UNIVERSITÉ CAEN NORMANDIE

UFR des Sciences

Université de Caen Normandie
Département mathématiques-informatique
2020–2021, 2e session session
29 juin 2021
Grégory Bonnet, Abdelkader Ouali, Bruno Zanuttini

L3 informatique, L3 mathématiques

Examen

Unité M.MIM5E2 : Aide à la décision et intelligence artificielle
2 h — Tous documents autorisés

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

Le barême est sur 21 points. La note sera tronquée à 20, c'est-à-dire que si vous obtenez 21/21 ou 20/21, vous aurez une note de 20/20, si vous obtenez 15/21, vous aurez une note de 15/20, etc.

1 Fouille de données (7 points)

Un restaurateur souhaite obtenir des informations sur les habitudes de consommation de ses clients. Pour cela, il collecte des informations sur les repas consommés sur l'année écoulée, et trouve :

- 7 421 repas constitués d'un plat et d'un dessert, dont 1 333 avec une boisson,
- 8 793 repas constitués d'une entrée, d'un plat et d'un dessert, dont 4 747 avec une boisson,
- 3 333 repas constitués d'une entrée et d'un plat, dont 1 545 avec une boisson,
- 453 repas constitués d'une entrée, d'un plat, de fromage et de dessert, tous avec une boisson.

Question 1 (2 points)**.** *On souhaite modéliser ces données comme une base de données transactionnelle. À quoi en correspondraient les items et les transactions ? Donner une représentation de ces données sous forme d'une table.*

Correction Les items évidents sont les items E, P, D, B, F, représentant la présence d'une entrée, d'un plat, d'un dessert, d'une boisson, et de fromage, respectivement, dans le repas. Un repas sera alors représenté par une transaction. On peut alors représenter les données sous la forme suivante, où chaque ligne correspond à un type de repas, dont le nombre est indiqué en première colonne, et où « X » représente la présence de l'item dans la transaction.

	E	P	D	B	F
1 333		X	X	X	
6 088		X	X		
4 747	X	X	X	X	
4 046	X	X	X		
1 545	X	X		X	
1 788	X	X			
453	X	X	X	X	X

□

Question 2 (3 points)**.** *Le restaurateur souhaite connaître les combinaisons d'éléments qui ont été choisies au moins 6 000 fois dans l'année. Doit-il rechercher des règles d'associations ou des itemsets ? Avec quel(s) seuil(s) ? Donner les combinaisons en question, en justifiant pour chacune.*

Correction Il s'agit de trouver des itemsets fréquents, avec un seuil de fréquence absolu de 6 000, ou encore un seuil de fréquence relatif de 30 % (soit 6 000 par rapport aux 20 000 transactions).

Pour trouver les combinaisons en question, on constate tout d'abord que la seule combinaison de taille 5 n'est supportée que par les 453 dernières transactions. La seule combinaison de taille 4 supportée par d'autres transactions est la combinaison EPDB, mais on voit qu'elle n'est supportée que par $4\,747+453 < 6\,000$ transactions.

Les combinaisons fréquentes sont donc de taille 3 au plus, et ne peuvent comporter l'item F (car elles ne seraient alors supportées que par les 453 dernières transactions). Il y a 4 combinaisons à considérer, et on voit que PDB, EPB et EPD sont fréquentes (supportées respectivement par $1\,333 + 4\,747 + 453$, par $4\,747 + 1\,545 + 453$, et par $4\,747 + 4\,046 + 453$ transactions), tandis que EDB n'est supportée que par $4\,747 + 453$ transactions.

Par antimonotonie de la fréquence, les sous-ensembles de taille 2 de ces combinaisons fréquentes PDB, EPB et EPD, ce qui revient à toutes les combisons de deux items parmi E, P, D, B, sont également fréquents, ainsi que les singletons E, P, D et B (et l'itemset vide). Au final, les itemsets fréquents sont donc PDB, EPB, EPD, EP, ED, EB, PD, PB, DB, E, P, D, B (et l'itemset vide). □

Question 3 (2 points)**.** *Le restaurateur souhaiterait pouvoir prédire, en utilisant ces statistiques, les situations où le client commandera une boisson, selon qu'il a commandé une entrée et/ou un plat et/ou un dessert et/ou du fromage. Il souhaite que ces prédictions soient justifiées par au moins 100 repas de l'année écoulée, et soient vérifiées au moins 1 fois sur 2 par les repas concernés de son historique. Doit-il rechercher des règles d'association ou des itemsets ? Avec quel(s) seuil(s) ? Donner une façon de prédire qui soit correcte, et une qui soit fausse trop souvent, en justifiant.*

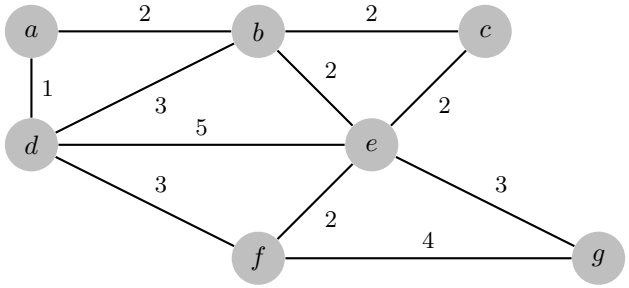
Correction Il s'agit cette fois de rechercher des règles d'association, qui concluent sur l'item B. Ces règles doivent avoir une confiance de 50 % au moins, et un support absolu de 100 au moins.

On voit que la règle EP⇒B est correcte, puisqu'elle est supportée par $4\,747 + 1\,545 + 543 > 100$ transactions et a une confiance de $\frac{4\,747+1\,545+543}{(4\,747+1\,545+543)+(4\,046+1\,788)} > 1/2$. La règle F⇒B est également correcte, avec un support de 453 et une confiance de 100 %, et il en existe encore d'autres.

Par opposition, la règle PD⇒B n'est pas correcte ; elle est supportée par $1\,333+4\,747+453$ transactions, mais a une confiance de $\frac{1\,333+4\,747+453}{(1\,333+4\,747+453)+(6\,088+4\,046)} < 1/2$. Il y a évidemment d'autres règles incorrectes pour la même raison, par exemple la règle P⇒B. □

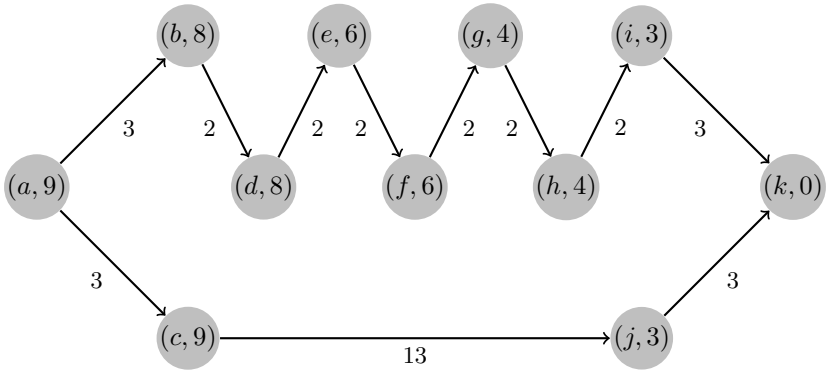
2 Planification (7 points)

On considère le graphe non orienté ci-dessous, qui représente un problème de planification. Les nœuds correspondent à des états et les arêtes à des actions. Les valeurs qui étiquettent chaque arête correspondent au coût de l'action.



Question 4 (4 points)**.** *Appliquer l'algorithme de Dijkstra pour calculer le plus court chemin entre les nœuds a et g. Décrire pour chaque étape de l'algorithme les valeurs prises par les tables des distances, des pères et des actions (une action peut être notée par exemple (a, b), pour l'action qui consiste à aller de a à b). Indiquer également l'état de la liste des ouverts.*

On considère maintenant le graphe orienté ci-dessous. Ce dernier décrit comme précédemment un problème de planification. Chaque nœud est noté (x, y), où x est l'identifiant du nœud (parmi $a, \ldots, k$) et y est la valeur heuristique de ce nœud au regard de l'état but.



Question 5 (3 points)**.** *Dessiner l'arbre de recherche construit avec l'algorithme A* lorsqu'il est utilisé pour calculer le plus court chemin allant de a (état initial) à k (état but).*

3 Contraintes (7 points)

On considère le CSP binaire discret $P = (X, D, C)$ défini par :
— $X = \{X_1, X_2, X_3, X_4\}$,
— $D(X_1) = \{a, b, c\}$, $D(X_2) = D(X_4) = \{a, c\}$, $D(X_3) = \{b\}$,
— $C = \{c_1, c_2, c_3, c_4, c_5\}$ avec $c_1 = (X_1 \neq X_2)$, $c_2 = (X_1 \neq X_3)$, $c_3 = (X_2 \neq X_3)$, $c_4 = (X_2 \neq X_4)$, $c_5 = (X_3 \neq X_4)$.

On choisit l'ordre statique d'instanciation des variables (X_1, X_2, X_3, X_4), et l'ordre statique (a, b, c) pour le choix des valeurs du domaine pour chacune des variables.

Question 6 (3 points)**.** *Appliquer l'algorithme de recherche Backtrack (SRA) au CSP P jusqu'à trouver la première solution. Expliquer les différentes étapes rencontrées dans l'arbre de recherche.*

Correction On explore $X_1 = a$, qui constitue une affectation partielle cohérente, puis $X_2 = a$. L'instanciation partielle obtenue falsifie la contrainte c_1, on revient donc sur l'affectation $X_2 = a$ et on passe à la valeur suivante du domaine de X_2, c'est-à-dire c. L'instanciation partielle obtenue satisfait la seule contrainte ne portant que sur X_1 et X_2 (c_1), on continue donc en affectant X_3 à la première (et seule) valeur de son domaine, c'est-à-dire b. À nouveau, l'affectation obtenue satisfait toutes les contraintes ne portant que sur X_1, X_2, X_3 (c_1, c_2, c_3), et on continue avec X_4. On affecte X_4 à a, et on voit que l'affectation obtenue est complète et satisfait toutes les contraintes. On a donc obtenu la solution de P donnée par $X_1 = a, X_2 = c, X_3 = b, X_4 = a$. □

Question 7 (4 points)**.** *Appliquer l'algorithme d'arc-consistance AC-3 comme procédure de filtrage à chaque nœud dans l'algorithme précédent (y compris à la racine), en expliquant l'état de la pile à chaque étape. Que remarque-t-on ?*

Correction On commence à la racine, avec les domaines définis par P, soit

$$D(X_1) = \{a, b, c\}, D(X_2) = \{a, c\}, D(X_3) = \{b\}, D(X_4) = \{a, c\}$$

La liste des couples de variables à traiter est l'ensemble des couples de variables concernées par une même contrainte, soit

$$Q = \{(X_1, X_2), (X_2, X_1), (X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2), (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)\}$$

On commence donc par traiter (X_1, X_2), et l'on voit que toutes les valeurs de $D(X_1)$ sont supportées dans $D(X_2)$ vis-à-vis de c_1 ; les domaines restent donc inchangés, et le couple (X_1, X_2) est supprimé de Q. On traite ensuite (X_2, X_1), et de même aucune valeur n'est à supprimer de $D(X_2)$, et (X_2, X_1) est supprimé de Q. On a alors toujours

$$D(X_1) = \{a, b, c\}, D(X_2) = \{a, c\}, D(X_3) = \{b\}, D(X_4) = \{a, c\}$$

et on a

$$Q = \{(X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2), (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)\}$$

On traite alors le couple (X_1, X_3), et on voit que $b \in D(X_1)$ n'est pas supporté dans $D(X_3)$ vis-à-vis de la contrainte c_2 ; on supprime donc b de $D(X_1)$, et on réintroduit le couple (X_2, X_1) dans Q ; on a donc

$$D(X_1) = \{a, c\}, D(X_2) = \{a, c\}, D(X_3) = \{b\}, D(X_4) = \{a, c\}$$

et

$$Q = \{(X_3, X_1), (X_2, X_3), (X_3, X_2), (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3), (X_2, X_1)\}$$

On voit alors, en traitant successivement les couples de Q, qu'aucune suppression n'est plus à faire.

On commence alors à développer l'arbre avec les domaines obtenus, en affectant donc a à X_1. On réduit donc $D(X_1)$ à $\{a\}$, et on relance donc AC-3 avec les domaines

$$D(X_1) = \{a\}, D(X_2) = \{a, c\}, D(X_3) = \{b\}, D(X_4) = \{a, c\}$$

et les couples

$$Q = \{(X_1, X_2), (X_2, X_1), (X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2), (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)\}$$

La seule valeur de $D(X_1)$, a, est supportée dans $D(X_2)$ vis-à-vis de c_1, donc (X_1, X_2) est simplement supprimé de Q. En revanche, $a \in D(X_2)$ n'est pas supportée dans $D(X_1)$, donc on la supprime de $D(X_2)$ et on obtient

$$D(X_1) = \{a\}, D(X_2) = \{c\}, D(X_3) = \{b\}, D(X_4) = \{a, c\}$$

avec

$$Q = \{(X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2), (X_2, X_4), (X_4, X_2), (X_3, X_4), (X_4, X_3)\}$$

En traitant (X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2), (X_2, X_4), on voit qu'il n'y a aucune valeur à supprimer. En revanche, en traitant (X_4, X_2), on voit que $c \in D(X_4)$ n'est pas supportée dans $D(X_2)$; on supprime donc c de $D(X_4)$, et aucun couple n'est à réintroduire dans Q (le couple (X_3, X_4) y est déjà). On a alors

$$D(X_1) = \{a\}, D(X_2) = \{c\}, D(X_3) = \{b\}, D(X_4) = \{a\}$$

et

$$Q = \{(X_3, X_4), (X_4, X_3)\}$$

En traitant ces couples successivement, on voit alors qu'il n'y a rien à supprimer dans aucun des domaines. On a alors

$$D(X_1) = \{a\}, D(X_2) = \{c\}, D(X_3) = \{b\}, D(X_4) = \{a\}$$

et $Q = \emptyset$. On affecte alors successivement les variables X_2, X_3, X_4, et on constate qu'AC-3 ne supprime plus aucune valeur. On obtient au final la même solution qu'à la question précédente.

On constate que sur cette instance, le filtrage a permis d'éviter tout retour en arrière dans la recherche. □