Community Detection

Leiden Algorithm

In many complex networks, nodes cluster and form relatively dense groups called communities. Communities are a property of many networks in which a particular network may have multiple communities such that nodes inside a community are densely connected. Nodes in multiple communities can also overlap.

For our use case, we have to identify which items are frequently bought together and in what quantity they are ordered so that we can place those items as required so that our robots can carry out the orders faster and more efficiently.

Leiden Algorithm

This algorithm solves the drawback of the Louvian algorithm which can sometimes give rise to weakly connected communities and also moving a node which was previously acting as a bridge between two communities can disconnect it from the old community.

The Leiden algorithm is much faster than the Louvian algorithm and solves the above problems.

The Leiden algorithm consists of three phases:

* Local moving of nodes

The algorithm starts with a weighted network of N nodes. In the first phase, the algorithm assigns a different community to each node of the network. Then for each node, it considers the neighbors and evaluates the gain of modularity by removing the particular node from the current community and placing it in the neighbor’s community. The node will be placed in the neighbor’s community if the gain is positive and maximized. The node will remain in the same community if there is no positive gain. This process is applied repeatedly and to make it quicker only those nodes are visited whose communities have been changed earlier. The first phase of the Leiden algorithm stops when a local maxima of modularity is obtained.

* Refinement of the partitions

In the refinement phase, the algorithm tries to identify refined partitions from the partitions proposed by the first phase.Communities proposed by the first phase may further split into multiple partitions in the second phase. The refinement phase does not follow a greedy approach and may merge a node with a randomly chosen community which increases the quality function. This randomness allows discovering the partition space more broadly. Initially, P-refined is set to a singleton partition, in which each node is in its own community. The algorithm then locally merges nodes in P-refined: nodes that are on their own in a community in P-refined can be merged with a different community. Importantly, mergers are performed only within each community of the partition P. In addition, a node is merged with a community in P-refined only if both are sufficiently well connected to their community in P.

* Aggregation of the network based on refined partitions

In this phase, the algorithm builds a new network considering communities found after refinement as nodes. Once this is done, the algorithm will reapply the first and second phases to the resulting network. These steps are repeated until there are no changes in the network and maximum modularity is obtained.

