Project proposal

Graph-Based Active Learning with Centrality Measures and GNNs on Tabular Data

Saleem Kheralden khsaleem@campus.technion.ac.il
Majd Bishara majedbishara@campus.technion.ac.il
Lana Haj lanahaj@campus.technion.ac.il

Motivation:

Active learning optimizes data labeling by selecting key points, especially when labeling is costly, or data is abundant. Traditional methods focus on uncertainty or diversity, ignoring data structure. By converting tabular data into a graph and using centrality measures like PageRank, we can identify the most central nodes, which likely represent critical areas of the data. Additionally, GNNs will enhance point representation and leverage graph structure for improved classification decisions.

Method:
Given main model such as Logistic regression:

1. **Graph Construction:**
   Given a tabular dataset, each row will be represented as a node in the graph, edges will be created according to a similarity of distance metric, by creating an edge between 2 points if the metric is greater than a specified threshold.
2. **Centrality Measures:**
   Specified centrality measures will be calculated and will be used to choose the points that will be classified and added to the labeled set.
3. **Active Learning Point Selection & GNN:**
   Points with high centrality are selected as candidates for labeling. The intuition is that these points are likely to represent pivotal areas of the data distribution, making them informative for model training.
   GNN will used on the graph to improve node embeddings and will predict labels for the points with the highest centrality.
   During implementation we will decide how and on which datapoints (such as nodes that were central in the previous iteration) the GNN will be trained on.
   The main classification model will also perform classification not necessarily on the GNN embeddings (will be tested)
   Finally the label, will be determined by both models according to some rule (like) as time based weights for both models)

Topics that will be explored in this project:

1. Best method to create edges (cosine, Euclidian distance…)
2. GNNs training scheme as discussed before.
3. Input data structure (Does the method work better on certain datasets?)
4. Does this method provide better performance over the original AL framework?