NET 4103/7431 Homework
Network science and Graph Learning

Vincent Gauthier
vincent.gauthier@telecom-sudparis.eu

Rubens Torres-Lacaze
05/05/2003
rubens.torres-lacaze@telecom-sudparis.eu

Lola Klein
21/02/2003
lola.klein@telecom-sudparis.eu

Lien du google colab associé :
https://colab.research.google.com/drive/1BrKd9I_Sj7unyQEWlvRtaIzclVLQurWv?usp=sharing
Lien du repo github associé : https://github.com/NoLilypad/DMComplexNet

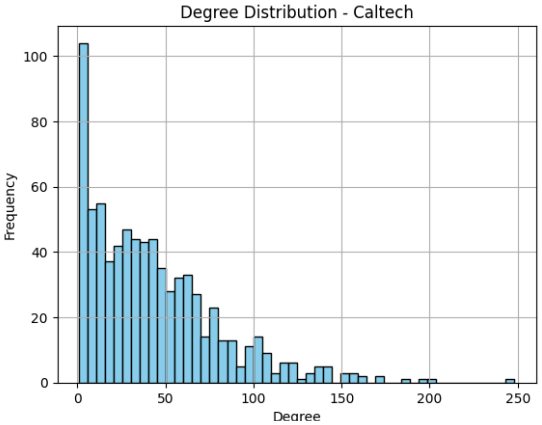
Question 2

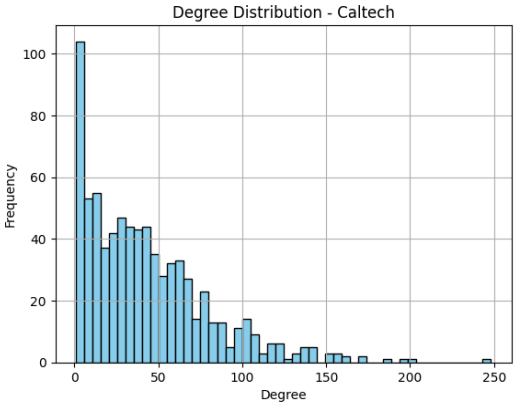
Question 2.a

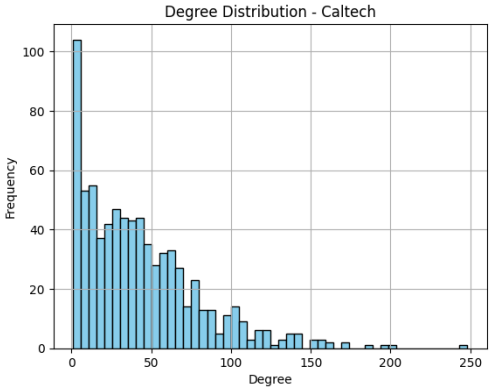
On trace la distribution de degrés comme vu sur le notebook à la section “Question 2.a”. On utilise comme prévu la library NetworkX, en traçant les histogrammes. On vérifie également si on peut déduire des informations supplémentaires.

On observe une distribution similaire dans la forme entre les trois universités, même si Caltech semble avoir une distribution moins généreuse que le MIT et Johns Hopkins.

De manière générale, la forme des histogrammes montre une structure à laquelle on pouvait s’attendre, il y a une forte inégalité entre des étudiants centraux, des noeuds hubs, qui sont minoritaires, et une majorité d’étudiants avec un faible taux de relations.







Questions 2.b

On calcule trois données des graphes :

- Le clustering global
- Le clustering moyen local
- La densité

On en obtient les résultats suivants, arrondis au millième :

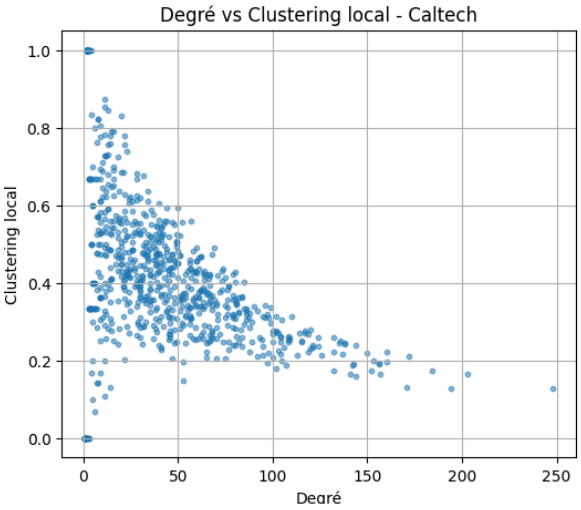
	Caltech36	MIT8	Johns Hopkins55
Clustering global	0.291	0.180	0.193
Clustering moyen local	0.409	0.271	0.268
Densité	0.0564	0.0121	0.0139

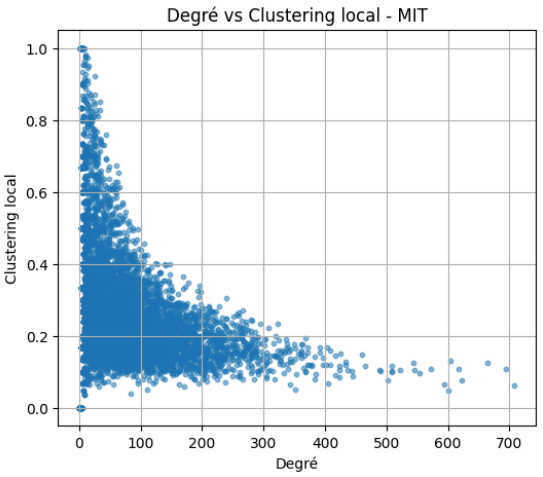
On a utilisé des fonctions déjà intégrées à NetworkX pour calculer la densité et les clustering.

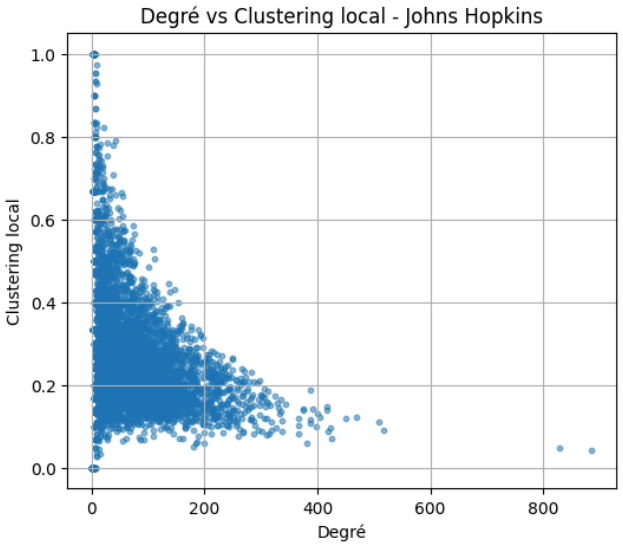
- On déduit des densités très faibles (<0.06) que les trois graphes sont sparses. Le plus petit réseaux, Caltech, a la plus grande densité, ce qui peut sembler logique qu'une plus petite université génère des relations plus "denses"
- Avec les résultats de clustering global (transitivité) et quelques recherches, on peut affirmer que Caltech est bien structuré en communautés (>0.2), de qui rejoint l'observation précédente, et que les deux autres ont des clusterings moyens, mais tout de même non aléatoires
- Le clustering moyen local, qui montre si "mes amis sont amis aussi", montre que des communautés sont bien formées, surtout à Caltech (>0.4) mais dans les deux autres aussi de manière significatives

Question 2.c

On trace les trois graphes, avec en ordonnée le clustering local moyen, et en abscisse sa densité.







On nous demande, à partir des résultats précédents et de ces nouveaux graphes, de discuter des différences et similarités entre les trois réseaux.
On remarque déjà visuellement la différence de taille de population à Caltech, qui est un réseau nettement plus petit, ce qui influe peut-être les autres paramètres, comme on a d'ailleurs pu le dire avant. En effet, ces graphes représentent des situations réelles.
Pour les trois graphes, on observe la décroissance de clustering local selon le degré : plus un noeud est connecté, plus ses "amis" sont isolés (en moyenne). Les noeuds à haut clustering qui ont un degré faible jouent le rôle de pont entre des communautés.
Caltech a des noeuds avec un clustering moyen supérieur de manière générale, quelle que soit le degré (il y a une sorte de borne inférieure autour de 0.2), donc les noeuds sont plus "liés en communauté".

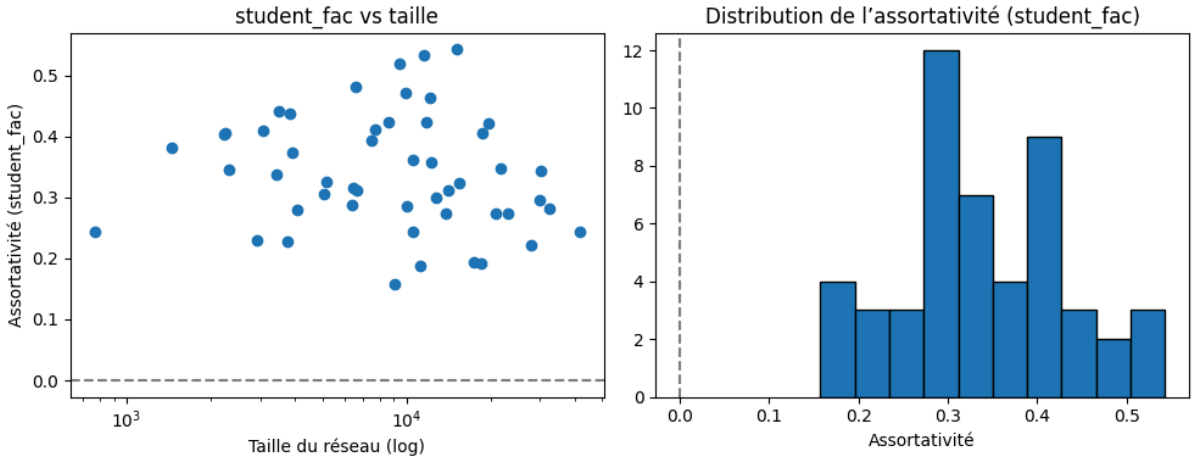
Chez le MIT et Johns Hopkins, la répartition est clairement plus dispersée, ce qui suggère un plus grand nombre de noeuds qui forment des ponts entre groupes. On peut encore une fois l'attribuer à la plus grande taille de leurs campus.
Ces scatters plots confirment donc les premières intuitions que l'on peut avoir.

Question 3

Question 3.a

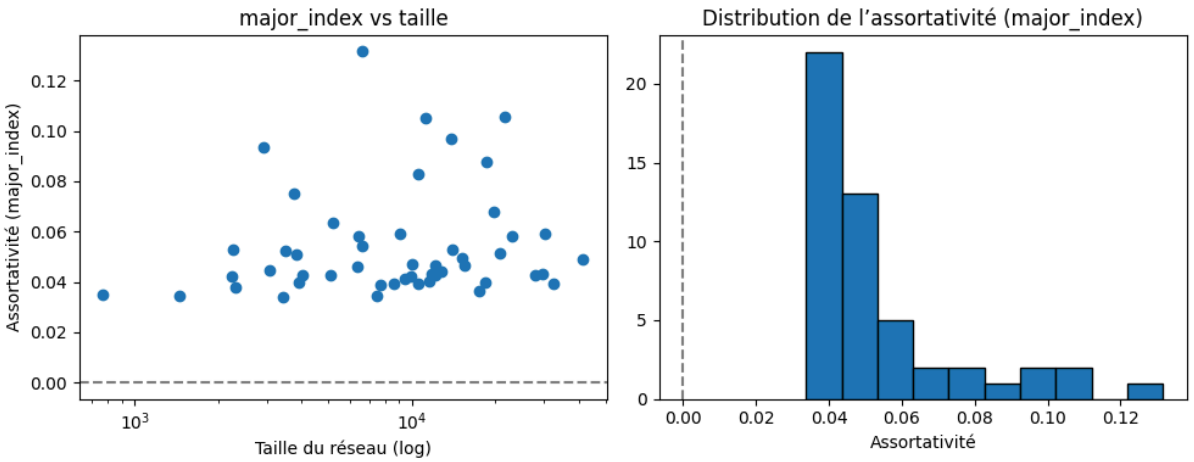
L'assortativité mesure la tendance des utilisateurs à se connecter avec d'autres ayant des attributs similaires. On analyse ici cinq attributs sur le maximum d' universités.
J'ai eu beaucoup de mal à compute les assortativités pour 100 réseaux, j'ai donc fait 5 puis 20 puis 50.
Pour 50 universités, j'ai mis plus de 30 minutes à compute toutes les assortativités.
Voilà les résultats pour les attributs, sauf le gender qui est dans l'énoncé.

Student Fac :



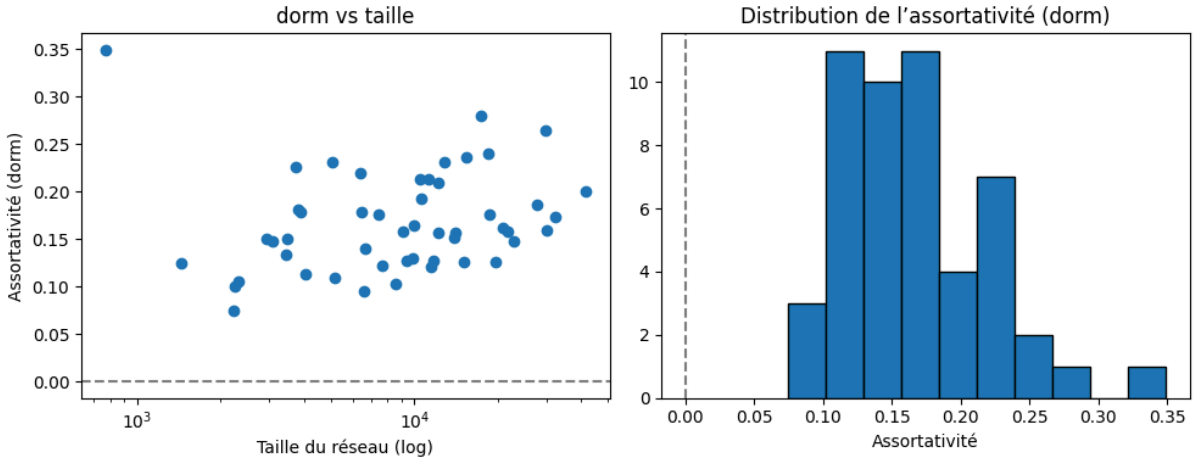
On a ici un attribut avec une très forte assortativité, ce qui semble relativement logique étant donné que le statut étudiant oriente beaucoup les relations qui se créent - en tant qu'étudiant on est plus susceptibles de parler à sa promo et sa filière qu'aux alumnis. La tendance semble être la même pour la plupart de tailles de réseaux même si les plus gros réseaux sont légèrement plus exclusifs dans leur relations, et on remarque nu pic autour de 0.3/0.4, une tendance donc assez importante.

Major index (matière principale) :



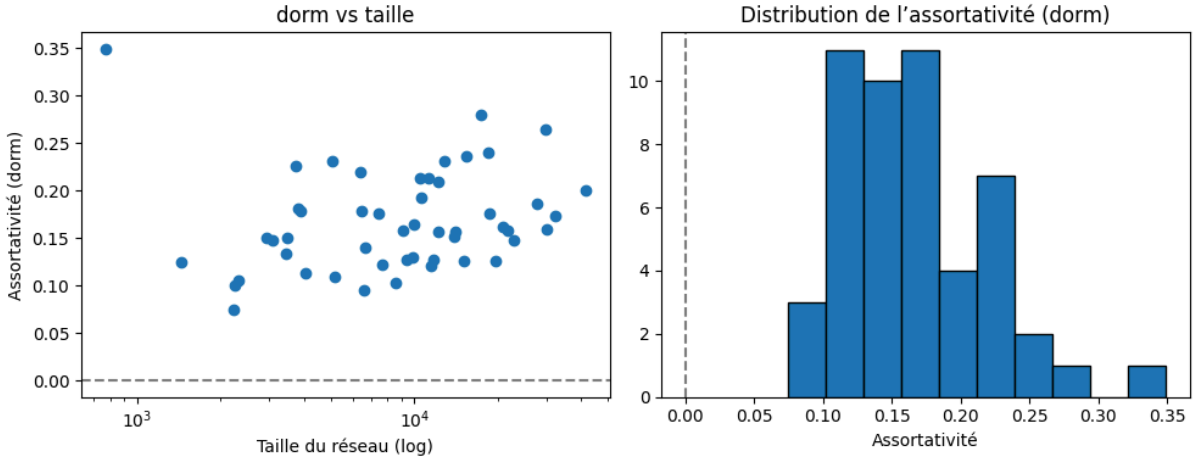
On observe ici ce qui semble être un “minimum d’assortativité”, autour de 0.35. Ce minimum s’explique par la segmentation sociale autour des filières. Plus l’université est grosse, plus cette séparation peut-être importante. Malgré tout, l’histogramme montre que cette assortativité est relativement faible par rapport à l’attribut précédent, et semble se concentrer sur un “minimum”.

Dorm (Dortoire, résidence étudiante) :



Ici, on observe une tendance qui serait que plus un réseaux est gros, plus les relations entre dorms deviennent “exclusives”, ce qui peut s’expliquer par la taille de ces dortoirs qui doit augmenter avec celle du campus, et le fait que plus le campus est grand, plus ils sont séparés. Il semble, selon l’histogramme, y avoir une assortativité “standard” autour de 0.15.

Degree (niveau de relations) :



Ici, on retrouve avec gender une des deux assortativités qui est parfois négative. Cela semble logique, dans la mesure ou certaines personnes “populaires” sont en relations avec ce qu’on pourrait appeler des “isolats” sociaux. Le scatter plot nous montre qu’il y a une légère tendance à ce que la taille augmente les relations entre personnes de même “degré social”.

Question 4

Question 4.b

On code ici des prédicteurs, en reprenant la classe abstraite qu’on nous donne dans le listing 1. La principale difficulté est de retrouver dans le cours et dans nos recherches les bonnes méthodes, avec les bons rappels.

Question 4.c

L’objectif est de tester la capacité de différents algorithmes à prédire des liens manquants dans un graphe social. Pour cela, on supprime aléatoirement une partie des arêtes du graphe, puis nous appliquons nos prédicteurs. On compare finalement les liens prédits avec ceux supprimés.

On suit les consignes. Pour le graphe **Caltech36**, j'obtiens les résultats suivant :
>>> Pour common_neighbors :

Top@50: TP=7, Precision=0.140, Recall=0.004
Top@100: TP=9, Precision=0.090, Recall=0.005
Top@200: TP=13, Precision=0.065, Recall=0.008
Top@300: TP=15, Precision=0.050, Recall=0.009
Top@400: TP=16, Precision=0.040, Recall=0.010
>>> Pour jaccard :

Top@50: TP=1, Precision=0.020, Recall=0.001
Top@100: TP=4, Precision=0.040, Recall=0.002
Top@200: TP=7, Precision=0.035, Recall=0.004
Top@300: TP=9, Precision=0.030, Recall=0.005
Top@400: TP=13, Precision=0.033, Recall=0.008
>>> Pour adamic_adar :

Top@50: TP=7, Precision=0.140, Recall=0.004
Top@100: TP=8, Precision=0.080, Recall=0.005
Top@200: TP=14, Precision=0.070, Recall=0.008
Top@300: TP=15, Precision=0.050, Recall=0.009
Top@400: TP=17, Precision=0.043, Recall=0.010

Les résultats sur Caltech36 montrent déjà que Common Neighbors et Adamic-Adar semblent surpasser Jaccard. Adamic-Adar obtient les meilleurs résultats globaux avec un Top@400 de 17 liens retrouvés. Toutefois, la précision diminue avec le nombre de prédictions k, ce qui est attendu. Le rappel reste très faible (maximum 1%), ce qui souligne la difficulté du problème : très peu de paires de nœuds sont réellement connectées. Ces résultats suggèrent que des méthodes simples comme Adamic-Adar peuvent être efficaces dans des réseaux sociaux denses, mais que la prédiction de liens reste difficile sans information supplémentaire comme ici.

Question 4.d

C’est la partie où on nous demande de comparer le code des question 4.b et 4.c avec deux graphes différents.

Pour les trois plus petits graphes (Caltech36, Reeds98 et Simmons88), j’obtiens les résultats suivants :

	graph	method	k	tp	precision	recall
2	Caltech36.gml	AdamicAdar	100	58	0.58	0.034835
0	Caltech36.gml	CommonNeighbors	100	58	0.58	0.034835
1	Caltech36.gml	Jaccard	100	38	0.38	0.022823
5	Reed98.gml	AdamicAdar	100	38	0.38	0.020202
3	Reed98.gml	CommonNeighbors	100	37	0.37	0.019670
4	Reed98.gml	Jaccard	100	32	0.32	0.017012
8	Simmons81.gml	AdamicAdar	100	60	0.60	0.018193
6	Simmons81.gml	CommonNeighbors	100	62	0.62	0.018799
7	Simmons81.gml	Jaccard	100	20	0.20	0.006064

Pour chaque graphe, on a:

- Supprimé aléatoirement 10% des arêtes (simulation de liens manquants)
- Appliqué les prédicteurs sur le graphe incomplet
- Classé toutes les paires non connectées selon leur score
- Évalué les top 100 prédictions via les métriques : nombre de vrais positifs (TP), precision@100, recall@100

Ces résultats montrent que **Common Neighbors** et **Adamic/Adar** sont les prédicteurs les plus efficaces, avec des performances presque identiques. Sur l’ensemble des trois graphes testés :

- Ils atteignent des précisions élevées (entre 0.58 et 0.62), en retrouvant plus de la moitié des arêtes supprimées parmi les 100 meilleures prédictions.
- Le recall reste faible, car seule une petite fraction des arêtes supprimées est prédite parmi un très grand nombre de paires possibles. C’est attendu dans les grands réseaux sociaux.
-

En revanche, le prédicteur Jaccard obtient des performances nettement inférieures sur tous les graphes. Cela peut s'expliquer par le fait que Jaccard pénalise les nœuds très connectés, ce qui est désavantageux dans des graphes sociaux comportant des hubs.

Question 5

Dans les questions précédentes, nous avons analysé les graphes sous plusieurs angles et en considérant plusieurs critères. Nous allons maintenant exploiter les données pour en tirer de nouvelles conclusions.
Tout d'abord nous cherchons à utiliser la structure du graphe afin de prédire les étiquettes manquantes des arêtes donc à prédire les groupes des uns en connaissant les groupes des autres.

Question 5.a

Nous manipulons ici des données qui sont déjà partiellement étiquetées et nous cherchons à prédire les labels des sommets restants. Étant donné ces informations, il est préférable d'utiliser un **algorithme de propagation par minimisation**, car celui-ci donne des résultats plus stables car on se base sur une formulation mathématique. Cet algorithme est plus coûteux et plus complexe qu'un LPA (Label Propagation Algorithm) mais prend mieux en compte la structure du graphe.

Question 5.b

L'algorithme de propagation par minimisation étudié dans le cours est l'algorithme de Zhu et. al. Cet algorithme se sert des modules pytorch et networkx afin d'implémenter une version vectorisée de notre algorithme.

On procède ainsi :

- On récupère A, la matrice d'adjacence et D la matrice des poids normalisée
- On initialise un vecteur Y afin de stocker les labels connus (Y_l) et les labels trouvés
- On répète ces étapes jusqu'à la convergence
    - $Y^{(t+1)} \leftarrow P.Y^{(t)}$
    - $Y_l^{(t+1)} \leftarrow Y_l^{(t)}$
- On retourne $Y^{(t)}$

Question 5.c

Dans cette question, le réseau choisi est celui de Caltech36 car c'est un réseau plus petit ce qui permet de réduire le temps de calcul.

Grâce à la fonction random.sample, on peut choisir un échantillon de sommets au hasard puis on leur retire artificiellement leurs étiquettes 'dorm', 'major_index' et 'gender'. On finit par prédire les attributs que l'on vient de retirer grâce à la fonction codée précédemment.

Question 5.d

On commence par implémenter les fonctions de la MAE (mean absolute error) et de l'accuracy.
La MAE sert à connaître à quel point les valeurs estimées sont proches des valeurs réelles alors que l'accuracy sert plutôt à savoir si le bon label a été trouvé.

Ensuite on calcule ces valeurs pour tous les scénarios évoqués précédemment et voici les résultats :

```
Mean absolute error
                  dix     vingt     trente
-----------  --------  --------  ---------
dorm          4.74001   3.63661  0.0538237
major_index   8.35087   5.46699  0.105355
gender       13.3439   11.5734   0.177327


Accuracy Score
                  dix     vingt    trente
-----------  --------  --------  --------
dorm         0.903771  0.903771  0.945384
major_index  0.803641  0.804941  0.873862
gender       0.707412  0.715215  0.788036
```

Ici on n'implémente pas le F1-score car il n'est pas adapté au problème : un F1-score considère des données qui peuvent être positives ou négatives. On traite alors des vrais positifs, des faux positifs, des vrais négatifs et des faux négatifs ce qui n'est pas le cas dans notre étude, ici il n'y a que des label ou une absence de label.

Question 5.e

De manière générale, on constate que plus le pourcentage de labels inconnus augmente, plus les prédictions se dégradent : la MAE auglente fortement et l'accuracy diminue, ce qui montre bien que l'algorithme dépend fortement du nombre de labels disponibles pour bien propager l'information.

On constate qu'en fonction des attributs, les résultats ne sont pas les mêmes car certains attributs se prêtent mieux à la propagation qu'aux autres, tout dépend de leur relation à la structure du réseau.

- Pour **'dorm'** la performance est la meilleure. Cela peut-être relié à la présence de communautés locales ou de clusters bien définis. On constate que ce label est également peu sensible au manque d'information.

- Pour **‘major_index’**, les résultats sont bons mais moins net et plus diffus dans le graphe. On peut émettre l’hypothèse que les étudiants d’un même major sont partiellement connectés mais moins fortement.
- Enfin pour **‘gender’** les prédictions sont moins cohérentes. On peut en conclure que le genre n’est pas structuré dans le graphe et ne peut pas vraiment être prédit par la topologie du réseau. Ceci rend ce label plus sensible à la perte d’information.

Enfin, on constate que pour 30% des labels manquants, on a encore des performances correctes mais qu’elles sont déjà bien dégradées, particulièrement pour les labels peu corrélés à la structure.

En conclusion, **la propagation seule est efficace si et seulement si l’attribut est aligné avec la topologie.**

Question 6

Question 6.a

Dans la question 5, nous avons procédé à la sélection de données aléatoires afin d'analyser les performances de l'algorithme de propagation. En distinguant l'efficacité par label, on a pu voir que certains labels étaient plus représentés dans la structure du graphe et que ceci impactait directement les résultats de l'algorithme.

Dans cette question de recherche, nous nous intéresserons non pas à l'efficacité sur les labels mais à l'efficacité d'un tel algorithme en fonction des données fournies : que se passe-t-il si on ne fournit à l'algorithme que des noeuds de très haut degré ou à l'inverse de très bas degré ?

Une formulation de la problématique peut être la suivante :
Comment la position structurelle (mesurée par le degré des nœuds) des nœuds étiquetés influence-t-elle la performance des algorithmes de propagation de labels en classification semi-supervisée ?

Hypothèse :
En retirant les labels des noeuds les plus connectés, on complexifie la propagation et on aurait donc des performances moins bonnes. A l'inverse, en retirant les labels des noeuds les moins connectés, on n'influe que peu sur la propagation des labels, les résultats sont donc meilleurs.

Question 6.b

Pour répondre à cette question, nous avons évalué les perfomances de l'algorithme de propagation de labels sur les attributs 'gender', 'dorm', et 'major_index' en retirant 20% des labels selon 3 stratégies :

- Suppression aléatoire des labels,
- Suppression des labels des 20% de noeuds avec le degré le plus élevé (noeuds les plus connectés),
- Suppression des labels des 20% de noeuds avec le degré le plus faible (noeuds peu connectés).

Pour chaque scénario nous avons calculé la MAE et l'accuracy.

Voici les résultats obtenus :

```
Mean absolute error
               randomly    high_degree    low_degree
-----------  ----------  -------------  ------------
dorm            4.74001        3.63661     0.0538237
major_index     8.35087        5.46699     0.105355
gender         13.3439        11.5734      0.177327


Accuracy Score
               randomly    high_degree    low_degree
-----------  ----------  -------------  ------------
dorm           0.903771       0.903771      0.945384
major_index    0.803641       0.804941      0.873862
gender         0.707412       0.715215      0.788036
```

Question 6.c

En analysant les résultats obtenus, on constate premièrement que les performances sont meilleures lorsque les labels sont retirés aux nœuds de degré faible. Cela confirme que ces nœuds, peu connectés, sont moins essentiels à la diffusion des labels. Leurs propres labels peuvent donc être retrouvés plus facilement grâce à leurs voisins connectés, sans nuire à l'ensemble du système.

La suppression des labels des nœuds à degré élevé provoque une dégradation notable mais moins sévère qu'avec une suppression complètement aléatoire, ce qui est contraire à l'hypothèse de départ. Cela montre que les nœuds très connectés participent fortement à la propagation, et leur absence réduit la qualité globale des prédictions.

La suppression aléatoire de labels donne systématiquement des résultats moins bons que la suppression ciblée sur les nœuds faiblement connectés. Cela suggère que le choix des nœuds étiquetés a une importance stratégique.

Ces résultats renforcent l'idée que l'information portée par les nœuds les plus connectés est cruciale pour la propagation efficace des labels. À l'inverse, les nœuds de faible degré peuvent être laissés non étiquetés sans que cela nuise fortement à la performance globale du modèle. Ainsi, dans un contexte de semi-supervision où le coût d'annotation est élevé, il serait judicieux de prioriser l'étiquetage des nœuds centraux ou à haut degré, car leur contribution à la diffusion des labels est déterminante.