Results of a New Classification Algorithm Combining *K* Nearest Neighbors and Recursive Partitioning

David W. Miller†

Bioreason, Inc., 150 Washington Avenue, Suite 303, Santa Fe, New Mexico 87501

Received July 24, 2000

We present results of a new computational learning algorithm combining favorable elements of two well-known techniques: *K* nearest neighbors and recursive partitioning. Like *K* nearest neighbors, the method provides an independent prediction for each test sample under consideration, while like recursive partitioning, it incorporates an automatic selection of important input variables for model construction. The new method is applied to the problem of correctly classifying a set of chemical data samples designated as being either active or inactive in a biological screen. Training is performed at varying levels of intrinsic model complexity, and classification performance is compared to that of both *K* nearest neighbor and recursive partitioning models trained using the identical protocol. We find that the cross-validated performance of the new method outperforms both of these standard techniques over a considerable range of user parameters. We discuss advantages and drawbacks of the new method, with particular emphasis on its parameter robustness, required training time, and performance with respect to chemical structural class.

INTRODUCTION

Modern chemical lead discovery and optimization efforts have received significant benefits from the use of computational learning methods. Such methods seek to automate the estimation of a known response variable (e.g., inhibition of a biological target) using one or more explanatory variables (molecular weight, pharmacophores, etc.), and have an obvious applicability to a discipline with such inherently complex and often multiple underlying physical mechanisms and increasing numbers of data samples. The utility of learning methods in molecular design can be generally divided into two categories. The first is prediction, where a set of untested molecules is screened with a model to estimate an unknown physical or biological property, often with the intent of selecting an ideal subset of them for future synthesis and/or biological screening. The second goal is interpretation, where a model is used to provide a physical explanation for a complex observed phenomenon in a set of data, perhaps to generate ideas for subsequent steps in a chemical design. In either case the modeled response variable historically has reflected a variety of properties, with recent examples being target inhibition,[1−3] membrane permeability,[4−6] toxicity,[7−9,16] and similarity to known drugs,[10,11] among others. The variety of model types is equally extensive, including the familiar linear models used in traditional QSAR equations,[12,13] the popular CoMFA models,[14] and nonlinear types of methods such as neural networks,[2,3,15] expert systems,[7−9,16,17] recursive partitioning,[1,18] kernel methods,[19,20] etc. Summaries of the use of such methods can be found in the literature.[15,21]

While the number of successful applications of computation learning methods in chemistry continues to grow, there is a continued need for improved methodologies. Predicting properties of small molecules using only structure or physical descriptors is a challenging problem, involving incomplete and often noisy data, as well as intrinsically complex physical mechanisms requiring a high-dimensional representation. Identifying the methods best suited for a particular prediction problem requires that new and more powerful techniques be explored, since even small performance improvements can be significant in the context of identifying leads from a large data set. In this paper we examine an adaptation of a recently described learning approach[22] that combines features from two distinct methods widely used within the drug-discovery arena: recursive partitioning and *K* nearest neighbors. Below we give a summary of these two well-known methods and a description of an approach combining favorable elements from each. We then present results of this hybrid method applied to a set of biological screening data, and show that its predictive performance is greater than that of both standard methods.

BACKGROUND

***K* Nearest Neighbors and Recursive Partitioning.** The type of learning methods we are concerned with perform the tasks of either regression or classification; that is, they attempt to describe a single response property (biological activity, for example) as a function of one or more explanatory variables (the terms "variables", "features", and "descriptors" are used interchangeably). The different learning methods differ mainly in the type of function used to construct the relationship between variables. In the *K* nearest neighbor (KNN) method,[23] property predictions are based on an average over the *K* samples nearest the test sample requiring the prediction. The measure of nearness in this approach is generally taken to be the distance (e.g., Euclidean) between samples in the feature space defined by the explanatory variables. The method typically uses a fixed decision rule so that, in the case of the classification problem for example, the test sample is predicted to belong to that

† Phone: (505) 995-8188, ext 211. Fax: (505) 995-8186. E-mail: dmiller@bioreason.com.

10.1021/ci0003348 CCC: $20.00
Published on Web 11/01/2000

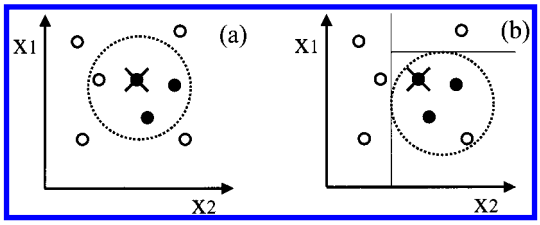

Figure 1. Decision regions for KNN and RP models, respectively. The figures represent samples in a 2-dimensional feature space, with each sample belonging to one of two classes (indicated by white and black circles). In KNN models (a), the decision region is defined using distances between samples, and consists in this case of the three neighbors closest to the test sample (the latter is indicated by crossed lines). In RP models (b), the decision region is created through a partitioning of the input-variable space (the figure shows two such partitions, one using each of the two input variables). The figure illustrates that with KNN models the test sample has a greater likelihood of being positioned near the center of the decision region.

class of which the majority of neighbor samples are representatives. Since there are no trained parameters other than the number of neighbor samples, *K*, this method is relatively fast, and is straightforward to implement. Despite its simplicity, it has been found to outperform many other flexible nonlinear methods, particularly when the number of explanatory variables is high.[24] This observation, and the fact that this method applies a widely followed (albeit oversimplified) principle of chemistry that similar structures often exhibit similar properties, may explain the frequent and often successful use of this technique in drug-discovery applications.

Recursive partitioning[25−27] (RP) refers to a particular class of the decision-tree method where the tree is grown using a greedy iterative procedure. The method is similar to KNN in that the predicted response property for each test sample is made using a fixed portion of the total data set. The main difference with RP methods is that these local neighborhoods are constructed not for each sample independently as in KNN but rather as a one-time, disjoint partitioning of the entire input feature space. Partitioning is performed using an optimization protocol designed to produce nearly pure regions, which in the classification problem are those in which the majority of member data samples have an identical class assignment. As with KNN, the classification decision rule in RP is to assign a test sample to the class most representative of its assigned region in sample space. The RP method is also fast and easy to implement, and has been applied to a variety of chemical design problems.

There is an interesting characteristic of the KNN and RP methods in that each has a fundamental strength that counterbalances a relative weakness in the other. In the case of KNN, a primary strength is that each test sample has a tendency to be located close to the center of the space defining its local neighborhood. This property, resulting from each test sample defining its own unique decision region, would suggest that each sample is well represented by its neighbors and is therefore more likely to be correctly classified by the resulting model. In contrast, the RP method constructs the decision regions in a global fashion, and as a result there are likely to be many data samples near the region boundaries. It can be argued that such a construction causes some samples to be poorly represented by their local regions and therefore classified with less accuracy. This distinction is illustrated graphically in Figure 1. Analogously, a primary strength of the RP method is its ability to perform intrinsic feature selection. The optimization procedure iteratively divides the input space using the single explanatory variable that best separates a given set of data samples into pure subsets. As a result, the final model tends to include only those variables important for providing a good classification, and excludes those that are found to be less relevant. In contrast, KNN implementations tend to use a more rigid procedure in which all input variables are equally represented in constructing the decision regions. In cases where many input variables are not important in defining the classes, such a procedure may result in suboptimal classification performance.

A Hybrid Method. The complementary nature of the KNN and RP methods suggests that a hybrid method combining the best elements of each may provide an improved performance. Recently Friedman[22] proposed such a method, which he calls "flexible metric nearest neighbor classification" (FMNN). The basic protocol for FMNN is to construct a local decision region for each test sample, as in KNN, but using a measure of distance that is locally adapted (via a recursive procedure) to favor the most significant input variables. In this way the method is able to combine favorable elements from both the KNN and RP procedures. As in KNN, each test sample tends to be centrally located in the decision region, while like RP, the method performs implicit feature selection by favoring input variables estimated to achieve the best classification performance.

A detailed description of Friedman's FMNN implementation is provided in the Appendix. Here we briefly outline the training protocol for the most general of his proposed techniques.

For each test sample, the following steps are taken.

(1) Initialize a neighbor list to include all training samples.

(2) Calculate a significance metric for each input variable. This metric measures the purity (with respect to class) of a small number of closest samples along each individual input variable. The number of samples used is user specified.

(3) Use these significance estimates to compute a revised distance between the test sample and all training samples in the current neighbor list.

(4) Update the neighbor list by removing those samples with the largest (revised) distance from the test. The number of samples removed is determined by a user-specified learning rate.

(5) Repeat steps 2−4 until only *K* training samples remain in the neighbor list, and make a response prediction for the test sample using the standard decision procedure (e.g., as in KNN or RP).

Our implementation of the FMNN algorithm includes two significant modifications. The first is a flexible measure of variable significance to handle situations in which either the prior class probabilities are highly skewed or the costs of misclassification are unequal. Such a modification is particularly well suited to property-prediction problems in chemistry, which tend to rely on data sets having many more inactive data samples than active ones. The second is an early stopping procedure to allow a response prediction to be made when the default neighborhood region is nearly pure. These changes are discussed in detail in the Appendix. The modified method is referred to in this paper as MFMNN (modified FMNN) to distinguish it from the original algo-

rithm. All of the studies reported here were made using this modified version, though we have verified that both FMNN and MFMNN give similar results in terms of predictive performance.

It is useful to briefly contrast the MFMNN algorithm with other proposed flexible adaptations of KNN. The machine learning literature contains numerous examples of KNN algorithms intended to perform some form of automatic feature selection. In these "adaptive-kernel" methods the distance metric used to locate neighbors employs flexible weights to modulate the contribution of each individual input variable. Thus, the distance between two samples can generally be written as

$$d_q(\vec{x}_1, \vec{x}_2) = \left(\sum_{i=1}^{p} |[W(\vec{x})(\vec{x}_1 - \vec{x}_2)]_i|^q \right)^{-q} \quad (1)$$

where $\vec{x}_1$ and $\vec{x}_2$ are the sample locations (in p dimensions), $W(\vec{x})$ is a vector of flexible weighting parameters, and q defines the specific distance metric ($q = 2$ is the Euclidean). Implementations typically follow one of two routes. The first employs "explicit" feature selection, in which some subset of the input variables is simply excluded from distance calculations, making $W(\vec{x})$ a vector of binary values.[20,28] The second common implementation seeks a more general (nonbinary) form of the variable-weighting vector $W(\vec{x})$.[29−31] In both cases the weights are typically chosen via numerical optimization techniques with the goal of maximizing classification performance, and parameter estimation can be performed either globally for all test samples or locally for each individually.

While similar to the MFMNN method in pursuing some form of feature selection, most derivative KNN methods suffer from two significant drawbacks. First, since estimates of feature significance are made using optimization, the methods require multiple training iterations, and thus can be extremely slow when the number of samples and/or input variables is large. In contrast, MFMNN uses statistical sampling to estimate feature significance, so repeated training cycles are not required. Second, because the KNN decision rule is a discontinuous function, training of most adaptive methods requires combinatorial optimization procedures, which cause the additional difficulties associated with searching inefficiency. In addition to these two disadvantages, there is a third specific to explicit feature-selection methods in that variable significance estimates are limited to values of zero or one, while optimal values may be intermediate. For a more general discussion of feature-selection methods in machine learning, see the excellent review by Blum and Langley.[32]

EXPERIMENTAL RESULTS

Data Set. Biological screening data were obtained from NCI.[33] The data are from a cell-based assay measuring protection from HIV-1 infection, and results are categorized as confirmed active (CA), confirmed moderately active (CM), or confirmed inactive (CI). Of 32 110 total compounds obtained the numbers in each group were 230, 444, and 31 436, respectively. The size of the data set was reduced by the removal of compounds with unusual atom types (Bi, Cd, Ge, etc.) and large molecular weights (>500), and the remaining CA- and CM-class compounds were then combined into one active class, yielding a total data set of 453 active and 26 803 inactive compounds. In an effort to reduce the time required for training the computational models, we elected to use only a subset of the inactive data set. Roughly 25% (6700) of the inactive samples were randomly chosen and combined with the 453 active samples to yield a total of 7153 compounds subsequently used for model building. Such a reduction provides a modest improvement in training time without adversely affecting model performance: our results have shown that even as little as 5% of the inactive data yield similar results with all three of the learning methods described here.

Molecular descriptors were derived from literature descriptions of the MDL MACCS keys,[34] a public key set based on small molecular fragments. A subset of 157 keys (of 166 total) was chosen for the representations, such that for each compound a 157-element integer vector is created which represents the number of times the compound is "hit" by each structural key. To reduce the high redundancy in the descriptor representation, the data matrix (samples/features) was reduced to 18 dimensions using principal components analysis (PCA), with the final dimensionality chosen so that approximately 95% of the sample variance was explained. The PCA procedure is commonly used in conjunction with learning methods in chemistry,[35−37] and results in a significant reduction in the time required to train models. Following PCA, the data were scaled to have zero mean and unit variance.

Experimental Protocol for Method Comparisons. Using the 453 active and 6700 inactive data samples from NCI, we have trained two-class classification models using KNN, RP, and MFMNN, with the goal of comparing their predictive performances. It is well chronicled in the machine learning literature that such comparisons of predictive learning methods can be difficult, and it is not always possible to draw unambiguous conclusions. Several potential complicating factors include a limited number of available data samples, a data set inherently biased in favor of a particular type of method, and a user with a greater expertise in one method over another. Because of these limitations, it is important that careful experimental design be applied if even qualitative judgments are to be made from the comparison. Our protocol is based on two primary objectives. First, it is desired that the method comparison be based on models trained independently to achieve optimal performance. Second, it is preferable that the (optimal) models be selected on the basis of *future* predictive performance, necessitating a resampling procedure to estimate the expected behavior of each model for unseen data. On the basis of these objectives, the training procedure given here was implemented.

For each type of learning method the following steps were taken.

(1) Initialize the model to have the highest desired level of "flexibility". For both KNN and MFMNN, model flexibility is parametrized by K, the number of neighbors used to make response-variable predictions. In both cases a small value of K means greater flexibility. For RP the flexibility is determined by the amount of pruning (via the optimal procedure proposed by Breiman et al.[26]) applied to the default decision tree, which is ordinarily grown until misclassification errors are minimal.

(2) Use *K*-fold cross-validation to train and test a model. The cross-validated classification performance is an estimate of future predictive ability for the specified level of model flexibility. In our experiments we use $K = 4$.

(3) Incrementally decrease the model flexibility and repeat step 2. This procedure is continued until the flexibility value has reached the desired lower limit.

(4) Select the optimal flexibility according to the model with the best cross-validated performance. This performance value can be used as a comparison with other models trained using the same procedure (including the identical random seed for cross-validation).

A few points regarding this protocol should be noted. Given an unlimited amount of data, it would be desirable to initially hold out a reasonable fraction (say, 20%) of the samples and apply the above procedure to only the remaining portion. Such a protocol allows the optimal models (with respect to flexibility) found in step 4 to be applied to completely unseen data, giving a more robust estimate of the predictive difference between models. In the present case we decided that the small number of available data samples (particularly active ones) made further partitioning undesirable. While our abbreviated approach appears nonetheless to have yielded informative results, a more rigorous protocol would clearly be helpful in drawing unambiguous and quantitative conclusions. Further discussion of model-selection and model-comparison techniques can be found in the literature.[38]

Training Results of KNN, RP, and MFMNN Models. Figure 2 shows the cross-validated classification performance for KNN, RP, and MFMNN for several values of the model flexibility (tree size for RP, and number of neighbors, *K*, for KNN and MFMNN). The *y* axis shows the enrichment ratio, defined as the fraction of compounds predicted to be active that are actually active, divided by the fraction of compounds in the total data set that are active. The *x* axis shows the fraction of active compounds correctly classified (often called coverage). These two quantities can be expressed as follows:

$$\text{enrichment ratio} = \frac{NN_{TP}}{N_A(N_{TP} + N_{FP})} \quad (2)$$

$$\text{coverage} = N_{TP}/N_A \quad (3)$$

where N is the total number of compounds, N_A is the total number of active compounds, and N_{TP} and N_{FP} are the number of true-positive and false-positive predictions, respectively (actives correctly classified versus inactives incorrectly classified). It is ordinarily desirable that both the enrichment ratio and the coverage be maximized, and the inverse relationship observed between the variables with all three methods makes it difficult to identify a performance that can be called optimal. However, the figure suggests that MFMNN measurably outperforms both KNN and RP over a considerable range of parameter space: for a given value of active coverage, MFMNN models show consistently superior discrimination between active and inactive samples. The particularly poor performance of the RP model was somewhat surprising, and may be due in part to the large disparity between the numbers of active and inactive samples in the data set. This is discussed at length in a later section.

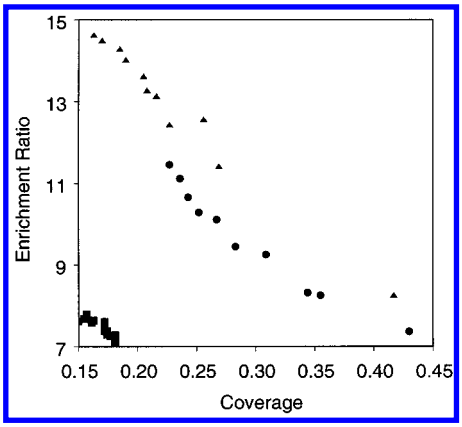

Figure 2. Performance of MFMNN (triangles), KNN (circles), and RP (squares) methods using chemical screening data. The *y* axis corresponds to the cross-validated enrichment ratio (eq 2), and the *x* axis to the fraction of active compounds correctly classified (eq 3). Data points correspond to different levels of model flexibility (numbers of nearest neighbors, *K*, chosen as even values ranging from 2 to 20, or (for RP) the level of decision tree pruning). The RP trees used the Gini purity metric for splitting, and MFMNN parameter values were as follows: $\alpha = 0.5$; $\beta = 1.0$; $L = 6$; maxpurity = 0.32; $C_{FN} = 1.0$. See the Appendix for a description of model parameters.

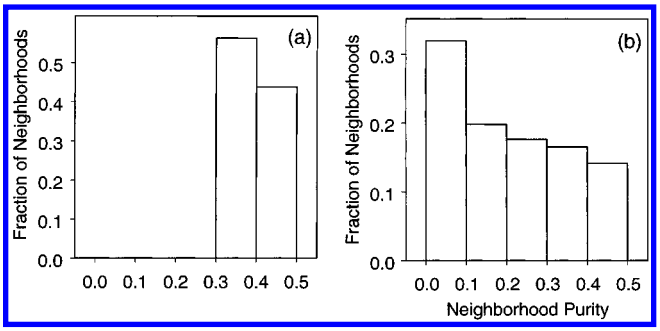

Figure 3. Purity of local neighborhoods with KNN (a) versus MFMNN (b) models. For every sample in the cross-validated training process the Gini metric of eqs 7 and 8 was used to calculate the purity of the final decision region. Here $K = 20$, and MFMNN parameters are the same as in Figure 2. As expected, the MFMNN procedure produces regions of greater purity (smaller Gini values) than KNN.

One unexpected result is the tendency of MFMNN models to exhibit greater enrichment and less coverage than KNN models for identical values of *K*. This behavior, evidenced by the larger number of MFMNN data points at lower coverage values, would suggest that the increased performance of MFMNN arises from the elimination of false positives rather than of false negatives.

The influence of the MFMNN method on neighborhood purity is illustrated in Figure 3. The figure shows that, for a fixed neighborhood size, the average purity of training regions increases substantially for MFMNN versus KNN. This result is expected, since the recursive procedure in MFMNN seeks to maximize purity directly. That the MFMNN models exhibit improved performance indicates that this increase in purity is accompanied by a decrease in misclassification errors, resulting in greater enrichment. The following section elaborates this result from the standpoint of selecting the most relevant input variables.

Results of varying the MFMNN model parameters are shown in Table 1. Values of enrichment and coverage are

Table 1. MFMNN Performance versus User Parameters[a]

trial	β	L	C_{FN}	coverage	enrichment
1	1.0	6	1.0	0.26	12.5
2	1.0	6	2.0	0.31	9.8
3	1.0	6	8.0	0.45	7.2
4	1.0	10	1.0	0.26	12.5
5	1.0	10	2.0	0.31	9.8
6	1.0	10	8.0	0.45	7.2
7	1.0	15	1.0	0.26	12.5
8	1.0	15	2.0	0.31	9.8
9	1.0	15	8.0	0.45	7.2
10	0.5	6	1.0	0.30	11.4
11	2.0	6	1.0	0.23	12.4
12	8.0	6	1.0	0.23	12.5

[a] Cross-validated enrichment and coverage values are shown for varying β, L, and C_{FN}. Values of α and maxpurity are fixed to be 0.5 and 0.32, respectively.

shown for varying β, L, and C_{FN} using the fixed value $K = 4$. Although parameters are not tested exhaustively, three principal conclusions are drawn from the table. First, model performance is relatively insensitive to the choice of the "significance window", L. Second, increasing C_{FN} causes a distinct increase in coverage and decrease in enrichment but produces relatively little change in absolute performance relative to those in Figure 2. This result is discussed in more detail below. Third, increasing β tends to cause a decrease in coverage and increase in enrichment though with somewhat inferior overall performance compared to those in Figure 2, particularly in cases where K is reasonably large (not shown). This behavior of the β parameter, which should reveal information about the number of irrelevant input features, is discussed more fully in the next section.

Figure 4 summarizes the performance of MFMNN from the standpoint of chemical structural class. The six templates depicted in the figure are highly representative of the active data set, with 174 (38%) of 453 active training compounds falling into exactly one class. Under each of the classes in Figure 4 is shown the number of active and inactive samples for the class, the number of true and false positives, and the class-specific enrichment ratio. The latter three values are based on the MFMNN model in Figure 2 corresponding to 42% coverage. Judgments regarding the quality of the class-based prediction values are deferred to the following section. However, two preliminary observations may be made from the data in the figure. First, the overall coverage of active compounds is significantly higher (74%) within the six chemical classes shown than it is for the full set of data (42%). This result is not surprising given the use of a neighbor-based method such as MFMNN: such a method can be expected to correctly predict the most active samples when these actives cluster together into groups, and this tends to be more likely within focused chemical classes. The second observation is that the combined enrichment ratio for the set of six classes (1.6) is significantly lower than that of the full set of data (8.1). While this difference is somewhat exaggerated by the large disparity in the number of inactive samples in the two cases, the performance within the six classes is clearly inferior to the all-data results, as can be seen by comparing the ratio of true-positive to false-positive rates for the two examples (a metric less sensitive to sample sizes). That the class-based ratio (3.6) is much smaller than the all-data ratio (15.5) indicates that the

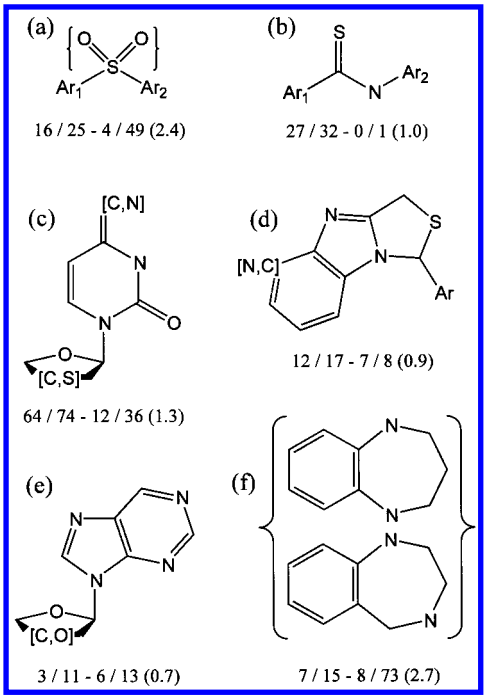

Figure 4. Results of MFMNN from the standpoint of chemical structure. Six major chemical classes are shown along with corresponding training populations and prediction results. Values under each structure correspond to the following training numbers: true positives/actives:false positives/inactives (enrichment ratio). Model parameters are the same as in Figure 2, with $K = 2$. The notation "Ar" corresponds to an aromatic ring.

predictive performance within the classes is significantly weakened. This discrepancy in class-based versus all-data prediction performance can have significant consequences, particularly for model interpretation and virtual screening results, and is discussed at length in the following section.

DISCUSSION

Results of this study indicate a measurable improvement in performance with MFMNN relative to KNN and RP models. Taking into account the difficulty in identifying an optimal model using enrichment ratio as a cost function, Figure 2 nevertheless indicates that MFMNN exhibits larger enrichment than KNN and RP for equivalent values of coverage, even if the effects of neighborhood size differ somewhat for the KNN and MFMNN methods.

Another significant finding was that although the number of parameters in MFMNN is relatively large compared to those in KNN and RP, model performance is nevertheless fairly robust. Varying the values β, L, and C_{FN} tended to show small and consistent changes in predictive performance. This parameter robustness is important both so that nonexperts can choose appropriate values in a reasonable amount of time and to provide confidence that the cross-validated performance will closely approximate performance on future data. Since several MFMNN parameters are chosen outside of the cross-validation procedure, extreme sensitivity to their values might suggest that cross-validated performance metrics might be artificially inflated. Results of Table 1 provide some assurance that MFMNN parameter selection might be reasonably straightforward.

The training time for MFMNN models was significantly longer than for KNN and RP models, but is still fast in comparison with other methods. To train a model using the parameters from Figure 2, with $K = 20$, required 257 s on a Pentium III 450 MHz processor. The equivalent KNN model required 66 s, while RP required 50 s (all three methods implemented in C++).

Despite the favorable behavior of MFMNN with respect to parameter selection and training time, the improvement in performance over KNN is relatively small. The choice of which algorithm to use may therefore be somewhat complicated, and may depend on factors other than the small number of benchmarks evaluated in this study. A more pertinent question may be whether the absolute performance of MFMNN, as presented in Figures 2 and 4, is considered sufficiently predictive as to be useful in a real drug-discovery application. The structural class-based results should provide the most reliable assessment of this question, since the full data set is so diverse that one cannot easily assess whether new (virtual screening) compounds fall within its chemistry space. From the class-based results of Figure 4, the sulfide/sulfone (a) and benzodiazepine (f) classes show the best performance, with enrichment ratios of 2.4 and 2.7, respectively. Likewise, for these two classes the true-positive prediction rates exceed the false-positive rates by factors of roughly 8 and 4. This level of predictive performance could be considered adequate for some virtual screening purposes. Of the four remaining structural classes, the pyrimidine (c), purine (e), and tricycle (d) classes show generally poor active/inactive discrimination, while the thioamide (b) class contains too few inactive representatives to draw reliable conclusions. The variation in class-based results, though not explained in the figure, was largely due to the presence or absence of distinct subclusters that further distinguished actives from inactives in each structural class. Such a phenomenon, clearly observed in classes a and f, greatly simplified the classification task. The most striking point when evaluating overall MFMNN performance is the distinction between the all-data results and the structural class-based results. That the class-based performance is uniformly inferior in our study supports the view that, within focused chemical classes, structural differences between active and inactive samples become increasingly subtle, making discrimination more difficult. As a consequence, models trained on diverse data can, when applied to a specific structural family, yield prediction results that differ substantially from, and are likely inferior to, those obtained in the context of the full data set. This is particularly true for models that employ simple nearest-neighbor principles and general 2D descriptors, as is true in the present study. Such a phenomenon can have important ramifications, particularly regarding the selection of a data set for virtual screening. These virtual sets will in many cases be precisely the type of focused chemical class depicted in Figure 4, both for the practical reason that virtual screening results are often intended to be the basis for a synthetic effort in the lab and because with a focused class it is easier to have confidence that the test samples fall within the chemistry space of the training set. While an ideal approach may therefore be to develop models trained only on highly homogeneous data, and to apply them only to compounds from the same structural class, our results suggest such an approach might yield only weakly predictive models unless one employs methodologies sufficiently more sophisticated than those used in the present study.

An important point with respect to model performance is that since we have used only a single set of data, our results may reflect characteristics of our experimental design rather than of the learning method itself. One particularly significant aspect of our study was the use of PCA to form the final set of descriptors for training. Since the most obvious potential benefit of MFMNN is the exclusion of irrelevant input variables, performance is likely best in cases where several variables are unimportant. In contrast, the use of PCA is likely to select variables that are highly relevant, creating a setting less well suited for MFMNN. The observation that increasing β worsens performance is evidence that the number of high-relevance features is likely to be large (see the Appendix). Friedman[22] provides additional discussion of how data-set and input-variable selection affects model performance; our results are within the range reported by that author using a wider variety of training conditions.

An interesting result of the current study is the performance of MFMNN with respect to the variable misclassification cost C_{FN}. This parameter allows the user to assign a higher cost to false-negative misclassifications, thus controlling the tendency for underrepresented samples (e.g., active molecules in the current study) to have a higher misclassification rate (see Breiman et al.[26] for a more general discussion). It is perhaps surprising that using equal costs ($C_{FN} = 1$) can yield a model with a large level of active coverage (more than 40% in Figure 2), given that the ratio of inactives to actives in the training set is nearly 15:1. This is likely a result of the tendency for molecules of similar structure (and activity) to form clusters, where the local ratio of class populations may be significantly different from, and perhaps nearly independent of, the global ratio. This clustering property is likely to vary with the particular training set and input descriptors. Table 1 suggests that the C_{FN} parameter may provide a useful mechanism for obtaining an acceptable level of active coverage for an arbitrary data set. It should be noted that the poor performance of standard RP in Figure 2 can in part be attributed to the large imbalance in active and inactive samples. The use of an increased cost for false-negative errors (analogous to $C_{FN} > 1$ in MFMNN) did lead to a modest improvement in prediction performance for RP, though results were still inferior even to those of the standard KNN method (not shown).

It is interesting to compare our results with those of another study performed with a different method on the same data set. Klopman and Tu[17] report classification results on NCI data using an expert system based on molecular fragments. The authors report an 82% prediction rate on a holdout data set using their trained model. Analogous rates can be determined for the MFMNN performance of Figure 2, using eqs 2 and 3 and the fact that the classification rate is given by $(N_{TP} + N_{TN})/N$, where N_{TN} corresponds to true-negative predictions. The resulting MFMNN values from Figure 2 are 94−95%. However, the significant differences in the two studies with respect to choice of training samples, input features, and relative class proportions preclude a direct comparison of the methods. Plans for future work include the testing of additional model types, as well as different sets of data and descriptors.

ACKNOWLEDGMENT

I thank Susan Bassett, Terry Brunck, and the reviewers for their many useful comments and suggestions.

APPENDIX

Implementation Details. The most general of Friedman's FMNN methods uses a modified distance metric derived using estimates of significance for each input variable. The modified distance between a test sample, $\vec{z}$, and a training sample, $\vec{x}_n$ has the form

$$d_q(\vec{z},\vec{x}_n) = \left(\sum_{i=1}^{p} |[W(\vec{z})(\vec{z} - \vec{x}_n)]_i|^q\right)^{-q} \qquad (4)$$

where $W(\vec{z})$ provides a weighting term for each of the p input features, and q indicates the particular distance metric ($q = 2$ indicates Euclidean).

The significance metric takes the form

$$W_i(\vec{z}) = \left(I_i^2(z_i)\Big/\sum_{k=1}^{p} I_k^2(z_k)\right)^{\beta/2} \qquad i = 1, \ldots p \qquad (5)$$

Here, $I_i^2(z_i)$ measures the importance of the ith input feature at the point $x_i = z_i$ and the exponential factor β modulates the overall influence of the weighting terms $W(\vec{z})$. Friedman suggests that, for classification problems with two classes, $I_i^2(z_i)$ should measure the purity (with respect to class) of a sample taken around the test location:

$$I_i^2(z_i) = \sum_{j=1}^{2} p(j|x_i = z_i) - 1/2 \qquad (6)$$

where $p(j|x_i = z_i)$ is the conditional probability density for class j, given that the position $x_i = z_i$ is known. As may be expected, there is a close relationship between this significance measure and the Gini purity measure common to RP classification:

$$G(R) = 1 - \sum_{j=1}^{2} p(j|\vec{x} \in R) \qquad (7)$$

where R defines a particular region of sample space. If this region is defined as $x_i = z_i$, the relationship between $I_i^2(z_i)$ and $G(R)$ is such that $I_i^2(z_i) = 1/2 - G_i(z_i)$. The interpretation of $I_i^2(z_i)$ is therefore that input variables defining pure neighborhoods around a test sample are deemed more significant, and are given a greater weight in the distance calculations. Since there are likely no training samples for which $x_i = z_i$ exactly, Friedman recommends using the set of samples in a small region, Δ_i, around the test point. The definition of Δ_i is based simply on the L neighbors nearest to the test sample along the single input variable x_i so that the relationship between Δ_i and L is $\sum_{n=1}^{N}(|x_{ni} - z_i| \le \Delta_i) = L$. The choice of L is user controlled.

Equations 4−6 provide a mechanism for selecting the K neighbors that are nearest (in a modified sense, using eq 4) to any given test sample. Rather than make this selection all at once, FMNN proposes a recursive procedure similar to RP for selecting the K neighbors. A neighbor list is defined for each test sample, and is initialized to all training samples. After the first iteration, when the modified distance between the test sample and all training samples is calculated, a fixed number of samples farthest away from the test sample are removed from the neighbor list. This process is then repeated until only K samples remain, and these are used to make the response prediction. The number of samples retained after each iteration k is $M_k = \alpha M_{k-1}$, where α ($0 \le \alpha \le 1$) is a user-controlled learning rate. The choice of α determines the balance between the stability with which convergence is achieved and the time required for training. Friedman suggests a value near 0.5.

The choice of β in eq 5 controls the extent to which the significance calculations, $W(\vec{z})$, modulate the intersample distances, with larger β values causing greater modulation. The choice of $\beta = 0$ corresponds to the standard KNN method, while $\beta \to \infty$ produces a model in which only the single most significant input variable (using the $I_i^2(z_i)$ metric) contributes to the distance in each training iteration. Friedman argues that smaller β values are warranted in situations where the number of important input variables is known to be large, since a small β will allow a greater number of variables to contribute to the distance calculations. Since in the present work we have used PCA to construct the input features, and thus predict a large number of these features to be important, we use values of β close to 1.

The current study employed two significant modifications to the original FMNN method, leading to the algorithm referred to as MFMNN in this paper. The first modification involves the measure of purity used to define the variable-significance metric, $I_i^2(z_i)$. The conditional probabilities $p(j|\vec{x} \in R)$ of the Gini metric are modified to have the form

$$p'(j|\vec{x} \in R) = \frac{C(j)\, N_j(R)}{\sum_j C(j)\, N_j(R)} \qquad (8)$$

where $N_j(R)$ is the number of class j samples within the region of interest, and $C(j)$ is the cost associated with misclassifying a sample of class j. This modification is equivalent to the standard RP method of incorporating unequal costs (see Breiman et al.[26]). Such a modification can be particularly important when the prior class probabilities are highly unequal, as is often the case with biological screening data, where the number of inactive compounds often highly exceeds the number of actives. In such cases a trained model is likely to perform significantly worse on the class with fewer representatives, whereas it may be the exact opposite behavior that is desired by the user. A higher cost of misclassifying active compounds can help mitigate this effect. In this study we used a fixed value of $C(j) = 1$ for misclassifying inactives (false positives) and a variable $C(j) = C_{FN}$ ($C_{FN} \ge 1$) for false-negative misclassifications.

The second modification to the FMNN method is an early stopping rule for nearly pure neighborhood regions. For each test-sample prediction, the nearest K neighbors are initially identified using an unmodified distance metric (e.g., standard KNN), and the purity of this default neighborhood is calculated. If the purity is above a user-specified threshold (parametrized as *maxpurity*), a response prediction is made immediately, without proceeding with the modified distance

protocol. A purity value lower than the specified threshold triggers the selection of a new set of *K* neighbors using the procedure outlined above. The purity metric used in this test is the modified Gini metric of eqs 7 and 8. At a typical setting of maxpurity = 0.32, MFMNN is triggered when a decision region has between a 20% and 80% representation of each of the two data classes; larger disparities between class representations trigger the standard KNN method.

It should be noted that, in addition to the significance metric proposed by Friedman for feature weighting, other authors have recently suggested slightly different approaches based on a similar type of sampling protocol.[30,31] It remains a matter of further study to determine whether and how these approaches differ in terms of classification performance and training time.

REFERENCES AND NOTES

(1) Rusinko, A., III; Farmen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. Analysis of a Large Structure/Biological Activity Data Set Using Recursive Partitioning. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1017−1026.

(2) Burden, F. R. Using Artificial Neural Networks to Predict Biological Activity from Simple Molecular Structural Considerations. *Quant. Struct.-Act. Relat.* **1996**, *15*, 7−11.

(3) King, R. D.; Hirst, J. D.; Sternberg, M. J. E. New Approaches to QSAR: Neural Networks and Machine Learning. Perspect. *Drug Discov. Des.* **1993**, *1*, 279−290.

(4) Goodwin, J. T.; Mao, B.; Vidmar, T. J.; Conradi, R. A.; Burton, P. S. Strategies Toward Predicting Peptide Cellular Permeability From Computed Molecular Descriptors. *J. Pept. Res.* **1999**, *53*, 355−369.

(5) Wessel, M. D.; Jurs, P. C.; Tolan, J. W.; Muskal, S. M. Prediction of Human Intestinal Absorption of Drug Compounds from Molecular Structure. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 726−735.

(6) Palm, K.; Luthman, K.; Ungell, A.; Strandlund, G.; Beigi, F.; Lundahl, P.; Artursson, P. Evaluation of Dynamic Polar Molecular Surface Area as Predictor of Drug Absorption: Comparison with Other Computational and Experimental Predictors. *J. Med. Chem.* **1998**, *41*, 5382−5392.

(7) Greene, N.; Judson, P. N.; Langowski, J. J.; Marchant, C. A. Knowledge-Based Expert Systems for Toxicity and Metabolism Prediction: DEREK, StAR, and METEOR. *SAR QSAR Environ. Res.* **1999**, *10*, 299−314.

(8) Enslein, K.; Gombar, V. K.; Blake, B. W. Use of SAR in Computer-Assisted Prediction of Carcinogenicity and Mutagenicity of Chemicals by the TOPKAT Program. *Mutat. Res.* **1994**, *305*, 47−61.

(9) Klopman, G.; Frierson, M. R.; Rosenkranz, H. S. The Structural Basis of the Mutagenicity of Chemicals in Salmonella Typhimurium: The Gene-Tox Data Base. *Mutat. Res.* **1990**, *228*, 1−50.

(10) Ajay; Walters, W. P.; Murcko, M. A. Can We Learn To Distinguish Between "Drug-Like" and "Nondrug-Like" Molecules? *J. Med. Chem.* **1998**, *41*, 3314−3324.

(11) Sadowski, J.; Kubinyi, H. A Scoring Scheme for Discriminating Between Drugs and Nondrugs. *J. Med. Chem.* **1998**, *41*, 3325−3329.

(12) Hansch, C.; Muir, R. M.; Fujita, T.; Maloney, P. P.; Geiger, E.; Streich, M. The Correlation of Biological Activity of Plant Growth Regulators and Chloromycetin Derivatives with Hammett Constants and Partition Coefficients. *J. Am. Chem. Soc.* **1963**, *85*, 2817−2824.

(13) Fujita, T.; Iwasa, J.; Hansch, C. A. New Substituent Constant, π, Derived From Partition Coefficients. *J. Am. Chem. Soc.* **1964**, *86*, 5175−5180.

(14) Cramer, R. D., III; Patterson, D. E.; Bunce, J. D. Comparative Molecular Field Analysis (CoMFA). 1. Effect of Shape on Binding of Steroids to Carrier Proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959−5967.

(15) Zupan, J.; Gasteiger, J. *Neural Networks in Chemistry and Drug Design*; Wiley & Sons: New York, 1999.

(16) (a) Klopman, G. Artificial Intelligence Approach to Structure−Activity Studies. Computer Automated Structure Evaluation of Biological Activity of Organic Molecules. *J. Am. Chem. Soc.* **1984**, *106*, 7315−7321. (b) Ridings, J. E.; Barratt, M. D.; Cary, R.; Earnshaw, C. G.; Eggington, E. E.; Ellis, M. K.; Judson, P. N.; Langowski, J. J.; Marchant, C. A.; Payne, M. P.; Watson, W. P.; Yih, T. D. Computer Prediction of Possible Toxic Action From Chemical Structure; an Update on the DEREK System. *Toxicology* **1996**, *106*, 267−279.

(17) Klopman, G.; Tu, M. Diversity Analysis of 14156 Molecules Tested by the National Cancer Institute for Anti-HIV Activity Using the Quantitative Structure−Activity Relational Expert System MCASE. *J. Med. Chem.* **1999**, *42*, 992−998.

(18) Chen, X.; Rusinko, A., III; Young, S. S. Recursive Partitioning Analysis of a Large Structure−Activity Data Set Using Three-Dimensional Descriptors. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1054−1062.

(19) Stanton, D. T.; Morris, T. W.; Roychoudhury, S.; Parker, C. N. Application of Nearest-Neighbor and Cluster Analyses in Pharmaceutical Lead Discovery. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 21−27.

(20) Hoffman, B.; Cho, S. J.; Zheng, W.; Wyrick, S.; Nichols, D. E.; Mailman, R. B.; Tropsha, A. Quantitative Structure−Activity Relationship Modeling of Dopamine D1 Antagonists Using Comparative Molecular Field Analysis, Genetic Algorithms-Partial Least-Squares, and K Nearest Neighbor Methods. *J. Med. Chem.* **1999**, *42*, 3217−3226.

(21) Dearden, J. C.; Barratt, M. D.; Benigni, R.; Bristol, D. W.; Combes, R. D.; Cronin, M. T. D.; Judson, P. N.; Payne, M. P.; Richard, A. M.; Tichy, M.; Worth, A. P.; Yourick, J. J. The Development and Validation of Expert Systems for Predicting Toxicity. The Report and Recommendations of ECVAM Workshop 24. http://www.jhsph.edu/~altweb/science/pubs/ECVAM/ecvam24.htm.

(22) Friedman, J. H. *Flexible Metric Nearest Neighbor Classification*; Technical Report No. 113; Department of Statistics, Stanford University: Stanford, CA, November 1994.

(23) Parzen, E. On the Estimation of a Probability Density Function and Mode. *Ann. Math. Stat.* **1962**, *33*, 1065−1076.

(24) Cherkassky, V. S.; Gehring, D.; Mulier, F. Comparison of Adaptive Methods for Function Estimation from Samples. *IEEE Trans. N.N.* **1996**, *7*, 969−984.

(25) Morgan, J. N.; Sonquist, J. A. Problems in the Analysis of Survey Data, and a Proposal. *J. Am. Stat. Assoc.* **1963**, *58*, 415−435.

(26) Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. *Classification and Regression Trees*; Wadsworth: Monterey, CA, 1984.

(27) Friedman, J. H. A Recursive Partitioning Decision Rule for Nonparametric Classification. *IEEE Trans. Comput.* **1977**, *26*, 404−408.

(28) Zheng, W.; Tropsha, A. Novel Variable Selection Quantitative Structure−Property Relationship Approach Based on the k-Nearest-Neighbor Principle. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 185−194.

(29) Kohavi, R.; Langley, P.; Yun, Y. The Utility of Feature Weighting in Nearest-Neighbor Algorithms. *Proceedings of the 9th European Conference on Machine Learning* (poster), Prague; Springer: Berlin, 1997.

(30) Hastie, T.; Tibshirani, R. Discriminant Adaptive Nearest Neighbor Classification. *IEEE Trans. Pat. Anal. Mach. Intell.* **1996**, *18*, 607−615.

(31) Daelemans, W.; Durieux, G.; Gillis, S. The Acquisition of Stress: A Data-Oriented Approach. *Comput. Linguistics* **1994**, *20*, 421−451.

(32) Blum, A. L.; Langley, P. Selection of Relevant Features and Examples in Machine Learning. *Artif. Intell.* **1997**, *97*, 245−271.

(33) National Cancer Institute, Bethesda, MD, http://www.nci.nih.gov.

(34) MDL Information Systems, Inc., 14600 Catalina St., San Leandro, CA 94577.

(35) Xue, L.; Godden, J.; Gao, H.; Bajorath, J. Identification of a Preferred Set of Molecular Descriptors for Compound Classification Based on Principal Component Analysis. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 699−704.

(36) Gini, G.; Lorenzini, M. Predictive Carcinogenicity: A Model for Aromatic Compounds, with Nitrogen-Containing Substituents, Based on Molecular Descriptors Using an Artificial Neural Network. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1076−1080.

(37) Vendrame, R.; Braga, R. S.; Takahata, Y.; Galvao, D. S. Structure−Activity Relationship Studies of Carcinogenic Activity of Polycyclic Aromatic Hydrocarbons Using Calculated Molecular Descriptors with Principal Component Analysis and Neural Network Methods. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1094−1104.

(38) Cherkassky, V. S.; Mulier, F. *Learning From Data. Concepts, Theory, and Methods*; Wiley & Sons: New York, 1998.

CI0003348