Graph Neural Networks for genome enabled prediction of complex traits.

I. Hounie[1], J. Elenter[1], G. Etchebarne[1], M. I. Fariello[1], F. Lecumberry[1]

1. Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

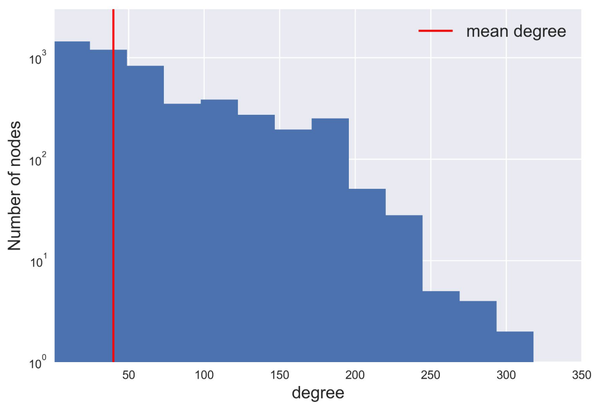
Introduction

Graph representations of genome wide marker information can be derived treating individuals as nodes, giving place to population graphs, where each genotype is supported on a node. We explore and compare different Graph Neural Network architectures for the prediction of complex traits, formulated as a node regression problem.
In addition, we propose a model that also leverages 1D convolutions, which aim to exploit local structures along the genome arising from linkage disequilibrium.
Besides, we evaluate different graph topologies and find that even using random graphs can aid the prediction task.
Models were evaluated on a dataset of Holstein cattle milk yield prediction [1].

Finding graph representations of SNP data

Kinship or pedigree matrices can be used to construct graphs with nodes as individuals, as well as Genomic similarity or relationship matrices.
These matrices can be viewed as the adjacency matrix of an association network i.e., one where nodes with a sufficient level of association between node attributes are connected. We employ pairwise Pearson correlations between genotypes to construct the graph.

However, using correlations directly often results in dense matrices. Thus, inducing sparsity is necessary to reduce computational costs. With this purpose, we explored two methods: thresholding and k-nearest neighbours (*knn*). It has been shown that under certain conditions thresholding is equivalent to Graphical Lasso [2].
Alternatively, *knn* graphs are obtained by keeping only the *k* larger weights for each node, resulting in a graph with constant degree. The topologies of graphs estimated with these two methods have significantly different structures.



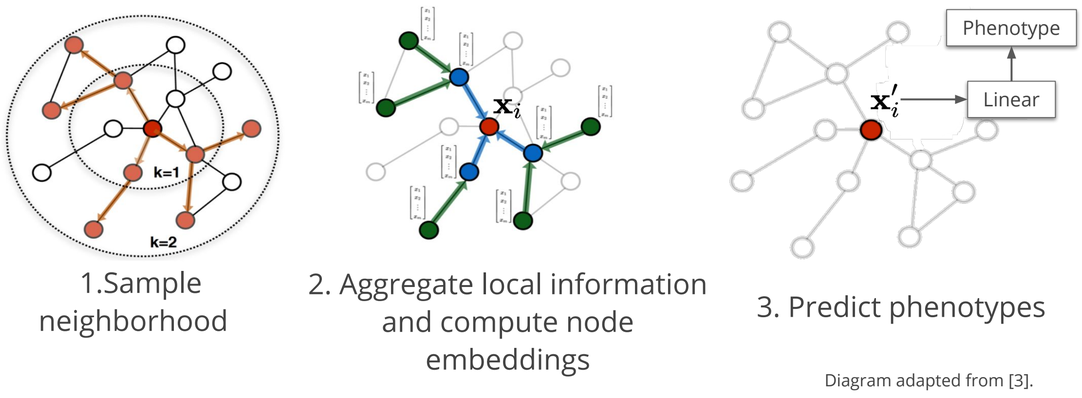
Node degree histogram for a graph with 5024 nodes, constructed using Pearson correlations with a threshold of 0.55. Although its mean degree is 40, graph connectivity is significantly different from a knn graph where the same degree is imposed to all nodes. Notably, thresholding produces a large number of nodes with few neighbours and some with large degree. For instance, while there are five nodes with degree greater than 300, 1378 nodes have degrees under 25. This is related to population structure.

Graph Neural Network architectures

We explored three popular GNN architectures: Graph Sage [3], Graph Attention [4], and Edge Convolutions [5]. These Graph Neural Network architectures can be described as the composition of layers consisting of a neighbourhood aggregation function and a node embedding extractor, followed by a non-linear activation. In the case of Graph Sage, a layer is described by the following equation:

$$\mathbf{x}_i' = \sigma(\mathbf{W}_1\mathbf{x}_i) + \sigma(\mathbf{W}_2 \cdot \mathrm{mean}_{j\in\mathcal{N}(i)}\,\mathbf{x}_j),$$

where σ is a non-linear activation function, $\mathbf{x}_i$ is the genotype of the *i-th* individual and $\mathcal{N}(i)$ its neighbourhood. $\mathbf{W}_1, \mathbf{W}_2$ are linear projection matrices.
A fully connected layer is used to output phenotype predictions. During training, node neighbourhoods were sampled to reduce computational costs and induce regularization.



1.Sample neighborhood

2. Aggregate local information and compute node embeddings

3. Predict phenotypes

Diagram adapted from [3].

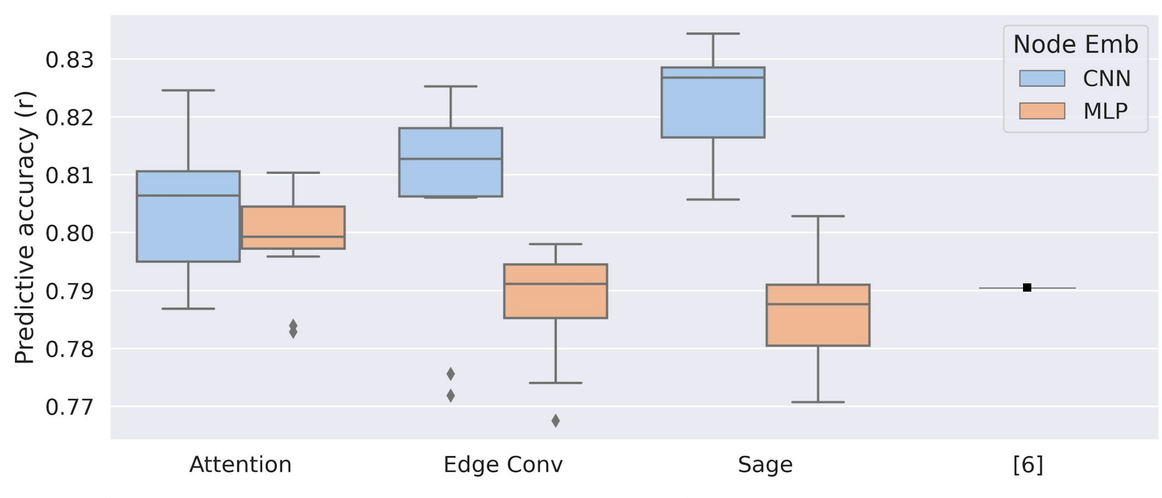
Using CNNs to extract node embeddings

After the neighborhood aggregation step, node and neighbourhood embeddings are computed. A simple way of constructing these embeddings is using a multi-layer perceptron. Due to their regularization properties and inductive biases, Convolutional Neural Networks (CNNs) are a promising alternative. CNNs may be able to exploit local structure due to SNPs in linkage disequilibrium.

$$\mathbf{x}_i^{'} = CNN(\mathbf{x}_i) + CNN(\mathrm{mean}_{j\in\mathcal{N}(i)}\,\mathbf{x}_j)$$

We used a CNN architecture with two main sections. The first one consists of three convolutional blocks (with batch normalization and LeakyRelu activation) and has a residual connection [7]. The second section, consists of two fully connected layers and also has a residual connection.

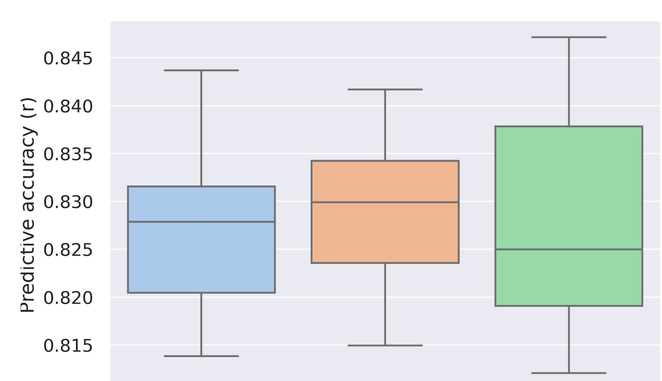
CNNs improved performance for all three Graph architectures.



GNN performance as measured by Pearson correlation (*r*) of single layer graph neural network models. Results correspond to 12 random train/test splits with 80 and 20% of the data, respectively.
All models were trained using dropout, batch normalization, early stopping and a reduce on plateau Learning Rate scheduler. SGD with Nesterov momentum with Sharpness Aware Minimization[8] was used. Results from [6] are included for comparison.

The effectiveness of random graphs

In order to assess the effect of graph topology on predictions, we generated two types of random graphs. The first random graph was constructed by sampling values from a Gaussian distribution with the same mean and variance as the original graph. The second one corresponds to an Erdös–Rényi model (i.e: edges are modelled by independent Bernoulli random variables), having the same mean degree. Then, we trained Sage based models on these graphs and compared their predictive accuracies with our best Sage model.
Randomly connected graphs did not degrade model predictive accuracy significantly.



Sage+CNN model trained using different graph adjacency matrices. All graphs have a mean degree of 40. Non-Random corresponds to the thresholded (0.55) Pearson correlation graph.

Conclusions and perspectives

Formulating complex trait prediction as a node regression problem on a population graph enables the use of graphical models, yet unexplored in genomic prediction literature.

Single layer models from different Graph Neural Network architectures achieved state of the art results in the milk yield prediction dataset. Including Convolutional Neural Networks in GNNs increased the predictive accuracy of all models.

In the light of random graph results, to what extent the graph exploits population structure, and how graph topology affects predictions remains unclear. How these results generalize to other datasets and traits demands further research.

Acknowledgements

This work was partially funded by project ANII FSDA 1-2018-1-154364.

The experiments presented in this work were carried out using ClusterUY (site: https://cluster.uy).

References

[1] Szyda, J., Liu, Z., Reinhardt, F., & Reents, R. (2005). Estimation of quantitative trait loci parameters for milk production traits in German Holstein dairy cattle population. *Journal of dairy science*, *88*(1), 356-367.

[2] Sojoudi, S. (2016). Equivalence of graphical lasso and thresholding for sparse graphs. *The Journal of Machine Learning Research*, *17*(1), 3943-3963.

[3] W.L. Hamilton, R. Ying, and J. Leskovec Inductive Representation Learning on Large Graphs.*arXiv:1706.02216 [cs.SI]*, 2017.

[4] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[5] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, *38*(5), 1-12.

[6] Yin, L., Zhang, H., Zhou, X., Yuan, X., Zhao, S., Li, X., & Liu, X. (2020). KAML: improving genomic prediction accuracy of complex traits using machine learning determined parameters. Genome biology, 21(1), 1-22.

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[8] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," arXiv preprint arXiv:2010.01412, 2020.