Statistics and Its Interface Volume 12 (2019) 227–237

Pólya urn model and its application to text categorization*

Haibin Zhang, Xianyi Wu†, and Xueqin Zhou

Pólya urn model is a basic model widely applied in statistics and text mining. Most algorithms to training the model are very slow and complicated so that it generally difficult to fit a Pólya urn model to big data sets. This paper proposes a new minorization-maximization (MM) algorithm for the maximum likelihood estimation (MLE) of the Pólya urn model in which the surrogate function is constructed by means of a simple convex function. The convergence of the MM algorithm is analyzed and the asymptotic normality of the corresponding MLE for non-identically distributed observations is also derived. The performance of this new MM algorithm is also compared with Newton method and other MM algorithms. The Pólya urn model is applied to text categorization. Its superiority to naive Bayes (NB) classifier, k-Nearest Neighbor (k-NN) and support vector machine (SVM) are demonstrated by a real newsgroup dataset.

Keywords and phrases: Pólya urn model, Minorization-maximization, Asymptotic properties, Text categorization.

1. INTRODUCTION

Pólya urn model was first devised by Eggenberger and Pólya in 1923, so as to model spread of contagious diseases, and then widely applied to engineering, natural and social sciences, and finance, etc. [11, 19]. This model allows for interpretations in both the view points of frequentist and Bayesian statistics. Within the frequentist framework, the model characterizes the process of repeatedly and randomly drawing a ball at each stage from a urn containing K ($K \geq 2$, known) colors of balls, in which the drawn ball is returned back to the urn along with c new balls of the same color at each stage. This process endows the urn with a self-reinforcing property: the rich gets richer. From the Bayesian perspective, the model characterizes the situation where an observer knows that the urn contains K colors of balls, but does not know in advance the proportion of each color in the urn, of which, the uncertainty is, in turn, modeled by a Dirichlet distribution, which is also the limiting distribution of the Pólya urn model interpreted in frequentist perspective. Almost all discrete probability distributions can be made related to Pólya urn model [17], such as the beta-binomial distribution and the Dirichlet-multinomial distribution. Dirichlet process (DP) [2, 4] is an extension of the Pólya urn model which chooses balls from a urn with possibly infinitely may colors of balls.

For parameter estimation, we focus on the estimation of the unknown prior parameters based on the empirical Bayesian method. The MLE of Dirichlet process priors was successfully solved by Korwar and Hollander [9]. Yang and Wu [24] discussed the case of Dirichlet process priors with monotone missing data. The maximum likelihood estimate is difficult to compute for Pólya urn model. Nearchal and Morel [14, 15] proposed an improved method for the computation of the maximum likelihood estimates. Yu and Shaw [25] developed an efficient algorithm for accurate computation of log-likelihood function. However, asymptotic properties have not been discussed in the literature. In this paper, the asymptotical normality of the MLE is proved.

There exists no closed-form solution for the MLE of Pólya urn model so that the numerical methods are required. The MM algorithm [6, 23] is a powerful tool of the most widely used optimization methods for numerical problems, creating a surrogate function in the first M step and maximized in the second M step. This two step process always drives the objective function uphill and is iterated until the parameters converge. Expectation-maximization (EM) is a special case of the MM algorithm and the surrogate function of EM is conditional expectation. In this article, we propose a new algorithm to compute the maximum likelihood estimates. The key step of an MM algorithm is to find a good surrogate function. Instead of a highly complicated surrogate function for Pólya urn model by Zhou and Lange [26], the surrogate function in our MM algorithm, based on the convexity of the likelihood function, is simpler and straightforward which avoids any construction of complex inequalities in the derivations. To evaluate the performance of this MM algorithm, we compared it to Newton's method, MM algorithm and two accelerated MM algorithms: SqMPE1 (minimal polynomial extrapolation) and SqRRE1 (reduced rank extrapolation) [26]. Here note that, by Hua and Zhang [5], Newton's method is equivalent to an EM algorithm.

Text categorization [12] has become a very useful tool to find relevant information in text mining and has a wide range of applications including spam filtering, language identification and sentiment analysis. In this paper, We apply the Pólya urn model to text categorization. Comparisons

*This research was supported by NSFC under grant No. 71771089.
†Corresponding author.

of its performance to other classifiers such as naive Bayes, k-NN and SVM show that this new classifier is more powerful.

This paper is organized as what follows. In Section 2, we make a brief review of the Pólya urn model. The maximum likelihood estimates of this model are presented in section 3. We compare the method of the MLE to other four different MM algorithms by simulations in Section 4. A few experiments of the model applied to text categorization are discussed in Section 5. Section 6 concludes the paper with some concluding remarks.

2. THE PÓLYA URN MODEL

There are two ways to define the Pólya urn model as described below.

Initially, an urn contains an unknown number $a_k \in \mathbb{N}^+$ of color k balls, $k = 1, 2, \cdots, K$. Denote by $\boldsymbol{a} = (a_1, a_2, \ldots, a_K)$. At each time $i = 1, 2, \ldots$, a ball is randomly picked out from the urn and then returned along with c new balls of the same color back to the urn, where c may be a positive, zero or negative integer, as long as it makes sense.

Denote by e_k is the column vector with 1 at the kth component and zero at the others and introduce a K-dimensional random binary vector $\boldsymbol{X}_i$, taking values from $\{e_1, e_2, \ldots, e_K\}$, to represent the color of the ball picked from the urn at time i, so that $\boldsymbol{X}_i = e_k$ indicates that a ball of color k is drawn from the run at stage i.

The stochastic process $\{\mathbf{X}_i, i \geq 1\}$ or its distribution is called *Pólya urn model*. Clearly,

$$\boldsymbol{Y}_n = (Y_1, Y_2, \cdots, Y_K)' = \sum_{i=1}^{n} \boldsymbol{X}_i$$

indicates the total numbers of the balls selected in the first n periods, categorized by colors.

Recall the notation $r^{(s,j)} = r(r+s)(r+2s)\cdots(r+(j-1)s)$ with the convention $r^{(s,0)} = 1$ and $r^{[j]} \triangleq r^{(1,j)} = r(r+1)\cdots(r+j-1)$ for generalized permutation numbers, where $r, s \in \mathbb{R}$ and $j \in \mathbb{N}$. It is easy to see that $r^{(s,j)} = s^j(\frac{r}{s})^{[j]}$. Moreover, denote by

$$\alpha = \frac{\sum_{k=1}^{K} a_k}{c}, p_k = \frac{a_k}{\sum_{k=1}^{K} a_k}, k = 1, 2, \cdots, K \tag{1}$$

where, as just stated, c is the number of the balls added to the urn at every stage. Let $\boldsymbol{p} = (p_1, p_2, \cdots, p_K)'$. Then, by Mahmoud [11], the joint distribution of $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n)$ is

$$\Pr(\boldsymbol{X}_1 = \boldsymbol{x}_1, \boldsymbol{X}_2 = \boldsymbol{x}_2, \cdots, \boldsymbol{X}_n = \boldsymbol{x}_n) = \frac{a_1^{(c,y_1)} \cdots a_K^{(c,y_K)}}{(a_1 + \cdots + a_K)^{(c,n)}} = \frac{(\alpha p_1)^{[y_1]} \cdots (\alpha p_K)^{[y_K]}}{\alpha^{[n]}}, \tag{2}$$

where $\boldsymbol{y} = (y_1, y_k \cdots, y_K) = \sum_{i=1}^{n} \boldsymbol{x}_i \in \{0, 1, \cdots, n\}^K$ satisfies $\sum_{k=1}^{K} y_k = n$ and, hence, the distribution of $\boldsymbol{Y}_n$ is

$$\begin{aligned} \Pr(\boldsymbol{Y}_n = \boldsymbol{y}) &= \binom{n}{y_1, \cdots, y_K} \frac{a_1^{(c,y_1)} \cdots a_K^{(c,y_K)}}{(a_1 + \cdots + a_K)^{(c,n)}} \\ &= \binom{n}{y_1, \cdots, y_K} \frac{(\alpha p_1)^{[y_1]} \cdots (\alpha p_K)^{[y_K]}}{\alpha^{[n]}}, \end{aligned} \tag{3}$$

where $\binom{n}{y_1, y_2, \cdots, y_K}$ is the combinatory number of the scenarios of $y_1, y_2, \cdots, y_K$ balls of different colors, so that $\boldsymbol{Y}_n$ asymptotically follows a multinomial distribution, i.e.,

$$\lim_{\alpha \to \infty} \Pr(\boldsymbol{Y}_n = \boldsymbol{y}) = \binom{n}{y_1, \cdots, y_K} p_1^{y_1} \cdots p_K^{y_K}.$$

With particular specifications, from the Pólya urn model a hand of extensively discussed models can be induced. For example, for $K = 2$, the values $c = 0, -1$ and 1 correspond to binomial, hyper-geometric and beta-binomial distribution respectively, whereas, for $K > 2$, $c = 0$ and 1 correspond to multinomial and Dirichlet-multinomial distribution, respectively. In addition, by (3), $a_1 = a_2 = \cdots = a_K = c$ leads to

$$\begin{aligned} \Pr(\boldsymbol{Y}_n = \boldsymbol{y}) &= \binom{n}{y_1, \cdots, y_K} \frac{(y_1)!c^{y_1}, \cdots, (y_K)!c^{y_K}}{K^{[n]}c^n} \\ &= \frac{1}{\binom{K+n-1}{K-1}}, \end{aligned}$$

which is the uniform distribution on $\{0, 1, \cdots, n\}$ when $K = 2$.

A few useful properties of Pólya urn model are collected here for later reference:

1. For every pair of positive integers $i, j \in \mathbb{N}^+$, it is easy to see

   $$E(\boldsymbol{X}_i) = \boldsymbol{p}$$

   and

   $$\mathrm{Cov}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \begin{cases} \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}' & i = j \\ \frac{1}{\alpha+1}[\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}'] & i \neq j \end{cases}.$$

   Hence, the means and variances of the numbers of balls of the K colors picked from the urn in the first n stages are accordingly

   $$E(\boldsymbol{Y}_n) = n\boldsymbol{p}$$

   and

   $$\mathrm{Cov}(\boldsymbol{Y}_n) = n(1 + \frac{n-1}{\alpha+1})[\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}'].$$

2. The sequence of vectors

   $$\boldsymbol{Z}_n \triangleq (Z_{1n}, \cdots, Z_{Kn})' = \boldsymbol{a} + c\boldsymbol{Y}_n, n \geq 1,$$

   i.e., the numbers of balls in the urn after the nth drawing, is a homogeneous Markov chain. Let $\boldsymbol{\rho}_n \triangleq (\rho_{1n}, \rho_{2n}, \cdots, \rho_{Kn})' = \frac{\boldsymbol{Z}_n}{\sum_{k=1}^{K} a_k + cn}$ be the proportions of balls of every color after the nth drawing. Define $\mathfrak{F}_n = \sigma\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$. Then $(\boldsymbol{\rho}_n)_{n \geq 1}$ is an $\mathfrak{F}_n$-martingale.

3. Let

$$M_n = \frac{\boldsymbol{Y}_n}{n}.$$

Then $M_n \xrightarrow{d} \mathrm{Dir}(\frac{a_1}{c}, \cdots, \frac{a_K}{c})$ and $\boldsymbol{\rho}_n \xrightarrow{d} \mathrm{Dir}(\frac{a_1}{c}, \cdots, \frac{a_K}{c})$, where Dir () indicates a Dirichlet distribution.

Clearly, Equation (2) simply states that, for every n, the random vectors in $\boldsymbol{X}$ are exchangeable, but not independent and identically distributed (i.i.d.) and, hence, allow for an interpretation in term of Bayesian model. Let $\mathrm{Mul}(\boldsymbol{\theta})$ be a multinoulli distribution with probabilities $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_K)'$ and $\boldsymbol{\theta}$ in turn a random vector follows a Dirichlet distribution $\mathrm{Dir}(\frac{a_1}{c}, \cdots, \frac{a_K}{c})$. Then the Pólya urn model can be generated by the procedure

$$\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n | \boldsymbol{\theta} \overset{\text{i.i.d.}}{\sim} \mathrm{Mul}(\boldsymbol{\theta}), \boldsymbol{\theta} \sim \mathrm{Dir}(\alpha \boldsymbol{p}), \tag{4}$$

where α is the precision parameter indicating the uncertainty on the random proportion of the balls of all colors and $\boldsymbol{p}$ the expected proportions of balls, satisfying $\alpha \in (0, \infty)$ and $\sum_{k=1}^K p_k = 1, p_k \geq 0$. The marginal distribution of $(\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n)$ under model (4) is the same as (2) and the posterior distribution $\Pr(\boldsymbol{\theta} | \boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n)$ is a Dirichlet distribution with parameters $\alpha \mathbf{p} + \mathbf{Y}_n$. Also, the prediction rule of Pólya urn model is

$$\begin{aligned} \Pr(\boldsymbol{X}_{n+1} = e_k | \boldsymbol{X}_1, \cdots, \boldsymbol{X}_n) &= \frac{\alpha p_k + y_k}{\alpha + n} \\ &= \frac{n}{\alpha + n} \frac{y_k}{n} + \frac{\alpha}{\alpha + n} p_k, k = 1, 2, \ldots, K. \end{aligned} \tag{5}$$

This equivalence states that a Pólya urn model with n rounds of ball drawing can be considered a two step model: first draw a K-vector $\boldsymbol{\theta}$ from $\mathrm{Dir}(\alpha \mathbf{p})$ and then draw n balls from an urn with replacement in which the proportions of balls are defined by the vector $\boldsymbol{\theta}$.

The correspondence between the unknown parameters $\boldsymbol{a}$ in the former way and $(\alpha, \boldsymbol{p})$ is presented in Equation (1).

3. PARAMETER ESTIMATION

In practice such as text mining, information retrieval and bioinformatics, observations are frequently organized in a number,say M, of groups so as to reflect different features of objects, of which each (with possibly unbalanced size) is subject to a Pólya urn model. In mathematical language, the statistical model of the dataset is:

$$\begin{aligned} &\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2} \cdots, \boldsymbol{X}_{in_i} \overset{\text{i.i.d.}}{\sim} \mathrm{Mul}(\boldsymbol{\theta}_i), \\ &\boldsymbol{\theta}_i \overset{\text{i.i.d}}{\sim} \mathrm{Dir}(\alpha \boldsymbol{p}), i = 1, \cdots, M. \end{aligned} \tag{6}$$

Denote by $\mathcal{X}_i = (\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2} \cdots, \boldsymbol{X}_{in_i})$, $\mathcal{X} = (\mathcal{X}_i, i = 1, 2, \ldots, M)$ and $\mathcal{Y}_i = (Y_{i1}, Y_{i2}, \ldots, Y_{iK})' = \sum_{j=1}^{n_i} \boldsymbol{X}_{ij}, i = 1, 2, \cdots, M$, implying that $\mathcal{Y}_i$ are independent but non-identically distributed random variables.

3.1 The maximum likelihood estimates

By (2) and (6), with the observations $\mathcal{X}$, up to a constant independent of the unknown parameters, the log-likelihood function for $\boldsymbol{p}$ and α is

$$\begin{aligned} \ell(\boldsymbol{p}, \alpha) = &\sum_{i=1}^M [\sum_{k=1}^K \sum_{j=0}^{y_{ik}-1} \log(\alpha p_k + j) \\ &- \sum_{j=0}^{n_i - 1} \log(\alpha + j) + \binom{n_i}{y_{i1}, \cdots, y_{iK}}], \end{aligned} \tag{7}$$

where the sum $\sum_{j=0}^{-1} \cdot$ is treated as zero. For every k, there exist two extreme cases: the case $y_{1k} = \cdots = y_{Mk} = 0$ implies that the MLE is $\hat{p}_k = 0$ and the case $y_{ik} = n_i, i = 1, 2, \cdots, M$ implies that $\hat{p}_k = 1$. Generally, the likelihood equations are

$$\begin{cases} \frac{\partial \ell(\boldsymbol{p}, \alpha)}{\partial p_k} := \sum_{i=1}^M \sum_{j=0}^{y_{ik}-1} \frac{\alpha}{\alpha p_k + j} = \lambda, \\ \frac{\partial \ell(\boldsymbol{p}, \alpha)}{\partial \alpha} := \\ \sum_{i=1}^M \left[\sum_{k=1}^K \sum_{j=0}^{y_{ik}-1} \frac{p_k}{\alpha p_k + j} - \sum_{j=0}^{n_i - 1} \frac{1}{\alpha + j} \right] = 0, \end{cases} \tag{8}$$

where λ is the Lagrange multiplier. By some algebraic computation, (8) is equivalent to the equations

$$\sum_{i=1}^M \sum_{j=0}^{y_{ik}-1} \frac{1}{\alpha p_k + j} = \sum_{i=1}^M \sum_{j=0}^{n_i - 1} \frac{1}{\alpha + j}, \tag{9}$$

where $k = 1, \cdots, K$ and $\sum_{k=1}^K p_k = 1$. The following lemma gives the uniqueness of its solution.

Lemma 3.1. *Equation (9) has a unique solution.*

Proof. We use the contradiction argument. Suppose there exist two solutions $(\boldsymbol{p}, \alpha)$ and $(\boldsymbol{p}', \alpha')$ to Equation (9) such that, without loss of generality, $\alpha > \alpha'$.

By equation (8), we have $\sum_{i=1}^M \sum_{j=0}^{y_{ik}-1} \frac{\alpha}{\alpha p_k + j} = \lambda = \sum_{i=1}^M \sum_{j=0}^{y_{ik}-1} \frac{\alpha'}{\alpha' p'_k + j}$, then,

$$\sum_{i=1}^M \sum_{j=0}^{y_{ik}-1} \frac{\alpha \alpha' (p'_k - p_k) + j(\alpha - \alpha')}{(\alpha p_k + j)(\alpha' p'_k + j)} = 0.$$

Then $p'_k < p_k$ because $\alpha > \alpha'$. The same logic gives rise to the inequality $p'_1 < p_1, p'_2 < p_2, \cdots, p'_K < p_K$. Then $1 = \sum_{i=1}^K p'_k < \sum_{i=1}^K p_k = 1$, which contradicts the basic fact. The case $\alpha < \alpha'$ can be similarly discussed. Thus, $\alpha = \alpha'$ and $\boldsymbol{p} = \boldsymbol{p}'$. □

The Hessian matrix H is negative definite under these unique solutions, so that the maximum likelihood estimates of $(\boldsymbol{p}, \alpha)$ are denoted by $(\hat{\boldsymbol{p}}_{\text{mle}}, \hat{\alpha}_{\text{mle}})$. Note that the Hessian matrix is

$$H(p_1, p_2, \cdots, p_{K-1}, \alpha) = \begin{pmatrix} H_1 & C \\ C' & \ell_{\alpha\alpha} \end{pmatrix},$$

where

$\diamond$ $H_1 = \text{diag}(h_1, \cdots, h_{K-1}) + b\mathbf{1}_{K-1}1'_{K-1}$ with $h_k = -\sum_{i=1}^{M}(\sum_{j=0}^{y_{ik}-1}\frac{\alpha^2}{(\alpha p_k+j)^2}), k = 1, 2, \ldots, K-1$ and $b = -\sum_{i=1}^{M}\sum_{j=0}^{y_{iK}-1}\frac{\alpha^2}{(\alpha(1-\sum_{k=1}^{K-1}p_k)+j)^2}$,

$\diamond$ $C = (\ell_{\alpha p_1}, \ell_{\alpha p_2}, \cdots, \ell_{\alpha p_{K-1}})'$ with $\ell_{\alpha p_k} = -\sum_{i=1}^{M} \times [\sum_{k=1}^{K-1}\sum_{j=0}^{y_{ik}-1}\frac{\alpha p_k}{(\alpha p_k+j)^2} - \sum_{j=0}^{y_{iK}-1}\frac{\alpha(1-p_1-\cdots-p_{K-1})}{(\alpha(1-\cdots-p_{K-1})+j)^2}]$, and

$\diamond$ $\ell_{\alpha\alpha} = -\sum_{i=1}^{M}[\sum_{k=1}^{K}\sum_{j=0}^{y_{ik}-1}\frac{p_k^2}{(\alpha p_k+j)^2} - \sum_{j=0}^{n_i-1}\frac{1}{(\alpha+j)^2}]$.

Clearly, the Hessian matrix H is negative definite if and only if

$$\begin{aligned}\ell_{\alpha\alpha} - C'H_1^{-1}C &= \ell_{\alpha\alpha} - \sum_{k=1}^{K-1} h_k^{-1}\ell_{\alpha p_k}^2 \\ &+ \frac{1}{b^{-1}+\sum_{k=1}^{K-1}h_k^{-1}}\sum_{i,j}(h_i^{-1}\ell_{\alpha p_i})(h_j^{-1}\ell_{\alpha p_j}) < 0,\end{aligned} \tag{10}$$

because H_1 is negative definite.

3.2 A new computation algorithm

Obviously, there exists no closed-form solution to the log-likelihood equations (8) or (9). Hence, we need to numerically solve them. There exist a few methods, e.g., Newton's iteration method and the MM [26] that numerically compute the solution to (8) or (9). The disadvantage of Newton's method is the computation of Hessian matrices in each iteration, which can be prohibitively expensive for large scale problems, whereas the MM method may require a huge number of iterations, and higher dimension can dramatically decrease the convergence in computation.

We in this paper propose a new MM algorithm that will be referred to as the N-MM algorithm.

Rewrite $\log\ell(\boldsymbol{p}, \alpha)$ in (7) as

$$\begin{aligned}\log\ell(\boldsymbol{p}, \alpha) = \sum_{i=1}^{M}[\sum_{k=1}^{K}\log\frac{\Gamma(\alpha p_k + y_{ik})}{\Gamma(\alpha p_k)} + \log\frac{\Gamma(\alpha)}{\Gamma(\alpha+n_i)} \\ + \binom{n}{y_1, \cdots, y_K}],\end{aligned}$$

of which the key element is the function $\log\frac{\Gamma(n+x)}{\Gamma(x)}$ of x. The following lemma provides a lower bound of this function.

Lemma 3.2. *For any $x, y \geq 0$,*

$$\log\frac{\Gamma(n+x)}{\Gamma(x)} \geq \log\frac{\Gamma(n+y)}{\Gamma(y)} + y[\psi(n+y) - \psi(y)]\log\frac{x}{y},$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$.

Proof. For any fixed y, denote by

$$f(x) = \log\frac{\Gamma(n+x)}{\Gamma(x)} - y[\psi(n+y) - \psi(y)]\log x, x \in (0, \infty).$$

Then the derivative of $f(x)$ is

$$\begin{aligned}f'(x) &= [\psi(n+x) - \psi(x)] - [\psi(n+y) - \psi(y)]\frac{y}{x} \\ &= \begin{cases} < 0 & \text{if } x < y, \\ 0 & \text{if } x = y, \\ > 0 & \text{if } x > y. \end{cases}\end{aligned}$$

Hence y is the minimum point of $f(x)$, i.e., for any $x \in (0, \infty)$, $f(x) \geq f(y)$. This proves the lemma. □

Because $-\log(\cdot)$ is a convex function, so is $\log\frac{\Gamma(\alpha)}{\Gamma(\alpha+n_i)} = -\sum_{j=1}^{n_i}\log(\alpha+j-1)$ of α. Then, for any $\alpha^{(n)}$, supposed to be obtained in iteration n of certain iteration algorithm, it follows that

$$\begin{aligned}\log\frac{\Gamma(\alpha)}{\Gamma(\alpha+n_i)} &\geq \log\frac{\Gamma(\alpha^{(n)})}{\Gamma(\alpha^{(n)}+n_i)} \\ &+(\psi(\alpha^{(n)}+n_i) - \psi(\alpha^{(n)})(\alpha^{(n)} - \alpha).\end{aligned} \tag{11}$$

In addition, by Lemma 3.2,

$$\begin{aligned}\log\frac{\Gamma(\alpha p_k + y_{ik})}{\Gamma(\alpha p_k)} &\geq \log\frac{\Gamma(\alpha^{(n)}p_k + y_{ik})}{\Gamma(\alpha^{(n)}p_k)} \\ &+ \alpha^{(n)}p_k[\psi(\alpha^{(n)}p_k + y_{ik}) - \psi(\alpha^{(n)}p_k)]\log\frac{\alpha}{\alpha^{(n)}}\end{aligned} \tag{12}$$

and

$$\begin{aligned}\log\frac{\Gamma(\alpha p_k + y_{ik})}{\Gamma(\alpha p_k)} &\geq \log\frac{\Gamma(\alpha p_k^{(n)} + y_{ik})}{\Gamma(\alpha p_k^{(n)})} \\ &+ \alpha p_k^{(n)}[\psi(\alpha p_k^{(n)} + y_{ik}) - \psi(\alpha p_k^{(n)})]\log\frac{p_k}{p_k^{(n)}}.\end{aligned} \tag{13}$$

With the preparations above, in what follows we describe the N-MM (means new minorization-maximization) algorithm for computing α and $\boldsymbol{p}$.

Firstly, for given $\boldsymbol{p} = \boldsymbol{p}^{(n)}$, define $g(\alpha, \alpha^{(n)}) = \sum_{i=1}^{M}\{\sum_{k=1}^{K}(\log\frac{\Gamma(\alpha^{(n)}p_k+y_{ik})}{\Gamma(\alpha^{(n)}p_k)} + \alpha^{(n)}p_k[\psi(\alpha^{(n)}p_k + y_{ik}) - \psi(\alpha^{(n)}p_k)]\log\frac{\alpha}{\alpha^{(n)}}) + \log\frac{\Gamma(\alpha^{(n)}+n_i)}{\Gamma(\alpha^{(n)})} + (\psi(\alpha^{(n)} + n_i) - \psi(\alpha^{(n)})(\alpha^{(n)} - \alpha) + \binom{n}{y_1, \cdots, y_K}\}$. Then $g(\alpha, \alpha^{(n)})$ is a surrogate function for $\log\ell(\boldsymbol{p}^{(n)}, \alpha)$ for any $\alpha \in (0, \infty)$ in the sense that

$$\log\ell(\boldsymbol{p}^{(n)}, \alpha^{(n)}) = g(\alpha^{(n)}, \alpha^{(n)})$$

and

$$\log\ell(\boldsymbol{p}^{(n)}, \alpha) \geq g(\alpha, \alpha^{(n)}) \text{ for } \alpha \neq \alpha^{(n)}.$$

The construction of the surrogate function $g(\alpha, \alpha^{(n)})$ reflects the first M of the MM algorithm. The second M of the algorithm maximizes $g(\alpha, \alpha^{(n)})$ rather than $\log\ell(\boldsymbol{p}^{(n)}, \alpha)$ to produce

(14)
$$\alpha^{(n+1)} = \frac{\sum_{i=1}^{M}\sum_{k=1}^{K}\alpha^{(n)}p_k^{(n)}[\psi(\alpha^{(n)}p_k^{(n)}+y_{ik})-\psi(\alpha^{(n)}p_k^{(n)})]}{\sum_{i=1}^{M}[\psi(\alpha^{(n)})+n_i)-\psi(\alpha^{(n)})]}.$$

It is thus readily seen that $\log\ell(\boldsymbol{p}^{(n)},\alpha^{(n+1)}) \geq g(\alpha^{(n+1)},\alpha^{(n)}) \geq g(\alpha^{(n)},\alpha^{(n)}) = \log\ell(\boldsymbol{p}^{(n)},\alpha^{(n)})$. Hence the N-MM iteration never decreases the log-likelihood. A convergence theory of MM algorithms can be found in, e.g., Hunter and Lange [6]. Note that Minka [16] derived this update from a different perspective.

Next, for given $\alpha = \alpha^{(n)}$ and the value $\boldsymbol{p}^{(n)}$ obtained from the previous iteration, define $h(\boldsymbol{p},\boldsymbol{p}^{(n)}) = \sum_{i=1}^{M}\{\sum_{k=1}^{K}[\log\frac{\Gamma(\alpha p_k^{(n)}+y_{ik})}{\Gamma(\alpha p_k^{(n)})} + \alpha p_k^{(n)}(\psi(\alpha p_k^{(n)} + y_{ik}) - \psi(\alpha p_k^{(n)}))\log\frac{p_k}{p_k^{(n)}}] + \log\frac{\Gamma(\alpha^{(n)}p_k+y_{ik})}{\Gamma(\alpha^{(n)}p_k)} + (\psi(\alpha^{(n)}) + n_i) - \psi(\alpha^{(n)})(\alpha^{(n)} - \alpha) + \binom{n}{y_1,\cdots,y_K}\}$, which plays the role of a surrogate function for $\log\ell(\boldsymbol{p},\alpha^{(n)})$. An application of Lagrange multiplier to the maximization of $h(\boldsymbol{p},\boldsymbol{p}^{(n)})$ under constraint $\sum_{k=1}^{K}p_k = 1$ generates the update rule of the probability vector $\boldsymbol{p}$, which is expressed as

(15)
$$p_k^{(n+1)} = \frac{\sum_{i=1}^{M}p_k^{(n)}[\psi(\alpha^{(n)}p_k^{(n)}+y_{ik})-\psi(\alpha^{(n)}p_k^{(n)})]}{\sum_{i=1}^{M}\sum_{k=1}^{K}p_k^{(n)}[\psi(\alpha^{(n)}p_k^{(n)}+y_{ik})-\psi(\alpha^{(n)}p_k^{(n)})]}.$$

To summarize, with (14) and (15), we have the following N-MM algorithm.

Algorithm 3.1.

Step 1. *Set initial value* $(\boldsymbol{p}^{(0)},\alpha^{(0)})$*; the moment estimate of* $(\boldsymbol{p},\alpha)$ *is a general choice of* $(\boldsymbol{p}^{(0)},\alpha^{(0)})$.

Step 2. *Update* $(\boldsymbol{p}^{(n)},\alpha^{(n)})$ *obtained from the n-th iteration with* $(\boldsymbol{p}^{(n+1)},\alpha^{(n+1)})$ *by (14) and (15).*

Step 3. *Stop when* $\frac{|\log(\boldsymbol{p}^{(n+1)},\alpha^{(n+1)})-\log(\boldsymbol{p}^{(n)},\alpha^{(n)})|}{|\log(\boldsymbol{p}^{(n)},\alpha^{(n)})|+1} < \epsilon$.

The required moment estimates of the parameters, which are used in the algorithm as initial values, can be deduced by property (a) in Section 2. To verify the convergence of Algorithm 3.1, one only needs to check the conditions R1-R6 but R4 and C2 in Vaida [20]. Therefore, by Theorem 3 of Vaida [20], we have the following theorem, of which a proof is given in Appendix A.

Theorem 3.1. *From any initial value* $(\boldsymbol{p}^{(0)},\alpha^{(0)})$, $(\boldsymbol{p}^{(n)},\alpha^{(n)}) \to (\boldsymbol{p}^{(\star)},\alpha^{(\star)})$ *as* $n \to \infty$, *for some stationary point* $(\boldsymbol{p}^{(\star)},\alpha^{(\star)})$. *Moreover,* $M(\boldsymbol{p}^{(\star)},\alpha^{(\star)}) = (\boldsymbol{p}^{(\star)},\alpha^{(\star)})$, *and, for every n, if* $(\boldsymbol{p}^{(n)},\alpha^{(n)}) \neq (\boldsymbol{p}^{(\star)},\alpha^{(\star)})$, *then the value of the likelihood strictly increases.*

3.3 Asymptotic normality of the MLE

First note that the treatment of the asymptotic theory for maximum likelihood estimates with independent but non-identically distributed random variables can be found in Leroy et al. [7]. Let $\boldsymbol{\phi} = (\boldsymbol{p},\alpha)$, $\boldsymbol{\phi}^0$ be the true value of the parameter and $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}}(Y_1,\cdots,Y_M)$ the solution to likelihood equations (8). Suppose that $\hat{\boldsymbol{\phi}}$ is a consistent estimator of $\boldsymbol{\phi}^0$.

Theorem 3.2. *The random vector* $\sqrt{n}(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}^0)$ *is asymptotically normal with zero mean and covariance matrix* $I^{-1}(\boldsymbol{\phi}^0)$.

The proof is given in Appendix B.

4. SIMULATION STUDY

In this section, we report a simulation study for Pólya Urn Model which was conducted to compare the performance of five different algorithms: N-MM, Newton's method, general MM algorithm and two accelerated MM algorithms: SqMPE1 (minimal polynomial extrapolation) and SqRRE1 (reduced rank extrapolation) (all can be found in Zhou and Lang, 2010). The simulation was proceeded under the following setting:

- Every group had the same sample sizes $n_i = 500$ and 1000 and parameter values $\alpha = 0.01, 0.1, 1$;
- M took two levels $M = 100$ and $M = 1000$;
- K took three levels $K = 5, 10, 50$;
- $\boldsymbol{p}$'s were generated by random sampling;
- $\epsilon = 10^{-6}$.

The average running times in seconds and mean squared errors of estimates (MSE) for each algorithm are reported in Table 1, where, the boldfaced numbers indicate that the average computation time of N-MM algorithm is much shorter than other algorithms for the most cases. For low dimensions, the MSE of N-MM algorithm did not show obvious advantage over other algorithms, whereas the MSE of N-MM algorithm had huge advantage over other algorithms when the dimension is high.

The difference of the convergence rate between N-MM and other other algorithms are depicted in Figure 1 for a set of simulated data under the setting $\alpha = 0.1$, $K = 5$, $M = 100$ and $n_i = 1000$. The curves in the figure show that N-MM converges more quickly than MM and the other three methods.

5. AN APPLICATION TO TEXT CLASSIFICATION

Text categorization that assigns a document to a text category is quite important in retrieving information from text. Commonly employed classifiers include Naive Bayes (NB), k-Nearest Neighbor (k-NN) and support vector machine (SVM). In this section, we report an experiment in which the Pólya urn model was applied to text categorization of the 20 Newsgroups dataset (available at http://www.qwone.com/~jason/20Newsgroups/) and the performance was compared with the classifiers naive Bayes, k-NN and SVM, using precision, recall and F-score as performance measures.

Table 1. The average running times in seconds (mean squared error of estimates are in parentheses), averaged over 500 runs

Methods	(n, α)	K=5		K=10		K=50	
		$M = 10^2$	$M = 10^3$	$M = 10^2$	$M = 10^3$	$M = 10^2$	$M = 10^3$
Newton's Method		0.061 (0.014)	0.041 (0.003)	0.035 (0.017)	0.049 (0.009)	0.069 (0.18)	0.088 (0.173)
MM		0.027 (0.016)	0.048 (0.003)	0.629 (**0.015**)	0.594 (0.008)	0.239 (0.175)	0.751 (0.158)
SqMPE1	(500,0.01)	0.039 (0.015)	0.033 (**0.003**)	0.077 (0.016)	0.731 (0.008)	0.264 (0.176)	0.812 (0.163)
SqMPE2		0.032 (**0.013**)	0.032 (0.003)	0.769 (0.016)	0.730 (**0.008**)	0.262 (0.176)	0.806 (0.162)
N-MM		**0.012** (0.013)	**0.021** (0.003)	**0.014** (0.017)	**0.025** (0.009)	**0.008** (**0.172**)	**0.039** (**0.153**)
Newton's Method		0.074 (0.007)	0.083 (0.004)	0.095 (0.011)	0.101 (0.011)	0.115 (0.184)	0.184 (0.225)
MM		0.593 (0.008)	0.742 (**0.004**)	0.861 (0.010)	0.961 (**0.010**)	2.109 (0.167)	2.224 (0.200)
SqMPE1	$(1000, 0.01)$	0.406 (0.008)	0.589 (0.004)	1.157 (**0.010**)	1.284 (0.010)	2.309 (0.168)	2.320 (0.207)
SqMPE2		0.389 (0.008)	0.485 (0.004)	1.068 (0.010)	1.289 (0.010)	2.321 (0.168)	2.356 (0.207
N-MM		**0.013** (**0.007**)	**0.025** (0.004)	**0.027** (0.011)	**0.029** (0.011)	**0.034** (**0.163**)	**0.048** (**0.185**)
Newton's Method		0.026 (0.135)	0.037 (**0.132**)	0.033 (0.959)	0.044 (0.829)	0.071 (32.329)	0.088 (34.926)
MM		0.124 (0.153)	0.135 (0.150)	0.138 (**0.943**)	0.174 (**0.765**)	0.184 (31.358)	0.190 (33.885)
SqMPE1	$(500, 0.1)$	0.138 (0.144)	0.148 (0.144)	0.141 (0.948)	0.179 (0.783)	0.179 (31.649)	0.185 (34.245)
SqMPE2		**0.006** (0.134)	0.140 (0.144)	0.141 (0.948)	0.178 (0.783)	0.178 (31.649)	0.182 (**34.206**)
N-MM		0.012 (**0.131**)	**0.133** (0.136)	**0.014** (0.958)	**0.025** (0.826)	**0.027** (**31.235**)	**0.052** (34.817)
Newton's Method		0.032 (**0.143**)	0.041 (**0.135**)	0.059 (1.039)	0.065 (1.100)	0.086 (35.882)	0.108 (44.126)
MM		0.329 (0.171)	0.327 (0.161)	0.379 (**0.951**)	0.365 (**0.984**)	0.608 (34.376)	0.608 (**42.307**)
SqMPE1	$(1000, 0.1)$	0.407 (0.162)	0.401 (0.153)	0.385 (0.986)	0.419 (1.018)	0.617 (34.883)	0.693 (43.216)
SqMPE2		0.408 (0.162)	0.393 (0.153)	0.382 (0.976)	0.384 (1.018)	0.635 (34.834)	0.677 (42.864)
N-MM		**0.019** (0.144)	**0.023** (0.136)	**0.024** (1.035)	**0.027** (1.094)	**0.033** (**33.779**)	**0.067** (42.968)
Newton's Method		0.026 (36.235)	0.036 (**21.583**)	0.039 (198.281)	0.045 (195.178)	0.067 (401.471)	**0.049** (434.337)
MM		0.087 (38.133)	0.088 (24.232)	0.101 (**184.037**)	0.119 (**181.361**)	0.117 (391.368)	0.141 (423.128)
SqMPE1	$(500, 0.5)$	0.114 (37.531)	0.108 (23.447)	0.120 (188.475)	0.114 (185.383)	0.113 (394.593)	0.126 (426.752)
SqMPE2		0.114 (37.531)	0.102 (23.449)	0.121 (188.179)	0.135 (185.383)	0.176 (395.647)	0.192 (427.695)
N-MM		**0.022** (**36.676**)	**0.029** (22.130)	**0.013** (187.179)	**0.033** (190.784)	**0.040** (**385.335**)	0.139 (**421.850**)
Newton's Method		0.037 (**30.668**)	0.041 (32.437)	0.066 (**108.808**)	0.053 (**228.878**)	0.009 (754.356)	0.064 (681.823)
MM		0.294 (30.586)	0.319 (35.010)	0.295 (154.702)	0.293 (254.120)	0.536 (647.361)	0.377 (653.377)
SqMPE1	$(1000, 0.5)$	0.393 (33.306)	0.390 (33.306)	0.282 (256.478)	0.352 (253.492)	0.436 (670.577)	0.392 (647.091)
SqMPE2		0.369 (37.770)	0.390 (33.356)	0.343 (170.517)	0.347 (264.245)	0.456 (670.647)	0.389 (651.467)
N-MM		**0.035** (32.784)	**0.042** (**30.346**)	**0.043** (191.243)	**0.052** (159.229)	**0.09** (**604.465**)	**0.194** (**627.590**)

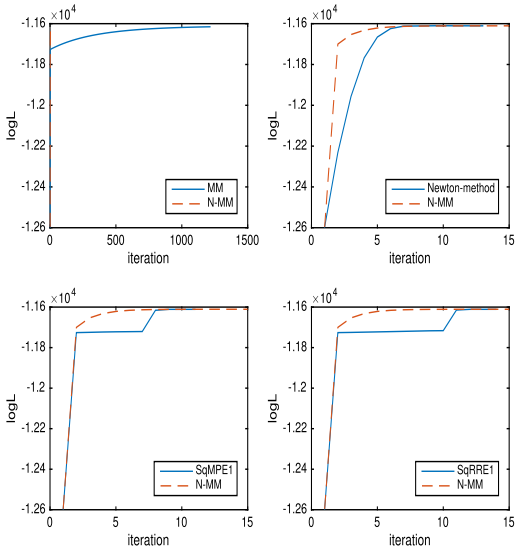

Figure 1. Algorithmic iterates for five methods.

5.1 Data description

Consisting of approximately 20,000 newsgroup documents, the 20 Newsgroups Text dataset is a publicly accessible and well-known dataset that is maintained by UCI's Knowledge Discovery in Database Archive for public use in testing text mining methodologies. The data, as the name indicates, are organized in 20 different newsgroups, with each corresponding to a different topic, as listed in the following Table 2. Some of the newsgroups are very closely related to each other, while others are highly unrelated.

We conducted three experiments, of which every analyzed a particular subset from the 20 Newsgroups Text. The first was carried out for two categories: sci.electronics and sci.med, the second dealt with three subgroups: talk.religion.misc, alt.atheism and soc.religion.christian, and in the last, the subgroups selected for analysis were five categories: comp.graphics, comp.os.ms windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware and comp.windows.x.

Table 2. Groups for 20 newsgroups

Group	Group
alt.atheism	rec.sport.hockey
comp.graphics	sci.crypt
comp.os.ms-windows.misc	sci.electronics
comp.sys.ibm.pc.hardware	sci.med
comp.sys.mac.hardware	sci.space
comp.windows.x	soc.religion.christian
misc.forsale	talk.politics.guns
rec.autos	talk.politics.mideast
rec.motorcycles	talk.politics.misc
rec.sport.baseball	talk.religion.misc

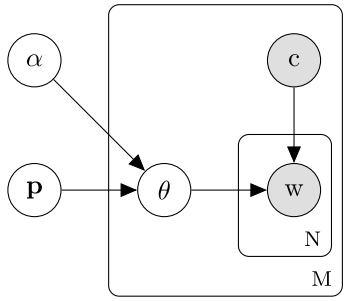

Figure 2. Graphical model representation of Pólya urn model for classification.

5.2 Preprocessing and representation of the text

The main aim of preprocessing the text is to reduce the problem's dimensionality by controlling the size of vocabulary. Four commonly used preprocessing steps of text classification were tokenization, normalization, stop word removal and stemming. Tokenization splits a text into words or other meaningful parts. Normalization includes lowercase conversion, words discard for those shorter than 3 or longer than 20 characters, and numbers and non-letter characters removal. Stop words were the words that are commonly encountered in texts with no importance in analysis. Stemming is used to identify the root/stem of a word.

After preprocessing of the datasets, we conducted a feature selection by the simple baseline method variance threshold that removed all features whose variance didn't meet some threshold. Then, we got the document vectors by "bag-of-words" assumptions [1]. The bag-of-words was a simplified representation, under which, a document was represented by the bag of its words, disregarding grammar and even word order.

5.3 Categorization methods and evaluation measures

In this subsection, some categorization methods and evaluation measures on text classification are concisely recalled.

5.3.1 Pólya urn classifier

Madsen et.al. [10] proposed to use Pólya urn to model documents. First fix a vocabulary of size K. Every document in the corpus is simply represented as a sequence of words without any consideration of such language aspects as grammar, paragraph structure and word order, etc. With the vocabulary, every word can be expressed as a vector from the set $\{e_1, e_2, \ldots, e_K\}$. To be specific, we are dealing with M documents $d_i = (w_{i1}, \cdots, w_{in_i}), i = 1, \cdots, M$, in which w_{ij} takes a value from $\{e_1, e_2, \ldots, e_K\}$ so as to indicate the position of a word in the vocabulary and n_i indicates the length of the document: If the jth word of d_i is identical to the kth of the vocabulary, then $w_{ij} = e_k$.

For those corpora, in which documents are all short essays talking about a single topic, the order of words in a document is, at least approximately, irrelevant and a document d_i can be thought of as an iid sample of n_i words from an multinomial distribution over the vocabulary. The distribution varies over documents according to a Dirichlet distribution. Each class is talking about one topic and a topic is a distribution on the vocabulary according to a latent Dirichlet allocation [3], each document choosing a class with probability $P(c_i = c) = q_c$, where $c = 1, \cdots C$ and $\sum_{c=1}^{C} q_c = 1$. Note that $P(w_{ij} = e_k | c_i = c) = \theta_{ck}$ is the weight for word k of class c. Denote by $\theta_j = (\theta_{j1}, \cdots, \theta_{jK})$, $\mathbf{q} = (q_1, q_2, \cdots, q_C)$, $\mathbf{p} = (p_1, \cdots, p_K)$. The document generating model can be characterized by

(1). Draw $\theta_j, j = 1, 2, \cdots \overset{\text{i.i.d.}}{\sim} \text{Dir}(\alpha\mathbf{p})$,
(2). Draw $c_i \overset{\text{i.i.d.}}{\sim} \text{Mul}(\mathbf{q})$, $i = 1, 2, \cdots, M$,
(3). Draw $w_{i1}, \cdots, w_{in_i} | c_i, \{\theta_j\}_{j=1}^{\infty} \overset{\text{i.i.d.}}{\sim} \text{Mul}(\theta_{c_i})$.

Parameter vector $\alpha\mathbf{p}$ can be interpreted as the initial number of balls of each color in the urn. According to the equations (2) and (6), the document is modeled as:

$$\Pr(d_i, c_i = c | \alpha, \mathbf{p}) = q_c \binom{n_i}{y_{i1}, \cdots, y_{iK}} \frac{(\alpha p_1)^{[y_{i1}]} \cdots (\alpha p_K)^{[y_{iK}]}}{\alpha^{[n]}},$$

where $\mathbf{y}_i = (y_{i1}, \cdots, y_{iK})$ denotes the frequencies of words appeared in document i. The Pólya urn classifier model is represented as a probabilistic graphical model in figure 2. As the figure makes clear, there are three levels to the Pólya urn classifier model representation. The parameters α and $\mathbf{p}$ are corpus-level parameters, assumed to be sampled once in the process of generating a corpus. The variables θ_j are document-level variables, sampled once per document. Finally, the variables c_i are observed category variables and the variables w_{dn} are observed word-level variables determined by the category variables and document-level variables. For a document d in the test set, Rennie [18] proposed three classifiers: normal (N), complement (C) and mixed (M), all of which assign the document to the class with the

Table 3. Evaluation Measures

	True Classification	*False Classification*
Predicted True Classification	TP (true positive)	FP (false positive)
Predicted false Classification	FN (false negative)	TN (true negative)

Table 4. Performance of classification algorithms

	Data Set											
	Two categories				Three categories				Five categories			
	PU	BN	KNN	SVM	PU	BN	KNN	SVM	PU	BN	KNN	SVM
Percision	**0.945**	0.903	0.612	0.783	**0.790**	0.658	0.529	0.587	**0.664**	0.557	0.318	0.439
Recall	**0.831**	0.727	0.588	0.564	**0.773**	0.645	0.490	0.564	**0.587**	0.527	0.279	0.428
F score	**0.884**	0.806	0.600	0.557	**0.776**	0.649	0.426	0.557	**0.586**	0.530	0.272	0.429

highest posterior probability. We choose the normal as the classifier, i.e. $\text{argmax}_c[\log \hat{q}_c + \sum_{k=1}^{K} f_k \log \hat{\theta}_{ck}]$, where f_k is the word frequency in the document d.

The weight for each class is estimated as a function of α and $\mathbf{p}$ coefficients by

$$\hat{\theta}_{ck} = \frac{T_{ck} + \hat{\alpha}^{\text{mle}} \hat{p}_k^{\text{mle}}}{T_c + \hat{\alpha}^{\text{mle}}},$$

where T_{ck} is the frequency word k appears in the documents of class c and T_c is the total number of words occurrences in class c in the test set. $\hat{p}_k^{\text{mle}}$ and $\hat{\alpha}^{\text{mle}}$ denote the maximum likelihood estimates of p_k and α in Pólya urn model, respectively. N_c is the total number of documents occurrences in class c and M is the number of the documents in the train test. The estimate $\hat{q}_c$ is $\frac{N_c}{M}$, which is the maximum likelihood estimate.

5.3.2 Contrast methods

Naive Bayes classifier has been widely used for text categorization [12]. The best class in NB classification is the most likely or maximum a posteriori, k-Nearest Neighbor is one of the most popular algorithms for text categorization [13]. In the classification process, k nearest documents to the test one in the training set are determined firstly. Then, the predication can be made according to the category distribution among these k nearest neighbors. Support vector machine is the supervised machine learning technique [23] and was first applied to text categorization by Joachims [8]. If the training data are linearly separable, SVM is trained via the optimization problem.

5.3.3 Evaluation measures

For a classification task, the precision and recall can be defined by the following contingency table 3. The Table should be signed clearly. Precision and recall are then defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}.$$

A measure that combines precision and recall is the harmonic mean of precision and recall, the F-score

$$F = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}.$$

5.4 Results comparisons

The results of the classification algorithms and their performance are listed in Table 4, in which the boldfaced numbers indicate the best among the experiments under the certain measures. It is shown that Pólya's urn classifier is better than naive Bayes classifier, k-Nearest Neighbor and support vector machine under the performance measures precision, recall and F-score.

6. DISCUSSION

This paper addressed Pólya's urn model and its application in text classification for parameter estimation. Because of the absence of a closed-form solution for the MLE, one needs to find MLE by means of numerical methods. To this end, based on the convexity of the functions, a new MM method is proposed which does not need to compute Hessian matrices as Newton's iteration method did and construct complex inequalities as the existing MM did [26]. We proved the convergence of the new MM method and the asymptotic normality of MLE. To examine the performance of this new MM algorithm, comparisons were made to Newton's method and a few other MM algorithms. The N-MM algorithm for Pólya Urn Model can not only have computational efficiency but also preserve accuracy from simulation studies. In experimental analysis, we reported applications in text categorization of 20 newsgroup data set by the Pólya's urn model, comparing to naive Bayes classifier, k-Nearest Neighbor and Support Vector Machine. We found that Pólya's urn classifier are better than naive Bayes classifier, k-Nearest Neighbor and support vector machine.

APPENDIX A

A.1 Proof of Theorem 3.1

Let $S_i = \{0, 1, \cdots, n_i\}$ be the support of the probability distribution function $P_i = \Pr(Y_i = y_i)$ which is defined in (3) and independent of the vector of unknown $\boldsymbol{\phi}$.

We verify conditions R1-R6 except R4 and C2 in Vaida [20] as what follows:

R1. Since $\sum_{k=1} p_k = 1, 0 < p_k < 1$ and $\alpha \in (0,\infty)$, then $(0,1)^{K-1} \times (0,\infty) \in \mathbb{R}^K$.

R2. It is obviously that $\log\ell(\boldsymbol{p},\alpha)$ is differentiable with continuous derivative.

R5. The surrogate functions $g(\alpha,\alpha^{(n)})$ and $h(\boldsymbol{p},\boldsymbol{p}^{(n)})$ are differentiable with continuous derivative.

C2. We only verify C2 because R3 and R6 can easily be established by C2.
By lemma 1, Equation (9) has a unique solution. Since the MM iteration never decreases the log-likelihood function, there exists a unique global maximum of the log-likelihood. Then $\Omega = \{(\boldsymbol{p}',\alpha') \in (0,1)^{K-1} \times (0,\infty) \in \mathbb{R}^K : \log\ell(\boldsymbol{p}',\alpha') > \log\ell(\boldsymbol{p},\alpha)\}$ is compact. Because there only exists one stationary point, then the set S is isolated.

A.2 Proof of Theorem 3.2

It suffices to verify Assumptions 3-11 in Leroy et al. (2016) one by one as what follows.

1. Assumption 3. The all partial derivatives exist by direct calculations.
2. Assumption 4 and Assumption 5. Note that $\frac{\partial P_i}{\partial p_k} = \alpha P_i[\sum_{j=1}^{y_{ik}} \frac{1}{\alpha p_k+j-1} - \sum_{j=1}^{n-y_{ik}} \frac{1}{\alpha(1-p_1-\cdots-p_{K-1})+j}]$, where $k = 1,2,\cdots,K-1$ and $i = 1,\cdots,M$. By the method of induction in n_i,
   (1). For $n_i = 1$, it is readily seen that $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial p_k} = 0$.
   (2). Suppose the desired result holds for every $n_i = m$,i.e., $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial p_k} = \sum_{\boldsymbol{y}_i} \alpha P_i[\sum_{j=1}^{y_{ik}} \frac{1}{\alpha p_k+j-1} - \sum_{j=1}^{m-y_{ik}} \frac{1}{\alpha(1-p_1-\cdots-p_{K-1})+j}] = 0$.
   (3). For $n_i = m+1$, there are three possibilities: $y'_{ik} = y_{ik}+1$, $y'_{iK} = y_{iK}+1$ and one of $y_{i1},\cdots,y_{ik-1},y_{ik+1},\cdots,y_{iK-1}$ increases by 1. Therefore, $\sum_{\boldsymbol{y}'_i} \frac{\partial P'_i}{\partial p_k} = (m+1)\frac{\alpha p_k+y}{\alpha+n} \sum_{\boldsymbol{y}_i} (\frac{\partial P_i}{\partial p_k} + P_i \frac{\alpha}{\alpha p_k+y}) + (m+1)\frac{\alpha(1-p_k)+y}{\alpha+n} \sum_{\boldsymbol{y}_i} (\frac{\partial P_i}{\partial p_k} - P_i \frac{\alpha}{\alpha(1-p_k)+y}) + (m+1)\frac{\alpha p_\ell+y_{i\ell}}{\alpha+n} \sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial p_k} = 0$, where $\ell = 1,\cdots,k-1,k+1,\cdots,k-1$.

   Then we have $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial p_k} = \frac{1}{\partial p_k}(\sum_{\boldsymbol{y}_i} P_i)$. By similar method, $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial \alpha} = \frac{1}{\partial \alpha}(\sum_{\boldsymbol{y}_i} P_i)$. $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial p_j \partial \alpha} = \frac{1}{\partial p_j \partial \alpha}(\sum_{\boldsymbol{y}_i} P_i)$, $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial p_k \partial p_j} = \frac{1}{\partial p_k \partial p_j}(\sum_{\boldsymbol{y}_i} P_i)$ and $\sum_{\boldsymbol{y}_i} \frac{\partial P_i}{\partial \alpha \partial \alpha} = \frac{1}{\partial \alpha \partial \alpha}(\sum_{\boldsymbol{y}_i} P_i)$, $j,k = 1,\cdots,K-1$.
3. Assumption 6-8. The convergence in probability in Assumptions 6, 7 and 8 are ensured by the weak law of large numbers. Note that $E(\frac{\partial \log P_i}{\partial p_k}) = \alpha E[\sum_{j=1}^{y_{ik}} \frac{1}{\alpha p_k+j-1} - \sum_{j=1}^{n_i-y_{ik}} \frac{1}{\alpha(1-p_1-\cdots-p_{K-1})+j}] = \frac{\partial P_i}{\partial p_k} = 0$. Then, $\lim_{M\to\infty} \frac{\sum_{i=1}^M E(\frac{\partial \log P_i}{\partial p_k})}{M} = 0$. Similarly, we also have $\lim_{M\to\infty} \frac{\sum_{i=1}^M E(\frac{\partial \log P_i}{\partial \alpha})}{M} = 0$.

In addition,

$$
\begin{aligned}
&E(\frac{\partial \log P_i}{\partial p_k \partial p_\ell}) \\
&= -\alpha^2 E(\sum_{j=0}^{y_{iK}-1} \frac{1}{\alpha(1-p_1-\cdots-p_{K-1})+j)^2}) \\
&\quad + \begin{cases} -\alpha^2 E[\sum_{j=0}^{y_{ik}-1} \frac{1}{(\alpha p_k+j)^2}, k = \ell \\ 0, k \neq \ell \end{cases} \\
&E[\sum_{j=0}^{y_{ik}-1} \frac{1}{(\alpha p_k+j)^2}] \\
&= \sum_{\boldsymbol{y}_i} P_i \sum_{j=0}^{y_{ik}-1} \frac{1}{(\alpha p_k+j)^2} \\
&= \sum_{j=0}^{n_i} \frac{Pi(Y_{i1}=y_{i1},\cdots,Y_{ik}>j,\cdots,Y_{iK}=y_{iK})}{(\alpha p_k+j)^2}.
\end{aligned}
$$

Let $s_i = \sum_{j=0}^{n_i-1} \frac{1}{(\alpha+j)^2}$ and $t_{ik} = \sum_{j=0}^{n_i-1} \frac{P(Y_{i1}=y_{i_1},\cdots,Y_{ik}>j,\cdots,Y_{iK}=y_{iK})}{(\alpha p_k+j)^2}$.
Then, we have that

$$
E(\frac{\partial \log P_i}{\partial p_k \partial p_\ell}) = \begin{cases} -\alpha^2[t_{ik}+t_{iK}] & k = \ell \\ -\alpha^2 t_{iK} & k \neq \ell \end{cases}.
$$

Similarly, $E(\frac{\partial \log P_i}{\partial p_k \partial \alpha}) = -\alpha[\sum_{k=1}^{K-1} t_{ik}p_k - t_{iK}p_K]$ and $E(\frac{\partial \log P_i}{\partial \alpha \partial \alpha}) = -[\sum_{k=1}^{K} t_{ik}p_k^2 - s_i]$. Next, we show that $\lim_{M\to\infty} \frac{\sum_{i=1}^M E(\frac{\partial \log P_i}{\partial \alpha \alpha})}{M}$ exists and other cases are similar. Since $s_{max} \sqrt[M]{r_1 r_2 \cdot r_M} = \sqrt[M]{s_1 s_2 \cdots s_M} \leq \frac{\sum_{i=1}^M s_i}{M} \leq s_{\max}$, where $s_{max} = \max\{s_1,s_2,\cdots,s_M\}$ and $r_i = \frac{s_i}{s_{max}}, i = 1,\cdots,M$. By $\lim_{M\to\infty} \sqrt[M]{r_1 r_2 \cdot r_M} = 1$ and squeeze theorem, $\lim_{M\to\infty} \frac{\sum_{i=1}^M s_i}{M} = s_{\max}$. By similar method, $\lim_{M\to\infty} \frac{\sum_{i=1}^M t_{ik}}{M} = t_{\max,k}, k = 1,\cdots,K$, where $t_{\max k} = \max\{t_{1k},t_{2k},\cdots,t_{Mk}\}$. And

$$
\begin{aligned}
&E(\frac{\partial \log P_i}{\partial p_k \partial p_\ell \partial p_q}) \\
&= -2\alpha^3 E(\sum_{j=0}^{y_{iK}-1} \frac{1}{(\alpha(1-p_1-\cdots-p_{K-1})+j)^3}) \\
&\quad + \begin{cases} -2\alpha^3 E[\sum_{j=0}^{y_{ik}-1} \frac{1}{(\alpha p_k+j)^3}], k = \ell = q \\ 0, \text{others} \end{cases}
\end{aligned}
$$

It is easy to show that $\lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^M E(\frac{\partial \log P_i}{\partial p_k \partial p_\ell \partial p_q})$ exists.

4. Assumption 9. It is easy to derive it by Assumption 8.
5. Assumption 10. First note that $I(p_k,p_\ell) = \lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^M E(-\frac{\partial \log P_i}{\partial p_k \partial p_\ell})$, $I(p_k,\alpha) = \lim_{M\to\infty} \frac{1}{M} \times \sum_{i=1}^M E(-\frac{\partial \log P_i}{\partial p_k \partial \alpha})$ and $I(\alpha,\alpha) = \lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^M =$

$E(-\frac{\partial \log P_i}{\partial\alpha\partial\alpha})$. This gives the Fisher information matrix

$$I(p_1,\cdots,p_{K-1},\alpha)=\begin{pmatrix} I(p_1,\cdots,p_{K-1}) & D \\ D^{'} & I(\alpha\alpha)\end{pmatrix},$$

where $I(p_1,\cdots,p_{K-1})=\alpha^2[\mathrm{diag}\{t_{\max 1},\cdots,t_{\max K-1}\}+t_{\max K}\mathbf{1}\mathbf{1}']$, $D_i = \alpha[\sum_{k=1}^{K-1} t_{\max k}p_k - t_{\max K}p_K]\mathbf{1}$ and $I(\alpha\alpha) = \sum_{k=1}^{K}(t_{\max k}p_k^2 - s_{\max})$. Therefore, $I^{-1}(p_1,\cdots,p_{K-1}) = \frac{\mathrm{diag}\{\frac{1}{t_{\max 1}},\cdots,\frac{1}{t_{\max K-1}}\}}{\alpha^2}[1 - \frac{1}{\sum_{k=1}^{K}\frac{1}{t_{\max k}}}\mathbf{1}\mathbf{1}'\mathrm{diag}\{\frac{1}{t_{\max 1}},\cdots,\frac{1}{t_{\max K-1}}\}]$. Next we show that the Fisher information matrix is positive definite. Because $I(p_1,\cdots,p_{K-1})$ is a positive definite matrix, $I(p_1,\cdots,p_{K-1},\alpha)$ is positive definite if and only if $I(\alpha\alpha) - D^{'}I^{-1}(p_1,p_2,\cdots,p_{K-1})D > 0$. For $K=2$, let $p_1=p$, then $p_2=1-p$, $I(\alpha\alpha) - D^{'}I^{-1}(p)D = [t_{\max 1}p^2 + t_{\max 2}(1-p)^2 - s_{\max}] - \frac{1}{t_{\max 1}+t_{\max 2}}[t_{\max 1}p - t_{\max 2}(1-p)]^2 = \frac{1}{t_{\max 1}+t_{\max 2}}[t_{\max 1}t_{\max 2} - (t_{\max 1}+t_{\max 2})s_{\max}]$. Because $\alpha+j > \alpha p+j$, it follows that $\frac{1}{\alpha+j} < \frac{1}{\alpha p+j}$. We need to show $t_{\max 1}t_{\max 2} - (t_{\max 1}+t_{\max 2})s_{\max} > 0$. On the one hand, $\frac{t_{\max 1}}{2} - s_{\max} = \sum_{j=0}^{n_{\max}-1}\frac{P(Y_{\max 1}>j)}{2(\alpha p+j)^2} - \sum_{j=0}^{n_{\max}-1}\frac{1}{(\alpha+j)^2} > \sum_{j=0}^{n_{\max}-1}\frac{P(Y_{\max 1}>j)}{2(\alpha p+j)^2} - \sum_{j=0}^{n_{\max}-1}\frac{P(Y_{\max 1}>j)}{2(\alpha+j)^2} > \sum_{j=0}^{n_{\max}-1}P(Y_{\max 1} > j)[\frac{1}{2(\alpha p+j)^2} - \frac{1}{2(\alpha+j)^2}] > 0$. On the other hand, by similar computation, we have $\frac{t_{\max 2}}{2} - s_{\max} > 0$. Thus, $I(p,\alpha)$ is a positive definite matrix. For $K>2$, by the similar method, $I(\phi)$ is a positive definite matrix. This completes the examination of Assumption 10.

6. Assumption 11. According to Assumption 4, let $B_{ik} = \frac{\partial P_i}{\partial p_k}$ and $B_i = \frac{\partial P_i}{\partial\alpha}$, then $A_i = \sum_{k=1}^{K}B_{ik}^2 + B_i^2$ are bounded. For M enough large, $E[A_i I\{A_i > \epsilon\sqrt{M}\}] = 0$, where $I\{A\}$ is the indicator of set A.

ACKNOWLEDGEMENTS

We would like to thank the Editor and the anonymous referees for their constructive and insightful comments, which have led to a highly improved version of the paper.

Received 20 October 2017

REFERENCES

[1] Aldous, D. J. (1985). Exchangeability and related topics. *École d'Été de Probabilitis de Saint-Flour XIII-1983 (pp. 1–198)*. Springer Berlin Heidelberg. MR0883646

[2] Blackwell, D. and MacQueen, J. (1973). Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, **1** 353–355. MR0362614

[3] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 993–1022.

[4] Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **1** 209–230. MR0350949

[5] Hua, Z. and Zhang, Y. (2012). EM vs MM: A Case Study. *Computational Statistics and Data Analysis*, **56**, **no. 12**, 3909–3920. MR2957841

[6] Hunter, D. R. and Lange, K. (2004). A Tutorial on MM Algorithms, *The American Statistician*, **no.1**, 30–37. MR2055509

[7] Leroy, F., Dauxois, J. Y., and Tubert, B. P. (2016). On the parametric maximum likelihood estimator for independent

[8] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *In European conference on machine learning*. Springer Berlin Heidelberg, 137–142.

[9] Korwar, R. M., and Hollander, M. (1973). Contributions to the theory of Dirichlet processes. *The Annals of Probability*, 705–711. MR0350950

[10] Madsen, R. E., Kauchak, E., and Elkan, C. (2005). Modeling Word Burstiness using the Dirichlet distribution, *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 545–552.

[11] Mahmoud, H. (2008). *Pólya urn models*. CRC Press. MR2435823

[12] Manning, C. D. (2008). Prabhakar, R. and Schütze H., *Introduction to Information Retrieval*, Cambridge University Press.

[13] Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT press. MR1722790

[14] Neerchal, N. K. and Morel, J. G. (1998). Large cluster results for two parametric multinomial extra variation models. *Journal of the American Statistical Association* 93 **(no. 443)**, 1078–1087. MR1649202

[15] Neerchal, N. K. and Morel, J. G. (2005). An improved method for the computation of maximum likeliood estimates for multinomial overdispersion models. *Computational Statistics and Data Analysis*, 49 (1), 33–43. MR2129162

[16] Minka, T. P. (2003). Estimating a Dirichlet Distribution, *Technical report*, Microsoft.

[17] Qu, Y., Beck, G. J., and Williams, G. W. (1990). Polya-Eggenberger Distribution: Parameter Estimation and Hypothesis Tests. *Biometrical journal*, 32 (2), 229–242. MR1062736

[18] Rennie, J. D., Shih, L., Teevan, J., and Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In ICML, 3,616–623.

[19] Sen, K. and Bhattacharya, S. (2010). Review of Applications of Urn Models, with Special Emphasis on Polya-Eggenberger Urn Models. *IUP Journal of Computational Mathematics*, 3 (2).

[20] Vaida, F. (2005). Parameter convergence for EM and MM algorithms. *Statistica Sinica*, 15 (3):831–840. MR2233916

[21] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media. MR1367965

[22] Wen, Y., Mukherjee, K., and Ray, A. (2013). Adaptive pattern classification for symbolic dynamic systems. *Signal Processing*, 93 (1), 252–260.

[23] Wu, T. T., and Lange, K. (2010). The MM alternative to EM. *Statistical Science*, 25 (4), 492–505. MR2807766

[24] Yang, L. and Wu, X. Y. (2013). Estimation of Dirichlet process priors with monotone missing data. *Journal of Nonparametric Statistics* 25 (4), 787–807. MR3174297

[25] Yu, P. and Shaw, C. A. (2014). An efficient algorithm for accurate computation of the Dirichlet-multinomial log-likelihood function. *Bioinformatics* 30 (11), 1547–1554.

[26] Zhou, H., and Lange, K. (2010). MM algorithms for some discrete multivariate distributions. *Journal of Computational and Graphical Statistics*, 19 (3), 645–665. MR2732497

Haibin Zhang
School of Statsitics
East China Normal University
500th Dongchuan Rd. Shanghai
the P.R. China
200241
E-mail address: 13248222762@163.com

Xianyi Wu
School of Statsitics
East China Normal University
500th Dongchuan Rd. Shanghai
the P.R. China
200241
E-mail address: xywu@stat.ecnu.edu.cn

Xueqin Zhou
College of science
Shanghai Institute of Technology
Haiquan Road 100, Shanghai
the P.R. China
200241
E-mail address: xueqinzhou@163.com