

Tri-clustering pour données de comptage

Margot Selosse, Antoine Gourru, Julien Jacques, Julien Velcin

► **To cite this version:**

Margot Selosse, Antoine Gourru, Julien Jacques, Julien Velcin. Tri-clustering pour données de comptage. Journées des Statistiques, Jun 2019, Nancy, France. hal-02115353

HAL Id: hal-02115353

<https://hal.archives-ouvertes.fr/hal-02115353>

Submitted on 30 Apr 2019

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

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

TRI-CLUSTERING POUR DONNÉES DE COMPTAGE

Margot Selosse¹ Antoine Gourru¹ Julien Jacques¹ & Julien Velcin¹

¹ *Université de Lyon, Lumière Lyon 2, laboratoire ERIC*

Résumé. Les données de comptage sont très utilisées dans le monde actuel pour modéliser les occurrences d'un évènement (apparence d'un mot dans un texte, passage d'une voiture à un carrefour, contact entre deux personnes d'un réseau sociaux etc.). Ce travail s'intéresse aux données de comptage dynamiques, lorsque les occurrences sont dénombrées sur plusieurs périodes différentes. Dans ce cas, les données sont vues comme un tenseur (ou cube). L'approche proposée développe un algorithme de tri-clustering, qui va simultanément créer des clusters en ligne, en colonne et en profondeur. La distribution de Poisson est utilisée pour modéliser les données, et un algorithme EM variationnel est décrit pour inférer les paramètres du modèle.

Mots-clés. tri-clustering, modèle des blocs latents

Abstract. Count data are widely used to model the occurrences of an event (appearance of a word in a text, car traffic at a crossroads, conversation between two people in a social network etc.). This work focuses on dynamic count data, where occurrences are enumerated over several different time periods. In this case, the data is seen as a tensor (or cube). The proposed approach develops a tri-clustering algorithm, which simultaneously creates clusters in lines, in columns and in depth. The Poisson distribution is used to model the data, and a variational EM algorithm is described to infer the model parameters.

Keywords. tri-clustering, latent block model

1 Introduction

Les données de comptage dénombrent les occurrences d'un évènement sur une période donnée. Elles sont donc très utilisées pour décrire le monde réel (données textuelles, trafic routier, réseaux sociaux etc.). Souvent, ces données sont elles mêmes dynamiques: les données sont collectées sur une longue durée (par exemple une année), qui est ensuite découpée en plusieurs périodes (des mois ou des semaines). Dans ce contexte, les données sont stockées en tenseur (ou cube); les individus sont en ligne, les variables qui les décrivent sont en colonne, et les différentes périodes sont en profondeur. Dans un contexte non supervisé, ce type d'objet est difficile à utiliser de manière brute car la plupart des algorithmes d'analyse requièrent une matrice en entrée. Pour pallier à ce problème, les experts procèdent à une agrégation des données sur l'une des dimensions ou traitent chaque

période de manière indépendante. Ce travail propose une approche de tri-clustering, qui consiste à former simultanément des clusters en ligne, en colonne et en profondeur. Ainsi, le croisement d'un cluster en ligne, d'un cluster en colonne et d'un cluster en profondeur est appelé tri-cluster ou bloc. Le tenseur original est donc résumé en plusieurs blocs homogènes, ce qui nous rend plus aptes à synthétiser et analyser les données.

2 Notations

Nous considérons un tenseur X avec un nombre I de lignes, J colonnes et une profondeur de K . Nous notons x_{ijk} un élément de X tel que $1 \leq i \leq I, 1 \leq j \leq J$ et $1 \leq k \leq K$. Etant dans un contexte de tri-clustering, nous supposons qu'il existe G clusters en ligne, H clusters en colonne et L clusters en profondeur. Pour cela, nous introduisons les matrices \mathbf{v} , \mathbf{w} et \mathbf{z} , qui correspondent respectivement aux partitions des clusters en ligne, colonne et profondeur. Ainsi v_i est un vecteur de taille G , tel que v_{ig} est égal à 1 lorsque la i -ème ligne appartient au g -ième cluster en ligne, et 0 dans le cas contraire. Similairement, w_{jh} est égal à 1 lorsque la j -ème colonne appartient h -ième cluster en colonne, et 0 autrement. Enfin, z_{kl} est égal à 1 lorsque la k -ème profondeur appartient l -ième cluster en profondeur, et 0 autrement.

3 Définition du modèle

Le modèle des blocs latents [1], noté LBM est l'un des modèles statistiques les plus utilisés en co-clustering (clustering des lignes et des colonnes pour une matrice). Il repose sur l'hypothèse que les éléments au sein d'un bloc sont les réalisations de variables aléatoires qui suivent une distribution paramétrique spécifique au bloc.

3.1 Hypothèses du modèle

Notre modèle étend le LBM pour des tenseurs, voici les hypothèses qui en découlent:

Hypothèse 1 *Les partitions v_i, w_j, z_k sont indépendantes pour tout $\{i, j, k\}$.*

Cela se traduit donc par:

$$\begin{aligned}
 p(v, w, z) &= p(v)p(w)p(z) \\
 &= \prod_i p(v_i) \prod_j p(w_j) \prod_k p(z_k) \\
 &= \prod_i \prod_g \alpha_g^{v_{ig}} \prod_j \prod_h \beta_h^{w_{jh}} \prod_k \prod_l \gamma_l^{z_{kl}}
 \end{aligned} \tag{1}$$

où α , β et γ sont les proportions de mélange des clusters, en ligne, colonne et profondeur respectivement.

Hypothèse 2 Conditionnellement à \mathbf{v} , \mathbf{w} et \mathbf{z} les éléments x_{ijk} d'un bloc sont indépendants et identiquement distribués.

Nous avons donc :

$$x_{ijk} \sim f(\cdot, \theta_{ghl}) \text{ pour tout } \{i, j, k\} \text{ tel que } v_{ig}w_{jh}z_{kl} = 1$$

. Ici, θ_{ghl} représente le paramètre de la distribution du bloc formé par les clusters g , h et l . Par la suite, nous l'appellerons simplement le bloc (g, h, l) .

Ainsi, nous obtenons:

$$p(X|\mathbf{v}, \mathbf{w}, \mathbf{z}) = \prod_{ijkghl} f(x_{ijk}, \theta_{ghl})^{v_{ig}w_{jh}z_{kl}}$$

L'équation du modèle peut donc être écrite:

$$\begin{aligned} p(X) &= \sum_{v \in \mathcal{V}} \sum_{w \in \mathcal{W}} \sum_{z \in \mathcal{Z}} p(x|\mathbf{v}, \mathbf{w}, \mathbf{z}) p(\mathbf{v}, \mathbf{w}, \mathbf{z}) \\ &= \sum_{v, w, z} \prod_{ig} \alpha_g^{v_{ig}} \prod_{jh} \beta_g^{w_{jh}} \prod_{kl} \gamma_g^{z_{kl}} \prod_{ijkghl} f(x_{ijk}; \theta_{ghl})^{v_{ig}w_{jh}z_{kl}} \end{aligned} \quad (2)$$

Où \mathcal{V}, \mathcal{W} et \mathcal{Z} sont l'ensemble des partitions possibles.

3.2 Le tri-clustering avec la distribution de Poisson

Dans ce contexte, il est considéré qu'un élément x_{ijk} est tiré d'une loi de Poisson de paramètre λ_{ijk} , soit:

$$x_{ijk} \in \mathbb{N} \sim \mathcal{P}(\lambda_{ijk})$$

Ainsi, nous avons:

$$f(x_{ijk}; \lambda_{ijk}) = e^{-\lambda_{ijk}} \frac{\lambda_{ijk}^{x_{ijk}}}{x_{ijk}!}$$

Le paramètre λ_{ijk} , lui, est considéré être une fonction d'un effet de bloc ζ_{ghl} , d'un effet de ligne μ_i , d'un effet de colonne ν_j et d'un effet de profondeur η_k :

$$\lambda_{ijk} = \mu_i \nu_j \eta_k \sum_{ghl} v_{ig} w_{jh} z_{kl} \zeta_{ghl}$$

Pour assurer l'identifiabilité du modèle, nous fixons μ_i , ν_j , et η_k tel que suit:

$$\mu_i = \sum_{j,k} x_{ijk}, \nu_j = \sum_{i,k} x_{ijk}, \text{ et } \zeta_k = \sum_{i,j} x_{ijk}.$$

En fixant μ_i , ν_j et η_k , le seul paramètre à estimer concernant la distribution de Poisson est $\zeta = (\zeta_{ghl})$.

4 Inférence du modèle

Le modèle est donc défini par ses variables latentes \mathbf{v} , \mathbf{w} et \mathbf{z} et ses paramètres $\boldsymbol{\theta} = \{\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$. Dans cette configuration, l'algorithme EM est une méthode classique pour maximiser la vraisemblance en présence de variables latentes. Cependant, en tri-clustering, il requiert le calcul de la probabilité $p(v_{ig}, w_{jh}, z_{kl} | \mathbf{x})$ qui n'est pas calculable. Un algorithme EM variationnel (VEM) est donc utilisé dans ce travail pour ne pas être obligé de calculer explicitement cette probabilité. L'algorithme VEM itère un certain nombre de fois Q les étapes suivantes (q représentant la q -ième itération) :

Mise à jour des partitions:

$$\begin{aligned} v_{ig}^q &= \frac{\alpha_g^{q-1} \exp(\sum_{jhkl} w_{jh}^{q-1} z_{kl}^{q-1} \log f(x_{ijk}; \lambda_{ijk}^{q-1}))}{\sum_{g'} \alpha_{g'}^{q-1} \exp(\sum_{jhkl} w_{jh}^{q-1} z_{kl}^{q-1} \log f(x_{ijk}; \lambda_{ijk}^{q-1}))}, \\ w_{jh}^q &= \frac{\beta_h^{q-1} \exp(\sum_{igkl} v_{ig}^q z_{kl}^{q-1} \log f(x_{ijk}; \lambda_{ijk}^{q-1}))}{\sum_{h'} \beta_{h'}^{q-1} \exp(\sum_{igkl} v_{ig}^q z_{kl}^{q-1} \log f(x_{ijk}; \lambda_{ijk}^{q-1}))}, \\ z_{kl}^q &= \frac{\gamma_l^{q-1} \exp(\sum_{igjh} v_{ig}^q w_{jh}^q \log f(x_{ijk}; \lambda_{ijk}^{q-1}))}{\sum_{l'} \gamma_{l'}^{q-1} \exp(\sum_{igjh} v_{ig}^q w_{jh}^q (x_{ijk} \log f(x_{ijk}; \lambda_{ijk}^{q-1})))}. \end{aligned}$$

Mise à jour des paramètres:

$$\alpha_g^q = \frac{\sum_i v_{ig}^q}{I}, \beta_h^q = \frac{\sum_j w_{jh}^q}{J}, \gamma_l^q = \frac{\sum_k z_{kl}^q}{K}, \zeta_{ghl}^q = \frac{\sum_{ijk} v_{ig}^q w_{jh}^q z_{kl}^q (x_{ijk})}{\sum_{ijk} v_{ig}^q w_{jh}^q z_{kl}^q (\mu_i \nu_j \eta_k)}.$$

5 Résultats sur données simulées

Nous avons réalisé trente tri-clustering sur jeux de données simulées avec $N = J = K = 100$. Le nombre de cluster pour la simulation et l'exécution étaient fixés à $G = H = L = 3$. Concernant les paramètres, nous avons $\boldsymbol{\alpha} = (0.2, 0.3, 0.5)$, $\boldsymbol{\beta} = (0.25, 0.35, 0.4)$ et $\boldsymbol{\gamma} = (0.25, 0.35, 0.4)$. Les effets de blocs $\boldsymbol{\zeta}$ sont disponibles dans la Table 1.

Sur ces trente simulations, nous avons calculé les ARI en ligne, en colonne et en profondeur. Les moyennes respectives sont égales à 0.96 1.00 et 0.92. Cela montre que l'algorithme EM variationnel estime bien les variables latentes. En ce qui concerne les paramètres, il est difficile de juger s'ils ont été bien estimés lorsque les partitions trouvées ne sont pas les partitions initiales. Nous avons donc sélectionné les vingt cas (sur trente) où les aris en ligne, colonne et profondeur étaient égaux à 1 pour estimer la qualité de l'estimation des paramètres. Dans les vingt cas, les paramètres sont parfaitement estimés (à 0.00% d'erreur). Ces résultats sont très satisfaisant et montrent l'efficacité de l'algorithme proposé.

Table 1: Paramètres $\zeta \times 10^{-13}$ utilisés pour simuler les données.

depth-cluster 1	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	3.2	4.7	2.9
row-cluster 2	1.4	1.5	7.8
row-cluster 3	2.3	6.2	2.0

depth-cluster 1	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	4.6	2.5	2.3
row-cluster 2	6.8	2.3	2.3
row-cluster 3	3.3	4.3	3.0

depth-cluster 1	col-cluster 1	col-cluster 2	col-cluster 3
row-cluster 1	4.9	3.2	3.0
row-cluster 2	2.9	2.8	2.6
row-cluster 3	4.2	4.2	3.8

6 Conclusion

Ce papier propose une technique de tri-clustering efficace pour l'analyse non-supervisée de données de comptage dynamique. Pour compléter ce travail, nous pourrions nous intéresser à la définition d'un critère de sélection de modèle, notamment pour être capable de choisir automatiquement le nombre de clusters en ligne, en colonne et en profondeur. De plus, il serait intéressant d'appliquer ce modèle à des données réelles.

References

- [1] G. Govaert and M. Nadif. *Co-Clustering*. ISTE Ltd and John Wiley & Sons Inc., 2014.