Expert Systems with Applications 39 (2012) 11022–11028

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Not so greedy: Randomly Selected Naive Bayes

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

[a] Department of Computer Science, China University of Geosciences, Wuhan, Hubei 430074, China
[b] Department of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada E3B 5A3
[c] Department of Electronic Engineering, China University of Geosciences, Wuhan, Hubei 430074, China

ARTICLE INFO

Keywords:
Naive Bayes
Attribute selection
Random search
Classification
Ranking
Class probability estimation

ABSTRACT

Many approaches are proposed to improve Naive Bayes, among which the attribute selection approach has demonstrated remarkable performance. Algorithms for attribute selection fall into two broad categories: filters and wrappers. Filters use the general data characteristics to evaluate the selected attribute subset before the learning algorithm is run, while wrappers use the learning algorithm itself as a black box to evaluate the selected attribute subset. In this paper, we work on the attribute selection approach of wrapper and propose an improved Naive Bayes algorithm by carrying a random search through the whole space of attributes. We simply called it Randomly Selected Naive Bayes (RSNB). In order to meet the need of classification, ranking, and class probability estimation, we discriminatively design three different versions: RSNB-ACC, RSNB-AUC, and RSNB-CLL. The experimental results based on a large number of UCI datasets validate their effectiveness in terms of classification accuracy (ACC), area under the ROC curve (AUC), and conditional log likelihood (CLL), respectively.

1. Introduction

A Bayesian network 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 given set of training instances with class labels. Given a test instance x, represented by a vector $\langle a_1, a_2, \ldots, a_m \rangle$, Bayesian network classifiers use Eq. (1) to classify it.

$$c(x) = \arg\max_{c \in C} P(c)P(a_1, a_2, \ldots, a_m | c), \tag{1}$$

where m is the number of attributes, a_j ($j = 1, 2, \ldots, m$) is the value of the jth attribute, C is the set of all possible class labels c, and $c(x)$ is the class label of x predicted by Bayesian network classifiers.

Assume that all attributes are independent given the class (called the attribute conditional independence assumption), the resulting classifier is called the Naive Bayes, simply NB. NB uses Eq. (2) to classify x.

$$c(x) = \arg\max_{c \in C} P(c) \prod_{j=1}^{m} P(a_j | c), \tag{2}$$

where the prior probability $P(c)$ and the conditional probability $P(a_j|c)$ can be computed by Eqs. (3) and (4), respectively.

$$P(c) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + 1}{n + n_c}, \tag{3}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n} \delta(a_{ij}, a_j)\delta(c_i, c) + 1}{\sum_{i=1}^{n} \delta(c_i, c) + n_j}, \tag{4}$$

where n is the number of training instances, n_c is the number of classes, n_j is the number of values of the jth attribute, c_i is class label of the ith training instance, a_{ij} is the jth attribute value of the ith training instance, and $\delta(\bullet)$ is a binary function, which is one if its two parameters are identical and zero otherwise.

Fig. 1 shows graphically the structure of Naive Bayes. In Naive Bayes, each attribute node has the class node as its parent, but does not have any parent from attribute nodes. Because the values of $P(c)$ and $P(a_j|c)$ can be easily estimated from training instances, Naive Bayes is easy to construct. Naive Bayes is the simplest form of Bayesian networks. It is obvious that the attribute conditional independence assumption made by Naive Bayes is rarely true in reality, which would harm its performance in the applications with complex attribute dependencies.

In order to weaken its attribute conditional independence assumption, many approaches have been presented. The related work can be broadly divided into five main categories (Jiang, Cai, & Wang, 2010): (1) structure extension (Bouchaala, Masmoudi, Gargouri, & Rebai, 2010; Liu, Yue, & Li, 2011; Park & Cho, 2012); (2) attribute weighting; (3) attribute selection; (4) instance weighting; (5) instance selection, also called local learning. In this

[*] Corresponding authors. Tel./fax: +86 27 67883716.
E-mail addresses: ljiang@cug.edu.cn (L. Jiang), zhcai@cug.edu.cn (Z. Cai).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.03.022

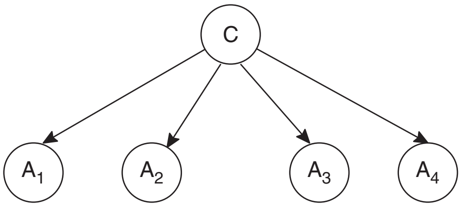

Fig. 1. An example of naive Bayes.

paper, we work on the approach of attribute selection and propose an improved Naive Bayes algorithm by carrying a random search through the whole space of attributes. We simply called it Randomly Selected Naive Bayes (RSNB).

In many real-world applications, ranking and class probability estimation are as important as classification. Therefore, we discriminatively design three different versions: RSNB-ACC, RSNB-AUC, and RSNB-CLL to meet the need of classification, ranking, and class probability estimation, respectively. The experimental results based on a large number of UCI datasets validate their effectiveness in terms of classification accuracy (ACC), area under the ROC curve (AUC) (Bradley, 1997; Hand & Till, 2001), and conditional log likelihood (CLL) (Grossman & Domingos, 2004; Jiang, Li, & Cai, 2009), respectively.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the related work on improving Naive Bayes via attribute selection. In Section 3, we propose our improved algorithm Randomly Selected Naive Bayes (RSNB). In Section 4, we report the experimental setup and results in detail. In Section 5, we draw conclusions.

2. Related work

The attribute selection approach selects an attribute subset from the whole attribute space and only uses the selected attribute subset to build Naive Bayes. Thus, Eq. (2) can be replaced by Eq. (5).

$$c(x) = \arg\max_{c\in C} P(c)\prod_{j=1}^{k} P(a_j|c), \tag{5}$$

where a_j ($j = 1,2,\ldots,k$), respectively is the value of the selected attribute A_j ($j = 1,2,\ldots,k$), k is the number of selected attributes.

Why does the attribute selection approach work well? Let us look back at the example given in our previous work (Jiang & Zhang, 2006). Assume that the attribute set $A = \{A_1,A_2,A_3,A_4\}$, in which A_2 and A_4 are completely dependent on A_1 and A_3 respectively (that is, $A_1 = A_2$, $A_3 = A_4$), and A_1 and A_2 are completely independent from A_3 and A_4. Then the true class probability distribution is:

$$P(A_1,A_2,A_3,A_4,C) = P(C)P(A_1|C)P(A_3|C). \tag{6}$$

However, the class probability estimation produced by Naive Bayes is:

$$P_{NB}(A_1,A_2,A_3,A_4,C) = P(C)P^2(A_1|C)P^2(A_3|C). \tag{7}$$

Obviously, the class probability estimation of Naive Bayes is inaccurate. However, if we do attribute selection and deploy Naive Bayes on the selected attribute subset A_1, A_3, the resulting Naive Bayes represents exactly the true class probability distribution. Thus, we believe that the attribute selection approach should work well for class probability estimation, and class membership probability-based classification and ranking.

Now, the only question left to answer is how to select the best attribute subset from the whole attribute space. In order to address this problem, many attribute selection algorithms are proposed and fall into two broad categories: filters and wrappers. Filters (Hall, 2000) use the general data characteristics to evaluate the selected attribute subset before the learning algorithm is run, while wrappers (Kohavi & John, 1997) use the learning algorithm itself as a black box to evaluate the selected attribute subset. Even so, both filters and wrappers can be encompassed within a common architecture shown in Fig. 2 (Tan, Steinbach, & Kumar, 2006). In this architecture, attribute subset search and attribute subset evaluation are two of the most important parts, and many attribute selection algorithms focus on them.

For example, Hall (2000) proposes a correlation-based filter algorithm for attribute selection (CFS). The central hypothesis of CFS is that good attribute subsets contain attributes that are highly correlated with the class variable, yet uncorrelated with each other. This method heuristically searches an attribute subset through a correlation based approach, and uses Eq. (8) to evaluate the merit of an attribute subset S containing k attributes.

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}}, \tag{8}$$

where $\overline{r_{cf}}$ is the average attribute-class correlation, and $\overline{r_{ff}}$ is the attribute-attribute inter-correlation. Obviously, the heuristic merit tries to search an attribute subset with bigger $\overline{r_{cf}}$ by removing irrelevant attributes and smaller $\overline{r_{ff}}$ by removing redundant attributes. After this, Lei and Liu (2004) propose another filter algorithm using a correlation-based approach, in which an entropy-based measure is used to indicate the correlation between each pair of attributes. Besides, they discussed the definition of strong relevant attributes, weak relevant attributes, irrelevant attributes and redundant attributes in detail, and introduced a framework to discriminate the irrelevant and redundant attributes. The central assumption of the method is that an optimal attribute subset should contains all strong relevant attributes and weak relevant but non-redundant attributes.

Ratanamahatana and Gunopulos (2003) propose another filter algorithm for attribute selection. They select relevant attributes by building decision trees. At fist, they ran standard C4.5 decision tree on the data sets 5 times and only select the attributes appearing in the first 3 levels of the simplified decision tree as relevant attributes. Then, they form a union of all the attributes from the 5 rounds. At last, they run Naive Bayes on the training and test data using only the selected attributes. One of the problems with using C4.5 to generate decision trees when there are too few training examples available is that it might give a constant decision (for example, classify all examples as Democrat in the voting domain) without generating the decision tree. In this case, the training set is re-sampled until a non-constant decision tree is produced.

Different from filter algorithms, wrapper algorithms (Kohavi & John, 1997) use the learning algorithm itself as a black box to evaluate the selected attribute subset. Thus, the most important

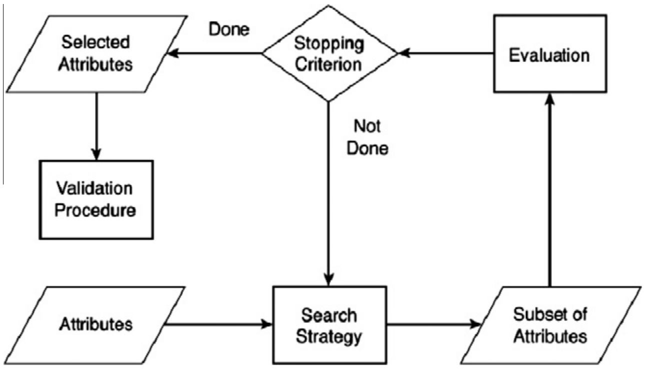

Fig. 2. An architecture of attribute subset selection.

part in wrapper algorithms is attribute subset search. For example, Langley and Sage (1994) propose a wrapper algorithm for attribute selection called selective Bayesian classifiers (SBC). SBC uses a forward greedy search to select an attribute subset through the whole space of attributes and uses Naive Bayes' classification accuracy to evaluate alternative attribute subsets and considers adding each unselected attribute which can improve Naive Bayes' classification accuracy at most on each iteration. Their experimental results proved their hypotheses that their algorithm will improve Naive Bayes' classification accuracy in domains that involve correlated attributes without reducing Naive Bayes' classification accuracy in domains that do not.

Our previous work (Jiang, Zhang, Cai, & Su, 2005) proposes another wrapper algorithm for attribute selection called Evolutional Naive Bayes (ENB). ENB conducts a genetic search to select an attribute subset through the whole space of attributes. ENB uses bit strings to encode hypotheses. the length of a bit string is the number of attributes and each bit represents the status of an attribute. "1" indicates that this attribute is selected and "0" indicates that this attribute is not selected. Besides, ENB chooses the classification accuracy of Naive Bayes deployed on the selected attribute subset as the fitness function. Our previous experimental results validate the effectiveness of ENB. A potential problem with ENB is that, too many parameters are involved and the classification performance of ENB is sensitive to these parameters.

3. Randomly Selected Naive Bayes

Our research starts from our comments to selective Bayesian classifiers (SBC) (Langley & Sage, 1994), which is a well-known attribute selection algorithm for Naive Bayes. SBC always adds a best attribute (the attribute that improves the classification accuracy of the resulting Naive Bayes at most) into the selected attribute subset on each iteration. To our knowledge, the search strategy used in SBC is so greedy that it often falls into a local optimization. Motivated by the success of random forests (Breiman, 2001) and random one-dependence estimators (Jiang, 2011), we propose to randomly select an attribute from o best attributes on each iteration, where $o = log_2 p + 1$ and p is the number of useful candidate attributes.

Traditionally, the goal of learning a classifier is for accurate classification. Thus, the performance of a classifier is measured by its classification accuracy (simply ACC in this paper) in the nature of things. However, recent work (Jiang et al., 2009; Jiang, Li, & Cai, 2009; Provost & Domingos, 2003; Saar-Tsechansky & Provost, 2004) in machine learning domain has shown that learning a classifier with accurate classification is often not enough. Learning a classifier with accurate ranking or probability estimation is also desirable. For example, in direct marketing, we often need to promote the top X% of customers during gradual roll-out, or we often deploy different promotion strategies to customers with different likelihood of buying some products. To accomplish these learning tasks, a ranking of customers in terms of their likelihood of buying is more useful than merely a classification of buyer or non-buyer. Obviously, when our goal is to learn classifiers with accurate ranking or class probability estimation, classification accuracy (ACC) no longer is an appropriate measure for classifiers. Recently, area under the ROC curve (AUC) (Bradley, 1997; Hand & Till, 2001), and conditional log likelihood (CLL) (Grossman & Domingos, 2004; Jiang et al., 2009) have been widely used as measures for ranking and class probability estimation performance of the learned classifiers, respectively.

The classification accuracy is the percentage of test instances correctly classified. The AUC of a classifier on a test dataset with two classes is computed by Eq. (9).

$$AUC = \frac{S_0 - n_0 \times (n_0 + 1)/2}{n_0 \times n_1}, \tag{9}$$

where n_0 and n_1 are the numbers of negative and positive instances, respectively, and $S_0 = \sum r_i$, where r_i is the rank of the i_{th} negative instance in the ranked list. For multiple classes, AUC can be computed using Eq. (10) (Hand & Till, 2001).

$$AUC = \frac{2}{n_c(n_c - 1)} \sum_{i<j} AUC(c_i, c_j), \tag{10}$$

where n_c is the number of classes and the $AUC(c_i,c_j)$ term is the AUC value of each pair of classes c_i and c_j.

Given a classifier G and a set of test instances $T = \{e_1, e_2, \ldots, e_i, \ldots, e_t\}$, where $e_i = \langle a_{i1}, a_{i2}, \ldots, a_{im}, c_i \rangle$, t is the number of test instances, m is the number of attributes, and c_i is the true class label of e_i. Then, the conditional log likelihood CLL of the classifier G on the test instance set T can be defined as:

$$CLL = \sum_{i=1}^{t} log P_G(c_i | a_{i1}, a_{i2} \ldots, a_{im}). \tag{11}$$

Now, the second problem with SBC occurs. The attribute subset evaluation used in SBC always is to maximize the classification accuracy of the resulting Naive Bayes. This leads to a serious mismatch between the learning process and the learning goal. In order

Table 1
Descriptions of UCI datasets used in the experiments.

Dataset	Instances	Numeric attributes	Nominal attributes	Classes	Missing values
anneal	898	6	32	5	N
anneal.ORIG	898	6	32	6	Y
audiology	226	0	69	24	Y
autos	205	15	10	7	Y
balance-scale	625	4	0	3	N
breast-cancer	286	0	9	2	Y
breast-w	699	9	0	2	Y
colic	368	7	15	2	Y
colic.ORIG	368	7	20	2	Y
credit-a	690	6	9	2	Y
credit-g	1000	7	13	2	N
diabetes	768	8	0	2	N
Glass	214	9	0	7	N
heart-c	303	6	7	5	Y
heart-h	294	6	7	5	Y
heart-statlog	270	13	0	2	N
hepatitis	155	6	13	2	Y
hypothyroid	3772	23	6	4	Y
ionosphere	351	34	0	2	N
iris	150	4	0	3	N
kr-vs-kp	3196	0	36	2	N
labor	57	8	8	2	Y
letter	20,000	16	0	26	N
lymph	148	3	15	4	N
mushroom	8124	0	22	2	Y
primary-tumor	339	0	17	21	Y
segment	2310	19	0	7	N
sick	3772	7	22	2	Y
sonar	208	60	0	2	N
soybean	683	0	35	19	Y
splice	3190	0	61	3	N
vehicle	846	18	0	4	N
vote	435	0	16	2	Y
vowel	990	10	3	11	N
waveform-5000	5000	40	0	3	N
zoo	101	1	16	7	N

Table 2
Experimental results for NB versus ENB-ACC, SBC-ACC, and RSNB-ACC: classification accuracy (ACC) and standard deviation.

Dataset	NB	ENB-ACC	SBC-ACC	RSNB-ACC
anneal	94.2328 ± 1.98	97.5733 ± 1.45 ○	96.0811 ± 2.37	96.9716 ± 1.40 ○
anneal.ORIG	88.2842 ± 2.27	88.9544 ± 2.35	89.3552 ± 2.29	89.5336 ± 1.93
audiology	71.0686 ± 4.50	72.0502 ± 5.45	74.0058 ± 6.01	73.1034 ± 4.64
autos	63.9024 ± 5.72	72.2927 ± 4.50 ○	69.7561 ± 7.32	72.2927 ± 6.82 ○
balance-scale	90.6240 ± 1.19	90.6240 ± 1.19	90.6240 ± 1.19	90.3360 ± 1.08
breast-cancer	73.2861 ± 5.36	70.6969 ± 5.90	72.3073 ± 5.86	71.8137 ± 5.02
breast-w	97.3390 ± 1.25	96.6526 ± 1.38	96.7671 ± 1.49	97.0818 ± 1.62
colic	78.7027 ± 3.71	81.0811 ± 3.51	83.3721 ± 3.77	82.4495 ± 3.54
colic.ORIG	74.4080 ± 6.15	73.8104 ± 4.58	72.8849 ± 3.95	74.2488 ± 4.73
credit-a	85.0725 ± 2.53	85.4783 ± 2.81	84.8696 ± 2.83	86.0580 ± 2.68
credit-g	75.7200 ± 3.40	75.4200 ± 2.56	74.6800 ± 2.40	75.2800 ± 2.80
diabetes	75.8094 ± 3.36	76.0947 ± 3.58	76.3550 ± 3.54	76.4604 ± 3.45
glass	57.8760 ± 7.31	57.2115 ± 7.68	56.1838 ± 7.64	56.7486 ± 6.17
heart-c	83.7650 ± 2.98	82.1847 ± 3.22	81.8503 ± 4.40	83.5683 ± 3.46
heart-h	83.2718 ± 4.68	79.7312 ± 5.08 ●	81.1631 ± 4.45	82.4535 ± 4.68
heart-statlog	83.4815 ± 4.24	82.0000 ± 4.12	81.1111 ± 5.27	82.5185 ± 4.28
hepatitis	84.5161 ± 6.03	82.0645 ± 6.18	81.2903 ± 7.15	82.8387 ± 4.63
hypothyroid	92.7996 ± 0.48	93.4146 ± 0.36 ○	93.5313 ± 0.40 ○	93.5737 ± 0.39 ○
ionosphere	90.4290 ± 3.08	90.9980 ± 2.98	90.9956 ± 3.07	91.8543 ± 2.29
iris	94.0000 ± 4.19	95.8667 ± 3.23	96.2667 ± 3.38	95.4667 ± 3.45
kr-vs-kp	87.7096 ± 1.18	94.3492 ± 0.76 ○	94.3366 ± 0.83 ○	94.3115 ± 0.84 ○
labor	95.8788 ± 5.55	89.5152 ± 12.55	80.7879 ± 9.20 ●	90.2424 ± 8.78
letter	69.9630 ± 0.72	70.6810 ± 0.73 ○	70.6380 ± 0.70 ○	70.7460 ± 0.71 ○
lymph	85.3977 ± 6.08	83.6460 ± 6.27	78.4046 ± 6.63	83.9264 ± 6.84
mushroom	95.4210 ± 0.43	99.1630 ± 0.27 ○	99.6775 ± 0.15 ○	99.0251 ± 0.28 ○
primary-tumor	46.4934 ± 2.88	45.9666 ± 3.90	43.2467 ± 4.57	46.1387 ± 3.53
segment	88.6840 ± 1.45	91.3333 ± 1.44 ○	90.7706 ± 1.69 ○	91.2641 ± 1.59 ○
sick	96.7975 ± 0.42	97.4232 ± 0.36 ○	97.4868 ± 0.38 ○	97.3914 ± 0.36 ○
sonar	75.7073 ± 6.23	73.9861 ± 6.25	70.0139 ± 5.36	75.4100 ± 4.56
soybean	91.9755 ± 2.28	91.5668 ± 2.31	91.3624 ± 1.98	92.7963 ± 2.01
splice	95.2727 ± 0.72	95.6552 ± 0.80	95.0658 ± 0.83	96.1066 ± 0.68
vehicle	60.3308 ± 2.34	62.4351 ± 2.80	61.3481 ± 2.87	63.6647 ± 2.74 ○
vote	90.2069 ± 2.51	95.3103 ± 1.99 ○	95.4483 ± 2.19 ○	95.1264 ± 2.33 ○
vowel	63.2323 ± 3.62	65.5758 ± 2.93	65.9394 ± 3.37 ○	66.5859 ± 3.84 ○
waveform-5000	79.9520 ± 0.83	81.0640 ± 1.07	81.1760 ± 0.88 ○	82.4640 ± 0.89 ○
zoo	94.4857 ± 4.75	95.8762 ± 4.39	93.4952 ± 4.64	95.6381 ± 2.62
Average	82.1138	82.7152	82.0180	83.2080

○, ● statistically significant improvement or degradation.

Table 3
Compared results of paired t-tests ($p = 0.05$): classification accuracy (ACC).

	NB	ENB-ACC	SBC-ACC	RSNB-ACC
NB	–	9	9	12
ENB-ACC	1	–	1	1
SBC-ACC	1	0	–	3
RSNB-ACC	0	0	1	–

Table 4
Compared results of ranking tests: classification accuracy (ACC).

Resultset	Wins–losses	Wins	Losses
RSNB-ACC	15	16	1
SBC-ACC	7	11	4
ENB-ACC	6	9	3
NB	−28	30	2

to address this problem, we need a discriminative learning approach to match the learning process and the learning goal (Jiang & Zhang, 2006; Jiang, Zhang, & Cai, 2006).

Based on above observation and analysis, propose an improved Naive Bayes algorithm by carrying a random search through the whole space of attributes. We simply called it Randomly Selected Naive Bayes (RSNB). In order to meet the need of classification, ranking, and class probability estimation, we discriminatively design three different versions: RSNB-ACC, RSNB-AUC, and RSNB-CLL. Please note that RSNB-ACC uses Naive Bayes' classification accuracy to evaluate selected attribute subsets and randomly select an attribute from o best candidate attributes that improve the classification accuracy of the resulting Naive Bayes at most on each iteration. RSNB-AUC and RSNB-CLL are similar to RSNB-ACC, except that the AUC and CLL of the resulting Naive Bayes are used to evaluate the selected attribute subsets respectively. Now, let us give the detailed algorithm descriptions of RSNB-ACC, RSNB-AUC, and RSNB-CLL, respectively.

Algorithm 1: RSNB-ACC (**A**)

```
Input: the original attribute set $\mathbf{A}$
Output: the selected attribute subset $\mathbf{A_s}$
    1. $\mathbf{A_s} = \phi$
    2. Let $p$ be the number of useful candidate attributes and initialize it to 0.
    3. For each attribute $A$ in $\mathbf{A}$
    4.     Considers adding $A$ into $\mathbf{A_s}$ and measures the ACC of NB on $\mathbf{A_s} \bigcup \{A\}$.
    5.     If the ACC of NB on $\mathbf{A_s} \bigcup \{A\}$ is higher than the ACC of NB on $\mathbf{A_s}$, $p = p + 1$.
    6. IF $p == 0$, returns $\mathbf{A_s}$
    7. Otherwise,
    8.     $o = log_2 p + 1$
    9.     Finds $o$ best attributes that improve the ACC of NB on $\mathbf{A_s}$ at most.
    10.    Selects an attribute $A_r$ from $o$ best attributes at random.
    11.    $\mathbf{A_s} = \mathbf{A_s} \bigcup \{A_r\}$ and $\mathbf{A} = \mathbf{A} - \{A_r\}$
    12.    Repeats steps 2–11.
```

Table 5
Experimental results for NB versus ENB-AUC, SBC-AUC, and RSNB-AUC: area under the ROC curve (AUC) and standard deviation.

Dataset	NB	ENB-AUC	SBC-AUC	RSNB-AUC
anneal	97.9225 ± 2.24	97.9063 ± 2.06	97.8211 ± 2.13	98.0085 ± 2.15
anneal.ORIG	95.7558 ± 4.83	95.6226 ± 5.04	95.3593 ± 5.09	95.4279 ± 5.94
audiology	78.4494 ± 1.05	78.4687 ± 0.99	78.2841 ± 1.04	78.6966 ± 1.04
autos	89.2211 ± 4.72	92.9028 ± 2.93 ○	92.0905 ± 3.31	93.2196 ± 3.67 ○
balance-scale	82.5213 ± 2.76	82.5213 ± 2.76	82.5213 ± 2.76	82.4693 ± 2.83
breast-cancer	68.7327 ± 7.72	68.1325 ± 8.51	68.1425 ± 8.62	68.7851 ± 8.07
breast-w	99.2318 ± 0.45	99.1469 ± 0.49	99.1486 ± 0.47	99.2254 ± 0.48
colic	83.4869 ± 4.43	85.9443 ± 4.18	85.7071 ± 3.88	86.7058 ± 3.85 ○
colic.ORIG	80.7588 ± 6.61	82.2462 ± 4.84	82.4771 ± 4.47	82.7184 ± 5.49
credit-a	91.5943 ± 1.87	91.2592 ± 1.56	91.4192 ± 1.78	91.9016 ± 1.85
credit-g	78.9895 ± 3.56	79.0182 ± 3.43	79.1588 ± 3.39	79.1966 ± 3.71
diabetes	82.5091 ± 3.36	84.1543 ± 3.62 ○	84.1539 ± 3.62 ○	83.2959 ± 3.53
glass	80.1387 ± 4.22	81.7662 ± 4.12	81.7299 ± 4.26	82.8040 ± 3.54
heart-c	84.1030 ± 0.34	84.0750 ± 0.34	84.0751 ± 0.37	84.0755 ± 0.35
heart-h	83.8887 ± 0.52	83.8885 ± 0.49	83.8733 ± 0.46	83.9086 ± 0.51
heart-statlog	90.7594 ± 2.74	89.6000 ± 2.33	89.4638 ± 2.22	90.3594 ± 2.77
hepatitis	89.0997 ± 6.46	84.1186 ± 7.59 ●	85.1559 ± 7.33	87.2283 ± 6.68
hypothyroid	85.1321 ± 6.78	83.1813 ± 7.88	83.4893 ± 7.39	84.3916 ± 7.08
ionosphere	93.3926 ± 3.20	95.1038 ± 2.77	94.8899 ± 2.89	95.6066 ± 2.69 ○
iris	98.8222 ± 1.15	98.7407 ± 1.06	98.7407 ± 1.06	99.0222 ± 1.05
kr-vs-kp	95.1710 ± 0.91	98.4553 ± 0.40 ○	98.6382 ± 0.38 ○	98.6328 ± 0.38 ○
labor	97.8095 ± 3.88	92.6071 ± 10.61	89.5833 ± 11.64	96.1190 ± 5.86
letter	96.8566 ± 0.15	97.0817 ± 0.14 ○	97.0860 ± 0.14 ○	97.1045 ± 0.14 ○
lymph	91.8005 ± 3.61	91.3757 ± 3.25	91.0153 ± 3.09	91.5522 ± 3.36
mushroom	99.7854 ± 0.03	99.9777 ± 0.01 ○	99.9814 ± 0.01 ○	99.9811 ± 0.01 ○
primary-tumor	81.4792 ± 2.17	81.3652 ± 2.17	80.8818 ± 2.32	81.4007 ± 2.24
segment	98.4338 ± 0.31	99.1672 ± 0.23 ○	99.1157 ± 0.27 ○	99.1528 ± 0.24 ○
sick	95.8348 ± 1.45	96.5503 ± 1.12	96.5808 ± 1.07	96.2933 ± 1.41 ○
sonar	83.8368 ± 5.54	83.0079 ± 4.77	80.5418 ± 6.60	83.6486 ± 5.74
soybean	99.8000 ± 0.15	99.8014 ± 0.18	99.7894 ± 0.17	99.8244 ± 0.14
splice	99.4432 ± 0.18	99.5090 ± 0.16	99.4913 ± 0.17	99.5401 ± 0.17 ○
vehicle	80.4163 ± 2.25	83.2314 ± 2.24 ○	83.4763 ± 2.12 ○	83.3140 ± 1.96 ○
vote	97.1752 ± 1.27	98.4290 ± 1.10 ○	98.2457 ± 1.18 ○	98.6235 ± 0.96 ○
vowel	95.0860 ± 0.98	96.0668 ± 0.80 ○	96.0668 ± 0.80 ○	96.2095 ± 0.79 ○
waveform-5000	95.2588 ± 0.39	95.5537 ± 0.38 ○	95.5850 ± 0.37 ○	95.6379 ± 0.41 ○
zoo	97.4835 ± 3.15	95.8254 ± 4.08	95.3268 ± 3.77	96.8644 ± 3.43
Average	90.0050	90.1612	89.9752	90.5818

○, ● statistically significant improvement or degradation.

Table 6
Compared results of paired t-tests ($p = 0.05$): area under the ROC curve (AUC).

	NB	ENB-AUC	SBC-AUC	RSNB-AUC
NB	–	10	9	13
ENB-AUC	1	–	0	1
SBC-AUC	0	0	–	1
RSNB-AUC	0	0	0	–

Table 7
Compared results of ranking tests: area under the ROC curve (AUC).

Resultset	Wins–losses	Wins	Losses
RSNB-AUC	15	15	0
SBC-AUC	8	9	1
ENB-AUC	8	10	2
NB	−31	1	32

Algorithm 2: RSNB-AUC (**A**)

```
Input: the original attribute set $\mathbf{A}$
Output: the selected attribute subset $\mathbf{A_s}$
    1. $\mathbf{A_s} = \phi$
    2. Let $p$ be the number of useful candidate attributes and initialize it to 0.
    3. For each attribute $A$ in $\mathbf{A}$
    4.     Considers adding $A$ into $\mathbf{A_s}$ and measures the AUC of NB on $\mathbf{A_s} \bigcup \{A\}$.
    5.     If the AUC of NB on $\mathbf{A_s} \bigcup \{A\}$ is higher than the AUC of NB on $\mathbf{A_s}$, $p = p + 1$.
    6. IF $p == 0$, returns $\mathbf{A_s}$
    7. Otherwise,
    8.     $o = log_2 p + 1$
    9.     Finds $o$ best attributes that improve the AUC of NB on $\mathbf{A_s}$ at most.
    10.    Selects an attribute $A_r$ from $o$ best attributes at random.
    11.    $\mathbf{A_s} = \mathbf{A_s} \bigcup \{A_r\}$ and $\mathbf{A} = \mathbf{A} - \{A_r\}$
    12.    Repeats steps 2–11.
```

Algorithm 3: RSNB-CLL (**A**)

```
Input: the original attribute set $\mathbf{A}$
Output: the selected attribute subset $\mathbf{A_s}$
    1. $\mathbf{A_s} = \phi$
    2. Let $p$ be the number of useful candidate attributes and initialize it to 0.
    3. For each attribute $A$ in $\mathbf{A}$
    4.     Considers adding $A$ into $\mathbf{A_s}$ and measures the CLL of NB on $\mathbf{A_s} \bigcup \{A\}$.
    5.     If the CLL of NB on $\mathbf{A_s} \bigcup \{A\}$ is higher than the CLL of NB on $\mathbf{A_s}$, $p = p + 1$.
    6. IF $p == 0$, returns $\mathbf{A_s}$
    7. Otherwise,
    8.     $o = log_2 p + 1$
    9.     Finds $o$ best attributes that improve the CLL of NB on $\mathbf{A_s}$ at most.
    10.    Selects an attribute $A_r$ from $o$ best attributes at random.
    11.    $\mathbf{A_s} = \mathbf{A_s} \bigcup \{A_r\}$ and $\mathbf{A} = \mathbf{A} - \{A_r\}$
    12.    Repeats steps 2–11.
```

Since our RSNB is inherently unstable, we stabilize the estimated class membership probabilities by building an ensemble of RSNB using bagging (Breiman, 1996) and then average the estimated class membership probabilities across the ensemble. Bagging has two parameters: the number of bagging iterations and the percentage of the training data to use for learning a RSNB in each iteration. In our experiments, we use the default parameter settings with 10 and 100, respectively. To our knowledge, bagging is widely used for decision tree learning. For example, Breiman (2001) uses bagging to scale up the classification accuracy of the learned random trees. Hall (2007) stabilizes the estimated attribute weights by building multiple decision trees using bagging.

4. Experiments and results

This section evaluates the performance of Randomly Selected Naive Bayes (RSNB) on 36 UCI datasets (Merz, Murphy, & Aha, 1997) published on the main web site of Weka platform (Witten & Frank, 2005). The properties of these datasets are shown in Table 1. In our experiment, Missing values are replaced with the modes and means from the available data. Numeric attribute values are discretized using the unsupervised ten-bin discretization implemented in Weka platform. Besides, we manually delete three useless attributes: the attribute "Hospital Number" in the data set "colic.ORIG", the attribute "instance name" in the data set "splice", and the attribute "animal" in the data set "zoo".

We ran three groups of experiments. The first group of experiment compares Naive Bayes (NB) with Evolutional Naive Bayes with ACC-based attribute subset evaluation (ENB-ACC), selective Bayesian classifiers with ACC-based attribute subset evaluation (SBC-ACC) and Randomly Selected Naive Bayes with ACC-based attribute subset evaluation (RSNB-ACC) in terms of classification accuracy (ACC).

The second group of experiment compares Naive Bayes (NB) with Evolutional Naive Bayes with AUC-based attribute subset evaluation (ENB-AUC), selective Bayesian classifiers with AUC-based attribute subset evaluation (SBC-AUC) and Randomly Selected Naive Bayes with AUC-based attribute subset evaluation (RSNB-AUC) in terms of area under the ROC curve (AUC).

The third group of experiment compares Naive Bayes (NB) with Evolutional Naive Bayes with CLL-based attribute subset evaluation (ENB-CLL), selective Bayesian classifiers with CLL-based attribute subset evaluation (SBC-CLL) and Randomly Selected Naive Bayes with CLL-based attribute subset evaluation (RSNB-CLL) in terms of conditional log likelihood (CLL).

In all experiments, the classification accuracy (ACC), area under the ROC curve (AUC), and conditional log likelihood (CLL) of each algorithm on each data set are averaged over five fivefold cross-validation runs. 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. The detailed experimental results are presented in Tables 2, 5, and 8, respectively. The symbols ○ and ● in the tables respectively denote statistically significant improvement or degradation over NB with the $p = 0.05$ significance level (Nadeau & Bengio, 2003). Besides, the averages are summarized at the bottom of the tables.

Finally, we conducted a corrected paired two-tailed t-test with the $p = 0.05$ significance level (Nadeau & Bengio, 2003) to compare

Table 8
Experimental results for NB versus ENB-CLL, SBC-CLL, and RSNB-CLL: conditional log likelihood (CLL) and standard deviation.

Dataset	NB	ENB-CLL	SBC-CLL	RSNB-CLL
anneal	−28.9761 ± 9.44	−19.0672 ± 8.24 ○	−18.6637 ± 8.07 ○	−19.4587 ± 7.95 ○
anneal.ORIG	−48.5308 ± 7.77	−44.0962 ± 7.01 ○	−43.9403 ± 7.10 ○	−44.2894 ± 6.92 ○
audiology	−131.0559 ± 29.22	−58.3976 ± 14.92 ○	−50.3858 ± 12.55 ○	−46.7342 ± 12.93 ○
autos	−88.4930 ± 20.78	−35.3663 ± 8.28 ○	−34.6289 ± 7.42 ○	−33.2148 ± 7.26 ○
balance-scale	−64.5042 ± 1.73	−64.5042 ± 1.73	−64.5042 ± 1.73	−64.8452 ± 1.68
breast-cancer	−36.8537 ± 6.46	−34.6096 ± 4.90	−34.6129 ± 4.97	−33.8611 ± 4.40
breast-w	−36.0164 ± 18.19	−21.4498 ± 9.81 ○	−18.6232 ± 9.99 ○	−16.7786 ± 8.07 ○
colic	−61.2191 ± 16.38	−36.3896 ± 7.14 ○	−36.7446 ± 6.95 ○	−32.3628 ± 5.70 ○
colic.ORIG	−43.6967 ± 9.87	−38.2033 ± 5.24	−38.1869 ± 5.41	−36.9792 ± 5.17 ○
credit-a	−57.6057 ± 10.24	−51.5548 ± 6.76 ○	−51.1526 ± 7.02 ○	−50.5846 ± 6.99 ○
credit-g	−105.5970 ± 11.80	−102.5801 ± 10.69	−102.6776 ± 10.72	−101.6642 ± 10.79 ○
diabetes	−81.4770 ± 10.50	−72.9347 ± 8.37 ○	−72.7327 ± 8.24 ○	−74.4471 ± 8.36 ○
glass	−48.4384 ± 7.43	−48.5222 ± 6.99	−48.5222 ± 6.99	−46.6195 ± 6.33
heart-c	−27.6918 ± 7.81	−26.7952 ± 6.00	−26.8008 ± 5.98	−24.6545 ± 5.71
heart-h	−27.8219 ± 9.75	−24.4869 ± 5.12	−25.8685 ± 6.73	−23.9873 ± 6.58
heart-statlog	−25.0366 ± 5.97	−23.3948 ± 4.34	−23.4374 ± 4.58	−22.5013 ± 4.27
hepatitis	−17.5025 ± 7.97	−14.7229 ± 5.09	−15.0625 ± 5.17	−12.1235 ± 4.14 ○
hypothyroid	−194.6662 ± 16.42	−174.0849 ± 13.27 ○	−175.8764 ± 13.07 ○	−169.8677 ± 11.13 ○
ionosphere	−69.8642 ± 24.27	−23.7417 ± 7.79 ○	−23.7138 ± 9.67 ○	−19.4491 ± 7.50 ○
iris	−5.1369 ± 2.80	−4.0431 ± 2.29	−4.0431 ± 2.29	−4.6706 ± 2.44
kr-vs-kp	−187.0934 ± 11.33	−167.2682 ± 7.18 ○	−166.0125 ± 6.53 ○	−166.2501 ± 6.41 ○
labor	−1.6057 ± 1.43	−2.1420 ± 1.95	−1.8879 ± 1.52	−1.8555 ± 1.42
letter	−5033.0208 ± 130.0	−4606.3879 ± 92.33 ○	−4647.7769 ± 109.8 ○	−4410.6723 ± 98.06 ○
lymph	−12.7275 ± 6.26	−12.1085 ± 6.02	−12.0353 ± 6.32	−11.6785 ± 5.03
mushroom	−219.3951 ± 27.60	−33.8142 ± 4.80 ○	−38.2175 ± 5.09 ○	−34.8794 ± 5.90 ○
primary-tumor	−131.6306 ± 10.30	−130.3551 ± 10.20 ○	−130.3551 ± 10.20 ○	−130.2588 ± 10.78
segment	−249.8758 ± 43.35	−110.0059 ± 17.23 ○	−113.5916 ± 17.80 ○	−109.2187 ± 16.88 ○
sick	−91.7533 ± 9.85	−64.7131 ± 8.50 ○	−64.4688 ± 8.51 ○	−69.6941 ± 8.41 ○
sonar	−46.3788 ± 15.44	−30.9046 ± 11.39 ○	−30.2045 ± 10.63 ○	−22.9869 ± 5.60 ○
soybean	−53.4057 ± 14.73	−26.8831 ± 6.16 ○	−29.4892 ± 6.73 ○	−28.4727 ± 5.56 ○
splice	−93.7026 ± 16.16	−85.3562 ± 13.81	−85.4009 ± 12.25	−78.1050 ± 12.32 ○
vehicle	−346.5676 ± 42.36	−149.1331 ± 14.93 ○	−147.9305 ± 13.60 ○	−146.3802 ± 11.07 ○
vote	−54.5886 ± 19.51	−14.1271 ± 5.66 ○	−13.7710 ± 5.92 ○	−12.5230 ± 5.20 ○
vowel	−195.4111 ± 18.99	−176.7776 ± 15.96 ○	−176.7776 ± 15.96 ○	−174.3893 ± 16.41 ○
waveform-5000	−756.9166 ± 39.30	−475.1928 ± 21.85 ○	−481.4488 ± 24.15 ○	−405.1173 ± 19.60 ○
zoo	−2.4009 ± 1.28	−2.3656 ± 1.46	−2.3215 ± 1.47	−2.5898 ± 1.14
Average	−241.0183	−194.6244	−195.8852	−185.6712

○, ● statistically significant improvement or degradation.

Table 9
Compared results of paired t-tests ($p = 0.05$): conditional log likelihood (CLL).

	NB	ENB-CLL	SBC-CLL	RSNB-CLL
NB	–	22	22	25
ENB-CLL	0	–	1	8
SBC-CLL	0	0	–	7
RSNB-CLL	0	1	1	–

Table 10
Compared results of ranking tests: conditional log likelihood (CLL).

Resultset	Wins–losses	Wins	Losses
RSNB-CLL	38	40	2
SBC-CLL	17	24	7
ENB-CLL	14	23	9
NB	−69	0	69

each pair of algorithms. The detailed compared results are presented in Tables 3, 4, 6, 7, 9 and 10, respectively. In Tables 3, 6, and 9, Each number in the tables indicates how many datasets the algorithm in the corresponding column achieves significant wins with regard to the algorithm in the corresponding row. In Tables 4, 7, and 10, 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 RSNB is overall better than ENB and SBC. Now, we summarize the highlights as follows:

1. In terms of classification accuracy (ACC), seen from the first row in Table 4, RSNB-ACC achieves significant wins on 16 datasets with regard to all the other competitors and only losses on 1 datasets.
2. In terms of area under the ROC curve (AUC), seen from the first row in Table 7, RSNB-AUC achieves significant wins on 15 datasets with regard to all the other competitors and surprisingly losses on 0 datasets.
3. In terms of conditional log likelihood (CLL), seen from the first row in Table 10, RSNB-CLL achieves significant wins on 40 datasets with regard to all the other competitors and only losses on 2 datasets.

5. Conclusions

The well-known attribute selection algorithm selective Bayesian classifiers (SBC) always adds a best attribute (the attribute that improves the classification accuracy of the resulting Naive Bayes at most) into the selected attribute subset on each iteration. To our knowledge, the search strategy used in SBC is so greedy that it often falls into a local optimization. In order to address this problem, we propose an improved Naive Bayes algorithm by carrying a random search through the whole space of attributes. We simply called it Randomly Selected Naive Bayes (RSNB). In order to meet the need of classification, ranking, and class probability estimation, we discriminatively design three different versions: RSNB-ACC, RSNB-AUC, and RSNB-CLL. The experimental results based on a large number of UCI datasets validate their effectiveness in terms of classification accuracy (ACC), area under the ROC curve (AUC), and conditional log likelihood (CLL), respectively.

Acknowledgements

We thank 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. 61075063), the Provincial Natural Science Foundation of Hubei (No. 2011CDA103), the Provincial Humanities and Social Science Foundation of Hubei (No. 2011B005), and the Fundamental Research Funds for the Central Universities (No. CUG110405).

References

Bouchaala, L., Masmoudi, A., Gargouri, F., & Rebai, A. (2010). Improving algorithms for structure learning in Bayesian Networks using a new implicit score. *Expert Systems with Applications, 37*(7), 5470–5475.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Grossman, D., & Domingos, P. (2004). Learning Bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the twenty-first international conference on machine learning* (pp. 361–368). Banff, Canada: ACM Press.

Hall, M. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the 17th international conference on machine learning* (pp. 359–366).

Hall, M. (2007). A decision tree-based attribute weighting filter for naive bayes. *Knowledge-Based Systems, 20*, 120–126.

Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning, 45*(2), 171–186.

Jiang, L. (2011). Random one-dependence estimators. *Pattern Recognition Letters, 32*(3), 532–539.

Jiang, L., Cai, Z., & Wang, D. (2010). Improving Naive Bayes for classification. *International Journal of Computers and Applications, 32*(3), 328–332.

Jiang, L., Li, C., & Cai, Z. (2009). Learning decision tree for ranking. *Knowledge and Information Systems, 20*(1), 123–135.

Jiang, L., Li, C., & Cai, Z. (2009). Decision tree with better class probability estimation. *International Journal of Pattern Recognition and Artificial Intelligence, 23*(4), 745–763.

Jiang, L., & Zhang, H. (2006). Learning naive Bayes for probability estimation by feature selection. In *Proceedings of the 19th Canadian Conference on Artificial Intelligence, CAI 2006. LNAI* (Vol. 4013, pp. 503–514). Springer Press.

Jiang, L., Zhang, H., & Cai, Z. (2006). Discriminatively improving naive Bayes by evolutionary feature selection. *Romanian Journal of Information Science and Technology, 9*(3), 163–174.

Jiang, L., Zhang, H., Cai, Z., & Su, J. (2005). Evolutional naive Bayes. In *Proceedings of the 1st international symposium on intelligent computation and its applications, ISICA 2005* (pp. 344–350). China University of Geosciences Press.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence Journal, 97*(1–2), 273–324 (special issue on relevance).

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the tenth conference on uncertainty in artificial intelligence* (pp. 339–406).

Lei, Y., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, 5*, 1205C1224.

Liu, W., Yue, K., & Li, W. (2011). Constructing the Bayesian network structure from dependencies implied in multiple relational schemas. *Expert Systems with Applications, 38*(6), 7123–7134.

Merz, C., Murphy, P., & Aha, D. (1997). UCI repository of machine learning databases. Dept of ICS, University of California, Irvine. http://www.ics.uci.edu/mlearn/MLRepository.html.

Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning, 52*(3), 239–281.

Park, H. S., & Cho, S. B. (2012). Evolutionary attribute ordering in Bayesian networks for predicting the metabolic syndrome. *Expert Systems with Applications, 39*(4), 4240–4249.

Provost, F. J., & Domingos, P. (2003). Tree induction for probability-based ranking. *Machine Learning, 52*(3), 199–215.

Ratanamahatana, C. A., & Gunopulos, D. (2003). Feature selection for the naive Bayesian classifier using decision trees. *Applied Artificial Intelligence, 17*, 475–487.

Saar-Tsechansky, M., & Provost, F. (2004). Active sampling for class probability estimation and ranking. *Machine Learning, 54*(2), 153–178.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining* (first ed.). Pearson Education.

Witten, I. H., & Frank, E. (2005). Data mining: Practical machine learning tools and techniques (2nd ed.) San Francisco: Morgan Kaufmann. <http://prdownloads.sourceforge.net/weka/datasets-UCI.jar>.