Apprentissage par Renforcement pour la Réfutation de Conjectures sur les Graphes

Par

Andy Torres

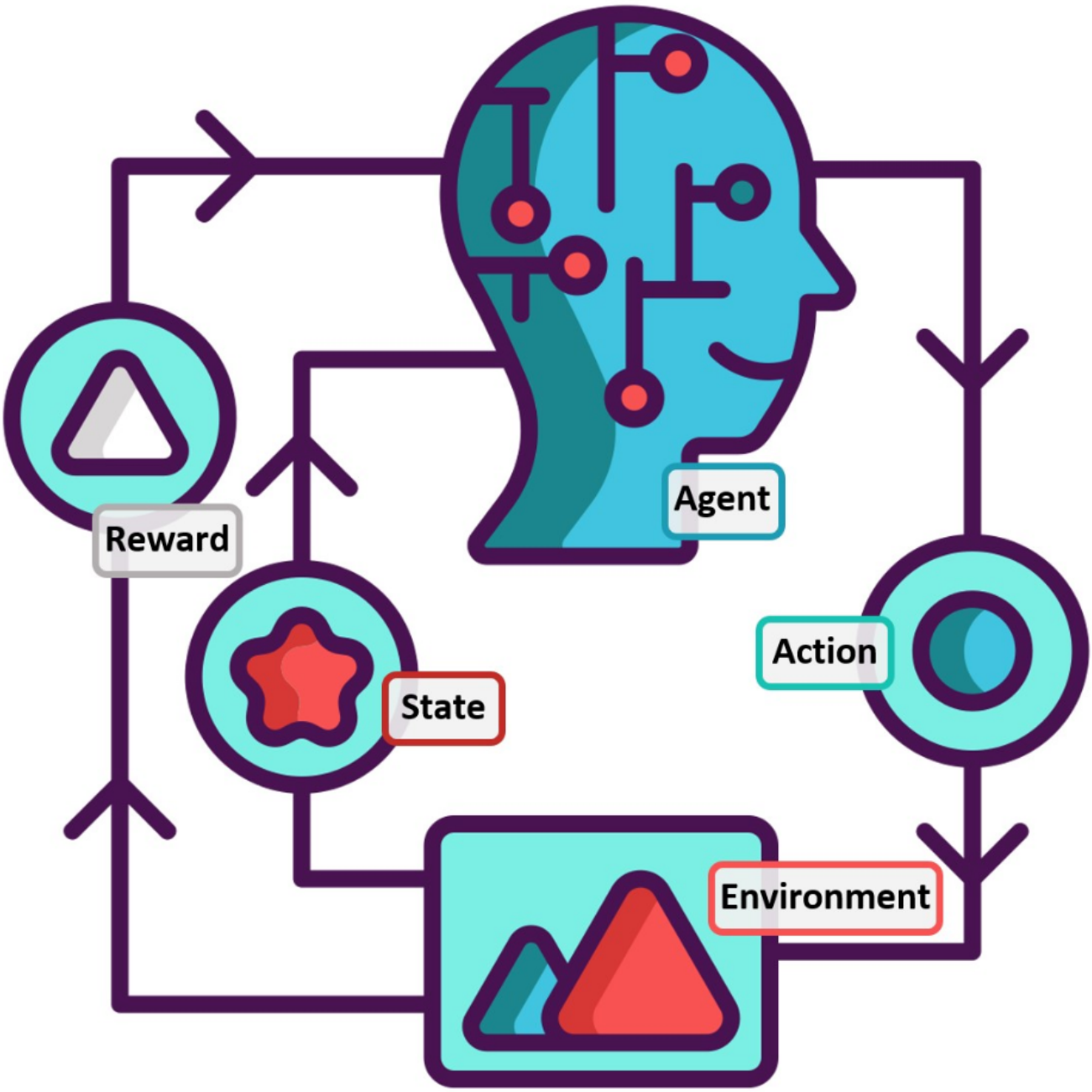
Stage de Licence 3 Informatique 2022-2023

Sous la supervision de Guillaume Fertin et Géraldine Jean

Laboratoire des Sciences du Numérique de Nantes (LS2N)

Table des Matières

1 Introduction



Source: www.sefidian.com/2022/09/06/reinforcement-learning-q-learning-numerical-example/

1.1 Définition

L’apprentissage par renforcement est une méthode d’apprentissage machine dans laquelle un agent prend des décisions dans le but de maximiser sa récompense cumulée. L’agent évolue dans un environnement dans lequel chacune de ses décisions est évaluée. Si son comportement est celui attendu, alors, il reçoit une récompense, sinon, il est fortement pénalisé. Ainsi, à la poursuite de la maximisation de sa récompense, il en apprend davantage sur son environnement et les comportements désirés et réprimandés, ce qui lui permet d’améliorer ses décisions au fil du temps.

L'apprentissage par renforcement trouve des applications dans divers domaines, tels que la robotique, les jeux, l'optimisation des ressources, la gestion des stocks et la prise de décision automatisée, où des agents autonomes doivent apprendre à agir de manière intelligente et adaptative dans des environnements complexes et changeants.

2 Objet du stage

L'objectif de ce stage est d'essayer de réfuter des conjectures portant sur les graphes grâce à l'utilisation d'un algorithme d'apprentissage par renforcement nommé "deep-cross entropy". Notre travail a également pour but de questionner la production parfois excessive de nouvelles conjectures et de fournir un outil efficace pour faire le tri entre les vraies et les fausses.

La méthode utilisée fonctionne comme suit:
À chaque itération, l'algorithme produit une certaine quantité de graphes donnée par l'utilisateur. Parmi les graphes générés, on sélectionne les 10% des meilleurs graphes (ceux avec la récompense la plus élevée) qui sont conservés à la prochaine itération. La récompense associée au graphe étant la quantité que l'on souhaite maximiser pour obtenir un contre-exemple. Puis, on utilise les 50% des meilleurs graphes qui servent à ajuster les paramètres du modèle afin de le "guider" vers le résultat souhaité. Ces ajustements sont réalisés grâce à la méthode d'optimisation de la descente du gradient stochastique qui met à jour les "poids", définis comme étant les paramètres permettant de contrôler la précision du modèle et ses décisions, de manière à maximiser une fonction objectif que l'on a définie comme étant la récompense.

Pour illustrer ce qui précède, on peut prendre l'exemple d'une conjecture telle qu'une quantité λ est toujours inférieure ou égale à 0 dans un graphe connexe. Ici, la récompense associée au graphe serait λ puisqu'on cherche à maximiser λ. Ainsi, un contre-exemple consisterait à trouver un graphe connexe dont λ serait strictement supérieur à 0 pour réfuter la conjecture (dans quel cas on s'arrête et on affiche le graphe trouvé). Pour cela, l'algorithme conserverait les 10% des meilleurs graphes ayant le λ le plus élevé, puis il utiliserait les 50% des meilleurs graphes comme support pour ajuster les paramètres du modèle et faire en sorte que l'algorithme génère davantage de graphes avec un λ élevé. Par conséquent, l'algorithme produirait des graphes avec un λ croissant qui tendrait vers une valeur positive (si un contre-exemple existe).

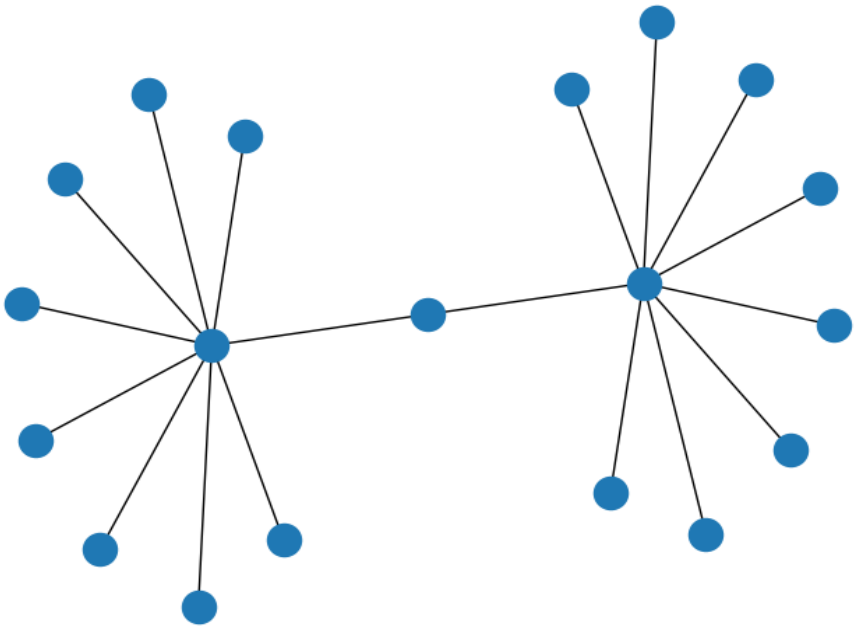
3 Réfutations de conjectures

3.1 Conjecture 2.1

3.1.1 Approche initiale

La conjecture 2.1 affirme que:
Pour tout graphe connexe d'au moins n ≥ 3 sommets, avec λ_1, la plus grande valeur propre de sa matrice d'adjacence et μ, son matching number.
Alors, $\lambda_1 + \mu \geq \sqrt{n-1} + 1$.
Par conséquent, nous réécrivons l'inéquation sous cette forme pour qu'elle dépende de 0 (ceci est une convention que nous avons prise pour une meilleure lisibilité):
$0 \geq \sqrt{n-1} + 1 - (\lambda_1 + \mu)$.
Il en découle la fonction de score qui sera identique au membre de droite de cette inéquation. En tentant de maximiser sa récompense (ici - (λ_1 + μ)) , le membre de droite augmentera progressivement jusqu'à potentiellement devenir positif.

Comme Adam Wagner, nous sommes parvenus à trouver un contre-exemple identique à 19 sommets en quelques heures.

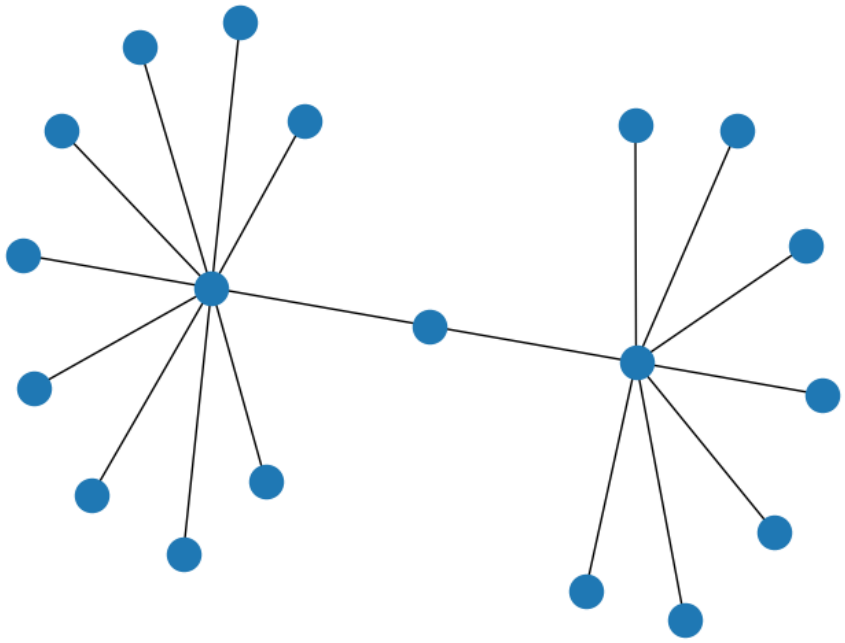


Contre-exemple à la Conjecture 2.1

3.1.2 Le codage de Prüfer

Cependant, cette implémentation convergeait relativement lentement vers un contre-exemple, ce qui nous limitait à quelques tests seulement par jour. C'est pourquoi nous avons décidé de changer la forme des graphes générés par la machine. En effet, l'apprentissage par renforcement pousse l'agent à générer des graphes dans l'espoir qu'il maximise sa récompense cumulée.

Or, bon nombre de graphes générés sont directement "jetés" du fait de leur non-conformité à nos attentes (notamment les graphes non connexes), ce qui ralentit d'autant la convergence vers un contre-exemple. Par ailleurs, nous avons observé que les contre-exemples trouvés par la machine étaient uniquement des arbres. Ainsi, nous avons forcé l'agent à ne générer que des arbres grâce à l'utilisation du codage de Prüfer permettant de représenter un arbre à n sommets en une séquence de n-2 sommets. Cela nous a permis de baisser drastiquement l'espace de stockage nécessaire aux graphes quelconques, représentés initialement par un tableau binaire en 2 dimensions (i et j) à n*(n-1)/2 cases (une pour chaque arête dans un graphe non orienté) où un "1" marque la présence d'un arc de i vers j et "0" son absence. Grâce à cette modification, le temps d'exécution a énormément diminué, passant de plusieurs heures à quelques minutes au maximum pour un résultat identique.

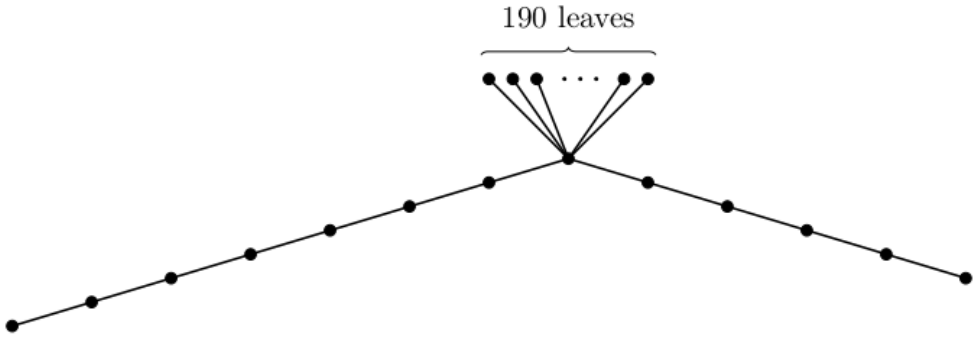


Second contre-exemple à la Conjecture 2.1
(On notera une légère asymétrie: on compte 7 sommets d'un côté et 9 de l'autre)

3.2 Conjecture 2.3

La conjecture 2.3 affirme que:
Pour tout graphe connexe d'au moins n ≥ 4 sommets, avec son diamètre D, sa proximité π, et l'ensemble des valeurs propres de sa matrice de distance $\partial 1 \geq \ldots \geq \partial n$.
Alors, $\pi + \partial_{\lfloor \frac{2D}{3} \rfloor} > 0$.

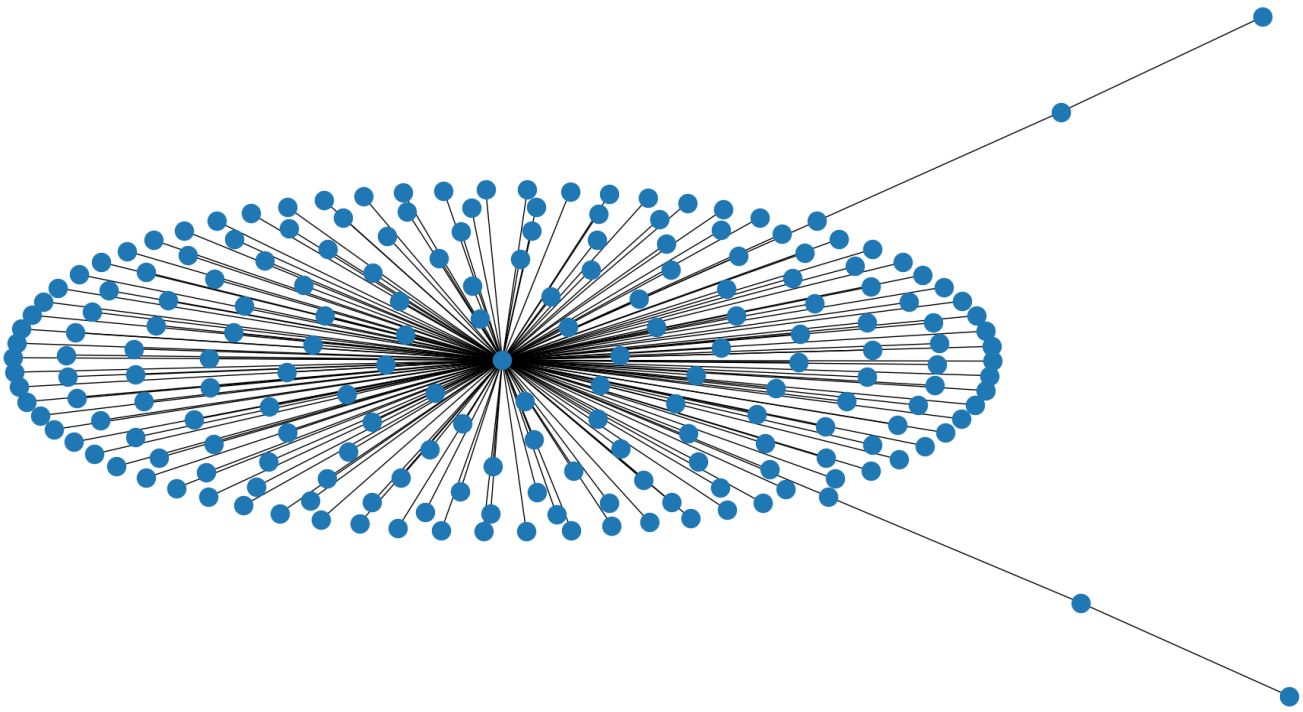
Dans son article, Adam Wagner n'avait pas réussi à trouver un contre-exemple à cette conjecture à cause d'un temps de convergence trop long. En effet, l'implémentation classique de la conjecture sur un graphe à 30 sommets seulement prenait déjà plusieurs jours pour converger vers une valeur de -0.4. Ainsi, il avait lui-même dessiné à la main ce à quoi ce contre-exemple devait ressembler.



Contre-exemple théorique

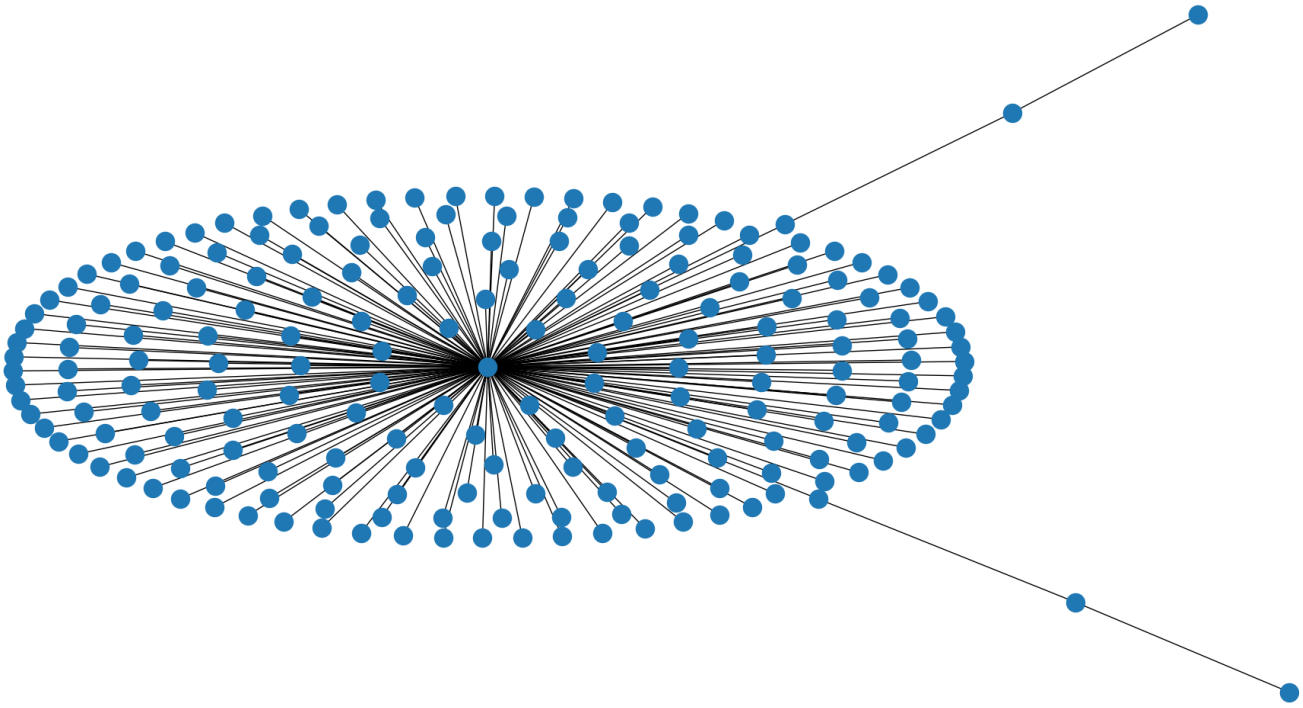
Il est important de souligner que ce graphe connexe à 203 sommets se trouve être un arbre.

Partant du postulat qu'il existait bien un contre-exemple et que ce dernier était un arbre, nous avons implémenté une seconde fois le codage de Prüfer. Cela nous a permis de trouver un contre-exemple à 203 sommets similaire à celui dessiné en moins d'une heure.



Contre-exemple à 203 sommets trouvé par la machine

Motivés par cet accomplissement, nous avons essayé de trouver un contre-exemple plus petit, ce qui nous a permis de découvrir le contre-exemple à 201 sommets suivant.



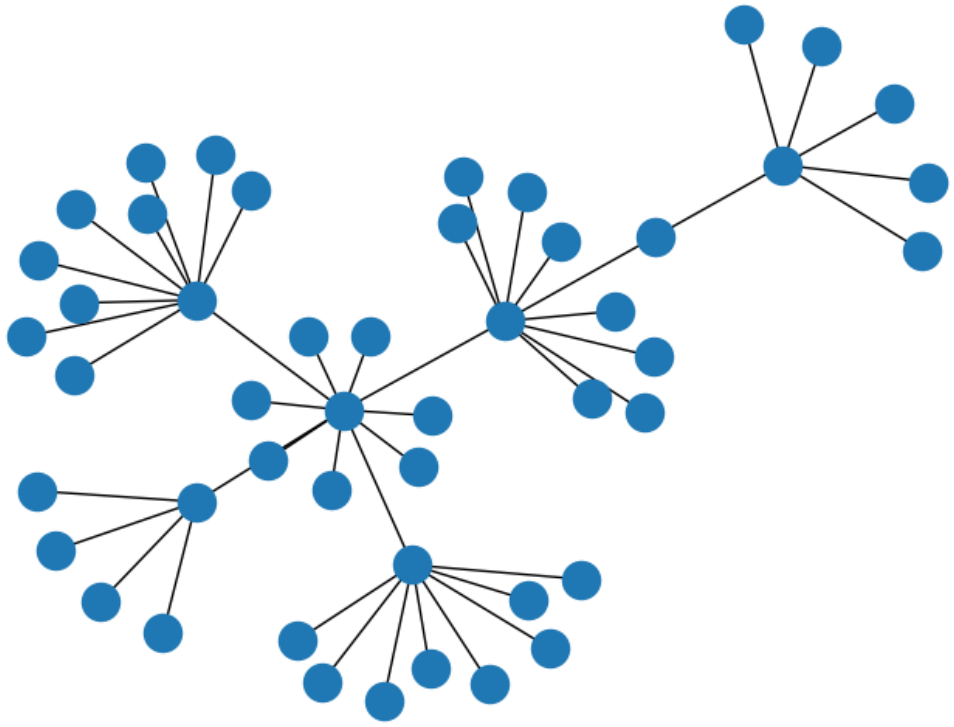
Contre-exemple à 201 sommets

3.3 Conjecture 2.4

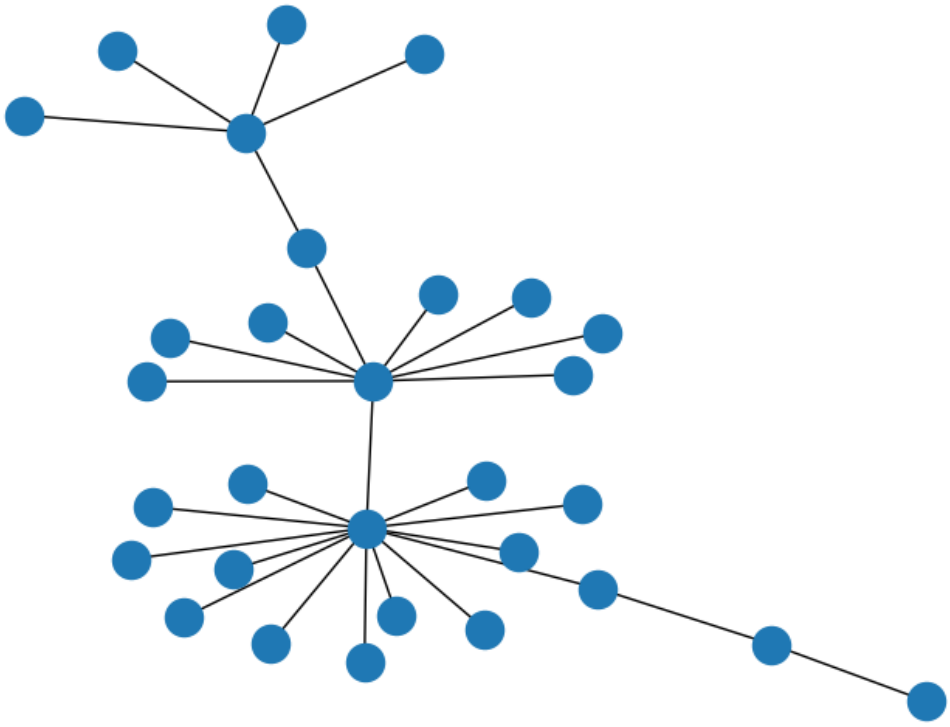
La conjecture 2.4 affirme que:
La séquence des valeurs absolues des coefficients non nuls de CPA(T) forme une séquence unimodale, et son sommet se trouve au même emplacement que le sommet des coefficients normalisés de CPD(T).

Un contre-exemple est trouvé lorsque la différence de position entre les deux sommets est d'au moins 0.3. Nous pensons que cela s'explique par le théorème 2.5 de l'article d'Adam Wagner qui implique qu'il existe une infinité d'arbres tels que la différence entre leurs 2 sommets définis ci-dessus soit d'au moins 0.3.

À l'instar d'Adam Wagner, nous avons des résultats similaires sur les graphes à 48 et 30 sommets présentés ci-dessous.

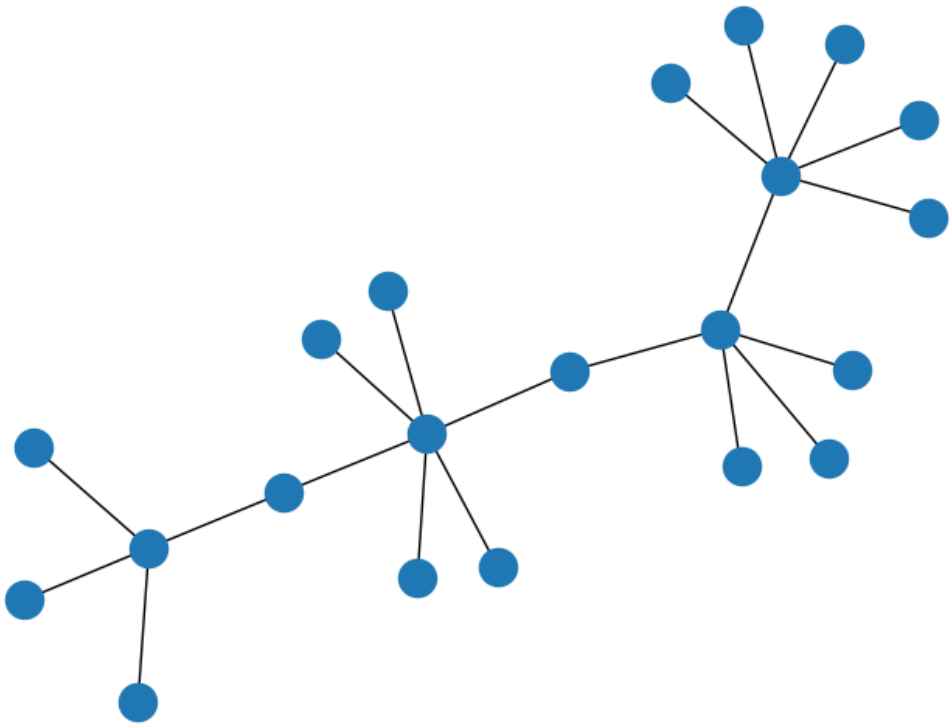


Contre-exemple à 48 sommets



Contre-exemple à 30 sommets

Cependant, nous avons également réussi à trouver le contre-exemple à 21 sommets dessiné ci-dessous.



Contre-exemple à 21 sommets à la Conjecture 2.4

3.4 Conjecture $\lambda_1 * \pi$

La conjecture $\lambda_1 * \pi$ affirme que:
Pour tout graphe à n sommets, le produit de sa valeur propre λ_1 et de sa proximité π est tel que:
$\sqrt{n-1} \leq \lambda_1 * \pi \leq n - 1$

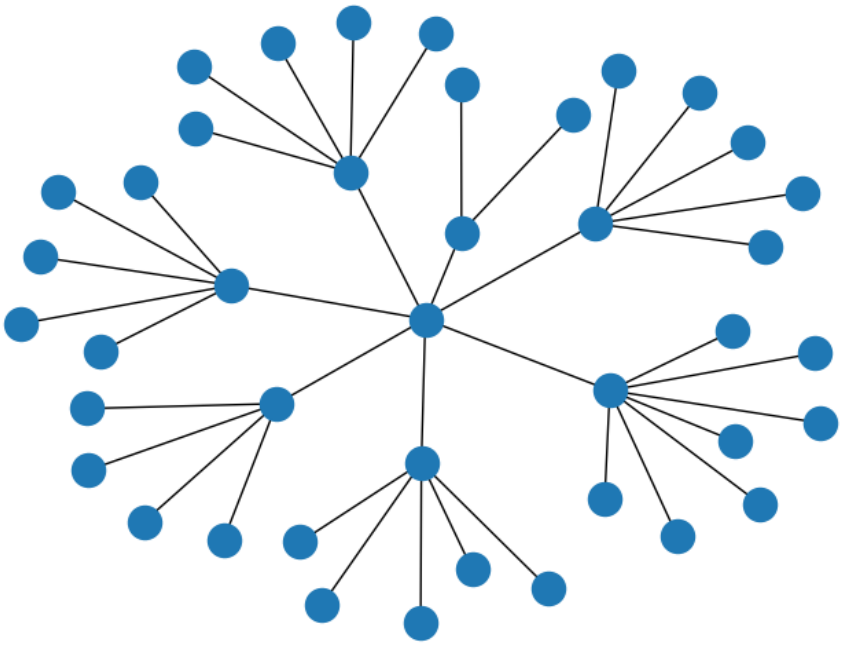
Cette conjecture étant un encadrement de la valeur $\lambda_1 * \pi$, nous pouvons la réécrire sous la forme de deux inégalités disjointes:

$\sqrt{n-1} - \lambda_1 * \pi \leq 0$ et $\lambda_1 * \pi - (n - 1) \leq 0$.

On en déduit les deux fonctions de score suivantes correspondant aux membres de gauche de chaque inégalité, soient $\sqrt{n-1} - \lambda_1 * \pi$ et $\lambda_1 * \pi - (n - 1)$.

Par conséquent, un contre-exemple est trouvé si et seulement si au moins l'une de ses fonctions retourne un résultat positif.

En appliquant la première fonction sur la borne inférieure ($\sqrt{n-1} - \lambda_1 * \pi$), nous avons trouvé le graphe à 41 sommets dessiné ci-dessous:



avec $\lambda_1 = 3.4606122417083864$ et $\pi = 1.825$ tels que $\sqrt{40} - 3.4606122417083864 * 1.825 = 0.008937979219 > 0$.

3.5 Conjecture $\lambda_1 + D$

La conjecture $\lambda_1 + D$ affirme que:
Pour tout graphe à n sommets, la somme de sa valeur propre λ_1 et de son diamètre D est telle que:
$\lambda_1 + D \leq n - 1 + 2 * cos \frac{\pi}{n+1}$ où π est sa proximité.

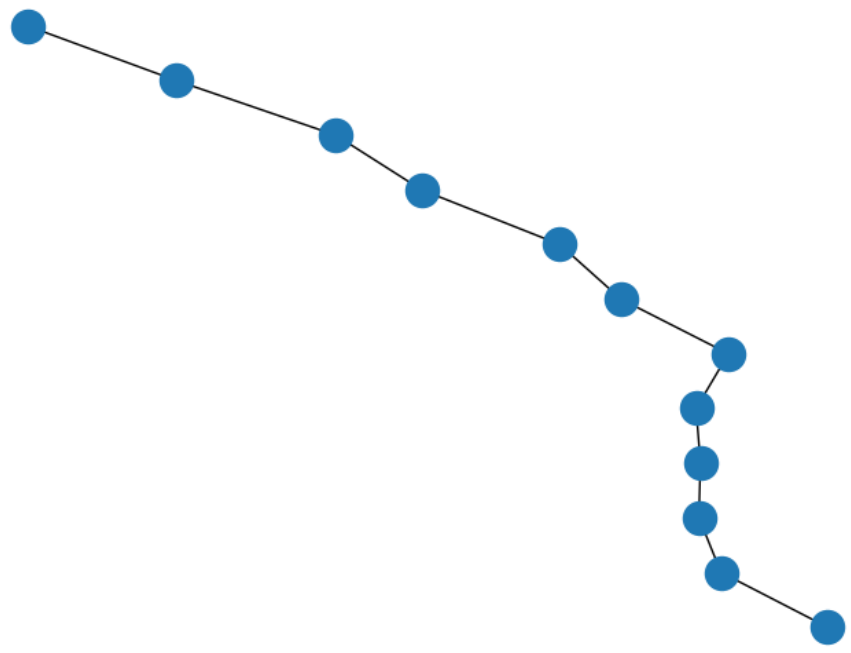
Ainsi, nous pouvons la réécrire une nouvelle fois en fonction de 0:

$\lambda_1 + D - (n - 1 + 2 * cos \frac{\pi}{n+1}) \leq 0.$

On en déduit la fonction de score suivante correspondant au membre de gauche de cette inégalité, soit $\lambda_1 + D - (n - 1 + 2 * cos \frac{\pi}{n+1})$.

Par conséquent, un contre-exemple est trouvé si et seulement si la fonction retourne un résultat positif.

En appliquant cette fonction, nous sommes parvenus à trouver le graphe suivant à 12 sommets:



avec $\lambda_1 = 1.941883634852104$ et $\pi = 3.272727272727273$ et D = 11 tels que
$1.941883634852104 + 11 - (12 - 1 + 2 * cos \frac{3.272727272727273}{12+1} = 0.004926801166581285 > 0.$

3.6 Conjecture $\lambda_1 - \alpha$

La conjecture $\lambda_1 - \alpha$ affirme que:
Pour tout graphe connexe à n sommets, la différence de sa valeur propre λ_1 et de son nombre indépendant α est telle que:

$\sqrt{n-1} - n + 1 \leq \lambda_1 - \alpha$

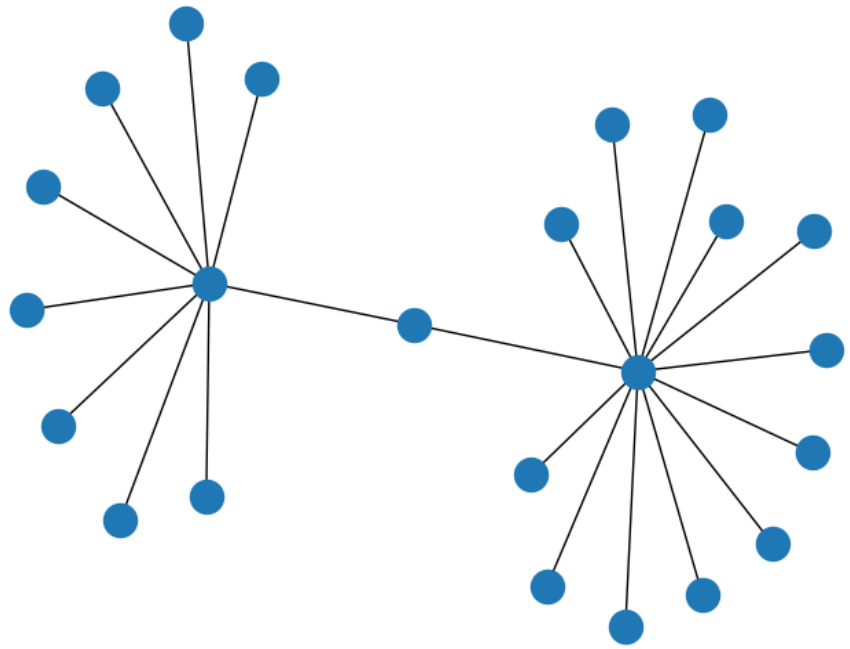
De manière analogue à la conjecture précédente, nous pouvons la réécrire en fonction de 0:

$\sqrt{n-1} - n + 1 - (\lambda_1 - \alpha) \leq 0$

On en déduit la fonction de score suivante correspondant au membre de gauche de cette inégalité soit $\sqrt{n-1} - n + 1 - (\lambda_1 - \alpha)$

Par conséquent, un contre-exemple est trouvé si et seulement si $\sqrt{n-1} - n + 1 - (\lambda_1 - \alpha)$ est positif.

En appliquant cette fonction, nous sommes parvenus à trouver le graphe suivant à 23 sommets:



avec $\lambda_1 = 3.6381407308541265$ et $\alpha = 21$ tels que $\sqrt{22} - 23 + 1 - (3.6381407308541265 - 21) = 0.05227502897 > 0$. On remarque qu'il ressemble au contre-exemple trouvé à la conjecture 2.1.

3.7 Conjecture $\lambda_1 * \mu$

La conjecture $\lambda_1 * \mu$ affirme que:
Pour tout graphe connexe à n sommets, le produit de sa valeur propre λ_1 et de son matching number μ est tel que:

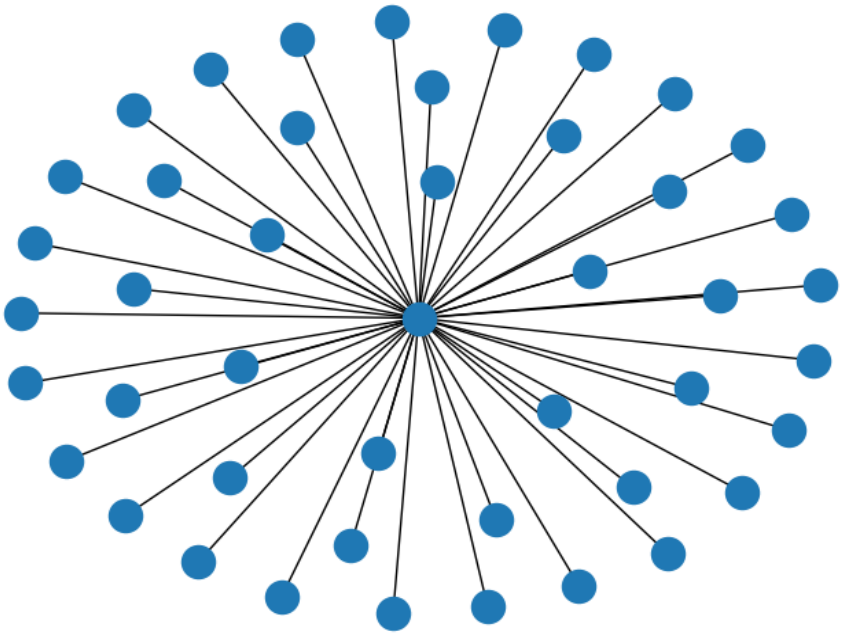
$\sqrt{n-1} \leq \lambda_1 * \mu$

Par conséquent, nous pouvons la réécrire sous la forme suivante comme pour les cas précédents: $\sqrt{n-1} - \lambda_1 * \mu \leq 0$

Il en découle la fonction de score suivante correspondant au membre de gauche de l'inégalité soit: $\sqrt{n-1} - \lambda_1 * \mu$

Ainsi, un contre-exemple est trouvé si et seulement si cette fonction retourne un résultat positif.

De nouveau, nous avons trouvé le contre-exemple à 45 sommets en étoile présenté ci-dessous:



avec λ_1 = 6.612937783920284 et μ = 1 tels que:

$\sqrt{44}$ - 6.612937783920284 * 1 = 0.0203117967 > 0

4 Conclusion

En conclusion, ce stage a été une opportunité passionnante d'explorer l'application de l'apprentissage par renforcement dans le domaine des graphes et de la réfutation de conjectures. Grâce à l'algorithme "deep-cross entropy", nous avons pu générer et évaluer un grand nombre de graphes dans le but de trouver des contre-exemples à des conjectures ouvertes depuis plusieurs années. Cela nous a permis de nous rendre compte de l'efficacité et de l'utilité de ce type d'approche. En effet, comme soulevé dans l'introduction, l'apprentissage par renforcement peut être un outil pertinent dans la boîte à outils d'un chercheur, lui permettant de gagner en précision et vitesse dans ses recherches. Concrètement, un mathématicien qui cherche à réfuter une théorie sur les graphes par exemple pourra utiliser cet outil afin de vérifier plus facilement ses hypothèses au lieu d'effectuer des calculs fastidieux, où la probabilité d'erreur est élevée, pour parvenir à un résultat identique.

Ce stage a également mis en évidence l'importance de la collaboration entre la théorie des graphes et l'apprentissage automatique. En combinant les connaissances et les techniques de ces deux domaines, nous avons pu explorer de nouvelles approches pour aborder des problèmes complexes et de nouvelles perspectives de recherche.

En somme, cela a été une expérience enrichissante qui m'a permis de développer des compétences avancées en apprentissage par renforcement et en théorie des graphes. Je suis reconnaissant d'avoir eu l'occasion de contribuer à la recherche dans ce domaine fascinant et j'espère pouvoir continuer à me former dans cette discipline au cours des prochaines années.

Pour conclure, je serais honoré que mon travail fasse partie d'un mouvement de fond visant à appliquer des techniques d'apprentissage automatique dans des domaines divers et variés pour faire avancer la recherche.

5 Glossaire

Descente du gradient stochastique (SGD): Algorithme permettant l'optimisation des paramètres du modèle. Il permet de guider le modèle vers la bonne direction en se basant sur une partie du jeu de données et non la totalité comme dans l'algorithme de descente du gradient original.

Matrice d'adjacence: Matrice carrée de taille N*N avec N le nombre de sommets du graphe. Elle représente les arêtes entre chacun des sommets du graphe.

Matrice de distance: Matrice carrée de taille N*N avec N le nombre de sommets du graphe. Elle représente les distances du plus court chemin entre chaque paire de sommets du graphe.

Valeur propre: Nombre qui caractérise certaines propriétés d'une matrice. Pour une matrice carrée A, un nombre λ est une valeur propre de A si et seulement s'il existe un vecteur non nul x tel que A*x = λ*x.

Matching number: Taille du plus grand ensemble d'arêtes dont aucune paire dans cet ensemble n'a de sommet en commun.

Proximité: Notion faisant référence à la proximité entre les sommets dans un graphe. Plus la distance du plus court chemin séparant 2 sommets du graphe est courte, plus la proximité est élevée.

CPA(T): Polynôme caractéristique de la matrice d'adjacence d'un graphe, calculé à partir du déterminant de cette matrice. Cela consiste à soustraire la matrice A multipliée par la matrice identité de dimension équivalente, puis à calculer le déterminant de cette nouvelle matrice. Le déterminant s'exprime donc sous la forme P(X) = det(X*I-A). Ces racines sont d'ailleurs égales aux valeurs propres de A.

Séquence unimodale: Séquence $(a_1, a_2, ..., a_n)$ telle qu'il existe un indice k ($1 \leq k \leq n$) où les éléments de la séquence $(a_1, a_2, ..., a_k)$ sont strictement croissants et les éléments de la séquence $(a_k, a_{k+1}, ..., a_n)$ sont strictement décroissants. Pour illustrer ce concept, on peut prendre l'exemple de la pente d'une montagne dont la hauteur augmente linéairement jusqu'au sommet puis diminue pendant la descente.

CPD(T): Polynôme caractéristique de la matrice de distance d'un graphe.

Normalisé: La normalisation d'un coefficient consiste à ramener ce coefficient à un nombre compris entre 0 et 1.

$p_D(T)$: Indice du "sommet" de la séquence formée par les coefficients normalisés de CPD(T).

Diamètre: Le diamètre d’un graphe correspond à la plus grande distance possible entre deux de ses sommets. Il s’agit donc de la plus grande distance dans la matrice de distance.

Ensemble indépendant: Ensemble de sommets donc aucune paire de cet ensemble ne partage d’arête en commun.

Nombre indépendant: Taille du plus grand ensemble indépendant.

6 Remerciements

J’aimerais à présent remercier Géraldine Jean et Guillaume Fertin d’avoir accepté ma candidature pour ce stage et de m’avoir accompagné et conseillé tout au long de cette période d’apprentissage. Je remercie également Adam Wagner pour son aide précieuse pour l’implémentation du codage de Prüfer.

7 Bibliographie

L’article de recherche d’Adam Wagner datant de 2021: https://arxiv.org/pdf/2104.14516.pdf

L’article de M. Aouchiche et P. Hansen d’où sont tirées les quatre dernières conjectures datant de 2009: https://www.sciencedirect.com/science/article/pii/S0024379509003061?via%3Dihub