Introduction à l'étude du modèle à blocs latents

Vincent Brault

29 juin 2011

Sous la direction de Gilles Celeux et Christine Keribin

1 Introduction

Le modèle à blocs latents (ou LBM) se retrouve dans des domaines très diversifiés comme le marketing, l'analyse d'expression de gènes ou encore l'analyse textuelle. L'intérêt de ce modèle est de pouvoir trier des tableaux simultanément sur les lignes et les colonnes mais il se heurte à des difficultés dans l'estimation et la sélection de la double structure cachée qu'il induit.

2 Estimation d'un modèle à blocs latents

Soit $x = \{(x_{ij})|i \in I, j \in J\}$ une matrice de données de dimension $n \times d$, où I est un ensemble de n objets (observations, cas) et J un ensemble de d variables (colonnes, attributs). Le but est d'opérer des permutations sur les objets et les variables pour construire une structure de correspondance sur $I \times J$.
Dans l'exemple de la figure 2 proposé par Govaert et Nadif dans un article de 2007 [4], nous supposons avoir un tableau binaire 10×7 dont nous voulons faire apparaître une structure (1). Nous pouvons faire une réorganisation sur les lignes (2) puis les colonnes (3). Dans cette dernière classification, nous voyons que nous pouvons résumer cette matrice par une autre de 3×3 (4).
L'objectif est de définir des partitions en lignes et en colonnes et permettre de

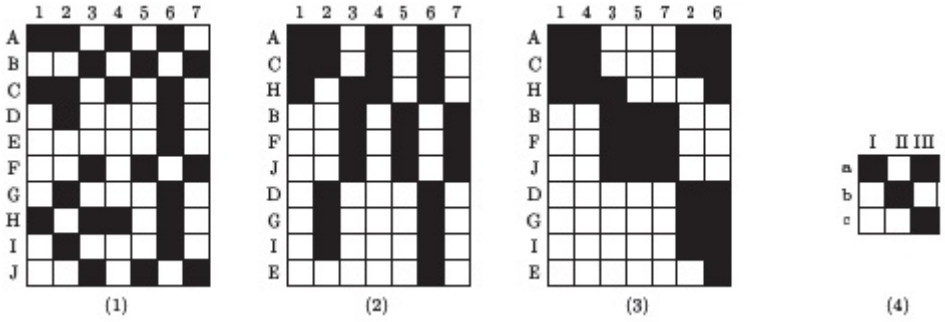

FIGURE 1 – Matrice de données binaires (1), réorganisées en partition sur I (2), en partitions simultanées sur I et J (3), et résumé des données binaires (4)

résumer l'information.

Pour bien fixer les idées, nous allons introduire un certain nombre de notations. Nous divison le tableau en g classes de l'ensemble I (donc sur les lignes) et nous utiliserons la matrice de classification notée $z = (z_{ik})_{i=1\dots n, k=1\dots g}$ où $z_{ik} = 1$ si et seulement si i appartient à la classe k et 0 sinon. Nous noterons $\pi_k = \mathbb{P}(z_{ik} = 1)$ la probabilité d'appartenir à la classe k. Nous pouvons alors constater que $\sum_{k=1}^{g} \pi_k = 1$.

De même, une partition en m classes de l'ensemble J sera représentée par la matrice de classification notée $w = (w_{jl})_{j=1\dots d, l=1\dots m}$ où $w_{jl} = 1$ si et seulement si j appartient à la classe l et 0 sinon. Nous noterons $\rho_l = \mathbb{P}(w_{jl} = 1)$.

Nous noterons $\mathcal{Z}$ l'ensemble des matrices de classifications possibles. C'est-à-dire que nous avons :

$$\mathcal{Z} = \left\{ z \in \{0,1\}^{n \times g} \,\middle|\, \forall i \in \{1, \dots, n\} \quad \sum_{k=1}^{g} z_{ik} = 1 \right\}$$

De même, les matrices w appartiennent à un ensemble similaire que nous noterons $\mathcal{W}$.

Remarque 2.0.1. Pour simplifier, nous ne mettrons plus les bornes dans les sommes et produits. Par exemple, nous noterons $\sum_i$ pour $\sum_{i=1}^{n}$.
Toutefois, pour le lecteur qui n'est pas encore familiarisé avec les notations, je rappelle les différentes bornes :

- i va de 1 à n et représente une ligne de la matrice x
- j va de 1 à d et représente une colonne de la matrice x
- k va de 1 à g et représente une classe de I (donc sur les lignes)
- l va de 1 à m et représente une classe de J (donc sur les colonnes)

2.1 Exemple

Les applications de ce modèle sont variées. Nous pouvons citer l'exemple d'un vidéo club qui souhaiterait trouver des groupes parmi ses clients regardant certains types de films. Ainsi, la case x_{ij} serait noire si l'utilisateur i a loué le film j. Une réorganisation simultanée des lignes et des colonnes permettrait de mettre en évidence des groupes d'utilisateurs ou de films et pouvoir mieux conseiller les clients.

2.2 Modèle à blocs latents

Dans le modèle à blocs latents que nous utilisons, nous devons commencer par faire un certain nombre d'hypothèses.

Hypothèse 2.2.1.

1. D'abord, nous voulons que les blocs soient rectangulaires et ne puissent pas se chevaucher (c'est-à-dire qu'ils forment un damier). Donc nous posons : $u_{ijkl} = z_{ik} w_{jl}$ avec u_{ijkl} vaut 1 si la $i$ème ligne est dans la classe k et la $j$ème colonne dans la classe l.
2. Nous supposons que les probabilités d'affectation des labels en ligne et en colonne sont indépendantes. De même pour la probabilité d'affectation de chaque ligne (et colonne) :

$$p(z, w) = p(z)p(w) = \prod_i p(z_i) \prod_j p(z_j) = \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}}$$

Nous pouvons constater que la loi des matrices de classification z est une multinomiale.

3. Nous supposons que, conditionnellement aux blocs kl, chaque case x_{ij} de la matrice est le résultat du tirage d'une variable X_{ij} de paramètre α_{kl}. C'est-à-dire que nous avons :

$$f(x|z,w;\alpha) = \prod_{i,j,k,l} \varphi(x_{ij};\alpha_{kl})^{z_{ik}w_{jl}}$$

où $\varphi(.;\alpha_{kl})$ la densité de X_{ij}.
Pour la suite, nous noterons $\theta = ((\alpha_{kl})_{k=1\dots g,l=1\dots m}, (\pi_k)_{k=1\dots g}, (\rho_l)_{l=1\dots m})$ l'ensemble des variables.

Le modèle à blocs latents est donc défini par :

$$\begin{aligned} f(x;\theta) &= \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} p(x,z,w;\theta) \\ &= \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} f(x|z,w;\theta)p(z,w;\theta) \\ &= \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} p(z,\theta)p(w,\theta)f(x|z,w;\theta) \end{aligned}$$

Remarque 2.2.2. Nous pouvons le voir comme une extension du modèle de mélange

Le cas particulier que nous allons utiliser est le modèle à blocs latents de Bernoulli défini par :

$$f(x;\theta) = \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} \varphi(x_{ij};\alpha_{kl})^{z_{ik}w_{jl}}$$

$$\text{avec} \quad \varphi(x_{ij};\alpha_{kl}) = \alpha_{kl}^{x_{ij}}(1-\alpha_{kl})^{1-x_{ij}}$$

2.3 Propriétés

Comme pour les modèles de mélange, nous voyons que le LBM ne sont pas identifiables à cause du label switching. Keribin a démontré que nous avons une identifiabilité générique sous les conditions suivantes :

1. Le nombre de lignes (et resp. de colonnes) doit être supérieur au double du nombre de classes en colonnes (et resp en lignes). C'est-à-dire que $n \geq 2m$ (et $d \geq 2g$).
2. Pour tout $1 \leq k \leq g$, la probabilité d'appartenir à la classe k est non nulle ($\pi_k > 0$) et les coordonnées du vecteur $r = \alpha\rho$ sont toutes distinctes.
3. De même sur les colonnes, c'est-à-dire que $\forall 1 \leq l \leq m$, $\rho_l > 0$ et les coordonnées du vecteur ${}^t\pi\alpha$ sont toutes distinctes.

Et le calcul numérique de la vraisemblance est difficile :

$$\begin{aligned} L(\theta) &= \log(f(x;\theta)) \\ &= \log\left(\sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} \left(\alpha_{kl}^{x_{ij}}(1-\alpha_{kl})^{1-x_{ij}}\right)^{z_{ik}w_{jl}}\right) \end{aligned}$$

Nous devons donc faire une somme sur tous les couples de matrices de $\mathcal{Z} \times \mathcal{W}$ donc nous avons un nombre d'opérations de l'ordre de grandeur de $g^n \times m^d$ (pour vous donner une idée : $3^{100} \times 2^{60} = 5.9e + 65$). Ceci n'est pas calculable avec les ordinateurs que nous possèdons actuellement et nous aurons des problèmes d'approximations.

3 Les algorithmes

Le modèle peut être vu comme un modèle de mélange dont il y a deux variables latentes Z et W. Par conséquent, l'algorithme EM semble être un bon point de départ pour le résoudre. Dans cette partie, je vais commencer par rappeler le principe de cet algorithme puis expliqeur les deux méthodes principales qui étaient mises en oeuvre lorsque j'ai débuté mon stage et enfin, j'expliquerai l'algorithme EP qui est celui que j'ai essayé d'adapter.

3.1 L'algorithme EM

Le principe de l'algorithme EM est de trouver de façon itérative l'estimateur du maximum de vraisemblance. En reprenant les calculs de Dempster [3], nous avons :

$$\begin{aligned} L(\theta) &= \log(f(x;\theta)) \\ &= \log\left(\frac{f(x,w,z;\theta)}{f(w,z|x;\theta)}\right) \\ &= \log f(x,w,z;\theta) - \log f(w,z|x;\theta) \\ &= \mathbb{E}\left[\log p(x,w,z;\theta)|x;\theta^{(c)}\right] - \mathbb{E}\left[\log p(w,z|x;\theta)|x;\theta^{(c)}\right] \\ &= Q(\theta|\theta^{(c)}) - H(\theta|\theta^{(c)}) \end{aligned}$$

Nous pouvons voir que la logvraisemblance est la différence de deux termes positifs et que donc si nous maximisons la quantité Q, nous aurons aussi une augmentation de L.
On montre que pour tout θ, on a $H(\tilde{\theta}|\theta^{(c)}) \leq H(\theta^{(c)}|\theta^{(c)})$ donc si on prend $\tilde{\theta}$ maximisant $Q(\theta|\theta^{(c)})$, nous avons :

$$L(\tilde{\theta}) - L(\theta^{(c)}) = Q(\tilde{\theta}|\theta^{(c)}) - Q(\theta^{(c)}|\theta^{(c)}) - H(\tilde{\theta}|\theta^{(c)}) + H(\theta^{(c)}|\theta^{(c)}) \geq 0$$

Alors, si nous trouvons à chaque étape ce $\tilde{\theta}$, nous sommes sûr de faire augmenter la logvraisemblance. Une variante consiste juste à prendre un $\tilde{\theta}$ qui augmenterait Q.
L'algorithme EM itère les deux étapes suivantes jusqu'à convergence :

1. Etape Estimation : détermination de la loi $p(z,w|x,\theta^{(c)})$ avec $\theta^{(c)}$ fixé et calcul de l'esperance
2. Etape Maximisation : maximisation de $Q(\theta|\theta^{(c)})$ en θ et prendre $\theta^{(c+1)} = \tilde{\theta}$

Faisons le calcul de $Q(\theta|\theta^{(c)})$ dans le cadre du LBM :

$$Q(\theta|\theta^{(c)}) = \sum_{i,k} s_{ik}^{(c)} \log \pi_k + \sum_{j,l} t_{jl}^{(c)} \log \rho_l + \sum_{i,j,k,l} e_{ijkl}^{(c)} \log(\varphi(x_{ij};\alpha_{kl}))$$

$$\text{avec } s_{ik}^{(c)} = P\left(z_{ik} = 1|\theta^{(c)}, X = x\right),\ t_{jl}^{(c)} = P\left(w_{jl} = 1|\theta^{(c)}, X = x\right)$$

$$\text{et } e_{ijkl}^{(c)} = P(z_{ik}w_{jl} = 1|\theta^{(c)}, X = x)$$

Remarque 3.1.1. Le problème qui empêche d'utiliser l'algorithme EM réside dans e_{ijkl} qui demande la sommation sur toutes les variables latentes et qui demanderait un temps de calcul très élevé.

3.2 Algorithme VEM

Pour cette partie, je vais reprendre la formulation faite par Keribin dans son article [5]. Nous allons réécrire la logvraisemblance en introduisant une distribution libre q_{zw} des variables latentes :

$$\begin{aligned}
L(\theta) &= \log(p(x;\theta)) \\
&= \int q_{zw}(z,w)\log(p(x;\theta))dzdw \\
&= \int q_{zw}(z,w)\log\left(p(x,z,w;\theta)\frac{p(x;\theta)}{p(x,z,w;\theta)}\right)dzdw \\
&= \int q_{zw}(z,w)\log\left(\frac{p(x,z,w;\theta)}{q_{zw}(z,w)}\frac{q_{zw}(z,w)}{p(z,w|x;\theta)}\right)dzdw \\
&= \int q_{zw}(z,w)\log\left(\frac{p(x,z,w;\theta)}{q_{zw}(z,w)}\right)dzdw + \int q_{zw}(z,w)\log\left(\frac{q_{zw}(z,w)}{p(z,w|x;\theta)}\right)dzdw \\
&= \mathbb{E}_{q_{zw}}\left[\log\left(\frac{p(x,Z,W;\theta)}{q_{zw}(Z,W)}\right)\right] + D(q_{zw}||p(z,w|x;\theta)) \\
&= \mathcal{F}(q_{zw};\theta) + D(q_{zw}||p(z,w|x;\theta))
\end{aligned}$$

La log vraisemblance est donc la somme d'une fonction $\mathcal{F}$ de la distribution libre que nous appellerons énergie libre et de la dissemblance de Kullback. De plus, cette dissemblance est toujours positive donc nous avons $L(\theta) \geq \mathcal{F}(q_{zw};\theta)$ pour toute distribution q_{zw}. Ainsi, chercher à maximiser $\mathcal{F}$ en q_{zw} permet d'avoir une borne inférieure de $L(\theta)$ et nous pouvons voir que $\mathcal{F}(p(z,w|x;\theta);\theta) = L(\theta)$. Toutefois, le calcul de l'énergie libre n'est pas forcément possible quelque soit la forme de q_{zw} en particulier pour $p(z,w|x;\theta)$.
Dans l'algorithme proposé par Govaert et Nadif (2007) [4], nous supposons que $q_{zw}(z,w) = q_z(z)q_w(w) = \prod_i q_z(z_i)\prod_j q_w(w_j)$. En écrivant ceci, nous allons réduire nos calculs à une famille ne contenant pas la probabilité désirée $p(z,w|x;\theta)$. Cette factorisation s'appelle l'approximation en champs moyen. Ainsi, nous avons :

$$s_{ik}^{(c)} \approx P_{q_z}\left(z_{ik} = 1;\theta^{(c)}\right), \quad t_{jl}^{(c)} \approx P_{q_w}\left(w_{jl} = 1;\theta^{(c)}\right)$$

$$\text{et} \quad e_{ijkl}^{(c)} \approx s_{ik}^{(c)}t_{jl}^{(c)}$$

Et nous obtenons l'algorithme suivant :

1. Etape E : Maximisation de l'énergie libre jusqu'à convergence :
   (a) Calcul de s_{ik} à t_{jl} et $\theta^{(c)}$ donnés
   (b) Calcul de t_{jl} à s_{ik} et $\theta^{(c)}$ donnés

On obtient $s^{(c+1)}$ et $t^{(c+1)}$

2. Etape M : Mise à jour de $\theta^{(c+1)}$

Remarque 3.2.1. Toutefois, comme nous savons que nous faisons une approximation, nous allons devoir évaluer la qualité des résultats obtenus. Deux critères pour évaluer cette approximation sont proposés dans les sections suivantes.
De plus, l'un des problèmes de cet algorithme est qu'il est très sensible aux conditions initiales et peut se faire piéger dans un maximum local.
En revanche, il est très rapide et permet de trier de grands tableaux en un temps relativement correct.

3.3 Algorithme SEM-Gibbs

Le principe de cet algorithme est de remplacer l'étape E par le tirage d'un échantillon des données manquantes sous la loi $p(z, w|x; \theta^{(c)})$.
Cette loi n'étant pas disponible, Keribin a utilisé un échantillonneur de Gibbs pour la simulation dont nous rapelons le principe :
On répète un certain nombre de fois :

1. tirer $Z^{(t+1)}$ suivant la loi $p(z|x, w^{(t)}; \theta^{(c)})$
2. tirer $W^{(t+1)}$ suivant la loi $p(w|x, z^{(t+1)}; \theta^{(c)})$

La loi stationnaire de la chaîne est alors $p(z, w|x; \theta^{(c)})$.
Nous obtenons ainsi l'algorithme suivant :

1. Etape E-S : On répète les deux étapes suivantes jusqu'à obtenir la loi stationnaire :
   (a) Estimation de la loi $p(z|x, w^{(t)}; \theta^{(c)})$ puis tirage de $Z^{(t+1)}$
   (b) Estimation de la loi $p(w|x, z^{(t+1)}; \theta^{(c)})$ puis tirage de $W^{(t+1)}$

   Nous obtenons $w^{(c+1)}$ et $z^{(c+1)}$
2. Etape M : Mise à jour de $\theta^{(c+1)}$

Remarque 3.3.1. L'avantage de SEM-Gibbs est qu'il est moins sensible aux conditions initiales mais il est évident que nous n'améliorons pas la logvraisemblance à chaque fois. SEM génère une chaine de Markov irréductible avec une unique distribution stationnaire ainsi, les paramètres vont fluctuer autour de l'EMV. On peut aussi se poser la question de l'influence du nombre de pas du schéma de Gibbs sur le résultat obtenu mais pour l'instant Keribin a montré qu'un seul pas suffit.

3.4 Initialisation du VEM par un SEM

L'algorithme SEM trouve une zone correcte pour le $\hat{\theta}$ mais pas la bonne valeur et VEM trouve un $\hat{\theta}$ très proche de la réalité quand nous lui donnons une initialisation proche du résultat réel. Nous avons montré durant mon stage qu'utiliser l'algorithme VEM avec une initialisation de SEM donne d'excellents résultats sur le calcul de $\hat{\theta}$ et sur la classification. Toutefois, ceci a été fait sur des données simulées et nous devons maintenant regarder la qualité sur des données réelles. Pour ceci, nous devons chercher un moyen de calculer ou d'approcher la logvraisemblance.

4 Étude de l'erreur de classification

Dans le paragraphe précédent, nous avons présenté la qualité des approximations de $\hat{\theta}$. Dans celui ci, nous allons nous concentrer sur la qualité des partitions obtenues. Pour cela, nous avons choisi d'utiliser la distance suivante que nous appelons erreur de classification et qui prend en entrée deux partitions $u = (z, w)$ et $u' = (z', w')$ pour renvoyer :

$$\delta(u, u') = 1 - \frac{1}{n \times d} \sum_{i,j,k,l} z_{ik} w_{jl} z'_{ik} w'_{jl}$$

Nous pouvons voir que cette erreur est plutôt sévère car si nous classons mal les lignes mais correctement les colonnes alors l'erreur sera quand même de 1.

4.1 Propriété de base

Dans un premier temps, nous allons reprendre la distance de deux matrices z et z' introduite par Govaert et Nadif [4]. Pour deux matrices z et z' qui ont le même nombre de lignes et de colonnes, nous allons noter :

$$e(z, z') = 1 - \frac{1}{n} \sum_{i,k} z_{ik} z'_{ik}$$

La distance sur les couples des matrices peut s'écrire à l'aide d'une relation simple entre les distances des matrices respectives. Plus précisément, nous avons pour deux partitions $(u, u') = ((z, w), (z', w'))$:

$$\delta(u, u') = e(z, z') + e(w, w') - e(z, z') \times e(w, w')$$

4.2 Erreur avec permutation

L'un des problèmes de l'erreur définie précédemment est que si nous avons les mêmes répartitions à une permutation près, nous pouvons avoir une erreur de 1, ceci vient du fait que notre modèle n'est pas identifiable. Pour contourner cette difficulté, nous travaillons plutôt sur l'erreur suivante :

Définition 4.2.1. Pour deux partitions z et z' qui ont le même nombre de composantes, nous allons noter :

$$e_{\mathcal{P}}(z, z') = 1 - \frac{1}{n} \max_{\sigma \in \mathcal{P}} \left(\sum_{i=1}^{n} \sum_{k=1}^{g} z_{ik} z'_{i\sigma(k)} \right)$$

4.3 Simulation

Nous avons montré empiriquement que les distances entre deux matrices de répartitions tendent vers des valeurs constantes quand nous augmentons n en fixant d et que $\delta_{\mathcal{P}}(u, u')$ tend vers 0 lorsque nous augmentons les deux en même temps. Toutefois, les résultats théoriques n'ont pas encore abouti.

5 Critère de sélection de modèles

Un problème important que nous devons résoudre est celui du nombre de composantes. En effet, quand on obtient un tableau à trier, nous ne connaissons pas forcément ce paramètre et nous proposons d'utiliser un critère de sélection de modèles.

5.1 AIC et BIC

Deux critères usuellement utilisés à l'heure actuelle sont Akaike's Information Criterion (AIC) et Bayesian Information Criterion (BIC)[8]. Ces deux critères sélectionnent le modèle minimisant la logvraisemblance pénalisée. Si nous notons k le nombre de paramètres du modèle $\mathcal{M}$, n le nombre d'observations et L la vraisemblance alors les critères sont définis ainsi :

$$\begin{aligned} AIC(\mathcal{M}) &= -2\log(L) + 2k \\ BIC(\mathcal{M}) &= -2\log(L) + k\log(n) \end{aligned}$$

Or, nous savons que le calcul de la logvraisemblance n'est pas réalisable.
Toutefois, nous avons une approximation avec la fonction énergie $\mathcal{F}$ mais nous ne savons pas comment évolue $\mathcal{F} - \log(L)$ en fonction des modèles. Est-elle constante ? Si oui, est-ce que cette différence est grande ou non ? Si non, croît-elle avec la complexité ?
Pour tenter de résoudre ce problème, nous avons commencé à faire des simulations qui ne sont pas encore concluantes à l'heure actuelle mais c'est un problème important à résoudre d'autant qu'il permettra aussi de quantifier la qualité des résultats produits par les algorithmes de la première partie.

5.2 Critère ICL

Nous avons cherché à adapter au LBM le critère ICL (Integrated Classification Likelihood) décrit dans l'article de Biernacki, Celeux et Govaert en 2010 [1]. Pour cela, nous allons noter θ les paramètres et z les variables latentes. Ainsi le critère ICL cherche à maximiser la valeur $ICL(\mathcal{M}) = \log\left(p\left(x, \hat{z}\right)\right)$ où $\hat{z}$ est le maximum a posteriori pour $\hat{\theta}$ (qui lui même maximise la vraisemblance).

5.2.1 Théorie

Pour la suite, nous arrivons dans un cadre bayésien puisque nous allons mettre des loi a priori sur les paramètres. Nous mettons des lois a priori de Dirichlet équilibrés sur chacun des paramètres et nous obtenons un critère explicite en prenant les notations suivantes :

- Nous allons noter $n_k = \sum_i z_{ik}$ le nombre de lignes dans la $k$ème classe.
- Nous allons noter $n^l = \sum_j w_{jl}$ le nombre de colonnes dans la $l$ème classe.
- Nous allons noter $n_k^l = \sum_{i,j} z_{ik} w_{jl} x_{ij}$ le nombre de cases noires dans le bloc (k, l) et $N_k^l = n_k n^l$ le nombre total de cases

Ainsi, le critère ICL devient :

$$\begin{aligned} ICL(\mathcal{M}) &= \log p(x, \hat{z}, \hat{w}) \\ &= \log\Gamma(g/2) + \log\Gamma(m/2) + (2mg - m - g)\log\Gamma(1/2) - \log\Gamma(n + g/2) - \log\Gamma(d + m/2) \\ &\quad + \sum_{k=1}^{g} \log\Gamma(n_k + 1/2) + \sum_{l=1}^{m} \log\Gamma(n^l + 1/2) \\ &\quad + \sum_{k,l} \log\Gamma(1/2 + n_k^l) + \log\Gamma(1/2 + N_k^l - n_k^l) - \log\Gamma(N_k^l + 1) \end{aligned}$$

Le problème est de choisir correctement le couple $(\hat{z}, \hat{w})$ puisque dans leur article sur les modèles de mélanges Celeux et co auteurs choisissent le z qui revient le plus fréquemment. Ce critère semble pour l'instant donner des résultats cohérents mais nous devons continuer à l'étudier et le confronter à plus de cas.

6 Conclusion et perspectives

Le modèle à blocs latents conduit à la recherche simultanée de très grands tableaux de données et se heurte donc à des difficultés dans l'estimation et la sélection de la double structure cachée qu'il induit ; l'objet de ma thèse sera de contribuer à la résolution de ces difficultés.
Comme nous l'avons vu, le calcul de l'estimateur du maximum de vraisemblance par l'algorithme EM s'avère impraticable et nous utilisons pour l'instant une approximation variationnelle et des versions stochastiques. Mon travail consistera également à étudier les qualités d'approximation de ces algorithmes du point de vue théorique (étude asymptotique sous différentes conditions) et pratique (évaluer les performances sur des données simulées et réelles).
La plupart des critères de sélection utilisant la valeur du maximum de vraisemblance non disponible pour ce modèle, mon travail consistera à construire des critères de sélection de modèles surmontant cette difficulté et d'étudier leur optimalité ainsi que le comportement pratique. En particulier, il me faudra observer le critère ICL que nous avons adapté durant mon stage.
Enfin, je serai amené à expérimenter mes innovations sur des données réelles d'analyse du transciptome dans le cadre d'une collaboration avec l'URGV sur la recherche de fonctions de gènes orphelins.

Références

[1] C. Biernacki ; G. Celeux and G. Govaert, *Exact and Monte Carlo calculations of integrated likelihoods for the latent class model*, Journal of Statistical Planning and Inference, 2010.

[2] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[3] Dempster A.P. ; Laird N.M. et Rubin D.B. *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1), 1-38.

[4] Gérard Govaert et Mohamed Nadif, *Block clustering with Bernoulli mixture models : Comparaison of different approaches*, http ://www.sciencedirect.com/, 2007.

[5] Christine Keribin, *Les méthodes bayésiennes variationnelles et leur application en neuroimagerie : une étude de l'existant*, http ://hal.archives-ouvertes.fr/inria-00430289/en/, 2009.

[6] Christine Keribin et Gérard Govaert et Gilles Celeux, *Estimation d'un modèle à blocs latents par l'algorithme SEM*, http ://hal.archives-ouvertes.fr/inria-00494796/en/, 2010.

[7] Malte Kuss and Carl Edward Rasmussen, *Assessing Approximations for Gaussian Process Classification*, 2005.

[8] Emilie Lebarbier and Tristan Mary-Huard, *Le critère BIC : fondements théoriques et interprétation*, http ://hal.archives-ouvertes.fr/docs/00/07/06/85/PDF/RR-5315.pdf, 2005.

[9] Thomas Minka, *A family of algorithms for approximate Bayesian inference*, Ph. D. thesis, MIT, 2001.

[10] Thomas Minka, *Power EP*, Technical Report MSR-TR-2004-149, Microsoft Research Cambridge, 2004.