9-2005

GA-Facilitated KNN Classifier Optimization with Varying Similarity Measures

Michael R. Peterson
Wright State University - Main Campus

Travis E. Doom
Wright State University - Main Campus, travis.doom@wright.edu

Michael L. Raymer
Wright State University - Main Campus, michael.raymer@wright.edu

Follow this and additional works at: <https://corescholar.libraries.wright.edu/knoesis>

Part of the [Bioinformatics Commons](#), [Communication Technology and New Media Commons](#), [Databases and Information Systems Commons](#), [OS and Networks Commons](#), and the [Science and Technology Studies Commons](#)

Repository Citation

Peterson, M. R., Doom, T. E., & Raymer, M. L. (2005). GA-Facilitated KNN Classifier Optimization with Varying Similarity Measures. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 3, 2514-2521.
<https://corescholar.libraries.wright.edu/knoesis/925>

This Conference Proceeding is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

GA-Facilitated *KNN* Classifier Optimization with Varying Similarity Measures

Michael R. Peterson
Dept. of CS & Egr.
Wright State University
Dayton, OH 45440
peterson.7@wright.edu

Travis E. Doom
Dept. of CS & Egr.
Wright State University
Dayton, OH 45440
doom@cs.wright.edu

Michael L. Raymer
Dept. of CS & Egr.
Wright State University
Dayton, OH 45440
mraymer@cs.wright.edu

Abstract- Genetic algorithms are powerful tools for *k*-nearest neighbors classifier optimization. While traditional *knn* classification techniques typically employ Euclidian distance to assess pattern similarity, other measures may also be utilized. Previous research demonstrates that GAs can improve predictive accuracy by searching for optimal feature weights and offsets for a cosine similarity-based *knn* classifier. GA-selected weights determine the classification relevance of each feature, while offsets provide alternative points of reference when assessing angular similarity. Such optimized classifiers perform competitively with other contemporary classification techniques.

This paper explores the effectiveness of GA weight and offset optimization for knowledge discovery using *knn* classifiers with varying similarity measures. Using Euclidian distance, cosine similarity, and Pearson correlation, untrained classifiers are compared with weight-optimized classifiers for several datasets. Simultaneous weight and offset optimization experiments are also performed for cosine similarity and Pearson correlation. This type of optimization represents a novel technique for maximizing Pearson correlation-based *knn* performance. While unoptimized cosine and Pearson classifiers often perform worse than their Euclidian counterparts, optimized cosine and Pearson classifiers typically show equivalent or improved performance over optimized Euclidian classifiers. In some cases, offset optimization provides further improvement for *knn* classifiers employing cosine similarity or Pearson correlation.

1 Introduction

Computational pattern recognition is a valuable tool in many fields of scientific inquiry. Objects are grouped into known classes (such as diseased or healthy tissue samples) and then characterized according to a set of measured features. Feature selection facilitates classification by removing non-salient features. Even features providing useful information may reduce classification accuracy when there are a limited number of training points [15]. This “curse of dimensionality”, motivates dimensionality reduction. Though no known deterministic algorithm finds the optimal set, a wide range of algorithms may find near-optimal feature sets [9].

The accuracy of some classification rules, such as *k*-nearest neighbors employing Euclidian distance, improves

by multiplying the value of each feature by a value proportional to its usefulness in classification. As a method of feature extraction, applying weights to features in proportion to usefulness improves *knn* accuracy and aids in the analysis of large datasets by isolating combinations of salient features [6]. Through use of a bit-masking vector, GAs have performed feature selection in combination with a *knn* classifier [14]. The GA chromosome includes one bit per feature, indicating whether to include the feature during classification, and assesses accuracy. Later works expand this approach for feature extraction [6, 11] by searching for an ideal set of feature weights. Prior to classification, the values of each feature are multiplied by normalized values of GA-identified weights. The hybrid GA/*knn* classifier described in [13] combines feature masking and weighting to simultaneously perform feature selection and extraction. The GA employs a weight vector for extraction and a mask vector for selection on its chromosome to test the effect of completely eliminating a feature from consideration without reducing its associated weight to zero. The objective function rewards smaller sets, which may lead to a tendency to mask features prematurely and not reintroduce them when appropriate.

More recently, cosine similarity has been employed as an alternative to Euclidian distance for *knn* classification. Cosine similarity has been demonstrated as a useful similarity measure for a diverse set of applications, including gene expression profiling [2] and document classification [3]. Careful adjustment of feature weights may improve the performance of classifiers employing cosine similarity [4], which assesses the angular closeness of two feature vectors, taken relative to a point of reference (i.e. the origin). Changing this point of reference changes the similarity between vectors and thus affects the performance of a cosine-based *knn* classifier. Peterson et. al. [10] employ a GA to simultaneously optimize feature weights and the point of reference (i.e. feature offsets) for cosine-based *knn*. By shifting offsets, the GA searches for an optimal point of reference to assess the angular similarity between patterns. Their GA eliminates the bit-masking scheme used by Raymer et. al. [13], and instead uses a population-adaptive mutation scheme to quickly eliminate non-salient features from classifier consideration. They report classification results highly competitive with contemporary classification techniques including support vector machines, feed-forward neural networks, and decision tree algorithms.

The authors present a novel form of optimization for *knn* classifiers employing Pearson correlation as a similar-

ity measure. Like cosine similarity, Pearson correlation is frequently used as a similarity measure to classify cancer tissues using microarray gene expression data [2]. Pearson correlation may provide an advantage for classifying noisy data since correlation measures rely more on the overall shape of data profiles than on the exact distance between data points, as is the case with Euclidian distance. As with classifiers employing Euclidian distance and cosine similarity, careful selection of feature weights in proportion to feature saliency improves the performance of the Pearson correlation-based k nn classification rule. Additionally, applying offsets to features affects classification. Applying an additive or subtractive shift of a single measurement (i.e. feature) affects the correlation between two feature vectors.

Figure 1 demonstrates offset optimization for a cosine-based k nn classifier. In (a), the cosine values are determined using the default origin. Taking $k = 5$, the test pattern is assigned to class 1, since 3 of its 5 nearest neighbors belong to class 1. In Figure 1 (b), the origin is shifted. From the new perspective, the 5 nearest neighbors belong to class 2, so the test point's label is now 2. Allowing a GA to search for an optimal set of offsets allows maximization of within-class pattern correlation and minimization of between-class correlation, hence improving accuracy. This paper proposes to apply weight-only and weight + offset optimization to Pearson and cosine-based k nn classification to improve accuracy. To the authors' knowledge, using offset optimization represents a novel technique to maximize accuracy for Pearson-based k nn. As is the case for cosine-based k nn [10], the authors expect that weight and offset optimization will significantly increase the performance of the Pearson-based classifier by searching for an optimal linear transform of the datasets while preserving feature independence.

Figure 1: Effect of the origin position on cosine-based k nn classification.

This experiments discussed in this paper compare the

performance of GA-optimized cosine similarity and Pearson correlation-based k nn classifiers against their untrained counterparts, untrained and weight-optimized traditional Euclidian distance k nn classifiers, and one another. Comparisons use well-publicized datasets from the UCI machine learning repository [1] and the protein solvation datasets employed in [10]. Because we replicate the GA and classification techniques described in [10], comparisons with other classification techniques are omitted due to space limitations. The interested reader is instead pointed to the comparison in [10] between the optimized cosine-based classifier and other contemporary algorithms. The remainder of this paper is laid out as follows. Section 2 of this paper discusses the classifiers, the GA parameters, the fitness function, and the datasets used in this study. Section 3 compares the GA-optimized and unoptimized classifiers on the various datasets. The paper concludes with remarks on the ability of the GA to improve classification accuracy using various similarity measures and optimization techniques.

2 Methods

2.1 Euclidian Distance-based K -NN Classification

In k nn classification, training patterns are plotted in d -dimensional space, where d is the number of features present. These patterns are plotted according to their observed feature values and are labelled according to their known class. An unlabelled test pattern is plotted within the same space and is classified according to the most frequently occurring class among its k -most similar training patterns; its nearest neighbors. The most common similarity measure for k nn classification is the Euclidian distance metric, defined between feature vectors $\vec{x}$ and $\vec{y}$ as:

$$euc(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^f (x_i - y_i)^2}$$

where f represents the number of features. Smaller distance values represent greater similarity. Classification occurs after identifying the k -most similar training points to a query point. Rather than using a standard voting scheme, the algorithm here assigns class labels to query points using a weighted scheme based upon each neighbor's proximity to the query point. Following the scheme employed by [8], if the data contains only two classes, the positive and negative class, then the query point $\vec{x}$ is classified according to the measure of q :

$$q = \sum_{i=1}^n euc(\vec{x}_i, \vec{x}) c(\vec{x}_i)$$

where

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

and n represents the number of training points and $\vec{x}_i$ is the i^{th} training point. Because smaller distances represent

greater similarity, the query point is assigned to the positive class if q is negative, otherwise it is assigned to the negative class. For a multi-class problem, the classification rule is generalized to use one q -score for each class. The query point is assigned to the class with the minimum q -score. For weight optimization, given a weight vector $\vec{w}$, each feature vector $\vec{x}$ is transformed by $\vec{w}$ such that the transformed feature vector $\vec{x}' = x_i * w_i$. Euclidian distance is invariant to offset shifting, so offset optimization is not performed for the Euclidian distance k nn classifier.

2.2 Cosine Similarity-based K -NN Classification

For the k nn classifier employing the cosine similarity measure, the cosine similarity between feature vectors $\vec{x}$ and $\vec{y}$ is

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$$

where “ $\cdot$ ” represents the dot product between the vectors, and $\|\vec{z}\|$ represents vector length. Unlike Euclidian distance, larger cosine values represent a greater similarity between vectors. Classification occurs after identifying the k -most similar neighbors by using a q -score, substituting cosine similarity for Euclidian distance. Because larger values indicate greater similarity, the query point is assigned to the positive class if the q -score is positive, otherwise the negative class is chosen.

While, feature weighting affects cosine-based k nn classification, feature offset shifting provides an additional opportunity for optimization. For cosine similarity the angle between two vectors is taken relative to the origin; a GA may perform feature extraction by shifting the position of the origin relative to the data. Independently shifting each feature positively or negatively changes the angular point of reference, affecting classification. For combined weight and offset optimization, given a weight vector $\vec{w}$ and an offset vector $\vec{O}$, each feature vector $\vec{x}$ is transformed such that the transformed feature vector $\vec{x}' = (x_i - O_i) * w_i$.

2.3 Pearson Correlation-based K -NN Classification

The Pearson correlation between feature vectors $\vec{x}$ and $\vec{y}$ is

$$\text{pear}(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^f (x_i - \bar{x})(y_i - \bar{y})}{(f - 1)S_x S_y}$$

where f is the number of features, $\bar{z}$ is the mean value of vector $\vec{z}$, and S_z is the standard deviation of $\vec{z}$. The Pearson correlation coefficient measures the strength of a linear relationship between $\vec{x}$ and $\vec{y}$, and ranges from -1, indicating a strongly inverse linear relationship, up to +1, indicating a strongly positive linear relationship. A coefficient value of 0 indicates the absence of any detectable relationship between vectors. The experiments described here only consider positively correlated training patterns similar. Like the classifiers using other similarity measures, query patterns are assigned class labels according to q -scores, now using Pearson correlation.

Offset shifting provides an additional opportunity for Pearson correlation-based k nn classifier optimization. Pearson correlation treats a vector $\vec{x}$ as repeated samples of a

random variable. Shifting the value of a single sample by a positive or negative offset for both $\vec{x}$ and $\vec{y}$ affects both the mean and standard deviation of both random variables, and hence changes the Pearson correlation between the feature vectors. The GA attempts to find a global weight and offset transformation maximizing within-class correlation while minimizing between-class correlation.

2.4 The Genetic Algorithm

The GA simultaneously evolves feature weights, offsets, and a k -value for classifier optimization. While high accuracy may be achievable using a non-linear transform involving combinations of features, learning the relative classification saliency of individual features remains an optimization goal. The GA enables knowledge discovery by maintaining the relative independence of features during feature extraction by applying individual weights and offsets to each feature. Figure 2 shows the GA chromosome. $W_1 \dots W_n$, represent real-valued weights for each of the n features. $O_1 \dots O_n$, represent real-valued offsets for each feature. K represents the k -value for classification. To avoid implicit weighting of features with different ranges of values, datasets are normalized by sum to the range of [1.0 ... 10.0] prior to classification. Weights range from 0.0 to 100.0, offsets from -15.0 to 25.0, and the k -value from integers from 1 to 25, 50, or 100, depending on the dataset size.

W_1	W_2	...	W_n	O_1	O_2	...	O_n	k
-------	-------	-----	-------	-------	-------	-----	-------	-----

Figure 2: Structure of the GA chromosome.

In [13], an explicit bit-mask on the GA chromosome, separate from the weights, implements feature selection. Masked features are removed regardless of their weights. Because the objective function rewards parsimonious feature sets, partially relevant features may be masked prematurely. The GA here follows the population-adaptive mutation in [10] to accomplish feature selection. In the absence of an explicit mask, weights must be reduced to zero to remove features. Genes selected for mutation are randomly shifted in a range defined by $\pm$ one standard deviation away from the mean value of the gene across the population. If the jump results in values beyond the allowable ranges, the edges of the ranges replace the current values. If a weight drops below zero, it becomes zero and the feature is removed. In early generations, the standard deviation is large, enabling fast searches through wide areas of the solution space. Later, as the population begins to converge, small mutations enable fine-tuning in a local region of the solution space.

Recombination uses uniform crossover with 0.5 probability per gene. Selection operators employ tournament size 2 selection. Because mutation drives dimensionality reduction, the GA employs a fairly high probability of 0.1 mutations per gene within children. The population consists of

100 chromosomes. The GA runs until the population converges to a single fitness value within 0.001. Due to the nature of the mutation scheme, convergence typically occurs within 50 generations.

A fitness function evaluates trained classifiers by considering overall accuracy on known test data, the difference (balance) between class accuracies, and the number of features considered. Balance is measured as the difference between the highest class accuracy and the lowest class accuracy among all classes. The GA seeks higher overall accuracy while avoiding bias among the different classes using as few features as possible for classification. The GA-minimized cost function is:

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

where C_{pred} , C_{mask} , and C_{bal} , are the cost function coefficients. The authors' experiments employ the same values as in [10]: $C_{pred} = 25.0$, $C_{bal} = 10.0$, and $C_{mask} = 1.0$. The GA places maximum emphasis upon achieving high accuracy; maintaining balance and reducing dimensionality are secondary goals. Adjustment of the above parameters will tune the GA to emphasize alternate goals.

To assess the optimization capability of the GA for each similarity measure, results for unoptimized classifiers are also obtained. For each measure, K -values are chosen for the untrained classifiers as the average k selected by the top five weight-only optimized classifiers in the hope that the GA generally selects appropriate k -values for the size and shape of each dataset considered.

2.5 Datasets

Often one cannot obtain an equal number of patterns for each class. Knn classifiers demonstrate a bias towards choosing the most frequent class, since a large number of neighbors may belong to the majority class regardless of feature space position. To avoid bias, class-balanced training, test, and bootstrap subsets are created for each GA run. Additionally, the fitness function favors balanced performance by reducing the voting bias. For each GA experiment, datasets are randomly split up into class-balanced training and test sets, with remaining patterns reserved for validation. After convergence, the trained classifier's performance is assessed over the remaining patterns using a variant of the bootstrap test method [5] in order to obtain an unbiased accuracy estimate and avoid overfitting.

To compare accuracy and performance of various optimized and unoptimized knn classifiers, four UCI datasets are selected. These datasets represent real-world classification problems and are frequently used to compare newly-developed algorithms. In addition, two molecular biology datasets describing protein solvation are included. The specific datasets employed are discussed below. All of the features in the datasets have been normalized to range from 1.0 to 10.0. Any other modifications to the datasets are listed below.

UCI Datasets – Experiments are conducted on the Pima Diabetes, Wisconsin Breast Cancer, Heart-statlog, and ionosphere datasets. Each dataset is split into 2 classes. Pima contains 6 continuous and 2 discrete features and has no missing values. The breast cancer data contains 9 continuous features. 16 missing measurements from the sixth feature are replaced by the mean value for that feature across all instances. Heart-statlog contains 13 continuous features with no missing values, while ionosphere consists of 34 continuous features with no missing values. The second feature contains a single value across all samples, so it is removed from consideration, reducing the dataset to 33 features before use.

Water Conservation – This set describes the biochemical environment of water molecules bound to protein surfaces. Ligands (molecules) may bind to the protein at a water molecule's location. Upon binding, water molecules will either form a hydrogen bond bridge between the protein and the ligand, and hence their position is conserved, or the water will be displaced from its position. This dataset consists of 8 numeric features characterizing the environment of water molecules in 30 unrelated protein structures. The classifier attempts to distinguish conserved from displaced water molecules using a minimal set of weighted and offset-shifted features. It consists of 3405 conserved and 2137 displaced water molecules. Due to the high degree of overlap between class distributions and the unbalanced class sizes, this dataset tests the classifier's ability to maintain predictive balance between classes.

Water Solvation – This dataset consists of all surface water molecules from the same 30 proteins as above, and an equal number of surface points containing no water molecules, for a total of 11,084 samples. Six features describe the biochemical environment of each location. The goal is to distinguish solvation sites from non-sites with high accuracy. As with the conservation data, there is a high degree of overlap between class distributions. The primary goal of classifier optimization is accuracy amplification. Further information regarding the water datasets may be found in [10].

For each dataset, 20 GA runs are conducted for each of eight classifiers. This includes untrained Euclidian, cosine, and Pearson-based knn classifiers (to obtain an estimate of the uncertainty arising from randomly splitting data into training, test, and validation sets), weight-optimized Euclidian, cosine, and Pearson-based knn classifiers, and finally weight and offset-optimized cosine and Pearson-based knn classifiers. For each dataset, the average bootstrap accuracy, average bootstrap accuracy balance, and the average number of features used for each classifier provide a robust basis for comparison.

3 Results & Discussion

3.1 Overall Classification Accuracy

For each dataset, the GA conducted 20 runs using data randomly split into training, test, and bootstrap validation subsets for each run. Table 1 shows the mean and standard deviation of overall accuracy for the three unoptimized and five optimized classifiers over each dataset. “eucUn” represents the untrained Euclidian-based k nn classifier, while “eucW” is weight-optimized Euclidian k nn. For the cosine similarity k nn classifiers, “cosUn” represents unoptimized classification, “cosW” is weight-optimized classification, and “cosW+O” is weight + offset-optimization. For Pearson k nn, “pearUn” is unoptimized classification, “pearW” is weight-optimization, and finally “pearW+O” represents the weight + offset-optimized classifier. The best result is shown in **bold**, as is any other result not found to have a significantly lower mean accuracy rate using t-tests with significance level $\alpha = 0.05$.

In general, the cosine and Pearson classifiers using both weight and offset optimization tend to have the best or near-best classification rates for each dataset, with the exception of the breast cancer data. The cosine and Pearson classifiers optimizing only weights often perform as well as their offset-optimized counterparts, though often with slightly larger standard deviations. Interestingly, the offset-optimized classifiers for Pima diabetes outperform both unoptimized and weight-only optimized classifiers at a statistically significant level. Hence, for some datasets, offset optimization does provide improved performance for both cosine and Pearson-based classifiers. This is not always the case, as there is often not a statistically significant difference in performance between the weight/offset and weight-only classifiers.

In general, optimized classifiers using any of the similarity measures tend to perform at similar levels. The weight-optimized cosine and Pearson classifiers do not outperform their Euclidian counterpart on the Pima diabetes dataset, but when optimizing offsets as well as weights, they outperform the Euclidian classifier. For water solvation, the two optimized Pearson classifiers outperform all other classifiers at a significant level, though offset optimization does not provide additional performance. Figure 3 provides box-plots of the accuracy distributions for each classifier over each dataset. The boxes have lines at the upper and lower quartile values and at the median value. Whiskers are extended to show the range of the data, while outliers are represented with a star. If there are no outliers, a dot is placed at the bottom whisker. Figure 3 (a) represents Pima diabetes results, (b) shows Wisconsin breast cancer accuracies, (c) gives heart-statlog results, (d) presents the results for the ionosphere dataset, while (e) and (f) demonstrate water conservation and solvation accuracies, respectively. In general, optimized classifiers generally significantly outperform their unoptimized counterparts while achieving similar or tighter standard deviations. For both the diabetes and breast cancer datasets, offset optimization improves the performance of Pearson and cosine classifiers over weight op-

timization alone.

3.2 Classification Balance

Table 2 presents the average balance of bootstrap accuracy over each experiment for each dataset. As before, the best results and those not significantly different are highlighted in bold. In general, the weight optimized Euclidian and weight+offset optimized Pearson and cosine classifiers exhibit the best balance. The training dataset is always class-balanced and the GA exerts equal selective pressure for balance regardless of classifier type, so one does not expect a large difference in balance between optimized classifiers. The difference in balance is more visible when comparing optimized with unoptimized classifiers. With the exception of the water datasets, the untrained Pearson classifier often exhibits very poor accuracy balance, even though the training sets are balanced. The balance shows marked improvement after GA optimization, both for weight-only and weight/offset optimizations for the breast and ionosphere datasets. For the breast and heart datasets, weight/offset optimized Pearson and cosine classifiers exhibit better balance than their weight-only counterparts, suggesting that allowing offset optimization permits the GA to find a better compromise between class distributions resulting in less classification bias for some datasets. These results indicate that even for balanced training data, the balance term in the GA fitness function does tend to reduce the classification bias in the final optimized classifiers for several datasets.

3.3 Feature Dimensionality Reduction

Table 3 presents the average number of features used by optimized classifiers to highlight the dimensionality reduction capabilities of the GA. The best results and those not significantly different are highlighted in bold. The GA places equal selective pressure for dimensionality reduction for each type of optimized classifier, so it is not surprising that each optimized classifier typically does not employ significantly fewer features than any other optimized classifier. Most datasets exhibit anywhere from a 25 to 50 percent reduction in dimensionality versus untrained classifiers. This reduction is less drastic than pure feature selection typically provides because features with low weights may still contribute to classification accuracy. In the ionosphere data, 14 to 15 features remain after optimization. It may be the case that for datasets with large dimensionality, application of a traditional feature selection algorithm followed by weight optimization of the remaining features by a GA will result in better accuracy and improved feature reduction. For datasets with fewer features, the GA effectively removes spurious features while determining the relative importance of remaining features through weight optimization.

3.4 Knowledge Discovery for Water Conservation

The weight-only optimized Pearson classifier resulted in one outlier result of about 71.5% bootstrap accuracy for the water conservation data, as seen in figure 3 (e). To determine whether this outlier is the result of a fortunate

Figure 3: Boxplots comparing classifier accuracy distributions for each dataset. The classifiers are: unoptimized Euclidian, weight-optimized Euclidian, unoptimized cosine, weight-optimized cosine, weight & offset-optimized cosine, unoptimized Pearson, weight-optimized Pearson, weight & offset-optimized Pearson.

Table 1:
Classifier Accuracy - Mean(stdDev)

dataset	eucUn	eucW	cosUn	cosW	cosW+O	pearUn	pearW	pearW+O
diabetes	69.07(2.59)	71.51(1.60)	63.60(1.31)	65.89(2.97)	72.56(1.44)	62.13(2.97)	65.63(2.94)	72.01(2.42)
breast	96.40(1.01)	94.48(1.23)	87.72(1.62)	86.87(2.21)	94.95(1.80)	76.83(14.97)	87.63(2.16)	94.94(1.40)
heart	79.37(2.13)	79.59(3.23)	77.11(3.23)	79.33(2.41)	79.30(3.06)	76.85(3.24)	78.54(4.05)	78.11(4.09)
ionosphere	77.38(5.54)	83.53(3.80)	65.09(4.72)	83.54(4.16)	83.98(2.83)	76.69(2.56)	84.79(3.05)	83.96(2.94)
water cons	59.41(0.80)	62.42(0.87)	59.70(0.76)	62.81(1.92)	62.94(0.79)	59.71(0.90)	63.07(2.13)	62.81(0.60)
water solv	65.66(0.61)	67.90(0.91)	65.68(0.56)	67.96(0.69)	68.11(0.68)	65.27(0.53)	68.65(0.62)	68.53(0.71)

Table 2:
Classifier Accuracy Balance - Mean(stDev)

dataset	eucUn	eucW	cosUn	cosW	cosW+O	pearUn	pearW	pearW+O
diabetes	10.95(3.20)	9.75(1.82)	12.14(3.58)	10.04(2.98)	10.29(2.72)	11.33(3.51)	12.23(5.29)	10.77(2.24)
breast	3.91(0.93)	5.77(2.33)	7.94(1.73)	7.84(2.56)	5.31(2.26)	33.26(37.78)	7.81(1.65)	5.02(2.58)
heart	14.10(1.70)	15.24(4.27)	17.19(3.79)	16.75(6.04)	15.33(3.63)	16.05(3.00)	16.05(3.20)	15.19(2.60)
ionosphere	36.07(13.86)	23.90(7.34)	62.99(14.65)	22.86(6.61)	22.58(8.56)	44.02(4.85)	22.34(4.38)	22.92(6.07)
water cons	3.87(0.82)	4.06(0.69)	3.98(1.04)	4.07(0.66)	4.44(1.29)	4.03(0.99)	4.04(1.29)	4.00(0.63)
water solv	4.28(1.78)	3.15(1.02)	3.90(1.47)	3.01(0.84)	3.38(1.16)	5.34(1.83)	3.07(0.73)	3.20(1.31)

splitting of the dataset into train, test, and validation subsets or the result indicates a significant optimization, 20 bootstrap tests using randomly split subsets for training and validation were conducted using the optimized parameters for weights and the k-value. Of the eight features in the dataset, the second feature has an unnormalized weight of 1.054, the weight of the third feature is 100.0, and the weight of the sixth is 93.252. All other features are removed from consideration. The optimized k-value is 10. Referring to table 1, the unoptimized Pearson classifiers achieve only 59.71% average bootstrap accuracy, while the weight-optimized Pearson classifier achieves an improved average 63.07% accuracy. For this particular optimization, the 20 bootstrap experiments achieve an average of 70.85% accuracy with 0.99% standard deviation, while maintaining an average accuracy balance of 5.95% with 2.02% standard deviation. The outlier represents a legitimate optimization, rather than an artifact of randomly splitting the data. While this paper focuses on the average optimization of each classifier technique, it is not unreasonable to examine the average performance of the best optimization of each classifier when comparing results to other classification techniques.

The optimized weights of selected features enable knowledge discovery. The relative weight of each feature in the trained classifier indicates the feature's importance to classification. The second feature receives smallest weight in the optimized classifier above. In the water conservation data, the second feature corresponds to atomic hydrophilicity, a measure of a water molecule's ability to make hydrogen bonds with neighboring atoms [7, 10]. The third and sixth features, corresponding to B-value and mobility, receive large weights. Both are varying measures of the thermal activity of the water molecule and its surrounding environment [10]. The weights indicate that the best classification rate occurs when giving the most consideration to the thermal mobility of the water and its environment while giving less consideration to the hydrophilicity. This type of insight is not easily gathered from classifier algorithms that do

not maintain the relative independence of features. The optimization techniques described here not only enhance classification accuracy, they also facilitate knowledge discovery for the problem domain of interest.

4 Conclusions

In conjunction with weight optimization, offset optimization represents an effective method for improving the performance of k -nearest neighbors classifiers employing cosine similarity or Pearson correlation. For some datasets, offset optimization improves classification accuracy and balance over weight-only optimized classifiers. While unoptimized Euclidian k nn often outperforms unoptimized cosine or person k nn, the GA-facilitated optimization techniques presented here allow the cosine and Pearson k nn to match or outperform optimized Euclidian k nn. The weight optimization maintains feature independence while discovering relative feature importance, potentially providing novel insight into the problem domain under consideration. When there may be uncertainty in feature measurements, as is often the case in biological or medical data, the use of cosine similarity or Pearson correlation may provide improved classification over Euclidian distance in the presence of GA optimization.

Because k nn classifiers are instance-based algorithms, meaning they store training samples, they are relatively slow compared to other types of classification algorithms, such as the parametric naive Bayes algorithm. Hence, k nn optimization proceeds more slowly than weight optimization with simpler algorithms [12]. However, the flexibility provided by offset optimization for k nn may provide an improvement in accuracy, justifying increased computational effort for some applications.

Future research should explore the ability of the population-adaptive mutation scheme to effectively search a wide portion of the search space before reaching convergence. While the technique appears successful for di-

Table 3:
Number of Features Used per Classifier - Mean(stdDev)

dataset	eucUn	eucW	cosUn	cosW	cosW+O	pearUn	pearW	pearW+O
diabetes	8.0(0.0)	5.30(1.10)	8(0.0)	5.65(1.59)	6.00(0.84)	8(0.0)	5.90(0.89)	5.80(1.21)
breast	9.0(0.0)	3.90(1.18)	9.0(0.0)	7.05(1.24)	4.10(1.55)	9.0(0.0)	7.00(1.18)	3.65(0.91)
heart	13.0(0.0)	8.60(1.60)	13.0(0.0)	8.40(1.59)	8.20(1.75)	13.0(0.0)	8.10(1.41)	8.50(2.01)
ionosphere	33.0(0.0)	14.20(2.62)	33.0(0.0)	14.20(2.93)	15.35(3.15)	33.0(0.0)	15.00(1.34)	14.45(3.02)
water cons	8.0(0.0)	4.65(0.73)	8.0(0.0)	5.35(1.01)	5.10(1.22)	8.0(0.0)	5.60(1.32)	5.25(0.83)
water solv	6.0(0.0)	4.30(0.71)	6.0(0.0)	4.35(0.79)	4.85(0.73)	6.0(0.0)	5.20(0.75)	5.25(0.83)

dimensionality reduction, its search properties should be compared with more traditional mutation techniques and other forms of real-value optimization, such as those applied in evolutionary strategies or artificial immune system algorithms. Specific operators for real-coded GAs, such as BLX-alpha or simplex crossover should be considered. Offset shifting and feature weighting may be useful for other types of classification algorithms or similarity measures, such as Mahalanobis distance. Feature weighting and offset shifting may not perform well in the presence of data with many features due to the large search space. The application of a GA-based feature selection algorithm as a preprocessing step may quickly eliminate features with low relevance, allowing the feature weighting and offset-shifting techniques presented here to be applied to data of larger dimensionality.

Bibliography

- [1] 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/~mllearn/MLRepository.html>.
- [2] 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*, 97:262–267, 2000.
- [3] E. Han and G. Karypis. Centroid-based document classification: Analysis & results. In *Principles of Data Mining and Knowledge Discovery: fourth European Conference*, pages 424–431, 2000.
- [4] 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*, pages 53–65, 2001.
- [5] A. K. Jain, R. C. Dubes, and C. C. Chen. Bootstrap techniques for error estimation. *IEEE Trans Pat. Anal. & Machine Intell.*, 9(5):628–633, Sept. 1987.
- [6] J. D. Kelly and L. Davis. Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm. In *4th Int. Conf. Genet. Algorithms & Applications*, pages 377–383, 1991.
- [7] 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.*, 23:536–547, 1995.
- [8] M. Kuramochi and G. Karypis. Gene classification using expression profiles: a feasibility study. In *Proceedings 2nd IEEE Int. Symp. on Bioinf. & Bioeng.*, pages 191–200, 2001.
- [9] H. Liu and H. Motodota. *Feature Selection for Knowledge Discovery and Data Mining*, pages 73–95. Kluwer Academic Publishers, Boston, MA, 1998.
- [10] M. R. Peterson, T. E. Doom, and M. L. Raymer. Ga-facilitated knowledge discovery and pattern recognition optimization applied to the biochemistry of protein solvation. In *GECCO 2004 Proceedings, LNCS 3102*, pages 426–437, 2004.
- [11] 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 *Int. Conf. Genet. Algorithms 93*, pages 557–564, 1993.
- [12] M. L. Raymer, T. E. Doom, L. A. Kuhn, and W. F. Punch. Knowledge discovery in medical and biological datasets using a hybrid bayes classifier/evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 33(5):802–813, 2003.
- [13] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain. Dimensionality reduction using genetic algorithms. *IEEE Trans Evol. Comp.*, 4(5):164–171, 2000.
- [14] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pat. Rec. Letters*, 10:335–347, 1989.
- [15] G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Trans Pat. Anal. & Machine Intell.*, 1:306–307, 1979.