

Construction de partitions géo-temporelles à partir d'une divergence de Kullback-Leibler modifiée en vue de la navigation dans une collection d'images personnelles

Marc Gelgon

Antoine Pigeau

Afshin Nikseresht

LINA FRE 2729 CNRS / INRIA ATLAS group

2, rue de la Houssinière 44322 Nantes - France

{prénom.nom}@univ-nantes.fr

Concours Jeune Chercheur : Non

Résumé

L'usage d'appareils mobiles équipés de capteurs photographiques entraîne le problème de la gestion de larges collections d'images personnelles. Suivant les études sur les besoins d'utilisateurs, nous proposons une technique d'organisation basée sur les méta-données géo-temporelles des images. Notre algorithme permet de construire incrémentalement une hiérarchie de deux partitions, l'objectif étant de représenter la structure temporelle ou géographique de la collection dans deux hiérarchies distinctes. Pour chaque partition de la hiérarchie, des critères statistiques (la vraisemblance complétée intégrée et une divergence de Kullback-Leibler modifiée) sont optimisés afin de déterminer ou mettre à jour leurs paramètres. L'approche probabiliste permet une flexibilité dans l'évolution des partitions.

Mots clefs

Classification d'images, Application sur les terminaux mobiles, Méta-données temporelles et géographiques, Classification statistique

1 Objectif et état de l'art

L'utilisation courante d'appareils mobiles équipés de capteurs photographiques (téléphone mobile, appareil photographique numérique) permet à un utilisateur de concevoir de grandes collections d'images. Un besoin essentiel étant de fournir aux utilisateurs des solutions pour rechercher ses images parmi des milliers, l'indexation de ce type de données multimédia est un domaine de recherche présentant de nombreux intérêts. Les produits Lifeblog de Nokia et MyLifeBits de Microsoft sont deux réponses récentes de la part d'industriels.

Les particularités de la tâche, comparées aux approches basées sur le contenu des données multimédia, réside dans les méta-données disponibles des images fournies par l'appareil d'acquisition (date, localisation géographique, paramètres de prise de vue) et les critères d'organisation préfé-

rés des utilisateurs. Des études [1] ont montré sans surprise que les interactions sociales, les événements, la date et les lieux sont des critères pertinents.

Dans ce papier, nous utilisons seulement les méta-données temporelles et géographiques attachées à chaque image. Nous faisons l'hypothèse que les coordonnées géographiques sont fournis par un système GPS/E-OTD. L'ensemble des données à structurer est ainsi un flux $\{(t, (x, y)) \in \mathbb{R} \times \mathbb{R}^2\}$. Notre contribution est une technique (les critères statistiques et l'algorithme incrémental pour les optimiser) pour construire automatiquement une classification d'images sous forme de hiérarchie de deux partitions (une partition *générale* et une partition *fine*), en se basant sur la date et le lieu de prise de vue. Deux hiérarchies distinctes sont construites pour la date et le lieu, respectivement à partir des méta-données temporelles et géographiques, avec une méthode presque identique. Les hiérarchies sont construites incrémentalement, au fur et à mesure que les méta-données sont ajoutées, les phases d'acquisition et de parcours de la collection étant très liées.

Notre approche est orientée vers la navigation dans la collection contrairement à une approche orientée requête. En effet l'utilisateur connaît la structure de la collection et les interactions homme-machine sur les appareils mobiles sont limitées. Notre objectif est de permettre à un utilisateur de parcourir l'organisation proposée à partir de résumés des classes, définis chacun par un petit sous-ensemble d'images visuellement représentative de leur classe. Le choix des images représentatives n'est pas résolu dans ce papier, mais des solutions existent dans le contexte de résumés de données vidéos. Notons de plus que les structures d'organisation proposées peuvent aussi améliorer la gestion des mémoires caches dans un contexte d'application mobile client-serveur [2].

La plupart des solutions d'organisation existantes utilise comme critère la date des images, celle-ci étant intuitive, disponible et fiable. La segmentation incrémentale d'un flux de données temporelles est proposée dans [3]. Les travaux les plus proches des nôtres organisent aussi les images à partir de la date et du lieu de prise de vue. Leur princi-

pal avantage est la prise en compte simultanée des deux critères, mais la solution proposée est basée sur des paramètres arbitraires et n'est pas incrémentale. Dans notre proposition nous évitons de tels paramètres grâce à une approche probabiliste et nous fournissons un algorithme incrémental. De notre point de vue, cette dernière propriété est importante pour organiser la collection, l'utilisateur n'ayant pas besoin de régulièrement penser à mettre à jour sa classification. En tournant en tâche de fond sur l'appareil mobile de l'utilisateur avec une faible priorité, la complexité de mise à jour de notre proposition est loin d'être aussi coûteuse que des codeurs vidéo en temps réels fonctionnant sur de telle plate-forme.

Ce papier est organisé comme suit. La section 2 présente la technique d'organisation, en détaillant les constructions des deux niveaux de la hiérarchie. La section 3 fournit des résultats expérimentaux et la section 4 résume notre contribution et propose quelques perspectives.

2 Construction de la hiérarchie de deux partitions

Initialisation :

ajout d'une nouvelle donnée dans l'ensemble D et itérer l'algorithme EM, initialisé avec le modèle obtenu après l'ajout de la nouvelle donnée précédente. Soit \mathcal{M}_f le modèle correspondant.

Phase de division :

ordonner les candidats $\{S_1, \dots, S_d\}$ à diviser selon le critère d'entropie $E = -\sum_{i=1}^n t_{ik} \cdot \log(t_{ik})$ (nous divisons les classes avec des paramètres se chevauchant afin de faire apparaître un nouveau groupe de données).

pour les α premières composantes ordonnées par entropie décroissante :

- diviser la composante et mettre à jour le modèle à partir de \mathcal{M}_f ;
- itérer l'algorithme EM jusqu'à convergence.

garder parmi les modèles testés le nouveau modèle \mathcal{M}_f optimisant le critère ICL.

Phase de fusion :

ordonner les paires de composantes à la fusion en fonction de leur distance de Mahalanobis (en quelque sorte, nous testons les fusions des composantes les plus proches).

pour les α premières composantes ordonnées selon une distance croissante de Mahalanobis :

- fusionner les composantes et mettre à jour le modèle à partir de \mathcal{M}_f ;
 - itérer l'algorithme EM jusqu'à convergence.
- garder le modèle \mathcal{M}_f optimisant le critère ICL.

Figure 1 – Algorithme pour déterminer la partition fine.

Notre objectif étant de faciliter le parcours de la collection sur un appareil mobile, nous devons prendre en compte

deux critères sensibles : les contraintes matérielles, la taille de l'écran en particulier, et le nombre de groupes pour représenter la structure des données. Notre solution consiste ainsi à construire une hiérarchie à deux partitions :

- en définissant manuellement le nombre de classes dans la partition *générale* selon les contraintes d'interfaces ;
- en déterminant, au fur et à mesure de l'ajout de nouvelles méta-données, le nombre de groupes dans la partition *fine* à l'aide d'un algorithme non-supervisé.

Pour chaque nouvelle prise de vue, nous devons ainsi mettre à jour 4 partitions : une partition *générale* et une partition *fine* pour chacune des hiérarchies temporelle et spatiale. Nous présentons ici les grandes lignes de notre approche, et explicitons les modèles et algorithmes dans les sections suivantes.

1. la construction de la partition *fine* suit la technique proposée dans [4]. Elle est basée sur une représentation des données avec des modèles de mélange probabiliste gaussien. Le critère ICL (la vraisemblance complétée intégrée) permet de déterminer le nombre de classes dans le modèle. Il présente l'avantage d'être robuste face aux données non gaussiennes. Un algorithme d'optimisation de ce critère, basé sur des fusion et divisions de classes, permet une recherche semi-locale dans les valeurs des paramètres pour mettre à jours le modèle incrémentalement. Enfin une estimation bayésienne des matrices de covariance préserve de l'instabilité des paramètres dans le cas où les classes sont associées à un trop petit échantillon de données. Le choix de l'approche basée sur les modèles de mélange présente deux avantages :
 - il permet d'éviter le problème d'explosion combinatoire inhérente aux problèmes de regroupement de données ;
 - il convient bien à la nature incrémentale de la tâche. Les affectations des données aux classes sont flexibles et permettent une évolution souple des partitions au fur et à mesure de l'ajout de nouvelles données.
2. la partition *compacte* des données est déterminée en identifiant des groupes de classes pertinents dans la partition *fine* (la technique diffère sensiblement de celle proposée dans [4]). Tandis que le nombre de classes dans la partition *fine* est déterminé automatiquement à partir des données, la partition *compacte* a un nombre de classes plus faible défini explicitement en prenant en compte les contraintes matérielles. L'objectif est de fournir un résumé de la collection à partir duquel l'utilisateur peut ensuite obtenir du détail avec la partition *fine*. Le regroupement des classes de la partition *fine* est formulé comme l'identification d'un modèle de mélange plus général, obtenu en minimisant une distance entre ces deux partitions. Cette distance est définie par une divergence de Kullback-Leibler modifiée. Le problème combinatoire du regroupement des classes de la partition *fine* peut être

résolu à l'aide d'un algorithme itératif opérant en alternant la phase d'estimation des paramètres du modèle général et la phase de ré-assignement des classes de la partition *fine* aux classes de la partition *compacte*. L'approche est similaire aux techniques classiques de partitionnement, mais elle opère sur les paramètres des classes et non pas sur les données initiales. L'approche présente les propriétés suivantes :

- la complexité en temps de calcul est faible puisque seuls les paramètres, et non pas les données initiales, de la partition *fine* sont utilisés ;
- la mise à jour de la partition *compacte* est incrémentale. L'initialisation des regroupements de la partition *fine* peut être fixée à partir de la configuration obtenue à l'étape précédente. En plus de réduire la complexité en temps de calcul, cela permet une meilleure stabilité de la partition *compacte* au cours du temps, un point important pour l'utilisateur ;
- la méthode n'est pas affectée par l'aspect non-gaussien des méta-données initiales, ce qui est le cas d'autres approches essayant de trouver une hiérarchie de modèles à partir des données.

2.1 Recherche de la partition *fine*

Nous optons pour une approche basée sur les modèles de mélange, les modèles probabilistes permettant d'obtenir les classes et les affectations des données aux classes. Ce dernier point est pertinent pour notre aspect incrémental, puisque les affectations des données aux classes sont flexibles et peuvent évoluer facilement lorsque l'ajout de nouvelles données dans la classification suggère une reconsidération de la structure précédemment trouvée.

Les données D (soit les lieux (x, y) ou les dates t) sont ainsi supposées être générées aléatoirement à partir d'un modèle de mélange gaussien.

Critère de sélection de modèles :

Dans le cadre des modèles de mélange, un bon critère de comparaison entre plusieurs hypothèses de classification ayant un nombre différent de classes est la vraisemblance complétée intégrée (ICL) [5]. Pour une hypothèse H_k , le critère ICL est défini par :

$$p(D, Z|H_K) = \int p(D, Z|\Theta_K, H_K)p(\Theta_K|H_K)d\Theta_K \quad (1)$$

où $\Theta_K = (\theta_1, \dots, \theta_K)$ est le vecteur de paramètres de H_K et $\theta_i = (w_i, \mu_i, \Sigma_i)$, $1 \leq i \leq K$ (w_i est la proportion de mélange de la composante i , Σ_i sa covariance et μ_i son centre). Le calcul pratique de cette expression exploite une approximation du critère BIC (1), exprimée par :

$$ICL = -ML + \frac{\nu_K \log(n)}{2} - \sum_{k=1}^K \sum_{i=1}^n t_{ik} \cdot \log(t_{ik}) \quad (2)$$

où ML est la log-vraisemblance du modèle de mélange optimisé, ν_K est le nombre de paramètres libres du modèle

à K composantes, n est le nombre de données et t_{ik} est la probabilité à posteriori pour une observation i d'avoir pour origine la composante k . Les t_{ik} sont en fait les espérances des affectations binaires probabilisées z_{ik} . En pratique, les z_{ik} sont déterminés pendant l'étape E de l'algorithme EM, décrit dans la suite.

En comparaison avec le critère BIC approxinant la vraisemblance marginalisée des données, ce critère optimise conjointement la vraisemblance des données et les labels inconnus z d'affectations des données au modèle. L'introduction de ces variables auxiliaires permet de prendre en compte la qualité de la partition lors de l'estimation des paramètres. L'expression (2) est une variation du critère BIC : le terme de droite ajouté a un comportement entropique et favorise les classes bien séparées [5].

Il est fréquent qu'une classe soit assignée à un petit échantillon de données, entraînant une mauvaise estimation de sa covariance. Nous traitons le problème en introduisant, à l'étape M de l'algorithme EM, des estimations de covariances régularisées calculées à partir des espérances des distributions à posteriori des matrices de covariances initiales (utilisant les modèles à priori conjugués de Gamma pour le temps (une dimension) et de Wishart pour le lieu (deux dimensions)).

Algorithme d'optimisation :

La recherche de partitions de données incrémentale suppose de pouvoir modifier facilement l'affectation des données aux classes et d'ajuster le nombre de composantes en fonction des nouvelles données. Notre proposition consiste à utiliser la partition obtenue au temps t comme initialisation de l'optimisation du critère ICL pour la partition à $t + 1$: cela garantit la stabilité de la partition au cours du temps et facilite ainsi son exploration par l'utilisateur.

L'algorithme Expectation-Maximization (EM) [6] permet d'optimiser localement le critère ICL à un nombre constant de composantes. Il se décompose en deux étapes, l'étape E, dans laquelle les probabilités d'affectation des données à chaque composante sont calculées, conditionnellement aux paramètres des modèles, et l'étape M, dans laquelle les paramètres des modèles sont estimés en se basant sur l'estimation courante des affectations des données aux modèles.

Il reste deux problèmes à résoudre :

- l'évolution du nombre de composantes ;
- l'optimisation du critère ICL (équation 2) : l'utilisateur prenant généralement ses images par paquets, le flux de données ne peut pas être modélisé comme une série de données indépendantes. De plus grands efforts de réorganisation des classes sont ainsi nécessaires que si les données étaient totalement indépendantes. Par exemple, au fur et à mesure que les données du même lieu sont ajoutées, les classes sont optimisées localement sur ce lieu spécifique et peuvent être ainsi plus difficiles à modifier si un nouveau lieu apparaît. L'optimisation au cours du temps est ainsi délicate et des minima locaux sont souvent obtenus si seul un algorithme EM classique

est utilisé.

Notre algorithme EM incrémental consiste à tester plusieurs divisions de composantes suivies de plusieurs fusions pour chaque nouvelle donnée rajoutée [4]. On alterne ainsi les phases de recherche semi-locale avec des itérations de l'algorithme EM, jusqu'à convergence. Les deux étapes permettent de minimiser le critère ICL et servent le même objectif : éviter les minima locaux et permettre l'évolution du nombre de composantes au cours du temps. L'algorithme est détaillé par la figure 1.

2.2 Recherche de la partition compacte

Soit M_f le modèle de mélange obtenu pour la partition *fine*. Trouver des groupes pertinents de ces classes peut être exprimé par la recherche d'un modèle M_c maximisant la log-vraisemblance des données D supposées être générées à partir de M_f (voir (3)). Cela revient à minimiser la divergence de Kullback-Leibler $KL(M_f||M_c)$ [7], définie par (4), qui représente la perte d'information due à l'approximation de M_f par M_c .

$$\hat{M}_c = \arg \max E_{M_f} [\ln p(D|M_c)] \quad (3)$$

$$\hat{M}_c = \arg \min \left[- \int M_f(x) \ln \frac{M_c(x)}{M_f(x)} dx \right] \quad (4)$$

Un problème est l'absence d'une forme approchée pour cette divergence dans le cas des modèles de mélange gaussien. Il est résolu en se basant sur une variation de cette divergence, proposée récemment par [8], qui consiste à minimiser la mesure de similarité suivante

$$d(M_f, M_c) = \sum_{i=1}^{m_f} w_f^i \min_{j=1}^{m_c} KL(N_f^i || N_c^j) \quad (5)$$

où N_f^i (resp. N_c^j) est la $i^{\text{ème}}$ composante de M_f (resp. de M_c), m_f, m_c représente respectivement le nombre de composantes dans M_f et M_c et w_f^i est la proportion de mélange de la composante i du modèle M_f .

Cette mesure de similarité présente deux propriétés :

– la complexité en temps de calcul est faible puisque la divergence de Kullback-Leibler entre deux gaussiennes, dont les paramètres sont (μ_1, Σ_1) and (μ_2, Σ_2) , est exprimée par l'expression simple suivante :

$$\frac{1}{2} \left(\log \frac{|\Sigma_2|}{|\Sigma_1|} + Tr(\Sigma_2^{-1} \Sigma_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2) - \delta \right) \quad (6)$$

où δ est la dimension de l'espace des données. Nous disposons ainsi d'une expression analytique facilement calculable pour déterminer une divergence de Kullback-Leibler modifiée.

Ainsi, suivant [8], nous optimisons localement un critère avec un algorithme itératif, détaillé par la figure 2. Il peut être comparé à un algorithme K-means appliqué aux composantes d'un modèle de mélange.

À partir de l'initialisation $\hat{\pi}^0$ obtenue à convergence au temps $t-1$ (si nécessaire en prenant en compte les apparitions ou disparitions de composantes de la partition *fine*).

$it = 0$

Répéter

Mise à jour des composantes pour le modèle M_c :

connaissant la classification courante $\hat{\pi}^{it}$, obtenue initialement ou calculer à l'étape précédente, mettre à jour les paramètres du modèle de mélange comme suit :

$$\hat{M}_c^{it} = \arg \min_{M_c \in \mathcal{M}_{m_c}} d(M_f, M_c, \hat{\pi}^{it}) \quad (7)$$

où \mathcal{M}_{m_c} est l'espace de tous les modèles de mélange avec m_c composantes pouvant être formé à partir des regroupements des composantes de M_f . Cette ré-estimation revient à mettre à jour chaque composante de M_c comme suit. Pour la composante j , le calcul conduit à l'expression suivante :

$$\hat{w}_c^j = \sum_{i \in \pi^{-1}(j)} w_f^i, \quad (8)$$

$$\hat{\mu}_c^j = \frac{\sum_{i \in \pi^{-1}(j)} w_f^i \mu_f^i}{\hat{w}_c^j}, \quad (9)$$

$$\hat{\Sigma}_c^j = \frac{\sum_{i \in \pi^{-1}(j)} w_f^i (\Sigma_f^i + (\mu_f^i - \hat{\mu}_c^j)(\mu_f^i - \hat{\mu}_c^j)^T)}{\hat{w}_c^j} \quad (10)$$

où $\pi^{-1}(j)$ est une notation simplifiée de $\hat{\pi}^{-1, it}(j)$, l'ensemble des composantes de M_f projetées dans la composante j de M_c .

Optimisation de la divergence de Kullback-Leibler modifiée 5 :

soit le modèle de mélange \hat{M}_c^{it} obtenu à l'étape précédente, rechercher la transformation π^{it+1} , définie à partir de $\{1, \dots, m_f\}$ vers $\{1, \dots, m_c\}$, qui regroupe au mieux les composantes de M_f pour construire les composantes \hat{M}_c^{it+1} , de la façon suivante :

$$\hat{\pi}^{it+1} = \arg \min_{\pi} d(M_f, \hat{M}_c, \pi) \quad (11)$$

En d'autres termes, chaque composante i de M_f est projetée vers la composante j de \hat{M}_c^{it} la plus proche, selon la convergence de Kullback-Leibler modifiée ((12) ci-dessus). Cette phase revient à une recherche exhaustive parmi les composantes sources et présente une complexité en temps de calcul faible grâce à la disponibilité de (6).

$$\pi^{it+1}(i) = \arg \min_j KL(N_f^i || N_c^j) \quad (12)$$

$it = it + 1$

jusqu'à convergence (i.e. $\pi^{it+1} = \pi^{it}$)

Figure 2 – Algorithme pour déterminer la partition compacte.

3 Expérimentation

Nous avons réalisé des expériences sur une collection personnelles réelles de 721 images. Elle regroupe des événements pris sur trois ans dans plusieurs pays. Le nombre de composantes de la partition *compacte* est fixé à 4.

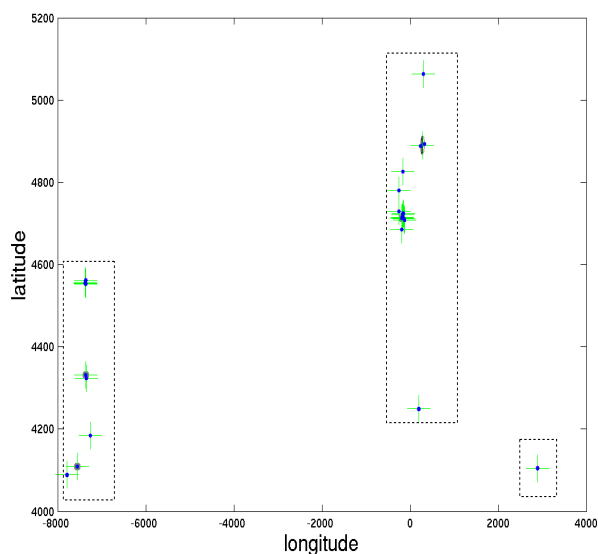


Figure 3 – Hiérarchie spatiale : les classes de la partition *fine* sont représentées par le signe +. La partition *compacte* est définie par les lignes en pointillées.

La figure 3 présente la partition spatiale obtenue. La partition *compacte* est définie par les lignes en pointillées et la partition *fine* par les signes +. Cette dernière comprend des classes de données très compactes, dû à la configuration très regroupées des données. L'utilisateur a pris en général beaucoup d'images dans des lieux précis et chaque lieu a été convenablement retrouvé dans 25 classes. La partition *compacte* est quant à elle composée de trois classes. Notre algorithme permet, dans le cas où une composante de la partition *compacte* n'est associée à aucune classe de la partition *fine*, d'adapter la complexité du modèle. Ainsi le modèle général obtenu ne contient que 3 composantes ce qui semble cohérent au vu de la structure des données.

La figure 4 présente la classification temporelle finale obtenue pour 150(a), 300(b) et 721(c) images. La partition *compacte* est définie par les lignes en pointillées et la partition *fine* par les lignes continues. La partition *fine* finale (figure 4(c)) comprend 41 classes et nous avons vérifié que chaque limite de classes correspondait bien à un changement d'événements dans la collection. Les partitions obtenues sont stables dans le temps puisque les différents états obtenus présentent des similarités. Les résumés proposés dans chacune des partitions sont correctes, les limites de classes étant visuellement justifiées. Les principaux groupes de données sont bien mis en valeur.

4 Conclusion

Ce papier propose une technique d'organisation d'une collection d'images personnelles acquises sur un mobile, à partir des méta-données temporelle et spatiale. Le choix de ces méta-données est motivé par leur disponibilité et leur

bonne interprétation par les utilisateurs. D'autres critères, par exemple sur le contenu de l'image, peuvent bien sûr être utilisés en conjonction avec ces méta-données.

La contribution de ce papier est une technique de classification automatique, limitant les paramètres arbitraires critiques et dépendant ainsi seulement de la structure des données. Le principe est de construire progressivement une hiérarchie de modèles de mélange de deux partitions, en fonction de l'ajout de nouvelles images dans la collection. Le critère ICL maintient une définition uniforme de la partition *fine* et un critère de divergence de Kullback-Leibler modifiée est utilisé pour résumer la partition *fine* à partir de ses paramètres. L'intérêt de l'approche est le très faible coût de calcul de la partition *compacte*, mise à jour incrémentalement au fur et à mesure de l'ajout de nouvelles images. Elle présente de plus l'avantage de s'emboîter parfaitement dans la partition *fine*.

Une perspective de travail est de proposer une méthode pour déterminer le nombre de classes dans la partition *compacte* en fonction de la structure des données. Pour chaque modification de la partition *fine*, il serait nécessaire de tester plusieurs solutions de résumés avec un nombre variable de classes et de sélectionner la partition *compacte* la plus pertinente. Une méthode basée sur un critère statistique, par exemple le critère AIC, est en cours d'étude.

Références

- [1] K. Rodden. How do people manage their digital photographs? Dans *ACM Conference on Human Factors in Computing Systems*, pages 409 – 416, Fort Lauderdale, 2003.
- [2] A. Myka, J. Yrjanainen, et M. Gelgon. Enhanced storing of personal content. US Patent 16660/10502275, Nokia corp., Juillet 2004.
- [3] J. C. Platt et B. A. Field M. Czerwinski. Photo-TOC : Automatic clustering for browsing personal photographs. Rapport technique MSR-TR-2002-17, Microsoft Research, Février 2002.
- [4] A. Pigeau et M. Gelgon. Building and tracking hierarchical partitions of image collections on mobile devices. Dans *ACM Multimedia conference*, pages 141–150, Singapore, nov 2005.
- [5] C. Biernacki, G. Celeux, et G. Govaert. Assessing a mixture model for clustering with the integrated classification likelihood. Dans *IEEE Transaction on pattern analysis and machine intelligence*, volume 22, pages 719–725, Juillet 2000.
- [6] A. P. Dempster, N. M. Laird, et D. B. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *J. R. Stat. Soc.*, pages 1–38, 1977.
- [7] C. Bishop. *Neural networks for Pattern Recognition*. Oxford University Press, 1995.
- [8] J. Goldberger et S. Roweis. Hierarchical clustering of a mixture model. Dans *Proc. of Neural Information Processing Systems (NIPS'2004)*, pages 505–512, 2004.

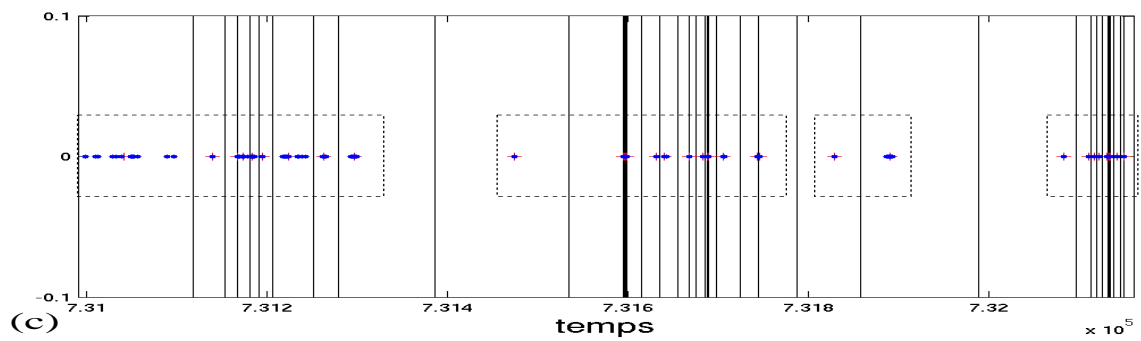
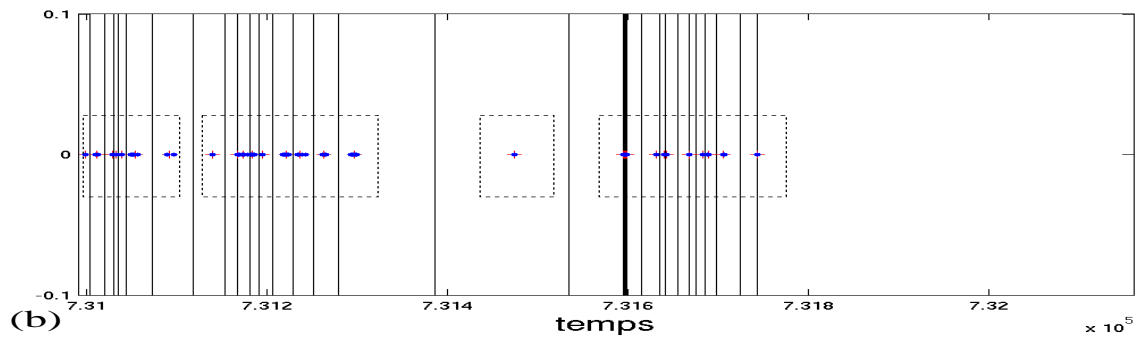
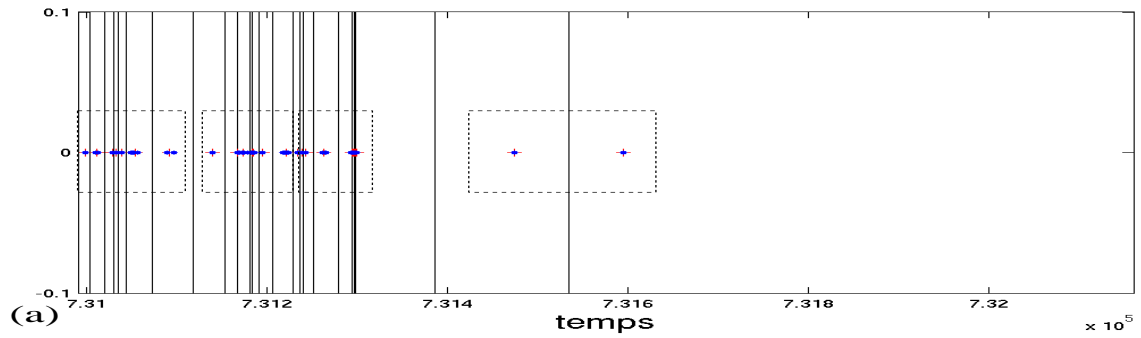


Figure 4 – Hiérarchie temporelle : plusieurs états de la partition obtenue pour 300, 450 et 721 données. Les limites de classes dans la partition fine sont représentées par les lignes continues. La partition compacte est définie par les lignes en pointillées.