Article

Bayesian Naïve Bayes Classifiers to Text Classification

Journal of Information Science
1–12
© The Author(s) 2016
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0165551510000000
jis.sagepub.com
SAGE

Shuo Xu
Research Centre for Information Science Theory and Methodology, Institute of Scientific and Technical Information of China, P. R. China

Abstract

Text classification is the task of assigning predefined categories to natural language documents, and it can provide conceptual views of document collections. The naïve Bayes (NB) classifier is a family of simple probabilistic classifiers based on a common assumption that all features are independent of each other, given the category variable, and it is often used as the baseline in text classification. However, classical NB classifiers with multinomial, Bernoulli and Gaussian event model are not fully Bayesian. This study proposes three Bayesian counterparts, where it turns out that classical NB classifier with Bernoulli event model is equivalent to Bayesian counterpart. Finally, experimental results on *20 newsgroups* and *WebKB* datasets show that the performance of Bayesian NB classifier with multinomial event model is similar to that of classical counterpart, but Bayesian NB classifier with Gaussian event model is obviously better than classical counterpart.

Keywords

Text Classification; Naïve Bayes Classifier; Event Model; Bayesian Naïve Bayes Classifier

1. Introduction

Text classification [1] is known as the task of assigning one or more predefined categories to natural language documents. Instead of manually classifying documents or hand-making automatic classification rules, many machine learning algorithms are used to automatically classify unseen documents on the basis of human-labelled training documents. Given the growing volume of online documents available through World Wide Web (WWW), news feeds, electronic mail, and digital libraries, this task is of great practical significance.

The naïve Bayes (NB) classifier is a family of simple probabilistic classifiers based on a common assumption that all features are independent of each other, given the category variable [2]. The different NB classifiers differ mainly by the assumptions they make regarding the distribution of features. The assumptions on distribution of features are called *event models* of the NB classifier [3]. For discrete features, multinomial or Bernoulli distributions are popular. These assumptions lead to two distinct models, which are often confused [4][5]. When dealing with continuous features, a typical assumption is Gaussian distribution.

Despite apparently over-simplifier assumptions, NB classifier works quite well in many complex real-world applications, such as text classification [6][7], keyphrase extraction [8], medical diagnosis [9]. This paradox is explained by Zhang that true reason for its competitive performance in classification lies in the dependence distribution [10]. In more details, how the local dependence of a feature distributes in each category, evenly or unevenly, and how the local dependencies of all features work together, consistently (supporting a certain category) or inconsistently (cancelling each other out), plays a crucial role.

As one of the most efficient inductive learning algorithms, NB classifier is often used as a baseline in text classification because it is fast and easy to implement. Moreover, with appropriate pre-processing, it is competitive with more advanced methods including support vector machines (SVMs) [4]. However, classical NB classifier, as standardly presented, is not fully Bayesian. At least not in the sense that a posterior distribution over parameters is estimated from training documents and then used for predictive inference for new document. Inspired by the success of Bayesian counterparts of many classical methods, Bayesian NB classifiers to text classification are studied in this work with the following contributions.

Corresponding author:
Shuo Xu, Institute of Scientific and Technical Information of China, No. 15 Fuxing Rd., Haidian District, Beijing 100038, P. R. China.
Email: xush@istic.ac.cn

- Bayesian NB classifiers with multinomial, Bernoulli and Gaussian event model are proposed in the paper, where it turns out that classical NB classifier with Bernoulli event model is equivalent to Bayesian counterpart.
- Bayesian NB classifier with multinomial event model is similar to that of classical counterpart, but Bayesian NB classifier with Gaussian event model is obviously better than classical counterpart.

The rest of this paper is organized as follows. Section 2 gives an overview of the related works in classical NB classifier and Bayesian methods. After classical NB classifier is briefly described in Section 3, a fully Bayesian NB classifier is proposed in Section 4. In Section 5, experimental results on 20 newsgroup data show that Bayesian NB classifier has similar performance with classical NB classifier, and Section 6 concludes this work.

2. Related work

In practice, the conditional independence assumption in NB classifier is rarely true, and as a result its probability estimates are often suboptimal. In order to reduce inaccuracies from naïve assumption, many approaches are proposed in literature. Such methods can be grouped into two categories. The first category comprises semi-naïve Bayes methods [11][12]. These methods are aimed at enhancing NB's accuracy by relaxing the conditional independence assumption. The second category includes feature weighting methods [13], though feature weighting is primarily been viewed as a means of increasing the influence of highly predictive feature and discounting features that have little predictive value [14][15].

Compared to classical methods, Bayesian methods [16] provide a natural and principled way of combining prior information with data, within a solid decision theoretical framework. One can incorporate past information about a parameter and form a prior distribution for future analysis. When new observations become available, the previous posterior distribution can be used as a prior. All inferences logically follow from Bayes' theorem. Therefore, many classical approaches are reformulated within a Bayesian framework, such as hidden Markov model (HMM) [17], principal component analysis (PCA) [18], SVM [19], multidimensional scaling (MDS) [20] and many others.

However, there is not a Bayesian treatment of classical NB classifier. To the best of my knowledge, only Rennie described the Bayesian NB classifier in his master thesis, and he found that Bayesian NB classifier performed worse than classical NB classifier [21]. In fact, Rennie's master thesis only considered multinomial event model, and did not care about Bernoulli and Gaussian event models. Furthermore, Dirichlet hyper-parameters were not tuned, which resulted in worse performance. In order to quantify the trade-off between various classification decisions and predict the risk that accompany such decisions, Di Nunzio put the classification decision of NB classifiers with multinomial, Gaussian, Bernoulli and Poisson event models within the framework of Bayesian decision theory [22], but it is not still fully Bayesian. It is worth noting that cost-sensitive NB classifiers are also applicable to Bayesian NB classifiers.

3. Classical Naïve Bayes classifier

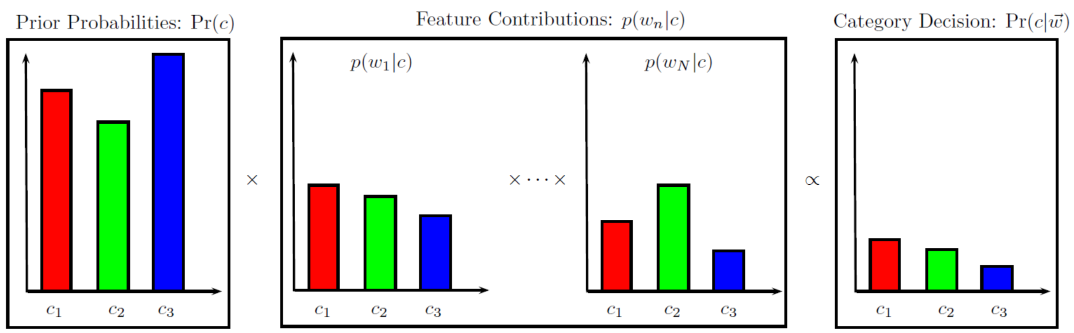

Prior Probabilities: $\Pr(c)$
Feature Contributions: $p(w_n|c)$
Category Decision: $\Pr(c|\vec{w})$
$p(w_1|c)$
$p(w_N|c)$
c_1 c_2 c_3
$\times$
$\times \cdots \times$
$\propto$

Figure 1. Decision-making procedure with the naïve Bayes classifier.

In the NB classifier, every feature w_n gets a say in determining which category $c \in \mathbb{N}_C = \{1, \cdots, C\}$ should be assigned to a unseen document $\vec{w} = (w_1, \cdots, w_N)$. The features for text classification are usually words, and the number of unique words can be quite large. To choose a category c from $\mathbb{N}_C$ for $\vec{w}$, NB classifier begins by calculating the prior probability $\Pr(c)$ of each category $c \in \mathbb{N}_C$, which is determined by assuming equiprobable classes, or checking the frequency of each category in the training document set. The contribution from each feature is then combined with this prior probability, to arrive at a likelihood estimate for each category. Figure 1 illustrates the decision-making procedure. This is known as the maximum a posteriori (MAP) decision rule. It can be defined formally as follows.

$$c \quad = \quad \arg\max_{c \in \mathbb{N}_C} p(c|\vec{w}) = \arg\max_{c \in \mathbb{N}_C} \Pr(c) p(\vec{w}|c) \tag{1}$$

As a matter of fact, NB classifier can be viewed as a generative process. To generate a document, NB classifier first choose a category for it, and then it generate each of the document's features (such as words) independently according to a category-specific distribution. Figure 2 illustrate the generative process. In this figure, an arrow indicates a conditional dependency between variables.

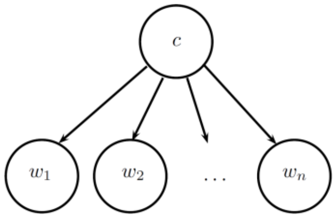


Figure 2. Bayesian network graph illustrating the generative process for the naïve Bayes classifier. In this figure, an arrow indicates a conditional dependency between variables.

Given a training document set $\mathcal{D} = \{(\vec{w}_m, c_m)\}_{m=1}^{\ell}$ with N_m word tokens from a given vocabulary of size V in the document m, $\vartheta_c = \Pr(c)$ can be easily estimated by counting the number of documents ℓ_c for each category $c \in \mathbb{N}_C$ in the set $\mathcal{D}$ by adding a smoothing prior $\alpha \geq 0$ as follows. However, in order to estimate $p(\vec{w}|c)$, one must assume a distribution or generate non-parametric models for the features, i.e., event model.

$$\hat{\vartheta}_c \quad = \quad \frac{\ell_c + \alpha}{\ell + C\alpha} \tag{2}$$

Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing [23].

3.1. Multinomial Event Model

With a multinomial event model, each document is represented by the set of word occurrences from the document. That is to say, the order of words is not captured. It yields the familiar *bag of words* representation for documents. It is not difficult to see that each document can also viewed as a histogram, with each element counting the number of occurrence of the resulting word in the document. Following the model, words for each category $c \in \mathbb{N}_C$ are usually generated by a separate multinomial $(\vec{\varphi}_c)$ where $\varphi_{c,v} = \Pr(v|c)$ is the probability of the category c generating the word v. Define $n^{(v)}$ to be the count of the number of times word v occurs in a particular document $\vec{w}$. The likelihood of observing a document $\vec{w}$ is given by

$$p(\vec{w}|c) \quad = \quad \frac{(\sum_{v=1}^{V} n^{(v)})!}{\prod_{v=1}^{V} n^{(v)}!} \prod_{v=1}^{V} \varphi_{c,v}^{n^{(v)}} \tag{3}$$

Similar to ϑ_c, a smoothed ML can be utilized to estimate $\varphi_{c,j}$ as follows. Here, $n_c^{(v)}$ is the number of times word v appears in the documents with category c in $\mathcal{D}$, $n_c^{(\cdot)} = \sum_{v=1}^{V} n_c^{(v)}$ and $\beta \geq 0$ is the smoothing prior.

$$\hat{\varphi}_{c,v} \quad = \quad \frac{n_c^{(v)}+\beta}{n_c^{(\cdot)}+V\beta} \qquad (4)$$

3.2. Bernoulli Event Model

In the Bernoulli event model, each document is represented by a vector $\vec{w}$ of binary features indicating which words occur and do not occur in the document. Each element w_v is a Boolean expressing the occurrence or absence of the *v*-th term. In other words, features are independent binary variables, and this model does not care about the number of times a word occurs in a document. Let $\varphi_{c,v} = \Pr(w_v|c)$ be the probability of the category *c* generating the word *v*, then the likelihood of a document $\vec{w}$ given a category *c* is given by

$$p(\vec{w}|c) \quad = \quad \prod_{v=1}^{V} \varphi_{c,v}^{w_v}(1-\varphi_{c,v})^{1-w_v} \qquad (5)$$

This event model is especially popular for classifying short texts [5]. It has the benefit of explicitly modelling the absence of words. Note that a NB classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one. This study estimates each of these class-conditional word probabilities $\varphi_{c,v}$ by straightforward counting of events, supplemented by a smoothing prior $\beta \geq 0$, as follows.

$$\hat{\varphi}_{c,v} \quad = \quad \frac{m_c^{(v)}+\beta}{\ell_c+2\beta} \qquad (6)$$

Here, $m_c^{(v)}$ denotes the number of documents with the category *c* containing the word *v*.

3.3. Gaussian Event Model

In text classification, it is very common that the documents are represented as term-frequency/inverse document frequency (TF×IDF) vectors. Because the TF×IDF value increases proportionally to the number of times a word appears in the document (i.e., term frequency, TF), but is offset by the frequency of the word in the corpus (i.e., document frequency, DF), which helps to adjust for the fact that some words appear more frequently in general. When dealing with continuous data, such as TF×IDF vectors, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. Another common technique is to use binning techniques [24][25] to discretize the feature values, to obtain a new set of Bernoulli-distributed features. In fact, the discretization may throw away some discriminative information [26].

According to the model, feature values of terms for each category $c \in \mathbb{N}_C$ are usually generated by a separate Gaussian $\mathcal{N}(\vec{\mu}_c, \vec{\sigma}_c^2)$ where $\vec{\mu}_c$ and $\vec{\sigma}_c^2$ are the mean and variance vectors of the feature values of words associated with category *c*, respectively. The likelihood of observing a document $\vec{w}$ is given by

$$p(\vec{w}|c) \quad = \quad \prod_{v=1}^{V} \mathcal{N}(w_v|\mu_{c,v}, \sigma_{c,v}^2) = \prod_{v=1}^{V} \frac{1}{\sqrt{2\pi\sigma_{c,v}^2}} \exp\left(-\frac{(w_v-\mu_{c,v})^2}{2\sigma_{c,v}^2}\right) \qquad (7)$$

Again, ML can be used to estimate $\vec{\mu}_c$ and $\vec{\sigma}_c^2$ from the training document set $\mathcal{D}$ as follows. In practice, if the ratio of data variance between words is too small, it will cause numerical errors. To address this problem, we artificially boost the variance by $\varepsilon = 1.0^{-9}$, a small fraction of the standard deviation of the largest dimension.

$$\hat{\mu}_{c,v} \quad = \quad \frac{1}{\ell_c} \sum_{m:c_m=c} w_{m,v} \qquad (8)$$

$$\hat{\sigma}_{c,v}^2 \quad = \quad \frac{1}{\ell_c} \sum_{m:c_m=c} (w_{m,v}-\hat{\mu}_{c,v})^2 \qquad (9)$$

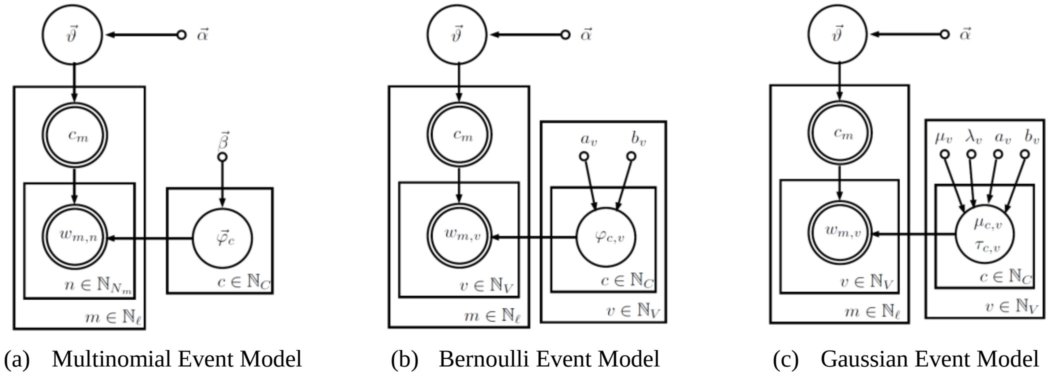
4. Bayesian multinomial NB classifier

The NB classifier, as standardly presented in Section 3, is not fully Bayesian. At least not in the sense that a posterior distribution over parameters is estimated from training documents and then used for predictive inference for new document. This section describes a fully Bayesian NB classifier in more details. The graphical model representation for

Bayesian NB classifier is shown in Figure 3. The Bayesian NB classifier can be viewed as a generative process, which can be described as follows.

(1) Draw a Multinomial($\vec{\vartheta}$) from Dirichlet($\vec{\alpha}$).
(2a) For each category $c \in \mathbb{N}_C$:
  (2a.1) Draw a Multinomial($\vec{\varphi}_c$) from Dirichlet($\vec{\beta}$);
(2b) For each category $c \in \mathbb{N}_C$:
  (2b.1) For each term $v \in \mathbb{N}_V$:
    (2b.1.1) Draw a Bernoulli($\varphi_{c,v}$) from Beta(a_v, b_v);
(2c) For each category $c \in \mathbb{N}_C$:
  (2c.1) For each term $v \in \mathbb{N}_V$:
    (2c.1.1) Draw a Gaussian($\mu_{c,v}, \tau_{c,v}$) from GaussianGamma($\mu_v, \lambda_v, a_v, b_v$);
(3) For each document $m \in \mathbb{N}_\ell$:
  (3.1) Draw a category c_m from Multinomial($\vec{\vartheta}$);
  (3.2a) For each word $n \in \mathbb{N}_{N_m}$ in document *m*:
    (3.2a.1) Draw a word $w_{m,n}$ from Multinomial($\vec{\varphi}_{c_m}$).
  (3.2b) For each word $v \in \mathbb{N}_V$:
    (3.2b.1) Draw a Boolean variable *x* from Bernoulli($\varphi_{c_m,v}$);
    (3.2b.2) If *x* is true, append the word *v* to document *m*; discard the word *v* otherwise.
  (3.2c) For each word $n \in \mathbb{N}_V$:
    (3.2c.1) Draw a term feature value from Gaussian($\mu_{c_m,v}, \tau_{c_m,v}$).

It is worth noting that one can generate the resulting documents from the above procedure for multinomial and Bernoulli event models, but one can only generate the resulting feature vector representations for Gaussian event model.



(a) Multinomial Event Model (b) Bernoulli Event Model (c) Gaussian Event Model

Figure 3. The graphical model representation for the Bayesian naïve Bayes classifier. In this figure, circle and double-circle variables indicate observed and latent variables, respectively. An arrow indicates a conditional dependency between variables, and stacked panes indicate a repeated sampling with the iteration number shown.

For convenience, let $\Phi = \{\vec{\varphi}_c\}, \{\varphi_{c,v}\}, \{\mu_{c,v}, \tau_{c,v}\}$ and $\Psi = \vec{\beta}, \{a_v, b_v\}, \{\mu_v, \lambda_v, a_v, b_v\}$ for multinomial, Bernoulli or Gaussian event model, respectively.

4.1. Parameter estimation

Given a training document set $\mathcal{D}$, priors $\vec{\alpha}$ and Ψ, MAP estimates of $\vec{\vartheta}$ and Φ can be calculated formally:

$$\begin{aligned}\left\{\hat{\vec{\vartheta}},\hat{\Phi}\right\} &= \arg\max p(\vec{\vartheta},\Phi|\mathcal{D},\vec{\alpha},\Psi)\\ &= \arg\max \Pr(\mathcal{D}|\vec{\vartheta})\, p(\vec{\vartheta}|\vec{\alpha})\times\Pr(\mathcal{D}|\Phi)p(\Phi|\Psi)\\ &= \arg\max \mathrm{Dirichlet}(\vec{\vartheta}|\vec{\ell}+\vec{\alpha})\times\arg\max\Pr(\mathcal{D}|\Phi)p(\Phi|\Psi)\end{aligned} \quad (10)$$

with $\vec{\ell}=\{\ell_c\}_{c=1}^{C}$. Following the mode of Dirichlet distribution, MAP parameter estimates for $\vec{\vartheta}$ can be expressed as follows. It is not difficult to see that (2) is equivalent to (11) when $\alpha_c=\alpha+1$ ($c\in\mathbb{N}_C$).

$$\hat{\vartheta}_c \quad = \quad \frac{\ell_c+\alpha_c-1}{\ell+\sum_{c'=1}^{C}(\alpha_{c'}-1)} \quad (11)$$

However, similar to NB classifier, in order to estimate Φ, one must assume an event model.

(1) Multinomial event model

$$\begin{aligned}\left\{\hat{\vec{\varphi}}_c\right\} &= \arg\max \Pr(\mathcal{D}|\{\vec{\varphi}_c\})p(\{\vec{\varphi}_c\}|\vec{\beta})\\ &= \arg\max\ \prod_{c=1}^{C}\mathrm{Dirichlet}(\vec{\varphi}_c|\vec{n}_c+\vec{\beta})\end{aligned} \quad (12)$$

with $\vec{n}_c=\{n_c^{(v)}\}_{v=1}^{V}$. Following the mode of Dirichlet distribution, MAP parameter estimates for $\left\{\hat{\vec{\varphi}}_c\right\}$ can be expressed as follows.

$$\hat{\varphi}_{c,v} \quad = \quad \frac{n_c^{(v)}+\beta_v-1}{n_c^{(\cdot)}+\sum_{v'=1}^{V}(\beta_{v'}-1)} \quad (13)$$

It is easy to see that (4) is equivalent to (13) when $\beta_v=\beta+1$ ($v\in\mathbb{N}_V$).

(2) Bernoulli event model

$$\begin{aligned}\left\{\hat{\varphi}_{c,v}\right\} &= \arg\max \Pr(\mathcal{D}\left|\left\{\varphi_{c,v}\right\})p(\left\{\varphi_{c,v}\right\}\right|a,b)\\ &= \arg\max\ \prod_{c=1}^{C}\prod_{v=1}^{V}\mathrm{Beta}(\varphi_{c,v}|n_c^{(v)}+\mathrm{a},\ell_c-n_c^{(v)}+\mathrm{b})\end{aligned} \quad (14)$$

Following the mode of Beta distribution, MAP parameter estimates for $\left\{\hat{\varphi}_{c,v}\right\}$ can be expressed as follows.

$$\hat{\varphi}_{c,v} \quad = \quad \frac{n_c^{(v)}+a-1}{\ell_c+a+b-2} \quad (15)$$

Again, (6) is equivalent to (15) when $a=b=\beta+1$.

(3) Gaussian event model

$$\begin{aligned}\left\{\hat{\mu}_{c,v},\hat{\tau}_{c,v}\right\} &= \arg\max \Pr(\mathcal{D}\left|\left\{\mu_{c,v},\tau_{c,v}\right\})p(\left\{\mu_{c,v},\tau_{c,v}\right\}\right|\mu_v,\lambda_v,a_v,b_v)\\ &= \arg\max\ \prod_{c=1}^{C}\prod_{v=1}^{V}\mathrm{GaussianGamma}(\mu_{c,v},\tau_{c,v}|\mu_{c,v}^{\mathcal{D}},\lambda_c^{\mathcal{D}},a_c^{\mathcal{D}},b_{c,v}^{\mathcal{D}})\end{aligned} \quad (16)$$

where

$$\begin{aligned}\mu_{c,v}^{\mathcal{D}} &= \frac{\lambda_v\mu_v+\ell_c\bar{\mu}_{c,v}}{\lambda_v+\ell_c}\\ \lambda_{c,v}^{\mathcal{D}} &= \lambda_v+\ell_c\\ a_c^{\mathcal{D}} &= a_v+\frac{\ell_c}{2}\\ b_{c,v}^{\mathcal{D}} &= b_v+\frac{1}{2}\left(\ell_c\bar{\sigma}_{c,v}^2+\frac{\lambda_v\ell_c(\bar{\mu}_{c,v}-\mu_v)^2}{\lambda_v+\ell_c}\right)\end{aligned} \quad (17)$$

with the mean $\bar{\mu}_{c,v}$ and variance $\bar{\sigma}^2_{c,v}$ of feature value for term v in documents with category c. Following the mode of GaussianGamma distribution [27][28], MAP parameter estimates can be expressed as follows.

$$\begin{aligned} \hat{\mu}_{c,v} &= \mu^{\mathcal{D}}_{c,v} = \frac{\lambda_v \mu_v + \ell_c \bar{\mu}_{c,v}}{\lambda_v + \ell_c} \\ \hat{\tau}_{c,v} &= \frac{2a^{\mathcal{D}}_c - 1}{2b^{\mathcal{D}}_{c,v}} = \frac{2a_v + \ell_c - 1}{2b_v + \left(\ell_c \bar{\sigma}^2_{c,v} + \frac{\lambda_v \ell_c (\bar{\mu}_{c,v} - \mu_v)^2}{\lambda_v + \ell_c} + \right)} \end{aligned} \tag{18}$$

4.2. Decision-making procedure

In order to assign a category c to a given document $\vec{w}$ with N word tokens, fully Bayesian inference should be used as follows.

$$\begin{aligned} c &= \arg\max \Pr(c|\vec{w}, \mathcal{D}, \vec{\alpha}, \Psi) \\ &= \arg\max \iint p(c, \vec{w}, \vec{\vartheta}, \Phi|\mathcal{D}, \vec{\alpha}, \Psi) d\vec{\vartheta} d\Phi \\ &= \arg\max \iint \Pr(c|\vec{\vartheta}) p(\vec{\vartheta}|\mathcal{D}, \vec{\alpha}) \Pr(\vec{w}|\Phi) p(\Phi|\mathcal{D}, \Psi) d\vec{\vartheta} d\Phi \\ &= \arg\max \int \Pr(c|\vec{\vartheta}) p(\vec{\vartheta}|\mathcal{D}, \vec{\alpha}) d\vec{\vartheta} \times \arg\max \int \Pr(\vec{w}|\Phi) p(\Phi|\mathcal{D}, \Psi) d\Phi \\ &= \arg\max \int \vartheta_c \, p\left(\vec{\vartheta}\middle|\mathcal{D}, \vec{\alpha}\right) d\vec{\vartheta} \times \arg\max \int \Pr(\vec{w}|\Phi) p(\Phi|\mathcal{D}, \Psi) d\Phi \\ &= \arg\max \frac{\Gamma\left(\sum_{c'=1}^{C} \alpha_{c'} + \ell\right)}{\prod_{c'=1}^{C} \Gamma(\alpha_{c'} + \ell_{c'})} \frac{\prod_{c'=1}^{C} \Gamma\left(\alpha_{c'} + \ell_{c'} + I(c = c')\right)}{\Gamma\left(\sum_{c'=1}^{C} \alpha_{c'} + \ell + 1\right)} \times \arg\max \int \Pr(\vec{w}|\Phi) p(\Phi|\mathcal{D}, \Psi) d\Phi \\ &= \arg\max \frac{\alpha_c + \ell_c}{\sum_{c'=1}^{C} \alpha_{c'} + \ell} \times \arg\max \int \Pr(\vec{w}|\Phi) p(\Phi|\mathcal{D}, \Psi) d\Phi \end{aligned} \tag{19}$$

where $\Gamma(\cdot)$ and $I(\cdot)$ is the Gamma and indicator function respectively. Again, an event model should be assumed in order to calculate the second term in (19).

(1) Multinomial event model

$$\begin{aligned} & \int \Pr(\vec{w}|\vec{\varphi}_c) p(\vec{\varphi}_c|\mathcal{D}, \vec{\beta}) d\vec{\varphi}_c \\ = & \int \prod_{n=1}^{N} \varphi_{c,w_n} \, p\left(\vec{\varphi}_c\middle|\mathcal{D}, \vec{\beta}\right) d\vec{\varphi}_c \\ = & \frac{\Gamma(\sum_{v=1}^{V} \beta_v + n_c)}{\prod_{v=1}^{V} \Gamma\left(\beta_v + n_c^{(v)}\right)} \frac{\prod_{v=1}^{V} \Gamma\left(\beta_v + n_c^{(v)} + n^{(v)}\right)}{\Gamma(\sum_{v=1}^{V} \beta_v + n_c + N)} \end{aligned} \tag{20}$$

(2) Bernoulli event model

$$\begin{aligned} & \int \Pr(\vec{w}|\vec{\varphi}_c) p(\vec{\varphi}_c|\mathcal{D}, a, b) d\vec{\varphi}_c \\ = & \prod_{v=1}^{V} \int \varphi_{c,v}^{w_v} (1 - \varphi_{c,v})^{1 - w_v} p(\varphi_{c,v}|\mathcal{D}, a, b) d\varphi_{c,v} \\ = & \prod_{v=1}^{V} \frac{(n_c^{(v)} + a)^{w_v} (\ell_c - n_c^{(v)} + b)^{1 - w_v}}{\ell_c + a + b} \end{aligned} \tag{21}$$

where $\vec{\varphi}_c = \{\varphi_{c,v}\}_{v=1}^{V}$. If $a = b = \beta + 1$, decision-making function for Bayesian NB classifier is equivalent to that of classical counterpart.

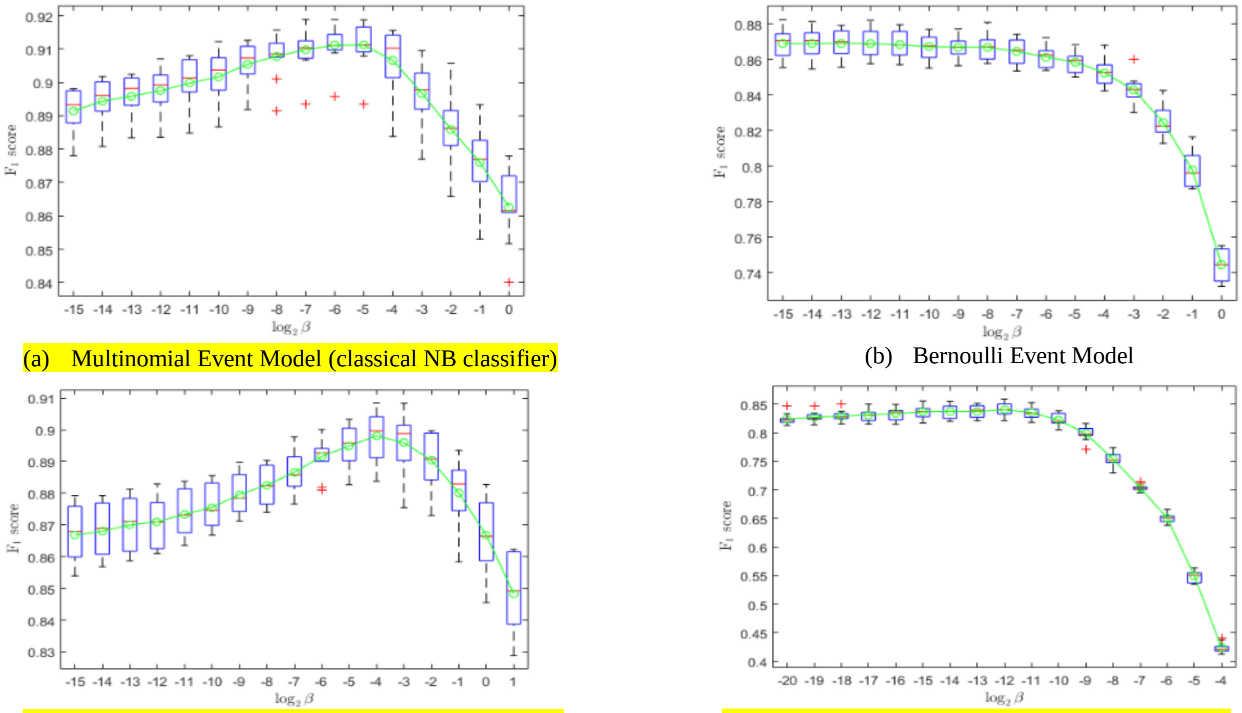
(3) Gaussian event model

$$\begin{aligned} & \int\int \Pr(\vec{w}|\vec{\mu}_c,\vec{\tau}_c)p(\vec{\mu}_c,\vec{\tau}_c|\mathcal{D},\mu_v,\lambda_v,a_v,b_v)d\vec{\mu}_c d\vec{\tau}_c \\ = & \prod_{v=1}^{V}\int\int \mathcal{N}(w_v|\mu_{c,v},\sigma_{c,v}^2)p(\mu_{c,v},\tau_{c,v}|\mathcal{D},\mu_v,\lambda_v,a_v,b_v)d\mu_{c,v}d\tau_{c,v} \\ = & \prod_{v=1}^{V}\left[\frac{(b_{c,v}^{\mathcal{D}})^{a_c^{\mathcal{D}}}\sqrt{\lambda_{c,v}^{\mathcal{D}}}}{\Gamma(a_c^{\mathcal{D}})}\frac{\Gamma(a_c^{\mathcal{D}}+\frac{1}{2})}{\left(b_{c,v}^{\mathcal{D}}+\frac{\lambda_{c,v}^{\mathcal{D}}\left(w_v-\mu_{c,v}^{\mathcal{D}}\right)^2}{2\left(\lambda_{c,v}^{\mathcal{D}}+1\right)}\right)^{a_c^{\mathcal{D}}+\frac{1}{2}}\sqrt{\lambda_{c,v}^{\mathcal{D}}+1}}\right] \end{aligned} \qquad (22)$$

where $\vec{\mu}_c = \{\mu_{c,v}\}_{v=1}^{V}$ and $\vec{\tau}_c = \{\tau_{c,v}\}_{v=1}^{V}$.

5. Experiments and Discussions

In this study, two benchmark dataset, *20 newsgroups* and *WebKB*, are utilized to evaluate the performance. *20 newsgroups* was collected and originally used for text classification by Lang [29], which contains 18,821 non-empty documents evenly distributed across 20 categories, each representing a newsgroup. *WebKB* contains webpages collected from computer science departments of various universities by the World Wide Knowledge Base (Web->Kb) project of the CMU text learning group. As with [30], the categories "Department" and "Staff" were discarded because there were only a few pages from each university. The category "Other" was also discarded, because pages were very different among the examples for this class. After these discarding operations, 4,199 webpages are left in the end. The same pre-processing and splitting with [4][21][31] are applied to these two datasets. The final vocabulary size for *20 newsgroups* and *WebKB* are 70,216 and 7,770, respectively. Please refer to [31] for more details.



(a) Multinomial Event Model (classical NB classifier)

(b) Bernoulli Event Model

(c) Multinomial Event Model (Bayesian NB classifier)

(d) Gaussian Event Model (Bayesian NB classifier)

Figure 4. The performance of 10-fold cross validation with $\log_2\beta$ in term of macro-average F_1 score on *20 newsgroups* dataset: (a) is for classical NB classifier, (c) and (d) are for Bayesian NB classifier. Since Bayesian NB classifier with Bernoulli event model is equivalent to that of classical counterpart (a=b=β+1), (b) is for classical and Bayesian NB classifiers.

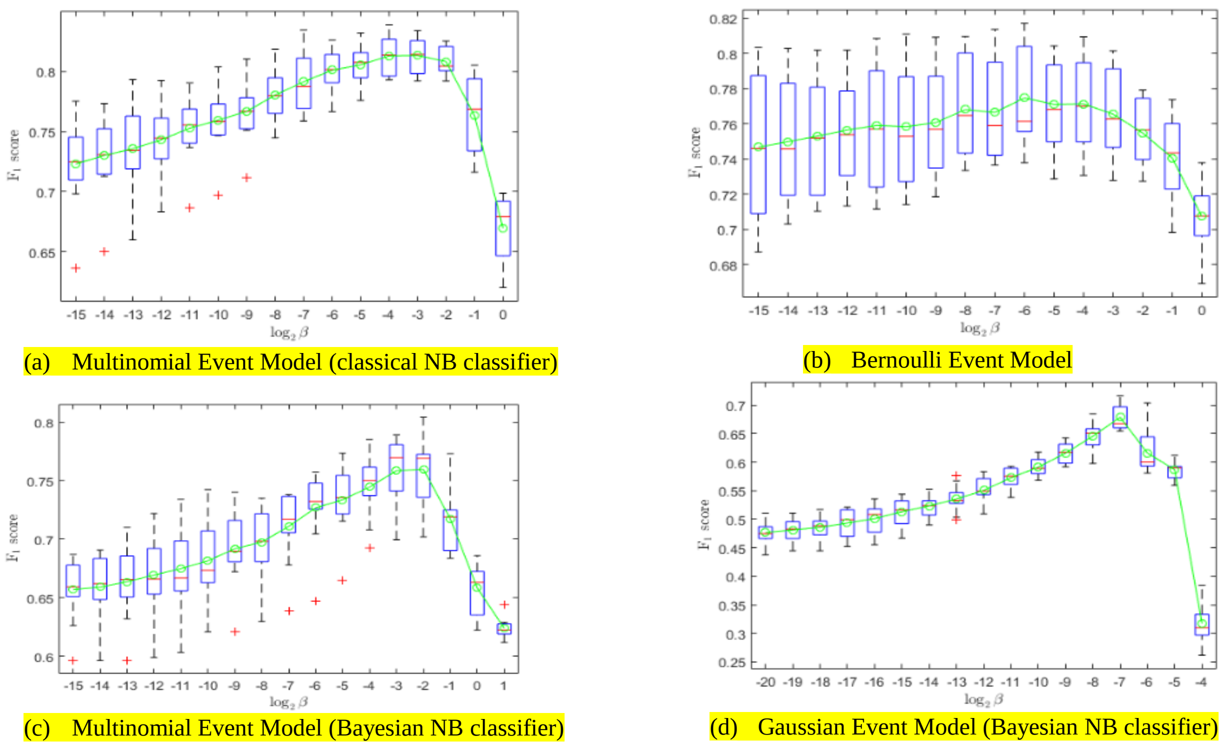
In order to generate continuous feature vector representations for Gaussian event model, we then do a kind of TF×IDF transformation as follows, and normalize each document to unit length [32].

$$\mathrm{TF}\times\mathrm{IDF}_{m,v} \quad = \quad \ln(1+\mathrm{TF}_{m,v})\times\log_2\left(\frac{\ell}{\mathrm{DF}_v}\right) \qquad (23)$$

To evaluate the performance of resulting classifiers, three standard measures for binary classification, precision, recall and F_ρ score, are utilized in this study. Precision, recall and F_ρ score ($\rho = 1$ in this study) are defined formally as:

$$\begin{aligned} \mathrm{P} &= \frac{TP}{TP+FP} \\ \mathrm{R} &= \frac{TP}{TP+FN} \\ \mathrm{F}_\rho &= (1+\rho^2)\frac{\mathrm{P}\times\mathrm{R}}{\rho^2\mathrm{P}+\mathrm{R}} \end{aligned} \qquad (24)$$

Here, *TP* (true positive) is the number of the correct positive predictions, *FP* (false positive) is the number of incorrect positive predictions, and *FN* (false negative) is the number of incorrect negative predictions.



(a) Multinomial Event Model (classical NB classifier)

(b) Bernoulli Event Model

(c) Multinomial Event Model (Bayesian NB classifier)

(d) Gaussian Event Model (Bayesian NB classifier)

Figure 5. The performance of 10-fold cross validation with log2β in term of macro-average F1 score on *WebKB* dataset: (a) is for classical NB classifier, (c) and (d) are for Bayesian NB classifier. Since Bayesian NB classifier with Bernoulli event model is equivalent to that of classical counterpart (*a*=*b*=β+1), (b) is for classical and Bayesian NB classifiers.

In classical NB classifier, α is fixed to 1, and β is tuned for multinomial and Bernoulli event models. It is not needed to tune parameters for Gaussian event model. For simplicity, the symmetric Dirichlet priors are used in Bayesian NB classifier, where $\alpha_c = 2$ ($c \in \mathbb{N}_C$), $\beta \equiv \beta_v$ ($v \in \mathbb{N}_V$) or $\beta \equiv$ a = b is tuned for multinomial and Bernoulli event models. As for Gaussian event model, μ_v ($v \in \mathbb{N}_V$) is fixed to sample mean, λ_v ($v \in \mathbb{N}_V$) is fixed to 1, and $\beta \equiv a_v = b_v$ ($v \in \mathbb{N}_V$) is tuned. In order to identify proper parameters, the grid search [33] with 10-fold cross validation is adopted. Let $\log_2\beta \in$

{-15, -14, …, 0} for NB classifier, $\log_2\beta \in$ {-15, -14, …, 1} and $\log_2\beta \in$ {-20, -19, …, -4} for multinomial and Gaussian event model in Bayesian NB classifier, respectively. The performance on *20 newsgroups* and *WebKB* in terms of macro-average F_1 score with β is reported in Figure 4 and Figure 5, respectively. From Figure 4 and Figure 5, it is not difficult to see that the performance is sensitive to the resulting parameters. Therefore, it is necessary for users to identify proper parameters in advance with grid search or other similar methods.

Table 1. Experimental Results on *20 newsgroups* dataset in term of precision (%), recall (%) and F1 score (%).

ID	Multinomial Event Model						Bernoulli Event Model			Gaussian Event Model					
	Classical			Bayesian						Classical			Bayesian		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
1	**80.72**	77.43	**79.04**	76.02	**81.50**	78.67	74.56	79.00	76.71	75.65	73.04	74.32	**75.15**	**78.68**	**76.88**
2	**72.93**	76.86	**74.84**	68.64	**80.46**	74.08	58.96	70.18	64.08	57.51	58.10	57.80	**76.25**	**58.61**	**66.28**
3	**78.27**	66.75	72.05	72.08	**72.08**	**72.08**	64.64	59.39	61.90	**58.42**	44.92	50.79	54.38	**66.24**	**59.73**
4	66.06	**73.98**	69.80	**71.6**8	71.68	**71.68**	61.61	66.33	63.88	**58.63**	54.59	**56.54**	47.84	**67.86**	56.12
5	**80.05**	81.30	**80.67**	75.00	**83.38**	78.97	65.53	74.55	69.74	**68.12**	61.04	**64.38**	52.06	**78.70**	62.67
6	85.79	80.10	**82.85**	**92.93**	70.41	80.12	84.31	69.90	76.43	71.92	**69.90**	70.89	**85.06**	66.84	**74.86**
7	**81.79**	72.56	76.90	81.12	**78.21**	**79.63**	56.80	78.21	65.80	62.91	**48.72**	54.91	**82.76**	43.08	**56.66**
8	86.96	91.14	89.00	**87.53**	**92.41**	**89.90**	83.12	83.54	83.33	**80.46**	79.24	79.85	77.73	**84.81**	**81.11**
9	**90.31**	95.98	**93.06**	88.84	95.98	92.27	87.09	93.22	90.05	**82.97**	86.93	84.91	80.43	**92.96**	**86.25**
10	**97.42**	**95.21**	**96.31**	97.39	93.95	95.64	93.35	88.41	90.82	**89.89**	82.87	**86.24**	73.32	**96.22**	83.22
11	96.77	**97.74**	**97.26**	97.46	96.24	96.85	97.61	91.98	94.71	86.79	92.23	89.43	**93.48**	**93.48**	**93.48**
12	87.01	**94.70**	90.69	**89.47**	94.44	**91.89**	84.26	87.88	86.03	73.90	85.10	79.11	**75.21**	**91.16**	**82.42**
13	77.53	72.01	74.67	79.49	72.01	**75.57**	71.07	64.38	67.56	68.29	57.00	62.14	**71.47**	**66.92**	**69.12**
14	90.38	83.08	86.58	**91.44**	**83.59**	**87.34**	86.13	75.25	80.32	61.20	**77.27**	68.30	**95.72**	62.12	**75.34**
15	86.10	**89.59**	87.81	**89.54**	89.09	**89.31**	85.26	82.23	83.72	69.11	**81.22**	74.68	**89.37**	78.93	**83.83**
16	78.65	**93.47**	85.42	**87.20**	90.70	**88.92**	84.50	84.92	84.71	**75.40**	83.92	79.43	70.88	**92.96**	**80.43**
17	**72.45**	**91.76**	**80.97**	71.71	91.21	80.29	74.38	82.14	78.07	69.05	71.70	70.35	**68.71**	**86.26**	**76.49**
18	94.78	**91.76**	**93.24**	**96.37**	84.84	90.24	95.61	81.12	87.77	79.21	**85.11**	82.05	**98.71**	81.65	**89.37**
19	**75.70**	**61.29**	**67.74**	72.11	58.39	64.53	71.84	56.77	63.42	62.42	**63.23**	**62.82**	**89.40**	43.55	58.57
20	**80.21**	59.76	**68.49**	65.95	**60.96**	63.35	72.46	59.76	65.50	60.24	**59.76**	**60.00**	**83.61**	20.32	32.69
avg.	**82.99**	**82.32**	**82.37**	82.60	82.08	82.07	77.65	76.46	76.73	70.60	70.79	70.45	**77.08**	**72.57**	**72.28**

To make it clear, category names corresponding to the first column are listed as follows: 1-alt.atheism, 2-comp.graphics, 3-comp.os.ms-windows.misc, 4-comp.sys.ibm.pc.hardware, 5-comp.sys.mac.hardware, 6-comp.windows.x, 7-misc.forsale, 8-rec.autos, 9-rec.motorcycles, 10-rec.sport.baseball, 11-rec.sport.hockey, 12-sci.crypt, 13-sci.electronics, 14-sci.med, 15-sci.space, 16-soc.religion.christian, 17-talk.politics.guns, 18-talk.politics.mideast, 19-talk.politics.misc, 20-talk.religion.misc.

Table 2. Experimental Results on *WebKB* dataset in term of precision (%), recall (%) and F1 score (%).

ID	Multinomial Event Model						Bernoulli Event Model			Gaussian Event Model					
	Classical			Bayesian						Classical			Bayesian		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
1	**87.00**	**88.60**	**87.80**	83.45	88.05	85.69	74.51	84.38	79.14	73.89	58.27	65.16	**88.89**	**70.59**	**78.69**
2	92.58	**92.58**	92.58	**94.39**	92.26	**93.31**	97.67	81.29	88.73	**73.31**	70.00	71.62	70.45	**96.13**	**81.31**
3	76.28	**83.42**	**79.69**	**81.82**	72.19	76.70	71.39	72.73	72.05	46.37	63.10	53.45	**67.21**	**76.74**	**71.66**
4	**81.30**	59.52	68.73	68.78	**77.38**	**72.83**	75.89	63.69	69.26	28.40	27.38	27.88	**61.40**	**41.67**	**49.65**
avg.	**84.29**	81.03	**82.20**	82.11	**82.47**	82.13	79.87	75.52	77.29	55.49	54.69	54.53	**71.99**	**71.28**	**70.33**

To make it clear, category names corresponding to the first column are listed as follows: 1-student, 2-course, 3-faculty, 4-project.

With the tuned parameters in Figure 4 and Figure 5, the experimental results on test data are reported in Table 1 and Table 2 in term of precision, recall and F_1 score. Table 3 shows two-tailed significance with 95% confidence interval by paired-samples t-test [34]. From Table 1 and Table 2, one can see that the performance of Bayesian NB classifier with multinomial event model is similar to that of classical counterpart, but Bayesian NB classifier with Gaussian event model is obviously better than classical counterpart. Table 3 also illustrates that there is no statistically significant difference between Bayesian and classical NB classifiers with multinomial event model, but as for Gaussian event model, the difference between Bayesian and classical NB classifiers is statistical significant, especially for *WebKB* dataset. This observation is not consistent with that of Rennie [21]. What's more, NB classifier with multinomial event model outperforms that with Bernoulli event model, and NB classifier with Bernoulli event model outperforms that with Gaussian event model.

Table 3. Two-Tailed Statistical Significance with 95% Confidence Interval by Paired-Samples T-Test.

Multinomial Event Model						Gaussian Event Model					
20 newsgroups			*WebKB*			*20 newsgroups*			*WebKB*		
P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
0.735	0.770	0.531	0.512	0.773	0.959	0.051	0.563	0.290	0.046	0.048	0.002

6. Conclusions

Text classification is a supporting technology in several information processing tasks, including controlled vocabulary indexing, content filtering (spam, pornography, etc.), information security, and others. Instead of manually classifying documents, many machine learning algorithms are trained to automatically classify documents based on annotated training documents. The naïve Bayes (NB) classifier is often used as the baseline in text classification. However, classical NB classifiers with multinomial, Bernoulli and Gaussian event model are not fully Bayesian.

Inspired by the success of Bayesian counterparts of many classical methods, such as HMM, PCA, SVM and MDS, this study proposes three Bayesian counterpart classifiers, where it turns out that classical NB classifier with Bernoulli event model is equivalent to Bayesian counterpart. As a matter of fact, one can easily generalize the approach in the work to construct alternative NB classifiers with exponential family [35] event model. Finally, experimental results on *20 newsgroups* and *WebKB* data sets show that Bayesian NB classifier with multinomial event model performs similarly with classical counterpart, but Bayesian NB classifier with Gaussian event model is obviously better than classical counterpart. What's more, NB classifier with multinomial event model outperforms that with Bernoulli event model, and NB classifier with Gaussian event model comes next to that with Gaussian event model.

Acknowledgements

This research received the financial support from National Science Foundation of China (ID: 71403255), and Key Technologies R&D Program of Chinese 12th Five-Year Plan (2011-2015) (ID: 2015BAH25F01).

References

[1] Aggarwal CC and Zhai C. A Survey of Text Classification Algorithms. In: Aggarwal CC and Zhai C (ed.) Mining Text Data. Berlin: Springer, 2012, pp. 163–222.

[2] Russell SJ and Norvig P. Artificial intelligence: a modern approach. 3nd ed. New Jersey: Prentice Hall, 2009.

[3] John GH and Langley P. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th International Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, 1995, pp. 338—345.

[4] McCallum A and Nigam K. A comparison of event models for naïve Bayes text classification. In: ICML/AAAI-98 Workshop on Learning for Text Categorization, AAAI, 1998, pp. 41–48.

[5] Metsis V and Androutsopoulos I, and Paliouras G. Spam filtering with naive Bayes – which naive Bayes? In: The 3rd Conference on Email and Anti-Spam, 2006.

[6] Rennie JDM, Shih L, Teevan J, and Karger DR. Tackling the poor assumptions of naive Bayes text classifiers. In: Proceedings of the 20st International Conference on Machine Learning, 2003.

[7] Bird S, Klein E, and Loper E. Natural language processing with python. MO'Reilly, 2009, pp. 245–250.

[8] Witten IH, Paynter GW, Frank E, Gutwin C, and Nevill-Manning CG. KEA: Practical automatic keyphrase extraction. In: Proceedings of the 4th ACM Conference on Digital Libraries, ACM, 1999, 254–255.

[9] Rish I. An empirical study of the naive Bayes classifier. In: IJCAI Workshop on Empirical Methods in AI, 2001.

[10] Zhang H. The optimality of naive Bayes. In: Barr V and Markov Z (ed.) Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2004, 562–567.

[11] Cerquides J and De Mántaras RL. Tan classifiers based on decomposable distributions. Machine Learning 2005; 59: 323–354.

[12] Zheng F, Webb GI and Suraweera P, and Zhu L. Subsumption resolution: An efficient and effective technique for semi-naive Bayesian learning. Machine learning 2012; 87: 93–125.

[13] Zaidi NA, Cerquides J, Carman MJ, and Webb GI. Alleviating naive Bayes attribute independence assumption by attribute weighting. Journal of Machine Learning 2013; 14: 1947–1988.

[14] Zhang H and Sheng S. Learning weighted naive Bayes with accurate ranking. In: Proceedings of the 4th IEEE International Conference on Data Mining, 2004, 567–570.

[15] Hall MA. A decision tree-based attribute weighting filter for naive Bayes. Knowledge-based System 2007; 20: 120–126.

[16] Congdon P. Applied Bayesian Modelling. 2nd Ed. New Jersey: Wiley, 2014.

[17] Goldwater S and Griffiths TL. A fully Bayesian approach to unsupervised part-of-speech tagging. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2007, 744–751.

[18] Bishop CM. Bayesian PCA. In: Advances in Neural Information Processing Systems 11, MIT Press, 1999, 382–388.

[19] Tipping ME. The relevance vector machine. In: Advances in Neural Information Processing Systems 12, MIT Press, 2000, 652–658.

[20] Oh, M-S and Raftery AE. Bayesian multidimensional scaling and choice of dimension. Journal of the American Statistical Association 2011; 96: 1031–1044.

[21] Rennie JDM. Improving multi-class text classification with naive Bayes. Master's thesis, Massachusetts Institute of Technology, 2001.

[22] Di Nunzio GM. A new decision to take for cost-sensitive naïve Bayes classifiers. Information Processing and Management 2014; 50: 653–674.

[23] Manning CD, Raghavan P, and Schütze H. Introduction to Information Retrieval. Cambridge University Press, 2008.

[24] Liu H, Hussain F, Tan CL, and Dash M. Discretization: An enabling technique. Data Mining and Knowledge Discovery 2002; 6: 393–423.

[25] Dougherty J, Kohavi R, and Sahami M. Supervised and unsupervised discretization of continuous features. Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, 1995, 194–202.

[26] Hand DJ and Yu K. Idiot's Bayes–Not so stupid after all? International Statistical Review 2001; 69: 385–398.

[27] Dearden R, Friedman N, and Russell S. Bayesian Q-learning. In: Proceedings of the 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, American Association for Artificial Intelligence, 1998, 761–768.

[28] Bernardo J and Smith AFM. Bayesian Theory. John Wiley & Sons, 2007.

[29] Liang K. NewsWeeder: Learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning, 1995, 331–339.

[30] Nigam K, McCallum AK, Thrun S, and Mitchell T. Text Classification from Labeled and Unlabeled Documents from EM. Machine Learning 2000; 39: 103–134.

[31] Cardoso-Cachopo A. Improving Methods for Single-Label Text Categorization. Master's thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

[32] Rennie J and Rifkin R. Improving Multiclass Text Classification with the Support Vector Machine. Massachusetts Institute of Technology. AI Memo AIM-2001-026, 2001.

[33] Xu S, Ma F, and Tao L. Learn from the information contained in the false splice sites as well as in the true splice sites using SVM. In: Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering, Atlantis Press, 2007, 1360–1366.

[34] Blalock HM. Social Statistics. 2nd Edition. New York: McGraw-Hill, 1979.

[35] Barndorff-Nielsen O. Information and exponential families in statistical theory. Wiley, Chichester 1978.