2.7 Label Propagation

Label Propagation is a graph-based algorithm used mainly for community detection in networks or graphs. Let’s go step by step and explain it clearly.

1. What is Label Propagation?

Label Propagation is a fast, iterative algorithm that assigns labels to nodes in a network to detect communities (groups of nodes that are densely connected internally).

- Each node has a label.
- Initially, each node can have a unique label.
- Nodes iteratively adopt the label that most of their neighbors have.
- Over time, labels propagate through the network, forming communities.

2. Key Idea

- Nodes with many connections to the same label will adopt that label.
- Eventually, nodes in the same community share the same label.
- It is unsupervised, so no prior information about communities is needed.

3. How It Works (Step by Step)

1. **Initialization**: Assign a unique label to each node.
   Example: Node A → label A, Node B → label B, etc.
2. **Iteration**:
   - For each node:
     - Look at the labels of its neighbors.
     - Update its label to the most frequent label among neighbors.
     - If there’s a tie, pick randomly among the most frequent labels.
3. **Convergence**: Repeat until labels no longer change or until a maximum number of iterations is reached.
4. **Output**: Nodes with the same label form a community.

4. Example

Consider a small graph:

A—B—C

| |

D   E

- Initial labels: A→A, B→B, C→C, D→D, E→E
- Iteration 1:
  - B sees A and C → adopts the label that is most frequent among neighbors → could become A or C
  - D sees A → adopts label A
  - E sees C → adopts label C
- Iteration 2+: labels propagate until stable communities:
  - Community 1: {A, B, D}
  - Community 2: {C, E}

5. Advantages

- Very fast and scalable for large networks.
- Simple and easy to implement.
- No prior knowledge of the number of communities is needed.

6. Disadvantages

- Results can be non-deterministic due to random tie-breaking.
- Sensitive to network structure; sometimes unstable communities.
- Not ideal for graphs with very weak community structures.

7. Applications

- Social network analysis (finding friend groups)
- Biological networks (protein interaction communities)
- Recommender systems (grouping similar users/items)
- Communication networks (detecting clusters)