1
Searching for dependencies in Bayesian classifiers

Michael J. Pazzani

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717
pazzani@ics.uci.edu
phone: (714) 824-5888
fax (714) 824-4056

ABSTRACT Naive Bayesian classifiers which make independence assumptions perform remarkably well on some data sets but poorly on others. We explore ways to improve the Bayesian classifier by searching for dependencies among attributes. We propose and evaluate two algorithms for detecting dependencies among attributes and show that the backward sequential elimination and joining algorithm provides the most improvement over the naive Bayesian classifier. The domains on which the most improvement occurs are those domains on which the naive Bayesian classifier is significantly less accurate than a decision tree learner. This suggests that the attributes used in some common databases are not independent conditioned on the class and that the violations of the independence assumption that affect the accuracy of the classifier can be detected from training data.

1.1 Introduction

The Bayesian classifier (Duda & Hart, 1973) is a probabilistic method for classification. It can be used to determine the probability that an example j belongs to class C_i given values of attributes of an example represented as a set of n nominally-valued attribute-value pairs of the form $A_1 = V_{1_j}$:

$$P(C_i | A_1 = V_{1_j} \& \ldots A_n = V_{n_j})$$

If the attributes are independent, this probability is proportional to Equation 1.1

$$P(C_i) \prod_k P(A_k = V_{k_j} | C_i) \tag{1.1}$$

Equation 1.1 is well suited for learning from data, since the probabilities $\hat{P}(C_i)$ and $\hat{P}(A_k = V_{k_j} | C_i)$ may be estimated from the training data. To determine the most likely class of a test example, the probability of each class is computed with Equation 1. A classifier created in this manner is sometimes called a simple (Langley, 1993) or naive (Kononenko, 1990) Bayesian classifier. One important evaluation metric for machine learning methods is the predictive accuracy on unseen examples. This is measured by randomly selecting a subset of the examples in a database to use as training examples and reserving the remainder to be used as test examples. In the case of the simple Bayesian classifier, the training examples are used to estimate probabilities and Equation 1.1 is then used

[1] *AI and Statistics V.* ©1995.

Domain	N	Naive Bayes	ID3
credit	250	.840	.798
horse-colic	200	.810	.783
iris	100	.931	.912
pima-diabetes	500	.755	.723
wine	125	.980	.934
wisc-cancer	500	.973	.952

TABLE 1.1. Accuracy of the naive Bayesian classifier and a simple decision tree learner.

Domain	N	Naive Bayes	ID3
glass	150	.417	.786
krkp	300	.843	.961
mushroom	800	.940	.996
voting	300	.904	.930

TABLE 1.2. Accuracy of the naive Bayesian classifier and decision tree learner on different problems from the UCI archive.

to predict the class of highest estimated probability of each example in the test set. The predictive accuracy is the proportion of agreements between predicted and actual classes.

On many problems, the accuracy of the naive Bayesian classifier is equal to or greater than that of more sophisticated machine learning algorithms. For example, Table 1.1 compares the naive Bayesian classifier to ID3, a simple decision tree algorithm (Quinlan, 1986) on several problems from the UCI archive of machine learning databases (Murphy & Aha, 1995). On each problem, both algorithms were run 24 times on the same training sets and tested on the same disjoint test sets consisting of all examples not in the training set. The number of examples in the training set is given in the second column, the mean accuracy of the Bayesian classifier in the third, and the mean accuracy of the decision tree learner in the fourth column. On each problem, the Bayesian classifier is significantly more accurate at the .05 level using a paired two-tailed t-test. However, on other problems, the reverse pattern is observed (see Table 1.2). On each problem, the Bayesian classifier is significantly less accurate than the decision tree at the .05 level using a paired two-tailed t-test and in one case (glass) the difference in accuracy is greater than 30%.

One possible explanation for the poor performance of the Bayesian classifier on this last set of problems is that the major assumption of the classifier, that the attributes are independent within each class does not hold. In this paper, we address this issue by searching for dependencies among pairs of attributes. For example, if there are three independent attributes, the product of Equation 1.1 would be as follows:

$$\hat{P}(A_1 = V_{1_j}|C_i)\hat{P}(A_2 = V_{2_j}|C_i)\hat{P}(A_3 = V_{3_j}|C_i)\hat{P}(C_i) \tag{1.2}$$

However, if it is known that A_1 and A_3 are not conditionally independent on class membership[2], and A_2 is not relevant in this problem, then Equation 1.3 should be used:

$$\hat{P}(A_1 = V_{1_j} \& A_3 = V_{3_j}|C_i)\hat{P}(C_i) \tag{1.3}$$

[2]In this paper, we will use "conditionally independent" to mean independent conditioned on class membership.

We say that the two attributes A_1 and A_3 are *joined* if Equation 1.3 is used instead of Equation 1.2. A more accurate classifier can result from joining the correct groups of attributes. Joining A_1 and A_3 and eliminating A_2 would be useful if the function to be learned were $A_1 \oplus A_3$. For example, consider classifying an example in which A_1, A_2 and A_3 were equal to 1. In this case, $\hat{P}(A_1 = 1|Class = True)$, $\hat{P}(A_2 = 1|Class = True)$, $\hat{P}(A_3 = 1|Class = True)$, $\hat{P}(A_1 = 1|Class = False)$, $\hat{P}(A_2 = 1|Class = False)$, $\hat{P}(A_3 = 1|Class = False)$, $\hat{P}(Class = True)$, and $\hat{P}(Class = False)$ and would all be close to 0.5 if estimated from from randomly selected examples, and the classifier that assumed independence (i.e., using Equation 1.2) would not be accurate at predicting whether an example is positive or negative. However, if Equation 1.3 were used, the classifier would be accurate since $\hat{P}(A_1 = 1\&A_3 = 1|Class = True)$ would be 1 and $\hat{P}(A_1 = 1\&A_3 = 1|Class = False)$ would be 0.

Joining is an operation that creates a new compound attribute that replaces the original two attributes in the classifier. The possible values of the new attribute are all possible combinations of the the values of the original attributes. For example, if there are two attributes `height` (with values `tall` and `short`) and `weight` (with values `heavy` and `light`), joining these two original attributes will create a new attribute called `weight_height` (with values `tall_heavy`, `tall_light`, `short_heavy` and `short_light`). In order to allow joining on continuously-valued attributes, we discretize them into k equal intervals. We have used a value of 5 for k on all domain and have not experimented with other possible values. Note that joining two attributes differs from conjoining them as is commonly used in some induction systems (e.g., Schlimmer, 1987; Ragavan & Rendell, 1993) because the compound attribute formed by joining is not a Boolean attribute.

Formally, two attributes A and B are independent within each class C_i if for all values V_j of A and V_k of B, $P(A = V_j\&B = V_k|C_i) = P(A = V_j|C_i)P(B = V_k|C_i)$. In practice, when these probabilities are estimated from training data, these quantities will rarely be equal. One approach to deal with this problem is to assume that attributes are conditionally independent unless there is a significant difference according to some statistical criteria (Kononenko, 1991). Here, we consider an alternative approach that does not directly address the question of conditional independence. Instead, we ask, for the purposes of maximizing predictive accuracy, whether it is better to join two attributes. If conditionally independent attributes are joined and the probabilities of joined attributes are estimated from training data, a less accurate classifier may result because the probability estimates of the joined attributes are less reliable than the estimates of individual attributes. This occurs because joined attributes have more values and as a consequence, on average there are fewer examples for each value when compared to the same attributes unjoined.

In this paper, we explore two alternative methods of joining attributes while learning Bayesian classifiers: Forward Sequential Selection and Joining (FSSJ) and Backward Sequential Elimination and Joining (BSEJ). Both methods are related to approaches that have been used in other learning methods for selecting relevant features (Kittler, 1986; John, Kohavi, Pfleger, 1994).

1.2 Selecting and Joining Attributes

Both of the algorithms for joining attributes that we explore have a similar control structure. Each algorithm maintains a set of attributes (which is a subset of the attributes used to describe the examples) to be used by the classifier and a description of which attributes have been joined. Each algorithm estimates the accuracy of a Bayesian classifier by leave-one-out cross-validation on the training data using the attributes joined as indicated. Leave-one-out evaluation is used because it allows a single Bayesian classifier to be constructed on the entire training set. To classify a training example, the contribution of that example to the probability estimates is subtracted out (Langley, 1993). Each algorithm considers a set of possible operations (such as joining two attributes) and selects the operation that most improves the accuracy of the classifier as measured by leave-one-out cross validation. The two algorithms differ in how they create an initial Bayesian classifier and the operators they use to improve upon the classifier. Both algorithms use what John, Kohavi and Pfleger (1994) call the wrapper model because attribute joining operates without knowledge of how the Bayesian classifier operates. Instead, the attribute joining algorithms are only concerned with the accuracy of the resulting classifiers, as determined by leave-one-out cross-validation on the training data.

1.2.1 Forward Sequential Selection and Joining

The forward sequential selection and joining (FSSJ) algorithm initializes the set of attributes to be used by the Bayesian classifier to the empty set. A Bayesian classifier with no attributes simply classifies all examples to the most frequent class that occurs in the training data. Next, two operators are used to generate new classifiers:

a. Consider adding each attribute not used by the current classifier as a new attribute conditionally independent of all other attributes used in the classifier.

b. Consider joining each attribute not used by the current classifier with each attribute currently used by the classifier.

At each step in the classifier, every addition and every joining of an unused attribute with a used attribute is considered and evaluated using leave-one-out on the training data. If no change makes an improvement, the current classifier is returned. Otherwise, the change that makes the most improvement is retained and the process of modifying the classifier is repeated. Note that by repeated applications of the second operator, more than two attributes may be joined. However, the original attributes cannot be joined separately with other attributes.

The forward selection and joining algorithm differs from forward sequential selection (Kittler, 1986; John, Kohavi and Pfleger, 1994; Moore and Lee, 1994; Caruana and Freitag, 1994) in that forward sequential selection and joining has the ability to join attributes, while forward sequential selection can only select subsets of attributes. Selecting a subset of the original attributes is useful in some classifiers such as nearest neighbor that have problems with irrelevant attributes. Irrelevant attributes do not tend to be a major problem for Bayesian classifiers since, for an irrelevant attribute A, $\hat{P}(A = V|C_i) = \hat{P}(A = V|C_j) = \hat{P}(A = V)$.

In the worst case, $O(A^3)$ Bayesian classifiers are constructed and evaluated (where A is the number of attributes) using FSSJ, since at most A steps (either adding or joining) occur, and the worst step requires considering joining O(A) unused attributes to O(A) used attributes.

1.2.2 Backward Sequential Elimination and Joining

The backward sequential elimination and joining (BSEJ) algorithm initially creates a Bayesian classifier treating all attributes as conditionally independent. It uses two operators for considering new hypotheses:

a. Consider replacing each pair of attributes used by the classifier with a new attribute that joins the pair of attributes.

b. Consider deleting each attribute used by the classifier.

Like the FSSJ algorithm, the BSEJ algorithm considers all one step modifications of the algorithm, evaluates these using leave-one-out cross validation on the training data, and permanently makes the change with the greatest improvement. If no change results in an improvement, the current classifier is returned. Otherwise, changes to the modified classifier are considered. By repeated application of the joining operator, several attributes may be joined. Like FSSJ, in the worst case, O(A^3) Bayesian classifiers are evaluated using BSEJ.

1.3 Experimental Results

We ran experiments comparing the algorithms for selecting and joining attributes to the naive Bayesian classifier. These experiments were run on the same domains and in the same manner as the previous experiments. For FSSJ and BSEJ, we also recorded the number of attributes selected for use by the classifier and the number of joins that were made. The results are shown in Table 3.

The first four domains are domains on which a decision tree learner is substantially more accurate than the naive Bayesian classifier. Forward sequential selection and joining of attributes significantly ($p < .05$ using a paired, two-tailed t-test) increases the accuracy on three of the domains (as indicated by a + following the accuracy) but significantly decreases the accuracy on one (as indicated by a $-$ following the accuracy) when compared to the naive Bayesian classifier. The last six domains are ones in which the naive Bayesian classifier is more accurate than a decision tree learner. Forward sequential selection and joining of attributes has no significant effect on four of these domains, and significantly decreases accuracy on two.

Backward sequential elimination and joining of attributes improves the accuracy of the Bayesian classifier on all four problems when the Bayesian classifier is less accurate than the decision tree, and has no significant effect on the six problems on which the Bayesian classifier is more accurate than the decision tree. In addition, on the glass problem, there is a large amount of improvement. On this problem, both FSSJ and BSEJ create classifiers that use approximately three of the nine available attributes. By comparing the total number of attributes (second column) to the number of attributes used after BSEJ (eight

Domain	Atts	Naive Acc	FSSJ Acc	FSSJ Atts	FSSJ Joins	BSEJ Acc	BSEJ Atts	BSEJ Joins
glass	9	.417	.765+	3.0	2.0	.768+	3.2	1.8
krkp	36	.843	.793–	4.4	1.4	.931+	33.5	6.0
mushroom	22	.940	.984+	2.3	1.0	.992+	21.0	3.9
voting	16	.904	.949+	2.7	.5	.920+	13.1	2.6
credit	15	.840	.836	5.0	2.7	.833	14.3	4.5
horse-colic	21	.810	.809	5.5	2.3	.802	20.1	6.5
iris	4	.931	.942	2.1	.5	.938	3.8	1.0
pima	8	.755	.740	3.7	2.0	.749	7.7	4.0
wine	13	.980	.954–	3.3	1.9	.975	12.8	0.8
wisc-cancer	9	.973	.959–	3.5	1.7	.971	8.6	0.8

TABLE 1.3. A comparison of the attribute joining algorithms on ten databases from the UCI Archive.

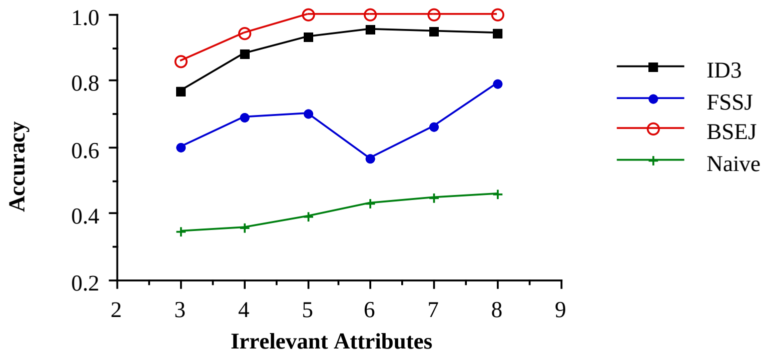

FIGURE 1. The accuracy of four learning algorithms at learning exclusive-or with irrelevant attributes as a function of the number of irrelevant attributes when training on 50% of the data and testing on the remaining 50%.

column), one can see that few attributes are eliminated by BSEJ. In contrast, FSSJ eliminates many more attributes than BSEJ resulting a much simpler classifier. On most problems, this simplicity makes little difference in accuracy. However, on some problems (e.g., krkp– Rook King Pawn chess end games), using fewer attributes and making fewer joins results in a classifier that is less accurate than BSEJ.

By joining attributes, BSEJ and FSSJ have the potential to overcome this limitation. To demonstrate how joining attributes allows a Bayesian classifier to learn non-linearly separable functions, we consider learning *exclusive-or* of two attributes, a function that is very difficult for a naive Bayesian classifier. We tested *exclusive-or* functions with 3, 4, 5, 6, 7, and 8 irrelevant attributes included in the example description. For each function, we ran 20 trials in which 50% of the examples were randomly selected for training and the remaining 50% were used to test the accuracy. On each training set, we tested ID3, the naive Bayesian classifier, the Bayesian classifier with Backward Sequential Elimination and Joining, and the Bayesian classifier with Forward Sequential Selection and Joining. Figure 2 shows the mean accuracy of the four algorithms, plotted as a function of the number of irrelevant attributes. Paired t-tests at the .001 level indicate that BSEJ is more accurate than the other algorithms on this problem. This particular problem illustrates a potential weakness of the FSSJ algorithm which does particularly poorly on this problem.

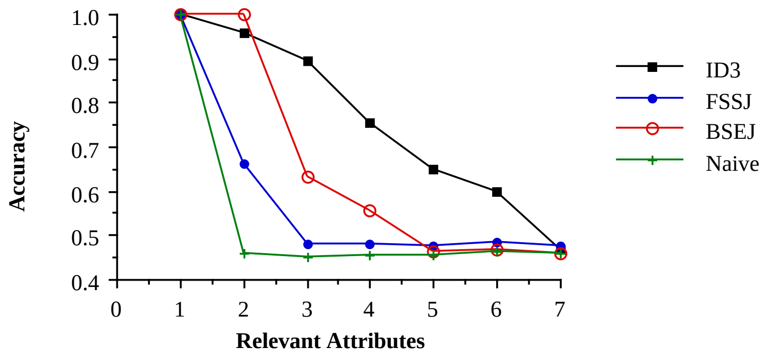

FIGURE 2. The accuracy of four learning algorithms at learning parity functions as a function of the number of relevant attributes. For each function, there were a total of 10 attributes and 1024 total examples. The algorithm is trained on 50% of the data and tested on the remaining 50%.

In order for FSSJ to consider joining two attributes, it must first construct a classifier in which one of the attributes is used independently. However, on exclusive-or, the Bayesian classifier with just one relevant attribute would not perform better than simply guessing the most frequent class and the correct relevant attributes are rarely joined. In contrast, BSEJ starts with the naive Bayesian classifier of all attributes and considers all pairs of joins. Since it always considers joining the two relevant attributes, it usually makes this join (especially on larger data sets) and consequently has a very high accuracy.

A potential weakness of the algorithm for joining attributes is that in order to join more than two attributes, it must first join two of the attributes and later join other attributes with the original two. This will not occur unless forming the first pair results in an increase in accuracy. This problem is common to all algorithms that use hill-climbing search. In a second experiment on artificial data, we evaluate the ability of the four algorithms to learn various parity functions. A parity function is true iff an odd number of the attributes have a value of true. We varied the number of relevant attributes from 1 to 7 and included irrelevant attributes so that each example had a total of 10 attributes. We ran 20 trials of each function, choosing 512 examples for training and 512 examples for testing. The results displayed in Figure 3 show that the Bayesian classifier with BSEJ is more accurate than the naive Bayesian classifier when there are fewer than 5 relevant attributes in the parity function.

When there are between 3 and 6 relevant features, ID3 is significantly more accurate than the Bayesian classifier with BSEJ. This shows that although BSEJ only considers joining pairs of attributes, it can improve the accuracy of the Bayesian classifier when there are more than two interacting attributes. However, there is still room for improvement since ID3 is more accurate. Although it would be possible to consider joining three (or more attributes), the computational complexity makes it impractical for most databases.

1.4 Related Work

1.5 Limitations and Future work

There are several limitations of the backward sequential elimination and joining method we have proposed. First, unlike the naive Bayesian classifier, it is not an incremental

Domain	Naive Bayes	BSEJ	FSS
glass	.417	.768+	.737
krkp	.843	.931+	.739
mushroom	.940	.992+	.984
voting	.904	.920	.956+
credit	.840	.833	.830
horse-colic	.810	.802	.811
iris	.931	.938	.941
pima-diabetes	.755	.749	.741
wine	.980	.975+	.957
wisc-cancer	.973	.971+	.956
x-or	.453	.921+	.444
Parity	.360	.556+	.446

TABLE 1.4. Accuracy of two methods for improving the naive Bayesian classifier: BSEJ proposed in this paper, and FSS proposed by Langley & Sage (1994).

algorithm. Second, for efficiency reasons, it only considers joining pairs of attributes. Although it can join more than two attributes, this must be done in multiple steps. Finally, the algorithm requires symbolic data only, so numeric domains must be discretized, losing some information.

The Bayesian classifier is a natural choice for minimizing misclassification costs. The probability than an example belongs to each class is returned by the Bayesian classifier and this information, together with information on the cost of various errors can be used to choose the class with the least expected cost of error. Pazzani, Merz, Murphy, Ali, Hume, and Brunk (1994) evaluate several algorithms for reducing the misclassification cost of learning algorithms and show that the Bayesian classifier performs well at reducing misclassification costs when it is as accurate as other learners. Since we have improved upon the accuracy of the Bayesian classifier, it is possible that the BSEJ algorithm will provide additional benefits in reducing misclassification costs.

1.6 Conclusions

We have shown that, when learning Bayesian classifiers, searching for dependencies among attributes results in significant increases in accuracy. We proposed and evaluated two algorithms and shown that the backward sequential elimination and joining algorithm provides the most improvement. The domains on which the most improvement occurs are those domains on which the naive Bayesian classifier is significantly less accurate than a decision tree learner. This suggests that the attributes used in some common databases are not conditionally independent and that the violations of the conditional independence assumption that affect the accuracy of the classifier can be detected from training data.

Acknowledgements

The research reported here was supported in part by NSF grant IRI-9310413 and ARPA grant F49620-92-J-0430 monitored by AFOSR. I'd like to thank Pedro Domingos, Dennis Kibler, Pat Langley and Kamal Ali for advice on Bayesian classifiers, Cullen Schaffer for advice on cross-validation and David Schulenburg and Clifford Brunk for comments on an earlier draft of this paper.

References

Almuallim, H., and Dietterich, T. G. (1991). Learning with many irrelevant features. In *Ninth National Conference on Artificial Intelligence*, 547-552. MIT Press.

Caruana, R., & Freitag, D. (1994). Greedy attribute selection. In Cohen, W., and Hirsh, H., eds., *Machine Learning: Proceedings of the Eleventh International Conference.* Morgan Kaufmann

Cooper, G. & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309-347.

Danyluk, A. & Provost, F. (1993). Small disjuncts in action: Learning to diagnose errors in the telephone network local loop. *Machine Learning Conference*, pp 81-88.

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

John, G. Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the subset selection problem *Proceedings of the Eleventh International Conference on Machine Learning.* New Brunswick, NJ.

Kittler, J. (1986). Feature selection and extraction. In Young & Fu, (eds.), *Handbook of pattern recognition and image processing.* New York: Academic Press.

Kononenko, I. (1990). Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In B. Wielinga (Eds..), *Current trends in knowledge acquisition.* Amsterdam: IOS Press.

Kononenko, I. (1991). Semi-naive Bayesian classifier. *Proceedings of the Sixth European Working Session on Learning.* (pp. 206-219). Porto, Portugal: Pittman.

Langley, P. (1993). Induction of recursive Bayesian classifiers. *Proceedings of the 1993 European Conference on Machine Learning.* (pp. 153-164). Vienna: Springer-Verlag.

Langley, P. & Sage, S. (1994). Induction of selective Bayesian classifiers. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence.* Seattle, WA

Moore, A. W., and Lee, M. S. 1994. Efficient algorithms for minimizing cross validation error. In Cohen, W. W., and Hirsh, H., eds., Machine Learning: *Proceedings of the Eleventh International Conference.* Morgan Kaufmann.

Murphy, P. M. , & Aha, D. W. (1995). UCI Repository of machine learning databases. Irvine: University of California, Department of Information & Computer Science.[Machine-readable data repository ftp://ics.uci.edu:/pub/machine-learning-databases/]

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., and Brunk, C. (1994). Reducing Misclassification Costs. *Proceedings of the Eleventh International Conference on Machine Learning.* New Brunswick, NJ.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Mateo, CA: Morgan Kaufmann.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning, 1*, 81-106.

Rachlin, Kasif, Salzberg & Aha, (1994). Towards a better understanding of memory-based reasoning systems. *Proceedings of the Eleventh International Conference on Machine Learning.* New Brunswick, NJ.

Ragavan, H. & Rendell, L. (1993). Lookahead feature construction for learning hard concepts. *Machine Learning: Proceedings of the Tenth International Conference.* Morgan Kaufmann

Schlimmer, J. (1987). Incremental adjustment of representations for learning. *Machine Learning: Proceedings of the Fourth International Workshop.* Morgan Kaufmann

Schaffer, C. (1994). A conservation law of generalization performance *Proceedings of the Eleventh International Conference on Machine Learning.* New Brunswick, NJ.