COIS 4400H

Assignment 3 - Question 2

Winter 2025

Dikshith Reddy Macherla

Student Id : 0789055

Bachelor Of Computer Science, Trent University

COIS 4400H- Data Mining

Professor : Sabine McConnell

21st March, 2025

1. Affinity Propagation

One clustering method I found particularly fascinating is Affinity Propagation, introduced by Frey and Dueck (2007). What stood out to me right away is that, unlike algorithms like K-means, it doesn’t require me to specify the number of clusters beforehand. That’s a huge advantage, especially when I’m exploring a new dataset and have no idea what structure it might contain. Affinity Propagation works by identifying "exemplars," or representative data points that serve as the centers of clusters. What makes this method unique is that it operates on a similarity matrix rather than a distance metric like Euclidean distance. This allows me to use custom similarity measures, such as cosine similarity or even domain-specific measures.

The algorithm uses a clever message-passing framework. Each data point sends and receives two kinds of messages: responsibility, which indicates how suitable a point is to serve as another’s exemplar, and availability, which reflects how appropriate it would be for a data point to choose that exemplar. These messages are updated iteratively until convergence. The result is a set of exemplars and assignments of each point to its closest exemplar.

I’ve personally applied Affinity Propagation to a text clustering project using cosine similarity between TF-IDF vectors. I didn’t have to guess the number of clusters, and it performed surprisingly well in grouping semantically related documents. It also handled varying cluster sizes and shapes better than other methods I tried.

2. Markov Clustering (MCL)

Another clustering method I’ve explored and found particularly powerful is **Markov Clustering (MCL)**. What sets it apart is that it’s designed specifically for graph data and does not rely on Euclidean distance. Instead, it detects clusters based on how information flows through a network, which makes it ideal for clustering in domains like bioinformatics, social networks, and recommendation systems. I learned about MCL from the dissertation by van Dongen (2000), and it quickly became one of my go-to methods when working with graph-based structures.

The core concept behind MCL is that in a graph, random walks tend to get “trapped” in densely connected regions, which naturally correspond to clusters. MCL models this behavior using two main operations on the adjacency matrix of the graph: **expansion** and **inflation**. Expansion simulates spreading out via random walks (matrix multiplication), while inflation amplifies stronger connections and suppresses weaker ones (by raising elements to a power and normalizing). These steps are repeated iteratively, and over time, the graph separates into distinct subgraphs—each representing a cluster.

I used MCL on a protein-protein interaction network, where conventional methods like K-means completely failed due to the non-Euclidean nature of the data. MCL, however, identified tight communities of proteins that had meaningful biological interpretations. One of the biggest strengths of MCL is that it doesn’t require me to specify the number of clusters in advance, and the results are highly interpretable.

References

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. Science, 315(5814), 972–976. https://doi.org/10.1126/science.1136800

van Dongen, S. (2000). Graph clustering by flow simulation (Doctoral dissertation, University of Utrecht). https://micans.org/mcl/