

INFO-F-311: Intelligence Artificielle

Projet 3: Réseaux Bayésiens

2024-2025

Axel ABELS

Tom LENAERTS

Yannick MOLINGHEN

Pascal TRIBEL

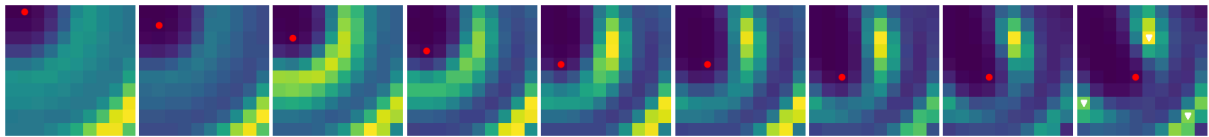


Figure 1. – Exemple d'évolution des croyances sur la position des gemmes. Le point rouge représente l'agent, et les triangles blancs représentent les trois gemmes cachées. Au plus la case est claire, au plus l'agent croit qu'une gemme est susceptible de s'y trouver.

1. Préambule

Dans ce projet, vous allez implémenter des techniques d'intelligence artificielle basées sur l'inférence statistique dans des réseaux bayésiens. Des fichiers de base sont fournis sur l'université virtuelle pour vous aider à démarrer.

1.1. Mise en place

Trois fichiers sont fournis : `visualisation.py`, `bayesian_network.py`, et `main.py`. Ils contiennent des fonctions pré-implémentées et des squelettes de code à compléter.

1.1.1. Dépendances

Installez les dépendances nécessaires avec `pip`, en particulier :

- `numpy`
- `matplotlib`

2. Localisation des gemmes

L'environnement est similaire à celui que vous avez utilisé dans les deux précédents projets, `l1e`, mais nous allons nous concentrer sur un problème plus spécifique. Vous disposez d'un agent évoluant dans une grille $n \times n$, initialement placé en (x_0, y_0) , qui utilise un *sonar* pour mesurer la distance à m gemmes. Pour modéliser plus fidèlement la réalité, les données mesurées par ce *sonar* sont bruitées par une normale $N(0, \frac{1}{2})$, de moyenne 0 et d'écart-type $\frac{1}{2}$. Le nombre de gemmes m est connu d'avance. Votre rôle est d'implémenter un réseau de Bayes pour déterminer, après chaque mouvement, des croyances sur la position de chacune des gemmes.

2.1. Réseau Bayésien

Le réseau de Bayes est constitué de deux types de nœuds :

- G_m , qui représentent les positions des gemmes et peuvent prendre toutes les coordonnées possibles dans la grille,
- $D_{i,j}$, qui décrivent les vecteurs de distances entre les coordonnées (i, j) et les gemmes.

G_i sont les racines du réseau, pointant vers les nœuds $D_{i,j}$. Un exemple avec une grille 3×3 et 2 gemmes est donné dans la Figure 2.

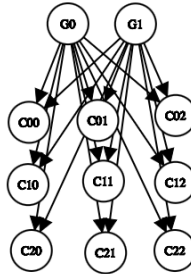


Figure 2. – Exemple de réseau de Bayes pour une grille 3x3 et 2 gemmes

2.1.1. Observation

Une observation à (i, j) est un vecteur de distances entre (i, j) et les gemmes, avec un bruit normal $N(0, \frac{1}{2})$.

2.1.2. Inférence

Soient $o_{i,j}$ le vecteur observation obtenu aux coordonnées (i, j) . Vous voulez décrire les nouvelles valeurs de G en déterminant pour quelles coordonnées de chaque gemme le *sonar* est le plus vraisemblable de vous avoir donné $o_{i,j}$. Pour ce faire, vous utilisez une fonction de vraisemblance définie dans la section suivante. On vous demande d'implémenter une méthode d'inférence par énumération, qui permet de mettre à jour le réseau bayésien lorsqu'une observation est réalisée. Dans le cadre de ce projet, il n'est nécessaire de mettre à jour que les nœuds G .

La Figure 1 montre un exemple d'évolution des valeurs de croyances sur G au fur et à mesure que des coups sont joués. Dans cette figure, le point rouge représente l'agent, et les triangles blancs représentent les gemmes.

3. Consignes

3.1. Implémentations et expériences

- Implémentez la fonction de vraisemblance `likelihood`: soit d la distance euclidienne entre le vecteur d'observations et le vecteur qui aurait été observé si les gemmes se trouvaient en `gem_positions` (voir `bayesian_network.py`), alors la vraisemblance est donnée par 2^{-d} . Justifiez cette formulation dans le rapport. **(1 point)**
- Implémentez une stratégie d'inférence par énumération pour mettre à jour les nœuds G du réseau bayésien après une observation. **(2 point)**
- Analysez la complexité de la méthode d'inférence en fonction de la taille de la grille n et du nombre de gemmes m , et vérifiez cette complexité expérimentalement en représentant le temps d'exécution pour différentes valeurs de n et m sur un graphique. **(2 points)**
- Proposez deux améliorations à la méthode d'inférence, qui interviennent sur différentes étapes du processus inférentiel. **(1 point)**

3.2. Rapport

Rédigez un rapport (en utilisant [Typst](#) ou [Latex](#)) présentant le cadre expérimental et les points demandés dans la section précédente avec leurs analyses et justifications respectives, en suivant la structure habituelle d'un rapport scientifique. Spécifiez aussi comment votre méthode d'inférence fonctionne, et développez-en les fondements probabilistes. Ce rapport compte pour **4 points** et une attention particulière sera portée à la mise en forme et la présentation générale des résultats.

4. Remise

Le livrable de ce projet se présente sous la forme d'un fichier zip contenant les sources Python du projet ainsi que votre rapport au format PDF. Le travail est **individuel** et doit être rendu sur l'Université Virtuelle pour le 10/11/2024 à 23:59. Une séance de questions-réponses sera organisée dans la semaine précédant la remise, si suffisamment de questions sont posées. N'hésitez pas à utiliser le forum de la page UV. Vous pouvez envoyer vos questions à l'adresse pascal.tribel@ulb.be