TP 3 : Études des mesures de centralités

Rushed Kanawati

27 octobre 2022

1. Télécharger les réseaux de benchmark : `dolphins.gml, karate.gml, foorball.gml, polbooks.gml` de `http://lipn.fr/~kanawati/datasets`
2. Développer une fonction `centrality_vis(graphe,centrality)` qui visualise le graphe `graphe` en modifiant la taille de chaque nœud d'une manière proportionnelle à la centralité du nœud selon la fonction `centrality`. Tester et montrer les résultats avec les mesure de centralités usuelles : `degree, closeness, betweeneess`.
3. Proposer une fonction `Borda_ER` qui mets en œuvre l'algorithme de `Borda` pour la fusion de préférences
4. Proposer une fonction `Condorcet_ER` qui mets en œuvre l'algorithme de `Condorcet local` pour la fusion des préférences.
5. Pour chaque réseaux de benchmark, calculer le tri des nds selon l'ensemble des centralités `degree, closeness, betweenness, pagerank, coreness` et en utilisant les deux fonction de fusion de préférences `borda` et `condorcet`.
6. Pour chaque réseau de benchamark, calculer la mesure de corrélation de rangs `kendall` entre les listes des nœuds triée selon les centralités suivantes : `degree, closeness, betweenness, pagerank, coreness`
7. Développer une fonction `embed(graphe)` qui renvoie une représentation du graphe `graphe` sous forme de vecteur de corrélations deux à deux des liste de nœuds triées selon les centralités suivantes : `degree, closeness, betweenness, pagerank, coreness`
8. Utiliser les fonctions de génération de graphes aléatoires `erdos_renyi`, `watts` et `barabasi` pour générer 100 graphes différents de chaque modèle. Calculer la représentation de chaque graphe génèeré avec la fonction `embed` et appliquer une fonction de clustering sur les représentations obtenues. Commenter le résultat.