Knowledge Discovery in Medical and Biological Datasets Using a Hybrid Bayes Classifier/Evolutionary Algorithm

Michael L. Raymer, *Member, IEEE*, Travis E. Doom, *Senior Member, IEEE*, Leslie A. Kuhn, and William F. Punch

***Abstract*—A key element of bioinformatics research is the extraction of meaningful information from large experimental data sets. Various approaches, including statistical and graph theoretical methods, data mining, and computational pattern recognition, have been applied to this task with varying degrees of success. Using a novel classifier based on the Bayes discriminant function, we present a hybrid algorithm that employs feature selection and extraction to isolate salient features from large medical and other biological data sets.**

We have previously shown that a genetic algorithm coupled with a k-nearest-neighbors classifier performs well in extracting information about protein-water binding from X-ray crystallographic protein structure data. The effectiveness of the hybrid EC-Bayes classifier is demonstrated to distinguish the features of this data set that are the most statistically relevant and to weight these features appropriately to aid in the prediction of solvation sites.

***Index Terms*—Bioinformatics, evolutionary computing, genetic algorithms, pattern recognition.**

I. Introduction

EXTRACTION of meaningful information from large biological datasets is a central theme of many bioinformatics research problems. We have previously demonstrated a hybrid algorithm consisting of a nearest-neighbors classifier in conjunction with an evolutionary computation (EC) feature extraction method that performs well in the prediction of conserved water binding to protein surfaces [1], [2], and in the classification of other biological data sets [3]. Here, we present a novel algorithm based on the Bayes classifier that exhibits an improved capability to eliminate spurious features from large datasets, aiding researchers in identifying those features that are related to the particular problem being studied. The effectiveness of this new technique for feature selection and extraction is demonstrated on several biological and medical data sets. A concrete example of the effectiveness of this approach is provided by demonstrating its success in predicting protein-water interactions.

Water molecules bound in ligand-binding sites of proteins often form hydrogen bounds with ligands, making them an essential part of the protein surface with respect to ligand design, docking, and screening [1] and [4]. Furthermore, surface-bound water molecules contribute to the formation and stabilization of surface grooves [5] and [6], while internal water molecules can make a significant contribution to the overall structural stability of the protein [7]. While water is clearly important to protein structure and function, the identification of protein surface sites favorable for solvent binding (as opposed to bulk solvent interactions) has proven difficult, in part because the majority of residues at protein surfaces are hydrophilic.

Empirical methods for determining the favorability of a solvation site do so by analogy with known sites. A site is evaluated and compared to a database of known solvation and nonsolvation sites, and predicted as being more similar to one than the other. A set of features to observe and compare between the solvation sites and nonsolvated sites must be selected prior to classification. We use our novel classifier to distinguish the features of the data set that are the most statistically relevant and to weigh these features appropriately to aid in the classification of possible water coordination sites.

The paper is organized as follows: Section II we present an overview of the main results of the paper and give a brief review of the fundamental concepts and related work. Section III introduces the Bayesian discriminant function with nonlinear weighting and details a Gaussian smoothing factor introduced to mitigate sampling bias. Section IV details our EC-based approach, while Section V reports the experimental results of this approach on several medical datasets and on the prediction of water solvation sites for ligand docking. We conclude with a discussion of these results in Section VI.

II. Preliminaries

A. Bayes Classifier

Consider the task of assigning a sample to one of C classes, $\{\omega_1, \omega_2, \ldots \omega_C\}$, based on the d-dimensional observed feature vector $\vec{x}$. Let $p(\vec{x}|\omega_i)$ be the probability density function for the feature vector, $\vec{x}$, when the true class of the sample is ω_i. Also, let $P(\omega_i)$ be the relative frequency of occurrence class ω_i in the samples. If no feature information is available, the probability that a new sample will be of class ω_i is $P(\omega_i)$—this probability is referred to as the *a priori* or prior probability. Once the feature

Manuscript received November 22, 2001, revised January 7, 2003. This work was supported in part by the National Science Foundation (NSF) under Grant No. EIA-0122582, by the Research Challenge Award, Wright State University, Dayton, OH, and the Ohio Supercomputer Center, Shared Resources Grant. This paper was recommended by Guest Editor N. Bourbakis.

M. L. Raymer and T. E. Doom are with the Department of Computer Science and Engineering, Wright State University, Dayton, OH 45435 USA (e-mail: mraymer@cs.wright.edu, travis.doom@wright.edu).

L. A. Kuhn is with the Department of Biochemistry, Michigan State University, East Lansing, MI 48824 USA (e-mail: kuhn@agua.bch.msu.edu).

W. F. Punch is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail punch@cse.msu.edu).

Digital Object Identifier 10.1109/TSMCB.2003.816922

values are obtained, we can combine the prior probability with the class-conditional probability for the feature vector, $p(\vec{x}|\omega_i)$, to obtain the *a posteriori* probability that a pattern belongs to a particular class. This combination is done using the following Bayes rule [8]:

$$P(\omega_j|\vec{x}) = \frac{p(\vec{x}|\omega_j)\, P(\omega_j)}{\sum_{i=1}^{C} p(\vec{x}|\omega_i)\, P(\omega_i)}. \tag{1}$$

Once the posterior probability is obtained for each class, classification is a simple matter of assigning the pattern to the class with the highest posterior probability. The resulting decision rule is the following Bayes decision rule:

given $\vec{x}$, decide ω_i if

$$P(\omega_i|\vec{x}) > P(\omega_j|\vec{x})\ \forall j.$$

When the class-conditional probability density for the feature vector and the prior probabilities for each class are known, the Bayes classifier can be shown to be optimal in the sense that no other decision rule will yield a lower error rate [9, pp. 10–17]. Of course, these probability distributions (both *a priori* and *a posteriori*) are rarely known during classifier design, and must instead be estimated from training data. Class-conditional probabilities for the feature values can be estimated from the training data using either a parametric or a nonparametric approach. A parametric method assumes that the feature values follow a particular probability distribution for each class and estimate the parameters for the distribution from the training data. For example, a common parametric method first assumes a Gaussian distribution of the feature values, and then estimates the parameters μ_i and σ_i for each class, ω_i, from the training data. A nonparametric approach usually involves construction of a histogram from the training data to approximate the class-conditional distribution of the feature values.

Once the distribution of the feature values has been approximated for each class, the question remains how to combine the individual class-conditional probability density functions for each feature, $p(x_1|\omega_i), p(x_2|\omega_i) \ldots p(x_d|\omega_i)$ to determine the probability density function for the entire feature vector: $p(\vec{x}|\omega_i)$. A common method is to assume that the feature values are statistically independent.

$$p(\vec{x}|\omega_i) = p(x_1|\omega_i) \times p(x_2|\omega_i) \times \cdots \times p(x_d|\omega_i). \tag{2}$$

The resulting classifier, often called the *naïve Bayes classifier* , has been shown to perform well on a variety of data sets, even when the independence assumption is not strictly satisfied [10]. The selection of the prior probabilities for the various categories has been the subject of a substantial body of literature [11]. One of the most common methods is to simply estimate the relative frequency for each class from the training data and use these values for the prior probabilities. An alternate method is to simply assume equal prior probabilities for all categories by setting $P(\omega_i) = 1/C, i = 1 \ldots C$.

B. Feature Costs and the Curse of Dimensionality

The selection of features to measure and include in the feature vector can have a profound impact on the accuracy of the resulting classifier, regardless of what specific classification rule is implemented. A common approach is to have human experts provide as many features as possible that are readily measurable and likely to be related to the classification categories. Unfortunately, there are several disadvantages to evaluating a profuse number of features in classification. First, each additional feature generally incurs an additional cost in terms of measurement time, equipment costs, and storage space. In addition, the computational complexity of classification grows with each additional feature. For some classifiers, the cost of each additional feature in computational complexity can be significant. In addition, the inclusion of spurious features (features unrelated to the classification categories) is likely to reduce classification accuracy. In fact, it is sometimes the case that the inclusion of features that do, in fact, contain information relevant to classification can result in reduced accuracy when the number of training samples is fixed. This phenomenon is sometimes referred to as *the curse of dimensionality* [12]. This effect was illustrated for a specific two-class problem with Gaussian distribution of feature values by Trunk [13].

C. Feature Selection

A number of techniques have been developed to address the problem of dimensionality, including feature selection and feature extraction. The main purpose of feature selection is to reduce the number of features used in classification while maintaining an acceptable classification accuracy. Less discriminatory features are eliminated, leaving a subset of the original features which retains sufficient information to discriminate well among classes. For most problems, the brute-force approach is prohibitively expensive in terms of computation time. Cover and Van Campenhout [14] have shown that to find an optimal subset of size n from the original d features, it is necessary to evaluate all $\binom{d}{n}$ possible subsets when the statistical dependencies among the features are not known. Furthermore, when the size of the feature subset is not specified in advance, each of the (2^d) subsets of the original d features must be evaluated. In the special case where the addition of a new feature always improves performance, it is possible to significantly reduce the number of subsets that must be evaluated using a branch and bound search technique [15]. Unfortunately, this sort of monotonic decrease in the error rate as new features are added is often not found in real-world classification problems due to the effects of the curse of dimensionality and finite training sample sizes.

Various heuristic methods have been proposed to search for near-optimal feature subsets. Sequential methods, including sequential forward selection [16] and sequential backward selection, involve the addition or removal of a single feature at each step. “Plus l—take away r” selection combines these two methods by alternately enlarging and reducing the feature subset repeatedly. The sequential floating forward selection algorithm (SFFS) of Pudil *et al.* [17] is a further generalization of the plus l, take away r methods, where l and r are not fixed, but rather are allowed to "float" to approximate the optimal solution as much as possible.

In a study of current feature selection techniques, Jain and Zongker [18] evaluated the performance of fifteen feature selection algorithms in terms of classification error and run time on a 2-class, 20-dimensional, multivariate Gaussian data set. Their findings demonstrated that the SFFS algorithm dominated the other methods for this data, obtaining feature selection results comparable to the optimal branch-and-bound algorithm while requiring less computation time.

When classification is being performed using neural networks, node pruning techniques can be used for dimensionality reduction [19]. After training for a number of epochs, nodes are removed from the neural network in such a manner that the increase in squared error is minimized. When an input node is pruned, the feature associated with that node is no longer considered by the classifier. Similar methods have been employed in the use of fuzzy systems for pattern recognition through the generation of fuzzy if-then rules [20], [21]. Some traditional pattern classification techniques, while not specifically addressed to the problem of dimensionality reduction, can provide feature selection capability. Tree classifiers [22], for example, typically partition the training data based on a single feature at each tree node. If a particular feature is not tested at any node of the decision tree, it is effectively eliminated from classification. Additionally, simplification of the final tree can provide further feature selection [23].

D. Feature Extraction

Feature extraction, a superset of feature selection, involves transforming the original set of features to provide a new set of features, where the transformed feature set usually consists of fewer features than the original set. While both linear and nonlinear transformations have been explored, most of the classical feature extraction techniques involve linear transformations of the original features. Formally, the objective for linear feature extraction techniques can be stated as follows:

Given an $n \times d$ pattern matrix $\mathcal{A}$ (n points in a d-dimensional space), derive an $n \times m$ pattern matrix $\mathcal{B}$, $m < d$, where $\mathcal{B} = \mathcal{A}\mathcal{H}$ and $\mathcal{H}$ is a $d \times m$ transformation matrix.

According to this formalization, many common methods for linear feature extraction can be specified according to the method of deriving the transformation matrix, $\mathcal{H}$. For unsupervised linear feature extraction, the most common technique is principal component analysis [9]. For this method, the columns of $\mathcal{H}$ consist of the eigenvectors of the $d \times d$ covariance matrix of the given patterns. It can be shown that the new features produced by principal component analysis are uncorrelated and maximize the variance retained from the original feature set [9]. The corresponding supervised technique is linear discriminant analysis. In this case, the columns of $\mathcal{H}$ are the eigenvectors corresponding to the nonzero eigenvalues of the matrix $\mathcal{S}_W^{-1}\mathcal{S}_B$, where $\mathcal{S}_W$ is the within-class scatter matrix and $\mathcal{S}_B$ is the between-class scatter matrix for the given set of patterns. Deriving $\mathcal{H}$ in this way maximizes the separation between class means relative to the covariance of the classes [9]. In the general case, the matrix $\mathcal{H}$ is chosen to maximize some criteria, typically related to class separation or classification accuracy for a specific classifier. In this view, feature selection is a special case of linear feature extraction, where the off-diagonal entries of $\mathcal{H}$ are zero, and the diagonal entries are either zero or one.

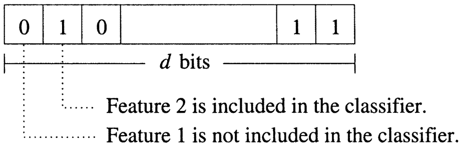

0
1
0
1
1
d bits
Feature 2 is included in the classifier.
Feature 1 is not included in the classifier.

Fig. 1. The d-dimensional binary vector, comprising a single member of the EC population for EC-based feature selection.

While feature extraction methods often outperform feature selection techniques in terms of classification accuracy, feature extraction generally does not completely remove features from consideration by the classifier, because each extracted feature is calculated from some combination of the original features. As a result, all of the original features must be measured, calculated, and/or stored for the trained classifier. Feature selection methods have the advantage of reducing the costs associated with feature measurement and storage. Hybrid methods attempt to reduce the number of features considered by a classifier, while recombining the remaining features to increase classification accuracy.

E. Evolutionary Computation in Feature Selection and Extraction

A direct approach to using EC for feature selection was introduced by Siedlecki and Sklansky [24]. In their work, an EC is used to find an optimal binary vector, where each bit is associated with a feature (Fig. 1). If the ith bit of this vector equals 1, then the i^{th} feature is allowed to participate in classification; if the bit is a 0, then the corresponding feature does not participate. Each resulting subset of features is evaluated according to its classification accuracy on a set of testing data using a nearest-neighbor classifier [25].

This technique was later expanded to allow linear feature extraction by Punch *et al.* [26] and independently by Kelly and Davis [27]. The single bit associated with each feature is expanded to a real-valued coefficient, allowing independent linear scaling of each feature, while maintaining the ability to remove features from consideration by assigning a weight of zero. Given a set of feature vectors of the form $\mathbf{X} = \{x_1, x_2, \ldots x_d\}$, the EC produces a transformed set of vectors of the form $\mathbf{X}' = \{w_1 x_1, w_2 x_2 \ldots w_d x_d\}$ where w_i is a weight associated with feature i. Each feature value is first normalized, then scaled by the associated weight prior to training, testing, and classification. This linear scaling of features prior to classification allows a classifier to discriminate more finely along feature axes with larger scale factors. A k-nearest-neighbors (knn) classifier is used to evaluate each set of feature weights. Patterns plotted in feature space are spread out along feature axes with higher weight values, and compressed along features with lower weight values. The value of k for the knn classifier is fixed and determined empirically prior to feature extraction.

In a similar approach, Yang and Honavar [28] use a simple EC for feature subset selection in conjunction with DistAI, a neural network-based pattern classifier [29]. As in other EC-based feature selectors, a simple binary representation was used where each bit corresponds to a single feature. The use of the EC for feature subset selection improved the accuracy of the DistAI classifier for nearly all of the data sets explored, while simultaneously reducing the number of features considered. Their hybrid classifier, GADistAI, outperformed a number of modern classification methods on the various data sets presented.

Vafaie and De Jong [30] describe a hybrid technique in which EC methods are employed for both feature selection and extraction[1] in conjunction with the C4.5 decision tree classifier system [31]. Again, a binary representation is used for feature subset selection using traditional EC techniques. In this system, however, the features seen by the classifier are functions of the original features composed of simple arithmetic operations. For example, one such feature might be $\{(F1 - F2) \times (F2 - F4)\}$, where $F1$, $F2$, and $F4$ represent values from the original feature set.

In our previous work, it was demonstrated that a knn classifier can be hybridized with an EC-based feature selection and extraction technique to reduce the number of features considered by the classifier while maintaining or increasing classification accuracy [3]. Here we present a new hybrid classifier loosely based on the idea of EC feature weighting. A parameterized discriminant function based on the Bayes classifier is developed, and an EC optimizer is used to tune the parameters of the new discriminant function. We show that this new hybrid system is effective at feature selection for various medical and biological data sets.

III. Bayesian Discriminant Functions

The Bayesian classifier has a computational advantage over the previously-employed knn classifier [3] in that the training data are summarized, rather than stored. The comparison of each test sample with every known training sample to find nearest neighbors during knn classification is a computationally expensive process, even when efficient search methods are employed [32]. In contrast, finding the marginal probability associated with a particular feature value is computationally efficient for both the parametric and nonparametric forms of the Bayesian classifier. Since EC-based hybrid classifiers require many classifications to be performed during feature selection and extraction, the use of a computationally efficient classifier such as a Bayes classifier is indicated.

Unfortunately, the direct application of feature weighting as described in [1] is not effective in conjunction with the Bayes classifier, because the Bayes decision rule is invariant to linear scaling of the feature space. In other words, multiplying the feature values for a given feature by a constant has no effect on the class-conditional probabilities considered by the classifier, as illustrated in Fig. 2. Direct scaling of the marginal probabilities is also ineffective for the naïve Bayes classifier, since the joint class-conditional probabilities are simply the products of the marginal probability values.

There are, nevertheless, several aspects of the Bayesian classifier that, when optimized, can yield better classification performance. One such area is is the manner in which the marginal probabilities for each feature will be combined into the multivariate class-conditional probability densities. For the naïve Bayes classifier, the class-conditional probability is the product of the marginal probabilities for each feature. A more general approach would be to encode the entire $d \times d$ covariance matrix describing the interrelationships between all the features being considered, and allow an EC-based optimizer to search for the covariance matrix which best describes the true multivariate distribution of the training data. Unfortunately, the search space involved in finding this covariance matrix grows as 2^{d^2}, even if the elements of the covariance matrix are binary-valued. For real valued matrix elements, the search space quickly becomes intractable, even for small problems.

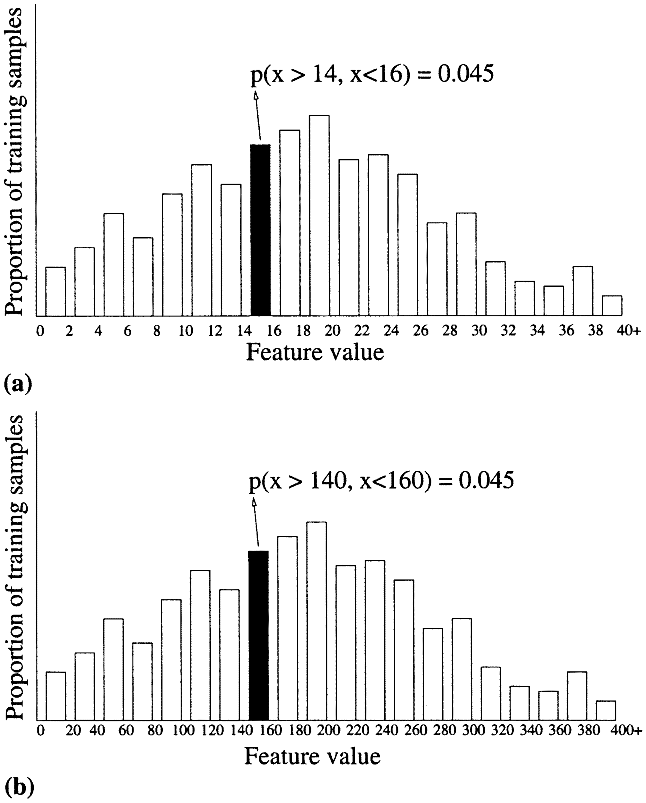


Fig. 2. Bayes decision rule is invariant to linear transformations of the feature space. For the feature shown here, the raw feature values (a) have been multiplied by 10 in (b). Using a nonparametric Bayes classifier, we find that the original feature value falls in the bin 14–16 (black rectangle) in the original histogram. The scaled feature falls in the equivalent bin of histogram (b), and the histogram values (marginal probabilities) of the two bins are identical, so the scaling has no bearing on the classification results.

We can simplify the problem somewhat by viewing the Bayes decision rule as a *discriminant function*—a function g of the feature vector $\vec{x}$. Consider, by way of example, a two-class decision problem. The Bayes discriminant function can be written as

$$g(\vec{x}) = P(\omega_1|\vec{x}) - P(\omega_2|\vec{x}). \tag{3}$$

Here, we would decide class 1 if $g(\vec{x}) > 0$, and class 2 if $g(\vec{x}) < 0$. The classification when $g(\vec{x}) = 0$ is arbitrary. The discriminant function, then, is uniquely associated with a particular classifier, mapping an input feature vector to a value associated with a particular class. According to Duda and Hart:

[1] The authors use the term "feature construction."

> "We can always multiply the discriminant functions by a positive constant or bias them by an additive constant without influencing the decision. More generally, if we replace every $g_i(\mathbf{x})$ by $f(g_i(\mathbf{x}))$, where f is a monotonically increasing function, the resulting classification is unchanged" [9, pp. 17–18].

Thus, we can design a parameterized classifier based on the concept of the discriminant function. We begin with a discriminant function based on the Bayes decision rule. Using this function as a model, we can design similar functions which classify well, but are more easily parameterizable for hybridization with EC optimization methods. After designing such a discriminant function and identifying the tunable parameters, we can use an EC to optimize these parameters with regard to a particular set of training and tuning data.

A. Nonlinear Weighting of the Bayes Discriminant Function

Consider the Bayes discriminant function

$$\begin{aligned} g(\vec{x}) =& P(\omega_1|\vec{x}) - P(\omega_2|\vec{x}) \\ =& \frac{P(\vec{x}|\omega_1) \times P(\omega_1) - P(\vec{x}|\omega_2) \times P(\omega_2)}{\sum_{i=1}^{2} P(\vec{x}|\omega_i) \times P(\omega_i)}. \end{aligned} \tag{4}$$

The denominator can be eliminated, since it does not affect the sign of $g(\vec{x})$, and thus does not affect the resulting classification. Since $a > b \Rightarrow \log(a) > \log(b)$, we can apply the log function to the *a posteriori* probabilities without changing the resulting classification. Thus, the following discriminant function is equivalent to the naïve Bayes discriminant.

$$\begin{aligned} g(\vec{x}) =& \log(P(\vec{x}|\omega_1) \times P(\omega_1)) \\ & - \log(P(\vec{x}|\omega_2) \times P(\omega_2)) \qquad (5) \\ =& (\log(P(\vec{x}|\omega_1)) + \log(P(\omega_1))) \\ & - (\log(P(\vec{x}|\omega_2)) + \log(P(\omega_2))) \qquad (6) \end{aligned}$$

where

$$\begin{aligned} \log(P(\vec{x}|\omega_i)) =& \log(P(x_1|\omega_i)) + \log(P(x_2|\omega_i)) \\ & + \cdots + \log(P(x_d|\omega_i)). \end{aligned} \tag{7}$$

Finally, we can parameterize this discriminant function, while maintaining a similar level of classification accuracy, by adding coefficients to each of the marginal probabilities

$$\begin{aligned} P^*(\vec{x}|\omega_i) =& C_1 \log(P(x_1|\omega_i)) + C_2 \log(P(x_2|\omega_i)) \\ & + \cdots + C_d \log(P(x_d|\omega_i)) + \log(P(\omega_i)). \end{aligned} \tag{8}$$

The values for the coefficients, $C_{1\ldots d}$, are supplied by an EC optimizer. The effect of these coefficients is to apply a nonlinear weighting to each of the marginal probabilities, which are then combined to produce a confidence value, P^*, for each class. While $P^*(\vec{x}|\omega_i)$ is no longer a joint probability distribution, the discriminant function is equivalent to the naïve Bayes discriminant function when $C_1 = C_2 = \cdots = C_d = 1$. Furthermore, the new function has several desirable features for hybridization with an EC optimizer. When a particular coefficient, C_j, is reduced to zero, the associated feature value, x_j, is effectively eliminated from consideration by the classifier. This allows us to perform feature selection in conjunction with classifier tuning. Furthermore, when the value of a coefficient, C_j is increased, the marginal probability value for the associated feature, x_j, has an increased influence on the value of the confidence value, $P^*(\vec{x}|\omega_i)$, for each class.

B. Gaussian Smoothing

The implementation for this discriminant function was based on the previously described nonparametric naïve Bayes classifier. The marginal probability distributions for each feature were approximated using histograms with 20 bins each. A Gaussian smoothing factor was applied in order to mitigate sampling anomalies that might introduce classification bias, and to reduce the sensitivity of the classifier to changes in bin size. Given a feature value, x_i, for feature i, and a class ω_j, then let $b_{\omega_j}(x_i)$ be the bin that x_i occupies in the histogram for class ω_j. When the Gaussian smoothing is applied, the effective marginal probability $p(x_i|\omega_j)$ depends on the histogram value of bin $b_{\omega_j}(x_i)$, as well as the histogram values of neighboring bins. Let $h_{\omega_j}\left(b_{\omega_j}(x_i)\right)$ be the histogram value for bin $b_{\omega_j}(x_i)$—that is, the proportion of the training samples of class ω_j that have values for feature i that belong in the bin $b_{\omega_j}(x_i)$—then the effective marginal probability for feature value x_i is

$$p(x_i|\omega_j) = \sum_{k=-\sigma}^{+\sigma} \left(G(k,\sigma) \times h_{\omega_j}\left(b_{\omega_j}(x_i) + k\right) \right) \tag{9}$$

where $G(k,\sigma)$ is the mass density function for the Gaussian distribution at $\mu = 0.0$, with variance σ^2

$$G(k,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2} \left(\frac{x}{\sigma}\right)^2. \tag{10}$$

The value of σ, a run-time parameter, determines the number of bins that will contribute to each effective marginal probability value. Fig. 3 illustrates the effect of Gaussian smoothing on the effective marginal probability for a particular feature value.

For the experiments detailed here, Gaussian smoothing was applied with $\sigma = 2.0$.

IV. EC Optimization of the Nonlinear Discriminant Coefficients

Several EC-based methods were employed to optimize the coefficients of the Bayes-derived discriminant function. The experiments described here employ a tradiational genetic algorithm (GA)-like method using two point crossover and bitwise point mutation.

During the execution of the EC, each coefficient vector is passed to the classifier for evaluation, and a cost (or inverse fitness) score is computed, based primarily on the accuracy obtained by the parameterized discriminant function in classifying a set of samples of known class. Since the evolutionary algorithm seeks to minimize the cost score, the formulation of the cost function is a key element in determining the quality of the resulting classifier. Coefficients are associated with each term

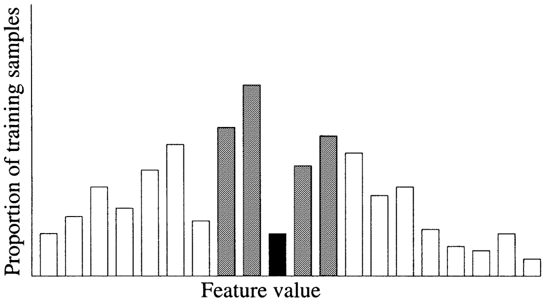


Fig. 3. Effects of Gaussian smoothing on the computation of effective marginal probabilities. Assuming that the current feature value falls in the center bin (black rectangle), and assuming $\sigma = 2$, then the two surrounding bins on either side (grey rectangles) also contribute to the effective marginal probability for the current feature.

in the EC cost function that allow control of each run. The following cost function is computed by the knn classifier for each individual (consisting of a weight vector and k value):

$$\begin{aligned}\text{cost}\,(\vec{w},k) =& C_{acc} \times Err\,(\vec{w},k) \\ &+ C_{pars} \times \text{nonzero}\,(\vec{w}) \\ &+ C_{bal} \times Bal\,(\vec{w},k) \qquad (11)\end{aligned}$$

where Err is the error rate; $\text{nonzero}\,(\vec{w})$ is the number of nonzero coefficients in the discriminant function coefficient vector $\vec{w}$; and Bal is the balance, defined as the difference between the maximum and minimum classification accuracy among all classes. Additionally, C_{acc}, C_{pars}, and C_{bal} are coefficients for each of these terms, respectively. The coefficients determine the relative contribution of each part of the fitness function in guiding the EC search. The values for the cost function coefficients were determined empirically in a set of initial experiments for each data set. Typical values for these coefficients are $C_{\text{acc}} = 20.0$, $C_{\text{pars}} = 1.0$, and $C_{\text{bal}} = 10.0$.

A. Representation Issues and Masking

The representation of the discriminant function coefficients on the chromosome is fairly direct—an integer value from 0 to 100. In order to infer the minimal set of features required for accurate classification, it is desirable to promote parsimony in the discriminant function—that is, as many coefficients should be reduced to zero as possible without sacrificing classification accuracy. While the cost function encourages parsimony by penalizing a coefficient vector for each nonzero value, a simple, integer-valued representation for the coefficients themselves does not provide an easy means for the EC to reduce coefficients to zero. Since the EC mutation operator tends to produce a small change in a single weight value, numerous mutations of the same feature weight are often required to yield a value at or near zero. Several methods were tested to aid the search for a minimal feature set, including reducing weight values below a predefined threshold value to zero, and including a penalty term in the cost function for higher weight values. The method that proved most effective, however, was a hybrid representation that incorporates both the EC feature selection technique of Siedlecki and Sklansky [24] and the feature weighting techniques of Punch *et al.* [26] and Kelly and Davis [27]. In this hybrid representation, a mask field is assigned to each feature. The contents of the mask field determine whether the feature is included in the classifier (see Fig. 4). In the initial implementation, a single mask bit was stored on the chromosome for each feature. If the value of this bit was 1, then the feature was weighted and included in classification. If, on the other hand, the mask bit for a feature was set to 0, then the feature weight was treated effectively as zero, eliminating the feature from consideration by the classifier. Since the masking fields comprised a very small section of the chromosome relative to the discriminant function coefficients, the number of mask bits associated with each feature was later increased to five. This increase had the effect of increasing the probability that a single bit mutation in a random location would affect the masking region of the chromosome. The interpretation of multiple mask bits is a simple generalization of the single bit case. When the majority of the mask bit values for a field are 1, then the field is weighted and included in classification. Otherwise, the field weight is reduced to 0, removing the feature from consideration by the knn. The number of mask bits is always odd so there is no possibility of a tie. Fig. 4 shows a typical EC chromosome for the nonlinear Bayes discriminant classifier with masking.

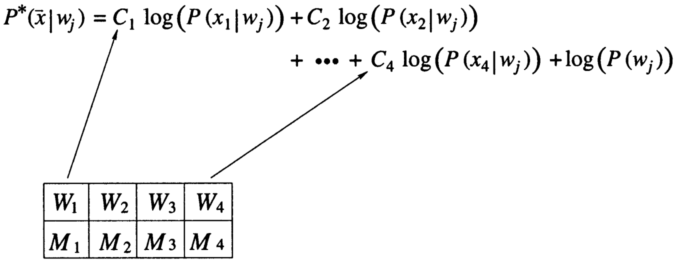

$P^*(\bar{x}|w_j) = C_1 \log(P(x_1|w_j)) + C_2 \log(P(x_2|w_j))$
$+ \cdots + C_4 \log(P(x_4|w_j)) + \log(P(w_j))$
W_1 W_2 W_3 W_4
M_1 M_2 M_3 M_4

Fig. 4. Example of the EC chromosome for optimization of the nonlinear discriminant coefficients. A four-dimensional problem is shown. Each coefficient, C_i, in the discriminant function is determined by the chromosome weight, W_i, and the masking field, M_i.

V. Experimental Results

A. Classification of UCI Data Sets

Classification of data from the University of California, Irvine (UCI) machine learning data set repository was performed to evaluate the effectiveness of the hybrid classifier on real-world data, and to facilitate comparison with other classifiers. The four data sets used for this evaluation are described in detail in [3], and at the UCI website [33]. A brief synopsis of each data set follows.

Hepatitis: This data consists of 19 descriptive and clinical test result values for 155 hepatitis patients [34], [35]. The two classes, survivors and patients for whom the hepatitis proved terminal, are strongly unbalanced—123 samples belong to the survivor class while 32 belong to the terminal class. The data includes qualitative, as well as both continuous and discrete-valued quantitative features. There are missing values, the number of which varies largely by feature. Many features have no missing values, while others have as many as 67 missing values out of 155 samples. The small sample size and incompleteness of this data set are typical of many medical classification problems.

TABLE I
Results of the Nonlinear-Weighted Bayes Discriminant Function (**Nonlinear**) on Various Data Sets From the UCI Machine Learning Data set Repository, Averaged Over 50 Runs. **Train** Refers to the Accuracy Obtained When Reclassifying the Data Used by the EC in Tuning (Optimizing) Feature Subsets and Weights. **Test** Refers to the Accuracy Obtained on an Independent Test Set for Each Experiment, Disjoint From the Training and Tuning Sets. The Number of Features is the Mean Number of Features Used in Classification Over All 50 Runs

Hepatitis	**Train**	**Test**	**Features**
Naïve Bayes	85.3	65.7	19
Nonlinear Bayes	95.4	79.4	6.5
EC/knn	86.0	69.6	8.1
EC/knn	87.2	73.1	8.9

Wine	**Train**	**Test**	**Features**
Naïve Bayes	98.8	94.7	13
Nonlinear Bayes	97.8	91.3	4.5
EC/knn	99.7	94.8	6.0
EC/knn	99.5	93.2	6.2

Ionosphere	**Train**	**Test**	**Features**
Naïve Bayes	93.0	90.1	34
Nonlinear Bayes	97.6	87.5	8.5
EC/knn	95.0	91.9	8.5
EC/knn	93.2	92.3	13.5

Pima	**Train**	**Test**	**Features**
Naïve Bayes	76.1	64.6	8
Nonlinear Bayes	76.2	70.4	3.9
EC/knn	80.0	72.1	3.1
EC/knn	79.1	72.9	3.9

Pima: Diabetes diagnosis information for native American women of the Pima heritage, aged 21 or over [36]. This data consists diagnostic information for 768 women; 268 of these patients tested positive for diabetes, while 500 tested negative. Six of the eight features are quantitative and continuous, consisting of various clinical test results. The remaining two features, age in years and number of times pregnant, are quantitative and discrete. There are no missing feature values in the data. The completeness and moderate dimensionality of this data set make it suitable for testing the ability of a classifier and feature extractor to maintain or increase classification accuracy while reducing dimensionality when there are fewer features to work with.

Wine: This data set consists of the results of a chemical analysis of wines derived from three different cultivars [37], [38]. There are 13 continuous features, with no missing values. There are 59, 71, and 48 members of each of the three classes, respectively. The three classes are nearly linearly separable, and linear discriminant analysis can obtain 98.9% accuracy over all three classes. This data set is thus better for evaluating feature selection capability than classifier accuracy.

Ionosphere: The 34 continuous features in this data set are derived from the signals read by a phased array of 16 high-frequency antennas in Goose Bay, Labrador [39]. These radar signals are designed to recognize structure in the ionosphere. Each reading consists of 17 pulses, with two attributes per pulse resulting in 34 features. There are 351 samples in this data set—225 are considered "good" readings, for which some structure in the ionosphere was detected, while 126 readings showed no structure. There are no missing feature values in this data. This data set was selected for evaluation of feature selection capability for higher-dimensionality data sets.

TABLE II
Execution Times (Wall Clock Time) for 200 Generations of EC Optimization of the knn and Nonlinear Discriminant Function Classifiers. For Each Data Set, the Number of Features (d). the Number of Classes (C). the Combined Training and Tuning Set Size (n), and the Mean Execution Time (Hours:Minutes:Seconds) Over 50 Runs Are Shown. Each Run Was Executed on a Single 250 MHz UltraSPARC-II Cpu of a Six-Cpu Sun Ultra-Enterprise System With 768 MB of System RAM. Runs Were Executed in Sets of Five With no Other User Processes Present on the System

Data set	**d**	**C**	**n**	**knn**	**nonlinear**
Pima	8	2	400	1:40:13	0:10:52
Hepatitis	19	2	240	1:05:48	0:24:42
Ionosphere	34	2	400	2:02:25	0:43:37
Wine	13	3	240	0:23:59	0:14:39

Table I compares the results of the EC/Bayes classifier with those of several previously developed classifiers, including a knn classifier using a traditional GA for feature extraction (GA/knn), and a knn classifier using an EC method employing Gaussian mutation without recombination for feature selection (EC/knn) [3].

The most evident aspect of the results on these four data sets is the feature selection capability demonstrated by the nonlinear discriminant function. For three of the four data sets, the minimum number of features used in classification was found by the nonlinear discriminant function in conjunction with the EC. Additionally, for the hepatitis data, the test accuracy obtained by the two discriminant function classifiers surpassed the other classifiers tested. For the other three data sets the accuracies obtained by the discriminant methods were similar to those obtained by other methods tested. The notable difference between **Train** and **Test** results for the hepatitis and ionosphere data sets suggest that the discriminant classifiers may be more prone to overfitting of the training and tuning data than the other classifiers.

Examination of the run times for the UCI data sets illustrates the advantage held by the discriminant-function-based classifiers over the EC/knn hybrid classifiers in terms of computational efficiency. Table II compares the execution times for 200 generations of EC optimization in conjunction with the nonlinear discriminant function and the knn classifier. In all cases the nonlinear discriminant classifier is significantly faster than the EC/knn—in the case of the Pima Indian diabetes data set the difference is nearly tenfold.

B. Classification of Medical Data

Two additional data sets, also selected from the UCI repository, were employed by [40], [41] in a comparative study of classification methods from statistical pattern recognition, neural networks, and machine learning. These two medical data sets, **thyroid** and **appendicitis**, are included here to facilitate comparison with these results. The thyroid data consists of 21 clinical test results for a set of patients tested for thyroid dysfunction [42]—15 of these features are binary-valued, while the other six are continuous. The training data consist of 3772 cases from the year 1985, while the testing data consist of 3428 cases from

TABLE III
Accuracy of Various Classifiers on the Hypothyroid and Appendicitis Diagnosis Data Sets. Results for the Discriminant Function Classifiers are Averaged Over Five EC Experiments. Results for the EC/knn Classifier Represent the Best of Five Experiments. **Train** Refers to the Accuracy Obtained in Reclassifying the EC Tuning Set; **Test** Refers to Bootstrap Accuracy Over 100 Bootstrap Sets

Thyroid	**Train**	**Test**	**Features**
EC/knn	98.5	98.4	3
Nonlinear Bayes	97.7	97.2	2.7

Appendicitis	**Train**	**Test**	**Features**
EC/knn	90.4	90.6	2
Nonlinear Bayes	80.4	67.0	2.6

the following year. The data are grouped into two classes, consisting of the patients that were/were not diagnosed as having certain categories of hypothyroid disorder. The two classes are highly unbalanced: the training data consist of 3487 negative diagnoses and 284 positive, while the testing data consist of 3177 negative samples and 250 positive. The appendicitis data consists of seven laboratory tests to confirm the diagnosis of acute appendicitis [43]. All seven features are continuous. This data set consists of only 106 samples in two classes. 85 patients had confirmed appendicitis while 21 did not.

For the thyroid and appendicitis data, the discriminant function-based classifiers were trained and tested in the same manner as the previously developed EC-hybrid classifiers [3]. For each data set, five experiments were conducted for each classifier. The appendicitis data set was re-partitioned into disjoint training/tuning and testing sets for each experiment. The much larger thyroid data set was pre-partitioned into training and testing sets in the UCI database [42]. For this data, only the initial random EC population was changed for each experiment. The results of these experiments are summarized in Table III.

The discriminant function based classifier performed well on the hypothyroid diagnosis data, utilizing a smaller feature set than the EC/knn at a slight cost in bootstrap test accuracy. The poor performance of the nonlinear classifier on the appendicitis data set, along with the previous results for the hepatitis and ionosphere data sets, suggests that the discriminant function classifier may be prone to overfitting when presented with small (<50 samples of each class) data sets for training, tuning, and testing.

C. Classification of Coordinated Water Molecules

One of the more challenging problems in structure-based rational drug design is modeling the interactions between a protein surface and the water molecules that surround the protein. For analyzing conserved solvation, the Brookhaven Protein Databank (PDB) [44] and [45] was screened for high-resolution crystallographic protein structures to provide a knowledge base of crystallographically observed solvation sites. All proteins included in the database were nonhomologous (had $\leq 25\%$ sequence identity, based on the PDB_Select list [46]), to avoid redundant structural information. Proteins with a resolution of $\leq \sim 2.0$ Å and low residual R-values were preferred. Table IV lists the 30 proteins selected.

The first hydration shell was defined as the set of water molecules within 3.6 Å of any protein surface atom, and thus capable of making a van der Waals' contact or hydrogen bond with the protein. The environment of each of first-shell water molecules from each structure was characterized according to the six features listed in Table V. The resulting database consisted of these six feature measurements for each of 5325 first-shell water molecules.

To distinguish protein solvation sites from nonsolvated sites, it was first necessary to generate a set of randomly spaced probe sites about the protein surface where no bound water was observed in the crystal structure. To generate the probe site positions, the solvent accessible molecular surface was computed for each of the structures in Table IV using MS [48]; a probe radius of 1.2 Å was used, with a surface density of $1\ \mathrm{dot}/\text{Å}^2$ and unit normals generated at each surface dot. For each protein, the minimum distance of any crystallographically observed water molecule from the protein surface, $d_{\min}$, and the maximum distance, $d_{\max}$ were determined. For each surface point, a probe site was generated and placed at a distance along the surface normal, selected at random over the range of distances from $d_{\min}$ to $d_{\max}$. Any probes overlapping crystallographically observed water sites or other probes (with positions within 3.2 Å) were removed. Finally, the same number of probe sites were selected for each protein as there were crystallographic water sites. This selection was done so that the distribution of distances of probes from the protein surface matched the distribution of distances for observed water molecules for that protein.

Non-solvent sites were generated using this technique for each of the 30 protein structures in Table IV. Each probe site was characterized using the same physical and chemical features that were used to characterize observed solvation sites (Table V). For observed solvation sites, the temperature factor (B-value) and occupancy value from the crystallographic data can be used to estimate the thermal mobility of the water molecule in the context of the protein crystal. However, since temperature factor and occupancy are undefined for probe sites, two new features were computed to approximate the local thermal mobility of a probe site. ABVAL is the average (arithmetic mean) B-value of all protein atoms within 3.6 Å of the probe position. NBVAL is a nonnormalized version of the same feature. That is, the simple sum of the B-values of all neighboring (within 3.6 Å) protein atoms.

In measuring the number of hydrogen bonds to other water molecules (HBDW) for probe sites, potential hydrogen bonds to neighboring probe sites were included in the calculation. In other words, probe sites and crystallographically observed water molecules were treated equally for purposes of feature value calculation. The final database consisted of the environments of 5325 crystallographically observed water molecules, and 5325 nonsolvated sites.

From this dataset, several independent optimization experiments were conducted using various EC parameters (recombination rate and method, selection strategy, etc.). Each experiment was conducted using nonintersecting training and tuning sets consisting of 1000 observed water binding sites and 1000 nonsolvated sites. After the optimization step was completed, the trained nonlinear Bayes classifier was tested using a variant

TABLE IV
PROTEINS INCLUDED IN THE SOLVATION KNOWLEDGE BASE

PDB Code	Protein	Resolution(Å)	Water Molecules
1ahc	α-momorcharin	2.0	163
1apm	cAMP-dependent protein kinase	2.0	207
1bia	bira bifunctional protein	2.3	43
1bsa	barnase	2.0	258
1ca2	carbonic anhydrase II	2.0	167
1cgf	fibroblast collagenase	2.1	181
1cgt	cyclodextrin glycosyltransferase	2.0	588
1chp	cholera toxin β pentamer	2.0	248
1dr2	dihydrofolate reductase	2.3	73
1gta	glutathione S-transferase	2.4	118
1hel	hen egg-white lysozyme	1.7	185
1lib	adipocyte lipid-binding protein	1.7	89
1nsb	neuraminidase	2.2	446
1poa	phospholipase A2	1.5	151
1syc	staphylococcal nuclease	1.8	69
1thm	thermitase	1.37	193
1udg	uracil-DNA glycosylase	1.75	121
2act	actinidin	1.7	272
2apr	acid proteinase	1.8	373
2cla	chloramphenicol acetyltransferase	2.35	104
2ctv	concanavalin A	1.95	146
2sga	proteinase A	1.5	220
2wrp	Trp repressor	1.65	170
3cox	cholesterol oxidase	1.8	453
3dni	deoxyribonuclease I	2.0	375
3enl	enolase	2.25	353
3grs	glutathione reductase	1.54	523
3tln	thermolysin	1.6	173
5cpa	carboxypeptidase A	1.54	315
See note 1	RTEM-1 β-lactamase	1.7	182

[1] Provided by Drs. Natalie Strynadka and Michael James, University of Alberta, Edmonton.

TABLE V
PHYSICAL AND CHEMICAL FEATURES USED TO REPRESENT PROTEIN-BOUND WATER MOLECULES AND PROTEIN SURFACE SITES

Tag	Feature	Description
ADN	Atomic density	The number of protein atom neighbors within 3.6 Å of the water molecule. This feature correlates with the local protein topography. Water molecules bound in deep grooves will have high ADN values, while those bound to protrusions will have low ADN values [5].
AHP	Atomic hydrophilicity	The hydrophilicity of the neighborhood of the water molecule is based on the frequency of hydration for each atom type in 56 high-resolution protein structures [47]. Each water molecule is assigned an AHP value equal to the sum of the atomic hydrophilicity values of all atom neighbors within 3.6 Å of the water molecule.
HBDP	Hydrogen bonds to protein	The number of hydrogen bonds between the water molecule and neighboring protein atoms. Each donor or acceptor atom within 3.5 Å is considered a potential hydrogen bond.
HBDW	Hydrogen bonds to water	The number of hydrogen bonds between the water molecule and other water molecules in the ligand-free protein structure, based on ≤ 3.5 Å distance between oxygen atoms in the two water molecules.
ABVAL	Average B-value of protein atom neighbors	The average (mean) temperature factor of all protein atoms within 3.6 Å of the water molecule.
NBVAL	Net B-value of protein atom neighbors	The sum of the B-values of all protein atoms within 3.6 Å of the water molecule.

of the bootstrap method. The set of 3325 solvation sites and 3325 probe sites not used in training the Bayes classifier or EC tuning of the weight coefficients was sampled with replacement to form a cross-validation testing set consisting of 100 water binding sites and 100 nonsolvated sites. The trained, weighted Bayes classifier was tested on this data, and the procedure was repeated 1000 times to provide an estimate of the mean accuracy and the standard deviation of the prediction accuracy for the classifier. The highest scoring EC experiment used two-point crossover with a frequency of 0.7 crossover events per individual per generation, bitwise mutation with a probability of 0.125 per bit per generation, and traditional fitness-proportionate selection. A population size of 200 individuals was used, and the run was conducted for 1000 generations.

The final testing accuracy for this experiment was 66.8%, which compares favorably with other knowledge-based methods for solvent site prediction. The bootstrap tests resulted in a cross-validation accuracy of 65.24% with a standard deviation of 3.38%. Of the six features provided to the classifier, two (atomic hydrophilicity and average temperature factor) were eliminated by the feature selection mechanism in the EC optimizer. Upon first inspection, elimination of the atomic hydrophilicity feature (AHP) is slightly surprising from a biochemical perspective, since this feature is the most direct measure of the tendency of local protein atoms to form hydrogen bonds with water molecules. A likely explanation is that AHP is removed due to its high correlation with other features such as atomic density and hydrogen bonds to protein. Thus,

TABLE VI
CONFUSION MATRIX FOR CROSS-VALIDATION TESTING OF THE TRAINED NONLINEARLY-SCALED BAYES CLASSIFIER ON SOLVATION SITE PREDICTION DATA

	Predicted	
Observed	Non-solvated	Solvated
Non-solvated	62.7%	37.3%
Solvated	32.6%	67.4%

the information contained in the AHP feature is redundant with information that can be obtained from other features in combination for purposes of solvent site classification.

The process of fitting water molecules to electron density data can be imprecise and somewhat interpretive when density is smeared or blurred, when coordinated water molecules exhibit significant thermal mobility, and when the lattice structure of the protein crystal induces solvent binding sites that would not otherwise be favorable in a globular protein structure. Because of these and other factors, accuracy in solvent site prediction, which is based on training from and comparison with crystallographic structures, is something of a relative measure.

Current algorithms for predicting the locations of bound water molecules can be divided into two classes. Theoretical approaches, such as GRID [49], use a potential energy function to evaluate the favorability of a probe site. For GRID, the potential energy function includes terms for Lennard-Jones interactions, electrostatic interactions, and a detailed evaluation of potential hydrogen bonds. For all theoretical approaches, the relative contribution of the terms in the potential energy function must be determined before solvation sites can be predicted.

In contrast, empirical methods determine the favorability of a solvation site by analogy with known sites. A site is evaluated and compared to a database of known solvation sites and nonsolvated sites, and predicted as being more similar to one than the other. A set of features to observe and compare between solvation sites and nonsolvated sites must be selected prior to classification. Several empirical methods for prediction of protein-bound water molecule locations have been developed. AQUARIUS2 [50] uses a knowledge base of the distributions of water molecules around polar atoms at the protein surface. A “likelihood” value is assigned to a putative water molecule location based on its geometric relationship to nearby polar protein atoms. If the site lies in a region highly occupied by water molecules in the knowledge base and has significant electron density, as determined by X-ray crystallography, it receives a higher score. The highest scoring positions in a 3-dimensional matrix surrounding the protein are selected as likely water molecule locations.

AUTO-SOL [51] predicts water sites based on the directionality of hydrogen bonds. A database of small-molecule crystal structures was analyzed to find the distribution of hydrogen-bond lengths and directions, and possible solvent sites are evaluated by AUTO-SOL according to how well their hydrogen-bond geometry matches this database. Current methods can reproduce $\sim 70\%$ of crystallographically observed solvent molecules within 1.5 Å of the experimental locations [51]. There remains, however, a tendency for many current methods to produce false positives, predicting solvation sites where none are observed in the crystal structure.

The EC-Bayes classifier obtains similar accuracy to these methods, while maintaining tight balance between the number of false-positives and false-negatives. Table VI demonstrates the class balance for cross-validation testing of the trained classifier.

VI. CONCLUSION

A key advantage of the discriminant function classifier over the nearest neighbor methods is the gain in computational efficiency obtained by estimating the class-conditional feature value distributions based on the training data, rather than storing every training sample and performing an all-pairs search for near neighbors for each test sample. While the experiments here were all executed for a fixed number of EC generations, it would be worthwhile to run several experiments constrained instead by wall-clock time. In this way, the efficiency advantage of the discriminant function-based classifier might be translated into further gains in classification accuracy relative to the near-neighbor methods.

The nonlinear discriminant function classifier, in conjunction with the EC feature extraction method, seems to exhibit the best feature selection capability of all the classifiers evaluated. In several cases, however, the additional reduction in the number of features, as compared to the GA/knn classifier, incurred a slight cost in terms of classification accuracy. This slight cost, however, is offset by the significant reduction in run time required for the EC-Bayes algorithm’s computation.

Various EC-hybrid classifiers have been demonstrated to be effective techniques for identifying the physical and chemical determinants of protein-water binding [52]. With its combined classification and feature selection capability, the nonlinear Bayes classifier proves to be an effective tool for mining this and other large structural biology data sets with significant savings in computation time.

The promise exhibited by the nonlinear Bayes classifier on biological data sets suggests several avenues for further investigation. One possible improvement would be to include the prior probabilities for each class on the EC chromosome for optimization. Intuitively, this might allow the hybrid classifier more ability to maintain control over the balance in predictive accuracy among classes, even when there is disparity in the number of training and tuning samples available for each class. Initial experimentation in this area, however, suggested that inclusion of the prior probabilities on the chromosome can exacerbate the problem of overfitting the training and tuning data, resulting in poor performance on independent test data.

ACKNOWLEDGMENT

The authors would also like to thank E. Goodman and R. Enbody for their assistance and critical feedback.

REFERENCES

[1] M. L. Raymer, P. C. Sanschagrin, W. F. Punch, S. Venkataraman, E. D. Goodman, and L. A. Kuhn, “Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest-neighbors genetic algorithm,” *J Mol. Biol.*, vol. 265, pp. 445–464, 1997.

[2] M. L. Raymer, W. F. Punch, F. D. Goodman, P. C. Sanschagrin, and L. A. Kuhn, “Simultaneous feature scaling and selection using a genetic algorithm,” in *Proc. 7th Int. Conf Genetic Algorithms (ICGA)*, T. Bäck, Ed., San Francisco, CA, 1997, pp. 561–567.

[3] M. L. Raymer, W. F. Punch, F. D. Goodman, L. A. Kuhn, and A. K. Jain, “Dimensionality reduction using genetic algorithms,” *IEEE Trans. Evol. Comput.*, vol. 4, pp. 164–171, July 2000.

[4] C. S. Poornima and P. M. Dean, “Hydration in drug design. 1. Multiple hydrogen-bonding features of water molecules in mediating protein-ligand interactions,” *J. Comput. Aided Mol. Design*, vol. 9, pp. 500–512, 1995.

[5] L. A. Kuhn, M. A. Siani, M. F. Pique, C. L. Fisher, E. D. Getzoff, and J. A. Tainer, “The interdependence of protein surface topography and bound water molecules revealed by surface accessibility and fractal density measures,” *J. Mol. Biol*, vol. 228, pp. 13–22, 1992.

[6] C. S. Poornima and P. M. Dean, “Hydration in drug design. 2. Influence of local site surface shape on water binding,” *J. Comput. Aided Mol. Design*, vol. 9, pp. 513–520, 1995.

[7] E. N. Baker and R. E. Hubbard, “Hydrogen bonding in globular proteins,” *Prog. Biophys. Mol. Biol*, vol. 44, pp. 97–179, 1984.

[8] T. Bayes, “An essay toward solving a problem in the doctrine of chances,” *Phil. Trans. Roy. Soc.*, vol. 53, 1763.

[9] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[10] P. Domingos and M. Pazzani, “Beyond independence: Conditions for the optimality of the simple bayesian classifier,” in *Proc. 13th Int. Conf. Machine Learning*, L. Saitta, Ed., San Francisco, CA, 1996, pp. 105–112.

[11] E. T. Jaynes, “Prior probabilities,” *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, pp. 227–241, 1968.

[12] A. K. Jain and B. Chandrasekaran, “Dimensionality and sample size considerations in pattern recognition in practice,” in *Handbook Statistics*, P. R. Krishnaiah and L. N. Kanal, Eds. North-Holland, The Netherlands, 1982, vol. 2, pp. 835–855.

[13] G. V. Trunk, “A problem of dimensionality: A simple example,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI–1, pp. 306–307, 1979.

[14] T. M. Cover and J. M. V. Campenhout, “On the possible orderings in the measurement selection problem,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 657–661, 1977.

[15] P. M. Narendra and K. Fukunaga, “A branch and bound algorithm for feature subset selection,” *IEEE Trans. Comput.*, vol. C-26, pp. 917–922, 1977.

[16] A. Whitney, “A direct method of nonparametric measurement selection,” *IEEE Trans. Comput.*, vol. C-20, pp. 1100–1103, 1971.

[17] P. Pudil, J. Novovicova, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognit. Lett.*, vol. 15, pp. 1119–1125, Nov. 1994.

[18] A. K. Jain and D. Zongker, “Feature selection: Evaluation, application, and small sample performance,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 153–158, Feb. 1997.

[19] J. Mao, K. Mohiuddin, and A. K. Jain, “Parsimonious network design and feature selection through node pruning,” in *Proc. Intl. Conf Pattern Recognition*, Jerusalem, Israel, October 1994, pp. 622–624.

[20] K. Nozaki, H. Ishibuchi, and H. Tanaka, “Adaptive fuzzy rule-based classification systems,” *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 238–250, Aug. 1996.

[21] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, “Selecting fuzzy if-then rules for classification problems using genetic algorithms,” *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 260–270, Aug. 1995.

[22] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, pp. 81–106, 1986.

[23] ——, “Simplifying decision trees,” *Int. J. Man-Mach. Stud.*, pp. 221–234, 1987.

[24] W. Siedlecki and J. Sklansky, “A note on genetic algorithms for large-scale feature selection,” *Pattern Recognit. Lett.*, vol. 10, pp. 335–347, 1989.

[25] T. M. Cover and P. F. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21–27, Jan. 1967.

[26] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland, and R. Enbody, “Further research on feature selection and classification using genetic algorithms,” in *Proc. Int. Conf. Genetic Algorithms 93*, 1993, pp. 557–564.

[27] J. D. Kelly and L. Davis, “Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm,” in *Proc. 4th Int. Conf. Genetic Algorithms Their Applications*, 1991, pp. 377–383.

[28] J. Yang and V. Honavar, “Feature subset selection using a genetic algorithm,” in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds. Norwell, MA: Kluwer, 1998, ch. 8, pp. 117–136.

[29] J. Yang, R. Parekh, and V. Honavar, “DistAl: An inter-pattern distance-based constructive learning algorithm,” in *Proc. Int. Joint Conf. Neural Networks*, Anchorage, AL, 1998.

[30] H. Vafaie and K. De Jong, “Evolutionary feature space transformation,” in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds. Norwell, MA: Kluwer, 1998, ch. 19, pp. 307–323.

[31] J. R. Quinlan, “The effect of noise on concept learning,” in *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonnell, and T. Mitchell, Eds. San Mateo, CA: Morgan Kaufmann, 1986, pp. 149–166.

[32] K. Fukunaga and P. M. Narendra, “A branch and bound algorithm for computing k-nearest neighbors,” *IEEE Trans. Comput.*, vol. C-26, pp. 750–753, 1975.

[33] C. L. Blake and C. J. Merz. (1998) UCI Repository of Machine Learning Databases. University of California, Irvine, Dept. Inform. Comput. Sci.. [Online]. Available: http://www.ics.uci.edu/~m1earn/IMLRepository.html

[34] P. Diaconis and B. Efron, “Computer-intensive methods in statistics,” in *Sci. Amer.*, 1983, vol. 248.

[35] G. Cestnik, I. Konenenko, and I. Bratko, “Assistant-86: A knowledge-elicitation tool for sophisticated users,” in *Progress in Machine Learning*, I. Bratko and N. Lavrac, Eds. Islamabad, Pakistan: Sigma, 1987, pp. 31–45.

[36] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, “Using the ADAP learning algorithm to forecast the onset of diabetes mellitus,” in *Proc. Symp. Computer Applications Medical Care*, 1988, pp. 261–265.

[37] S. Aeberhard, D. Coomans, and O. de Vel, “The Classification Performance of RDA,” Dept. Computer Science Dept. Mathematics Statistics, James Cook University of North Queensland, Tech. Rep. 92-01, 1992.

[38] ——, “Comparison of Classifiers in High Dimensional Settings,” Dept. Comput. Sci., Dept. Math. Stat., James Cook University of North Queensland, Australia, Tech. Rep. 92-02, 1992.

[39] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, “Classification of radar returns from the ionosphere using neural networks,” in *Proc. Johns Hopkins APL Tech. Dig.*, vol. 10, 1989, pp. 262–266.

[40] S. Weiss and I. Kapouleas, “An empirical comparison of pattern recognition, neural nets, and machine learning classification methods,” in *Proc. 11th Int. Joint Conf. Artificial Intelligence*, N. S. Sridharan, Ed., Detroit, MI, 1989, pp. 781–787.

[41] ——, *An Empirical Comparison of Pattern Recognition Neural Nets, and Machine Learning Classification Methods*. San Mateo, CA: Morgan Kaufmann, 1990.

[42] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazurus, “Inductive knowledge acquisition: A case study,” in Proc. 2nd Australian Conf. Applications Expert Systems, Sydney, Australia, 1986.

[43] A. Marchand, F. V. Lente, and R. Galen, “The assessment of laboratory tests in the diagnosis of acute appendicitis,” *Amer. J. Clin. Pathol.*, vol. 80, no. 3, pp. 369–374, 1983.

[44] E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, *Protein Data Bank*. Bonn/Cambridge/Chester: Data Comm. Int. Union Crystallography, 1987, pp. 107–132.

[45] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, “The protein data bank: A computer-based archival file for macromolecular structures,” *J. Mol. Biol.*, vol. 112, pp. 535–542, 1977.

[46] U. Hobohm, M. Scharf, R. Schneider, and C. Sander, “Selection of representative protein data sets,” *Protein Sci.*, vol. 1, pp. 409–417, 1992.

[47] L. A. Kuhn, C. A. Swanson, M. E. Pique, J. A. Tainer, and E. D. Getzoff, “Atomic and residue hydrophilicity in the context of folded protein structures,” *Proteins: Str. Funct. Genet.*, vol. 23, pp. 536–547, 1995.

[48] M. L. Connolly, “Solvent-accessible surfaces of proteins and nucleic acids,” *Science*, vol. 221, pp. 709–713, 1983.

[49] R. C. Wade, K. J. Clark, and P. J. Goodford, “Further developments of hydrogen bond functions for use in determining energetically favorable binding sites on molecules of known structure,” *J. Med. Chem.*, vol. 36, pp. 140–147, 1993.

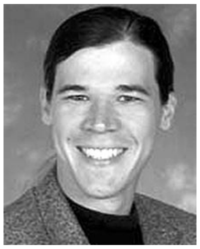
[50] W. R. Pitt, J. Murray-Rust, and J. M. Goodfellow, “AQUARIUS2: Knowledge-based modeling of solvent sites around proteins,” *J. Comp. Chem.*, vol. 14, no. 9, pp. 1007–1018, 1993.

[51] A. Vedani and D. W. Huhta, “An algorithm for the systematic solvation of proteins based on the directionality of hydrogen bonds,” *J. Am. Chem. Soc.*, vol. 113, pp. 5860–5862, 1991.

[52] M. L. Raymer, D. Holstius, and L. A. Kuhn, “Identifying the determinants of favorable solvation sites,” Protein Eng., 2001, submitted for publication.

Michael L. Raymer (M'03) received the B.S. degree in computer science from Colorado State University, Fort Collins, in 1991, the M.S. degree in computer science from Michigan State University, East Lansing, in 1995, and the Ph.D. degree in computer science from Michigan State University in 2000.

He joined the Department of Computer Science and Engineering at Wright State University, Dayton, OH as an Assistant Professor in 2001, where he is also a member of the Biomedical Sciences Program faculty. His research interests include evolutionary computation, pattern recognition, computational biology, protein structure and function, and bioinformatics. He is cofounder of the company Forensic Bioinformatic Systems, which performs computational analysis of forensic DNA evidence.



Travis E. Doom (SM'03) received the M.Sc. and Ph.D. degrees from Michigan State University, East Lansing.

He is Assistant Professor at Wright State University (WSU), Dayton, OH in the Department of Computer Science and Engineering. He also holds positions in the Department of Electrical Engineering and is a member of the graduate faculty in Biomedical Sciences. He is codirector of the bioinformatics research group at WSU and a cofounder of Forensic Bioinformatics Services, a company which performs computational analysis of forensic DNA evidence. His primary areas of teaching and research interest include digital systems, bioinformatics, computer architecture and operating systems, design automation, and computational mathematics/theory.

Leslie A. Kuhn received the B.A. degree in computer mathematics and the Ph.D. degree in biophysics from University of Pennsylvania, Philadelphia.

She studied under J. S. Leigh, Jr. developing new methods for predicting protein transmembrane segments and their packing within the membrane. Her postdoctoral research with J. Tainer and E. Getzoff at The Scripps Research Institute focused on modeling the atomic structures of proteins and peptides, analyzing the interactions of ligands with proteins, and designing proteins that bind new ligands. She is now an Associate Professor in the Department of Biochemistry at Michigan State University, East Lansing. She directs the Protein Structural Analysis and Design Lab. Her research group focuses on developing new computational techniques for modeling protein-ligand interactions, designing new ligands to proteins, and predicting the influence of hydration and flexibility in molecular recognition.

William F. Punch received the B.S. degree in biochemistry in 1979 and the Ph.D. degree in computer science in 1989 from Ohio State University, Columbus.

He joined the Department of Computer Science and Engineering at Michigan State University, East Lansing, where he is currently Associate Professor. His research interests include the theory and application of evolutionary computation, data mining, computational biology, and knowledge-based systems. He is codirector of the Genetic Algorithms Research and Application Group (GARAGe) and a founding member of the Center for Biological Modeling (CBM), both at Michigan State.