



**HAL**  
open science

## Apprentissage de CP-nets probabilistes

Damien Bigot, Jérôme Mengin, Bruno Zanuttini

► **To cite this version:**

Damien Bigot, Jérôme Mengin, Bruno Zanuttini. Apprentissage de CP-nets probabilistes. Journées Francophones sur les Réseaux Bayésiens et les Modèles Graphiques Probabilistes (JFRB 2014), Jun 2014, Paris, France. hal-01072534

**HAL Id: hal-01072534**

**<https://hal.science/hal-01072534>**

Submitted on 8 Oct 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Apprentissage de CP-nets probabilistes

Damien Bigot and Jérôme Mengin and Bruno Zanuttini

## 1 Introduction

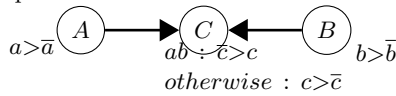
Le développement de systèmes interactifs d'aide à la décision et de systèmes de recommandation nécessite des modèles capables d'utiliser les préférences d'un utilisateur pour l'orienter dans ses choix. Les formalismes de représentation compacte de préférences font l'objet de travaux soutenus en intelligence artificielle depuis plus d'une quinzaine d'années [3]. Ces formalismes permettent l'expression de modèles suffisamment flexibles et riches pour décrire des comportements de décision complexes. Pour être intéressants en pratique, ces formalismes doivent de plus permettre l'élicitation des préférences de l'utilisateur, et ce en restant à un niveau admissible d'interaction. La configuration de produits et la recherche basée sur des préférences constituent de bons exemples de problèmes de décision où les préférences de l'utilisateur ne sont pas connues a priori [7, 6].

Lorsque les préférences sont qualitatives et structurées par des interactions simples entre attributs (notamment, lorsqu'elles sont séparables), on peut faire appel à des représentations du type CP-net [3] ; l'un des intérêts de ce modèle est qu'il permet d'apprendre par le biais de quelques questions, les préférences de l'utilisateur [5]. Malheureusement, dans le cadre de la recommandation, il est assez rare de connaître les préférences d'un utilisateur, et on a rarement le temps d'apprendre les préférences d'un utilisateur avant ou pendant la phase de recommandation. Par contre, la plupart des entreprises possèdent un historique des ventes déjà effectuées. Ces ventes peuvent être utilisées pour apprendre les préférences d'un groupe d'utilisateurs, sous la forme d'une distribution de probabilité sur les préférences des utilisateurs. C'est pour cela que l'on introduit les CP-nets probabilistes.

Nous présenterons donc une extension des CP-nets : les PCP-nets [2, 4]. À partir de cette extension, nous donnerons les différentes requêtes utilisables. Enfin, nous montrerons comment apprendre un PCP-net à partir d'un historique de ventes, lorsqu'une structure incluant notre PCP-net est connue.

## 2 CP-nets probabilistes

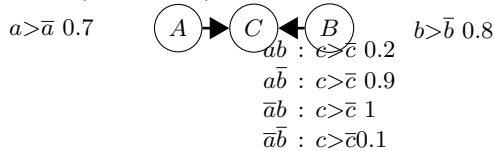
Un CP-net, sur un ensemble de variables  $\mathcal{X} = \{X_1, \dots, X_n\}$ , est un graphe orienté  $G = \{\mathcal{X}, A\}$  où  $A$  est l'ensemble des arêtes qui représentent des dépendances **préférentielles** entre les variables. A chaque sommet  $X_i \in \mathcal{X}$  est associée une table de préférence locale sur  $X_i$  en fonction de ses parents.



D'après la Figure ci-dessus, comme les CP-nets s'utilisent de façon Ceteris-Paribus ("toutes choses égales par ailleurs") et d'après la table de B, l'objet  $ab\bar{c}$  est préféré à  $a\bar{b}\bar{c}$  ; ce qu'on note  $\succ$ , on obtient donc  $ab\bar{c} \succ a\bar{b}\bar{c}$ , et on dit que  $ab\bar{c}$  domine  $a\bar{b}\bar{c}$ . Grâce à la table sur A, on a  $a\bar{b}\bar{c} \succ a\bar{b}c$ , donc par transitivité, on a  $ab\bar{c} \succ a\bar{b}c$ . Un CP-net représente donc un ordre partiel, transitif et irréflexif sur  $\mathcal{O} = \prod_{X_i \in \mathcal{X}} X_i$ . On introduit aussi l'objet optimal, c'est-à-dire l'objet qui n'est dominé par aucun autre objet. Pour les CP-nets acycliques, cet objet est unique et dans notre exemple c'est  $ab\bar{c}$ .

Comme nous nous intéressons aux préférences d'un groupe d'utilisateurs, nous allons ajouter des probabilités dans les tables de préférences locales afin de construire une distribution de probabilité sur les CP-nets.

Dans un PCP-net[2], les tables contiennent, pour chaque paire  $(X_i, u)$ , avec  $X_i \in \mathcal{X}$  et  $u \in U$  où  $U$  représente les parents de  $X_i$ , une distribution de probabilité sur l'ensemble des ordres possibles sur  $\underline{X}$  ; dans le cas de variables booléennes, cette distribution est entièrement définie par la probabilité que l'ordre soit  $x > \bar{x}$ . Dans la suite, on note  $p(X, u : x > \bar{x})$  cette probabilité.



On définit alors la probabilité d'un CP-net  $N$ , compatible avec un PCP-net (c-à-d avec la même structure  $G$ ), par :

$$P(N) = \prod_{(X, u)} p(X, u : \succ_{X, u}^N)$$

Compte tenu d'un PCP-net  $\mathcal{N}$ , les tâches de

raisonnement consistant à calculer les probabilités associées à des événements intéressants :

- La probabilité qu'un objet  $o$  domine un objet  $o'$ [2]. Cette probabilité est notée  $P_{\mathcal{N}}(o \succ o')$ . Son calcul est #P-difficile pour les PCP-net acyclique et un algorithme FPT est disponible en  $O(2^{2k^2} n)$  avec  $n$  le nombre de variables et  $k$  le nombre de valeurs différentes entre les deux objets pour les PCP-nets arborescents.
- La probabilité qu'un objet  $o$  soit optimal. Cette probabilité est notée  $P_{\mathcal{N}}(o = \text{opt})$ . Son calcul est linéaire dans le cadre des PCP-nets acycliques. On peut aussi transformer le PCP-net en réseau bayésien [4] associé (on remplace dans chaque table  $p(X, u : x > \bar{x})$  par  $p(x|u)$ ).
- Enfin, l'objet qui a la plus forte probabilité d'être optimal. Son calcul est linéaire dans le cas des PCP-nets arborescents[2]. Pour les acycliques, on utilise un algorithme probabilité maximum sur le réseaux bayésiens associés.

**Exemple 1.** Reprenons le PCP-net de la Figure 2 et les objets  $o_1 = ab\bar{c}$  et  $o_2 = a\bar{b}c$ .

La probabilité  $P(o_1 = \text{opt}) = p(a > \bar{a}) * p(b > \bar{b}) * p(ab : \bar{c} > c) = 0.7 \times 0.8 \times 0.8 = 0.448$

Enfin,  $P(o_1 \succ o_2) = p(b > \bar{b}) * (p(ab : \bar{c} > c) + p(a\bar{b} : \bar{c} > c)) = 0.8 * (0.8 + 0.1) = 0.72$

### 3 Apprentissage de PCP-nets

On suppose un PCP-net cible  $\bar{\mathcal{N}}$  représentant la distribution de probabilité sur les relations de préférence d'un groupe d'agents  $A$  et un PCP-net courant  $\mathcal{N}$ .

**Apprentissage offline** Pour cette partie, on peut utiliser un historique de vente. On suppose que cet historique représente les objets optimaux du groupe d'agents  $A$  à un moment donné. On utilisera donc cet historique comme une distribution d'objets optimaux.

Lorsque la structure de  $\bar{\mathcal{N}}$  est donné, on cherche à partir de l'historique à estimer pour chaque règle  $(X, u : x > \bar{x})$  sa probabilité. Pour cela, on montre facilement que :

$$p(X, u : x > \bar{x}) = P(\text{opt}[X] = x | \text{opt}[U] = u)$$

On note  $|ux|$  le nombre d'objets dans l'historique qui satisfont  $u$  et  $x$ , et  $|u|$  le nombre d'objets qui satisfont  $u$ . Alors, notre probabilité peut être estimée par :

$$P(X, u : x > \bar{x}) \sim |ux| / |u|$$

Lorsque  $|u| = 0$ , on peut utiliser, lorsqu'on en dispose, un historique de vente sous contrainte.

Dans le cas où la structure n'est pas connue, il est possible de l'apprendre en utilisant les algorithmes d'apprentissage sur les réseaux

bayésiens. Étant donné que les dépendances entre les variables préférentielles ne sont pas réversibles contrairement aux réseaux bayésiens, il faudra ajouter une phase d'élicitation afin d'orienter correctement ces dépendances, dans le but d'apprendre aussi la bonne distribution de probabilité sur les CP-nets.

**Apprentissage online** Dans le cas où on observe, l'un après l'autre, des objets considérés comme optimaux par des agents de  $A$ , on peut mettre à jour un PCP-net. Pour cela, on modifie la probabilité de chaque règle qui définit cet objet comme optimal avec la formule suivante [1] :

$$p(X, u : x > \bar{x}) = \eta \times P(x|o) + (1 - \eta) \times p(X, u : x > \bar{x})$$

où  $\eta$  est un taux d'apprentissage (dans notre cas,  $\eta = 1/k$  où  $k$  est le nombre de fois où on a visité la règle et  $P(x|o) = 1$  si  $o$  satisfait  $x$  et 0 sinon.

Lors de cette phase d'apprentissage, on peut poser des questions à l'utilisateur courant afin d'ajouter des contraintes sur l'objet lorsque l'on s'aperçoit que relation de préférence possède une probabilité proche de 0.

**Résultats** Afin de tester notre formule de mise à jour, on a utilisé le protocole d'apprentissage suivant : on cherche à apprendre le PCP-net  $\bar{\mathcal{N}}$  qui représente les relations de préférences du groupe d'agents  $A$ . Pour cela, on simule un ensemble  $m$  d'agents inclus dans  $A$ . Pour chaque agent, on tire son CP-net selon  $\bar{\mathcal{N}}$ . Puis on prend son objet optimal. Enfin, on modifie les règles qui rendent cet objet optimal avec la formule donnée plus haut.

Pour vérifier que le PCP-net apprit correspondait bien au PCP-net qu'on cherchait à apprendre, on estime la divergence de Kullback-Lieble sur la distribution d'objets optimaux des deux PCP-nets ainsi qu'une distance au carré entre les vecteurs de paramètres des deux PCP-nets. On remarque pour les deux distances, que notre PCP-net apprit converge vers le PCP-net cible lorsqu'on a vu entre 70 et 150 agents quel que soit le nombre de variables.

### 4 Conclusion

On a proposé dans cet abstract, une façon de représenter une distribution de CP-net afin d'avoir des requêtes du type : "Quel est la probabilité qu'un objet  $o$  soit optimal?" ou encore : "Quel est l'objet maximalement optimal?" permettant une utilisation pour de la recommandation. De plus, on a proposé une méthode permettant d'apprendre un PCP-net à partir d'un historique représentant une distribution d'objets optimaux.

## REFERENCES

- [1] Singer Y. Bauer E., Koller D. Updates rules for parameter estimation in bayesian network. In *Proceedings of UAI 1997*, 1997.
- [2] Mengin Jérôme Zanuttini Bruno Bigot Damien, Fargier Hélène. Probabilistic conditionnal preference network. In *Proceedings of UAI 2013*, 2013.
- [3] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *JAIR*, 21:135–191, 2004.
- [4] Mattei N. Rossi F. Brent K. Cornelio C., Goldsmith J. Dynamic and probabilistic cp-nets. In *Proceedings of Australasian Joint Conference on Artificial Intelligence 2013*, 2013.
- [5] Frédéric Koriche and Bruno Zanuttini. Learning conditional preference networks with queries. In *Artificial Intelligence*, pages 685–703, 2010.
- [6] Daniel Mailharro. A classification and constraint-based framework for configuration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, pages 383–397, September 1998.
- [7] Paolo Viappiani, Boi Faltings, and Pearl Pu. Preference-based search using example-critiquing with suggestions. *JAIR*, 27:465–503, 2006.