Active Learning for Mesh Segmentation: Comparing uncertainty quantification methods for learning on unstructured data

Sofia Lima
Computational Biology Department
Carnegie Mellon University
slima2@andrew.cmu.edu

Parker Simpson
Computational Biology Department
Carnegie Mellon University
psimpson@andrew.cmu.edu

Aruneshwar Venkat
Computational Biology Department
Carnegie Mellon University
aruneshv@andrew.cmu.edu

Jen Yi Wong
Computational Biology Department
Carnegie Mellon University
jenyiw@andrew.cmu.edu

1 Objective

The purpose of this project is to assess the efficacy of active learning for the task of segmenting body parts in 3D meshes of human anatomies (Fig. 1). Obtaining segmentation labels for mesh data is an expensive and time-consuming task. Decreasing the number of samples needed to train an effective segmentation model could save a lot of resources. The dataset chosen for this project comes from researchers in Perceiving Systems at the Max Planck Institute for Intelligent Systems [1]. The goal of this project is to compare the results of active learning with two query selection methods to offline learning and random query selection. Our two query selection methods will be uncertainty sampling and query by committee. We will try two deep learning models, respectively, as base learners.

Figure 1: Sample of GNN segmentation results on MPI Faust mesh dataset.

2 Background

In this dataset, the human body is segmented into 12 regions. This task is equivalent to node-classification in a graph problem. We can model this problem with convolutional neural networks (CNNs) or with graph neural networks (GNNs; powerful for unstructured data such as social networks and protein-protein interactions). Leveraging knowledge in quantified uncertainty may provide advantages and has been demonstrated in medical image segmentation research [2]. One project describes MC dropout with a sampling based model for mesh data called PointNet++ and how running inference multiple times with active MC dropout allows the algorithm to select uncertain samples at multiple granularity levels such as scene-level and point-level [3, 4]. We aim to follow this

active learning framework such that the most uncertain scene-level instance is selected at each round of the active learning algorithm.

Some work has demonstrated quantifying the model uncertainty in GNNs [5, 6]; however, very little work has been conducted exploring the potential advantages of putting the mesh segmentation task in an active learning setting. We will use an ensemble technique for our first uncertainty quantification (UQ) method [2]. For our second UQ method, we will quantify uncertainty using recently developed techniques with monte carlo (MC) dropout during inference [3].

3 Methods

3.1 Active Learning

We aim to analyze potential advantages of active learning on this mesh segmentation task. Similar to protocols in homework exercises, we will follow the prototypical active learning algorithm as follows:

Algorithm 1 Active Learning Algorithm

Obtain initial labeled training data, where $D_L = \{(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)\}$, where each $\mathbf{x}_i \in \mathcal{X}$ is an input instance and $y_i \in \mathfrak{C}$ its corresponding label
repeat
  Learn model: $h = Learn(D_L); h \in \mathcal{H}$
  Select the next point to label, $\mathbf{x}_i \in \mathcal{X}$, according to the data access model
  Pay "oracle" for $\mathbf{x}_i$'s label, $y_i \in \mathfrak{C}$
  Update training data: $D_L = D_L \cup \{(\mathbf{x}_i, y_i)\}$
until stopping condition satisfied
Output final model: $h = Learn(D_L)$

We will apply this active learning framework to our mesh segmentation problem and compare the segmentation results based on mean intersection over union (meanIoU) across different models and query selection methods.

3.2 Uncertainty Quantification Methods

3.2.1 MC Dropout

For the MC dropout, nodes are randomly dropped with a small probability and are dropped for both the training run and the run on unlabeled data. For GNNs, methods also exist to drop the edges of the graph[7, 8]. However, for the scope of this project and for better comparison with CNNs, we will only explore MC dropout of the nodes. We will use multiple runs of the GNN with MC dropout to obtain a probabilistic prediction and each subsequent instance after the training dataset will be selected by choosing the point with the highest uncertainty, quantified using entropy.

3.2.2 Query by Committee

Deep ensemble methods have been shown to improve node classification accuracy on GNNs[9]. Compared to the MC dropout method, it is expected to be more robust but also more computationally expensive [2]. We will use ensemble methods with bagging and each subsequent training instance after the initial training dataset will be selected by quantifying committee disagreement via soft voting.

3.3 Models

3.3.1 GraphSAGE

GraphSAGE is a frequently used GNN architecture capable of inductive node embedding[10] Several advantages of using this model includes its ability to generalize to unseen nodes and its applicability to both feature-rich nodes and nodes with few or no features.

3.3.2 MeshCNN

MeshCNN [11] is a CNN model designed specifically for triangular meshes. Traditional convolution is designed to operate on input with a consistent grid structure, making it incompatible with mesh data that has irregular and non-uniform structures. MeshCNN redefines the notions of convolution and pooling in the context of triangular mesh data to create a model that was able to achieve state of the art performance (circa 2019) in both classification and segmentation tasks.

4 Benchmarking metrics

1. Adequacy. How computationally efficient is this approach? Do we observe either of the two benefits of active learning: higher accuracy or lower sample complexity?
2. Effectiveness. What is the ability to reflect different levels of uncertainty in predictions?
3. Generalizability. How well can this method generalize to other neural network architectures, algorithms, and learning tasks?

5 Timeline

Table 1: Timeline

Deadline	Milestone
14 March	Proposal due
End March	Create figures for GraphSAGE
Mid April	Create figures for meshCNN
End April	Prepare Project Report and Presentation
25 April	Presentation
5 May	Final Report

References

[1] Federica Bogo et al. "FAUST: Dataset and evaluation for 3D mesh registration". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE, June 2014.

[2] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

[3] Erik Harutyunyan. *Active Learning for 3D mesh semantic segmentation*. Last accessed 9 March 2023. 2022. URL: `https://yerevan2022.pydata.org/cfp/talk/DPVYSA/`.

[4] Charles Ruizhongtai Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space". In: *Advances in neural information processing systems* 30 (2017).

[5] Pál András Papp et al. "DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks". In: *35th Conference on Neural Information Processing Systems (NeurIPS 2021)* (2021).

[6] Sai Munikoti et al. "A general framework for quantifying aleatoric and epistemic uncertainty in graph neural networks". In: *Neurocomputing* 521 (2023), pp. 1–10.

[7] Indro Spinelli et al. "FairDrop: Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning". In: (2021).

[8] Jie Wang et al. "GUIDE: Training Deep Graph Neural Networks via Guided Dropout Over Edges". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–13.

[9] Steven J. Krieg et al. "Deep ENSEMBLES FOR GRAPHS WITH HIGHER-ORDER DEPENDENCIES". In: *International Conference on Learning Representations 2023* (2023).

[10] William L Hamilton, Rex Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In: *31st Conference on Neural Information Processing Systems (NIPS 2017)* (2017).

[11] Rana Hanocka et al. "Meshcnn: a network with an edge". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12.