

Université D'Alger 1 Benyoucef Benkhedda



Département Informatique Spécialité : Analyse et Sciences de données

Module : Analyse des réseaux sociaux

RAPPORT DE Mini projet 01: Rapport d'Analyse du Réseau du Club de Karaté de Zachary

Réalisé par :

- Ait Mohand Aya Groupe 01
- Achebane Soumia Groupe 01
- khamis Sirine meriame Groupe 02
- Haddadj Malak Samira Groupe 01

2025/2026

1 Introduction

Ce projet consiste à modéliser et analyser le réseau social du karaté club de Zachry à l'aide du langage Python et de la bibliothèque NetworkX. L'objectif est de représenter le réseau sous forme de graphe non orienté, défini à partir de sa matrice d'adjacence construite manuellement, puis d'appliquer différentes mesures structurelles permettant de caractériser la topologie du réseau.

L'analyse porte notamment sur le calcul de l'ordre et de la taille du graphe, la distribution des degrés, les coefficients de clustering, la détection de motifs fréquents, de cliques et de k-cores, ainsi que l'identification des nœuds centraux selon plusieurs indicateurs de centralité.

Une interface graphique interactive a également été développée pour permettre la visualisation du graphe, la manipulation dynamique des nœuds et arêtes, ainsi que l'export des résultats au format CSV ou image. Cette approche combine ainsi la modélisation algorithmique, l'analyse quantitative et l'interactivité pour une exploration complète du réseau.

2 Modélisation du Graphe

Nous avons modélisé le réseau du club de karaté comme un graphe non orienté et non pondéré, car les relations d'amitié sont symétriques entre les membres. Nous avons utilisé une matrice d'adjacence 34×34 pour représenter ces connexions, avec des 1 indiquant une amitié et des 0 son absence. Ce choix nous a permis de visualiser facilement l'ensemble du réseau et de calculer efficacement ses propriétés structurelles.

3 Analyse Structurelle : Comparaison Manuel vs NetworkX

3.1 Ordre et Taille du Graphe

Les deux méthodes donnent des résultats identiques (34 nœuds et 83 arêtes), confirmant la correcte implémentation manuelle. L'ordre mesure la taille du réseau tandis que la taille indique l'interconnectivité globale.

3.2 Distribution des Degrés

```
1 # Calcul manuel des degrés
2 degrees_manual = [sum(row) for row in adj_matrix ]
3
4 # Calcul NetworkX
5 degrees_nx = dict( G.degree () )
```

L'analyse de la distribution des degrés révèle une structure hétérogène du réseau, où quelques membres très populaires (nœuds 34, 1 et 33 avec respectivement 18, 16 et 11 connexions) contrastent avec la majorité des membres ayant peu de relations. Cette distribution inégale, caractéristique des réseaux sociaux réels, montre que l'essentiel des connexions est concentré autour de quelques individus centraux.

3.3 Coefficient de Clustering

Le coefficient de clustering mesure la tendance des nœuds à former des triangles relationnels.

```
# Calcul manuel
C(i) = (2 * triangles(i)) / (deg(i) * (deg(i) - 1)) #local
CC(i) = somme(C(i)) / |V| # global
# Calcul NetworkX
clustering_nx = nx.clustering(G)
clustering_global = nx.average_clustering(G)
```

3.3.1 Coefficient global

Le coefficient moyen du réseau est de 0,5613. Cette valeur élevée indique que le réseau est très soudé avec une forte proportion de triangles d'amitié.

3.3.2 Coefficients locaux

Chaque nœud possède son propre coefficient. Par exemple, les nœuds 8, 13 et 15 ont un coefficient de 1,0 (tous leurs amis sont amis entre eux), tandis que les nœuds 1, 33 et 34 ont des coefficients plus faibles. Cela révèle que certains membres appartiennent à des groupes très soudés, tandis que d'autres ont des amis moins connectés entre eux.

Le triangle est le motif fréquent principal observé, confirmant la forte cohésion sociale et la tendance à former des groupes fermés.

3.4 Cliques et Sous-structures

```
# Calcul manuel
maximal_cliques_manual
# Calcul NetworkX
cliques_nx = list(nx.find_cliques(G))
max_clique = max(cliques_nx, key=len)
```

L'analyse des cliques identifie les sous-groupes sociaux où tous les membres sont connectés entre eux, mesurant la cohésion locale et révélant la structure fine du réseau pour la détection de communautés. La plus grande clique compte 5 nœuds et on dénombre 10 cliques maximales.

3.5 k-Cores

```
# Calcul manuel
K_cores_manual(A,K)
# Calcul NetworkX
k_cores_nx = nx.core_number(G)
max_k_core = max(k_cores_nx.values()) # k-core maximal = 4
```

Les k-cores identifient le noyau central le plus interconnecté (12 nœuds), montrant la résilience du réseau et son organisation hiérarchique en couches structurelles. Le k-core maximal est de 4, composé des nœuds 1, 2, 3, 4, 8, 9, 14, 20, 31, 32, 33, 34.

4 Identification des Nœuds Centraux

4.1 Centralité par Degré

```
# Calcul manuel
degree centrality_manual(A)
# Calcul NetworkX
nx.degree centrality(G)
```

Cette mesure identifie les membres les plus populaires et connectés du réseau basé sur le nombre de connexions directes.

4.2 Centralité de Vecteur Propre

```
# Calcul manuel
eigenvector centrality manual(A)
# Calcul NetworkX
eigenvector_nx = nx.eigenvector centrality (G, max_iter =1000)
```

Cette mesure évalue l'influence en considérant l'importance des connexions, où être connecté à des nœuds importants renforce sa propre influence.

4.3 Centralité de Proximité

```
# Calcul manuel
closeness centrality manual(A)
# Calcul NetworkX
nx.closeness centrality(G)
```

Identifie les nœuds à diffusion rapide de l'information, mesurant l'efficacité à atteindre rapidement l'ensemble du réseau.

4.4 Centralité d'Intermédierité

```
# Calcul manuel
betweenness centrality manual(A):
# Calcul NetworkX
nx.betweenness centrality(G)
```

Identifie les ponts contrôlant le flux d'information entre différentes parties du réseau, mesurant le contrôle informationnel.

5 Interface Interactive et Visualisation

Une interface graphique interactive a été développée en utilisant la bibliothèque Tkinter, le toolkit graphique standard de Python, pour compléter l'analyse structurelle. Cette interface permet :

1. Visualisation dynamique du graphe avec disposition fixe (spring_layout, seed=42) pour une cohérence visuelle, intégrée via FigureCanvasTkAgg de Matplotlib

2. Manipulation en temps réel via quatre modes opérationnels gérés par les widgets

Tkinter :

- Sélection : Consultation des propriétés des nœuds (degré, voisinage, métriques)
- Ajout d'arêtes : Création de nouvelles connexions entre nœuds sélectionnés
- Ajout de nœuds : Insertion de nouveaux membres dans le réseau
- Suppression : Élimination d'éléments (nœuds ou arêtes)

Fonctions d'analyse intégrées :

1. Calcul automatique des métriques (centralité, clustering, intermédierité)
2. Génération de rapports statistiques complets via filedialog
3. Export des résultats aux formats CSV et PNG
4. Réinitialisation du graphe à l'état original

L'interface valide expérimentalement les résultats de l'analyse structurelle : les nœuds 34,1 et 33 apparaissent effectivement comme les plus centraux, confirmant leur rôle pivot dans la cohésion du réseau. La possibilité de réinitialiser le graphe à son état original garantit la reproductibilité des analyses tout en permettant des explorations hypothétiques de modifications structurelles.

Cette approche interactive enrichit l'analyse quantitative en offrant une compréhension intuitive de la topologie du réseau et de sa dynamique relationnelle, complétant ainsi les mesures statistiques par une exploration visuelle et manipulable.

6 Export et Visualisation des Résultats

L'objectif de cette section est de capitaliser les résultats obtenus et de les présenter sous forme exploitable pour l'analyse et la documentation.

Export des Données Structurées

Un fichier CSV complet a été généré (`analyse_complete_karate.csv`) regroupant l'ensemble des métriques calculées pour chaque nœud du réseau. Ce fichier contient :

- Les propriétés structurelles : degré, coefficient de clustering, nombre de triangles
- Les valeurs k-core indiquant l'appartenance aux couches centrales du réseau
- Les quatre mesures de centralité (degré, proximité, intermédiarité, vecteur propre)

La matrice d'adjacence a également été exportée (`matrice_adjacence_karate.csv`) pour permettre des analyses ultérieures ou des réutilisations du jeu de données.

Visualisation Graphique du Réseau

Une visualisation complète du réseau a été produite, intégrant plusieurs dimensions d'analyse :

1. Taille des nœuds : Proportionnelle au degré de connexion
2. Couleur des nœuds : Représentant la valeur k-core selon une échelle colorimétrique
3. Labels : Affichant les numéros d'identification des membres
4. Légende : Indiquant l'échelle des valeurs k-core

Cette visualisation permet d'identifier immédiatement la structure centrale du réseau et la distribution des rôles parmi les membres.

Analyse Comparative des Centralités

Quatre graphiques en barres présentent une analyse comparative des différentes mesures de centralité :

1. Centralité de degré : Popularité directe des membres
2. Centralité de proximité : Capacité à diffuser rapidement l'information
3. Centralité d'intermédiarité : Rôle de pont entre les communautés
4. Centralité de vecteur propre : Influence basée sur l'importance des connexions

Cette analyse multidimensionnelle révèle les différents types d'influence au sein du réseau.

Distribution des Degrés

Un histogramme amélioré présente la distribution des degrés avec :

1. Affichage des valeurs exactes sur chaque barre
2. Mise en évidence de l'hétérogénéité des connexions

3. Identification des degrés les plus fréquents

Conclusion des Exports

L'ensemble des exports réalisés permet :

1. Une documentation complète des analyses via les fichiers CSV
2. Une visualisation professionnelle pour rapports et présentations
3. La reproductibilité des analyses grâce à la conservation des données brutes
4. Une analyse comparative facilitée par les graphiques multidimensionnels

Ces éléments constituent une base solide pour l'interprétation sociologique du réseau et pour d'éventuelles analyses complémentaires.