Improving Simple Bayes

Ron Kohavi Barry Becker Dan Sommerfield

Data Mining and Visualization Group
Silicon Graphics, Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94043
{becker,ronnyk,sommda}@engr.sgi.com

Abstract. The simple Bayesian classifier (SBC), sometimes called Naive-Bayes, is built based on a conditional independence model of each attribute given the class. The model was previously shown to be surprisingly robust to obvious violations of this independence assumption, yielding accurate classification models even when there are clear conditional dependencies. We examine different approaches for handling unknowns and zero counts when estimating probabilities. Large scale experiments on 37 datasets were conducted to determine the effects of these approaches and several interesting insights are given, including a new variant of the Laplace estimator that outperforms other methods for dealing with zero counts. Using the bias-variance decomposition [15, 10], we show that while the SBC has performed well on common benchmark datasets, its accuracy will not scale up as the dataset sizes grow. Even with these limitations in mind, the SBC can serve as an excellent tool for initial exploratory data analysis, especially when coupled with a visualizer that makes its structure comprehensible.

1 Introduction to the Simple-Bayesian Classifier

In supervised classification learning, a labelled training set is presented to the learning algorithm. The learner uses the training set to build a model that maps unlabelled instances to class labels. The model serves two purposes: it can be used to predict the labels of unlabelled instances, and it can provide valuable insight for people trying to understand the domain. Simple models are especially useful if the model is to be understood by non-experts in machine learning.

The simple Bayes classifier (SBC), sometimes called Naive-Bayes, is built based on a conditional independence model of each attribute given the class [11, 7]. Formally, the probability of a class label value C_i for an unlabelled instance $X = \langle A_1, \ldots, A_n \rangle$ consisting of n attribute values is given by

$$\begin{aligned}
&\mathrm{P}(C_i \mid X) \\
&= \mathrm{P}(X \mid C_i) \cdot \mathrm{P}(C_i)/\mathrm{P}(X) && \text{by Bayes rule} \\
&\propto \mathrm{P}(A_1, \ldots, A_n \mid C_i) \cdot \mathrm{P}(C_i) && P(X) \text{ is same for all label values.} \\
&= \prod_{j=1}^{n} \mathrm{P}(A_j \mid C_i) \cdot \mathrm{P}(C_i) && \text{by conditional independence assumption.}
\end{aligned}$$

The above probability is computed for each class and the prediction is made for the class with the largest posterior probability. This model is very robust and continues to perform well even in the face of obvious violations of this independence assumption.

The probabilities in the above formulas must be estimated from the training set. We address two separate issues related to the SBC: how to treat unknown values and how to estimate the probabilities (especially when some of the counts are zero). A large scale comparison of these variants on 37 datasets from the UCI Repository [20] was done. We emphasize the extreme cases that led to interesting insights.

Using the bias-variance decomposition, we show that while the SBC has performed well on common benchmark datasets, its accuracy will not scale up as the dataset sizes grow.

2 Improving the "Naive" Simple Bayesian Classifier

We investigate the various options that one could choose when using the SBC. For each of these options we conducted experiments to show the differences in error. We also can explain in some cases why these differences arise and when one option is preferable. Before describing the different options, we describe the methodology used throughout the paper.

2.1 Experimental Methodology

We chose all the datasets reported in Domingos and Pazzani [5], except lung-cancer, labor-negotiations, and soybean (small), which had fewer than 100 instances. We added more datasets, especially larger ones, such as segment, mushroom, letter, and adult for a total of 37. The specific datasets are shown below in Table 2.

Our main concern with estimating accuracy is that the estimate should be precise. Therefore, we ran different inducers on these datasets in two forms. If the dataset was large or artificial, indicating that a single test set would yield accurate estimates, we used a training-set/test-set as defined in the source for the dataset (*e.g.*, Statlog defined the splits for DNA, letter, satimage; CART defined the training size for waveform and led24) or a 2/3, 1/3 split, and ran the inducer once; otherwise, we performed 10-fold cross-validation to improve the reliability of the estimate.

The extreme, and therefore interesting, results are shown graphically. We show both the absolute difference in error as bars, and the relative error rates (*i.e.*, one error rate divided by another) as symbols (*e.g.*pluses). Relative error rates are especially useful when the error itself is low. For example, reducing the error rate by 0.5% may not seem significant in terms of absolute errors, but if the initial error rate was only 1%, the error would be halved! If each error costs a significant amount of money, then the error ratio is most important. Note that both types of information are shown on the same graph, with the left y-axis

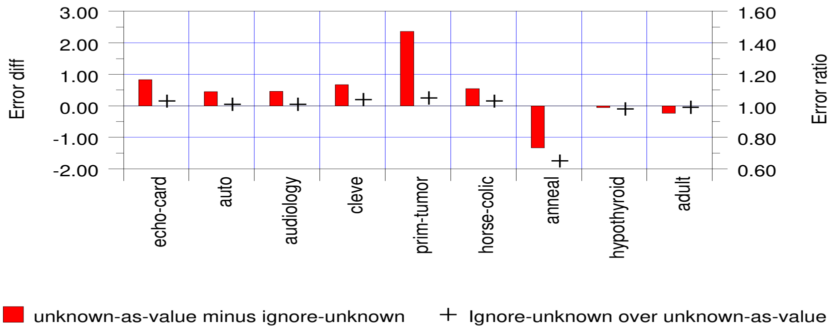

Fig. 1. Comparison of ignoring unknown values and considering them a separate value. The left axis shows the scale for the bars (absolute error differences); bars above zero indicate that ignoring unknowns is better. The right axis shows the scale for the pluses (relative error ratio); pluses above the one line show better performance for ignoring unknowns.

showing the scale for the bars and the right y-axis showing the scale for the relative errors.

2.2 The Basic Classifier

We begin with a very simple SBC model. Continuous attributes are discretized into 10 bins of uniform size and frequency counts are used to estimate the probabilities. If there is a class label value with zero counts, that class is ignored and will never be predicted. If there is a zero count for a class label C and an attribute value A, the conditional probability, $P(A \mid C)$, will be zero.

Ties are broken in favor of the class with more instances in the original dataset. This is important especially for this simple version because all classes can end up with zero posterior probability, in which case we predict the majority class.

2.3 How Should Unknowns be Treated?

The first option we investigate is how to handle unknowns. One can either consider unknowns to be a separate value, as was done by Domingos and Pazzani [5], or they can be ignored for a given instance by not including the matching term in the overall product.

The optimal treatment of unknowns depend on their meaning in the domain. If the unknown has a special meaning (*e.g.*, an unknown (blank) for the army rank of a person), it is likely that treating it as a separate value will be better. If, however, the unknowns represent truly missing information because the data was corrupted or the entry was mistakenly left blank, the latter approach should be better, as it matches the Bayesian definition of marginalizing the appropriate attribute. Figure 1 shows the experimental results for the datasets that differed.

Over all the datasets, the average error rate for considering unknowns to be a separate value was 20.30% and for ignoring them it was 20.20%. In most datasets (not shown) the unknown treatment was not important. Those that differed were generally better for ignoring unknowns, except for the anneal dataset, where a significant increase in error was observed. The encoding of the anneal dataset at the UCI Repository appears to be flawed[1]. For this dataset, we converted the unknowns to dashes and called the file anneal-U, which we will use in the rest of this paper.

One reason to ignore unknowns in the algorithm is that users can always map their unknowns to a separate values, while if unknowns are considered a separate value, users cannot cause certain values to be ignored. We conclude that it is better for algorithms to ignore unknowns, and in cases where unknowns represent a special value, such as anneal, the unknowns should be converted to a separate value. In the rest of the experiments, unknowns will be treated as true missing values.

2.4 Estimating Probabilities

The class probabilities and the conditional probabilities in the above experiments were based on pure frequency counts. An attribute value that does not occur together with a given class label value will produce a zero estimate for $P(A \mid C)$, eliminating class C from consideration. To overcome this problem of a single value controlling the outcome, we examine two general approaches from the literature:

The no-match approaches Replace a zero count (no-match) for $P(A \text{ and } C)$ with a factor that is inversely proportional to the number of instances, m. The different approaches use a different numerator, but the idea is the same. Clark and Niblett [4] and Domingos and Pazzani [5] used $P(C)/m$. In $\mathcal{MLC}$++ [14], the default was $0.5/m$.

Laplace approaches Given a predefined factor f, if there are N matches out of n instances for a k value problem, estimate the probability as $(N+f)/(n+kf)$. For a two valued problems with $f = 1$, we get the well-known Laplace's law of succession [11] $(N+1)/(N+2)$.

Table 1 summarizes the average errors and the average error ratios relative to *No-matches-PC* (the No-match approach with the numerator factor set to $P(C)$) for all the datasets. We can see that frequency counts is the worst performer, and Laplace's law of succession as second worst. *No-matches-PC* is somewhere in the middle. Very small settings for no-matches, such as $0.01/m$ and similar

[1] The description file says that "The '-' values are actually 'not-applicable' values rather than 'missing-values' (and so can be treated as legal discrete values rather than as showing the absence of a discrete value)" yet there are no dashes in the file. In addition, we tested C4.5 on the original and new encoding of the anneal dataset. Under the original encoding, the 10-fold cross-validation error was 8.23% and under the encoding with dashes, it decreased to 1.22%.

Approach	Average error ratio relative to No-matches-PC	Average error
Laplace-m	0.96	18.58
No-matches-0.01	0.97	18.51
Laplace-0.01	0.98	18.70
No-matches-PC	1.00	18.62
No-matches-0.1	1.00	18.64
No-matches-0.5	1.02	18.76
Laplace-0.1	1.02	18.83
Laplace-1 (law of succession)	1.11	19.59
No-matches-0 (frequency counts)	1.17	20.16

Table 1. Comparison of different methods for estimating probabilities. *No-match-f* denotes replacing zeroes with the given factor f over the number of instances. *Laplace-f* denotes adding f to the numerator and f times the number of possible values to the denominator. *Laplace-m denotes adding a factor $1/m$ for m instances.*

corrections for Laplace seem to perform best. Laplace-m sets the adjustment to be $1/m$, making it smaller as the file size grows.

Figure 2 shows the errors and error ratios for three of the variants and for datasets that had significant differences. We can see that frequency counts (No-matches-0) performs generally worse than No-matches-PC, except on the cars and mushroom datasets where it performs significantly better. Laplace$-m$ seems to take the best of both worlds. It tracks No-matches-PC on most datasets, except cars and mushroom where it tracks No-matches-0 well. The error differences can be explained by two distinct and opposite effects. We begin with an explanation of why frequency counts performs poorly sometimes.

When the conditional probability is set to zero based on frequency counts, it is possible to rule out a class because of a single attribute value; moreover, sometimes all classes are ruled out! An opposite effect happens when the probabilities are biased too far away from zero as with Laplace's law of succession. In those cases, a single strong predictor can be weakened too much. Correcting zero counts hurt performance on the cars and mushroom datasets because these datasets rely on a single strong predictor being able to override many weaker predictors for other classes.

Both methods for correcting frequency counts seem to work best when very small correction values are used, which to our knowledge has not been previously reported.

If, in addition to unknown handling and zero counts, we also discretize the data using entropy minimization [6], the average absolute error for all datasets decreases from 18.58% to 18.13% with an average relative error ratio of 0.94.

2.5 Limitations of the SBC

While the SBC shows good performance on many of the datasets from UCI, it is still a very limited classifier. It is a "global" classifier and cannot make local predictions as nearest-neighbors or decision trees can. Therefore, the simple

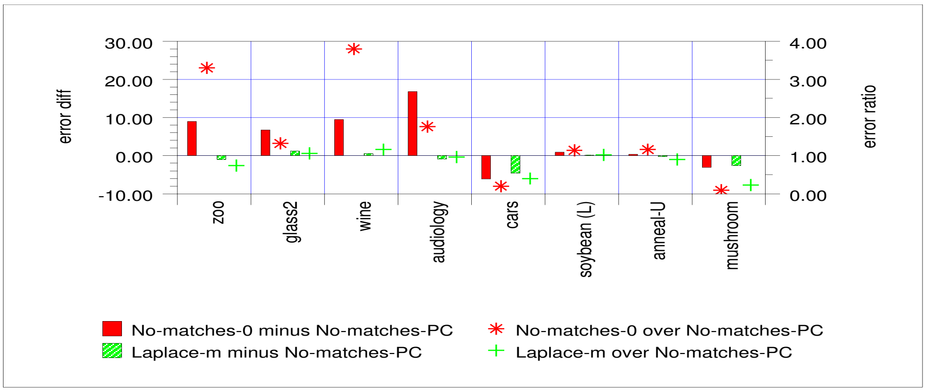

Fig. 2. Comparison of three probability estimation methods. The baseline chosen was No-matches-PC. Absolute errors and relative error ratios are shown with respect to this baseline. The left axis shows the scale for the bars (absolute error differences); bars above zero show worse performance than No-matches-PC. The right axis shows the scale for the pluses and asterisks (relative error ratios); symbols above one show worse performance.

Bayesian inducer cannot be *consistent* in the statistical sense without additional strong assumptions (an inducer is consistent if the classifiers it produces approach the Bayes optimal error as the dataset size grows to infinity). Proofs have been given for decision tree inducers [12] and for nearest-neighbor inducers [8] under mild assumptions.

In the bias-variance decomposition of error [15, 10], the error is the sum of two terms: the squared bias and the variance. The bias measures how well the induced classifiers fit the data (low values are good), and the variance measures the stability (low values indicate stability).

The SBC usually has low variance as perturbations of the training sets will rarely cause large changes in its predictions, which are based on probabilities. Contrast this with decision tree inducers that are unstable [2, 1] because if two attributes are ranked very closely at the root of a subtree, their order might change when the training set is perturbed, and cause the whole subtree to differ. However, the SBC usually has high bias because of its inability to locally fit the data.

Figure 3 shows the bias-variance decomposition as described by Kohavi and Wolpert for the large datasets and two inducers: simple-Bayesian and MC4 (a decision tree inducer in $\mathcal{MLC}$++).[2] The evaluation set sampling (used to compute the bias and variance) was 30%. The internal sample process to generate training sets was half of the remaining 70% (so training sets were 35% of the original dataset); ten such samples were generated. For datasets with fewer than 3000 instances, the whole process was repeated ten times and averaged (for a total of 100 runs).

The figure shows that the performance of SBC is generally inferior for all large

[2] The bias-variance decomposition algorithm in $\mathcal{MLC}$++ requires support routines that are unavailable in C4.5, which is why we used MC4 here.

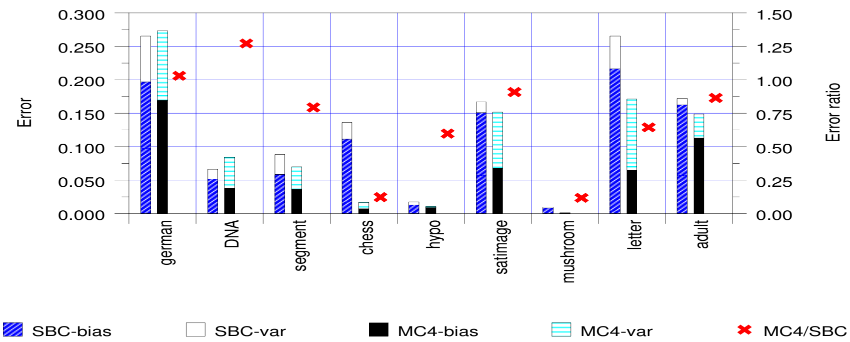

Fig. 3. Bias-variance decomposition for the larger datasets for SBC and MC4, the $\mathcal{MLC}$++ decision tree inducer, which is similar to C4.5. The lower bar denotes bias, the upper bar denotes variance, and the sum indicates the total error. The left axis shows the error for the bars (lower is better). The right axis shows the ratio of MC4 to SBC (lower than one indicates MC4 is better).

datasets (except for DNA which is much better). Looking at the decomposed terms, the variance of the the simple Bayesian inducer is always lower (except for chess and hypothyroid). Since more data cannot change the bias of the simple Bayesian model, we can conclude that error will not decrease much as the dataset size grows.

3 Comparison with Other Classifiers

In the previous sections we proposed solutions for some of the decisions required for the SBC. We reduced the overall error from 20.30% for the basic SBC to 18.13%, which is a relative improvement of 10.7%. Table 2 shows the dataset characteristics and absolute errors for C4.5, C4.5-rules [21], and our SBC.

The average error for C4.5 is 17.85%, for C4.5-rules it is 17.90%, and for SBC it is 18.19%. If we ignore the big datasets (datasets DNA through adult in the table), C4.5's error is 20.83%, C4.5-rules's error is 20.93%, and SBC's error is 20.10%. The simple-Bayesian inducer and C4.5 are very fast inducers, never taking more than a few minutes. C4.5-rules took over 4.5 hours to build a ruleset for the adult dataset.

The SBC is a good fast algorithm. Its accuracy is very good on small datasets but it may asymptote to a high error rate, making it less useful as a classifier for very large databases.

4 Related Work

The SBC model is very simple and its explanatory power was previously noted by Kononenko [17], who wrote that "Physicians found such explanations [using conditional probabilities] as natural and similar to their classification. They also summed up evidence for/against a diagnosis."

Dataset	Train/ test size	Data set size	No of attr cont/ nom	C4.5 error	C4.5-rules error	SBC error
zoo	91/10-CV	101	0/16	7.05± 0.71	7.55±0.74	2.91± 1.48
echocardiogram	118/10-CV	131	6/1	37.62± 1.29	37.93±1.21	38.85± 3.16
lymphography	133/10-CV	148	3/15	23.42± 1.05	22.71±0.99	16.10± 2.98
iris	135/10-CV	150	4/0	5.20± 0.49	4.53±0.50	7.33± 1.85
hepatitis	140/10-CV	155	6/13	20.75± 1.08	21.14±1.07	15.46± 2.84
glass2	147/10-CV	163	9/0	20.82± 0.96	19.42±0.95	19.67± 2.00
wine	160/10-CV	178	13/0	7.02± 0.61	6.41±0.58	1.14± 0.76
auto	184/10-CV	205	15/10	18.96± 1.03	22.95±1.00	25.31± 3.69
sonar	187/10-CV	208	60/0	27.42± 0.92	27.28±1.00	25.48± 2.46
glass	193/10-CV	214	9/0	33.17± 0.94	34.06±0.96	29.89± 2.29
led24	200/3000	3200	0/24	34.33± 0.87	35.43±0.87	35.90± 0.88
audiology	203/10-CV	226	0/69	22.35± 0.84	23.68±0.86	21.28± 2.23
breast (L)	257/10-CV	286	0/9	26.15± 0.73	29.29±0.77	26.59± 2.24
cleve	273/10-CV	303	6/7	24.02± 0.76	20.27±0.81	17.12± 2.32
solar	291/10-CV	323	3/9	29.44± 0.69	27.61±0.76	28.48± 1.51
waveform-21	300/4700	5000	21/0	29.74± 0.67	28.57±0.66	21.43± 0.60
primary-tumor	305/10-CV	339	0/17	57.99± 0.80	59.56±0.83	51.35± 2.84
liver-disorder	310/10-CV	345	6/0	34.67± 0.77	33.45±0.80	43.78± 2.35
ionosphere	316/10-CV	351	34/0	10.79± 0.57	10.22±0.55	10.28± 1.43
horse-colic	331/10-CV	368	7/15	14.76± 0.57	17.07±0.63	20.14± 2.55
cars	353/10-CV	392	7/1	2.40± 0.27	1.91±0.23	2.04± 0.63
vote	392/10-CV	435	0/16	4.97± 0.31	4.42±0.31	9.66± 0.68
soybean (L)	615/10-CV	683	0/35	8.20± 0.39	8.07±0.34	6.59± 0.85
crx	621/10-CV	690	6/9	14.55± 0.37	15.41±0.41	12.90± 0.79
breast	629/10-CV	699	10/0	5.25± 0.24	4.71±0.25	3.00± 0.50
pima	691/10-CV	768	8/0	25.31± 0.51	25.54±0.52	24.10± 1.75
vehicle	761/10-CV	846	18/0	27.22± 0.47	27.15±0.46	38.88± 1.55
anneal/U	808/10-CV	898	6/32	1.41± 0.12	1.47±0.13	1.45± 0.44
german	900/10-CV	1000	7/13	28.96± 0.42	29.08±0.47	25.90± 1.80
DNA	2000/1186	3186	0/180	7.34± 0.76	6.91±0.74	6.66± 0.72
segment	2079/10-CV	2310	19/0	3.30± 0.11	3.98±0.13	6.88± 0.52
chess	2130/1066	3196	0/36	0.47± 0.21	1.13±0.32	12.85± 1.03
hypothyroid	2847/10-CV	3163	7/18	0.73± 0.05	0.77±0.06	1.42± 0.29
satimage	4435/2000	6435	36/0	14.55± 0.79	14.80±0.79	18.20± 0.86
mushroom	5416/2708	8124	0/22	0.00± 0.00	0.26±0.10	0.78± 0.17
letter	15000/5000	20000	16/0	12.36± 0.47	13.44±0.48	25.02± 0.61
adult	32561/16281	48842	6/8	14.03± 0.27	15.82±0.29	15.82± 0.29

Table 2. Characteristics of datasets and a comparison of C4.5, C4.5-rules, and SBC. The datasets are sorted by training set size. 10-CV indicates 10-fold cross-validation. The numbers after the error indicate the standard deviation of the mean error. The SBC model discretizes using entropy, estimates probabilities using Laplace-m, and ignores unknown values during classification.

Some versions of the SBC, most notably the version described by Cestnik [3], have used an alternative formulation that is mathematically equivalent, but requires estimating $P(C|A)$ instead of $P(A|C)$. Comparisons (not reported here) showed insignificant differences in accuracy between the two methods.

Many researchers have noted the good performance of SBC, including Clark and Niblett [4], Kononenko [17], Langley and Sage [19], and Domingos and Pazzani [5]. Proposed extensions generally resulted in little improvements [16, 18, 22], although some recent proposals seem promising [9, 13].

5 Summary

We studied different options for handling unknowns, estimating probabilities, and discretizing. Through a large scale comparison of 37 datasets, we were able to pinpoint interesting datasets where error differences were significant and explained many of the reasons for different error results. We proposed a new method for estimating probabilities, Laplace-m, that outperformed the other methods on the datasets we tested on.

Using the bias-variance decomposition, we showed that while the SBC performs well on small datasets, it will not generally scale very well to larger datasets because of its strong bias component. We compared the SBC with C4.5 and C4.5-rules and showed that it is accurate and outperforms both inducers on many of the smaller datasets from the UCI repository.

Acknowledgments We would like to thank Pedro Domingos, Jim Kelly, Mehran Sahami and Joel Tesler for their excellent comments and suggestions. We would like to thank Eric Bauer and Clay Kunz for their work on $\mathcal{MLC}$++ [14]. The experiments described here were all done using $\mathcal{MLC}$++.

References

1. Leo Breiman. Heuristics of instability in model selection. Technical Report Statistics Department, University of California at Berkeley, 1994.
2. Leo Breiman. Bias, variance, and arcing classifiers. Technical report, Statistics Department, University of California, Berkeley, 1996. Available at: http://www.stat.Berkeley.EDU/users/breiman/.
3. Bojan Cestnik. Estimating probabilities: A crucial task in machine learning. In Luigia Carlucci Aiello, editor, *Proceedings of the ninth European Conference on Artificial Intelligence*, pages 147–149, 1990.
4. P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
5. Pedro Domingos and Michael Pazzani. Beyond independence: conditions for the optimality of the simple bayesian classifier. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 105–112. Morgan Kaufmann, July 1996.

6. James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In Armand Prieditis and Stuart Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 194–202. Morgan Kaufmann, July 1995.
7. Richard Duda and Peter Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.
8. E. Fix and J.L. Hodges. Discriminatory analysis—nonparametric discrimination: Consistency properties. Technical Report 21-49-004, report no. 04, USAF School of Aviation Medicine, Randolph Field, Tex, February 1951.
9. Nir Friedman and Moises Goldszmidt. Building classifiers using bayesian networks. In Lorenza Saitta, editor, *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 157–165. Morgan Kaufmann, July 1996.
10. Stuart Geman, Eli Bienenstock, and Reneé Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–48, 1992.
11. Irving John Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods.* M.I.T. Press, 1965.
12. Louis Gordon and Richard A Olshen. Almost sure consistent nonparametric regression from recursive partitioning schemes. *Journal of Multivariate Analysis*, 15:147–163, 1984.
13. Ron Kohavi. 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*, pages 114–119, 1996.
14. Ron Kohavi, Dan Sommerfield, and James Dougherty. Data mining using MLC++: A machine learning library in C++. In *Tools with Artificial Intelligence*, pages 234–245. IEEE Computer Society Press, 1996. Received the best paper award. `http://www.sgi.com/Technology/mlc`.
15. Ron Kohavi and David H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference.* Morgan Kaufmann, July 1996. Available at `http://robotics.stanford.edu/users/ronnyk`.
16. Igor Kononenko. Semi-naive bayesian classifiers. In *Proceedings of the sixth European Working Session on Learning*, pages 206–219, 1991.
17. Igor Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence*, 7:317–337, 1993.
18. Pat Langley. Induction of recursive bayesian classifiers. In *Proceedings of the European Conference on Machine Learning*, pages 153–164, April 1993.
19. Pat Langley and Stephanie Sage. Scaling to domains with many irrelevant features. In Russel Greiner, editor, *Computational Learning Theory and Natural Learning Systems.* MIT Press, to appear.
20. Christopher J. Merz and Patrick M. Murphy. UCI repository of machine learning databases.
`http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1996.
21. J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, California, 1993.
22. Moninder Singh and Gregory M. Provan. A comparison of induction algorithms for selective and non-selective bayesian classifiers. In *Machine Learning: Proceedings of the Twelfth International Conference*, pages 497–505, July 1995.