HAL
archives-ouvertes.fr

Simultaneous Clustering and Model Selection for Multinomial Distribution: A Comparative Study

Md Abul Hasnat, Julien Velcin, Stéphane Bonnevay, Julien Jacques

▶ To cite this version:

Md Abul Hasnat, Julien Velcin, Stéphane Bonnevay, Julien Jacques. Simultaneous Clustering and Model Selection for Multinomial Distribution: A Comparative Study. Intelligent Data Analysis, Oct 2015, Saint Etienne, France. hal-01203561

HAL Id: hal-01203561

https://hal.inria.fr/hal-01203561

Submitted on 24 Sep 2015

Simultaneous Clustering and Model Selection for Multinomial Distribution: A Comparative Study

Md. Abul Hasnat, Julien Velcin, Stéphane Bonnevay, and Julien Jacques

Laboratoire ERIC, Université Lumière Lyon 2, Lyon, France
md.hasnat@univ-lyon2.fr, julien.velcin@univ-lyon2.fr,
stephane.bonnevay@univ-lyon1.fr, julien.jacques@univ-lyon2.fr

Abstract. In this paper, we study different discrete data clustering methods, which use the Model-Based Clustering (MBC) framework with the Multinomial distribution. Our study comprises several relevant issues, such as initialization, model estimation and model selection. Additionally, we propose a novel MBC method by efficiently combining the partitional and hierarchical clustering techniques. We conduct experiments on both synthetic and real data and evaluate the methods using accuracy, stability and computation time. Our study identifies appropriate strategies to be used for discrete data analysis with the MBC methods. Moreover, our proposed method is very competitive w.r.t. clustering accuracy and better w.r.t. stability and computation time.

Keywords: Multinomial Distribution, Model-Based Clustering.

1 Introduction

Model-Based Clustering (MBC) estimates the parameters of a statistical model for the data and produces probabilistic clustering [6, 7, 15, 19]. To use the MBC method for clustering data as well as automatically selecting K (number of clusters), it is necessary to generate a set of candidate models. A simple approach to generate these models is to separately estimate them using an Expectation-Maximization (EM) method [13] with $K = 1, \ldots, K_{max}$. However, it can be computationally inefficient for higher dimensional data and higher K_{max} value.

Figueiredo and Jain [5] proposed a MBC method that integrates both model estimation and selection task within a single EM algorithm. A different strategy, called hybrid MBC [19], generates a hierarchy of models from K_{max} clusters by merging the parameters. Indeed, such an approach naturally saves computation time as it does not explicitly learn $K = K_{max} - 1, \ldots 1$ components models from the data. In this paper, we propose a hybrid MBC method with the Multinomial Mixture (MM) model and then empirically compare it with other MBC methods. Moreover, we explicitly addresses two related issues: (1) initialization [3]: how to set the initial parameters for the EM method and (2) model selection [2]: which criterion to use for selecting the best model. Therefore, based on an empirical study, we aim to answer the following questions: (a) which method should be used for initialization? (b) how to efficiently generate a set of models? (c) what

is the difference among "learning from data" and "estimating from K_{max} model parameters"? and (d) what is the best model selection method?

Our overall contribution is to perform a comparative study among different MBC methods with the MM. Individually, we: (1) propose (Sec. 3.6) a novel MBC method and compare it with the state-of-the-art methods; (2) perform empirical study on different initialization methods (Sec. 3.3) and (3) compare different model selection methods (Sec. 3.5). We conduct experiments with synthetic and real text data (for document clustering [19]) and identify particular methods that should be used for initialization, candidate models estimation and model selection. Therefore, the above contributions and experiments will naturally answer the questions raised at the end of the previous paragraph.

In the remaining part of this paper, we study the background and related work in Sec. 2, discuss different methods in Sec. 3, present the experimental results with discussion in Sec. 4 and finally draw conclusions in Sec. 5.

2   Background and Related Work

Model-Based Clustering (MBC) [6, 15] is a well-established method for cluster analysis and unsupervised learning. MBC assumes a probabilistic model (e.g., mixture model) for the data and then estimates the model parameters by optimizing an objective function (e.g., model likelihood). The Expectation Maximization (EM) [13] is mostly used in MBC to estimate the model parameters. EM consists of an Expectation step (E-step) and a Maximization step (M-step) which are iteratively employed to maximize the log likelihood of the data.

MBC methods have been exploited with the Gaussian distribution to analyze continuous data [6, 15, 5, 2, 7]. Besides, they have been proposed to analyze discrete data using the Multinomial distribution [14, 17] and directional data using the directional distributions [1, 9, 10]. In this paper, we only study and compare the MBC methods with the Multinomial distribution.

The Multinomial Mixture (MM) is a statistical model which has been used for cluster analysis with discrete data [14, 20, 17]. Meilă and Heckerman [14] studied the MBC methods with MM and compared them w.r.t. accuracy, time and number of clusters. They found that the EM method significantly outperforms others, which motivates us to solely focus on the EM related approaches.

Initialization of the EM method has significant impact on the clustering results [13, 3, 12], because with different initializations it may converge to different values of the likelihood function, some of which can be local maxima, i.e., sub-optimal results. To overcome this, several initialization strategies have been proposed, see [3] for details. Meilă and Heckerman [14] investigated three initialization strategies for the EM with MM. In this paper, we consider their [14] observations as well as empirically evaluate additional initialization methods for the EM method which were discussed by Biernacki et al. [3].

In order to automatically select K (number of components), MBC method can be used by first generating a set of candidate models with different values of K and then selecting the optimal model using a model selection criterion [6, 15].

This strategy needs to address two issues: (a) how to generate the models? and (b) how to select the best model? This paper considers both of these issues. Particularly, we focus on the candidate models generation task and propose a novel solution based on the Hybrid MBC (HMBC) [19] method.

HMBC method is a two-staged model that exploits both partitional and hierarchical clustering. It begins with a partitional clustering with K_{max} clusters and then use the Hierarchical Agglomerative Clustering (HAC) on those cluster parameters to generate a hierarchy of mixture models. It has differences with the Model-Based Hierarchical Clustering (MBHC) which employs the HAC on each data point [6]. In practice, for a large number of samples, such MBHC method is inefficient w.r.t. the required time and memory [19]. Several HMBC methods have been proposed with different probability distributions, see [19], [8], [10] and [18]. Among these, [18] proposed a method in the context of Bayesian analysis. However, it requires an explicit analysis of the features, which can be computationally inefficient for higher dimensional data. An efficient mixture model simplification/fusion method is recently proposed in [8] for the Gaussian distribution and in [10, 9] for the directional distributions. They use information divergences among the mixture models. In this paper, we follow a similar approach and propose a novel HMBC method with the MM.

Model selection is one of the most prominent issues in cluster analysis [15, 5, 2, 7]. In general, a statistical model selection criterion is often used with the MBC method, which is also called the parsimony-based approach [15]. See [5] for a list of different criteria. A different approach performs model selection by analyzing an evaluation graph, see [16] for such a method called the L-method. To select model with MM, [14] uses the likelihood value. Recently, [17] proposed the Minimum Message Length (MML) criterion for the MM. In this paper, we aim to present a comparative study among these methods.

This paper has similarity with two previous work [14] and [17]. However, the key differences are: (1) it proposes a novel method to efficiently generate candidate models; (2) investigate additional initialization methods proposed in [3] and (3) explore a wide range of model selection methods.

3   Methodologies

In the following sub-sections, first we present the model for the data, then discuss the relevant algorithms and finally propose a complete clustering method.

3.1   Multinomial Mixture Model

Let $\mathbf{x}_i = x_{i,1}, x_{i,2}, \ldots, x_{i,D}$ is a D dimensional discrete count vector of order V, i.e. $\sum_{d=1}^{D} x_{i,d} = V$. Moreover, $\mathbf{x}_i$ is assumed to be an independent realization of the random variable $\mathbf{X}$, which follows a V-order Multinomial distribution [4]:

$$\mathcal{M}(\mathbf{x}_i|V, \boldsymbol{\mu}) = \binom{V}{x_{i,1}, x_{i,2}, \ldots, x_{i,D}} \prod_{d=1}^{D} \mu_d^{x_{i,d}} \tag{1}$$

here, $\boldsymbol{\mu}$ is the D dimensional parameter with $0 \leq \mu_d \leq 1$ and $\sum_{d=1}^{D} \mu_d = 1$. The set of samples can be modeled with a Multinomial Mixture (MM) model of K components:

$$f\left(\mathbf{x}_i | \Theta_K\right) = \sum_{k=1}^{K} \pi_k \, \mathcal{M}(\mathbf{x}_i | V, \boldsymbol{\mu}_k) \tag{2}$$

In Eq. (2), $\Theta_K = \{(\pi_1, \boldsymbol{\mu}_1), \ldots, (\pi_K, \boldsymbol{\mu}_K)\}$ is the set of model parameters, π_k is the mixing proportion with $\sum_{k=1}^{K} \pi_k = 1$ and $\mathcal{M}(\mathbf{x}_i | V, \boldsymbol{\mu}_k)$ is the density function (Eq. (1)) associated with the kth cluster.

3.2   Expectation Maximization Method

To cluster data with the model (Eq. (2)), we estimate its parameters using an Expectation Maximization (EM) [13] method that maximizes the log-likelihood:

$$L\left(\Theta\right) = \sum_{i=1}^{N} log \sum_{k=1}^{K} \pi_k \mathcal{M}\left(\mathbf{x}_i | \boldsymbol{\mu}_k\right) \tag{3}$$

where N is the number of samples. In the Expectation step (E-step), we compute posterior probability as:

$$\rho_{i,k} = p\left(z_i = k | \mathbf{x}_i\right) = \frac{\pi_k \prod_{d=1}^{D} \mu_{k,d}^{x_{i,d}}}{\sum_{l=1}^{K} \pi_l \prod_{d=1}^{D} \mu_{l,d}^{x_{i,d}}} \tag{4}$$

where $z_i \in \{0,1\}^K$ denotes the cluster label of the ith sample. In the Maximization step (M-step), we update π_k and $\mu_{k,d}$ as:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \rho_{i,k} \quad \text{and} \quad \mu_{k,d} = \frac{\sum_{i=1}^{N} \rho_{i,k} \, x_{i,d}}{\sum_{i=1}^{N} \sum_{r=1}^{D} \rho_{i,k} \, x_{i,r}} \tag{5}$$

The E and M steps run iteratively until certain convergence criterion (e.g., difference of log-likelihood) is met or until a maximum number of iterations.

3.3   Initialization for the EM Method

The EM method requires the initial values of the parameters as an input. We examine the following five methods to initialize the EM:

- **Random:** set the initial values randomly with $0 \leq \mu_d \leq 1$ and $\sum_{d=1}^{D} \mu_d = 1$.
- **rndEM [12]:** run a large number of random start and select the one which provides maximum likelihood value (Eq. (3)).
- **Small EM (smEM) [3]:** run multiple short runs of randomly initialized EM and choose the one with the maximum likelihood value. Here, short run means we do not wait until convergence and stop the algorithm when limited number of EM iterations is completed.

- **Classification EM (CEM) [3]:** it is similar to the smEM, except a classification stage is inserted between the E and M steps. The classification step involves assigning each point to one of the K components using the conditional probabilities (Eq. (4)) computed in the E step.
- **Stochastic EM (SEM) [3]:** it is similar to the smEM, except a stochastic step is inserted between the E and M steps. The stochastic step assigns $\mathbf{x}_i$ at random to one of the mixture components K according to the Multinomial distribution with the conditional probabilities (Eq. (4)).

3.4 Candidate Models Generation

Multiple EM (Mul-EM): This is the simplest way to generate the candidate models. In this approach, the EM method is run K_{max} times to generate the candidate models with $K = 1, \ldots, K_{max}$ clusters.

Integrated-EM (Int-EM): This approach [5, 17] do not explicitly generates the candidate models. Instead, it employs a single EM method that estimates the MM with K clusters and evaluate it at the same time. It begins with $K = K_{max}$ clusters and estimate its parameter. Then it annihilates a cluster with minimum π_k and estimate parameters with $K - 1$ clusters. This process continues within a single EM method until $K = 1$. See the EM-MML algorithm of [17] for details.

EM followed by Hierarchical Agglomerative Clustering (EM-HAC): This is our proposed model generation method, which aim is to generate a hierarchy of Multinomial Mixture (MM) models. Therefore, we exploit the Hierarchical Agglomerative Clustering (HAC) on the mixture model parameters $\hat{\Theta}_K$. In general, the HAC permits a variety of choices based on three principal issues: (a) the dissimilarity measure between clusters; (b) the criterion to select the clusters to be merged and (c) the representation of the merged cluster.

We use the symmetric Kullback–Leibler Divergence [4] (sKLD) as a measure of the dissimilarity between two Multinomial distributions as:

$$sKLD = \frac{D_{KL}\left(\boldsymbol{\mu}_a, \boldsymbol{\mu}_b\right) + D_{KL}\left(\boldsymbol{\mu}_b, \boldsymbol{\mu}_a\right)}{2} \text{ , where, } D_{KL}\left(\boldsymbol{\mu}_a, \boldsymbol{\mu}_b\right) = \sum_{d=1}^{D} \mu_{a,d}\, ln\left(\frac{\mu_{a,d}}{\mu_{b,d}}\right)$$

(6)

We choose "minimum sKLD" as the merging criterion (issue (b)). Besides we use the "complete linkage" criteria which is determined empirically.

In this clustering strategy, the set of models is represented by their parameters. After determining the clusters to be merged, similar to [8, 10], we compute the merged cluster parameters (issue (c)) as:

$$\pi_{merged} = \sum_{l \in \hat{\Theta}_{sub}} \pi_{l,k} \text{ and } \boldsymbol{\mu}_{merged} = \frac{\sum_{l \in \hat{\Theta}_{sub}} \pi_l \boldsymbol{\mu}_l}{\pi_{merged}}$$

(7)

where $\hat{\Theta}_{sub} \subseteq \hat{\Theta}_{K_{max}}$. As an outcome, we obtain a set of MMs with different K, which will be explored further for model selection.

3.5   Model Selection

Consider that, after HAC we have a set of MMs with $K_{max}, \ldots, 1$ components. The task of model selection can be defined as selecting the mixture model with K_o components such that $\hat{\Theta}_{K_o} = \{(\hat{\pi}_1, \hat{\mu}_1), \ldots, (\hat{\pi}_{K_o}, \hat{\mu}_{K_o})\}$. We consider parsimony-based [15] and evaluation graph based [16] methods in this work.

In the parsimony-based method[15], an objective function is employed, which minimizes certain model selection criteria. Such criteria involve the negative log likelihood augmented by a penalizing function in order to take into account the complexity of the model. One of the most widely used criteria is called the Bayesian Information Criterion (BIC) [6]:

$$BIC(K) = -2L(\hat{\Theta}) + \nu log\,(N) \tag{8}$$

where $\nu = KD - 1$ is the number of free parameters of the MM. The Integrated Completed Likelihood (ICL) criterion adds BIC with the mean entropy [2]:

$$ICL(K) = BIC(K) - 2\sum_{i=1}^{N} \log\left(p(z_i|\mathbf{x}_i)\right) \tag{9}$$

where $p(z_i|\mathbf{x}_i)$ is the conditional probability of the classified class label $z_i \in \{1, \ldots, K\}$ for the sample $\mathbf{x}_i$. The Minimum Message Length (MML) criterion, which has been recently proposed for MM, has the following form [17]:

$$MML(K) = \frac{D}{2} \sum_{k:\hat{\pi}_k > 0} \log\left(\frac{N\,\hat{\pi}_k}{12}\right) + \frac{K_{nz}}{2}\log\frac{N}{12} + \frac{K_{nz}\,(D+1)}{2} - L(\hat{\Theta}) \tag{10}$$

where K_{nz} is the number of clusters with non-zero probabilities. After computing the values of the model selection criteria for different $K \in \{1, ..., K_{max}\}$, we select K_o as the one that provides the minimum value of certain criterion.

For the evaluation graph based method, we consider the L-method (see [16] for details), where the knee point is detected in the plot constructed from the BIC values. The idea is to fit two lines at the left and right side of each point within the range $2,...,K_{max} - 1$. Finally, select the point as K_o that minimizes the total weighted root mean squared error.

3.6   Complete clustering method with MM

We propose a complele clustering method with the MM which clusters data and selects the number of clusters automatically. It consists of the following steps:

- **Step 1:** Apply the EM algorithm (Sec. 3.2) to estimate MM parameters with K_{max} clusters, i.e., $\hat{\Theta}_{k_{max}}$.
- **Step 2:** Apply the HAC method (Sec. 3.4) on $\hat{\Theta}_{k_{max}}$ to generate a set of models $\{\hat{\Theta}_k\}_{k=k_{max}-1,...,2}$.
- **Step 3:** Apply a model selection method (Sec. 3.5) to select $\hat{\Theta}_{K_o}$, i.e., the mixture model with the optimal number of components K_o.

4 Experimental Results and Discussion

We conduct experiments using both simulated and real data. For the evaluation, we compute the Adjusted Rand Index (ARI) [11], which is a pair counting based similarity measure among two clustering. Therefore, high value of ARI indicates highly similar clustering and hence high accuracy. For a dataset, we compute the ARI among the clustering result of a particular method and the true labels.

We evaluate the methods using the clustering accuracy, stability and computation time. We run each experiment 10 times and record the average value of the ARI as the accuracy, standard deviation of the ARI as the stability[1] and the average computation time.

4.1 Experimental Datasets

Simulated Datasets: We draw a finite set of discrete count vectors $\chi = \{\mathbf{x}_i\}_{i,\ldots,N}$ from MMs with different numbers (3, 5 and 10) and types: well-separated (***ws***) and not well-separated (***nws***) of clusters. Similar to [17], the types are verified using the sKLD[2] values. We consider samples of different dimensions: 3, 5, 10, 20 and 40. For each MM, we generate 100 sets of data each having 1000 i.i.d. samples. In the synthetic data generation process, first we contruct a MM model with K clusters. The model parameters ($\boldsymbol{\mu}_k$) for each cluster is sampled from a Dirichlet distribution. The order (V_k) of each cluster is sampled randomly from a certain range between $0.5D$ to $1.5D$. After determining the cluster parameters ($\boldsymbol{\mu}_k$) and orders (V_k) we draw the data samples.

Real Datasets: We consider 8 text datasets used in [20]. They consist of discrete count vectors, extracted from different documents collections. The choice was due to its good representation of different characteristics, such as the number of observations (documents), number of features (terms) and the number of clusters. The chosen datasets are listed in Table 1. We refer the readers to the Sec. 4.2 of [20] for additional details about the construction of these datasets.

4.2 Comparisons

First we compare the initialization strategies listed in Sec. 3.3 and consistently use the best one for the rest of the experiments. Afterward, we evaluate the model generation methods discussed in Sec. 3.4. Finally, we evaluate the model selection strategies discussed in Sec. 3.4.

---

[1] Stability provides a measure of robustness w.r.t. different initializations. A stable method should provide similar results for different runs, irrespective of its initialization. Therefore, a smaller value of the standard deviation indicates similar results for different runs and hence higher stability of the clustering method.

[2] A lower sKLD value among the cluster parameters indicates well-separated clusters, whereas higher value indicates less separation or a certain amount of overlap. Besides computing the sKLD value, we also verified the separation by observing the Bayes error rate among the clusters.

Table 1. Document text datasets for real data experiments. **N** denotes number of samples, **D** denotes number of features and **K** denotes the number of clusters. The source of the datasets are - **NG20**: 20 Newsgroups, **Classic**: ACM/CISI/CRANFIELD/MEDLINE, **Ohscal**: OHSUMED, **K1b**: WebACE, **Hitech**: SJM-TREC, **Reviews**: SJM-TREC, **Sports**: SJM-TREC and **La12**: LAT-TREC.

	NG20	Classic	Ohscal	K1b	Hitech	Reviews	Sports	La12
N	19949	7094	11162	2340	2310	4069	8580	6279
D	43586	41681	11465	21839	10080	18483	14870	31472
K	20	4	10	6	6	5	7	6

Initialization Methods: The experimental settings for the initialization methods (see Sec. 3.3) consist of: 1 trial for *Random*, 100 trials for *rndEM*, 5 trials with 50 maximum EM iterations for *smEM* and *CEM* and 1 trial with 500 maximum EM iterations for *SEM*. The initial parameters obtained from these methods are experimented with the EM method discussed in Sec. 3.2. Fig. 1 illustrates the results w.r.t. the clustering accuracy for both simulated[3] and real datasets. From all experimental results we have the following observations:

- For the simulated data, the *smEM* is the best method while the *CEM* is very competitive. However, for the real data *smEM* provides the best accuracy (except the *sport* dataset). The second choice is the *CEM* method.
- In terms of stability, *smEM* is the best for simulated data and *CEM* is best for the real data.
- In terms of computation time, these methods can be ordered as follows: *Random* < *rndEM* < *CEM* < *smEM* < *SEM*.

Similar to [14], we emphasize on the clustering accuracy as the main criteria to evaluate the initialization methods. Therefore, we choose the *SEM* method for further experiments.

Model Generation Methods: In this experiment, we aim to generate a set of candidate models with the methods discusses in Sec. 3.4. Among them, the *Mul-EM* and *EM-HAC* explicitly generate the models and the *Int-EM* generates them implicitly. All methods are initialized with the *smEM* method. Moreover, same initializations are used in *Int-EM* and *EM-HAC*. Settings of these methods consist of: 100 maximum number of EM iterations, 10^{-5} as the convergence threshold for the log-likelihood difference, $K_{min} = 2$ and $K_{max} = 15$, execept for *NG20* $K_{max} = 30$. Fig. 2 illustrates a comparison of these methods w.r.t. the accuracy[4] and stability. Table 2 provides a comparison[5] of the computation time for real data. From all experimental results we have the following observations:

---

[3] Due to limited space, we show results only for **nws** simulated samples with $K = 3$.

[4] This computation considers that the true numbers of clusters are known.

[5] Time comparison for the synthetic data provides similar observation as real data. Therefore, to save space we do not present those results.
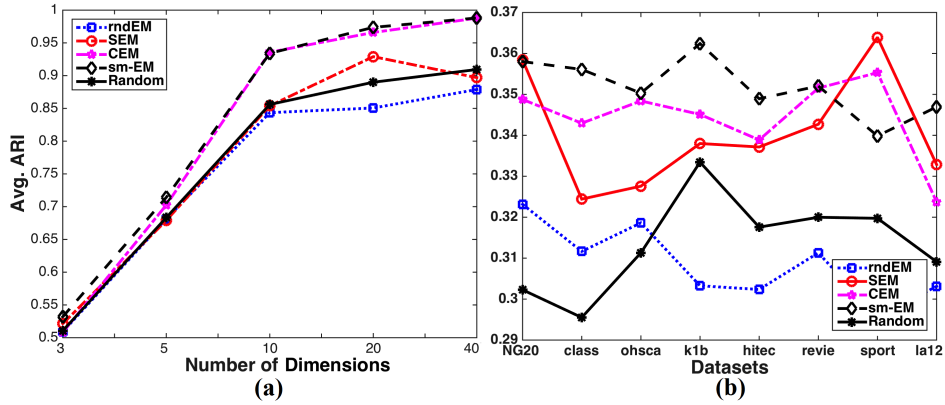
Fig. 1. Illustration of the accuracy of the initialization methods, computed from: (a) simulated **nws** samples with $K = 3$ and (b) real text datasets.
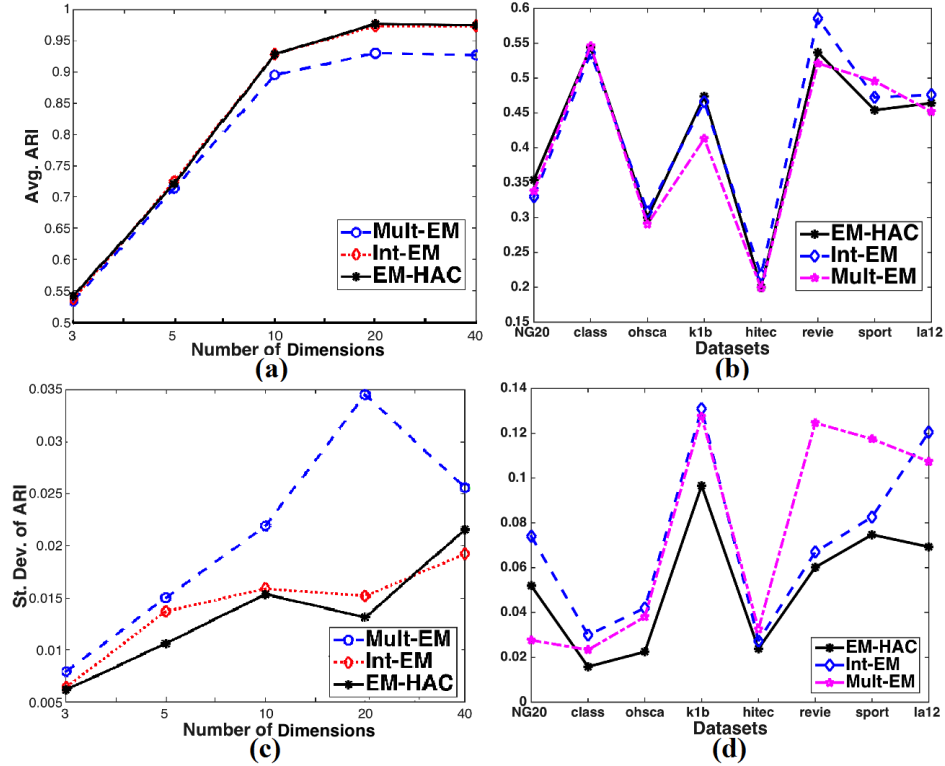

Fig. 2. Illustration of the clustering *accuracy* in (a) and (b), and *stability* in (c) and (d) for the model generation methods. (a) and (c) are computed from the simulated **nws** samples; (b) and (d) are computed from real text datasets.

- For the simulated data: *EM-HAC* and *Int-EM* are very competitive w.r.t. accuracy and time (results not shown). *EM-HAC* is the best on stability. *Mul-EM* was always performing worse except in a very few experiments.
- For the real[6] data, no single method outperforms others w.r.t. the accuracy. *EM-HAC* performs best in 3 datasets, *Int-EM* is best in 4 datasets and *Mul-EM* is best in 1 dataset. *EM-HAC* is best w.r.t. the stability (7 out of 8 datasets). Most interestingly, *EM-HAC* shows significantly better performance in terms of computation time as it is ~ 2.5 times faster than *Int-EM* and ~ 9 times faster than *Mul-EM*.

Based on the above experiments and observations, we can suggest that *Int-EM* is preferred when only accuracy is concerned. However, *EM-HAC* is preferred when stability and time has importantce besides accuracy.

Table 2. Comparison of the computation time (in seconds) among the model generation methods.

	NG20	Classic	ohscal	k1b	hightech	reviews	sports	la12
EM-HAC	108.5	6.9	19.2	3.8	3.6	9.9	17.7	19.9
Int-EM	353.2	10.8	42.2	9.6	8.2	21.7	46.3	44.2
Mult-EM	2844.0	54.4	95.6	29.1	20.7	59.1	104.1	134.6

Model Selection Methods: We evaluate different model selection criteria (see Sec. 3.5) with the *EM-HAC*. Moreover, we consider the *MML* with *Int-EM*, also called *EM-MML*, as proposed in [17]. Fig. 3 illustrates a comparison with both simulated and real data w.r.t. the rate of correct number of components selection. Our observations from these results are as follows:

- For the simulated data: *BIC* provides the best rate (except $K = 3$). *ICL* is equivalent to the *BIC* for higher K. Rate of *MML* decreases with the increase of K. Moreover, *MML* performs better with *EM-HAC* rather than with *Int-EM*. The *LM* provides mediocre accuracy for all clusters. The *LLH* criterion fails significantly.
- For the real data: *LM* provides very good ($\sim 90\%$) rate for 4 (*classic, hightech, review* and *la12*) datasets. Among the other methods, *MML* shows success in the *review* dataset, *LLH* is successful for the *classic* dataset.

From the above observations we realize that, the L-method (*LM*) is the best choice with the proposed clustering method. However, we want to emphasize that it is yet necessary to conduct further research on the model selection issue as there is no single method which uniquely provides reasonable rate for all data.

---

[6] In this paper we are interested only to compare different MM based MBC methods. We refer readers to [20] for a comparison among different other methods. From [20] we observed that, the *mixmns* (*Mul-EM* in this paper) performs better than the non-MBC methods, such as the *kmns* (k-means) and the *skmns* (spherical k-means).
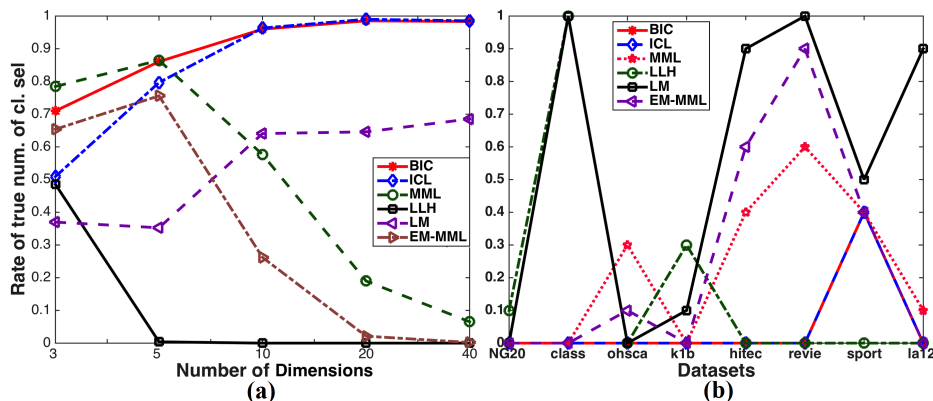
Fig. 3. Illustration of the rate of correct model selection, results in (a) are computed from the simulated samples and results in (b) are computed from real text datasets.

5   Conclusions

In this paper, we present a comparative study among different clustering methods with the Multinomial Mixture models. We experimentally evaluate the related issues, such as initialization, model estimation and generation and model selection. Besides, we propose a novel method for efficiently estimating the candidate models. Experimental results on both simulated and real data show that: (a) small run of EM (smEM) is the best choice for initialization (b) proposed hybrid model-based clustering, called EM-HAC is the best choice for candidate models estimation and (c) L-method is the best choice for model selection. As future work, we foresee the necessity to conduct further research on the model selection issue. Moreover, it is also necessary to evaluate these methods on more real-world discrete datasets obtained from a variety of different contexts.

Acknowledgments

References

[1] Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von Mises-Fisher distributions. In: Journal of Machine Learning Research. pp. 1345–1382 (2005)
[2] Biernacki, C., Celeux, G., Govaert, G.: Assessing a mixture model for clustering with the integrated completed likelihood. IEEE TPAMI 22(7), 719–725 (2000)

[3] Biernacki, C., Celeux, G., Govaert, G.: Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. Computational Statistics & Data Analysis 41(3), 561–575 (2003)

[4] Bishop, C.M., et al.: Pattern recognition and machine learning, vol. 4. springer New York (2006)

[5] Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE TPAMI 24(3), 381–396 (2002)

[6] Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97(458), 611–631 (2002)

[7] Fraley, C., Raftery, A.E.: Model-based methods of classification: using the mclust software in chemometrics. Journal of Statistical Software 18(6), 1–13 (2007)

[8] Garcia, V., Nielsen, F.: Simplification and hierarchical representations of mixtures of exponential families. Signal Processing 90(12), 3197–3212 (2010)

[9] Hasnat, M.A., Alata, O., Trémeau, A.: Unsupervised clustering of depth images using watson mixture model. In: 22nd International Conference on Pattern Recognition (ICPR). pp. 214–219. IEEE (2014)

[10] Hasnat, M.A., Alata, O., Trémeau, A.: Model-based hierarchical clustering with Bregman divergence and Fisher mixture model: Application to depth image analysis. Statistics and Computing (2015)

[11] Hubert, L., Arabie, P.: Comparing partitions. Journal of classification 2(1), 193–218 (1985)

[12] Maitra, R.: Initializing partition-optimization algorithms. IEEE/ACM Trans. on Computational Biology and Bioinformatics 6(1), 144–157 (2009)

[13] McLachlan, G.J., Krishnan, T.: The EM algorithm and extensions. Wiley series in probability and statistics, Wiley, 2. ed edn. (2008)

[14] Meilă, M., Heckerman, D.: An experimental comparison of model-based clustering methods. Machine Learning 42(1-2), 9–29 (2001)

[15] Melnykov, V., Maitra, R.: Finite mixture models and model-based clustering. Statistics Surveys 4, 80–116 (2010)

[16] Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: IEEE Conf. on Tools with Artificial Intelligence. pp. 576–584 (2004)

[17] Silvestre, C., Cardoso, M.G., Figueiredo, M.A.: Identifying the number of clusters in discrete mixture models. arXiv preprint arXiv:1409.7419 (2014)

[18] Vaithyanathan, S., Dom, B.: Model-based hierarchical clustering. In: Proc of the Uncertainty in Artificial Intelligence. pp. 599–608 (2000)

[19] Zhong, S., Ghosh, J.: A unified framework for model-based clustering. Journal of Machine Learning Research 4, 1001–1037 (2003)

[20] Zhong, S., Ghosh, J.: Generative model-based document clustering: a comparative study. Knowledge and Information Systems 8(3), 374–384 (2005)