Synthetic Dataset Generation

We generated different sets of synthetic datasets, containing 2 classes. For all our experiments, we assumed the synthetic dataset to be a dataset with one view and one relation, that is, every dataset generated synthetically contained one graph and one hypergraph.

The degree distribution of the hyperedges of all the hypergraphs were chosen based on the analysis we had done on the real-world datasets. We used a modified version of the power law distribution where 75% of the hyperedges contained less than 3% of the total number of nodes, 20% of the hyperedges contained between 3% and 50% of the nodes, and the remaining 5% of the hyperedges connected more than half the nodes in the dataset.

We generated graphs and hypergraphs with 1000 nodes and 2 classes, and used a graph homophily value of 0.6 and a hypergraph homophily of 0.4. What this meant was that 60% of the edges of the graph connected nodes taking the same label. In the case of hypergraphs, it meant that 40% of the hyperedges deviated from the expected class distribution. Based on what we noticed from the real-world datasets, we fixed this deviation rate to lie between 1.2 and 3, that is, one of the classes occurs anywhere between 1.2 times and 3 times more than what is expected of it.

Let the desired skew factor be $1{:}skew$, that is class B contains $skew$ times the number of nodes as A does. Let n be the number of nodes in the dataset and let h_g be the homophily factor chosen for the graph. Then the graph is generated as follows: continue adding edges till the graph is connected. For every edge, we need to choose the vertices it needs to connect; with probability $1 - h_g$, pick two vertices at random, one from each class and with probability h_g, pick two vertices from the same class and connect them. The way to pick two vertices from the same class is to pick two vertices at random (without replacement) from class A with a probability $1/(skew + 1)$ and two vertices from class B with a probability $skew/(skew + 1)$.

For generating a hypergraph, there are two steps: first choose the number of nodes k that would go into a hyperedge (for which we sample from the modified power law distribution), and then choose k different nodes based on the homophily and skew factors. Due to the high number of nodes that could potentially be connected by a single hyperedge, the hypergraph gets completely connected pretty soon, and hence we do not use the connectivity of the hypergraph as a criterion for stopping the generation process. Instead, we choose the number of hyperedges we want in the graph, possibly as a function of the number of nodes in the hypergraph. The algorithm that was used to construct hypergraphs is described in algorithm 1.

Algorithm 1 Synthetic Hypergraph Generation

```
function GENERATEHYPERGRAPH($numNodes$, $skew$, $homophily$, $numHyperedges$)
    ▷ Nodes 1...ceil(numNodes/(skew+1)) belong to class A and the remaining to class B
    for $i = 1 : numHyperedges$ do
        r ← $rand(0, 1)$
        if $r < 0.75$ then
            k ← Random integer between 2 and 0.03 * numNodes
        else if $r < 0.95$ then
            k ← Random integer between 0.03 * numNodes and 0.5 * numNodes
        else
            k ← Random integer between 0.5 * numNodes and numNodes
        r ← $rand(0, 1)$
        if $r < homophily$ then
            s ← rand(1.2, 3)
            if $i$ %2 $! = 0$ then                              ▷ Increase fraction of class A
                ANodes ← Pick $(k * s)/(s + skew)$ class A nodes at random
                BNodes ← Pick $(k * skew)/(s + skew)$ class B nodes at random
            else
                ANodes ← Pick $k/(1 + s * skew)$ class A nodes at random
                BNodes ← Pick $(k * s * skew)/(1 + s * skew)$ class B nodes at random
            Connect ANodes and BNodes with a hyperedge
        else Pick $k$ nodes at random and connect using a hyperedge
```