

Projet - Inférence approchée dans les réseaux bayésiens

Le but de ce projet est d'évaluer les trois algorithmes d'inférence approchée vus en cours : la méthode des rejets, la méthode "likelihood weighting" et la méthode de Gibbs. Comme ces trois méthodes sont correctes, elles convergent vers le même résultat quand le nombre d'échantillons approche l'infini. Il existe des résultats théoriques sur la qualité de l'approximation. Le but du projet est de se donner une idée en pratique sur des exemples connus, et sur des réseaux générés aléatoirement.

Par exemple, dans le classique "AI : A modern approach"[1] de Russel et Norvig, on trouve les deux figures ci-dessous (Figure 1) qui représentent l'erreur en fonction du nombre d'échantillons sur des requêtes effectuées sur l'exemple du réseau sur l'assurance. Notre but est produire des résultats similaires pour analyser les performances de ces algorithmes.

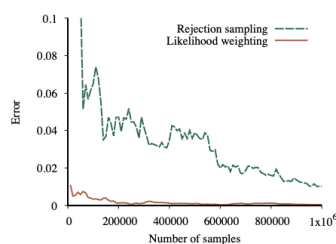


Figure 13.19 Performance of rejection sampling and likelihood weighting on the insurance network. The x-axis shows the number of samples generated and the y-axis shows the maximum absolute error in any of the probability values for a query on *PropertyCost*.

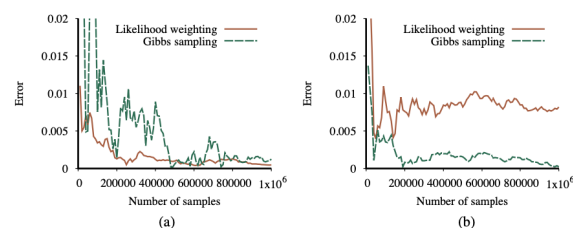


Figure 13.22 Performance of Gibbs sampling compared to likelihood weighting on the car insurance network: (a) for the standard query on *PropertyCost*, and (b) for the case where the output variables are observed and *Age* is the query variable.

FIGURE 1 – Performances reportées dans [1], chapitre 13

Travail à réaliser

On note V l'ensemble des variables d'un réseau bayésien. Dans ce projet, vous pouvez vous limiter aux réseaux bayésiens où toutes les variables sont binaires. Implémentez les trois méthodes d'inférence approchée vue en cours. On suppose que les requêtes sont de la forme $\mathbb{P}(Q \mid X_1 = v_1, \dots, X_k = v_k)$ où

- $Q \in V$: Q est une variable du réseau bayésien.
- $k \geq 0$: on n'a pas forcément d'observation.
- $\{X_1, \dots, X_k\} \subseteq V \setminus \{Q\}$: on peut avoir des observations pour les variables du réseau bayésien autre que la variable Q .
- les valeurs v_i peuvent prendre les valeurs vrai ou faux

Implémentez les trois méthodes d'inférence approchée : la méthode des rejets, la méthode "likelihood weighting" et la méthode de Gibbs.

Utilisez votre implémentation sur la petite base de données fournie. La base de donnée vous donne un certain nombre de réseaux bayésiens au format `.bif`. L'exemple classique du cours est donné au format `.bif` en fin de ce document. La description d'un réseau commence par le mot clé `network` suivi d'un nom du réseau. Viennent ensuite la liste de variables (dans notre cas, elles sont binaires, donc elles auront toutes la même forme). Ensuite, viennent la description des tables de probabilités conditionnelles. L'exemple est suffisamment clair et je ne détaillerai pas plus le langage de description des réseaux bayésiens. La base de données vous donnera également une requête par réseau. La requête sera écrite sous la forme suivante :

$$\text{Prob}(\text{VarName} \mid \text{VarName1}=v_1, \text{VarName2}=v_2, \dots, \text{VarNamek}=v_k) = [p_{\text{false}}, p_{\text{true}}]$$

où

- Prob est un mot clé,

- `VarNamej` est le nom d'une variable du réseau bayésien,
- `vj` est `true` ou `false`
- p_{false} et p_{true} sont les valeurs calculés avec une méthode exacte.

L'exemple de la requête du TD 6 est donnée après la description du réseau en fin de document.

Vous devez rédiger un rapport dans lequel vous devrez comparer les performances des algorithmes, tant sur le plan de la complexité algorithmique (temps de calculs), que sur la qualité de l'approximation. Dans le rapport, vous n'avez pas à expliquer de nouveau les algorithmes d'inférence. Cependant, vous devrez bien décrire et analyser vos résultats. On vous demande d'écrire une section sur la performance des algorithmes sur l'exemple classique du cours sur le tremblement de terre. On vous demande ensuite dans une autre section de discuter les résultats agrégés sur toutes les requêtes de la base de données. Enfin, on vous demande d'avoir une section de conclusion où vous indiquerez entre autres les difficultés que vous avez rencontrées, les limites de l'étude que vous avez réalisé, et ce que vous n'avez pas eu le temps de faire.

Bonus

Génération

Vous pouvez générer vos propres réseaux bayésiens pour réaliser plus de tests. Pour effectuer ce bonus, vous devrez décrire dans votre rapport comment vous générez les réseaux bayésiens et si votre méthode implique des contraintes sur le type de réseaux bayésiens générés. Vous devrez évidemment utiliser des réseaux générés par vos soins pour comparer et évaluer les algorithmes d'inférence approchée et reporter les résultats et leur analyse dans le rapport.

Généralisation

Vous pouvez implémenter les algorithmes pour des réseaux bayésiens où les variables ne sont pas nécessairement binaires. Vous pouvez alors tester vos algorithmes sur des exemples fréquemment utilisés dans la littérature (en particulier, vous devriez pouvoir vérifier les figures ci-dessus pour le domaine de l'assurance). Vous pourrez trouver ces exemples sur la page <https://www.bnlearn.com/bnrepository/>. Vous pourrez donc aussi générer des réseaux bayésiens qui utilisent des variables qui ne sont pas nécessairement binaires. Vous pourrez donc générer par vous même les graphiques de la Figure 1. Reportez au moins ce résultat dans le rapport, et toute autre étude que vous conduirez dans ce cas.

Contraintes & Deadlines

- Le projet s'effectue en binôme. Ce travail fait partie de votre évaluation, c'est donc à vous de le réaliser ! En effectuant ce projet, je vous demande de tenir les engagements suivants :
 - vous vous *engagez* à ce que les membres du binôme soient les seuls auteurs de tout le code et de tout le rapport.
 - vous vous *engagez* à ne pas consulter du code source implémentant des réseaux bayésiens ou les algorithmes d'inférence approchée.
 - vous vous engagez également à ne partager aucune partie de votre rapport, ni aucune section de votre code source avec des étudiants autre que votre binôme.
 - une seule exception est autorisée si vous vous attaquez au bonus sur la génération : vous avez alors le droit d'utiliser une bibliothèque pour calculer avec une méthode exacte la réponse à une requête.
- Vous pouvez réaliser votre implémentation soit dans en python, soit en java.
- Vous devrez soumettre votre travail pour *le vendredi 27 mai 17 :00* via la plateforme moodle. Pour ce faire, un *seul* membre du binôme doit soumettre le projet.
- vous devez soumettre un rapport et une archive contenant votre code source. Le *seul* format accepté pour le rapport est le format pdf, le *seul* format accepté pour le format d'archive

est le format zip. Si vous utilisez un autre format, votre code ou votre rapport ne seront pas évalués.

- vous pouvez utiliser des bibliothèques pour manipuler ou représenter des graphes,
- Respectez le travail d'autrui : citez tout site web, tout ouvrage et tout article que vous aurez utilisé pour réaliser ce travail.
- Un binôme peut être constitué d'étudiants venant de différents groupes de TD.

Références

- [1] Stuart J. Russell and Peter Norvig. *Artificial Intelligence : a modern approach*. Pearson, 4 edition, 2021.

```

network unknown {
}
variable Burglary {
  type discrete [ 2 ] { True, False };
}
variable Earthquake {
  type discrete [ 2 ] { True, False };
}
variable Alarm {
  type discrete [ 2 ] { True, False };
}
variable JohnCalls {
  type discrete [ 2 ] { True, False };
}
variable MaryCalls {
  type discrete [ 2 ] { True, False };
}
probability ( Burglary ) {
  table 0.001, 0.999;
}
probability ( Earthquake ) {
  table 0.002, 0.998;
}
probability ( Alarm | Burglary, Earthquake ) {
  (True, True) 0.95, 0.05;
  (False, True) 0.29, 0.71;
  (True, False) 0.94, 0.06;
  (False, False) 0.001, 0.999;
}
probability ( JohnCalls | Alarm ) {
  (True) 0.9, 0.1;
  (False) 0.05, 0.95;
}
probability ( MaryCalls | Alarm ) {
  (True) 0.7, 0.3;
  (False) 0.01, 0.99;
}

```

$\text{Prob}(\text{Burglary} \mid \text{JohnCalls}=\text{True}, \text{MaryCalls}=\text{True}) = [0.7148927097115598, 0.28510729028844006]$