Towards a Better Understanding of Memory-Based Reasoning Systems

John Rachlin Simon Kasif
Steven Salzberg
Dept. of Computer Science
Johns Hopkins University
Baltimore, MD 21218
lastname@cs.jhu.edu

David W. Aha
Navy Center for Applied Research in AI
Naval Research Laboratory
Washington DC 20375
aha@aic.nrl.navy.mil

Abstract

We quantify both experimentally and analytically the performance of memory-based reasoning (MBR) algorithms. To start gaining insight into the capabilities of MBR algorithms, we compare an MBR algorithm using a value difference metric to a popular Bayesian classifier. These two approaches are similar in that they both make certain independence assumptions about the data. However, whereas MBR uses specific cases to perform classification, Bayesian methods summarize the data probabilistically. We demonstrate that a particular MBR system called PEBLS works comparatively well on a wide range of domains using both real and artificial data. With respect to the artificial data, we consider distributions where the concept classes are separated by functional discriminants, as well as time-series data generated by Markov models of varying complexity. Finally, we show formally that PEBLS can learn (in the limit) natural concept classes that the Bayesian classifier cannot learn, and that it will attain perfect accuracy whenever Bayes does.

1 INTRODUCTION

As scientific databases continue to grow, the analysis of scientific data becomes an increasingly important application area for machine learning research. Much of the research on scientific data is directed towards predicting properties of a physical process. Such processes are often described in terms of a function defined over the attributes of the domain or a stochastic model such as a Markov chain. The first question one should ask when studying such problems is: what is the best machine learning technique to use for this problem domain? A general solution to this problem remains elusive. It has been argued (e.g., by the case-based reasoning community) that instead of producing a description of the problem domain in terms of logical rules, functional descriptions, or a complex statistical model, it is possible to store a collection of memories (cases) and perform prediction by interpolating from them. In this practical context, the second question to ask is: are memory-based reasoning (MBR) methods as effective for a given problem domain as more complex approaches; e.g., probabilistic networks. This paper addresses this question both experimentally and formally.

We were originally motivated to study the relative strengths and weaknesses of probabilistic vs. memory based reasoning approaches because of the recent application of these methods to the analysis of biological data such as DNA and RNA sequences, proteins, and genetic mapping data. Consider, for example, the problem of predicting the secondary structure of a protein from its primary amino acid sequence. These amino acids have many important physical and chemical properties (e.g., hydrophobicity, polarity, and various geometric features) that influence their folding behavior. Thus, the solution to this fundamentally important problem from molecular biology might be expressed as a function of a large number of (currently unknown) variables. Alternatively, it may be modelled as a stochastic process involving the forces between the atoms that comprise the protein. Although both approaches are areas of active research, a precise description of the physical process and all the relevant attributes is not known. Instead, most current methods approach this problem using a symbolic representation for the 20 amino acids as the primary input. Partially as a result of this representational limitation, most current approaches for predicting protein structure have met with only moderate success, typically achieving about 64% accuracy. More intricate methods obtain slightly better results, but the problem is still largely unsolved. Two relatively successful techniques for this and similar problems in computational biological have been Bayesian and MBR methods (Zhang, Mesirov, & Waltz, 1992; Cost & Salzberg, 1993; Yi & Lander, 1993; Delcher, Kasif, Goldberg, & Hsu, 1993). In addition, there are other prominent domains where MBR and probabilistic methods are currently competing for best performance, e.g., speech recognition and natural language parsing.

We therefore decided to perform a detailed study of the relative strengths of Bayesian and MBR approaches for a range of different prediction tasks involving symbolic attributes. As a baseline for the comparison, we also demonstrate the performance of a nearest neighbor algorithm using the simple overlap metric. This study is particularly intriguing because of several important similarities that exist between MBR methods (with value-difference metrics) and Bayesian classifiers:

- both methods make the same independence assumptions,
- both require the same computational time and space during training, although the output space complexity and classification time is higher for the MBR algorithm,
- these are the primary methods employed for the study of the fundamental biology problems cited above, and
- the Bayesian classifier abstracts case-specific information into a set of conditional and unconditional probabilities, while MBR stores the cases themselves. Thus, our comparisons highlight the utility of using case-specific information.

Our analysis substantially enhances our understanding of the relative strengths and weaknesses of these methods, and suggests that MBR has a number of important advantages.

We begin by detailing these algorithms in Section 2. Section 3 then demonstrates that PEBLS (our MBR system) attains comparatively good accuracies on a set of standard classification tasks. In Section 4 we study how our classifiers perform on data characterized by some underlying functional description. We then compare their performance on data generated by Markov models, which corresponds to the approach taken for scientific data analysis problems in molecular biology (Haussler, Krogh, Mian, & Sjolander, 1992) and speech recognition (Rabiner, 1989). Our comparison indicates that PEBLS has faster initial learning rates than Bayes for this class of problems. We then prove in Section 5 that PEBLS can learn (in the limit) a larger class of functions than can the Bayesian classifier. We also prove that PEBLS will always attain perfect accuracy (in the limit) whenever Bayes does.

2 DESCRIPTIONS AND RELATED WORK

The nearest neighbor (NN) and simple Bayesian classifiers are well documented elsewhere. Briefly, NN classifies an instance X according to the class of the stored instance whose Euclidean distance to X is minimal. NN has been analyzed for its learning behavior in the limit (Cover & Hart, 1967), in PAC analyses (Aha, Kibler, & Albert, 1991; Albert & Aha, 1991), and for its average case behavior (e.g., (Langley & Iba, 1993)). These analyses concerned numeric-valued features. Stanfill and Waltz (Stanfill & Waltz, 1986; Stanfill, 1987) introduced the Value Difference Metric (VDM) to define similarity when using symbolic-valued features and empirically demonstrated its benefits. They argued that the VDM is more appropriate than the simple *overlap* distance measure, which defines distance as the number of mismatching features between two instances. More recently, Cost and Salzberg (Cost & Salzberg, 1993) demonstrated that PEBLS, which uses a modification of the VDM distance measure called MVDM, can outperform the overlap measure and several other learning algorithms on a number of practical problems. However, these studies did not perform a detailed analysis of MVDM's learning ability, which is a primary contribution of this paper.

The MVDM used by PEBLS defines the distance between two examples as the sum of the value differences across all features. The value difference between two values v_1 and v_2 of a given feature is defined as:

$$\delta(v_1, v_2) = \sum_{i=1}^{k} \left| \frac{v_{1_i}}{V_1} - \frac{v_{2_i}}{V_2} \right| \qquad (1)$$

where k is the number of classes, v_{1_i} is the number of times v_1 occurred for instances of class C_i and V_1 is the number of times v_1 occurred for all classes. Thus the ratio v_{1_i}/V_1 is the probability of class C_i

given that the feature has value v_1, $P(C_i|v_1)$.

The simple Bayesian classifier represents each concept with a single probabilistic expression, defined by class base rates and feature probability distributions. For each class C_i and feature value v_j, we estimate $P(v_j|C_i)$ from the training data. A new point is classified into C_i if $P(C_i)\prod_j P(v_j|C_i)$ is maximal. This classifier has been evaluated in some recent machine learning papers (e.g., (Clark & Niblett, 1989) and its average case behavior has also been analyzed (e.g., (Langley & Iba, 1992)). In addition, many variations on the Bayesian approach have been considered (e.g., (Kononenko, 1991; Michie & Attar, 1991)). (Of course, the classic work on Bayesian classifiers goes back many years, to well before the machine learning community began using them (Duda & Hart, 1973; Aoki, 1965; Hughes, 1968; Chandrasekan, 1971; Chandrasekan & Harley, 1969; Albrecht & Werner, 1968).)

Computationally, PEBLS computes the same statistics as the Bayesian classifier. However, while Bayes summarizes these statistics in simple rules, PEBLS uses them as part of an MBR classifier.

3 EMPIRICAL STUDIES ON EXISTING DATABASES

In this section, we describe a number of experiments designed to better understand the relative performance characteristics of Bayes, PEBLS, and NN using the overlap method (which counts the number of feature value mismatches between two examples). In particular, we wanted to determine if the performance of PEBLS was comparable to or better than the Bayesian classifier. With this goal in mind, we selected eight datasets from the Univeristy of California Irvine (UCI) Repository of Machine Learning Databases. All methods treated the feature values as symbols, even if the values were numbers. In other words, if a feature had values 1, 2, and 3, we would treat those exactly the same is if they were A, B, and C. Continuous valued-features were discretized by dividing them into 10 equal intervals.

In the case of the letter recognition and mushroom databases, which are both relatively large, we trained the algorithms on 90% of the examples, and tested on the remaining 10%. This process was repeated on ten randomly chosen training sets. The values contained in the table below represent the average of these ten trials. All other tests were conducted using the leave-one-out strategy (i.e., each instance is tested after first training on all other instances in the dataset). The results shown in Table 1 clearly demonstrate that although the relative performance of Bayes and PEBLS varies, PEBLS is the best method overall on the eight databases. It performs much better than Bayes in the tictactoe, letter, and soybean domains, and it also performs substantially better in the zoology and iris domains, though the difference is not as striking. Performance in the other three domains is roughly equal. The NN method using the overlap metric showed mixed results. Because the nature of the concept class is not perfectly understood for these data sets, we turned to artificial data to better gauge the specific conditions under which one method outperforms another.

Table 1. Results on eight UCI databases. The highest accuracy for each database appears in boldface.

	BAYES	PEBLS	NN
TicTacToe	69.4	**94.4**	81.7
Letter	74.6	**95.7**	88.5
Soybean	81.4	**93.2**	91.2
Zoology	89.1	95.0	**96.0**
Iris	90.7	**95.3**	92.0
Promoter	**91.5**	90.6	80.5
WI-Breast	**96.7**	**96.7**	95.6
Mushroom	99.7	**100.0**	**100.0**

4 TESTS ON ARTIFICIAL DATA

In this section we describe tests that compare the performance of Bayes, PEBLS, and NN when applied to artificial datasets. We are primarily interested in two types of data: those which exhibit some specific class distribution based on an underlying functional description, and those which are generated by some simple stochastic process. Artificially defined datasets are of interest because they allow one to control the nature and distribution of examples, and are thus more amenable to formal analysis. In Section 4.1 we consider a number of basic functional distributions for 2-dimensional data on a finite grid. We show relative learning curves for PEBLS and Bayes. In Section 4.2, we consider linearly separable data sparsely distributed over a five-dimensional feature-space. In Section 4.3, we consider stochastic data generated by two simple Markov models.

4.1 TWO-DIMENSIONAL FEATURE SPACE

Our first set of tests considered 10,000 points on an evenly spaced 100×100 grid on the unit square. For simplicity, the examples have just two class labels, A and B. Some of the class distributions that we examined are shown in Figure 1. Note that our classification methods treat grid coordinates as ar-

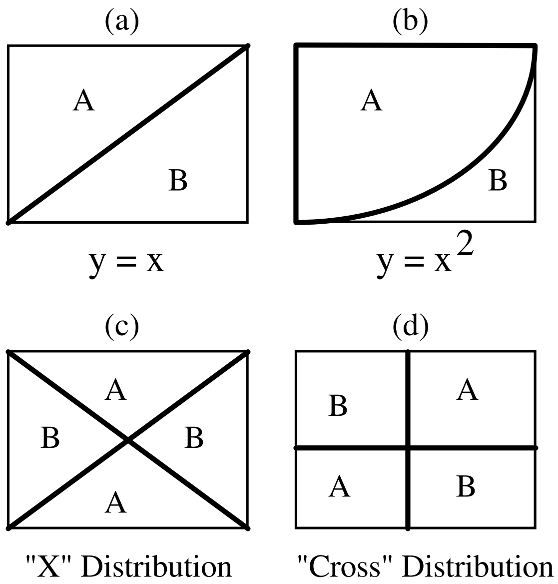

Figure 1: Class distributions with examples uniformly distributed on a 2D lattice

bitrary symbols in a feature space, and thus have no knowledge of the underlying Euclidean space.

In each experiment, we trained the algorithms on some fixed percentage of the points, and then tested on *all* grid points. All points were randomly selected from the uniform distribution. Figure 2 shows the relative performances of Bayes, PEBLS, and NN on these class distributions. In all cases, PEBLS did as well as or better than Bayes. It is interesting to note that in the case when the two classes are separated by the function $y = x^2$, Bayes does not converge to 100% accuracy even when trained on all possible examples. (See Section 5.1 for more details.)

PEBLS and Bayes show some unusual behavior when applied to the "Cross" distribution. While Bayes' accuracy remains at 50%, PEBLS begins to improve, but then eventually drops off, and when all instances are trained, the performance returns to the 50% level. This occurs because for for a specific value of x (or y), the number of points in Class A and Class B is about the same. As a result, the x and y values considered independently provide no hint as to the relative probability of each class. Thus, Bayes can only guess the correct class. However, PEBLS can take advantage of the fact that for any point, it will assign a distance of zero to itself. When the sample size is small, other points will be assigned non-zero distances due to imperfect sampling. Thus, as the training set size begins to increase, so does PEBLS' accuracy. However, the Modified Value Dis-

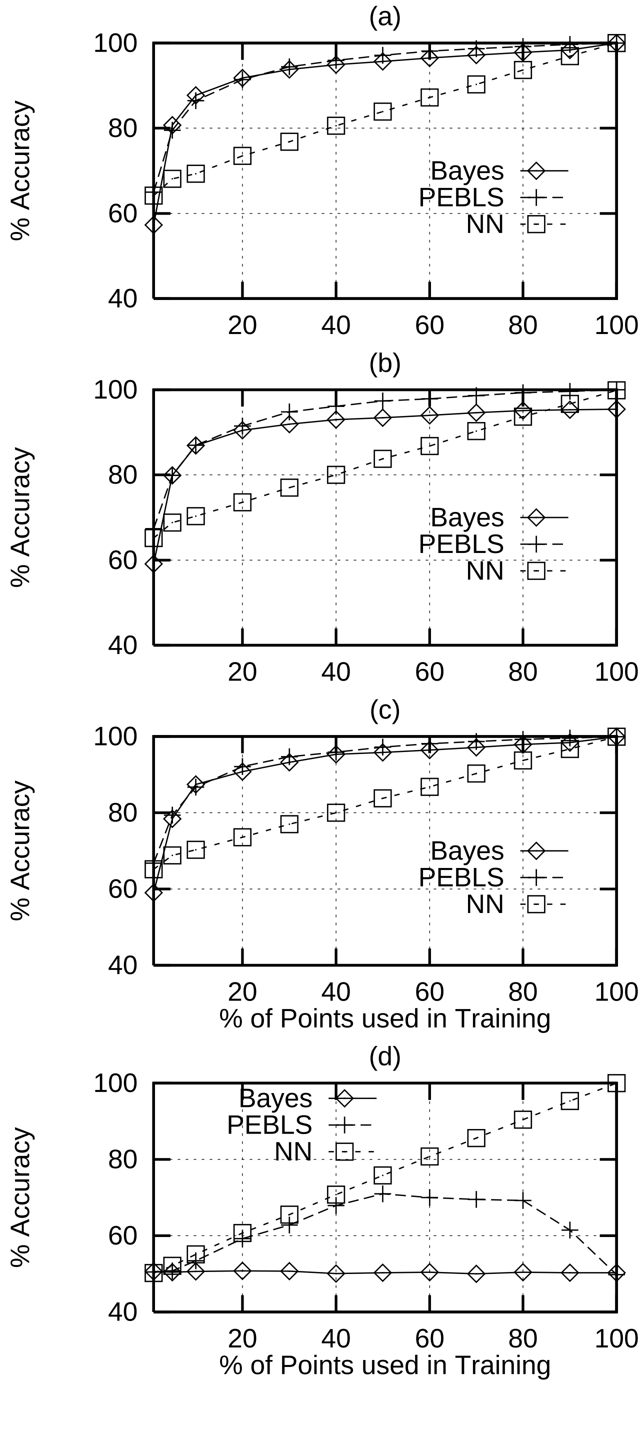

Figure 2: Comparative accuracy of Bayes, PEBLS (MVDM), and NN (Overlap) on (a) $y = x$, (b) $y = x^2$, (c) "x"-distribution, and (d) "cross"-distribution.

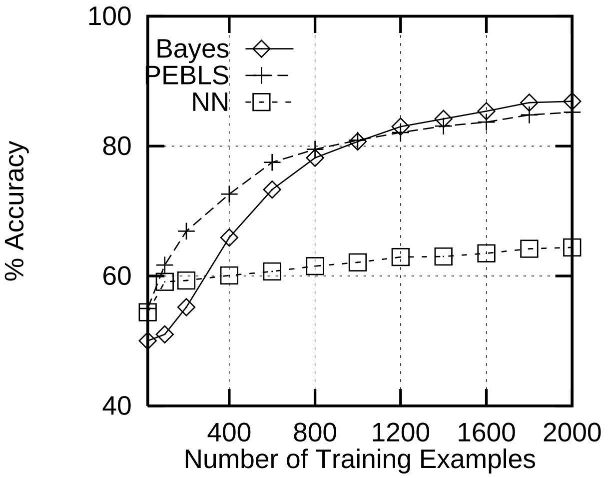

Figure 3: Sparse data in a 5-dimensional Hypercube

tance Metric employed by PEBLS will eventually assign a zero distance between all points in the space once every point in this particular class distribution is included in the training set. When this occurs, PEBLS will also have to guess the correct classification. This explains why PEBLS' accuracy begins to degrade as the training set size increases beyond a certain point. Figure 2 shows that NN using the overlap metric normally produces poor results, although its accuracy is proportional to the training set size, and thus increases linearly.

4.2 SPARSE DATA IN A 5-DIMENSIONAL FEATURE SPACE

In our next experiment, we considered linearly separable examples in higher dimensions. In particular, we focused on a 5-dimensional hypercube containing 100^5 possible instances. (Again, coordinate values are treated only as symbols.) For this domain, we generated three random five-dimensional hyperplanes. For each hyperplane, we created a training set and a test set of 2,000 instances each, where 50% of the instances were above the hyperplane (class A), and 50% of the instances were below (class B). Unlike the previous experiment, our dataset represents only a tiny fraction of the total space (i.e., 4,000 out of 10^{10} possible points). Thus, the probability that a test example will be in the training set is negligible.

Figure 3 summarizes our results. For each training size, three different training subsets were chosen. Each point in the figure is thus an average of nine different values: three distinct hyperplanes, and for each hyperplane, three different training sets for each sample size. Figure 3 shows that when the training size is small, PEBLS outperforms Bayes, but when we are using the complete training set, Bayes does slightly better. As before, we also show how NN performed. Its relatively low accuracy underscores the importance of using more sophisticated distance metrics in symbolic feature domains.

4.3 MARKOV SEQUENCES

In this section we apply PEBLS, Bayes, and nearest-neighbor using the overlap metric to data sequences generated by two simple Markov models. The literature on Markov models is quite extensive. (For a thorough introduction see (Rabiner, 1989).) A Markov model is simply a graph in which the nodes represent states and the arcs define transitions between states. Attached to each arc is a transition probability which defines the probability that a given arc is actually taken when in a particular state. When a transition is taken, the model outputs a single symbol according to an output probability distribution, M, associated with each particular transition. This distribution defines for each of N different symbols, the probability that any particular symbol will be generated. (In the experiments described below, we set N=100, although preliminary results suggest that the outcome is similar for other values of N.)

In our first experiment, the goal was to distinguish which of two models produced a particular output sequence. We considered the Markov model defined by the simple chain of states shown in Figure 4. The model begins in state 0. At each time step, the model makes a transition to the next state while outputting a single symbol. (Since for every state there is only one possible next state, the transition probabilities are all 1.0.) The model shown in the figure will thus produce an output sequence of length five each time it is executed.

In this experiment, we created two such models, each with their own particular (randomly assigned) output distributions. We classify output sequences according to the model that produced it. Our dataset consists of 2000 total sequences, half of which are used for training, and half for testing. Furthermore, the training and test instances are divided evenly between output sequences generated by each of the two random models. Our goal, therefore, is to determine whether PEBLS or Bayes is better able to predict which of the two models generated a particular output sequence.

We trained on a varying number of instances, and always tested on all 1000 test instances. For each

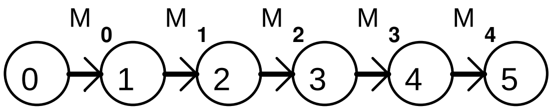

Figure 4: A simple Markov model that generates five independent features

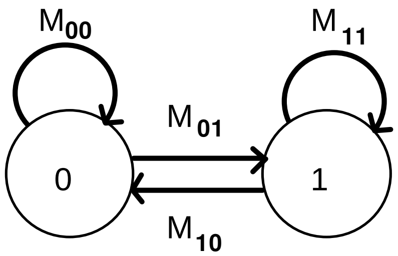

Figure 5: A Two-State Markov Model. The final state defines the class of the output sequence

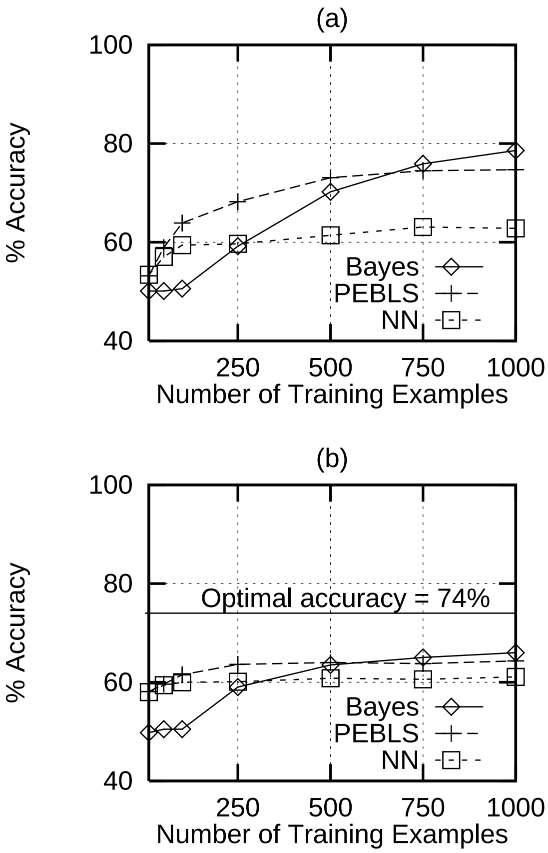

Figure 6: (a) Distinguishing two models, (b) Predicting the final state.

training percentage, we generated three random training subsets, and this entire procedure was repeated for three pairs of random Markov models. Thus each point in Figure 6a is the average of nine values.

In our second experiment involving Markov processes, we consider the two-state model shown in Figure 5. In this case, our goal is to allow a single model (with randomly assigned transition probabilities and output matrices) to generate all of the training and test sequences. We classified each sequence according to the final state of the model. Using known feed-forward algorithms, it is possible to predict, given a complete description of the Markov model and the output, the most probable final state of the model. Classification based on this algorithm is optimal. The problem, of course, is that the underlying model is not normally known. Nevertheless, this optimal classification rate is useful because it provides a basis for comparison by showing how close to the optimal PEBLS and Bayes actually come. Results of this experiment are provided in Figure 6b.

Although these experiments are quite different, the overall character of the two graphs is strikingly similar. Both results show that when the training set is small, PEBLS will significantly outperform Bayes. As we increase the number of training examples, Bayes eventually does better, though not substantially better. Nearest-neighbor using the overlap metric outperformed Bayes when the number of training instances was small, but its overall behavior is generally much worse. These graphs reveal behavior very similar to that shown in Figure 3. This demonstrates that PEBLS and Bayes exhibit consistent performance characteristics over a wide range of scientific domains.

5 ANALYSIS

In this section we give a preliminary theoretical analysis of the performance of Bayes vs. PEBLS. These results also apply to classifiers using VDM. We ask the following questions:

- What distributions are learnable by PEBLS and not learnable by Bayes, and vice versa?
- What is the convergence rate of PEBLS vs. Bayes on important distributions?

We focus here on the first of these questions. Although this analysis is straightforward, it provides substantial intuition for the relative strengths and weaknesses of PEBLS vs. Bayes. The second question is a subject for future work.

5.1 BAYES CLASSIFIER AND LINEARLY SEPARABLE CONCEPTS

In section 4.1, we noted that under certain conditions, Bayes did not converge to 100% accuracy even when the classifier was trained on the entire space of examples. Of course, this is not very surprising because we are making an independence assumption that is known to be false. In this section we review ability of Bayes to learn concept classes in Z^2 when the discriminant function is of the form $y = ax$. It is easy to show that if the separator is of the form $y = ax$ on the rectangle $[0, 0] \times [k, ka]$ then Bayes does converge to 100% accuracy, which is somewhat counter-intuitive at first glance because the features are dependent. However, with a little analysis, one can prove that for many discriminant functions on the unit square (including $y = ax$) Bayes does not converge to 100% accuracy. To illustrate this we consider a *linearly separable* case where the discriminating function is $y = ax$. We will show that for this function, Bayes will not converge to 100% accuracy when $a \neq 1$.

Assume that all points in the unit square above $y = ax$ are in class A, and all points below are class B. Without loss of generality, assume that $a \geq 1$. Note that for a point to be classified by Bayes as Class A, we must have the condition that:

$$P(A|x, y) > P(B|x, y) \tag{2}$$

Making some common independence assumptions, we can write:

$$P(A|x, y) = \frac{P(A)P(x|A)P(y|A)}{P(x)P(y)} \tag{3}$$

Using Bayes' rule for $P(x|A)$ and $P(y|A)$, we can rewrite equation 3 as:

$$P(A|x, y) = \frac{P(A|x)P(x)P(A|y)P(y)}{P(A)P(x)P(y)}$$

Performing the same calculation for $P(B|x, y)$ and applying Eq. 2 yields a new expression for when Bayes will classify a point as A:

$$\frac{P(A|x)P(A|y)}{P(A)} > \frac{P(B|x)P(B|y)}{P(B)} \tag{4}$$

For the class separator $y = ax$ in the range $(0, 0) \ldots (1, 1)$ we have:

$$P(A) = \frac{1}{2a}; P(B) = 1 - \frac{1}{2a}$$

$$P(A \mid x) = 1 - ax; P(A \mid y) = y/a$$

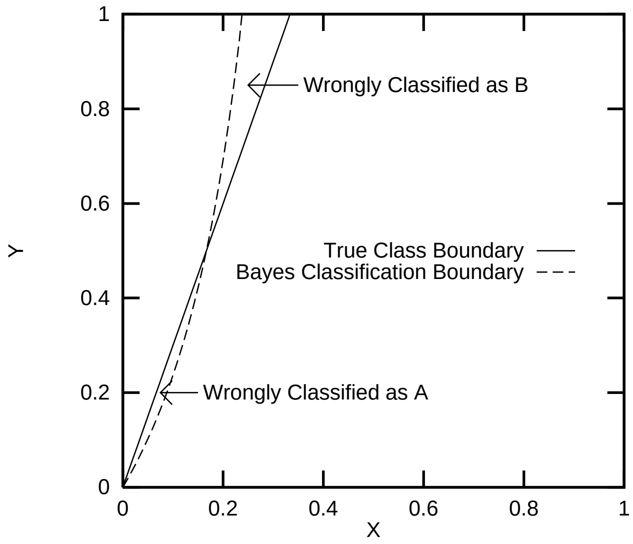

Figure 7: Bayes fails for the separator $y = 3x$.

$$P(B \mid x) = ax; P(B \mid y) = 1 - y/a$$

Substituting into equation 4 tells us that an example will be classified as A when

$$y > \frac{a^2 x}{2a + 2ax - 2a^2 x - 1} \tag{5}$$

For $a = 1$, this gives just $y > x$; i.e., the Bayesian classifier will converge to 100% accuracy (which is consistent with the experimental results in Section 4.1). However, if $y = 3x$ (for example), the condition for being classified as A is

$$y > \frac{9x}{5 - 12x}$$

This function is plotted in Figure 7 along with the original function $y = 3x$. It shows clearly where the Bayesian classifier will make mistakes.

Piecewise integration over the difference between this function and the function $y = 3x$ allows us to predict a convergence for Bayes of 97.1% in the limit (when all examples in the lattice have been seen). This value has been verified experimentally. A similar analysis allows one to predict the 4.6% error rate generated by the discriminator $y = x^2$ considered in Section 4.1.

5.2 MVDM LEARNING FUNCTIONS

Given the example for $y = ax$, we would like to see if PEBLS has similar difficulty learning this kind of concept class. Somewhat surprisingly, we can show that PEBLS can actually learn any concept on Z^2 provided the concept is separated by a function. We recall again that our algorithms treat all inputs as arbitrary symbols in a feature space, and thus have no knowledge of the underlying Euclidean space.

Theorem 1 *Let G be a finite $N \times N$ grid. Let A be a concept that is a subset of G, and assume A is separated from B (the complement of A) by a function f. f is a function from $1 \ldots N$ to $1 \ldots N$. That is, (x, y) is in A iff $f(x) \geq y$ and (x, y) is in B iff $f(x) < y$. Then, the MVDM classifier will achieve 100% accuracy on A and B at the limit.*

Proof: The proof is by contradiction. Assume (x_1, y_1) is a point in A that PEBLS classifies incorrectly. Since the distance from (x_1, y_1) to itself is zero, this implies there is a point (x_2, y_2) in B whose distance to (x_1, y_1) is also zero. From the definition of MVDM, this means that

$$|P(A|x_1) - P(A|x_2)| = 0$$

$$|P(A|y_1) - P(A|y_2)| = 0$$

Since $P(A|x_i) = f(x_i)$, we have $f(x_2) = f(x_1)$. That is,

$$y_2 > f(x_2) = f(x_1) \geq y_1$$

Therefore, $y_2 > y_1$. Let A_1 be the set of all x such that $f(x) \geq y_1$. Similarly, let A_2 be the set of all x such that $f(x) \geq y_2$. If x is in A_2, $f(x) \geq y_2$. Then since $y_2 > y_1$, $f(x) > y_1$ and therefore x is in A_1. Thus A_2 is a subset of A_1. However, $P(A|y_1)$ is the size of A_1 and $P(A|y_2)$ is the size of A_2; therefore, since $P(A|x_1) = P(A|x_2)$, A_1 and A_2 are the same size. So we have two sets in Z^2, one of which is contained in the other and the sizes are the same; thus they must be equal. However, x_1 is a member of A_1, but it cannot be a member of A_2 because $y_2 > f(x_1) \geq y_1$. □

This theorem basically shows that at the limit, MVDM can learn any concept in Z^2 separated by a function. A similar theorem can be shown for concept classes on multidimensional grids. Clearly, in practice the classifier rarely sees every example. Nevertheless, this is an important observation in view of the limitations of a simple Bayes classifier to deal with even linearly separable concepts at the limit. The natural question to ask is what is the convergence rate of MVDM to the limit for this class of concepts. This is considered in the full version of this paper.

5.3 BAYES LEARNING IMPLIES MVDM LEARNING

We showed in the previous section that it is possible that both Bayes and MVDM will yield very poor accuracy prediction even at the limit. Therefore, we decided to investigate whether there are any distributions for which the Bayesian classifier obtains 100% accuracy while PEBLS does not. To simplify the discussion, let's consider the situation at the limit, that is when all the examples have been seen.

It turns out that, for *any* finite domain, PEBLS will *always* achieve 100% accuracy if Bayes is known achieve to 100% accuracy. The proof is straightforward and is given below.

Theorem 2 *Given a finite domain for which the Bayesian classification rule achieves 100% accuracy at the limit, the* PEBLS *program using MVDM will also achieve 100% accuracy at the limit.*

Proof: In the limit, a classifier will see every possible example in a domain. Since we know that Bayes classifies every point correctly, we know that for any point $p_1 = (x_1, y_1)$ in class A,

$$\frac{P(A|x_1)P(A|y_1)}{P(A)} > \frac{P(B|x_1)P(B|y_1)}{P(B)}$$

This must be true in order for Bayes to classify the point correctly as A.

Now consider how PEBLS will classify the point. Since it has seen the point before (in the limit), it will always find the point itself as its own nearest neighbor, since the distance from a point to itself is always 0 using MVDM. Thus it will classify it correctly unless there is another point $p_2 = (x_2, y_2)$ that is labelled B that has distance 0 from p_1. As explained earlier, the distance between p_1 and p_2 as computed by the MVDM is the following:

$$|P(A|x_1) - P(A|x_2)| + |P(B|x_1) - P(B|x_2)| +$$
$$|P(A|y_1) - P(A|y_2)| + |P(B|y_1) - P(B|y_2)|$$

Assuming that such a point exists, then this distance is zero. Therefore all four terms must be zero, and thus $P(A|x_1) = P(A|x_2)$, $P(B|x_1) = P(B|x_2)$, etc. Now, since p_2 is in class B, then

$$\frac{P(A|x_2)P(A|y_2)}{P(A)} < \frac{P(B|x_2)P(B|y_2)}{P(B)}$$

because the Bayesian classifier classifies it correctly. But since $P(A|x_1) = P(A|x_2)$ etc., we can substitute x_1 and y_1 for x_2 and y_2 in this equation, which gives us a contradiction. Therefore no point in class B will have distance 0 from any point in class A. □

6 CONCLUSION

Despite their computational similarity, PEBLS and the Bayesian classifier display some rather striking differences in their overall classification accuracy when applied to symbolic domains. In particular, we demonstrated that PEBLS outperforms Bayes under a wide range of conditions, including cases in which the instances are both dense and sparse with respect to the feature space, and in both simple and complex scientific domains. In the limit when instances completely fill the space, we have shown that if Bayes can learn the distribution perfectly, then PEBLS will always learn perfectly. Indeed, while Bayes cannot perform perfectly for some simple class distributions, PEBLS can achieve perfect accuracy on a wide class of distributions. These results support the view that MVDM is a good similarity measure for adapting memory-based methods to symbolic domains.

Acknowledgements

This material is based upon work supported by the National Science foundation under Grants No. IRI-9223591 and IRI-9220960.

References

Aha, D. W., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37–66.

Albert, M. & Aha, D. W. (1991). Analyses of instance-based learning algorithms. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pp. 553–558.

Albrecht, R. & Werner, W. (1968). Error analysis of a statistical decision method. *IEEE Transactions on Information Theory*, *10*, 34–38.

Aoki, M. (1965). On some convergence questions in bayesian optimization problems. *IEEE Trans. on Automatic Control*, *10*, 180–182.

Chandrasekan, B. (1971). Independence of measurements and the mean recognition accuracy. *IEEE Transactions on Information Theory*, *17*, 452–456.

Chandrasekan, B. & Harley, T. (1969). Comments on the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, *15*, 421–423.

Clark, P. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–284.

Cost, S. & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, *10*(1), 57–78.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27.

Delcher, A. L., Kasif, S., Goldberg, H. R., & Hsu, B. (1993). Probabilistic prediction of protein secondary structure using causal networks. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pp. 316–321.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis*. John-Wiley.

Haussler, D., Krogh, A., Mian, S., & Sjolander, K. (1992). Protein modeling using hidden markov models. Tech. rep. UCSC-CRL-92-23, Computer and Information Sciences, University of California, Santa Cruz.

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, *14*, 55–63.

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

Langley, P. & Iba, W. (1992). An analysis of bayesian classifiers. In *Proceedings of the Tenth National Converence on Artificial Intelligence*, pp. 223–228.

Langley, P. & Iba, W. (1993). Average-case analysis of a nearest neighbor algorithm. In *Proceedings of IJCAI-93*, pp. 889–894 Chambery, France.

Michie, D. & Attar, A. (1991). Use of sequential bayes with class probability trees. *Machine Intelligence*, *12*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pp. 257–286.

Stanfill, C. (1987). Memory-based reasoning applied to English pronunciation. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pp. 577–581.

Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, *29*(12), 1213–1228.

Yi, T.-M. & Lander, E. (1993). Protein secondary structure prediction using nearest-neighbor methods. *Journal of Molecular Biology*, *232*, 1117–1129.

Zhang, X., Mesirov, J., & Waltz, D. (1992). A hybrid system for protein secondary structure prediction. *Journal of Molecular Biology*, *225*, 1049–1063.