

# Rapport Projet Décision dans l'incertain

## I. Introduction

On cherche à approximer des probabilités dans des réseaux bayésiens grâce à des algorithmes d'échantillonnages. Nous avons donc implémenté les trois algorithmes : méthode du rejet, weighted sampling et algorithme de Gibbs.

Le projet comporte donc plusieurs parties majeurs :

- Lecture du fichier
- Menu
- Algorithmes

Le projet a été implémenté pour lire des fichiers texte de tout type, la lecture se fait ligne par ligne, un fichier .bif, comme celui donné en exemple, ou un fichier .txt peuvent fonctionner. Cependant, ce fichier doit respecter la mise en page présentée dans le fichier exemple pour ne pas provoquer d'erreur à la lecture.

L'exécution du programme requiert uniquement la commande "main.py", puis l'utilisation du menu pour naviguer entre les différentes fonctionnalités :

- Approximer les probabilités d'un réseau bayésien binaire
- Calculer l'erreur moyenne et le temps moyen d'exécution

La première partie sera utilisée pour analyser les performances des différents algorithmes sur le réseau étudié en cours, celui du tremblement de terre, en choisissant les paramètres souhaités : algorithme de son choix et nombre d'échantillons. Cette fonction affiche ensuite l'erreur obtenue et le temps d'exécution que cela a pris pour pouvoir comparer les différents algorithmes.

La seconde fonctionnalité permet d'appliquer l'ensemble des algorithmes sur différentes tailles d'échantillons et sur plusieurs réseaux pour faire des moyennes d'erreur et de temps d'exécution. Il y a un choix à faire dans ce menu pour décider si on compare les trois algorithmes ou seulement le weighted sampling et gibbs, en effet, comme nous l'expliquerons plus tard, le temps d'exécution de l'algorithme de rejet augmente très rapidement quand on augmente le nombre de d'échantillon. C'est pourquoi, après plusieurs tentatives infructueuses, nous avons choisi de ne plus lancer d'algorithme de rejet sur 10e6 échantillons.

La comparaison des trois algorithmes s'effectue donc sur des échantillons allant de 1000 à 10000 par tranche de 1000.

La comparaison sans la méthode de rejet s'effectue sur des échantillons allant de 10000 à 100000 par tranche de 10000.

Dans tous les cas, le programme permet de suivre la progression pour savoir où en est la génération des échantillons. Une fois fini, les graphes sont générés et stockés dans le dossier graphe.

## II. Performance des algorithmes sur le réseau du tremblement de terre

Les données pour le temps sont ici exprimées en seconde.

Méthode du rejet :

```
Nombre d'échantillon : 1000  
Erreur : 0.002107290288440089  
Temps : 17.111189603805542
```

```
Nombre d'échantillon : 10000  
Erreur : 0.0020072902884400445  
Temps : 160.75473976135254
```

```
Nombre d'échantillon : 100000  
Erreur : 0.00015729028844008175  
Temps : 1847.320050239563
```

Weighted sampling :

```
Nombre d'échantillon : 1000  
Erreur : 0.23305583513993322  
Temps : 0.0329587459564209
```

```
Nombre d'échantillon : 10000  
Erreur : 0.06700100675031115  
Temps : 0.40990543365478516
```

```
Nombre d'échantillon : 100000  
Erreur : 0.027274443231533385  
Temps : 3.394434928894043
```

```
Nombre d'échantillon : 1000000  
Erreur : 0.0036397082528414337  
Temps : 40.30885863304138
```

Gibbs :

```
Nombre d'échantillon : 1000  
Erreur : 0.01789270971155993  
Temps : 0.08776402473449707
```

```
Nombre d'échantillon : 10000  
Erreur : 0.001407290288440055  
Temps : 0.8916521072387695
```

```
Nombre d'échantillon : 100000  
Erreur : 0.001542709711559953  
Temps : 8.827605247497559
```

```
Nombre d'échantillon : 1000000  
Erreur : 0.0013092902884400681  
Temps : 134.09118008613586
```

Comme dit lors de l'introduction, nos tests ont révélé que le temps d'exécution de la méthode de rejet augmentait drastiquement lorsque le nombre d'échantillons augmente, ce qui ne nous a pas permis de pousser les tests de cette méthode jusqu'à des approximations avec  $10^6$  d'échantillons. Cependant, on constate que l'erreur diminue également beaucoup lorsqu'on génère plus d'échantillons.

En comparaison, les temps de la méthode du likelihood weighting sont relativement bas, l'exécution sur  $10^6$  échantillons ne prend que deux fois plus de temps que pour  $10^3$  échantillons avec la méthode de rejet. En contrepartie, l'approximation est bien plus performante pour la méthode du rejet qui, dès 1.000 ou 10.000 échantillons, permet d'obtenir une erreur plus faible que celle obtenue par la méthode du weighted sampling avec  $10^6$  échantillons.

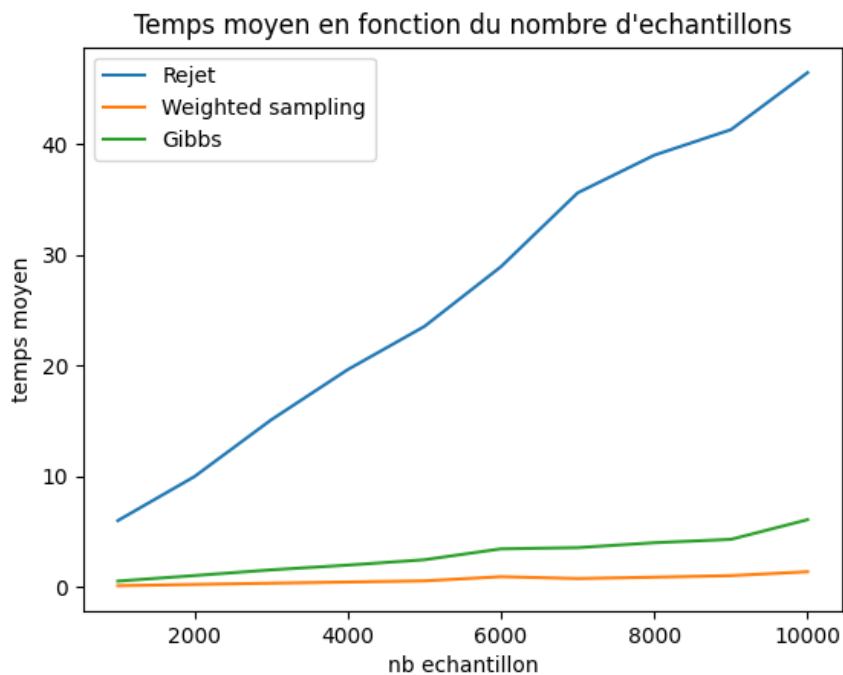
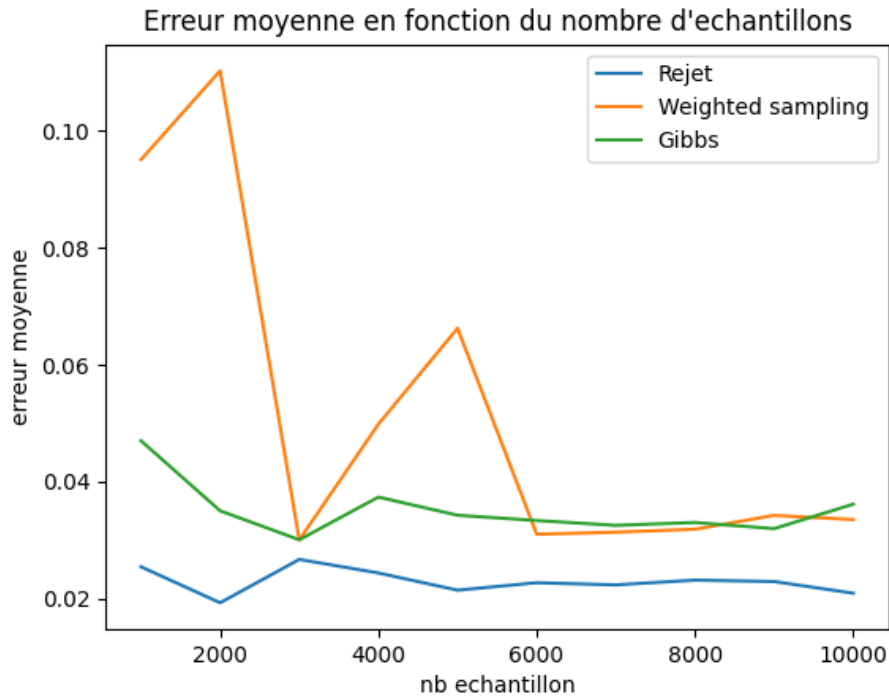
On remarque même que les meilleures approximations ont été obtenues par la méthode du rejet. Par exemple, l'évaluation des probabilités sur  $10^5$  échantillons a donné une erreur proche de  $10^{-4}$ , qui est une très bonne approximation, malgré un temps d'exécution bien plus élevé que les autres méthodes.

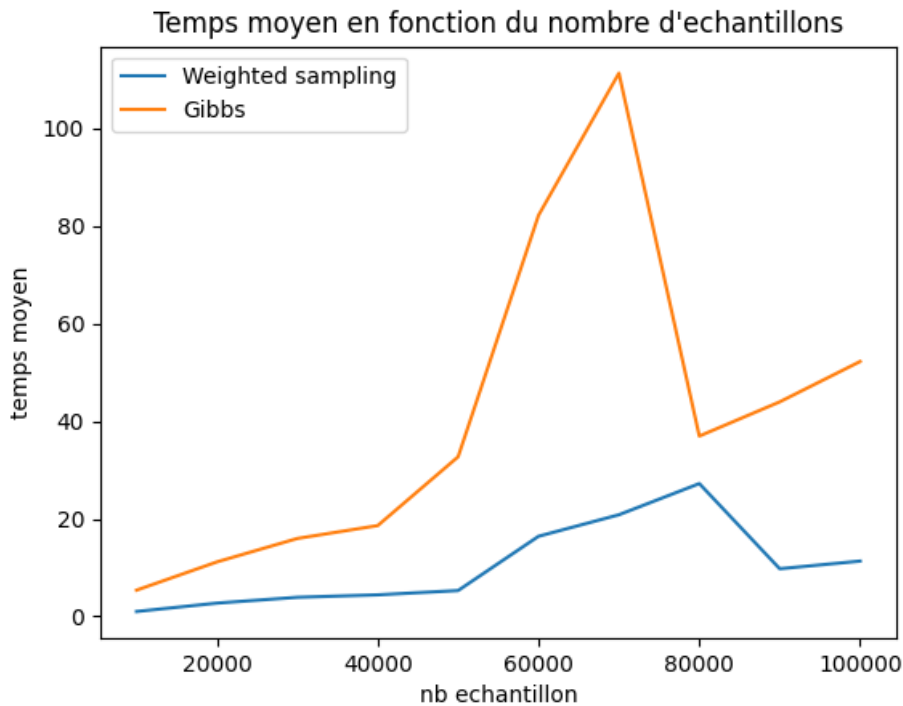
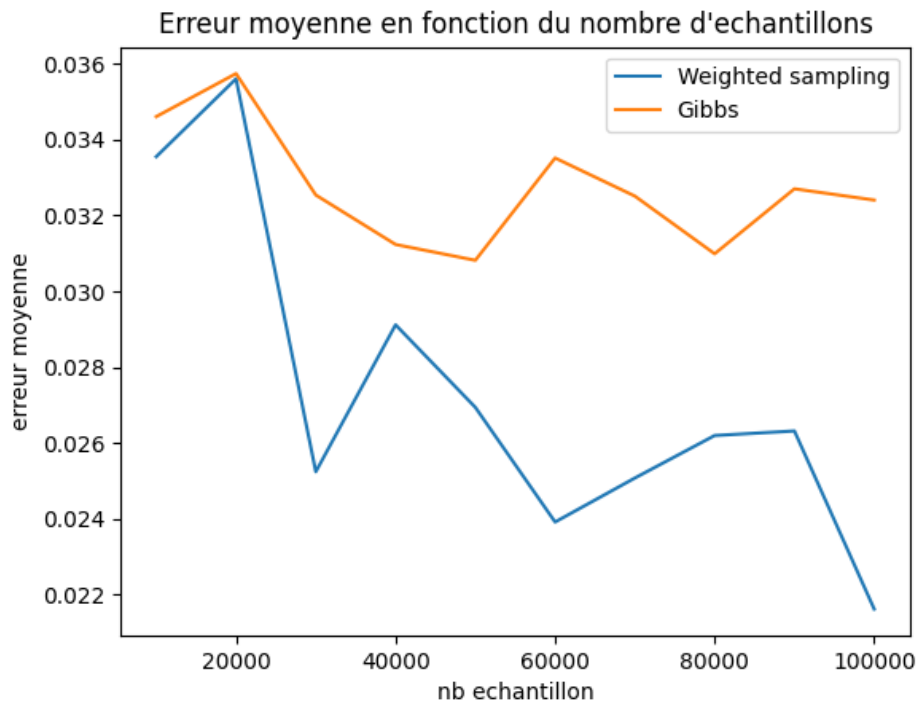
On constate que pour Gibbs, l'erreur est très faible dès  $10^4$  échantillons, ajouter davantage d'échantillons n'améliore pas réellement les performances de l'algorithme puisque l'erreur ne réduit pas vraiment et le temps d'exécution augmente énormément.

Ainsi, sur ce premier réseau, l'algorithme du rejet est celui permettant d'obtenir l'erreur la plus faible, au prix d'un temps de calcul bien plus long, ensuite la méthode de gibbs permet également de très bon résultat dès 10000 échantillons. Enfin, la méthode du likelihood sampling offre une approximation relativement moins précise que les deux autres bien qu'elle soit vraiment plus rapide que les deux autres.

### III. Comparaison des performances des algorithmes sur l'ensemble de la base de données

L'ensemble des graphiques n'ont pas été réalisés sur les mêmes ordinateurs et ne sont donc pas comparable entre eux car la puissance de calcul n'est pas la même sur les deux ordinateurs.





Comme explicité dans la partie précédente, on remarque que l'algorithme de rejet est bien plus long à exécuter que les autres méthodes, ce qui est logique puisqu'une partie des échantillons générés n'est pas gardée et qu'il faut donc en créer des nouveaux. L'algorithme de Gibbs est également plus long que l'algorithme du weighted sampling puisqu'il doit

recalculer les proba conditionnel à chaque fois en fonction de ce que sa couverture de markov a tiré comme valeur. En termes de temps, c'est donc l'algorithme du likelihood sampling qui permet d'estimer le plus rapidement des probabilités. Il reste rapide même sur des grands nombre d'échantillons comme  $10^5$ , ou  $10^6$  comme le montre la partie précédente.

Cependant, pour les erreurs, on constate encore une fois que la méthode de rejet offre des performances supérieures aux deux autres méthodes car elle donne l'erreur minimale pour des échantillons de petites tailles. On remarque également que les méthodes de rejet et de Gibbs sont plus stables que celle du weighted sampling puisque, et ceux même sur des tailles importantes, ce dernier algorithme montre des erreurs moyennes très dispersées. Malgré tout, la méthode du likelihood sampling est celle qui s'améliore le plus lorsqu'on augmente le nombre d'itérations de l'algorithme, en effet, jusqu'à 10000 échantillons, les performances sont quasiment triplés et jusqu'à 100000, elles sont encore pratiquement doublé, tandis que les autres méthodes ne s'améliore pas autant lorsqu'on augmente le nombre d'échantillon.

## IV. Conclusion

La partie ayant généré le plus de problèmes dans son implémentation a été la méthode de Gibbs. Les résultats étaient assez éloignés des résultats réels sans pour autant être complètement incohérents et ce, même sur des générations de  $10^5$  échantillons. Nous avons mis du temps à comprendre ce que nous avions mal codé pour avoir de tels résultats.

Une des limites auxquelles nous avons dû faire face est le temps d'exécution, nos ordinateurs n'ayant pas des capacités très importantes, chaque simulation demandait de tourner parfois pendant plusieurs heures, et nous empêchait de tester sur d'avantage d'échantillons, comme ce fut le cas pour l'algorithme du rejet. Nous n'avons pas pu le tester sur des échantillons aussi grands que les deux autres méthodes puisqu'il demandait déjà une trentaine de minutes pour tourner sur  $10^5$  échantillons.

La lecture de fichier a été conçue pour lire des réseaux bayésiens non binaires, cependant, par manque de temps, les algorithmes n'ont pas pu être adaptés pour travailler sur ce type de réseaux.

De plus, la génération n'a également pas pu être implémentée par manque de temps, nos difficultés avec la méthode de Gibbs ayant demandé beaucoup d'investissement.