Clustering de séquences d'évènements temporels

Romain Guigourès**, Dominique Gay*, Marc Boullé*, Fabrice Clérot*

*Orange Labs
prenom.nom@orange.com,
**Zalando
prenom.nom@zalando.de

Résumé. Nous proposons une nouvelle méthode de clustering et d'analyse de séquences temporelles basée sur les modèles en grille à trois dimensions. Les séquences sont partitionnées en clusters, la dimension temporelle est discrétisée en intervalles et la dimension évènement est partitionnée en groupes. La grille de cellules 3D forme ainsi un estimateur non-paramétrique constant par morceaux de densité jointe des séquences et des dimensions des évènements temporels. Les séquences d'un cluster sont ainsi groupés car elles suivent une distribution similaire d'évènements au cours du temps. Nous proposons aussi une méthode d'exploitation du clustering par simplification de la grille ainsi que des indicateurs permettant d'interpréter les clusters et de caractériser les séquences qui les composent. Les expériences sur des données artificielles ainsi que sur des données réelles issues de DBLP démontrent le bien-fondé de notre approche.

1 Introduction

Les données contenant une information temporelle constituent un défi pour le processus de découverte de connaissances (Yang et Wu, 2006). Les données temporelles sont complexes dans le sens où un objet de la base est décrit par une ou plusieurs séquences d'éléments ordonnés dans le temps. Selon la nature des éléments temporels (catégoriels ou numériques, ponctuels ou continus dans le temps), il existe une grande diversité de méthodes d'extraction de connaissances (Mörchen, 2007). Ici, nous nous intéressons aux données de séquences d'évènements catégoriels et ponctuels, où chaque évènement d'une séquence est associé à un temps t, et que nous appelons simplement séquences d'évènements temporels. La fouille de séquences d'évènements temporels trouve des applications dans de nombreux domaines : e.g., dans le domaine médical, Patnaik et al. (2011) explore des bases de dossiers médicaux électroniques de patients à la recherche de motifs d'évènements temporels fréquents ; dans le domaine du Web, Masseglia et al. (2008) et Saleh et Masseglia (2011) extraient des comportements fréquents d'utilisateurs par période de temps ; en sciences sociales, Studer et al. (2010) cherche à grouper des individus selon leur parcours de vie. La majeure partie des efforts de recherche s'est focalisée sur l'extraction de motifs fréquents dans les données de séquences d'évènements temporels (ou TAS pour "Temporally-Annotated Sequences", voir e.g., (Giannotti et al., 2006)). Dans cet article, nous nous intéressons au problème de clustering de séquences : le but est de créer des groupes de séquences qui partagent des caractéristiques similaires. Dans la

plupart des méthodes de l'état de l'art, il est nécessaire de définir une mesure de (dis)similarité entre séquences ainsi que le nombre de clusters à trouver et d'autres paramètres : e.g., Studer et al. (2010) utilisent l'approche des k-medoids couplée à une distance basée sur les opérations d'insertion-suppression et de substitution (dont les coûts sont à définir) de transitions dans une séquence. Le paramétrage de telle méthode est souvent complexe et peut dépendre du domaine d'application et de la quantité de données dont on dispose. Le choix d'un clustering trop fin (un grand nombre de clusters) ne garantit pas que les clusters sont statistiquement valides et peut mener au sur-apprentissage alors qu'un clustering grossier (peu de clusters) nous apporte une information peu précise sur la structure sous-jacente des données et nous offre ainsi un résumé trop général des données. De plus, la dimension temporelle des séquences d'évènements temporels est primordiale et doit être prise en compte pour grouper des séquences similaires, i.e. qui suivent la même distribution d'évènements au cours du temps.

Notre contribution est la suivante. Nous proposons KHC, une méthode de co-clustering de séquences d'évènements temporels basée sur les modèles en grille (Bondu et al., 2013) : le co-clustering (Dhillon et al., 2003; Nadif et Govaert, 2010) consiste à partitionner simultanément et de manière cohérente les trois dimensions de la base de séquences d'évènements temporels ; ici, les séquences sont partitionnées en clusters, ainsi que les évènements, et le temps est discrétisé en intervalles. Nous en déduisons une mesure de dissimilarité entre clusters nous permettant d'accéder par classification hiérarchique ascendante à la granularité nécessaire à l'analyse. En section 3, nous validons expérimentalement la méthode sur des données synthétiques et réelles et proposons des indicateurs utiles à l'interprétation des clusters révélés par la méthode.

2 Séquences temporelles, modèles en grilles et clustering

Contexte et notations. Une séquence s d'évènements temporels de taille $k > 0$ est un ensemble d'observations ordonnées $s_i = \Big\langle (t_{i_1}, e_{i_1}), (t_{i_2}, e_{i_2}), \ldots, (t_{i_{k_i}}, e_{i_{k_i}}) \Big\rangle$, tel que $\forall j, 1 \leq j \leq i_{k_i}, t_j \in \mathbb{R}+$ et $e_j \in E$ avec E un ensemble non-ordonné d'évènements catégoriels. Une base de données de séquences temporelles est simplement un ensemble de séquences temporelles ainsi définies $\mathcal{D} = \{s_1, \ldots, s_n\}$. Nous proposons de représenter un ensemble de séquences temporelles par une base de données à trois variables (ou dimensions) : S pour les identifiants de séquences, T pour la variable temps et E pour la variable évènement. Dans la suite, un objet (s, t, e) de $\mathcal{D}$ sera appelé un point de la base.

Cette représentation tridimensionnelle des données se prête bien à l'usage des modèles en grilles (Bondu et al., 2013) pour le clustering – plus précisément nous utilisons le cadre de travail MODL (Minimum Optimized Description Length) et la méthode de coclustering KHC [1] déjà instanciée dans le cas des données fonctionnelles (Boullé, 2012). Le but est de partitionner les variables catégorielles (identifiants de séquences et évènements) et de discrétiser la variable numérique "temps". Le résultat est une grille tridimensionnelle dont les cellules sont définies par un groupe d'identifiants de séquence, un groupe d'évènements et un intervalle de temps. Le meilleur modèle M^* (i.e., la grille optimale) est la grille la plus probable connaissant les données. Pour obtenir le modèle de grille optimal M^*, nous utilisons une approche Bayesienne dite Maximum A Posteriori (MAP) et parcourons l'espace des modèles en optimisant un critère

1. Khiops Coclustering : `http://www.khiops.com`

Bayesien, noté $cost$. Le critère $cost$ établit un compromis entre la précision et la robustesse du modèle en grille et est défini comme suit :

$$cost(M) = -\log(\underbrace{p(M \mid D)}_{\text{posterior}}) = -\log(\underbrace{p(M)}_{\text{prior}} \times \underbrace{p(D \mid M)}_{\text{vraisemblance}}) \tag{1}$$

En utilisant un prior hiérarchique (sur les paramètres de la grille) uniforme [2] à chaque étage de la hiérarchie, nous obtenons une expression analytique exacte du critère d'évaluation à optimiser $cost$ (déduit de la même manière que Boullé (2012)) :

Critère d'évaluation. Un modèle de grille (pour le coclustering de séquences temporelles) est optimal s'il minimise le critère $cost$:

$$\begin{aligned} cost(M) = {} & \log n + \log a + \log N + \log B(n, k_S) + \log B(a, k_E) \\ & + \sum_{i_S=1}^{k_S} \log \binom{N_{i_S} + n_{i_S} - 1}{n_{i_S} - 1} + \sum_{i_E=1}^{k_E} \log \binom{N_{i_E} + n_{i_E} - 1}{n_{i_E} - 1} + \log \binom{N + k - 1}{k - 1} \\ & + \log N! - \sum_{i_S=1}^{k_S} \sum_{j_T=1}^{k_T} \sum_{i_E=1}^{k_E} \log N_{i_S j_T i_E}! \\ & + \sum_{i_S=1}^{k_S} \log N_{i_S}! - \sum_{i=1}^{n} \log n_i^S! + \sum_{i_E=1}^{k_E} \log N_{i_E}! - \sum_{i=1}^{a} \log n_i^E! + \sum_{j_T=1}^{k_T} \log N_{j_T}! \end{aligned} \tag{2}$$

où n est le nombre de séquences, a le nombre d'évènements de E, N le nombre total d'évènements temporels (i.e., le nombre de points de la base), k_S (resp. k_E, k_T) le nombre de clusters de séquences (resp. le nombre de clusters d'évènements, le nombre d'intervalles de temps), $k = k_S k_E k_T$ le nombre de cellules de la grille, N_{i_S} (resp. N_{j_T}, N_{i_E}, $N_{i_S j_T i_E}$) est le nombre cumulé de points du cluster de séquences i_S (resp. dans l'intervalle de temps j_T, du cluster d'évènements i_E, de la cellule (i_S, j_T, i_E) de la grille), n_{i_S} (resp. n_{i_E}) le nombre de séquences dans le cluster i_S (resp. le nombre de valeurs d'évènements dans le cluster i_E), et enfin n_i^S (resp. n_i^E) le nombre de points de la séquence i (resp. le nombre de points ayant pour valeur d'évènement i). Notons que $B(n, k_S)$ est le nombre de divisions de n éléments en k_S sous-ensembles et $B(a, k_E)$ est défini de manière similaire.
Les deux premières lignes correspondent à la probabilité a priori du modèle et constituent le terme de régularisation du modèle : les modèles complexes (beaucoup de clusters pour les variables catégorielles et/ou beaucoup d'intervalles pour la variable numérique) seront pénalisés. Les deux dernières lignes correspondent à la vraisemblance du modèle : les modèles les plus proches des données seront préférés ; le cas extrême avec un point par cellule aura une vraisemblance maximale, mais une probabilité a priori très faible et donc une valeur de $cost$ très forte. Une grille avec une faible valeur de $cost$ indique une forte probabilité $p(M \mid D)$ de la grille connaissant les données. En termes de théorie de l'information, le logarithme négatif de probabilités s'interprète comme une longueur de codage. Ainsi, selon le principe MDL (Minimum Description Length), le critère $cost$ peut s'interpréter comme la longueur de codage du

2. L'uniformité se situe à chaque étage de la hiérarchie : le prior n'est donc pas uniforme ; dans ce cas, l'approche MAP n'est pas une simple maximisation de la vraisemblance.

modèle de grille plus la longueur de codage des données connaissant le modèle ; et une faible valeur de $cost$ indique aussi une forte compression des données en utilisant le modèle M.

Le critère $cost$ est optimisé en suivant une stratégie gloutonne ascendante : *(i)* on part de la grille au grain le plus fin, *(ii)* on considère toutes les fusions possibles entre groupes de valeurs ou intervalles, et *(iii)* on réalise la meilleure fusion si le critère $cost$ décroit après fusion. Ce processus est réitéré tant qu'il y a amélioration du critère. La grille obtenue constitue une estimation de la densité jointe des séquences et des dimensions des évènements temporels (i.e., des trois variables S, T et E). Notons que KHC est libre de tout paramètre utilisateur (i.e., nous n'avons pas à choisir le nombre de clusters de séquences ou d'évènements, ni le nombre d'intervalles de temps) ; de plus sa complexité en temps est sub-quadratique : $\Theta(N\sqrt{N}\log N)$ où N est le nombre de points de la base – pour les détails complémentaires voir (Boullé, 2012).

Mesure de dissimilarité et simplification de la structure de grille.
Bien qu'optimale, la grille générée par KHC peut s'avérer trop fine pour une analyse directe par un utilisateur, e.g., plusieurs dizaines de clusters de séquences peuvent être générés. Nous proposons une méthode de simplification de la grille par fusions successives de clusters ou d'intervalles, en choisissant la fusion qui dégrade le moins la qualité de la grille. Pour ce faire, nous introduisons une mesure de dissimilarité entre deux clusters (ou intervalles) qui caractérise l'impact de la fusion sur le critère $cost$.
Soient $c_{.1}$ et $c_{.2}$ deux clusters d'une dimension de la grille M (i.e., deux groupes de valeurs d'identifiants de séquences ou d'évènements, ou deux intervalles contigus de temps). Soit $M_{c_{.1}\cup c_{.2}}$ le modèle de grille après avoir fusionné $c_{.1}$ et $c_{.2}$. La dissimilarité $\Delta(c_{.1}, c_{.2})$ entre deux clusters est définie comme la différence du critère $cost$ après et avant fusion :

$$\Delta(c_{.1}, c_{.2}) = cost(M_{c_{.1}\cup c_{.2}}) - cost(M) \tag{3}$$

Ainsi, si l'on fusionne les clusters qui minimisent Δ, nous obtenons la grille sub-optimale M' (avec un grain plus grossier, i.e., simplifiée) qui dégrade le moins le critère $cost$ et donc avec une perte d'information minimale par rapport à la grille avant fusion. Le taux d'information de la nouvelle grille M' est défini par

$$\tau(M') = (cost(M') - cost(\mathcal{M}_\emptyset))/(cost(M^*) - cost(M_\emptyset)) \tag{4}$$

où $M_\emptyset$ est le modèle nul, i.e., la grille dont aucune dimension n'est partitionnée. En construisant ainsi une hiérarchie ascendante des clusters, en partant de M^* et au pire jusqu'à $M_\emptyset$, l'utilisateur pourra s'arrêter au niveau de grain voulu et nécessaire pour une analyse en contrôlant le nombre de clusters ou le pourcentage d'information gardée. Notons que les fusions s'effectuent indistinctement sur toutes les dimensions en fonction de Δ.

3 Validation expérimentale

Dans cette section nous proposons des expériences sur données simulées afin de démontrer l'efficacité de la méthode en termes de pertinence pour retrouver les motifs simulés dans les données ainsi qu'en terme de temps de calcul pour des données allant jusqu'au million de points. Nous rapportons aussi les résultats de la méthode sur un jeu de données réelles. Les expériences sont réalisées sur un PC de bureau cadencé à 3,8GHz avec 2Go de RAM.

3.1 Données simulées

Exemple à 2 motifs. Considérons deux motifs M_1 et M_2 définis sur le domaine de valeurs de temps $T = [0, 1000] \subseteq \mathbb{R}^+$ et l'ensemble d'évènements $E = \{a, b, c, d, e, f, g, h, i, j, k, l\}$ tels que :

M_1	M_2
si $t \in T_1^{M_1} = [0; 250]$ alors $e \in E_1^{M_1} = \{a, b, c\}$	si $t \in T_1^{M_2} = [0; 100]$ alors $e \in E_1^{M_2} = \{j, k, l\}$
si $t \in T_2^{M_1} =]250; 500]$ alors $e \in E_2^{M_1} = \{d, e, f\}$	si $t \in T_2^{M_2} =]100; 400]$ alors $e \in E_2^{M_2} = \{g, h, i\}$
si $t \in T_3^{M_1} =]500; 750]$ alors $e \in E_3^{M_1} = \{g, h, i\}$	si $t \in T_3^{M_2} =]400; 600]$ alors $e \in E_3^{M_2} = \{d, e, f\}$
si $t \in T_4^{M_1} =]750; 1000]$ alors $e \in E_4^{M_1} = \{j, k, l\}$	si $t \in T_4^{M_2} =]600; 1000]$ alors $e \in E_4^{M_2} = \{a, b, c\}$

Considérons 10 séquences temporelles générées selon le motif M_1 et 10 séquences temporelles selon le motif M_2 (nous menons aussi les mêmes expériences pour 50 et 100 séquences générées par motif). Nous générons une base de données D de 2^{20} points (soit en moyenne plus de 5.10^4 points par séquence). Chaque point est un triplet de valeurs dont un identifiant de séquence choisi aléatoirement (parmi 20), une valeur de temps aléatoire t sur T suivant la loi uniforme et une valeur d'évènement générée en fonction du motif M_i, i.e., une valeur choisie aléatoirement dans l'ensemble $M_i(t)$. Par ailleurs, nous considérons des versions bruitées de cette base à différents niveaux de bruits $\eta = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Lors de la génération d'un point, la probabilité que la valeur d'évènement respecte le motif M_i est $p(e \in M_i(t)) = 1 - \eta$ et $p(e \in \{E \setminus M_i(t)\}) = \eta$.

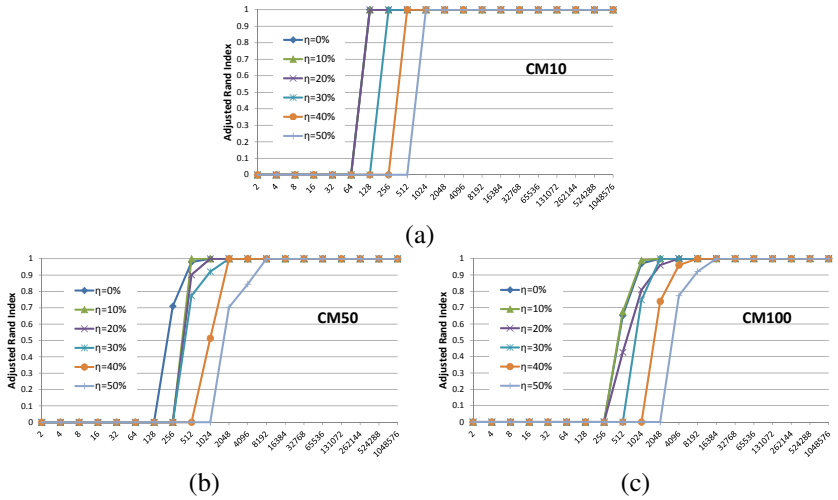

FIG. 1 – *Evolution de l'ARI pour des bases de séquences suivant 2 motifs, pour CM = 10, 50 et 100 séquences par motif et à différents niveaux de bruit en fonction du nombre de points N*

Nous appliquons KHC à des sous-ensembles de D de tailles croissantes, faisant varier ainsi le nombre de points de 2^1 à 2^{20}. Nous calculons la valeur de l'indice de Rand ajusté (ARI) pour chaque grille générée pour évaluer la concordance entre les clusters de séquences trouvés par KHC et les deux motifs sous-jacents. Les résultats sont rapportés dans les figures 1(abc). Nous observons que pour des petits sous-ensembles de données de D, il n'y a pas assez de points

pour que KHC découvre de motifs significatifs : aucun cluster de séquences n'est découvert pour $N \leq 64$ (i.e., en moyenne 3 points par séquences). Pour $CM = 10$ (10 séquences par motifs en figure 1(a)), à partir de $N = 128$ points (soit en moyenne seulement 6 points par séquence), $ARI = 1$ et les deux motifs sous-jacents sont découverts. Nous remarquons aussi que pour un niveau de bruit $\eta \leq 0.1$, $N = 128$ points suffisent encore à trouver les deux clusters de séquences, puis plus le bruit augmente, plus le nombre de points nécessaires à la découverte des deux motifs augmente. Enfin, augmenter le nombre de points jusqu'à 2^{20} ne provoque pas de sur-apprentissage, la valeur de ARI est stable à 1. Les mêmes observations tiennent lorsque $CM = 50$ ou $CM = 100$; nous observons aussi que plus il y a de courbes par motifs (i.e. plus CM est grand), plus il faut de points pour découvrir les deux motifs.

Temps de calcul. La figure 2 rapporte les temps de calcul des différentes versions de bases de données à 2 motifs pour $CM = 10, 50, 100$, en fonction du nombre de points. D'une manière générale, on observe que le temps de calcul augmente, comme attendu, avec le nombre de points d'apprentissage, mais aussi avec CM et le niveau de bruit. Retenons aussi que pour la base la plus *difficile*, i.e., $N = 2^{20}$, $CM = 100$, (soit en moyenne 5200 points par séquence) et $\eta = 0.5$, KHC retrouve les motifs recherchés en moins de 1h30. Ici, le temps de calcul dépend du nombre de valeurs de temps différentes prises dans T (potentiellement 2^{20}), car lors de la discrétisation de T, KHC explore toutes les coupures possibles entre deux valeurs de temps présentes dans les données ; nous avons réalisé les mêmes expériences avec des valeurs de temps *entières* prises dans T : dans ce cas, pour $N = 2^{20}$, $CM = 100$ et $\eta = 0.5$, KHC trouve les motifs recherchés en 13 minutes.

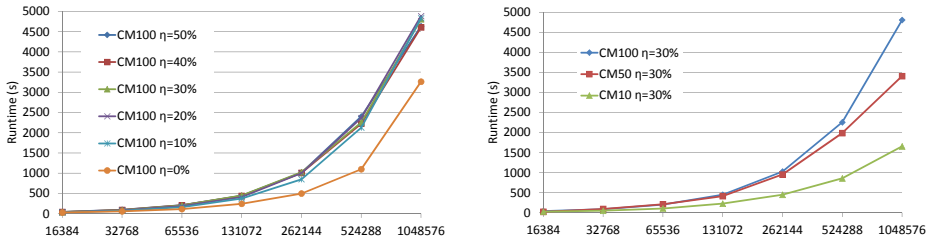

FIG. 2 – *Temps de calcul en fonction du nombre de points, du nombre de séquences par motifs (et donc dans la base) et du niveau de bruit.*

Visualisation et caractérisation des clusters. Considérons la grille tridimensionnelle résultant de KHC sur la base de données à 2 motifs telle que $N = 2^{20}$ points, $CM = 10$ et $\eta = 0.5$. Nous proposons 3 visualisations différentes basées sur la fréquence des cellules, l'information mutuelle et le contraste des cellules – chacune d'entre elles apportant une information différente sur les clusters de séquences découverts. Bien que KHC génère des grilles tridimensionnelles, la dimension S étant partitionnée en deux clusters de séquences, nous proposons des visualisations sur les deux autres dimensions pour chaque cluster de séquences. Nous présentons ces visualisations dans la figure 3.

Visualisation de la fréquence. En figures $3M_1$(a) et $3M_2$(a), la plus classique des visualisations consiste à représenter le nombre de points par cellule, i.e. $N_{i_S j_T i_E}$ pour la cellule (i_S, j_T, i_E). Nous apercevons déjà les cellules les plus fréquentes qui correspondent à la définition des mo-

tifs sous-jacents malgré le niveau de bruit $\eta = 0.5$.
Visualisation de l'information mutuelle. Pour un cluster de séquences c_{i_S}, l'information mutuelle entre les variables T^{π_M} et E^{π_M} issues du partitionnement π_M des variables temps et évènement généré par le modèle de grille M, est défini comme suit :

$$MI(T^{\pi_M}; E^{\pi_M}) = \sum_{i_1=1}^{i_1=k_T} \sum_{i_2=1}^{i_2=k_E} MI_{i_1 i_2} \quad \text{où} \quad MI_{i_1 i_2} = p(c_{i_1 i_2}) \log \frac{p(c_{i_1 i_2})}{p(c_{i_1 .})p(c_{.i_2})} \tag{5}$$

Ainsi, les $MI_{i_1 i_2}$ représentent la contribution de la cellule $c_{i_1 i_2}$ à l'information mutuelle. Si $MI_{i_1 i_2} > 0$, alors $p(c_{i_1 i_2}) > p(c_{i_1 .})p(c_{.i_2})$, et on observe un excès d'interactions entre $c_{i_1 .}$ et $c_{.i_2}$ localisé dans la cellule $c_{i_1 i_2}$ définie par l'intervalle T_{i_1} et le groupe d'évènements E_{i_2}. Inversement, si $MI_{i_1 i_2} < 0$, alors $p(c_{i_1 i_2}) < p(c_{i_1 .})p(c_{.i_2})$, et on observe un déficit d'interactions dans la cellule $c_{i_1 i_2}$. Enfin, si $MI_{i_1 i_2} = 0$, alors soit $p(c_{i_1 i_2}) = 0$ auquel cas la contribution à l'information mutuelle est nulle et il n'y a pas d'interactions ; soit $p(c_{i_1 i_2}) = p(c_{i_1 .})p(c_{.i_2})$ et la quantité d'interactions dans $c_{i_1 i_2}$ est celle attendue en cas d'indépendance des partitions. En figures $3M_1$(b) et $3M_2$(b), nous rapportons les valeurs des $MI_{i_1 i_2}$. Les cellules des deux motifs sous-jacents apparaissent clairement en rouge en raison d'une forte contribution à l'information mutuelle alors que les points issus du bruit sont mis en évidence par les cellules en bleu.

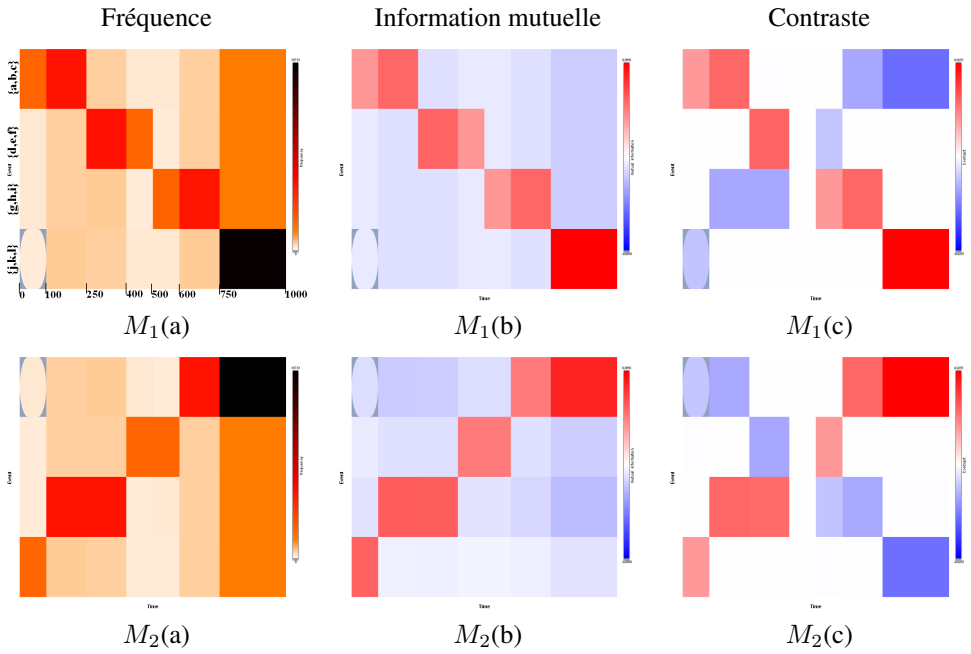

FIG. 3 – *Visualisation de la fréquence, de l'information mutuelle conditionnelle à un cluster de séquences et du contraste pour les 2 clusters trouvés par* KHC *correspondant aux 2 motifs sous-jacents M_1 et M_2. En abscisses, la discrétisation du temps en 7 intervalles et en ordonnées, la partition des évènements en 4 groupes : $E = \{a, b, c\} \cup \{d, e, f\} \cup \{g, h, i\} \cup \{j, k, l\}$.*

Visualisation du contraste. Pour le couple de variables partitionnées à visualiser (T^{π_M}, E^{π_M}), le contraste entre un contexte, i.e., un cluster de séquences c_{i_S} et le reste des clusters de séquences $\{c_i\}_{i \neq i_S}$ est défini comme suit :

$$Contraste(c_{i_S}) = MI((T^{\pi_M}, E^{\pi_M}); c_{i_S}) = \sum_{i_1=1}^{i_1=k_T} \sum_{i_2=1}^{i_2=k_E} MI_{i_1 i_2; i_S} \quad (6)$$

$$\text{où} \quad MI_{i_1 i_2; i_S} = p(c_{i_1 i_2 i_S}) \log \frac{p(c_{i_1 i_2 i_S})}{p(c_{i_1 i_2 .}) p(c_{..i_S})}$$

Comme précédemment, le signe des $MI_{i_1 i_2; i_S}$ qualifiera le contraste entre c_{i_S} et le reste des données (i.e., les clusters de séquences $\{c_i\}_{i \neq i_S}$). Les valeurs de $MI_{i_1 i_2; i_S}$ sont rapportées dans les figures $3M_1$(c) et $3M_2$(c). Prenons le cluster de séquences de c_{M_1}. Les cellules blanches indiquent qu'il n'y a pas de contraste à cet endroit entre c_{M_1} et le reste des données (ici, c_{M_2}) : par exemple la cellule $([0; 100], \{g, h, i\})$, malgré le bruit, n'est pas caractéristique de c_{M_1} ; en effet, la probabilité du groupe d'évènements $\{g, h, i\}$) dans l'intervalle de temps $([0; 100]$ n'est pas significativement différent selon qu'on se trouve dans c_{M_1} ou c_{M_2}. De même la cellule $([401; 500], \{d, e, f\})$ présente un contraste nul puisqu'elle est commune aux deux motifs sous-jacents. Les cellules rouges indiquent ce qui caractérise c_{M_1} par rapport à c_{M_2} : dans ces cellules, la probabilité de points est bien supérieure pour c_{M_1} que pour c_{M_2}. Les cellules bleues indiquent un contraste négatif : la probabilité de points y est plus faible pour c_{M_1} que pour c_{M_2}.

3.2 Données réelles

A partir de la base de données DBLP (Ley, 2009), nous considérons tous les auteurs qui ont publié des articles parus dans les actes de neuf conférences dont la thématique première est les bases de données et/ou la fouille de données (CIKM, VLDB, SIGMOD, ICDE, ICDM, KDD, SDM, PAKDD, PKDD). Pour chaque auteur, nous considérons l'année de publication et l'évènement lié à la publication (i.e., le nom de la conférence). Nous constituons ainsi une base de données de séquences d'évènements temporels à trois dimensions (auteur, année, évènement). Les points de la base sont dupliqués lorsqu'un auteur a publié plusieurs fois dans la même conférence la même année.

La base D ainsi constituée est composée de plus de 21000 auteurs qui ont publié de 1975 à 2012 dans neuf conférences distinctes – en tout 58547 points. KHC calcule la grille optimale pour D en 215 secondes (soit moins de 4 minutes). La grille optimale est constituée de 9 groupes d'auteurs, 15 intervalles de temps et 9 groupes d'évènements (une conférence par cluster).

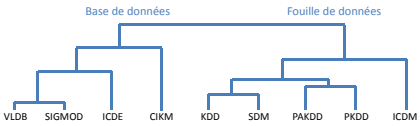

FIG. 4 – *Dendrogrammme des conférences*

En utilisant la mesure de dissimilarité Δ pour les clusters d'évènements, nous observons que les conférences orientées *fouille de données* (FD) sont plus similaires entre elles et il en est de même pour les conférences orientées *Base de données* (BD) ; aussi, nous obtenons le dendrogramme des conférences en figure 4. Comme attendu, pour une résolution à 2 clusters, on identifie bien les conférences orientées *Base de données* et *Fouille de Données*.

La granularité de la grille optimale est acceptable pour une analyse sans simplification ; toutefois pour des raisons de limitation de pages, en figure 6, nous détaillons les résultats pour 4 des 9 neuf clusters d'auteurs que nous identifions comme des clusters d'auteurs spécialisés en fouille de données. Bien que la grille soit constituée de 9 clusters d'auteurs, ces clusters contiennent plusieurs centaines (voire milliers) d'auteurs. De plus, la plupart des auteurs ont très peu publié (1 ou 2 fois voir figure 5) dans les conférences de la base. Pour analyser chacun des clusters nous proposons une mesure pour identifier les auteurs les plus typiques d'un cluster :

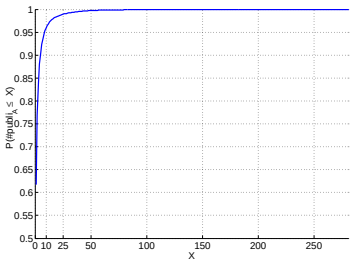

FIG. 5 – *Distribution cumulée empirique du nombre d'auteurs ayant publié X fois, soit $p(\#publi_A \leq X)$ la probabilité qu'un auteur a publié moins de X fois.*

Typicité d'une valeur d'un cluster. Pour une valeur v_i d'un cluster c de la partition X^M de la variable X selon le modèle de grille M, la typicité est définie par :

$$\tau(v_i) = \frac{1}{1 - P_{X^M}(c)} \sum_{\substack{c_j \in X^M \\ c_j \neq c}} P_{X^M}(c_j)(cost(M|c \setminus v, c_j \cup v) - cost(M)) \qquad (7)$$

où $P_{X^M}(c)$ est la probabilité d'avoir un point avec une valeur du cluster c, $c \setminus v$ est le cluster c duquel on a retiré la valeur v, $c_j \cup v$ est le cluster c_j auquel on a rajouté la valeur v et $M|c \setminus v, c_j \cup v$ le modèle de grille M qui a subi les modifications précitées. Intuitivement, une valeur v_i est représentative d'un cluster c et dite typique, si elle est proche de c et très différente (en moyenne) des autres clusters $c_j \neq c$. Aussi, pour un cluster d'auteurs, les auteurs qui ont peu publié (1 ou 2 fois, voir figure 5) sont souvent moins typiques que ceux qui sont les plus prolifiques, puisque si on les déplace vers un autre cluster, la différence de $cost$ (dans la formule de la typicité) sera faible.

Dans la figure 6, nous présentons pour chacun des quatre clusters d'auteurs (un par ligne), les auteurs les plus typiques (colonne 1), ainsi que la fréquence de publications (colonne 2) et le contraste (colonne 3) par une grille à deux dimensions Année $\times$ Conférence.

Tout d'abord, nous remarquons que les bornes de discrétisation du temps (les années de 1975 à 2012) découvertes par KHC, suit le lancement des conférences : 1975 pour VLDB et SIGMOD, 1984 pour ICDE, 1993 pour CIKM, 1995 pour KDD, 1998 pour PAKDD [3], 2001 pour ICDM et SDM ; puis, à partir de 2002, on obtient un intervalle par année (à l'exception de 2010-2011).

3. PAKDD a été lancé en 1997, toutefois les publications de cette année ne sont pas référencées par DBLP.

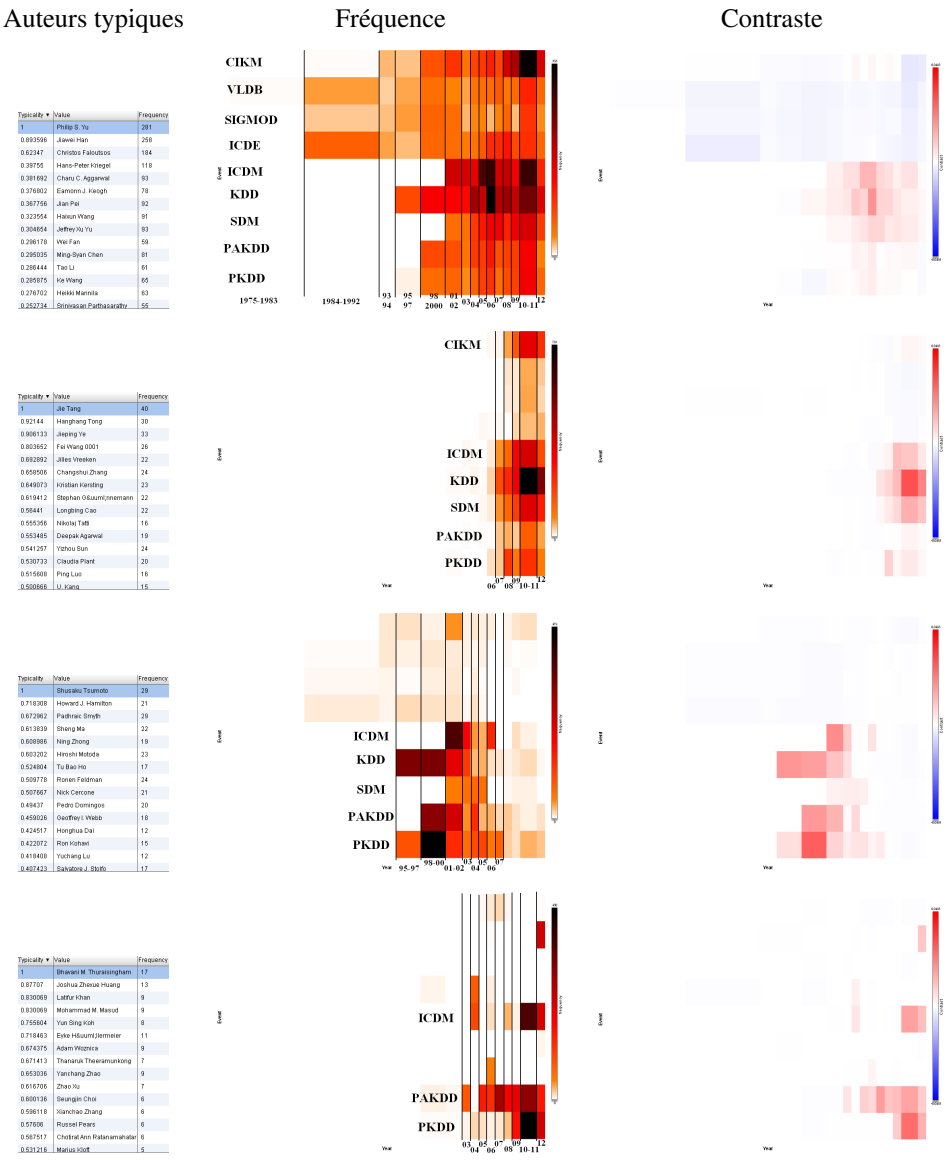

FIG. 6 – *Visualisations des clusters d'auteurs, des auteurs les plus typiques des clusters et du contraste entre clusters*

En ce qui concerne les auteurs, le premier cluster regroupe les auteurs très prolifiques, avec un large spectre de publications et qui publient dans les neuf conférences depuis 1984. Sans surprise, les plus typiques de ce cluster sont les chercheurs seniors et reconnus dans leur communauté. Ce qui caractérise ce groupe d'auteurs par rapport aux autres auteurs de la base (le contraste), c'est leur excès de publications à ICDM, KDD, SDM entre 2003 et 2011. Le deuxième cluster regroupe les auteurs "jeunes" spécialisés en FD qui ont publié entre 2006 et 2012 dans les conférences FD mais aussi CIKM (qui contient une composante FD). Ce qui caractérise ce cluster d'auteurs est leur densité de publications lors des six dernières années dans les conférences FD. Le troisième cluster regroupe un autre type de chercheurs séniors en FD, qui ont publié dans les conférences FD dans les premières années de leur lancement (de 1995 à 2006) et moins récemment, soit en raison de leur âge avancé, soit en raison d'autres activités de recherche (e.g., S. Tsumoto et T.B. Ho en bio-informatique, P. Smyth et P. Domingos en Machine Learning ou encore R. Kohavi passé dans l'industrie). Le dernier cluster regroupe un autre type de jeunes chercheurs en FD qui ont publié principalement à ICDM, PAKDD et PKDD les dernières années – ce qui fait contraste par rapport aux autres auteurs. Ce sont principalement des auteurs d'origine Européenne ou de la zone Pacifique-Asie qui publient dans les conférences régionales (PAKDD et PKDD).
Les cinq autres clusters d'auteurs sont composés d'auteurs orientés BD. De même, différentes classes d'âge d'auteurs apparaissent : e.g., un cluster de seniors qui ont publié dans les conférences BD depuis leur lancement jusqu'en 2012 ; un cluster de jeunes auteurs qui ont beaucoup publié en BD à partir de 2003-2004 ; ou encore un cluster singulier d'auteurs dont la caractéristique est un excès de publications à CIKM dans la dernière décennie.
Cette étude nous a permis de grouper les auteurs en fonction de leur séquence (ou trajectoire) de publications dans les neuf conférences de la base au cours du temps. La grille ainsi que les indicateurs proposés nous ont permis d'analyser et de visualiser les résultats du clustering. Ainsi, nous sommes capables d'identifier facilement les conférences des communautés FD et BD et les auteurs qui y publient, d'identifier les auteurs les plus typiques d'un cluster d'auteurs et de déterminer avec la mesure de contraste ce qui différencie un groupe d'auteurs du reste des auteurs de la base.

4 Conclusion & discussion

Nous avons proposé une méthode de clustering et d'analyse de séquences temporelles basée sur les modèles en grille. Les identifiants de séquence sont groupés en clusters, ainsi que les évènements, et la dimension temporelle est discrétisée en intervalles – le tout forme ainsi une grille tridimensionnelle (ou tri-clustering). Obtenir la grille optimale (au sens Bayésien) ne nécessite aucun paramétrage utilisateur. Pour exploiter la grille, nous avons proposé *(i)* une mesure de dissimilarité entre clusters afin de sélectionner le grain de la grille tout en contrôlant la perte d'information, *(ii)*, un critère (la typicité) pour identifier les valeurs les plus représentatives d'un cluster, *(iii)* ainsi que deux critères basés sur l'information mutuelle pour caractériser, interpréter et visualiser les clusters trouvés. Nos différentes propositions ont été validées sur des données simulées ainsi que sur des données réelles issues de DBLP. Notons qu'une étude du comportement asymptotique des indicateurs proposés a été réalisée et qu'une étude complète des trajectoires antenne-antenne d'utilisateurs du mobile – que l'on peut voir comme des séquences d'évènements – a été menée à l'échelle d'un pays (voir (Guigourès, 2013)).

Références

Bondu, A., M. Boullé, et D. Gay (2013). Les modèles en grilles. Principes, évaluation, algorithmes et applications. *Tutorial given at EGC*.

Boullé, M. (2012). Functional data clustering via piecewise constant nonparametric density estimation. *Pattern Recognition 45*(12), 4389–4401.

Dhillon, I. S., S. Mallela, et D. S. Modha (2003). Information-theoretic co-clustering. In *KDD'03*, pp. 89–98. ACM Press.

Giannotti, F., M. Nanni, et D. Pedreschi (2006). Efficient mining of temporally annotated sequences. In *SDM*.

Guigourès, R. (2013). *Utilisation des modèles de co-clustering pour l'analyse exploratoire des données*. Ph. D. thesis, Université Paris 1 Panthéon-Sorbonne.

Ley, M. (2009). DBLP - some lessons learned. *PVLDB 2*(2), 1493–1500.

Masseglia, F., P. Poncelet, M. Teisseire, et A. Marascu (2008). Web usage mining : extracting unexpected periods from web logs. *DMKD 16*(1), 39–65.

Mörchen, F. (2007). Unsupervised pattern mining from symbolic temporal data. *SIGKDD Explorations 9*(1), 41–55.

Nadif, M. et G. Govaert (2010). Model-based co-clustering for continuous data. In *ICMLA*, pp. 175–180.

Patnaik, D., P. Butler, N. Ramakrishnan, L. Parida, B. J. Keller, et D. A. Hanauer (2011). Experiences with mining temporal event sequences from electronic medical records : initial successes and some challenges. In *KDD*, pp. 360–368.

Saleh, B. et F. Masseglia (2011). Discovering frequent behaviors : time is an essential element of the context. *KaIS 28*(2), 311–331.

Studer, M., N. S. Müller, G. Ritschard, et A. Gabadinho (2010). Classer, discriminer et visualiser des séquences d'événements. In *EGC*, pp. 37–48.

Yang, Q. et X. Wu (2006). 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making 5*(4), 597–604.

Summary

We suggest a novel way of clustering and analysing time annotated sequences by density estimation. Our model is a tri-dimensional data grid in which the sequences are partitioned into clusters, the temporal dimension is discretized into intervals and the event dimension is partitioned into groups. And the 3D cells of the grid form a nonparametric estimator of the joint density of the sequences and dimensions of the temporal events. Thus, the sequences of a cluster are similar in the sense that they follow the same density along the time dimension. We also suggest a way for exploiting the clustering through the simplification of the grid; together with new indicators useful for analyzing the learned clusters and characterizing their main components. Experiments are lead on both synthetic and real data to demonstrate the efficiency and effectiveness of the approach.