

Modélisation de l’incertitude — Master 2 MIAGE IA²

Travaux dirigés N° 2 : Raisonnement probabiliste

Andrea G. B. Tettamanzi
Université côte d’Azur
andrea.tettamanzi@univ-cotedazur.fr

Année universitaire 2020/2021

Résumé

On va comparer la classification bayésienne naïve avec un réseau bayésien à l’aide d’un jeu de données bien connu.

1 Introduction

Pour cette séance de TD, on va utiliser le célèbre jeu de données “Chess” (King-Rook vs. King-Pawn, ou roi + tour contre roi + pion), créé et décrit pour la première fois par Alen Shapiro [2] et disponible dans le *Machine Learning Repository* de l’Université de Californie à Irvine, à la page [https://archive.ics.uci.edu/ml/datasets/Chess+\(King-Rook+vs.+King-Pawn\)](https://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King-Pawn)).

2 Consignes

1. On commencera par récupérer les données brutes en ligne. Le fichier `ks-vs-kp.data` contient une description du jeu de données ainsi que les noms des 36 attributs disponibles pour prédire la classe de chacun enregistrement. C’est un problème de classification binaire : on veut savoir si le blanc peut gagner ou pas. On appellera ça `class`.
2. Appliquer à ce problème la classification bayésienne naïve.
3. Maintenant, on va construire un réseau bayésien, en considérant la liste de dépendances suivante (obtenue avec l’algorithme RSMAX2 [1], avec quelques ajustements manuels) :
 - `hdchk` \leftarrow `mulch` ;
 - `rxmsq` \leftarrow `qxmsq` ;
 - `simpl` \leftarrow `bkon8` ;
 - `wkcti` \leftarrow `cntxt` ;
 - `wkna8` \leftarrow `cntxt` ;
 - `bkspr` \leftarrow `rxmsq` ;
 - `wkpos` \leftarrow `cntxt`, `wkna8` ;
 - `bkona` \leftarrow `bkspr` ;
 - `bkxcr` \leftarrow `bkspr` ;
 - `dsopp` \leftarrow `bkspr`, `rxmsq` ;
 - `reskr` \leftarrow `wkcti`, `wkpos` ;
 - `bkxbq` \leftarrow `bkona`, `bkxcr` ;
 - `bkxwp` \leftarrow `bkxcr` ;
 - `dwipd` \leftarrow `reskr`, `wkcti` ;
 - `rimmx` \leftarrow `bkxcr` ;
 - `blxwp` \leftarrow `bkxwp`, `rkxwp` ;

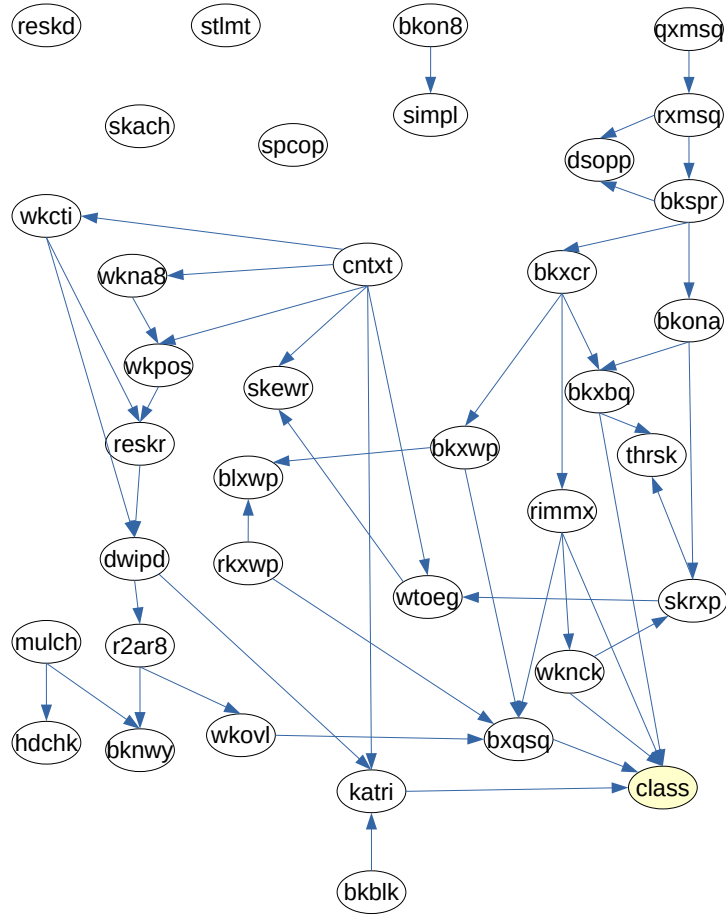


FIGURE 1 – Structure du réseau bayésien pour la classification du jeu de donnée “chess”.

- $r2ar8 \leftarrow dwipd$;
- $wknck \leftarrow rimmx$;
- $bknwy \leftarrow mulch, r2ar8$;
- $skrxp \leftarrow bkona, wknck$;
- $wkovl \leftarrow r2ar8$;
- $bxqsq \leftarrow bkxwp, rimmx, rkxwp, wkovl$;
- $thrsk \leftarrow bkxbq, skrxp$;
- $wtoeg \leftarrow cntxt, skrxp$;
- $skewr \leftarrow cntxt, wtoeg$;
- $katri \leftarrow cntxt, dwipd, bkblk$;
- $class \leftarrow bkxbq, bxqsq, rimmx, wknck, katri$.

Les dix variables $bkblk$, $bkon8$, $cntxt$, $mulch$, $qxmsq$, $reskd$, $rkxwp$, $skach$, $spcop$ et $stlmt$ ne dépendent pas des autres.

Le graphe dirigé acyclique correspondant à cette structure de réseau bayésien est montrée, pour votre convenance, en Figure 1.

4. Notez qu’on est dans le premier scénario mentionné au transparent n° 20 : la structure est donnée (construite à la main à l’étape précédente) et toutes les variables sont observables (les observations sont les enregistrements du jeu de données). On doit donc « apprendre »

juste les tables de probabilités conditionnelles. En fait, plus que d'apprentissage, il s'agit de faire les calculs, en estimant ces probabilités sur la base des fréquences des observations.

5. Utilisez ce réseau bayésien avec les tables de probabilités ainsi calculées pour faire la classification de ce jeu de données.
6. Comparez et discutez les résultats de ces deux modèles.

Rendez votre code et vos observations dans un archive zippé par courriel.

Références

- [1] Marco Scutari. Bayesian network constraint-based structure learning algorithms : Parallel and optimised implementations in the bnlearn R package. *CoRR*, abs/1406.7648, 2014.
- [2] Alen Shapiro. *Structured Induction in Expert Systems*. Addison-Wesley, 1987.