Projet -- Apprendre la structure d'un réseau bayésien

Dans le cours et les TDs, on a travaillé avec des réseaux bayésiens qui nous étaient donnés. Le but de ce projet est donc de vous faire réfléchir sur la fabrication d'un réseau bayésien. Ce projet s'inspire d'une partie d'un projet donné dans une université américaine [1].

Contexte

Un réseau bayésien est souvent construit avec l'aide d'un expert et d'une base de données. L'expert (qui parfois est simplement le décideur) peut avoir des intuitions sur les indépendances, ce qui est une cause, ce qui est plutôt une conséquence. Parfois, des experts peuvent ne pas s'entendre à ce sujet. Si un expert donne la structure du réseau (i.e. il donne le graphe orienté acyclique), il n'est pas difficile de calculer les tables de probabilités conditionnelles.

Si on n'a pas d'expert pour nous aider à construire le graphe, peut-on chercher un bon graphe avec les données ? On peut dire que l'on veut *apprendre* la structure d'un réseau bayésien. Malheureusement, c'est un problème difficile (c'est un autre problème NP-complet[1, 2]), c'est un sujet de recherche et on trouve d'ailleurs plusieurs thèses récentes sur le sujet.

La structure du graphe est à trouver parmi tous les graphes acycliques orientés (DAG pour *directed acyclic graphs*). Malheureusement, l'espace des DAG est (très) large, donc on ne pourra pas énuméré tous les graphes. On pourra donc utiliser des techniques d'IA (recherche locale) pour chercher un bon graphe.

Evidemment, pour faire cette recherche, on a besoin d'une mesure de la qualité de notre graphe. C'est ce qu'on va expliquer dans la section suivante.

Score

Supposons qu'on ait un DAG G et un ensemble de données D. Une idée pour former un score est d'estimer la probabilité que ces données aient pu être générées par la distribution de probabilité qui est encodée par le réseau bayésien qui a pour structure G. On veut donc estimer $\mathbb{P}(G \mid D)$.

Supposons que nos variables soient $X_1, \ldots, X_i, \ldots, X_n$. La variable X_i peut prendre r_i valeurs discrètes. Dans notre graphe G, on va noter q_i le nombre de façons d'instantier les variables parentes de X_i. On va noter π_{ij} la $j^{\text{ième}}$ instantiation des variables parentes de X_i.

Exemple : Si toutes les variables sont binaires, alors $r_i = 2$ pour tous les i. Si X_i possède k parents, les variables parentes de X_i ont $q_i = 2^k$ instantiations. Le nombre de lignes de la table de probabilité jointe de X_i est donc $q_i r_i$.

On va noter θ_{ijk} la valeur se trouvant dans les tables de probabilités conditionnelle pour la $j^{\text{ième}}$ instantiation des parents de X_i, i.e., pour $k \leq r_i$, $\mathbb{P}(X_i = k \mid \pi_{ij}) = \theta_{ijk}$. Evidemment, toutes ces valeurs ne sont pas indépendantes, on a besoin de fixer seulement $(r_i - 1)q_i$ valeurs. Avec G et tous les θ_{ijk}, on a un réseau bayésien complet.

1. pour ne pas influencer votre travail, je ne la cite pas volontairement ici. Evidemment, vous pourrez trouver ce projet, mais je pense que ça ne vous avancerait pas beaucoup.

On note m_{ijk} le nombre de fois où dans nos données D, X_i prend la valeur k et où les parents de X_i prennent la valeur π_{ij}. Il peut être montré que sous certaines hypothèses, les $\mathbb{P}(\theta_{ijk})$ suivent une distribution de Dirichlet $D(a_{ij1}, \ldots, a_{ijr_i})$. Les valeurs des a_{ijk} peuvent permettre de biaiser initialement les probabilités θ_{ijk}. Comme nous n'avons pas de justification pour le faire, on va considérer que chaque $a_{ijk} = 1$. Ainsi, si la valeur $1 \leq k \leq r_i$ de π_{ij} apparait m_{ijk} fois, la distribution des θ_{ij} est mise à jour et suit la loi $Dir(1+m_{ij1}, \ldots, 1+m_{ikr_i}$. Dans [3, 4], il est montré que

$$\mathbb{P}(G \mid D) = \mathbb{P}(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(a_{ij0})}{\Gamma(a_{ij0} + m_{ij0})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk} + m_{ijk})}{\Gamma(a_{ijk})}$$

où $a_{ij0} = \sum_{k=1}^{r_i} a_{ijk}$ et $m_{ij0} = \sum_{k=1}^{r_i} m_{ijk}$ et où Γ est la fonction Gamma.

Pour des questions de facilités de calculs, on va préférer passer au logarithme (on préfère sommer des logarithmes de petits réels plutôt que de prendre le produit de petits réels). Aussi, plusieurs bibliothèques de calculs ont une implémentation efficace du logarithme de la fonction Γ. Le *score bayesien* est ce que l'on peut chercher à optimiser dans notre recherche :

$$\log \mathbb{P}(G \mid D) = \log \mathbb{P}(G) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\log\left(\frac{\Gamma(a_{ij0})}{\Gamma(a_{ij0} + m_{ij0})} \right) + \sum_{k=1}^{r_i} \log\left(\frac{\Gamma(a_{ijk} + m_{ijk})}{\Gamma(a_{ijk})} \right) \right]$$

Sans faire d'hypothèse particulière, on n'a pas de raison qu'un graphe soit plus probable qu'un autre. Donc le graphe qui maximisera le score maximesera également

$$score(G \mid D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\log\left(\frac{\Gamma(a_{ij0})}{\Gamma(a_{ij0} + m_{ij0})} \right) + \sum_{k=1}^{r_i} \log\left(\frac{\Gamma(a_{ijk} + m_{ijk})}{\Gamma(a_{ijk})} \right) \right]$$

En choisissant $a_{ijk} = 1$, on obtient

$$score(G \mid D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\log\left(\frac{\Gamma(r_i)}{\Gamma(r_i + m_{ij0})} \right) + \sum_{k=1}^{r_i} \log\left(\Gamma(1 + m_{ijk}) \right) \right]$$

Données

On va utiliser deux bases de données connues (légèrement simplifiées) et votre but et de proposer un réseau bayésien qui représente chacune des bases de données.
"titanic" : Les données représentent les données de 887 passagers, chaque ligne représentant un passager. Les données contiennent des informations comme l'âge, le prix de leur billet, la classe du billet, si le passager avait des enfants/parents à bord, et si la personne a survécu ou non. On a discrétisé certains attributs pour avoir des variables discrètes avec peu de valeurs. Par rapport aux données du projet kaggle, on a aussi enlevé des attributs comme le numéro du billet, le numéro de cabine qui seraient plus durs à discrétiser.

"wine" : c'est un dataset classique du UCI machine learning repository. Les attributs sont des caractéristiques de vins portugais. Lorsqu'on utilise ce dataset pour faire de la classification, on cherche à apprendre l'attribut qualité. Comme pour le dataset précédent, les données originales sont numériques, et les données que l'on vous donne sont discrétisées en plusieurs classes.

Travail à réaliser

— Pour les deux bases de données, estimez le nombre de DAG qui sont possibles.
— Proposez un réseau bayésien qui représentent chaque base de données.
Le but ici n'est pas nécessairement d'avoir un réseau bayésien "optimal". Votre but est de proposer la structure d'un réseau bayésien qui vous parait bonne, et vous devez justifier votre choix. Plusieurs réseaux bayésiens sont capables de représenter la même probabilité jointe complète.
Expliquez comment vous êtes parvenus à cette structure et si vous pouvez généraliser facilement votre méthode à d'autres données.
— Etudiez/discutez vos résultats. Est-ce que les dépendances conditionnelles sont vérifiées dans les données ? Est-ce que la direction des arcs vous semblent refléter la causalité ? On ne vous demande pas nécessairement d'être exhaustif ici, mais exercez un sens critique à vos résultats.
— Vous avez un réseau bayésien pour chaque dataset. Intéressons nous maintenant au problème de classification où il faut apprendre les attributs `survived` pour le dataset `titanic` et `quality` pour le dataset `wine`.
Avec le réseau bayésien, vous êtes maintenant capable de calculer une requête du type $\mathbb{P}(\mathit{Quality} \mid \mathbf{o})$ où $\mathbf{o}$ est une nouvelle donnée (i.e. chaque attribut possède une valeur, évidemment l'attribut `quality` est absent de $\mathbf{o}$!).
Comparez avec la méthode de la classification naïve bayésienne. Est-ce que pour chacun de vos deux réseaux la classification naïve bayésienne obtiendrait de bons résultats ? Est-ce que l'hypothèse naïve convient bien ou pas ?
Si vous le souhaitez, vous pouvez former un réseau bayésien et calculer et classifieur naif bayésien sur un ensemble de données d'entrainement, et comparer les résultats sur un ensemble de test.
— Le travail sera synthétisé dans un rapport soumis en format pdf seulement. Essayez d'être précis et synthétique. Si vous en sentez le besoin, vous pouvez mettre du texte ou des résultats dans une section annexe pour aider le lecteur.

Contraintes & Deadlines

— Vous devrez soumettre votre travail pour *le mercredi 12 avril 23 :59* via la plateforme `myCourse`. Pour ce faire, un seul membre du binôme doit soumettre le projet. Les soutenances seront organisées la semaine du 17 avril (environ 15min de soutenance par groupe, vous n'avez rien à préparer, la soutenance sera une discussion autour de vos résultats).
— Lors de la soumission, vous devez soumettre une seule archive au format ZIP. Le nom de l'archive doit être composé des noms des auteurs (si les auteurs du projet sont Margaret Hamilton et Grace Hopper, le nom du l'archive sera quelque chose comme `HamiltonHopper.zip`). Quand on ouvre l'archive, on aura un répertoire `HamiltonHopper` qui contiendra quatre choses :
  — *un rapport* : le *seul* format accepté pour votre rapport est le format PDF. Le rapport sera au plus de 10 pages, et ne devra pas contenir de code. Vous pouvez ajouter une annexe (donc dépasser les 10 pages), mais l'évaluateur ne sera pas forcé de la consulter. Vous pouvez supposer que le lecteur connait le cours (donc vous n'avez pas à expliquer des notions vues en cours).
  — Dans un répertoir `src`, on trouvera tout le code source utilisé pour réaliser ce projet. Il sera traité comme les annexes du rapport (i.e. l'évaluateur n'est pas

obligé de le consulter, mais il le peut).
- deux fichiers qui contiennent la structure de chacun de vos réseaux bayésiens : `wine.gph` et `titanic.gph`. Le fichier texte correspondant représente la structure de votre réseau, chaque ligne représente une arête. Une arête est un couple d'attributs séparés par une virgule. L'arête va du premier attribut au second attribut. Par exemple la ligne suivante représente une arête `ph` → `alcohol` dans le réseau bayésien.
  `ph, alcohol`
  Le nom des attributs est celui présent dans les données.

— vous pouvez utiliser au choix python ou java pour programmer votre solution.
— vous pouvez utiliser des bibliothèques pour manipuler ou représenter des graphes ou faire des calculs, mais vous ne pouvez pas utiliser de bibliothèques qui manipulent des réseaux bayésiens !
— Respectez le travail d'autrui : citez tout site web, tout ouvrage et tout article que vous aurez utilisé pour réaliser ce travail.
— Le projet s'effectue en binôme (un binôme peut être constitué d'étudiants venant de différents groupes de TD). Vous pouvez parler de stratégie de recherche ou d'évaluation avec vos camarades, mais vous n'avez pas le droit de montrer ou partager la structure de votre réseau, votre rapport, ou du code. Ce travail fait partie de votre évaluation, c'est donc à vous de le réaliser !

Références

[1] David Maxwell Chickering. *Learning Bayesian Networks is NP-Complete*, pages 121–130. Springer-Verlag, learning from data : artificial intelligence and statistics v edition, January 1996.

[2] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of bayesian networks is NP-hard. *J. Mach. Learn. Res.*, 5 :1287–1330, dec 2004.

[3] Gregory F. Cooper and Edward H. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9 :309–347, 1992.

[4] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks : The combination of knowledge and statistical data. *Mach. Learn.*, 20(3) :197–243, sep 1995.