MADI
Apprentissage de paramètres et algorithme EM dans les BNs

Gaspard Ducamp (3200233)
Yoann Taillé (3200171)

UPMC - 2017

Table des matières

1 Expérimentations

1.1 Génération de données

Nous avons généré des données.

1.2 Apprentissage des paramètres à partir d'une base de données complètes

Il s'agit ici d'apprendre les paramètres par comptage.
Plus une base aléatoirement générée sera petite, moins l'apprentissage sera précis, comme montré dans le graphe ci-dessous.

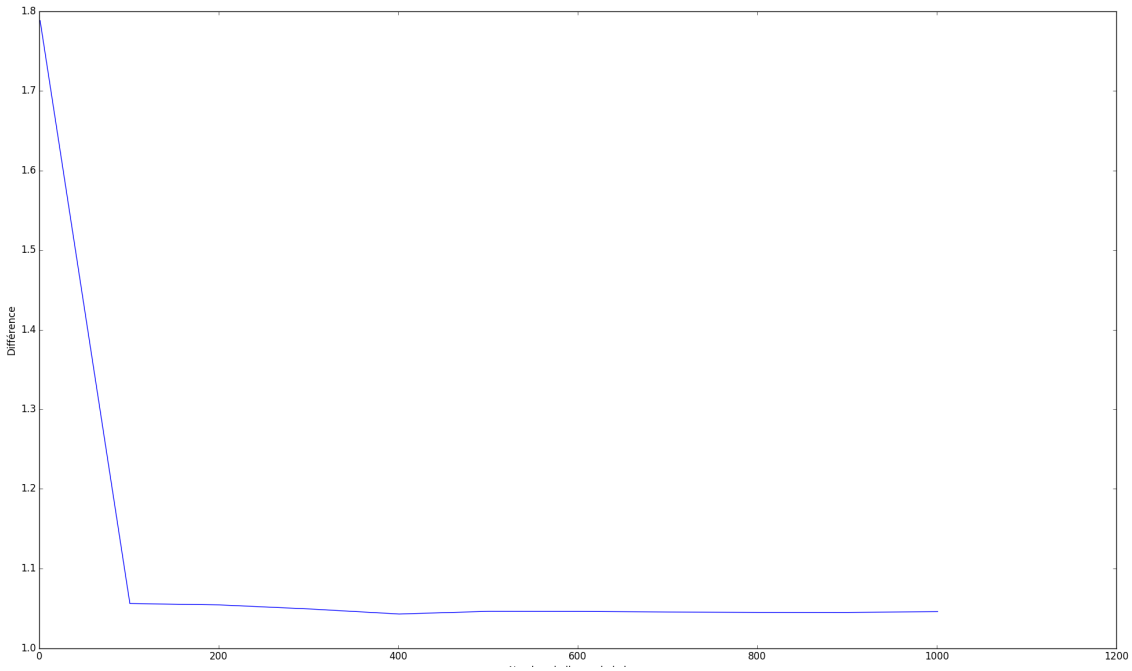

FIGURE 1 – Différence entre le BN original et un BN appris selon la taille de la base (en nombre de lignes)

1.3 Apprentissage des paramètres à partir d'une base de données avec valeurs manquantes

L'algorithme EM consiste, pour chaque ligne, à effectuer une inférence sur les paramètres des variables ayant une valeur inconnue (étape E). On effectue ensuite un maximum de vraisemblance prenant en compte ces paramètres et les cas connus de la base (étape M), puis on reprend l'étape E avec le résultat et ainsi de suite jusqu'à convergence. Cette dernière intervient si les paramètres ne changent pas d'une itération à l'autre, ou si un certain nombre d'itérations a été atteint.
L'inférence est limité à un certain nombre de variables, afin d'être sûr ne pas faire exploser le temps d'exécution.
Les résultats obtenus sont satisfaisants, dépendant tout de même de la taille des données. Comme montré ci-dessous, on peut remarquer que si la base est complète, EM sera moins performant que si des valeurs y sont inconnues. Nous expliquons cela par le fait que, dans le cas d'une base complète, on effectue en fait un comptage, certains cas n'étant donc pas représentés. Si des données sont manquantes, on effectue une inférence qui fournira de meilleurs paramètres.

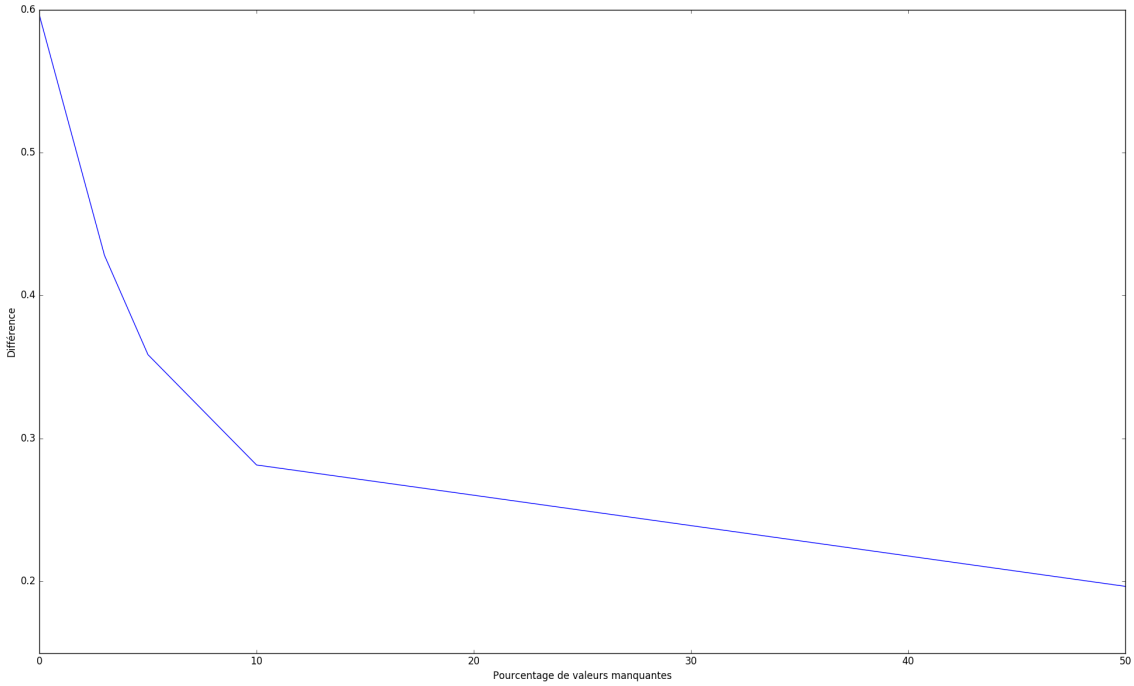

FIGURE 2 – Différence entre le BN original et un BN appris sur une base de 1000 lignes selon le pourcentage de valeurs manquantes

Un petit exemple graphique, parce que c'est joli :

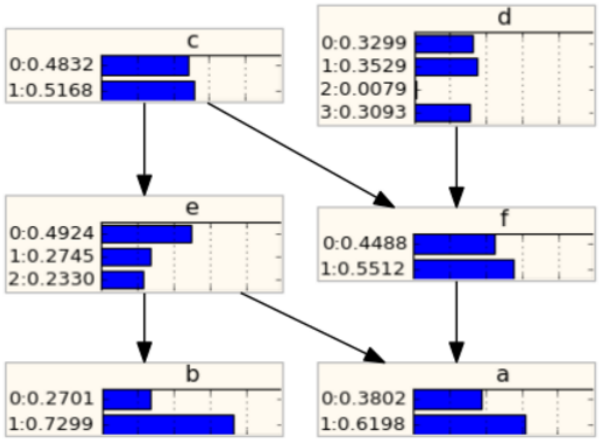

FIGURE 3 – BN originel

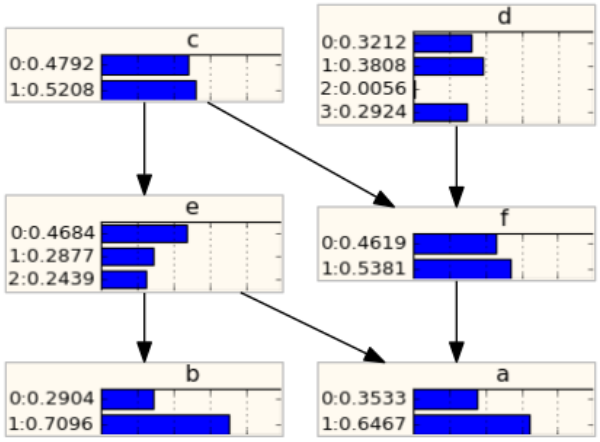

FIGURE 4 – BN dont les paramètres ont été appris sur une base de 1000 lignes avec 10% de valeurs manquantes

1.4 Apprentissage EM des paramètres d'une variable cachée

La variable ajoutée au BN fourni a comme probabilités marginales [0.3,0.7] comme conseillé dans l'énoncé. Nous avons fait le choix de donner à ses nouveaux enfants des CPTs aux paramètres catégoriques, chacun ayant une valeur à probabilité certaine. Par exemple, $P(e = 2 \mid x = 0) = 1$. Aucune autre modification n'a été apportée au reste du réseau fourni.
Une variable cachée aura toujours une valeur inconnue dans les bases générées. Notre algorithme EM est applicable à une base avec variable cachée, et donne encore une fois de bons résultats, notamment avec des probabilités discriminantes. Des observations similaires aux précédentes peuvent être faites sur la variation de l'erreur en fonction du pourcentage de valeurs manquantes, avec les mêmes explications.

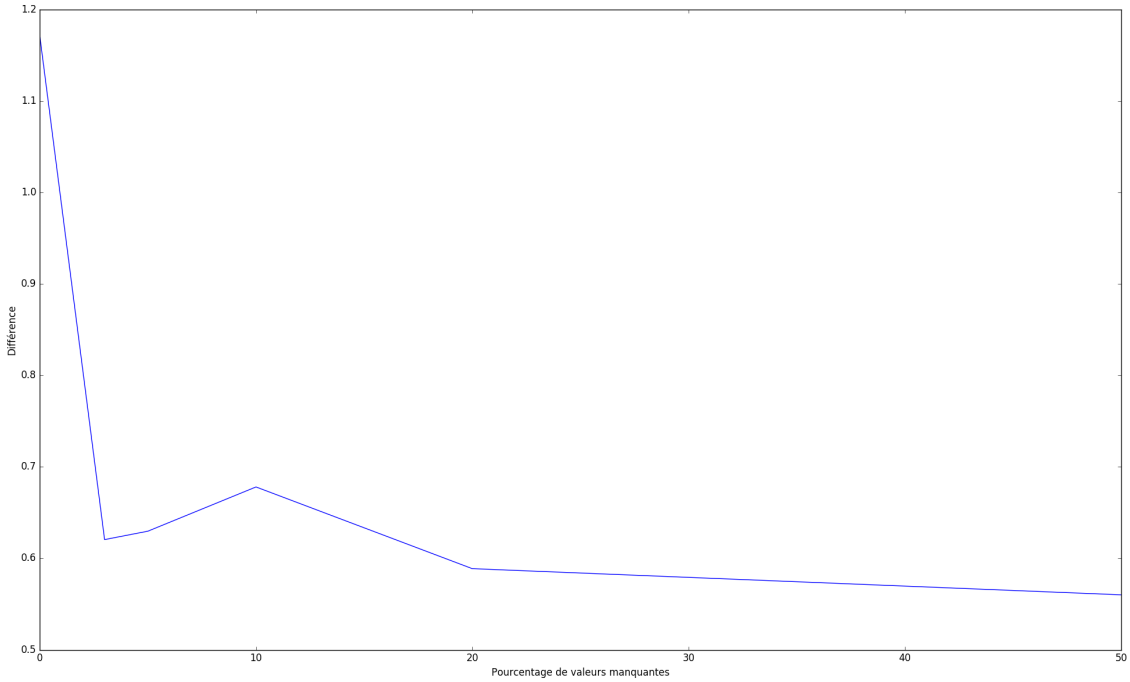

FIGURE 5 – Différence entre le BN à valeur cachée et un BN appris sur une base de 1000 lignes selon le pourcentage de valeurs manquantes

D'autres résultats graphiques, parce que c'est toujours aussi joli :

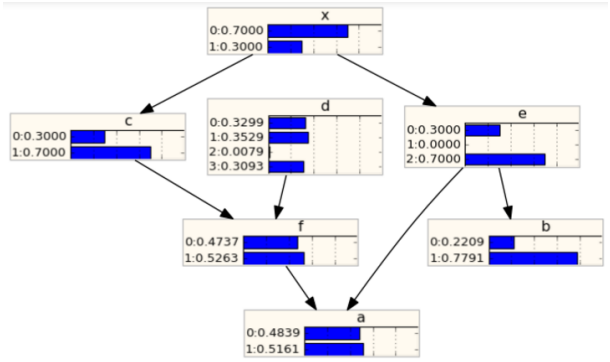

FIGURE 6 – BN à valeur cachée originel

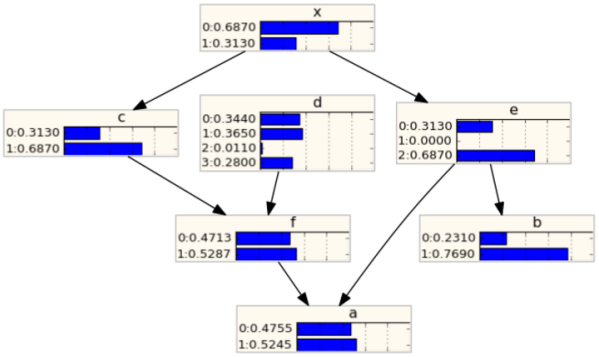

FIGURE 7 – BN dont les paramètres ont été appris sur une base de 1000 lignes avec 10% de valeurs manquantes

1.5 Apprentissage non supervisé des paramètres d'un Classifieur Naïf Bayes

Comme spécifié dans le cours, un CNB peut être représenté par un réseau bayésien, une variable, représentant le choix de classe, ayant pour enfants celles représentant les features de ces classes. On obtient donc, en lisant la base de données donnée, la structure suivante :

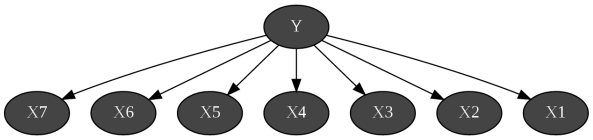

FIGURE 8 – Réseau Bayésien de type CNB

Nous avons décidé de fixer le nombre de classes possibles à 10, n'ayant pas réussi à trouver de méthode pour en déterminer un nombre optimal. La variable Y du BN précédent aura donc 10 valeurs possibles. Une fois la structure du BN construite, nous lui appliquons l'algorithme EM en apprenant les paramètres sur la base fournie. Nous obtenons donc les probabilités de chaque classe, et des features sachant la classe.

Un exemple de BN appris sur la base de 10000 lignes :

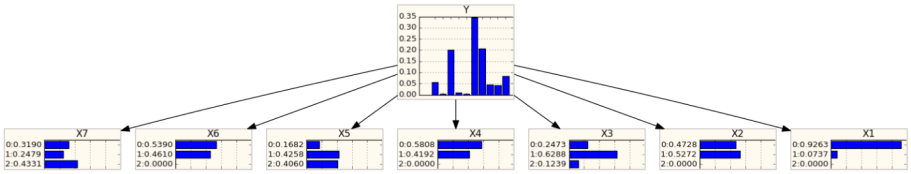

FIGURE 9 – Réseau Bayésien de type CNB appris sur la base

Afin de classer une base de test, nous effectuons un maximum de vraisemblance visant à exprimer la probabilité de chaque classe en fonction des cas. Nous nous basons sur la propriété : $P(Y \mid X1, X2...) \propto P(X1 \mid Y) \cdot P(X2 \mid Y) \cdot ... \cdot P(Y)$, les Xi étant indépendants sachant Y. Le résultat est donc la probabilité $P(Y \mid X)$ pour chaque cas X de la base.

L'étiquetage qui s'ensuit est simple : nous affectons à chaque cas la classe maximisant cette probabilité.