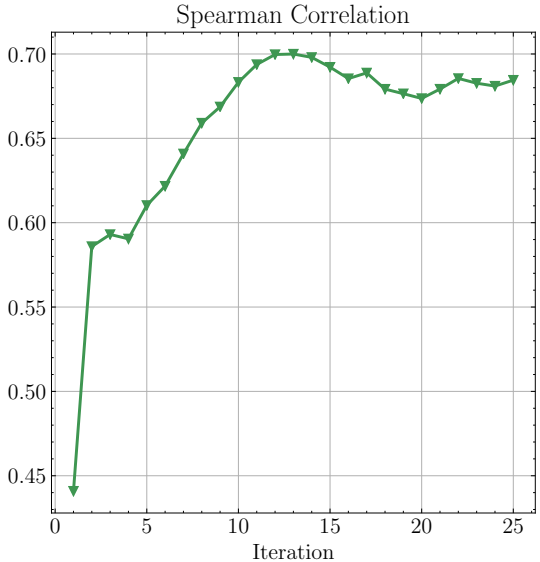












Conclusion

- SCN has shown significantly better performance comparing with GCN. This result can be even compared with the state-of-the-art approaches that include genetic and biology features that represent protein evolution
- It has proved the hypothesis that the most significant property of the protein is its geometric structure

Further research

- Combine SCN approach with features engineered for GraphQA, VoroCNN, and other state-of-the-art algorithms
- Try spherical convolution layers in other graph neural network structures such as variational autoencoders