

# Biml GNN (trouver un nom)

Bonhoure Timothé, Martinez Christophe

October 29, 2023

## Contents

<b>1</b>	<b>Méthode</b>	<b>2</b>
<b>2</b>	<b>Étude du dropout</b>	<b>2</b>
<b>3</b>	<b>Performance du décodeur</b>	<b>2</b>
<b>4</b>	<b>Test du dropout en présence du décodeur</b>	<b>2</b>
<b>5</b>	<b>Performance de décodeur</b>	<b>2</b>
<b>6</b>	<b>Performance de décodeur</b>	<b>2</b>
6.1	Méthode . . . . .	2
6.2	Performance . . . . .	2
6.3	Performance avec le degré . . . . .	3
<b>7</b>	<b>Reconstruction de graphe</b>	<b>3</b>
7.1	Méthode . . . . .	3
7.2	Résultats . . . . .	4
7.3	Résultats bis . . . . .	4
<b>8</b>	<b>Ablation study</b>	<b>4</b>

## Abstract

# 1 Méthode

Dans le cadre de ce projet, nous avons décidé de développer notre propre décodeur pour la prédiction de liens. Ce décodeur est composé de deux couches linéaires séparées par une fonction d'activation ReLU. Pour évaluer son efficacité, nous avons mis en place deux modèles :

- Un VGAE (Variational Graph Autoencoder) avec un encodeur composé d'une couche GCNConv suivie de deux couches GCNConv pour encoder la moyenne et l'écart type dans l'espace latent.
- Un GAE (Graph Autoencoder) composé d'une seule couche GCNConv.

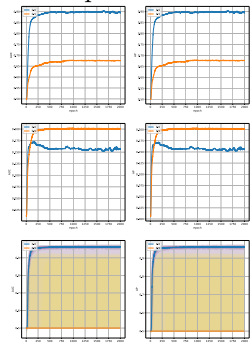
Pour la préparation des données nous utilisons la méthode suivante :

```
torch_geometric.transforms.RandomLinkSplit(is_undirected=True, split_labels=True, num_val=0)
```

Cette méthode nous permet de générer des jeux d'entraînement et de test de liens existants dans les données d'origine (liens positifs) et non existants dans les données d'origine (liens négatifs). Elle nous permet aussi de définir que ces liens sont non orientés et que donc pour chaque paire d'indices formant le lien, le premier indice est inférieur au deuxième. Nous avons opté pour l'utilisation de l'optimiseur Adam avec un taux d'apprentissage de 0,01 et un terme de régularisation (`weight_decay`) de  $5e^{-4}$ .

## 2 Étude du dropout

Nous avons essayé d'améliorer le modèle VGAE en y incorporant une couche de dropout entre la première et la deuxième couche. Nous avons laissé la valeur de dropout à la valeur par défaut soit 0.5. Les modèles ont été entraînés sur 2000 époques



modèle	Sans Degrée		Avec Degrée	
	AUC	AP	AUC	AP
VGAE	Data	Data	d	d
VGAE with dropout	Data	Data	d	d

## 3 Performance du décodeur

## 4 Test du dropout en présence du décodeur

## 5 Performance de décodeur

## 6 Performance de décodeur

### 6.1 Méthode

Nous testons notre décodeur sur ces deux modèles en comparant les performances avec et sans l'ajout du décodeur.

### 6.2 Performance

Nous effectuons les prédictions en utilisant les données de latitude, longitude et pays de chaque nœud. Les modèles sont entraînés sur 1000 époques, et ce processus est répété 10 fois pour obtenir une moyenne des performances par époque. Tous les modèles partagent des hyperparamètres équivalents : les encodeurs produisent des vecteurs de 32 valeurs. On trace ensuite l'AUC et l'AP calculées à chaque époque pour chaque modèle.

Modèle	AUC	AP
GAE	0.72	0.64
GAE avec décodeur	0.94	0.93
VGAE	0.83	0.75
VGAE avec décodeur	0.97	0.96

Table 1: Performance du décodeur en moyenne sur 10 au bout de 1000 époques

### 6.3 Performance avec le degré

Les degrés de chaque nœud sont établis grâce à notre connaissance du graphe réel. Cependant, lorsque nous envisageons d'ajouter un nouvel aéroport à notre base de données, il devient envisageable d'estimer le degré potentiel du nœud associé à cet aéroport en se basant sur les flux d'arrivées et de départs des avions de cet aéroport. Par conséquent, la réalisation de ce test ne nous semble pas inappropriée.

Modèle	AUC	AP
GAE	0.74	0.66
GAE avec décodeur	0.97	0.97
VGAE	0.78	0.70
VGAE avec décodeur	0.97	0.96

Table 2: Performance du décodeur en moyenne sur 10 au bout de 1000 époques avec le degré de chaque nœud

## 7 Reconstruction de graphe

### 7.1 Méthode

Nous avons choisi de tester nos modèles d'une manière plus visuelle, en cherchant à reconstruire un graphique représentant les liaisons aériennes entre les différents aéroports de notre ensemble de données. Pour ce faire, nous avons suivi une préparation des données similaire à celle décrite dans la section précédente. Nous avons ensuite entraîné nos modèles en utilisant une base de liens positifs pour l'entraînement. Par la suite, nous avons généré des graphiques en encodant les mêmes liens positifs et en décodant un ensemble de liens provenant de l'ensemble de données d'origine, ainsi qu'un nombre équivalent de liens négatifs. Nous avons ensuite enregistré plusieurs métriques, notamment le nombre de liens correctement prédits, le nombre de liens faussement prédits (qui n'existent pas dans nos données), le nombre de liens manqués (qui existent dans nos données mais ont été rejetés par le modèle), et le nombre total de liens rejetés. Nous avons utilisés 4 modèles :

- un GAE avec un encodeur formé d'une seule couche de convolution.
- un VGAE sans notre décodeur.
- un VGAE avec notre décodeur.
- un même GAE mais avec notre décodeur.

Chaque modèle a été testé sur un total de 27 093 liens possibles, comprenant 13 547 liens positifs et 13 546 liens négatifs. L'objectif était de couper ce jeu de liens en deux en conservant les liens les plus plausibles pour le modèle. Pour ce faire, chaque modèle attribue un score entre 0 et 1 à chacun des 27 093 liens. Ensuite, à l'aide de la méthode `torch.quantile(z, 0.5)`, nous déterminons un seuil de score pour ne conserver que les  $13\,546 \pm 1$  liens ayant un score supérieur au seuil. Ce sont ces liens qui seront considérés comme prédits positivement par le modèle. Dans nos résultats, les pourcentages associés aux liens corrects et faux sont basés sur le nombre de liens considérés positifs par le modèle (Corrects + Faux) et le pourcentage de liens manqués par rapport au total de liens positifs réels (13 547). La prédiction est faite sur les données de **latitude**, **longitude**, et **pays** de chaque nœud.

Les liens en noir sont les liens correctement prédits.

Les liens en rouge sont les liens faussement prédits.

Les liens en vert sont les liens manqués.

## 7.2 Résultats

Modèle	Min	Max	Moyenne	Écart type	Seuil
GAE	0	1	0.75929	0.42746	1
VGAE	0	1	0.71422	0.45044	1
GAE avec décodeur	0	0.99980	0.49157	0.39727	0.54739
VGAE avec décodeur	0	1	0.56486	0.42113	0.75337

Table 3: Statistiques des scores établies par les modèles

Modèle	Corrects		Faux		Manqués		Rejetés
GAE	13298	64.7%	7257	35.3%	249	1.84%	6538
VGAE	13381	71.0%	5473	29.0%	166	1.23%	8239
GAE avec décodeur	11804	87.1%	1743	12.9%	1743	12.9%	13546
VGAE avec décodeur	12199	90.0%	1348	10%	1348	10%	13546

Table 4: Résultats de prédictions

Il est notable que les modèles sans décodeur ont manqué beaucoup moins de liens par rapport aux modèles avec décodeur. Cependant, cette amélioration s’accompagne d’une plus grande acceptation de faux liens qui auraient dû être rejetés. Cette tendance s’explique en partie par la manière dont ces modèles attribuent des scores, sans faire de distinction nette entre les liens potentiellement positifs et les liens considérés comme certainement positifs.

## 7.3 Résultats bis

Lors de nos tests, nous avons exploré l’incorporation des degrés de chaque nœud pour la prédiction de liens. Nous avons alors cherché à tester de manière plus visuelle nos modèles.

Modèle	Min	Max	Moyenne	Écart type	Seuil
GAE	0	1	0.73815	0.43963	1
VGAE	0	1	0.75625	0.41074	1
GAE avec décodeur	0	1	0.49266	0.45323	0.5414
VGAE avec décodeur	0	1	0.51289	0.43108	0.5917

Table 5: Statistiques des scores établies par les modèles en tenant compte du degré des nœuds

Modèle	Corrects		Faux		Manqués		Rejetés
GAE	13239	66.2%	6745	33.8%	308	2.27%	6797
VGAE	11665	74.5%	3995	25.5%	1882	13.9%	11427
GAE avec décodeur	12695	93.7%	852	6.3%	852	6.3%	12695
VGAE avec décodeur	12535	92.5%	1011	7.5%	1012	7.5%	13547

Table 6: Résultats de prédictions en tenant compte du degré des nœuds

Nous pouvons constater que tous les modèles obtiennent de meilleures performances lorsque le degré des nœuds est pris en compte. Notamment, dans ce contexte, le modèle GAE avec décodeur surpasse en performance le modèle VGAE avec décodeur.

## 8 Ablation study