Journal of Experimental & Theoretical Artificial Intelligence

ISSN: 0952-813X (Print) 1362-3079 (Online) Journal homepage: http://www.tandfonline.com/loi/teta20

Naive Bayes text classifiers: a locally weighted learning approach

Liangxiao Jiang , Zhihua Cai , Harry Zhang & Dianhong Wang

To cite this article: Liangxiao Jiang , Zhihua Cai , Harry Zhang & Dianhong Wang (2013) Naive Bayes text classifiers: a locally weighted learning approach, Journal of Experimental & Theoretical Artificial Intelligence, 25:2, 273-286, DOI: 10.1080/0952813X.2012.721010

To link to this article: https://doi.org/10.1080/0952813X.2012.721010

Journal of Experimental & Theoretical Artificial Intelligence
Vol. 25, No. 2, June 2013, 273–286

Taylor & Francis
Taylor & Francis Group

Naive Bayes text classifiers: a locally weighted learning approach

Liangxiao Jiang[a], Zhihua Cai[a]*, Harry Zhang[b] and Dianhong Wang[c]

[a]*Department of Computer Science, China University of Geosciences, Wuhan, Hubei 430074, China;* [b]*Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick E3B5A3, Canada;* [c]*Department of Electronic Engineering, China University of Geosciences, Wuhan, Hubei 430074, China*

(*Received 24 September 2011; final version received 6 May 2012*)

Due to being fast, easy to implement and relatively effective, some state-of-the-art naive Bayes text classifiers with the strong assumption of conditional independence among attributes, such as multinomial naive Bayes, complement naive Bayes and the one-versus-all-but-one model, have received a great deal of attention from researchers in the domain of text classification. In this article, we revisit these naive Bayes text classifiers and empirically compare their classification performance on a large number of widely used text classification benchmark datasets. Then, we propose a locally weighted learning approach to these naive Bayes text classifiers. We call our new approach locally weighted naive Bayes text classifiers (LWNBTC). LWNBTC weakens the attribute conditional independence assumption made by these naive Bayes text classifiers by applying the locally weighted learning approach. The experimental results show that our locally weighted versions significantly outperform these state-of-the-art naive Bayes text classifiers in terms of classification accuracy.

Keywords: text classification; naive Bayes; locally weighted learning; multinomial naive Bayes; complement naive Bayes; the one-versus-all-but-one model

1. Introduction

A Bayesian network (Pearl 1988) consists of a structural model and a set of conditional probabilities. The structural model is a directed acyclic graph in which nodes represent attributes and arcs represent attribute dependencies. Attribute dependencies are quantified by conditional probabilities for each node given its parents. Bayesian networks are often used for classification problems, in which a learner attempts to construct a classifier from a collection of labelled training instances. The resulting classifiers are called Bayesian network classifiers. Classification on new instances is performed with Bayes' rule (Duda and Hart 1973; Mitchell 1997; Tan, Steinbach, and Kumar 2006) by selecting the class with the largest posterior probability.

The naive Bayes classifier is the simplest of the Bayesian network classifiers, which assumes that all attributes of the instances are independent of each other given the context of the class. This is the so-called attribute conditional independence assumption. Although this assumption is rarely true in many real-world applications, the naive Bayes classifier often performs surprisingly well (Langley, Iba, and Thomas 1992; Domingos and Pazzani

*Corresponding author. Email: zhcai@cug.edu.cn

ISSN 0952–813X print/ISSN 1362–3079 online

http://dx.doi.org/10.1080/0952813X.2012.721010
http://www.tandfonline.com

1997; Friedman, Geiger, and Goldszmidt 1997; Zhang 2005) and even is competitive with state-of-the-art classifiers such as C4.5 (Quinlan 1993). This paradox is explained by many research efforts (Domingos and Pazzani 1997; Friedman et al. 1997). Thanks to its attribute conditional independence assumption, the parameters for each attribute can be estimated separately and easily, and this greatly simplifies learning, especially when the number of attributes is large.

Text classification is just such a domain with a large number of attributes, in which the number of attributes is the number of different words occurring in the document. To our knowledge, the number of different words occurring in the document is often very large, especially when the document is the real-world data from the Web, UseNet and Newswire articles. Therefore, naive Bayes classifiers are widely used to address the text classification problem by many researchers (Sahami 1996; Koller and Sahami 1997; McCallum and Nigam 1998; Nigam, McCallum, Thrun, and Mitchell 1998; Rennie, Shih, Teevan, and Karger 2003). In this article, we call these naive Bayes classifiers used for text classification naive Bayes text classifiers. Their detailed description can be found in Section 2.

As discussed before, all these naive Bayes text classifiers with the strong attribute conditional independence assumption still enjoy high classification accuracy. This fact raises the question of whether a classifier with less restrictive assumption can perform even better. Responding to this question, we propose a locally weighted learning approach to these state-of-the-art naive Bayes text classifiers by combining locally weighted learning (Atkeson and Moore 1997; Frank, Hall, and Pfahringer 2003; Jiang, Li, and Cai 2009) with naive Bayes learning in this article. The basic idea of our approach is building naive Bayes text classifiers on a subset (called local training data) of the training data set, instead of on the whole set. Although the attribute conditional independence assumption made by naive Bayes text classifiers is always violated on the whole training data, it could be expected that the dependencies within the local training data is weaker than that on the whole training data. Thus, the naive Bayes text classifiers built on the local training data could perform better. The experimental results on a large number of widely used text classification benchmark datasets (Han and Karypis 2000; Forman 2003; Witten and Frank 2005; Su and Zhang 2006) show that our locally weighted versions significantly outperforms original naive Bayes text classifiers in terms of classification accuracy.

The rest of this article is organised as follows. In Section 2, we revisit some state-of-the-art naive Bayes text classifiers, such as multinomial naive Bayes (MNB), complement naive Bayes (CNB) and the one-versus-all-but-one (OVA) model. In Section 3, we propose a locally weighted learning approach to these state-of-the-art naive Bayes text classifiers. In Section 4, we report the experimental setup and results in detail. In Section 5, we draw conclusions and outline the main directions for future work.

2. Naive Bayes text classifiers

Due to being fast, easy to implement and relatively effective, naive Bayes text classifiers with the strong attribute conditional independence assumption are well studied and numerous naive Bayes models are proposed (Sahami 1996; Koller and Sahami 1997; McCallum and Nigam 1998; Nigam et al. 1998; Rennie et al. 2003).

Among these models, the multi-variate Bernoulli model is first proposed, which assumes that a document is represented by a vector of binary attributes indicating which words occur and do not occur in the document. When calculating the class membership

probability $P(c|d)$ that a document d belongs to the class c, one multiplies the probability of all the attribute values, including the probability of non-occurrence for words that do not occur in the document. This model is more traditional in the field of Bayesian networks, and is appropriate for text classification tasks that have a fixed number of attributes (McCallum and Nigam 1998). However, the main shortcoming with it is that the information of the number of times a word occurs in a document is not captured.

To overcome the shortcoming confronting the multi-variate Bernoulli model, the multinomial model is proposed by capturing the information of the number of times a word occurs in a document. This multinomial model is widely called MNB. According to the experimental results by McCallum and Nigam (1998), MNB provides on average a 27% reduction in error rate over the multi-variate Bernoulli model at any vocabulary size. Therefore, our research starts from our revisiting to MNB.

Given a test document d, represented by a word vector $\langle w_1, w_2, \ldots, w_m \rangle$, MNB uses Equation (1) to classify the document d.

$$c_{\mathrm{MNB}}(d) = \arg\max_{c \in C} P(c) \left(\sum_{i=1}^{m} f_i \right)! \prod_{i=1}^{m} \frac{P(w_i|c)^{f_i}}{f_i!}, \tag{1}$$

where m is the number of words, w_i $(i = 1, 2, \ldots, m)$ is the ith word occuring in the document d, f_i is the frequency count of word w_i in the document d, $P(c)$ is the prior probability that the document d occurs in the class c, $P(w_i|c)$ is the conditional probability that the word w_i occurs in the class c, C is the set of all possible class labels c and $c(d)$ is the class label of d predicted by MNB.

Because the factorial terms do not make any difference to the classifier's classification accuracy, Equation (1) can be simplified as

$$c_{\mathrm{MNB}}(d) = \arg\max_{c \in C} P(c) \prod_{i=1}^{m} P(w_i|c)^{f_i}. \tag{2}$$

Besides, applying the natural logarithm to Equation (2) does not make any difference to the classifier's classification accuracy yet. Thus, Equation (2) can be further simplified as

$$c_{\mathrm{MNB}}(d) = \arg\max_{c \in C} \left[\log P(c) + \sum_{i=1}^{m} f_i \log P(w_i|c) \right], \tag{3}$$

where the prior probability $P(c)$ and the conditional probability $P(w_i|c)$ can be estimated by Equations (4) and (5) with Laplace smoothing, respectively:

$$P(c) = \frac{\sum_{j=1}^{n} \delta(c_j, c) + 1}{n + l}, \tag{4}$$

$$P(w_i|c) = \frac{\sum_{j=1}^{n} f_{ji} \delta(c_j, c) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ji} \delta(c_j, c) + m}, \tag{5}$$

where n is the number of training documents, l is the number of classes, c_j is the class label of the jth training document, m is the number of words, f_{ji} is the frequency count of word w_i in the jth training document and $\delta(\bullet)$ is a binary function, in which one of its two parameters are identical and zero otherwise.

One systemic problem confronting MNB is that when one class has more training documents than the others, MNB selects poor weights for the decision boundary. This is due to an understudied bias effect that shrinks weights for classes with few training documents. To balance the amount of training documents used per estimate and to deal with skewed training data, a complement class version of MNB, called CNB is proposed (Rennie et al. 2003). CNB uses Equation (6) to classify the document d. Please note that Equation (6) is a little different from that given by Rennie et al. (2003).

$$c_{\mathrm{CNB}}(d) = \arg\max_{c \in C}\left[-\log P(\overline{c}) - \sum_{i=1}^{m} f_i \log P(w_i|\overline{c}) \right], \tag{6}$$

where $\overline{c}$ is the complement classes of the class c (all classes except the class c), the prior probability $P(\overline{c})$ and the conditional probability $P(w_i|\overline{c})$ can be computed by Equations (7) and (8) respectively:

$$P(\overline{c}) = \frac{\sum_{j=1}^{n} \delta(c_j, \overline{c}) + 1}{n + l}, \tag{7}$$

$$P(w_i|\overline{c}) = \frac{\sum_{j=1}^{n} f_{ji}\delta(c_j, \overline{c}) + 1}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ji}\delta(c_j, \overline{c}) + m}. \tag{8}$$

The OVA is a direct combination of MNB and CNB, which uses Equation (9) to classify the document d. Please also note that Equation (9) is a little different from that given by Rennie et al. (2003).

$$c_{\mathrm{OVA}}(d) = \arg\max_{c \in C}\left[(\log P(c) - \log P(\overline{c})) + \sum_{i=1}^{m} f_i(\log P(w_i|c) - \log P(w_i|\overline{c})) \right], \tag{9}$$

where the probabilities $P(c)$, $P(w_i|c)$, $P(\overline{c})$, and $P(w_i|\overline{c})$ are computed using Equations (4), (5), (7), and (8), respectively.

Berger (1999) and Zhang and Oles (2001) have found that OVA performs much better than MNB. Rennie et al. (2003) attribute the improvement with OVA to the use of the complement weights and found that CNB performs better than OVA and MNB since it eliminates the biased MNB weights. To our knowledge, an ensemble classifier generally outperforms each of its base classifiers when the two necessary conditions for ensemble learning are met. Now, OVA is such an ensemble classifier, it should be better than its single classifier CNB. However, our knowledge is in conflict with the conclusion drawn by Rennie et al. (2003). The experimental results in Section 4 seems to support our knowledge.

3. A locally weighted learning approach

In traditional Bayes learning, numerous approaches are proposed to improve the classification accuracy of naive Bayes by weakening its attribute conditional independence assumption (Jiang 2011; Jiang, Wang, and Cai 2012). The basic idea of the local learning approach is building a naive Bayes on the neighbourhood of the test instance, instead of on the whole training data.

The local learning approach is actually a kind of training data selection approach (Jiang 2011; Jiang et al. 2012), namely the selected training instances are dropped into the neighbourhood of the test instance, which helps to weaken the effects of attribute dependencies that may exist in the whole training data. As naive Bayes requires relatively little data for training, the neighbourhood can be kept small, thereby reducing the chance of encountering strong dependencies (Kohavi 1996; Frank et al. 2003). Therefore, although the attribute conditional independence assumption of naive Bayes is always violated on the whole training data, it could be expected that the dependencies within the neighbourhood of the test instance is much weaker than that on the whole training data and thus the conditional independence assumptions required for naive Bayes are likely to be true.

In addition, it has been observed that the classification accuracy of naive Bayes does not scale up in large data sets (Kohavi 1996). This feature makes it especially fit to be a local model embedded into another model. For example, Frank et al. (2003) propose an improved algorithm called locally weighted naive Bayes (LWNB) by applying locally weighted learning (Atkeson and Moore 1997) to naive Bayes. In LWNB, the k nearest neighbours of the test instance are firstly found and each of them is weighted in terms of its distance to the test instance. Then a local naive Bayes is built from the locally weighted training instances.

Locally weighted learning (Atkeson and Moore 1997) is a meta learning method, which has been successfully used to improve some fast and effective algorithms. In addition to LWNB, locally weighted linear regression (LWLR) (Mitchell 1997) and locally weighted C4.4 (LWC4.4) (Jiang et al. 2009) are other two cases in point. Just as discussed before, naive Bayes text classifiers, such as MNB, CNB and the OVA model, are fast and relatively effective algorithms for addressing the text classification problems. This fact raises the question of whether such a locally weighted learning algorithm can be used to improve the classification performance of these state-of-the-art naive Bayes text classifiers. Responding to this question, we propose a locally weighted learning approach to these state-of-the-art naive Bayes text classifiers by combining locally weighted learning (Atkeson and Moore 1997) with naive Bayes learning. We expect that the attribute conditional independence assumption made by these naive Bayes text classifiers are weakened within the local training data. We call our new approach locally weighted naive Bayes text classifiers (LWNBTC).

LWNBTC uses naive Bayes text classifiers in exactly the same way as naive Bayes is used in LWNB: a local naive Bayes text classifier is built on the neighbourhood of a test document. The k nearest training documents in this neighbourhood are first found via the k-nearest neighbour algorithm and each of them is weighted according to its distance to the test document, with less weights being assigned to the documents that are further from the test document. Then a local naive Bayes text classifier is built from the locally weighted training documents. The detailed algorithm can be depicted below.

Algorithm: LWNBTC ($\mathbf{D}, d, k$)
Input: a training document set $\mathbf{D}$, a test document d, and the neighbourhood size k
Output:the predicted class label of d

(1) Use Equation (10) to calculate the distance $\text{dis}(d, d_j)$ between d and each training document d_j.
(2) Find d's k nearest neighbours $d_1, d_2, \ldots, d_k$.
(3) Initialise the locally weighted training documents set $\mathbf{LWD} = \{d_1, d_2, \ldots, d_k\}$.

(4) For each neighbour d_j, $j = 1, 2, \ldots, k$.
(5) Set the weight of d_j to W_j, defined by Equation (16).
(6) Build naive Bayes text classifiers on **LWD**.
(7) Use the built naive Bayes text classifiers to predict the class label of d.
(8) Return the class label of d.

As seen from our LWNBTC algorithm, the local training document set is determined using the k-nearest neighbour algorithm. A user-specified parameter k controls how many nearest training documents are used. Thus, similar to other locally weighted algorithms, our LWNBTC also is a k-related algorithm. Fortunately, we get almost the same conclusion as that of LWNB (Frank et al. 2003), LWLR (Mitchell 1997) and LWC4.4 (Jiang et al. 2009): LWNBTC is not particularly sensitive to the choice of k as long as it is not too small (generally, it is not less than 30, we set it to 30 in our later experiments). This characteristic makes our LWNBTC very attractive to other k-related algorithms, which require fine-tuning of the choice of k to achieve good results.

Another practical problem is how to define the distance between each pair of documents. In our algorithm, we define the distance between each pair of documents x and y as:

$$\mathrm{dis}(x, y) = \sqrt{\sum_{i=1}^{m}\left[I(C; w_i)\frac{f_i(x) - f_i(y)}{\max_{f_i} - \min_{f_i}}\right]^2}, \tag{10}$$

where $f_i(x)$ and $f_i(y)$ are the frequency count of word w_i in the document x and y, respectively, $\max_{f_i}$ and $\min_{f_i}$ respectively, are the maximum and minimum values of f_i in all training documents and $I(C; w_i)$ is the mutual information between the class variable C and the ith word w_i, which can be defined as

$$I(C; w_i) = \sum_{c \in C} P(c, w_i) \log \frac{P(c, w_i)}{P(c)P(w_i)}, \tag{11}$$

where $P(c)$ is the number of times all words appeared in the documents belonging to class c divided by the total number of all words occurrences, $P(w_i)$ is the number of times the word w_i appears in all the documents divided by the total number of all words occurrences and $P(c, w_i)$ is the number of times the word w_i appeared in the documents belonging to c divided by the total number of all words occurrences.

Our LWNBTC algorithm also is a meta-learning algorithm, it should be appropriate for all naive Bayes text classifiers. In our later experiments, we mainly include there state-of-the-art naive Bayes text classifiers: MNB, CNB and OVA. For simplicity, we denoted them by LWMNB, LWCNB, and LWOVA, respectively. When our LWNBTC is applied to MNB, CNB and OVA, the classification equations are similar to Equations (3), (6) and (9), respectively. The only difference is the equation for computing the probabilities $P(c)$, $P(w_i|c)$, $P(\overline{c})$ and $P(w_i|\overline{c})$. In our LWNBTC, Equations (4), (5), (7) and (8) are replaced by Equations (12), (13), (14) and (15), respectively.

$$P(c) = \frac{\sum_{j=1}^{k} W_j \delta(c_j, c) + 1}{\sum_{j=1}^{k} W_j + l}, \tag{12}$$

$$P(w_i|c) = \frac{\sum_{j=1}^{k} W_j f_{ji} \delta(c_j, c) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{k} W_j f_{ji} \delta(c_j, c) + m}, \tag{13}$$

$$P(\overline{c}) = \frac{\sum_{j=1}^{k} W_j \delta(c_j, \overline{c}) + 1}{\sum_{j=1}^{k} W_j + l}, \tag{14}$$

$$P(w_i|\overline{c}) = \frac{\sum_{j=1}^{k} W_j f_{ji} \delta(c_j, \overline{c}) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{k} W_j f_{ji} \delta(c_j, \overline{c}) + m}, \tag{15}$$

where k is the number of the neighbours, W_j is the weight of the jth neighbour d_j, which can be defined as

$$W_j = \frac{1.0}{1.0 + \mathrm{dis}(d, d_j)^2}. \tag{16}$$

Please note that the probabilities $P(c)$, $P(w_i|c)$, $P(\overline{c})$ and $P(w_i|\overline{c})$ are computed from the neighbourhood of the test document d, namely d's k nearest neighbours $d_1, d_2, \ldots, d_k$. Because the neighbourhoods are different for different test documents, those probabilities need to be computed once for each different being predicted document d.

Our experimental results on a large number of widely used text classification benchmark datasets (Han and Karypis 2000; Forman 2003; Witten and Frank 2005; Su and Zhang 2006; Jiang et al. 2012) show that our locally weighted versions significantly outperforms original naive Bayes text classifiers in terms of classification accuracy. However, our improvements turn an eager learning algorithm into a lazy learning one. Like all the other lazy learning algorithms, LWNBTC simply stores training documents and defers the effort involved in learning until prediction time. When called upon to predict a test document, LWNBTC constructs naive Bayes text classifiers using a weighted set of training documents in the neighbourhood of the test document. Compared to original naive Bayes text classifiers, LWNBTC needs to find and weight the k nearest neighbours for each test document (the first five steps), which has the time complexity of $O(nml + nlogn + k)$. Once these steps are completed, the training-time complexity and the test-time complexity of the local naive Bayes text classifiers are much lower than those of original naive Bayes text classifiers (because the value of k is much smaller than the value of n).

4. Experiments and results

The purpose of these experiments is to evaluate the classification performance of our new approach LWNBTC. Therefore, we conduct our experiments to compare LWNBTC (including LWMNB, LWCNB and LWOVA) with each original naive Bayes text classifier, including MNB, CNB and the OVA, in terms of classification accuracy.

We implement our LWNBTC in Weka platform (Witten and Frank 2005) and set the number of neighbours (the size of neighbourhood) of it to 30. We ran our experiments on 19 widely used text classification benchmark datasets published on the main web site of Weka platform (Witten and Frank 2005), which represent a wide range of domains and data characteristics. The description of these 19 datasets is shown in Table 1. Other detailed source information of these datasets is described by Han and Karypis (2000).

Table 1. Descriptions of text classification benchmark datasets used in our experiments.

Dataset	Documents number	Words number	Classes number
fbis	2463	2000	17
la1s	3204	31,472	6
la2s	3075	31,472	6
new3s	9558	26,832	44
oh0	1003	3182	10
oh10	1050	3238	10
oh15	913	3100	10
oh5	918	3012	10
ohscal	11,162	11,465	10
re0	1657	3758	25
re1	1504	2886	13
tr11	414	6429	9
tr12	313	5804	8
tr21	336	7902	6
tr23	204	5832	6
tr31	927	10,128	7
tr41	878	7454	10
tr45	690	8261	10
wap	1560	8460	20

In our experiments, the classification accuracy of each algorithm on each dataset is obtained via five runs of five-fold cross-validation. Runs with the various algorithms are carried out on the same training sets and evaluated on the same test sets. In particular, the cross-validation folds are the same for all the experiments on each data set. Tables 2–4 show the classification accuracy of each classifier on each dataset. The symbols $\circ$ and $\bullet$ in the tables, respectively, denote statistically significant improvement or degradation over the original naive Bayes text classifiers with the $p = 0.05$ significance level (Nadeau and Bengio 2003). Besides, the averaged classification accuracy values and $w/t/l$ values are summarised at the bottom of the tables. Each entry $w/t/l$ in the tables means that our improved versions win on w data sets, tie on t data sets and lose on l data sets, compared to the original naive Bayes text classifiers.

Finally, for additional insight into the results, we conducted a corrected paired two-tailed t-test with the $p = 0.05$ significance level (Nadeau and Bengio 2003) to compare each pair of algorithms. The detailed compared results are presented in Tables 5 and 6. Each number in Table 5 indicates how many datasets the algorithm in the corresponding column achieves significant wins with regard to the algorithm in the corresponding row. In Table 6, the first column is the difference between the total number of wins and the total number of losses that the algorithm in the corresponding row achieves comparing with all the other algorithms, which is used to generate the ranking. The second and third columns represent the total numbers of wins and losses, respectively.

From our experimental results, we can see that our locally weighted versions significantly outperform the original naive Bayes text classifiers. Now, let us summarise the highlights briefly as follows:

(1) CNB is a little better than MNB with seven wins and four losses, and OVA is much better than MNB with 10 wins and zero losses.

Table 2. Experimental results for MNB versus LWMNB: classification accuracy and standard deviation.

Dataset	MNB	LWMNB
fbis	76.90 ± 1.56	81.57 ± 1.49 ○
la1s	88.22 ± 1.05	90.44 ± 1.01 ○
la2s	89.65 ± 0.94	91.70 ± 0.86 ○
new3s	79.12 ± 0.79	84.85 ± 0.75 ○
oh0	89.59 ± 1.87	90.17 ± 1.57
oh10	80.44 ± 1.75	80.91 ± 1.79
oh15	83.55 ± 2.28	83.55 ± 2.26
oh5	86.42 ± 2.42	87.30 ± 2.03
ohscal	74.52 ± 0.78	75.01 ± 0.78
re0	79.92 ± 1.80	81.04 ± 1.84
re1	82.63 ± 1.82	84.80 ± 1.59 ○
tr11	84.20 ± 2.00	87.78 ± 2.28 ○
tr12	80.57 ± 4.46	86.84 ± 3.79 ○
tr21	62.04 ± 7.69	90.23 ± 2.59 ○
tr23	71.06 ± 5.26	85.39 ± 4.95 ○
tr31	94.56 ± 1.70	96.44 ± 1.47 ○
tr41	94.44 ± 1.66	95.47 ± 1.28
tr45	83.30 ± 3.47	92.20 ± 2.08 ○
wap	80.23 ± 1.92	82.56 ± 2.14 ○
Average	82.18	86.75
w/*t*/*l*	–	12/7/0

Table 3. Experimental results for CNB versus LWCNB: classification accuracy and standard deviation.

Dataset	CNB	LWCNB
fbis	76.64 ± 1.16	80.28 ± 1.17 ○
la1s	86.22 ± 0.87	89.20 ± 1.12 ○
la2s	87.93 ± 1.05	90.39 ± 0.90 ○
new3s	74.63 ± 0.82	83.17 ± 0.67 ○
oh0	92.02 ± 1.59	92.98 ± 1.53 ○
oh10	81.62 ± 1.91	82.97 ± 2.16 ○
oh15	84.49 ± 2.07	85.10 ± 2.04
oh5	90.52 ± 2.27	91.63 ± 2.08
ohscal	76.18 ± 0.57	77.84 ± 0.66 ○
re0	81.80 ± 1.78	84.36 ± 2.06 ○
re1	84.89 ± 1.26	87.15 ± 1.17 ○
tr11	82.27 ± 2.73	88.07 ± 2.08 ○
tr12	86.26 ± 4.27	88.37 ± 3.62
tr21	83.32 ± 10.04	89.82 ± 2.76
tr23	69.00 ± 6.53	85.29 ± 5.73 ○
tr31	94.46 ± 1.49	96.55 ± 1.25 ○
tr41	94.17 ± 1.27	95.60 ± 1.30 ○
tr45	87.19 ± 2.42	93.28 ± 2.31 ○
wap	77.29 ± 1.72	80.62 ± 1.72 ○
Average	83.73	87.51
w/*t*/*l*	–	15/4/0

Table 4. Experimental results for OVA versus LWOVA: classification accuracy and standard deviation.

Dataset	OVA	LWOVA
fbis	80.71 ± 1.59	83.79 ± 1.56 ○
la1s	88.22 ± 1.06	90.34 ± 1.14 ○
la2s	90.06 ± 0.92	91.84 ± 0.97 ○
new3s	79.58 ± 0.84	85.12 ± 0.76 ○
oh0	91.16 ± 1.85	91.78 ± 1.62
oh10	81.98 ± 2.32	82.53 ± 2.27
oh15	84.45 ± 1.98	84.62 ± 2.25
oh5	89.19 ± 2.34	89.56 ± 2.03
ohscal	75.66 ± 0.78	76.67 ± 0.69 ○
re0	81.20 ± 1.77	82.30 ± 2.15
re1	84.09 ± 1.57	85.96 ± 1.27 ○
tr11	85.61 ± 2.20	87.88 ± 2.41
tr12	83.51 ± 4.93	87.60 ± 3.57
tr21	71.49 ± 7.43	89.64 ± 2.64 ○
tr23	71.34 ± 5.81	85.78 ± 5.23 ○
tr31	94.91 ± 1.47	96.44 ± 1.41 ○
tr41	94.65 ± 1.47	95.49 ± 1.25
tr45	86.43 ± 2.69	92.81 ± 2.19 ○
wap	79.94 ± 1.92	82.41 ± 2.04 ○
Average	83.90	87.50
$w/t/l$	–	11/8/0

Table 5. Summary test on classification accuracy ($p = 0.05$).

	MNB	CNB	OVA	LWMNB	LWCNB	LWOVA
MNB	–	7	10	12	15	17
CNB	4	–	6	9	15	10
OVA	0	1	–	8	12	11
LWMNB	0	3	2	–	5	7
LWCNB	0	0	0	4	–	5
LWOVA	0	0	0	0	4	–

Table 6. Ranking test on classification accuracy ($p = 0.05$).

Resultset	Wins − losses	Wins	Losses
LWOVA	46	50	4
LWCNB	42	51	9
LWMNB	16	33	17
OVA	−14	18	32
CNB	−33	11	44
MNB	−57	4	61

Table 7. Experimental results for SMO versus LWMNB, LWCNB, and LWOVA: classification accuracy and standard deviation.

Dataset	SMO	LWMNB	LWCNB	LWOVA
fbis	77.55 ± 1.61	81.57 ± 1.49 ∘	80.28 ± 1.17 ∘	83.79 ± 1.56 ∘
la1s	83.79 ± 1.65	90.44 ± 1.01 ∘	89.20 ± 1.12 ∘	90.34 ± 1.14 ∘
la2s	86.00 ± 1.28	91.70 ± 0.86 ∘	90.39 ± 0.90 ∘	91.84 ± 0.97 ∘
new3s	70.99 ± 1.00	84.85 ± 0.75 ∘	83.17 ± 0.67 ∘	85.12 ± 0.76 ∘
oh0	80.92 ± 2.17	90.17 ± 1.57 ∘	92.98 ± 1.53 ∘	91.78 ± 1.62 ∘
oh10	74.51 ± 2.18	80.91 ± 1.79 ∘	82.97 ± 2.16 ∘	82.53 ± 2.27 ∘
oh15	73.21 ± 3.20	83.55 ± 2.26 ∘	85.10 ± 2.04 ∘	84.62 ± 2.25 ∘
oh5	77.34 ± 2.94	87.30 ± 2.03 ∘	91.63 ± 2.08 ∘	89.56 ± 2.03 ∘
ohscal	73.78 ± 1.07	75.01 ± 0.78 ∘	77.84 ± 0.66 ∘	76.67 ± 0.69 ∘
re0	74.65 ± 1.91	81.04 ± 1.84 ∘	84.36 ± 2.06 ∘	82.30 ± 2.15 ∘
re1	72.17 ± 2.08	84.80 ± 1.59 ∘	87.15 ± 1.17 ∘	85.96 ± 1.27 ∘
tr11	75.02 ± 4.79	87.78 ± 2.28 ∘	88.07 ± 2.08 ∘	87.88 ± 2.41 ∘
tr12	70.85 ± 5.46	86.84 ± 3.79 ∘	88.37 ± 3.62 ∘	87.60 ± 3.57 ∘
tr21	79.70 ± 2.35	90.23 ± 2.59 ∘	89.82 ± 2.76 ∘	89.64 ± 2.64 ∘
tr23	75.00 ± 6.04	85.39 ± 4.95 ∘	85.29 ± 5.73 ∘	85.78 ± 5.23 ∘
tr31	90.38 ± 2.10	96.44 ± 1.47 ∘	96.55 ± 1.25 ∘	96.44 ± 1.41 ∘
tr41	87.15 ± 3.03	95.47 ± 1.28 ∘	95.60 ± 1.30 ∘	95.49 ± 1.25 ∘
tr45	80.75 ± 4.09	92.20 ± 2.08 ∘	93.28 ± 2.31 ∘	92.81 ± 2.19 ∘
wap	81.31 ± 2.00	82.56 ± 2.14	80.62 ± 1.72	82.41 ± 2.04
Average	78.16	86.75	87.51	87.50
w/*t*/*l*	–	18/1/0	18/1/0	18/1/0

(2) OVA is better than CNB with six wins and one loss. This conclusion is in conflict with the conclusion drawn by Rennie et al. (2003) because they find that CNB completely overwhelms MNB and even performs better than OVA.
(3) Our locally weighted versions LWMNB, LWCNB and LWOVA significantly outperform the original naive Bayes text classifiers MNB, CNB and OVA with 12, 15 and 11 wins, respectively and surprisingly zero losses.
(4) Although our improvement is significant, it turns an eager text classification algorithm into a lazy one, which incurs higher time complexity. Therefore, in real-world application, an appropriate algorithm should be chosen according to different text classification tasks. Generally speaking, our LWCNB and LWOVA are preferred when high classification accuracy is the sole concern. When the computational cost and/or comprehensibility are also important, CNB and OVA should be considered first.

In another group of experiments, we observe the classification performance of our LWNBTC with different k values and get almost the same conclusions as long as the choice of k is not too small (generally, it is not less than 30). This characteristic makes our LWNBTC very attractive to other k-related algorithms, which require fine-tuning of k to achieve good results. Due to the limit of space, we have not presented the detailed experimental results here.

Besides, in order to further validate the effectiveness of our new proposed algorithms, we design a group of experiment to compare them with the well-known SMO algorithm (Platt 1998; Keerthi, Shevade, Bhattacharyya, and Murthy 2001), which is an improved SVM classification algorithm. From the detailed compared results in Table 7, we can see

that all our new proposed algorithms LWMNB, LWCNB and LWOVA significantly outperform SMO with 18 wins and zero losses. This result is acceptable, because SMO is a general classification algorithm instead of a classification algorithm specially designed for text classification.

5. Conclusions and future work

In this article, we revisit some state-of-the-art naive Bayes text classifiers and empirically compare their classification performance on a large number of widely used text classification benchmark datasets. Then, we propose a locally weighted learning approach to these state-of-the-art naive Bayes text classifiers by combining locally weighted learning with naive Bayes learning. We expect that the attribute conditional independence assumption made by these naive Bayes text classifiers are weakened within the local training data. We call our new approach LWNBTC. The experimental results show that our locally weighted versions significantly outperform original naive Bayes text classifiers in terms of classification accuracy.

In traditional Bayes learning, numerous approaches are proposed to improve the classification accuracy of naive Bayes by weakening its attribute conditional independence assumption. In this article, we focus our attention only on the locally weighted learning approach. Therefore, applying some other improved approaches, such as the structure extension approach and the attribute selection approach, to these state-of-the-art naive Bayes text classifiers is a main research direction for our future work. Besides, how to calculate the distance between a pair of documents is a key problem. Therefore, applying some newly proposed distance functions, such as ODVDM (Li and Li 2011) and MSFM (Li and Li 2012), to LWNBTC is another research direction for our future work.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. The work was partially supported by the National Natural Science Foundation of China (No. 60905033 and No. 61203287), the Provincial Natural Science Foundation of Hubei (No. 2011CDA103) and the Fundamental Research Funds for the Central Universities (No. CUG110405 and No. CUG090109).

References

Atkeson, C.G., and Moore, A.W. (1997), 'Locally Weighted Learning', *Artificial Intelligence Review*, 11, 11–73.

Berger, A. (1999), 'Error-correcting Output Coding for Text Classification', in *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden.

Domingos, P., and Pazzani, M. (1997), 'On the Optimality of the Simple Bayesian Classifier Under Zero–One Loss', *Machine Learning*, 29, 103–130.

Duda, R.O., and Hart, P.E. (1973), *Pattern classification and scene analysis*, New York, NY: Wiley and Sons.

Forman, G. (2003), 'An Extensive Empirical Study of Feature Selection Metrics for Text Classification', *Journal of Machine Learning Research*, 3, 1289–1305.

Frank, E., Hall, M., and Pfahringer, B. (2003), 'Locally Weighted Naive Bayes', in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann, pp. 249–256.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997), 'Bayesian Network Classifiers', *Machine Learning*, 29, 131–163.

Han, E., and Karypis, G. (2000), 'Centroid-based Document Classification: Analysis and Experimental Results', in *Proceedings of the 4th European Conference on the Principles of Data Mining and Knowledge Discovery*, Berlin: Springer Press, pp. 424–431.

Jiang, L. (2011), 'Random One-dependence Estimators', *Pattern Recognition Letters*, 32, 532–539.

Jiang, L., Li, C., and Cai, Z. (2009), 'Decision Tree with Better Class Probability Estimation', *International Journal of Pattern Recognition and Artificial Intelligence*, 23, 745–763.

Jiang, L., Wang, D., and Cai, Z. (2012), 'Discriminatively Weighted Naive Bayes and its Application in Text Classification', *International Journal on Artificial Intelligence Tools*, 21, 1250007-1–1250007-19.

Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., and Murthy, K.R.K. (2001), 'Improvements to Platt's Smoalgorithm for Svm Classifier Design', *Neural Computation*, 13, 637–649.

Kohavi, R. (1996), 'Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-tree Hybrid', in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Palo Alto, CA: AAAI Press, pp. 202–207.

Koller, D., and Sahami, M. (1997), 'Hierarchically Classifying Documents Using Very Few Words', in *Proceedings of the Fourteenth International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann, pp. 170–178.

Langley, P., Iba, W., and Thomas, K. (1992), 'An Analysis of Bayesian Classifiers', in *Proceedings of the Tenth National Conference of Artificial Intelligence*, Palo Alto, CA: AAAI Press, pp. 223–228.

Li, C., and Li, H. (2011), 'One Dependence Value Difference Metric', *Knowledge-Based Systems*, 24, 589–594.

Li, C., and Li, H. (2012), 'A Modified Short and Fukunaga Metric Based on the Attribute Independence Assumption', *Pattern Recognition Letters*, 33, 1213–1218.

McCallum, A., and Nigam, K. (1998), 'A Comparison of Event Models for Naive Bayes Text Classification', in *Working Notes of the AAAI/ICML Workshop on Learning for Text*, Palo Alto, CA: AAAI Press, pp. 41–48.

Mitchell, T.M. (1997), *Machine Learning* (1st ed.), New York: McGraw-Hill.

Nadeau, C., and Bengio, Y. (2003), 'Inference for the Generalisation Error', *Machine Learning*, 52, 239–281.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (1998), 'Learning to Classify Text From Labeled and Unlabeled Documents', in *Proceedings of the Fifteenth National Conference of Artificial Intelligence*, Palo Alto, CA: AAAI Press, pp. 792–799.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems*, San Francisco, CA: Morgan Kaufmann.

Platt, J. (1998), 'Fast training of support vector machines using sequential minimal optimization', in *Advances in Kernel Methods – Support Vector Learning*, eds. B. Schoelkopf, C. Burges and A. Smola, Cambridge, MA: MIT Press, pp. 185–208.

Quinlan, J.R. (1993), *C4.5: Programs for Machine Learning* (1st ed.), San Mateo, CA: Morgan Kaufmann.

Rennie, J.D., Shih, L., Teevan, J., and Karger, D.R. (2003), 'Tackling the Poor Assumptions of Naive Bayes Text Classifiers', in *Proceedings of the Twentieth International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann, pp. 616–623.

Sahami, M. (1996), 'Learning Limited Dependence Bayesian Classifiers', in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Palo Alto, CA: AAAI Press, pp. 335–338.

Su, J., and Zhang, H. (2006), 'A Fast Decision Tree Learning Algorithm', in *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, Palo Alto, CA: AAAI Press, pp. 500–505.

Tan, P.N., Steinbach, M., and Kumar, V. (2006), *Introduction to data mining* (1st ed.), Boston, MA: Pearson.

Witten, I.H., and Frank, E. (2005), *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.), San Francisco, CA: Morgan Kaufmann.

Zhang, H. (2005), 'Exploring Conditions for the Optimality of Naive Bayes', *International Journal of Pattern Recognition and Artificial Intelligence*, 19, 183–198.

Zhang, T., and Oles, F.J. (2001), 'Text Categorization Based on Regularized Linear Classification Methods', *Information Retrieval*, 4, 5–31.