IDIAP RESEARCH REPORT

DYNAMICAL DIRICHLET MIXTURE MODEL

Le Chen [a] David Barber [a]
Jean-Marc Odobez [a]

IDIAP–RR 2007-02

APRIL 2007

[a] IDIAP Research Institute

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Rapport de recherche de l'IDIAP 2007-02

DYNAMICAL DIRICHLET MIXTURE MODEL

Le Chen David Barber Jean-Marc Odobez

APRIL 2007

Résumé. In this report, we propose a statistical model to deal with the discrete-distribution data varying over time. The proposed model – HMM+DM – extends the Dirichlet mixture model to the dynamic case: Hidden Markov Model with Dirichlet mixture output. Both the inference and parameter estimation procedures are proposed. Experiments on the generated data verify the proposed algorithms. Finally, we discuss the potential applications of the current model.

Table des matières

1 Introduction

The discrete-distribution data, or proportions or shares allotted to different categories, come from many fields. Examples include the bag-of-word representation of documents in the information retrieval field [BYRN99], and also the same representation for images in the computer vision field [SZ03, QMO+05]. In most cases, we do not care about the total number of words in each document (or image). Recall that the cosine similarity measure between two documents normalizes the lengths of both documents, and also the probabilistic latent aspect model (pLSA) [Hof99] treats each document as a discrete distribution over all words in the vocabulary. That is, we only care the relative proportions of each word in both cases. So it is reasonable to normalize these data by their word accounts. Then we get the discrete-distribution data in these two examples. Still another example comes from the speech processing field [HES00, BW90], where multilayer perceptron (MLP) is trained to get the posteriors (discrete-distribution data) as the features for further process.

It is natural to treat the underlying stochastic process to generate these discrete-distribution data as the Dirichlet process [Ron89, Aic82]. Similar to Gaussian case (Gaussian mixture model, GMM [DHS01]), Dirichlet mixtures (DM) are, furthermore, viewed powerful enough to represent the distributions on the compact probabilistic simplex with multiple symmetric or asymmetric modes [BZV04].

The discrete-distribution data which vary over time are also very common. The last example above in speech processing is indeed one such example. In video analysis, if we extract the bag-of-word features from one or several consecutive frames, after normalization, we get the discrete-distribution data varying over time. In social sciences, the budget shares of households or market shares of firms also very over time.

There may be many ways to impose the temporal constraints on the discrete-distribution data. In [HES00], Hynek Hermansky and et al. proposed to embedding the discrete-distributional output of multilayer perceptron (MLP) into hidden Markov model (HMM) by taking the log and Karhunen Lòeve transform (KLT) on the discrete-distribution data. And then they use the GMM to model the data densities as the HMM's emission probabilities.

In this paper, we propose another way to model the discrete-distribution data varying over time : hidden Markov model with Dirichlet mixture emission (HMM+DM). Instead of transform the discrete-distribution data to another domain like [HES00], we model the data in a more direct way using Dirichlet mixtures. We further give the parameter estimation algorithm, with the inference procedure as a subroutine, for HMM+DM model. As a by-product or a special case, we propose the expectation maximization (EM [DLR77]) algorithm to estimate parameters of the Dirichlet mixtures. To alleviate the mess of notations in the derivations, we propose a summation rule similar to Einstein summation in Section 2.3.

From another view point, any mixture density model has its dynamic or HMM counterpart. For example, GMM's dynamic counterpart is HMM+GMM ; Factor analysis corresponds to Kalman filter. So it is natural to investigate the dynamic counterpart of the Dirichlet mixtures : that is, HMM+DM.

The outline of this report is as follows. In Section 2, we specify the proposed HMM+DM model, with the detailed derivations for inference procedure gathered in Appendix A and those for parameter estimation in Appendix B. Some experiments on the simulated data are presented in Section 3 to verify the proposed algorithms. Then in Section 4, we make a brief review of the related work. Some discussion and future work are given in Section 5. We express our acknowledgement in Section 6. Finally we list the main matlab codes in Appendix C.

2 Dynamic extension of Dirichlet mixtures – HMM+DM model

After introducing the model and notations in Section 2.1, we present the inference procedure and parameter estimation procedure in Section 2.2 and Section 2.3 respectively. Treating the static Dirichlet mixtures as a special case of HMM+DM, we give the EM procedure for parameter estimation of the Dirichlet mixtures in Section 2.4.

2.1 Model specification

In Fig. 1, we show the HMM+DM model. The shadowed circles represents observables, which are discrete distributions here. The unshadowed circles are hidden states. There are two sets of hidden states : one is the Markov hidden state h_t (or simply call it as hidden states) ; the other is the mixture indicator m_t.

Suppose that there are K hidden states. The transition matrix between hidden states is $B = [b_{ij}] \in \mathbb{R}_+^{K\times K}$ [1], with

$$b_{ij} = p(h_{t+1} = j | h_t = i) \quad .$$

[1] $x \in \mathbb{R}_+$ means $x \geq 0$.

The initial probabilities are $\pi = \{\pi_i \in [0,1] \mid \sum_i \pi_i = 1, i = 1, ..., K\}$. For each hidden state $h_t = k$, the emission probability is a mixture of Dirichlet :

$$A_k = \begin{pmatrix} a_{k,11} & \cdots & a_{k,1N} \\ \vdots & \ddots & \vdots \\ a_{k,M1} & \cdots & a_{k,MN} \end{pmatrix}, \quad \in \mathbb{R}_+^{M\times N} \quad ,$$

where there are M mixture components. We assume, without generality, the same number of mixture components for different hidden state $h_t = k$. The dependence of between the Markov hidden states and the mixture indicators is characterized by :

$$C = [C_{ij}] \in \mathbb{R}_+^{K\times M}, \quad C_{ij} = p(m = j \mid h = i) \quad .$$

Finally the set of parameters are $\theta = \{A, B, \pi, C\}$.

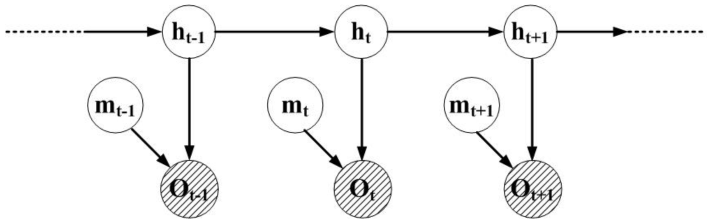

FIG. 1 – Graphical representation of HMM+DM

Note : in fact, Fig.1 is a general graphical representation of the Hidden Markov model with mixture densities as emission probability. If we change the emission as Gaussian density, we get the HMM+GMM model. In general, we could derive the case that the emission density is the mixture of exponential families. Then HMM+GMM and current HMM+DM are two special case. This model will leave for future investigation.

2.1.1 Notations :

In the following, D is the number of sequences in the data set. K is the number of the hidden states. And M is the number of mixture components.

The data set consists of D sequences

$$\mathcal{X} = \{X^d \mid d = 1, ..., D\}$$

with each sequence

$$X^d = \left\{ x_t^d \;\mid\; x_t^d \in \mathbb{R}_+^N, \sum_{n=1}^{N} x_{tn}^d = 1, t = 0, ..., T_d \right\}$$

having length of $T_d + 1$. Corresponding hidden states are

$$\mathcal{H} = \{H^d \mid d = 1, ..., D\}$$

with each d^{th} sequence's hidden states being

$$H^d = \left\{ h_t^d \in \{1, ..., K\}, m_t^d \in \{1, ..., M\} \mid t = 0, ..., T_d \right\} \quad .$$

Some special functions : $\Gamma(x)$ is gamma function define as

$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} \mathrm{d}t \quad .$$

$\Psi_n(\cdot)$ denotes the polygamma function of order n, which is the $(n+1)^{th}$ derivative of log [2] gamma function :

$$\Psi_n(x) = \frac{\mathrm{d}^{n+1}}{\mathrm{d}x^{n+1}} \log \Gamma(x) \quad .$$

See [AS64] for many useful properties of both gamma and polygamma functions.

Dirichlet distribution The Dirichlet distribution is defined as

$$\mathrm{Dir}\,(x \mid a) = \frac{\Gamma(\sum_{i=1}^n a_i)}{\prod_{i=1}^n \Gamma(a_i)} \prod_{i=1}^n x_i^{a_i - 1}$$

where the random variable $x \in \mathbb{R}_+^n, \sum_{i=1}^n x_i = 1$ and the parameter $a \in \mathbb{R}_+^n$.

2.2 Inference procedure

In this part, we give the general inference procedure of HMM + mixture density model. We only list the recursion formulae. Please refer to Appendix A for the detailed derivations of these recursions listed in this subsection.

Note : in the following, we only consider one sequence. If necessary, the index for the sequence d will appear as the super-scripture just after time index t. For example, for d^{th} sequence, α_{km}^t will be written as $\alpha_{km}^{t,d}$.

2.2.1 Forward and backward recursions

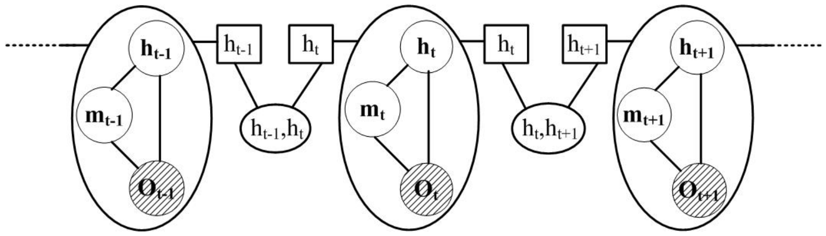

FIG. 2 – Junction tree of HMM+DM

By transforming the original graphical model in Fig. 1 to the junction tree in Fig. 2, we get to know that the following quantities

$$\begin{aligned} p(h_t, m_t \mid x_0, ..., x_T) \\ p(h_t, h_{t+1} \mid x_0, ..., x_T) \\ p(h_t \mid x_0, ..., x_T) \end{aligned}$$

are needed to calculate.

[2]If not stated explicitly, log function is based on e, instead of 2 or 10.

Alpha Recursion Alpha recursion is a forward iteration as :

$$\alpha^t_{h_t m_t} \triangleq p(x_0, .., x_t, h_t, m_t)$$

Then for $t = 1, \cdots T$,

$$\alpha^{t+1}_{h_{t+1} m_{t+1}} = \frac{\sum_{h_t, m_t} \bar{\alpha}^t_{h_t m_t} \cdot b_{h_t h_{t+1}} \cdot C_{h_{t+1} m_{t+1}} \cdot p(x_{t+1} \mid h_{t+1}, m_{t+1})}{\sum_{h'_t, m'_t, h'_{t+1}, m'_{t+1}} \bar{\alpha}^t_{h'_t m'_t} \cdot b_{h'_t h'_{t+1}} \cdot C_{h'_{t+1} m'_{t+1}} \cdot p(x_{t+1} \mid h'_{t+1}, m'_{t+1})}$$

and for $t = 0$

$$\bar{\alpha}^0_{km} = \pi_k \cdot C_{km} \cdot p(x_0 \mid h_0 = k, m_0 = m)$$

For numerical stability, we prefer the following normalized alpha recursion :

$$\begin{aligned} \bar{\alpha}^t_{h_t m_t} &\triangleq p(h_t, m_t \mid x_0, .., x_t) \\ \bar{\alpha}^{t+1}_{h_{t+1} m_{t+1}} &= \frac{\sum_{h_t, m_t} \bar{\alpha}^t_{h_t m_t} \cdot b_{h_t h_{t+1}} \cdot C_{h_{t+1} m_{t+1}} \cdot p(x_{t+1} \mid h_{t+1}, m_{t+1})}{\sum_{h'_t, m'_t, h'_{t+1}, m'_{t+1}} \bar{\alpha}^t_{h'_t m'_t} \cdot C_{h'_{t+1} m'_{t+1}} \cdot b_{h'_t h'_{t+1}} \cdot p(x_{t+1} \mid h'_{t+1}, m'_{t+1})} \end{aligned}$$

Note, $\bar{\alpha}^t_{h_t m_t}$ is the filtered estimate of the state. The initial sates can be calculated as follows :

$$\bar{\alpha}^0_{km} = \frac{\pi_k \cdot C_{km} \cdot p(x_0 \mid h_0 = k, m_0 = m)}{\sum_{k'=1}^{K} \sum_{m'=1}^{M} \pi_{k'} \cdot C_{k'm'} \cdot p(x_0 \mid h_0 = k', m_0 = m')}$$

Beta Recursion Beta recursion is a backward iteration :

$$\begin{aligned} \beta^t_{h_t m_t} &\triangleq p(x_{t+1}, .., x_T \mid h_t, m_t) \\ \beta^{t-1}_{h_{t-1} m_{t-1}} &= \sum_{h_t, m_t} b_{h_{t-1} h_t} \cdot C_{h_t m_t} \cdot p(x_t | h_t, m_t) \cdot \beta(h_t) \end{aligned}$$

Gamma Recursion Gamma values are the smoothed estimates of the states, given the whole set.

$$\begin{aligned} \gamma^t_{h_t m_t} &\triangleq p(h_t, m_t \mid x_0, .., x_T) \\ \gamma^t_{h_t m_t} &= \sum_{h_{t+1}, m_{t+1}} \frac{\bar{\alpha}^t_{h_t m_t} \cdot b_{h_t h_{t+1}}}{\sum_{h'_t, m'_t} \bar{\alpha}^t_{h'_t m'_t} \cdot b_{h'_t h_{t+1}}} \cdot \gamma^{t+1}_{h_{t+1} m_{t+1}} \end{aligned}$$

The initial state can be initialized as

$$\gamma^T_{km} = \bar{\alpha}^T_{km} \quad .$$

The $\eta^t_{h_t}$ variables We need the following variables as the separators in the junction tree :

$$\eta^t_{h_t} = p(h_t \mid x_0, ..., x_T) = \sum_{m=1}^{M} \gamma^t_{h_t m}$$

The $\xi^t_{h_t h_{t+1}}$ variables We need also the following variables :

$$\xi^t_{h_t h_{t+1}} \triangleq p(h_t, h_{t+1} \mid x_0, .., x_T) = \sum_{m_t, m_{t+1}} \frac{\bar{\alpha}^t_{h_t m_t} \cdot b_{h_t h_{t+1}}}{\sum_{h'_t, m'_t} \bar{\alpha}^t_{h'_t m'_t} \cdot b_{h'_t h_{t+1}}} \cdot \gamma^{t+1}_{h_{t+1} m_{t+1}}$$

2.2.2 Hard clustering based on Gamma variables

We can use the following formulae to get the hard clustering results from the posteriors :

$$(h_t^d)^* = \arg\max_k \eta_k^{t,d} \tag{1}$$

$$(m_t^d)^* = \arg\max_m \gamma_{km}^{t,d} \mid_{k=(h_t^d)^*} \quad . \tag{1'}$$

2.3 Parameter estimation

Here, we use the maximal likelihood criterion to estimate the parameters of HMM+DM model. We use the EM algorithm to deal with the hidden states in this model.

In the E step, we have

$$q(h_t^d = k, m_t^d = m) = \gamma_{km}^{t,d}$$
$$q(h_t^d = k, h_{t+1}^d = k') = \xi_{kk'}^{t,d} \quad .$$

As intermediate results, we should also calculate the following filtered state

$$p(h_t^d = k, m_t^d = m \mid x_0^d, \cdots, x_t^d) = \bar{\alpha}_{km}^{t,d} \quad .$$

In the M step, we optimize the energy term or the expected complete log likelihood. To alleviate the mess of notations in the derivations, we propose the following summation rules first :

Summation Convention : For notational convenience, we use the following convention similar to Einstein summation : all the subscripts or superscripts of typewriter style mean the dummy indexes, which should be summed over accordingly ; and those of normal Italians style are the free indexes, which should not be summed over. As you can see, some of t above are summed over $[0 : T]$, while some are summed over $[0 : T-1]$. The convention is that try the maximal legal span of the dummy indexes, otherwise the span should explicitly denoted behind.

Using this summation convention, the energy term can be written as :

$$\begin{aligned} E(A, B, U, \pi, C \mid \mathcal{X}, \mathcal{H}) = &\langle \log p(\mathcal{X}, \mathcal{H}) \rangle_{q(\mathcal{H})} \\ =&\gamma_{\mathtt{km}}^{0,\mathtt{d}} \log \pi_{\mathtt{k}} + \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log C_{\mathtt{km}} + \xi_{\mathtt{ij}}^{\mathtt{t,d}} \log b_{\mathtt{ij}} \\ &+ \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log \Gamma(\sum_{n=1}^{N} a_{\mathtt{k,m}n}) - \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log \Gamma(a_{\mathtt{k,mn}}) + \gamma_{\mathtt{km}}^{\mathtt{t,d}} (a_{\mathtt{k,mn}} - 1) \log x_{\mathtt{tn}}^{\mathtt{d}} \end{aligned} \tag{2}$$

See Appendix B for the detailed derivation.

Then it easy to see that the updates for B, π, γ and C are

$$\pi_k^{new} \propto \sum_{d=1}^{D} \sum_{m=1}^{M} \gamma_{km}^{0,d} \quad \text{or} \propto \gamma_{\mathtt{km}}^{0,\mathtt{d}} \tag{3}$$

$$b_{ij}^{new} \propto \sum_{d=1}^{D} \sum_{t=0}^{T_d-1} \xi_{ij}^{t,d} \quad \text{or} \propto \xi_{ij}^{\mathtt{t,d}} \tag{4}$$

$$C_{km}^{new} \propto \sum_{d=1}^{D} \sum_{t=0}^{T_d} \gamma_{km}^{t,d} \quad \text{or} \propto \gamma_{km}^{\mathtt{t,d}} \tag{5}$$

where $k, i, j = \{1, ..., K\}$, and $m = \{1, ..., M\}$.

2.3.1 Estimation of Dirichlet components

In the following, we should only maximize the following quantity as part of the energy term in Eq.2 :

$$
\begin{aligned}
L(A) = &\gamma_{\mathtt{km}}^{\mathtt{t,d}} \log\Gamma(\sum_{n=1}^{N} a_{\mathtt{k,mn}}) - \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log\Gamma(a_{\mathtt{k,mn}}) + \gamma_{\mathtt{km}}^{\mathtt{t,d}} a_{\mathtt{k,mn}} \log x_{\mathtt{tn}}^{\mathtt{d}} \\
s.t. \quad & a_{k,mn} > 0
\end{aligned}
\tag{6}
$$

In order to alleviate the notations, we can restate this sub-problem of estimating the $(k,m)^{th}$ Dirichlet distribution in the following simple way : we discard the default index of k and m. So the parameters are alleviated as $\{a_1, ..., a_N\}$ from $\{a_{k,m1}, ..., a_{k,mN}\}$. The scale parameter is simplified as γ from $\gamma_{km}^{\mathtt{t,d}}$. The mean of random variables in canonical form is $\log \bar{x}_n = (\gamma_{km}^{\mathtt{t,d}}/\gamma) \log x_{\mathtt{tn}}^{\mathtt{d}}$. Note, $\bar{x}$ is the geometric mean of dataset weighted by the posterior $\gamma_{km}^{t,d}/\gamma$, where $\gamma_{km}^{t,d}/\gamma$ can be interpreted as follows :

$$
\begin{aligned}
\gamma_{km}^{t,d}/\gamma &= \frac{p(\hat{h}_t^d = k, \hat{m}_t^d = m \mid \mathcal{X})}{\sum_{d=1}^{D}\sum_{t=1}^{T_d} p(\hat{h}_t^d = k, \hat{m}_t^d = m \mid \mathcal{X})} \\
&= \frac{p(\hat{h} = k, \hat{m} = m, \hat{t} = t, \hat{d} = d \mid \mathcal{X})}{\sum_{d=1}^{D}\sum_{t=1}^{T_d} p(\hat{h} = k, \hat{m} = m, \hat{t} = t, \hat{d} = d \mid \mathcal{X})} \\
&= \frac{p(\hat{h} = k, \hat{m} = m, \hat{t} = t, \hat{d} = d \mid \mathcal{X})}{p(\hat{h} = k, \hat{m} = m \mid \mathcal{X})} \\
&= p(\hat{t} = t, \hat{d} = d \mid \mathcal{X}, \hat{h} = k, \hat{m} = m)
\end{aligned}
$$

where $\hat{\cdot}$ means random variable. As each (d,t) pair identifies one sample, the above quantity is the likelihood of the $(d,t)^{th}$ sample generated by the $(k,m)^{th}$ Dirichlet.

We list the change of variables as follows :

$$
\gamma \leftarrow \gamma_{km}^{\mathtt{t,d}} \tag{7}
$$

$$
a_n \leftarrow a_{km,n} \tag{7'}
$$

$$
\log \bar{x}_n \leftarrow (\gamma_{km}^{\mathtt{t,d}}/\gamma) \log x_{\mathtt{tn}}^{\mathtt{d}} \tag{7''}
$$

where $n = 1, \cdots, N$.

Then the problem can be stated as finding the maximal of

$$
\begin{aligned}
L(a) = \; & \log\Gamma(\sum_{n=1}^{N} a_n) - \log\Gamma(a_{\mathtt{n}}) + a_{\mathtt{n}} \log \bar{x}_{\mathtt{n}} \\
& \text{s.t. } a_n > 0, n = 1, ..., N
\end{aligned}
\tag{8}
$$

In this part, we use the Newton method to maximize Eq.8. As this subproblem is convex[3], there exists unique global maximum.

Newton method : Taking first and second order derivatives of Eq.6, we have the following gradient and Hessian :

$$
\frac{\partial L}{\partial a_{k,mn}} (\triangleq g_{k,mn}) = \gamma_{km}^{\mathtt{t,d}} \Psi_0(\sum_{n=1}^{N} a_{k,mn}) - \gamma_{km}^{\mathtt{t,d}} \Psi_0(a_{k,mn}) + \gamma_{km}^{\mathtt{t,d}} \log x_{\mathtt{tn}}^{\mathtt{d}} \tag{9}
$$

$$
\frac{\partial^2 L}{\partial a_{k,mn}^2} = \gamma_{km}^{\mathtt{t,d}} \Psi_1(\sum_{n=1}^{N} a_{k,mn}) - \gamma_{km}^{\mathtt{t,d}} \Psi_1(a_{k,mn}) \tag{10}
$$

[3]The concavity of the log likelihood of the Dirichlet distribution comes from the fact that the Dirichlet distribution belongs to the exponential family. Also see [Ron86, Ron89] for another direct proof.

$$
\begin{aligned}
\frac{\partial^2 L}{\partial a_{k,mn}\partial a_{k,mn'}} &= \gamma_{km}^{\mathtt{t,d}} \Psi_1(\sum_{n=1}^{N} a_{k,mn}) \quad n \neq n' &(11)\\
\frac{\partial^2 L}{\partial a_{k,mn}\partial a_{k',m'n'}} &= 0 \quad k \neq k' \quad \text{or} \quad m \neq m' &(12)
\end{aligned}
$$

From Eq. 12, we can see that parameters of different mixture components are uncorrelated. So we can find the best parameters of each individual mixture component respectively. In the following, we calculate the $(k,m)^{th}$ mixture component using the notation alleviation in Eq.7.

The Hessian can be written in matrix form as

$$
\mathrm{H} = -\Lambda + z\mathbf{1}\mathbf{1}^{\mathrm{T}} \tag{13}
$$

where Λ is a diagonal matrix

$$
\begin{aligned}
\Lambda^{nn} &= \Psi_1(a_n) \geq 0 \quad \text{and} \\
z &= \Psi_1(\sum_{n=1}^{N} a_n) \geq 0 \quad .
\end{aligned}
$$

The non-negative properties above come from the properties of trigamma function.

The sub-problem of estimating a single Dirichlet component is a convex problem [Ron89]. And there exists globally optimal solution.

Then the Newton update is

$$
a^{new} = a^{old} - \mathrm{H}^{-1} g
$$

or equivalently in component case

$$
a_n^{new} = a_n^{old} + \frac{g_n^{old}}{\Lambda^{nn}} + \frac{1}{\Lambda^{nn}} \cdot \frac{\sum_{n'=1}^{N} g_{n'}^{old} \cdot (\Lambda^{n'n'})^{-1}}{z^{-1} - \sum_{n'=1}^{N} (\Lambda^{n'n'})^{-1}}
$$

where

$$
g_n^{old} = \Psi_0(\sum_{n'=1}^{N} a_{n'}^{old}) - \Psi_0(a_n^{old}) + \log \bar{x}_n \quad . \tag{14}
$$

See Appendix B for the inverse of the Hessian matrix.

To avoid a becomes negative during the iterations, we simply set the negative components to a small positive value. Or we can also use the Ronning's method to reset all components of a to the samples' minimal value [Ron89]. Bouguila and et al. propose to re-parameterize a as e^b. However, after this parameterization, the original convex property will not remain. Please see Appendix B for more detailed analysis.

The whole algorithm is listed at Algorithm 1.

Algorithm 1: Estimation of the m^{th} mixture component associated with k^{th} hidden state by Newton method.

Input:
- $\left\{a_n^0 \leftarrow a_{k,mn}^0 \mid n = 1, ..., N\right\}$
- $\left\{\log \bar{x}_n \leftarrow \frac{\gamma_{km}^{\mathtt{t,d}} \log x_{\mathtt{tn}}^{\mathtt{d}}}{\gamma} \mid \gamma = \gamma_{km}^{\mathtt{t,d}}, n = 1, ..., N\right\}$

Result:
- $\{a_{k,mn} \leftarrow a_n \mid n = 1, ..., N\}$

begin
  Initialize $a_n \leftarrow a_n^0$ for all $n = 1, ..., N$;
  $s \leftarrow +\infty$;
  while $s \geq \epsilon$ **do**
    for $n = 1$ **to** N **do**
      $g_n \leftarrow \Psi_0(\sum_{n=1}^N a_n) - \Psi_0(a_n) + \log \bar{x}_n$;
      $\lambda_n \leftarrow \Psi_1(a_n)$;
    end
    $z \leftarrow \Psi_1(\sum_{n=1}^N a_n)$;
    for $n = 1$ **to** N **do**
      $h_n \leftarrow \frac{g_n}{\lambda_n} + \frac{1}{\lambda_n} \cdot \frac{\sum_{n'=1}^N g_{n'} \cdot (\lambda_{n'})^{-1}}{z^{-1} - \sum_{n'=1}^N (\lambda_{n'})^{-1}}$;
    end
    $a \leftarrow a + h$;
    if $\exists n$ *s.t.* $a_n \leq 0$ **then**
      $a \leftarrow \max(a, \epsilon_1)$;
      $s \leftarrow +\infty$;
    else
      $s \leftarrow h^T \cdot g$;
    end
  end
  return $\{a_n \mid n = 1, ..., N\}$
end

2.3.2 Initialization Algorithm

Parameter initialization is an important issue for both the Algorithm 1 above and the Algorithm 3 in the latter of this section. In this part, we first give the initialization algorithm for single Dirichlet. Based on this algorithm, we further propose an algorithm to initialize the mixture of Dirichlet. Finally, we deal with the initialization problem of the HMM+DM model.

Initializing single Dirichlet In the following, we assume x is a $K \times 1$ vector, following Dirichlet distribution of $\mathrm{Dir}(a)$. As there are K parameters in vector a, we should construct K functions to estimate a. It is easy to see the following integrals :

$$\langle x_k \rangle = \frac{Z([a_1, ..., a_k + 1, ..., a_K])}{Z([a_1, ..., a_K])} = \frac{a_k}{\sum_{k'=1}^K a_{k'}} \tag{15}$$

$$\left\langle (x_k)^2 \right\rangle = \frac{Z([a_1, ..., a_k + 2, ..., a_K])}{Z([a_1, ..., a_K])} = \frac{(a_k + 1) \cdot a_k}{(1 + \sum_{k=1}^K a_k)(\sum_{k=1}^K a_k)} \tag{16}$$

where

$$Z([a_1, ..., a_K]) = \frac{\prod_{k=1}^K \Gamma(a_k)}{\Gamma(\sum_{k=1}^K a_k)}$$

is the normalization constant for Dirichlet distribution.

Eq. 15 offers $K-1$ equations. So we can use the second equation (Eq. 16) to get another constraint as :

$$\sum_{k=1}^{K} a_k = \frac{\langle x_i \rangle - \langle x_i^2 \rangle}{\langle x_i^2 \rangle - \langle x_i \rangle^2} \tag{17}$$

for any $i = 1, \cdots, K$. Note : In general, there is no theoretical guarantee that Eq.17 is positive. But in the case that all the random variables are proportional data, the numerator of Eq.17 is positive. The denominator of Eq.17 is a variance, and therefore positive. So for proportional data, Eq.17 is positive.

So the estimation of the parameter is

$$\hat{a}_k = \frac{\langle x_1 \rangle - \langle x_1^2 \rangle}{\langle x_1^2 \rangle - \langle x_1 \rangle^2} \cdot \langle x_k \rangle, \qquad k = 1, \cdots, K.$$

Another intuitive way is to use several or all of the second order moments to estimate the scale parameter $\sum_k a_k$:

$$\hat{a}_k = \frac{1}{K} \sum_{i=1}^{K} \frac{\langle x_i \rangle - \langle x_i^2 \rangle}{\langle x_i^2 \rangle - \langle x_i \rangle^2} \qquad or \qquad \hat{a}_k = \sqrt[K]{\prod_{i=1}^{K} \frac{\langle x_i \rangle - \langle x_i^2 \rangle}{\langle x_i^2 \rangle - \langle x_i \rangle^2}}, \qquad k = 1, \cdots, K$$

which are the algebraic and geometric mean over all estimates. Ronning [Ron89] suggests instead using the samples' minimum value to set the parameters. In our practice, we prefer to use the geometric mean as the initial parameters.

Initializing mixture of Dirichlet Bouguila, and etc. [BZV04] propose the initialization algorithm based on fuzzy C-means and methods of moments (MM). Here, we use a similar procedure based on K-Means algorithm to initialize the parameters. We have also tried the Gaussian mixture model (GMM). Our experiences show that K-Means seems better than GMM.

See Algorithm 2 for the complete algorithm.

Initializing HMM+DM In case of HMM+DM, there are KM Dirichlet components, which are grouped into to K groups, and each group is a Dirichlet mixture model with M components.

Ideally, we could initialize the parameters in the following way : First, we initialize these KM Dirichlet distributions using Algorithm 2, where we discard all the temporal constraints. Then in order to take the temporal constraints into account, we propose to fit a relaxed HMM+DM model : there are KM hidden states, and each hidden state corresponds one single Dirichlet distribution. We get the following relaxed transition probabilities :

$$p(h, m \mid h', m') \quad .$$

Then we could find the best transition probabilities by minimizing the following distance between the true parameters (relaxed transition probabilities) and the parameters in the factorized forms :

$$\sum_{h'=1}^{K} \sum_{m'=1}^{M} \mathrm{KL}\left[p(h, m \mid h', m') \parallel p(h \mid h') p(m \mid h)\right] \quad .$$

By introducing Lagrange multipliers, we can easily find the minimum are achieved by

$$p(h \mid h') \propto \sum_{m'=1}^{M} \sum_{m=1}^{M} p(h, m \mid h', m')$$

$$p(m \mid h) \propto \sum_{h'=1}^{K} \sum_{m'=1}^{M} p(h, m \mid h', m')$$

However, as there is a permutation-invariant property for the above initialization, after permuting the rows and corresponding columns of $p(h, m \mid h', m')$ [4], we may get significantly different initial values of both $p(h \mid h')$ and $p(m \mid h)$. So by taking the permutation-invariant property into account, we propose the following criterion for find the best $p(h \mid h')$ (or B) and $p(m \mid h)$ (or C) :

$$\min_{\sigma} \min_{p(h|h'),p(m|h)} \sum_{h'=1}^{K} \sum_{m'=1}^{M} \mathrm{KL}\left[p_\sigma(h, m \mid h', m') \parallel p(h \mid h')p(m \mid h)\right] \quad .$$

where σ is a permutation among KM elements, and the the minimization is over all $(KM)!$ such permutations. It will be soon intractable with large K and M.

In current report, we simply initialize KM Dirichlet components using Algorithm 2 and then group orderly each M components to one Markov hidden state to get A. For π, B and C, we randomly initialize them. Better solution for initialization in the dynamic case is left for future investigation.

Algorithm 2: Initialization algorithm for the Dirichlet mixture.

Input:
- Number of mixture components : M.
- A data set for initializing parameter : $X = \left\{x_t \in \mathbb{R}_+^{N\times 1} \mid t = 0, ..., T; x_t \geq 0; \sum_{n=1}^{N} x_{tn} = 1\right\}$

Result:
- $A = [a_{mn}] \in \mathbb{R}_+^{M\times N}$, with each row $a_{m\cdot}$ corresponding one Dirichlet component.
- $\pi \in \mathbb{R}_+^{M\times 1}$, the prior probabilities for each Dirichlet.

```
begin
    Apply GMM or K-means algorithm on the data set X to get M clusters with the posterior of each
    sample belonging to each cluster : p(m | x_t);
    s ← 0;
    for m = 1 to M do
        y ← zeros(N, 1);
        z ← zeros(N, 1);
        π_m ← 0;
        for t = 0 to T do
            y ← y + x_t · p(m | x_t);
            z ← z + x_t ⊙ x_t · p(m | x_t);
            π_m ← π_m + p(m | x_t);
        end
        s ← π_m;
        w ← sum(log((y − z) ⊘ (z − y ⊙ y)))/N;
        a_m· ← y' · e^w;
    end
    for t = 0 to T do
        π_m ← π_m/s;
    end
    return A and π
end
```

2.3.3 Parameter estimation algorithm of HMM+DM :

Before arriving the final algorithm for parameter estimation of HMM+DM, we should give the stopping criterion for the whole algorithm. As EM is a low bound maximization algorithm, we should only check the

[4] here we treat $p(h, m \mid h', m')$ as a $KM \times KM$ matrix with each row being a distribution

low bound each time. The low bound consists two term : the entropy term and the energy term :

$$\log(p(\mathcal{X})) \quad \geq \quad E(A, B, \pi, \gamma \mid \mathcal{H}) + Etr(\mathcal{H}) \tag{18}$$

Where the energy term is given by Eq. 2, and the entropy term is as follows :

$$\begin{aligned} Etr(\mathcal{H}) &= -\langle \log q(\mathcal{H}) \rangle_{q(\mathcal{H})} \\ &= 2\eta_{\mathtt{k}}^{\mathtt{t,d}} \log \eta_{\mathtt{k}}^{\mathtt{t,d}} - \eta_{\mathtt{k}}^{0,\mathtt{d}} \log \eta_{\mathtt{k}}^{0,\mathtt{d}} - \eta_{\mathtt{k}}^{T_d,\mathtt{d}} \log \eta_{\mathtt{k}}^{T_d,\mathtt{d}} - \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log \gamma_{\mathtt{km}}^{\mathtt{t,d}} - \xi_{\mathtt{ij}}^{\mathtt{t,d}} \log \xi_{\mathtt{ij}}^{\mathtt{t,d}} \quad . \end{aligned} \tag{19}$$

See Appendix B for the detailed derivation.

Then the low bound can be then calculated as

$$\begin{aligned} \mathcal{L}(\theta \mid X) =& \gamma_{\mathtt{km}}^{0,\mathtt{d}} \log \pi_{\mathtt{k}} + \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log C_{\mathtt{km}} + \xi_{\mathtt{ij}}^{\mathtt{t,d}} \log b_{\mathtt{ij}} \\ &+ \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log \Gamma(\sum_{n=1}^{N} a_{\mathtt{k,mn}}) - \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log \Gamma(a_{\mathtt{k,mn}}) + \gamma_{\mathtt{km}}^{\mathtt{t,d}} (a_{\mathtt{k,mn}} - 1) \log x_{\mathtt{tn}}^{\mathtt{d}} \\ &+ 2\eta_{\mathtt{k}}^{\mathtt{t,d}} \log \eta_{\mathtt{k}}^{\mathtt{t,d}} - \eta_{\mathtt{k}}^{0,\mathtt{d}} \log \eta_{\mathtt{k}}^{0,\mathtt{d}} - \eta_{\mathtt{k}}^{T_d,\mathtt{d}} \log \eta_{\mathtt{k}}^{T_d,\mathtt{d}} - \gamma_{\mathtt{km}}^{\mathtt{t,d}} \log \gamma_{\mathtt{km}}^{\mathtt{t,d}} - \xi_{\mathtt{ij}}^{\mathtt{t,d}} \log \xi_{\mathtt{ij}}^{\mathtt{t,d}} \end{aligned} \tag{20}$$

Finally, we list the parameter estimation algorithm of HMM+DM in Algorithm 3.

Algorithm 3: Parameter Estimation of HMM+DM

Input:
- Number of the mixture components : M
- Number of the hidden states : K
- The Data set $\mathcal{X} = \{x_t^d \mid x_t^d \in \mathbb{R}_+^N, \sum_{n=1}^N x_{tn} = 1, d = 1, ..., D, t = 0, \cdots, T_d\}$

Result: Maximal likelihood estimation of :
$\theta = \left\{A \in \mathbb{R}_+^{M\times N\times K}, B \in \mathbb{R}_+^{K\times K}, C \in \mathbb{R}_+^{K\times M}, \pi \in \mathbb{R}_+^{K\times 1}\right\}$

begin
  Call Algorithm 2 with all or part of the data to initialize $\hat{A}$ with KM components;
  Initialize $A^{new}(m, :, k)$ as $(m(k-1)+1)^{th}$ row of $\hat{A}$;
  Initialize $\pi^{new}, B^{new}, C^{new}$ randomly;
  Initialize $L^{old} \leftarrow \infty$;
  $L^{new} \leftarrow \mathcal{L}(\theta^{new} \mid \mathcal{X})$ by Eq.20;
  while $\left|L^{old} - L^{new}\right| \geq \epsilon$ **do**
    Update $\theta^{old} \leftarrow \theta^{new}$;
    `// E step -- Inference`
    Inference using Algorithm 5 with θ^{old} to get $\gamma_{km}^{t,d}, \xi_{kk'}^{t,d}$;
    `// M step`
    For $k, k' = 1, \cdots, K$, $m = 1, \cdots, M$ and $n = 1, \cdots, N$, calculate the following quantities :
      $\hat{\gamma}_{km} \leftarrow \gamma_{km}^{\mathtt{t,d}}$; $\hat{\xi}_{kk'} \leftarrow \xi_{kk'}^{\mathtt{t,d}}$; $\hat{\pi}_k \leftarrow \gamma_{\mathtt{km}}^{0,\mathtt{d}}$; $\log \bar{x}_{km,n} \leftarrow \gamma_{km}^{\mathtt{t,d}} \log x_{\mathtt{tn}}^{\mathtt{d}}$;
    Update π^{new}, B^{new}, C^{new} by normalizing $\hat{\pi}_k$, $\hat{\xi}_{kk'}$ and $\hat{\gamma}_{km}$;
    For $k = 1, \cdots, K$ and $m = 1, \cdots, M$:
      Update $A^{new}(m, :, k)$ by the Algorithm 1 with $A^{old}(m, :, k)$ and $\log \bar{x}_{km,\cdot}/\hat{\gamma}_{km}$;
    $L^{old} \leftarrow L^{new}$;
    $L^{new} \leftarrow \mathcal{L}(\theta^{new} \mid X)$ by Eq.20;
  end
  return $A^{new}, B^{new}, C^{new}, \pi^{new}$
end

2.4 Static Dirichlet mixture model revisited

Here, we present how to estimate the static Dirichlet mixture model as a special case of HMM+DM. For static model, there is no link between consecutive two hidden states. Then each state can be viewed as a new start state. Or equivalently the transition probability matrix B has the following form :

$$b_{ij} = \pi_j \quad \text{for all } i, j = 1, ..., K$$

for static model. In this case, there is no need to use two hidden variable h_t and m_t to index one Dirichlet distribution. So without loss of generality, in the following, we assume that the number of mixture components for each state m_t is one (or $M = 1$). And also there is no need to use d to index which sequence. We could simply pool together all the samples from different sequences. So in the following, we omit the sub- and superscripture for m_t and d.

Algorithm 4: Parameter Estimation of static Dirichlet mixture model.

Input:
– Number of the mixture components : M
– The Data set $X = \{x_t \mid x_t \in \mathbb{R}_+^N, \sum_{n=1}^N x_{tn} = 1, t = 0, \cdots, T\}$
Result:
– MLE of $\hat{A} \in \mathbb{R}_+^{M \times N}$ and $\hat{\pi} \in \mathbb{R}_+^{M \times 1}$

```
begin
    Initialize A^new by the Algorithm 2;
    Random initialize π^new as a probability vector;
    L^new ← ℒ(A^new, π^new | X);
    L^old ← ∞;
    while |L^old − L^new| ≥ ε do
        Update A^old ← A^new and π^old ← π^new;
        // E step -- Inference
        for t = 0 to T do
            s ← 0;
            for k = 1 to K do
                ᾱ_k^t ← π_k^old · Dir(x_t· | a_k·^old) and s ← s + ᾱ_k^t;
            end
            for k = 1 to K do
                ᾱ_k^t ← ᾱ_k^t / s;
            end
        end
        // M step
        s ← 0;
        for k = 1 to K do
            π_k^new ← Σ_{t=0}^T ᾱ_k^t and s ← s + π_k^new;
            a_k·^new ← by calling Algorithm 1 with (π_k^new, a_k·^old, x̄_k);
        end
        for k = 1 to K do
            π_k^new ← π_k^new / s;
        end
        L^old ← L^new;
        L^new ← ℒ(A^new, π^new | X);
    end
    return A^new and π^new
end
```

2.4.1 E step

Then, we can see that the inference procedure (E step) degenerates to the following simple form.
For $t = 0$,

$$\bar{\alpha}_k^0 \quad \propto \quad \pi_k \cdot \mathrm{Dir}\left(x_{0\cdot} \mid a_{k\cdot}\right)$$

and for $t = 1, ..., T$,

$$\bar{\alpha}_k^t = \frac{\sum_{k'} \bar{\alpha}_{k'}^{t-1} \cdot \pi_k \cdot p(x_t \mid h_t = k)}{\sum_{k',k''} \bar{\alpha}_{k'}^{t-1} \cdot \pi_{k''} \cdot p(x_t \mid h_t = k'')}$$

by deleting m, C and $b_{ij} = \pi_j$ in the normalized Alpha recursion. As $\sum_{k'} \bar{\alpha}_{k'}^{t-1} = 1$, we get

$$\bar{\alpha}_k^t \propto \pi_k \cdot p(x_t \mid h_t = k) = \pi_k \cdot \mathrm{Dir}\,(x_{t\cdot} \mid a_{k\cdot})$$

So, in summary, for all $t = 0, \cdots, T$,

$$\bar{\alpha}_k^t \quad \propto \quad \pi_k \cdot \mathrm{Dir}\,(x_{t\cdot} \mid a_{k\cdot})$$

And

$$\gamma_k^t \quad = \quad \sum_{k'=1}^{K} \frac{\bar{\alpha}_k^t \cdot b_{kk'}}{\sum_{i=1}^{K} \bar{\alpha}_i^t \cdot b_{ik'}} \cdot \gamma_{k'}^{t+1} = \sum_{k'=1}^{K} \frac{\bar{\alpha}_k^t}{\sum_{i=1}^{K} \bar{\alpha}_i^t} \cdot \gamma_{k'}^{t+1} = \bar{\alpha}_k^t$$

$$\xi_{ij}^t \quad = \quad \frac{\bar{\alpha}_i^t \cdot b_{ij}}{\sum_k \bar{\alpha}_k^t \cdot b_{kj}} \cdot \gamma_j^{t+1} = \frac{\bar{\alpha}_i^t}{\sum_k \bar{\alpha}_k^t} \cdot \gamma_j^{t+1} = \bar{\alpha}_i^t \cdot \gamma_j^{t+1} = \bar{\alpha}_i^t \cdot \bar{\alpha}_i^{t+1}$$

2.4.2 M step

In the M step, we maximize the energy term, which could be simplified from Eq.2 by removing parameters B and U which is deterministic here, and scriptures d and m, , and also replacing b_{ij} with π_j :

$$\begin{aligned} E(A, \pi \mid X, H =& \gamma_{\mathtt{k}}^0 \log \pi_{\mathtt{k}} + \xi_{\mathtt{ij}}^{\mathtt{t}} \log \pi_{\mathtt{j}} + \gamma_{\mathtt{k}}^{\mathtt{t}} \log \Gamma(\sum_{n=1}^{N} a_{kn}) - \gamma_{\mathtt{k}}^{\mathtt{t}} \log \Gamma(a_{\mathtt{kn}}) + \gamma_{\mathtt{k}}^{\mathtt{t}}(a_{\mathtt{kn}} - 1) \log x_{\mathtt{tn}} \\ =& \bar{\alpha}_{\mathtt{k}}^{\mathtt{t}} \log \pi_{\mathtt{k}} + \bar{\alpha}_{\mathtt{k}}^{\mathtt{t}} \left(\log \Gamma(\sum_{n=1}^{N} a_{kn}) - \log \Gamma(a_{\mathtt{kn}}) + a_{\mathtt{kn}} \cdot \bar{x}_{\mathtt{kn}} \right) - \underbrace{\bar{\alpha}_{\mathtt{k}}^{\mathtt{t}} \log x_{\mathtt{tn}}}_{Const.} \end{aligned} \tag{21}$$

where

$$\bar{x}_{kn} = \frac{\bar{\alpha}_k^{\mathtt{t}} \log x_{\mathtt{tn}}}{\sum_{t=1}^{T} \bar{\alpha}_k^t} = \frac{\sum_{t=0}^{T} \bar{\alpha}_k^t \log x_{tn}}{\sum_{t'=0}^{T} \bar{\alpha}_k^{t'}}$$

So for parameter π we have the following update

$$\pi_k \propto \bar{\alpha}_k^{\mathtt{t}} = \sum_{t=0}^{T} \bar{\alpha}_k^t \quad k = 1, ..., K \quad .$$

For k^{th} Dirichlet, we optimize $a_{k\cdot}^{new}$ by calling the Algorithm 1 with $a_{k\cdot}^{old}$ as the initial parameter, $\bar{\alpha}_k^{\mathtt{t}}$ as the scale parameter γ, and $\bar{x}_{k\cdot}$ as the expected variable in canonical form. In the very beginning, we call the Algorithm 2 to initialize A.

2.4.3 The whole EM algorithm for mixture of Dirichlet

Before arriving the final algorithm, we should give the stopping criterion. We similarly check the low bound each time :

$$\log(p(X)) \geq E(A, \pi \mid X, H) + Etr(H) \triangleq \mathcal{L}(A, \pi \mid X) \tag{22}$$

TAB. 1 – Estimation of one single Dirichlet distribution with 100 random samples.

	Real Parameter	**Para. by MM**	**Estimated Para.**
Dim.1	3.000	3.061	2.991
Dim.2	2.000	2.164	2.134
Dim.3	4.000	3.897	3.759
Dim.4	5.000	5.101	4.930
Dim.5	8.000	8.261	7.962
Dim.6	10.000	10.993	10.724
Dim.7	20.000	19.859	19.379
Euclidian dist.	0.000	1.061	0.997
Average data log likelihood	11.501	11.550	11.552

TAB. 2 – Estimation of one single Dirichlet distribution with 2000 random samples.

	Real Parameter	**Para. by MM**	**Estimated Para.**
Dim.1	3.0000	3.0667	3.0658
Dim.2	2.0000	2.0039	1.9706
Dim.3	4.0000	4.0292	4.0269
Dim.4	5.0000	4.9658	4.9676
Dim.5	8.0000	8.0515	8.0428
Dim.6	10.0000	10.1336	10.1031
Dim.7	20.0000	20.0790	20.0442
Euclidian dist.	0.0000	0.1823	0.1462
Average data log likelihood	11.5976	11.5989	11.5992

where the energy term $E(A, \pi \mid X, H)$ can be calculated according Eq.21, and the entropy term is

$$\begin{aligned} Etr(H) &= -\sum_{t=0}^{T}\sum_{k=1}^{K} \bar{\alpha}_k^t \log \bar{\alpha}_k^t \\ \text{or} \quad &= -\bar{\alpha}_{\mathtt{k}}^{\mathtt{t}} \log \bar{\alpha}_{\mathtt{k}}^{\mathtt{t}} \quad \text{by summation convetion} \end{aligned}$$

Although the static case is a special case of the Algorithm 3, for clarity, we list this algorithm in the Algorithm 4.

3 Experiments on artificial data

In this section, we do some experiments on randomly generated data to verify the proposed algorithms.

3.1 Parameter estimation of one single Dirichlet density

In Table.1 and Table.2 we test the estimation of one single Dirichlet distribution with 100 and 2000 samples of 7 dimension. The ”Para. by MM” in both tables refers to parameters estimated by moment matching algorithm (Algorithm 2 with one mixture component or $M = 1$). And the ”Estimated Para.” refers to parameters estimated by the Newton method (Algorithm 1). Each time, ”Estimated Para.” achieves the highest data log-likelihood as we expect. The Algorithm 1 with $\epsilon = 10^{-5}$ in these two cases take both three Newton iterations to get converged.

TAB. 3 – Estimation of the mixture of Dirichlet with 2000 random samples.

	π	A					Aver. likelihood
Real para.	0.20	3	3	4	6	5	5.2324
	0.30	10	7	1	9	10	
	0.50	2	6	2	9	10	
Init. para.	0.2940	2.3979	4.1703	3.5407	8.5459	5.6007	5.1325
	0.3180	8.7408	5.8869	1.3761	7.6242	8.6959	
	0.3880	2.217	6.8453	2.1893	9.0607	12.2773	
Est. para.	0.1884	3.0373	2.866	4.1019	5.6673	4.8019	5.2341
	0.3046	9.377	6.7094	0.98267	8.6981	9.8639	
	0.5070	1.9929	6.1727	2.1247	9.3191	10.0062	

3.2 Parameter estimation of Dirichlet mixtures

We generated 2000 samples from a mixture of Dirichlet model with three Dirichlet distributions. The samples lie in 5-dimensional space, or more strictly the 4-dimensional probability simplex. The real parameters are shown in the first row "Real para." of Table 3, where each sub-row is one Dirichlet. The second row "Init. para." is the parameters estimated by Kmeans+Moment Matching (Algorithm 2). The last row "Est. para." shows the estimated parameters by the EM algorithm described in Algorithm 4.

Fig.3 illustrates that the algorithm performs properly. The "Aver. likelihood" is the data likelihood (Eq.22) divided by the number of samples to avoid big numbers and make it invariant to the number of samples. The likelihood increases monotonically in each EM steps.

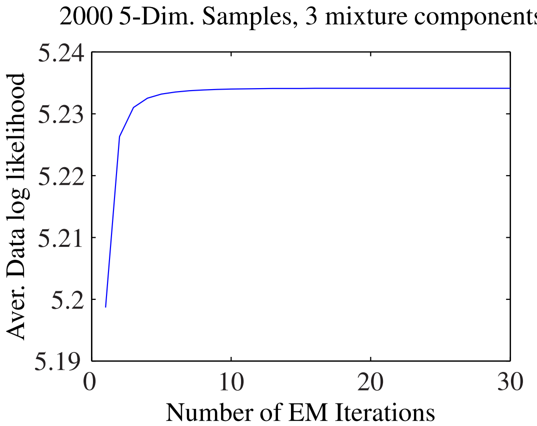

FIG. 3 – Experiments on parameter estimation of Dirichlet mixture model.

3.3 Parameter estimation of the HMM+DM

In order to test the parameter estimation of HMM+DM model, we randomly generate 6 sequences, the length of each sequence is

$$\begin{pmatrix} 1156 & 2430 & 3034 & 3810 & 4456 & 4751 \end{pmatrix} .$$

There are all together 19637 samples. The real parameters and estimated parameters are listed in the Table 4, where "A.L.L" is the average data log likelihood. Other parameters are $D = 6$, $K = 2$, $M = 3$ and $N = 4$. Note : in this experiment, we randomly initialize π, B and C. Unfortunately We didn't record the initial values. The values of π, B and C listed in the "Init. para" of the table are randomly generated afterwards.

TAB. 4 – Estimation of the HMM+DM model with randomly generated 6 sequences, 19637 samples.

	Real para.				Init. para.				Est. para.			
π	0.30	0.70			0.322	0.678			0.5081	0.4919		
B	0.033	0.967			0.926	0.074			0.551	0.449		
	0.445	0.555			0.473	0.527			0.966	0.034		
C	0.308	0.559	0.134		0.848	0.387	0.785		0.317	0.263	0.421	
	0.259	0.325	0.416		0.192	0.475	0.250		0.547	0.147	0.306	
A	6.0	5.0	10.0	5.0	13.631	8.364	3.347	5.258	10.193	9.140	3.048	3.986
	1.0	7.0	8.0	10.0	8.494	3.800	9.047	6.137	5.127	5.089	2.043	7.052
	9.0	9.0	3.0	10.0	2.908	2.521	10.643	5.041	1.935	0.993	2.945	2.881
	5.0	5.0	2.0	7.0	1.475	8.007	8.560	11.184	1.018	7.192	8.026	10.190
	10.0	9.0	3.0	4.0	7.782	11.113	3.277	6.520	8.085	7.587	2.813	7.903
	2.0	1.0	3.0	3.0	4.054	2.843	2.892	9.250	6.073	4.926	10.003	5.069
A.L.L.	2012.4				1845.4				2013.1			

From Table 4, we can see the estimated parameters coincide well with the real parameters after eliminating the permutation effects. In fact, we could find the correct order of the estimated parameters by values of A. If we use the matlab convention on multidimensional array (A is a $3 \times 4 \times 2$ array), we first should reorder the third dimension of A. Then for each $A(:,:,k)$ (k=1,2), we reorder the first dimension respectively. Then we should reorder B, π and C accordingly. After these reorders, we get the following estimated parameters :

$$\begin{array}{ccccl}
\text{real parameters} & & \text{estimated parameters} & & \\
\begin{pmatrix} 0.30 & 0.70 \end{pmatrix} & \sim & \begin{pmatrix} 0.4919 & 0.5081 \end{pmatrix} & & \pi \\
\begin{pmatrix} 0.033 & 0.967 \\ 0.445 & 0.555 \end{pmatrix} & \sim & \begin{pmatrix} 0.034 & 0.966 \\ 0.449 & 0.551 \end{pmatrix} & & B \\
\begin{pmatrix} 0.308 & 0.559 & 0.134 \\ 0.259 & 0.325 & 0.416 \end{pmatrix} & \sim & \begin{pmatrix} 0.306 & 0.547 & 0.147 \\ 0.263 & 0.317 & 0.421 \end{pmatrix} & & C \\
\left(\begin{array}{cccc} 6.0 & 5.0 & 10.0 & 5.0 \\ 1.0 & 7.0 & 8.0 & 10.0 \\ 9.0 & 9.0 & 3.0 & 10.0 \\ \hline 5.0 & 5.0 & 2.0 & 7.0 \\ 10.0 & 9.0 & 3.0 & 4.0 \\ 2.0 & 1.0 & 3.0 & 3.0 \end{array}\right) & \sim & \left(\begin{array}{cccc} 6.073 & 4.926 & 10.003 & 5.069 \\ 1.018 & 7.192 & 8.026 & 10.190 \\ 8.085 & 7.587 & 2.813 & 7.903 \\ \hline 5.127 & 5.089 & 2.043 & 7.052 \\ 10.193 & 9.140 & 3.048 & 3.986 \\ 1.935 & 0.993 & 2.945 & 2.881 \end{array}\right) & & A
\end{array}$$

where estimated A, B, C coincide well with the real ones. However there is a big difference in the estimated π. The reason is that estimating π depending on the number of sequences. Currently, there are only 6 sequences, which are far from enough to get a good estimation of π.

Fig.4 illustrates that the algorithm performs properly. The log-likelihood is composed by two terms : Energy and Entropy. The energy term is further composed by two contributors : Energy from Dirichlet mixtures (i.e. A), and Energy from the initial and transition probabilities (i.e. π,B and C). We can see that in the first ten iterations or so, the contribution of the energy from Dirichlet mixtures are more significant than that from the initial and transition probabilities. However, in the last 150 iterations or so, the energy from the initial and transition probabilities becomes more significant than that from the Dirichlet mixtures.

By applying the Eq.1, we get the hard clustering decisions from the soft one (γ itself). After eliminating the permutation problem, we can get the confusion matrices among all the 19637 samples from 6 sequences. See Table 5 for the confusion matrices. The classification accuracies are $e_h = 0.8827$ for the hidden states h (or one minus the frame error rate, see Section 3.4), $e_m = 0.7921$ for the indicators m. The reason that $e_m < e_h$ is that according to Eq.1, if the system make the wrong decision on the first step for h, then the second step for deciding m will be a blindly guess.

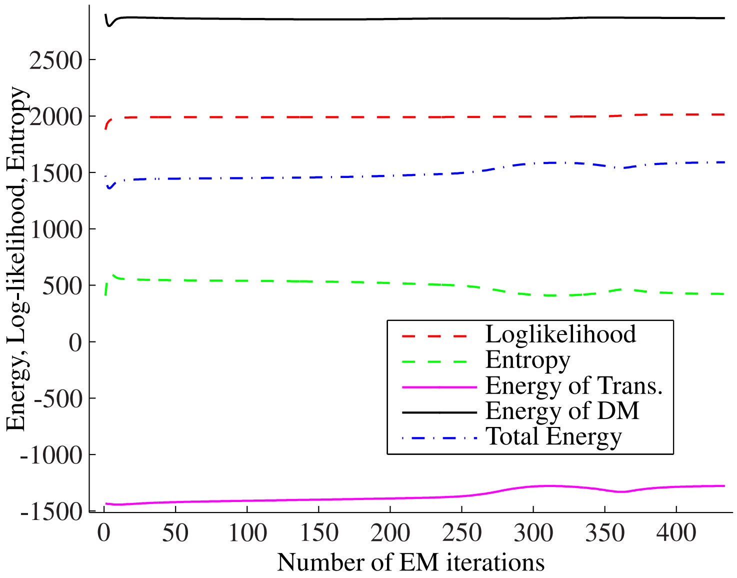

FIG. 4 – Experiments on parameter estimation of HMM+DM model

TAB. 5 – Confusion matrix for the 19637 samples in 6 sequences.

	$h=1$	$h=2$
$h=1$	4682	1433
$h=2$	870	12652

	$m=1$	$m=2$	$m=3$
$m=1$	4147	577	672
$m=2$	486	7081	214
$m=3$	1509	624	4327

3.4 Comparisons of HMM+GMM and HMM+DM

In this part, we compare HMM+GMM and HMM+DM on simulated data. We will show that if the data are indeed generated by some HMM+DM model, the estimated HMM+DM model will achieve better results than the HMM+GMM model.

Performance measures We use the *word error rate* (WER) and the *frame error rate* (FER) as measures to compare the two models.

WER is the sum of insertions, deletions, and substitutions, divided by the sequence length,

$$\text{WER} = \frac{\text{Sub+Del+Ins}}{\text{Sequence length}} \times 100\%$$

. As each word, or sequence, has its own WER, we average them over all sequences to get the WER measure for particular model.

FER is defined as one minus the ratio between the correctly recognized frames and the number of all frames,

$$\text{FER} = (1 - \frac{\text{correct frames}}{\text{total frames}}) \times 100\%$$

. We also use the confusion matrices. Please see the experimental part of [ZGPBM06] for more details about these measures.

Comparison protocol The comparison protocol is as follows : 1000 sequences with lengths uniformly distributed on $[1, 20]$ are generated by the same HMM+DM model in Section 3.3. We record at each time which

TAB. 6 – Average Edit distances.

	HMM+DM	HMM+GMM	HMM+GMM(Log,KLT)
WER (%)	11.75	35.14	35.62
FER (%)	12.21	33.30	45.71
Confusion matrices	$\begin{pmatrix} 2899 & 726 \\ 624 & 6808 \end{pmatrix}$	$\begin{pmatrix} 470 & 3155 \\ 527 & 6905 \end{pmatrix}$	$\begin{pmatrix} 1104 & 2521 \\ 2312 & 5120 \end{pmatrix}$

hidden state and which mixture component generate the current sample as the ground truth. HMM+DM and HMM+GMM models are trained on these data. Then we use Eq.1 to decode these sequences. Finally, WER, FER and confusion matrices are computed to show the performances of these two models.

Experimental results In Table 6, we list the experimental results. We can see in these generated data, the HMM+DM model achieves the best performances in both measures.

"HMM+GMM" in Table 6 is the model trained directly on the discrete-distribution data with full covariances. The mean vectors and covariance matrices are also initialized by K-means algorithm. π, B and C are randomly initialized as HMM+DM.

"HMM+GMM(Log,KLT)" in Table 6 is the model trained on the transformed data. We first take the log on the data to make them distributed more normally. Then KLT transform are applied on these data to decouple the correlations between components. Finally, we apply HMM+GMM with diagonal covariance to these data.

The HMM+DM model trained on these data are (after reorder)

$$\pi = \begin{pmatrix} 0.506 \\ 0.494 \end{pmatrix} \quad B = \begin{pmatrix} 0.048 & 0.952 \\ 0.469 & 0.531 \end{pmatrix} \quad C = \begin{pmatrix} 0.319 & 0.518 & 0.164 \\ 0.237 & 0.336 & 0.426 \end{pmatrix} \quad A = \left(\begin{array}{cccc} 5.819 & 4.677 & 9.443 & 5.028 \\ 1.011 & 7.183 & 8.020 & 10.187 \\ 8.678 & 8.401 & 2.874 & 8.977 \\ \hline 4.933 & 5.036 & 2.064 & 7.427 \\ 9.621 & 8.689 & 2.905 & 3.836 \\ 1.925 & 0.977 & 2.998 & 2.969 \end{array}\right)$$

which again coincide well with the real parameters except π (see the second column of the Table 4 for the real parameters).

4 Related work

Estimating single Dirichlet distribution dates back to Ronning's paper in 1989 [Ron89], where the Newton-Raphson method was used. Narayanan then gave the algorithm explicitly in [Nar91]. The Algorithm 1 in this report is essentially the same with Narayanan and Ronning's algorithm. Minka [Min03] gave the fixed-point iteration methods to estimation single Dirichlet.

The Dirichlet mixture model is proposed by Bouguila and et al. in [BZV04]. They use Newton method (or natural gradient descent method [iABNK+87]) to optimize the incomplete log-likelihood directly. We propose, instead, a EM framework for parameter estimation in this report (see Algorithm 4).

HMM+GMM model has been widely used in many fields, including speech processing [Ben96, RJ93] [...], computer vision[...] and etc. A good reference for the derivation of this model can be found at [Bil97]. To the best of our knowledge so far, there is no work on HMM with Dirichlet mixture emission.

5 Discussion and future work

As Dirichlet and Gaussian distributions all belong to exponential family, it is useful to generalize HMM+GMM and current HMM+DM to HMM with mixture of exponential family emission. For convenience, we call this generalization as HMM+EFM. From another view point, HMM+EFM can be viewed as the dynamic (or HMM) extension of the clustering methods with Bregman divergences [BMDG05]. This extension is under our current investigation.

One potential improvement of this algorithm will be the suitable initialization. Some heuristic or greedy methods could be devised here to improve avoid the problem coursed by permutation-invariant property of clustering methods used in Algorithm 2.

In their work on Dirichlet mixtures [BZV04], Bouguila and et al. proposed to use an entropy-based criterion for model selection. We will also investigate the model selection for HMM+DM model. And we will further relax the constraint that all Dirichlet mixtures associated with the Markov hidden states have the same number of Dirichlet components.

Finally, we will find some applications of the current model. Actually, HMM+DM can be applied to any scenarios where time-varying bag-of-word features are extracted. Before applying the HMM+DM model on these features, we should better to apply pLSA [Hof99] model to get lower dimensional features $p(z \mid d)$ (z is a latent aspect and d is a document or feature). The reasons are two folds : Firstly, according to our experiences, the computational costs are almost linear with the feature dimension. So dimension reduction will, especially in case of very large vocabulary, greatly improve the algorithm's efficiency while reducing the number of parameters. Secondly, as many bag-of-word features are very sparse, after applying pLSA model, we could get more compact representation of original data.

6 Acknowledgement

Le would like to express the his acknowledgement to Professor Ronning for some useful discussions and his hard-copy mail about his two papers [Ron89, Ron86]. Le would also like to thank Dr. Minka for helpful discussions on the curvature of the single Dirichlet distribution. Finally, many thanks go to Dong Zhang who pointed out the comparison protocol in Section 3.4.

A Derivations for inference procedure

Here, we list the detailed derivations of the inference procedure presented in Section 2.2.

Normalized Alpha Recursion Alpha recursion is a forward iteration. For numerical stability, we use the normalized alpha recursion as follows :

$$\begin{aligned}
\bar{\alpha}^{t}_{h_t m_t} &\triangleq p(h_t, m_t \mid x_0, .., x_t) \\
\bar{\alpha}^{t+1}_{h_{t+1} m_{t+1}} &= \frac{p(x_0, .., x_{t+1}, h_{t+1}, m_{t+1})}{p(x_0, .., x_t, x_{t+1})} \\
&= \frac{\sum_{h_t, m_t} p(x_0, .., x_{t+1}, h_t, m_t, h_{t+1}, m_{t+1})}{\sum_{h'_t, m'_t, h'_{t+1}, m'_{t+1}} p(x_0, ..., x_{t+1}, h'_t, m'_t, h'_{t+1}, m'_{t+1})} \\
&= \frac{\sum_{h_t, m_t} p(x_0, .., x_t, h_t, m_t) \cdot p(h_{t+1} \mid h_t) \cdot p(m_{t+1} \mid h_{t+1}) \cdot p(x_{t+1} \mid h_{t+1}, m_{t+1})}{\sum_{h'_t, m'_t, h'_{t+1}, m'_{t+1}} p(x_0, ..., x_t, h'_t, m'_t) \cdot p(h'_{t+1} \mid h'_t) \cdot p(m'_{t+1} \mid h'_{t+1}) \cdot p(x_{t+1} \mid h'_{t+1}, m'_{t+1})} \\
&= \frac{\sum_{h_t, m_t} p(h_t, m_t \mid x_0, .., x_t) \cdot p(h_{t+1} \mid h_t) \cdot p(m_{t+1} \mid h_{t+1}) \cdot p(x_{t+1} \mid h_{t+1}, m_{t+1})}{\sum_{h'_t, m'_t, h'_{t+1}, m'_{t+1}} p(h'_t, m'_t \mid x_0, ..., x_t) \cdot p(h'_{t+1} \mid h'_t) \cdot p(m'_{t+1} \mid h'_{t+1}) \cdot p(x_{t+1} \mid h'_{t+1}, m'_{t+1})} \\
&= \frac{\sum_{h_t, m_t} \bar{\alpha}^{t}_{h_t m_t} \cdot b_{h_t h_{t+1}} \cdot C_{h_{t+1} m_{t+1}} \cdot p(x_{t+1} \mid h_{t+1}, m_{t+1})}{\sum_{h'_t, m'_t, h'_{t+1}, m'_{t+1}} \bar{\alpha}^{t}_{h'_t m'_t} \cdot b_{h'_t h'_{t+1}} \cdot C_{h'_{t+1} m'_{t+1}} \cdot p(x_{t+1} \mid h'_{t+1}, m'_{t+1})}
\end{aligned}$$

Note, $\bar{\alpha}^{t}_{h_t m_t}$ is the filtered estimate of the state. The initial sates can be calculated as follows :

$$\begin{aligned}
\bar{\alpha}^{0}_{km} &= p(h_0 = k, m_0 = m \mid x_0) \\
&= \frac{p(x_0 \mid h_0 = k, m_0 = m) \cdot p(m_0 = m \mid h_0 = k) \cdot p(h_0 = k)}{\sum_{k'=1}^{K} \sum_{m'=1}^{M} p(x_0 \mid h_0 = k', m_0 = m') \cdot p(m_0 = m' \mid h_0 = k') \cdot p(h_0 = k')} \\
&= \frac{\pi_k \cdot C_{km} \cdot p(x_0 \mid h_0 = k, m_0 = m)}{\sum_{k'=1}^{K} \sum_{m'=1}^{M} \pi_{k'} \cdot C_{k'm'} \cdot p(x_0 \mid h_0 = k', m_0 = m')}
\end{aligned}$$

Beta Recursion Beta recursion is a backward iteration as

$$\begin{aligned}
\beta^{t}_{h_t m_t} &\triangleq p(x_{t+1}, .., x_T \mid h_t, m_t) \\
\beta^{t-1}_{h_{t-1} m_{t-1}} &= \sum_{h_t, m_t} b_{h_{t-1} h_t} \cdot C_{h_t m_t} \cdot p(x_t | h_t, m_t) \cdot \beta(h_t)
\end{aligned}$$

Gamma Recursion Gamma values are the smoothed estimates of the states, given the whole set.

$$\begin{aligned}
\gamma^{t}_{h_t m_t} &\triangleq p(h_t, m_t \mid x_0, .., x_T) \\
\gamma^{t}_{h_t m_t} &= \sum_{h_{t+1}, m_{t+1}} p(h_t, m_t, h_{t+1}, m_{t+1} \mid x_0, .., x_T) \\
&= \sum_{h_{t+1}, m_{t+1}} p(h_t, m_t \mid x_0, .., x_T, h_{t+1}, m_{t+1}) \cdot p(h_{t+1}, m_{t+1} \mid x_0, .., x_T) \\
&= \sum_{h_{t+1}, m_{t+1}} p(h_t, m_t \mid x_0, .., x_t, h_{t+1}, m_{t+1}) \cdot \gamma^{t+1}_{h_{t+1} m_{t+1}} \\
&= \sum_{h_{t+1}, m_{t+1}} \frac{p(h_t, m_t, h_{t+1}, m_{t+1} \mid x_0, .., x_t)}{p(h_{t+1}, m_{t+1} \mid x_0, .., x_t)} \cdot \gamma^{t+1}_{h_{t+1} m_{t+1}}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{h_{t+1},m_{t+1}} \frac{p(h_t,m_t \mid x_0,..,x_t)\cdot p(h_{t+1}\mid h_t)\cdot p(m_{t+1}\mid h_{t+1})}{\sum_{h'_t,m'_t} p(h'_t,m'_t,h_{t+1},m_{t+1}\mid x_0,..,x_t)}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
&= \sum_{h_{t+1},m_{t+1}} \frac{p(h_t,m_t \mid x_0,..,x_t)\cdot p(h_{t+1}\mid h_t)\cdot p(m_{t+1}\mid h_{t+1})}{\sum_{h'_t,m'_t} p(h'_t,m'_t\mid x_0,..,x_t)\cdot p(h_{t+1}\mid h'_t)\cdot p(m_{t+1}\mid h_{t+1})}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
&= \sum_{h_{t+1},m_{t+1}} \frac{\bar{\alpha}^t_{h_tm_t}\cdot b_{h_th_{t+1}}}{\sum_{h'_t,m'_t}\bar{\alpha}^t_{h'_tm'_t}\cdot b_{h'_th_{t+1}}}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
\text{(or)} \quad &= \sum_{h_{t+1},m_{t+1}} \frac{\alpha^t_{h_tm_t}\cdot b_{h_th_{t+1}}}{\sum_{h'_t,m'_t}\alpha^t_{h'_tm'_t}\cdot b_{h'_th_{t+1}}}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}}
\end{aligned}
$$

The initial state can be initialized as

$$\gamma^T_{km} = \bar{\alpha}^T_{km} \quad .$$

The $\eta^t_{h_t}$ variables We need the following variables as the separators in the junction tree :

$$
\begin{aligned}
\eta^t_{h_t} &= p(h_t \mid x_0,...,x_T) \\
&= \sum_{m=1}^{M} p(h_t, m_t = m \mid x_0,...,x_T) \\
&= \sum_{m=1}^{M} \gamma^t_{h_tm}
\end{aligned}
$$

The $\xi^t_{h_th_{t+1}}$ variables We need also the following variables :

$$
\begin{aligned}
\xi^t_{h_th_{t+1}} &\triangleq p(h_t,h_{t+1}\mid x_0,..,x_T) \\
&= \sum_{m_t,m_{t+1}} p(h_t,m_t,h_{t+1},m_{t+1}\mid x_0,...,x_T) \\
&= \sum_{m_t,m_{t+1}} p(h_t,m_t\mid x_0,..,x_T,h_{t+1},m_{t+1})\cdot p(h_{t+1},m_{t+1}\mid x_0,..,x_T) \\
&= \sum_{m_t,m_{t+1}} p(h_t,m_t\mid x_0,..,x_t,h_{t+1},m_{t+1})\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
&= \sum_{m_t,m_{t+1}} \frac{p(h_t,m_t,h_{t+1},m_{t+1}\mid x_0,..,x_t)}{p(h_{t+1},m_{t+1}\mid x_0,..,x_t)}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
&= \sum_{m_t,m_{t+1}} \frac{p(h_t,m_t \mid x_0,..,x_t)\cdot p(h_{t+1}\mid h_t)\cdot p(m_{t+1}\mid h_{t+1})}{\sum_{h'_t,m'_t} p(h_{t+1},m_{t+1},h'_t,m'_t\mid x_0,..,x_t)}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
&= \sum_{m_t,m_{t+1}} \frac{\bar{\alpha}^t_{h_tm_t}\cdot b_{h_th_{t+1}}\cdot p(m_{t+1}\mid h_{t+1})}{\sum_{h'_t,m'_t} p(h'_t,m'_t\mid x_0,..,x_t)\cdot p(h_{t+1}\mid h'_t)\cdot p(m_{t+1}\mid h_{t+1})}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
&= \sum_{m_t,m_{t+1}} \frac{\bar{\alpha}^t_{h_tm_t}\cdot b_{h_th_{t+1}}}{\sum_{h'_t,m'_t}\bar{\alpha}^t_{h'_tm'_t}\cdot b_{h'_th_{t+1}}}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}} \\
\text{or} \quad &= \sum_{m_t,m_{t+1}} \frac{\alpha^t_{h_tm_t}\cdot b_{h_th_{t+1}}}{\sum_{h'_t,m'_t}\alpha^t_{h'_tm'_t}\cdot b_{h'_th_{t+1}}}\cdot \gamma^{t+1}_{h_{t+1}m_{t+1}}
\end{aligned}
$$

B Derivations for parameter estimation

In this section, we list some derivation involved in Section 2.3.

B.1 Inverse of Hessian matrix in Eq.13

The inverse of the Hessian can be easily computed by the its special structure. By using the following fact :

$$(I \pm XX^T)^{-1} = I \mp X(I \pm X^T X)^{-1} X^T \quad ,$$

which is one special case of the matrix inverse lemma. we have

$$\begin{aligned}
\mathrm{H}^{-1} &= -(\Lambda - z\mathrm{1}\mathrm{1}^{\mathrm{T}})^{-1} \\
&= -\Lambda^{-\frac{1}{2}}(\mathrm{I} - (\sqrt{z}\Lambda^{-\frac{1}{2}}\mathrm{1})(\sqrt{z}\Lambda^{-\frac{1}{2}}\mathrm{1})^T)^{-1}\Lambda^{-\frac{1}{2}} \\
&= -\Lambda^{-1} - \frac{1}{z^{-1} - \sum_{n=1}^{N}(\Lambda^{nn})^{-1}}\Lambda^{-1}\mathrm{1}\mathrm{1}^{\mathrm{T}}\Lambda^{-1}
\end{aligned}$$

B.2 Energy term

The complete log likelihood is

$$\begin{aligned}
\log p(\mathcal{X}, \mathcal{H}) &= \sum_{d=1}^{D} \log p(X^d, H^d) \\
&= \sum_{d=1}^{D} \log \left\{ p(h_0)p(m_0 \mid h_0)p(x_0^d \mid h_0, m_0) \prod_{t=0}^{T_d-1} \left[p(h_{t+1} \mid h_t)p(m_{t+1} \mid h_{t+1})p(x_{t+1}^d \mid h_{t+1}, m_{t+1}) \right] \right\} \\
&= \sum_{d=1}^{D} \log \left\{ p(h_0) \left(\prod_{t=0}^{T_d} p(m_t \mid h_t) \right) \left(\prod_{t=0}^{T_d-1} p(h_{t+1} \mid h_t) \right) \left(\prod_{t=0}^{T_d} p(x_t^d \mid h_t, m_t) \right) \right\} \\
&= \sum_{d=1}^{D} \log \left\{ \pi_{h_0} \left(\prod_{t=0}^{T_d} C_{h_t m_t} \right) \left(\prod_{t=0}^{T_d-1} b_{h_t h_{t+1}} \right) \left(\prod_{t=0}^{T_d} \mathrm{Dir}\left(x_t^d \mid a_{h_t, m_t \cdot}\right) \right) \right\} \\
&= \sum_{d=1}^{D} \log \left\{ \pi_{h_0} \left(\prod_{t=0}^{T_d} C_{h_t m_t} \right) \left(\prod_{t=0}^{T_d-1} b_{h_t h_{t+1}} \right) \left(\prod_{t=0}^{T_d} \left[\frac{\Gamma(\sum_n a_{h_t, m_t n})}{\prod_n \Gamma(a_{h_t, m_t n})} \prod_{n=1}^{N} (x_{tn}^d)^{a_{h_t, m_t n} - 1} \right] \right) \right\} \\
&= \sum_{d=1}^{D} \log \pi_{h_0} + \sum_{d=1}^{D} \sum_{t=0}^{T_d} \log C_{h_t m_t} + \sum_{d=1}^{D} \sum_{t=0}^{T_d-1} \log b_{h_t h_{t+1}} + \sum_{d=1}^{D} \sum_{t=0}^{T_d} \log \Gamma(\sum_{n=1}^{N} a_{h_t, m_t n}) \\
&\quad - \sum_{d=1}^{D} \sum_{t=0}^{T_d} \sum_{n=1}^{N} \log \Gamma(a_{h_t, m_t n}) + \sum_{d=1}^{D} \sum_{t=0}^{T_d} \sum_{n=1}^{N} (a_{h_t, m_t n} - 1) \log x_{tn}^d
\end{aligned}$$

where for clarity, we omit the super-scripture of d in h_i^d and m_i^d.

Then the energy term is

$$\begin{aligned}
\langle \log p(\mathcal{X}, \mathcal{H}) \rangle_{q(\mathcal{H})} = &\sum_{d=1}^{D} \left\langle \log \pi_{h_0^d} \right\rangle_{q(h_0^d)} &= \sum_{d=1}^{D} \sum_{k=1}^{K} \sum_{m=1}^{M} \gamma_{km}^{0,d} \log \pi_k \\
&+ \sum_{d=1}^{D} \sum_{t=0}^{T_d} \langle \log C_{h_t m_t} \rangle_{q(h_t^d, m_t^d)} &+ \sum_{d=1}^{D} \sum_{t=0}^{T_d} \sum_{k=1}^{K} \sum_{m=1}^{M} \gamma_{km}^{t,d} \log C_{km} \\
&+ \sum_{d=1}^{D} \sum_{t=0}^{T_d-1} \left\langle \log b_{h_t h_{t+1}} \right\rangle_{q(h_t^d, h_{t+1}^d)} &+ \sum_{d=1}^{D} \sum_{i,j=1}^{K} \sum_{t=0}^{T_d-1} \xi_{ij}^{t,d} \log b_{ij}
\end{aligned}$$

$$
\begin{array}{ll}
+\sum_{d=1}^{D}\sum_{t=0}^{T_d}\left\langle \log\Gamma(\sum_{n=1}^{N} a_{h_t,m_t n})\right\rangle_{q(h_t^d,m_t^d)} & +\sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{km}^{t,d}\log\Gamma(\sum_{n=1}^{N} a_{k,mn}) \\
-\sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{n=1}^{N}\left\langle \log\Gamma(a_{h_t,m_t n})\right\rangle_{q(h_t^d,m_t^d)} & -\sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{km}^{t,d}\log\Gamma(a_{k,mn}) \\
+\sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{n=1}^{N}\left\langle (a_{h_t,m_t n}-1)\log x_{tn}^d\right\rangle_{q(h_t^d,m_t^d)} & +\sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{km}^{t,d}(a_{k,mn}-1)\log x_{tn}^d
\end{array}
$$

So by the summation convention, we get Eq.2.

B.3 Entropy term

Then entropy term can be calculated as

$$
\begin{aligned}
Etr(\mathcal{H}) &= -\left\langle \sum_{d=1}^{D}\log q(H^d)\right\rangle_{q(\mathcal{H})} \\
&= -\sum_{d=1}^{D}\left\langle \log\left(q(h_0^d,m_0^d)\prod_{t=1}^{T_d}\frac{q(h_{t-1}^d,h_t^d)q(h_t^d,m_t^d)}{q(h_{t-1}^d)q(h_t^d)}\right)\right\rangle_{q(H^d)} \\
&= \sum_{d=1}^{D}\left\langle \sum_{t=0}^{T_d}2\log q(h_t^d)-\log q(h_0^d)-\log q(h_{T_d}^d)-\sum_{t=0}^{T_d}\log q(h_t^d,m_t^d)-\sum_{t=1}^{T_d}\log q(h_{t-1}^d,h_t^d)\right\rangle_{q(H^d)} \\
&= \sum_{d=1}^{D}\sum_{t=0}^{T_d}2\left\langle \log q(h_t^d)\right\rangle_{q(h_t^d)}-\left\langle \log q(h_0^d)\right\rangle_{q(h_0^d)}-\left\langle \log q(h_{T_d}^d)\right\rangle_{q(h_{T_d}^d)} \\
&\quad -\sum_{d=1}^{D}\sum_{t=0}^{T_d}\left\langle \log q(h_t^d,m_t^d)\right\rangle_{q(h_t^d,m_t^d)}-\sum_{d=1}^{D}\sum_{t=1}^{T_d}\left\langle \log q(h_{t-1}^d,h_t^d)\right\rangle_{q(h_{t-1}^d,h_t^d)} \\
&= \sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{k=1}^{K}2\eta_k^{t,d}\log\eta_k^{t,d}-\sum_{d=1}^{D}\sum_{k=1}^{K}\eta_k^{0,d}-\sum_{d=1}^{D}\sum_{k=1}^{K}\eta_k^{T_d,d} \\
&\quad -\sum_{d=1}^{D}\sum_{t=0}^{T_d}\sum_{k=1}^{K}\sum_{m=1}^{M}\gamma_{km}^{t,d}\log\gamma_{km}^{t,d}-\sum_{d=1}^{D}\sum_{t=0}^{T_d-1}\sum_{i,j=1}^{K}\xi_{ij}^{t,d}\log\xi_{ij}^{t,d} \\
\text{or} &= 2\eta_{\mathtt{k}}^{\mathtt{t,d}}\log\eta_{\mathtt{k}}^{\mathtt{t,d}}-\eta_{\mathtt{k}}^{0,\mathtt{d}}\log\eta_{\mathtt{k}}^{0,\mathtt{d}}-\eta_{\mathtt{k}}^{T_d,\mathtt{d}}\log\eta_{\mathtt{k}}^{T_d,\mathtt{d}}-\gamma_{\mathtt{km}}^{\mathtt{t,d}}\log\gamma_{\mathtt{km}}^{\mathtt{t,d}}-\xi_{\mathtt{ij}}^{\mathtt{t,d}}\log\xi_{\mathtt{ij}}^{\mathtt{t,d}} \quad \text{(by summation convention)}
\end{aligned}
$$

d Newton method with nonnegative constraints . In previous part, we
in to impose the non-negative constraints by

B.4 Problem with Bouguila's parametrization

In Section 2.3.1, we do not treat the nonnegative constraints on the Dirichlet parameters $a_n > 0$. Bouguila and et al. [BZV04] propose to use the following change of variable to impose the nonnegative constraints explicitly :

$$a_n = e^{b_n} \quad .$$

Then the problem is

$$L(b) = \log\Gamma(\sum_{n=1}^{N}\exp(b_n)) - \log\Gamma(\exp(b_{\mathtt{n}})) + \exp(b_{\mathtt{n}})\cdot\log\bar{x}_{\mathtt{n}} \tag{23}$$

for any $b_n \in \mathbb{R}$.

$$\begin{aligned}
\frac{\partial L(b)}{\partial b_n} =& \exp(b_n) \cdot \Psi_0(\sum_{n=1}^{N} \exp(b_n)) - \exp(b_n) \cdot \Psi_0(\exp(b_n)) + \exp(b_n) \cdot \log \bar{x}_n \\
\frac{\partial^2 L(b)}{\partial b_n^2} =& \exp(b_n) \cdot \Psi_0(\sum_{n=1}^{N} \exp(b_n)) - \exp(b_n) \cdot \Psi_0(\exp(b_n)) \\
&+ \exp(2b_n) \cdot \Psi_1(\sum_{n=1}^{N} \exp(b_n)) - \exp(2b_n) \cdot \Psi_1(\exp(b_n)) \\
&+ \exp(b_n) \cdot \log \bar{x}_n \\
\frac{\partial^2 L(b)}{\partial b_n b_{n'}} =& \exp(b_n + b_{n'}) \cdot \Psi_1(\sum_{n=1}^{N} \exp(b_n)) \qquad n \neq n'
\end{aligned}$$

So the Hessian is

$$\begin{aligned}
\mathrm{H} =& zrr^T + \Lambda \\
\text{where} \quad z =& \Psi_1(\sum_{n=1}^{N} \exp(b_n)) \\
r =& (e^{b_1}, \cdots, e^{b_N})^T \\
\Lambda_{nn} =& -\exp(2b_n) \cdot \Psi_1(\exp(b_n)) \\
&+ \exp(b_n) \cdot \Psi_0(\sum_{n=1}^{N} \exp(b_n)) - \exp(b_n) \cdot \Psi_0(\exp(b_n)) \\
&+ \exp(b_n) \cdot \log \bar{x}_n
\end{aligned}$$

The problem for this method is that after changing of variables, the objective function is no longer convex, which can be easily verified by some numerical trials. For example, if we generate 6 samples on 2-dimensional probabilistic simplex :

$$\begin{pmatrix} 0.468 & 0.456 & 0.657 & 0.168 & 0.407 & 0.250 \\ 0.258 & 0.100 & 0.305 & 0.404 & 0.319 & 0.419 \\ 0.275 & 0.444 & 0.038 & 0.428 & 0.274 & 0.331 \end{pmatrix}$$

Then $\log \bar{x}_n = \begin{pmatrix} -1.005 & -1.294 & -1.437 \end{pmatrix}^T$. For a point $a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}^T$, or equivalently $b = \begin{pmatrix} 0.000 & 0.693 & 1.099 \end{pmatrix}^T$, we can calculate that

$$H = \begin{pmatrix} -0.185 & 0.363 & 0.544 \\ 0.363 & -1.875 & 1.088 \\ 0.544 & 1.088 & -3.884 \end{pmatrix}$$

with eigenvalues : $\begin{pmatrix} -4.391 & -1.606 & 0.052 \end{pmatrix}$.

So we prefer to use the original parametrization.

C Code list

In this section, we list some core procedures of this report written by the authors. You can find all the sourse codes in http://www.idiap.ch/~cle/papers/resources/SourseCodes_for_HMMDM.tar.gz.

Generating samples from single Dirichlet distribution :

```
function Data = GenDir(a,n)
% Generat samples from single Dirichlet distribution.
% Input:
%   a: M-by-1 vector. Dirichlet parameter
%   n: Number of samples.
% Output:
%   Data: M-by-N matrix with each column being one sample.
% Note: This function is adapted from Minka's function: dirichelt_sample.m
```

Generating samples from a mixture of Dirichlet distributions :

```
function Data = GenMixtureDir(A,Pi,Number)
% Generate samples from Dirichlet mixture model.
% Input:
%   A:  M-by-N matrix. Parameters for Mixture of Dirichlet
%       each row is one Dirichlet.
%       N: Data dimension
%       M: number of mixture
%   Number: Number of samples to generate.
%   Pi: M-by-1 vector. Prior distribution for each Dirichlet.
% Output:
%   Data: N-by-Number matrix with each column being one sample.
```

Initializing the mixture of Dirichlet (Algorithm 2) :

```
function [A, Pi] = MomentMatchingInitDM(M,DataSet)
% Initialize the mixture of Dirichlet by Kmeans + Moment Matching.
% Input:
%   M:  Number of mixture components (M >=1).
%   DataSet: N-by-T sample matrix with
%            N is the data dimension
%            T is the number of samples.
% Output:
%   A:  M-by-N matrix, with each row corresponding one mixture component.
%   Pi: M-by-1 probability vector.
% Note: use the kmeans.m in the statistics toolbox
```

Estimating single Dirichlet distribution in Algorithm 1 :

```
function aNew = EstDirchlet(a,nx)
% Estimating single Dirichlet distribution by Newton method.
% Input:
%     a:  Initial parameter for the Dirichlet distr. (Column vector)
%     nx: Mean of log samples.
% Output:
%     aNew: a column vector corresponding to Dirichlet parameters.
```

"`Script_Test_SingleDirichlet.m`" is a demo and test script for this function.

Evaluate the data's log-likelihoods relative to a Dirichlet model :

```
function p = Dirichlet_loglike(a, data)
% Evaluate the data's log-likelihoods relative to a Dirichlet model.
% Input:
%   a:    a N-by-1 column vector, Dirichlet parameter.
%   data: a N-by-T matrix, with each column being one sample (sum to one).
% Output:
%   p: a 1-by-T row vector for log likelihoods.
% Note: This procedure is adapted from Minka's dirichlet_logProb.m
```

Calculating the samples' likelihood for given Dirichlet model (either mixture or not) :

```
function obslik = dataLikelihood_DM(A,data,isLog)
% Calculate the data likelihood for the Dirichlet mixture.
% Input:
%     A: M-by-N-by-K matrix, parameters of DM.
%        M*K is number of mixture components.
%        N is sample dimension.
%        When K =1, A is a matrix; When K=1 and M=1, it is a single Dir.
%     data: N-by-T matrix. T is the sample number.
%     isLog: 0 - output likelihood (Default), otherwise log likelihood.
% Output:
%     obslik: T-by-M-by-K matrix.
%             obslik(t,m,k) is the t^th sample's (log-)likelihood on
%             (m,k)^th mixture components.
```

Estimating the parameters of the mixture of Dirichlet by EM algorithm (Algorithm 4)

```
function [A, Pi] = EstMixDirichlet(Data, M)
% Estimate the parameters of the mixture of Dirichlet by EM algorithm.
% Input:
%   Data: N-by-T data matrix. N is data dimension; T is number of samples.
%   M: number of mixture components.
% Output:
%   A:  M-by-N matrix, with each row corresponding to one Dirichlet.
%   Pi: M-by-1 probability vector.
```

Calculate the Entropy

```
function E = entropy_base_e(Distr)
% Calculate the Entropy using log based on e, instead 2.
% Input:
%   Distr: a matrix, with each row is a distribution.
% Output:
%   E: a column vector, corresponding to each the entropy of each distr.
```

Inference procedure for general HMM + Mixture of density model (Algorithm 5) :

```
% function [Gm,Xi] = forback(B,C,Pi,obslik)
% Inference procedure for general HMM + Mixture of density model.
% Input:
%     B: K-by-K probability transition matrix.
```

```
%        K is the number of hidden states.
%        B(i,j) = p(h=j|h=i). Then each row of B should sum to 1.
%     C: K-by-M probability matrix.
%        M is the number of mixture densities.
%        C(i,j) = p(m=j|h=i). Then each row of C should sum to 1.
%     Pi: K-by-1 column vector, initial probability of hidden states.
%     obslik: T-by-M-by-K likelihood (not log likelihood) arrays.
%             T is the sample number.
%             obslik(t,m,k) is t^th samples likelihood on (m,k)^th component.
% Output: the smoothed states
%     Gm: T-by-M-by-K matrix. Gm(t,m,k) = p(h_t = k,m_t = m |X_1,...,X_T),
%     Xi: (T-1)-by-K-by-K matrix. Xi(t,k1,k2)=p(h_t=k1,h_{t+1}=k2 |X_1,...,X_T),
%          t=1,...,T-1.
```

Generating one sequence of random samples from HMM+DM model :

```
function Data = GenDynamicMixtureDir(A,B,C,Pi,Number)
% Generate one sequence of random samples from HMM+DM model.
% Input:
%     A: M-by-N-by-K positive array.
%        M is number of Dirichlet components.
%        N is sample's dimension.
%        K is the number of hidden states.
%        A(m,:,k) is the Dirichlet corresponding to (m,k)^th Dirichlet.
%     B: K-by-K probability transition matrix.
%        B(i,j) = p(h=j|h=i). Then each row of B should sum to 1.
%     C: K-by-M probability matrix.
%        C(i,j) = p(m=j|h=i). Then each row of C should sum to 1.
%     Pi: K-by-1 column vector, initial probability of hidden states.
%     Number: Number of samples to generate.
%             T is the sample number.
%             obslik(t,m,k) is t^th samples likelihood on (m,k)^th component.
% Output:
%   Data: N-by-Number matrix with each column is one sample.
%   Ind:  2-by-Number matrix, with Ind(1,t) in {1,...,K} denoting which
%         hidden state sample t belongs. And Ind(2,t) in {1,...,M}
%         denoting which Dirichlet generating sample t.
```

Estimating parameters of HMM+DM by EM algorithm (Algorithm 3) :

```
function [A, B, C, Pi] = EstHMMDM(Data, K, M)
% Estimate parameters of HMM+DM by EM algorithm.
% Input:
%   Data: D-by-1 cell, with Data{d} is N-by-Td data matrix.
%         D is the number of sequences;
%         N is data dimension;
%         Td is number of samples in d^th sequence.
%   K: number of hidden states.
%   M: number of mixture components.
% Output:
%     A: M-by-N-by-K positive array.
%        A(m,:,k) is the Dirichlet corresponding to (m,k)^th Dirichlet.
```

```
%     B: K-by-K probability transition matrix.
%        B(i,j) = p(h=j|h=i). Then each row of B should sum to 1.
%     C: K-by-M probability matrix.
%        C(i,j) = p(m=j|h=i). Then each row of C should sum to 1.
%     Pi: K-by-1 column vector, initial probability of hidden states.
```

Calculating the hard clustering results from the smoothed posteriors :

```
function Ind = HardClusterGamma(Gm)
% Calculate the hard clustering results from the smoothed posteriors Gamma.
% Input:
%   Gm: the smoothed posteriors Gamma. 1-by-D cell.
%       Gm{d} is a Td-by-M-by-K array:
%         D is sequence number
%         M is the number of hidden states for Dirichlet mixture
%         K is the number of hidden states for h
%         Td is sample number of the d^th sequence
% Output:
%   Ind: 1-by-D cell.
%       Ind{d} is a 2-by-Td matrix. The t^th sample's is generated
%       by (h=Ind{d}(1,t),m=Ind{d}(2,t))^th Dirichlet.
```

Algorithm 5: General inference procedure for HMM + Mixture density model

Input:
- Model parameters $B \in \mathbb{R}_+^{K\times K}$, $C \in \mathbb{R}_+^{K\times M}$ and $\pi \in \mathbb{R}_+^{K\times 1}$, where
  - K is the number of hidden states.
  - M is the number of mixture components.
- Samples likelihood $L \in \mathbb{R}_+^{T\times M\times K}$, where T is the sample number. L_{mk}^t is the t^{th} sample's likelihood for $(m,k)^{th}$ mixture component.

Result:
- The smoothed probability $\{\gamma_k^t \mid t = 0, \cdots, T; k = 1, \cdots, K\}$
- The joint probability of consecutive two states $\{\xi_{ij}^t \mid t = 0, \cdots, T-1; i, j = 1, \cdots, K\}$

begin
  `// Forward pass`
  for $t = 0$ **to** T **do**
    $s \leftarrow 0$;
    for $k = 1$ **to** K **do**
      for $m = 1$ **to** M **do**
        if $t = 0$ **then**
          $v \leftarrow \pi_k \cdot C_{km} \cdot L_{mk}^t$;
        else
          $v \leftarrow \sum_{k'=1}^{K} \sum_{m'=1}^{M} \bar{\alpha}_{k'm'}^{t-1} \cdot b_{k'k} \cdot C_{km} \cdot L_{km}^t$;
        end
        $\bar{\alpha}_{km}^t \leftarrow v$;
        $s \leftarrow s + v$;
      end
    end
    for $k = 1$ **to** K **do**
      for $m = 1$ **to** M **do**
        $\bar{\alpha}_{km}^t \leftarrow \bar{\alpha}_{km}^t / s$;
      end
    end
  end
  `// Backward pass`
  for $t = T$ **to** 0 **do**
    if $t \neq T$ **then**
      for $k = 1$ **to** K **do**
        $c_k \leftarrow \sum_{i=1}^{K} \sum_{m=1}^{M} \bar{\alpha}_{im}^t \cdot b_{ik}$;
      end
    end
    for $k = 1$ **to** K **do**
      for $m = 1$ **to** M **do**
        if $t = T$ **then**
          $\gamma_{km}^t \leftarrow \bar{\alpha}_k^t$;
        else
          $\gamma_{km}^t \leftarrow \sum_{k'=1}^{K} \sum_{m'=1}^{M} \frac{\bar{\alpha}_{km}^t \cdot b_{kk'}}{c_{k'}} \cdot \gamma_{k'm'}^{t+1}$;
          $\xi_{kk'}^t \leftarrow \sum_{m=1}^{M} \sum_{m'=1}^{M} \frac{\bar{\alpha}_{km}^t \cdot b_{kk'}}{c_{k'}} \cdot \gamma_{k'm'}^{t+1}$;
        end
      end
    end
  end
  return $\{\gamma_{km}^t, \xi_{kk'}^{t'} \mid t = 0, \cdots, T; t' = 0, \cdots, T-1; k, k' = 1, \cdots, K; m = 1, \cdots, M\}$
end

Références

[Aic82] J. Aichison. The statistical analysis of compositional data. *journal of the Royal Statistical Society*, Ser. B(44) :139–177, 1982. 3

[AS64] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, New York, 1964. 5

[Ben96] Yoshua Bengio. Markovian models for sequential data. Technical Report 1049, Dept. IRO, Universite de Montreal, 1996. 20

[Bil97] Jeff Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997. 20

[BMDG05] Arindam Banerjee, Srujana Merugu, Inderjit Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6 :1–48, 2005. 21

[BW90] H. Bourlard and C.J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12) :1167–1178, December 1990. 2

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman Publishing Co. Inc, 1999. 2

[BZV04] Nizar Bouguila, Djemel Ziou, and Jean Vaillancourt. Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. *IEEE Transaction on Image processing*, 13(11) :1533–1543, 2004. 3, 11, 20, 21, 25

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd ed.)*. Wiley Interscience, 2001. 3

[DLR77] A. P. Dempster, N. M. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B(39) :1–38, 1977. 3

[HES00] Hynek Hermansky, Daniel P.W. Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proceedings of Acoustics, Speech, and Signal Processing*, volume 3, pages 1635–1638, 2000. 2, 3

[Hof99] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 289–29, San Francisco, CA, 1999. Morgan Kaufmann. 2, 21

[iABNK+87] Shun ichi Amari, Ole E. Barndorff-Nielsen, Robert E. Kass, Steffen Lauritzen, and Calyampudi R. Rao. *Differential Geometry in Statistical Inference*, chapter 2, pages 19–94. Hayward, California : Institute of Mathematical Statistics, 1987. 20

[Min03] Thomas P. Minka. Estimating a Dirichlet distribution. Technical report, Carnegie Mellon University, 2003. 20

[Nar91] A. Narayanan. Algorithm as 266 : Maximum likelihood estimation of the parameters of the Dirichlet distribution. *Applied Statistics*, 40(2) :365–374, 1991. 20

[QMO+05] Pedro Quelhas, Florent Monay, Jean-Marc Odobez, Daniel Gatica-Perez, Tinne Tuytelaars, and L. Van-Gool. Modeling scenes with local descriptors and latent aspects. In *Proceedings of the 10th International Conference on Computer Vision*, volume 1, pages 883 – 890, 2005. 2

[RJ93] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey : Prentice Hall, 1993. 20

[Ron86] Gerd Ronning. On the curvature of the trigamma function. *Journal of Computational and Applied Mathematics*, 15 :397–399, 1986. 8, 21

[Ron89] Gerd Ronning. Maximum-likelihood estimation of Dirichlet distribution. *Journal of Statistical Computation and Simulation*, 32 :215–221, 1989. 3, 8, 9, 11, 20, 21

[SZ03] Josef Sivic and Andrew Zisserman. Video Google : A text retrieval approach to object matching in videos. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, page 1460, 2003. 2

[ZGPBM06] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Modeling individual and group actions in meetings with layered HMMs. *IEEE Transactions on Multimedia*, 8 :509– 520, 2006. 19