TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 3
ISSN: 2054 - 7390

Classifying Documents with Poisson Mixtures

Hiroshi Ogura, Hiromi Amano, Masato Kondo
Department of Information Science, Faculty of Arts and Sciences at Fujiyoshida, Showa University, 4562 Kamiyoshida, Fujiyoshidacity, Yamanashi 403-0005, Japan;
ogura@cas.showa-u.ac.jp, kayanm@cas.showa-u.ac.jp, mkondo@nr.showa-u.ac.jp

ABSTRACT

Although the Poisson distribution and two well-known Poisson mixtures (the negative binomial and K-mixture distributions) have been utilized as tools for modeling texts for over last 15 years, the application of these distributions to build generative probabilistic text classifiers has been rarely reported and therefore the available information on applying such models to classification remains fragmentary and even contradictory. In this study, we construct generative probabilistic text classifiers with these three distributions and perform classification experiments on three standard datasets in a uniform manner to examine the performance of the classifiers. The results show that the performance is much better than that of the standard multinomial naive Bayes classifier if the normalization of document length is appropriately taken into account. Furthermore, the results show that, in contrast to our intuitive expectation, the classifier with the Poisson distribution performs best among all the examined classifiers, even though the Poisson model gives a cruder description of term occurrences in real texts than the K-mixture and negative binomial models do. A possible interpretation of the superiority of the Poisson model is given in terms of a trade-off between fit and model complexity.

Keywords: Poisson distribution, Negative binomial distribution, K-mixture distribution, Text classification, Akaike's information criterion, Bayesian information criterion

1 INTRODUCTION

The Poisson distribution is one of the most fundamental discrete distributions for describing the probability of count data (the probability of a given number of events) occurring in a fixed interval of time or space. For text modeling, the Poisson distribution is appropriate for describing the number of occurrences of a certain word in documents of fixed length when the assumption that each word occurs independently holds in an approximate sense. It has been well established, however, that the Poisson model does not fit observation data [1]. The reason for the failure of the Poisson model is that, for most words, the predicted variance, which is

DOI: 10.14738/tmlai.24.388
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.388

equal to the Poisson mean (the expected number of occurrences during the given interval), systematically underestimates the actual variance. Although this imperfect description of word distributions by the Poisson model can be used for keyword selection in information retrieval [2] and for feature selection in text categorization [3-5], improvement of the Poisson model will inevitably be needed in various fields where word distributions are analyzed quantitatively.

As proposed by Church and Gale [1], the description by the usual Poisson distribution can be improved by extension to Poisson mixtures. Here, a Poisson mixture is a probability mass function that is expressed as a sum of finite or infinite Poisson distributions using a certain weighting function. Indeed, the K-mixture [6] and the negative binomial distributions [1], both of which are Poisson mixtures in the sense that they are expressed in the form of infinite superposition of the Poisson distribution, have been found to give a better description of the observed variance in actual documents than that of the usual Poisson, and these Poisson mixtures have been successfully utilized during the last 15 years [7-13].

In spite of the clear success of the K-mixture and negative binomial models for describing word distributions in real texts, attempts to utilize these models to construct generative probabilistic classifiers, however, have rarely been reported. To the best of our knowledge, the main studies on text classifiers using the usual Poisson model and the K-mixture and negative binomial models can be summarized as follows.

- Kim et al. [14, 15] used the Poisson distribution to build a text classifier and showed that their classifier performs much better than the multinomial naive Bayes classifier. However, since their proposed method is a sophisticated one in which additional parameter tuning is required, their classifier is not fully suitable for easy use.

- Eyheramendy et al. [16] compared the performance of four probabilistic models in text classification: the Poisson, Bernoulli, multinomial, and negative binomial models. They found that the multinomial model performs best in terms of the micro-F1 measure, and also that the Poisson and Bernoulli models are very similar in performance and are the second-best choices; the negative binomial model was found to be the worst. In short, their result showed that the usual Poisson and negative binomial models do not outperform the multinomial naive Bayes classifier.

- Airoldi et al. [17, 18] presented statistical models based on the Poisson and negative binomial distributions for text and showed that their models perform better than the widely used multinomial naive Bayes classifier in text classification tasks. The overall behavior of their classifiers indicated that the negative binomial performs best; the Poisson, the second best; and the multinomial, the worst. However, the difference in classification accuracy among the three classifiers examined was sometimes too small to judge which is the best and which is the second best, and therefore was not sufficient to

make a convincing argument that the Poisson and the negative binomial are superior to the multinomial.

The point emerging from this review of the literature is that the information on the application of the Poisson and negative binomial distributions for building generative probabilistic classifiers is still fragmentary and even contradictory. Furthermore, the application of the K-mixture model to text classifiers, which is a widely used Poisson mixture along with the negative binomial distribution, has not yet been reported.

The question motivating this study is whether the multinomial distribution embedded in the most widely used naive Bayes classifiers can be replaced with the usual Poisson, the negative binomial, or the K-mixture. The purpose of this work is therefore to show that these three models are useful tools for describing word distributions in real texts and to show the extent to which the models can be appropriately used in text classification. To determine whether these three models are useful in classification tasks, the accuracy of the proposed classifiers with the three models are examined using three standard datasets. The results lead us to conclude that these classifiers perform much better than the multinomial naive Bayes classifier does, if we construct the three classifiers with appropriate consideration of document length normalization. Another important finding is that, among the three examined classifiers, the classifier with the usual Poisson model performs best, contrary to our intuitive expectation based on the Poisson model giving a cruder description of word distributions in real texts than do the negative binomial and the K-mixture models. The origin of this better performance of the Poisson can be explained in terms of a trade-off between fit and model complexity, as will be presented later.

The rest of this paper is organized as follows. In the next section, we will describe the frameworks of the three models, (i.e., the Poisson, negative binomial, and K-mixture models) for texts, and how to construct classifiers by using these frameworks. Two different methods for normalizing document length are also described in the next section. In Section 3, we summarize our experiments on automatic text classification. Section 4 presents the results of the experiments, and in Section 5, the observed characteristics of the proposed classifiers are discussed. In Section 6, we give our conclusions and suggest directions for future investigation.

2 FORMULATION OF CLASSIFIERS

2.1 Multinomial naive Bayes

First, we briefly review the multinomial naive Bayes and some notation and symbols that will be used later. The framework described here is a standard one [19, chapter 6] and thus we use it as a reference classifier in our experiments.

The multinomial naive Bayes classifier is widely used in text categorization because it can achieve good performance in various tasks and because it is simple enough to be practically

implemented even when the number of features is large. The simplicity is due primarily to the following two assumptions. First, an individual document is assumed to be represented as a vector of word counts (bag-of-words representation). Since this representation greatly simplifies further processing, all three of the generic probabilistic classifiers investigated in this work inherit this first assumption. Next, documents are assumed to be generated by repeatedly drawing words from a fixed multinomial distribution for a given class, and word emissions are thus independent.

From the first assumption, documents can be represented as vectors of count-valued random variables. The ith document in a considered class c is then expressed as

$$d_{ci} = (x_{i1}, x_{i2}, \cdots, x_{ij}, \cdots, x_{i|V|}), \tag{1}$$

where x_{ij} is the count of the jth word t_j in the ith document belonging to class c and $|V|$ is vocabulary size; in other words, we have assumed here that the vocabulary of the considered dataset is given by $V = \{t_1, t_2, \cdots, t_{|V|}\}$ where t_j is the jth word in the vocabulary. From the second assumption, the probability of the document d_{ci} given by vector (1) is

$$p(d_{ci} \mid \theta_c) = \frac{\left(\sum_{j=1}^{|V|} x_{ij}\right)!}{\prod_{j=1}^{|V|}(x_{ij}!)} \prod_{j=1}^{|V|} \theta_{cj}^{x_{ij}}, \tag{2}$$

where θ_{cj} is the probability for the emission of t_j and is subject to the constraints $\sum_{j=1}^{|V|} \theta_{cj} = 1$. Note that for text classification, the parameters θ_{cj} must be evaluated for each possible class c. We use the estimator for θ_{cj} given by

$$\hat{\theta}_{cj} = \frac{1 + \sum_{i=1}^{|D_c|} x_{ij}}{|\ V\ | + \sum_{i=1}^{|D_c|} \sum_{j=1}^{|V|} x_{ij}}, \tag{3}$$

where $|D_c|$ is the number of training documents belonging to the considered class c. To classify a new document with a given feature vector $d = (x_1, x_2, \cdots, x_{|V|})$, the multinomial naive Bayes classifier calculates a class specific probability for class c as

$$p(c|d) \propto p(c)p(d|\theta_c) = p(c) \frac{\left(\sum_{j=1}^{|V|} x_j\right)!}{\prod_{j=1}^{|V|}(x_j!)} \prod_{j=1}^{|V|} \theta_{cj}^{x_j}. \tag{4}$$

Here, $p(c)$ is the prior probability of class c which is estimated from a training set by $p(c) = |D_c|/|D|$ where $|D|$ is the total number of training documents in the used dataset. We estimate θ_{cj} in eq. (4) by using eq. (3) for each specified class c. The document is assigned to the class with highest probability $p(c|d)$. Taking the logarithm of eq. (4) and neglecting class-independent quantities, we obtain the decision function of the multinomial naive Bayes classifier:

$$w(c|d) = \log p(c) + \sum_{j=1}^{|V|} x_j \log \theta_{cj}. \quad (5)$$

The criterion is to assign d to the class c such that eq. (5) is maximized.

2.2 Poisson classifier

A well-known approach to obtaining high-performance generative probabilistic classifiers is to construct classifiers in a hierarchical manner by using conjugate prior/likelihood combinations. Studies following this approach have already been reported for the Dirichlet/multinomial [20], gamma/negative binomial [21], beta/binomial [22], and gamma/Poisson [23] combinations. We have reported that the beta/binomial and gamma/Poisson pairs give classification performance similar to that of support vector machines and clearly surpass that of multinomial naive Bayes classifier [23]. Here, however, we do not deal with such sophisticated hierarchical models and focus our attention toward building simpler classifiers which allow easy and effective implementation similarly to the multinomial naive Bayes classifier. For this reason, we do not employ the formulation of Kim et al. [14, 15] and instead use a simpler formulation that is basically the same as the formulation of Eyheramendy et al. [16] for our Poisson classifier.

Assumptions used to build the Poisson classifier are very similar to those of the multinomial naive Bayes:

1. An individual document is assumed to be represented as a vector of word counts.
2. The probability of the occurrence of a document d is a product of independent terms, each of which represents the probability of the number of emissions (i.e., the count) of an individual word.
3. The probability of the number of emissions is given by the usual Poisson distribution.

From the third assumption, the probability that there are x_{ij} occurrences of word t_j in the ith document belonging to class c is given by the usual Poisson distribution in the following form:

$$p(x_{ij}|c) = \frac{e^{-\lambda_{cj}} \lambda_{cj}^{x_{ij}}}{x_{ij}!}. \quad (6)$$

Here, λ_{cj} is the expected number of occurrences of t_j in a document belonging to class c and is estimated by

$$\hat{\lambda}_{cj} = \frac{C_1 + \sum_{i=1}^{|D_c|} x_{ij}}{C_2 + |D_c|}, \quad (7)$$

where C_1 and C_2 are smoothing parameters to prevent $\hat{\lambda}_{cj}$ from being zero, and $|D_c|$ the number of training documents belonging to class c. Note that the smoothing used in eq. (7) is similar to the Laplace smoothing used in eq. (3). Following [16], we set $C_1 = 0.001$ and $C_2 = 1$.

Combining the second assumption with eq. (6), the conditional probability of the occurrence of a document $d = (x_1, x_2, \cdots, x_{|V|})$ given class c is expressed as

$$p(d|c) = \prod_{j=1}^{|V|} \frac{\lambda_{cj}^{x_j} \exp(-\lambda_{cj})}{x_j!} \propto \prod_{j=1}^{|V|} \lambda_{cj}^{x_j} \exp(-\lambda_{cj}), \tag{8}$$

and thus a class specific probability for class c and the decision function, corresponding to eqs. (4) and (5) of the multinomial case, respectively, are given by

$$p(c|d) = p(c)p(d|c) = p(c) \prod_{j=1}^{|V|} \frac{\lambda_{cj}^{x_j} \exp(-\lambda_{cj})}{x_j!}, \tag{9}$$

$$w(c|d) = \log p(c) + \sum_{j=1}^{|V|} (x_j \log \lambda_{cj} - \lambda_{cj}), \tag{10}$$

for the Poisson classifier. In the training phase, the parameters of the Poisson distributions are evaluated through the estimator, eq. (7), for each possible class and then in the test phase, the classifier assigns the class c that has the highest value of the decision function, eq. (10), to a test document.

2.3 K-mixture classifier

For the K-mixture classifier, the third assumption of the Poisson classifier described above is replaced with the following assumption: "The probability of the number of emissions is given by the K-mixture distribution." The other two assumptions remain in their original forms. The new assumption leads us to the expression of the probability of x_{ij} occurrences of word t_j in the ith document belonging to class c as

$$p(x_{ij}|c) = (1 - \alpha_{cj})\delta_{x_{ij},0} + \frac{\alpha_{cj}}{\beta_{cj} + 1} \left(\frac{\beta_{cj}}{\beta_{cj} + 1} \right)^{x_{ij}}, \tag{11}$$

where α_{cj} and β_{cj} are parameters of the K-mixture distribution satisfying $0 < \alpha_{cj} < 1$ and $0 < \beta_{cj}$, respectively, and the $\delta_{x_{ij},0}$ is Kronecker's delta [1, 6]. Since we used the method of moments to estimate the parameters, the estimators of α_{cj} and β_{cj} are given by

$$\hat{\beta}_{cj} = \frac{1}{2} \left(\frac{\hat{\sigma}_{cj}^2}{\hat{\lambda}_{cj}} + \hat{\lambda}_{cj} - 1 \right), \tag{12}$$

$$\hat{\alpha}_{cj} = \frac{\hat{\lambda}_{cj}}{\beta_{cj}}, \tag{13}$$

where $\hat{\lambda}_{cj}$ is the smoothed sample mean given by eq. (7) and $\hat{\sigma}_{cj}^2$ is the sample variance defined as

$$\hat{\sigma}_{cj}^2 = \frac{1}{|D_c| - 1} \sum_{i=1}^{|D_c|} \left(x_{ij} - \hat{\lambda}_{cj}\right)^2. \tag{14}$$

Equations (12) and (13) can be derived by solving the expressions of mean and variance of the K-mixture given by Church and Gale [1] for α and β.

The second assumption with eq. (11) yields the conditional probability of document $d = (x_1, x_2, \cdots, x_{|V|})$ given class c in the following form:

$$p(d|c) = \prod_{j=1}^{|V|} p(x_j|c) = \prod_{j=1}^{|V|} \left\{ (1 - \alpha_{cj})\delta_{x_j,0} + \frac{\alpha_{cj}}{\beta_{cj} + 1} \left(\frac{\beta_{cj}}{\beta_{cj} + 1} \right)^{x_j} \right\}. \tag{15}$$

Thus we arrive at the decision function:

$$\begin{aligned} w(c|d) &= \log p(c) + \sum_{\{j|x_j=0\}} \log\left(1 - \alpha_{cj} + \frac{\alpha_{cj}}{\beta_{cj} + 1}\right) \\ &+ \sum_{\{j|x_j>0\}} \left\{ \log \alpha_{cj} - (1 + x_j) \log(\beta_{cj} + 1) + x_j \log \beta_{cj} \right\} \end{aligned} \tag{16}$$

The decision of the K-mixture classifier is to assign document d to class c such that eq. (16) is maximized.

2.4 Negative binomial classifier

For the negative binomial classifier, we replace the third assumption with the following statement: "The probability of the number of emissions is given by the negative binomial distribution." The probability of x_{ij} occurrences of word t_j in the ith document belonging to class c can be expressed as

$$P(x_{ij}|c) = \binom{N_{cj} + x_{ij} - 1}{x_{ij}} p_{cj}^{x_{ij}} (1 + p_{cj})^{-N_{cj} - x_{ij}}, \tag{17}$$

where $N_{cj} > 0$ and $p_{cj} > 0$ are parameters of the negative binomial distribution [1]. As in the K-mixture classifier, we used the method of moments to estimate the parameters N_{cj} and p_{cj}, which results in the estimators being expressed in the form:

$$\hat{p}_{cj} = \frac{\hat{\sigma}_{cj}^2}{\hat{\lambda}_{cj}} - 1, \tag{18}$$

$$\widehat{N}_{cj} = \frac{\hat{\lambda}_{cj}}{\hat{p}_{cj}}, \tag{19}$$

where $\hat{\lambda}_{cj}$ is the smoothed sample mean given by eq. (7) and $\hat{\sigma}_{cj}^2$ is the sample variance given by eq. (14). Here, Equations (18) and (19) are obtained by solving the expressions of mean and variance of the negative binomial given by [1] for the parameters.

The probability of the document d belonging to class c is thus calculated by

$$P(d|c) = \prod_{j=1}^{|V|} p(x_j|c) = \prod_{j=1}^{|V|} \left\{ \binom{N_{cj} + x_j - 1}{x_j} p_{cj}^{x_j} (1 + p_{cj})^{-N_{cj} - x_j} \right\}, \tag{20}$$

which is modified to give the decision function of the negative binomial classifier:

$$\begin{aligned} w(c|d) = \log p(c) \\ + \sum_{j=1}^{|V|} \{ \log \Gamma(N_{cj} + x_j) \\ - \log \Gamma(N_{cj}) + x_j \log p_{cj} - (N_{cj} + x_j) \log(1 + p_{cj}) \}. \end{aligned} \tag{21}$$

Note that we have substituted factorials with Gamma functions through the relation $\Gamma(n) = (n-1)!$ and have omitted the term $\log \Gamma(x_j + 1)$ that is independent of class label c and thus not necessary for classification purposes. The substitution of factorials with a gamma function is needed when x_j takes a real, non-integer value, which occurs through the procedures of document length normalization described in the next subsection. The decision of the negative binomial classifier is to assign d to the class c such that eq. (21) is maximized.

2.5 Normalization of document length

Thus far we have neglected the fact that the document lengths in the considered dataset differ from one another. In other words, we have assumed that each document in the dataset has the same length in terms of total word count. Of course, this is not necessarily true. Since the usual Poisson, K-mixture, and negative binomial distributions express the probability of a number of events occurring *in a fixed interval*, it is obvious that some normalization of document length is necessary when we try to apply these models to document classification. To normalize all the different lengths of training documents to be a predefined standard value, we used two different methods: L_1 normalization and pseudo-document normalization.

2.5.1 L_1 normalization

We consider the ith training document in class c: $d_{ci} = (x_{i1}, x_{i2}, \cdots, x_{i|V|})$. The document length of d_{ci} in an L_1 sense is simply given by the total number of occurrences of all terms:

$$l_i = \sum_{j=1}^{|V|} x_{ij}. \tag{22}$$

The normalization of the L_1 norm of document vector d_{ci} to be a predefined standard value of l_0, can be achieved through the conversion of each word count in d_{ci} by using

$$x_{0ij} = w_i x_{ij}, \tag{23}$$

where w_i is the ratio of the actual length l_i to the normalized length l_0; that is,

$$w_i = \frac{l_i}{l_0}. \tag{24}$$

To obtain the parameters of the usual Poisson, K-mixture, and negative binomial models for a normalized dataset in which each length of all the training documents is normalized to be exactly l_0, we use following procedure.

- The smoothed sample mean, eq. (7), is estimated from x_{0ij} given by eq. (23) instead of using the original count value, x_{ij}. We use the notation $\hat{\lambda}_{0cj}$ for the sample mean obtained in this manner, which expresses the sample mean of word occurrences of t_j over all the training documents in class c for the normalized dataset.

- The sample variance, eq. (14), is replaced with that using x_{0ij} and $\hat{\lambda}_{0cj}$, and the resultant variance is denoted as $\hat{\sigma}_{0cj}$, indicating the sample variance of word t_j in class c for the normalized dataset.

- The parameters of the K-mixture distribution for the normalized dataset, denoted by $\hat{\beta}_{0cj}$ and $\hat{\alpha}_{0cj}$, are estimated by eqs. (12) and (13), respectively, by changing $\hat{\lambda}_{cj}$ and $\hat{\sigma}_{cj}$ to $\hat{\lambda}_{0cj}$ and $\hat{\sigma}_{0cj}$.

- The parameters of the negative binomial distribution for the normalized dataset, denoted by $\hat{p}_{0cj}$ and $\hat{N}_{0cj}$, are estimated by eqs. (18) and (19), respectively, by changing $\hat{\lambda}_{cj}$ and $\hat{\sigma}_{cj}$ to $\hat{\lambda}_{0cj}$ and $\hat{\sigma}_{0cj}$.

The procedure for L_1 normalization described above is computationally simpler than the procedure for the pseudo-document normalization presented below.

2.5.2 Pseudo-document normalization

This normalization method is basically the same as proposed by [17] and [18]. In this method, all the training documents belonging to class c are firstly concatenated into a single huge document. The resultant length of this huge document is given by $L = \sum_{i=1}^{|D_c|} l_i$ where l_i is defined by eq. (22) and $|D_c|$ the number of training documents belonging to class c. Then, the huge document is split into equally sized pseudo-documents, each of which has exactly l_0

words. We then regard all the pseudo-documents obtained in this manner as the training set of normalized documents for class c and reconstruct the document vector d_{ci} by counting occurrences of each word in each of the pseudo-documents. Since each of the pseudo-documents has a predefined standard document length l_0, we denote the component of the reconstructed vector as x_{0ij}, which can be used in eqs. (7) and (14) to obtain the sample mean and the variance for normalized dataset without any corrections. ($|D_c|$ in eqs. (7) and (14), the number of training documents belonging to class c, should be reinterpreted as the number of pseudo-documents for this case.) Again, we denote the mean and variance as $\hat{\lambda}_{0ij}$ and $\hat{\sigma}_{0ij}$, respectively.

Estimating parameters of the K-mixture distribution for normalized dataset, $\hat{\beta}_{0cj}$ and $\hat{\alpha}_{0cj}$, and estimating those of the negative binomial distribution, $\hat{p}_{0cj}$ and $\hat{N}_{0cj}$, are also straightforward; explicitly, the estimation of these parameters can be achieved by using eqs. (12), (13), (18), and (19) directly with $\hat{\lambda}_{0ij}$ and $\hat{\sigma}_{0ij}$ obtained from the procedures described above.

2.5.3 Conversion of parameters for non-normalized test document

We consider the case where we try to classify a test document having an actual word count l. It has been shown that if we estimate the distribution parameters of the Poisson, K-mixture, and negative binomial distributions with a normalized dataset in which each document length is normalized to be exactly l_0, then the parameters for the test document having the actual length l should be given as follows [1] :

$$\hat{\lambda}_{cj} = w\,\hat{\lambda}_{0cj}, \quad \text{(Poisson)} \tag{25}$$

$$\hat{\alpha}_{cj} = \hat{\alpha}_{0cj}, \;\; \hat{\beta}_{cj} = w\,\hat{\beta}_{0cj}, \quad \text{(K-mixture)} \tag{26}$$

$$\hat{N}_{cj} = \hat{N}_{0cj}, \;\; \hat{p}_{cj} = w\,\hat{p}_{0cj}, \quad \text{(negative binomial)} \tag{27}$$

where the parameters with subscript 0 on the right-hand side are those estimated for the normalized dataset obtained through the L_1 normalization or the pseudo-document normalization, and the parameters without subscript 0 on the left-hand side are those for the test document having the actual length l. In these equations, w is the ratio of the actual length to the normalized length; that is, $w = \frac{l}{l_0}$.

In the training phase of each classifier, we used one of the two normalization methods described above to obtain the parameters for the normalized dataset, and then in the test phase, eqs. (25)~(27) were used to adjust the parameters to the values suitable for the non-normalized test document.

3 EXPERIMENTAL EVALUATION

To clarify the characteristics of the proposed three classifiers with the usual Poisson, K-mixture, and negative binomial models, we performed text classification experiments using three standard document corpora. In the experiments, the performance of the proposed three classifiers is compared with that of the baseline multinomial naive Bayes classifier.

3.1 Dataset

In our experiments, we chose three different datasets that represent a wide spectrum of text classification tasks.

The first one is the 20 Newsgroups dataset which was originally collected with a netnews-filtering system [24] and contains approximately 20,000 documents that are partitioned nearly evenly across 20 different UseNet newsgroups. We use the 20news-18828 version from which cross-posts have been removed to give a total of 18,828 documents. (Original dataset is available from: http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html. 20News-18828 is available from: http://people.csail.mit.edu/jrennie/20Newsgroups/.) Consequently, 20 Newsgroups is a single-labeled dataset with approximately even class distribution, and the task is to apply one of the 20 possible labels to each test document. We build an initial vocabulary from all words left after stop word, punctuation, and number token removal. Uppercase letters are converted to lowercase letters and no stemming algorithm is applied. Here, words are defined as alphabetical strings enclosed by whitespace. The size of the initial vocabulary is 110,492 words.

The second dataset is SpamAssassin which is available as part of the open-source Apache SpamAssassin Project 2 for public use. (The corpus is available online at http://spamassassin.apache.org/publiccorpus/.) It consists of email divided into three categories: “Easy Ham”, which is email unambiguously ham (i.e., not spam), “Hard Ham” which is not spam but shares many features with spam, and finally “Spam”. The task is to apply these three labels to test emails. We use the latest version of all datasets, and combine “easy ham” and “easy ham 2” datasets to form our Easy Ham dataset; similarly, “spam” and “spam 2” datasets are combined to form our Spam dataset. The preprocessing before building the initial vocabulary was the same as for the 20 Newsgroups. The resulting corpus is just over 6,000 messages with an initial vocabulary of 151,126 words.

The third test collection is the Industry Sector dataset which is a collection of corporate Web pages organized into hierarchical categories based on what a company produces or does. Although it has a hierarchy with three levels of depth, we do not take the hierarchy into account and use a flattened version of the dataset. This dataset contains a total of 9,555 documents divided into 104 categories. (We obtained the dataset from http://www.cs.umass.edu/ mccallum/code-data.html. Because it was found that one of the

original 105 categories was empty, the remaining 104 categories having documents were used in our experiments.) We use all 9,555 documents in our experiments without removing the multi-labeled documents because the fraction of multi-labeled documents is very small and the effect of these documents is negligible. (Only 15 documents out of 9,555 belong to two classes; thus, they cannot affect our results considerably.) The largest and smallest categories have 105 and 27 documents, respectively, and the average number of documents per category is 91.9. For this dataset, we remove HTML tags by skipping all characters between "<" and ">", and we did not use a stop list. The resulting vocabulary has 64,202 words.

For all three datasets, we use 10-fold cross-validation to make maximal use of the data. Ten obtained values of performance are averaged to give the final result.

3.2 Feature selection

To investigate the effect of vocabulary size on classification performance, we use a simple feature selection method based on the collection term frequency as follows. First, we count the collection term frequency, CF, which is the total frequency of each word throughout the entire dataset. Then, we select all words that satisfy $CF \geq N_0$ where N_0 is a predefined integer. The feature selection by CF is one of the simplest methods, but is sufficient for the task at hand, namely, comparing different classifiers at each vocabulary size. The resultant vocabulary sizes after feature selection are summarized in Table 1.

Table 1: Vocabulary size obtained by feature selection with CF.

Feature selection	20 Newsgroups	SpamAssassin	Industry Sector
Initial vocabulary	110,492	151,126	64,202
$CF \geq 2$	64,065	53,886	37,634
$CF \geq 5$	34,124	21,258	21,216
$CF \geq 10$	21,697	12,749	14,317
$CF \geq 20$	13,709	7,754	9,455
$CF \geq 50$	7,314	3,869	5,329
$CF \geq 100$	4,252	2,085	3,233
$CF \geq 200$	2,180	1,077	1,770
$CF \geq 500$	748	402	665
$CF \geq 1000$	255	176	290

Count-valued document vectors $\{d_{ci}\}$ are constructed from document term frequency (number of occurrences of a considered word in a document) for each word in a vocabulary at each vocabulary level. Since we use the 10-fold cross-validation, 1/10 of the original count vectors $\{d_{ci}\}$ are used as test vectors and the rest are used as original training vectors. In the training phase, the original training vectors are supplied to the three classifiers which normalize the training vectors by L_1 normalization or pseudo-document normalization and then estimate

the distribution parameters. In addition to the two normalization methods, the distribution parameters without any normalization are directly calculated from the original training vectors to clarify the effect of document length normalization. To classify test vectors in the test phase, the classifiers use the three types of distribution parameters: those obtained without normalization, those obtained by L_1 normalization and those obtained by pseudo-document normalization.

3.3 Implementation issues

All the classifiers used in this study are implemented in the Java programming language. Supplementary information is as follows:

- For calculating the sample variance, we slightly modified eq. (14) for the following reason. The estimator of p_{cj}, eq. (18), requires $\hat{\sigma}_{cj}^2 > \hat{\lambda}_{cj}$ to satisfy the constraint $\hat{p}_{cj} > 0$. To ensure that the constraint is satisfied, if $\hat{\sigma}_{cj}^2$ calculated by eq. (14) is less than or equal to $\hat{\lambda}_{cj}$, we always replace the original value of $\hat{\sigma}_{cj}^2$ with $\hat{\sigma}_{cj}^2 = \hat{\lambda}_{cj} + \varepsilon$ in which a constant ε is set to 0.1 after a preliminary classification experiment on the 20 Newsgroups dataset. This happens when a considered word t_j fails to appear in any of the training vectors for the considered class. In this case, $\hat{\sigma}_{cj}^2$ calculated by eq. (14) is approximately equal to $\hat{\lambda}_{cj}^2$ and thus much smaller than $\hat{\lambda}_{cj}$.

- In L_1 normalization, we use $l_0 = 1{,}000$ for the normalized document length while in the case of the pseudo-document normalization, we set $l_0 = 100$. The value of $l_0 = 1{,}000$ for L_1 normalization was determined after a preliminary experiment on 20 Newsgroups dataset (we tried $l_0 = 100, 1000, 10000$ and found that $l_0 = 1000$ gives the best classification performance.) , while $l_0 = 100$ for pseudo-document normalization was determined to ensure a sufficient number of pseudo-documents for all categories in the three datasets used.

- To compute the log of the gamma function in eq. (21), components available in the Apache Commons Mathematics Library (http://commons.apache.org/math/) are used.

4 RESULTS

As in our previous study [23], we also use the simplest measure of classification performance in this study, that is, accuracy, which is simply defined as the ratio of the total number of correct decisions to the total number of test documents in the dataset used. Note that for a single-labeled dataset and a single-labeled classification scheme as in this work, the micro-averaged precision and recall are equivalent, and hence equal to the F1 measure [25], which we call "accuracy" here.

4.1 Effect of document normalization

Figures 1, 2, and 3 show the performance of the classifiers for the 20 Newsgroups, SpamAssassin, and Industry Sector datasets, respectively.

In all these figures, the top-left plot (a) shows classification accuracy without normalization, the middle-left plot (b) shows classification accuracy with L_1 normalization, and the bottom-left plot (c) shows classification accuracy with pseudo-document normalization. The plots on the right side in Figs. 1, 2, and 3 (i.e., plots (a'), (b'), and (c')), show the same information as plots (a), (b), and (c), respectively, but with the horizontal axes on a logarithmic scale to show the lower vocabulary region clearly. Note that the accuracies of the multinomial classifier in each figure are identical in all plots (a)~(c'), because L_1 or pseudo-document normalization was only applied to the classifiers using the Poisson, negative binomial, and K-mixture models and was not applied to the multinomial naive Bayes classifier. In Figs. 1, 2, and 3, the accuracy curves of the multinomial classifier in plots (b) and (c), and those in plots (b') and (c') are thus simple replicas of those in plot (a) and plot (a'), respectively.

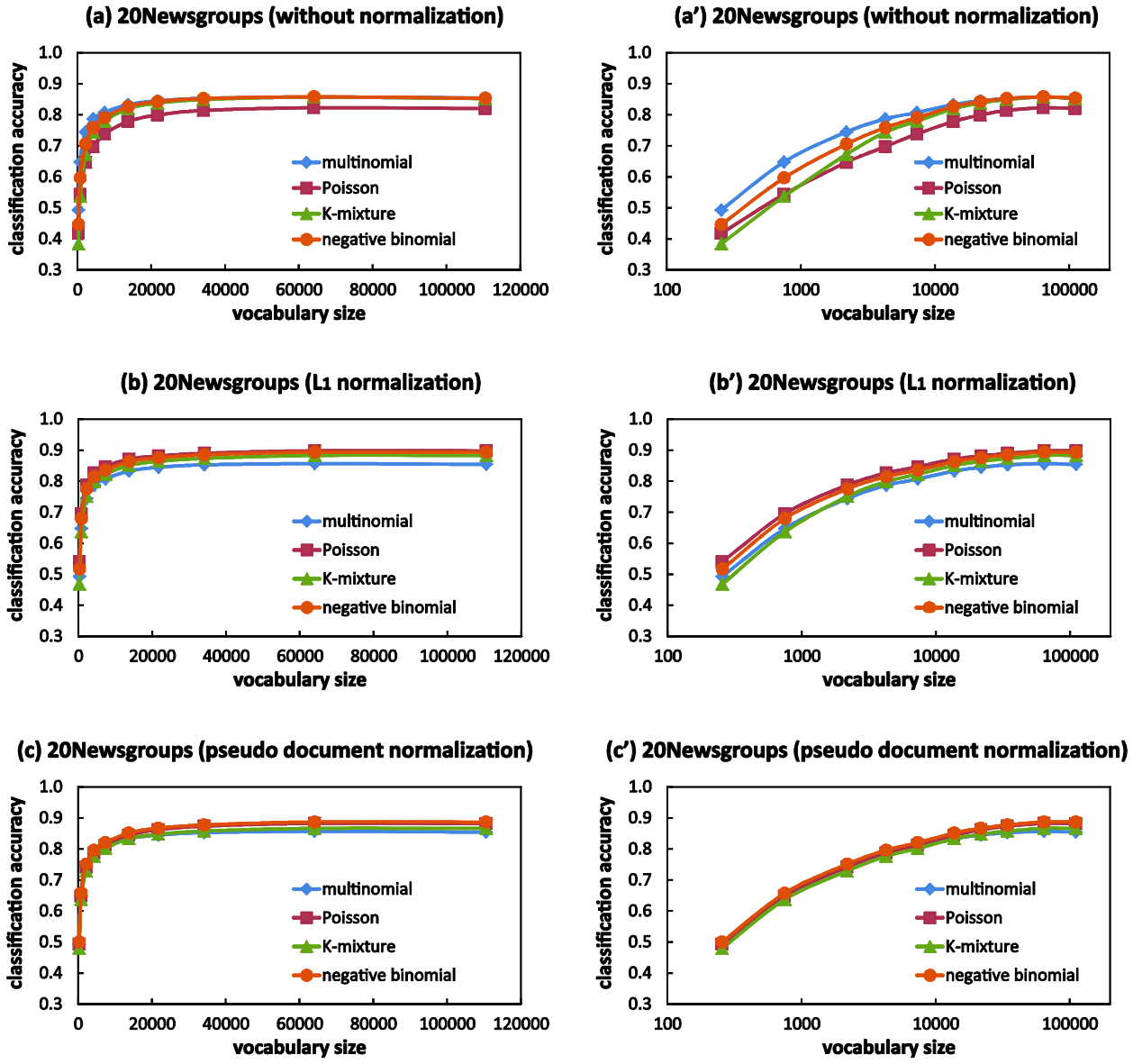

Figure 1: Classification performance of examined four classifiers on 20 Newsgroups dataset. (a) and (a') show the performance without document length normalization; (b) and (b'), with L_1 normalization; and (c) and (c'),

pseudo-document normalization. In (a), (b), and (c), the horizontal axes are linear while they are logarithmic in (a'), (b') and (c') in order to show the lower vocabulary region clearly.

The reason for this special treatment of the multinomial naive Bayes classifier is that the estimated parameter $\hat{\theta}_{cj}$ with eq. (3) for this classifier represents the probability of selecting the word t_j at an arbitrary position of documents in class c *with any arbitrary document length*. Similarly, the probability of the document d_{ci} for the multinomial naive Bayes calculated by use of eq. (2) is valid for documents in class c with any arbitrary document length $\sum_{j=1}^{|V|} x_{ij}$. On the other hand, when we calculate the probability of the document d for the Poisson, K-mixture and negative binomial models by use of eqs. (8), (15) and (20), the document length of d should be normalized to l_0, which is the document length used at estimating parameters. This is because the Poisson distribution describes the probability of count data occurring *in a fixed interval*, and as a consequence, the probability of x_{ij} occurrence of word t_j given by use of eqs. (6), (11) and (17) for the Poisson, K-mixture and negative binomial models are, in a rigorous sense, only valid for the documents with fixed length l_0.

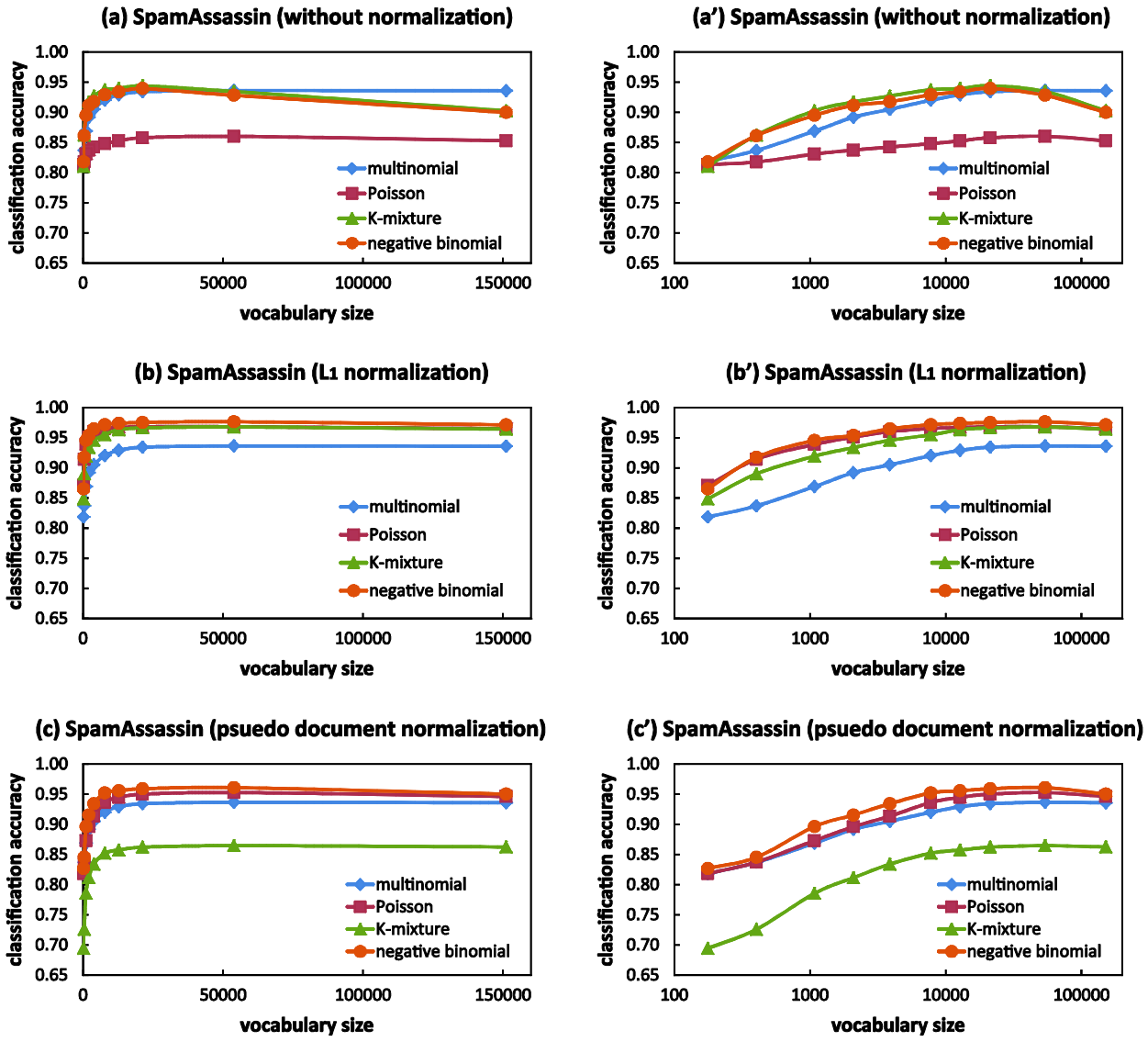

Figure 2: The same as in Fig. 1, but for the SpamAssassin dataset.

Based on the results shown in Figs. 1, 2, and 3, we first consider the effect of document length normalization on classification accuracy. The overall trends of the accuracy curves in these figures clearly indicate that the normalization of document length is fundamentally important to achieve better performance for the classifiers using the Poisson, negative binomial, and K-mixture models. Detailed observations are given below.

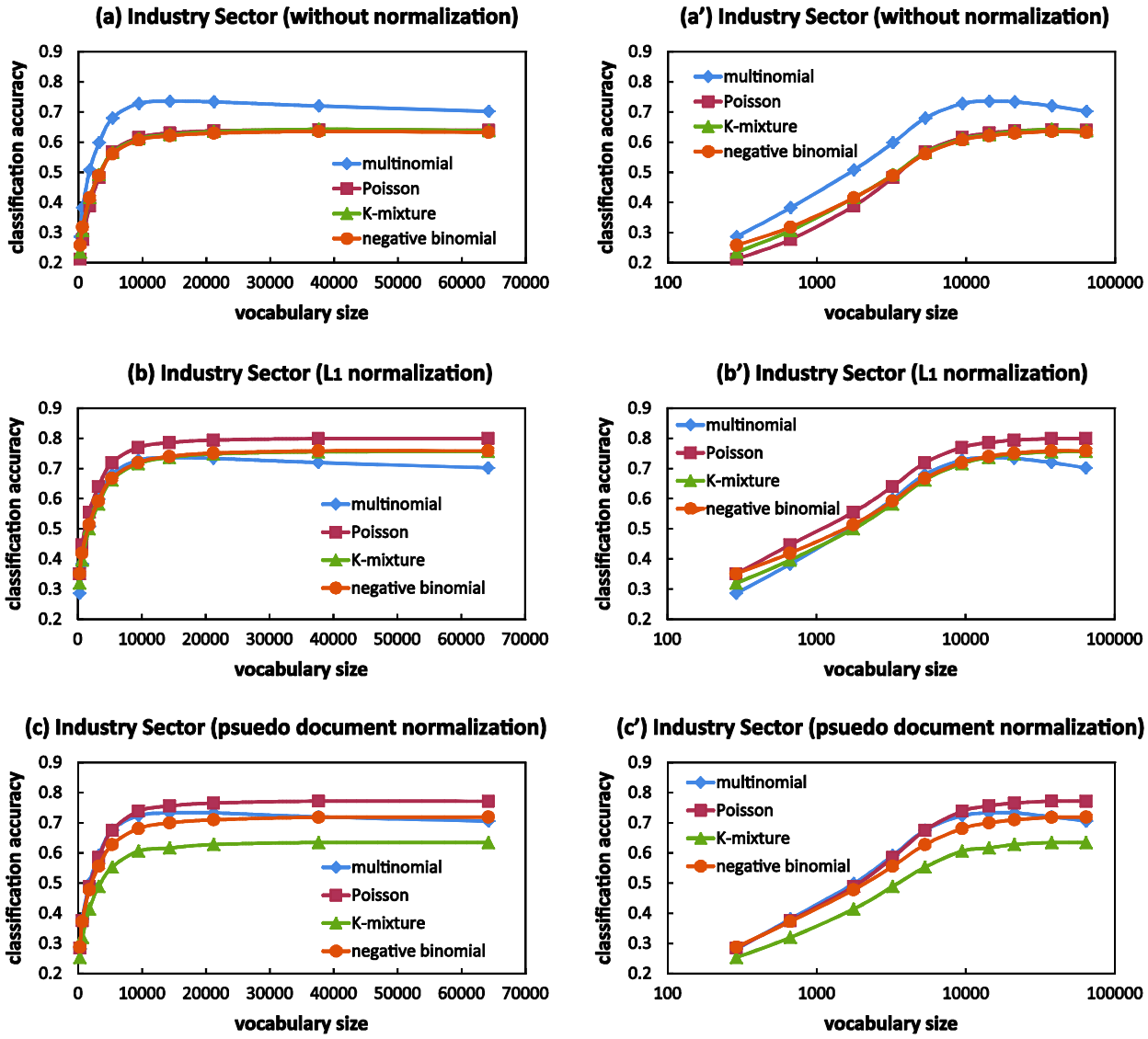

Figure 3: The same as in Fig. 1, but for the Industry Sector dataset.

- For the 20 Newsgroups dataset, it can be seen from Figs. 1(a) and 1(a') that the performance of the K-mixture and negative binomial classifiers without normalization are similar to that of the baseline multinomial classifier especially in the higher vocabulary region and that the Poisson classifier is apparently worse than that of the multinomial classifier for the non-normalized data. On the other hand, Figs. 1(b), (b'), (c), and (c') show that the accuracies of the Poisson, K-mixture, and negative binomial classifiers are higher than that of the multinomial classifier for normalized data.

- For the SpamAssassin dataset, the results are consistently the same or very close to those of the 20 Newsgroups dataset. Figures 2(a) and (a') show that the K-mixture and negative binomial classifiers achieve accuracy similar to that of the multinomial classifier but the Poisson classifier fails to achieve that level of performance for non-normalized data. It is also confirmed from Figs. 2(b), (b'), (c), and (c') that both types of normalization bring better performances to the classifiers using the Poisson, negative binomial, and K-

mixture models than the one using the baseline multinomial model. The K-mixture classifier with pseudo-document normalization is the only exception and performs worse than the multinomial classifier does (Figs. 2(c) and (c')).

- The influence of document length normalization on classification accuracy is most evident for the Industry Sector dataset as seen in Fig. 3. Compared with the multinomial classifier, the Poisson, negative binomial, and K-mixture classifiers appear to give worse performance for non-normalized data (Figs. 3(a) and (a')), while they perform better than the multinomial classifier does for normalized data especially in the higher vocabulary region (Figs. 3(b), (b'), (c), and (c')). Again, the K-mixture classifier with pseudo-document normalization (Figs. 3 (c) and (c')) is the only exception and exhibits much worse performance than the multinomial classifier.

In short, Figs. 1~3 show that the document length normalization is effective for improving the performance with the Poisson, negative binomial, and K-mixture models. Although the degree of improvement differs by dataset, normalization typically achieves much better performance than that of the baseline multinomial classifier, in contrast to the case that non-normalized data are used.

In a comparison between the two different types of normalization, L_1 normalization seems to bring about better improvement than pseudo-document normalization does, and this tendency is clearest for the K-mixture classifier, as seen in plots (b') and (c') of Figs. 2 and 3. The difference in the degree of improvement between the two normalization methods can be ascribed to the following reason. In the pseudo-document normalization, each term occurrence is treated as being equally important while in the L_1 normalization, the event of a word occurrence has remarkably different weight according to the original document length. This is because, in the L_1 normalization, the occurrence of a word in a short document more heavily weighted than the occurrence of the same word in a long document. The conversion of the L_1 normalization in this manner is reasonable and considered to bring about the better performance because, compared with long documents, short documents usually have fewer unnecessary terms that are irrelevant to the topic and the ratio of informative terms that represent a concept of the topic is higher.

4.2 Comparative performance behavior

In this subsection, we compare the four classifiers in term of classification performance. First, we consider the performance in the non-normalized case. Clearly, the multinomial classifier is the best performer when we use non-normalized data, as is clearly exhibited in plot (a') of Figs. 1, 2, and 3. As for the other three classifiers in the non-normalized case, the negative binomial and K-mixture classifiers give similar performance and are superior to the Poisson classifier. Indeed, the negative binomial and K-mixture classifiers perform similarly to the multinomial classifier and apparently better than the Poisson classifier (Figs. 1(a), 1(a'), 2(a), and 2(a')). The superiority of the negative binomial and K-mixture classifiers over the Poisson classifier for non-normalized data can be attributed to their flexibility in modeling text because

they can describe the overdispersion of word frequency which is often encountered in real texts but is not modeled well by the Poisson distribution.

We next consider the cases where normalized data are used. The overall trends in the accuracy curves suggest that the Poisson, negative binomial, and K-mixture models achieve much better performance than the multinomial model for normalized data, as described in the previous subsection. To examine which model is best for normalized data, we further compare the three classifiers except the multinomial classifier on the basis of the results shown in Figs. 1∼3. Observation of the six cases (3 datasets × two normalization methods) shows that the usual Poisson classifier performs best in two of six cases (Figs. 3(b') and 3(c')), the negative binomial performs best in one case (Fig 2(c')), and they perform similarly in the other three cases (Figs. 1(b'), 1(c'), and 2(b')). Therefore, the usual Poisson classifier appears to be the best performer, the negative binomial classifier is found to perform at a similarly high level, and compared with the K-mixture classifier, they are better for normalized data. These results raise the question as to why the usual Poisson model, which performs poorly for non-normalized data as described above, performs well for normalized data. In the next section, this question is discussed by considering the trade-off between fit and model complexity.

5 DISCUSSION

To examine the behavior of the usual Poisson, negative binomial, and K-mixture models for normalized datasets from a different perspective, we attempt to calculate the two most commonly used penalized model selection criteria, the Akaike's information criterion (AIC) and the Bayesian information criterion (BIC), for the three datasets and we investigate the relation between these criteria and classification performance. In general, penalized model selection criteria are statistics of the form [26, 27]

$$-2L(\hat{\theta}_M) + kp_M, \tag{28}$$

where $\hat{\theta}_M$ is a parameters vector obtained by maximum likelihood estimation (MLE), $L(\hat{\theta}_M)$ is the log-likelihood, k is a known positive quantity, and p_M is the number of parameters in the model under consideration. The maximized log-likelihood obtained through MLE, that is, $L(\hat{\theta}_M)$ in the first term of eq. (28), reflects the fit of the considered model to the observed data, while p_M in the second term is regarded as a measure of the complexity of a considered model. The second term penalizes models for the number of parameters used. The two terms of eq. (28) thus pull in opposite directions, apparently expressing a trade-off between fit and model complexity. The penalized model selection criteria are intended to help select the best model from among several competing models, that is, a value of a criterion is calculated for each model under consideration, and the model with the smallest value is chosen as the best one.

The two most commonly used penalized model selection criteria, the AIC and BIC, are defined as [27-29]

$$AIC = -2L(\hat{\theta}_M) + 2p_M, \tag{29}$$

$$BIC = -2L(\hat{\theta}_M) + p_M \log n, \tag{30}$$

where n is the number of observations. The difference between their penalty terms seen in eqs. (29) and (30) arises from their different foundations [27]. In the following, we calculate AIC and BIC for each word t_j for each class c in the considered dataset by using

$$AIC_M(cj) = -2\sum_{i=1}^{|D_c|} \log p_M(x_{ij}|c) + 2p_M, \tag{31}$$

$$BIC_M(cj) = -2\sum_{i=1}^{|D_c|} \log p_M(x_{ij}|c) + p_M \log|D_c| \tag{32}$$

where $|D_c|$ is the number of documents belonging to considered class c, and the subscript M specifies the model and indicates Poisson, negative binomial, or K-mixture. In eqs. (31) and (32), $p_M(x_{ij}|c)$ means the model-dependent probability of x_{ij} occurrences of word t_j in the ith document of class c, and is given by eq. (6) for the Poisson, by eq. (11) for the K-mixture, and by eq. (17) for the negative binomial, respectively. Note that the parameters for each distribution model (i.e., $\hat{\lambda}_{cj}$ for the Poisson, $\hat{\alpha}_{cj}$ and $\hat{\beta}_{cj}$ for the K-mixture, and $\hat{N}_{cj}$ and $\hat{p}_{cj}$ for the negative binomial) are estimated from all the documents belonging to class c in the considered dataset and are used to calculate $p_M(x_{ij}|c)$ for each model. We used the method of moments described in the previous section to estimate distribution parameters. (Parameters obtained by the method of moments do not coincide with those obtained by MLE in a strict sense. However, our experiences showed that an iterative calculation to obtain the MLE solution for the negative binomial parameters, which is given by [1], is not stable and can be easily affected by outliers. A similar trend was observed for MLE of the K-mixture parameters, and thus, we used the method of moments which offers more robust estimations.) Also, at the estimation of parameters and at the calculation of $p_M(x_{ij}|c)$, all the documents are normalized by L_1 normalization with $l_0 = 1000$. The number of parameters for each model, p_M, is set to be $p_P = 1$ (Poisson), $p_K = 2$ (K-mixture) and $p_{NB} = 2$ (negative binomial). To compare the AIC and BIC values with the classification performances, a further step of averaging AIC and BIC over all categories is needed because the classification performance was obtained from entire documents of the considered dataset and thus reflect averaged classification accuracy over all categories. The averaged AIC and BIC for each word t_j in the vocabulary are obtained through

$$AIC(\text{Poisson / K} - \text{mixture / negative binomial}, t_j) = \frac{1}{|C|}\sum_{c=1}^{|C|} AIC_M(cj), \tag{33}$$

$$BIC\big(\text{Poisson / K} - \text{mixture / negative binomial}, t_j\big) = \frac{1}{|C|}\sum_{c=1}^{|C|} AIC_M(cj), \tag{34}$$

where $|C|$ is the number of classes in the considered dataset, and $AIC_M(cj)$ and $BIC_M(cj)$ are the AIC and BIC of word t_j for class c as given by eqs. (31) and (32), respectively.

Figures 4, 5, and 6 show the scatter plots between two of three text models in terms of AIC and BIC for the 20 Newsgroups dataset, SpamAssassin dataset, and Industry Sector dataset, respectively. One data point in each plot of these figures corresponds to a word in the vocabulary of the considered dataset; we calculated the AIC or BIC for two different models by using eq. (33) or (34) and these values were used as the x- and y-coordinates of the data point.

From overall trends in AIC and BIC depicted in Figs. 4, 5, and 6, we can find that an arbitrary pair among the three models which has a strong positive correlation with each other in terms of AIC and BIC. Another finding is that AIC and BIC behave fundamentally the same. This can be explained from eqs. (31) and (32) by noting that the number of parameters, p_P, p_K, and p_{NB}, and the number of documents, $|D_c|$, are common for all words in a given class of a dataset under consideration, and hence the difference between AIC and BIC is always a common constant for all words.

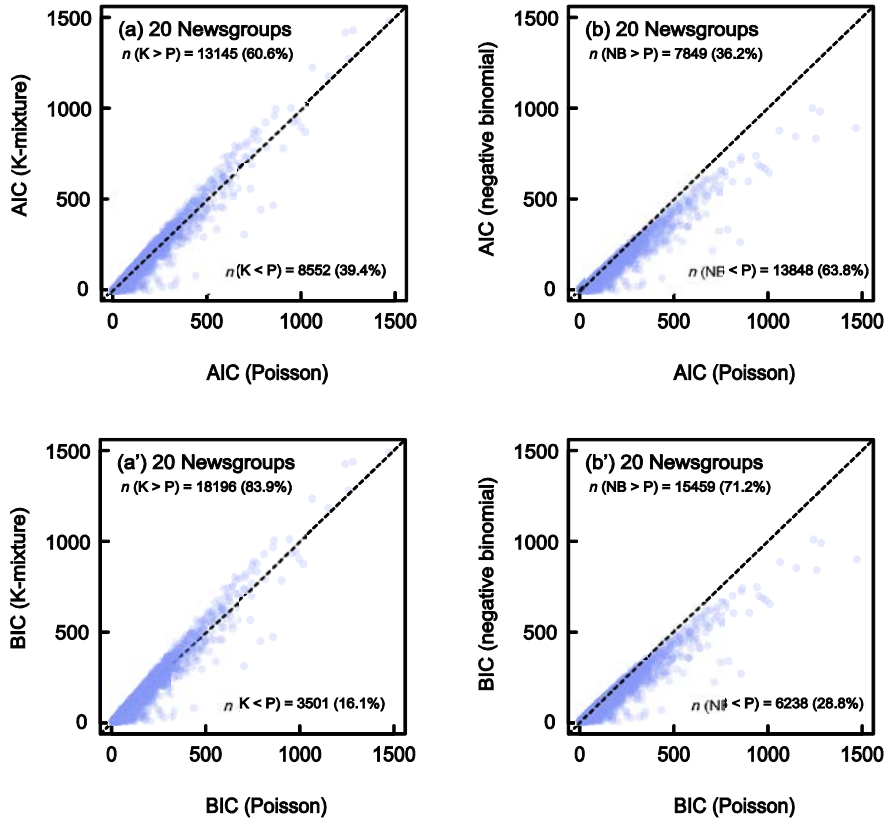

Figure 4: Scatter plots between two of three text models in terms of AIC and BIC for the 20 Newsgroups dataset. (a) shows the correlation between the Poisson model and the K-mixture model in terms of AIC, and (b)

shows that between the Poisson model and the negative binomial model. (a') and (b') are the same as (a) and (b), respectively, but show BIC. In all cases, the vocabulary level used is $CF > 10$ (21,697 words).

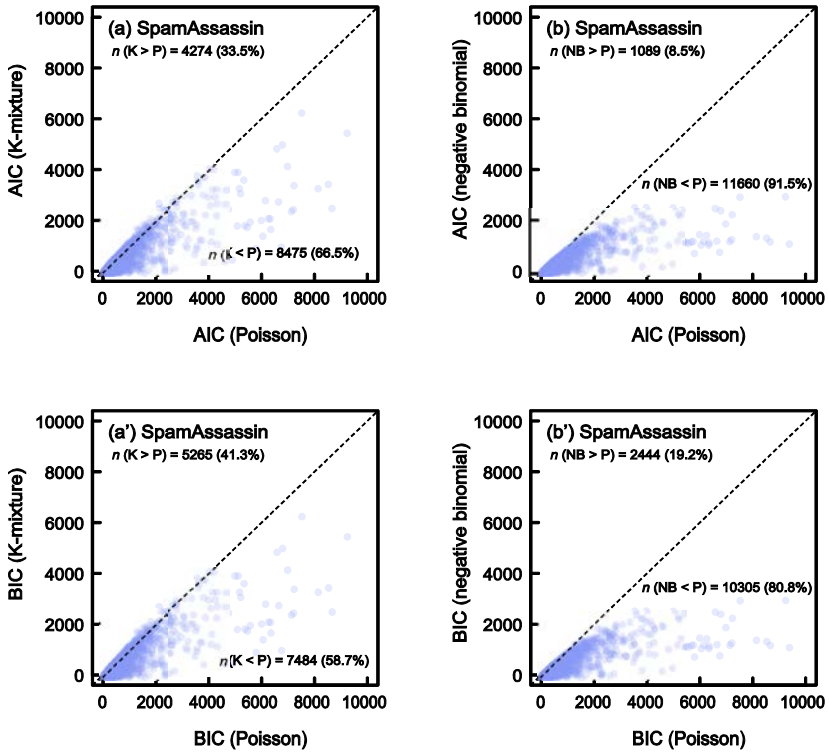

Figure 5: The same as in Fig. 4, but for the SpamAssassin dataset. In all cases (a)~(b'), the vocabulary level used is $CF > 10$ (12,749 words).

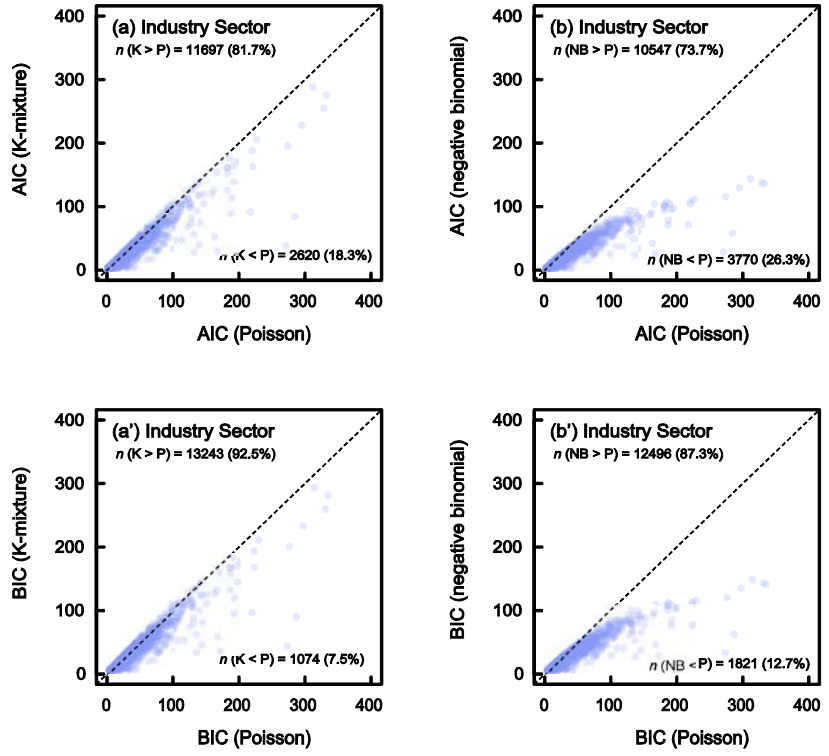

Figure 6: The same as in Fig. 4, but for the Industry Sector dataset. In all cases (a)~(b'), the vocabulary level used is $CF > 10$ (14,317 words).

Dotted diagonal lines in each plot of Figs. 4, 5, and 6 represent the function $y = x$ and the numbers of data points satisfying $y > x$ and $y < x$ are depicted in each plot. For example, in Fig. 4(b'), the x and y axes represent the BIC of the Poisson model and that of the negative binomial model, respectively, and the number of words satisfying $BIC(\text{negative binomial}) > BIC(\text{Poisson})$ located in the upper half-plane over the dotted line in the plot area is 'n(NB>P) = 15459 (71.2%)' and the number of data points satisfying $BIC(\text{negative binomial}) < BIC(\text{Poisson})$ located in the lower half-plane is 'n(NB < P) = 6238 (28.8%)'. The total of these two data points, 21697, gives the vocabulary of the 20 Newsgroups dataset chosen under the condition that $CF > 10$, as Table 1 shows. At first glance, the number of data points in the upper half-plane in Fig. 4(b') seems to be much smaller than that in the lower half-plane; however, a closer look indicates that the data points located in the upper half-plane are much more densely plotted and hence there are more points in the upper half-plane.

To compare the three models in terms of AIC and BIC, we tentatively use the numbers of data points in the upper and the lower half-planes. For example, from Fig. 4(b'), we find that the relation $BIC(\text{negative binomial}) > BIC(\text{Poisson})$ holds for about 70% words in the vocabulary and therefore we can conclude that the Poisson model is more suitable than the negative binomial for the 20 Newsgroups dataset because the former gives smaller BICs for most words. We have made similar comparisons for each scatter plot in Figs. 4, 5, and 6 and have found that the comparisons in terms of BICs are qualitatively consistent with the classification accuracy described in the previous section. Table 2 summarizes our comparisons of three text models in terms of BIC. In the statements of comparison of text models in Table 2, for example, 'NB > K-mixture≳ Poisson' for the SpamAssassin dataset, NB means negative binomial and the symbols '>' and '≳' should be read as "better than" and "better than or equivalent to", respectively. We have tentatively used the following evaluation criteria to compare the models. For comparing the K-mixture and Poisson models, if the percentage of n(K > P) is larger than 60% we conclude that 'Poisson > K-mixture', or if the percentage is less than 40% we conclude 'K-mixture > Poisson', and otherwise (the percentage is in the range of 40%~60%) we conclude that 'Poisson ≳ K-mixture' or 'K-mixture ≳ Poisson'. The same evaluation criteria have been used for comparing the negative binomial and Poisson models. For comparing the K-mixture and negative binomial models, if the difference in the ratio between n(K > P) and n(NB > P) is more than 10%, then we conclude that one is better than the other; otherwise we conclude that one is better than or equivalent to the other. The rather loose criteria with a 20% range of tolerance described above arise from a consideration that the absolute values of the ratios n(K > P) and n(NB > P) have a degree of uncertainty. The existence of this uncertainty is intuitively recognized from the fact that the ratio of, for example, n(K>P), calculated with AIC and that with BIC take different values by amount up to 20%. Since we do not have enough evidence to prove that either the AIC or BIC is better than

the other, the ratio of n(K>P) should be considered to have the same degree of uncertainty. The comparisons of three text models are derived from the following considerations:

- For 20 Newsgroups, n(K > P)=83.9% indicates that the Poisson model is better than the K-mixture model, n(NB > P)=71.9% indicates that the Poisson model is also better than the negative binomial model, and the comparison of n(K > P)=83.9% and n(NB > P)=71.9% leads us to conclude that the negative binomial model is better than the K-mixture model. Thus the results are summarized as ‘Poisson > NB > K-mixture’ as shown in Table 2.

- For SpamAssassin, n(K < P)=58.7% indicates that the K-mixture model is better than or equivalent to the Poisson model, n(NB < P)=80.8% shows that the negative binomial model is better than the Poisson model. Thus the results are summarized as ‘NB > K-mixture ≳Poisson’.

- For Industry Sector, n(K > P)=92.5% indicates that the Poisson model is better than the K-mixture model, n(NB > P)=87.3% also means that the Poisson model is better than the negative binomial model, and the comparison of n(K > P)=92.5% and n(NB > P)=87.3% leads us to conclude that the negative binomial model is better than or equivalent to the K-mixture model. The results are summarized as ‘Poisson > NB ≳ K-mixture’.

Table 2: The percentages of words in the upper and lower half-planes in the scatter plots of BICs (plot (a') and (b') in Figs. 4, 5, and 6) and comparison of text models derived from the numbers of the data points.

dataset	K-mixture vs. Poisson		NB vs. Poisson		comparison of text models
	n(K > P)	n(K < P)	n(NB > P)	n(NB < P)	
20 Newsgroups	83.9%	16.1%	71.2%	28.8%	Poisson > NB > K-mixture
Spam Assassin	41.3%	58.7%	19.2%	80.8%	NB > K-mixture ≳ Poisson
Industry Sector	92.5%	7.5%	87.3%	12.7%	Poisson > NB ≳ K-mixture

Table 3 shows the classification performance corresponding to Table 2 and the comparisons of models in terms of classification accuracy. As seen in Tables 2 and 3, the result of comparing the three models in terms of BIC (Table 2) and that in terms of classification accuracy (Table 3) are reasonably consistent with each other, and the slight discrepancy is only that ‘K-mixture ≳ Poisson’ for the SpamAssassin dataset in Table 2 is replaced with ‘Poisson ≳ K-mixture’ in Table 3. Of course, we can consider that this discrepancy is not a fundamental difference because there is some uncertainty in n(K>P) and therefore the comparison results in Tables 2 and 3 are fundamentally the same.

Table 3: Comparisons of the three models in terms of classification accuracy. For each dataset, the vocabulary used is determined by the condition *CF*>10 and all the document vectors are normalized to l_0=1000 by L_1 normalization. Values are shown as accuracy±σ where σ is the standard deviation calculated through 10-fold cross-validation.

dataset	classification accuracy			comparison of text models
	Poisson	K-mixture	NB	
20 Newsgroups	88.11±0.61%	86.38±0.58%	87.51±0.62%	Poisson > NB > K-mixture
Spam Assassin	96.70±0.84%	96.33±0.63%	97.38±0.76%	NB > Poisson ≳K-mixture
Industry Sector	78.58±1.43%	73.63±1.30%	73.98±1.37%	Poisson > NB ≳ K-mixture

We now consider the meaning of the consistency between Tables 2 and 3 described above. By definition, an information criterion such as AIC and BIC having the form of eq. (28) indicates that given a finite quantity of data available for modeling, a model with a higher degree of freedom will have greater instability, resulting in reduced prediction ability [27]. The situation is very similar in classification tasks [30, chapter 4]. If we try to build a classification model that fits the training data too well in order to lower the training error, then the generalization error in classifying unknown test data becomes larger due to overfitting. In this sense, the log-likelihood in eq. (28) corresponds to the degree of fitting in the training phase of classification and that represents how well the classification model fits the training data, and the second term of eq. (28) corresponds to the penalty for overfitting that will lead to misclassification in the test phase. The consistency between Tables 2 and 3 allows for an intuitive interpretation that both the degree of positive influence due to maximizing the log-likelihood with a precise description of the word distribution estimated through BIC and the degree of negative influence due to overfitting also estimated through BIC agree well with the actual trends, i.e., the actual amelioration and the deterioration in classification accuracy for the three models used.

We next consider the reason why the usual Poisson model performs better for a normalized dataset than the K-mixture and negative binomial models do while it behaves worst among the three models for a non-normalized dataset. Figures 7, 8, and 9 show the scatter plots between the mean and variance for each word in a vocabulary for the 20 Newsgroups, SpamAssassin, and Industry Sector datasets, respectively. The plots labeled (a) in these figures show the case for non-normalized datasets and the plots labeled (b) show for the normalized cases using L_1 normalization with $l_0 = 1000$. Comparing with plots (a) and (b), the positive correlation between the mean and variance appears to become stronger in the case of the normalized datasets compared with non-normalized datasets. In other words, when we want to describe the distribution of each word precisely, the mean and variance of each word are necessary in the case of a non-normalized dataset, while specifying both of these two values seems excessive in the case of a normalized dataset because these two values have strong positive correlation as seen in the plots labeled (b). This finding from Figs. 7, 8, and 9 can be

reinterpreted in terms of the number of model parameters needed to describe the word distribution because the mean and variance can be directly converted to $\hat{\alpha}_{cj}$ and $\hat{\beta}_{cj}$ for the K-mixture model and to $\hat{p}_{cj}$and $\widehat{N}_{cj}$ for the negative binomial model. From the finding described above, we can deduce a possible explanation of the good performance of the usual Poisson models for normalized datasets as follows. For non-normalized data, specifying two parameters in the distribution model is necessary to obtain a large value of log-likelihood in eq. (28) and thus the models having two parameters, (i.e., the K-mixture and negative binomial models), are advantageous over the Poisson model for non-normalized data. On the other hand, using two parameters to describe the word distribution in normalized datasets is expensive in the sense that the effect of decreasing the information criteria by maximizing the log-likelihood with two parameters is restrictive and thus the influence of the penalty term becomes larger for normalized data. This situation makes the one-parameter Poisson model superior to the two-parameter K-mixture and negative binomial models in the case of a normalized dataset.

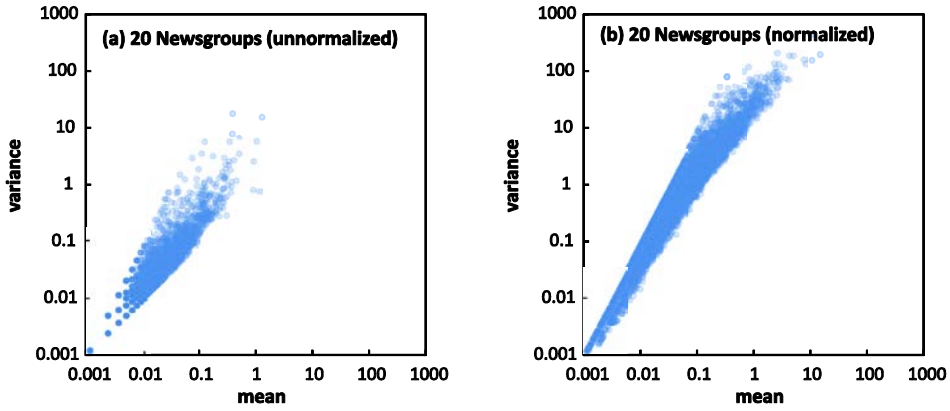

Figure 7: Scatter plots between the mean and variance of each word in a vocabulary for the 20 Newsgroups dataset. (a) shows the case of non-normalized data and (b) shows the case of normalized data by use of L_1 normalization with l_0=1000. The vocabulary level used is *CF*>10 (21,697 words) and the means and variances are calculated from all the documents belonging to the 'alt.atheism' category (the first category in alphabetical order) from the dataset.

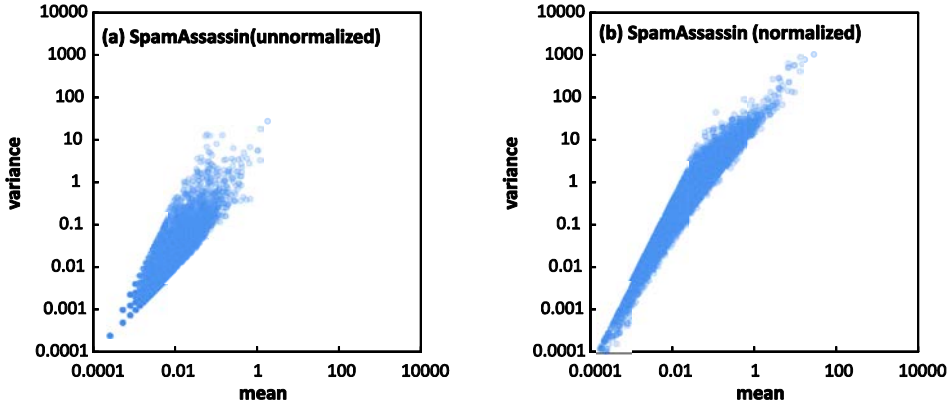

Figure 8: The same as in Fig. 7, but for the SpamAssassin dataset. The vocabulary level used is *CF*>10 (12,749 words) and all the documents belonging to 'Easy Ham' (the first category in alphabetical order) are used to calculate the mean and the variance of each word.

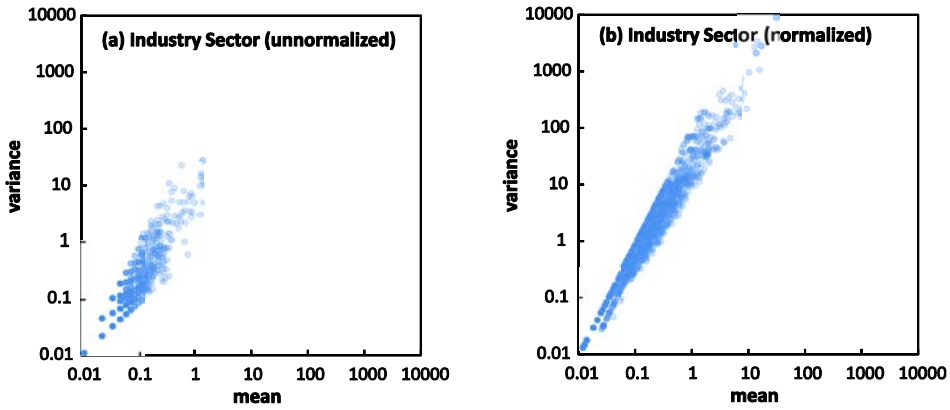

Figure 9: The same as in Fig. 7, but for the Industry Sector dataset. Vocabulary level used is $CF > 10$ (14,317 words) and all the documents belonging to 'accident.and.health.insurance.industry' (the first category in alphabetical order) are used to calculate the mean and variance of each word.

6 CONCLUSION

The usual Poisson distribution and two well-known Poisson mixtures (the K-mixture and negative binomial distributions) have been utilized to build three types of generative probabilistic text classifiers. The classifier frameworks and assumptions used in constructing the classifiers were demonstrated with practical techniques for parameter estimation and document length normalization. The performance of the proposed classifiers was examined through experiments on automatic text categorization of the 20 Newsgroups, SpamAssassin, and Industry Sector datasets. For comparison, a classifier using the multinomial distribution (i.e., the standard multinomial naive Bayes classifier) was also applied to the same datasets.

The results showed that, in the case of non-normalized datasets in which each document length is different from the others, the multinomial naive Bayes classifier performs best but that the classifiers with the K-mixture and negative binomial distributions perform similarly to the multinomial naive Bayes classifier; the Poisson classifier performs worst. On the other hand, the results for normalized datasets, in which each document is normalized to exactly the same length, showed that the three classifiers with the usual Poisson, K-mixture, and negative binomial distributions perform much better than the multinomial naive Bayes classifier. It was also shown from the results for the normalized datasets that the classifier with the Poisson distribution performs best among all the examined classifiers, even though the Poisson model gives a cruder description of term occurrence in real texts than the K-mixture and negative binomial models do.

The origin of the superiority of the Poisson classifier for normalized datasets was discussed in terms of a trade-off between fit and model complexity. Through the discussion, we found that the Bayesian information criterion, which is one of the widely used information criteria,

can qualitatively give a reasonable description of model suitability that is consistent with the classification accuracy of the examined classifiers.

At present, understanding of the relation between the information criteria and the actual classification performance is limited, although our results indicate a strong correlation. We consider that further quantitative analysis is needed before reaching a final conclusion, and such an investigation is planned for future research.

We thank Dr. Yusuke Higuchi for useful discussion and illuminating suggestions. This work was supported in part by JSPS Grant-in-Aid (Grant No. 25589003).

REFERENCES

[1]. K. Church, W. A. Gale, Poisson Mixtures, Natural Language Engineering 1 (1995) 163--190.

[2]. K. Church, W. A. Gale, Inverse Document Frequency (IDF): A Measure of Deviations from Poisson, in: Proceedings of the Third Workshop on Very Large Corpora, 1995, pp. 121--130.

[3]. H. Ogura, H. Amano, M. Kondo, Feature selection with a measure of deviations from Poisson in text categorization, Expert Systems with Applications 36 (2009) 6826--6832.

[4]. H. Ogura, H. Amano, M. Kondo, . Distinctive characteristics of a metric using deviations from Poisson for feature selection, Expert Systems with Applications 37 (2010) 2273--2281.

[5]. H. Ogura, H. Amano, M. Kondo, Comparison of metrics for feature selection in imbalanced text classification, Expert Systems with Applications 38 (2011) 4978--4989.

[6]. S. Katz, Distribution of content words and phrases in text and language modelling, Natural Language Engineering 2 (1996), 15--59.

[7]. J. Gao, M. Li, K. Lee, N-gram distribution based language model adaptation, in: ICSLP2000 Proceedings of International Conference on Spoken Language Processing, 2000, pp. 497--500.

[8]. J. Gao, K. Lee, Distribution-based pruning of backoff language models. in: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000, pp. 579--588.

[9]. Y. Gotoh, S. Renals, Variable Word Rate N-grams, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 3, 2000, pp. 1591--1594.

[10]. M. Jansche, Parametric Models of Linguistic Count Data, in: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 2003, pp. 288--295.

[11]. M. Saravanan, S. Raman, B. Ravindran, A Probabilistic Approach to Multi-document Summarization for Gererating a Tiled Summary, in: ICCIMA '05 Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications, 2005, pp. 167--172.

[12]. M. Saravanan, B. Ravindran, S. Raman, Improving Legal Document Summarization Using Graphical Models, in: Proceedings of the 2006 conference on Legal Knowledge and Information Systems: JURIX 2006: The Nineteenth Annual Conference, 2006, pp. 51--60.

[13]. H. Pande, H. S. Dhami, Distributions of different parts of speech in different parts of a text and in different texts, The modern journal of applied linguistics, ISSN 0974-8741 (2010) 152--170.

[14]. S. Kim, H. Seo, H. Rim, Poisson Naive Bayes for Text Classification with Feature Weighting. in: International Workshop on Information Retrieval with Asian Languages, 2003, pp. 33-40.

[15]. S. Kim, K. Han, H. Rim, H. Myaeng, Some effective techniques for naive Bayes text classification, IEEE transactions on knowledge and data engineering 18 (2006) 1457--1466.

[16]. S. Eyheramendy, D. Lewis, D. Madigam, On the Naive Bayes Model for Text Categorization, in: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003, pp. 332--339.

[17]. E. M. Airoldi, W. Cohen, S. E. Fienberg, Statistical Models for Frequent Terms in Text, Tech. Report CMU-CALD-04-106, School of Computer Science, Carnegie Mellon Univ. (2004).

[18]. E. M. Airoldi, A. G. Anderson, S. E. Fienberg, K. K. Skinner, Who Wrote Ronald Reagan's Radio Addresses?, Bayesian Analysis 1 (2006) 289--320.

[19]. T. Mitchell, Machine Learning, McGraw Hill, 1997.

[20]. R. E. Madsen, D. Kauchak, C. Elkan, Modeling word burstiness using the Dirichlet distribution, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 545--552.

[21]. S. Clinchant, E. Gaussier, The BNB distribution for text modeling, in: Advances in Information Retrieval. 30th European Conference on IR Research, 2008, pp. 150--161.

[22]. B. Allison, An improved hierarchical Bayesian Model of Language for document classification, in: Proceedings of the 22nd International conference on computational linguistics, 2008, pp. 25--32.

[23]. H. Ogura, H. Amano, M. Kondo, Gamma-Poisson Distribution Model for Text Categorization, ISRN Artificial Intelligence Vol. 2013, (2013) Article ID 829630, http://dx.doi.org/10.1155/2013/829630.

[24]. K. Lang, NewsWeeder: Learning to Filter Netnews, in: Proceedings of the 12th International Machine Learning Conference, 1995, pp. 331--339, Morgan Kaufmann.

[25]. N. Slonim, G. Bejerano, S. Fine, N. Tishby, Discriminative feature selection via multiclass variable memory Markov model, Journal on Applied Signal Processing, 2 (2003) 93--102.

[26]. J. Kuha, AIC and BIC - Comparisons of Assumptions and Performance, Sociological Methods and Research, 33 (2004) 188--229.

[27]. S. Konishi, G. Kitagawa, Information Criteria and Statistical Modeling, Springer, 2007.

[28]. M. P. Burnha, D. R. Anderson, Multimodel Inference, Sociological Methods and Research 33 (2004) 261--304.

[29]. M. Ye, P. D. Meyer, S. P. Neuman, On model selection criteria in multimodel analysis, Water Resources Research, 44 (2008) doi:10.1029/2008WR006803.

[30]. P. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.