## Introduction

Le MBTI est un questionnaire qui permet de classifier des individus selon quatre dimensions binaires. Chaque dimension représente deux préférences opposées tel qu’un individu n’en possède qu’une seule en même temps. Le MBTI contient plus d’une centaine de questions. Le but de ce projet est de proposer une solution en utilisant des réseaux bayésiens pour minimiser le nombre de questions posées.

## Réseaux Bayésiens

Un réseau bayésien modélise les relations entre les différentes variables d’un système. Ces variables peuvent avoir plusieurs états qui apparaissent plus ou moins souvent que d’autre. Un réseau bayésien est un graph acyclique où chaque nœud représente une variable et chaque arc représente la dépendance conditionnelle entre deux variables. Les réseaux bayésiens permettent donc de modeler les relations conditionnelles entre différentes variables d’un système. Un réseau bayésien aléatoire (RBA) est un ensemble de réseaux bayésiens qui sont construits aléatoirement. Les RBA sont utilisé pour avoir des résultats plus uniformes lors de l’utilisation de réseaux bayésiens. En effet, les réseaux bayésiens sont des modèles biaisé qui ne représente que certain aspect d’un système. Un RBA contient plusieurs variations de réseaux bayésiens et représente donc mieux le modèle original.

Dans mon approche, la création d’un réseau bayésien est définit par 5 paramètres :

|  |  |
| --- | --- |
| Nom | Description |
| Dimensions | Nombre d’attributs dans le modèle |
| Minimum par niveau | Le nombre minimum de nœuds par niveau |
| Maximum par niveau | Le nombre maximum de nœuds par niveau |
| Minimum de parents | Le nombre minimum de parents par nœud |
| Maximum de parents | Le nombre maximum de parents par nœud |

L’algorithme consiste à créé le réseau niveau par niveau. Chaque niveau contient un ensemble de variables choisis où chacune est représentée par un seul nœud. L’ordonnément des niveaux et la sélection de leurs nœuds est complétement aléatoire. Dans le cadre de ce projet, chaque question du MBTI est une variable aléatoire. La création du réseau est aléatoire à cause du manque de connaissances concrètes sur les différentes variables. En effet, dans le MBTI il est difficile de dire quelles questions ont une influence sur d’autres questions. Vu que la création du réseau est aléatoire, il est facile de généré un RBA; celui-ci est un ensemble de réseaux bayésiens aléatoires.

La création d’un réseau bayésien n’est qu’une partie du problème posé pour ce projet, l’utilisation du réseau s’avère être une tâche complexe avec une multitude de possibilité. Dans mon cas, le RBA est utilisé pour minimiser la quantité de questions à répondre sur le MBTI. En effet, lorsqu’un utilisateur a déjà répondu à quelques questions, le RBA peut deviner les réponses à d’autres questions et, dans certains cas, peut même trouver directement le résultat du MBTI. Dans la prochaine section, je présente deux expériences liées à cette tâche.

## Expériences

Dans cette section, je présente deux expériences qui permettent d’évaluer les performances de mon RBA. Le but ultime de ce projet et de présenter une solution au problème de minimisation de questions, mais l’évaluation de cette solution doit être faite avec une tache différente. En effet, un RBA peut être utilisées de maintes façons et les deux expériences présentées utilisent le même processus de construction pour le RBA mais divergent dans leurs utilisations.

La première expérience consiste à évaluer les performances du réseau bayésien à compléter les questions non répondu du MBTI. Les données sont divisées aléatoirement en deux sous-ensembles; les données d’entrainement et celle de teste. Le RBA est construit avec les données d’entrainement. Chaque vecteur de réponse du sous-ensemble de teste est aléatoirement modifié; une portion du vecteur est enlevée, cette portion est un paramètre appelé taux de mutation. Dans cette expérience, le RBA est utilisé pour deviner la valeur originale des données enlevées. La tâche de prédiction est un succès si la confiance de celle-ci est au-dessus d’un certain nombre appelé « threshold ». Les résultats suivant sont présenté en termes de précision (« accuracy »), de couverture (« coverage ») et du nombre moyen de question bien deviné par individu (gain).

Les deux tableaux précédents montrent l’influence du taux de mutation et du « threshold » de confiance sur les mesures de performances décrites. On peut constater que dans le meilleur des cas, le système est capable de deviner 70% des questions avec une précision de 90% et plus et ce sur plus de 90% des données de teste.

La seconde expérience consiste à calculer le nombre minimal de questions nécessaire pour trouver le résultat du MBTI pour un individu. Comme pour l’expérience précédente, les données sont divisées aléatoirement et le RBA est construit de la même façon. Pour ce faire, il suffit de prendre un vecteur des données de teste et d’enlever une question à la fois tout en gardant une haute confiance du résultat. Il s’agit donc d’un problème de minimisation où on commence avec un vecteur avec le nombre maximal de dimensions et on va essayer d’enlever le plus possible de dimension sans affecter le score total du vecteur. Le nombre de questions enlevées est la mesure principale de performance (gain), la perte moyene de confiance dans le résultat et la couverture de l’approche sont aussi considérée.

Les trois tableaux précédents présentent les résultats pour trois des quatre types du MBTI. Pour chacune des verticales, le RBA est entraîné avec seulement les questions propres à cette verticale. On peut voir que la perte moyenne est minimale (< 0.1%), que la couverture est relativement basse (entre 34% et 50%) mais que le gain est très haut (>88%). Cela signifie que pour un individu aléatoire il y a de 30% à 50% de probabilité qu’on puisse deviner son type sans 88% et plus de questions. Donc avec seulement 12% des questions et moins il est possible de deviner le type du MBTI d’une personne. Le fait que la couverture de cette approche est basse est un problème qui peut être réglé en ajustant la création du RBA et en réduisant la confiance minimale requise lors de l’expérience.

## Le code

Le code est divisé en plusieurs « packages » et ceux-ci sont décrit dans le tableau suivant.

|  |  |
| --- | --- |
| Builder | Contient les algorithmes propres à la création des réseaux bayésiens |
| Controllers | Contient les algorithmes de teste et d’évaluation |
| Evaluation | Contient des classes qui aident aux tâches d’évaluations |
| Minimization | Contient les algorithmes qui servent à réduire le nombre de dimensions |
| Network | Contient toutes les structures propres aux réseaux bayésiens |
| Utilities | Contient toute autre classe |