Apprentissage non Supervisé d'un Mélange Discret et Fini Basé sur la Distribution de Dirichlet Multinomiale: Application à la Modélisation de Texture

Unsupervised Learning of a Finite Discrete Mixture Model Based on the Multinomial Dirichlet Distribution: Application to Texture Modeling

Nizar Bouguila, Djemel Ziou
DMI, Faculté des Sciences
Université de Sherbrooke
{bouguila, ziou}@dmi.usherb.ca

Résumé

Ce papier présente un nouveau modèle de mélange basé sur la distribution Multinomiale Dirichlet (MDD). Pour l'estimation des paramètres de ce mélange, On propose un algorithme non supervisé basé sur le maximum de vraisemblance et la méthode de Fisher. Ce mélange est utilisé pour produire un nouveau modèle de texture. Les résultats expérimentaux concernent le résumé des images de textures et sont appliqués pour la base de données Vistex de MIT Media Lab.

Mots Clef

Mélange de Dirichlet Multinomiale, Maximum de Vraisemblance, Méthode de Fisher, Texture, Matrice de Cooccurrence, *Vistex*.

Abstract

This paper presents a new finite mixture model based on the Multinomial Dirichlet distribution (MDD). For the estimation of the parameters of this mixture we propose an unsupervised algorithm based on the Maximum Likelihood (ML) and Fisher scoring methods. This mixture is used to produce a new texture model. Experimental results concern texture images summarizing and are reported on the Vistex texture image database from the MIT Media Lab.

Keywords

Multinomial Dirichlet Mixture, Maximum Likelihood, Fisher's Scoring method, Texture, Cooccurrence Matrix, *Vistex*.

1 Introduction

Les activités humaines et scientifiques génèrent beaucoup de données. Ces données peuvent être incomplètes, redondantes ou erronées. Les méthodes statistiques de reconnaissance des formes sont particulièrement utiles pour comprendre la signification de ces données. Les mélanges finis de densités de probabilité apparaissent comme étant les modèles fondamentaux dans le domaine statistique de reconnaissance des formes. Un modèle de mélange consiste à plusieurs fonctions de densités de probabilité. Ces fonctions sont dites les composantes du mélange et choisies comme étant des Gaussiennes dans la majorité des cas. Toutefois, le mélange de Gaussiennes n'est pas le bon choix dans toutes les applications et il échoue à trouver les *bonnes* structures quand les partitions sont clairement non Gaussiennes [22]. Dans [5] on a montré que la distribution de Dirichlet peut être un trés bon choix pour surmonter les inconvénients de la Gaussienne. Dans ce papier, on présente une autre distribution, basée sur les densités Multinomiale et de Dirichlet, qu'on appelle la MDD (la Distribution Multinomiale Dirichlet). On prouve à travers un nouveau modèle de texture, que contrairement à la Gaussienne, la MDD est un choix excellent pour modéliser les données discrètes (vecteurs de comptage). Afin de résoudre le problème de l'estimation des paramètres du mélange de MDD, nous proposons un algorithme basé sur la méthode de Fisher et nous utilisons un critère d'entropie pour déterminer le nombre de composantes.
Le papier est organisé comme suit. La section suivante décrit le mélange de MDD en détails. Dans le section 3, on propose une méthode pour estimer les paramètres de ce mélange. Dans la section 4, on présente une méthode d'initialisation des paramètres et on donne l'algorithme d'estimation complet. La section 5 est consacrée aux résultats expérimentaux. On termine le papier avec quelques remarques et conclusions.

2 Le Mélange Multinomiale Dirichlet

Soit $\vec{c} = (c_1, \ldots, c_{dim})$ un vecteur de comptage (e.g. la fréquence de dim caractéristiques dans un document). le vecteur $\vec{c}$ suit une distribution Multinomiale de paramètre $\vec{P} = (P_1, \ldots, P_{dim})$:

$$p(\vec{c}|\vec{P}) = \frac{|\vec{c}|!}{c_1!c_2!\ldots c_{dim}!} \prod_{k=1}^{dim} P_k^{c_k} \qquad (1)$$

Le conjugué à priori de $\vec{P}$ est la distribution de Dirichlet:

$$p(\vec{P}) = D(\alpha_1,\ldots,\alpha_{dim}) = \frac{\Gamma(|\vec{\alpha}|)}{\prod_{k=1}^{dim}\Gamma(\alpha_k)}\prod_{k=1}^{dim} P_k^{\alpha_k-1} \tag{2}$$

where

$$|\vec{\alpha}| = \sum_{k=1}^{dim}\alpha_k \tag{3}$$

$$P_k > 0 \tag{4}$$

$$\sum_{k=1}^{dim} P_k = 1 \tag{5}$$

La distribution Beta est un cas spécial de la distribution de Dirichlet lorsque $k = 2$. La moyenne et la variance de la distribution de Dirichlet sont données par les formules suivantes:

$$E(P_i) = \frac{\alpha_i}{|\vec{\alpha}|} \tag{6}$$

$$Var(P_i) = \frac{\alpha_i(|\vec{\alpha}| - \alpha_i)}{|\vec{\alpha}|^2(|\vec{\alpha}|+1)} \tag{7}$$

et la covariance entre P_i et P_j est:

$$Cov(P_i,P_j) = \frac{-\alpha_i\alpha_j}{|\vec{\alpha}|^2(|\vec{\alpha}|+1)} \tag{8}$$

Prenons la Dirichlet comme densité à priori, la densité jointe est:

$$p(\vec{c},\vec{P}|\vec{\alpha}) = \frac{\Gamma(|\vec{c}|+1)\Gamma(|\vec{\alpha}|)}{\prod_{k=1}^{dim}\Gamma(\alpha_k)\Gamma(c_k+1)}\prod_{k=1}^{dim} P_k^{\alpha_k+c_k-1} \tag{9}$$

alors [5]:

$$p(\vec{c}|\vec{\alpha}) = \frac{\Gamma(|\vec{c}|+1)\Gamma(|\vec{\alpha}|)}{\Gamma(|\vec{c}|+|\vec{\alpha}|)}\prod_{k=1}^{dim}\frac{\Gamma(c_k+\alpha_k)}{\Gamma(\alpha_k)\Gamma(c_k+1)} \tag{10}$$

On appelle cette densité la MDD.
Un mélange de MDD avec M composantes est défini comme suit:

$$p(\vec{c}/\Theta) = \sum_{j=1}^{M} p(\vec{c}/j,\Theta_j)P(j) \tag{11}$$

avec $P(j)$ ($0 < P(j) < 1$ et $\sum_{j=1}^{dim} P(j) = 1$) sont les proportions et $p(\vec{c}/j,\Theta_j)$ est la MDD. Le symbole Θ représente tout l'ensemble des paramètres à estimer:

$$\Theta = (\vec{\alpha_1},\ldots,\vec{\alpha_M},P(1),\ldots,P(M))$$

avec $\vec{\alpha_j}$ est le vecteur de paramètres pour la $j^{ième}$ population. Dans le développement qui suit, on utilise la notation suivante: $\Theta_j = (\vec{\alpha_j},P(j))$ pour j = 1 ... M.

3 Estimation Basée sur le Maximum de Vraisemblance

Le problème qui consiste à estimer les paramètres d'un mélange était le sujet d'études diverses [6] [14]. Durant les deux dernières décénies, la méthode de maximum de vraisemblance est devenue l'approche la plus utilisée pour ce genre de problèmes. Parmi la variété des méthodes itératives qui étaient suggérées comme alternatives pour optimiser les paramètres d'un mélange, le EM (Expectation Maximization) est l'approche la plus utilisée. Le EM a été proposé par Dempster et al. [7] pour estimer le maximum de vraisemblance des modèles stochastiques. Cet algorithme donne une procédure itérative dont la forme pratique est en géneral simple. Toutefois, il présente l'inconvénient suivant: le besoin de spécifier le nombre de composantes chaque fois. Pour résoudre ce problème, beaucoup de critères ont été proposés, comme le AIC [9], MDL [10] et BIC [8]. Le maximum de vraisemblance associé à un échantillon d'observations est le choix de paramètres qui maximisent la fonction de densité de probabilité de l'échantillon. Alors, avec l'estimation basée sur le maximum de vraisemblance, le problème de déterminer Θ devient:

$$max_\Theta p(\vec{c}/\Theta) \tag{12}$$

avec les contraintes: $\sum_{j=1}^{M} P(j) = 1$ et $P(j) > 0 \quad \forall j \in [1,M]$. Ces contraintes permettent de prendre en considération les probabilités *a priori* $P(j)$. En utilisant les multiplicateurs de Lagrange, on maximise la fonction suivante:

$$\begin{aligned}\Phi(\vec{c},\Theta,\Lambda) &= ln(p(\vec{c}/\Theta)) + \Lambda(1-\textstyle\sum_{i=1}^{M} P(i)) \\ &+ \mu\textstyle\sum_{j=1}^{M} P(j)ln(P(j))\end{aligned} \tag{13}$$

avec Λ est le multiplicateur de Lagrange. Pour faciliter le calcul, on a remplacé la fonction $p(\vec{c}/\Theta)$ dans l'équation 12 par la fonction $ln(p(\vec{c}/\Theta))$. Si on suppose qu'on a N vecteurs donnés $\vec{c_i}$ qui sont indépendents, on peut écrire:

$$p(\vec{c}/\Theta) = \prod_{i=1}^{N} p(\vec{c_i}/\Theta) \tag{14}$$

$$p(\vec{c_i}/\Theta) = \sum_{j=1}^{M} p(\vec{c_i}/j,\Theta_j)P(j) \tag{15}$$

Remplaçons les équations 14 et 15, on obtient:

$$\begin{aligned}&\Phi(\vec{c},\Theta,\Lambda) = \textstyle\sum_{i=1}^{N} ln(\sum_{j=1}^{M} p(\vec{c_i}/j,\Theta_j)P(j)) \\ &+\Lambda(1-\textstyle\sum_{j=1}^{M} P(j)) + \mu\textstyle\sum_{j=1}^{M} P(j)ln(P(j))\end{aligned} \tag{16}$$

Afin de déterminer automatiquement le nombre de composantes qui modélisent le mélange, on utilise un critère d'entropie qui a été dejà utilisé dans le cas de mélange de Gaussiennes [11]. Alors, le premier terme dans l'équation 16 est le logarithme de la fonction de vraisemblance, et il atteint son maximum global lorsque chaque composante représente seulement un vecteur de caractéristiques. Le dernier terme

(entropie) atteint son maximum lorsque tous les vecteurs de caractéristiques sont modélisés par une seule composante, càd, quand $P(j1) = 1$ pour $j1$ et $P(j) = 0, \forall j, j \neq j1$. L'algorithme commence avec un grand nombre de composantes dans le mélange, et tant qu'il procède, les composantes se concurrencent pour modéliser les données. Le choix de μ est critique pour la performance de l'algorithme, puisqu'il spécfie un compromis entre la vraisemblance des données et le nombre de composantes à trouver. On choisis μ comme étant le rapport entre le premier et le dernier terme dans l'équation 16 dans chaque itération t, càd,

$$\mu(t) = \frac{\sum_{i=1}^{N} ln(\sum_{j=1}^{M} p^{t-1}(\vec{c_i}/j,\Theta_j)P^{t-1}(j))}{\sum_{j=1}^{M} P^{t-1}(j)ln(P^{t-1}(j))} \quad (17)$$

Pour résoudre ce problème d'optimisation, il faut déterminer la solutions des équations suivantes:

$$\frac{\partial}{\partial\Theta}\Phi(\vec{c},\Theta,\Lambda) = 0 \quad (18)$$

$$\frac{\partial}{\partial\Lambda}\Phi(\vec{c},\Theta,\Lambda) = 0 \quad (19)$$

En calculant la dérivée par rapport à Θ_j, on obtient [12]:

$$\frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda) = \sum_{i=1}^{N} p(j/\vec{c_i},\Theta_j)\frac{\partial}{\partial\Theta_j}ln(p(\vec{c_i}/j,\Theta_j)) \quad (20)$$

avec $p(j/\vec{c_i},\Theta_j)$ est la probabilité a posteriori.
Comme $p(\vec{c_i}/j,\vec{\alpha_j})$ est indépendant de $P(j)$, alors on obtient [12]:

$$P(j)^{new} = \frac{\sum_{i=1}^{N} p^{old}(j/\vec{c_i},\vec{\alpha_j}) + \mu(p(j)^{old}(1 + ln(p(j)^{old})))}{N + \mu\sum_{j=1}^{M}(p(j)^{old}(1 + ln(p(j)^{old})))} \quad (21)$$

Pour estimer les paramètres $\vec{\alpha}$ on va utiliser la méthode de Fisher [13]. Cette approche est une variante de la méthode de Newton-Raphson. En effet, l'équation 18 peut être approximé par un développement limité au point Θ_{j_0}:

$$\begin{aligned} &\frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda) \simeq \frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda)(\Theta_{j_0}) \\ + \quad &(\Theta_j - \Theta_{j_0})\frac{\partial^2}{\partial\Theta_j^2}\Phi(\vec{c},\Theta,\Lambda)(\Theta_{j_0}) \end{aligned} \quad (22)$$

Comme $\frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda) = 0$, alors

$$\Theta_j \simeq \Theta_{j_0} - (\frac{\partial^2}{\partial\Theta_j^2}\Phi(\vec{c},\Theta,\Lambda)(\Theta_{j_0}))^{-1}\frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda)(\Theta_{j_0}) \quad (23)$$

Donc une mise à jours de l'estimation de $\hat{\Theta}_j^{(k+1)}$ en tenant compte de $\hat{\Theta}_j^{(k)}$ est donnée par:

$$\begin{aligned} \hat{\Theta}_j^{(k+1)} = \hat{\Theta}_j^{(k)} \quad &- \quad (\frac{\partial^2}{\partial\Theta_j^2}\Phi(\vec{c},\Theta,\Lambda)(\Theta_j))^{-1}_{\Theta_j=\hat{\Theta}_j^{(k)}} \\ &\times \quad (\frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda)(\Theta_j))_{\Theta_j=\hat{\Theta}_j^{(k)}} \end{aligned} \quad (24)$$

Ce qui est équivalent à:

$$\hat{\Theta}_j^{(k+1)} = \hat{\Theta}_j^{(k)} - H^{-1}(\hat{\Theta}_j^{(k)})(\frac{\partial}{\partial\Theta_j}\Phi(\vec{c},\Theta,\Lambda)(\Theta_j))_{\Theta_j=\hat{\Theta}_j^{(k)}} \quad (25)$$

avec H est la matrice de Hessian evaluée à l'estimation courante. Une variante de cette approche est la méthode de Fisher, où le Hessian H est remplacé par l'opposé de la matrice de l'information de Fisher. La méthode de Fisher est basé sur la première et la deuxième dérivé de la fonction de maximum de vraisemblance. Selon l'quation 20 on a:

$$\frac{\partial}{\partial\alpha_{jl}}\Phi(\vec{c},\Theta,\Lambda) = \sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j})\frac{\partial}{\partial\alpha_{jl}}ln(p(\vec{c_i}/j,\vec{\alpha_j})) \quad (26)$$

La dérivé de $ln(p(\vec{c_i}/j,\vec{\alpha_j})$ par rapport à α_{jl} est la suivante [12]:

$$\begin{aligned} \frac{\partial}{\partial\alpha_{jl}}ln(p(\vec{c_i}/j,\vec{\alpha_j})) \quad &= \quad \Psi(|\vec{\alpha_j}|) - \Psi(|\vec{c_i}| + |\vec{\alpha_j}|) \\ &+ \quad \Psi(c_{il} + \alpha_{jl}) - \Psi(\alpha_{jl}) \end{aligned} \quad (27)$$

avec $\Psi(.)$ est la fonction digamma. Alors:

$$\begin{aligned} &\frac{\partial}{\partial\alpha_{jl}}\Phi(\vec{c},\Theta,\Lambda) \\ = \quad &(\Psi(|\vec{\alpha_j}|) - \Psi(\alpha_{jl}))\sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j}) \\ + \quad &\sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j})(\Psi(c_{il} + \alpha_{jl}) - \Psi(|\vec{c_i}| + |\vec{\alpha_j}|)) \end{aligned} \quad (28)$$

comme α_{jl} doit être strictement positive, on a effectué le changement de variable suivant $\alpha_{jl} = e^{\beta_{jl}}$, avec β_{jl} est un nombre réel. Alors, la dérivé partielle de Φ (équation 16) par rapport à β_{jl} est comme suit [12]:

$$\begin{aligned} &\frac{\partial}{\partial\beta_{jl}}\Phi(\vec{c},\Theta,\Lambda) \\ = \quad &\alpha_{jl}[(\Psi(|\vec{\alpha_j}|) - \Psi(\alpha_{jl}))\sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j}) \\ + \quad &\sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j})(\Psi(c_{il} + \alpha_{jl}) - \Psi(|\vec{c_i}| + |\vec{\alpha_j}|))] \end{aligned} \quad (29)$$

En calculant la deuxième dérivé du logarithme de la fonction de vraisemblance on obtient [12]:

$$\begin{aligned} &\frac{\partial^2}{\partial\beta_{jl}^2}\Phi(\vec{c},\Theta,\Lambda) \\ = \quad &\frac{\partial}{\partial\beta_{jl}}\Phi(\vec{c},\Theta,\Lambda) \\ + \quad &\alpha_{jl}^2(\Psi^{'}(|\vec{\alpha_j}|) - \Psi^{'}(\alpha_{jl}))\sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j}) \\ + \quad &\alpha_{jl}[\Psi(|\vec{\alpha_j}|) - \Psi(\alpha_{jl}))\sum_{i=1}^{N}\frac{\partial}{\partial\beta_{jl}}p(j/\vec{c_i},\vec{\alpha_j}) \\ + \quad &\sum_{i=1}^{N}[\frac{\partial}{\partial\beta_{jl}}p(j/\vec{c_i},\vec{\alpha_j})(\Psi(c_{il} + \alpha_{jl}) - \Psi(|\vec{c_i}| + |\vec{\alpha_j}|)) \\ + \quad &p(j/\vec{c_i},\vec{\alpha_j})(\Psi^{'}(c_{il} + \alpha_{jl}) - \Psi^{'}(|\vec{c_i}| + |\vec{\alpha_j}|))] \end{aligned} \quad (30)$$

$$\begin{aligned} &\frac{\partial^2}{\partial\beta_{jl_1}\partial\beta_{jl_2}}\Phi(\vec{c},\Theta,\Lambda) \\ = \quad &\alpha_{jl_1}\alpha_{jl_2}\Psi^{'}(|\vec{\alpha_j}|)\sum_{i=1}^{N} p(j/\vec{c_i},\vec{\alpha_j}) \\ + \quad &\alpha_{jl_1}[\Psi(|\vec{\alpha_j}|) - \Psi(\alpha_{jl_1}))\sum_{i=1}^{N}\frac{\partial}{\partial\beta_{jl_2}}p(j/\vec{c_i},\vec{\alpha_j}) \\ + \quad &\sum_{i=1}^{N}[\frac{\partial}{\partial\beta_{jl_2}}p(j/\vec{c_i},\vec{\alpha_j})(\Psi(c_{il_1} + \alpha_{jl_1}) - \Psi(|\vec{c_i}| + |\vec{\alpha_j}|)) \\ - \quad &\alpha_{jl_2}p(j/\vec{c_i},\vec{\alpha_j})\Psi^{'}(|\vec{c_i}| + |\vec{\alpha_j}|)]] \end{aligned} \quad (31)$$

avec $\Psi^{'}(.)$ est la fonction trigamma. Notons qu'on a besoins de calculer la dérivé de la probabilité *a posteriori*

$p(j/\vec{c_i},\vec{\alpha_j})$ par rapport à β_{jl} [12]:

$$\begin{aligned}\frac{\partial}{\partial\beta_{jl}}p(j/\vec{c_i},\vec{\alpha_j}) &= \alpha_{jl}\times p(j/\vec{c_i},\vec{\alpha_j})(1-p(j/\vec{c_i},\vec{\alpha_j}))\\ &\times\ (\Psi(|\vec{\alpha_j}|)-\Psi(|\vec{c_i}|+|\vec{\alpha_j}|)\\ &+\ \Psi(c_{il}+\alpha_{jl})-\Psi(\alpha_{jl}))\end{aligned} \tag{32}$$

Prenons un ensemble de paramètres initiaux, la méthode de Fisher peut être utilisée maintenant. Le schéma itératif de cette méthode est donnée par l'équation suivante:

$$\begin{aligned}\begin{pmatrix}\hat{\beta}_{j1}\\ \vdots\\ \hat{\beta}_{jdim}\end{pmatrix}^{new} &= \begin{pmatrix}\hat{\beta}_{j1}\\ \vdots\\ \hat{\beta}_{jdim}\end{pmatrix}^{old}\\ &+ \begin{pmatrix} Var(\hat{\beta}_{j1}) & \dots & Cov(\hat{\beta}_{j1},\hat{\beta}_{jdim})\\ \vdots & \ddots & \vdots\\ Cov(\hat{\beta}_{jdim},\hat{\beta}_{j1}) & \dots & Var(\hat{\beta}_{jdim})\end{pmatrix}^{old}_{old}\\ &\times \begin{pmatrix}\frac{\partial}{\partial\hat{\beta}_{j1}}\Phi\\ \vdots\\ \frac{\partial}{\partial\hat{\beta}_{jdim}}\Phi\end{pmatrix}\end{aligned} \tag{33}$$

avec j est le numéro de la classe: $1\leq j\leq M$.
La matrice de covariance est l'inverse de la matrice d'information de Fisher $\mathbf{I}$. Cette matrice $\mathbf{I}$ est:

$$\mathbf{I}=I_{l_1l_2}=-E[\frac{\partial^2}{\partial\beta_{jl_1}\partial\beta_{jl_2}}\Phi(\vec{c},\Theta,\Lambda)] \tag{34}$$

Comparons cette méthode itérative à une autre méthode quasi-Newton présentée par l'équation suivante: [6]:

$$\beta_{jl}^{new}:=\beta_{jl}^{old}-\eta\frac{\partial}{\partial\beta_{jl}}\Phi(\vec{c},\Theta,\Lambda) \tag{35}$$

avec $0<\eta\leq 1$, notons la présence d'un terme en plus, qui est l'inverse de la matrice de l'information de Fisher. Regardons alors l'interprétation géométrique de l'équation 33:

$$\begin{aligned}\beta_{jl}^{new}:=\beta_{jl}^{old}-&\\ \underbrace{\begin{pmatrix} Var(\hat{\beta}_{jl}) & \dots & Cov(\hat{\beta}_{jl},\hat{\beta}_{jdim})\end{pmatrix}^{old}_{old}\begin{pmatrix}\frac{\partial\Phi}{\partial\hat{\beta}_j}\\ \vdots\\ \frac{\partial\Phi}{\partial\hat{\beta}_{jdim}}\end{pmatrix}^{old}}_{NG}&\end{aligned} \tag{36}$$

Le gradient ordinaire $\frac{\partial}{\partial\beta_{jl}}\Phi(\vec{c},\Theta,\Lambda)$ a été remplacé par le terme NG, appelé *Gradient Naturel* par Amari [20].
Soit $S=\{\vec{\beta}=(\beta_1,\dots,\beta_{dim})\in\mathbb{R}^{dim}\}$ l'espace de paramètres dans lequel la fonction de vraissemblance (Φ) est définie. Quand S est Euclidien avec un systèmes de coordonnées orthonormal, alors le carré de la distance $\vec{\partial\beta}=(\partial\beta_1,\dots,\partial\beta_{dim})$ entre un vecteur $\vec{\beta}$ et $\vec{\beta}+\vec{\partial\beta}$ est le suivant:

$$|\vec{\partial\beta}|^2=\sum_{l_1=1}^{dim}(\partial\beta_i)^2 \tag{37}$$

Mais, si le système est nonorthonormal, alors la distance quadratique est donné comme suit [20]:

$$|\vec{\partial\beta}|^2=\sum_{l_1=1}^{dim}\sum_{l_2=1}^{dim}g_{l_1l_2}(\partial\beta_{l_1})(\partial\beta_{l_2}) \tag{38}$$

La $(dim)\times(dim)$ matrice $G=(g_{l_1l_2})$ est appelé tenseur Riemmanien. Cette matrice est réduite à la matrice unité $I_{dim\times dim}$ dans le cas Euclidien orthonormal. Sachant que la Dirichlet est une denisté exponentielle, on peut affirmer que l'espace des paramètres de notre fonction de vraisemblance Φ, décrite par l'équation 16, est Riemanien. En effet, selon Amari la famille des probabilités exponentielles est définie dans un espace de Rieman dont la métrique est la matrice de l'information de Fisher [20]. Donc, on n'a pas un système de coordonnées linéaires, et la longueur de $\vec{\partial\beta}$ est donnée par l'équation 38. Sachant que la direction la plus raide de la fonction Φ au niveau de $\vec{\beta}$ est définie par $\vec{\partial\beta}$ qui minimise $\Phi(\beta+\vec{\partial\beta})$, avec $|\vec{\partial\beta}|$ a une longueur fixe et suffisamment petite, on peut déduire que la dérivé de Φ ne peut pas être la même dans un espace Euclidien et un autre Riemanien. La relation entre le gradient naturel $\frac{\check{\partial}\phi}{\partial\beta_l}$ et le gradient ordinaire est donnée par l'équation suivante [21]:

$$\frac{\check{\partial}\phi}{\partial\beta_l}=G^{-1}\frac{\partial\phi}{\partial\beta_l} \tag{39}$$

avec G est la matrice de l'information de Fisher. Ce résultat a été confirmé par quelques expérimantations. En effet, on a implémenté les deux méthodes et on a trouvé que l'approche de l'équation 35 ne donne pas des résulats satisfaisants en la comparant à la méthode de Fisher.

4 Algorithme

Afin de rendre l'algorithme de l'estimation moins sensible au maximum locaux, on a utilisé une initialisation basée sur le Fuzzy C-means [16] et la méthode des moments (MM). En examinant les équations 7 et 8, on peut remarquer que les dim premiers moments d'ordre 1 et les dim moments d'ordre 2 aboutissent à un total de $C_{dim}^{2\times dim}$ combinaisons possibles d'quations pour résoudre les dim paramètres. Selon Fieltiz et Myers [17] une façon symétrique de procéder sera de choisir les $dim-1$ équations d'ordre 1 et le premier équation d'ordre 2. La raison pour laquelle on ne choisit pas l'équation d'ordre 1, numéro (dim), est que cette dernière est une combinaison linéaire des autres. Alors on a:

$$\alpha_l=\frac{(c_{11}^{'}-c_{21}^{'})c_{1l}^{'}}{c_{21}^{'}-(c_{11}^{'})^2}\quad l=1,2\dots,dim-1 \tag{40}$$

et

$$\alpha_{dim}=\frac{(c_{11}^{'}-c_{21}^{'})(1-\sum_{l=1}^{dim}c_{1l}^{'})}{c_{21}^{'}-(c_{11}^{'})^2} \tag{41}$$

avec

$$c_{1l}^{'}=\frac{1}{N}\sum_{i=1}^{N}\frac{c_{il}}{|\vec{c_i}|}\quad l=1,2\dots,dim \tag{42}$$

$$c_{21}^{'} = \frac{1}{N}\sum_{i=1}^{N}\frac{c_{i1}^2}{|\vec{c_i}|^2} \quad (43)$$

Donc notre méthode d'initialisation peut être résumer comme suit:

Algorithme d'initialisation

1. Appliquer le Fuzzy C-means pour obtenir les éléments de chaque composante.
2. Appliquer le MM pour chaque composante j pour obtenir le vecteurs de paramètres $\vec{\alpha_j}$.
3. Affecter chaque donnée à une composante, en supposant que le modèle actuel est correct.
4. Mettre à jours les $P(j)$ en utilisant cette équation:

$$P(j) = \frac{\text{Nombre total d'élements dans la class j}}{N} \quad (44)$$

5. Si le modèle actuel et le nouveau modèle sont suffisamment proches, terminer, sinon aller à 2.

On peut noter que cet algorithme d'initialisation prend la distribution en compte. Contrairement à d'autres méthodes d'initialisation *classique* qui utilisent juste des algorithmes comme le K-Mean , on a introduit la méthode des moments avec un shéma itératif pour raffiner les résultats. En utilisant cette méthode d'initialisation, on suppose dès le début qu'on a un mélange de MDD. Cette méthode d'initialisation est désignée pour des grandes bases de données. Lorsque on travail avec des petites bases de données, on peut appliquer le Fuzzy C-means et le MM une seule fois. Ainsi notre algorithme pour estimer les paramètres d'un mélange de MDD est le suivant:

Algorithme d'estimation des paramètres d'un mélange de MDD

1. Entrée: des données de dimension $dim, c_i, i = 1,\ldots,N$ et un grand nombre de composantes M.
2. Algorithme d'initialisation.
3. Mettre à jours les $\vec{\alpha_j}$ en utilisant l'équation 33, $j = 1,\ldots,M$.
4. Mettre à jours les $P(j)$ en utilisant l'équation 21, $j = 1,\ldots,M$.
5. Si $p(j) < \epsilon$ enlever la composante j, sinon aller à 3.
6. Si on a la convergence, terminer, sinon aller à 3.

Si l'échantillon de données est suffisamment grand, alors le test de convergence peut être fait en utilisant une méthode statistique utilisée dans [18]. Cette méthode utilise la forme quadratique du vecteur de gradient. Prenons les statistiques données par l'équation 45. On peut montrer que ce test statistique suit une distribution khi-deux avec dim degrés de liberté. Les itérations continuent tant que S est plus petit que $\chi^2_{dim}(\nu)$ pour un ν fixe. D'autres tests de convergence peuvent être utilisés comme la stabilisation des vecteurs $\vec{\beta_j}$ ou bien de la valeur de la fonction de vraisemblance.

$$S = \begin{pmatrix} \frac{\partial}{\partial\hat{\beta}_{j1}}\Phi & \cdots & \frac{\partial}{\partial\hat{\beta}_{j\,dim}}\Phi \end{pmatrix} \times \begin{pmatrix} Var(\hat{\beta}_{j1}) & \cdots & Cov(\hat{\beta}_{j1},\hat{\beta}_{jdim}) \\ \vdots & \ddots & \vdots \\ Cov(\hat{\beta}_{jdim},\hat{\beta}_{j1}) & \cdots & Var(\hat{\beta}_{jdim}) \end{pmatrix} \times \begin{pmatrix} \frac{\partial}{\partial\hat{\beta}_{j1}}\Phi \\ \vdots \\ \frac{\partial}{\partial\hat{\beta}_{jdim}}\Phi \end{pmatrix} \quad (45)$$

5 Application: Modélisation de Texture

L'analyse et la modélisation de Texture est un domaine très important dans l'analyse d'images et même fondamantal dans beaucoup d'applications comme l'automation industrielle, la télédétection et l'analyse des images médicales. La texture est basée essentiellement sur des proprietés de voisinage. Un récent travail de Tuceryan et Jain donne une synthèse des différentes approches structurelles et statistiques pour modéliser les textures [1].

Une des méthodes les plus utilisées pour modéliser les textures est la méthodes spatiale de niveaux de grix qui a été proposée par Haralick [2]. Cette méthode est basée sur les fonctions de probabilités jointes de deux pixels dans une position relative (matrice de cooccurrence). Les matrices de cooccurrence sont souvent utilisées comme étant des caractéristiques intermédiaires et la réduction de dimensionalité est performée en calculant des caractéristiques comme celles décrites dans [3] [4]. Le principal inconvénient dans l'utilisation des matrices de cooccurrence est le grand espace mémoire demandé pour les sauvegarder et la difficulté de les utiliser dans des approches statistiques à cause de leurs nature numérique.

Dans ce qui suit, on va utiliser $\{I(x,y),\ 0 \le x \le K-1,\ 0 \le y \le L-1\}$, pour dénoter une image $K \times L$ avec G niveaux de gris. La $G \times G$ matrice de cooccurrence $C(d_1,d_2)$ pour un vecteur de déplacement $\vec{d} = (d_1,d_2)$ est défnie comme suit. L'entrée (i,j) de $C(d_1,d_2)$ est le nombre d'occurrence du pair des niveaux de gris i et j qui sont à une distance $\vec{d}$. Formellement, on peut définir ca comme suit:

$$c(i,j;\vec{d}) = Card\{(r,s) : I(r,s) = i, I(r+d_1, s+d_2) = j\} \quad (46)$$

avec $Card\{\}$ est le nombre d'éléments de l'ensemble. Il est possible de caractériser une texture particulière par une collection de matrices de cooccurrence estimées pour des déplacements différents d_1 et d_2. En effet, supposons qu'on a dim vecteurs de déplacements $\vec{d}$, alors pour chaque entrée (i,j) on va avoir un vecteur de dim éléments:

$$\vec{c}(i,j;\vec{d_1},\ldots,\vec{d}_{dim}) = (c_1(i,j;\vec{d_1}),\ldots,c_{dim}(i,j;\vec{d}_{dim})) \quad (47)$$

Comme $\vec{c}$ est un vecteur de fréquences, les dim matrices de cooccurrence peuvent être modélisées par un mélange de MDD. Ce modèle est utilisé pour deux applications: le résumé et la recherche des images de textures.

5.1 Résumé d'Images de Texture

Le résumé automatique des images de texture est un important problème dans les applications d'analyse d'images. Pour le problème de résumé des images de texture, on cherche à affecter chaque image à la bonne classe en se basant sur des mesures faites sur l'image entière. Cette application est très importante dans le cas de la recherche d'images par le contenu. le résumé des bases de données simplifie la tâche de rechercher des images en limitant la recherche d'images similaires à un domaine plus petit de la base de données. le résumé est très efficace aussi pour la navigation. La connaissance des catégories des images dans une base de données, permet aux utilisateurs de trouver les images qu'ils cherchent plus rapidement [15]. Un diagramme présentant notre approche pour résoudre le problème de résumé des textures est presenté par la figure 1. Les entrées du classificateur sont des images provenant d'un nombre finis de classes de textures. Ces images sont séparées en des images de test qu'on ne connait pas leurs classes de texture et d'autres pour l'apprentissage (leurs classes sont connus). La phase d'apprentissage est nécéssaire pour adapter le classificateur à chaque possible classe de texture. Toutes les images d'entrées passent par la phase de calcul de la matrice de cooccurrence et après la phase d'estimation des paramètres du mélange dans laquelle les matrices de cooccurrence sont modélisées par un mélange de MDD. Après cette étape, chaque classe de texture est representée par un mélange de MDD. Finalement, on a l'étape de classification qui utilise les mélanges estimés pour les images dont les classes sont inconnus afin de déterminer leurs classes. En effet, le mélange estimé est comparé à celui qu'on a appris en utilisant la distance pour les densités $\mathcal{L}_2$ définis comme suit [19]:

$$D(e,t) = \int (p(\vec{c}|e) - p(\vec{c}|t))^2 dc \qquad (48)$$

avec e et t représentent respectivement un mélange de MDD d'une image de texture à classifier ($p(\vec{c}|e)$) et un mélange de MDD obtenu par l'apprentissage et qui représente une classe de texture $p(\vec{c}|t)$). Alors la classification va être performée en utilisant cette règle: une image e est affectée à une classe t_1 si $D(e,t1) < D(e,t) \forall t \neq t1$.

Pour les résultats présentés dans ce papier, on a utilisé la base de données de texture *Vistex* obtenu du MIT Media Lab. Dans notre expérience, chacune des 512 × 512 images de la base de donnée *Vistex* était divisée en images de taille 64 × 64. Comme chaque 512 × 512 "image mère" a contribuée à 64 images, idéallement toutes les 64 images doivent être classées dans la même classe. Dans cette application, six groupes homogènes de textures, "bark", "fabric", "food", "metal", "water" et "sand" on été utilisés pour créer une nouvelle base de données. Une base de données

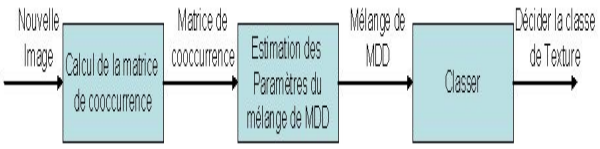

FIG. 1 – *diagramme pour le problème de résumé d'images de texture.*

de 1920 images de taille 64 × 64 pixelles est obtenue. Quatre images des classes de texture bark, fabric et metal ont été utilisées pour obtenir 256 images pour chacun de ces groupes, et six images des classes water, food et sand ont été utilisés pour obtenir 384 images pour ces catégories. Des exemples de ces images pour chacune des catégories sont donnés par la figure 2. Après le calcul des matrices de

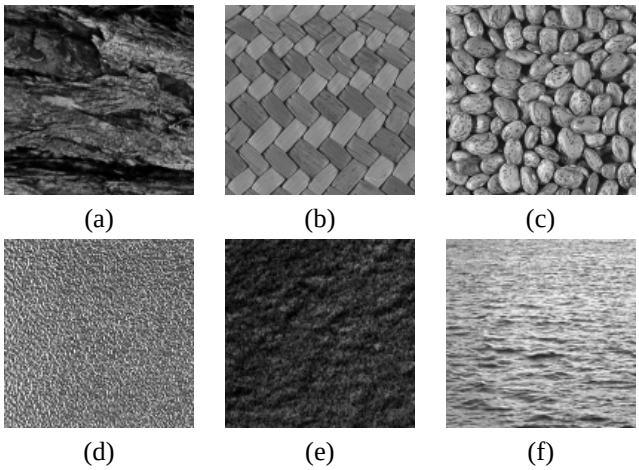

FIG. 2 – *Exemples d'images de chaque groupe (a) Bark, (b) Fabric, (c) Food, (d) Metal, (e) Sand, (f) Water.*

cooccurrence, chaque catégorie d'images sera modélisée par un mélange de MDD. Le tableau 1 donne les paramètres de ces mélanges lorsqu'on considère 8 déplacements. La matrice de confusion de la classification des images de texture est donnée par le tableau 2. Dans cette matrice de confusion, la cellule (i,j) représente le nombre d'images de la catégorie i qui sont classées comme catégorie j. Le nombre d'images mal classées était égal à 38 images, ce qui représente une éxactitude de 98.02 . Pour la comparaison, on a utilisé le mélange de Gaussiennes à la place des MDD. le tableau 3 montre la matrice de confusion lorsque le mélange de Gaussiennes est utilisé (81 images mal classées, càd une exactitude de 95.79) Grâçe à ces tableaux, on peut déduire que la performance des MDD est meilleure.

Après que la base des données est résumée, on a fait une autre expérience qui consiste à rechercher des images similaire à une requête donnée.

Catégorie	paramèters des mélanges de MDD									
	classe	P(j)	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
Bark	classe1	0.153	006.959	006.789	006.943	006.770	006.924	006.909	006.981	006.988
	classe2	0.241	011.032	012.354	011.799	011.743	011.622	012.353	011.198	011.982
	classe3	0.133	004.283	006.602	6.709	006.871	005.152	006.645	007.542	007.331
	classe4	0.201	024.685	015.902	016.022	015.908	021.489	015.920	013.616	013.620
	classe5	0.141	005.853	006.475	006.323	006.474	005.998	006.342	006.404	006.455
	classe6	0.131	005.658	005.941	006.023	005.964	005.795	006.035	006.244	006.206
Fabric	classe1	0.258	025.386	022.075	023.681	022.517	026.823	025.745	025.192	024.311
	classe2	0.261	112.721	120.025	119.351	109.494	105.727	105.704	113.218	117.024
	classe3	0.266	556.039	595.921	573.386	583.624	512.379	518.183	522.132	532.666
	classe4	0.215	011.525	011.477	011.380	011.455	012.298	12.331	12.229	11.879
Food	classe1	0.205	022.871	022.037	020.157	022.649	023.620	023.145	022.324	021.280
	classe2	0.165	004.914	007.579	006.025	006.969	008.663	010.282	012.798	011.390
	classe3	0.144	040.978	037.249	042.325	037.690	032.248	030.280	028.814	032.241
	classe4	0.164	004.684	007.368	005.662	006.805	008.725	010.188	012.726	011.234
	classe5	0.146	096.283	084.071	107.244	085.898	066.255	064.771	057.927	071.078
	classe6	0.176	012.889	013.675	012.205	013.282	015.050	015.165	015.388	014.079
Metal	classe1	0.233	019.143	016.351	020.814	018.257	014.195	013.714	014.422	013.947
	classe2	0.121	006.077	005.706	006.141	006.019	005.650	005.508	005.599	005.557
	classe3	0.239	014.560	015.379	013.888	015.914	016.619	016.503	016.439	017.173
	classe4	0.242	017.159	016.779	016.589	016.944	017.009	016.151	016.517	016.376
	classe5	0.165	005.848	007.433	005.629	005.579	007.619	008.433	007.985	007.971
Sand	classe1	0.287	007.062	008.344	007.069	007.607	008.798	008.944	009.106	008.709
	classe2	0.337	013.811	014.133	013.802	013.945	014.409	014.645	014.781	14.419
	classe3	0.376	128.296	114.760	129.365	122.650	110.878	106.602	104.713	109.643
Water	classe1	0.504	846.694	832.655	853.937	857.329	797.566	804.157	805.096	822.290
	classe2	0.496	135.545	135.492	135.237	135.450	136.719	136.355	136.490	135.512

TAB. 1 – *Estimation des paramètres des mélanges MDD pour différentes catégories lorsqu'on considère 8 déplacements.*

	Bark	Fabric	Food	Metal	Sand	Water
Bark	252	0	0	0	4	0
Fabric	0	250	6	0	0	0
Food	0	5	379	0	0	0
Metal	0	0	0	256	0	0
Sand	2	0	0	0	382	0
Water	1	0	0	5	2	376

TAB. 2 – *Matrice de confusion pour un classificateur basé sur le mélange de MDD.*

	Bark	Fabric	Food	Metal	Sand	Water
Bark	243	0	0	3	8	2
Fabric	0	240	10	0	4	2
Food	0	12	367	4	0	1
Metal	0	0	0	249	2	5
Sand	7	0	0	0	374	3
Water	4	0	0	8	6	366

TAB. 3 – *Matrice de confusion pour un classificateur basé sur le mélange de Gaussiennes.*

5.2 Recherche par le Contenu d'Images de Texture

Pour chercher des images qui sont similaires à une requête, on suit deux étapes . D'abord la distance $\mathcal{L}_2$ est utilisée pour trouver les deux catégories les plus proches à la requête en comparant cette dernière à chaque image représentative des différentes catégories. La même mesure de distance est utilisée dans l'étape suivante pour déterminer la similarité entre la requête et les élements des deux composantes les plus proches. Les meilleurs n images sont alors prises (n dépend de l'éxpérience). Pour mesurer le taux de réussite dans la recherche, chaque image était utilisée comme requête et le nombre d'images pertinentes était noté. la précision et le rappel qui sont les mesures les plus utilisées par la communité de la recherche de l'information, ont été calculés en utilisant les équations 49 et 50. On a pris la moyenne de ces mesures aprés les avoir calculés sur toutes les requêtes:

$$\text{précision} = \frac{\text{nombre des images retrouvées et pertinentes}}{\text{nombre des images retrouvées}} \tag{49}$$

$$\text{rappel} = \frac{\text{nombre des images retrouvées et pertinentes}}{\text{nombre total des images pertinentes}} \tag{50}$$

Comme chaque 512×512 images de la base de données *Vistex* donne 64 images, alors en prenant une image requête, idéallement toutes les 64 images doivent être retrouvées

	Nombre d'images recherchées				
Méthode	16	48	64	80	96
Mélange de MDD	0.93	0.95	0.92	0.80	0.66
Mélange de Gaussiennes	0.87	0.91	0.85	0.77	0.66

TAB. 4 – *Précision obtenue pour la Base de données de Texture.*

	Nombre d'images recherchées				
Méthode	16	48	64	80	96
Mélange de MDD	0.23	0.71	0.92	0.94	0.96
Mélange de Gaussiennes	0.21	0.68	0.82	0.92	0.93

TAB. 5 – *Rappel obtenu pour la Base de données de Texture.*

et considérées pertinentes. Les tableaux 4 et 5 présentent l'éxactitude de la recherche en terme de précision et de rappel en utilisant les deux méthodes. On donne les résultats lorsque 16, 48, 64, 80, 96 et 128 images sont recherchées à la suite d'une requête.

6 Conclusion

Dans ce papier, on a introduit un nouveau mélange de densités basé sur la distribution Multinomiale et celle de Dirichlet. On a estimé les paramètres de ce mélange en utilisant le maximum de vraisemblance et la méthode de Fisher. Les expérimentation concernent un nouveau modèle de texture et ses applications pour le résumé des bases de données d'images pour une recherche efficace. Une évaluation claire et compréhensive de ce modèle est donnée en utilisant un grand nombre d'images de texture et en la comparant à un autre modèle. Le mélange de MDD est actuellement utilisé dans beaucoup d'autres applications qui incluent des données discrétes où des matrices de cooccurrence comme le Textmining, le Webmining et la segmentation du courrier éléctronique.

Remerciement

L'accomplissement de ce travail est rendu possible grâçe au support de Bell Canada à travers leur programme universitaire R&D.

Références

[1] M. Tuceryan and A. K. Jain, Texture Analysis, *The Handbook of Pattern Recognition and Computer Vision*, pp. 207-248, 1998.

[2] R. M. Haralick, K. Shanmugan, and I. Dinstein Textural features for image classification, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, pp. 610-621, 1973.

[3] M. Unser, Sum and Difference Histograms for Texture Classification, *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 8, No. 1, pp. 118-125, 1986.

[4] A. L. Vickers and J.W. Modestino, A Maximum Likelihood Approach to Texture Classification, *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 4, No. 1, pp. 61-68, 1982.

[5] N. Bouguila, D. Ziou and J. Vaillancourt, Novel Mixtures Based on the Dirichlet Distribution: Application to Data and Image Classification , *IAPR International Conference on Machine Learning and Data Mining, lNCS2734*, pp. 172-181, 2003.

[6] R. A. Redner and H. F. Walker, Mixture Densities, Maximum Likelihood and EM Algorithm , *SIAM Review*, vol. 26, No. 2, pp. 195-239, 1984.

[7] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm , *Journal of the Royal Statistical Society, B*, vol. 39, pp. 1-38, 1977.

[8] G. Schwarz, Estimating the Dimension of a Model , *The Annals of Statistics*, vol. 6, No. 2, pp. 461-464, 1978.

[9] H. Akaike, A New Look at the Statistical Model Identification , *IEEE Transactions on Automatic Control*, vol. AC-19, No. 6, pp. 716-723, 1974.

[10] J. Rissanen, Modeling by Shortest Data Description , *Biometrika*, vol. 14, pp. 465-471, 1978.

[11] S. Medasani and R. Krishnapuram, Categorization of Image Databases for Efficient Retrieval Using Robust Mixture Decomposition, *Computer Vision and Image Understanding*, vol. 83, pp. 216-235, 2001.

[12] N. Bouguila, D. Ziou and J. Vaillancourt, Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application, *Submitted to IEEE Transactions on Image Processing*.

[13] C. R. Rao, *Advanced Statistical Methods in Biomedical Research*, New York: John Wiley and Sons, 1952

[14] G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley, 2000

[15] N. Bouguila, D. Ziou and J. Vaillancourt, Probabilistic Multimedia Summarizing Based on the Generalized Dirichlet Mixture, *Supplementary Proceedings of ICANN/ICONIP 2003*, pp 150-154, 2003.

[16] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[17] B. D Fielitz and B. L Myers, Estimation of Parameters in the Beta Distribution, *Decision Sciences*, vol. 6, pp. 1-13, 1975.

[18] S. C. Choi and R. Wette, Maximum Likelihood Estimation of Parameters of the Gamma Distribution and their Bias, *Technometrics*, vol. 11, No. 4, pp. 683-690, 1969.

[19] D. Xu, j. C. Principe, J. Fisher and H. C. Wu, A Novel measure for Independant Component Analysis (ICA), *IEEE ICASSP98*, vol. 2, pp. 1161-1164, 1998.

[20] S. Amari, O. E. Barndorff-Nielsen, R. E. Kass, S. L. Lauritzen and C. R. Rao, *Differential Geometry in Statistical inference*, vol. 10, Institute of Mathematical Statistics, Hayward, California, 1987.

[21] S. Amari, Natural Gradient Works Efficiently in Learning *Neural Computation*, vol. 10, pp. 251-276, 1998.

[22] A. E. Raftery and J. D. Banfield, Model-Based Gaussian and Non-Gaussian Clustering *Biometrics*, vol. 49, pp. 803-821, 1993.