GA-Facilitated Knowledge Discovery and Pattern Recognition Optimization Applied to the Biochemistry of Protein Solvation

Michael R. Peterson, Travis E. Doom, and Michael L. Raymer

Department of Computer Science and Engineering, Wright State University, Dayton, OH 45345 {mpeterso,doom,mraymer}@cs.wright.edu

Abstract. The authors present a GA optimization technique for cosine-based k-nearest neighbors classification that improves predictive accuracy in a class-balanced manner while simultaneously enabling knowledge discovery. The GA performs feature selection and extraction by searching for feature weights and offsets maximizing cosine classifier performance. GA-selected feature weights determine the relevance of each feature to the classification task. This hybrid GA/classifier provides insight to a notoriously difficult problem in molecular biology, the correct treatment of water molecules mediating ligand binding to proteins. In distinguishing patterns of water conservation and displacement, this method achieves higher accuracy than previous techniques. The data mining capabilities of the hybrid system improve the understanding of the physical and chemical determinants governing favored protein-water binding.

1 Introduction

Computational pattern recognition has proven to be a valuable tool in the analysis of biological data. Generally, objects are gouped into classes (such as diseased and healthy cells), and then characterized according to a variety of features. Feature selection facilitates classification by removing non-salient features. Even features providing some useful information may reduce accuracy when there are a limited number of training points available [1]. This "curse of dimensionality", along with the expense of measuring additional features, motivates feature dimensionality reduction. Though no known deterministic algorithm finds the optimal feature set for a classification task, a wide range of feature selection algorithms may find near-optimal feature sets [2].

The accuracy of some types of classification rules, such as k-nearest neighbors, improves by multiplying the value of each feature by a value proportional to its usefulness in classification. The assignment of weights to each feature as a form of feature extraction improves classifier accuracy over the knn classifier alone, and aids in the analysis of large datasets by isolating combinations of salient features [3]. Through use of a bit-masking feature vector, GAs have successfully performed feature selection in combination with a $k$nn classifier [4]. This approach has been expanded for feature extraction [3,5] by searching for an ideal

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 426–437, 2004.

set of feature weights. Prior to classification, each feature value is multiplied by normalized values of GA-identified weights. The hybrid GA/$k$nn classifier described in [6] combines feature masking and feature weighting to simultaneously perform feature selection and extraction. The GA employs a weight vector for extraction and a mask vector for selection, allowing the GA to test the effect of completely eliminating a feature from consideration without reducing its associated weight completely to zero. The GA fitness function rewards smaller feature sets, leading to a tendency to mask features prematurely and not reintroduce them when appropriate.

Here, we present a novel hybrid GA/$k$nn system that eliminates the mask vector and instead employs a population-adaptive mutation technique allowing for improved simultaneous feature selection and extraction on the weight vector. Additionally, a cosine similarity measure replaces the traditional Euclidian distance metric for $k$nn classification. Cosine similarity is an effective similarity measure for diverse applications, including document classification [7] and gene expression profiling [8]. As with Euclidian distance, $k$nn classifiers employing cosine similarity may achieve improved classification through careful adjustment of feature weights [9]. Furthermore, the cosine similarity measure allows for a novel form of GA optimization by searching for an optimal set of feature offsets. These offsets affect the cosine of the angles between various data points considered by the $k$nn classifier, and thus may be optimized. In some cases, cosine similarity may be less prone to errors in attribute measurements. Euclidian distance is highly dependent upon the magnitude of measured attributes, since fluctuations in magnitude will directly affect the calculated distance between points. In contrast, the cosine measure depends more on the overall shape of the data distribution than on feature magnitude. Thus, in cases where the magnitude of features measured across experiments can vary, as in many biological experiments, cosine similarity is less susceptible to noise-induced error than Euclidian distance metrics [8].

This hybrid GA/$k$nn system provides new insight into the role of water molecules during the binding of drugs or other ligands to the protein surface. Protein surface-bound water molecules often form hydrogen bonds to a docking drug or other ligand, and are an essential part of the protein surface with respect to ligand screening, docking, and design [10]. It is thus important to identify the areas of the protein surface where water molecules will not be displaced upon ligand binding. However, the identification of favorable protein surface sites for solvent binding has proven difficult, in part because the majority of protein surface residues are hydrophilic.

Among the various attempts to treat water molecules during ligand binding, the *Consolv* system [10] employs a (GA/$k$nn) classifier to distinguish water molecules bound in the protein's ligand-binding site from those displaced upon ligand binding. *Consolv* improved on previous solvation techniques including *Auto-Sol* [11] and *AquariusII* [12]. The hybrid GA/classifier described here improves upon *Consolv's* reported accuracy while mining feature weights to aid in understanding the properties governing protein-water interactions.

2 Methods

2.1 Cosine-Based *K*nn Classification

The hybrid GA/classifier described here employs k-nearest neighbors classification. Unlike many common learning algorithms, $k$nn techniques do not construct an explicit description of the target function when a training set is provided. These algorithms only generalize beyond the training points when a query point is presented. Based upon the assumption that the classification of an unseen data point will be most similar to that of training points that share similar attributes, $k$nn approximates the target function over a small neighborhood of training points most similar to each test point.

When selecting the most similar neighbors, it is important to employ an appropriate similarity measure. There are several available for $k$nn classifiers; the most common of which is the Euclidean distance between two points within d-dimensional space, where d is the number of measured attributes. Another is the cosine of the angle between two vectors, each representing a data point within d-dimensional space. In addition to these measures, any other distance metric, such as Mahalanobis distance, may also be employed.

The $k$nn classifier described in this work employs cosine similarity as defined below. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are attribute vectors representing two data points, then the cosine of the angle between them is defined as

$$cos(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}{\|\boldsymbol{x}_i\| \|\boldsymbol{x}_j\|} \tag{1}$$

where "$\cdot$" represents the dot product between the two vectors, and $\|\boldsymbol{x}_j\|$ represents vector length. Larger cosine values represent a greater degree of similarity between vectors. When taking the cosine similarity between a query point and all training points, the k points with the largest similarity values are the nearest neighbors.

After neighbor identification, a class is assigned to the query point. Unlike traditional $k$nn classification, the cosine-based $k$nn classifier described here does not use a simple voting scheme for class assignment. Classification occurs using a weighted scheme based on how similar each neighbor is to the query point. If the data contains only two classes, then the query point $\boldsymbol{x}$ is classified by the value of the measure q[13]:

$$q = \sum_{i=1}^{n} cos(\boldsymbol{x}_i, \boldsymbol{x}) c(\boldsymbol{x}_i) \tag{2}$$

where

$$c(\boldsymbol{x}_i) = \begin{cases} 1 & : \text{ if } \boldsymbol{x}_i \in \text{ the positive class} \\ -1 & : \text{ otherwise} \end{cases}$$

If q is positive, then the query point is assigned to the positive class, otherwise it is assigned to the negative class. For problems with more than two classes,

a seperate q function can be applied for each class, with the largest resulting function representing the class label applied to the query point.

In addition to feature weight evolution, the accuracy of the cosine classifier can be further improved by transformation of the coordinate space. Typically, the angle between two vectors is determined relative to the origin within the feature space. If the data is shifted by different amounts in each feature dimension, then the relative angle between any two given points changes. By shifting the origin within the feature space, a GA can improve the classification of new data. As demonstrated in Figure 1, this shifting may change the assigned class label for a test point. Figure 1(a) illustrates the behavior of an unshifted $k = 5$ nearest neighbor classifier in two-dimensional feature space. Here, no offset is applied. Among the points with the highest angular similarity to the test pattern, three belong to class 1, and 2 belong to class 2. The test pattern is labeled as belonging to class 1 since the sum of the cosine of angles to class 1 points is larger than that of class 2 points. In (b), the origin is shifted, thus changing the point of reference. Now, all of the nearest neighbors in terms of cosine similarity belong to class 2, so the test point is labeled as belonging to class 2. The hybrid GA/classifier system described here optimizes cosine $k$nn classifiers with respect to feature weights, feature offsets, and the k-value.

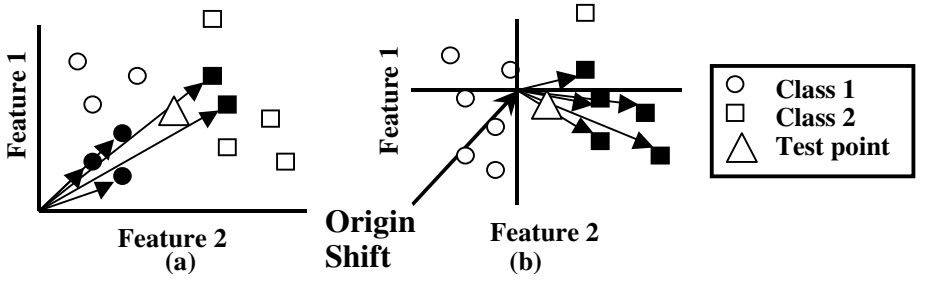

Fig. 1. Effect of the origin position on cosine-based $k$nn classification

2.2 Use of the Genetic Algorithm

Our goal is to provide a pattern recognition technique that improves classification accuracy while enabling knowledge discovery. To that end, an optimization technique should preserve the independence of features; that is, it must not define new features as combinations of the originals. In this manner, the influence of each feature upon the performance of the optimized classifier may easily be examined, thereby providing useful insight into the nature of the given problem.

The GA used here employs the chromosome shown in Figure 2. For an n-dimensional dataset, the first n genes on the chromosome represent the unnormalized real-valued weights of each feature. Weights evolve on the interval [0.0 ... 100.0]. The next n genes represent real-valued feature offsets for each feature. Prior to classification, each feature in a dataset is normalized by sum to the

interval [1.0 ... 10.0]. Typically, offsets are permitted to evolve on the interval [-15.0 ... 30.0] so that the classifier may view a feature either from within or from outside of its range. During the training process, the GA learns appropriate reference points by shifting offsets. For cases where classifiers are trained only through weight and k-value optimization, the offsets are simply ommited from the chromosome. The final gene is an integer representing the k-value for the $k$nn classifier. Typically, k may take any value over [1 ... 100], though for small datasets a more limited range may be specified.

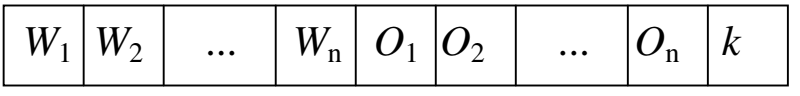

Fig. 2. Structure of the GA chromosome

In previous work involving GA optimization of $k$nn classifiers, dimensionality reduction is accomplished by employing feature masks using bit sets on the chromosome to explicitly perform feature selection [6]. The seperation of feature extraction and selection on the chromosome introduces the possibility that a partially relevant feature may be masked prematurely. If the GA gives preference to chromosomes masking a larger number of features, then a prematurely masked feature may never be unmasked. To avoid this, dimensionality reduction is accomplished using population-adaptive mutation. In the absence of an explicit feature mask, the weight of a feature must be reduced to zero in order to remove the feature from consideration. Population-adaptive mutation increases the likelihood that the weights of spurious features will be reduced to zero. When a gene is selected for mutation, the value of the gene is shifted randomly within a range depending upon the current variance of the gene across the GA population. The new value is randomly chosen from a Gaussian distribution with a mean equal to the gene's current value and a standard deviation based on the feature variance across the population. Under population-adaptive mutation, a gene may mutate either above or below its permitted range. In such cases, the minimum or maximum value is set as the new gene value. Thus, mutation may easily cause a feature weight to be set to zero due to this boundary effect, effectively removing that feature from consideration. As a feature weight gradually decreases across the population, the probability that the feature will be masked increases. Conversely, features with generally high weights are unlikely to become masked under population-adaptive mutation, thus preventing premature masking. Because population-adaptive mutation easily performs both feature selection and extraction, it is especially useful for mining relevant meaning from

the remaining feature weights, which indicate the relevance of selected features for a given classification task.

Selection operations are implemented using tournament selection with a tournament of size 2. Recombination is implemented using uniform crossover with a crossover probability of 0.5 per gene. Because population-adaptive mutation largely drives dimensionality reduction, the GA uses a fairly high mutation rate of 0.1 mutations per gene. The population size is typically 50 or 100 chromosomes. The GA runs either to convergence or for 200 generations, whichever occurs first. During training, chromosomes are evaluated by applying their weight and offset vectors, as well as the k-value, to the feature set by performing classification on a set of patterns of known class using a cosine-based $k$nn classifier. The fitness function contains components measuring overall classification accuracy, the balance of accuracy among classes, and the number of features employed. The fitness function gives preference to chromosomes providing high, balanced accuracy using as few features as possible. The GA-minimized cost function is:

$$\begin{aligned} cost(\boldsymbol{w}, k) = {} & C_{pred} \times \% \text{ of incorrect predictions} \\ & + C_{mask} \times \# \text{ of unmasked features} \\ & + C_{bal} \times \text{class accuracy difference} \end{aligned}$$

where C_{pred}, C_{mask}, and C_{bal}, are the cost function coefficients. For the authors' experiments the empirically derived values: $C_{pred} = 25.0$, $C_{bal} = 10.0$, and $C_{mask} = 1.0$ are used. The function gives highest preference to maintaining high overall accuracy, with balanced accuracy as a secondary goal. The number of features employed recieves a relatively small coefficient in order to prevent the GA from prematurely removing features from consideration.

For each experiment, data patterns are randomly split into class-balanced training and test sets, with the remaining points withheld for bootstrap validation upon the completion of GA training. At the completion of each GA experiment, the quality of the optimized classifier is assesed using a variant of the bootstrap test method [14] in order to obtain an unbiased accuracy measurement as well as a simple measure of this measurement's variance. The bootstrap test helps ensure that reported accuracies are not the result of GA overfitting of the test data.

2.3 Experiments on Biological Datasets

In order to demonstrate the utility of offset optimization, experiments are performed using a dataset containing diabetes diagnostic information for Native Americas of Pima heritage [15]. The dataset consists of diagnostic information for 768 adult women, 500 of whom tested negative for diabetes and 268 of whom tested positive. Six of the eight features representing clinical measurements are quantitative and continuous. The remaining two features are quantative and discrete. There are no missing feature instances. This dataset is suitable for testing the ability of a GA/classifier system to simultaneously extract features and boost accuracy due to its moderate dimensionality, its completeness, and

its unbalanced class representation. This dataset is available from the UCI machine learning repository [16]. For comparison purposes, results for this dataset using a number of well-known classification techniques, as implemented in the Waikato Environment for Knowledge Analysis (WEKA) data mining software package [17] are also presented. Results from WEKA classifiers reflect accuracy after 10-fold cross-validation.

To demonstrate the knowledge discovery abilities of the hybrid GA/classifier, experiments are performed on two datasets describing protein-water interactions, created during development of the *Consolv* system. The first dataset describes the environments of water molecules bound to protein surfaces. Water molecules in this set belong to one of two classes: those displaced from the protein surface when a molecule (such as a drug) binds to the protein, and those conserved. When a ligand binds to a protein, it may displace water molecules at some locations and bind directly to the surface. In other locations, the ligand forms a hydrogen bond to a water molecule, which in turn forms a hydrogen bond to the protein surface, as illustrated in Figure 3. By accounting for water molecules involved in protein-ligand binding, accurate prediction of water conservation or displacement facilitates the design of ligands with higher complimentarity to the protein surface. Eight features are provided to characterize the local environment of water molecules in 30 independently solved, unrelated protein structures [6]. The chemical and physical features describing each water molecule include the number of protein atoms surrounding the water molecule (ADN), the frequency with which the types of atoms surrounding the water molecule are found to bind water molecules in another database of proteins (AHP), a measure of the thermal mobility (crystallographic temperature factor) of the water molecule (BVAL), the number of hydrogen bonds between the water molecule and the protein (HBDP), the number of hydrogen bonds to other water molecules (HBDW), and three additional normalizations on temperature factor of either the water molecule (MOB) or of its neighboring atoms (ABVAL and NBVAL). The goal of the GA/knn classifier is to distinguish conserved from displaced water molecules using a minimal set of features. Examination of the selected features and their corresponding weights found by the GA/knn classifier leads to a better understanding of the underlying physical and chemical properties salient to water binding interactions. The dataset describing water conservation and displacement consists of 5542 water molecules; 3405 conserved and 2137 displaced.

The second dataset consists of a set of all surface water molecules from the same 30 proteins, and an equal number of non-solvated probe sites, for a total of 11,084 samples. For each water molecule and probe site, all features (except for BVAL and MOB) from the first dataset are used. For this dataset, the goal of the GA/knn classifier is to distinguish solvation sites from non-sites with high accuracy using a minimal feature set. As before, the selected feature weights provide insight into the properties governing the interactions between water molecules and the protein surface.

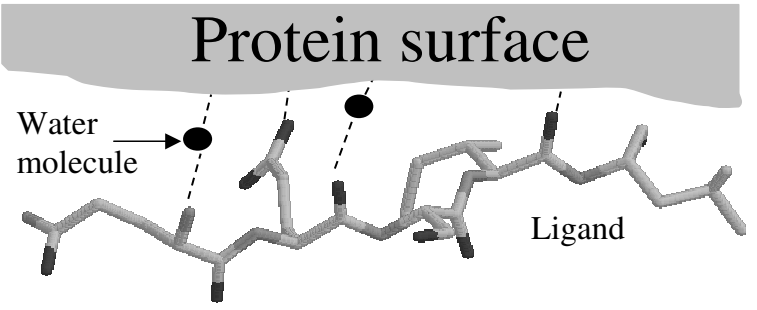

Fig. 3. Ligand-binding interface

For each GA run, the first dataset is split into class-balanced training and test datasets consisting of 1068 waters each, with all remaining waters reserved for bootstrap validation. The second dataset is split into balanced training and test sets consisting of 2128 waters each, with remaining patterns reserved for bootstrap validation. As with the diabetes dataset, results using WEKA classifiers are presented for comparison purposes. The two water datasets will be made publically available at birg.cs.wright.edu/water/.

3 Results

3.1 Offset Inclusion During Optimization

Table 1 presents the best three optimizations obtained for the Pima Diabetes dataset including offset optimization (left) and using only weight and k-value optimization (right). Results reflect bootstrap validation. When offsets are included on the GA chromosome, the classifier achieves between 75 and 77% accuracy with an accuracy balance of approximately 7% between the classes. In contrast, the GA-trained classifier is unable to achieve better than 70% accuracy with a 10% class imbalance when offsets are not optimized. For this dataset, cosine-based $k$nn classifiers clearly benifit from offset optimization. Table 2 presents the cross-validated performance of the best 3 of 18 WEKA classifiers according to both accuracy and class accuracy balance. Logistic is a regression-based classifier, SMO is a support vector machine, DecisionStump and j48 are both decision tree-based classifiers, and NeuralNetwork is a backpropagation-based neural classifier. The most accurate WEKA classifier, Logistic, achieves higher accuracy than the GA-trained classifier, but it exhibits a high classification bias toward negative labelling, with an imbalance of 31.86%. Even the most balanced WEKA classifier, DecisionStump, exhibits a high imbalance of 16.18%. The GA is able to train a cosine-based $k$nn classifier to achieve an accuracy competitive with the best WEKA classifiers without resorting to an unbalanced bias toward one class or the other.

Table 1. Pima Diabetes results, with and without offset optimization

With Offsets							Weights Only						
	Accuracy (%)							Accuracy (%)					
ID	Total	Neg	Pos	Bal	K	#F	ID	Total	Neg	Pos	Bal	K	#F
1	76.72	75.0	78.38	8.50	12	7	1	69.88	66.54	73.13	10.71	25	6
2	76.08	67.27	84.66	17.39	12	7	2	69.28	63.57	74.84	13.23	4	6
3	75.16	71.59	79.63	10.24	37	6	3	68.99	66.38	71.53	9.58	10	7

Table 2. Pima Diabetes WEKA classification results

Top 3 by Accuracy(%)					Top 3 by Balance(%)				
Classifier	Total	Neg	Pos	Bal	Classifier	Total	Neg	Pos	Bal
Logistic	77.08	88.20	56.34	31.86	DecisionStump	71.35	77.00	60.82	16.18
SMO	76.43	89.00	52.99	36.01	j48.J48	74.35	81.20	61.57	19.63
NaiveBayesSimple	75.91	83.80	61.19	22.61	NeuralNetwork	74.48	81.40	61.57	19.83

3.2 Ligand-Binding Water Conservation

The primary goal for experimental research on protein-bound water molecules is to classify whether specific water molecules on the protein surface are conserved or displaced upon ligand binding. This goal is manifested by achieving a high classification accuracy for this dataset during GA training. A secondary goal remains elucidation of the determinants of water conservation. This goal is met by examining the final relative feature weights of the various features evolved during classifier optimization. The inclusion of feature weights on the GA chromosome and the use of population-adaptive mutation for feature selection and extraction successfully yields combinations of features that provide improved distinction between conserved and displaced water molecules.

The left side of Table 3 presents the bootstrap results of the three best optimizations for the water conservation dataset. For comparison, the three best WEKA classifier results in terms of both accuracy and balance are presented on the right side of the table. While the most accurate WEKA classifiers achieve slightly higher accuracy, they all exhibit a notable bias towards the conserved class, indicating that they are unable to distinguish meaningful information in the dataset and thus resort to a preference toward the more frequently occuring class. In contrast, the GA-trained cosine $k$nn classifier achieves similar accuracy without significant bias toward either class, using as few as 4 of the 8 available features. The utility of using a GA favoring class-balanced results during optimization is clear. Section 4 discusses the biological implications of the optimized feature weights obtained for this dataset.

3.3 Solvation Site Prediction

As with the previous dataset, the goals for experiments investigating the physical and chemical determinants of solvation sites on the protein surface are two-fold.

Table 3. Water conservation results, GA (left) and WEKA classifiers (right)

Bootstrap Accuracy (%)					Feature Weights, Offsets			Top 5 WEKA Classifiers by Accuracy				
Total	Disp	Cons	Avg Bal	K	ADN	AHP	BVAL	Classifier	Total (%)	Disp	Cons	Bal
65.29	66.57	64.00	4.10	48	-	-	.426, 4.70	NeuralNetwork	66.62	44.17	80.70	36.53
64.76	63.91	65.61	3.89	29	-	.096, 4.321	.281, 2.29	j48.J48	66.02	37.06	84.20	47.14
64.31	63.52	65.10	3.61	26	-	-	.207, 1.08	ADTree	65.97	44.27	79.59	35.32
Feature Weights, Offsets								Top 5 WEKA Classifiers by Balance				
Total		HBDP	NBVAL		HBDW	MOB	ABVAL	Classifier	Total (%)	Disp	Cons	Bal
65.286		-	.101, 3.68		-	.406, -.51	.067, -12.94	IB1	61.35	48.62	69.34	20.72
64.762		-	.078, 2.63		.115, 4.85	.238, 2.88	.192, -13.67	KernelDensity	61.30	47.73	69.81	22.08
64.309		.193, -14.96	-		.281, -9.48	.227, -1.95	.093, -12.51	NaiveBayesSimple	64.06	49.93	72.92	22.99

The first goal is to train a classifier to accurately identify favored solvation sites given the properties of a protein surface at varying localities. The second goal to determine the relative importance of the various chemical and physical factors governing solvation. Examination of the selected features and their evolved weights within a trained classifier leads to biological insights into the properties governing protein solvation.

The left side of Table 4 presents the best three results obtained for the solvation dataset, while the right side presents the best three WEKA classifiers in terms of both classification accuracy and class balance. The best GA-trained classifier achieves a mean bootstrap accuracy of 69.91% using five of the six available features. In contrast, the best WEKA classifiers achieve similar though slightly lower accuracy than the best optimized cosine $k$nn classifier while maintaining a similar level of prediction balance. The main benefit of employing a hibrid GA/classifier system for the solvation dataset is the ability to elucidate the biological relevance of each feature through feature selection and extraction in order to form a more complete understanding of protein-water interactions.

Table 4. Water solvation results, GA (left) and WEKA classifiers (right)

Bootstrap Accuracy (%)					Weights, Offsets		Top 5 WEKA Classifiers by Accuracy				
Total	non	site	Avg Bal	K	ADN	AHP	Classifier	Total(%)	Non	Site	Bal
69.91	67.78	72.04	4.36	80	.252, -11.84	.177, -7.70	Logistic	69.33	65.50	73.16	7.67
69.48	65.79	73.16	7.38	67	.271, -8.42	.253, -15.00	NeuralNetwork	69.29	66.00	72.58	6.58
69.42	64.38	74.47	10.08	83	.242, -5.36	.222, 9.70	VotedPerceptron	69.25	66.75	71.74	4.98
Feature Weights, Offsets							Top 5 WEKA Classifiers by Balance				
Total		HBDP	HBDW		ABVAL	NBVAL	Classifier	Total(%)	Non	Site	Bal
69.91		.352, -11.77	.105, -1.31		-	.113, 2.75	IB1	63.56	63.45	63.68	0.23
69.48		.154, -14.90	.113, -15.0		-	.209, -12.89	j48.J48	68.99	68.75	69.24	0.49
69.31		.205, -5.86	.165, 15.00		-	.166, 9.64	j48.PART	68.03	68.56	67.49	1.06

4 Discussion

Results obtained for the Pima diabetes dataset demonstrate the utility of optimizing offsets in addition to feature weights for a cosine-based $k$nn classifier.

Evolution of both feature weights and offsets provides the GA an opportunity for classifier optimization that cannot be leveraged in Euclidian distance-based $k$nn classifiers. GA optimization of cosine-based classifiers may significantly boost the performance of a pattern recognition system. When compared to WEKA classifiers, results indicate that the hybrid GA/classifier system described here outperforms all other tested methods in terms of simultaneously increasing classification accuracy while maintaining class balance.

The features selected by the GA and their corresponding weights provide an opportunity to mine biologically relevant information from the optimized classifier systems. Consider the resulting features obtained by the GA during the best run on the conserved water dataset. All four selected features (BVAL, MOB, ABVAL, and NBVAL) relate to the thermal mobility of a given water molecule or of its surrounding atoms. In previous research, the *Consolv* system often favored BVAL and MOB, but almost always removed the other two features from consideration. Other features, such as AHP, HBDP, and HBDW, each relating to a water molecule's surrounding atomic environment, are more frequently considered [6]. In that work, 64.2% bootstrap accuracy is the highest reported result. Here, the GA-optimized cosine classifier increases accuracy over previously published results using only variations on thermal mobility. These results suggest that most of the information necessary to determine water molecule conservation upon ligand binding may be extracted from the thermal mobility and occupancy values of the water molecule and its neighbors. While other features may be related to conservation, they may be correlated with the temperature factor in such a way that they bring no additional information to the classification problem.

For the solvation dataset, the trained classifier consistently employs all measured features execpt for ABVAL. Features such as ADN, AHP, and ABDP typically recieve higher weights. These three features each depend upon the amount and type of atoms neighboring a probe site, suggesting that the local atomic environment of a probe site is more relevant than the thermal mobility of atoms surrounding the site in determining the favorability for solvation at the given site.

While maintaining a balanced accuracy level competitive with contemporary classification techniques, the hybrid GA/cosine classifier system described here provides the ability to mine insight into the relative importance of the various features provided for a given problem. This property permits the GA/cosine classifier system to be employed in cases where traditional techniques that do not maintain feature independence would not be well-suited for knowledge discovery.

References

[1] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 306–307, 1979.

[2] H. Liu and H. Motodata, *Feature Selection for Knowledge Discovery and Data Mining*, pp. 73–95. Boston, MA: Kulwer Academic Publishers, 1998.

[3] J. D. Kelly and L. Davis, "Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm," in *Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications*, pp. 377–383, 1991.

[4] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335–347, 1989.

[5] 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. International Conference on Genetic Algorithms 93*, pp. 557–564, 1993.

[6] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 5, pp. 164–171, 2000.

[7] E. Han and G. Karypis, "Centroid-based document classification: Analysis & results," in *Principles of Data Mining and Knowledge Discovery: fourth European Conference*, pp. 424–431, 2000.

[8] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. A. Jr., and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Science*, vol. 97, pp. 262–267, 2000.

[9] E. Han, G. Karypis, and V. Kumar, "Text categorization using weight adjusted k-nearest neighbor classification," in *Advances in Knowledge Discovery and Data Mining: fifth Pacific-Asia Conference*, pp. 53–65, 2001.

[10] 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.

[11] 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.

[12] 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.

[13] M. Kuramochi and G. Karypis, "Gene classification using expression profiles: a feasibility study," in *Proceedings of the Second Annual IEEE International Symposium on Bioinformatics and Bioengineering*, pp. 191–200, 2001.

[14] A. K. Jain, R. C. Dubes, and C. C. Chen, "Bootstrap techniques for error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 628–633, Sept. 1987.

[15] 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 *Proceedings of the Symposium on Computer Applications and Medical Care*, pp. 261–265, IEEE Computer Society Press, 1988.

[16] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases." University of California, Irvine, Dept. of Information and Computer Sciences, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[17] I. H. Witten and E. Frank, *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*, pp. 265–319. San Francisco, GA: Morgan Kaufmann, 2000.