Actes Cap'06, pp. 155-170. Presses Universitaires de Grenoble.

Co-classification sous contraintes

Ruggero G. Pensa, Céline Robardet, Jean-François Boulicaut

INSA Lyon, LIRIS CNRS UMR 5205
Bâtiment Blaise Pascal
F-69621 Villeurbanne cedex, France
ruggero.pensa@insa-lyon.fr
celine.robardet@insa-lyon.fr
jean-francois.boulicaut@insa-lyon.fr

Résumé : La co-classification est une technique de classification conceptuelle importante. Dans le cas de données catégorielles, il s'agit de calculer des collections de bi-clusters, i.e., des clusters d'objets et de couples atttributs-valeurs associés (propriétés booléennes). En marge du besoin classique d'optimiser une fonction objectif sur la qualité des groupements, l'amélioration de la pertinence des bi-clusters calculés reste une tâche difficile. Tout d'abord, il faudrait pouvoir exprimer l'intérêt subjectif de l'analyste, e.g., la définition déclarative de ses attentes au regard de sa connaissance du domaine. Ensuite, même si de telles spécifications existent, par exemple au moyen de contraintes sur les bi-clusters, l'exploitation de ces contraintes lors du processus heuristique de classification reste un problème ouvert. A notre connaissance, la classification sous contraintes n'a été que peu étudiée et n'a concerné des types de contraintes simples. Tout d'abord, nous considérons la co-classification plutôt qu'une classification monodimensionnelle. Ensuite, nous étudions de nouveaux types de contraintes utiles à l'analyse de données ordonnées, par exemple dans le temps. Enfin, nous montrons que notre cadre générique de co-classification à partir de motifs locaux peut être exploité pour la co-classification sous contraintes. Nous réalisons une validation expérimentale sur deux jeux de données d'expression de gènes.

1 Introduction

De nombreuses techniques de fouille de données ont été développées pour assister la découverte de connaissances à partir de grandes matrices booléennes. Ce type de données permet d'enregistrer quelles sont les propriétés satisfaites (attributs ou colonnes) par un certain nombre d'objets (lignes). Par exemple, dans la matrice $\mathbf{r}$ (Table 1), l'objet t_2 satisfait seulement les propriétés g_2 et g_5. L'une des applications qui motive nos recherches concerne l'analyse de données booléennes pour l'étude du transcriptome (e.g., des matrices qui codent la sur-expression de gènes dans un certain nombre de conditions expérimentales (Besson *et al.*, 2005)). Les processus de fouille s'appuient souvent sur des techniques de classification ("clustering") qui fournissent des motifs globaux, i.e., des regroupements prenant leur sens dans l'ensemble

	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

TAB. 1 – Un contexte booléen **r**

des données. La classification a été très étudiée, y compris dans le cas particulier des données booléennes. Il s'agit de calculer des partitions sur les objets et/ou les propriétés de sorte qu'une fonction objectif qui détermine la qualité des groupements soit optimisée (Jain & Dubes, 1988). De nombreux algorithmes calculent de bonnes partitions mais peu de méthodes permettent d'interpréter les groupements au moyen de caractérisations symboliques. Ce problème a motivé la classification conceptuelle (Fisher, 1987) et, notamment, les approches de co-classification ("co-clustering", "bi-clustering", voir (Madeira & Oliveira, 2004) pour une synthèse). Le but d'une co-classification est de calculer des bi-clusters, i.e., des associations d'ensembles d'objets (non nécessairement disjoints) avec des ensembles de propriétés. Un exemple de bi-partition dans **r** est $\{\{\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}\}, \{\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\}\}\}$. Le premier bi-cluster dit que la caractérisation des objets de $\{t_1, t_3, t_4\}$ est qu'ils partagent presque toujours les propriétés de $\{g_1, g_3, g_4\}$. De même , les propriétés dans $\{g_2, g_5\}$ sont caractéristiques des objets de $\{t_2, t_5, t_6, t_7\}$.

Nous nous intéressons à la pertinence des bi-partitions. Lorsque l'analyste utilise un algorithme de classification, il/elle a un très faible contrôle sur les groupes qui seront calculés. Typiquement, il est possible de choisir parmi plusieurs métriques ou bien décider d'une stratégie d'initialisation. Tous ces réglages opérationnels sont conceptuellement éloignés d'une spécification déclarative des propriétés souhaitées pour la bi-partition. Nous considérons qu'il est important que les analystes puissent spécifier leurs attentes (intérêt subjectif) au moyen de contraintes et qu'il faudrait des techniques de co-classification qui produisent des résultats cohérents vis-à-vis de ces spécifications. Le modèle simple de (Mannila, 1997) aide à formaliser ce point de vue. Une classification peut être vue comme un processus d'évaluation d'une requête inductive qui calculerait $\{\phi \in \mathcal{L} \mid q(r, \phi) \text{ est vrai}\}$. où **r** serait une matrice booléenne, $\mathcal{L}$ désignerait le language des bi-partitions sur une telle matrice, et le prédicat q spécifierait les propriétés attendues sur la bi-partition ϕ. Une vision plutôt classique est que ce prédicat va exprimer une contrainte d'optimisation sur la fonction objectif utilisée, e.g., le coefficient τ de Goodman-Kruskal (Robardet & Feschet, 2001) ou la perte d'information mutuelle dans (Dhillon *et al.*, 2003). On peut également trouver d'autres contraintes comme la définition du nombre de bi-clusters, le fait que certains objets (resp. propriétés) doivent (resp. ne doivent pas) être ensemble, etc. Autrement dit, nous aimerions pouvoir réaliser la tâche comme une sélection de bi-partitions en supposant que toutes

les bi-partitions aient été calculées a priori. Nous savons bien qu'un tel calcul est impossible. On comprend donc la nature heuristique des algorithmes de classification qui utilisent des méthodes d'optimisation locales pour la fonction objectif (i.e., la satisfaction de la contrainte d'optimisation globale ne peut pas être garantie). Combiner ces heuristiques avec la satisfaction d'autres contraintes sur les bi-clusters est clairement un problème difficile. A notre connaissance, l'exploitation de contraintes n'a pas encore été étudiée pour la co-classification. L'introduction de contraintes dans des processus de classification mono-dimensionnels (e.g., K-Means, classification hiérarchique) a motivé quelques travaux (Wagstaff *et al.*, 2001; Basu *et al.*, 2002; Klein *et al.*, 2002; Basu *et al.*, 2004; Davidson & Ravi, 2005b; Davidson & Ravi, 2005a). Des contraintes simples ont été considérées comme "must-link" et "cannot-link". La principale motivation de ces études était la classification semi-supervisée (i.e., améliorer les techniques prédictives lorsque l'ensemble d'apprentissage ne contient que peu d'instances étiquetées). Non seulement nous considérons la co-classification au lieu de la classification mais aussi notre point de vue est différent. Puisque certaines des contraintes spécifient l'intérêt subjectif de l'analyste, nous devons les prendre en compte même si elles nous pénalisent du point de vue de la fonction objectif : nous ne cherchons pas de bons bi-clusters grâce aux contraintes mais plutôt de bons bi-clusters malgré les contraintes.

Notre contribution concerne donc d'abord de nouveaux types de contraintes. A coté des extensions des contraintes "must-mink" et "cannot-link" pour la co-calssification, nous considérons le cas où l'une ou même les deux dimensions sont ordonnées. Dans le contexte de l'analyse de données d'expression booléennes, c'est par exemple le cas des données qui enregistrent l'évolution de l'expression des gènes au cours du temps. Nous proposons de spécifier si une collection de bi-clusters doit ou ne doit pas être cohérente vis-à-vis de tels ordres, i.e., les contraintes "interval" et "non-interval". Notre seconde contribution concerne le cadre algorithmique pour calculer des bi-partitions satisfaisant les contraintes spécifiées. Nous avons récemment proposé un cadre de co-classification générique (Pensa *et al.*, 2005). Notre idée est de calculer des bi-partitions à partir de motifs locaux (bi-ensembles) qui capturent des associations localement fortes, par exemple des concepts formels. Nous montrons qu'il est possible de l'étendre vers la classification sous contraintes, autrement dit qu'il est possible d'exploiter les contraintes fixées au niveau global (sur les bi-clusters) pour en dériver des contraintes exploitables au niveau local (sur les bi-ensembles).

La Section 2 formalise le problème traité. La Section 3 résume le cadre introduit dans (Pensa *et al.*, 2005) et discute de son extension vers la co-classification sous contraintes. La Section 4 est dédiée à une validation expérimentale sur deux jeux de données réelles. La Section 5 est une brève conclusion.

2 Formalisation du problème

Soit $\mathcal{T} = \{t_1, \ldots, t_m\}$ un ensemble d'objets et $\mathcal{G} = \{g_1, \ldots, g_n\}$ un ensemble de propriétés booléennes. Le contexte booléen à explorer est $\mathbf{r} \subseteq \mathcal{T} \times \mathcal{G}$, où $r_{ij} = 1$ si la propriété g_j est satisfaite par l'objet t_i, et 0 sinon. La tâche de co-classification est ainsi définie : on veut calculer une partition $P^{\mathcal{T}}$ de K groupes (clusters) d'objets (notés $\{P_1^{\mathcal{T}}, \ldots, P_K^{\mathcal{T}}\}$) et une partition $P^{\mathcal{G}}$ de K groupes de propriétés (notés $\{P_1^{\mathcal{G}}, \ldots, P_K^{\mathcal{G}}\}$)

avec une bijection σ entre les deux partitions, tel que chaque groupe d'objets est caractérisé par un seul groupe de propriétés ($\sigma : P^{\mathcal{T}} \rightarrow P^{\mathcal{G}}$). La bi-partition constituée de K bi-clusters est notée $\mathcal{P} = \{P_1, \ldots, P_K\}$ et $P_i = (P_i^{\mathcal{T}}, \sigma(P_i^{\mathcal{T}}))$.

Par la suite, $\mathcal{D}$ représente soit $\mathcal{T}$ soit $\mathcal{G}$. Définissons maintenant quelques contraintes primitives qui nous paraissent intéressantes pour la co-classification.

– **optimisation** Soit $f(\mathbf{r}, \mathcal{P})$ une fonction objectif, soit une contrainte d'optimisation

$$\mathcal{C}_{opt}(\mathbf{r}, f, \mathcal{P}) \text{ est satisfaite ssi } \mathcal{P} = \underset{\phi \in \mathcal{L}}{\operatorname{argmin}}\, f(\mathbf{r}, \phi)$$

où $\mathcal{L}$ est le langage des bi-partitions. Chaque algorithme de co-classification possède sa propre fonction objectif et une approche heuristique locale pour aborder l'optimisation.

– **must-link étendue** Si deux points x_i et x_j sont impliqués dans une contrainte "must-link" étendue, notée $\mathcal{C}_{eml}(x_i, x_j, \mathcal{P})$, ils doivent être dans le même bi-cluster de $\mathcal{P}$.

– **cannot-link étendue** Si deux points x_i, x_j sont impliqués dans une contrainte "cannot-link" étendue, notée $\mathcal{C}_{ecl}(x_i, x_j, \mathcal{P})$, il ne peuvent pas être dans le même bi-cluster de $\mathcal{P}$.

– **interval** et **non-interval** Si un ordre ($\preceq$) est défini sur $\mathcal{D}$, une contrainte "interval" sur cette dimension, notée $\mathcal{C}_{int}(\mathcal{D}, \mathcal{P})$, exige que chaque groupe sur $\mathcal{D}$ soit un intervalle : $\forall k = 1 \ldots K$, si $x_i, x_j \in P_k^{\mathcal{D}}$ alors $\forall x_l$ tel que $x_i \preceq x_l \preceq x_j$, $x_l \in P_k^{\mathcal{D}}$. Une contrainte "non-interval", notée $\mathcal{C}_{non-int}(\mathcal{D}, \mathcal{P})$ spécifie que les groupes sur $\mathcal{D}$ ne doivent pas être des intervalles : $\forall k = 1 \ldots K$, $\exists x_i, x_j \in P_k^{\mathcal{D}}$, $\exists x_l \in \mathcal{D}$ tel que $x_i \preceq x_l \preceq x_j$, $x_l \notin P_k^{\mathcal{D}}$.

Dans les approches existantes, les contraintes "must-link" et "cannot-link" s'appliquaient à l'une des dimensions. Pour une co-classification, il est naturel d'autoriser de telles contraintes sur les deux dimensions, éventuellement simultanément.

Les contraintes "interval" et "non-interval" sont utiles lorsque l'une ou les deux dimensions sont ordonnées (e.g., dans le temps ou dans l'espace). Par exemple, dans l'analyse de données d'expression de gènes, les conditions biologiques peuvent être ordonnées dans le temps (données cinétiques pour suivre l'évolution de l'expression des gènes au cours du temps). Une contrainte "interval" peut être utilisée pour trouver des groupes qui ne concernent que des instants adjacents (i.e., qui constituent des intervalles de temps continus), tandis qu'une contrainte "non-interval" peut être utilisée pour trouver des groupes qui ne sont pas des intervalles. Dans le premier cas, nous pouvons capturer des associations qui caractérisent chaque stade de la période d'échantillonnage. Dans le second cas, nous pouvons mettre en évidence des interactions qui sont en quelque sorte non dépendantes du temps.

3 Une approche "Local-vers-global" (L2G)

3.1 Utilisation de motifs locaux pour construire des bi-partitions

Dans (Pensa *et al.*, 2005), nous avons introduit un cadre générique de co-classification. Nous le résumons avant de montrer comment il peut être exploitée pour la prise en

compte de contraintes fixées par l'analyste. L'idée principale consiste à calculer les bi-partitions à partir de bi-ensembles qui capturent des association localement fortes entre des ensembles d'objets et des ensembles de propriétés. Formellement, un bi-ensemble est un élément $b_j = (T_j, G_j)$ ($T_j \subseteq \mathcal{T}$, $G_j \subseteq \mathcal{G}$) et nous supposons qu'une collection de bi-ensembles a priori intéressants $\mathcal{B}$ a été d'abord extraite dans $\mathbf{r}$. On décrit b_j au moyen d'un vecteur booléen $< \mathbf{t}_j >, < \mathbf{g}_j > = < t_{j1}, \ldots, t_{jm} >, < g_{j1}, \ldots, g_{jn} >$ où $t_{jk} = 1$ si $t_k \in T_j$ (0 sinon) et $g_{jk} = 1$ si $g_k \in G_j$ (0 sinon). On cherche K groupes de bi-ensembles $\{P_1^{\mathcal{B}}, \ldots, P_K^{\mathcal{B}}\}$ ($P_i^{\mathcal{B}} \subseteq \mathcal{B}$). On peut alors définir le centroïde d'un groupe de bi-ensembles $P_i^{\mathcal{B}}$ comme $\mu_i = < \tau_i >, < \gamma_i > = < \tau_{i1}, \ldots, \tau_{im} >$, $< \gamma_{i1}, \ldots, \gamma_{in} >$ où τ et γ sont des calculs classiques de centroïde :

$$\tau_{ik} = \frac{1}{|P_i^{\mathcal{B}}|} \sum_{b_j \in P_i^{\mathcal{B}}} t_{jk}, \quad \gamma_{ik} = \frac{1}{|P_i^{\mathcal{B}}|} \sum_{b_j \in P_i^{\mathcal{B}}} g_{jk}$$

Définissons maintenant notre distance entre un bi-ensemble et un centroïde :

$$d(b_j, \mu_i) = \frac{1}{2} \left(\frac{|\mathbf{t}_j \cup \boldsymbol{\tau}_i| - |\mathbf{t}_j \cap \boldsymbol{\tau}_i|}{|\mathbf{t}_j \cup \boldsymbol{\tau}_i|} + \frac{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i| - |\mathbf{g}_j \cap \boldsymbol{\gamma}_i|}{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i|} \right)$$

Il s'agit de la moyenne des différences symmétriques pondérées sur un ensemble de composantes. On suppose que $|\mathbf{t}_j \cap \boldsymbol{\tau}_i| = \sum_{k=1}^{m} a_k \frac{t_{jk} + \tau_{ik}}{2}$ et $|\mathbf{t}_j \cup \boldsymbol{\tau}_i| = \sum_{k=1}^{m} \frac{t_{jk} + \tau_{ik}}{2}$ où $a_k = 1$ if $t_{jk} \cdot \tau_{ik} \neq 0$, 0 sinon. Intuitivement, l'intersection est égale à la moyenne entre le nombre d'objets en commun, et la somme des poids de leurs centroïdes. L'union est la moyenne entre le nombre d'objets et la somme des poids de leurs centroïdes. Ces mesures sont définies de façon similaire sur les propriétés.

Les objets t_j (resp. les propriétés g_j) sont assignés à l'un des K clusters (noté i) pour lequel τ_{ij} (resp. γ_{ij}) est maximum. Il est possible de décider que des objets et/ou propriétés puissent appartenir à plus d'un bi-cluster en contrôlant le niveau de recouvrement pour chaque bi-cluster. Grâce à notre définition d'appartenance à un cluster, déterminée par la valeur de $\boldsymbol{\tau}_i$ et $\boldsymbol{\gamma}_i$, il suffit d'adapter l'étape d'assignation des clusters en utilisant un seuil défini par l'utilisateur. L'algorithme CDK-MEANS simplifié par rapport à (Pensa *et al.*, 2005) est rappelé dans la Table 2 (e.g., recouvrement non considéré). Il calcule une bi-partition pour un jeu de données $\mathbf{r}$ à partir d'une collection de bi-ensembles $\mathcal{B}$ fournie (e.g., des concepts formels dans $\mathbf{r}$). Dans notre exemple jouet, CDK-MEANS va pouvoir calculer la bi-partition donnée en Section 1.

3.2 Intégration des contraintes dans L2G

Nous proposons une extension significative du cadre L2G quand des contraintes définies par l'utilisateur sont spécifiées. L'idée centrale est que, pour calculer une bi-partition qui satisfait une contrainte globale, nous pouvons partir d'une collection de motifs locaux qui ne violent pas une représentation locale de cette contrainte. En utilisant cette contrainte au niveau local (éventuellement associée à une stratégie de propagation), l'idée est qu'il soit être possible d'obtenir une bi-partition qui va satisfaire la contrainte au niveau global. Il faut remarquer que, étant donné l'état de l'art en extraction de bi-ensembles sous contraintes, il existe des algorithme efficaces pour une

CDK-MEANS (**r** est un contexte booléen, $\mathcal{B}$ est une collection de bi-ensembles de **r**, K est le nombre de bi-clusters, et MI est le nombre maximal d'itérations.)

1. Soient $\mu_1 \dots \mu_K$ les centroïdes initiaux. $it := 0$.
2. Répéter
   (a) Pour chaque bi-ensemble $b_i \in \mathcal{B}$, l'affecter au cluster $P_k^{\mathcal{B}}$ tel que $d(b_i, \mu_k)$ soit minimal.
   (b) Pour chaque cluster $P_i^{\mathcal{B}}$, calculer $\boldsymbol{\tau_i}$ et $\boldsymbol{\gamma_i}$.
   (c) $it := it + 1$.
3. Jusqu'a ce que le centroïde ne change pas ou bien que $it = MI$.
4. Pour chaque $t_j \in \mathcal{T}$ (resp. $g_j \in \mathcal{G}$), l'affecter au premier cluster $P_i^{\mathcal{T}}$ (resp. $P_i^{\mathcal{G}}$) tel que τ_{ij} (resp. γ_{ij}) soit maximal
5. Renvoyer $\{P_1^{\mathcal{T}} \dots P_K^{\mathcal{T}}\}$ et $\{P_1^{\mathcal{G}} \dots P_K^{\mathcal{G}}\}$

TAB. 2 – Pseudo-code de CDK-MEANS

grande classe de contraintes. Par exemple, dans nos applications, nous utilisons D-MINER (Besson *et al.*, 2005) pour calculer des collections complètes de concepts formels satisfaisant des contraintes définies par l'utilisateur, e.g., des contraintes de taille minimale.

Discutons maintenant de la réutilisation du principe de CDK-MEANS et donc des possibilités de traduction des contraintes globales en contraintes locales, et, si nécessaire, comment propager l'information capturée au niveau local jusqu'au niveau global.

Pour traiter des contraintes "must-link", il est possible de forcer cette contrainte dans la collection de bi-ensembles utilisée. En particulier, étant donnée une contrainte "must-link" étendue entre un objet/propriété x_i et un objet/propriété x_j, une collection de bi-ensembles $\mathcal{B}$ satisfait potentiellement la contrainte ssi $\forall b = (T, G) \in \mathcal{B}$, si $x_i \in T$ (ou G) alors $x_j \in T$ (or G), et vice versa. Comme le coefficient de chaque objet/propriété dans chaque centroïde dépend du nombre de bi-ensembles qui contiennent cet objet/propriété, si x_i et x_j sont impliqués dans une contrainte "must-link" étendue, alors leurs coefficients seront les mêmes dans chaque centroïde. Par suite, le deux seront associé automatiquement au même groupe.

Une condition nécessaire pour une contrainte "cannot-link" étendue est l'exclusion dans $\mathcal{B}$ des bi-ensembles qui violent la contrainte. Puis, étant donnée une contrainte "cannot-link" étendue entre un objet/propriété x_i et un objet/propriété x_j, une collection de bi-ensembles $\mathcal{B}$ satisfait potentiellement la contrainte ssi $\forall b = (T, G) \in \mathcal{B}$, si $x_i \in T$ (ou G) alors $x_j \notin T$ (or G), et vice versa. Cette condition ne peut pas assurer la satisfaction de la contrainte dans la bi-partition finale, et un contrôle ultérieur est nécessaire. En particulier, à l'étape 2a de CDK-MEANS (cf. Table 2), avant d'ajouter le bi-ensemble contenant x_i (resp. x_j) à un cluster, nous devons nous assurer qu'un autre bi-ensemble contenant x_j (resp. x_i) n'a pas été affecté à ce cluster.

Pour traiter des contraintes "interval" et "non-interval", il est possible de forcer la même contrainte dans la collection de bi-ensembles utilisée. Toutefois, dans le cas de

la contrainte “interval”, elle pourrait être trop sélective en pratique. A contrario, dans le cas d’une contrainte “non-interval”, elle pourrait ne pas l’être suffisamment. Pour cette raison, nous proposons de travailler au niveau local sur une contrainte “interval” relaxée et une contrainte “non-interval” renforcée. Étant donné un ordre sur $\mathcal{D}$, une contrainte “max-gap” sur cette dimension, notée $\mathcal{C}_{maxgap}(\mathcal{D}, l, b)$, est satisfaite ssi, pour chaque paire d’éléments consécutifs $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \leq l$. Dualement, une contrainte “min-gap” sur $\mathcal{D}$, notée $\mathcal{C}_{mingap}(\mathcal{D}, l, b)$, est satisfaite ssi, pour chaque paire d’éléments consécutifs $x_i, x_j \in b$, $x_i \prec x_j$, $|\{x_h \notin b | x_i \prec x_h \prec x_j\}| \geq l$. La première est utilisée pour le traitement local de “interval” et la seconde aide au traitement de “non-interval”. Clairement, la satisfaction des contraintes sur la bi-partition résultat n’est pas assurée mais le comportement en phase de calcul est satisfaisant (cf. Section 4). Il ne s’agit ici que d’une première étape et les stratégies de propagation doivent être étudiées.

Lorsque l’on dispose d’une collection de bi-ensembles extraite dans les données, il est toujours possible de sélectionner par post-traitement une sous-collection de ces motifs pour exploiter des contraintes au niveau local. On peut aussi considérer le calcul des motifs locaux utiles à une classification sous contrainte donnée, et ainsi exploiter les techniques efficaces d’extraction de motifs sous contraintes. En effet, les propriétés duales bien connues de monotonicité et d’anti-monotonicité vis-à-vis des relations de spécialisation permettent d’exploiter de nombreuses contraintes, notamment sur les bi-ensembles (voir, e.g., (Besson *et al.*, 2005)). Par suite, si les contraintes au niveau local sont des conjonctions/disjonctions de contraintes (anti-)monotones, elle peuvent être poussées jusque dans la phase d’extraction des bi-ensembles.

Considérons d’abord la contrainte d’inclusion. Soit $x_i \in \mathcal{D}$, un ensemble $X \subseteq \mathcal{D}$ satisfait une contrainte d’inclusion $\mathcal{C}_{incl}(x_i, X)$ ssi $x \in X$. Cette contrainte est monotone et, par suite, la version locale des contraintes “must-link” et “cannot-link” peut être représentée par une conjonction et/ou une disjonction de contraintes (anti-)monotones. En effet, une version locale pour “must-link” $\mathcal{C}_{eml}(x_i, x_j)$ peut être réécrite comme $(\mathcal{C}_{incl}(x_i) \wedge \mathcal{C}_{incl}(x_j)) \vee (\neg\mathcal{C}_{incl}(x_i) \wedge \neg\mathcal{C}_{incl}(x_j))$. Pour la contrainte “cannot-link” $\mathcal{C}_{ecl}(x_i, x_j)$, elle peut être réécrite comme $\neg\mathcal{C}_{incl}(x_i) \vee \neg\mathcal{C}_{incl}(x_j)$. Cette contrainte est anti-monotone puisqu’elle est la disjonction de deux contraintes anti-monotones.

La contrainte “min-gap” est anti-monotone. En effet, soit $b_1 = (X_1, Y_1)$ et $b_2 = (X_2, Y_2)$, tels que $X_1 \subseteq X_2$. On a $S_2 = \{x_h \notin X_2 | x_i \prec x_h \prec x_j\} \subseteq S_1 = \{x_h \notin X_1 | x_i \prec x_h \prec x_j\}$. Par suite, si $|S_2| \geq l$, alors $|S_1| \geq l$. Par contre, la contrainte “max-gap” n’a pas de propriété de monotonicité par rapport à l’inclusion ensembliste. En effet, pour une dimension $D = \{x_1, x_2, \ldots, x_n\}$, une contrainte “max-gap” $\mathcal{C}_{maxgap}(D, 1)$, n’est pas satisfaite par l’ensemble $X_1 = \{x_2, x_3, x_7\}$. Cependant, elle l’est par son sur-ensemble $X_2 = \{x_2, x_3, x_5, x_7\}$, et par son sous-ensemble $X_0 = \{x_2, x_3\}$. Elle n’est donc ni monotone ni anti-monotone[1].

[1] Elle peut cependant être exploitée dans la phase de génération des candidats.

4 Application à des données d'expression des gènes

4.1 Motivations

L'un des problèmes majeurs en biologie concerne l'étude de l'évolution des cellules des organismes uni/pluri-cellulaires. Cela permet, par exemple, de comprendre les mécanismes qui se cachent derrière la différenciation cellulaire, et c'est un des défis le plus excitants en biologie moléculaire. Un moyen d'étudier l'évolution d'un organisme, consiste à analyser la variation de l'expression de ses gènes tout au long de son cycle de vie. Dans les phases initiales du cycle, chaque étape est caractérisée par le développement d'un nombre de fonctions particulières, et par conséquence par l'interaction de différentes protéines. À travers l'étude du niveau d'expression des gènes qui interagissent dans le différentes phases du développement, il est possible de les associer à des fonctions biologiques putatives. Le profil d'expression typique d'un gène durant le cycle de vie de la cellule, a tendance à avoir des pics dans les étapes où ils sont "allumés", et ils sont généralement sous-exprimés dans la partie restante du cycle. Pendant la transition d'un échantillon temporel au suivant, il y a des groupes de gènes qui deviennent sous-exprimés et d'autres qui deviennent sur-exprimés.

Un biologiste peut rechercher les gènes co-régulés et la (co-)classification est l'une des techniques phares pour cet objectif. Toutefois, un inconvénient majeur vient de ce que les étapes du développement ne sont pas toujours identifiées par les algorithmes de classification utilisés. Les bi-clusters peuvent contenir échantillons issus des différentes étapes du développement, ou bien, ils ne constituent pas un ensemble d'échantillons temporels consécutifs. Si la partition calculée sur la dimension des conditions expérimentales ne correspond pas aux phases réelles du développement, les clusters des gènes ne coïncident pas non plus avec les vrais groupes fonctionnels. La phase d'interprétation biologique devient très difficile. D'un autre côté, la relation temporelle entre les ensembles de conditions biologiques est souvent très forte, et de nombreux algorithmes de classification peuvent renvoyer des intervalles de temps parfaits. Si maintenant, notre biologiste veut découvrir les interactions entre les gènes appartements à des phases différentes du cycle de vie, il/elle va devoir chercher une bi-partition qui cherche à s'écarter des groupements par phases du développement.

Nous allons présenter l'utilisation des contraintes "interval" et "non-interval" dans de tels problèmes d'analyse de données d'expression. Nous montrons que l'utilisation de ces contraintes dans notre cadre exploitant des motifs locaux fournit des résultats plus pertinents.

4.2 Méthode d'évaluation

Un critère général pour évaluer les résultats d'une classification consiste à comparer la partition calculée avec une partition "correcte". Cela signifie que les instances des données sont déjà associées à des étiquettes jugées correctes, et que l'on va pouvoir quantifier la conformité entre étiquettes calculées et correctes. Une mesure classique est l'indice de Rand (Jain & Dubes, 1988) pour évaluer la conformité entre deux partitions de m éléments.

Si $\mathbf{C} = \{C_1 \dots C_s\}$ est la structure issue de la classification et que $\mathbf{P} = \{P_1 \dots P_t\}$ est une partition prédéfinie, chaque paire de points peut être affecté au même cluster ou à deux clusters différents. Soit a le nombre de paires appartenant au même cluster de $\mathbf{C}$ et au même cluster de $\mathbf{P}$. Soit b le nombre de paires dont les points appartiennent à deux clusters différents de $\mathbf{C}$ et à deux clusters différents de $\mathbf{P}$. La conformité entre $\mathbf{C}$ et $\mathbf{P}$ peut être estimée au moyen de la formule :

$$Rand(\mathbf{C}, \mathbf{P}) = \frac{a+b}{m \cdot (m-1)/2}$$

Cet indice prends des valeurs entre 0 et 1 et il est maximisé quand $s = t$. Nous utilisons cet indice de Rand pour calculer la précision dans nos expériences.

Pour évaluer la valeur ajoutée de la contrainte "interval", nous mesurons le nombre de sauts à l'intérieur d'une partition.

Définition 1 (nombre de sauts)
Soient $\mathcal{D} = \{x_1, \dots, x_n\}$ un ensemble ordonné de points et $P_k^{\mathcal{D}}$ un cluster sur ces points, il y a un saut lorsque pour un nombre $l > 1$, si $x_i \in C$, $x_{i+l} \in P_k^{\mathcal{D}}$ et $\forall h$ tel que $i < h < i + l$, $x_h \notin P_k^{\mathcal{D}}$. Soit J_k le nombre de sauts dans un cluster $P_k^{\mathcal{D}}$. Etant donnée une partition $\mathcal{P}^{\mathcal{D}} = \{P_1^{\mathcal{D}}, \dots, P_K^{\mathcal{D}}\}$, le nombre de sauts noté N_J est donc

$$N_J = \sum_{\forall P_k^{\mathcal{D}} \in \mathcal{P}^{\mathcal{D}}} J_k$$

.

Lorsque $N_J = 0$, les clusters sont exactement des intervalles. Comme la contrainte "interval" est une contrainte souple, nous calculons la moyenne des N_J sur un ensemble d'instances de classifications (avec une initialisation aléatoire) pour mesurer l'efficacité de l'approche.

Nous voulons aussi évaluer la qualité intrinsèque de la co-classification au moyen d'un critère interne. Pour ce faire, une mesure intéressante est le coefficient τ symétrique de Goodman and Kruskal's (Goodman & Kruskal, 1954). Il est évalué dans une table de contingence $\mathbf{p}$ et il discrimine correctement les bi-partitions par rapport à l'intensité du lien fonctionnel entre leurs deux partitions (Robardet & Feschet, 2001). Soit p_{ij} la fréquence des relations entre an objet d'un cluster C_i^o et une propriété d'un cluster C_j^p, et $p_{i.} = \sum_j p_{ij}$ et $p_{.j} = \sum_i p_{ij}$. Nous utilisons le coefficient $\tau_{\mathcal{S}}$ qui évalue la réduction proportionnelle de l'erreur entraînée par la connaissance de C^o sur la prédiction de C^p et vice versa. Il est défini de la manière suivante :

$$\tau_{\mathcal{S}} = \frac{\frac{1}{2}\sum_i \sum_j \left(p_{ij} - p_{i.}p_{.j}\right)^2 \frac{p_{i.}+p_{.j}}{p_{i.}p_{.j}}}{1 - \frac{1}{2}\sum_i p_{i.}^2 - \frac{1}{2}\sum_j p_{.j}^2}$$

Enfin, pour évaluer les performances de notre méthode, nous utilisons le coefficient de comparaisons, défini par la moyenne des produits du nombre d'itérations nécessaires pour compléter la classification, et le nombre des bi-ensembles, i.e. :

$$CC = \frac{\sum_i^N |\mathcal{B}| \cdot NI_i}{N}$$

	COCLUSTER			CDK-MEANS			
Dataset	N_J	$Rand$	$\tau_{\mathcal{S}}$	N_J	$Rand$	$\tau_{\mathcal{S}}$	CC
malaria	0.85	0.761	0.494	0.3	0.877	0.438	3.063M
drosophila	6.39	0.692	0.513	4.29	0.601	0.424	1.652M

TAB. 3 – Résultats d'une co-classification sans contrainte.

où, N indique le nombre d'exécutions, $|\mathcal{B}|$ est la taille de la collection de bi-ensembles, et NI_i est le nombre d'itérations à la i^{eme} exécution.

4.3 Utilisation de la contrainte "interval"

Nous avons étudié l'impact de la contrainte "interval" dans deux jeux de données Puces ADN, malaria et drosophila. Le premier (Bozdech *et al.*, 2003) concerne le transcriptome du cycle de développement intraerythrocytique du Plasmodium Falciparum, i.e., un agent responsable de la malaria humaine. Les données fournissent le profil d'expression de 3 719 gènes dans 46 échantillons biologiques. Chaque échantillon correspond à un instant de temps du cycle de développement : il commence avec l'invasion des cellules rouges du sang par le mérozoïte, et il est divisé en trois phases : anneau, trophozoïte et schizonte (concernant respectivement le moustique, le foie, et le sang). Après 48 heures, le cellule se réplique et se divise. En correspondance de l'instant 17h et 29h il y a deux transitions brusques. Le second jeu de données est décrit dans (Arbeitman *et al.*, 2002). Il concerne l'expression des gènes de la Drosophile melanogaster durant son cycle de vie. Les niveaux d'expression de 3 944 gènes sont mesurés pour 57 périodes séquentielles de temps divisé en stade embryonnaire, larvaire et pupaire. Les données d'expression numériques présentées dans (Bozdech *et al.*, 2003) ont été discretisées en utilisant une des méthodes de codage des propriété décrites dans (Becquet *et al.*, 2002) : pour chaque gène g, nous avons affecté la valeurs booléenne 1 aux échantillons dont le niveau d'expression était supérieur à X% de la valeur maximale. Nous avons choisi X=25% pour malaria et X=35% pour drosophila. Pour les motifs locaux, nous avons utilisé D-MINER (Besson *et al.*, 2005) pour extraire les concepts formels dans les deux matrices booléennes dérivées.

Nous avons d'abord appliqué l'algorithme COCLUSTER (Dhillon *et al.*, 2003) et la version sans contraintes de CDK-MEANS avec $K = 3$ (i.e., avec l'idée d'identifier les trois stades du développement). L'initialisation des deux algorithmes étant aléatoire, nous avons calculé la moyenne de toutes les mesures sur 100 exécutions. Nous avons mesuré le coefficient N_J, l'indice de Rand par rapport au partitionnement réel disponible dans la littérature, et le coefficient de Goodman-Kruskal pour évaluer la qualité intrinsèque de la bi-partition. Les résultas sont dans la Table. 3.

Il y a une différence significative entre les deux jeux de données. Dans malaria, le nombre moyen de sauts (N_J) est déjà petit avec les deux algorithmes. En particulier, si COCLUSTER obtient un bon coefficient de Goodman-Kruskal, les bi-clusters obtenus avec CDK-MEANS sont plus cohérents avec la connaissance biologique disponible (i.e., la partition à un indice de Rand élevé). D'un autre coté, le nombre de comparaisons

est plutôt élevé. Ce que nous attendons dans un tel cas, c'est qu'une approche basée sur les contraintes utilise moins de ressources pour des résultats similaires.

Au contraire, pour Drosophila, les deux algorithmes échouent dans la découverte du partitionnement correct au regard de la connaissance disponible. Le nombre de sauts est dans le deux cas élevé alors que l'indice de Rand est relativement petit. Dans un tel contexte, nous souhaitons obtenir de meilleurs résultats avec une approche basée sur les contraintes.

Nous avons utilisé la contrainte "interval" sur les conditions expérimentales. Nous avons appliqué différents niveaux de la contrainte "max-gap", et nous avons étudié l'impact sur la partition finale en mesurant le coefficient N_J, l'indice de Rand, le coefficient de Goodman-Kruskal, et le nombre moyen de comparaisons. Les résultats sont en Figure 1 et Figure 4 (resp. pour malaria et drosophila).

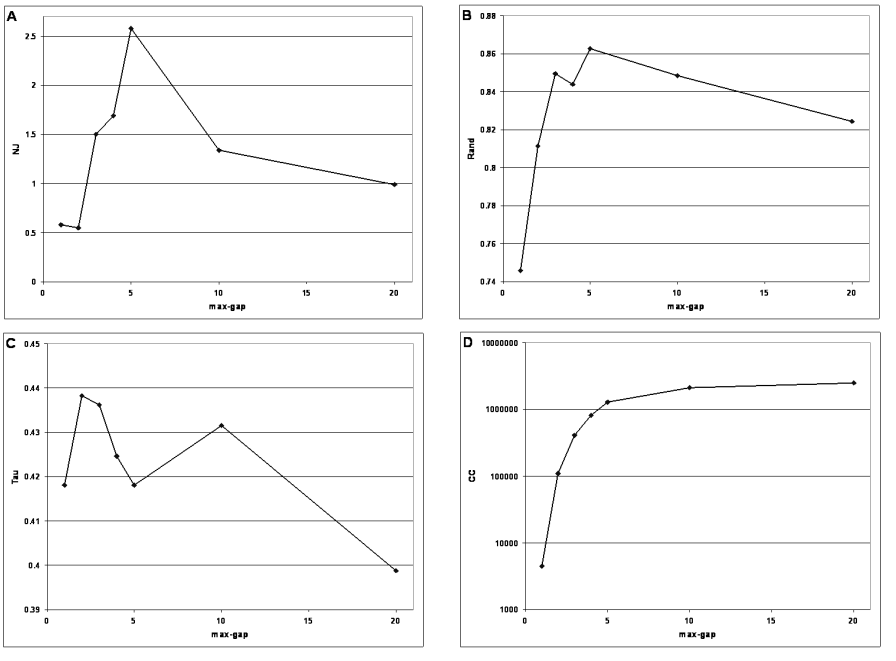

FIG. 1 – Résultats pour malaria.

Pour malaria, nous observons les meilleurs résultats en terme de nombre de sauts (cf. Fig. 1a) pour une contrainte "max-gap" de 1 et 2. Dans le second cas (max-gap=2), l'indice de Rand (Fig. 1b) est plus élevé, et le coefficient de Goodman-Kruskal (Fig. 4c) est maximal (et similaire à celui obtenu sans contraintes, cf Tab. 3). Les nombres moyens des comparaisons pour ces valeurs de "max-gap" sont réduits sensiblement (d'un facteur 8, pour max-gap=3, jusqu'à 28 pour max-gap=2). Quand la valeur du max-gap est 1, le nombre moyen de comparaisons est environ 1/1000 de celui obtenu sans la spécification d'aucune contrainte. Quand max-gap=5, nous avons obtenu un coefficient

N_J plutôt élevé, mais l'indice de Rand est max et similaire à celui que nous obtenions sans contraintes. Dans ce cas, un choix optimal semble être max-gap=2. Il réduit sensiblement le temps de calcul, et produit de bons résultats de classification.

Notons également que notre définition de contrainte "max-gap" fonctionne pour des intervalles de temps ouverts. Cependant, le cycle de développement cellulaire du plasmodium est circulaire. En sélectionnant une contrainte d'intervalle ouvert, nous somme toujours en mesure d'obtenir une séquence circulaire d'intervalles.

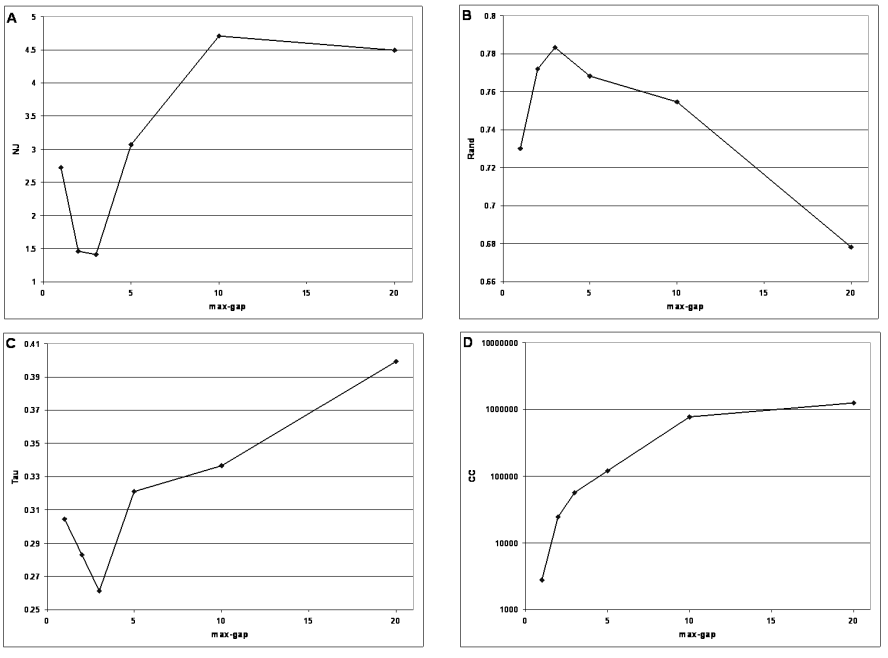

FIG. 2 – Résultats pour drosophila.

Pour drosophila, l'amélioration est plus évidente. Les résultats d'une co-classification sans contrainte montrent qu'une bonne partition (avec un coefficient de Goodman-Kruskal élevé) contient beaucoup de sauts. Avec une contrainte max-gap de 2 ou 3, nous pouvons réduire significativement le nombre de sauts (Fig. 4a) et augmenter la qualité de la partition (Fig. 4b) par rapport à la connaissance biologique disponible. Le fait que, pour ces valeurs de max-gap, le coefficient de Goodman-Kruskal soit minimum, indique que la partition qui satisfait au mieux les contraintes n'est pas forcément la "meilleure". De plus, le nombre moyen de comparaisons est réduit de 60 (max-gap=2) et 30 (max-gap=3).

4.4 Utilisation de la contrainte "non-interval"

Nous avons montré comment la contrainte "interval" pouvait aider à la découverte d'intervalles temporels. Pour certains jeux de données (e.g., malaria), une approche

		τ		$Rand$		N_J	
bi-part.	**inst.**	**mean**	**std.dev**	**mean**	**std.dev**	**mean**	**std.dev**
co :MF	56	0.5605	0.0381	0.82	0.06	0.25	0.61
co :mixed	44	0.1156	0.0166	0.51	0.02	7.52	2.07
co :overall	100	0.3648	0.2240	0.69	0.16	3.45	3.90
cdk :unconst	100	0.4819	0.0594	0.88	0.04	1.00	0.20
cdk :int	100	0.4609	0.0347	1.00	0.00	0.00	0.00
cdk :nonint	100	0.1262	0.0761	0.53	0.04	6.94	1.93

TAB. 4 – Résultats pour les individus adultes de la drosophile.

sans contrainte produit déjà des intervalles corrects. La question devient : est-il possible de découvrir des associations de gènes différentes qui se produisent entre des points appartenant à des intervalles différents ? Dans quelle mesure les contraintes “interval” et “non-interval” peuvent-elles être utilisées pour guider la tâche de co-classification quand les algorithmes classiques renvoient des résultats instables ?

Pour répondre à ces questions, nous avons appliqué la contrainte “non-interval” aux données concernant les échantillons de la phase adulte du cycle de développement de la drosophile. Les échantillons de temps t_1 à t_{10} concernent les premiers jours du cycle de vie des individus mâles. Les échantillons de t_{11} à t_{20} concernent les individus femelles. Les résultats sont dans la Table 4. Nous avons appliqué les contraintes “interval” et “non-interval” avec respectivement max-gap=5 et min-gap=5.

Quand nous appliquons CDK-MEANS (avec $k = 2$) sans spécifier aucune contrainte sur ce jeu de données, les deux intervalles $t_1, \ldots, t_{10}$ et $t_{11}, \ldots, t_{20}$ sont bien identifiés dans toutes les 100 exécutions de l’algorithme (la valeur moyenne de N_J est 1). COCLUSTER semble être plus instable. Il renvoie parfois un cluster de mâles et un cluster de femelles, parfois les deux sexes sont mélangés. La valeur moyenne de N_J est 3.45, et l’écart type est 113% de la moyenne. Quand nous imposons la contrainte “non-interval”, la valeur moyenne de N_J est élevée (environ 6.94), tandis que l’écart type est plus petit (28% de la moyenne) par rapport aux résultats de COCLUSTER. Quand on impose la contrainte “Interval”, CDK-MEANS renvoie toujours des bi-clusters mâle et femelle parfaits ($N_J = 0$ et $Rand = 1$).

Les résultats montrent qu’au moyen des contraintes “interval” ou “non-interval”, l’utilisateur obtient une forme de contrôle sur la forme de la bi-partition. Sur l’analyse des données concernant les individus drosophiles adultes, un algorithme comme COCLUSTER a parfois trouvé des bi-clusters où le sexe était le paramètre majoritairement discriminant. Parfois il a capturé des interactions entre mâles et femelles. Ce que nous voulons, c’est permettre une supervision de ce processus de classification par la spécification de contraintes.

Nous avons donc montré une certaine valeur ajoutée dans l’utilisation des contraintes pour des analyses de données d’expression temporelles. Les contraintes ont été ap-

pliquées sur la dimension des conditions expérimentales (objets). On peut considérer d'autres applications pour l'analyse de données biologiques et, e.g., considérer un ordre sur la dimension des gènes (ordre spatial des gènes sur les séquences ADN). Beaucoup d'autres applications reposent sur des données ordonnées, par exemple l'analyse de données géospatiales ou encore la fouille de données textuelles.

5 Travaux connexes

La classification sous contraintes est un domaine de recherche relativement nouveau. La plupart des travaux existants l'étudie pour des applications en classification semi-supervisée, i.e., quand les données étiquetées sont limitées et/ou chères à collecter. Une solution est d'utiliser la connaissance apportée par les instances étiquetées à l'intérieur d'un algorithme de classification. Dans (Wagstaff *et al.*, 2001), les auteurs proposent une adaptation simple du K-MEANS qui force des contraintes "must-link" et "cannot-link" lors d'un processus de classificaton mono-dimensionnel. (Basu *et al.*, 2002) propose une approche de classification sous contraintes qui utilise les données étiquetées dans la phase d'initialisation et de classification. Une autre approche de classification semi-supervisée dite basée sur une mesure est présentée dans (Klein *et al.*, 2002). (Bilenko *et al.*, 2004) intègre contraintes et mesure dans un algorithme de type K-MEANS. Ce même objectif d'intégration est présenté dans (Basu *et al.*, 2004) qui propose un modèle probabiliste. D'autres travaux connexes se concentrent sur la faisabilité du traitement des contraintes dans une approche de type K-MEANS (Davidson & Ravi, 2005b), et dans une classification hiérarchique ascendante (Davidson & Ravi, 2005a). Dans ces travaux, l'objectif est d'améliorer la qualité de classifications supervisées (prédictions) quand peu d'instances sont étiquetées. Notre objectif n'est pas d'aider la prédiction. Nous travaillons toujours dans cadre non supervisé où les contraintes sont utilisées pour spécifier des attentes sur la forme des bi-partitions et ainsi exprimer une certaine connaissance du domaine. Notons aussi que, dans le cas des contraintes "interval" et "non-interval", l'utilisateur doit seulement spécifier si il veut ou non des intervalles, sans savoir si un élément particulier x est dans le même groupe qu'un autre élément y. Nous avons aussi réalisé des expériences préliminaires pour tester l'efficacité des contraintes "must-link" et "cannot-link" étendues. Nous avons ainsi montré que, avec quelques contraintes, CDK-MEANS permet d'améliorer la qualité des résultats par rapport à la variable de classe. Rappelons aussi que nos contraintes peuvent être appliquées soit dans la dimension des objets, soit dans celle des propriétés. Pour les contraintes "must-link" et "cannot-link" étendues, il est aussi possible d'impliquer les deux dimensions dans la même contrainte.

Un travail connexe en analyse de données d'expressions est (Sese *et al.*, 2004). C'est un algorithme pour calculer un nombre de clusters spécifié par l'utilisateur qui sont contraints à des motifs locaux maximisant la variance inter-classe. Les motifs locaux sont utilisés aussi pour contraindre la partition. Notons cependant que, dans ce cas, les contraintes ne sont pas spécifiées de manière déclarative. Notre méthode construit une bi-partition qui satisfait des contraintes déclarative définies par l'utilisateur à travers une sélection de motifs locaux.

Des travaux prenant en compte la dimension temporelle des données sont (Zhao &

Zaki, 2005) et (Madeira & Oliveira, 2005). Il s'agit dans le deux cas d'extraire des bi-ensembles dans les données numériques d'expression où une notion de temps est présente. Les deux approches ne calculent donc pas une bi-partition, mais des interactions locales fortes.

6 Conclusion

La co-classification est une approche intéressante en classification conceptuelle. Dans les données catégorielles, il fournit des bi-partitions qui optimisent, au moins localement, des mesures objectives de la qualité des groupements. L'amélioration de la qualité des groupements reste une tâche difficile dans les processus d'analyse exploratoire des données réelles. Premièrement, il est difficile de capturer les aspects d'intérêt subjectif, e.g., l'espérance de l'analyste à partir de sa connaissance du domaine. Puis, quand ces espérances peuvent être spécifiées de façon déclarative, les utiliser durant le processus de calcul est un défi. Pour calculer des bi-partitions qui satisfont des contraintes définies par l'utilisateur, nous avons montré qu'il était possible utiliser un cadre générique de co-classification basé sur les motifs locaux, une approche simple mais puissante. Des nouveaux types de contraintes pour le bi-clusters ont été considérées, e.g., les contraintes "interval" et "non-interval" pour des données ordonnées. Une perspective à court terme pour cette recherche, est de formaliser le propriétés des contraintes globales (i.e., les contraintes pour le bi-partitions) pouvant être transformées, de façon plus ou moins automatique, dans des contraintes à niveau local. Il faut également étudier des stratégies de propagation de contraintes depuis le niveau local jusqu'au niveau global et ainsi garantir la satisfaction des contraintes fixées par l'analyste (hors contrainte d'optimisation de la fonction objectif) dans les bi-partitions calculées.

Remerciements. Ce travail a été partiellement financé par l'ACI Masse de Données Bingo "Bases de données inductives pour la génomique" (ACI MD 46, CNRS STIC) et le contrat Européen IQ "Inductive Queries for Mining Patterns and Models" FP6-516169 (Bras FET du programme IST).

Références

ARBEITMAN M., FURLONG E., IMAM F., JOHNSON E., NULL B., BAKER B., KRASNOW M., SCOTT M., DAVIS R. & WHITE K. (2002). Gene expression during the life cycle of drosophila melanogaster. *Science*, **297**, 2270–2275.

BASU S., BANERJEE A. & MOONEY R. J. (2002). Semi-supervised clustering by seeding. In *Proceedings ICML 2002*, p. 27–34, Sydney, Australia : Morgan Kaufmann.

BASU S., BILENKO M. & MOONEY R. J. (2004). A probabilistic framework for semi-supervised clustering. In *Proceedings ACM SIGKDD 2004*, p. 59–68, Seattle, USA : ACM Press.

BECQUET C., BLACHON S., JEUDY B., BOULICAUT J.-F. & GANDRILLON O. (2002). Strong association rule mining for large gene expression data analysis : a case study on human SAGE data. *Genome Biology*, **12**.

BESSON J., ROBARDET C., BOULICAUT J.-F. & ROME S. (2005). Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis*, **9(1)**, 59–82.

BILENKO M., BASU S. & MOONEY R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings ICML 2004*, p. 81–88, Banff, Canada : ACM Press.

BOZDECH Z., LLINÁS M., PULLIAM B. L., WONG E., ZHU J. & DERISI J. (2003). The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum. *PLoS Biology*, **1**(1), 1–16.

DAVIDSON I. & RAVI S. S. (2005a). Agglomerative hierarchical clustering with constraints : Theoretical and empirical results. In *Proceedings PKDD 2005*, volume 3721 of *LNCS*, p. 59–70, Porto, Portugal : Springer.

DAVIDSON I. & RAVI S. S. (2005b). Clustering with constraints : Feasibility issues and the k-means algorithm. In *Proceedings SIAM SDM 2005*, Newport Beach, USA.

DHILLON I. S., MALLELA S. & MODHA D. S. (2003). Information-theoretic co-clustering. In *Proceedings ACM SIGKDD 2003*, p. 89–98, Washington, USA : ACM Press.

FISHER D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, **2**, 139–172.

GOODMAN L. A. & KRUSKAL W. H. (1954). Measures of association for cross classification. *Journal of the American Statistical Association*, **49**, 732–764.

JAIN A. & DUBES R. (1988). *Algorithms for clustering data*. Englewood cliffs, New Jersey : Prentice Hall.

KLEIN D., KAMVAR S. D. & MANNING C. D. (2002). From instance-level constraints to space-level constraints : Making the most of prior knowledge in data clustering. In *Proceedings ICML 2002*, p. 307–314, Sydney, Australia : Morgan Kaufmann.

MADEIRA S. C. & OLIVEIRA A. L. (2004). Biclustering algorithms for biological data analysis : A survey. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **1**(1), 24–45.

MADEIRA S. C. & OLIVEIRA A. L. (2005). A linear time biclustering algorithm for time series gene expression data. In *Proceedings WABI 2005*, volume 3692 of *LNCS*, p. 39–52, Mallorca, Spain : Springer.

MANNILA H. (1997). Inductive databases and condensed representations for data mining. In *Proceedings ILPS'97*, p. 21–30, Port Jefferson, USA : MIT Press.

PENSA R. G., ROBARDET C. & BOULICAUT J.-F. (2005). A bi-clustering framework for categorical data. In *Proceedings PKDD 2005*, volume 3721 of *LNAI*, p. 643–650, Porto, Portugal : Springer-Verlag.

ROBARDET C. & FESCHET F. (2001). Efficient local search in conceptual clustering. In *Proceedings DS'01*, volume 2226 of *LNCS*, p. 323–335, Washington, USA : Springer-Verlag.

SESE J., KUROKAWA Y., MONDEN M., KATO K. & MORISHITA S. (2004). Constrained clusters of gene expression profiles with pathological features. *Bioinformatics*, **20**(17), 3137–3145.

WAGSTAFF K., CARDIE C., ROGERS S. & SCHRÖDL S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings ICML 2001*, p. 577–584, Williamstown, USA : Morgan Kaufmann.

ZHAO L. & ZAKI M. J. (2005). Tricluster : An effective algorithm for mining coherent clusters in 3d microarray data. In *Proceedings ACM SIGMOD 2005*, p. 694–705, Baltimore, Maryland, USA : ACM Press.